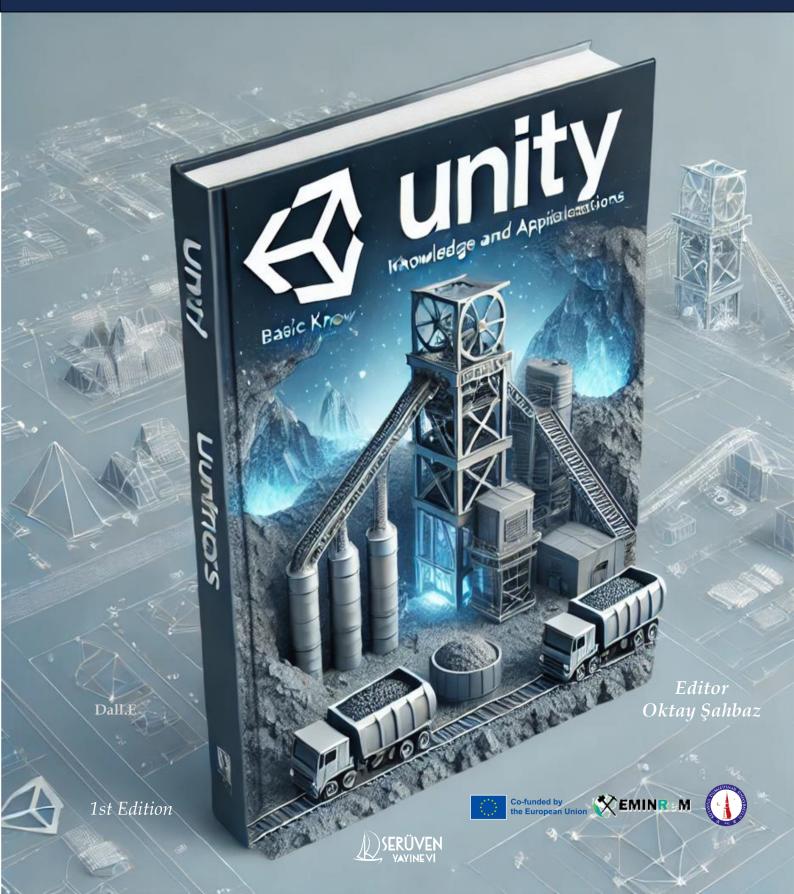
With Applications for Mining

Kaan Erarslan



Genel Yayın Yönetmeni / Editor in Chief • C. Cansın Selin Temana Kapak & İç Tasarım / Cover & Interior Design • Serüven Yayınevi Birinci Basım / First Edition • © Kasım 2024 ISBN • 978-625-6172-65-4

#### © copyright

Bu kitabın yayın hakkı Serüven Yayınevi'ne aittir. Kaynak gösterilmeden alıntı yapılamaz, izin almadan hiçbir yolla çoğaltılamaz. The right to publish this book belongs to Serüven Publishing. Citation can not be shown without the source, reproduced in any way without permission.

#### Serüven Yayınevi / Serüven Publishing

Türkiye Adres / Turkey Address: Kızılay Mah. Fevzi Çakmak 1. Sokak Ümit Apt No: 22/A Çankaya/ANKARA Telefon / Phone: 05437675765 web: www.seruvenyayinevi.com e-mail: seruvenyayinevi@gmail.com

**Baskı & Cilt / Printing & Volume** Sertifika / Certificate No: 47083

With Applications for Mining

1<sup>st</sup> Edition

Kaan Erarslan

Editor Oktay Şahbaz





# PREFACE

In the first quarter of the twenty-first century, computer electronic hardware and software are developing at a dizzying pace. A large technological matrix is emerging with artificial intelligence, deep learning, the Internet of Things (IoT), smart systems, and visual communication elements. Visualization and imaging technologies continue to develop as an independent area and a component of this large matrix.

Computer-aided modeling, visual communication tools, and classical methods provide extraordinary designs and products. In the last quarter century, one of the breakthroughs that have affected our lives has been the platforms called real-time development (game) engines, which generally appeal to every professional discipline and business area. It is a useful and empowering new tool for visual design and application developers to have in their software toolbox.

Developments in Virtual-Augmented Reality (VR-AR) and Serious Games are the subject of many innovative studies in educational curriculum and design, and their positive contributions are revealed by scientific research.

Educational materials developed in this field can be used on computers, the web, mobile devices, special headsets, and smart glasses. Therefore, all device options with their unique character offer a wide working area for application developers.

Unity is one of the most important cross-platform software for developing applications in VR-AR and Serious Games (Gamification). Unity, a real-time development engine with a history of nearly a quarter of a century, is software that can be used to develop all VR-AR and Serious Game applications. Unity offers application opportunities in all educational disciplines with these features. For this, Unity knowledge and engineering field knowledge is required.

"Introduction to Unity, a Real-Time Development Engine with Mining Applications" aims to provide the basic knowledge and applications in these subjects. The focus is on the visual design and application aspect rather than game development. Educational materials are mostly video-based visual materials and limited written documents due to the visual nature of the subject. In the last decade, many comprehensive and detailed video documents have been prepared in this field. It is possible to access the training records, which are paid or free, on the Internet. In fact, the most important source in this field is the constantly developing visual and written archive on the internet. Written documents may become outdated because of developments in computer software. However, these notes have been prepared to provide a written document about general information and some applications in a way that is suitable for those learning from scratch.

In the book, Unity essentials are explained, as well as general information about C# coding is also provided. Virtual and Augmented Reality topics were discussed. For the training to be successful, application studies and video materials should be considered together with the presented tutorials. In other words, practice studies, videos, and written documents are all components of a whole, parts of understanding the subject better and being successful. Apart from this, the most important key to success is individual interest, motivation, and efforts during and after the training process.

Additional to the basic knowledge of Unity, mining engineering application examples are also included. The book, which consists of tutorials, also has the feature of being educational material.

# Acknowledgments

Acknowledgments are extended to the European Union, Brussels for supporting the 101082621-EMINReM-ERASMUS-EDU-2022-CBHE "Master Program in Eco-Mining Engineering and Innovative Natural Resources Management - EMINRem" project and to the Kütahya Dumlupinar University Scientific Research Projects Unit for providing laboratory facilities with the "Search and Rescue Training in Mines Using VR/AR Technologies"-DPÜ-BAP 2022-63 project.

# Files in Share

The course book, asset packages, C# scripts, materials and video tutorials are presented in the folders and files.

https://drive.google.com/drive/folders/156AOkT8JP6khzlcnr02u7BzcIzeWZnjC?usp=sharing

# Table of Contents

PREFACE	1
Acknowledgments	2
Files in Share	2
INTRODUCTION	6
1. INSTALLATION OF NECESSARY SOFTWARE	7
1.1.Unity 3D installation	7
1.2.Java JDK	9
1.3.Android Studio	9
1.4.Xcode for IOS Systems	
1.5.Vuforia	
2. UNITY PANELS	
2.1.Using Unity 3D Panels	
2.2.Game Tab/Mod	
2.3. Game and Scene Perspectives	
3. SCENE DESIGN	
3.1.Adding a GameObject	
3.2.Object Properties	
3.3.Materials	
3.4.Lights	25
3.4.1 Point Light	25
3.4.2 Spotlight	
3.4.3 Area Light	
3.4.4 Reflection Probe	
3.4.5 Light Probe Group	
3.5.Rigidbody	
3.6.Physic Material	
3.7.Particle System	
4. C# SCRIPTS FOR UNITY	
4.1.Basic C# Information	
4.2.C# Scripts in Unity	
4.3.Console Messages	
4.4.Variables	
4.5.Methods/Functions	53
4.6.Arithmetical Operators	

4.7. Conditional Statements and if statements	57
4.8.Array	60
4.9.List Variables	61
4.10.Basic C# Coding for Unity	65
4.11.Object and Component Access	68
4.12.Collision/Interaction of Objects in Unity – On Collision	73
4.13.Adding Gestures with Mouse and Keyboard	79
4.14.Controlling and Moving Objects	82
4.15.Interacting and Animating with Rigidbody	86
4.16.Activating/Deactivating Objects (SetActive) and Destroying Objects (Destroy)	88
4.17.Object Control with Mouse	90
4.18.Cloning Objects – Prefab and Instantiate	92
4.19.Adding Objects with Instantiate Coding – Spawn	93
4.20.Raycast – Collision Control by Ray Spreading	107
4.21.Ray Emission from Camera	111
4.22.Ray Propagation from Mouse	113
4.23.FPS/P (First Person Shooter/Perspective) Applications	114
5. ADDING SCENE OBJECTS AND ASSET RESOURCES	127
5.1.Unity Asset Store	127
5.2.Sketchfab	127
5.3.GrabCAD, Rigmodels and 3DWarehouse-Sketchup	127
5.4.Terrain	128
5.5.Terrain + Standard Assets Using Unity Asset Store	136
6. DEPLOYING PLATFORMS	143
6.1.Build Settings	143
6.2.Cameras	146
6.3.Parent Child Relation	148
6.4.Skybox	148
6.5.Adding Audio and Video files	150
6.6.Animation	153
6.7.Switching Between Cameras in Unity	158
6.7.1. 1 <sup>st</sup> Method	158
6.7.2. 2 <sup>nd</sup> Method	163
7. LIGHT AND TEXTURE WITH URP	167
7.1.Universal Render Pipeline (URP) – Post Processing Volume - Glow Effect	167
8. USER INTERFACE (UI)	173

8.1.UI Text-Button-World Space- Interface Objects-3D Texts	173
8.2.Switch between scenes-UI Button	179
8.3.UI Text – World Space	
9. FPS VE TPS APPLICATIONS	
9.1.FPS-First Person Shooter and RPG-Role Playing Game	
9.2.TPS Third Person Shooter-Starter	
9.3.Third Person Character Controller – Armature Change	
9.4.Transferring Blender Designs to Unity	211
10. VIRTUAL REALITY – VR APPLICATIONS	216
10.1.Virtual Reality (VR) Application for Cardboard Devices	216
10.2.Configuring the HelloCardboard Scene	221
10.3.Moving in a VR Scene	230
11. AUGMENTED REALITY – AR APPLICATIONS	237
11.1.Augmented Reality – AR	237
11.2.Vuforia AR Engine	237
11.3.VUFORIA integration in UNITY project	
11.4.Getting the License Key and Downloading Database from Vuforia to Unity	250
11.5.AR-Vuforia and Multi-Image Target	
11.6.Deployment into Mobile Devices	
11.7.Augmented Reality with Video Player	
11.8.Vuforia AR – Ground Plane	
11.9.Similar Examples for Mining	
12. USE OF ANIMATION AND ADOBE MIXAMO CHARACTERS IN UNITY	
12.1.Relevant C# Script Codes	
BIBLIOGRAPHY	
AFTERWORD	

# **INTRODUCTION**

Unity and similar real-time development engines are used to develop games, VFX movie scenes, and science and educational materials. With its cross-platform feature, it is possible to develop applications for many different device options, such as computers, mobile devices, game consoles, virtual reality (VR) headsets, and augmented reality (AR) smart glasses.

Unity, which can be used in almost every professional discipline, has applications in education, fine arts, health sciences, and engineering.

To become an educational material or field application developer in any professional field, it is necessary to learn the basic information of the Unity engine first. This book, which is prepared based on tutorials according to the subject titles for beginners, mining engineering, which includes many disciplines from earth sciences to construction machinery, from electrical systems to office management, from occupational health and safety to education, from management to labor law, has been preferred as the application area.

#### **Book content:**

Installation Editor's introduction **Physical materials** Solid objects and gravity Particle system C# basics Basic C# coding for Unity Collision control with C# Object control with mouse with C# Raycast C# coding Scene design and adding objects Cameras Terrain design Output for PC (EXE), Mobile (APK), VR Headset and AR Smart-Glass Adding audio and video Simple animations Working with UI (User Interface) Scene transitions with UI First Person (FP) and Third Person (TP) applications for PC and mobile devices Virtual reality (VR) applications (for Cardboard) Augmented Reality (AR) applications (with mobile devices) Using animation-animator in Unity with Adobe Mixamo library and C# coding

Open pit, underground mine, ore preparation-enrichment facilities, ventilation, various machines sample application applications developed for the titles It is located as.

The book can be used as educational material for beginners as well as for teaching levels.

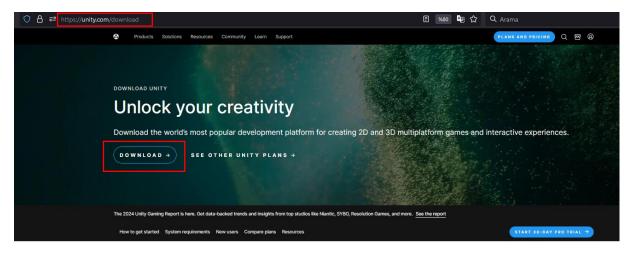
# **1. INSTALLATION OF NECESSARY SOFTWARE**

Within the scope of the course, all the applications will be developed with **the Unity Real Time Development Engine**. On the other hand, different software will be needed for the applications to be developed as **PC-based**, **Web-based**, **mobile** and mobile-**AR/VR**.

The software that may need to be installed for the course, depending on the applications, is listed below. Here, Unity is absolutely required. Others are conditional and optional.

# 1.1.Unity 3D installation

It is possible to reach the download page from the relevant page that will open after browsing with Unity on Google or from the link <u>https://unity.com/download</u>



# Create with Unity in three steps

1.	Do	wnlo	ad	the	Unit	ty Hu	ıb	

Follow the instructions onscreen for guidance through the installation process and setup. Download for Windows Download for Mac Instructions for Linux

#### 2. Choose your Unity version

Install the latest version of Unity, an older release, or a beta featuring the latest indevelopment features. Visit the download archive

#### 3. Start your project

Begin creating from scratch, or pick a template to get your first project up and running quickly. Access tutorial videos designed to support creators, from beginners to experts. Access our Pro Onboarding Guide

First, create a Unity account with the email you will use in all your work. Since the system matches email accounts, it is important to have the same email account (i.e. Unity asset store).

Now we can download the **Unity Hub** program. This program plays a central role in all installation and project operations. It is recommended not to use the beta trial version.

On the Hub, you can make your first download by clicking the **Add** button in the **Install** section. The system will recommend the version with **LTS** (Long-Term Support). Start the installation by selecting the latest recommended LTS version (e.g. 2022.3.50f).

Unity Hub 3.8.0		- 🗆 X
× ¢	Installs	te Install Editor
Projects	All Official releases Pre-releases Q Sear	rch
Installs		
🔄 Learn 🚢 Community	Unity (2023.2.4f1)       C:\Program Files\Unity\Hub\Editor\2023.2.4f1\Editor\Unity.exe       Android     UWP       WebGL     Windows	٥
	Unity (2022.3.16f1) LTS     C:\Program Files\Unity\Hub\Editor\2022.3.16f1\Editor\Unity.exe     Android UWP WebGL Windows	¢
🞍 Downloads		

Before installing, be sure to mark the following additional subcomponents:



On the Hub, it is possible to add another version by clicking **Add** from the **Install** tab. For older versions, **download archive** selection should be made.

Install Unity Edit	or	×
Official releases	Pre-releases	Archive
		g for? Visit our download archive for access to Long-Term Support program releases.
🛛 Beta program	webpage	

C Products So	olutions Resources Community Learn	Support		PLANS AND PRICING Q E KE	
		Unity download	d archive		
	То	learn more about the release version	ons, visit <u>Unity Releases</u> .		
Unity 6 2023 2022 2021 2020 2019 2018 2017 Unity 5					
All versions LTS (Default) Tech Stream					
Version	Release date	Release notes	Hub installation	Downloads	
2022.3.40f1	30 Tem 2024	Read	INSTALL →	See all	
2022.3.39f1	23 Tem 2024	Read	INSTALL →	See all	
2022.3.38f1	16 Tem 2024	Read	INSTALL >	See all	

Unity's website will open, and you will be given the opportunity to choose a version.

For example, you can find and download Unity 2019.1.14. This version is still preferable for **VR** (Virtual Reality) and **AR** (Augmented Reality) applications. However, it does not support the latest updates.

#### 1.2. Java JDK

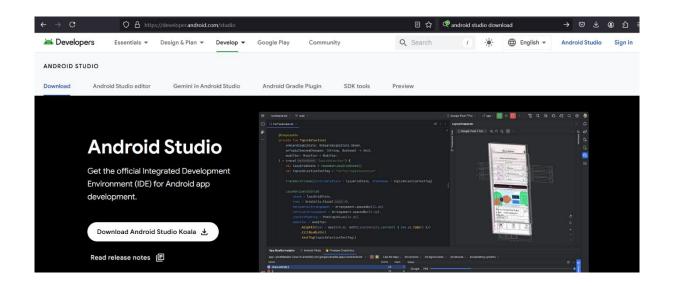
If <u>not</u> installed with Unity, **Java SE JDK** must be installed manually. To do this, a search on Google with "Java SDK download" will reach the download page on **Oracle's** website. Windows users should download and install the relevant package (zip or exe) from this tab. The macOS version must be downloaded for the IOS operating system.

C O A https://www.oracle.com/java/technolog	gies/downloads/#jdk1	7-windows	目☆	Q Arama	⊠ 🍕 🌞
ORACLE Q Products In	dustries Resource	s Support Events D	Developer Partners	② View Accounts	Contact Sales
Java downloads Tools and resources Java archive					
	ے Looking f	or other Java downlo	oads? OpenJDK Early Ad	ccess Builds JRE for Consumers	
Java 17 available now					
Java 17 LTS is the latest long-term support release for the Java redistribute, at no cost, under the Oracle No-Fee Terms and Co		pinaries are free to use in pro	oduction and free to	Learn about Java SE Subscription	
JDK 17 will receive updates under these terms, until at least Se	ptember 2024.				
Java SE Development Kit 17.0.2 downloads					
Thank you for downloading this release of the Java <sup>™</sup> Platform, Standard Edition Development Kit (JDK <sup>™</sup> ). The JDK is a development environment for building applications and components using the Java programming language.					
The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.					
Linux macOS Windows					
Product/file description	File size	Download			
x64 Compressed Archive	171.34 MB	https://download.oracle.c	com/java/17/latest/jdk-17_wi	ndows-x64_bin.zip (sha256 🖄)	

#### 1.3.Android Studio

If the required components are <u>not</u> installed during Unity installation, Android Studio should be installed for Android mobile applications. You can access the relevant page by typing "android studio

download" on Google to do this. From the table at the bottom of the open page, the latest version for Windows or Mac should be downloaded and installed.



#### **1.4.Xcode for IOS Systems**

For compilation processes of IOS mobile systems, the **XCode** package should be downloaded and installed. The relevant page can be reached by typing **XCode** in Google. First, it is necessary to create a user account. At this stage, your Apple phone must be ready due to security protocols.



#### 1.5.Vuforia

One of the pioneering companies that develops augmented reality applications is **PTC**. Mobile AR applications are developed practically using a product called **Vuforia**. We can reach the download page by typing **Vuforia SDK download** on Google. As a developer, download and install the package related to Unity.

vuforia <sup>,</sup> engine <sup>,</sup> developer portal	Home Downloads	Library Support	Pricing	Log In Register
SDK Samples Tools				
Release Version 10.25	~			
By downloading the Vuforia Engine SDK	Samples and Tools, you ag	ree to the developer agree	ment.	
Vuforia Engine 10.25 Use the Vuforia Engine SDK to build augme devices and digital eyewear. Vuforia Engine and Visual Studio.				
Download for Unity add-vuforia-package-10-25-4.unitypackage	je	Download		
Download for Android vuforia-sdk-android-10-25-4.zip		Download		

Make sure to create a user account in Vuforia. Although it is not compulsory, it is recommended to use the same email address as Unity.

Vuforia is Unity's official partner for applications in augmented reality, specifically for the **Hololens 2** headsets.

# 2. UNITY PANELS

# 2.1. Using Unity 3D Panels

First, a project should be started. To do this, select the **Project** tab on **Unity Hub** and perform the first step of creating a new project.

Unity Hub 3.8.0					-	- 🗆 X
	Project	S		A	dd 👻 Ne	ew project
Projects				Q s		
Installs	* %	NAME	CLOUD		EDITOR VERSION	
<ul> <li>Learn</li> <li>Community</li> </ul>	ŚŚ	IMVR C:\A UNITY\IMVR	CONNECTED	2 months ago	2022.3.16f1	\$ •••
	62	<b>GVR2</b> C:\A UNITY\GVR2	CONNECTED	4 months ago	2022.3.16f1	\$ •••
	50	<b>gvr11</b> C:\A UNITY\gvr11	CONNECTED	4 months ago	2022.3.16f1	÷ •••
	8	<b>GVR</b> C:\A UNITY\GVR	CONNECTED	4 months ago	2022.3.16f1	÷ •••
	62	<b>Area</b> C:\A UNITY\Area	NOT CONNECTED	5 months ago	2023.2.4f1	÷ •••
Downloads	50	<b>Atay1</b> C:\A UNITY\Atay1	NOT CONNECTED	5 months ago	2022.3.16f1	\$ •••
	2-	ConicCrusher	CONNECTED		2022 2464	~

If more than one version is installed, the version we want to use is selected first (*currently*≥2022.3.1.50f).

Unity Hub 3.8.0		- 🗆 X
	New project Editor Version: 2022.3.16f1 LTS 🗘	
i = All templates	Q Search all temp Q Filter	
Core	2D (Built- 😥 2023.2.4f1	
Sample	Core	
Learning	3D (Built-In Render Pipeline) Core	
	Universal 2D Core	3D (Built-In Render Pipeline) This is an empty 3D project that uses Unity's built-in renderer. Read more
	Core Universal 3D	
	High Definition 3D	Project name My project
		Cancel Create project

Here, the templates for the version are listed, and the name of the project (**Project name**) and folder (**Location**) are specified. By giving the name **Project1** to the first project and specifying its folder, our editor will be opened with **Create**.

Unity Hub 3.8.0		- 🗆 X
	New project Editor Version: 2022.3.16f1 LT	т  \$
i≡ All templates	Q Search all templates	3D (Built-In Render Pipeline) This is an empty 3D project that uses Unity's
<ul><li>Core</li><li>Sample</li></ul>	Core 2D (Built-In Render Pipeline)	built-in renderer.
Learning	3D (Built-In Render Pipeline) Core	PROJECT SETTINGS Project name My project
	Core	Location C:\A UNITY
	Core	Unity Organization KErarslan
	High Definition 3D	Connect to Unity Cloud ?
		Cancel Create project

The general appearance of the empty panels is as shown in the figure. In the upper left corner is the project's name, **Project1**, and the **SampleScene**, which shows which scene it is in ①.

**SampleScene** name is given by the system. When you look at the scene sections, there is a window at the top left that shows the scene named **SampleScene** and the **objects** (assets) in the scene, called Hierarchy<sup>2</sup>. Each asset (object) here can also be called a **Game Object**. When we right-click our mouse, the objects in the menu can be added.

At first, there appears to be only a **Main Camera** and a **Directional Light**. The part where the scene is located is in the window called **#Scene**<sup>3</sup>.

In the section titled **Project**<sup>(4)</sup>, there are **Favorites**, **Assets**, and **Packages**. Here are the files of the objects and packages that will be added to the project to be added to the scene. All the movements here are simultaneous and parallel with Windows Explorer.

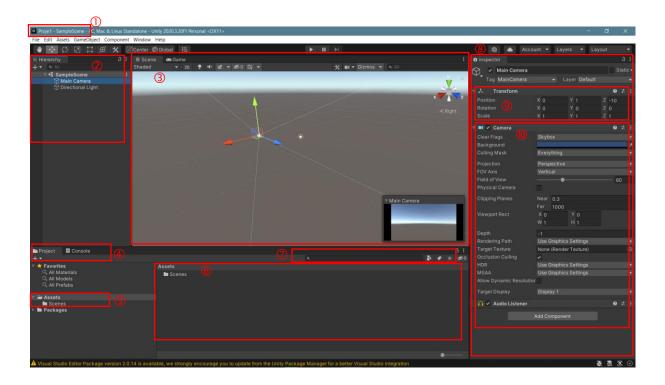
The most important and frequently used subheading in the **Project** window is the section called **Assets** because it is the section where all kinds of objects to be added to the scene are located. We can think of these assets as various **scene** objects such as **2D**, **3D** objects, materials, sound, light, video, effects, animation, C# codes, etc. A large window belonging to **Assets** is also positioned to show the assets inside **6**.

There is an asset search window just above this window  $\bigcirc$ . As can be seen in the figure, **Scenes** are also assets.

The Inspector window is where we will make arrangements for each scene component, we select in the Hierarchy window<sup>®</sup>. There may be dozens of sub-parameters specific to each of the hundreds of objects in Unity, and dozens of sub-parameters belonging to them. These will be visible in the **Inspector** window.

We see the details of the **Main Camera** selected in our empty scene in the **Inspector**. Under the checked part **Main Camera** that shows that the object is selected and in use, there is the **Transform** window where the objects in the scene, also known as **Game Objects**, are located, which are basically located in most of the scene's (**x**, **y**, **z**) coordinates; **Position**, rotational angles; **Rotation** and size scales; **Scale**. (9).

Again, in the editor of the **Main Camera**, **Inspector**, there is a section where the **Camera** properties are defined<sup>®</sup>. Below is the empty audio editor, **Audio Listener**. Below that is the **Add Component** button, which adds various properties to the selected object.



4	roje1 ·	Sample	Scene - PC, Ma	ac & Linux Star	ndal
File	Edit	Assets	GameObject	Component	Wi
	New S	icene		Ctrl+N	1
	Open	Scene		Ctrl+C	
	Open	Recent S	Scene		>
	Save			Ctrl+9	
	Save A	As		Ctrl+Shift+S	
	Save A	As Scene	Template		
	New P	roject			
	Open	Project			
	Save F	Project			
	Build :	Settings.		Ctrl+Shift+E	
	Build	And Run		Ctrl+E	
	Exit				

When you examine the project's menus, you will see File, Edit, Assets, GameObject, Component, Window, and **Help**. The **File** menu includes options for scene operations, file recording, project creation and platform operations.

There are editor operations in the **Edit** window. The main ones are copy, paste, select, play, freeze, name and similar operations. The most important operations are **Project Settings**, project-related arrangements and Preferences, pre-preferences for various settings.

The **Assets** menu is the section where transactions related to assets are located. Here, there is a very rich tab titled **Create** to create assets.

	<ul> <li>SampleScene - PC, Ma</li> </ul>		ndalone - U
dit	Assets GameObject	Component	Window
	Undo Selection Change		Ctrl+Z
	Redo Delete Game Obj	ects	Ctrl+Y
	Select All		Ctrl+A
	Deselect All		Shift+D
	Select Children		Shift+C
	Select Prefab Root	Ctrl+	-Shift+R
	Invert Selection		Ctrl+I
	Cut		Ctrl+X
	Сору		Ctrl+C
	Paste		Ctrl+V
	Paste As Child	Ctrl+	-Shift+V
	Duplicate		Ctrl+D
	Rename		
	Delete		
	Frame Selected		F
	Lock View to Selected		Shift+F
	Find		Ctrl+F
	Play		Ctrl+P
	Pause		-Shift+P
	Step	Ctr	I+Alt+P
	Sign in		
	Sign out		
	Selection		>
	Project Settings		
	Preferences		
	Shortcuts		
	Clear All PlayerPrefs		
	Graphics Tier		>
	Grid and Snap Settings.		

Assets	GameObject	Component	Window	Hel	p
	eate			>	Folder
Op De Re Co	ow in Explorer ben elete name opy Path ben Scene Addi ew in Package N	tive	Alt+Ctrl+C		C# Script 2D > Shader > Testing > Playables > Assembly Definition Reference TextMeshPro >
lm Ex Fir Se	port New Asse port Package port Package nd References In lect Dependen	n Scene		>	Scene Template Scene Template From Scene Prefab Prefab Variant
	fresh import		Ctrl+R		Audio Mixer
Ex Ru Up	import All tract From Prefi n API Updater odate UXML Sch				Material Lens Flare Render Texture Lighting Parameters Lighting Settings Custom Render Texture
110	Son I # Urolact				Animator Controller Animation Animator Override Controller Avatar Mask
					Timeline Signal
					Physic Material
					GUI Skin Custom Font UI Toolkit >
					Legacy >
ages					Brush

Also, the **Import New Asset** option is used to import a new asset object (file). **Packages** prepared outside Unity and in a format that Unity will accept can be imported into the project via the **Import Package** tab.

The **GameObject** submenu is a menu we can use when we want to add a new object, **GameObject**, to our scene from the objects in Unity. The same operations can be done in the **Hierarchy** window by right clicking our mouse.

The **Component** menu contains the content of the **Add Component** key that we use to assign various properties to objects.

In the **Window** submenu, the most important and frequently used sub-tabs, in addition to window arrangements, are **Asset Store** and **Package Manager**. Asset Store connects to the **Unity Asset Store** web page and allows us to include many free and paid asset packages in our project. The other option is the **Package Manager** option, which allows asset

packages to be managed and added to our project. Package Manager has an important function that manages standard Unity packages and the assets we add to our assets via the Asset Store.

GameObject	Component	Window	Help		
Create Er	npty	Ctrl+Shift+N			
Create Er	npty Child	Alt+	Shift+N		
Create Er	mpty Parent	Ctrl+	Shift+G		
3D Objec	:t		>		
Effects			>		
Light			>		
Audio			>		
Video			>		
UI			>		
Camera					
Center O	n Children				
Make Par	ent				
Clear Par	ent				
Set as fire	st sibling		Ctrl+=		
Set as las	t sibling		Ctrl+-		
Move To	View	Ctr	l+Alt+F		
Align Wit		Ctrl+	-Shift+F		
Align Vie	w to Selected				
Toggle A	ctive State	Alt+	Shift+A		

Component	Window	Help	
Add		Ctrl+Shift+A	
Mesh		>	
Effects		>	
Physics	Physics		
Physics 2	Physics 2D		
Navigati	ion	>	
Audio		>	
Video		>	
Renderir	ng	>	
Tilemap		>	
Layout		>	
Playable	!S	>	
Miscella	neous	>	
Scripts		>	
UI		>	
Event		>	

dow Help	
Panels	>
Next Window	Ctrl+Tab
Previous Window	Ctrl+Shift+Tab
Layouts	>
Collaborate	
Plastic SCM	
Asset Store	
Package Manager	
Asset Management	>
TextMeshPro	>
General	>
Rendering	>
Animation	>
Audio	>
Sequencing	>
Analysis	>
AI	>
LII Toolkit	>

This section, which is a general introduction, can be understood by practicing while developing a project. The only way to understand and learn menus, their dozens of submenus, and hundreds, thousands of control parameters is to use them in the project.

Package Manager	l		: = ×
+ ▼ Packages: My Assets ▼ Sort: Name ↓ ▼	Filters 🔻		
		∓ +	3D Free Modular Kit
		<u>+</u>	
		Ŧ	
		Ŧ	Version 1.2 - September 07, 2020 asset store
		Ŧ	
		Ŧ	★ WEBGL DEMO ★ (http://blitz3dfr.free.fr/)
		Ŧ	
		Ŧ	freeware games/apps.
		Ŧ	Images & Videos
		Ŧ	
		Ŧ	
		*	
		<u>+</u>	
			View images & videos on Asset Store
			Package Size Supported Unity Versions
			Size: 13,2 MB (Number of files: 53) 2018.4.15 or higher
			Purchased Date
			Release Details

In conclusion,

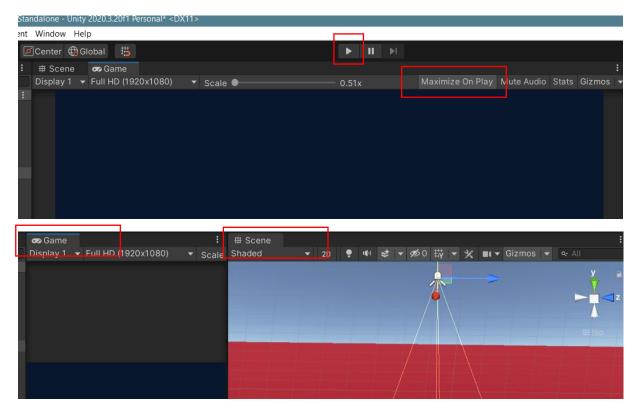
Understanding and learning Unity 3D will be the best possible by following the course, practicing the visual video materials, and applying them in person due to the visual character of this very comprehensive program.

## 2.2.Game Tab/Mod

In addition to the **#Scene** window where stage design is done in Unity, there is also a **Game** window to see what will be encountered when the game is started.

🚳 Proje1 - Samp	oleScene - PC, Mac 8	& Linux Star	ndalone - Unity	2020.3.20f1 Per	sonal* <dx1′< th=""></dx1′<>
File Edit Asset	ts GameObject Co	omponent	Window Hel	р	
₩ 🛟 💭	) 🛛 İ 🏵	×	]Center 🕀 G	lobal 🖽	_
'⊟ Hierarchy		a :	# Scene	😎 Game	
+ - Q. All			Shaded		🍷 🕪 📚
💿 🔊 Samı	pleScene*				
👰 Dir	<b>in Camera</b> rectional Light ane				

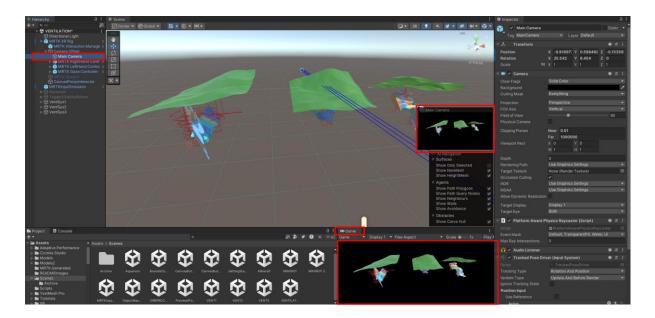
When the play button for this window is clicked, the display window belonging to the **Game** becomes active. If the **Play Maximized** option is active, the game scene will cover the entire screen. The game is closed by pressing the same button. The design page is activated by selecting the **#Scene** tab. It is also possible for both windows to be active at the same time. For this, the **Game** tab is fixed as a window in the desired position by dragging it. It can also be brought back to its previous position in the same way.



The **Game** mode feature will be used continuously in future scenes, and its function will be better understood.

### 2.3. Game and Scene Perspectives

To ensure that the perspective in the Scene and the angle in Game mode are the same, the camera must first be selected. Then, pressing **Ctrl+Shift+F** together will move the camera to the same transformation properties and perspective in both windows. The same applies to other camera types that will be added to the Scene.

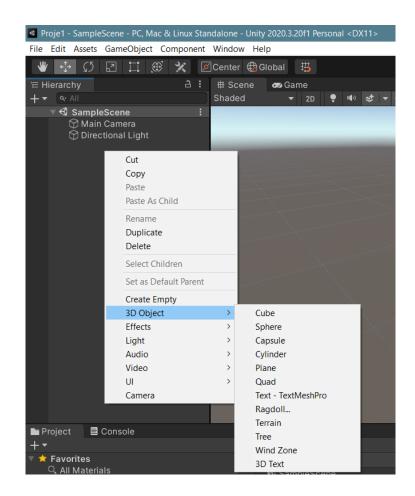


# **3. SCENE DESIGN**

# 3.1.Adding a GameObject

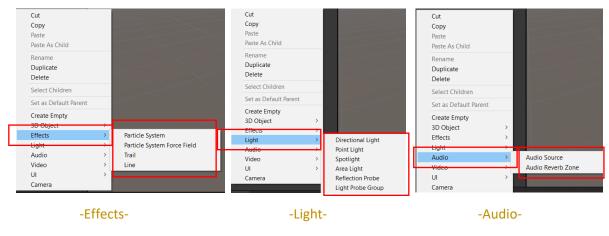
To add an object (**GameObject/Asset**) to the scene, internal or external sources can be used. First, in our project named Project1, let's create some game objects from Unity's standard libraries in the scene named **SampleScene**.

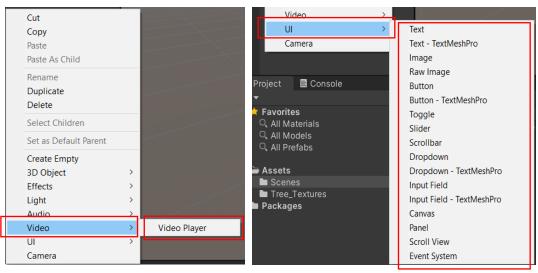
In the **Hierarchy** section, we can see the titles of the objects that we can add by right clicking our mouse and the objects under these titles. **Create Empty** is used to add an empty **GameObject** that will be used for different purposes. The most used object group is **3D Object** - a 3-dimensional object.



As can be seen, under the **3D Object** heading; Cube, Sphere, Capsule, Cylinder, Surface/Plane, Quad, **TextMeshPro, Rich Text Adding**, and **Ragdoll**, creating a character that is designed with a joint structure that we can move, Terrain, Terrain/Field Design, Tree Adding, Wind Zone, Wind Effect Adding, 3D Text – 3D Text Adding subheadings are included.

The menu and submenu components that we will see in detail while creating a project are in the figures below. These are, respectively, Effects, Light, Audio, Video, UI and Camera.



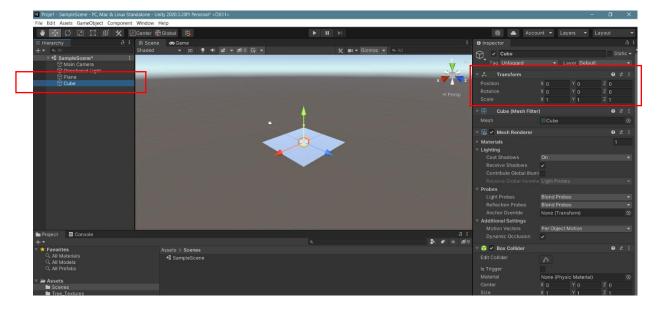


-Video-

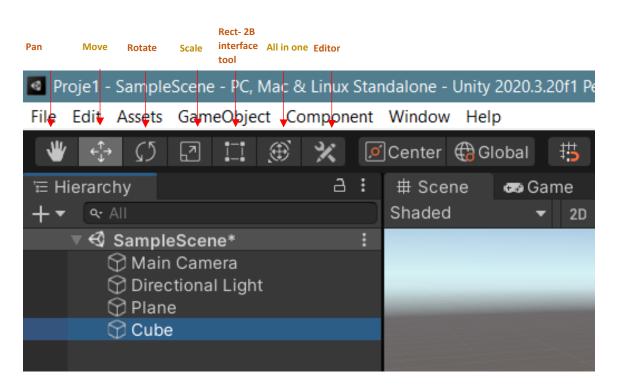


## **3.2.Object Properties**

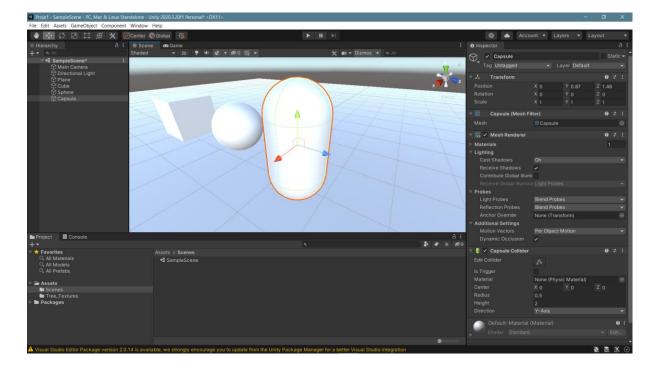
Let's add a few objects under **3D Object** to our scene. These will be a plane, a cube, a sphere and a capsule, respectively. But first, let's add a **Plane** and a **Cube** one after the other.



When the **Transform** window is examined, it will be seen that both Plane and Cube objects come with standard coordinates-**Position** of (0,0,0), angular rotation values-**Rotation** of (0,0,0) and scale settings-**Scale** of (1,1,1). To change these values, the numerical values in the window can be changed. **Transform** properties of objects can also be adjusted by selecting the shortcut keys located in the upper left corner of the project window.

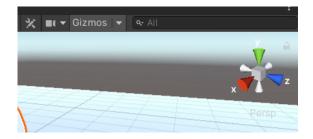


After selecting an object added to the list or an existing object with the mouse, if we hover over **#Scene** with the mouse and press the **F key**, it will be possible to focus and approach the object.



In practice, the rotation of the scene is done with the right click of the mouse, the zoom in and out with the rollers, and the **Pan** operation is done when the rollers are pressed.

Another issue is to clearly determine the positions of objects in the scene. For this, it is useful to use direction arrows called **Gizmo**. As can be seen in the figure below, x-z is used to define the horizontal and y-vertical directions.



By clicking on the green **y-key**, we can look at our scene vertically from the top and horizontally from the sides with the red **x-key** and blue **z-key**. As a standard, a 3D image is given by taking perspective into account. **Pers** (perspective) expression under the **Gizmo** shows this. By clicking on the **Pers**, the image will be converted to **Iso**, or isometric format.

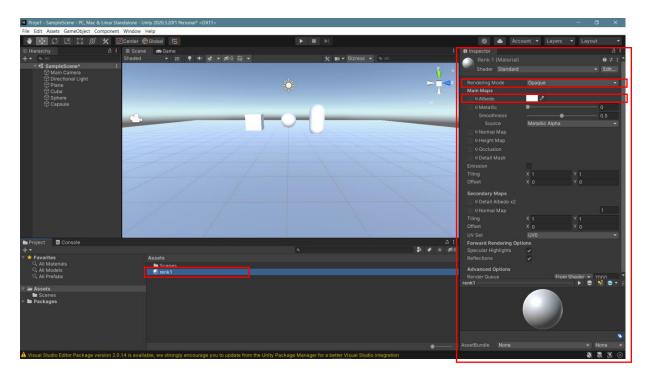
# 3.3. Materials

After creating objects, material is usually assigned to them. This can be a solid color, an image or a texture file. For this purpose, let's right click on the Assets window.

Proje1 - SampleScene - PC, Mac & L								- 0	×
File Edit Assets GameObject Com									
👋 🔄 🗘 🔄 🖽 😫 🖤	* 🖻	Center 🌐 G	lobal 📆			Ø	📥 Account 👻 Layers	▼ Layout	
						: O Inspector			
+ ▼		Shaded	▼ 20 9 10 25 ▼ 150 117 ▼	*	∎t ▼ Gizmos I ▼	Asse			
★ 4 Samplescene* G Main Camera G Directional Light G Cube G Sphere G Capsule	·	4		* 	Create Snow in Explore Open Delete Rename Copy Path	Alt+Ctrl+C	Folder C# Script 2D Shader Testing Playables Assembly Definition Assembly Definition Reference	> > > 2	Open
					Open Scene Addit	tive	TextMeshPro	>	
				View in Package N	Aanager	Scene			
				Import New Asset Import Package Export Package. Find References In Select Dependenc	> Scene	Scene Template Scene Template From Scene Prefab Prefab Variant Audio Mixer			
					Refresh	Ctri+R	Material		
					Reimport		Lens Flare		
					Reimport All		Render Texture		
					Extract From Prefa	b	Lightmap Parameters		
Project Console					Run API Updater		Lighting Settings Custom Render Texture		
					Update UXML Sch	ema	Animator Controller		
😫 Favorites					Open C# Project		Animation		
Q, All Materials Q, All Models Q, All Prefabs			Scenes		Properties	Alt+P	Animator Override Controller Avatar Mask	_	
🖉 🗃 Assets							Timeline Signal		
In Scenes In Packages							Physic Material		
						Asset Lat	GUI Skin Custom Font UI Toolkit	>	
							Legacy	>	
			Assets			AssetBunc	Brush	→ No	
Visual Studio Editor Package vers	sion 2.0.1	4 is available	e, we strongly encourage you to update from the U	nity Package Manager for a t	etter Visual Studio integration		Terrain Layer	* 3	

A window will open with a click. When you select **Create** here, a sub-window opens, and a comprehensive asset list appears. Since our goal is to create material, the **Material** option is clicked from the list. After giving a name to our new material/material, we can move on to editing information

about this asset in the Inspector. Our material is opaque and white by default. Let's check the white box in the **Main Maps** window to determine the new colon.



Let's specify the color we want from the **Color** window that opens.

<complex-block></complex-block>	Proje1 - SampleScene - PC, Mac & Linux Sta					- a ×
Network	File Edit Assets GameObject Component	t Window Help				
• • • • • • • • • • • • • • • • • • •	🕷 🖓 🖸 🖾 🕱 🗶 🕼	🖉 Center 🌐 Global 🛛 🟭			🖗 📥 Account 🝷	
Bangbackcrank   Concole   Sphre   <					Inspector	
Wink Cannes       United Cannes <td></td> <td>Shaded 🔹 20 🌻 🕪 🤩 🕶 🕫 0 📅</td> <td>🗧 👻 🖬 🕶 Gizmos</td> <td>si 🕶 🔍 🗣 All</td> <td>Renk 1 (Material)</td> <td>0 z : _</td>		Shaded 🔹 20 🌻 🕪 🤩 🕶 🕫 0 📅	🗧 👻 🖬 🕶 Gizmos	si 🕶 🔍 🗣 All	Renk 1 (Material)	0 z : _
Const      Const      Packages     Const      Cons	▼ 4 SampleScene* E			RGB 0-25+	Shader Standard Rendering Mode Main Maps Opar All Discolo Metallic Snoothness Snoure Neight Map Oheight Map Oberlail Map Oberlail Absdo x2 ONormal Map Tilling X 1 Offset X 0 Secondary Maps Oberlail Absdo x2 ONormal Map Tilling X 1 Offset X 0	V 1     V 1     V 0     V 1     V 0
A Fundación     Assets     B     0     Advanced Options       Q. All Models     Assets/renk1,mat     Assets/renk1,mat     Advanced Options       Packages     Assets/renk1,mat						
	<ul> <li>★ Favorites</li> <li>Q. All Materials</li> <li>Q. All Prefabs</li> <li>Q. All Prefabs</li> <li>Y ➡ Assets</li> <li>E Scenes</li> </ul>	🖿 🥥	B A Hex * S	o co co co co co co co co co c	Reflections  Advanced Options Render Queue	From Shader * 2000
		Annata/conk1 mat			AssetBundle None	▼ None ▼
	A Visual Studio Editor Package version 2.0		from the Unity Package Manager for a better Visual St			

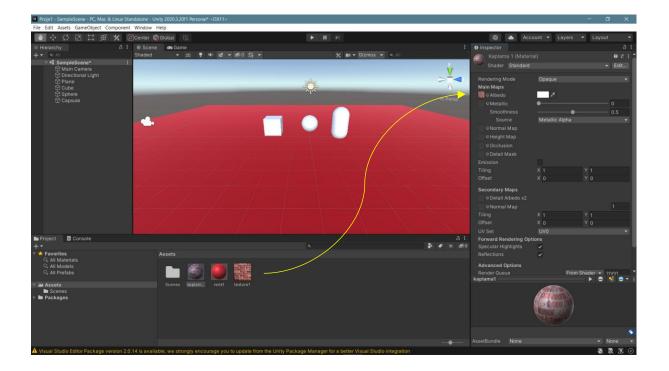
Now, it is possible to see the assigned color of the material that we created in the asset window. It should not be forgotten that there will be no change in the scene unless our objects in the **Assets** window are applied to the **Scene**. For this, we assign our color object in the scene by dragging it to the **Plane**, for example, or by dragging it to the **Plane** in the **Hierarchy**.

	tandalone - Unity 2020.3.20f1 Personal* <dx11></dx11>				- a x
File Edit Assets GameObject Component	it Window Help				
👻 🕂 Ø 🖾 🖽 🛠 I	🖉 Center 🌐 Global 🛛			🏟 🔺 Account 👻 Layers	✓ Layout ✓
			i 🛛 Insp		a i
	Shaded 🔹 20 🌻 🍽 🤩 🕶 💋 0 뀨 👻	🗙 🖬 🕶 Gizmos 💌 🕾 All			e z : <u>*</u>
					← Edit
한 Directional Light 다 Plane 다 Cube 다 Sphere 한 Capsule			Renc Arriso	g X 1 et X 0 ondary Maps Detail Albedo x2 DNormal Map	V 1 Y 0 Y 1 Y 0
			Offsi UV S		Yo
Project E Console				vard Rendering Options	
+-				cular Highlights 🗸	
All Materials All Materials All Models All Prefabs	Assets		Refie	ections 🗸 anced Options	Shader 👻 2000 *
Q All Pretabs			Rend renk1	der Queue From	Shader 🔻 2000
¥ ⊒a Assets Mar Scenes ▶ Mar Packages					•
= २ म 🗟 🛈	>> ⊴ 🖳 刘 💽 🖻 📦 🖬	á 🚍 🛷 🖪		∧ ū ≏ <b>□</b> @	4 <sup>3</sup> ) 16:56 (4 <sup>3</sup> ) 30.01.2022 € 30.01.2022 € 1)

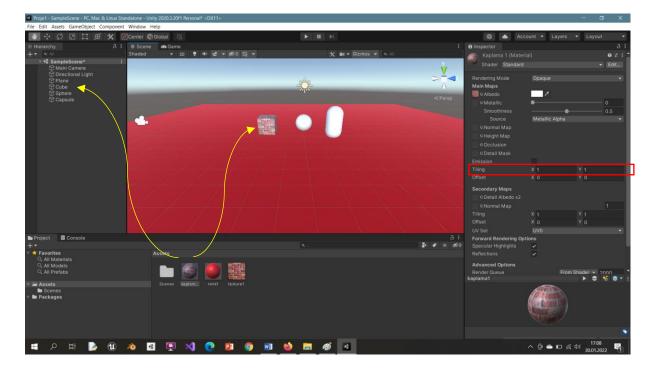
It is possible to do the same process on other objects and with different colors if desired. Another way of covering is to use a ready texture or image file. As can be seen, we added the **Texture1.JPG** file to the **Assets** folder by dragging it to the folder on Windows Explorer or directly to this window. To cover this material, we created a **Material** named **kaplama1**.

Proje1 - SampleScene - PC, Mac & Linux Star				- U X
File Edit Assets GameObject Component	Window Help			
	Center 🌐 Global 📅			Account 🕶 Layers 👻 Layout 🔍
	III Scene 🚥 Game		i Inspector	
+ - « All		🛱 👻 🖬 🕶 Gizmos	💌 🔍 All 🦳 Kaplama 1 (Ma	iterial) 🛛 🛛 🕯 🐮 📩
♥ I SampleScene* : I Main Camera I Directional Light			Shader Stand	ard - Edit Opaque -
<ul> <li>♀ Plane</li> <li>♀ Cube</li> <li>♀ Sphere</li> <li>♀ Capsule</li> </ul>	_	÷.	A Main Maps ≪ Persp. ○Albedo	×
(T) Capsule			• Metallic	
	Contraction of the second second		Smoothness	0.5 Metallic Alpha
			©Normal Map	metanic Apria
	and the second second		⊂ Height Map	
	and the second second second		Occlusion	
	and the start of the		O Detail Mask	
	1 1 1 1 1 1		Emission	
			Tiling Offset	X 1 Y 1 X 0 Y 0
			Secondary Maps • Detail Albedo >	
			O Normal Map	
			Tiling	
Project E Console			UV Set	UV0 -
+ - Favorites			ට : Forward Rendering ა අ ≠ ණ9 Specular Highlights	
			Reflections	
Q, All Materials Q, All Models			Advanced Options	
Q All Prefabs			Render Queue	From Shader 👻 2000
🔻 📾 Assets	Scenes kaplam., renk1 texture		kaplama1	▶ ● 🤞 ● - :
In Scenes ▶ In Packages				
P 🖬 Packages				
			AssetBundle None	✓ None ✓
A Visual Studio Editor Package version 2.0.	14 is available, we strongly encourage you to update	e from the Unity Package Manager for a better Visual Stu	dio integration	o 🔊 💈 🕷

Now, let's drag our **Texture1.JPG** file to the box named **Albedo** in the **Main Maps** section in the Inspector window of the **kaplama1** material.



It is now possible to connect our material covered with **Texture1.JPG** to an object in the scene. To do this, we simply drag our kaplama1 material onto our cube or the **Cube** on the **Hierarchy**.



The coating frequency can be changed by determining how many times the selected texture will be repeated in the **x** and **y** directions on the **Tiling**.

# 3.4.Lights

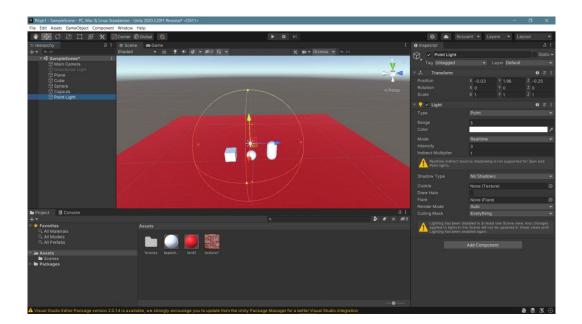
The subject of **light** is a very comprehensive title in game engines and has a direct impact on quality. Here, the addition of light types will be examined first. **Standard** light types are in the Light submenu. Here, **Directional Light** was already the current light type. It will be possible to see other light types more clearly by making the box for this inactive position in the **Inspector** or by deleting it.

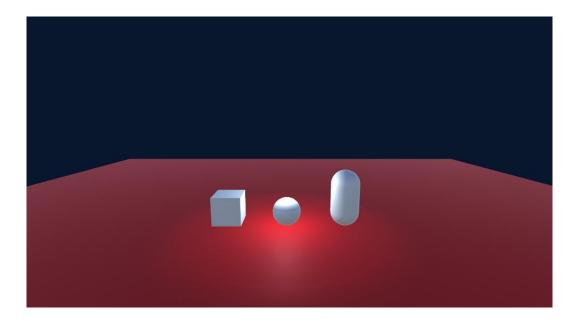
Let's try **Point Light**, **Spotlight**, **Area Light** types in that order. **Probe light** types can be examined at a later stage.

In this section, the information is given in a limited way due to its visual character and many light settings are demonstrated in practice during the lesson.

#### 3.4.1 Point Light

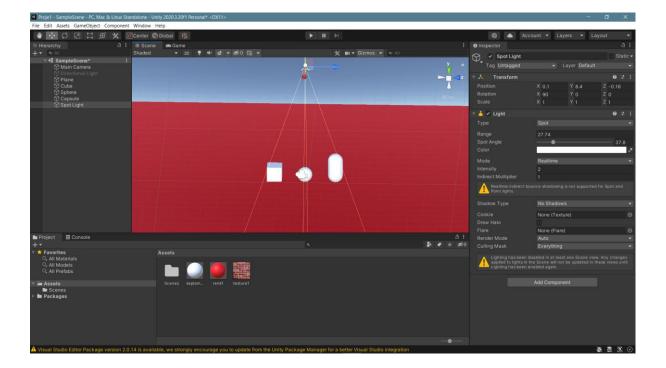
When **Point Light** is used, point light is applied to objects in a certain position, light color and intensity, and the following instant images are obtained. The images are shown first in the scene and then in **Game Mode**.

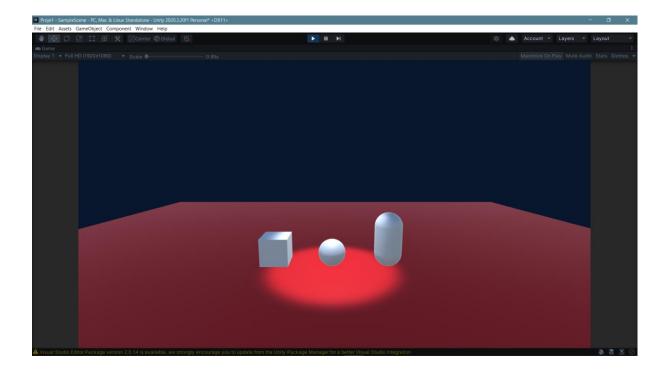




# 3.4.2 Spotlight

In this type of light, light application is achieved in a way that creates a cone.

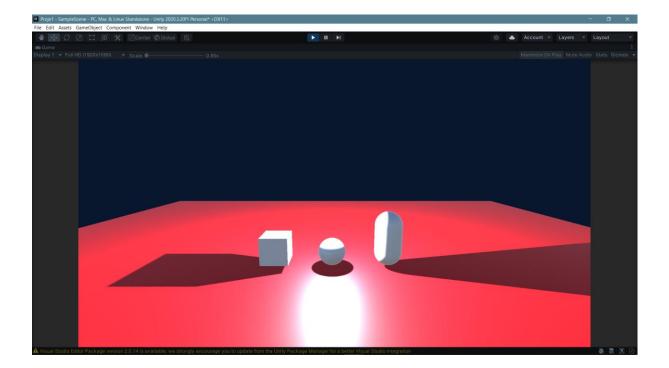




### 3.4.3 Area Light

This type of light serves the purpose of illuminating a certain area. **Area Light** has **Spot**, **Directional**, **Point**, and **Baked** options as **Type** and produces different results.

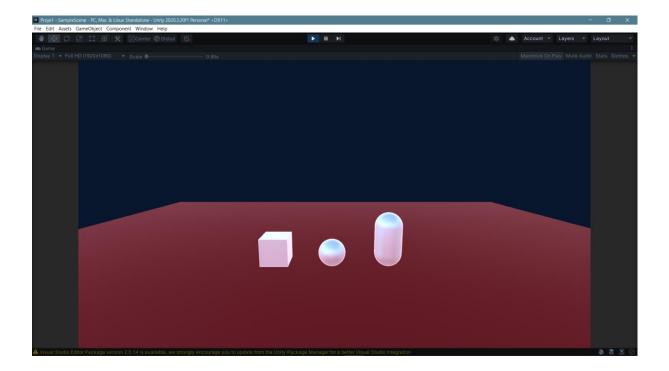
Proje1 - SampleScene - PC, Mac & Lin	nux Standalone - Unity 2020.3.	20f1 Personal* <dx11></dx11>					a ×
File Edit Assets GameObject Comp	onent Window Help						
👋 🖅 🗘 🖾 🎞 🛎 🌶	🕻 🖉 Center 🌐 Global				🔍 📤 Aci	count 🕶 Layers 💌 La	
TE Hierarchy a I	œ Game ⊞ Scene				O Inspector		
+ × ( 🗢 All	Shaded 🔫 2D	📍 🕪 🔩 🖛 🕫 0 🛱 👻	💥 🖬 🕶 Gizmos 👻	l≪ All	😭 🗸 Area Light		
♥ 🕄 SampleScene* i 分 Main Camera					Tag Untagged	- Layer Default	
C Directional Light							
💬 Plane				2 <b>&gt; 1</b> <	Transform		
Cube				Δ	Position Rotation	X -0.06 Y 1.32 X 90 Y 0	Z -0.21 Z 0
Capsule					Scale	X 90 Y 0 X 1 Y 1	
🛠 Area Light			-				
					🔨 🍳 🖌 Light		0 1 1
		and the second se				55	
							1
	the state of the s						
			<u>`</u>				
						None (Texture)	
						None (Flare)	
						Auto	
						Everything	
Project 🛛 E Console					Lighting has been disa	ibled in at least one Scene view. / Scene will not be updated in the ibled again.	ny changes
+ • • ★ Favorites				🏷 🛷 ★ 💋9	Lighting has been ena		
Favorites Q All Materials							
Q, All Models	<b>—</b>						
Q, All Prefabs							
🔻 🖨 Assets		kaplam renk1 texture1					
Scenes							
🕨 🖿 Packages							
				-•			
A Visual Studio Editor Package version	on 2.0.14 is available, we st	rongly encourage you to update from the U	nity Package Manager for a better Visual Stud	io integration			0228



### 3.4.4 Reflection Probe

In this type of light, illumination and reflections are calculated via a probe.

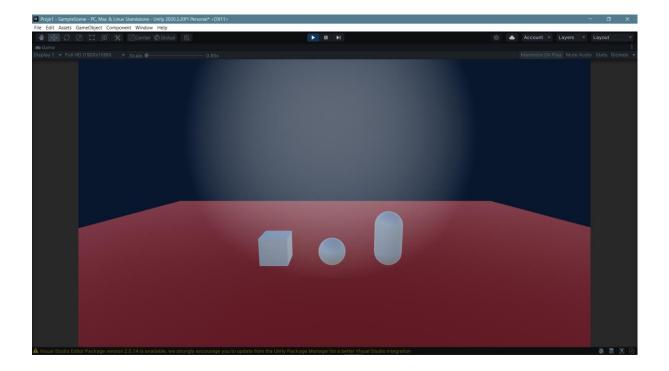
Proje1 - SampleScene - PC, Mac & Lin	nux Standalone	- Unity 2020.3.	20f1 Persona	i* <dx11></dx11>												
File Edit Assets GameObject Comp	oonent Windo	w Help														
🖤 🔂 C) 🖾 🖽 🎐	Cente	r 🜐 Global						II ÞI				(Q) 🔺 /				
'⊟ Hierarchy a I	or Game											Inspector				
+ • • • All			. 140 -	\$ • 960	<b>₩</b> •			🛠 🖬 🗸 Gizi				Reflection Prol				
✓ I SampleScene* I											y a	Tag Untagged		Layer Defau		¥.
Directional Light     S Plane											Z	🐨 🦂 🛛 Transform				
Cube													X 0.07	Y 1.63		
Sphere     Capsule																
Capsule Reflection Probe																
												🔻 🤤 🖌 Reflection Probe				
						A										e I
													On Awal			-
														al faces		
												Importance				
							_ <u>_</u>									
							1									
												T Cubemap Capture Set				
Project 🛛 Console											a :	Resolution HDR	128			
+*										* *		Shadow Distance				
🔻 📩 Favorites												Clear Flags	Skybox			
Q. All Materials Q. All Models					and the second s							Background				1.
Q, All Prefabs												Reflection Pi 🔤 🖂 —		0		
🔻 🖨 Assets		Scenes	kaplam	renk1	texture1											
E Scenes																
🕨 🖿 Packages																
													_	_		
											•					
A Visual Studio Editor Package version	on 2.0.14 is a	vailable, we st	rongly enco	ourage you t	o update from	m the Unity Pack	age Manag	er for a better Vi	sual Studio integration						* 3 3	0



# 3.4.5 Light Probe Group

Grouped probes can be used in stage lighting. Here, Halo from Effects is also added with Add Component.

Proje1 - SampleScene - PC, Mac & Linux Standa							
File Edit Assets GameObject Component W	/indow Help						
🖤 💀 Ø 🛛 🞞 😸 🗶 🔳o	enter 🌐 Global 🛛 🔢	► I			🖗 📥 Acc	count 🕶 Layers 💌 L	
12 Hierarchy A I 😎 Gam					0 Inspector		
+ • • All Shaded	▼ 2D 📍 4N 💐 ▼ 950 蒜		💥 💵 👻 Gizmos 👻 💁 All		D Light Probe Grou	p	
I SampleScene* i		<u>_</u>		<u>v</u> =	Tag Untagged	<ul> <li>Layer Default</li> </ul>	
💮 Directional Light				► Z	Transform		0 2 1
😚 Plane 💮 Cube						X 0.02 Y 4.78	Z -0.23
💮 Sphere				≡ Right		X O Y O	Z 0
Capsule     Getain Capsule     Getain Capsule				a kigni			
Gr sight in the second			8/->		👻 📴 🖌 Light Probe Group		071
						Edit Light Probes	
						Z Eur Light Probes	
		<u> </u>					
					🔻 🖶 🖌 Halo		0 : :
							1
	and the second second second second second second second second second second second second second second second					5	
						Add Component	
Project 🖪 Console				a 1			
+ -				□: ► ● ± ∞9			
+ <del>•</del> ▼ ★ Favorites							
Q, All Materials Q, All Models							
Q, All Prefabs							
🔻 🗃 Assets	Scenes kaplam renk1 te	xture1					
🕨 🖿 Scenes							
Packages							
A Visual Studio Editor Package version 2.0.14	is available, we strength appointing you to u	edata from the Linity Deckage Manage	e for a bottor Visual Chudio Integration				* * * * •



Lighting in Unity has very rich content and detail. For example, while creating a project, scenario-based lighting can be designed by using special lighting mode templates such as Universal Render Pipeline (URP), using Post Processing components, or by adding many special scene lighting features via Package Manager within the project. In future topics, these issues can be included according to the lighting needs of the project.

## 3.5.Rigidbody

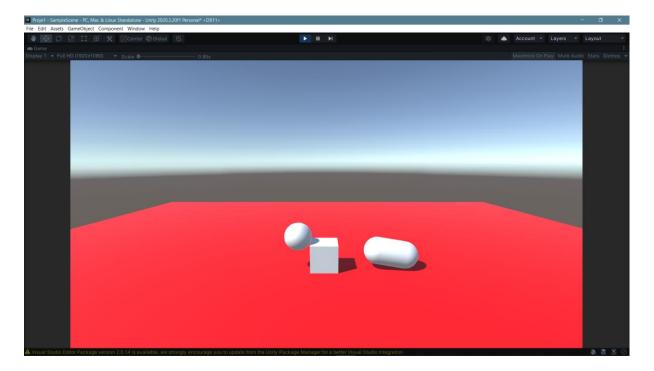
By giving physical properties to objects, they can be made to move in game mode. When an object has a volumetric mass, it is expected to be affected by gravity and falls towards the ground accordingly or rises upwards in the opposite case. **Rigidbody** is added to the object's properties by clicking **Add Component** in the **Inspector** section of the object.

In the figure, the objects in the project are placed on top of each other. This image does not change when **Game Mode** is on.

Proje1 - SampleScene - PC, Mac & Lin												a x
File Edit Assets GameObject Comp	onent Window H	lelp										
👋 🚯 🗘 🗵 🖽 🛠	🕻 🖉 Center 🌐	Global							🖗 📥 Acc	count 👻 l	.ayers 🔻 La	ayout 👻
									Inspector			a I
						🛠 🖬 🗸 Gizmos			Main Camera			Static 🛩
					siles.			y a	Tag MainCamera		Layer Default	
() Main Camera () Directional Light () Plane () Cube () Sphere								V			Layer Deraun	
									Transform			071
	(							<u>A</u>		X 8.78	Y 3.94	Z -0.09
Capsule								Sec.				Z 0
												Z 1
		-			and the second se	_	_	🔻 💷 🖌 Camera			97:	
							Clear Flags	Skybox		*		
						Background			1			
								Culling Mask Projection	Everything		*	
						Perspective				-		
								Vertical		-		
			* Main Camera			era			•	60		
									Use Graphi			
							a :			der Texture)	0	
Project 🗟 Console									Occlusion Culling HDR	V Usa Granhi	ics Settings	
+ - Favorites		Assets					. (5.77		MSAA		ics Settings	
All Materials									Allow Dynamic Resolution			
다. All Models 다. All Prefabs		And a second sec										
				THE REAL						Display 1		
▼									🔒 🖌 Audio Listener			● ‡ :
Scenes Packages										Add Compon		
												1
A Visual Studio Editor Package version	in 2.0.14 is availab	ble, we stro	ingly encourage you	to update from the	he Unity Package Ma	inager for a better Visual	Studio Integration					0 2 2 %

However, when a solid physical body property is added to the cube, sphere and capsule with the **Rigidbody**, it will be seen that the objects move with the acceleration of gravity.

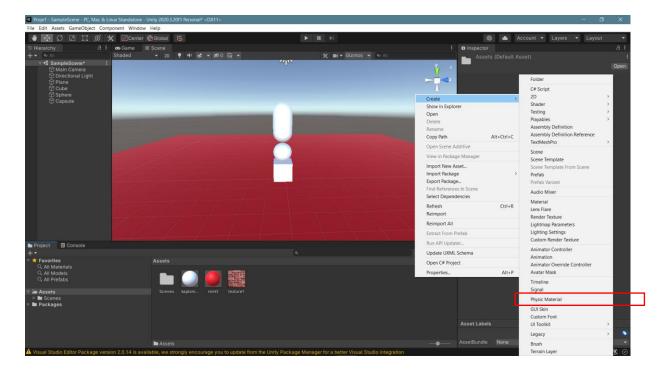
Projet - samplescene - PC, Mac & Li	nux Standaione - Unity 2020.3.2011 Persi	onal- <dxtt></dxtt>						- U A
File Edit Assets GameObject Comp	ponent Window Help							
👋 🔄 Q 🔄 🗎 🛎 ≯	🗶 🖉 Center 🌐 Global 🛛 👪					🖗 📥 Acc		
TE Hierarchy a 1	Game					Inspector		
+ • 🔍 🔍 All	Shaded 👻 2D 🌻 📢	\$ * 160 蒜 *		t ♥ Gizmos   ♥   ٩- Al		Ø ⊻ -		
♥ 🕄 SampleScene* : ⑦ Main Camera			silve			Tag Untagged	✓ Layer De	
💬 Main Camera					V			
🕄 Plane						Transform		
Cube					Δ	Position		
Capsule						Rotation		
						Scale		
		_			and the owner where the owner where the owner where the owner where the owner where the owner where the owner w	🔻 🏢 🛛 Capsule (Mesh Fill		
						Mesh		
						🔻 🗒 🖌 Mesh Renderer		021
						► Materials		
						⊤ Lighting		
						Cast Shadows		
							~	
						T Probes		
						Light Probes	Blend Probes Blend Probes	-
						Reflection Probes Anchor Override	None (Transform)	• ⊙
						<ul> <li>Additional Settings</li> </ul>	None (mansionin)	
						Motion Vect		
					a :	Dynamic Oc		
Project Console					≓ + ≠ ¢≶9	Components that	Search	_ cannot be
+ - Favorites	Assets				•• • × 400			
Q. All Materials						Rigidbo Event Ti		
다. All Models 다. All Prefabs						Priction		
C All Pleiaus						3 Grid		
🔻 🖨 Assets						II Grid Lay	out Group	
E Scenes Packages						Horizon	tal Layout Group	
rockuges						🞽 Spring J		
						🔀 Spring J		
						💭 Sprite M		
						Sprite R		
A Visual Studio Editor Package versi	on 2.0.14 is available, we strongly ei	ncourage you to update from the	e Unity Package Manager for a b	etter Visual Studio integration		# TMP_Sp		



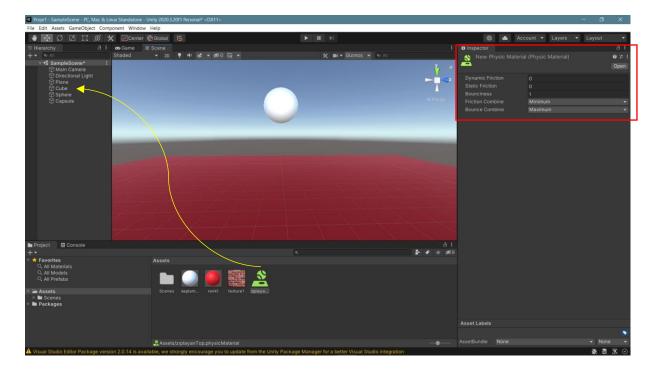
Here the capsule falls next to the cube, and the sphere also falls and continues to roll towards the end of the floor.

# 3.6.Physic Material

To move the physics properties to a higher level, it is necessary to define a special material type, **Physic Material**, and assign/bind it to the object. To do this, open the menu with the right mouse button in the **Assets** window. Select **Physic Material** from **Create**.



After naming the **Physic Material** as **ZiplayanTop**, let's edit its properties in the **Inspector**.



If we want the sphere to **bounce** continuously, we must change the **friction** settings of the physics material and drag this material to the **Sphere**, that is, to our sphere. In **game** mode, we can see that our sphere (ball) is constantly **bouncing**.

# 3.7.Particle System

One of the most functional objects is the **Particle System** – particle/particle system. In order to better understand the subject, it is possible to proceed with an example study in the form of fire burning. For this purpose, a few cylinders were laid on their sides and painted with brown material to make wood appear.

Right-click on our mouse in Hierarchy and select **Effect -> Particle System**. This object is designed to automatically scatter particles around.

File Edit Assets GameObject Component Window	w Help				9 .
🖤 🕂 Ø 🛛 🎞 😂 🗶 🗹 Pivot	GLocal II			🖗 📥 Acc	ount • Layers • Layout •
🕾 Hierarchy 👌 i 🛷 Game	III Scene			0 Inspector	
+ • • All Shaded	+ 20 • • st + 950 ∰ +	💥 🖬 🕶 Gizm		Particle System	
⊤ SampleScene* I				Tag Untagged	Layer Default
Main Camera  Directional Light			<u> </u>		
Plane			► 🙀 🔫 Z	Transform	
Odunlar     Particle System					
C Parocie System					
				🔻 🐮 🛛 Particle System	
					Open Editor
				Particle System	7
				Duration 5	
				Looping - Prewarm	
			▼ Particle Effect	Start Delay 0	•
			Pause Restart Stop	Start Lifetime 5 Start Speed 5	
			Playback Speed 1.00	30 Start Size	
			Playback Time 14.45	Start Size 1 3D Start Rotation	
			Particles 50 Speed Range 5.0 - 5.0	Start Rotation 0	
			Simulate Layers Nothing •	Flip Rotation 0 Start Color	
			✓ Resimulate	Gravity Modifier 0	
			Show Bounds	Simulation Space	ocal 👻
			Show Only Selected		caled 👻
Project El Console			ai		ocal 👻
+*			► + ± ± ± ± ± ± ± ± ± ± ± ± ± ± ± ± ± ±	Emitter Velocity B	igidbody 👻
👻 🚖 Favorites					
Q, All Materials Q, All Models		antas an		Stop Action	lone 👻 🗸
Q All Prefabs				Particle System Curves	
					Optimize Remove
Assets					
> Scenes					
🕨 🖿 Packages					
A Visual Studio Editor Package version 2.0.14 is av	ailable, we strongly encourage you to update from	the Unity Package Manager for a better Vis	ual Studio integration		0 2 2 2 3

There are many settings in the Inspector for **Particle System**. Let's examine the most important ones here. It is possible to go into more detail and apply them during the course.

	🖗 🔺 Account 👻 Layers 👻 Laye	but
	1 Inspector	а
imos 👻 🔍 All	♀ Particle System	Static -
<u> </u>	Tag Untagged   Layer Default	-
	Transform	07:
x		0.13
< Persp		0
		1
	🔻 🍟 🛛 Particle System	0 I i
	Ope	n Editor
	Particle System	
Particle Effect     Pause Restart Stop     Playback Speed 1.00     Playback Time 232.20     Particles 50     Speed Range 3.1 - 3.1     Simulate Layers Nothing      ✓ Resimulate     Show Bounds     Show Only Selected     Show Only Selected	Duration 5 Looping 5 Start Delay 0 Start Delay 0 Start Delay 0 Start Delay 2 Start Speed 3.07 3D Start Size 7 X 1 Y 1 Z 1 3D Start Rotation 7 X 0 Y 0 Z 0 Flip Rotation 0 Start Color 7 Gravity Modifier 0 Start Color 7 Gravity Modifier 0 Start Color 1 Gravity Modifier 0 Start Color 1 Simulation Space 1 Deta Time Scaled 1 Deta Time Scaled 1 Deta Time 5 Scaling Mode Local 1 Play On Awake* 7 Emitter Velocity Rigidbody 1 Max Particles 1000 0 Auto Random Seed 7 Stop Action None 1 Common 1 Start Color 1 Start Color 1 Start Color 1 Scaling Mode 1 Play On Awake* 7 Emitter Velocity Rigidbody 1 Stop Action 1 Start Color 1 Stop Action 1 Start Color 1 Start Colo	
	Particle System Curves	
	Optimize	Remove

Duration determines the time the particle will appear. Looping ensures that particle production continues in a cycle structure. Start-Lifetime regulates the particle life, Start Speed regulates the particle speed. Start Color determines the initial color of the particle. This color can be changed to a light red or orange color for a flame image. Gravity Modifier regulates gravity. A value of zero ensures that the particle rises as standard. Negative values ensure faster rise. However, positive values mean that the particles return to the ground. In some applications, if the particles are desired to fall, this value should be made positive.

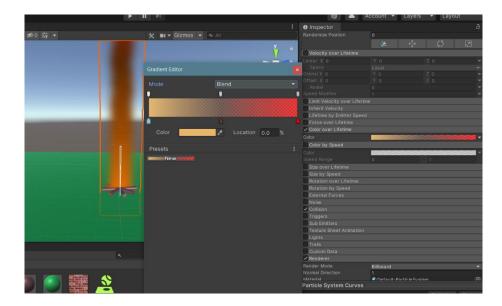
:	€ Inspector Start Color Gravity Modifier Simulation Space Simulation Space Delta Time Scaling Mode Play On Awake* Emitter Velocity Max Particles Auto Random Seed Stop Action Culling Mode	0 Local 1 Scaled Local V Rigidbody 1000 V None Automatic	
	Ring Buffer Mode Emission Rate over Time Rate over Distance Bursts	Disabled 10 0	
irt Stop	Time Count	Cycles Interval Probability 1   0.010  1.00  + -	
68.11 ⊃ 0 - 2.0 lothing ▼	✓ Shape Shape Angle Radius Radius Thickness Arc Mode	Cone 25 1 360 Random	
ad 	Spread Length Emit from: Texture Position Rotation Scale	0         5           Base         1000000000000000000000000000000000000	
	Particle System Curves =		

When we scroll down in the **Inspector**, we will see the **Emission** settings. Rate over **Time** determines the number of particles to be produced per second. In the **Bursts** section, it is possible to obtain an explosion effect by determining which number of particles will be released at which time with the **+** button.

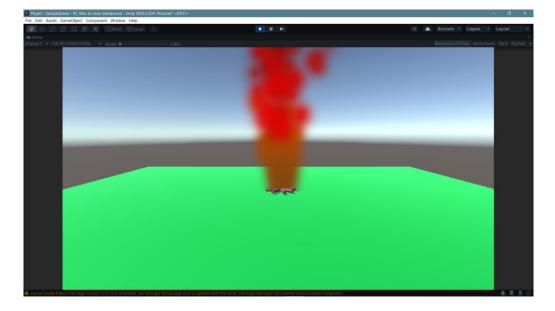
**Shape** determines the volumetric shape in which the particles will spread.

**Color over Lifetime, Renderer** and dozens of settings and their sub-settings can be changed to observe the results. Such changes can be made until the desired image is obtained.

In the **Color over Lifetime** setting, the color change and transparency level can be determined from the beginning to the end moment.



When specifying color, if **Gradient** - gradual change is selected, there are two settings in the window that opens: **Color and Mode**. The first and last color of the particles are determined by **Color**, and the **Alpha** percentage in Mode determines the transparency levels. If desired, the intermediate color and transparency level can also be assigned.



A flame image should be used to capture a more realistic image in the flame image. A flame image can be downloaded to the **Assets** section by searching for flame texture on **Freepik** and similar sites. Some presets are required to use this image.

First, a flame image is placed in the **Albedo** of a **Material** in the **Assets** section. However, this image is set to **Shader -> Standard**. This needs to be made compatible with the **Particle System**.

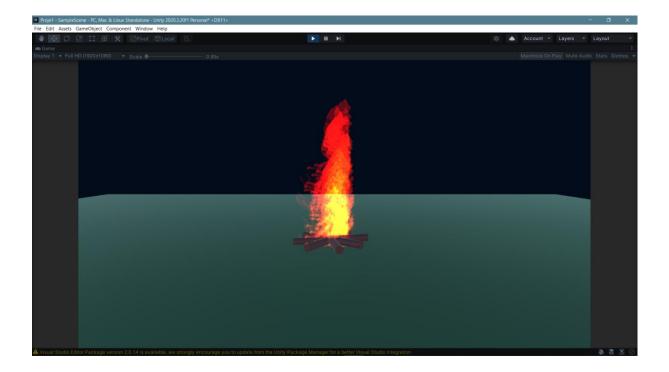
**Shader > Legacy Shaders>Particles>Additive** selections should be made for this. The result is more realistic.

Inspector		а	Inspector			a :
Alev (Material)		0 I :	Alev (Ma	terial)		0:::
Shader Standard		▼ Edit	Shader S	Standard		Edit
				٩		
Rendering Mode	Opaque	<b>•</b>	Rendering Me	Shaders		
Main Maps			Main Maps	Mobile		
🕹 🛛 Albedo	68		🏅 © Albedo	Nature		
© Metallic	•	— o	<ul> <li>Metallic</li> </ul>	UI		
Smoothness	•	- 0.5	Smoothi	VR		0.5
Source	Metallic Alpha	-	Sourc	FX		
Normal Map			© Normal	GUI		
© Height Map			○ Height N	Particles Skybox		
© Occlusion			© Occlusio	Sprites		
© Detail Mask			© Detail M	Unlit		
Emission	~		Emission	Autodesk Interactive		
©Color	HDR d		©Color	Standard		
Global Illumination	None		Global II	Standard (Specular setup)		
Tiling	X 1 Y 1		Tiling	Legacy Shaders		
Offset	X 0 Y 0		Offset	X 0	Y 0	
Oliset			Unset	^ 0		
Secondary Maps			Secondary Ma	aps		
© Detail Albedo x2			© Detail Alb	pedo x2		
Normal Map		1		lap		
Tiling	X 1 Y 1		Tiling	X 1	Y 1	
Offset	X 0 Y 0		Offset	X 0	Y O	
UV Set	UV0	-	UV Set	UV0		
Forward Rendering Optio	ns		Forward Rend	lering Options		
Specular Highlights	~		Specular High	lights 🗸		
Reflections			Reflections	~		
alev	▶ ●	🈼 🤤 🔻	alev		▶ ●	5 🤤 🗸

Inspector		a :	Inspector		a :
Alev (Mat	erial)	07:	Alev (Material)		0711
Shader St	tandard	▼ Edit	Shader Standard		F Edit
Rendering M	۹.				
Main Maps	Legacy Shaders		Rendering Mc	Particles	
Albedo	Bumped Diffuse		Additive-M	ultiply	
	Bumped Specular Decal	0	Additive		0
	Diffuse	0.5	Smooth Alpha Blende		0.5
	Diffuse Detail			d Premultiply	0.5
O Normal     I	Diffuse Fast		© Normal Anim Alpha E		
	Lightmapped Parallax Diffuse		Multiply		
	Parallax Specular		Occlusic VertexLit Ble		
	Particles	>	© Detail M	lided	
Emission	Reflective		Emission		
© Color	Self-Illumin Specular		©Color		
Global II	Transparent		Global II		
Tiling	VertexLit		Tiling		
Offset	X 0	YO	-	0 Y 0	
Secondary Ma	os		Secondary Maps		
© Detail Albe			© Detail Albedo x2		
Normal Ma	qe	1	© Normal Map		
Tiling	X 1	Y 1		1 Y 1	
Offset	x o	Y O	9	0 Y 0	
UV Set	UV0			JVO	
Forward Rende	ering Options		Forward Rendering Options		
Specular Highli	ghts 🗸			/	
Reflections	~		Reflections	·	
alev			alev	▶ ●	😼 🤤 🔻 🗄
Proje1 - SampleScene - PC, Ma File Edit Assets GameObject	c & Linux Standalone - Unity 2020.3.20f1 Personal* <dx Component Window Help</dx 	11>			- 0 >
	🖹 🗶 🖉 Pivot 🎯 Local 👸	Þ	ны	🖗 📥 Account 👻 Layers	<ul> <li>Layout</li> </ul>
	ali ∰ Scene ∞ Game Shaded → 2D ♥ ₦ & ▼	¢60 ∰ +	💥 🖩 🕶 Gizmos I 🕶 👁 All	Inspector     V Particle System	
♥ I SampleScene* I Main Camera I Directional Light	1		ý -	Tag Untagged → Layer De	
⊕ Plane ▷ © Oduniar			~ <b>~</b> ~		0 ≠ 37 Z 0.13
				Rotation         X         -90         Y         0           Scale         X         1         Y         1	
	_			▼ ¥ Particle System	
				Particle System	Open Editor
				Duration 3	Q
			V Particle Effect	Looping Prewarm Start Delay 0	
			Pause Restart Stop Playback Speed 1.00	Start Lifetime 3 Start Speed 1	
		2×	Playback Time 222.61	3D Start Size X 1 Y 1 Z 1 3D Start Rotation ✓	
			Particles 25 Speed Range 1.0 - 1.0	X 0 Y 0 Z 0 Flip Rotation 0	
			Simulate Layers Nothing    Resimulate	Start Color Cravity Modifier 0 Simulation Space Local	
			Show Bounds Show Only Selected	Simulation Speed 0.5 Delta Time Scaled	
Project 🖾 Console		à	.∃: ► # ★ #5		-
+ - Favorites All Materials			★ ★ 201	Max Particles 1000 Auto Random Seed 2	
Q, All Materials Q, All Models Q, All Prefabs		i 🛛 🦱 🍋 🗱	*	Stop Action None Particle System Curves	
🔻 🗃 Assets	Materia Scenes ale	ev glowing kaplam renkt texture	1 zplaya		Optimize Remov

Or, from **Main Camera -> Clear Flags -> Solid Color** option, a solid color can be selected instead of the **Skybox** in the background. This will give us a different image.

In Scenes
 In Packages



With the **Particle System**, many different effects can be developed that produce particles downward, upward. **Lateral** directions depending on the scenario and need.

Another example is the scene of a truck that crashed into a mine, and its engine caught fire.



# 4. C# SCRIPTS FOR UNITY

This section contains introductory information for those interested in coding. Unity is a real-time development engine based on the **C#** programming language. This language is used in coding (script) in the developed computer, mobile device, VR/AR applications. This section covers the most basic C# coding that may be required within the course scope.

Let us state that our aim will be more about Unity-specific structures and the developed command library over the years rather than teaching the C# language. In addition, even if it is ready, code files will be used within the scope of the course. Therefore, Unity C# basics are briefly discussed to give an idea. Other coding information within Unity applications will be briefly emphasized.

## 4.1.Basic C# Information

After the scene preparations, doing C# coding to develop various events will be necessary. If the codes are **not connected** to the scene and the **objects** that make up the scene, the game will **not work**. For this, both coding and connections must be known.

C# gives console and visual outputs. There is a window for **Console** outputs (DOS screen) in Unity. Visual outputs are already observed movements and actions in the game. With coding, we can access every object and its sub-areas and settings on the Inspector screen.

Visual Studio 2022 had already been installed in Unity for C# coding. However, if Visual Studio was not added while installing Unity, open the **Unity Hub** and go to the **Installs** section.

Unity H	lub 3.8.0								- 0	×
KE		٥	Proj	ects	;		Ad	1 - bt	New projec	t
٩	Projects						Q Se			
8	Installs				NAME	CLOUD		EDITOR VERSIO	N	
): (	Learn Community				<b>MinPro1</b> D:\UNITY\MinPro1	CONNECTED	a few seconds ago	2022.3.14f1		
					MineModels D:\UNITY\MineModels		an hour ago	2022.3.14f1		
					UGM1 D:\UNITY\UGM1	CONNECTED	7 days ago	2022.3.14f1		
					CSUnity1 D:\UNITY\HOLOLENS\CS		19 days ago	2022.3.14f1		
					FPSApp1 D:\UNITY\FPSApp1	CONNECTED	19 days ago	2022.3.14f1		
+	Downloads				AdobeMixamo D:\UNITY\AdobeMixamo	CONNECTED	21 days ago	2022.3.14f1		
	Bomiloada			22	M_Mine1		a manth ara	2022 24464	~	

Select the **settings** button in the upper right corner of the Unity 2022.3 version window. From the window that opens, select the **Add Modules** tab.

<b>B</b>	Unity (2023.2.2f1) D:\Program Files\2023.2.2f1\Editor\Unity.exe Android UWP WebGL Windows	Add modules Show in Explorer	\$
8	Unity (2022.3.14f1) LTS D:\Program Files\2022.3.14f1\Editor\Unity.exe Android UWP WebGL Windows	Uninstall	¢

Here, Visual Studio will automatically be selected if it has yet to be chosen. After that, the installation process continues. Also, **Add Modules** is used when there is a module that you want to add that has not been installed before, such as the IOS module.

To access the code files from Unity, double-click on the script file is enough. However, a definition is required to do this if Visual Studio was added later. The connection is established via **Edit>Preferences>External Tools**.

Preferences				: 🗆 ×
General V2D Physics Analysis Profiler Asset Pipeline Colors Diagnostics External Tools Ci Cache Jobs Package Manager Scene View Scene Template V Search Indexing Timeline	External Tools			
	External Script Editor	Visual Studio 2022 [17.9.34607]		•
	Generate .csproj files for: Embedded packages Local packages Registry packages Git packages Built-in packages Local tarball Packages from unknown sources Player projects Regenerate project files	<b>*</b>		).22 enabled
UI Scaling Visual Scripting	Image application	Open by file extension		-
	Revision Control Diff/Merge	Custom Tool		-
	Tool Path			Browse
	Two-way diff command line			
	Three-way diff command line			
	Merge arguments			
		VCS diff tools were found. Please set up a custom to	ol or install one of the following tools:	

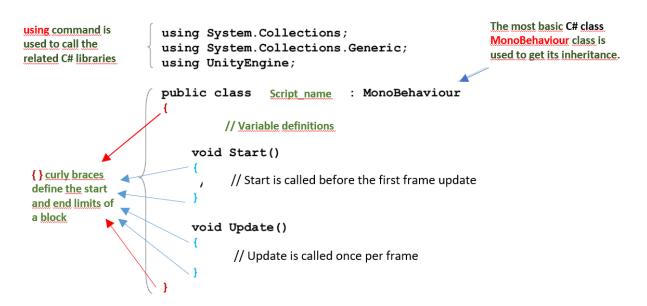
The C# program opens with a certain **template** and is ready for the codes we will add. The script file's name is also the class structure's name. **Class** is the most basic building block of **object-oriented programming** and is an object that contains variables, methods and functions and is designed to be used and accessed repeatedly. Adding **MonoBehaviour** at the entrance inherits the most basic C# class structure found in the **C# library** with the opportunities it provides. Let's look at the general view and template.

**GENERAL COMPOSITION** 

```
The most basic C# class
using command is
                         using System.Collections;
                                                                                    MonoBehaviour class is
used to call the
                         using System.Collections.Generic;
                                                                                    used to get its inheritance.
related C# libraries
                         using UnityEngine;
                         public class
                                                           : MonoBehaviour
                                            Script name
                                    // Variable definitions
                              void Start()
 class block with
                                     // Start is called before the first frame update
 the same name
 as the <u>C</u># file
                              void Update()
                                      // Update is called once per frame
                              3
                         }
```

C# codes-scripts consist of **blocks**. Commands other than library and variable definitions **must be** written in a **block** whose boundaries are defined by { } braces.

#### **GENERAL COMPOSITION**



Variables are used to carry parametric values inside or outside the block. Space is allocated in **memory** according to their types. These memory areas are assigned values according to their **numerical**, **verbal** or **type**.

The code developer determines variable names. While making a definition, many special characters (*i.e.*,  $\ddot{o}$ ,  $\ddot{s}$ ,  $\ddot{u}$ ,  $\ddot{g}$ , *i*, c,  $\dot{a}$ ,  $\ddot{a}$ ,  $\ddot{a}$ ,  $\ddot{a}$ ,  $\ddot{n}$ ,  $\ddot{B}$ ,  $\ddot{e}$ ,  $\ddot{y}$ ), spaces, and dashes, **cannot** be used. One of the most important elements besides variables is **functions-methods**. Commands are written into existing or created methods according to the character of the software to be developed.

# **Variables**

In the operation z=x+y, x, y, and z are variables.

Variables are valued carriers allocated space in computer memory. They can be for any purpose, such as verbal, numerical, vectorial, etc.

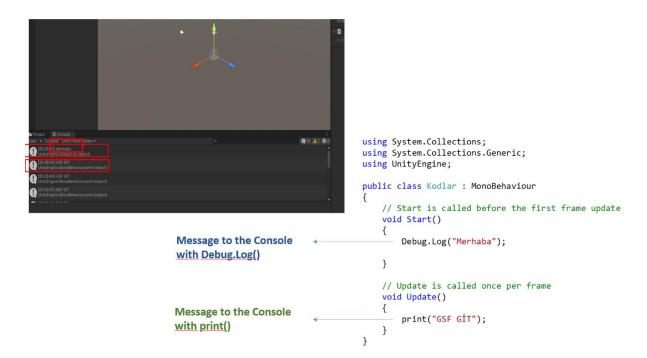
int a=10, b, c; int[] dizi;	// integer // integer array
dizi=new int[20];	// allocation for 20 elements in the memory
dizi[0]=27;	// assignment to the first element of the array
tamsayi=new List <int>();</int>	<pre>// definition of list-type variable</pre>
tamsayi.Add(10);	// assignment of 10 to the first element of the array
tamsayi.RemoveAt(0);	// <u>removing the first</u> element of <u>the array</u>
float x, y, z;	// <u>float number variables</u>
string ad[30];	// character type variables
bool basla=true;	// variable that can be true or false
Vector3 koordinat;	// 3 dimensional vectorial coordinate

and many similar variable definitions special to Unity

C/C++/C# statements are terminated with ;. Missing this character is an error source that will prevent the program from running.

Inclusion of library <	<pre>using System.Collections; using System.Collections.Generic; using UnityEngine;</pre>
name as the script 🛛 🖌	public class KameraKontrol : MonoBehaviour
externally accessible variable externally inaccessible variable	<pre>public OyunKontrol oyunK; private float hassasiyet = 5f; private float yumusaklik = 2f; Vector2 gecisPos; Vector2 camPos; GameObject oyuncu;</pre>
Method that will run once at	<pre>// Start is called before the first frame update     void Start()</pre>
the beginning	<pre>{     oyuncu = transform.parent.gameObject;     //camPos.x = 12f;     //camPos.y = 12f;</pre>
	<pre>} // Update is called once per frame</pre>
Method that will work during run time	<pre>void Update() {     if (oyunK.oyunAktif)</pre>
	<pre>Vector2 farePos = new Vector2(Input.GetAxis("Mouse X"), Input.GetAxis("Mouse Y")); farePos = Vector2.Scale(farePos, new Vector2(yumusaklik * hassasiyet, yumusaklik * hassasiyet));</pre>
	<pre>gecisPos.x = Mathf.Lerp(gecisPos.x, farePos.x, 1f / yumusaklik); gecisPos.y = Mathf.Lerp(gecisPos.y, farePos.y, 1f / yumusaklik); camPos += gecisPos; transform.localRotation = Quaternion.AngleAxis(-camPos.y, Vector3.right);</pre>
	<pre>oyuncu.transform.localRotation = Quaternion.AngleAxis(camPos.x, oyuncu.transform.up); }</pre>
	<pre>if (Input.GetKey("escape")) {     Application.Quit(); } </pre>
	}

The output of a script that gives **Console** output will be seen in the **Console** window in Unity.



Since the subject of coding is very comprehensive, existing coding examples will be repeated and better understood over time. Since this is not the main goal of this course, only quick and superficial information is included. However, it is useful to briefly explain two very common patterns such as **if()** and **for()**.

In the program flow, some operations will sometimes develop conditionally. **If**, **else** and **if-else** conditional statements are created to express this. General template:

## if(condition)

{

### command(s)

}

Here, the part inside the if block will be executed if the condition expression is true. Example:

int a, b=5;

if(b>0) { a=b\*b; }

**else** pattern is used to code situations that only have two possibilities, while **if else** conditional sentences are used when there are more than two situations.

If the same process is repeated many times in coding, they are included in repeating patterns called cycles. Although there are different methods for these, the most commonly used is **for** block patterns.

### for(initial value; final value; increment/decrement amount)

{

```
command(s)
```

}

```
Example:

x=0;

for (i=1; i<=100; i++)

{ x=x+i;

print(x);

}
```

Here, the value of the integer variable **x**, which is initially *zero*, is increased *one by one* until it reaches *100*, and each value is printed in the **Console** window. If you notice, the variable **i** is used as a **counter** and also as an **increment** element in the command. In the block, **i=1**; is assigned as the *first* value, it is increased *one by one* with the command **i++**; and if it is *less than or equal* to *100* with the expression **i<=100**, the command(s) in the block are **executed** again. The increment and print command lines are written once in the **block** and executed *100* times.

It will be possible to see C# codes in various trainings. One of the most important learning methods is to repeat these codes first and gradually enter the process of understanding, comprehending, interpreting and producing by developing basic knowledge.

After the basic part and code composition information, let's see the use of **C# Script** in our Unity project.

Since the C# Unity library has a large volume, let's start from the simplest level of codes and progress by adding new information according to specific needs.

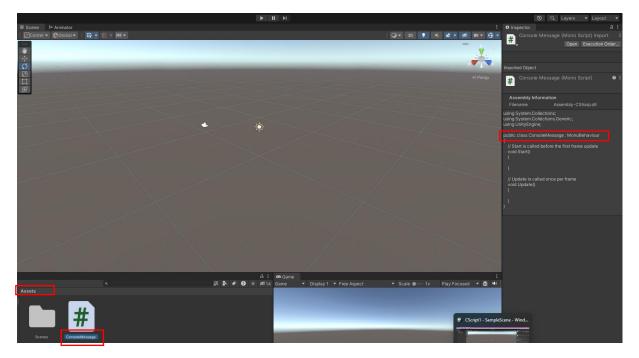
# 4.2.C# Scripts in Unity

Let's start seeing how C# information can be used in Unity. For this, we will create a new project. Create a code file with **Create>C# Script** by clicking the right button of the mouse in the **Assets** section.

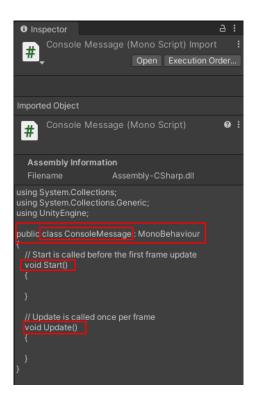
CScript1 - SampleScene - Windows, Mac,					- 0 ×
File Edit Assets GameObject Compor	ent Services Window Help				
Өке • 🔺					🧐 Q, Layers 🔹 Layout 🔹
+• All 8	🛛 📝 Conter 🕶 🕲 Global 🕶 🛤 💌 🛤 💌 🕬 💌			Q = 20 💡 K. 🛫 = 🕫 HI = 🔂	📲 🛌 Assets (Default Asset) 👘 1
y ♥ SampleScene ! @ Main Carrera ♡ Directional Light			Folder C#Script 20 Visual Scripting Shader Shader Variant Collection Testing Reyable: Assembly Definition Assembly Definition	e beg	Geen
		Create Show in Explorer Open	> TextMeshPro Text	;	
		Delete Rename Copy Path Alt+ Ctrl	Scene Scene Template Scene Template From Scene Scene Template Pipeline		
		Open Scene Additive View in Package Manager	Prefab Variant		
		Import New Asset Import Package Export Package Find References In Scene Select Dependencies	Audio Mixer Material Material Variant Lens Flare Bender Texture		
		Refresh Ctri Reimport	*R Lightmap Parameters Lighting Settings Custom Render Texture		
		Reimport All	Animator Controller	Maria Maria Maria Maria	
Project 🔛 🗟 Console		Extract From Prefab	Animation		3
+ •		Update UXML Schema	Animator Override Controller	▼ Scale 🛛 🛶 1x Play Focused 🔹 🗸 🗰	
🕨 🖈 Favorites		Generate Lighting Ctrl+Shift	+L Avatar Mask		
r ⊆er Assets ⊳ En Packages		Open C# Project View in Import Activity Window	Timeline Signal		
e E Packagos		Properties Alt	+ P Physic Material		
			GUI Skin Custom Font		Asset Labels
	Second		Legacy Ul Toolkit	<b>&gt;</b>	AssetBundle None + None +

The point to be noted here is that special characters should not be used in file naming, similar to naming web pages. In our study, the **C# file** was named **ConsoleMessage.cs**. The file will be opened in the **Assets** window with the standard **C# template**, and accordingly, the **class** in the program will have the same name as the file name.

It is very important that the file naming process is synchronized with **Create>C# Script** and that the **class** name in the created file has the same name. If the file name and class name are not the same, the program will be **incorrect**. If, by mistake, the C# file name and the class name are different, the class name must be <u>changed</u> and **matched** according to the file, or the file name must be changed according to the **class** name.



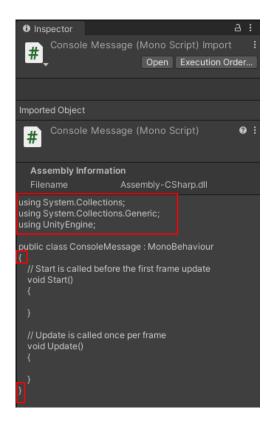
The file content is displayed in the **Inspector** section. This template code consists of a **class ConsoleMessage** block and two methods (functions) named **Start()** and **Update()**.



The C# code sentences to be used in the program are written inside the **class**, and method (function) blocks are created below it.

As seen here, the **methods** (functions) are automatically named within the framework of this template in which the program will be written, and their boundaries are determined with {...}.

In order to access the commands in the library, the program requires using the files that contain them. Therefore, our template starts with **using** ... statements and access is provided to the library commands.



Now let's start developing simple programs on this template.

# 4.3.Console Messages

As the name of the project suggests, our program aims to send messages to the console window of the Unity editor. In other words, messages will be sent to the screen in the codes, and these will be displayed in the Console window. Let's double-click on the **ConsoleMessage.cs** file and open it in Visual Studio 2022.



	Git Project Build Debug Test Analyze Tools Extensions Window Help   P Search + CScript			- 🛛 ×
	📴 🖓 - 🖓 -   Debug -   Any CPU - 🕨 Attach to Unity - 🗅 🍏 -   🎼 🖓 🖕 📽 🧡 🆕 🕼 🗍	1 <sup>98</sup> ■ <b>□</b> 负负间。		13 F
ConsoleMessage.cs +	×	10.0 m	- 0	Solution Explorer • 9 ×
2 2 4 4 4 87 5 6 7 8 9 9 10 11 12 12 13 14 14 15 16 15 16 17 17 19		- 🗞 San()	<u>^</u>	<ul> <li>The second</li></ul>
Show output from:	• ] = = = = = = = 0			
Error List Output				4 F Solution Explorer Git Changes

The content we see in the **Inspector** is seen in the compiler editor for processing. **Start()** method becomes active when the program runs and runs once. **Update()** runs 24/60 times per second, depending on the screen frequency. Here, it will be enough to write it in **Start()**.

**Debug.Log()** command is used to send a message to the screen (**Console**). **print()** command can also be used for the same purpose.

To use commands, we can write our message between two "...." For example:

## Debug.Log("UNITY");

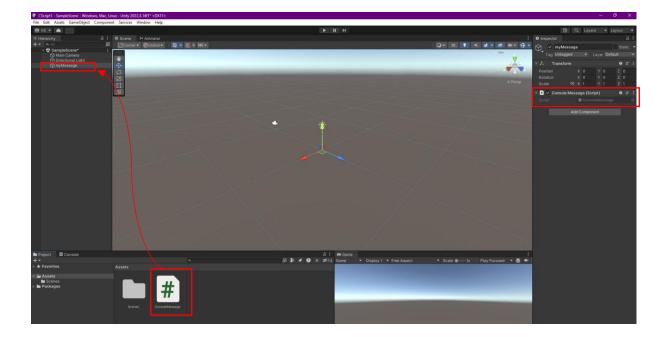
Command sentences must end with ;. Write the command and save it by pressing **Ctrl+S** or clicking on the icon

	00 File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help   夕Search - CScript1 薬の・の 1週・盛 12月18 タ・マ・「Debug - Any-CPU - ▶ Attachto Unity・♪ ① - ■ 同 要 該 12世 12日 日 日 日 日 日 日 日					
ConsoleMessage.cs • X ConsoleMessage • @consoleMessage 1 Vusing System.Collections;						
ŝ	S Assemb	oly-CSharp	👻 🔩 ConsoleMessage	<ul> <li>Start()</li> </ul>		
Lie Lie		1 ~us	sing System.Collections;			
		2 us	sing System.Collections.Generic;			
		3 us	sing UnityEngine;			
		4				
	10	5 ypu	ublic class ConsoleMessage : MonoBehaviour			
		6 {				
		7	<pre>// Start is called before the first frame update</pre>			
		8 void Start()				
	9					
		100	Debug.Log("UNITY");			
		11				
		12	}			
		13				
	14 // Update is called once per frame					
		15	void Update()			
		16	{			
		17				
		18	}			
		19 }				
		20				

When we return to Unity, this change will also be visible in the **Inspector**.

<ul> <li>Cscript1 - SampleScene - Windows, Mac, Ur</li> </ul>			
File Edit Assets GameObject Componer	Services Window Help		
⊖ KE • 🔺			<ol> <li>Q. Leyers + Leyout +</li> </ol>
	B Scene      ≻ Animator		O Inspector à 1
+ • (m All 20	Conter 🔹 ChGlobal 🐑 📴 👻 🖂 👻 🗰	🔾 + 20 🥊 K, 🛃 + 55 🗰 + 🔂 +	Console Message (Mono Script) Import 1
∀ SampleScene ! Main Camera			Open Execution Order
Directional Light	*	V ·	
	Ø		Imported Object
			Console Message (Mono Script) 9 1
	● ÷ C) □ □ □ □ □ □ □ □ □ □ □ □ □		# Console Message (Mono Script) • I
			Assembly Information
			Filename Assembly-CSharp.dll
			using System.Collections;
			using System Collections Generic;
	•		using UnityEngine;
			public class ConsoleMessage : MonoBehaviour
			{     // Start is called before the first frame update
			void Start()
			DebugLog("UNITY");
			2
			// Update is called once per frame
			void Update()
			, )
Project Console		à I 👁 Game I	
+ •		nt so came . nt 14 Same . ■ Display 1 . Free Aspect . Scale . 1x Play Focused	N
> 🖈 Favorites	Assets		No.
v 🗃 Assets			
Scenes			
🗁 🖿 Packages			
			k
	Scenes ConsileMessage		

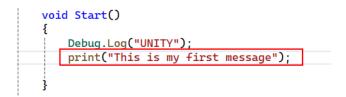
In order to see the result of a line we added to the template, the project and the code file must be linked. For this purpose, let's click right and add an empty object with **Hierarchy>Game Object**. This object is named **myMessage**. Let's drag the **ConsoleMessage.cs** file and link it to this object.



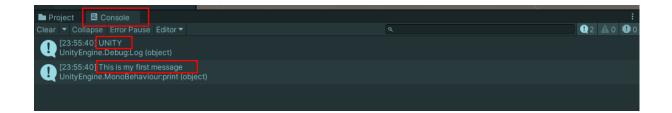
Now, click on **Play Mode** and see the message we wrote at the bottom of the screen. The message can also be viewed in the **Console** section.

File Edit Assets GameObject Compor	nent Services Window Help		
😝 КЕ 👻 📥			
			1
+ - ( AII	🔍 🔟 Center 👻 🌐 Global 👻 🗮 👻 🖽 💌		🔾 🕈 2D 📍 🥵 🛫 🕫 💷 🕂 🕁 🖛
r			Perp
Project E Console			
+*		🖉 🦆 🕖 🗯 🕫 14 Game 🔻 Display 1 🔻 Free Aspect	🝷 Scale 🛛 🛶 1x 🛛 Play Focused 👻 🎘 🕪
► ★ Favorites			
≄ barAssets Mar Scenes ⊳ Mar Packages	Scenes Conscient/resarge		
Q UNITY			

Add another line to the program. This time, see that we can also transfer the message with the **print()** command.



Save and return to Unity and see the result in the Console window in Play Mode.



## 4.4.Variables

As stated in the general definitions, program variables are program elements that will carry numbers or various values. Variables are **memory location** names that are allocated space in memory and can take on multiple values within the program flow, and these assigned values can change.

Variables have types according to the information to be assigned and are categorically defined as **public** and **private**, which basically means that they are *open* or *closed* to **external access**.

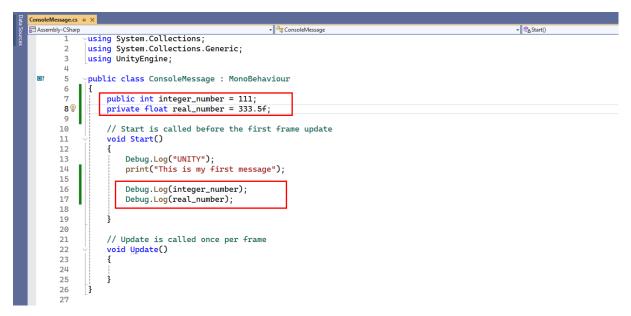
Immediately following this definition is the **declaration type**: **int** (integer), **float** (real number), **string** (character), **Vector3** (3D vector coordinate information), etc. For example:

### public int sayilar;

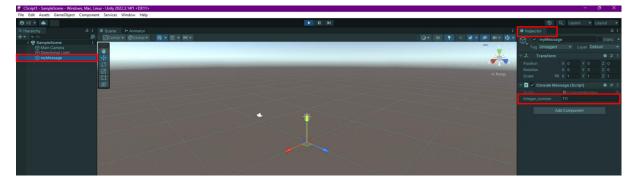
### private float angles;

Within the framework of visual programming logic, it is possible to define a variable for each parameter in the **Inspector** section of the editor and assign values to these fields. A variable is normally considered **private** unless it is described in front of another definition.

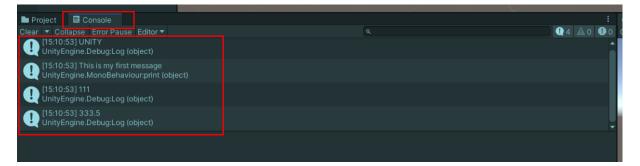
Let's do some applications in our project. Define two variables named **integer\_number** and **real\_number**, the first one is an integer, and the second one is a *real* number. Also, assign the initial values to the memory in the definition line. In real number assignments, the expression **f** is added to the end of the number, meaning **float**.



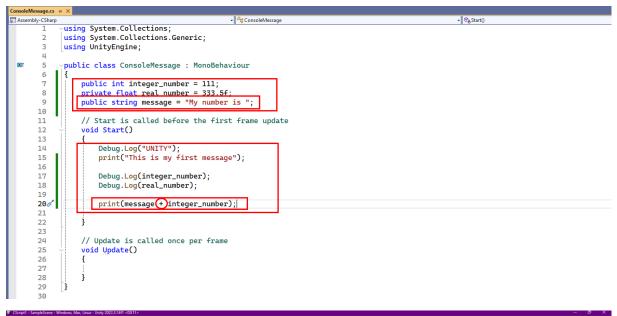
Now, save and return to Unity and run **Play** Mode. If you pay attention, you can see that the publicly defined variable has become accessible to outsiders in the **Inspector**. It is also possible to enter a new value in this field.



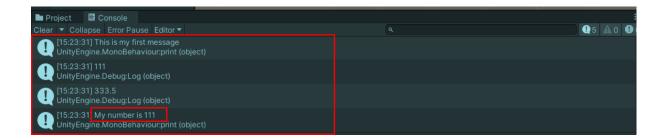
The results of the four message commands in the program will be displayed in the **Console** window.



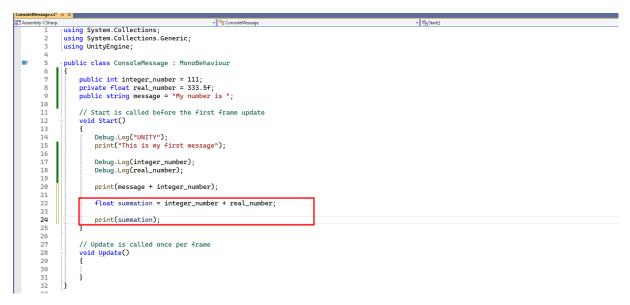
A character (**string**) variable can also be combined with numerical expressions and displayed on the screen. The + operator used here is not only for arithmetic addition but also for using two values together.

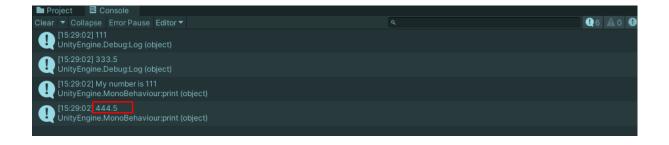






Let's give an example of arithmetic operations. Here, we define a variable in the **Start** method (function) and assign the **sum** of the values of the two previously defined variables.





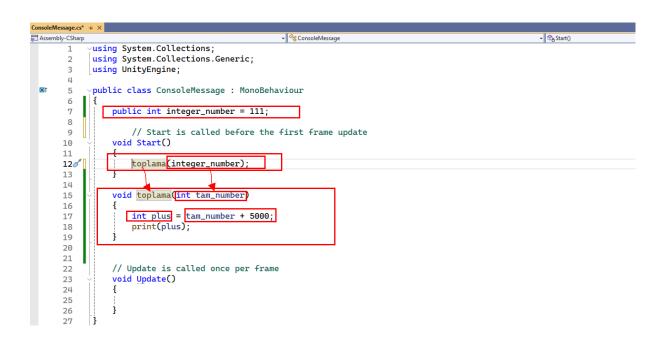
# 4.5.Methods/Functions

Apart from the **Start()** and **Update()** methods (functions), we can also develop our methods. In these function blocks, which have characteristics such as **type**, **name**, and parentheses, pre-definitions such as public, private, and **type definitions** such as **int**, **float**, **and void** can be made.

In order to call and use the methods created in the program flow, it is sufficient to write their names. A **void**-type method is not expected to return a value when it returns to the line it was called. It is not necessary to transfer data to these types of methods. However, in methods with a type such as **int** or

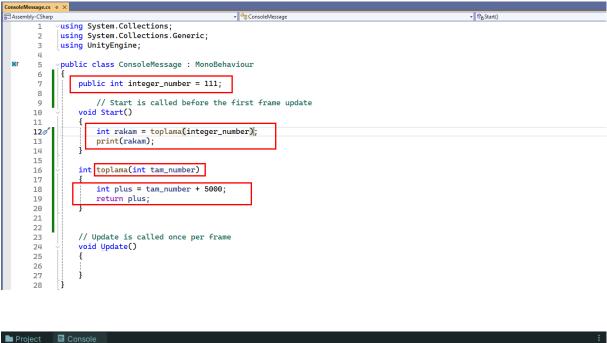
**float**, it is expected to produce a value of this type when it returns to the line it was called. In addition, there is a value transfer for the method to be processed.

Now, remove some lines from the same project and focus on this issue. Leave only the **integer** variable in the variable definition. Create a **void add()** method inside the **Start()** method. *integer\_number* variable value is transferred to this method. In the method definition line, a local variable called *int tam\_number* is defined to hold this transferred value. *tam\_number* value is directly equivalent to *integer\_number*, i.e. 111. After defining another variable named *int plus* in the method, the *integer\_value + 5000* operation is performed for the value assignment. The result, 5111, is printed on the **console**.





If the **add()** function had been int **add()** instead of **void**, the same result would have been obtained. However, the **int add()** function was expected to return a value with the **return** command.





We can achieve the same result by performing more variables and operations. The basis of objectoriented programming logic is to create modular program parts. Complex integrated and iterative software provides great savings from repeating the same operations over and over again.

# 4.6.Arithmetical Operators

Arithmetic operators are used for mathematical operations. These are basically four operations and some additional special operators:

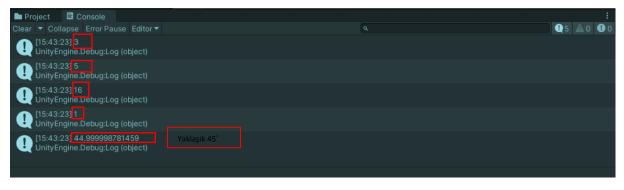
Operator	Function	Example C# code
+	addition	$x=x+y; \equiv x+=y;$
-	subtraction	$w=w-q; \equiv w-=q;$
*	multiplication	$c=c^{*}(a+b); \equiv c^{*}=(a+b);$
/	division (quotient)	d=d/5; ≡ d/=5;
++	add 1	$x++; \equiv x=x+1; \equiv x+=1;$
	decrease 1	$x-; \equiv x=x-1; \equiv x-=1;$
%	mod operator (remainder)	z=10%3 (remainder 1)

Also, some commonly used C# Mathf class methods are listed:

Function	Math function	Meaning	Example C# code
Sqrt(x)	$\sqrt{x}$	Square Root	y=Mathf.Sqrt(x);
Abs(x)	x	Absolute Value	z=Mathf.Abs(x+y);
Pow(x,y)	<i>x<sup>y</sup></i>	Power	w=Mathf.Pow(x,y);
Tan(x)	tan(x)	Tangent	angle=Mathf.Tan(x*(Mathf.Pi/180)); // radian↔degree
Atan(x)	arctan(x)	Arc Tangent	val=Atan(angle*(180/Mathf.Pi)); // radian↔degree

Let's re-arrange the C# codes of our project in Visual Studio. Here, the square root of 9, the absolute value of -5, the 2nd power of 4 (square of 4), the tangent of 45, and the arc tangent of 1 are calculated. In the C# programming language, angles are in radians. If we want to work in degrees, we need to convert from radians to degrees and from degrees to radians.

using System.Collections; using System.Collections.Generic; using UnityEngine; public class ConsoleMessage : MonoBehaviour { float number1 = 4f, number2 = 2f; float number3 = 45.0f; float number4 = 9f; float number5 = -5f; // Start is called before the first frame update void Start() double x = Mathf.Sqrt(number4); Debug.Log(x); double y = Mathf.Abs(number5); Debug.Log(y); double p = Mathf.Pow(number1, number2); Debug.Log(p); double ang = Mathf.Tan(number3 \* (Mathf.PI / 180)); Debug.Log(ang); double val=Mathf.Atan(1)\*(180/Mathf.PI); Debug.Log(val); 3 // Update is called once per frame L void Update() { } 3



\*Yaklaşık: approximately

# 4.7. Conditional Statements and if statements

In some cases, operations may or may not be done in the program flow. We need to reflect these in the codes with conditional statements.

if(condition)

 $\{...\}$ 

In the general template, we can write as follows: if the condition in the parentheses is met (**true**), the part inside the if block is executed. Otherwise (**false**), these lines will not be executed.

It is necessary to make comparisons in conditional sentences. Relational operators are used for this purpose:

>	greater	if(a>b)
>=	greater or equal	if(x>=y*z)
<	smaller	if(alfa<90)
<=	less or equal	if(u+w<=p/q)
==	equal	if(b==c*c)
!=	not equal	if(b*b-4*a*c != 0)

Now, add sample if statements to the previous program and see the results:

```
ConsoleMessage.cs 👳

    Age ConsoleMessage

Assembly-CSharp
                                                                                                                              - 😪 Start()
         1
               using System.Collections;
         2
               using System.Collections.Generic;
         3
              using UnityEngine;
        4
 ot
        5
               public class ConsoleMessage : MonoBehaviour
        6
               ł
        7
                    float number1 = 4f, number2 = 2f;
                    float number3 = 45.0f;
        8
        9
                    float number4 = 9f;
       10
                    float number5 = -5f;
       11
       12
                         // Start is called before the first frame update
       13
                   void Start()
       14
                    ş
       15
                        double x = Mathf.Sqrt(number4); Debug.Log(x);
       16
                        double y = Mathf.Abs(number5); Debug.Log(y);
                        double p = Mathf.Pow(number1, number2); Debug.Log(p);
double ang = Mathf.Tan(number3 * (Mathf.PI / 180)); Debug.Log(ang);
       17
       18
                        double val=Mathf.Atan(1)*(180/Mathf.PI); Debug.Log(val);
       19
       20
       21
                        if(x<y) { double z= x+y; Debug.Log(z); }</pre>
                        if(p>=y) { double q = p * y; Debug.Log(q); }
if(ang==1) { Debug.Log("Tan"+number3+" = "+ang); }
       22
       23 🖗
       24
       25
       26
            I
                        // Update is called once per frame
       27
                    void Update()
       28
                    Ł
       29
       30
                    }
       31
              3
```



In cases where there are multiple conditions, it is necessary to use logical operators to connect these conditions:

Operator	Meaning	Example C# code
&&	and	if ( a+b >= c+d && a*a<(c/d)) { }
	or	if( b*b-4*a*c > 0    a+b+c <d*e) td="" {="" }<=""></d*e)>
!	not	if (!a) { } // converted to true>false and false>true

Let's add an **if** statement to the project code that connects three conditions with **&&** and see that a message is sent to the console because all of them are true.

```
nsoleMe New Project (Ctrl+Shift+N)
Assembly-CSharp
                                                                  - ConsoleMessage
                                                                                                                                     - CaStart()
                using System.Collections;
         1
         2
               using System.Collections.Generic;
         3
               using UnityEngine;
               public class ConsoleMessage : MonoBehaviour
{
         4
         5
         67
                     float number1 = 4f, number2 = 2f;
float number3 = 45.0f;
         8
         9
                     float number4 = 9f;
        10
                     float number5 = -5f;
       11
                          // Start is called before the first frame update
        12
        13
                     void Start()
        14
                     $
        15
                          double x = Mathf.Sqrt(number4); Debug.Log(x);
        16
                          double y = Mathf.Abs(number5); Debug.Log(y);
                         double p = Mathf.Pow(number1, number2); Debug.Log(p);
double ang = Mathf.Tan(number3 * (Mathf.PI / 180)); Debug.Log(ang);
        17
        18
                          double val=Mathf.Atan(1)*(180/Mathf.PI); Debug.Log(val);
        19
        20
                          if(x<y) { double z= x+y; Debug.Log(z); }
if(p>=y) { double q = p * y; Debug.Log(q); }
if(ang==1) { Debug.Log("Tan"+number3+" = "+ang); }
       21
       22
        23
        24
                         if (x < y && p > y && ang < x * y) Debug.Log("ALL CONDITIONS ARE CORRECT");
       25 8
        26
        27
        28
            I
                         // Update is called once per frame
                     void Update()
        29
        30
                     ł
        31
                    3
        32
        33
```



For the expressions connected with **&&** to be true, all of them must be true. For only one of the conditions associated with **||** to be true, it is enough for the true result to come out of the parenthesis and for the operation (block) connected to the **if** statement to be executed.

The code information is provided below in text form that allows copy/paste.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class ConsoleMessage : MonoBehaviour
ł
  float number1 = 4f, number2 = 2f;
  float number3 = 45.0f;
  float number4 = 9f;
  float number5 = -5f;
    // Start is called before the first frame update
  void Start()
  {
    double x = Mathf.Sqrt(number4); Debug.Log(x);
    double y = Mathf.Abs(number5); Debug.Log(y);
    double p = Mathf.Pow(number1, number2); Debug.Log(p);
    double ang = Mathf.Tan(number3 * (Mathf.PI / 180)); Debug.Log(ang);
    double val=Mathf.Atan(1)*(180/Mathf.PI); Debug.Log(val);
    if(x<y) { double z= x+y; Debug.Log(z); }
    if(p>=y) { double q = p * y; Debug.Log(q); }
    if(ang==1) { Debug.Log("Tan"+number3+" = "+ang); }
    if (x < y && p > y && ang < x * y) Debug.Log("ALL CONDITIONS ARE TRUE");
  }
   // Update is called once per frame
  void Update()
  {
  }
}
```

## 4.8.Array

They are variables that can carry more than one data and have space made available for them in memory. Many similar data in accordance with the matrix structure are addressed and processed with index values within the group they belong to.

An **array** variable can be one or more dimensions. To allocate space in memory, there is a difference between square brackets [] in the definition sentence and a field size assignment with the new command.

int[] say;

### say= new int[5];

In the example, instead of the variable named say having a single value, it is specified as **int say**; and it is an array as **int[] say**; and then five integers are allocated in memory. The values in an array structure connected to the same variable name are separated from each other according to their **addresses/index** numbers in the array. Although this index sequence number is from 1 to 5 in normal mathematical matrices, the index number starts from **0** in C#. Therefore, the sequence numbers will be between *0-4*.

The following example shows which sequence number element of the **say** variable the value assignment is made to.

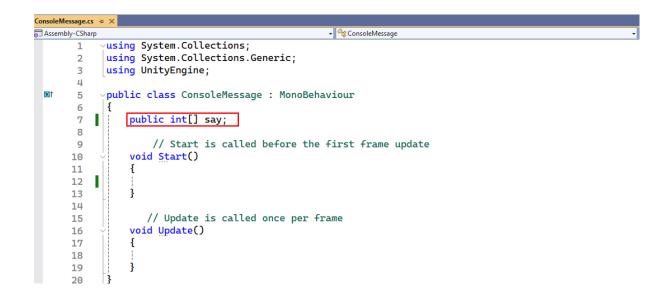
## say[0]=3;

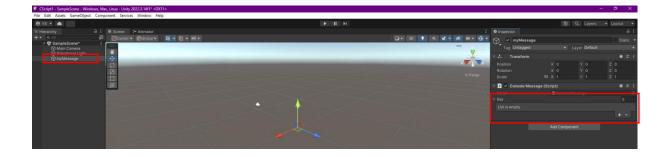
Here, the index value is 0. Its mathematical equivalent is the 1st row.

If we want to print to the console, it will be written in a similar way as

## Debug.Log(say[0]);

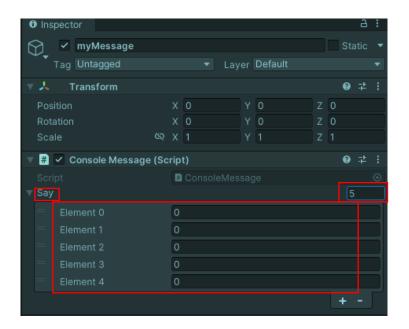
If the **public** property is given when defining the **say** variable, it can be accessed from the editor, size assignment and even element data entry can be done from here.





Here, both the number of elements of the **say** variable can be determined, and subsequently, data can be entered into the field that will be opened as many times as the number of elements determined.

For example, the number 5 was entered, and as a result, space was opened for 5 elements from the index value 0 to 4, and the initial value 0 was assigned to all of them.



## 4.9.List Variables

List variables have similar properties to array variables. However, it is not necessary to enter the matrix size (number of array elements) at the beginning. It is a structure that creates space for itself in memory as data is entered. The number of elements can increase and decrease. They are not fixed-sized like arrays but have a **dynamic** structure.

When making a definition, memory is created with the list declaration and **new** command.

List <int> rakam;

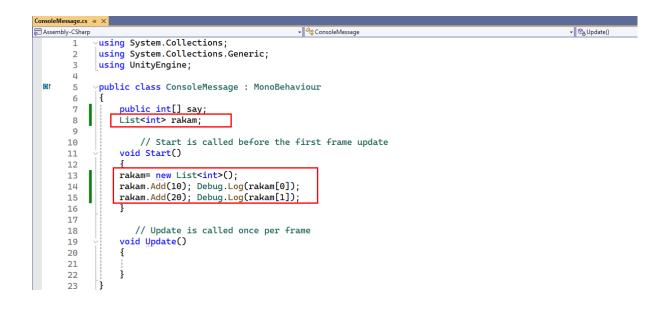
rakam= new List<>();

here, the variable type is declared.

rakam.Add(10);

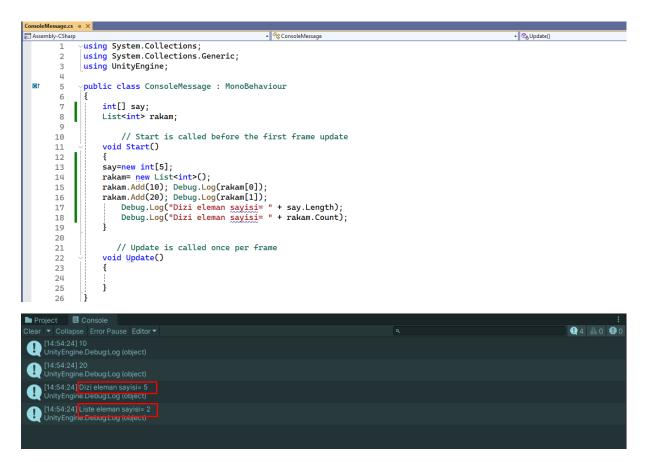
rakam.Add(20);

Add command is used to transfer data, and both spaces are made available in memory and data is assigned, respectively. The array structure and indexing method are the same. If we want to get output,





**digit.Count** and **count.Length** commands can be used to determine the number of elements in the list and array.



\*Dizi eleman sayisi: number of elements in the array

To remove an element from a list,

rakam.Remove(assigned value);

or

#### rakam.RemoveAt(index number);

can be used.

After the deletion process, the index order of the elements after the removed element is redetermined and the sequence number is updated.

Let's make an example where **for** structure we saw in the previous part of our topic loops between 0-4 5 times, and the **int i** variable controlling the loop will be assigned a value that is 100 times larger.

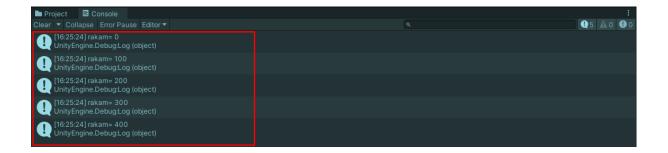
Assembly-CSharp	- 😤 ConsoleMessage	✓ 𝔅A Start()			
<pre>4 //using UnityEngine.Random; 5 8</pre>	aviour st frame update // Value assignment // Writing on the console	- 🖏Start()			
27 void Update() 28 { 29 } 30 } 31 }					
Project 🖪 Console		:			
Clear ▼ Collapse Error Pause Editor ▼		Q5 🗛 0 😡 0			
16:00:23] 0 UnityEngine.Debug:Log (object)					
[16:00:23] 100 UnityEngine.Debug:Log (object)					
(16:00:23) 200 UnityEngine.Debug:Log (object)					
(16:00:23) 300 UnityEngine.Debug:Log (object)					
16:00:23] 400 UnityEngine.Debug:Log (object)					

Another important **loop** structure that can be used with arrays and lists is **foreach**. A variable of the same type that can be used in our array or list is defined. The general template for our example is stated as follows:

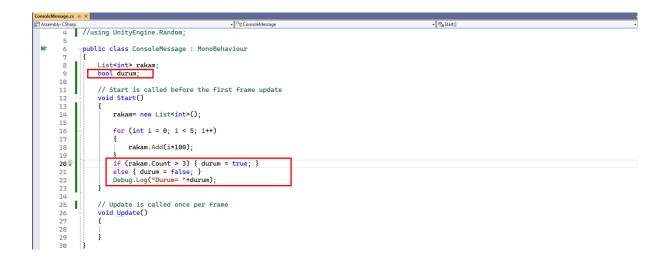
## foreach (int val in rakam)

If we reflect it on our code lines,

ConsoleMessage.	ionsoleMessage.cs 🕫 🗙				
Assembly-CSh		nsoleMessage	✓ C <sub>B</sub> Start()		
4	<pre>//using UnityEngine.Random;</pre>				
0 1 <b>0</b>	<pre>public class ConsoleMessage : MonoBehaviour </pre>				
7 8 9 10	List <int> rakam; // Start is called before the first fram</int>	ne update			
11 12	void Start()				
13 14	<pre>rakam= new List<int>();</int></pre>	-			
15 16 17 18 19	<pre>for (int i = 0; i &lt; 5; i++) {     rakam.Add(i*100); }</pre>				
20 21 22	foreach(int val in rakam)				
22 x 23 24 25	3	]			
25 26 27 28 29 30 31	<pre>// Update is called once per frame void Update() { }</pre>				
27 28 29 30					



**bool** type variables can only take two values, **true** and **false**. Let's give an example with codes.

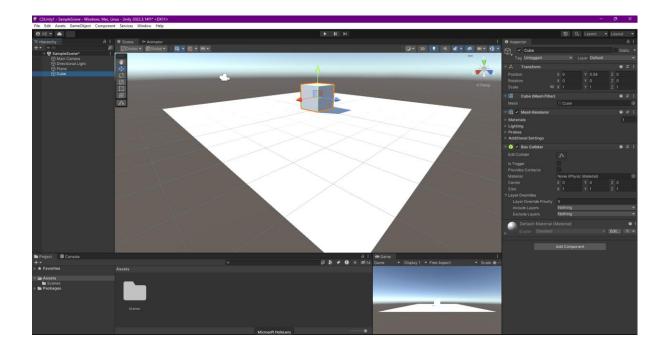


**bool** status variable is **true** if there are more than three numbers in the list. Otherwise, it is **false**.

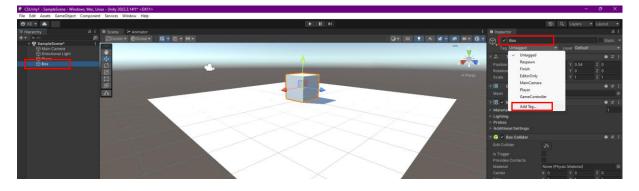


# 4.10.Basic C# Coding for Unity

In the following parts of our topic, let's see the basic and frequently used information about Unity objects and applications in **C#**. For this, let's start a new project. In the preparation phase, let's add a **Plane** and a **Cube** to our scene.



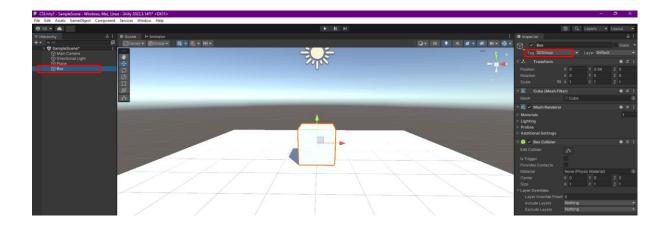
Let's change the name of the **Cube** object to **Box**.



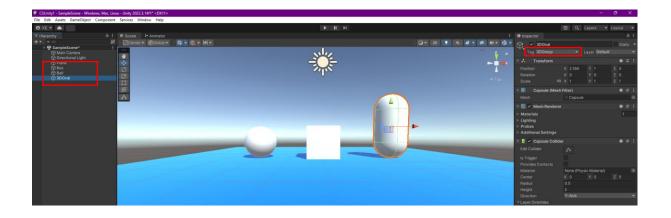
Let's create a new Tag named **3DGroup** for our **Box** object and assign it.

Inspector     Tags & Layer	s	a: 0≠:
▼ Tags Tag 0	3DGroup	1
► Sorting Layers ► Layers		+

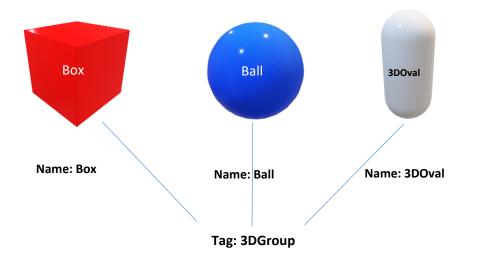
The principle of matrix indexing starting from **0** is also seen here in the listing of the **3DGroup** tag as **Tag 0**.



Similarly, let's add a **Sphere** and a **Capsule** to our scene. Let's name them **Ball** and **3DOval**, respectively, and let's name their **Tag 3DGroup**.



Let's see the **Name** and **Tag** information of our objects as a diagram. **Names** are seen as individual, and **Tag** information is seen as a common group name.



As you can see, the names of the three objects in our scene are different, and their labels are the same. **Name** and **Tag** are two different properties used to access the objects in the scene. **Name** can be used as an individual assignment for each object. However, let's remember that the same names can be given to different objects. **Tag** is generally labeled for grouping purposes.

In an application, for example, we may want to destroy each object based on its name, or we may want to destroy objects that are members of a group collectively. In such cases, we will need to access the **Name** or **Tag** information.

# 4.11.Object and Component Access

We need to access the object we want to process with code and the **Component** information in the **Inspector** section. Hierarchically, we need to access the object first and then the **Component** section.

For example, we want to access the **MeshRenderer** component (**Component**) of an object. Below is the access coding for two different situations.

In the **first case**, there is a **C#** file linked to the object. That is, the code file has already accessed the object to be accessed.

gameObject.GetComponent<MeshRenderer>(); or in shorter writing,

## GetComponent<MeshRenderer>();

We can access the **MeshRenderer** property of the object with command.

In the **second case**, we want to access the **MeshRenderer** component (**Component**) of a different object in the scene that our code is not bound to. When the code file is not bound to this object, we will need a more detailed description. At this point, we can use the **Name** property of the object.

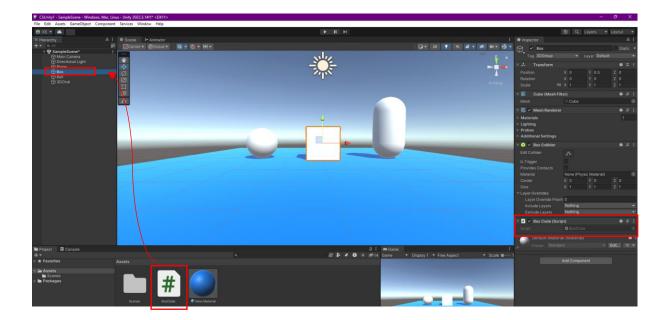
## GetObject.Find("Box").GetComponent<MeshRenderer>();

If we want to access all three objects, we can write our code using the common group Tag.

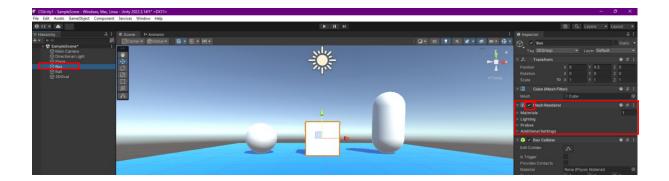
### GetObject.FindWithTag("3DGroup").GetComponent<MeshRenderer>();

## GetObject.FindGameObjectWithTag("3DGroup").GetComponent<MeshRenderer>();

Now, we can use this basic information in our Unity project. Let's create a **C#** Script to connect to our **Cube** object named **Box** and open it in Visual Studio (here, the file is named **BoxCode.cs**). Let's make it connected to our **Cube** (**Box**) object.



**MeshRenderer Component** contains settings related to the visibility of the object. The check box in the **Inspector>MeshRenderer** window of our object named **Box** shows whether this feature is active (**true**) or passive (**false**). Since it is currently active, the object is visible, and the box is **ticked**.



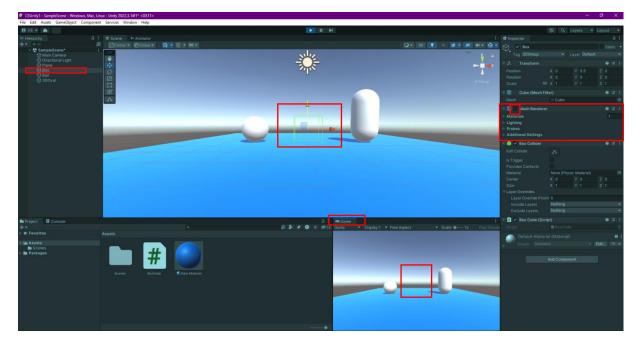
Let's make this object **invisible**. Unchecking the box means assigning the **bool** value called enabled for **MeshRenderer** to false as the code equivalent.

## GetComponent<MeshRenderer>().enabled=false;

Let's write this line in Visual Studio, save it and run it in Unity **Play** Mode.

BoxCode.	cs ⇒ X	
🕞 Assem	bly-CSharp	- 🕫 BoxCode
	1	<pre>vusing System.Collections;</pre>
	2	using System.Collections.Generic;
	3	using UnityEngine;
	4	
10	5	<pre>ypublic class BoxCode : MonoBehaviour</pre>
	6	
	7	// Start is called before the first frame update
	8	void Start()
	9	{
	100	<pre>GetComponent<meshrenderer>().enabled = false;</meshrenderer></pre>
	11	3
	12	
	13	// Update is called once per frame
	14	void Update()
	15	
	16	
	17	
	18	}

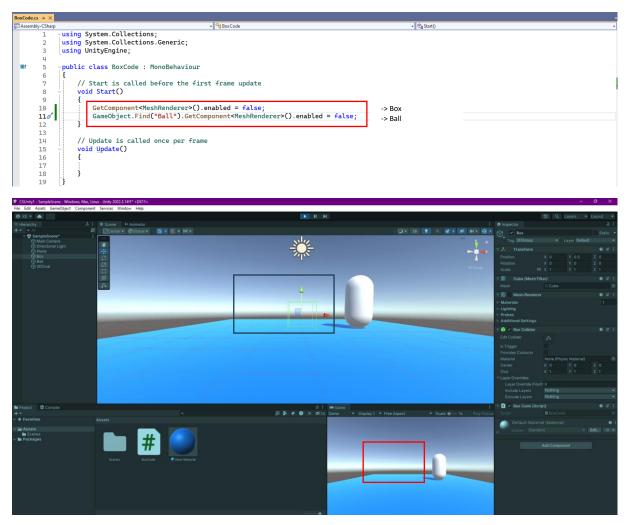
As can be seen, the **MeshRenderer** check box became **unchecked**, and the **Box** became invisible on the **Game** screen.



At this stage, let's access another object from our code file that it is not connected to and make it invisible. We can use the information we saw earlier to reach the **Sphere** object, which we named **Ball**.

GameObject. Find("Ball").GetComponent<MeshRenderer>().enabled=false;

Here, we first access the object named **Ball** with **GameObject.Find("Ball")** and the **MeshRenderer** active/passive property with **GetComponent<MeshRenderer>().enabled**. We set it equal to false to make it invisible.



If we want to make all objects with **3DGroup tags** invisible, we can use our array variable and **foreach** information since there is more than one object. Firstly,

GameObject[] objects;

Let's create an object matrix with a definition.

### objects=GameObject.FindGameObjectWithTag("3DGroup");

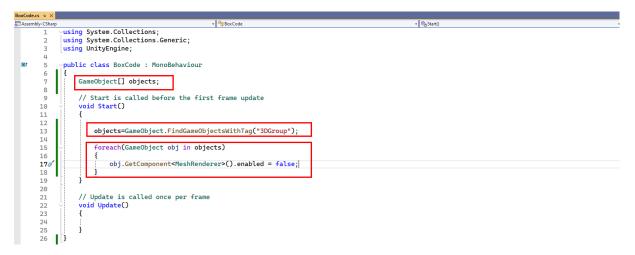
Let's include all objects tagged with **3DGroup** in the array with the coding,

foreach(GameObject obj in objects)

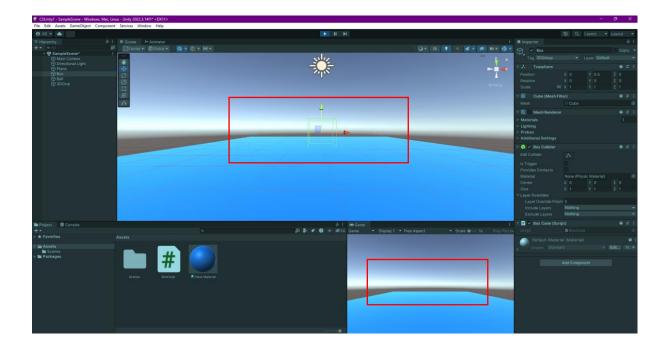
#### { obj.GetComponent<MeshRenderer>().enabled=false;

}

Let's uncheck the **MeshRenderer** property of the objects in the matrix with the conversion. Now, let's turn it into a script in Visual Studio.

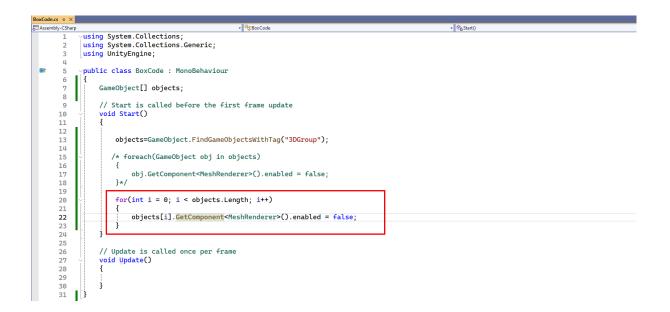


We can see that since the **Tag** value of the other three objects in our scene, except **Plane**, is **3DGroup**, they all become invisible.



Let's do the same process with the **for**-loop pattern to reinforce our knowledge about array variables. For this, use the **foreach** block,

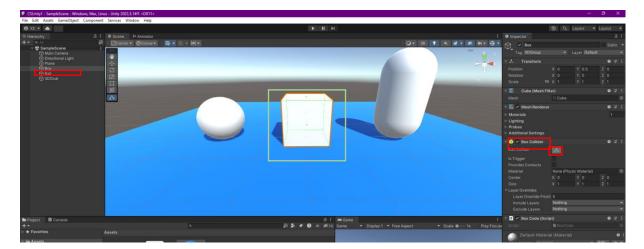
let's make a **comment** line by placing it between /\* .... \*/ and disable it. Then, create a **for** loop block that will loop between 0 and **objects.Length**.



# 4.12.Collision/Interaction of Objects in Unity - On Collision

In this section, let's try to answer the question of how objects can collide or interact with each other in our applications using code.

First, the **Inspector** content of our objects that will interact should contain **Collider** and **Rigidbody** components. **Collider** means collider, and **Rigidbody** means a physical, rigid body that also has **gravity**. Normally, the **Collider** property of cube, sphere and similar objects that we add with **Create>3DObject** is assigned depending on the shape. **Box Collider** is also added while creating our object named **Box**. **Sphere Collider** is directly assigned to the object named **Ball**, and **Capsule Collider** is directly assigned to the object name **3DOval**. A **collider** can be thought of as a wireframe surrounding an object. If we want to change its shape, we can **edit** its borders, enlarge or reduce it with the button



Let's add Inspector>Add Component>RigidBody to our Box, Ball and 3DOval objects.



**Collisions** can be in the form of a hard collision (**Collider**) and a soft collision (**Trigger**). For the type of interaction between a person entering the water and the water, the **Is Trigger**  $\Box$  box must be selected.

**Collision** control functions/methods, included in the C# Unity Engine library.

OnCollisionEnter()

OnCollisionExit()

**OnCollisionStay()** 

**OnCollisionEnter()** works once when first contacted. **OnCollisionExit()** works once when the collision is over. **OnCollisionStay()** runs continuously during the collision.

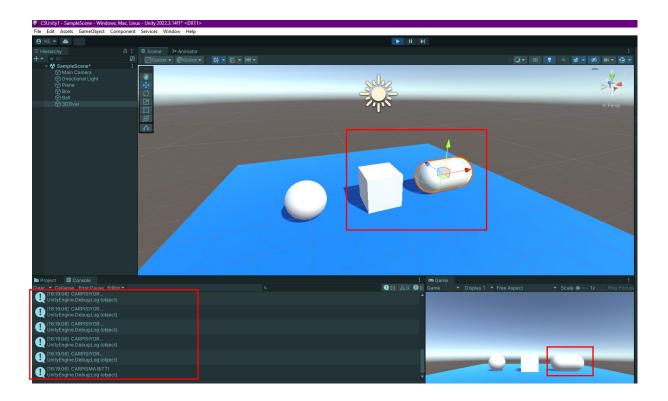
Let's start the application in our Project and write the collision script file. Since we don't need the **Start()** and **Update()** methods in these operations, we can delete them. Instead, let's write three methods that control the collision and let's write our message with the **Debug.Log** command to declare the stage where the collision occurred. The system assigns a collision variable in the parentheses of the **OnCollisionXYZ()** methods. The information about the object that was hit is stored in this parameter. It is possible to change the name of the variable if we want.

Translation:

SIMDI CARPTI (Now collided) CARPISMA BITTI (Collision ended) CARPISIYOR... (Now colliding)

0 File Edit View G	Sit Project Build Debug Test Analyze Tools Extensions Window Help 🔎 Search 🕶 CSUnity	
🖗 • 🐵   🎁 • 🚅 💾 🗑	🗃   🍤 • 🖓 •   Debug 🔹 Any CPU 🔹 🕨 Attach to Unity • ▷ 🍏 •   📭   🗟 🚽 📽 🤠 🍉 🕼 🗐	2월 🔲 🖓 🖓 📮
BoxCode.cs -> ×		
Assembly-CSharp	• RegBoxCode	<ul> <li>ConcollisionStay(Collision collision)</li> </ul>
2 u 3 u 4	<pre>ising System.Collections; ising System.Collections.Generic; ising UnityEngine; public class BoxCode : MonoBehaviour</pre>	
6   <b>1</b> 7   8   9   10	<pre>private void OnCollisionEnter(Collision collision) {     Debug.Log(" SIMDI CARPTI "); }</pre>	
11 12 13 14 15 16	<pre>private void OnCollisionExit(Collision collision) {</pre>	
18 17 18 19 <b>6</b> 20 21	<pre>private void OnCollisionStay(Collision collision) {     Debug.Log(" CARPISIYOR  "); }</pre>	

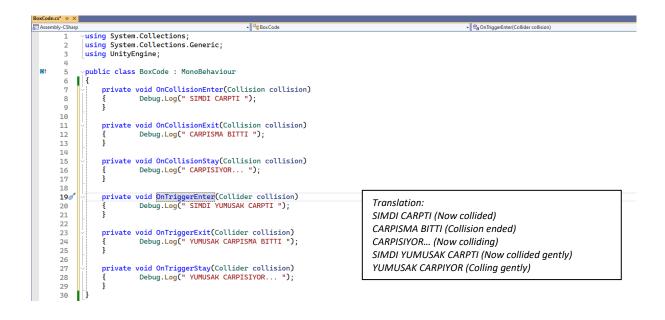
In **Play** mode, let's **drag** the capsule to our cube object, crash it and release it. The impact will knock over the capsule with physical properties. In the meantime, messages will appear in the **Console** window indicating that the collision has occurred, is ongoing and has finished.



If we want the capsule to tip over, click the **X**, **Y** and **Z** check boxes under **RigidBody>Constraints>Freeze Rotation**.

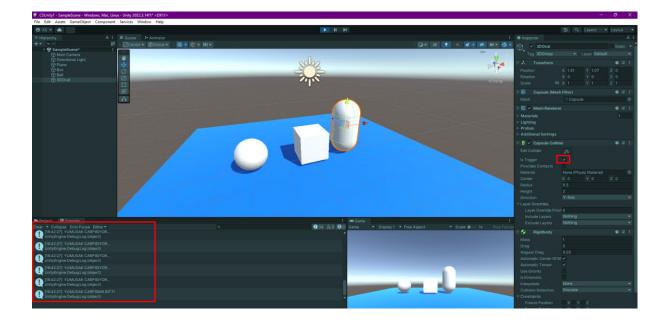
😎 Game								Exclude Layers		Nothing		
Game		Display 1	Free Aspect	-	Scale (	▶— 1x	Play Focuse					
•								v 🕤	Rigidbody		0 ≓ :	
								Mass		1		
								Drag		0		
								Angul	lar Drag	0.05		
									natic Center Of M	~		
					_			Autor	natic Tensor	~		
								Use G	Gravity	~		
								ls Kin	ematic			
	-	_				_	_	Interp	olate			
	_						_	Collis	ion Detection	Discrete		21.
								▼ Const	traints			
								Fre	eze Position			
								Fre	eze Rotation	✓ X ✓ Y 🖌 Z		
								⊤ Layer	r Overrides			
									lude Layers	Nothing		
								Exe	clude Layers	Nothing	- 	

Let's also see the **soft collision**. For this, make a small change in the code lines. With the same logic, place the **OnTriggerEnter()**, **OnTriggerExit()** and **OnTriggerStay()** methods and the relevant **Debug.Log()** messages.



Notice that in the **OnTrigger** commands, the variable definition in parentheses is **Collider**.

When we run it in **Play** Mode like this, we will still get the hard collision messages. In order to see the result of the codes we added, we need to click the **Is Trigger** check box of at least one or two of the objects that will collide. After this is done, the capsule that collides with the cube will give the soft collision messages. It will also not fall over...

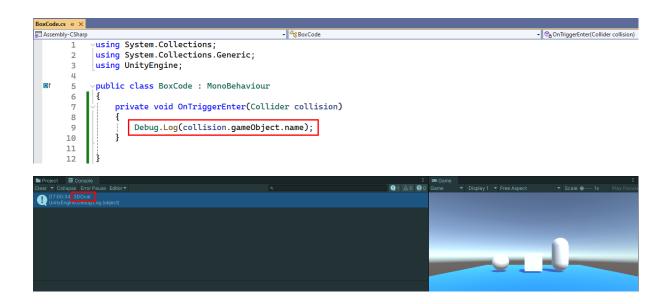


We have stated that the variable that is automatically opened in our method carries information about the collided object. To see it concretely, write the command

#### Debug.Log(collision.gameObject.name);

#### to the **OnTriggerEnter()** method.

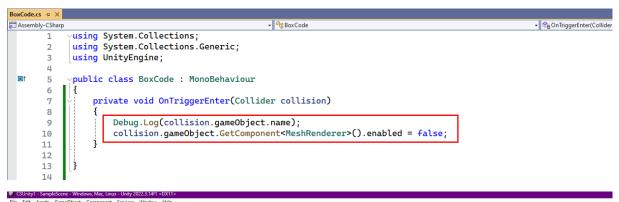
Let's drag the cube into Unity again and collide it with the capsule. See the message in the **Console**. Do not forget that the code file is connected to the cube, and the **3DOval** information, which is the object it collides with, comes to the **Collider** collision variable in the method.

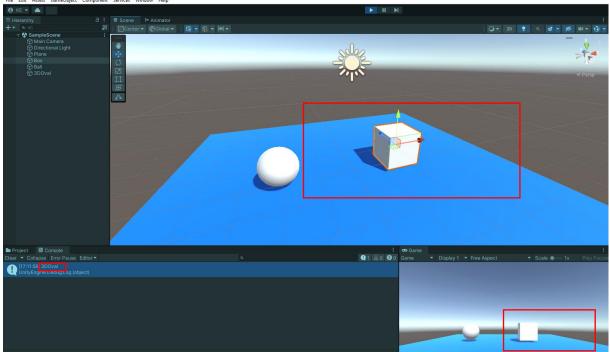


If we can reach the hit object, we can also perform operations on it; for example, we can make it invisible.

#### collision.gameObject.GetComponent<MeshRenderer>().enabled = false;

In this command, the **gameObject** to which the collision variable defined in parentheses is bound is made inactive and becomes invisible.



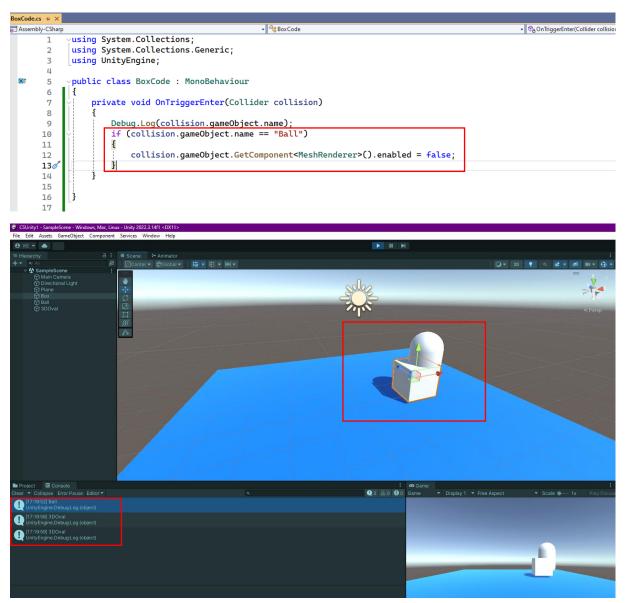


As a result, we understand through a message that it touches the capsule, and we can see that the **MeshRenderer** part of the capsule named **3DOval** is invisible because it is **unchecked**.

If we want to make an object with a certain name invisible, we need to make a conditional statement. Do this application on the **sphere** named **Ball**.

```
if(collision.gameObject.name=="Ball")
{
  collision.gameObject.GetComponent<MeshRenderer>().enabled = false;
}
```

#### Now, transfer it to a **C# script** file.



Here, when we touched the cube to our **Ball** object, it both gave a message and destroyed it. However, when we touched the cube to the capsule a few times, it only showed a message but did not destroy it.

## 4.13. Adding Gestures with Mouse and Keyboard

In the applications, it is a basic need for the user to enter data with the mouse and keyboard.

Considering the dynamics of the movement in the application process, it is understood that it should be used in the **Update()** method.

It is common practice to detect whether a **keystroke** has been made on the mouse or keyboard in Unity and to control it with **if** statements to react to it.

Frequently used **methods** for **keyboard** buttons:

Input.GetKey (*KeyCode.button*);

Input.GetKeyDown(KeyCode.button);

Input.GetKeyUp(KeyCode.button);

for Mouse buttons:

Input.GetMouseButton(#index);

Input.GetMouseButtonDown(#index);

Input.GetMouseButtonUp(#index);

GetKey and GetMouse: works continuously GetKeyDown and GetMouseDown: works once when pressed GetKeyUp and GetMouseUp: works once when pressed

#if the index value is **0**, then **left** mouse button #if the index value is **1** then **right** mouse button

We can write the first codes in our project.

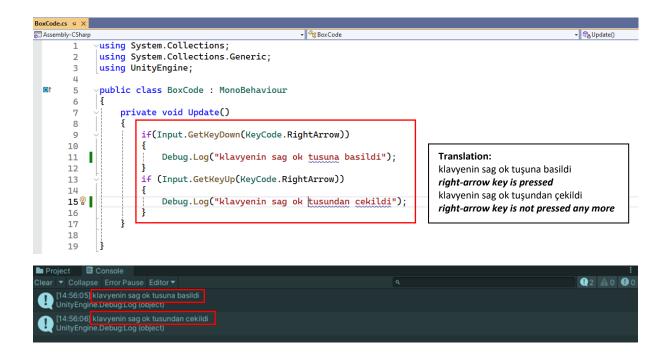
For example, when the **right arrow key** on the keyboard is pressed, we can send a message to the console to test that it is detected. For this,

if(Input.GetKeyDown(KeyCode.RightArrow)
 { Debug.Log(" klavyenin sag tusuna basildi "); }
if(Input.GetKeyUp(KeyCode.RightArrow)
 { Debug.Log(" klavyenin sag tuşundan cekildi "); }

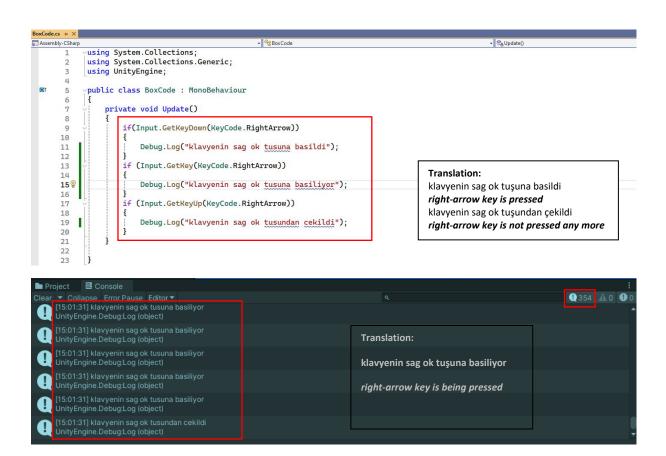
Translation:

klavyenin sag ok tuşuna basildi right-arrow key is pressed

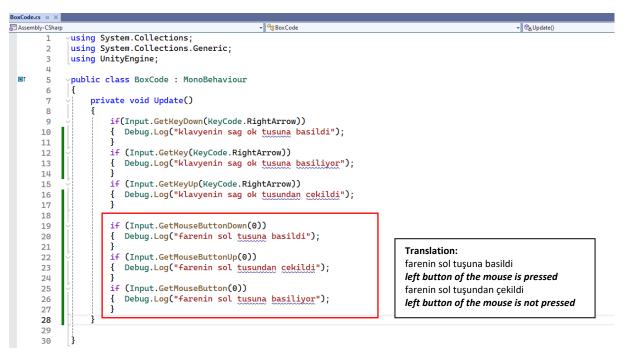
Now, on Visual Studio:



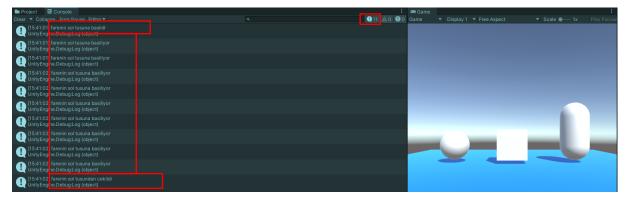
If we add a third **if** statement as **if(Input.GetKey(KeyCode.RightArrow))**, it will give the message **Debug.Log("keyboard right arrow key is being pressed");** (in English) if we press the **right arrow key** (here, the screen message is given 352 times).



Let's do a similar process for the **left mouse button**. The left button **code** was expressed as **0**. Add to our codes to declare the moment the left button is pressed and released and during the pressing process.

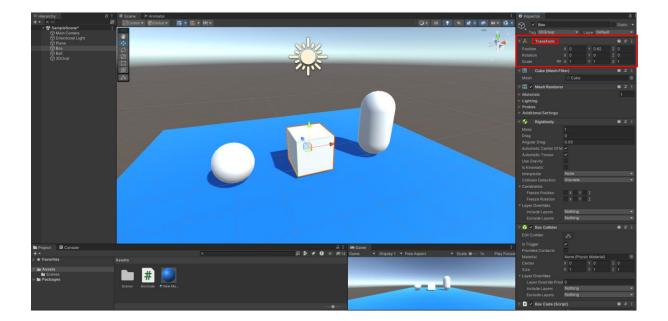


Even a short press and release gives a total of 11 messages.



# 4.14.Controlling and Moving Objects

It is possible to access all **Component** properties in the **Inspector** section of an object in the scene with C# codes. **Inspector>Transform** properties contain the most basic parameters considered in components. These are **Position (x, y, x coordinates), Rotation (axial rotation angles in x, y, z directions)** and **Scale (scale values in x, y, z directions)**.



To make changes in this area, the **transform.Translate()** function will be used. In addition, by accessing the **Rigidbody** component, movement control can be provided with the **velocity()** and **AddForce()** commands.

Movement according to the object's coordinates,

#### transform.Translate(x,y,z);

or according to the coordinates of the scene,

#### transform.Translate(x,y,z,Space.World);

we can do it with command templates.

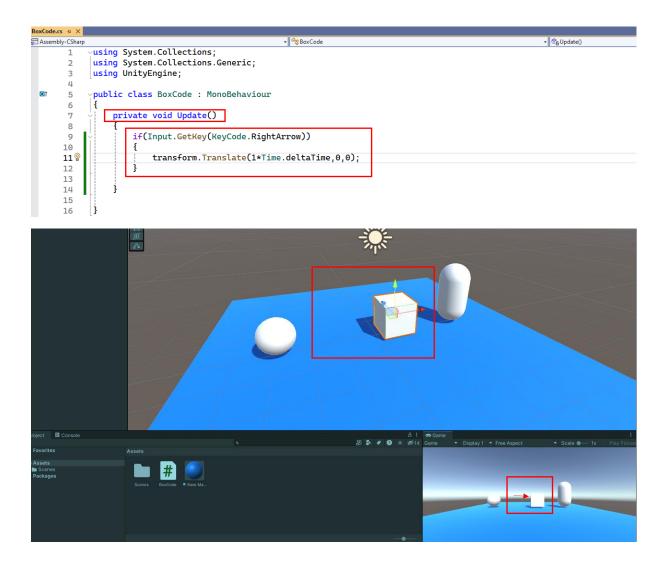
For example,

with the transform.Translate(1,0,0); command, 1 unit forward movement is provided on the X axis.

When we write this command in the **private void Update()** method, the movement occurs in the **X** direction if we **press** the key. However, the movement speed may differ depending on the configuration of the computers. To solve this problem and to ensure that the movement speed is equal on each computer, the axial movement unit is multiplied by the **Time.deltaTime** time value.

#### transform.Translate(1\*Time.deltaTime,0,0);

Let's apply this information to the codes connected to the object named **Box** in our project. If we press the **right arrow key** on the **Play** Mode Game screen, we will see that the **Box** object moves **forward** on its own **X**-axis.



For an upward movement, it is possible to write as

#### transform.Translate(0,1\*Time.deltaTime,0);

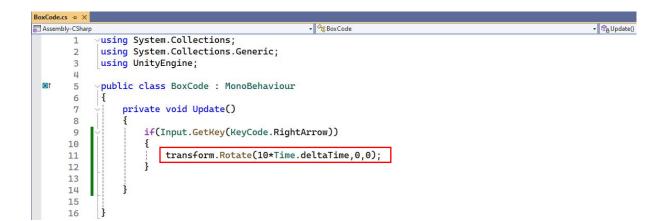
for a **downward** movement, it is possible to write as

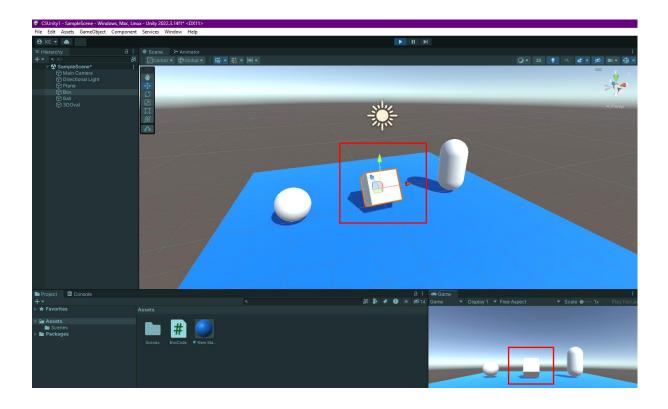
#### transform.Translate(0,-1\*Time.deltaTime,0);

If we want to **rotate** the object, we must access the **Inspector>Transform>Rotation** parameter and use the relevant command. Similar to the **Position** control, we can use the **transform.Rotate(x,y,z)**; and **transform.Rotate(x,y,z,Space.World)**; template. For example, if we want it to **rotate** as long as we press the **right arrow key** on the X-axis, we can write

#### transform.Rotate(1\*Time.deltaTime,0,0);

However, since **1\*Time.deltaTime** will **rotate** very slowly, a value of **10\*Time.deltaTime** or higher can be entered.





For the **Scale** operation, it will be necessary to write a different code than the other command lines. We can use an expression of the **Vector3** variable type and enlarge our **Box** object in all directions as long as the **right arrow key** is pressed.

#### transform.localScale += Vector3.one\*Time.deltaTime;

For scaling in certain directions, try the terms forward, up, down, back, left, and right instead of one in the expression Vector3.one\*Time.deltaTime. We can set up the equation with the - sign to shrink.

transform.localScale -= Vector3.one\*Time.deltaTime;

## 4.15. Interacting and Animating with Rigidbody

**Rigidbody** is a **physics component** element. Let's continue our topic on **Box**, which is the object that we connected to C# Script in the project. We added **Inspector>Add Component>Rigidbody** element to our **Cube** object named **Box**. We will need to access this element (**Component**) with codes and add **GetComponent<Rigidbody>()** and velocity function to animate it by giving it a speed value. It is possible to set its speed with a multiplier value and **Time.deltaTime**.

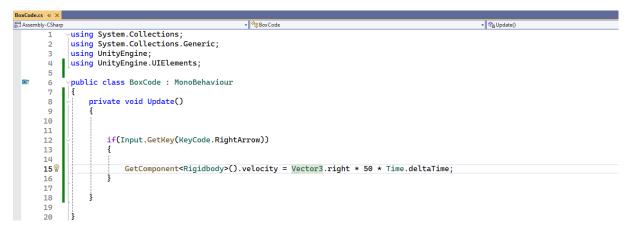
#### GetComponent<Rigidbody>().velocity=Vector3.right\*50\*Time.deltaTime;

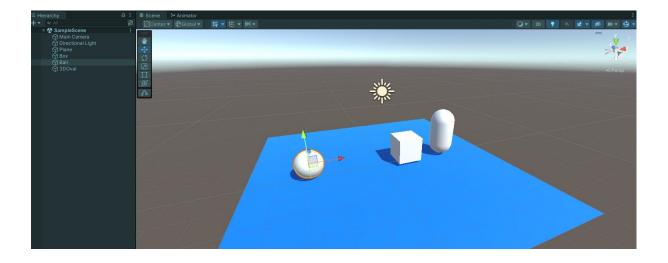
In order for this expression to take a **3D vector** value with the term it is equal to, in the scene,

Vector3.right // 1 unit right on the X axis Vector3.left // 1 unit left on the X axis Vector3.up // 1 unit up on the Y axis Vector3.down // 1 unit down on the Y axis Vector3.forward // 1 unit forward on the Z axis Vector3.back // 1 unit back on the Z axis

on its axis; transform.right // X axis 1 unit right transform.left // X axis 1 unit left transform.up // Y axis 1 unit up transform.down // Y axis 1 unit down transform.forward // Z axis 1 unit forward transform.back // Z axis 1 unit back

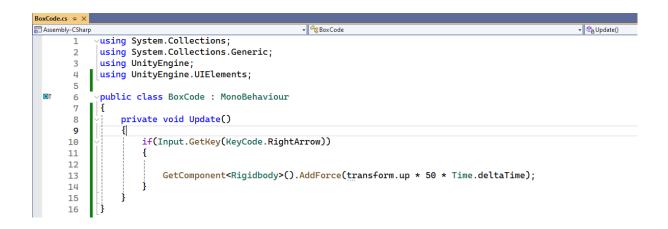
When we apply it to the script file, we can see the movement of the **Box** object when we press the right arrow key. Here, we see that once we start the movement, it continues at a constant speed.

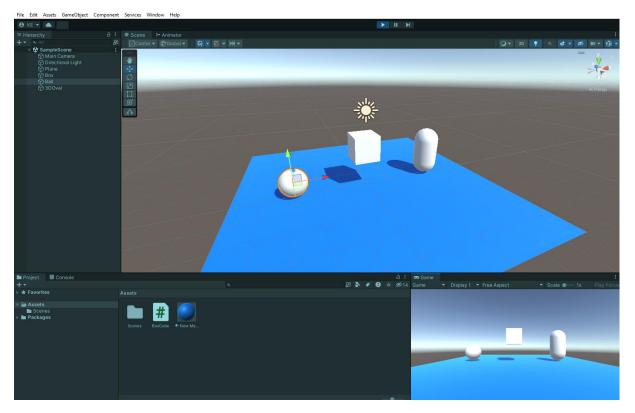




AddForce articulation can also be used to move with force application. Let's structure the command to move the **Box** object up,

#### GetComponent<Rigidbody>().AddForce(transform.up\*50\*Time.deltaTime);

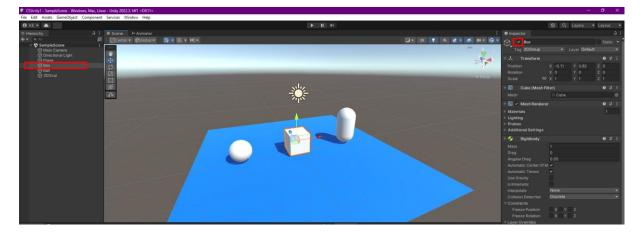




As long as we press the right arrow key, we see a movement that gradually accelerates and speeds up.

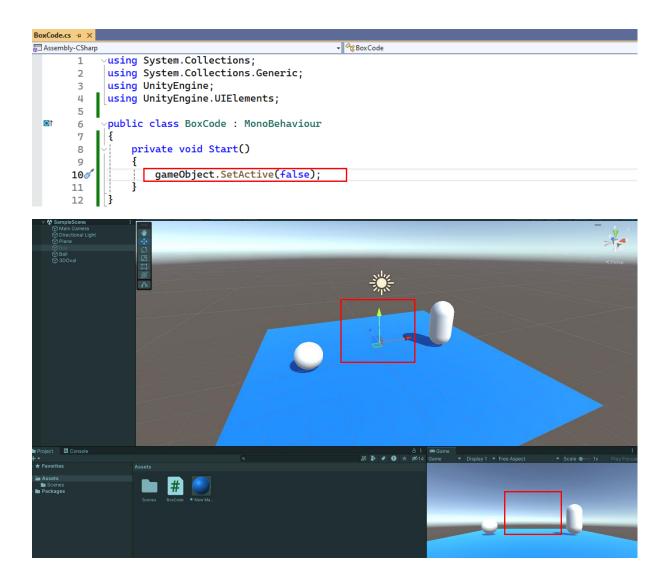
# 4.16.Activating/Deactivating Objects (SetActive) and Destroying Objects (Destroy)

We have seen that elements of objects such as **Inspector>MeshRenderer** can be active or passive. In this section, let's see that the object itself can be made **active** or **passive**. An object can be made completely active or passive by ticking the **check box** on the line with its name in the **Inspector** and making it checked/unchecked. This is not a **Component**/element but the object itself.



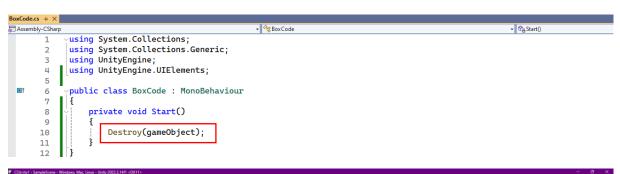
When this box is deselected, the object is not deleted, but the object's visibility and all its components become **inactive.SetActive()** is the control command used for this purpose.

The general form for making an object inactive is **gameObject.SetActive(false)**; In the **Play** Mode Game window, the **Box** object will be **inactive** and **invisible**.



In the case of **SetActive(true)**, its visibility and all elements will be turned on.

**Destroy()** command is used to delete the object. Its general form is **Destroy(object, time)**. If we write **Destroy(gameObject)**; in the script file, the object to which the C# file is attached will be understood as **gameObject** and the **Box** will be completely deleted along with the **Hierarchy** and **Inspector** sections in **Play** Mode.



File Edit Assets GameObject Compone		
ӨКЕ- 🔺		
≡ Hierarchy a		
+	Ø Center = @0xbal = □ + 0, = M +	
Trees SampleScare		
Project Console	ài œCame i	
+-	🐁 👔 🌢 🗸 🔀 🕸 90 🕸 9514 Game 👻 Display 1 * Free Aspect 🔹 Scale 🐠 - 1x Play Focus	
🗉 🖈 Favorites	Assets	
* ∰ Assts Mascons • ∰ Packages		

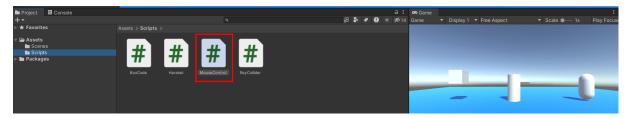
#### Destroy(gameObject, 3.0f);

If we write as such, the deletion will take place after 3 seconds.

## 4.17.Object Control with Mouse

It is possible to hold and drag objects onto the scene with the **mouse**. One of the most practical methods for this is to write code for the objects to be controlled with the mouse.

Let's create a C# Script named MouseControl.cs under Assets>Scripts and open it in Visual Studio.



For the application, we can use the **OnMouseDrag()** method in the Unity library. For this, we will calculate two **Vector3**-type variables. One will carry values for the coordinates of the mouse, and the other will carry values for the coordinates of the object.

We will also define a float-type variable for the third coordinate assignment. **Input.mousePosition.x** and **Input.mouse.Position.y** parameters can be used to transfer the **x**, **y** and **z** information of the

mouse to the **Vector3** type variable that determines the mouse position. We can enter a fixed value for the **z** coordinate.

Vector3 mousePosition = new Vector3(Input.mousePosition.x, Input.mousePosition.y, posZ); Vector3 objPosition=Camera.main.ScreenToWorldPoint(mousePosition);

Here, the vector variable **mousePosition** is defined, and the mouse position coordinates are assigned.

The vector **objPosition** converts the **mousePosition** information from the screen to the **WorldPoint** setting.

The screen is actually a 2-dimensional plane, and different perspective calculation methods can be used for the third dimension. While there are short explanations in the codes within the scope of the course, various written and visual details about their mathematical, geometric and trigonometric infrastructure can be found on the web.

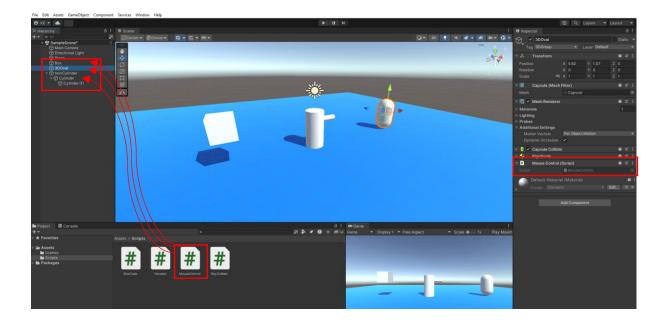
Finally, the transformation of the object to which these codes will be connected will be made.

transform.position = objPosition;

Let's code in Visual Studio.



We can connect the file we saved to all our objects (*Box, 3DOval and Cylinder*) in the scene.

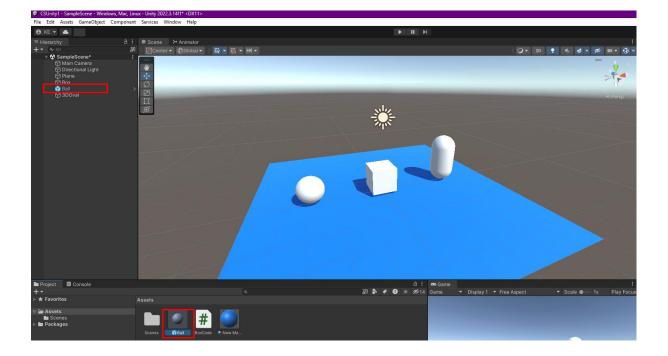


In Play mode, we can observe that we can hold all three objects with our mouse and drag them to the desired direction and height and that those with physical materials and jumping features move in this way.

# 4.18. Cloning Objects – Prefab and Instantiate

We may want to produce and duplicate objects with codes while the program is running. To do this, we need to create a prefab copy of that object in the **Assets** section in the **Hierarchy**. This form is called prefab.

Let's drag our **Sphere** object named **Ball** on the scene to the Assets section. See that the icon of the **Ball** object changes, and a copy is created in the Assets section. This copy is a **prefab Ball** asset.



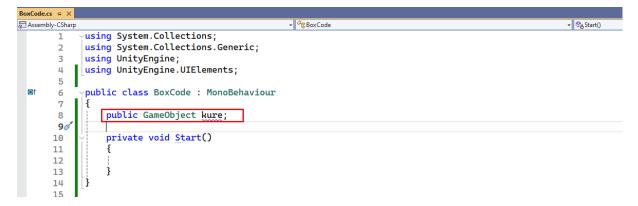
If we drag the **Ball** asset into the Assets section to the scene over and over again, **Balls** with the same properties will be created. Changing the property of one of them will affect all of them.

After the **Ball prefab** is obtained, we can delete the Ball object in the scene. Even if this object is deleted from the **Hierachy**, we can drag as many clones as we want from Assets and add them to the scene.

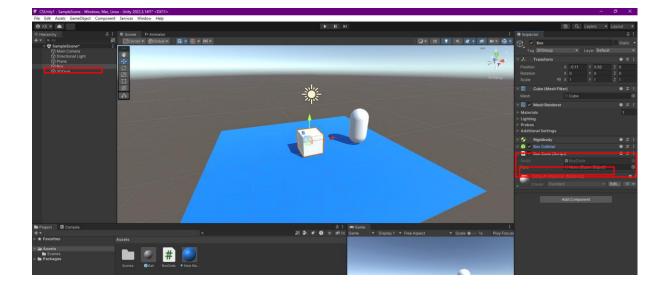
## 4.19. Adding Objects with Instantiate Coding – Spawn

**Instantiate** command is used to **add**, **load** and **clone prefab assets** with codes in the scene. This is also called **spawn** (reproduction/egg laying).

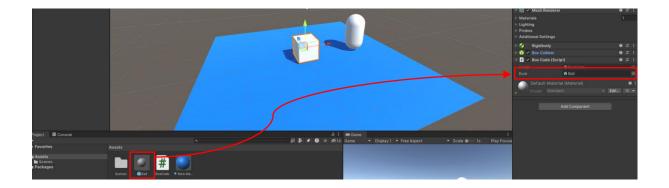
Let's make an edit according to our new title in **BoxCode.cs** file and define a **public GameObject** accessible in the editor. Here, the variable name is given as **kure**.



With this change, a variable field named **kure** has appeared in the **Inspector** section of the **Box** object because it is **public**. However, since a **prefab object** has not been connected yet, it is in the **None (Game Object)** state.



The Ball prefab in the Assets section is dragged and connected to this area.



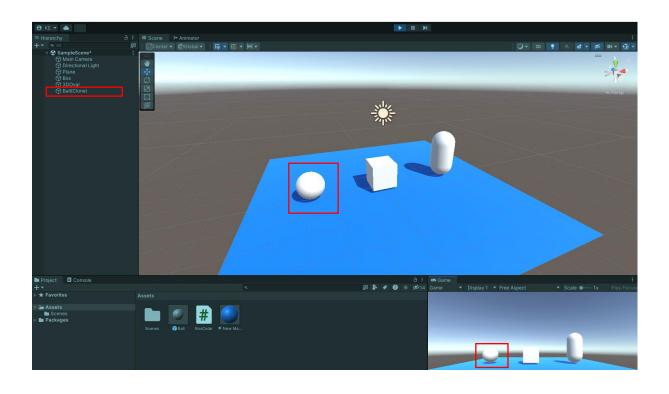
Now, we can use the **Instantiate()** command in the code lines.

The general form of this command is,

#### Instantiate (object, coordinate(x,y,z), angle(x,y,z))

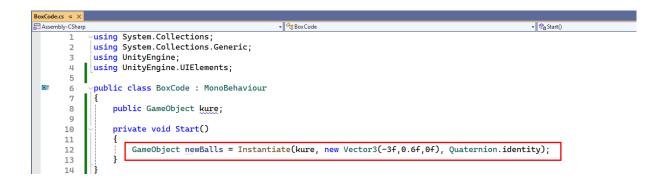
However, it only allows a prefab copy of the invisible **Ball** object that we deleted from the **Hierarchy** with **Instantiate(kure)** to be added to the scene. **kure** in the **Instantiate(kure)** command is the **kure** defined as public **GameObject** in the code file and connected to the box in the editor with the **Ball prefab**.

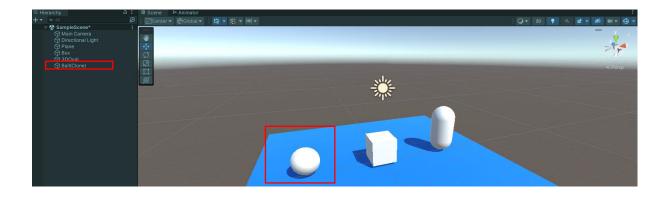
BoxCode.c	s ⇒ ×	
🔓 Assemb	ly-CShar	• • • • • • • • • • • • • • • • • • •
	1	<pre>vusing System.Collections;</pre>
	2	using System.Collections.Generic;
	3	using UnityEngine;
	4	using UnityEngine.UIElements;
	5	
10	6	<pre>vpublic class BoxCode : MonoBehaviour</pre>
	7	
	8	public GameObject kure;
	9	
	10	private void Start()
	11	
	12 🖗	GameObject newBalls = Instantiate(kure);
	13	
	14	



A **clone** of the **Ball** object is added to the scene with **Play** mode. This object appears in the **Hierarchy** with the name **Ball(Clone)**.

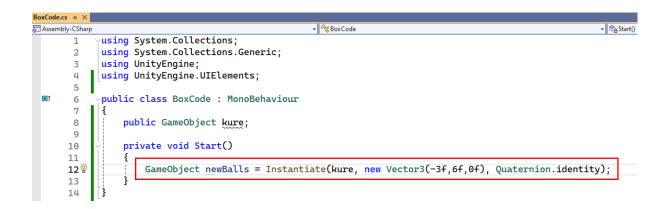
Now, let's write the **Instantiate** command in accordance with the wide format. Here, **kure** object (i.e. **Ball prefab**) to **Box**, the identity function (i.e. its original rotation values) connected to the **Querternian** command used to determine the **position** and **angle** we want to appear; **Quaternian.identity**.

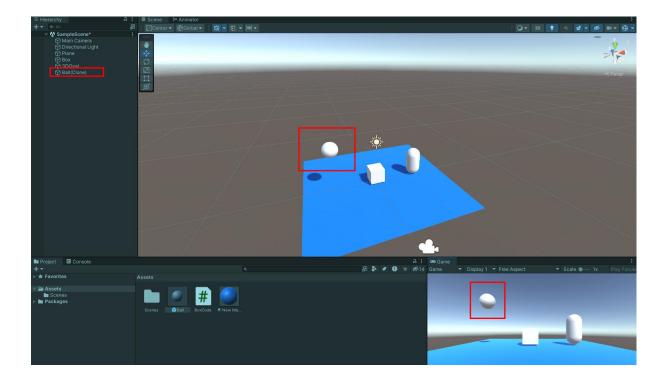




We have obtained a similar image to the previous one. However, our definition and parametric modification capabilities have increased.

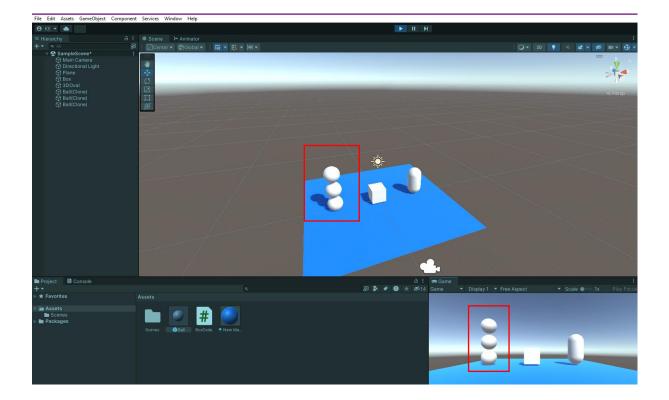
Now, specify the coordinate where the sphere will appear as (-3f, 6f, 0f). The **Ball** that appears above will fall due to the **Gravity** effect in the **Rigidbody**.



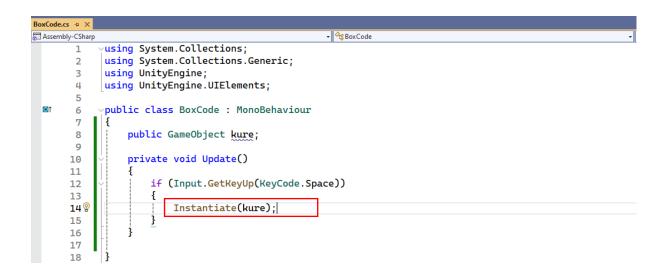


Let's make changes to the code to clone 3 Balls. For this purpose, let's create a **for** loop and make them fall. With the expression **new Vector3(-3f, 4f+i,0f)** in the loop, each **clone kure** is made to appear four units higher in the **Y** direction (**vertical axis**) by the value of the **i** variable. In this way, the spheres (**kure**) are prevented from appearing at the same point.

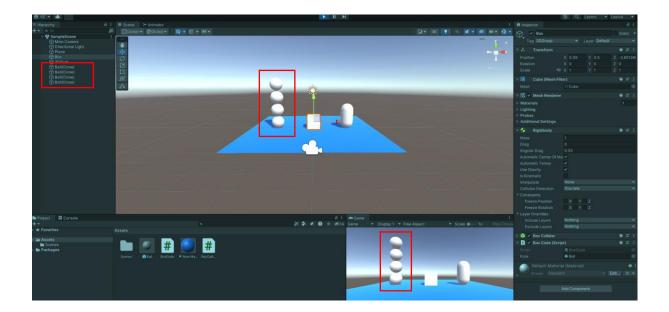
BoxCode.	cs ⊕ ×	
Assem	bly-CSharp	p - Assard
	bly-CSharp 1 2 3 4 5 6 7 8 9 10 11 12 13 14	
	15 16 17	



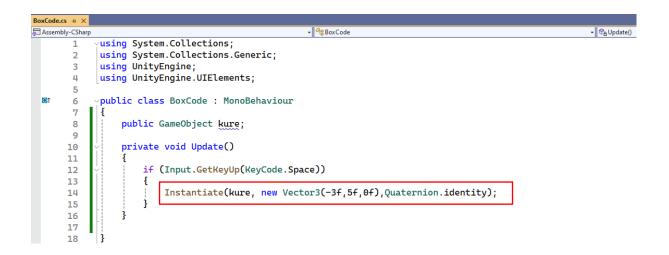
Another scenario is that when any key is pressed the **Ball prefab** object is **added** to the scene.



In its simplest form, as a result of these codes, each time the **Space Bar** is pressed, the **Ball prefab** is added to the same coordinate. However, since it is a **Rigidbody**, they push each other upwards. In the example run, the space bar was pressed four times.

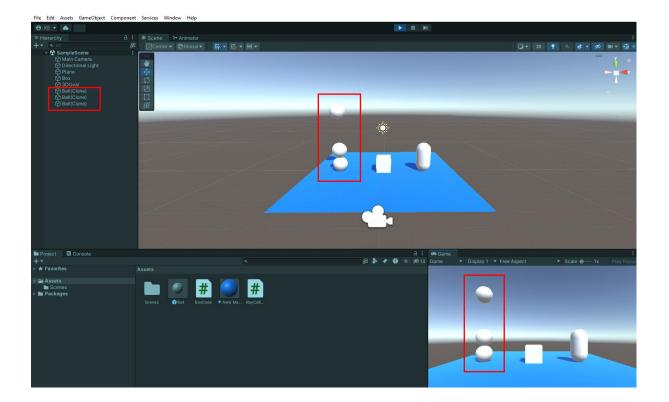


However, if we want the **Ball prefabs** to fall from the top instead of pushing them from the bottom to the top by keystrokes, we need to enter **new Vector3 (x,y,z)** coordinates of the point we want them to fall from.

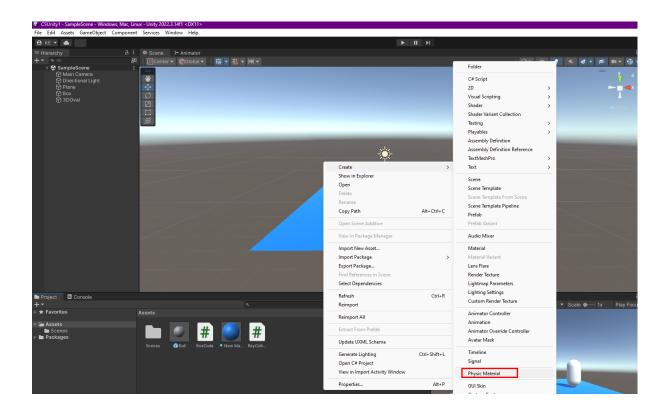


When we run the application, **Ball prefabs** start falling from the coordinate (-3f,5f,0f) depending on the **Space Bar** keystroke. Also, notice that a **Ball(Clone)** is added to the **Hierarchy** with each keystroke.

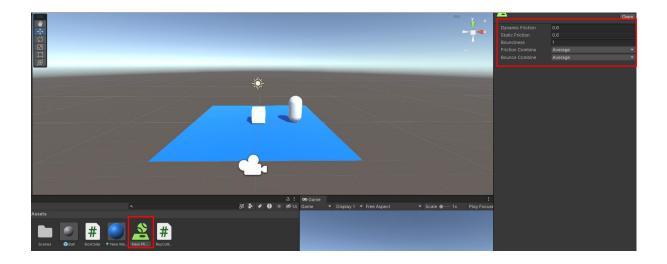
Since there is no physical jumping feature in the **Ball prefab**, the falling spheres (**kure**) pile up on top of each other.



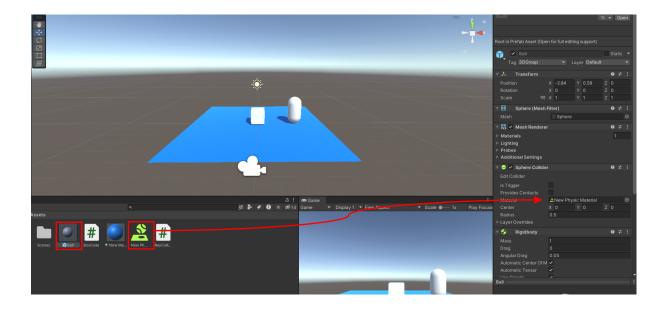
For the **Ball prefab**, first add a **Physic Material** to the Assets section.



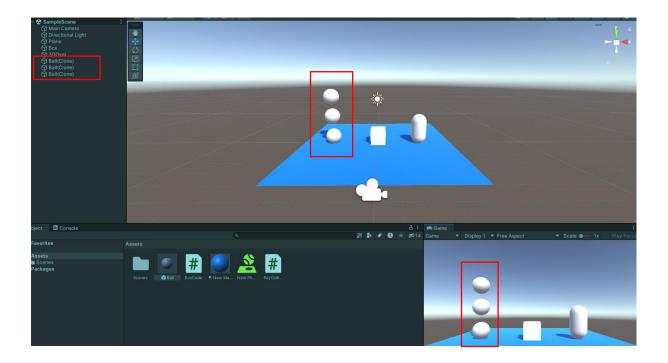
Then, adjust its **bounciness** settings.



Now, drag and connect the **Physic Material** to the **Material** field of the **Ball prefab**.



Run it in **Play** mode again. We can see that the balls bounce under the effect of the physics material, and after a while, they move left and right and fall from the **Plane**.

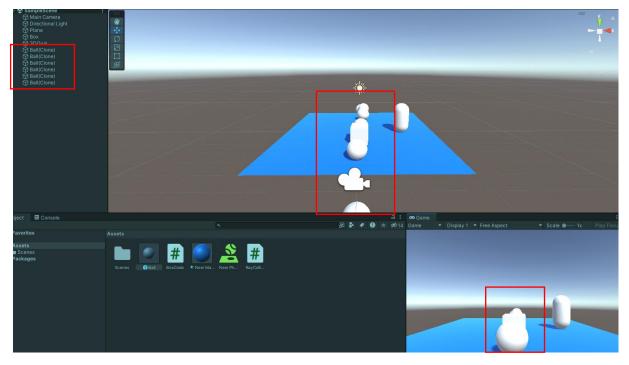


#### Alternatively, if we write

#### Instantiate(kure, new Vector3(-3f,5f,0f),Quaternion.identity); instead of

#### Instantiate(kure, transform.position, transform.rotation);

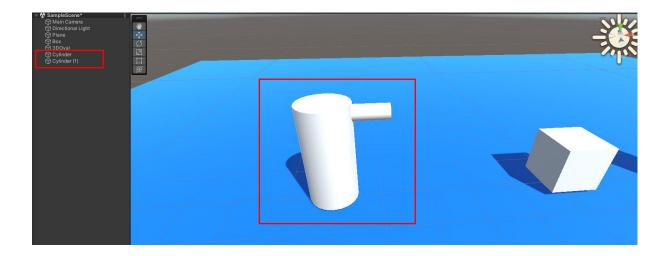
every time we press the key, the code will appear in the **Transform position** and **rotation** of the **sphere** we are connected to. However, this will collide with the present sphere(s). Instead, we can open an empty game object and connect the **code** file to it.



#### Using Lerp and eulerAgles in Movements

**Lerp** literally means linear **interpolation**. It is generally used in Unity C# coding to provide accelerated movement. **Euler** is the **angular exponent value** used in **trigonometry**, which forms the basis of **natural logarithms**. It is used to define objects in terms of angles.

Let's create two cylinders, one large and one small, in our scene for the project that will rotate a **GameObject** horizontally by **0**, **90**, **180** and **270** degrees with keystrokes. Place the smaller cylinder on top of the larger one, as shown in the figure.



Now, create an **empty GameObject** object with **Hierarchy>Create>Empty**. This can be named **twinCylinder**. Drag two cylinders to this object and create a **parent-child** relationship. Let's make sure that all objects are in the **center**. If they are not, move them to the center (**0**, **0**, **0**) position with **Transform>Reset**.

≔ Hierarchy	9:	# Scene > Animator
+ ▼ (• All	因	🛛 🗹 Center 🔻 🌐 Global 🔻 📗 🐺 🔽 🖽 💌 🗮 💌
🔻 🔂 SampleScene*		
🔂 Main Camera		
💬 Directional Light		
💬 Plane		
Box		\$5
G 3DOval		
v 🕞 twinCylinder		
Cylinder		
😭 Cylinder (1)		

Add a **new C# Script** file to the Assets section. Drag the file named **Hareket.cs** to the **twinCylinder** empty object and connect it.



Since it will be a **3D vector** movement, define a variable by resetting its initial value.

#### Vector3 vec = Vector3.zero;

As there will be rotation depending on the key, continuity is required. Therefore, write the codes in the **Update()** method.

Since the rotation will be in the **Inspector>Transform** area, we can check/update the **vector direction/angle** by assigning **Vector3.Lerp** to the **eulerAngles** function as follows.

#### transform.eulerAngles = Vector3.Lerp (transform.eulerAngles, vec, 0.1f);

Here, it has a template like Vector3.Lerp(from where, to where, in how much time). Because there is transform.eulerAngles on both the left and right of the equation, it is understood that there will be a change depending on the vec variable (y-axis) and not the value of this point. The speed of this update is in float time type, and 0.1f is assigned here.

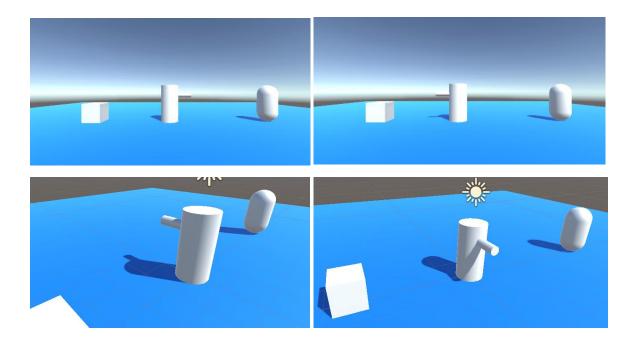
Now, specify what will happen when the **W**, **A**, **S**, and **D** keys are pressed and assign a value to our vector variable **vec**.

if (Input.GetKeyDown(KeyCode.W))
{ vec = new Vector3 (0f, 0f, 0f); }
if (Input.GetKeyDown(KeyCode.D))
{ vec = new Vector3 (0f, 90f, 0f); }
if (Input.GetKeyDown(KeyCode.S))
{ vec = new Vector3 (0f, 180f, 0f); }
if (Input.GetKeyDown(KeyCode.A))
{ vec = new Vector3 (0f, 270f, 0f); }

The value of the **vec** variable is constantly updated in the **Update()** method. If a key is pressed, it is returned to its new position.

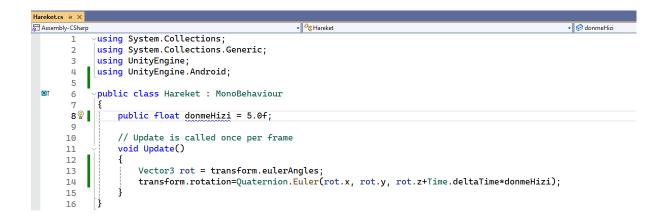
embly-CSharp	•	Ag Hareket	✓ CB Update()	
1	<pre>vusing System.Collections;</pre>			
2	using System.Collections.Generic;			
3	using UnityEngine;			
4				
5	ypublic class Hareket : MonoBehaviour			
6	{			
7	Vector3 vec = Vector3.zero;			
8				
9	<pre>// Update is called once per frame</pre>			
10	<pre>void Update()</pre>			
11	{			
12	<pre>transform.eulerAngles = Vector3.</pre>	<pre>Lerp(transform.eulerAngles, vec, 0.1+</pre>	F);	
13				
14	if(Input.GetKeyDown(KeyCode.W))			
15				
16	vec =new Vector3(0f, 0f, 0f)	;		
17	<pre>if (Input.GetKeyDown(KeyCode.D))</pre>			
18	1+ (Input.GetReyDown(ReyCode.D))			
19 20	vec = new Vector3(0f, 90f, 0	£),		
20	vec - new vectors(or, sor, or	+J,		
22	if (Input.GetKeyDown(KeyCode.S))			
23	{			
24	vec = new Vector3(0f, 180f, 0	0f):		
25				
26	if (Input.GetKeyDown(KeyCode.S))			
27				
280	vec = new Vector3(0f, 270f, 0	0f);		
29	}			
30	3			
31	1			

Since the **code** file is connected to the parent **twinCyliner**, both cylinders in the child state will move simultaneously.

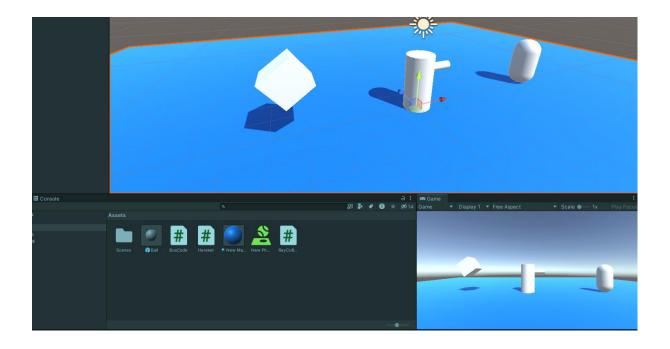


The main purpose of using the Vector3.Lerp function is to soften the sharpness of movements and provide a certain smoothness in turns.

Now, assign an **alternative** rotation **script** to our **Box** object and try this on **Box**. **Position** it up and remove **gravity**. Reorganize the **Motion.cs** code as follows.



Let's define the variable **donmeHizi**, which is a **real** and **public number**, and assign it a value like 5.0f. Define the **rot** variable of type **Vector3** in the **Update** method and assign the **transform** information with the **eulerAngles** function.



In **Play** mode, it will be seen that the **Box** object **rotates** at the specified speed without any keystroke.

As the speed parameter is **public**, the editor can also change it.

# 4.20.Raycast – Collision Control by Ray Spreading

In Unity, **rays** that are **not visible** to the user can be emitted between **objects**, and collision control can be provided.

Previously, **Collider** and **Rigidbody** physics elements were required for two objects to **collide**. In **Raycast**-type collisions, both objects need to have a **Collider**, but a **Rigidbody** is not required. However, there is no harm in having it.

Ray propagation can generally be in two ways:

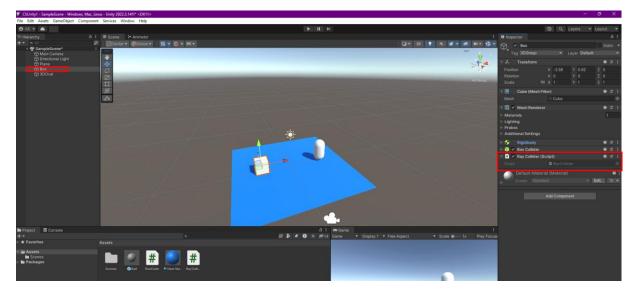
i- From Object

ii- From Camera

iii- From Mouse

### i- Ray propagation from Object

Let's come to the Unity project since how it is used in the application will be better understood. We had two objects named **Box (Cube)** and **3DOval (Capsule)** in the scene. Now, create a **C# Script** named **RayCollider.cs** to connect to the **Box** object.



Create a variable defined as **RaycastHit** in the script file that carries the information about the object that the ray (**Ray**) hits.

### RaycastHit obj;

Hence, the **obj** variable has the feature of carrying some information about the object that the **ray hits**.

Since the ray will be sent for continuous control purposes, write our codes in the **Update()** method.

If we consider the **ray** as a **physics element**, the code **formula/template** for sending the ray can be written as follows.

### Physics.Rascast(start\_position, ray\_direction, out variable, ray\_length)

We can adapt it for our project as follows,

### Physics.Rascast(transform.position, transform.right, out obj, 5.0f)

Here,

with transform.position, we determine the position of the object (Box) to which these codes are connected as the starting point. transform.right will be sent to the right side (X-axis) of the object (Box) (if it were Vector3, it would be the X-axis according to the scene reference). With out obj, the information about the object that the ray hits is assigned to obj, a variable of type RascastHit, as an output. The maximum ray length (max\_length) is written as a unit distance in float type.

If we put this code expression in the **if** form, we can determine whether the **ray hit** an **object**. If this **collision** has occurred, it can be tested with a separate **if** statement whether it is the **targeted object**. If it **hits** the **right object**, we can **declare** this to the **console** with a **message**. The coded version of this setup is given below.

### if (Physics.Raycast(transform.position, transform.right, out obj, 3.0f)

```
{ if (obj.collider.gameObject.tag== "3DGroup")
```

{ Debug.Log(" Ray collided 3DGroup "); }

}

In the second **if** statement, **obj**, which carries the information about the object that the ray **hits** and the **tag** name of the **gameObject** that it collides with (**collider**), **3DGroup**, gives a message with **Debug.Log**.



CSUnity1 - SampleScene - Windows, M				
File Edit Assets GameObject Com	nponent Services Window Help			
😝 KE 🔻 📥				
	A : # Scene ≻ Animator			
+ ▼ Q+ All ▼ ✿ SampleScene*	₽ Denter ▼ @Global ▼	装 - 協 - 王		
<ul> <li></li></ul>				Persp.
	8			
	Đ			
Project 🗟 Console			I 😎 Game	
Clear Collasse Error Pause Editor     Constant State Constant State     Constate     Constant State     Constant     Constant State     Const			❶ 188 ▲ 0 ● 0 Game ▼ Display 1 ▼ Fre	e Aspect 🔹 Scale 🔶 1x Play Focus
[13:46:14] Ray collided 3DGroup UnityEngine.Debug:Log (object) [13:46:14] Ray collided 3DGroup UnityEngine.Debug:Log (object)				
[13:46:14] Ray collided 3DGroup UnityEngine.Debug:Log (object)				
[13:46:14] Ray collided 3DGroup UnityEngine.Debug:Log (object)				

When we run it in **Play** Mode, we pull the **cube** (**Box**) to the **right**. When we reach the **maximum length** of 3.0f, **Debug.Log()** command runs continuously. When we zoom out, the messages are **cut off**.

Alternatively, instead of **transform.right** in the code, we can specify a direction we want with **new Vector3(X, Y, Z)** coordinates [**new Vector3(1,0,0)** for the right direction (X)].

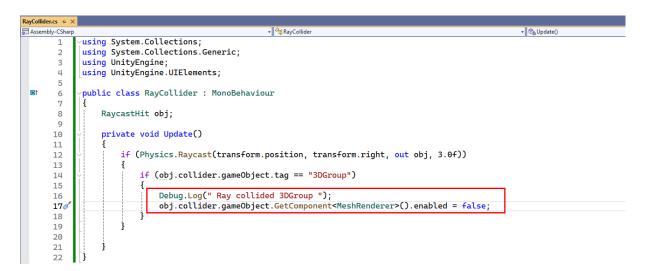
Another alternative is to use a **name** instead of a **tag**.

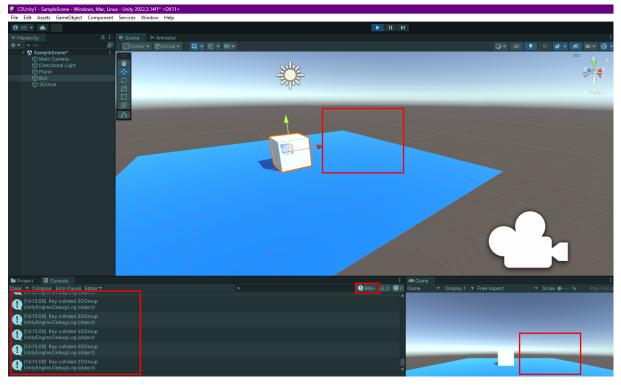
if (obj.collider.gameObject.name== "3DOval")

Project 🗧 Console		😎 Game		:
Clear V Collapse Error Pause Editor V	🔮 999+ 🔬 0 🔮 0	Game •	<ul> <li>Display 1 </li> <li>Free Aspect</li> </ul>	<ul> <li>Scale</li></ul>
(13:43:24) Ray collided 3DOval UnityEngine.Debug:Log (object)				
[13:43:24] Ray collided 3DOval UnityEngine.Debug:Log (object)				
[13:43:24] Ray collided 3DOval UnityEngine.Debug:Log (object)				
[13:43:24] Ray collided 3DOval UnityEngine.Debug:Log (object)		_		
[13:43:24] Ray collided 3DOval UnityEngine.Debug:Log (object)				

After checking that it works with the message when the **ray hits** the object we specified, we can access its **Mesh Renderer** property and make it **invisible**.

### obj.collider.gameObject.GetComponent<MeshRenderer>().enabled=false;





When the capsule enters the **ray range**, it becomes **invisible**. However, since it is not deleted, the **Debug.Log** command continues to work.

Alternatively, if you use

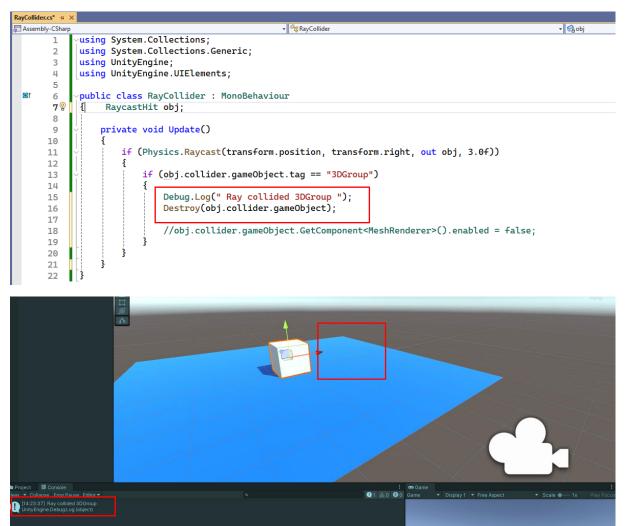
obj.collider.gameObject.SetActive(false);

The **capsule** becomes **invisible**, and the **collision** stops.

We can also run it by deleting the object.

### Destroy(obj.collider.gameObject);

With this operation, the capsule named **3DOval** is **deleted**, **all** its **components** are **closed**, and the **collision stops**.



# 4.21.Ray Emission from Camera

To emit a **Ray** (ray) from the **camera**, first, a **ray** is **created**, and for this, a **Ray**-type **variable** must be defined. This variable should find the **camera object** to be used by **name**, reach its **Camera Component field**, and **emit** a **ray** from the point whose coordinate is determined with the **ViewportPointToRay** (screen coordinate) command. Now, let's apply this information to a variable named **Ray** type **isinYay**.

Ray isinYay = GameObject.Find("Main Camera").GetComponent<Camera>().ViewportPointToRay(new Vector3(0.5f,0.5f,0f));

In this section, which can take values between 0 and 1, we mean the **center** point of the **camera** with the coordinate (0.5f, 0.5f, 0). While this variable is taking a value, if the **physical ray** produced **hits** an

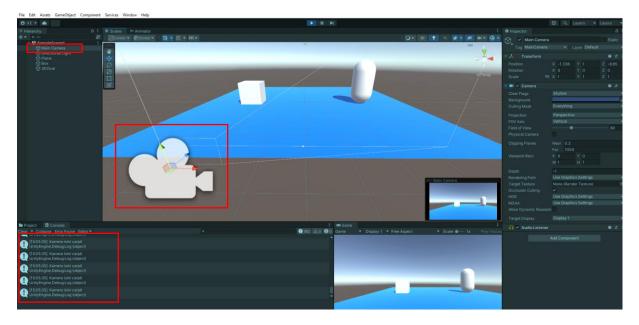
**object**, transfer its information to our **RastcastHit** type variable **obj**. The following code with a **bool** type **return** can be used to test this.

### Physics.Raycast(isinYay, out obj)

To see if this **ray hits** the **object** defined by **tag** or **name**, send a message to the **console** with the **Debug.Log** command.

RayCollide	er.cs ⊕ ×		
🔂 Assem	bly-CSharp		- 😪 Update()
	1	<pre>vusing System.Collections;</pre>	
	2	<pre>using System.Collections.Generic;</pre>	
	3	using Unity.VisualScripting;	
	4	using UnityEngine;	
	5	using UnityEngine.UIElements;	
	6		
10	7	<pre>ypublic class RayCollider : MonoBehaviour</pre>	
	8	{	
	9	RaycastHit obj;	
	10		
	11		
	12	<pre>&gt; private void Update()</pre>	
	13	{	
	14	Ray isinVay = GameObject Find("Main Camera") GetComponent	<camera>().ViewportPointToRay(new Vector3(0.5f, 0.5f, 0f));</camera>
	15	hay isinay - dameobjecth inde hain camera y.deccomponene	cameras (). viewporci orneronaly (new veccors(o.or, o.or, or)),
	16	<pre>if(Physics.Raycast(isinYay, out obj))</pre>	
	17	s s	
	188	if (obj.collider.gameObject.tag == "3DGroup")	
	19	s	
		<pre>Debug.Log(" Kamera isini carpti ");</pre>	
	20	Debug.Log(" kamera isini carpti ");	// Kamera isini carpti: camera-ray hit
	21	3	
	22	3	
	23	3	
	24 25	[}	

When we switch to **Play** Mode, this time we need to move the camera **left** and **right** since it is a **ray emitter**. When the **center** point of the camera (0.5f,0.5f,.0f) aligns with the objects whose **tag** information is **3dGroup**, that is, both the cube and the capsule, we can see the **Debug.Log** message reflected in the **console**.



Ray is frequently used in FPS (first-person shooter) type applications.

We have seen that **Ray** propagation from objects and cameras, and when it hits, it can give a message and change the **Mesh Renderer** settings of the object it hits or its properties.

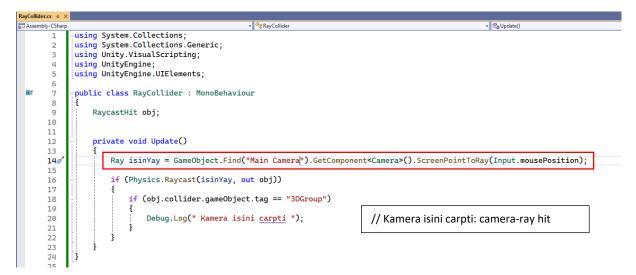
# 4.22.Ray Propagation from Mouse

Another type of **Ray propagation** is done via **Mouse**. The only **difference** from the previous camera application is the addition of the **ScreenPointToRay** function instead of the **ViewportPointToRay**. Here, using the **mouse position** is a common application, and this can be provided with **Input.mousePosition**.

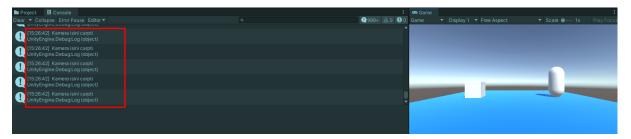
The following command line can be written for the **Ray** type variable to get a value and emit a ray.

Ray isinYay = GameObject.Find("Main Camera").GetComponent<Camera>().ScreenPointToRay(Input.mousePosition);

Let's apply it to the code without making any other changes to the program.



**Debug.Log** messages were successfully sent to the **console** when the **mouse** was **moved** and **hovered** over both **3DGroup** objects (**cube or capsule**). Messages were also stopped when the mouse was not over these two objects.

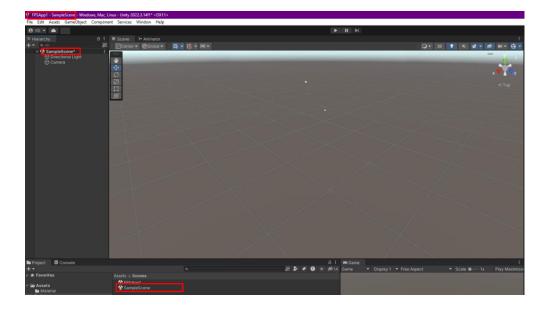


In **Physics.Raycast(isinYay, out obj)** expression, a **distance** definition can be added as **Physics.Raycast(isinYay, out obj,100f)**. When this is not the case, the ray length is **infinite**. It is preferable to set a distance in terms of performance and to maintain the computer processor. However, this distance should be determined with a couple of trials and its lower limit should be found.

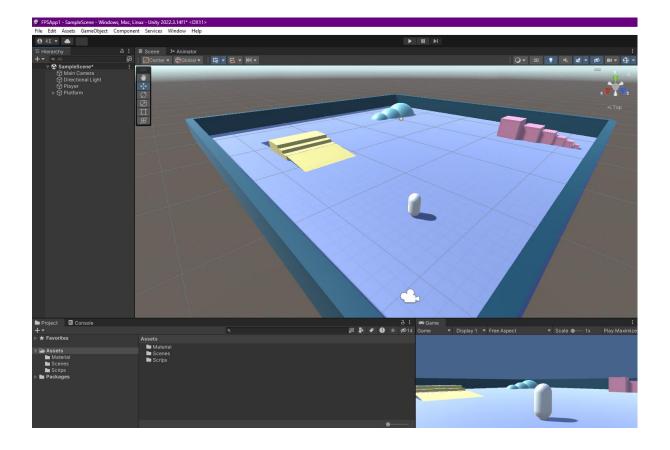
# 4.23.FPS/P (First Person Shooter/Perspective) Applications

One of the common applications is to move the scene from the eyes and perspective of the person moving in the **3D environment**. Depending on the scenario, various jobs and tasks can be performed during this movement. But first, this movement must be provided. There are **FPS assets** available for use by Unity Technologies or other manufacturers in the Asset Store. In this section, let's see how we can create our character within simple scene setups and provide movement with codes with a few alternative approaches.

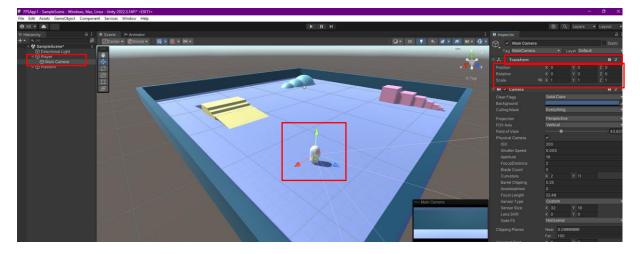
For the first approach, a project called **FPSApp1** was created for the application, the purpose of which is to control movements and camera with **keys** and a **mouse**.



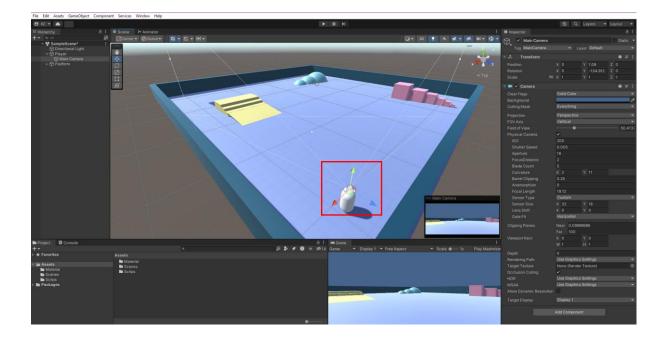
A platform for the application was established for the scene (**SampleScene**). A simple design was made for the platform with a **Plane, Cube**, and **Sphere**. A **Capsule** called **Player** was added.



Let's drag the **Main Camera** under the **Player** (**Capsule**) object and establish a **parent-child** relationship. Reset the camera **transform** properties so that it references the **capsule**.



Now, **position** the camera at the **top** of the **capsule**, which can be considered **head** level, where we loaded the **FPS** role. Adjust the angle of view.



After physically setting up the order, create and develop **two C# code files** required for the **capsule** (**Player**) in the **FPS** role and the **camera** representing our **eye** (*M.Akkuzu* approach was used here).

We created two files named PlayerControl.cs and CameraControl.cs in the Assets>Scripts folder.

**Screen geometry, trigonometric,** and **axial** coordinate calculation information is not included in the content of this document and can be found in various written and visual sources. Theoretical information is applied in this section, and short explanations are provided.

Now, we can write the codes in Visual Studio. Let's specify a **public float** type variable that will carry the speed of the **player** (capsule) and a variable for the **z-axis speed**.

Since the movements are continuous, we write the relevant codes in the **Update()** method. First, transfer the information from the **Input.GetAxis("Horizontal")** and **Input.GetAxis("Vertical")** ready functions to the **float** type **x** and **y** variables.

float x = Input.GetAxis("Horizontal"); float y = Input.GetAxis("Vertical");

These functions also **automatically** provide the use of the **arrow-direction keys** and the **A**, **S**, **D**, **and W** keys. This will **transfer** the **coordinate information** required for movement in the **horizontal plane**. However, the **depth** dimension is not included in this scope. For this, it will be necessary to assign a key (**vertical**) to a **float**-type variable to **ascend** and **descend**. For example, the **Q** key can be used for ascending and the **E** key for descending.

if (Input.GetKey(KeyCode.Q)) dikey = 1; if (Input.GetKey(KeyCode.E)) dikey = -1; Now, we can calculate the three parameters we will need in the **displacement** command **transform.Translate()**. The **x**, **y** and **dikey** (vertical) parameters are multiplied by the **velocity** variable defined as a **float** and **Time.deltaTime** for the time standard.

```
x *= Time.deltaTime * hiz;
y *= Time.deltaTime * hiz;
dikey *= Time.deltaTime * hiz;
```

Now, we can combine these three parameters into the transform.Translate() construct.

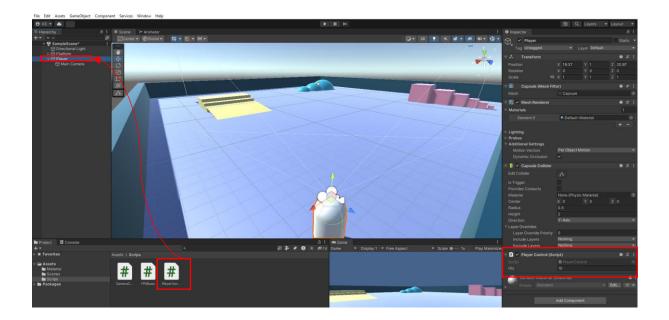
transform.Translate(x, dikey, y);

And coding in Visual Studio...

FPSBasic.cs	s	CameraControl.cs PlayerControl.cs + X
🕤 Assemb	ly-CSha	rp • 🔗 PlayerControl
	1	<pre>yusing System.Collections;</pre>
	2	<pre>using System.Collections.Generic;</pre>
	3	using UnityEngine;
	4	
10	5	<pre>ypublic class PlayerControl : MonoBehaviour</pre>
	6	{
	7	
	8	<pre>public float hiz = 10;</pre>
	9	float dikey = 0;
	10	
	11	void Update()
	12	Ę
	13	<pre>float x = Input.GetAxis("Horizontal");</pre>
	14	<pre>float y = Input.GetAxis("Vertical");</pre>
	15	<pre>if (Input.GetKey(KeyCode.Q)) dikey = 1;</pre>
	16	<pre>if (Input.GetKey(KeyCode.E)) dikey = -1;</pre>
	17	<pre>x *= Time.deltaTime * hiz;</pre>
	18	y *= Time.deltaTime * hiz;
	19	dikey *= Time.deltaTime * hiz;
	20	<pre>transform.Translate(x, dikey, y);</pre>
	21	}
	22	3

Then, drag and connect the PlayerControl.cs file to the Player object.

Do not forget that if the written codes are not connected to the relevant objects, they will **only** exist as a **non-functional** entity under Assets.



In case of adverse reaction to direction keys, **x** and **parameters** can be multiplied by - transform.**Translate(-x, dikey, -y);** 

Now, test in Play mode and see that movement is provided with arrow keys and ASDW keys.

Come to camera control and CameraControl.cs file.

Although we see three-dimensional movement on the screen, the third-dimensional coordinate of the objects is obtained by **perspective calculations** made on a **two-dimensional** plane. Therefore, both two-dimensional vector and three-dimensional vector calculations must be used. **Mathf.Lerp** and **Quaternion** functions will also be needed to avoid sharp movements and for **angle** calculations.

Use float-type variables for precision and smooth transitions in movements.

#### float hassas=5f; float yumusak=2f;

We can also plan to use two **Vector2** to carry the camera and **new position information**, and a **GameObject** type variable for the **rotation transition** calculations.

Vector2 = newPos; Vector2 = camPos; GameObject player;

In the Start() method, assign the first transform value to the game object named player.

#### player=transform.parent.gameObject;

We can perform the other operations in the **Update()** method since there will be continuity.

Let's create a **Vector2** type **farePos** variable for the **mouse position** and assign the **X** and **Y** coordinates of the mouse with the **Input.GetAxis("X")** and **Input.GetAxis("Y")** commands. This process will be repeated 24 times per second or more.

### Vector2 farePos = new Vector2(Input.GetAxis("Mouse X"), Input.GetAxis("Mouse Y"));

After **transferring** the mouse coordinates to this variable, a **new scale** calculation can be made using these coordinates and the sensitivity and smoothness variables.

### farePos = Vector2.Scale(farePos, new Vector2(yumusak \* hassas, yumusak \* hassas));

Now, we calculate the axial coordinates of the mouse position changes. Assign values to the X and Y elements of the variable we defined above as Vector2 newPos. Do this with the real number math function Mathf and the Lerp, which smooths the transition with interpolation. Since the third parameter in the (float a, float b, float t) pattern of this function is the float t (time) type, we can specify it as 1f/yumusak here.

### newPos.x = Mathf.Lerp(newPos.x, farePos.x, 1f / yumusak); newPos.y = Mathf.Lerp(newPos.y, farePos.y, 1f / yumusak);

With these two components, the new **camera position** can be calculated with a simple addition.

### camPos+=newPos;

We can calculate the rotation locally with the command whose general form is **Quaternion.AngleAxis(float angle, Vector3 axis)**.

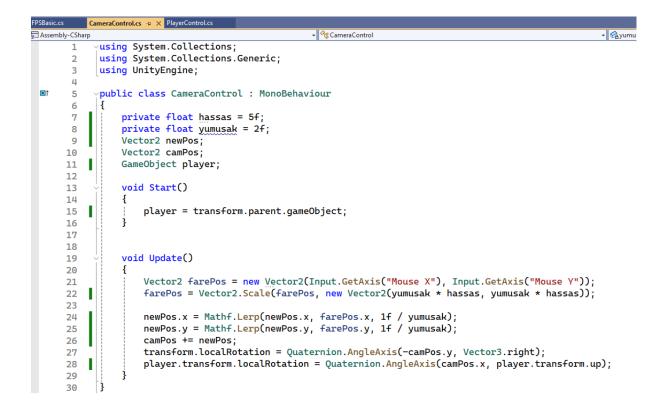
### transform.localRotation= Quaternion.AngleAxis(-camPos.y, Vector3.right);

Here, the negative of the camera position is made for the **angle**, and the **right orientation** is made for the **3D vector** parameter. In some **local** and **global** coordinate differences, the keyboard and mouse can go in the opposite direction of the desired direction. In this case, a correlation should be made with -/+ markings.

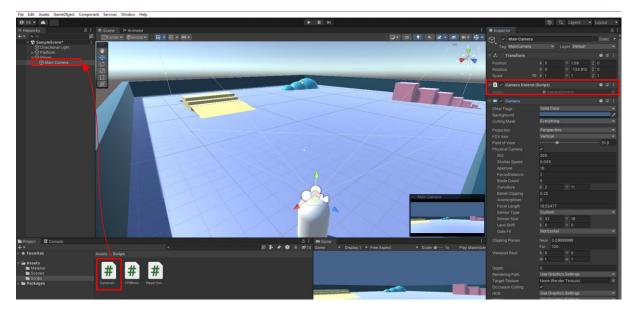
The last process is to calculate the game object **Player**.

### player.transform.localRotation = Quaternion.AngleAxis(camPos.x, player.transform.up);

Here, the local rotation angle and **3D vector position** are calculated.

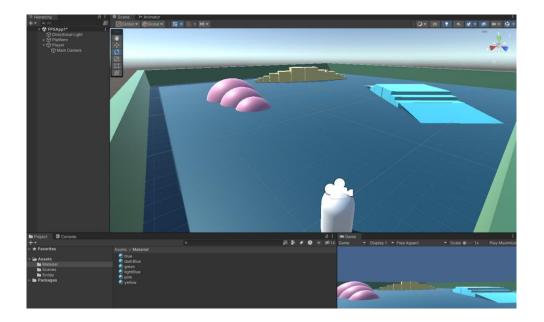


Now, drag and connect the script file to our camera on the scene and make the codes effective on this object. Run it in play mode.

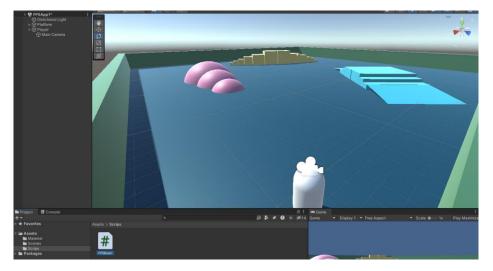


For a better application, the **Player** (capsule) Mesh Renderer check box can be unchecked to make the capsule invisible. In the camera's **Clipping Planes** settings, Near can be changed to 0.01 and Far to 1000 or greater.

For the **second approach**, let's create a new scene named **FPSApp1**. The scene design is similar to **SampleScene** with a small change; the object colors are different. In this application, unlike the previous one, the **FPS** body (**Player**) and camera controls will be combined in a C# file.



Now, create a file named **FPSBasic.cs** under the **Assets>Script** folder and open it in Visual Studio.



This scene utilizes the User1 Productions (YouTube) approach.

We can define **three public float-type** variables to control **walking**, **running** and **jumping** movements. Also, define a public **Transform** type variable for the camera transform reference and a **public float** type variable for the **camera control sensitivity**.

public float walkSpeed = 5f; // Speed of Walking
public float runSpeed = 10f; // Speed of running
public float jumpForce = 5f; // Force of jumping
public Transform cameraTransform; // Reference to the camera transform
public float mouseSensitivity = 2f; // Mouse sensitivity for camera control

Also, add a **Vector3** type variable for the **player speed** coordinates and a **bool** type variable for the **jump** control with the Unity variables **CharacterControl**.

### private CharacterController controller; // character control variable

### private Vector3 playerVelocity; // velocity vector private bool isJumping; // jump - true or false

In the beginning, access the controller variable to the **CharacterController** component of the object that will be used in the character control (after connecting). Now, make sure that the **cursor position** is **centered** on the screen.

```
controller = GetComponent<CharacterController>();
Cursor.lockState = CursorLockMode.Locked; // Lock the cursor to the center of the screen
```

If the continuity of the movement is considered, we should write the control codes of the **character** (**player**) and the **camera** to the **Update()** block.

Whether or not the running movement will occur can be linked to the **left shift key** being pressed. We can provide this information during **vectorial** movement by transferring the result to a **float**-type variable (*for example,* **moveSpeed**). While calculating **moveSpeed**, we can use the **?** condition **operator** means that if the left shift key is pressed, **walkSpeed** is **activated**; otherwise, **runSpeed** is activated. These variables were previously defined as **public float**. In addition, during the player movement, during **walking/running**, we can transfer the **horizontal** and **vertical axis** information to **float-type** variables with the **Input.GetAxis()** command.

The **axial data** and **speed data**, and the **3D vectorial information** of the control variable are transferred to the move variable.

```
float moveSpeed = Input.GetKey(KeyCode.LeftShift) ? runSpeed : walkSpeed;
float horizontal = Input.GetAxis("Horizontal");
float vertical = Input.GetAxis("Vertical");
Vector3 move = transform.right * horizontal + transform.forward * vertical;
controller.Move(move * moveSpeed * Time.deltaTime);
```

There are ready-made functions used for CharacterController type variables.

Let's set up an **if** conditional sentence to **reset** the **velocity** on the **y-axis** after the **jump** with the **isGrounded** function and to **stop** the **jump**.

```
if (controller.isGrounded)
{ playerVelocity.y = 0f;
isJumping = false;
}
```

When the jump is connected to a **key**, the **GetButtonDown** function connected to the data input command (**Input**) can be used. Here, the "**Jump**" keyword can be used, and **isJumping** is connected to the sentence by negating it. When the conditions are **true**, a formula is applied that adds the

square root (Sqrt) of the jumping power and physical gravity to the player speed (playerVelocity). isJumping is set to true.

```
if (Input.GetButtonDown("Jump") && !isJumping)
{ playerVelocity.y += Mathf.Sqrt(jumpForce * -2f * Physics.gravity.y);
    isJumping = true;
}
```

Since the **y-axis** is **vertical/depth**, the physics **gravity** and the **time** regulator **Time.DeltaTime** are multiplied by the **playerVelocity.y** axis to control the **jump** with gravity. Then, the control is reflected in the movement of the variable with the **controller.Move** command.

```
playerVelocity.y += Physics.gravity.y * Time.deltaTime;
controller.Move(playerVelocity * Time.deltaTime);
```

After player control, let's come to camera control. Here, **two** separate calculations are needed **horizontally** and **vertically**.

Data assignment can be made with two **ready-made function templates Input.GetAxis("Mouse X")** and **Input.GetAxis("Mouse Y")** for two **float**-type variables. In order to prevent **sharp** movement, we multiply it with the **mouseSensitivity** we defined earlier. We use these two-coordinate data in three-dimensional **vector** transformation in **rotation**. Here, **Vector3.up** is a useful **ready-made function**.

```
float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity;
float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity;
transform.Rotate(Vector3.up * mouseX);
```

For the camera's **vertical movement**, first, the current **angular** information is transferred to the **3D** vector variable, and then the desired new **angle** is calculated by taking the difference between the **x**-**axis** component of this information and the **y**-**axis** value of the mouse. If this **x**-**axis** value is more than 180 degrees, 360 degrees are **subtracted** from it. Otherwise, the **x**-**axis** value is calculated directly with the **Matf.Clamp** command as a **3D vector**.

Camera rotation angles are also calculated with the Quaternion.Euler() command.

Vector3 currentRotation = cameraTransform.rotation.eulerAngles;

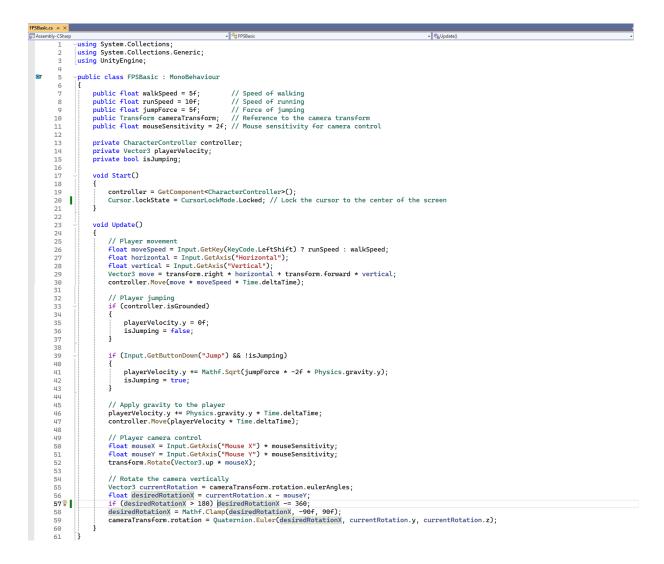
float desiredRotationX = currentRotation.x - mouseY;

if (desiredRotationX > 180) desiredRotationX -= 360;

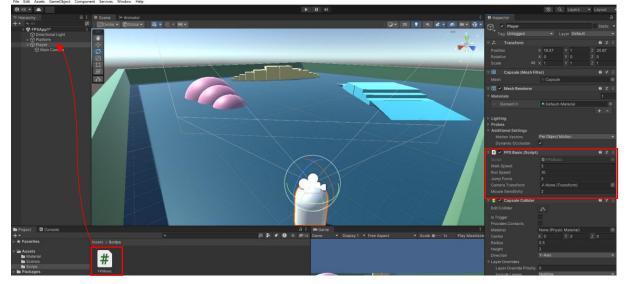
desiredRotationX = Mathf.Clamp(desiredRotationX, -90f, 90f);

cameraTransform.rotation = Quaternion.Euler(desiredRotationX, currentRotation.y, currentRotation.z);

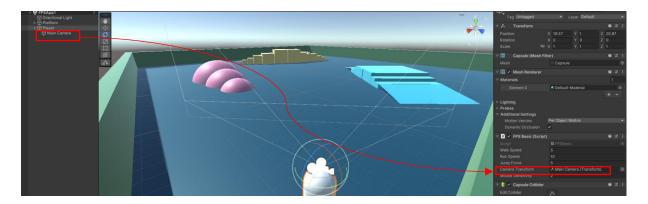
This step-by-step process sequence and its explanations are coded in Visual Studio.



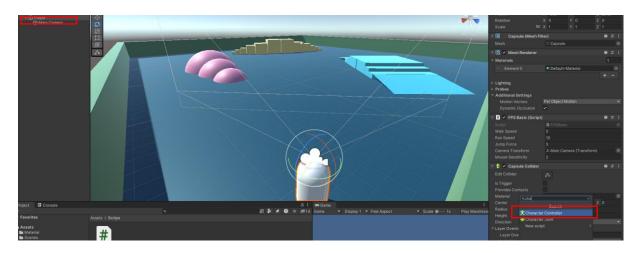
### Now, we can drag the code file to the Player and turn it into a component.



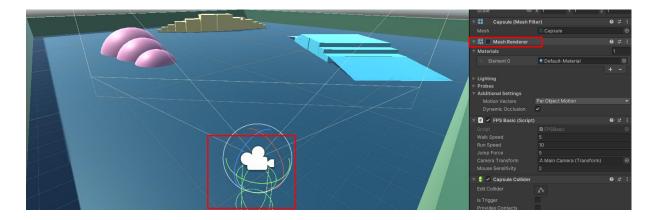
The **Camera Transform** section of the **FPSBasic** script is **empty**, and the camera to be controlled needs to be **connected** here. Drag the **Main Camera** here and connect it.



Another issue is the addition of the **CharacterControl** component defined and used in the codes to the **Player** (**capsule**) object.



If we turn off the **Mesh Renderer** feature of the **capsule**, we will not see the capsule when looking **down**, and its shadow will not appear.



Experience that we move with the **arrow keys** in **play** mode, that we can **look** in every direction with the mouse, and that we can determine the **direction** of movement. While moving, the **left shift key** simultaneously **switches** to **running** mode, and when the **space bar** is pressed simultaneously, the **player** will be seen to **jump**.

As can be seen, we have made coding that is different from the previous approach but ultimately works at a similar level.

**Note:** The codes can be used on our projects one-on-one using various written and visual sources. In some sources, the codes are provided ready-made. It is useful to use them **even without** understanding them in the beginning. Using **ready-made codes** is a **natural** part of application development. As a result of doing and writing many applications with coding, the developer can start to make his comments. This requires **time** and **effort**.

Similarly, locomotion in a mine area is given below.



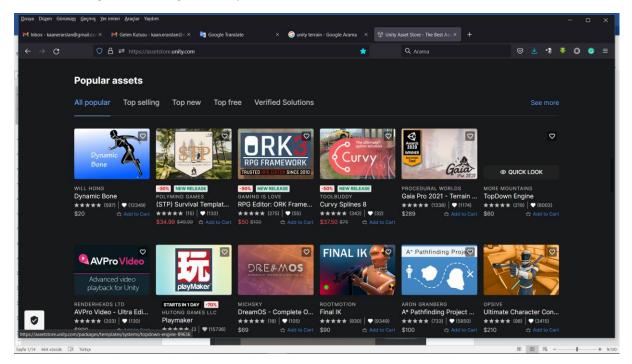


# 5. ADDING SCENE OBJECTS AND ASSET RESOURCES

Before we continue adding **GameObjects** to our scene, it is useful to give brief information about a few websites: **Unity Asset Store**, **Sketchfab**, and **others** such as **Rigmodels**, **GrabCad**, and **3D Warehouse**.

# 5.1.Unity Asset Store

When you enter the address **assetstore.unity.com** and examine it, a site awaits us with thousands of free and paid 2D and 3D objects, packages, ready-made games, etc., that we can add to our scenes. With this content, which makes many of our jobs easier, avoiding having to do everything from scratch and not having to write long codes is possible.



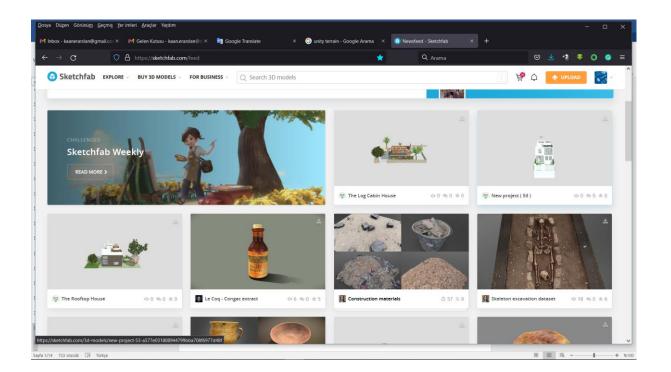
Here, by opening the account we opened on **Unity Hub**, we will be directly connected to our project. By accessing this site from within the project, we will be able to add assets to our Package Manager.

# 5.2.Sketchfab

Another external source is the website called **Sketchfab.com**. There are thousands of ready-made objects, animated designs, and stage materials that belong to many different professions and disciplines. We can add files to our scene by opening an account and downloading/adding them to our Asset window/folder.

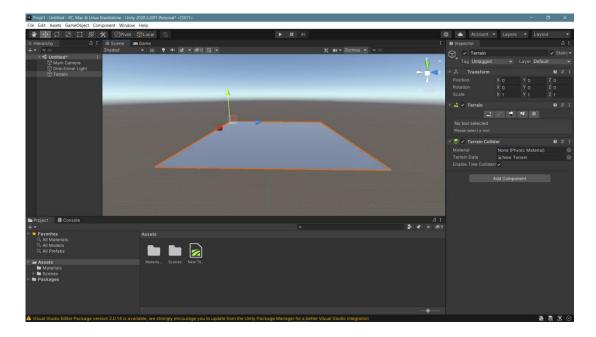
# 5.3.GrabCAD, Rigmodels and 3DWarehouse-Sketchup

**GrabCAD.com**, **Rigmodels.com**, and **3DWarehouse.com** (**Sketchup**) are other very important resources and websites that provide a huge asset library that is mostly free.



### 5.4.Terrain

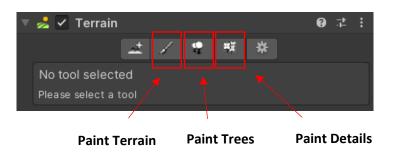
Unity's tool for creating impressive terrain design quickly is **Terrain**. Adding **Terrain** to the **Hierarchy** creates a flat area on the stage that is dozens of times larger than a normal plane.



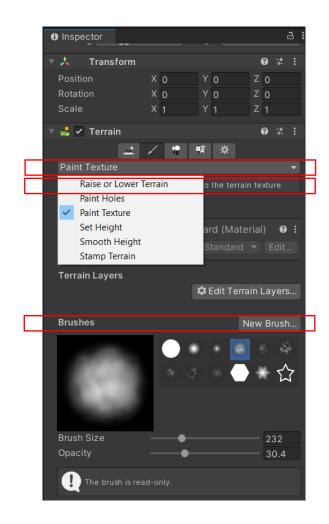
Terrain and Terrain Collider will be seen when examining Terrain's Inspector, apart from the Transform features.

Inspector				а	:
🕞 🗸 🗹 Terrain			✓ S	Stati	c 🕶
Tag Untagged		Layer Defau	ılt		
🔻 🙏 🔹 Transform			0	÷	
Position	X 0	Y 0	ZO		
Rotation	X 0	Y 0	ZO		
Scale	X 1	Y 1	Z 1		
🔻 🍰 🗸 Terrain			9		
*	1	₩ *			
No tool selected					
Please select a tool					
🔻 矿 🖌 Terrain Colli	der		0		
Material	None (Ph	ysic Material)			
Terrain Data	🗟 New T	errain			
Enable Tree Collide	rs 🗸				
	Add Compo	nent			

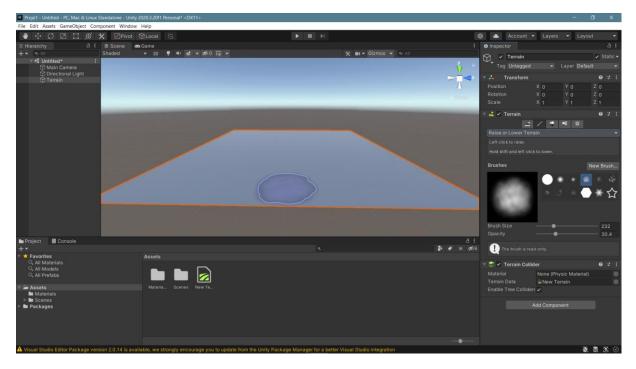
The most important functions for Terrain: Paint Terrain, Paint Trees and Paint Details.

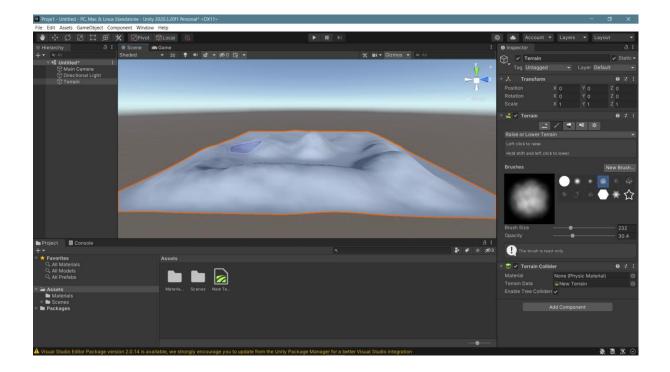


**Paint Terrain** is used to create **height** and **depth** in the field. When this feature is selected and the related menu is opened, its sub-parameters appear. **Raise or Lower** must be selected to obtain height. The same selection is made by lowering the ground with the **Shift** key. Another selection to be made is to select a brush head from the **Brushes** section. When our mouse is moved on the Terrain, the ground will rise or fall according to this pattern.



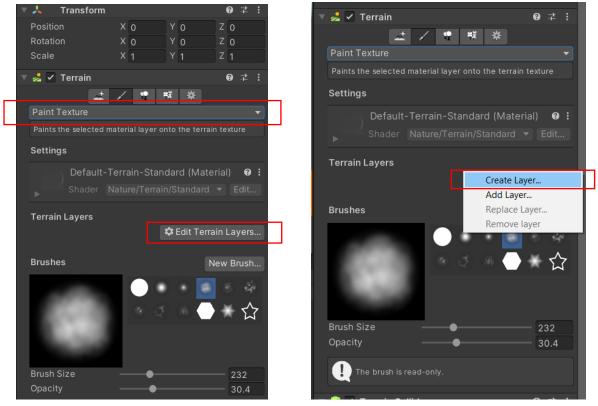
In the following figures, the shape left by the brush on the field and the heights created by moving the mouse are seen.

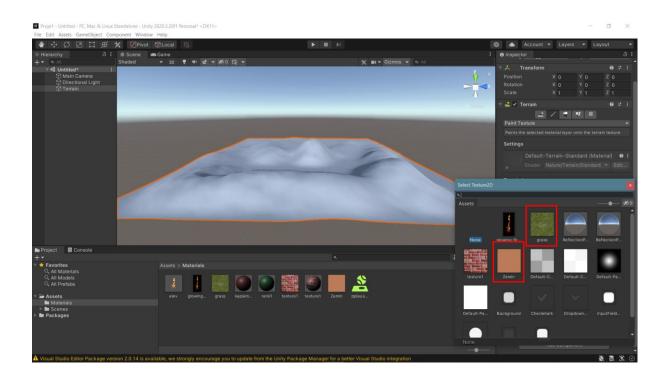




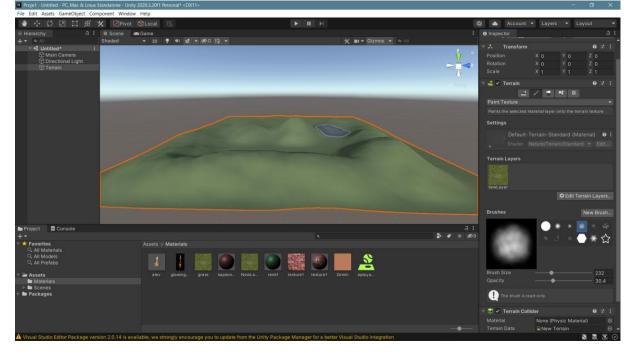
As you can see, a terrain topography was quickly obtained. The brush selected in the **Inspector** also has settings for brush size and solidity-effect degree, such as **Brush Size** and **Opacity**. The effects of changes in these can be observed when creating a field.

To add texture to the field, select **Paint Texture** from the list under **Paint Terrain**. Under **Paint Texture**, click **Edit Terrain Layers-> Create Layer** to open a selection window for a new texture layer.

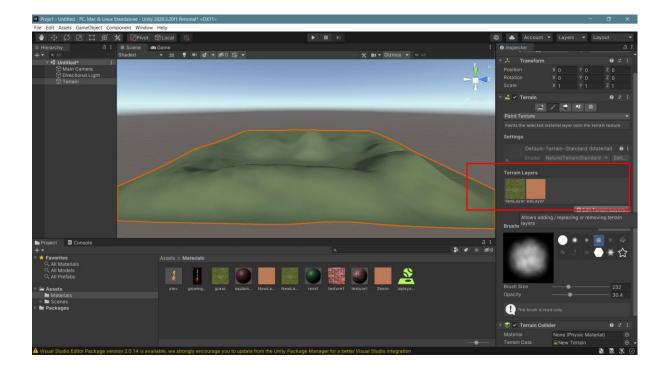


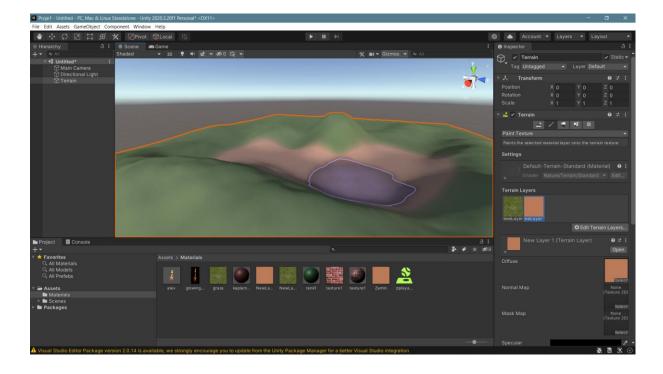


This window shows the files that can be used under the project. In the example, two JPEG files have been added to be used on the ground. One is a plain soil color made in **Paint Brush**, the other is a file found by searching the internet for "**ground texture**" or "**grass texture**". When the "**Grass**" texture is selected, the field is automatically covered with this texture because it is the first and only selection.



Once again, **Edit Terrain Layer > Create Layer** ground selection will appear in the **Terrain Layers** section; the second texture will appear. After that, the desired areas will be painted with the ground color with the brush.

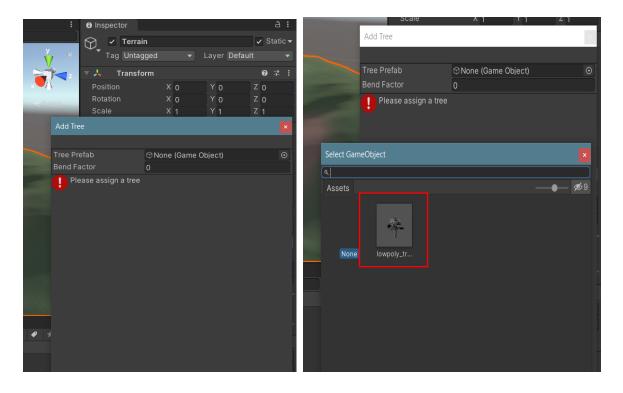




To **add trees** to the field, we first selected a low poly model from the **Asset Store** or **Sketchfab.com** and added it to our project. Now, from the **Terrain** menu, **Paint Trees** and **Edit Trees** can be selected.

差 🗹 Terrain				0		:
	/	¥.	*			
Paint Trees Click to paint trees.	Paint T	rees				
Hold shift and click to erase trees.						
Hold Ctrl and click to e	erase only t	rees of t	the selecte	d type		
Trees						
No trees defined.						
Mass Place Trees		🎝 Ed	dit Trees.	Re	fres	h
<b>Settings</b> Brush Size Tree Density		•	•	— 4 — 8		
Tree Height	Random?	· -				
Lock Width to Heigh	nt 🖌					
Tree Width	Random?					

We add the tree we took to our project from the window that opens with the consecutive Edit Trees >Add Tree > No (Game Object) > Select GameObject > lowpoly\_Tree > Add selections. In the areas where we move our brush, trees appear in a short time and a forest is formed.





It is possible to create a real nature image on the field by doing the same process using grass or flower models with **Terrain > Paint Detail > Edit Detail > Add Grass Texture** file selection and then, **Add**.

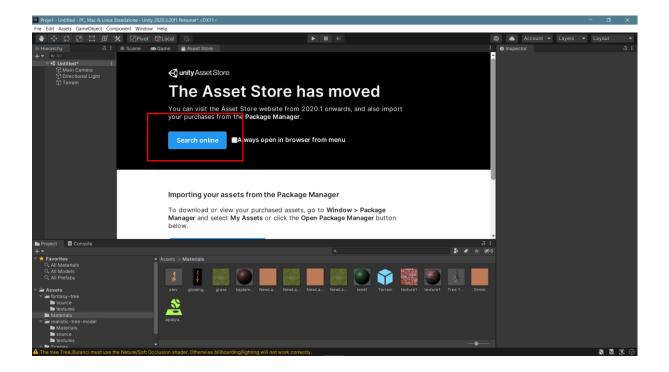
🔻 🧏 🗹 Terrain	0 :t :
<u></u>	
Paint Details	
Click to paint details.	
Hold shift and click to erase details.	
Hold Ctrl and click to erase only details of the s	elected type.
Details	
No Detail objects define <mark>d.</mark>	
🛱 Edit Detail	s Refresh
Detail patches currently allocated: 1024 Detail instance density: 16777216	
Brushes	New Brush
	8 ÷
8 J N	) + ☆
Brush Size	- 100
Opacity — —	0.365
Target Strength -	0.0625



### 5.5.Terrain + Standard Assets Using Unity Asset Store

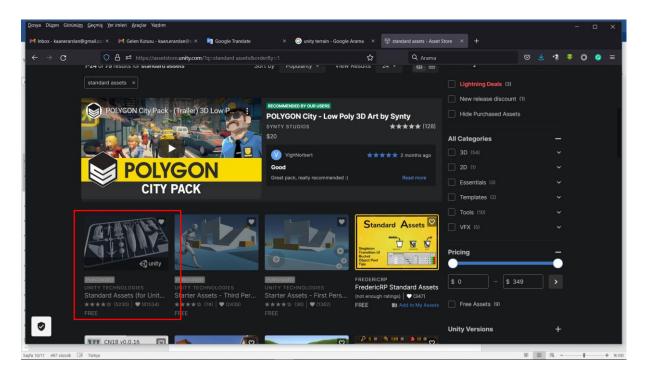
There are Unity Assets with very rich content to create natural landscapes. For this, the Unity Asset Store can be opened, and many ready-made assets, both paid and free, can be downloaded. For this, **Window>Asset Store** should be selected. Then, we connect to the website with the **Search Online** selection. It should not be forgotten that the **same account ID** should be used in both **Unity Hub** and **Unity Asset Store**. In this way, it is possible to download assets directly from the internet to our project.



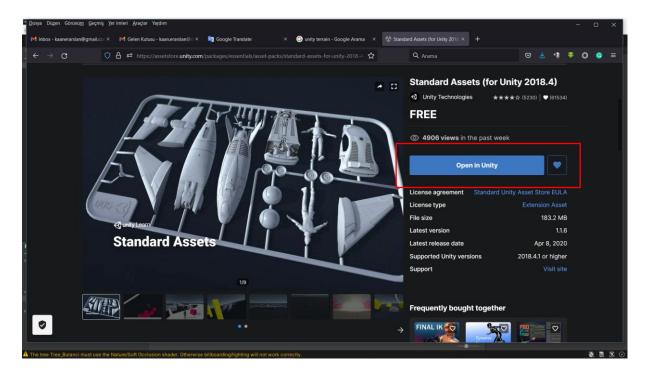


Here, Standard Assets, which is free and belongs to Unity Technologies, should be searched.

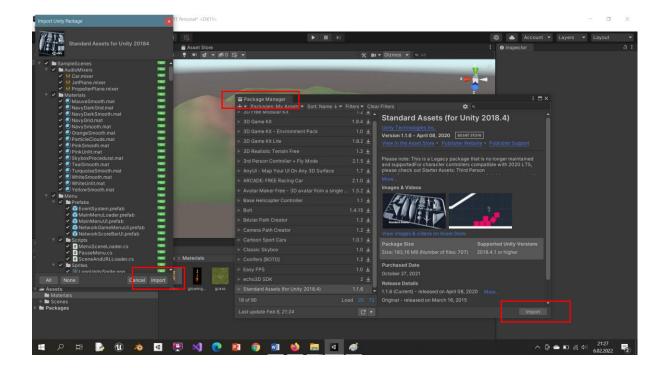
<u>D</u> osya Düzen Görünü <u>m</u> Geçmiş Yerimleri <u>A</u> raçlar Yardım		- 🗆 X
M Inbox - kaanerarstan@gmail.co × M Gelen Kutusu - kaan.erarstan@c × 🎝 Google Translate × Gunity terrain - Google Arama × 🚯 standard assets - Asset		
← → C O B ≓ https://assetstore.unity.com/packages/3d/environments/urban/polygon-city-low-poly-3d-art l: ☆ Q Arama		☺ 坐 🔹 🔻 O 📀 ≡
Ends soon Lunar New Year Mega Bundles Daring Rundle   Braiding Rundle		×
Cunity Asset Store Q standard assets	× 61 🗢	ඌ 🏾 🥶
3D 2D Add-Ons Audi standard assets		Sell Assets
Quality assets     Trusted       Over 11,000 five-star assets     Image: Constraint of the star assets	Community support Supported by 100,000+ forum me	imbers
1-24 of 75 results for standard assets Sort by Popularity + View Results 24 + 🎛 🎛	Refine by	clear filters
standard assets × POLYGON City Pack - (Trailer) 3D Low P	Lightning Deals (3) New release discount Hide Purchased Assets	
SYNTY STUDIOS \$20	All Categories	-
POLYGON     VighNorbert ***** 2 months ago     Good     Great pack, really recommended :)     Read more		*
	Essentials (3)	章 III 156 + %100



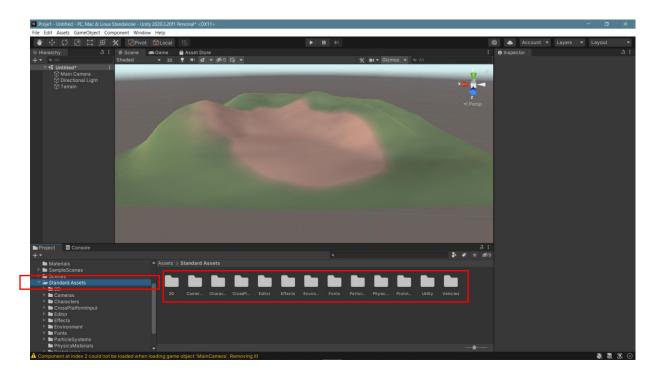
With this selection, we will first add the asset, which contains very rich materials, textures, scenes, C# codes, etc., to our **Assets** archive, and then download it directly to our **Package Manager** with Open in Unity.



**Standard Assets** will be visible and importable in the **Package Manager**. After import, another window will open and show all package content. With the **Import** key in this window, the package is added to our Assets window.



When we follow the path we followed before in **Terrain** under **Standard Assets**, we will see that new trees, textures, grass and flower models, vehicles, features, and **First** and **Third Person Shooter** (**FPS** and **TPS**) codes and features have been added.



With the terrain components offered by Standard Assets, much more successful and realistic field designs can be made. An application environment can be created with houses, transportation vehicles, people, animals, etc. added to this scene.

Moreover;

By positioning **Assets>StandardAssets>Environment>Water>Water4>Prefabs>Water4Advanced**, prefabs in the scene, lake, sea, and stream can be added.

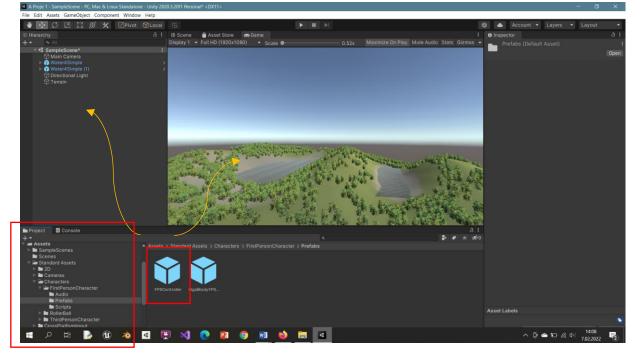


Another feature offered by **Standard Assets** is the **FPS** and **TPS** character options that allow movement within the scene.

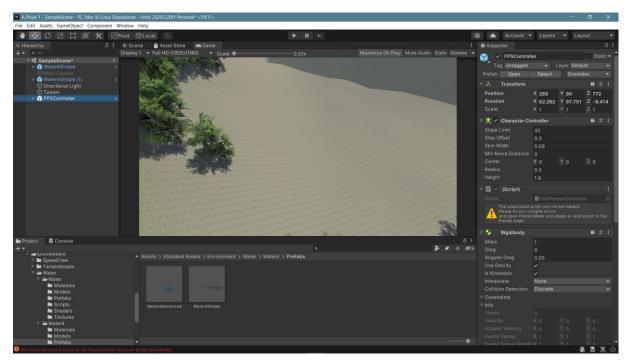
For an FPS scene, the Main Camera needs to be cleared (or inactive). Afterwards;

### Assets>Standard Assets>Character>FirstPersonCharacter>Prefabs>FPSController

The prefab control is added by dragging it to the scene or **Hierarchy**, following this order.



**FPSController** has its camera. Therefore, the position and viewpoint of the **FPSController** in the scene are adjusted.



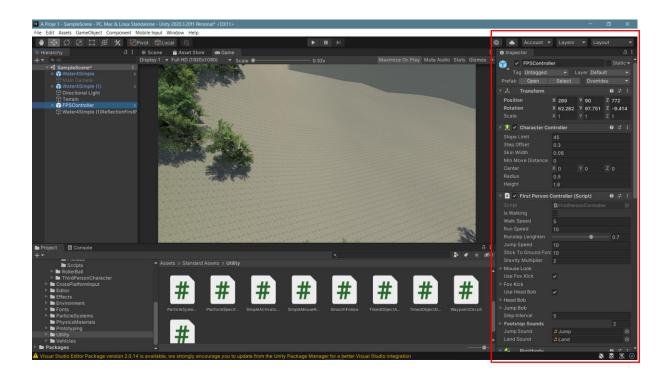
In Unity 2022, a line of code under **Standard Assets** needs to be added, and a small change needs to be made to a line of code. To do this, in the **Assets>Utility>SimpleActivatorMenu.cs** file, add

using UnityEngine.UI; and change the public GUIText camSwitchButton; line to public GUIText camSwitchButton; as

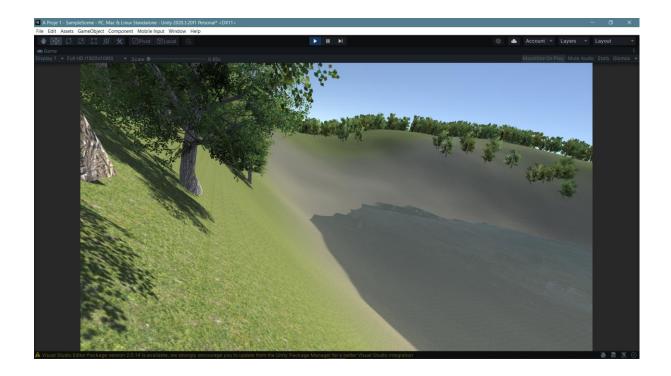
GUIText >Text:

R	Dosya Düzen Görünüm Git Proje Derle Hata-Ayıkla Test Analiz Araçlar Uzantılar Pencere Yardım Ara (Ctrl+Q)	A Proje 1	K	- 0	×
	이 - 이 🚯 + 🎰 🎬 🚰 - 연 - 🛛 Debug - 🖌 Any CPU 💦 🕨 Unity'ye lliştir + 🔬 🔯 🕼 🕼 🖄 🖄 👘 🗇 🦷 🗮 🕅 👘 👘 🖉			🖻 Live Share	æ
Ara	SimpleActivatorMenuus + ×			- 0	8
ç Kut		camSwitchButton		- +	lirim
usu	1 @using System;			- E	e e
4	2 using UnityEngine;				)zelli
	3 using UnityEngine.UI;				der
	4 5 #pragma warning disable 618				
	6 ©namespace UnityStandardAssets.Utility				
	7 {				
	8 📮 public class SimpleActivatorMenu : MonoBehaviour				
	11 // to gameobjects in the scene				
	120 public Text camSwitchButton;				
	13 public GameObject[] objects;				
~	15 16 private int m CurrentActiveObject;				
1					
1					
1	19 e private void OnEnable()				
1					
-	100 % - 🖉 Sorun bulunamadi 🔰 🚿 • 🤇				
2	Hata Listesi Tüm Çözüm • 🛞 0 Hata 🚺 0 Uyan 🕕 1 lieti 🦎 Derleme + IntelliSense •			×4-× م م	
-		roje Dosva	Ci Gizleme		
-		Assembly-CSharp-firstpass SimpleActivatorMenu.cs			
-					
÷.					
			Karati Danatia	las files - 🔎 🗖	
	Hazır	Ť	Kaynak Denetim	hineEkle 🔺 🛛 🐴	đ
2					
- Canda 16	116 706 réserit 🗍 Tirten	14 B			PV 100

Settings in the Inspector of **FPSController** can be changed depending on personal preferences.



As a result, an application is run that can be navigated with the **arrow keys**, **w**, **a**, **s**, and **d keys**, jumped with the **space bar** and controlled 360-degree viewing movements with our **mouse**.

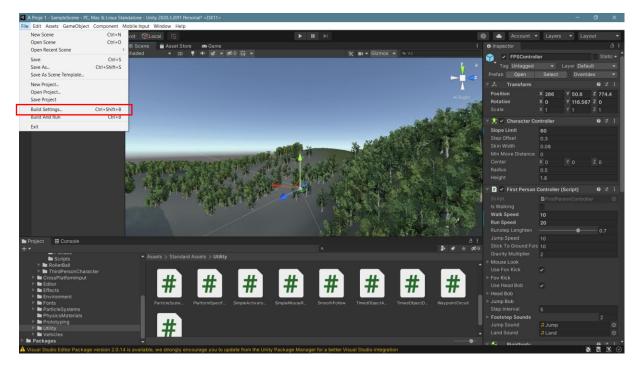


# 6. DEPLOYING PLATFORMS

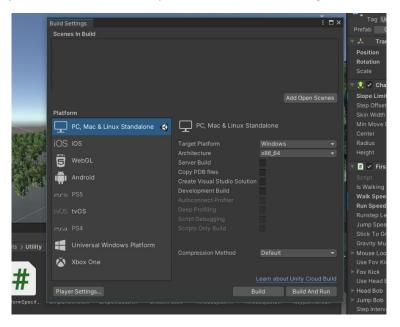
# 6.1.Build Settings

It was stated that **Unity** is a **cross-platform**. This means that projects produced in Unity can be adapted to computers, web, mobile devices, VR (virtual reality), AR (augmented reality), and game console platforms. Let's see how the FPS application we developed for Terrain can be converted into a computer application in **EXE** format.

After the project is completed, **Build Settings** should be selected from the **File** menu.



The window that opens contains the scene, platforms and other settings.



Our scene must be added to the list with the **Add Open Scenes** button. Since we will not be switching to another platform, changing the **Platform** section is unnecessary. However, another platform should be selected according to the type of application, and the **Switch Platform** button that will appear after the selection should be used to switch.

A Proje 1 - SampleScene - PC, Mac & Linux S	Standalone - Unity 2020.3.20f1 Personal* <dx11></dx11>				- 6	2 ×
File Edit Assets GameObject Component	Mobile Input Window Help					
🖤 😳 C 🗵 🗉 🛠 🌾	ZPivot GLocal 🙃			🏟 🔺 Account	👻 Layers 👻 Layout	t 👻
	🖶 🌐 Scene 💼 Asset Store 🛛 🕬 Game		Project Settings		16	X 81
						atic 🗸 🌲
Y • • • • • • • • • • • • • • • • •		i 🗆 🛪 (Add Open Scenes)	Adaptive Performance Audio Editor Graphics Input Manager Physics Physics 2D Piayer Physics 2D Piayer Physics SC Scene Template	Player Company Name Product Name Version Default Icon Default Cursor	DefaultCompany A Proje 1 0.1 None (Testure 20 Select	2 1 1.4
	PLAttorm PC, Mac & Linux Standalone PC, Mac & Li	PC, Mac & Linux Standalone      Architecture     X46,84     Copy PDB files     Create Visual Studio Solution     Development Build     Autoconnect Profiler     Peop Profiling     Script Debugging     Script Debugging	Schol tempate Schol tempate Scholzer Ads Analytics Cloud Dirytos Collaborate Collaborate Collaborate Collaborate Collaborate TextMesh Pro TextMesh Pro TextMesh Pro TextMesh Pro Timpline Version Control XR Plugin Management	Cursor Hotspot 	(Terurr 27 Bare 8 ♥ ® ♥	
Project Console     Console     Scripts     Scripts     Miniefaal     CossPatominput     CossPatominput     Beroronent     Beroronent     ParticleSystems     ParticleSystems		Compression Method Default	bjectA TimedObjectD	Color Space* Auto Graphics API for Vindows Auto Graphics API for Jac Color Gamuel For Mac* = sRG8 WaypointCircsut > Foottee Sound	Gamma + - - - 5	2
Prototyping     Dullity     Dullity     Detectes     Packages	14 is available, we strongly encourage you to u	update from the Unity Package Manager for a better Visual Stud	o integration	Jump Sound Land Sound	# Jump # Land	0 0 2 2 2 3 2 0

One of the most important buttons under Build Setting is Player Settings.

Player Settings are in the Project Settings window, along with many other settings.

🏟 Project Settings				: 🗆 ×			
		۹		a			
Adaptive Performance Audio	Player	Player @					
Editor	Company Name	Company Name     DefaultCompany       Product Name     A Proje 1       Version     0.1					
Graphics	Product Name						
Input Manager Package Manager							
Physics Physics 2D	Default Icon			None (Texture 2D)			
Player Preset Manager				Select			
Quality Scene Template Script Execution Order	Default Cursor	Default Cursor					
▼ Services				Select			
Ads Analytics	Cursor Hotspot		X O	YO			
Cloud Build	Ģ		Ē	-			
Cloud Diagnostics Collaborate			~				
In-App Purchasing	Settings for PC, Mac & Linux Standalone						
Tags and Layers	▶ Icon			ī			
TextMesh Pro Time	Resolution and	d Presentation					
Timeline Version Control	Splash Image						
XR Plugin Management							
	Rendering			,			
	Color Space*			a 🔻			
		API for Windows	~				
		Auto Graphics API for Mac 🗸					
			~				
	Color Gamut For	Mac*					
	= sRGB						
				+, -			

**Company Name** and **Product Name** information must be entered for the project title that will be converted into an application. An image file prepared for the icon image that will appear in the folder is placed in the **Default** Icon section.

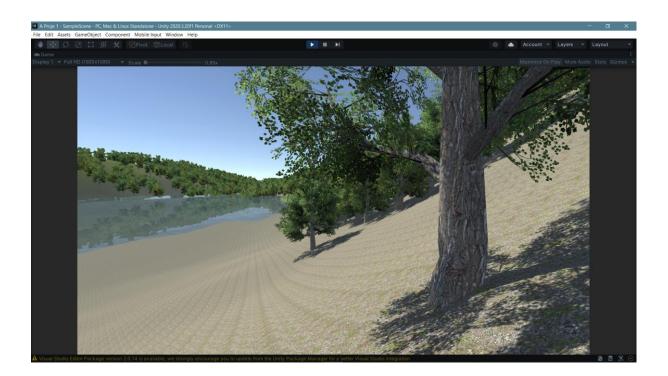
🌣 Project Settings		: 🗆 ×						
Adaptive Performance Audio	Player ø							
Editor	Company Name	DPU						
Graphics	Product Name	AraziGezisi						
Input Manager Package Manager	Version	0.1						
Physics Physics 2D Player	Default Icon							
Preset Manager Quality Scene Template	Default Cursor	Select None (Texture 2D)						
Script Execution Order ▼ Services Ads Analytics	Cursor Hotspot	Select X 0 Y 0						
Cloud Build Cloud Diagnostics	ц, ios	<u> </u>						
Collaborate	Settings for PC, Mac & Linux Standalone							
In-App Purchasing Tags and Layers	► Icon							
TextMesh Pro Time	Resolution and Presentation							
Timeline	► Splash Image							
Version Control XR Plugin Management								
Xit i ugin Management	Other Settings							
	Rendering Color Space*	Gamma 👻						
	Auto Graphics API for Windows							
	Auto Graphics API for Mac							
	Auto Graphics API for Linux	✓						
	Color Gamut For Mac*							
	= sRGB							
		+						

These definitions are sufficient for the PC application. Now we can proceed to convert the scene to an application (**EXE program**). To do this, click the **Build and Run** button in the **Build Settings** window. With this selection, the program asks us to create/open the folder where the application will be created.

In the open window, right-click on the mouse and select **Folder** from the **New** option. Save the folder name of the application and select it. The program will automatically start compiling and creating **EXE**. It is normal for this process to take a certain amount of time, depending on the computer's hardware power.

After the application is compiled, the game <u>starts</u> immediately. After visiting the field, press **Alt+F4** to **exit** the game. The **EXE** file and other component files and folders are created under the specified folder.

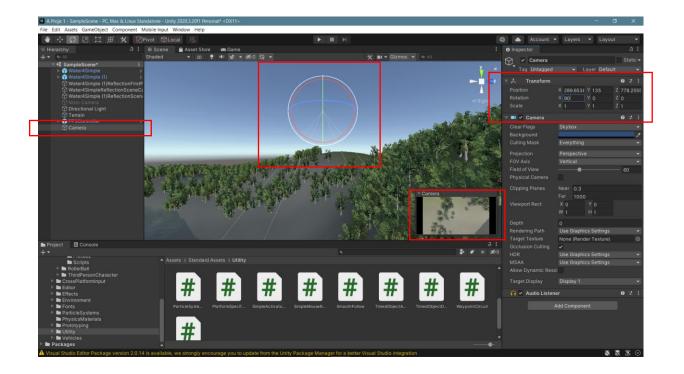
A Proje 1	- SampleScene - PC, Mac & Lin	nux Standalone - Unity 2020.3.20f1 Personal*	<dx11></dx11>						σ×
		ent Mobile Input Window Help		67520701 00122001 1	12611		07270 070700 07070700		
*	Interest mental interest investmental	Pivot GLocal 15		► II			🖗 📥 Account	<ul> <li>Layers</li> <li>Layou</li> </ul>	n -
Hierarch + ▼			no-Game 4 - 愛 = 愛 0 珙 =			Gizmos - Q-All	i Inspector		
	SampleScene*				Ê		FPSContro		
	Build Windows				×		Tag Untagged	✓ Layer Default	
			in the second second second second second second second second second second second second second second second			Build Settings			
	$\leftarrow \rightarrow \lor \uparrow \square \cong Belg$	geler > A UNITY > A Proje 1 >	× 0	9 Ara: A Proje 1		Scenes In Build			
<i>C</i> ▲	Düzenle 👻 Yeni klasör			E •	0	<ul> <li>Scenes/SampleScene</li> </ul>			
00	📙 A Proje 1 🕜		Değiştirme tarihi		Boyut				
đ	📕 .vs	E as	7.02.2022 14:23	Dosya klasörü					
	Assets	Assets	7.02.2022 13:05	Dosya klasörü					
	📮 Library	Library	7.02.2022 17:03	Dosya klasörü				Add Open Scen	
	Logs	Logs		Dosya klasörü		Platform			
	_ obj	obj		Dosya klasörü		5			
	Packages	Packages		Dosya klasörü	56	PC, Mac & Linux Standalone	PC, Mac & Linux Star	idalone	
	ProjectSet	ProjectSettings	7.02.2022 13:00	Dosya klasörü		iOS ios	Target Platform	Windows	
	Temp	Temp	7.02.2022 17:03	Dosya klasörű				x86_64	
	UserSettin	UserSettings	7.02.2022 13:00	Dosya klasörü	邂逅	WebGL			
1	and the second sec	Klasör				Android	Copy PDB files		
	Görünüm	Kisayol			- , M		Create Visual Studio Solution Development Build		
	Sıralama ölçütü Gruplandırma ölçütü	Microsoft Access Veritabanı			ži		Autoconnect Profiler		
	Grupiandirma oiçutu Yenile	SketchBook Bmp Image File							
Proje ct		Microsoft Word Belgesi		Klasör Seç İpta	e i				
+•		Microsoft PowerPoint Sunusu			4				
		Microsoft Publisher Belgesi				Universal Windows Platform			
		Etrl+Z						Default	
• • •	Visual Studio ile aç	Metin Belgesi	4	4	#	Xbox One			
- Ir	Quina Crigini verim	Microsoft Excel Çalışma Sayfa	。 #	#	#				
	Yeni	> 🔯 WinRAR ZIP arşivi	mspeci SinpleActivato		noothFollow	T Player Settings		ild Build And Ru	
- P 🖬	Ozellikler	Particle System Platt	annapeen an anna anna anna anna anna ann						
	mysicsmatenais						Footstep Sounds		
P 🖿 U		#							
P 🖿 V	/ehicles							# Land	
	iyes Iudio Editor Daokano version	2.0.14 is qualitable, we strength opening	to you to undate from the L	laitu Daokana Managar fa	co bottor View	at Chudio Integration	Dialdhody	×	



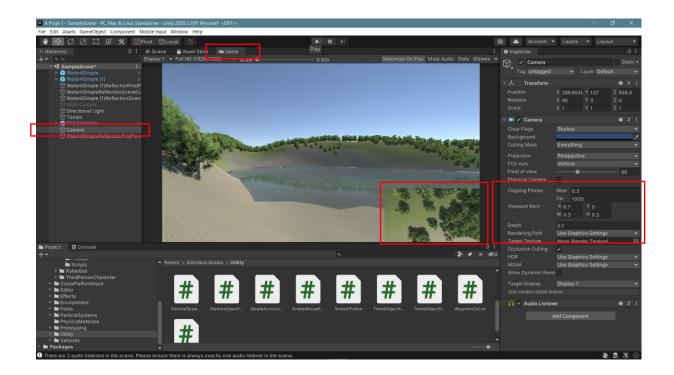
## 6.2.Cameras

Other **cameras** can be added next to the current camera, including their viewing angles, screen sizes, and locations. To do this, right-click on our mouse under Hierarchy, create a camera, determine its angle, and position it in place.

In the example, a camera is added, positioned to see the field from above, and its angle is adjusted.



To get results from the **Inspector>Viewport Rect** setting, the positions on the screen are set with **X** and **Y**, and the dimensions of the new camera on the screen are set with **W** and **H**. If the **Depth** setting is a number greater than zero, it will be seen together on top of the **FPS** camera. It is possible to see these settings more clearly in **Game** mode.



However, finally the **Camera** needs to be dragged under the **FPSController**. It is possible to add other cameras and different angle views.

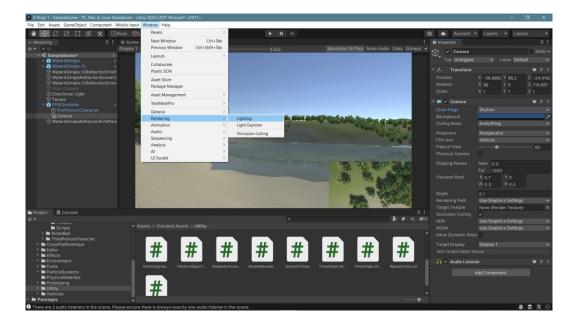
# 6.3.Parent Child Relation

By dragging the camera under the **FPSController**, a **Parent-Child** connection is created between them. The camera becomes the child of the **FPSController** and is subject to the movements of the **FPSPlayer**. In game mode, when the **FPS** moves, the other camera will follow it, and the image above will follow the image below.

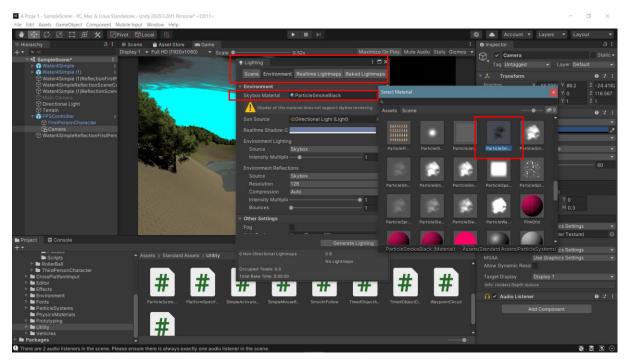


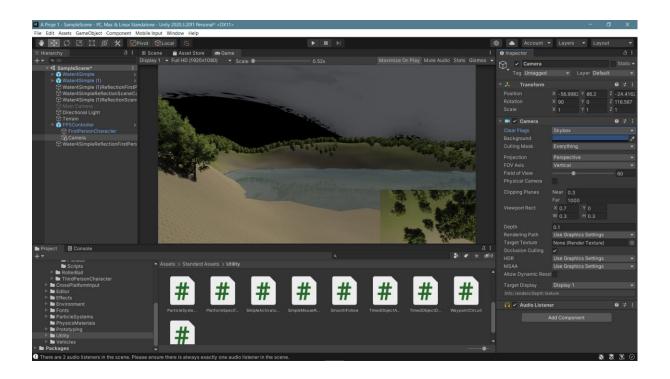
## 6.4.Skybox

The **sky** seen behind the scenes is a standard application. If desired, it can be changed to a fixed color or different sky backgrounds. Skybox assets can also be found in the Asset Store. For the Skybox application, the relevant window is opened via **Window>Rendering>Lighting**.



In the **Lighting** window settings, **Environment>Skybox Material** opens the window with the image files. The selected image file is assigned as the new sky.

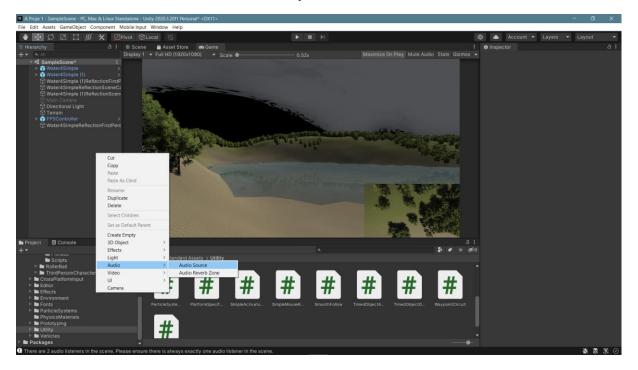




In a normal project where Standard Asset is not used, **Skybox** change is simpler. **The main Camera is selected**, and a fixed color can be selected instead of Skybox in Inspector.

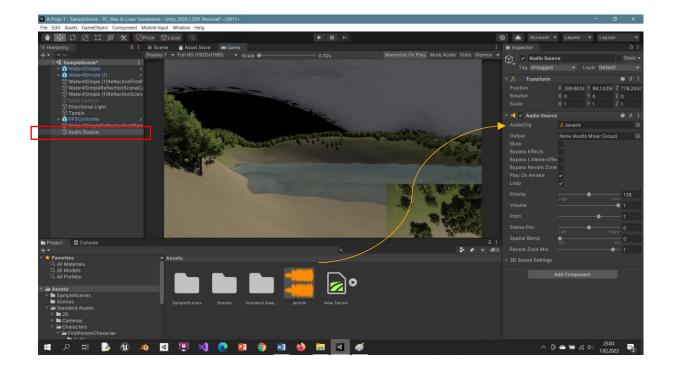
# 6.5.Adding Audio and Video files

An audio file is added to the scene via **Hierarchy>Audio>AudioSource**.

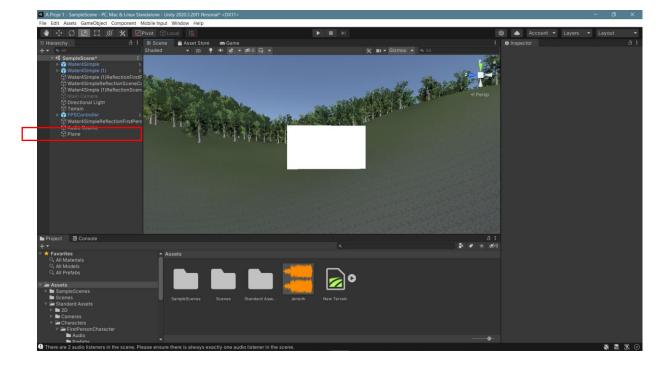


Afterward, Audio is selected, and the previously prepared audio file in the **Inspector** is dragged to the **AudioClip** area and connected. The **loop** check box is checked to make the audio file loop continuously.

Inspector	a :
Audio Source	Static 🗸
Tag Untagged 👻 Layer Defa	ult 👻
Transform	0 ‡ :
Position X 289.6538 Y 94.1425	
RotationX 0Y 0ScaleX 1Y 1	Z 0 Z 1
Audio Source	0 ‡ :
AudioClip None (Audio Clip)	$\odot$
Output None (Audio Mixer Gro Mute	oup) 💿
Bypass Effects	
Bypass Listener Effer	
Bypass Reverb Zone Play On Awake  ✓	
Loop	
Priority High	Low 128
Volume	-• 1
Pitch	1
Stereo Pan	Right 0
Spatial Blend	3D 0
Reverb Zone Mix	<b>●</b> 1
▶ 3D Sound Settings	
Add Component	



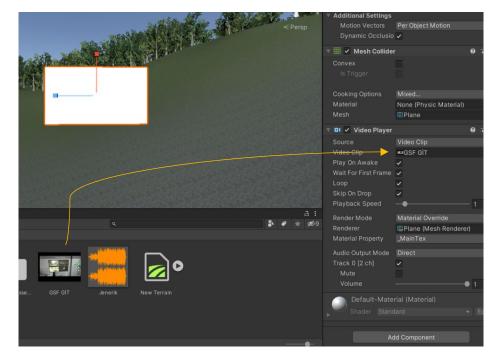
To add an image file to the scene, it is possible to add a **Plane** to which the file will be assigned. The **Plane** is positioned on the scene and its dimensions are adjusted.



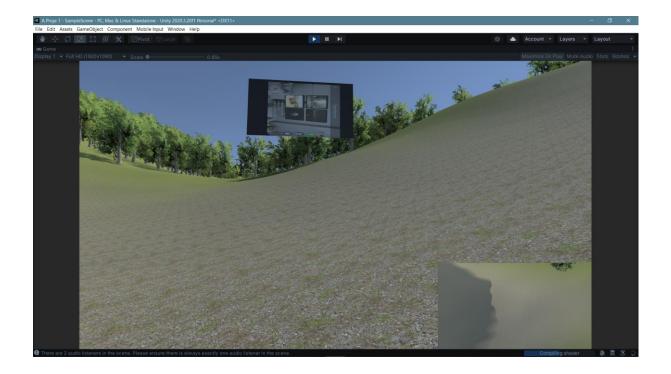
To add the prepared **video film**, first select **Plane**. The component is added with the **Add Component>VideoPlayer** selection.

ଔୢ୕୕⊻								S	Stati	c▼
Tag	Untagged			L	ay	er Defau				
▼ 🙏 👘	Transform							8		
Positior			718.9			51.3		7	14.9	•
Rotatio			0			0		9	0	
Scale			0.984	114				1.	779	47
	Plane (Mesh F	ilt	er)					8		
Mesh			Plane	9						
▼ 🖽 🗸	Mesh Render	er						0		
► Materia								1		
<ul> <li>Lightin</li> </ul>										
	ອ Shadows	0	n							•
	eive Shadows									
	ribute Global I									
			aht P							
▼ Probes										
Light	Probes	B	lend F	rob	bes					
Refle	ction Probes	B	lend F	rot	bes		-			•
Anch	or Override	N	one (1	Frar	nsfe	orm)				$\odot$
Addition	onal Settings									
Motie	on Vectors	P	er Obj	ect	M	otion				
Dyn	۹vi						×			
▼ ⊞ 🔽		(	Searc	h				_		
	DI Video Diov		Jeare					9		
Conve:	Video Play									
	Visual Effe		)f Vier							
	# Target Fiel	u C	J viev	N						
Cookin	New script									
Mataria										

**Video Player** has been added to the **Inspector** connected to the **Plane**. The **video file** that was previously prepared and added to the Assets section is dragged to the **Video Clip** area and connected here. If the **Loop** box is checked, it will loop.



The **Plane** added to the field now has a video player feature.



## 6.6.Animation

Let's add an **animation** without changing the scene. Animations can be added to a certain extent in Unity. The relevant object is selected and applied to it.

Let's add any 3D object or a cylinder to our scene and scale it like the gold image. Let's choose **Window>Animation>Animation**.

Lyos	A Proje 1 - SampleScene - PC, Mac & Linux Standalone - Unit				
Project       Consol         Project       Consol         Project       Consol         Amazion       Amazion         Amazion       Amazion         Amazion       Amazion         Amazion       Amazion         Consol       Amazion         Project       Consol         Project       Consol         Amazion       Amazion         Amazio	File Edit Assets GameObject Component Mobile Input	Window Help			
Autorial Console     Auto	👑 🕂 💭 🔝 🎞 🤀 🗶 🖉 Pivot 🐨 L	Panels >		📦 📥 Account 👻 Layers 👻	
Supplication     S				i Inspector	
Project I Console For example Console<	+ - All Shaded	Previous Window Ctrl+Shift+Tab	💥 🖬 🕶 Gizmos 👻 👁 Ali	🛇 🧹 Cylinder	Static -
Visites Single Reflections Conserved Singl		Layouts >		<b>Y</b>	
Wateralsingenet licetonics instructions on the second of the second o		Collaborate		V - rag Untagged + Layer D	
Wated Single (BaltectonScene)       Wated Single (BaltectonScene)		Plastic SCM		🗙 😹 🔍 🔻 🖈 Transform	071
Water Careera   Water Single Relater Consult up 11   FFS Consult   Water Single Relater Consult up 11   Water Single Relater Consult up 12 <td></td> <td>Asset Store</td> <td></td> <td></td> <td>Z 981.5</td>		Asset Store			Z 981.5
Ause Management     Objectional Light     Opicate     Ause Management     Opicate     Ause Management     Opicate     Ause Management     Opicate     Ause Management     Opicate		Package Manager		Rotation X 0 Y 90	
Project Console      Anatom	🛱 Directional Light	Asset Management >	the the set	Scale X 0.69866 Y 0.03	572E Z 1
Project Console     Conso		TextMeshPro >	a series of the	👻 📰 🛛 Cylinder (Mesh Filter)	
Project Console     Conso	G Water4SimpleReflectionFirstPers	General >		Mesh ECylinder	o
Cylinder      Advantation     Advantation     Advantation     Cylinder      Advantation     Advantation     Cylinder      Cylinder      Advantation     Cylinder      C		Rendering >		T 🖽 🗸 Mesh Renderer	0 2 1
Audo       Animator         Sequencing       Animator Plrameter         Anaytiss       Animator Plrameter         Anaytiss       Animator Plrameter         Anaytiss       Cast Shadows         Ul Tookit       Sequencing         Anaytiss       Cast Shadows         Ul Tookit       Sequencing         Anaytiss       Cast Shadows         Ul Tookit       Sequencing         Anaytiss       Cast Shadows         Project       Console         +       Cast Shadows         Cast Shadows       Control Vertices         Al Modelis       Sequencing         Antonio Strate       Sequencing         Antotoolis       Sequencing		THINGTON		E Materials	
Analysis     Anatarials     Analysis					
A 1 A 2 A 2 A 2 A 2 A 2 A 2 A 2 A 2 A 2			Animator Parameter		
U Tooket  Contribute Global I Contribute Global Contribute	10251210			Receive Shadows 🗸	
Receive Global III. Light Probes     Receive Global     Receive Global IIII. Light Probes     Receive Global III. Lig	250303			Contribute Global I	
Project © Console Project © Console Project © Console • Project © Console • Additional Settings • Motion Vector • Project © Console • Additional Settings • Console • Additional Settings • Console • Additional Settings • Console • Cons	100000			Receive Global IIIu Light Probes	
Reflection Probes Biand Probes Anchor Overrise Rome (Transform) Moment Transform) Moment Coclusio Moment Coclusi					
Achor Override None (Transform)     Additional Settings     Call Material     Call     Call Material     Call     C					•
Project E Console      Project E Console      Project E Console      Project B Console				Contraction of the second second second second second second second second second second second second second s	<u> </u>
Image: Strandard Assee:       Scenes					
Project Console      Proj					
Consular Collidar      Consular Collidar					· · · ·
C. Al Materials       Call Materials       Edit Collider       Edit Collider         C. Al Materials       Call Materials       Edit Collider       Is Trigger         Material       CampieScenes       Scenes       Standard Asse       atim       OSF OIT       Aneraik       New Material       Ocnter       X 596046 Y o       Z -0.9400         Raduus       D.Sconool       Height       2       Direction       Waterial       Other         Material       OSF OIT       Janerik       New Material       New Terrain       Direction       WAterial       Other         Material       Cameras       V       Cameras       Other       Cameras       Other       Cameraid       Other         V       Cameras       V       Cameras       Other       New Material       Other       Terrain	Project 🗄 Console				
C. Al Materials       C. Al Materials       Edit Collider       Edit Collider         C. Al Materials       C. Al Materials       Stample/Scenes       Is friger         Material       Material       Scenes       Standard Asse       atm       OSF OIT       New Material       New Terrain         Material       Scenes       Scenes       Standard Asse       atm       OSF OIT       New Material       New Terrain         Megiphic       Direction       V*Axis       Direction       V*Axis       Terrain       Mew Material       Image: New Materi	+ V			🔭 👻 🙁 💯 😌 😴 🤤 💟 Capsule Collider	0 ≠ 1
C All Prefsbs C All Prefsbs Scenes Scenes Standard Asse C All Prefsbs Scenes Scenes Scenes Standard Asse Standard					
C. All Predod       Material       None (Physic Material)       ©         C Attarial       SampleScenes       SampleScenes       SampleScenes       SampleScenes       SampleScenes       SampleScenes       SampleScenes       New Material       New Material       Caterial       None (Physic Material)       ©         C Attarial       SampleScenes       Scenes       Standard Asse       atun       OSF GIT       Jenerik       New Material       New Terrain       Material       None (Physic Material)       ©         C Attarial       SampleScenes       Scenes       Standard Asse       atun       OSF GIT       Jenerik       New Material       New Terrain       Material       New Terrain       New Material       0       Caterial       Caterial       New Material       0       Caterial       New Material       0       Caterial       New Material       0       Caterial       New Material       New Material       0       Caterial       New Material       New				Is Trigger	
> B SampleScenes       SampleScenes       Scenes       Standard Asse       attin       OSF OIT       Junrarik       New Material       New Terrain       Radius       0.5000001         > B SampleScenes       Scenes       Standard Asse       attin       OSF OIT       Junrarik       New Material       New Terrain       Height       2       Direction       MAxis       Direction       MAxis       Terrain       New Material       New M	Q. All Prefabs				erial) O
■ Scenes     SampleScenes     Scenes     Standard Asse     altan     GSF GIT     Junerik     New Terrain     Height     2       > ■ 20     >     >     Camprais     Direction     V-Axis     ✓       > ■ Camprais     >     >     ● 20     ✓     New Material (Material)     ● 2 ± 1	🔻 🗃 Assets				
Canadard Assets     C		- $ -$			
P ■ 2D Direction M-Axis  P ■ Cameras P ■ Cameras P ■ Cameras P ■ Cameras P ■ Cameras P ■ Cameras P ■ P ■ P ■ P ■ P ■ P ■ P ■ P ■ P ■ P ■			aitin GSF GIT Jenerik New Material New Terrain		
v The Characters On the American (Material) 0 ≠ i ■					*
Thatacters				New Material (Material)	0.2 1
	<ul> <li>Characters</li> <li>FirstPersonCharacter</li> </ul>			Shader Standard	← Edit
Mudio	🖿 Audio				Contraction of the second
	Drofahe *	there is always exactly one audio listener in t	ne scene		

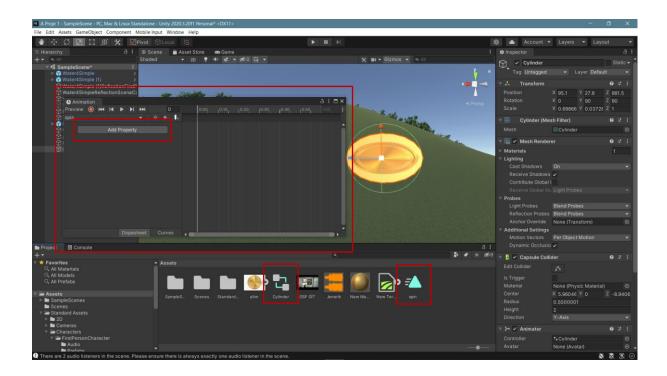
A window opens for the animation. Here, a new animation is started with the **Create** button.



First, a file name is given to create the file related to the animation.

	mponent Mobile Input Window Help							
S 🖸 🖬 😸	🛠 🕢 Pivot 🖓 Local 👸					🖗 📥 Account 🗨	🕶 Layers 🔻 La	
	a 🗄 🌐 Scene 🕋 Asset Store					Inspector		
🗣 All		? ● \$ * \$ 0 \$ *	*	🕻 🔳 🕶 Gizmos 👻 🔍 All		🕥 🖌 Cylinder		
SampleScene*					¥ =	Tag Untagged		
Create New Animatic	a 				x	Transform		0 ‡
$\frac{1}{2} \leftarrow \rightarrow \vee \uparrow \square \circ$	Belgeler > A UNITY > A Proje 1 > Asse	s> v O ,2 /	ra: Assets			Position	X 95.1 Y 27.8	Z 981.
Dúzenle • Veni kla			E • 0			Rotation		Z 90
				The second second		Scale	X 0.69866 Y 0.037	
📕 A A UNITY D		Değiştirme tarihi T	ür Boyut	2. solar states		Cylinder (Me	sh Filter)	0 7
A Ders UNITY	SampleScenes	7.02.2022 13:05 E	losya klasörü	All and a second		Mesh	Cylinder	
🗧 📙 A UNITY	Scenes		losya klasörü 🥟		THE	🔻 🎛 🖌 Mesh Rende		0 3
3D Models	Standard Assets		losya klasörü 🛛 🔛	and the second s	and the second sec		rer	
SDDXF						► Materials ▼ Lighting		
📜 3DDXFInpo						Cast Shadows		
📜 A Proje 1						Receive Shadows		
.vs						Contribute Global		
AraziGezis								
Assets						T Probes		
	~ <		> = >	×			Blend Probes	
Dosya adı:	lew Animation.anim					Reflection Probes		
Kayıt türü: a	nim (* anim)						None (Transform)	
						Additional Settings	Per Object Motion	
					a i	Dynamic Occlusio		
je ot 🔺 Klasörleri Gizle		Ka	ydet  ptal		en:			
avorites						🔻 🔒 🖌 Capsule Coll		03
All Materials								
All Models All Prefabs		o begin animating Cylinder, create ar	Animator and an Animation Clip	📙 🦳 г				
		Create		National (Constanting)	<u>_</u> ]0		None (Physic Mater	
Issets SampleScenes							X 5.96046 Y 0	Z -8.9
Scenes				rik New Material New		Radius Height		
Standard Asset:						Direction	2 Y-Axis	
2D Cameras								
Characters						New Materia		
						Shader Stan	and a seal	✓ Ed
FirstPersor     Audio						snader atan	dard	10 OPA

We named the animation **spin**. After that, a new window will open where we will create the animation.



With AddProperty, animation types will be displayed. Rotation is selected under Transform.

A Proje 1 - SampleScene - PC, Mac & Linux Standalone - Unity 2020.3.2011 Personal* <dx11></dx11>					$\sigma \times$
File Edit Assets GameObject Component Mobile Input Window Help			The HEADER BRANDSAUM	a mitoriorioriana autor	
👋 😳 🖸 🛄 🤀 🗶 🖉 Pivot 🎧 Local 👸	E II H	e		Layers 🔻 Lay	
Hierarchy ∂ i ∰ Scene ∰ Asset Store ons Game + ▼ ≪ All Shaded ▼ 20 ♥ ♥ ♥ ♥ ♥ 0		u ▼ Gizmos ▼ ≪ All	Inspector		
+ v c∧m Snaded + 20 + 40 ≥ + 900 ±	× •		℃_ ✓ Cylinder		Static -
► 😚 Water4Simple >			Tag Untagged	<ul> <li>Layer Defa</li> </ul>	ult 👻
Water4Simple (1)     Source (1)      Water4Simple (1)ReflectionFirstP		ו 🛌	Transform		
Water4SimpleReflectionSceneC:		T. T. T. T. T. T. T. T. T. T. T. T. T. T		X 95.1 Y 27.8	Z 981.5
C Animation	a:⊡×	< Perso			Z 90
Preview () H4 H4 > >1 >H 0 0.00 10:10 10:20	10:30, 10:40, 10:50, 1:00, 1			X 0.69866 Y 0.0372	
😚 spin 🗢 🔶 🖡		< Pesp	Cylinder (Me	sh Filter)	0 : :
Add Property		- Caller		Cylinder	0
💮 , 🕹 Position	:	it at	= 📆 🛩 Mesh Render		0 2 1
©i ARotation State	÷ 🔰 🦯			er	
🖻 🚟 Mesh Renderer			<ul> <li>Materials</li> <li>Lighting</li> </ul>		
Equip Collider			Cast Shadows		
			Receive Shadows		
				Blend Probes	
			Reflection Probes		
			Additional Settings	None (Transform)	
Dopesheet Curves				Per Object Motion	
Project B Console		A I	Dynamic Occlusio		
		5 4 ± 259			
* Favorites Assets			🔻 🔒 🖌 Capsule Coll		071
Q. All Materials		<u></u>			
9. All Models	Ng 🖵 🛛 💷 🦲 .				
	/Y La 🚥 📒 💹 .			None (Physic Materia	
Assets SampleS Scenes Standard altr			Center Radius	X 5.96046 Y 0	Z -8.9406
Scenes			Height		
			Direction	Y-Axis	
⊨ I 20 ⊨ In Cameras					
🔻 🚘 Characters			→ → → Animator		
FirstPersonCharacter Audio				Cylinder	
Drofahe 👻				None (Avatar)	•
There are 2 audio listeners in the scene. Please ensure there is always exactly one audio listen	r in the scene.			1	

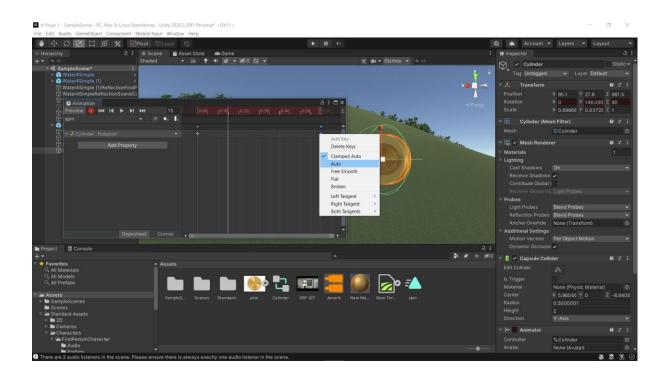
The animation determination process starts by pressing the **red button** - Enable/Disable. Here, the **timetable** and the **Rotation** line in the **Inspector** turn red.



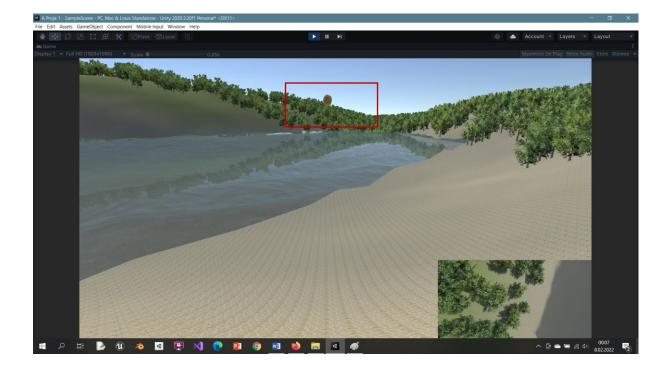
Let's hold the line on the **timeline** and bring it to 60'. 60 frames are equal to one second. The aim is to determine which position it will be in the 60th frame. In the example, it is planned to rotate around 360 degrees. For this reason, a rotation value of 359 degrees is written around the **Y-axis**. The animation can be seen by pressing the play button.

Proje 1 - SampleScene - PC, Mac & Linux Standalone - Unity 2020.3.2011 Personal* <dx11> Edit Assets GameObject Component Mobile Input Window Help</dx11>				σ
	(a)	📥 Account 🕤	🕶 Layers 💌 Laye	out
erarchy a i ill Scene 📾 Asset Store 😎 Game	1	0 Inspector		
≪ All Shaded ▼ 20 ♥ ● 💐 ▼ 🕬 🛱 ▼	At HI - Cirmon - Co. III			
T 43 SampleScene* :		D - Cylinder		
Y Water4Simple     Y     Water4Simple (1)		Tag Untagged		
Water4Simple (I) > Water4Simple (I)ReflectionFirstP	x 🗾 🔁 🖌	🔥 Transform		
Water4SimpleReflexSceneCe			X 95.1 Y 27.8	Z 981.
C Animation 3 i 🗆 🗙	< Persp		X 0 Y 146.093	3 Z 90
Preview 🕘 🚧 🖌 🕨 🖬 🛤 15 0.00, jo.10 j.0.20, jo.30, j0.40, j0.50, 1.00, I	and the second s		X 0.69866 Y 0.03728	ε Z 1
spin 🔹 🔶 🔸 🖡	to she will be set	Cylinder (Me	ob Elltor)	0 7
	A REAL	Mesh		
Cylinder: Rotation	All all all all all all all all all all		Cylinder	
Add Property		🔣 🗹 Mesh Rende		0 7
		Materials		
		Lighting		
			s 🖌	
		Probes		
			Blend Probes	
			None (Transform)	
Dopesheet Curves	방생님님은 것은 것은 것은 것은 것은 것을 가지 않는다.	Additional Settings		
			Per Object Motion	
ject 🗄 E Console			o 🖌	
	å <b>«</b> ★ ø¢9 ,	🔒 🖌 Capsule Coll	lider	0.5
avorites Assets All Materials		Edit Collider		
All Models				
All Prefabs		ls Trigger Material	Name (Discusion Mathematics)	
Assets Google Arama — M		Center	None (Physic Material X 5.96046 Y 0	I) Z -8.9
SampleS Scenes Standard a		Radius	0.5000001	8.5
Scenes		Height		
Standard Assets		Direction	Y-Axis	
Cameras				
Characters		Animator		0 7
🐨 🖬 FirstPersonCharacter				
· 오 버 🖪 10 20 43 10 11 10 10 10 10 10 10 10 10 10 10 10	~		· 📥 🖷 信 中) 23	3:56

There is one last setting to achieve a continuous rotation instead of a **choppy** movement. This is achieved by **right-clicking** on the **anchors** in the first and last frames and selecting **Auto** in the window that opens.



In game mode, our gold will rotate around itself. You can find other types of animations by trying them.



## 6.7. Switching Between Cameras in Unity

In scenes with multiple cameras, various methods can be applied to switch between cameras; **UI Button** applications, keying in camera numbers, navigating between cameras with a keyboard character, etc. In this tutorial, two methods and **two C# code files** will be shared.

## 6.7.1. 1<sup>st</sup> Method

Let's drag the low poly city file we got from Sketchfab to our project. We named the main camera **Camera0** (not required but for clarity). Let's add **three** additional cameras to look at the city from different angles by selecting Camera with the right mouse button in Hierarchy. In this application, the cameras are named **Camera1**, **Camera2**, and **Camera3**. Also, to open an **empty game object** in the same place, let's right click the mouse and select **Create Empty**. This object is named **KameraGecis**.

	eras - Windows, Mac, Linux - Unity 2022.3.14f1 <dx11></dx11>			- 0 ×
	mponent Services Mobile Input Window Help			
ӨКЕ - 🌰				O O Lawer - Default -
= merarchy	📾 🛤 Scene 🗢 Game		0 Inspector	
+ • . N. M	P @Center • @Local • 😳 • 🗒 • 🚧 •	Q + 20 📍 K. 😹 - 🕫 🖩 - 🔂	😭 🖌 Camera0	
T D LowPolyCityCameras		A Contraction of the second seco	Tag MainCamera	
Camera0 Camera1			v 🙏 Transform	• * :
Camera2     Camera3			Position	X 43.08042 Y 6.168493 Z -18.52663
Cameraa Orirectional Light			Rotation	X 43.08042 Y 6.168493 Z -18.52663 X 16.673 Y -152.396 Z 0
Citv1		< Persp		
🕀 KameraGecis				0 7 1
			🔻 💷 🔽 Camera	
				Skybox
			Background Culling Mask	Everything
		THERE AND A REPORT OF A REPORT		
			Projection FOV Axis	Perspective   Vertical
			Field of View	60
				Far 1000
		A COURSE FRAME SOLO		Use Graphics Settings •
				None (Render Texture)
		All All All All All All All All All All		
				Use Graphics Settings +
		Camora0	MSAA Allow Dynamic Resolution	Use Graphics Settings 🔹
				Display 1 🔹
			🔒 🖉 Audio Listener	
■ Project		a 1		
+-		s 🛛 🕹 🖊 🕲 🛪 🕫 1		
C, All Models				
Q, All Prefabs	<b>In In In In #</b>			
🕆 🖙 Assets				
> In Malzemeler In MineModels	Malzemeler MixeModels Scenes Standard CameraDn 🕄 City1 KameraGe			
MineModels Scenes				

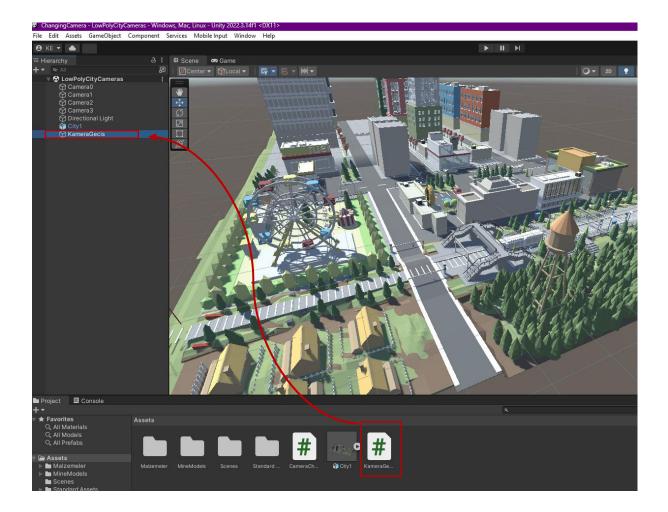
Let's create a C Sharp (C#) file named **KameraGecisleri.cs** with **Create>C# Script** in the Assets section. Now, write the following codes to this file.



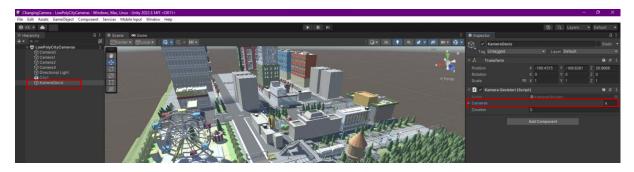
As an editor, if **Visual Studio** is installed with Unity installation, this editor will automatically open when the file is **double-clicked**, and codes can be written. If Visual Studio is not installed, you can open the **KameraGecisleri.cs** file with Notepad, WordPad, etc. and perform the writing process.

```
using UnityEngine;
public class KameraGecisleri : MonoBehaviour
{
  public GameObject[] Cameras; // cameras array
  public int counter = 0;
                                         // the counter variable
  void Start()
  {
    foreach (var item in Cameras)
    {
      item.SetActive(false); //make the cameras passive
    }
    Cameras[0].SetActive(true);
    counter++;
  }
  void Update()
  {
    if (Input.GetKeyDown(KeyCode.Tab)) // change the cameras with Tab key
    {
                                           // make active the current camera
      foreach (var item in Cameras)
      {
        item.SetActive(false); // make the camera passive
      }
      Cameras[counter].SetActive(true); // activate the current camera
      counter++;
                                          // increase the counter by one
      if (counter == Cameras.Length) // if the total number of cameras are reached
      {
        counter = 0; // initialize the counter
      }
    }
  }
}
```

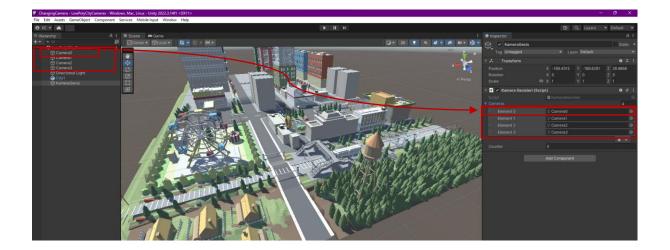
After the **KameraGecisleri.cs** file is written, let's drag and connect this file to the object we opened as **GameObject** and named **KameraGecis**.



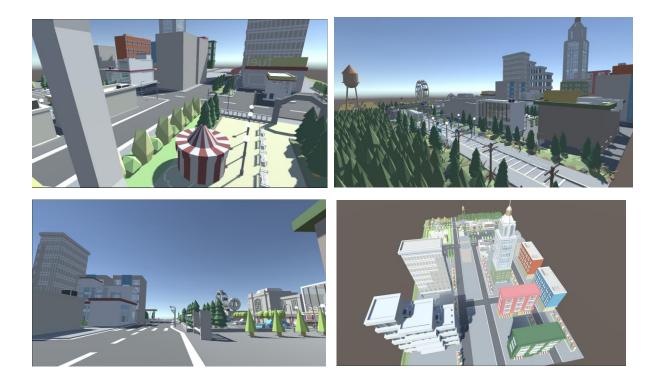
According to the code content, the initial value for **KameraGecis>Inspector>Cameras** is zero. Let's make it **4**.



When the number is set to 4, a camera list will open under **Cameras**. Let's drag and match all our cameras in the **Hierarchy** here so that they match exactly.



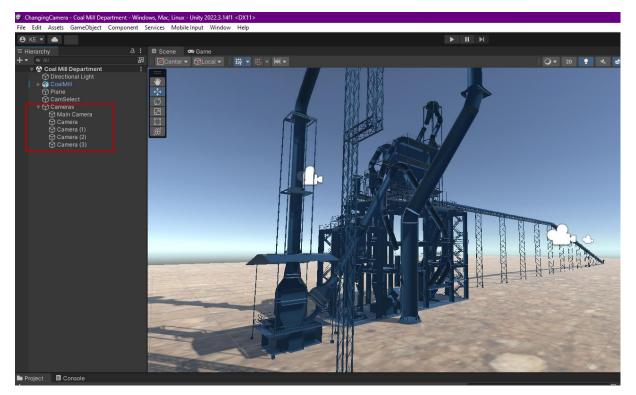
When we run it in **Play** mode, it will now be possible to switch between **4** scenes in order with the **Tab** key.



In this method, the number of cameras is <u>not</u> specified in the C# Script codes. However, the codes are based on the principle of specifying the number of cameras in the **Inspector** after connecting to the empty **GameObject** (KameraGecis) in the **Hierarchy** and dragging and matching the cameras to their places in the **drop-down list**.

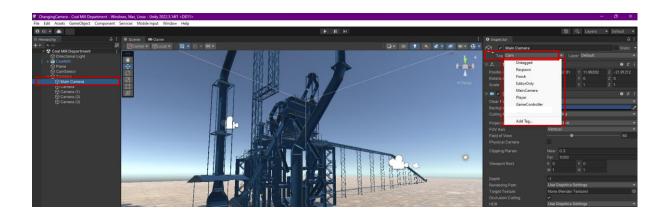
## 6.7.2. 2<sup>nd</sup> Method

In this training application, a mining field facility was used, and a total of **5** cameras with different perspectives were placed. The cameras were dragged under the **empty object** called **CamSelect**, which was opened with **Create Empty** in **Hierarchy**, and gathered under one roof (this is not necessary; they can remain separate).



The **Cam** tag should be given to the **Tag** section of the cameras, as it is defined that way in the codes that will be shared shortly.

To do this, these operations must be performed in order on the cameras in the **Hierarchy**; first, let's select the camera and open its **Tag** in the **Inspector**.



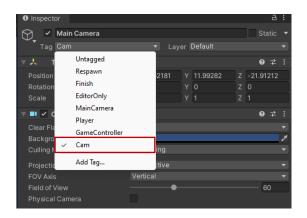
To add a **Cam** tag that does not exist here, select **Add Tag**.

	<li>Inspect</li>	or						а	:
•	$\odot$	Main Camera						Static	٦
	Tag	Cam	-	Laye	Default				Ŧ
	<b>⊽</b> 从 − τ	Untagged					(	9.≓:	:
	Position	Respawn	21	81	11.99282	z	-21.9	91212	
	Rotation	Finish			0		0		
	Scale	EditorOnly							
	▼ ■	MainCamera					6	9 ≓	:
	Clear Fla	Player							÷
	Backgro	I GameController	_						8
	Culling N		ing	9					Ŧ
	Projectio	Add Tag	ctiv	/e					Ŧ
	FOJECUC EOV Avi		Vertical						

Now, define our **Tag** by pressing the **+** key. Write **Cam** here and save it with **Save**.

Inspector		а:
Tags & Layers		❷辈∶
▼ Tags Tag 0		
New Tag Name	Cam Save	+
► L		

Now all our cameras can be given the Cam tag that appears in the Tag list.



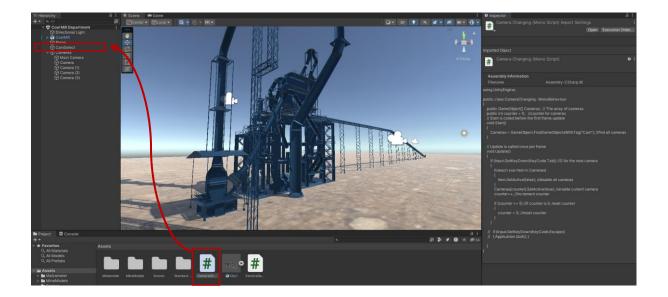
Also, in this application, an empty **GameObject** is created with **Hierarchy>Create Empty** and its name is changed to **CamSelect**.

A C# Script file called **CameraChanging.cs** has been created in the Assets section.

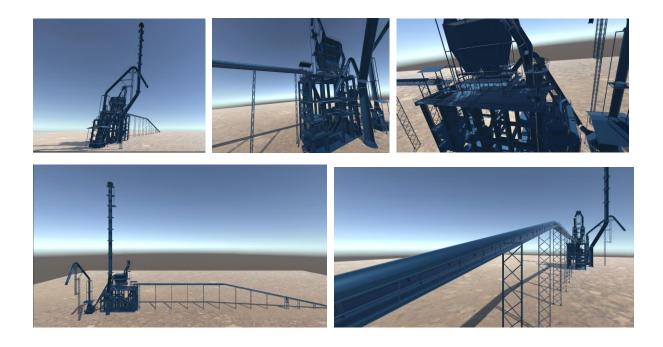
As before, this file was opened in the editor and the following code lines were written.

```
using UnityEngine;
public class CameraChanging : MonoBehaviour
{
  public GameObject[] Cameras;
                                         // cameras array
  public int counter = 0;
                                         // the counter variable
  void Start()
  {
    Cameras = GameObject.FindGameObjectsWithTag("Cam") //find the cameras with Cam tag
  }
  void Update()
  {
    if (Input.GetKeyDown(KeyCode.Tab)) // change the cameras with Tab key
    {
      foreach (var item in Cameras)
                                        // make the cameras passive
      {
        item.SetActive(false); //kameralrı pasifleştir
      }
      Cameras[counter].SetActive(true); // make active the current camera
      counter++; //sayacı artır
                                         // increase the counter by one
      if (counter == 5) // if the counter is 5
      {
        counter = 0; // initialize the counter
      }
    }
  }
}
```

Next, let's drag and drop the CameraChanging.cs code file to our CamSelect object in Hierarchy.



Our application is ready. Now, we run it in **Play** mode.



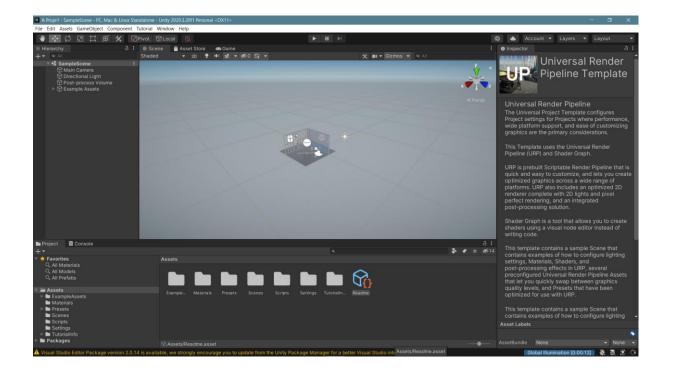
In this application, the codes are based on placing 5 cameras. For fewer or more cameras, the codes will need to be changed. In the first method, there is no limit on the number. In this method, it is not necessary to connect the cameras to the list.

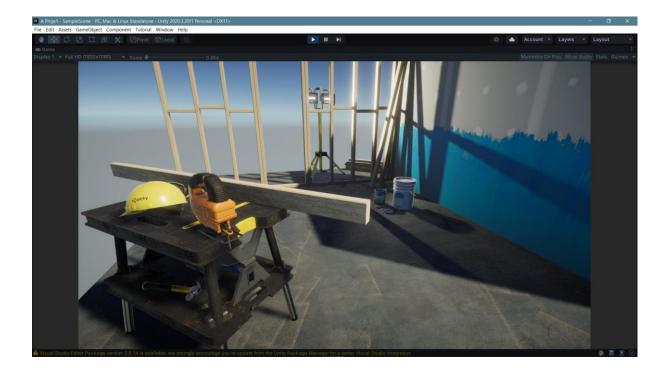
# 7. LIGHT AND TEXTURE WITH URP

# 7.1. Universal Render Pipeline (URP) – Post Processing Volume- Glow Effect

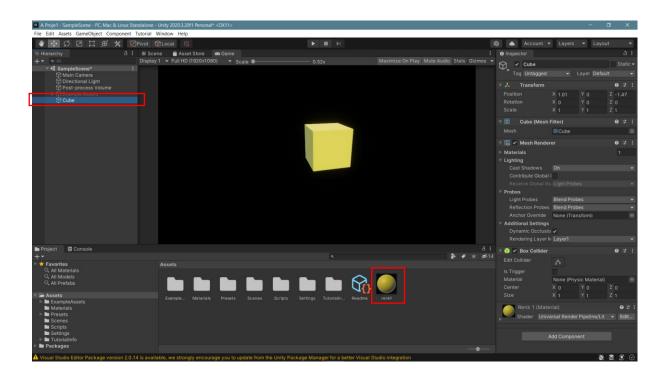
**Universal Render Pipeline (URP)** is a template in which special lighting effects are defined. If the project is planned to use URP, the process will be faster and easier if the relevant selection is made in **Unity Hub**. When the project is opened, a template scene is ready. Scene and game mode images are below.

Unity Hub 2.4.2								- 0	0 ×
🚭 unity						🛆 Hul	o 3.0.1 Available	¢	KE
Projects	Projects	Create a new project with	Unity 2020.3.20/1		-	_ × _	ADD	NEW	-
s Learn	Project Name	Templates			Settings		form	Last Modified 🕆	Q
Community     Installs	ISG1Gama C:\Users\Asus\Docur Unity Version: 2019.1	о́ Ш	0	0	Project Name * A Proje1		latform 👻	3 days ago	:
	VRC2 C:\Users\Asus\Docur Unity Version: 2019.1	2D	3D	High Definition RP	Location * C:\Users\Asus\Documents\A UNITY		latform 👻	10 days ago	:
	AizanoiVR C:\Users\Asus\Docur Unity Version: 2019.1	()	0	0			latform 👻	17 days ago	:
	<b>Aizanoi</b> C:\Users\Asus\Docur Unity Version: 2019.1	Universal Render Pipeline	<b>8№</b> Karting Microgame	<b>(O)</b> AR			latform 👻	20 days ago	:
	VRC1 C:\Users\Asus\Docur Unity Version: 2019.1	0	<u>+</u>	<u>+</u>			latform 👻	23 days ago	÷
	AizanoiGame C:\Users\Asus\Docur Unity Version: 2019.1.14	ifi			CANCEL	CREATE	Platform 👻	24 days ago	:
	AizanoiFPS C:\Users\Asus\Docume Unity Version: 2020.3.20	nts\A UNITY\AizanoiFPS )f1			2020.3.20f1 ~	Current	platform 👻	a month ago	:
	Theresia C:\Users\Asus\Docume Unity Version: 2019.1.14				2019.1.14f1 ~	Current	platform 👻	a month ago	





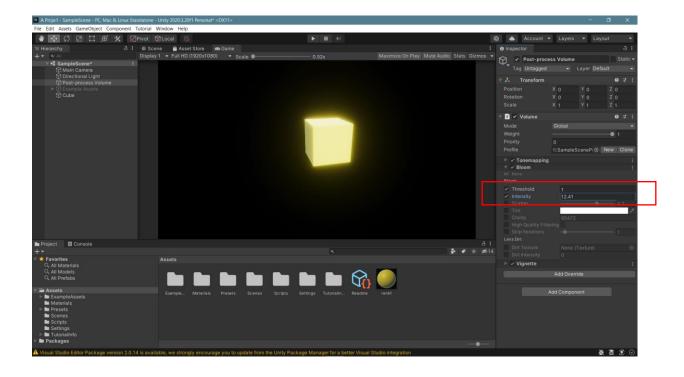
For the **glow effect** example work, let's delete or make passive the **GameObject** named **Example Assets** that creates the workshop seen in the scene. Thus, we can easily apply our design in the preset scene. After Example Assets are closed/deleted, let's add a **cube** to the empty scene. Now, create a **Material** for the cube and give it a yellow color.



The most critical object in the scene is the **Post Process Volume**. The most critical setting on the Inspector is **Bloom**. The Intensity setting under **Bloom** is used to achieve the **glow effect**.

	s Standalone - Unity 2020.3.20f1 Personal* <dx11></dx11>		- o	×
le Edit Assets GameObject Componer	Rivot Clocal		😰 💩 Account 🕶 Layers 👻 Layout	
Hierarchy a	i ∰ Scene	Netletre Or	n Play Mute Audio Stats Gizmos 🔸 👝	
✓ SampleScene* ⑦ Main Camera	Scale	0.52x Maximize On	Tag Untagged  Laver Default	
☆ Directional Light			V         Transform         V         Z           Position         X 0         Y 0         Z           Rotation         X 0         Y 0         Z           Scale         X 1         Y 1         Z	
			Volume 🤅	97
			Profile SampleScenePrØ New	Clo
			► ✓ Tonemapping	
			▼ ✓ Bloom All None	
			Catter Common Co	0.7
			Clamp 65472	
Project E Console			A Lens Dirt	
			Dirt Texture None (Texture)	
* Favorites			► ✓ Vignette	
Q, All Materials Q, All Models				
Q, All Prefabs			Add Override	
🗃 Assets		🗤 🥏	Add Component	
ExampleAssets     Materials     Presets     Scenes     Scripts     Settings			Add Lomponent	
Tutorialinfo Packages				

**Intensity** - when the intensity is increased, the result will be reflected on the screen.



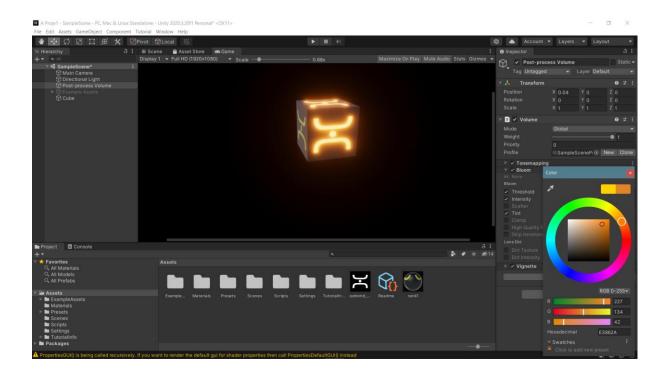
A Proje1 - SampleScene - PC, Mac & Linux Standalor						u x
File Edit Assets GameObject Component Tutoria	al Window Help					
👋 🚱 💭 🖾 🏛 🏶 🛠 🖉 Pivot			¢	🕘 🔄 Account 👻	Layers 🔻 Lay	out 🔻
	Scene 🚔 Asset Store 🛷 Game			Inspector		аı
	olay 1 🔻 Full HD (1920x1080) 👻 Scale 🔍	0.52x	Maximize On Play Mute Audio Stats Gizmos 👻	Post-proces	s Volume	Static -
✓ ④ SampleScene* :				Tag Untagged	<ul> <li>Layer Def</li> </ul>	ault 👻
Directional Light						Corvera and a
Post-process Volume				Transform		071
► ① Example Assets O Cube					X 0 Y 0 X 0 Y 0	Z O Z O
c) subs					× 0 T 0 X 1 Y 1	2 0 Z 1
					AL	
				🔻 🗐 🖌 Volume		07≓ :
					Global	•
						• 1
					SampleSceneP(@)	New Clone
				► ✓ Tonemapping		1
						1
					0.56	
					•	0.7
					65472	
						— 1
Project E Console			a i			
+-			🎝 🛷 🚖 💋 14			0
🔻 🚖 Favorites				► ✓ Vignette		
Q. All Materials Q. All Models	the second second second second second second second second second second second second second second second s		$\sim$	- i vignette		
Q, All Prefabs	i in in in in in				Add Override	_
🔻 🗃 Assets	and the second statements the second statements and the second	in the second second second second				_
🕨 🖿 ExampleAssets					dd Component	
Materials Presets						
Presets Scenes						
E Scripts						
Image Settings Image International Info						
▶ ■ Packages			-•-			
A Visual Studio Editor Package version 2.0.14 is av	vailable, we strongly encourage you to update from the Unity	Package Manager for a better Visual				

The brightness will increase as the **threshold** setting is reduced.

You can try and see which setting will give you the results you want. If you add a JPEG file with a dark background under Assets and add the yellow material to the **Base Map** section, the result will change dramatically.

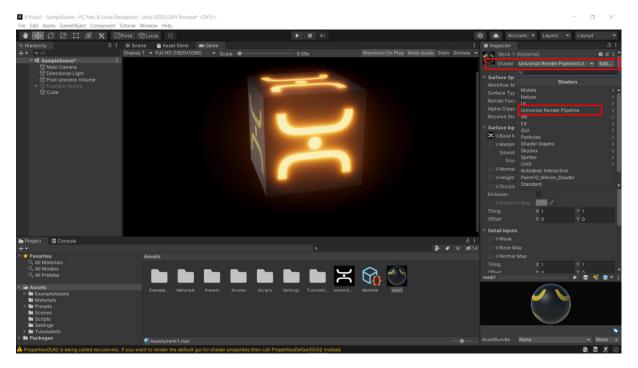
A Proje1 - SampleScene - PC, Mac & Linux Stand	dalone - Unity 2020.3.20f1 Personal* <dx11></dx11>			- a ×
File Edit Assets GameObject Component Tu	itorial Window Help			
🖤 🔂 🖸 🖾 🖽 🗶 💋 PI	ivot 🎯Local 🐻		@ 📥 Ac	count 🕶 Layers 👻 Layout 🔍
TE Hierarchy a I	# Scene Asset Store Store Game		: O Inspector	a :
+ ★ ≪ All I ▼ ∯ SampleScene* I ⑦ Main Camera	Display 1    Full HD (1920x1080)    ★ Sci	le —● 0.68x	Maximize On Play Mute Audio Stats Gizmos * O * Po Tag Un	st-process Volume Static +
		-	▼ <mark>人 Tran</mark> Position Rotation Scale	storm
		X	▼ ■ ▼ Volu Mode Weight Priofity Profile	me  Global Global T O SampleSceneP(O) New Clone
			► J Taner ▼ J Blom All None Blom	1
			<ul> <li>Threaho</li> <li>Intensity</li> <li>Scatter</li> <li>Tint</li> <li>Ctamp</li> <li>High Qu</li> <li>Skip Inte</li> </ul>	12.41 0.7 65472 ality Filtering
Project E Console			A I Lens Dirt	
+-			Dirt Text	
🔻 🚖 Favorites			► ✓ Vigne	COLUMN 2001
Q, All Materials Q, All Models Q, All Prefabs	i ba ba ba		<b>余</b> 〇	Add Override
Assets     Markets     Markets     Markets     Markets     Markets     Scripts     Scripts     Scripts     Scripts     Settings     Markets     Tutorialinfo     Markets     Packages	Example Materials Presets	Scenes Scripts Settings Tutorialin exmond	Readme rent	Add Component
A PropertiesGUI() is being called recursively. If	you want to render the default gui for shader	properties then call PropertiesDefaultGUI() instead		× 2 0

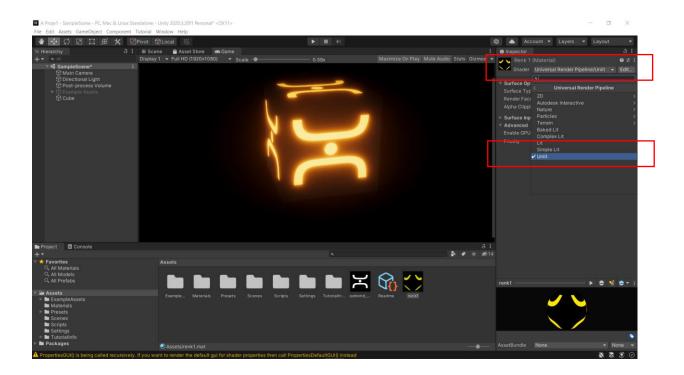
Observe the halo formed around the cube with the tint settings and colors.



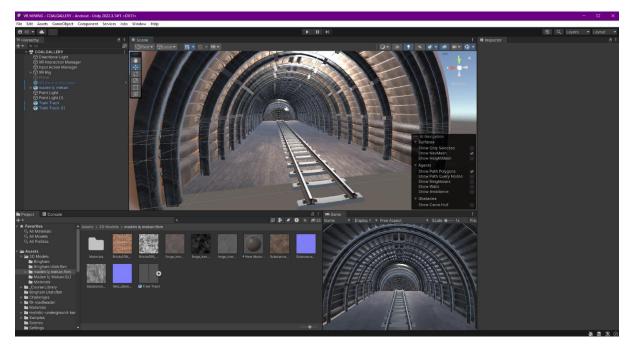
For the **non-glare** surface on the left side;

Let's select the color material. Selection is made in the order **Shader>UniversalRenderPipeline>Unlit**.





Light effects can also be used in a coal mine gallery.

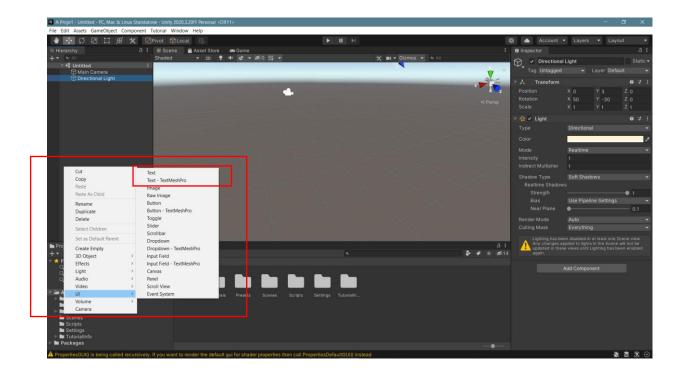


# 8. USER INTERFACE (UI)

# 8.1.UI Text-Button-World Space- Interface Objects-3D Texts

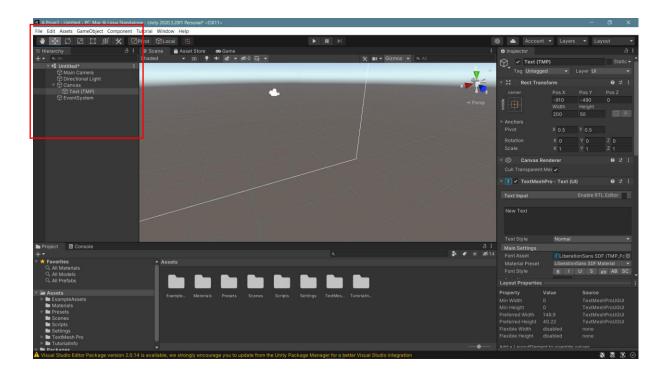
**UI-User Interface-Interface** is an important object that will interact with the user on the screen. Many applications are directed by a menu and its submenus. Therefore, with **UI**, **Text**, **Text-TextMeshPro**, **Image** etc. options will be able to do this design.

When you select **UI** with the right mouse button under **Hierarchy**, the submenu that is connected will open. The most used object is **Text**, and the more qualified text-writing object is Text-**TextMeshPro**.

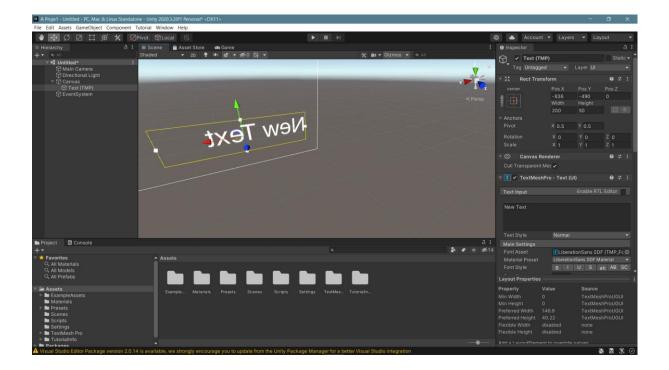


Let's select **TextMeshPro**. Then click **Import TMP Essentials** and **Import TMP Examples & Extras**. The parameters related to writing will open in the **Inspector**.

TMP Importer	: □×				- T.	a x
TMP Essentials					_	
that are essential for using TextMesh Pro. These	xtMesh Pro, as such we need to add resources to your project e new resources will be placed at the root of your project in	н н		🖗 📥 Account 🔻	Layers 🔻 Lay	The sea
		🛠 📰 🕶 Gizmos 💌 🔍 All		Inspector		a :
	ort TMP Essentials			Tag Untagged	+ Layer UI	Static -
TMP Examples & Extras The Examples & Extras package contains additio	ion resources and examples that will make discovering and		4	10 Section and Section 2.		
learning about TextMesh Pro's powerful features same folder as the TMP essential resources.	is easier. These additional resources will be placed in the		× 💌 💽	Center P	nn PosX PosY	Ø ≇ : Pos Z
			< Persp	8 10 5	-910 -490	0
	Ar examples a causa				Vidth Height 100 100	1031161
				Anchors		
				Pivot X	0.5 Y 0.5	A
					X O Y O	ZO
						Z 1
				T 💿 Canvas Rende		0 2 1
				Cull Transparent Met		
				TextMeshPro -	Text (UI)	0 2 1
				Text Input	Enable RT	L Editor
				New Text		
	A second and			Main Settings		
Project B Console			∂i ≱ ≠ ★ ø11		None (TMP_Font Ass	
+ ▼ ▼ ★ Favorites					B I U S ;	ab AB SC
역, All Materials 역, All Models				Auto Size		
Q, All Prefabs						
Assets ExampleAssets	Example Materials Presets Scenes Scripts	s Settings TutorialIn		Property Value		second dimensions
🖿 Materials				Min Width 0 Min Height 0		shProUGUI shProUGUI
Presets Scenes				Preferred Width 0		shProUGUI
E Scripts				Preferred Height 0 Flexible Width disab		shProUGUI
■ Settings ► ■ TutorialInfo				Flexible Height disab		
P ackages						
A PropertiesGUI() is being called recursively. If yo	ou want to render the default gui for shader properties then call	PropertiesDefaultGUI0 instead			r.	0 2 2 2 4



A **frame** that will cover the entire screen, which is actually **2D**, and a text called **New Text** have been created. With an object added to the **UI**, two more objects called **Canvas** and **EventSystem** are automatically added. **Canvas** is the frame of our screen. **EventSystem** is the object related to the operation to be performed. When we come back, we realize that this frame is very large. In fact, the frame is a 2D structure that can be considered independent in the 3D scene.



This 2D frame and text that appears to be huge is actually a small text in the lower left corner of the screen.

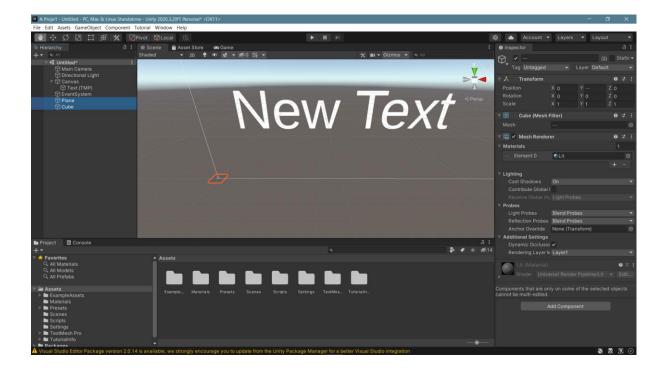
A Projet - Untitled - PC, Mac & Linux Standalo		mai* <dx11></dx11>									9.66	U .
File Edit Assets GameObject Component 1	utorial Window Help											
🕷 🖓 🖸 🖾 🖽 🕷									🖗 📥 Accour	nt 🔻 Layers	▼ Lay	
		Store 🛷 Game							Inspector			
									🕥 🗸 Text (1	MP)		Static 🗸 着
⊽ 🚭 Untitled* i									Tag Untage		Layer UI	01010
Main Camera  Directional Light									rag Untage	jeu 🔹	Layer UI	_
T Canvas									🔻 👯 🛛 Rect Tra	nsform		0 2 1
C EventSystem									8	-836	-490	
									ê 😐			
										×о	Y O	ZO
		_						_	⊽ ⊚ Canvas I			0 2 1
												• +- :
										t Met 🗸		
									🔻 🔳 🖌 TextMes	hPro - Text (U		0 1 1
									Text Input		Enable RTL	Editor
									New Text			
	10.00											
	New Text									Normal		
Project 🗟 Console									Main Settings			
								♣ ♥ ★ Ø14		ELiberat	ionSans SDF	(TMP_Fc⊙
🔻 🚖 Favorites	Assets									Liberation	Sans SDF Ma	iterial 👻
Q. All Materials Q. All Models											USa	b AB SC
Q. All Prefabs									Layout Properties			
						_						
Assets     ExampleAssets									Property Min Width	Value D	Source TextMest	
Materials									Min Width Min Height		TextMest	
🕨 🖿 Presets									Preferred Width		TextMest	
Scenes Scripts											TextMest	
Scripts Settings												
TextMesh Pro												
En TutorialInfo												
A Visual Studio Editor Package version 2.0.14	is available, we strongly	encourage you to u	pdate from the Un	ity Package Mana	ger for a be	tter Visual Studio inter	gration				*	

A Proje1 - Untitled - PC, Mac & Linux Standalo										a x
File Edit Assets GameObject Component 1	Tutorial Window Help	e								
😻 🕂 🗘 🖸 🖾 😫 🗶 🖉	Pivot 🕼Local 🕴						🖗 📥 Account		▼ Laya	
TE Hierarchy a 1	🖶 Scene 📓 Asse	et Store 🛷 Game					Inspector			
+ • 0 • All		20 📍 🗤 😂 🕶 💋			🖬 🕶 Gizmos 📼 🔍 All		A Z Text (TM	P)		Static -
+ • € «An	Shaded •	20 🕈 🗣 🦉 🧟 × 🕉			∎ • Gizmos • € Aı	<persp.< td=""><td>Conter Conter Conter Conter Conter Conter Conter Pivot Rotation Scale</td><td></td><td>Pos Y -490 Height 50 Y 0.5 Y 0 Y 1</td><td>Static • • = = = Pos Z 0 </td></persp.<>	Conter Conter Conter Conter Conter Conter Conter Pivot Rotation Scale		Pos Y -490 Height 50 Y 0.5 Y 0 Y 1	Static • • = = = Pos Z 0 
			2				<ul> <li>✓ ③ Canvas Re Cull Transparent I</li> <li>✓ TextMeshi</li> <li>Text Input</li> </ul>	enderer Me: ✔ Pro - Text (UI		0 ≠ : 0 ≠ :
■ Project 🛛 🛱 Console	<b>H</b>	ewText				ai	New Text Text Style Main Settings	Normal		
+*						🎝 🕈 🔺 💋 14	Font Asset	EL iberati	ionSans SDF	(TMP Fc ⊙
▼ ★ Favorites Q. Ail Materials Q. Ail Models Q. Ail Prefabs ▼ ➤ Assets	Assets		le le le				Material Preset Font Style Layout Properties Property	Liberation	Sans SDF Ma USal	terial 🔫
Parasets     Parasets     Materials     Scenes     Scenes     Scripts     Stripts     TextMesh Pro     TotMesh Pro				TextMes Tutorialin			Min Width 0 Min Height 0 Preferred Width 1 Preferred Height 4 Flexible Width 0		TextMesh TextMesh TextMesh TextMesh none none	iProUGUI iProUGUI
🔳 ମ H 🌛 🛈 🌶	🖳 🖻 (6	刘 💽 😰	🥥 🕺 😫 🛢	. 🖪 🛷 🛛	<u> </u>			Q 🛥 🛥 /	(本 中) 23 8.02。	40 2022 <b>2</b>

Let's add a **Plane** and a **Cube** to our scene.

A Projet - Untitled - PC, Mac & Linux Standard			ATTA								
File Edit Assets GameObject Component					ACCOUNTS AND ADDRESS AND ADDRESS ADDRE			-	atin myanen myanyanyanya	-	
	Pivot GLo	cal 🤴						¢		▼ Layers ▼ I	
		Asset Store							Inspector		
+ * (@ All	Shaded	🔫 2D 📍	111 き * 150	⇒ →		🛠 🖬 🕶 Gizmos 👻 🔤	All		Cube		Static 🔫 🧂
Y     42 Unitited*     I       C) Main Camera     (C) Main Camera       C) Draws     (C) Camvas       C) EventSystem     (C) Plane       (C) Dube     (C) Dube					*		Δ	Persp	Tag Untagged Tag Untagged Position Rotation Scale Cube (Mesh Mesh Cube (Mesh Mesh Mesh Mesh Mesh Cube (Mesh Mesh Cube (Mesh Cube (Mesh Mesh Cube (Mesh Cube (Mesh	Cube rer Lit On Light Probes Biend Probes	lefault ÷ 0 ≠ 1
	100								* Additional Settings		
Project E Console							1.000	aı		p 🖌	
+*							÷ *	★ 14		v Layer1	+
<ul> <li>★ Favorites</li> <li>Q. All Materials</li> <li>Q. All Models</li> <li>Q. All Prefaces</li> <li>★ ExampleAssets</li> <li>★ Materials</li> <li>★ Presets</li> <li>★ Soriests</li> <li>★ Soriests</li> <li>★ Soriests</li> <li>★ Soriests</li> <li>★ Soriests</li> </ul>	Ass E	<b>kits</b> Xample Materials	Presats Sc	senes Scripts	Settings TextMes.	Tutorialin			O Societation     Societa	A None (Physic Mat X 0 Y 0 X 1 Y 1	
TutorialInfo Darkanes											
Visual Studio Editor Package version 2.0.1-	4 is available,	we strongly encour	rage you to update	e from the Unity Pa	ackage Manager for a l	etter Visual Studio Integration					0222

When we pull back, see where these objects are located in the **UI text** structure.



However, in **Game** mode, in the same scene, we understand that the **UI** structure is a 2D object on the screen, while other objects are in 3D structure but in a different size.

A Proje1 - Untitled - PC, Mac & Linux Standale					- u x
File Edit Assets GameObject Component	Tutorial Window Help				
🔹 🕂 🖸 🖸 🖉 🛠 🖾			l l	🖗 📥 Account 👻 Layers	▼ Layout ▼
	# Scene 💼 Asset Store 😎 Game			Inspector	
+ - @ All	Display 1 • Full HD (1920x1080) • Scale •	0.59x	Maximize On Play Mute Audio Stats Gizmos 👻	Q	
♥ 43 Untitled* É 13 Main Camera				Tag Untagged 🔹 L	ayer Default 🔹
Directional Light					
T 🕀 Canvas				Transform	
Text (TMP) CEVentSystem					Y — Z 0 Y 0 Z 0
					Y 0 2 0 Y 1 Z 1
				T III Cube (Mesh Filter)	θ ‡ Ι
					٥
				🔻 📆 🖌 Mesh Renderer	
				▼ Materials	
				Element 0	
				⊤ Lighting	
				Cast Shadows On	
				Receive Global IIIu Light Probe	
				Light Probes Blend Prob	
				Reflection Probes Blend Prob	
	New Text			Anchor Override None (Tran	sform) 💿
	New Text			▼ Additional Settings	
Project 🗟 Console			∂: ► ≠ ∞14	Dynamic Occlusio 🖌	
+ - v * Favorites	▲ Assets	9		Rendering Layer V Layer1	
Q. All Materials	- //558(5			Lit (Material)	<b>9</b> ± 1
Q, All Models Q, All Prefabs		Rea Rea Rea Rea		Shader Universal Render	
T 🛥 Assets				Components that are only on some o cannot be multi-edited.	
Image: Image: Materials					
🕨 🖿 Presets					
<ul> <li>Scenes</li> <li>Scripts</li> </ul>					
Settings					
🕨 🖿 TextMesh Pro					
TutorialInfo Packages					
A Vieual Studio Editor Package version 2.0.1	A is available, we strongly ancourage you to undate from the	Linity Dackana Manager for a better Visual S	tudio interration		× = * 0

Now let's drag the **New Text** to the middle of the stage and place the anchors around it.

A Proje1 - Ulsahnesi - PC, Mac & Linux Standa						
File Edit Assets GameObject Component	Tutorial Window Help					
🖤 💀 🗘 🖾 🎞 🛎 🗶 🖉	ZPivot GLocal 👸				🖗 🔺 Account 🔻 La	iyers 🔻 Layout 🔫
	🖶 🜐 Scene 💼 Asset Store 🛷 Game				Inspector	a I
+ - @ All	Shaded 🔹 20 📍 🕸 😻 🛷 0 🛱 👻		💥 💵 🕶 Gizmos 👻 🔤 All		Text (TMP)	Static 🕶 🧴
♥ S Ulsahnesi* E S Main Camera				Y =	$\Psi_{-}$	Layer UI     ✓
🛱 Directional Light					T :: Rect Transform	0 2 1
♥ ⑦ Canvas ⑦ Text (TMP)		47%			and the second second second second second second second second second second second second second second second	
C EventSystem		8			custom Left	Top Pos Z 2 0
Plane	and the second se			< Persp	§ □ Right	
Cube		1 Λ <u></u>			-5	7 🛄 B
					Anchors	
					Pivot X 0.5	
		- Anna - Anna - An			Rotation X 0	
		NEW BAAL	-		Scale X 1	
	46%	- 3 102 - P	43%		🔻 🕥 🛛 Canvas Renderer	0 2 1
		1 8			Cull Transparent Met 🗸	
					🔻 🔳 🗸 TextMeshPro - Tex	ct(UI)
					Text Input	Enable RTL Editor
					New Text	
					HUH TEAL	
		48			Text Style Norm	nal
Project E Console				a i	Main Settings	
+•				♣ # 1014		perationSans SDF (TMP_Fc⊙
Favorites Q All Materials	Assets					ationSans SDF Material 🛛 👻
🔍 All Models						I U S ab AB SC
Q, All Prefabs					Layout Properties	
🔻 🚘 Assets	The second second second second		the submersion of the second second second second second second second second second second second second second		Property Value	Source
🕨 🖿 ExampleAssets		Scripts Settings TextMe				
Im Materials ► Im Presets						
🖿 Scenes					Preferred Width 146.9 Preferred Height 40.22	TextMeshProUGUI TextMeshProUGUI
<ul> <li>Scripts</li> <li>Settings</li> </ul>					Flexible Width disabled	none
TextMesh Pro						
E TutorialInfo     Deckages						
	14 is available, we strongly encourage you to update from t	the Unity Package Manager for a	a better Visual Studio integration			0 2 2 2 4

Now, move the text to the middle of the stage and change the text and color in the **Text** Input section.

A Proje1 - Ulsahnesi - PC, Mac & Linux Standal	one - Unity 2020.3.20f1 Personal* <dx11></dx11>					- a ×
File Edit Assets GameObject Component 1	Futorial Window Help					
🕷 💀 🖸 🖾 🖽 🗶 🖾				ρ 🔺		
	🖽 Scene 📓 Asset Store 🛛 🕬 Game			i 🕘 Inspe		
+ •   • All		ø60	🛠 📷 🕶 Gizmos 👻 🕾 All		TextMeshPro - Text (U	⊪ <b>e</b> ≢∶*
<ul> <li>♥ Ulsahneai* i</li> <li>♥ Main Camera</li> <li>♥ Directional Light</li> <li>♥ ♥ Canvas</li> <li>&gt; ♥ Text (TMP)</li> <li>♥ EventSystem</li> <li>♥ Plane</li> <li>♥ Cube</li> </ul>		•e		Text In GSF G	ıput örsel İletişim Tasarım	Enable RTL Editor
(T) Cube		GSF Görsel İletişim Tasarım		Font A Materi Font S Font S	iettings sset ELiberat al Preset Liberation tyle B I	vitionSans SDF (TMP.Fc © nSans SDF Material ❤ U S ab AB SC
					Color Gradient de Tags	
				Spacin	ient 📑 🖉 🖥	0 Word 0 0 Paragrapi 0 = 두 드 11 = 4 전 적
Project B Console				A i Wrapp	ing Enabled	
+ -				♣ 🖉 🚖 💋 14 Overfic		
♥ ★ Favorites Q. All Materials Q. All Models	Assets		- 1		ntal Mapping Characte al Mapping Characte	
Q, All Prefabs					Properties	
Der Assets     ExampleAssets     Materials     Materials     Deresets     Scripts     Scripts     Strings     Der TextMesh Pro     Portionalinfo	Example Materials Presets	Scenes Scripts Settings TextMes	futorialin., Ulsahnesi	Property Min Widt Min Heig Preferred Preferred Flexible I Flexible I	h D ht O 1 Width 517.59 1 Height 81.62 Width disabled	Source TextMeshProUGUI TextMeshProUGUI TextMeshProUGUI TextMeshProUGUI none
E Dackanes						
4 Visual Studio Editor Package version 2.0.14	4 is available, we strongly encourage you to u	pdate from the Unity Package Manager for a bett	er Visual Studio integration			😒 🗷 🐼 🛇

The image in game mode will look like this.

The Last care Gard Ciper Convert Last is were in the last in the l	A Proje1 - Ulsahnesi - PC, Mac & Linux Standal	Ione - Unity 2020.3.20f1 Personal* <dx11></dx11>				- o ×
N Horsekty I is som Asket som Okare   V Maketsing   Opport   V Maketsing   Opport    Opport <td>File Edit Assets GameObject Component</td> <td>Tutorial Window Help</td> <td></td> <td></td> <td></td> <td></td>	File Edit Assets GameObject Component	Tutorial Window Help				
	🗶 💀 🗘 🖾 🖾 🌾 🗶	Pivot 🕼Local 🎼			🖗 📤 Account 🔻	
Console     C					Inspector	
Vestisser   O Exores   O Ex					TextMeshPr	o-Text(UI) 60 ≭ i Î
Project     Console     C	Image: Operation of the second se				Text Input	Enable RTL Editor
Project Console   Project Console   Spacing Options (em Characte 0 Word 0   All Models   All Models   All Models   All Models   All Models   Call Models   Eample::::::::::::::::::::::::::::::::::::	🛱 Plane					Normal
Color Gradient: Override Tags: Spacing Options (em. Character 0 Word 0 Alignment: Spacing Options (em. Character 0 Word 0 Vertice Magning: Overtice Magning: Spacing Options (em. Character 0 Vertice Magning: Overtice Magning: Spacing Options (em. Character 0 Vertice Magning: Overtice Magning: Spacing Options (em. Character 0 Vertice Magning: Overtice Magning: Spacing Options (em. Character 0 Vertice Magning: Overtice Magning: Spacing Options (em. Character 0 Vertice Magning: Overtice Magning: Spacing Options (em. Character 0 Vertice Magning: Spacing Options (em. Character 0 Vertice Magning: Overtice Magning: Spacing Options (em. Character 0 Vertice Magning: Overtice Magning:			Tasarım		Material Preset Font Style Font Size	LiberationSans SDF Material   B I U S ab AB SC
						×
Project Console     A sets     Character     A sets     Character     A sets     Character						
Assets     Counting     Assets     Counting     Assets     Counting     Assets     Counting     Assets     Counting     Character     Charact						
Assets     Add Materials     CAll Materials	Project E Console			a i	Wrapping	
Q. All Materials       Periodical Magerials       Periodical Magerials         Q. All Materials       Periodical Magerials       Call Materials         Q. All Materials       Periodical Magerials       Call Materials         Value       Source       Source         Materials       Persents       Scenes         Scenes       Scripts       Settings         Scripts       Source       Min Height         Of Sources       Prefered Height       Site Stripts         Scripts       Source       Min Height       O         Scripts       Source       Source       Min Height       O         Scripts       Source       Source       Min Height       O       TextMeshProUGUI         Fewbles Height       Biologith       Source       Source       Min Height       On one         No tourselevel       Source       Source       Source       Source <td></td> <td></td> <td></td> <td>🎝 🖉 🕆 🖽 14</td> <td></td> <td>Overflow 👻</td>				🎝 🖉 🕆 🖽 14		Overflow 👻
Assets     Bample. Materials Presets Scenes Scripts Settings TextMes. Tutoralin Ukahned     Property Value     Source     Min Width 0     TextMeshProUGUI     Prefered Width 517.59     TextMeshProUGUI     Prefered Width 616.2     Prefered	역, All Materials 역, All Models			4		
ExampleAsets     Marketing     Presets     Marketing     Presets     Schreib     Schr	Q. All Prefabs			$\mathbf{S}$	Layout Properties	
	ExampleAssets     Materials     Presets     Scenes     Scenes     Settings     TextMesh Pro     Proteinlinfo	Esample Materialis Presets Scenes	The second second second second second	and the second second second second second second second second second second second second second second second	Min Width D Min Height D Preferred Width 517 Preferred Height 81.6 Flexible Width disa Flexible Height disa	TextMeshProUGUI TextMeshProUGUI 59 TextMeshProUGUI 52 TextMeshProUGUI abled none abled none
		A is available, we strongly encourage you to undate from the	ne I Inity Package Manager for a better Visual	Studio Integration		

### 8.2.Switch between scenes-UI Button

Let's assume that we added a button to our scene with the **UI Button** and that we want to add a **new Scene** to our project and switch to the new scene with this button. We can see the most practical and fastest way of doing this in an example study.

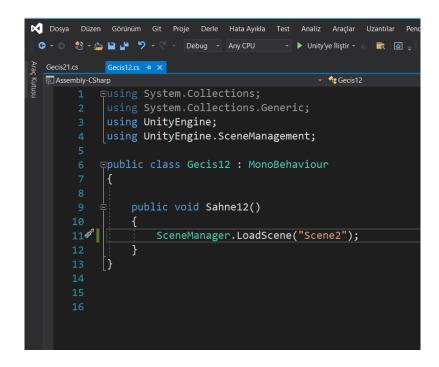
Now, continue without changing the name of our scene named **SampleScene**. Let's add a **UI>Button** here.

The **Canvas** and **Text** section under the **Button** has also been created. Change the **Text** color and size. Also, create a **GameObject** in **Hierarchy**. Here we set its name as **gec12**.

UIButtonScene - SampleScene - PC			y 2020.3.20f1 P	ersonal <	0X11>									- 0	×
File Edit Assets GameObject Cor	mponent Window	/ Help													
👋 💀 🗘 🖾 🖽 🎬	* Pivot										Ø	٠			-
'⊞ Hierarchy													Inspector		a 1
+ - All				0 9 9	11 🛫 🕶 1				* •	II - Gizmos - 🧟			Sample Scene (Sc		÷
<ul> <li>▼ € SampleScene</li> <li>↑ Min Camera</li> <li>↑ Directional Light</li> <li>↑ ⊂ Carwas</li> <li>▶ ⊕ Button</li> <li>↑ ⊕ Button</li> <li>↑ ⊕ Button</li> <li>↑ ⊕ gec12</li> </ul>											< Pers		0		Open
							E	Button1							
Project 🛛 Console												аı			
+*											₽ # #				
<ul> <li>★ Favorites</li> <li>Q. All Materialis</li> <li>Q. All Models</li> <li>Q. All Prefabs</li> <li>★ Assets</li> <li>★ Assets</li> <li>★ Scenes</li> <li>▶ Packages</li> </ul>	Assets														
	Assets/Scene											•	Asset Labels		
A Visual Studio Editor Package ver				rage you	to update f	rom the Unity	ackage Manager	for a better Visual S	Studio Integr	ration				* 2	

Now, create a **C# script** in the Assets section. Here we name it **Gecis12**. When we double-click this object, the **Gecis12.cs** file will open in Visual Studio 2022. The simple coding required for transitioning between scenes is done in the file.

UButtonScene - SampleScene - P	C, Mac & Linux Standale	one - Unity 2	:020.3.20f1 Pers	onal <dx11></dx11>							- o ×
File Edit Assets GameObject Co	omponent Window H	lelp				 					
🕷 🔁 🖸 🖾 🕷									P	٠	Account 👻 Layers 👻 Layout 👻
18 Hierarchy											Inspector     A I
+* « All		naded	👻 2D	? 40 S	▼ 150 菜 ▼		🛠 🖬 🕶 Gizmos	👻 🗣 All			Gecis 12 (Mono Script) Import Set 1
• <b>49</b> SampleScene						Button1			< 90	isp.	Cecis 12 (Mono Script) Import Set E Copen Execution Order Imported Object Cecis 12 (Mono Script) • E Copen Execution Order Imported Object Cecis 12 (Mono Script) • E Cecis 12 (Mono Script) • E Script) • Cecis 12 (Mono Script) • E Using Using System Collections; Using Using System Collections; Using Using Secies Cecis 12 (MonoBehaviour { public ciass Gecis 12 (MonoBehaviour for the Script Analysis (Script) • Cecis 27 (Script Analysis (S
Project E Console						_				aı	
									5 4 3	1 909	
<ul> <li>+ + Favorites         <ul> <li>Q. Al Materials</li> <li>Q. Al Models</li> <li>Q. Al Prefatos</li> </ul> </li> <li>* 2 Assets         <ul> <li>B Scenes</li> <li>B Packages</li> </ul> </li> </ul>	Gecist2 Ge		ample Sc	Kcene2							
	🗃 Assets/Scenes/G										Asset Labels
A Visual Studio Editor Package ve			ingly encourage	e you to upd		ger for a better Visua	I Studio Integration				o 2 2 X



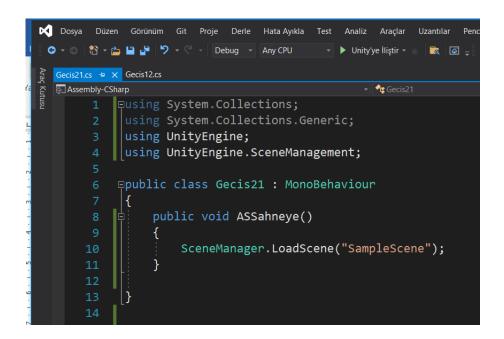
The command to switch between scenes is **SceneManager.LoadScene("Scene2")**; Here we have a second scene, and its name is planned as **Scene2**.

Now let's save this scene and open a new scene under the **File** menu in the project. Name the scene as **Scene2**, as in the coding.

Create a **UI>Button** in the same way. Change its text to **Button2** and color **red**. Add a **GameObject** and name it **gec21**.

UIButtonScene - Scene2 - PC, Mac & Linux Standalor										
File Edit Assets GameObject Component Windo	w Help									
👋 💀 💭 🖾 🎞 🥶 🗶 🖉 Pivot	Cocal							(P)	Account - Layers - Layout	
		🕫 Game							Inspector	
+ • • • All	Shaded	<b>▼</b> 2D	<b>9</b> (44) a	\$ • 1160 聯 •			🛠 🖿 🕶 Gizmos 👻 🔤 All		Scene 2 (Scene Asset)	
✓ Scene2 :								<u>×</u> =	7	Open
G Directional Light ▶ G Canvas G EventSystem G gec21										
	_			_	-	_	_	_		
						Button2				
	1000									
Project E Console								a :		
+*										
🔻 🚖 Favorites 💦 Assets > Scen										
Q. All Materials Q. All Models		-	-							
Q. All Prefabs	#	$\leftrightarrow$	€)							
V 📾 Assets	-									
Scenes Gecist2     Mackages			Scene2							
🔳 🔎 मा 🌛 🛈 🔕	۹ 📮	I 📢 I	2 😰	o 💿	🚯 🔋		5		へ 豆 🌰 🖬 🧖 中) 23:09 9.02.2023	, <b>R</b> 1

Create a C# file named **Gecis21.cs** for this scene and write similar codes in Visual Studio.



The function name is **ASSahneye()**, and our command can be written as **SceneManager.LoadScene("SampleScene")**; That means switch to our first scene.

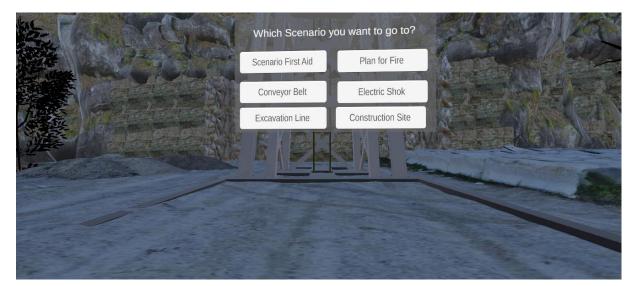
The next step is to connect the codes to the necessary objects. Now, drag and connect the code files to the **gec12** and **gec21 GameObjects**. When we select the **Buttons** in both scenes, let's press the **+** button in the **OnClick()** section in the **Inspector**. Drag the **gec12** object to the **None Object** section. Thus, we have connected our **gec12** object to the **Button** and the codes we connected to it. But another process is to specify which function we want to connect to the button in this code file. To do this, open the **No Function** section, select the name of the code file from the list, and select the **Gecis12> Scene12** function from the function list that opens under that name.

• 0 Z I 🛎 🗙 🖬	Pivot GLocal			II E	0 🔺	Account - Layers	s 🔻 Layout
ierarchy	∂ i ∰ Scene	co Game			1	0 Inspector	
			♥ ● ● \$ ▼ \$60 蒜 ▼	💥 🔳 👻 Gizmos			
¥ 40 SampleScene				4	«Perp	<ul> <li>▼ ✓ Image</li> <li>Source Image</li> <li>Color</li> <li>Material</li> <li>Raycast Target</li> <li>Raycast Padding</li> <li>Maskable</li> <li>Image Type</li> <li>Fill Center</li> <li>Pixels Per Unit M</li> </ul>	Control of the second
						🔻 🗟 🖌 Button	
				Button1		Interactable Transition Target Graphic Normal Color Highlighted Color Pressed Color Selected Color Disabled Color Color Multiplier Fade Duration	Color Tint XButton (Image)
roject 🖪 Console					a i		Automatic
Favorites Assets	> Scenes				<b>\$ \$</b> \$ \$\$9		
A All Materials Q. All Models Q. All Models All Prefabs Assets Scenes Gec	# #	<b>€</b> €	€ ene2			On Click () Runtime (* Geci gec12 ©	
Packages Uec	972 9803601 V	Jampie d'Cas				Shader UI/D	laterial (Material) Default +
						Button -	

Then, do the same thing in the second scene.

File Edit Assets GameObject Component Window Hel					
W I O P I S X Privat GL		<ul> <li>н</li> </ul>	ô 🔺	Account - Layers	✓ Layout ▼
12 Hierarchy 3 1 # 9				O Inspector	a i
	scene oosoame ided	🛠 💵 👻 Gizmos 💌 👁 All		• Inspector	
⊤ 🕄 Scene2 :			V	🔻 🖂 🖌 Image	
Main Camera     Directional Light			Ý 1		UISprite O
🔻 😚 Canvas					1
Button     EventSystem				Material Raycast Target	None (Material) O
C gec21			< Persp	Raycast Padding	×
				Maskable	2
					Sliced +
		4			
		A	-	🔻 📑 🖌 Button	
		Button2			Z
		Bullonz		Transition Target Graphic	Color Tint 👻
				Normal Color	⊠Button (Image) ⊙
				Highlighted Color	1
					1
					1
				Disabled Color Color Multiplier	· · · · · · · · · · · · · · · · · · ·
				Fade Duration	0.1
Market Annual Market States					Automatic -
■ Project  □ Console ++			∃: ₽ ★ ∲\$		Visualize
Favorites Assets > Scenes				On Click ()	
Q All Materials Q All Models				Runtime (+ Geci	
Q All Prefabs # #				Runtime (♥ Gecr	szt.ASSanneye •
T Assets				a good to	
Scenes Gecis12 Geci	is21 SampleSc Scene2				
🕨 🖿 Packages				Default UI M	aterial (Material) 🛛 🕄
				► Shader UI/D	
					mponent
				Button +	
A Visual Studio Editor Package version 2.0.14 is available	e, we strongly encourage you to update from the Unity Package	e Manager for a better Visual Studio integration			O 25 25 26

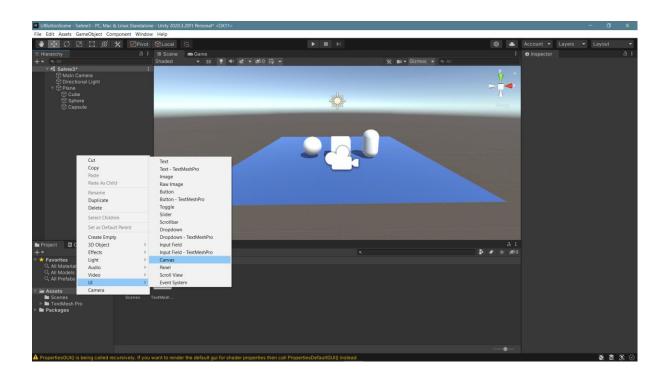
When the game starts, when **Button1** is pressed, we move to the second scene; when the second scene is pressed, and **Button2** is pressed, we move to the first scene. The scenes are very simple and only contain **UI>Buttons**. However, the purpose is to see the transition. After the scenes are designed for the purpose, it is possible to switch between many scenes. The design of buttons and interfaces has become a field of expertise. In this example, the operation of the process is discussed in its simplest form.



In a **mining project**, a more comprehensive menu driven application is given below.

### 8.3.UI Text – World Space

This application is for positioning **UI Texts** inside our scene. Normally, **Canvas** and **Text**, **Button**, **Image**, etc. subcomponents, which cover the whole screen due to their pixel structure and are very small next to our 3D scene, can be included in the scene with the size and positioning we want. Let's start by adding **Plane**, **Cube**, **Circle**, **Capsule** and **UI>Canvas** to our scene...



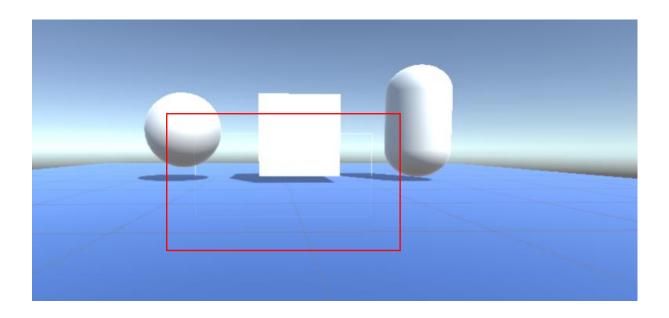
When we select UI>Canvas, change the setting in Inspector>Canvas>RenderMode to WorldSpace.

UIButtonScene - Sahne3 - PC, Mac File Edit Assets GameObject Cor			3.20f1 Personal*	<dx11></dx11>								-	o x
			10		_	▶ 11	1991		ô 🔺	Account 👻	11466600	1031 INCOM	
👑 🔂 🖸 🛄 🌐		E BLocal				F 11				Account	Layers	▼ Layou	ut 👻
+ + a All		Shaded		9 44 2	•ø60 ∰ •			🛠 🛍 🔻 Gizmos 👻 🔤 All					-
v € Sahne3*		1								🛱 🗸 Car			Static 🗸
C Main Camera									V	A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF	tagged -		597m
🛱 Plane									>†4	1 (B) (S) (S)	Transform		071
Capsule							sille		1	Some values			
Cube						1	AUDE						Pos Z 0
Canvas CentSystem		Summer of the local division of the local di		_	and the second value of th								U
() Evenisystem													11 R
										► Anchors			
										Pivot			
						r				Rotation			Z O
										Scale			Z 1
										🔻 🔳 🖌 Cany			0 ‡ i
										Render Mod			ce - Overla <del>-</del>
												en Space - O	
										Sort Orde Target Di		en Space - C Id Space	amera
										Additional S			
										⊤ 📧 🗸 Cany			011
										UI Scale Mo		Constant Pix	
				_		- / ·							Kel GIZe (*)
Project Console									a:				
	Assets								• • • •				
Q. All Materials										🔻 江 🗹 Grap			021
Q. All Models Q. All Prefabs										Script Ignore Reve			aycaster ⊙
										Blocking Ob		Norre	
Assots Scenes										Blocking Ma		Everything	•
TextMesh Pro													
Packages													
A				-									
A PropertiesGUI() is being called re	ecursively. If you	r want to rende	er the default gu	i for shader pr	operties then call	PropertiesDefaultGl	JII) instead					1	0 2 2

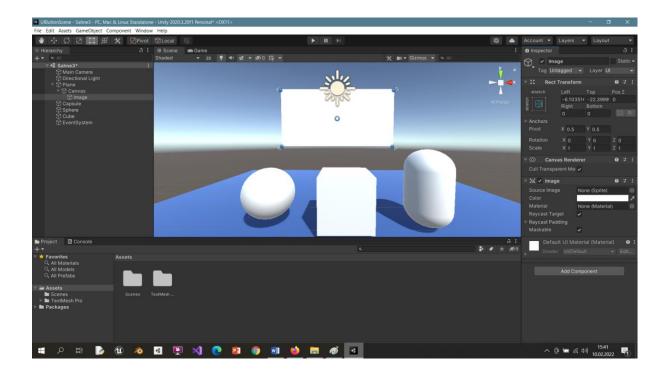
We can see how small our Plane, Cube, Circle and Capsule objects are.

UIButtonScene - Sahne3 - PC, Mac & Linux Standal	one - Unity 2020	3.20f1 Personal*	<dx11></dx11>					- a x
File Edit Assets GameObject Component Wind	ow Help							
🖤 🕂 🗘 🗹 🎞 🤀 🛠 🖉Pivo							φ 🔺	Account 🕶 Layers 👻 Layout 💌
								O Inspector a 1
+ ★	Shaded	<b>▼</b> 2D	<b>9</b> 40	き • 150 蒜 •		🛠 🖿 👻 Gizmos 🔹 🖭 All		😭 🗸 Canvas 🔤 Static 🗸
							¥ =	Tag Untagged - Layer UI -
🛱 Directional Light								▼ :: Rect Transform ● 같 :
♥ ♀ Plane ♥ Canvas								Reset
Capsule							< Persp	Move to Front
Sphere     Cube								Move to Back
EventSystem								Remove Component
								Anchors Copy Component     Pivot Paste Component As New
								Turke component to their
		-	-	_	_		_	Rotation Paste Component Values Scale Find References In Scene
								V III V Can Properties
								Reset Position
								Reset Rotation
								Keset Scale
								Copy Position Paste Position
								Sorting Copy Rotation
								Order in Paste Rotation
								Additional Copy Scale
								Paste Scale
Project E Console							а :	UI Scale M Paste World Placement
+*								Tota Rentering
🔻 🚖 Favorites Assets								Dynamic Pixels Per L 1 Reference Pixels Per 100
Q. All Materials Q. All Models								
🔍 All Prefabs								ㅋ 그 🖉 Graphic Raycaster 🛛 🤤 랴 i
Assets								Script ∷ GraphicRaycaster ⊙ Ignore Reversed Graj ✓
Scenes Scenes								Blocking Objects None •
TextMesh Pro Packages								Blocking Mask Everything +
								Add Component
A PropertiesGUI() is being called recursively. If yo	want to rend	er the default o	ui for sha	der properties then call Pror	pertiesDefaultGUI0 instead			0 2 2 2 4

To move our **Canvas** from being an object covering the screen to an in-stage plane, bring the **Canvas** under the **Plane** to the **Parent-Child** position and provide coordinate unity with the **RectTransform>ResetPosition** operation. Then, reduce the **X**, **Y**, **Z** values in **RectTransform>Scale** to fit our stage; for example, **0.03**. When we focus on the **Canvas**, we can see its location and size.



Now we can move the **Canvas** to the point we want and add **Text**, **Image**, **Button**, etc. to it. Let's continue the example by adding **UI>Image** under **Canvas** after positioning. If we want the **Image** to cover the **Canvas**, we can bring it to the size we want with the **Stretch** or direct scale button under the **Anchor**. The initial opening color of the **Image** is white.



Change the color and transparency via **Color**. It's time to add text. Let's add **UI>TextMeshPro** to Canvas. We wrote **GSF GIT** instead of *New Text* in the **Text** section and made size and position settings.

UIButtonScene - Sahne3 - PC, Mac & I	Linux Standalone - Unity 2020.	3.20f1 Personal* <dx11></dx11>					- a ×
File Edit Assets GameObject Comp	onent Window Help						
👋 🕂 🗘 🗹 🔝 🏽 🏶 🦎	K Pivot GLocal						
'⊞ Hierarchy						Inspector	
+* @ All	Shaded	▼ 2D 🤋 🕪 🛫 死0 蒜		🔆 💵 👻 Gizmos: 💌 👳 All		Text (TMP)	) Static <del>v</del> 着
♥ € Sahne3*	1				¥, a	Tag Untagged	
Directional Light			Nº1			T :: Rect Transfo	
♥ Plane ♥ ♥ Canvas						custom Left	Top Pos Z
🗇 Image			200			E	47: -5.69695 0
⊢ 🗘 Text (TMP) ⑦ Capsule						별 🛄 Right	
C Sphere			GSE GIT				917 4.52308:
Cube EventSystem						P Anchors	
C Eventsystem						Pivot X 0.5	
						Rotation X o	
						Scale X 1	
						🔻 💿 🛛 Canvas Rend	
						Cull Transparent Me	×
						🔻 🍸 🗹 TextMeshPr	o - Text (UI) 🥹 로 🗄
							Enable RTL Editor
			Text Input	Enable KTL Eultor			
						GSF GIT	
Project 🛛 Console					ai	Text Style	Normal 🔫
+*					× ∉ ± 159	Main Settings	
🔻 👷 Favorites 🛛 🗛	Assets						ELiberationSans SI O
Q. All Materials						Material Preset	LiberationSans SDF M+
Q, All Prefabs							B I U S ab AB SC
V 🖬 Assets							72
Scenes							-
TextMesh Pro Packages							Min Max WD% Line
r 🖬 Fackages							1
						Layout Properties	
A PropertiesGUI() is being called recr	ursively. If you want to rende	er the default gui for shader properties then	call PropertiesDefaultGUI() instead				· • 2 2 2 2

Now, check the image in **game** mode.

<ul> <li>UIButtonScene - Sahne3 - PC, Mac</li> </ul>	& Linux Standalor	ne - Unity 2020	.3.20f1 Personal* <dx< th=""><th>11&gt;</th><th></th><th></th><th></th><th>- o x</th></dx<>	11>				- o x
File Edit Assets GameObject Co	mponent Windo	w Help						
👋 🕂 🗘 🖾 🛄 😸		Cal				0 🔺		s 🔻 Layout 👻
12 Hierarchy			🕫 Game				Inspector	
+ • • • All			▼ Free Aspect	▼ Scale —●	1.2x	Maximize On Play Mute Audio Stats Gizmos 🕶	🕞 🗸 Text (TMF	) 🗌 Static 🛨 着
▼ 4 Sahne3* ☆ Main Camera							Tag Untagged	
🛱 Directional Light							🔻 👯 🛛 Rect Transf	
👻 💬 Canvas 🗇 Image								
⊕ mage ► ⊕ Text (TMP) ⊕ Capsule							별 🔲 Right	
💮 Sphere								6917 4.52308:
Cube					GSF GIT		► Anchors Pivot X 0.1	5 Y 0.5
C) Eventsystem					GSF GIT			
					the second second second second second second second second second second second second second second second se			
			and the second second				T 💿 Canvas Ren	
								0: 🗸
							TextMeshP	
				-			Text Input	Enable RTL Editor
							GSF GIT	
Project E Console							Text Style	Normal 💌
+*						🎝 🛷 🌟 🕫 959	Main Settings	
🔻 🚖 Favorites								ELiberationSans SI O
All Materials     All Models							Material Preset Font Style	LiberationSans SDF M + B I U S
Q. All Prefabs								ab AB SC
🔻 🖬 Assets								72 AB SC
Scenes							Auto Size	
TextMesh Pro Packages								s Min Max WD% Line
								1
							Color Gradient Override Tags	
						<b>_</b> _	Layout Properties	
A PropertiesGUI() is being called n	cursively. If you	want to rend	er the default oui for	shader properties then call P	ropertiesDefaultGUI() instead			0 2 2 2 4

To add text to the scene, select **3DObject>3DText** in **Hierarchy** and place the text in the desired location on the scene. The resolution of the text can be increased by decreasing the **Character Size** and increasing the Font Size.

			.20f1 Personal*	<dx11></dx11>							a x
File Edit Assets GameObject Cor	mponent Window	Help									
👋 💀 🗘 🖾 🖽 📽	🛠 🖉 Center 🛛							P	Account 👻 Layers	▼ Layou	it 👻
' Hierarchy									Inspector		a :
+ • • • All		Shaded	·▼ 2D	📍 🕪 🛫 🕫 0 🗄	₩ <b>-</b>		🛠 🖬 🕶 Gizmos 💌 🔤 All		New Text		Static 🕶 🔒
✓ Sahne3*	A 1 Booked ● 20 ● 0 me http://www.feetings. Breaked ● 20 ● 0 0 me breaked ● 20 • 0 me										
∀ Ô Plane ∀ O Canvas											and an international statement
💬 image						Hello Work		-Penne			Z O
🗇 Capsule											
Cube						salles				er	0 7 1
EventSystem     New Text					_	A CONTRACTOR OF THE OWNER					1
		dour Help the Channe Shaded 20 4 2 4 2 5 4 2 5 4 2 5 4 2 5 2 5 2 5 2 5									
					-						
									♥ Probes		
										None (Transf	torm) 💿
										Der Ohing M	
Project 🖾 Console								a :			otion +
+*								♣ # # # #99		·	
🔻 🚖 Favorites									🔻 👖 🛛 Text Mesh		0 2 1
Q All Materials											
Q. All Models Q. All Prefabs											
V 🛥 Assots											
Scenes Enterthese Provide Control Science Pr		xtMesh									
Packages											
											*
			Image: Constraint of the second se								
											A i Static ← fault ← fault ← fault ← 2 2.5325 Z 0 Z 1 0 ↓ i 1 0 ↓ i 1 0 ↓ i 2 .5325 Z 0 Z 1 0 ↓ i 2 .5325 Z 0 Z 1 0 ↓ i 2 .5325 Z 0 Z 1 0 ↓ i 2 .5325 Z 0 Z 1 0 ↓ i 0 ↓ i 2 .5325 Z 0 Z 1 0 ↓ i 0 ↓ i ↓ i ↓ i ↓ i ↓ i ↓ i ↓ i ↓ i
									Politistyle		
PropertiesGUI() is being called re	cursively. If you wi	ant to render	r the default g	ul for shader properties th	hen call Proper	rtiesDefaultGUI() instead				*	2 2 0

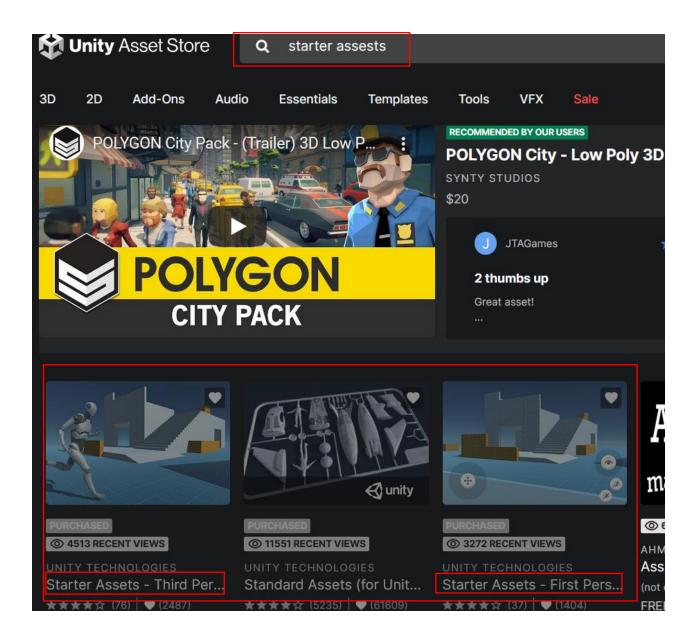
Font types, colors, sizes, and locations are up to the designer. Check it out in **Game** mode, too.

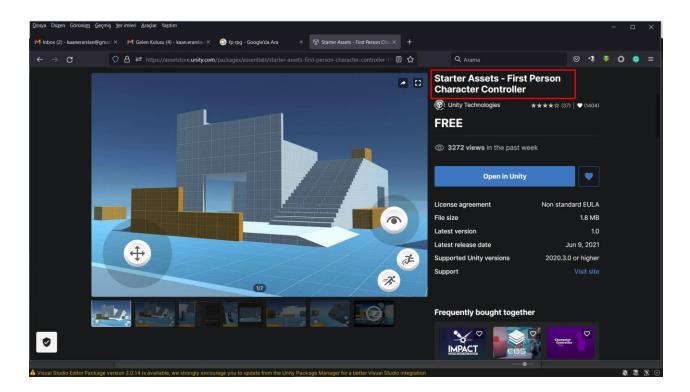


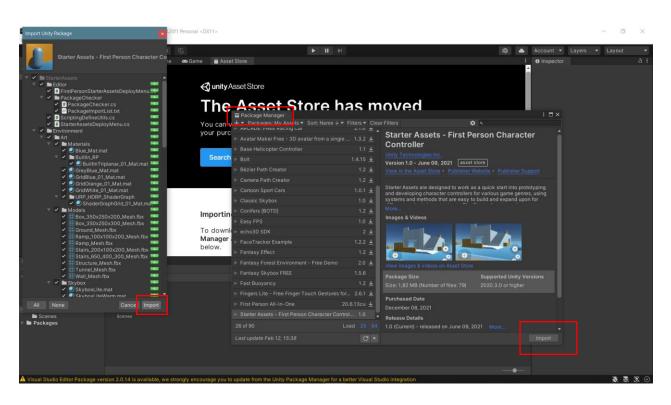
# 9. FPS VE TPS APPLICATIONS

## 9.1.FPS-First Person Shooter and RPG-Role Playing Game

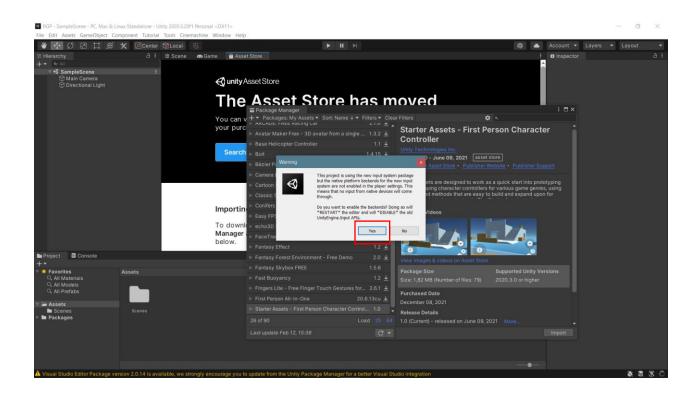
If the game has a plot where you play through the eyes of the game hero and in his shoes, this type of game is called **FPS-First Person Shooter** or **FP-RPG-First Person Role Playing Game**. For the application, we can search for **Starter Assets**, which is offered free of charge by **Unity Technologies** in **the Unity Asset Store**. In this way, *Standard Assets* was previously downloaded. Now, there are two more packages on the list; **Starter Assets First Person** and **Starter Package Third Person**. Now, add the **First Person** one to our assets and import it from the **Package Manager**.

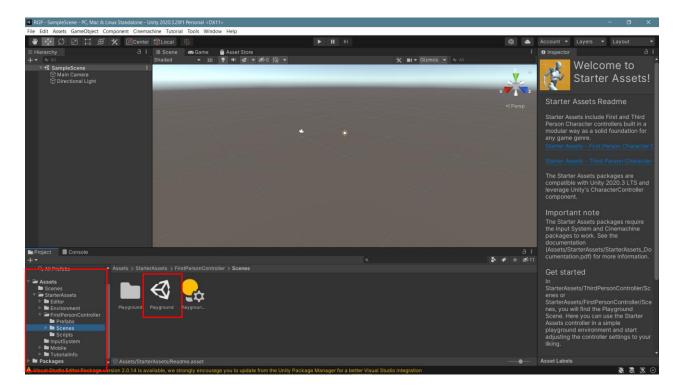




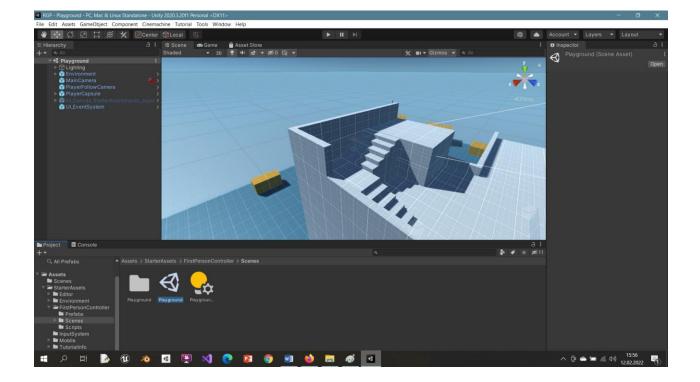


After this process, a confirmation window will open to **re-adjust** Unity settings. After saying **Yes**, Unity will be closed and automatically **reopened** with its new settings. Standard Assets are visible in our Assets window. Now, let's go to **Starter Assets>FirstPersonController>Scenes** and see the **PlayGround** scene below it.



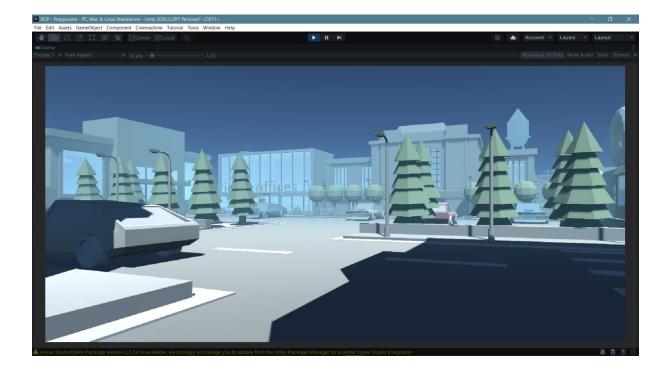


When we double click on the **PlayGround** scene file, our new scene with all its edits will appear on the screen.



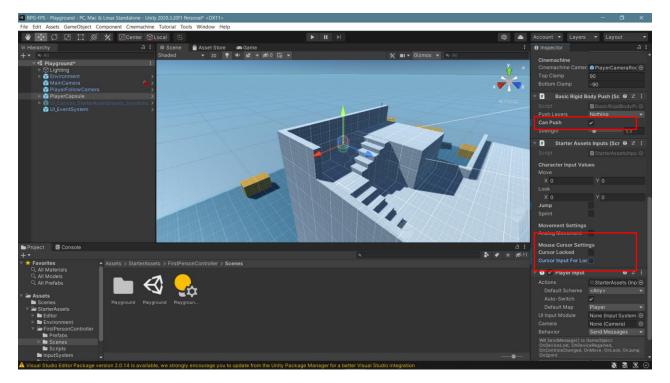
When the game is launched, we will see that we can move around the stage with the arrow and shift keys and the mouse, look and move in all directions, and jump with the spacebar.

It is possible to create or add our own scene instead of the **Environment** that creates the scene. In this way, we start to move within our own design. By taking a Low Poly city model from the Sketchfab site to our scene and making the **Environment** object passive, it became possible to walk around the city.

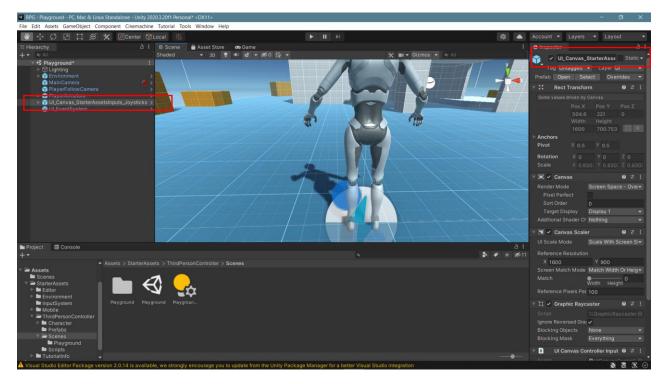


It is also possible to use this application on mobile devices. However, since there are no keys like on the keyboard on mobile devices, these controls will be made with keys called virtual **TouchPad** or **joystick**.

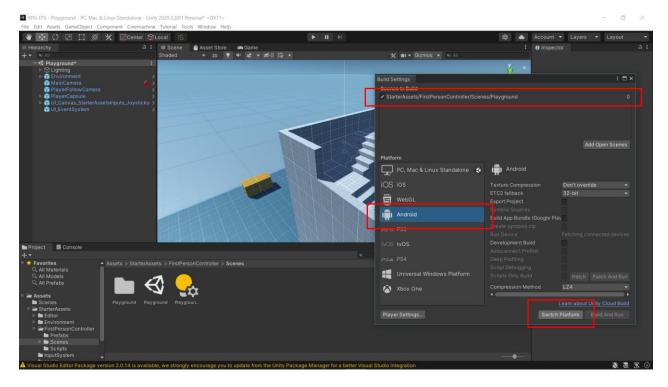
First, let's go to the **Movement Curser Settings** setting in the **PlayerArmature Inspector** section and **uncheck** the boxes. Check the **Can Push** box in the **Basic Rigid Body Push** section so that it can push objects.



Now, activate the **UICanvasSaterterAssetsIputs\_Joysticks** object which is previously passive in the **Hierarchy** section.

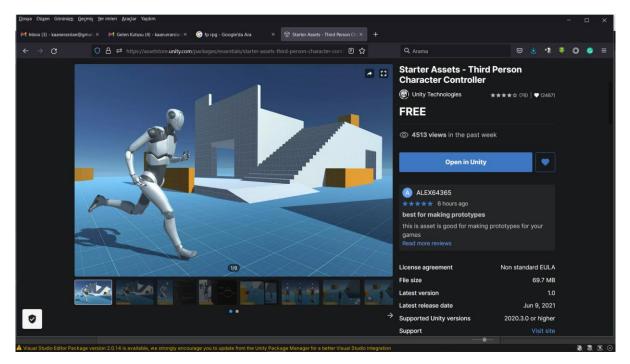


Control is now enabled with the added **joystick**, <u>not</u> the **keyboard**. In this way, by going to **Build and Settings**, it will be possible to convert it to **Android** or IOS application according to the **phone-tablet** type and play it with the **joystick** on mobile devices.



# 9.2.TPS Third Person Shooter-Starter

A **Third Person Shooter** or **Third Person Role Playing Game** type application is a type of game setup where we follow the player and see their movements in the form of animations. Let's follow a similar path for this. This time, **Starter Assets - Third Person** will be downloaded and imported from the Asset Store.

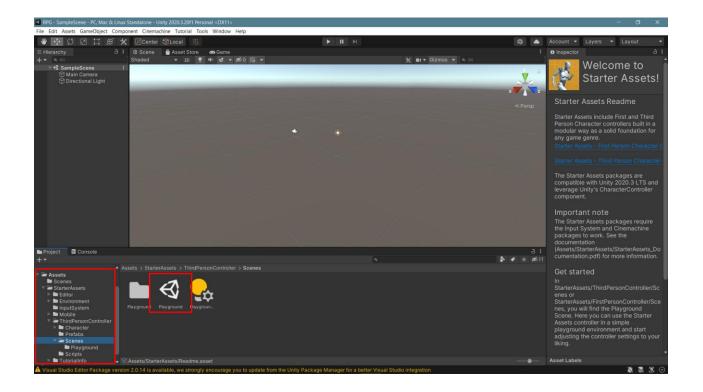


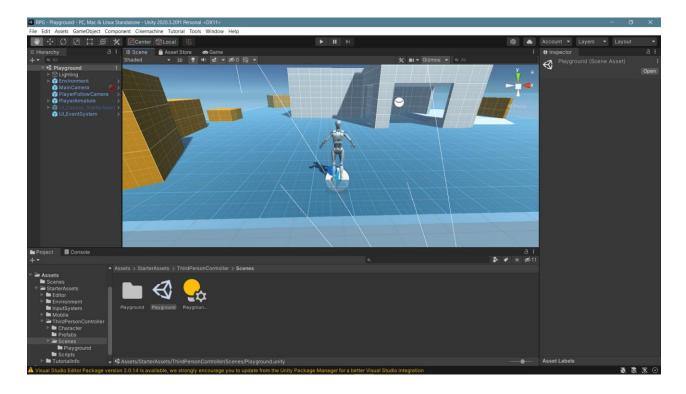
Import Unity Package	Ausdam       Package Manage       Package Manage       Package Manage       Package Manage         Ausdam       Package Manage       Package Manage       Package Manage       Package Manage         Ausdam       Package Manage       Package Manage       Package Manage       Package Manage         Ausdam       Package Manage       Package Manage       Package Manage       Package Manage         Ausdam       Package Manage       Package Manage <td< th=""></td<>			
Starter Assets - Third Person Character Ci. 1 Store			2 III + Gizmos + 4-All	ector A : Package 'Starter Assets - Third Person Edit View in Package Manager mation
✓ # ScriptingDefineUtils.cs			i c	× <sup>1.0.0</sup>
ThirdPersonStarterAssetsDeployMenuted	+ ▼ Packages: My Assets ▼ Sort: Name ↓ ▼ Fi		ar Filters 🗢 🤉	
			Starter Assets - Third Person Character	Account Layout     Account Layout
🔻 🖌 Materials 🔤	⊨ Simple City pack plain		Accout Layout   Layout   Image: Layout   Package: Starter Assets - Third Package:   Image: Image:   Table:   Image: Image:   Table:   Image: Image:   Image:   age: Imag	
				ing character controllers for
✓ ✓ ④ BuiltInTriplanar_01_Mat.mat	► Simple VR	1.0.2 ±		that are easy to build and
of the yarde junction at the second sec	► Sketchfab for Unity v0.0.85a (i	alpha) 🕹		
🖌 🥥 GridOrange_01_Mat.mat	Skybox Series Free	4.2 🕤	Starter Assets are designed to work as a quick start into prototyping	
	► Snow Mountain		and developing character controllers for various game genres, using	backages require the Input System
	► Space Robot Kyle	1.0 4		is will automatically install when
✓ ✓ Models ✓ ■ Box_350x250x200_Mesh.fbx				
				1000000 C
Ground_Mesh.fbx				EDUDDORY CONSIST.
				npty
Stairs_650_400_300_Mesh.fbx				
		_	View Images & videos on Asset Store	
Wall_Mesh.fbx			Package Size Supported Unity Versions	
			Purchased Date	
All None Cancel Import				
			Release Details	
Packages			1.0 (Current) - released on June 09, 2021 More	-
A Visual Studio Editor Package version 2.0.14 is available, we strongly encourage you	to update from the Unity Package Manager for a bett	er Visual St		0223

Similarly, you will be asked for **restart** confirmation for the settings to take effect, click **Yes** and the project will be closed and **restarted**.

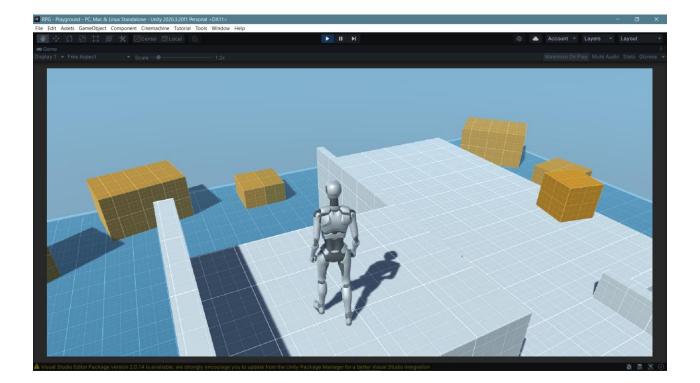
RPG - SampleScene - PC, Mac & Li									a ×
File Edit Assets GameObject Co	the second second second second second second second second second second second second second second second s	I Tools Cinemachine Window H	elp	► II ►		ô 🔺	Account •	Layers 💌 Lay	vout 👻
	I III Scene	🖨 Asset Store 🛛 📾 Game				:	Inspecto		A :
+ * [ @r All	Shaded	≠ 20 📍 📫 😎 ≠ 9	60 瑛 ▼		🔆 🖬 🕶 Gizmos 👻 💁 🛝	1	Pack		
▼ SampleScene	-			_		x T z	Informati Display r		
	1		■ Package Manager     +      +      Packages: My Asset	s ▼ Sort: Name ↓ ▼ Filters ▼ Clea	or Filters	¢ <	: = ×		
					Starter Assets - Th	nird Person Charact	er 👘		
					Controller				
			Simple V Warning Sketchfa		- June 09, 2021 Asset Store • Pub	asset store			
			Skybox 1  Snow Me  Space Re  SpeedTu  SplineMe	This project is using the new input system but the native platform backends for the system are not enabled in the players sett means that no input from native devices through. Do you want to enable the backends? Doi "RESTART" the editor and will "DISABLE UnityFignine.Input aPIs.	new input ings. This will come ng so will	work as a quick start into proto trollers for various game genres, easy to build and expand upon	using for		
			<ul> <li>Spline PI</li> </ul>	Yes	No				vork as a s for terms and ind ind ind ter Assets Jointy terms to the sense ind the sense tall when kage into
			Sport Car - 3D model	1.0 单	~~~				
Project 🖬 Console			SpriteLights	3.1 ±					
+•					View images & videos on Asse				
Favorites Q All Materials					Package Size Size: 69.72 MB (Number of file	Supported Unity Ve s: 110) 2020.3.0 or higher	rsions		
Q, All Models Q, All Prefabs									
					Purchased Date				
Assets Scenes	Scenes				December 04, 2021 Release Details				
Packages			All 90 packages shown	~~ · · · ·	1.0 (Current) - released on Jur	ne 09 2021 More			
			Last update Feb 12, 16:58						
									Vork as a fis for harmo and and writer Assets Juity System work. The son Statter purchased into Version
🖷 오 🖽 📄	ف 10	a 🖳 刘 📀	1 🧿 🙍 😆	) 🛅 🚿 🖪			^ @		7:01

In the package that comes to the **Assets** section, we will just need to **double-click** the **Playground** scene in **Assets>Starter Assets>Third Person Controller>Scenes**.



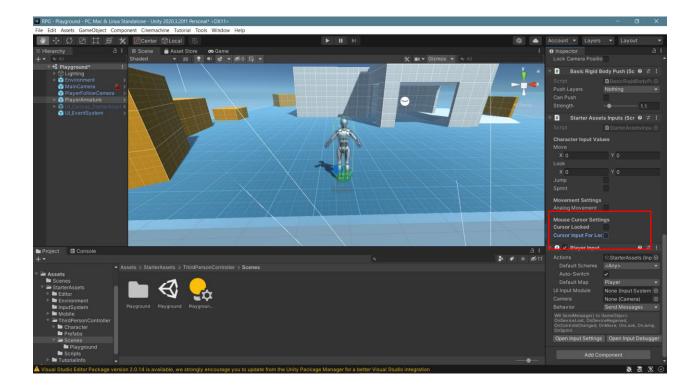


The player being followed on stage is a robot. When the game starts, it will be seen that we can control the movements with the arrow keys, shift, space bar and mouse.

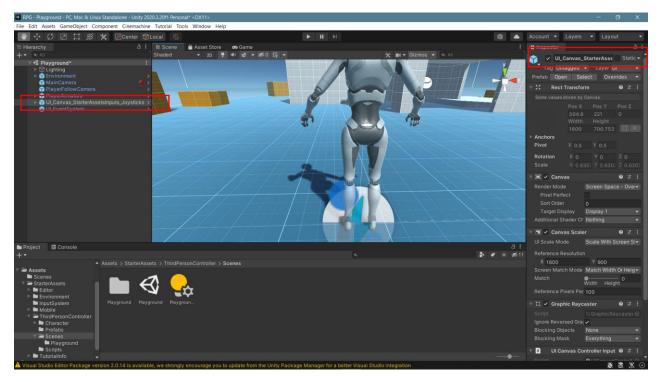


This game can also be adapted to mobile devices. However, since there are no keys like on a keyboard on mobile devices, these controls will be made with a virtual **TouchPad** or **keys**.

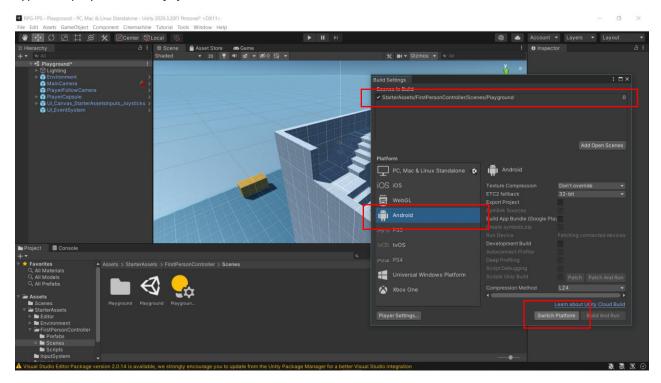
First, let's go to the **Movement Curser Settings** setting in the **PlayerArmature Inspector** section and **uncheck** the boxes. If the **Can Push** box is checked in the **Basic Rigid Body Push** section, it will also be possible to push objects.



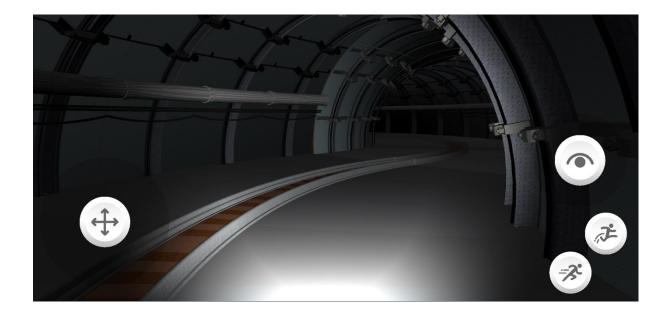
Then, activate the **UICanvasSaterterAssetsIputs\_Joysticks** object which is previously passive in the **Hierarchy** section.



Control is now enabled with the added **joystick**, <u>not</u> the **keyboard**. In this way, by going to **Build Settings**, it will be possible to convert it to **Android** or IOS application according to the **phone-tablet** type and play it with the **joystick** on mobile devices.



By using a mine gallery model, virtual joystick application can be performed. Here, FPS object also carries the light source (Spotlight) and makes illumination of the gallery with it.



# 9.3. Third Person Character Controller – Armature Change

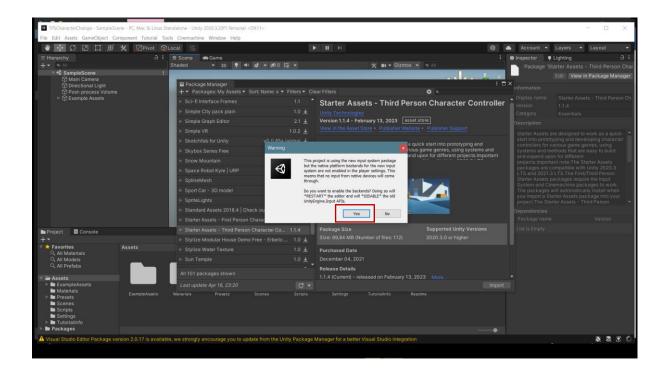
The app can be experienced on Unity 2020.3.10f and above. The **Universal Render Pipeline (URP)** template will be used.



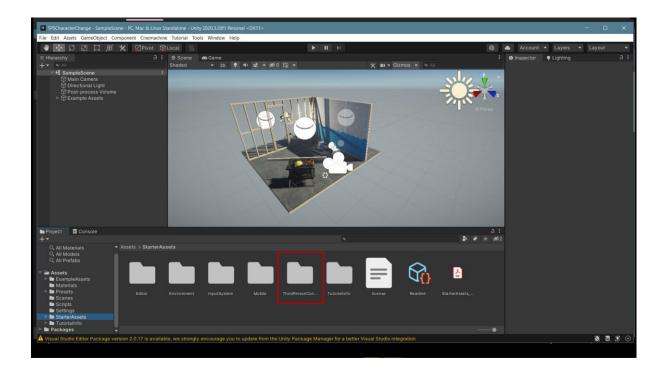
From Window -> Asset Store, Starter Assets will be downloaded to our Third Person Assets project.

							_	
TPSCharacterChange - Samples	Scene - PC, Mac & Linux S	tandalone - Unity 2020.3.20f1 Personal <dx11< th=""><th>2</th><th></th><th></th><th></th><th></th><th>1 X</th></dx11<>	2					1 X
File Edit Assets GameObject (	Component Tutorial W	indow Help						
🖤 🕂 Ø 🖬 🖽	🗙 🕢 Pivot 🚱			ны		📤 Account 👻 L	Layers 👻 Layout	-
		# Scene Game				Inspector     P Li	ighting	a i
+ • • •		Shaded 🛛 🕈 20 🌻 🕪 📚	▼ 950 ₩ ▼	🛠 🖬 👻 Gizmo	is i ≠Qr.All	Package 'Star		erson Char
✓ I SampleScene Main Camera							dit View in Package	e Manager
Directional Light		Package Manager			10			
Post-process Volum		+   Packages: My Assets   Sort:			¢ •			Parson Ch
		▶ Sci-fi Interface Frames	1.1	Starter Assets - Third Pe	erson Character Controller	Version		en aronn cent
		Simple City pack plain				Category		
		➢ Simple Graph Editor		Version 1.1.4 - February 13, 2023		Description		
		Simple VR				Starter Assets are		guick 🔺
		► Sketchfab for Unity		Starter Assets are designed to work as	a quick start into prototyping and			
		▷ Skybox Series Free	4.2 🕢	developing character controllers for var	ious game genres, using systems and	systems and meth		
		► Snow Mountain		methods that are easy to build and expand upon for different projects import		and expand upon projects.Important		ts .
		► Space Robot Kyle   URP		Images & Videos		packages are con		20:3
					A Long to Long	Starter Assets pac		ut
		Sport Car - 3D model	1.0 🛓	12 - > _ 18		System and Clnemachine packages to work The packages will automatically install whe		
		► SpriteLights	3.1 ±			you import a Starter Assets package project. The Starter Assets - Third Pe		
		Standard Assets 2018.4   Check c	ut Starte 1.1.6 m			Dependencies		
		Starter Assets - First Person Char.		View images & videos on Asset Store		Package name		n.
		Starter Assets - Third Person Cha		Package Size	Supported Unity Versions	List is Empty		
Project		Stylize Modular House Demo Free		Size: 69,84 MB (Number of files: 112)				
Favorites		Stylize Water Texture	1.0 ±	Purchased Date				
Q, All Materials Q, All Models		<ul> <li>Sun Temple</li> </ul>	1.0 ±	December 04, 2021				A : arson Char Manager Person Ch Prosen Ch aracter aracter alld ts to 0.3 to to to to to to to to to to to to to
Q, All Prefabs		Sun remple	1.0 •	Release Details				
∀ 📾 Assets		All 101 packages shown		1.1.4 (Current) - released on February 1	2 2022 Mars			
Assets ExampleAssets		Last update Apr 16, 23:20		the content of the about the burger				
Materials					DesiderA			
Scenes				Click proje	<ul> <li>to import assets from the package into you ect.</li> </ul>			
Scripts Settings								
Settings Tutorialinfo								
🕨 🖿 Packages								
A Visual Studio Editor Package							* 1	

After import, accept the **restart** request.



Since the **URP** template is used, the sample scene will appear on the screen.



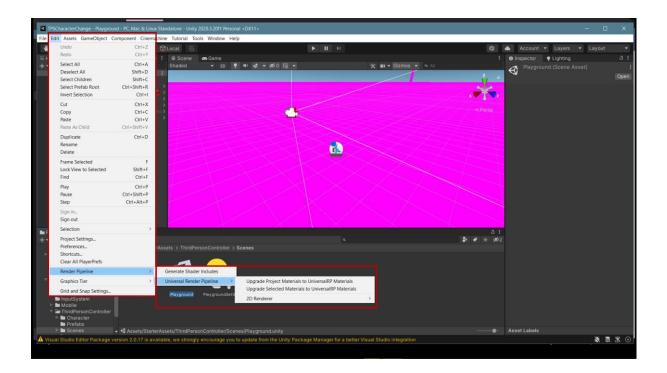
Under Starter Assets, double click on ThirdPersonController->Scenes->Playground to load the scene.



A pink screen will appear because there is an incompatibility between **URP** and **ThirdPersonController (TPS)**.

TPSCharacterChange - Playground -			<dx11></dx11>					- C	x נ
File Edit Assets GameObject Comp	ponent Cinemachine Tut	torial Tools Window Help							
👋 🐨 🖸 🖾 🗯 🏓	K Pivot GLocal					Ø	📤 Account 👻 Layers	- Layout	
		Scene 🛷 Game					Inspector PLighting		
+ * @ AI	Sha	ded 🔻 2D 💡	40 28 * 150 段 *		🔆 🖬 👻 Gizmos 👻 🤷 All		A Playground (Scene		
	i dincuta_loysticks			62		<pre>APmp</pre>	G Playground (Scene		, Open
Project Console	and the second second			a		-∃: ≱ ≠ ± ≠52			
Materials A Materials A Scenes Scripts Setings Setings Setings Setings Environment Impulystem Mobile ThirdPersonController Partabs	•	ThirdPersonController > Se							
	Assets/StarterAssets/T	hirdPersonController/Scenes							
A Visual Studio Editor Package version	ion 2.0.17 is available, we	e strongly encourage you to	update from the Unity Packa	ge Manager for a better V	isual Studio integration			*	0 2 8

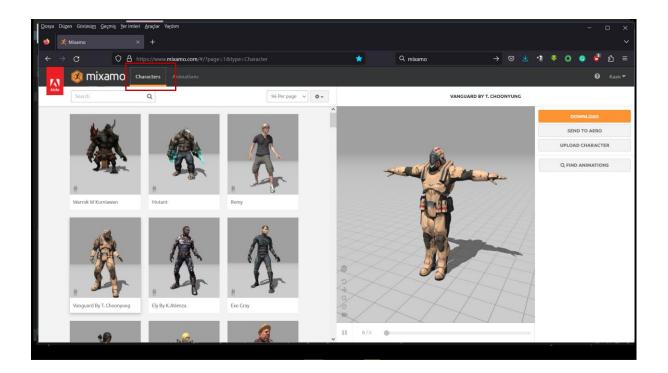
To solve the problem, select Edit->Render Pipeline->Universal Render Pipeline->Upgrade Project Materials to UniversalRP Materials.



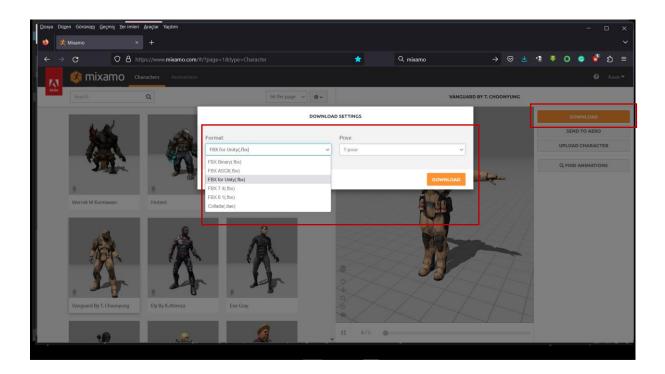
**Proceed** key will be proceeding here.

					- 10 C
TPSCharacterChange - Playground - PC, Mac & Linux Star					- 🗆 × '
File Edit Assets GameObject Component Cinemachine					
🖤 💀 🗘 🖾 🎞 🏽 🛠 📈 🖾 Pivot 😋				📤 Account 🕶 Layers 🔫	
	# Scene 🗢 Game			Inspector     Inspector	a :
+* (* AI	Shaded 🔹 2D 🌻 🕪 😂 💌 💋 0	描▼ X■I▼ Gizmos	er All	Playground (Scene Asset	) =
♥ S Playground I ► S Lighting			A	~	Open
Environment			~		
MainCamera     MainCamera     PlayerFollowCamera			x 🗸 🗸 🗸 z		
Player Unow Califera     Player Armature					
In ULConves_StarterAssetsInputs_Joysticks > In ULEventSystem					
ULEVentsystem >					
	1-1-1-0-0				
	77770				
	Materia	I Upgrader			
	1	The upgrade will overwrite materials in your project. Make sure to have a project backup before			
	R	proceeding.			
		Proceed Cancel			
Project Console			a:		
Materials     Assets > StarterAss     Presets					
Scenes					
E Scripts					
Settings StarterAssets					
🖻 🖿 Editor					
Environment Playground InputSystem	Playground PlaygroundSetti				
Mobile					
ThirdPersonController     Deracter					
Prefabs					
	ets/ThirdPersonController/Scenes/Playground.unity				
A Visual Studio Editor Package version 2.0.17 is availab					022%

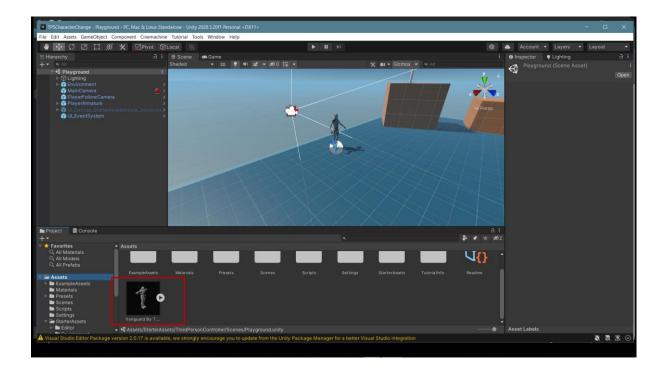
The problem will be solved. Now let's determine the character that we will replace with the robot. For this, let's choose a character from the **Adobe Mixamo** site.



From the window that opens with the **Download** button, select **FBX for Unity (fbx)**.



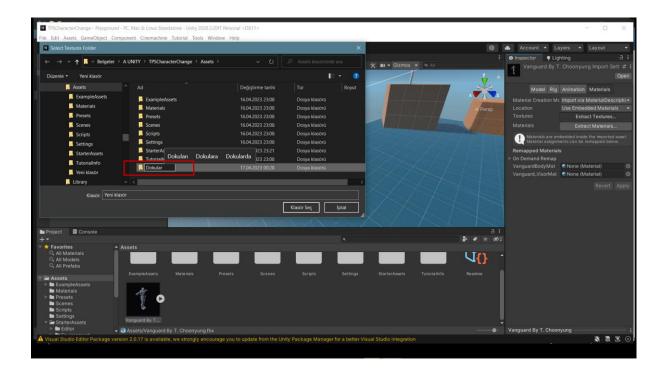
Now, drag the **fbx** file we downloaded into our Unity project (maybe under Assets).



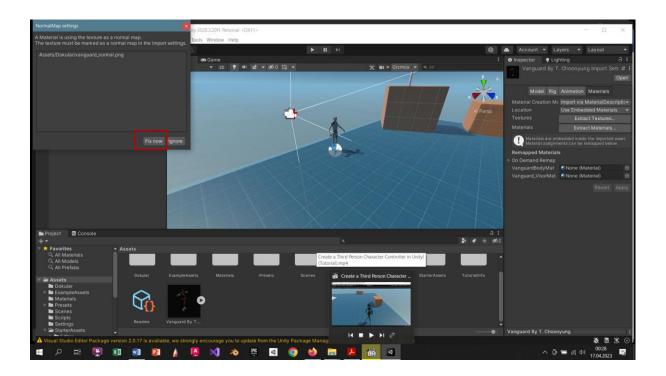
The character is completely white, and the textures are not visible. To solve this, select the file and click **Inspector->Materials->Textures->Extract Textures**.



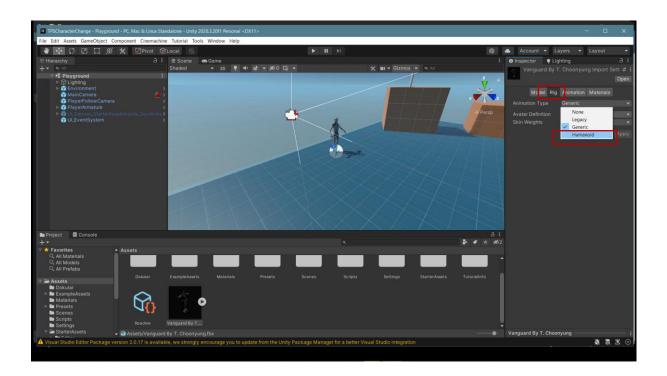
It asks us for a folder name. Open a new folder with the **right mouse button** and select it.



Then, press **Fix** now to solve the material problem.



Also, press Inspector->Rig->Animation Type->Humanoid.

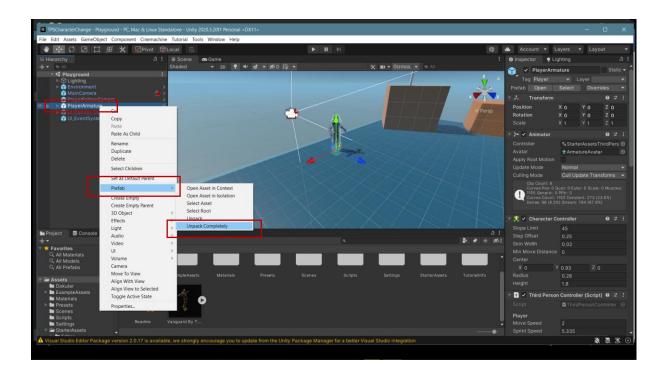


Confirm the changes with the **Apply** button.

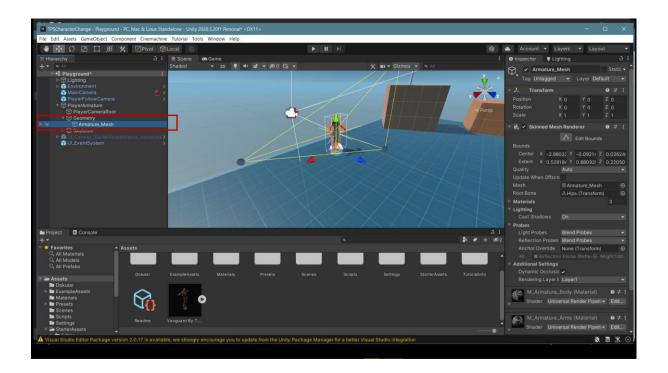
TPSCharacterChange - Playground - PC, 1	Mac & Linux Standalone - Unity i	2020.3.20f1 Personal <dx11></dx11>				-	_ × _
File Edit Assets GameObject Component							
♥☆ Ø Z I .# X	Pivot GLocal 🐻		► II FL		<u></u>	🔺 Account 👻 Layers 👻	Layout 👻
≅ Hierarchy	a : ⊞ Scene	😎 Game				Inspector     PLighting	a i
	Shaded	▼ 2D 🜻 🍽 💐 ▼ 🕫 0 蒜子		🛠 🖬 🕶 Gizmos 💌 🔤 All		Vanguard By T. Choonyung	g import Sett 🕫 🗄
	1				¥ = =		Open
Control Environment     MainCamera					1	Model Rig Animation	Materials
PlayerFollowCamera					×	Animation Type Humanoid	-
<ul> <li>PlayerArmature</li> <li>ConvestStarterAssetsInput</li> </ul>	ita Joysticks >	· · · · · · · · · · · · · · · · · · ·	4		Persp	Avatar Definition Create From	This Model 👻
😭 UI_EventSystem	·		-				<ul> <li>Configure</li> </ul>
							Bones) 👻
					1.1		
			T Car				Rever Apply
Project E Console					a :		
					<b>₽</b> € ± 152		
🔻 🚖 Favorites 🔶 Asse	ets						
Q, All Models					Î Î		
Q. All Prefabs							
🔻 🗁 Assets	Example Assets	Presets			1 OTOMONING		
ExampleAssets	$\sim$						
Materials	MA D	D					
Scenes	<b>N1</b>						
Scripts Settings	Readme Vanguard By T.						
T StarterAssets						Vanguard By T. Choonyung	1
A Visual Studio Editor Package version 2	.0.17 is available, we strongly	encourage you to update from the Unity F	ackage Manager for a better Visi	al Studio integration			

Color and textures are matched.

Now, in the **Hierarchy** section, unpack all the sub-parts with **PlayArmature->Prefab->Unpack Completely**.



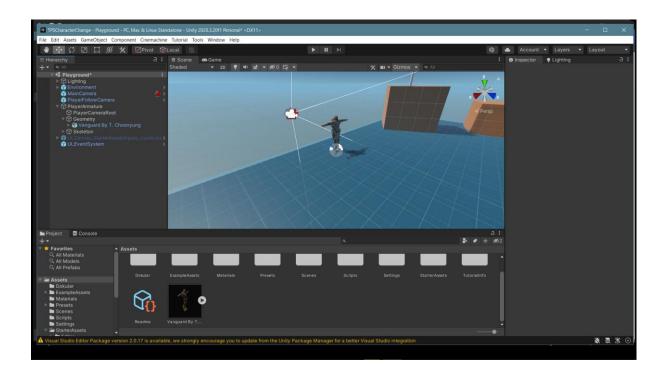
In the next step, to **delete** the robot: **PlayArmature->Geometry->Armature\_Mesh** will be deleted. The robot is now removed from the scene.



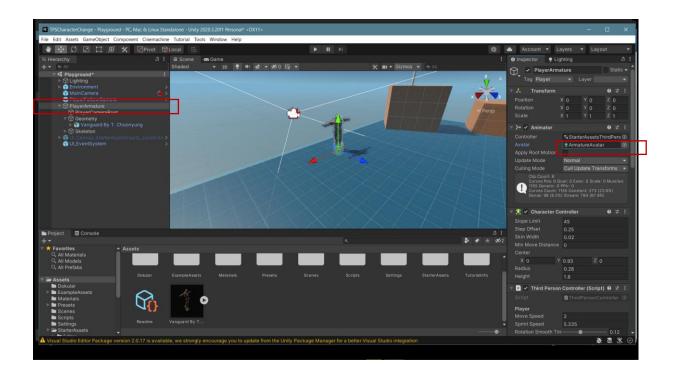
Drag the **fbx** character we downloaded from **Mixamo** and prepared for use under **Geometry**.



Our **fbx** (film box) character came instead of robot.

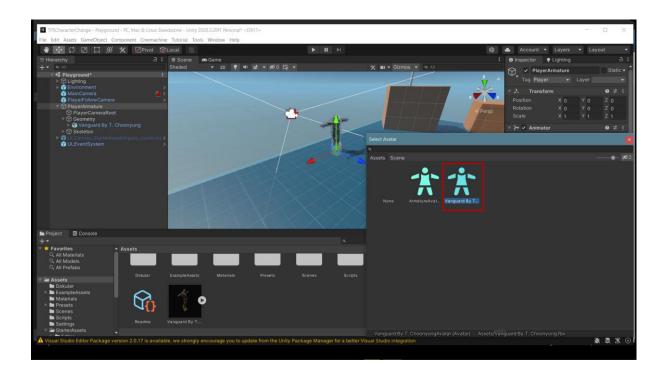


Finally, we need to change the avatar. To do this, open the list by pressing **PlayArmature>Inspector>Animator>Avatar**.



Select the avatar to which we added the FBX file.

Now, in the game mode, there is the character we selected instead of the robot.

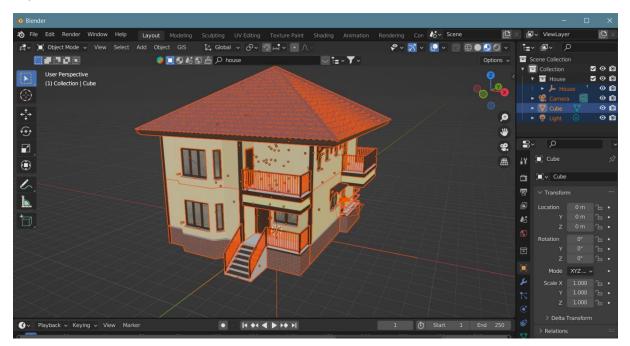


Similarly, a **mine worker** and the environment can also be used for the TPS application.

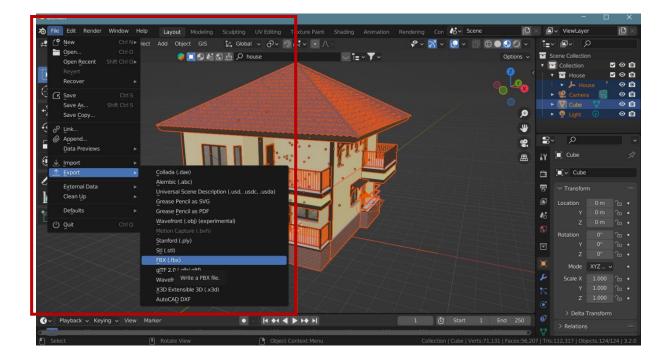


### 9.4. Transferring Blender Designs to Unity

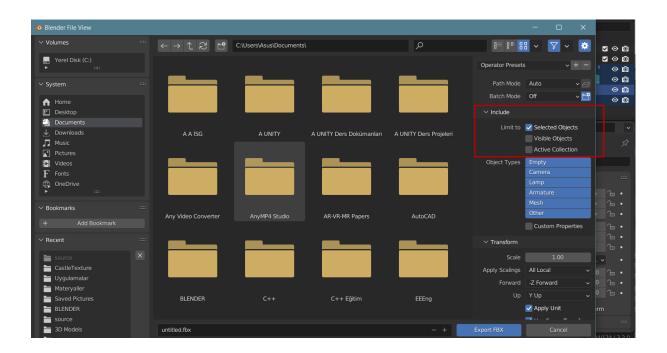
Blender is one of the most important design and animation packages for Unity. However, the files may not be fully transferred every time. In order to ensure the visibility of textures and colors when transferring Blender scenes or objects to Unity, the entire scene developed in Blender or certain object(s) is first selected.



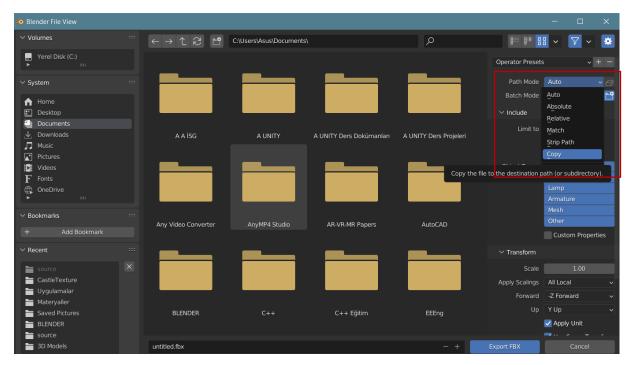
Then, click on File->Export->FBX (.fbx) option.



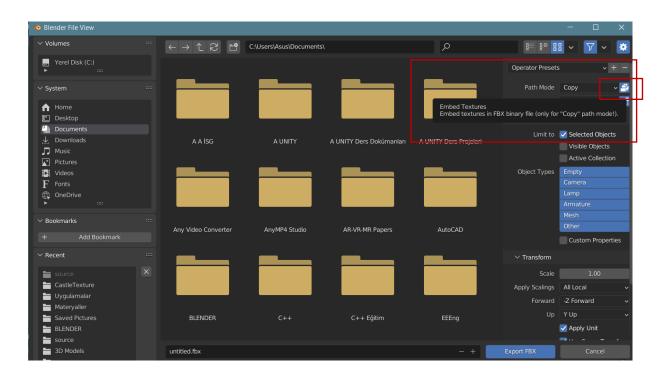
In the window that opens for recording, the following operations are performed in order: **Include**->Limit to section is clicked and **Selected Objects** is activated. Thus, the scene or object(s) we selected are included in the export process.



Copy is selected under the Path Mode section.



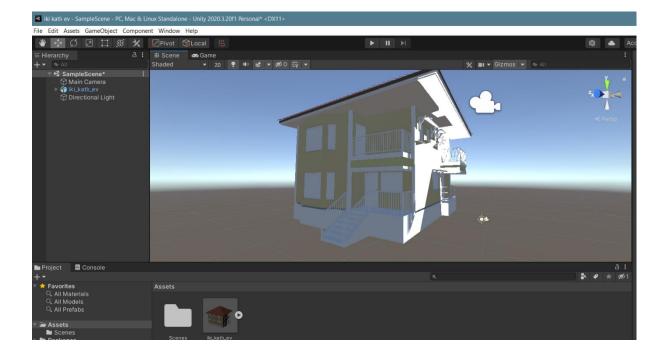
Again, the Embed Textures box next to Path Mode is selected and activated.



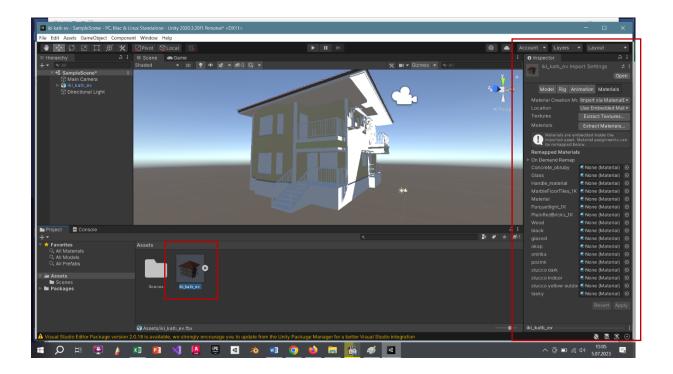
Then, give our file a name under the folder we specified and save it.

O Blender File View						– U .	×
✓ Volumes	$\leftarrow \rightarrow \uparrow \bigcirc$	C:\Users\Asus\Documents		<b>م</b>	<b>₽</b> ₽₽ <b>₽</b> ₽	~ 7 ~	*
₩ Yerel Disk (C:)					Operator Presets	~ +	
✓ System ····					Path Mode C		2
A Home					Batch Mode C	ff v	Ľ
Desktop					$\sim$ Include		
<ul> <li>Documents</li> <li>Jownloads</li> <li>Music</li> <li>         Pictures     </li> </ul>	A A İSG	A UNITY	A UNITY Ders Dokümanları	A UNITY Ders Projeleri		Selected Objects Visible Objects Active Collection	
Videos F Fonts OneDrive						mpty amera amp rmature	
<ul> <li>Bookmarks</li> <li>+ Add Bookmark</li> </ul>	Any Video Converter	AnyMP4 Studio	AR-VR-MR Papers	AutoCAD	c	lesh ther Custom Propertie:	25
✓ Recent ····					✓ Transform		
source					Scale Apply Scalings A	1.00 II Local	
Uygulamalar					Forward -2	Z Forward	
<ul> <li>Materyaller</li> <li>Saved Pictures</li> </ul>	BLENDER	C++	C++ Eğitim	EEEng		Up	
BLENDER						Apply Unit	
source							
B SOURCE	iki_katlı_ev			- +	Export FBX	Cancel	

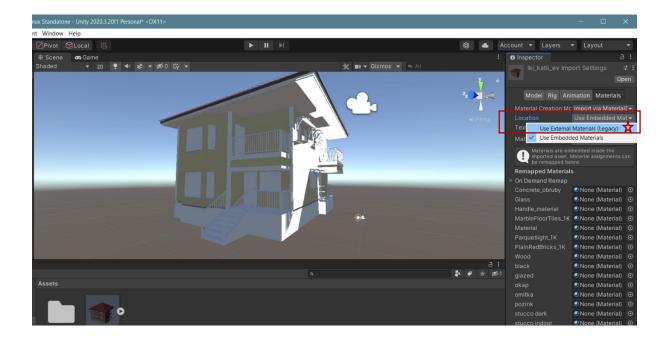
Now, come to our Unity scene. Drag the **FBX** file to the Assets section.



It seems that the textures and coatings have not arrived completely. Now, select our **FBX** object in the Assets section and look at its settings in the **Inspector** section.



Select Materials->Location->Use Embedded Materials (Legacy).



Then, click **Apply**. In some cases, click **Fix Now** in the window that can be opened for customization purposes.



As you can see, the textures, colors and textures in Blender have been transferred.

## **10. VIRTUAL REALITY – VR APPLICATIONS**

## 10.1.Virtual Reality (VR) Application for Cardboard Devices

It is possible to say that **cardboard**-type **headsets** are the most economical VR headsets for experiencing virtual reality applications. These **stereoscopic** (superimposing 2 images) based devices have a chamber where mobile phones can be placed. Therefore, the main application output platform is mobile devices such as mobile phones and tablets.



It is possible to print cardboard-type devices with Unity. For this purpose, the **Google Cardboard XR Plugin** provides a template that will make the process easier.

Let's open the address <u>https://github.com/googlevr/cardboard-xr-plugin</u> on the **GitHub** site where the plugin is located. Copy the link by clicking on the link in the **HTTPS** in the window that opens in the **Code** section.

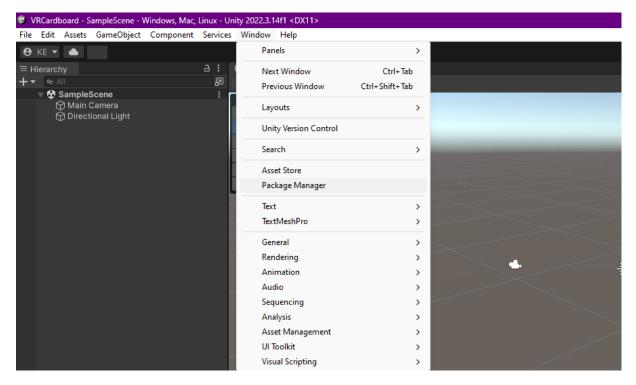
googlevr / cardboard-xr-plugin					
de 💿 Issues 🕦 🎝 Pull requests 🕦 🙂 Security	🗠 Insights				
© cardboa	rd-xr-plugin (Public)				
	🕈 1 Branch 🔊 25 Tags			Add file 👻 <> Code 👻	About
🗢 xinyunhû	929 Cardboard XR Plugin v1.23.0		Local		Google Cardboard XR Plugin for Unity
allstar		Cardboard XR Pluc	Clone		다 Readme 화 Apache-2.0 license
.github		Cardboard XR Plug	HTTPS SSH GitHub	CLI Copy url to clipboard	
Editor		Cardboard XR Plug	https://github.com/goog	levr/cardboard-xr-plu 🗜	
Resources		Cardboard XR Plug			
🖿 Runtime		Cardboard XR Plug	🔄 Open with GitHub Deskt	op	
Samples~		Cardboard XR Plug	Download ZIP		
	TING.md	Cardboard XR Plug			Releases 25
	TING.md.meta				on Dec 15, 2023
🗋 Editor.met	1				
	d				Packages



After this preliminary preparation, develop a simple Google VR Cardboard project in Unity.

In **Unity 2021.3** or later (here 2022.3.1.14f), create a 3D project. We named our project **VRCardboard** in this application.

Go to Window>Package Manager

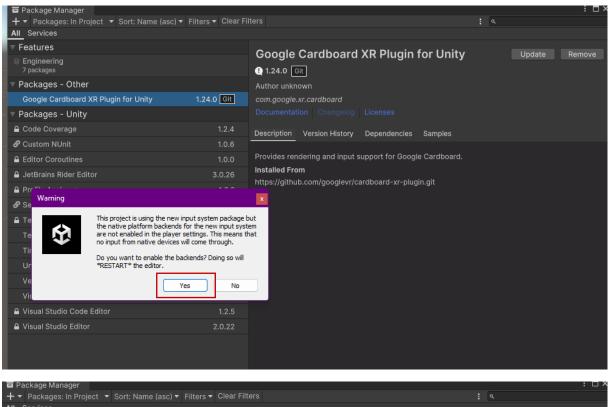


Open the at the menu by pressing the + key here. Select the Add package from git URL... line.

■ Package Manager + ▼ Packages: In Project ▼ Sort: Name (as Add package from disk	c) ▼ Filters ▼ Clear Fil	ters : Q
Add package from tarball Add package from tarball Add package from git URL Add package by name Fackages - Only		Code Coverage     Unlock     Remove       1.2.4 · June 01, 2023     Release       Image: Signification of the Engineering feature.
Code Coverage Custom NUnit Custom Coroutines	1.2.4 1.0.6 1.0.0	From <b>Unity Registry</b> by Unity Technologies Inc. <i>com.unity.testtools.codecoverage</i> Documentation Changelog Licenses
<ul> <li>▲ JetBrains Rider Editor</li> <li>▲ Profile Analyzer</li> <li>         Ø Settings Manager     </li> </ul>	3.0.26 1.2.2 2.0.1	Description Version History Dependencies Samples Use this package to export code coverage data and reports from your automated tests. Additionally,
	1.1.33 3.0.6 <b>1</b>	the Code Coverage package offers a Coverage Recording feature which allows capturing coverage data on demand, for manual testing or when there are no automated tests in the project.
Unity UI Version Control	1.7.6 1.0.0 2.2.0	
Visual Scripting  Visual Studio Code Editor Visual Studio Editor	1.9.1 1.2.5 2.0.22	

Paste the link we just copied here and press the Add button.

🖬 Package Manager		: 🗆 ×
+ ▼ Packages: In Project ▼ Sort: Name (asc) ▼ Filters ▼ Clear		
ኝ* Add package from git URL		
https://github.com/googlevr/cardboard-xr-plugin.git		Add
<ul> <li>Engineering</li> <li>7 packages</li> </ul>	■ 1.2.4 · June 01, 2023 Release	
Packages - Unity		



All Services		
▼ Features		Code Coverage Unlock Remove
<ul> <li>Engineering</li> <li>7 packages</li> </ul>		□ 1.2.4 · June 01, 2023 Release
Packages - Other		So is installed as part of the Engineering feature.
Google Cardboard XR Plugin for Unity	1.24.0 Git	From Unity Registry by Unity Technologies Inc.
Packages - Unity		com.unity.testtools.codecoverage
🔒 Code Coverage	1.2.4	
	1.0.6	Description Version History Dependencies Samples
Editor Coroutines	1.0.0	
🔗 Input System	1.7.0	Use this package to export code coverage data and reports from your automated tests. Additionally, the Code Coverage package offers a Coverage Recording feature which allows capturing coverage
JetBrains Rider Editor	3.0.26	data on demand, for manual testing or when there are no automated tests in the project.
Profile Analyzer	1.2.2	
🔗 Settings Manager	2.0.1	

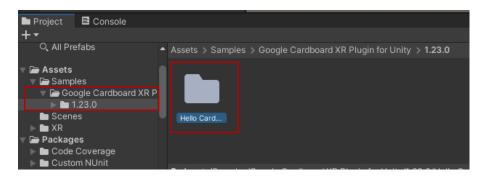
The **add-on** is now installed in our package manager. If there is any **update** suggestion, we can do it by clicking **Update**.

■ Package Manager     +      +      Packages: In Project      Sort: Name (asc)     All Services	▼ Filters ▼ Clear Filters ▼ Filters ▼ Filters ▼ Filters ▼ Filters ▼ Clear Filters ▼ Clear Filters ▼ Filters ▼ Clear Filters ▼ Clear Filters ▼ Clear Filters ■ Clear Filter	ilters : Q
Features  Engineering 7 packages  Packages - Other		Google Cardboard XR Plugin for Unity Update Remove Q 1.23.0 Git Click to check for updates and update to latest version
Google Cardboard XR Plugin for Unity  Packages - Unity  Code Coverage	1.23.0 Git	Author unknown version com.google.xr.cardboard Documentation Changelog Licenses
Custom NUnit Editor Coroutines	1.0.6 1.0.0	Description Version History Dependencies Samples Provides rendering and input support for Google Cardboard. Installed From
<ul> <li>▲ JetBrains Rider Editor</li> <li>▲ Profile Analyzer</li> <li>         Ø Settings Manager     </li> </ul>	3.0.26 1.2.2 2.0.1	instaned From https://github.com/googlevr/cardboard-xr-plugin.git
Test Framework	1.1.33	

Since the preset scene, which is a **template** for the package, is located in the **Samples** section, open it and **import** the sample scene named **Hello Cardboard**.

Google Cardboard XR Plugir	for Unity	Update	Remove
(1.23.0 Git			
Author unknown			
com.google.xr.cardboard			
Documentation Changelog Licenses			
Description Version History Dependencies	Samples		
Hello Cardboard 2,01 MB V Basic sample, minimal project setup.			Reimport

After this process, we can see that the folder is placed in the **Project** section.



We can close the package manager.

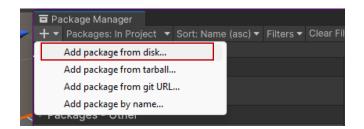
Some computers may not be able to do this, so another way to get the package is to download it as a **ZIP** file.

🐉 master 🔹 🐉 1 Branch 🛇 26 Tags		Q Go to file	<> Code 🝷	About
xinyunh0929 Cardboard XR Plugin v1.24.0		D Clone	(?)	Google Cardboard XR Plug
xinyumoszs Carubbard XK Piugin V1.24.0		HTTPS GitHub CLI		🛱 Readme
🖿 .allstar	Cardboard XR Plugi			좌 Apache-2.0 license
💼 .github	Cardboard XR Plugi	https://github.com/googlevr/cardboard-x	r-plu 🖸	- Activity
				Custom properties
Editor	Cardboard XR Plugi			☆ 279 stars
Resources	Cardboard XR Plugi	다. Open with GitHub Desktop		24 watching
Runtime	Cardboard XR Plugi			<b>父 121</b> forks
Kunume	Cardboard XK Plugi	Download ZIP		Report repository

After the compressed file is downloaded to the hard disk, open it and extract it.

			00,00
∼ Bugün			
Screenshot_20240428-175407_VRWalk.jpg	28.04.2024 17:56	JPG Dosyası	187 KB
🚞 cardboard-xr-plugin-master	28.04.2024 18:39	Dosya klasörü	
in cardboard-xr-plugin-master.zip	28.04.2024 12:47	WinRAR ZIP arşivi	17.143 KB
$\sim$ Geçen hafta			

In the Package Manager, press + and click the Add package from disk option.



The file type Unity is looking for here is **JSON**. Therefore, we can include the **Google VR** package in our project by selecting the **package.json** file.

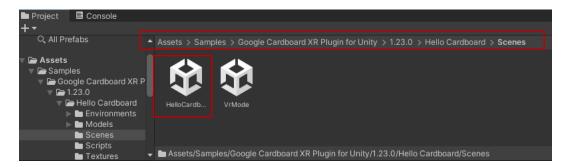
Select package on c	disk		×
$\leftrightarrow$ $\rightarrow$ $\checkmark$ $\uparrow$	📄 > İndirilenler > care	dboard-xr-plugin-master > cardboard-xr-plugin-ma	ter > ح C cardboard-xr-plugin-master ۵
Düzenle 🔻 Yeni k	klasör		
🏫 Giriş 🛛 🛛	Ad	Değiştirme tarihi	
🗾 Galeri	$\checkmark$ Geçen ay		
✓	🗐 package.json	8.03.2024 00:14	
> 📑 Belgeler	🚞 .allstar	8.03.2024 00:14	
	🧰 .github	8.03.2024 00:14	
> 🚞 Ekler	Editor	8.03.2024 00:14	
> 🧾 Masaüstü	Resources	8.03.2024 00:14	Önizlenecek bir dosya seçin.
> 🔀 Resimler	🦰 Runtime	8.03.2024 00:14	
	Samples~	8.03.2024 00:14	
🧾 Masaüsl 🖈			
🛓 İndirilen 🖈			
📑 Belgeler 🖈			
🔀 Resimle 🖈			
	Dosya adı:		✓ package.json file (*.json) ✓
			Aç İptal

The next part continues as described above. Restarting, Update, Sample addition operations are the same.

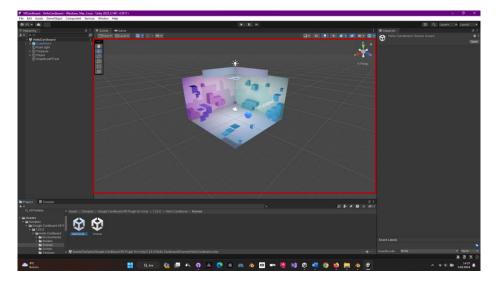
## 10.2.Configuring the HelloCardboard Scene

To open the sample scene, go to Assets/Samples/Google Cardboard/<version>/Hello Cardboard/Scenes.

Let's select Add Open Scenes and select HelloCardboard.



With this selection, the ready scene will appear on the screen. A **VR room** is seen as the content.



Now, let's change our platform to **Android** (IOS for Apple). Select **File>Build Setting>Android** and click **Switch Platform**.

Ө КЕ • 🔺			• Ш. Н.		
Te Hierarchy 3 1 Scene					
+ • • All 200	t♥ @Local♥ 🐺▼ 🗒 = ₩ ♥			Q * 20 📍 4 🛃 * 95	■• ⊕ •
C HelleCardbeard     C Post light     O Float Rot     Of Transit     Of Transit     Of Transit     Of Transit     Of Transit     Of Transit					
				i n×	
	Scenes in Build				
				Add Open Scenes	
	Platform	🖶 Android			
2 A	🖵 Windows, Mac, Linu	Texture Compression ETC2 failback	Use Player Settings 32-bit		
1.00	Dedicated Server	Export Project	32-01		
	👘 Android				
line and the second second second second second second second second second second second second second second	WebGL		Disabled		
			Default device	<ul> <li>Refresh</li> </ul>	
1523	Universal Windows	Platform Build to Device Patching is disabled for Release build			
		Script Debugging Compression Method			
	Asset Import Overrides	<ul> <li>Compression Method</li> </ul>			
	Max Texture Size No Di	remide •			
	Texture Compression No On	remide -			
Project Console	Player Settings			Switch Platform	a :
+-					
Q, All Prefabs Assets > Samples > Google					
v Da Assets					

Let's add our current scene to the Scenes in Build section by clicking Add Open Scenes.

Build Settings			: 🗆 ×
Scenes In Build			
<ul> <li>Samples/Google Cardboard XR Plugin for U</li> </ul>	nity/1.23.0/Hello Cardboard/Scen	es/HelloCardboard	0
			Add Open Scenes
Platform	🖷 Android		
Windows, Mac, Linux	Texture Compression	Use Player Settings	•
Dedicated Server	ETC2 fallback	32-bit	<b>*</b>
*	Export Project Symlink Sources		
🖬 Android 🗘	Build App Bundle (Google Play)		
MabGI	Create symbols.zip	Disabled	▼.

At this stage, we need to make a series of settings. In the **Build Settings** window, click on **Player Settings**. The titles in the settings may vary depending on the version. In this study, version 2022.3 is taken as basis.

In **Player>Icon** section, **Company Name** and **Product name** can be changed. Here the company name is changed to **DPU**.

🌣 Project Settings					: 🗆 ×
				٩	
Adaptive Performance Audio	Player				0 ≠ :
Editor	Company Name	DPU			
Graphics	Product Name	VRCar	dboard		
Input Manager Memory Settings	Version	0.1			
Package Manager Physics Physics 2D	Default Icon				None (Texture 2D)
Player					Select
Preset Manager Quality Scene Template Script Execution Order	Default Cursor				None (Texture 2D) Select
Services	Cursor Hotspot	хo		Y O	38161
Tags and Layers TextMesh Pro					
Time	· · · · · · · · · · · · · · · · · · ·			Û	4
Timeline UI Builder	Settings for Android				
Version Control	► Icon				

In the next step, go to the **Player Settings>Resolution and Presentation** submenu. <u>Uncheck</u> the **Optimized Frame Pacing** box.

In the options under **Orientation**, instead of **Auto Rotation**, you can select **Landscape Right** or **Landscape Left**. Here **Landscape Right** is selected.

Build Settings	Project Settings			: 🗆		
Scenes In Build			٩			
<ul> <li>Samples/Google Carc Adaptive Performance Audio</li> </ul>		Player				
	Audio Editor	Resolution and Presentation				
	Graphics	Fullscreen Mode	Fullscreen Window			
	Input Manager	Resizable Window				
	Memory Settings Package Manager	Hide Navigation Bar	~			
	Physics		✓			
	Physics 2D					
Platform	Player	Resolution Scaling				
Windows, Mac,	Preset Manager Quality		Disabled			
1 <u> </u>	Scene Template					
Dedicated Serv	Script Execution Order	Blit Type	Always			
<u> </u>	Services	Supported Aspect Ratio				
Android	Tags and Layers TextMesh Pro		Native Aspect Ratio			
HTHL	Time	Orientation				
5 WebGL	Timeline	Default Orientation*	Auto Rotation			
	UI Builder	Allowed Orientations for Auto Rotation	Portrait			
Universal Wind	Version Control		Portrait Upside Down			
iOS ios	Visual Scripting XR Plug-in Management	Portrait Upside Down	Landscape Right			
10.5 103	XK Flug-III Management	Landscape Right	Landscape Left			
PS4 PS4			<ul> <li>Auto Rotation</li> </ul>			
		Use 32-bit Display Buffer*	Auto Notation			
PJS PS5		Disable Depth and Stencil*				
		Render Over Native UI*				
▼ Asset Import Overrid			Don't Show			

Now, go to the **Other Settings** section related to the **Player**. Select **ETC2** as the **Texture compression** format.

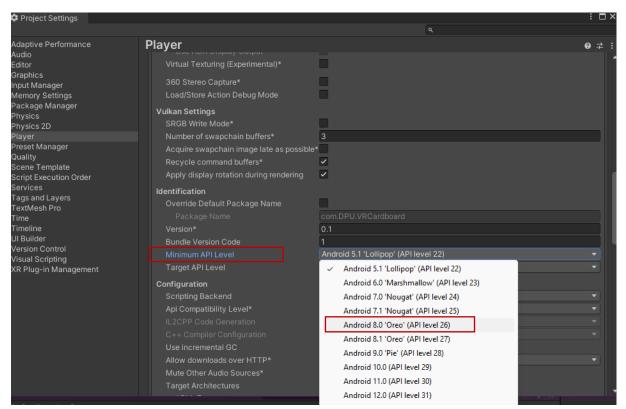
Package Manager	Multithreaded Rendering*	✓	
Physics	Static Batching	✓	
Physics 2D	Dynamic Batching		
Player			
Preset Manager	Sprite Batching Threshold	• 300	
Quality	GPU Compute Skinning*		
Scene Template	Graphics Jobs (Experimental)		
Script Execution Order Services	Texture compression format	ASTC	
Tags and Layers	Normal Map Encoding	ETC	
TextMesh Pro	Lightmap Encoding	ETC2	
Time	HDR Cubemap Encoding	✓ ASTC	
Timeline	Lightmap Streaming		
Ul Builder	Streaming Priority	DXT	
Version Control		DXT + RGTC(BC4, BC5)	
Visual Scripting	Frame Timing Stats		
XR Plug-in Management	OpenGL: Profiler GPU Recorders	<ul> <li>Image: A start of the start of</li></ul>	

#### Vulkan Settings> Apply display rotation during rendering box to disable it.

Vulkan Settings		
SRGB Write Mode*		
Number of swapchain buffers*	3	
Acquire swapchain image late as possible		
Recycle command buffers*	~	
Apply display rotation during rendering		

#### Check if the company name is the same as above.

Identification		
Override Default Package Name		
	com.DPU.VRCardboard	
Version*	0.1	
Bundle Version Code	1	



#### Set the Android level as Minimum API Level>Android 8.0 'Oreo' (API Level 26).

#### Select Android 13.0 (API Level 33) for Target API Level.

Build Settings	🌣 Project Settings			: 🗆 ×
Scenes In Build			٩	
<ul> <li>Samples/Google Carc</li> </ul>	Audio	Player Virtual Texturing (Experimental)*		@ ⊉ :
1	Editor Graphics Input Manager	360 Stereo Capture*		
	Memory Settings			
	Package Manager Physics Physics 2D	Vulkan Settings SRGB Write Mode*	<u> </u>	
Platform	Player Preset Manager	Number of swapchain buffers*	3	
Windows, Mac,	Quality	Acquire swapchain image late as possible <sup>4</sup> Recycle command buffers*	✓ ✓	
Dedicated Serv	Scene Template Script Execution Order		~	
Android	Services Tags and Layers	Identification		
-π.	TextMesh Pro			
WebGL	Time Timeline	Package Name Version*	com.DPU.VRCardboard	
	Ul Builder	Bundle Version Code	1	
Universal Wind	Version Control	Minimum API Level	Android 5.1 'Lollipop' (API level 22)	
iOS ios	Visual Scripting XR Plug-in Management		Automatic (highest installed)	
Pra PS4		Configuration	<ul> <li>Automatic (highest installed)</li> </ul>	
PJ4 F34			Android 5.1 'Lollipop' (API level 22)	
Prs PS5			Android 6.0 'Marshmallow' (API level 23)	
▼ Asset Import Overrid			Android 7.0 'Nougat' (API level 24) Android 7.1 'Nougat' (API level 25)	
Max Texture Size			Android 8.0 'Oreo' (API level 26)	
Texture Compression			Android 8.1 'Oreo' (API level 27)	
Player Settings		Mute Other Audio Sources* Target Architectures	Android 9.0 'Pie' (API level 28)	
			Android 10.0 (API level 29)	
ugin for Unity >1.23.0 > H			Android 11.0 (API level 30)	
			Android 12.0 (API level 31)	
			Android 12.0 (12L, API level 32)	
			Android 13.0 (API level 33)	

On the same page, set the **Configuration>Scripting Backend** selection to **IL2CPP**.

Configuration		
Scripting Backend	Mono	
Api Compatibility Level*	✓ Mono	
	IL2CPP	
	Release	

Click the **ARM64** box in the **Target Architectures** section.

Set the Internet Access section to Require.

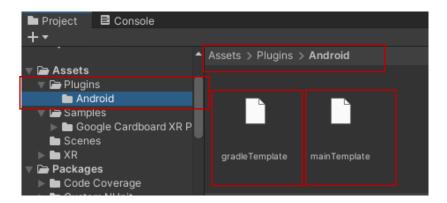
Install Location	FIEI		· · · · · · · · · · · · · · · · · · ·
Internet Access	Auto	þ	 •
Write Permission	· · ·	Auto	•
Filter Touches When Obscured		Require	
Sustained Performance Mode			

*Note: If the Unity version is 2023.1 or later, select Activity in the Application Entry Point and deselect GameActivity.* 

Open the **Player>Publishing Settings** window. Here, let's select the **Custom Main Gradle Template** and **Custom Gradle Properties Template** boxes.

Build Settings	🗘 Project Settings		: 🗆 X
Scenes In Build			
<ul> <li>Samples/Google Carc</li> </ul>	Audio	Player	0 ≓ :
	Editor Graphics Input Manager Memory Settings Package Manager Physics	Publishing Settings Keystore Manager Project Keystore Custom Keystore	
Platform	Physics 2D Player	The application will be signed with a debug key	
Windows, Mac,	Preset Manager Quality Scene Template		
Dedicated Serv	Script Execution Order Services		
WebGL	Tags and Layers TextMesh Pro Time	Project Key Alias Debug	
Universal Wind	Timeline Ul Builder		
	Version Control Visual Scripting	Build	
iOS ios	XR Plug-in Management	Custom Main Manifest Custom Launcher Manifest	
₽л₄ PS4		Custom Main Gradle Template	
DOF		Assets\Plugins\Android\mainTemplate.gradle	
Prs PS5		Custom Launcher Gradle Template	
▼ Asset Import Overrid		Custom Base Gradle Template Custom Gradle Properties Template	
Max Texture Size			
Texture Compression		Assets\Plugins\Android\gradleTemplate.properties	
Player Settings		Custom Gradie Settings Template Custom Proguard File	

This last operation caused the **Plugins** subheading to open in **Project>Assets**. When we look here, we see that two files named **gradleTemplate** and **mainTemplate** were added under the **Android** folder.



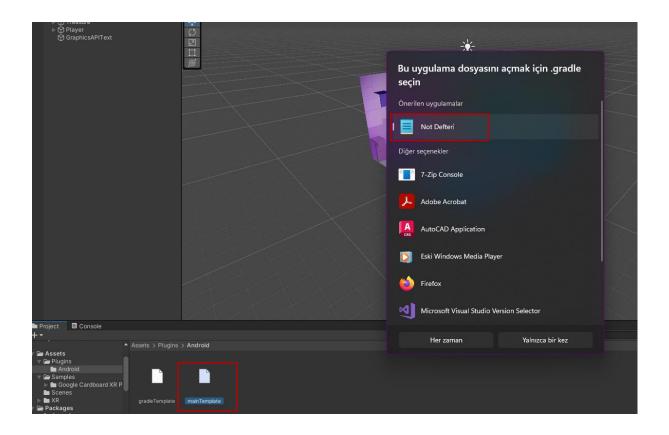
Now, an addition needs to be made to these files. The text of these two code snippets, which can also be copied from the page

<u>https://developers.google.com/cardboard/develop/unity/quickstart?hl=en</u> and how to add them are listed below.

First code block of four lines:

```
implementation 'androidx.appcompat:appcompat:1.4.2'
implementation 'com.google.android.gms:play-services-vision:20.1.3'
implementation 'com.google.android.material:material:1.6.1'
implementation 'com.google.protobuf:protobuf-javalite:3.19.4'
```

Let's copy it from here. Let's double-click on the **mainTemplate.gradle** file to open it. We will be asked to choose an editor. **Notepad** is sufficient for this job.



Once the file is opened, we need to determine where to copy the copied code block.

	Kaspersky	٠	mainTemplate.gradle	×	KameraGecis.cs	CameraChanging.cs	+
Dosy	/a Düzenle	Görünüm	1				
	ly plugin: 'co PPLY_PLUGINS**		d.library'				
	endencies { implementatio EPS**}	n fileTr	ee(dir: 'libs',	incl	ude: ['*.jar']) <sub>4</sub>		
and	roid { ndkPath "**ND	KPATH**"					
	compileSdkVer buildToolsVer						
		patibili	ty JavaVersion.V ty JavaVersion.V		-		

Here, paste the codes to the place indicated by the arrow above the **\*\*DEPS\*\***} expression and save it.

Kaspe	sky	• mai	nTemplate.gradle	×	KameraGecis.cs	CameraChan	nging.cs	+	
Dosya D	üzenle Gör	rünüm							
apply plu **APPLY_P dependenc		ndroid.li	ibrary'						
•	· ·	ileTree(d	lir: 'libs',	incl	ude: ['*.jar'])				
impleme impleme	ntation 'con ntation 'con	n.google. m.google.	android.mate	play rial	t:1.4.2' -services-vision:20.1 :material:1.6.1' f-javalite:3.19.4'	.3'			
**DEPS**}									

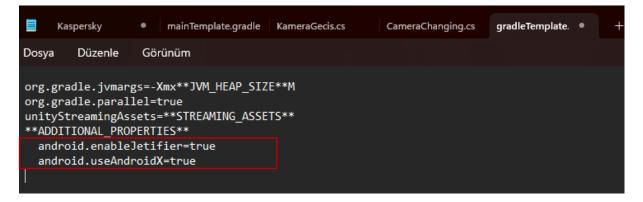
Open the second file, i.e. **gradleTemplate.gradle**, in the same way, for example, with **Notepad**. As you can see, this file has relatively short content.

	Kaspersky	mainTemplate.gr	adle Kame	raGecis.cs	CameraChanging.	cs gradleTempla	te ×
Dosya	Düzenle	Görünüm					
org. unit	gradle.paral	sets=**STREAMING_					

Copy the two lines of code given below.

android.enableJetifier=true
android.useAndroidX=true

Paste it to the location shown in the file and save it.



Go to Project Settings>XR Plug-in Management and select the Cardboard XR Plugin box.

Project Settings      Adaptive Performance Audio Editor Graphics Input Manager Initialize XR on Startup Initialize X
Audio Editor Editor Graphics Q I Input Manager I Input Manager
Input Manager
Memory Settings
Package Manager Plug-in Providers 😯
Physics 2D ARCore
Player Cardboard XR Plugin
Quality Oculus
Scene Template Unity Mock HMD
Script Execution Order Services
Tags and Layers
TextMesh Pro Information about configuration, tracking and migration can be found below.
Timeline View Documentation
UI Builder Version Control
Visual Scripting
XR Plug-in Management

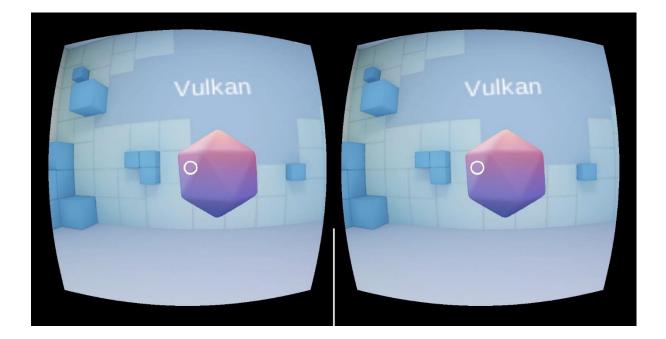
Come to **Build Settings**. Connect our smart mobile device with developer features enabled to the computer. Here, **Refresh** the **Run Device** section and select the **mobile device**.

Build Settings			: 🗆 ×
Scenes In Build			
<ul> <li>Samples/Google Cardboard XR Plugin for Units of the second</li></ul>	nity/1.23.0/Hello Cardboard/Scen	es/HelloCardboard	
Distance			Add Open Scenes
Platform	🖷 Android		
Windows, Mac, Linux	Texture Compression	Use Player Settings	
	ETC2 fallback	32-bit	
Dedicated Server	Export Project		
🛋 Android 😒			
·π·	Build App Bundle (Google Play)		
WebGL	Create symbols.zip Run Device	Disabled Samsung SM N975F (RF8NB069L8L)	▼ Refresh
	Build to Device	Default device	Patch And Run
Universal Windows Platform	Patching is disabled for Release build		
iOS ios	Development Build	All compatible devices <enter ip=""></enter>	
			<u> </u>
PS4 PS4		<ul> <li>Samsung SM N975F (RF8NB069L8L)</li> </ul>	
ars PS5			
₽JE P30 ▼	Compression Method	LZ4	
Asset Import Overrides			
Max Texture Size No Override 🔹			
Texture Compression No Override -			Learn about Unity Build Automation
Player Settings			Build 🔻 Build And Run

In the window that opens with the **Build and Run** key, define an **APK** file name (**VROda1**) by opening a new folder (here **APK1**) or directly in the current location.

👳 Build Android					×
$\leftarrow$ $\rightarrow$ $\checkmark$ $\uparrow$ $\frown$ Bu bilgisayar $\rightarrow$	Yeni Birim (D:) > UNITY > VRCardboard > APK1	~ C		م ra	>
Düzenle 🔻 Yeni klasör					?
🚞 A EĞİTİM MATEF 🛛 🗛 🗎	^ Değiştirme tarihi	Tür Boy	rut		
VRRoomAPK2	Aramanızla eşleşe	n öğe yok.			
📩 Ders Notları - Ka					
VRGallery1					
🗸 💻 Bu bilgisayar					
> 🐸 Windows (C:)					
> 🛭 🚔 Yeni Birim (D:)					
> 🛬 Ağ					
Dosya adı: VROda1					~
Kayıt türü: apk (*.apk)					~
∧ Klasörleri Gizle			Kaydet	İptal	

After installing the project as an **APK** file on the phone, the application with two stereoscopically symmetrical images will start working.



A **3D** room experience can be achieved by attaching a **cardboard** headset. The headset moves, and the Treasure changes color with the **Reticler Pointer** and can be transported with the button under the headset.

At this point, we can remove the room and objects on the stage and position a 3D environment of our design on the stage. In this way, we can achieve 3D experience for many designs.

There are tutorials on the internet that show what happens when the aimer hovers over an object. Actions that will happen when the object is in the **On Click** state can be done with **C# coding**.

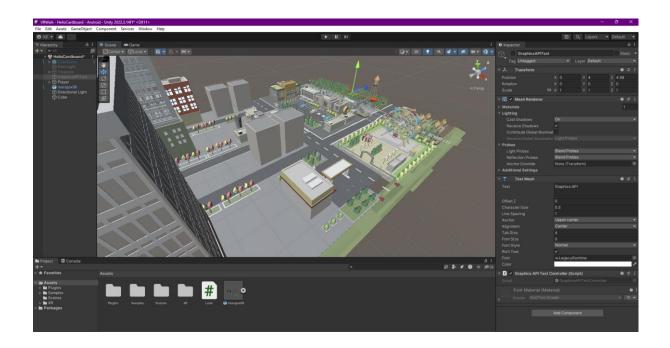
### 10.3. Moving in a VR Scene

**Cardboard** devices, except **Google Daydream**, usually do not have a control device. Therefore, the application is passive in a space and is done by looking around. So, is there no way to move around in the scene? Yes, there is! This section explains how this can be done with **C# Script** codes.

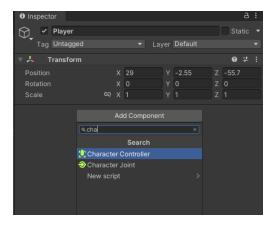
We have achieved 3D experience with the VR Room APK file and a cardboard-type headset. Now, first deactivate the ready scene, namely the **Cube Room, GraphicsAPIText, Point Light** and **Treasure** objects, by deselecting them in the **Inspector**.

ӨКЕ 🕶 📥	Inspector						
'≔ Hierarchy 🔒 🗄	GraphicsAPI	ſext					Sta
+ ▼ • All 🛛	Tag Untagged			🔹 Laj	yer	Default	
	🔻 🙏 🛛 Transform						0
Point light	Position		x O			4	4.99
► 💮 Treasure	Rotation		X 0			0	0
GraphicsAPIText	Scale	Ø	X 1			1	1
▶ 🏵 Player	🔻 🔣 🖌 Mesh Renderer						0

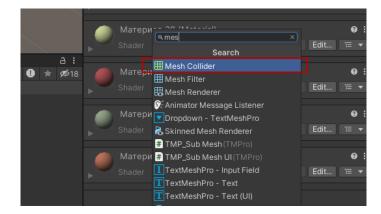
In the **Asset** section, download the **Sketchfab>Low Poly City**, **fbx** file that we used before and drag it to the stage.



Position the **Player** object in the **Hierarchy** in the city. Add the **Character Controller** component to the **Player** with **Add Component** in the **Inspector**.



To prevent the player from passing through the city and falling under it, select the city object and add a **Mesh Collider** with an **Add Component** in the **Inspector**.



Also, create the **C# Script** file named **Look.cs**, whose codes are given below, in the **Assets** section and write the following script.

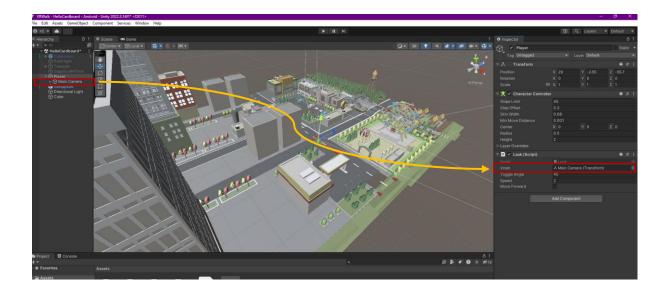
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Look : MonoBehaviour
  public Transform vcam;
  public float toggleAngle=45.0f;
  public float speed = 2.0f;
  public bool moveForward;
  private CharacterController cc;
  // Start is called before the first frame update
  void Start()
  {
    cc = GetComponent<CharacterController>();
    //DontDestroyOnLoad(this);
  }
  // Update is called once per frame
  void Update()
  {
    if (vcam.eulerAngles.x <= toggleAngle && vcam.eulerAngles.x < 90.0f)
    {
      moveForward = true;
    }
    else
    {
      moveForward = false;
    }
      if (moveForward)
      {
         Vector3 forward = vcam.TransformDirection(Vector3.forward);
         cc.SimpleMove(forward * speed);
      }
    }
  }
```

The code file ensures that the **Player** stands still when the camera is viewed 45 degrees up and down and moves forward at angles between them (*toggleAngle=45.0f*).

In the **Inspector**, when you look at **Look (script)**, you will see that the **Vcam** section is empty and says **None (Transform)**.

Inspector								a	:
💬 🗸 Player									-
Tag Untagged			▼ La	yer	Default				T
▼ 🙏 Transform							0		:
									-
Position									_
Rotation									_
Scale									
🔻 😃 🗹 Character Contro	oller								:
Slope Limit		4	5						
Step Offset									
Skin Width			.08						
Min Move Distance			.001						
Center									
Radius									
Height									
▶ Layer Overrides									
🔻 ≢ 🖌 Look (Script)							0		:
Script									۲
Vcam			None (Tran	sfor	·m)				$\odot$
Toggle Angle									
Speed									
Move Forward									
		Ad	d Componer						

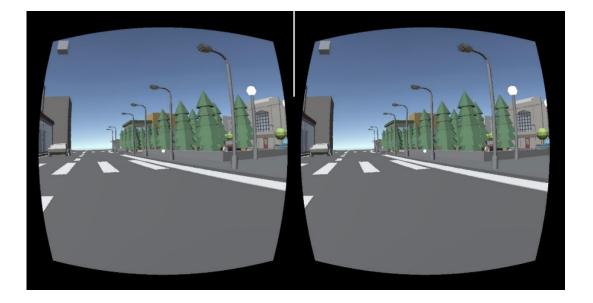
Drag and match the Main Camera here, which is located under the Player in the Hierarchy.



Go to **Build Settings**. While our phone is connected, create our **APK** file by pressing **Build and Run**.

	Build Android			×
	← → ∽ ↑ 💼 → Bu bilgisayar → Yeni Birim (D:) UNITY → VRW	alk > APK	✓ C APK M	م lasöründe ara
Build Settings				· · · · · · · · · · · · · · · · · · ·
Scenes In Build Samples/Google Cardboard XR Plugir	Düzenle 🔻 Yeni klasör			🗐 - (
		ğiştirme tarihi Tür Bo	yut	
	🚅 G:\ 💉 📴 VRWalk.apk 28	.04.2024 15:13 APK Dosyası		
	Madencilikte VR			
	🛅 Blender - A Tem			
Platform	BALYAGROUND			
Windows, Mac, Linux	🗖 Ders Notları - Ka			
Dedicated Server				
Android (	🗸 💻 Bu bilgisayar			
WebGL	> 🐸 Windows (C:)			
Universal Windows Platform	> 🛥 Yeni Birim (D:)			
iOS ios	> 🐏 Ağ			
≂ra PS4	Dosya adı: VRWalk.apk			~
ers PS5	Kayıt türü: apk (*.apk)			Ŷ
<				Kaydet    ptal
Asset Import Overrides     Max Texture Size No Override	<ul> <li>Klasörleri Gizle</li> </ul>			rayuer iptai
Texture Compression No Override				
Player Settings		Build 🔻 Build And Run		

We can open the application on our phone and do the first check. We can test the motion control by turning the mobile device in all directions.



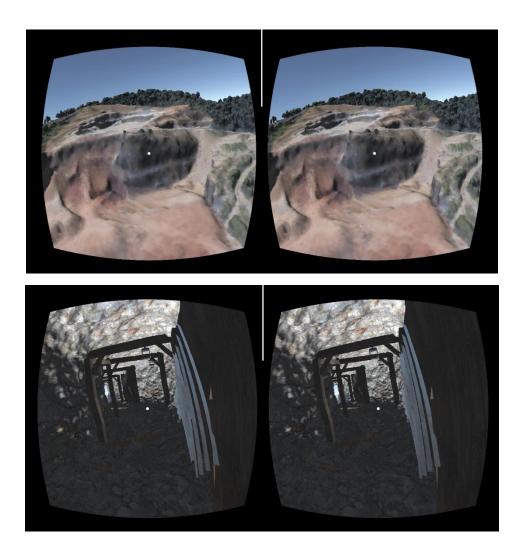
As a result, we can observe the result in 3D by placing our phone on our cardboard headset. We can control it by moving our head left and right and up and down and experiencing the VR application.

Some phones may require the **Cardboard Google LLC** app to be installed from **Google Play** for this app to work.

On Google Play, with keywords like **VR Player** and **VR Converter**, we can find applications that convert our mp4 etc. files to VR (for example **VR Video Converter** and **VR Game**).

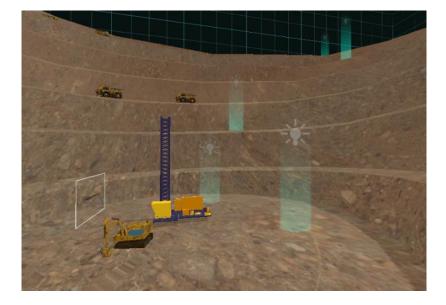
There are also many videos in VR standards on YouTube. For example, a search with keywords such as VR Video can find many videos suitable for VR viewing and can be watched in 3D on a VR headset.

Similarly, for open pit and underground mine gallery scenes, it is possible to deploy on mobile device and visualize in Google Cardboard headset.

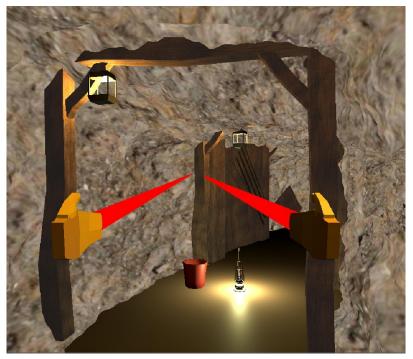


We can walk through and control movements in the scenes by using the C# codes.

Although the Meta Oculus Quest 2/3 tutorial is not taken here, some outputs are given to show the ability to have VR deployments for mining cases. Here, Meta and Unity MRTK (Mixed Reality Toolkit) templates are utilized.







## 11. AUGMENTED REALITY – AR APPLICATIONS

## 11.1.Augmented Reality – AR

**Augmented reality** can be thought of as the merging of virtual assets with the real world. If there is the possibility of interactive management and intervention with these assets, the term **Mixed Reality** is used for this concept.

For augmented reality, special glasses called **Smart-Glass**, which are usually expensive, are used. However, smartphones or tablets that everyone can have can also be used for this purpose, although they do not provide the same experience.

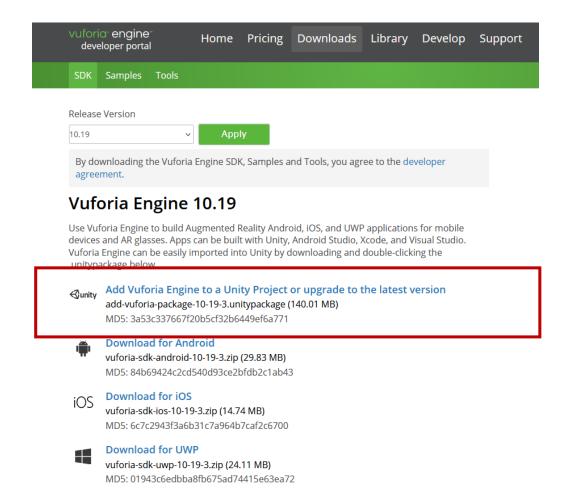
Various engines can be used to develop AR applications with Unity. Examples include **AR Foundation**, **AR Core, AR Kit**, and **Vuforia**. Due to its practical use and reason, this section will show the **Vuforia** application.

## 11.2.Vuforia AR Engine

This software has **Asset** downloads that work integrated with the **Unity Asset Store**. For this purpose, an account must first be opened on the **Vuforia** website. Using the same email address as Unity will be advantageous.

vuforia <sup>,</sup> engine <sup>,</sup>			
developer portal	Home Pricing Downloac	ls Library Develop Supp	DORT Log In Register
		a Developer Accou nload development tools, get e in the Vuforia community.	
	First Name *	Last Name *	
	Company *	Select Country of Residence * $\sim$	
	Email Address *	Username *	
	Password *	Confirm Password *	
	□ I agree to the terms of the Vuforia □	reCAPTCHA Gizlilik - Şərtlər	
	acknowledge that my personal det accordance with PTCs privacy polic purposes by PTC Inc. its subsidiarie Network, solely for the promotion of services.	ails will be processed in and may be used for marketing and members of the PTC Partner	

Vuforia's Unity app is located under the **Downloads** menu, but it is not required for this project.



Here it will be necessary to use the **License Manager** and **Target Manager** titles for preliminary preparation.

vuforia <sup>,</sup> engine <sup></sup> developer portal	Home	Pricing	Downloads	Library	Develop	Support	Hello KaanErarslan   Log Out	t
Account Manager	License Manager	Target Mar	nager Credenti	als Manager				
Dashboard Account Info Account Usage Payment Method	New to V track im Vuforia		to the docs and le s and environmer		D U D	ownload the SDK	<b>DK and Tools</b> for Unity, Android, IOS and ria with many samples available.	
Billing Statements		PI Reference	2			ownload Sample: ownload Tools	5	
	Vufor	ia Relea	ises		S	upport		
	Vuforia Release Release		erres	st releases o	si V D	et support for yo upport options. iew latest issues viscuss topics ontact us	ur issues by using our various	

First, let's select this menu and use the **Get Basic** button in the window that appears to create a **Database License** with the **License Manager**.

vuforia <sup>-</sup> engine <sup>-</sup> developer portal	Home	Pricing [	Downloads	Library	Develop	Support	Hello KaanErarslan   Log Out
Account Manager	License Manager	Target Manag	ger Credenti	als Manager			
License Ma	nager				Get Basi	c Buy Premiur	n Buy Cloud Add On
Learn more about Create a license ke		ion.					
Search							
Name		Primary UUID	і Туре	2	Statu	s 🗸	Date Modified
er		N/A	Basi	с	Active	2	Jun 04, 2023
C. d		N/A	Basi	с	Active	2	Jan 28, 2021
kaWks		N/A	Basi	с	Active	2	Jan 14, 2021

In this window, specify the License name and check the agreement box. Continue by clicking Confirm.

vuforia <sup>,</sup> engine <sup></sup> developer portal	Home	Pricing	Downloads	Library	Develop	Support	Hello KaanErarslan   Log Out
Account Manager	License Manager	Target Mar	nager Credenti	als Manager			

Back To License Manager

## Add a license key to your Basic plan

License Name * GSF-GIT	
You can change this later	
By checking this box, I acknowledge	that this license key is subject to the terms and conditions of the Vuforia Developer Agreement.
Cancel Confirm	

Now, a **license** has been opened with the name we created on our list.

vuforia <sup>®</sup> engine <sup>®</sup> developer portal	Home	Pricing D	ownloads	Library	Develop	Support	Hello KaanErarslan   Log Out
Account Manager L	icense Manager	Target Manage	er Credenti	als Manager			
License Mana	ager				Get Basi	c Buy Premiu	m Buy Cloud Add On
Learn more about lice Create a license key fo		n.					
Search							
Name	P	Primary UUID ①	) Туре	2	Statu	s 🗸	Date Modified
GSF-GiT	Ν	I/A	Basi	с	Active	2	Jan 01, 2024
ers	Ν	I/A	Basi	c	Active	2	Jun 04, 2023
	Ν	I/A	Basi	с	Active	2	Jan 28, 2021
L.kaW.ks	Ν	I/A	Basi	с	Active	2	Jan 14, 2021

Next step, we can go under the **Target Manager**. Here, click **Add Database** to create a database.

vuforia <sup>,</sup> engine <sup>,</sup> developer portal	Home	Pricing	Downloads	Library	Develop	Support	Hello KaanErarslan	Log Out
Account Manager	License Manager	Target Mar	nager Credenti	als Manager				
Target Mana	ager						Add Databa	ase
Use the Target Man	ager to create and	d manage d	atabases and ta	argets.				
Search								
Database		ту	/pe		Targets		Date Modified	
RK t	l	D	evice		3		Feb 21, 2021	
R R	I	D	evice		1		Jun 04, 2023	
on Ba_tes d€	I	D	evice		8		Feb 10, 2021	
Ca d		D	evice		5		Feb 16, 2021	

Enter the name of the database; here, do not use special characters (such as some Turkish characters) and spaces like in web page names. **Device** option is suitable as **Type**. After naming, create our database with the **Create** button.

Manager			Add Database
get Manager to create	and manage databases and targets.	_	
	Create Database	_	
	Database Name * GSF-GiT-Veri-Tabani	Date Modified	
	<u></u>	Feb 21, 2021	Ť
	Туре:	un 04, 2023	Î
sinden	<ul><li>Device</li><li>Cloud</li></ul>	-eb 10, 2021	î
	O VuMark	-eb 16, 2021	Î
		Cancel Create	

Our database will be visible in the list.

vuforia <sup>,</sup> engine <sup>,</sup> developer portal	Home	Pricing	Downloads	Library	Develop	Support	Hello KaanErarslan   Log Out
Account Manager	License Manager	Target Man	ager Credenti	als Manager			

#### **Target Manager**

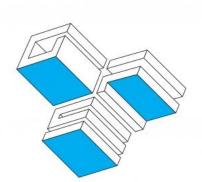
Search

Use the Target Manager to create and manage databases and targets.

Database	Туре	Targets	Date Modified	
RK t	Device	3	Feb 21, 2021	Î
	Device	1	Jun 04, 2023	Î
- Barcender	Device	8	Feb 10, 2021	Î
GSF-GiT-Veri-Tabani	Device	0	Jan 01, 2024	Ť
	Device	5	Feb 16, 2021	Ť

Now, prepare to add an object to the database that opens empty, or you can also use a ready-made image file.

Here, we set our target image as the **GSF Faculty** logo. Take the high-resolution file from the faculty page to the hard disk. File name: **GSF-LOGO.png** 



# KÜTAHYA DUMLUPINAR ÜNIVERSITESI GÜZEL SANATLAR FAKÜLTESI

To add this image to our database in Vuforia, all we need to do is type the phrase "**GSF-GiT-Database**" that we just created in **Target Manager** and whose name we see in the list. The window that opens allows us to upload various visual objects to the database. This object can be a **JPEG** or **PNG** file. It is important to note that the quality of the visual object must be high. For **Image Target** type applications, **Add Target** is selected to upload the visual file we plan to upload.

vuforia: engine developer portal	e <sup>∞</sup> Home	Pricing Dov	vnloads Library	Develop S	<b>upport</b> Hello K	aanErarslan 🗸   Log Out
Account Manager	License Manager	Target Manager	Credentials Manager			
Target Manager > GSF-0	iiT-Veri-Taba					
GSF-GiT-Ver	i-Tabani <sub>Edit</sub>	Name				
Targets (0)	_					
Add Target						Download Database (All)
Target Name		Туре	Rating ①	Statu	Date N	lodified

The window that opens offers options to upload files to the database.

Туре:			
	$\bigcirc$	07	
Image	Multi	Cylinder	Object
File:			
			Browse
.jpg or .png (max fi	le 2mb)		
Width:			
Enter the width of same scale as your	augmented virtual c	units. The size of the ta ontent. Vuforia uses n culated when you uple	neters as the defa
Enter the width of same scale as your	augmented virtual c	ontent. Vuforia uses n	neters as the defa
Enter the width of same scale as your unit scale. The targ Name:	augmented virtual co get's height will be cal	ontent. Vuforia uses n	neters as the defa bad your image.

Now, upload our target file to the **File** section with **Browse**. The file name will be seen as **GSF-LOGO.png**. Write **1** to the **Width** section. **GSF-LOGO** automatically appears in the **Name**. Add it with the **Add button**.

Add Target			
Туре:			
	$\bigcirc$	07	
Image	Multi	Cylinder	Object
File:			
GSF-LOGO.png			Browse
.jpg or .png (max file	2mb)		
Width:			
1			
	ugmented virtual co	ontent. Vuforia uses	target should be on th meters as the default lload your image.
GSF-LOGO			
Name must be unique this will be reported		hen a target is detec	ted in your application
			<b></b>
		Cancel	Add

You may be asked to change the properties of the file:

Add Target			
Туре:			
	$\bigcirc$	$\Box \nabla$	
Image	Multi	Cylinder	Object
File:			
GSF-LOGO.png			Browse
.jpg or .png (max file <b>Width:</b> 1	: 2mb)		
same scale as your a	augmented virtual c	units. The size of the ta ontent. Vuforia uses n culated when you upl	neters as the default
GSF-LOGO			
		hen a target is detecte	ed in your application

To overcome the problem, the file was saved in **JPEG** format with **Paint** and accepted as a problemfree file. Our file appears in the database list that opens. At this stage, **Vuforia** performs the test that appears with the phrase "**Processing**" to determine the file **quality**.

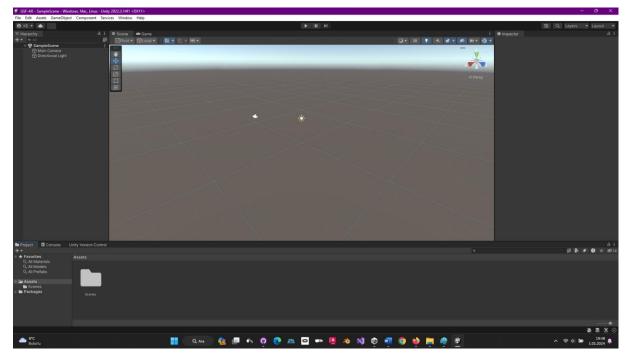
Target Manager > GSF-GIT-Veri-Taba					
GSF-GiT-Veri-Tabani Edit Name Type: Device					
Targets (1)					
Add Target				Download Database (All)	
Target Name	Туре	Rating (i)	Status 🗸	Date Modified	
GSF-LOGO	Image	*****	Processing	Jan 01, 2024 19:38	

When we refresh the page after waiting for a short time, it displays the **quality level (Rating)** out of 5 stars. **5 stars** is the strongest considered file quality. The system can work up to **3 stars**, but it is likely to break in the mobile application.

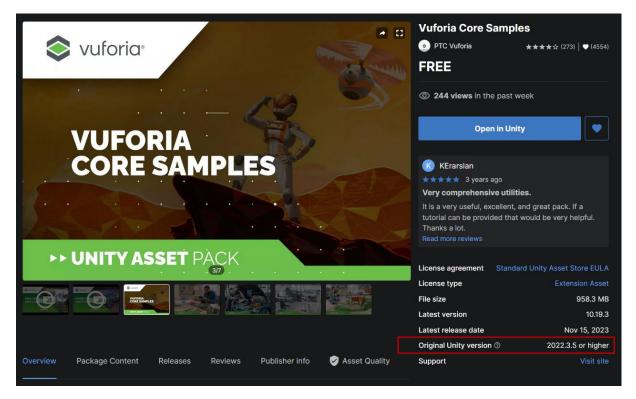
vuforia: engine: Hor developer portal	ne Pricing Download	s Library De	velop Support	Hello KaanErarslan 🗸 丨 Log Out
Account Manager License Manaş	ger Target Manager Crede	ntials Manager		
Target Manager > GSF-GiT-Veri-Taba				
GSF-GiT-Veri-Tabani Type: Device	Edit Name			
Targets (1)				
Add Target			_	Download Database (All)
Target Name	Туре	Rating (i)	Status 🗸	Date Modified
GSF-LOGO	Image	****	Active	Jan 01, 2024 19:38

## 11.3.VUFORIA integration in UNITY project

Now, continue by opening a project in Unity with version **2022.3.5 or higher**. In this application, version 2022.3.14f is used.

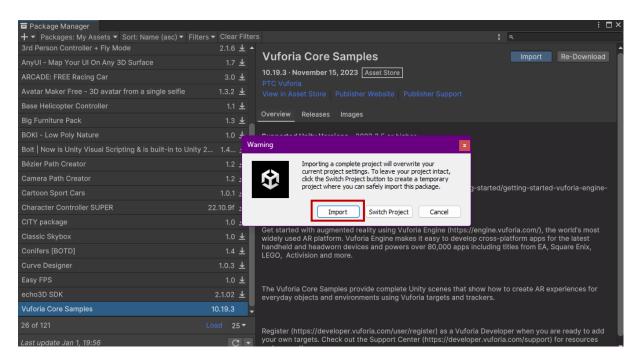


Log in to the **Unity Asset Store** with **Windows->Asset Store**. Reach the **PTC Vuforia Core Samples** section by typing **Vuforia Core Samples**. If you notice, Unity requires a version of **2022.3.5** or **higher**.



Now, add it to our assets ("Add to Assets") and open it in our project ("Open in Unity"). If we are using it for the first time, perform the Install process. Then Import.

Add it to our assets ("Add to Assets") and open it in our project ("Open in Unity"). If we are using it for the first time, go on the Install process. Then, import.



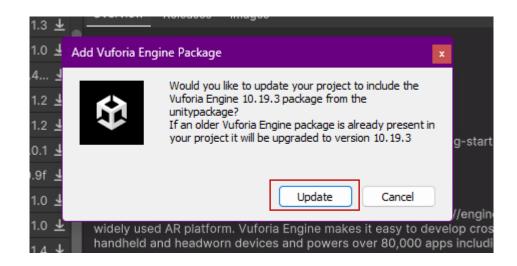
We can accept the warning that there are some updates.

🖬 Package Manager			: 🗉
+ ▼ Packages: My Assets ▼ Sort: Name (asc) ▼ Filters	s 🕶 Cle	ear Filte	
3rd Person Controller + Fly Mode		.6 <u>+</u> 4	
AnyUI - Map Your UI On Any 3D Surface		.7 <u>¥</u>	Vuforia Core Samples Import Re-Download
ARCADE: FREE Racing Car		.0 <b>±</b>	10.19.3 · November 15, 2023 Asset Store PTC Vuforia
Avatar Maker Free - 3D avatar from a single selfie	1.3	.2 <u>¥</u>	
Base Helicopter Controller		.1 <u>¥</u>	
Big Furniture Pack		.3 <u>+</u>	Overview Releases Images
BOKI - Low Poly Nature		.0 🛓	
Bolt   Now is Unity Visual Scripting & is built-in to Unity 2	1.4	J V	/arning x
Bézier Path Creator		.2 🚽	This Unity Package has Package Manager dependencies.
Camera Path Creator		.2 🚽	M. State and the second second second second second second second second second second second second second se
Cartoon Sport Cars		.1 🚽	g-started/getting-started-vuforia-engine-
Character Controller SUPER	22.10.9	əf <u>4</u>	Install/Upgrade Skip
CITY package		.o <u>4</u>	
Classic Skybox		.0 <b>±</b>	Get started with augmented reality using Vuforia Engine (https://engine.vuforia.com/), the world's most widely used AR platform. Vuforia Engine makes it easy to develop cross-platform apps for the latest
Conifers [BOTD]		.4 <b>±</b>	handheld and headworn devices and powers over 80,000 apps including titles from EA, Square Enix, LEGO, Activision and more.
Curve Designer		.3 <u>+</u>	
Easy FPS		.o <b>±</b>	
echo3D SDK	2.1.0	2 🛨	The Vuforia Core Samples provide complete Unity scenes that show how to create AR experiences for everyday objects and environments using Vuforia targets and trackers.
Vuforia Core Samples	10.19	.3 .	
26 of 121 l		25 🔻	Register (https://developer.vuforia.com/user/register) as a Vuforia Developer when you are ready to add
Last update Jan 1, 19:56		G -	your own targets. Check out the Support Center (https://developer.vuforia.com/support) for resources

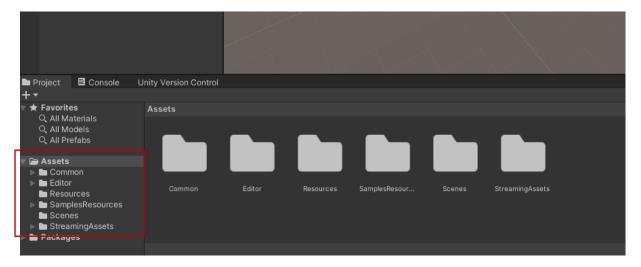
Get all the files by clicking Next.

asc) ▼ Filters ▼ Cle		: Q
1	6 ⊈ ▲ 7 ⊈ Vuforia Core Samples 0 ⊈ 10.19.3 · November 15, 2023 Asset Store	Import
gle selfie 1.3. 1 1	Import Unity Package	' Support
1 11 ilt-in to Unity 2 1.4	Import Content Step 1 of 2	
1 1.0 22.10.9 1	CommonEditor.asmdef	.com/getting-started/getting-started-v a in Unity. gine (https://engine.vuforia.com/), the v
1 1 1.0 1	✓     IF UsersionChecker.cs     New       ✓     IF VersionChecker.cs     New       ✓     Fonts     New       ✓     As SourceSansPro.ttf     New       ✓     Loops     New	site develop cross-platform apps for 0,000 apps including titles from EA, Sq
2.1.0 10.19.	✓ ⊠ BackGround_x081.png New ✓ ⊠ BackGround_x108.png New	cenes that show how to create AR exp rgets and trackers.
Load	<ul> <li>✓ SB BackGround_x162.png</li> <li>✓ SB BackGround_x216.png</li> <li>✓ SB BackGround_x324.png</li> <li>✓ SB BackGround_x324.png</li> <li>✓ SB ForeGround_x081.png</li> <li>✓ SF ForeGround_x108.png</li> <li>✓ SF ForeGround_x108.png</li> <li>✓ SF ForeGround_x126.png</li> <li>✓ SF ForeGround_x216.png</li> <li>✓ SF ForeGround_x216.png</li> <li>✓ SF ForeGround_x324.png</li> </ul>	r) as a Vuforia Developer when you are ps://developer.vuforia.com/support) fc
	Cancel Next	

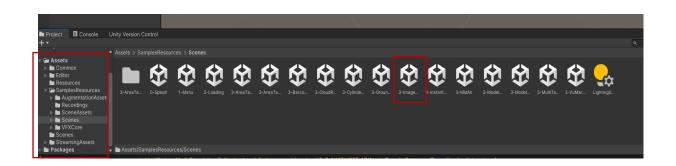
Confirm the information about the update with the **Update** button.



Go to our **Scene** in the project and see the changes under Assets.



When we go to **Assets->SampleResources->Scenes**, we see that there are many ready-made scenes. Here, **double-click** the **3-Image Target** scene.

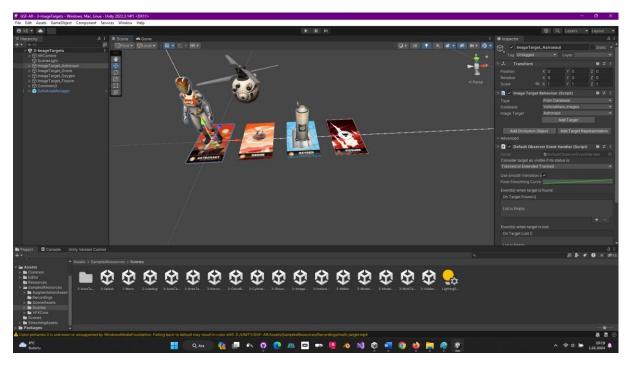


By **Image Tracking**, we will see that a **UI Canvas** has been opened. At the same time, new **GameObjects** have been opened in the **Hierarchy** window.

<ul> <li>GSF-AR - 3-imageTargets - Windows,</li> </ul>				/							- 0 ^
File Edit Assets GameObject Con	nponent Servi	ices Window Help				I BERNELLAS SAN				Internal Antonio	
Ө КЕ • 📥					100	H H					Løyers • Løyout •
Hierarchy     +      •      or		🖶 Scene 🔊 Game									
<ul> <li>♥ 3-imageTargets</li> <li>▷ ARCainera</li> <li>○ SceneLight</li> <li>▷ ImageTargetAstronaut</li> <li>▷ ImageTargetDrone</li> <li>▷ ImageTargetDrone</li> <li>▷ ImageTargetFisure</li> <li>▷ OmageTargetFisure</li> <li>▷ OmageTargetFisure</li> <li>▷ OmageTargetFisure</li> </ul>		Constantial Constantia Constantia Constantia Constantia Constantia Constantia	<b>₩</b> • 5 - ₩•				D.	20 • « « • •	< Persp	3-Image Targets (Scene As	set) 🔮 i Openi
i a 🖬 SafakeadAmager											
										Asset Labels	
											· · · · · · · · · · · · · · · · · · ·
											<ul> <li>None</li> </ul>
Project Console Unity Ve											8 i
	ts > SamplesR										ar er er er støris
	urea Ta. D-Sp ssets/Samplesf		g 3-AreaTa 3-AreaTa Targets.unity	\$\$ \$\$ \$	a. 3-Groun. <b>3-Image.</b>	3-Instant. 3-Misar	3-Model_ 3-Model_ 3-M	ð & <del>.</del> .			
Color primaries 0 is unknown or unsu	upported by Wi	IndowsMediaFoundation, Fa	ling back to default may n	suft in color shift. D/UNITY/GSF	AR/Assets/SamplesResou						× 2 0
S*C Buluthu			Q AN	- 🦀 💷 🔨 (	o 💽 🙉 🖸	🚺 🦚	N) 💿 📲 🥥	📦 🤚 🧟 👳			

When we double-click on any of the objects starting with **Image**, sample objects in the 3D part of the canvas will be seen.

We aim to import the **GSF-LOGO** into our database, place the 3D object we want on it, and even play the video.



In order to perform these operations, we need to access the information and database in **Vuforia** and import it into the Unity project.

# 11.4.Getting the License Key and Downloading Database from Vuforia to Unity

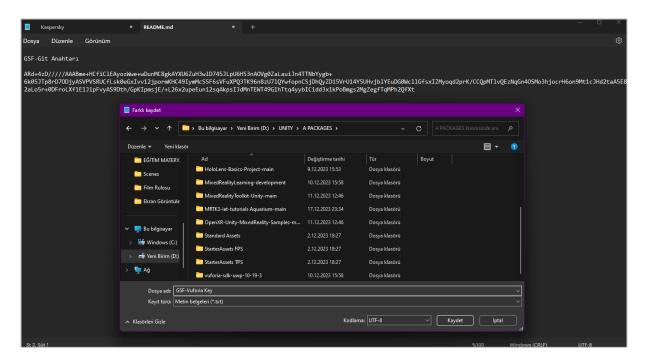
vuforia <sup>,</sup> engine <sup>,</sup> developer portal	Home	Pricing	Downloads	Library	Develop	Support	Hello KaanErarslan   Log Out
Account Manager	License Manager	Target Mana	ager Credenti	als Manager			
License Mar	nager				Get Basic	: Buy Premiu	m Buy Cloud Add On
Learn more about li Create a license key		on.					
Search							
Name		Primary UUIE	) i) Type	e	Status	~	Date Modified
GSF-GIT		N/A	Basi	c	Active		Jan 01, 2024
ers in a second		N/A	Basi	c	Active		Jun 04, 2023
EC		N/A	Basi	с	Active		Jan 28, 2021
ka Wiks		N/A	Basi	c	Active		Jan 14, 2021

Let's go to the License Manager section in Vuforia and select GSF-Git.

On the page that opens, there is a **license key** prepared for use. When we click on it with the mouse, this long key is copied to memory.

vuforia <sup>,</sup> engine <sup>,</sup> developer portal	Home	Pricing I	Downloads	Library	Develop	Support	Hello KaanErarslan   Log Out
Account Manager	License Manager	Target Mana	ger Credent	ials Manager			
License Manager > GSF-	GIT						
GSF-GIT Edit N	ame Delete License Key	,					
License Key	Jsage						
Please copy the licen	se kev below into vo	ur app					
ARd+4zD////AAABme ODjyASVPVSRUCfLskC	+HCfiClEAyozWwe+wDu DeGxIvv12jpormKHC491	unMC8gkAYXU6Zu LymMcS5F6sVFuX	PQ3TK96n8zU71	QYwfopnCSjDh	QyZD15VrU14Y5	UHvjblYEuDGOW	
	<pre>K/CCQpMTlvQEzNqGn405 sqAkpsIJdMnTEWT49G1</pre>	-				ASSDEN/GDKIDM	
Plan Type: Basic							
Status: Active Created: Jan 01, 2024	4 18:58						
License UUID: 6bf6d		8cedabe1bc8					
History:							
License Created - Too	day 18:58						

To use this key copied to memory without entering Vuforia every time, we can copy it to Notepad and save it as a text file.



Now, go to the **Target Manager** in Vuforia. Our goal is to prepare and download the database for Unity.

vuforia <sup>,</sup> engine <sup>,</sup> developer portal	Home	Pricing Dov	vnloads Lib	rary Develop	Support	Hello KaanErarslan   Log Out
Account Manager	License Manager	Target Manager	Credentials Ma	anager		
Target Man	ager					Add Database

Search				
Database	Туре	Targets	Date Modified	
RK	Device	3	Feb 21, 2021	i
ER STATE	Device	1	Jun 04, 2023	
on≓Ba'tesi de	Device	8	Feb 10, 2021	
GSF-GiT-Veri-Tabani	Device	1	Jan 01, 2024	
ECHid	Device	5	Feb 16, 2021	

Now, switch to our database named **GSF-Git-Database** with this key. To download the database that we want to download and have loaded one shape (**GSF-LOGO.png**) for now, select the box of our shape as **Target Name** and press **Download Database**.

Account Manager	License Manager	Target Manager	Credentials Manager							
Target Manager > GSF-GIT-Veri-Taba										
GSF-GiT-Ver	GSF-GiT-Veri-Tabani Edit Name									
Targets (1)										
Add Target					Download Database (1)					
Target Name		Туре	Rating (i)	Status 🗸	Date Modified					
1 selected Delete										
Sister GSF-LOC	60	Image	****	Active	Jan 01, 2024 19:38					

Here, Unity Editor should be selected from the options presented to us and **Download** should be clicked.

Download Data	abase							
1 of 1 active targets will b	1 of 1 active targets will be downloaded							
Name: GSF-GIT-Veri-Tabani								
Select a development pl	atform:							
O Android Studio, Xcode	e or Visual Studio							
O Unity Editor								
	Cancel	Download						

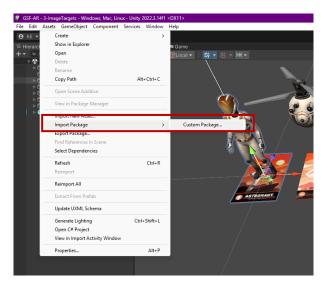
The database compilation process begins.



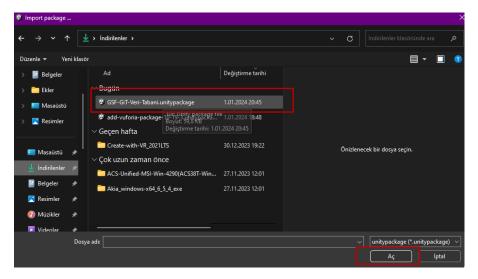
We can see the downloaded **unitypackage** file on the hard disk.

🕫 GSF-GiT-Veri-Tabani.unitypackage	1.01.2024 20:45	Unity package file	60 KB

Go back to our Unity scene. Here we go to the Assets->Import->Package->Custom Package submenu.



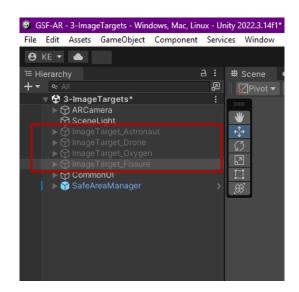
Now, add the package file we just downloaded from Vuforia to our project.



Then, Import.

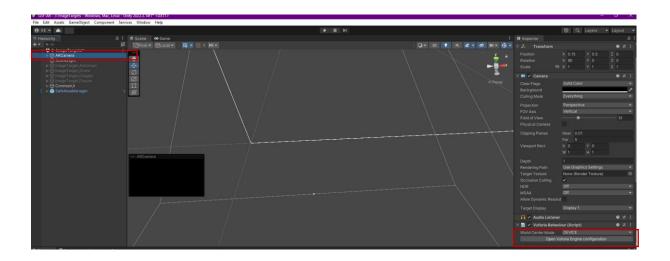


Make the sample objects such as **Astronaut, Drone, Oxygen,** and **Fissure** in the **Hierarchy** section of our scene invisible by clicking on their boxes.

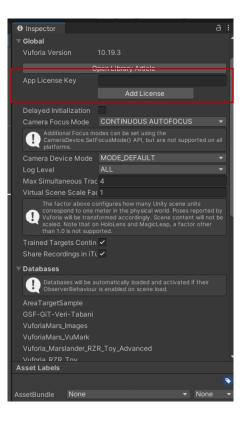


Now it's time to get our database object instead.

Go to **ARCamera** and open the settings in the **Inspector** section. Press **Vuforia Engine** configuration.



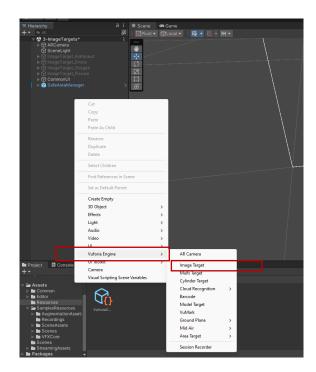
Here, a window will appear where we will enter the Vuforia **database key**. For this purpose, let's see the box called **App License Key**. As you may recall, the key that we memorized in Vuforia **License Manager** and that we can save in **Notepad** so that we can use it in our future work will be placed in this box.



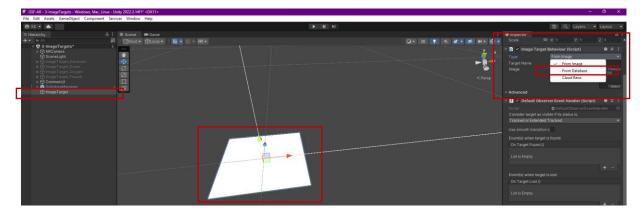
Paste the license key...



After removing the sample objects, we need to place our **Image** object. To do this, select **Vuforia>Image Target** from the menu that opens with the right click of our mouse in the **Hierarchy** section.



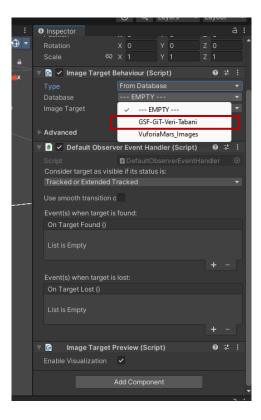
A plane-like object is placed in our scene. Open the menu to the **Inspector->Type** section of this object. Click on the **From Database** option in this menu.

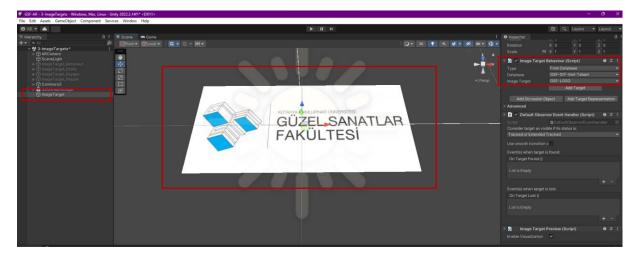


In the new settings that appear, there is a **Database** option under **Type**. We will import the database that we created in Vuforia and download it in **unitypackage** format here.

:	Inspector				а:
€	Rotation	x 0	Y 0	z 0	
2					
	🔻 健 🖌 Image Target B	Behaviour (Scr	ript)	0	
x	Туре	From Databa			
Г	Database	EMPTY -			
	Image Target	EMPTY -			Ŧ
		Add	Target		
	▶ Advanced				
	🔻 # 🖌 Default Observ	er Event Hand	dler (Script)	0	
		🖩 DefaultOb			
	Consider target as visi	ole if its status			
	Tracked or Extended 1	racked			
	Use smooth transition				
	Event(s) when target is	found:			
	On Target Found ()				
	List is Empty				
	Event(e) when terret is			+	
	Event(s) when target is On Target Lost ()	iost:			
	List is Empty				
				+	
	🔻 🕼 🛛 Image Target I	Preview (Scrip	ot)	0	
_	Enable Visualization	~			
		Add Componei			

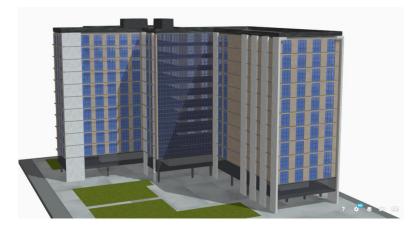
When you click on the Database section, you see our database in the list that opens. Select it...

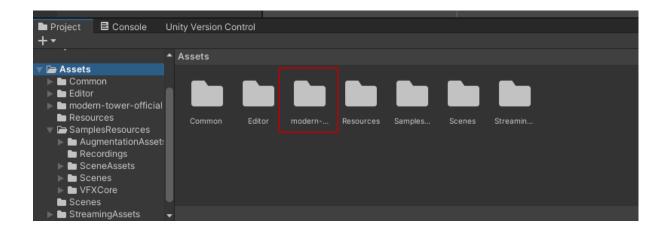




With the selection, the **GSF-LOGO** figure will be placed on the **Image** on the stage.

We plan to place a **3D school model** on the **Image** in the scene. We find a low poly building from **Sketchfab** and place it in the **Asset** folder.





This 3D building model is placed on top of the **GSF-LOGO** plate taken from our database after various size reductions and positionings.



Now, it is time to deploy our mobile device. For this, the **Build Settings** window opens. There are many scenes listed in the **Scenes in Build** section.

/	Provenue			
	Build Settings			: 🗆 × 🛛 < F
	Scenes In Build			
	<ul> <li>SamplesResources/Scenes/Io-Splash</li> <li>SamplesResources/Scenes/I-Menu</li> <li>SamplesResources/Scenes/I-Loading</li> <li>SamplesResources/Scenes/I-CloudReco</li> <li>SamplesResources/Scenes/I-CroundPlar</li> <li>SamplesResources/Scenes/I-GroundPlar</li> <li>SamplesResources/Scenes/I-ImageTarg</li> <li>SamplesResources/Scenes/I-ImageTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-ModelTarg</li> <li>SamplesResources/Scenes/I-Amatra</li> </ul>	rgets re geTargets etsStandard etsAdvanced ts		0 1 2 3 4 5 6 7 8 9 10 11 11 12 13 14 15
				Add Open Scenes
	Platform	🖵 Windows, Mac, Linux		
	🖵 Windows, Mac, Linux 🛭 🕸	Target Platform	Windows	<b></b>
	Dedicated Server	Architecture Copy PDB files	Intel 64-bit	
	Android	Create Visual Studio Solution Development Build		
	WebGL	Autoconnect Profiler Deep Profiling		
	📒 Universal Windows Platform	Script Debugging Compression Method	Default	
lex-building-ar				
	Prs PS5			

However, since our project is about **Image Target/Tracking**, the boxes of the ones other than **Image Targets** are checked and excluded from the process.

Also, since this will be a mobile application, the **Switch Platform** process is done by selecting the **Android** platform.

Build Settings			: 🗆 ×				
Scenes In Build							
SamplesResources/Scenes/0-Splash							
SamplesResources/Scenes/1-Menu							
SamplesResources/Scenes/2-Loading							
SamplesResources/Scenes/3-CloudReco							
SamplesResources/Scenes/3-CylinderTargets							
SamplesResources/Scenes/3-GroundPlane							
SamplesResources/Scenes/3-MidAir							
<ul> <li>SamplesResources/Scenes/3-ImageTarget</li> </ul>			0				
SamplesResources/Scenes/S-instantimage							
SamplesResources/Scenes/3-ModelTarget							
SamplesResources/Scenes/3-ModelTarget							
SamplesResources/Scenes/3-MultiTargets							
SamplesResources/Scenes/3-VuMarks							
SamplesResources/Scenes/3-AreaTargets							
SamplesResources/Scenes/3-AreaTargetC	apture						
SamplesResources/Scenes/3-Barcodes							
			Add Open Scenes				
			Add Open Scelles				
Platform	🗰 Android						
🖵 Windows, Mac, Linux 🛭 🕸 🛛		Use Player Settings	•				
	ETC2 fallback		<b>▼</b>				
Dedicated Server	Export Project						
📺 Android	Build App Bundle (Google Play)						
		Disabled					
5 WebGL	Create symbols.zip		· · · · · · · · · · · · · · · · · · ·				
	Run Device	Default device	✓ Refresh				
Universal Windows Platform			Patch And Run				
iOS ios	Development Build						
PJA PS4							
ers PS5							
·	Compression Method	LZ4	· · · · · · · · · · · · · · · · · · ·				
▼ Asset Import Overrides							
Max Texture Size No Override 🗸							
Texture Compression No Override -							
restare compression into oronade			Learn about Unity Build Automation				
Player Settings			Switch Platform Build And Run				

Now, we can connect our phone with developer features to the computer with a cable and **Refresh** the **Run Device** section.

Platfo	orm			🛱 Android					
Ţ	Windows, Mac	c, Linux	î	Texture Compression		Player Settings			-
•	Dedicated Ser	ver		ETC2 fallback Export Project	32-b	it			
ı Î	Android		Ŷ	Symlink Sources Build App Bundle (Google Play)					
5	WebGL			Create symbols.zip	Disal	bled			
				Run Device	Defa	ult device			Refresh
	Universal Wind	dows Platform			~	Default device			
					i:	All compatible devices			
iOS	iOS			Development Build		<enter ip=""></enter>			
7.14						Samsung SM N975F (RF8NB069L8L)			
ərs									
			•	Compression Method	LZ4				
Ass	et Import Overri	des							
Max T	exture Size	No Override							
Textu	re Compression	No Override					Learr	n about Unity Build A	Automation
Playe	er Settings						Bı	uild 🔻 Build	And Run

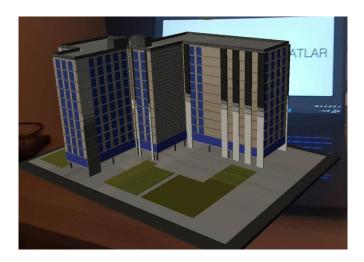
When we select the **Build and Run** section, in the window that opens, we create a new folder called **GSFLogoApp** with the right click of our mouse to register the application, and we specify the name of our **APK** file as **GSFLogo.apk**.

😨 Build Android						×
← → ∽ ↑ 🗖 → Bu bi	lgisayar > Yeni Birim (D:) > UNITY > G	SF-AR > GSFLogoApp		C		م
Düzenle 🔻 Yeni klasör					∎▼	•
🚞 EĞİTİM MATERY. 🛛 🗛		Değiştirme tarihi	Tür	Boyut		
🚞 Scenes		Aramanızla eşleşer	n öğe yok.			
🚞 Film Rulosu						
🚞 Ekran Görüntüle						
∽ 💻 Bu bilgisayar						
> 🐸 Windows (C:)						
> 🛋 Yeni Birim (D:)						
> 🋬 Ağ 🔰						
Dosya adı GSFLogo						~
Kayıt türü apk (*.apk)						~
<ul> <li>Klasörleri Gizle</li> </ul>				[	Kaydet İptal	

The printout process starts with the save button. It will take a few minutes to complete. Finally, it returns to our stage, stating that it was successful.

File Edit Assets GameObject Com	ponent Services Window Help						
Ө КЕ 🔹 📥			E E	11 H			Q Layers • Layout •
= Hierarchy	à i ∰ Scene 🛷 Game					i O Inspector	
+ * [ * All	8 Pivot - Clocal -				Q + 20 • 4 2 • 50		
A monget argets     Generation	* 0 11	PUI destings Societ and Societ an	e ets NTargets 2tsStandard 1thAdvanced 3 8			-Porp	
		Platform Undows, Mac, Linux Deficiated Server Android WebOL	Android     Texture Compression     ETC2 fallback     Export Project     Synthm Sources     Build App Bundle (Boogle Play     Cheate synthes.zip     Run Device	Disabled Default device	Add Open Scenes • • Refease		
Project Console Unity Ve		Universal Windows Platform	Build to Device Patching is disabled for Release to				a i
+ • Asset • Se Assets • De Common • De Editor • De Editor	s y modern-tower-efficial-complex-building-a						ga 🎝 🖌 🧿 👘 🔊 sé is
Resources     SemplesResources     Ba SugmentationAsset     Recordings     Ba SconeAssets     Ba SconeAssets     Ba SconeAssets		Max resture size recovernoe • Texture Compression No Override • Player Setsings			Learn about Unity Build Automation Build 🔹 Build And Run		
Build completed with a result of 'Suc	ceeded' in 348 seconds (348152 ms)						0 🛢 🕷
Bulutha		🚦 Q. Ara 🤷 💷 🗠	v 🧕 💽 🙇 🖻	📼 🖪 🔌 刘 💿	🐖 🧿 🧅 🧮 🖉 🗒		∧ ♥ 0) 🆢 22244

When the application was opened on the phone, the **GSF Logo** was placed on the computer screen and tested. As a result, the building was displayed in front of the screen as an augmented reality application.



This application can be added to the **displacement** and **rotation** operations called **Lean and Touch**.

It is also possible to place a **Plane** instead of placing a 3D model on the logo that comes to the stage as a plane and turn that **Plane** into a video player by adding a **Video Player**.

Another application alternative provided by Vuforia is the application called **Ground Plane**, which allows the desired 3D object and video player to be positioned on the **ground**, <u>independent</u> of any Image Target, because of ground scanning.

Again, after scanning a 3D space, augmented reality objects can be added to the desired objects or points.

It is planned to prepare additional notes for this and similar applications. However, independent of the notes, there are many supporting materials in digital media, primarily their resources.

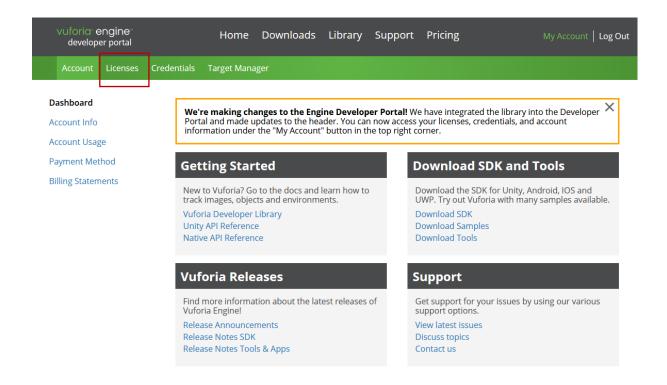
# 11.5.AR-Vuforia and Multi-Image Target

In addition to the single image-target application under the title of Augmented Reality, a similar study can be done on **multiple** images. Essentially, this means placing many images in the Vuforia Database and ensuring their recognition within an application of Unity. Therefore, a 3D model, animation or video can be assigned to each shape (picture). In this way, AR studies can also be done collectively on the pictures in the book (catalogue), called **AR-Book** (**Augmented Reality Book**).

**Vuforia AR** application is based on the principle of converting image file(s) uploaded to a database created here into a Unity package, downloading them and entering them into the Unity project. Let's repeat the upload and Unity package downloading process previously done for the **GSF logo** for the Engineering Faculty and Dumlupinar University logos.

It is thought that explaining and repeating some parts of the first application from the beginning would be beneficial, and there will be partial similarities with the previous explanation.

Let's log in to our **Vuforia** account.



Here we go to the Licenses tab.

All licenses we created before will be opened.

vuforia <sup>,</sup> engine <sup>,</sup> developer portal	Home Downlo	oads Library	Support Pricing	My Account   Log O
Account Licenses Credentia	ls Target Manager			
Licenses			Get Basic Buy Prer	nium Buy Cloud Add On
Learn more about licensing. Create a license key for your ap	plication.			
Search				
Name	Primary UUID 🛈	Туре	Status 🗸	Date Modified
GSF-GIT	N/A	Basic	Active	Jan 01, 2024
	N/A	Basic	Active	Jun 04, 2023
	N/A	Basic	Active	Jan 28, 2021
	N/A	Basic	Active	Jan 14, 2021
Showing 1-4 of 4	«	1	> >>	25 per page ▼

Click **GSF-GIT** and copy its license by clicking with our mouse. This license information will be needed in our Unity project.

Licenses GSF-GIT
GSF-GIT Edit Name Delete License Key
License Key Usage
Please copy the license key below into your app
ARd+4zD/////AAABme+HCfiClEAyozWwe+wDunMC8gkAYXU6ZuH3w1D745JLpU6H53nAOVg0ZaLauiJn4TTNbYygb+6k05JTp8rD7 ODjyASVPVSRUCfLsk0eGxIvv12jpormKHC49IymMc55F6sVFuXPQ3TK96n8zU71QYwfopnC5jDhQyZD15VrU14Y5UHvjb1YEuDG0W cllGfsxIZMyoqd2prK/CCQpMTlvQEzNqGn4OSMo3hjocrH6on9Mt1cJHdZtaA5E82aLo5r+0DFroIXflE1JipFvyAS9Dth/GpKIpm sjE/+L26x2upeEuni2sqAkpsIJdMnTEWT49G1hTtq4yybIC1dd3x1kPoBmgs2MgZegfTqMPh2QfXt
Plan Type: Basic
Status: Active
Created: Jan 01, 2024 18:58 License UUID: 6bf6d4b0d23447c3b64008cedabe1bc8
History: License Created - Jan 01, 2024 18:58

Now, go to Target Manager and select the title we previously named **GSF-GIT Database**.

vuforia <sup>,</sup> engine <sup>.,</sup> developer portal	Home Downloads	Eibrary Support	Pricing	My Account   Log Ou
Account Licenses Crec	lentials Target Manager			
Target Manager				Add Database
Jse the Target Manager to	create and manage databases and	d targets.		
Search				
Database	Туре	Targets	Date Modified	
	Device	3	Feb 21, 2021	Î
	Device	1	Jun 04, 2023	
	Device	8	Feb 10, 2021	
SSF-GiT-Veri-Tabani	Device	1	Jan 01, 2024	
	Device	5	Feb 16, 2021	
Showing 1-5 of 5	« «	1 >	>>	25 per page ▼
ast updated: Today 5:09 PM R	lefresh			

Here we will see that we have only uploaded an image file named **GSF-Logo**. Let's click **Add Target** to add two logo files.

vuforia <sup>•</sup> engine <sup>-</sup> developer portal	Home	Downloads	Library	Support	Pricing	My Account   Log Out
Account Licenses Credentials	Target Manager					
Target Manager > GSF-GiT-Veri-Taba						
GSF-GiT-Veri-Tabani Type: Device	Edit Name					
Targets (1)						
Add Target						Download Database (All)
Target Name	Туре	e	Rating (i)	Sta	atus 🗸	Date Modified
GSF-LOGO	Ima	ge	****	Ac	tive	Jan 01, 2024 19:38

First, we select the Faculty of Engineering logo file and add it (Add).

(Note: Possible errors are also shown)

Туре:			
	$\bigcirc$	07	
Image	Multi	Cylinder	Object
File:			
Dumlupınar_Üni	versitesi_Mühendis	lik_Fakültesi_Logo.jp	Browse
.jpg or .png (max fi	le 2mb)		
Width:			
1			
Enter the width of y same scale as your	augmented virtual o	units. The size of the ontent. Vuforia uses culated when you up	meters as the defa
Enter the width of y same scale as your	augmented virtual o	ontent. Vuforia uses	meters as the defa
Enter the width of y same scale as your unit scale. The targ	augmented virtual o	ontent. Vuforia uses culated when you up	meters as the defa
Enter the width of y same scale as your unit scale. The targ Name: Dumlupinar_Ünive	augmented virtual c et's height will be cal ersitesi_Mühendislik_ que to a database. W	ontent. Vuforia uses culated when you up	meters as the defa load your image.

Here, the letter **Ü** caused problems because it does not exist in English.



#### Let's rename and add.

Name:	
DPUMuhendislikLogosu	

Name must have no spaces and may only contain: numbers (0-9), letters (a-z), underscores ( \_ ) and dashes ( - )

Name must be unique to a database. When a target is detected in your application, this will be reported in the API

Now the upload will start. If we wait a bit and refresh the page, the quality of our shape will be rated with stars. 4 stars is a quality level that is considered good.

vuforia: engine- developer portal	Home Downloads	Library Supp	oort Pricing	My Account   Log Out
Account Licenses Credentials	Target Manager			
Target Manager > GSF-GiT-Veri-Taba				
GSF-GiT-Veri-Tabani Type: Device Targets (2)	Edit Name			
Add Target				Download Database (All)
Target Name	Туре	Rating 🛈	Status 🗸	Date Modified
DPUMuhendislikLogosu	Image	****	Active	Mar 06, 2024 17:18
GSF-LOGO	Image	*****	Active	Jan 01, 2024 19:38

#### Now, upload the university logo with Add Target.

vuforia <sup>®</sup> engine <sup>®</sup> developer portal	Home Download	ds Library Sup	port Pricing	My Account   Log Out
Account Licenses Credentials	Target Manager			
Target Manager > GSF-GiT-Veri-Taba				
GSF-GiT-Veri-Tabani Type: Device	Edit Name			
Targets (2)				
Add Target				Download Database (All)
Target Name	Туре	Rating ①	Status 🗸	Date Modified
DPUMuhendislikLogosu	Image	****	Active	Mar 06, 2024 17:18
STRAM GSF-LOGO	Image	*****	Active	Jan 01, 2024 19:38

In the window that opens, select the DPU logo file and add it (Add).

	$\square$	07	
Image	Multi	Cylinder	Object
File:			
dpu-logo.png			Browse
.jpg or .png (max file	2mb)		
Width:			
1			
same scale as your a	<u> </u>	nits. The size of the ta ontent. Vuforia uses n ulated when you uplo	neters as the defau
same scale as your a	augmented virtual co	ntent. Vuforia uses n	neters as the defau
same scale as your a unit scale. The targe	augmented virtual co	ntent. Vuforia uses n	neters as the defau
same scale as your a unit scale. The targe Name: dpu-logo	augmented virtual co t's height will be calc ueto a database. Wh	ntent. Vuforia uses n	neters as the defau bad your image.
same scale as your a unit scale. The targe Name: dpu-logo Name must be uniqu	augmented virtual co t's height will be calc ueto a database. Wh	ntent. Vuforia uses n ulated when you upl	neters as the defau bad your image.

Add Target

Here we encountered another error. Vuforia is a program that distinguishes images. So, try again by selecting a new file.

File:		
dpu-logo.png	Browse	
Invalid file format.Only 8 bit gray scale or 24 bit RGB of file type JPG or PNG are allowed.	e	
.jpg or .png (max file 2mb)		

Try again after making changes within the warnings or taking the shape from another source.

Add Target			
Туре:			
$\Box$	$\bigcirc$	07	
Image	Multi	Cylinder	Object
File:			
DPULOGO.jpg			Browse
.jpg or .png (max file	2mb)		
Width:			
1			
	augmented virtual co	ontent. Vuforia uses	target should be on th meters as the default bload your image.
DPULOGO			
Name must be uniq this will be reported		hen a target is detec	ted in your application

	Cancel	Add
--	--------	-----

This image file is also rated 4 stars and is available.

# GSF-GiT-Veri-Tabani Edit Name

Type: Device				
Targets (3)				
Add Target				Download Database (All)
Target Name	Туре	Rating (i)	Status 🗸	Date Modified
Target Name DPULOGO	Type Image	Rating ① ★★★★★	Status 🗸	Date Modified Mar 06, 2024 17:34

Then, select all these files and click on **Download Database (all)**, where there are 3 images.

GSF-GiT-Veri-Tabani Edit Name Type: Device				
Targets (3)				
Add Target				Download Database (3)
Target Name	Туре	Rating (i)	Status 🗸	Date Modified
3 selected Delete				
DPULOGO	Image	****	Active	Mar 06, 2024 17:34
DPUMuhendislikLogosu	Image	****	Active	Mar 06, 2024 17:18
GSF-LOGO	Image	****	Active	Jan 01, 2024 19:38

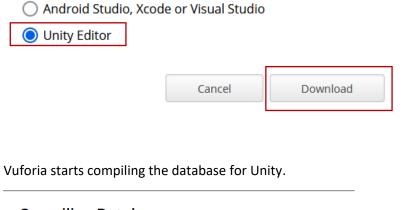
In the window that opens, select the Unity Editor and download it (Download).

# **Download Database**

3 of 3 active targets will be downloaded

Name: GSF-GiT-Veri-Tabani

#### Select a development platform:

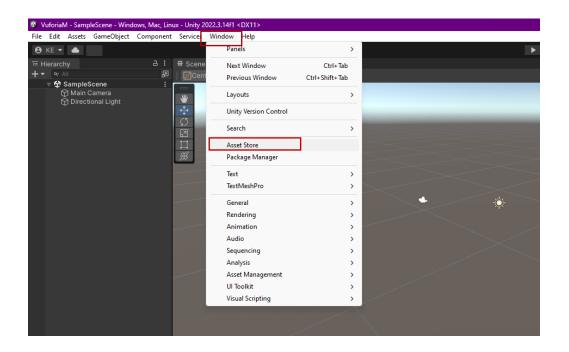




In our web browser, we see that the file named **GSF-GİTVeri\_Tabani.unitypackage** has been downloaded to the hard disk.



Create a project in Unity. Click **Window>Asset Store>Search Online**.

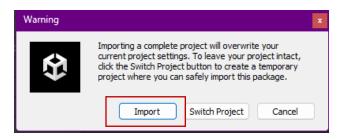


Here, search for Vuforia Core Samples and open it in Unity by selecting it. It will be displayed directly under the **Package Manager**. Since we have downloaded it before, we can continue by **Importing**. If we have not downloaded it before, we can select **Download** or **Update** if there is an update.

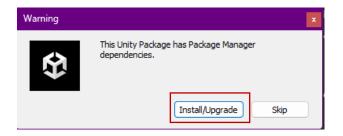
Vuforia recommends a minimum Unity version of 2022.3.16.

Package Manager		: 🗆 ×					
+ ▼ Packages: My Assets ▼ Sort: Name (asc) ▼ Filte	ers 🔻 Clear Fil	Iters : ٩					
2D Game Kit	1.9.5 🛓	Vuforia Core Samples Import Update					
3D Free Modular Kit	1.2 🛓	10.19.3 · November 14, 2023 Asset Store					
3D Game Kit - Environment Pack	1.0 🛓	PTC Vuforia					
3D Game Kit Lite	1.8.2 🛓						
3D Realistic Terrain Free	1.3 🛓	Overview Releases Images					
3D Scifi Kit Starter Kit	1.2 🛓						
3D The Blacksmith's House	1.2 🛓	Supported Unity Versions 2022.3.5 or higher					
3rd Person Controller + Fly Mode	2.1.6 🛓	Package Size Size: 962,01 MB (Number of files: 1421)					
AnyUI - Map Your UI On Any 3D Surface	1.7 🛓	Purchased Date October 24, 2020					
ARCADE: FREE Racing Car	3.0 <del>V</del>	See the Unity Getting Started (https://library.vuforia.com/getting-started/getting-started-vuforia-					
Avatar Maker Free - 3D avatar from a single selfie	1.3.2 🛓	engine-unity) guide to learn more about working with Vuforia in Unity.					
Base Helicopter Controller	1.1 🛓						
Big Furniture Pack	1.3 🛓	Get started with augmented reality using Vuforia Engine (https://engine.vuforia.com/), the world's					
BOKI - Low Poly Nature	1.0 🛓	most widely used AR platform. Vuforia Engine makes it easy to develop cross-platform apps for the					
Bolt   Now is Unity Visual Scripting & is built-in to Unity	′ 1.4 <u>↓</u>	latest handheld and headworn devices and powers over 80,000 apps including titles from EA Enix, LEGO, Activision and more.					
Bézier Path Creator	1.2 🛓						
Camera Path Creator	1.2 🛓	The Vuforia Core Samples provide complete Unity scenes that show how to create AR experiences					
Cartoon Sport Cars	1.0.1 🛓	for everyday objects and environments using Vuforia targets and trackers.					
Character Controller SUPER	22.10.9f 🛓						
CITY package	1.0 🛓	Register (https://developer.vuforia.com/user/register) as a Vuforia Developer when you are ready to					
Classic Skybox	1.0 🛓	add your own targets. Check out the Support Center (https://developer.vuforia.com/support) for resources and support!					
Conifers [BOTD]	2.0 🛓	resources and support:					
Curve Designer	1.0.3 🛓						
Easy FPS	1.0 🛓						
echo3D SDK							
Vuforia Core Samples	10.19.3 🕤						
26 of 130	oad 25 <del>-</del>						
Last undate Mar 7, 12:28							

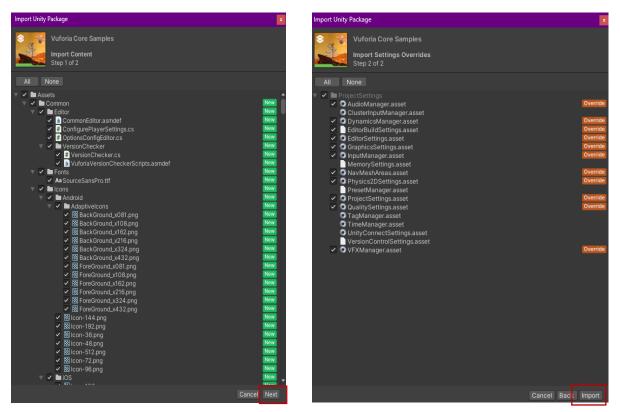
After clicking **Import**, if a warning comes up that the project will be processed, continue by clicking **Import**.



Unity may inform you that there are some updates and ask for confirmation. Press Install/Upgrade.



In the window with options for import operations, click **Next** and then click **Import** in the window that opens, which contains many settings.



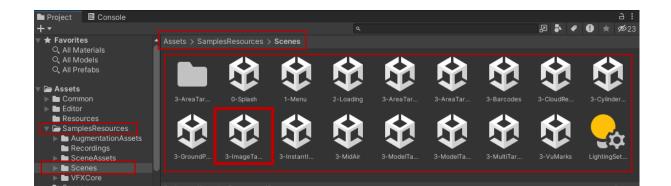
Finally, if Vuforia has an up-to-date package, it will be updated with **Update**. Continue by saying **Update**.

Add Vuforia Eng	gine Package 🛛 🗙
Ŷ	Would you like to update your project to include the Vuforia Engine 10. 19.3 package from the unitypackage? If an older Vuforia Engine package is already present in your project it will be upgraded to version 10. 19.3
	Update Cancel

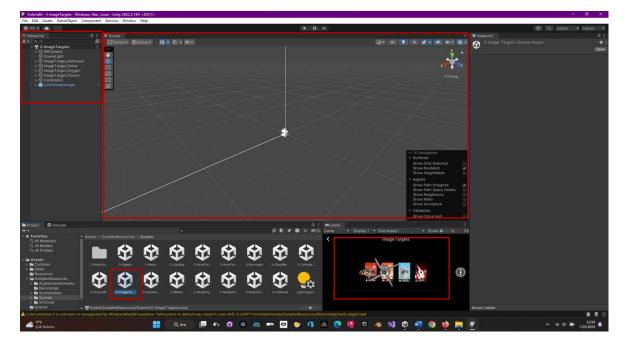
At the end of this process, we can see that new folders have been created under **Project>Assets**.

Project 🛛 🖻 Console						
+-					۹	
🔻 🖈 Favorites	Assets					
Q, All Materials Q, All Models Q, All Prefabs						
🔻 🗁 Assets		_			_	
<ul> <li>Common</li> <li>Editor</li> <li>Resources</li> </ul>	Common	Editor	Resources	SamplesRe	Scenes	Streaming
<ul> <li>SamplesResources</li> <li>Scenes</li> <li>StreamingAssets</li> </ul>						
Packages						

Go to the **Scenes** section under the **SampleResources** folder. Access Vuforia's rich augmented reality scenes. In the scenes, select the **ImageTarget** scene and open this scene with a double click.



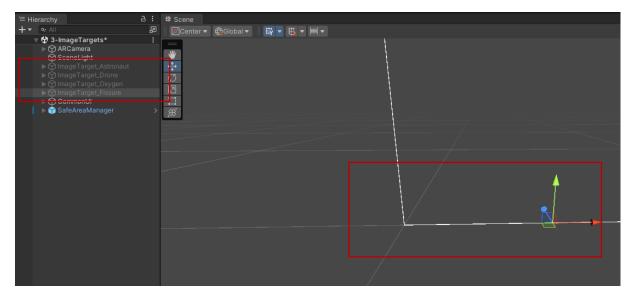
Scene will open.



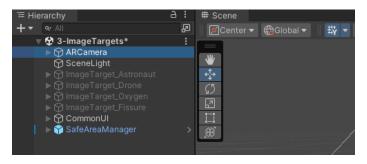
Models can be viewed by double-clicking any **Image Target** in the scene that comes with the **UI (User Interface) Canvas** structure.

🕸 VuforiaM - 3-ImageTargets - Windows, Mac, I							- 0 X
File Edit Assets GameObject Component	: Services Window Help						
Өке • 📥						S Q Layers	<ul> <li>Layout</li> </ul>
					Inspector		
+ • 역 사 [관]	Center • Global • 🐺 • 🐯 •	H.E.		20 📍 4 2 - 00 11 - () -	😭 🖌 ImageTarget_Asti	ronaut	
⊢ I ARCamera				¥ .	Tag Untagged		
Consul light				_ <u>_</u>			071
ImageTarget_Drone	0						
⊢ () ImageTarget_Oxygen ⊢ () ImageTarget_Fissure	2						
E CommonUl							
📔 🕨 😚 SafeAreaManager 🔷 🔿					🔻 🕜 🧭 Image Target Behav		
					Download ne	w Vuforia Engine version: 1	20.3
						From Database	
						VuforiaMars.Jmages	
		📥 🔘				Astronaut Add Target	
			<u>L</u>				
		8 N P /			Add Occlusion C	bject Add Target Rep	resentation
			LL SS		🔻 🛢 🖌 Default Observer Ev		
				Al Navigation     Surfaces			
				Show Only Selected	Consider target as visible if Tracked or Extended Track		
				Show NavMesh 🖌			
				T Agents	Use smooth transition on po Pose Smoothing Curve		
				Show Path Polygons 🛛 👻			
				Show Path Query Nodes Show Neighbours	On Target Found ()		
				Show Walls			
				Show Avoidance			
		/		<ul> <li>Obstacles</li> </ul>			

Since we will be creating our own **database**, let's disable all **Image Target** objects and free up the space of objects in the scene.



The **ArCamera** is generated already.



Choose Open Vuforia Configuration in the Inspector.

VutonaM - 3-ImageTargets - Windows,		>			-	U .
File Edit Assets GameObject Comp	onent Services Window Help					
🔁 KE 🔹 📥					<li>Q Layers •</li>	
					Inspector	
+ ▼ @ All ▼ ✿ 3-ImageTargets*	2 Center GGlobal -			Q ▼ 20 🕴 46 💐 7 💋 III ▼ 🔂 ▼	ARCamera	Static 🔹
Image: Image: ARCamera				- <u>v</u> , ~	Tag Untagged   Layer Default	
SceneLight ► (2) ImageTarget_Astronaut	*					0 2 1
► ImageTarget_Drone	(5				Position X 0.15 Y 0.5	
► ⑦ ImageTarget_Oxygen ► ⑦ ImageTarget_Fissure						
Image: Image: Provide the image of the i	** の 回					
SafeAreaManager	> <u>8</u>				Camera	0 ≓ :
					Clear Flags Solid Color	•
					Background	/
				ARCamera	Culling Mask Everything	
					Projection Perspective	*
					FOV Axis Vertical Field of View	51
					Physical Camera	
				$\mathbf{i}$	Clipping Planes Near 0.01	
		<u></u>		_ <u>_</u> · · ·		
				Al Navigation		
				▼ Surfaces		
				Show Only Selected Show NavMesh		
				Show HeightMesh	Rendering Path Use Graphics Settings	*
				· regenits	Target Texture None (Render Texture) Occlusion Culling	•
					HDR Off	
				Show Neighbours	MSAA Off	
				Show Walls Show Avoidance		
					Target Display Display 1	
				Show Carve Hull	Audio Listener	0 ≓ 1
Project 🗟 Console			A : 😎 Game		🖸 🗹 Vuforia Behaviour (Script)	0 # 1
+•			🔊 🐉 🕫 🔮 🜟 🕫 23 Game 🔹 Display 1	1 • Free Aspect • Scale • 1x Pia	Download new Vuforia Engine version: 10.20.3	
∀ ★ Favorites     Assi     Q     All Materials     Assi			<	Image Targets		
Q, All Models Q, All Prefabs		00000			Open Vuforia Engine configuration	
· ·					Init Error Handler (Script)	0 # 1
v ≅ Assets ⊨ Common 3	B-AreaTar 0-Splash 1-Men	u 2-Loading 3-AreaTar 3-AreaTar 3-Barco	ies 3-CloudRe 3-Cylinder		Script DinitErrorHandler	
Editor					Frame Rate Settings (Script)	0 2 1
Resources					Pranie Rate Settings (Script)	• • •

In the incoming menu, go to the window that says **App License Key**.



Remember the **license key** we copied in Vuforia. We memorized it, but if there is other information written on it, copy it again.

vuforia: e develop	engine <sup></sup> er portal		Home	Downloads	Library	Support	Pricing	Μ	
Account	Licenses	Credentials	Target Mana	ager					
Licenses > 0	iT Edit Nar	ne Delete License age	Key						
ARd+4zD// ODjyASVPV cllGfsxI2	////AAABme+1 /SRUCfLsk0e /Myoqd2prK/(	GxIvv12jpormKHC CCQpMTlvQEzNqGn	wDunMC8gkAYX 49IymMcS5F6s 40SMo3hjocrH	U62uH3w1D745JLpU VFuXPQ3TK96n8zU7: 6on9Mt1cJHd2taA5J Cidd3x1kPoBmg=2M(	LQYwfopnCSjD 182aLo5r+0DF	hQyZD15VrU14 roLXf1E1JipFv	(5UHvjblYEuDGOW	Copied to clipboar	d
-	ive in 01, 2024	18:58 b0d23447c3b64	008cedabe1b	c8					
-	eated - Jan 0	1, 2024 18:58							

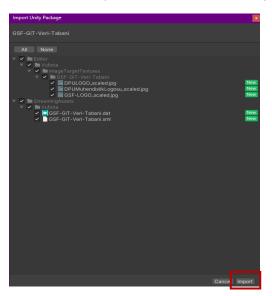
#### Paste the key into the App License Key section in Unity.



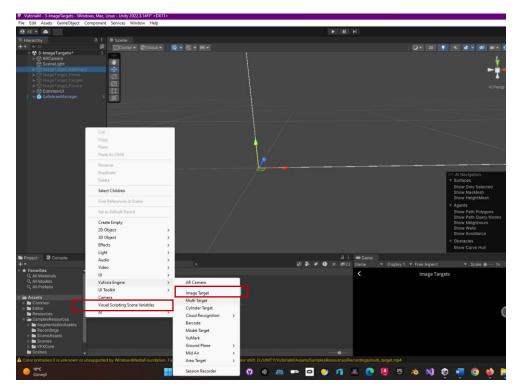
Now, drag the **GSF-GiT-Data-Base.unitypackage** file that we downloaded from **Vuforia** to the Assets section of our project.

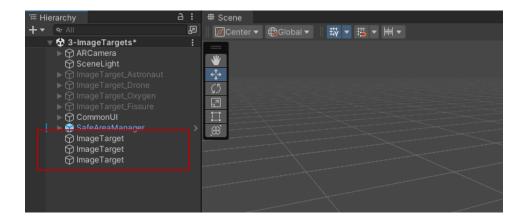
🕸 GSF-GiT-Veri-Tabani.unitypackage	6.03.2024 17:39	Unity package file 180 KB	
------------------------------------	-----------------	---------------------------	--

A window will open where we can see the package content and our JPG files. Click Import.

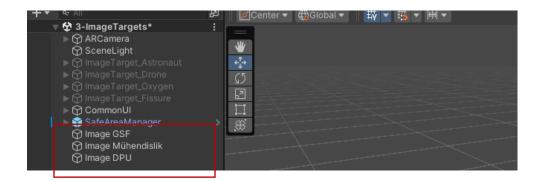


At this stage, we need to create new **Image Target** objects instead of the ones we made passive. Open a window with the right mouse button in the **Hierarchy** section and click the **Vuforia Engine** option. An object named **Image Target** will be added to our object list. Since we will be working with three targets (multiple), repeat this process twice more and increase the number of **Image Targets** to three.

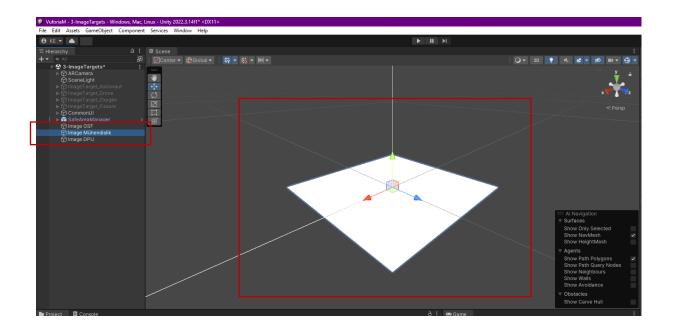




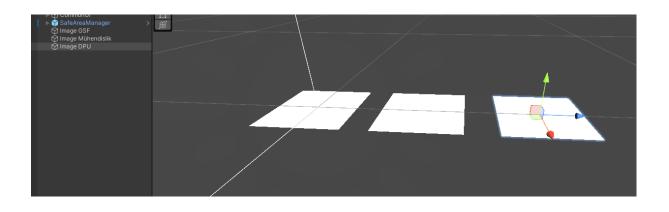
We can name each one to avoid confusion.



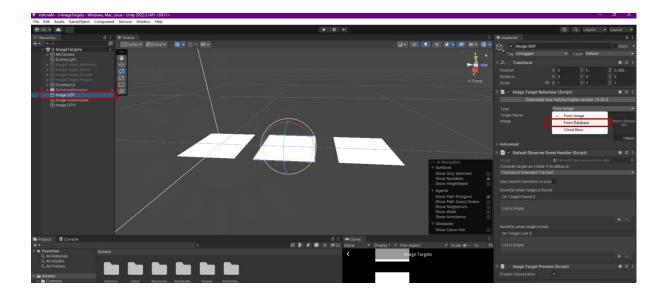
Double-clicking on one of these target objects in the **Hierarchy** will bring the scene into focus. Clicking on the other two will reveal that all three are on top of each other.



So, reposition the plane image objects so that they are next to each other in the scene.



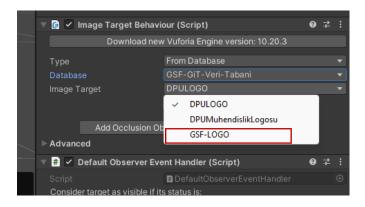
Now, it's time to match these white cards with the images in our database. To do this, first select the **Image GSF** object. In the **Inspector** section, select **From Database** under the **Type** option.



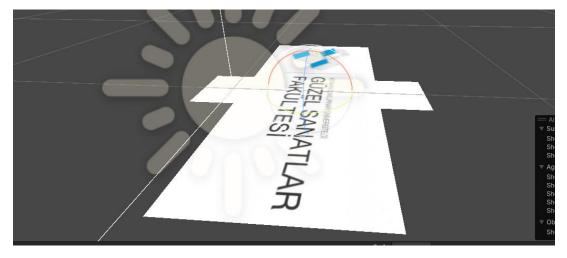
Here, select the database we created in the **Database** section.



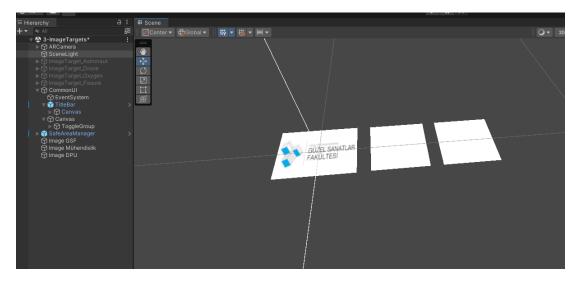
Then, select the **GSF-LOGO** file as the **Image Target** in this database. After the selection, the match is made automatically without any further action.



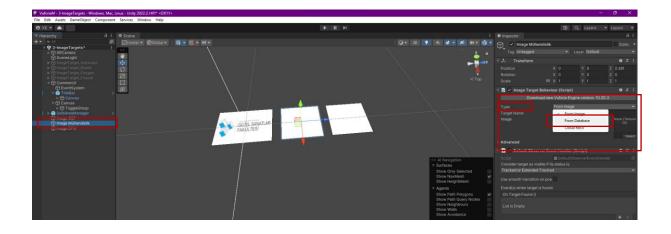
The **GSF-LOGO** shape is matched with the **Image Target** object in the scene.

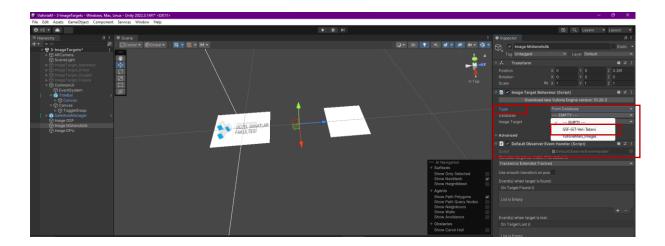


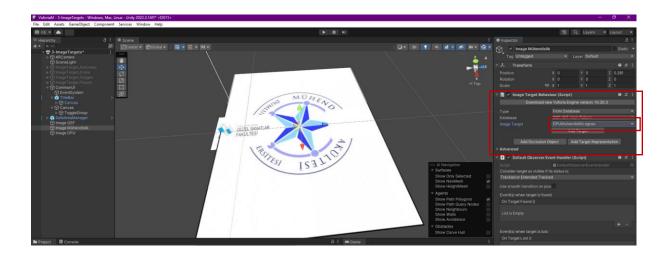
However, for the shape to be compatible with the target, the image must be selected and resized.



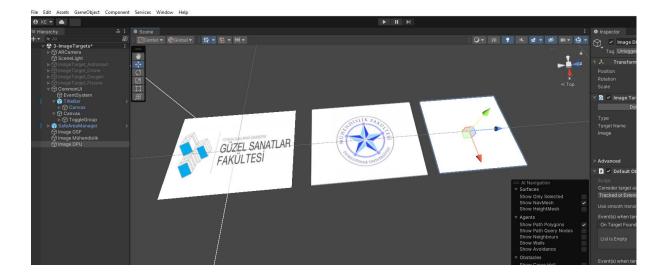
Now, select the Image Target object and repeat the same process for it.



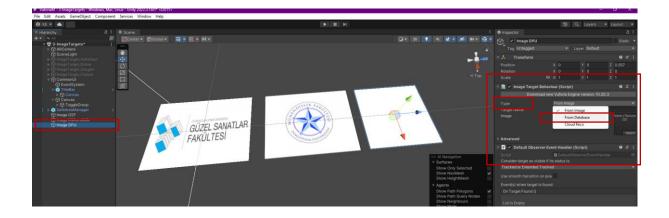




Adjust the place and position of the image.

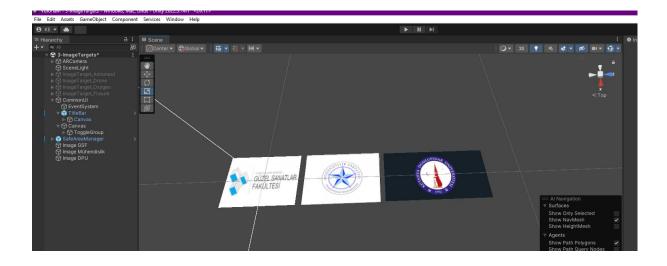


Finally, match the third image to the Image DPU target.



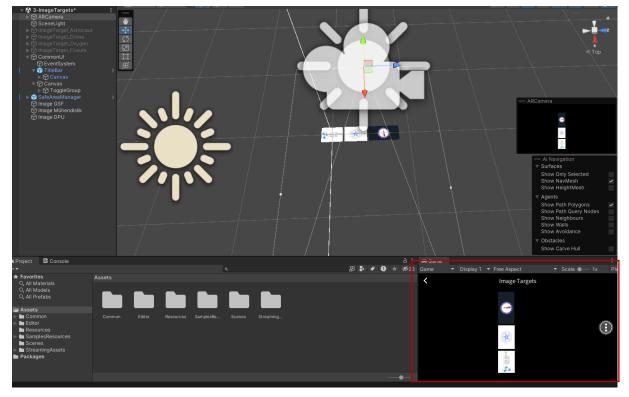
🔻 🕞 🖌 Image Target Behaviour	(Script)	0 ∓ :	🔻 🕼 🖌 Image Target Behavi	iour (Script)	0 ‡ :
Download new Vu	uforia Engine version: 10.20.3		Download nev	v Vuforia Engine version: 10.20	.3
 .,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	om Database - EMPTY		Type Database	From Database GSF-GiT-Veri-Tabani	* *
Image Target	GSF-GiT-Veri-Tabani	-	Image Target	DPULOGO Add Target	<b></b>
▶ Advanced	VuforiaMars_Images		Add Occlusion Ol	bject Add Target Repres	entation
🔻 🗰 🗹 Default Observer Event I	Handler (Script)	0 ‡ :	▶ Advanced		

Adjust the place and position of the image.

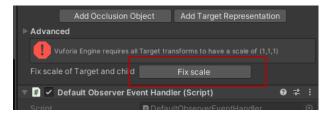


A dark background photo was used here for the DPU logo, but a white background image would be more functional.

Three images can be placed in the frame by changing the **ARCamera** position.



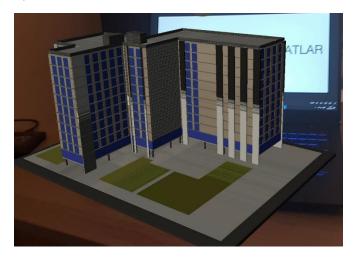
When **Image Target** objects are selected, if a scale warning appears in the **Advanced** section, click **Fix Scale** and then confirm with **Ok**.



At this stage, the kind of operation that will be done on the image/photo displayed with the augmented reality camera should be determined.

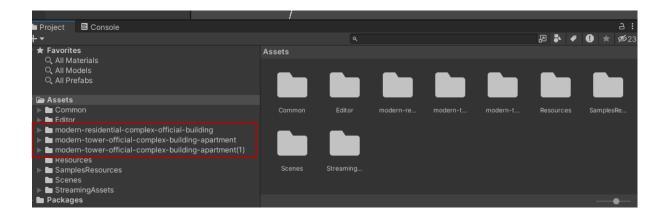
As you may recall, in our single **Image Target** application, we used the Modern Tower Official Complex Building Apartment model from the **Sketchfab** site.

We matched it with GSF-LOGO, and when we brought the phone/tablet to this picture, the building opened in 3D.

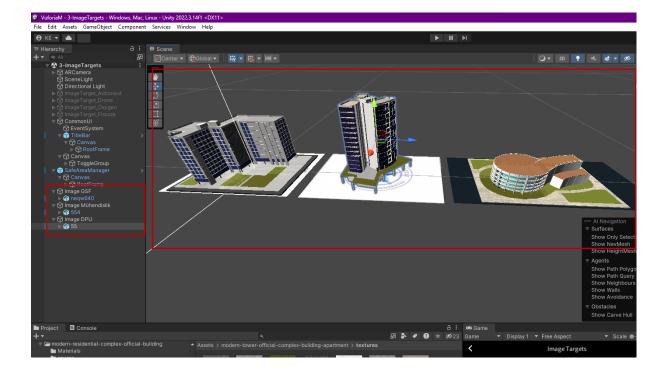


Now, assign the **Image Target** images of the three in a similar way. For this purpose, we can use **Unity Asset Store** or **Sketchfab**.

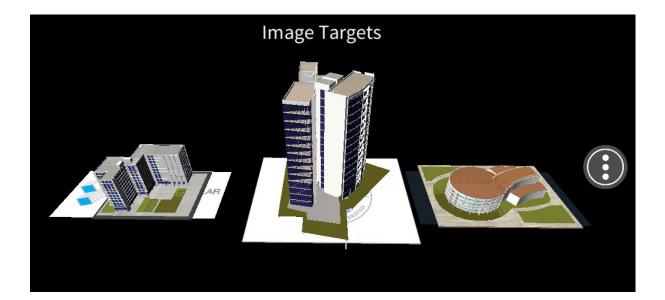
In the multi-target shape application, three building models were downloaded from **Sketchfab**, whose names are shown in the figure.



The buildings were brought into the scene one by one, scaled, rotated and moved to position them on the cards. They were also dragged into and connected to the **Image Targets** in the **Archive**.

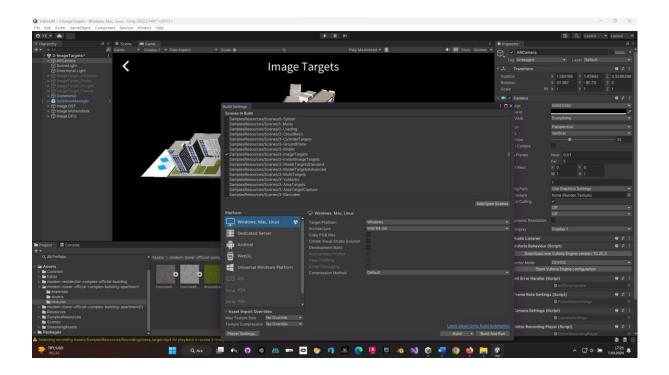


The image on the game window is given below.



# 11.6.Deployment into Mobile Devices

Let's repeat the steps in the **Single Image Target** application here. This will open the **Build Settings** window. There are many scenes listed in the **Scenes in Build** section.



However, since our project is about **Image Target**, the boxes of the ones other than **Image Targets** are checked and excluded from the process.

Also, since this will be a mobile application, the **Switch Platform** process is done by selecting the **Android** platform.

Build Settings			: 🗆 ×				
Scenes In Build							
SamplesResources/Scenes/0-Splash SamplesResources/Scenes/1-Menu SamplesResources/Scenes/2-Loading SamplesResources/Scenes/3-CloudReco SamplesResources/Scenes/3-CrylinderTargets SamplesResources/Scenes/3-GroundPlane SamplesResources/Scenes/3-GroundPlane							
SamplesResources/Scenes/3-ImageTargets 0							
Sampleskesources/Scenes/3-ModelTargetsStandard Sampleskesources/Scenes/3-ModelTargetsStandard SamplesResources/Scenes/3-ModelTargetsAdvanced SamplesResources/Scenes/3-Multi Targets SamplesResources/Scenes/3-VuMarks SamplesResources/Scenes/3-AreaTargets SamplesResources/Scenes/3-AreaTargetCapture SamplesResources/Scenes/3-AreaTargetCapture SamplesResources/Scenes/3-AreaTargetCapture							
Add Open Scenes							
🖵 Windows, Mac, Linux 🛭 🛠	Texture Compression	Use Player Settings	•				
Dedicated Server	ETC2 fallback Export Project	32-bit	•				
Android	Symlink Sources Build App Bundle (Google Play)						
WebGL	Create symbols.zip	Disabled	•				
		Default device	▼ Refresh				
Universal Windows Platform			Patch And Run				
100, 100							
iOS ios	Development Build						
с Рла PS4							
ərs PS5	Compression Method	LZ4					
Compression Method     Compression Method     Compression Method     Compression Method							
Max Texture Size No Override 🔻							
Texture Compression No Override -			Learn about Unity Build Automation				
Player Settings			Switch Platform Build And Run				

Let's connect our phone with developer features to the computer with a cable and **Refresh** the **Run Device** section.

Platfo	rm			🖷 Android				
Ţ	Windows, Mac	;, Linux	Î	Texture Compression		Player Settings		-
	Dedicated Ser	ver		ETC2 fallback Export Project	32-b	it		
ıЩ	Android		٩	Symlink Sources Build App Bundle (Google Play)				
5	WebGL			Create symbols.zip Run Device	Disa Defa	bled ult device		✓ Refresh
	Universal Wind	dows Platform	۱	Build to Device Patching is disabled for Release build	~	Default device All compatible devices		Patch And Run
				Development Build		<enter ip=""></enter>	_	
					Ľ	Samsung SM N975F (RF8NB069L8L)		
			Ŧ	Compression Method	LZ4			
▼ Asse	et Import Overri	des						
Max T	exture Size	No Override						
Textur	e Compression	No Override					Lear	n about Unity Build Automation
Playe	er Settings						В	uild 🛛 👻 Build And Run

Go to the **Player Settings** section and specify the **Company Name** and application logo. In the **Graphics API** section, <u>delete</u> the **Vulkan** and **OpenGLE2** options other than **OpenGLES3** with the "-" key.

🌣 Project Settings						: 🗆 >
Adaptive Performance	Player					0 ≓ :
Audio Editor	Company Name		DPU			
Graphics	Product Name		VuforiaM			
Input Manager	Version		10.19.3			
Memory Settings Package Manager	Default Icon					
Physics Physics 2D Player	Default icon					MAR5
Preset Manager Quality Scene Template	Default Cursor					None (Texture 2D)
Script Execution Order Services						Select
Ads	Cursor Hotspot		X 0		Y 0	
Analytics - Gaming Services Environments	Ţ			÷	Î	4
Tags and Layers	Settings for Android					
<ul> <li>TextMesh Pro Settings</li> </ul>	▶ Icon					
Time Timeline	Resolution and Presentation	ion				
UI Builder Version Control	Splash Image					
Visual Scripting						
XR Plugin Management	Rendering					
	Color Space*		Gamm			
	MSAA Fallback		Down	grade		
	Auto Graphics API					
	Graphics APIs					
	= OpenGLES3					
						+
	Require ES3.1					
	Require ES3.1+AEP					
	Require ES3.2					
ort overrides for local development.	Color Gamut*					
maximum texture size or compression						
an speed up asset imports and platfor						+

In the **Identification>Package Name** section, change the package name to match the definition above; **com.DPU.VuforiaM** 

Adaptive Performance Audio       Player       Imput Manager Load/Store Action Debug Mode         Input Manager Memory Settings       SRGB Write Mode*       Imput Manager         Physics       SRGB Write Mode*       Imput Manager         Physics 2D       Recycle command buffers*       3         Physics 2D       Recycle command buffers*       -         Piayer       Perset Manager Apply display rotation during rendering       -         Version*       Override Default Package Name Package Name       -         Services Environments       Diaga       -         Ads Analytics - Gaming Services Environments       Bundle Version Code       10193         Tage and Layers       Target API Level       Android 8.0 'Oreo' (API level 26)       -         Time Time Timeine       Scripting Backend       L2CPP       -         Version Control Visual Scripting       Api Compatibility Level*       IL2CPP       -         Version Control Visual Scripting       Configuration       -       -
Audio       Load/Store Action Debug Mode
Audio       Load/Store Action Debug Mode
Input Manager Memory SettingsVulkari SectingsPackage Manager Physics Physics 2DSRGB Write Mode*Player PlayerAcquire swapchain buffers*Player Ouality Scene Template Script Execution Order Services Environments Tags and Layers TesttingsAcquire swapchain during renderingDistrict SettingsOverride Default Package Name Version ControlTime T
Version*       10.19.3         Ads       Bundle Version Code       10193         Environments       Minimum API Level       Android 8.0 'Oreo' (API level 26)         Tags and Layers       Target API Level       Automatic (highest installed)         Settings       Configuration         Time       Scripting Backend       IL2CPP         Ul Builder       Api Compatibility Level*       INET Framework         Version Control       IL2CPP Code Generation       Faster runtime
Analytics - Gaming Services     Bundle Version Code     10193       Finitronments     Minimum API Level     Android 8.0 'Oreo' (API level 26)       Tags and Layers     Target API Level     Automatic (highest installed)       TextMesh Pro     Scriptings     Configuration       Time     Scripting Backend     IL2CPP       Timeline     Api Compatibility Level*     INET Framework       Version Control     IL2CPP Code Generation
Environments     Minimum API Level     Android 8.0 'Oreo' (API level 26)       Tags and Layers     Target API Level     Automatic (highest installed)       TextMesh Pro     Settings     Configuration       Time     Scripting Backend     IL2CPP       Timeline     Api Compatibility Level*     INET Framework       Version Control     IL2CPP Code Generation
Tags and Layers     Minimum APTEven     Android also Give (APTeven 20)       Tags and Layers     Target API Level     Automatic (highest installed)       TextMesh Pro     Configuration       Settings     Configuration       Time     Scripting Backend       Ul Builder     Api Compatibility Level*       Version Control     IL2CPP Code Generation
Version Control     Configuration       Time     Scripting Backend       IL2CPP       Version Control       IL2CPP Code Generation
Time     Scripting Backend     IL2CPP       Timeline     Api Compatibility Level*     .NET Framework       UI Builder     Api Compatibility Level*     .NET Framework       Version Control     IL2CPP Code Generation     Faster runtime
Timeline     Api Compatibility Level*     .NET Framework       UI Builder     IL2CPP Code Generation     Faster runtime
UI Builder         Api Compatibility Level*         .NET Framework         T           Version Control         IL2CPP Code Generation         Faster runtime         T
Visual Scripting C++ Compiler Configuration Release
XR Plugin Management Use incremental GC
Allow downloads over HTTP* Not allowed
Mute Other Audio Sources*
Target Architectures
ARMv7
ARM64
x86 (Chrome OS)
x86-64 (Chrome OS and Magic Leap 2)
Enable Army9 Security Features for Arm64
Split APKs by target architecture
Target Devices All Devices
Install Location Prefer External
Internet Access Auto
Write Permission

When we select the **Build and Run** section, in the window that opens, we create a new folder named **VuforiaM** with the right click of our mouse to register the application, and we specify the name of our **APK** file as **VuforiaM.apk**.

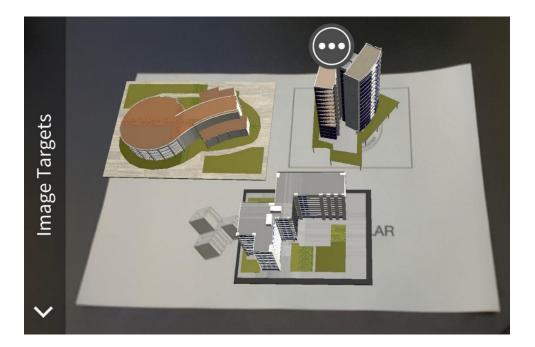
© Build Android								
< → < ↑ <	> Bu bilgisayar > Yeni Birim (D:) > UNITY >	VuforiaM		C				
Düzenle 🔻 Yeni klasör					E	- ?		
🚞 plataform-for-ja	Ad	Değiştirme tarihi	Tür	Boyut				
🚞 A EĞİTİM MATER	Assets	7.03.2024 17:26	Dosya klasörü					
source	💳 Library	7.03.2024 17:30	Dosya klasörü					
E KE	🔁 Logs	7.03.2024 17:23	Dosya klasörü					
	Packages	7.03.2024 12:47	Dosya klasörü					
	ProjectSettings	7.03.2024 13:29	Dosya klasörü					
V 💻 Bu bilgisayar	CAR QCAR	7.03.2024 16:34	Dosya klasörü					
> 🐸 Windows (C:)	Temp	7.03.2024 17:26	Dosya klasörü					
> 🚄 Yeni Birim (D:)	UserSettings	7.03.2024 12:55	Dosya klasörü					
> 🂱 Ağ	🔜 VuforiaM	7.03.2024 17:33	Dosya klasörü					
Dosya adı: 🛛 VuforiaM 🗸 🗸 🗸 V								
Kayıt türü: apk (*.a	Kayıt türü:  apk (*.apk)							
<ul> <li>Klasörleri Gizle</li> </ul>				C	Kaydet	İptal		

Build Android					×
	oilgisayar 🔸 Yeni Birim (D:) 🔸 UNITY 🔸	VuforiaM > VuforiaM		C VuforiaM klasöründ	le ara 👂
Düzenle 🔻 Yeni klasör					∃ - ?
🚞 plataform-for-ja 🛛 🗛 Ad		Değiştirme tarihi	Tür	Boyut	
🚞 A EĞİTİM MATEF		Aramanızla eşleşe	n öğe yok.		
📩 source					
🚞 КЕ					
👻 💻 Bu bilgisayar					
> 🐸 Windows (C:)					
> 👍 Yeni Birim (D:)					
> 💱 Ağ					
Dosya adı: VuforiaM					~
Kayıt tür <mark>ü:</mark> apk (*.apk)					~
<ul> <li>Klasörleri Gizle</li> </ul>				Kaydet	Íptal

The deployment process starts with the **save** button. It will take a few minutes to complete. Finally, it returns to our stage, stating that it was successful.

<ul> <li>GSF-AK - 3-imageTargets - Android - Unity 2</li> </ul>							- 0 ^
File Edit Assets GameObject Component	services window Help		1120	In the second second second second second second second second second second second second second second second			make many processing processing
Өке • 📥			181	11 H			<ol> <li>Q. Løyers • Løyout •</li> </ol>
	à i @ Scene ∞ Game					i O Inspector	
The Heatershy  Constraints of the second sec		10 • 10 • 144 • Euclearen bald Soner in bald Sampelikeauresti Genesi - Spish Sampelikeauresti Genesi - Mariu Sampelikeauresti Genesi - Jouditos Sampelikeauresti Genesi - Jouditos Sampelikeauresti Genesi - Jouditos Sampelikeauresti Genesi - Jouditos Sampelikeauresti Genesi - Makit Sampelikeauresti Genesi - Makit Sampelikeauresti Genesi - Makit	rgets re jeTargets etSStandard atsAdvanced		0* 20 • < • •		4 T
		Sampiselitesauresid cened?-VAMAT ing Sampiselitesauresid cened?-VAMAT ing Sampiselitesauresid cened?-VAMAT ing Sampiselitesauresid cened? Platform Dedicated Sarver Ing Android WebDL Universid Windows Platform	Android Android Testuse Compression EC2 fallback Export Indect Synthis Sources build App Bundle (Boogle Play Bund Davice Build to Davice	Disabled Default device Patch	I Add Dpart Scenes,		
Project Console Unity Version C			Patching is deabled for Release bu				à i
y Sasets > M Common > M Editor y Samodern-tower-official	dem-tower-official-complex-building- negatid2. megatid2.				Learn about Livity Build Automation Build R Build And Run		D <b>b e 0</b> * \$\$\$5
Build completed with a result of 'Succeeded	f in 348 seconds (348152 ms)						0 🛢 🕷 🔾
Buluthu		🚦 Q. Ara 🤷 💻 🕫	o o o 💿 🗠	📼 🖪 \land 📢 🍳	🧧 🧿 🍁 🗮 💇 🗒		∧ ♥ 0) 🆢 22-44

When the application was opened on the mobile device, it was tested on an A4 paper with three logos. As a result, the buildings matched with the three images were displayed as an augmented reality application.



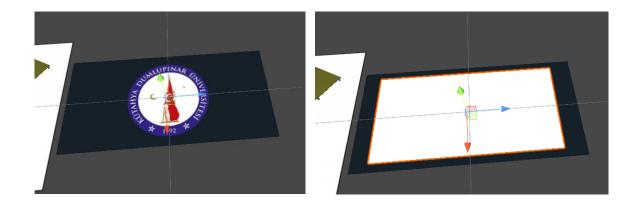
Wherever the app sees three logos, whether individually or in groups, it will display the building model matched on top of the logos in augmented reality format.

Logos will be displayed individually or in groups on wall posters, computer screens, business cards or documents.

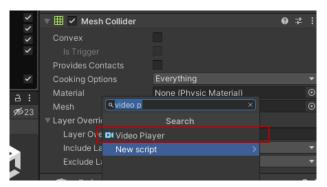
The size of the matched models and their angles relative to the camera can be adjusted by the developer.

# 11.7. Augmented Reality with Video Player

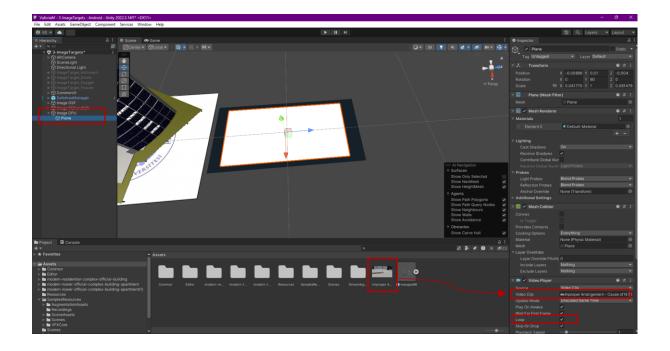
In this application, let's place a **Plane** instead of a 3D model on one of the cards. Here, the **DPU logo** is used. Adjust the size and position of the **Plane** so that it is on top of the logo.



Now, add a Video Player with Add Component in the Inspector section of the Plane.



Drag and connect a video file that we prepared before to the **Video Player>Video Clip** section that we added to the **Plane** and activate the **Loop** box. Connect the **Plane** under **Image DPU**.



When we start the application on our phone, it will see the DPU logo, and the matching video will play on it. If desired, the sound of the video can be turned off by clicking the **Mute** check box.

Audio Output Mode	Direct	▼
Track 0 [2 ch]	✓	
Mute		
Volume		• 1

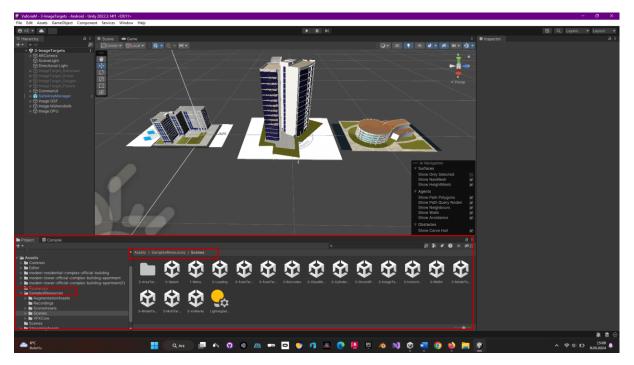
After getting information about single and **multi Image Target** applications, we can continue with **Ground Plane**, which is the application type where the model is displayed by scanning on the ground/surface without the need for any image...

# 11.8.Vuforia AR – Ground Plane

One type of augmented reality application is to open models directly on the ground plane without a target. In this study called **Ground Plane**, we will continue from the point reached in single and multiple target (**Image Target**) applications.

As you may recall, we downloaded the **Vuforia Core Samples** asset from the **Asset Store** and applied single and multiple targets (**Image Target**) to our project. Let's continue with the same project.

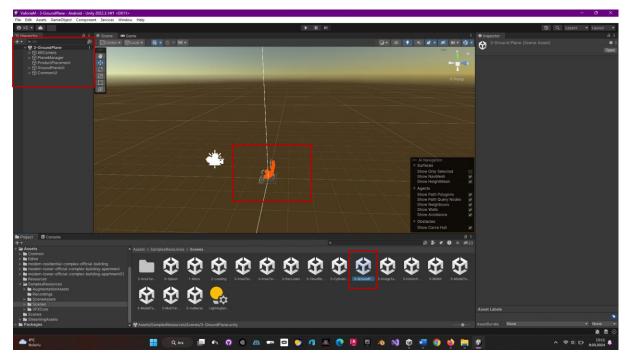
There are many AR application scenes available under **SamplesResources>Scenes** in the **Assets** section.

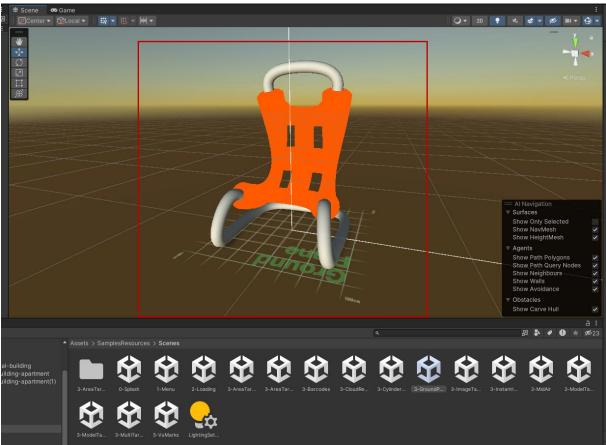


Now, open the Ground Plane scene from these AR models by double-clicking on it.

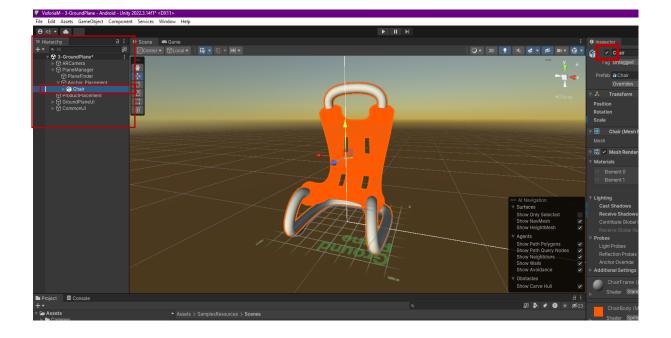


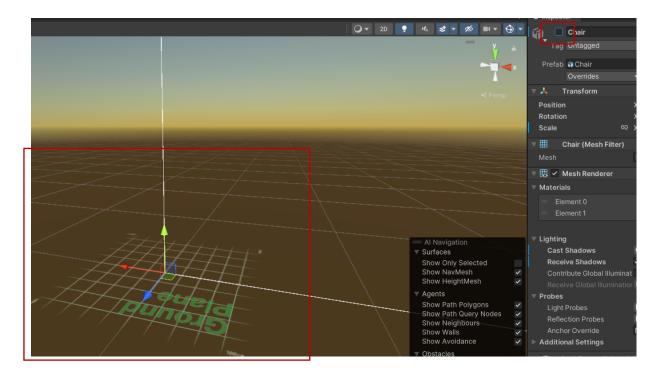
The **Ground Plane** scene is a template with a rocking chair model inside a UI (user interface) framework.



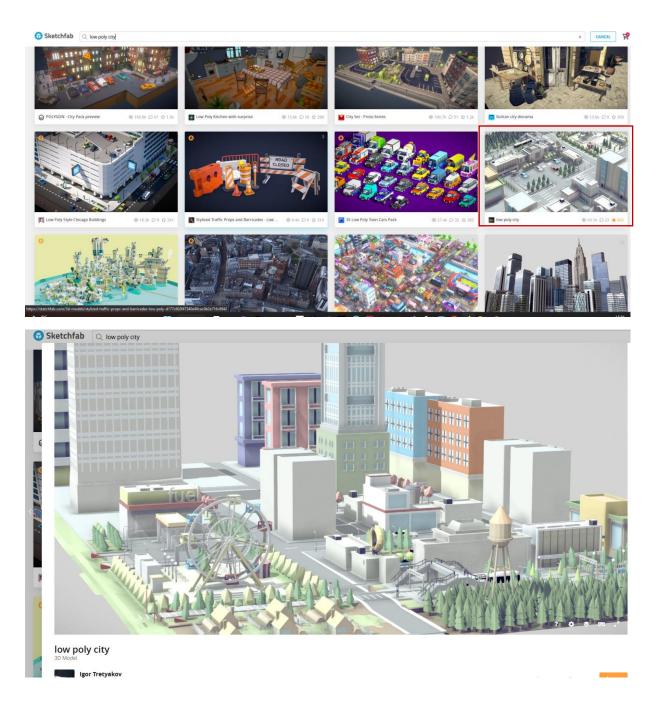


Since we want to open our model on the ground, make this object named **Chair** passive. For this purpose, access the chair with **Hierarchy>PlaneManager>Anchor\_Placement>Chair** and **uncheck** its box to make it invisible.





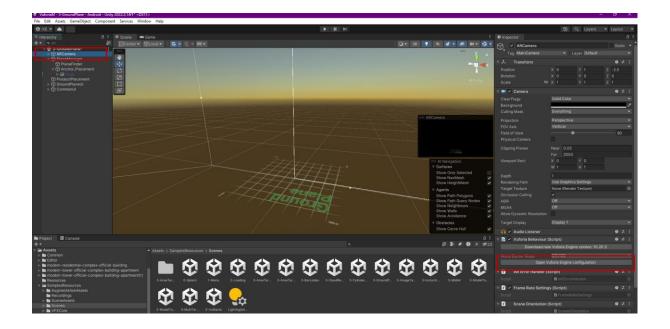
Use the low poly city model found on **Sketchfab**, which we use from time to time in Unity tutorials.



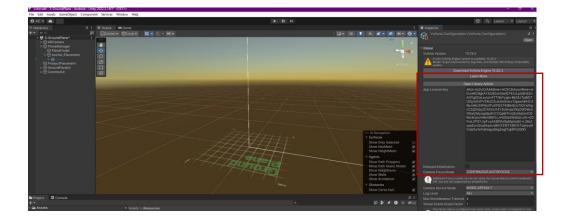
During the preparation phase of the scene, we will need the **license key** we created earlier on the **Vuforia** site. For this purpose, **GSF-Git > License Key**, located in the **Vuforia Licenses** section, will be used.

vuforia <sup>,</sup> engine <sup>-</sup> developer portal	Home	Downloads	Library	Support	Pricing	My Account   Log Out
Account Licenses Credentials	Target Mana	ger				
Licenses > GSF-GIT GSF-GIT Edit Name Delete Licens	e Key					
License Key Usage Please copy the license key below int ARd+4zD////AAABme+HCfiClEAvozWee	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	6ZuH3w1D745JLpU€	5H53nA0Vq0Za	LauiJn4TTNbY	vab+6k05JTp8rD7	
ARd+4zD/////ARABme+HCficlERyozWwe+wPunMC5gKAYXU62uH3wID745JLpU6H53nA0Vg02aLauIJAfTTNEYygb-6k05JTp8rD7       ODjyASVFVSRUCflsk0eGxIm(12jpormKHC49IymMc5SF6sVFuXPQ3TK96n8zU71QYwfopnC5jDhQyZD15VTU14Y5UHvjb1YEuDG0W         olloffsxIZMyoqd2prK/CCQmsUvpZaVGn60S03bjocrH609Mt0Jd4ZaA5E82aLo5+HODFroLXf1EIJjpFvyAS9Dth/GpKIpm       Sopied to dipboard         sjE/+L26x2upeEuni2sgAkpsIJdMnTEWT49G1hTtq4yybICidd3x1kPoBmgs2MgZegfTqMPh2QfXt       Plan Type: Basic						
Status: Active Created: Jan 01, 2024 18:58 License UUID: 6bf6d4b0d23447c3b6	4008cedabe1bo	8				
History: License Created - Jan 01, 2024 18:58						

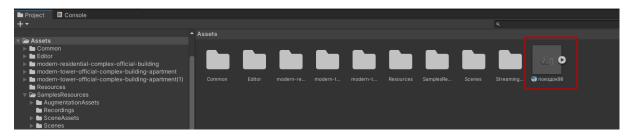
After copying the **license key**, click **Inspector>Open Vuforia Engine** configuration in **ARCamera**.



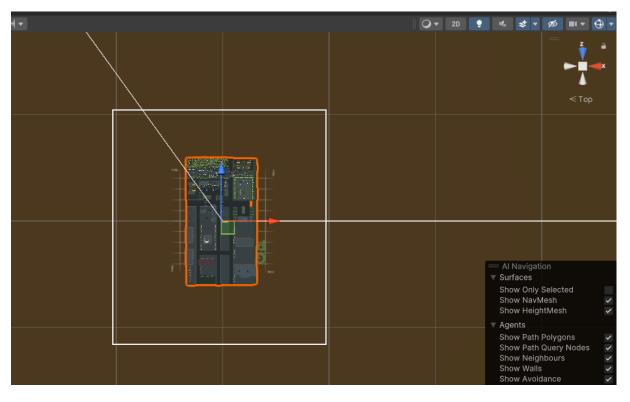
Since we are on the same project, we see that this **key** is already in the **App License Key** section. If we are in a new project, we will need to paste it here.



Drag the Sketchfab, low poly city, **fbx** file into **Assets**.



Then, position it in the **Ground Plane** area on the stage by minimizing it and using the gizmo to control it from the top, right and left.



Drag and attach the model file under **PlaneManager>Anchor\_Placement**. Check that the **Chair** object is inactive.

VuforiaM - 3-GroundPlane - Android - Uni	nity 2022.3.14f1* <dx11></dx11>		
File Edit Assets GameObject Compon	nent Services Window Help		
⊖ KE - ▲			
≔ Hierarchy 금 🗄			
+ - • All	🛛 🛛 📝 Center 🔻 🎲 Local 🔻 📅 🔻 🖽 💌	0-	2D
♥ 3-GroundPlane*     Port Common     PlaneManager     Port P			1000

After the scene design, open the **File>Build Settings** window. Here, the **Android** (IOS for iPhone) platform should be switched to, and only **Ground Plane** should be selected from the scenes.

Build Settings			: 🗆 ×
Scenes In Build			
SamplesResources/Scenes/0-Splash SamplesResources/Scenes/1-Menu SamplesResources/Scenes/2-Loading SamplesResources/Scenes/3-CloudReco SamplesResources/Scenes/3-CvilinderTard SamplesResources/Scenes/3-CvilinderTard			٥
SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-ImageTarget SamplesResources/Scenes/3-ModelTarget SamplesResources/Scenes/3-ModelTarget SamplesResources/Scenes/3-ModelTarget SamplesResources/Scenes/3-VuMarks SamplesResources/Scenes/3-AreaTargetC SamplesResources/Scenes/3-AreaTargetC SamplesResources/Scenes/3-Barcodes	s Targets SStandard sAdvanced		
			Add Open Scenes
Platform	🖷 Android		
Windows, Mac, Linux	Texture Compression ETC2 fallback Export Project Symlink Sources	Use Player Settings 32-bit	•
💼 Android 😪	Build App Bundle (Google Play)		
WebGL	Create symbols.zip Run Device	Disabled Default device	▼ ▼ Refresh
Universal Windows Platform			
iOS ios Pra PS4	Patching is disabled for Release build Development Build Autoconnect Profiler Deep Profiling Script Debugging		
rrs PS5 ▼	Compression Method	LZ4	
Asset Import Overrides     Max Texture Size No Override     Texture Compression No Override     Player Settings			Learn about Unity Build Automation Build <b>v</b> Build And Run

Connect the **mobile** device and check.

Build Settings			: □ >			
Scenes In Build						
Scenes in Bulia SamplesResources/Scenes/1-Menu SamplesResources/Scenes/2-Loading SamplesResources/Scenes/3-CloudReco SamplesResources/Scenes/3-CloudReco SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-VuMarks SamplesResources/Scenes/3-AreaTargets SamplesResources/Scenes/3-AreaTargets						
SamplesResources/Scenes/3-Barcodes						
1			Add Open Scenes			
			Add open occines			
Platform	🖷 Android					
🖵 Windows, Mac, Linux 🏾 🗍	Texture Compression	Use Player Settings				
Dedicated Server	ETC2 fallback	32-bit				
Dedicated Server	Export Project					
🖬 Android 🕸	Symlink Sources Build App Bundle (Google Play)					
WebGL	Create symbols.zip	Disabled				
	Run Device	Default device	✓ Refresh			
Universal Windows Platform		✓ Default device	Patch And Run			
iOS ios	Patching is disabled for Release build	All compatible devices				
103 103	Development Build Autoconnect Profiler	<enter ip=""></enter>	_			
PS4 PS4		Samsung SM N975F (RF8NB069L8L)				
Prs PS5	Compression Method					
▼ Asset Import Overrides						
Max Texture Size No Override 🔻						
Texture Compression No Override -			Learn about Unity Build Automation			
Player Settings			Build 🔻 Build And Run			

Make a few changes to the **Player Settings** to prevent applications from being overwritten.

We named this application **VuforiaGP** as the **Production Name**. We ensured that the same name was used in the **Package Name** section. The abbreviation **GP** is considered a **Ground Plane**. Of course, every designer can create a naming system.

Project Settings			: 🗆
Adaptive Performance Audio	Player		€ ‡
Editor	Company Name	DPU	
Graphics		VuforiaGP	
Input Manager Memory Settings		10.19.3	
Package Manager Physics Physics 2D			MARS
Player Preset Manager			" Select
Quality Scene Template	Default Cursor		None (Texture 2D)
			Select
Services		X O	Y O

🍄 Project Settings			: 0
Adaptive Performance	Player		0 ;
Audio Editor			
Graphics			-
Input Manager			<b>*</b>
Memory Settings		Low Quality	<b>•</b>
Package Manager Physics		Low Quality	•
Physics 2D		<b>×</b>	
Player		0	
Preset Manager			
Quality		<b>~</b>	
Scene Template Script Execution Order	On OpenGL, Profiler GPU Recorders may		
▼ Services			
Ads			
Analytics - Gaming Service Environments			
Tags and Lavers			
TextMesh Pro     ■	360 Stereo Capture*		
Settings Time	Load/Store Action Debug Mode		
Time	Vulkan Settings		
UI Builder	SRGB Write Mode*		
Version Control	Number of swapchain buffers*	3	
Visual Scripting XR Plugin Management	Acquire swapchain image late as possible	•	
Art hagin hanagement		<b>~</b>	
	Apply display rotation during rendering		
	Identification		
	Override Default Package Name	~	1
	Package Name	com.DPU.VuforiaGP	
		10.19.3	<b> </b>
	Bundle Version Code	10193	
	Minimum API Level	Android 8.0 'Oreo' (API level 26)	•
	Target API Level	Automatic (highest installed)	•
	Configuration		

When we click on the **Build and Run** section, we will be asked in which folder the application will be created and its name. Enter our preference for this project; **VuforiaGP.apk** 

The application will be created on both the disk and the phone by clicking the **save** button.

Build Android						×
$\leftrightarrow \rightarrow \sim \uparrow$	🗅 > Bu bilgisayar > Yer	ni Birim (D:) > UNITY >	VuforiaM > VuforiaM		C VuforiaM klas	öründe ara 🔎
Düzenle 🔻 Yeni klasö	ōr					∎ - ?
🚞 A EĞİTİM MATEF	Ad		Değiştirme tarihi	Tür	Boyut	
source	📄 VuforiaM.apk		7.03.2024 17:43	APK Dosyası	110.031 KB	
🚞 KE						
🛑 Ders Notları - Ka						
✓ 💻 Bu bilgisayar │						
> 🐸 Windows (C:)						
> 🖽 Yeni Birim (D:)						
> 🦕 Ağ						
Dosya adı: Vufo	oriaGPlapk					~
Kayıt türü: apk (						~
<ul> <li>Klasörleri Gizle</li> </ul>					Kaydet	İptəl

After opening the application on the **mobile** device, the ground/surface scan is performed. When the **honeycomb** texture is seen, the model can be loaded onto the surface/ground.



By touching the honeycomb, the model will appear here.

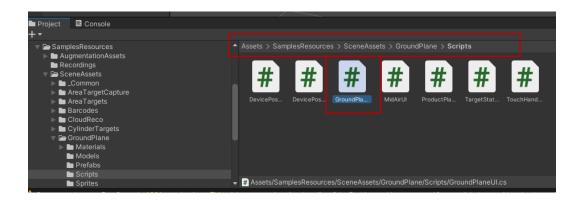


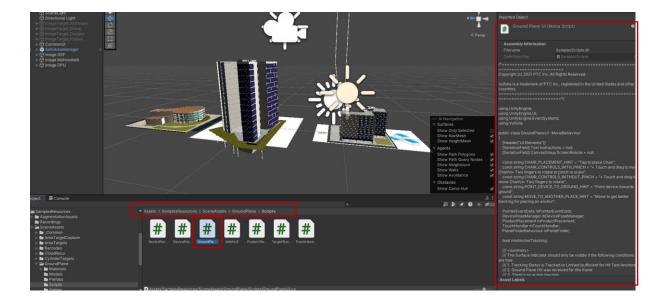
Turning the mobile device allows the honeycomb to be viewed in different locations and sizes, and the model reappears at that point.

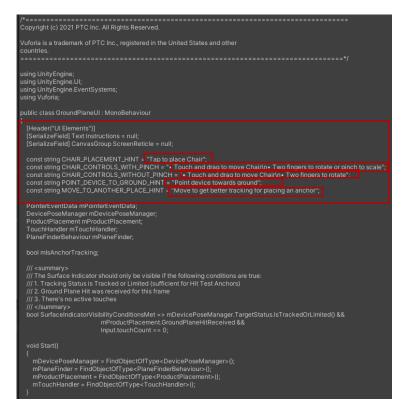
The same model is displayed in front of the computer below.



Another issue is the **UI (interface)** texts seen on the screen. **Texts** such as **Product Placement, Tap to place Chair, Touch and Drag to move Chair, Two fingers to rotate** are templates and are for the **disabled** chair and can be **changed**. For a change, the relevant lines can be found in the **script (C# code**) files and changed with the desired expressions.







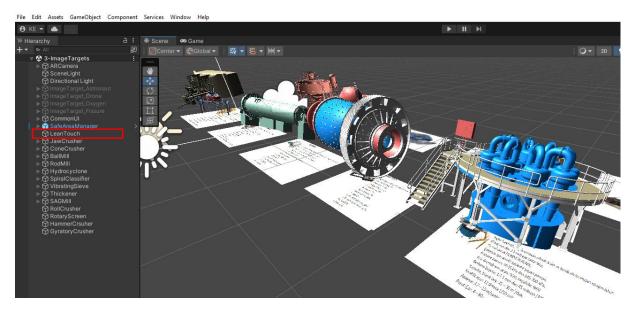
Even the **honeycomb** - **reticle\_ground\_surface** (sight) and other similar figures seen on the screen under **Sprites** can be changed.



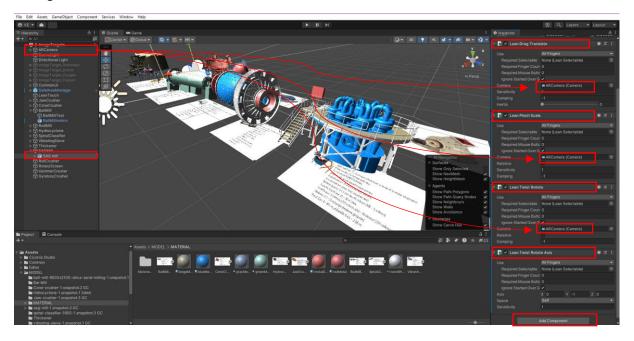
Although the commands include **dragging** and **rotating** the model on the mobile device screen, if these features do not work, the asset named **Lean Touch** can be downloaded and used from the Unity Asset Store. Information about this is included in the asset introduction.



In the example, **mineral processing** machines are used for **Image Target** application of **Vuforia**. After **Import**ing process of **Lean Touch** asset, right-click on **Hierarchy** window and add **Lean Touch** object.



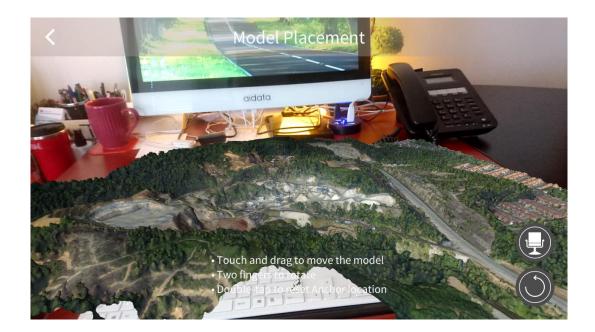
Now, we can **add** the related **Lean Touch** components to the objects located on the scene. For each asset, **Lean Drag Translate, Lean Pinch Scale, Lean Twist Rotate** and **Lean Twist Rotate Axis** components are added (e.g., **SAG mill**). Drag the **ARCamera** to the components in the **Inspector** having **Camera** use.



As a result, when the **APK** file is run, **SAG mill** object can be controlled by **fingers**. It can be **dragged**, **scaled** and **rotated**.

# 11.9. Similar Examples for Mining

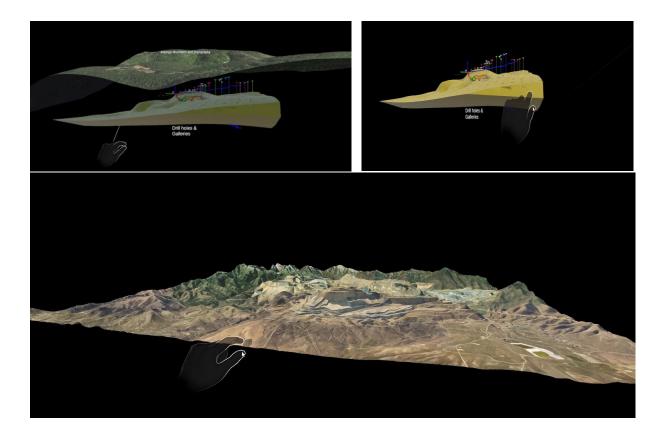
The applications developed for an open pit mine and a quarry are given below. One is in front of the computer, and the other is on the floor of the university corridor.







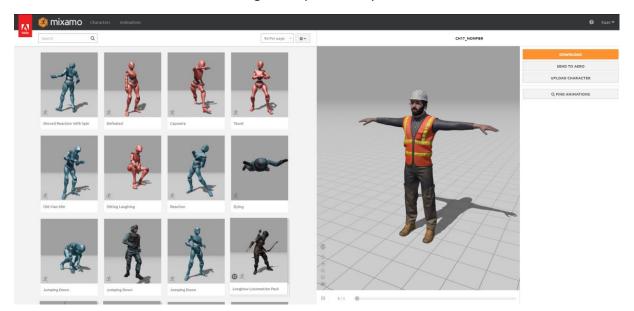
Although, Hololens 2 applications are not included in the book, some of deployments are shown to give an idea.



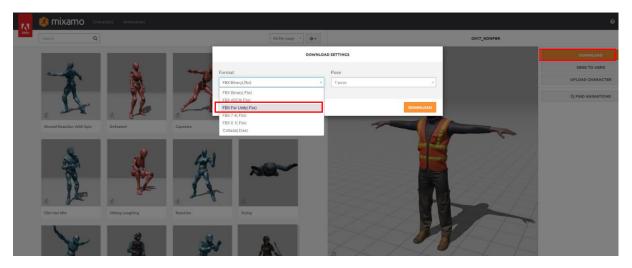
# 12. USE OF ANIMATION AND ADOBE MIXAMO CHARACTERS IN UNITY

Adobe Mixamo (www.mixamo.com) has a large library of characters and animations. The characters and animations found here can be combined and turned into interactive animations in the animator editor in Unity.

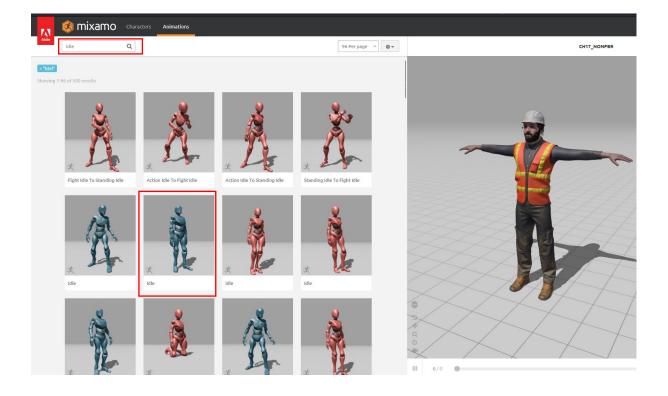
In this application, we can download a **worker armature** (*body*) and some animations from Mixamo and how we can control it with **C# coding** developed in Unity.



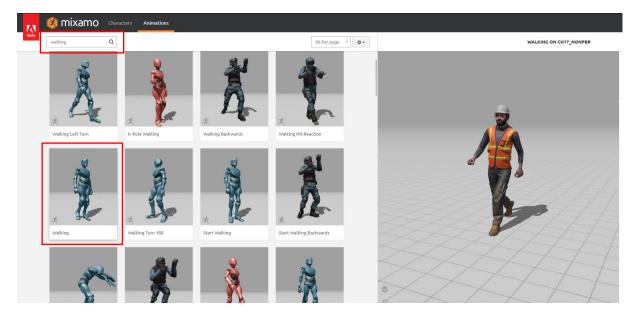
Let's download the root stance of the selected character, **T-Pose**. Here, from the window that appears with the **Download** button, download it by selecting the **FBX For Unity** (.Fbx) line.

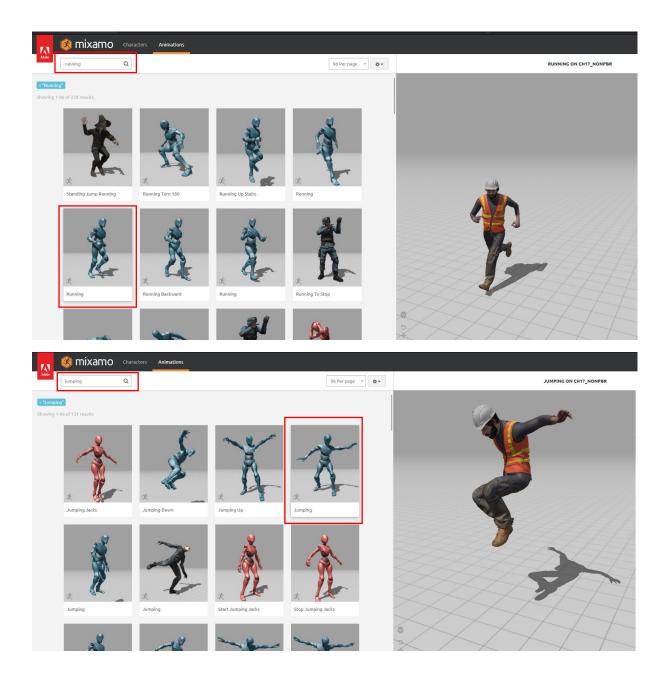


Now, search for the word **Idle** under the **Animations** tab and choose one of the many options that appear that we find appropriate. With this choice, the animation is combined with our character in the **T-Pose** state and plays on it. Download this by clicking the **Download** button and selecting Unity.



Similarly, search for **Walking, Running and Jumping** animations, observe them on the worker and **download** them in the same way.





Now, collect the downloaded files in a folder to place in the **Assets** section of the Unity project we will create. [*Extra kneeling and CPR are also included here*]

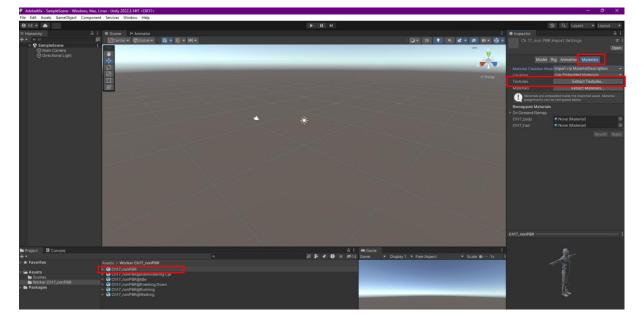
Ad	Değiştirme tarihi	Tür	Boyut
Ch17_nonPBR.fbx	1.06.2024 13:26	3D Object	63.720 KB
Ch17_nonPBR@Administering Cpr.fbx	1.06.2024 13:35	3D Object	64.417 KB
Ch17_nonPBR@ldle.fbx	1.06.2024 13:32	3D Object	64.220 KB
Ch17_nonPBR@Jumping.fbx	1.06.2024 21:47	3D Object	63.896 KB
Ch17_nonPBR@Kneeling Down.fbx	1.06.2024 13:33	3D Object	63.983 KB
Ch17_nonPBR@Running.fbx	1.06.2024 19:38	3D Object	63.776 KB
Ch17_nonPBR@Walking.fbx	1.06.2024 19:32	3D Object	63.807 KB

AdobeMix - SampleScene - Windows, Ma	ac, Linux - Unity 2022.3.14f1 <dx< th=""><th>1&gt;</th><th></th><th></th><th></th></dx<>	1>			
File Edit Assets GameObject Compo	onent Services Window Help				
😝 КЕ 👻 📥					
'≡ Hierarchy a	3 🗄 🗰 Scene 🏼 >> Animato				:
+ - • All	🖉 🛛 🖉 Center 🔻 🌐 Globa	▼   ₩ ▼ 豊 ▼ Ⅲ ▼		📿 🔻 2D 📑 '	≰ <b>\$</b> ▼ 95 ■ ▼ ⊕ ▼
					= ,
중 Main Camera 중 Directional Light	♥ ☆ Ø 20				× ♥ersp
	11 (#				
			*		
	101-01-				
Project E Console			a :	🕶 Game	
+•			P 🎝 🕈 🕈 🚺 ★ 🕫 14	4 Game 🔻 Display 1 🔻 Free Aspect	▼ Scale ●— 1x F
► ★ Favorites	Assets > Worker Ch17_nd				
V 🕞 Assets	If Ch17_nonPBR If Ch17_nonPBR@Admit	nistering Cpr			
Scenes	6 Ch17_nonPBR@ldle				
Worker Cn17_nonPBK     Dackages	Image: Ch17_nonPBR@Knee Ch17_nonPBR@Knee Ch17_nonPBR@Runn				
	Ch17_nonPBR@Walk				

Open our Unity project. Place our folder (*Worker Ch17\_nonPBR*) in the Assets section.

Before we import the character into our scene, we need to make some presets on the FBX files.

When we select the **Ch17\_nonPBR** file with the **T-Pose** position, we see that it has no colors. Let's assign its textures. Extract the textures to a folder by clicking **Materials>Textures>Extract Textures**.



When you click on **Extract Texture**, you will be asked which folder the textures will be extracted to. We will need to open this folder, right-click, give it a name, and select it.

Select Textures Folder		× -		S Q Layers • Layout
⊢ → ∽ ↑ 🛅 > Bu bilgisayar > Yerel Disk	(D:) > UNITY > AdobeMix > Assets > Worker Ch17_nonPBR	Verker Ch17, nonPUR Masor.	i Oinspecto Q + 20 • 14 ad + ¢5 m + ⊕ + Ch 1	Anon PBR Import Settings
				.0
Dutanik • Ven Makar ) © Ondhors - Personal 2 © Devolosis ) © Musiker • © Musiker • © Resenter	^ Değiştirme tərihi ∏ür Böyut   Aramanısta eşleşen öğe yok		≪ Fienge Tratures Material © Ware Promogee > Ch0 Dame	y None (Material)
Demo Construint     Tein Rulosu     Tein Rulosu     Gan Eranden     San Eranden     San Eranden     San Eranden     Sander (C)     Sander	Gorialima (k) → Gorialima (k) → Gruptandama (k) → Vende Geri (k) Kogan (k) Geri (k) Kogan (k) Open (k) Kogan (k) Open (k) Gall here Syster Erijen verin →		CN7,her	
hypet Console	Veri P Kado Ovelikier S Koopi Microsoft Virol Bage C Microsoft Virol Bage C Microsoft Access Date Microsoft Access Date Microsoft Access Date Microsoft Access Date Microsoft Access Date Microsoft Access Date Microsoft Access Date Microsoft Access Date Microsoft Access Date Microsoft Access Date Microsoft Access Date	esi abase t Sunusu Jocument à E @@Game • Ø @ # Ø014 Dame •	Cht7.nev#1	a

Here, the folder was created by naming it **Textures**. Select it with **Select Folder**.

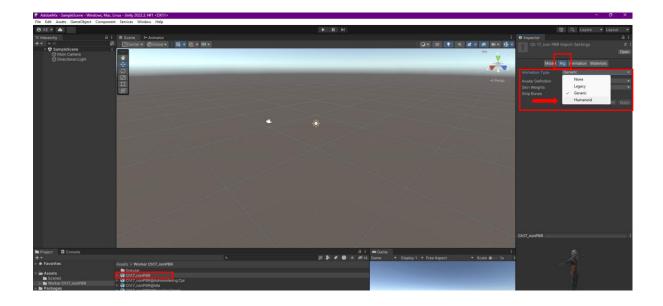
Select Textures Folder							
$\leftarrow \rightarrow \checkmark \uparrow$	Bu bilgisayar 🔸 Yerel Disk (D:) 🔸	UNITY > AdobeMix > Assets > Work	er Ch17_nonPBR >		C Worker Ch17_n	onPBR klasör 🔎	
Düzenle 🔻 Yeni klasör							?
> 📥 OneDrive - Personal	<u>^d</u>	Değiştirme tərihi	Tor	Boyut			
> 🗾 Belgeler	🚞 Dokular	1.06.2024 20:02	Dosya klasörü				
> 🚽 Downloads							
> 🧾 Masaüstü							
> 🕐 Müzikler							
🗸 🔀 Resimler							
> 🦰 Ekran Görüntüleri							
> 🦳 Film Rulosu							
> 🗾 Videolar							
) 📩 Kaan Erarslan							
👻 💻 Bu bilgisayar							
> 🚟 Windows (C:)							
- VI D:-I/(D.)							
Klasö : Du	okular						
					Klasör Seç	İptəl	

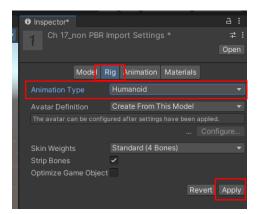
If there is a need for correction in texture assignments, it warns. Fix this by saying **Fix Now**.

1	NormalMap settings	x
	A Material is using the texture as a normal map. The texture must be marked as a normal map in the import settings.	
	Assets/Worker Ch17_nonPBR/Dokular/Ch17_1001_Normal.png Assets/Worker Ch17_nonPBR/Dokular/Ch17_1002_Normal.png	
	Fix now lanore	

Thus, the colors of the worker character appear.

Now, see a series of operations that need to be done on all **FBX** files. The **Animation Types** in the **Rig** section of all downloaded character files must be converted to **Humanoid** form.





Also, some animations need to have some repetition motion. Below, the changes made under **Animations** are shown in two consecutive frames.

Here;

- \* Loop Time and Loop Pose boxes should be activated (checked).
- \* Root Transform Rotation > Bake Into Pose should be active and turn Based Upon into Original.
- \* Root Transform Position (Y) > Bake Into Pose should be active.

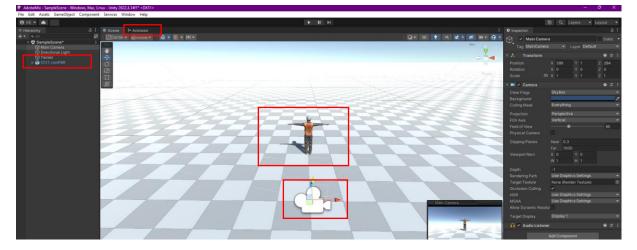
a : Start	0		End 1	
# 1 ▲				
open				
	001.030	~		ן 😐 ו
Cy	cle Offset	0		
Root	Transform Rotatio			
Ba	ke Into Pose	~		n 🕘
Ba	sed Upon	Original		•
		0		
Root	Transform Positio	in (Y)		
			loop match	
				Ţ.
<sup>d in</sup> Off	fset	0		
		n (XZ)		
			loop match	ן 😐 ן
End	sed Upon	Center of Mass		
1.0 Mirror				
	ive Reference Po			
Po				
O: Avera Avera	age Velocity: (0.000, age Angular Y Speed	0.000, 0.000) I: 0.0 deg/s		
30 EDS				
► Motio				
1				
			Rover An	volv
atch 🦲				doub.
mixamo	o.com		2D _‡→ 🏅	<b>*</b> - 1
•			•	
atch 🥘				
▼				
	0:00	(000.0%) Frame 0		
atch 🤤 🚽				•
		Provide     Start     0       Open     Loop Time     Loop Pose       Cycle Offset     Root Transform Rotation       Bake Into Pose     Based Upon       Offset     Root Transform Position       Bake Into Pose     Based Upon (at Stall)       In     Offset       Root Transform Position     Based Upon (at Stall)       Bake Into Pose     Based Upon (at Stall)       In     Offset       Pose Frame     Average Velocity: (0.000, Average Velocity: (0.000, Average Angular Y Speed)       In     Notion       In     Import Messages       In     Import Messages	Start       0         Open       Loop Time       ✓         Loop Pose       ✓       Cycle Offset       0         Root Transform Rotation       Bake Into Pose       ✓       Based Upon       Original         Offset       0       O       Root Transform Position (Y)       Based Upon (at Sto Original         Based Upon (at Sto Original       Offset       0       O         Root Transform Position (XZ)       Bake Into Pose       Based Upon (at Sto Original         Offset       0       Center of Mass         In       Mask       O       Pose Frame       O         Additive Reference P(       Pose Frame       O       Average Velocity (0.000, 0.000)         Verrage Velocity (0.000, 0.000, 0.000)       Average Average Arguing (Y Speed: 0.0 deg/s)       Events         Mask       Mask       Mask       Mask       Mask       Mask         Intent       Intent Messages       Intent Messages       Intent Messages       Intent Messages	Start       0       End       1         Open       Loop Time       -       Loop Pose       loop match         Cycle Offset       0       -       -       -         Root Transform Rotation       Base Into Pose       -       loop match         Base Into Pose       0       -       -       -         Root Transform Position (Y)       Base Into Pose       -       loop match         Based Upon       Original       -       -       -         Offset       0       -       -       -         Based Upon (XZ)       Based Upon Center of Mass       -       -         Based Upon       Center of Mass       -       -         Or Frame       0       -       -       -         Pose Frame       0       -       -       -         Additive Reference Pr       -       -       -       -         Pose Frame       0       -       -       -         Additive Reference Pr       -       -       -       -         Mask       -       -       -       -       -         Motion       -       -       -       -       -

In this way, all **FBX** files should be converted to **Humanoid** form, changes should be made in the **Animations** section and **Apply** should be done.

After the operations on the files included in the project are completed, we can move on to the animation and animator phase.

Let's add a Terrain to our scene to provide freedom of movement.

Then, position the Idle (fbx with T-Pose) file and the Main Camera to see it.



If you notice, since the characters have animation, the **Animator** tab also appeared next to the **Scene** tab. If this is not opened, we can open it by pressing **Windows>Animation>Animator**.

Create a folder under Assets named Create>Folder>Character Animator and go into it.

			/	1				
Project 🖻 Console								а:
+-				民	•	•	•	<b>1</b> 5
▶ ★ Favorites	Assets > Character An	imator						
<ul> <li>C⇒ Assets</li> <li>Character Animator</li> <li>Scenes</li> <li>Worker Ch17_nonPBR</li> <li>Packages</li> </ul>								

Here, create an animator controller with **Create>Animator Controller** and name it **WorkerController**.

Project 🖹 Console		
+-		٩
▶ ★ Favorites	Assets > Character Animator	
🔻 🖨 Assets	🔁 WorkerController	
Character Animator		
Scenes		
Worker Ch17_nonPBR		
Packages		

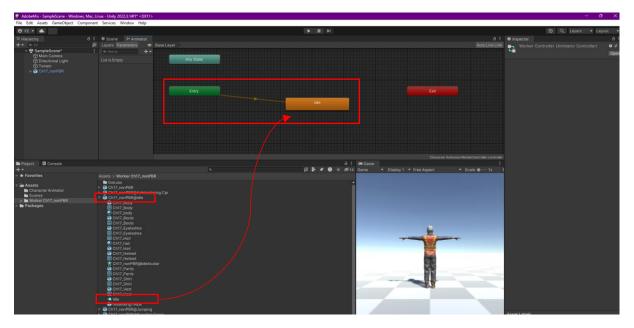
When this process is done, **three states** are created in the **editor** in the **Animator** tab, **including Any State**, **Entry** and **Exit** blocks.

E 👻 📥				③ Q. Layers • Layout
				1 0 Inspector
			Auto Live L	mi Worker Controller (Animator Controller)
SampleScene*				
SampleScene*     Main Camera     Directional Light     Orerain     Gr17_nonPBR		ny Slave		
		Enty		

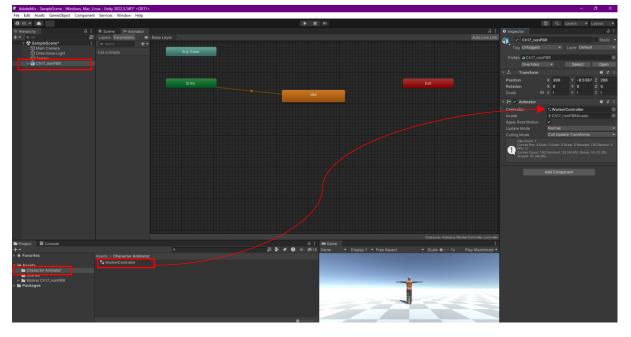
First, make animations of the transition from the **idle** position to **walking** and **running** states.

Drag and drop the **Idle** animation **Idle** listed under the **Idle fbx** file to the animator editor. The **Idle** status will appear in the editor.

Also, a link will be automatically established from **Entry** to **Idle** status.



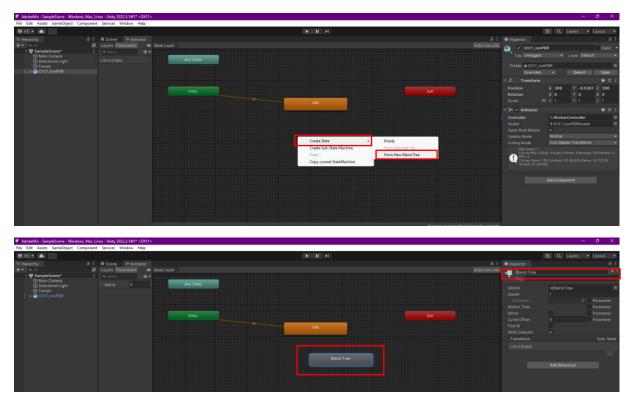
Another process is to connect the controller named **Assets>Character Animator>WorkerController** to the empty **Inspector>Animator>Controller** section of our **Ch17\_nonPBR** object in **Hierarchy**.



When we switch to **Player Mode**, we see that our worker no longer stands in **T-Pose** and makes slight oscillations in the **Idle** position.

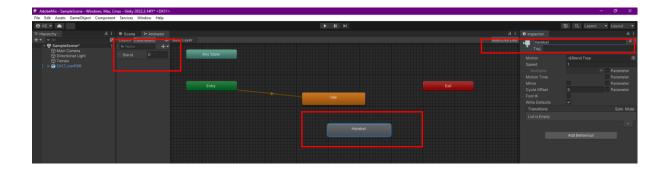


See the method that is normally used to establish a connection between states in the next step. Since the **walking** and **running** movements are connected and **transitional**, create a state that will **blend** them. **Right-click** in the editor and select **Create State>From New Blend Tree**.



Change the name of this state to **Hareket**. The **Blend Tree** state can also be thought of as an organizer where a mix of movements can be made and passed.

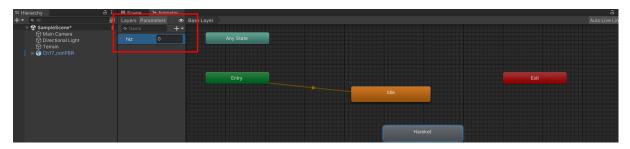
Another noteworthy point is that a parameter called **Blend** has been created in the **Parameters** section of the **Animator** tab.



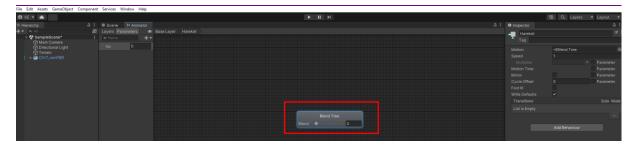
Right-click on the **Parameters>Blend** parameter and delete it with **Delete**.



Instead, let's press + and add a **Float** type parameter, which is preferred in transition **animations** and which we will call **hiz**.



When we double-click on the movement, a layout and inspector will appear.



When the **Blend Tree box** is selected, **hiz** must be selected as the **Parameter**.

File Edit Assets GameObject	Componen	t Services Window H	Jelp	 						
Ө КЕ 🔹 📥					► II H				③ Q. Layers	
≔ Hierarchy + ▼								0 Inspector		a I
+ • [] @r All								Blend Tree		<b>U</b> :
<ul> <li></li></ul>								Blend Type		•
									Blend	
										hiz
					والمستحد المتحديق والمراجع					
					Blend Tree					
						0				

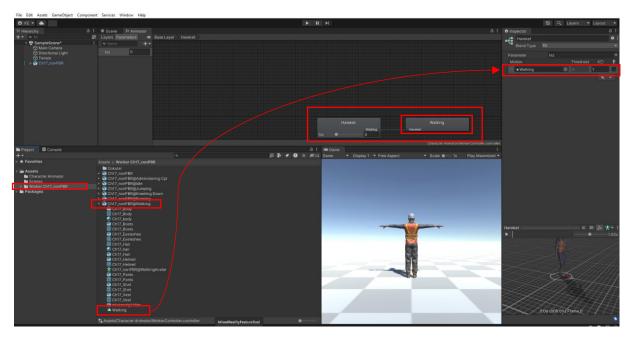
In this case, the adjustment line is also displayed in the **Blend Tree** box.

Өкс 🔹 📥			③ Q. Layers	
		6 Inspector		a i
		228 Hareket		
♥ SampleScene* Main Camera		Hareket Blend Type		
© Main Camera © Directional Light © Terrain ► @ Ch17_nonPBR		Parameter Motion		

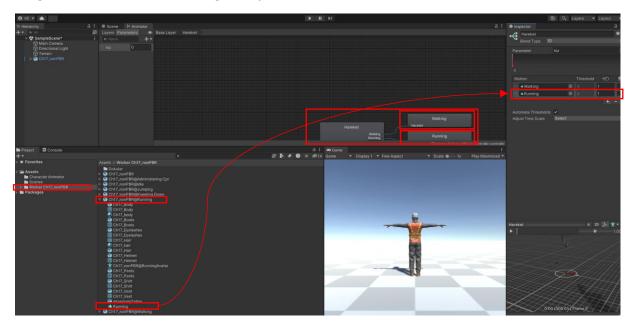
We can start adding motions to blend. Add a motion field with **Inspector>Motion>+>Add Motion Field**.



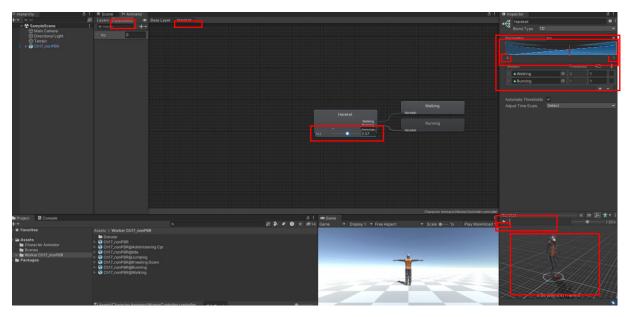
Let's **drag** the **Assets>Worker Ch17\_nonPBR>Walking** animation to the opened area. This process also added the **Walking** animation to the **Blend Tree (Hareket)**.



Drag our other animation Running to Inspector>Motion>+>Add Motion Field.

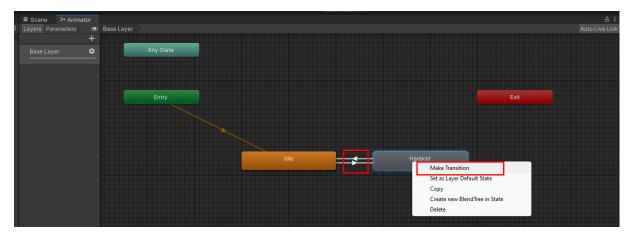


Here, the **Parameter** graph shows the **Walking-Running** transition **from 0 to 1**. Start the animation in the **Hareket** animation preview window below and test how it accelerates from **walking** to **running** by increasing the **hiz** setting on the **Hareket** box in the animator **from 0 to 1**.

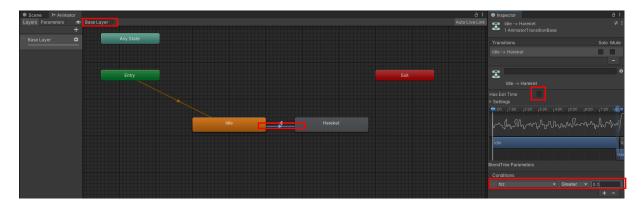


So, with **Blend Tree** we have achieved the **blending** and **transition** of two animations.

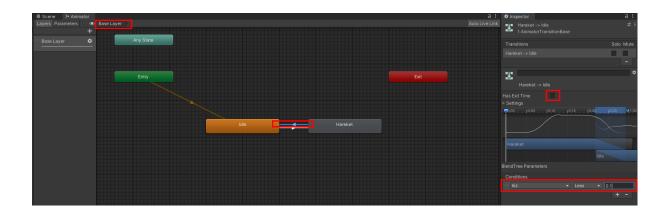
Now, go to the **Base Layer** and create a link between **Idle** and **Hareket** (**Blend Tree**). To do this, rightclick on **Idle** and add a link to the **Hareket** box with **Make Transition**. Similarly, add a link to **Idle** from the **Hareket** box (**state**) with **Make Transition**.



Click on the direction arrow between **Idle** and **Hareket** to switch from **Idle** to **Hareket** (Blend Tree). In the Inspector section, **uncheck** the **Has Exit Time** box. We should add the condition that will provide the transition to the **Conditions** section. By pressing +, the number box where we will specify the number that is the **hiz** parameter and the **walking** condition [**Greater**] comes to the list. We can enter a value like **0.1** here.



For the worker to stop, the **hiz** parameter must be less than 0.1 (**Less**). For this, select the arrow in the opposite direction. Unselect the **Has Exit Time** box and add to **Conditions** with **+**. Here, select **Less** instead of **Greater** and write 0.1 in the **hiz** box.



Thus, we have completed the necessary operations in the animator section. Now, when the speed (hiz) goes above 0.1, it walks (Walking); above 0.5, it starts running (Running); and below 0.1, it stops (Idle).

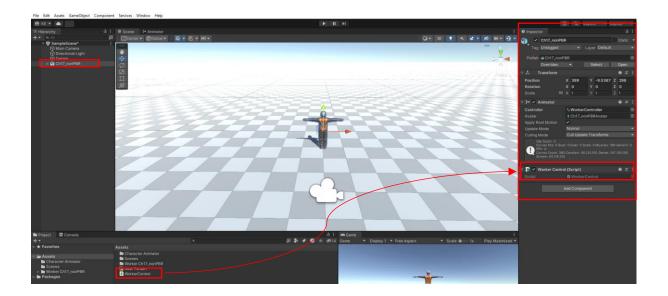
Now it's time for the keystroke to trigger these movement processes. Here, we need to provide **stop-walk-run** controls when a **key** is pressed on the **keyboard**. Therefore, we have to code the **C#** script.

Let's create a file under Assets with Create>C# Script. WorkerControl is used as the file name here.

Double-click on the **WorkerControl.cs** file created under **Assets** and open it in Visual Studio. If Visual Studio is not installed, it will open in the editor we specified as **Preferences>External Tools>External Script Editor** (example: Notepad).

AdobeMix - SampleScene - Windo ile Edit Assets GameObject C	omponent Services Window Help				
≡ Hierarchy	a : # Scene ≻ Animator				
+ • • All	🔊 Layers Parameters 👁 Ba	se Layer		Auto Live Link	
			Folder		
😭 Main Camera 😭 Directional Light	= Base Layer 🌼	Any State	C# Script		
			2D >		
► 🏟 Ch12_nonPBR			Visual Scripting >		
			Shader >		
			Shader Variant Collection		
			Testing >	Exit	
			Playables >		
			Assembly Definition		
			Assembly Definition Reference		
			TextMeshPro >		
		Create	> Text >		
		Show in Explorer	Scene	Hareket	
		Open			
		Delete	Scene Template Scene Template From Scene		
		Rename			
		Copy Path Alt+	Scene Template Pipeline Ctrl+ C Prefab		
		Open Scene Additive	Prefab Variant		
		View in Package Manager	Audio Mixer		
		Import New Asset	Material		
		Import Package	> Material Variant		
		Export Package	Lens Flare		
		Find References In Scene	Render Texture		
		Select Dependencies	Lightmap Parameters		
		Refresh	Ctrl+R Lighting Settings		
		Reimport	Custom Render Texture	Character Animator/WorkerController.controll	
Project E Console		Reimport All	Animator Controller	■ Display 1 ▼ Free Aspect ▼ Scale ● 1x Play Maximized	
<ul> <li>★ Favorites</li> <li>☞ Assets</li> <li>■ Character Animator</li> </ul>	Assets Character Animator Scenes Worker Ch17_nonPBR	Extract From Prefab	Animation Animator Override Controller		
		Update UXML Schema	Avatar Mask		
Scenes Worker Ch17_nonPBR	New Terrain	Generate Lighting Ctrl+	Shift+L Timeline		
■ Worker Ch17_nonPBR		Open C# Project	Signal		
		View in Import Activity Window	Physic Material		
		Properties	Alt+P GUI Skin		

Now, connect the script file we created to our worker on the scene.

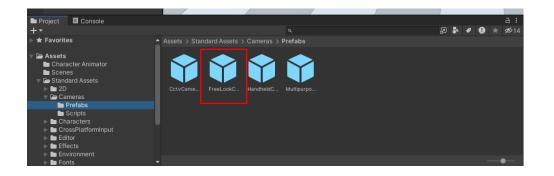


The **WorkerControl.cs** file content is given at the end of the topic. However, before using it, **Standard Assets** should be imported for camera control. This asset belongs to **Unity Technologies** and is not currently available in the Asset Store. However, it is available on the course's **Google Drive**.

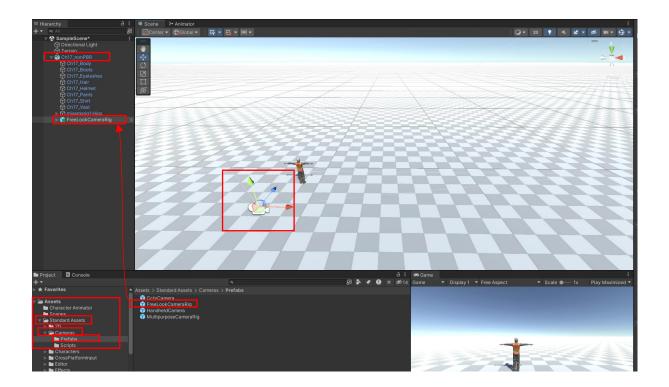
After the import process of the Standard Assets package is completed.

#### Assets>Standard Assets>Cameras>Prefabs>FreeLookCameraRig

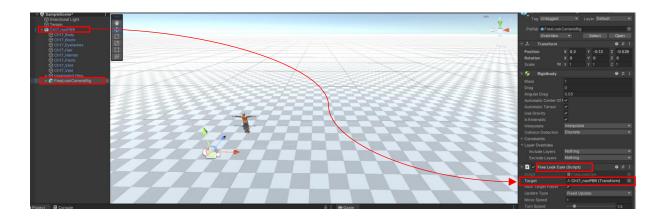
We will add the camera named above to our scene. The purpose here is to follow the worker's movements from different angles depending on the mouse.



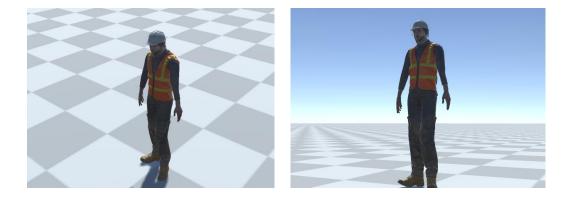
Connect this camera to our scene, to our **Ch17\_nonPBR** worker object. Remove the **Main Camera** to avoid conflicts.



In order for the camera to follow the worker, the worker object (Ch17\_nonPBR) in the Hierarchy must be connected to the FreeLookCameraRig's Inspector>Free Look Cam (Script)>Target.



In this way, the camera can be rotated around the worker with mouse movements.



Another thing that is missing is the worker being able to turn in the direction the camera is pointed at and look at the camera.

This has been achieved by adding the FreeLookCameraRig camera and coding accordingly.

Before switching to play mode, it will work after advancing to the end of the topic, writing the code file completely or downloading it from **Google Drive** and connecting it to our worker object. It is important to follow the live narration in the course to avoid any problems.

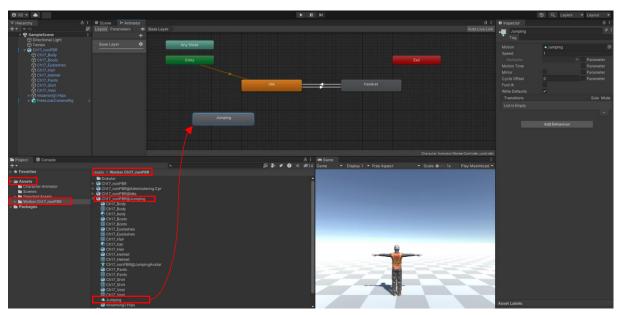
The operations performed up to this point were performed for **standing**, **walking** and **running** movements. Due to the transitional nature of **walking** and **running**, **Blend Tree** and **Float** were used as parameter type.

Now, see how to integrate some animations that we downloaded from **Mixamo**. Here, assuming that there is no transitional movement, we will cover the system binding of the **jumping**, **kneeling** and **administering Cpr** animations. These movements will be linked to the **Idle** position. **Jumping** and **kneeling** will be linked directly to **Idle**, while **Administering Cpr** will be linked to the **Kneeling** movement.

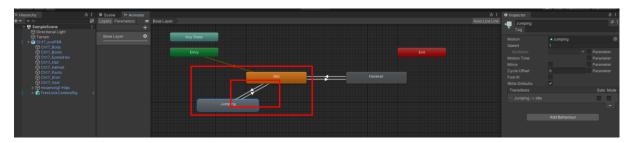
Let's start with Jumping.

Come to our animator window.

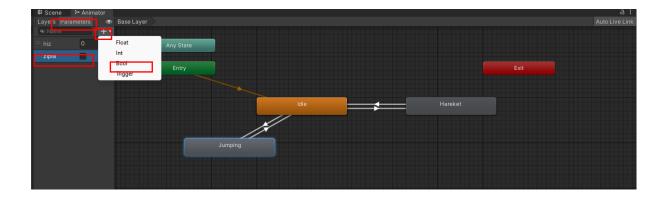
### Drag the animation named **Jumping** under Assets>WorkerCh17\_nonPBR>Ch17\_nonPBR>Ch17\_nonPBR@Jumping to the animator.



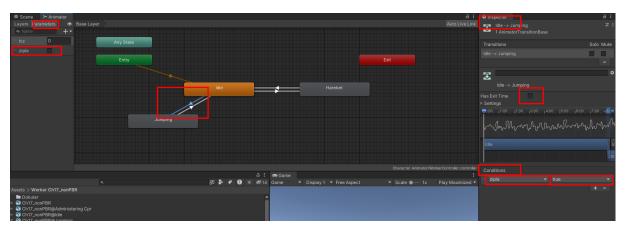
To establish the connection between **Idle** and **Jumping**, right click on **Idle** and use the direction line with **Make Transition**. Similarly, right click on **Idle** from **Jumping** and establish the connection line with **Make Transition**.



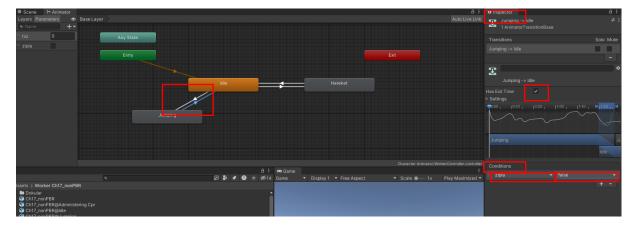
Now, we need to specify which parameter we will use to provide this connection. Add a **Bool** type parameter to the **Parameters** section with +. Here the name **zipla** is given.



At this stage, select the **Idle->Jumping** link. In the **Inspector**, unselect **Has Exit Time**. In **Conditions**, select **zipla** from the list by pressing +. We don't change the **true** selection because it will **jump** when the key is pressed.



Similarly, select **Jumping->Idle** link. In the **Inspector**, unselect **Has Exit Time**. In **Conditions**, select **zipla** from the list that opens with **+** and set its condition to **false**.

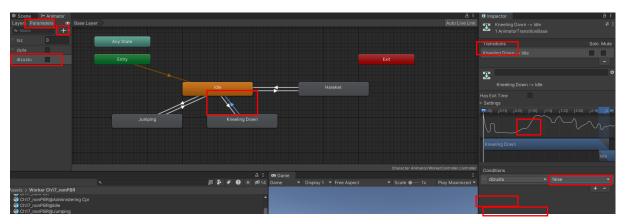


Test this **animation**, which is connected to the **Space** bar key in the code line, by pressing the **Space** bar while in **Idle** mode in **Play** mode.



Similarly, add the **Kneeling Down** animation. Drag the relevant animation file from **Assets** to the **animator**. Establish the connections with **Idle**. **Add** the **Bool** type variable to the **Parameters** section.

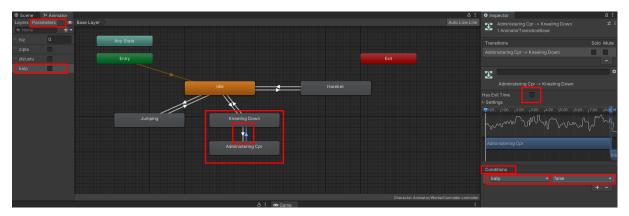
Make the **true** and **false** selections by adding the **Has Exit Time** and **Conditions** lists in the connections.



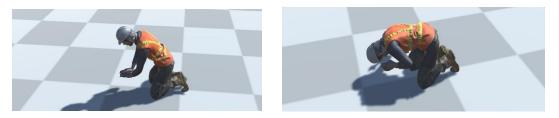
In the coding, the **Z** key was used in the laptop collapse animation. When you press **Z**, the laptop will collapse and when you pull it, it will rise. After testing in player mode, let's move on to our other movement.



Here, connections were established between **Kneeling Down** and **Administering Cpr** with **Make Transition's**. **Bool** type **kalp** variable was added to **Parameters**. In the **Inspector**, **Has Exit Time** and **Conditions->kalp**; **true** and **false** operations were done with similar repetitions.



After setting the codes to press the **X** key, test it in **Play** mode.



## 12.1.Relevant C# Script Codes

Let's create a C# file named **WorkerControl.cs** under **Assets**. The file name should not be different from this.

The reason for this is that the file name and the **class** name inside must be the same.

#### public class WorkerControl : MonoBehaviour

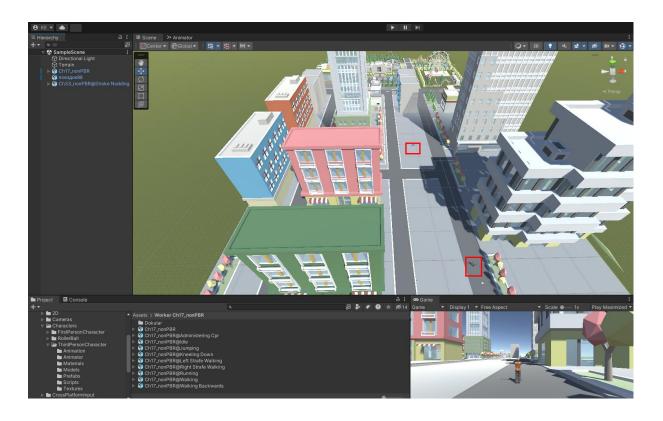
**Codes** can be created by writing them in the Visual Studio editor or an editor such as Notepad, or by copying/pasting the file content given below.

The more practical way is to use the ready-made version of the file.

The script file is also located in the **C# Script Files** folder of **Google Drive**, where the course documents are located.

After downloading this file from **Drive**, it can be placed in the Assets section of the project and linked to our **Ch17\_nonPBR** object.

The scene was tested in this environment as well, by adding a low-poly city (from Sketchfab) and a casualty lying on the ground (from Mixamo) to make it an **emergency response** scenario for a casualty.



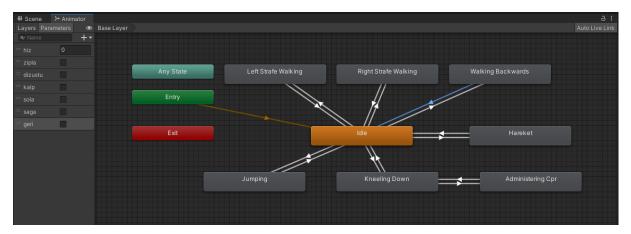


It is also possible to download animations for right, left and backward movements from Mixamo, add them to the animator and add coding.

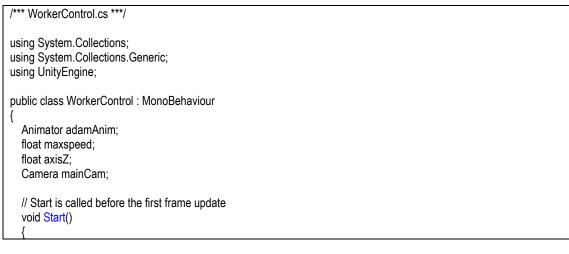
Here;

A: left, S: back, D: right, R: 180° rotation

It is coded to ensure orientation.



The final version of the **WorkerControl.cs** code file with these additions is below and on **Google Drive**.



```
adamAnim = GetComponent<Animator>();
  mainCam = Camera.main;
}
// Update is called once per frame
void Update()
{
  if (Input.GetKey(KeyCode.W)) // walking
  { maxspeed = 0.3f;
     axisZ = maxspeed * Input.GetAxis("Vertical");
   if (Input.GetKey(KeyCode.W) && Input.GetKey(KeyCode.LeftShift)) // running
     { maxspeed = 1.0f;
       axisZ = maxspeed * Input.GetAxis("Vertical"); }
  }
  else // standing
  { maxspeed = 0.0f;
     axisZ = maxspeed* Input.GetAxis("Vertical");
  // walking to right
  if (Input.GetKeyDown(KeyCode.D))
  { adamAnim.SetBool("saga", true); }
  if (Input.GetKeyUp(KeyCode.D))
  { adamAnim.SetBool("saga", false); }
  // walking to left
  if (Input.GetKeyDown(KeyCode.A))
  { adamAnim.SetBool("sola", true); }
  if (Input.GetKeyUp(KeyCode.A))
  { adamAnim.SetBool("sola", false); }
  // walking back
  if (Input.GetKeyDown(KeyCode.S))
  { adamAnim.SetBool("geri", true); }
  if (Input.GetKeyUp(KeyCode.S))
  { adamAnim.SetBool("geri", false); }
  // iumpina
  if (Input.GetKeyDown(KeyCode.Space))
  { adamAnim.SetBool("zipla", true); }
  if (Input.GetKeyUp(KeyCode.Space))
  { adamAnim.SetBool("zipla", false); }
  // kneeling
  if (Input.GetKeyDown(KeyCode.Z))
  { adamAnim.SetBool("dizustu", true); }
  if (Input.GetKeyUp(KeyCode.Z))
  { adamAnim.SetBool("dizustu", false); }
  // heart massage-artificial respiration
  if (Input.GetKeyDown(KeyCode.X))
  { adamAnim.SetBool("dizustu", true);
    adamAnim.SetBool("kalp", true); }
  if (Input.GetKeyUp(KeyCode.X))
  { adamAnim.SetBool("dizustu", false);
     adamAnim.SetBool("kalp", false); }
  // to switch between the animations Vector3.ClampMagnitude can be used
  // to do this a variable like Vector3 is necessary; it is named (vektor) here
  Vector3 vektor = new Vector3 (0, 0, axisZ);
  adamAnim.SetFloat("hiz", Vector3.ClampMagnitude(vektor, 1f).magnitude, 1f, Time.deltaTime * 3f);
```

```
//adamAnim.SetFloat("hiz", maxspeed);
// to make the worker and the assigned Free Look Camera follow the mouse
Vector3 kameraYon = mainCam.transform.TransformDirection(vektor);
kameraYon.y = 0.0f;
transform.forward = Vector3.Slerp(transform.forward, kameraYon, Time.deltaTime * 3f);
//transform.forward = kameraYon;
}
```

In this way, some animations required for a wounded intervention scenario have been fulfilled.

**It should not be forgotten** that a healthy project is more possible in a **classroom** environment and in live lessons. Because in this way, there is a chance to intervene in problems that may occur in the processes of the trainees.

# BIBLIOGRAPHY

Abbaszadeh, S., and Rastiveis, H. (2017). "A comparison of close-range photogrammetry using a nonprofessional camera with field surveying for volume estimation," in ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-4/W4, (Heipke: International Society for Photogrammetry and Remote Sensing), 1–4. DOI: 10.5194/isprs-archives-XLII-4-W4-1-2017.

Adams, D. M., Pilegard, C., & Mayer, R. E. (2016). Evaluating the cognitive consequences of playing portal for a short duration. Journal of Educational Computing Research, 54(2), 173–195.

Aerial-Craft (2022). Rawang Mine Quarry, Malaysia. Retrieved from <u>https://sketchfab.com/3d-models/rawang-quarry-2a-11jan17-malaysia-f3eae47abe694f36a9a98578315acc1e</u>

Ahmad, S. M. S., Fauzi, N. F. M., Hashim, A. A., & Zainon, W. M. N. W. (2013). A study on the effectiveness of computer games in teaching and learning. International Journal of Advanced Studies in Computers, Science and Engineering, 2(1), 1.

Ahn, J., Ahn, E., Min, S., Choi, H., Kim, H., and Kim, G. J. (2019). "Size perception of augmented objects by different AR displays," in HCI International 2019—Posters, Vol. 1033, ed. C. Stephanidis (London: Intech Open), 337–344. DOI: 10.1007/978-3-030-23528-4\_46

Al-Ansi, A.M., Jaboob, M., Garad, A., Al-Ansi, A., 2023. Analyzing augmented reality (AR) and virtual reality (VR) recent development in education. Social Sciences & Humanities, Volume 8, Issue 1, 1-10.

Aldrich, C. (2009). Virtual worlds, simulations, and games for education: A unifying view. Innovate: Journal of Online Education, 5(5), 1.

Alhalabi, W. S. (2016). Virtual reality systems enhance students' achievements in engineering education. *Behaviour & Information Technology*, *35*(11), 919-925. https://doi.org/10.1080/0144929X.2016.1212931.

Aloqaily, M, Bouachir, O and Karray, F, 2023. Digital twin for healthcare immersive services: fundamentals, architectures, and open issues, Digital Twin for Healthcare, Ed. El Saddik, A., Academic Press, 39-71, ISBN 9780323991636,

https://www.sciencedirect.com/science/article/pii/B9780323991636000081.

American Society for Photogrammetry and Remote Sensing (2019). What is ASPRS?. Available online at: <u>https://www.asprs.org/organization/what-is-asprs.html</u>

Apple Vision Pro, 2024. <u>https://www.apple.com/apple-vision-pro/</u>

Atay, (2024). Atay Holding, <u>https://www.atay.com.tr/atay-holding-alm-m</u>.

Azuma, R.T., (1997). A survey of augmented reality. Presence: Teleoperators and Virtual Environments, 6(4), 355-385.

Bartlett, J. D., Lawrence, J. E., & Khanduja, V. (2018). Virtual reality hip arthroscopy simulator demonstrates sufficient face validity. Knee Surgery, Sports Traumatology, Arthroscopy. <u>https://doi.org/10.1007/s00167-018-5038-8</u>.

Beckem, J. M., & Watkins, M. (2012). Bringing life to learning: Immersive experiential learning simulations for online and blended courses. Journal of Asynchronous Learning Networks, 16(5), 61–70.

Bellanca, J. L., Orr, T. J., Helfrich, W. J., MacDonald, B., Navoyski, J., & Demich, B. (2019). Developing a virtual reality environment for mining research. 2019 SME Annual Conference and Expo and CMA 121st National Western Mining Conference, 597–606.

Bhai, R., (2024). Factory 3D Model, https://sketchfab.com/3d-models/factory-3d-modele723f4fe48ec4b9da52ec6e4a442286b

Buckless, F. A., Krawczyk, K., & Showalter, D. S. (2014). Using virtual worlds to simulate real-world audit procedures. Issues in Accounting Education, 29(3), 389–417.

Capturingaworld, (2024). Roadstone Quarry, Allenwood, Sketchfab. <u>https://sketchfab.com/3d-models/roadstone-quarry-allenwood-3d-mesh-1e14c3c1c67347d7b1e0de7eeb3de996</u>.

Cheng, K.H. and Tsai, C.C. (2012). Affordances of augmented reality in science learning: Suggestions for future research. Journal of Science Education and Technology, 22, 449-462. DOI: 10.1007/s10956-012-9405-9.

Cohen, L., Manion, L., & Morrison, K. (2007). *Research methods in education*. London: Routledge.

DePorres, D., & Livingston, R. E. (2016). Launching new doctoral students: Embracing the Hero's journey. Developments in Business Simulation and Experiential Learning, 43(1), 121–128. Engati, 2024. Augmented Reality application. <u>https://www.engati.com/glossary/augmented-reality</u>

El Jamiy, F., and Marsh, R. (2019). Survey on depth perception in head mounted displays: distance estimation in virtual reality, augmented reality, and mixed reality. IET Image Process. 13, 707–712. DOI: 10.1049/iet-ipr.2018.5920.

Erarslan, K., (2022). Augmented Reality Applications on Quarries and Mines, Journal of Scientific Reports-B, 3, June, 13-24.

Fedorko, G. (2021). Application possibilities of virtual reality in failure analysis of conveyor belts. Engineering Failure Analysis, 128(May), 105615. https://doi.org/10.1016/j.engfailanal.2021.105615

Fei, D., & Anbi, Y. (2011). Safety education based on virtual mine. Procedia Engineering, 1922-1926.

Fischer, X., (2024). Asperge Cavity full 3D Network and Landmarks, Sketchfab, <u>https://sketchfab.com/3d-models/realistic-underground-basecave-40-</u> <u>46466ec0558945e9aac9dad15aaeb9f3</u>.

Fonseca, H, (2024). Coal Mill, Grabcad, <u>https://grabcad.com/library/13-coal-mill-department-1</u>.

Fonstad, M. A., Dietrich, J. T., Courville, B. C., Jensen, J. L., and Carbonneau, P. E. (2013). Topographic structure from motion: a new development in photogrammetric measurement. Earth Surf. Process. Landforms 38, 421–430. DOI: 10.1002/esp.3366.

Foster, P. J., & Burton, A. (2006). Modelling potential sightline improvements to underground Mining vehicles using virtual reality. Institution of Mining and Metallurgy. Transactions. Section A: Mining Technology, 115(3), 85–90. <u>https://doi.org/10.1179/174328606X128714</u>

Franklin, G., (2024). Latform for Jaw Crusher, Grabcad, https://grabcad.com/library/plataform-for-jaw-crusher-1

Freina, L., & Ott, M. (2015). A Literature Review on Immersive Virtual Reality in Education: State of the Art and Perspectives. eLearning & Software for Education, (1).

Game Developer, 2024. <u>https://www.gamedeveloper.com/game-platforms/exploring-the-pc-game-engine-landscape</u>

Garcia, F., (2024). Platform for Jaw Crusher, Grabcad. https://grabcad.com/library/plataform-for-jaw-crusher-1

Gegenfurtner, A., Quesada-Pallarès, C., & Knogler, M. (2014). Digital simulation-based training: A meta-analysis. British Journal of Educational Technology, 45(6), 1097–1114. HTC Vive VR Headset. <u>https://www.vive.com/us/</u>

Geithner, S., & Menzel, D. (2016). Effectiveness of learning through experience and reflection in a Project Management simulation. Simulation & Gaming, 47(2), 228–256 (2016). Doi: 1046878115624312.

GEOPS U-Paris-Saclay, (2024). Malpierre Quarry, Sketchfab. https://sketchfab.com/3d-models/malpierre-34085fa5c4d04335882f25f3b4ee94e7

GeoWeek News, 2024. Hololens mining 3d data revolution. https://www.geoweeknews.com/news/hololens-mining-3d-data-revolution

Giovanello, S. P., Kirk, J. A., & Kromer, M. K. (2013). Student perceptions of a role-playing simulation in an introductory international relations course. Journal of Political Science Education, 9(2), 197–208.

Glen, I., (2024). A longwall coal mine layout with shearing machine, 3D Warehouse, https://3dwarehouse.sketchup.com/model/5dfdd8d8f02863aa873da0b62456d2f9/UG-Longwall-coal-mine-layout-with-shearing-machine-track-roof-support

Grabcad, (2024). https://grabcad.com

Gupta, P., 2021. VR simulation solutions for the mining industry-2. Tecknotrove. <u>https://tecknotrove.com/newsletter/vr-simulation-solutions-for-the-mining-industry-2/</u>

Gül, Y (2019). Açık Maden İşletmelerinde İnsansız Hava Aracı (İHA) Uygulamaları. Jeoloji Bülteni. Cilt 62, Sayı 1, 99 - 112, 01.01.2019. https://doi.org/10.25288/tjb.519506.

Gürer, S., Sürer, E., Erkayaoğlu, 2023. MINING-VIRTUAL: A comprehensive virtual reality-based serious game for occupational health and safety training in underground mines. Safety Science, 166, 1-13.

Gürer, S., 2021. Development of a Virtual Reality-Based Serious Game for Occupational Health And Safety Training in Underground Mining, Middle East Tech Univ, MSc Dissertation, 91 pg.

Huang, H.-M., Rauch, U., & Liaw, S.-S., 2010. Investigating learners' attitudes toward virtual reality learning environments: Based on a constructivist approach. *Computers and Education*, *55*(3), 1171-1182. <u>https://doi.org/10.1016/j.compedu.2010.05.014</u>.

Huang, Hsiu-Mei, & Liaw, S.-S., 2018. An Analysis of Learners' Intentions Toward Virtual Reality Learning Based on Constructivist and Technology Acceptance Approaches. *The International Review of Research in Open and Distributed Learning*, *19*(1). <u>https://doi.org/10.19173/irrodl.v19i1.2503</u>

Hugues, O., Gbodossou, A., & Cieutat, J.-M. (2012). Towards the Application of Augmented Reality in the Mining Sector: Open-Pit Mines. International Journal of Applied Information Systems, 4(6), 27–32. https://doi.org/10.5120/ijais12-450760

IoT & Industry 4.0, (2020). b.telligent. <u>https://www.btelligent.com/en/portfolio/industry-40/</u>

Kanani, H. (2019). AR and VR in Mining Industry : Transforming the Future. Plutomen, Oct.1, https://pluto-men.com/ar-and-vr-in-mining-industry-transforming-the-future/#

Kavanagh, S., Luxton-Reilly, A., Wuensche, B., & Plimmer, B. (2017). A Systematic Review of Virtual Reality in <u>Education</u>. *Themes in Science and Technology Education*, *10*(2), 85-119.

Kebo, V., & Staša, P. (2012). Mining processes control and virtual reality. Proceedings of the 2012 13thInternationalCarpathianControlConference,ICCC2012,274–277.https://doi.org/10.1109/CarpathianCC.2012.6228653

Kelly, I, 2022. The Drift: Device bringing augmented reality to the mining face. Northern Ontario Business. <u>https://www.northernontariobusiness.com/industry-news/mining/the-drift-device-bringing-augmented-reality-to-the-mining-face-5772449</u>

Kesim, M. & Özarslan, Y. (2012). Augmented reality in education: current technologies and the potential for education. Procedia - Social and Behavioral Sciences, 47, 297-302.

Martin, S., Diaz, G., Sancristobal, E., Gil, R., Castro, M., & Peire, J. (2011). New technology trends in education: Seven years of forecasts and convergence. Computers and Education, 57(3), 1893-1906.

Kim, H., & Choi, Y. (2019). Performance comparison of user interface devices for controlling mining software in virtual reality environments. Applied Sciences (Switzerland), 9(13). https://doi.org/10.3390/app9132584.

Kirner, T.G., Reis, F.M.V., & Kirner, C. (2012). Development of an interactive book with Augmented Reality for teaching and learning geometric shapes. Information Systems and Technologies (CISTI), 1-6.

Kunz, B. R., Wouters, L., Smith, D., Thompson, W. B., and Creem-Regehr, S. H. (2009). Revisiting the effect of quality of graphics on distance judgments in virtual environments: a comparison of verbal reports and blind walking. Atten. Percept. Psychophys. 71, 1284–1293. DOI: 10.3758/APP.71.6.1284.

Jones, R., & Bursens, P. (2015). The effects of active learning environments: How simulations trigger affective learning. European Political Science, 14(3), 254–265.

Kızıl, M. ve Joy, J. (2001). What can Virtual Reality do for Safety? University of Queensland, St. Lucia QLD.

Kızıl, M. S., Kerridge, A. P., and Hancock, M. G. (2004). Use of virtual reality in mining education and training. CRCMining Conference, Noosa Head, Queensland, Australia, 15-16 June 2004. Brisbane, Australia: Cooperative Research Centre - Mining (CRCMining).

Li M., Sun, Z., Jiang, Z., Tan, Z. ve Chen, J., 2020. A Virtual Reality Platform for Safety Training in Coal Mines with AI and Cloud Computing, Discrete Dynamics in Nature and Society, vol. 2020, ID 6243085, 7 pg.

Lin, T.-J., & Lan, Y.-J. (2015). Language learning in virtual reality environments: Past, present, and future. *Educational Technology and Society*, *18*(4), 486-497.

Martin, S., Diaz, G., Sancristobal, E., Gil, R., Castro, M., and Peire, J. (2011). New technology trends in education: Seven years of forecasts and convergence. Computers and Education, 57 (3), 1893-1906.

Michalak, D. (2012). Applying the augmented reality and RFID technologies in the maintenance of mining machines. Lecture Notes in Engineering and Computer Science, 1, 256–260.

Microsoft Hololens, 2024. https://www.microsoft.com/tr-tr/hololens

Moore, P., 2021. Immersive Technologies boosts worksite VR platform new mine standards training tool. International Mining. https://im-mining.com/2021/03/30/immersive-technologies-boosts-worksite-vr-platform-new-mine-standards-training-tool/

Njamga, K. (2024). Gravel pit. Grabcad, https://grabcad.com/kelean.njamga-1.

Nickel, C., Knight, C., Langille, A., & Godwin, A. (2019). How Much Practice Is Required to Reduce Performance Variability in a Virtual Reality Mining Simulator? Safety(5(2)), 18.

<u>Oculus Quest 3 Metaverse, 2024. https://www.meta.com/quest/quest-3https://www.meta.com/quest/quest-3/</u>

Outcropwizard, Bonn, Germany (2022). Grube Theresia, Morshausen, Germany. Retrieved from <a href="https://sketchfab.com/3d-models/ppc-kgale-2016-09-08-677a883d45ea48fbb974a0470f98a2ed">https://sketchfab.com/3d-models/ppc-kgale-2016-09-08-677a883d45ea48fbb974a0470f98a2ed</a>

Özalp, R., 2024. Blender ile Açık Ocak Tasarımı, Proje Çalışması. GSF, DPÜ.

Patrimoine, L. D., (2024). Coal Mine Gallery, Sketchfab. <u>https://sketchfab.com/3d-models/coal-mine-gallery-2cc9dfc11fe243039f9900f0c31414ae</u>

Pine, J. 2018. 10 Real Use Cases for Augmented Reality AR is set to have a big impact on major industries. https://www.inc.com/james-paine/10-real-use-cases-for-augmented-reality.html

Premier Mapping, Cullinan, South Africa, (2024). PPC KGale Quarry, Botswana, Retrieved from https://sketchfab.com/3d-models/ppc-kgale-2016-09-08-677a883d45ea48fbb974a0470f98a2ed

Premier Mapping, Cullinan, South Africa, (2024). Lyttleton Quarry, New Zealand. Retrieved from https://sketchfab.com/3d-models/2016-09-01-lyttelton-05612c2bc3844d249b905a21f05aa594

Rice, R. (2009). The augmented reality hype cycle. <u>http://www.sprxmobile.com/the-augmented-reality-hype-cycle /2014</u>.

Rigmodels, (2024). https://rigmodels.com

Ritz, L. T., & Buss, A. R. (2016). A Framework for Aligning Instructional Design Strategies with Affordances of CAVE Immersive Virtual Reality Systems. *TechTrends*, *60*(6), 549-556. <u>https://doi.org/10.1007/s11528-016-0085-9</u>.

Rutten, N., van Joolingen, W. R., & van der Veen, J. T. (2012). The learning effects of computer simulations in science education. Computers & Education, 58(1), 136–153.

Samsung Gear VR, 2024. https://www.samsung.com/tr/support/model/SM-R322NZWATUR/

Sanlab, 2024. https://sanlab.net/simulator-systems/vr-heavy-equipment-simulator/

Sap, 2024. https://www.sap.com/mena/products/scm/industry-4-0/what-is-augmented-reality.html

Sarkar A. S., (2024), Flotation Machine, Grabcad. https://grabcad.com/library/flotation-machine-1

Sawyer, B. 2002. Serious games: Improving public policy through game-based learning and simulation. USA: Woodrow Wilson International Center for Scholars.

Scales, M., 2019. VIRTUAL REALITY: MacLean offers VR training for bolter operators.Canadian Mining Journal (CIM). May 9. https://www.canadianminingjournal.com/news/virtual-reality-maclean-offers-vr-training-for-bolter-operators/

Shih, Y.-C. (2015). A virtual walk through London: culture learning through a cultural immersion experience. *Computer Assisted Language Learning*, *28*(5), 407–428.

Siewiorek, A., Gegenfurtner, A., Lainema, T., Saarinen, E., & Lehtinen, E. (2013). The effects of computer-simulation game training on participants' opinions on leadership styles. British Journal of Educational Technology, 44(6), 1012–1035.

Sketchfab Inc. (2024). https://sketchfab.com

Smallman, H. S., and St John, M. (2005). Naive realism: misplaced faith in realistic displays. Ergon. Design Q. Hum. Fact. Appl. 13, 6–13. DOI: 10.1177/106480460501300303

Sony Morpheus, 2024. Playstation VR. https://www.playstation.com/en-us/ps-vr/

SPH Engineering, Riga, Latvia (2022). Retrieved from https://www.ugcs.com/news-entry/announcing-release-of-ugcs-update-with-added-search-patterns-for-sar-operations

Squelch, A. (2001). Virtual reality for mine safety training in South Africa. Journal of the Southern African Institute of Mining and Metallurgy, 209-216.

Stothard, P.M., Squelch, A.P., Van Wyk, E.A., Schofield, D., Fowle, K., Caris, C., Kizil, M., & Schmid, M. (2008). Taxonomy of Interactive Computer-based Visualisation Systems and Content for the Mining Industry - Part 1. Proceedings of the AUSIMM Future Mining Conference 2008, Sydney.

<u>Simşek, İ. ve Can, T., 2019.</u> Yüksek Öğretimde Sanal Gerçeklik Kullanımı ile İlgili Yapılan Araştırmalara Yönelik İçerik Analizi, Folklor/edebiyat, cilt:25, sayı: 97-1, 2019/1.

Tan, B., Zhu, H., Wang, N., Jiang, Y., & Jia, G. (2012). Application of virtual reality in fire teaching of mining. ICCSE 2012 - Proceedings of 2012 7th International Conference on Computer Science and Education, Iccse, 1079–1081. https://doi.org/10.1109/ICCSE.2012.6295250

Tichon, J. ve Burgess-Limerick, B., 2021. A review of virtual reality as a medium for safety related training in the minerals industry, *Journal of Health & Safety Research & Practice*, vol. 1, no. 3, pp. 33–40.

Tiwari, S. R., Nafees, L., & Krishnan, O. (2014). Simulation as a pedagogical tool: Measurement of impact on perceived effective learning. The International Journal of Management Education, 12(3), 260–270.

Toraño, J., Diego, I., Menéndez, M., & Gent, M. (2008). A finite element method (FEM) - Fuzzy logic (Soft Computing) - virtual reality model approach in a coalface longwall mining simulation. Automation in Construction, 17(4), 413–424. https://doi.org/10.1016/j.autcon.2007.07.001

Unity 3D (2024). Unity Technologies Inc. https://unity.com

Unreal Engine (2022). Epic Games Inc. <u>https://www.unrealengine.com/en-US/</u>

Valsev, (2024). Realistic Underground Cave, Sketchfab, <u>https://sketchfab.com/3d-models/realistic-underground-basecave-40-46466ec0558945e9aac9dad15aaeb9f3</u>

Van Krevelen, D.W.F. & Poelman, R. (2010). A Survey of Augmented Reality Technologies, Applications and Limitations. The International Journal of Virtual Reality, 9(2):1-20.

Vangorp, P., Laurijssen, J., and Dutré, P. (2007). The influence of shape on the perception of material reflectance. ACM Trans. Graphics 26:77. DOI: 10.1145/1239451.1239528

Vuforia, PTC Inc., (2024). https://www.ptc.com/en/products/vuforia

Wang, D., He, L., & Dou, K. (2013). StoryCube: supporting children's storytelling with a tangible tool. The Journal of Supercomputing. DOI: 10.1007/s11227-012-0855.

Wang, X., Kim, M.J., Love, P.E.D., and Kang, S.C. (2013). Augmented Reality in built environment: Classification and implications for future research. Automation in Construction, 32, 1–13.

Warehouse, (2024). https://3dwarehouse.sketchup.com

Wikipedia, 2024a. https://tr.wikipedia.org/wiki/Stereoskop

Wikipedia, 2024b. <u>https://en.wikipedia.org/wiki/Google\_Cardboard#/media/File:Google-Cardboard.jpg.</u>

Wojciechowski, R., Walczak, K., White, M., & Cellary, W. (2004). Building virtual and augmented reality museum exhibitions. Proceedings of 9th international conference on 3D web technology (Web3D 2004), 135-144.

Xie, J., Li, S., & Wang, X. (2022). A digital smart product service system and a case study of the miningindustry:MSPSS.AdvancedEngineeringInformatics,53(216),https://doi.org/10.1016/j.aei.2022.101694

Xie, J., Liu, S., & Wang, X. (2022). Framework for a closed-loop cooperative human Cyber-Physical System for the mining industry driven by VR and AR: MHCPS. Computers and Industrial Engineering, 168(February), 108050. <u>https://doi.org/10.1016/j.cie.2022.108050</u>

Yılmaz, M.R., Göktaş, Y. (2018). Using Augmented Reality Technology in Education. Journal of Çukurova University Education Faculty, 47(2), 510-537.

Yin, C., Song, Y., Tabata, Y., Ogata, H., & Hwang, G. J. (2013). Developing and implementing a framework of participatory simulation for mobile learning using scaffolding. Educational Technology & Society, 16(2), 137–150.

Zhang, H. (2017). Head-mounted display-based intuitive virtual reality training system for the mining industry. International Journal of Mining Science and Technology, 27(4), 717–722. https://doi.org/10.1016/j.ijmst.2017.05.005

Zhang, X., Wang, A., & Li, J. (2011). Design and application of virtual reality system in fully mechanized mining face. Procedia Engineering, 26, 2165–2172. <u>https://doi.org/10.1016/j.proeng.2011.11.2421</u>

Zhou, Z., Cheok, A.D., Pan, J., & Li, Y. (2004). Magic story cube: An interactive tangible interface for storytelling. Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology, 364-365. DOI: 10.1145/1067343.1067404.

# **AFTERWORD**

We have entered a big world with this book, which has a tutorial-based design for beginners of Unity. After combination of the knowledge of computer graphics, animation, C# codes, it is possible to deploy Unity projects to computers, mobile devices, consoles, VR headsets and AR smart glass platforms.

Being an expert in this huge volume of work and interdisciplinary work with field experts enables us to obtain satisfactory and efficient results. However, it is also possible to get results without the opportunity to work together with Unity experts with knowledge at a level that can meet the needs. **"Introduction to Unity, a Real-Time Development Engine** *with applications for mining*" serves to this idea.

Unity information will be further deepened with mathematical computations, numerical analysis, optimization, statistics, VR headsets such as Meta Oculus, HTC Vive and AR smart glasses such as MS Hololens 2, Apple Pro Vision. In addition, multi-user applications, integration of artificial intelligence programs, and its use with IoT Internet of Things are included in the Unity's future perspective.

If you've gotten this far, thank you. Good luck and wish you success.