INTRODUCTION TO UNITY A REAL-TIME DEVELOPMENT ENGINE

With Applications for Mining

Kaan Erarslan



INTRODUCTION TO UNITY A REAL-TIME DEVELOPMENT ENGINE

With Applications for Mining

1st 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.

This book includes basic information about Unity, a real-time development engine, and its applications in mining engineering. 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
Acknowledgments
Files in Share
INTRODUCTION
1. INSTALLATION OF NECESSARY SOFTWARE
1.1.Unity 3D installation
1.2.Java JDK
1.3.Android Studio
1.4.Xcode for IOS Systems
1.5.Vuforia
2. UNITY PANELS
2.1.Using Unity 3D Panels
2.2.Game Tab/Mod
3. SCENE DESIGN
3.1.Adding a GameObject
3.2.Object Properties
3.3.Materials
3.4.Lights
3.4.1 Point Light
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# SCRIPT CODES FOR UNITY
4.1.Basic C# Information
4.2.C# SCRIPTS IN UNITY
4.3.Console Messages
4.4.Variables
4.5.Methods/Functions
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 st Method	158
6.7.2. 2 nd 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	184
9. FPS VE TPS APPLICATIONS	189
9.1.FPS-First Person Shooter and RPG-Role Playing Game	189
9.2.TPS Third Person Shooter-Starter	194
9.3.Third Person Character Controller – Armature Change	199
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	245
11.4.Getting the License Key and Downloading Database from Vuforia to Unity	250
11.5.AR-Vuforia and Multi-Image Target	263
11.6.Deployment into Mobile Devices	287
11.7.Augmented Reality with Video Player	292
11.8.Vuforia AR – Ground Plane	294
11.9.Similar Examples for Mining	307
12. USE OF ANIMATION AND ADOBE MIXAMO CHARACTERS IN UNITY	309
12.1.Relevant C# Script Codes	329
BIBLIOGRAPHY	
AFTERWORD	341

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 will be required for the course to be installed depending on the applications is listed below:

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. Download the Unity Hub	3
Follow the instructions onscreen for guidance	3
through the installation process and setup.	1
Download for Windows	i i
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 installing by selecting the 2022.x LTS version.

Unity Hub 3.8.0		- 🗆 X
¢ * 🔊	Installs	Install Editor
Projects	All Official releases Pre-releases Q Search	
 Learn Community 	Unity (2023.2.4f1) C:\Program Files\Unity\Hub\Editor\2023.2.4f1\Editor\Unity.exe Android UWP WebGL Windows	¢
	Unity (2022.3.16f1) Lts C:\Program Files\Unity\Hub\Editor\2022.3.16f1\Editor\Unity.exe Android UWP WebGL Windows	٥
🛓 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 Editor X					
Official releases	Pre-releases	Archive			
Can't find the ver and patch release	rsion you're looking es, or join our beta	for? Visit our download arch program releases.	for access to Long-Term Support		
🛛 Beta program	webpage				

Products S	olutions Resources Community Learn Su	apport:		PLANS AND PRICING Q I KE
		Unity download a	rchive	
To learn more about the release versions, visit <u>Unity Releases</u> . Unity 6 2023 2022 2021 2020 2019 2018 2017 Unity 5 All versions <u>LTS (Default)</u> Tech Stream				
Version	Release date	Rolease notes	Hub Installation	Downloads
2022.3.40f1	30 Tem 2024	Read		See all
2022.3.39f1	23 Tem 2024	Read		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/	technologies/downloads/#jd	lk17-windows	₿☆ Q	. Arama	♥ * ♥
ORACLE Q Prod	ucts Industries Resou	rces Support Events Develope	r Partners	(2) View Accounts	Contact Sales
Java downloads Tools and resources Java arc	hive				
	الله Lookin	g for other Java downloads?	OpenJDK Early Access Build	s JRE for Consumers	
Java 17 available now					
Java 17 LTS is the latest long-term support release for redistribute, at no cost, under the Oracle No-Fee Term	the Java SE platform. JDK is and Conditions.	17 binaries are free to use in production	and free to	out Java SE Subscription	
JDK 17 will receive updates under these terms, until a	t least September 2024.				
Java SE Development Kit 17.0.2 downlo	ads				
Thank you for downloading this release of the Java™ programming language.	Platform, Standard Edition	Development Kit (JDK M). The JDK is a c	levelop <mark>me</mark> nt environment for	bui <mark>ld</mark> ing applications and compo	nents using the Java
The JDK includes to <mark>ols for developing</mark> and testing pro	ograms written in the Java p	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.com/java	/17/latest/jdk-17_windows-x6	4_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: engine: developer portal	Home Downloads	Library Support	Pricing	Log In Register
SDK Samples Tools				
Release Version 10.25				
By downloading the Vuforia Engine SDF	K, Samples and Tools, you ag	ree to the developer agree	ement.	
V. f				
Use the Vuforia Engine SDK to build augm devices and digital eyewear. Vuforia Engin and Visual Studio.	nented reality Android, iOS, a ne can be used in projects bu	ind UWP applications for n uilt with Unity, Android Stud	nobile dio, Xcode,	
Download for Unity add/vuforia-package-10-254.unitypacka	oge	Download	1 SHA-256	
Download for Android vuloria-sdk-android-10-25-4 zip		Download	1 SHA-256	

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						3	- D	×
KE + O	Pro	jects	5			Add 👻 🚺	lew proje	ct
Projects					Q			
🖨 Instalis	*	010	NAME	CLOUD		EDITOR VERSION	N)	
🔊 Learn		ŝę	IMVR C:\A UNITY\IMVR		2 months ago	2022.3.16f1		
		60	GVR2 C:\A UNITY\GVR2	CONNECTED	4 months ago	2022.3.16f1		
		ŝę	gvr11 C:\A UNITY\gvr11	CONNECTED	4 months ago	2022.3.16f1		
		60	GVR C:\A UNITY\GVR	CONNECTED	4 months ago	2022.3.16f1		
		60	Area C:\A UNITY\Area	NOT CONNECTED	5 months ago	2023.2.4f1		
Downloads		60	Atay1 C:\A UNITY\Atay1	NOT CONNECTED	5 months ago	2022.3.16f1		
			ConicCrusher					

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		- <u> </u>
	New project Editor Version: 2022.3.16f1 LTS C	
All templates	Q. Search all tem; Q. Filter	
 Core Sample 	Core	•
Learning	3D (Built-In Render Pipeline) Core	3D (Built-In Render Pineline)
	Core	This is an empty 3D project that uses Unity's built-in renderer.
	Universal 3D Core	
	- Hiah 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 LTS 🗘	
All templates	Q Search all templates	3D (Built-In Render Pipeline) This is an empty 3D project that uses Unity's built-in renderer.
Sample	2D (Built-In Render Pipeline) Core	Read more
Learning	3D (Built-In Render Pipeline) Core	PROJECT SETTINGS Project name My project
	Core Universal 2D	Location C:\A UNITY
	Core O	Unity Organization KErarslan
	— Hiah 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⁽²⁾. 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**³.

In the section titled **Project** (4), 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** (5) 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[®]. 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[®]. Below is the empty audio editor, **Audio Listener**. Below that is the **Add Component** button, which adds various properties to the selected object.



File	Edit Assets Ga	ameObject	Component	W
	New Scene		Ctrl+N	
	Open Scene		Ctrl+O	É.,
	Open Recent Scen	ne		2
	Save		Ctrl+S	
	Save As		Ctrl+Shift+S	
	Save As Scene Ter	mpiate		
	New Project			
	Open Project			
	Save Project			
	Build Settings		Ctri+Shift+B	
	Build And Run		Ctri+B	
	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.

t	Assets GameObject Com	ponent Window
1	Undo Selection Change	Ctrl+Z
22	Redo Delete Game Objects	Ctrl+Y
100	Select All	Ctrl+A
ġ	Deselect All	Shift+D
222	Select Children	Shift+C
2	Select Prefab Root	Ctrl+Shift+R
þ	Invert Selection	Ctrl+I
1	Cut	Ctri+X
ŝ	Сору	Ctrl+C
ŝ	Paste	Ctrl+V
ł.	Paste As Child	Ctrl+Shift+V
1	Duplicate	Ctri+D
ŝ	Rename	
	Delete	
a a	Frame Selected	F
j	Lock View to Selected	Shift+F
1	Find	Ctr1+F
	Play	Ctrl+P
Ì	Pause	Ctrl+Shift+P
1013	Step	Ctrl+Alt+P
	Sign in	
i.	Sign out	
10200	Selection	
1	Project Settings	
	Preferences	
ŝ	Shortcuts	
No.	Clear All PlayerPrefs	
4	Graphics Tier	
i i	Grid and Snap Settings	

ssets GameObject Component Window He	(p
Create 3	Folder
Show in Explorer Open Delete Rename Copy Path Alt+Ctrl+C	C# Script 2D Shader Testing Playables
Open Scene Additive View in Package Manager	Assembly Definition Assembly Definition Reference TextMeshPro
Import New Asset Import Package > Export Package Find References In Scene Select Dependencies	Scene Scene Template Scene Template From Scene Prefab Prefab Variant
Refresh Ctrl+R Reimport	Audio Mixer
Reimport All Extract From Prefab Run API Updater Update UXML Schema	Material Lens Flare Render Texture Lightmap Parameters Lighting Settings Custom Render Texture
	Animator Controller Animation Animator Override Controller Avatar Mask
	Timeline Signal
	Physic Material
	GUI Skin Custom Font UI Toolkit
	Legacy
ges	Develo

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	mpty	Ctrl+	Shift+N
Create Er	mpty Child	Alt+	Shift+N
Create Er	mpty Parent	Ctrl+	Shift+G
3D Object	ct		>
Effects			5
Light			>
Audio			>
Video			>
UI			>
Camera			
Center O	n Children		
Make Par	rent		
Clear Par	ent		
Set as fin	st sibling		Ctri++
Set as las	st sibling		CtrI+-
Move To	View	Ctr	1+Alt+F
Align Wit	th View	Ctrl-	-Shift+F
Align Vie	w to Selected		
Toggle A	ctive State	Alt+	Shift+A

Add Mesh Effects Physics Physics 2D Navigation	Ctrl+Shift+A
Mesh Effects Physics Physics 2D Navigation	: : :
Effects Physics Physics 2D Navigation	: : :
Physics Physics 2D Navigation	1
Physics 2D Navigation	3
Navigation	
Audio	3
Video	3
Rendering	3
Tilemap	
Layout	3
Playables	
Miscellaneous	
Scripts	3
UI	3
Event	3

dow Help	
Panels	>
Next Window	Ctrl+Tab
Previous Window	Ctrl+Shift+Tab
Layouts	5
Collaborate	
Plastic SCM	
Asset Store	
Package Manager	
Asset Management	3
TextMeshPro	2
General	3
Rendering	2
Animation	3
Audio	2
Sequencing	2
Analysis	2
AI	>
and an an an an an an	

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.

T Package Manager			1 🗆 ×
+ = Packages: My Assets = Sort: Name 4 •	Filters - Ciea	r Filters O	N 5
	1.2 + *	3D Free Modular Kit	
	1.0.4 ±		
	1.0 ±		
	1.8.2 ±	Version 1.2 - September 07, 2020	asset store
	1,3 🛓		Velante • Rublistier Support
	2.1.5 🛓	* WEBGL DEMO * (http://blitz3dfr.fr	ge.fr/)
	17 4	3D Free Modular Kit is a small modula	ir kitt free for commercial of
	2.1.0 ±	Money gamenappe.	W ==
	e _ 132 ±	Images & Videos	
	1.1 🛓		A CONTRACT OF A
	1,4.15 🛓		
	1.2 土		
	12 ±		
		View Images & videos on Asiat Store	
			Supported Unity Versions
		Size:13,2 MB (Number of files: 53)	2018.4.15 or higher
		Purchased Date	-
		Release Details	Constant Mark
			07, 2020 Mitting.
C 100 100 100 100 20 20 20			Developer

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 - SampleScene -	PC, Mac & Linux Star	ndalone - Unity	/ 2020.3.2	Of1 Pe	rsona	il* <d< th=""><th>X11</th></d<>	X11
File Edit Assets GameC	Object Component	Window He	lp				
👋 🗘 🖸 🗄	I 🛞 💥 🗖]Center 🌐 G	lobal	#5			
'≡ Hierarchy	a i	# Scene	😎 Gar	ne			
+ - All		Shaded	•	2D		10	\$
🔻 🕄 SampleScene'	* :						
⑦ Main Camera ⑦ Directional L	a light						

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.

Standalone	- Unity 2020.3.20f1 Pers	sonal* <dx11></dx11>						
ent Windo	w Help			F				
Center	Global 📅				► II ►			
: # Sc	ene 😎 Game							1
Displa	y 1 🗢 Full HD (1920)	x1080) 👻 Sca	ale 🔍		0.51x	Maximize On Play	Nute Audio	Stats Gizmos -
ors Gam	ie		: # Scene					
Display	1 👻 Full HD (1920x	(1080) 🝷 Sca	le Shaded		📍 🕪 🤹	- ø0 # - * =	🗸 Gizmos 🗸	Q+ All

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

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.

Cut Copy Paste Paste As Child		Cut Copy Peste Paste As Child		Cut Copy Paste Paste As Child		
Rename Duplicate Delete		Rename Duplicate Delete		Rename Duplicate Delete		
Select Children		Select Children		Select Children		1.00
Set as Default Parent		Set as Default Parent		Set as Default Parent		
Create Empty		Create Empty 3D Object		Create Empty		
Effects >	Particle System	Light 2	Directional Light	Effects	5	
Light	Particle System Force Field	Audio	Point Light	Light		and the second second
Audio /	Line	Video	Spotlight	Audio	2	Audio Source
		UI >	Area Light	Video	12	Audio Reverb Zone
Camera		Camera	Light Probe Group	Camera	1	

-Effects-

-Light-



-Video-

-UI-User Interface -

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.

Projet - SampleScene - PC Mar. & Linux S										
File Edit Assets GameObject Component	nt Window He	(p								
🖤 🖬 Ø 🗷 🖽 🛠	Etenter 60	liobal 🐻					题	📥 Account 🖛 Layers	- Loy	
등 Hierarchy 경		on Geme					Inspector			
+• •	Staded	▼ 10	· · · · · · · · · · ·		X III + Gzinos +	NAL.	Asie	ta (Default Asset)		
• 4 Sampkilcen*				0	Ciente Solow in Explore Operi Dietin Remanne Capy Nath Open Sonne Additive Vere in Bickage Monager Import New Aster Import NexAge Export Reckage Export Reckage	Faider C & Solpt 2D Shader Lenna Playables Alt+CrifeC Astamby Definition R Soles Playables Astamby Definition R Soles Playables Soles Playables Soles Playables Soles Playables Playables Soles Playables Soles Playables Playables Soles Playables Soles Playables Playables Soles Playables Soles Playables Soles Playables Playables Soles Playables Soles Playables Playables Soles Playables Playables Soles Playables Soles Playables Playables Soles Playables Playables Soles Playables Soles Playables Playables Soles Playables Soles Playables Playables Soles Playables Soles Play		2	Open	
	/					Select Dependencies Refresh Reimport Reimport All Listract From Prefab	Ctil+R	Audio Mose Material Less Flano Render Testure Lightmap Parameters Lightmap Parameters		
Be Project E Console						Run API Updatter.		Custom Render Texture		
+*						Opdate Uxint Schema		Animator Controller		
C. All Motorials C. All Motorials C. All Prefats	n	Scenes.				Properties_	Alt+P	Animation Animator Override Controller Avatar Mask		
a Assets In Sceles								Timeline Signal		
P in Packages								Physic Material		
						Assoc Lal	GUI Skin Custom Font UI Toolkit	5		
								Legecy		
	1	Assets					AssetBunc	Brush		Nana 👻
A Visual Studio Editor Package version 2	d 14 is evaluable	e, we strongly	encourage you to update from t	the Unity Package Man	ager for a better Visual Studio	o integration		Terrain Layer		0 2 2 4

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.

Projel - SampleScene - R., Mar. & Linux St.	enthalizes - Using 20203-2011 Remonal* - DX11 -						- 0	F 18.
File Edit Assets GameObject Component	t Window Help							
🖤 🐨 🖉 🖾 🖽 🗶 🛛	🛙 Center 🌐 Diobal 🛛 🛅				f2) 📥 Acci	bant 🔹 Layers	- Loyout	
IE Hierarchy & I	III Scene Gene				O Inspector			
+* ***	Staded - 20 - 4 & - 5	50 E -	🛠 🖬 + Gizmos (+ 1974)					071
↑ * 4 Miles Paraple Bicenes* ○ Main Cannets ○ Diate Charles ○ Plane ○ Cultures ○ Capsule			Color		Renk T (Marretal) Bruder Standard Rendering Mode Main Mape o Albeda o Metalik Smonthana Source o Normal Map o Occubion o Octail Mask Emispion Taing Offset Secondary Mape o Octail Mesoc x2 o Normal Map	Opeque Matalic Apria × 1 × 0	* 1 * 0	0 7 1 1 Edit.
	1			RGB 0-255+		× Q	YO	
Project Console			18 M	255 241	Forward Rendering Oct	ions		
+=			0	0 200	SpecialerHighlights	2		
 Ravorites All Materials All Modelle All Profates 	Assets		B A Second	0 255 FF0000	Reflections Advanced Options Ronder Queue	⊊ ≢tom	Shador	990 - T
· av Assets			= Swatches					
Ma Schnes • Ma Packages								
					Anterio ante		222.4	
	Assets/renk1.mot							
Visual Studio Earch Pack age version 2.0	in a is evaluable, we encourage you to	update troot the unity Package Manager fo	ar a penerovisual studio integration				* *	

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**.



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**.



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 k**aplama1** 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.

Projet - SampleScene - PC Mac & L							
File Edit Assets GameObject Com	nponent Window Help						
🖤 🔣 Ø 🗷 🖽 😫	🛠 🗐 Center 🌐 Diobal 🔚				🕸 📥 Aci	count 🔹 Layers 🔹 🗌	.oyosit 📼
THerarchy 3 i ++ 5 m	on-Geme ≣ Boene Shaded + 20 ♥ 41 ct + d5-0 1		X III + Gizmas + N III		0 Inspector		3.1. Sublement
A an Campbelliceret I Main Campbelliceret I Main Campbelliceret I S Parte Codemon					Area Lipet Tage Untragged Tage Untragged Tage Untragged A. Transform Powlion Render Coder Rende Coder Mode	Layer Default Layer Default X -0.06 V 1.32 X -0 V 0 X 1 V 1 Point S5 Soft Shadows G Soft Shadows G Norle (Fastan) Norle (Fastan)	Static- 9 2 1 7 -0.21 2 0 2 1 9 2 1 - - - - - - - - - - - - -
Proyect E Console ** ** * Favorites C, All Materials C, All Materials *	Asels Image: Asels crone: Aspan. vert			3.1 3 # + 20	Render Abde Cuttor Mask Market in the second second second Market in the second second second second Uption is the second	Autor Everything Control not head one come Sopre with the updated in th end op in . Add Component	ی Any tites get ا
A Visual Studio Editor Park and unt	sion 2.0.14 is evaluable, we strungly encourate you to	undate from the Unity Package Man	oper for a tester Visual Studio Internatio				



3.4.4 Reflection Probe

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

Projet - SampaScene - PC Mac & L													
File Edit Assets GameObject Com	ponent Window	Help											
* • • • • • • • • • • • • • • • • • • •	K Etenter	Ciobal							\$Q 4	Account -			
T Hierarchy 3 1	Geme a	E Boene							- O Inspector				
+* * W			······································	Re l		大			Ca T Bettartie	n Broha			
1 SampleScene*	-								Tag Historia	1			
St Discritorial Loart									THE MEHILIER		CASEL DRIVE		
Triane									Transform			e :	
Cutte									Position	X 0.07		7 -0.23	
Capsule									Rotation	<u>8</u> e	<u> </u>	- 18º	
E Heflection Probe									Scale	× 1	3(1	Z 1.	
									🗉 😂 🖌 Reflection	Probe			
									Wide State Office				
									TIMA	TEAM			-
									Tetrosh Mode	CD Awa	180		
									Time Slicing	Individ	al faces		
				-	and a second				Runtime Settings				
					1000				Importance				
					- C 1				New Designation				
				a la sub						× 5		25	
									Box Offset				
									T Cunemap Captor	e Settings			
and the second second second													
In Project El Console								-	HOR				
								* * * *	RV Shadow Distan	a g			
Q All Materials		APPARTS							Clear Hags	SKYDOX			
Q. Alt Models		- -								_			
L, Alt Pretabs				110002					Raflection Pi	•			1 - 4 - 1
- Ausota											-		
Discenses													
- m recoges													
										1			
											· /		
											/		
A Visuel Studio Editor Package versi	ion 210 14 is eval	dable, we st	rongly encourage you to	update front the	Unity Package Ma	stager for a better Vi	sual Studio integration					* 3 3	8 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.

d Projet - SampleScene - PC Mar & L						
File Edit Assets GameObject Com	iponent Window Help					
🖤 🔛 Ø 🖾 🖽 😫	Center @Discal 15			🕸 📥 Acc	pant 🖛 Layers 📼 1	
THerarchy 3 i	ono Gemen E Boenne Shaden + 20 ♥ 40 at + d0/0 57 +	X #1 + Gizmas + + M		0 Inspector		
V 🗳 SampleScene* 🔰				The light probe unout	The second second	
C Man Carrena C Directional Cont C Plane C Cube C Schere				Contransform Position	× 0.02 ¥ 4.78	● 2 1 Z -0 23
Capsule			22.44eg00	Scale	X 2 X 2	2 2
C) Light Hobe Ordep				😸 🛩 Light Probe Group		
			_			
	Ç.					
				Talo Halo		Ø ₹ 1
					3	(
Project El Console			3.1			ļ ,
+			ê 🕈 🕈 🖬 🕬 🤉			
 Favorites All Materials All Models All Prefails 	Assets					
- Ausota	bearies kaptam., rentr lastanat					
 In Scenes In Packages 						
Vision Statistic English Parkage we'r	um 2.0.1.2 is evaluable, we structure encourage you to undate in	in the Unity Package Manager for a better Visial Studio lotera	odiciti			



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 mass like a solid object, it is expected to be affected by gravity and fall towards the ground accordingly or rise 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.

🚽 Projet – SampreScene - PC, Mar. & La	nux Standalcina -	Anity 2020 3 2	Off Besidnal* «DX11»									a x
File Edit Assets GameObject Comp	ponent Window	/ Help										
🖉 🖬 🗘 🖾 🖽 🐉	Center	Ciobal							80 📥 Ad	count 🖛 Laye	m = La	iyout 🔫
T fileranchy 3 1 ++ 5 11 * 6 SampleScenot 1 17 Main Camera	Geme Shaded	E Boene + 20	হ ৰাৱান কণ	174 -	- Abr	文 🖬 + Giri	nas 🖅 🗤 w	ء ا	B Inspector Main Camera Tag MainCamera		ver Detaut	eð 1: Statie⊷
Constitunal Light Constitute Const Const Constitute Constitute Constitute Constitute					-7~	K Transform Position Rolation Scale	X 8.78 X 9.961 X 1	V 3.94 V -90 V 1	• 2 1 2 -0.09 2 0 2 1			
									■ ✓ Camera Clear Flags Beckground Culling Mask	Skybox Everything		071 7 7
					P				Projection FOV Axis Field of View Physical Camera	Perspective Vertical		60
							* Main Camera			Noar 0.3 Far 1000 X 0 W 1	V ő H T	
									Depth Rendering Path Torget Texture Occlusion Culling	-1 Use Graphics : Nore (Render	Settings Texture)	•
+ • Favoritos C. All Misterials C. All Models		Assets		197224				\$ P ± #9	HDR MSAA Allow Dynamic Resolutio	Use Graphics 1 Use Graphics 1 n	Settings Settings	•
C, Alt Pretabs				CONTRACTOR OF					Target Display	Display 1		*
Assots									R Audio Listener			1 v v 1
• In Packages										Add Component		
A Visual Studio Editor Package vers	0 2 0 14 is ev	Mable, Weist	ongly energinage you	to update from t	ne Unity Package M	anager for a better Vi	sual Studio Integration	-0				

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.

Proje1 - SampleScene - PC, Mac B.	Ensur Standalouse - Unity 20203-2011 Personal* + DX11 +				6 ×
File Edit Assets GameObject Cor	nponent Window Help				
● ● Ø 図 目 感	🛠 🗐 Center 🖶 Diobal 🛛 👸		(Q)	🐣 Account 🖛 Layers 📼	
TE Hierarchy 3-1 ++ Not	onsGeme ≣Boene Shaded + 20 ¶ 41 at + at60 by	* X BI + Gizmag * V H	i O Inspector		
** Vill Man Cannes © Diectonal Light © Pare © Dide © Splare © Splare © Capsile			Tag Uning Tag Uning Salar Postion Relation Salar Education Education E	eel • Cayer Defect m X 0 V - X 0 V 0 X 1 V 1 (Mesh Filter) ndereer De De	(2) Statle+ (2) S
N Present III Danishie			Contribute Git Procese State Probas Light Phobes Light Phobes Anator Over Additional Sett Additional Sett Dynamic Git Dynamic Git Dynamic Git	sball filemin I munitat Light Proces Nors Bitend Proces Bitend Proces Bitend Proces Bitend Proces Nors Cransform) No Saarch	•
+*			🔹 🥐 📹 👷9 Components that y	Disidenti	cannot be
Forcerits GAU Motorian GAU Motorian GAU Motorian GAU An Products An American An American Bat Scenes Processes Processes	Assets In Internet Superior State		multi-edited.	Singlichonly 2D E Creat Trigget E Creat Trigget E Oriet a yout Croup E Oriet a yout Croup E Oriet a yout Croup E Spring Joint 2D E Spring Joint 2D E Spring Adak E Spring Adak E Spring E Rondear E Spring E Rondear	
A Visual Studio Editor Parkage per	sion 2.0.14 is evaluable, we structly encourage you to u	ordete from the Unity Package Menager for a better Visual Studio inter	and ou	TMP_Spate Animator	



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

In order 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.

🖬 Perjet - Samuelicene - D', Mar G						
File Edit Assets GameObject Co	mponent Window Help					
- 中の回は市	* Pivot Skoal B			100 🛋 A	coount · Layers ·	
The Herarstry A	L 🕶 Game 🕂 Scenn			@ inspector		
4 - A M	Shaded - 70 🕈 🗣 😹 - 🕫 🖓	× •	- Giamos, -: A All	CA 2 Particle System		
SampleScene*	+- <u></u>			Ten Untenand	 Lover Default 	
() Directional Light		ALC: NOT THE OWNER.	and the second se			
() Park		and the second s		Transform		
Seattle System				Position	A -4,924018 Y -0.37	20/12
Suger House (APAN)				Boale		- 11 - 11 - 11 - 11 - 11 - 11 - 11 - 1
		The second second				
				Particle System		
						Open Editoria
				Famile System		
		and a second	(All the second s	Primaria		
			Particip Effect			
			Contract of the second se	and an analysis	2	
			Proyestick upters 1.00			
			Factoria 50	10 Start Hetates		
			Speed Fange 5.0 - 5.0	BEART HIStation		
			Simulate Loyers - Notreng -	ThirtColor		14
			🖌 Resetutete	Drawny Meditier	2	
			Show Bounce	Linuarkon üpeles	1.	
			anow deny periods	Outin Time	distint	
Re Project El Console				Hity On Analast	Some	
1			🕴 🛷 👘 🕬	THE HE WEEKING	agitteey	
C. All Materials				Auto Natsion Send	*	
O, Alt Models						
CL, All Prefabre				Particle System Curves		
- Assets	Natana Scones kaplaw Is	esti testarel actava-				
Moterials						
Packages						
A Misual Brudin Editor Package ve	rsian 2.0,14 is available, we strongly knowinge you to up	idate from the Unity Pickage Manager for a to	itter Vicual Statin Integration			

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.

	ô 🍝	Account 👻 Layers	✓ Layout
	Inspector		a
rmos 👻 🔤 All	Particle Sy	stem	Static -
· · · · ·	Tag Untagged	✓ Layer Det	fault 🔫
	🔻 🙏 🛛 Transform		0 : 1
	Position	X -4.924018 Y -0.3	37 Z 0.13
×	Rotation	X -90 Y 0	Zo
< Persp		X 1 Y 1	Z 1
and the second se	🔻 🐮 🛛 Particle Syst	tem	0 ‡ :
			Open Editor
	Particle System		
Particle Effect Pause Restart Stop Playback Speed 1.00	Duration Looping Prewarm Start Delay Start Lifetime Start Speed	5 0 5 307	
Playback Time 232.20	X 1 30 Start Potation	Υ] Z]	-
Pantoles 50 Speed Range 3.1 - 3.1 Simulate Layers Nothing ▼ ♥ Resimulate Show Bounds Show Only Selected	X o Flip Rotation Start Color Gravity Modifier Simulation Space Simulation Space Deta Time Scaling Mode Play On Awake* Emitter Velocity	Y O Z O O Local Scoled Local Rialdbody	*
	Max Particles Auto Random Seed Stop Action Communication Particle System Curv	1000 V None	•
	, station and		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.
	i 🚯 Inspector			
	Start Color		-	
	Gravity Modifier	0		
1	Simulation Space	Local		
-N	Simulation Speed	1		
1	z Delta Time	Scaled		
A	Scaling Mode	Local		
	Play On Awake*	~		
rso	Emitter Velocity	Rigidbody		
	Max Particles	1000		
	Auto Random Seed	<u> </u>		
	Stop Action	None		
	Ping Buffer Morte	Display		
	Emission	Disabled		
	Rate over Time	10		
	Rate over Distance			
	BUISTS			
	Time Count	Cycles Interval Probability		
	0.000 30	1 0.010 1.00		
		1+		
	V Shape			
	Shape	Cone		
	Angle	25		
	Radius	1.0		
	Radius Thickness			
	Arc	360		
	Mode	Random		
	Spread	0		
	Length			
	Emit frame	Base		
	f Texture	III None (Texture 2D)	٥	
9	f Texture Position	None (Texture 2D) X 0 Y 0 Z 0	0	
9	t Texture 69 Position Rotation	X 0 Y 0 Z 0 X 0 Y 0 Z 0 X 0 Y 0 Z 0	0	
	t Texture 69 Position Rotation Scale	X 0 Y 0 2 0 X 0 Y 0 2 0 X 0 Y 0 2 0 X 1 Y 1 2 1	0	
- 9	Exercision Section Scale Particle System Curve	X 0 Y 0 Z 0 X 0 Y 0 Z 0 Z 0 X 0 Y 0 Z 0 Z 0 X 1 Y 1 Z 1	0	

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 **<u>Freepik</u>** 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		a	Inspector			a :
Alev (Material)		07:	Alev (Ma			0 i i
Shader Standard		👻 Edit	Shader S	Standard	-	Edit
Rendering Mode	Opaque	*	Rendering Mr	۹ Shaders		
Albedo O Metallic Smoothness Source O Normal Map O Height Map O Colusion O Detail Mask Emission O Color Global Illumination Tiling Offset	Metallic Alpha HDR X None X 1 X 0	0 0.5 * 1 0	Albedo O Albedo O Metallic Smoothi Sourc O Normal I O Height N O Occlusik O Detail M Emission Global II Tilling Offset	Mobile Nature UI VR FX GUI Particles Skybox Sprites Unlit Autodesk Interactive Standard Standard (Specular setup) Legacy Shaders X 0	Y 0	0.5
Secondary Maps © Detail Albedo x2 © Normal Map Tiling Offset UV Set Forward Rendering Optic Specular Highlights Deflections	X 1 Y X 0 Y UV0	1 1 0 ~	Secondary Ma • Detail Alt • Normal M Tiling Offset UV Set Forward Rend Specular Highl	aps bedox2 lap X 1 X 0 UV0 lering Options lights	Y 1 Y 0	1
alev		▶ ≘ % ≥ -	alev		• •	8 🔵 🔻

Alev (Material) Alev (Material) Alev (Material) Alev (Material) Cabada Ca	B Incodet				A 1				
Sundar: Sandar: Adv (Material) Sundar: Sandar: Capacy Shadar: Each	Inspector		2		Inspector				
Shadari Sandari Man Mays Legscy Shaders Bumpd Of/Ludo Legscy Shaders O Abardi Bumpd Of/Ludo O Abardi Dempid Secular O Abardi Diffuse Fast O Abardi Bardiari Diffuse O Abardi Diffuse Fast O Codari Secular O Abardi Abardiari Diffuse O Abardi Bardiari Diffuse O Abardi Abardiari Diffuse O Abardi Bardiari Diffuse O Abardi Bardiari Diffuse O Abardi Bardiari Diffuse O Abardi Diffuse O Abardi Bardiari Diffuse	Alev (Materia	an	•	7 - 1- 1	Alev (Mate	erial)			011
Rendering M Liggery Shadars Main Mags Burnped Offlue Smooth Diffuse 0.5 Socie Diffuse East Octavity Decal Octavity Decal Octavity Decal Socie Althor Monte Endedding Octavity Decal Socie Althor Monte Endedding Octavity Decal Socie Althor Monte Endedding Octavity Decal Socie Nomel Maps Ottavity Volutivity Octavity Nomel Maps Ottavity Volutivity Socie Volutivity Ottavity Volutivity Octavity Volutivity Octavity Volutivity Octavity Volutivity Octavity Volutivity Ottavity	Shader Stan	dard	• E	Edit	Shader St	tandard			Edit
Main Maps Legsey Shaders I o Athordo Bumped O'Investore Bumped Specular Diffuse O Motalid Diffuse fast U Haynth Particles U Haynth	Rendering M			-	Rendering Mr	4			-
Cottada Burged Specular Decal Smooth Difuse Smooth Difuse Smooth Difuse Sourd Difuse Difuse Decal Cottada	Main Maps	Legacy Shaders			Main Maps		Particles		
O Metalli Bumped Specular O Smooth Diffuse O Smooth Diffuse Fast O O Headle Additive O O Headle Additive (Soft) O.S Smooth Diffuse Detail O.S O Headle Parallas Diffuse Detail O.S O Dockall M Parallas Diffuse O.S O Cockall Tansparenth O.S O Othell M Mado X2 O.S O.S O Othell M Mado X2 O.S O.S O Othell M Mado X2 O.S O.S O Morall Map Titing V.D V.D O Morall Map Titing V.D V.D O Market Mede Concontenteratinterat	a o Albedo Bu	mped Diffuse			Albedo	~Additive-Mul	tiply		
Smooth Difuse Bail Smooth Difuse Fast Difuse Fast Difu	O Metallic Do	mped Specular	o		OMetallic	Additive			0
Source Drifuse Datall Ordenal Lightmapped: Ordenal Lightmapped: Ordenal X Parallax Specular Ordenal X X 0 Y 0 Specular Highlights X 0 Y 0 UV Set UV Set Secondar Highlights Secondar Highlights Second	Smooth Dif	fuse	0.	5	Smooth	Additive (Soft)			0.5
O Nomal Diffuse Fast Undight Parallax Struce O Occlusi Parallax Struce O Detail More Secondary Maps O Detail More Y 0 U Set UV0 Forward Rendering Options Specular Highlights Paralex V 0 O Detail More Y 0 U Set UV0 Forward Rendering Options Specular Highlights Paralex V 0 O Detail More V 0 U Set UV0 Forward Rendering Options Specular Highlights Paralex V 0 O Detail More V 0 V Set UV0 Forward Rendering Options Secondary V 0 V Set V 0 <td>Sourc Dif</td> <td>ífuse Detail</td> <td></td> <td>- -</td> <td>Sourc</td> <td>Alpha Blended</td> <td>Premultiply</td> <td></td> <td>0.5</td>	Sourc Dif	ífuse Detail		- -	Sourc	Alpha Blended	Premultiply		0.5
Lighthyspeed Parallax Offsee Parallax Secular Parallax Secular		ffuse Fast			ONermal	Anim Alpha Ble	ended		
Parellas Declarity Parellas Definition Occularity Parellas Declarity Parellas Parella	E Heicht N	Jhtmapped			Olleistet	Multiply			
COUCUMAR Particles O Oberlai Nether two Emission Secondary Maps Color Secondary Maps Socondary Maps Color Color Y O UV Set UV OV Forward Rendering Options Specular Highlights Reflections alev Color Socondary Maps Color Color Color Specular Highlights Reflections Socondary	O Occlusio Pa	rallax Diffuse				Multiply (Doub	le)		
Obtain & Reflective Emission Secondary Maps Obtain & X Obtain & X Offset X Offset X Obtain & X Offset X Offset X Offset X Obtain & Maps Obtai	o Dotoil M Pa	rticles	2		Occlusie	VertexLit Blend	led		
Emission Global II Tiling Sadet Illumin Global II Tiling > Offset X 0 Y 0 Secondary Maps O Detail Albedo x2 O Normal Map Offset X 0 Y 0 Secondary Maps O Detail Albedo x2 O Normal Map Offset X 0 Y 0 UV Set UV 0 V 0 Voltest X 0 Y 0 Secondary Maps O Detail Albedo x2 O Normal Map Offset X 0 Y 0 UV Set UV 0 V 0 Voltest X 0 Y 0 Specular Highlights Reflections alev Image N 0 Image N 0 N 0	Re	flective							
Coloding Specular Global II Tansparent Tiling VartaxLit Offsat X 0 Secondary Maps Operatin Maps Onormal Map Image: Secondary Maps Operatin Maps Image: Secondary Maps Offsat X 0 UV Set UV 0 UV Set UV 0 Special Highlights Secondary Maps	Emission	lf-Illumin			Emission				
Global II Transparent Ting Offset VertaxLi Offset VortaxLi Offset VortaxLi Offset VortaxLi Offset VortaxLi Offset VortaxLi Offset VortaxLi Offset VortaxLi Offset VortaxLi Offset VortaxLi VortaxL	© Color Sp	ecular			© Color				
Tiling Offset X 0 Y 0 Secondary Maps Obtail Abedo X2 ONormal Map Tiling X 1 Y 1 Offset X 0 VV 3et UV 3et Specular Highlights Performance Medicing Options Specular Highlights Performa	Global II Tra	ansparent	2	T	Il IsdolÐ				•
Offset X 0 Y 0 Secondary Maps ODetail Abledo x2 ONormal Map 1 Tiling X 1 Y 1 Offset X 0 Y 0 UV Set UV 0 Forward Rendering Options Specular Highlights F Reflections alev Performed years with a second of the second of	Tiling ve	rtexLit			Tiling				
Secondary Maps Obtail Albedo x2 Ohomal Map Tiling X1 UV Set UV Set UV Set UV Set Reflections Specular Highlights Reflections alev Secondary Maps Obtail Albedo x2 Ohomal Map X1 V1 UV O Forward Rendering Options Specular Highlights Reflections Specular Highlights Reflections Specular Highlights Comment Maps Comment Specular Highlights Comment Specular Highlights Specular Highlights Comment Specular Highlights Comment Specular Highlights Comment Specular Highlights Specular Highlights Specul	Oriset	×o	Y D		Offset	x	2		
O Detail Albedo x2 O Normal Map I Illing X1 Y1 Offsat X0 Y0 V0	Secondary Maps				Secondary Mag	ps			
O Normal Map Tuing X1 Offset X0 VV Set UV Set UV O Forward Rendering Options Specular Highlights Reflections Reflections Set UV O Forward Rendering Options Specular Highlights Reflections Reflections Set UV O Forward Rendering Options Specular Highlights Reflections Reflections Set UV O Forward Rendering Options Specular Highlights Reflections Set UV O Forward Rendering Options Specular Highlights Reflections Specular Highlights Reflections Set UV O Forward Rendering Options Specular Highlights Reflections Specular Highlights Reflections Set UV O Forward Rendering Options Specular Highlights Reflections Specular Highlights Reflections Set UV O Forward Rendering Options Specular Highlights Reflections Specular Highlights Reflections Set UV O Forward Rendering Options Specular Highlights Reflections Specular Highlights Reflections Set UV O Forward Rendering Options Specular Highlights Reflections Set UV O Forward Rendering Options Specular Highlights Forward Rendering Options Specular Highlights Forward Rendering Options Specular Highlights Forward Rendering Options Forward Rendering Options Forward Rendering Options Forward Rendering Options Forward	⊙Detail Albedo	x2				edo x2			
Tilling X 1 Offset X 0 UV Set UV0 Forward Rendering Options Specular Highlights Reflections alev	Normal Map		1		© Normal Ma	ар			1
Offset X 0 V 0 UV Set UV 0 Forward Rendering Options Specular Highlights Reflections alev Performance With Highlights Performance With Highlights Per	Tiling) Y 1		Tiling	x	8	Y 1	
UV Set VV Set	Offset	× o	YO		Offset	x	5	YO	
Forward Rendering Options Specular Highlights Reflections alev Performer Reflections Performer Reflections Performer Reflections	UV Set	UV0			UV Set	U	/0		
Specular Highlights Reflections alex Specular Highlights Reflections alex Specular Highlights Reflections alex Specular Highlights Reflections alex Specular Highlights Reflections Specular Highlights Reflections Ref	Forward Renderin	g Options			Forward Rende	aring Options			
Reflections alev Reflections	Specular Highlight	is 🛃			Specular Highli	ghts 🗸			
alev	Reflections	~			Reflections	~			
Reprint Supporter Reprin Supporter Rep	alev		- > = 🥳	9 -	alev			▶ ●	8 🤤 🗝 🕴
Projet & Sequence K. Mod Allans Statutes Internativ(CDSS) 0 Table Sequence K. Mod Allans Statutes Internativ(CDSS) 0 Table Sequence K. Mod Allans Statutes Internativ(CDSS) 0 Statutes Cancelest Concernet Worker Internative Concernet Vorker Internative Concer							\sim		
Rept Bornel									
Projet Surgicity A loca is functional likely status John Hansand' status - - - The Status Cancebert Marcine -									
File 181 Assti GamChycki Umborki Umbor	📕 Projut - SampioScene - PC, Mar. & L	inor Standalone - Mixty 2020 3 2011 Personal® - Oct 1	8						- 6 3
Interactive 3) IS Server Server <td>File Edit Assets GameObject Com</td> <td>iponent Window Help</td> <td></td> <td>► 0 = 0</td> <td></td> <td></td> <td>a) 🔺</td> <td>Vacount - Lavers -</td> <td>Lovout</td>	File Edit Assets GameObject Com	iponent Window Help		► 0 = 0			a) 🔺	Vacount - Lavers -	Lovout
Called a considered by the considered by th	T filerarchy 3 I	fill Scienter dos Gerner					0 Inspector		a,
While Camera Strain Gramma In Provide System In Provide	++ GampleScene* (Shaded + 20 9 41 68 + 66	9.117	× •	I + Gizmas + v M		Particle Syste	m	
Still Bards	C Main Carrena C Directional Light					<u></u>	Tag Unsagged	Ciger Dene	
Provide system Riscov X = 40 Y 0 Z 1 Provide system Particle Effect Particle System P 2 P Provide System Particle Effect Particle System P P Provide System Particle System P	C Hane C Odunter					1		X -4 924018 ¥ -0 37	Z 0.13
Project Donole 0 Project Donole 0 D	annoe syviem					1.744	Rotation Scale	X -90 Y 0 X 1 Y 1	
Proved ID Console Proved ID Console Provide States Provide		-		1		_	👻 😵 Particle System		
Project Donole 1 2 <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td>Perficie Sedam</td><td></td><td>Open Editor.</td></t<>							Perficie Sedam		Open Editor.
Provide Particine Effect Particine ZS Show Days Show Days </td <td></td> <td></td> <td></td> <td></td> <td></td> <td>-</td> <td>Duration</td> <td>1</td> <td>0</td>						-	Duration	1	0
Player, Resort Stor Player, Speed 100 Player, Sp			1		Particle Effect		Looping Prewarm		
Project E Console +* Frontice Q All Materials Q All M					Pauca	Restart Stop	Start Limburg Start Spood		(
Protect II Console +* Frontes QA M Markeish QA M Markeish			PE		Playback Speed Playback Time	222.61	3D Start Size X 1 V		
C All Materials C					Particles Speed Rance		3D Diart Rotation X m / Y		
Brown Bounds Show Bou					Simulato Layoro	Notning -	Start Color Grant's Vectors		
And Andresian Assets Asset Assets					Show Bounds		Setulation Busics Setulation Reviet	Local 0.5	
Image: Contract Image: Contract Image: Contract Image: Contract Image: Contract Image: Contract Image	N Desert 18 Provide				Snow Druy Se	AND	Della fice Scaling Units	Scaled Local	
Q. Al Moderla Assets Assets Assets Assets Assets Assets Assets Asset and Ass	t.					₽ ₽ ± \$9	Entition Valocity Maa Partition	Reidbedy	
	Q. All Materials Q. All Models	Assets		internal AN			Auto Rendom Seed Glop Action	V None	

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.

Materia In Materia In Scenso In Packago



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# SCRIPT CODES 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 Hub 3.8.0						- 🗆 ×
· •	Projects	1		A	sd 🔹 🚺	vew project
O Projects				Q Se		
Installs		NAME	CLOUD	MODIFIED ^	EDITOR VERSIO	N
😁 Learn		MinPro1 D:\UNITY\MinPro1	CONNECTED	a few seconds ago	2022.3.14/1	، ا
		MineModels D:\UNITY\MineModels		an hour ago	2022.3.1411	
		UGM1 D:\UNITY\UGM1		7 days ago	2022.3.14f1	
		CSUnity1 D:\UNITY\HOLOLENS\CS		19 days ago	2022.3.14f1	
		FPSApp1 D:\UNITY\FPSApp1		19 days ago	2022.3.141	÷ •••
Downloads		AdobeMixamo D:\UNITY\AdobeMixamo	CONNECTED	21 days ago	2022.3.1411	
	xe	M_Mine1	(D. connectes)		0000 01124	A

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.

Ŧ	D:\Program Files\2023.2.2f1\Editor\Unity.exe		
	Android UWP WebGL Windows	Add modules	
		Show in Explorer	
ŵ	Unity (2022.3.14f1) LTS	Uninstall	2
	D:\Program Files\2022.3.14f1\Editor\Unity.exe		X
	Android UWP WebGL Windows		

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	External Tools		۹.
7 20 Physics ▼ Analysis Profiler Asset Pipeline	External Script Editor Generate .csproj filas for	Visual Studio 2022 [17.9.34607]	
Colors Diagnostics External Tools	Embedded packages Local packages Registry packages	Š	
Gi Gache Jobs Package Manager Scene View Scene Template Search Indexing Timeline	Git packages Built-in packages Local tarball Packages from unknown sources Player projects Ragenerate project files		
UI Scaling Visual Scripting	Image application	Open by file extension	
	Revision Control Diff/Merge Tool Path Two-way diff command line Three-way diff command line Merge arguments	Custom Tool	
	No supports	d VCS diff tools were found. Please set up a custom to surregeer Diffuerce	ool 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 C# file
                              void Update()
                              Ł
                                     // Update is called once per frame
                        }
```

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, <u>vectorial</u>, etc.

int a=10, b, c:	// integer
int[] dizi:	// 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>	// definition of list-type variable
tamsayi.Add(10);	// assignment of 10 to the first element of the array
tamsayi.RemoveAt(0);	// removing the first element of the array
float x, y, z;	// float number variables
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.

ſ	using System.Collections;
Inclusion of library	using System.Collections.Generic;
class baying the same	using UnityEngine;
class having the same	
name as the script	public class KameraKontrol : MonoBehaviour
externally accessible variable	public OyunKontrol oyunK;
externally increasible variable	private float hassasiyet = 5f;
externally maccessible variable	private float yumusaklik = 2f;
	Vector2 gecisPos;
	Vector2 camPos;
	GameObject oyuncu;
	<pre>// Start is called before the first frame update</pre>
Method that will run once at	<pre>void Start()</pre>
the beginning	
the beginning	oyuncu = transform.parent.gameObject;
	//camPos.x = 121;
	//cambos.y = 12r;
	1
	// Update is called once per frame
Method that will work during	<pre>void Update()</pre>
Wethou that will work during	
run time	if (oyunK.oyunAktif)
	Vector2 farePos = new Vector2(Input.GetAxis("Mouse X"), Input.GetAxis("Mouse Y"));
	<pre>farePos = Vector2.Scale(farePos, new Vector2(yumusaklik * hassasiyet, yumusaklik * hassasiyet));</pre>
	gecisPos.x = Mathf.Lero(gecisPos.x, farePos.x, lf / vumusaklik):
	gecisPos.v = Mathf.Lerp(gecisPos.v, farePos.v, 1f / yumusaklik):
	camPos += gecisPos;
	transform.localRotation = Quaternion.AngleAxis(-camPos.y, Vector3.right);
	oyuncu.transform.localRotation = Quaternion.AngleAxis(camPos.x, oyuncu.transform.up);
	ł
	if (Input.GetKev("escape"))
	Application.Quit();
	1
)

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.



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.



Ansagent o x			- 0	Selution Explorer
bly-CStarp	- CanadeMessage	- ReStert0	- +	ま る・= 5日日
1 -using System.Collection	ns;		÷-	Septh Sciutors Explanar (Cr
2 using System.Collection	ns.Generic;			TE Solution (CScape)) (Fa
3 using UnityEngine;				Assembly CSharp
4	and a Manager and and			
5 public class consolered	spade : Houppenarion.			
7 1 // Start is called	before the first frame undate			
8 word Start()	berere ene ritise riture apares			
9 (
10			1	
11 1				
12				
13 // Update is calle	d once per frame			
14 void Update()				
15				
16				
12 1				
18 3				
10				

A Distant Pro			1011 (Part) 100	
			• 1 3	
tput from:	· · · · · · · · · · · · · · · · · · ·			
				Real Provide Street

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



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



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.



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 created methods 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++; ≡ x=x+1; ≡ 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^y</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.

vusing 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); // Update is called once per frame 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 4
Assembly-CSharp
                                                              - % ConsoleMessage
                                                                                                                             - CaStart()
               using System.Collections;
         1
        2
              using System.Collections.Generic;
        3
              using UnityEngine;
        4
  OI
        5
               public class ConsoleMessage : MonoBehaviour
        6
               {
        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
       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 9
       24
       25
       26
            L
                        // Update is called once per frame
       27
                    void Update()
       28
                    {
       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) th="" {="" }<=""></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.





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); }</pre>
    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.

ConsoleMe	essage.cs	• x
assemb 🖉	oly-CSharp	
	1	using System.Collections;
	2	using System.Collections.Generic;
	З	using UnityEngine;
	4	
10	5	<pre>ypublic class ConsoleMessage : MonoBehaviour</pre>
	6	
	7	public int[] say;
	8	
	9	<pre>// Start is called before the first frame update</pre>
	10	void Start()
	11	E
	12	
	13	3
	14	
	15	// Update is called once per frame
	16	void Update()
	17	
	18	
	19	3
	20	



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.

0 Inspector			ĩ	9] :
🕤 🗹 myMessage			Stati	c 💌
Tag Untagged		Layer Default		•
🔻 🙏 🛛 Transform			0 5	ŧ (1
Position	X 0	YO	Z 0	Ĩ
Rotation	X 0	Y O	Z 0	1
Scale (Q				
🔻 # 🖌 Console Message (S	cript)		0 7	1 II.
Script	Consolek			0
Say			5	
Element 0	0			
Element 1	0			
Element 2	0			
Element 3	0			
Element 4	0			
			+ -	4

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.



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.



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.cs	a X		
日 Assembly-CSharp	+ 9g.Consolet	dessage	- 😤 Start()
4	//using UnityEngine.Random;		
Bt 6	public class ConsoleMessage : MonoBehaviour		
8	List <int> rakam;</int>		
10 11 12	<pre>// Start is called before the first frame void Start() {</pre>	update	
13 14	<pre>rakam= new List<int>();</int></pre>		
15 16 17 18	<pre>for (int i = 0; i < 5; i++) { rakam.Add(i*100); }</pre>		
20 21	<pre>foreach(int val in rakam) {</pre>		
22 ¥ 23 24	}		
25 26 27 28 29	<pre>// Update is called once per frame void Update() { </pre>		
31			



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.



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.c	s⇔×	
Assembl	ly-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	3

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)

xCode.cs -= ×		
Assembly-CSharp	- ⁴ gBoxCode	- RBOnCallisionStay(Collision collision)
1	using System.Collections;	
2	using System.corrections.generic;	
4	using sharyengane,	
81 5	public class BoxCode : MonoBehaviour	
6		
7	private void OnCollisionEnter(Collision collision)	
9	Debug.Log(" SIMDI CARPTI "):	
10		
11		
12	private void OnCollisionExit(Collision collision)	
13	L Debug Log(" CARPISMA BITTI ")	
14	bebugteby, envesion barra 7,	
16		
17	private void OnCollisionStay(Collision collision)	
18		
190	Debug.Log(" CARPISIYOR ");	

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									Ex	clude Lavers	Nothing	*	
Game		Display 1	▼ F	ree Aspect		Scale 🔵 –		Play Focuse					
1									- I 🖓	Rigidbody		0 ‡ :	
									Mass		1		
									Drag		0		
									Angu	lar Drag	0.05		
									Auto	matic Center Of I	N 🗸		
									Auto	matic Tensor	~		
									Use (Gravity	~		
									ls Kir	iematic			
	-	_	6	1	-		_	_	Interp	oolate	None		
	1	(million					-		Collis	sion Detection	Discrete		
									T Cons	traints			
									Fre	eeze Position			
									Fr	eeze Rotation	🗸 X 🗸 Y 🗹 Z		
									▼Laye	r Overrides			
									Inc	clude Layers	Nothing		
										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.





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.	cs 🖶 🗙		
Assem	bly-CSharp	→ 🦓 BoxCode -	10
	1	<pre>vusing System.Collections;</pre>	
	2	using System.Collections.Generic;	
	3	using UnityEngine;	
	4	using UnityEngine.UIElements;	
	5		
0	6	vpublic class BoxCode : MonoBehaviour	
	7		
	8	public GameObject kure;	
	9		
	10	private void Start()	
	11	· · · · · · · · · · · · · · · · · · ·	
	12 😰	<pre>GameObject newBalls = Instantiate(kure);</pre>	
	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**.

Assembly-CSharp			+ 🕞 Start(
	1	vusing System.Collections;	1.41
	2	using System.Collections.Generic;	
	3	using UnityEngine;	
	4	using UnityEngine.UIElements;	
	5		
10	6	<pre>>public class BoxCode : MonoBehaviour</pre>	
	7	1	
	8	public GameObject kure;	
	9	· · · · · · · · · · · · · · · · · · ·	
	10	private void Start()	
	11		
	128	GameObject newBalls = Instantiate(kure, new Vector3(-3f,6f,0f), Quaternion.identi	tv):
	13		
	14		



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.





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**.

≔ Hi€	erarchy	□:	# Scene Manimator
+*	Qr All	R	🔟 Center 🔻 🌐 Global 👻 📅 🔻 🛲 🖛 🖛 🖛
5	🛠 SampleScene*		
	分 Main Camera 分 Directional Light 分 Plane		
Г	⑦ Box ⑦ 3DOval ♥ ⑦ twinCylinder		(j) 図
	♀ 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.

Harakates -0 ×	10 million and	I a second					
g Assembly-CSharp	• M Harekat	. Ca⊔pdate()					
1	using System.Collections;						
2	using System. Collections.Generic;						
5	using unicyengine;						
-4 E	public class Harakat : ManaBahawiour						
6	I I						
7	Vector3 vec = Vector3.zero:						
8	vectors vec - vectors ready						
9	// Update is called once per frame						
10	void Update()						
11							
12	transform.eulerAngles = Vector3.Lerp(transform.eulerAngles,	vec, 0.1f);					
13							
14	if(Input.GetKeyDown(KeyCode.W))						
15	E .						
16	vec =new Vector3(0f, 0f, 0f);						
17							
18	if (Input.GetKeyDown(KeyCode.D))						
19							
20	vec = new Vector3(0f, 90f, 0f);						
21	3 AC (There is a structure of the state of the						
22	1+ (Input.GetKeyLown(KeyLode.5))						
23							
24	vec - new vectors(or, 1801, 01);						
25	if (Input GetKeyDown(KeyCode S))						
20	{						
280	vec = new Vector3(0f, 270f, 0f):						
29							
30	3						
	5 · · · · · · · · · · · · · · · · · · ·						

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(" Dev collided 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**.





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 E Console	E/ 🖬	9 Game	1
Clear Collapse Error Pause Editor	1969+ #LO 100 G	ame · Display I · Free Aspect	💌 Scele 🜒 — 1x - Play Focula
(15.43.24) Ray calified 3DOvat UnityEngine Debug Lag (object)			
(13:43:24) Ray calided 3DOval UnityEngine Debug: og (akject)			
13:43:24) Ray calided 3DOvel UnityEngine Debug Log (abject)			
(13:43:24) Ray calified 3DDval UnityEngine DebugL og (abject)			
(1) 13 43:24) Ray collided 300val UnityEngine Debug: og (object)			
	1		

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.

Accembly-CSham	- Or Base of	er - Palloditeli
1	using System Collections:	C - Standard
2	using System Collections Generic:	
4	using Unity Visual Scripting:	
4	using UnityEngine:	
5	using UnityEngine UTElements:	
6	and one yengeneroze cenercy,	
t 7	public class RayCollider : MonoBehaviour	
8	A sector and a sector and a sector and a sector	
a	BaycastHit obi:	
2		
10	mycuschie obj,	
10	, and the set of the s	
10 11 12	nrivate void Undate()	
10 11 12	<pre>private void Update() {</pre>	
10 11 12 13 14	<pre>private void Update() { Ray isinYay = GameObject.Find("Main Cameobject.Find("Main Cameobject.Find("Ma</pre>	era").GetComponent≺Camera>().ViewportPointToRay(new Vector3(0.5f. 0.5f. 0f)
10 11 12 13 14	private void Update() { Ray isinYay = GameObject.Find("Main Ca	era").GetComponent <camera>().ViewportPointToRay(new Vector3(0.5f, 0.5f, 0f)</camera>
10 11 12 13 14 15	<pre>private void Update() { Ray isinYay = GameObject.Find("Main Ca if(Physics.Raycast(isinYay.out obj))</pre>	era").GetComponent <camera>().ViewportPointToRay(new Vector3(0.5f, 0.5f, 0f)</camera>
10 11 12 13 14 15 16 17	<pre>private void Update() { Ray isinYay = GameObject.Find("Main Ca if(Physics.Raycast(isinYay, out obj)) }</pre>	era").GetComponent <camera>().ViewportPointToRay(new Vector3(0.5f, 0.5f, 0f)</camera>
10 11 12 13 14 15 16 17 18	<pre>private void Update() { Ray isinYay = GameObject.Find("Main Ca if(Physics.Raycast(isinYay, out obj)) { if (obj.collider.gameObject.tag == } }</pre>	<pre>wera").GetComponent<camera>().ViewportPointToRay(new Vector3(0.5f, 0.5f, 0f) "3DGroup")</camera></pre>
10 11 12 13 14 15 16 17 18 9	<pre>private void Update() { Ray isinYay = GameObject.Find("Main Ca if(Physics.Raycast(isinYay, out obj)) { if (obj.collider.gameObject.tag == { } } } </pre>	<pre>wera").GetComponent<camera>().ViewportPointToRay(new Vector3(0.5f, 0.5f, 0f) "3DGroup")</camera></pre>
10 11 12 13 14 15 16 17 18 9 19 20	<pre>private void Update() { Ray isinYay = GameObject.Find("Main Ca if(Physics.Raycast(isinYay, out obj)) { if (obj.collider.gameObject.tag == {</pre>	<pre>mera").GetComponent<camera>().ViewportPointToRay(new Vector3(0.5f, 0.5f, 0f) "3DGroup") "3DGroup")</camera></pre>
10 11 12 13 14 15 16 17 18 9 19 20 21	<pre>private void Update() { Ray isinYay = GameObject.Find("Main Ca if(Physics.Raycast(isinYay, out obj)) { if(obj.collider.gameObject.tag == { Debug.Log(" Kamera isini carp) } } }</pre>	"3DGroup") "); // Kamera isini carpti: camera-ray hit
10 11 12 13 14 15 16 17 18 9 20 20 21 22	<pre>private void Update() { Ray isinYay = GameObject.Find("Main Ca if(Physics.Raycast(isinYay, out obj)) { if (obj.collider.gameObject.tag == { Debug.Log(" Kamera isini carp) } }</pre>	"3DGroup") ""); // Kamera isini carpti: camera-ray hit
10 11 12 13 14 15 16 17 18 20 21 20 21 22 23	<pre>private void Update() { Ray isinYay = GameObject.Find("Main Ca if(Physics.Raycast(isinYay, out obj)) { if (obj.collider.gameObject.tag == {</pre>	"3DGroup") "); // Kamera isini carpti: camera-ray hit

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.

Project 🖬 Console	3. ex Came
Clear * Collapse EnforPause Editor*	😡 999+ 🚓 🛛 😡 Ö Game 🔻 Display 1 🔻 Free Aspect 🔹 Spale 🖝 1x - Piny Focu
UnityEngine.DebugLog (object)	
15:20,42) Kamera laini carpti UnityEngine DebugLog (object)	
15:26:42] Kamera Islili carpli UnityEngine.Debugs.og (object)	
() [15:26:42] Kamara iaini carpti UnityEngine:DebugLog (otsecti	
UnityEngine:DebugLog (object)	

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...



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.



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.

🛃 🗹 Terrain				0 ≓	1
4	1	联	*		
Paint Trees Click to paint trees.	Paint Tr	ees			
Hold shift and click to	erase trees.				
Hold Ctrl and click to e	erase only tr	ees of	the selecte	ed type.	
Trees					
No trees defined.					
Mass Place Trees		🕸 E	dit Trees.	Refres	h
Settings					
Brush Size) —		- 40	
Tree Density			•-	83	
Tree Height	Random?	2 -		••	
Lock Width to Heigh	t 🖌				
Tree Width	Random?	× -		8)	

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 ‡ i
<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	e selected type.
Details	
No Detail objects define <mark>d.</mark>	
🛱 Edit Det	ails Refresh
Detail patches currently allocated: 1024	4
Detail instance density: 1677/216	
Brushes	New Brush
	😸 8 🔅
4 J W	$\overline{}$ + $\overline{}$
	~
100	
Brush Size	100
Opacity — 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.

Dasya Düzen Gönünüm Geçmiş Verimleri Araçlar Vaçıları		- a x
🖬 Intoor - Kaareeranslein Algimia Loo: K 🛛 M. Gelen Auturu - Kaanzeeralein (in x 🤚 Google Translate 🛛 X 😳 uurity terrain - Google Atama X 🔯 standard assets - Asset		
🗧 🔆 O 🛔 # https://assetstore.unity.com/pockages/Jd/environments/utbar/polygon.ctry.tow-poly-3d-art I 🏠 🔍 Arama		◎ 🛓 🔹 🐐 🖉 🗧 =
Ends soon Lunar New Year Mega Bundles Daring Bundle Brasing Bundle		×
🔂 Unity Asset Store Q standard assets	🔹 🖸 🗢	ළු 🖩 🥳
3D 2D Add-Ons Aud standard assets		Sell Assets
Quality essets Over 11,000 live-star assets Trusted Rated by 85,000+customers	Community support Supported by 100,000+ forum me	
1-24 of 75 results for standard assets Sort by Popularity + View Results 24 + 🖽 🎟	Refine by	
POLYGON City Pack - (Trailer) 3D Low P i i POLYGON City - Low Poly 3D Art by Synty svritv struteros 20	Hide Purchased Assets	
Vighkorbers 🖈 🛧 🖈 🛪 2 months age	3D I541	
CITY PACK	Essentials (3)	*
da 2011 - 501 rezcasi CEB Tunige		8 8 - 1 + 5



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:

	niim Git Proje Derle Hata-Ayıkta Test Analiz Araçlar Uzanbilar Pencere Yardım (Ara (Enf.+Q).	A Proje 1	<mark>10 - 1</mark> 0
- 0 8 - 5 8 3	🕐 - 🕾 - Debugi + Any CPU - 🕨 Unitarye ligar + 🛛 💼 🕼 🛫 🔚 👘 🐨 🗰 🖉 🔅 👘		년 Live Shar
SimpleActiveRorMenu.cs			
Assembly CSharp first	asse - 👫 UnityStandardAssetsUtilitySimpleActivatorMonu	 camSwitchButton 	
	ng System;		
2 usi	ng UnityEngine;		
3 US1	ng UnityEngine.UI;		
	espace UnityStandardAssets.Utility		
7 {			
	public class SimpleActivatorMenu : MonoBehaviour		
	// An incredibly simple menu which, when given references		
	<pre>// to gameobjects in the scene</pre>		
124	public Text camSwitchButton;		
	<pre>public GameObject[] objects;</pre>		
	<pre>private int m_CurrentActiveObject;</pre>		
	private void OnEnable()		
30			Ten 17 Pale 10 201
the states			
Tim Coron	Alberta Address Difference - Institution		
	Contrast of Assard And Advantage and a second second	Dura Dura	C. Cidere Darres
Mark Market		rioje Croya	Constant Classifier Constitution
Kod Apkie	27 olztemevi kaldurun	Ascembly-CSharp-firstpass SimpleActivator	Venuce 5 Etkin
Kod Aptie	zz gizłemeyć kałdirm	Assembly-CSharp-Firstpass SimpleActivator	Menucs S Elkin
Kod Apkie	ze gódiemegi kakterin	Assembly-CStarp-firstpass SimpleActivator	Menucs 5 Ebdn
Kod Aykie ICEOC7II Gereks Nets Linnes - Çed:	Registermey's katsern	Assembly-CSteep-Firstpass SimpleActivator	Venucs S Elkin

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 - Sampreficeras - PC, March Linux S	canaliahanan - Unity 2020-3.2011 Personal? + 22211				- D ×
File Edit Assets GameObject Component.	Mobile Input Window Help				
🖤 🖬 🗘 🖪 🖽 🗶 📈	Pivot Stocel is			🕸 📥 Account	- Loyers - Loyout -
	III Scene 📓 Asset Store 🛛 Game		2 Project Settings		15× 31
++ (*)0 III					atice
🕫 🛱 SampleScene*	and sectory.		Adaptive Performance	Plaver	9 2 9
S Marten Carriero	Scenes In Guild		Audio		
🕫 🧰 Watori/Simple (1)			Editor	Company Name	DesaurCompany
C Directional Light			Input Manager	Version	
G FPSCanboller			Package Manager		14
Water#Simple (f)Reflection Pro			Physics 20		itonii iTosture 20i
C Water4SimpleReflectionScinel			Player		and the second se
		Add Open Scenes	Preset Manager Ouwity		Table 1 A
	Platform		Scene Template	Default Cursor	None Treeture 201
		C Mar & Linux Standardon	Script Execution Order		A second s
	- PC, Mac & Linux Stanpatone		Ada		Senet
	iOS los	Target Platform Windows +	Analytics		
		Archilecture x86,64 +	Cloud Build		9 .
	S WebGL		Collaborate	Burning and the second structure and	
	Android	Copy PDB files		Seconds for HG, Mac & Linux Standar	
	ALL COMPANY	Greate Visual Studio Schution	Tags and Layers TextMain Dec	> loan	
		Development Build	Time	Resolution and Presentation	
			Timalina Venues Control	Solash Image	
			X8 Plugin Management	Tother Cellins	
				Rendering	
Bullant El Consola	All the second strategies and the second strategies of the			Color Space*	Damma +
+*	Universal Windows Platform	and an			
Contraction of Contraction	Abury Dove				2
- In RollerBall					2
E ThirdPersonCharacter			1	Color Gamut For Mac*	
 Bit Groas Platforminput Bit Editor 	Playor Settings_	Build Build And Run			
Effects		and the second sec			
Environmant	Contraction of the second second			These successions in the second second	
ParticleSystems		Simplex covate. Simple weaters Sind otor-onew Times	Reportanti Heriodologiconos	Step interval	5
En PhysicoMaterials				 Footstep Sounds 	
Prototyping				Jump Sound	alump O
- In Verschra					ÆLand O
in Packages					a - 1 3
A Visual Studio Editor Package version 2.0.1	4 is evailable, we strongly encourage you to a	odate from the Unity Package Manager for a better Visual Studi	o integration		8 2 X O

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.

C Project Settings				: 🗆 ×			
		۹.					
Adaptive Performance Audio	Player			0 7 0			
Editor	Company Name		DefaultC	ompany			
Graphics	Product Name		A Proje 1				
Package Manager	Version		0.1				
Physics Physics 2D Player	Default Icon			None (Texture 2D)			
Preset Manager Quality	Default Cursor			Select None			
Scene Template Script Execution Order Services Ads Analytics Cloud Build				(Texture 2D) Select			
	Cursor Hotspot		× o	Y O			
Cloud Diagnostics							
Collaborate	Settings for PC, Mac & Linux Standalone						
Tags and Layers	⊳ Icon						
TextMesh Pro Time	▶ Resolution and Presentation						
Timeline Version Control	▶ Splash Image						
XR Plugin Management	T Other Settings						
	Rendering						
	Color Space*		Gamm	18 💌			
	Auto Graphics	API for Windows	~				
	Auto Graphics API for Mac 🗸						
	Auto Graphics API for Linux 🖌						
	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	Player		0 ‡ ≎			
Editor	Company Name	DPU				
Graphics	Product Name	AraziGezisi				
Input Manager Package Manager	Version	0.1				
Physics Physics 2D Player	Default Icon		-			
Preset Manager			Select			
Quality Scene Template Script Execution Order	Default Cursor		None (Texture 2D)			
▼ Services			Select			
Ads Analytics Cloud Build	Cursor Hotspot	xo	YO			
		-				
Cloud Diagnostics	ų 100	Ð	5.0			
In-App Purchasing	Settings for PC, Mac & Linux Standalone					
Tags and Layers	▶ Icon					
TextMesh Pro Time	Resolution and Presentation					
Timeline Version Control	▶ Splash Image					
XR Plugin Management	Other Settings Rendering Color Space* Auto Graphics API for Windows Auto Graphics API for Mac Auto Graphics API for Linux Color Gamut For Mac* sRGB	Gamma ✓ ✓				
			t. - ∫ .			

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.





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.





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.



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**.

A Proje 1 - SampleScene - PC, Mac & Linux Standalone - U	July 20203.2011 Personal* + DX							
File Edit Assets GameObject Component Mobile Input	t Window Help							
😻 🤹 💭 🔝 🖽 😹 🗶 🖉 Pivot 🕾	Panels	2				🕸 📥 Account -	Loyen - Loy	
10 Herarchy 3. 1 TE Scene	Next Window	Cul+Tab				Inspector		
++ Shared Shared	Previous Window	Ctrl+Shift+Tab		🗶 🖬 + Gizmos + 🔬 🛛	· · · · · · · · · · · · · · · · · · ·	19 - Cylinder		
🖓 😫 SampleScene* 🔰 İ	Layouts	2				Teg Untagged	Bayel Defai	(to 5.00
A 🔁 Watertfilingle (1)	Collaborate				<u> </u>			
WaterAsimals (1)ReflectionFirstP	Plastic SCM					Transform		
C Water4 Simple He flection Scene Cr	Asset Store				•	Robelion	X 95.1 Y 27.8	A 981.5
<u>Фман сапада</u>	Package Manager			A		Scale	X 0 69866 X 0 03728	
C Directional Light	Asset Management	2		2 Menuna				
TPSController >	TextMeshFro	2		- AND		Cylinder (Me	sh Filter)	071
Water4SimpleReflectionFistPers	General				and the second s	Mesh	LICylinder.	•
S Plane	Rendering	>			and and a second se	👘 🖉 Mesh Render		071
CyEnder	Animation		Animation Chi+6	Section 1		Materials		
	Sequencing	5	Animator Parameter			Lighting		
	Analysis	2				Cast Shadows		
	Al	3				Contribute Global	-	
	Ul Tooliot	×.				Receive that a little		
						= Probils		
1993						Light Probes	Blend Probes	
						Reflection Probes	Bland Probes	100
1985 C						Anchor Override	None (Transform)	0
No. Contraction of the Contracti						Additional Settings		
						Dynamic Occiutio	Her Object Moven	
Project E Console								
Favorites	Assets					🗧 🗧 🗹 Cepsule Coli	ider	
Q. All Materials								
G. All Pretates		_						
				S Streets			None IPhysic Material	
Assets						Conter	X 5.96046 Y 0	I -8.9406
Scenes				Jimer k New Material		Hainht	10.5000001	
Standard Asseta						Direction	V-Axin	
e Barnenas								
English Characters						Contracteria		
Audio							dand	COLUMN
Be Distance Y		A lotter for hard start	622244					
a more are a autorio usualitets trime scene, miease ensui	te there is almoy's exactly on	a non-subtrenen da	and second					

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.

A Proje 1 - Sampreficiene - PC, Mar. & Linux Scandalneet - Unity 2020.3.2011 Rev.	onari «2001»						0 ×
File Edit Assets GameObject Component Mubile Input Window Help							
🖤 🕂 🖉 🔛 🖾 🥵 🗶 🖉 Pivot Citocut <table-cell></table-cell>		I I I		1	Account -	Layers - Laye	ait 🖓
Therarchy 3 i II Scene Asset Store	ens Games ens al set a central 2001 a		W He + Games +	1 0	Inspector		
12 SampleScene* i					Cylinder		
Create New Animation					Teg Untagged		(t): 0.95
← → ← ↑ ■ = Belgeter > A UNETY > A Proje 1 > Asse	e> → 0 }	P AnnAurth			Fostion 7 Rotation	X 95.1 V 27.8 X 0 V 90	● 2 i 7 981.5 7 90
	Leader Street	1943	2.11			X 0.69866 V 0.03728	
A CONTROL AD	Degujarrae tarita	tur boyut	AND AND AND		U Cylinder (Mes	h Filter)	
SampleScenes	7.02.2022 13.05	Dosya klasičná	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			DCylinder	Ø
10 Models Scenes	7.02.2022 17:29	Dotya Kasona Domia Manani		1 mar 1	😸 🖌 Mesh Rendere		071
3DDXF		Donya Mindola			Materials		
3DDXPinpor					Lighting		
📕 A Proje 1					Receive Shadows	5	
B vs							
AraziGezis							
Assets			and the second second		Probus		
A CONTRACTOR OF A CONTRACTOR O					Light Probes	Blend Probes	
Dosya ada New Animation anim		14			Anchor Dyerride	Nooe (Transform)	(6)
Kayit turu: anim (fianim)				- 19 - 19 - 19 -	Additional Settings		
						Per Object Motion	100
Proje ct 🖉 Klastvileri Gizle	Γ	Kaydet İstal		a 1	Dynamic Occlusio	e :	
+1	5		4	🕈 🖌 👘 🖬 🖓	🔒 🗹 Capsule Collid		1000 E
Fovenices							
C, Att Models							
C, All Prefate	to begin animating Cylinder, creat	te an Animator and an Animatio	n Cép		Material	None Physic Material	
 ■ Assets ■ GampinScenes ■ Scenes ■ Standard Accet ■ Standard Accet 		raate	sch New Material Sta	n Terreir	Conter Redius Holght Direction	X 5.96046 Y 0 0.5000001 2 Y-Axe	Z -8.9406
- Demenara					Contraction in a second		0.43
Generation Conversion Conversion					Thoday Stand	in the second se	- 561
Audio							
There are 2 audio listeness in the scene. Please ensure there is always e	cactly one audio listener in the sce	ene.					

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 2 - Sampreficera - PC, March Linus	Standaltane - Unity 2020 3 2011 Per	-112231						0 ×
File Edit Assets GameObject Component	it. Mobile Input. Window. Help							
* + Ø 🛛 🗆 # 🗙						🖄 📥 Account 🗝	Layers - Lay	
Therarchy a	1 III Scelle Asset Store	e de Game				Inspector		
-+	Shaped + 20	· · · · · · · · · · ·		🛠 🖬 + Gizmos + 🖕 📖				Statio -
SampleBcene*					2	Teg Untager	bavel Defai	ALC: NO
🖌 🕤 Waterdfilmpla (1)					V			
Water4Simple (1)ReflectionFi	nstF					Transform		
C3) waters of the first of the	and the second second	aller .	-		•	Fostion	A 95.1 1 27.5	7.981.5
C Animation	a ar in			And the second s		Scala	X 0 60866 X 0 0373	90
				Redde and			N D BROOM IN C MANA	
	A. 1996-199746			a setting		Cylinder (Me	sh Filter)	0 7 1
Add Prop	erty ATransfor			- The state		Mesh.	Gylinder	œ.
22	A Positio	n n			The second	👘 🖉 Mesh Render		0 7 1
(a)	A Soule		÷ /			Materials		
	Mesh Ren	2047/81				- Lighting		
			_			Cast Shedows		100
						Probles		
						Light Probes	Blend Probes	
						Anchor Charries	None (Transform)	(0)
						Additional Settions	None Constantion	
Do	pesheet Curves +					Motion Vectors	Per Object Motion	100
B Distant B Passada					4.1			
+*					1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1			1.00.00
Favorites	- Assets					Gapeure Con	des.	CHORE IN
C. All Materials			1			Edit Calider		
C. All Pretates		- En 🙈 🕨	1 - UKRIN	🦲 🖂 =🔺				
							None IPhysic Materia	
Assets			er OSF.OT Janeerie	Many Ma			X 5.96946 Y 0	Z -8.9406
Scenes						Hadara	0.5000001	
🔻 🚍 Standard Acceta						Direction	2. 19. 89. 1	
P III 20								
T De Characture						2+ 2: Animistor		0.2 1
 FirstPersonOharaster 							t⊋Cylinder	
Profate							None (Avatar)	. 🤊 .
C There are 2 audio listeners in the scene	Please ensure there is always e	wactly one audio listener in the scen	e.				3	

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.

A Proje 3 - Sampreficerary - PC, Mar. R Linux Standardsen -	Unity 2020 3 3011 Personal! < 22(1)				- a ×
File Edit Assets GameObject Component Mobile Inpu	at Window Help			track provid prevanance	
W +F Q 20 E1 88 X 2Pivot D	(Local) In	► II +1		tit) 📥 Account	• Loyers • Loyout •
Therarchy O I TO Scen	e Asset Store de Game			i Dinspector	
SamoleScene* I			X III + Sterilos - N	Cylinder	
Weler4Simple				Tog Untagged	 Layer Default
20 Waterd Single (1) 20 Waterd Single (1)RollectionFintP				Transform	e 2 i
Animation Provide III - He He He He He He He He He He He He He	15 [190] Varie] (200) (0)	aio×		Postion Retation Scelle	X 95.1 V 27.8 Z 981.5 X 0 Y 146.093 Z 90. X 0 69866 Y 0.03728 Z 1
- e			a all a south	- Dulladar Di	
			A REAL	Mesn	Difference (5)
- AGylinder: Rotation					Contraction Contraction
Add Property				Materials	
				Cast Shadows	04
				Receive Shadow	5.4
				Contribute Glob	
				Receive Galler I	
				Probes	Right Division
				Reflection Probe	e Bland Probes
				Anchor Override	None (Transform) (B)
Dopediest	Survey, a me			Additional Setting	
		and the second second second second second second second second second second second second second second second		Motion Vectors	Per Object Motion 🔹
Project E Console				a I Oynamic Occius	ie •
			- 215	🔴 🔁 🗧 📒 🖉 Capsule Co	llider ⊕⊄ i
Q. All Materials					
G, All Models		. 🖛 🛲 🎫 🌠	🚺 🕞 n = 🔺		
Cit An Procades			🗾 🚾 ° 🔼		None Physic Material)
Assets	Sampleli Scenes, Standard	altin pog - Google Arema — M.	Ma. New Tet. spin		X 5.96040 Y 0 Z -8.9406
Scenes	7			Hednis	13.500.0001
 Standard Assets Standard Assets 				Direction	Y-Aza
🕫 🖿 Carnettas.				Animator	
 Ter Characters Ter Citat Parane Character 				Controller	P.O.Bala
		the second second second second second second second second second second second second second second second se			Augusta (D)
ा 🔉 म 🜛 🛈 🐴 🖻	1 🖽 刘 💽 😰 🎯	🖼 📦 🔚 🔄 🛷			🕽 🛎 🖼 🕼 d = 7.02.2022 📲

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. 1st 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**.



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. 2nd 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**.



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

Inspector		а:
Tags & Layers		0 ≠ :
▼ Tags		
Tag O		
New Tag Name	Cam	+
	Sava	

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.

Medly Hole 24.2								a.
€ unity					۵	Hub 3.0.1 Available	t)	2
9 Projects	Projects	Create a new paraject with	Uony 2020.3.2011		- 0	ADD	NEW	
t.eam	Project Name	Templates			Settings	form	Last Modified	Q
I Installs	ISG1Gama C.\Users\Asus\Docar Ueity Version: 2019.1	° #	•	•	Project Name A Proje1	latform -	3 days ago	3
	VRC2 C. Uherry Asses/Docar Unity Westing: 2016;1	20	30	High Definition RP	C:\Users\Asus\Documents\A UNITY +++	latform -	10 days ago	1
	AlzanolVR C:10sers'Acue\Docue Unity Version: 2018;3	0	0	0		latform +	17 days ago	
	Alzanol ChUsers'Mess/Docar Unity Version, 2019,1	Universal Render Pipeline	Karting Microgume	,89, AR		latform +	20 days ago	
	VRC1 C:\Users:\Asus\Docur Unity Version: 2019.7	0	<u>*</u>	<u>*</u>		latform -	23 days ago	
	AizanoiGame C. UJaern'Anum/Docum Unity Version: 2015.1.14	n			CANCEL	attorn -	24 days ago	1
	AlzanoIFPS Children JasshDocame Unity Version: 2020 3 20	ntavA.UNITV\ArzaniaF≓% If1			2020.3.20f1 - Cu	rrent platform –	a month ago	1
	Theresia Crititisers (Acud) Document Unity Version, 2019 7, 14	ntel,4,0H(TV\Theresia			2019.1.14f1 - Ct	rrent 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.

Image: Control Image	File Edit Assets GameObject Component Tuto	orial Window Help					ы X
Provector 0 Provector 0	V V C I S X ZPM	not Gigliocel in	De H DE		🕸 📤 Account 🕶	Layers - Laya	sit -
	THeathy 3 I B	🗉 Scene 📓 Asset Store 🛛 🚥 Game			B Inspector		
C Parts process Yournel C Parts process Yournel C Clow C C C C C C C C C C C C C C C C C C C	+ + · · · · · · · · · · · · · · · · · ·	ssolay 1 + Full HO (1923)#1080) + Sciala 🖲	0.524	Maximize On Play Multi Audio Stats Gizmos +	Cube Tog Untagged		State
Project Bonode And Mancink C Add Ma	C Post-process Volume				Postion > Rotation > Scole >	(1,01 V.0 (0 V.0 (1 V.1	Z -1.47 Z 0 Z 1
Mein Mein Mein					- III Cube (Mesh Fi		
Project Consider And Meterials And						BCube	
Project B Console Autorisis Aut					- Mesh Rendere		0 ¥
Project Bronsole Assets Asset A					► Materials		
Project Control of a document Project Control of a document Control of a document Project Control of a document Project Control of a document Control of a document Project Control of a document Con					T Lighting Cast Shedows	0m	
Project BiConsole Project BiConsole					Examples Erroration		
Lighe Protoses Acchor Overlice Acchor					Probes		
Refuence Proteine Bend Proteine Refuence Proteine Protect B Console Favrites Asset						Elena Prabes	
Province Pr					Reflection Probes	Blend Probes	
Prodect ≧ Conside Prodect ≧ Conside					Additional Settings Dynamic Occlusic		
Procet ■ Console						Layert:	
Favatts Q. Alf Madetalia Q. Alf Madeta	Project B Console			a i	- 😰 🖓 Box Collider		
Q. All Materialis Q. All Materialis Is Trager Q. All Materialis Q. All Materialis Is Trager Q. All Materialis Q. All Materialis Is Trager A service Is Trager Materialis Is Example Assets Is In the limit Peaks Strips Is In the limit Peaks Is Example Assets Is In the limit Peaks Is In the limit Peaks Materialis Peaks Is In the limit Peaks Is Sends Is In the limit Peaks Is In the limit Peaks Sends Is In the limit Peaks Is In the limit Peaks Is Sends Is In the limit Peaks Is In the limit Peaks	Favorites						
ExempleAsels Fiendle, Malerials Presets Script Settings Traviality, Resonal Scores, Script Settings Traviality, Resonal Scores, Script Settings, Traviality, Resonal Scores, Script, Settings, Traviality, Resonal Scores, Script, Settings, Traviality, Resonal Scores, Script, Settings, Traviality, Resonal Scores, Script, Settings, Traviality, Resonal Scores, Script, Settings, Traviality, Resonal Scores, Script, Settings, Traviality, Resonal Scores, Script, Settings, Traviality, Resonal Scores, Script, Settings, Traviality, Resonal Scores, Script, Scores, Script, Scores, Script, Scores, Script, Scores, Script, Scores, Script, Scores, Script, Scores, Script, Scores, Script, Scores,	C, All Materials C, All Models C, All Profess				is Trigger Material (Center)	None (Physic Material)	
B Scription	 Assets In ExampleAssets In Materials In Projects In Scenes 				Renk 1 Metern Cheder Univer	n) sal Render Pipeline/Lit	2 1 0 1 1 - E01
Add Component Add Component Add Component	Te ScriptA Te Settings Tetroiolinfo Te Pockages				Ad	d Component	

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**.



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



A Proje 1 - SampleScene - PC Mar Ar Linux Stan	datione - Unity 202032011 Personal* - DX	11.					-	a ×
File Edit Assets GameObject Component Tu	itorial Window Help							
🖤 💽 🖸 🖾 🗶 🖉 🖉					6	Account -	Layers -	Loyout -
TE Herarchy ∂ 1 + + + + + = = + • © SampleBeeno* 1 G Main Carrera	El Scene 🖬 Asset Store 🐽 6 Disolay 1 - Full HD (1920x1080)	eine • Scala Ø	0,52%	Maximize On Play - Mute Aut	i Stats Dizmos +	Inspector Post-proce Tag Unbagged	ss Volume – Layer	, ð. 1 Statie v Default -
O Directional Light Post-process Volume Common Assemm Common Assemm Come						Transform Readion Relation Scale	X 0 X 0 X 0 Y 0 X 1 Y 1	● ≠ 1 2 0 2 0 2 1
						■ Volume Mode Weight Priority Profile	Diotel 0 1: SampleSceneP	● ≓ !
						Poremapping Variante Variante	0.56 12.41	• • • • • •
Project E Console								
	Assets	ing Dag Dag	a beitrage Tutelide oversed	K 🙀 🧼	D e * 1914	Det interdie F - Vignette	att Overnide ad Component	
En Tutorialinfo En Pockages Visual Studio Editor Pack age version 2.0.14	is available, we strongly encourage y	su to update from the Unity F	ackage Manager for a better Visu	el Studio integration				

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.



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**.





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.





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



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 Projet - Untitled - PC, Mar. & Losa Stand	tatione - Unity 2020.3.2011 Personal*			- a ×
File Edit Assets GameObject Componen	t Tutorial Window Help			
	Pivet Cabooal in		🛱 📥 Account	- Loyers - Loyout -
T Hearthy 3.	I III Scene 📓 Asset Store 📾 Geme		1 Dispector	
+ • • • •			elmize On Play, Mule Audile, Stats, Dizmos, + 🔊 😼 🚽	(pv) Statle -
			C, Tog Unlago	a - Layer Default -
C Directional Light			Transform	e ≠ 1
() Text (TMP)			Position	X0 Y- Z0
EventSystem			Retation	
C Plane C Cube			Scele	
			Till Cube (Mer	sh Filter) Ø ヹ 1
			Mesh	
			🗉 🔛 🖌 Mesh Ren	dever 671
		and the second se	- Materials	
			Element 0	CLit O
			- Lighting	
			Cast Shadows	
			Contribute Giot	
			Receive Grobe	
			Frobat	Filmed Dephan
T T			Reflection Prot	es Bland Probes
	Many Tave		Anctor Overtia	e None (Transform) @
	INCW IEXL		Additional Settin	u n
Project E Console			Dynamii: Occla	nio 🖌
Fovoritos	Anata		Rendering Lay	n N Lagert
Q. All Materials			La Maturi	
Q, All Models	and the first the line is	Rea Rea Rea -	Juider U	
ExampleAssets			Components that are cannot be multi-edit	only on some of the beliected objects ad.
 Materials Pronets 				Add Component
🖿 Scenes				
Scripts				
 TextMesh Pro 				
Tataiallafo				
A Visual Studio Editor Package version 2.0	0.14 is evailable, we strongly encourage you to update from the	Unity Package Manager for a better Visual Studio	integration	日 第 第 8 日

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



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

A Projet - Ubahness - PC Mat & Emur Standa	alone - Unity 2020.5.2011 Personal* < 0X11 >					- a ×
File Edit Assets GameObject Component	Tutorial Window Help					
🖤 🚻 🖉 🖾 🖽 🗶 🖾	Plvot Gibocel in			40	Account -	Layers - Loyout -
TE Hierarchy A. 1	III Scienter 📓 Asset Store 🕫 Ga				O Inspector	
++ + ++ ++	Shaped - 20 - 41 -	1 - (20) (1	🗶 🗰 + Gizmos + 🧐		ToytMuchBr	- Toys (UR)
• € Usahosi* i		1 ····	<u>-</u>		Text Input GSF Dörsei betsim 1	Enable HTL Editor
		GSE Gareel II	otisim			Normal *
		OSF OUISEI II	euşini		Main Settings	
	-	Tasarim		-	Font Asset Material Preset Font Style Pont Size Auto Size	B U S ob AB SC 36
					Vertex Color Color Gradient Override Tags	· · · · · · · · · · · · · · · · · · ·
						Character () Word () Line () Panagraph ()
						토프로토토 토프 = 시 전 자
E Depart P Console				2.1		Enabled
+ ·				A # 16 8514	Overflow	Overflow +
Favorites	- Assets					
G. All Materials					Vertical Mapping	Character
Q, All Prefats					ana ana ang ang ang ang ang ang ang ang	
					ayour Properties	
Material MicropieAssets MicropieAssets					Roperty Van Vin Wicth D Vin Height D Preferred Width 517 Preferred Height 01.4 Realtie Width dius Realtie Width dius	AN SOUTCE TestMashProVOLD TestMashProVOLD SE TestMashProVOLD (2 TestMashProVOLD (2 TestMashProVOLD) (2 TestMashProVOLD) (2 TestMashProVOLD) (2 TestMashProVOLD) (2 TestMashProVOLD)
Di Totorialinfo De Parkanas						
A Visual Studio Editor Package version 2.0.1	14 is evailable, we strongly encourage yo	u to update from the Unity Package Man	ager for a better Visual Studio Integration			◎ 遼 憲 臻

The image in game mode will look like this.


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**.



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.





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**.



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.



Then, do the same thing in the second scene.

File Edit Assets GameObject Co	omponent Windo	w Help								
🖉 👫 🖉 🖾 🖽 😹	X ZPivot	SLocal	10	(F C)	II. (#0)		段 🌢	Account . Leyen	+ Loyout	
To Mesanthy			ette Germe					Inspector		
+• • •		Shaced	+ 20 📳 🗣 😂	• #0 H •		🛠 🛤 + Gizmos + 🕞 🔤		- S. P. Mana		
Main Camera								Source Image	1910 Seette	
CO Directional Light									and the second second	
Button									None (Material)	
C EventSystem										
CHARTO								 Baycast Padding Maskable 	0.050	
								Image Type	Sliced	
		_			4					
								Pixels Per Unit M	91	
			and the owner where the party is not	4	STREET, STREET		-	- E & Button		
					Button2	►				-
				4	Dattonz	<u> </u>		Tamer Grootie	Soler Fint	
								Normal Color		-14
										1
								Pressed Color		- 61
								Disabled Color	-	-9
									• 5	
Be Project El Consola							ā i		Automatic	
+							♦ ♥ ★ #8			
Q All Materials	Assoto > Scen							On Click (j		
C. All Models			AA					Runtime (= Geo	c21 ASSanneye	
	#	HT I	99					∎gec21 ⊙		
 Assets Scenes 	Oreigt2	Gedis21	SampleSoScene2							
E Packages								Default ULW		01
								Shinter (1800		
								Button +		
A Visuel Studio Editor Package ve	ersion 2 0 14 is ev	west	rongly encourage you to upda	te from the Unity Package Mana	ger for a better Visual Studio	integration			* *	38 0

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.

LiBertonSteve - Sahae3 - PC Mac & Linux Standalor	- Linity 2020.3.2011 Resemant - COX11 -					a ×
File Edit Assets GameObject Component Windo	/ Help					
🖤 🐼 🖉 🖾 🖽 🗱 🗶 🖉 Pivot	SLocal II I	C	9 🌰	Account -	ayers - Li	sycut -
To Hierarchy 3: 1	II Scene as Game			@ Inspector/		3.1
+ • . N /0	Shaded + 20 🝷 📲 😂 + 🚳 🛱 🖓 +	🔀 🗰 + Gramos + 🐟 🛲		RA Z Cam	ad	Static-
Shine Campan			2.	Tag Units		er bl 🛛 🖛
C Directional Light			V_			
() Plane				- II Redt T	ransform	0.2 1
C Sphere	3	2	0			1 N.X.
() Cuite	27		11.12			n nan e
Canves PErentSystem			_			HI
(T) Chemokania						11 (20)
						\$ <u>.</u>
						20
			-	- IX - Canva		0 7 I.
					Scients	ipace - Ovena -
				Pixel Party	Screen Space	Overlay
				Sort Order	Screen Space	- Camera
				Target Old	World Space	le
				Addisional Sha	ider CF Nothing	0.5
				- HE 🖉 Canva	s Scaler	0 # E
					Constant	l Pixel Size 🔹
Project El Consolo			a i			1
+•		N 2.4				
Pavoritos Asseta				- 11 × Grant	e Roveaster	0 2 1
C. All Models						PERSONAL PROPERTY AND
G, All Prefats					ed Glar	Sector and the sector
1 Tel Annula						2.
Scenes Scenes					< Everymi	ng) -
TextMesh Pm Pm						
m Packagas					Add Camponent	1
PropertiesGUIQ is being called recursively. If you	want to render the default gui for shader properties then call PropertiesDefaultGUI0	instead				

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

d UBentonScene - Sahael - PC Maca									×
File Edit Assets GameObject Cor	nponent Window	Help							
1 中 Ø 図 目 郷	K Pivot						- (1)	Account - Loyers - Loyout	
Therarchy +* Gobre3* Main Camera		ID Scerie Shaded	- 000-Gemm + 20	4) & - 200 II -		🗶 🛤 + Games + 👳 ai	+	© Inspector © ✓ Canvas Tag Unitaggied • Layer UI	-3, 1 Statie+
C Directional Light C Directional Light C Capsula C Sphate C Cataba C Evendbystem							< Projection of the second sec	Rest Transform Rest Rest	
+•							2 · · · #	Paste World Placement	_
 * Province Q. Ali Models Q. Ali Models Q. Ali Models Q. Ali Preface * ar Assets * Bio Genes * Bio Testification Dro * Bio Packages 	Assets Scenas 1	extilue (b						Add Component	9 ¥ 1
PropertiesGUI0 is being called re	culsively. If you v	ant to rende	e the detault guilton	shader properties then call P	PropertiesDefaultGU() instead			表 表	38 (3)

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.



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

Cleanorsonal - sehred - of Ma	e w Linux Stimita	tone - Unity Xiz	0.3.2011 Personal* CDX1	12				- 11 8
File Edit Assets GameObject C	omponent Win	sow Help	_					
🖷 🕂 🗘 🖸 🛄 🕫	X Plu					n 🕹	Account - Layer	s - Loyout -
Therarchy			con Cerne					
+* .**		Display 1	+ Free Aspect	• Scala 🔴	1.21	Maianuze ChiPlay Mute Audio Stats Dizmos +	FR Z Text (TMP	a Station
1 Sahne3* 19 Main Camera							Teg Untagged	💌 Layar UL 💌
Directional Light							- 12 Reat Transf	om 9 ≠ 1
Y C Canvas								
C Image								7471 -5.69695 0
Capaute							B Bight	Battom
C Sphere							- 7.9	5917 4.523081 million
C EventSystem					GSF GÍT		Pivot X 0.1	5 ¥ 0.5
C. 1998 C. 1997 (1997)							Rotation X o	×0 70
							Scale X-1	
			_	_			Canvas Ref	den 0 2 1
							Dall Transparent M	
					100			
							TextMeshP	19 - Text (UI) 🛛 2
				-			Text Input	Enoble RTL Editor
							ess off	
The Project 🖾 Consola							Text Style	Normal
+•						2 ≠ 10 mil	Main Settings	
Pavoritos							Font Asset	III iberationSans SI 🕢
Q All Models							Kont Style	n i tr s
C, All Prefats								an AH SC
· av Assets								72
Scenes								
Packages								s Min - Mex WD% Line
								1
							Lavour Prenerties	- Andrew Street, and the state of
A Properties (200 is teinin called	weather the	or want to ters	des the detault out for	shader properties then call Pic	scellins/Tetain/GLED instead			

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.



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.





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



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.

	Pivot GLocal				► II H	0	Account •	Layers + Li	ayout ·
serarchy		as Game					i O Inspector	Lighting	
	Shaded	÷ 20 📲	4 4 - 10 - 10	2	% mt + Gizmon	100 MAR	Package 19		
C Main Camera						and the second second	A CONTRACTOR OF A CONTRACTOR OFTA CONTRACTOR O	Edit View In Pa	ickage Mariage
C Directional Light	- Ba	kage Manager	Soft Name 2 + E	mars - 173			nternation		
C Example Assets		filmer From the		1.1					
					Starter Assets - Third Per	son Character Control	ler Vandon		
	1 90	pie City pack plain		10 2	Shifty Territologies		Citegory		
		ple urbph Ealter		문제 복	Version 114 - Peuritary 13, 2023 [see		Description		
	90	pie VA					Utāmer Asnetu z	in decisined to wa	
	ske	schifab for Unity	90.0 85a (b)	phal ±	Starter Assets are designed to work as a q		tonettime for y		
	E Sky	box Series Free		4.2 🖸	developing character controllers for variou methods that are easy to build and expand	us game genies, using systems and d upon for different projects important	exatams and to		
	+ 5n:						traja dtr. mport		
	- Spe				Images & Videos		LTE pric 2021.		rg agagan id Pictori
	i ⊳ Spi				Internet Reserved Internet		Stafter Assetti : Svetsve unit Cir	is the agent require 11	
	P Spc						file packages		
	spr						prototic and proto	arter Assets packs ner Assets - Third	
	F 1966						Dependensies		
	⊫ Stu						Packegi mme		
kolant 🗃 Consola	in Sta	ter Assets - Third Pers	on Character Co	1.1.24	Package Stre		The provide Energy		
MASTO C. SELEVICION	5 Sty	Ize Modulor House Der	mo Free - Erbeito	1.0 ±					
Favorites Ass		ize Water Texture			Purchased Date				
G, All Motenais				1.0 ±	Depember 04, 2021				
O, All Printable					Release Details				
Assets	All 10	I packages shown			114 (Current) - released on February 13.				
ExampleAssets	Lants						ent		
Presents E Presents E Scents S Settings - Turnealinto	sampileAssets Materialis	Presenta	Lenar	Senata	Settings Utonatives Olick to project	a import assets from the package into	eyzələr		

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)**.

	A REAL PROPERTY.						1000
 Tescharwiterchange - Maypround 	I - PC, Mac & Unior Standard	ve - only zoons zon reasonal spo	3			2	ы. A.
File Edit Assets GameObject Con	nporent: Cinemachine Tu	conar tools window Help	Water Hard		154/1	alianti teorizzationenes interactioneneses pass	144 A 144
	× Helinne afree	li bi	U.S. MARKE		.0.	A MECOUNT - Layers - Lay	yout •
E Hierarchy	0 i H	Scene do Game	-t - 50 51 -	W at a Company of Article		G Inspector S Lighting	d :
S Playground			Construction of the second sec			S sessimon (acene vased)	10000000
 C Lighting. 					N.	10	(subsen)
G MaleiCamera						ltr.	
PlayerFollowCompta					1.44		
	meren destruction		-				
🐨 MLEventSystem							
						R I	
						1 0	
Project E Console					5 e e esta		
Materials *		ThirdPersonController > Scene					
Protots							
E Scripts		え 🦳					
Settings							
Editor		St.					
Environment	Payarounit Ha	roround RaygroundSatti					
Mobile							
Character							
Prefals							
Scener .	Azsets/StorterAccoto/	ThimPortonController/Scenec/Pla	yground unity			Asset Labels	
A Visual Studies Editor Package ser	sion 2.0.17 is available w	e strongly encourage you to upda	ite from the Unity Package Manager for a	beller Viscal Studio integration			8 2 3 O

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



Proceed key will be proceeding here.



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**.

	ent Cinemachine Tutorial To	ools Window Help					
000183	Pivot @Local ib		II 11		0 .	Account . Layers .	Layout
seranchy	à i III Scene	as Game			1.	Inspector Lighting	
		+ 20 + 4+ st + 160 🛱 •		K me + Gizmou + Artin		Vanguard By T. Choonyun	
Playground					y.		
P C Lighting							
MainCanwra						Model Rig Animation	Materials
PlayerAmolure					And Address of the Owner, which the Owne	Material Creation Mc Import via A	fatenalDésc
	and entropy and a second				(Parap	Location Dec	1 74-11
Contraction (Contraction)						Materials Estre	A Materials
					and the second second	EAU OS	a neascanara.
						Autorial assignments sale be to	the reported
			T Car		Veral and a	Remapped Materials	
					65238UU •		
						VanguardBodyMat None (Ma	terial)
						Vanguard_VisonMat. El None (Me	terual)
					Star Call		
oject 😐 Console					A + 102		
Fovorites • Aus	ots						
All Motenais					Un ·		
L All Prefabe					<u>n</u>		
					reactive 1		
Accets							
Assets ExampleAssets							
Assets E ExampleAssets Materials Process	4						
Assets DE ExampleAssets De Prevaits De Stenes	10						
Assets ExampleAssets Materials Diferents Diferents Differents Sections	1.						
Assets E ExompleAssets Disconsis B Scenas S Scenas S Scenas S Scenas S Sectors S Settings S Statuer Assets	To .						

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.

					Areas played to
	PIVOR CELOCOI 15			Ci occount - Lavers	- LEVOUt
 de 	5haded	- 20 - 41 - 20 - 200	12 mr + Gizmou + Aria	Value of the second sec	a Manual Instant San a
= 🗣 Playground			1	Virigine dy trend	Singung migori aeria
 Stugnting Environment 					200
Male Canvera				Model Rig Anim	ation Materials
PlayerFollow Compta				Animation Type Hum	anaid
 Busiling the second seco	Among and a	C		Persp Avatar Definition Great	te From This Model Configue
				Skin Weights Stan Optice te Game Obje	tard (4 Bones)
		H K S			
miant E Consola				\$ # # #2	
Project 🗏 Console					
roject ≣Consols Favorites Assets S All Motenais				-	
roject ∎Console Favorites Alisets Q. Ali Mordeln Q. Ali Perlaos	auler tractoriation	Malecali. Presenta Acen		*	
roject ■ Console Fovoritos Q. Al Motennas Q. Al Puetos D. Al Puetos Provenov	Huber transformer to	Malecals systems Science	n Scipto Sellingo Stanierasiers	тисинията	
reject Console Forvorites Q. Al Maderiale Q. Al Maderiale Q. Al Maderiale Dokular EcomplexAssets Materiale Propet		Malerala Balance Stream	an sergit sening Sinterior a	Tai of lat min	
Reject Console Foventiae Q. All Modernale Q. All Postace U Dockular Discrete Discrete Secrete Secrete Secrete Discrete	nuter transportant	Malerati etaini ana	lej (vienne) (vienne) n screfe senny siumerksers	Taiya ka mia	

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.

Edit Assets GameObject Component	Cinemachine Tutorial Tools	Window Help					
	Privot GLocal 15				0	📥 Account • Layers • 1	avout
Reparchy	a i il Scene i	🗱 Game				I O Inspector V Lighting	
	Shaded	• 20 🕈 🕂 🗐	• ▶0 報 •	1/2 me = Gizmos	Set seat	- 🗑 🗸 Armature_Mesh	
 C Lighting 				110		Tag Untagged · Layer	
Trivino nemeral	12					Transform	e t
PlayerFplow Campta						Position X o X o	2.0
* C PlayerArmature					THE TAX	Rotation X o Y o	Z o
DS PlayerCarseseRoot CO Decreetty					1 10 100	Scale X 1 Y 1	
C Armature Mash					ATA	Skinned Mesh Renderer	0 2
TEL DARMEN	Concession in the local division of the loca				and the second se		
O ULEventSystem	2					Lenger	
						Center X -2 98021 Y -0.091	21, 2, 0,020
						Extent ¥ 0.528184 ¥ 0.880	921 Z 0.720
						Quality Auto	
						Update When Offson	
						Mesti EArmature.	Mesh
						Root Bone AHps (Tran	stormi
						Materials	
						Carl Sharawii Do	
						Probes	
roject. 🗐 Console						Light Probes Bland Probe	
						2 Reflection Proces Bland Proces	
Fovorites Assets		a <u>. 1</u> 1 7 <u>2</u>	3 3 A A	<u></u>			
R All Models						• 41 Reflector Proce (Refle	
D, All Poiltable				and the second	The second second second second second second second second second second second second second second second s	 Additional Settings 	
					StatterAssets TutorsEnfo	Department Coccuse -	
Dokular	este la second					Horseshid anyes a service	
ExampleAssets						M_Armature_Body (Mater	(a)
Presets	JA 9	,				Stoder Universal Render P	ipelir Ed
Scenes ·	- U						
Scripts	eacme Vangeard By T					M_Armature_Arms (Mater	iat) 🛛 🖼
🖨 StarterAssets						Shader Universal Render P	ipci - Edi

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



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

File Edit Assets GameObject Component Cinemachin	ne Tutorial Tools Window Help						
🗶 🔢 🕼 🖂 🖽 🖾 Pivok 😒	gLocal ib	► U		(6)	📥 Account - Lavers	+ Layout	
Eliveranchy di i +• ⇔ //	III Scene 🍫 Gome Shaded + 20 🕈	+	% mt - Gizmou		O Inspector Uphling		
Physpround Officiants Officiants Officiants PhyserFindstor PhyserFindstor PhyserFindstor Officiants Offi				Trace			
In Project III Console +*				at Are e ato			
All Misterialis All Misterialis All Misterialis All Misterialis All Misterialis							
Conservice Construction Con	Value and Value and Value and Value and Value and Value and Value and Value and Value and Value and Value and V	ang ang ang ang ang ang ang ang ang ang					
							100 0

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.

🤏 Blender File View						- 🗆 ×
✓ Volumes		C:\Users\Asus\Documents\		Q		8 - 7 - 🗱
Freel Disk (C:)					Operator Preset	s + + +
∽ System					Path Mode	Сору 🗸 🗃
n Home				Embed Textures Embed textures in Fi	BX binary file (only fo	"Copy" path mode!).
Documents					Limit to	Selected Objects
Downloads Music Pictures	A A ISG	A UNITY	A UNITY Ders Dokümanları	A UNITY Ders Projeleri		Visible Objects Active Collection
Videos					Object Types	Empty
F Fonts						Camera
OneDrive						Ampatura
						Mesh
✓ Bookmarks	Any Video Converter	Ansd4R4 Studio	AD J/D MD Panare	ANTACAD		Other
	Any fact converter	Angeli + acoulo	Alexience apers	HUNCHU		Custom Properties
∼ Recent					✓ Transform	
늘 source	×				Scale	1.00
CastleTexture					Apply Scalings	All Local +
🛅 Uygulamalar					Forward	-Z Forevard
Materyaller						Ville
Saved Pictures	BLENDER		C++ Eğitim	EEEng		N ADD TON
BLENDER						Apply Unit
source					-	
3D Models	untitled.fbx			- +	Export FBX	Cancel

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

Volumes	< → 1 2 Ľ	C\Users\Asus\Documents\		م	1 1 1	8 - 7 - *
Yerel Disk (C:)					Operator Presets	• + -
∽ System					Path Mode	Copy 🗸 💆
A Home					Batch Mode	Off 🗸 🔛
📰 Desktop					😒 Include	
Documents					Limit to	Selected Objects
L Downloads	A A İSG	A UNITY	A UNITY Ders Dokümanları	A UNITY Ders Projeleri		Visible Objects
J Music						Active Collection
Videos					Object Types	Empty
F Fonts						Gamera
CrieDrive						Lamp
						Amature
🗠 Bookmarks 🚥						Mesh
+ Add Beokmark	Any Video Converter	AnyMP4 Studio	AR-VR-MR Papers	AutoCAD		other
						Custom Properties
∼ Recent ····					\sim Transform	
🖿 source 🛛 🗵					Scale	
astleTexture					Apply Scalings	All Local 🗸 🗸
늘 Uygulamalar					Forward	-Z Forward 🗸
Materyaller	21 (11/2/2011		20122221	122223		Y Up
BI ENDER	BLENDER	64+	L++ Egitim	EEEng		Acoly Unit
a source						
3D Models	iki_katlı_ev			- +	Export FBX	
E source	- Constanting			and the second sec		

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.

googievr / cardboard-xr-plugin						
ode 💿 Issues 🗊 🕮 Pull requests 🗊) 🗇 Security 🗠 Insights					
	🧰 cardboard-xr-plugin 🔤			⊙Watch 24 +		
	P master + P T Branch (\$ 25 Bigs		o lo tie 🙁 Add	The - O Code +	About	
	 xinvunh0929 Cardboard XR Plusin v123.0 		Liscal		Google Cardboard XR Plugin for Unity	
		Cardboard XR Pluz	Clone		Readme	
	🖿 .github	Cardboard XR Plus	HTTPS SSH GitHub CLI	Eupy unite tipboard		
	🖿 Editor	Candboard XR Plug	https://github.com/googlevr/co	ardboard-xr-p2x	 Custom properties 275 starti 	
	🖿 Resources	Candboard XR Plug				
	E Runtime	Cardobard XR Plug	中 Open with GitHub Desktop		♀ 117 farks. Report repository	
	Samples~	Candboard XR Plug	Download ZIP			
		Cardobard XR Plug			Releases 25	
					on Dec 15, 203	
	D Editor.meta					
					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.

VRCardboard - SampleScene - Windows, Mac, Linux - L	nity 2022.3.14f1 <dx11></dx11>	
File Edit Assets GameObject Component Service	Window Help	
😝 ке 🔻 📥	Panels	>
TE Hierarchy	Next Window C	trl+Tab
SampleScene :		
Directional Light	Layouts	<u>,</u>
	Unity Version Control	
	Search	>
	Asset Store	
	Package Manager	
	Text	,
	TextMeshPro	,
	General	>
	Rendering	>
	Animation	>
	Audio	>
	Sequencing	>
	Analysis	>
	Asset Management	>
	UI Toolkit	>
	Visual Scripting	>

Go to Window>Package Manager

Open the at the menu by pressing the + key here. Select the **Add package from git URL**... line.

E Package Manager + ▼ Packages: In Project ▼ Sort: Name (as :) ▼ Fi	lters 👻 Clear Fil	i 🗆 X
Add package from disk Add package from tarball Add package from git URL Add package by name Fronkoutes - Onnov		Code Coverage Unlock Remove
E Code Coverage	1.2.4	From Unity Registry by Unity Technologies Inc.
Custom NUnit	1.0.6	com.unity.testtools.codecoverage
Editor Coroutines	1.0.0	
🔒 JetBrains Rider Editor	3.0.26	Description Version History Dependencies Samples
Profile Analyzer	1.2.2	
	2.0.1	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
Carter Test Framework	1.1.33	data on demand, for manual testing or when there are no automated tests in the project.
TextMeshPro	3,0.6 🕜	
Timeline	1.7.6	
Unity UI	1.0.0	
Version Control	2.2.0	
Visual Scripting	1.9.1	
🗳 Visual Studio Code Editor	1.2.5	
Visual Studio Editor	2.0.22	

Paste the link we just copied here and press the **Add** button.

Package Manager		 : 🗆 ×
+ Packages; In Project Sort: Name (asc) Hitters Clear		
ጓ* Add package from git URL		
https://github.com/googlevr/cardboard-xr-plugin.git		Add
 Engineering 7 packages 	🔒 1.2.4 - June 01, 2023 Release	
Packages - Unity		ļ

■ Package Manager + ▼ Fackages: In Project ▼ Sort: Name (asc) ▼ Filters ▼ Clear Fi	lters	:
All Services		
■ Features ■ Engineering 7 packages	Google Cardboard XR Plugin for Unity	Update Remove
T Packages - Other	Author unknown	
Google Cardboard XR Plugin for Unity 1.24.0 Git	com.google.xr.cardboard	
■ Packages - Unity		
Code Coverage 1.2.4	Description Version History Dependencies Samples	
Custom NUnit 1.0.6		
Editor Coroutines 1.0.0	Provides rendering and input support for Google Cardboard.	
La JetBrains Rider Editor 3.0.26	https://github.com/googlevr/cardboard-xr-plugin.git	
Warning So Te Te Te Tre Oo you want to enable the backends? Doing so will RESTART* the editor. Yes No Visual Studio Code Editor 1.2.5 Visual Studio Editor 2.0.22	Dut em hat	
Package Manager + Packages: In Project Sort: Name (asc) Filters Clear Filt All Services	ers	: DX

All Services						
T Features		Code Coverade				
Engineering 7 packages		■ 1.2.4 · June 01, 2023 Release				
Packages - Other		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				
A Code Coverage	1.2.4					
🔗 Custom NUnit	1.0.6	Description Version History Dependencies Samples				
Editor Coroutines	1.0.0					
𝔗 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)) 🔻 Fliters 👻 Clear F			
All Services				
▼ Features		Coords Cordboord VD Dlugin	forlinity	
		Google Cardboard XR Plugin	for Unity	Update Remove
7 peskeges		() 1.23.0 OI	Click to check for	or updates and update to latest
T Packages - Other		Author unknown	version	
Google Cardboard XR Plugin for Unity	1.23.0 Git	com.google.xr.cardboard		
Packages - Unity				
Code Coverage	1.2.4	Description Version History Dependencies	Samples	
P Custom NUnit	1.0.6			
Editor Coroutines	1.0.0	Provides rendering and input support for Goog	le Cardboard.	
JetBrains Rider Editor	3.0.26	Installed From	oin ait	
Profile Analyzer	1.2.2	Thtps://githubicom/googlevi/caruboard-xi-piug	an-ar-	
P Settings Manager	2.0.1			
E 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 Plugin for Unity	Update Remove
1.23.0 Git	
Author unknown	
com google xr.cardboard	
Description Version History Dependencies Samplas	
Hello Cardboard 2,07 M8 🗸	Reimport
Basic sample, minimal project setup.	

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		Clone		Google Cardboard XR Plug
🖿 allstar	Cardboard XR Plug	HTTPS GitHub CLI		山 Readme 4월 Apache-2.0 license
🖿 .github	Cardboard XR Plug	https://github.com/googlevr/cardboard-x	r-plu	-∧- Activity
Editor	Cardboard XR Plugi	Clone using the web URL		Custom properties ☆ 279 stars
Resources	Cardboard XR Plug	🔛 Open with GitHub Desktop		24 watching
Runtime	Cardboard XR Plug	Download ZIP		9 121 forks Report repository

After the compressed file is downloaded to the hard disk, open it and extract it.

		1.01	00, at
∨ 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	
📴 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	disk			×
÷ → ∽ ↑	> İndirilenler > cardboard-	xr-plugin-master > cardboard-xr-plugin-master >	~ C	
Düzenle 🔻 Yenî	klasör			= - •
Giriş	Ad ∽ Geçen ay	Değiştirme tarihi		
Galeri	package.json	8.03.2024 00:14		
OneDrive -	🛅 allstar	8.03.2024 00:14		
E Beigeier	-github	8.03.2024 00:14		
	Editor	8.03.2024 00:14		
Masaustu -	Carl Resources	8.03.2024 00:14	Önizlenecek bir dosya s	eçin.
Kesimier	🧰 Runtime	8.03.2024 00:14		
💴 Masaüst 🖈 🛓 İndirilen 🖈	🚞 Samplet~	8.03.2024 00:14		
📑 Belgeler 🖈				
🔀 Resimle 🖈				
	Dosya adı:			package.json file (*.json) 🗸
				Aç Íptal

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.

File Edit Assets Carried



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**.

8 H * A					19	• Ш н		
Herechy								
+ • • • ♥ HeleCardboard ■ Confluent ○ Paintant • ♥ Transm • ♥ Finer		Libra -		R R H - HE			91 N 1 1 1 1 1	
S GradessPPTeet				Weld Santage Scenes in Baild			191	(Frank
							AND DEPUTY PROPERTY	
			Settom	- Antroid				
			💭 Windows, Mac, Linux 😝 Teinture Commission 1	Um Przyir Settings S.S. no				
			🖬 Tentralit	e Fogoriensten Depon Project Strette Angelineste (Scoope Play) Cristel Applitunde (Scoope Play) Cristel sendenssage Sun Grevier				
			WebGL		Disabled			
			Universal Windows Paliform		No. and goaters	Parch Analita		
					Patching to Basilian Tol Painces Saw			
					Autopowers without			
					 Compression Method 			
				• A said Import Germére Mai Territe Site - No Ormide - Terture Composition - No Demide -				
Bragest B Carade				(RuyunSenngs)			Sector Pattern(1) Sector not	A.I
+• 6 # 5464								4 a #47

Let's add our current scene to the Scenes in Build section by clicking Add Open Scenes.

Build Settings			: 🗆 ×
Scenes In Build			
Samples/Google Cardboard XR Plugin for Ur Samples/Google Cardboard XR Plugin for Ur	nity/1.23.0/Hello Cardboard/Scen	es/HelloCardboard	
Platform	🖷 Android		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)		
WohGl	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					: 🗆 >
Adaptive Performance Audio	Player				@‡:
Editor	Company Name	DPU			
Graphics	Product Name	VRCardboard	i		
Input Manager Memory Settings	Version	0.1			
Package Manager Physics Physics 2D	Default Icon				None (Texture 2D)
Player					Select
Preset Manager Quality Scene Template	Default Cursor				None (Texture 2D)
Script Execution Order					Select
Services Tags and Layers	Cursor Hotspot	× 0		Y O	
TextMesh Pro	P		4	5	4
Time Timeline	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.



Now, go to the **Other Settings** section related to the **Player**. Select **ETC2** as the **Texture compression format**.

Package Manager	Multithreaded Rendering	2	
Physics	Static Batching	✓	
Physics 2D	Dynamic Batching		
Preset Manager	Sprite Batching Threshold	•	300
Quality	GPU Compute Skinning*	~	
Scene Template	Graphics Jobs (Experimental)		
Script Execution Order	Texture compression format	ASTC	
Tags and Layers	Normal Map Encoding	ETC	
TextMesh Pro	Lightmap Encoding	ETC2	
Time	HDR Cubemap Encoding	× ASTC	
Timeline Ul Builder Version Control	Lightmap Streaming Streaming Priority	DXT DXT + RGTC(BC4, BC5)	
Visual Scripting	Frame Timing Stats		
XR Plug-in Management	OpenGL: Profiler GPU Recorders	<u>×</u>	

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 possit	sle 🗌	
Recycle command buffers*	~	
Apply display rotation during rendering		

Check if the company name is the same as above.

Identification		
Override Default Package Name		
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				
 ✓ Samplee//Soogle Carc Platform Image: Platform Image: Windows, Mac, Mac, Mace,	Adaptive Performance Audio Editor Graphics Input Manager Memory Settings Package Manager Physics Physics 2D Player Preset Manager Quality Scene Template	Player Virtual Texturing (Experimental)* 360 Stareo Capture* Load/Store Action Debug Mode Vulkan Settings SROB Write Mode* Number of swapchain buffers* Acquire swapchain image late as possible Racycle command buffers*	3	0 t i
Dedicated Serv	Script Execution Order	Apply display rotation during rendering	2	
Android	Services Tags and Layers TextMesh Pro Time TimeIne UI Builder Version Control Vision Control	Identification Dverride Default Package Name Package Name Vorsion ¹ Bundle Version Code Minimum API Level	com DPU VRCattboard C.1 1. Android 5.1 'Lollipop' (API level 22)	
iOS los	XR Plug-in Management		Automatic (highest installed)	
Para PS4 Paris PS5 Assot Import Overrid Max Texture Size I Texture Compression I Player Settings		Configuration Scripting Backend Api Compatibility Level* U.2CPP Code Generation Dete Compiler Configuration Use incremental GC Allow downloads over HTTP* Mute Other Audio Sources* Target Architectures.	 Automatic (highest installed) Android 5.1 'Lollipop' (API level 22) Android 6.0 'Marshmallow' (API level 23) Android 7.0 'Nougat' (API level 24) Android 7.1 'Nougat' (API level 25) Android 8.0 'Dreo' (API level 26) Android 8.1 'Dreo' (API level 27) Android 9.0 'Pic' (API level 28) Android 0.0 (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 (API level 31) Android 12.0 (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.

Target Architectures	100- est	
ARMv7	~	
ARM64		

Set the Internet Access section to Require.

Ilistali Location	FIGICI EXICILIAI	
Internet Access	Auto	¥
Write Permission	 Auto 	Ť
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		: 0)
Scenes In Build		٩	
 Samples/Google Carc 	Adaptive Performance Audio	Player	● ≠ :
	Editor Graphics Input Manager Memory Settings Package Manager Develop	Publishing Sattings Keystore Manager Project Keystore Custom Keystore	
Platform	Physics 2D Player	() The application will be signed with a debug key	
L Windows, Mac,	Preset Manager Quality		
Dedicated Serv	Script Execution Order Services		
Android	Tags and Layers TextMesh Pro	Project Key Allas Debug	
WebGL	Timeline UI Builder		
Universal wind	Version Control	Build	
iOS ios	KR Plug-in Management	Custom Main Manifest Custom Launcher Manifest	
Pura PS4		Custom Main Gradle Template 🛛 🗹	
		Assets\Plugins\Android\mainTemplate.gradle	
PJE PS5		Custom Launcher Gradie Template	
TAsset Import Overrid		Custom Base Gradle Template	
Max Texture Size		Custom Gradie Properties Template	
Texture Compression 1		Assets/Plugins/Android/gradieTemplate.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.



Here, paste the codes to the place indicated by the arrow above the ****DEPS****} expression and save it.

	Kaspersky	٠	mainTemplate.gradle	×	KameraGecis.cs	CameraChangi	ng.cs	+
Dos	ya Düzenle (Görünüm	i i i i i i i i i i i i i i i i i i i					
app **/ dep	oly plugin: 'com APPLY_PLUGINS** pendencies { implementation implementation ';	.androi fileTr android	d.library' ee(dir: 'libs', : x.appcompat:appc	incl	ude: ['*.jar']) t:1.4.2'			
i i j	implementation ' implementation ' implementation '	com.goo com.goo com.goo	gle.android.gms: gle.android.mate gle.protobuf:pro	play rial tobu	-services-vision:20.1 :material:1.6.1' f-javalite:3.19.4'	.3'		
[)EPS}							

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.



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	XR Plug-in Mana	gement		a	
Editor Graphics Input Manager Memory Settings Package Manager Physics Physics 2D Player	Initialize XR on Startup Plug-in Providers • ARCore	≣ ✓	+	8	4
Preset Manager Quality Scene Template Script Execution Order Services Tags and Layers TextMesh Pro	Oculus Unity Mock HMD	tion tracking and mig	ation can be found below.		
Time Timeline UI Builder Version Control Visual Scripting XR Plug-in Management	View Documentation				

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			
 Samples/Google Cardboard XR Plugin for U 	nity/1.23.0/Hello Cardboard/Scen	es/HelloCardboard	0
			Add Open Scenes
Platform	🗰 Android		
💭 Windows, Mac, Linux 🏫	Texture Compression	Use Player Settings	
-	ETC2 fallback	32-bit	
Dedicated Server	Export Project		
🚔 Android 🔂			
NYML	Build App Bundle (Google Play)	Dicabled	
5 WebGL	Run Device	Samsung SM N975F (RF8NB069L8L)	Refresh
Universal Windows Platform		Default device	Patch And Run
		All compatible devices	j.
iOS los	Development Build	<enter ip=""></enter>	
PEA PS4		 Samsung SM N975F (RF8NB069L8L) 	
	Deep Profiling Societ Dobuscies		
ers PS5	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				X
← → ∨ ↑ 📄 > Bu bilgisayar > Yeni Birim (D:) > UN	TY > VRCardboard > APK1		C APK1 klasöründe	ara 🔊
Düzenle 🔫 Yeni klasör				E • 🧕
🛅 A EĞİTİM MATEI 🛛 🛆	Değiştirme tarihi	Tür	Boyut	
C VRRoomAPK2	Aramanızla eşleş	en öğe yok.		
🫅 Ders Notları - Ka				
C VRGallery1				
👻 💻 Bu bilgisayar				
> 🐸 Windows (C:)				
> 🛥 Yeni Birim (D:)				
> 💱 Ağ				
Dosya adı: VROda1				~
Kayıt türü: apk (*.apk)				~
			Kaydet	Íptal

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**.

e *	(E 🕶 📥		Inspect	tor							c
'≡ Hie	erarchy	a :	M 🔳	GraphicsAPIT	ext						Stati
+-	Q. All	园	Tag	Untagged				Layer	Default		
1	HelloCardboard	* :	∀ 3 8 - 1	Transform							0 ī
1.11	M Point light		Position			X	0		4		4.99
	► 🗑 Treasure		Rotation				0		0		0
	GraphicsAPITe	ext	Scale		60					Z	1
	Player		- B - I	Mesh Renderer							05

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)**.

8 Inspector			a	1		
Player			Static			
Tag Unlagged		Layer Default				
Transform			0 ‡			
	X 29	Y -2.55	Z -55.7			
Rotation	X 0					
🕆 😥 🖌 Character Contre	oller		07			
Slope Limit	45					
Step Offset						
	0.08					
Min Move Distance	0.001					
	X: O					
Radius						
Height						
🔻 😫 🗹 Look (Script)			07			
	A None (I	(ransform)				
Toggle Angle						
Speed						
	Add Comos	ment				

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							×	
Build Settings	← → × ↑	> Bu bilgisayar → Yeni Birim (D.)	UNITY > VRWalk > APK					ρ	
Scenes in Build V Samples/Google Cantboard XR Pilipi	Düzenle • Yeni klasor			8-					
	📓 Wdeslar 🛛 🥐		Değiştirme tatihi		Boyut				
	1 🛋 Gay 💦 📌	URWalk.apk		APK Dosyata					
	🛅 Madencilikte VR								
	🛅 Blender - A Tem								
Flatform	BALVAGROUND								
🖵 Windows, Mac, Linux	🛅 Ders Notlan - Ka								
Dedicated Server	1								
in Android	🗸 🜉 Bubiginyar								
WebGL	> 🎬 Windows (Ci)								
1. Universal Window's Platform	> = Wari Birim (D.)								
DS as	5 💇 4g								
	Para a ser United	n an t							
- 115 <u>-2</u> PB+	Kenet türü: eck (*	apk)							
are P35									
Asset Import Overrides	 Klasörluri Gizla 						Kaydet	lptal	
Texture Cumprespon No Ovorridu									
Player Settings.			100L	Bulld And Dun					

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.

With an account you can dow license keys, and participat	inload development tools, get e in the Vuforia community.	
First Name *	Last Name *	
Company *	Select Country of Residence *	
Email Address *	Username *	
Password *	Confirm Password *	
Ben robot değil	Im CLAPTERA Inclusion - Control	

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: engine- developer portal	Home	Pricing	Downloads	Library	Develop	Support	Hello KaanErarslan Log Out		
Account Manager	License Manager	Target Man	ager Credenti	als Manager					
Dashboard Account Info	Getti	ng Starte	ed		D	ownload S	DK and Tools		
Account Usage	New to track Im	Vuforia? Go to lages, objects	o the docs and le and environmer	arn how to its.	Do UV	Download the SDK for Unity, Android, IOS and UWP. Try out Vuforia with many samples available.			
Payment Method Billing Statements	Vuforia Developer Library Unity API Reference Native API Reference					Download SDK Download Samples Download Tools			
	Vufor	ia Relea	ses		Su	upport			
	Find mo Vuforia	re informatio Enginel	n about the late	st releases o	f Ge su	et support for youpport for youpport options.	our issues by using our various		
	Release Release Release	Notes SDK Notes Tools &	ents & Apps		Di	ew latest issues scuss topics ontact us			

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**.

vuforial engine developer portal	Home	Pricing	Downloads	Library	Develop	Support	Hello KaanErarslan Log Out
Account Manager	License Manager	Target Mana	ager Credenti	als Manager			
License Ma	nager				Get Basi	c Buy Premiu	m Buy Cloud Add On
Learn more about Create a license ke	licensing. y for your applicati	on.					
Search							
Name		Primary UUI	О ① Тур	2	Status	5 V	Date Modified
er		N/A	Bas	c	Active	E.	Jun 04, 2023
		N/A	Bas	c	Active		Jan 28, 2021
kaWiks		N/A	Bas	c	Active		Jan 14, 2021

In this window, specify the **License name** and check the agreement box. Continue by clicking **Confirm**.

vuforia: engine: developer portal	Home	Pricing I	Downloads	Library	Develop	Support	Hello KaanErarslan Log Out
Account Manager	License Manager	Target Mana	ger Credenti	ials Manager			
Back To License Manage	r						
	a kay ta ya	ur Pacie	nlan				
Add a licens	se key to yo		pian				
GSF-GiT							
You can change this la	ter						
By checking this be	ox, I acknowledge th	at this license	key is subject t	o the terms a	and condition	s of the Vuforia	Developer Agreement.
Cancel	onfirm						

Now, a **license** has been opened with the name we created on our list.

vuforia engine developer portal	Home	Pricing	Downloads	Library	Develop	Support	Hello KaanErarslan Log Out
Account Manager	License Manager	Target Mana	ager Credent	ials Manager			
License Ma	nager				Get Bas	ic Buy Premiu	m Buy Cloud Add On
Learn more about Create a license ke	licensing. y for your applicati	ion.					
Search							
Name		Primary UUI	О 🛈 Тур	e	Statu	5 🗸	Date Modified
GSF-GiT		N/A	Bas	ic	Activ	e	Jan 01, 2024
er		N/A	Bas	ic	Activ	9	Jun 04, 2023
ic d		N/A	Bas	ic	Activ	9	Jan 28, 2021
L.kaWurks		N/A	Bas	ic	Activ	e	Jan 14, 2021

Next step, we can go under the **Target Manager**. Here, click **Add Database** to create a database.

vuforia: engine: developer portal	Home	Pricing D	ownloads	Library	Develop	Support	Hello KaanErarslar	Log Out
Account Manager	License Manager	Target Manag	er Credenti	als Manager				
Target Man	ager		e 26				Add Da	tabase
Use the Target Mar	hager to create and	i manage data	bases and ta	irgets.				
Search								
Database		Туре			Targets		Date Modified	
RK t		Devi	ce		3		Feb 21, 2021	
R	•	Devi	ce		1		Jun 04, 2023	Ĩ
on Battes d€	L	Devi	ce		8		Feb 10, 2021	
IC I	1. Sec. 1.	Devi	ce		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 C	Database
get Manager to	create and manage databases and targets.		
	Create Database		
	Database Name * GSF-GiT-Veri-Tabani	Date Modified	
		Feb 21, 2021	Î
	Туре:	un 04, 2023	÷.
sinden	 Device Cloud 	Feb 10, 2021	
	O VuMark	Feb 16, 2021	*
	Cancel		

Our database will be visible in the list.

vuforia: engine: developer portal	Home	Pricing [Downloads	Library	Develop	Support	Hello KaanErarslan Log Out
Account Manager	License Manager	Target Manaş	ger Credentia	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	
ER	Device	1	Jun 04, 2023	
on Barcendon	Device	8	Feb 10, 2021	
G <mark>SF-GiT-Ve</mark> ri-Tabani	Device	0	Jan 01, 2024	
₩iC;=id	Device	5	Feb 16, 2021	

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**





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 Home	Pricing Do	wnloads Librar	y Develop	Support	Hello KaanErarslan 🗸 丨 Log Out
Account Manager	License Manager	Target Manager	Credentials Mana	ger		
Target Manager > GSF-G	T-Veri-Taba					
GSF-GiT-Ver	i-Tabani edit	Name				
Targets (0)						
Add Target]					Download Database (All)
Target Name		Туре	Rating	g 🛈 S	tatus 🗸	Date Modified

The window that opens offers options to upload files to the database.

	\square	00	1
\Box	\Box		T
Image	Multi	Cylinder	Object
File:			
			Browse
Enter the width of y same scale as your unit scale. The targe	our target in scene u augmented virtual c et's height will be cal	units. The size of the ta ontent. Vuforia uses m culated when you u <mark>p</mark> lo	rget should be on leters as the defau lad your image.
Name.			
Name must be unig	ue to a database. W	hen a target is detecte	d in your applicati

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:			
\Box	\bigcirc	00	
Image	Multi	Cylinder	Object
File:			
GSF-LOGO.png			Browse
.Jpg or .png (max file	2mb)		
Width:			
1			
Enter the width of yo	ur target in scene i	units. The size of the t	arget should be on t
same scale as your a unit scale. The target Name:	ugmented virtual c 's height will be cal	ontent. Vuforia uses n culated when you upl	neters as the defaul oad your image.
GSF-LOGO			
Name must be unique this will be reported	ie to a database. W in the API.	hen a target is detecte	ed <mark>in your applicatio</mark>
		Cancel	Add

You may be asked to change the properties of the file:

Target Manager > GSF-GiT-Veri-Taba...

	\square	07	
Image	Multi	Cylinder	Object
File:			
GSF-LOGO.png			Browse
wiath:			
Enter the width of y same scale as your unit scale. The targe	our target in scene (augmented virtual c tt's height will be cal	units. The size of the sontent. Vuforia uses culated when you up	target should be on meters as the defau load your image.
I Enter the width of y same scale as your unit scale. The targe Name: GSF-LOGO	our target in scene i augmented virtual c tt's height will be cal	units. The size of the s ontent. Vuforia uses culated when you up	arget should be on meters as the defau load your image.
I Enter the width of y same scale as your unit scale. The targe Name: GSF-LOGO Name must be uniq this will be reported	our target in scene (augmented virtual c et's height will be cal ue to a database. W l in the API.	units. The size of the i ontent. Vuforia uses culated when you up 'hen a target is detect	target should be on meters as the defat load your image. red in your applicati

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**.

GSF-GIT-Veri-Tabani Type: Device	Edit Name			
Targets (1)				
Add Target				Download Database (All)
Target Name	Туре	Rating ①	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) [*] Home	Pricing Do	wnloads Libra	ry Develop	Support	Hello KaanErarslan 🛩 Log Out
Account Manager	License Manager	Target Manager	Credentials Man	ager		
Target Manager > GSF-G	iT-Veri-Taba					
GSF-GiT-Ver	i-Tabani 🖽	Name				
Type: Device						
Targets (1)						
Add Target						Download Database (All)
Target Name		Туре	Ratin	g ① S	tatus 🗸	Date Modified
□ १डा॰ GSF-LOG	O	Image	**	*** A	ctive	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.



Get all the files by clicking Next.

(asc) 🔻 Filters 🔻	Clear Filters	: a
	2.1.6 ± ▲ Vuforia Core Samples 1.7 ± 10.19.3 · November 15, 2023 Asset Store	Import
gla selfic	1.3 Import Unity Package	Support
ilt-in to Unity 2	1 Import Content Step 1 of 2 1.4 All None	
22	1 ✓ Assets 1 ✓ Common 1.0 ✓ ✓ 1.0.1 ✓ ✓ 1.0.2 ✓ ✓ 1.0.3 ✓ ✓ 1.0.4 ✓ ✓ 1.0.5 ✓ ✓ 1.0.6 ✓ ✓ 1.0.7 ✓ ✓ 1.0.8 ✓ ✓ 1.0.1 ✓ ✓ 1.0.2 ✓ ✓ 1.0.3 ✓ ✓ 1.0.4 ✓ ✓ 1.0.5 ✓ ✓	com/getting-started/getting-started-v a in Unity.
	1 ✓ Image: Control Contro Contro Control Contro Control Contro Control Control Control Con	jine (hitps://engine.vuforia.com/), the sy to develoo cross-platform apps for a 0,000 apps including titles from EA, So
1	1 ▼ ✓ ■ Icons New 2.1.0 ▼ ✓ ■ Android New 2.1.0 ▼ ✓ ■ AdaptiveIcons New ✓ 圏 BackGround_x081.png New	senes that show how to create AR exp rgets and trackers.
Lot		r) as a Vuforia Developer when you arr ps://developer.vuforia.com/support) fr
	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.



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 engine developer portal	Home	Pricing [Downloads	Library	Develop	Support	Hello KaanErarslan Log Out
Account Manager	License Manager	Target Mana	ger Credenti	als Manager			
License Ma	nager				Get Basic	: Buy Premiur	n Buy Cloud Add On
Learn more about Create a license ke	licensing. y for your applicati	on.					
Search							
Name		Primary UUID	ανΤ 🗊	e	Status	· •	Date Modified
GSF-GIT	_	N/A	Bas	ic	Active		Jan 01, 2024
ers		N/A	Bas	ic	Active		Jun 04, 2023
EC d		N/A	Bas	ic	Active		Jan 28, 2021
ka W tks		N/A	Bas	ic	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.

vuforio: engine: developer portal	Home	Pricing	Downloads	Library	Develop	Support	Hello KaanErarslan Log Out
Account Manager	License Manager	Target Mana	iger Credent	ials Manager			
License Manager > GSF-	GIT						
GSF-GIT Edit N	ame Delete License Key						
License Key	Jsage						
Please copy the licen	se key below into yo	ur app					7
ARd+42D////AAABme ODjyA5VPV5RUCfLsko cllCfsxIZNycqd2ptP sjE/+L26x2upcEuni2	+HCIICIEAyozWwe+wDu DeGxIvv1ZjpormKHC49I (/CCQpMTlvQEzNqGn408 SaqAkpaIJdMnTEWT49G1	nMC8gkAYXU6Z ymNc55F6sVFu Mo3hjocrH6cn hTtq4yybICide	aH3w1D745JLpU6) CPQ3IK36nBzU71(OMt1cJHd2taA5E) d3x1kPcEmgs2Mg1	H53nAOVg0ZaL QYwfopnC5jDh B2aLc5r+0DFr ZegfTqMPh2Qf	auijn4ITNbYyg gyzDi5VrUl4Y5 blXf1E1jipFvy {t	b+6x05JTp8rD7 UHvjb1YEuDG0W AS9Dth/GpKIpm	
Plan Type: Basic							
Status: Active							
Created: Jan 01, 2024	4 18:58						
License UUID: 6bf6d	4b0d23447c3b64008	cedabe1bc8					
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.
	Kaspersky		• README.md	• +							- 0	×
Dosy	a Düzenle	Görünüm										
GSF-	Git Anahtar:											
ARd+ 6k05 2alc	4zD/////AAAI JTp8rD70Djy/ 5r+80FroLXf	Bme+HCfiClEAyoz ASVPVSRUCfLskØe 1E1JipFvyAS9Dth	Wwe+wDunMC8gkAYXU GxIvv12jpormKHC49 /GpKIpmsjE/+L26x2	6ZuH3w1D7453LpU6H53nAOVg0ZaLaui3n4 LymMcS5E6sVFuXPQ3TK96n8zU71QYwfopnl upeEuni2sqAkpsEJdMnTEWT49G1hTtq4yyl	TTNbYygb+ CSjDhQyZD15VrU14 bICidd3x1kPoBmgs	YSUHvjb1YEuDGØWc 2MgZegfTqMPh2QfX	11GfsxI (t	ZMyoqd2prK/CCQpMT1	vQEzNqGi	1405Mo3hjocr	H6on9Mt1cJHd2	2taA5E
			Farki kaydet									
			< → < ↑ 🚺	🕽 > Bu bilgisayar > Yeni Birim (D:) > UNITY >	A PACKAGES >		G					
			Düzenle 👻 Yeni klast					.				
			EGITIM MATERY Scenes Film Rulosu	Ad HoloLens-Basics-Project-main MixedRealityLearning-development MixedRealityToolkit-Unity-main	Değiştirme tənhi 9.12.2023 15:53 10.12.2023 15:58 11.12.2023 12:46	Tür Dosya klasörü Dosya klasörü Dosya klasörü	Boyut					
			🛅 Ekran Görüntüle	MRTK3-iet-tutonals Aquarium-main	17.12.2023 23:34	Dosya klasörü						
			🛩 📮 Bu bilgisəyər > 🖼 Windows (C:)	OpenXR-Unity-MixedReality-Samples-m Standard Assets Tool	11.12.2023 12:46 2.12.2023 18:27	Dosya klasŏrū Dosya klasŏrū						
			> 🚽 Yeni Birim (D:)	StarterAssets TPS	2.12.2023 16:27	Dosya klasörü Dosya klasörü						
			- ¥ Ag	vufona-sdk-uwp-10-19-3	10.12.2023 15:58	Dosya klasõrü						
			Dosya adı: GSF- Kayıt turu: Meti	Vuforia Key n belgeleri (* txt)					v v			
			 Klasörlen Gizle 		Kodla	ma: UTF-8		Kaydet [ip	tal			
512	5011								Miller	0145 (CD1E)	LITE A	

Now, go to the **Target Manager** in Vuforia. Our goal is to prepare and download the database for Unity.

Account manager License Mana	er Target Manager Credent	lais Manager		
arget Manager			Add I	Database
se the Target Manager to create	and manage databases and ta	argets.		
Search				
latabase	Туре	Targets	Date Modified	
RK T	Device	3	Feb 21, 2021	1
ER	Device	1	Jun 04, 2023	1
n Baites de	Device	8	Feb 10, 2021	1
SF-GiT-Veri-Tabani	Device	1	Jan 01, 2024	1
ECHIC	Device	5	Feb 16, 2021	1

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	ri-Tabani 👪	t Name				
Targets (1)						
Add Target					Download Database (1)	
Target Name		Туре	Rating ①	Status 🗸	Date Modified	
1 selected Delete						
🗹 🍿 GSF-LOO	50	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	base	
1 of 1 active targets will b	e downloaded	
Name: GSF-GiT-Veri-Tabani		
Select a development pla	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.

6 Inspector		a :
T Global		
Vuforia Version	10.19.3	
0	pen Library Article	
App License Key		
	Add License	
Delayed Initialization		
Carriera Focus Mode	CONTINUOUS AUTOFOCUS	
Additional Focus mo CameraDevice.SetF platforms.	des can be set using the ocusMode() API, but are not supported on	
Camera Device Mode	MODE_DEFAULT	•
Log Level	ALL	•
Max Simultaneous Trac		
Virtual Scene Scale Fat		
The factor above co correspond to one in Vuforia will be tranal scated. Note that or than 1.0 is not support	infigures how many Unity scene units nater in the physical world. Poses reported formed accordingly. Scene content will not HobLens and MagicLeap, a factor other arted.	by be
Trained Targets Contin	2	
Share Recordings in iTu	~	
⊤ Databases		
Databases will be an ObserverBehaviour	tomatically loaded and activated if their is enabled on scene load.	
AreaTargetSample		
GSF-GiT-Veri-Tabani		
VuforiaMars_Images		
VuforiaMars_VuMark		
Vuforia_Marslander_RZI	R_Toy_Advanced	
Vuforia P79 Tov		
Asset Labels		
		•
AssetBuncle None	 None 	1.2

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.



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.

/			
Build Settings			: 🗆 × 🔪 🗧
Scenes In Build			
 SamplesResources/Scenes/0-Splash SamplesResources/Scenes/1-Menu SamplesResources/Scenes/1-Loading SamplesResources/Scenes/3-Cloudf SamplesResources/Scenes/3-Cloudf SamplesResources/Scenes/3-Cloudf SamplesResources/Scenes/3-MidAir SamplesResources/Scenes/3-Intagt 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-AreaTa SamplesResources/Scenes/3-AreaTa SamplesResources/Scenes/3-AreaTa) ecc rTargots Plane argots argotsStanidard argotsAdvoncod rrgots rgots rgots rgots es		0 1 2 3 4 5 6 7 8 9 10 11 12 13 13 14 15
Platform	🖵 Windows, Mac. Linux		Add Open Scenes
🖵 Windows, Mac, Linux 🗳	Target Platform	Windows	*
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	
g-ar Pr4 PS4			

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/2-Loading SamplesResources/Scenes/2-Loading SamplesResources/Scenes/3-CylinderTarg SamplesResources/Scenes/3-CylinderTarg SamplesResources/Scenes/3-GroundPlane	ets		
SamplesResources/Scenes/3-MicAir			0
SamplesResources/Scenes/3-mistantimege SamplesResources/Scenes/3-ModelTarget SamplesResources/Scenes/3-ModelTarget SamplesResources/Scenes/3-MultTargets SamplesResources/Scenes/3-Warks SamplesResources/Scenes/3-AreaTargetC SamplesResources/Scenes/3-AreaTargetC	Targeta SStandard SAdvanced apture		
Distform	- Android		Add Open Scenes
	T ANGION		
🖵 Windows, Mac, Linux 😵	Texture Compression	Use Player Settings	· · · · · ·
Dedicated Server	ETC2 fallback	32-bit	*
	Export Project		
in Android	Build App Bundle (Google Play)		
	Create symbols.zip	Disabled	-
	Run Device	Default device	.▼ Refresh
Universal Windows Platform			Patch And Run
	Patching is disabled for Release built	ds	
iOS ios	Development Build		
- DS4			
134 104			
	Compression Method	L44	· · · ·
May Taylura Siza No Override -			
Texture Compression No Overlide •			
			Pean accus Unity, Suid 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.

Platform			🖷 Android				
Windows, Ma	c, Linux rver	Î	Texture Compression ETC2 fallback Export Project	Use 32-t	Player Settings it	-	•
Android		Ŷ	Symlink Sources Build App Bundle (Google Play) Create symbols zip	Disa	bled		
U WebGL			Run Device	Defa	ult device		✓ Refresh
Universal Win iOS IOS Pra PS4 Prs PS5	dows Platform		Build to Device Patching is disabled for Release build Development Build Autoconnect Profiler Deep Profiling Script Debugging Compression Method		Default device All compatible devices <enter ip=""> Samsung SM N975F (RF8NB069L8L)</enter>	Patch /	snd Run
Asset Import Overr	ides		Compression Mexilou				
Max Texture Size	No Override						
Texture Compression	No Override					Learn about	Unity Build Automation
Player Settings						Bulla	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						×
\leftrightarrow \rightarrow \checkmark \uparrow \square \rightarrow Bu	bilgisayar > Yeni Birim (D:) > UNITY	→ GSF-AR → GSFLogoApp		C GS		م
Düzenle 🔻 Yeni klasör					∎ -	0
🚞 EĞİTİM MATERY, 🛛 🗛	i ^	Değiştirme tarihi	Tür	Boyut		
Cenes Cenes		Aramanızla eşleşe	en öğe yok.			
🦰 Film Rulosu						
🚞 Ekran Görüntüle						
✓ 💻 Bu bilgisayar						
> 🐸 Windows (C:)						
> 👍 Yeni Birim (D:)						
> 🥸 Ağ						
Dosya adı GSFLogo						~
Kayıt türü apk (*.apk)						~
A Klacötleri Gizle				ſ	Kavdet	
A Klasofien Gizle				Ľ	inter pro-	<u> </u>

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 GameCitiject Comp	connit Services. Window Help						
0 // · ·			10	1001062			DO IN DISTING DISTOR
Alter and a second	A 1 Bloom Conce 21 Differe Otherwork 1 22 Differe Otherwork 23 Differe Otherwork 24 Differe Otherwork 25 Differe Otherwork 25 Differe Otherwork 25 Differe Otherwork 25 Differe Otherwork 26 Differe Otherwork 27 Differe Otherwork 28 Differe Otherwork 29 Differe Otherwork 20	The American State of the American Stat			0+ 0 • 6 2 • 7	- Species	(B) Co. Second Co. A result of A result
		Madawa Madawa Madu Linux	Anticide Todum Competentian Todum Competentian Troz Selects Expert Inspect Section Section Antic Acc Bandle (Bange File) Comment periods algo Han Device Exactly Section	Les l'heyer Softings 30-lie Disobert Defaut ravers Facut	* * Facty Are from		
+:			Development Build				30 👂 🗣 🔮 👘 1015
Australiant Australia	recent four afford correct output	ne eran 254 254 - 254 7 Asad lager Domitika Min Tomor for Holberthe r Tenne Convesion <i>Helberthe</i> (Reger Serings)			• Laura dikur liptig thank kanaratan kanas • kurda kanarkan		
Twill completed with a waint of Sund	nacioni en 848 paccente (308162 mp.						
and and and and and and and and and and		📑 🤇 Q.A.I. 🥻 💻 🗗	0 0 🙇 🖻	📼 🖪 🛪 🔊 🍳	🖷 🗿 🏟 🛤 💇 🛢		n ♥ di ₩ 141.404 \$

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.

Account Licenses Crede	ntials Target Manager			
icenses			Get Basic Buy Pre	emium Buy Cloud Add On
earn more about licensing. Treate a license key for your	application.			
Search				
Name	Primary UUID ①	Туре	Status 🗸	Date Modified
SSF-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

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+42D/////AAABme+HCfiClEAyozWwe+wDunMC8gkAYXU6ZuH3w1D745JLpU6H53nAOVg0ZaLauiJn4TTNbYygb+6k05JTp8rD7 ODjyASVPVSRUCfLsk0eGxIvv12jpormKHC49IymMc55F6sVFuXPQ3TK96n8zU71QYwfopnC5jDhQyZD15VrU14Y5UHvjb1YEuDG0W cllGfsxIZMyoqd2prK/CCQpMT1vQEzNqGn4OSMo3hjocrH6on9Mt1cJHdZtaA5E82aLo5r+0DFroIXf1E1JipFvyAS9Dth/GpKIpm sjE/+L26x2upeEuni2sqAkpsIJdMnTEWT49G1hTtq4yybICidd3x1kPoBmgs2MgZegfTqMPh2QfXt
Plan Type: Basic
Status: Active
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**.

Account Licenses Credentials	Target Manager			
Target Manager				Add Database
Use the Target Manager to create a	nd manage databases and	targets.		
Search				
Database	Туре	Targets	Date Modified	
	Device	Э	Feb 21, 2021	
	Device	1	Jun 04, 2023	
	Device	8	Feb 10, 2021	
GSF-GiT-Veri-Tabani	Device	1	Jan 01, 2024	
	Device	5	Feb 16, 2021	
Showing 1-5 of 5 🦟		1		25 per page ♥

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: engine: developer portal	Home Downlo	ads Library Sup	oort Pricing	My Account Log Out
Account Licenses Credentials	Target Manager			
Target Manager > GSF-GIT-VerI-Taba GSF-GIT-VerI-Tabani	Edit Name			
Targets (1)				
Add Target				Download Database (All)
Target Name	Туре	Rating ①	Status 🗸	Date Modified
ि ्रम्ह GSF-LOGO	Image	*****	Active	Jan 01, 2024 19:38

First, we select the Faculty of Engineering logo file and add it (Add).

(Note: Possible errors are also shown)

\Box	\Box	07	
Image	Multi	Cylinder	Object
File:			
Dumlupinar_Üni	versitesi_Mühendis	lik_Fakültesi_Logo.jpg	Browse.
Width:			
Width:			
Width: 1	your torget in scope	unite. The size of the ta	rgot chould be
Width: 1 Enter the width of same scale as your unit scale. The targ	your target in scene augmented virtual c et's height will be ca	units. The size of the ta content. Vuforia uses m lculated when you uplo	rget should be eters as the de ad your image.
Width: 1 Enter the width of y same scale as your unit scale. The targ Name:	your target in scene augmented virtual o et's height will be ca	units. The size of the ta ontent. Vuforia uses m Iculated when you uplo	rget should be eters as the dei ad your image.
Width: 1 Enter the width of same scale as your unit scale. The targ Name: Dumlupinar_Ünive	your target in scene augmented virtual o et's height will be ca ersitesi_Mühendislik.	units. The size of the tai ontent. Vuforia uses m lculated when you uplo Fakültesi_Logo	rget should be eters as the dei ad your image.
Width: 1 Enter the width of same scale as your unit scale. The targ Name: Dumlupinar_Ünive Name must be unit	your target in scene augmented virtual o et's height will be ca ersitesi_Mühendislik que to a database. W	units. The size of the ta content. Vuforia uses m lculated when you uplo _Fakültesi_Logo /hen a target is detected	rget should be eters as the de ad your image. d in your applic

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 (-)
	A A A A A A A A A A A A A A A A A A A

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 Suj	pport Pricing	My Account Log Out
Account Licenses Credentials	Target Manager			
Target Manager > GSF-GiT-Veri-Taba				
GSF-GiT-Veri-Tabani	dit Name			
Targets (2)				
Add Target				Download Database (All)
Target Name	Туре	Rating ①	Status 🗸	Date Modified
DPUMuhendislikLogosu	Image	*****	Active	Mar 06, 2024 17:18
□ १३₽ GSF-LOGO	Image	*****	Active	Jan 01, 2024 19:38

Now, upload the university logo with Add Target.

vuforia: engine developer portal	Home Download	s Library Sup	port Pricing	My Account Log Out
Account Licenses Credentials	Target Manager			
Target Manager > GSF-GiT-Veri-Taba				
GSF-GiT-Veri-Tabani	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
GSF-LOGO	Image	*****	Active	Jan 01, 2024 19:38

In the window that opens, select the DPU logo file and add it (Add).

Add Target

	\bigcirc	$\Box \nabla$	
Image	Multi	Cylinder	Object
File:			
dpu-logo.png			Browse
.jpg or .png (max file	: 2mb)		
Width:			
Ч			
I Enter the width of yo same scale as your a unit scale. The targe	our target in scene u augmented virtual c t's height will be cal	units. The size of the ta ontent. Vuforia uses n culated when you uple	arget should be on th neters as the default bad your image.
Il Enter the width of ye same scale as your a unit scale. The targe Name:	our target in scene u augmented virtual ci t's height will be cal	inits. The size of the ta ontent. Vuforia uses n culated when you uple	arget should be on th heters as the default bad your image.
Il Enter the width of ye same scale as your a unit scale. The targe Name: dpu-logo	our target in scene u augmented virtual c t's height will be cal	inits. The size of the ta ontent. Vuforia uses n culated when you uplo	arget should be on ti reters as the default bad your image.

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.	
.jpg or .png (max file 2mb)	

Try again after making changes within the warnings or taking the shape from another source.

Add Target

Type:

\Box	\Box	$\Box \nabla$	
Image	Multi	Cylinder	Object
File:			
DPULOGO.jpg			Browse
.jpg or .png (max file Width:	e 2mb)		
1			
Enter the width of y same scale as your unit scale. The targe	our target in scene u augmented virtual c et's height will be cal	units. The size of the t ontent. Vuforia uses r culated when you upl	arget should be on the neters as the default oad your image.
Name:			
DPULOGO			

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Cancel	Add
Cancer	Auu

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 ①	Status 🗸	Date Modified
	Image	****	Active	Mar 06, 2024 17:34
DPUMuhendislikLogosu	Image	*****	Active	Mar 06, 2024 17:18

Then, select all these files and click on **Download Database (all)**, where there are 3 images.

Targets (3)				
Add Target				Download Database (3)
Target Name	Туре	Rating ①	Status 🗸	Date Modified
3 selected Delete				
DPULOGO	Image	*****	Active	Mar 06, 2024 17:34
DPUMuhendislikLogosu	Image	*****	Active	Mar 06, 2024 17:18
SEF-LOGO	Image	*****	Active	lan 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) ▼ Filter	s 👻 Clear Fi	lters :	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 Wrone					
3D Game Kit Lite	1.8.2 🛓						
3D Realistic Terrain Free	1.3 ±	Overview Releases mades					
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 土	Purchased Date October 24, 2020					
ARCADE: FREE Racing Car	3.0 🛓	See the Unity Catting Started (https://library.yufaris.com/gatting.c	started lociting_started_usions_				
Avatar Maker Free - 3D avatar from a single selfie	1.3.2 🛓	engine-unity) guide to learn more about working with Vuloria in Ur	iity,				
Base Helicopter Controller	1.1 🛓						
Big Furniture Pack	1.3 🛓	Cet started with augmented reality using Vuforia Engine Intros //e	ngine 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 🛓	Enix, LEGO, Activision and more.					
Bezier Path Creator	1.2 🛓						
Camera Path Creator	1.2 🛓	The Veforia Core Sampler provide complete Unity scenes that the	w how to create 4P 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://deve	loper.vuforia.com/support) for				
Conifers [BOTD]	2.0 土	resources and support.					
Curve Designer	1.0.3 ±						
Easy FPS	1.0 🛓						
echo3D SDK							
Vuforia Core Samples	10.19.3 🕤						
26 of 130	ati 25 *						
Last update Mar 7, 12:28	C +						

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**.



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.

🕏 YuforiaM - 3 ImageTargets - Windows, Ma	e; Linux - Unity 2022.3;1411 = D(111)					- a x	
Fill - A	ni seven ninos nep	18100			10 IS Louis	• Livisit •	
Mitteauty a	1 Alberte					a i	
+ • • • • • • • • • • • • • • • • • • •	S Donne Bonnes Reider He		1	 BrageTarget_Arbrevul Tag Unagote Layer A Transform 			
C2 Integratings, Astronomic C2 Integratings, Astronomic C3 Integratings, Convent C3 Integratings, Chiparn C3 Integratings, Chiparn C3 Conventual			stee	Poston 2 (reson 2 Sale 19 2			
+ 😋 SafafrankAsrager	385			> Di K. Image Target Deheviour (Scalat)			
				Download new Via	203		
				Type Ein Detamos We Instant An	on Satstana forsMoles_mages Osinsut		
					Asid Tenant		
				Add Dautacian Object			
		<i>MT [</i>]	<u> </u>	🗢 🗃 🖉 Detpuit Observer Event Handlar (Scrigt) 🛛 🛛 🖓			
			- 11 Noviganon • Surfaces Show Only Gelected	Constant District Constant Con			
			Show HapMost W Show HapModel + Aperts	Concentration for price of Concentration for an of Concentration for an of Concentration for the one Concentration for the one Concentration			
			Show Path Polygens ar Show Path Geory Hindes Show Treighteoure				
			Show Walk Show Anodance = Ofertades	Gora Imaty			

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.

VotoriaM - 3-ImageTergets - Windo	ins, Mac Linux - Unity 2022.3,14/1* <dx1< th=""><th>1</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>- D X</th></dx1<>	1								- D X
File Edit Assats SameObject Co	mponant Sanicas Window Halp									
🔁 KE + 📥									D Q Lavers	• Laynel •
= Herarchy	a = B Scene									
+* e.c	a moutre menne						< # - # m- +	AllCanors		TI State 14
Contrager angers *								Tito Untacood	· Layer Detput	
👰 ScannLight	100						1	A Trenstorn		0 2 1
A CEL MARK STREET	125								X 0.15 K 0.5	
 A limit inter (boots) 	1007									
) 🖯 CarmonUL										
SatoArcut/onlager	() ()							T . Camera		
									Selectory .	
									Transfer of the local division of the local	
						-	in the	CitingMask	Everything	•
							1000	Projectore	Periodicity	
									Ventical	
						0		Popular View		- 25
						χ -				
				e 112 - 11		\rightarrow		California Ministeri	Tate 5	
						A. The	= Altertates	Vineson Nerr		
						- X	Guttassa			
						X	Show Only Selected			
						X	Show HolghtMesly		One Graphics Settings	
						201	Apenta		None (Render Texture)	A
						\sim	Show Peth Polygone	Dedusion Calling:	2	
						A.	Show Neighbours	MEAN .	CH .	
						1	Drow Wels			
							Circles Internation		Dediet	
							Show Carve Hull			
Bezweit B Constle					à i estate			Autor Literate		
Trans - man				2510 1	19/21 Gene - Deglay		- Stain D- 1x P		(acitil)	
= 🛪 Havoritare 🔹 🔹					6	Image Targets		Downitan	en vutoria engine ser peri re	2003
C, All Model	-		AK AK	- A - A				Monte Planter Data	PILLAND	-
Q. All Prefats			MM	MA				Con	volura e gran com guitalen.	
- 🚔 Assets		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	$-\Psi$	ৰু খ				😤 🖸 init Error Handler	(Script)	9 Ŧ I
Comman							<u> </u>			
The Resources							(1)	🕆 🗋 🖌 Prome Rate Settle	ge (Script)	071

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" er portal		Home	Downloads	Library	Support	Pricing		My Account
Account	Licenses	Credentials	Target Mana	ager					
Licenses > 0	SF-GIT								
	: T								
G2F-G	Edit Nan	ne Delete License	Key						
Liconsoli	(au Lie								
LICENSE	ley Us	age							
Please copy	y the license	e key below into	your app						
ARd+4zD//	///AAABme+H	CfiClEAyozWwe+	wDunMC8gkAYX	U6ZuH3w1D745JLpU	6H53nAOVg0Za	LauiJn4TTNbY	ygb+6k05JTp8rD	17	
OD)YASVPV cllGfsxIZ	'SRUCÍLskoed Myoqd2prK/C	SxIvv12jpormKHC CCQpMT1vQEzNqGn	49IymMc55F6s 40SMo3hjocrH	VFuXPQ3TK96n8zU7. 6on9Mt1cJHd2taA5	lQYwfopnC5jD E82aLo5r+0DF	hQyZD15VrU14 YroLXf1E1JipFy	Y5UHVjblYEuDGO 7yAS9Dth/GpKIp	Copied to dip	board
∋jE/+L26x	2upeEuni2aq	yAkpsIJdMnTEWT4	9G1hTtq4yybI	Cidd3x1kPoBmgs2M	gZegfTqMPh2Q	fXt			
Plan Type:	Basic								
Status: Acti	ive								
Created: Ja	n 01, 2024 1	18:58							
License UU	IID: 6bf6d4b	0d23447c3b64	008cedabe1b	c8					
History:									
License Cre	ated - 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.



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	t Behaviour (Script)	07‡ :
Downl	oad new Vuforia Engine version: 10.20.3	4	Down	load new Vuforia Engine version: 10.2	0.3
Туре	From Database	•]	Туре	From Database	•
Database	EMPTY	-	Database Image Target	GSF-GIT-Veri-Tabani DPULOGO	
Image Target	GSF-GiT-Veri-Tabani			Add Target	
► Advanced	VuforiaMars_Images		Add Occi	usion Object Add Target Repre	sentation
🔻 🗰 🗸 Default Obse	rver Event Handler (Script)	0 7± 1	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-CloudReco SamplesResources/Scenes/3-GroundPlane SamplesResources/Scenes/3-GroundPlane	iets				
SamplesResources/Scenes/3-ImageTarget			٥		
SamplesResources/Scenes/3-ModelTargetsStandard SamplesResources/Scenes/3-ModelTargetsStandard SamplesResources/Scenes/3-MudelTargetsAtvanced SamplesResources/Scenes/3-MultTargets SamplesResources/Scenes/3-AreaTargetSatvances SamplesResources/Scenes/3-AreaTargetCapture SamplesResources/Scenes/3-AreaTargetCapture SamplesResources/Scenes/3-AreaTargetCapture					
Distinguis			Add Open Scenes		
	W Android				
🖳 🤐 Windows, Mac, Linux 🖇 👔	Texture Compression	Use Player Settings	•		
Dedicated Server	ETC2 fallback	32-bit	•		
Dedicated Server	Export Project				
im Androld	Symlink Sources Build App Bundle (Google Play)				
🛱 WebGL	Create symbols.zip	Disabled			
	Run Device	Default device	 Refresh 		
Universal Windows Platform			Patch And Run		
100 100	Patching is disabled for Release built	ds	3.		
105 105	Development Build				
Pra PS4					
PUTE PS5	Script Debugging				
	Compression Method	144T			
Asset import Overrides					
Testure Composition No Overlide -			10 11 000000000000 Ma		
reade compression incovernue			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.

Platform		🖷 Android		
Windows, Ma	c, Linux rver	Texture Compression ETC2 fallback Export Project Symlink Sources	Use Player Settings 32-bit	
WebGL		Create symbols.zip Run Device	Disabled Default device	▼ Refresh
Universal Win iOS IOS Pra PS4 Prs PS5	dows Platform	Build to Device Patching is disabled for Release build Development Build Autoconnect Profiler Deep Profiling Script Debugging Compression Method	Default device All compatible devices <enter ip=""> Samsung SM N975F (RF8NB069L8L) LZ4</enter>	
Asset Import Overn Max Texture Size Texture Compression Player Settings	ides No Override No Override		<u>Learn</u> Bu	about Unity Build Automation ild 🔹 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		l i				: 🗖
Adaptive Performance	Player					0 ‡
Editor	Company Name	D	PU			
Graphics	Product Name	v	uforiaM			
Input Manager	Version	1	0.19.3			
Memory Settings						1
Physics	Default Icon					(C) MARS
Physics 2D						
Player						" Select
Preset Manager Quality	Default Cursor					
Scene Template						(Texture 2D)
Script Execution Order						Select
V Services	Cursor Hotspot		0		Y 0	
Ads Analytics - Gaming Services					(575 (14)	
Environments	<u> </u>			•	9	4
Tags and Layers	Settings for Android					
Settings	⊫ Icon					
Time Timeline	Resolution and Presentation					
UI Builder Version Control	🕨 Splash Image					
Visual Scripting						
XR Plugin Management	Rendering					
	Color Space*		Gamma			
	MSAA Fallback		Downgrade			
	Auto Graphics API					
	Graphics APIs					
	= OpenGLES3					
						+
	Require ES3.1					
	Require ES3.1+AEP					
	Require ES3.2					
	Color Gamut*					
ort overrides for local development.	= sRGB					
an speed up asset imports and platforr	n					+ - [_]

In the **Identification>Package Name** section, change the package name to match the definition above; **com.DPU.VuforiaM**

Project Settings			÷□×
		٩	
Adaptive Performance	Player		@ ≓ :
Audio Editor	Load/Store Action Debug Mode		-
Graphics Input Manager Memory Settings Package Manager Physics Physics 2D Player Preset Manager Quality Scene Template Script Execution Order Script Execution Order Services Ads	Vulkan Settings SRGB Write Mode* Number of swapchain buffers* Acquire swapchain image late as possible* Recycle command buffers* Apply display rotation during rendering Identification Override Default Package Name Package Name Version* Bundle Version Code	3 Com.DPU.VuforiaM 10.19.3 10102	
Analytics - Gaming Services Environments	Bundle Version Code	10193	
Tags and Layers	Tarret ADL aval	Automatic (biobact installed)	
▼ TextMesh Pro	Talget APT Level	Automatic (ingriest installeu)	
Settings	Configuration		
Timeline	Scripting Backend	IL2CPP	
UI Builder	Api Compatibility Level*	.NE1 Framework	
Version Control	IL2CPP Code Generation	Faster runtime	
Visual Scripting VP Plugin Management	C++ Compiler Configuration	Release	
AR Flugin 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 Armv9 Security Features for Arm64		
	Split APKs by target architecture		
	Target Devices	All Devices	
	Install Location	Prefer External	
	Internet Access		
	Write Permission	Internal	

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		٩
Düzenle 🕶 Veni klasör				=	- 💿
plataform-for-ja	Ad	Değiştirme tarihi	Tür E	loyut	
A EĞITIM MATEI	Assets	7.03.2024 17:26	Dosya klasõrü		
SOUTCE	C Library	7.03.2024 17:30	Dosya klasörü		
	Logs	7.03.2024 17:23	Dosya klasõrü		
	Packages	7.03,2024 12:47	Dosya klasõrü		
	ProjectSettings	7.03.2024 13:29	Dosya klasóru		
V 📮 Bu bilgisayar	C 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	CarloriaM	7.03.2024 17:33	Dosya klasörü		
Dosya adı: Vufori	aM				Ý
Kəyıt türü: 🛛 əpk (*.	apk)				: 4
∧ Klasörleri Gizle				Kaydet	ptal

😨 Build Android						×
← → ∽ ↑ 🧰 Bu bilgisayar	r > Yeni Birim (D:) > UNITY >	VuforiaM > VuforiaM		C Vu		ara 🔎
Düzenle 🔻 Yeni klasör						E • 👩
🛅 plataform-for-ja 🛛 🗛 Ad		Değiştirme tarihi	Tür	Boyut		
🛅 A EĞİTİM MATER		Aramanızla eşleş	şen öğe yok.			
source						
i ke						
✓ 💻 Bu bilgisayar						
> 🔛 Windows (C:)						
> 🖽 Yeni Birim (D:)						
> 💱 Ağ						
Dosya adı: VuforiaM						Ŷ
Kayıt tür <mark>ü:</mark> apk (*.apk)						
∧ Klasörleri Gizle				C	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.

W GSF-Ah - 2 mage arges - Annual - Un	th 200201401 <0x11+						
File Edit Assets Game/Object Compo	next Services Window Help		11.00	Concernent of the Concernent o			station sealer instantions of the search of
0.n · .			18	SINGLES			D. St. Layers D. Corral St. +
The Harranthy	A 1 R Scene - Come					🚺 🔿 inspessor	A 1
	Construction Const	Constant in the second	rgetis me me anti- netication participation		0* 0 1 4 4 4	2 M 1 9 1	
		Pisifia Windows, Mac, Snuv Ecidezed Server Mark Antroid Marked Markovski Workowski Workows Markovski Workowski W	Andrad Tokun Compression Tokun Compression Totalance Totalance Totalance Synthematic Security Totalance T	Las Payar Soffrey Sjölit Dhader Dalar Genes Daar	Antenet Service		
Project B Console - Unity Versit							
Access Control Contro Control Control Control Control Control Control Control	noden twe allow anyor building in tweath: growth	August 204 August 204 August 204 August David Overskaa Mais Ingras Doverskaa Mais Ingras Doverskaa Mais Ingras Doverskaa August Station Haldweite • Station Station			Laan deg 1913 Nui Austein Laan deg 1913 Nui Austein Hurd C. K. Ruidander		2) \$ \$ \$ \$
Test completed with a sessit of Success	esen in \$48 pacemic (\$44162mp)						a 🛎 🔿
- 3% Hukey		📲 (Q.44) 🤷 🖉 🖉	s 🧕 🙋 🙇 🖻	📼 🚺 🔌 🕺 🔅	- 🖷 🧶 😫 🗮 🖉 🛢		n ♥ d) ₩ 12241

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	Ψ.
Track 0 [2 ch]	2	
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: engine: developer portal	Home Download	ls Library	Support	Pricing	My Account Log Out
Account Licenses Credenti	als Target Manager				
Licenses 9 GSF-GIT GSF-GIT Edit Name Delete Lic	rense Key				
License Key Usage Please copy the license key below	into your app]
ARd+izD////AAABmc+HCfiClEAyor ODjyASVPVSRUCfLsk0eGxImc2jpoz clisfsxIZMycqd2pxK/ccgBulyQki sjE/+L26x2upeEuni2sqAkpsIJdMn1	WWeiteDunHCSgkAYXU62uH3wlD745J MMKHC49IymHcS5F6eVFuXPQ3TK96H3 MUGch4OSkoShjeczHeonMKlc/Hdzt TEWT49GlhTtg4yybICidd3xlkPoZmga	apU6H53nAOVg0Za 2071QYwfopnCSjD AA5E5ZsLo5r+ODF s2MgZegfTqMFh2Q	LauiJn4TTNbY: hQy2D15VrU14) roLXf1E1J1pFv fXt	/db+6k05JTp8rD7 /SUHvjb1YEuDG0W /yAS9Dth/GpKIpm	Copied to clipboard
Plan Type: Basic					
Created: Jan 01 2024 18:58					
License UUID: 6bf6d4b0d23447c	3b64008cedabe1bc8				
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.



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/I-Splash SamplesResources/Scenes/I-Menu SamplesResources/Scenes/I-Loading SamplesResources/Scenes/I-Loading SamplesResources/Scenes/I-ChodRaco SamplesResources/Scenes/I-GroundPlane SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-MidAr SamplesResources/Scenes/I-Area Targets SamplesResources/Scenes/I-Area Targets SamplesResources/Scenes/I-Area Targets	nets si xTargets sStendard sdvanced i apturc		
Platform	iii Android		Add Open Scenes
	T Sharow		
windows, Mac, Linux	Texture Compression	Use Player Settings	•
Dedicated Server	ETG2 Infibility	32-DK	
	Synvink Sources		
Android 🕸	Build App Bundle (Google Play)		
WebGL	Create symbols.zip	Disabled	
U.S. S. S. S. S. S. S. S. S. S. S. S. S.		Default device	✓ Refresh
Liniversal Windows Platform			
100 100			
105 105	Development Build		
Pira PS4			
ers PS5	Compression Method	LZ4	
Asset Import Overrides			
Max Texture Size No Override -			
Texture Compression No Override 🔹			Learn about Unity Build Automation
			Duild - Duild And Dun
Player Setongs.			Build * Build And Run

Connect the **mobile** device and check.

Build Settings			: 🗆 :
Scenes In Build			
SamplesResources/Scenes/0-Splash SamplesResources/Scenes/1-Menu SamplesResources/Scenes/2-Loading SamplesResources/Scenes/3-CloudReco SamplesResources/Scenes/3-CloudReco SamplesResources/Scenes/3-Middar SamplesResources/Scenes/3-Middar SamplesResources/Scenes/3-Instantmage SamplesResources/Scenes/3-Middar SamplesResources/Scenes/3-Middar SamplesResources/Scenes/3-Middar SamplesResources/Scenes/3-Middar SamplesResources/Scenes/3-Middar SamplesResources/Scenes/3-Middar SamplesResources/Scenes/3-Middar SamplesResources/Scenes/3-Middar SamplesResources/Scenes/3-AreaTargets SamplesResources/Scenes/3-AreaTargets SamplesResources/Scenes/3-AreaTargets	ets 5 Targets SStandard SAdvanced apture		
Platform	🖷 Android		Add Open Scenes
Windows Mar Linux	Taylura Cemerateion	Lise Player Settings	
	ETC2 fallback	32-bit	*
Dedicated Server	Export Project		
📫 Android 😌	Symfinic Sources Build App Bundle (Google Play)		
WebGI	Create symbols.zip	Disabled	
	Run Device	Default device	✓ Retresh
Universal Windows Platform		 Default device 	Patch And Run
102.05	Patching is disabled for Release build	All compatible devices	
103 103	Development Build	<enter ip=""></enter>	
Pira PS4		Samsung SM N975F (RF8NB069L8L)	
ers PS5			
	Compression Method	LZ4	
Asset Import Overrides			
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			10
Adaptive Performance	Player		0 ≠
Editor	Company Name	DPU.	
Graphics		VuforiaGP	
Input Manager Memory Settings		10.19.3	
Package Manager Physics Physics 2D	Default icon		
Preset Manager			Select
Quality Scene Template	Default Cursor		None (Texture 20)
F Services			Subot

🗘 Project Settings			10
Project Settings Adaptive Performance Aud b Editor Graphics Input Manager Memory Settings Package Manager Physics Physics 2D Player Preset Manager Cuality Secue Template Sortict Execution Order Services Ads Analytics - Gaming Service Environments Tags and Layers TextMesh Pro Settings Time TimeIne UltBuilder	Player Graphics Jobs (Experimental) Textue compression format Nermal Map Encoding Lightmap Encoding HDR Cutempa Encoding Lightmap Streaming Piterby Frame Training Stats OpenGL: Profiler GPU Recorders CopenGL: Profiler GPU Recorders Milow HDR Display Output* Use HDR Display Output* Use HDR Display Output* Use HDR Display Output* Use HDR Display Output* Load/Store Action Debug Mode Virkan Settings	C S S S S S S S S S S S S S S S S S S S	•
Version Control Visual Scripting XR Plugin Management	Number of swapchain buffers* Acquire swapchain image late as possibl Recycle command buffers* Apply display totation during rendering identification Override Default Package Name		
	Package Name Version* Bundla Version Code Minimum APILevel Target APILevel Configuration	com.DPU.VuforiaGP 10.19.3 10193 Android 8.0 'Oreo' (API lavel 26) Automatic (highest installed)	

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					×
← → ~ ↑ □	> Bu bilgisayar > Yeni Birim (D:) > UNITY >	VuforiaM > VuforiaM		C VuforiaM klasöründe ara	م
Düzenle 🔻 🛛 Yeni klasör				∎ •	(2)
🛅 A EĞİTİM MATEF	Ad	Değiştirme tarihi	Tūr	Boyut	
🚞 source	📄 VuforiaM.apk	7.03.2024 17:43	APK Dosyası	110.031 KB	
C KE					
🚞 Ders Notları - Ka					
🗸 💻 Bu bilgisayar					
> 불 Windows (C:)					
> 🖽 Yeni Birim (D:)					
> 🛬 Ağ					
Damasta Wefer	-C0 ask				1000
Kavit türü: ank (*	aoki				~
					(*
∧ Klasörleri Gizle				Kaydet İptal	

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	



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.



Here, the folder was created by naming it **Textures**. Select it with **Select Folder**.

zenle 🔻 Yeni klasör					🗐 🔹 🧕
👝 OneDrive - Personal	^	Dedictions table	The P	977 777	
🗧 Belgeler	🚞 Dokular	1.05.2024 20.02	Docya klasörű		
🛓 Downloads					
💴 Masaüstü					
🙆 Müzikler					
🔀 Resimler					
Ekran Görüntüleri					
🧮 Film Rulosu					
🔽 Videolar					
🛅 Kaan Erarslan					
🚞 Kaan Erarslan 🐖 Bu bilgisoyar					

If there is a need for correction in texture assignments, it warns. Fix this by saying **Fix Now**.

NormalMap settings	
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 Ignore	

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.



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.



Here, create an animator controller with Create>Animator Controller and name it WorkerController.

Project 🛛 🖻 Console		
+*		q
🕞 ★ Favorites	Assets > Character Animator	
🔻 🗁 Assets	- WorkerController	
🖿 Character Animator		
Scenes Worker Ch17_nonPBR Scenes		
P De Packages		

When this process is done, **three states** are created in the **editor** in the **Animator** tab, **including Any State**, **Entry** and **Exit** blocks.

# Accorbin - ExepteScene - Modo	we, March	ma - Unity 2612:3-18114 (0311)	×					(*) O	* ×
File Edit Assets GerneOkjeut C	of pener	t Seniers Window Holp							
0 III * 🔺								CO: CA: LANNESS CA	
TE Harantyn, 3 1 + 1 + 1 + 1 • ∰ Hangatosaw 3 Devotrei Lyn 3 Devotrei Lyn • ∰ Chloretta 9 Devotrei	а I И 1	E Score - Province Logies (Frankling) - A No	tan Layer			A 1	O wagestar Worker Carender (Antender Controller)	(51) Openi	
				Ditry					

First, make animations of the transition from the idle position to walking and running states.

Drag and drop the Idle animation 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**.

10 Miles 100				1011	102.40		190 10% America	of Lond .
					NUTL.		news reservations	
The House and App	= :	Stete 20				 O http://www.		a:(
		Loyers Pattern	15	a Lager - Hunsher		Illens Tree		(Q))
 SamplaScene* Max Carera 						Disent Type		
Directoral Light		100 20					Dend	
								No
								-
					Sendine			

In this case, the adjustment line is also displayed in the **Blend Tree** box.



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 - Wind	ows, Mac, Li	nux - Unity 2022.3.14f1 < DX11							
File Edit Assets GameObject	Component	Services Window Help							
😑 KE 🔹 📥					E II FI				
= Hierarchy		# Scene > Animator							
4- • •		Levers Parameters of	🔊 Base Layer						
♥ ✿ SampleScene					Folder				
Main Comen Otrectional Light Otrectional Light Otrectional Light Otrectional Light Otrectional Light Otrectional Light		Base Loyer 🌼	Any State		C# Script				
					20	2			
					Visuel Scripting				
					Shader	- 2			
			Fatra		Shader Variant Collection			Fuit	
					Testing				
					Playables				
					Assembly Definition				
					Assembly Definition Reference				
					TextMeshPro	- C - C - C - C - C - C - C - C - C - C			
			Create	> Text >	Hareket				
			Show in Explorer		Scene				
			Open		Scene Template				
			Delete		Science Terreplate Promi Science				
			Rename		Scene Template Pipeline Protab				
			Copy Path Alt+1	Ctrl+C					
			Open Scene Additive		Prefab Variant				
			View In Package Manager		Audio Mixer				
			Import New Asset		Material				
			Import Package	>	Material Genant				
			Export Package		Lens Flare				
			Find References In Scene		Render Texture				
			Select Dependencies		Lightmap Parameters				
					Lighting Settings				
			Refrech	Ctrl+R	Custom Render Texture			Conception of the second second	Compared and the sector
B Project S Consola			Reimport			- 1		- carved Annual Come	1
+			Reimport All		Animator Controller Animation		 Display 1 - Free Aspect 	• Scale • 1+ 1	lay Maximizett =
> W Favoritos		Assets Character Animetor Scenes	Extract Frien Prefait		Animator Override Controller				
 Assets Character Animator Scenes 			Update UXML Schema		Avatar Mask				
		Worker Ch17_hor/P88			Timeline				
Worker Ch17_comPBR		New Letter	Generate Lighting Ctrl+S	Shift+L	Sincel				
> En Packages			Open C# Project		Juliu				
			View in Import Activity Window		Physic Material			1	
			Properties	Alt+P	GUI Skin Custom Font				

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); }
  // jumping
  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);
```

331

```
//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.

Aerial-Craft (2022). Rawang Mine Quarry, Malaysia. Retrieved from <u>https://sketchfab.com/3d-models/rawang-quarry-2a-11jan17-malaysia-f3eae47abe694f36a9a98578315acc1e</u>

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.

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. https://www.apple.com/apple-vision-pro/

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> 46466ec0558945e9aac9dad15aaeb9f3.

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-Longwallcoal-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. https://www.btelligent.com/en/portfolio/industry-40/

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 https://sketchfab.com/3d-models/ppc-kgale-2016-09-08-677a883d45ea48fbb974a0470f98a2ed

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). <u>https://rigmodels.com</u>

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. https://www.unrealengine.com/en-US/

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.