



Unity

Öğr. Gör. Gözde Mihran ALTINSOY

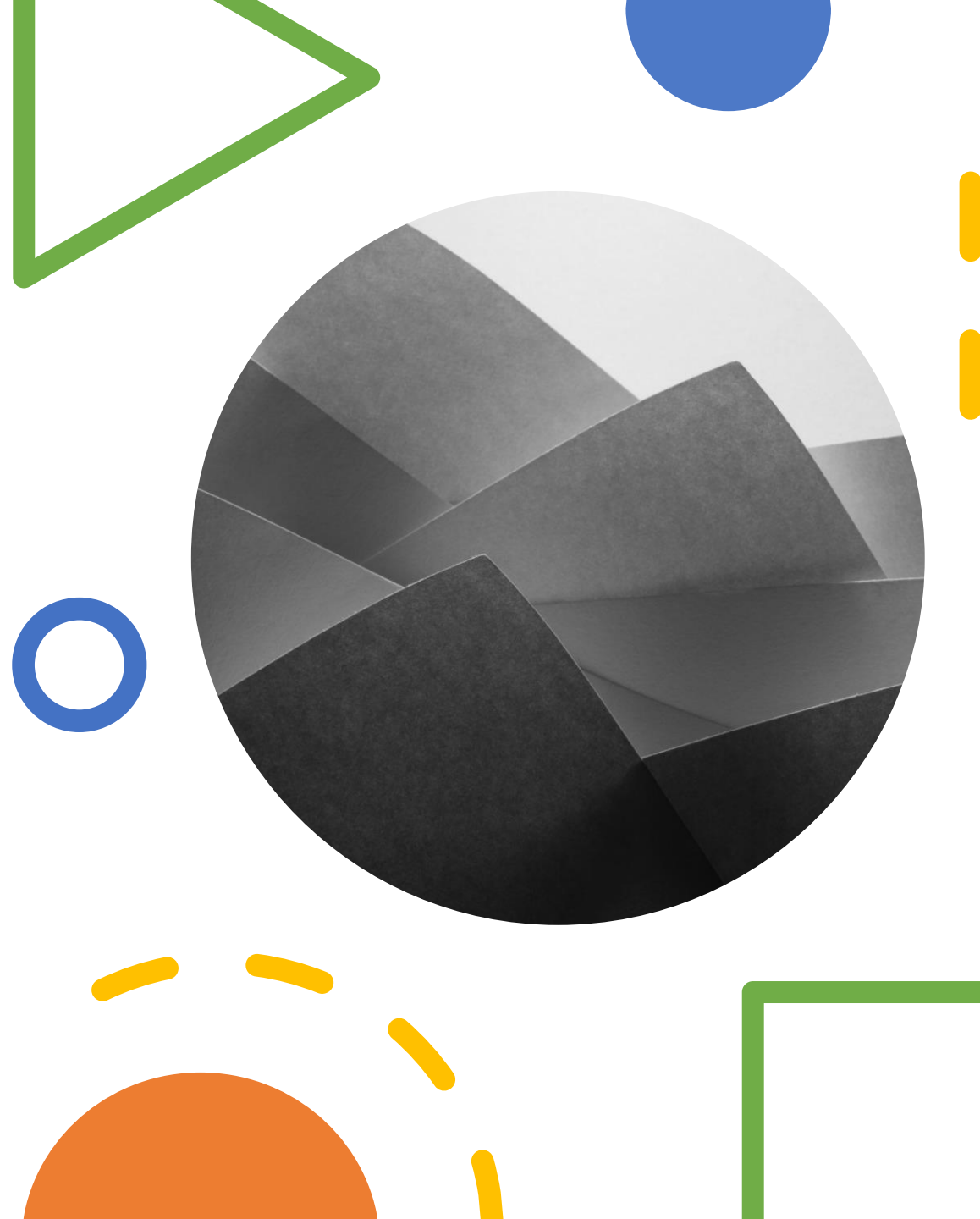
Oyun Motoru Nedir?



Oyun motoru sadece bir oyuna özel olmayan temel işlemleri daha hızlı yapabilmemizi sağlayan uygulamalardır.



Bu sayede temel işlemleri oyun motoruna bırakarak biz kendi oyunumuza özel fonksiyonlara ve mekaniklere daha fazla vakit ayırabiliriz.



Neden Oyun Motoru Kullanıyoruz?



Oyun Teknolojileri



Sesler
Grafikler



Oyuna Özel Kodlar



Oyun motoru

Grafik
Fizik
Ağ
Girdiler



İşletim Sistemi

Donanım ile etkileşim
Donanım kontrolü



Donanım

İşlemci
Ekran Kartı
Bellek vb.



Unity Avantajları



Unity ile her türden oyun geliştirebiliriz.



Fizik motoru gibi gelişmiş özelliklere sahiptir.



Öğrenmesi kolay, uzmanlaşması ise vakit alabilir.



Gelişmiş dokümantasyon ve yaygın kullanıma sahiptir.

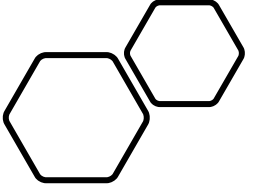


Çoklu (cross) platform desteği vardır.



Unity ücretsiz versiyonu vardır yani lisans ücreti ödemek zorunda değiliz.

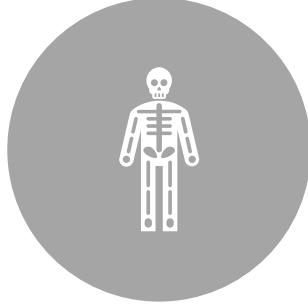




Unity Dezavantajları



GÖRSEL MATERYALLER
DÜZENLEMELERİNİN UNITY
İÇİNDE YAPILAMAMASIDIR.



3D MODELLEME
OLMAMASIDIR.



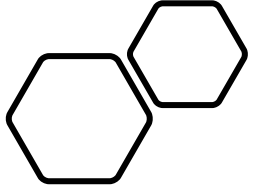
SES DOSYALARININ
DÜZENLEMESİNİN
YAPILAMAMASIDIR.



İHTİYACIMIZ OLMAYAN
KAYNAKLARIN PROJEDE YER
ALMASIDIR.



Kavramlar ve Araçlar



Kavramlar ve Araçlar



Visual Studio ve IDE



.NET Framework

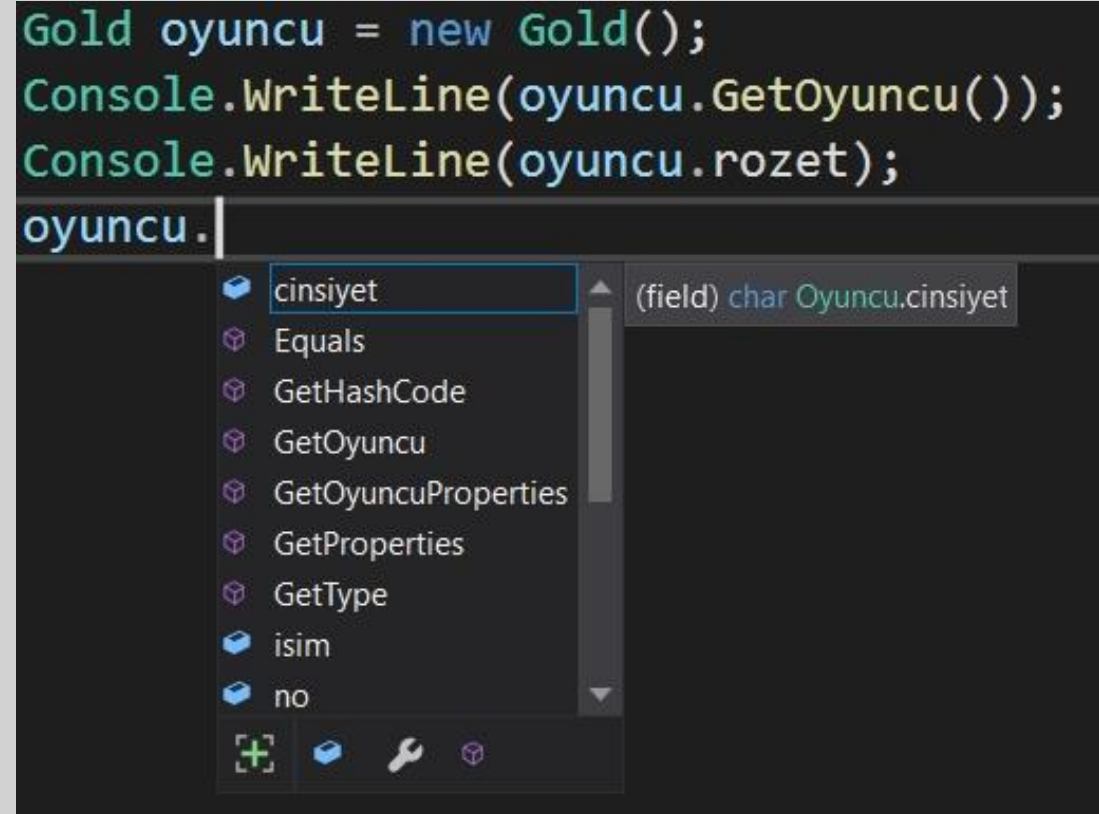


C#

Visual Studio ve IDE

- IDE = Integrated Development Environment (Tümleşik Geliştirme Ortamı)
- Visual Studio bir IDE'dir.
- Bir geliştirme yapmak için Text Editor'lerini de kullanabiliriz. (Ör; Visual Studio Code)
- IDE Avantajları
 - Intellisense (Otomatik kod tamamlama özelliği)

```
Gold oyuncu = new Gold();
Console.WriteLine(oyuncu.GetOyuncu());
Console.WriteLine(oyuncu.rozet);
oyuncu.
```

A screenshot of the Visual Studio IDE showing a code completion menu. The code in the background is: `Gold oyuncu = new Gold();`, `Console.WriteLine(oyuncu.GetOyuncu());`, `Console.WriteLine(oyuncu.rozet);`, and `oyuncu.`. The completion menu is open below `oyuncu.` and lists several options: `cinsiyet` (highlighted), `Equals`, `GetHashCode`, `GetOyuncu`, `GetOyuncuProperties`, `GetProperties`, `GetType`, `isim`, and `no`. To the right of the menu, a tooltip shows the type of the selected property: `(field) char Oyuncu.cinsiyet`. At the bottom of the menu, there are icons for adding new items, a cube, a wrench, and a cube.

.NET Framework

- Framework (Yazılım iskeleti)
 - İçinde kütüphaneler barındırır.
 - İçine yeni kütüphaneler eklenebilir.
- .NET = Framework
- C# = Yazılım Dili



Neden C#?

- Unity .NET Framework ile çalışıyor.
- Biz bu framework ile C# dilinde iletişime geçiyoruz.
- Unity yeni sürümlerinde ~~JavaScript~~ ve ~~Boo~~ dillerine destek vermeyi kesti.
- Unity'de geliştirme sadece C# dili ile yapılıyor.



C# Avantajları

- Sadece oyun geliştirmek için değil; web sitesi, masaüstü uygulama, mobil uygulama gibi bir çok platformda çalışabilecek uygulamalar geliştirmek için kullanılabilir.
- En popüler geliştirme dilleri arasında yer almaktadır.



Microsoft Docs

docs.microsoft.com son kullanıcıların, geliştiricilerin ve BT uzmanlarının Microsoft belgelerine ulaşabileceği merkezdir. Hızlı başlangıçlarımızı, öğreticilerimizi, API başvurularımızı ve kod örneklerimizi inceleyin.

 Windows	 Azure	 Visual Studio	Office	Microsoft 365	.NET
ASP.NET	SQL	Enterprise Mobility + Security	 Microsoft Surface	Dynamics 365	 Xamarin
Azure DevOps	System Center	 PowerShell	Microsoft Graph	Microsoft Education	Gaming

Docs Dizini

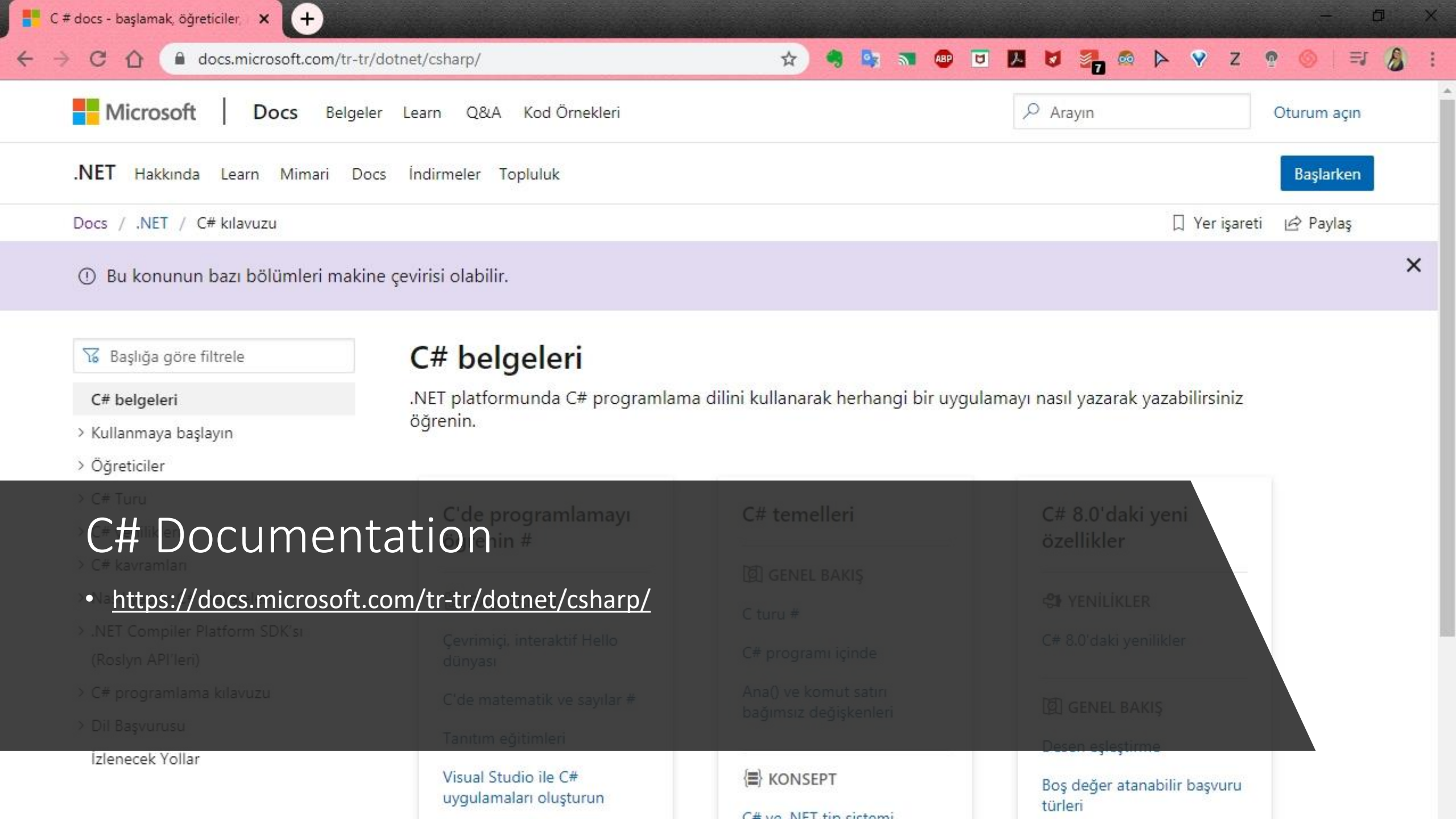
.NET

Internet Information Services

Office 365

Microsoft Docs

<https://docs.microsoft.com/tr-tr/>

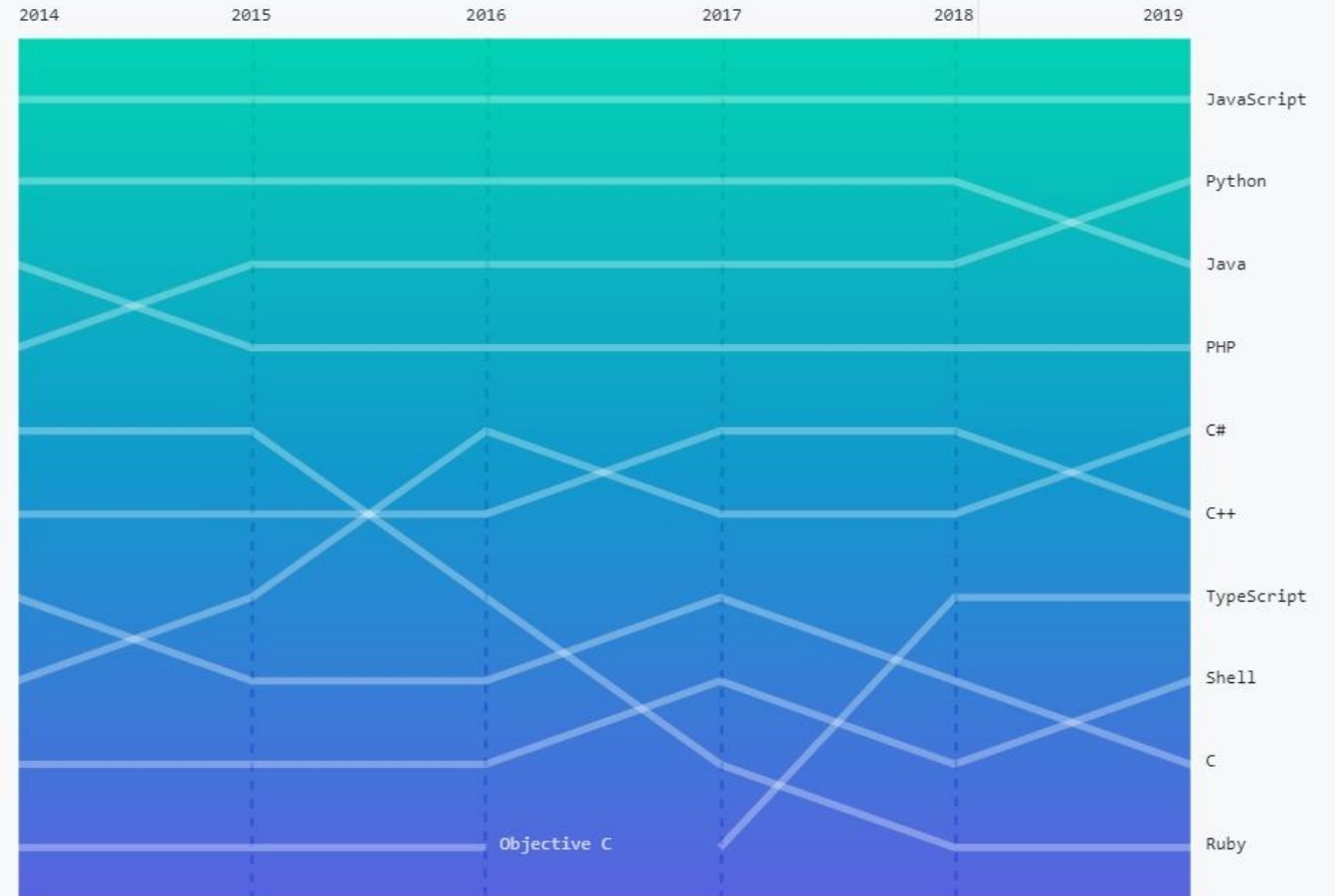


Top languages

Top languages over time

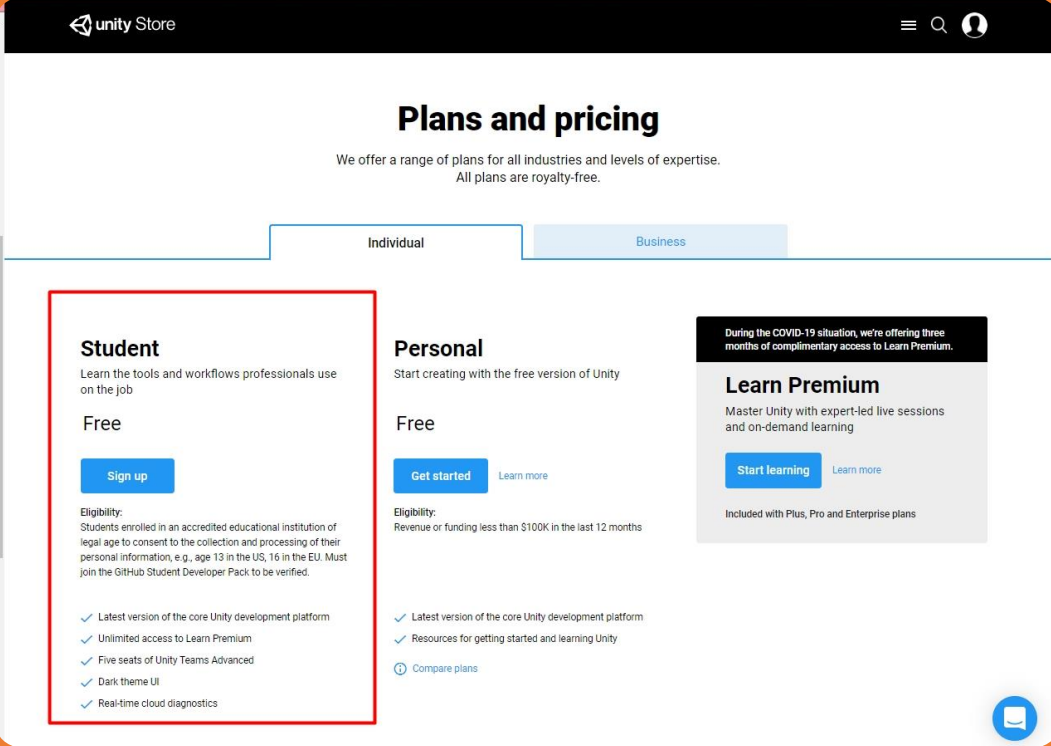
This year, C# and Shell climbed the list. And for the first time, Python outranked Java as the second most popular language on GitHub by repository contributors.*

In the last year, developers collaborated in more than 370 primary languages on GitHub.



Unity Kurulumu

<https://store.unity.com/#plans-individual>



unity Store

Plans and pricing

We offer a range of plans for all industries and levels of expertise.
All plans are royalty-free.

Individual Business

Student

Learn the tools and workflows professionals use on the job

Free

[Sign up](#)

Eligibility:
Students enrolled in an accredited educational institution of legal age to consent to the collection and processing of their personal information, e.g., age 13 in the US, 16 in the EU. Must join the GitHub Student Developer Pack to be verified.

- ✓ Latest version of the core Unity development platform
- ✓ Unlimited access to Learn Premium
- ✓ Five seats of Unity Teams Advanced
- ✓ Dark theme UI
- ✓ Real-time cloud diagnostics

Personal

Start creating with the free version of Unity

Free

[Get started](#) [Learn more](#)

Eligibility:
Revenue or funding less than \$100K in the last 12 months

- ✓ Latest version of the core Unity development platform
- ✓ Resources for getting started and learning Unity

[Compare plans](#)

During the COVID-19 situation, we're offering three months of complimentary access to Learn Premium.

Learn Premium

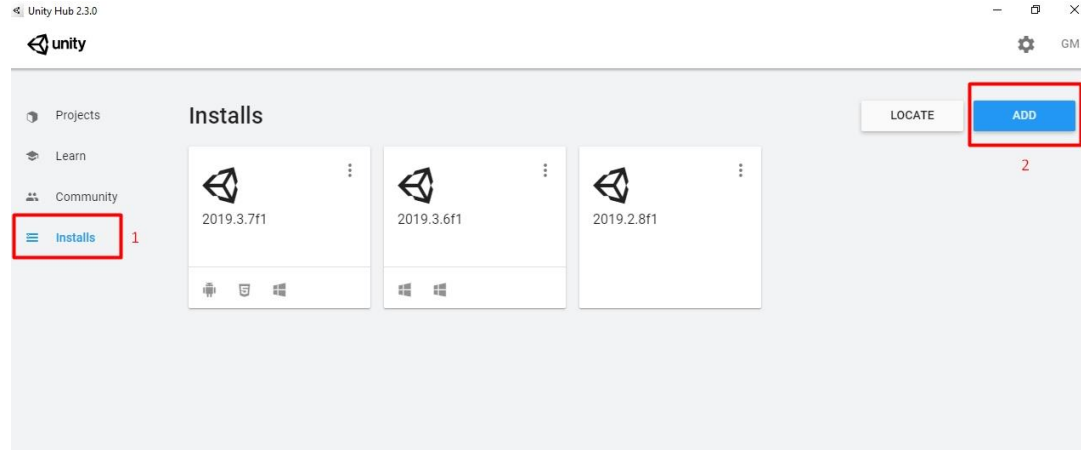
Master Unity with expert-led live sessions and on-demand learning

[Start learning](#) [Learn more](#)

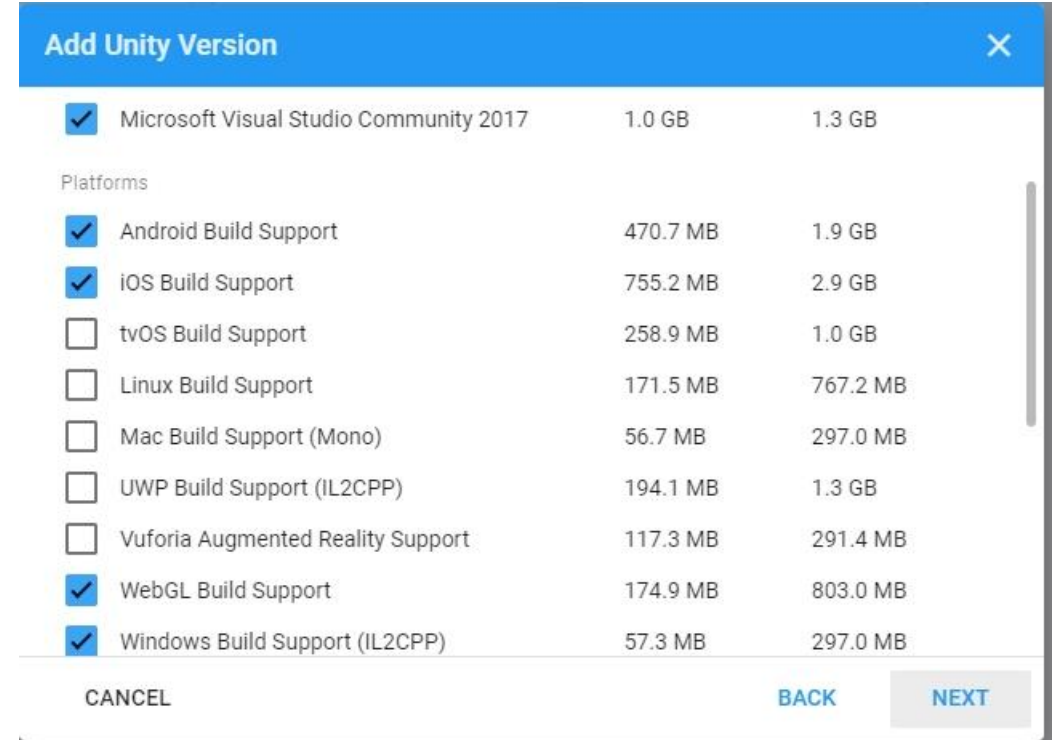
Included with Plus, Pro and Enterprise plans

Unity Hub

Yüklü Versiyonlar



İsteğe Bağlı Seçimler



Konsol Nedir?

- Konsol grafiksel olmayan bir kullanıcı arayüzü ile, kullanıcıdan bilgi alma ya da kullanıcıya bilgi vermek amacıyla kullanılan siyah bir ekrandır.

Problem Tanımı

Yapılacak bir işin veya çözülecek bir sorunun olabildiğince detaylı düşünülüp, analiz edilmesidir.

Örnek:



Problem tanımı 1:

Oyunun ana karakterini hareket ettirmek.



Problem tanımı 2:

Oyunun ana karakterini

- Klavyenin yön tuşlarını kullanarak,
- Yön tuşlarının belirttiği yöne paralel olacak şekilde,
- Saniyede X birim hızlandırarak,
- oyuncu tarafından kontrol edilmesini sağlayacak yazılımın geliştirilmesi.

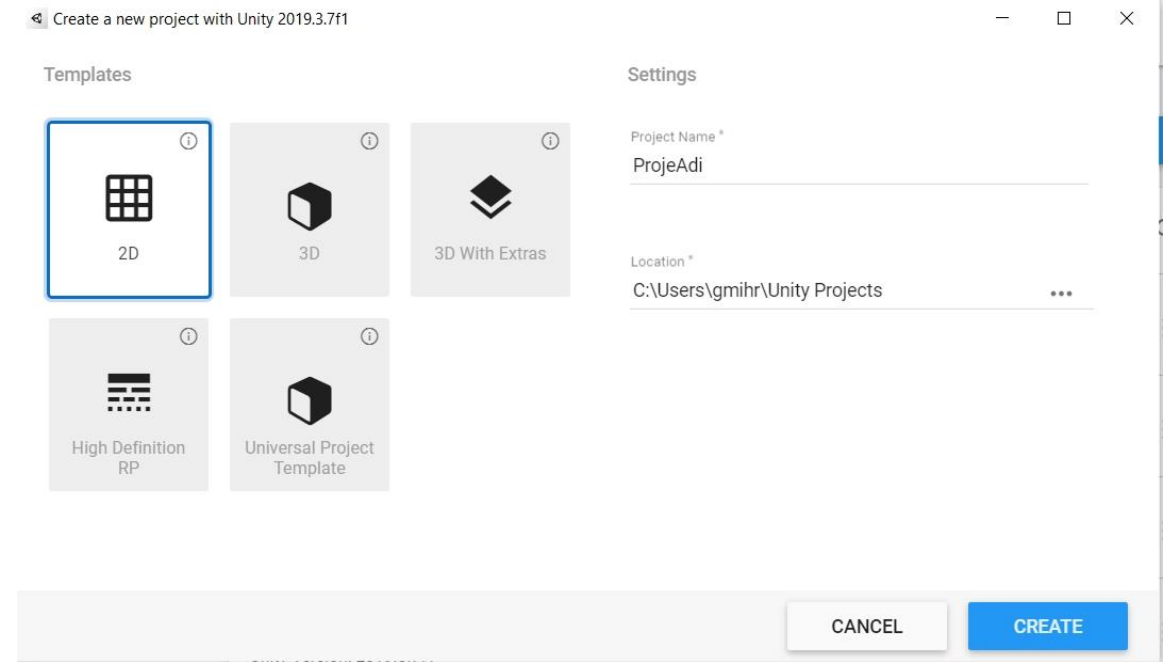
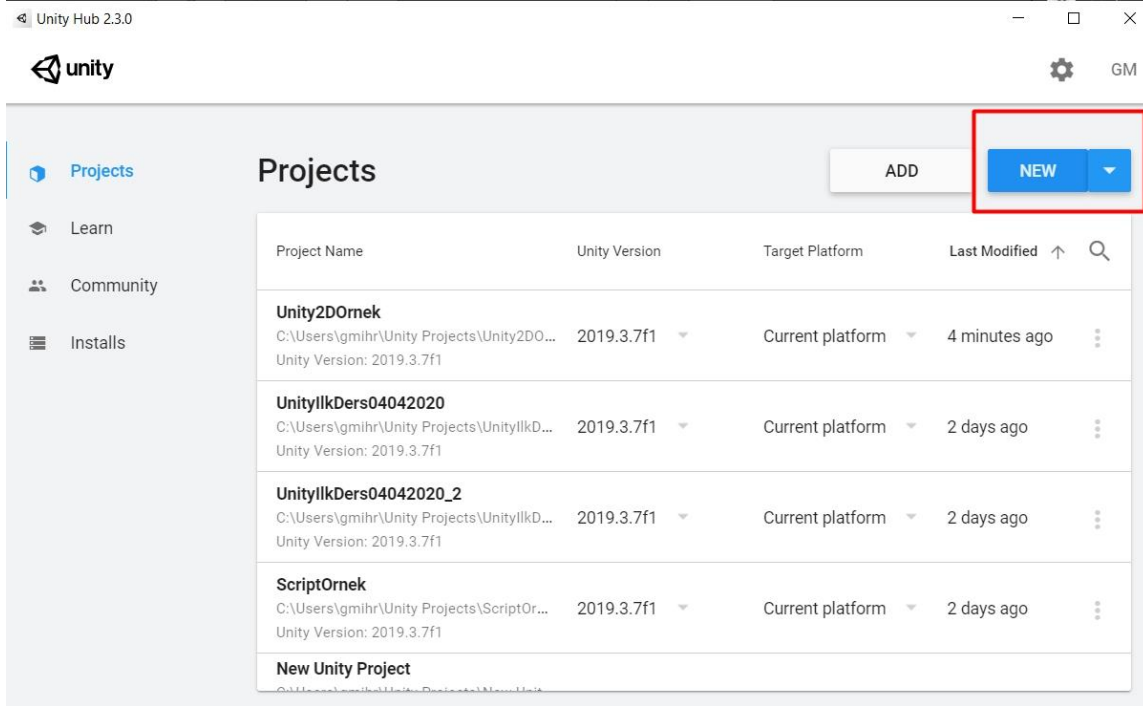


Log

Hata takibi

Dijital hareketler

Unity Yeni Proje Açmak



Hierarchy

All

SampleScene

Main Camera

Scene

Shaded

2D

Game

Asset Store

0

Gizmos

All

Create

Show in Explorer

Open

Delete

Rename

Copy Path

Alt+Ctrl+C

Open Scene Additive

View in Package Manager

Import New Asset...

Import Package

Export Package...

Find References In Scene

Select Dependencies

Refresh

Ctrl+R

Reimport

Reimport All

Extract From Prefab

Run API Updater...

Update UIElements Schema

Open C# Project

Folder

C# Script

Shader

Testing

Playables

Assembly Definition

Assembly Definition Reference

TextMeshPro

Scene

Prefab Variant

Audio Mixer

Material

Lens Flare

Render Texture

Lightmap Parameters

Custom Render Texture

Sprite Atlas

Sprite Library Asset (Experimental)

Sprites

Tile

Sprite Shape Profile

Animator Controller

Animation

Animator Override Controller

Avatar Mask

Timeline

Signal

Physic Material

Physics Material 2D

GUI Skin

Custom Font

UIElements

Legacy

Project

Console

Assets > Scripts

Script1

Assets

Scenes

Scripts

Packages

Assets/Scripts

AssetBundle

None

Asset Labels

AssetBundle

None

AssetBundle

None

AssetBundle

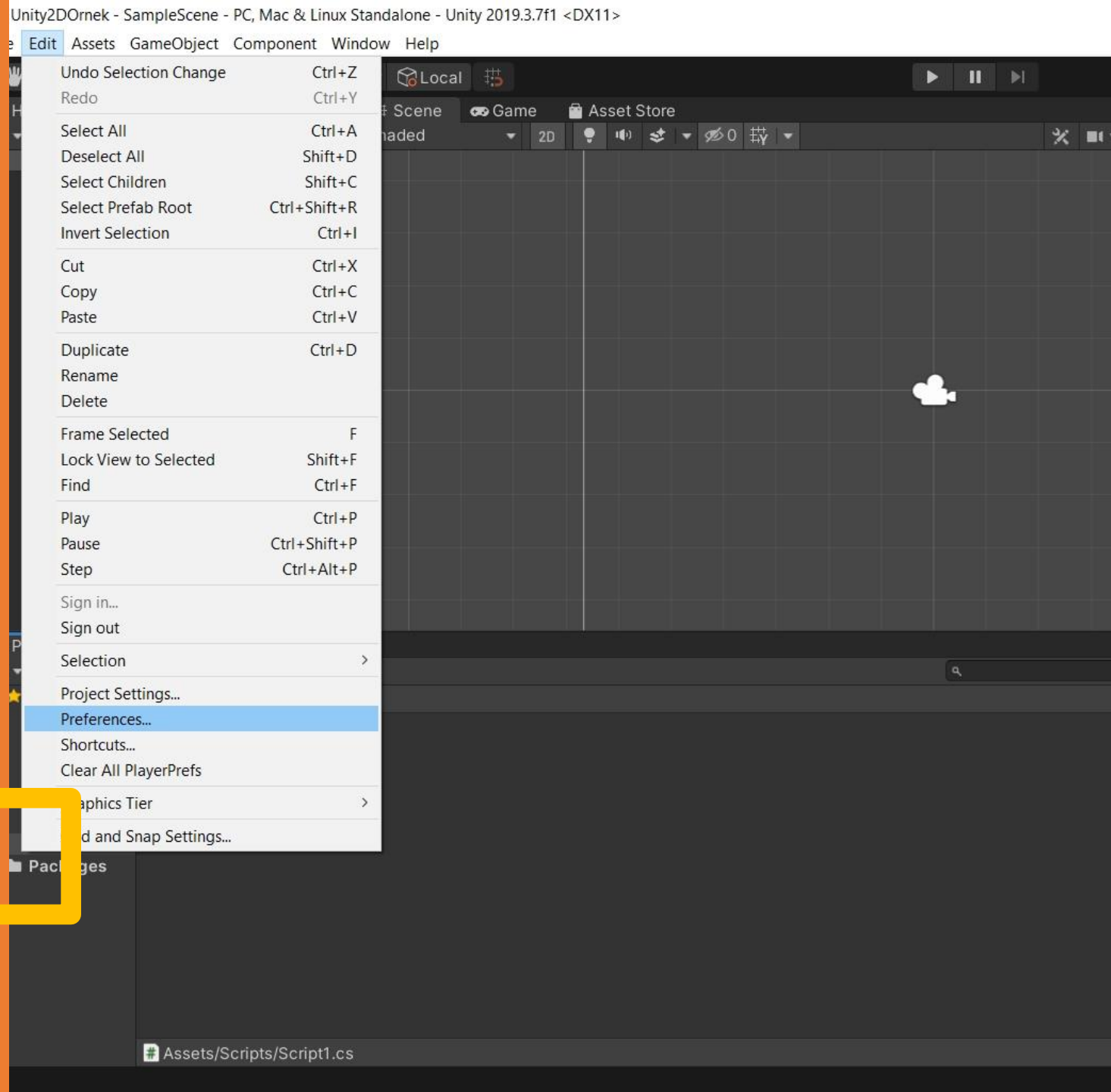
None

AssetBundle

None

Editor Seçimi

1-Preferences



Editor Seçimi

2-External Tools

Preferences

General

2D

Animation

Grid Brush

Tile Palette

Analysis

Profiler

Cache Server (global)

Colors

External Tools

GI Cache

UI Scaling

External Tools

External Script Editor

Generate all .csproj files.

Add .unityproj's to .sln

Editor Attaching

Image application

Revision Control Diff/Merge


No supported VCS diff tools were found. Please install the following tools:

- SourceGear DiffMerge
- TkDiff
- P4Merge
- Araxis Merge
- TortoiseMerge
- WinMerge
- PlasticSCM Merge
- Beyond Compare 4

Android

☒ JDK Installed with Unity (recommended)

JDK C:/Program Files/Unity/Hub/Editor/2019.3.7f1/

 JDK installed with Unity is based on OpenJDK 1.8.0_152. License information is in the install folder (AndroidPlayer)

☒ Android SDK Tools Installed with Unity (recommended)

Unity - Manual: Unity User Manu x

docs.unity3d.com/Manual/UnityManual.html

unity | DOCUMENTATION

Manual Scripting API

Search manual...

unity3d.com →

Version: 2019.3 ▾

Language : English ▾

Unity Manual

- **Unity User Manual (2019.3)**
 - Documentation versions
 - Switching between Unity versions in the documentation
 - Offline documentation
- + Packages
- + Working in Unity
- + Importing
- + Input

Unity User Manual (2019.3)

Use the Unity Editor to create 2D and 3D games, apps and experiences. Download the Editor at unity3d.com.

The Unity User Manual helps you learn how to use the Unity Editor and its associated services. You can read it from start to finish, or use it as a reference.

Unity Documentation

<https://docs.unity3d.com/Manual/index.html>

Unity - Scripting API: x +

docs.unity3d.com/ScriptReference/index.html

unity | DOCUMENTATION Manual **Scripting API** Search scripting... unity.com →

Version: 2019.3 C#

Scripting API

- + UnityEngine
- + UnityEditor
- + Unity
- + Other

Welcome to the Unity Scripting Reference!

This section of the documentation contains details of the scripting API that Unity provides. To use this information, you should be familiar with the basic theory and practice of scripting in Unity which is explained in the [Scripting section](#) of our manual.

The scripting reference is organised according to the classes available to scripts which are described along with their methods, properties and any other information relevant to their use.

The pages are extensively furnished with example code ("examples"); notwithstanding anything in the [Terms of Service](#) to the contrary, Unity grants you a non-exclusive, non-transferable, non-sublicensable, royalty-free license to access, to use, to modify, and to distribute the examples without crediting Unity.

API are grouped by namespaces they belong to, and can be selected from the sidebar to the left. For most users, the **UnityEngine** section will be the main port of call.

Scripting API

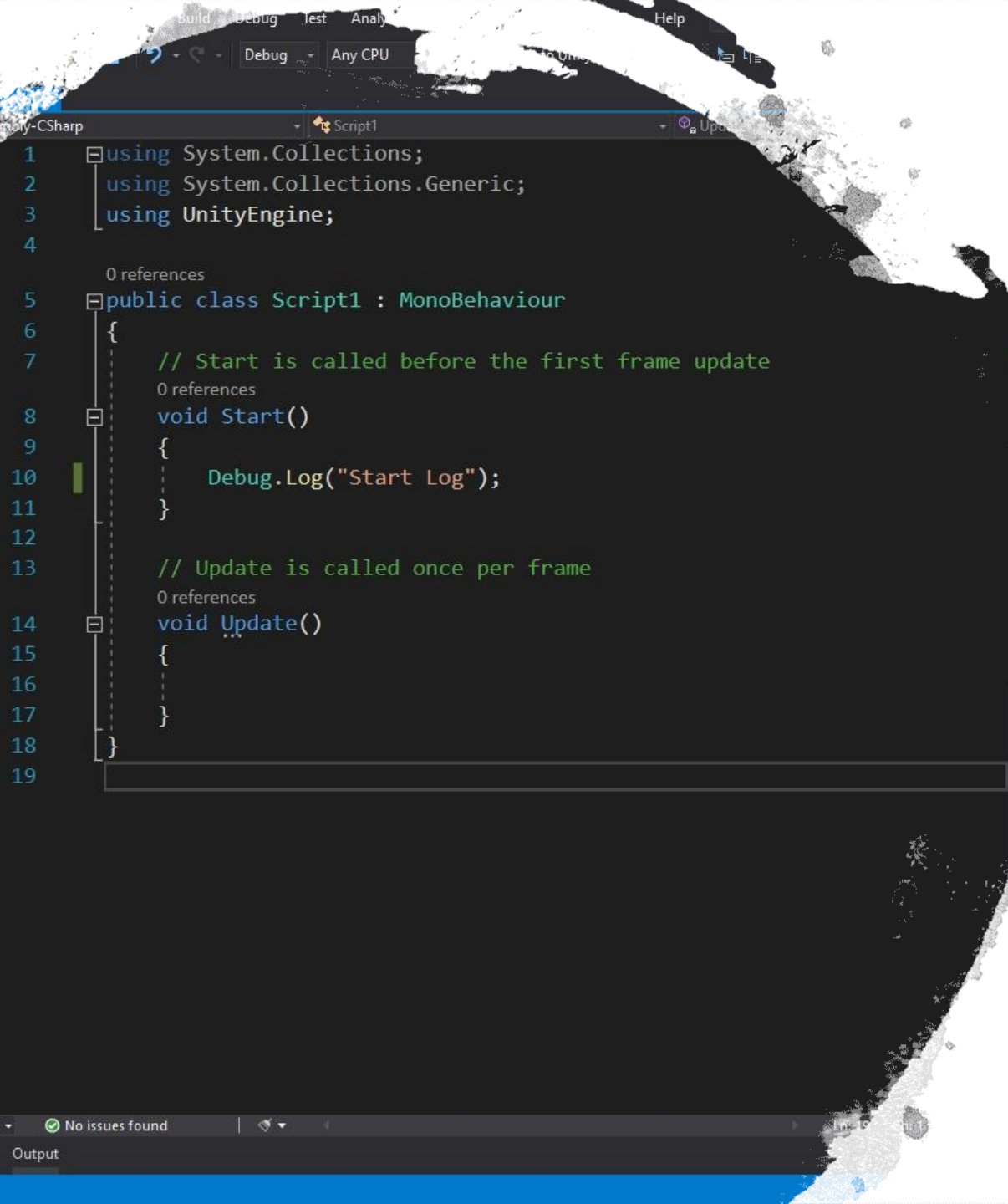
Doküman İçinde Arama Yapmak

docs.unity3d.com/ScriptReference/30_search.html?q=log

DOCUMENTATION Manual Scripting API log

Your search for "log" resulted in 232 matches:

- [Debug.Log](#)
Logs a message to the Unity Console.
- [Mathf.Log](#)
Returns the logarithm of a specified number in a specified base.
- [Logger.Log](#)
Logs message to the Unity Console using default logger.
- [Logger.filterLogType](#)
To selective enable debug log message.
- [Logger.IsLogTypeAllowed](#)
Check logging is enabled based on the LogType.
- [Logger.logEnabled](#)
To runtime toggle debug logging [ON/OFF].
- [Logger.LogError](#)
A variant of Logger.Log that logs an error message.
- [Logger.LogException](#)
A variant of Logger.Log that logs an exception message.
- [Logger.LogFormat](#)
Logs a formatted message.
- [Logger.LogWarning](#)
A variant of Logger.Log that logs an warning message.
- [LogOption](#)
Option flags for specifying special treatment of a log message.
- [LogOption.None](#)
Normal log message.



Script Çalıştırmak

1.Adım :

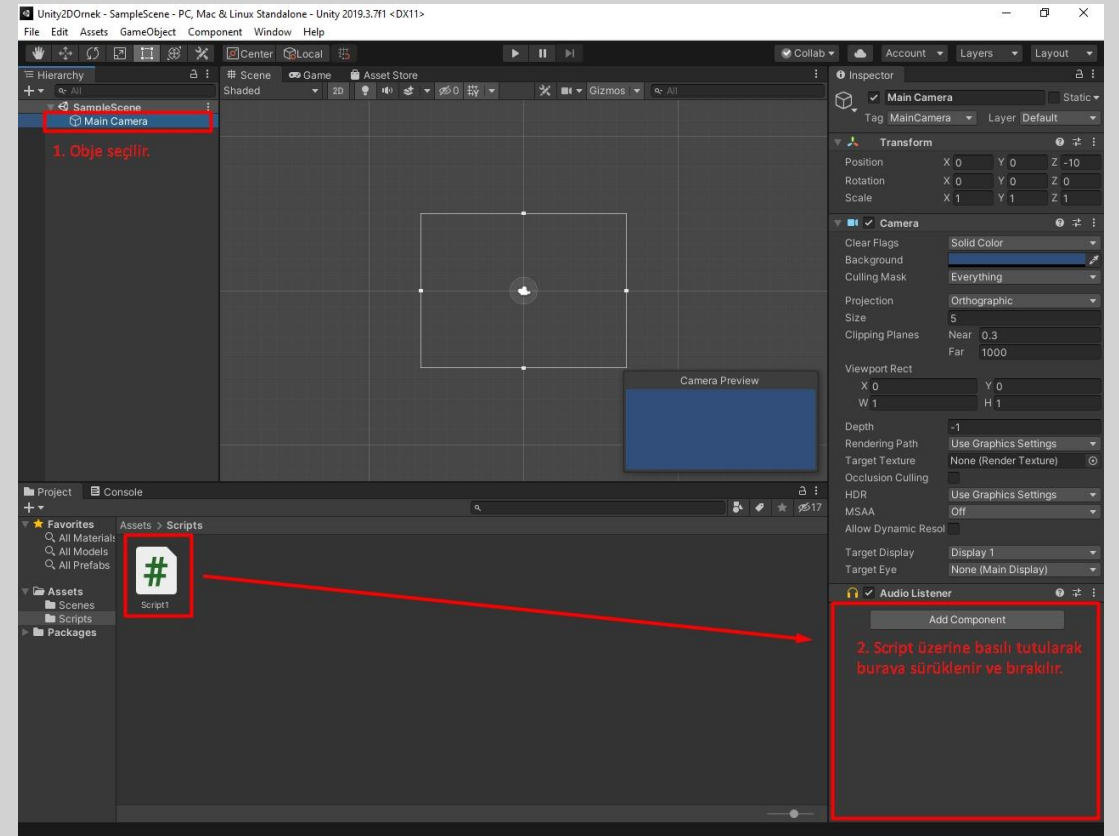
Kod düzenlenir.

Dosya kaydedilir. (Ctrl + S)

Script Çalıştırmak

2.Adım : (Yöntem-1)

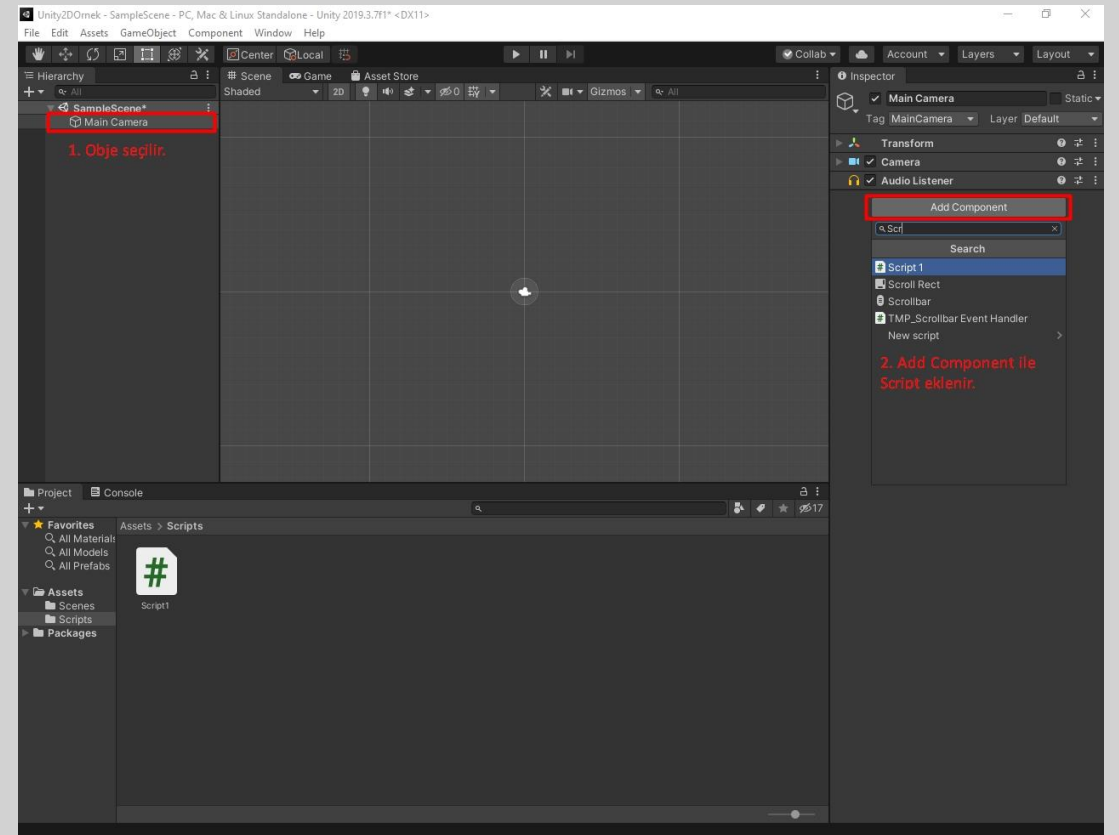
- Script çalıştırabilmek için Script'in hiyerarşideki objelere eklenmesi gerekir.
- Örneğin; Main Camera objesine Script eklemek için;
 - Main Camera seçilir.
 - Script üzerine basılı tutularak sürüklenir ve Add Component kısmına bırakılır.
 - Böylece Script objeye eklenmiş olur.

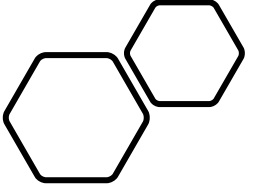


Script Çalıştırmak

2.Adım : (Yöntem-2)

- Script çalıştırabilmek için Script'in hiyerarşideki objelere eklenmesi gerekir.
- Örneğin; Main Camera objesine Script eklemek için;
 - Main Camera seçilir.
 - Add Component'e tıklanarak Script seçilir.
 - Böylece Script objeye eklenmiş olur.

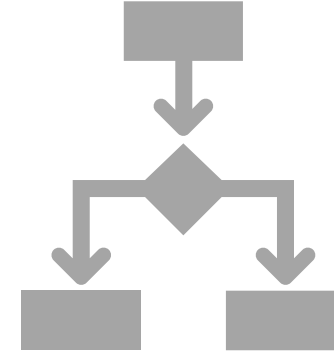




Script Örnek-1 SORU:



Unity Console kısmına «Proje çalıştı»
Log'unu veren Script'i oluşturunuz.



Bir objenin Start metodunda bu
Script'in çalışmasını sağlayınız.



Script Örnek-1

CEVAP:

```
void Start()  
{  
    Debug.Log("Proje çalıştı.");  
}
```


Binary Sayı Sistemi

Transistor ve Kondansatör



Açık (1) – Kapalı (0)



Binary Sayı Sistemi

1'ler ve 0'lar

$$15_{10} \rightarrow 1111_2$$

$$532_{10} \rightarrow 1000010100_2$$

1000010111

$2^0 \rightarrow 1 \times 1$	= 1
$2^1 \rightarrow 1 \times 2$	= 2
$2^2 \rightarrow 1 \times 4$	= 4
$2^3 \rightarrow 0 \times 8$	= 0
$2^4 \rightarrow 1 \times 16$	= 16
$2^5 \rightarrow 0 \times 32$	= 0
$2^6 \rightarrow 0 \times 64$	= 0
$2^7 \rightarrow 0 \times 128$	= 0
$2^8 \rightarrow 0 \times 256$	= 0
$2^9 \rightarrow 1 \times 512$	= 512

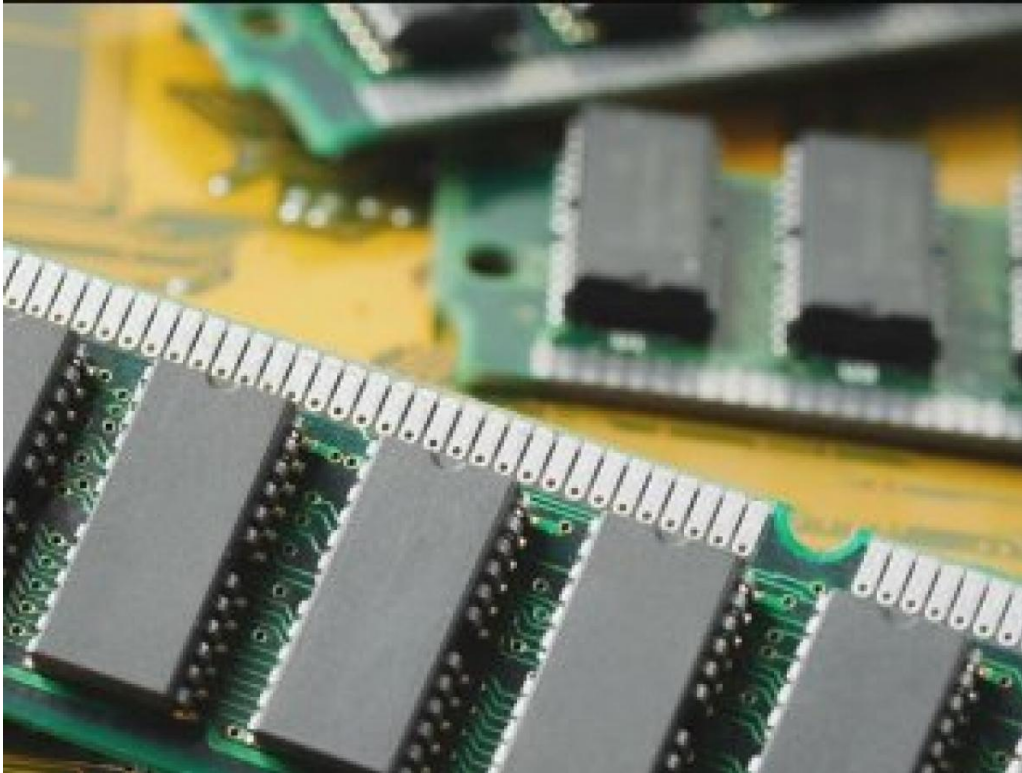
$$\begin{array}{r} + \\ \hline 535 \end{array}$$

Birimler Arası Karşılaştırma

Kapasite	Değer
1 bit	0 veya 1
1 Byte	8 Bit
1 Kilobyte (KB)	1024 B
1 Megabyte (MB)	1024 KB
1 Gigabyte (GB)	1024 MB
1 Terabyte (TB)	1024 GB
1 Petabyte (PB)	1024 TB
1 Exabyte (EB)	1024 PB
1 Zettabyte (ZB)	1024 EB
1 Yottabyte (YB)	1024 ZB

Bellekler

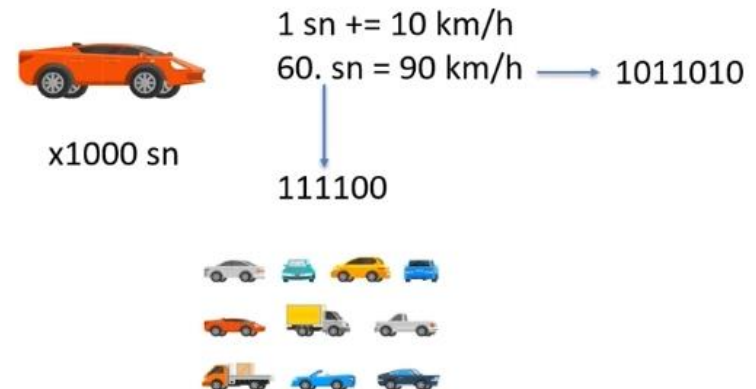
Bilgisayar Dünyası = Binary



Oyunda Kaynak Kullanımı

- Oyunumuzun kaynak kullanımını azaltmak adına ihtiyacımız olmayan verilerin bellek alanına kaydedilmesini engellemeliyiz.

Araba yarışı oyunu



Binary Örnek-1

SORU:

Sayıları 10'luk sayı sistemine çeviriniz.
(Binary - Decimal)

- 11101101
- 1100
- 10
- 1110010
- 100001

Binary Örnek-1

CEVAP:

- $(11101101)_2 \rightarrow 128 + 64 + 32 + 8 + 4 + 1 = (237)_{10}$
- $(1100)_2 \rightarrow 8 + 4 = (12)_{10}$
- $(10)_2 \rightarrow (2)_{10}$
- $(1110010)_2 \rightarrow 64 + 32 + 16 + 2 = (114)_{10}$
- $(100001)_2 \rightarrow 32 + 1 = (33)_{10}$

Binary Örnek-2

SORU:

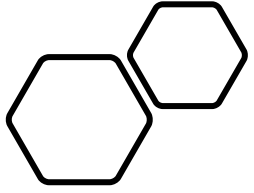
- Sayıları 2'lik sayı sistemine çeviriniz.
(Decimal - Binary)

- 173
- 100
- 966
- 27
- 1

Binary Örnek-2

CEVAP:

- $(173)_{10} \rightarrow (10101101)_2$
- $(100)_{10} \rightarrow (1100100)_2$
- $(966)_{10} \rightarrow (1111000110)$
- $(27)_{10} \rightarrow (11011)_2$
- $(1)_{10} \rightarrow (1)_2$



Class (Sınıf) ve Objeler (Nesneler)



OYUN OBJELERİ
(GAME OBJECTS)



OBJECT = NEW CLASS()

Objeler

- State (durum) = Karakteristik özellikler
 - Örneğin;
 - Maksimum hız
 - Renk
 - Konum
- Behaviour (davranış)
 - Örneğin;
 - Arabayı kullanmak
 - Sahneden arabanın yok olması
- Identity (kimlik)



State



- State (durum)
 - Maksimum hız
 - Renk
 - Haritaki konum

} Field (alan)
- Veri türleri
 - Oyuncu puanı = 14;
 - State = Oyuncu puanı
 - Field → int oyuncuPuanı = 14;
 - Field = Değişken
- Field
 - Dışarıdan erişim verilmemelidir.
 - Erişim için Property kullanılır.

Behaviour



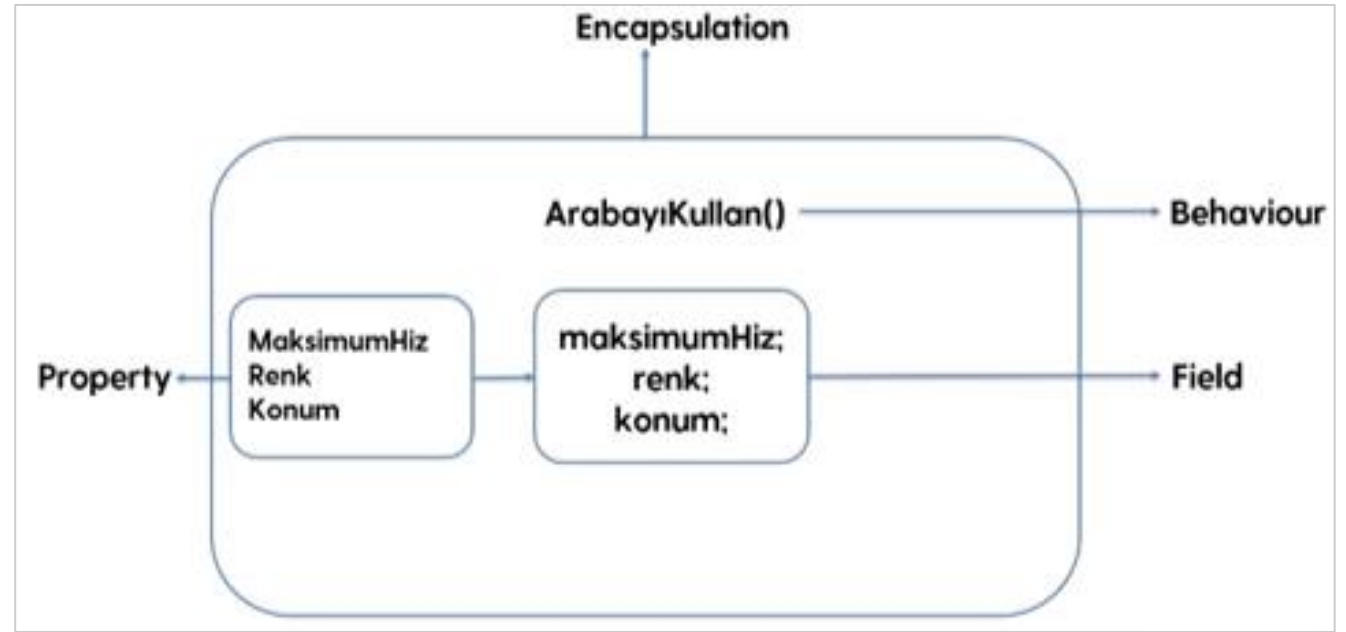
- Behaviour (Davranış)
 - Örneğin;
 - Arabayı sürmek
 - Arabanın sahneden yok olması (destroy)
- Erişim metotlarla (fonksiyonlarla) yapılır.

Identity

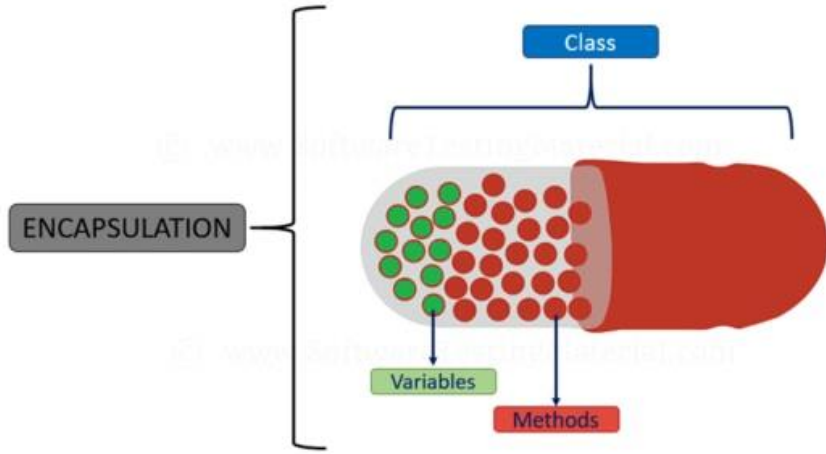
- Identity (kimlik)
 - Nesneye bir isim vermemizdir.
 - Nesnenin bellekteki adresini belirler.



Encapsulation (Kapsülleme)



- Kapsülleme nesne yönelimli programlamada herhangi bir nesnenin metotlarını, verilerini ve değişkenlerini diğer nesnelerden saklayarak ve bunlara erişimini sınırlandırarak yanlış kullanımlardan koruyan bir konsepttir.



Kapsülleme Metafor

Public : Her yerden erişilebilir. (Kalıtım ile aktarılır.)

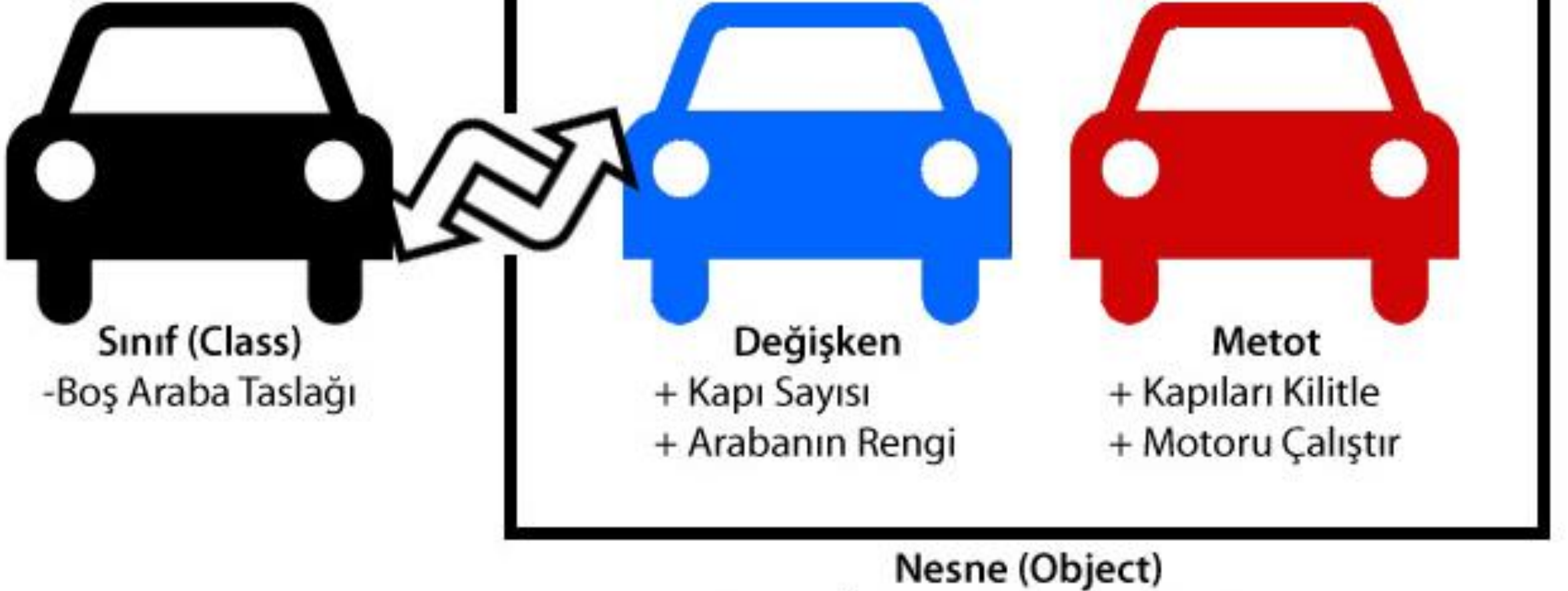
Private : Sadece tanımlandığı sınıf içerisinde erişilebilir. (Kalıtım ile aktarılmaz.)

Protected : Tanımlandığı sınıfta ve o sınıfı miras (kalıtım) alan sınıflardan erişilebilir. (Kalıtım ile aktarılır.)

Internal : Sadece bulunduğu projede erişilebilir. (Kalıtım ile aktarılır.)

Class (Sınıf) ve Object (Nesne)

- Sınıf (class), nesnenin özelliklerini ve davranışlarını tanımlar. Nesne (Object) ise sınıfın aktif bir örneğidir.
- Bir sınıf kod yazma aşamasında tanımlanır ve her zaman var olur. Ancak nesne çalışma zamanında oluşturur ve uygulama çalıştığı sürece kullanılabilir.



DOG

Breed
Size
Age
Color

Eat()
Sleep()
Sit()
Run()

Breed = Neapolitan Mastiff
Size = Large
Age = 5 years
Color = Black



Breed = Maltese
Size = Small
Age = 2 years
Color = White



Breed = Chow Chow
Size = Midium
Age = 3 years
Color = Brown



Car class



Car
Objects



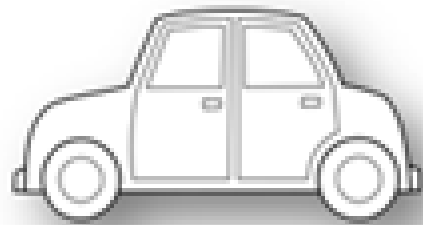
Green
Ford
Mustang
Gasoline



Red
Toyota
Prius
Electricity

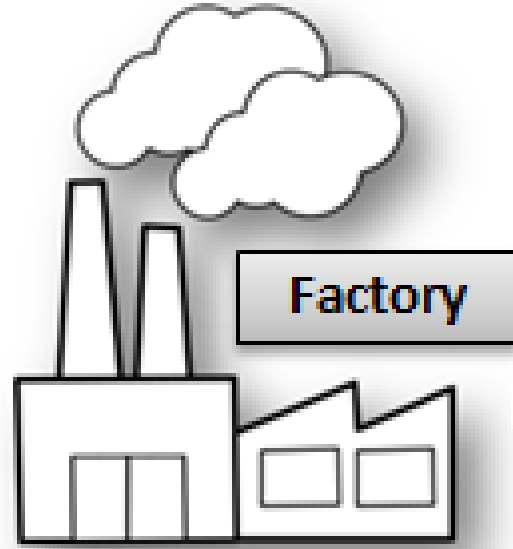


Blue
Volkswagon
Golf
Deisel



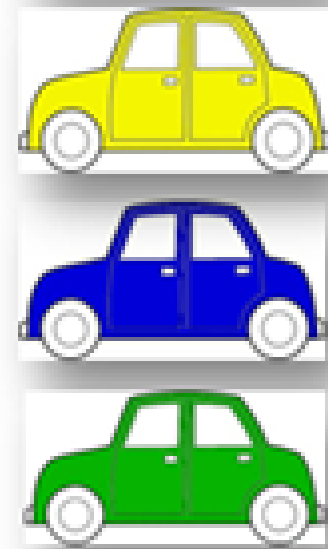
**Class
(pattern)**

*Pattern of car
of same type*



Constructor

*Sequence of actions
required so that factory
constructs a car object*



Objects

Car
*Can create many
objects from a
class*

Metotlar

Metot, program içerisinde belirli görevleri yerine getirmek için özelleştirilmiş kod bloklarına verilen isimdir.

Erişim Belirleyici | Dönüş Tipi | Metot Adı (Parametre Listesi)

```
{  
    //Metot Gövdesi  
}
```

Metot Kullanmanın Avantajları



Metotlar aynı kodların sürekli tekrarlanmasını önler.



Programların okunabilirliğini ve anlaşılabilirliğini artırır.



Aynı kod bloklarının farklı farklı yerlerde yeniden kullanılmasına olanak sağlar.

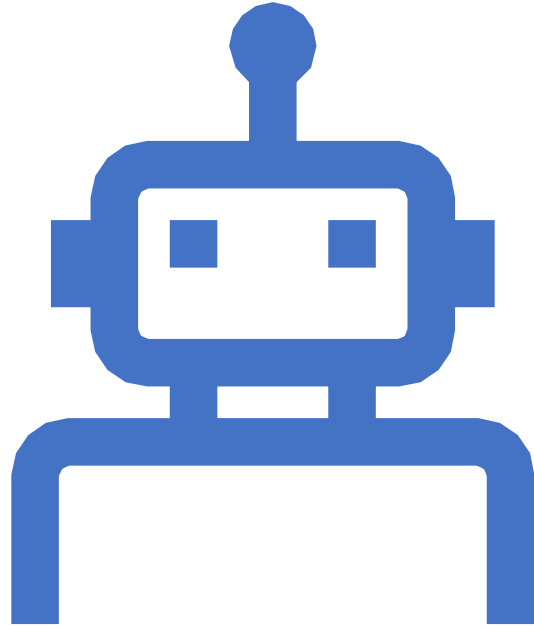


Metotlar nesne tabanlı programlamada önemli bir yere sahiptir.

Metot Tanımlarken Dikkat Edilecek Hususlar

- Metotlar her zaman bir sınıfın içerisinde tanımlanırlar.
- Metotların erişim belirleyicisinin yazılması zorunlu değildir. Yazılmadığı takdirde private olarak kabul edilir.
- Her metodun geri dönüş tipi mutlaka belirtilmelidir. Geriye herhangi bir değer döndürülüyorsa dönüş tipi void olarak belirtilir. Geriye döndürülecek değer return anahtar kelimesiyle bildirilir.
- Metotlar parametrelili veya parametresiz olarak kullanılabilirler. Eğer metot birden fazla parametre alıyorsa her bir parametre arasına virgül konularak birbirinden ayrılır.





Constructor (Yapılandırıcılar)

Yapılandırıcıların (constructor) görevi oluşturulan nesneyi ilk kullanıma hazırlamasıdır. C# da tüm sınıflar (class) tanımlansın ya da tanımlanmasın değer tiplerine sıfır, referans tiplerine "null" değerini atayan varsayılan bir yapılandırıcı vardır. Yapılandırıcısı tanımlandıktan sonra varsayılan yapılandırıcı bir daha kullanılmaz.

Yapılandırıcıların bilinen temel özellikleri:

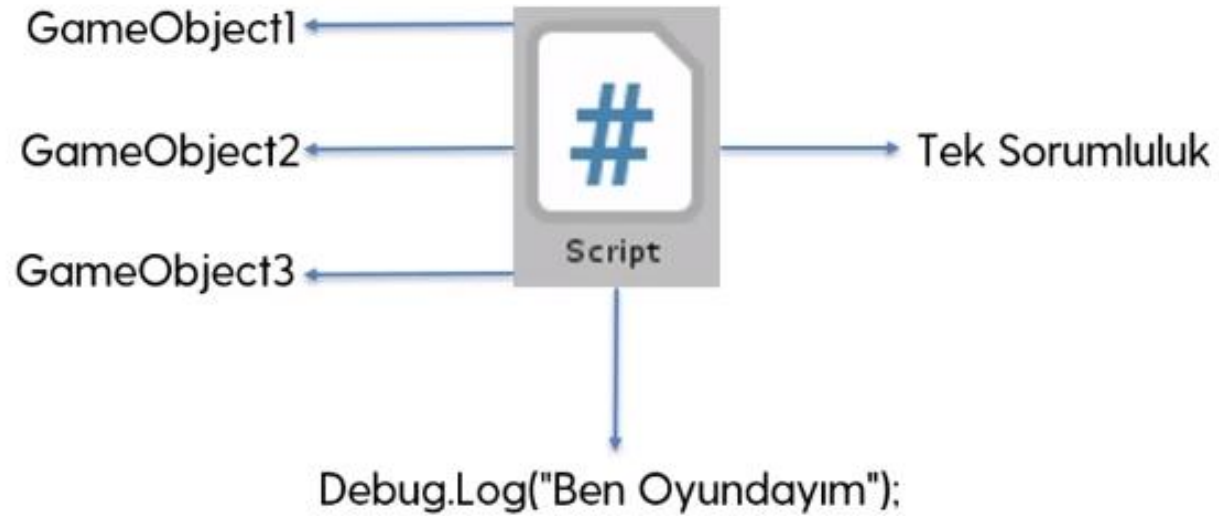
- Kendi sınıfı ile aynı isme sahip olması,
- Genellikle açık bir dönüş tipi olmaması,
- Başka sınıflar tarafından kullanılabilmesi için erişimin public olmasıdır.

Oyun Objeleri ve Tasarım



- Unity Bileşenleri (Component)
- Game Object → Sahneye sürüklenip bırakılan objelerdir.
- Sprite → 2D Grafik nesneleridir. Yani grafiksel varlıklardır.
- Sprite Sheets (Tümleşik Resim) → Resimlerin birbirinin peşi sıra dizilmeleri ve resimde yer alan nesnenin, her karede farklı pozisyonunun kullanılması ile oluşturulmuş dizi serisidir.
- Transform Component (Position, Rotation, and Scale) → Objenin sahnedeki konumu
- Sprite Renderer → Objenin sahnede bir görsel olarak gösterilebilmesi yani render edilebilmesi için gereklidir.
- Script → Transform Component

- Single Responsibility Principle (SPR)



Script

Hand icon

Undo Inspector Ctrl+Z

Redo Ctrl+Y

Select All Ctrl+A

Deselect All Shift+D

Select Children Shift+C

Select Prefab Root Ctrl+Shift+R

Invert Selection Ctrl+I

Cut Ctrl+X

Copy Ctrl+C

Paste Ctrl+V

Duplicate Ctrl+D

Rename

Delete

Frame Selected F

Lock View to Selected Shift+F

Find Ctrl+F

Play Ctrl+P

Pause Ctrl+Shift+P

Step Ctrl+Alt+P

Sign in...

Sign out

Selection >

Project Settings...

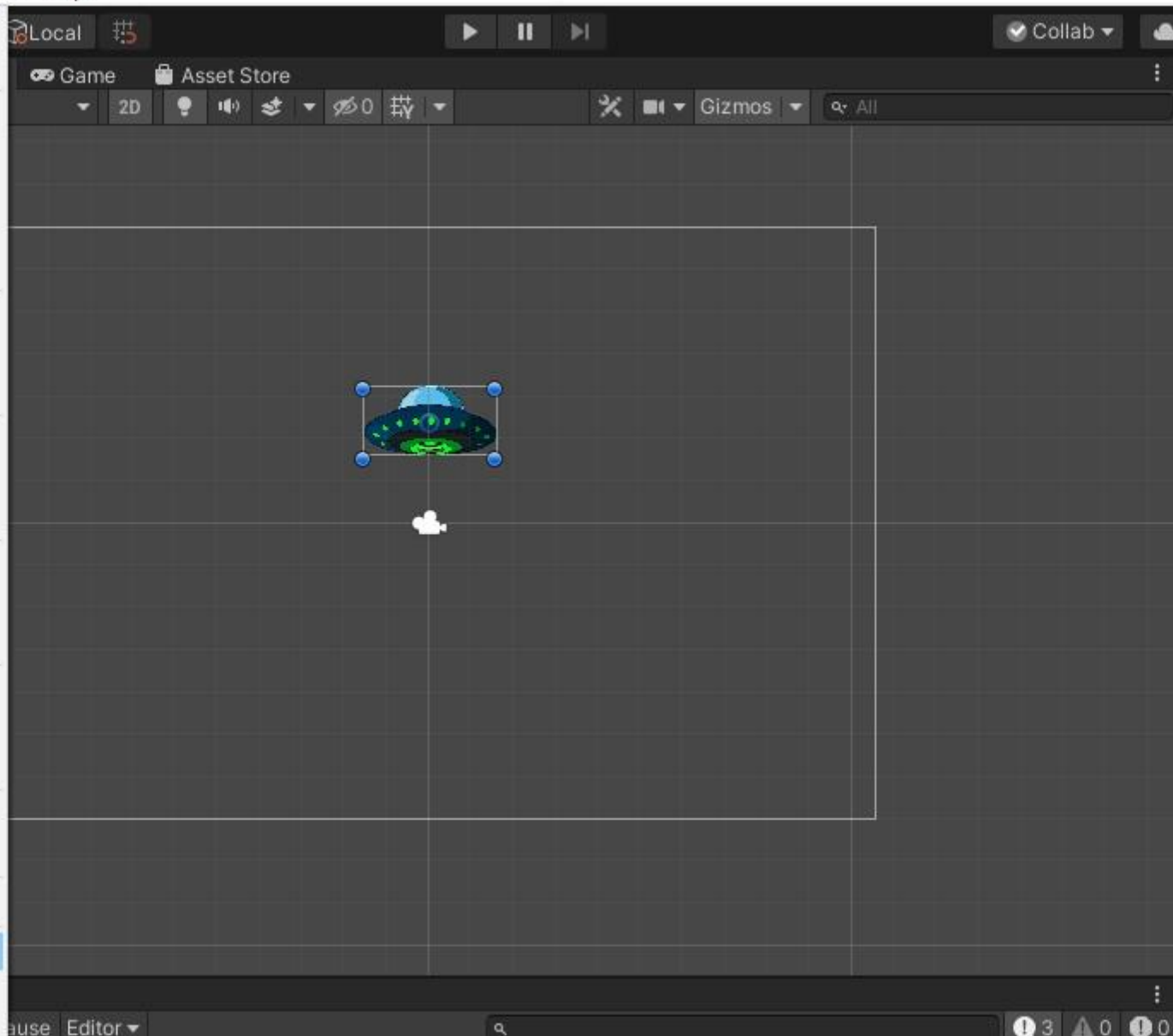
Preferences...

Shortcuts...

Clear All PlayerPrefs

Graphics Tier >

Grid and Snap Settings



Inspector

uzaygemisi2 Static

Tag Untagged Layer Default

Transform

Position X 0.06 Y 2.42 Z 0

Rotation X 0 Y 0 Z 0

Scale X 0.3267 Y 0.3267 Z 0.3267

Sprite Renderer

Sprite uzaygemisi2

Color

Flip X Y

Draw Mode Simple

Mask Interaction None

Sprite Sort Point Center

Material Sprites-Default

Additional Settings

Sorting Layer Default

Order in Layer 0

Rigidbody 2D

Body Type Dynamic

Material None (Physics Mate)

Simulated

Use Auto Mass

Mass 1

Linear Drag 0


Angular Drag 0.05

Gravity Scale 1

Collision Detection Discrete

Sleeping Mode Start Awake

Interpolate None

Graphics	Gravity	X 0	Y 0
Input Manager	Default Material	None (Physics Material 2D) 	
Physics	Velocity Iterations	8	
Physics 2D	Position Iterations	3	
Player	Velocity Threshold	1	
Preset Manager	Max Linear Correction	0.2	
Quality	Max Angular Correction	8	
Script Execution Order	Max Translation Speed	100	
Tags and Layers	Max Rotation Speed	360	
TextMesh Pro	Baumgarte Scale	0.2	
Time	Baumgarte Time Of Impact Scale	0.75	
VFX	Time To Sleep	0.5	
XR Plugin Management	Linear Sleep Tolerance	0.01	
	Angular Sleep Tolerance	2	
	Default Contact Offset	0.01	

