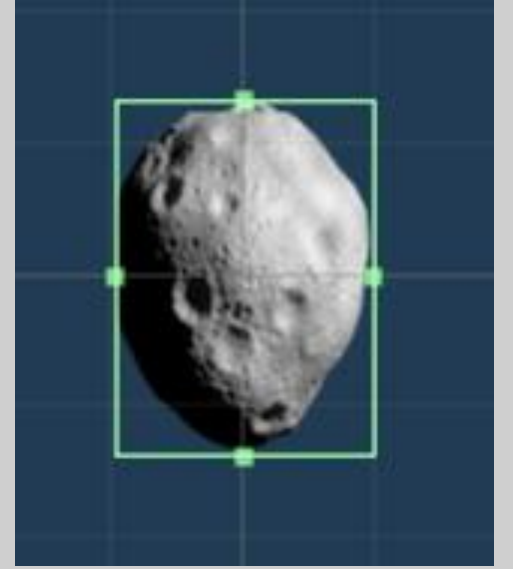


Unity

Öğr. Gör. Gözde Mihran ALTINSOY

Objenin Oyun Ekranının Dışına
Çıkmasını Engellemek

Proje alıřtırıldığında Mouse imlecinin bulunduėu konum Collider2D component'inin en ve boy deėerlerinin tam ortasına gelecek řekilde konumlanmaktadır.
Bu yzden bu koordinat deėerlerinin yarısına ihtiyaımız vardır.



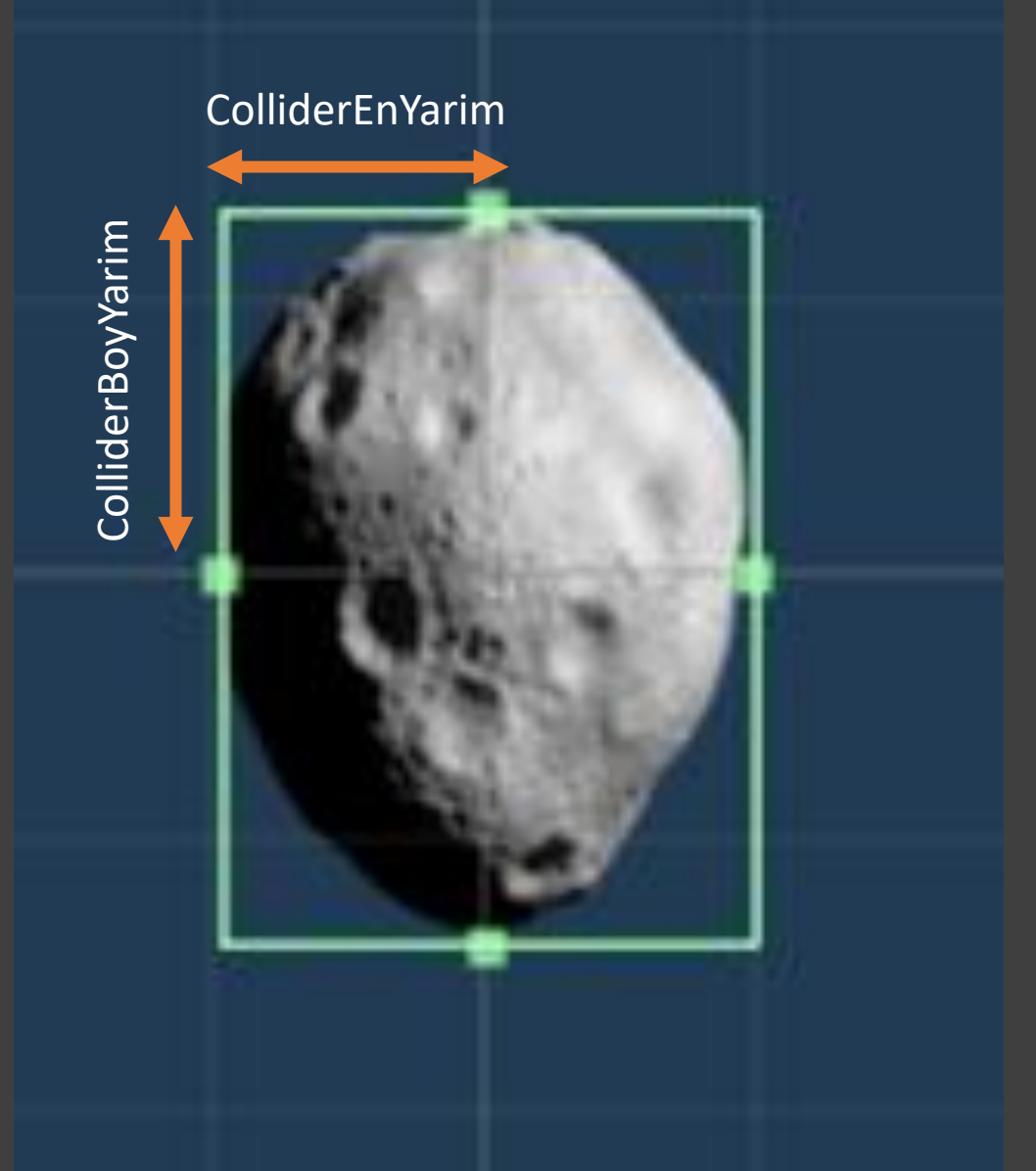


BoxCollider2D koordinat değerlerinin yarısını alalım

```
public class HareketKontrol : MonoBehaviour
{
    float colliderBoyYarim;
    float colliderEnYarim;
    private void Start()
    {
        //Collider component'inin referansını alıyoruz
        BoxCollider2D collider = GetComponent<BoxCollider2D>();
        //Yükseklik y ekseninde olduğu için;
        colliderBoyYarim = collider.size.y / 2;
        colliderEnYarim = collider.size.x / 2;
    }
}
```

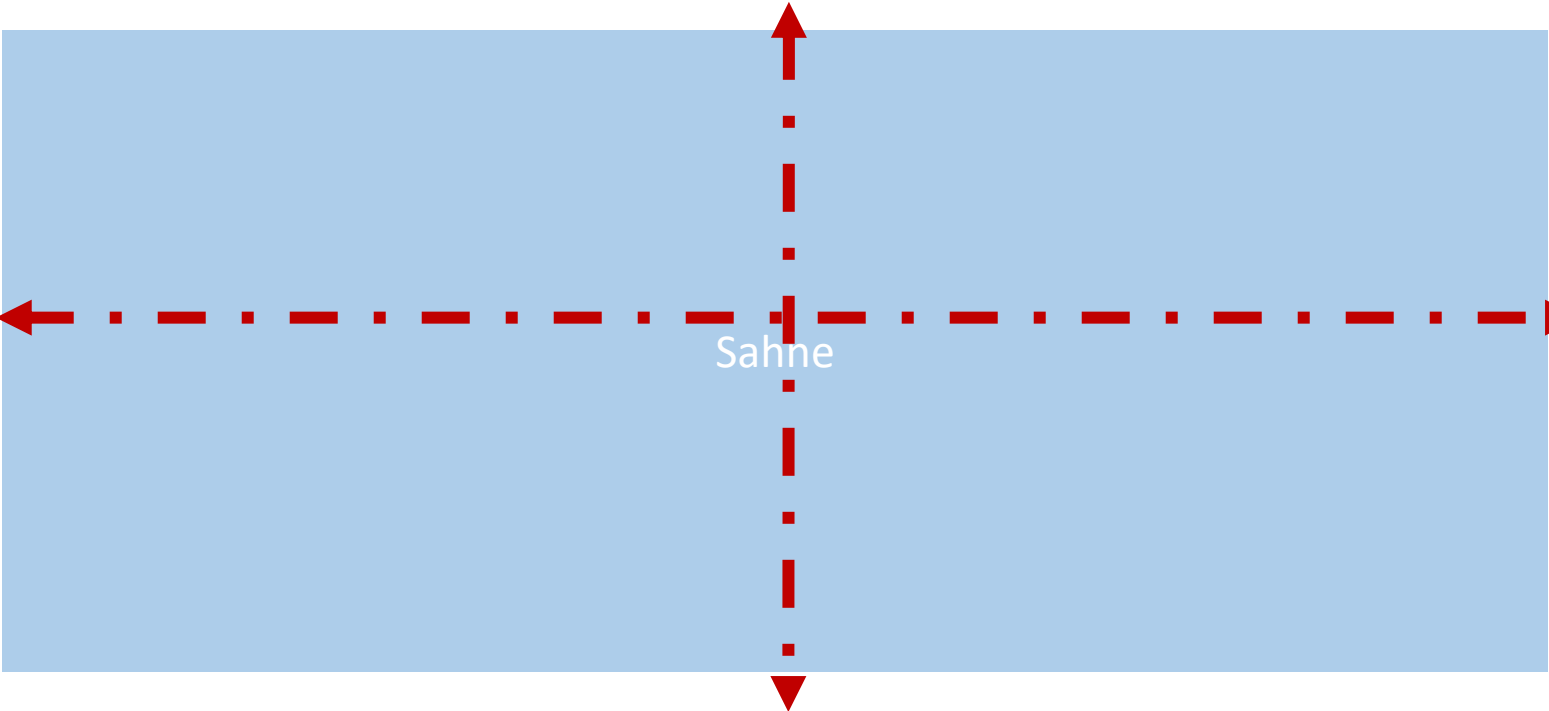
Objenin Oyun Ekranının Dışına Çıkmasını Engellemek

- Kontrolcünün ekranın dışına çıkması durumunda objenin sahnenin dışına çıkmasını engelleyip, ekranda kalmasını sağlayalım.
- Bunun için EkrandaKal fonksiyonu ile kursörün konumunu kontrol edelim.
- Eğer sahnenin dışına çıkarsa yeni konumunun sahnenin sınırına gelmesini sağlayalım.



Vector3(x,y,z)

Ust = sagUstKose.y

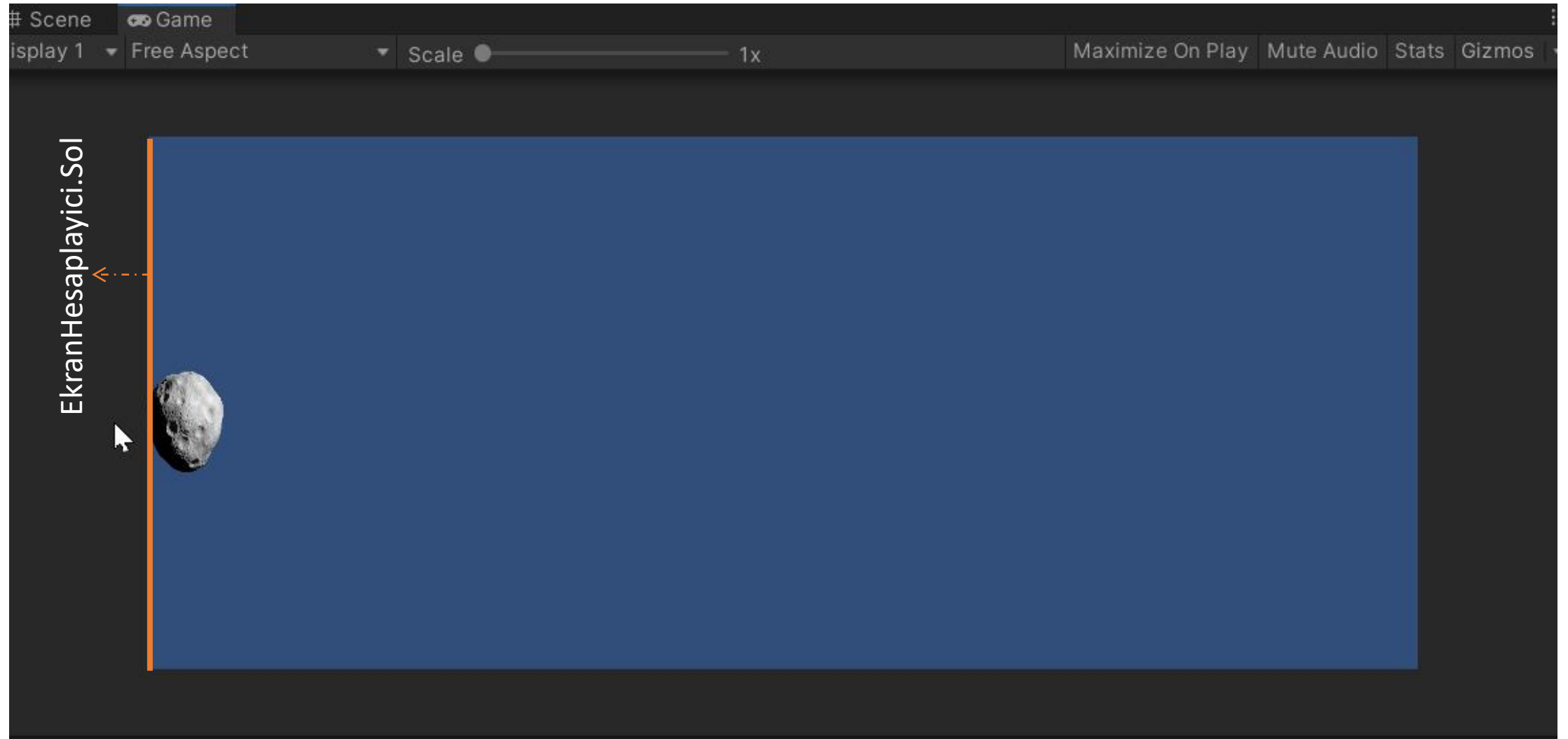


Sol = solAltKose.x

Sahne

Sag = sagUstKose.x

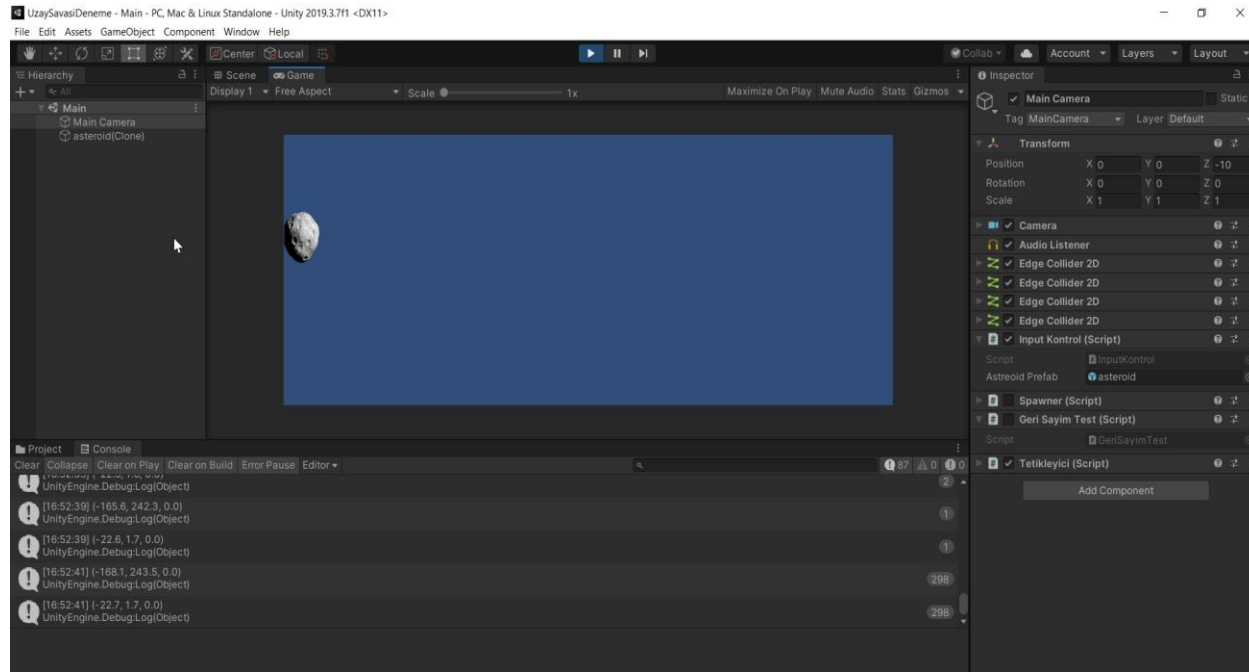
Alt = solAltKose.y

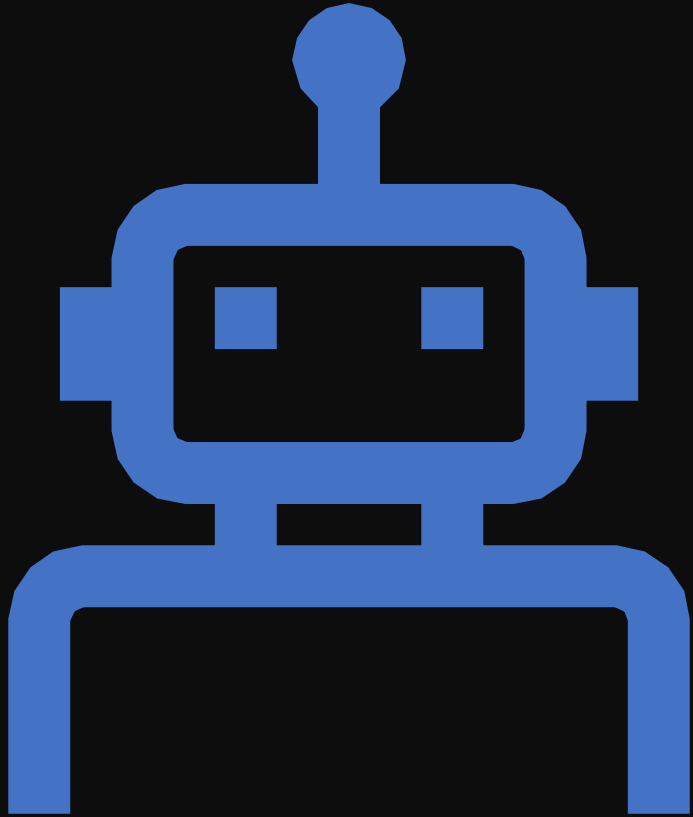


HareketKontrol.cs

```
void EkrandaKal()
{
    Vector3 position = transform.position;
    //Debug.Log(position.x + " " + colliderEnYarim + " " +
    EkranHesaplayici.Sol + (colliderEnYarim + EkranHesaplayici.Sol));
    if (position.x - colliderEnYarim < EkranHesaplayici.Sol)
        position.x = EkranHesaplayici.Sol + colliderEnYarim;
    else if (position.x + colliderEnYarim >
    EkranHesaplayici.Sag)
        position.x = EkranHesaplayici.Sag - colliderEnYarim;
    if (position.y + colliderBoyYarim > EkranHesaplayici.Ust)
        position.y = EkranHesaplayici.Ust - colliderBoyYarim;
    else if (position.y - colliderBoyYarim <
    EkranHesaplayici.Alt)
        position.y = EkranHesaplayici.Alt + colliderBoyYarim;
    transform.position = position;
}
```

```
void Update()
{
    Debug.Log(Input.mousePosition);
    //Astreoid mouse imlecini takip edecek
    Vector3 position = Input.mousePosition;
    position.z = -Camera.main.transform.position.z;
    position = Camera.main.ScreenToWorldPoint(position);
    transform.position = position;
    Debug.Log(position);
    EkrandaKal();
}
```



Oyuncu ile Klavye Mouse Etkileşimi (INPUT)

- HumanTrait
- ImageConversion
- Input**
- IntegratedSubsystem
- IntegratedSubsystemDescriptor
- Joint
- Joint2D
- JointAngleLimits2D
- JointDrive
- JointLimits
- JointMotor
- JointMotor2D
- JointSpring
- JointSuspension2D
- JointTranslationLimits2D
- JsonUtility
- Keyframe
- LayerMask
- LazyLoadReference<T0>
- LensFlare
- Light
- LightBakingOutput
- LightmapData

Static Methods

GetAccelerationEvent	Returns specific acceleration measurement which occurred during last frame. (Does not allocate temporary variables).
GetAxis	Returns the value of the virtual axis identified by axisName.
GetAxisRaw	Returns the value of the virtual axis identified by axisName with no smoothing filtering applied.
GetButton	Returns true while the virtual button identified by buttonName is held down.
GetButtonDown	Returns true during the frame the user pressed down the virtual button identified by buttonName.
GetButtonUp	Returns true the first frame the user releases the virtual button identified by buttonName.
GetJoystickNames	Returns an array of strings describing the connected joysticks.
GetKey	Returns true while the user holds down the key identified by name.
GetKeyDown	Returns true during the frame the user starts pressing down the key identified by name.
GetKeyUp	Returns true during the frame the user releases the key identified by name.
GetMouseButton	Returns whether the given mouse button is held down.
GetMouseButtonDown	Returns true during the frame the user pressed the given mouse button.
GetMouseButtonUp	Returns true during the frame the user releases the given mouse button.
GetTouch	Call Input.GetTouch to obtain a Touch struct.
IsJoystickPreconfigured	Determine whether a particular joystick model has been preconfigured by Unity. (Linux-only).
ResetInputAxes	Resets all input. After ResetInputAxes all axes return to 0 and all buttons return to 0 for one frame.



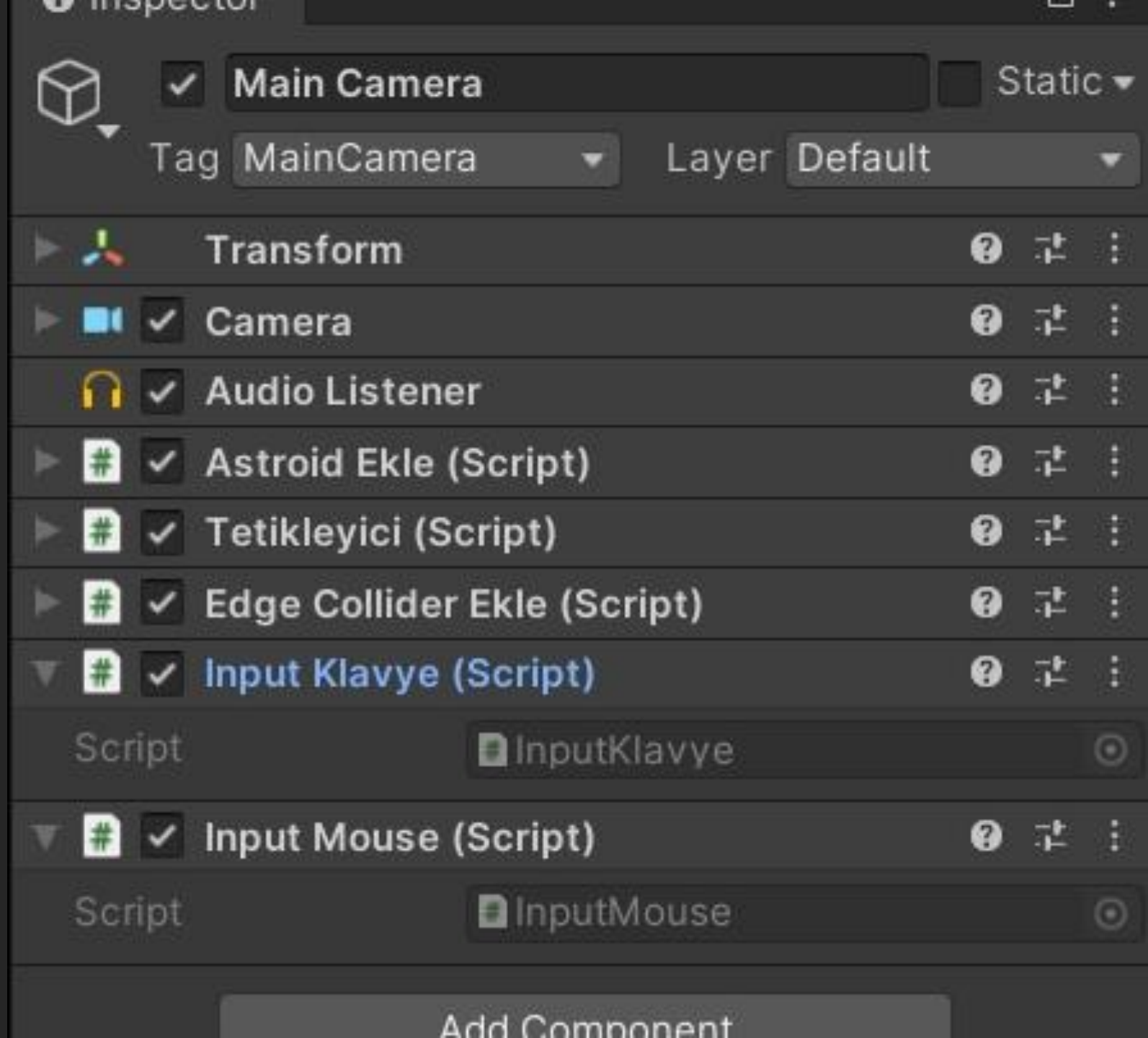
Input GetMouseButton

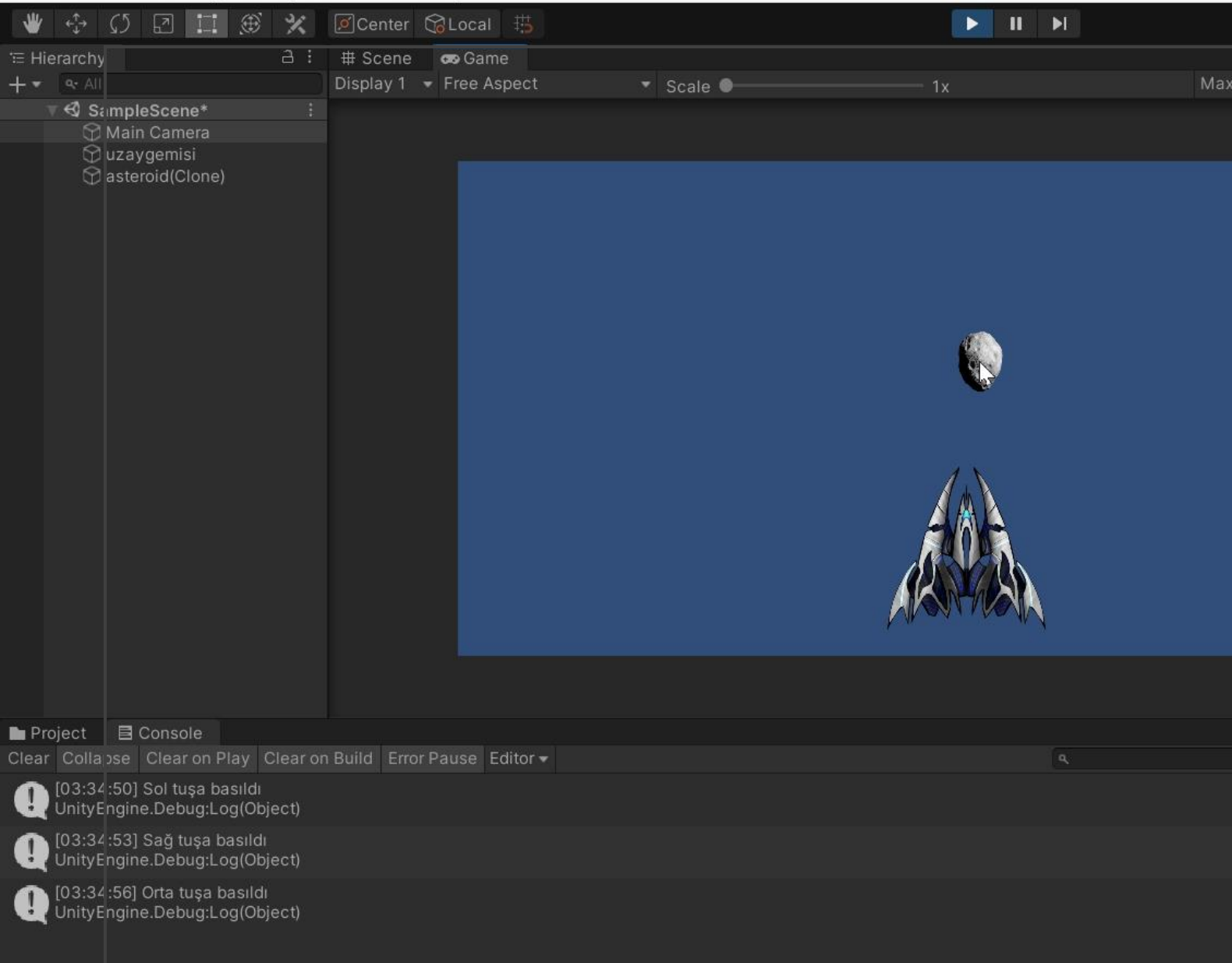
- Farenin bir tuşuna basılı tutuluyorsa true değerini döndürür.
- Input.GetMouseButtonDown fonksiyonu ilgili tuşa basıldığında gerçekleşir.
- Input.GetMouseButtonUp ilgili tuş bırakıldığında gerçekleşir.
- Mouse sol tuşuna basıldığında 0 değerini döndürür.
- Mouse sağ tuşuna basıldığında 1 değerini döndürür.
- Mouse orta tuşuna basıldığında 2 değerini döndürür.
- Kullanımı:
 - Input.GetMouseButton(mouseTuşu : int)

InputMouse.cs

```
void Update()
{
    if (Input.GetMouseButtonDown(0))
        Debug.Log("Sol tuşa basıldı");
    else if (Input.GetMouseButtonDown(1))
        Debug.Log("Sağ tuşa basıldı");
    else if (Input.GetMouseButtonDown(2))
        Debug.Log("Orta tuşa basıldı");
}
```

InputMouse
Component
Main
Camera'ya
eklenir.





Farenin Sol, Sağ
ve Orta tuşlarına
sırasıyla
basıldığında
resimdeki Log'lar
oluşur.

Input anyKey ve Input anyKeyDown

anyKey

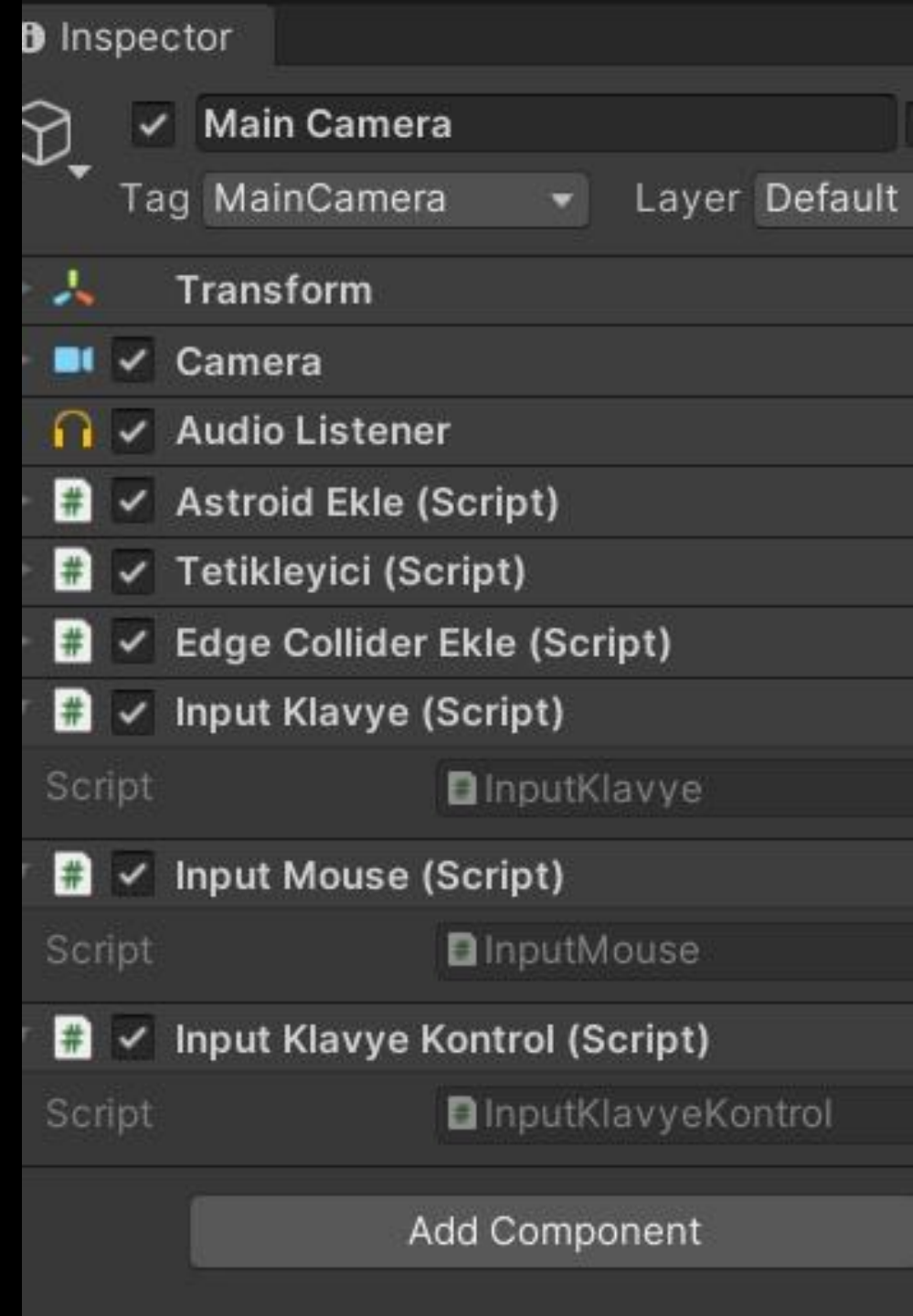
- Herhangi bir mouse ya da klavye tuşuna basılıp basılmadığını döndürür.
- Tuşa mevcut frame'de basılması gerekli değildir.
- Daha önceki bir kareden basılı tutulan bir tuş true değer döndürür.
- Kullanımı:
 - Input.anyKey : boolean:

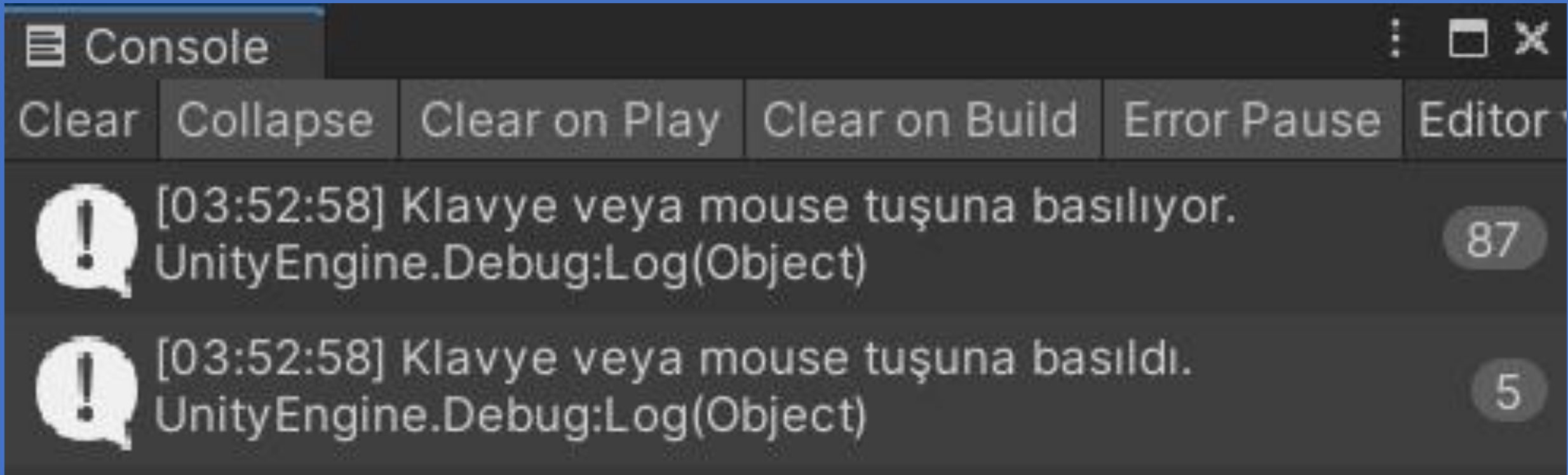
anyKeyDown

- Mevcut karede (frame) herhangi bir mouse ya da klavye tuşuna basılıp basılmadığını döndürür.
- Tuşa mevcut frame'de basılması gerekir.
- Daha önceki bir kareden basılı tutulan bir tuş true değer döndürmez.
- Kullanımı:
 - Input.anyKeyDown : boolean

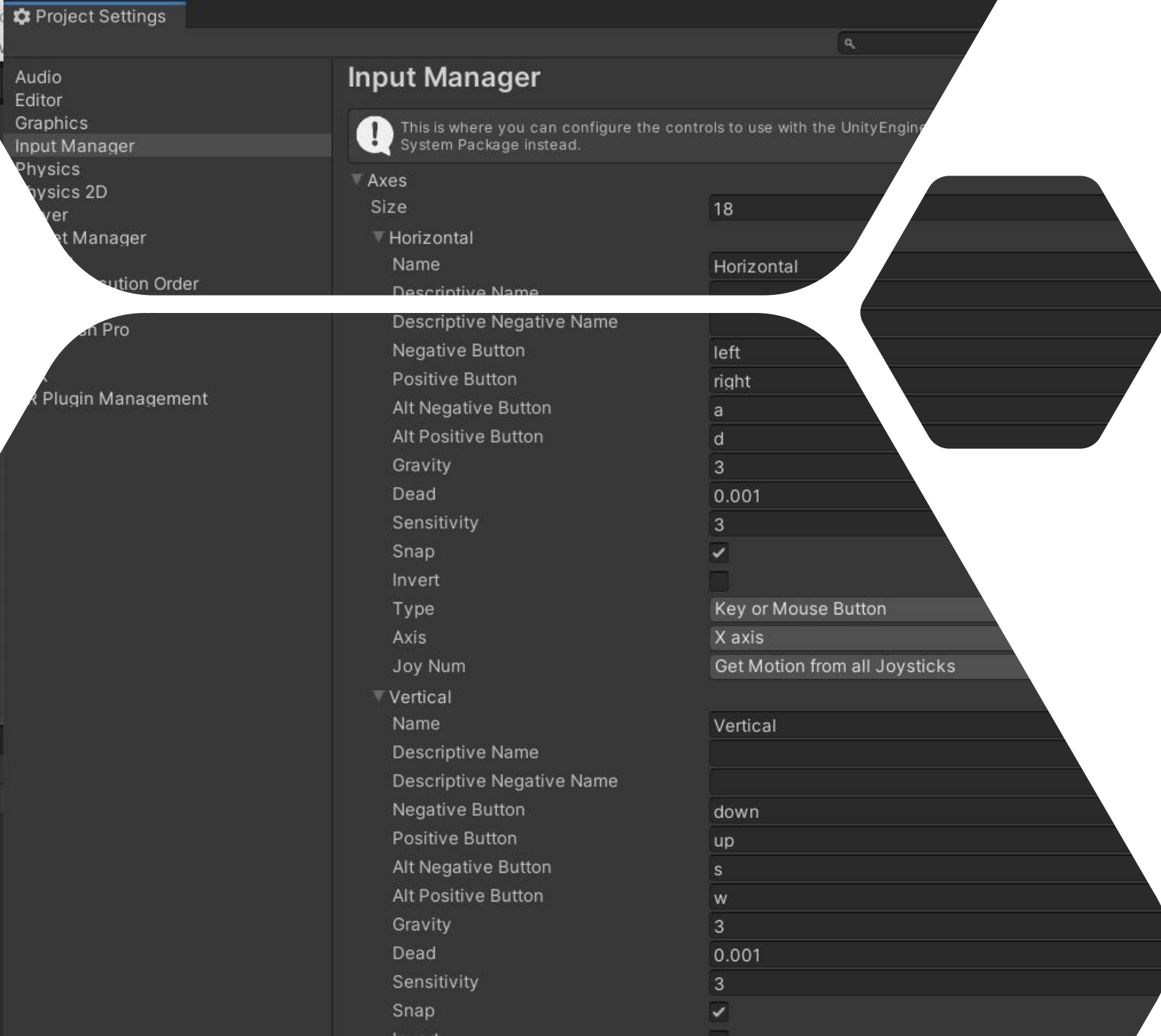
InputKlavyeKontrol.cs

```
void Update()
{
    if (Input.anyKey)
        Debug.Log("Klavye veya mouse tuşuna basılıyor.");
    if (Input.anyKeyDown)
        Debug.Log("Klavye veya mouse tuşuna basıldı.");
}
```





Klavye veya Mouse tuşlarından herhangi birine tıklandığında oluşan Log'lar resimdeki gibidir.



Horizontal & Vertical

- Horizontal
 - Left,a → Negatif
 - Right,d → Pozitif
- Vertical
 - Down,s → Negatif
 - Up,w → Pozitif

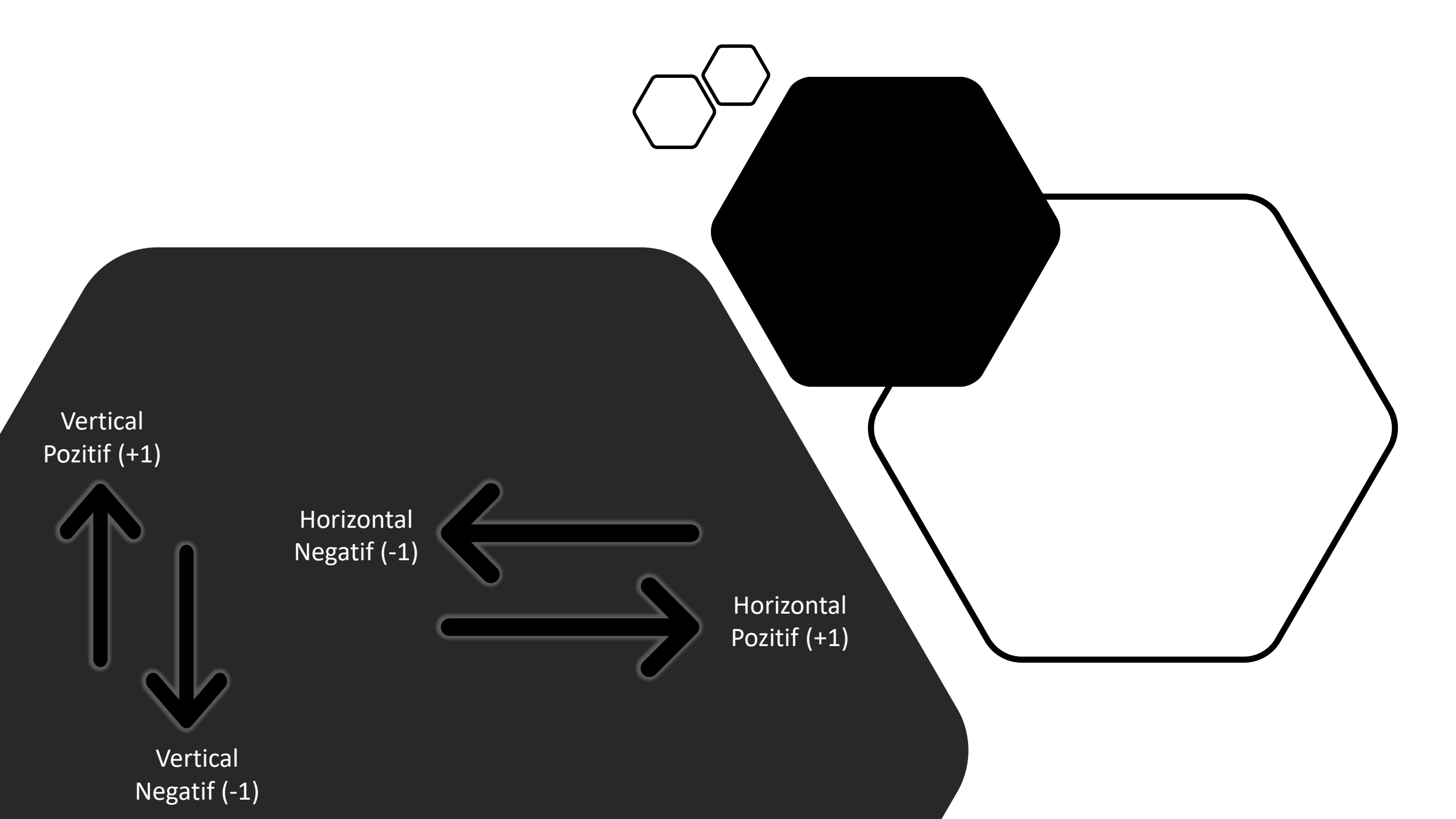


Input.GetAxis

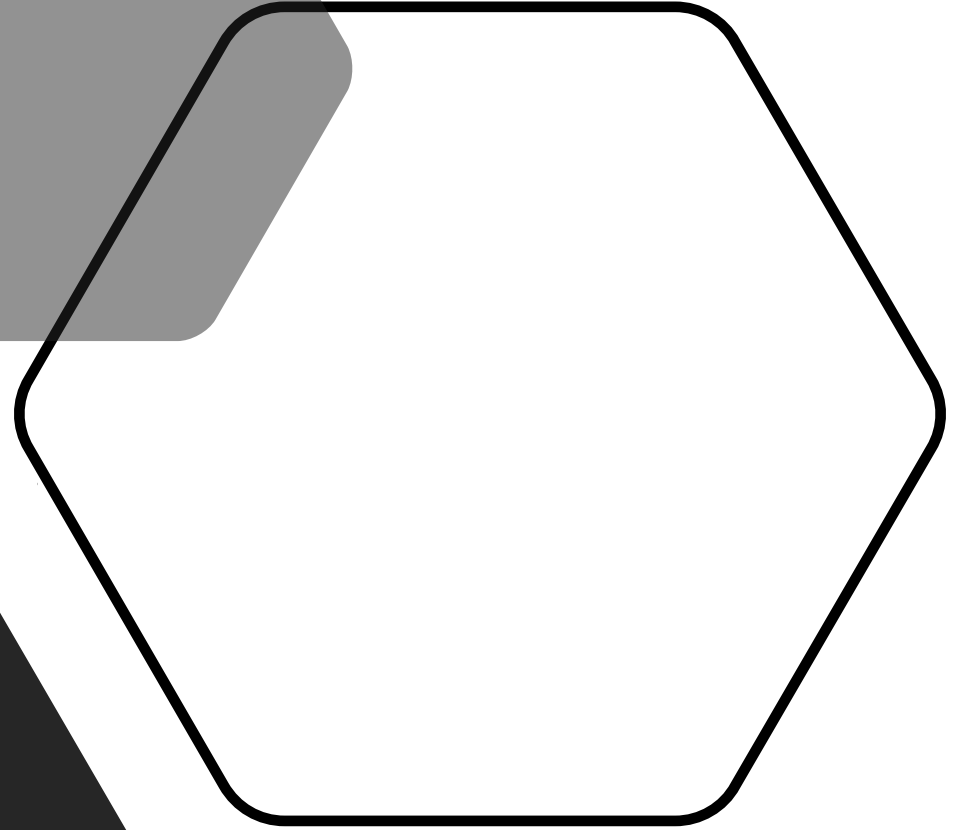
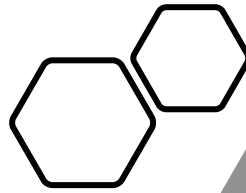
- Bu fonksiyon ile yukarı, aşağı, sağ ve sol tuşlarına klavyeden basılma durumu kontrol edilebilir.
- Input.GetAxis() fonksiyonunda yumuşatma işlemi uygulanmaktadır. Yani dönen değerler float değerlerdir.
- Döndürdüğü değerler minimum -1, maksimum 1 olabilir.
- Kullanımı:
- Input.GetAxis("Axis Adı" : String)
- «Axis Adı» "Horizontal" ya da "Vertical" parametrelerini alabilir.
 - «Vertical» ile yukarı ve aşağı ok tuşlarına basılması kontrol edilebilir.
 - Yukarı ok tuşuna basıldığında pozitif, aşağı ok tuşuna basıldığında negatif, hiçbir tuşa basılmadığında 0 değerlerini döndürür.
 - "Horizontal» ile sol ve sağ ok tuşlarına basılması kontrol edilebilir.
 - Sağ ok tuşuna basıldığında pozitif, sol ok tuşuna basıldığında negatif, hiçbir tuşa basılmadığında 0 değerlerini döndürür.




Horizontal (Yatay)
Vertical (Dikey)




Input.GetAxis



```
void Update()
{
    Debug.Log("Horizontal:" + Input.GetAxis("Horizontal"));
    Debug.Log("Vertical:" + Input.GetAxis("Vertical"));
}
```

The image shows a dark-themed Unity console window. On the left, there is a list of log messages, each preceded by a small icon of an exclamation mark inside a circle. The messages are: [01:20:50] Vertical:0,1241115 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,29622 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,3986436 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,546777 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,6738036 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,784242 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,8042114 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,9998601 UnityEngine.Debug:Log(Object), [01:20:51] Vertical:1 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,8379745 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,7024216 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,6013186 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,4650898 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,3730234 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,458785 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,3151945 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,1685359 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,05957529 UnityEngine.Debug:Log(Object), [01:20:50] Vertical:0,1593148 UnityEngine.Debug:Log(Object). On the right side of the console, there are small circular icons with numbers: 1, 1, 1, 1, 1, 1, 1, 1, 14, 1, 1, 1, 1, 1, 1, 1, 1. A large, semi-transparent white arrow points upwards from the bottom left towards the center of the text. The text is in a large, white, sans-serif font.

Klavyeden Yukarı Ok (↑) Tuşuna Basıldığında Log'lar



Klavyeden Aşağı Ok (↓) Tuşuna Basıldığında Log'lar

[01:23:20] Vertical:0
UnityEngine.Debug:Log(Object)
[01:23:15] Vertical:-0,2046897
UnityEngine.Debug:Log(Object)
[01:23:15] Vertical:-0,3224175
UnityEngine.Debug:Log(Object)
[01:23:16] Vertical:-0,441639
UnityEngine.Debug:Log(Object)
[01:23:16] Vertical:-0,5679819
UnityEngine.Debug:Log(Object)
[01:23:16] Vertical:-0,6711669
UnityEngine.Debug:Log(Object)
[01:23:16] Vertical:-0,7575228
UnityEngine.Debug:Log(Object)
[01:23:16] Vertical:-0,8524950
UnityEngine.Debug:Log(Object)
[01:23:16] Vertical:-0,9613035
UnityEngine.Debug:Log(Object)
[01:23:17] Vertical:-1
UnityEngine.Debug:Log(Object)
[01:23:17] Vertical:-0,8616829
UnityEngine.Debug:Log(Object)
[01:23:17] Vertical:-0,7690405
UnityEngine.Debug:Log(Object)
[01:23:17] Vertical:-0,6914176
UnityEngine.Debug:Log(Object)
[01:23:17] Vertical:-0,6149344
UnityEngine.Debug:Log(Object)
[01:23:17] Vertical:-0,5297734
UnityEngine.Debug:Log(Object)
[01:23:17] Vertical:-0,3959674
UnityEngine.Debug:Log(Object)
[01:23:17] Vertical:-0,3013012
UnityEngine.Debug:Log(Object)
[01:23:17] Vertical:-0,2156065
UnityEngine.Debug:Log(Object)
[01:23:17] Vertical:-0,102622
UnityEngine.Debug:Log(Object)

103

1

1

1

1

1

1

1

1

22

1

1

1

1

1


1

1

1

1

1



Klavyeden Sol Ok (←) Tuşuna Basıldığında Log'lar

UnityEngine.Debug:Log(Object)
[01:25:33] Vertical:0
UnityEngine.Debug:Log(Object)
[01:25:30] Horizontal:-0,1528152
UnityEngine.Debug:Log(Object)
[01:25:30] Horizontal:-0,2649432
UnityEngine.Debug:Log(Object)
[01:25:30] Horizontal:-0,3694818
UnityEngine.Debug:Log(Object)
[01:25:30] Horizontal:-0,4808919
UnityEngine.Debug:Log(Object)
[01:25:30] Horizontal:-0,5742624
UnityEngine.Debug:Log(Object)
[01:25:30] Horizontal:-0,6685704
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,7599831
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,8624313
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,95622
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-1
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,8859373
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,7636375
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,6704815
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,7601506
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,8605714
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,7616026
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,6547828
UnityEngine.Debug:Log(Object)
[01:25:31] Horizontal:-0,7791208

117

1

1

1

1

1

1

1

1

1

1

13

1

1

1

1

1

1

1

1



Klavyeden Sağ Ok (→)
Tuşuna Basıldığında
Log'lar



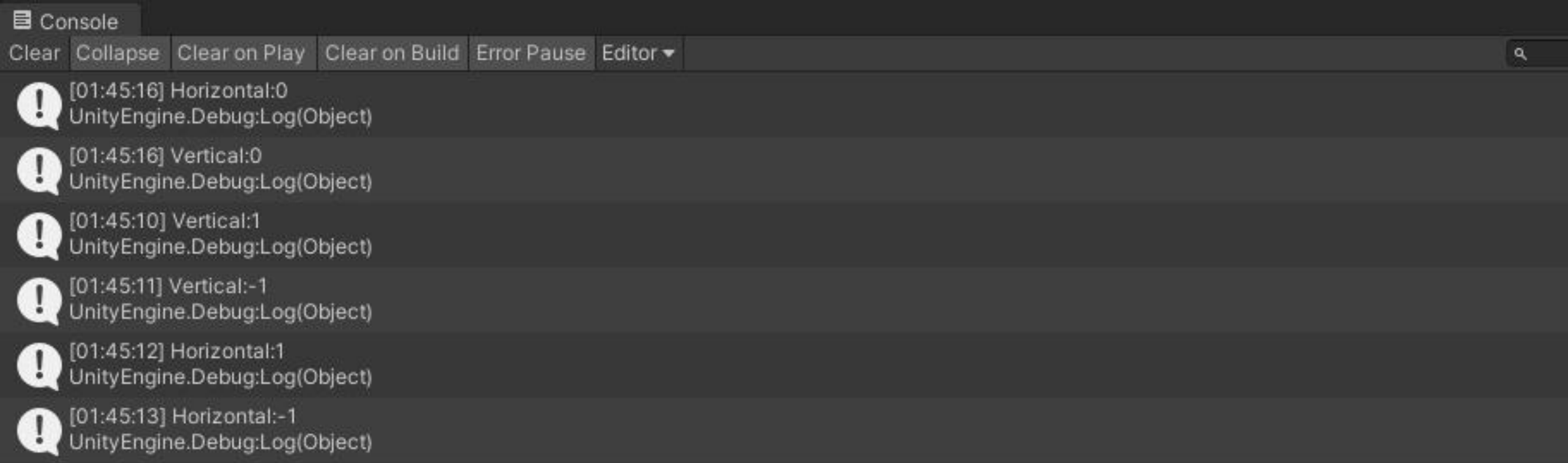
Input.GetAxisRaw

- Bu fonksiyon ile yukarı, aşağı, sağ ve sol tuşlarına klavyeden basılma durumu kontrol edilebilir.
- Input.GetAxisRaw() fonksiyonun GetAxis fonksiyonundan farkı 0,1 ve -1 değerlerini döndürmesidir.
- Kullanımı:
- Input.GetAxisRaw("Axis Adı" : String)
- «Axis Adı» "Horizontal" ya da "Vertical" parametrelerini alabilir.
- «Vertical» ile yukarı ve aşağı ok tuşlarına basılması kontrol edilebilir.
 - Yukarı ok tuşuna basıldığında 1, aşağı ok tuşuna basıldığında -1, hiçbir tuşa basılmadığında 0 değerlerini döndürür.
- "Horizontal» ile sol ve sağ ok tuşlarına basılması kontrol edilebilir.
 - Sağ ok tuşuna basıldığında 1, sol ok tuşuna basıldığında -1, hiçbir tuşa basılmadığında 0 değerlerini döndürür.



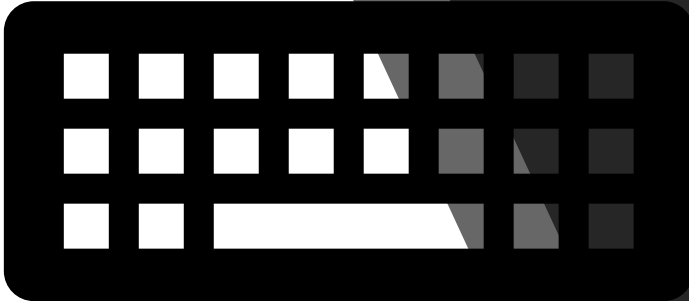
Input.GetAxisRaw

```
void Update()
{
    Debug.Log("Horizontal:" + Input.GetAxisRaw("Horizontal"));
    Debug.Log("Vertical:" + Input.GetAxisRaw("Vertical"));
}
```



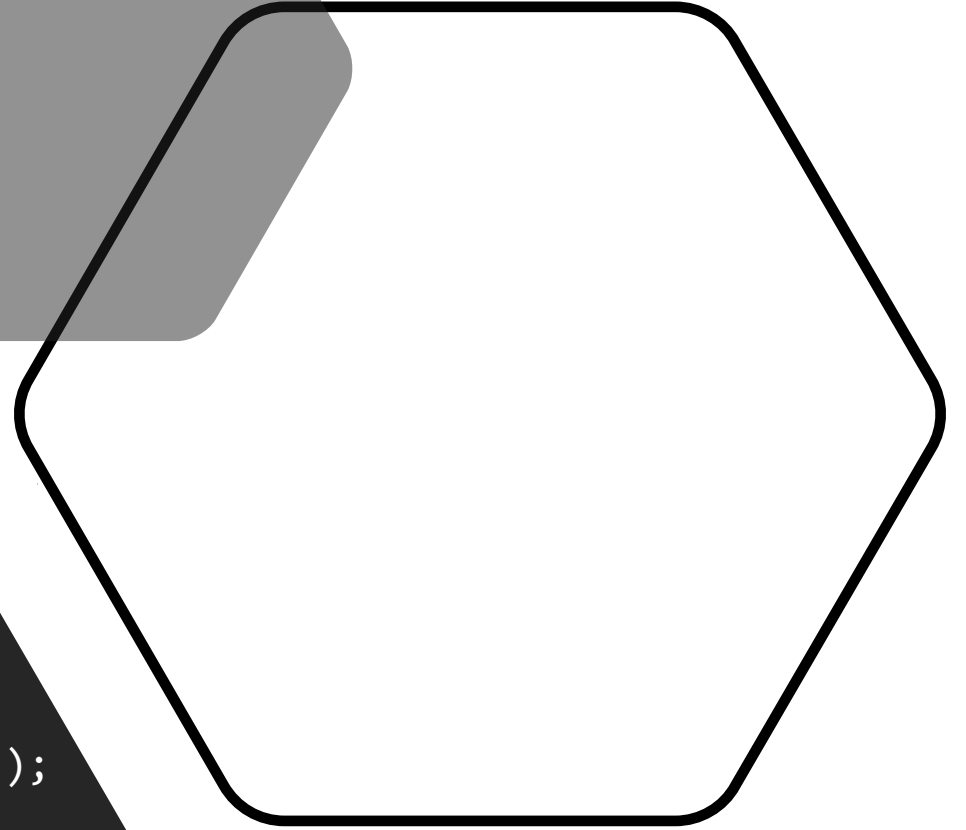
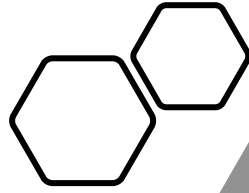
Klavyeden sırasıyla
↑ ↓ → ← tuşlarına |
basıldığında

Input GetButton Metotları



- Klavyeden ya da fareden bir tuşa basmak iki adımda gerçekleşir:
 - GetButton: İlgili tuşa basılı tutulup tutulmadığını kontrol eder.
 - GetButtonDown: Tuşa bastığımız anda gerçekleşen metottur.
 - GetButtonUp: Tuşu bıraktığımız an gerçekleşen metottur.
- Kullanımı:
 - Input.GetButton("Önceden Belirtilmiş Tuşun Özel Adı" : String)
- Ör:
 - Input.GetButtonDown("Fire1") Farenin sol tuşuna veya CTRL tuşuna basıldığı anda true değerini döndürür.
 - Input.GetButtonUp("Fire1") Farenin sol tuşu veya CTRL tuşu bırakıldığı anda true değerini döndürür.

Input GetButton



```
void Update()
{
    Debug.Log("Getbutton" + Input.GetButton("Fire1"));
    Debug.Log("GetButtonDown" + Input.GetButtonDown("Fire1"));
    Debug.Log("GetButtonUp" + Input.GetButtonUp("Fire1"));
}
```

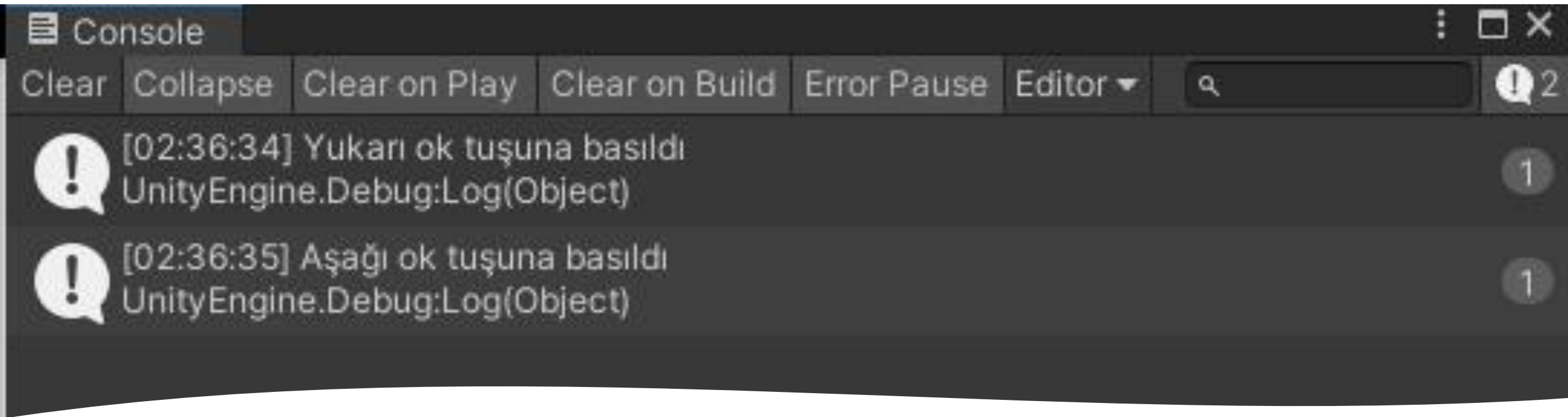



Mouse Sol Tuşuna veya
Klavyeden Ctrl tuşuna
basıldığında Log'lar

Input.GetKey



- Bu fonksiyonda klavyedeki tüm tuşların basılma durumunu kontrol edebiliriz.
- Kullanımı:
 - `Input.GetKey(tusKodu : KeyCode)`
 - `KeyCode` tuş kodu `Esc` tuşu için `KeyCode.Escape`, `Home` tuşu için `KeyCode.Home`, `A` tuşu için `KeyCode.A` vb. gibidir.
- Bu işlem ilgili tuşa basılı tutulduğu sürece gerçekleşir.
- `Input.GetKeyDown` tuşa basıldığı an, `Input.GetKeyUp` ile tuş bırakıldığı an `true` değerler döndürür.



GetKeyDown
Metodu ile Yukarı
Ok ve Aşağı Ok
Tuşlarına Basılması

- void Update()
- {
- if (Input.GetKeyDown(KeyCode.UpArrow))
- Debug.Log("Yukarı ok tuşuna basıldı");
- if (Input.GetKeyDown(KeyCode.DownArrow))
- Debug.Log("Aşağı ok tuşuna basıldı");
- }

Klavye Tuşları ile Objeyi Kontrol Etmek



UzayGemisiKontrol.cs

```
const float hareketgucu = 10;
```

- Öncelikle hareketin hızını belirleriz.
- Hızın sahip olmasını istediğimiz değeri deneyerek bulabiliriz.
- Hız çarpanı oyun oynama sırasında değişmeyeceği için onu sabit (const) olarak tanımlayabiliriz.




UzayGemisiKontrol.cs

```
void Update()
{
    Vector3 position = transform.position;
    float yatayInput = Input.GetAxis("Horizontal");
    float dikeyInput = Input.GetAxis("Vertical");
    if (yatayInput != 0)
        position.x += yatayInput * hareketgucu * Time.deltaTime;
    if (dikeyInput != 0)
        position.y += dikeyInput * hareketgucu * Time.deltaTime;
    transform.position = position;
}
```









Inspector




 ☒ **uzaygemisi** ☐ Static ▾




Tag **Untagged** ▾ Layer **Default** ▾




Prefab **Open** **Select** **Overrides** ▾



▶  **Transform** ?  



▶  ☒ **Sprite Renderer** ?  

▶  **Rigidbody 2D** ?  

▶  ☒ **Box Collider 2D** ?  

▼  ☒ **Uzay Gemisi Kontrol (Script)** ?  

Script  **UzayGemisiKontrol** 

 **Sprites-Default** ? 

▶ Shader **Sprites/Default** ▾

