

Oyun Programlama

Öğr. Gör. Zafer SERİN

PROJE OLUŞTURMA

- Unity3D ile proje oluşturmak için öncelikle Unity Hub indirilmeli ve istenen Unity3D sürümü seçilerek yükleme yapılmalıdır. Genelde LTS(Long Term Support – Uzun Süreli Destek) sürümlerinin yüklenmesi hatalar ile karşılaşma ihtimalini azaltacaktır ve bu sürümlere uzun süreli destek verilecektir. Proje oluştururken 2D 2 boyutlu projeler oluşturmak için 3D ise 3 boyutlu projeler oluşturmak için kullanılır.

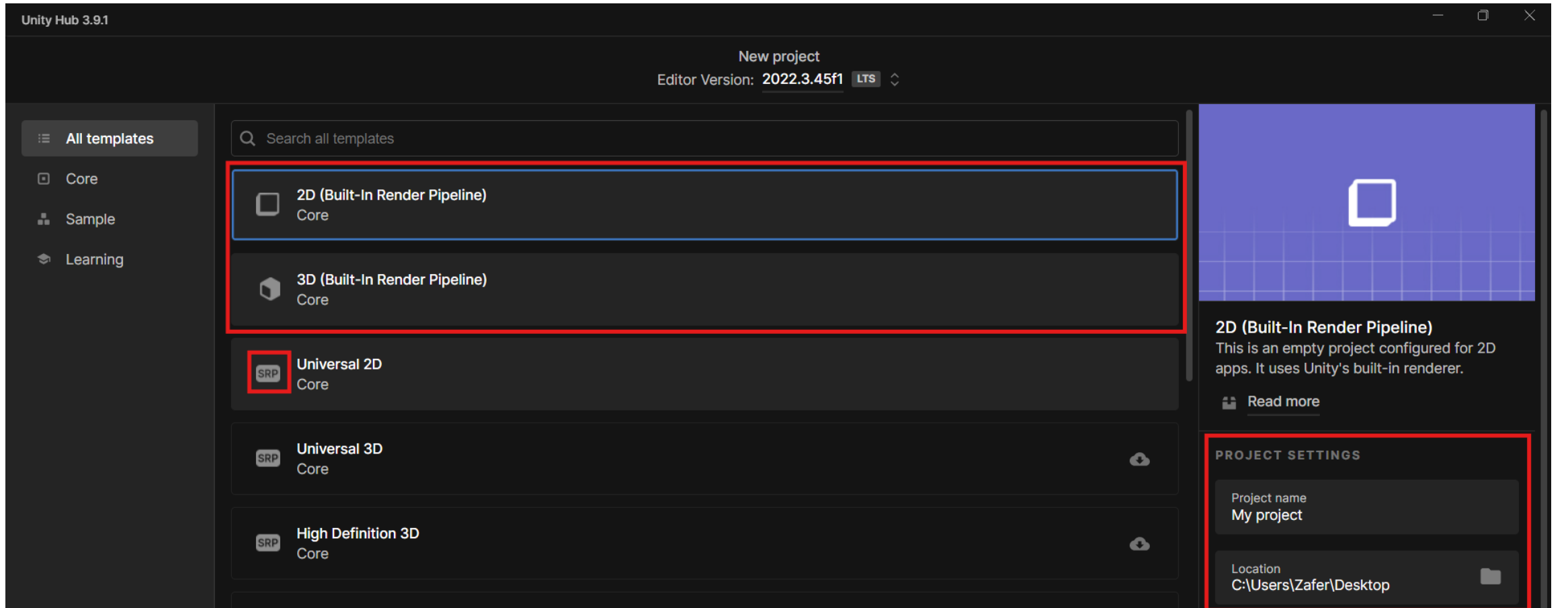
PROJE OLUŞTURMA

- Proje oluştururken SRP(Scriptable Render Pipeline) ve BRP(Built Render Pipeline) terimleri ile karşılaşılabilir. SRP Unity3D'de BRP'nin yerini alan ve geliştiricilere daha fazla kontrol sağlayan bir sistemdir. SRP, geliştiricilerin grafik işleme sürecini kendi ihtiyaçlarına göre özelleştirmelerine olanak tanır.
- Proje oluştururken VR(Virtual Reality) sanal gerçeklik projeleri oluşturmak için, AR(Augmented Reality) artırılmış gerçeklik projeleri oluşturmak için ve MR(Mixed Reality) ise VR ve AR projelerinin birlikte geliştirilmesi için kullanılır.

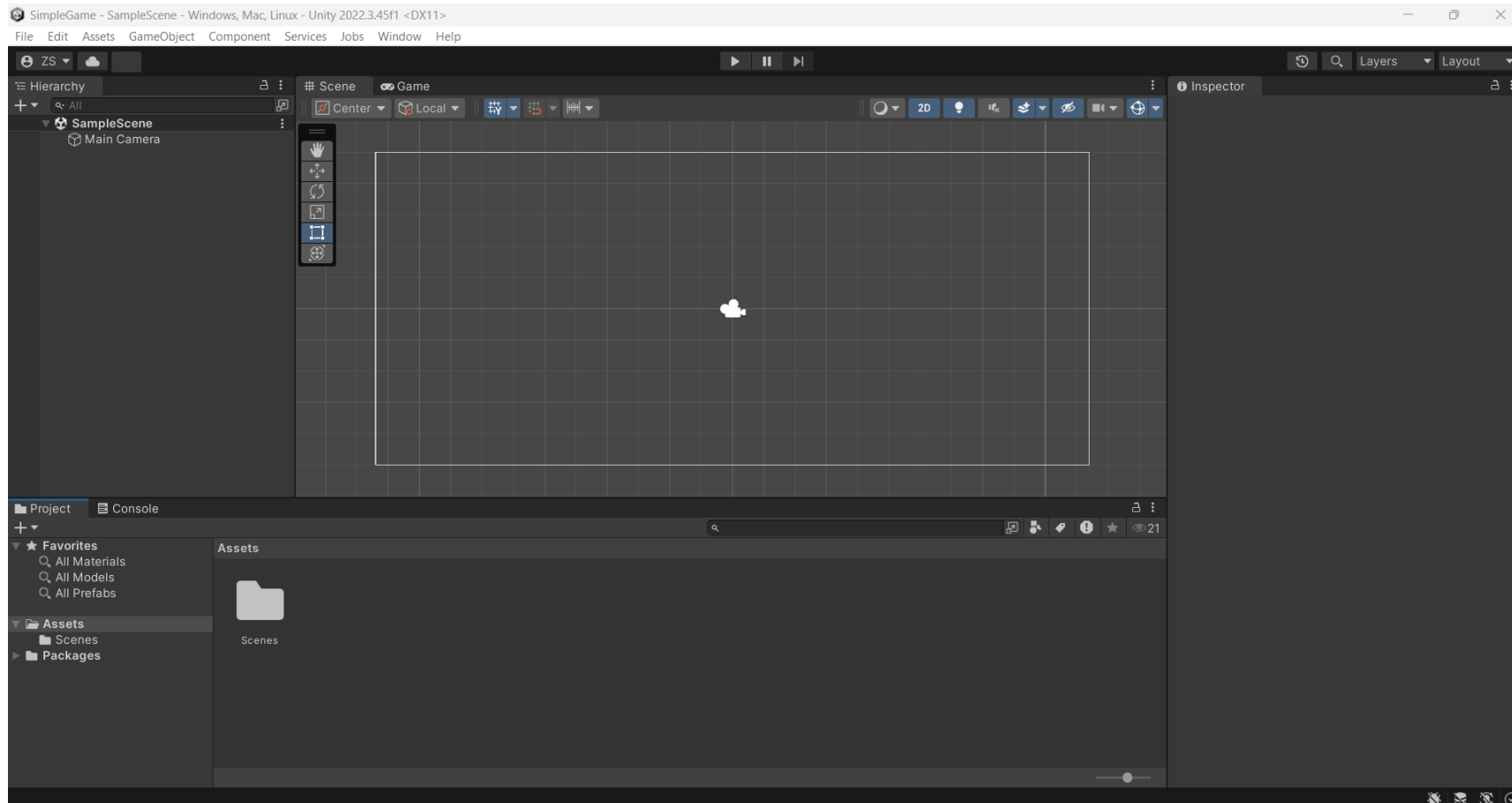
PROJE OLUŞTURMA

- Unity 2022.3.45f1 LTS sürümü için 2 boyutlu projeler oluşturmak için '2D(Built-in Render Pipeline)' ve 3 boyutlu projeler oluşturmak için ise '3D(Built-in Render Pipeline)' seçilebilir. Sağ tarafta ise projenin adı(Project name) ve kaydedileceği yer(Save location) gibi özellikler ayarlanabilir.

PROJE OLUŐTURMA



PROJE OLUŐTURMA



ÖNEMLİ MENÜLER

- File -> Build Settings yolu izlenerek Unity3D ile geliştirilen oyunun ayarlarına erişilebilir. Açılan pencerede kullanılan sahneler, çıktı alınabilecek platformlar gibi pek çok ayara ulaşılabilir.
- File -> Build Setting -> Player Settings yolu izlenerek oyunun platform özellikleri, ikonu, çözünürlüğü, tam ekranda çalışıp çalışmayacağı, açılış logosu gibi pek çok özellik belirlenebilir. Bu menü Project Settings menüsü olarakta isimlendirilir ve proje ile ilgili pek çok ayar bu pencereden yönetilir.

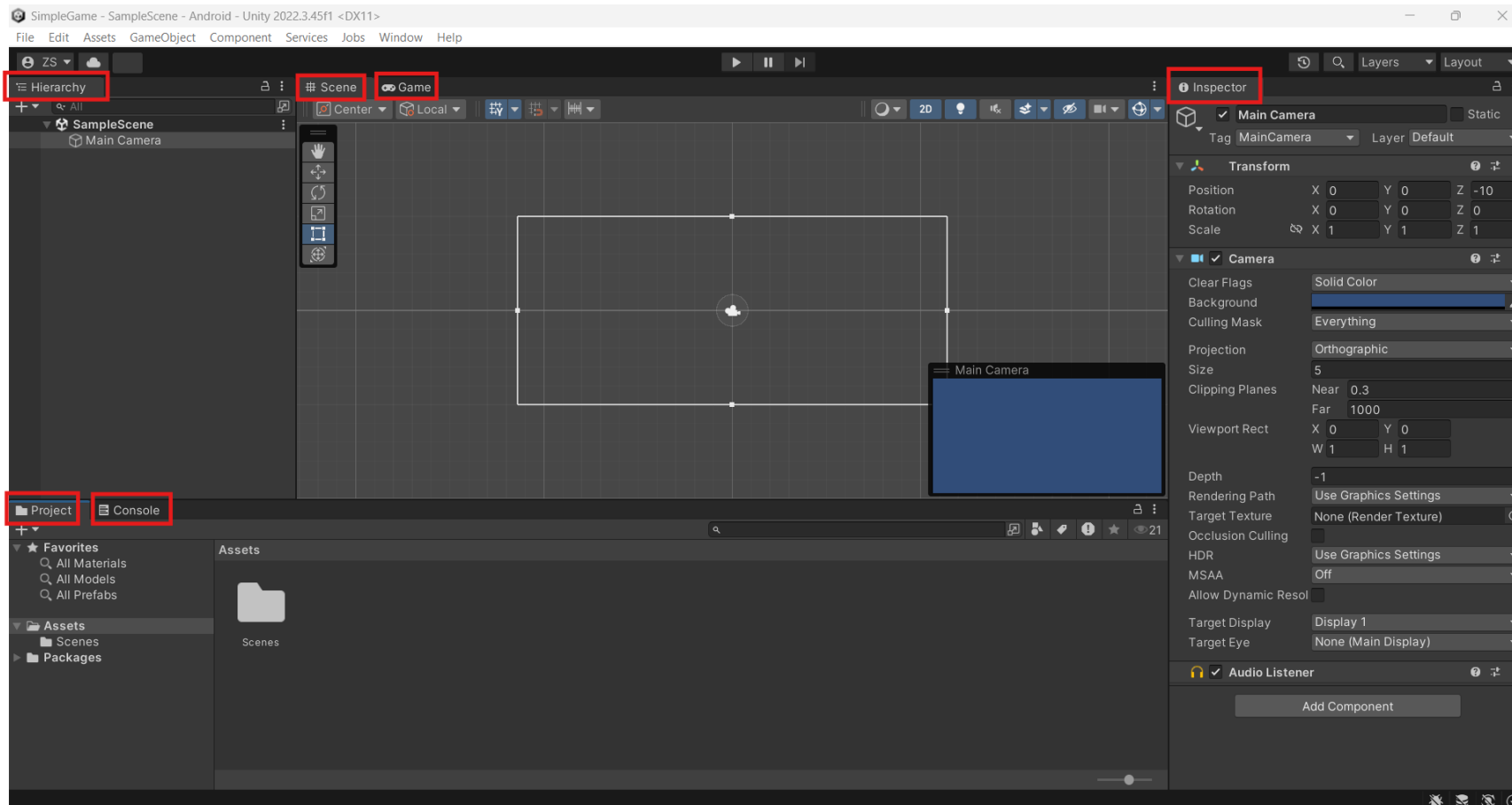
ÖNEMLİ MENÜLER

- Edit -> Preferences yolu izlenerek Editörle ilgili genel ayarlara erişilebilir. Özellikle bu menüdeki 'External Tools' kısmından kullanılmak istenen kod editörü değiştirilebilir. Eğer Android gibi platformlara çıktı alınmak istenirse yine 'External Tools' kısmından JDK, NDK ve SDK yolları belirtilebilir.

ARAYÜZ

- Unity3D arayüzünde Hierarchy penceresi ekranda bulunan nesneleri, Project penceresi projede bulunan dosya ve klasörleri, Console penceresi konsola yazılan çıktıları ve durumları, Inspector penceresi bir nesneye eklenen componentleri veya genel özelliklerini, Scene penceresi oyun sahnesini, Game penceresi ise oyun çalıştırıldığında nasıl görüneceğini gösterir.

ARRAYÜZ

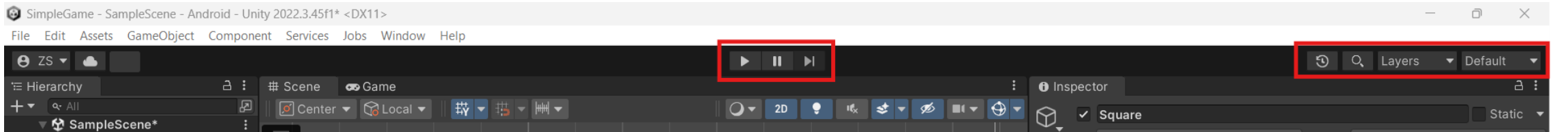


ARAYÜZ

- Unity3D arayüzünde üst menüde 3 adet buton ve ekranın sağ üst köşesinde ek kısımlar görüntülenecektir. Burada ekranın üstündeki 3 buton sırasıyla oyunu başlatmayı, durdurmayı ve adım adım ilerletmeyi sağlar. Burada önemli bir nokta vardır. Oyun çalışırken yapılan değişiklikler kesinlikle geçerli olmaz ve oyun durdurulduğunda tekrar eski haline getirilir. Bu nedenle yapılan değişiklikler deneme amaçlı değilde kalıcı olarak yapılacaksa mutlaka oyun çalışmıyorken yapılmalıdır.

ARAYÜZ

- Üst menüde sağdaki butonlar ise sırasıyla geçmiş işlemler, arama, katmanlar ve arayüzdeki pencereleri konumlandırma için kullanılır.

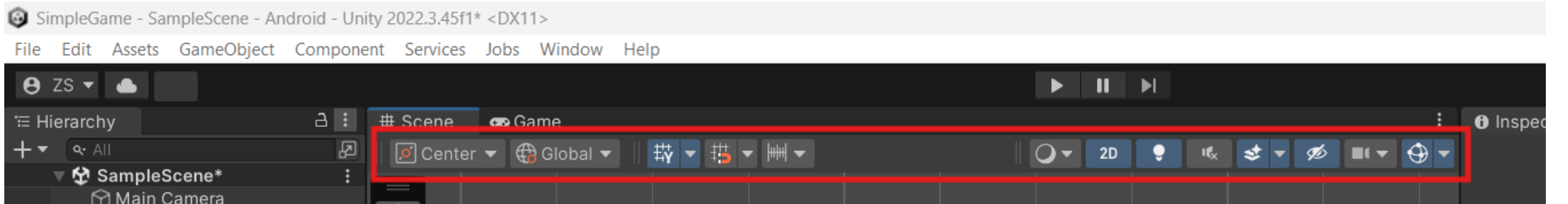


ARAYÜZ

- Unity3D arayüzünde Scene menüsü altında bir bar görünür. Buradaki işlemler sırasıyla: Nesneye uygulanan rotasyon gibi işlemlerin merkezden mi yoksa pivot noktasından mı olacağı, nesnenin konum değiştirme işleminin local mi global mi olacağı, sıradaki 3 buton nesnenin konumunun değiştirilirken ızgaraya sabit kalınarak hizalamanın yapılıp yapılmayacağı, gölgelendirme ve ışıklandırma modları, 2 boyutlu veya 3 boyutlu ortamda çalışılması, sahne ışıklandırmasının kullanılıp kullanılmayacağı, ses modunun açılıp açılmayacağı, sis ve gökyüzü gibi görsellerin gösterilip gösterilmeyeceği

ARAYÜZ

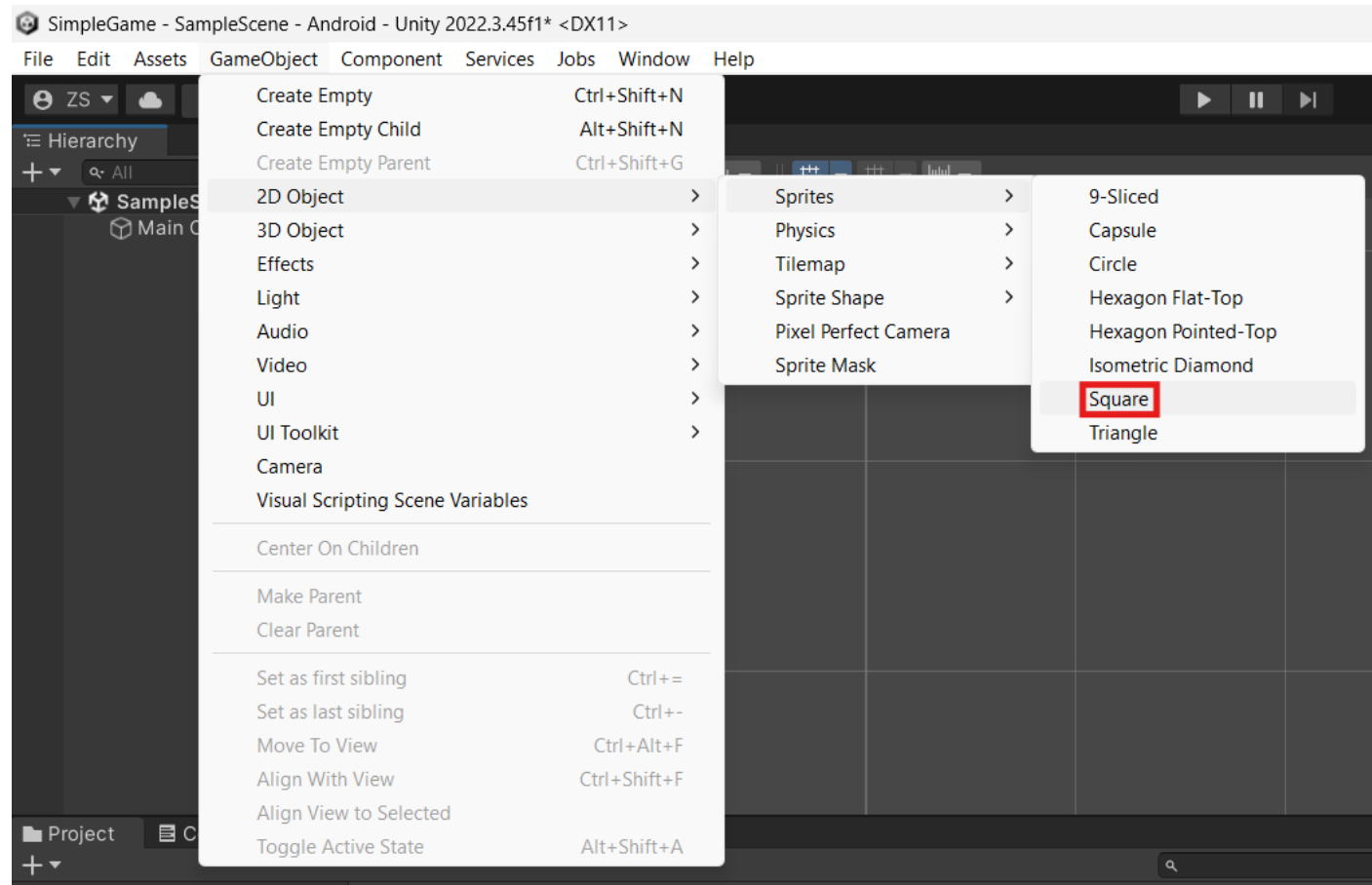
- Gizlenen nesnelerin gösterilip gösterilmeyeceği, ekran kamerasının ayarları ve ekrandaki simgelerin(gizmos) gösterilip gösterilmeyeceği gibi ayarları içerir.



SPRITE VE NESNE OLUŞTURMA

- Unity3D'de basit nesneler hazır olarak bulunur. Hierarchy paneline sağ tıklanarak veya üstteki GameObject menüsü kullanılarak 2 boyutlu ve 3 boyutlu hazır görseller oluşturulabilir. GameObject -> 2D Object -> Sprites -> Square yolu izlenerek 2 boyutlu bir kare sprite oluşturulabilir. Aynı yol izlenerek oluşturulabilecek diğer hazır nesnelerde görüntülenebilir.

SPRITE VE NESNE OLUŞTURMA



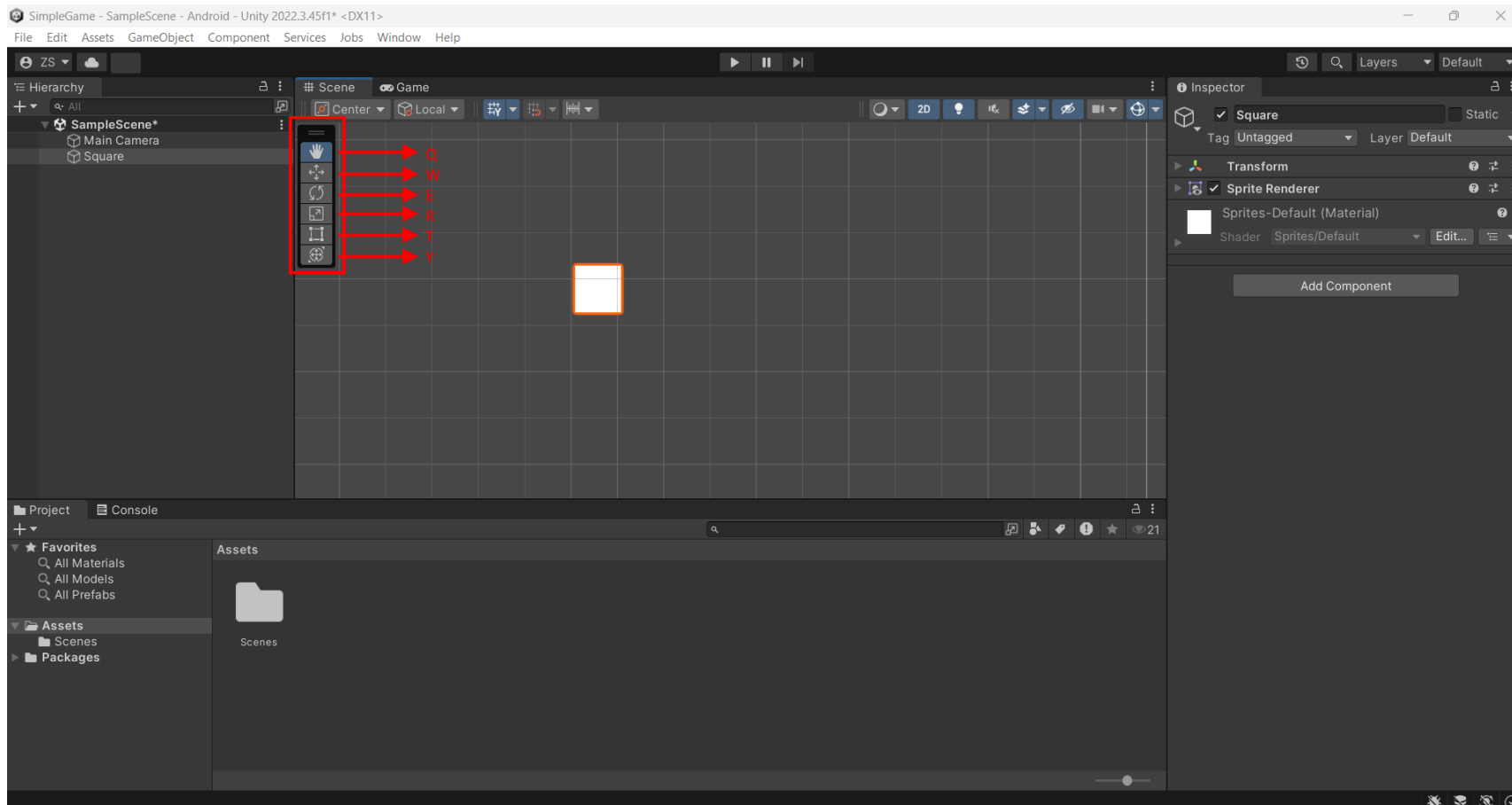
SPRITE VE NESNE OLUŞTURMA

- Hierarchy penceresinde varsayılan olarak gelen 'Main Camera' nesnesi ana kameradır ve oyunun başlatıldığında nasıl görüneceği bu kamera üzerinden görüntülenebilir. Bu kameranın gösterdiği alan oyun da gösterilecektir.
- Hierarchy penceresinde hangi nesne seçilirse Inspector penceresinde de varsayılan olarak onun özellikleri ve componentleri gösterilir.

SPRITE VE NESNE OLUŞTURMA

- Scene penceresinde:
- Q tuşu ile Pan moduna geçilerek sahnede hareket edilebilir.
- W tuşu ile seçilen nesnenin konumu değiştirilebilir.
- E tuşu ile seçilen nesnenin rotasyonu değiştirilebilir.
- R tuşu ile seçilen nesnenin büyüklüğü değiştirilebilir.
- T tuşu ile tüm eksenlerde ortak değişiklik yapılabilir.
- Y tuşu ile W, E ve R tuşu ile yapılan işlemlerin tamamı birlikte yapılabilir; ancak arayüz biraz karmaşık hale gelebilir.
- Yukarıdaki işlemlerin tamamı Scene penceresinde ki sol menü ile de yapılabilir

SPRITE VE NESNE OLUŖTURMA



COMPONENT(BİLEŞEN) KAVRAMI

- Unity3D'de component(bileşen) bir GameObject'in davranışını ve özelliklerini tanımlayan yapıdır. Hierarchy penceresinde bulunan her bir nesne componentler içermektedir. Bu componentler isteğe göre eklenip çıkarılabilir; ancak bütün componentler kaldırılrsa bile Transform componenti kaldırılamaz. Square nesnesinin Inspector penceresi incelenirse şunlar görülür: Transform, Sprite Renderer ve Sprites-Default(Material).

COMPONENT(BİLEŞEN) KAVRAMI

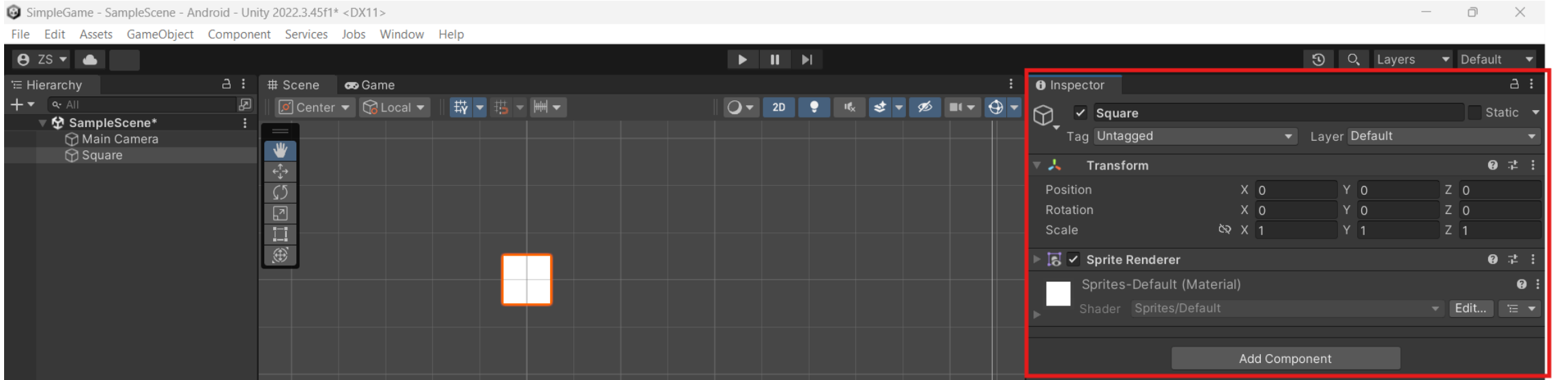
- Burada Transform nesnenin konumunu, rotasyonunu ve büyüklüğünü x, y ve z eksenlerinde tutan componenttir. 2 boyutlu ortam da bir proje açılırsa z eksenini değişimleri net bir şekilde görülemeyebilir. Sprite Renderer nesnenin görüntülenmesini sağlayan componenttir. Sprites-Default(Material) ise nesnenin rengini, kaplamasını vb. içeren bir materyaldir. Bu yapı component gibi Inspector'da görüntülense de teknik açıdan bir component olmadığı söylenebilir.

COMPONENT(BİLEŞEN) KAVRAMI

- Bunun dışında Inspector menüsünde nesne tik işareti ile aktif ve deaktif hale getirilebilir, nesnenin adı değiştirilebilir, nesneye bir tag(etiket) verilerek kod tarafında daha kolay bir ulaşım sağlanabilir, nesnenin layerı(katman) değiştirilebilir, nesneye işaretçi rengi atanarak daha kolay bir şekilde izlenebilir, menü kilitlenebilir ve nesne static olarak işaretlenebilir.
- Unity3D'de sahnede hareket etmeyen, animasyonu olmayan nesneler static olarak işaretlenebilir. Bu optimizasyon ve FPS açısından oldukça olumlu bir etki oluşturur.

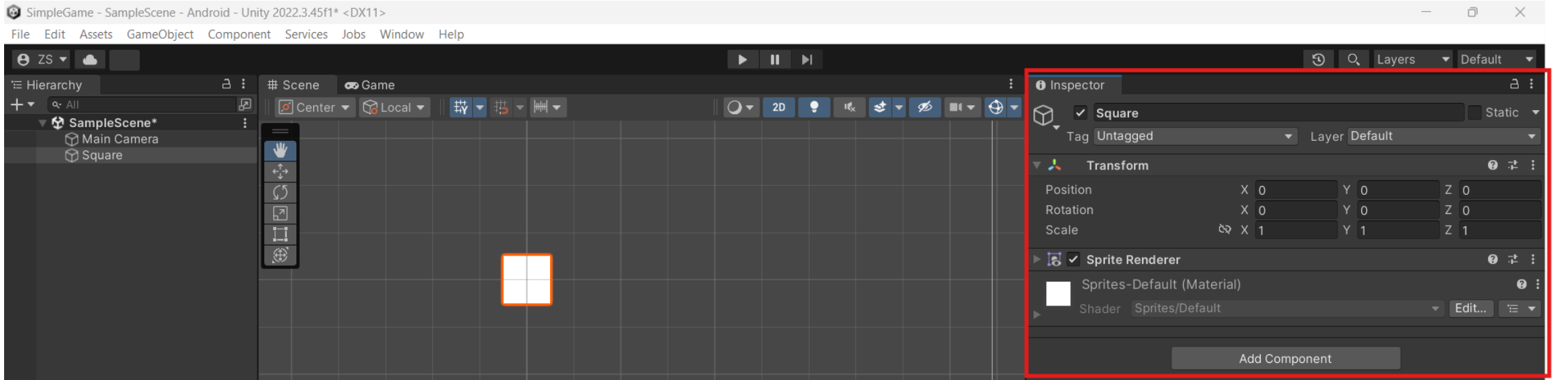
COMPONENT(BİLEŞEN) KAVRAMI

- Inspector menüsünün ve componentlerin yanındaki üç nokta işaretlerine tıklanarak ek ayarlar yapılabilir.



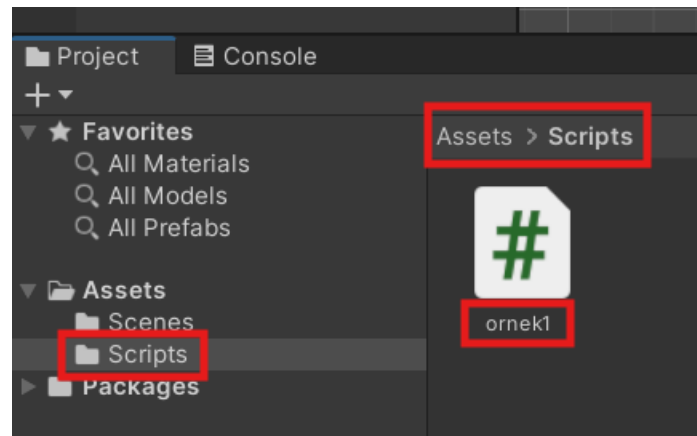
COMPONENT(BİLEŞEN) KAVRAMI

- Inspector menüsünün ve componentlerin yanındaki üç nokta işaretlerine tıklanarak ek ayarlar yapılabilir.



SCRIPT OLUŞTURMA

- Unity3D C# programlama dilini kullanan bir oyun motorudur. Ekranda Project penceresi klasörleme için kullanılan bir penceredir. Burada Scripts adında bir klasör oluşturularak(sağ tık -> Create -> Folder) bu klasör içerisinde sağ tık -> Create -> C# Script yolu izlenerek bir C# kodu oluşturulabilir.



SCRIPT OLUŞTURMA

- Kod çift tıklanarak açılabilir. Unity Visual Studio IDE'sini kullanmaktadır; ancak istenirse Rider veya Visual Studio Code gibi programlarda kullanılabilir. Unity 2022.3.45f1 sürümünde varsayılan olarak gelen kod sonraki sayfada gösterilmiştir.

SCRIPT OLUŞTURMA

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;

4. public class ornek1 : MonoBehaviour
5. {
6.     // Start is called before the first frame update
7.     void Start()
8.     {
9.
10.    }

11.    // Update is called once per frame
12.    void Update()
13.    {
14.
15.    }
16. }
```

SCRIPT OLUŞTURMA

- Varsayılan kodda 1-3. satırlar arası kullanılan namespacesi ifade etmektedir. Bunlar Unity3D tarafından bizlere sağlanan classları, bunların fieldlarını, propertylerini, metotlarını ve indexerlerini içerir. 4. satırda oluşturduğumuz classın adı ve bu classın MonoBehaviour classından kalıtım aldığını görürüz. 6. ve 11. satır yorum satırıdır. Classımızın içerisinde 2 adet metot bulunmaktadır. Bunlar Start(7-10. satır arası) ve Update(12-15. satır arası) metotlarıdır. Bunlar dışında temel olarak 3 metottan daha bahsedilebilir. Bunlar: Awake(), FixedUpdate() ve LateUpdate()

SCRIPT OLUŞTURMA

- Awake(): Unity3D'de ilk ve 1 kez çalışan metottur. Eklendiği nesnenin scripti aktif olmasa bile 1 kez çalışır.
- Start(): Unity3D'de Awake() metodundan sonra oyun başlar başlamaz 1 kez çalışan metottur.
- Update(): Her bir frame de 1 kez çalışan metottur. Sürekli çalışır şeklinde düşünülebilir. Çalışma sayısı bilgisayarın sistem özelliklerine göre değişir. 150 FPS alan bir bilgisayarda saniye de 150 kez çalışırken 10 FPS alan bir bilgisayarda saniye de 10 kez çalışır. Fiziksel işlemler için uygun olmayabilir.

SCRIPT OLUŞTURMA

- FixedUpdate(): Saniye de daima 50 kez(0.02 saniye de 1 kez) çalışır. Bilgisayar 50 FPS veremiyorsa ne olur?
- LateUpdate(): Update()’ten sonra yapılmak istenen işlemler için kullanılır. Bunun dışında Update() ile aynıdır.

SCRIPT OLUŞTURMA

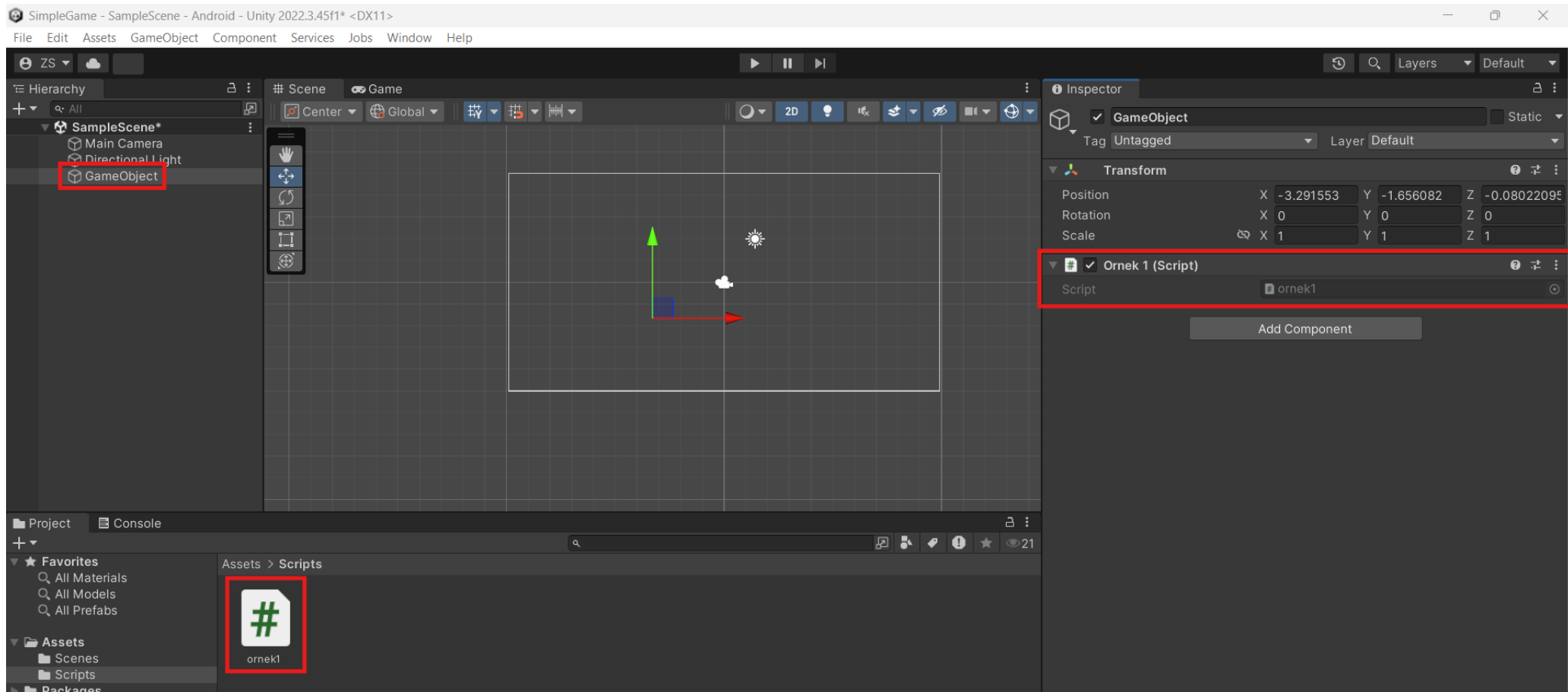
- Unity3D'de bir scriptin aktif olması için bir nesneye sürükle-bırak yoluyla veya 'Add Component' butonu ile eklenmesi gerekir. Bir nesneye eklenmeyen script çalışmayacaktır. Yazılan scriptin adı ile koddaki classın adının aynı olmasına dikkat edilmelidir yoksa hatalar ile karşılaşılabilir.

SCRIPT OLUŞTURMA

- Unity3D'de konsola bir şey yazdırmak için print() metodu kullanılabilir. Start metodu içerisine aşağıdaki kod yazılıp ekrandaki boş bir nesneye eklendiğinde oyun başlatıldığında Console penceresinde ilgili çıktı görünecektir. Scriptlerin de birer component olduğu unutulmamalıdır!

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


SCRIPT OLUŞTURMA



SCRIPT OLUŖTURMA

