

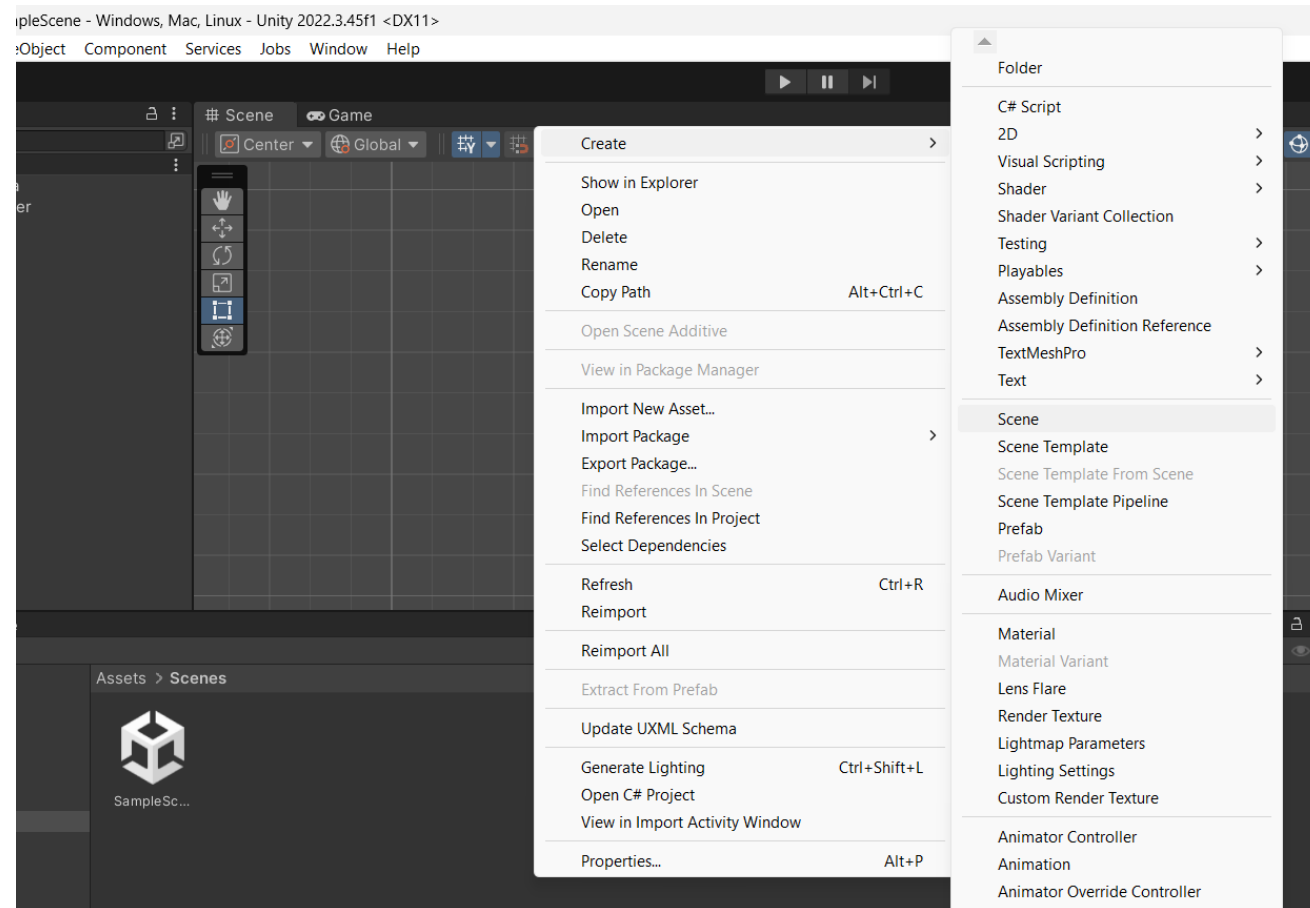
Oyun Programlama

Öğr. Gör. Zafer SERİN

SAHNE(SCENE) EKLEME

- Unity3D'de yeni bir sahne(scene) eklemek için File -> New Scene yolu izlenebilir veya Project penceresinde istenilen konuma sağ tıklanarak Create -> Scene yolu izlenebilir. Böylece yeni bir sahne oluşturulmuş olur.

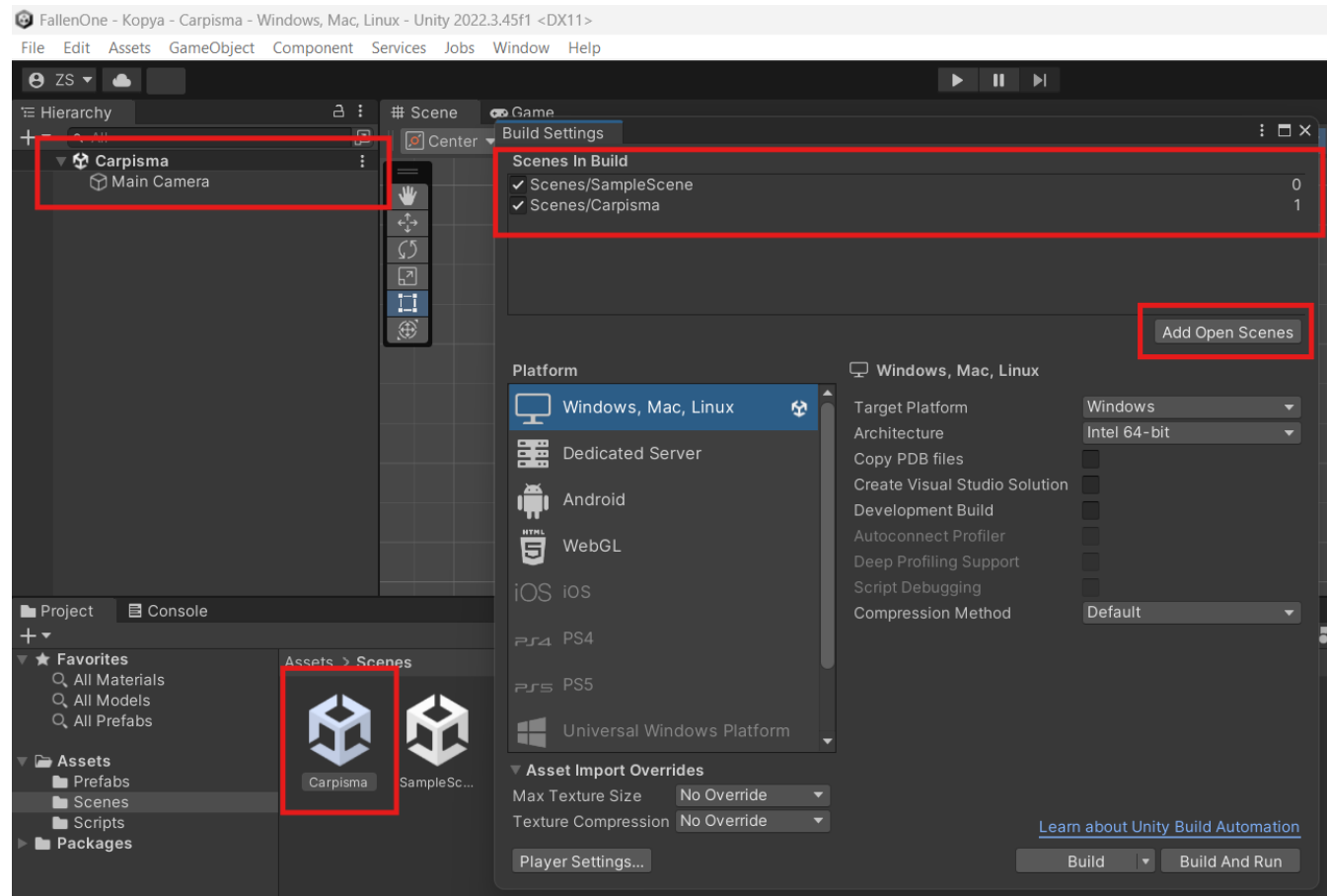
SAHNE(SCENE) EKLEME



SAHNE(SCENE) EKLEME

- Eklenen sahne çift tıklanarak açıldıktan sonra File -> Build Settings -> Add Open Scenes yolu izlenerek Scenes in Build kısmına eklenmesi tavsiye edilir. Zorunlu olmamakla birlikte bu işlem yapılmazsa sahneler arası geçişte problemler yaşanacaktır. Sahnenin buraya eklenmesi önemlidir. Bu işlem sonrası sahneye adı ile veya verilen index numarası ile erişilebilir.

SAHNE(SCENE) EKLEME



SAHNE(SCENE) EKLEME

- Eklenen sahne çift tıklanarak açıldıktan sonra File -> Build Settings -> Add Open Scenes yolu izlenerek Scenes in Build kısmına eklenmesi tavsiye edilir. Zorunlu olmamakla birlikte bu işlem yapılmazsa sahneler arası geçişte problemler yaşanacaktır. Sahnenin buraya eklenmesi önemlidir. Bu işlem sonrası sahneye adı ile veya verilen index numarası ile erişilebilir.

SAHNE(SCENE) EKLEME

- Klavyeden 'S' tuşuna basıldığı anda bu yeni sahneyi yüklemek için aşağıdaki kod bloğu kullanılabilir.

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class SahneGecisi : MonoBehaviour
{
    private void Update()
    {
        if(Input.GetKeyDown(KeyCode.S))
        {
            SceneManager.LoadScene(1);
            // SceneManager.LoadScene("Carpisma");
        }
    }
}
```

ÇARPIŞMA KONTROLÜ

- Unity3D'de çarpışma kontrolü için kullanılan 6 adet önemli metot bulunmaktadır. Bunlar:
- `OnCollisionEnter2D(Collision2D collision)`: 2 boyutlu ortamda scripti atadığımız nesneye başka bir nesne çarptığı anda 1 kez çalışır. Burada `Collision2D collision` parametresi scripti atadığımız nesneye çarpan nesnenin özelliklerini referans eder. Nesne'de Collider olmalıdır.
- `OnCollisionStay2D(Collision2D collision)`: 2 boyutlu ortamda scripti atadığımız nesneye başka bir nesne çarptığı sürece çalışır. `Rigidbody2D`'nin `Sleeping Mode`'u önem arz eder.

ÇARPIŞMA KONTROLÜ

- Unity3D'de çarpışma kontrolü için kullanılan 6 adet önemli metot bulunmaktadır. Bunlar:
- `OnCollisionExit2D(Collision2D collision)`: 2 boyutlu ortamda scripti atadığımız nesneye bir nesne çarptıktan sonra temas kesildiği anda 1 kez çalışır.
- `OnTriggerEnter2D(Collider2D collision)`: 2 boyutlu ortamda scripti atadığımız nesnenin alanına girildiğinde 1 kez çalışır. Bunun için `BoxCollider2D`'nin `isTrigger` özelliği açılmalıdır.

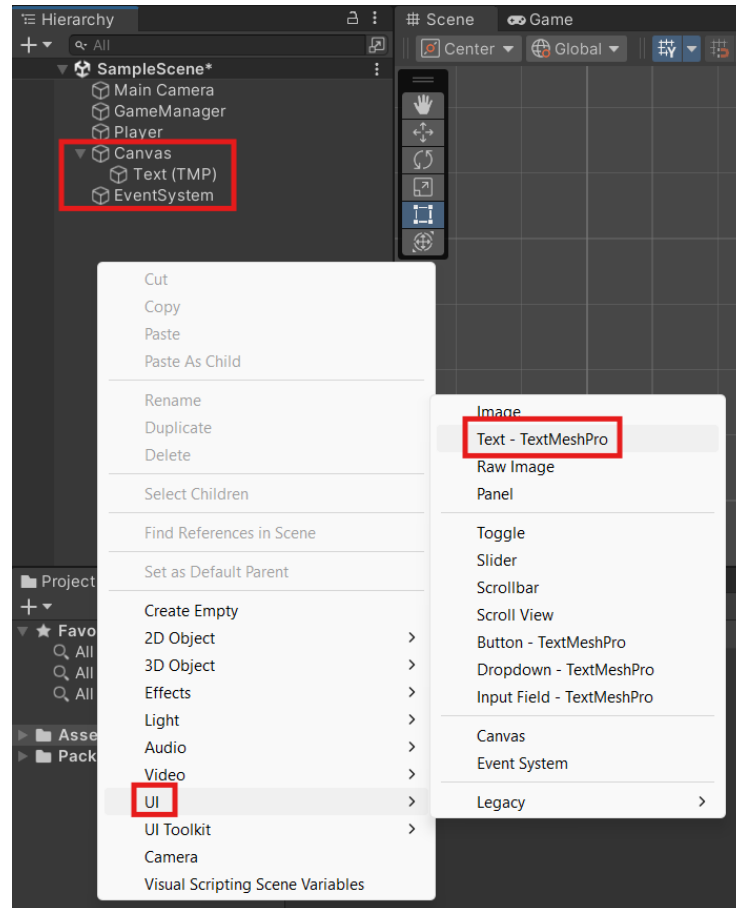
ÇARPIŞMA KONTROLÜ

- Unity3D'de çarpışma kontrolü için kullanılan 6 adet önemli metot bulunmaktadır. Bunlar:
- `OnTriggerStay2D(Collider2D collision)`: 2 boyutlu ortamda scripti atadığımız nesnenin alanına bir başka nesne girip orada kaldığı sürece çalışır. Yine `Rigidbody2D`'nin `Sleeping Mode`'u önemlidir.
- `OnTriggerExit2D(Collider2D collision)`: 2 boyutlu ortamda scripti atadığımız nesnenin alanına bir başka nesne girip oradan çıktığı anda 1 kez çalışan metottur.

NESNE VE COMPONENT AKTİFLİĞİ

- Bir nesneyi aktif veya deaktif hale getirmek için `SetActive()` metodu kullanılır. Bu metot `true` parametresi alırsa nesne aktif hale gelirken `false` parametresi alırsa nesne deaktif hale gelir. Scripti atadığımız nesneyi ifade etmek için `gameObject` propertysi kullanılırken bir başka nesneyi ifade etmek için `GameObject` classı kullanılabilir.
- Bir componenti aktif ve deaktif hale getirmek için `enabled` propertysi kullanılır. Bu propertye `true` değeri atanırsa component aktif hale gelirken `false` değeri atanırsa component deaktif hale gelir.

NESNE VE COMPONENT AKTİFLİĞİ



NESNE VE COMPONENT AKTİFLİĞİ

- Unity3D'de bir nesneyi yok etmek için Destroy() metodu ve overloadları kullanılabilir.
- Unity3D'de Hierarchy penceresine sağ tıklanarak UI kısmından arayüz elemanları eklenebilir. Herhangi bir UI elemanı eklendiğinde Canvas ve EventSystem nesneleri de otomatik olarak eklenecektir. Burada Canvas bir tuval gibi düşünülebilir. Canvas UI elemanlarının konumunu, büyüklüğünü vb. gösteren bir yapıdır. EventSystem ise kullanıcı girdilerini almak için kullanılabilecek bir nesnedir.