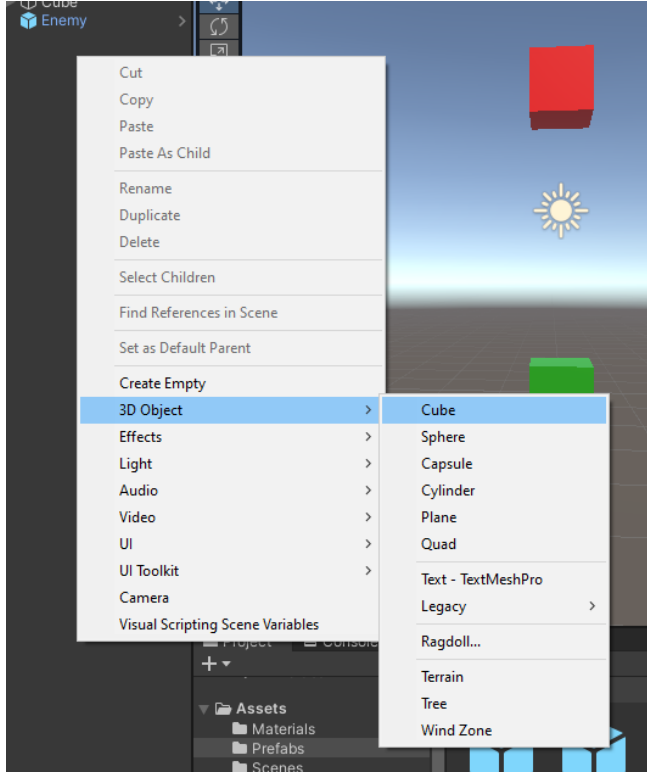


3. Hafta Raporu – Aleks Dulda 21360859025

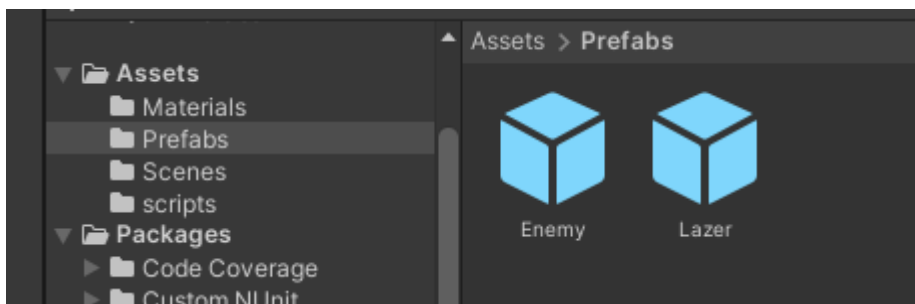
Enemy prefab oluřturma:



Hierarchy kısmından 3D obje olarak Cube öęesini ekliyoruz. Bunu Mouse saę tıkını kullanarak -> 3D obje -> Cube olarak seçebilirsiniz.

Oluřturduęumuz Cube öęesini enemy olarak adlandırabilirsiniz.

Oluřturduęumuz Cube'ü Hierarchy den Project penceresinde önceden oluřturduęumuz "Prefabs" klasörüne sürüklenerek prefab haline getirdik



Bu řekilde prefabımızı oluřturmuř olduk.

Enemy hareketini sağlayan kodun yazılması:

```
Unity Betiği (1 varlık başvurusu) | 0 başvuru
public class Enemy_Sc : MonoBehaviour
{
    [SerializeField]
    public float speed=3f; //hareket süresi
    [SerializeField]
    private Player_Sc player; //Kalıtım alındı

    Unity İletisi | 0 başvuru
    void Start()
    {
        player=GameObject.FindObjectOfType<Player_Sc>();
    }
}
```

private Player_Sc player; //Kalıtım alındı ve Playerin dinamik olarak ekranın dışına çıkmaması için gereken verileri player_sc kodlarından çekmiş olduk.

```
// Update is called once per frame
Unity İletisi | 0 başvuru
void Update()
{
    transform.Translate(Vector3.down * speed * Time.deltaTime);

    if(transform.position.y < -5.4f)
    {
        float randomX = Random.Range(-player.xVal, player.xVal);
        transform.position = new Vector3(randomX, (player.yVal+1), 0);
    }
}
```

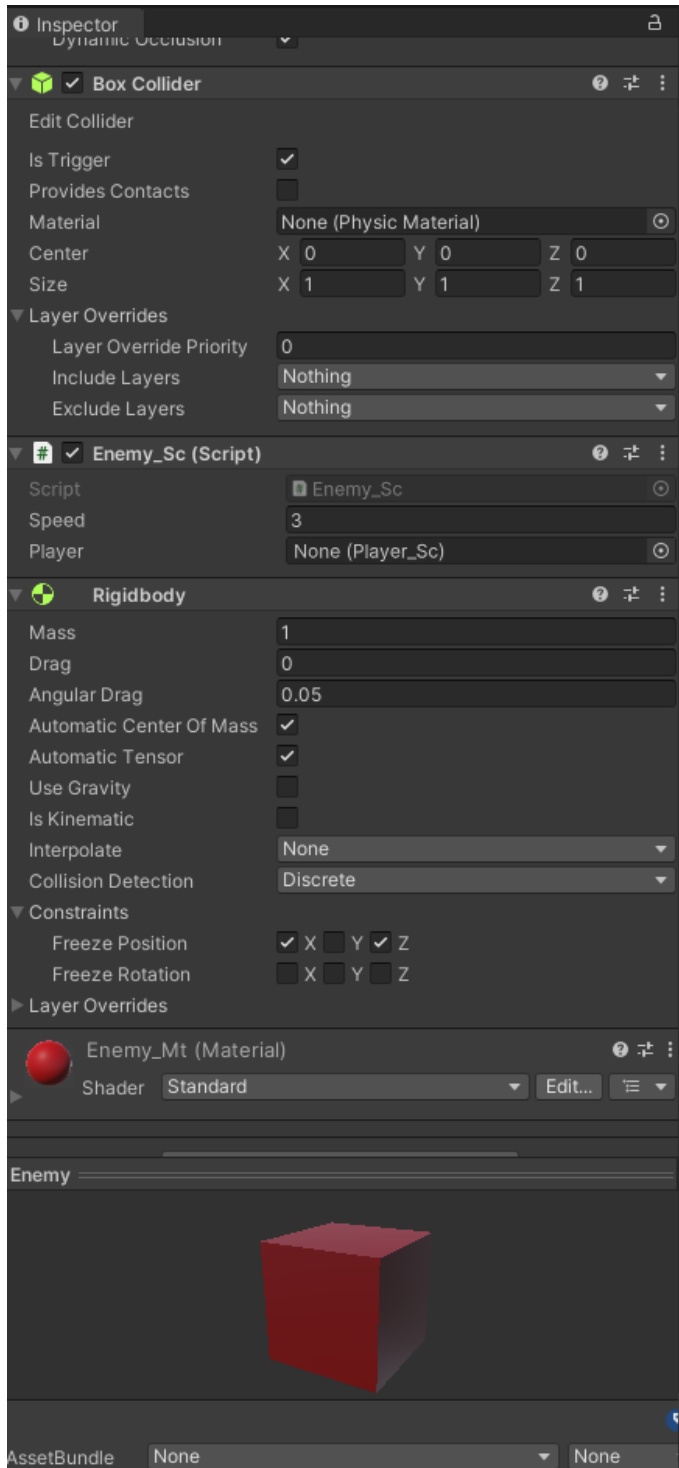
Bu kısımda

transform.Translate(Vector3.down * speed * Time.deltaTime); komutuyla enemy objemizin hareketi sağlandı.

If koşulu sayesinde oluşacak enemy onjesinin rastgele belirli sınırlar içersinde random olarak oluşmasını sağladık.

Sınırları kalıtım aldığımız Player_Sc scriptinde belirlediğimiz sınırları kullandık.

Rigidbody ve Collision component'lerinin eklenmesi:

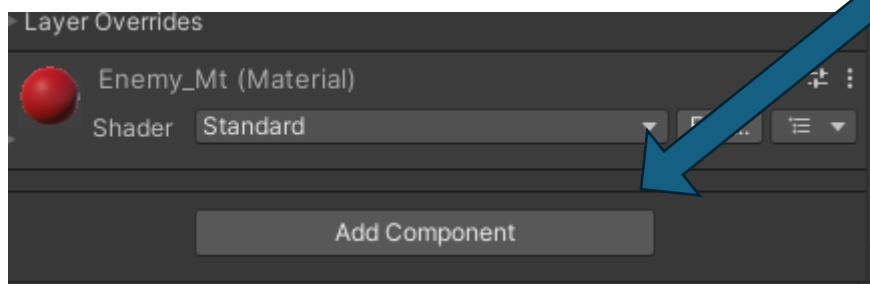


Collider ve Rigidbody kullanarak objelerin hareketi ve gerekli temasların kontrollerinin sağlanması için bu componentleri ekledik.

Collider kısmında “In Trigger” check box’ını işaretliyoruz. Ki objeler birbirinin içinden geçebilsin ve temasları kontrol edilebilsin.

Rigidbody kısmında ise “Use Gravity” kapattık ve “Freeze Position” kısmından X ile Z kordinatlarını Kilitledik ki sadece Y ekseninde hareket edebilsin.

Bu Component’leri Inspector penceresinin en alt kısmında bulunan “Add Component” kısmından ekleyebilirsiniz.



OnTriggerEnter fonksiyonu ve çarpışmaların tespiti ile gerekli aksiyonların alınması:

```
void OnTriggerEnter (Collider other) //temasa giren
{
    if(other.tag == "Player")
    {
        player.Damage();
        Destroy(gameObject); //player'i yok et
    }
    else if(other.tag == "Lazer")
    {
        Destroy(other.gameObject); //lazeri yok et
        Destroy(gameObject); // enemyi yok et
    }
}
```

Bu kısımda “OnTriggerEnter” fonksiyonu ile Objelere eklediğimiz “Tag” etiketleri sayesinde objelerin birbirine olan etkileşimlerini kontrol ederek gerekli koşullara göre işlemler sağlanmakta.

İlk koşulda Player ile Enemy temasında oluşacak durum mevcuttur. Player ile Enemy temasında Player’ın canlarından 1 birim eksilmesi için “Damage” fonksiyonu çağırılmıştır ve Player’i yok etmektedir

Diğer koşulda ise Lazer ile Enemy karşılaşmasında oluşacak durum mevcuttur. Lazer ile Enemy temasta bulununca **önce** lazeri (çünkü enemy önce yok olursa kodların devamı gelmez) sonra Enemy yok edilir.

Oyuncu hasar alma işlemlerinin yapılması:

```
1 başvuru
public void Damage()
{
    Health--;
    if (Health <= 0)
    {
        Debug.Log("GAME OVER");
        Destroy(gameObject);
    }
}
```

Oyuncunun hasar alabilmesi fonksiyonunu Player_Sc dosyasında bu Damage fonksiyonu ile yapılmıştır.

Bu fonksiyon bir önceki sayfada anlatılan koşullar içerisinde çağırılıp player'in canını bir birim azaltmakta ve eğer player'in canı kalmaz ise oyunu bitirmektedir.

Bu şekilde bu hafta için istenenler yapılmış oldu.

Kaynakça

[https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))

<https://chatgpt.com/>

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

Github link

<https://github.com/AleksDulda/GamePrograming>

https://github.com/AleksDulda/GamePrograming/tree/main/Rapor/Week_3