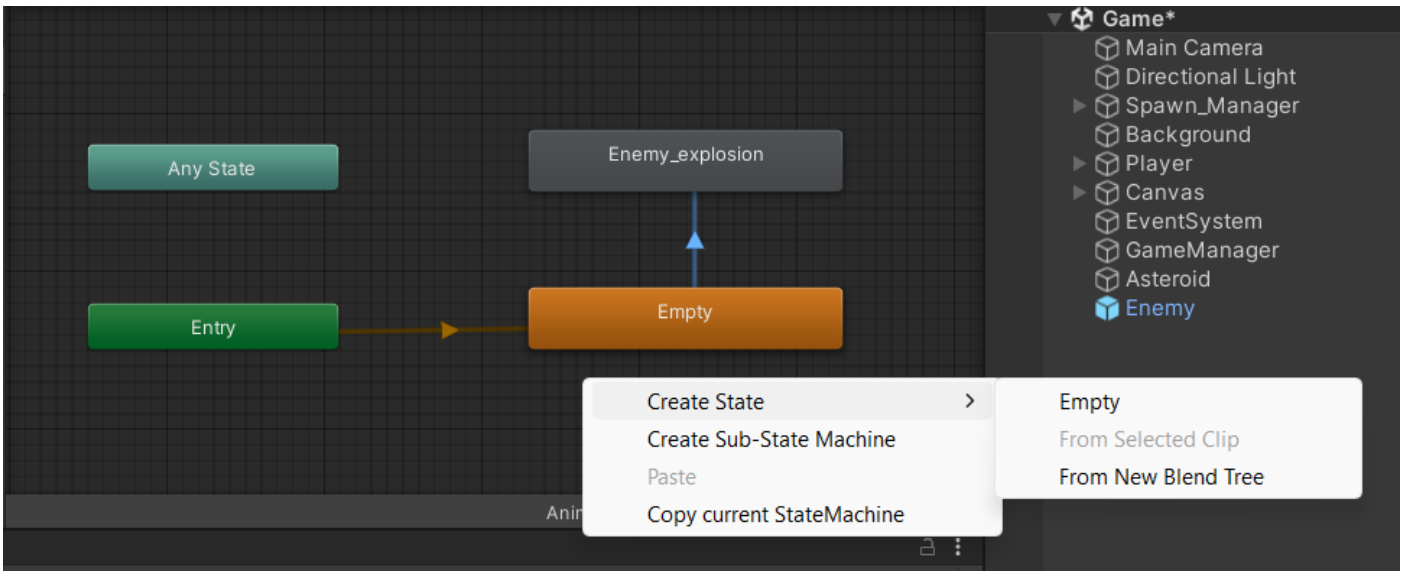


Enemy Yok Etme Animasyonu

Öncelikle enemy prefab dosyanızı seçin ve animasyon penceresinden bir animasyon oluşturun. Daha sonra buraya sprites klasöründeki patlama animasyonlarını ekleyin.

Ardından enemy scriptinize gelerek yok olma anında olmaları belirleyin. Bunun için önce bir animatör değişkeni oluşturun ve animatör değişkeninin atamasını yapın. Daha sonra ontrigger fonksiyonunda yok olma anlarına önce ileride kullanacağımız OnPlayerDeath tetikleyicisi aktifleştirin. Enemy yok olduğunda hareketini hızını sıfırlayarak durdurun ve destroy fonksiyonu ile animasyon süresi bitince yok olacak şekilde ayarlayın. Bunu hem lazer ile hem player çarpışması ile yok olma durumlarında ayarlayın.

Daha sonra animatör penceresine gelin. Bu pencerede Enemy için oluşturduğunuz patlama animasyonunun açmalısınız. Bunu yaptıktan sonra boş bir state oluşturun ve geçişleri bağlayın.



▼ Enemy_Explode_Sequence

4 references

Animator anim;

anim = GetComponent<Animator>();

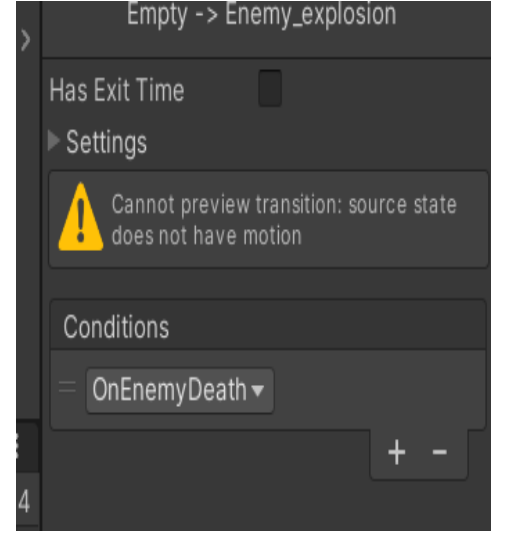
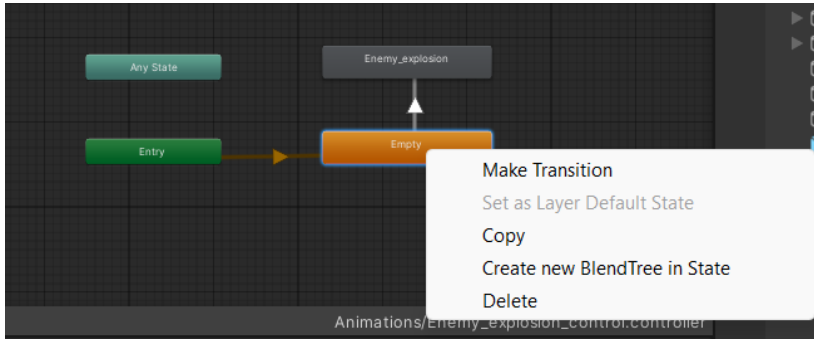
```
void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Player"){
        player.Damage();
        anim.SetTrigger("OnEnemyDeath");
        this.speed = 0;
        Destroy(this.gameObject, 2.0f);
    }
    else if(other.tag == "Laser"){
        /* Always destroy the object of script which you are coding
        /* So.. destroy laser first then destroy enemy
        Destroy(other.gameObject);

        if(player != null)
        {
            player.UpdateScore(10);
        }
        anim.SetTrigger("OnEnemyDeath");
        this.speed = 0;
        Destroy(this.gameObject, 2.0f);
    }
}
```

Daha sonra OnPlayerDeath triggerini şekildeki gibi layers kısmından oluşturun ve seçimini kaldırın.



Oluşturduğunuz bu bağlantılardan enemy explosion animasyonu ile empty arasındaki bağlantıyı seçin ve açılan pencereden has exit time seçeneğini kapatın, ardından condition olarak oluşturduğunuz OnPlayerDeath triggerini ekleyin.



Asteroid Ekleme

Öncelikle sprites klasöründen asteroid assetini alın ve sahnenize sürükleyin. Daha sonra asteroid_sc adında bir script dosyası oluşturun. Daha sonra kullanmak üzere bu 3 değişkeni oluşturun.

Ardından Asteroidin oyun başladığında sürekli dönmesi için z ekseninde dönüşünü ayarlayın. Bunu update fonksiyonu içinde aşağıdaki gibi yapın.

```
transform.Rotate(Vector3.forward*rotationSpeed*Time.deltaTime);
```

Oyun asteroid yok edildiğinde başlayacağı için spawn manager nesnesi ile bağlantı kurulmalı. Bu sebeple oluşturduğunuz spawnManager değişkenine start fonksiyonunda gereken atamayı yapın.

```
spawnManager = GameObject.Find("Spawn_Manager").GetComponent<SpawnManager_sc>();
```

```
1 reference
[SerializeField] float rotationSpeed = 20.0f;

1 reference
[SerializeField] GameObject explosionPrefab;

4 references
[SerializeField] SpawnManager_sc spawnManager;

0 references
```

Daha sonra spawn manager dosyasında düzenleme yapın. Spawn rutinlerinizi start fonksiyonundan alın ve onları çağırarak bir fonksiyonun içine alın. Böylelikle asteroid yok olduğuna bu fonksiyon çağırılacak.

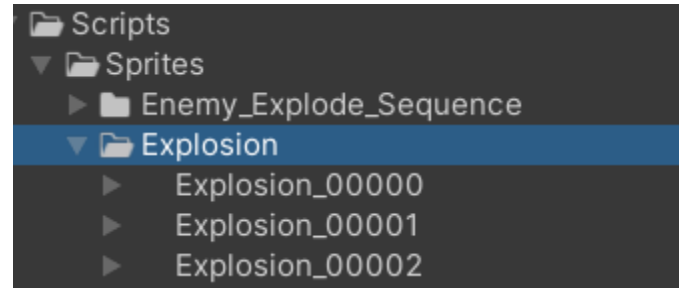
```
2 references
public void Spawner()
{
    StartCoroutine(SpawnRoutine());
    StartCoroutine(SpawnBonusRoutine());
}
```

Son olarak patlama animasyonu eklenecek.

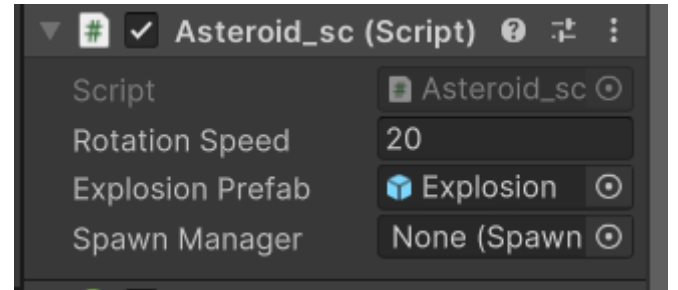
Öncelikle sprites klasöründen bir explosion_0000 assetini sahneye sürükleyin. Ardından explosion_sc oluşturun ve bu nesne ile bağlayın. Explosion_sc içinde sadece bu nesneyi yok eden fonksiyon çağırılacak.

```
0 references
void Start()
{
    Destroy(this.gameObject, 2.3f);
}
```

Sahnenizdeki nesnenizin adını explosion olarak değiştirin ve animasyon oluşturma tıklayıp sprites klasörünün içindeki explosion assetlerinin hepsini animasyona sürükleyin ve kaydedin. Daha sonra sahenizdeki nesnenizi prefab klasörüne sürükleyin ve prefab haline getirin.



Şimdi bu asteroid nesnesini seçin ve daha önceden oluşturduğunuz explosionPrefab değişkeni ile explosion prefabi bağlayın. Asteroide rigidbody2d ve circle collider2d eklentilerini ekleyin. Colliderdan edit collider ile colliderinizi asteroidiniz şekline göre ayarlayın ve isTrigger eçeneğini işaretleyin. Daha sonra rigidbody2d eklentisinden gravity scale değerini 0 yapın.



Son olarak asteroid_sc ye gelin ve OnTriggerEnter2d fonksiyon ekleyin ve şekildeki eklemeleri yapın.

```
0 references
public void OnTriggerEnter2D(Collider2D other){
    if( other.tag == "Laser"){
        Instantiate(explosionPrefab,transform.position,Quaternion.identity);
        Destroy(other.gameObject);
        spawnManager.Spawner();
        Destroy(this.gameObject);
    }
}
```

Bu şekilde asteroid laser ile patlatıldığında yok olma animasyonu başlayacak ve bitişide spawner başlatılacak ve oyun başlayacak.