1. Düşman Gemisi Patlama Animasyonu

Animation Controller Ayarları

1. **Animator Controller Oluşturma:**
    - *Window → Animation → Animator* üzerinden bir Animator Controller oluşturun.
    - Düşman gemisi için yeni bir Animator Controller ekleyin ve EnemyExplosion adında yeni bir *State* oluşturun.
2. **Loop Time ve Exit Time Ayarları:**
    - EnemyExplosion animasyon klibini seçin.
    - **Inspector** penceresinden:
        - Loop Time **kapalı** olmalı (patlama bir kez oynatılmalı).
        - Has Exit Time **açık** olarak ayarlanmalı.
    - **Transition Duration:** 0.1 saniye gibi kısa bir süre ayarlayın.
3. **Transition Eklemek ve Trigger Ayarlamak:**
    - Animator penceresinde:
        - **Default State**'i "Idle" veya "Moving" gibi normal hareket durumlarına ayarlayın.
        - EnemyExplosion state'i ile Idle state'i arasında bir **Transition** ekleyin.
    - **Trigger Ekleme:**
        - Animator'da Trigger parametresi oluşturun (örneğin: Explode).
4. **Trigger'ın Script ile Ayarlanması:**

---

2. Nesne Yok Etmede Gecikme Eklenmesi

- Yukarıdaki kod örneğinde Destroy(gameObject, 2f); ile nesne yok edilmeden önce 2 saniye gecikme ayarlanmıştır.
- Bu, patlama animasyonunun oynatılması için gerekli süreyi sağlar.

```
0 references
public class ExplosionSC : MonoBehaviour
{
    // Start is called before the first frame update
    0 references
    void Start()
    {
        Destroy(this.gameObject, 2.5f);
    }
```

---

3. Hız Sabitleme ile Zarar Vermenin Önlenmesi

- Düşman gemisinin yok edilmesi sırasında hızını sıfırlayarak çarpışmaları önleyin.

```
Debug.Log(other.transform.name+ "çarpıştı");
if(other.tag == "Player"){
    speed = 0;
    PlayerSC player = other.transform.GetComponent<PlayerSC>();
    player.Damage();
    Debug.Log("Player Health"+ player.health);
    Destroy(this.gameObject);


}
if(other.tag == "Bullet"){
    speed = 0;
    Destroy(other.gameObject);
    if(playerScoreCont != null){
        playerScoreCont.ScoreUp(10);
    }
    Destroy(this.gameObject);
}
```

---

4. Asteroid Sistemi

Asteroid Ekleme

1. **Prefab Oluşturma:**
    - Bir asteroid 3D modeli (örneğin bir küre) ekleyin.
    - **Collider** ekleyin (Sphere Collider).
    - **Rigidbody** ekleyin ve Use Gravity **kapalı** olmalı.
2. **Z Ekseninde Sürekli Hareket Ettirme:**
3. **Asteroid Patlama Animasyonu (Prefab ile Dinamik Ekleme):**
    - Patlama animasyonu için ayrı bir prefab oluşturun.
    - Asteroid çarpıştığında bu patlama efektini sahneye ekleyin:

```
public class AstroidSC : MonoBehaviour
{
    // Start is called before the first frame update
    [SerializeField]
    1 reference
    private float _rotateSpeed = 20.0f;
    [SerializeField]
    1 reference
    GameObject ExplosionAnimation;
    [SerializeField]
    2 references
    SpawnManagerSC _spawnManager;
    0 references
    void Start()
    {
        _spawnManager = GameObject.Find("SpawnManager").GetComponent<SpawnManagerSC>();
    }

    // Update is called once per frame
    0 references
    void Update()
    {
        transform.Rotate(Vector3.forward * _rotateSpeed * Time.deltaTime);
    }
    0 references
    void OnTriggerEnter2D(Collider2D other){
        if(other.tag == "Player"){
            PlayerSC player = other.transform.GetComponent<PlayerSC>();
            if(player != null){
                player.Damage();
            }
            Destroy(this.gameObject);
        }
        if(other.tag == "Bullet"){
            _spawnManager.StartSpawning();
            Destroy(other.gameObject);
            Instantiate(ExplosionAnimation, transform.position, Quaternion.identity);
            Destroy(this.gameObject);
        }
    }
```

---

5. Spawn Routineleri

Asteroid yok edildikten **3 saniye** sonra yeni asteroid spawn edilmesi:

Asteroid yok olduğunda çağırın:

```
1 reference
IEnumerator SpawnEnemies()
{
    // Düşmanları her spawnInterval sürede bir üretiyor
    yield return new WaitForSeconds(3f);
    while (spawnActive)
    {
        SpawnEnemy();
        yield return new WaitForSeconds(spawnInterval);
    }
}
```

---

6. Thruster ve Hasar Animasyonu

1. **Sağ ve Sol Motor için Animasyon Ekleme:**
    - Animator'da RightEngineDamaged ve LeftEngineDamaged adında state'ler ekleyin.
    - Trigger parametreleri (RightDamage ve LeftDamage) oluşturun.
2. **Trigger'ları Script ile Çağırmak:**

```
2 references
public void Damage()
{
    health--;
    if (health == 2)
    {
        rightEngine.SetActive(true);
    }
    else if (health == 1)
    {
        leftEngine.SetActive(true);
    }
    if (health <= 0)
    {
        Debug.Log("Oyuncu öldü!");
        spawnManager.StopSpawning(); // Spawn işlemi durduruluyor

        Destroy(gameObject); // Oyuncu yok ediliyor
    }
    uiManager.UpdateLivesImg(health);
}
```

Github: https://github.com/emirrdvn/OyunProgramlama