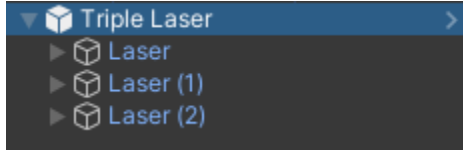


6.Hafta Ders Raporu – Batuhan Şengül

3'lü Atış

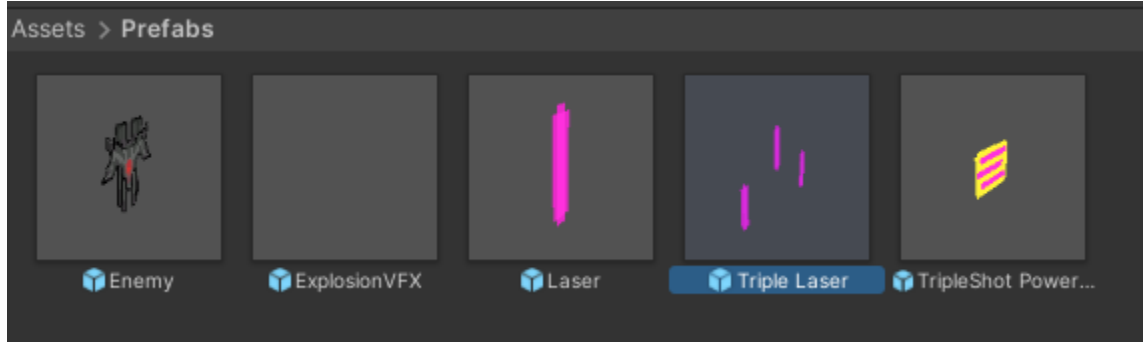
3'lü Lazer Prefabi



“Triple Laser” adlı boş bir nesne oluşturulup içine 3 adet “Laser” prefabi eklenip konumları ayarlanır.

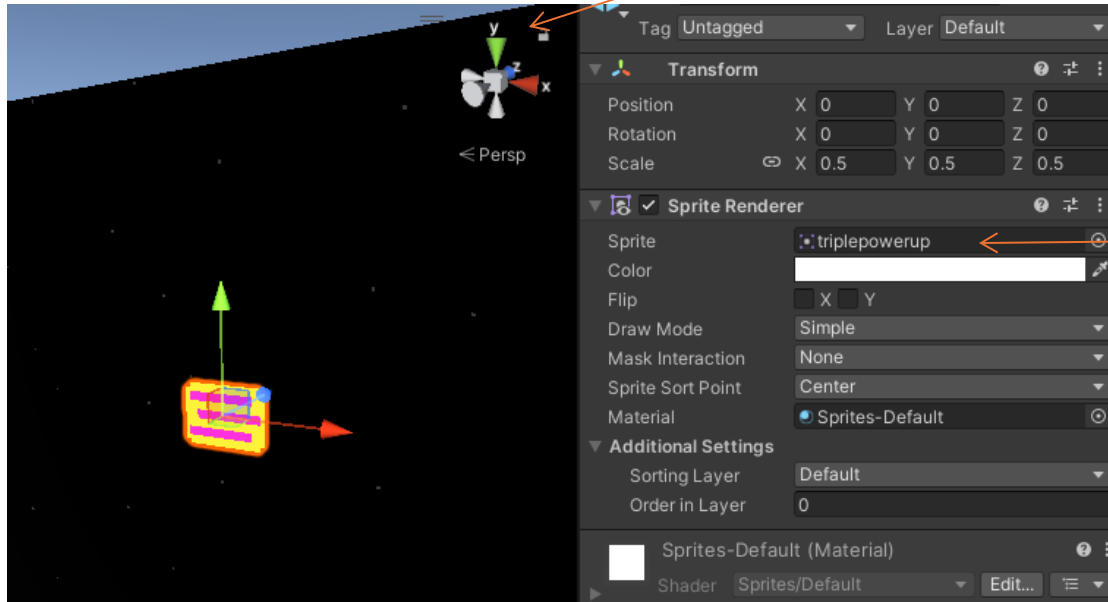
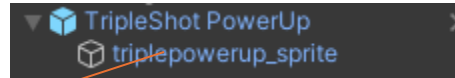
Oluşturulan bu oyun nesnesi Assets ->

Prefabs klasörüne taşınır ve 3'lü lazer prefabi oluşturulmuş olunur.

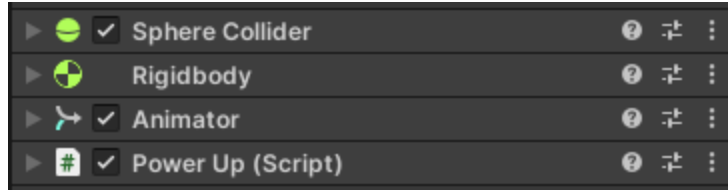


3'lü Atış Güçlendirme Prefabi

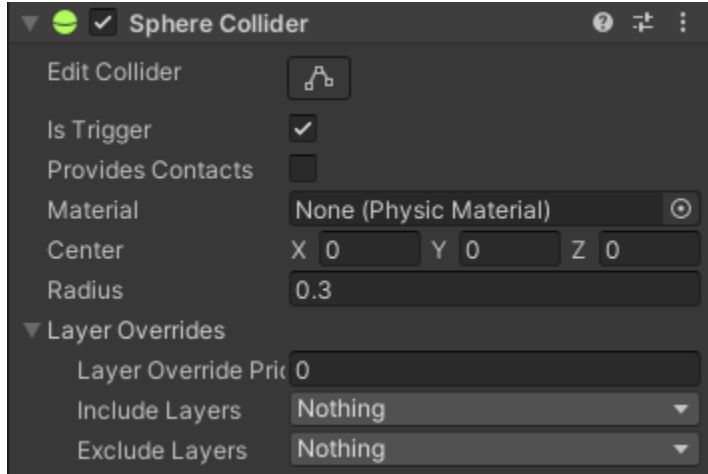
“TripleShot PowerUp” adlı boş bir nesne oluşturulup güçlendirme için kullanılacak sprite çocuk nesnesi olarak bu nesnenin içine atılır.



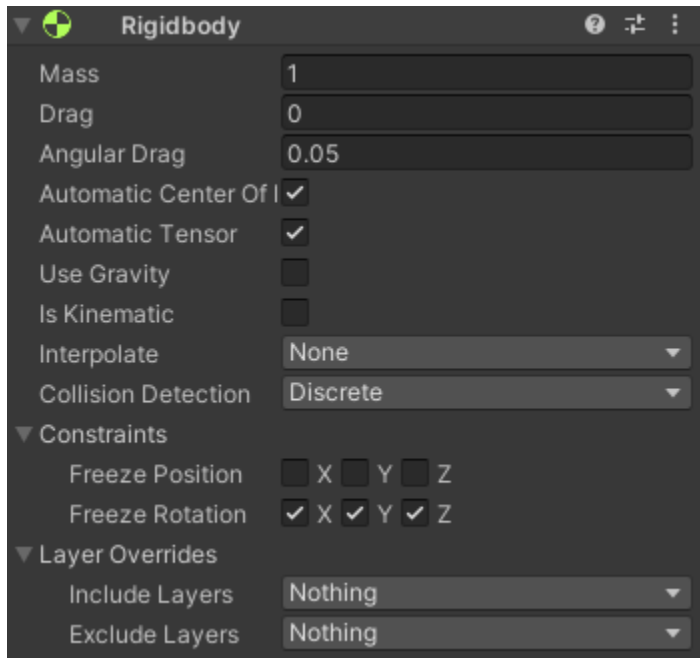
TripleShot PowerUp” oyun nesnesine sphere colider, rigidbody, animator ve güçlendirmenin oyun mantığını içerecek “PowerUp” scripti oluşturulup eklenir.



- **Sphere Collider**



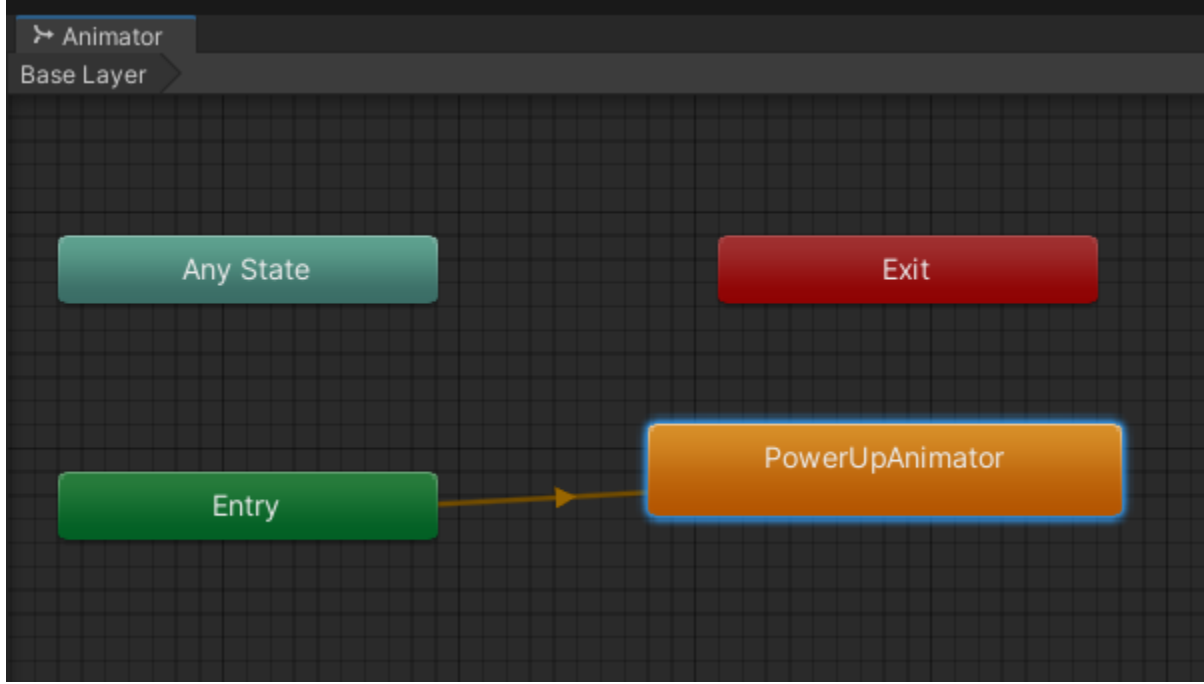
- **Rigidbody**



- **Animator**

Animation Controller

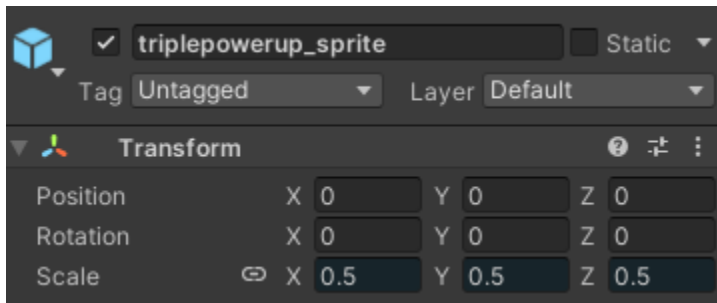
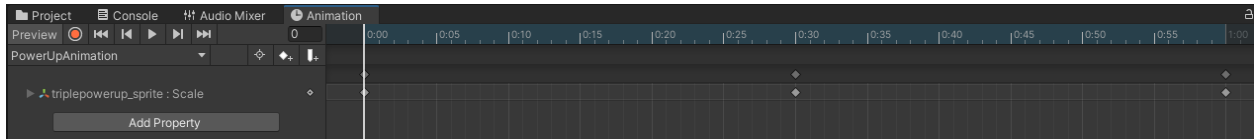
Animator için nesnenin animasyon bilgilerini tutan bir animation controller oluşturulup eklenir.



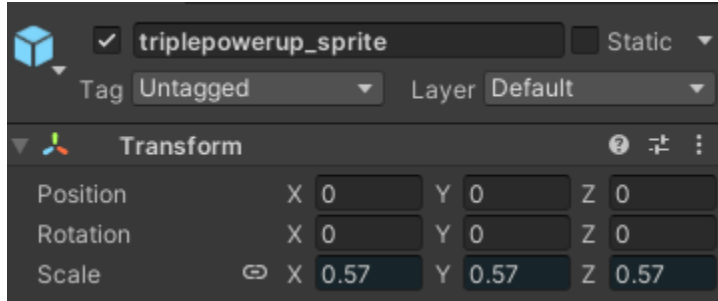
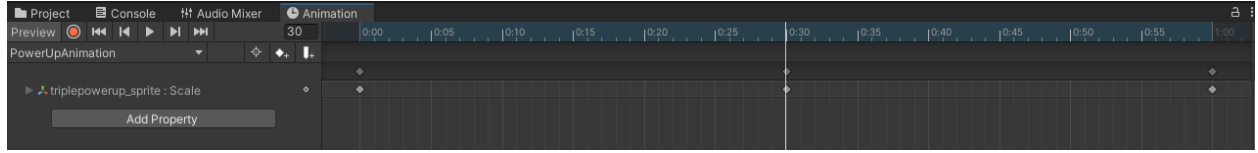
TripleShot PowerUp kontrolcüsünün içi bu şekildedir buradan girişte oluşturduğumuz animasyon sürekli kendini oynatacaktır. Bu animasyon basitçe nesnenin sprite'nın scale'ini büyütüp küçültmektedir.

Animasyon-Scale Büyültüp Küçültme

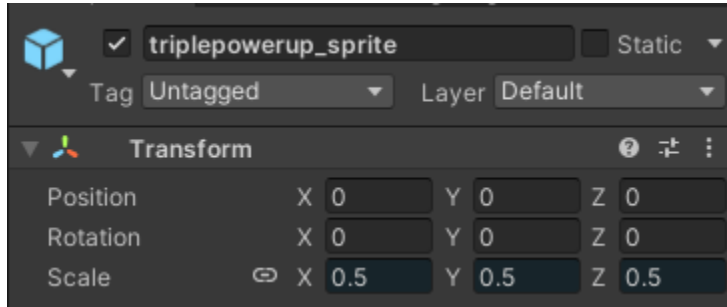
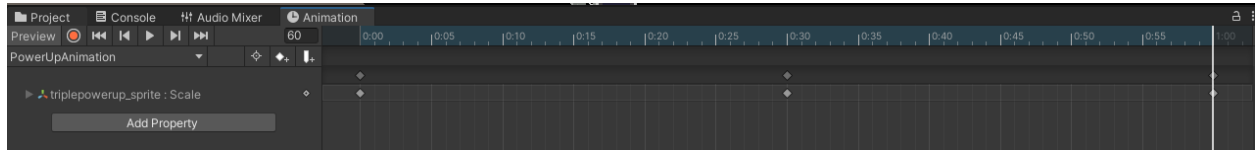
Animasyonun başı:



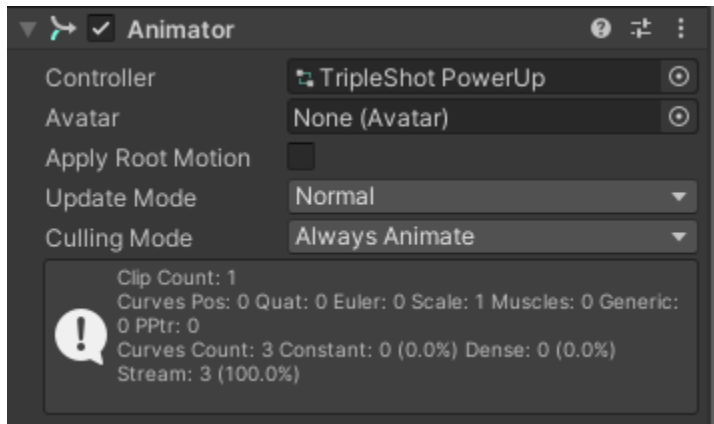
Animasyonun ortası:



Animasyonun Sonu:



Kontrolcüde bu ayarlar yapıp Animator'a eklenir.



- **PowerUp Script**

1. Değişkenler

İhtiyacımız olacak değişkenler oluşturulur. Oyunun geliştirme sürecinde ileride başka güçlendirmelerde ekleneceği için Enum tipinde tip değeri tanımlanmıştır oluşturulmuştur. Her güçlendirme yukarıdan aşağıya hareket edeceği için de hız ve bu hızı rastgeleleştirecek bir çarpanın tanımlaması da yapılmıştır.

```
2 references
public enum PowerUpType
{
    TripleShotPowerup,
    Option2, //diğer powerupları bu şekilde eklemek daha kolay olacak
    Option3,
}
public PowerUpType type;
private PlayerMovement pm;

[Header("Downward Movement")]
[SerializeField] private float speed = 1f;
[SerializeField] private float speedRandomMultiplier = 1f;
```

2. Start() Fonksiyonu

```
Unity Message | 0 references
private void Start()
{
    pm = GameObject.FindGameObjectWithTag("Player").GetComponent<PlayerMovement>();
    speed = speed + Random.Range(0, speedRandomMultiplier);
}
```

1. Oyun hierchy'sinde oyuncu bulunup içerisinden 'PlayerMovment' bileşeni alınır.
2. Hız bileşeni belirlenen çarpanın rastgele bir miktarı eklenerek hız rastgeleleştirilir.

3. Update() Fonksiyonu

```
Unity Message | 0 references
private void Update()
{
    transform.position = transform.position + new Vector3(0, -speed * Time.deltaTime, 0);
    CheckPosition();
}
```

1. Scriptin bağlı olduğu oyun nesnesi aşağı doğru hareket ettirilir

2. CheckPostion() fonksiyonu çağırılır.

4. CheckPosition() Fonksiyonu

```
1 reference
private void CheckPosition()
{
    if (transform.position.y < -(pm.yBorderValue + 1.20f))
    {
        Destroy(gameObject);
    }
}
```

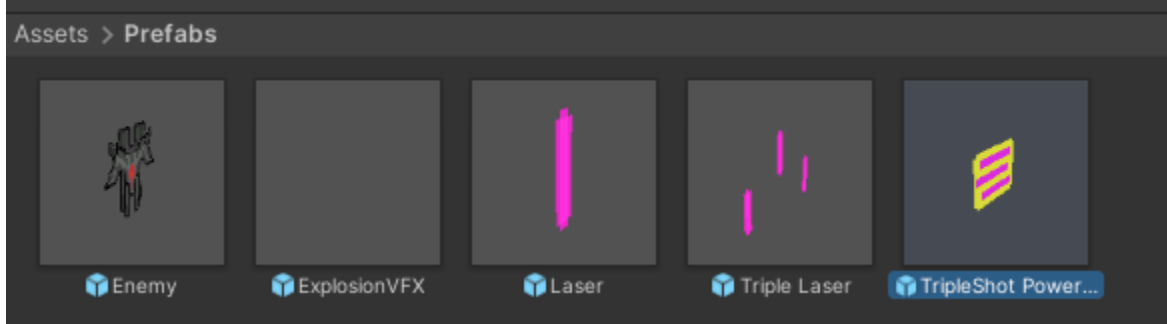
1. Scriptin bağlı olduğu nesnenin konumu pm içerisindeki ekran limit değeriyle kıyaslanır.
2. Eğer altında bir değer olursa güçlendirme nesnesi yok edilir.

5. OnTriggerEnter(Collider other) Fonksiyonu

```
Unity Message | 0 references
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Player"))
    {
        switch (type)
        {
            case PowerUpType.TripleShotPowerup:
                pm.StartTriplePowerup();
                break;
            default:
                Debug.LogWarning("PowerUp type doesnt declared!");
                break;
        }
        Destroy(gameObject);
    }
}
```

1. Güçlendirme nesnesiyle temasa geçen oyuncu nesnesim mi diye kontrol edilir.
2. Eğer oyuncuysa, güçlendirme oyun nesnesinin tipine bağlı kod bloğu çalışır.
3. Eğer 3lü Atış Güçlendirme tipindeyse, pm(PlayerMovement)tan güçlendirme efektini başlatacak StartTriplePowerup() Fonksiyonu çağırılır.

Bu ayarlamaların hepsi yapıldıktan sonra “TripleShot PowerUp” nesnesi Assets -> Prefabs konumuna taşınıp 3’lü atış güçlendirme prefabi oluşturulur.



PlayerMovement Güncellemeleri

- Eklenen Değişkenler

```
[Header("TripleShot Power-Up Settings")]
[SerializeField] private bool tripleShotIsActive=false;
[SerializeField] private GameObject tripleLaserPrefab;
[SerializeField] private float tripleShotDuration = 5f;
Coroutine tripCoroutine;
```

- HandleShoot() Fonksiyonu Mantık Güncellemesi

```
1 reference
private void HandleShoot()
{
    if(Input.GetKey(KeyCode.Space) && fireCooldown <= 0)
    {
        fireCooldown = timer;
        GameObject laser;
        if (tripleShotIsActive)
        {
            laser = Instantiate(tripleLaserPrefab, transform.position + Vector3.up, Quaternion.identity);
        }
        else
        {
            laser = Instantiate(laserPrefab, transform.position + Vector3.up, Quaternion.identity);
        }

        laser.transform.parent=laserContainer.transform;

        if(laser.TryGetComponent<Rigidbody>(out Rigidbody rb))
        {
            rb.AddForce(Vector3.up * projSpeed,ForceMode.Impulse);
            Destroy(laser, projLifespan);
        }
    }
}
```

Ateş edilmek istendiğinde “tripleShotIsActive” değişkenini control ederek eğer aktif ise oluşturulan prefab 3’lü lazer prefab olur, aktif değilse tek lazer oluşturulmaya devam eder.

- Eklenen Fonksiyonlar

1. StartTriplePowerup() Fonksiyonu

```
1 reference
public void StartTriplePowerup()
{
    if (tripCoroutine == null)
    {
        tripleShotIsActive = true;
        tripCoroutine = StartCoroutine(TripleEffect());
    }
    else
    {
        StopCoroutine(tripCoroutine);
        tripleShotIsActive = true;
        tripCoroutine = StartCoroutine(TripleEffect());
    }
}
```

Bu fonksiyon “PowerUp” scriptinden çağırılmaktadır, sayaca bağlı olarak 3lü atış özelliğini aktif kılar.

2. IEnumerator TripleEffect() Fonksiyonu

```
2 references
public IEnumerator TripleEffect()
{
    yield return new WaitForSeconds(tripleShotDuration);
    tripleShotIsActive = false;
}
```

Bu 3lü atış özelliğini belirlenen süre kadar aktif tutar.

SpawnManager Güncellemeleri

- **Eklenen Değişkenler**

İhtiyacımız olacak değişkenler oluşturulmuştur.

```
[Header("TriplePowerup Spawn Settings")]
[SerializeField] private GameObject triplePowerUpPrefab, powerUpContainer;
[Range(0f, 1f)][SerializeField] private float tripSpawnChance = .5f;
GameObject lastSpawnedPowerUp;
```

- **Start() Fonksiyonu Güncellemesi**

```
Unity Message | 0 references
void Start()
{
    pm = GameObject.FindGameObjectWithTag("Player").GetComponent<PlayerMovement>();
    StartCoroutine(EnemySpawner());
    StartCoroutine(TriplePowerUpSpawner());
}
```

3'lü güçlendirmeleri sürekli şansa bağlı olarak canlandıracak coroutine'i başlatacak satır eklenmiştir.

- **Eklenen Fonksiyonlar**

1. **IEnumerator TriplePowerUpSpawner () Fonksiyonu**

```
1 reference
IEnumerator TriplePowerUpSpawner()
{
    while (pm.lives > 0)
    {
        float spawnChanceLocal = Random.Range(0f, 1f);
        if(spawnChanceLocal < tripSpawnChance)
        {
            lastSpawnedPowerUp = Instantiate(triplePowerUpPrefab,
            (new Vector3(Random.Range(-pm.xBorderValue, pm.xBorderValue), pm.yBorderValue + 1.2f, 0)),
            Quaternion.identity);
            lastSpawnedPowerUp.transform.parent = powerUpContainer.transform;
            yield return new WaitForSeconds(5f);
        }

        yield return new WaitForSeconds(1f);
    }
}
```

1. Her döngüde oyuncunun canı 0'dan büyük mü diye kontrol edilir.
2. Eğer canı 0'dan büyükse, rastgele bir sayı üretilir(spawnChanceLocal).
3. Üretilen sayı belirlenen oluşma şansı(tripSpawnChance)'ndan küçük mü diye kontrol edilir.

3. Eğer küçükse, pm(PlayerMovement) dan bel'rl' olan ekran sinirlari içerisinde aralık olarak alan bir rastgele bir X konumu, sabit Y ve Z konumlarında "triplePowerUpPrefab" I yani 3'lü güçlendirme prefabi oluşturulur. Oluşturulan nesne lastSpawnedPowerUp oyun nesnesine atanır.
4. Oluşturulan güçlendirmenin hierchyde parent objesi belirlenir.
5. Belirli bir süre beklenir.
6. Oyuncunun canı olana kadar, bu döngü her saniye tekrardan kontrol edilir.

Kaynakça

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

<https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UIBasicLayout.html>

<https://chatgpt.com/>

Proje Kodu ve Github Repo

Kod: <https://github.com/bathuchan/btu-gameprogramming-BatuhanSengul/tree/main/Reports/6.Hafta>

Proje Repo: <https://github.com/bathuchan/btu-gameprogramming-BatuhanSengul>

Hazırlayan

Batuhan Şengül – 20360859008- bathu.sengul@gmail.com