



OYUN PROGRAMLAMAYA GİRİŞ (TEKNİK SEÇMELİ DERS III)

Dr. Öğr. Üyesi EMRE GÜNGÖR

FİNAL RAPORU + VİZE RAPORU

2022-2023 Güz Dönemi

31/12/2022

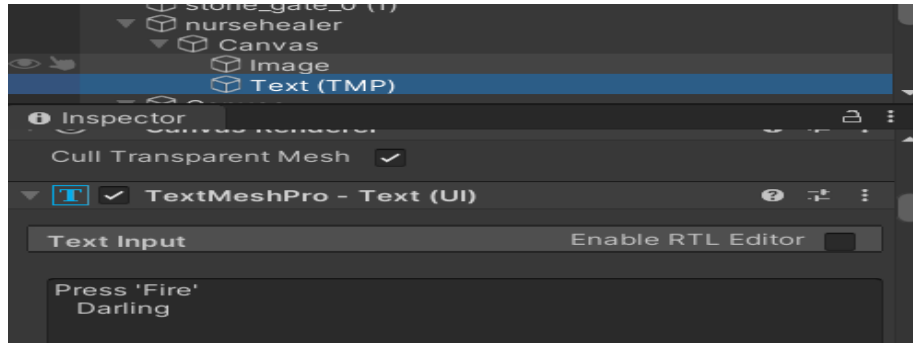
HAZAR HASCAN – 201913172016

VİDEO LİNKİ : <https://youtu.be/HISD8sPxdFs>

Healing Nurse



Can yenileme işlemi için hemşire nesnesi oluşturdum. Bu nesnenin özelliği yanına yaklaştığımızda Collider'ımızda belirlediğimiz alan içerisi Player nesnemiz girdiğinde ona sol tık'a bastığında 2 saniye aralıklarla +10 can veriyor. Nurse'ün içine canvas oluşturup text ile 'fire' a basarak can alacağını gösterdik.



nursehealer objecti ve canvas'ın içindeki nesneler

```

5 public class HealingNurse : MonoBehaviour
6 {
7     private GameObject PlayerObj;
8     private ScPlayerDM PlayerDM;
9     //private ScPlayerData Player1Data;
10
11     private bool isPlayerEntered = false;
12     private bool healState = false;
13
14     //Zamanlayici
15     private float timer;
16     [SerializeField] private float timerMax = 2f;
17     [SerializeField] private float healAmount = 10f;
18
19     // Start is called before the first frame update
20     void Start()
21     {
22         timer = timerMax;
23     }

```

İlk önce HealingNurse script dosyamıza bakarsak PlayerDM ile scPlayerDM'den metodları çağırıyoruz. isPlayerEntered'i bool olarak tanımladık default değerini false atadık. healState'i bool olarak tanımladık default değerini false atadık. Timer kullandık çünkü heal alanına girdiğinde spamlamasını istemeyiz, belli süre içerisinde belli canı alması daha iyi. Timer'ı 2 saniye olarak belirledik. healAmount değerini 10 olarak belirledik her 'fire1' tuşuna bastığımızda 10 can eklenmiş oluyor. Start()da timerla ilgili kodumuzu yazdık.

Asıl bölüme gelirsek Update() metodunda sol tık'a tıkladığında ve player alanın içinde olduğunda player'ı heal'lama işlemini yapıyoruz.

```

if(Input.GetButton("Fire1") && isPlayerEntered == true)
{
    //TextObj.SetActive(false);
    if(healState == false) healState = true;
}
if(healState == true)
{
    HealPlayer();
}

```

HealPlayer fonksiyonunda ise timer'la süreyi kontrol ediyoruz. Player'ın health durumu maksimum sağlığından fazla olamayacak şekilde if koşuluna sokup işlemi yaptırıyoruz. Hurt normalde zarar verme fonksiyonu ama -1 ile çarptığımızda can ekleme işlemini yapmış oluyoruz.

```
timer = timer - Time.deltaTime;
if(timer <=0f)
{
    if(PlayerDM.getPlayerHealth() >= PlayerDM.getPlayerMaxHealth())
    {
        healState = false;
    }
    PlayerDM.hurt(-1 * healAmount);
    timer = timerMax;
    //Debug.Log(PlayerDM.getPlayerHealth());
}
```

onTriggerEnter2D metodunda collider'ımızı other olarak tanımladıktan sonra alana giren kişini tag'ini kontrol ediyoruz. Tag player'a eşit ise if koşulundaki komutlar çalışmaya başlıyor.

```
if(other.gameObject.tag == "Player")
{
    Debug.Log("Girdi");

    PlayerObj = other.gameObject;
    PlayerDM = PlayerObj.GetComponent<ScPlayerDM>();

    TextObj.SetActive(true);

    isPlayerEntered = true;
}
```

onTriggerExit2D metodunda bir önceki metod gibi tag kontrolü yapıp, player ise if koşulu sağlanmış olup içinde işlemler yapılır.

```
if(other.gameObject.tag=="Player")
{
    Debug.Log("Çıktı");

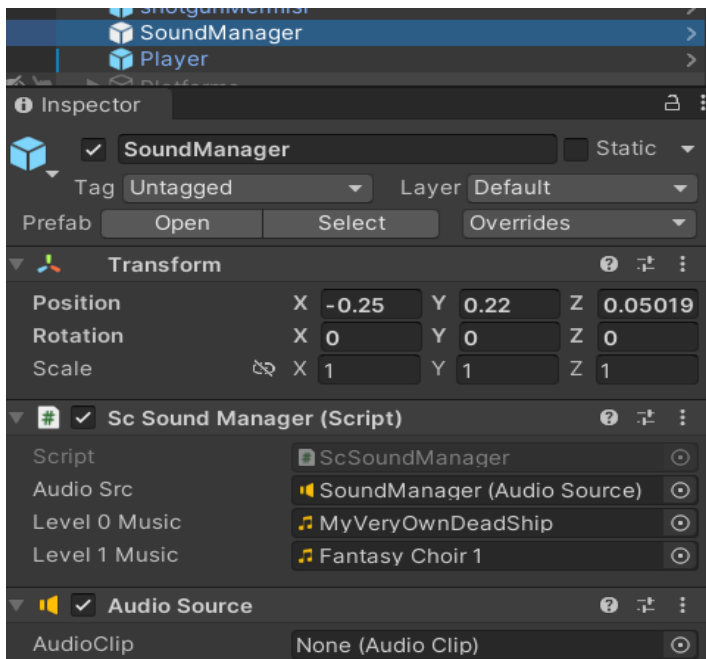
    TextObj.SetActive(false);

    isPlayerEntered = false;
}
```

Sound Manager

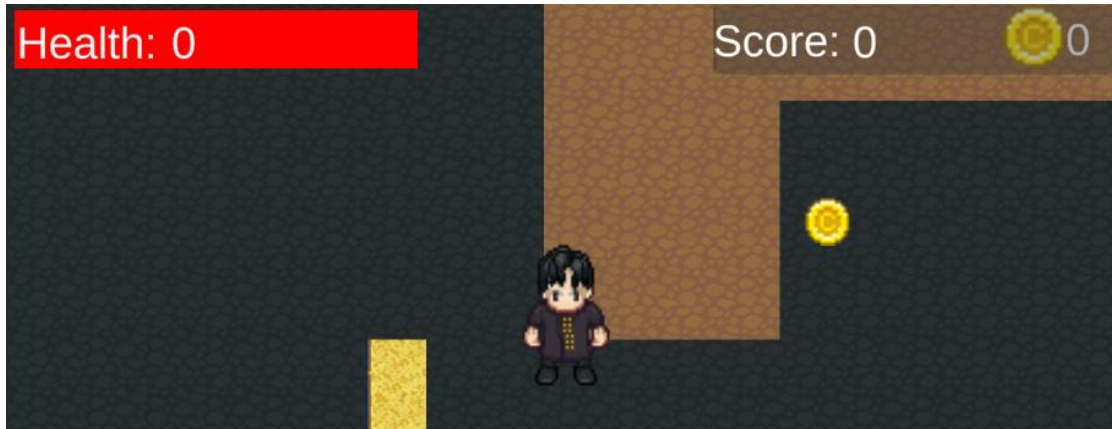
```
4 using UnityEngine.SceneManagement;
5
6 public class ScSoundManager : MonoBehaviour
7 {
8     [SerializeField] public AudioSource AudioSource;
9     [SerializeField] private AudioClip Level0Music;
10    [SerializeField] private AudioClip Level1Music;
11
12    Scene currentScene;
13    // Start is called before the first frame update
14    void Start()
15    {
16        //Debug.Log("Screen Width:"+Screen.width);
17        //Debug.Log("Screen Height:"+Screen.height);
18        currentScene = SceneManager.GetActiveScene();
19        if(currentScene.buildIndex==0)
20        {
21            this.AudioSrc.clip = Level0Music;
22            AudioSource.Play();
23            AudioSource.loop=true;
24        }
25        if(currentScene.buildIndex==1)
26        {
27            this.AudioSrc.clip = Level1Music;
28            AudioSource.Play();
29            AudioSource.loop=true;
30        }
31    }
32 }
```

SoundManager scriptinde sahnelere göre müzik işlemini yaptım. Bu işlem için unityEngine'deki SceneManager kütüphanesini ekliyoruz. AudioSource componentini SoundManager nesnesine ekledik ve serialize field olarak tanımladık. Start metodunda currentScene'yi aktif olan scene'ye veriyoruz. Bu işlemi yaptıktan sonra build settings'deki id'lere göre if koşuluna soktuktan sonra eğer 0'sa Level0Music, 1 ise Level1Music'i çaldırıp Loop'a aldım.



SoundManager object'inin inspector görüntüsünde gördüğümüz gibi script'i çalıştırdık. Prefab olarak yaptım aynı zamanda çünkü ana menüde de direk kullandım. Auido Source componenti de aşağıda gözüküyor.

InGameGUI



GUI'mizi oluştururken ilk başta canvas oluşturuyoruz. GUI'mizin içinde healthBar, HealthTitle, HealthAmount, ImageCoin, CoinCount, Panel, ScoreTitle ve ScoreAmount bulunmaktadır. Burada sağlıkla ilgili nesneler sol tarafa sabitlenmiş, kalan nesneler ise sağ tarafa sabitlenmiş olarak yaptım. Canvas'ımızı Screen Space – Camera olarak yaptım ve pixel perfect işaretli.

```
//GUI Elements
[SerializeField] private GameObject HealthBarObj;
[SerializeField] private GameObject HealthBarTextObj;
[SerializeField] private GameObject BoneCountObj;
[SerializeField] private GameObject ScoreCountTextObj;

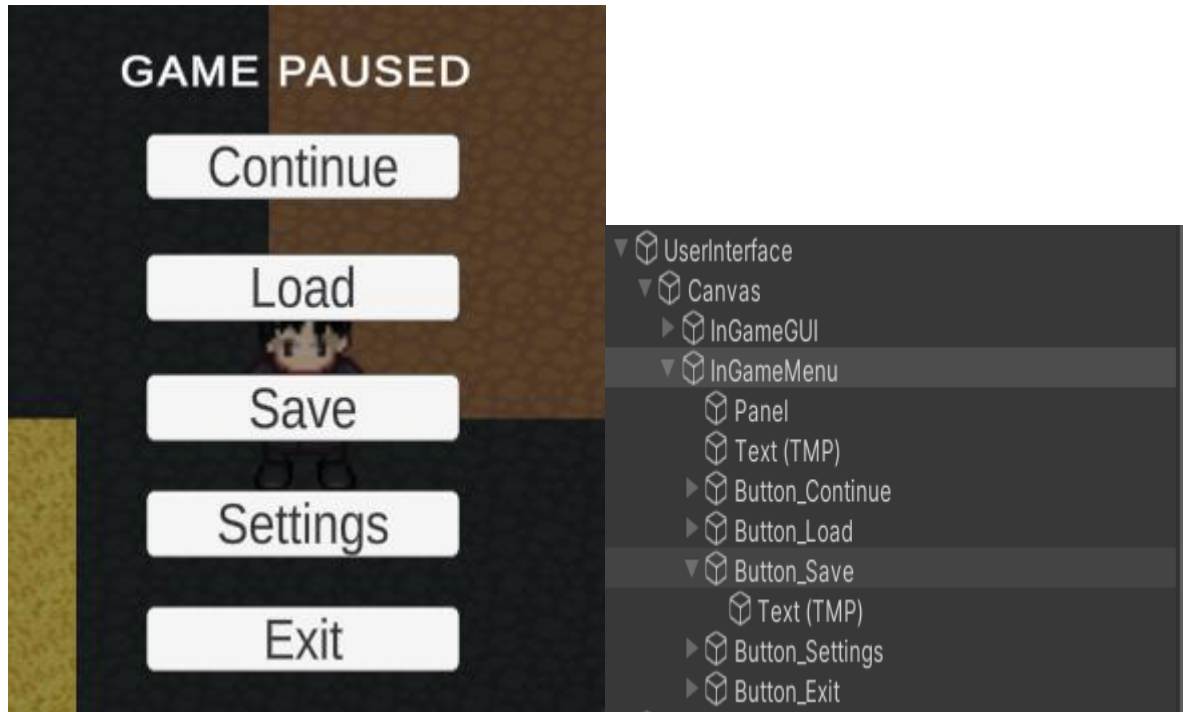
private RectTransform HealthBarTransform;
private TextMeshProUGUI HealthBarText;
private TextMeshProUGUI BoneCountText;
private TextMeshProUGUI ScoreCountText;
```

GameObjectleri tanımladıktan sonra ilk önce start metodun da text ve transform dosyalarını obj dosyalarıyla GetComponent işlemi yapıp bitiriyoruz. Sonrasındaki kodu start() içinde başlatırsak değerler değişmiyor o yüzden Update() içine yazıyoruz. ScPlayerData'dan çektiğimiz verileri Player1Data ile getirip ToString() ile dönüştürüp yazdırdık.

```
HealthBarText.text = Player1Data.getCurrHealth().ToString();
BoneCountText.text = Player1Data.getBoneCount().ToString();
ScoreCountText.text = Player1Data.getScore().ToString();
```

InGameMenu

Menü işleminde GUI'de oluşturduğumuz canvas'ın için InGameMenu nesnesi oluşturup içine butonlar, text dosyası ve panel oluşturuyoruz.



Burada menümüzün arkasındaki siyahlık panel sayesinde geliyor, en üstteki GAME PAUSED yazısı text diğer kalan nesneler butonlar. Kod olarak oluşturduğumuz script'in update metodu içinde Cancel yani "esc" tuşunu ve P tuşuna basıldığında metodu çalıştırmasını istiyoruz.

```
if(Input.GetButtonDown("Cancel") || Input.GetKeyDown(KeyCode.P))  
{  
    button_ContinuePauseToggle();  
}
```

```
public void button_ContinuePauseToggle()  
{  
    if(isPause==false)  
    {  
        Time.timeScale=0f;  
        InGameMenuObj.SetActive(true);  
        isPause=true;  
    }  
    else  
    {  
        Time.timeScale=1f;  
        InGameMenuObj.SetActive(false);  
        isPause=false;  
    }  
}  
public void button_Exit(){  
    Application.Quit();  
}
```

Burada timescale ile zamanı durdurup sonrasında tekrar devam ettiriyoruz. SetActive ile de menüyü true false gösterip göstermeme durumu.

TELEPORTS

Teleport sistemi olarak farklı bir sistem kullandım. Önce kapıları oluşturduktan sonra onlara boxcollider2D componenti ekleyip trigger'ı işaretledim. Sonrasında 2 scriptim var, birincisi teleportlarda çalışan, 2.si ise player'a eklediğim.

```
[SerializeField] private Transform destination;  
  
public Transform GetDestination()  
{  
    return destination;  
}
```

Bu script teleporter nesnelerine eklediğim script, burada destination olarak ilkinde 2. Teleportu, ikincisine ilk teleportu veriyoruz.

```
private GameObject currentTeleporter;  
void Update()  
{  
    if (Input.GetKeyDown(KeyCode.E))  
    {  
        if (currentTeleporter != null)  
        {  
            transform.position = currentTeleporter.GetComponent<Teleporter>().GetDestination().position;  
        }  
    }  
}  
private void OnTriggerEnter2D(Collider2D collision)  
{  
    if (collision.CompareTag("Teleports"))  
    {  
        currentTeleporter = collision.gameObject;  
    }  
}  
private void OnTriggerExit2D(Collider2D collision)  
{  
    if (collision.CompareTag("Teleports"))  
    {  
        if (collision.gameObject == currentTeleporter)  
        {  
            currentTeleporter = null;  
        }  
    }  
}
```

Player teleport scriptinde ise Update metodunda Input ile 'E' tuşunu bekliyoruz. Değer null değil ise teleport işlemi yapılmış oluyor. BoxCollider2d eklediğimizden dolayı o bölgeye girdiğini anlaması için OnTriggerEnter2D ve OnTriggerExit2D metodları yazıyoruz. Bu scripti player'a verdikten sonra sistem sorunsuz bir şekilde çalışıyor.

Enemy'deki yeni özellikler

Enemy'miz oyunun player kadar ana karakterlerinden biri sayılır, burada ona da healthBar eklemenin iyi olacağını düşündüm ve gayet iyi oldu.

```
[SerializeField] private float EnemyHurtDamage=40f;  
  
[SerializeField] public float max_Health=100f;  
[SerializeField] private float min_Health=0f;  
[SerializeField] public float curr_Health=60f;  
  
public GameObject healthBar;  
public Slider healthBarSlider;
```

Burada düşmanımızın can değerlerini tanımladık fakat önemli olan düşmanlarımızın çeşitine göre bu değişiyor. [SerializeField] özelliği ile kolayca değiştirebiliyoruz. GameObject olarak healthBar ve can barı için Slider oluşturdum o da healthBarSlider olarak tanımlandı.

```
if(col.CompareTag("Bullet"))  
{  
    healthBar.SetActive(true);//sa  
    healthBarSlider.value = CalculateHealthPercentage();//sa  
    float damage= col.gameObject.GetComponent<ScShotGunBullet>().getBulletDamage();  
    this.setHealth(this.getCurrHealth()-damage);  
    Debug.Log("Enemy Current Health:"+ this.getCurrHealth());  
}
```

Buradaki kodda düşmanımızın OnTriggerEnter2D metodunda eğer tag'i mermi olan karşılaşmasını görüyoruz. Burada yanında //sa bulunanlar can barıyla ilgili olan kısım mermiyle etkileşime girdiği anda düşmanımızın can barı aktif oluyor. Slider'da ise .value ile değeri alıyoruz. Sliderlar'da value değeri 0-1 arasında olduğu için internette çeşitli araştırmalar sonucu yüzde hesaplayıp o değeri döndürürsek daha iyi sonuç aldığımızı gördüm. Buradaki işlem sonucu aldığımız değer bar'ın oyunda düzgün bir şekilde gösterilmesini sağlıyor. 2. Satırdaki metod setHealth() fonksiyonunda da yazıldı ve çalışıyor.

```
private float CalculateHealthPercentage()//sa  
{  
    return (curr_Health / max_Health);  
}
```

```
public void setHealth(float HealthAmount){  
    if(HealthAmount > this.max_Health){  
        this.curr_Health=this.max_Health;  
        healthBarSlider.value = CalculateHealthPercentage();//sa  
    }  
    else{  
        this.curr_Health = HealthAmount;  
        healthBarSlider.value = CalculateHealthPercentage();//sa  
    }  
}
```

ScPlayerData

Bu scriptimiz player ile yaptığımız çoğu işlem için kilit nokta görevi görüyor diyebiliriz. Player'ımızla ilgili can bilgileri max, min ve anlık olmak üzere sonrasında score bilgisi tutuluyor. Buradaki metodları yazmamızın sebebi kullandığımız farklı scriptlerde bu metodları çağırarak farklı fonksiyonları çalıştırmak diye açıklanabilir. getScore metodu ile o anki score değeri döndürülüyor. IncreaseScore metodu score'u 1 arttırıyor. getMinHealth metodu oyuncunun minimum canını kullanmamızın lazım olduğu yerde bize değeri veriyor, mesela oyuncunun canı minimum'un altına düştüğünde ölmesindeki yer. getMaxHealth metodu oyuncunun maksimum canını kullanmamız lazım olan gerektiği yerde çağırıyoruz, mesela heal alırken anlık can değeri maksimum health'i geçmemesini istediğimizde. getCurrHealth ise anlık can değerini veriyor çoğu yerde kullandık.

```
public class ScPlayerData
{
    private float max_Health=100f;
    private float min_Health=0f;
    private float curr_Health=90f;
    private int score=0;
    private int bone_count=0;
    private float playerPosX,playerPosY;

    public int getScore(){return this.score;}
    public void increaseScore(){this.score++;}
    public int getBoneCount(){return this.bone_count;}
    public void increaseBoneCount(){this.bone_count++;}
    public float getMinHealth(){return this.min_Health;}
    public float getMaxHealth(){return this.max_Health;}
    public float getCurrHealth(){return this.curr_Health;}
    public void setHealth(float HealthAmount){
        if(HealthAmount > this.max_Health)
            this.curr_Health=this.max_Health;
        else
            this.curr_Health = HealthAmount;
    }
}
```

ScPlayerDM

PlayerDM scripti kalabalık olan scriptlerden bir tanesi denebilir. Bu scriptte ilk olarak ScPlayerData'dan veri alabilmek için Player1Data oluşturduk ve bunu start içinde new olarak yazdıktan sonra lazım olan yerlerde kullandık. GUI'mizde oluşturduğumuz GameObjectler ScPlayerDM içinde, getPlayerHealth(), getPlayerMaxHealth() ve playerScore ile ilgili metodlar yazdık. Update metodunda eğer playerımızın anlık can değeri min can değerinin altına düşer ise player'ımızın SetActive özelliğini false yapıp zamanı durduruyoruz.

```
if(Player1Data.getCurrHealth()<Player1Data.getMinHealth())
{
    this.gameObject.SetActive(false);
    Time.timeScale=0f;
    //Destroy(this.gameObject);
}
```

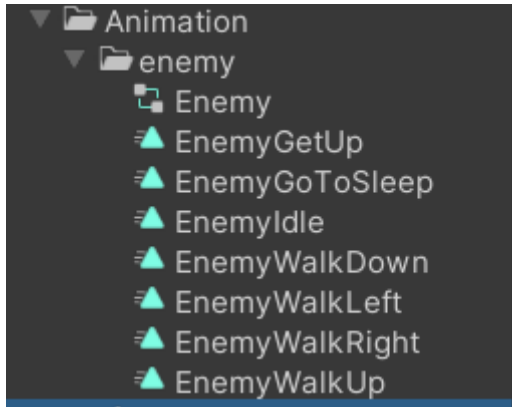
PlayerDM'de OnCollisionEnter2D metodunda karşılatığı nesnenin tag'i Enemy ise ScEnemyController'ı kullanmak için çağırdığımız enemyContr ile player'ımıza zarar verdiriyoruz.

```
void OnCollisionEnter2D(Collision2D col)
{
    if(col.gameObject.tag == "Enemy")
    {
        Debug.Log("Player Collision with Enemy");
        //Destroy(col.gameObject);
        //Destroy(this.gameObject);
        //col.gameObject.SendMessage("ApplyDamge",10);
        ScEnemyController enemyContr;
        enemyContr = col.gameObject.GetComponent<ScEnemyController>();
        //Player1Data.setHealth(Player1Data.getCurrHealth()-enemyContr.getEnemyHurtDamage());
        this.hurt(enemyContr.getEnemyHurtDamage());
        //Debug.Log("Player Current Health:"+ Player1Data.getCurrHealth());
    }
}
```

Hurt metodu derste kiyle aynı şekilde yazıldı. Burada float olarak amount değeri tanımlıyoruz. Player1Data'mızla metodları çağırıp en sonunda -amount deyip hasar kadar can düşüyoruz ve bunu debug.log'la ekrana yazdırdım.

```
public void hurt(float amount)
{
    Player1Data.setHealth(Player1Data.getCurrHealth()-amount);
    Debug.Log("Player Current Health:"+ Player1Data.getCurrHealth());
}
```

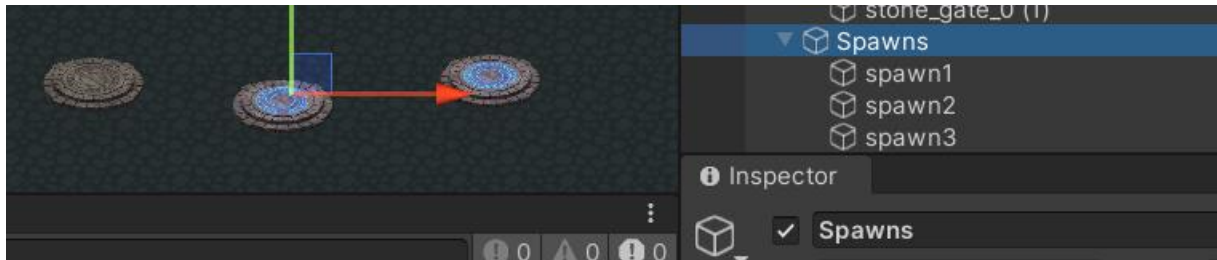
ENEMY ANIMATIONS



Düşmanımda yürüme animasyonu tanımladım. 4 farklı yöne doğru animasyon var çapraz yönlerde en yakın animasyonu sağlıyor. GetUp ve GoToSleep ise player'ımızın yanına gelip veya yanında olmaması ile alakalı animasyonlardır

ENEMY SPAWNPOINTS

Oyunun devamlılığı için düşmanımızın tekrardan doğması gerekir. Burada çeşitli sorunlarla karşılaştım ilk olarak düşmanların hepsini öldürdüğümde unity hata vererek yeni düşman yaratmıyordu onun için haritanın uzak bir yerine duvarların içinde öldürülemeyecek 3 düşman türünden koyup herhangi bir sorun olmadan tekrar canlanmalarını sağladım. İlk olarak 20 saniye bekliyoruz silah alma heal'lanma işlemleri için ondan sonrasında her 10 saniyede bir yeni düşman doğmuş oluyor.



3 farklı spawn var bu 3 farklı düşman türünden dolayı bu şekilde tasarladım. Her spawn'dan farklı düşman türü doğuyor. Spawnslar sorunsuz bir şekilde düzgün çalışıyorlar ve oyuncu için score kasma sürekli hale gelmiş oldu.

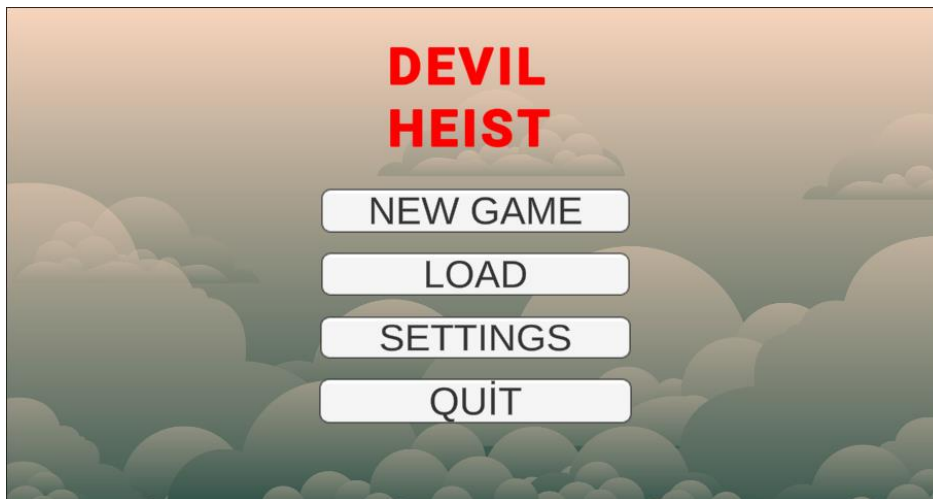
```

public class SpawnManager : MonoBehaviour
{
    public GameObject spawnPoint1;
    /*public GameObject spawnPoint2;
    public GameObject spawnPoint3;*/
    // Start is called before the first frame update
    void Start()
    {
        InvokeRepeating(nameof(Spawn), 20f, 5f);
    }
    void Spawn()
    {
        Instantiate(spawnPoint1, transform.position, Quaternion.identity);
        /*Instantiate(spawnPoint2, transform.position, Quaternion.identity);
        Instantiate(spawnPoint3, transform.position, Quaternion.identity);*/
    }
}

```

Spawnla ilgili SpawnManager scriptim var. Bu script'de Instantiate metodunu derste de gördüydük onu kullandım, burada SpawnPoint gameobject olarak tanımladıktan sonra gerisi start içindeki koda kalıyor. İnternette araştırıp öğrendim, sondaki değerler düşmanın ne kadar sık doğma süresiyle alakalı.

ANA MENÜ



Ana menünün oyun için görünümü bu şekilde tasarladım. Canvas içinde butonlar, oyun ismi için text ve arka plan için Image ekledik. Ana menümüz level(0)'da bulunuyor. MenuManager Objectimiz var, SoundManager prefab'dan buraya da ekledim, menü de müzik çalıyor. Scriptler ile scene arası geçiş var.

EnemyHealthBar oyun için görüntüsü :



1000 canlı boss diyebiliriz buna onun için cisim büyük ve can barı daha kocaman.

Enemy'nin tasarımı :



Genel olarak basic enemy diyebiliriz. Hızlı fakat görüş açısı düşük, hasarı düşük, canı diğer türlere göre az öldürmesi kolay bir türdür.

Speed	5
Check Radius	5
Attack Radius	5
Enemy Hurt Damage	25
Max_Health	100
Min_Health	0
Curr_Health	100
Health Bar	HealthBar
Health Bar Slider	HealthBar (Slider)

Enemy(2)'nin tasarımı :



Bu biraz daha özelleştirilmiş düşman türü. Basic olandan daha güçlü fakat en güçlüsü değil diyebiliriz. Ortalama değerlere sahip bazı özellikleri sayesinde küçümsenmemesi gerekir derim.

Speed	3
Check Radius	8

Enemy Hurt Damage	50
Max_Health	200
Min_Health	0
Curr_Health	200

Enemy(1)'in tasarımı :



Diğerlerinden kat kat daha güçlü ve eğer size dokunursa direk ölüyorsunuz. Hasar diğeri player'ımızın maxHealth değeriyle aynı. Dezavantajı ise yavaş olması diğer düşmanlara göre çok yavaş sayılır. Onun için uzaktan durmanızda yarar var diyebiliriz.

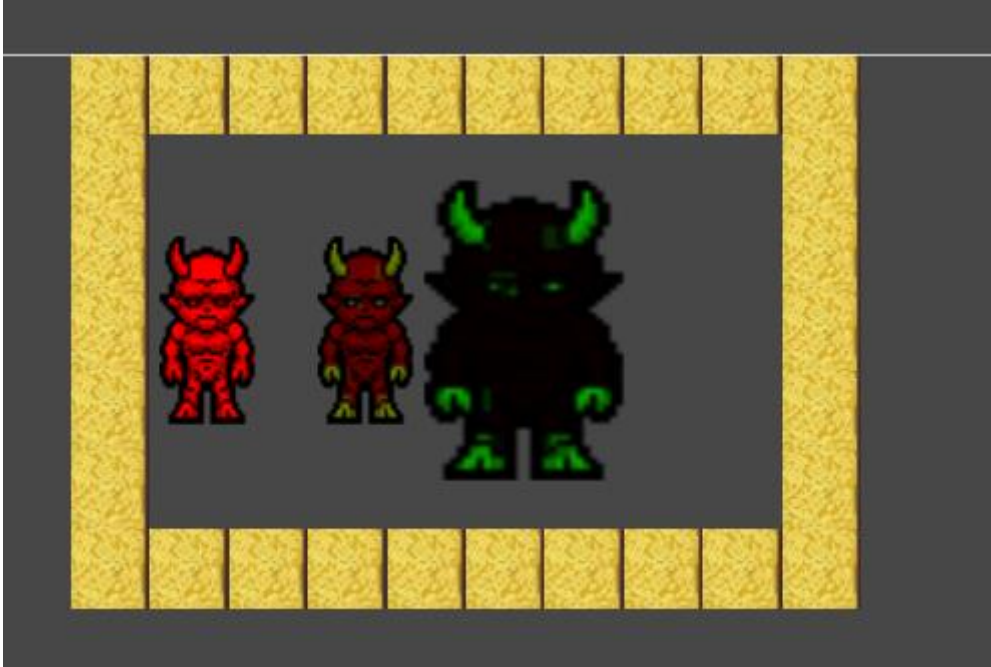
Speed	2
Check Radius	10

Enemy Hurt Damage	100
Max_Health	1000
Min_Health	0
Curr_Health	1000

Teleport Noktaları oyun için görünüşü:



**SPAWN POINTLERIN HATA VERMEDEN DEVAM ETMESİ İÇİN ÖLDÜRÜLEMİYEN
3 DÜŞMANI KOYDUĞUM YER :**



Vizede yolladığım rapor aşağıdadır...

1- OYUN HAKKINDA KISA ÖZET

Tasarladığım oyunda haritanın belli bir bölümünde düşmanla karşılaşacağımız bölge olacak ve düşmanları öldürüp score (puan) toplayacağız. Topladığımız puanlarla çeşitli silahlar alıp tekrar o bölgede düşmanlarla savaşacağız.

2-A OYUN HANGİ KİTLEYE HITAP EDİYOR ?

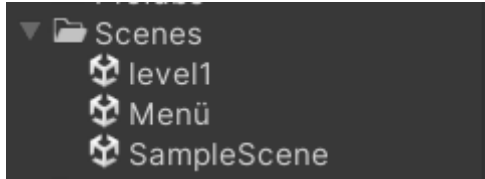
Oyunumun faaliyet alanı oyuna meraklı genç yaşlı fark etmeksizin tüm yaş kitlesine hitap eden bir oyun. Oyun hakkında ilgisi olan herkesin alışkan olduğu düşman öldürüp kazanım elde etme tarzına düşkün insanlar daha da çok sevecektir.

2-B KİTLEYE NASIL ULAŞILACAK ?

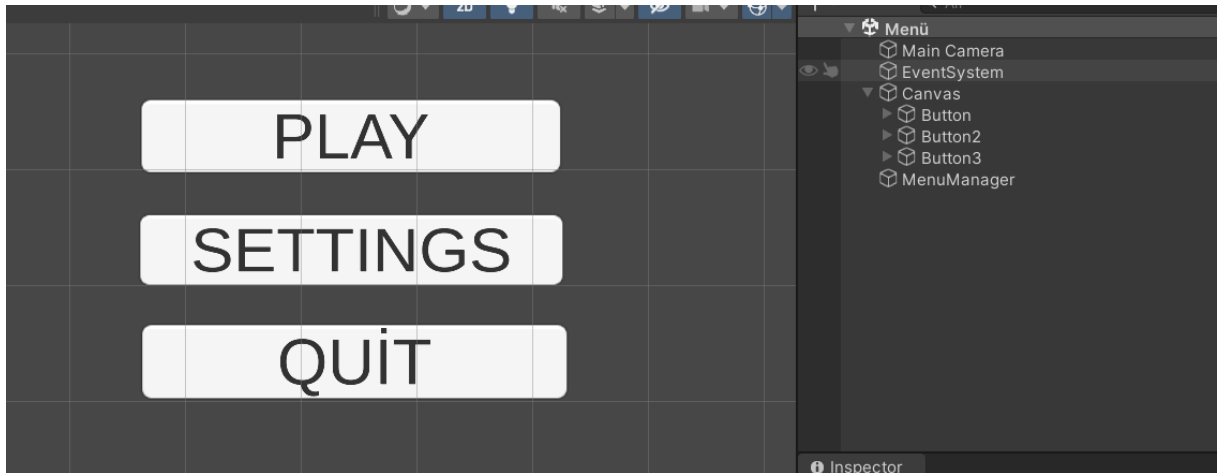
Oyuncu kitlesine çeşitli oyun marketlerinden (steam, epic games, ...) gibi yerlerden onay aldıktan sonra ulaşabilirim ya da internet sitesi hazırlayıp oradan indirme şeklinde de olabilir. Reklam olarak sosyal medyadaki oyun sayfaları ya da çeşitli oyun web siteleri (merlin'in kazanı vb.) oyun reklamları olabilir.

3- OYUN HAKKINDA KOD VE TASARIM AÇIKLAMALARI

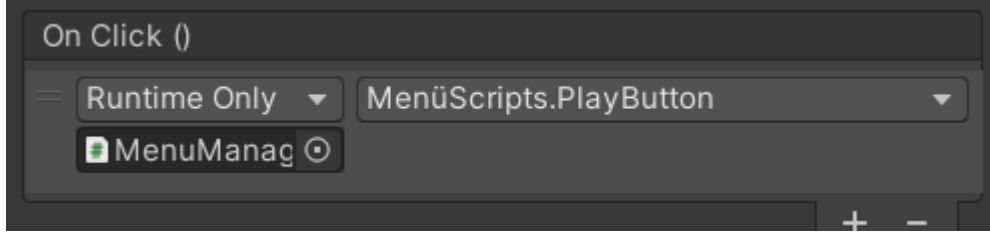
Şu anlık 2 temel sahnemiz var. Menü ve level1.



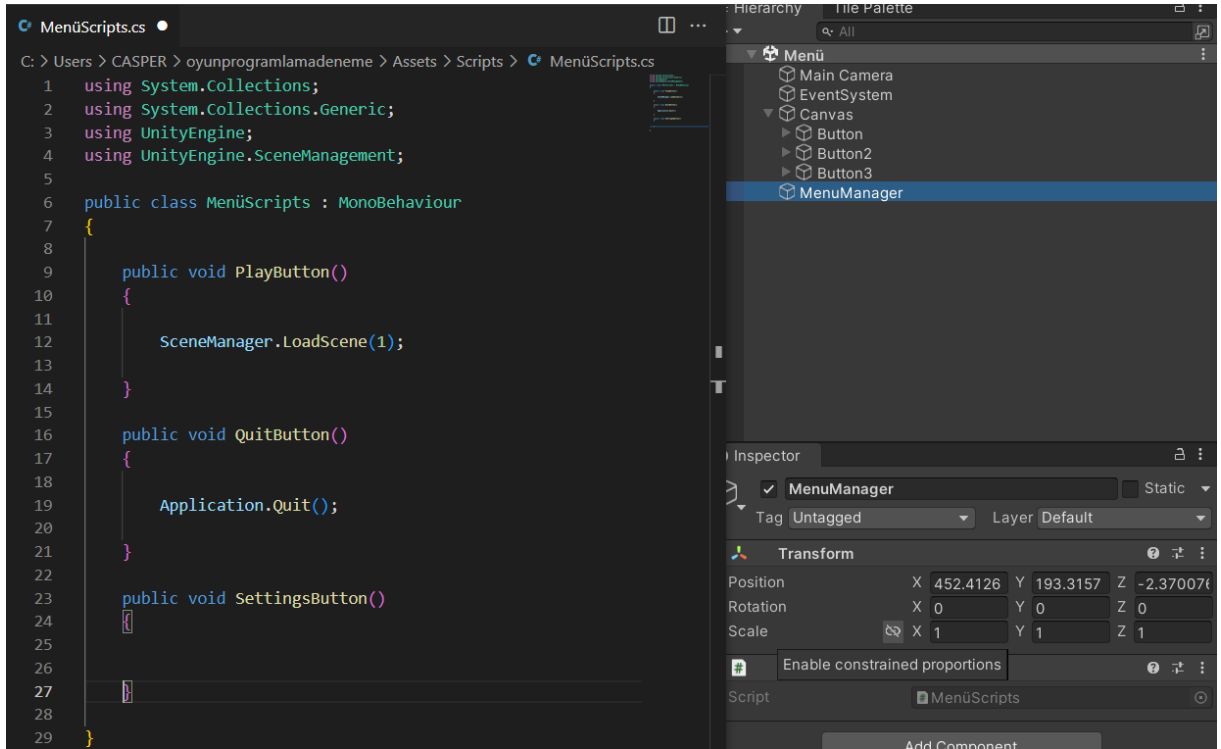
Menü sahnesinde 3 butonumuz var fakat şu anlık 2 tanesi aktif olarak çalışıyor. Play ve Quit butonları.



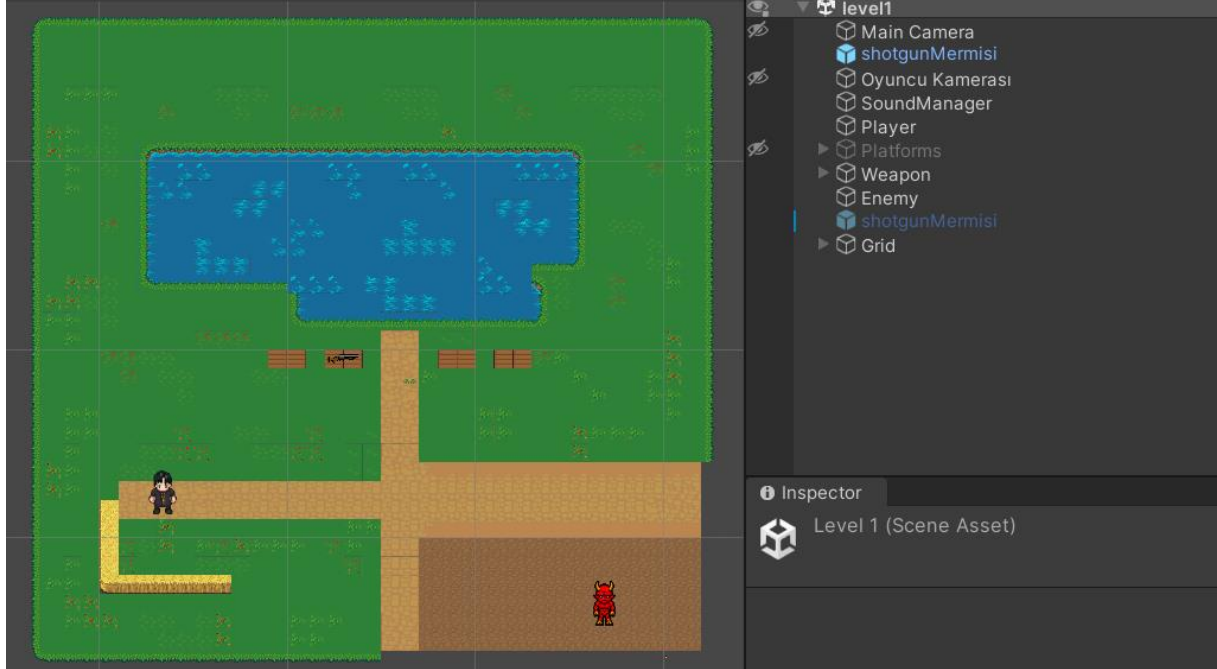
MenuManager diye bir object oluřturduk ve ona oluřturduėum MenüScripts scriptimizi attık. Sahne olarak build settingsden 2 sahnemizi numaralandırıp LoadScene metoduyla onclick durumunda o sahneyi yüklettik.



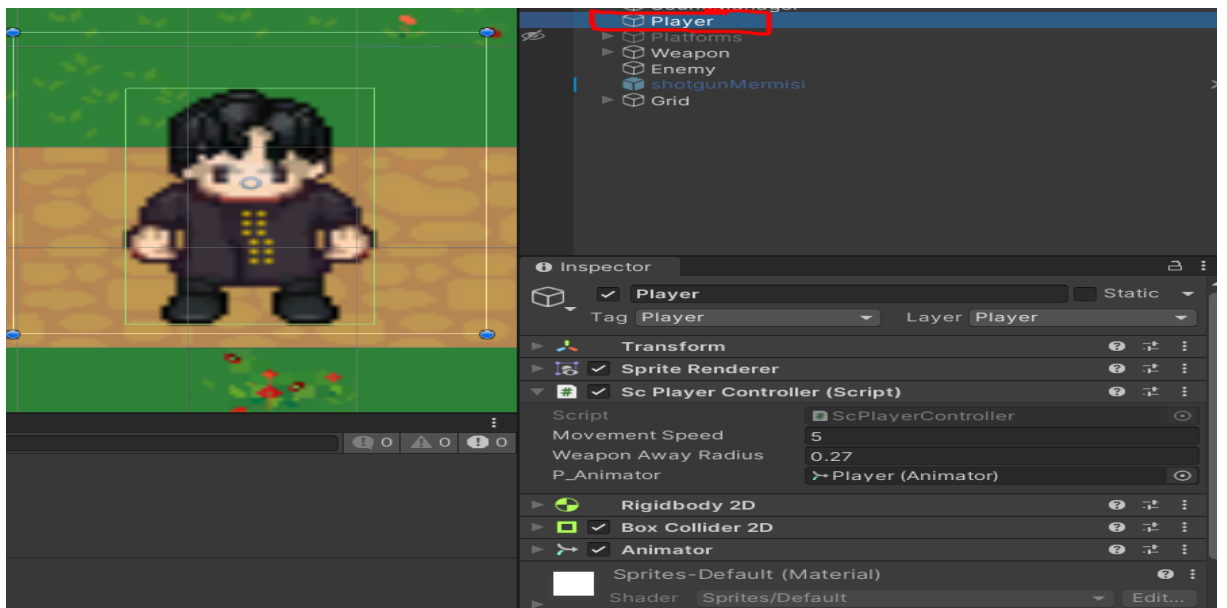
(play butonunun onclick durumu)



Ana sahnemiz level1 sahnesi oyunumuzu oynayabildiğimiz sahnedir. Tanımlı bir çok object, script kod, tile'lar, oyuncumuzu takip eden kamera(cinemachine virtual camera) gibi bir çok şey bulunmaktadır.



Kısa kısa oluşturduğumuz objectleri tanımlayacağım. Player'dan başlayalım. Player bizim oyundaki ana karakterimiz. Oyunu oynayan kişi her şeyi onun üzerinden yapacak ve onu kontrol edecek.



Player'ımız bir çok component'e sahip. (rigidbody 2d, animator ...)

```

if(weaponObj != null && isWeaponEquipped==true)
{
    Vector3 mousePos = Camera.main.ScreenToWorldPoint(Input.mousePosition);
    Vector2 direction = mousePos - this.transform.position; //WeaponObj.transform.position;
    float angle = Vector2.SignedAngle(Vector2.right, direction);
    float angleRad = angle * Mathf.PI/180;
    weaponAngle = angle;

    //Debug.Log("Angle : "+angle);
    WeaponObj.transform.eulerAngles = new Vector3(0,0,angle);
    WeaponObj.transform.localPosition = new Vector2(weaponAwayRadius*Mathf.Cos(angleRad),weaponAwayRadius*Mathf.Sin(angleRad));
}

if(isWeaponEquipped==true && Input.GetButtonDown("Fire1"))
{
    //Debug.Log("Fire"); ateş edip etmediğini kontrol ettim
    if (WeaponObj != null)
    {
        WeaponObj.GetComponent<ScShotGun>().shoot(weaponAngle);
    }else {Debug.LogError("Errorr: WeaponObjs is "+WeaponObj);}
}

void OnTriggerEnter2D(Collider2D col)
{
    //Debug.Log(col.CompareTag("Weapon"));
    if(isWeaponEquipped==false && col.CompareTag("Weapon"))
    {
        //Debug.Log("Is triggered");
        //Debug.Log(col.gameObject.name + " : " + gameObject.name + " : " + Time.time);
        WeaponObj = col.gameObject;
        col.gameObject.transform.position = new Vector2(this.gameObject.transform.position.x+weaponAwayRadius, this.gameObject.transform.position.y);
        col.gameObject.transform.parent = this.gameObject.transform;
        isWeaponEquipped=true;
    }
}

```

Player'ımızın ScPlayerController scriptinde çeşitli kodlar var. Silahı eline alma, ateş etme gibi temel özellikleri çalıştırıyorlar. Bunları çeşitli matematik ifadeleri ile elinde döndürebiliyoruz. Açılarını kullanıp elinde tuttuğu silahın yönünü değiştirebiliyoruz.

```

// Update is called once per frame
void Update()
{
    CheckInput();
    Animate();
}

void FixedUpdate()
{
    Move();
    //playerRB.velocity = new Vector2(h_Input*movementSpeed,v_Input*movementSpeed);
}

void Move()
{
    playerRB.velocity = new Vector2(h_Input*movementSpeed,v_Input*movementSpeed);
}

void Animate()
{
    p_Animator.SetFloat("Mov_X", h_Input);
    p_Animator.SetFloat("Mov_Y", v_Input);
}

```

Player'ı hareket ettirmemize yarayan kodlar ve animasyonda da bize yardımcı olacak Animate metodu.

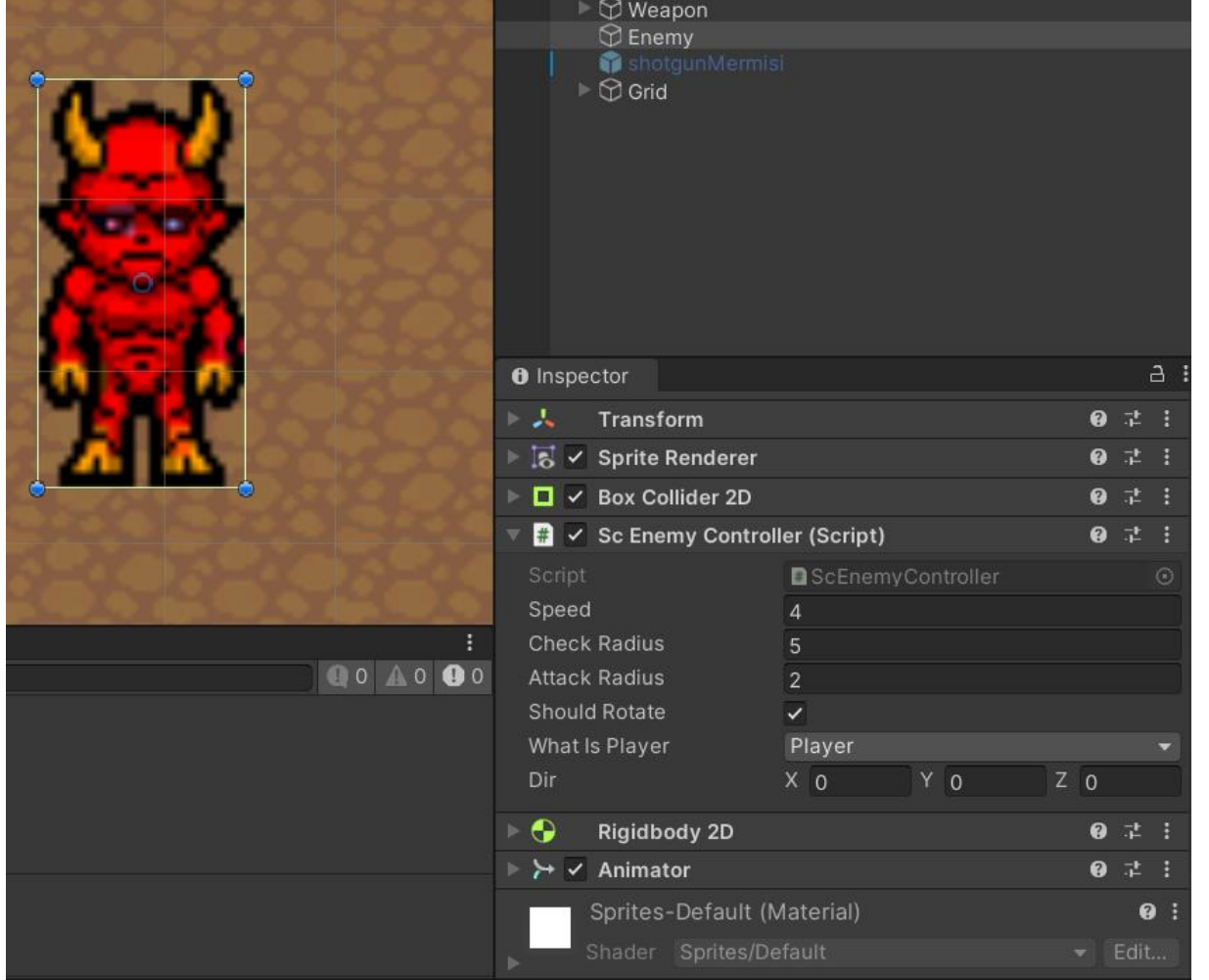
```

void CheckInput()
{
    h_Input = Input.GetAxis("Horizontal");
    v_Input = Input.GetAxis("Vertical");
}

```

Inputu burdan alıp değerlendiriyoruz.

Enemy object'ı gene Player gibi bir çok component'e sahip. Enemy'ı tasarlarken internetten araştırıp çeşitli kaynaklara baktıktan sonra player'ı belli bir mesafede görünce takip edecek şekilde tasarladım.



Düşmanın nesneyi takip etmesi ScEnemyController scriptiyle oluyor. Animator ile düşmana yürüme animasyonu ve bekleme animasyonu ekledim.

```

public class ScEnemyController : MonoBehaviour
{
    public float speed;
    public float checkRadius;
    public float attackRadius;

    public bool shouldRotate;

    public LayerMask whatIsPlayer;

    private Transform target;
    private Rigidbody2D rbEnemy;
    private Animator eAnim;
    private Vector2 movement;
    public Vector3 dir;

    private bool isInChaseRange;
    private bool isInAttackRange;

```

Enemy scriptindeki tanımlı değişkenlerim.

```

void OnTriggerEnter2D(Collider2D col)
{
    //Debug.Log("enemy - " + col.gameObject.name + " : " + gameObject.name + " : " + Time.time);

    if(col.CompareTag("Bullet"))
    {
        Destroy(this.gameObject);
    }
}

```

Düşmanımıza mermi değdiğinde onu yok eden kodumuz.

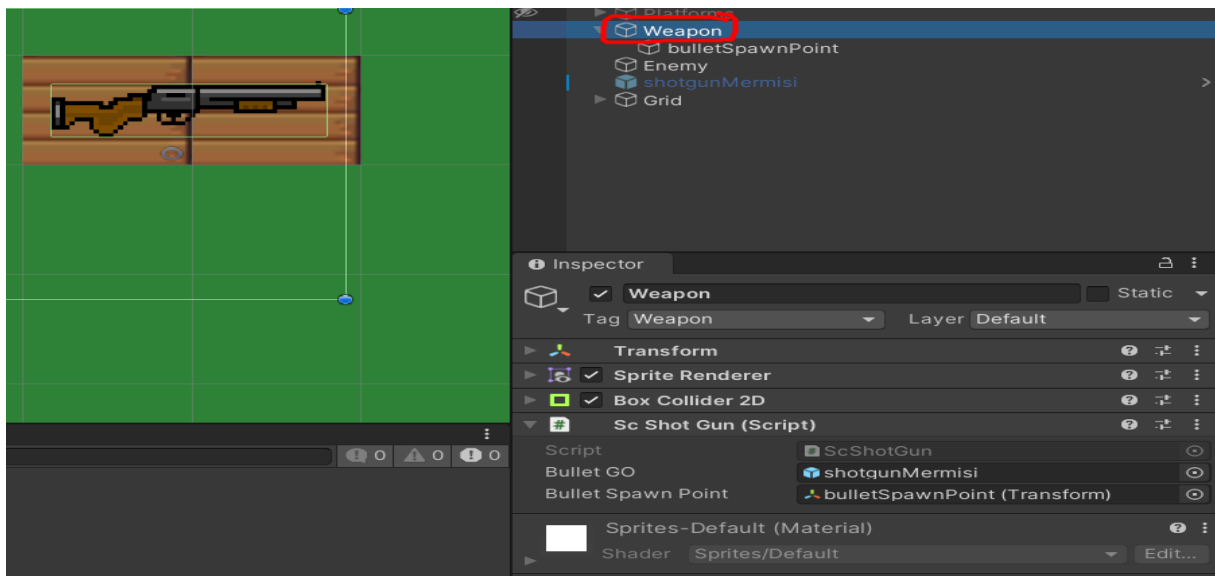
Rigidbody2D componentini tanımladık. Animator'da eAnim tanımladık. isInChaseRange ve isInAttackRange gibi komutlarla düşmanın takip etmesini kontrol ediyoruz. Target de Player tag'ini takip etmesini yazdık. MoveCharacter ile düşmanı da hareket ettirdik.

```

24 private void Start()
25 {
26     rbEnemy = GetComponent<Rigidbody2D>();
27     eAnim = GetComponent<Animator>();
28     target = GameObject.FindWithTag("Player").transform;
29 }
30 private void Update()
31 {
32     eAnim.SetBool("isRunning", isInChaseRange);
33
34     isInChaseRange = Physics2D.OverlapCircle(transform.position, checkRadius, whatIsPlayer);
35     isInAttackRange = Physics2D.OverlapCircle(transform.position, attackRadius, whatIsPlayer);
36
37     dir = target.position - transform.position;
38     float angle = Mathf.Atan2(dir.y, dir.x) * Mathf.Rad2Deg;
39     dir.Normalize();
40     movement = dir;
41     if(shouldRotate)
42     {
43         eAnim.SetFloat("X", dir.x);
44         eAnim.SetFloat("Y", dir.y);
45     }
46 }
47
48 private void FixedUpdate()
49 {
50     if(isInChaseRange && !isInAttackRange)
51     {
52         MoveCharacter(movement);
53     }
54     if(isInAttackRange)
55     {
56         rbEnemy.velocity = Vector2.zero;
57     }
58 }
59
60 private void MoveCharacter(Vector2 dir)
61 {
62     rbEnemy.MovePosition((Vector2)transform.position + (dir * speed * Time.deltaTime));
63 }

```

Weapon objectimiz Player'ımızın kullanacağı silahtır. Bu silahın ele alınması, silahın döndürülmesi ve ateş etmesi gibi özellikleri var.



Çeşitli componentlara sahip gene weapon objectimiz.

ScShotGun scriptimiz de çeşitli özelliklerimiz var.

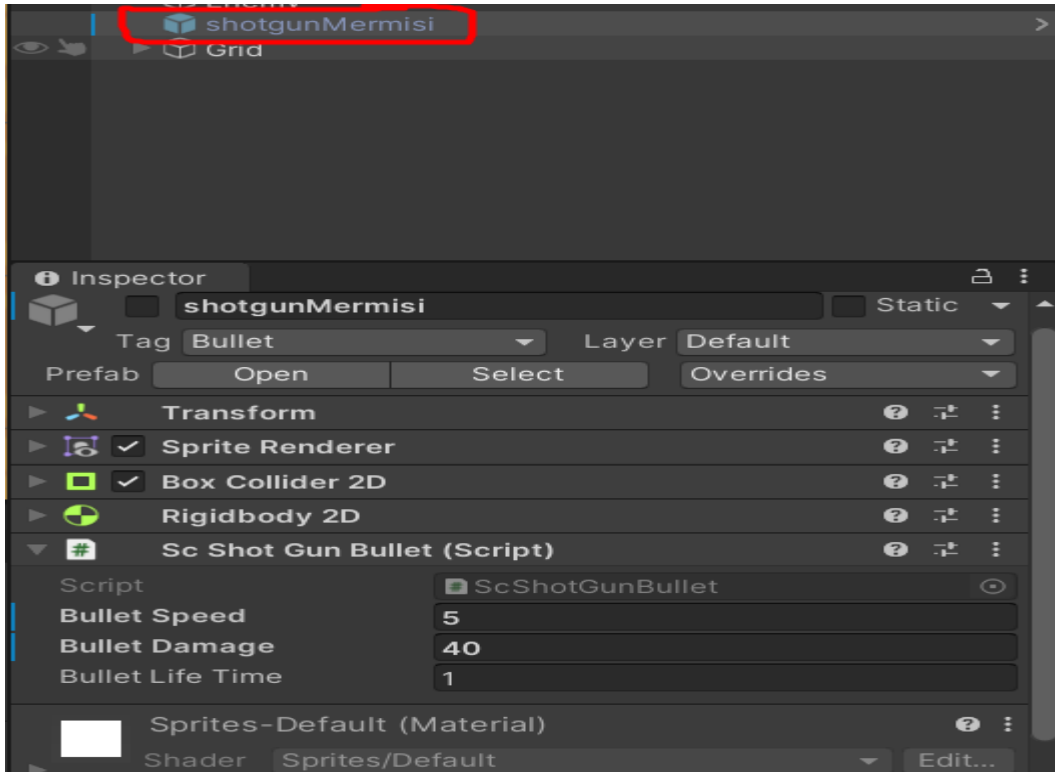
```
public class ScShotGun : MonoBehaviour
{
    //mermi için nesne oluşturuyoruz
    [SerializeField] private GameObject BulletGO;

    //merminin başlangıç noktası
    [SerializeField] private Transform bulletSpawnPoint;

    public void shoot(float angle)
    {
        if(BulletGO!=null)
        {
            Vector3 positionBullet = bulletSpawnPoint.position;
            Quaternion rotationBullet = Quaternion.Euler(0,0, BulletGO.transform.eulerAngles.z+angle);
            GameObject bullet = Instantiate(BulletGO , positionBullet, rotationBullet);

            bullet.GetComponent<ScShotGunBullet>().shootBullet(angle);
        }
    }
}
```

bulletSpawnPointimiz'i Unity de silahın ağzı olacak şekilde ayarladım. Silah objectimizi genel olarak tek başına ele alamayız attığı mermi ile çok bağlantılıdır. shootBullet(angle) ile mesela mermiden çağırıyoruz.



shotgunMermi sınıfı ayrıca prefabs'da bulunuyor.

Script olarak ScShotGunBullet diye yeni bir script oluşturduk.

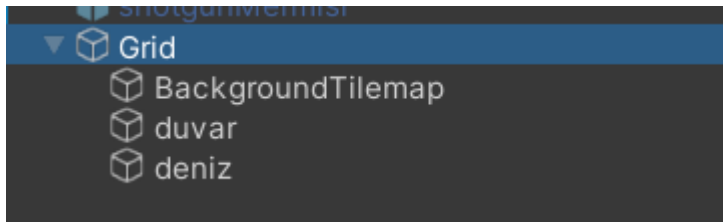
```
public class ScShotGunBullet : MonoBehaviour
{
    [SerializeField] float bulletSpeed;
    [SerializeField] float bulletDamage;
    [SerializeField] float bulletLifeTime=3f;
    Vector2 bulletSpeed2;

    public void shootBullet(float angle)
    {
        bulletSpeed2 = new Vector2(bulletSpeed*Mathf.Cos(angle*Mathf.PI/180),bulletSpeed*Mathf.Sin(angle*Mathf.PI/180));
        //Debug.Log("sadasd" +bulletSpeed2);
        //Debug.Log("sadasd" +bulletSpeed);
        this.GetComponent<Rigidbody2D>().velocity = bulletSpeed2;
        Destroy(this.gameObject, bulletLifeTime);
    }

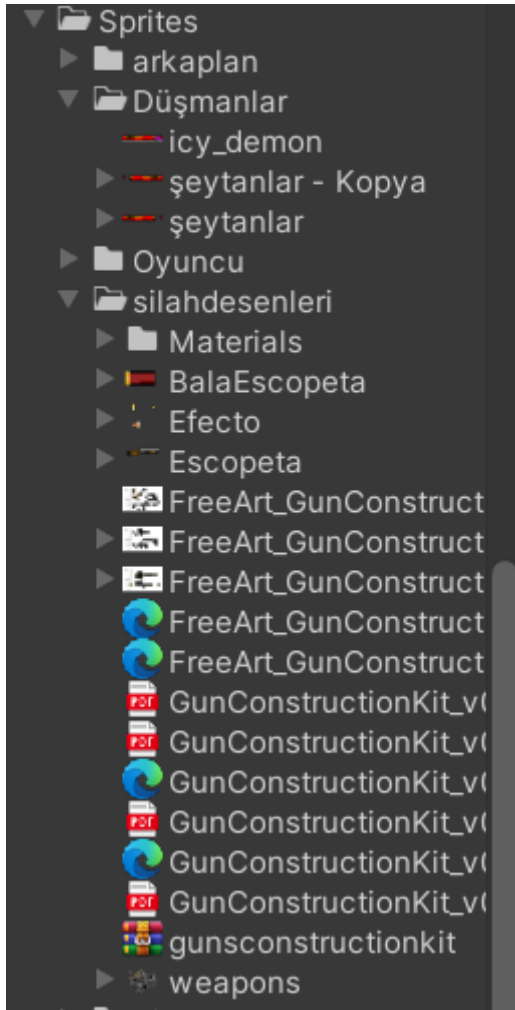
    void OnTriggerEnter2D(Collider2D col)
    {
        //Debug.Log("enemy - " + col.gameObject.name + " : " + gameObject.name + " : " + Time.time);

        if(col.CompareTag("Obstacle"))
        {
            BoxCollider2D m_ObjectCollider = this.gameObject.GetComponent<BoxCollider2D>();
            m_ObjectCollider.isTrigger = false;
        }
    }
}
```

Çeşitli tanımladığımız değişkenler var. bulletLifeTime=3f ayarladık çünkü merminin sonsuza kadar gitmesini istemedim. Burada OnTriggerEnter2D metodunda CompareTag ile obstacle gösterdiğimiz yerde duvara çarptığında trigger'ı false ettik ve mermi duvardan geçemedi. Bulletspeed2 vector2 içinde tanımlanan fonksiyonda çeşitli matematik fonksiyonları var.



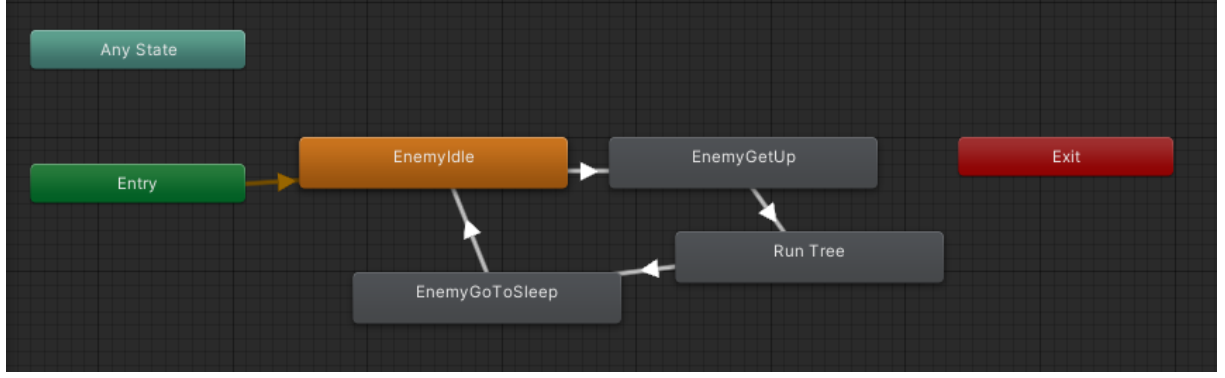
Burada Grid olarak 3 tane tilemap aldık. BackgroundTileMap de haritamızdaki yer deseni gözüktüyor. Duvar ve denizde durum farklı onları karakter içinden geçemiyor. Oralara gelince duruyor. Bunun nedeni duvar ve denizde TilemapCollider2D bulunması.



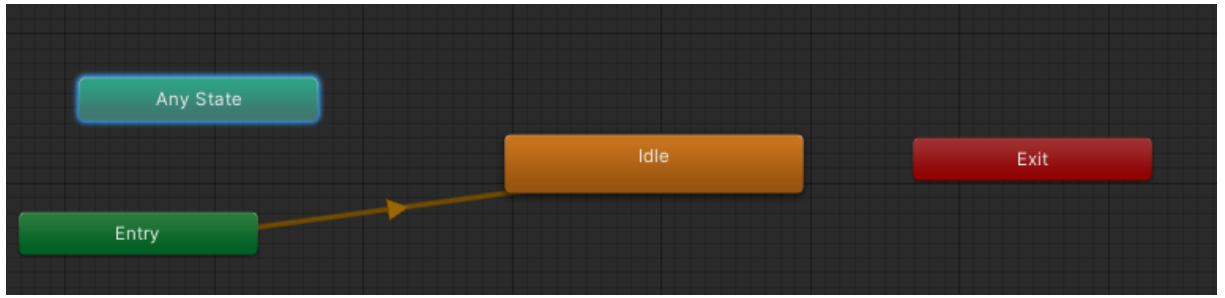
Sprites'da desenleri tutuyoruz. Silah desenlerimiz, oyuncumuzun desenleri, düşmanımızın deseni, mermimizin deseni... Bunları ise sprite mode'dan multiple'ı seçerek sprite editöre giriyoruz. Sprite Editörde slice diyerek o desenleri tek tek elde ediyoruz.



Animation klasöründe karakterimizin ve tüm animasyonu olabilecek karakterlerin .anim dosyalarını burada tutuyoruz. Burada çeşitli anim dosyalarını görüyoruz.



Enemy'mizin Animator'daki görünümü bu şekilde.



Player'ımızın Animator'daki görünümü

Genel olarak bu şekilde topladım. Kodlar, tablolar oyundaki görüntüler gibi şeyleri hazırladım. Videoda çok daha fazla şeyi anlattım ve en son oynanış da bulunuyor.

4- OYUNUN GELİRİ NEREDEN GELECEK ?

Oyun içine reklam alınarak para kazanılabilir. Çeşitli mağaza silah desenleri veya çeşitli farklı silahlar konularak satılabilirler. Oyun satışı normal olarak en temel gelir olacaktır.

5- KULLANICININ OYUN DENEYİMİ ?

Kullanıcı oyuna ilk girdiğinde menü sayfası ile karşılaşacak. Play tuşuna basıp oyuna girecek ve oyunu oynamaya başlayacak.

6- Video linkim : <https://youtu.be/rollASri0sc>

https://www.youtube.com/watch?v=rollASri0sc&ab_channel=HazarHascan

İyi günler dilerim.