

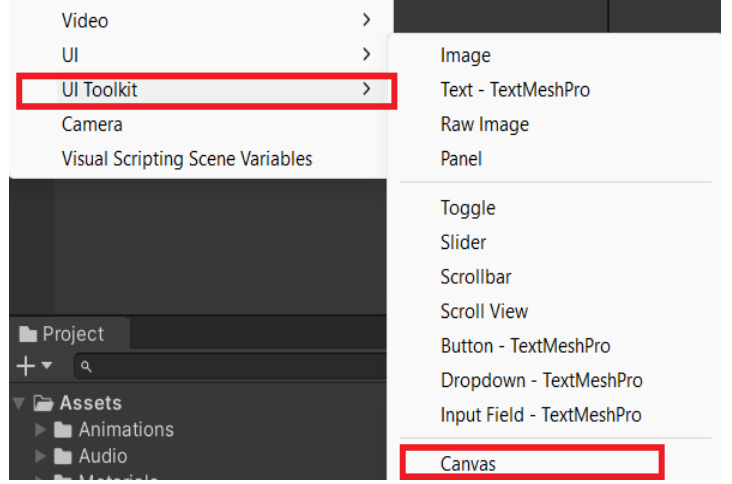
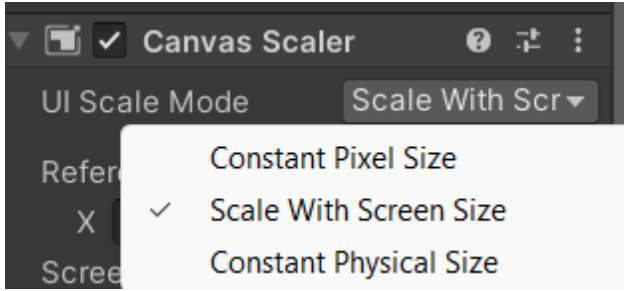
Öğrenci: Ahmet Yusuf Birdir

Numara: 21360859026

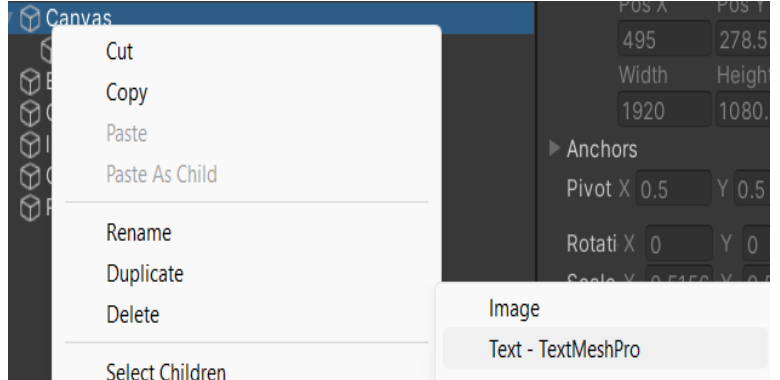
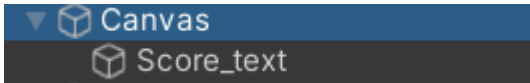
Kullanıcı Arayüzü

Score Gösterimi

Öncelikle sahneniniz hierarchy kısmında bir adet canvas nesnesi oluşturun. Ardından inspector penceresinden canvasınızın modunu ayarlayın.

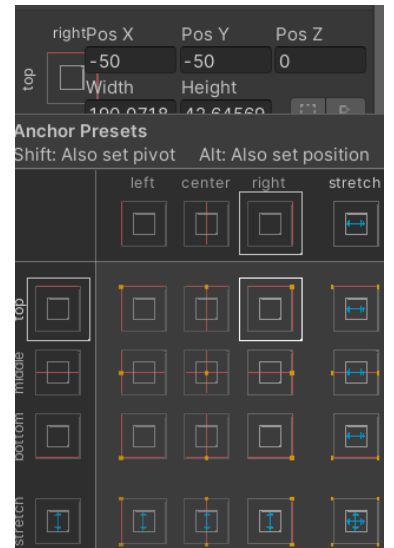
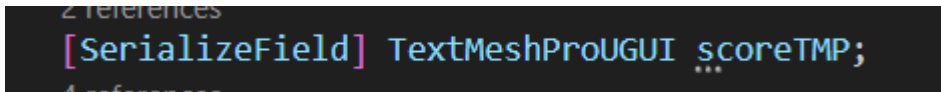


Daha sonra UI_Manager_sc isminde bir script dosyası oluşturun. Bu dosyanızı canvas nesnenize bağlayın. Score'u göstermek için canvas nesnenizin çocuğu olacak şekilde 1 adet text nesnesi oluşturun.



Score nesnenizin inspector penceresinden bölge ve koordinatları seçip scoru ekrande nerde göstermek istediğiniz ayarlayın.

Ardından UI_Manager_sc dosyasını açın ve içine score değişimi için bir adet TextMeshPro değişkeni oluşturun. Ardından unitydeki text nesnenizi bu değişkeninin ile sürükleyerek bağlayın.



Daha sonra scorunuzun ilk değerini start fonksiyonu içerisinde verin ve scorunuzu güncelleyecek olan fonksiyonu yazın.

```
scoreTMP.text = "Score: " + 0;
```

```
1 reference
public void UpdateScoreTMP(int score){
    scoreTMP.text = "Score: " + score;
}
```

Scorun oyuncumuza bağlı olması amacıyla player_sc isimli dosyayı açın ve ui_manager değişkeni oluşturun. Oluşturduğunuz nesneyi canvas nesneniz ile bağlayın.

5 references

```
UI_Manager_sc manager_UI;
```

```
manager_UI =GameObject.Find("Canvas").GetComponent<UI_Manager_sc>();
```

Bu nesne ile player bir enemy yok ettiğinde score nesnesine erişerek puan ekleyecek fonksiyonu çağırarak. Bu fonksiyon güncel score puanının üstüne puan ekleyecek ve güncellemesi için text güncelleyen ui_manager içindeki fonksiyonu çağırarak.

```
1 reference
public void UpdateScore(int points){

    score += points;

    if(manager_UI != null)
    {
        manager_UI.UpdateScoreTMP(score);
    }

}
```

Şimdi ise enemy_sc script dosyasını açın. Enemy_sc içinde OnTriggerEnter2d fonksiyonu içinde daha önceden oluşturduğumuz player nesnesinden updateScore fonksiyonuna istediğiniz puanı parametre olarak göndererek score'u yandaki gibi güncelleyin.

```
void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Player"){
        player.Damage();
        Destroy(this.gameObject);
    }
    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);

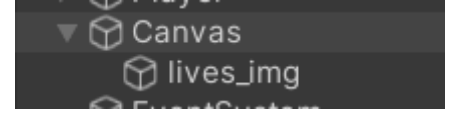
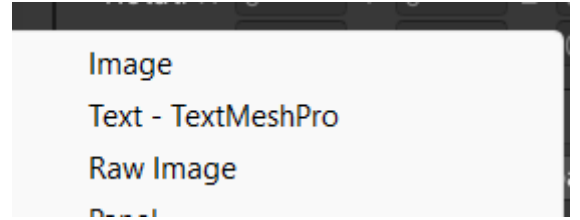
        if(player != null)
        {
            player.UpdateScore(10);
        }

        Destroy(this.gameObject);
    }
}
```

Can Gösterimi

Öncelikle Canvas nesnesinin içine UI kısmından bir adet raw image nesnesi oluşturun ve ekranda score nesnesini nerede gözükmelerini ayarladığınız gibi can kısmında nerede gözükeceğini ayarlayın.

Ardından ui_manager_sc dosyasını açın ve içine can miktarınızı barındıracak bir dizi ve unitydeki can nesnesine bağlanacak livesImg değişkeni oluşturun. Bu oyunda 3 can olacağından boyutunda oluşturuldu.

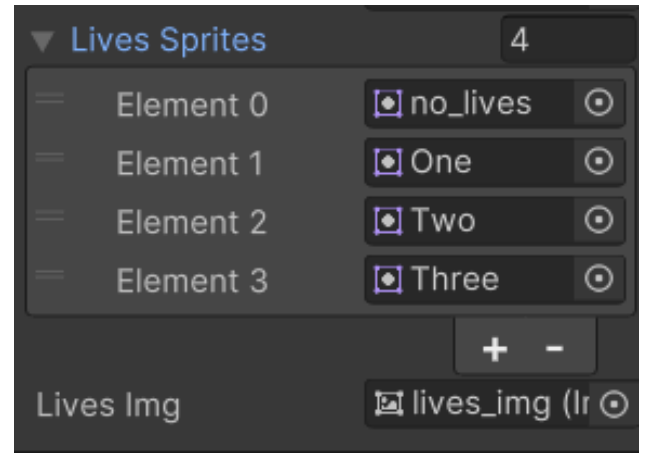
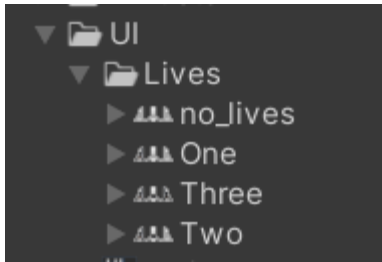


```
livesImg.sprite = livesSprites[3];
```

2 references

```
[SerializeField] Image livesImg;
```

LivesImg değişkeni ile sahnedeki lives img nesnesini bağlayın. Ardından dizinizin elemanlarına sprite klasöründeki UI dosyasının içindeki lives img'leri sırasıyla yerleştirin.



Start fonksiyonu içinde lives img'in değerini en başta 3 can olacak şekilde ayarlayın. Şimdi UI manager dosyasına bu diziyi ekranda görünüşünü düzenleyen fonksiyonu yazın.

```
public void UpdateLivesImg(int lives)
{
    livesImg.sprite = livesSprites[lives];
}
```

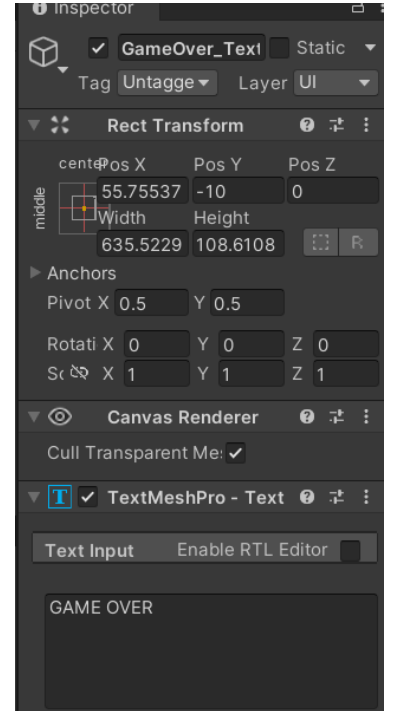
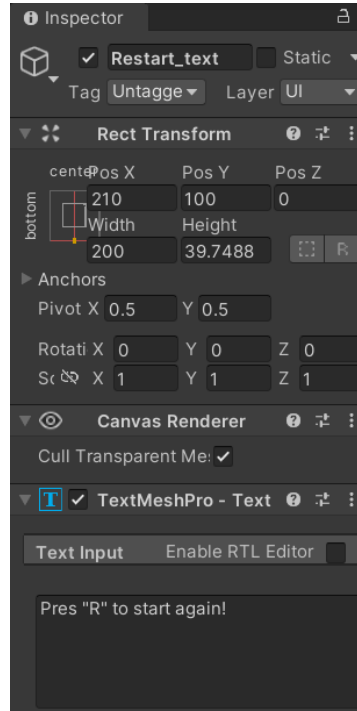
```
livesImg.sprite = livesSprites[3];
```

Son olarak player sc içerisinde hasar alındığında can görselini güncelleyecek fonksiyonu çağırın. Fonksiyonu kalkan bonusu aktif değil iken hasar aldığında değişecek şekilde ayarlayın.

```
public void Damage(){
    if(!isShieldBonusActive){
        this.playerHealth--;
        manager_UI.UpdateLivesImg(playerHealth);
    }
}
```

GameOver ve Yeniden Başlatma

Öncelikle sahnenize GameManager adında boş bir nesne oluşturun. Ardından GameManager_sc isimli bir script dosyası oluşturun ve GameManager nesnesine bağlayın. Ardından canvas nesnesine 2 adet text nesnesi oluşturun ve içine gerekli yazıları girin. UI_manager dosyasında gerekli değişkenleri oluşturun ve bu nesneleri unity içerisinde canvas nesnesine bağlayın.



Son olarak ui_manager içinde start fonksiyonunda oyun ilk başladığında bu nesneler gözükmeyecek şekilde ayarlayın.

```
gameOverTMP.gameObject.SetActive(false);
restartTMP.gameObject.SetActive(false);
```

Daha sonra UI manager içinde oyunu bitirmesi için çağrı yapacak bir fonksiyon oluşturun ve düzenleyin. Fonksiyon önce GameManager nesnesi içinden oyunun bittiğini gösterecek fonksiyonu çağırın ardından daha önceden gizlediğimiz oyun bitiş text nesnelerini görünür kılın ve gameOver flicker etkisini yapacak fonksiyonu çağırın.

```
1 reference
void GameOverSequence()
{
    if(gameManager != null)
    {
        gameManager.GameOver();
    }

    gameOverTMP.gameObject.SetActive(true);
    restartTMP.gameObject.SetActive(true);
    StartCoroutine(GameOverFlickerRoutine());
}
```

Flicker fonksiyonu oyun tekrar başlayana ya da kapanana kadar yarım bir süre açık kalıp bir süre kapalı kalacak ve bu şekilde devam edecek bir fonksiyon şeklinde yazın.

```
1 reference
IEnumerator GameOverFlickerRoutine()
{
    while(true)
    {
        gameOverTMP.gameObject.SetActive(true);
        yield return new WaitForSeconds(0.5f);
        gameOverTMP.gameObject.SetActive(false);
        yield return new WaitForSeconds(0.5f);
    }
}
```

GameManager_sc dosyasının içinde ise oyunun bitiren ve tekrar başlatacak fonksiyonları yazın. İstenilen tuşa basıldığında ve oyun bitişi onaylandığında oyun sahnesini yeniden yükleyin.

```
2 references
bool isGameOver = false;
0 references
```

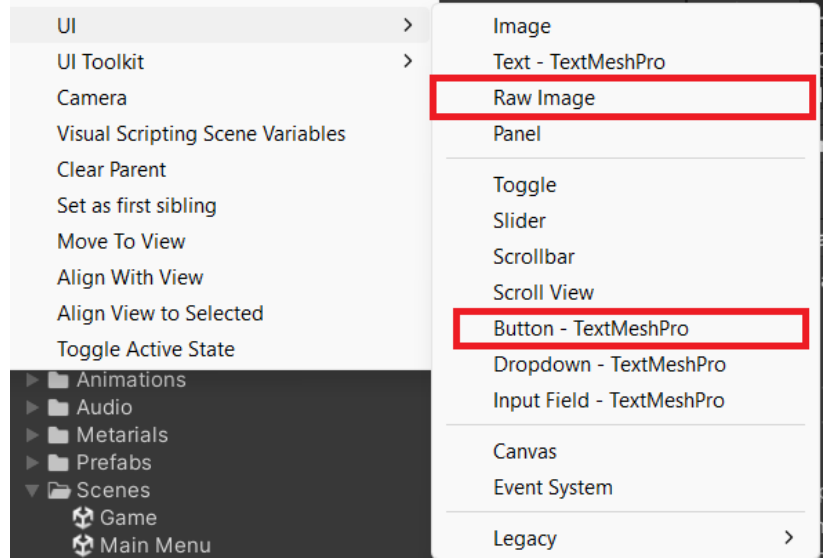
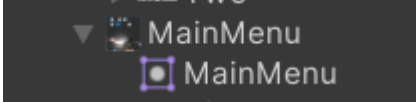
```
1 reference
public void GameOver()
{
    isGameOver = true;
}
```

```
0 references
void Update()
{
    if(Input.GetKeyDown(KeyCode.R) && isGameOver)
    {
        SceneManager.LoadScene("Game");
    }
}
```

Ana Menü

Yeni bir sahne oluşturun ve main menu adını verin. Bu sahneye bir adet canvas nesnesi oluşturun. İçine arkaplan, menu resmi için 2 adet img nesnesi ve 1 adet buton oluşturun.

Arkaplan için space_background resmini ilk img nesnenize verin. 2. img nesnesine ise main menu sprite'ını atayın.



Arkaplan resminizi ekranı kaplayacak formatta yerleştirin.

Son olarak Buton nesnenizin textini new game olarak ayarlayın.

