



İSTANBUL
GELİŞİM
ÜNİVERSİTESİ

İSTANBUL GELİŞİM MESLEK YÜKSEKOKULU
BİLGİSAYAR TEKNOLOJİLERİ BÖLÜMÜ
BİLGİSAYAR PROGRAMCILIĞI PROGRAMI

FRUIT NINJA 2D
MOBİL OYUN PROGRAMLAMA

FİNAL PROJE ÖDEVİ

HAZIRLAYAN
220111549 – Burak İŞLER

ÖDEV DANIŞMANI
Öğr. Gör. Adnan Kürşat Teke

İSTANBUL - 2024

ÖDEV TANITIM FORMU

YAZAR ADI SOYADI	: Burak İŞLER
ÖDEVİN DİLİ	: Türkçe
ÖDEVİN ADI	: Fruit Ninja 2D
BÖLÜM	: Bilgisayar Teknolojileri
PROGRAM	: Bilgisayar Programcılığı
ÖDEVİN TÜRÜ	: Final
ÖDEVİN TESLİM TARİHİ	: 29.05.2024
SAYFA SAYISI	: 17
ÖDEV DANIŞMANI	: Öğr. Gör. Adnan Kürşat Teke

KABUL VE ONAY SAYFASI

220111549 numaralı **Burak İŞLER**'in, **Fruit Ninja 2D** adlı çalışması, benim tarafımdan Final ödevi olarak kabul edilmiştir.

Adnan Kürşat Teke
Öğretim Görevlisi

ÖZET

Bu oyunumuzda öncelikle dünyaca ünlü olan 3D Fruit Ninja oyunu baz alındı ardından tasarım ve 2D olarak kodlandırma kısımları yapıldı oluşan sonuçta ise herkesin oynayabileceği klasik fruit ninjadan hiçbir eksiği kalmayan bir oyun ortaya çıkardık.

İÇİNDEKİLER

ÖZET	I
ÖN SÖZ	III
GİRİŞ	1
KODLAR	4

ÖN SÖZ

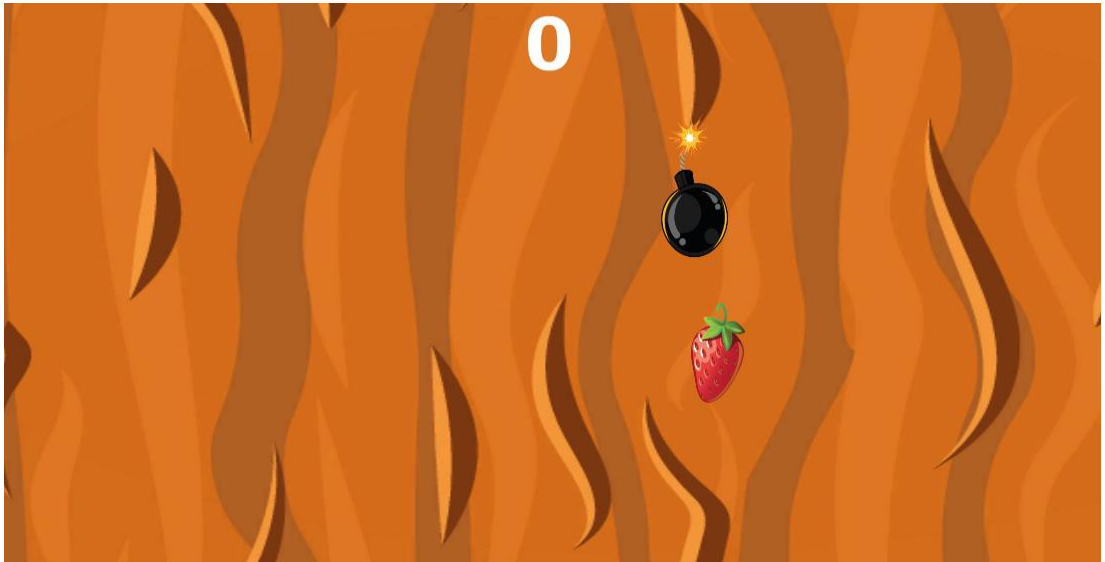
Projenin hazırlanış sürecinde her türlü yol gösterici olan, bilgi birikimiyle bana ve grup arkadaşlarıma yardımcı olan, olumlu davranışlarıyla beni heveslendiren değerli ödev danışmanım Öğr. Gör. Adnan Kürşat Teke'ye teşekkür ederim.

Burak İşler

GİRİŞ



Giriş olarak oyunumuzun açılış sahnesinden başlamak isterim. Bu kısımda bir play butonu ve oyunumuzun ismi ile oyuna başlıyoruz.

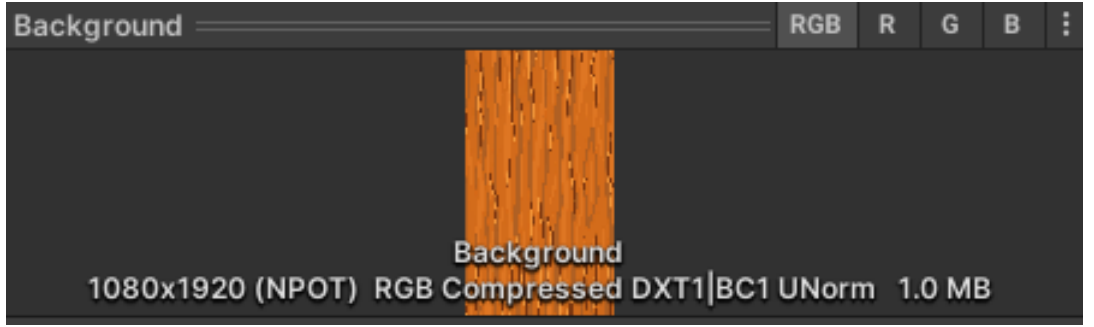


Devamında ise başla'ya tıkladığımızda ilk nesnelerimiz spawn oluyor ve oyunumuz başlıyor eğer yanlışlıkla bombayı kesersek tüm skorumuzu kaybediyoruz.



Burada gördükleriniz oyunumuza eklediğimiz tüm nesneler.

Bomba , Elma , Çilek , Karpuz ve Muz



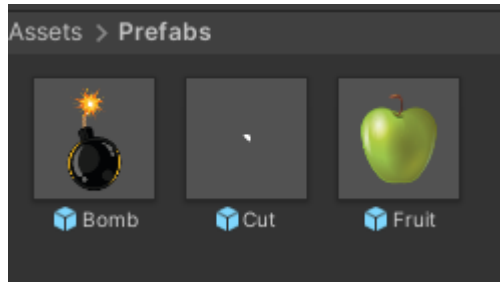
Arka plan için ise , bulduğumuz meyve kesilen tahtalardan esinlendiğimiz bir resim görüyoruz.



Bu kısımda sahnelerimiz mevcut oyunumuz 2 sahneden oluşan arayüzü basit bir 2D oyun.



Kod dosyalarımız.



Prefabs içeriğimiz.

KODLAR

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Cut : MonoBehaviour
6  {
7      private Manager manager;
8
9      void Start() {
10         manager = GameObject.FindObjectOfType<Manager>();
11     }
12     private void OnTriggerEnter2D(Collider2D other) {
13         if(other.gameObject.tag == "Fruit") {
14             Destroy(other.gameObject);
15             manager.score += 1;
16         }
17         else if(other.gameObject.tag == "Bomb") {
18             manager.restartGame();
19         }
20     }
21 }
22
```

Cut.cs

Bu kod "Cut" adında bir sınıfı temsil ediyor. Bu sınıf, nesneleri (meyve ve bomba) kesme işlemlerini yönetiyor. Oyun başladığında, sahnedeki `Manager` adlı yöneticiyi bulup `manager` değişkenine atıyor. `OnTriggerEnter2D` metodu, başka bir nesneyle çarpıştığında devreye giriyor. Eğer çarpışılan nesne "Fruit" (meyve) etiketi taşıyorsa, o nesneyi yok edip skoru 1 artırıyor. Eğer çarpışılan nesne "Bomb" (bomba) etiketi taşıyorsa, oyunu yeniden başlatıyor.

```
Diğer Dosyalar - Sword

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Sword : MonoBehaviour
6 {
7     public GameObject cutPrefab;
8     public float cutLifeTime;
9
10    private bool dragging;
11    private Vector2 swipeStart;
12    void Start()
13    {
14    }
15
16
17
18    void Update()
19    {
20        if(Input.GetMouseButtonDown(0)) {
21            dragging = true;
22            swipeStart = Camera.main.ScreenToWorldPoint(Input.mousePosition);
23        }
24        else if(Input.GetMouseButtonUp(0) && dragging) {
25            dragging = false;
26            cutSpawner();
27        }
28    }
29
30    private void cutSpawner() {
31        Vector2 swipeEnd = Camera.main.ScreenToWorldPoint(Input.mousePosition);
32        GameObject cutInstance = Instantiate(cutPrefab, swipeStart, Quaternion.identity);
33
34        cutInstance.GetComponent<LineRenderer>().SetPosition(0, swipeStart);
35        cutInstance.GetComponent<LineRenderer>().SetPosition(1, swipeEnd);
36
37        Vector2[] collidePoints = new Vector2[2];
38        collidePoints[0] = Vector2.zero;
39        collidePoints[1] = swipeEnd - swipeStart;
40
41        cutInstance.GetComponent<EdgeCollider2D>().points = collidePoints;
42
43        Destroy(cutInstance, cutLifeTime);
44    }
45
46 }
```

Sword.cs

Bu kod ise "Sword" adında bir sınıfı temsil ediyor. Bu sınıf, ekranda fare ile sürükleyerek kılıç hareketleri yapmanızı sağlıyor. Sol fare tuşuna basıldığında, sürükleme işlemi başlıyor ve tuşu bıraktığınızda bu sürükleme durduruluyor ve `cutSpawner` metodu çağrılıyor. `cutSpawner` metodu, sürüklemeye başladığınız ve bıraktığınız yerleri belirleyip, bu iki nokta arasında bir çizgi oluşturan bir kesme efekti (`cutPrefab`) oluşturuyor. Bu efekt, belirli bir süre sonra yok ediliyor (`cutLifeTime`). Kısacası, fare ile yapılan hareketler kılıç darbesine dönüştürülüyor ve ekranda bir kesme efekti oluşturuluyor.

```
Diğer Dosyalar | SpawnerController

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class SpawnerController : MonoBehaviour
6  {
7      public GameObject prefab;
8
9      public float minX;
10     public float maxX;
11
12     public Sprite[] sprites;
13
14     public float startTime;
15     private float time;
16     void Start()
17     {
18     }
19
20
21
22     void Update()
23     {
24         if(time <= 0) {
25             spawnObject();
26             time = startTime;
27         }
28         else {
29             time -= Time.deltaTime;
30         }
31     }
32
33     private void spawnObject() {
34         GameObject newPrefab = Instantiate(prefab);
35         newPrefab.transform.position = new Vector2(
36             Random.Range(minX, maxX),
37             transform.position.y
38         );
39
40         Sprite randomSprite = sprites[Random.Range(0, sprites.Length)];
41         newPrefab.GetComponent<SpriteRenderer>().sprite = randomSprite;
42     }
43 }
44
```

SpawnerController.cs

Bu koda da "SpawnerController" adında bir sınıf temsil ediliyor. Bu sınıf, belirli aralıklarla ekranda rastgele pozisyonlarda yeni nesneler oluşturuyor. `Update` metodunda, bir zamanlayıcı çalışıyor ve sıfıra ulaştığında `spawnObject` metodu çağrılarak yeni bir nesne oluşturuluyor. `spawnObject` metodu, prefab nesnesini alıp belirli bir x aralığında rastgele bir pozisyonda sahneye yerleştiriyor ve bu nesneye rastgele bir sprite atıyor. Yani, bu kod belirli aralıklarla ekranda rastgele yerlerde farklı görünümlerde nesneler üretmeyi sağlıyor.

```
Diğer Dosyalar playbutonu
1 using UnityEngine;
2 using UnityEngine.SceneManagement;
3 using UnityEngine.UI;
4
5 public class playbutonu : MonoBehaviour
6 {
7     public Button Button;
8
9
10    void Start()
11    {
12        if (Button != null)
13        {
14            Button.onClick.AddListener(OnButtonClicked);
15        }
16        else
17        {
18            Debug.LogError("Button is not assigned in the inspector.");
19        }
20    }
21
22    void OnButtonClicked()
23    {
24        SceneManager.LoadScene(1);
25    }
26 }
```

Playbutonu.cs

Bu kodda ise oyumuzun start sahnesinde ekrana gelen BAŞLA butonu bulunmakta.
Buton ataması ve click ile birlikte 0. sahneden 1. sahneye geçiş yapılmasına yarıyor.

```
Diğer Dosyalar ObjectMove
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class ObjectMove : MonoBehaviour
6 {
7     public Rigidbody2D rb;
8
9     public float minX;
10    public float maxX;
11    public float minY;
12    public float maxY;
13
14    public float lifetime;
15    void Start()
16    {
17        rb.velocity = new Vector2(
18            Random.Range(minX, maxX),
19            Random.Range(minY, maxY)
20        );
21
22        Destroy(gameObject, lifetime);
23    }
24 }
25
```

ObjectMove.cs

Bu kod "ObjectMove" adında bir sınıfı temsil ediyor. Bu sınıf, bir nesnenin rastgele bir hızda hareket etmesini ve belirli bir süre sonra yok olmasını sağlıyor. `Start` metodunda, nesnenin `Rigidbody2D` bileşenine rastgele bir hız değeri atanıyor (x ve y yönünde), böylece nesne sahnede hareket etmeye başlıyor. Ayrıca, nesne belirli bir süre (`lifetime`) sonra otomatik olarak yok ediliyor. Kısacası, bu kod, nesnelerin sahnede rastgele yönlerde hareket etmesini ve belirli bir süre sonra silinmesini sağlıyor.

```
Diğer Dosyalar Manager
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using UnityEngine.SceneManagement;
6
7  public class Manager : MonoBehaviour
8  {
9      public int score;
10     public Text scoreText;
11     void Start()
12     {
13     }
14
15     // Update is called once per frame
16     void Update()
17     {
18         gameScore();
19     }
20
21     private void gameScore() {
22         scoreText.text = score.ToString();
23     }
24
25     public void restartGame() {
26         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
27     }
28 }
29
30
```

Manager.cs

Bu kod "Manager" adında bir sınıfı temsil ediyor. Bu sınıf, oyunun skor takibini yapıyor ve gerektiğinde oyunu yeniden başlatıyor. `Update` metodunda, `gameScore` metodu çağrılarak `scoreText` adlı UI öğesine güncel skor yazılıyor. `restartGame` metodu ise çağrıldığında, mevcut sahneyi yeniden yükleyerek oyunu baştan başlatıyor. Kısacası, bu kod, oyunun skorunu güncellemeyi ve oyunu yeniden başlatmayı sağlıyor.

Kaynakça

- https://www.youtube.com/watch?v=xTT1Ae_ifhM
- <https://gamedevacademy.org/how-to-create-a-fruit-ninja-game-in-unity/>
- <https://github.com/topics/fruit-ninja?l=c%23>
- https://www.youtube.com/watch?v=AEjqrV_YM9U
- <https://dev.to/devdevcharlie/motion-controlled-fruit-ninja-game-using-three-js-tensorflow-js-18de>