

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
SEMİNER DERSİ PROJESİ

Unity ile Metin Tabanlı Oyun Geliştirme

Ayşenur Erkin

2022-2023 BAHAR DÖNEMİ

ÖNSÖZ

Bu rapor, Unity oyun motorunu kullanarak metin tabanlı oyun geliştirme projesi hakkındadır. Metin tabanlı oyunlar, oyuncuların hikâye ve senaryo yoluyla oyun dünyasına girmesine olanak tanıyan popüler bir oyun türüdür. Bu projede, metin tabanlı bir oyun geliştirmek için Unity ve C# programlama dili kullanılmıştır

Bu rapor, metin tabanlı oyun geliştirme süreci hakkında fikir sahibi olmak isteyen veya metin tabanlı bir oyun geliştirmek isteyen herkese faydalı olabilmesi için hazırlanmıştır.

Ayşenur Erkin

Bursa 2023

İÇİNDEKİLER

ÖNSÖZ	2
İÇİNDEKİLER	3
ÖZET	4
1. GENEL BİLGİLER	5
1.1. Unity Nedir?	5
1.2. Unity Arayüzüne Genel Bakış?.....	6
1.3. Metin Tabanlı Oyun Nedir?.....	8
1.4. Metin Tabanlı Oyunların Tarihçesi.....	8
2. KULLANILAN TEKNOLOJİLER	10
2.1. Unity Oyun Motoru	10
3. UNITY İLE METİN TABANLI OYUN GELİŞTİRME SÜRECİ.....	11
3.1. Proje Oluşturma.....	11
3.2. Senaryo Oluşturma.....	12
3.3. Kodlama.....	13
4. GEREKLİ KURULUMLAR	20
4.1. Unity Oyun Motorunun Kurulumu	20
5. SONUÇLAR.....	21
6. KAYNAKLAR	22

ÖZET

Bu rapor, Unity oyun motorunun kullanımıyla metin tabanlı oyunların geliştirilmesi sürecine odaklanmaktadır. Metin tabanlı oyunlar, birçok farklı oyuncu kitlesi tarafından sevilen ve popüler bir oyun türüdür. Bu oyunlar, metinlerin ve hikayelerin kullanıldığı, oyuncuların seçimlerinin hikâyeyi etkilediği, interaktif bir deneyim sunar.

Rapor, metin tabanlı oyunların ne olduğuna ve neden popüler olduklarına bir giriş yaparak başlamaktadır. Ardından Unity' nin genel bilgileri verilmektedir, Unity' nin ne olduğu ve arayüzüne genel bir bakış sunulmaktadır. Daha sonra metin tabanlı oyunların tanımı yapılarak, bu oyun türünün tarihçesi üzerinde durulmaktadır.

Raporun ikinci bölümünde, kullanılan teknolojiler ele alınmaktadır. Bu bölümde Unity Oyun Motoru ve onun metin tabanlı oyun geliştirmedeki rolü açıklanmaktadır.

Üçüncü bölüm, Unity ile metin tabanlı oyun geliştirme sürecini detaylı olarak ele almaktadır. Proje oluşturma adımları, senaryo oluşturma ve kodlama süreci gibi konular bu bölümde açıklanmaktadır.

Dördüncü bölüm, gerekli kurulumları içermektedir. Bu bölümde Unity Oyun Motorunun nasıl kurulacağına dair adımlar anlatılmaktadır.

Beşinci bölümde, raporun sonuçları sunulmaktadır.

Raporun altıncı ve son bölümünde ise kullanılan kaynaklar belirtilmektedir.

1. GENEL BİLGİLER

1.1. Unity Nedir?

Unity, bir oyun geliştirme platformudur. Geliştiricilere oyunlar, simülasyonlar ve diğer interaktif içerikler oluşturmak için kullanılan bir yazılımdır. Unity, kullanıcı dostu bir arayüze sahiptir ve çeşitli platformlarda çalışabilen oyunlar geliştirmek için güçlü bir altyapı sunar.

Unity, 2D ve 3D oyunlar için geniş bir özellik yelpazesi sunar. Geliştiriciler, karmaşık oyun dünyalarını oluşturmak, nesneleri yerleştirmek, fizik tabanlı etkileşimler eklemek, animasyonları oluşturmak ve oyun mekaniğini programlamak gibi birçok görevi yerine getirebilirler.

Unity' nin sunduğu avantajlardan biri, geliştiricilere hızlı ve kolay bir oyun geliştirme süreci sağlamasıdır. Bu, hazır özellikler ve kaynaklar sayesinde mümkün olur. Unity, geliştiricilere birçok oyun geliştirme aracı sunar ve bunlar oyun yapımını hızlandırırken, aynı zamanda karmaşık işlemleri kolaylaştırır.

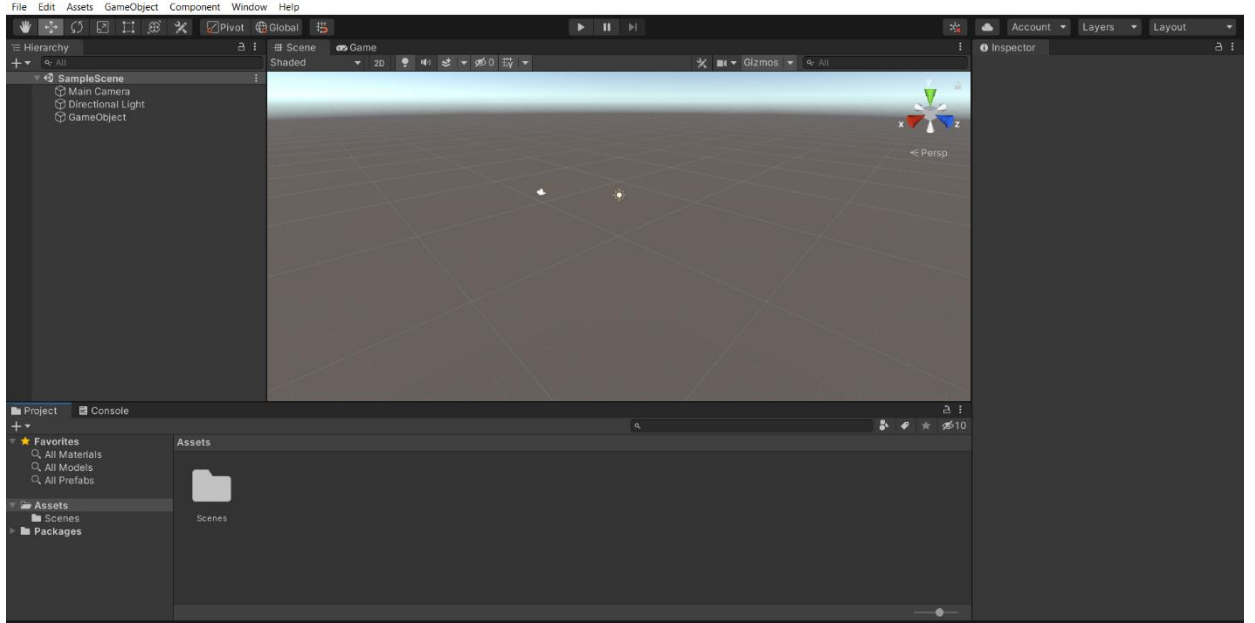
Unity' nin popülerliği, erişilebilir olmasıyla da ilişkilidir. Unity' nin kullanıcı dostu arayüzü ve kolay öğrenilebilir yapısı, yeni başlayanlar için bile oyun geliştirme sürecini daha erişilebilir hale getirir. Ayrıca, Unity' nin çoklu platform desteği, geliştiricilerin oyunlarını farklı platformlarda yayınlama ve daha geniş bir kitleye ulaşma imkânı sağlar.

Bunların yanı sıra, Unity' nin büyük bir topluluğunun olması da avantajlıdır. Unity' nin aktif ve büyük bir kullanıcı topluluğu vardır. Bu topluluk, geliştiricilerin sorunlarına çözüm bulmalarına yardımcı olur, rehberlik sağlar ve fikir alışverişinde bulunmalarını sağlar. Bu da geliştiricilerin destek ve yardım kaynaklarına kolayca erişebilmelerini sağlar.

Sonuç olarak, Unity' nin sunduğu hazır özellikler, kaynaklar, erişilebilirlik, çoklu platform desteği, geniş özellik yelpazesi ve büyük topluluk gibi avantajlar, oyun geliştiricilerin tercih etmesinde önemli bir rol oynar. Bu faktörler, Unity' nin popülerliğini ve metin tabanlı oyunların geliştirilmesindeki etkinliğini artırır.

1.2. Unity Arayüzüne Genel Bakış

Unity' nin arayüzü, öncelikle kullanıcıların sahne, oyun objeleri ve bileşenler gibi kaynakları kolayca görebilecekleri bir çerçeve sunar. Buna ek olarak, Unity' nin kütüphaneleri, geliştiricilerin farklı fonksiyonları ekleyebilecekleri ve projelerinde kullanabilecekleri hazır bileşenler sunar. Bu bileşenler, örneğin ışıklandırma, fizik ve animasyon özelliklerini içerebilir.



Şekil 1 Unity Arayüzü

Unity Arayüzünün Temel Bölümleri:

- 1) Sahne (Scene) Paneli: Oyun dünyanızın 3D veya 2D temsilini gösteren bir görünümdür. Sahne Görünümü, oyun alanınızdaki nesnelerin düzenini görmek ve düzenlemek için kullanılır. Nesneleri yerleştirebilir, düzenleyebilir ve konumlandırabilirsiniz.
- 2) Oyun (Game) Paneli: Oyununuzun gerçek zamanlı bir ön izlemesini sağlar. Oyunu çalıştırdığınızda, oyunun nasıl görüneceğini bu görünümde görebilirsiniz. Oyununuzun performansını test etmek ve oynanışını denetlemek için kullanılır.
- 3) Proje (Project) Paneli: Projenizin dosya yapısını ve içeriğini görüntüler. Burada oyununuz için kullanılan tüm dosyaları, sahneleri, betikleri, grafikleri ve sesleri düzenleyebilirsiniz. Dosyaları sürükleyip bırakarak projenize ekleyebilir ve düzenleyebilirsiniz.

- 4) Hiyerarşi (Hierarchy) Paneli: Sahnedeki nesnelerin hiyerarşisini gösterir. Bu, sahnedeki nesneleri düzenlemenize, gruplandırmanıza ve hiyerarşik olarak organize etmenize olanak sağlar. Nesneleri sürükleyip bırakarak yerlerini değiştirebilir ve düzenleyebilirsiniz.
- 5) Inspector Paneli: Seçilen nesnenin özelliklerini ve bileşenlerini görüntüler. Nesnelerin konumlarını, dönüşlerini, ölçeklerini ve diğer özelliklerini buradan ayarlayabilirsiniz.
- 6) Console Paneli: Konsol, oyununuzun çalışması sırasında oluşan hataları, uyarıları ve hata ayıklama bilgilerini görüntüler. Oyununuzun çıktılarını takip etmek ve hata ayıklama işlemlerini gerçekleştirmek için kullanılır.

Unity' nin arayüzü, bu temel bileşenlerin yanı sıra diğer paneller, araçlar ve sekmeler içerir. Geliştiriciler, özelleştirilebilir bir çalışma alanı oluşturarak arayüzü ihtiyaçlarına göre düzenleyebilirler.

1.3. Metin Tabanlı Oyun Nedir?

Bir metin tabanlı oyun (text-based game), oyuncuların hikâye ve olayları takip ederek metinlerle etkileşime geçtiği bir oyun türüdür. Oyun, oyuncunun verdiği seçimlere göre ilerler ve çeşitli sonuçlar sunar.

Bu tür oyunlar, sadece metin tabanlı olmakla kalmayıp genellikle komutları veya seçenekleri yazarak oynanır. Örneğin, “Kuzey'e git” veya “Sandığı aç” gibi komutlar girilerek oyun ilerletilir. Oyuncular, verilen metin tabanlı ipuçlarından yola çıkarak çözülmesi gereken bulmacaları çözebilir veya karakterlerle etkileşim kurabilir.

1.4. Metin Tabanlı Oyunların Tarihçesi

Metin tabanlı oyunlar, oyun endüstrisinin erken dönemlerine dayanan bir tarihe sahiptir. 1960'lı yıllardan itibaren ortaya çıkan metin tabanlı oyunlar, anahtar kelimeler ve komutlar aracılığıyla oyunculara sorular soran ve yanıtlar veren basit etkileşimler sunuyordu. 1970'lerde ise “Adventure” (veya “Zork”) gibi oyunlarla popülerlikleri arttı. Bu dönemde, metin tabanlı oyunlar büyük bir takipçi kitlesi kazandı.



Şekil 2 Popüler Metin Tabanlı Macera Oyunu Zork

1980'lerde bilgisayarların yaygınlaşmasıyla metin tabanlı rol yapma oyunları (RPG) popülerlik kazandı. 1990'ların başlarında grafik tabanlı oyunların yükselişiyle bir gerileme yaşansa da MUD' lar (Multi-User Dungeon) metin tabanlı çok oyunculu oyunlar olarak popülerliğini korudu.

2000'lerden itibaren metin tabanlı oyunlar yeniden yükselişe geçti. Gelişmiş internet teknolojileri ve mobil cihazların yaygınlaşmasıyla erişilebilirlikleri arttı. Ayrıca, interaktif kurgu türündeki metin tabanlı oyunlar da popülerlik kazandı.

Sonu olarak, metin tabanlı oyunlar oyun endüstrisinin tarihinde önemli bir yere sahiptir. "Adventure" ve "Zork" gibi oyunlar, bu türün gelişiminde önemli rol oynamıştır. Metin tabanlı oyunlar, hala birçok oyuncunun ilgisini çeken bir oyun türü olarak varlığını sürdürmektedir.

2. KULLANILAN TEKNOLOJİLER

2.1. Unity Oyun Motoru

Unity oyun motorunu metin tabanlı oyun geliştirmek için tercih etmek, bir dizi önemli avantajı beraberinde getirir. İlk olarak, Unity’ nin kullanım kolaylığı geliştirme sürecini daha verimli hale getirir. Kullanıcı dostu arayüzü, geliştiricilere sahneleri oluşturma, nesneleri düzenleme, oyunun diğer bileşenlerini yönetme gibi görevleri kolayca gerçekleştirme imkânı sunar.

Ayrıca Unity, çoklu platform desteği sayesinde oyununuzun farklı platformlarda çalışmasını sağlar. Bu, oyununuzun web, mobil cihazlar veya masaüstü gibi çeşitli platformlarda erişilebilir olmasını sağlar ve kitleye daha geniş bir erişim sağlar.

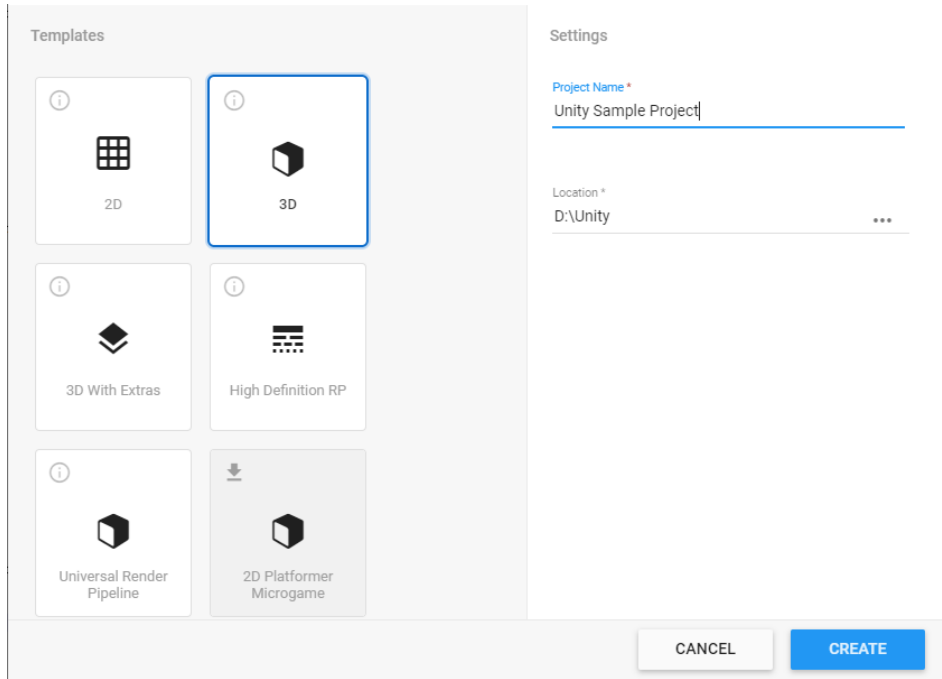
Unity’ nin geliştirici topluluğu ve geniş kaynaklara erişim imkânı, metin tabanlı oyun projelerimizi desteklemek için büyük bir avantajdır. Bir sorunla karşılaştığınızda veya yeni bir özelliği eklemek istediğinizde, Unity’ nin topluluğundan yardım alabiliriz. Bu, geliştirme sürecini kolaylaştırır ve daha hızlı ilerlememizi sağlar.

3. UNITY İLE METİN TABANLI OYUN GELİŞTİRME SÜRECİ

3.1. Proje Oluşturma

Aşağıdaki adımlar, projenin başarılı bir şekilde oluşturulması için takip edilmesi gereken adımları içermektedir:

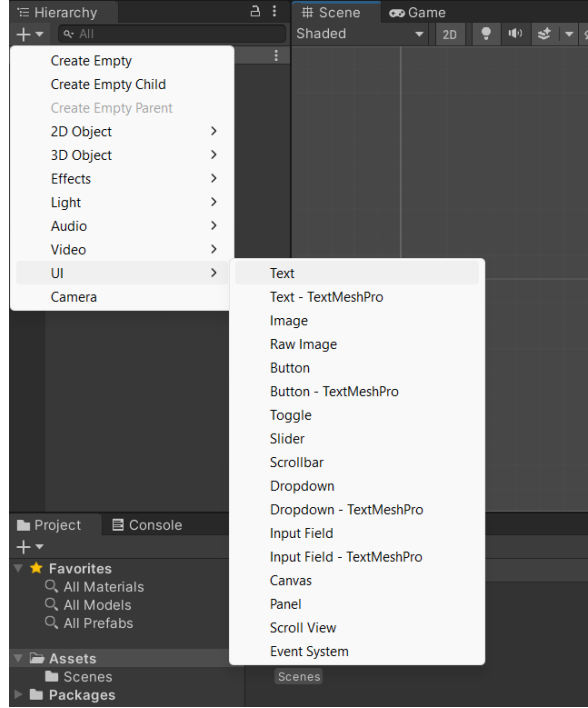
- 1) Unity Hub' u Açma: İlk adımda, Unity Hub uygulamasını başlatmamız gerekmektedir. Bu kısım Unity projelerimizi yönetmemizi sağlar.



Şekil 3 Unity Hub

- 2) Yeni Proje Oluşturma: Unity Hub ekranında, “New Project” butonuna tıklayın. Bu buton, yeni bir proje oluşturmanızı sağlayan seçenekleri açacaktır.
- 3) Proje Adı: “Project Name” alanına, oluşturacağınız projenin adını yazın.
- 4) Konum Seçimi: “Location” alanında, proje dosyalarının kaydedileceği konumu belirleyin.
- 5) Şablon Seçimi: “Templates” alanında, projeniz için kullanmak istediğiniz şablonu seçin. Bu raporda, metin tabanlı bir oyun geliştireceğimizden dolayı, 2D şablonunu seçmeniz gerekecektir.
- 6) Proje Oluşturma: Gerekli alanları doldurduktan sonra, “Create” butonuna tıklayın. Unity, projenin dosyalarını oluşturmak için çalışmaya başlayacaktır.

- 7) Unity Editörü Açma: Unity, projenizi açmak için Unity Editörü' nü otomatik olarak başlatacaktır.
- 8) Proje Çalışması: Artık proje üzerinde çalışmaya başlayabilirsiniz. Örneğin, yeni bir metin eklemek için “UI> Text” menüsü kullanabiliriz.



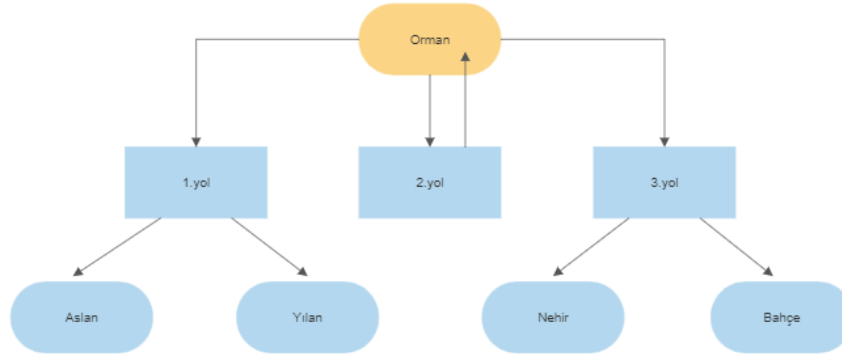
Şekil 4 Yeni Metin Ekleme için UI> Text Menüsü

3.2. Senaryo Oluşturma

Metin tabanlı bir oyun geliştirirken, başarılı bir senaryo oluşturmak son derece önemlidir. Bu yüzden metin tabanlı oyunumuz için bir senaryo yazmamız gerekmektedir.

Oyunumuzun senaryosu şu şekildedir: “Oyuncu kendini büyülü bir ormanda bulur. Önünde üç farklı yol açılır. Birinci yol seçildiğinde, oyuncunun önüne aslanlı ve yılanlı olmak üzere iki farklı yol daha çıkar. İkinci yol seçildiğinde, yol büyük bir kaya tarafından kapatılmıştır ve oyuncu geri dönerek yeni bir seçim yapmak zorunda kalır. Üçüncü yol seçildiğinde, oyuncu burada nehir boyunca ilerlemek veya bahçeye girmek gibi iki farklı seçenek arasından birini seçmek zorundadır.”

Senaryo



Şekil 5 Oyunun Akışını Gösteren Şema

3.3. Kodlama

Oyunumuz metin tabanlı bir oyun olduğu için şekil 4’ de gösterildiği metin elementini eklememiz gerekmektedir. Oluşturduğumuz metin kutusundaki yazıları istediğimiz gibi değiştirebilmek için bir script dosyası oluşturmamız gerekmektedir. Oluşturduğumuz script dosyasını çift tıkladığımızda, şekil 6’ da gösterilen kodlar ekranda görüntülenecektir.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class YazıDegistirme : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     void Start()
9     {
10
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16
17     }
18 }
19
```

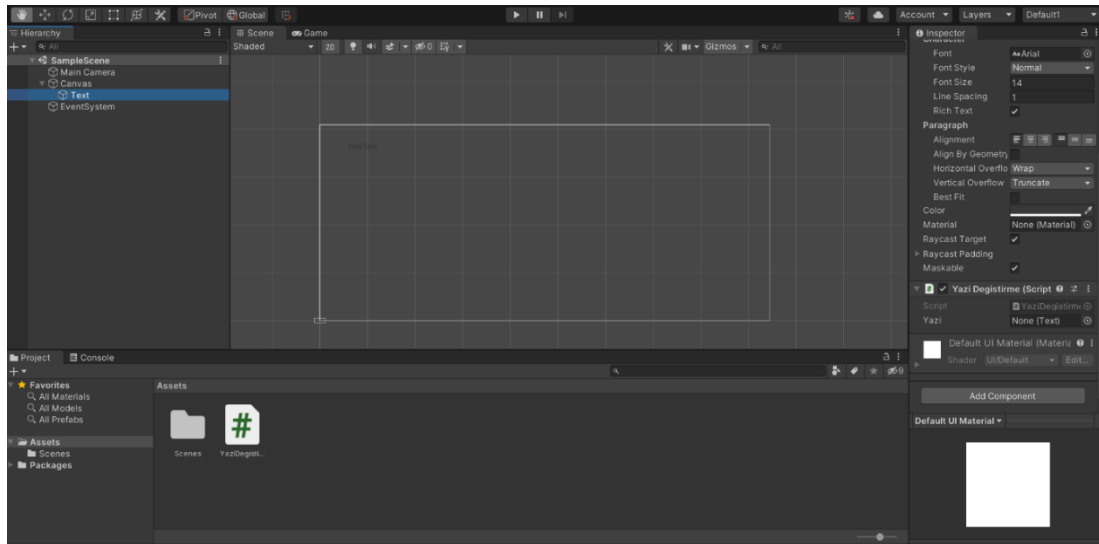
Şekil 6 Kod 1

Devamında, şekil 7’ de gösterildiği gibi bir yazı değişkeni eklememiz gerekmektedir ve bu yazı değişkenini metin elementiyle ilişkilendireceğiz. Böylelikle script dosyası aracılığıyla yazıyı istediğimiz gibi değiştirebileceğiz.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class YazıDegistirme : MonoBehaviour
7 {
8     public Text yazı;
9     // Start is called before the first frame update
10    void Start()
11    {
12        yazı.text = "Merhaba, Macera Oyuna Hosgeldin!\nDevam etmek için Enter'a tikla.";
13    }
14
15    // Update is called once per frame
16    void Update()
17    {
18    }
19 }
20
21
```

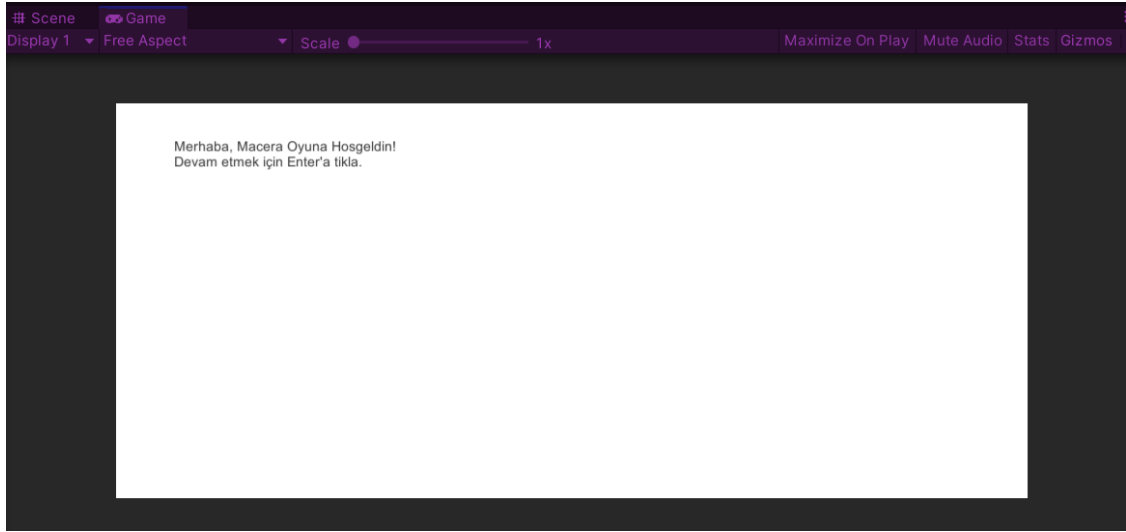
Şekil 7 Kod 2

Unity’ e geçtiğimizde, kodumuzdaki metin elementini belirtmemiz istenecektir. Bunun için, metin elementimizi sahip olduğumuz Text bileşeniyle ilişkilendirmemiz gerekmektedir. Bu ilişkilendirmeyi yapabilmek için, kodumuzdaki metin elementini seçip, Inspector panelindeki Text bileşenine sürükleyip bırakmamız gerekmektedir. Böylece metin elementini doğrudan yazdığımız koddaki değişkenle bağlamış olacağız.



Şekil 8 Unity Inspector Panelinde Text Bileşeni ile Metin Elementini İlişkilendirme

Kodu çalıştırdığımızda ekranda çıkan görüntü şekil 9’ daki gibidir.



Şekil 9 Metin Elementinden Oluşan Görüntü

Oyuncunun Enter tuşuna basıp basmadığını anlamak için, oyunun devam etmesi gereken anlarda kontrol etmemiz gerekmektedir. Bu kontrolü yapabilmek için Update metodu içinde, şekil 10’ da gösterildiği gibi bir if koşulu içine bir metod yazabiliriz. Böylelikle senaryomuz devam edecektir.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class YazıDegistirme : MonoBehaviour
7 {
8     public Text yazı;
9     // Start is called before the first frame update
10    void Start()
11    {
12        yazı.text = "Merhaba, Macera Oyuna Hosgeldin!\nDevam etmek için Enter'a tıkla.";
13    }
14
15    // Update is called once per frame
16    void Update()
17    {
18        if(Input.GetKeyDown(KeyCode.Return)){
19            yazı.text = "Karanlık bir ormanda uyandın. Ac ve susuzsun. Onunde uc yol var hangisini secersin? \nBirinci yol için 1'e, ikinci yol için 2'ye, 3.yol için 3'e tıkla.";
20        }
21    }
22 }
23
```

Şekil 10 Kod 3

Şekil 11’ de kodda, “bolumler” adında özel (private) bir enum tanımlanmıştır. Enum, belirli bir veri kümesini temsil etmek için kullanılan bir veri türüdür. Enum, oyunun farklı bölümlerini sembolik olarak temsil etmektedir. Bu bölümler “orman”, “yol_1”, “yol_2”, “yol_3”, “aslan”, “yılan”, “nehir”, “bahçe” ve “giris” olarak adlandırılmıştır.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class YazıDegistirme : MonoBehaviour
7 {
8     public Text yazı;
9     private enum bolumler(orman, yol_1, yol_2, yol_3, aslan, yılan, nehir, ev, giris);
10    private bolumler aktifBolum;
11
12
13    // Start is called before the first frame update
14    void Start()
15    {
16        aktifBolum = bolumler.giris;
17    }
18
19    // Update is called once per frame
20    void Update()
21    {
22        if(aktifBolum == bolumler.giris){
23            bolumGiris();
24        }else if(aktifBolum == bolumler.orman){
25            bolumOrman();
26        }
27    }
28
29
30    void bolumGiris()
31    {
32        yazı.text = "Merhaba, Macera Oyuna Hosgeldin!\nDevam etmek için Enter'a tıkla.";
33        if(Input.GetKeyDown(KeyCode.Return)){
34            aktifBolum = bolumler.orman;
35        }
36    }
37
38    void bolumOrman()
39    {
40        yazı.text = "Karanlık bir ormanda uyandın. Ac ve susuzsun. Onunde uc yol var hangisini secersin? \nBirinci yol için 1'e, ikinci yol için 2'ye, 3.yol için 3'e tıkla.";
41    }
42 }
43
```

Şekil 11 Kod 4

Oyunun ilk başladığında, aktif bölümü “giris” bölümü olarak belirlemiştik. Şekil 12’deki kodda, Update metodu aktif bölümü sürekli olarak güncellemektedir.

Bu güncelleme işlemi, her güncelleme adımında (Update) aktif bölümün değerini kontrol ederek gerçekleştirilir. Oyunda ilerledikçe veya belirli koşullar sağlandıkça aktif bölüm değişebilir. Update metodu içindeki kod, bu değişiklikleri takip ederek aktif bölümü güncellemektedir. Böylece oyun, her güncelleme adımında doğru bölümde devam edebilmektedir.

```
6 public class YazıDegistirme : MonoBehaviour
7 {
8     public Text yazı;
9     private enum bolumler{orman, yol_1, yol_2, yol_3, aslan, yilan, nehir, bahce, giris};
10    private bolumler aktifBolum;
11
12
13    // Start is called before the first frame update
14    void Start()
15    {
16        aktifBolum = bolumler.giris;
17    }
18
19    // Update is called once per frame
20    void Update()
21    {
22        if(aktifBolum == bolumler.giris){
23            bolumGiris();
24        }else if(aktifBolum == bolumler.orman){
25            bolumOrman();
26        }else if(aktifBolum == bolumler.yol_1){
27            bolumYol_1();
28        }else if(aktifBolum == bolumler.yol_2){
29            bolumYol_2();
30        }else if(aktifBolum == bolumler.yol_3){
31            bolumYol_3();
32        }else if(aktifBolum == bolumler.aslan){
33            bolumAslan();
34        }else if(aktifBolum == bolumler.yilan){
35            bolumYilan();
36        }
37        else if(aktifBolum == bolumler.nehir){
38            bolumNehir();
39        }
40        else if(aktifBolum == bolumler.bahce){
41            bolumBahce();
42        }
43    }
44 }
```

Şekil 12 Kod 5

Şekil 13, 14 ve 15’ de ki metotlar bölümlerin oyun içindeki davranışını ve metin tabanlı açıklamalarını kontrol etmektedir.

```
53
54 void bolunOrman()
55 {
56     yazi.text = "Karanlık bir ormanda uyandın. Ac ve susuzsun. Onunde uc yol var hangisini secersin? \nBirinci yol icin 1'e, ikinci yol icin 2'ye, 3.yol icin 3'e tikla.";
57     if(Input.GetKeyDown(KeyCode.Alpha1)){
58         aktifBolun = bolunler.yol_1;
59     }else if(Input.GetKeyDown(KeyCode.Alpha2)){
60         aktifBolun = bolunler.yol_2;
61     }else if(Input.GetKeyDown(KeyCode.Alpha3)){
62         aktifBolun = bolunler.yol_3;
63     }
64 }
65
```

Şekil 13 Kod 6

```
66
67 void bolunYol_1()
68 {
69     yazi.text = "Birinci yola saptın ve uzun bir yuruyusun ardından karsına bir yol ayrımı daha çıktı. İlk yolun başında bir aslanın beklediğini gördün. İkinci yolun başında ise bir yılan beklemekte."
70     if(Input.GetKeyDown(KeyCode.A)){
71         aktifBolun = bolunler.aslan;
72     }else if(Input.GetKeyDown(KeyCode.Y)){
73         aktifBolun = bolunler.yılan;
74     }else if(Input.GetKeyDown(KeyCode.G)){
75         aktifBolun = bolunler.orman;
76     }
77 }
78
79 void bolunYol_2()
80 {
81     yazi.text = "Yamaktan düşen kaya parçaları yolu kapatmış. Geri dönmek için G'ye bas.";
82     if(Input.GetKeyDown(KeyCode.G)){
83         aktifBolun = bolunler.orman;
84     }
85 }
86
87 void bolunYol_3()
88 {
89     yazi.text = "Karsına bir yol ayrımı çıktı. Hem acsın hem de susuz.\nNehire doğru inmek için N'ye. Bahçeye doğru tırmanmak için B'ye, geri dönmek için ise G'ye bas.";
90     if(Input.GetKeyDown(KeyCode.N)){
91         aktifBolun = bolunler.nehir;
92     }else if(Input.GetKeyDown(KeyCode.B)){
93         aktifBolun = bolunler.bahce;
94     }else if(Input.GetKeyDown(KeyCode.G)){
95         aktifBolun = bolunler.orman;
96     }
97 }
98
```

Şekil 14 Kod 7

```
99
100 void bolunAslan()
101 {
102     yazi.text = "Aslanla karsılaşacak kadar cesur olduğunu bilmiyordum. Aslan seni bir lokmada yuttu.\nOyun bitti...";
103 }
104
105 void bolunYılan()
106 {
107     yazi.text = "Yılanın yanından geçmeye çalışırken yılan seni isirdi. Önce felc geçirdin ve acı içinde son nefesini verdin...\nOyun bitti...";
108 }
109
110 void bolunNehir()
111 {
112     yazi.text = "Nehirden kana kana su içtin ve yoluna devam edecek gücü kendinde bulduktan sonra karanlık ormandan kurtuldun...\nTebrikler!";
113 }
114
115 void bolunBahce()
116 {
117     yazi.text = "Bahçeye tırmanarak karnini doyurdun fakat yedigin meyveler zehirli olduğu için ormandan çıkamadan öldün...\nOyun bitti...";
118 }
119
```

Şekil 15 Kod 8

Tüm bu kodlamaların implementasyonuyla, metin tabanlı bir oyunun farklı bölümlerini ve oyuncunun seçimlerine bağlı olarak ilerlemesini başarılı bir şekilde gerçekleştirmiş oluruz. Bu kodlar, oyuncuya anlaşılır bir şekilde metinler sunar ve oyuncunun girdiği tuşlara göre oyunun ilerlemesini kontrol eder.

Böylece, oyunun akışını yönlendiren ve oyuncuya etkileşim imkânı sunan bir yapı oluşturmuş oluruz.

Sonuç olarak, bu implementasyon ile metin tabanlı oyununuzda oyunculara çeşitli seçenekler sunabilir, onların kararlarına bağlı olarak oyunun akışını değiştirebiliriz.

4. GEREKLİ KURULUMLAR

4.1. Unity Oyun Motorunun Kurulumu

Unity' de Metin tabanlı bir oyun geliştirmek için öncelikle Unity' yi kurmamız gerekmektedir.

Unity indirme linki: <https://unity3d.com/get-unity/download>

- 1) İndirilen Unity yükleyici dosyasını çalıştırın.
- 2) Yükleyici çalıştığında, "Unity Hub" adlı bir programın da yüklenmesi istenecektir.
- 3) Kurulum sihirbazının yönergelerini izleyerek ilerleyin.
- 4) Kurulum seçeneklerini belirledikten sonra, kurulumun tamamlanması için "İleri" veya "Devam Et" gibi ilerleme seçeneklerini seçerek devam edin.
- 5) Kurulum tamamlandığında, Unity' nin kullanılabilir hale gelmesi birkaç dakika sürebilir. Unity kurulumunu tamamladıktan sonra, Unity Hub veya doğrudan Unity Editörü üzerinden yeni projeler oluşturabilir ve geliştirmeye başlayabilirsiniz.

5. SONUÇLAR

Unity oyun motoru, metin tabanlı oyunların geliştirilmesi için kullanışlı bir araçtır. Unity'nin kullanıcı dostu arayüzü, geniş özellikleri ve çeşitli platformlarda dağıtım imkânı, metin tabanlı oyunlar oluşturmak isteyen geliştiricilere büyük bir avantaj sağlar.

Ayrıca, Unity'nin geniş ekosistemi ve topluluğu, geliştiricilere kaynaklara, eklentilere ve desteklere kolay erişim sağlar.

Unity'nin metin tabanlı oyun geliştirme için sunduğu olanaklar, yaratıcı fikirleri gerçeğe dönüştürme ve oyun deneyimini oyunculara sunma konusunda büyük bir potansiyel sunar.

Sonuç olarak, Unity oyun motoru, metin tabanlı oyun geliştirme için güçlü bir araçtır ve bu alanda yaratıcı ve etkileyici oyunlar oluşturmanıza yardımcı olabilir.

6. KAYNAKLAR

1. <https://learn.unity.com/tutorial/text-based-games>
2. <https://learn.unity.com/tutorial/recorded-video-session-text-adventure-game-part-1#>
3. <https://www.kodeco.com/14048776-how-to-make-an-adventure-game-like-king-s-quest>
4. https://en.wikipedia.org/wiki/Text-based_game#:~:text=A%20text%20game%20or%20text,of%20bitmap%20or%20vector%20graphics.
5. <https://learn.microsoft.com/en-us/archive/msdn-magazine/2014/august/unity-developing-your-first-game-with-unity-and-csharp>