



T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

SÜRÜKLE-BIRAK PROGRAMLAMA

İRFAN DERDİYOK

BİTİRME ÇALIŞMASI

DANIŞMANI : Dr.Öğr.Üyesi Salim CEYHAN

BİLECİK

26 Haziran 2022



T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

SÜRÜKLE-BIRAK PROGRAMLAMA

İRFAN DERDİYOK

BİTİRME ÇALIŞMASI

DANIŞMANI : Dr.Öğr.Üyesi Salim CEYHAN

BİLECİK

26 Haziran 2022

BİLDİRİM

Bu çalışmada bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

İmza

İRFAN DERDİYOK

Tarih: 26 Haziran 2022

ÖZET

BİTİRME ÇALIŞMASI

SÜRÜKLE-BIRAK PROGRAMLAMA

İRFAN DERDİYOK

Bilecik Şeyh Edebali Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Danışman: Dr.Öğr.Üyesi Salim CEYHAN

2022, 36 Sayfa

Jüri Üyeleri

İmza

.....
.....
.....

.....
.....
.....

Yazılım dünyasına ilk adımını atacak olan öğrenciler pek fazla detaya maruz kalmaktadırlar. Bu başlangıç için gereksiz detaylar öğrencilerin morallerini bozabilmekte ve öğrenme süreçlerini yavaşlatabilmektedir. Bu sorunun önüne geçebilmek için sadeleştirilmiş bir yazılım programlama dili oluşturmak arzu edilmektedir. Bu projeyle birlikte öğrenciler sadece programlamanın mantığını kavramakla uğraşacaklar. Sözdizimi ve detaylar arka planda işlenecek. Öğrencilerin daha kolay öğrenmesi adına programlamayı sürükle-bırak sistemi üzerine kurulmuştur.

ABSTRACT

THESIS

DRAG - DROP PROGRAMMING

İRFAN DERDİYOK

**Bilecik Şeyh Edebali University
Engineering Faculty
Department of Computer Engineering**

Advisor : Assist.Prof. Salim CEYHAN

2022, 36 Pages

Jury

Sign

.....
.....
.....

.....
.....
.....

Students who will take their first step into the software world are exposed to many details. Unnecessary details for this start can upset students' morale and slow down their learning processes. In order to avoid this problem, it is desirable to create a simplified software programming language. With this project, students will only deal with understanding the logic of programming. Syntax and details will be processed in the background. It is built on a drag-and-drop system for students to learn more easily.

ÖNSÖZ

Bitirme çalışmamın başından sonuna kadar emeği geçen ve beni bu konuya yönlendiren saygı deđer hocam ve danışmanım Sayın Dr.Öğr.Üyesi Salim CEYHAN’a tüm katkılarından ve hiç eksiltmediğı desteğinden dolayı teşekkür ederim.

İRFAN DERDİYOK

26 Haziran 2022

İÇİNDEKİLER

ÖNSÖZ	v
ŞEKİLLER TABLOSU	ix
1 GİRİŞ	1
1.1 Projenin İçeriği	1
1.2 Öğrenciler İçin Projenin Önemi	1
1.3 Benzer Projeler	1
1.3.1 Human resource machine	1
1.3.2 Scratch	2
1.3.3 MBlock	2
1.3.4 Projenin github linki	2
2 KULLANILAN YAZILIMLAR ve YÖNTEMLER	3
2.1 Unity Oyun Motoru	4
2.2 Github	4
3 SÜRÜKLE-BIRAK PROGRAMLAMA	5
3.1 Arayüz	5
3.1.1 Giriş menüsü	6
3.1.2 Ana menü	7
3.1.3 Kod objeleri menüsü	10
3.1.4 Kod menüsü	10
3.2 Fonksiyonlar	10
3.2.1 Sürükleyip bıraklama	10
3.2.2 Derleyici	12
3.2.3 Değişken oluşturma	13
3.2.4 Koşul oluşturma	14
3.2.5 Döngü oluşturma	16
3.2.6 Veri yazdırma	17
3.2.7 Matematiksel işlemler	18

3.2.8	Soru oluřturma ve kontrol etme	19
3.2.9	FPS sınırlandırma	20
4	SONUÇLAR VE ÖNERİLER	22
5	EKLER	23
5.1	Uygulama İerisinden Bazı Ekran Görüntüleri	23
	KAYNAKLAR	25
	ÖZGEÇMİŐ	26

ŞEKİLLER TABLOSU

Şekil 1.1	Human resource machine	2
Şekil 3.1	TextMesh pro paketini projeye ekleme	5
Şekil 3.2	Giriş menüsü	6
Şekil 3.3	Animation menüsü ve animasyonun oluşturulması	6
Şekil 3.4	Animasyonu döngüye alma	7
Şekil 3.5	Sahneler arası geçiş kod bloğu.	7
Şekil 3.6	Ana menü	8
Şekil 3.7	Ana menü hierarchy yapısı	8
Şekil 3.8	Content objesi bileşenleri	9
Şekil 3.9	Butona fonksiyon atama ve parametre değeri verme	9
Şekil 3.10	Sürükleyip bırakma fonksiyonu.	11
Şekil 3.11	Objeyi sabitleme fonksiyonu.	12
Şekil 3.12	Derleyici fonksiyonu.	13
Şekil 3.13	Değişken sınıfı.	14
Şekil 3.14	Değişken oluşturma fonksiyonun hierarchy sekmesindeki yapısı	14
Şekil 3.15	Koşul fonksiyonu.	15
Şekil 3.16	Koşul fonksiyonun hierarchy sekmesindeki yapısı	15
Şekil 3.17	Döngü fonksiyonu.	16
Şekil 3.18	Döngü fonksiyonun hierarchy sekmesindeki yapısı	16
Şekil 3.19	Veri yazdırma fonksiyonu.	17
Şekil 3.20	Yaz fonksiyonun hierarchy sekmesindeki yapısı	18
Şekil 3.21	Matematik fonksiyonun hierarchy sekmesindeki yapısı	18
Şekil 3.22	Matematik fonksiyonu.	19
Şekil 3.23	Soru oluşturma ve kontrol fonksiyonu.	20
Şekil 3.24	Fps sınırlandırma kod bloğu.	21
Şekil 5.1	Örnek soru ekran görüntüsü	23
Şekil 5.2	Fibonacci dizisi bulma	23
Şekil 5.3	Doğru cevap ekran görüntüsü	24
Şekil 5.4	Oyun oynanış ekran görüntüsü	24

Şekil 5.5	Oyun görev ekranı görüntüsü	24
-----------	---------------------------------------	----

1 GİRİŞ

1.1 Projenin İçeriği

Uygulama içerisinde bulunan hazır objeleri (kod parçacıklarını) sürükleyip bırak yaparak algoritma oluşturabilmeleri sağlanacak. Kullanıcının objelerle oluşturduğu algoritma arka planda koda çevrilecek. Eğer kullanıcı hata yaparsa öncelikle arka planda sorun çözülmeye çalışılacak. Arka planda çözülemeyecek bir hata ise kullanıcının rahatlıkla anlayabileceği bir uyarı verilecek. Uygulama içerisine egzersiz soruları eklenecek.

1.2 Öğrenciler İçin Projenin Önemi

Öğrencilerin yazılıma başlarken yaptıkları sözdizimi hataları sonucunda kodları çalışmayabiliyor. Bunun sonucunda öğrenciler o hatayı çözmek için zamanlarını harcayabiliyor. Motivasyonları kırılabiliyor. Bu gibi durumları yaşamaları azaltılacak.

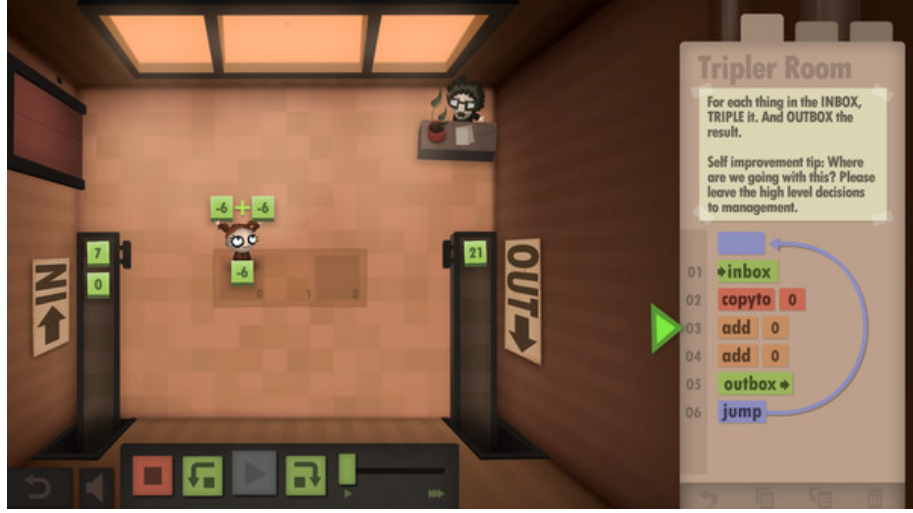
1.3 Benzer Projeler

Projenin oluşturulması sırasında mevcut olan projelerden ilham alındı. Bu projeler sırasıyla;

- Human resource machine
- Scratch
- MBlock

1.3.1 Human resource machine

Human resource machine oyunu içerisinde iki platform arasında verilerin kod blokları aracılığıyla taşınmasını oynanış olarak ele almaktadır. Giriş platformdan çıkış platformuna verilerin taşınması sırasında özel isteklerde bulunuluyor. Giriş platformundan alınan iki sayının toplanmasından oluşan verinin çıkış platformuna taşınması örnek verilebilir [2].



Şekil 1.1: Human resource machine

1.3.2 Scratch

Genel olarak çocukları hedef alan bir uygulamadır. Çocukların kendi animasyonlarını ve oyunlarını yapabilmeleri istenmiştir. Çocukların yaptığı çalışmalarını arkadaşlarına ve topluluklarına paylaşma imkanı sağlar.

1.3.3 MBlock

Scratch uygulamasının altyapısını kullanan bir uygulamadır. Scratch ile birebir özelliklere sahiptir. Scratch uygulamasının mobilde her cihazda çalışmaması gibi sorunları çözmektedir.

1.3.4 Projenin github linki

Proje kaynak kod olarak aşağıdaki github linkinde sunulmuştur.

<https://github.com/irfanderdiyok/surekle-birak-programlama>

2 KULLANILAN YAZILIMLAR ve YÖNTEMLER

Projenin detaylarına vakit ayıra bilinmesi için arayüz ve animasyon işlemlerini kolaylaştırması adına oyun motoru kullanıldı. Ayrıca yapılan projenin çapraz platformda çalışabilmesi istenmiştir. Bundan dolayı Unity oyun motoru kullanıldı. Diğer oyun motorlarının yerine tercih edilmesinin sebepleri;

- Ücretsiz olması
- Türkçe ve İngilizce kaynağa sahip olması
- Düşük sistem gereksinimi
- Çapraz platform çıktı verebilmesi
- Kendisine ait varlık (assets) mağazası olması
- Yardımcı paketlerin fazla olması

Alternatif olarak seçilebilecek oyun motorları;

- Unreal Engine
- Godot
- Gamemaker studio

Projenin herhangi bir sebepten ötürü hata alması, dosyaların silinmesi veya yapılan işlemlerin geri alınmak istendiği vakit projenin yedeğine ihtiyaç duyulur. Projenin yedeğini düzenli olarak alınması unutulabilir. Veya sadece bazı değişiklikleri geri alınması istendiğinde projenin yedeği olsa bile çok zahmetli bir sürecin içine girilir. Bu tarz durumlardan sakınmak ve işi kolaylaştırmak için yedekleme sistemleri kullanılması gerekmektedir. Bu projede yaygın olarak kullanılmasından dolayı github tercih edildi.

Alternatif yedekleme sistemleri;

- Bitbucket
- GitLab
- Google Cloud Source Repositories
- SourceForge
- Gitbucket

2.1 Unity Oyun Motoru

Unity, öncelikli olarak bilgisayarlar, konsollar ve mobil cihazlar için video oyunları ve simülasyonları geliştirmek için kullanılan ve Unity Technologies tarafından geliştirilen çapraz platform bir oyun motorudur. İlk kez yalnızca Apple'ın 2005'teki Worldwide Developers Conference'da OS X için ilan edildi, bu tarihten itibaren 27 platformu hedeflemek üzere genişletildi. Unity oyun motoru; film sektörü, otomotiv sektörü, mimari, mühendislik ve inşaat gibi video oyunları dışındaki farklı endüstriler tarafından da benimsenmiş ve kullanılmaktadır [2].

2.2 Github

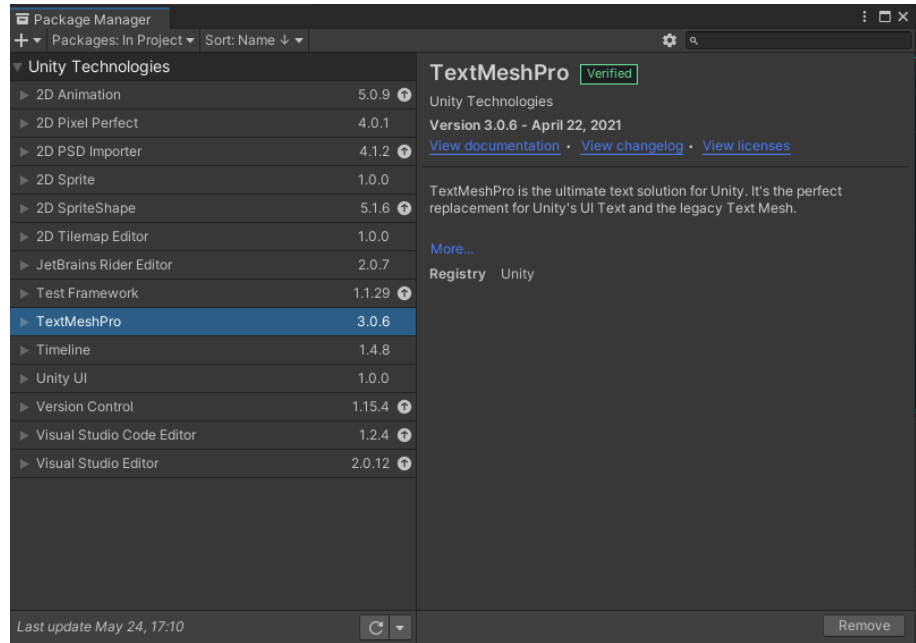
GitHub, sürüm kontrol sistemi olarak Git kullanan yazılım geliştirme projeleri için web tabanlı bir depolama servsidir. 8 Şubat 2008 tarihinde Tom Preston-Werner, Chris Wanstrath, P. J. Hyett, Scott Chacon tarafından kuruldu. GitHub özel depolar için ücretli üyelik seçenekleri sunarken, açık kaynaklı projeler için ücretsizdir. Mayıs 2011 itibarıyla GitHub açık kaynaklı projeler tarafından tercih edilen en popüler depolama servsidir [3].

3 SÜRÜKLE-BIRAK PROGRAMLAMA

Proje içerisinde unity oyun motorunun 2020.3.24f1 sürümü kullanıldı. Ayrıca unity başlangıç ekranında 2D şablonu seçilerek proje başlandı. Şablon olarak 2D seçilmemesi durumunda sprite paketi dahil edilmelidir. Projenin tamamı canvas objesinin içerisinde oluşturuldu.

3.1 Arayüz

Uygulama içerisinde bulunan menülerin, butonların ve panellerin farklı ekran çözünürlükleri uygun olacak şekilde boyutlandırılması gerekmektedir. Bu işlemi hem kolaylaştırmasıyla birlikte birkaç yeni özellik (glow, outline, neon) eklemesi için için **TextMesh Pro** paketi projeye eklendi. Ekleme işlemi **Window > Package Manager > TextMesh Pro** yolu izlenerek yapılır.



Şekil 3.1: TextMesh pro paketini projeye ekleme

Projenin arayüzünü unity içerisindeki canvas objesinin alt elemanı olarak oluşturuldu. Ayrıca yazılan kodların tamamı konuma bağlı olarak referans edildi. Bundan dolayı objelerin bulundukları konum çok önemlidir. Herhangi bir objenin konumu değiştirildiği takdirde yazılan kod hata verebilmektedir.

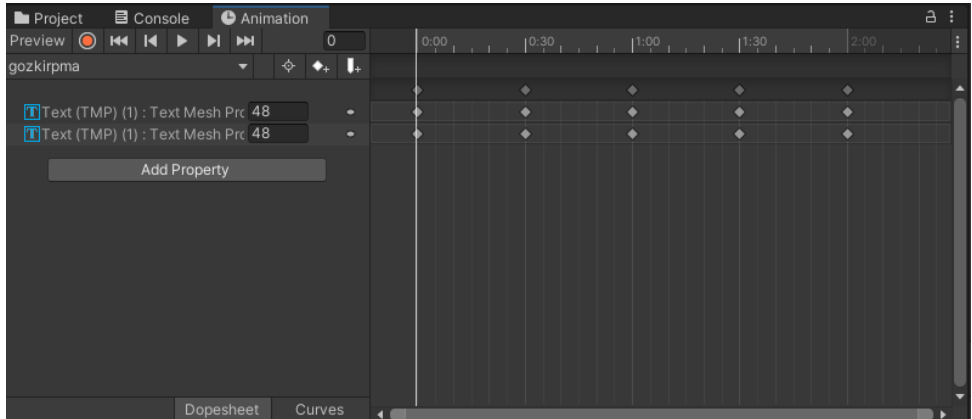
3.1.1 Giriş menüsü

Uygulama ilk açıldığında karşılaşılan penceredir. Uygulamanın ismi ve uygulamaya giriş yapmak için yapılması gereken işlemi kullanıcıya gösteren yardım metni bulunmaktadır. Yardım metninin belirginleşmesi için göz kırpma animasyonu yapıldı.



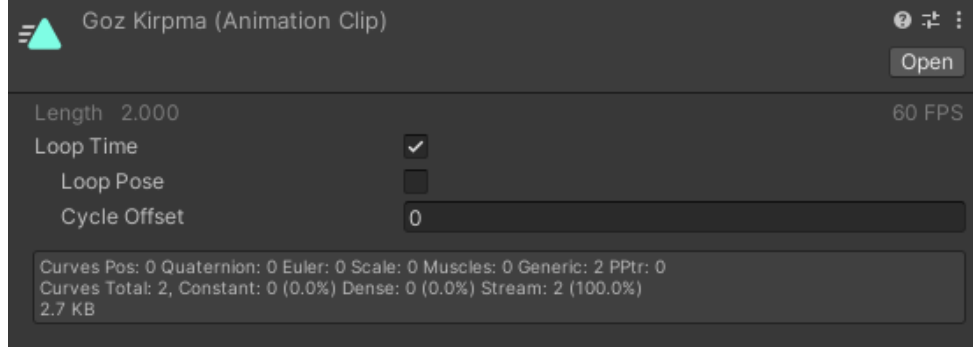
Şekil 3.2: Giriş menüsü

Animasyon oluşturmak için **Window > Animation > Animation** yolu takip edilmelidir. Açılan menüde metnin her saniye büyütülmesi gerekmektedir. Unity kendi içerisinde yapılan animasyonu eşit parçalara yayar. Bu şekilde daha yumuşak bir animasyon elde edilir.



Şekil 3.3: Animation menüsü ve animasyonun oluşturulması

Yapılan bu animasyonun **Loop Time** parametresi aktif hale getirilmedi. Bu sayede animasyon döngüye girerek göz kırpma efekti elde edilir.



Şekil 3.4: Animasyonu döngüye alma

Kullanıcı herhangi bir tuşa bastığı zaman sahneler arasında geçişi sağlayacak olan kod bloğu. SceneManagement kütüphanesinin eklenmesi gerekmektedir. Ayrıca Update fonksiyonun içerisine yazılmaz. Update fonksiyonu sayesinde koşul her karede kontrol edilecektir. Proje içerisinde update fonksiyonu olabildiğince az kullanıldı. Bunun sebebi update fonksiyonu özellikle mobil cihazlarda performans sorunu oluşturmaktadır.

```
1 using UnityEngine;
2 using UnityEngine.SceneManagement;
3
4 public class MenuController : MonoBehaviour
5 {
6     private void Update()
7     {
8         if (Input.anyKey)
9         {
10             SceneManager.LoadScene("AnaMenu");
11         }
12     }
13 }
```

Şekil 3.5: Sahneler arası geçiş kod bloğu.

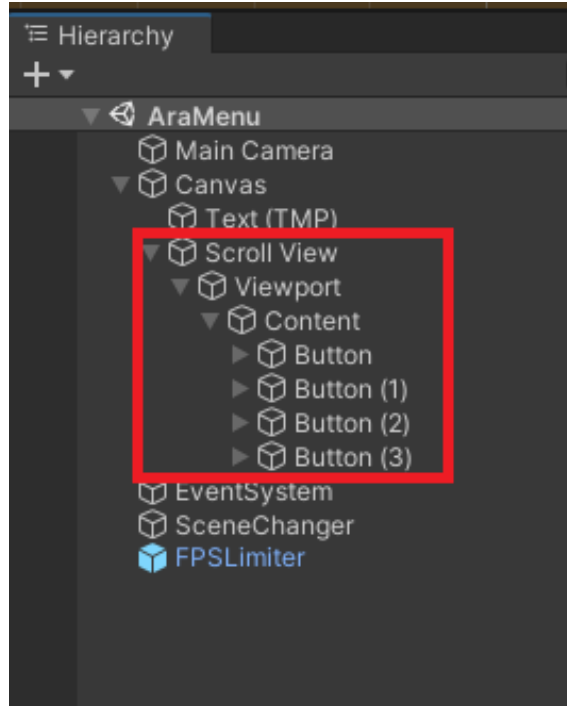
3.1.2 Ana menü

Kullanıcı herhangi bir tuşa bastığında ana menü yüklenir. Ana menü içerisinde yazılım temelleri ve oyuna geçişi sağlayan butonlar bulunur.



Şekil 3.6: Ana menü

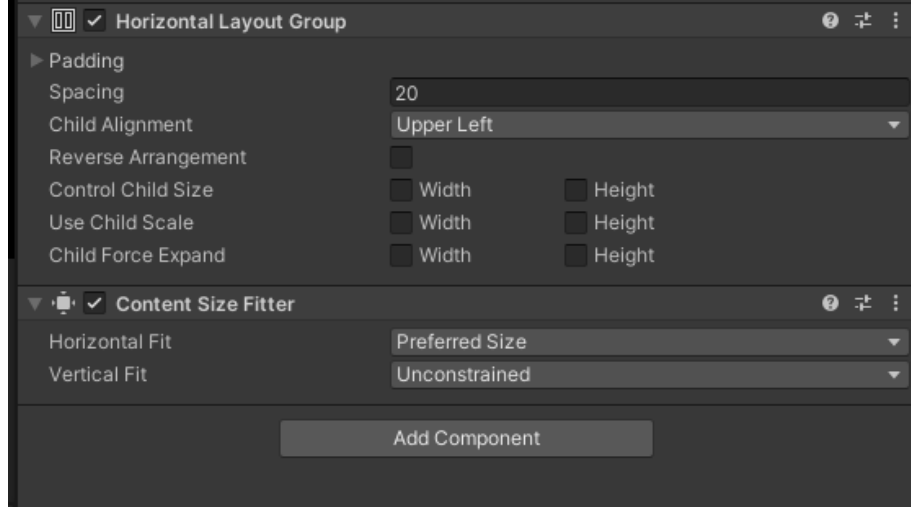
Menünün yatay düzlemde hareket edebilmesi için Unity'nin içerisinde bulunan Scroll View bileşeni kullanıldı. Hierarchy sekmesininde ana menünün yapısı aşağıdaki görseldeki gibi olmalıdır.



Şekil 3.7: Ana menü hierarchy yapısı

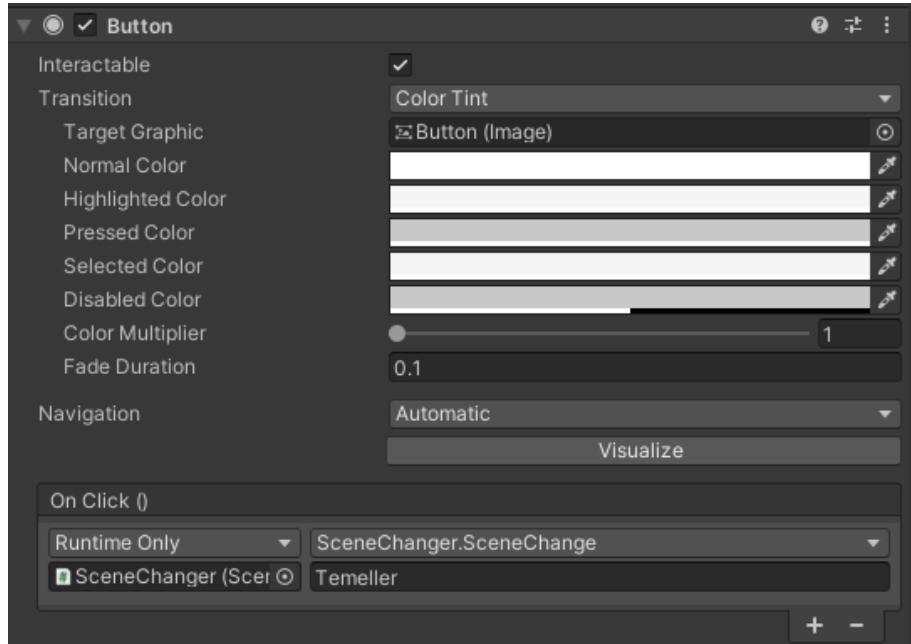
Ayrıca Content objesine Horizontal layout group ve Content size filter bileşenleri verilmesi gerekmektedir. Bu bileşenler sayesinde daha sonradan ana menüye yeni sahneler

eklendiği takdirde otomatik olarak sıralama yapar ve genişliği ayarlar.



Şekil 3.8: Content objesi bileşenleri

Sahneleri yüklemek için butonlara sahne değiştirme fonksiyon verildi. Fonksiyonun parametresini değiştirerek istenen sahneye geçiş sağlanabilir.



Şekil 3.9: Butona fonksiyon atama ve parametre değeri verme

3.1.3 Kod objeleri menüsü

Kod objelerinin tutulduğu menüdür. Kod objelerinin her bölümde tamamı açık olarak gelmemektedir. Bu sebepten dolayı kod objeleri menüden çıkarıldığı vakit unitynin menü elemanlarını sıralayabilmesi için **Vertical layout group** ve **Content size filter** bileşenlerine ihtiyaç duymaktadır.

3.1.4 Kod menüsü

Kod objeleri menüsünde bulunan objelerin, sürükle-bırak yapılarak oluşturulan algoritmanın tutulduğu menüdür. Kod objeleri menüsü ile aynı bileşenlere sahiptir.

3.2 Fonksiyonlar

Bu başlıkta uygulama içerisinde kullanıcın kod objeleriyle ifade ettiği algoritmayı koda çevrilmesi ve varsa hataları kullanıcıya bildirilmesi için gerekli olan fonksiyonlar verilmektedir.

3.2.1 Sürükleyip bıraklama

Kod objeleri menüsünde bulunan objelerin kod menüsüne sürüklenmesini ve uygun durumda içerisinde bulunan prefabs objesini aktifleştiren fonksiyodur. **UnityEngine.EventSystems** kütüphanesi eklenmelidir. Ayrıca bu kütüphanenin Interfacelerini implement edilmelidir. Bu fonksiyona sahip bir objeye tıklandığında pivot noktasını fare pozisyonuna sabitler. Üst objesinin içerisinde yukarı veya aşağı yönde hareket ettirilirse yer değişimi sağlar

```

1 public void OnDrag(PointerEventData eventData)
2 {
3     this.transform.position = eventData.position;
4     if (bosObjе.transform.parent != bosObjеParent)
5         bosObjе.transform.SetParent(bosObjеParent);
6
7     int newSiblingIndex = bosObjеParent.childCount;
8     for (int i = 0; i < bosObjеParent.childCount; i++)
9     {
10         if (this.transform.position.y > bosObjеParent.GetChild(i).
            position.y)
11         {
12             newSiblingIndex = i;
13             if (bosObjе.transform.GetSiblingIndex() <
                newSiblingIndex)
14                 newSiblingIndex--;
15             break;
16         }
17     }
18     bosObjе.transform.SetSiblingIndex(newSiblingIndex);
19 }
20
21 public void OnEndDrag(PointerEventData eventData)
22 {
23     if (prefabObjesi != null)
24     {
25         prefabObjesi = GameObject.Instantiate(prefabObjesi);
26         prefabObjesi.transform.SetParent(parentToReturnTo);
27         prefabObjesi.transform.SetSiblingIndex(bosObjе.transform.
            GetSiblingIndex());
28         Destroy(this.gameObject);
29     }
30     else
31     {
32         this.transform.SetParent(parentToReturnTo);
33         this.transform.SetSiblingIndex(bosObjе.transform.
            GetSiblingIndex());
34     }
35     GetComponent<CanvasGroup>().blocksRaycasts = true;
36     Destroy(bosObjе);
37 }

```

Şekil 3.10: Sürükleyip bırakma fonksiyonu.

Bu fonksiyon üst objeye verilir. Eğer içerisine herhangi bir obje girerse o objeyi kendi alt objesi olarak ayarlar. Ayrıca öncesinde alt elemanı olan obje başka bir üst objeye geçerse içerisindeki boş objeyi de yeni üst objeye taşır. Son olarak objenin sürüklenme işlemi tamamlanırsa objenin pozisyon hatasını gidererek sabitlenmesini sağlar.

```

1 public void OnPointerEnter(PointerEventData eventData)
2 {
3     if (eventData.pointerDrag == null)
4         return;
5
6     DragDrop d = eventData.pointerDrag.GetComponent<DragDrop>();
7     if (d != null)
8     {
9         d.bosObjjeParent = this.transform;
10    }
11 }
12 public void OnPointerExit(PointerEventData eventData)
13 {
14     if (eventData.pointerDrag == null)
15         return;
16
17     DragDrop d = eventData.pointerDrag.GetComponent<DragDrop>();
18     if (d != null && d.bosObjjeParent == this.transform)
19     {
20         d.bosObjjeParent = d.parentToReturnTo;
21     }
22 }
23 public void OnDrop(PointerEventData eventData)
24 {
25     DragDrop d = eventData.pointerDrag.GetComponent<DragDrop>();
26     if (d != null)
27     {
28         d.parentToReturnTo = this.transform;
29     }
30 }

```

Şekil 3.11: Objeyi sabitleme fonksiyonu.

3.2.2 Derleyici

Kullanıcın kod objelerini kullanarak oluşturduğu algoritmayı hierarchy penceresindeki üst ve alt (parent and child) objelerin sıralamasını referans olarak koda çevrilmesini sağlar.Çalıştır tuşuna basıldığı zaman objelerin konum referanslarından yola çıkarak objelerin hangi fonksiyona ait olduğunu tespit edip çalıştırmayı sağlar.

```

1 public void Derle(GameObject kod)
2 {
3     if (sifirla)
4     {
5         sonuc.text = "";
6     }
7     sifirla = false;
8     foreach (Transform kodObjesi in kod.transform)
9     {
10
11         if (kodObjesi.CompareTag("print"))
12         {
13             yazFonksiyon(kodObjesi.gameObject);
14         }
15         if (kodObjesi.CompareTag("degisken"))
16         {
17             degiskenFonksiyon(kodObjesi.gameObject);
18         }
19         if (kodObjesi.CompareTag("if"))
20         {
21             kosulFonksiyon(kodObjesi.gameObject);
22         }
23         if (kodObjesi.CompareTag("dongu"))
24         {
25             donguFonksiyon(kodObjesi.gameObject);
26         }
27     }
28
29 }

```

Şekil 3.12: Derleyici fonksiyonu.

3.2.3 Değişken oluşturma

Değişken oluşturulması ve saklanması için gerekli olan fonksiyondur. Kopya değişken oluşturulmasını engeller. Derleyiciden gelen veri gameobject parametresini kullanarak değişken oluşturmaya sağlar.

```

1 public class DegiskenOlusturma : MonoBehaviour
2 {
3     public string degiskenAdi;
4     public string degiskenDegeri;
5 }
6
7 public void degiskenFonksiyon(GameObject veri)
8 {
9     DegiskenOlusturma yeniDegisken = new DegiskenOlusturma();
10    yeniDegisken.degiskenAdi = gameObject.transform.GetChild(0).
        GetComponentInChildren<TMP_InputField>().text;
11    yeniDegisken.degiskenDegeri = gameObject.transform.GetChild
        (1).GetComponentInChildren<TMP_InputField>().text;
12    degiskenler.Add(yeniDegisken);
13 }

```

Şekil 3.13: Değişken sınıfı.



Şekil 3.14: Değişken oluşturma fonksiyonun hierarchy sekmesindeki yapısı

3.2.4 Koşul oluşturma

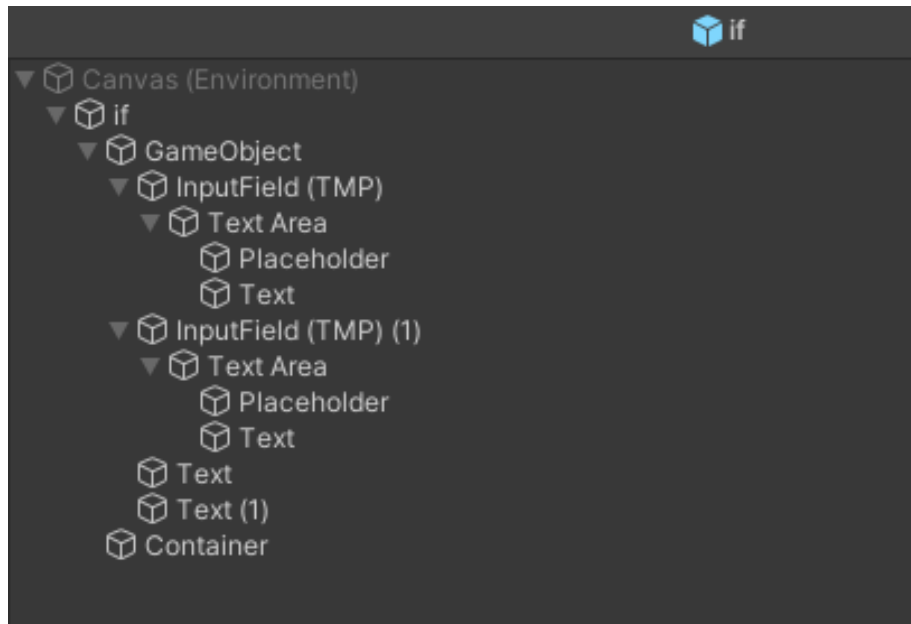
Derleyiciden gelen koşul gameobject parametresini kullanarak gerekli olan dönüşümler yapılır. Daha sonra koşul sağlanırsa tekrardan derleyiciden derlenir.


```

1 public void kosulFonksiyon(GameObject kosul)
2 {
3     string kosul1 = kosul.transform.GetChild(0).GetChild(0).
        GetComponentInChildren<TMP_InputField>().text;
4     string kosul2 = kosul.transform.GetChild(0).GetChild(1).
        GetComponentInChildren<TMP_InputField>().text;
5     int kosulSayi1;
6     int kosulSayi2;
7     try
8     {
9         kosulSayi1 = int.Parse(kosul1);
10        kosulSayi2 = int.Parse(kosul2);
11    }
12    catch (Exception e)
13    {
14        kosulSayi1 = 0;
15        kosulSayi2 = 0;
16    }
17    if (kosulSayi1 < kosulSayi2)
18    {
19        Derle(kosul.transform.GetChild(1).gameObject);
20    }
21 }

```

Şekil 3.15: Koşul fonksiyonu.



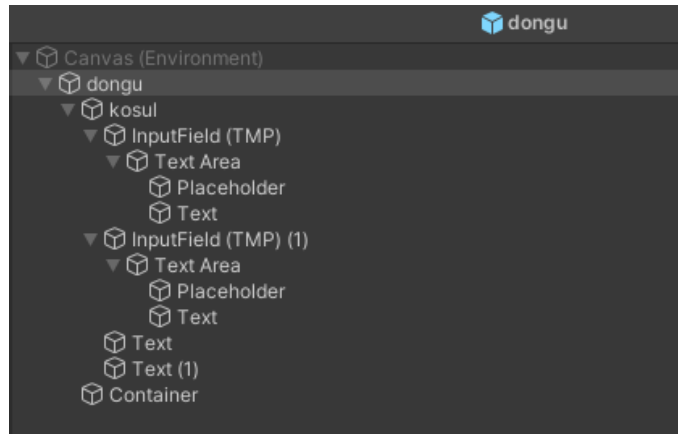
Şekil 3.16: Koşul fonksiyonun hierarchy sekmesindeki yapısı

3.2.5 Döngü oluşturma

Derleyiciden gelen koşul gameobject parametresini kullanarak gerekli olan dönüşümler yapılır. Daha sonra koşul sağlanırsa tekrardan derleyiciden derlenir.

```
1 public void donguFonksiyon(GameObject dongu)
2 {
3     string dongu1 = dongu.transform.GetChild(0).GetChild(0).
        GetComponentInChildren<TMP_InputField>().text;
4     string dongu2 = dongu.transform.GetChild(0).GetChild(1).
        GetComponentInChildren<TMP_InputField>().text;
5
6     int donguSayi1;
7     int donguSayi2;
8
9     try
10    {
11        donguSayi1 = int.Parse(dongu1);
12        donguSayi2 = int.Parse(dongu2);
13
14    }
15    catch (Exception e)
16    {
17        donguSayi1 = 0;
18        donguSayi2 = 0;
19
20    }
21
22    for (int i = 0; donguSayi1 < donguSayi2; donguSayi1++)
23    {
24        print(i);
25        Derle(dongu.transform.GetChild(1).gameObject);
26    }
27 }
```

Şekil 3.17: Döngü fonksiyonu.



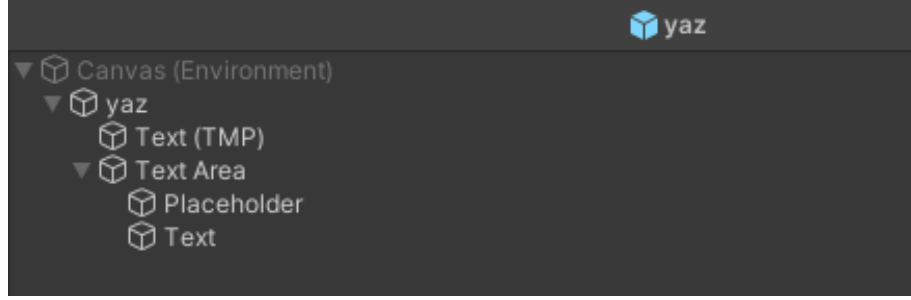
Şekil 3.18: Döngü fonksiyonun hierarchy sekmesindeki yapısı

3.2.6 Veri yazdırma

Uygulama içerisinde verilerin basılması için gerekli olan fonksiyondur. Derleyiciden gelen metin gameobject parametresini kullanarak sonuç değerine verileri basmayı sağlar.

```
1 public void yazFonksiyon(GameObject metin)
2 {
3     Transform temp = metin.transform.GetChild(1).GetChild(0);
4     if (temp.CompareTag("textkutusu"))
5     {
6         sonuc.text = sonuc.text + temp.GetComponent<TMP_InputField>
7             >().text + " ";
8     }
9     else
10    {
11        String tempVar = temp.GetChild(0).GetComponent<
12            TextMeshProUGUI>().text;
13        foreach (DegiskenOlusturma degisken in degiskenler)
14        {
15            if (tempVar == degisken.degiskenAdi)
16            {
17                sonuc.text = sonuc.text + degisken.degiskenDegeri;
18            }
19        }
20        int elemanSayisi = metin.transform.GetChild(1).childCount;
21        for (int i = 1; i < elemanSayisi; i++)
22        {
23            Transform temp2 = metin.transform.GetChild(1).GetChild(i).
24                GetChild(0);
25            if (temp2.CompareTag("textkutusu"))
26            {
27                sonuc.text = sonuc.text + temp2.GetComponent<
28                    TMP_InputField>().text + " ";
29            }
30            else
31            {
32                String tempVar = temp2.GetChild(0).GetComponent<
33                    TextMeshProUGUI>().text;
34                foreach (DegiskenOlusturma degisken in degiskenler)
35                {
36                    if (tempVar == degisken.degiskenAdi)
37                    {
38                        sonuc.text = sonuc.text + degisken.
39                            degiskenDegeri;
40                    }
41                }
42            }
43        }
44    }
45 }
```

Şekil 3.19: Veri yazdırma fonksiyonu.



Şekil 3.20: Yaz fonksiyonun hierarchy sekmesindeki yapısı

3.2.7 Matematiksel işlemler

Kullanıcın dört temel matematiksel işlemi yapmasını sağlayan kod blokları oluşturuldu. Ayrıca mod alma ve iki sayının en büyüğünü bulma gibi yardımcı işlemlerde eklendi.



Şekil 3.21: Matematik fonksiyonun hierarchy sekmesindeki yapısı

Sadece sayı olan değişkenler üzerinde işlem yapılması için gerekli olan kontroller yapıldı. Sayı olan değişkenler seçilen işarete göre matematiksel işlem yapıldıktan sonra değişkene atama yapılır.

```

1  public void matematikFonksiyon(GameObject matematik)
2  {
3      foreach (DegiskenOlusturma degisken in degiskenler)
4      {
5          if (degisken1 == degisken.degiskenAdi)
6          {
7              if (degisken.isString)
8              {
9                  matematik.transform.GetChild(0).GetChild(0).GetChild
10                     (0).GetComponent<TextMeshProUGUI>().text = "";
11                  matematik.transform.GetChild(0).GetChild(2).GetChild
12                     (0).GetComponent<TextMeshProUGUI>().text = "";
13                  matematik.transform.GetChild(0).GetChild(4).GetChild
14                     (0).GetComponent<TextMeshProUGUI>().text = "";
15                  return;
16              }
17              else
18              {
19                  deger1 = float.Parse(degisken.degiskenDegeri);
20              }
21      }
22  }

```

Şekil 3.22: Matematik fonksiyonu.

3.2.8 Soru oluşturma ve kontrol etme

Kullanıcıya soru sormak ve sorulan soruların cevapları kontrol etmek için gerekli olan fonksiyondur. Kullanıcın kod menüsünde oluşturduğu yapı kontrol edilerek istenen cevaba ve algoritma sırasına sahip olup olmadığı kontrol edilir.

```

1 public void OrunutIncele(Transform kod)
2 {
3     foreach (Transform kodObjesi in kod)
4     {
5         if (kodObjesi.CompareTag("print"))
6         {
7             oruntu += "p";
8         }
9         if (kodObjesi.CompareTag("degisken"))
10        {
11            oruntu += "v";
12        }
13        if (kodObjesi.CompareTag("if"))
14        {
15            oruntu += "i<";
16            OrunutIncele(kodObjesi.GetChild(1));
17            oruntu+=">";
18        }
19        if (kodObjesi.CompareTag("dongu"))
20        {
21            oruntu += "l<";
22            OrunutIncele(kodObjesi.GetChild(1));
23            oruntu+=">";
24        }
25        if (kodObjesi.CompareTag("matematik"))
26        {
27            oruntu += "m";
28        }
29    }
30 }
31
32 if (sonuc.text == GorevIcerigi && oruntu ==istenenOruntu)
33 {
34
35     gorev.Zafer();
36 }
37
38
39 }

```

Şekil 3.23: Soru oluşturma ve kontrol fonksiyonu.

3.2.9 FPS sınırlandırma

Uygulamanın saniyede altmış kez yenilenmesi için gerekli olan kod parçasıdır. Uygulamanın gereksiz işlem gücü harcamasını engellemek için kullanılır.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class FPSLimiter : MonoBehaviour
6 {
7     private void Awake ()
8     {
9         Application.targetFrameRate = 60;
10    }
11 }
```

Şekil 3.24: Fps sınırlandırma kod bloğu.

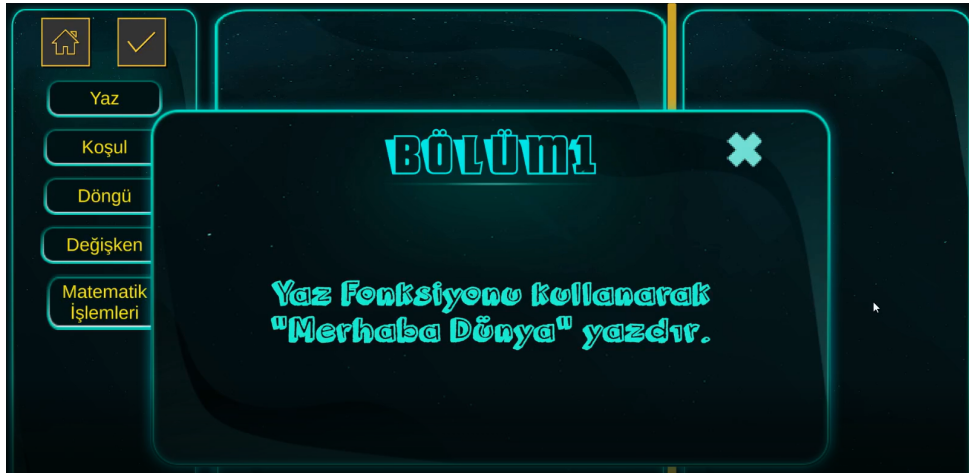
4 SONUÇLAR VE ÖNERİLER

Projenin temel amacı olan kod objeleri kullanarak algoritma yazma sistemi tamamlandı. Uygulamanın temel haliyle öğrenciler **sözdizimi (syntax)** hatalarıyla uğraşmadan algoritma yazmaları ve yazılımın temel yapılarını öğrenmeleri yardımcı olacaktır. Bunun yanı sıra, algoritma yazma sisteminde değişikliklere gidilerek kod objelerini kullanarak ilerlenen bir oyun yapıldı. Uygulamanın temel halinden sıkılan öğrencilerin ilgisini çekebilir. Her iki kısım için fonksiyonların mantığını anlamaları için sorular eklendi. Projenin fikir aşamasında olduğu sırada eklenmesi düşünülen uzaktan soru ekleme sistemi eklenemedi. Bu sistem projeye ileri bir vakit eklenmesi öğretmenlere fayda sağlayacaktır. Uygulamanın ara yüzünü fütüristik olarak dizayn edilmesi düşünüldü. Ancak kullanılan görsellerin ve animasyonların yeterli olmamasından dolayı tasarımda tam olarak başarılı olunamadı.

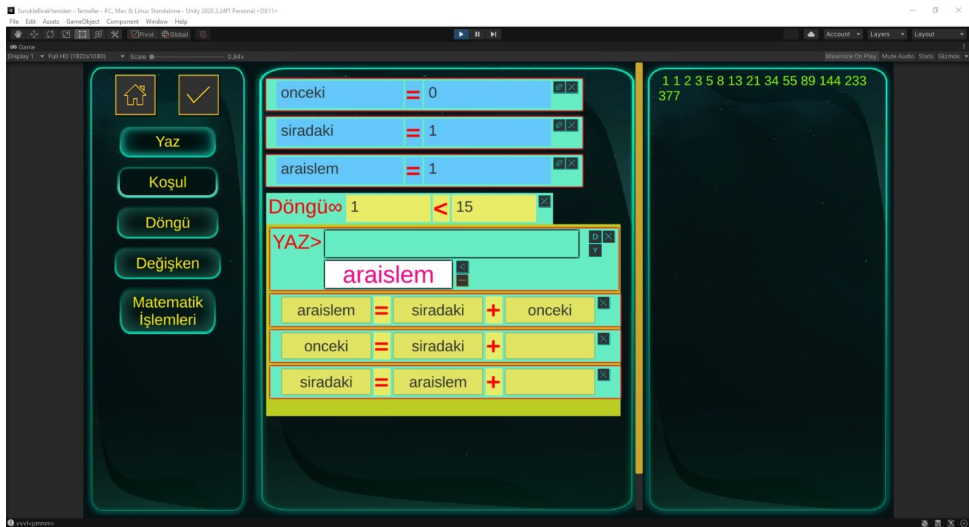
5 EKLER

<https://github.com/irfanderdiyok/SurukleBirakYeniden>

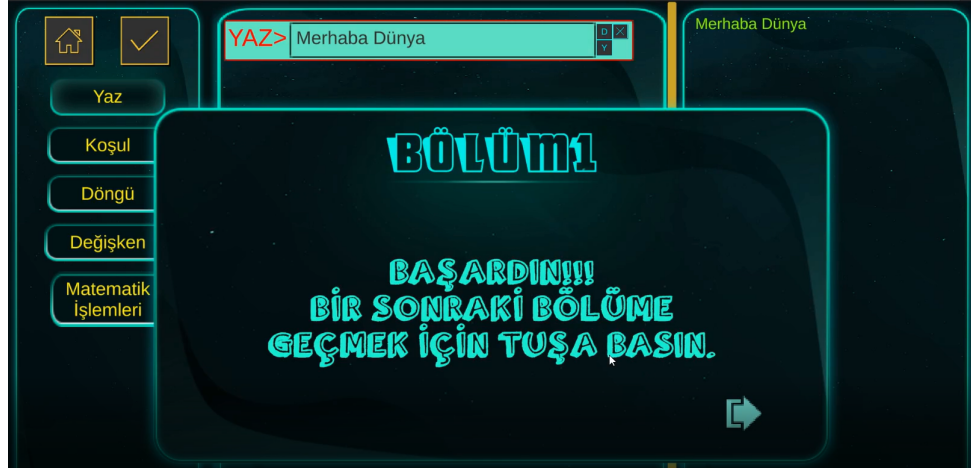
5.1 Uygulama İçerisinden Bazı Ekran Görüntüleri



Şekil 5.1: Örnek soru ekran görüntüsü



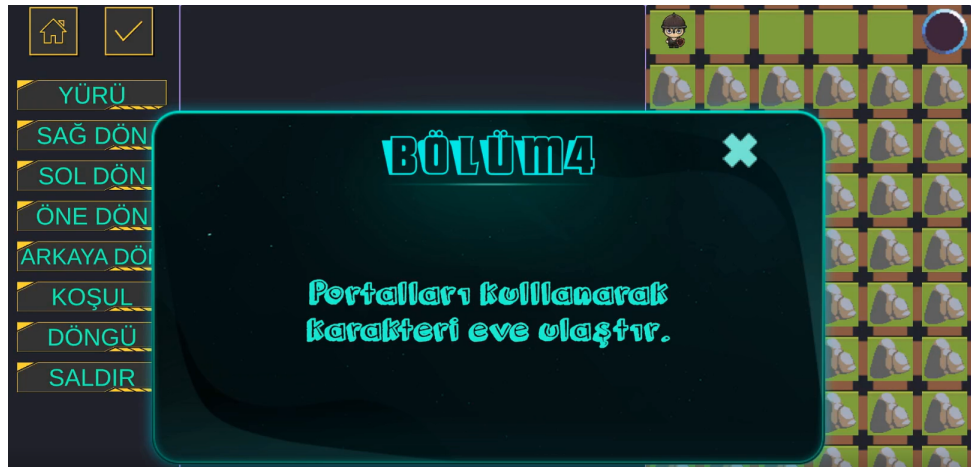
Şekil 5.2: Fibonacci dizisi bulma



Şekil 5.3: Doğru cevap ekran görüntüsü



Şekil 5.4: Oyun oynanış ekran görüntüsü



Şekil 5.5: Oyun görev ekranı görüntüsü

KAYNAKLAR

- [1] https://store.steampowered.com/app/375820/Human_Resource_Machine/ [Ziyaret Tarihi: 30 Mayıs 2022]
- [2] [https://tr.wikipedia.org/wiki/Unity_\(oyun_motoru\)](https://tr.wikipedia.org/wiki/Unity_(oyun_motoru)) [Ziyaret Tarihi: 6 Nisan 2022]
- [3] <https://tr.wikipedia.org/wiki/GitHub> [Ziyaret Tarihi: 6 Nisan 2022]
- [4] <https://sungraphica.itch.io/sci-fi-game-ui-collection-free-version> [Ziyaret Tarihi: 6 Nisan 2022] [Ziyaret Tarihi: 6 Nisan 2022]

ÖZGEÇMİŞ

KİŞİSEL BELGELER

Adı Soyadı : İrfan DERDİYOK

Uyruğu : T.C.

Doğum Yeri ve Tarihi: Erzurum 09/01/1999

Adres : İstanbul

Telefon : +905

E-mail : irfanderdiyok@hotmail.com

EĞİTİM DURUMU

Lisans Öğrenimi : BŞEÜ Bilgisayar Mühendisliği Bölümü

Bitirme Yılı : Devam Ediyor

Lise : Vefa Poyraz Anadolu Lisesi / İstanbul

İŞ DENEYİMLERİ

Yıl : 2020-Devam ediyor

Kurum : Bionluk Freelancer

Stajlar : Yapılmadı

İLGİ ALANLARI:

Dijital Oyunlar

YABANCI DİLLER:

İngilizce

BELİRTMEK İSTEDİĞİNİZ DİĞER ÖZELLİKLER: