

**T.C.**  
**TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**

**STAJ RAPORU**

**MATEMATİK ÖĞRETMENİM**

**HARUN ESAD AŞIK**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMAN: Doç. Dr. ERDİNÇ UZUN**

**2020**

**1.**

# 1. İÇİNDEKİLER

1. İçindekiler .....	2
2. RAPOR KONUSU .....	3
2.1. RAPOR KONUM .....	3
2.2. BENZER KONUDAKİ PROJELER.....	3
2.2.1. Uygulama İsmi: Matematik Zeka Oyunları .....	3
2.2.2. Uygulama İsmi: Matematiksel oyunlar .....	3
3. PROJEMİN ÇÖZDÜĞÜ SORUNLAR .....	4
4. KODLAMA .....	4
5. PROJEMDE KARŞILAŞTIĞIM SORUNLAR VE ÇÖZÜMLERİ .....	5
5.1. SORUN -1 .....	5
5.2. ÇÖZÜM -1.....	5
5.3. SORUN -2 .....	5
5.4. ÇÖZÜM -2.....	5
6. PROJEMİN MEVCUT EKSİKLİKLERİ .....	6
7. PROJE TASARIMI .....	7
8. KAYNAK KODLARI VE İŞLEVLERİ .....	8
9. KULLANIM KILAVUZU .....	18
10. KAYNAKÇA .....	23
11. ÖZGEÇMİŞ.....	24

## **2. RAPOR KONUSU**

### **2.1. RAPOR KONUM**

Öğrenciler için toplama, çıkarma, çarpma ve bölme pratikleri yapabileceği, aynı zamanda öğrencilerin dört işlemi daha hızlı bir şekilde yapabileceği bir mobil uygulama. Bu mobil uygulamanın kullanımının kolay olması amaçlanmıştır. Bu mobil uygulama hem ilkokul hem lise hem üniversite öğrencilerine uygundur.

Bu mobil uygulamada öğrencilerin kendi derecelerini öğrenebilmeleri için oyun sonunda alınan puana göre bir derecelendirme yapılmıştır. Öğrenciler kendilerini geliştirmek için aldığı dereceyi geliştirmeye çalışmalıdır. Bu derecelendirmede öğrencilerin azmini artırmak amaçlanmıştır.

### **2.2. BENZER KONUDAKİ PROJELER**

#### **2.2.1. Uygulama İsmi: Matematik Zeka Oyunları**

Bu uygulama ile sayısal işlem becerilerinizi geliştirebilirsiniz. Uygulamanın amacı matematik soruları çözmek ve matematiksel zekayı geliştirmektir. Uygulamadaki matematik testleri ile pratik yaparak matematiksel düşünme ve problem çözme yeteneğinizi ilerletebilirsiniz. Bu uygulama matematik ustası olmak isteyenler için tasarlandı. Zor matematik oyunu isteyenlere göre tasarlanmıştır.

#### **2.2.2. Uygulama İsmi: Matematiksel oyunlar**

Bu uygulama en küçüklerin bile oynayabileceği çok basit ve kolaydır. Beyninizi eğitmek için tasarlanan bu uygulama çocuklar, yetişkinler, ebeveynler, büyükanne ve büyükbaba dahil olmak üzere her yaş için tasarlanmıştır. Renkli çalışma sayfalarıyla toplama, çıkarma, çarpma ve bölme işlemlerinin basit olmasıyla kolaylıkla oynanabilecek eğlenceli bir oyun.

## **3.**

### **3. PROJEMİN ÇÖZDÜĞÜ SORUNLAR**

Yaptığım uygulama matematikle ilgili problemi olan öğrenciler için bire birdir. Bu uygulama matematik sınavlarında zamanı yetiştirmeyen öğrencilerin matematik sınavlarında daha hızlı bir şekilde dört işlemi yapıp bu sınavlarda zamanı yetiştirme yolunda ilerleme kat etmesini sağlar.

Bu uygulama öğrencilerin daha pratik olmasını sağlar, daha analitik düşüncelerini sağlar. Öğrencilerin karşılaştığı problemleri daha kolay çözebilmesini sağlar.

Bu uygulama öğrencilerin sürekli yapacağı işlemleri daha kolay yapmalarını sağlar. Öğrenciler süreyle olan sorunlarını bu uygulama ile daha kolay aşabilecekler.

Matematiksel işlemlerin genel olarak hayatta daha kolay yapılabilmesine olanak sağlar. Hayatta daha hızlı matematiksel işlemlerin yapılabilmesini sağlar.

### **4. KODLAMA**

Uygulamamı hayata geçirmek için c# programlama dilini kullandım ve yaptığım uygulamayı unity oyun motorunda tasarlayıp visual studio da c# ile kodladım.

## **5. PROJEMDE KARŞILAŞTIĞIM SORUNLAR VE ÇÖZÜMLERİ**

### **5.1. SORUN -1**

Ben uygulamamda en başta json editör kullanarak yapmaya çalıştım. Ancak bir sorunla karşılaştım. Yaptığım uygulamada json editörü aktif hâle getirmek için yazdığım kodlarla json editörü aktif hâle getiremedim. Bu yüzden uygulamayı yapmak için farklı bir yöntem bulmaya çalıştım.

### **5.2. ÇÖZÜM -1**

Uygulamamda unity oyun motorunu kullandım. Unity oyun motorunda c# kodlama dilini kullanarak uygulamayı yaptım. Public class kullanarak dışarıdan (unity) kendim belirledim soruların ne olacağını. Böylelikle soruları, cevapları, hangi cevabın doğru olduğunu kendim yazdım. Bütün uygulamanın tasarımını unity ile planladım.

### **5.3. SORUN -2**

Uygulamamda ilk başta puanlamayı soru adedine göre ayarladım. Yani belli bir soru adedi var ve bu sorular bittiğinde oyun da bitiyor.

### **5.4. ÇÖZÜM -2**

Çözüm olarak oyunun puanlamasını süreye bağladım. Yani soruların belli bir sayısı yok, soruları cevapladıkça devamı gelecek şekilde düzenledim. Belli bir süre bittiğinde kaç soru doğru cevaplanmışsa ona göre puan alınıyor.

## 6. PROJEMİN MEVCUT EKSİKLİKLERİ

Uygulamamda tasarımsal olarak eksikleri anlaticam. Uygulamamda soruların yanıtlanmasındaki doğrulama kısmı eksik. Yapmak istediğim doğru cevap verildiğinde cevap verilen şık yeşil renge dönüşecek, yanlış cevap verildiğinde cevap verilen şık kırmızı renge dönüşecek ve doğru cevap yeşil renge dönüşecek.

Uygulamamda soruları kendim belirledim ve bütün sorular rastgele bir şekilde soruluyor. Benim yapmaya çalıştığım json editör kullanarak soruları kolaydan zora doğru ayarlamaktı. Ama json editör ile programlayamadığım için sorular rastgele kolay zor fark etmeden geliyor.

Uygulamamda alınan puanlar puanı aldıktan sonra gelen sayfada gözükyor. Aslında yapmak istediğim puanları hem puanı aldıktan sonraki sayfada hem de başka bir sayfada alınan en iyi 5 puan olarak kaydetmekti.

Uygulamamda yaptığım bölümler toplama, çıkarma, çarpma, bölme. Yaptığım bölümleri gelecekte çeşitlendirebilirim. Mesela üslü sayılar, köklü sayılar, denklemler.

Uygulamamda gelecekte bir işlemin dışına çıkabilirim. Demek istediğim şu an yaptığım şekliyle sadece bir işlem (toplama vb.) dışında iki işlem (aynı zamanda toplama çarpma vb.) şeklinde veya üç işlem şeklinde yapmak.

## 7. PROJE TASARIMI

Öncelikle yaptığım uygulamanın ana menüsünden bahsetmek istiyorum:

Ana menüde bölümlere girebilmek için 4 buton tasarladım: “TOPLAMA”, “ÇIKARMA”, “ÇARPMA”, “BÖLME”. Ana menüde oyundan çıkabilmek için 1 buton tasarladım: “ÇIKIŞ”

Ana menüdeki bölümlerden toplama bölümünde soru butonu, 4 tane cevap butonu, 45 saniyeden 0 saniyeye doğru birer birer azalan bir süre texti, doğru cevap verildiğinde 10 puan artan, yanlış cevap verildiğinde 10 puan azalan ama asla 0 puanın altına düşmeyen bir puan texti tasarladım.

Ana menüdeki bölümlerden çıkarma bölümünde soru butonu, 4 tane cevap butonu, 45 saniyeden 0 saniyeye doğru birer birer azalan bir süre texti, doğru cevap verildiğinde 10 puan artan, yanlış cevap verildiğinde 10 puan azalan ama asla 0 puanın altına düşmeyen bir puan texti tasarladım.

Ana menüdeki bölümlerden çarpma bölümünde soru butonu, 4 tane cevap butonu, 45 saniyeden 0 saniyeye doğru birer birer azalan bir süre texti, doğru cevap verildiğinde 10 puan artan, yanlış cevap verildiğinde 10 puan azalan ama asla 0 puanın altına düşmeyen bir puan texti tasarladım.

Ana menüdeki bölümlerden bölme bölümünde soru butonu, 4 tane cevap butonu, 45 saniyeden 0 saniyeye doğru birer birer azalan bir süre texti, doğru cevap verildiğinde 10 puan artan, yanlış cevap verildiğinde 10 puan azalan ama asla 0 puanın altına düşmeyen bir puan texti tasarladım.

Toplama, çıkarma, çarpma, bölme bölümlerinden sonra “SONUÇ” ekranı tasarladım. Bu sonuç ekranında “OYUN SONU” yazan bir text, alınan puanı yazan bir text, alınan dereceyi gösteren bir text ve ana menüye geri dönebilmemizi sağlayan “YENİDEN OYNA” butonunu tasarladım.

## 8. KAYNAK KODLARI VE İŞLEVLERİ

```
using System.Collections;
//C# kütüphane (nesne koleksiyonlarını tanımlayan arabirimler ve sınıflar içerir).
using System.Collections.Generic;
//C# kütüphane (genel koleksiyonları tanımlayan arabirimler ve sınıflar içerir).
using UnityEngine; //C# kütüphane (unity aktif hâle gelir).
using UnityEngine.SceneManagement;
//C# kütüphane (unityde sahnelerdeki işlemler için).
public class Toplama : MonoBehaviour // Dışarıdan ulaşılabilecek sınıf tanımlama.
{
    public void toplama() //Dışarıdan ulaşılabilecek bir değişken tanımlama.
    {
        SceneManager.LoadScene("Toplama"); //Toplama sahnesine yönlendirme
    }
    public void cikarma() //Dışarıdan ulaşılabilecek bir değişken tanımlama.
    {
        SceneManager.LoadScene("Çıkarma"); //Çıkarma sahnesine yönlendirme
    }
    public void carpma() //Dışarıdan ulaşılabilecek bir değişken tanımlama.
    {
        SceneManager.LoadScene("Çarpma"); //Çarpma sahnesine yönlendirme
    }
    public void bolme() //Dışarıdan ulaşılabilecek bir değişken tanımlama.
    {
        SceneManager.LoadScene("Bölme"); //Bölme sahnesine yönlendirme
    }
    public void cikis() //Dışarıdan ulaşılabilecek bir değişken tanımlama.
    {
        Application.Quit(); //Uygulamadan çıkılır.
    }
    public void yeniden() //Dışarıdan ulaşılabilecek bir değişken tanımlama.
    {
        SceneManager.LoadScene("Anamenu"); //Anamenu sahnesine yönlendirme.
    }
}
```



```

using System.Collections;

//C# kütüphane (nesne koleksiyonlarını tanımlayan arabirimler ve sınıflar içerir).

using System.Collections.Generic;

//C# kütüphane (genel koleksiyonları tanımlayan arabirimler ve sınıflar içerir).

using UnityEngine;                                //C# Kütüphane (unity aktif hâle gelir).

[System.Serializable]                             //Serileştirme.

public class soru                                //Dışarıdan ulaşılabilecek bir sınıf tanımlama.
{

    public string soruadı, cevapa, cevapb, cevapc, cevapd;

    //Dışarıdan ulaşılabilecek veri tanımlama.

    public int cevap;                            //Dışarıdan ulaşılabilecek bir veri tanımlama.

    public soru(string soru,string cevap1, string cevap2, string cevap3, string cevap4,int
    answer)                                     //Dışarıdan ulaşılabilecek bir değişken tanımlama.

    {

        soruadı = soru;                        //Başka bir sınıfta kullanabilmek için fonksiyon tanımlama.

        cevapa = cevap1;                      //Başka bir sınıfta kullanabilmek için fonksiyon tanımlama.

        cevapb = cevap2;                      //Başka bir sınıfta kullanabilmek için fonksiyon tanımlama.

        cevapc = cevap3;                      //Başka bir sınıfta kullanabilmek için fonksiyon tanımlama.

        cevapd = cevap4;                      //Başka bir sınıfta kullanabilmek için fonksiyon tanımlama.

        cevap = answer;                       //Başka bir sınıfta kullanabilmek için fonksiyon tanımlama.

    }

}

```

```

using System.Collections;

//C# kütüphane (nesne koleksiyonlarını tanımlayan arabirimler ve sınıflar içerir).

using System.Collections.Generic;

//C# kütüphane (genel koleksiyonları tanımlayan arabirimler ve sınıflar içerir).

using UnityEngine;                                //C# kütüphane (unity aktif hâle gelir).

public class SOR : MonoBehaviour                    // Dışarıdan ulaşılabilecek sınıf tanımlama.
{
    public List<SOR> sorular;                      //Başka bir sınıftan veri çekip listeleştirme.
}

```

```

using System.Collections;

//C# kütüphane (nesne koleksiyonlarını tanımlayan arabirimler ve sınıflar içerir).

using System.Collections.Generic;

//C# kütüphane (genel koleksiyonları tanımlayan arabirimler ve sınıflar içerir).

using System.Threading;

//C# kütüphane (Bir iş parçacığı oluşturup denetler, önceliğini ayarlar).

using UnityEngine;                                //C# kütüphane (unity aktif hâle gelir).

using UnityEngine.UI;                             //C# kütüphane (unity arayüzünü tasarlamamızı sağlar).

```

```
using UnityEngine.SceneManagement;
```

```
//C# kütüphane (unityde sahnelerdeki işlemler için).
```

```
public class oyun : MonoBehaviour //Dışarıdan ulaşılabilir sınıf tanımlama.
```

```
{
```

```
public oyun o; //Bir sınıftan değişken çağırmak için yapılan işlem(dışarıdan ulaşılır).
```

```
public Text soruadı, cevapa, cevapb, cevapc, cevapd, skor, zaman, sonuc;
```

```
//Dışarıdan ulaşılabilir veri tanımlama.
```

```
SOR s; //Bir sınıftan değişken çağırmak için yapılan işlem.
```

```
public List<bool> sorulan; //Kontrollü bir şekilde dışarıdan ulaşılabilir liste tanımlama.
```

```
public int cevap, puan; //Dışarıdan ulaşılabilir veri tanımlama.
```

```
public float sure; //Dışarıdan ulaşılabilir veri tanımlama.
```

```
void Start() //Başlangıçta yapılan işlemler.
```

```
{
```

```
sure = 45; //Veri tanımlama.
```

```
s = GetComponent<SOR>(); //Hedef sınıfa ulaşmak.
```

```
for (int i = 0; i < s.sorular.Count; i++) //Döngü tanımlama.
```

```
{
```

```
sorulan.Add(false); //Sorulan listesine soru eklememek.
```

```
}
```

```
soruekleme(); //Değişken çağırma.
```

```
}
```

```
void Update() //Tekrarlanan işlemler.
```

```
{
```

```
if (sure > 0) //Eğer sure 0'dan büyükse.
```

```

{
sure -= Time.deltaTime; //Surenin birer birer azalması.
zaman.text = sure.ToString("00"); //Surenin zaman olarak bir text tanımı.
}
Else //Değilse.
{
SceneManager.LoadScene("SONUC"); //SONUC sahnesine yönlendirme.
o.oyunsonu(); //Sınıf içinden değişken çağırma.
}
}

public void soruekleme() //Dışarıdan ulaşılabilirlik değişken
tanımlama.
{
for (int i = 0; i < sorulan.Count; i++) //Döngü oluşturma.
{
if (sorulan[i] == false) //Eğer sorulan listesinde sayı yoksa.
{
int sayi = Random.Range(0, sorulan.Count);
//Sorulan listesindeki sayıyı rastgele belirleme.
if (sorulan[sayi] == false) //Eğer rastgele sayı gelmiyorsa.
{
sorulan[sayi] = true; //Rastgele sayı getir.
soruadi.text = s.sorular[sayi].soruadi; //Soruadi' nin sorular listesinde bir text tanımı.
cevapa.text = s.sorular[sayi].cevapa; //Cevapa' nin sorular listesinde bir text tanımı.
cevapb.text = s.sorular[sayi].cevapb; //Cevapb' nin sorular listesinde bir text tanımı.
}
}
}
}

```

```

cevapc.text = s.sorular[sayi].cevapc; //Cevapc' nin sorular listesinde bir text tanımı.
cevapd.text = s.sorular[sayi].cevapd; //Cevapd' nin sorular listesinde bir text tanımı.
cevap = s.sorular[sayi].cevap; //Cevap' ın sorular listesindeki tanımı.
}
Else //Değilse.
{
soruekleme(); //Değişken çağırma.
}
break; //Döngüyü kırma.
}
if (i == sorulan.Count - 1) //Eğer sorulan listesinin sayısının bir eksiğine eşitse.
{
SceneManager.LoadScene("SONUC"); //SONUC sahnesine yönlendirme.
o.oyunsonu(); //Sınıf içinden değişken çağırma.
}
}
}
public void cevaplar(int rakam) //Dışarıdan ulaşılacak değişken tanımlama.
{
if (rakam == cevap) //Eğer iki veri birbirine eşitse.
{
puan += 10; //Puan 10' ar artar.
skor.text = "PUAN:" + puan; //Skor texti tanımlanır.
soruekleme(); //Değişken çağırma.
sonuc.text = "DOĞRU CEVAP"; //Sonuc textinde yazacak olan.
}
}

```

```

}
Else //Değilse.
{
if (puan > 0) //Eğer puan 0' dan büyükse.
{
puan -= 10; //Puan 10' ar eksilir.
skor.text = "PUAN:" + puan; //Skor texti bir ifadeye eşitlenir.
soruekleme(); //Değişken çağırma.
sonuc.text = "YANLIŞ CEVAP"; //Sonuc textinde yazacak olan.
}
Else //Değilse.
{
soruekleme(); //Değişken çağırma.
sonuc.text = "YANLIŞ CEVAP"; //Sonuc textinde yazacak olan.
}
}
}
public void oyunsonu() //Dışarıdan ulaşılabilecek değişken tanımlama.
{
PlayerPrefs.SetInt("Puan:", puan); //Puan kayıt etme.
}
}

```

```

using System.Collections;

//C# kütüphane (nesne koleksiyonlarını tanımlayan arabirimler ve sınıflar içerir).

using System.Collections.Generic;

//C# kütüphane (genel koleksiyonları tanımlayan arabirimler ve sınıflar içerir).

using UnityEngine;                                //C# kütüphane (unity aktif hâle gelir).

using UnityEngine.UI;                             //C# kütüphane (unity arayüzünü tasarlamamızı sağlar).

using UnityEngine.SceneManagement;

//C# kütüphane (unityde sahnelerdeki işlemler için).

public class SON : MonoBehaviour                    //Dışarıdan ulaşılabilecek sınıf tanımlama.
{

public oyun o;      //Bir sınıftan değişken çağırmak için yapılan işlem(dışarıdan ulaşılır).

public Text skor,mesaj;                            //Dışarıdan ulaşılabilecek veri tanımlama.

public int puan;                                    //Dışarıdan ulaşılabilecek veri tanımlama.

void Start()                                         //Başlangıçta yapılan işlemler.
{

Cursor.visible = true;                             //Görünür olma durumu.

Cursor.lockState = CursorLockMode.None; //Değişiklik olmama durumu.

skor.text = "PUANINIZ:" + PlayerPrefs.GetInt("Puan:");

//Skor textini kayıtlı ifade hâline getirme.

if (PlayerPrefs.GetInt("Puan:") <= 40 && PlayerPrefs.GetInt("Puan:") >= 0)

//Eğer kayıtlı puan 0 ve 40 dahil 0 ve 40 arasında ise.

```

```

{
mesaj.text = "KÖTÜ";           //Mesaj textini gösterir.
}

else if(PlayerPrefs.GetInt("Puan:") <=80 && PlayerPrefs.GetInt("Puan:") > 40)
//Eğer kayıtlı puan 40 ve 80 dahil 40 ve 80 arasında ise.

{
mesaj.text = "ZAYIF";           //Mesaj textini gösterir.
}

else if (PlayerPrefs.GetInt("Puan:") <= 120 && PlayerPrefs.GetInt("Puan:") > 80)
//Eğer kayıtlı puan 80 ve 120 dahil 80 ve 120 arasında ise.

{
mesaj.text = "VASAT";           //Mesaj textini gösterir.
}

else if (PlayerPrefs.GetInt("Puan:") <= 160 && PlayerPrefs.GetInt("Puan:") > 120)
//Eğer kayıtlı puan 120 ve 160 dahil 120 ve 160 arasında ise.

{
mesaj.text = "İYİ";           //Mesaj textini gösterir.
}

else if (PlayerPrefs.GetInt("Puan:") <= 200 && PlayerPrefs.GetInt("Puan:") > 160)
//Eğer kayıtlı puan 160 ve 200 dahil 160 ve 200 arasında ise.

{

```



```

mesaj.text = "ÇOK İYİ";                //Mesaj textini gösterir.

}

else if (PlayerPrefs.GetInt("Puan:") <= 240 && PlayerPrefs.GetInt("Puan:") > 200)

//Eğer kayıtlı puan 200 ve 240 dahil 200 ve 240 arasında ise.

{

mesaj.text = "MÜKEMMEL";                //Mesaj textini gösterir.

}

else if (PlayerPrefs.GetInt("Puan:") <= 280 && PlayerPrefs.GetInt("Puan:") > 240)

//Eğer kayıtlı puan 240 ve 280 dahil 240 ve 280 arasında ise.

{

mesaj.text = "OLAĞANÜSTÜ";                //Mesaj textini gösterir.

}

else if (PlayerPrefs.GetInt("Puan:") <= 320 && PlayerPrefs.GetInt("Puan:") > 280)

//Eğer kayıtlı puan 280 ve 320 dahil 280 ve 320 arasında ise.

{

mesaj.text = "DOĞAÜSTÜ";                //Mesaj textini gösterir.

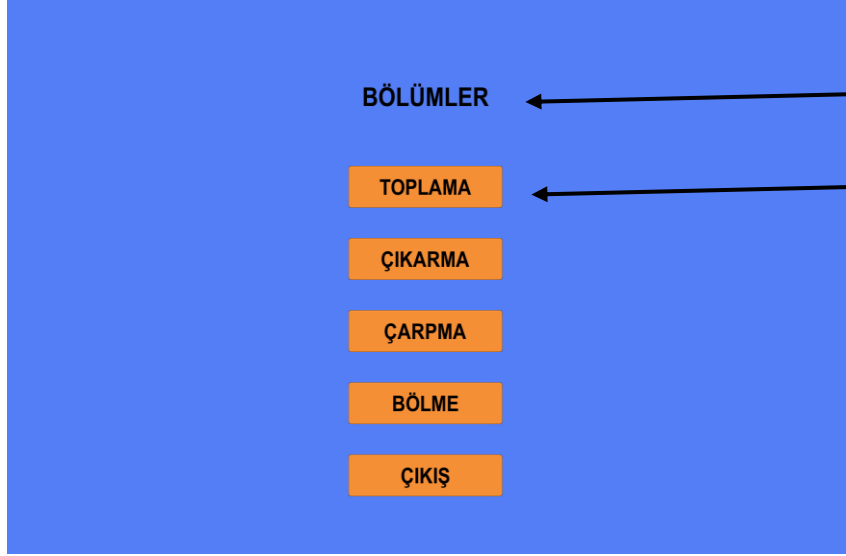
}

}

}

```

## 9. KULLANIM KILAVUZU

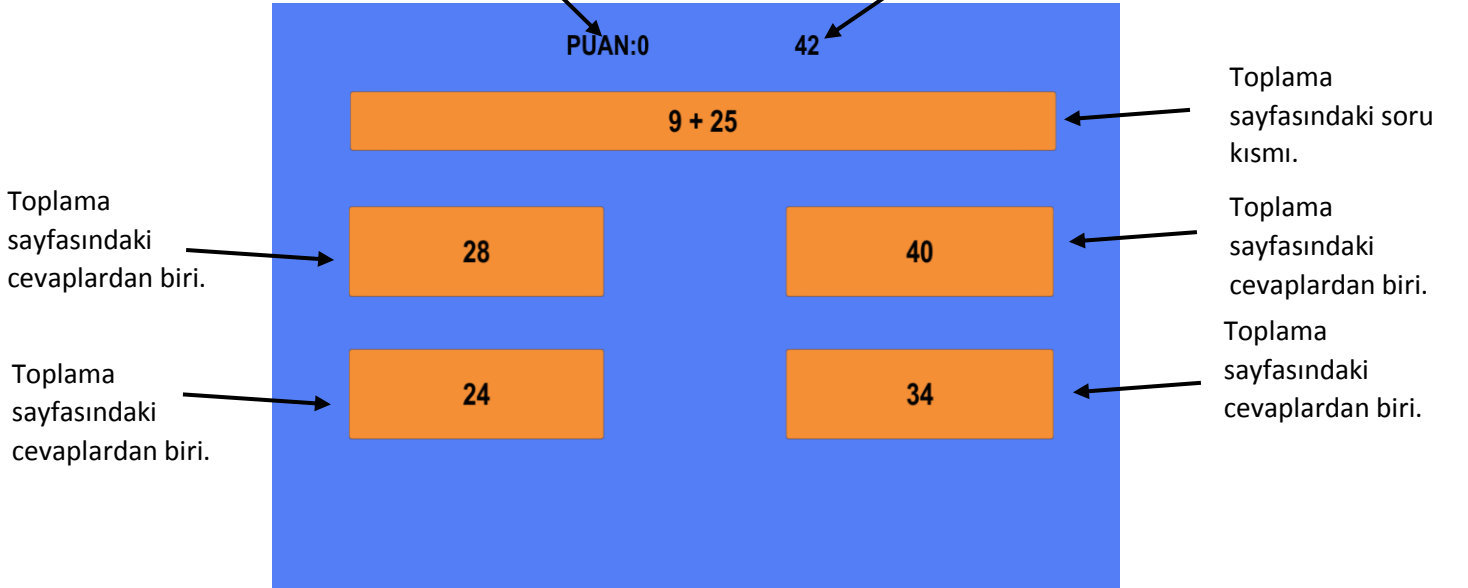


Ana menüdeki bölümler sayfası.

Ana menüde toplama butonuna tıklayarak toplama sayfasını açarız.

Toplama sayfasındaki puan 0' dan başlar, verilen cevap doğru ise 10 puan artar, yanlış ise 10 puan azalır. Asla 0' ın altına düşmez.

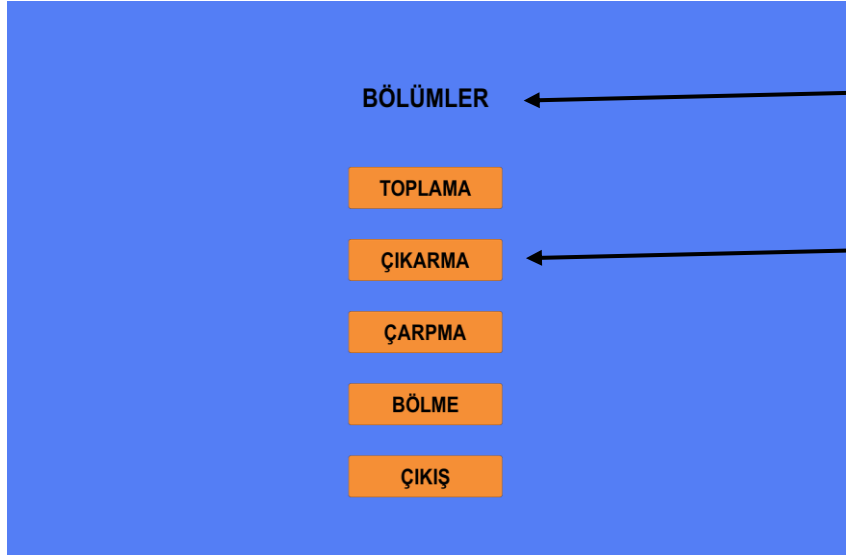
Toplama sayfasındaki süre 45' den başlar ve 1' er 1' er azalır. Süre 0' a geldiğinde oyun biter ve oyun sonu sayfası gelir.



Toplama sayfasındaki soru kısmı.

Toplama sayfasındaki cevaplardan biri.

Toplama sayfasındaki cevaplardan biri.



BÖLÜMLER

TOPLAMA

ÇIKARMA

ÇARPMA

BÖLME

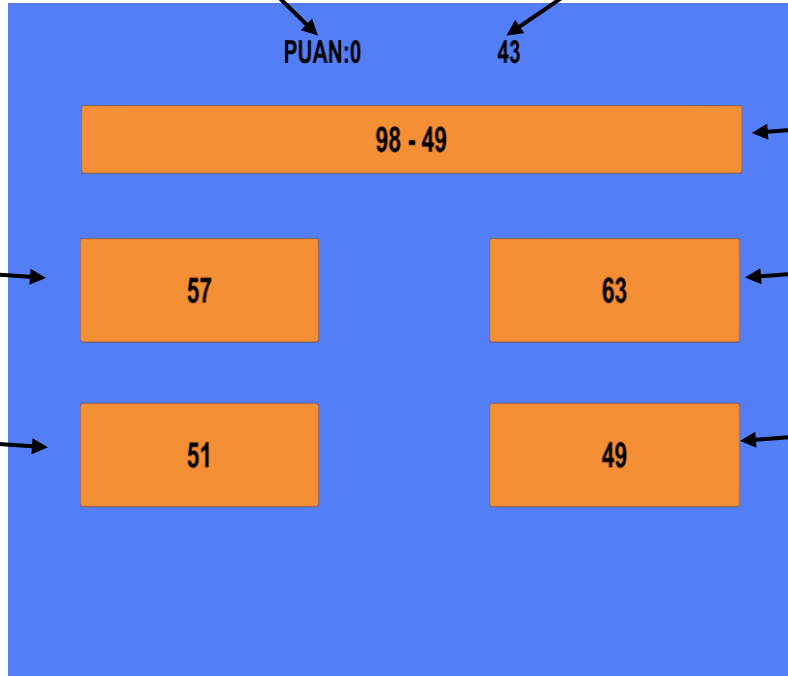
ÇIKIŞ

Ana menüdeki bölümler sayfası.

Ana menüde çıkarma butonuna tıklayarak çıkarma sayfasını açarız.

Çıkarma sayfasındaki puan 0' dan başlar, verilen cevap doğru ise 10 puan artar, yanlış ise 10 puan azalır. Asla 0' ın altına düşmez.

Çıkarma sayfasındaki süre 45' den başlar ve 1' er 1' er azalır. Süre 0' a geldiğinde oyun biter ve oyun sonu sayfası gelir.



PUAN:0

43

98 - 49

57

63

51

49

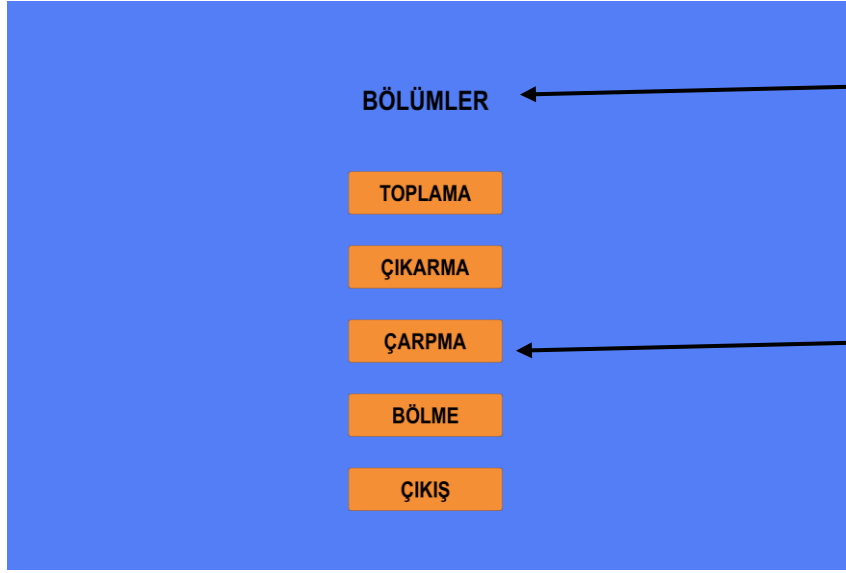
Çıkarma sayfasındaki soru kısmı.

Çıkarma sayfasındaki cevaplardan biri.

Çıkarma sayfasındaki cevaplardan biri.

Çıkarma sayfasındaki cevaplardan biri.

Çıkarma sayfasındaki cevaplardan biri.

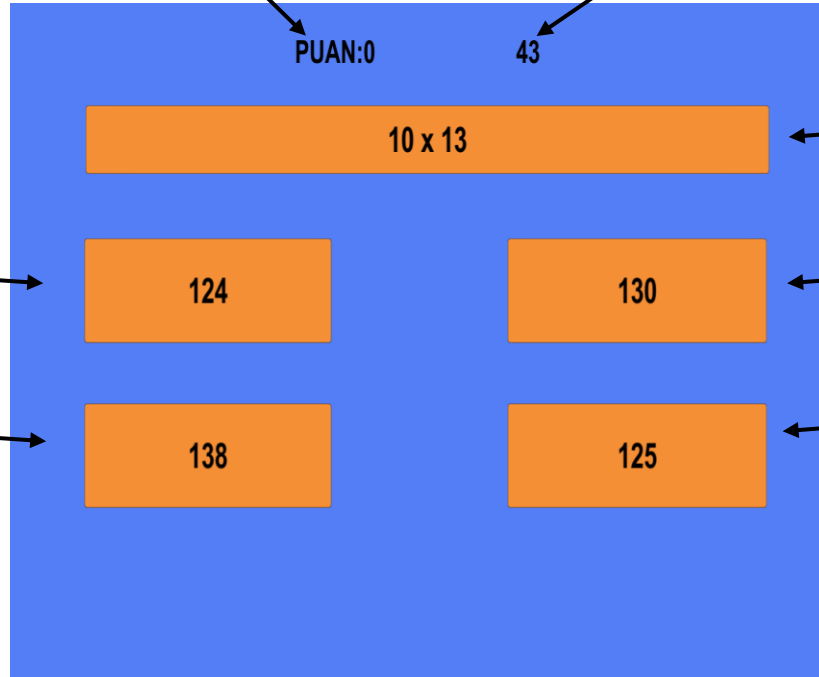


Ana menüdeki bölümler sayfası.

Ana menüde çarpma butonuna tıklayarak çarpma sayfasını açarız.

Çarpma sayfasındaki puan 0' dan başlar, verilen cevap doğru ise 10 puan artar, yanlış ise 10 puan azalır. Asla 0' ın altına düşmez.

Çarpma sayfasındaki süre 45' den başlar ve 1' er 1' er azalır. Süre 0' a geldiğinde oyun biter ve oyun sonu sayfası gelir.



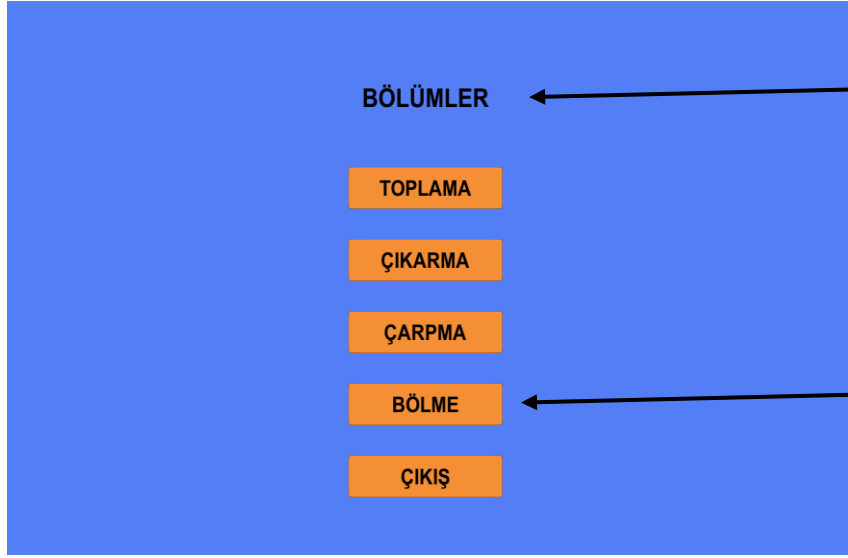
Çarpma sayfasındaki soru kısmı.

Çarpma sayfasındaki cevaplardan biri.

Çarpma sayfasındaki cevaplardan biri.

Çarpma sayfasındaki cevaplardan biri.

Çarpma sayfasındaki cevaplardan biri.



Ana menüdeki bölümler sayfası.

Ana menüde bölme butonuna tıklayarak bölme sayfasını açarız.

Bölme sayfasındaki puan 0' dan başlar, verilen cevap doğru ise 10 puan artar, yanlış ise 10 puan azalır. Asla 0' ın altına düşmez.

Bölme sayfasındaki süre 45' den başlar ve 1' er 1' er azalır. Süre 0' a geldiğinde oyun biter ve oyun sonu sayfası gelir.

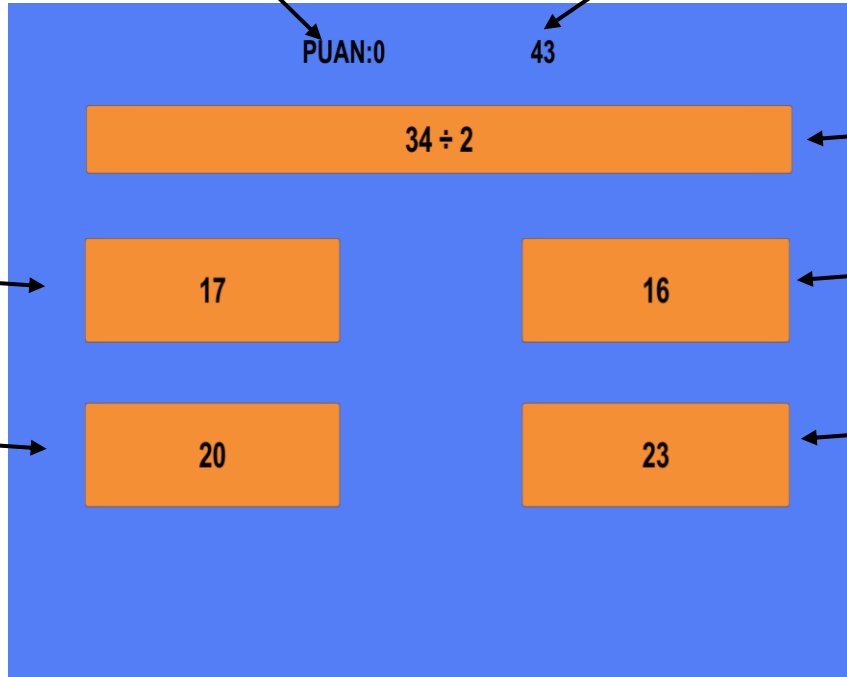
Bölme sayfasındaki cevaplardan biri.

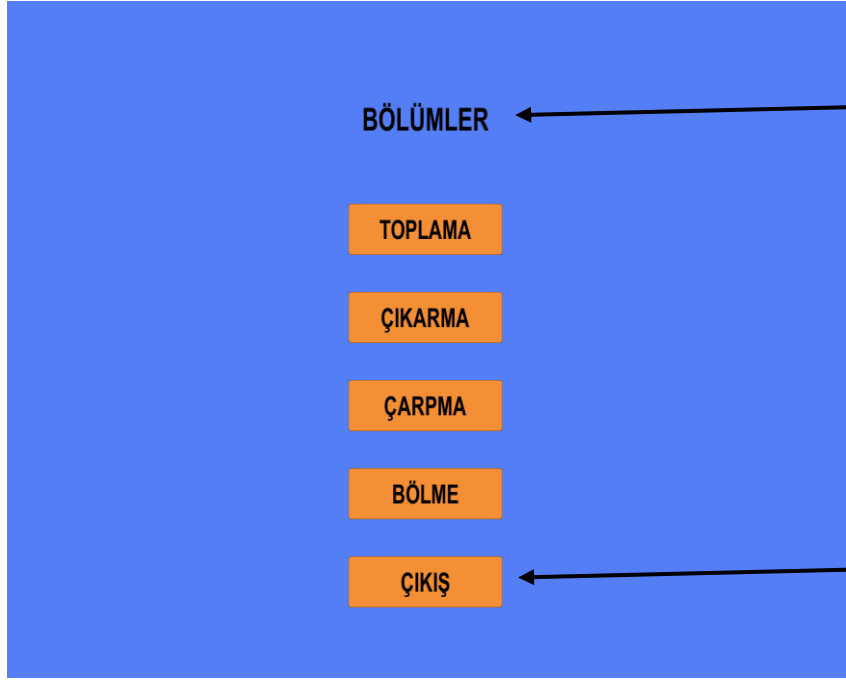
Bölme sayfasındaki cevaplardan biri.

Bölme sayfasındaki soru kısmı.

Bölme sayfasındaki cevaplardan biri.

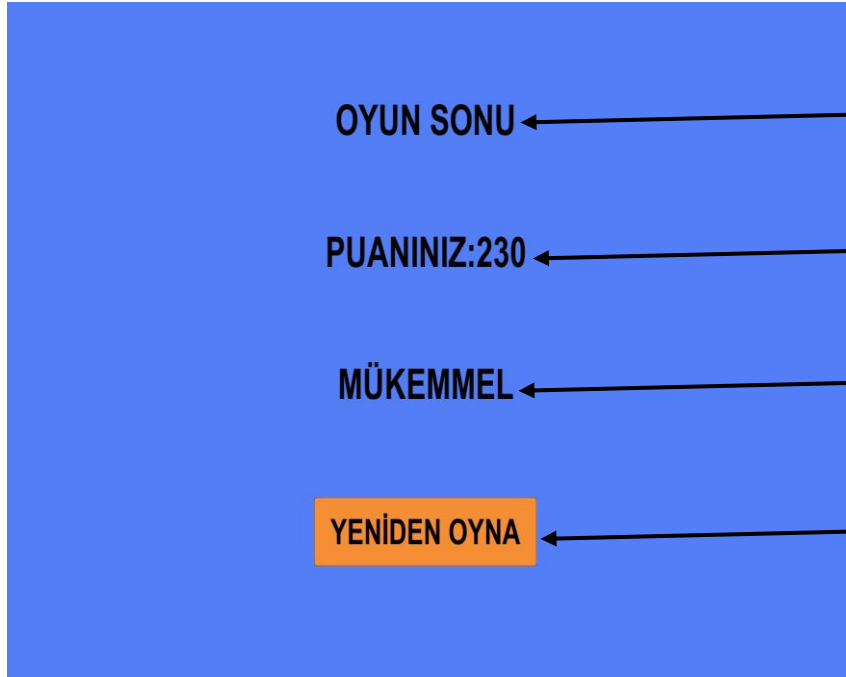
Bölme sayfasındaki cevaplardan biri.





Ana menüdeki bölümler sayfası.

Ana menüde çıkış butonuna tıklayarak çıkış sayfasını açarız.



Oyun bittikten sonra açılan puan ekranıdır.

Oyun sonu sayfasında oyun bittikten sonra alınan puanı gösterir.

Oyun sonu sayfasında oyun bittikten sonra alınan puana göre alınan dereceyi gösterir.

Oyun sonu sayfasında yeniden oyna butonuna tıklayarak ana menüyü açarız.

## 10. KAYNAKÇA

<https://unity.com/>

//Unity oyun motorunu kullanarak projemi yaptım.

<https://visualstudio.microsoft.com/tr/>

//Visual studio ile projemi kodladım

## 11. ÖZGEÇMİŞ

İstanbul' da Bakırköy' de dünyaya geldim.

Bağlar ilköğretim Okulu' nda okudum.

Bağlar Ortaokulu' nda okudum.

Recep Güngör Anadolu Lisesi' nde okudum. (Lise 1)

Barbaros Anadolu Lisesi' nde okudum. (Lise 2)

Fusun Yönder Lisesi' nde okudum. (Lise 3)

Uğur Temel Lisesi' nde okudum. (Lise 4)

Namık Kemal Üniversitesi' nde hazırlık okudum.

Namık Kemal Üniversitesi' nde bilgisayar mühendisliği okuyorum.