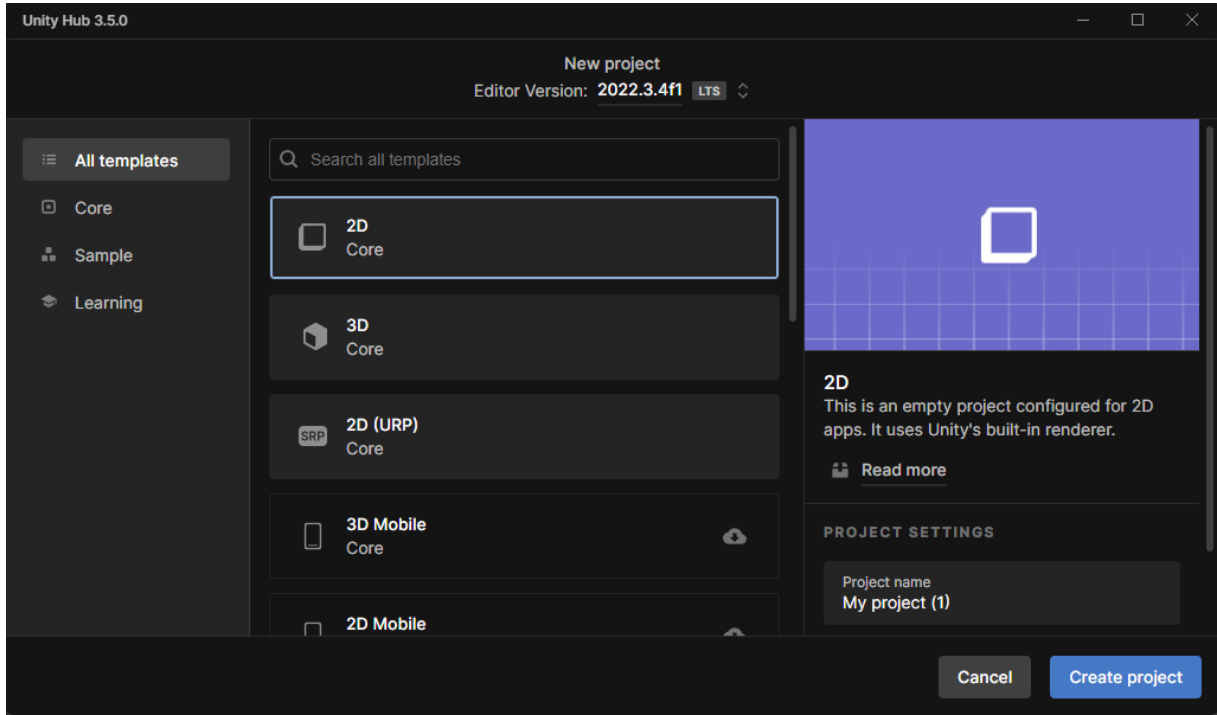
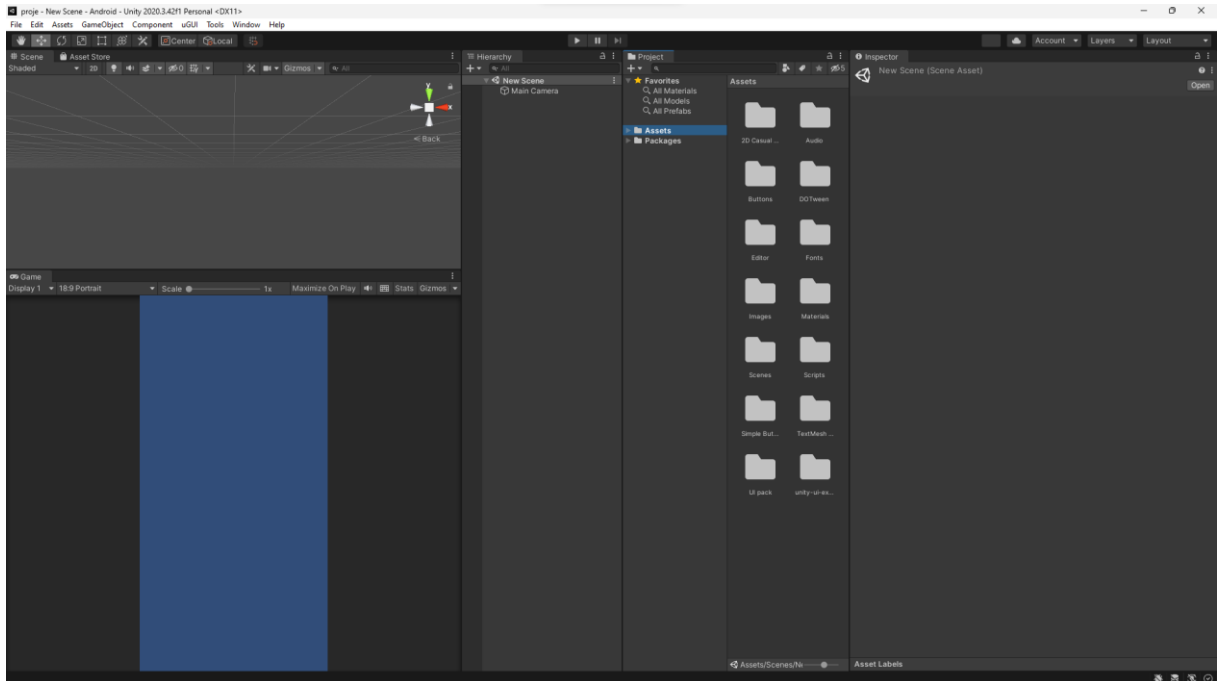


İlk olarak unity üzerinden yeni bir proje oluşturdum.



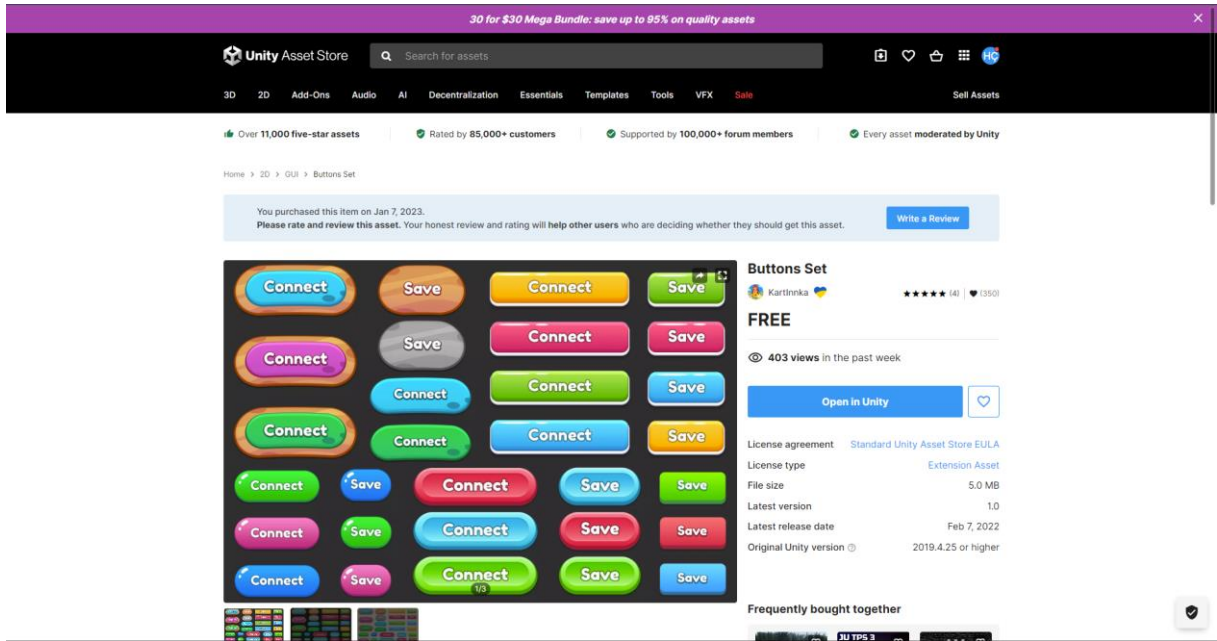
Karşıma gelen boş erkanda Scenes kısmına girip kaç adet sahne kullanmam gerekiyor ise sahnelerimi oluşturdum



Kullandığım sahneler aşağıda mevcut

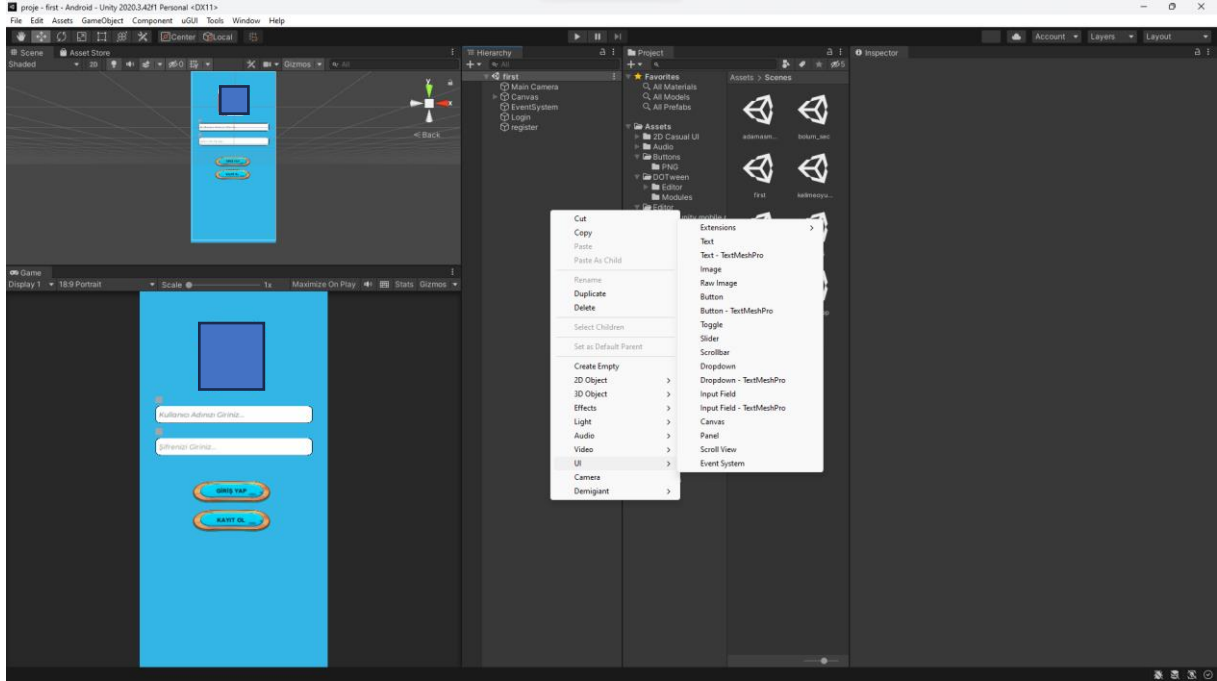


Ardından assetstore.unity.com sitesi üzerinden projede kullanılacak olan gerekli assetler projeye eklendi

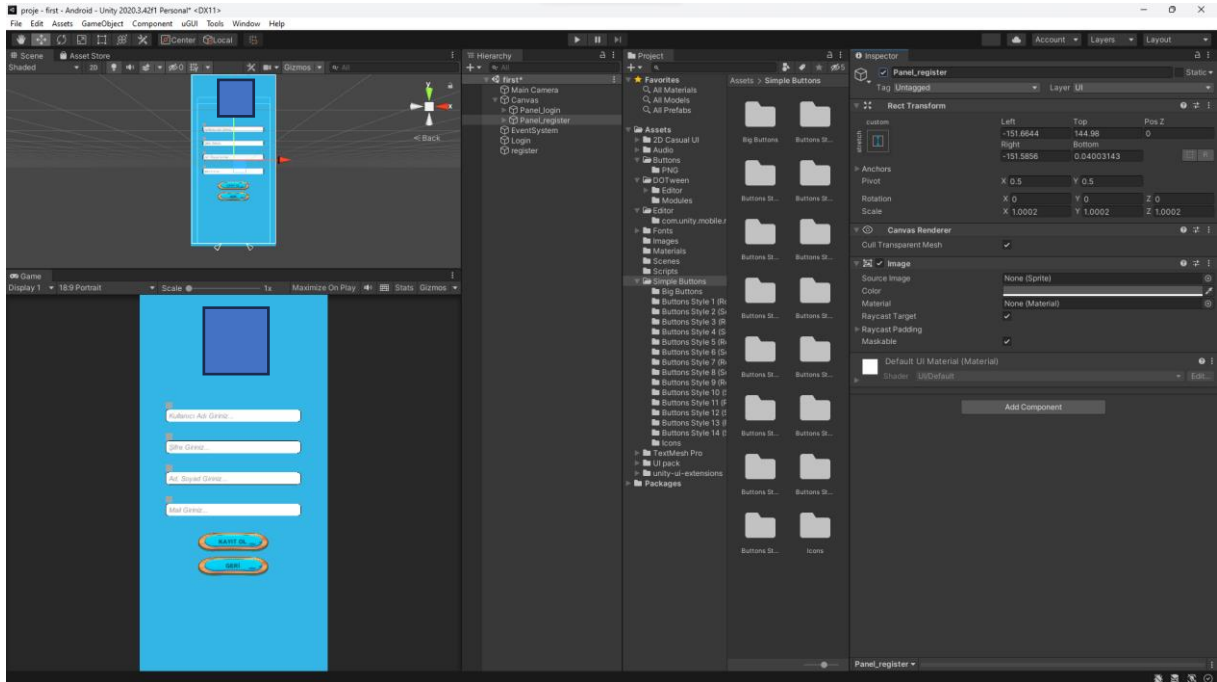


GİRİŞ

Hierarchy kısmından gerekli öğeleri ve tasarımları ekleyerek sayfayı oluşturdum



Aynı sayfa içerisinde kayıt ol kısmı da gömüldü.



Giriş yapılabilmesi için sunucuda bulunan veri tabanına gerekli olan sorguların yollanması. Hata alınması durumunda hata sonucunun verilmesi.

```

13 public class login : MonoBehaviour
14 {
15     private string baseURL=" ";
16     string _URL;
17     public Text _message;
18     public InputField _usernameInput;
19     public InputField _passwordInput;
20     public static string id;
21     IEnumerator _enumeratorLogin(){
22         Debug.Log(baseURL);
23         _URL= baseURL+"usr="+ _usernameInput.text+"&pwd="+ _passwordInput.text;
24         WWW _wb = new WWW(_URL);
25         yield return _wb;
26         Debug.Log(_URL);
27         if (_passwordInput.text.Length > 0 && _usernameInput.text.Length > 0){
28
29             if (_wb.error == null)
30             {
31                 //baglanti basarili
32
33                 toJSON(_wb.text);
34
35             }else{
36                 _message.text=("İnternet bağlantı kontrol ediniz");
37             }
38
39         } else
40         _message.text ="İtfen bütün alanları doldurunuz";
41
42     }
43
44
45 private void Awake()
46 {
47     DontDestroyOnLoad(this.gameObject);
48 }
49 void toJSON(string _json){
50     loginJson _loginJson = JsonUtility.FromJson<loginJson>(_json);
51     if (!_loginJson.success)
52     {
53         _message.text=_loginJson.message;
54     }else{
55         _message.text=_loginJson.message;
56         id=_loginJson.id;
57         SceneManager.LoadScene("bolum_sec");
58     }
59 }

```

Sunucudan alınan verileri daha rahat kullanabilmek için JSON formatında aldım.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using System;
4  public class loginJson
5  {
6      public bool success;
7      public string message;
8      public string name;
9      public string email;
10     public string id;
11 }
12

```

Kayıt olunabilmesi için veri tabanına eklenmesi gereken verilerin gönderilmesi.

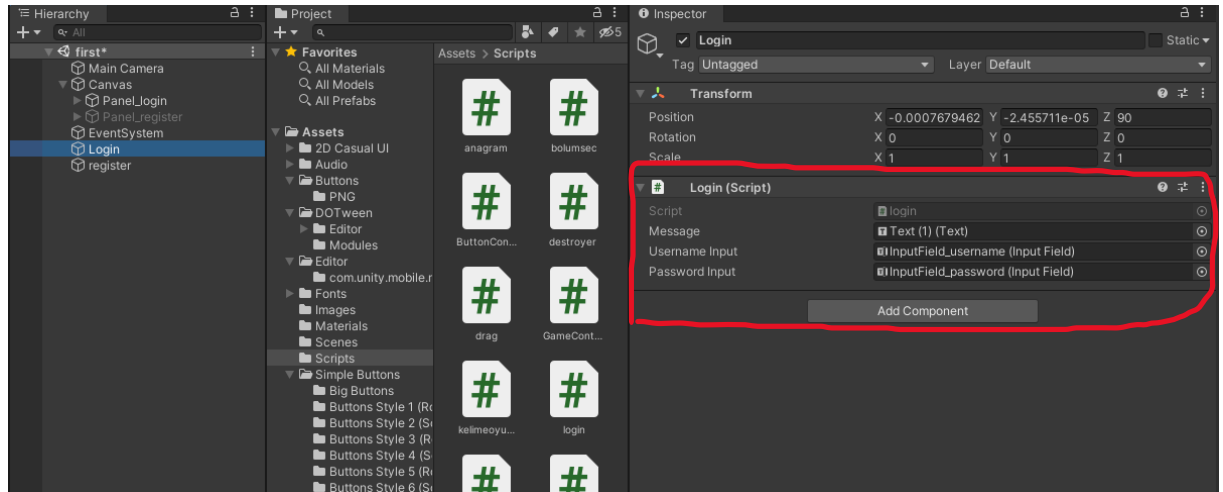
```

14 public class register : MonoBehaviour
15 {
16     public InputField usernameField;
17     public InputField passwordField;
18     public InputField emailField;
19     public InputField nameField;
20     public Text _message;
21
22     public void cal(){
23
24
25     if(usernameField.text.Length > 0 && passwordField.text.Length > 0 && emailField.text.Length > 0 && nameField.text.Length > 0)
26     StartCoroutine(KayitOl());
27     else
28     _message.text = "Lütfen bütün alanları doldurunuz !";
29
30 }
31
32 IEnumerator KayitOl()
33 {
34     WWWForm form = new WWWForm();
35     form.AddField("unity", "KayitOl");
36     form.AddField("username", usernameField.text);
37     form.AddField("name", nameField.text);
38     form.AddField("password", passwordField.text);
39     form.AddField("email", emailField.text);
40
41
42     using (UnityWebRequest www = UnityWebRequest.Post("http://localhost:5000/register", form))
43     {
44         yield return www.SendWebRequest();
45
46         if (www.result != UnityWebRequest.Result.Success)
47         {
48             Debug.Log(www.error);
49         }
50         else
51         {
52             _message.text = "Kayıt Başarılı !";
53             _message.text=www.downloadHandler.text;
54         }
55     }
56 }
57 }
58
59
60 }

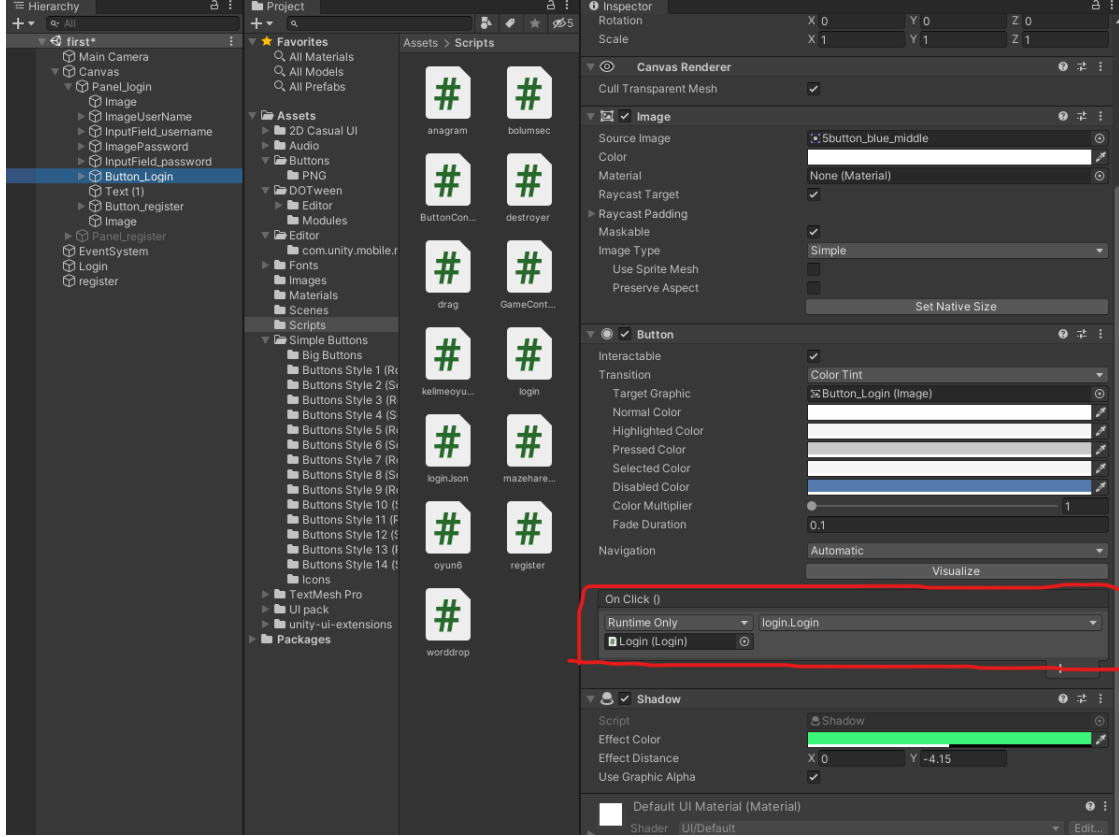
```

Gerekli kodlar yazıldıktan sonra her bir buton için açılan fonksiyonlara gerekli atamaları yaptım.

Öncelikle Boş bir Obje oluşturup yazdığımız Scripti(kod) objenin içine attım ki butonlarda kullanılabilsin. Oluşturduğumuz değişkenlere gerekli objeleri ekledik bu kısım aşağıda belirtildi. Bütün etkileşimli öğelerde bu yol izlendi



Ardından Buttonlara On Click aktif edildiğinde yapılması gereken işlemin foksiyonunu ekledim. Bu fonksiyonlar eklediğim kod görüntülerinden mevcuttur. Bu sayfada ek olarak butona basıldığı an değil giriş yapılabildiği an sayfanın değişmesi için diğer sayfalardan ayrı olarak eğer giriş yapılırsa sayfa değiştir kısmı eklendi(sarı alan). Örneğin;



```

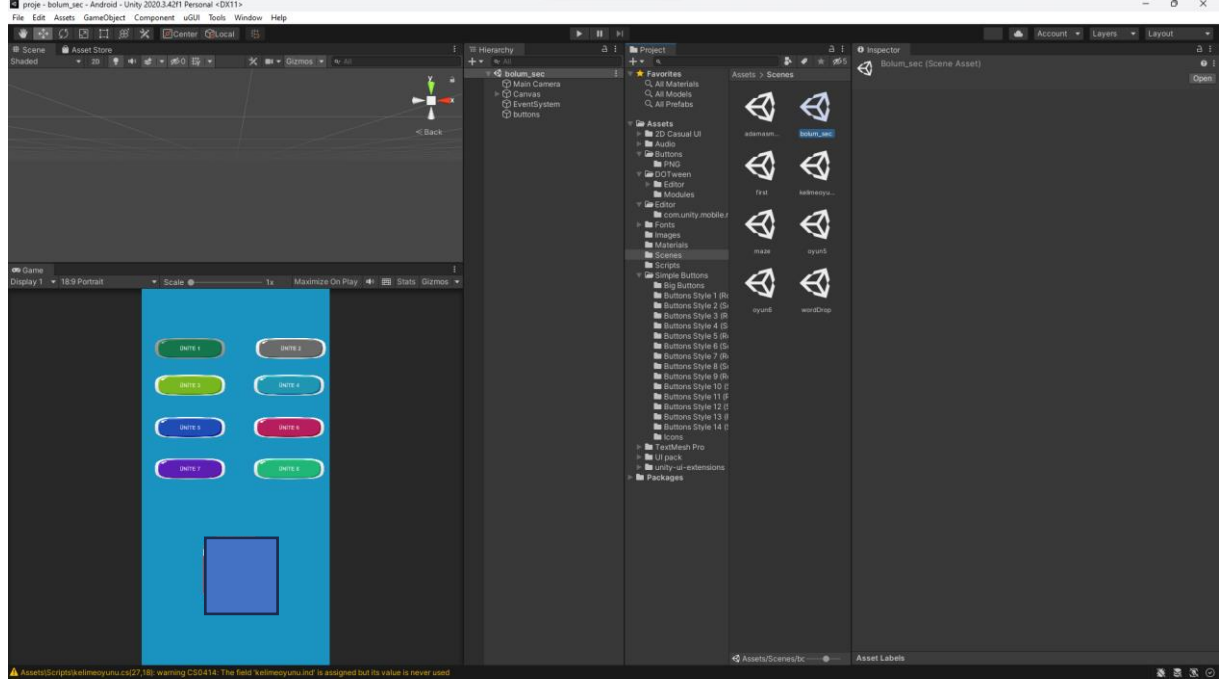
17 public Text _message;
18 public InputField _usernameInput;
19 public InputField _passwordInput;
20 public static string id;
21 IEnumerator _enumeratorLogin(){
22     Debug.Log(baseUrl);
23     _URL= baseUrl+"usr="+_usernameInput.text+"&pwd="+_passwordInput.text;
24     WWW _wb = new WWW(_URL);
25     yield return _wb;
26     Debug.Log(_URL);
27     if (_passwordInput.text.Length > 0 && _usernameInput.text.Length > 0){
28
29         if (_wb.error == null)
30         {
31             //baglanti basarili
32
33             toJSON(_wb.text);
34
35         }else{
36             _message.text=("İnternet bağlantınız kontrol ediniz");
37         }
38     } else
39     _message.text ="Lütfen bütün alanları doldurunuz";
40
41
42
43     }
44
45 private void Awake()
46 {
47     DontDestroyOnLoad(this.gameObject);
48 }
49 void toJSON(string _json){
50     loginJson _loginJson = JsonUtility.FromJson<loginJson>(_json);
51     if (!_loginJson.success)
52     _message.text=_loginJson.message;
53     else{
54         _message.text=_loginJson.message;
55         id=_loginJson.id;
56         SceneManager.LoadScene("bolum_sec");
57     }
58
59
60 }
61
62
63 public void Login(){
64     StartCoroutine(_enumeratorLogin());
65

```

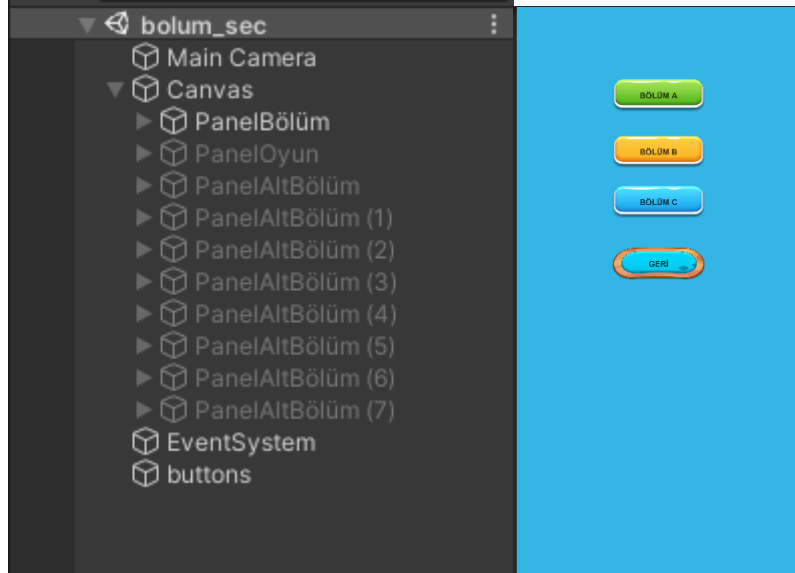
Bütün butonlarda aynı işlem uygulandı.

BÖLÜMLERİN SEÇİMİ

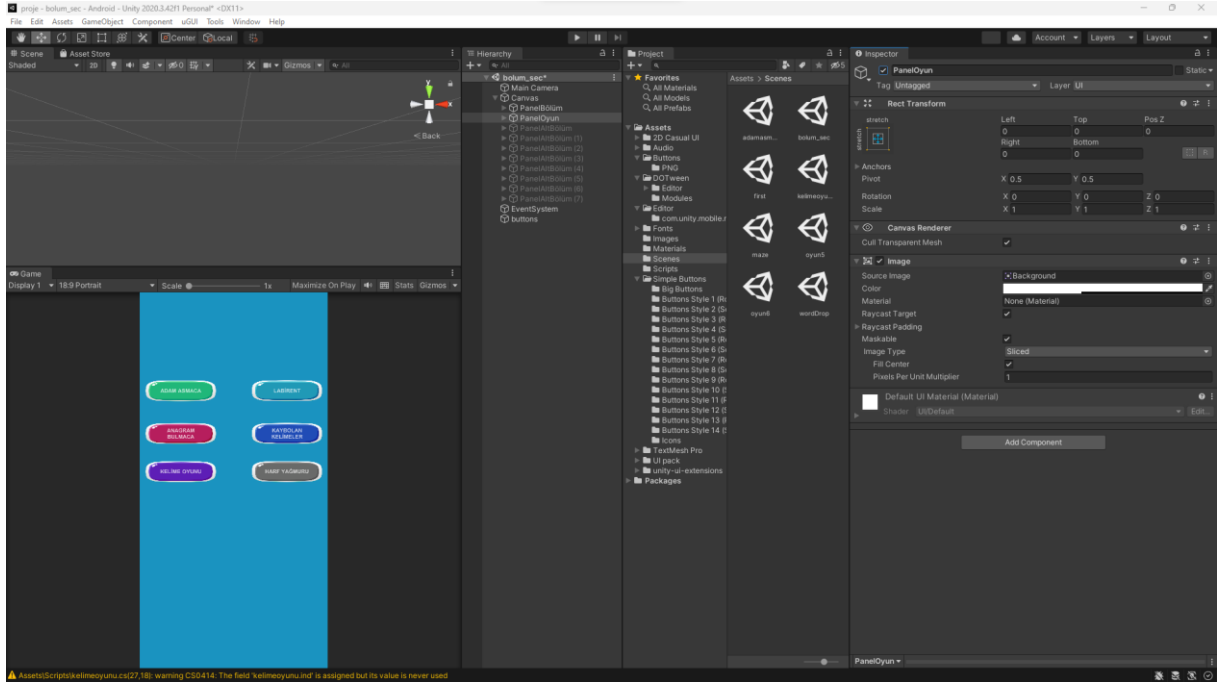
Bölümlerin seçilebilmesi için gerekli butonlar eklendi gerekli arka planlar ve tasarımları ilk sayfada olduğu gibi eklendi.



Her bir bölümün 3 adet alt bölümü bulunduğu için toplamda 8 adet alt bölüm oluşturuldu.



Bölümler seçildikten sonra hangi oyunun seçileceği kısım da aynı şekilde butonlar eklenerek oluşturuldu.



Bu kısımda her bir buton kendine has bir fonksiyon ile çalışıyor. Bu fonksiyonlar hangi kelime grubu ile oyunun oynanacağını belirliyor.

```
public void fonka1(){
    words=a1;
    wordsA=a1a;
}

public void fonka2(){
    words=a2;
    wordsA=a2a;
}

public void fonka3(){
    words=a3;
    wordsA=a3a;
}

public void fonka4(){
    words=a4;
    wordsA=a4a;
}

public void fonka5(){
    words=a5;
    wordsA=a5a;
}

public void fonka6(){
    words=a6;
    wordsA=a6a;
}

public void fonka7(){
    words=a7;
    wordsA=a7a;
}
```

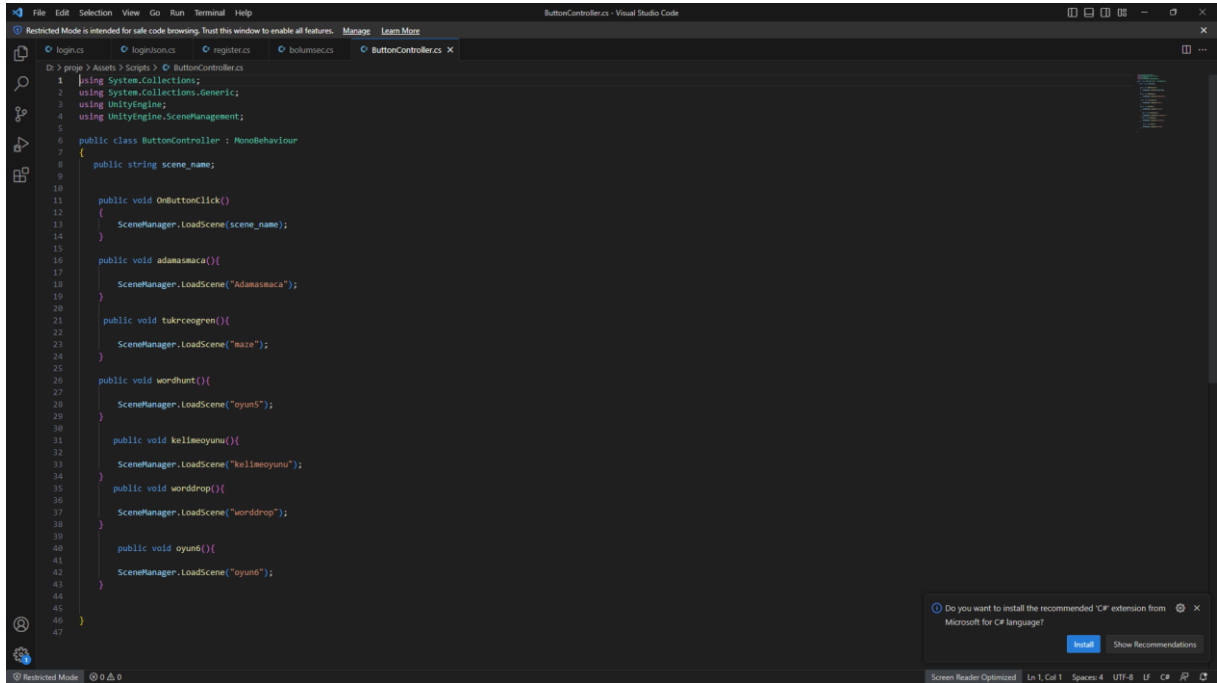
Aşağıdaki kod parçasında butonlardaki seçimler sonucunda Static olarak oluşturulan dizilerin diğer sayfalara yollanması için gerekli oluşturmaların yapıldığını görebilirsiniz.

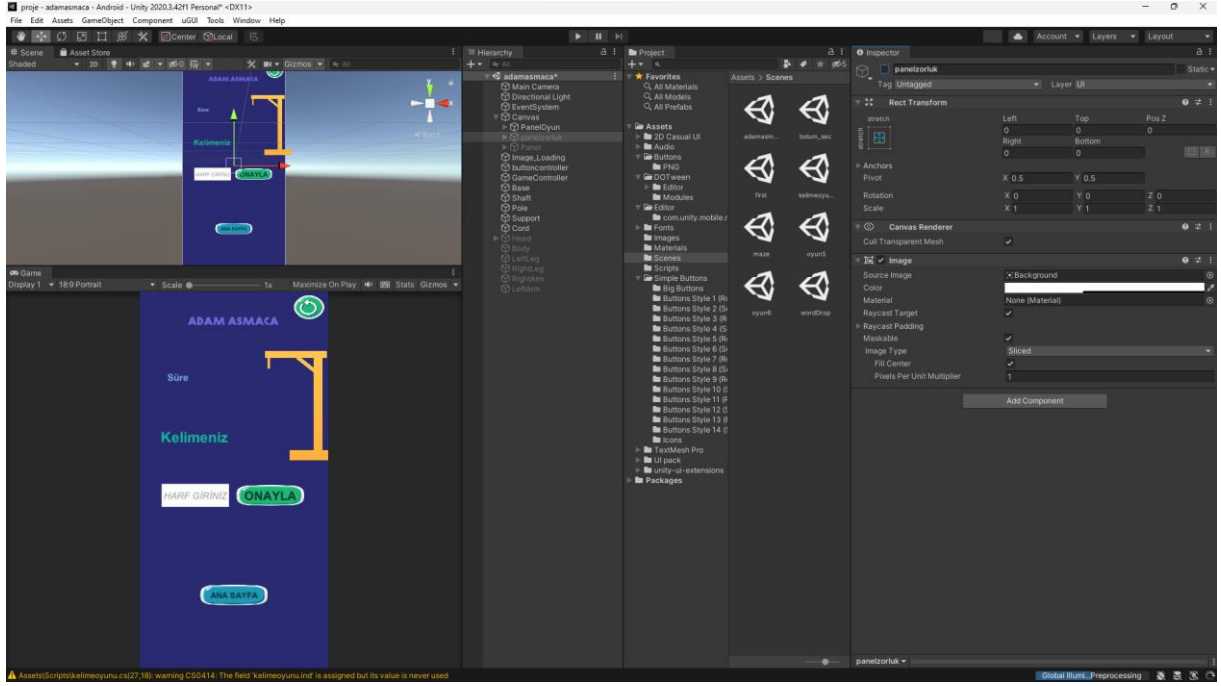
```

public static string[] words;
public static string[] wordsA;
public static string[] a1={"Ressam","Sergi","İlkokul","Plan","İş","Başlamak","Gelecek","Meslek","Ortaokul","Kariyer","Çalışmak","Çizmek"};
public static string[] a1a={"Resim yapan sanatçı",
"Alıcının görmesi , seçmesi için dizilmiş şeylerin tümü",
"İlk mektep , iptidai mektep",
"Bir işin, bir eserin gerçekleştirilmesi için uyulması tasarlanan düzen ",
"Birinden istenen hizmet veya birine verilen görev",
"Bir işe girişmek , harekete geçmek",
"Daha gelmemiş , yaşanacak zaman , istikbal, atı",
"Üğras",
"Eğitim yoluyla bir yandan hayata bir yandan da liseye hazırlayan okul",
"Bir meslekte zaman ve çalışmayla elde edilen aşama ",
"Bir şeyi oluşturmak veya ortaya çıkarmak için emek harcamak",
"Çizgi çekmek"
};
public static string[] b1={"Yıkamak",
"Anlatmak",
"Dünyaya gelmek",
"Hazırlamak",
"Okumak",
"Girmek",
"Plan yapmak",
"Yola çıkmak",
"Seyahat etmek",
"İstemek",
"Dua etmek",
"Yemek yapmak"
};
public static string[] b1a={"Su veya başka bir sıvı kullanarak bir şeyitemizlemek",
"Bilgi vermek , izah etmek",
"İnsan , doğmak",
"Bir şeyi kullanılacak , yararlanılacak duruma getirme , kurmak",
"Yazılmış bir metnin iletmek istediği şeyleri öğrenmek",
"Dışarıdan içeriye girmek",
"Yapılacak bir işi belli bir düzene göre düzenlemek",
"Araca binmek üzere yol üstünde durmak",
"Uzak yerleri gzerek görmek , yolculuk etmek",
"İstek duymak , arzulamak",
"Allah'a yalvarmak",
"Aşçının yaptığı iş"
};

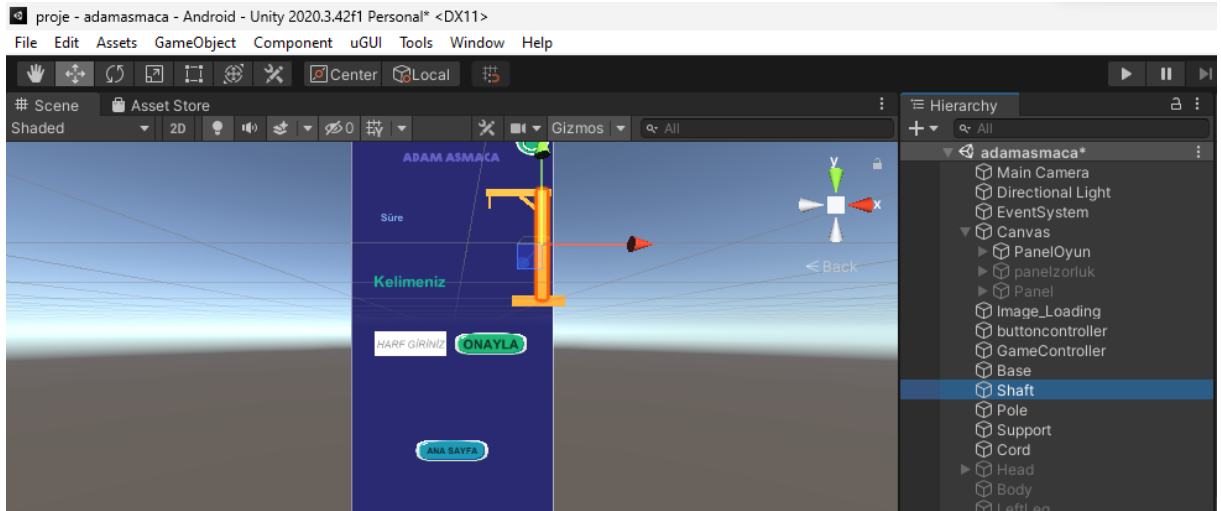
```

İlk olarak bu sayfada kullandığımız ButtonController.cs isimli scriptimiz içerisinde hangi sayfaya gitmek istediğimizi belirten 7 adet fonksiyon bulunuyor bu fonksiyonlar 6 adet oyunu gösteren butonlara atandı bu sayede her bir script içerisinde tek tek yazmak yerine topluca yazmış olduk.





Aslında direği ver her bir parça tek tek eklendi tek tek düzenlendi.



Oyunun oynanabilmesi için gerekli kodlar aşağıda bulunuyor kırmızı alan bir önceki sayfamızda seçtiğimiz kelimeleri oluşturduğumuz static öğeler sayesinde bu sayfaya çağırımızı sağlıyor.

Ayrıca Giriş yapıldıktan sonra her oyunda kullanıcı id değeri saklanarak bu kullanıcıya özel olan kayıtlar tutuluyor.

Start isimli fonksiyon oyun başlangıcında hangi değerlerin kullanılması gerektiğini ayarlamak ile görevli.

Bu oyunda ve bütün oyunlarda üst üste aynı kelime gelmemesi için bir kontrol mekanizması bulunuyor. Bu mekanizma İF bloğu içerisinde State değişkeni ile kontrol ediliyor.

```

public class GameController : MonoBehaviour
{
    public Text timeField,dogru;
    public GameObject[] hangMan;
    public GameObject replayButton;
    public Text wordToFindField;
    public InputField key;
    public Text soru;
    private float time=60;
    private float time2=0;
    private float puans;
    private static string[] words;
    private static string[] ques;
    private string chosenWord;
    private string hiddenWord;
    private string que;
    private int fails;
    private bool gameEnd=false;
    private bool gameStart=false;
    private int index;
    private string id;
    private int temp;
    public GameObject panel;
    public Text sonuc,cevap;

    private bool state;
    void Start()
    {
        id=login.id;
        key.characterLimit=1;
        words=bolumsec.words;
        ques=bolumsec.wordsA;
        index=Random.Range(0,words.Length);
        chosenWord=words[index];
        que=ques[index];
        Debug.Log("Chosen word is "+ chosenWord);
        chosenWord=chosenWord.ToUpper();

        for(int i=0; i<chosenWord.Length; i++)
        {
            char letter = chosenWord[i];

```

Replay olarak adlandırılan fonksiyon oyunun yeniden başlatılması istendiğinde her şeyi sıfırlamak ile görevli.

Update fonksiyonumuz bu oyunda ve bütün oyunlarda süre sayacını ve puanları anlık güncellemek ile görevli

```

void Update()
{
    soru.text=que;
    time2+=Time.deltaTime;

    if(gameEnd==false && gameStart==true)
    {
        time -=Time.deltaTime;
        timeField.text = time.ToString();
    }

    if(time<=0){
        gameEnd=true;
        panel.SetActive(true);
        sonuc.text="SÜREİNİZ BİTTİ KAYBETTİNİZ";
        cevap.text=chosenWord;
    }
}

public void replay(){
    hiddenWord="";
    dogru.text="";
    id=login.id;
    key.characterLimit=1;
    words=bolumsec.words;
    ques=bolumsec.wordsA;
    index=Random.Range(0,words.Length);
    chosenWord=words[index];
    que=ques[index];
    Debug.Log("Chosen word is "+ chosenWord);
    chosenWord=chosenWord.ToUpper();

    for(int i=0; i<chosenWord.Length; i++)
    {
        char letter = chosenWord[i];
        Debug.Log(letter);

        if(char.IsWhiteSpace(letter))
        {
            hiddenWord+=" ";
        }
        else{
            hiddenWord+="_";
        }
    }
}

```

Finish fonksiyonumuz oyunun bittiği anda çalışmak ve puanı hesaplamak için ve puanımızı veri tabanına kaydetmek için sendData Enumeratorünü çalıştırmak ile görevli.

SendData enumeratörü kullanıcı bilgilerine göre o kullanıcının oyun içerisinde geçirdiği süreyi ve puanı kayıt etmemizi sağlıyor.

```
public void finish(){
    puans=(int)time*2;
    StartCoroutine(sendData());
}

IEnumerator sendData()
{
    WWWForm form = new WWWForm();
    form.AddField("unity", "data");
    form.AddField("id", id);
    form.AddField("game", "Adam asmaca");
    form.AddField("time", time2.ToString());
    form.AddField("point", puans.ToString());

    using (UnityWebRequest www = UnityWebRequest.Post(' ', form))
    {
        yield return www.SendWebRequest();

        if (www.result != UnityWebRequest.Result.Success)
        {
            Debug.Log(www.error);
        }
        else
        {
            Debug.Log("Form upload complete!");
        }
    }
}
```

Zorluk ayarı için gerekli fonksiyonlar.

```
public void kolay(){
    gameStart=true;
    time=60;
    temp=60;
}

public void orta(){
    gameStart=true;
    time=45;
    temp=45;
}

public void zor(){
    gameStart=true;
    time=30;
    temp=30;
}
```

Oyunun ana mekaniklerinin çalıştığı OnGUI bölümü bu kısımda girilen harflerin doğruluğu kontrol ediliyor ve yanlış ise bir parça ekrana veriliyor.

```

private void OnGUI() {

    Event e=Event.current;
    if(state && !gameEnd)
    {
        state = false;
        string pressedLetter= key.text.ToUpper();
        key.text ="";
        Debug.Log("Keydown event was trigged : " + pressedLetter);

        if(chosenWord.Contains(pressedLetter))
        {
            int i=chosenWord.IndexOf(pressedLetter);
            while(i!=-1)
            {
                hiddenWord=hiddenWord.Substring(0,i)+ pressedLetter+hiddenWord.Substring(i+1);
                Debug.Log(hiddenWord);
                chosenWord=chosenWord.Substring(0,i)+ "_" +chosenWord.Substring(i+1);
                Debug.Log(chosenWord);

                i=chosenWord.IndexOf(pressedLetter);
            }
            wordToFindField.text=hiddenWord;
        }else

        // add a hanman body
        {
            hangMan[fails].SetActive(true);
            fails++;
        }

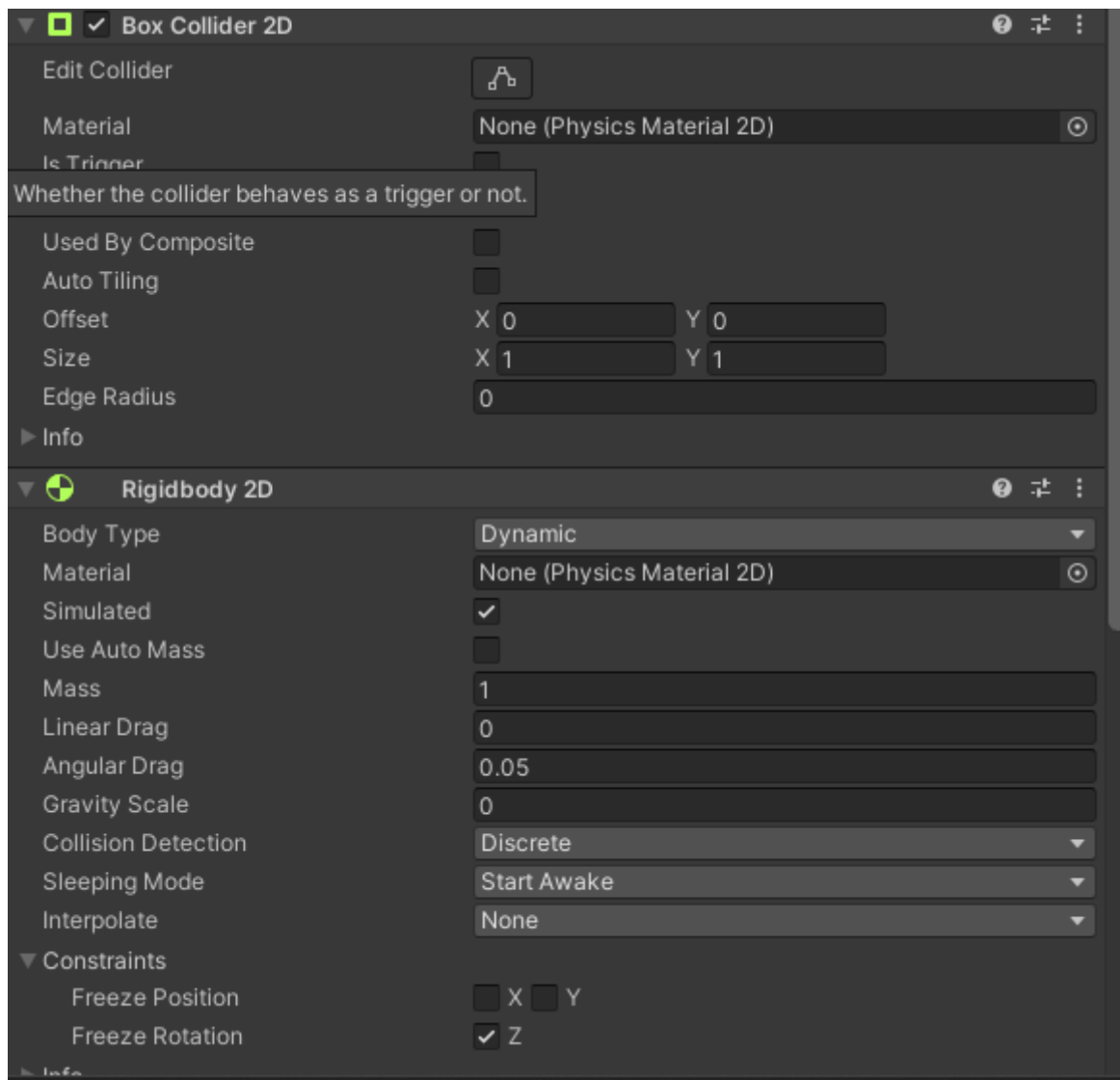
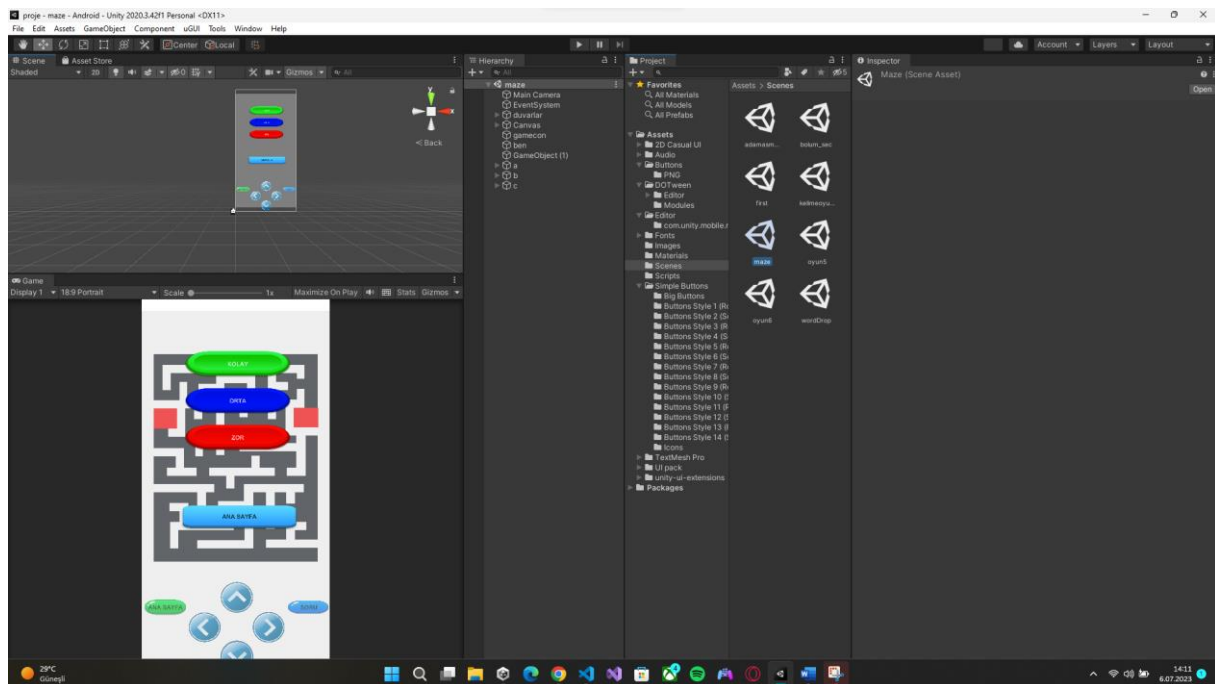
        //CaSE lost game
        if(fails==hangMan.Length){

            gameEnd=true;
            panel.SetActive(true);
            cevap.text=chosenWord;
            sonuc.text="Kaybettiniz";
            replayButton.SetActive(true);
        }
        // case won game
        if(!hiddenWord.Contains("_"))

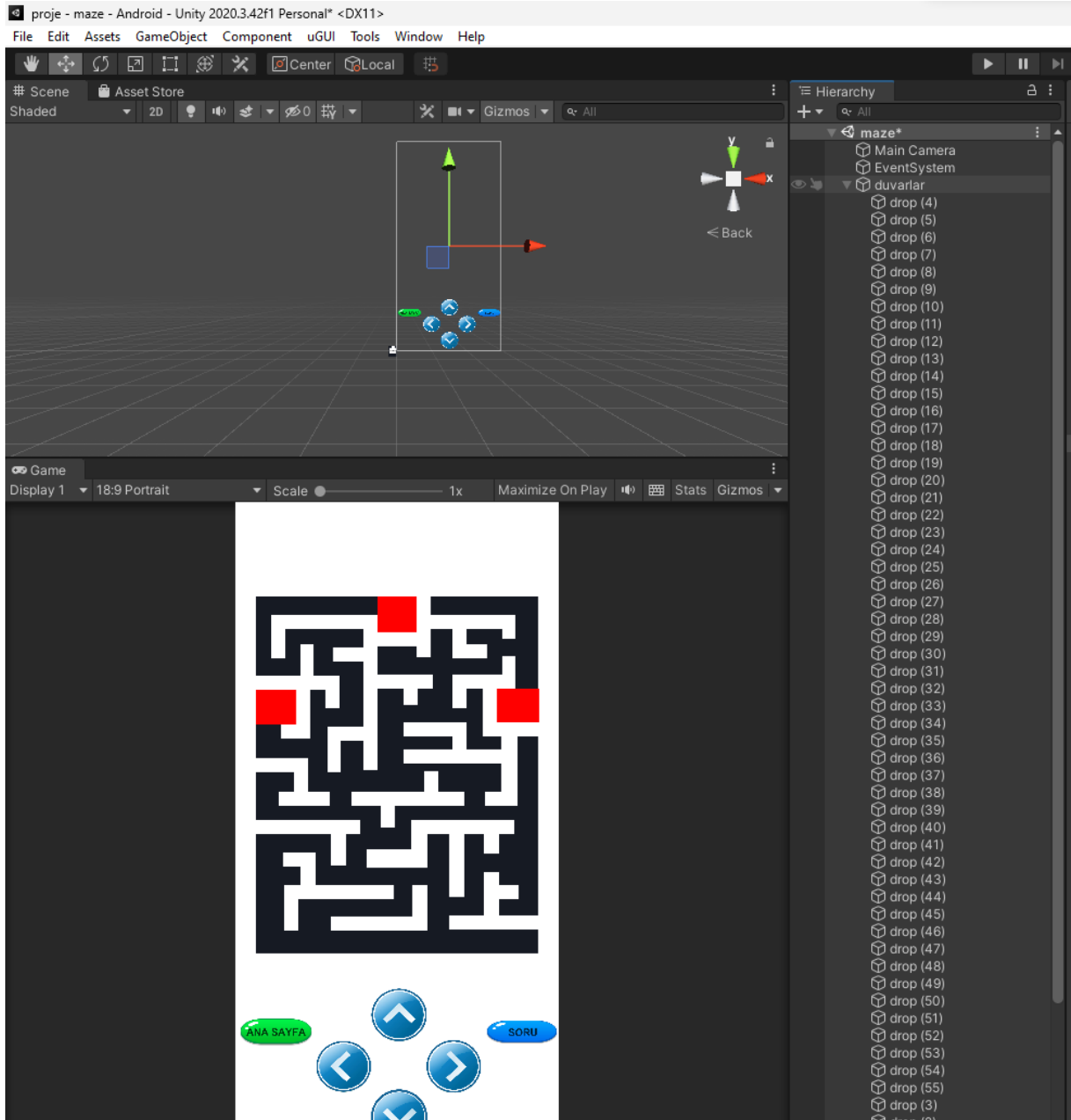
```

LABİRENT

Her oyunda olduğu gibi karşımıza zorluk ayarı çıkıyor. Bu oyunda Yönettiğimiz kutu duvarların içerisinden geçmemesi için fiziksel öğeler kullandık. Bir öğeye fizik eklememizi sağlayan componentler ekledik.



Gerekli duvarları ekleyip labirenti oluşturduk ve bu duvarlar projede görüntü kirliliği yaratmaması için bir klasörde topladık.



Yazdığımız kodun çalışabilmesi için bir objenin içerisinde gerekli butonları textleri bu kısımda her oyunda olduğu gibi ekledik.


```

private void Update() {

    time2+=Time.deltaTime;
    timeField.text = time.ToString();
    time -=Time.deltaTime;

    if(time<=0)
    {
        Time.timeScale =0;
        timeField.text ="0";
        sonuc.text="Sureniz bitti";
        panel.SetActive(true);

    }

    public void kolay(){
time=60;
Time.timeScale = 1;
}
    public void orta(){
time=45;
Time.timeScale = 1;
}
    public void zor(){
time=30;
Time.timeScale = 1;
}
}

```

```

login.cs loginbonus.cs register.cs bolumsec.cs ButtonController.cs GameController.cs mazehareket.cs X
D:\> proje > Assets > Scripts > mazehareket.cs
304 private Rigidbody2D rb;
305 public float movespeed=7f;
306 private string[] words=bolumsec.words;
307 private string[] words2=bolumsec.wordsA;
308 public GameObject panel;
309 private string chosenword;
310 private string sonuc;
311 public Text sonu;
312 public TextMeshPro a,b,c;
313 public Text timeField,puanfield;
314 private int puan;
315 private int inde;
316 private float time=1;
317 private float time2=0;
318 private string[] answers={"", "", ""};
319 private int index2;
320 private string cevap_a="";
321 private string cevap_b="";
322 private string cevap_c="";
323 public Text sonuc;
324 private string id;
325
326 void Start()
327 {
328     Time.timeScale = 0;
329     id=login.id;
330     rb = GetComponent<Rigidbody2D>();
331 }
332
333 public void finish(){
334     puan=(int)time*2;
335     StartCoroutine(sendData());
336 }
337
338 IEnumerator sendData()
339 {
340     MdfForm form = new MdfForm();
341     form.AddField("unity", "data");
342     form.AddField("id", id);
343     form.AddField("game", "maze");
344     form.AddField("time", time2.ToString());
345     form.AddField("point", puan.ToString());
346
347     using (UnityWebRequest www = UnityWebRequest.Post("http://192.168.1.100:8080/api/finish", form))
348     {
349         yield return www.SendWebRequest();
350         if (www.result != UnityWebRequest.Result.Success)
351             Debug.LogError(www.error);
352     }

```

Yeni soru getirmemizi sağlayan kod parçası;

```

public void yenile(){
    puanfield.text = puan.ToString();
    sonuc.text = "";
    Time.timeScale = 1;
    this.transform.position = new Vector2(-9,-3);
    inde=Random.Range(0,words.Length);
    chosenword=words[inde];
    sorua=words2[inde];
    soru.text=sorua;
    index2=Random.Range(0,answers.Length);
    Debug.Log(index2);
    answers[index2]=chosenword;
    for(int i=0;i<3;i++){

        if (i==index2) continue;
        tempStr=randomString[Random.Range(0,288)];
        if(answers[i]!=chosenword)
            answers[i]= tempStr;
    }
    cevap_a=answers[0];
    cevap_b=answers[1];
    cevap_c=answers[2];
    a.text=cevap_a;
    b.text=cevap_b;
    c.text=cevap_c;
}

```

Bu kısımda vektörler kullanarak fizik eklediğimiz hareket eden objemizin hangi yöne hareket etmesi gerektiğini belirleyen fonksiyonlar bulunuyor. Örneğin sol yön tuşuna basıldığında Sol fonksiyonu devreye girerek obje sola yönelim sağlıyor.

```

public void sol(){

    rb.velocity = Vector2.left * moveSpeed;

}

public void sag(){

    rb.velocity = Vector2.right * moveSpeed;;

}

public void alt(){

    rb.velocity = Vector2.up * -moveSpeed;

}

public void stop(){

    rb.velocity = Vector2.zero;

}

public void ust(){

    rb.velocity = Vector2.up * moveSpeed;

}

void OnCollisionEnter2D(Collision2D cls){

```

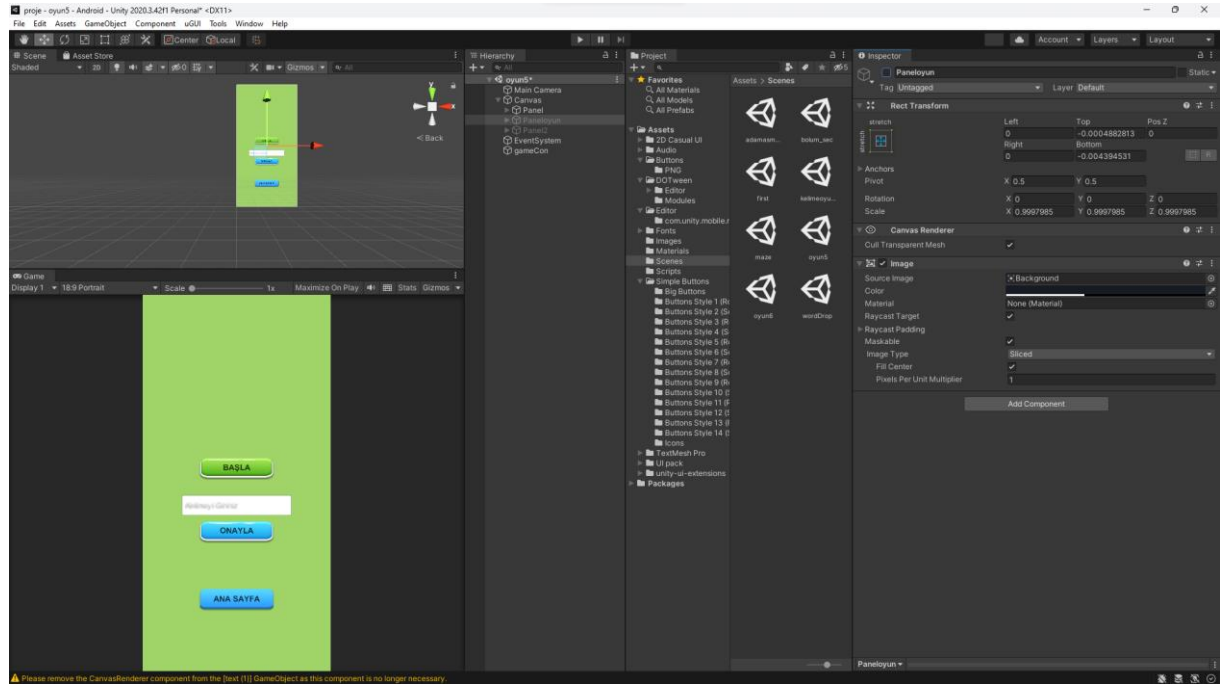
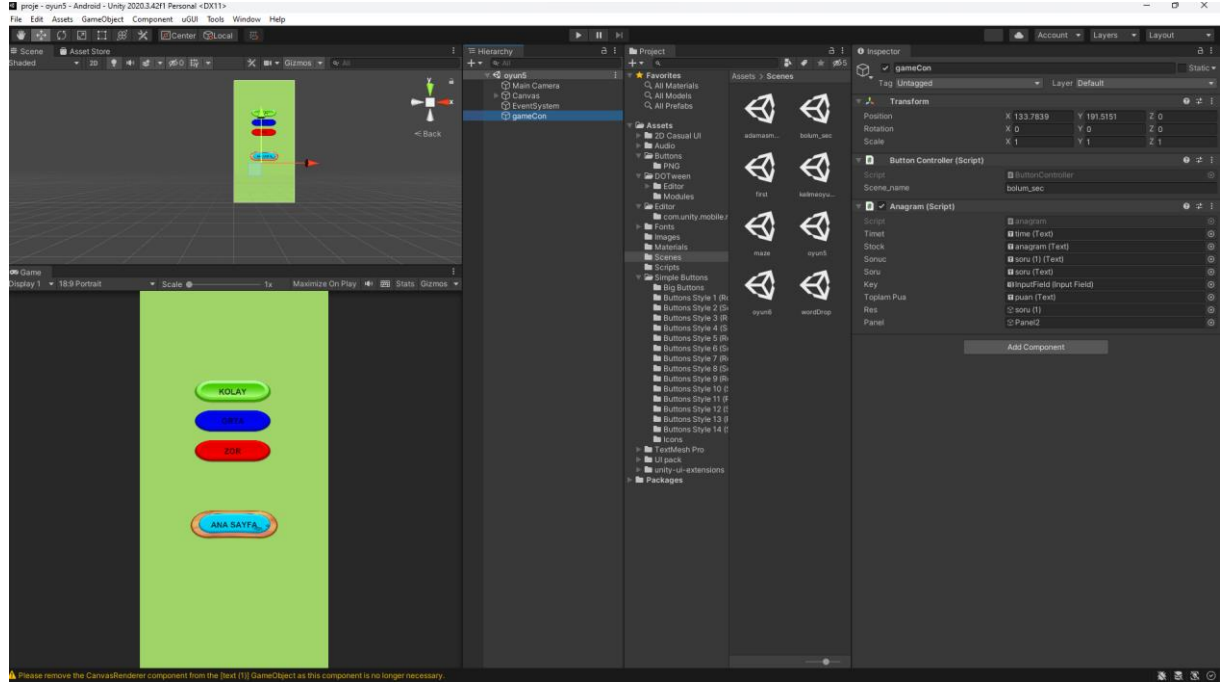
Bu fonksiyonumuzda objemizin temas ettiği ögenin doğru olup olmadığını kontrol ediyor sonuca göre işlemleri yaptırıyoruz.

```
}  
void OnCollisionEnter2D(Collision2D cls){  
  
    if(cls.gameObject.name.Equals("a")){  
        if(cevap_a==chosenword)  
        {sonuc.text="Doğru";  
        puan+=20;  
        Time.timeScale = 0;  
        }else{sonuc.text="Yanlış";}  
    }  
    if(cls.gameObject.name.Equals("b")){  
        if(cevap_b==chosenword)  
        {sonuc.text="Doğru";  
        puan+=20;  
        Time.timeScale = 0;  
        }else{sonuc.text="Yanlış";}  
    }  
    if(cls.gameObject.name.Equals("c")){  
        if(cevap_c==chosenword)  
        {sonuc.text="Doğru";  
        puan+=20;  
        Time.timeScale = 0;  
        }else{sonuc.text="Yanlış";}  
    }  
}
```

ANAGRAM BULMACA

Eklenmesi gereken öğeler ekleniyor tasarımlar yapılıyor.

Bu oyunuz direkt olarak kelimelerle ilgili olduğu için fiziksel bir obje bulunmuyor.



Gerekli tanımlamalar, seçilen bölüme uygun kelimelerin getirilmesi.

Start, Update, finish, sendData, kolay, orta, zor fonksiyonlarının tanımlanması.

```
public class anagram : MonoBehaviour
{
    private string id;
    private string[] words=bolumsec.words;
    private string[] words2=bolumsec.wordsA;
    private string chosenword;
    private string stockword;
    private string sorua;
    public Text timet;
    public Text stock;
    public Text sonuc;
    public Text soru;
    public InputField key;
    private float time=1;
    private float time2=0;
    public Text toplamPua;
    private int index;
    private int puan=0;
    public GameObject res,panel;
    string kontrol;
    int temp;

    void Start()
    {
        id=login.id;
        Time.timeScale = 0;
    }
}
```



```

void Update()
{
    time -=Time.deltaTime;
    time2+=Time.deltaTime;
    timet.text = time.ToString();
    toplamPua.text=puan.ToString();
    if(time<0)
    {
        Time.timeScale = 0;
        timet.text ="0";
        res.SetActive(true);
        panel.SetActive(true);

    }

    if(puan<0){

        Time.timeScale = 0;

        res.SetActive(true);
        sonuc.text="Puanınız Bitti!";

    }

}

```

```

IEnumerator sendData()
{
    WWWForm form = new WWWForm();
    form.AddField("unity", "data");
    form.AddField("id", id);
    form.AddField("game", "anagram oyunu");
    form.AddField("time", time2.ToString());
    form.AddField("point", puan.ToString());

    using (UnityWebRequest www = UnityWebRequest.Post("http://192.168.1.100:8080/submit" form))
    {
        yield return www.SendWebRequest();

        if (www.result != UnityWebRequest.Result.Success)
        {
            Debug.Log(www.error);
        }
        else
        {
            Debug.Log("Form upload complete!");
        }
    }
}

public void kolay(){
    time=90;
}

public void orta(){
    time=60;
}

public void zor(){
    time=30;
}
}

```

Oyunun bütün mekaniksel kodlamaları bu kısımda bulunuyor. Rasgele bir kelime seçilip bu kelimenin harfleri rasgele şekilde karıştırılıyor bu kısımda kelimenin orijinal dizilimi şeklinde karışmaması için kontrol yapıyor.

Objemize eklediğimiz metin kutularına soru ve anagram yollanıyor doğru cevap alınması durumunda doğru aksi takdirde yanlış uyarısı verilip puan kısmına yansıtılıyor.

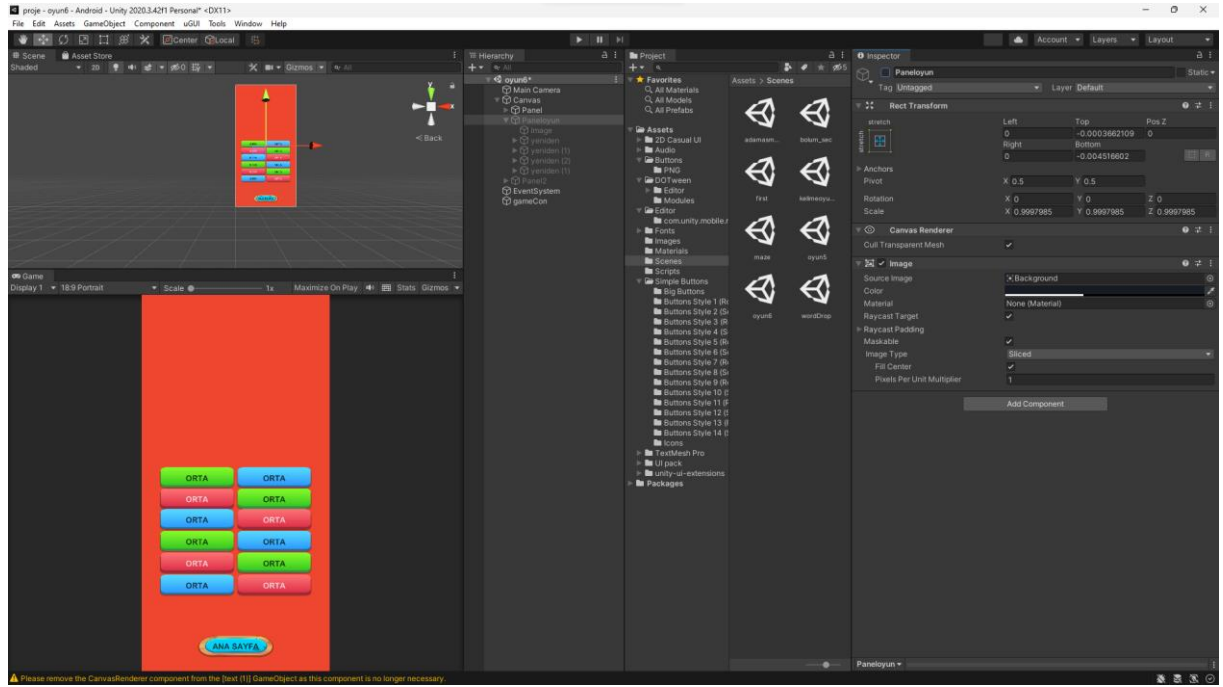
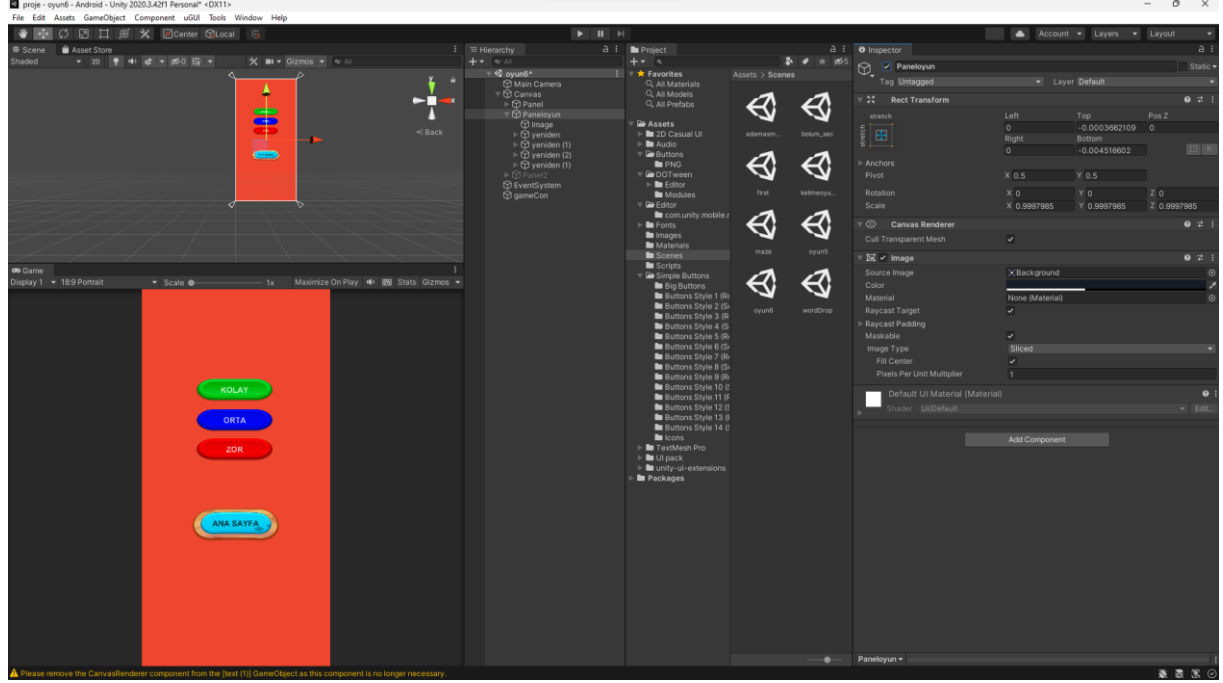
```
37 public void next(){
38     Time.timeScale = 1;
39     index=UnityEngine.Random.Range(0,words.Length);
40     if(index==temp){
41
42         next();
43         Debug.Log("değiştii");
44
45     }
46
47     temp=index;
48     chosenword=words[index];
49     sorua=words2[index];
50     char[] characters = chosenword.ToCharArray();
51     System.Random rng = new System.Random();
52     int n = characters.Length;
53
54     while (n > 1)
55     {
56         n--;
57         int k = rng.Next(n + 1);
58         var value = characters[k];
59         characters[k] = characters[n];
60         characters[n] = value;
61         Debug.Log(characters[n]);
62     }
63
64     string anagram= new string(characters);
65     string kontrol= anagram;
66     if(kontrol.ToLower()==chosenword.ToLower())
67     next();
68     Debug.Log(anagram);
69     Debug.Log(chosenword);
70     int state = Array.IndexOf(words, anagram);
71     if(state != -1)
72     next();
73     stock.text="";
74     stock.text = anagram.ToLower();
75     if(stock.text == "i")
76     next();
77     soru.text = sorua;
78 }
```

KAYBOLAN KELİMELER

Zorluk seçim ekranı ve ardından oyun ana ekranı tasarlanıyor.

Her bir bölümde 8 kelime bulunduğu için bu oyunda 8 buton kullandık.

Her bir buton için ayrı değer atandı.



Gerekli tanımlamalar, seçilen bölüme uygun kelimelerin getirilmesi.

Start, Update, finish, sendData, kolay, orta, zor fonksiyonlarının tanımlanması.

```
File Edit Selection View Go Run Terminal Help
oyun6.cs - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
D:\proje> Assets> Scripts> oyun6.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using System.IO;
5 using System.Text;
6 using UnityEngine.UI;
7 using System.Linq;
8 using UnityEngine.Networking;
9
10 public class oyun6 : MonoBehaviour
11 {
12
13     public Text a,b,c,d,ee,f,g,h,i,j,k,l;
14     public Text soru,timet,puant,sonuc;
15     public GameObject A,B,C,D,E,F,G,H,I,J,K,L,panel;
16     private float time=0;
17     private int puan;
18     public int Eksipuan;
19     private string chosenwords;
20     private static string[] words;
21     private static string[] wordsa;
22     private int index;
23     private int pos=5;
24     private int index2;
25     private string id;
26
27     void Start()
28     {
29         id=login.id;
30         words=bolumsec.words;
31         wordsa=bolumsec.wordsA;
32     }
33     public void finish(){
34         puan=(int)time*2;
35         StartCoroutine(sendData());
36     }
37
38     IEnumerator sendData()
39     {
40         WWWForm form = new WWWForm();
41         form.AddField("unity", "data");
42         form.AddField("id", id);
43         form.AddField("game", "kaybolan kelimeler");
44         form.AddField("time", time.ToString());
45         form.AddField("point", puan.ToString());
46
47         using (UnityWebRequest www = UnityWebRequest.Post("http://192.168.1.100:8080/api/finish", form))
48         {
49             yield return www.SendWebRequest();
50
51             if (www.result != UnityWebRequest.Result.Success)
52             {
53                 Debug.Log(www.error);
54             }
55             else
56             {
57                 Debug.Log("Form upload complete!");
58             }
59         }
60     }
61
62     void Update()
63     {
64         time -=Time.deltaTime;
65         timet.text = time.ToString();
66         if(time<=0){
67             Time.timeScale = 0;
68             panel.SetActive(true);
69             sonuc.text = "Süreniz Bitti! Kaybettiniz";
70         }
71         if(wordsa.Length==0){
72             Time.timeScale = 0;
73             panel.SetActive(true);
74             sonuc.text = "Tebrikler!!";
75         }
76     }
77 }
```

```

public void zor(){

    Eksipuan=15;
    yenile();
}
public void orta(){
    Eksipuan=10;
    yenile();
}
public void kolay(){
    yenile();
    Eksipuan=5;
}

```

Bu kısımda yeni soru istendiğinde ya da soru geçildiğinde rasgele olarak yeni bir soru getirilmesi ve ekrana verilmesi sağlanıyor.

```

public void yenile(){
    words=bolumsec.words;
    wordsa=bolumsec.wordsA;
    index=Random.Range(0,words.Length);
    chosenwords=wordsa[index];
    soru.text=chosenwords;

    a.text=words[0];
    b.text=words[1];
    c.text=words[2];
    d.text=words[3];
    ee.text=words[4];
    f.text=words[5];
    g.text=words[6];
    h.text=words[7];
    i.text=words[8];
    j.text=words[9];
    k.text=words[10];
    l.text=words[11];
}

```

8 farklı buton için 8 adet fonksiyon bulunuyor hepsinde isimler dışında kodlar aynı durumda. Bu kısımda seçilen sorunun cevabı bu basılan buton ise buton kendini yok ediyor(Destroy).

Değil ise kullanıcının süresinden azalma yaşıyor.

```
public void aCEVAP(){
    if(a.text==words[index]){
        words = words.Where(e => e != a.text).ToArray();
        Destroy(A);
        wordsa = wordsa.Where(e => e != chosenwords).ToArray();
        index=Random.Range(0,words.Length);
        chosenwords=wordsa[index];
        soru.text=chosenwords;
    }else{
        time-=Eksipuan;
    }
}

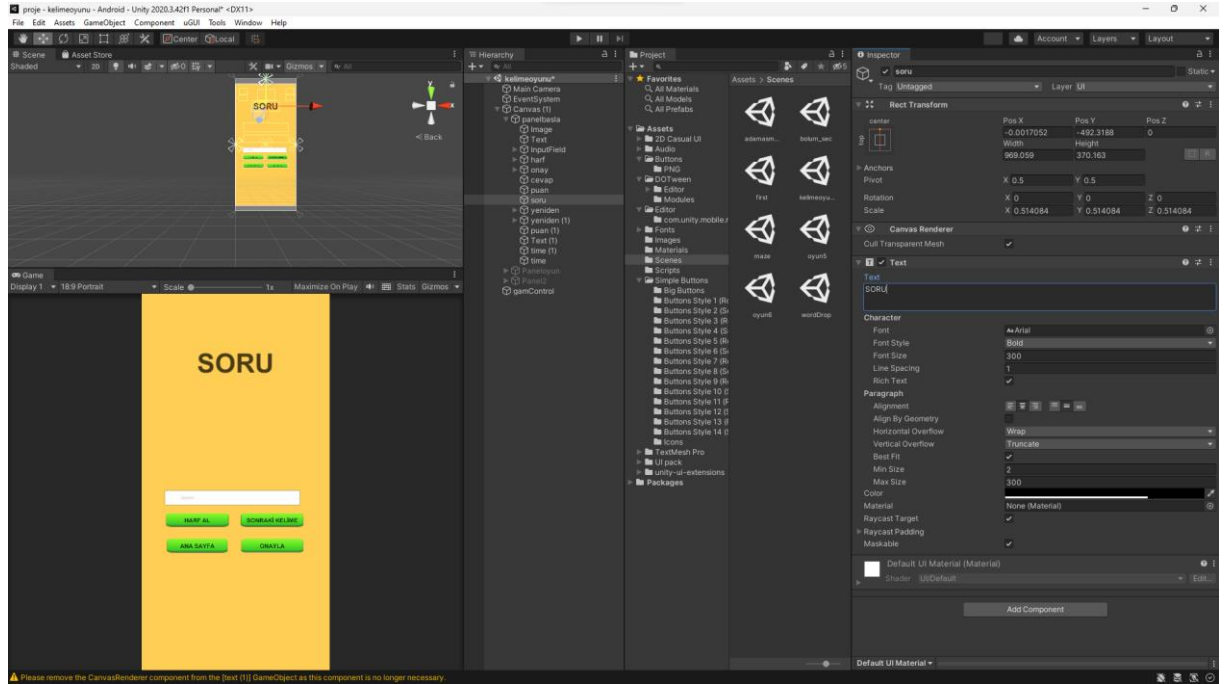
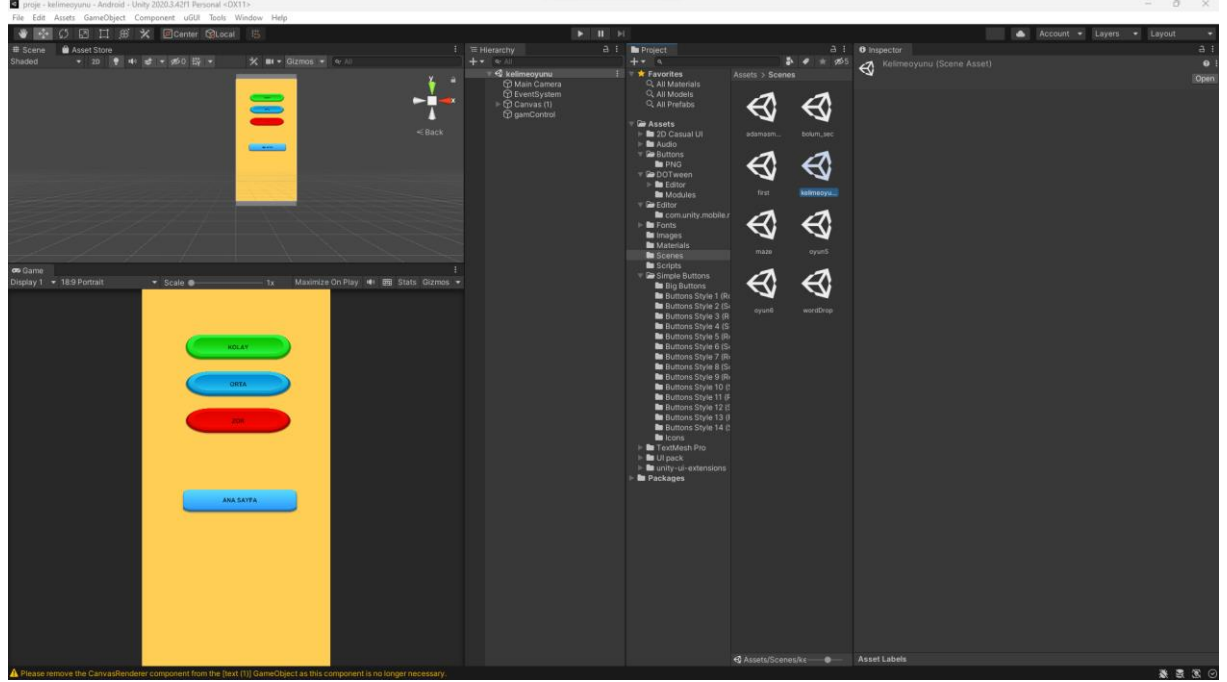
public void bCEVAP() {
    if(b.text==words[index]){
        words = words.Where(e => e != b.text).ToArray();
        Destroy(B);
        wordsa = wordsa.Where(e => e != chosenwords).ToArray();
        index=Random.Range(0,words.Length);
        chosenwords=wordsa[index];
        soru.text=chosenwords;
    }else{
        time-=Eksipuan;
    }
}

public void cCEVAP() {
    if(c.text==words[index]){
        words = words.Where(e => e != c.text).ToArray();
        Destroy(C);
        wordsa = wordsa.Where(e => e != chosenwords).ToArray();
        index=Random.Range(0,words.Length);
        chosenwords=wordsa[index];
        soru.text=chosenwords;
    }else{
        time-=Eksipuan;
    }
    Debug.Log(wordsa);
}
```

KELİME OYUNU

Televizyon programlarında aşına olduğumuz kelime oyununun bizim kelimelerimize uyarlanmış versiyonu bulunuyor karşımızda.

İlk olarak zorluk ekranı ardından ana oyun ekranı tasarlanıyor butonlar ekleniyor gerekli metin kutuları ekleniyor ve kodlama kısmına geçiş yapılıyor.



Gerekli tanımlamalar, seçilen bölüme uygun kelimelerin getirilmesi.

Start, Update, finish, sendData, kolay, orta, zor fonksiyonlarının tanımlanması.

```
9 public class kelimeoyunu : MonoBehaviour
10 {
11     private string[] words=bolumsec.words;
12     private string[] words2=bolumsec.words4;
13     private string chosenword;
14     private string stockword;
15     private string soru;
16     public Text timet,time2;
17     public Text stock;
18     public Text puan;
19     public Text sonuc;
20     public Text soru;
21     public InputField key;
22     private float time,time2;
23     private int rnd;
24     private int puan=1;
25     private char c;
26     private int[] index={};
27     private int inde=0;
28     private bool gameEnd=false;
29     public Text toplamPua;
30     private int top;
31     private int inde;
32     private bool zorluk=false;
33     private bool start=false;
34     private string id;
35     private int temp;
36     public GameObject panel;
37     public Text paneltext;
38
39     void Start()
40     {
41         id=login.id;
42         inde=Random.Range(0,words.Length);
43         chosenword=words[inde];
44         soru=words2[inde];
45
46         for(int i = 0; i < chosenword.Length; i++){
47             stockword+=" ";
48         }
49         stock.text=stockword;
50         puan=stockword.Length*10;
51     }
52     public void finish(){
53         StartCoroutine(sendData());
54     }
55     IEnumerator sendData()
56     {
57         WWWForm form = new WWWForm();
58         form.AddField("unity", "data");
59         form.AddField("id", id);
60         form.AddField("game", "kelime oyunu");
61         form.AddField("time", time.ToString());
62         form.AddField("point", puan.ToString());
63
64         using (UnityWebRequest www = UnityWebRequest.Post(www.url, form))
65         {
66             yield return www.SendWebRequest();
67
68             if (www.result != UnityWebRequest.Result.Success)
69             {
70                 Debug.Log(www.error);
71             }
72             else
73             {
74                 Debug.Log("Form upload complete!");
75             }
76         }
77     }
78 }
79
80
```



```

void Update()
{
    if(start){
        soru.text=sorua;
        stock.text=stockword;
        toplamPua.text=top.ToString();
        puans.text=puan.ToString();
        if(puan<=0 || time2<=0 )
        {
            gameEnd=true;
            panel.SetActive(true);
            paneltext.text=chosenword;
        }
        else{
            Debug.Log(Time.deltaTime);
            Time.timeScale=1;
            time +=Time.deltaTime;
            timet.text = time.ToString();
        }

        if(zorluk && start && !gameEnd)
        {
            time2-=Time.deltaTime;
            timet2.text=time2.ToString();
        }
    }
}

```

```

public void kolay(){
    start=true;
    zorluk=true;
    time2=60;
}

public void orta(){
    zorluk=true;
    time2=30;
    start=true;
}

public void zor(){
    zorluk=true;
    time2=15;
    start=true;
}

```

Bu fonksiyonlar dışında oyunumuzun mekaniksel olarak çalışabilmesi için gerekli 3 adet daha fonksiyon yazdık bunlar;

getGues, getLetter, next.

getGues kullanıcının yaptığı tahminin doğru olup olmadığını kontrol ediyor.

getLetter kullanıcı harf istediğinde rasgele bir noktadan kullanıcıya harf veriyor ama alabileceği en yüksek puandan harf sayısı kadar puan düşüyor.

Next bütün oyunu sıfırlayıp yeni soru üretiyor.

```

1
2     public void getLetter(){
3     StringBuilder sb = new StringBuilder(stockword);
4     while(true){
5         rnd=Random.Range(0,chosenword.Length);
6         if(sb[rnd]=='_'){
7             break;
8         }
9     }
10
11 }

```

```

74
75     public void getGues(){
76
77         if(key.text.ToUpper() == chosenword.ToUpper())
78         {
79             stockword=chosenword.ToUpper();
80             sonuc.text="CEVAP DOGRU";
81             top+=puan;
82             key.text="";
83         }else{
84             sonuc.text="CEVAP YANLIŞ";
85             puan-=10;
86             key.text="";
87         }
88     }
89
90 }

```

```

public void next(){
    stockword="";

    while(temp==inde)
        temp=Random.Range(0,words.Length);

    inde=temp;

    chosenword=words[inde];
    sorua=words2[inde];
    soru.text=sorua;
    for(int i = 0; i < chosenword.Length; i++){
        stockword+="_";
    }
    stock.text=stockword;
    puan=stockword.Length*10;
}

```

HARF YAĞMURU

Gerekli tasarımları yaptığımız kısım.

Bu oyunumuzda 3 adet fiziksel obje olan duvar ve 1 adet fiziksel obje olan elma kullandık

Yine oyun başında zorluk ekranı bizi karşılıyor ardından oyun ekranımıza geçiş yapıyoruz.

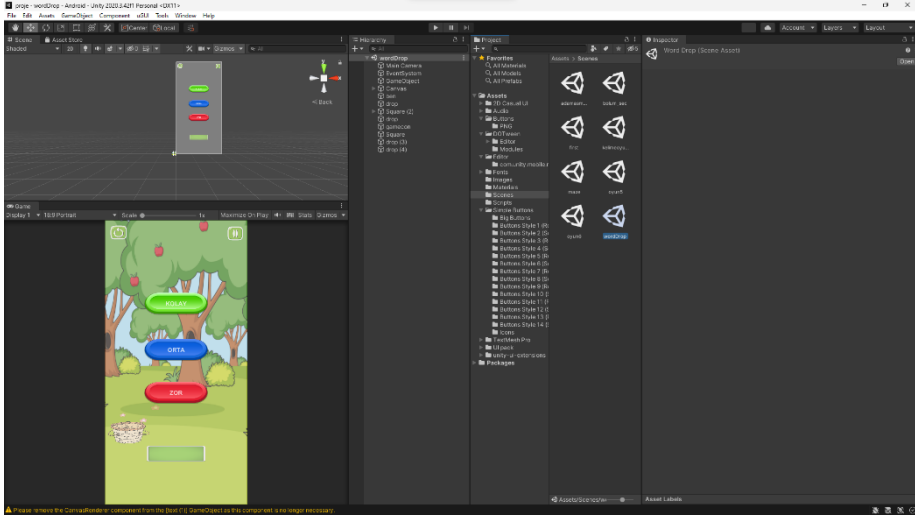
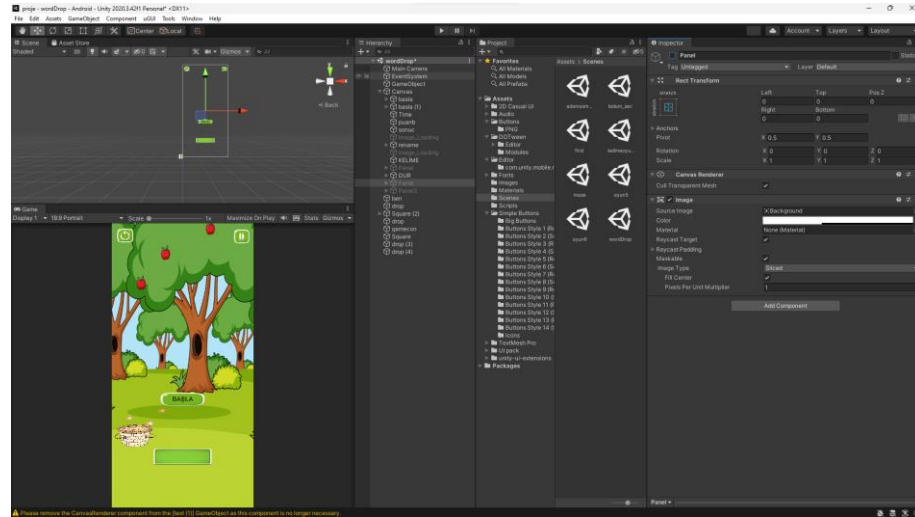
Elmamız aslında ekranın dışında bir noktada bir duvar üzerinde sabit durumda duruyor.

Bu elmamız düşen elmalarımızın orijinal hali.

Düşen her elma bu elmanın kopyası durumunda.

Yer çekimi olduğu için elmayı duvar üzerinde koymak zorunda kaldık.

Bu oyunumuzda ekstra olarak durdurma butonumuz da bulunuyor.



Gerekli tanımlamalar, seçilen bölüme uygun kelimelerin getirilmesi.

Start, Update, finish, sendData, kolay, orta, zor fonksiyonlarının tanımlanması.

```
11 public class worddrop : MonoBehaviour
12 {
13
14     public GameObject item;
15     float time;
16     float time2;
17     float time3;
18     public Text cevap,time3text,sonuc,puantext,kelime;
19     private string[] words=bolumsec.words;
20     private string[] words2=bolumsec.wordsA;
21     private char[] charlist;
22     private string chosenWord;
23     private int puan;
24     private Rigidbody2D rg;
25     float timeDelay;
26     private string name1;
27     private bool startf;
28     private string id;
29     private int index;
30     GameObject myGameObject;
31     public GameObject panel;
32     TextMeshPro myText ;
33     private int temp;
34
35     void Start()
36     {
37         id=login.id;
38         Time.timeScale = 0;
39         time = 0f;
40         time2 = 0f;
41         timeDelay = 1f;
42         index=Random.Range(0,words.Length);
43         chosenWord=words[index];
44
45         rg= GetComponent<Rigidbody2D>();
46
47     }
```

```

IEnumerator sendData()
{
    WWWForm form = new WWWForm();
    form.AddField("unity", "data");
    form.AddField("id", id);
    form.AddField("game", "word drop");
    form.AddField("time", time2.ToString());
    form.AddField("point", puan.ToString());

    using (UnityWebRequest www = UnityWebRequest.Post(' ', form))
    {
        yield return www.SendWebRequest();

        if (www.result != UnityWebRequest.Result.Success)
        {
            Debug.Log(www.error);
        }
        else
        {
            Debug.Log("Form upload complete!");
        }
    }
}

public void finish(){
    StartCoroutine(sendData());
}

```

```

public void kolay(){
    time3=120;
    Time.timeScale = 1;
}
public void orta(){
    time3=90;
    Time.timeScale = 1;
}
public void zor(){
    time3=60;
    Time.timeScale = 1;
}
public void start(){
    startf = true;
    Time.timeScale = 1;
}

```

Update kısmınızda Gökyüzünde rasgele elma oluşturma ve bunları rasgele harflerde rasgele pozisyonlarda oluşturmak için ekstra kodlarımız bulunuyor.

```

void Update()
{
    if(startf){
        kelime.text=words2[index];
    }

    if(cevap.text!=chosenWord){

        time3 -=Time.deltaTime;
        time3text.text = time3.ToString();
        puantext.text=puan.ToString();

        if(time3<=0){
            sonuc.text="SÜRENDİZ BİTTİ";
            Time.timeScale = 0;
            time3text.text="0";
            panel.SetActive(true);
        }

        time2+=Time.deltaTime;
        time = time + 1f * Time.deltaTime;
        if (time >= timeDelay) {
            time = 0f;
            Vector3 randomSpawnPosition = new Vector3(Random.Range(-12,-7),10,0);

            GameObject clone=Instantiate(item, randomSpawnPosition,Quaternion.identity);
            name1=chosenWord[Random.Range(0,chosenWord.Length)].ToString();
            clone.name=name1;
            clone.GetComponentInChildren<TextMeshPro>().text=name1;
        }

    }else{
        sonuc.text="TEBRİKLER";
        puan+=10;
        puantext.text=puan.ToString();
        next();
    }
}

```

Bu fonksiyonumuz elmalarımızın sepete dokunduğunu anlamak ve o anda elmayı yok etmek için kullandığımız fonksiyon elmamızın üzerindeki harfi ekrana basmamızı da bu fonksiyon sağlıyor.

```

void OnCollisionEnter2D(Collision2D cls){

    for(int i=0; i<chosenWord.Length; i++){
        Debug.Log(cls.gameObject.name);
        if(cls.gameObject.name.Equals(chosenWord[i].ToString())){
            Debug.Log(cls.gameObject.name);
            cevap.text+=chosenWord[i];
            Destroy(cls.gameObject);
            break;
        }
    }
}

```

Sepete yanlış elma aldığımızda sepeti temizlemek için bastığımız butonumuzun çalıştırdığı fonksiyon.

```

public void rename(){

    cevap.text="";

}

```

Yeni kelime üretmesi için yazdığımız fonksiyon.

```

57
58     public void next(){
59
60     while(index==temp)
61     temp = Random.Range(0,words.Length);
62     index = temp;
63     chosenWord=words[index];
64     sonuc.text="";
65     cevap.text="";
66
67     }
68

```

Oyunu durdurma butonuna bastığımızda zaman akış hızını 0 ya da başlattığımızda 1 yapan fonksiyonlarımız.

```

public void dur(){
    Time.timeScale = 0;
}
public void devam(){
    Time.timeScale = 1;
}

```

Destroyer.cs isimli scriptimiz elmaların sepete değmemesi ve ekrandan çıkması halinde cihazımız hafızasında yer kaplamaması için elmaları yok etmesi için yazıldı.

```

5  public class destroyer : MonoBehaviour
6  {private Rigidbody rg;
7      // Start is called before the first frame update
8      void Start()
9      {
10         rg= GetComponent<Rigidbody>();
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16
17     }
18
19     void OnCollisionEnter2D(Collision2D cls){
20
21         Destroy(cls.gameObject);
22     }
23
24 }
25

```

Drag.cs isimli scriptimiz sepetimizi parmağımızla hareket etmesi için gerekli kodları yazmak için kullanıldı.


```
5 public class drag : MonoBehaviour
6 {
7     bool canMove;
8     bool dragging;
9     Collider2D collider;
10    void Start()
11    {
12        collider = GetComponent<Collider2D>();
13        canMove = false;
14        dragging = false;
15    }
16 }
17
18
19 void Update()
20 {
21     Vector2 mousePos = Camera.main.ScreenToWorldPoint(Input.mousePosition);
22
23     if (Input.GetMouseButtonDown(0))
24     {
25         if (collider == Physics2D.OverlapPoint(mousePos))
26         {
27             canMove = true;
28         }
29         else
30         {
31             canMove = false;
32         }
33         if (canMove)
34         {
35             dragging = true;
36         }
37     }
38
39     if (dragging)
40     {
41         this.transform.position = mousePos;
42     }
43     if (Input.GetMouseButtonUp(0))
44     {
45         canMove = false;
46         dragging = false;
47     }
48 }
49 }
```