

Harry Potter: Memory Master

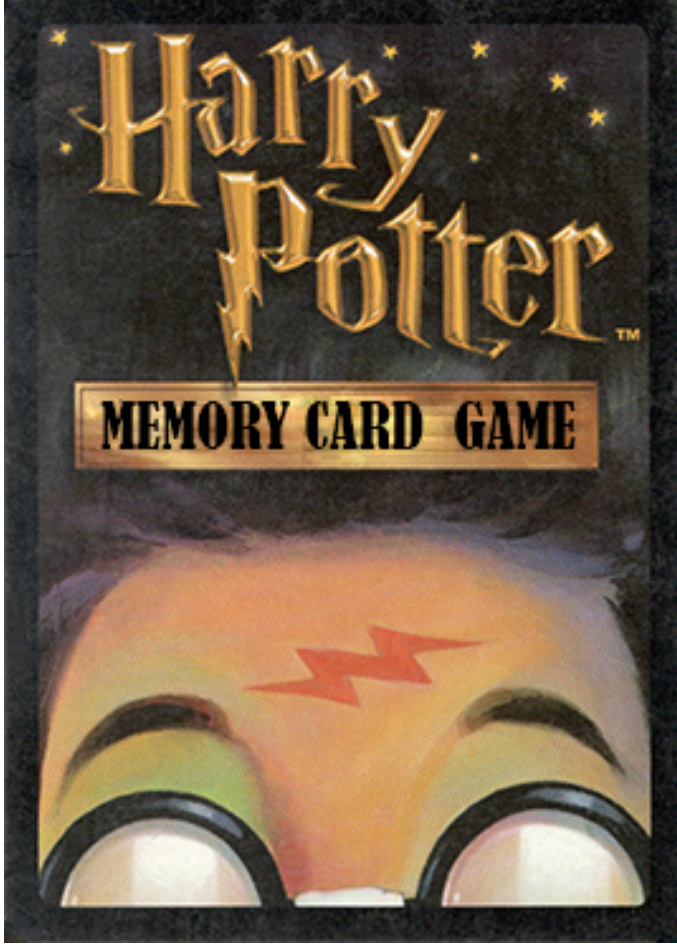


Fig. 1.

I. ÖZET

Bu rapor Yazılım Laboratuvarı I Dersinin 3.Projesini açıklamak ve sunumunu gerçekleştirmek amacıyla oluşturulmuştur.Raporda projenin tanımı , yapım aşaması ,kullanılan araç ve yöntemler , kod parçacıkları vb. bulunmaktadır. Proje aşamasında yararlanılan kaynaklar raporun son bölümünde bulunmaktadır.

II. AMAÇ

Bu proje ile Android uygulama ve bulut bilişim teknolojilerinin kullanılması amaçlanmaktadır. Projede belirtilen süre içinde zorluk seviyesine göre kartları doğru eşleştirmek gerekmektedir. Projeyi Android uygulama olarak geliştirmemiz beklenmektedir.

Kocaeli Üniversitesi.

III. GİRİŞ

Bu projede istenilen kotlin veya java ile android mobil geliştirme yapmamız istenmektedir.

A. Başlamadan önce

Projeye başlamadan projeyi KOTLİN ve ANDROID STUDIO ile yapmaya karar verdik.Database olarak ise FIREBASE kullanmaya karar verdik. ve kodlamaya girmeden önce mobil uygulama ve mobil oyunlar hakkında çeşitli makalelerden detaylı araştırma yaptık.Bu projede nasıl kullanmamız gerektiğini belirledikten sonra projeye başladık.

B. Başlangıç

Projede istenilen şekilde kullanıcı giriş ekranı yapmaya başladık.Kullanıcı kayıt etmek için firebasein hazır fonksiyonu olan auth.signInWithEmailAndPassword(email, password) fonksiyonunu kullandık bu sayede şifreleri güvenli bir şekilde tuttuk.

C. ilerleyiş

Daha sonra şifremizi güvenli bir şekilde değiştirmek amacıyla yine firebasein bize sunduğu hizmet olan şifremi unuttum kolaylığını projemize ekledik.Daha sonra oyunu yapmaya başladık.

IV. SONUÇ

Firebase firestore databaseini kullanarak verilerimizi ev,foto,evpuanı vs. olarak giriş yaptık.Daha sonra onları çekmemiz gerektiği için fun InitMemoryCard (bkznz. Fig. 7.) isminde bir fonksiyon yazmamız gerekti.Bu fonksiyonda verileri tek tek aldık ve daha önceden oluşturduğumuz MemoryCardData isimli data classlarımızı doldurduk.Her eve ait 4 adet mutable dizi oluşturduk. ve bu data kartlarımızı bu dizilere ekledik.Her oyun farklı kart gelmesi için dizileri karıştırmak için (.shuffle()) kullandık.Daha sonra oyunun zorluğuna göre bu dizilerden yeterli sayıda büyücüğü alıp geçici birer diziye gönderdik.Oyunda her karttan iki tane olacağından dolayı.geçici dizileri kendine bir daha ekledik ve bu dizileri en son bir dizide toplayıp shuffle attık ve oyunun algoritmasını yazdık.

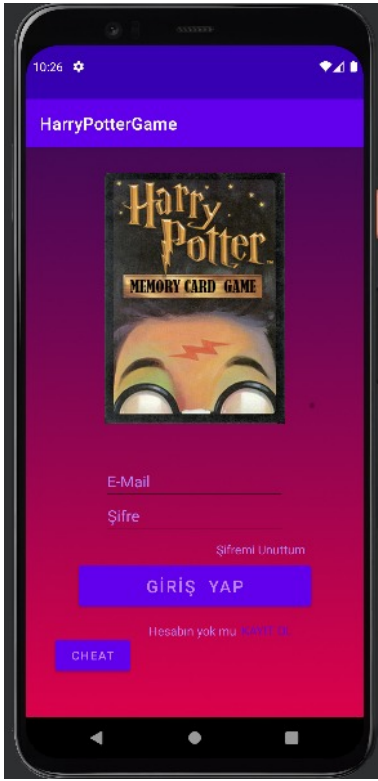


Fig. 2. Giriş Ekranı İçin Oluşturduğumuz Ekran



Fig. 4. 6x6 tek kişilik ekrandan bir görüntü

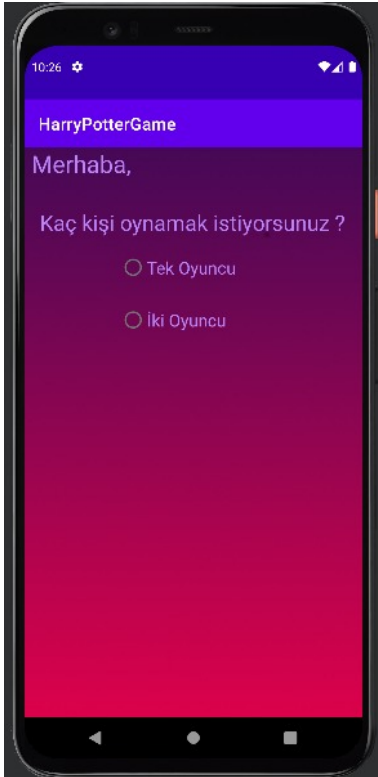


Fig. 3. Kaç kişili ve oyun zorluğunu seçtiğimiz radio buttonlu ekran

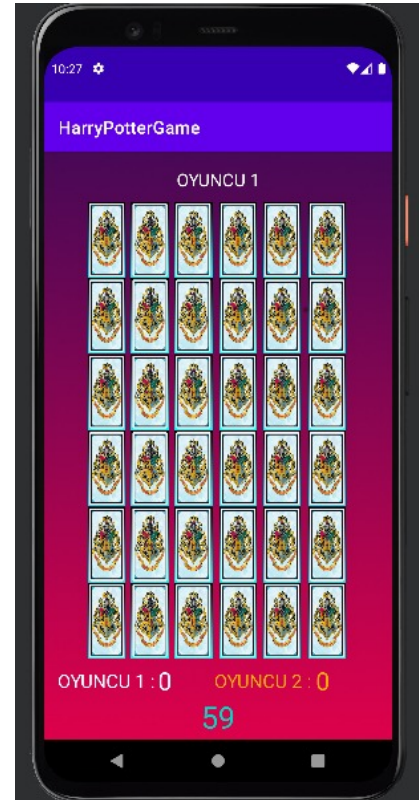


Fig. 5. 6x6 çift kişilik ekrandan bir görüntü


```
if(!this::mediaPlayer.isInitialized){
    mediaPlayer = MediaPlayer.create( context: this, R.raw.prologue)
    mediaPlayer.start()
}

if(!this::mediaPlayer1.isInitialized){
    mediaPlayer1 = MediaPlayer.create( context: this, R.raw.shocked)
}

if(!this::mediaPlayer2.isInitialized){
    mediaPlayer2 = MediaPlayer.create( context: this, R.raw.congratulations)
}

if(!this::mediaPlayer3.isInitialized){
    mediaPlayer3 = MediaPlayer.create( context: this, R.raw.dogrulkart)
}
}
```

Fig. 6. Sesleri çaldırmak için tanımladığımız mediaplayerlar

```
fun InitMemoryCard(houseName: String, wizardName: String, memoryCard: MemoryCardData) {

    val db = FirebaseFirestore.getInstance()
    val docRef = db.collection(houseName).document(wizardName)

    docRef.get()
        .addOnSuccessListener { document ->
            if (document != null) {

                if (document.data == null) {
                    Log.e(tag: "bosbos", msg: "DATA IS NULLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL")
                }

                val houseName = document.data!!["ev"].toString()
                val wizardPoint = document.data!!["pun"] as Long
                val housePoint = document.data!!["syuan"] as Long
                val photoBase64 = document.data!!["foto"].toString()

                memoryCard.isFlipped = false
                memoryCard.isMatched = false
                memoryCard.houseName = houseName
                memoryCard.wizardPoint = wizardPoint
                memoryCard.photoStringBase64 = photoBase64
                memoryCard.housePoint = housePoint

            } else {
                Log.d(tag: "TAG", msg: "No such document")
            }
        }
        .addOnFailureListener { exception ->
            Log.d(tag: "TAG", msg: "get failed with ", exception)
        }
}
```

Fig. 7. Verileri firestore'dan çekmek için yazdığımız fonksiyon

```
InitMemoryCard( houseName: "gryffindor", wizardName: "AlbusDumbledore", MemoryCardData1)
InitMemoryCard( houseName: "gryffindor", wizardName: "ArthurWeasley", MemoryCardData2)
InitMemoryCard( houseName: "gryffindor", wizardName: "HarryPotter", MemoryCardData3)
InitMemoryCard( houseName: "gryffindor", wizardName: "HermioneGranger", MemoryCardData4)
InitMemoryCard( houseName: "gryffindor", wizardName: "LilyPotter", MemoryCardData5)
InitMemoryCard( houseName: "gryffindor", wizardName: "MinervaMcGonagall", MemoryCardData6)
InitMemoryCard( houseName: "gryffindor", wizardName: "PeterPettigrew", MemoryCardData7)
InitMemoryCard( houseName: "gryffindor", wizardName: "RemusLupin", MemoryCardData8)
InitMemoryCard( houseName: "gryffindor", wizardName: "RubeusHagrid", MemoryCardData9)
InitMemoryCard( houseName: "gryffindor", wizardName: "SiriusBlack", MemoryCardData10)

InitMemoryCard( houseName: "slytherin", wizardName: "AndromedaTonks", MemoryCardData11)
InitMemoryCard( houseName: "slytherin", wizardName: "BellatrixLestrange", MemoryCardData12)
InitMemoryCard( houseName: "slytherin", wizardName: "DoloresUmbridge", MemoryCardData13)
InitMemoryCard( houseName: "slytherin", wizardName: "DracoMalfoy", MemoryCardData14)
InitMemoryCard( houseName: "slytherin", wizardName: "EvanRosier", MemoryCardData15)
InitMemoryCard( houseName: "slytherin", wizardName: "HoraceSlyughorn", MemoryCardData16)
InitMemoryCard( houseName: "slytherin", wizardName: "LetaLestrange", MemoryCardData17)
InitMemoryCard( houseName: "slytherin", wizardName: "LuciusMalfoy", MemoryCardData18)
InitMemoryCard( houseName: "slytherin", wizardName: "MarcellusMalfoy", MemoryCardData19)
InitMemoryCard( houseName: "slytherin", wizardName: "Severus Snape", MemoryCardData20)
InitMemoryCard( houseName: "slytherin", wizardName: "TomRiddle", MemoryCardData21)
```

Fig. 8. InıtMemoryCard fonksiyonu ile memorycard datalarımızın içini dolduruyoruz

```
fun ReturnBitmap(base64String: String): Bitmap {  
  
    val imageBytes = Base64.decode(base64String, Base64.DEFAULT)  
    val decodedImage = BitmapFactory.decodeByteArray(imageBytes, 0, imageBytes.size)  
    return decodedImage  
}
```

Fig. 9. Base 64 stringini fotoğrafa çeviren fonksiyonumuz

```

HashMap.put(image1, allTemplList[0])
HashMap.put(image2, allTemplList[1])
HashMap.put(image3, allTemplList[2])
HashMap.put(image4, allTemplList[3])
HashMap.put(image5, allTemplList[4])
HashMap.put(image6, allTemplList[5])
HashMap.put(image7, allTemplList[6])
HashMap.put(image8, allTemplList[7])
HashMap.put(image9, allTemplList[8])
HashMap.put(image10, allTemplList[9])
HashMap.put(image11, allTemplList[10])
HashMap.put(image12, allTemplList[11])
HashMap.put(image13, allTemplList[12])
HashMap.put(image14, allTemplList[13])
HashMap.put(image15, allTemplList[14])
HashMap.put(image16, allTemplList[15])
HashMap.put(image17, allTemplList[16])
HashMap.put(image18, allTemplList[17])
HashMap.put(image19, allTemplList[18])
HashMap.put(image20, allTemplList[19])
HashMap.put(image21, allTemplList[20])
HashMap.put(image22, allTemplList[21])
HashMap.put(image23, allTemplList[22])
HashMap.put(image24, allTemplList[23])
HashMap.put(image25, allTemplList[24])
HashMap.put(image26, allTemplList[25])
HashMap.put(image27, allTemplList[26])
HashMap.put(image28, allTemplList[27])
HashMap.put(image29, allTemplList[28])
HashMap.put(image30, allTemplList[29])
HashMap.put(image31, allTemplList[30])
HashMap.put(image32, allTemplList[31])
HashMap.put(image33, allTemplList[32])
HashMap.put(image34, allTemplList[33])
HashMap.put(image35, allTemplList[34])
HashMap.put(image36, allTemplList[35])

```

Fig. 10. hashmap kullanarak 6x6 oyun zorluğunun fotoğraf ve kart bilgileri eşleştirmesi

```
override fun onStart() {  
    super.onStart()  
  
    val timer = object: CountdownTimer( millisFuture: 46000, countDownInterval: 1000) {  
        override fun onTick(millisUntilFinished: Long) {  
  
            pastTime = 46000F/1000F - millisUntilFinished/1000F  
            remainingTime = millisUntilFinished/1000F  
  
            timeText.text = ""+ remainingTime  
  
        }  
    }  
}
```

Fig. 11. Geçen ve kalan süreyi tuttuğumuz kod parçasığı

```
fun ResetGame(imageViewList : List<ImageView>){  
  
    for(imageView in imageViewList){  
  
        imageView.setImageResource(R.drawable.arkason)  
        firstCard = null  
        secondCard = null  
        isTwoCardsFlipped = false  
        isOneCardFlipped = false  
  
    }  
}
```

Fig. 12. Eğer yanlış yapılırsa kartların arka yüzünü gösteren fun ResetGame() fonksiyonumuz

Projeyi Hazırlayanlar:

Berke Kara-190202086

Mehmet ALi Mehteroğlu-200202014

- <https://www.youtube.com/watch?v=U4Wtjewy7EYt=3622sabchannel=RahulPandey>
- <https://www.youtube.com/watch?v=QTScRS9MhqA>
- <https://console.firebase.google.com/?pli=1>
- <https://www.youtube.com/watch?v=7vWoi8j5vL4abchannel=AdobeinaMinute>
- <https://stackoverflow.com/>