

# Yazılım Lab II

## Proje 2

Gökhan Çobanoğlu  
200201015  
Bilgisayar Mühendisliği

Bilal Cihangir Erdem  
200201076  
Bilgisayar Mühendisliği

### I. ÖZET

Bu projede bizden mobil programlama ile bir kelime oyunu yapmamız beklenmektedir.

Oyun içerisinde bir kelime havuzunun olması ve bu kelime havuzunda olan kelimelerle oyunun arayüzünde olan kelimeler eşleştirilerek bir puan sistemi yapılması istenmiştir.

Projede yazılım dili olarak Kotlin ve Android Studio kullanılmıştır.

Kelime havuzu olarak kelimelistesi.txt adında bir dosya kullanılmıştır. Kelime sayısı 50.000 den fazla olacak şekilde ayarlanmıştır. Havuzdaki kelimeler en az 3 harften oluşmuştur. Ayrıca kelimeler anlamlı kelimelerdir.Özel isim yer isimleri gibi kelimeler yoktur.

### II. GİRİŞ

Oyunun arayüzü 8 sütun 10 satırdan oluşmaktadır.Bu alan üzerinde kelime seçilerek anlamlı kelimeler oluşturulacaktır.

Oyun ilk başladığında yukarıdan aşağıya doğru 3 satır tamamen dolu olacak şekilde rastgele harfler indirilecek ve oyun başlanacaktır.Her bir matris alanına bir harf gelecek şekilde tasarlanılmıştır. Harfler yukarıdan aşağı yönde düşmektedir. En üst kısımdan aşağıya doğru birim birim tabana kadar inmesi beklenmektedir.

Kullanıcının arayüzdeki kelimeleri dokunarak seçerek anlamlı kelime bulması ve bu kelimeler ile puan alması amaçlanmıştır.

Her bir harfin puanı bize verilen pdfte verilmiştir.Buna göre düzenleme yapmamız istenmiştir. Aynı zamanda sesli ve sessiz harflerin belli oranda gelmesi istenmiştir.

Oyun başlangıcından sonra belirlenen süre aralıklarında yukarıdan aşağıya doğru rastgele herhangi bir sütundan harf düşmesi beklenmektedir. Harfler sesli veya sessiz olmak üzere karışık bir şekilde olması gereklidir.

Harf düşmesi işlemi ilk başta 5 saniyede bir gerçekleşecektir. Bu süre puanın her 100 ve katlarına ulaştığında bir saniye azalacaktır. Azalma işlemi 1 saniye ulaşıncaya kadar devam edecektir. 1 saniye ulaştığında oyunun devamında bu süre ile harf düşmesi beklenmektedir.

Harf düşmesi işlemi ilk başta 5 saniyede bir gerçekleşecektir. Bu süre puanın her 100 ve katlarına ulaştığında bir saniye azalacaktır. Azalma işlemi 1 saniye ulaşıncaya kadar devam edecektir. 1 saniye ulaştığında oyunun devamında bu süre ile harf düşmesi beklenmektedir.

Oluşturulan kelimenin yanlış olma durumu kayıt altına alınacaktır. 3 kez yanlış girilmesi durumunda tüm sütunlardan

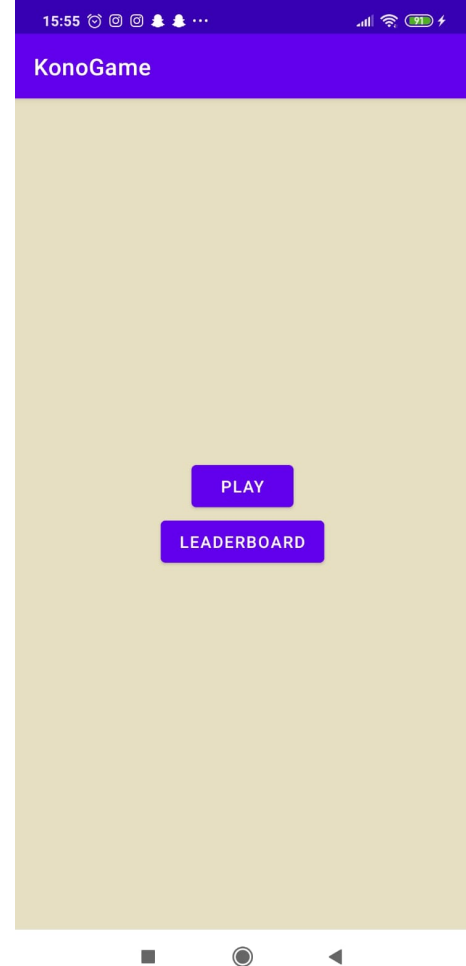


Fig. 1: Main Menu

harfler düşecektir ve hatalı kelime sayısı sıfırlanacaktır. Her üç yanlış kelime girilmesinde aynı işlem gerçekleşecektir.

Oyun bir sütunun yukarıya doğru harfler ile tamamen dolması ile sonlanacaktır. Sonlandırdıktan sonra puanı güncel puan listesine eklenecektir. Güncel puan listesi sıralı bir şekilde olmalıdır. En üstte en yüksek puandan en düşük puana kadar sıralanması beklenmektedir.

### III. YÖNTEM

Bu proje Kotlin dili ve Android Studio kullanılarak yapılmıştır. Projeye oyun alanından önce bir menü yapılarak

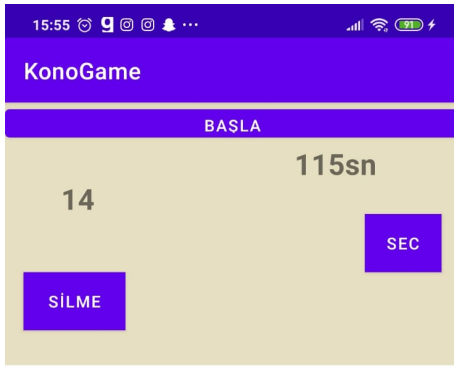


Fig. 2: Main Activity

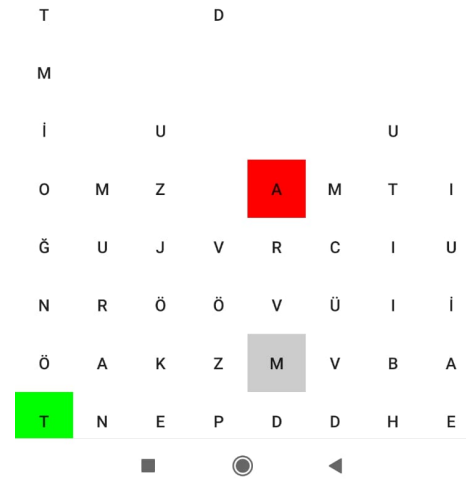
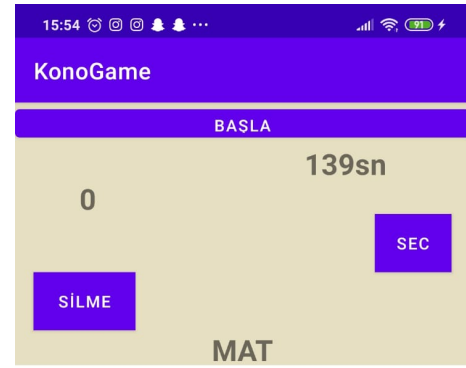


Fig. 3: Main Activity

başlanmıştır. Bu menüde play ve leaderboard adında iki buton bulunmaktadır. Play butonuna basınca oyunun olduğu kısma bizi götürecektir. Leaderboard ile güncel puan listesi en yüksekten en düşüğe görünecektir.

Play butonuna bastıktan sonra gelen oyun alanını tasarlayarak oyuna başladık. Satır ve sütunları togglebutton ile doldurarak 8 sütun 10 satırlık bir oyun alanı oluşturduk. Oyundaki önemli class ve fonksiyonları aşağıda açıklamalarıyla bulunmaktadır.

Main Activity classı oluşturduk. MainActivity sınıfı, AppCompatActivity sınıfını genişletir. Bu sınıf, ActionBar desteği kullanan aktiviteler için bir temel sınıftır.

onCreate yöntemi içinde, setContentView yöntemi çağrılır ve aktivitenin düzeni, activitymain.xml adlı XML dosyasında tanımlanır.

Kullanıcı, düzen dosyasında belirtilen bir düğmeye tıkladığında, startgame() yöntemi çağrılır. Bu yöntem, başlatılacak etkinliğin sınıfını (GameActivity) belirten bir yeni Intent oluşturur ve startActivity() yöntemi kullanarak etkinliği başlatır.

Benzer şekilde, kullanıcı başka bir düğmeye tıkladığında

leaderboard() yöntemi çağrılır. Bu yöntem, başlatılacak etkinliğin sınıfını (get) belirten bir yeni Intent oluşturur (ancak gerçek sınıf adıyla değiştirilmelidir) ve startActivity() yöntemi kullanarak etkinliği başlatır.

Genel olarak, bu MainActivity sınıfı kullanıcı arayüzünü yönetir ve diğer aktiviteleri başlatır.

val textfield ile kelime butonlarının işlemi yapılmıştır. Butonlar togglebutton olduğundan 1 veya 0 olmasına göre bu butonların üstündeki harflerle farklı kontroller yapılır. button.isSelected 0 ise harfi seçip kelimeyi textviewin içine ekler butonu seçilmiş yapar ve rengini değiştirir. button.isSelected 1 ise harfi textviewden siler ve butonun seçilmemiş rengine geri döndürür.

singleletterdrop fonksiyonu ile rastgele bir harf düşürme işlemi yapılmıştır. Bu işlev, belirli bir harf grubundan rasgele harfler üretir ve bunları rasgele bir sıradaki geçiş düğmelerine atar.

İlk olarak, "harfler" adlı bir dizge değişkenini bir dizi Türkçe karakterle başlatır. Ardından, 0 ile 65 (dahil) arasında sekiz rasgele tam sayıdan oluşan bir dizi oluşturur.

Ardından, 0 ile 7 (dahil) arasında rastgele bir tamsayı üretir

ve o sırada hangi geçiş düğmesinin olduğunu kontrol eder. Bu düğmenin metni boşsa, "harfler" dizisinden o düğmeden önce oluşturulan rasgele sayıya karşılık gelen karakteri atar. Ayrıca düğmenin textOn ve textOff özelliklerini de aynı karaktere ayarlar.

rowdrop fonksiyonu ise 3 yanlış yapınca sütunların üstüne rastgele harf dolduran fonksiyon. Bir kelime oyunu uygulamasında, kullanıcının karşısına gelen 8 butondan oluşan bir satırda rastgele harfler yerleştirir. Öncelikle "ABCÇDEFGGHHIJKLMNOÖPRSŞTUÜVYZAEİİÖÜÜ" harf kümesi tanımlanır ve daha sonra bu harflerden rastgele 8 adet harf seçilir. Ardından bu harfler, kullanıcının karşısına gelen 8 butona sırayla yerleştirilir.

Bu işlem, kullanıcının yeni kelime oluşturmak için butonlara tıklaması durumunda tekrarlanır ve yeni bir satırda rastgele harfler yerleştirilir

initTimer fonksiyonu ise Timer hazırlama ve başlama fonksiyonudur.

wordkontrol fonksiyonu kelime seçildikten sonra tıklanan kelime kontrol tuşudur. text dosyasından kontrol yapıp ona göre işlem yapar.

removeCharAt fonksiyonu tuşlarla seçilen harfleri geri almak için kullanılan fonksiyondur.

readfromfile fonksiyonu ile external storagede bulunan text dosyasını getirir içindeki skorları bir listeye atar ve return olarak o listeyi döndürür.

writetofile fonksiyonu ise yeni skoru listeye ekler ve o listeyi de text dosyanın üstüne yazar.

Game activity classı: Kod, AppCompatActivity'yi genişleten bir GameActivity sınıfı tanımlar. Birkaç üye değişkenine (flag, selectedword, selectedwordcount ve falseSelection) ve birkaç üye işlevine (singleletterdrop(), rowdrop(), initTimer(), wordkontrol(), dkontrol(), cleanup(), removeCharAt(), totalkontrol() sahiptir. ), reversekontrol(), kontrol() ve satirdoldurma()).

onCreate() işlevi, etkinliğin içerik görünümünü gamescreen adlı bir düzen dosyasına ayarlar.

initTimer() işlevi, bir CountdownTimer nesnesi oluşturur ve onu başlatır. CountdownTimer nesnesinin süresi 190000 milisaniyedir (3 dakika 10 saniye) ve her 1000 milisaniyede (1 saniye) bir tıklanır. CountdownTimer nesnesinin onTick() işlevi içinde, singleletterdrop() ve rowdrop() işlevlerinin ne zaman ve ne sıklıkta çağrılacağını belirleyen birkaç koşullu ifade vardır. Ayrıca, kullanıcının çok fazla yanlış seçim yapıp yapmadığını kontrol eden (wrongSelection != 3) ve bir satır harf bırakmak için rowdrop()'u çağıran bir koşullu ifade vardır. Son olarak, değiştirme düğmelerinden herhangi birine basılıp basılmadığını kontrol eden ve basıldıysa CountdownTimer'ı iptal eden, değiştirme düğmesi metnini temizleyen ve yeni bir etkinlik (Process sınıfının bir örneği) başlatan bir koşullu ifade vardır.

removeCharAt() işlevi, girdi olarak bir dize (str) ve bir dizin (dizin) alır ve verilen dizindeki karakter çıkarılmış olarak dizeyi döndürür.

Process classı: Bu etkinlik, oyun sona erdiğinde oluşturulur ve oyuncunun son puanını görüntüler ve yeni puanla bir liderlik tablosu dosyasını günceller.

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
    fun startGame(view: View) {
        val intent = Intent(packageContext, this@MainActivity, GameActivity::class.java)
        startActivity(intent)
    }
    fun leaderboard(view: View) {
        val intent = Intent(packageContext, this@MainActivity, get::class.java)
        startActivity(intent)
    }
}
```

Fig. 4: Main Activity

```
//Gizli başlatıldığında 5 satır doldurma
if (flag == 0) {
    satirdoldurma()
    satirdoldurma()
    satirdoldurma()
    satirdoldurma()
}

val textField = findViewById<TextView>(R.id.textfield)
//Kelimeleri butonların içine ekler
//Butonları togglebuton olduğundan 1 veya 8 classına göre bu butonların
// Butonları harflerle farklı kontroler yaparlar
//Buton.isSelected == 0 olduğunda harfi seçip kelime textview in
//içine ekler butonu seçilene kadar tekrar edilebilir
//Buton.isSelected == 1 olduğunda harfi textview den çıkar butonu seçilene kadar tekrar edilebilir
buttons.forEachIndexed { index, button } {
```

Fig. 5: Textfield

onCreate() işlevi, etkinlik oluşturulduğunda çağrılır ve içerik görünümünü "oyun sonu" düzenine ayarlar. Ardından, bu aktiviteyi başlatmak için kullanılan niyetten nihai puanı alır ve puanı görüntülemek için bir TextView metnini ayarlar.

readFromFile() işlevi, cihazın harici depolama dizininde bulunan "Leaderboard.txt" adlı dosyanın içeriğini okur. Bir BufferedReader kullanarak dosyanın satırlarını okur ve bunları bir dizge listesi olarak döndürür.

writeToFile() işlevi, puanlar listesine yeni puanı ekleyerek ve bunları dosyaya geri yazarak skor tablosu dosyasını günceller. Önce lider tablosu dosyası için bir Dosya nesnesi oluşturur ve ardından bir for döngüsü kullanarak listenin her satırını dosyaya yazar. Son olarak, writeText() işlevini kullanarak yeni skoru dosyaya yazar. Bir IOException oluşursa, yığın izini yazdırır.

Kodun, harici depolama dizininin var olduğunu ve "Leaderboard.txt" dosyasının önceden oluşturulmuş olduğunu varsaydığını unutmayın. Ayrıca skorun bir string olduğunu varsayar ve "puantext" tuşu ile bir extra olarak Intent'e iletildiğini varsayar.

## IV. DENEYSEL SONUÇLAR

### REFERENCES

- [1] <https://www.youtube.com/watch?v=1CHDASXojNQlist=Wlindex=14>
- [2] <https://stackoverflow.com/questions/tagged/android-studio>
- [3] <https://kotlinlang.org/docs/home.html>
- [4] <https://www.youtube.com/watch?v=MyEfGHie5BE>
- [5] <https://www.youtube.com/watch?v=Yi8mxXsroJ4>
- [6] <https://developer.android.com/codelabs/build-your-first-android-app0>
- [7] <https://developer.android.com/studio/run>

```

//Rastgele harfleri sütunların üstünden en üstüne koyan fonksiyon
fun singleletterdrop(){...}
// 3 yanlış yapılıncaya kadar rastgele harf dolduran fonksiyon
fun rowdrop(){...}
//Timer hazırlama ve başlatma
//Bu fonksiyon aynı timer bittikten sonra veya herhangi sütun
//dolduğunda oyunun bitimini gösteren aktiviteyi çağırır
//Rastgele harfleri oluşturma fonksiyonu onlick fonksiyonunun
//içinde oluşturulur böylece harfler kayarak iner
fun initTimer(view: View){...}
//Kelime seçildikten sonra tıklanan kelime kontrol tuşu
//text dosyasından kontrol yapıp doğru çıkarsa ona göre işlem yapan
fun wordkontrol(view: View) {...}
fun dkontrol() {...}
//Kelime yazısını temizleme tuşu
fun cleanup(view: View) {...}

```

Fig. 6: Bazı Fonksiyonlar

```

//External Storageda bulunan text dosyasını getirir içindeki skorları bir
// listeye atar ve return olarak o listeyi döndürür
fun readFromFile(): List<String> {...}
//Girdi olarak text dosyasından okunan listeyi ve yeni skoru alır.
// Yeni skoru listeye ekler o listeyi de text dosyasının üstüne yazar
fun writeToFile(fileName: String, lines: List<String>, data:String) {...}

```

Fig. 7: Dosya fonksiyonları

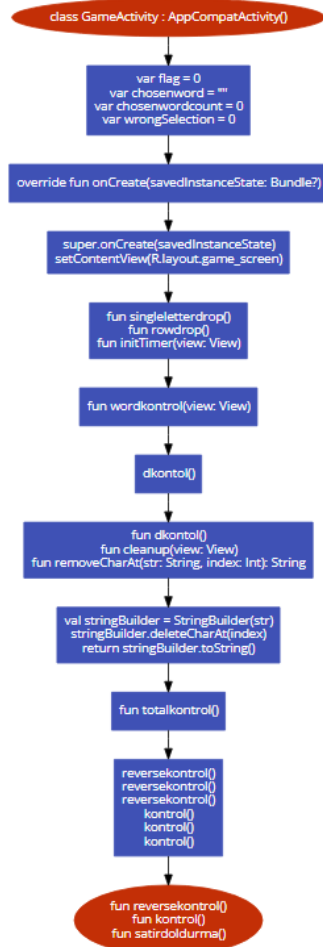


Fig. 8: akış diyagramı

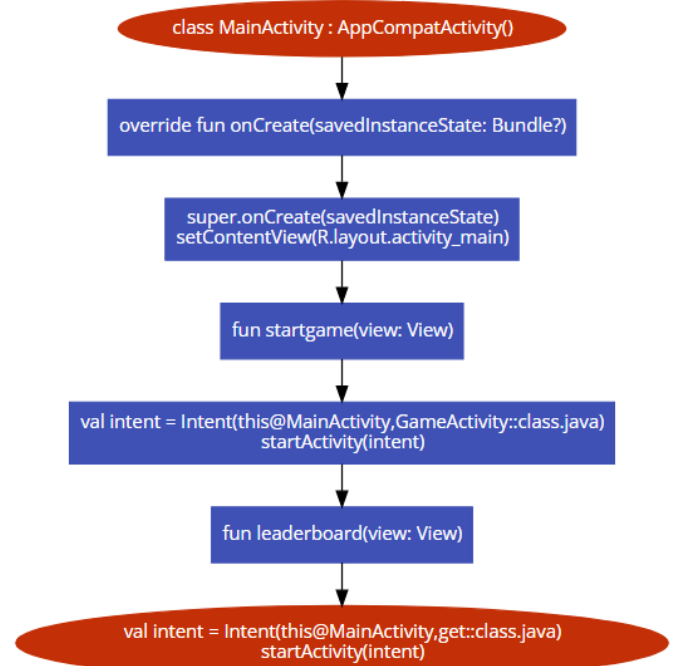


Fig. 9: akış diyagramı 2