

# Yazılım Laboratuvarı II

## Proje II

### Bilgisayar Mühendisliği Bölümü

### Kocaeli Üniversitesi

Furkan Kasap  
210201107

#### I. ÖZET

Bu rapor Yazılım Laboratuvarı 2 dersinin 2. projesi için oluşturulmuştur. Bu proje Kotlin dili kullanılarak geliştirilmiştir. Algoritmayı geliştirme ortamı olarak Android Studio kullanılmıştır.

Belgede akış diyagramı, özet, giriş, yöntem, deneysel sonuçlar gibi projeyi açıklayan başlıklara yer verilmiştir. Belge sonunda projenin sonucu ve projeyi hazırlarken kullanılan kaynaklar bulunmaktadır.

#### II. GİRİŞ

Bu rapor, mobil programlama kullanılarak kelimeye bağlı bir oyun geliştirme projesine yönelik hazırlanmıştır. Bu projenin amaçları arasında mobil programlama hakkında bilgi ve beceri kazanmak, oyun geliştirme ve uygulama oluşturma becerisi edinmek, ve dinamik özelliklere sahip bir program geliştirmek yer almaktadır. Projenin programlama dili olarak Kotlin, geliştirme ortamı olarak Android Studio kullanılmıştır ve oyun içeriği oyun ekranındaki harflerden kelime oluşturmaya çalışmaktır.

Bu doğrultuda, bir kelime havuzu oluşturulması gerekmektedir. Python programlama dili kullanılarak bir script oluşturulmuştur. Bu script, Türk Dil Kurumu'nun resmi web sitesinde yer alan kelime listesini tarayarak havuzda yer alacak kelimeleri seçmiştir. Script, Türkçe harfler dışındaki karakterleri filtrelemiş ve yalnızca en az 3 harfli kelimeleri havuza dahil etmiştir. Ayrıca, havuzda yer alacak kelimelerin anlamlı olması için özel isimler ve yer isimleri gibi kelimeler filtelenmiştir. Sonuç olarak 58831 kelimelik bir türkçe kelime listesi oluşturulmuştur.

#### III. YÖNTEM

##### A. Giriş Ekranı

GirisActivity sınıfı, AppCompatActivity sınıfından kalıtım alır ve onCreate fonksiyonu, etkinlik oluşturulduğunda çağrılır. Bu fonksiyonda, setContentView fonksiyonu kullanılarak, etkinliğin görsel arayüzü belirlenir. Bu kodda, activitygiris adlı bir XML dosyası kullanılır. Bu xml dosyasında sadece bir buton bulunur.

Daha sonra, findViewById fonksiyonu kullanılarak, "btnplay" adlı düğme ögesi tanımlanır ve playButton adlı bir değişkene atama yapılır.

setOnClickListener fonksiyonu, playButton düğmesine tıklanması durumunda yapılacak işlemleri belirler. Bu kodda, bir Intent nesnesi oluşturulur ve MainActivity sınıfına yönlendirme yapılır.

Bu kodun amacı, kullanıcının uygulamayı başlatması durumunda karşılaşılabilecek ilk ekranı oluşturmak ve bir oyun başlatma düğmesi eklemek. Kullanıcı Play butonuna tıkladığında ise oyun başlar.

##### B. Oyun Ekranı

1) *GridAdapter*: Bu sınıf, kelime bulma oyunu için GridView için özel bir adaptör olan GridAdapter sınıfını içerir. Bu adaptör, oyun tahtasının görünümünü ve işlevselliğini yönetir.

GridAdapter sınıfı, bir dizi özellik ve yöntem içerir. Bu özellikler, oyun tahtasının genişliği, yüksekliği, tahta elemanlarını depolamak için bir diziyi, seçili harfleri depolamak için bir ArrayList'i ve harflerin puanları için bir HashMap'i içerir. Bu yöntemler, oyun tahtasındaki bir elemanın değerini ayarlama, oyun tahtasındaki bir elemanın değerini alma, seçili harfleri eklemek veya kaldırmak, oyun tahtasındaki elemanları kaydırmak, seçili harfleri temizlemek gibi çeşitli işlevleri yerine getirir.

GridAdapter sınıfı ayrıca getView() yöntemini içerir. Bu yöntem, her bir oyun tahtası elemanının nasıl görüntüleneceğini belirler. Eğer bir elemanın değeri boş ise, arka planı temizler ve tıklanmayı devre dışı bırakır. Eğer bir elemanın değeri boş değilse, arka planı mavi bir kare olarak ayarlar ve tıklanmaya izin verir. Kullanıcının bir harf seçtiğinde, seçili harfleri izlemek ve harf değerlerini hesaplamak için gereken işlemler gerçekleştirilir.

Bu adaptör sınıfı, oyunda kullanılan Grid şeklindeki tahtanın düzenlenmesi ve yönetilmesini sağlar, oyun tahtasının görüntüsünü ve işlevselliğini kontrol eder.

2) *MainActivity*: Oyun, 8 sütun ve 10 satırdan oluşan bir kare biçimli tahta üzerinde oynanır. Oyuncu, harflerden oluşan rastgele bir kelimeyi oluşturmak için tahtadaki harfleri seçer. Oyuncu, tahtadaki harfleri tıklayarak kelimeyi oluşturur ve ardından "Kontrol" düğmesine tıklar. Eğer oluşturduğu kelime,

Türkçe bir kelime ise oyuncu puan kazanır ve tahtadaki harfler silinir. Oluşturduğu kelime Türkçe bir kelime değilse oyuncu yanlış bir giriş yaptığı anlamına gelir ve kelime silinir. Oyuncu aynı zamanda "Sil" düğmesine tıklayarak da oluşturduğu kelimenin son harfini silerek geri alabilir.

MainActivity sınıfındaki değişkenler:

- gameIsRunning: Companion object, oyuncunun oyunun devam edip etmediğini takip etmek için kullanılır.
- gameIsFinished: Oyunun bittiğini belirtmek için kullanılır.
- gameCoroutine: Oyun döngüsünün çalıştırılmasında kullanılan bir Job nesnesidir.
- numColumns ve numRows: Tahtanın sütun ve satır sayısını belirtir.
- gridView: Oyun tahtasını içeren GridView nesnesidir.
- adapter: GridView'in içeriğini yönetmek için GridAdapter sınıfının bir örneğidir.
- wordTextView: Oluşturulan kelimenin görüntülediği TextView nesnesidir.
- buttonCheck ve buttonDelete: Sırasıyla "Kontrol" ve "Sil" düğmeleridir.
- pauseButton, restartButton, ve continueButton: Sırasıyla "Duraklat", "Yeniden Başlat" ve "Devam Et" düğmeleridir.
- pointsTextView: Oyuncunun puanını gösteren TextView nesnesidir.
- wrongAttempt: Oyuncunun yanlış tahmin sayısını takip etmek için kullanılır.
- random: Rastgele kelime seçmek için kullanılan bir Random nesnesidir.
- score: Oyuncunun toplam puanını tutar.
- tempScore: Oyuncunun son kelime için kazandığı geçici puanı tutar.
- words: Oyunda kullanılabilen Türkçe kelimelerin bir HashSet nesnesidir.
- wordList: Dosyadan okunan kelimelerin bir listesi.

MainActivity sınıfındaki fonksiyonlar:

- clearGame(): Oyundaki değişkenlerin sıfırlanmasını sağlar.
- clearOpacities(): Harflere tıklandığında saydamlaştıktan sonra geri alma işlevini yapar.
- showAlertDialog: Oyunu tekrar başlatma butonuna tıklandığında çıkan diyalog fonksiyonudur.
- loadWordsFromAssets(): Kelimeleri text dosyasından okuyup listeye atar.
- setAnswer(): Oyun esnasında tıklanan harflerin cevap kısmında gözükmelerini sağlar.
- initGrid(): Oyunun başlangıç esnasında en alt 3 satıra kutuları düşürme işlevini gerçekleştirir.
- startGame(): Oyunun başlamasını ve kutuların düşmesini sağlar.
- calculateGameSpeed(): Oyun hızının puana göre ayarlar.
- getRandomLetter(): Belirlenen oranda harflerin rastgele seçilme işlevini sağlar.

- dropRowBox(): 3 kere art arda hatalı kelime girilirdiğinde 1 satır kutu düşürür.
- dropBox(): Rastgele herhangi bir sütuna kutu düşürür.
- checkGameIsFinished(): Oyunun bitip bitmediğini en üstteki satıra bakarak kontrol eder.

### C. Çıkış Ekranı

Bu kısım, "ResultActivity" adlı bir aktivite sınıfı içerir. Bu aktivite, oyunun sonuç ekranını gösterir ve kullanıcının puanını ve en yüksek puanını gösterir.

İlk önce, "onCreate" fonksiyonu çağrılır ve "setContent" fonksiyonu kullanılarak aktivitenin düzeni ayarlanır. Daha sonra, "scoreTextView", "highScoreTextView" ve "restartButton" adlı üç farklı görüntü ögesi, ilgili görüntü ögesi kimliklerine bağlanarak, puan, en yüksek puan ve yeniden başlatma düğmesi tanımlanmış olur.

Sonra, "intent.getIntExtra" fonksiyonu kullanılarak, önceki aktiviteden gelen puan bilgisi alınır ve "scoreTextView" ögesi üzerinde görüntülenir. Daha sonra, "getSharedPreferences" fonksiyonu kullanılarak, GAMEDATA adlı bir paylaşılan tercih dosyası alınır ve bu dosyadan en yüksek puan bilgisi alınır.

Eğer kullanıcının yeni puanı, mevcut en yüksek puandan daha yüksekse, "if" bloğu içindeki kod çalışır. Bu kod, "highScoreTextView" ögesi üzerinde kullanıcının yeni puanını gösterir ve aynı zamanda "editor.putInt" fonksiyonunu kullanarak, HIGHSCORE adlı bir anahtar altında kullanıcının yeni puanını paylaşılan tercih dosyasına kaydeder.

Eğer kullanıcının yeni puanı, mevcut en yüksek puandan daha düşükse, "else" bloğu içindeki kod çalışır ve sadece "highScoreTextView" ögesi üzerinde mevcut en yüksek puanı gösterir.

Son olarak, "restartButton" üzerine bir "setOnClickListener" fonksiyonu atanır. Bu fonksiyon, "MainActivity" aktivitesine bir intent oluşturarak oyunu yeniden başlatır ve "finish" fonksiyonunu kullanarak "ResultActivity" aktivitesini sonlandırır.

## IV. SONUÇ

Bu proje ile mobil programlama hakkında bilgi ve beceri kazanılmıştır ve oyun geliştirme ve uygulama oluşturma becerisi edinilmiştir.

## KAYNAKLAR

- [1] <https://developer.android.com/guide/components/activities/intro-activities>
- [2] <https://developer.android.com/develop/ui/views/text-and-emoji/fonts-in-xml>
- [3] <https://developer.android.com/kotlin?hl=tr>
- [4] [https://github.com/CanNuhlar/Turkce-Kelime-Listesi/blob/master/turkce\\_kelime\\_listesi.txt](https://github.com/CanNuhlar/Turkce-Kelime-Listesi/blob/master/turkce_kelime_listesi.txt)



Ek1: Deneysel Sonuç 1



Ek2: Deneysel Sonuç 2