

YAZILIM LABORATUVARI II  
2.PROJE  
KOCAELİ ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
GONCAGÜL KOÇAK  
200201108  
[200201108@kocaeli.edu.tr](mailto:200201108@kocaeli.edu.tr)

### **Proje Tanımı**

Bu dokümanda Android Studio’da Java dili kullanılarak bir mobil uygulama olan kelime oyunu yapılmaya çalışılmıştır.Dokümanda projenin yapım aşamalarına yer verilmektedir.

kullanılmıştır.Proje kelime oyunudur.Bunun için bir kelime havuzu oluşturur.Kelime dosyası yetmiş binden fazla kelime içermektedir.Bu kelimeler en az 3 harfli, özel isimler içermeyen kelimelerden oluşmaktadır.Kelime dosyası bu şekilde düzenlenmiştir.

### **Giriş**

Projede mobil uygulama gerçekleştirmek için Android Studio platformu kullanılmıştır.Android studio android uygulamaları geliştirmek için kullanılan bir yazılım aracıdır.Android Studio birçok farklı cihazda ve işletim sistemi sürümünde çalışabilen uygulamaların tasarlanması için görsel bir arayüz düzenleyicisi sunar.Bu da projenin ilerlemesinde epey kolaylıklar sağlamıştır.Bu proje Android Studioda gerçekleştirilmiştir.Java dili

### **Araştırmalar ve yöntemler**

Proje Android Studio java dili kullanılarak gerçekleştirilmektedir.İlk olarak Android Studio nun sdk versiyonları ayarlandı.Buna uygun emülatör seçildi.Proje bu emülatörde test edilmektedir.Emülatörün sdk sı ayarlandı.Bu sayede kullanılan telefonun yani mobil aletin özellikleri belirlenmiş oldu.Daha sonrasında activity\_main.xml dosyasında projenin arayüzü çizildi.Bir adet TableLayout

kullanıldı.Rastgele harflerin gösterileceği yer için bir textview oluşturuldu.Harfler 10 satır 8 sütun olarak oluşturulan matris üzerinde gösterilmiştir.Bu matriste yukarıdan aşağıya,soldan sağa doğru satır sütunların sayıları da yazılmıştır.

Rastgele oluşturulan her harf için alanlar ayrılmıştır.Burada rastgele harfler gösterilmektedir.

komar  
kombi  
kombili  
kombina  
kombinasyon  
kombine

Yukarıda kelimeler.txt dosyasından bir kesit verilmiştir.Dosyada bu şekilde harfler alt alta yazılmıştır.Oluşturulan kelimeler kontrol edilirken bu durum dikkate alınmıştır.



Burada arayüzün temeli oluşturulmuştur.İşlemler bu arayüzde gösterilecektir.

```
<TextView
    android:id="@+id/textView"
    android:layout_width="0dp"

    android:layout_height="21dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Bu kod TextView öğesinin XML dosyasında tanımlanmasını sağlar.TextView, ekranda metin göstermek için kullanılan bir bileşendir.Ayrıntılarına bakılacak olunursa;

Android:id özelliği ,TextView öğesine benzerlik bir kimlik tanımlar.Bu id özelliği diğer oluşturulan öğeler için de kullanılmıştır

Android:layout\_width özelliği TextView'in yatay genişliğini belirler.

Android:layout\_height özelliği TextView'in dikey yüksekliğini belirler.

Diğer öğeler TextView öğesinin hangi kısıtlamalara göre konumlandırılacağını belirler.

Daha sonrasında iki adet buton oluşturulmuştur. Bu butonlardan biri silme işlevi diğeri kontrol işlemini sağlamaktadır. Matris üzerinden seçilen kelimelerle oluşturulan kelimenin kelimeler.txt dosyasında mevcut olup olmadığını kontrol eder. Buna göre mesaj yazılır. Ayrıca harflerin üzerine tıklanıldığında kırmızı renk yanmaktadır.

MainActivity.java sınıfında ilk olarak gerekli olan kütüphaneler eklenmiştir. import edilmesi gereken metodlar yüklenmiştir.

```
private static final int ROW_COUNT = 10;
private static final int COL_COUNT = 8;
private char[][] letters = new char[COL_COUNT][ROW_COUNT];
private Random random = new Random();
private Handler handler = new Handler(Looper.getMainLooper());
private boolean isFalling = false;
private int count = 0;
private int selectedRow = -1;
private int selectedCol = -1;
private String selected_letter;
private String selectedLetters = "";
```

Row\_count ve col\_count sabitleri, oyun tahtasının satır ve sütun sayısını belirler.

Letters değişkeni, oyun tahtasındaki harfleri depolamak için iki boyutlu

bir karakter dizisidir. Tahta sütunlarını temsil etmek için iki boyut kullanılır.

Random değişkeni oyun sırasında rastgele harf üretimi için kullanılmıştır.

isFalling değişkeni, seçilen harfin tahta üzerinde düşmekte olup olmadığını belirtir.

Count değişkeni, oyun sırasında kaç harf seçildiğini sayar.

selectedRow ve selectedCol değişkenleri, kullanıcının seçtiği harfin tahta üzerindeki satır ve sütun numaralarını tutar.

Selected\_letter değişkeni, kullanıcının seçtiği harfi tutar.

selectedLetters değişkeni, kullanıcının seçtiği tüm harfleri bir yerde toplar.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    TableLayout tableLayout = findViewById(R.id.table_layout);
    generateInitialLetters();
    drawTable(tableLayout);
    startFalling();
    TextView selectedLetterTextView;
    selectedLetterTextView = findViewById(R.id.selected_letter);
}
```

Bu kod, bir Android uygulamasında "onCreate" yönteminin bir parçasıdır ve aktivite oluşturulduğunda çalışır. Aşağıdaki işlemleri gerçekleştirir:

super.onCreate(savedInstanceState); yöntemi, üst sınıf olan

AppCompatActivity'dan onCreate yöntemini çağırır.

setContentView(R.layout.activity\_main); yöntemi, bu aktivitenin kullanacağı XML dosyasının tanımını sağlar. Bu durumda, "activity\_main.xml" adlı dosyada tanımlanan görünüm bileşenleri bu aktiviteye yüklenecektir.

findViewById(R.id.table\_layout); yöntemi, aktivitedeki TableLayout bileşenini bulur ve tableLayout değişkenine atar. TableLayout, bir tablo görünümü oluşturmak için kullanılır.

generateInitialLetters(); yöntemi, oyun tahtasında ilk harflerin oluşturulmasını sağlar.

drawTable(tableLayout); yöntemi, oluşturulan harflerin tableLayout bileşeninde görüntülenmesini sağlar.

startFalling(); yöntemi, harflerin düşmeye başlamasını sağlar.

findViewById(R.id.selected\_letter); yöntemi, seçilen harfin görüntülenmesi için TextView bileşenini bulur ve selectedLetterTextView değişkenine atar.

Burada oyunun temel bileşenlerini oluşturulur ve görüntüleme işlemleri başlatılır.

```
ArrayList<String> words = new ArrayList<>();

try {
    InputStream is = getAssets().open( fileName: "kelimeler.txt");
    Scanner scanner = new Scanner(is);
    while (scanner.hasNextLine()) {
        String line = scanner.nextLine();
        String[] splitWords = line.split( regex: "\\s+");
        for (String word : splitWords) {
            words.add(word);
        }
    }
    scanner.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

Burada kelimeler.txt dosyasındaki kelimeler listeye atılır.

ArrayList<String> words = new ArrayList<>(); öncelikle, words adında bir ArrayList nesnesi oluşturulur. Bu nesne, dosyadan okunan kelimeleri tutmak için kullanılacaktır.

InputStream getAssets().open("kelimeler.txt"); satırı, "assets" klasöründe bulunan "kelimeler.txt" adlı dosyayı açar ve InputStream olarak okur. "assets" klasörü, uygulama tarafından kullanılan dosyaların tutulduğu bir klasördür.

Scanner scanner = new Scanner(is); satırı, dosyayı okumak için bir Scanner nesnesi oluşturur.

while (scanner.hasNextLine()) { satırı, dosyanın sonuna kadar satır satır okuma işlemini başlatır.

String line = scanner.nextLine(); satırı, dosyadan okunan satırı bir değişkene atar.

String[] splitWords = line.split("\\s+"); satırı, satırı boşluk

karakterlerine göre bölerek kelimeleri bir diziye atar.

```
for (String word : splitWords) {  
    words.add(word);  
}  
// satırı, kelime dizisindeki her kelimeyi words listesine ekler.
```

scanner.close(); satırı, dosya okuma işlemini sonlandırır.

```
catch (IOException e) {  
    e.printStackTrace();  
}  
// bloğu, dosya okuma işlemi sırasında bir hata oluşması durumunda hatanın ekrana yazdırılmasını sağlar.
```

Bu kod, "kelimeler.txt" adlı dosyadan okunan kelimeleri bir ArrayList nesnesine yükler ve uygulama tarafından kullanılabilir hale getirir.

Oluşturulan harf tablosundaki her bir TextView nesnesine bir dokunma olayı dinleyicisi eklenir. Dokunma olayı gerçekleştiğinde, seçilen TextView nesnesinin sıra ve sütun numarası alınır ve arka plan rengi değiştirilir. Ayrıca, seçilen harf seçilen harfler dizisine eklenir ve bu harfler bir üstteki TextView nesnesinde görüntülenir. Dokunma olayı sona erdiğinde, seçim kaldırılır ve arka plan rengi tekrar şeffaf hale getirilir.

```
Button controlButton = findViewById(R.id.button1);  
controlButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // read words from file and add them to ArrayList  
        ArrayList<String> wordsList = new ArrayList<>();  
        try {  
            InputStream inputStream = getAssets().open("kelime");  
            BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream));  
            String line;  
            while ((line = reader.readLine()) != null) {  
                wordsList.add(line.trim().toLowerCase(Locale.getDefault()));  
            }  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
});
```

activity\_main.xml dosyasında yer alan "button1" adlı butona bir OnClickListener ekliyor. Butona tıklandığında, öncelikle "kelimeler.txt" dosyasından okunan kelimelerin yer aldığı bir ArrayList oluşturuluyor. Daha sonra, seçilen harflerin yan yana yazıldığı bir kelime oluşturuluyor ve bu kelime, oluşturulan ArrayList'te aranıyor. Eğer kelime listede yer alıyorsa bir Toast mesajı ile kelimenin listede olduğu bildiriliyor ve seçilen harfler matris üzerinden siliniyor. Eğer kelime listede yer almıyorsa yine bir Toast mesajı ile kelimenin listede olmadığı bildiriliyor.

```
Button clearButton = findViewById(R.id.button2);  
clearButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        selectedLetters = "";  
        TextView selectedLetterView = findViewById(R.id.selectedLetterView);  
        selectedLetterView.setText(selectedLetters);  
    }  
});
```

Bu kod, "Temizle" adlı bir düğmenin tıklanma olayını ele alır ve seçilen harfleri temizler. Düğme tıklandığında, seçilen harfler boş bir dize olarak ayarlanır ve daha önce

seçilen harflerin görüntülediği bir metin görünümü güncellenir.

```
private void stopFalling() {
    isFalling = false;
}

private void redrawTable() {
    TableLayout tableLayout = findViewById(R.id.table_layout);
    tableLayout.removeAllViews(); // clear the existing table
    drawTable(tableLayout); // draw the updated table
}
```

stopFalling() metodu, harflerin düşüşünü durdurmak için kullanılır. Harfler düşerken isFalling değişkeni true olur ve bu metod çağrıldığında isFalling değişkeni false olarak güncellenir, böylece harfler düşmeyi durdurur.

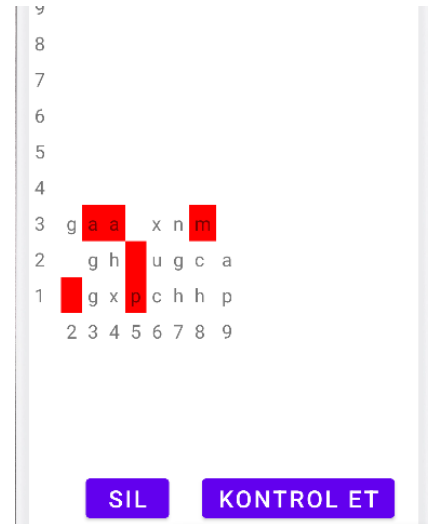
redrawTable() metodu, mevcut tabloyu temizleyip yeniden çizmek için kullanılır. Öncelikle, tablonun içeriğini temizlemek için removeAllViews() yöntemi kullanılır. Daha sonra, güncellenmiş tabloyu drawTable() metodu kullanarak çizer. Bu yöntem, yeni harflerin eklenmesi veya harflerin silinmesi durumunda tabloyu güncellemek için kullanılabilir.

```
private void startFalling() {
    Thread thread = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                try {
                    Thread.sleep( millis: 500); // wait for 500
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                if (fallLetters()) {
                    break;
                }
            }
            handler.post(new Runnable() {
```

Bu kod, harflerin düşmesini sağlamak için bir arka plan iş parçacığı oluşturur ve her 500ms'de bir harflerin düşmesini kontrol eder. Eğer bir harf düştüyse, düşme işlemini durdurur. Düşen harfler ekranın üzerinde yeniden çizilir ve sonuç olarak oluşan yeni görüntü gösterilir.

drawTable() metodunda ise harf tablosu oluşturur. Tablo, satır ve sütun numaralarını içerir ve her hücrede bir harf veya boş bir alan bulunur. Her satır ve sütun için birer TextView ögesi oluşturulur ve TableLayout'a eklenir. Satır numaraları, tablonun en üst kısmında, sütun numaraları ise tablonun sol tarafında yer alır.

Sonuç ve Deneysel Sonuçlar:



## Deneyisel Sonuçlar:

Android Studioda java dili kullanılmıştır. Mobil uygulama oluşturulmaya çalışılmıştır.

## KABA KOD:

1:BAŞLA.

2:Kelimeler.txt dosyasını düzenle.

3:Harf matrisini oluştur.

4:Satır ve sütun sayılarını yazdır.

5:Her harf için textView oluştur, metin alanı oluştur.

6:Rastgele harfler üret.

7:Harfler üzerine tıkla.

8:oluşan kelimeyi dosyadaki kelimelerle karşılaştır.

9:Kelime mevcutsa “kelime” listede bulundu mevcut değilse sil butonuna tıkla mesajı yayınla.

10:Kelime mevcutsa matris üzerinden tıkladığın harfleri sil.

11:BİTİR.

- <https://www.youtube.com/watch?v=DnY745XPI0Q>
- <https://www.youtube.com/watch?v=WW2ElrV0xu8>

## Kaynakça:

- <https://developer.android.com/studio/intro>.
- <https://www.youtube.com/watch?v=WW2ElrV0xu8>
- <https://youtu.be/WW2ElrV0xu8>