**Web Text Combiner**

Berk Sunduri

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

berksunduri@gmail.com

**1-)Projenin Tanımı**

Bu kısım sadece projenin açıklamasını okuyup edindiğim ön bilgiye göre yazılmıştır.

Bize verilen pdf dosyasında projenin amacını öğrendim. Verilen isterleri dikkatlice okudum

Projede bizden istenen şeyin mobil programlama kullanılarak bir oyun geliştirmek olduğunu anladım.

**1.1-)Problem Tanımı**

Bu kısımda bizden yapmamız istenilenler içermektedir.

Birinci adımda bizden oyunda kullanılacak olan kelime havuzunun oluşturulması istendi.

İkinci ister olarak oyunun başlangıç anında harflerin düşüp oyunun başlaması istendi.

Üçüncü aşamada ise bizden bulunan kelimenin arayüz üstünden seçilmesi ve hızlı bir şekilde kontrolünün yapılıp hatalı veya doğru olduğunu tespit etmemiz istendi.

Dördüncü ister olarak bizden puan hesap işleminin doğru yapılması istendi

Beşinci ister olarak 3 kez anlamsız kelime girildiğinde tüm sutunlardan harflerin düşmesi istendi.

Altıncı ister olarak harf özelliklerinin eklenmesi ve bunların çalışma şekillerini kodlamamız istendi.

Yedinci isterde ise bizden oyunun son bulması ve puanın hesaplanarak puan tablosuna eklenmesi istendi

Son isterde ise arayüz tasarımı ve harf geçişlerini kodlamamız istendi.

**2-)Yapılan Araştırmalar ve Karşılaşılan Sıkıntılar**

Bu kısım proje öncesi ve sonrası araştırmaları ve de projenin yapım aşamasındaki sıkıntıları ve çözümlerini içermektedir.

İlk karşılaştığım sorun hangi programlama dilini kullanacağım olduğuydu. IDE olarak Android Studio, programlama dili olarakta Kotlin dilini seçtim.

Daha sonrasında kotlinde ilk defa bir mobil oyun geliştirdiğim için bunun mekaniklerini öğrenmek beni baya zorladı. Bunun için stackoverflow’dan yardım aldım.  
  
Diğer bir zorluksa harf özelliklerini kodlamak benim için bir başarısızlık oldu. Ne kadar denesemde başarısız oldum. Süre kısıtlamasından dolayı bu özelliği oyunumdan çıkarmaya karar aldım.

**2.1-Proje Sırasında Yararlanılan Teknolojiler**

Projeyi Kotlin dili kullanarak Android Studio IDE’sinde yazdım.

Programı yaparken Kotlin’in birkaç kütüphanesinden yararlandım.

**3-)Tasarım**

**3.1-Akış Diyagramı**

Kısım ektedir.(1)

**4-)Genel Yapı**

**4.1-Kullanıcı Kısmı**

Oyun başladığında, her sütunun alt üç satırı rastgele oluşturulmuş harflerle doldurulur. Kullanıcılar, üzerlerine dokunarak bu harflerle etkileşim kurabilir. Kullanıcı bir harfe dokunduğunda, ekranın alt kısmında oluşan kelime bölümüne eklenir.

Kullanıcılar, herhangi bir sütun veya satırdan harf seçerek kelimeler oluşturabilir. Harflerin seçilme sırası, oluşturulan kelimenin sırasını belirleyecektir.

Kullanıcı geçerli bir kelime oluşturduğunda, kelimeyi doğrulamak ve silmek için "OK" düğmesine dokunabilir. Kelime oyunun kelime listesinde bulunursa, seçilen harfler ızgaradan kaldırılacak ve kullanıcı kelimedeki her harfin değerine göre puan kazanacaktır.

Kullanıcılar, oluşturulan kelimeyi istedikleri zaman "CLEAR" düğmesine dokunarak temizleyebilirler. Bu eylem, seçilen harflerin seçimini kaldırır ve oluşturulan kelime bölümünü temizler.

Oyun ilerledikçe, her sütunun üstünden yeni harfler düşmeye devam edecek. Bir sütun dolduğunda, kullanıcının düşen yeni harfler için alan yaratmak üzere mevcut harfleri kullanarak kelimeler oluşturması gerekir.

Oyun, herhangi bir sütun tamamen dolduğunda sona erer ve üstten daha fazla harf düşemez. Kullanıcı bir "Oyun Bitti" mesajı ile bilgilendirilecek ve oyun yeniden başlayacaktır.

Puanlama:

Oyun, oluşturulan kelimedeki her harfin değerine göre puan hesaplar. Puan değerleri harfin Türkçedeki sıklığına göre belirlenir. Örneğin, 'A' ve 'E' harfleri daha düşük puan değerlerine sahipken, 'J' ve 'Ğ' gibi daha az kullanılan harfler daha yüksek puan değerlerine sahiptir.

Ek özellikler:

Oyun, kelime doğrulama için kullandığı geçerli Türkçe kelimelerin bir listesini içerir.

Oyundaki harfler renkle ayırt edilir: ünlüler yeşildir ve ünsüzler kırmızıdır.

Kullanıcının mevcut puanları ekranda görüntülenir. Kullanıcı geçerli kelimeler oluşturdukça puanlar toplanır.

**4.2-Kod Kısmı**

Oyun Etkinliği:

Ana oyun etkinliği, kullanıcı etkileşimlerini yönetir ve oyun mantığını yönetir. Oyun tahtasını başlatır, rastgele harfler üretir ve oyun durumunu yönetir.

Anahtar İşlevler:

initializeGameBoard(): Alttaki üç satırı rasgele harflerle doldurarak ve ızgaranın geri kalanı için boş hücreler oluşturarak oyun tahtasını başlatır.

createRandomLetter(): Türkçedeki harf sıklığına göre rastgele bir harf üretir.

createLetterView(): Oluşturulan harfle bir TextView oluşturur ve metin boyutu, renk ve arka plan gibi uygun özellikleri atar.

createEmptyCell(): Izgara düzeninde boş bir hücre oluşturur.

isValidWord(): Oluşturulan kelimenin geçerli kelimeler listesinde olup olmadığını kontrol eder.

calculateWordScore(): Harf değerlerine göre geçerli bir kelimenin puanını hesaplar.

Harf Animasyonu:

Harf animasyonu, düşen harflerin hareketlendirilmesinden ve oyun tahtasının güncellenmesinden sorumludur.

Anahtar İşlevler:

startLetterFallAnimation(): Tüm sütunlar için harf düşme animasyonunu başlatır.

animateLetterFall(): Bir harfi, dolu bir hücreye veya sütunun altına ulaşana kadar geçerli konumundan sonraki satıra yinelemeli olarak taşır. Oyun tahtasını günceller ve oyun bitti durumunu kontrol eder.

Oyun Durumu Yönetimi:

Oyun durumu yönetimi, kullanıcı girişi, kelime oluşumu ve oyun koşullarıyla ilgilenir.

Anahtar İşlevler:

onLetterClick(): Kullanıcının bir harfi tıklamasını, oluşan kelimeye eklemesini ve oyun tahtasını güncellemesini sağlar.

onOkButtonClick(): Oluşturulan kelimeyi doğrular, puanı hesaplar, oyun tahtasını günceller ve harf düşme animasyonunu başlatır.

onClearButtonClick(): Oluşturulan kelimeyi temizler ve seçili harflerin seçimini kaldırır.

checkGameOver(): Herhangi bir sütunun dolu olup olmadığını kontrol ederek oyun bitti durumuyla sonuçlanır. Oyun biterse oyunu yeniden başlatır.

removeSelectedLetters(): Başarılı kelime oluşumundan sonra seçilen harfleri oyun tahtasından kaldırır. Seçilen harfler çıkarıldıktan sonra her sütunda kalan harfler boş alanları dolduracak şekilde düşecektir. Bu işlev, kullanıcı geçerli bir kelime oluşturduktan ve "Tamam" düğmesini tıkladıktan sonra çağrılır. Tahtadan kaldırılan harfler boş hücrelerle değiştirilecek ve oyunun kesintisiz akışını sağlamak için kalan harfler için düşme animasyonu tetiklenecektir.

**5-)Kütüphaneler**

Bu kısımda projeye include ettiğim kütüphaneler bulunmakta:

1. android.animation.ObjectAnimator
2. android.content.Context
3. android.graphics.Color
4. android.graphics.drawable.GradientDrawable
5. android.os.Bundle
6. android.os.Handler
7. android.os.Looper
8. android.util.TypedValue
9. android.view.Gravity
10. android.view.View
11. android.view.ViewGroup
12. android.widget.Button
13. android.widget.FrameLayout
14. android.widget.GridLayout
15. android.widget.TextView
16. android.widget.Toast
17. androidx.appcompat.app.AppCompatActivity
18. androidx.core.view.children
19. kotlinx.coroutines.CoroutineScope
20. kotlinx.coroutines.Dispatchers
21. kotlinx.coroutines.launch
22. kotlinx.coroutines.withContext
23. java.io.BufferedReader
24. java.io.InputStreamReader
25. java.util.\*
26. kotlin.random.Random

**6-)Referanslar**

1-)Kotlin Docs

”https://kotlinlang.org/docs/home.html

2-) Android Studio Docs

” https://developer.android.com/docs “

3-) Stack Over Flow ”[https://www.](https://www.geeksforgeeks.org/closest-pair-of-points-using-divide-and-conquer-algorithm/)stackoverflow.com”