

Yazılım Laboratuvarı 2 1. Proje

Muhammet Eren Gür

Ahmet Suna

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

200202082@kocaeli.edu.tr - 210202071@kocaeli.edu.tr

I. ÖZET

Bu doküman Yazılım Laboratuvarı 2 dersi 2. Projesi için çözümümüzü açıklamak için hazırlanmıştır. Dokümanda projenin genel olarak açıklanabilmesi için proje tanımı, araştırmalar ve yöntem, geliştirme ortamı, kod bilgisi, deneysel sonuçlar, sonuç ve kaynakça konu başlıklarına yer verilmiştir.

II. PROJE TANIMI

A. Proje Tanımı

Mobil programlama kullanılarak oyun geliştirilmesi yapılacaktır. Oyun kelime oyunudur. Programlama Dili: Android Studio, Flutter, React Native.. vb

Mobil programlama kodları Android veya İOS için geliştirilecektir. Her ikisinden biri için geliştirme işlemi yapılabilir. Kodlama için istenilen programlama dili kullanılabilir.

Kelime havuzu için 50.000 den fazla kelime kullanılması gerekmektedir.

B. İsterler

İsterler arayüz isterleri, oyun yapısı ve kelime havuzu alt başlıkları altında toplanmıştır.

1) Arayüz İsterleri:

- Kelime havuzunun oluşturulması.
- Oyunun başlangıç anında harflerin düşüp oyunun başlaması işlemi.
- Bulunan kelimenin arayüz üstünde seçilmesi ve hızlı bir şekilde kontrolünün yapıp hatalı veya doğru olduğunun tespit edilmesi.
- Puan hesaplanması işleminin doğru yapılması.
- 3 kez anlamsız kelime girilme durumunda tüm sütunlardan harflerin düşmesi durumu.
- Harf özelliklerinin eklenmesi ve çalışma şekilleri (buz vs).
- Oyunun son bulması ve puanın hesaplanarak puan tablosuna eklenmesi.
- Arayüz tasarımı ve harf geçişleri.

2) Oyun Yapısı:

- Oyun 8x10'luk(yani 8 sütun 10 satır olacak) bir 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.
- Belirli oranlarda sesli ve sessiz harfler gönderilmelidir.
- Harf düşmesi işlemi ilk başta 5 saniyede bir gerçekleşecektir. Bu süre puanın her 100 ve katlarına ulaştığında bir saniye azalacaktır.
- Oyun içerisinde doğru kelimenin oluşturulması ile puan hesaplanması beklenmektedir.

3) Kelime Havuzu:

- Kelime havuzu oluşturulurken Türk Dil Kurumunda yer alan kelimeler kullanılması beklenmektedir.
- Kelime sayısı 50.000 den fazla kelime içermesi gereklidir.
- Havuzdaki kelimeler en az 3 harfli kelime olmalıdır.
- Havuzdaki kelimeler tek bir kelime şeklinde olmalıdır. Birden fazla kelime içeren yapılar bulunmayacaktır.
- Kelimeler anlamlı kelimeler olması gereklidir.
- Özel isim yer isimleri gibi kelimeler olmamalıdır.

III. ARAŞTIRMALAR VE YÖNTEM

Projemizde mobil kısım için React Native, kelime havuzunun tutulması içinde Firebase Cloud Storage kullanıldı.

Mobil kısım için React Native seçmemizin başlıca sebepleri :

- Güncel dokümantasyon.
- Düzenli güncellemelerle hataların ve eksikliklerin giderilmesi.
- Animated API sayesinde kolaylıkla animasyon uygulanabilmesi.
- Sadece gerekli durumlarda sayfanın render edilmesiyle performans kazancı sağlamak.

Veri tabanı olarak Firebase kullanılmasının başlıca sebepleri :

- Dinamik yapısı sayesinde her bir objenin farklı özellikleri bulunabiliyor.
- Json yapısı kullanılarak verilerin hızlı bir şekilde aktarılabilmesi.
- Sorguların gerçek zamanlı uygulamalar için optimize edilmesi.

Mobil oyunumuzda ilk önce karakter dizisi ve puan dizisi oluşturduk. Oyun alanı için 8x10 bir matris oluşturuldu. Oyun ilk açıldığında bu matrisin ilk 3 sütununu rastgele harflerle dolduruldu. Bu rastgele doldurma işlemi

için JavaScriptte bulunan Math.random() fonksiyonu karakter dizinin indisi olarak verildi. Rastgele aralığı verebilmek için karakter dizisinin boyutu kullanıldı. Rastgele karakter seçme adımları aynı şekilde rastgele bir karakter düşmesi ve tüm sütunlardan rastgele karakter düşmesi işlemleri için de gerçekleştirildi. Karakter düşmesi için gereken periyodik işlemlerin periyodik bir şekilde yapılabilmesi için setInterval() fonksiyonu kullanılmıştır. Bu fonksiyon React native'de useEffect() fonksiyonu içerisinde gerçekleştirildi. İşlemin düzgün şekilde gerçekleşmesi için useEffect'in "cleanup" fonksiyonu içerisinde clearInterval() fonksiyonu çağırılmıştır. Oluşturulan karakterlerin matriste de aşağı inebilmesi için gerekli fonksiyon bu periyodik fonksiyon içine eklenmiştir. Düşme işleminin görsel anlamda da güzel bir şekilde gerçekleşebilmesi için React native'de bulunan Animated API kullanılmıştır. Düşme işleminin görsel anlamda iyileştirilmesi için Animated.timing() fonksiyonunun parametreleri birkaç sefer değiştirilmiştir. Puan hesaplama ve silme işlemleri onaylama butonuna tıklanmasıyla birlikte tetiklenmektedir. Puanlama ve silme işlemleri için kelime havuzuna ilgili kelime için sorgu atılır eğer kelime bu havuz içinde bulunuyorsa puanlama ve silme işlemleri gerçekleştirilir. CloudStorage'de sorgulamayı hızlandırmak adına kelime havuzunu ilk harflerine göre gruplandırdık böylece her sorguda tüm veri havuzunda gezmek yerine sadece ilk harfinin geçtiği kelimelerde geziyoruz. Puanlama işleminde ilgili kelimenin içerdiği karakterler karakter dizisindeki karakterler ile karşılaştırılır. Karakterin indisi puanlama dizisinin indisine koyularak ilgili puan bulunur. Bulunan karakter puanı toplam puana eklenir. Silme işlemi için 8x10'a eklenen her karakterin özel bir id'si bulunuyor bu id'ler karşılaştırılarak her seferinde sadece ilgili karakterlerin silinmesi sağlanıyor. Oyun içinde bulunan "wrongAttemp" değişkeni kullanıcının kaç tane yanlış yaptığı bilgisini tutuyor. Eğer bu değişken 3'e eşitse tüm sütunlardan rastgele karakterler düşüyor ve "wrongAttemp" state'i sıfırlanıyor. Oyunun bitiş kontrolü için "isGameFinish" state'i bulunmaktadır. Eğer bir sütun tamamiyle dolduysa bu değişken "true" olur. Bu değişken "true" olduğunda skor "AsyncStorage" API ile hafızaya yazılır ve skor sayfasına geçilir. Skor sayfası için skorlar hafızadan çekilerek bir diziye atılır. Bu dizi büyükten küçüğe sıralandı. Sadece ilk 10 indisin bastırılması sağlandı.

A. Karşılaşılan Problemler

Karşılaştığımız diğer problem ise sesli ve sessiz harflerin denge ile düşmesini sağlamak. Bu sorun için sesli ve sessiz harfleri ayrı listelerde tutup her yeni harf ekleneceğinde toplam sesli sessiz harf sayısını for döngüsü ile hesaplanarak sessiz/sesli harf oranı 2 den büyük ise sesli harf küçük ise sessiz harf indirilerek 1 e 2 oranı oluşturulmuştur.

Karşılaştığımız diğer problem ise düşme süresinin puana göre azaltılması işlemi. Bu sorun useState yapısı yardımı ile oluşturulmuş değişken ile puanın belli puan aralıklarına ulaştıkça düşmesi ile kolay bir şekilde çözülmüştür.

Karşılaştığımız diğer problem ise buz harflerin dokunduğu yeri buz yapması işlemi. Bu sorun buz harfin düştüğü sütundaki ilk harfi bulana kadar for döngüsünü çalıştırıp indis bulmamız ve o indisin type bilgisini buz olarak değiştirmek olarak çözülmüştür.

Karşılaştığımız diğer problem ise buz harflerine dokunan yeni normal harfin buz harfe dönüşmesi bu sorun için her harfin düştüğü sütundaki harfin type bilgisi kontrol edilerek buzharf veya normal harf olarak düşmesi sağlanmıştır.

B. Kazanımlar

- Mobil oyun tasarımı.
- React Native'de animasyon kullanımı.
- Dinamik özelliklere sahip bir program geliştirmek.
- Mobil programlama hakkında bilgi ve beceri kazanılması.
- Dinamik özelliklere sahip bir program geliştirmek.

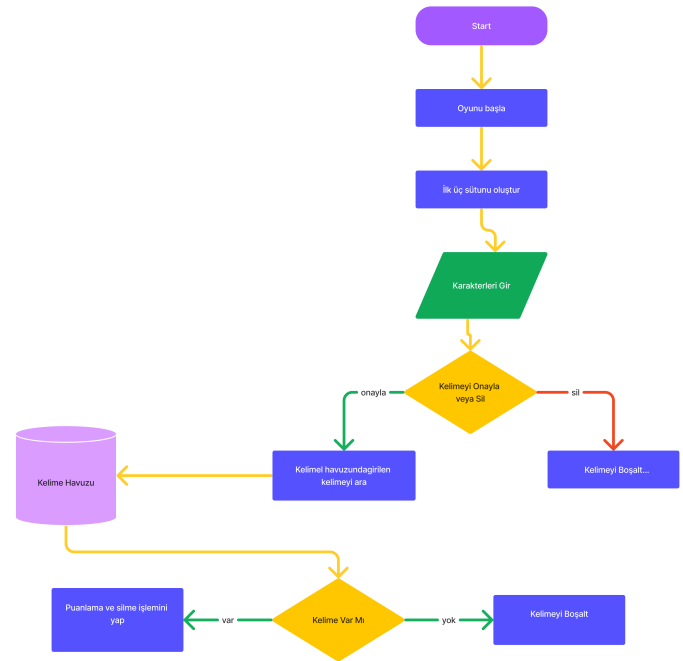
IV. GELİŞTİRME ORTAMI

Mobil kısım ve kelime havuzu düzenlenmesi için Vscode geliştirme ortamı kullanılmıştır. Organize bir takım çalışması için Git sürüm kontrol sistemi kullanılmıştır.

V. KOD BİLGİSİ

Geliştirilen projenin kod içeriğine ait bazı bilgiler akış diyagramı, istatistik ve projenin çalıştırılması alt başlıkları altında toplanmıştır.

A. Akış Diyagramı



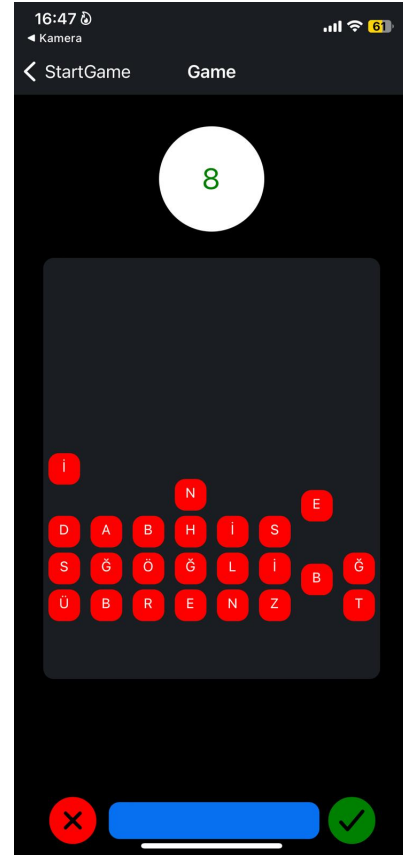
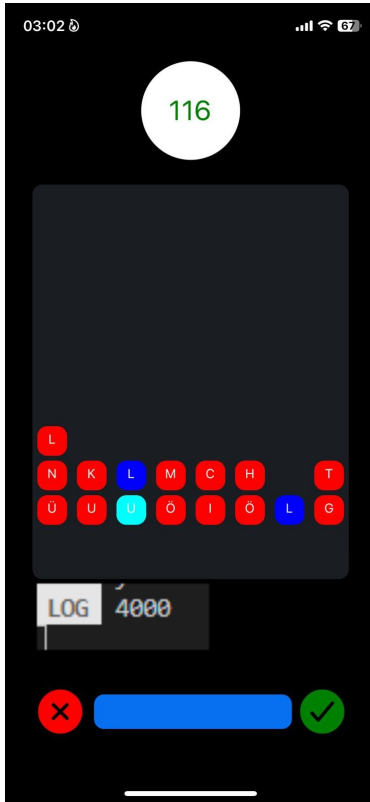
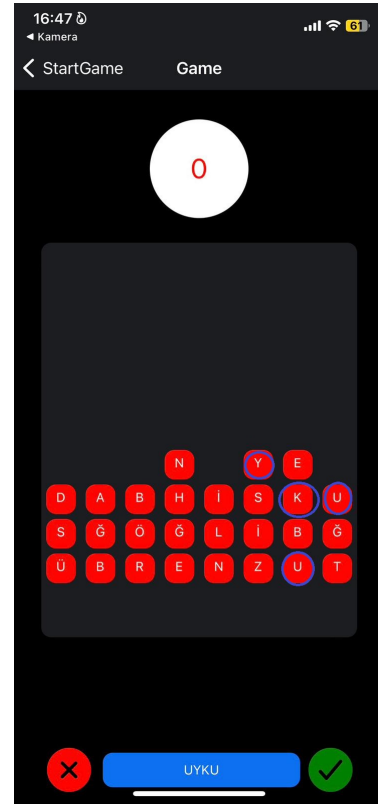
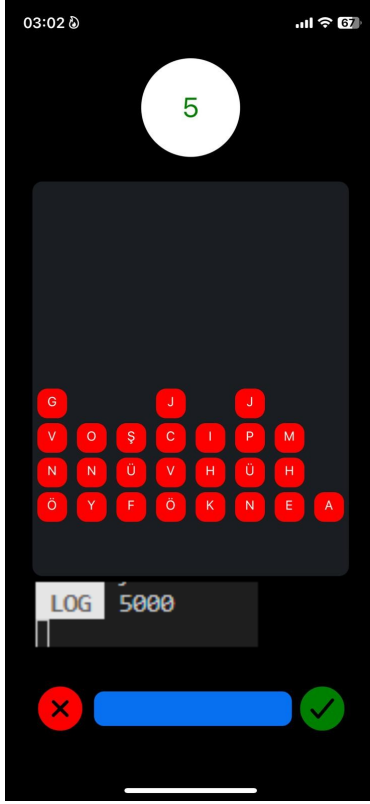
B. İstatistik

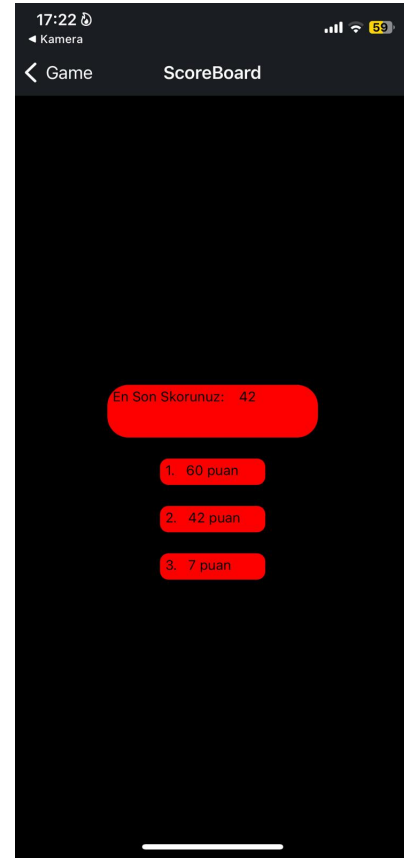
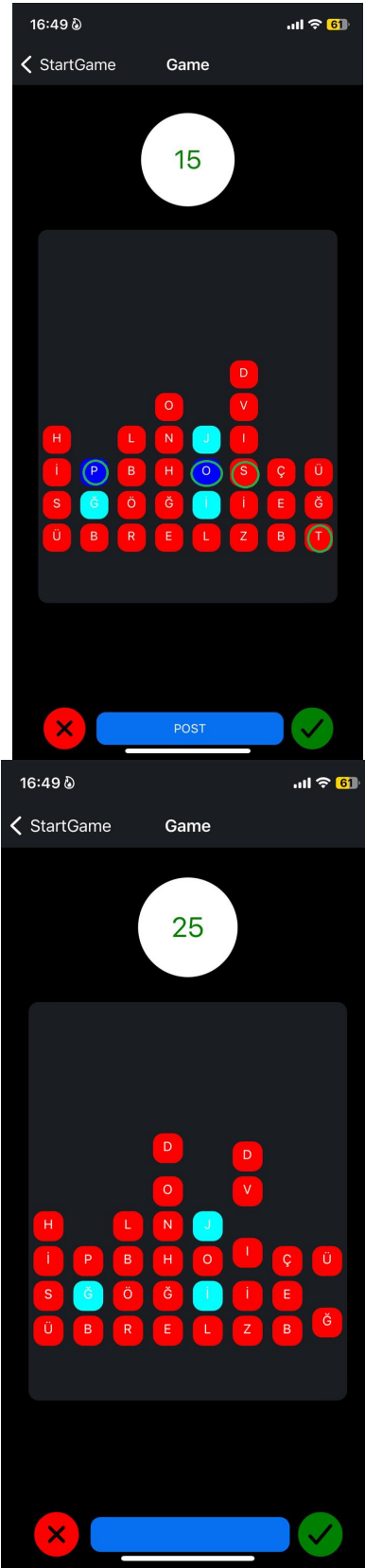
Projede 3 tane sayfa 5 tane bileşen oluşturulmuştur.

C. Projenin Çalıştırılması

Projenin çalıştırılabilmesi için Node.js yüklü olmalıdır. Projenin bulunduğu dizinde tek seferlik "npm install" komutunu çalıştırarak gerekli paketlerin kurulumunu sağlamalısınız. Ardından "npx expo start" komutuyla proje başlatılabilir.

VI. DENEYSEL SONUÇLAR





VII. SONUÇ

Başarıyla isterleri karşılayan bir mobil oyun geliştirilmiştir. Bu mobil oyun animasyonlar ile desteklenerek görsel anlamda görünüm güzelleştirilmiştir.

VIII. KAYNAKÇA

- <https://docs.expo.dev/versions/latest/>
- <https://reactnative.dev/docs>
- <https://tr.legacy.reactjs.org/docs/hooks-effect.html>