

HARRY POTTER OYUN

ALİ KUTAY BİLGİLİOĞLU
KOCAELİ ÜNİVERSİTESİ
Bilgisayar Mühendisliği(i.ö.)
200202108@kocaeli.edu.tr

ÖZET

Kullanıcı kaydı bulunduran, skor tutan müzikli, ağ veritabanı üzerinde çalışan bir hafıza oyunu. Anahtar Kelimeler: Mobil oyun, Android Studio, Kotlin, XML, Firebase

I. GİRİŞ

Tasarlanan uygulamada kullanıcı giriş yapıp zorluk seçerek oyuna başlayabilmektedir. Kullanıcının bilgileri ve kart bilgileri firebase gerçek zamanlı veritabanında tutulmaktadır. Kullanıcıya geri dönüş sağlayan ses efektleri kullanılmıştır, böylece kullanıcı bir hata yaptığında ya da doğru yaptığında rahat bir şekilde anlayabilmektedir. Kullanıcının kalan süresine göre hata yaptığında aynı oranda puan kesilmektedir, kullanıcının süresi bittiğinde oyun bitiş ekranına gönderilip kaybettiği bildirilir, eğer kullanıcı süre bitmeden tüm kartların eşini bulabilirse kullanıcıya bitiş ekranında kazandığına dair bilgi verilir.

Kotlin ve Android Studio IDE'sinin sahip olduğu uyumluluk ve emulatörü sayesinde testler yapılarak ilerlendi, XML'de tanımlanan görüntülere işlevsellik kazandırabilmek için Kotlin programlama dili kullanıldı. Veri alıp/vermeyi kolaylaştırmak için yine kotlin sınıfları oluşturuldu.

II. Temel Bilgiler

Mobil uygulama kotlinde, Android Studio IDE'si üzerinden yazılıp, arayüzün oluşturulması için XML formatı kullanıldı. Gerçek zamanlı veritabanı için ise Amerika sunucuları üzerinde çalışan Firebase kullanıldı.

III. Yöntem ve Program Mimarisi

Bu kısımda programın farklı özelliklerini oluşturmak için kullandığımız araçlar ve yöntemler üzerinde durularak ayrıntılı olarak bilgi verilecektir. Program mimarisi daha detaylı bir şekilde açıklanacaktır.

A. Veritabanı Oluşturulması

Firebase'de hesap açıldı ve uzak sunucuda çalışan bir gerçek zamanlı veritabanı oluşturuldu. Bu veri tabanı uygulama paketine bağlandı ve gerekli bağılıklar iki gradle scriptlerine de yazıldı, böylece artık hiçbir problem yaşamadan sadece değişken oluşturarak internet sunucusu üzerinde çalışan veri tabanı çalışır hale getirildi

Veritabanını ilk başta test edebilmek için bir kayıt arayüzü oluşturuldu ve kayıtlar girilerek veritabanına veri yollandı, böylece kullanıcılar için kayıt sistemi hazır haldeydi, daha sonra ise giriş kontrolü yapıldı.

Karakterlerin fotoğrafları internetten bulundu, interneti daha az yorması için ve oyunun daha hızlı yüklenebilmesi için özellikle küçük boyutlu, ancak kartların da şekline uyabilmesi için yüksekliği genişliğinden fazla olan fotoğraflar seçildi, böylece bu fotoğraflar kullanılırken kötü bir görüntü oluşmasının önüne geçilmiş olacaktır.

Görüntüler yardımcı bir websitesi sayesinde Base64 formatına çevrildi. Base64 formatı aynı zamanda css için de kullanıldığı için ve bu kullanımlar için özel ön ekler bulundurdıkları için bazı sitelerde sıkıntı yaşandı. Görüntüler test edildi ve kabul edilebilir veri çıkartan bir websitesi bulunup orda veriler Base64'e çevrildi.

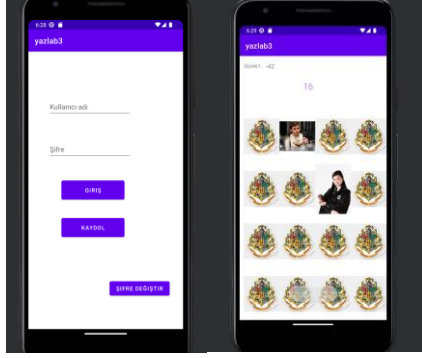
Bu veriler Kotlin aracılığı ile kod üzerinden veritabanına diğer kart verileri ile yüklendi bu sadece tek seferlik bir işlem olduğu için ve text editorun içinde yazılmış olan base64 verilerini bilgisayar kaldırmakta zorlandığı için 44 kartın verisi eklendikten sonra ilgili kod programdan silinmiştir.

ama kartların evi aynıysa, üçüncüsü eğer kartlar farklıysa ve ev de farklıysa, buna göre bir skor belirlenir ve ekrana yazdırılır. Eğer doğru bir eşleşme varsa uygulama bu eşleşmenin oyunun son eşleşmesi olup olmadığını kontrol eder böylece eğer son eşleşmeysen oyun bitirme ekranına gidilir, müzik çalınır.

Oyundan birkaç kare:

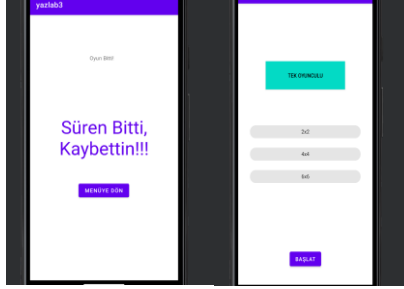
Giriş ekranı

Oyun ekranı



Oyun Son ekranı

Seçim Ekranı



IV. Deneyisel Sonuçlar

Mobilde bir oyun yapmanın zahmeti ve zorluğu deneyimlenmiş oldu.

Firestore'in çok daha kullanışlı ve gerçek hayat projelerine ne kadar da uygun olduğu anlaşıldı, herhangi bir detaya bulaşmadan sadece mobil için değil web için de kullanılabileceği anlaşıldı

İnteraktif uygulama mantığı öğrendi aynı zamanda müzik kullanımının nasıl olacağı da öğrenilmiş oldu

V. SONUÇ

Verinin internet üzerinden kullanımı öğrenildi, bir uygulamanın dosyalarını internette bulundurarak ve oyunu internetten yükleyerek aslında uygulamanın durgun halde az miktarda yer kullanması tecrübe edildi.

Kotlin programlama dili tecrübe edildi, java ile arasındaki ilişki ve kullanım farkı daha rahat anlaşıldı örneğin; bir değişkeni null safe

olarak tanımlayarak veritabanından aldığı bir veride sıkıntı olması durumunda programın exception göndererek uygulamayı durdurup nerede değişkene null gönderilmeye çalışıldığı belirtiliyor, böylece runtime hatalarından kaçınmak çok daha fazla kolaylaşıyor.

VI. Yalancı Kod

- Başla
- Harry Potter müziğini başlat
- Kullanıcıya girişi sor
- Eğer kullanıcı kayıt tuşuna basarsa kayıt sayfasına yönlendir.
- Kayıt sayfasında kullanıcı ismini ve şifresini sor, kullanıcı butona bastığında doldurduğu verileri gerçek zamanlı veritabanına ulaştır.
- Eğer giriş tuşuna basılırsa kullanıcının girdiği kullanıcı adı ve şifreyi veritabanında arat.
- Eğer veritabanında kayıt varsa toast mesajı ile kullanıcıyı bilgilendirerek kullanıcıyı seçim sayfasına at.
- Eğer veritabanında kayıt yoksa kullanıcıyı toast mesajı ile bilgileri yanlış girdiğine dair uyar
- Seçim ekranında butona basıldığında kullanıcının seçmiş olduğu tercihleri sakla ve oyun arayüzüne yönlendir.
- Kullanıcının seçtiği moda göre kartları ayarla, gereksiz sayıda kart oluşmaması ya da kartların çok küçük gözükmemesi için kullanıcının seçtiği mod kadar arayüzdeki kartları etkinleştir
- Kart verilerini veritabanından yükle
- Kartların resimlerini hazırla ve kullanıcının tıklamasını bekle
- Süreyi başlat
- Kullanıcı tıkladığı zaman tıklanan karta tıklandı bilgisini yükle
- Eğer kullanıcının açtığı ikinci kartsa değerlendirme fonksiyonuna git
- Eğer kullanıcının açtığı kartlar aynıysa puanını hesaplayıp ekrana yazdır, aynı kartı bulduğu için geri bildirim gönderen müziği çal, kullanıcının eşini bulduğu kartları invisible yap böylece kartlarını arayüzden sil sonra oyun bitimi fonksiyonuna yönlendir.
- Oyun moduna göre eğer toplam eş bulunan kartların sayısı tüm kartların sayısıyla aynıysa oyunu bitir ve kullanıcının kazandığını müzik ve görsel ile bildir.
- Eğer kullanıcının seçtiği kartlar aynı değilse kartların aynı evden olup olmadığını kontrol et

- Eğer evler aynı ise verilen formüle göre hesaplama yapıp yanlış kartların tekrar arkasını çevir
- Eğer evler aynı değilse verilen formüle göre skoru hesapla, ekrana yazdır, kartların tekrar arkasını çevir.
- Eğer süre bittiyse kullanıcıyı bitiş ekranına yolla.
-
- Kullanıcı eğer bitiş ekranındaki butona tıklarsa ana menüye yönlendir ve kullanıcıya tekrardan oyun modunu seçmesini bekle
- Kullanıcı eğer giriş ekranına dön butonuna tıklarsa kullanıcıyı giriş ekranına yönlendir.
- Tekrardan kullanıcıdan giriş yapmasını veya kaydolmak için kaydolma sayfasına yönlendirmesini bekle

VII. KAYNAKÇA

- <https://firebase.google.com/docs>
- <https://kotlinlang.org/docs/home.html>
- <https://stackoverflow.com/questions/29737499/how-can-i-resize-the-avd-emulator-window-in-android-studio>
- <https://codersguidebook.com/how-to-create-an-android-app/play-sounds-music-android-app>
- <https://www.javatpoint.com/kotlin-android-toast>
- <https://www.appsloveworld.com/kotlin/100/97/android-creating-multiple-onclicklisteners-in-a-for-loop-kotlin>
- <https://www.tutorialkart.com/kotlin/kotlin-mutable-list-remove/#:~:text=Kotlin%20List%20%E2%80%93%20Remove%20an%20Element,like%20to%20remove%2C%20as%20argument.>
-