

YAZILIM LABORATUVARI 1

PROJE 3

TARIK GÖREN
Bilgisayar Mühendisliği
200202022@kocaeli.edu.tr

ÖMER ARAN
Bilgisayar Mühendisliği
190202012@kocaeli.edu.tr

ÖZET

Bu proje ile amacımız android studio, kotlin dilini ve cloud database kullanımını aynı zamanda ilgili kavramları öğrenmek, uygulamak ve cloud veritabanı ile çalışırken dikkat etmemiz gereken noktaları keşfetmek aynı zamanda tecrübe edinmekti. Bir hatırlama oyunu olan hafıza kart oyunu adıyla geliştirdiğimiz uygulamamızı kotlin kaynakları, android studio kaynakları ve firebase veritabanını kullanmak için google kaynaklarını kullandık. Bu proje ile Android uygulama ve bulut bilişim teknolojilerinin kullanılması amaçlanmaktadır. Projede belirtilen süre içinde zorluk seviyesine göre kartları doğru eşleştirmek gerekmektedir. Projeyi Android uygulama olarak geliştirmeniz beklenmektedir.

I. GİRİŞ

Bulut sistemler günümüzde çok yaygın olması sebebiyle ilginizi bu yöne çekmek ve en azından bir proje geliştirmek maksadıyla android studio IDE ve aracını kullanarak kotlin dilinde konusu olan bir oyun projesi geliştirdik.

Bulut Bilişim: Bulut platformunda kısıtlamaya gidilmemiştir. IBM, Google, Azure ve Amazon gibi Cloud platformlarından yararlanabilirsiniz.

Bu proje kapsamında bizden aşağıdaki amaçların gerçekleştirilmesi istenmiştir:

- Giriş ekranı: Oyun ilk açıldığında ekranda açılacak sayfa giriş ekranı olmalıdır. Kullanıcı bu ekranda, kullanıcı adı ve şifresi ile giriş yapabilmeli, şifre değiştirebilmeli ve kaydolabilmelidir.
- Oyun ekranı: Kullanıcı giriş yaptıktan sonra karşısına gelecek ekran oyun ekranı olmalıdır. Burada Tek Oyuncu ve Çoklu Oyuncu Olarak iki farklı seçenek bulunmalıdır. Oyun ekranı ilk açıldığında

“BAŞLA” butonu bulunmalıdır. Oyuncu BAŞLA butonuna tıkladığında oyun ve süre başlatılır.

- Oyun başlatıldığında kartlar kapalı şekilde dağıtılmalıdır. Oyundaki kartların her birinden birer çift bulunmaktadır. Buradaki amaç açılan kartın diğer çiftini bulabilmektir. Oyunda kartlar ilk olarak rastgele dağıtılır.
- Arka plan müziği: Oyun esnasında arka planda bir müzik çalması beklenmektedir. Bu müzik oyun oynanırken çalmaya devam eder ve 3 durumda bu müzik değişecektir. 1) Kartın eşi bulunduğu farklı bir müzik ile uyarı verilir. 2) Oyun süresi bittiği zaman arka fon müziği olumsuz bir uyarı verir. 3) Süre bitmeden bütün kartların eşi bulunca arka fon müziği kazandınız anlamında olumlu bir uyarı verir. Ayrıca oyun müziğinin istenilen durumda tamamen kapatılabilmesi için bir buton eklenmelidir.
- Kullanıcı bilgileri telefonda tutulmayacak bulut üzerinden doğrulama (log-in) yapılacaktır.
- Kart bilgileri telefonda tutulmayacak bulut üzerinde bir veri tabanında saklanacak ve kullanıcı oradan erişecektir.
- Android telefonu olmayan öğrenciler uygulamayı emulator üzerinde gerçekleyebilirler.

II. OYUN MODLARI

• Tek Oyuncu

- Kartlar oyunun başında rastgele arka yüzleri kapalı olacak şekilde dağıtılır. Oyuncu bir kartın üzerine tıklar ve kart açılır. Daha sonra oyuncu farklı bir karta tıklayarak kartın eşini bulmaya çalışır.
- Oyun skoru: Oyun süresi 45 saniyedir. Oyunda her kartın bir puanı ve ait olduğu bir ev bulunmaktadır. Oyun skoru her hamle sonrasında ekranda anlık olarak gösterilecektir.
- Örn- Harry Potter (Puan:10 , Ev: Gryffindor)
- Oyuncu doğru bir eşleştirme yaparsa $[(2 * \text{kartın puanı} * \text{evin katsayısı}) * (\text{kalan süre} / 10)]$ kadar puan kazanır.
- Yanlış bir eşleştirme durumunda iki kart aynı evden ise $[(\text{kartların toplam puanı} / \text{evin katsayısı}) * (\text{geçen süre} / 10)]$ kadar puan kaybeder.
- Yanlış bir eşleştirme durumunda iki kart farklı evden ise $[(\text{kartların puan ortalaması} * \text{Ev1katsayı} * \text{Ev2katsayı}) * (\text{geçen süre} / 10)]$ kadar puan kaybeder.
- Ev katsayıları
 - * Gryffindor : 2
 - * Slytherin : 2
 - * Hufflepuff : 1
 - * Ravenclaw : 1

• Çoklu Oyuncu

- Kartlar oyunun başında rastgele arka yüzleri kapalı olacak şekilde dağıtılır. 1. Oyuncu oyuna başlar ve bir kartı seçer. Daha sonrasında kartın eşini bulmaya çalışır. Eğer kartın eşini bulursa aynı oyuncu oyuna devam eder. Eğer kartın eşini bulamazsa sıra rakip oyuncuya geçer.
- Oyun skoru: Oyun süresi 60 saniyedir. Oyunda her kartın bir puanı ve ait olduğu bir ev bulunmaktadır. Her oyuncu sırayla seçim yapar. Doğru bir eşleştirme yapan oyuncu tekrar oynama hakkına sahiptir. Oyun skoru her hamle sonrasında ekranda anlık olarak gösterilecektir.
- Örn - Harry Potter (Puan:10 , Ev: Gryffindor)

- Oyuncu doğru bir eşleştirme yaparsa $(2 * \text{kartın puanı} * \text{evin katsayısı})$ kadar puan kazanır.
- Yanlış bir eşleştirme durumunda iki kart aynı evden ise $(\text{kartların toplam puanı} / \text{evin katsayısı})$ kadar puan kaybeder.
- Yanlış bir eşleştirme durumunda iki kart farklı evden ise $(\text{kartların puan ortalaması} * \text{Ev1katsayı} * \text{Ev2katsayı})$ kadar puan kaybeder.
- Ev katsayıları
 - * Gryffindor : 2
 - * Slytherin : 2
 - * Hufflepuff : 1
 - * Ravenclaw : 1

III. MÜZİKLER

- Oyun süresince çalacak müzik:
- Kartın eşi bulunduğu anda:
- Oyun süresi bittiği zaman:
- Süre bitmeden bütün kartların eşi bulununca: olmak üzere 4 adet müzik bulunmaktadır.

IV. YÖNTEM

A. Başlangıç

Öncelikle bizden istenen programları ve programların içerisinde kullanacağımız kütüphaneleri entegre ettik. İsterleri gerçekleştirmeye başladık. Bizden istenen ilk isterden başladık. Android Studio dokümanları ve ilgili eğitim videoları ile android projesinin test kısmında denemeler yaparak kart verilerini firebase veritabanımızdan çekmeyi denedik.

B. Geliştirme Ortamı

Bu proje diğerlerine nazaran kısıt içermekteydi bu nedenle Android Studio ile Kotlin dilinde geliştirmeler yaptık. Android Studio IDE'si JetBrains tarafından geliştirilmiş bir IDE aynı zamanda yine kendi ürünleri olan IntelliJ Idea altyapısını kullanıyor. BU durumda IDE ayarlarını yapmak daha kolay ve alışması hızlı oldu. Kotlin dili ise syntax açısından javaya benzemiyor olsa da işler arka tarafta aynı şekilde yürüyor.

V. PSUEDO KOD

Bir ayıklama ve multithread projesinin yalancı kodu senaryoları gerçekleşme sırasına uygun bir durumda açıklayarak gerçekleşir.

Uygulama Çalıştırılır :

login fragment ile ekran karşılanır
hesap var ise
bilgiler girilir
logine tıklanır
hesap yok ise
registra basılır
kayıt işlemi gerçekleşir
ardından otomatik olarak
oyun ayarları ekranına yönlendirilir
burada modlar bulunmakta
oyun modunda
2*2 - 4*4 - 6*6 olmak üzere 3 mod
player modunda
1 player
2 player olmak üzere 2 mod bulunmakta
gerekli button işaretleri yapıldıktan sonra
start game denilir ve oyun ekranı
belirli konfigürasyonlara yani seçilenlere göre açılır
oyun oynamak için
bir kare resim seçilir
ardından aynı resim mi diye
bir diğer resim seçilir
eğer ki bu resimlerin id'si
aynı ise : puan verilir
hesaplamalar yapılarak
score' a eklenir
eğerki farklı ise
resimler tekrar ters döner
ve eksi puan alınır
hesaplamalar yapılarak
score' a yansıtılır
oyun bu şekilde
tek mod ise
45 saniye
çift oyunculu mod ise
60 saniye sürer
bu süre dolmadan oyun biterse
oyun bitiş ekranı gelir
ve oyundan kazandığımız
score bilgisini verir
eğer süre biterse

oyun kazanılmaz
yine de oyun bilgisi verilir
oyun kapatılır.

VI. SONUÇLAR

Proje amacı en başta cloud veritabanı kullanımını öğrenmek, bunları efektif bir biçimde öğrenmek ve uygulamaktır. Bu işlemleri öğrenirken birden fazla noktaya odaklanmak yorucu ve bir hayli zordur. Daha önce deneyim etmediğimiz bir IDE ve araçları barındıran Android studio aynı zamanda proje mantığı ve kodlarını yazarken kullandığımız Kotlin dili ve yine projemizi bağlamamız gereken veritabanı ile ortaya bir ürün çıkarmak aynı zamanda bu ürünün hatalardan uzak iyi bir kullanıcı deneyimi sunması için geliştirmelerimizi göze hitap ettiği kadar clean code prensiplerine uygun durumda olmasını da önemsedik. Birden fazla fragment yardımıyla gerekli yönlendirmeleri yaparak kullanıcı açısından kolay bir arayüz sunmaya çalıştık. Oyun kuralları gayet açık ve verilen skor hesap formülleri ile backend tarafında anlık olarak skor değerini yeniliyoruz. Android bir uygulama geliştirmek ve bu adımların nasıl olduğunu öğrenmek birçok bilgi ve deneyim sağladı.

KAYNAKLAR

[1] Android Studio dokümanları

<https://developer.android.com/studio>

<https://developer.android.com/>

[BtkAkademi-girisSeviyeAndroid](#)

[BtkAkademi-ileriSeviyeAndroid](#)

[2] Kotlin Dokümanları

<https://developer.android.com/kotlin>

<https://talentgrid.io/tr/kotlin-nedir-kotlin-ogrenmek-icin-te>

<https://kotlinlang.org/docs/home.html>

https://www.w3schools.com/KOTLIN/kotlin_oop.php

<https://github.com/JetBrains/kotlin>

<https://www.btkakademi.gov.tr/portal/course/kotlin-ile-android-mobil-uygulama-gelistirme-egitimi-temel-seviye-10274>

<https://gelecegiyazanlar.turkcell.com.tr/konu/kotlin>

<https://stackoverflow.com/questions/21498534/hide-menuitem-in-some-fragments>

<https://developer.android.com/guide/navigation/navigation-custom-back>

<https://developer.android.com/reference/kotlin/android/os/Bundle>

[3] Google Firebase Dokümanları

<https://firebase.google.com/docs/firestore>

<https://firebase.google.com/docs/auth>

<https://medium.com/swlh/firebase-authentication-with-kotlin-46da70bf8a4d>

<https://medium.com/fnplus/cloud-firestore-kotlin-33892886ce64>

<https://por-porkaew15.medium.com/crud-firestore-with-kotlin-e967743db5ab>

EKRAN GÖRÜNTÜLERİ



