

KOCAELİ ÜNİVERSİTESİ YAZLAB1 PROJE3

1st Zeynep SİNOĞLU
Öğrenci No:200202044

2nd Melih BAŞAYİĞİT
Öğrenci No:200202044

I. ÖZET

Bu doküman Yazılım Laboratuvarı-1 dersi 3. projesi için çözümümüzü açıklamaya yönelik oluşturulmuştur. Dokümanda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Doküman sonunda projemizi hazırlarken kullandığımız kaynaklar ve proje derlenirken dikkat edilmesi gereken hususlar bulunmaktadır.

II. GİRİŞ

A. Projenin Amacı

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.

III. PROJE İSTERLERİ

A. Giriş 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.

B. Oyuncu 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.

C. Oyun Zorluk Seviyesi

- 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.

Identify applicable funding agency here. If none, delete this.

D. 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.
- 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 $[(\text{kartların puan ortalaması} * \text{Ev1 katsayı} * \text{Ev2 katsayı}) * (\text{geçen süre} / 10)]$ kadar puan kaybeder

E. 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.
- Oyuncu doğru bir eşleştirme yaparsa $(2 * \text{kartın puanı} * \text{evin katsayısı})$ kadar puan kazanır.
- Yanlış bir eşleştirme durumunda iki kart farklı evden ise kadar puan kaybeder.
- Rastgele dağıtılan kartların bilgisi (ön yüzlerinde hangi karakterin bulunduğu bilgisi) ayrıca bir not defterinde tutulup anlık olarak takip edilebilecektir.
- 4*4 ve 6*6 dağıtılan destelerde her evden eşit sayıda karakter bulunması gerekmektedir.
- Veri tabanı tutulacak. Veritabanında kullanıcı adı, şifresi, ID bilgisi, e-posta hesabı ve kart bilgileri (adı, evi, puanı, kartı resmi) bilgileri tutulmalıdır.
- Resimler veri tabında Base64 tipinde tutulmalıdır.
- Not: Bulut platformunda kısıtlamaya gidilmemiştir. IBM, Google ve Amazon gibi cloud platformundan yararlanabilirsiniz.

F. Kısıtlar

- Proje Android Studio programı kullanılarak geliştirilmiştir.
- Arayüz olarak sanal emülatör kullanıldı.
- İsterlerin birçoğu çalışıyor.

IV. YÖNTEM

Android Studio idesi kullanılarak kotlin ile oluşturulan harry potter oyununu yapmayı hedefledik. Bazı kısıtlar olan bu oyunda çoklu veya tekli oynama seçenekleri mevcut.

V. KABA KODUMUZ

- Classlar :
- Card : Data class şeklinde tanımlanmıştır. Oyun içindeki tek bir karta karşılık gelecektir. İçerisinde kartın ID'si, İsmi, Evi, Puanı, Resmi(Base64) ve lock değişkeni tutulmaktadır. Class'ın herhangi bir fonksiyonu yoktur.
- Movement : Oyuncuların hangi hamleleri yaptığını kaydetmek için oluşturulan data class'dır. Her oyuncunun id'sini ve yaptığı işlemi gösterir.
- MainActivity : Kotlin Android'te verilen ana sınıftır. Program buradan başlatılır. Diğer aktivitelere kontrollü şekilde geçişi sağlanır. Aynı zamanda giriş / kayıt ekranıdır. OnCreate fonksiyonu altında giriş, kayıt ve şifremi unuttum fonksiyonları butonlara aktarılır. Firestore cloud ile bağlantı kurulur. Eğer doğru email-şifre ikilisi girilirse kullanıcı id'si ve maili oyun menüsüne gönderilir ve oyun menüsüne geçiş yapılır
- login() : Giriş yapmayı sağlar.
- register() : Sunucuya kayıt olmayı sağlar.
- resetPassword() : Eposta adresine şifre sıfırlama maili gönderir.
- switchMenuActivity() : Oyun menüsüne geçiş yapar
- MenuActivity : Kullanıcının oyun modunu ve zorluk seviyesini seçtiği bölümdür. Aynı zamanda müzikleri kapatabilme imkanı vardır. Zorluk seçimi için difficultyFragment açılır.
- DifficultyFragment : Oyun menüsünü bloklamadan üst tarafta zorluk seçimini yapılmasını sağlar. Ekstra olarak oyun zorluğu ve oyun modunu değişken olarak saklar.
- GameActivity : Oyunun bulunduğu aktivitedir. Tüm kartların sunucudan çekilip kart sınıfına atanmasını ve bununla beraber kartların gerekli özelliklere göre evlere atılmasını sağlar. Burada tüm kartlar tutulur oyun zorluğuna göre kart adedi hesaplanıp kartlar karılıp seçilir. Kartların çiftleme işlemi gerçekleşir. Aynı zamanda oyun haritası oluşturulduğu için harita veritabanında tarih tutularak saklanır. Kart listesi GameCardAdapter sınıfına yollanır. GameActivity içerisindeki RecyclerView'in adapter'i GameCardAdapter olarak belirlenir.
- getCards() : Veri tabanından kart çekme işlemidir.
- GameCardAdapter : RecyclerView için oluşturulmuş adapter sınıfıdır. Kartların listesini alır ve her karta ulaşmak için o kartın GameCardHolder'ı tutulur. Kartı

seçme puan alma süre güncelleme tarzı işlemler burada yapılır

- gameOver() : Oyunu sonlandırır restartTimer() : timer'ı sıfırlar getCardHouse() : bir kartın evini döndürür getCardPoint() : bir kartın puanını döndürür getCardHouseK() : bir kartın evinin puan katsayısını döndürür reducePoint() : yanlış eşleşme sonucu skoru azaltır getPoint() : doğru eşleşme sonucu skoru artırır pair() : iki kartın eşleşmesi sonucu çift olup olmadığını döndürür convertStringToImage() : veri tabanından çekilen base64 olan stringi Bitmap resmi haline çevirir.
- GameCardHolder : GameCardAdapter için tasarlanmış olup RecyclerViewItem'ların binding'lerini tutmak için kullanılır.

KAYNAKÇA

Projeye başlamadan önce kotlin ve android studio nedir, nasıl kullanılır gibi temel soruların cevapları için araştırmalar yapıldı. References bölümünde ise bu süreç içerisinde bize yardımcı olan kaynak sitelerin adresleri yer almaktadır.

REFERENCES

- [1] <https://www.btkakademi.gov.tr/portal/course/kotlin-ile-android-mobil-uygulama-gelistirme-egitimi-temel-seviye-10274>
- [2] <https://www.btkakademi.gov.tr/portal/course/kotlin-ile-android-mobil-uygulama-gelistirme-ileri-seviye-10359>
- [3] <https://www.youtube.com/watch?v=BkBF1151KQI>
- [4] <https://www.youtube.com/watch?v=ttHX4cyoqQ>

PROJEYE DAIR RESIMLER





