

2022-23 Yazılım Lab. I - Proje III

1. Öğretim

Talha ÜNAL
200201038

Erkan ŞARLI
200201069

Özet :

Bu proje android studio ve java kullanılarak memory match card game'in harry potter temasına uyarlanarak implement edilmiş halidir. Proje için cloud gibi teknolojilerden yararlanılmıştır.

I. GİRİŞ

Projede Android Studio ve Java programlama dilini kullanarak Harry Potter temalı bir kart eşleştirmeli hafıza oyunu geliştirdik. Ayrıca kartların ve kullanıcı bilgilerinin bulut tarafında saklanması gerekiyordu, bunun için ise MongoDB'nin bulut hizmeti olan Mongo Atlas'tan yararlandık. Bulut tarafına bağlanmak için node.js ve express.js kullanarak kendi API'mızı yazdık. Authentication ve kart bilgileri için buradan Java tarafından yapılan http isteklerine gerekli bilgileri gönderdik. Oyunun zorluk seviyesi ve oyuncu sayısı seçenekleri için dinamik bir yol izledik ve hepsini tek bir class'ta parametrelere bağlı olarak gerçekleştirdik.

II. YÖNTEM

Projeye başlarken projenin ilk ve en önemli konsepti olan kart eşleştirme algoritmasını android üzerinde gerçekleştirmeye karar verdik. Kartların tutulacağı alan için GridLayout kullandık. Ayrıca skor, sıranın hangi kullanıcıda olduğu, ne kadar zamanın kaldığını gösterebilmek için elimizdeki grid layoutu bir linear layout un içine yerleştirdik. Oyun ekranı için bir aktivite, butonların kendine has özellikleri olacağı için de butonlar için diğer bir class oluşturduk. Butonlarda butonun eşleştirilip eşleşmediği, çevrilip çevrilmediği, sahip olduğu puan, hangi eve ait olduğu gibi bir çok has özellikler için değişkenler tanımladık. Ardından oyun için oluşturduğumuz aktivitede her buton için buton class'ına ait bir obje oluşturduk ve bu butonları ekranda render ettik, butonlara tıklandığında onClick fonksiyonu ile eşleşmenin doğru olup olmadığı gibi durumları kontrol ettik.

Butonlara ait görsellerin ve diğer bilgilerin bulut üzerinden çekilmesi isteniyordu, bunun için node.js ve express.js ile kullandığımız bir API'den yararlandık. Basitçe bu api'ye http isteği oluşturduğumuzda api, MongoDB'nin cloud hizmeti olan Mongo Atlas'a bağlanıyor, ardından response olarak elde ettiği buton bilgilerini toplu bir şekilde bizim programımıza gönderiyor. Burada yaşadığımız bir problem ise http isteğinin tamamlanmasının ardından callback fonksiyonundaki verilere erişmek idi. Bunun için internette yaptığımız araştırmalar sonucu Future

class'ının problemimize çare olabileceğini öğrendik ve projemizde uyguladık. Burada yapılan şey basitçe http isteğinden verilerin tamamıyla dönmesini beklemek oldu. Ardından elde ettiğimiz base64 stringleri buton class'ının constructor'ına parametre olarak gönderdik ve buton class'ında bu base64 leri her ayrı buton için decode edip Drawable'a çevirdikten sonra butonların sahip oldukları görüntüleri ayarlamış olduk.

Ödevde ayrıca zorluk seviyesi ve tek oyunculu, çok oyunculu seçenekleri isteniyordu. Bu iki görevin çözümü için hiçbir yeni class oluşturma gibi yollara gitmeden tamamen dinamik bir şekilde projemizde uyguladık. Tek oyunculu veya çok oyunculu için bulduğumuz çözüm aslında tamamen skor tabelasında değişmeye gitmekten ibaretti. Eğer kullanıcı çok oyunculu seçeneğini seçmişse boolean olarak tanımladığımız bir oynama sırası değişkeninden yararlanarak yanlış eşleme durumlarında bu değişkeni tersine çevirdik ve sıra diğer oyuncuya geçmiş oldu. Zorluk seçenekleri içinse kodun bir çok yerinde kullandığımız dinamik bir numberOfElements integer değişkenini değiştirerek zorluğa göre daha fazla veya daha fazla sayıda buton oluşturduk.

Giriş yapma, kayıt olma, şifre değiştirme gibi kısımlara gelecek olursak, api tarafında request olarak kullanıcı adı ve şifre gibi bilgileri alarak mongo atlas veri tabanına bakarak duruma göre 200,401,404 gibi cevaplar dönerekten kullanıcının girdiği giriş ve kayıt bilgilerinin kontrollerini status code'lar eşliğinde onayladık ve reddettik. Örneğin aynı kullanıcının tekrardan kayıt olması gibi durumları engelledik. Kullanıcı kayıt olurken email kullanıcı adı ve şifre bilgilerini aldık. Ayrıca şifre değiştirme ekranı için kullanıcı adı mevcut şifre ve yeni şifre'yi istedik ve eğer kullanıcının girdiği mevcut şifre yanlış ise şifre değiştirme işlemini engelledik.

Zaman kısmına gelindiğinde ise Countdown timer class'ını kullandık, bu class'ın override ettiği iki metod sayesinde ekrandaki saniye kısmını güncelledik ve süre bittiğinde haberdar olduk. onTick yani her saniye geçtiğinde çalışan fonksiyon üzerinde ekranda kaç saniye kaldığını gösteren TextView'i set ederek anlık olarak sayacı görüntüledik. onFinish yani süre bittiğinde çalışan fonksiyon ile kaybettiğini belirten bir müzik çalarak bütün kartları çevirdik.

Kartların her seferinden elimizdeki kart havuzundan rastgele bir biçimde çekilmesi gerekiyordu, ancak 4x4 ve 6x6 gibi durumlarda her evin sahip olduğu 11 kart üzerinden eşit sayıda kart olması gerektiği ödevde belirtilmişti, yani 4x4 oyun için her evden 2 adet kart 6x6 için her evden 4, 4, 5, 5 adet kart çekmemiz aynı

zamanda bunları rastgele bir biçimde seçmemiz gerekiyordu. Bunun için her aralıktan ayrı ayrı rastgele kart çekmemiz gerekti ve aynı kartı bir daha seçmememiz için seçtiğimiz kartların tutulacağı bir liste oluşturduk, kart her seçilirken bu listeye baktık ki bu kart daha önce seçilip seçilmemiş mi anlayabilelim.

III. KAYNAKÇA

- <https://stackoverflow.com/>
- <https://developer.android.com/>
- <https://www.geeksforgeeks.org/introduction-to-android-development/>
- <https://www.tutorialspoint.com/android/index.htm>

Email	
username	
password	
REGISTER	TEK OYUNCU
username	ÇOK OYUNCU
password	
LOGIN	
please login	
CHANGE PASSWORD	

ZORLUK SEVİYESİ

2X2

4X4

6X6

37

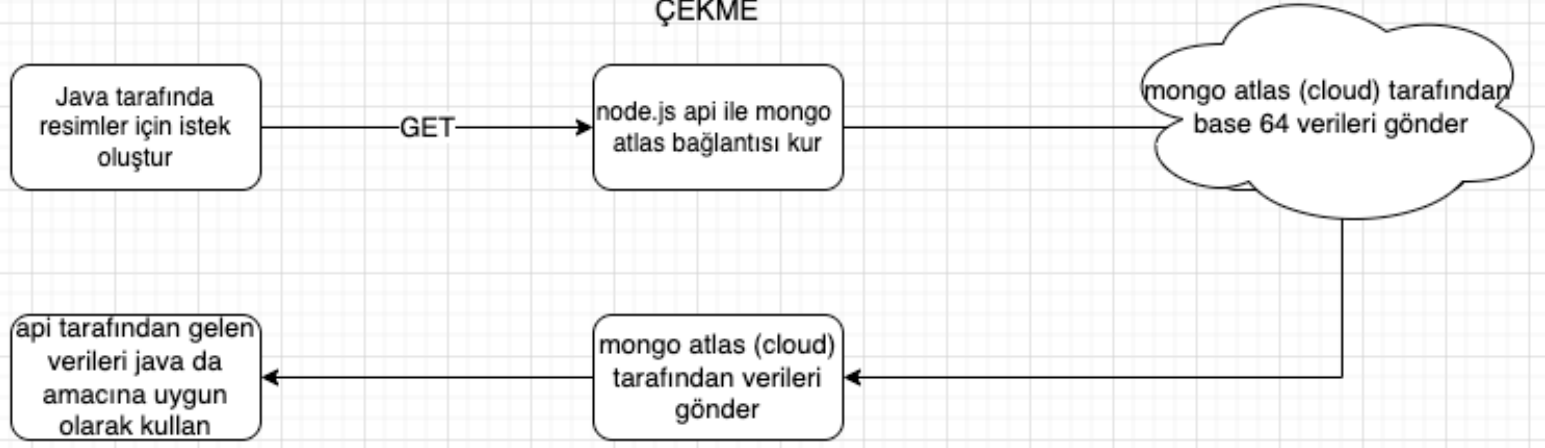
SKOR A: -2 - SKOR B: 48

SIRA : B



MUSIC: ON

RESİMLERİ ÇEKME



GİRİŞ YAPMA

