

YAZILIM LABORATUVARI 2

2. PROJE

Barış KAKILLI

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

baris_kakilli38@hotmail.com

Özet

Bu rapor Yazılım Laboratuvarı - 1 dersi 2. proje için açıklamaya yöneliktir.

Raporda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken geliştirme ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Rapor sonunda projemi hazırlarken kullandığım kaynaklar bulunmaktadır.

1. Proje Tanımı

Projenin amacı mobil programlama hakkında bilgi ve beceri kazanılmasıdır. Projede android ve ios kısıtlaması bulunmamaktadır.

1.1 Projenin Özellikleri

Proje android için gerçekleştirilmiştir. Yazılım dili olarak kotlin, UI olarak da jetpack compose tercihi yapılmıştır.

Projenin Türkçe kelime veritabanında düzenleme yapmak için Python dilinde kısa bir program da yer almaktadır. [Türkçe kelime veritabanı için bu sayfa kullanılmıştır.](#)

2. Kullanıcı Arayüzü

Projede UI(Kullanıcı arayüzü) için jetpack compose tercih edilmiştir. Jetpack compose androidde yeni bir arayüz çeşididir ve android'in eski tarz arayüz stili olan XML'e göre her şey kod ile yapılmaktadır.

Arayüz olmasını istediğiniz öğeleri belirtmek için fonksiyon başına @Composable açıklaması konmalıdır. Bu açıklama fonksiyonun bir arayüz olduğunu belirtir.

Yaşadığım bir sıkıntı bu bir oyun olduğu için sürekli olarak arayüzün güncellenmesi gerekiyor ama uygulama telefonlarda çalışacağı için bu güncelleme işlemini çok hızlı yapamam. O sebepten ötürü arayüz 2 durumda güncelleniyor.

1- Her 50 msde bir

2- kullanıcı bir bloğun üstüne tıkladığı zaman

bu güncelleme kriterlerinde uygulamayı hem kendi telefonumda hem de android studionun sunduğu emulator üzerinden test ettim ve bir takılma yaşanmadığını onayladım.

3. Oyun Döngüsü

Oyun döngüsü için Hücre bilgilerini içeren 10x8lik bir 2 boyutlu dizi oluşturuyorum. Bu hücre bilgileri oyun durumlarına göre her zaman güncelleniyor(bir bloğa tıklandığında renk değişmesi, bloğun havadan düşmesi, blok ile doğru kelime oluşturulması durumunda bloğun diziden silinmesi)

```
data class CellInfo(  
    val row: Int,  
    val col: Int,  
    var letter: Char? = null,  
    var isClicked: Boolean = false,  
    var isfreezing: Boolean = false,  
    var isfreezeblock: Boolean = false,  
    var freezeblockcount: Int = 0  
)  
  
val cells = Array(10) { row ->  
    Array(8) { col ->  
        CellInfo(row, col)  
    }  
}
```

Oyunun ana döngüsünü sağlayan gameLoop isimli fonksiyon Coroutine içerisinde sonsuz bir döngü dönüyor bu döngüyü dönerken telefonda fazla batarya harcamamak için her 150 msde bir dönüyor. Blok düşürme, Buz bloğu işlemleri ve oyunun bittiğini anlama bu döngü içerisinde gerçekleşiyor ve yukarıdaki kod parçasında gözüken 2 boyutlu dizi üzerinde güncellemeler yapıyor. Yapılan bu güncellemeler arayüzde güncelleme etkisi yaratıyor bunun sayesinde kullanıcı yaptığı işlemleri gecikme olmadan görebiliyor.

4. Oyun Mekanikleri

Oyundaki mekaniklerden birisi olan buz bloğu yanındaki blokları buza çeviriyor bu buzla kaplı blokların kırılması için 2 kere kullanılması gerekiyor ilk kullanım buzı kırıyor ikinci kullanım ise bloğu kaldırıyor. Buz blokları bir bloğu birden fazla kez buza çevirebiliyor o sebepten buz bloğu 4 adet bloğu buza çevirdikten sonra özelliğini kaybediyor ve normal bir blok gibi davranıyor.

Oyunun devamlılığının sağlanabilmesi için bloklar düşerken mevcut oyun durumunu göze alarak düşüyor. Eğer mevcut oyun durumunda %40 dan az sesli harf bulunuyorsa Türkçe sesli karakterlerden rastgele birisi düşüyor. Eğer %40 dan fazla ise Türkçe sessiz karakterlerden rastgele birisi düşüyor.

onCellClicked fonksiyonu kullanıcı bloğa tıkladığında yapılan işlemleri kontrol ediyor eğer blok tıklı değilse tıklı yapıyor eğer tıklı ise tekrar tıklamada kaldırıyor.



onDeleteClicked fonksiyonu kullanıcı Ret butonuna tıkladığında mevcut cümledeki her kelimeyi kaldırıyor.

onAcceptClicked fonksiyonu kullanıcı kabul butonuna tıkladığında kelimeyi Türkçe karakter listesinde arıyor eğer mevcutsa kullanıcıya puan veriyor eğer mevcut değilse kullanıcının hatasını 1 yükseltiyor. Kullanıcı hatası 3 olduğunda oyuna her satırdan bir blok düşüyor ve hata tekrardan sıfırlanıyor.



BEY



Current Score: 4

Fails: 1/3

Oyun bittiğinde kullanıcı eski skorlarını en yüksekten en düşüğe sıralı bir şekilde oyun sonu ekranında görüyor. Kullanıcı oyun

bittikten sonra oyuna tekrardan başlayabilir.

Game Over

Your score is 22
Your old scores are 9
4
22
0

Restart

Bloklar Yukarıdan her 5 saniyede bir düşüyor. Eğer kullanıcı 100den fazla skora sahipse 4, 200den fazla skorda 3, 300den fazla skorda 2, 400den fazla skorda 1 saniyede bir düşüyor.

Oyunun Renk Şeması:

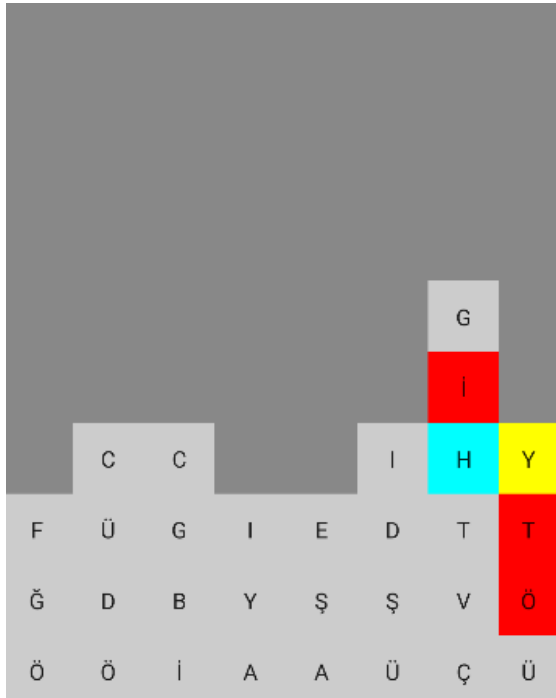
Eğer Blok Tıklıysa: Mavi

Eğer Blok hem buz bloğu hem de buz tutmuşsa: Açık Mavi

Eğer Blok buz bloğuyysa: Sarı

Eğer Blok buz tutmuşsa: Kırmızı

Eğer Blok standart durumda ise: açık gri



Oyunda kelime tanıma sistemi için kelimenin en az 3 harf uzunluğunda ve 2

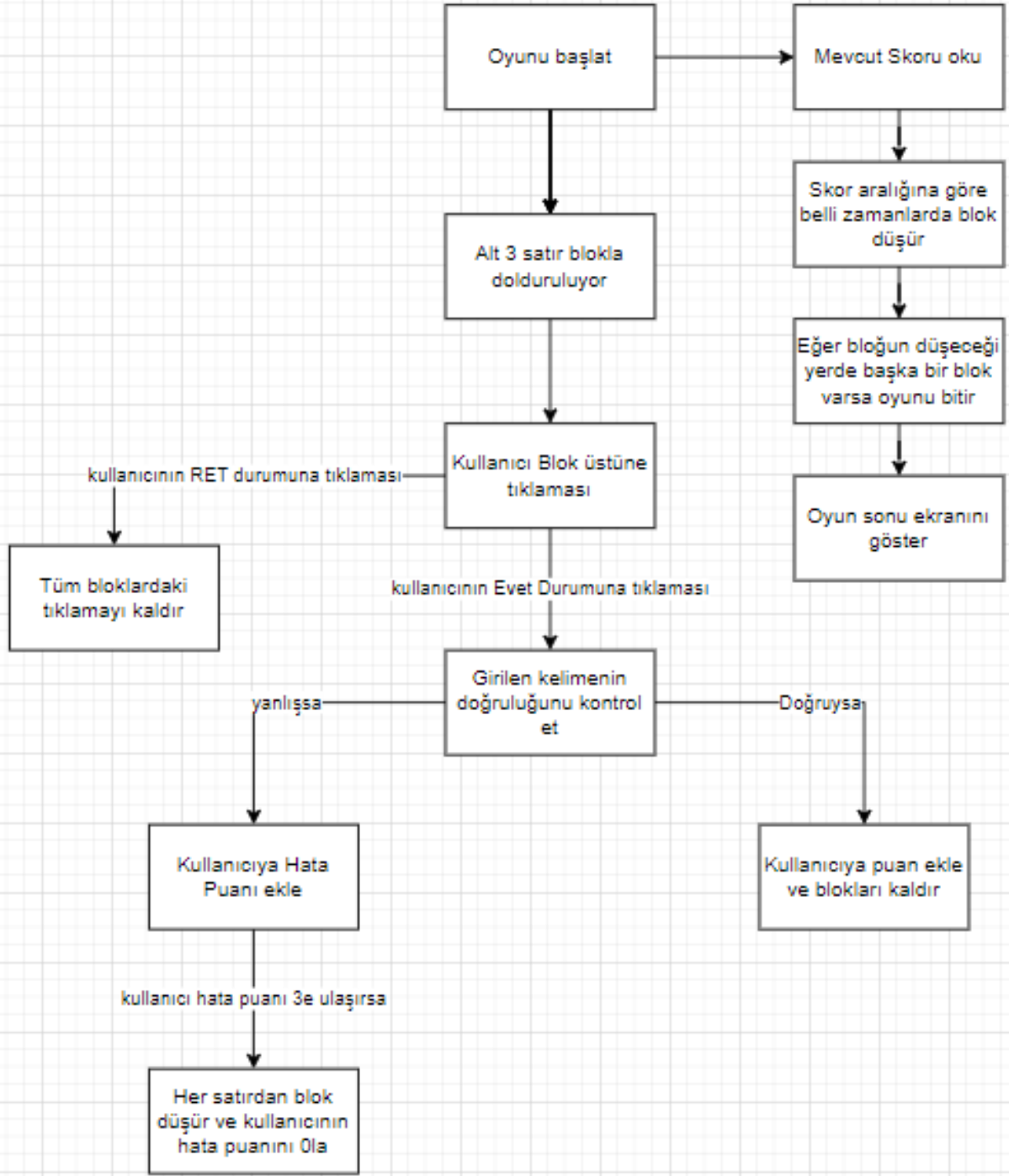
kelimeden oluşmaması gerekiyor bunları sağlamak için kullandığım veritabanındaki güncellemeleri python üzerinde gerçekleştirdim

```
def formatfiles():  
    # read all the text files  
    print("Formatting files...")  
    for file in os.listdir("text"):  
        # open the file  
        with open("text/" + file, "r") as f:  
            new_text = ""  
            # read the file line by line  
            text = f.readlines()  
            for line in text:  
                if len(line) <= 3:  
                    continue  
                if line.__contains__(" ")  
                    continue  
                new_text += line  
            with open("textfiltered/" + file, "w")  
            as f:  
                f.write(new_text)
```

Yukarıdaki python kodu kelimeleri mevcut isterlere uygun şekilde filtreleyip başka bir yere kaydediyor. Kaydedilen kelimeler ise sonrasında oyun içine aktarılıyor. Aktarılan dosya sadece okuma izni olan bir bölüme aktarıldığından kelimelerde değişiklik olması engelleniyor ve istediğim dili kullanma fırsatı elde edebiliyorum.

5. Geliştirme Ortamı

Projeyi Android Studio ve Pycharm kullanıp geliştirdim. Projeyi tek başıma yaptığım için versiyon kontrol sistemi kullanmadım. Arayüz için Compose, Kelimeleri düzenleme için Python geri kalan tüm işlemler için Kotlin kullandım.



Kaynakça

1. Arayüz için
[Jetpack Compose UI App Development Toolkit](#)
2. Türkçe Kelime Veritabanı için
[GitHub - ekartal/turkce-kelime-database](#)

3. karşılaştığım çeşitli problemler:
[Stack Overflow](#)
4. Akış diyagramını oluşturmak için:
[diagrams.net](#)