ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Semestrálna práca z predmetu
Vývoj aplikácií pre mobilné zariadenia

APLIKÁCIA DOOM

vypracoval: Jozef Rechtorík
študijná skupina: 5ZYI24
cvičiaci: doc. Ing. Patrik Hrkút, PhD.
termín cvičenia: streda bloky 1-2

V Žiline dňa 11.6.2024

1) Popis a analýza riešeného problému

Vytvorenie aplikácie o hre **Doom (1993)**, kde bude obsah pre fanúšikov o hre.

Aplikácia pôsobí **moderne**, ovláda sa **intuitívne** a reaguje na **zmenu tmavého** a **svetlého** režimu, taktiež na **otočenie obrazovky**.

Aplikácia na začiatku spustenia zobrazí hlavné menu, v ktorom sú možnosti:

- **MY PLAY** (Priechod hrou po jednotlivých leveloch)
- **LORE** (Príbeh hry)
- **MONSTERS** (Typy nepriateľov, ktorí sa v hre nachádzajú)
- **ID SOFTWARE** (Informácie o vývojároch)
- **INTERESTING FACTS** (Zaujímavosti)
- **QUIZ** (Quiz (10 otázok) na tému Doom)

V obrazovke MY PLAY je možné ku každému levelu pridať nejakú poznámku, potom si ich môže hráč prezrieť, zobrazia sa len tie, kde niečo hráč napísal.

V obrazovke MONSTERS je zoznam všetkých nepriateľov, každý nepriateľ obsahuje animáciu z hry a keď naň používateľ klikne tak sa zobrazí informácia o tom koľko má daný nepriateľ životov a o tom aké poškodenie dokáže hráčovi spôsobiť.

V obrazovke QUIZ je možnosť zahrať si krátky quiz, kde je 10 otázok a za každú sa hráčovi pripočíta jeden bod, na konci je toto skóre zobrazené. Taktiež má hráč na konci možnosť zahrať si quiz znova, alebo ísť do Menu.

Žiadne podobné aplikácie som na Obchode Play nenašiel.

2) Návrh riešenia problému

Vytvoril som 8 obrazoviek. Jedna z nich je hlavné menu, od ktorého je prístup ku ďalším šiestim obrazovkám. Posledná obrazovka je v MY PLAY, kde sa dá prejsť ku poznámkam čo si hráč zapísal.

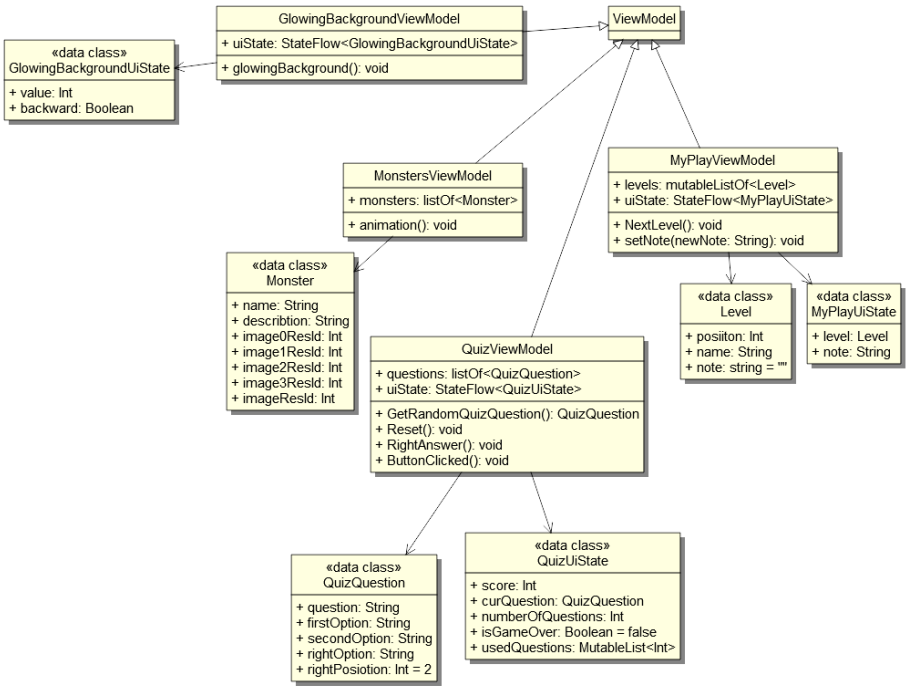
Na zachovávanie stavu v obrazovkách ako napríklad MY PLAY, kde som ako stav určil ako napríklad text ktorý je v textovom poli alebo aktuálny level, som použil dátovú triedu.

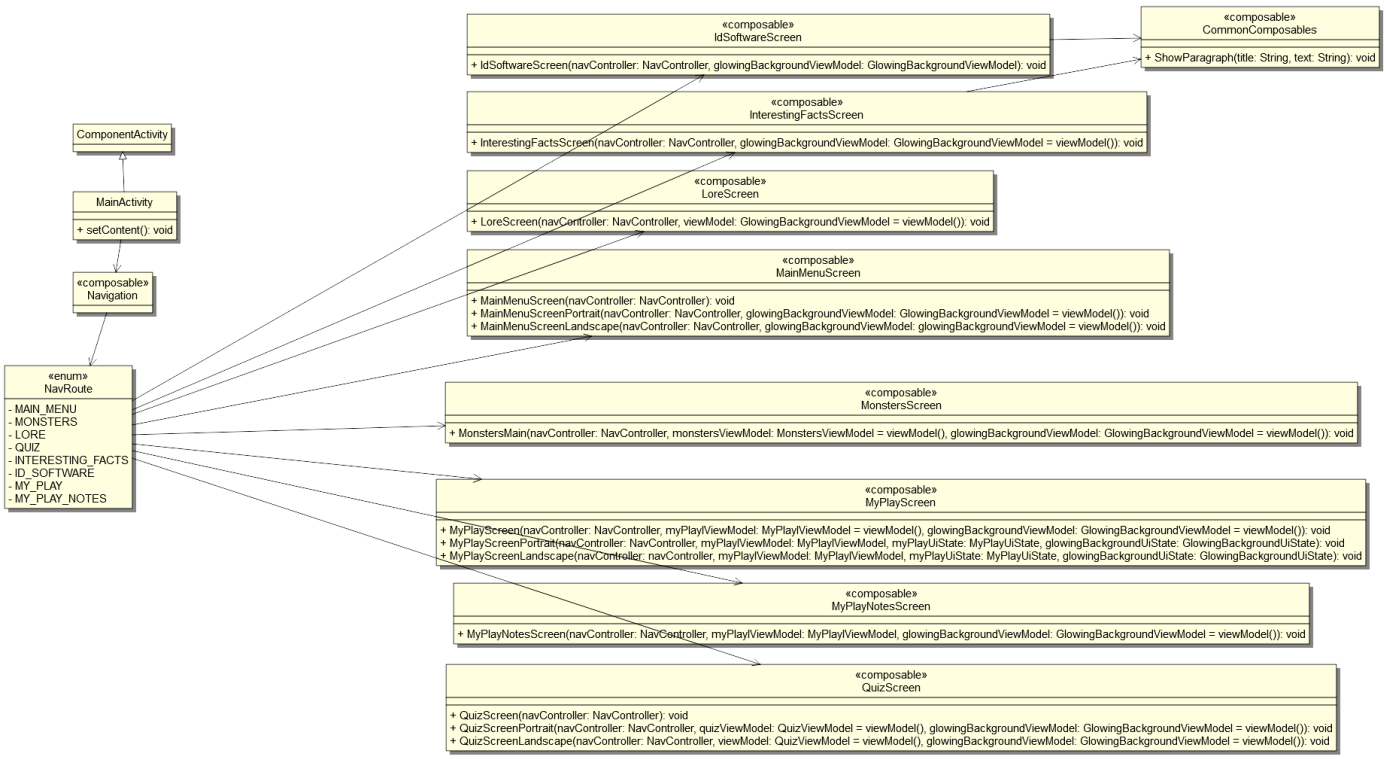
Na oddelenie logiky od UI som použil ViewModel triedy, ktoré sa starali o logiku daných obrazoviek, ako napríklad v obrazovke QUIZ, kde som pripočítal body ak hráč zadal správnu odpoveď.

Na obrazovky, ktoré majú statický obsah, ako napríklad INTERESTING FACTS, som nepoužil samostatný ViewModel, keďže ho nebolo treba, no aj obrazovky so statickým obsahom používajú ViewModel – GlowingBackgroundViewModel, ktorý sa stará o pohyblivé pozadie. To sa aktualizuje 20 krát za sekundu, pomocou korutiny, ktorá beží na tomto ViewModeli.

Informácie o nepriateľoch sú uchované v ViewModeli – MonstersViewModel.

3) UML diagram





4) Popis implementácie

1) **Hlavné menu**:

   Pozostáva z **pohyblivého pozadia,** loga a 6 tlačidiel

2) **MY PLAY**:

   Táto obrazovka obsahuje vlastný **ViewModel**, **UiState** na uchovávanie a pracovanie so **stavom**. Na vrchu obrazovky je **progress bar** ktorý zobrazuje, koľko z hry hráč prešiel. Využil som **LinearProgressIndicator**. Taktiež sa vypíše koľko percent hráč z hry prešiel, názov aktuálneho levelu. Pod ním je tlačidlo na prejdenie na ďalší level, ktoré posunie aktuálny level na nasledujúci a uloží poznámku, taktiež sa aj **aktualizuje UI**. V strede obrazovky je textové pole na zapísanie poznámky, ktorá sa **automaticky ukladá**. Pod týmto textovým poľom je tlačidlo ktoré slúži na zobrazrenie všetkých poznámok, tým že sa presunieme na ďaľšiu obrazovku, ktorá ich bude obsahovať. Poznámky sa ukladajú do **dátovej triedy** Level, ktorá obsahuje atribút „*note*“ – poznámka.

3) **ALL NOTES**:

   Obrazovka obsahuje tlačidlá, ktoré reprezentujú levely, ktorých poznámka nieje prázna. Po ich stlačení sa vypíše to čo si hráč zapísal. Nezobrazia sa tie levely, ktoré nemajú zapísanú poznámku.

4) **LORE**:

   V obrazovke LORE sú použité Text bloky a obrázky. Na spodku sú 3 tlačidlá ktoré reprezentujú kapitoly v hre, každé tlačidlo zobrazí levely, ktoré sa v danej kapitole nachádzajú. Na zobrazenie jednotlivých composables som využil **Lazy Column**, aby som zlepšil **optimalizáciu**.

5) **MONSTERS**:

   V tejto obrazovke sa nachádzajú všetky druhy nepriateľov a taktiež som im **implementoval animáciu**, ktorá sa skladá zo 4 obrázkov. O to aby sa tieto obrázky menili v cykle sa stará **korutina**. Po **stlačení** na **nepriateľa** sa vypíšu informácie o ňom. Na zobrazenie nepriateľov som využil **Lazy Column,** aby sa nevykreslovali nepriatelia ktorí na obrazovke niesú, keďže by sa museli inak animovať aj keď by ich nebolo vidno.

6) **ID SOFTWARE**, **INTERESTING FACTS**:

   V obrazovke ID SOFTWARE a INTERESTING FACTS sú použité Text bloky a obrázky a ViewModel na pohyblivé pozadie.

7) **QUIZ**:

V táto obrazovka taktiež využíva **ViewModel**, **UiState** a k tomu **dátovú triedu** ktorá reprezentuje samotné otázky. Táto obrazovka pozostáva z nadpisu na vrchu obrazovky, pod ním je Textový blok ktorý nesie informáciu o tom koľko hráč získal skóre a na koľkej otázke je. Je to implementované pomocou **dátovej triedy**, ktorá nesie informáciu o **stave** tejto obrazovky. V strede je samotná otázka, ktorá na vypísanie spúšťa **vlastný composable** ktorý som vytvoril. Pod ňou sú 3 tlačidlá, ktoré reprezentujú odpovede na otázku. Po ich stlačení sa zistí či je odpoveď správna a ak áno, **ui stavová trieda sa aktualizuje -** pripočíta skóre a počet otázok, ak nie je správna odpoveď, pripočíta sa iba aktuálny počet otázok. V priebehu quizu sa môže prejsť na hlavné menu, alebo keď hráč zadal desiatu odpoveď, vypíše sa koľko získal skóre a má možnosť hrať znova alebo ísť do hlavného menu.

8) **Prechod medzi obrazovkami**:

Na prechod som využil **NavController**, ktorý ovláda to na akej sme aktuálne obrazovke, samotné cesty (typu String) mám uložené v **enum triede NavRoute**, ktorá obsahuje inštancie ciest ku každej obrazovke. Keď idem naspäť volám metódu **popBackStack()** navControllera.

5) Zoznam použitých zdrojov

Pomôcka pri programovaní: ChatGPT

Využíval som umelú inteligenciu hlavne pri riešení problémov na ktoré som dlho nevedel prísť ako napríklad pri animovaní nepriateľov, keďže sa obrázky menili ale iba keď neboli na obrazovke, vďaka Lazy Columnu, ďalej som umelú inteligenciu využil pri layoute obrazoviek a zarovnávaní composables.

Nepoužíval som **umelú inteligenciu** na **generovanie tried** či **metód**, všetky triedy a balíčky som naprogramoval **sám**.

Zdroje:

Nastavenie ikony:

https://www.youtube.com/watch?v=m6qBOTjZ4Lw

Zbavenie sa bielych častí obrazovky:

https://www.youtube.com/watch?v=pLMw9Vlbfgw

Obrázky:

Nepriatelia:

https://steamcommunity.com/sharedfiles/filedetails/?id=121188304

Id Software logo:

https://en.wikipedia.org/wiki/Id_Software

Commander Keen:

https://cults3d.com/en/3d-model/game/commander-keen-lamp-riktoky

Quake logo:

https://seeklogo.com/vector-logo/399071/quake

Doomguy:

https://myhotposters.com/products/doom-1-old-classic-retro-game-poster-2334

Multiplayer obrázok:

https://www.pngegg.com/en/png-pzwmk

Bill Gates v reklame:

https://www.youtube.com/watch?v=Mni7B4H33OE

John Romero:

https://sk.pinterest.com/pin/24418022953415562/

Mick Gordon:

https://gamerant.com/doom-composer-unlikely-return-future-sequels/

E1M1 speedrun:

https://www.youtube.com/watch?app=desktop&v=ax8yoQtKtLc

Obsah