Fakulta riadenia a informatiky

Informatika

Semestrálna práca

Hra obesenec

2020/2021 Samuel Slivovský, 5ZYI23

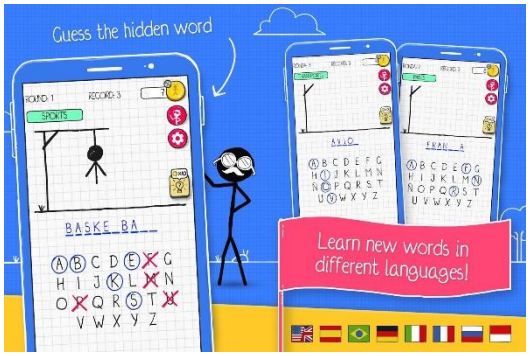
Obsah

Aplikácie s podobným zameraním

Hra obesenec (po anglicky Hangman) je svetom známa hra. Variácií na túto hru v podobe mobilnej aplikácie je niekoľko. Rozhodol som sa vybrať 4 najpopulárnejšie.

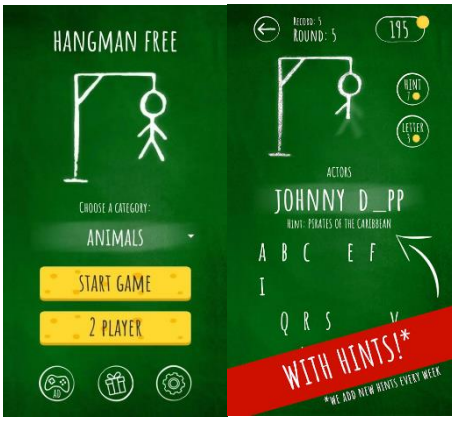
Hangman od spoločnosti senior games:

- Zábavné prevedenie samotnej hry pomocou komiksového štýlu
- Animácie
- Bodový systém s možnosťami odmien
- 8 dostupných jazykov jednotlivých slov na hľadanie
- Možnosť hrania proti ďalšiemu hráčovi (2-player mode)
- Systém skóre



Hangman Free 💯 od spoločnosti SONNAR:

- Prevedenie pripomína školskú tabuľu a kriedu
- Systém ukladania doposiaľ najvyššieho dosiahnutého skóre
- Možnosť nápovedy alebo doplnenie písmena
- 26 rôznych kategórií slov
- Prepojenie so serverom wikipedia v rámci učenia sa významu slov



- **Obesenec - Uhádni slovo od spoločnosti Soodex Labs:**
- Podpora slovenského jazyka a slov
- Unikátne prevedenie
- Systém rebríčka s najlepšími hráčmi
- PvP o mince
- Medaile a trofeje
- Systém powerUpov

Analýza navrhovanej aplikácie

Ako už vyplýva z názvu tak mnou vytvorená aplikácia je reprezentácia hry obesenec. Vzhľadom na jednoduchosť konceptu je aplikácia veľmi intuitívna. Je to ideálna hra na rozšírenie svojej slovnej zásoby alebo na uniknutie nude počas dlhého cestovania. Hra obsahuje systém skóre kde si bude hráč môcť porovnávať svoje predošlé skóre. Hra končí keď padnú životy hráča na 0 alebo nestihne uhádnuť slovo v časomiere, prípadne ju môže ukončiť aj sám. V prípade ak sa hráčovi podarí uhádnuť všetky slová danej obtiažnosti tak dostane bonusové body na základe počtu nepoužitých powerUpov. PowerUpy hráč získava počas hry. Dostupné powerUpy sú:

- Doplnenie písmena
- Zvýšenie života
- Zastavenie času

Po ukončení hry sa hráčove skóre zapíše do databázy. Hra obsahuje aj tabuľku typu highscore kde je top 10 hráčov a taktiež históriu hier týchto hráčov.

Návrh architektúry aplikácie

MainActivity:

Vytvára prvotný fragment TitulkaFragment.

TitulkaFragment:

Fragment titulného zobrazenia. Na pozadí tohto fragmentu je grafické zobrazenie obeseneckej slučky a buttony na prenesenie hráča do fragmentu menu alebo fragmentu s tabuľkou skóre.

MenuFragment:

Fragment menu obsahujúci TextInput kde hráč zadá svoje meno a 3 buttony pre výber obtiažnosti. Pokiaľ hráč nezadá žiadne meno, tak ho hra nepustí na ďalší fragment a upozorní ho na to.

HraFragment:

Najdôležitejší fragment. Obsahuje imageView kde sa striedajú obrázky obesenca vzhľadom na počet životov hráča. Hráč interaktuje pomocou buttonov ktoré predstavujú abecedu, tieto buttony sú obalené v samostatnom layoute aby ich bolo možné schovať. Nachádza sa tu niekoľko rôznych textViews ktoré zobrazujú skóre, životy, čas, počet jednotlivých powerUpov a v prípade prehry alebo uhádnutia všetkých slov vypísanie finálneho textu. Využívam aj 4 floating action buttony (fab). Hlavný fab slúži ako menu powerUpov. Zvyšné 3 fab predstavujú jednotlivé powerUpy. Posledným prvkom sú 2 buttony ktoré sa zobrazia vždy po uhádnutí slova. Hráč si pomocou nich môže vybrať či chce pokračovať alebo skončiť. Pokiaľ hráč prehrá alebo uhádne všetky slová tak sa mu zobrazí len button na ukončenie hry.

Metódy:

- onViewCreated: po vytvorení fragmentu sa do zoznamu slov naplnia slová na základe vybranej obtiažnosti. Zavolá sa metóda novaHra a nastavia sa onTickListenery pre časovače. Taktiež sa tu nastavujú onClickListenery pre všetky buttony fragmentu.
- slovoNaNajdenie: funkcia pre výber náhodného slova zo zoznamu
- updateObr: funkcia pre aktualizovanie obrázku obesenca na základe počtu životov
- novaHra: nastaví atribúty na ich inicializačné hodnoty, pokiaľ sa hráč rozhodne pokračovať tak sa metóda zavolá znova, s tým rozdielom že sa preskočí inicializovanie mena hráča, jeho bodov a životov. Zavolá aj metódu zadaj kde pošle do parametra medzeru aby sa v prípade viac slovných slov automaticky doplnili medzery.
- textInicializacia: funkcia ktorá nainicializuje textView
- doplnPismeno: funkcia powerUpu na doplnenie náhodného písmena hľadaného slova. Pokiaľ by funkcia vybrala medzeru alebo písmeno ktoré už bolo nájdené tak sa rekurzívne zavolá.
- zadaj: funkcia ktorá sa stará o logiku pre zadané písmeno, sleduje aj stav hry a teda rozhoduje či hráč prehral alebo uhádol slovo.
- powerUp: funkcia ktorá sleduje podmienku či má hráč dostať powerUp. Šanca na získanie powerUpu sa mení na základe obtiažnosti.
- nastavButttony: funkcia ktorá nastaví onClickListenery pre všetky buttony abecedy
- boloNajdene: boolean funkcia ktorá pozerá či hráč našiel slovo
- schovajButton: schovanie jednotlivých buttonov po ich použití

- prehra: funkcia ktorá sa zavolá ak hráč prehrá. Odkryje layout s buttonom koniec a text ktorý hárčovi oznámi jeho prehru a prezradí mu slovo ktoré mal nájsť. V prípade že hráč uhádne všetky slová tak zmení text na výherný.
- schovajVsetkyButtony: funkcia ktorá skryje layout s buttonmi
- odkryVsetkyButtony: funkcia ktorá odkryje layout s buttonmi

KoniecFragment:

Fragment ktorý vypíše hráčove dosiahnuté skóre v danom pokuse. Nachádza sa tu buttony pre navigáciu späť na titulku alebo do fragmentu so skóre. Fragment prijíma výsledné body hráča ako bundle ktrorý následne pošle aj s menom hráča do databázy.

SkoreFragment:

Fragment kde si bude hráč môcť pozrieť top 10 skóre. Vo fragmente sa nachádza recyclerView ktorý tieto skóre zobrazuje. Fragment po vytvorení vytiahne z databázy pomocou metódy getBest najlepších hráčov a pošle ich do adaptéru pre recylclerView. Recyler view má v sebe poradové číslo, meno hráča, dátum kedy hráč získal dané skóre a nakoniec aj jeho skóre. Pokiaľ hráč podrží na nejaké meno tak sa mu zobrazí fragment s históriou skóre daného hráča.

FragmentHistoriaHraca:

Fragment v ktorom sa zobrazí história hráča pomocou recylcerView. Ten obsahuje poradové číslo, dátum a získané body. Narozdiel od skoreFragmentu prijíma tento fragment bundle v podobe mena hráča ktoré následne zadá do metódy databázy getHistoriaHraca. Tento zoznam následne pošle do adaptéru pre recyclerView.

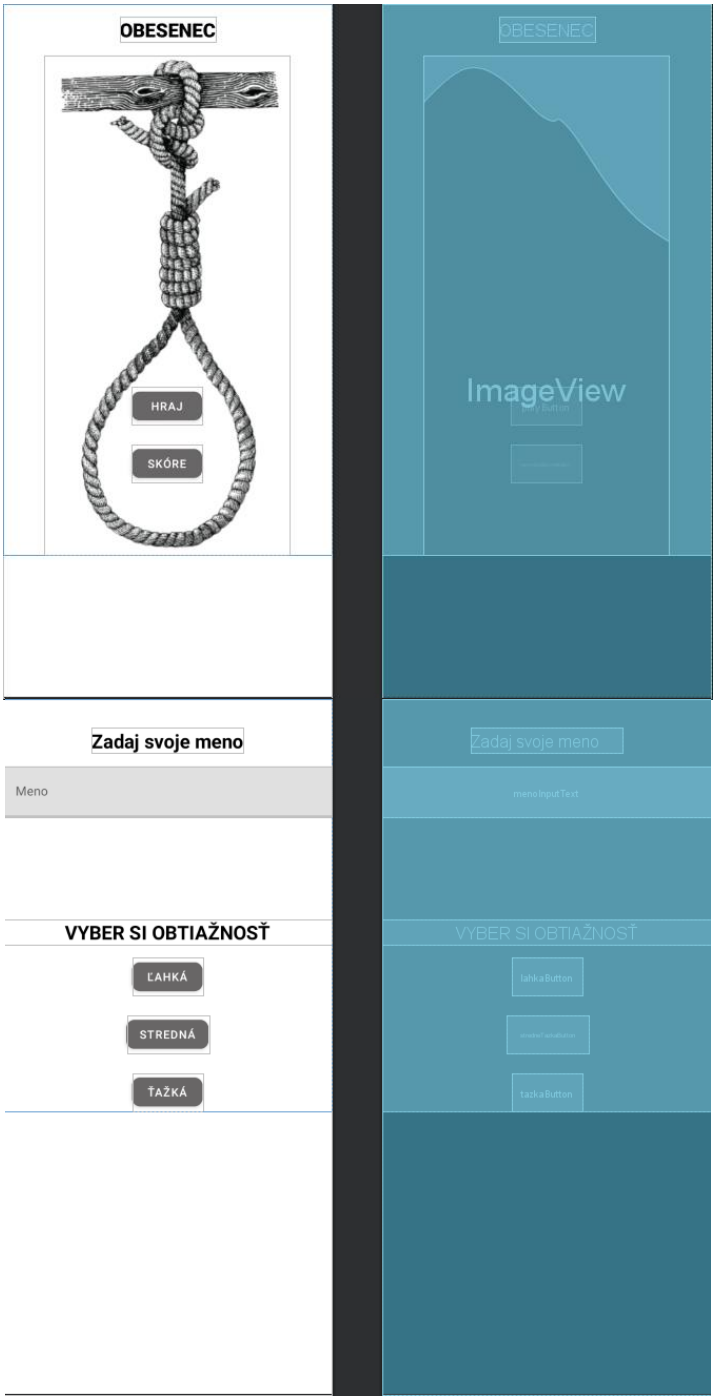
Databáza:

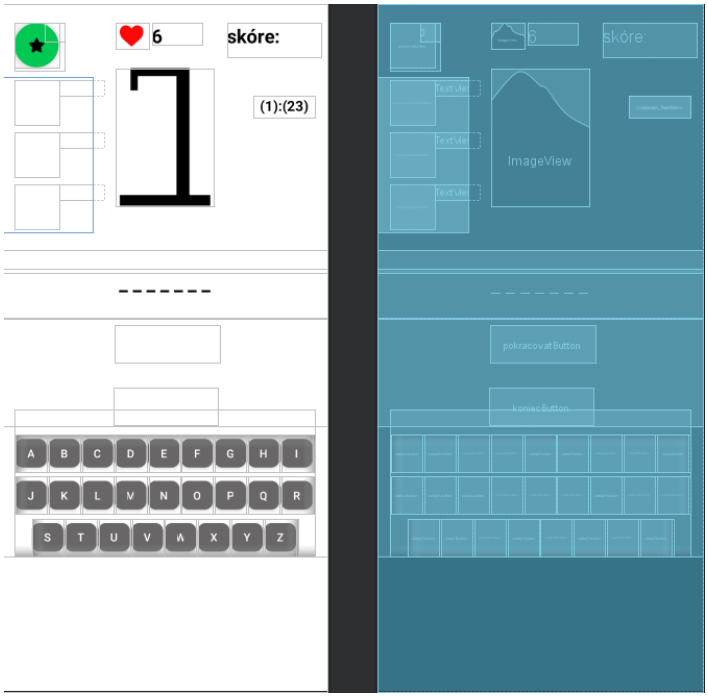
Databázu riešim pomocou troch tried. Trieda Skore v sebe obsahuje jednotlivé atribúty a konštruktor. Trieda SkoreDatabaza túto databázu vytvorí. SkoreDatabazaDao obsahuje jednotlivé funkcie pre insert a gettre.

ViewModely:

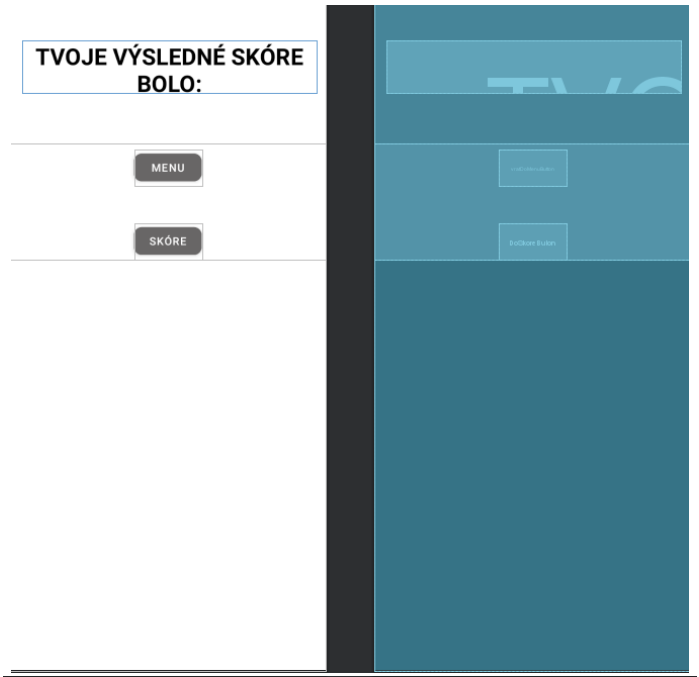
Využil som aj dva viewModely pomocou ktorých získavam určité dáta. UdajeViewModel v sebe drží meno hráča. SlovaViewModel v sebe drží zoznam slov na základe obtiažnosti.

Ukážka návrhu obrazoviek aplikácie

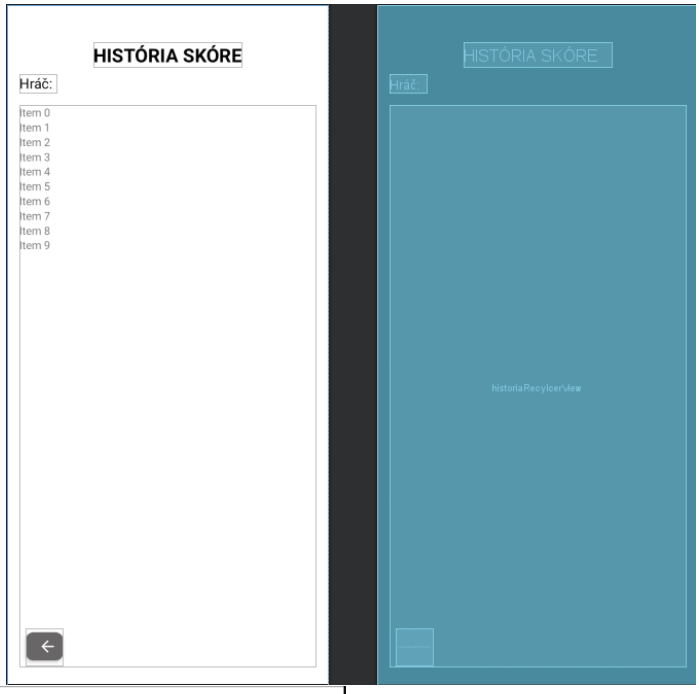



6
skóre:
(1):(23)
ImageView
pokracovat Button
koniec Button


1
4
Skóre: 1
00:43
1
KARTA
POKRAČOVAŤ
KONIEC


TVOJE VÝSLEDNÉ SKÓRE BOLO:
MENU
SKÓRE


TOP 10 HRÁČOV
Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9
TOP 10 HRÁČOV
skoreRecyclerView



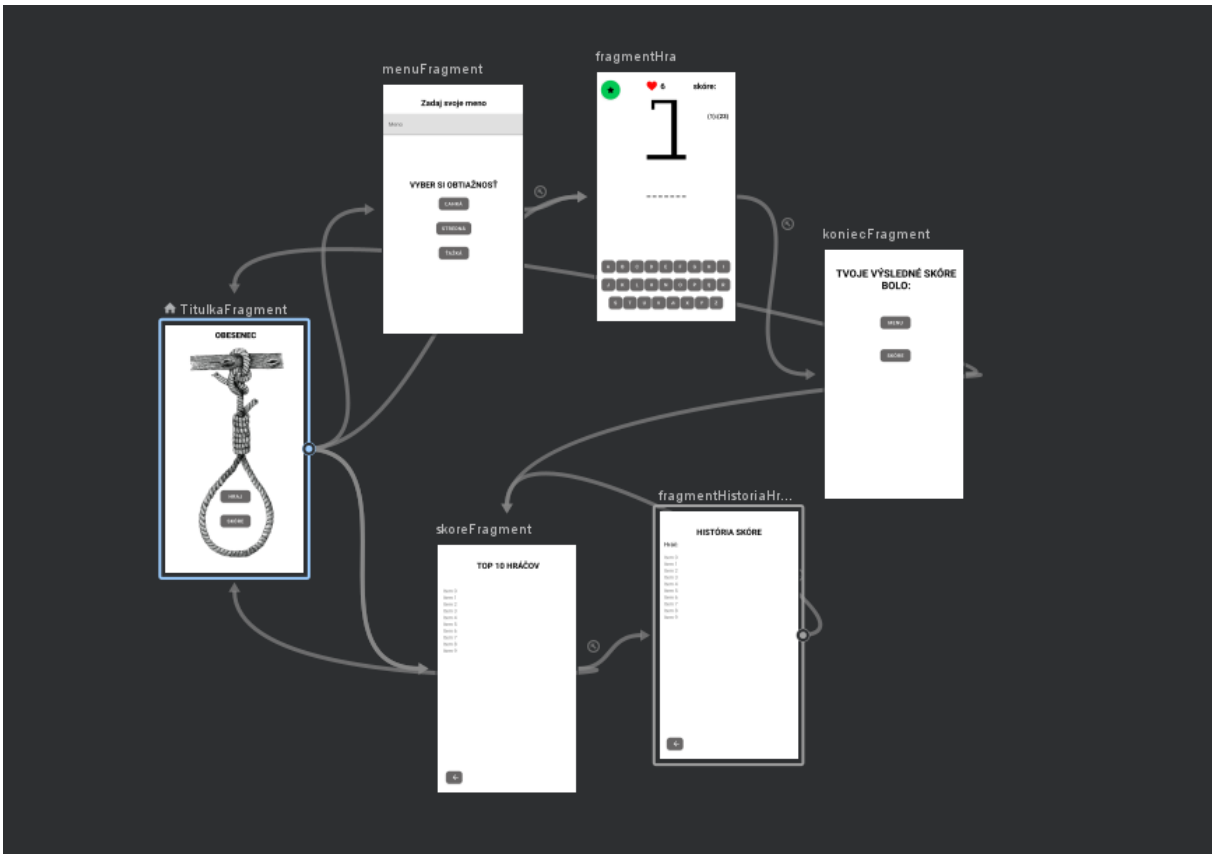
HISTÓRIA SKÓRE

Hráč: Samuel

1	0	24.05.2021
2	1	24.05.2021

TOP 10 HRÁČOV

1	fgfd	22.05.2021	22
2	assa	20.05.2021	17
3	asas	140.05.2021	1
4	sdds	20.05.2021	1
5	Samuel	24.05.2021	1
6	Samuel	24.05.2021	0


menuFragment
Zadaj svoje meno
VYBER SI OBTIAŽNOSŤ
fragmentHra
skóre:
koniecFragment
TVOJE VÝSLEDNÉ SKÓRE BOLO:
TitulkaFragment
OBESENEC
skoreFragment
TOP 10 HRÁČOV
fragmentHistoriaHr...
HISTÓRIA SKÓRE

Ikonka hry