

Hra - 2048



Semestrálna práca z predmetu VAMZ.

Obsah

Popis a analýza riešeného problému	3
Špecifikácia zadania.....	3
Definovanie problému.....	3
Podobné aplikácie a odosobnenie sa	3
Návrh riešenia problému.....	4
Krátka analýza	4
UseCase diagram	4
Stručný popis funkcionality	4
Návrh aplikácie	5
Class diagram.....	5
Prehľad tried.....	6
Stručný popis.....	7
Popis implementácie	8
Aktivity a fragmenty	8
Uloženie konfigurácie.....	8
Dynamický Layout	8
Použitie senzora	8
Notifikácie	8
Resources	8
Vlastný štýl	8

Popis a analýza riešeného problému

Špecifikácia zadania

Úlohou je naprogramovať vlastnú verziu známej mobilnej hry z názvom 2048. Cieľom hry je čo najdlhšie zlučovať prvky matice po nami volených smeroch do doby pokiaľ sa nám celá herná plocha nezaplní. Ak tak nastane hra končí a hráč dúfa že prekoná do tej doby najväčšie nahrané skóre. Skóre vzniká ako súčet zlúčených prvkov ktoré sa podarí hráčovi pomocou úprav v smere zlúčiť. Zlúčiť je možné iba tie prvky matice ktoré sú v danom smere lineárne a majú rovnakú hodnotu. Program by mal obsahovať intuitívne užívateľské rozhranie z prijateľnou voľnosťou pohybu hráča.

Definovanie problému

Problematikou implementácie spomínanej hry je túžba odosobniť sa od iných už vypracovaných riešení ktoré zväčša používajú nachlp rovnakú funkcionálnu a zároveň rovnakú vyhodnocovaciu logiku úpravy matice.

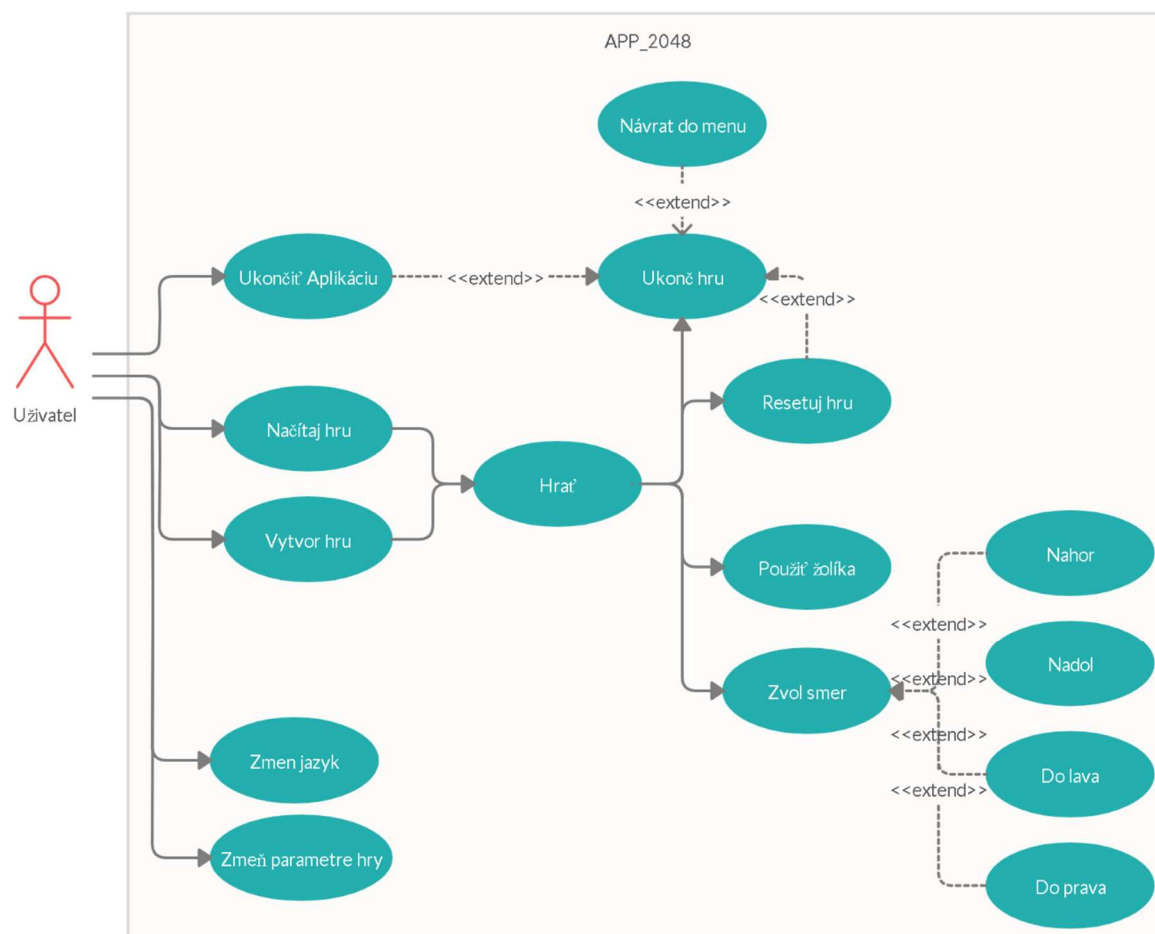
Podobné aplikácie a odosobnenie sa

Ako už bolo spomínané existuje niekoľko zväčša podobných implementácií ktoré sa odlišujú iba v grafike užívateľského rozhrania a hracej plochy. Z toho dôvodu som prijal za vhodné vytvoriť si vlastnú logiku na úpravu hernej matice ktorá síce nieje až tak spoľahlivá ako originálna logika, a to z dôvodu že nebolo možné ju dostatočne dlho testovať a tým pádom sa zrovnávať z danou originálnou logikou je skoro nemožné, no ponúka jemne odlišný spôsob úpravy herného pola. Zároveň druhou hlavnou zmenou oproti pôvodnej verzii je implementácia tzv. žolíka, ktorého môže hráč použiť raz, počas hry. Jeho použitie spôsobí že sa mu vygeneruje (zresetuje) herné pole no zo zachovaným dosiaľ dosiahnutého bodovania. Vďaka tejto implementácii má hráč viacero možností ako dosiahnuť vyššie ohodnotenie ako v pôvodnej hre. Okrem týchto zmien sa tu taktiež nachádza malý systém archivmentov, upozornenie na úspechy daného hráča ktoré počas hry dosiahne.

Návrh riešenia problému

Krátka analýza

UseCase diagram

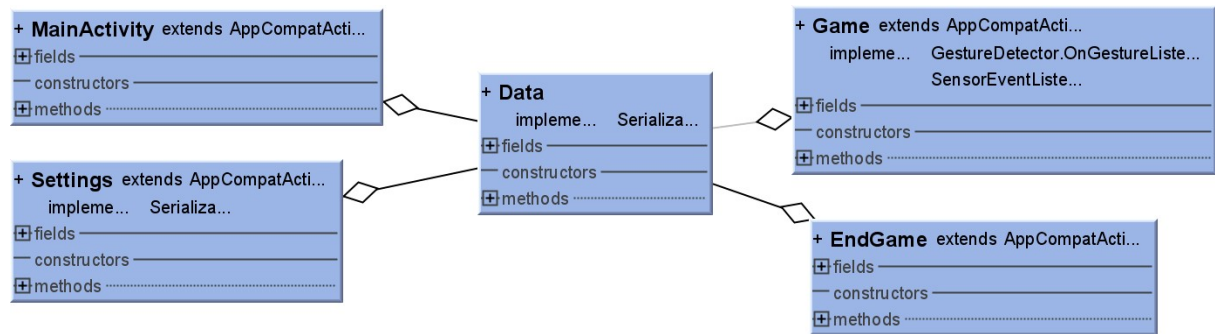


Stručný popis funkcionality

Užívateľ pri otvorení aplikácie je ponuknutý ľahký dizajn užívateľského prostredia ktoré je príjemné na používanie a prehľadné. Na začiatku sú mu predložené štyri možnosti z ktorých má na výber a to konkrétne prejsť na nastavenia, načítať hru, vytvoriť hru a ukončiť ju. V nastaveniach má užívateľ možnosť zmeniť jazyk užívateľského rozhrania a zároveň zmeniť veľkosť herného pola. Následne po vrátení si môže hráč zvoliť či si praje načítať hru alebo začať novú hru. Obidve voľby ho privedú do hry ktorú ovláda na základe pohybu prsta po displeji v osiach X a Y. Ak by sa dostal hráč do "úzkych" môže využiť funkciu žolíka ktorý mu hernú plochu premaže. Túto funkciu aktivuje hráč zatrasením telefónom a môže ju využiť raz za hru. Na danej vrstve sa nachádzajú ešte dve tlačítka, a to reset hry ktorá kompletne premaže hráčov progress, alebo možnosť hry predčasne ukončiť. Po zvolení tejto možnosti sa hráč dostane na posledné okno aplikácie kde je mu predstavené jeho finálne skóre a informácia o tom či pokoril doposiaľ najvyššie skóre. Keď sa hráč informuje, môže hru ukončiť, začať novú hru alebo sa vrátiť do hlavného menu, skade má taktiež možnosť hru ukončiť.

Návrh aplikácie

Class diagram



Prehľad tried

```
+ Game extends AppCompatActivity
    impleme... GestureDetector.OnGestureListe...
                SensorEventListe...

fields
- fin... SETTINGS_REQUEST_CO... :int
- nJoker :Buil...
- nHSc... :Buil...
- zolik :int
- NotifHS :int
- acel... :fl...
- acelL... :fl...
- shake :fl...
- sm :SensorMana...
- newGa... :Butt...
- endGa... :Butt...
- dt :Data
- score :TextVi...
- best :TextVi...
- cSc... :TextVi...
- cBest :TextVi...
- gameFieldT... :TextView...
- activ... :Activ...
~ fin... MIN_DISTAN... :int
- downX :fl...
- downY :fl...
- upX :fl...
- upY :fl...
- gestureDetec... :GestureDetec...
- newGameListe... :OnClickListe...
- endGameListe... :OnClickListe...

constructors

methods
# onSaveInstanceState... (outSta... Bun... ):void
# onRestoreInstanceState... (savedInstanceState... Bun... ):void
+ pridajPrvok():void
+ testKoniec... ():boole...
+ swictc... (paVol... int):boole...
- smerHoreTest():boole...
- smerDoleTest():boole...
- smerPravaTest():boole...
- smerLavaTest():boole...
+ doSmer(paVol... int):void
- smerHore():void
- smerDole():void
- smerPrava():void
- smerLava():void
- upravSc... (paA:int, paB:int):void
+ upravSc... (paZvacsi... int):void
+ setGame():void
+ vytvorNovu():void
+ upravZobrazovaciPlo... ():void
+ vykonaj(tmp:int):void
~ Zmen():void
- ukonc():void
# onActivityRes... (requestCode:int, resultCode:int, data:Intent):void
+ onBackPressed():void
# onCreate(savedInstanceState... Bun... ):void
+ onTouchEv... (event:MotionEv... ):boole...
+ onDown(e:MotionEv... ):boole...
+ onShowPress(e:MotionEv... ):void
+ onSingleTa... (e:MotionEv... ):boole...
+ onScr... (e1:MotionEv..., e2:MotionEv..., distanc... fl..., distanc... fl... ):boole...
+ onLongPress(e:MotionEv... ):void
+ onFli... (downEve... MotionEv..., moveEve... MotionEv..., velocit... fl..., velocit... fl... ):boole...
+ onSensorChang... (event:SensorEvent):void
+ onAccuracyChan... (sensor:Sensor, accura... int):void
# notificationJo... ():void
# notificationHighS... ():void
- createNotificationChann... ():void
- createNotificationChan... ():void
```

```
+ Data
    impleme... Serializa...

fields
- jazyk:int
- pouzitieZol... :boole...
- existujeHra :boole...
- Hra:int...
- hraSi... :int
- Score:int
- high2... :int
- high3... :int
- high4... :int

constructors

methods
~ setJazyk(tmp:int):void
~ getJazyk():int
~ setExit... (tmp:boole... ):void
~ getExist... ():boole...
~ setScore(tmp:int):void
~ getSc... ():int
~ setHigh2x2(tmp:int):void
~ getHigh2... ():int
~ setHigh3x3(tmp:int):void
~ getHigh3... ():int
~ setHigh4x4(tmp:int):void
~ getHigh4... ():int
~ setType(druh:int):void
~ getTy... ():int
~ setZolik(tmp:boole... ):void
~ getZolik():boole...
~ vytvorPoleHry():void
~ setPrvokPola(Hodno... int, x:int, y:int):void
~ getHraProk(x:int, y:int):int
```

```
+ EndGame extends AppCompatActivity
fields
- nHScoreWJ :Buil...
- dt :Data
- bNewGa... :Butt...
- bMainMe... :Butt...
- bEndGa... :Butt...
- tHighSc... :TextVi...
- tScoreVal... :TextVi...
- tSc... :TextVi...
- bNewGameOnClickListe... :OnClickListe...
- bMainMenuOnClickList... :OnClickListe...
- bEndGameOnClickListe... :OnClickListe...

constructors

methods
# onSaveInstanceState... (outSta... Bun... ):void
# onRestoreInstanceState... (savedInstanceState... Bun... ):void
+ onBackPressed():void
~ Zmen():void
# onCreate(savedInstanceState... Bun... ):void
# notificationHighS... ():void
- createNotificationChann... ():void
```

```
+ MainActivity extends AppCompatActivity
fields
- fin... SETTINGS_REQUEST_CO... :int
- close :Butt...
- settings :Butt...
- newGa... :Butt...
- loadGa... :Butt...
- action... :Action...
- dt :Data
- closeOnClickListe... :OnClickListe...
- settingsOnClickListe... :OnClickListe...
- newGameOnClickListe... :OnClickListe...
- loadGameOnClickListe... :OnClickListe...

constructors

methods
# onSaveInstanceState... (outSta... Bun... ):void
# onRestoreInstanceState... (savedInstanceState... Bun... ):void
- Zmen():void
- settingActi... ():void
# onActivityRes... (requestCode:int, resultCode:int, data:Intent):void
- openGa... (Druh:int):void
- openGa... ():void
+ onBackPressed():void
# onCreate(savedInstanceState... Bun... ):void
```

```
+ Settings extends AppCompatActivity
    impleme... Serializa...

fields
- dt :Data
- rGroup :RadioGro...
- rButton :RadioButt...
- SVK :Butt...
- ENG :Butt...
- Back :Butt...
- LANG :TextVi...
- GAME :TextVi...
- svkOnClickListe... :OnClickListe...
- engOnClickListe... :OnClickListe...
- backListe... :OnClickListe...

constructors

methods
# onSaveInstanceState... (outSta... Bun... ):void
# onRestoreInstanceState... (savedInstanceState... Bun... ):void
~ Zmen():void
+ onBackPressed():void
# onCreate(savedInstanceState... Bun... ):void
```

Stručný popis

Class MainActivity

Main funguje ako rozcestník ktorý je predstavený užívateľovi na začiatku hry a prostredníctvom ktorého sa rozhodne a určí čo chce z aplikáciou vykonávať.

Class Settings

Trieda nastavení slúži používateľovi na nastavenie parametrov hry a užívateľského rozhrania.

Class Data

Táto trieda nám slúži ako zberník na dáta v ktorom sú uložené všetky informácie potrebné k bezproblémovému priebehu hry.

Class Game

Najdôležitejšia trieda aplikácie v ktorej sa okrem procesov pre hru nachádza aj celá logika ako sa má hra vykonávať.

Class EndGame

EndGame zobrazuje výsledky hry a zároveň pomáha užívateľovi sa rozhodnúť či chce sa vrátiť do hlavnej ponuky, ukončiť hru alebo začať novú hru.

Popis implementácie

Aktivity a fragmenty

Aplikácia obsahuje štyri hlavné aktivity pomocou ktorých sme schopný splniť všetky požiadavky užívateľa na funkcionálnu danej aplikácie. Sú nimi hlavné menu, nastavenia, aktivita pre hru a ukončenie.

Uloženie konfigurácie

Z dôvodu správneho fungovania otáčania displeja bolo potrebné dákym spôsobom uložiť pred danou akciou dáta každej aktivity a následne po otočení displeja ich znova implementovať. V prípade ak by sme tak neurobili dochádzalo ku problémom z prekladom textov, zobrazovanými dátami a v niektorých prípadoch to mohlo viesť až ku pádu aplikácie.

Dynamický Layout

V prípade inteligentnejšieho zadefinovania hracej plochy bolo nutné si zadefinovať dve dynamické polia, jedno horizontálne a jedno vertikálne ktoré nám pomáhajú vykreslovať hraciu plochu bez nutnosti zadefinovania každej jednej verzie hracej plochy. Najskôr sa vytvoria riadky hracej plochy ktoré obalujú jednotlivé stĺpce ktoré sa následne do nich vkladajú a zároveň práve tieto stĺpce slúžia na zobrazenie hodnoty pre dané políčko v poli.

Použitie senzora

Senzory alebo senzor bol využitý pri tvorbe bonusu pre hráča (žolíka) ktorého aktivuje zatrasením telefónu z využitím metódy `onSensorChange`. Okrem neho sa dá považovať za senzor aj samostatný display ktorý slúži na ovládanie hry pomocou pohybu prsta po sklíčku mobila (Swipe).

Notifikácie

Notifikácie zastávajú rolu odmienu pre užívateľa alebo tzv. achievementov kedy pri určitých udalostiach ako sú napr. prekročenie doterajšieho najvyššieho skóre, minúť žolíka a tak podobne, informujú užívateľa že sa niečo udialo.

Resources

Všetky využité zdroje (prevážne obrázky v rôznych formách ako notification icon a pod.) sa nachádzajú v zložke `res` vo vytvorenej aplikácii a v prípade že by programátor ich chcel neskôr využiť a hlavne upraviť, nachádzajú sa aj v domovskom adresári v zložke `resources` kde sú v pôvodnom formáte a teda `png`.

Vlastný štýl

Definovanie vlastného štýlu bolo využité prevažne z dôvodu neskoršej úpravy grafického rozhrania a aj z dôvodu budúcich úprav, kedy programátor nemusí mať až takú rozšírenú znalosť jednotlivých aktivít v prípade že bude chcieť zmeniť vzhľad aplikácie.