



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

**Fakulta elektrotechnická
Katedra počítačů**

Bakalářská práce

Rozšíření mobilní aplikace pro podporu odvykání kouření

Bakalářská práce

Václav Šídlo

Květen 2020

Vedoucí práce: Ing. Václav Burda

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šídlo** Jméno: **Václav** Osobní číslo: **474410**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Otevřená informatika**
Studijní obor: **Software**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Rozšíření mobilní aplikace pro podporu odvykání kouření

Název bakalářské práce anglicky:

Expansion of mobile app to support smoking cessation

Pokyny pro vypracování:

1. Prostudujte existující literaturu věnovanou metodám podpory odvykání kouření na mobilních zařízeních. Seznamte se s metodami, které mohou pomoci kuřákům s rozptýlením při chuti na cigaretu.
2. Seznamte se s problematikou tvorby mobilních aplikací pro OS Android.
3. Vyberte vhodné metody z bodu 1. a implementujte je v mobilní aplikaci.
4. Otestujte funkčnost na minimálně 5 uživatelích.

Seznam doporučené literatury:

- [1] Ubhi, H.K., Kotz, D., Michie, S., van Schayck, O.C.P., Sheard, D., Selladurai, A., West, R. (2016) Comparative analysis of smoking cessation smartphone applications available in 2012 versus 2014. Addictive behaviors, 58, 175-181. doi: 10.1016/j.addbeh.2016.02.026
- [2] Michie, S., Hyder, N., Wallia, A. & West, R. (2011). Development of a taxonomy of behaviour change techniques used in individual behavioural support for smoking cessation. Addictive behaviors, 36(4), 315-319. doi: 10.1016/j.addbeh.2010.11.016
- [3] Šálená, A. (2018). Mapování uživatelsky dostupných aplikací pro odvykání kouření. Bakalářská práce, 59. Praha: Univerzita Karlova v Praze, 1. lékařská fakulta, Klinika adiktologie 1. LF UK v Praze.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Václav Burda, Minion Interactive s.r.o., Jablonec nad Nisou

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **03.02.2020** Termín odevzdání bakalářské práce: **22.05.2020**

Platnost zadání bakalářské práce: **30.09.2021**

Ing. Václav Burda
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování / Prohlášení

Úvodem bych rád poděkoval své rodině a přátelům za podporu během celého studia. Dále děkuji svému vedoucímu Ing. Václavu Burdovi a supervizoru doc. Ing. Danielu Novákovi, Ph.D. za veškerou podporu a cenné rady při vypracovávání této práce.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 22. 5. 2020

.....

Abstrakt / Abstract

Tato bakalářská práce se zabývá návrhem a implementací rozšíření existující mobilní aplikace pro podporu odvykání kouření o další metodu rozptýlení pro odvykající kuřáky ve chvílích, kdy mají největší chuť na cigaretu. Odreagování má podpořit úspěšnost odvykání. V rámci analýzy druhů rozptýlení byl zvolen a implementován modul minihér. Modul obsahuje 6 her, konkrétně Piškvorky, Reversi, 2048, Labyrint, Kuličky a Bublíny. Pro hry Piškvorky a Reversi byl navržen a implementován umělý protivráč. Pro hry Labyrint, Kuličky a Bublíny bylo nutné navrhnout a implementovat logiku pohybu a kolizí jednotlivých koulí na herní ploše. Na závěr práce je popsána metodika a výsledky uživatelského testování.

Klíčová slova: mobilní aplikace, android, kouření, odvykání, hry, piškvorky, reversi, 2048, kuličky, bubliny, labyrint

This bachelor thesis focuses on designing and implementing an extension of an existing mobile application that supports smoking cessation, providing yet another method of distraction for quitting smokers in times when they are most craving for a cigarette. This distraction should increase the success rate of giving up smoking. As a part of an analysis of different kinds of distractions, a minigames module was chosen and implemented. The module consists of the following 6 minigames: tic-tac-toe, reversi, 2048, labyrinth, marbles, and bubbles. For the tic-tac-toe and reversi minigames, an artificial opponent was implemented. For the labyrinth, marbles, and bubbles minigames, there was a need for designing and implementing the logic of movement and collisions of the individual balls on the game area. The last part of the thesis describes the method and results of user testing.

Keywords: mobile application, android, smoking, cessation, games, tic-tac-toe, othello, 2048, marbles, bubbles, labyrinth

Title translation: Extension of mobile app that supports smoking cessation (Bachelor thesis)

Obsah /

1 Úvod	1
1.1 Motivace	2
1.2 Cíle	2
2 Analýza technik rozptýlení	3
2.1 Dostupné techniky	3
2.1.1 Technika motivace	3
2.1.2 Technika maximalizace sebekontroly	4
2.1.3 Technika alternativ- ních činností	4
2.2 Implementace technik ve stá- vajících aplikaci	4
2.2.1 Volba techniky	5
2.2.2 Volba miniher	5
3 Volba platformy	6
3.1 Volba operačního systému	6
3.2 Volba nejnižší podporované verze	6
4 Android	8
4.1 Historie	8
4.2 Programovací jazyky	8
4.2.1 Java	8
4.2.2 Kotlin	8
4.2.3 Porovnání	9
4.2.4 Výběr jazyka	9
4.3 Základní elementy	9
4.3.1 Aktivita	9
4.3.2 Služba	9
4.3.3 Přijímač vysílání	10
4.3.4 Poskytovatel obsahu	10
4.3.5 Fragment	10
4.3.6 Android Manifest	11
4.4 Uživatelské rozhraní	11
4.4.1 Layout	11
4.4.2 View	11
4.4.3 Canvas	12
4.5 Animace	12
4.6 Persistence dat	12
4.6.1 Shared Preferences	12
4.6.2 SQLite databáze	12
4.7 Vybraná rozhraní	13
4.7.1 Akcelerometr	13
4.7.2 Detektor škálování ob- razovky	13
5 Návrh a implementace	15
5.1 Hra 2048	15
5.1.1 Existující řešení	15
5.1.2 Návrh funkcionality a uživatelského rozhraní ...	16
5.1.3 Grafický návrh	16
5.1.4 Implementace	17
5.2 Hra Reversi	18
5.2.1 Návrh funkcionality a uživatelského rozhraní ...	19
5.2.2 Grafický návrh	19
5.2.3 Implementace	20
5.3 Hra Piškvorky	21
5.3.1 Návrh funkcionality a uživatelského rozhraní ...	21
5.3.2 Grafický návrh	21
5.3.3 Implementace	21
5.4 Hra Bublíny	22
5.4.1 Existující řešení	22
5.4.2 Návrh funkcionality a uživatelského rozhraní ...	23
5.4.3 Grafický návrh	23
5.4.4 Implementace	24
5.5 Hry Kulička a Labyrint	26
5.5.1 Návrh funkcionality	26
5.5.2 Grafický návrh	26
5.5.3 Implementace	27
6 Umělý protihráč pro hru Piškvorky	30
6.1 Popis algoritmu	30
6.1.1 Heuristiky	31
6.1.2 Ohodnocovací funkce ...	32
6.2 Implementace	32
6.2.1 Paralelizace	32
7 Uživatelské testování	34
7.1 Průběh a uživatelé	34
7.1.1 Testování uživatelé	34
7.1.2 Testovací dotazník	36
7.2 Vyhodnocení	36
8 Závěr	38
Literatura	39
A Testovací dotazník	43
A.1 O Aplikaci	43
A.2 Zadání	43
A.3 Otázky	43
B Příložené soubory	47

Tabulky / Obrázky

3.1. Zastoupení mobilních OS na trhu za poslední 4 roky	6
3.2. Zastoupení jednotlivých verzí OS Android na trhu v září 2019 ..	7
6.1. Definované pozice a jejich hodnoty pro ohodnocovací funkci ve hře Piškvorky	32
1.1. Přehled fází odvykání	1
4.1. Životní cyklus aktivity v OS Android	10
5.1. Ukázka oficiální aplikace hry 2048	15
5.2. Ukázka modifikace hry 2048 ..	15
5.3. Ukázka vzhledu prototypu hry 2048	16
5.4. Doménový model hry 2048	17
5.5. Finální vzhled hry 2048	18
5.6. Ukázka vzhledu prototypu hry Reversi	19
5.7. Finální vzhled hry Reversi	21
5.8. Finální vzhled hry Piškvorky ..	22
5.9. Ukázka hry Bubble Shooter ...	23
5.10. Ukázka hry Bubble Shooter Space	23
5.11. Ukázka vzhledu prototypu hry Bublíny	24
5.12. Doménový model hry Bublíny .	25
5.13. Finální vzhled hry Bublíny	26
5.14. Ukázka prototypu hry Kulička	27
5.15. Ukázka prototypu hry Labyrint	27
5.16. Doménový model her Kulička a Labyrint	28
5.17. Ukázka obrazovky pro kalibraci Gyroskopu ve hrách Kulička a Labyrint	29
5.18. Finální vzhled hry Kulička	29
5.19. Finální vzhled hry Labyrint ...	29
6.1. Ukázka jednotlivých vrstev prohledávání v algoritmu Minimax	30
7.1. Graf se zastoupením věkových skupin	34
7.2. Graf se zastoupením pohlaví ..	35
7.3. Graf se zastoupením uživatelů OS Android	35
7.4. Graf s pokročilostí uživatelů v používání OS Android	35

Kapitola 1

Úvod

Rozšiřovaná aplikace[1] je mobilní aplikace pro podporu odvykání kouření založená na klinických studiích a praxi. Aplikace byla vyvinuta ve spolupráci s Klinikou adiktologie 1. lékařské fakulty Univerzity Karlovy v Praze. Proces odvykání ve spolupráci s touto aplikací je rozdělen do 3 fází o celkové délce trvání 2 měsíců a každodenním 5 – 10 minutovým terapeutickým sezením s virtuálním adiktologem Adamem, viz obrázek 1.1.



Obrázek 1.1. Přehled fází odvykání

Jednotlivé fáze jsou navrženy tak, aby maximálně připravily a pomohly kuřákovi s odvykáním kouření. První fáze začíná tím, že kuřák je informován o tom, že přesně za 10 dní přestává kouřit. V průběhu jednotlivých terapeutických sezení, které jsou každý den a trvají zpravidla 5 – 10 minut, je kuřák připravován na proces odvykání. Kuřák se formou vhodných terapeutických otázek a informací o zkušenostech dozvídá, jaké nástrahy jej při odvykání čekají a s čím mohou s vysokou pravděpodobností nastat problémy. Kuřák si během jednotlivých sezení vytvoří plán, co bude dělat v kritických situacích při chuti na cigaretu. Promyslí a sepíše si, z jakých důvodů chce přestat kouřit a co mu konec kouření přinese. K těmto záznamům se pak může ve chvílích největší chuti na cigaretu vrátit a překonat je tak. První fáze je zároveň fází, ve které dochází k největší interakci s kuřákem.

Druhou fází je takzvaný Den D. Jedná se o první den, který by měl kuřák zvládnout zcela bez cigarety. V této fázi zůstává, dokud to nezvládne a na konci dne nepotvrdí

svůj úspěch. V případě, že to nezvládne, spouští se mu speciální sezení, které má za cíl zjistit, proč selhal a přesvědčit ho o tom, aby to nevzdal a zkusil znovu.

Třetí fáze trvá 6 týdnů a jedná se o kritickou fázi, ve které je pro odvykajícího kuřáka nejtěžší se nevrátit k cigaretě. V této fázi je úkolem aplikace odvykajícího kuřáka podpořit a utvrdit jej v tom, že odvykání je správná věc. Aplikace k tomu využívá několik technik[2]. Technikami jsou motivace, maximalizace sebekontroly a poskytnutí alternativních činností. Motivace a maximalizace sebekontroly je v aplikaci řešena samotnou terapií. Poskytnutí alternativních činností je řešeno tzv. SOS tlačítkem.

SOS tlačítko obsahuje možnost zobrazit rychlé tipy, jak zabavit ruce či ústa tak, aby kuřák mohl překonat svou chuť na cigaretu. Dále obsahuje možnost rychle zavolat někomu ze svých blízkých, aby si kuřák mohl popovídat a odvést myšlenky od kouření.

1.1 Motivace

Ačkoliv kouření zabíjí ročně přes 6 milionů lidí a způsobuje miliardové škody[3], celosvětově kouří více než miliarda lidí[4]. Přes veškerá negativa i to, že až 92 % kuřáků se někdy pokusilo přestat kouřit, jen 8 % kuřáků vydrželo déle než 3 měsíce[5].

Vzhledem k ambicím aplikace účinně pomáhat kuřákům s odvykáním je nutné, aby aplikace využívala veškeré dostupné a účinné terapeutické techniky. Rozšíření aplikace o metody rozptýlení v kritické fázi odvykání může zvýšit procento kuřáků, kteří svou závislost překonají[6].

1.2 Cíle

Cílem práce je provést analýzu dostupných technik rozptýlení pro kuřáky v kritické fázi odvykání. Na základě analýzy zvolit vhodnou techniku pro aplikaci a implementovat ji.

Kapitola 2

Analýza technik rozptýlení

Obsahem této kapitoly je analýza technik a metod rozptýlení, které jsou porovnány. Následně je z nich vybrána technika, která bude dále v této práci implementována.

2.1 Dostupné techniky

Základním kamenem podpory odvykání kouření jsou takzvané techniky změny chování. Techniky změny chování jsou komponenty terapie, jejichž cílem je trvalá změna chování jedince, na kterého jsou aplikovány[7]. Tyto techniky se využívají ve všech odvětvích léčebných kúr. Například v alkoholové odvykací terapii či hubnutí.

Aby byly techniky pro odvykání kouření účinné, měly by podporovat změnu kuřákovy myšlenky ve smyslu redukce impulzů a příležitostí, při kterých si chce zapálit a podporovat motivaci a dovednost těmto impulzům vzdorovat[2]. Techniky by tedy měly podporovat motivaci kuřáka přestat kouřit, učit jej rozpoznávat situace, které přecházejí chuti na cigaretu a efektivně je zvládat. Stejně tak by měly podporovat sociální okolí kuřáka v jeho motivaci[8].

Klasifikací technik podporujících odvykání kouření se zabývala studie[2], která porovnávala příručky oddělení pro odvykání kouření a definovala hlavní kategorie těchto technik. Jedná se o techniky motivace, maximalizace sebekontroly a poskytnutí alternativních činností[2]. Jednotlivé techniky jsou detailně popsány v následujících podkapitolách.

2.1.1 Technika motivace

Motivace je jedním ze základních stavebních kamenů odvykání kouření. Motivace má zásadní vliv na rozhodnutí přestat kouřit. 70 % kuřáků věří, že zásadní podmínkou k tomu, aby mohli přestat kouřit, je odhodlání přestat a víra, že to opravdu dokáží. Naopak pocit, že k odvykání potřebují cizí pomoc, je u nich vnímán jako projev vlastní slabosti[9]. Z dlouhodobého hlediska však vysoká počáteční motivace neznamená, že bude mít kuřák větší šanci stát se trvale nekuřákem[10]. Důležité totiž je udržet motivaci abstinujícího kuřáka na vysoké úrovni v průběhu odvykání tak, aby se k cigaretě v nějaké krizové situaci nevrátil, neboť motivace se silně odvíjí od toho, jak se člověk v danou chvíli cítí[8].

Pro efektivní podporu odvykání je tedy třeba, aby aplikace udržovala vysokou míru motivace abstinujícího kuřáka. Tuto míru motivace může poskytovat několika způsoby. Především je nutné připomínání, proč kuřák přestal kouřit, jaké dlouhodobé výhody má nekouření či zdravotní aspekty, jak pro samotného kuřáka, tak pro jeho okolí, například děti. Další možností může být jistá forma odměn pomocí ocenění v aplikaci, za splnění jednotlivých cílů, nebo pochvala a zhodnocení dosavadních úspěchů. Terapie se může ale zaměřovat i na vytvoření obrazu sebe sama po dokončení terapie a jaké konkrétní výhody to danému kuřákovi přinese[8].

2.1.2 Technika maximalizace sebekontroly

Sebekontrola, neboli vůle a její maximalizace, je dalším ze základních stavebních kamenů procesu odvykání. Sebekontrola totiž kuřákovi umožňuje nejen přestat kouřit, ale zároveň u odvykání vydržet. Nejtěžší není přestat kouřit, ale vydržet. To je hlavní důvod, proč se většina odvykajících navrátí zpět ke kouření v prvních osmi dnech ode dne, kdy přestali kouřit a pouze 3-5 % jich vydrží nekouřit déle než 6-12 měsíců[11]. Úkolem terapie by tedy mělo být připravení kuřáka na to, co jej při odvykání čeká a jakými způsoby může daným situacím čelit. Jedná se především o chvíle nepopsatelných chutí na cigaretu, nebo nevolností, které deficit drogy doprovází.

V průběhu terapie je tedy nutné kuřákům poskytovat v hojné míře informace o tom, jaké nejspíš nastanou problémy a jak jim čelit. Terapie by se měla snažit s kuřákem definovat rizikové situace, jejich předcházení i řešení tak, aby když nastanou, na ně mohl odvykající kuřák náležitě reagovat. Může se jednat o pauzy v práci, posezení v restauraci s přáteli, či jiné sociální akce, při kterých kuřákovo okolí kouří a kuřák sám může být pod vlivem alkoholu, který sám o sobě snižuje sebekontrolu a má prokazatelný vliv na to, jestli se člověk ke kouření vrátí[12]. Je také možné doporučení alternativních zdrojů nikotinu, jako jsou nikotinové náplasti a žvýkačky[2].

2.1.3 Technika alternativních činností

Technika alternativních činností je postavená na tom, že ve chvílích největší chuti na cigaretu odvádí kuřákovu pozornost od cigarety na jiné téma. Díky odvedení pozornosti kuřák překoná několik minut největší chuti a může tak pokračovat v odvykání. Velmi populární je technika takzvaných čtyř D, v angličtině delay, deep breathing, drink water, do something else, překladem odložení kouření na později, hluboké dýchání, pití vody a dělání něčeho jiného[6].

Odvedení pozornosti můžeme rozdělit na dva případy. V prvním případě je kuřák konfrontován s cigaretami a zkušenostmi ostatních a ve druhém případě jsou jeho myšlenky od cigaret naopak odváděny[6].

Pro konfrontaci s cigaretami a zkušenostmi ostatních se hodí sdílení informací formou tipů a diskusí. Velmi výhodný je model sociální interakce, ve kterém mohou odvykající kuřáci navzájem sdílet své zkušenosti, komentovat je a vzájemně se podporovat[6].

Pro odvedení pozornosti od cigaret se mohou dokonale hodit jednoduché hry, které člověka velmi rychle vtáhnou do děje. Hry nejenže odvedou kuřákovu pozornost od cigaret, ale zároveň zaměstnají i jeho prsty. Bylo zjištěno, že hraní Tetrisu ve chvílích chuti na cigaretu tyto chutě snižuje. Tento jev je přisuzován tomu, že člověk nedokáže hrát vizuálně zajímavou hru a přitom si představovat sám sebe, jak kouří[13].

2.2 Implementace technik ve stávající aplikaci

Stávající aplikace pro podporu odvykání kouření všechny výše zmíněné techniky v průběhu terapie využívá a zvyšuje tím šanci, že kuřák vydrží dlouhodobě nekouřit.

Technika motivace je používána v hojné míře během terapie, kdy aplikace nechává kuřáka vyplnit důvody, proč chce přestat kouřit. Připomíná výhody nekouření a dává odměny formou odznaků, které jsou v průběhu terapie přidělovány za jednotlivé kroky v definování cílů a motivace, zvládnutý den D, kdy kuřák přestává kouřit, také za první týden a první měsíc nekouření.

Technika maximalizace sebekontroly je taktéž aplikována během celé terapie. Na začátku terapie se snaží s kuřákem definovat jeho rizikové situace a jak na ně bude

reagovat. Tímto se kuřák může v první fázi terapie na tyto situace psychicky připravit a potenciálně jim předcházet. Zároveň aplikace s kuřákem definuje řešení, jak bude reagovat na krizové situace, či s ním vytvoří plán odvykání.

Technika odvedení pozornosti je implementována v takzvaném SOS tlačítku. Skrze tlačítko se dají zobrazit krátké tipy. Jedná se jak o tipy, které odvádí kuřákovu pozornost, tak tipy, které posilují sebekontrolu a utvrzují jej v tom, že přestat je správné. Další možností je rychlá volba zavolat blízkému a popovídat si s ním.

■ 2.2.1 Volba techniky

Techniky motivace a maximalizace sebekontroly jsou velmi dobře zakomponovány v samotné terapii, na které je aplikace postavena. Technika odvedení pozornosti v SOS tlačítku však není dostatečná, neboť textové tipy nemusí vyhovovat každému a navíc jich je v aplikaci prozatím velmi malé množství. Zavolání známému může být efektivní, ale pokud se kuřák například nachází na místě, kde nemůže telefonovat, pak se nedá tato metoda použít.

Z výše zmíněných důvodů bylo zvoleno, že se tzv. SOS tlačítko rozšíří o modul miniher, který bude poskytovat okamžité rozrušení a odvedení pozornosti od chuti na cigaretu. Modul miniher navíc doplní již implementované techniky tak, aby aplikace poskytovala co nejširší škálu dostupných technik a zvyšovala tak šanci, že bude terapie zakončena úspěchem.

■ 2.2.2 Volba miniher

Po zvolení toho, že bude implementován modul miniher, bylo nutné definovat hry, které budou implementovány. Hry byly rozděleny do tří skupin dle jejich charakteristik.

První skupina obsahuje velmi jednoduché a krátké hry, u kterých se nemusí přemýšlet, jsou velmi rychle pochopitelné a není k nim třeba žádná jiná znalost. Zvoleny byly hry 2048 a Bubliny. Cílem hry 2048 je pomocí skládání dlaždic vytvořit dlaždici s hodnotou 2048, cílem hry Bubliny je pomocí spojování bublin stejné barvy vyčistit herní plochu.

Druhou skupinou jsou také jednoduché hry, ty se ale zaměřují na zaměstnání rukou a jejich šikovnost. Zvoleny tedy byly hry, které pracují s gyroskopem, díky kterému může uživatel ovládat hru pohybem ruky. Jako vhodné hry byly zvoleny Labyrint a Kulička. Jejich cílem je pomocí pohybu telefonem dostat kuličku skrze labyrint do cílové díry, respektive vhodit kuličky do díry ve stanoveném pořadí.

Třetí skupinou jsou hry, které se soustřeďují na zaměstnání mysli. Bylo nutné zvolit složitější hry, které vyžadují taktickou hru, ale které však zná většina populace tak, aby byly co nejvíce přístupné. Jako vhodné byly zvoleny hry Piškvorky a Reversi. Vzhledem k tomu, že se jedná o hry pro dva hráče, bylo nutné k těmto hrám implementovat umělé protihráče.

Kapitola 3

Volba platformy

Obsahem této kapitoly je volba operačního systému, na kterém bude rozšíření implementováno. Následuje volba nejnižší verze systému, kterou bude aplikace podporovat a volba programovacího jazyka.

3.1 Volba operačního systému

Stávající aplikace pro podporu odvykání kouření je již implementována pro oba hlavní mobilní operační systémy Android a iOS. Jak ukazuje tabulka 3.1, zastoupení obou hlavních operačních systémů je stabilní v průběhu posledních let. Nejoblíbenější je operační systém Android se stabilním ziskem přes 70 % globálního trhu. Následuje jej systém iOS, který používá více než 20 % populace. Ostatní operační systémy v průběhu času ztrácí na popularitě a jsou v minoritě.

OS / rok	2017[%]	2018[%]	2019[%]	2020[%]
Android	72,6	75,5	75,5	72,8
iOS	19,6	20,5	22,7	26,5
Ostatní	7,8	4,0	1,8	0,7

Tabulka 3.1. Zastoupení mobilních OS na trhu za poslední 4 roky[14].

V rámci rozsahu této práce bylo možné implementovat rozšíření pouze pro jeden operační systém. Z tohoto důvodu byl zvolen operační systém Android, který má větší zastoupení na trhu. Do budoucna se však počítá s tím, že bude modul implementován i pro operační systém iOS.

3.2 Volba nejnižší podporované verze

Po zvolení platformy bylo nutné definovat nejnižší verzi systému, kterou bude aplikace podporovat pro zpětnou kompatibilitu.

Jak ukazuje následující tabulka 3.2, tak více než třetinu zastoupení na trhu měla v září 2019, kdy probíhala volba platformy, nejnovější verze OS Android 9.0, následována sestupně staršími verzemi. Je zajímavé, že verzi OS Android 5.0 Lollipop, která byla představena na konci roku 2014[15] stále používalo přes 9 % zařízení.

OS / zastoupení	Podíl na trhu[%]
9.x Pie	33,4
8.x Oreo	26,0
7.x Nougat	14,6
6.x Marshmallow	13,5
5.x Lollipop	9,2
4.4 KitKat	2,3
Ostatní	1,0

Tabulka 3.2. Zastoupení jednotlivých verzí OS Android na trhu v září 2019[14].

Jako nejnižší verze, kterou bude modul aplikace, a tím i samotná aplikace podporovat, byla zvolena verze 5.0. Podporou verze 5.0 a vyšší dojde k pokrytí 96,7 % zařízení.

Zvolená verze je v souladu s již nastavenou verzí zpětné kompatibility, kterou stávající aplikace má definovanou, takže v rámci implementace nebylo třeba zasahovat do stávající logiky.

Kapitola 4

Android

V této kapitole je popsán operační systém Android. Je zde popsána historie, programovací jazyky a základní elementy, které se používají při vývoji aplikací.

4.1 Historie

Historie operačního systému Android sahá do roku 2003, kdy čtveřice autorů Rich Miner, Nick Sears, Chris White a Andy Rubin začala vytvářet nový systém pro chytřejší zařízení. V roce 2005 pak projekt koupila společnost Google. V roce 2007, nedlouho po uvedení prvního iPhone od společnosti Apple, byla představena první veřejná beta verze 1.0. V září roku 2008 pak byl představen první telefon s tímto systémem, byl jím HTC Dream[16].

Od této doby prošel Android zásadní proměnou a vývojem. Stal se z něj nejpobulárnější operační systém s více než 70 % zastoupením na trhu, viz kapitola 3.1. Nejedná se již pouze o operační systém pro mobilní telefony, ale i tablety, chytré televize, hodinky a automobily.

Od roku 2008 je Android v kontinuálním vývoji. Majoritní verze vychází zpravidla každý rok. V době psaní této práce byla známa poslední verze Android 11, která má vyjít v první polovině roku 2020.

4.2 Programovací jazyky

Aplikace pro OS Android mohou být vyvíjeny buď v jazyce Java, nebo jazyce Kotlin. Dále v této kapitole budou oba jazyky popsány a srovnány. Posledním bodem je výběr jazyka pro tuto práci.

4.2.1 Java

Jazyk Java je multiplatformní, objektově orientovaný jazyk, který je vyvíjen od roku 1995[17]. Původně jej vyvíjela společnost Sun Microsystems, nyní jej vyvíjí společnost Oracle.

Zdrojový kód je kompilován do takzvaného byte-kódu, který je následně spouštěn ve virtuálním stroji Java Virtual Machine – JVM. V OS Android je standardní JVM nahrazena Dalvik Virtual Machine, či od verze 4.4 Kitkat Android Runtime[18].

4.2.2 Kotlin

Programovací jazyk Kotlin je multiplatformní, kompilovaný, staticky typovaný programovací jazyk. Vyvíjen je od roku 2011 společností JetBrains a od roku 2012 je vyvíjen jako OpenSource. Od roku 2017 je oficiálně podporován pro vývoj pro Android[19]. Od roku 2019 je pak oficiálně doporučeným jazykem k vývoji Android aplikací[20].

Jazyk Kotlin byl od začátku vyvíjen primárně pro Android s cílem vylepšit nedostatky jazyka Java. Hlavním přínosem je to, že kód je kratší, tedy i méně náchylný na chyby. Další výhodou je, že odstraňuje riziko chyb typu NullPointerException, neboť zavádí typy, pro které hodnota null není povolena

■ 4.2.3 Porovnání

Výhodou jazyka Kotlin je kompaktní kód, který je méně náchylný na chyby. Naopak velkou výhodou jazyka Java je jeho popularita a komunita, takže pro jazyk Java je mnohem rozsáhlejší dokumentace i výběr knihoven.

Pro rozsáhlejší aplikace se může projevit nevýhoda jazyka Kotlin v tom, že jeho kompilování je časově náročnější a vývojová prostředí jako Android Studio mohou být méně efektivní v napovídání a doplňování kódu.

■ 4.2.4 Výběr jazyka

Pro tuto práci byl jako programovací jazyk zvolen jazyk Java. Důvodů je několik. Hlavním důvodem je to, že stávající část aplikace je napsána v jazyce Java a nebylo by ideální začít další modul programovat v jazyce Kotlin. Druhým důvodem je, že osobně programuji v Java a tudíž mi je tento jazyk bližší a umím s ním pracovat efektivněji.

■ 4.3 Základní elementy

V této kapitole jsou popsány základní stavební prvky aplikací pro OS Android.

■ 4.3.1 Aktivita

Aktivita je základním stavebním prvkem všech aplikací. Zpravidla se jedná o uzavřenou část aplikace, se kterou může uživatel interagovat. Velmi často se jedná o zobrazení přes celou obrazovku, ale aktivita může být i částí jiné obrazovky, či plovoucí okno[21].

Aktivita má celý životní cyklus, nejdůležitější jsou dvě metody, které má implementovány skoro každá aktivita.

■ onCreate()

Tato metoda je volána při vytvoření aktivity. Zpravidla se v ní implementuje konfigurace prvků uživatelského rozhraní.

■ onPause()

Tato metoda je volána, pokud přestává uživatel s aktivitou pracovat. Zpravidla se jedná o zobrazení nové aktivity, například když uživatel přechází do jiného menu, na jinou obrazovku a podobně. V této metodě by měla být uložena všechna data, se kterými uživatel pracoval tak, aby mohla být obnovena při jeho návratu.

■ onResume()

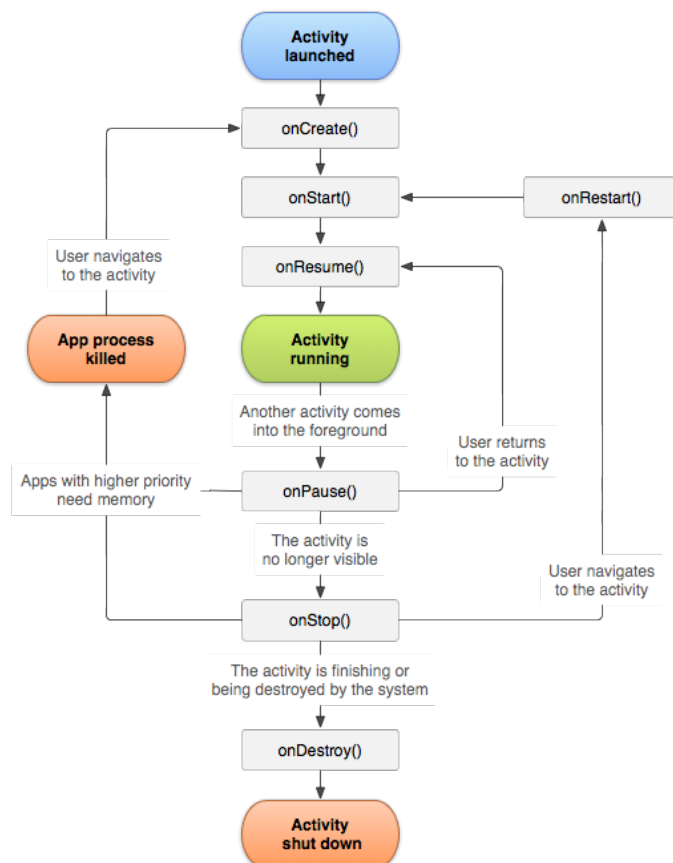
Tato metoda je volána, pokud aktivita přechází do popředí a začíná interagovat s uživatelem. V této metodě by se měla implementovat logika inicializace proměnných, které jsou uvolňovány v metodě onPause(). Například zapnutí kamery a přístup k ní.

Celý životní cyklus aktivity je možné vidět na obrázku 4.1.

■ 4.3.2 Služba

Služba je základní komponenta, která obsluhuje dlouhotrvající činnosti a nemá uživatelské rozhraní. Služba může běžet na pozadí i v případě, že uživatel neinteraguje s aplikací. Jedná se například o synchronizaci dat se serverem, či poslech hudby na pozadí[22].

Každá služba musí být zaregistrována v AndroidManifest.xml4.3.6.



Obrázek 4.1. Životní cyklus aktivity v OS Android

4.3.3 Přijímač vysílání

Přijímač vysílání, neboli `BroadcastReceiver`, je implementací patternu `publish-subscribe`. Jedná se o to, že jednotlivé aplikace se mohou registrovat k odposlouchávání zpráv a změnách stavu systému. Může se jednat o změnu polohy, změnu stavu síťového připojení, či například notifikace o příchozí zprávě SMS.

Výhodou je to, že systém může vyslat zprávu a pouze aplikace, které o ní mají zájem ji zpracují. K zaregistrování přijímání konkrétního typu zpráv se dá použít buď `AndroidManifest.xml` 4.3.6, nebo programově v průběhu běhu aktivity [23].

4.3.4 Poskytovatel obsahu

Poskytovatel obsahu, neboli `ContentProvider`, je entita, která je zodpovědná za sdílení strukturovaných dat napříč aplikacemi. V systému je nativně implementován `ContentProvider` pro seznam kontaktů a kalendář [24].

Výhodou takto sdílených dat je granulární nastavení přístupových práv k jednotlivým datům. Pokud však aplikace nepotřebuje sdílet data s ostatními aplikacemi, pak je lepší použít jiné metody ukládání dat, viz kapitola 4.6.

4.3.5 Fragment

Fragment je modulární složkou jednotlivých aktivit. Slouží k rozmělnění aktivity na menší kusy. Každý fragment musí být součástí nějaké aktivity. Aktivity se většinou skládají z více fragmentů, ale může být i jeden. Fragmenty mají svůj vlastní životní cyklus, který je ale provázaný s životním cyklus aktivity. Díky fragmentům se dá vhodně

rozčlenit uživatelské rozhraní a lépe tak měnit rozhraní pro různě velká zařízení jako jsou telefony a tablety[25].

■ 4.3.6 Android Manifest

Soubor `AndroidManifest.xml` je soubor, který musí mít každá aplikace. Obsahem souboru jsou základní informace nutné ke zveřejnění a instalaci aplikace. Součástí informací je minimální podporovaná verze systému, definice ikon, práv, která aplikace vyžaduje, definice jednotlivých aktivit, přijímačů a služeb[26].

■ 4.4 Uživatelské rozhraní

V této kapitole je popsáno několik základních stavebních prvků uživatelského rozhraní, které jsou dále používány v této práci.

■ 4.4.1 Layout

Layout definuje rozložení jednotlivých prvků v uživatelském rozhraní. Layout se skládá buď ze samotných prvků, nebo skupin prvků, které rekurzivně obsahují další prvky. Typů rozložení je několik druhů[27].

■ LinearLayout

LinearLayout je jednoduchý layout, který řadí prvky postupně buď vertikálně, nebo horizontálně. Layout je velmi flexibilní, neboť vhodným zanořováním horizontálních a vertikálních lineárních layoutů lze docílit prakticky jakékoliv kombinace[28].

■ RelativeLayout

RelativeLayout je layout, který umísťuje prvky relativně vůči prvkům na stejné úrovni, či vůči mateřskému prvku[29].

■ ConstraintLayout

ConstraintLayout je flexibilní layout, který umísťuje prvky pomocí relativního pozicování. Každý prvek proto musí mít určenou relativní pozici vůči ostatním prvkům. Dále má spoustu dalších konfigurovatelných atributů, jako margin, padding, minimální a maximální výšku a šířku a další[30].

■ 4.4.2 View

View je základním prvkem komponent uživatelského rozhraní. Každý prvek z něj dědí. Jedná se o pravoúhelníkovou oblast na obrazovce displeje, která má nějaký vzhled a logiku pro události. Například tlačítko je View, které má naimplementovanou logiku pro chování při události kliknutí[31].

Pokud chce uživatel udělat vlastní implementaci View, pak stačí vytvořit třídu, která od View dědí. Následně může přepisovat různé metody. Ty, které byly použity v této práci, jsou popsány níže[32].

■ `onDraw(Canvas canvas)`

Tato metoda je volána vždy, když má dojít k vykreslení dané komponenty. Komponenta vykreslí svůj požadovaný vzhled do canvasu, viz kapitola 4.4.3. Poté operační systém vykreslí komponentu na obrazovku.

■ `onMeasure(int widthMeasureSpec, int heightMeasureSpec)`

Tato metoda je volána, když je vypočteno, jak velká je komponenta na obrazovce zařízení. Komponenta si při tomto volání může uložit svou velikost a kalkulovat s ní pak při vykreslování svého vzhledu.

- `invalidate()`

Metoda `invalidate` znevalidní aktuální vykreslení `View`, a to se překreslí tak, že zavolá znovu metodu `onDraw(Canvas canvas)`. Tato metoda se používá v případě, že se uvnitř `View` změní stav a je třeba jej překreslit.

■ 4.4.3 Canvas

`Canvas` je struktura, která drží bitmapu se vzhledem komponenty k vykreslení. Je používán v případě vlastní implementace `View`, kdy je `Canvas` před vykreslením naplněn vzhledem a následně vykreslen na obrazovku[33].

■ 4.5 Animace

V případech, kdy aplikace chce oživit obsah, který zobrazuje uživateli, může použít animace.

Android poskytuje několik rozhraní pro animování obsahu, standardně se animují ikony, či přechody mezi obrazovkami[34].

V rámci této práce je však nutné animovat jednotlivé hry, tudíž vlastní implementace `View`. To s sebou přináší specifický přístup animování. Princip animování je v tom, že se `View` ve smyčce překresluje tak dlouho, dokud animace neskončí[32].

```
@Override
protected void onDraw(Canvas canvas) {
    // Vykreslení View do canvasu

    if (isSomeAnimationActive()) {
        // vyvoláme další překreslení
        invalidate();
    }
}
```

V každé iteraci při překreslení se musí všechny objekty animace aktualizovat o čas, který uběhl od posledního vykreslení, čímž je zaručeno, že animace postupně skončí a budou na různě rychlých zařízeních trvat stejně dlouho. Na pomalejších se pouze prvky vykreslí méněkrát, díky čemuž nebude animace tak plynulá.

■ 4.6 Persistence dat

Vzhledem k tomu, že většina aplikací potřebuje persistentně ukládat data, poskytuje OS Android několik možností, jak tato data ukládat[35].

■ 4.6.1 Shared Preferences

`SharedPreferences` je rozhraní pro ukládání a získávání dat. Jedná se o nejjednodušší rozhraní, které OS Android poskytuje. Veškerá data, která jsou uložena, jsou uložena v párech klíč:hodnota. Toto rozhraní je velmi jednoduché na použití a vhodné pro případy, kdy není třeba ukládat velké množství strukturovaných dat[36].

■ 4.6.2 SQLite databáze

Pro případy, kdy je třeba ukládat větší množství strukturovaných dat, je vhodné použít databázi. OS Android poskytuje plnou podporu pro `SQLite` databáze. Zároveň vytvořená databáze je privátní pro danou aplikaci, takže k datům nemají ostatní aplikace přístup[37].

4.7 Vybraná rozhraní

V rámci vývoje aplikací pro OS Android může vývojář získat přístup k různým rozhraním a službám zařízení, na kterých je systém spuštěný. K tomu, aby vývojář nemusel vše programovat sám, může využít již implementovaná řešení. Vybrané služby a rozhraní, která jsou použita v této práci, jsou popsány v následujících kapitolách.

4.7.1 Akcelerometr

Moderní mobilní zařízení jsou vybavena hned několika různými druhy senzorů, jako jsou senzory pohybu, přetížení, polohy a dalších. Jedním z nich je akcelerometr. Akcelerometr je zařízení na měření sil, které na zařízení působí ve všech osách včetně gravitace[38].

V OS Android se s akcelerometrem pracuje velmi jednoduše. V aktivitě, která chce akcelerometr použít, stačí instanciovat senzor a registrovat jej.

```
private SensorManager sensorManager;
private Sensor sensor;
...
sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
sensor = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);

// registrovani senzoru pri startu aktivity
sensorManager.registerListener(
    this, sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
    SensorManager.SENSOR_DELAY_GAME);
```

Následně implementovat metodu, která je volána v případě změny hodnoty v senzoru.

```
@Override
public void onSensorChanged(SensorEvent sensorEvent) {
    if (sensorEvent.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        // IMPLEMENTACE
    }
}
```

4.7.2 Detektor škálování obrazovky

Velmi často v případech vlastní implementace zobrazení, viz kapitola 4.4.2, je třeba rozpoznávat kliknutí na toto zobrazení a další standardní dotyková gesta. Pro usnadnění rozpoznávání těchto dotykových gest, jako je posouvání, či škálování, je možné využívat systémově implementované detektory.

Detektor škálování obrazovky, neboli ScaleGestureDetector, je detektor, který z dat dotyků prstu na obrazovce určuje, jestli se jedná o gesto škálování. Pokud ano, pak poskytuje aktivitě informaci, jak moc uživatel aktuálně škáluje[39].

Pokud vývojář chce detektor použít, pak musí vytvořit třídu ScaleListener, která dědí od třídy SimpleOnScaleGestureListener.

```
class ScaleListener extends SimpleOnScaleGestureListener {
    @Override
    public boolean onScale(ScaleGestureDetector detector) {
        // Zde vývojář pracuje s daty z detektoru, pomocí
        // detector.getScaleFactor()
        return true;
    }
}
```

Následně vytvořit instanci `ScaleGestureDetector` a předat jí implementovanou třídu `ScaleListener`.

```
ScaleGestureDetector scaleGestureDetector =  
new ScaleGestureDetector(context, new ScaleListener());
```

Nyní již stačí detektoru předávat data o dotyku.

```
@Override  
public boolean onTouch(View view, MotionEvent motionEvent) {  
    scaleGestureDetector.onTouchEvent(motionEvent);  
    return true;  
}
```

Kapitola 5

Návrh a implementace

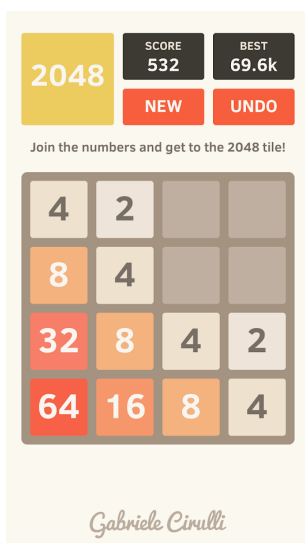
Obsahem této kapitoly je popis návrhu a implementace jednotlivých miniher. Pro každou hru bylo nutné prostudovat stávající řešení, na základě toho definovat, jakou funkcionalitu by měla daná hra poskytovat. Následoval grafický návrh a finální implementace. Tento proces je pro každou hru popsán v následujících kapitolách.

5.1 Hra 2048

Hra 2048[40] je světoznámá hra, která se hraje na herní desce velikosti 4x4. Hra začíná tím, že na ploše je jeden herní kámen s hodnotou 2. Hráč může kameny přesouvat do všech světových stran. Každým tahem jeden nový kámen s hodnotou 2 přibude. Cílem hráče je pokusit se co nejčastěji spojovat kameny se stejnou hodnotou, díky čemuž se kameny sloučí do jednoho a vznikne tím tak jeden nový kámen s dvojnásobnou hodnotou. Takto hráč tahá a spojuje, dokud nezaplní herní desku tak, že již nemůže kameny pohnout. V tuto chvíli prohrává. Pokud se mu však podaří vytvořit kámen s hodnotou 2048, vyhrává. Hráč i po výhře může pokračovat dále ve hře a pokoušet se tak pokořit své nejvyšší skóre a získat kostku co nejvyšší hodnoty. Teoreticky nejvyšší možná hodnota kamene, který hráč může vytvořit, je kámen s hodnotou 65 536.

5.1.1 Existující řešení

Ke zhodnocení byly zvoleny dvě alternativy hry 2048 z dostupných verzí na Google Play v době psaní této práce. Obě mají totožný název 2048[41][42]. První zmíněná je oficiální mobilní aplikace pro oficiální webovou aplikaci[40], která s hrou 2048 přišla jako první. Druhá aplikace je její modifikace, která poskytuje další funkcionalitu. Vzhled verzí je vidět na následujících obrázcích 5.1 a 5.2.



Obrázek 5.1. Ukázka oficiální aplikace hry 2048



Obrázek 5.2. Ukázka modifikace hry 2048

Jak je možné vidět, uživatelské rozhraní je jednoduché a srozumitelné. Hlavní ovládací prvky jsou viditelné a zobrazují vše, co uživatel potřebuje.

Ohledně funkcionalit, modifikace hry 2048 poskytuje možnost změny velikosti herní desky, jiný barevný motiv či citlivost animací. Ani jedna aplikace však neměla podporu změny rychlosti animace, což může být problém v případě, že chce člověk animace rychlejší a zdržují jej.

5.1.2 Návrh funkcionality a uživatelského rozhraní

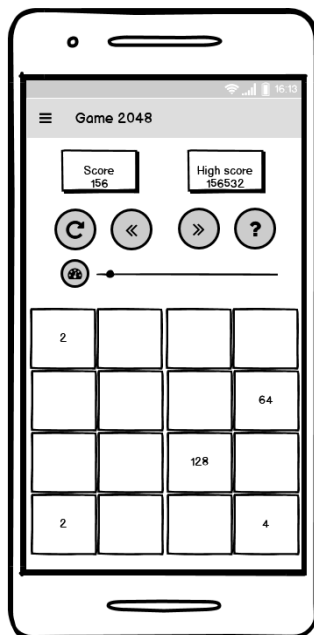
Při návrhu funkcionalit a uživatelského rozhraní bylo čerpáno z kombinace prvků z již existujících řešení. Bylo definováno, že by uživatelské rozhraní mělo podporovat následující funkcionality:

- Uložení stavu hry
- Vrátit se o tah zpět
- Posunout se o tah vpřed
- Reset hry
- Nápopvěda
- Počítání skóre
- Animovaný pohyb dlaždic
- Posuvník pro volbu rychlosti animace

Ovládací prvky by měly být jednoduché a intuitivní. Uživatel by pokud možno neměl vůbec používat nápovědu. Pohyb dlaždic po herní desce by měl být řešen pomocí gest tak, aby byl plně využit potenciál dotykové obrazovky mobilních zařízení.

5.1.3 Grafický návrh

Před implementací byl vytvořen prototyp grafického návrhu pomocí nástroje Balsamiq Wireframes. Prototyp obsahuje všechny prvky uživatelského rozhraní, které byly definovány v předchozí kapitole. Ukázka prototypu je zobrazena na následujícím obrázku 5.3



Obrázek 5.3. Ukázka vzhledu prototypu hry 2048

5.1.4 Implementace

Doménový model

V rámci řešení byl navrhnout doménový model, který je zobrazen na následujícím obrázku 5.4. Model obsahuje následující entity:

- Game 2048

Entita Game2048 je základní entitou, která udržuje informace o aktuální, předchozím a následujícím herním stavu a skóre s těmito stavy spojenými.

- Grid

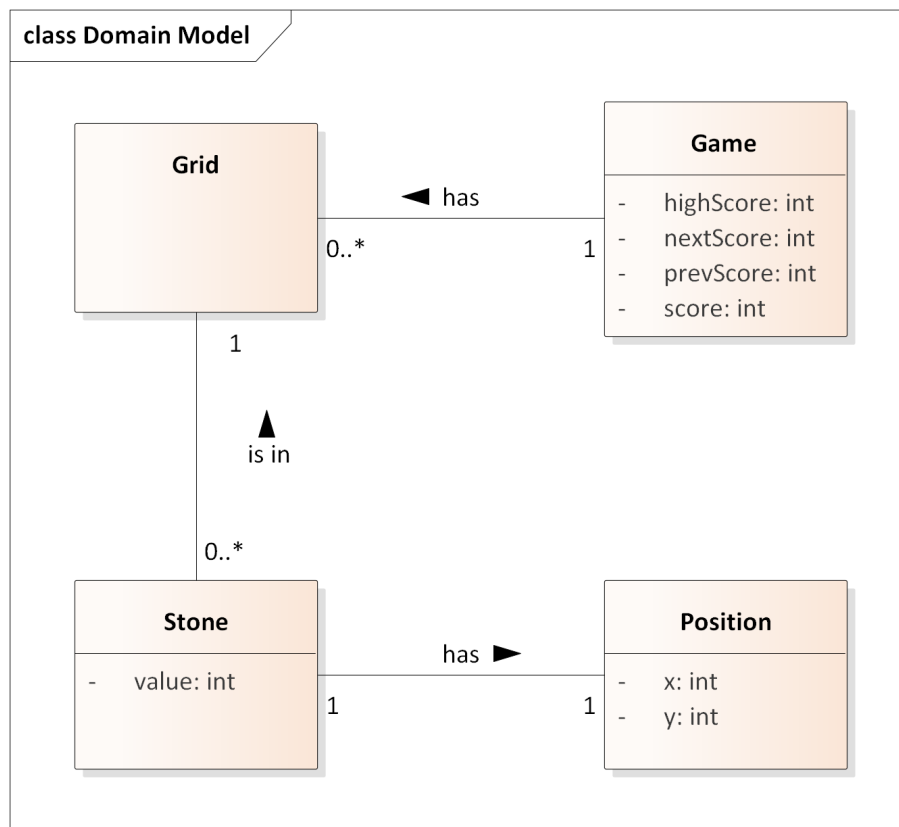
Entita Grid slouží k strukturovanému uchovávání kamenů na herní desce.

- Stone

Entita Stone je entita reprezentující jednotlivé kameny na herní desce. Entita udržuje hodnotu a pozici daného kamenu.

- Position

Entita Position je pomocná entita, která reprezentuje pozici ve dvourozměrné herní desce.



Obrázek 5.4. Doménový model hry 2048

Herní logika

Herní logika hry 2048 je v zásadě jednoduchá. Po každém tahu stačí pouze přesunout všechny kameny v daném směru a v případě možnosti sloučení je sloučit. Při slučování se inkrementuje herní skóre. Po každém tahu se uloží herní stav pro případ tahu zpět. Dále je implementováno persistentní ukládání herního stavu v případě opuštění aplikace, aby při návratu do aplikace došlo k obnovení aktuálního herního

stavu. Toto persistentní ukládání je řešeno serializací do textového řetězce a ukládání do objektu SharedPreferences, viz kapitola 4.6.1.

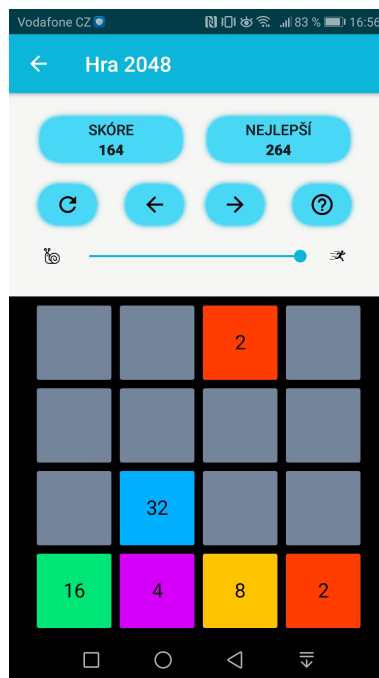
■ Vzhled a animace

Ovládací prvky v horní části obrazovky jsou řešeny standardními prvky umístěnými v layoutu aktivity, viz kapitola 4.4.1.

Samotná herní deska je vykreslována dynamicky do Canvasu, viz kapitola 4.4.3. Díky dynamickému vykreslování je zajištěno, že vždy budou herní pole čtvercová a zároveň je možné dynamicky vykreslovat animace. Logika vykreslování animací je více popsána v kapitole 4.5.

Pro hru 2048 byly definovány tři typy animací. Jedná se o animaci posunu, což je animace trajektorie dlaždice po přísmce. Následně animace sloučení, což je animace posunu následována nahrazením dlaždic novou dlaždicí s dvojnásobnou hodnotou. Poslední typ animace je animace objevení nové dlaždice, což je postupné zvětšení dlaždice z jejího středu do plné velikosti. Délka animací je uživatelsky konfigurovatelná pomocí posuvníku.

Finální vzhled implementované hry 2048 je možné vidět na následujícím obrázku 5.5.



Obrázek 5.5. Finální vzhled hry 2048

5.2 Hra Reversi

Hra Reversi[43], nazývaná též Othello, je desková hra, která se hraje na herní desce 8x8. Hraje se s kameny, které mají jednu stranu bílou a druhou stranu černou. Hra začíná umístěním čtyř kamenů, dvou bílých a dvou černých, do středu herní plochy. Hru začíná bílý hráč.

Jednotlivé tahy hraje hráč tak, že položí kámen své barvy vždy takovým způsobem, aby mezi nově položeným kamenem a nějakým jeho původním byl alespoň jeden kámen protihráče. Takto může obklíčit jeden, či více protihráčových kamenů v jednom, či více

směrech. Všechny takto obklíčené kameny jsou následně překlopeny na barvu hráče, který je obklíčil. Hráči se takto střídají. Pokud hráč nemá žádný možný tah, je přeskočen a znovu hraje druhý hráč.

Hra takto pokračuje, dokud ani jeden hráč nemá žádný možný tah, nebo dokud není zaplněna celá herní plocha.

Vítězem se stává hráč, který má na konci hry na herní desce více kamenů své barvy.

■ 5.2.1 Návrh funkcionality a uživatelského rozhraní

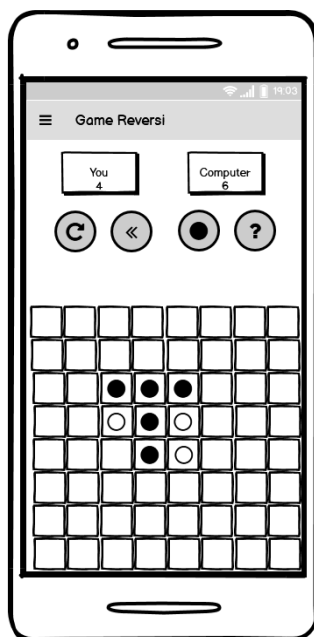
Při návrhu funkcionality bylo vycházeno primárně z návrhu pro hru 2048, viz kapitola 5.1.2, aby byla funkcionality a rozhraní sjednocené napříč hrami. Protože se jedná o hru pro dva hráče, jednou z funkcionalit je umělý protihráč, dále by uživatelské rozhraní mělo podporovat následující funkcionality:

- Reset hry
- Vrátit se o tah zpět
- Změnit barvu, za kterou uživatel hraje
- Náповěda
- Počítání skóre
- Umělý protihráč

Ovládací prvky by měly být jednoduché a intuitivní. Herní plocha stačí statická, neboť matice 8x8 je i na menších zařízeních stále dostatečně velká tak, aby uživatel mohl bezchybně klikat na pozice, kam chce hrát svůj tah.

■ 5.2.2 Grafický návrh

Před implementací byl stejně jako pro ostatní hry vytvořen prototyp grafického návrhu pomocí nástroje Balsamiq Wireframes. Prototyp obsahuje všechny prvky uživatelského rozhraní, které byly definovány v předchozí kapitole. Zároveň je snaha o dodržení designu napříč jednotlivými hrami. Ukázka prototypu je zobrazena na následujícím obrázku 5.6



Obrázek 5.6. Ukázka vzhledu prototypu hry Reversi

■ 5.2.3 Implementace

■ Doménový model

Doménový model a jeho entity jsou stejné jako pro hru 2048, viz kapitola 5.1.4.

■ Herní logika

Pro hru Reversi bylo nutné implementovat logiku, která kontroluje pravidla a možné tahy, které hráč může zahrát. Kromě toho bylo třeba implementovat umělého protihráče.

Umělý protihráč je vytvořen jako bezstavový chamtivý hráč, který se řídí hodnotou součtu pozic, které mu daný tah získá. Jednotlivá pole na herní desce totiž nejsou stejně hodnotná. Nejhodnotnější jsou pole v rozích. Roh je výhodný v tom, že jej protihráč nemůže získat zpět. Pokud hráč vlastní více rohů, pak má možnost získat okraje herní desky, které již nemůže ztratit a skrze něž může snáze kontrolovat herní desku. Nejméně hodnotná jsou pole v okolí rohů, neboť pouze skrze ně může protihráč obsadit roh.

Matice je vytvořena v souladu s touto strategií a je následující[44]:

```
{ 99, -8, 8, 6, 6, 8, -8, 99},
{ -8,-24, -4, -3, -3, -4,-24, -8},
{ 8, 1, 7, 4, 4, 7, 1, 8},
{ 6, 1, 1, 1, 1, 1, 1, 6},
{ 6, 1, 1, 1, 1, 1, 1, 6},
{ 8, 1, 7, 4, 4, 7, 1, 8},
{ -8,-24, -4, -3, -3, -4,-24, -8},
{ 99, -8, 8, 6, 6, 8, -8, 99}
```

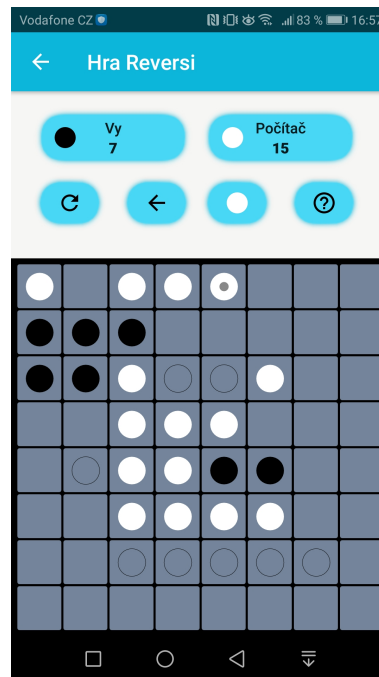
Stejně jako pro ostatní hry je implementováno persistentní ukládání herního stavu v případě opouštění aplikace pomocí serializace do textového řetězce a uložení do objektu SharedPreferences, viz kapitola 4.6.1.

■ Vzhled a animace

Ovládací prvky v horní části obrazovky jsou řešeny standardními prvky umístěnými v layoutu aktivity, viz kapitola 4.4.1.

Herní deska je dynamicky vykreslována do Canvasu[33], stejně jako pro ostatní hry je díky tomu možné dynamicky vykreslovat animace a ostatní herní prvky. Pro hru Reversi se animuje pouze překlápění kamenů. Herní deska je doplněna o tenké kroužky, které značí dostupné tahy, stejně jako označení posledního tahu.

Finální vzhled implementované hry Reversi je možné vidět na následujícím obrázku 5.7.



Obrázek 5.7. Finální vzhled hry Reversi

5.3 Hra Piškvorky

Hra piškvorky je desková hra, která se hraje na herní desce se čtvercovými poli. Deska je velká většinou 15x15 polí. Hrají ji dva hráči, jeden má křížky, druhý kolečka, případně černé a světlé kameny. Hráči se po tazích střídají, začíná hráč s křížky. Výhercem je hráč, který jako první vytvoří nepřerušenu řadu pěti svých symbolů.

5.3.1 Návrh funkcionality a uživatelského rozhraní

Při návrhu funkcionality bylo vycházeno primárně z návrhu pro hru Reversi, viz kapitola 5.2.1, aby byla funkcionality a rozhraní sjednocené napříč hrami. Jediným rozdílem je to, že zobrazení 15x15 polí na herní desce může být na menších displejích drobné. Z tohoto důvodu je vyžadováno zvětšování a posouvání herní desky tak, aby uživatel mohl pohodlně klikat na pozice, kam chce zahrát svůj tah.

5.3.2 Grafický návrh

Vzhledem k tomu, že se hra piškvorky hraje stejně jako hra Reversi na desce se čtvercovými poli s jediným rozdílem a to počtu polí, kterých je v případě hry piškvorky 15x15, byl v rámci sjednocení přepoužitý design hry Reversi, viz kapitola 5.2.2.

5.3.3 Implementace

Doménový model

Doménový model a jeho entity jsou stejné jako pro hru 2048, viz kapitola 5.1.4.

Herní logika

Pro hru Reversi bylo nutné implementovat logiku, která kontroluje pravidla a možné tahy, které hráč může zahrát. V podstatě se jedná o kontrolu, jestli již není pole zaplněné a jestli nenastal konec hry. Kromě toho bylo třeba implementovat umělého

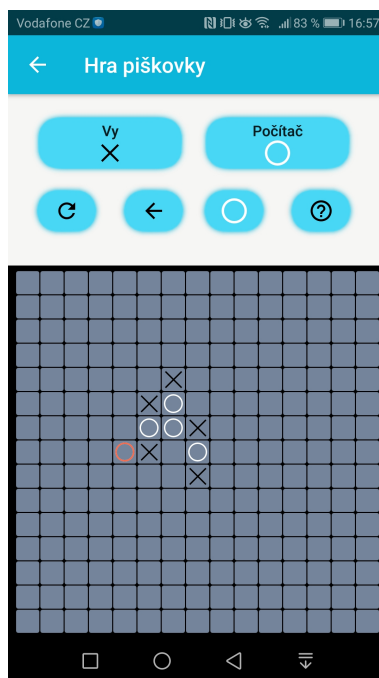
protihráče, který je sofistikovanější, než umělý protihráč ve hře Reversi. Jeho vlastnosti a implementace jsou podrobně popsány v kapitole 6. Stejně jako pro ostatní hry je implementováno persistentní ukládání herního stavu v případě opouštění aplikace pomocí serializace do textového řetězce a uložení do objektu SharedPreferences, viz kapitola 4.6.1.

■ Vzhled

Ovládací prvky v horní části obrazovky jsou řešeny standardními prvky umístěnými v layoutu aktivity, viz kapitola 4.4.1.

Herní deska je dynamicky vykreslována do Canvasu[33], stejně jako pro ostatní hry je díky tomu možné dynamicky vykreslovat herní prvky. Součástí je zobrazení posledního tahu tak, aby se hráč mohl orientovat. Ke zvětšování a posouvání herní desky je použit ScaleGestureDetector, viz kapitola 4.7.2. Zvětšit herní plochu lze maximálně 5x, větší zvětšení nemá smysl, neboť by pak jedno pole bylo velké jako celý displej.

Finální vzhled implementované hry Piškvorky je možné vidět na následujícím obrázku 5.8.



Obrázek 5.8. Finální vzhled hry Piškvorky

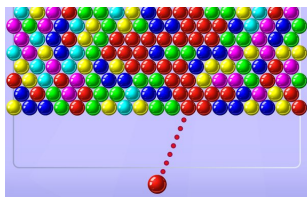
5.4 Hra Bublíny

Hra Bublíny je jednoduchá antistresová dovednostní hra. Smyslem hry je vystřelovat bublíny různých barev do skupiny bublin, která se v průběhu hry zvětšuje. Vždy, když se hráči podaří vystřelit bublinu tak, že po zásahu ostatních bublin vznikne skupina 3 a více bublin pohromadě, pak celá tato skupina zmizí. Pokud je hráč šikovný, podaří se mu odstranit všechny existující bublíny a tím vyhrává.

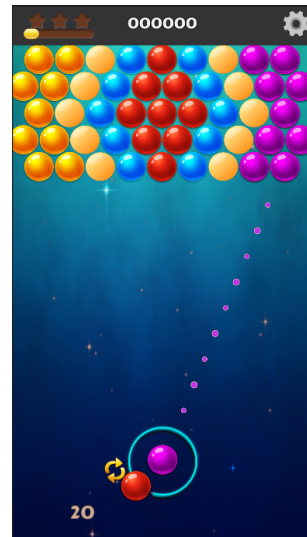
5.4.1 Existující řešení

Variací hry Bublíny je nespočetně. Nejspíše první verzí je hra Puzzle Bobble[45] od společnosti Taito z roku 1994. Ke srovnání byly vybrány hry Bubble Shooter[46]

a Bubble Shooter Space[47]. Jedná se o hry dostupné zadarmo na Google Play. Hry podobného typu jsou velmi oblíbené, dokazuje to i to, že hra Bubble Shooter má přes 100 milionů stažení. Jak je možné vidět z obrázků 5.9 a 5.10, hry mají jednoduché a intuitivní rozhraní.



Obrázek 5.9. Ukázka hry Bubble Shooter



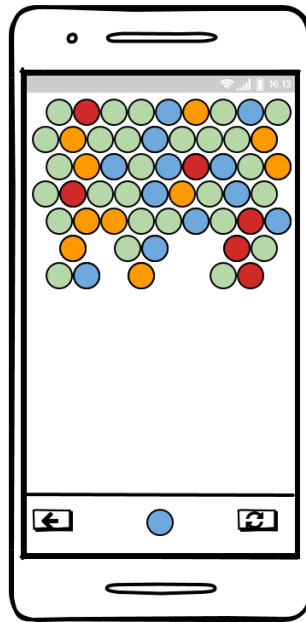
Obrázek 5.10. Ukázka hry Bubble Shooter Space

■ 5.4.2 Návrh funkcionality a uživatelského rozhraní

Při návrhu funkcionalit a uživatelského rozhraní bylo vycházeno z předpokladu, že ovládání hry by mělo být řešeno pouze dotykem. Zároveň by mělo jít zvolit obtížnost hry, případně hru restartovat.

■ 5.4.3 Grafický návrh

Před implementací byl stejně jako pro ostatní hry vytvořen prototyp grafického návrhu pomocí nástroje Balsamiq Wireframes. Cílem byla maximalizace herní plochy, z tohoto důvodu rozhraní obsahuje pouze dvě tlačítka pro ovládání hry. Ukázka prototypu je zobrazena na následujícím obrázku 5.11.



Obrázek 5.11. Ukázka vzhledu prototypu hry Bublíny

■ 5.4.4 Implementace

■ Doménový model

V rámci řešení byl navržen doménový model, který je zobrazen na následujícím obrázku 5.12. Model obsahuje následující entity:

- GameBubbles

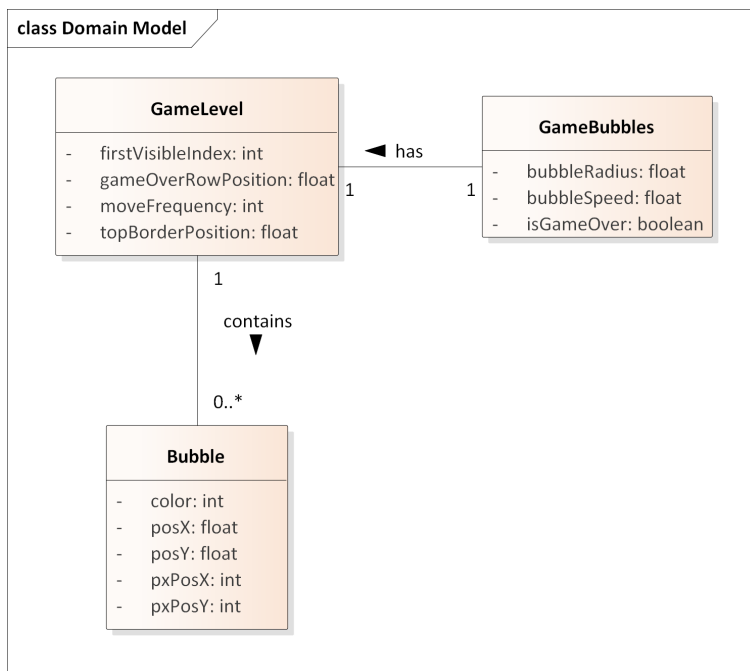
Entita GameBubbles je základní entitou, která udržuje informace o aktuální herní úrovni, rychlosti a velikosti bublin.

- GameLevel

Entita GameLevel je základní entitou pro jednotlivé herní úrovně. Entita má na starosti posouvání bublin, jejich správu, či vytváření nových bublin k vystřelení. Entita je parametricky nastavitelná tak, aby šlo konfigurovat jednotlivé herní úrovně.

- Bubble

Entita Bubble je entita reprezentující bublinu. Entita drží informace o aktuálním indexu entity, její poloze na ploše displeje a barvě.



Obrázek 5.12. Doménový model hry Bubliny

■ Herní logika

Ve hře Bubliny bylo nutné implementovat logiku střelby nové koule a nárazu do ostatních. Střelba je implementována jako pohyb po trajektorii. V případě, kdy koule narazí do ostatních, nebo do horního okraje displeje, pak se zastaví. Pokud při kontaktu vytvoří skupinu bublin stejné barvy, které jsou alespoň 3, pak se tyto bubliny odstraní. Po jejich odstranění dojde ke kontrole, jestli některé bubliny nevisí bez spojení s vrchním okrajem displeje. Pokud tomu tak je, pak jsou odstraněny také. Zároveň jednou za několik tahů dojde k posunu všech koulí směrem dolů, čímž se přiblíží k linii prohry. Čím častěji se tak děje, tím je hra těžší.

Hra má definovány tři herní úrovně obtížnosti. Jedná se o lehkou, střední a těžkou obtížnost. Obtížnosti se liší frekvencí, jak často dochází k posunu bublin a množstvím herních bublin, které je postupně třeba odstranit. Jednotlivé úrovně jsou poměrně dlouhé, tím je ale docíleno, že u nich uživatel zůstane potřebných několik minut.

Pro pohyb koule po trajektorii v případě vystřelení bylo třeba implementovat odraz koulí v případě dotyku stěn. Odraz je řešen zákonem odrazu[48], kdy úhel dopadu se rovná úhlu odrazu.

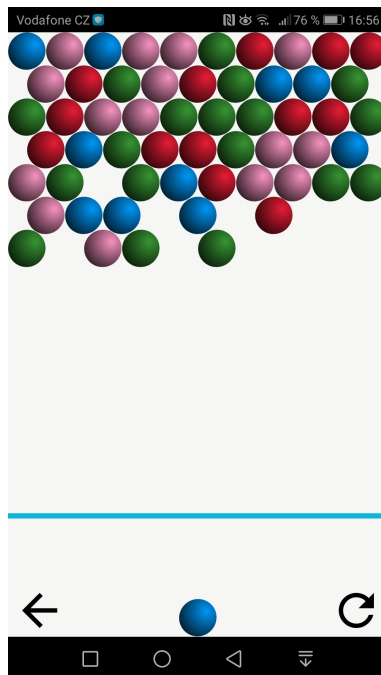
```

if (narazDoSteny) {
    rychlostX = -rychlostX
}
  
```

■ Vzhled a animace

Stejně jako u ostatních her je herní deska dynamicky vykreslována do Canvasu[33]. Pro tuto hru jsou implementovány 3 typy animací. Prvním typem je animace pohybu po trajektorii při vystřelení bubliny. Druhým typem je zpřesnění polohy po nárazu do ostatních koulí. Tato animace se používá, když koule narazí do ostatních koulí, tak se pomalým pohybem posune na svojí přesnou pozici v mřížce. Posledním typem je animace splasknutí, která je využívána při odstranění bubliny.

Finální vzhled implementované hry Bubliny je možné vidět na následujícím obrázku 5.13.



Obrázek 5.13. Finální vzhled hry Bublíny

5.5 Hry Kulička a Labyrint

Hra Kulička je oddechová dovednostní hra. Hra začíná tím, že na herní desce je několik kuliček a několik děr. Cílem hry je pomocí naklánění telefonu dostat všechny kuličky do děr. Červená kulička, která je jen jedna, musí spadnout do díry až jako poslední.

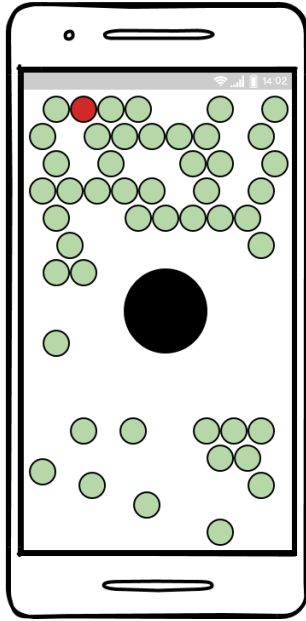
Hra Labyrint je variací na hru s kuličkou. Na herní desce je pouze jedna kulička a jedna díra. Následně je deska plná stěn v podobě labyrintu. Cílem je co nejrychleji dostat kuličku do díry.

5.5.1 Návrh funkcionality

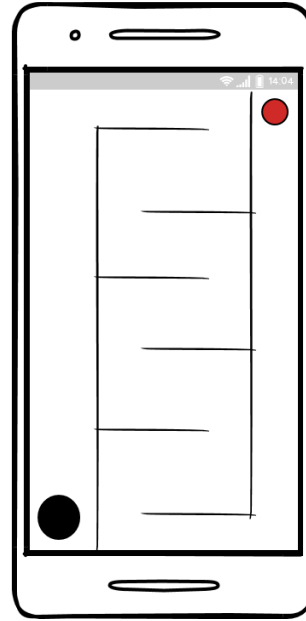
Při návrhu funkcionality bylo definováno, že by obě variace hry měly být rozděleny na jednotlivé herní úrovně dle obtížnosti. Zároveň bylo definováno, že musí být přítomna podpora kalibrace senzoru gyroskopu tak, aby mohlo být ovládání hry přesné pro každé zařízení.

5.5.2 Grafický návrh

Před implementací byl stejně jako pro ostatní hry vytvořen prototyp grafického návrhu pomocí nástroje Balsamiq Wireframes. Cílem byla maximalizace herní plochy, z tohoto důvodu rozhraní obsahuje pouze dvě tlačítka pro ovládání hry. Žádoucí je, aby se tato tlačítka po několika vteřinách schovala a tím pádem byla herní plocha úplně čistá. Zobrazit tlačítka by mělo jít po kliknutí na displej. Ukázka prototypu her Kuličky a Labyrint je vidět na obrázcích 5.14 a 5.15.



Obrázek 5.14. Ukázka prototypu hry Kulička



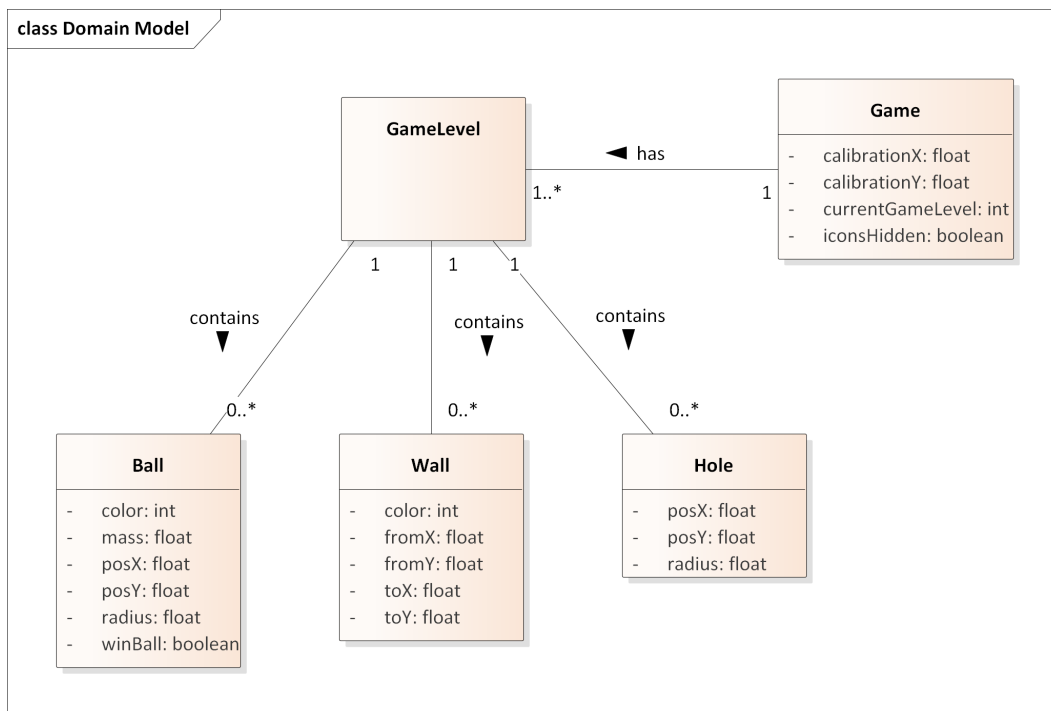
Obrázek 5.15. Ukázka prototypu hry Labyrint

■ 5.5.3 Implementace

■ Doménový model

V rámci řešení byl navržen doménový model, který je zobrazen na následujícím obrázku 5.16. Model obsahuje následující entity:

- Game
 - Entita Game je základní entitou. Entita drží informace o aktuální herní úrovni, kalibraci senzorů a stavu viditelnosti ikon.
- GameLevel
 - Entita GameLevel je entitou, která reprezentuje jednu herní úroveň. Udržuje informace o herních prvcích.
- Ball
 - Entita Ball je entita reprezentující kuličku. Má informace o aktuální pozici, barvě, hmotnosti, velikosti a také jestli se jedná o výherní kuličku.
- Wall
 - Entita Wall je entita, která reprezentuje zeď jako pevnou překážku pro jednotlivé kuličky. Zeď má pozici a barvu.
- Hole
 - Entita Hole je entita reprezentující díru, do které můžou kuličky padat. Má pozici a velikost.



Obrázek 5.16. Doménový model her Kulička a Labyrint

■ Herní logika

Hry Kulička a Labyrint využívají pro pohyb kuliček po herní desce vstupy z akcelerometru. Práce s akcelerometrem je blíže popsána v 4.7.1. V závislosti na vstupu dat z gyroskopu jsou aktualizovány rychlosti a zrychlení jednotlivých koulí. V závislosti na zrychlení je následně aktualizována rychlost a pozice jednotlivých koulí.

```

// currFrameTime je čas,
// který uběhl od poslední aktualizace rychlosti a polohy
xSpeed += xAccel * currFrameTime;
ySpeed += yAccel * currFrameTime;

posX -= xSpeed * currFrameTime;
posY -= ySpeed * currFrameTime;
  
```

Náraz koule do jakékoliv stěny, ať se jedná o hranu displeje, či stěnu ve hře Labyrint, je řešen, stejně jako u hry Bublíny, zákonem odrazu[48], kdy úhel dopadu se rovná úhlu odrazu.

Logika nárazů jednotlivých koulí do sebe je řešena pomocí sady rovnic pro jedno-rozměrné pružné kolize[49].

■ Kalibrace

Jelikož gyroskop na každém zařízení může být jinak zkalibrován, či nemusí být nastaven tak, aby pro vodorovnou pozici předával hodnoty 0, musela být do aplikace přidána podpora kalibrace. Kalibrace spočívá v tom, že si aplikace uloží hodnoty z akcelerometru ve vodorovné pozici, a ty pak při výpočtech započítává. Obrazovka kalibrace, která je přidána do menu nastavení je zobrazena na následujícím obrázku 5.17. Hodnoty jsou po stisku tlačítka Kalibrovat senzory uloženy do objektu Shared-Preferences, viz kapitola 4.6.1.



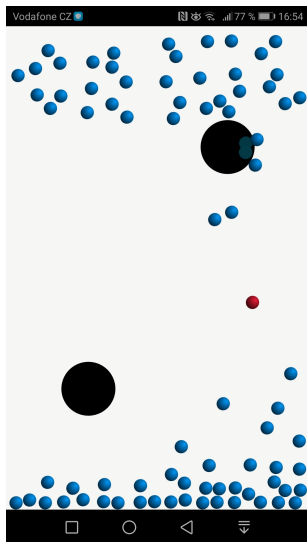
Obrázek 5.17. Ukázka obrazovky pro kalibraci Gyroskopu ve hrách Kulička a Labyrint

Ve výpočtech se pak s uloženými hodnotami pracuje takto:

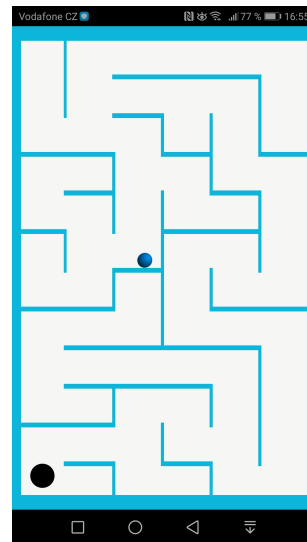
```
xAccel = sensorValuesX - calibrationX;
yAccel = sensorValuesY - calibrationY;
```

■ Vzhled a animace

Finální vzhled implementovaných her Kulička a Labyrint je možné vidět na následujících obrázcích 5.18 a 5.19.



Obrázek 5.18. Finální vzhled hry Kulička



Obrázek 5.19. Finální vzhled hry Labyrint

Kapitola 6

Umělý protivráč pro hru Piškvorky

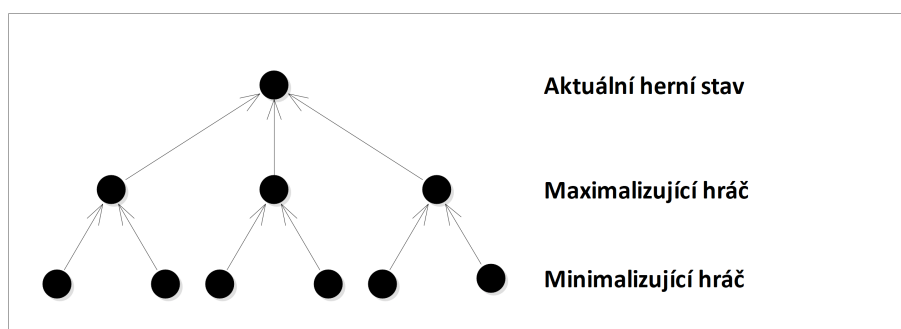
V této kapitole je popsána implementace umělého protivráče pro hru Piškvorky, který je založen na algoritmu minimax s alpha-beta ořezáváním.

6.1 Popis algoritmu

Minimax je algoritmus založený na iterativním prohledávání stavového prostoru do hloubky. Principem je minimalizace maximálních možných ztrát. Možné zisky a ztráty jsou definovány ohodnocovací funkcí, která jednotlivým stavům herní plochy přiřazuje číselnou hodnotu[50]. Více o ohodnocovací funkci použité v této práci je popsáno v kapitole 6.1.2.

Samotný běh algoritmu je založen na předpokladu, že oba hráči hrají optimálně. Logika je rozdělena mezi dva hráče. Hráče, jež maximalizuje a hráče, jež minimalizuje. Prvním hráčem je hráč, pro kterého hledáme nejlepší tah a tento hráč se snaží zahrát co nejlepší tah, tedy maximalizuje. Druhým hráčem je protivráč, který se naopak snaží zahrát tak, aby hráč, který je na tahu, získal co nejméně výhodnou pozici, tedy minimalizuje.

Algoritmus takto iterativně prohledává po jednotlivých tazích a pokud pozice dosáhne konečného stavu, vrátí hodnotu herní desky. Konečným stavem může být buď stav, ve kterém dochází k ukončení hry, tedy výhra, prohra a remíza, nebo algoritmus dospěl do maximální definované hloubky. V opačném případě se algoritmus pokusí zahrát na všechny dostupné pozice a prohledat je až do koncového stavu. V závislosti na výsledných hodnotách jednotlivých pozic vrátí minimální, nebo maximální hodnotu ze všech možných tahů.



Obrázek 6.1. Ukázka jednotlivých vrstev prohledávání v algoritmu Minimax

```
private int minimax(Board board, Color onMove,
boolean maximizingPlayer, int depth) {

    int boardValue = boardEvaluator.evaluateBoard(board, onMove);

    if (depth <= 0) {
        return boardValue;
    }
}
```

```

}

List<Position> emptyPositions =
    getEmptyPositionsOnBoard(board);

int bestValue;
if (maximizingPlayer) {
    bestValue = Integer.MIN_VALUE;

    for (Position position : emptyPositions) {

        // položim nový kámen na herní desku
        Board newBoard = placeNewStoneOnBoard(position, onMove, board);

        currValue = minimax(newBoard, getOpponentsColor(onMove), false, depth-1);

        // snažím se najít tah, pro který bude hodnota desky největší
        if (currValue > bestValue) {
            bestValue = currValue;
        }
    }
} else {
    // minimizing player

    bestValue = Integer.MAX_VALUE;

    for (Position position : emptyPositions) {

        // položim nový kámen na herní desku
        Board newBoard = placeNewStoneOnBoard(position, onMove, board);

        currValue = minimax(newBoard, getOpponentsColor(onMove), false, depth-1);

        // snažím se najít tah, pro který bude hodnota desky nejmenší
        if (currValue < bestValue) {
            bestValue = currValue;
        }
    }
}

return bestValue;
}

```

6.1.1 Heuristiky

Pro každou hru, která má větší větvící faktor, tedy větší množství možných tahů v každém stavu hry, je algoritmus minimax nepoužitelný. Důvodem je to, že časová náročnost algoritmu je $O(b^d)$, kde b je průměrný větvící faktor a d je hloubka, do které algoritmus prohledává. Z tohoto důvodu bylo přistoupeno k využití některých heuristik.

První použitou heuristikou je takzvané alpha-beta ořezávání. Alpha-beta ořezávání je optimalizace běhu algoritmu. Ta funguje tak, že pokud je známo, že daný tah nemůže být lepší, než nějaký, který již byl nalezen, tak se daný podstrom přestává prohledávat. Tímto je velmi zefektivněn běh algoritmu, přesto je nalezeno nejlepší řešení.

Další přidanou heuristikou je logika, která omezuje prohledávání pouze na plochu definovanou jako obdélník, který ohraničuje všechny kameny, jež jsou aktuálně položeny a rozšiřuje ji o 2 řady z každé strany. Velmi se tím snižuje počet pozic, které jsou prohledávány, zvláště na začátku hry, kdy se místo plochy 15x15 prohledává například pouze plocha 5x5. Tato heuristika byla přidána z toho důvodu, že prakticky nikdy nedochází k tahům, které by byly o více jak dvě řady vzdáleny od nějakého již položeného kamene.

Pozice	Hodnota
Dvojice	5
Trojice	3000
Čtveřice	6000
Pětice	1000000
Trojice z jedné strany ohraničená	30
Čtveřice z jedné strany ohraničená	4000
Čtveřice, pokud má v sobě mezeru	4000

Tabulka 6.1. Definované pozice a jejich hodnoty pro ohodnocovací funkci ve hře Piškvorky

6.1.2 Ohodnocovací funkce

Nezbytností ke správné funkci algoritmu je ohodnocovací funkce. Ohodnocovací funkce je zvolena tak, aby ohodnocovala definované herní pozice[51]. Pro každou pozici je definována hodnota. Pozice a jejich hodnoty jsou v následující tabulce 6.1.

6.2 Implementace

Celá implementace tohoto algoritmu je řešena jako samostatná knihovna naprogramovaná v jazyce Java verze 1.6. Verze 1.6 je zvolena proto, protože se jedná o nativní verzi, ve které je naprogramován OS Android.

V rámci implementace byl naimplementován algoritmus popsáný výše, společně s jeho heuristikami.

6.2.1 Paralelizace

Vzhledem k tomu, že výpočetní náročnost algoritmu je i přes heuristiky vysoká, byla naimplementována paralelizace. Vzhledem k tomu, že výpočet jednotlivých větví je datově nezávislý, byla paralelizace jednodušší. Jediná data, ke kterým přistupují všechna vlákna současně jsou hodnoty alpha a beta. Přístup ke čtení je bez problémů, pro zápis je použit semafor.

Samotná paralelizace je navržena tak, že na začátku výpočtu jsou možné tahy pro počáteční herní stav rovnoměrně rozděleny mezi vlákna, která pak pracují na prohledávání stavového prostoru nezávisle. Pseudokód paralelizace je níže.

```
int[] minimaxMultiThread(Board board, Color onMove, boolean maxPlayer, int depth) {

    // inicializujeme globalni nejlepší hodnotu a pozice pro jednotlivá vlákna
    globalBestValue = Integer.MIN_VALUE;
    List<int[]> bestPositions = new ArrayList();

    // zjistíme dostupný počet vláken
    int cores = Runtime.getRuntime().availableProcessors();

    // pro každé vlákno vytvoříme seznam pozic, které má ohodnotit
    List<List<int[]>> positionsToSearchPerThread =
        getPositionsToSearchPerThread(cores, board);

    Thread[] threads = new Thread[cores];

    // jednotlivá vlákna spustíme
    for (int i = 0; i < cores; i++) {
        threads[i] = new Thread(new Runnable() {
            @Override
            public void run() {
                // tato metoda již nainicializuje proměnné pro jednotlivá vlákna
            }
        });
    }
}
```



```
        // a spusti samotny minimax algoritmus na jednotlivé pozice
        findBestPositionToPlayOneThread(board, onMove, maxPlayer,
            depth, positionsToSearchPerThread.get(i));
    }
});
threads[i].start();
}

// počkame na dokončení jednotlivých vláken
for (int i = 0; i < cores; i++) {
    threads[i].join();
}

// vybereme nejlepší zjištěnou pozici
bestPosition = getBestPositionFromThreads();

// vrátíme nejlepší pozici
return bestPosition
}
```

Kapitola 7

Uživatelské testování

Testování na uživateli je jednou z nejlepších metod, jak ověřit správnou funkčnost a uživatelskou přívětivost aplikace. Díky pozorování a zpětné vazbě jednotlivých uživatelů je možné zjistit chyby designu, nápady, jak použitelnost aplikace zvýšit a pochopit chování uživatelů při používání dané aplikace[52].

Pro samotné testování není třeba mnoho uživatelů. Pro odhalení většiny chyb stačí provést testování s 5 uživateli. Při každém dalším účastníkovi již míra nalezení nových chyb výrazně klesá[53].

7.1 Průběh a uživatelé

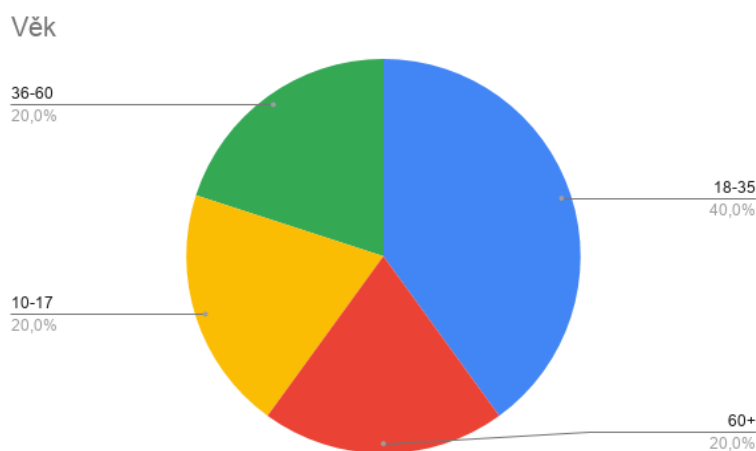
Vzhledem ke karanténním opatření z důvodu koronavirové pandemie v době psaní této práce nebyl možný osobní kontakt s uživateli, kteří aplikaci testovali. Jako náhrada byl zvolen dotazník, který byl rozeslán a testování proběhlo distanční formou.

7.1.1 Testování uživatelé

Testování se zúčastnilo celkem 10 osob ve všech věkových kategoriích, viz obrázek 7.1. Jednalo se o 6 mužů a 4 ženy, viz obrázek 7.2. Většina uživatelů, celkem 7, denně používá OS Android a řadí se převážně ke středně pokročilým uživatelům, viz obrázky 7.3 a 7.4.

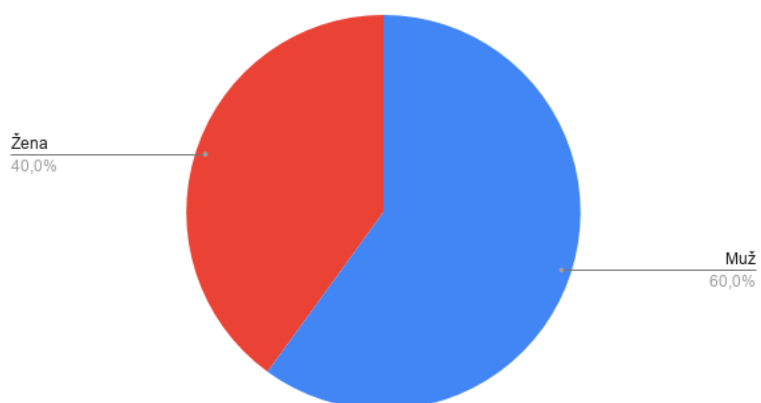
Na uživatele, kteří byli vybráni, nebyly kladeny žádné nároky, neboť stávající aplikace cílí na všechny kuřáky, kteří jsou napříč celým spektrem společnosti.

Důležité je, že se testování zúčastnili jak zkušení uživatelé, tak nováčci, kteří nemusí být natolik zkušení v ovládání chytrého mobilního telefonu a mohou mít větší problémy s uživatelskou přívětivostí aplikace.

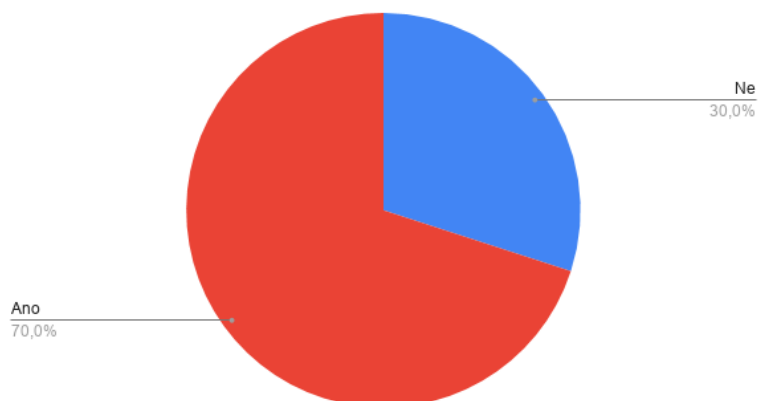


Obrázek 7.1. Graf se zastoupením věkových skupin

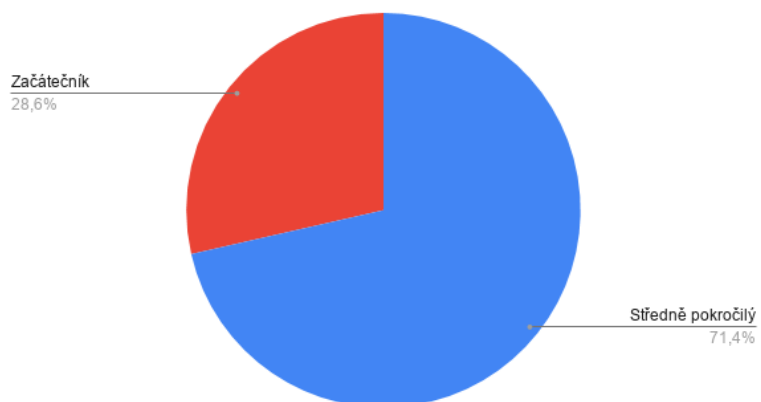
Pohlaví

**Obrázek 7.2.** Graf se zastoupením pohlaví

Používáte denně zařízení s OS Android?

**Obrázek 7.3.** Graf se zastoupením uživatelů OS Android

Jak pokročilý uživatel OS Android jste?

**Obrázek 7.4.** Graf s pokročilostí uživatelů v používání OS Android

7.1.2 Testovací dotazník

Dotazník je součástí přílohy A. Je rozdělen na 3 části. V první části je obecně představena stávající aplikace pro podporu odvykání kouření, jež je následována sérií otázek o osobních informacích o uživateli, jako je věk, pohlaví a míra zkušeností s OS Android. Míra zkušeností je vyplňována pouze v případě, že uživatel denně OS Android používá.

Druhou částí je samotné testování, na začátku sekce jsou uvedeny pokyny, které má uživatel vykonat. Pokyny jsou obecné, jde především o vyzkoušení si jednotlivých her, případně otestování složitější funkcionality, jako je škálování herní plochy u hry Piškvorky, či kalibrace senzorů pro hry Labyrint a Kulička. Poté, co uživatel splní jednotlivé pokyny, je vyzván k zhodnocení jednotlivých her.

Třetí a poslední částí dotazníku je série otázek, která zjišťuje celkovou spokojenost s výběrem a funkčností her v celém modulu. Stejně jako jsou zjišťovány nápady na budoucí rozvoj modulu.

7.2 Vyhodnocení

Celé testování proběhlo bez větších zádrhelů. Všem uživatelům se podařilo aplikaci nainstalovat a spustit, stejně jako si zahrát všechny hry. Objevené chyby nebyly kritické a týkaly se především vylepšení uživatelského rozhraní. Relevantní připomínky a chyby byly po dokončení testování opraveny a doimplementovány do aplikace. Hodnocení jednotlivých her je popsáno níže. Všechny odpovědi společně s dotazníkem jsou součástí přílohy A.

■ Hra 2048

Hra 2048 byla mezi uživateli oblíbená, její průměrné skóre bylo 8,5/10 a 4x byla ve výběru her, které by si uživatelé zahráli znovu. Na hře uživatelé ocenili možnost měnit rychlost animací. Také byla objevena chyba, kdy pokud se hráč posunul o tah zpět, následně tah vpřed a opět tah zpět, pak se již nic nestalo. Tato chyba byla po ukončení testování upravena, stejně jako byl implementován návrh jednoho uživatele, aby pokud není tah zpět, či vpřed dostupný, tak je tlačítko s akcí spojené zašedlé.

■ Hra Kulička

Hra Kulička skončila s průměrným skóre 8,7/10. Ve hře došlo k nalezení chyby, kdy se po návratu na výběr herní úrovně neaktualizovala maximální herní úroveň. Jednomu uživateli by se líbilo, kdyby v prvních úrovních, kdy je na herní ploše kuliček málo, byly tyto kuličky větší. Kalibrovat senzory se povedlo většině uživatelů, avšak většina z nich by bez tohoto upozornění nevěděla, že zde tato možnost existuje. Tento problém byl vyřešen přidáním informace o možnosti kalibrace v návodu hry.

■ Hra Bubliny

Hra Bubliny skončila s průměrným skóre 9/10. Uživatelé s ní byli spokojeni. Jeden uživatel zpočátku nevěděl, co dělat, ale následně vše pochopil. Jeden uživatel objevil vážnou chybu, kdy při rychlém klikání a střelení bublin došlo k pádu hry. Chyba byla opravena.

■ Hra Labyrint

Hra Labyrint skončila s průměrným skóre 9,2/10 a jednalo se tak o nejoblíbenější hru z výběru. Uživatelům se líbila fyzika pohybu kuličky. Jediné, co by uživatelé ocenili, by bylo, kdyby kulička nemusela být v cíli celým průměrem, ale stačilo, aby se jej pouze dotkla.

■ Hra Reversi

Hra Reversi skončila s průměrným skóre 7,8/10. Jednalo se o nejméně oblíbenou hru a to hlavně z důvodu, že uživatelé tuto hru neznají a tudíž je nebavila. Chyby

ve hře nebyly nalezeny žádné. Jako vylepšení bylo navrženo nastavení obtížnosti protihráče.

■ Hra Piškvorky

Hra Piškvorky skončila s průměrným skóre 8,3/10. Škálovat herní plochu se podařilo všem uživatelům. Jediná výtka byla k tomu, že se herní plocha škáluje k jedné straně displeje a uživatel pak musí desku posunout. Stejně jako u hry Reversi bylo navrženo, aby šla nastavovat obtížnost protihráče.

Průměrné skóre celého modulu bylo 9,3/10, což je velmi dobrý výsledek. Nejoblíbenějšími byly hry Bubliny, Kulička a Labyrint. Bylo to převážně z toho důvodu, že hry jsou jednoduché na hraní a člověk u nich nemusí přemýšlet. Naopak deskové hry Reversi a Piškvorky natolik oblíbené nebyly. Celkově však byli uživatelé spokojeni. Na otázku o jaké hry by mohl být modul rozšířen, byly získány nápady na hry jako Bombičky, Had, či panáček přeskakující překážky při klepnutí na obrazovku.

Kapitola 8

Závěr

Cílem této práce bylo rozšíření stávající aplikace pro podporu odvykání kouření o metodu rozptýlení, která by pomohla kuřákům v kritické fázi odvykání se ke kouření nevrátit. V rámci práce mělo dojít k analýze technik a volby techniky pro tuto práci, následně k seznámení se s vývojem aplikací pro OS Android, samotná implementace a následné testování na uživateli.

Všechny cíle této práce byly úspěšně dokončeny. Nejprve byla provedena analýza dostupných technik rozptýlení. Na základě této analýzy byl jako rozšíření zvolen modul miniher. Po této volbě bylo vybráno 6 miniher, konkrétně hry 2048, Kulička, Labyrint, Bublíny, Reversi a Piškvorky. Jednotlivé hry byly následně navrženy a implementovány. Před implementací došlo k seznámení se s vývojem aplikací pro OS Android.

V rámci implementace došlo k řešení několika různých problémů. Prvním bylo dynamické vykreslování animací na herní desku, které se týkalo všech her. Dalším problémem byla práce s akcelerometrem ve hrách Kulička a Labyrint a následná implementace reálného chování pohybu kuliček po herní ploše, jejich vzájemných srážek a odrazů o okraje obrazovky. Pro deskové hry Reversi a Piškvorky museli být implementováni umělí protihráči. Pro hru Reversi byl implementován bezstavový chamtivý hráč, který pracuje s maticí hodnot jednotlivých pozic. U hry Piškvorky byl formou knihovny implementován umělý protihráč na principu algoritmu minimax, který byl následně zpararelizován. Pro hru Piškvorky bylo také implementováno škálování herní plochy, které velmi zlepšilo uživatelský zážitek při ovládání této hry.

V závěru práce došlo k testování modulu na vzorku 10 uživatelů, díky čemuž došlo k ověření funkčnosti a uživatelské přívětivosti jednotlivých her a modulu jako takového. Na základě testování bylo upraveno několik uživatelských prvků tak, aby uživatelské rozhraní a funkčnost celého modulu byla uživatelsky přívětivější.

Modul jako celek je připraven k distribuci pro větší množství uživatelů, která je plánována v nejbližší době po dokončení této práce pomocí začlenění modulu do produkční verze aplikace, která je volně stáhnutelná na Google Play.

Literatura

- [1] Kulháněk A., Gabrhelík R., Novák D., Burda V. a Brendryen H. eHealth Intervention for Smoking Cessation for Czech Tobacco Smokers: Pilot Study of User Acceptance. *Adiktologie*. 2019, 81-85.
- [2] Michie Susan, Hyder Natasha, Walia Asha a West Robert. Development of a taxonomy of behaviour change techniques used in individual behavioural support for smoking cessation.. *Addictive behaviors*. 2011, 315-319.
- [3] World Health Organization. *WHO report on the global tobacco epidemic 2011*. https://www.who.int/tobacco/global_report/2011/en/.
- [4] World Health Organization. *Fact Sheets - Tobacco*. <https://www.who.int/news-room/fact-sheets/detail/tobacco>.
- [5] Ron Borland, Timea R Partos, Hua-Hie Yong, K Michael Cummings a Andrew Hyland. How much unsuccessful quitting activity is going on among adult smokers? Data from the International Tobacco Control Four Country cohort survey. *Addiction (Abingdon, England)*. 2012, 107 (3), 673—682. DOI 10.1111/j.1360-0443.2011.03685.x.
- [6] B. Ploderer, W. Smith, J. Pearce a R. Borland. "A mobile app offering distractions and tips to cope with cigarette craving: a qualitative study". *JMIR Mhealth Uhealth*. "2014", "2" ("2"), "e23".
- [7] Rachel N Carey, Lauren E Connell, Marie Johnston, Alexander J Rothman, Marijn de Bruin, Michael P Kelly a Susan Michie. "Behavior Change Techniques and Their Mechanisms of Action: A Synthesis of Links Described in Published Intervention Literature". *Annals of Behavioral Medicine*. 2018, 53 (8), 693-707. DOI 10.1093/abm/kay078.
- [8] EM. Greenhalgh, S. Stillman a C Ford. *7.8 How can relapse be prevented*. 2016. <https://www.tobaccoinaustralia.org.au/chapter-7-cessation/7-8-how-can-relapse-be-prevented>.
- [9] J. Balmford a R. Borland. "What does it mean to want to quit?". *Drug Alcohol Rev*. "2008", "27" ("1"), "21–27".
- [10] R. Borland, H. H. Yong, J. Balmford, J. Cooper, K. M. Cummings, R. J. O'Connor, A. McNeill, M. P. Zanna a G. T. Fong. "Motivational factors predict quit attempts but not maintenance of smoking cessation: findings from the International Tobacco Control Four country project". *Nicotine Tob. Res.*. "2010", "12 Suppl" "4–11".
- [11] J. R. Hughes, J. Keely a S. Naud. Shape of the relapse curve and long-term abstinence among untreated smokers. *Addiction*. "2004", "99" ("1"), "29–38".
- [12] C. Y. Lam, M. S. Businelle, C. J. Aigner, J. B. McClure, L. Cofta-Woerpel, P. M. Cinciripini a D. W. Wetter. "Individual and combined effects of multiple high-risk triggers on postcessation smoking urge and lapse". *Nicotine Tob. Res.*. "2014", "16" ("5"), "569–575".

- [13] Jessica Skorka-Brown, Jackie Andrade, Ben Whalley a Jon May. Playing Tetris decreases drug and other cravings in real world settings. *Addictive Behaviors*. 2015, 51 165–170. DOI 10.1016/j.addbeh.2015.07.020.
- [14] *GlobalStats - Zastoupení mobilních operačních systémů na trhu*.
gs.statcounter.com/os-market-share/mobile/worldwide/.
- [15] "Android". "Android 5.0 Lollipop".
<https://www.android.com/versions/lollipop-5-0/>.
- [16] John Callaham. *The history of Android OS: its name, origin and more*.
<https://www.androidauthority.com/history-android-os-name-789433/>.
- [17] Andrew Binstock. *Java's 20 Years Of Innovation*.
www.forbes.com/sites/oracle/2015/05/20/javas-20-years-of-innovation/#475df6c11d7c.
- [18] "Jain Ayusch". "Android Internals: ART vs DVM deep dive".
android.jlelse.eu/android-internals-art-vs-dvm-deep-dive-def34cf664d7.
- [19] "Maxim Shafirov". "Kotlin on Android. Now official".
<https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official/>.
- [20] "Frederic Lardinois". "Kotlin is now Google's preferred language for Android app development".
techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development.
- [21] "Android Developers". "Activity".
<https://developer.android.com/reference/android/app/Activity>.
- [22] "Android Developers". "Services overview".
<https://developer.android.com/guide/components/services>.
- [23] "Android Developers". "Broadcasts overview".
<https://developer.android.com/guide/components/broadcasts>.
- [24] "Android Developers". "Content providers".
<https://developer.android.com/guide/topics/providers/content-providers>.
- [25] "Android Developers". "Fragments".
<https://developer.android.com/guide/components/fragments>.
- [26] "Android Developers". "App Manifest Overview".
<https://developer.android.com/guide/topics/manifest/manifest-intro>.
- [27] "Android Developers". "Layouts".
<https://developer.android.com/guide/topics/ui/declaring-layout>.
- [28] "Android Developers". "Linear Layout".
<https://developer.android.com/reference/android/widget/LinearLayout>.
- [29] "Android Developers". "Relative Layout".
<https://developer.android.com/guide/topics/ui/layout/relative>.
- [30] "Android Developers". "Constraint Layout".
<https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout>.
- [31] "Android Developers". "View".
<https://developer.android.com/reference/android/view/View>.
- [32] "Daniele Bottillo". "Creating Android custom view".
<https://medium.com/@dbottillo/creating-android-custom-view-6d8d46122cf5>.

-
- [33] "Android Developers". "*Canvas*".
<https://developer.android.com/reference/android/graphics/Canvas>.
- [34] "Android Developers". "*Introduction to animations*".
<https://developer.android.com/training/animation/overview>.
- [35] "Android Developers". "*Data and file storage overview*".
<https://developer.android.com/training/data-storage>.
- [36] "Android Developers". "*SharedPreferences*".
<https://developer.android.com/reference/android/content/SharedPreferences>.
- [37] "Android Developers". "*Save data using SQLite*".
<https://developer.android.com/training/data-storage/sqlite>.
- [38] "Android Developers". "*Motion sensors*".
https://developer.android.com/guide/topics/sensors/sensors_motion.
- [39] "Android Developers". "*Drag and scale*".
<https://developer.android.com/training/gestures/scale>.
- [40] Game 2048. *Game 2048*.
<https://2048game.com/>.
- [41] Gabriele Cirulli. *2048*.
<https://play.google.com/store/apps/details?id=com.gabrielecirulli.app2048>.
- [42] S2Apps.com. *2048*.
<https://play.google.com/store/apps/details?id=com.s2apps.game2048>.
- [43] "Ted Landau". *Othello : Brief and Basic*.
<http://www.tlandau.com/files/Othello-B%26B.pdf>.
- [44] Samssoft. *Strategy Guide for Reversi and Reversed Reversi*.
<http://samssoft.org.uk/reversi/strategy.htm>.
- [45] "Arcade history". "*Puzzle Bobble*".
<https://www.arcade-history.com/?n=puzzle-bobble&page=detail&id=21224>.
- [46] Bubble Shooter. *Bubble Shooter*.
<https://play.google.com/store/apps/details?id=bubbleshooter.orig>.
- [47] MiniGamz. *Bubble Shooter Space*.
play.google.com/store/apps/details?id=com.minigamz.bubbleshooterspace.
- [48] "Encyklopedie fyziky". "*Odraz vlnění*".
<http://fyzika.jreichl.com/main.article/view/184-odraz-vlneni>.
- [49] "Khan Academy". "*What are elastic and inelastic collisions?*".
<https://www.khanacademy.org/science/physics/linear-momentum/elastic-and-inelastic-collisions/a/what-are-elastic-and-inelastic-collisions>.
- [50] "Rajiv Bakulesh Shah". "*minimax*".
<https://xlinux.nist.gov/dads/HTML/minimax.html>.
- [51] L.V. Allis, H.J. vanden Herik a M.P.H. Huntjens. *Go-Moku and Threat-Space Search*.
web.archive.org/web/20140411074912/http://chalmersgomoku.googlecode.com/files/allis1994.pdf.
- [52] "Kate Moran". "*Usability Testing 101*".
<https://www.nngroup.com/articles/usability-testing-101/>.
- [53] "Jakob Nielsen". "*Why You Only Need to Test with 5 Users*".
<https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.

Příloha A

Testovací dotazník

A.1 O Aplikaci

Testovaná aplikace je mobilní aplikace pro podporu odvykání kouření založená na klinických studiích a praxi. Aplikace byla vyvinuta ve spolupráci s Klinikou adiktologie 1. lékařské fakulty Univerzity Karlovy v Praze. Proces odvykání ve spolupráci s touto aplikací je rozdělen do 3 fází o celkové délce trvání 2 měsíců a každodenním 5-10 minutovým terapeutickým sezením s virtuálním adiktologem Adamem.

Tato práce rozšiřuje tzv. SOS tlačítko o modul miniher, který bude poskytovat okamžité rozrušení a odvedení pozornosti od chuti na cigaretu.

A.2 Zadání

Prosím, splňte úkoly dle zadání.

- Zapněte aplikaci, přihlašte se a dostaňte se na modul miniher skrze SOS tlačítko.
- Zkuste si postupně zahrát jednotlivé hry.
- U her Kulička a Labyrint zkuste najít kalibraci senzorů a zkalibrovat je.
- U hry Piškvorky zkuste škálovat herní plochu.

A.3 Otázky

Pod otázkami jsou uvedeny odpovědi respondentů.

■ Věk

- 10-17 – 2x
- 18-35 – 4x
- 36-60 – 2x
- 60+ – 2x

■ Pohlaví

- Muž – 6x
- Žena – 4x

■ Používáte denné zařízení s OS Android?

- Ano – 7x
- Ne – 3x

■ Jak dlouho používáte OS Android?

- Méně než rok

- 1-3 roky – 2x
- Více než 3 roky – 5x

■ **Jak pokročilý uživatel OS Android jste?**

- Začátečník (telefon používám pouze k volání a SMS) – 2x
- Středně pokročilý (umím používat většinu funkcionalit, telefon umím efektivně ovládat) – 5x
- Profesionál

■ **Jak se Vám líbila hra 2048? Na škále od 1 do 10.**

Odpovědi: 7, 7, 7, 8, 8, 8, 10, 10, 10, 10.

■ **Narazil jste na nějaké problémy, nebo Vás napadají vylepšení?**

- Nevím, jestli je to způsobené emulátorem, ale když dám šipku doleva a pak šipku doprava a pak chci opět použít šipku doleva, tak už žádnou akci nevyvolá. Jinak hra super, jen mi přijde zbytečný posuvník rychlosti, ale to je subjektivní názor.
- Nenarazila, moc se mi líbí funkcionalita zrychlení a zpomalení pohybu dlaždic.
- Pokud už nemohu více kroků zpět, tak by dle mého dané tlačítko mělo změnit barvu, abych věděl, že je neaktivní.
- Zprvu jsem nepochopil smysl hry. Poté, co jsem si všiml tlačítka nápovědy již bylo vše v pořádku.

■ **Jak se Vám líbila hra Kulička? Na škále od 1 do 10.**

Odpovědi: 8, 8, 8, 8, 8, 9, 9, 9, 10, 10.

■ **Narazil jste na nějaké problémy, nebo Vás napadají vylepšení?**

- V prvních úrovních, kdy je pouze pár kuliček ve hře, bych ocenila větší velikost kuliček. Na velkém displeji telefonu jsou kuličky opravdu malinké.
- Hratelnost je super, ale bez předešlého shlednutí pravidel jsem nevěděl, že mám kuličky dostat doprostřed v nějakém pořadí. Doporučil bych tedy pravidla zobrazit při zapnutí hry.
- Při návratu na výběr úrovně se mi neaktualizuje maximální úroveň.

■ **Jak se Vám líbila hra Bubliny? Na škále od 1 do 10.**

Odpovědi: 8, 8, 8, 9, 9, 10, 10, 10.

■ **Narazil jste na nějaké problémy, nebo Vás napadají vylepšení?**

- Když klikám moc rychle, tak mi aplikace spadne.
- Na začátku jsem nevěděl, co mám dělat.

■ **Jak se Vám líbila hra Labyrint? Na škále od 1 do 10.**

Odpovědi: 5, 8, 9, 10, 10, 10, 10, 10, 10.

■ **Narazil jste na nějaké problémy, nebo Vás napadají vylepšení?**

- Kulička musí být v cíli celým průměrem, osobně bych preferoval konec pouze po dotyku. Obtížnost bludišť by bylo dobré zvýšit nebo přidat další prvky (např. oblasti, kterých se hráč nesmí kuličkou dotknout).
- Moc oceňuji velikost kuličky, měkké odražení kuličky od stěn a plynulou reakci na mé pohyby.
- Aby kulička nemusela být v cíli celá.

■ **Podářilo se Vám zkalibrovat senzory?**

- Ano – 7x
- Ne – 3x

■ **Pokud se Vám nepodařilo zkalibrovat senzory, popište, kde jste hledal, či z jakého důvodu se to nepodařilo.**

- Nenašel jsem.
- Bez upozornění bych nevěděla, že tato možnost existuje.
- Bylo by dobré dát uživateli vědět, že kalibrace úspěšně proběhla – po stisknutí tlačítka se zdánlivě nic nedělo.
- Sice se mi to povedlo, ale nejdříve jsem možnost kalibrace hledala pod ikonkou info už přímo ve hře. Až poté mě napadlo, že by ta možnost mohla být ve výběru úrovní hry.
- Prvně jsem hledal u hry Labyrint přímo v úrovni. Pak jsem si toho všiml při kroku zpět. Ale na první dobrou bez této otázky jsem to přehlédl.

■ **Jak se Vám líbila hra Reversi? Na škále od 1 do 10.**

Odpovědi: 5, 5, 5, 8, 9, 9, 9, 9, 9, 10.

■ **Narazil jste na nějaké problémy, nebo Vás napadají vylepšení?**

- Líbilo by se mi nastavení obtížnosti. Jinak super animace překlápění.
- Hru jsem neznal. Hra mě moc nebaví. Neumím hrát tuto hru.
- Při změně barvy by se dle mého mělo ukázat upozornění, že hráč přijde o rozehranou partii. Při dohrání je špatně čitelný výsledek, takže i přes velice špatnou volbu vlastních tahů věřím tomu, že jsem vyhrál.
- Game over dialog je průsvitný, takže není vidět písmo.

■ **Jak se Vám líbila hra Piškvorky? Na škále od 1 do 10.**

Odpovědi: 5, 6, 8, 9, 9, 9, 9, 9, 9, 10.

■ **Podařilo se Vám škálovat herní plochu, případně měl jste nějaké problémy?**

- 4x Ano
- Možná bych jen ocenila, pokud by se plocha zvětšovala a zmenšovala v místě, kde pohybuji prsty. Takhle se to vždy zvětší či zmenší v jednom určitém místě (vlevo nahoře) a já si potom musím přejet na místo, kde potřebuji být. Přiblíží se to tak většinou úplně mimo herní pole.

■ **Narazil jste na nějaké problémy, nebo Vás napadají vylepšení?**

- Stejně jako u Reversi by se mi líbilo nastavení obtížnosti.
- Přišlo mi, že někdy AI player měl delší odezvu.

■ **Jak se Vám celkově líbí modul? Na škále od 1 do 10.**

Odpovědi: 8, 9, 9, 9, 9, 9, 10, 10, 10, 10.

■ **Jaká hra se Vám líbila nejvíce?**

- 2048
- 2x Labyrint
- 3x Kulička
- 2x Reversi
- 2x Piškvorky
- 2x Bublíny

■ **Napadají Vás nějaká obecná vylepšení? Grafika, orientace v aplikaci, atd..**

- Sjednotit barvy napříč aplikací - homescreen je laděný do zelena, zatímco hry jsou modré.
- Dle mého by to chtělo zapracovat na UX. Říct si prostě, kde by co uživatel hledal..

- Ve hře Piškvorky udělat detailní návod.
- V Reversi je těžký protihráč.
- Úroveň Piškvorek

■ **Jaká hra by se Vám líbila jako rozšíření?**

- Snake
- Flappy bird
- Panáček přeskakující překážky na klepnutí, to by bylo super!
- Z hlediska účelu aplikace bych to viděl asi na Bubbles, u které se člověk nebude nijak nervovat a ani nebude muset příliš myslet.
- Bombičky



Příloha B

Přiložené soubory

K této práci jsou přiloženy následující soubory:

- Text této práce ve formátu pdf.
- Zdrojové soubory pro mobilní aplikaci a knihovnu piškvorkového protihráče.
- Instalační soubor mobilní aplikace ve formátu apk.