



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH APLIKACE PRO ANDROID A JEJÍ STRATEGIE MONETIZACE

ANDROID APPLICATION DESIGN AND MONETIZATION STRATEGY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FILIP DOLEŽEL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN LUHAN, Ph.D., MSc

BRNO 2020

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Filip Doležel**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Manažerská informatika
Vedoucí práce: **Ing. Jan Luhan, Ph.D., MSc**
Akademický rok: 2019/20

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

Návrh aplikace pro Android a její strategie monetizace

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem práce je návrh aplikace pro platformu Android včetně analýzy implementačních nástrojů a návrhu následné implementace a testování. Součástí práce bude i analýza způsobů monetizace této aplikace.

Základní literární prameny:

GENADINIK, A. MobileApp Marketing and Monetization. CreateSpace Independent Publishing Platform, 2014. 149 p. ISBN 978-1502383822.

HORTON, J. Android Game Programming by Example. 1st ed. Birmingham: Packt Publishing Ltd., 2015. 498 p. ISBN 978-1-78528-012-2.

HUGHES, J. Android Apps Marketing: Secrets to Selling your Android App. 1st ed. Indianapolis: QUE Publishing, 2010. 309 p. ISBN 978-0-7897-4633-7.

LAUDON, K. C. and C. G. TRAVER. E-Commerce 2016: Business, Technology, Society. 12th ed. New Jersey: Pearson, 2016. 912 p. ISBN 978-0-133-93895-1.

PORTALES, R. Mastering Android Game Development. 1st ed. Birmingham: Packt Publishing Ltd., 2015. 498 p. ISBN 978-1-78355-177-4.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2019/20

V Brně dne 29.2.2020

L. S.

doc. RNDr. Bedřich Půža, CSc.
ředitel

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

Abstrakt

Bakalářská práce se zabývá návrhem mobilní aplikace pro platformu Android. Posuzuje implementační nástroje určené pro tento vývoj, následně obsahuje návrh implementace a testování, což je nedílnou součástí tvorby aplikace. V další části se věnuje analýze možných způsobů monetizace Android aplikace a vybírá vhodnou strategii monetizace pro vyvíjenou aplikaci. Obsahuje také vytvořenou personu typického uživatele aplikace dle předchozí analýzy.

Summary

The bachelor's thesis deals with the design of a mobile application for the Android platform. It assesses the implementation tools designed for this development, then contains design of implementation and testing which is an integral part of the application development process. The next section is dedicated to an elaborated analysis of possible ways of monetization of an Android application and selection of a suitable monetization strategy for the developed application. It also includes an example of a typical user of the application according to analysis.

Klíčová slova

Android, návrh, mockup, mobilní aplikace, analýza, vývoj, monetizace, Unity, Godot, architektura, analýza uživatelů

Keywords

Android, design, mockup, mobile application, analysis, development, monetization, Unity, Godot, architecture, user analysis

Bibliografická citace

DOLEŽEL, F. *Návrh aplikace pro Android a její strategie monetizace*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2020. 44 s. Vedoucí Ing. Jan Luhan, Ph.D., MSc.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 31. 5. 2020

.....
podpis autora

Poděkování

Na tomto místě bych rád poděkoval vedoucímu práce Ing. Janu Luhanovi Ph.D., MSc za jeho cenné rady, trpělivost a věcné připomínky, které byly velkým přínosem pro tuto práci.

Dále bych také rád poděkoval Ing. Jiřímu Horákovi za věcné rady a připomínky, které k této bakalářské práci měl, a také za uvolení k vypracování oponentury této práce.

Největší poděkování patří mé přítelkyni a rodině za jejich podporu při psaní této práce a podporu po dobu celého studia, bez které by jeho dokončení bylo velmi složité.

Obsah

| | | |
|----------|---|-----------|
| 1 | Úvod | 3 |
| 2 | Vymezení problému a cíle práce | 4 |
| 2.1 | Co si vybavit pod mobilní hrou | 4 |
| 2.2 | Rozdíl mezi mobilní hrou a aplikací | 4 |
| 3 | Teoretická východiska práce | 6 |
| 3.1 | Android | 6 |
| 3.1.1 | Z historie do současnosti | 6 |
| 3.1.2 | Vlastnosti systému | 6 |
| 3.1.3 | Android One | 7 |
| 3.1.4 | Google Play | 8 |
| 3.2 | Architektura Androidu | 8 |
| 3.2.1 | Linux Kernel | 8 |
| 3.2.2 | Hardware Abstraction Layer (HAL) | 9 |
| 3.2.3 | Native C/C++ Libraries | 9 |
| 3.2.4 | Android Runtime | 9 |
| 3.2.5 | Java API Framework | 9 |
| 3.2.6 | System Apps | 10 |
| 3.3 | Komponenty Android aplikace | 11 |
| 3.3.1 | onCreate() | 12 |
| 3.3.2 | onStart() | 12 |
| 3.3.3 | onResume() | 12 |
| 3.3.4 | onPause() | 12 |
| 3.3.5 | onStop() | 12 |
| 3.3.6 | onDestroy() | 12 |
| 3.4 | Barevné spektrum | 13 |
| 3.4.1 | Fyzikální vnímání barev | 14 |
| 3.4.2 | Fyziologické vnímání barev | 15 |
| 3.4.3 | Psychologické vnímání barev | 15 |
| 3.4.4 | Vizuální vnímání barev | 15 |
| 3.5 | Uživatelské rozhraní a prožitek | 15 |
| 4 | Analýza implementačních nástrojů | 17 |
| 4.1 | Unity | 17 |
| 4.1.1 | Prostředí | 18 |
| 4.1.2 | Licenční model | 21 |
| 4.1.3 | Požadavky pro vývojáře | 21 |
| 4.2 | CocosSharp | 21 |
| 4.3 | Godot | 22 |
| 4.3.1 | Prostředí | 22 |
| 4.3.2 | Licence | 23 |
| 4.3.3 | Požadavky pro vývojáře | 24 |
| 4.4 | Zhodnocení | 24 |

| | | |
|-----------|--|-----------|
| 5 | Analýza způsobů monetizace | 27 |
| 5.1 | Zpoplatnění stažení | 27 |
| 5.2 | Reklamy a reklamní systémy | 27 |
| 5.2.1 | Výdělek z reklam | 27 |
| 5.2.2 | AdMob | 28 |
| 5.2.3 | Media.net | 28 |
| 5.2.4 | Platby a nákupy v aplikaci | 28 |
| 5.2.5 | Nákup herní měny | 28 |
| 5.2.6 | Prémiový účet | 29 |
| 5.2.7 | Poplatky v aplikaci | 29 |
| 6 | Analýza uživatelů | 30 |
| 6.1 | Průzkum uživatelů | 30 |
| 6.2 | Typický uživatel | 32 |
| 7 | Návrh a popis aplikace | 34 |
| 7.1 | Popis aplikace | 34 |
| 7.2 | Návrh aplikace | 34 |
| 7.3 | Mockup aplikace | 35 |
| 7.4 | Návrh implementace a testování | 36 |
| 7.5 | Návrh monetizace | 38 |
| 8 | Závěr | 40 |
| 9 | Seznam použitých zkratk a symbolů | 43 |
| 10 | Seznam příloh | 44 |

1. Úvod

Vývoj mobilních aplikací pro Android je v současné době velmi populární. Existuje spousta literatury o vývoji v prostředí Android. Při průzkumu nebyla nalezena literatura, jež by se věnovala kompletním přípravám, které je nutné provést před implementací aplikace. Tyto poznatky by měly pomoci všem začátečníkům, kteří by rádi vyvíjeli mobilní aplikace a neví, jak přesně začít.

V rámci této práce je rozebráno, jaký je rozdíl mezi mobilní aplikací a hrou. Je vysvětlen význam operačního systému Android, jeho historie, současný stav a jaké úlohy v tomto směru plnila společnost Google a jejich program. Pro vývoj aplikace v jakémkoliv operačním systému je nutná alespoň základní znalost architektury systému. Po seznámení s operačním systémem je část práce věnována i komponentám, které jednotlivé Android aplikace musí obsahovat pro jejich správnou kompatibilitu se systémem. Barevné spektrum a kompozice v aplikaci je důležitou součástí uživatelského vnímání aplikace a právě z tohoto důvodu bude v kapitole věnována část barvám a jejich vnímání. Kladný dojem z aplikace ve valné většině případů zajišťuje uživatelské rozhraní. Všemi výše popsány částmi se zabývá kapitola 3.

V kapitole 4 dochází k analýze implementačních nástrojů Unity, CocosSharp a Godot. Porovnání bude provedeno od uživatelského rozhraní až po licenční podmínky a následně bude vyhodnoceno, který implementační nástroj při návrhu aplikace bude nejvíce vyhovovat našim požadavkům.

V případě, že chceme aplikaci monetizovat, je potřeba provést analýzu trhu a analýzu způsobů monetizace. Z těchto poznatků bude vytvořena monetizační strategie. Analýza trhu nám pomůže při návrhu a mockupu mobilní hry a vyplyne z ní i typický uživatel. Pro lepší představu bude vypracována persona typického uživatele. Všechny tyto části jsou obsaženy v kapitolách 5 a 6.

Následný návrh a mockup aplikace s jednotlivými postupy, jak by se mělo v těchto případech postupovat a co vše do těchto částí patří, bude vypracován v kapitole 7. Dále v kapitole nalezneme informace o návrhu implementace a testování aplikace, součástí je také návrh monetizační strategie.

2. Vymezení problému a cíle práce

Jako cíl bakalářské práce jsem si vyznačil navrhnout aplikaci pro Android s ukázkou možností, jak ji monetizovat. Problémem klasických mobilních aplikací bývá fakt, že jsou většinou vyvinuty pro určitou cílovou skupinu a jen málo z nich se dá zařadit do aplikací všeobecných, jako jsou například zprávy, mobilní kontakty či kalendář. Pokud bych chtěl vyvíjet právě univerzální aplikace, například kalendář, vidím problém v tom, že každý mobil vybavený systémem Android tuto aplikaci již obsahuje a málokterý uživatel si tak vyhledává či stahuje jinou aplikaci. Za nevýhodu lze také považovat fakt, že těchto aplikací je Google Play (cloud úložiště aplikací) plné. Z toho důvodu jsem se rozhodl právě pro vytvoření hry. Tuto specifickou aplikaci jsme si vybral také proto, že na ní lze vyzdvihnout většinu používaných a častých způsobů monetizace. Tyto způsoby lze nastavit jednotlivě, můžeme také použít jiný business plán pro každou z her.

2.1. Co si vybavit pod mobilní hrou

Za mobilní hru lze považovat software, který spadá do kategorie takzvaného zábavného software. Tento software slouží především k tomu, aby své uživatele pobavil, odreagoval, případně uklidnil. Lze jej dělit podle různých kategorií. Jednou z možností je dělení na základě platform, pro které je software vytvořen. Například mobilní zařízení versus desktop. Dále se dá řadit podle operačních systémů, jako jsou Windows, IOS, Android a jiné.

Jak už sám nadpis napovídá, mobilní hra je hra, která je vytvořena pro mobilní zařízení. Pod tímto zařízením si většina z nás představí hlavně mobilní telefon či tablet. Tato mobilní zařízení jsou nejčastěji vybavena operačními systémy Android nebo IOS, které jsou ve světě mobilních zařízení nejznámější díky své rozšířenosti, ale také díky svému vstřícnému přístupu pro vývojáře. Ti mohou vyvíjet své hry a následně je publikovat do cloud obchodů, ať už se jedná o Android obchod Google Play či IOS obchod App store. Tyto obchody se po úspěšné kontrole aplikace na nežádoucí faktory postarají o veškerou distribuci a s tím spojené povinnosti, což je ideální přístup pro vývojáře, kteří nemusí pro svoji aplikaci tak složitě vytvářet vlastní prodejní prostředí. [6] [23]

2.2. Rozdíl mezi mobilní hrou a aplikací

V první řadě musíme říct, že i v případě mobilní hry se jedná o mobilní aplikaci. Obě aplikace běží pod operačním systémem a k jejich spuštění je potřeba dodržovat jistá pravidla a vnést do nich stejné vlastnosti, aby s nimi uměl operační systém dobře spolupracovat. Dá se tedy říct, že mobilní hra je podmnožinou toho, co můžeme vnímat jako mobilní aplikace.

Markantním rozdílem zde je však samotný obsah. Typická aplikace v sobě obsahuje jednotlivé formuláře, tlačítka, seznamy, texty či fotografie, které jsou na sebe vázány komplexním způsobem použití a není zde u většiny aplikací nutný velký multimediální obsah. Mobilní hra je však pravý opak. U mobilní hry je vytvořena scéna, ve které se daná hra odehrává. Dává také velký prostor pro zvukové a vizuální vjemy, které spolu dohromady musí dynamicky integrovat. Hra jako taková je poté mnohem složitější na vývoj a výpočetní jednotku zařízení, na kterém je spuštěna. Základem každé hry je zvolení

si takzvaného herního enginu, což je framework, který se stará o dynamické vykreslování všech objektů ve scéně. Každý engine je poté vhodný k jinému typu her a je důležité si dopředu analyzovat, zda uspokojí všechny vaše požadavky a nároky na ně. [6] [23] [22]

3. Teoretická východiska práce

V této části práce stručně rozvedu operační systém Android, jeho historii, srovnání a zastoupení jednotlivých verzí a vysvětlení programu Android One. Také zde přiblížím, jak systém pracuje s každou aplikací a co je nutné pro jeho správnou funkcionalitu. Následně se zaměřím na popis Obchodu Play, kde se nachází aplikace podporované systémem. Barevné spektrum a vnímání barev hraje při vývoji také podstatnou část, a proto je tomuto tématu věnována podkapitola 3.4. Na závěr se dostanu k vysvětlení user experience (česky uživatelského rozhraní), dále jen UI, a jeho význam pro aplikaci, aby byla co nejprívětivější pro uživatele. Samotný návrh aplikace a UI naleznete až v další části.

3.1. Android

Multiplatformní operační systém na bázi Linuxu s open-source distribucí, i takto se dá popsat operační systém Android. Je využíván hlavně pro mobilní zařízení. Své zastoupení nachází především v chytrých telefonech, tabletech nebo hodinkách. Posledních několik let se začíná prosazovat i v chytrých televizorech, autech, pračkách či jiných domácích spotřebičích. [18] [10]

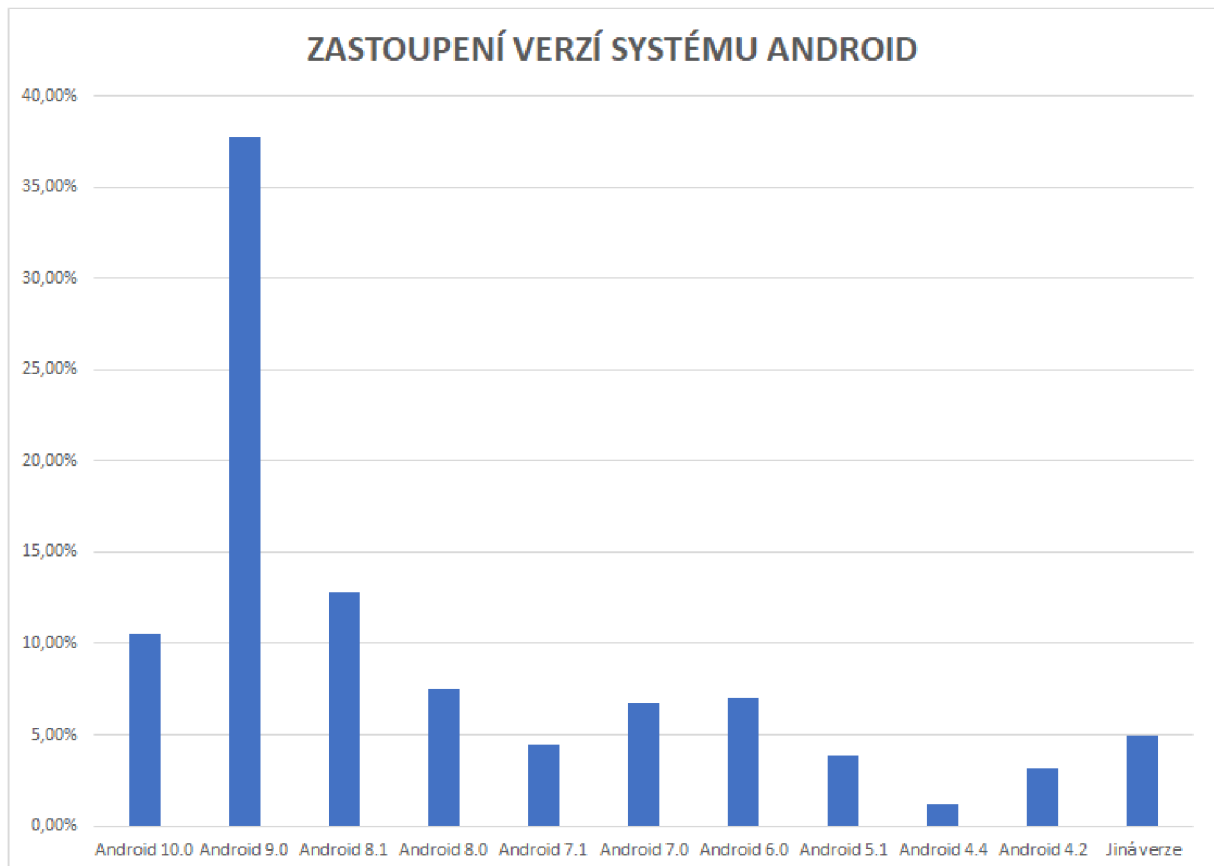
3.1.1. Z historie do současnosti

Společnost Android Inc. vznikla a představila svůj produkt Android ve verzi Alpha 1.0 roku 2003 v Kalifornii. O necelé dva roky později tuto společnost odkoupil Google. Roku 2007 byla založena organizace Open Handset Alliance, která v současné době stojí za zrodem a následným vývojem operačního systému. Tato organizace pod sebou zajišťuje desítky firem od těch nejmenších až po obrovské nadnárodní korporáty v mobilním businessu, jako jsou například společnosti NVIDIA, Intel, Samsung či již zmíněný Google a mnohé další. [24] [18] Počátky prodeje mobilních zařízení nebyly nijak zářivé. Jako první mobil s operačním systémem Android se představil HTC Dream, který měl na trhu chytrých telefonů necelé procento zastoupení. Postupem času se však Android stal čím dál více populárním a v současné době se může pyšnit prvenstvím v počtu zařízení využívajících tento operační systém. Roku 2019 počet aktivních Android zařízení přesáhl počet 2,5 miliardy [29]. V současné době je nejnovější Android 10, jehož původní název je Q [19]. Aktuální zastoupení jednotlivých verzí systému Android na aktivních zařízeních naleznete v grafu 3.1.

3.1.2. Vlastnosti systému

Díky své kompatibilitě a podpoře velkého množství zařízení nejrůznějších značek je tato platforma velmi rozšířená. Tato vlastnost je téměř ojedinělá, jen malé množství operačních systémů toto umožňuje. S velkou výhodou přichází ale i úskalí - chybí optimalizace systému na konkrétní platformu. V tomto ohledu má operační systém Apple iOS významně nahoru. Tato nevýhoda je kompenzována způsobem, jak snadno a rychle se dokáže Android přizpůsobit danému zařízení a umožnit vytváření nastavby. Cestou nastavby se do svých mobilních zařízení vydala většina společností, například Samsung se svým One UI dříve TouchWiz nebo nastavba MIUI od Xiaomi. Existuje zde i verze čistého Androidu,

který neobsahuje žádnou nastavbu, již se vydala společnost Google, která do vybraných telefonů, za předpokladu, že splní určité podmínky, je ochotná distribuovat svou verzi čistého Androidu. Taková zařízení jsou obvykle zařazena do programu Android One. Mobilní zařízení s operačním systémem Android vydává velké množství firem, což zaručuje dynamičtější a progresivnější vývoj nových zařízení. Fakt, že společnosti mohou samovolně upravovat systém a vytvářet vlastní nastavby, zapříčiňuje, že aktualizace musí čekat, dokud je jednotlivé společnosti také nevydají. [24] [10] [16]



Obrázek 3.1: Graf zastoupení jednotlivých verzí OS Android

3.1.3. Android One

Společnost Google v roce 2014 představila program Android One. V té době byl nejaktuálnější Android 5 s názvem Lollipop. Program se vyznačuje tím, že společnost Google vydává grafické uživatelské rozhraní, takzvané GUI, systému Android tak, že výrobci telefonů jej nemohou změnit a běží pouze na vybraných modelech mobilních telefonů, které splní podmínky k připuštění do programu. Výhodou zařazení telefonu do tohoto programu je, že společnost Google se zavazuje k pravidelným aktualizacím. Bezpečnostní aktualizace minimálně po dobu tří let a 2 roky trvající minimální podpora aktualizací operačního systému. [4]

3.1.4. Google Play

Google Play, dříve známý jako Android Market, je služba poskytovaná společností Google. Slouží jako oficiální distributor Android aplikací, které si uživatel může stáhnout do svého mobilního zařízení. Nyní pod sebe Google Play schraňuje velké množství různých aplikací a multimediálního obsahu jako jsou třeba knihy, filmy nebo hudba. Na počátku ale tomu tak nebylo, služby byly rozděleny do několika aplikací a k jejich spojení došlo až při vzniku Google Play.

3.2. Architektura Androidu

Alespoň základní znalost architektury systému, pro který je aplikace určená, je podmínkou každého úspěšného projektu. Proto zde bude postupně popsána architektura Androidu za pomoci využití schématu a obrázku vzestupně. Vzestupný popis je volen z důvodu samotných stavebních bloků Androidu a začíná se tak od samotného jádra, postupně se propracovává až k nejvyššímu oddílu, kde běží aplikace. [11] [18] [19] [16]

3.2.1. Linux Kernel

Linux Kernel je základním kamenem celého operačního systému Android. Upravené jádro tohoto systému je zredukováno na své funkcionality a ochuzeno o funkce, které mobilní operační systém nepotřebuje ke svému běhu a životu. Valná většina funkcí tohoto populárního operačního systému byla poté přizpůsobena tak, aby byl umožněn plynulý běh na mobilním zařízení. [12]

Komponenta Power Management slouží k řízení a distribuci elektrické energie pro ostatní komponenty. Je obsažena právě v jádru operačního systému, kde zastává nepostradatelnou roli, a je postavena na nízké úrovni správy napájení narozdíl od Linuxu samotného, který jej staví na vrchol. Byla implementována nezávisle na Linux Kernel z toho důvodu, že jeho vlastní správa napájení zcela nevyhovovala potřebám mobilního operačního systému. V prostředí Linuxu existují dvě metody správy napájení, a to APM (Advanced Power Management) a ACPI (Advanced Configuration and Power Interface) [18]. Systém Android nevyužívá ani jednu z uvedených možností a namísto toho si zřídil svou vlastní, která se nazývá právě Power Management. Metoda systému umožňuje uvést aplikace do režimu spánku v co nejkratší době a pro spuštění, znovunapojení či neuvádění aplikace do režimu spánku si musí jednotlivé aplikace zažádat prostřednictvím WakeLock, který zabrání přechodu. Pokud tyto WakeLock neexistují, Power Management je přepne do režimu spánku. Nejvýznamnější změny se Power Management dočkal s příchodem Androidu 9 Pie, kdy došlo k jeho přepracování. [11]

Jádro dále zodpovídá za bezpečnost celého systému a přímou interakci s hardwarem mobilního zařízení. Touto interakcí je dosaženo úplné abstrakce hardwaru od vyšších softwarových vrstev, které komunikují pouze s jádrem.

Nalezneme zde i několik dalších komponent sloužících ke správě procesů, paměti, síťové vrstvy, ovladačů a dalších. Interakce se správou napájení, základní grafikou a vstupně výstupními zařízení jsou zde také obsaženy. Pod pojmem vstupně výstupní zařízení si zde lze představit GSM funkce telefonu a dále GPS, 4G, Wi-Fi, fotoaparát, USB a další. Komponenty také zajišťují řízení procesů do takové míry, že jednotlivé procesy mohou běžet současně bez vzájemného ovlivnění. V případě, že dojde k výpadku některého procesu,

nedojde k celkovému výpadku systému a jádro samotné se jej snaží opravit a znovu spustit bez toho, aniž by to uživatel mobilního zařízení zaregistroval. Vzájemnou komunikaci mezi jednotlivými procesy a aplikacemi zajišťuje modul Binder. [11] [18]

3.2.2. Hardware Abstraction Layer (HAL)

Tato vrstva byla původně obsažena pouze v jádru samotného systému. Postupem času s rozmachem mobilních technologií, které kladly čím dál větší důraz na možné využití nových senzorů a hardwarového vybavení, došlo k jejímu rozšíření i mimo jádro. Pro odlehčení práce jádra byla tato komponenta zpřístupněna pro nadřazené rozhraní, které s ní komunikuje místo jádra. Vrstva zpracovává jednotlivé požadavky a následně je zašle jádru na vyhotovení. Samotný HAL se poté skládá z několika modulů obsahující knihovny. Každý modul implementuje rozhraní pro jiný typ hardwarové komponenty. Tyto knihovny jsou načteny v případě, že některá z aplikací si žádá přístup k hardwarovému zařízení. [11]

3.2.3. Native C/C++ Libraries

Nad Linuxovým jádrem nalezneme vrstvu situovaných knihoven, které poskytují přístup aplikacím k různým systémovým komponentám. Knihovny jsou napsány v programovacím jazyku C nebo C++, samotný název vrstvy poté napovídá, že se jedná o nativní knihovny. Vrstva úzce spolupracuje s vrstvou běžící na stejné úrovni Android Runtime a pod nimi umístěnou vrstvou HAL. K těmto knihovnám můžeme přistupovat z API rozhraní Java, které jsou pro tyto účely vytvořeny. Pokud je při vývoji aplikace vyžadován kódový jazyk C nebo C++, můžeme k těmto aplikacím přistupovat nativně z kódu a není tak zapotřebí využívat API. [11]

3.2.4. Android Runtime

Android Runtime je umístěn na stejné vrstvě jako nativní knihovny a pracuje s bitovým mezikódem DEX, který vzniká kompilací souborů CLASS a JAR. Tento bajtový mezikód je i značně kompaktnější než zmíněné soubory. Android Runtime používá proces zvaný Ahead-Of-Time pro překlad bitového kódu do nativních instrukcí procesu v zařízení. U zařízení obsahujících Android 5 a výše se každá aplikace spouští ve vlastním procesu. Android Runtime je konstruován tak, aby zvládl spouštět jednotlivé virtuální nástroje ve větším počtu. Od Androidu 9 tato část systému dokáže převádět soubory DEX z aplikací do efektivnějšího strojového kódu. Dalšími úkoly jsou například podpory ladění, diagnostika a hlášení o možných selháních systému. [11]

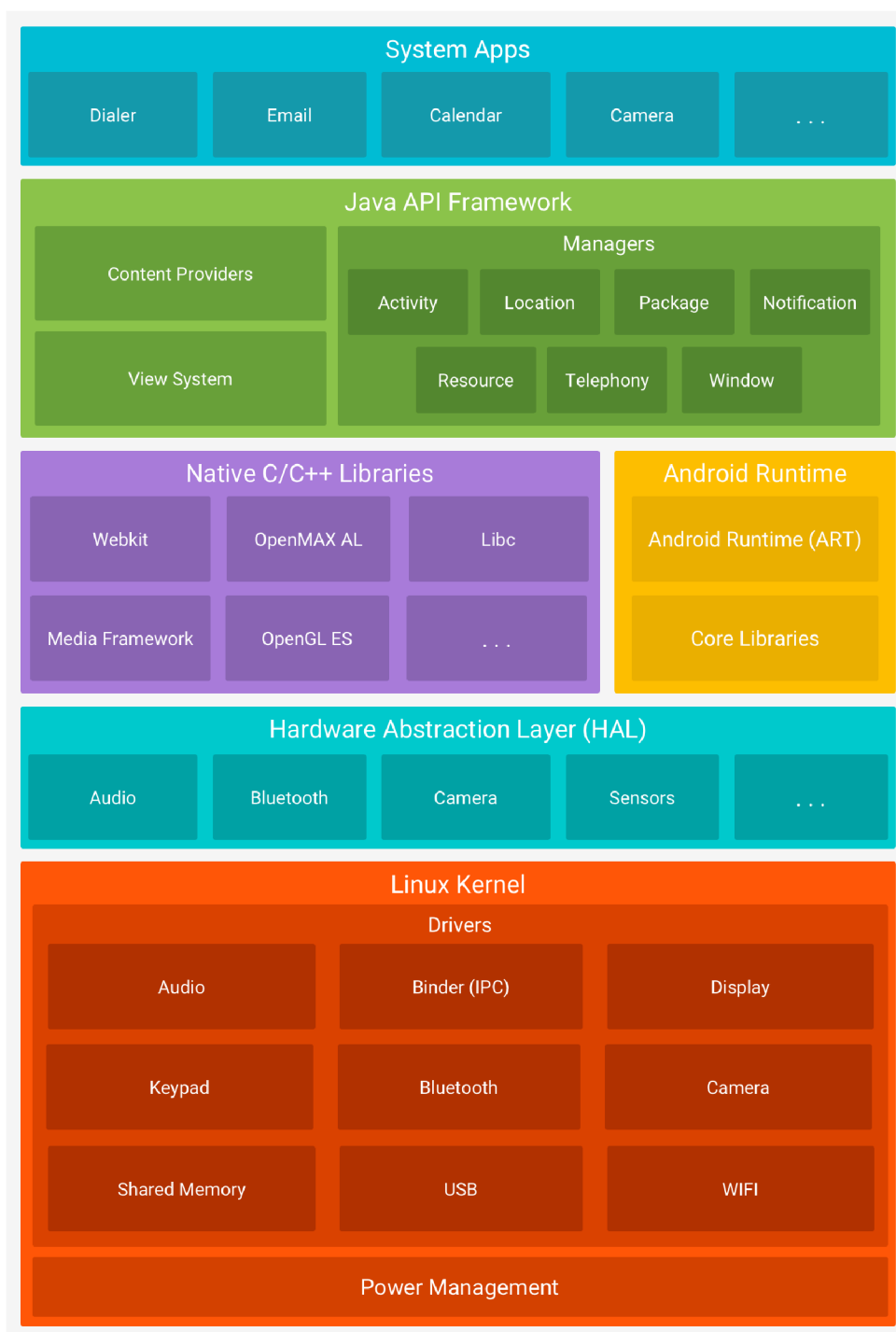
3.2.5. Java API Framework

V této úrovni nalezneme řadu funkcí operačního systému Android, které jsou k dispozici pomocí rozhraní API implementovaného v jazyce Java. Tato vrstva je nejdůležitější vrstvou pro vývojáře Android aplikací, protože aplikacím poskytuje základní systémové služby. Nalezneme zde například modul Package Manager, který slouží ke správě nainstalovaných aplikací, jež jsou zde uloženy v seznamu, který je následně interpretován jako databáze. Dalším důležitým modulem je Windows Manager, jež spravuje okna aplikací a

určuje, která okna se překrývají a které komponenty budou nadřazené. View System se poté stará o základní grafické prvky uživatelského rozhraní. Do prvků můžeme zahrnout ikonky, editace textu, tlačítka a další. Nalezneme zde i službu Activity Manager, která spravuje životní cyklus aplikací. [11]

3.2.6. System Apps

Na nejvyšší úrovni systému stojí aplikace. Pod nimi si můžeme představit například základní sadu aplikací, které v mobilním zařízení nalezneme již od prvního spuštění, jako jsou kalendář, SMS či kamera. Nebo aplikace třetích stran, které si uživatel instaluje sám, a může to být právě mobilní hra. Základní sada aplikací nemá žádná zvláštní privilegia či pravomoci a chová se stejně jako aplikace třetích stran. Systémové aplikace dále umožňují možné ulehčení vámi implementovaných aplikací. Aplikaci musíte přizpůsobit tak, aby do ní zapadly vstupní a výstupní možnosti systémových aplikací. V praxi to znamená, že nemusíte psát komponentu aplikace pro zaslání emailů, ale můžete využít již obsaženou aplikaci ze základní sady k jejich odeslání. [11] [18] [19]



Obrázek 3.2: Architektura systému Android [11]

3.3. Komponenty Android aplikace

Všechny aplikace, které jsou na operačním systému Android spustitelné, se skládají z velké řady přesně definovaných funkcí a metod. Tyto metody a funkce se spouštějí v určitém pořadí za přísného dodržení přesně definovaných situací. Cyklus aktivity aplikace můžeme rozdělit do tří hlavních fází. Do aktivity na popředí, pozastavené a zastavené aktivity. V případě, že byl systém nucen obnovit aktivitu, která se nachází ve fázi pozastavení či

zastavení, dojde k restartu aktivity. Ta se pak následně zobrazí jako aktivita na popředí. Životní cyklus aplikace naleznete na obrázku níže. [9] [18]

3.3.1. onCreate()

Když se aktivita spustí, první se provede metoda onCreate(). Na pozadí mezitím vytváří uživatelské rozhraní, kdy probíhá konfigurace proměnných a objektů potřebných ke správnému fungování aktivit. I když dojde k zavolání metody onCreate(), aktivita zatím zůstává běžet pouze na pozadí a je neviditelná, dokud nedojde k načtení všech potřebných zdrojů. Metoda je také volána při změně vertikálního nebo horizontálního zobrazení. [9]

3.3.2. onStart()

V metodě onStart() se připravují a realizují činnosti potřebné k zobrazení aplikace uživateli a jejího následného reagování na jeho požadavky. Dokud se aktivita nachází v metodě onStart(), stále nekomunikuje s uživatelem. [9]

3.3.3. onResume()

OnResume() je metoda velmi podobná metodě onStart(). Hlavním rozdílem zde je, že k volání metody dochází v situaci, kdy aktivita již byla spuštěna, ale byla jinou aktivitou posunuta do pozadí a nyní je požadována uživatelem znovu do popředí tak, aby s ní uživatel mohl komunikovat. [9]

3.3.4. onPause()

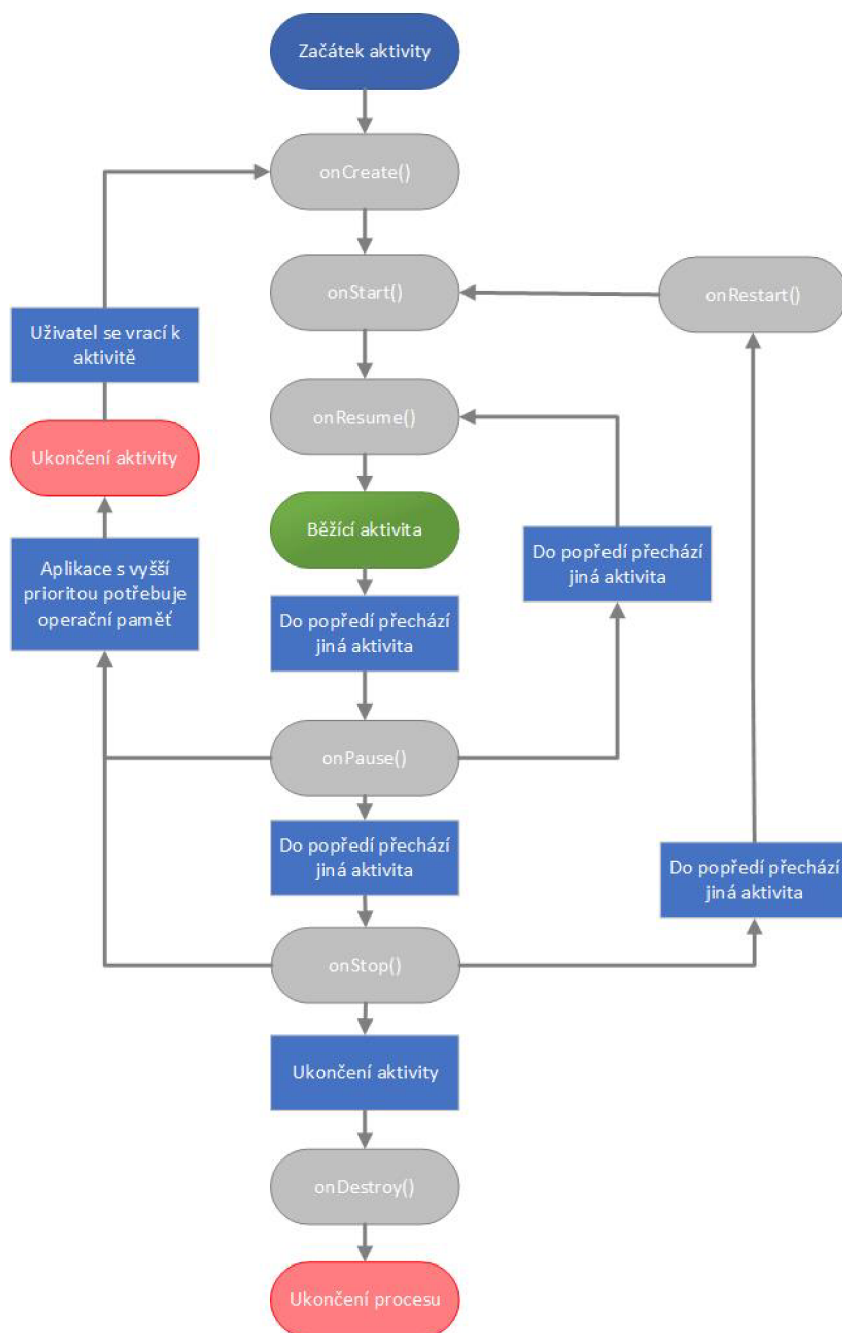
K volání metody onPause() dochází v situaci, kdy spuštěná aktivita je odsunuta do pozadí nově spuštěnou aktivitou. Aktivita, kterou nahrazuje nově spuštěná, se přesouvá do pozadí. V tuto chvíli je vhodné naplánovat automatické ukládání změn, se kterými aktivita pracovala, a uložit je do databáze, souboru či jiných prostředků k uchování dat. [9]

3.3.5. onStop()

Volání metody onStop() probíhá v případě jiného důvodu, než je nedostatek paměti. U této metody není podmíněný způsob volání a nemusí tak k volání dojít vůbec. Nedojde ovšem k ukončení aktivity, ale pouze k jejímu zastavení. Nevyžaduje proto takové množství vynaložených prostředků k chodu a je omezena na minimum. Uživatel tuto aktivitu může znovu zobrazit na popředí. V tomto případě se volá metoda onRestart(). [9]

3.3.6. onDestroy()

Metoda onDestroy() se volá při ukončování životního cyklu aktivity. U této metody opět není podmíněný způsob volání. V případě, že k volání dojde, spuštění ukončování životního cyklu započne bez ohledu na to, zda se jedná o explicitní nebo implicitní aktivitu. [9]



Obrázek 3.3: Životní cyklus aplikace [9]

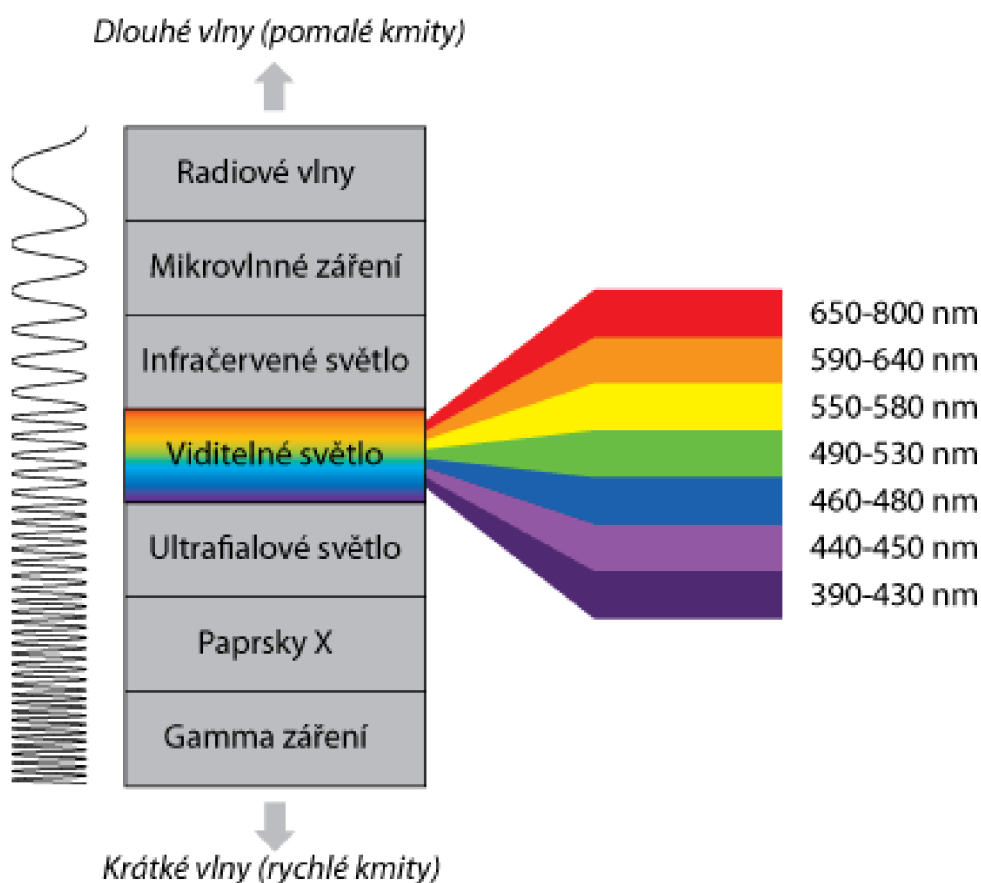
3.4. Barevné spektrum

Barvy, jejich význam a vnímání jsou nedílnou součástí úspěchu aplikace. Význam, jaký mají barvy pro lidské chápání a vnímání, je naprosto nepostradatelným faktorem aplikace. Vnímání barev ovlivňuje velká řada faktorů, proto na ně jednotlivé vědní disciplíny nahlíží zcela jiným způsobem. V případě, že barvy budou voleny a vnímány jinak, než jak je chápán jejich význam širokou veřejností, může snadno dojít k nežádoucím komplikacím a nepochopení ze strany uživatele. [14]

Působení barev můžeme shrnout do čtyř základních působností. Do fyzikální, fyziologické, psychologické a v neposlední řadě vizuální. Asociativní působení barev je časté spojování předmětů s barvou a lze ho zařadit do speciálního psychologického působení barev. Jako asociativní vnímání barev bývá nejčastěji označování modré jako vody, červené jako jablka či zelené jako trávy. Tyto barvy a jejich asociace jsem si nevybral náhodou, jedná se totiž o základní barvy, jejichž kombinací se následně dá stvořit jakákoliv jiná. [14]

3.4.1. Fyzikální vnímání barev

Barva z fyzikálního hlediska je elektromagnetické vlnění vnímatelné lidským okem. Světlo je nedílnou součástí toho, abychom barvy mohli vnímat. Odrazy od předmětů jsou poté jednou ze základních podmínek k vnímání. Různé druhy světelných paprsků mohou zkreslit barevnost či celý význam barev. Lidské oko dokáže vnímat vlnovou délku barev přibližně od 390 nm do 700 nm. Toto vnímání se ovšem může u každého exempláře lišit. Pokud bychom se zaměřili na frekvenci, lidské oči ji dokáží vnímat přibližně v rozsahu 430 THz - 750 THz, vnímaný rozsah se ale stejně jako u vlnové délky může u každého lišit. Pro rozpad světelných paprsků lze použít světelný hranol, který paprsek rozbije na spektrum barev. [14]



Obrázek 3.4: Rozsah viditelného spektra barev [27]

3.4.2. Fyziologické vnímání barev

Za fyziologické vnímání barev se považuje účinnost světla na lidský organismus, zejména pak na lidský zrak a mozek, které tyto informace zpracovávají. Barvy poté mohou mít pozitivní, ale i negativní vliv na lidský organismus. [14]

3.4.3. Psychologické vnímání barev

Toto vnímání vychází z účinku barev na lidskou psychiku. Do této kategorie také spadají asociace, symbolika či stimulace, které v nás barevný podnět vyvolá. Asociace a symbolika jsou jednou z nejčastěji využívaných vlastností zabarvení v aplikaci. Ulehčují tak vnímání lidí, například červené tlačítko je storno nebo odstranit, zatímco zelené značí, že je vše v pořádku.

Tabulka 3.1 ukazuje určitou symboliku a emoce několika barev. Barvy nikdy nemají jen pozitivní nebo negativní emoční význam. Tato skutečnost je spjata především s rozdílností jednotlivých kultur a jejich vnímání symboliky barev. [14]

| Barva | Symbolika | Pozitivní emoce | Negativní emoce |
|---------|-------------------------|---------------------------|---------------------------|
| černá | tma, smrt, nicota | úcta, autorita, vážnost | zlo, strach, nákaza |
| šedá | neutralita, průměrnost | vyváženost, pokora | pasivita, chudoba, nuda |
| bílá | světlo, čistota, mír | nevinost, pořádek | chlad, izolace, nejistota |
| žlutá | světlo, slunce, mládí | radost, rozum, soulad | závist, faleš, nedůvěra |
| zelená | tráva, regenerace, léto | laskavost, naděje, klid | napětí, nadřazenost |
| červená | krev, oheň, srdce | láska, energie, hrdost | nebezpečí, válka, agrese |
| modrá | voda, nebe, zima | spravedlnost, inteligence | deprese, odevzdanost |

Tabulka 3.1: Symbolika a emoce barev

3.4.4. Vizuální vnímání barev

Toto vnímání je dáno tím, jak se jednotlivé barvy reprezentují v ploše vnímání prostoru. Teplé barvy mají tendenci vystupovat do popředí námi vnímaného objektu, zatímco chladné barvy dělají pravý opak.

3.5. Uživatelské rozhraní a prožitek

Uživatelské rozhraní a prožitek můžeme chápat jako sadu metod, pravidel a postupů, které využíváme při návrhu uživatelského prostředí pro daný projekt. Zpravidla do těchto projektů spadají aplikace, webové stránky, informační systémy a jiné, kde se očekává nějaká interakce s uživatelem a poskytovateli projektu není lhostejný vizuální dojem. Návrhář také zodpovídá za smysluplné a relevantní prvky aplikace, aby nedocházelo ke zmatečné funkci.

Při tvorbě uživatelského rozhraní se nejčastěji řídíme trojicí otázek: „Proč“, „Co“ a „Jak“. Návrháři UX začínají otázkou „Proč“, následuje otázka „Co“ a poté „Jak“. Zodpovězením sérií otázek získá návrhář představu produktu, na který bude uživatel pozitivně reagovat.

Otázka „Proč“ zahrnuje především motivaci uživatelů k reakcím v aplikaci. Tato otázka by měla také zahrnovat motivaci uživatelů aplikaci využívat. Pokud neexistuje smysluplná odpověď na otázku, proč na tento prvek má uživatel reagovat, je lepší jej odstranit a vymyslet jiný způsob.

Co se týče otázky „Co“, měla by se aplikovat na to, jaká je funkčnost aplikace. Co vše může nabídnout uživatelům, co můžou s aplikací dělat a co od ní očekávají. Tyto věci se nejlépe zjišťují při testování aplikace uživateli a konzultací s nimi.

Otázkou „Jak“ zodpovídáme související věci s designem a funkcionalitou aplikace. Jakým způsobem estetické prvky zakomponovat do prvků aplikace a jak na ně uživatelé reagují. Stejně jako u předchozí otázky i zde se nejlépe věci spjaté s uživateli zjišťují testováním a dotazníky.

Můžeme také hovořit o 7 faktorech, na kterých se nejlépe zjišťuje, jaký je uživatelský prožitek. Mezi tyto faktory patří: užitečný, použitelný, najít, důvěryhodný, žádoucí, přístupné a cenné. Všechny tyto faktory by měly být aplikovatelné na jakýkoliv prvek v aplikaci. Pokud by některý z faktorů nešel aplikovat, je třeba zvážit jeho působnost a důležitost v aplikaci.

Tvorba uživatelského rozhraní v systému Android se definuje pomocí layoutů. Layouty jsou hierarchicky skládány z View Groups, které jsou složeny z jednotlivých Views. Rozhraní se poté dá definovat dvěma způsoby, buď při běhu aplikace a nebo pomocí XML souboru před samotným během. [14] [13]

4. Analýza implementačních nástrojů

Vývoj aplikace pro operační systém Android je spjat především s Android SDK. Jedná se o sadu nástrojů a knihoven, které nezbytně podporují vytvoření instalačního souboru formátu APK, v němž je aplikace spustitelná pro systém Android. Pro vývoj her není toto prostředí zcela vhodné.

Pro vytvoření hry určené pro mobilní zařízení s operačním systémem Android se často doporučuje sáhnout po game engine software¹. Game engine se nabízí v různých verzích licencí, ať už se jedná o placené za určitých podmínek, placené či neplacené licence. Téměř každý game engine je svou licenční politikou ojedinelý a je zapotřebí jej důkladně znát, aby nedošlo k jejímu porušení.

Game engine vývojáři často nabízí pomyslné usnadnění vývoje hry, kdy si nemusí zajišťovat veškeré propojení s operačním systémem, případně vytvářet všechny potřebné systémy. Většina základních systémů, ať už se jedná o fyziku, umělou inteligenci či v některých případech i grafiku, bývá ve všeobecné základní verzi již obsažená. Zde je však potřeba klást velký důraz na značně omezenou možnost použití. Ve valné většině případů je nutné tyto systémy modifikovat pro dané potřeby. Game engine také ve značné míře pomáhají samotné integraci jednotlivých komponent. Přináší další ulehčení pro vývojáře, kterému dovolí se zabývat především samotnou podstatou hry a jejím laděním. Před začátkem vývoje hry bývá výběr game engine velmi důležitou a neopominutelnou součástí samotného procesu. Z tohoto důvodu níže naleznete jedny z nejpoužívanějších game engine nástrojů, které byly vybrány pro následné porovnání. Jedná se o engine Unity, CocoSharp a Godot. Všechny engine mají několik rozdílných vlastností. Především se jedná o vlastnosti licencí, využití pro rozsah hry, programovací jazyk či vestavěné systémy. Jednotlivé engine se poté hodí i pro různé typy her. Níže naleznete jejich srovnání a popis vývojového prostředí. [16] [25]

4.1. Unity

Multiplatformní game engine Unity se řadí mezi jeden z nejrozšířenějších a nejnámějších enginů pro vytváření her. První verze byla vydána v roce 2005 a již tehdy slavila velký úspěch. Za tu dobu, co Unity figuruje na poli vývoje her, vyšlo několik verzí, z čehož poslední byla vydána v dubnu roku 2020. Od roku 2018 game engine podporoval více než 25 platforem, pro které umožňuje vyvíjet. Unity lze využít k vytváření dvourozměrných (2D) a trojrozměrných (3D) her. S rozmachem technologií bylo umožněno v enginu vyvíjet i programy a hry pro rozšířenou a virtuální realitu, která je v posledních letech na vzestupu. Engine byl postupem času využíván v jiných než herních průmyslech a to především v automobilovém průmyslu, architektuře, stavebnictví, strojírenství a ve filmech. Popularita tohoto enginu dala vzniknout Unity Assets Store, ve kterém nalezneme různé objekty, modely, hudbu či části kódu. Tyto věci mohou pomoci k vývoji projektu a jsou zde umístěny buď zcela zdarma nebo za menší poplatek. Každý komponent, který v Asset Store nalezneme, má svoji vlastní licenci, kterou je nutné prostudovat, aby nedošlo

¹Game engine se používá ke snadnějšímu programování her a obsahuje funkce k jejich animaci, programování a mnoho dalších.

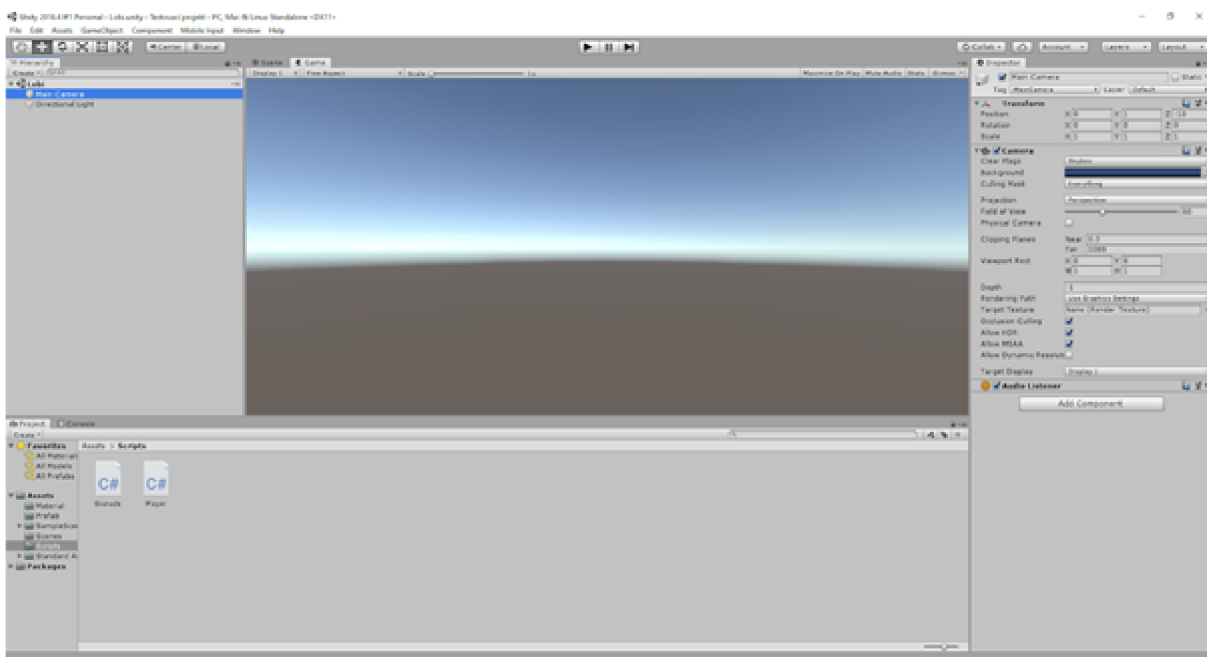
k jejímu porušení. Také zde nalezneme i velkou spoustu prefabů. Prefab je základní objekt obsahující veškeré potřebné věci k jeho integraci do projektu. [21] [30]



Obrázek 4.1: Logo Unity [30]

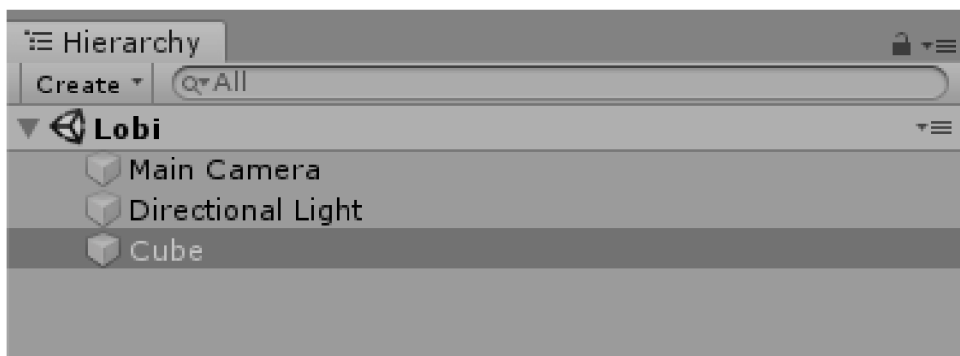
4.1.1. Prostředí

Samotné prostředí je značně intuitivní a skládá se z několika oken, která se k vývoji využívají. Každé okno má své specifické vlastnosti a má na starost rozdílné věci. Na horní liště se v Unity nachází klasické schovávací menu se základním ovládáním importu/exportu projektů, zakládání nových projektů a komponent k tomu potřebných, a v neposlední řadě také ukládání. Záložka Edit umožňuje a rozšiřuje editaci projektu a možnosti spojené s testováním. Assets a jeho menu je určeno k editaci jednotlivých modelů assets. V záložce GameObject je vše spojené s herními objekty jako jsou světelné kompozice a základní 2D a 3D objekty, které lze používat k prototypování či pohledu kamer. Záložka Component je určena k manipulaci s objekty ve scéně. Pro ovládání zobrazených oken, ze kterých se skládá celá pracovní plocha, a k jejich přizpůsobení, se používá záložka Window. Poslední záložka Help funguje jako rychlý rádce.



Obrázek 4.2: Pohled na prostředí Unity

Pokud Unity necháme bez úprav oken v původním rozložení, které nás přivítá při prvním spuštění, v levé části nalezneme seznam objektů v pracovní scéně.

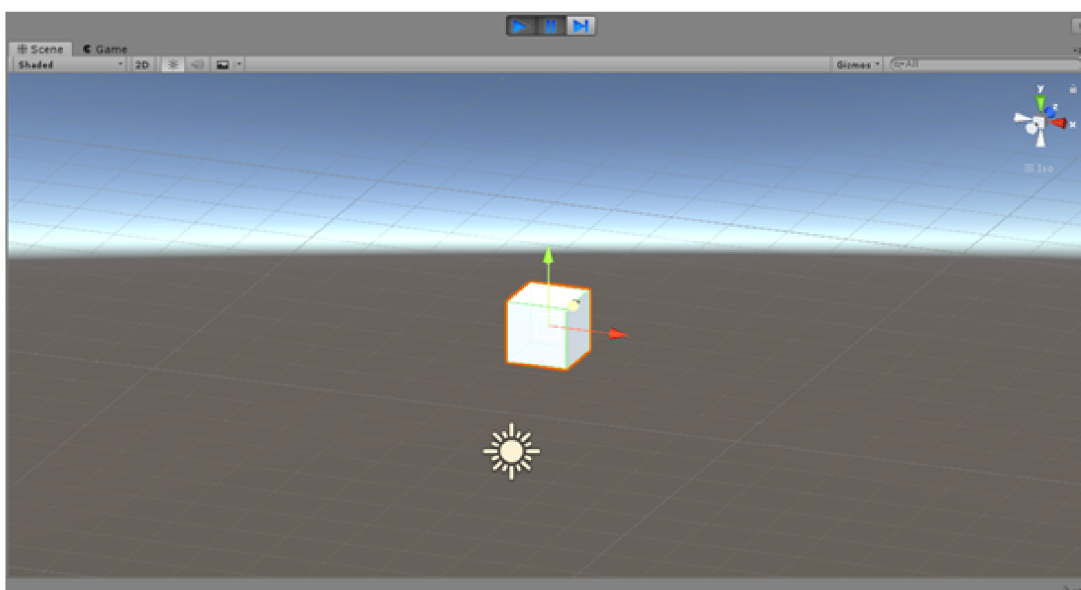


Obrázek 4.3: Seznam objektů pracovní plochy

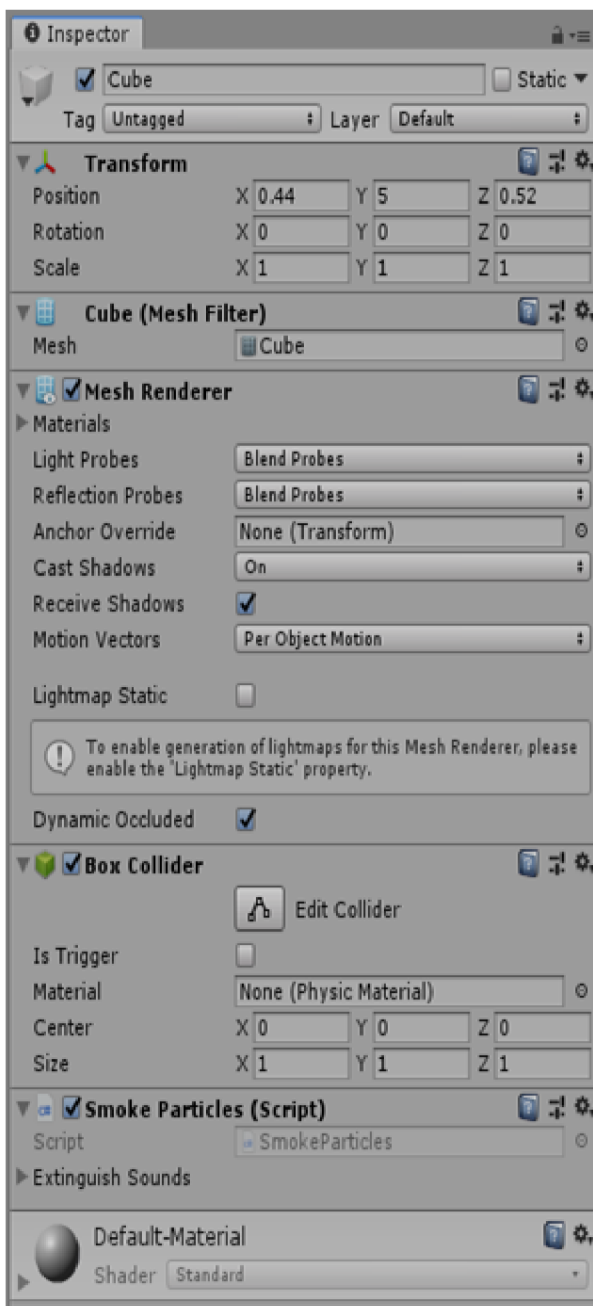
Pod seznamem objektů nalezneme okno se skripty a případnými dalšími assety. Pro tvorbu skriptů se nejčastěji používají programovací jazyky C# a C++. Samotné skripty se poté v Unity neprogramují, ale využívá se programů určených k této činnosti, jako je například Microsoft Visual Studio. Do obsahu skriptu je možné nahlédnout za pomoci pravého okna. Obsah tohoto okna se přizpůsobuje v závislosti na tom, jaký typ assetu je označen a připraven k úpravě.



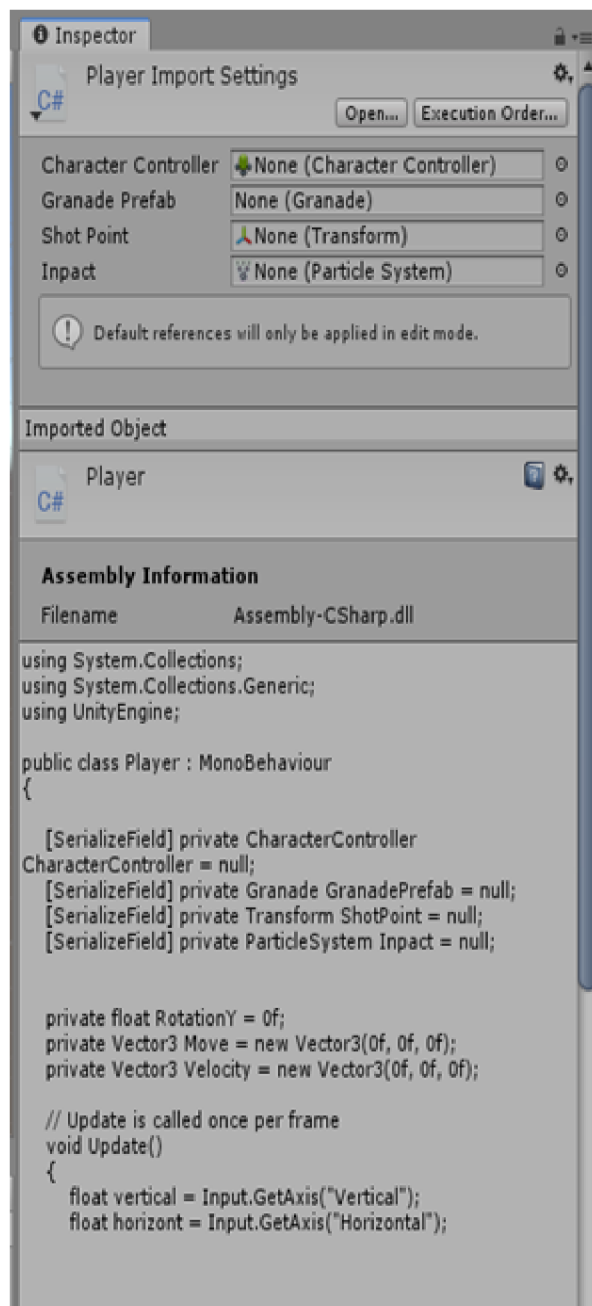
Obrázek 4.4: Seznam skriptů a assetů



Obrázek 4.5: Pracovní plocha



Obrázek 4.6: Pravé pracovní okno přizpůsobené k zobrazení objektu



Obrázek 4.7: Pravé pracovní okno přizpůsobené k zobrazení skriptů

Hlavní pracovní plochou, která je zároveň i centrálním oknem, je scéna. V tomto okně se nachází pohled na vyvíjenou hru, konkrétně na scénu ze hry. V této části engine pracujeme s jednotlivými objekty a situujeme je podle potřeby. Objekty do této části dostáváme především pomocí importu či vložení základních objektů, které Unity nabízí. V pravé části okna nalezneme osový kříž, který se přizpůsobí podle potřeby na 2D nebo 3D grafiku. Nad scénou samotnou jsou 3 tlačítka se symbolikou play, pause a další. Tlačítka se využívají ke spouštění a testování scény, viz obrázek 4.5.

4.1.2. Licenční model

V roce 2016 došlo ke změně licenčních požadavků společnosti Unity. Nyní jsou nastaveny bezplatné licence pro osobní použití, v případě komerčního použití se jedná o bezplatnou licenci, pokud se jedná o osobu nebo malou společnost, která ročně generuje zisk menší než 100 000 \$. Pokud tento zisk přesáhne, dostává se do kategorie placené licence, která je stanovena na 40 až 150 \$ měsíčně na základě ročního generovaného zisku společnosti. [30]

4.1.3. Požadavky pro vývojáře

Pro vývoj v game engine Unity je zapotřebí alespoň střední vývojářský počítač, který obsahuje operační systém Windows nebo macOS. V případě, že využíváme operační systém Windows, je zapotřebí mít verzi Windows 10 s architekturou procesoru x86, x64 podporující SSE2 a grafickými ovladači DX10 a vyšší. Pokud bychom využívali macOS, parametry jsou zde velmi podobné. Tyto požadavky ovšem ruší případ, kdybychom chtěli vyvíjet pro rozšířenou či virtuální realitu. V takovém případě je potřeba minimálně grafická karta Nvidia GTX 1050 Ti, AMD Radeon RX 470 nebo lepší. [30]

4.2. CocosSharp

CocosSharp je jedna z možných sekcí, které poskytuje game engine Cocos2d. Tato sekce je přizpůsobena k tvoření 2D her v prostředí Microsoft Visual Studio. Jak už samotný název napovídá, CocosSharp se programuje v jazyce C# a využívá se zde možností Xamarinu, který je připraven pro vývoj aplikací pro operační systém Android. Xamarin zde následně využívá knihovny obsažené v CocosSharpu pro vývoj her.

Nejedná se o game engine v pravém slova smyslu, ale spíše o platformu k vyvíjení her. V současné době je tato platforma na ústupu. Ten má na svědomí především velké množství game engine, které tuto platformu nahrazují.

Vývoj v této platformě je poněkud složitější než v ostatních popsaných případech. Nemůžeme se spolehnout na předvytvořené věci nebo pomoc ze strany zvoleného softwaru k vývoji hry. Jedním z hlavních nedostatků je kolizní systém, není zde tak vyvinut jako u ostatních porovnávaných programů.

Licenční podmínky jsou vcelku jasné, protože se jedná o bezplatnou licenci k použití. Hardwarová náročnost je stejná jako u softwaru Microsoft Visual Studio. [5]



Obrázek 4.8: Logo Cocos2d [5]

4.3. Godot

Godot patří stejně jako Unity do multiplatformních game engine. Vývoj tohoto engine byl zahájen v roce 2007. Roku 2014 byl zdrojový kód Godotu uvolněn pro veřejnost na GitHubu v rámci licence MIT. Na přelomu roku 2014 a 2015 byla představena verze s označením 1.0. To značilo první stabilní verzi pro vývojáře. V průběhu roku 2015 došlo k velkému updatu v Godotu a připojení k společnosti Software Freedom Conservancy. Dotsud nejvýznamnější update vyšel na začátku roku 2018, kdy došlo k vylepšení práce s 3D grafikou a podporou pro virtuální a rozšířenou realitu. Vývojové prostředí běží na operačních systémech Windows, macOS a Linux a umožňuje vytvářet hry cílené pro počítače, konzole, mobily či internetové prohlížeče. Celkem se jedná o více než jedenáct platform. Programování v prostředí Godot je podporováno jazykem C++ nebo vlastním skriptovacím jazykem s názvem GDScript. Tento skriptovací vysokoúrovňový a dynamický jazyk je velmi podobný jazyku Python. Nově v Godot engine nalezneme i AssetLib s různými druhy assetů. [8]



Obrázek 4.9: Logo Godot [8]

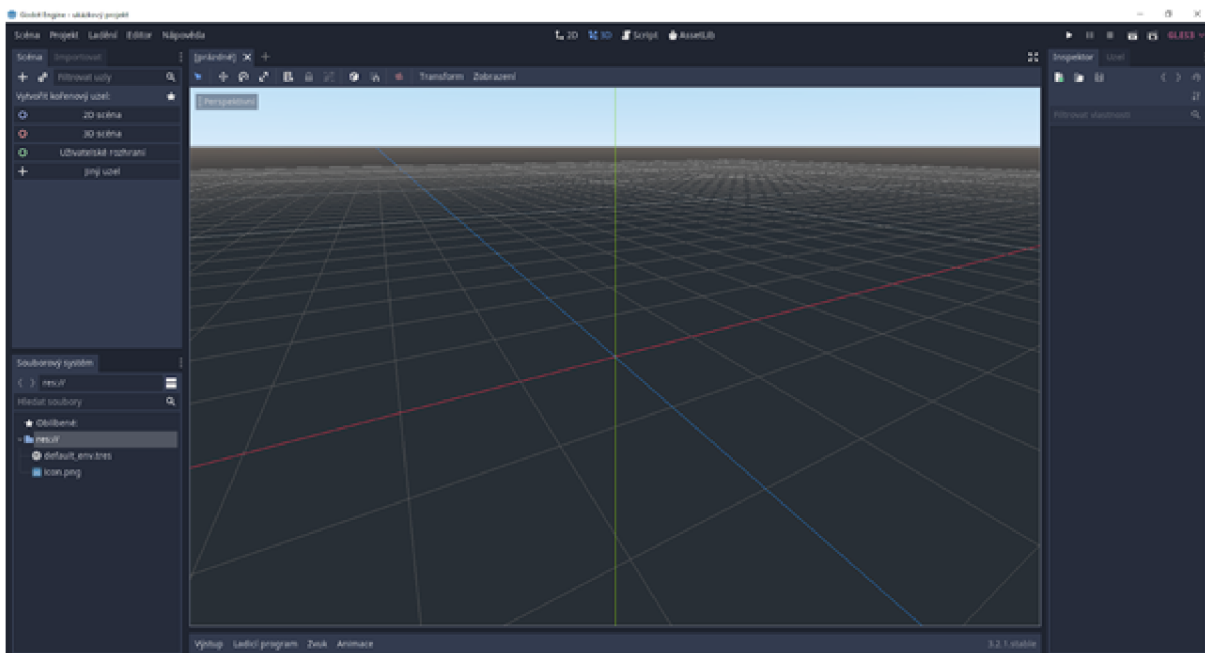
4.3.1. Prostředí

Samotné prostředí Godot engine je velmi podobné Unity. Nalezneme v něm několik jednotlivých oken. Každé okno má vlastní funkcionalitu a využití. Ve vrchní části softwaru nalezneme tři oddělené sekce. Sekcí v levé části je klasické menu s rozbalovacími možnostmi. Nahoře uprostřed je implementováno přepínání scén na 2D a 3D grafiku společně s obsaženými Scripty a AssetLib. Poslední sekci umístěnou v pravé části je část pro samotné spouštění hry a její testování.

V položce menu Scéna jsou obsaženy komponenty podporující tvorbu nového projektu, import a export, a také ukládání jednotlivých scén. Projekt je položka menu, ve které můžeme nastavovat jednotlivé vlastnosti projektu a nástroje k tomu potřebné. Položka Ladění umožňuje nastavení scén, které se následně projeví v ladění a testování projektu. Sekce menu Editor umožní nastavení oken v aplikaci a poslední položka Nápověda obsahuje informace potřebné k seznámení s vývojovým prostředím a samotným Godotem.



Obrázek 4.10: Prostředí Godot, detail horní části



Obrázek 4.11: Prostředí Godot

Hlavní pracovní plochu engineu Godot nalezneme ve střední části softwaru. Z obou stran je obklopeno dalšími okny. Toto okno úzce spolupracuje se středovými tlačítky, kdy na jejich interakci reaguje i obsah okna. V případě zvolení 2D nebo 3D grafiky nalezneme pracovní plochu, do níž se umísťují jednotlivé prvky a assety. Když bude aktivní tlačítko Script, zobrazí se plocha, ve které můžeme aktivně psát zdrojové kódy pro jednotlivé objekty. Tato plocha dominuje obrázku 4.11.

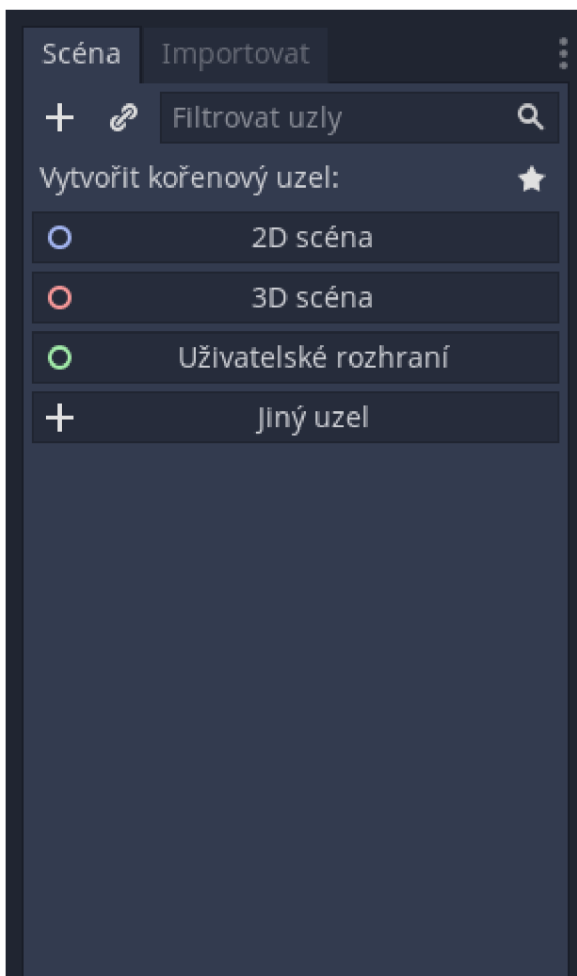
V levé části vývojového engineu nalezneme dvě od sebe oddělená okna. První z nich slouží ke tvorbě scény a editaci jednotlivých objektů scény, obrázek 4.12. Ve spodním okně je patrné členění projektu a struktura složek projektu, obrázek 4.13.

V pravé části engineu je umístěno okno, které slouží k zobrazení a editaci vlastností jednotlivých objektů. Nalezneme zde úpravu velikosti, rotace či fyzikální vlastnosti. Do fyzikálních parametrů můžeme počítat i viditelnost a materiál, kterým je objekt vybaven. Jednou z dalších možností je zobrazení skriptů, které daný objekt obsahuje, a jejich možné spuštění.

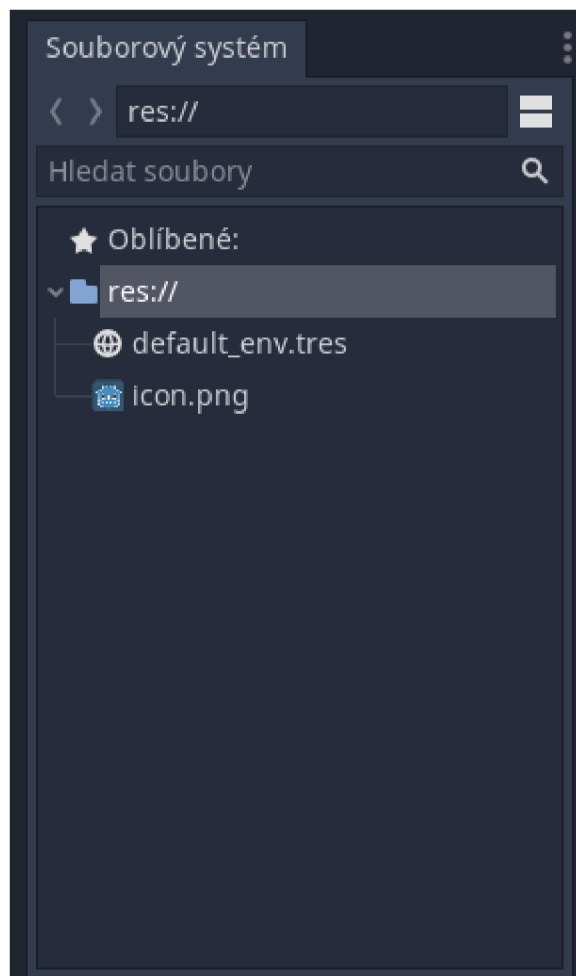
4.3.2. Licence

V roce 2014, kdy byl kód Godotu uvolněn pro veřejnost, byl opatřen licencí MIT. Tato licence vznikla na Massachusettském technologickém institutu. Softwaru, které jsou pod touto licencí uvolněny pro uživatele, je možné použít jako proprietární software. Podmínkou je, že aplikace musí obsahovat text licence pro daný software.

Následně došlo ke zveřejnění pod licencí open source, což přineslo velkou vlnu vývojářů, které tento software zaujal. Je možné také přispívat do kódu samotného softwaru, a pokud je kód přínosný, při dalším updatu je přidán do softwaru. [8]



Obrázek 4.12: Levé horní pracovní okno, editace objektů scény



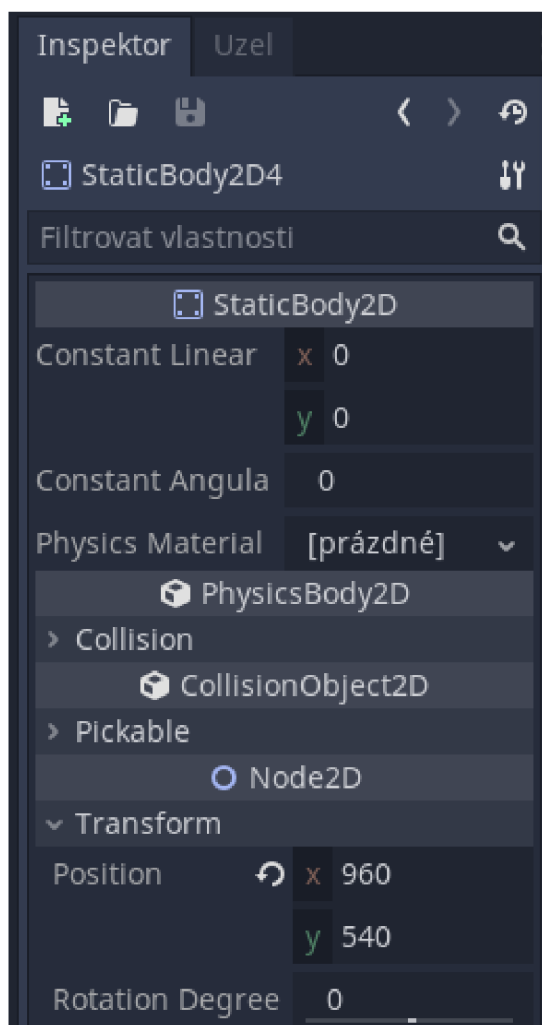
Obrázek 4.13: Levé spodní pracovní okno, správa struktury projektu

4.3.3. Požadavky pro vývojáře

Pro vývoj v prostředí Godot není zapotřebí nijak zvlášť výkonného počítače. Hardware požadavky systému jsou Operační systém Windows 10 nebo macOS 10.10 a vyšší. V případě, že se jedná o operační systém Linux, je potřeba 64 nebo 32 x86 bitové verze. Grafická karta je doporučena jako minimální AMD Radeon HD 7000 nebo NVIDIA GeForce 8 series. Když se jedná o vývoj aplikace s 3D grafikou, doporučuje se minimální grafická karta GeForce GTX 1050 a adekvátní modely z řady AMD. V případě potřeby RAM je zapotřebí 4 GB a pokud se bude jednat o vývoj projektu obsahující 3D grafiku, doporučuje se 8 GB. [8]

4.4. Zhodnocení

Na základě zkušeností s jednotlivými game engine a jejich porovnání jsem se rozhodl k využití game engine Godot. Nejlépe vyhovuje demoverzi hry, kterou se chystám pro ukázkou navrhnout. A to z důvodu, že game engine Unity se hodí spíše pro střední a velké projekty nebo projekty vyžadující podporu 3D grafiky. Jeho velkou předností je však



Obrázek 4.14: Pravé pracovní okno přizpůsobené k zobrazení objektu

velká komunita a Asset store. Další předností může být fakt, že jednotlivé objekty jsou snadněji nastavitelné v kolizní vzdálenosti a lze tak snadno vybalancovat správné kolize s okolními objekty, na které mají reagovat. Tuto možnost nabízí i engine Godot. Cocosharp má v tomto ohledu horší a pracnější vlastnosti a díky těmto důležitým faktorům je vyřazen z výběru implementačních nástrojů.

Pokud hodnocení implementačních nástrojů opřeme o grafické možnosti engineu Godot a Unity, jasně nám vyplyne, že engine Unity je lépe vybaven pro práci s 3D grafikou. Naopak Godot v oblasti 2D grafiky daleko přesahuje vlastnosti Unity. Hardwarové požadavky hrají v tomto případě ve prospěch Godotu, který nepotřebuje tak výkonný hardware.

Komunita a možnosti využití prefabů a různých assetů je velkou výhodou obou engineů. Je však pravdou, že Unity má více možností, které se dají k projektu využít. Nevýhodou však je, že valná většina komponentů je placená. Godot narozdíl od toho nevyniká tak velkými možnostmi, ale téměř všechny assety jsou poskytnuty zdarma případně za symbolické ceny.

Co se týče jazyků, ve kterých lze vyvíjet, jsou na tom oba porovnávané enginey velmi podobně. Jako velkou výhodou u engineu Godot vidím možnost využití vlastního skriptovacího jazyka. Tento jazyk je přizpůsoben a přesně napsán pro potřeby tohoto engineu.

Vzhledem k tomu, že jsem se rozhodl vytvořit návrh 2D hry, jsem se po srovnání implementačních nástrojů rozhodl k využití game engine Godot. Tento engine by měl pokrýt veškeré požadavky, které jsou kladeny na implementační nástroj. Jako programovací jazyk doporučuji využít jeho vlastní programovací skript GDScript, který je pro tyto účely navržen.

5. Analýza způsobů monetizace

Způsobů, jakými lze vydělat na mobilní aplikaci, je hned několik a jejich zapracování do aplikace je nutné zvolit dle cílové skupiny. Efektivita výtěžku jednotlivých aplikací se liší na základě zvoleného způsobu monetizace. Výtěžek u dvou aplikací, které mají totožnou monetizační strategii, nemusí být stejný. Výtěžky se mohou lišit z důvodu, že pro jednu aplikaci mohla být zvolena špatná monetizační strategie, a stažení aplikací zde hraje také důležitou roli. Ke každému způsobu monetizace se přistupuje jednotlivě a vypracovává se k němu i marketingový plán. Populární způsoby, jak monetizovat aplikaci, jsou popsány v následujících podkapitolách. Jak vizuálně může takový obchod v aplikaci vypadat naleznete na obrázku 5.1. [15] [16] [7] [17]

5.1. Zpoplatnění stažení

Placené a neplacené aplikace, i takto lze chápat základní členění aplikací v obchodě Google Play. V tomto členění dokážeme nalézt základní způsob monetizace aplikací, a to zpoplatnění jejich stažení. Tato metoda se volí především u aplikací známých firem sloužících k odborným účelům a poskytujícím nadstandardní služby, které aplikace zdarma nenabízí. Výtěžek aplikace je poté roven ceně aplikace krát počet stažení. [15]

5.2. Reklamy a reklamní systémy

V případě, že máte aplikaci, na které chcete profitovat, ale nechcete zároveň zpoplatnit její stažení, je řada způsobů, jak dosáhnout zisku. Jedním z těchto způsobů je možnost monetizace za pomoci reklamních systémů. V současné době je velká spousta systémů, které lze využít. Jedním z nejpopulárnějších je AdMob od společnosti Google. Reklamy je možné zobrazit v několika typech, jako jsou bannery, vsunuté reklamy nebo reklamy s nabídkou odměny.

Bannery jsou zpravidla obdelníkové reklamy, které jsou umístěné ve vrchní nebo spodní části plochy, a zabírají část obrazovky. Obsah bannerů se jednou za čas automaticky obnoví na aktuálnější. Vsunutá reklama je reklama obsažená přes celou zobrazovací plochu. Nejčastěji se zobrazuje při přestávkách nebo předělech zobrazení jednotlivých aplikací. Může se jednat o reklamu statickou i dynamickou, která často obsahuje krátký videospot. Typ reklamy s nabídkou odměny se od ostatních způsobů liší tím, že uživatel ji zobrazuje cíleně za účelem obdržení nějaké odměny v aplikaci. [15] [28] [1] [7] [17]

5.2.1. Výdělek z reklam

Výtěžek aplikací závisí na počtu zobrazení a kliknutí na reklamy. Ve druhém případě hraje obsah reklamy také velkou roli. Každý obsah je ohodnocen jinou hodnotou. Cena za kliknutí se pohybuje od řádu desetitisícin za zhlédnutí či kliknutí až po desetinné řády, které produkují nejvýdělečnější reklamy. Denní přibližný výtěžek se dá poté spočítat pomocí jednoduchého vzorce, kdy se zohlední počet zobrazených reklam, možných druhů zobrazených reklam a ceny za zobrazenou reklamu. Je uvažován hypotetický výtěžek reklamy 0.10 \$ za reklamu. [3] [26]

$$DV = PCR * \frac{ZR}{RR}$$

DV - denní výdělek, PCR - průměrný výdělek za zobrazenou reklamu, ZR - Zobrazené reklamy, RR - Různost reklam

Aplikace byla stažena přibližně 50 miliony uživatelů, z čehož ji denně používá přibližně 8-10 milionů. Zde vezmeme číslo 10 milionů zhlédnutí. Reklamní systém poté dokáže zobrazit tisíc různých reklam denně, jejichž průměrná hodnota zobrazení nebo rozkliknutí činí 0.10 \$. Po provedení matematických operací dostaneme výsledek výdělku 1000 \$ denně.

$$\frac{10000000}{1000} * 0,1 \$ = 1000 \$ \text{ denně}$$

5.2.2. AdMob

Tato reklamní platforma byla roku 2009 nabídnuta k řešení reklam pro platformy mobilních operačních systémů společností Google. Roku 2013 došlo k přepracování AdMob a stal se z něj AdSense pro mobilní zařízení. Statistiky AdMob uvádějí, že každý měsíc dojde přes jejich služby k zobrazení přes 40 miliard mobilních reklam. Tato skutečnost z něj činí největšího poskytovatele reklam na světě. Finance, které jsou zachyceny prostřednictvím systému AdMob, si vydavatel aplikace může vyzvednout vždy, když výdělek přesáhne hodnotu v přepočtu 2000 korun českých. Skutečnost, že tento reklamní systém je vytvořen a provozován společností Google, zaručuje jeho vysokou bezchybnost a integritu. [2] [3]

5.2.3. Media.net

Media.net patří mezi jednoho z největších konkurentů AdMob. Nabízí jedno z nejkompaktnějších portfolií pro výběr reklamního systému. Jedná se o původního tvůrce reklamního formátu display-to-search, což je reklamní formát, který uživateli nabízí zboží a aplikace vyhledávané někdy v minulosti. Reklamní systém je plně kompatibilní s operačním systémem Android a iOS, což umožňuje vydavateli aplikace zpeněžit aplikace na všech systémech za použití jednoho poskytovatele reklam. Media.net umožňuje výplatu až po dosažení určité hranice výdělku, také nabízí pro nově příchozí nárůst 10% jejich výdělku po dobu prvních tří měsíců. [20]

5.2.4. Platby a nákupy v aplikaci

V případě, že se rozhodneme použít tento model, je potřeba, aby byla dobře vymyšlená marketingová strategie a způsoby monetizace. Dobře zvolená monetizační strategie zajistí vyšší příjmy než příjmy, kterých by bylo dosaženo nerozvážným jednáním. Jako poplatky v aplikaci lze chápat velkou spoustu prvků a služeb. [15]

5.2.5. Nákup herní měny

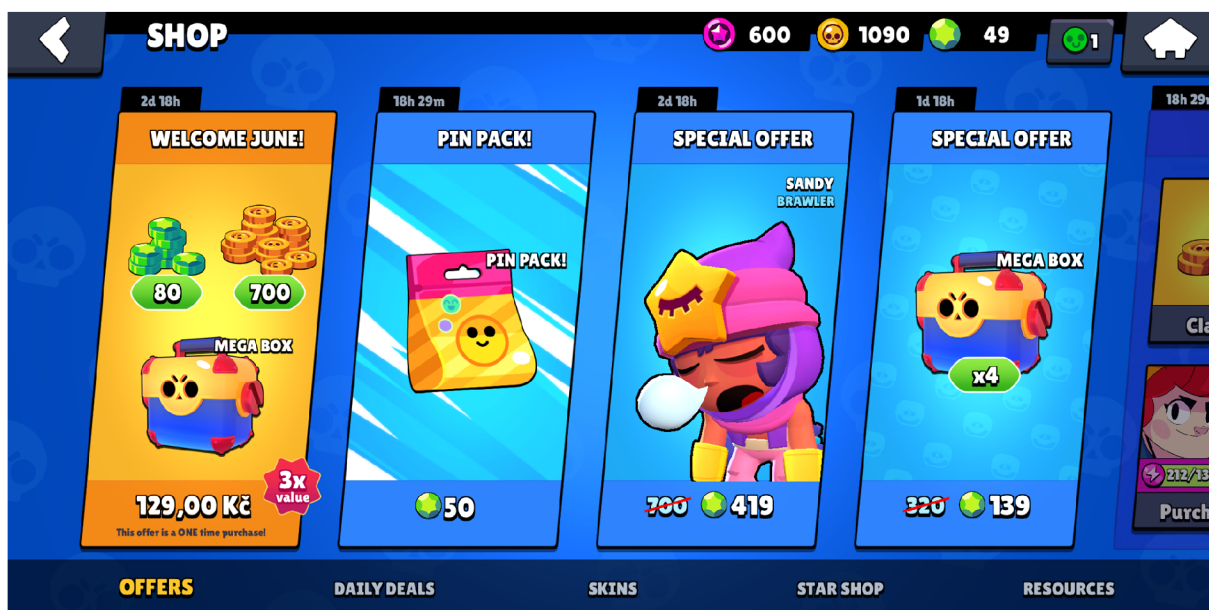
Jedním z poplatků v aplikaci může být nákup herní měny. Jedná se spíše o metodu, která je využívána v herním průmyslu. Při využití této metody se uživateli aplikace podbízí k zakoupení herní měny běžně používané ve hře na běžně dosažitelné herní věci nebo speciální herní měny, za kterou je možné koupit běžné věci za malou cenu nebo speciální věci dostupné jen za speciální měnu. [15]

5.2.6. Prémiový účet

Prémiový účet je další možností monetizace aplikace. Znamená to, že za měsíční poplatek uživatel dostává speciální věci a funkce, které jsou bez tohoto účtu uzamčeny. Jedná se o jeden z nejpopulárnějších monetizačních modelů, který lze využít jak u aplikace, tak ve hře. Jeho předností a zároveň slabinou je obsah poskytovaných služeb za cenu prémiového účtu. Dalším tahákem k porízení prémiového účtu může být i zbavení se reklamního obsahu, který jinak aplikace obsahuje. Jednou z nejčastějších cen za prémiový účet je 4.99 \$ a 9.99 \$ za měsíc. [15]

5.2.7. Poplatky v aplikaci

Dalším velmi populárním způsobem výtěku peněz z aplikace je poskytnutí věcí za poplatek. Tato možnost se dá postavit nad několika možnostmi. Jednou z nich je poplatek za odstranění reklam v aplikaci. Tato služba je velmi populární v aplikacích, které uživatel používá denně nebo je potřebuje k práci. Stejný druh aplikací je často dostupný jen v určité demo verzi a za přístup k plné verzi je požadován jednorázový poplatek. Velkou popularitou v aplikacích je i takzvaný inapp nebo ingame obchod. V těchto obchodech se nachází doplňkové služby pro dané aplikace. Může se jednat o předměty, měnu či rozšiřitelný obsah. [15]



Obrázek 5.1: Vlastní zpracování ukázky obchodu ze hry Brawl Stars

6. Analýza uživatelů

V této podkapitole se budeme zabývat průzkumem uživatelů a určíme cílovou skupinu. Pro tento účel jsem vytvořil dotazník, díky kterému jsem analyzoval chování a reakce uživatelů na reklamní systémy, poplatky nebo čas, který na mobilním zařízení tráví. Charakteristika uživatele a volba strategie pro monetizaci by díky tomuto dotazníku mohla být snadnější a lépe přizpůsobená. Byly také vytvořeny fiktivní profily lidí z cílové skupiny, která vyplynula z dotazníku.

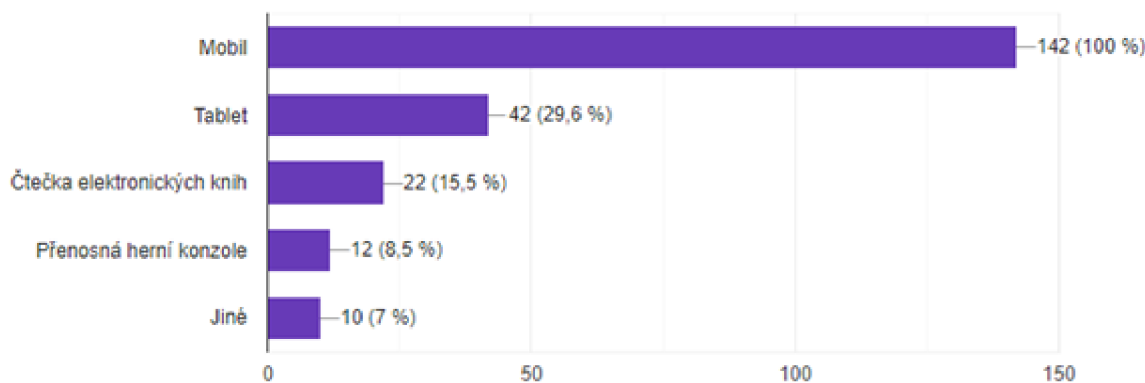
6.1. Průzkum uživatelů

Pro získání informací o uživateli byl vytvořen anonymní dotazník. Celkem jej zodpovědělo 142 účastníků. Počet dotázaných mužů byl jen o necelé 3 % vyšší než žen. Nejpočetnější věkovou kategorií dotázaných byl věk 19-25 let s procentuálním zastoupením 54,2 %. 51 % dotázaných byli studenti, zbylí dotázaní uživatelé byli v pracovním poměru, OSVČ nebo na mateřské dovolené a jeden dotázaný byl dokonce v důchodu. Podíl lidí pracujících v IT sektoru byl 13,4 %. U dotázaných pracovníků v IT sektoru jsem očekával odpovědi, které se více přikláněly na stranu bezpečnostních opatření, což se i potvrdilo.

Na otázku: „Jaké mobilní zařízení vlastníte?“ 100 % dotázaných uvedlo mobilní telefon a 42 z celkového počtu dotazovaných disponuje i tabletem. Následné další vlastnictví mobilních zařízení je uvedeno v grafu 6.1.

Jaké mobilní zařízení vlastníte?

142 odpovědí



Obrázek 6.1: Graf vlastněných zařízení

Z dotazníku bylo dále zjištěno, že 69 % dotázaných disponuje operačním systémem Android. Apple IOS upřednostňuje u mobilních zařízení 21,1 % dotázaných a 7,7 % dotázaných vlastní více mobilních zařízení a kombinují u nich systém Android i IOS. Zbylá procenta vlastní mobilní zařízení s jiným operačním systémem.

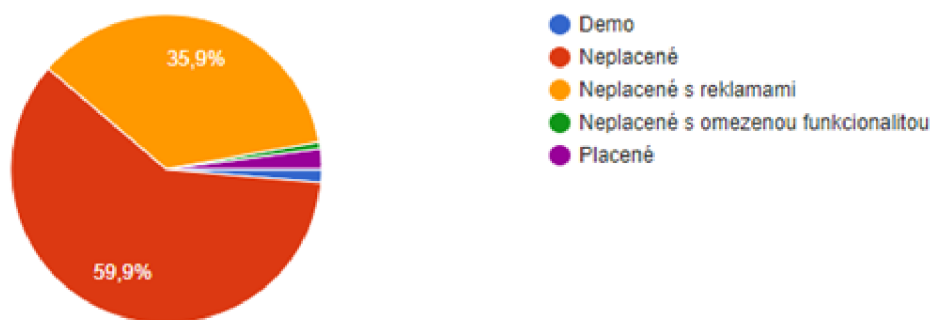
Z otázky: „Kolik času na mobilních zařízení trávíte?“ bylo zjištěno, že 93 % dotázaných na mobilních zařízení tráví více než jednu hodinu denně. Z těchto 93 % tráví na mobilních zařízení jednu až tři hodiny denně 62 %, čtyři až šest hodin denně 23,9 % a zbylých 7 % více než šest hodin denně. Na typ činností, na které dotázaní využívají své mobilní zařízení, byla vytvořena otázka, na kterou měli možnost odpovědět více možnostmi a odpovědi

jsou následující. 137 dotázaných tráví čas na mobilních zařízeních především komunikací s lidmi, 76 dotázaných využívá tato zařízení k práci a téměř všichni je využívají k zábavě jako jsou hudba, filmy a podobné se zastoupením 95 dotázaných. Hraní her na mobilních zařízeních provozuje 53 dotázaných.

Otázka spojená s nejčastěji stahovanými aplikacemi pak hovoří v jasný prospěch bezplatných aplikací nebo bezplatných aplikací obsahujících reklamy, a to se zastoupením 95,8 % dotázaných. Zbýlá procenta z této otázky se rozprostřela do aplikací placených, neplacených s omezenou funkcionalitou a aplikací poskytujících zdarma pouze demo verzi viz graf 6.2.

Jaké aplikace nejčastěji stahujete?

142 odpovědí

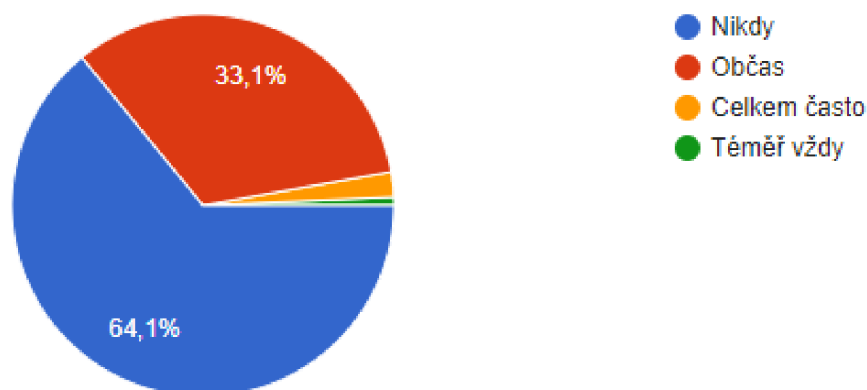


Obrázek 6.2: Graf nejčastěji stahovaných aplikací

Obvyklost nákupu mobilních aplikací můžeme vidět na grafu 6.3, ze kterého vyplývá, že placené aplikace nejsou u dotázaných ve velké oblibě.

Jak často kupujete mobilní aplikace?

142 odpovědí



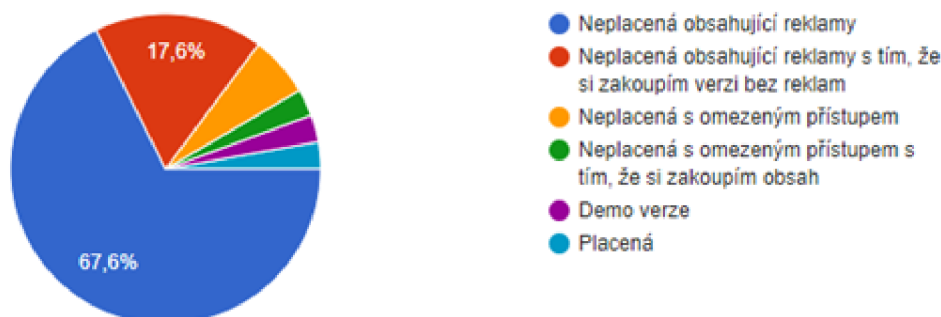
Obrázek 6.3: Graf, jak často kupují dotázaní aplikace

Nadpoloviční většina (79 dotázaných) využívá při hraní her speciálních bonusů ve hře, dostupných za zhlédnutí reklamy. Opakem je ovšem pořízení si doplňků ve hře za reálné

peníze. V tomto případě 104 dotázaných si nikdy nepořídili žádnou věc tímto způsobem. Výsledky na otázku: „Raději upřednostním aplikaci, která je:“ naleznete na obrázku 6.4.

Raději upřednostním aplikaci, která je:

142 odpovědí

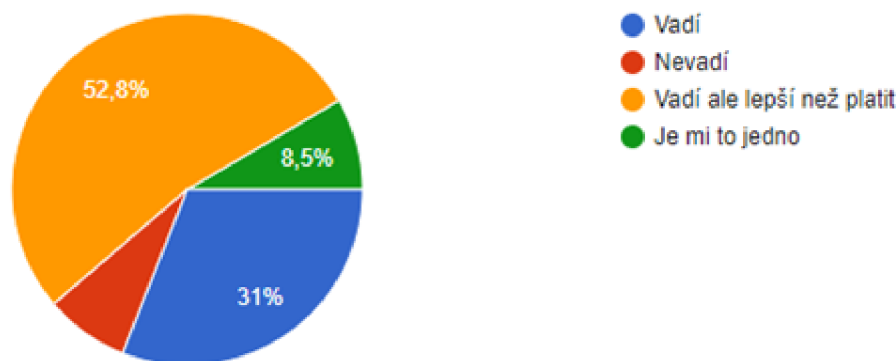


Obrázek 6.4: Graf upřednostněných aplikací

Jedním z nejpodstatnějších výsledků průzkumu je odpověď na otázku, jestli dotázaným vadí reklamy v aplikaci. Díky této otázce zjistíme, že 119 dotázaným reklamy v aplikaci vadí, 75 dotázaných ale raději bude mít v aplikaci reklamy, než aby za ni platili.

Reklamy v aplikaci mi:

142 odpovědí

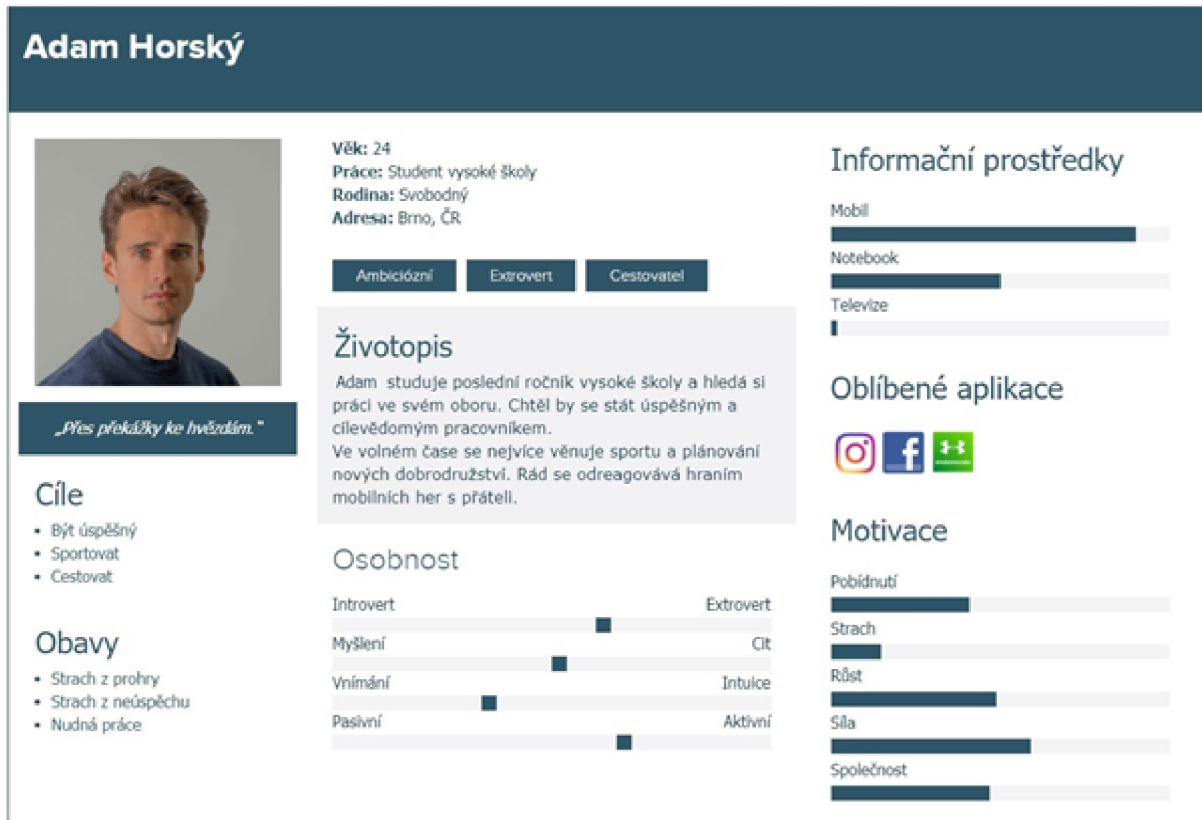


Obrázek 6.5: Graf reakce dotázaných na reklamy v aplikaci

6.2. Typický uživatel

Na základě upřesnění představy o typickém uživateli bylo navrženo několik person. Jedna z nich je obsahem této kapitoly a jedná se o obrázky 6.6, druhou personu 10.1 poté naleznete v přílohách. Typickým uživatelem navržené aplikace je jedinec ve věkovém rozmezí 19 až 35 let, je studentem nebo pracujícím a disponuje mobilem s operačním systémem Android. Nejčastěji stahuje aplikace, které jsou neplacené, a k nákupu aplikace se přiklání jen velmi zřídka.

Aplikace si vybírá podle recenzí, doporučení a stahuje, co se mu líbí. Nejčastěji své mobilní zařízení využívá k zábavě, práci a komunikaci s lidmi. Mobilní hry hraje několikrát týdně nebo i pravidelně každý den. Využívá speciálních bonusů ve hře, které získá za zhlédnutí reklamy, a doplňky ve hře si kupuje pouze výjimečně. Reklamy mu sice v aplikaci vadí, ale je ochoten je strpět výměnou za bezplatnost aplikace.



Obrázek 6.6: První persona

7. Návrh a popis aplikace

V této kapitole se budeme zabývat návrhem aplikace a jejím popisem. Bude následovat mockup několika základních obrazovek. Návrh implementace aplikace, její možnosti monetizace a způsoby jejího testování budou součástí.

7.1. Popis aplikace

Jako navrhovanou aplikaci jsem se rozhodl vytvořit mobilní hru s názvem Cat's infinity space. Jedná se o skákačí hru s příběhem zasazeným do vesmíru. Jejím hlavním aktérem je kocour Gray, který postupně prochází jednotlivé levely hry a sbírá vesmírné kuličky a ostatní objekty. Každá kulička mu postupně nabíjí jeho kouzelný obojek, jenž mu umožní stát se plazma kuličkou, která může v propojujících levelech a v pozdějších fázích hry procházet úzkými prostory a v některých případech i clonami. Clony neumožňují projít Grayovi v jeho kočičí formě. Pro postup do dalšího levelu se Grayovi po sesbírání všech objektů otevře portál, kudy se dostane do další části vesmíru. Aby hra nebyla tak jednoduchá, po cestě se Gray setká s velkým množstvím nástrah a nepřátel, ať už se jedná o nástrahy, které je možné jen přeskočit, až po nepřátele, které může překonat jen za pomoci kouzelného obojku. Celá hra je zasazena do příběhu záchrany před zánikem galaxie.

7.2. Návrh aplikace

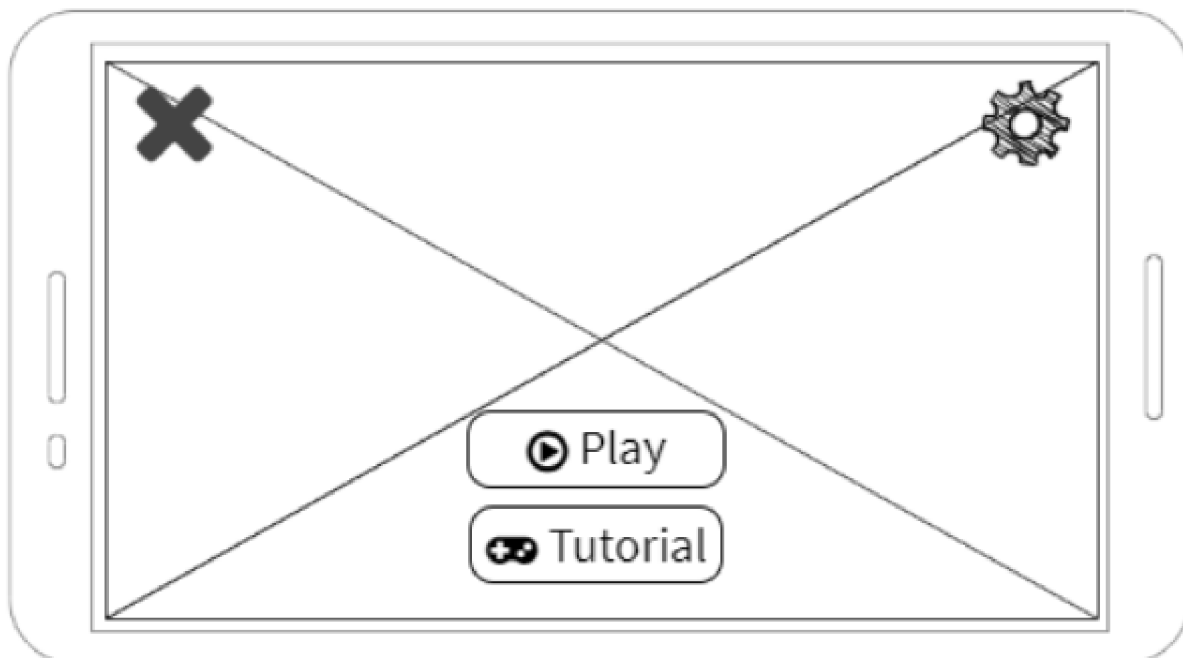
Při návrhu aplikace bylo využito poznatků z literatury, které byly zasazeny do zkušeností z praxe. Návrh aplikace je oproštěn o grafické prvky a barevnost, vychází ze základních elementů a do barevnosti a konečného vzhledu je dotvořen až v mockupu. Základní vzhled aplikace a návrh jednotlivých obrazovek byl realizován podle konceptu hry. Jsou zde využity pouze základní prvky hry a pouze jednoduché grafické prvky, které nijak přesně nespécifikují, jakým způsobem by měla grafika hry vypadat. V této fázi projektu by se měly rozvrhnout jednotlivé důležité obrazovky. Neznamena to, že v této části bude specifikována každá jednotlivá úroveň hry, ale budou zde vyzdviženy základní zobrazení jednotlivých obrazovek. Z návrhu aplikace by měly být vytvořeny základní materiály pro vytvoření mockupu.

Na návrhovém obrázku 10.2 je úvodní obrazovka hry. Je zde patrné, že hlavním prvkem této návrhové obrazovky je název aplikace společně s načítacím prvkem, který zobrazuje připravenost a načtenost hry, jež je ve fázi spouštění. Po načtení lišty do plné šíře, tedy kompletního zaplnění, je hra přepnuta na další obrazovku 7.1. Na této obrazovce je patrné menu hry, nalezneme zde pozadí hry, které v návrhu není graficky zpracováno, a v něm umístěná dvě hlavní tlačítka uprostřed obrazovky. Tato tlačítka plní základní funkcionality pro zahájení hry nebo spuštění tutorialu. V horní části se nachází další dvě tlačítka pro nastavení a ukončení hry.

V případě aktivace tlačítka Nastavení se zobrazí krátké nastavení hry, které zahrnuje připojení do služeb Google Play a ostatních podporovaných systémů třetí strany. Dále nastavení umožňuje manipulaci s nastavením zvuku, vibrací či notifikacemi. Návrh vzhledu nalezneme na obrázku 10.3.

Při stisku tlačítka Play se aplikace přepne do nové obrazovky pro výběr úrovně. Úrovně jsou odemčeny vždy při dokončení předchozí úrovně nebo splnění zadaného úkolu. Zá-

kladní rozvržení nalezneme v obrázku 10.4. Poslední návrhovou obrazovkou s označením 10.5 je koncept a proporce hry samotné. V této části návrhu by měla být specifikována jedna úroveň hry, ve které jsou proporcčně rozmístěny jednotlivé prvky. Návrhová obrazovka také může obsahovat návrh ovládání pro uživatele samotné hry. Ke konkrétní specifikaci a proporcím dochází až v mockupu aplikace.



Obrázek 7.1: Ukázka návrhu obrazovky menu

7.3. Mockup aplikace

Při kompletaci mockupu¹ se striktně vychází z návrhu aplikace. Využívá se zde objektů, které se ve hře budou používat. Tyto objekty již mají téměř dokončený grafický design a jejich velikosti odpovídají velikosti v aplikaci. V mockupu se dále nachází i jednotlivé ovládací prvky aplikace. Těmito prvky se rozumí tlačítka pohybu, skoku a použití speciální schopnosti. V mockupu nemusíme najít žádný konkrétní level, ale pouze ukázkový, který se používá pro objasnění velikosti prvků a pro ukázkou jejich použití.

Na mockupovém obrázku 10.6 je vidět počáteční obrazovka. Je zde již konečný design a název úvodní obrazovky ze hry. Tato obrazovka má za úkol udržet hráče v zájmu hraní hry, dokud nedojde k načtení všech potřebných věcí. Po uplynutí načítací obrazovky se dostaneme do menu, které se oproti návrhu změnilo a obsahuje tlačítka Play a Shop, viz obrázek 10.7.

V případě, že zvolíme tlačítka Play, dostaneme se do výběru úrovně (obrázek 10.8), kde můžeme vidět jednotlivé úrovně, které jsou přístupné. Na nepřístupných úrovních je zámek. Po výběru úrovně se dostaneme do zvolené úrovně, kde nalezneme několik prvků, jak je vidět na obrázku 7.2. V dolní části obrazovky jsou umístěny ovládací prvky aplikace.

¹Model, který má stejnou velikost jako skutečná věc.

V pravé dolní části je ovládací prvek na skok a použití superschopnosti. V případě, že dojde ke smrti kocoura Gray, hráči se zobrazí obrazovka 10.9. Tato obrazovka odpočítává určitou dobu, po kterou je možné, aby se hráč oživil a navrátil do hry za zhlédnutí reklamy. Pokud tak neudělá, po uplynutí času se hra ukončí a hráč se vrátí zpět do výběru úrovně hry.



Obrázek 7.2: Mockup konkrétní úrovně hry

7.4. Návrh implementace a testování

Pro návrh implementace jsem se rozhodl vypracovat přibližný časový plán realizace projektu prostřednictvím metody PERT. Pro odhad časových údajů tvorby bylo zohledněno několik faktorů, jež v sobě zahrnují osobní a odborné zkušenosti a doporučení, které jsou obsaženy v odborné literatuře, která byla využita pro tvorbu této práce.

Implementace by měla být tvořena dílčími úkoly, které se následně rozpadají do podúkolů. K větší přehlednosti zde budu rozebírat pouze dílčí úkoly postupu tvorby navrhované aplikace. Na začátku každého projektu by měla proběhnout analýza, zda je projekt realizovatelný a na jaké dílčí kroky se projekt následně rozpadne. Časová náročnost analýzy se liší projekt od projektu, v případě navrhované hry mi tato část zabrala 16 hodin. Po zhodnocení analýzy a kladnému vyjádření k realizaci projektu se přistupuje k založení samotného projektu a prostoru v Git aplikaci². Časový rámec potřebný pro tyto věci se pohybuje v řádu hodin, v našem případě se jedná o dobu 5 hodin. Následně se pokročí k návrhu samotné aplikace. Návrh aplikace byl popsán v sekci 7.2 a zabral 30 hodin práce. Pro přesné rozměry věcí v aplikaci se poté využije mockup, tato část je popsána v kapitole 7.3. Pro vypracování mockupu bylo poté zapotřebí 48 hodin práce. Pro další členění dílčích úkolů a doby jejich trvání bylo využito metody PERT, protože se jednotlivé úkoly větví a provádí se paralelně. V části, která je demonstrována pomocí metody PERT, dochází k prvnímu rozpadu na část grafickou a programovou. V grafické části dochází

²Git aplikace slouží pro správu verzí vytvářené aplikace.

k tvorbě pozadí, jednotlivých objektů, hlavní postavy, nepřátel a to vše nezávisle na sobě. V programové části se začínají kompletovat základní funkce nutné pro pohyb postavy a nepřátel. Po zhotovení všech potřebných komponent se jednotlivé části aplikace začnou spojovat, vznikají tak objekty s funkcemi a dochází ke tvorbě kolizí. Po otestování správnosti kolizního systému se grafická a programová část aplikace znovu větví. Větev úkolů určených pro grafickou práci se začne zabývat návrhem a realizací jednotlivých obrazovek aplikace, jako je Menu, Shop či Nastavení. Dochází také k tvorbě a úpravě základní úrovně hry. Úkoly určené pro programování jsou poté doladěny, následně testovány urovně hry a příprava podpůrných funkcí pro jednotlivé obrazovky. Celkovou pracnost, která vyplynula z metody PERT, naleznete popsanou o odstavec níže. Po celou dobu tvorby aplikace docházelo k průběžnému testování a čas testování je zahrnut do časové náročnosti jednotlivých úkolů. Po ucelení aplikace a dokončení správnosti přechodů mezi obrazovkami se může přistoupit k vydání verze na Google Play, toto vydání by mělo zabrat přibližně 4 hodiny.

Z metody PERT vyplynul diagram, který nalezneme na obrázku 10.10, z něj jasně vyplývá, že aplikace by nám měla trvat přibližně 118 hodin práce. Kritická cesta, na které je největší pravděpodobnost zpoždění vychází přes uzly 1, 5, 7, 10, 14 a 16. Činnosti na této cestě je dobré mít pod drobnohledem, aby se dalo co nejrychleji zareagovat na případný problém vzniklý na kritické cestě. Popis činností a hodnotu přechodů poté naleznete v tabulce 10.1 a 10.2 umístěné v přílohách. Po sečtení hodin, které byly vynaloženy na tvorbu plánovaného projektu se dostaneme na 221 hodin. Tento čas by nám měla zabrat prvotní verze aplikace, kterou lze publikovat na obchodu Play.

Při programování jednotlivých funkcí se využívá programovací jazyk GDScript. Tento jazyk se řadí mezi jazyky skriptovací. Mělo by se postupovat po malých částech, program se nejdříve přizpůsobuje pro základní grafické prvky, které jsou vytvořeny pro základní programovací funkce. Pod těmito prvky si můžeme představit jednoduché objekty, jako je třeba obdélník, který nám bez textur a jen s jednoduchým designem plní úkol plošiny, po které se dá pohybovat. Mezi první funkce, které se budou implementovat, budou určité patřit funkce pohybové. Následně budou implementovány základní kolizní funkce, které budou podrobně dokončeny až ve chvíli, kdy bude dokončen zbytek grafických prvků. Poté budou implementovány funkce pro správné interakce s předměty a ukončení hry. V poslední fázi by mělo dojít k implementaci funkcionality superschopnosti.

Následná část implementace by měla být namířena na design jednotlivých úrovní a přizpůsobení pro hráče. Mělo by se vycházet z jednoduchých až téměř primitivních úkolů a překážek ve hře až po složité a komplexní požadavky vytvářeny na hráče.

Testování aplikace probíhá po celou dobu její implementace. Pokud některá z funkcí vykazuje činnost, která pro ni nebyla určena, mělo by dojít k jejich podrobnému přezkoumání, rozpadnutí na nejmenší elementy a případnému přepsání do požadované podoby.

Po přetestování všech funkcí ve hře by mělo dojít k testování celé aplikace. Nejdříve by aplikaci měli testovat sami vývojáři, aby v případě objevení fatální chyby mohlo dojít k jejímu rychlému odstranění. Následně by mělo dojít k testování na uživatelích v menší míře. V tomto testování se odhalují chyby, které většinou vychází ze špatného designu jednotlivých úrovní. Po testování uzavřené skupiny by mělo dojít k velkému množství lidí tím, že se aplikace zveřejní v obchodu Play a zpřístupní se široké veřejnosti. Chyby, které objeví uživatelé aplikace v této fázi, se poté reportují službou Google nebo funkcí vytvořenou v aplikaci. Chyby se následně opraví až po vydání aktualizace aplikace.

```

const Gravity = 100
const MovementSpeed = 500
const JumpUp = 1300

func Move():
>| # pusobeni gravitacni sily
>| self.Motion.y += Gravity
>|
>| var newX = 0
>| if(Input.is_action_pressed("button_left")):>|>|
>| >| newX -= self.MovementSpeed
>| if(Input.is_action_pressed("button_right")):
>| >| newX += self.MovementSpeed
>| self.Motion.x = newX
>|
>| if(Input.is_action_pressed("button_up") and .on_ground()):
>| >| self.Motion.y = -self.JumpUp
>|
>| self.Motion = .slide(self.Motion, self._FloorDirection)
>| pass

```

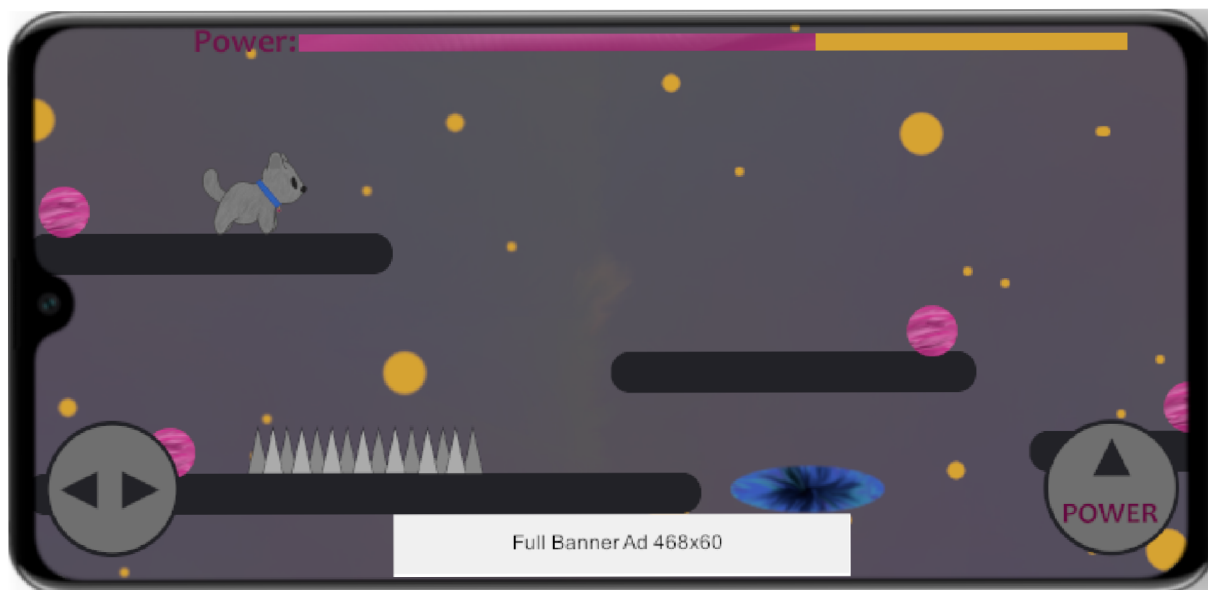
Obrázek 7.3: Ukázka GDScriptu a funkce Move() určené pro pohyb

7.5. Návrh monetizace

Z dotazníku vyplynulo, že způsob monetizace by měl být skrze reklamy. Proto bude v aplikaci umístěno několik možností, jak reklamy využít. Jednou z nich bude umístění banneru s reklamami o rozměrech 468x60 pixelů ve výběru úrovně hry a ve hře samotné, jak lze vidět na obrázku 7.4. Dalším způsobem bude chvíle, kdy hráči z nějakého důvodu skončí hra. Bude mu automaticky nabídnuta možnost oživit se za zhlédnutí reklamy a hrát dále. Tato možnost se nabídne vždy jen jednou za danou úroveň, aby hra nebyla tak jednoduchá. Jako reklamní systém, který bude v aplikaci implementován, bude využito systému AdMob od společnosti Google.

Součástí hry bude i speciální měna, za kterou si hráč bude moci koupit doplňující balíčky do hry. Tato měna půjde v omezené míře získat i ve hře, ale hlavním způsobem jejího získání bude možnost koupit si ji. Cenové rozmezí hry bude stanoveno v různém rozsahu od základních balíčků obsahujících menší množství měny až po balíčky obsahující velké množství. Cena měny by mohla být předběžně stanovena 2:1, to znamená, že za 2 koruny získáte jednu měnu ve hře. Přepočty do jiných měn jsou samozřejmostí. Základní množství měny bude poté navýšeno o určité procento měny obsažené v zakoupené balíčku.

Čím více měny zakoupený balíček obsahuje, tím větší je i bonusové procento získané měny. Poslední možností monetizace, kterou v základní fázi bude hra obsahovat, bude možnost koupit si verzi bez reklam. Tlačítko pro koupi verzi bez reklam bude umístěno hned na hlavní stránce, aby bylo jasně viditelné.



Obrázek 7.4: Ukázka umístění banneru ve hře

8. Závěr

Cílem této práce bylo vytvořit návrh aplikace pro Android se strategií její monetizace. Tento návrh obsahuje seznámení s Androidem jako operačním systémem, možnosti, jaké Android nabízí pro vývojáře, a na čem stojí funkcionalita aplikace. Aby bylo možné vybrat velké spektrum možností strategie pro monetizaci, bylo po zvážení možnosti výběru aplikací rozhodnuto, že bude vytvořena mobilní hra. Součástí práce je také přiblížení důležitosti výběru barev a vnímání jejich významu pro vývoj aplikace a uživatelského rozhraní, jehož popis je také součástí práce.

Po seznámení s Androidem a životním cyklem aplikací došlo ke srovnání možných implementačních nástrojů k vývoji mobilní hry. Všechny tyto nástroje jsou přizpůsobeny pro vývoj mobilních her. Jedná se o nástroje Unity, CocosSharp a Godot. Došlo ke srovnání několika základních parametrů jako jsou základní možnosti vývoje v těchto nástrojích, uživatelské rozhraní, požadavky pro vývoj či licenční podmínky. Z následného zhodnocení implementačních nástrojů byl vybrán implementační nástroj Godot a jako programovací jazyk jeho vlastní skriptovací jazyk GDScript.

Po provedení analýzy implementačních nástrojů došlo k vypracování možností monetizace aplikace. Byly zde zahrnuty aktuální a populární způsoby monetizace mobilních aplikací v současné době. Byly zde také srovnány dva nejpobulárnější reklamní systémy současnosti AdMob od společnosti Google a Media.net od stejnojmenné společnosti. V této části byl i ukázkový příklad výdělku aplikace za předem specifikovaných podmínek.

Pro správný návrh aplikace a správně zvolený způsob monetizace byl vypracován dotazník pro veřejnost. Z nashromážděných údajů byly vypracovány dvě osoby typických uživatelů a došlo k výběru strategie monetizace, která je popsána v poslední části práce společně s návrhem a mockupem aplikace, způsobem implementace a s tím spojeným testováním. Pro představu časové pracnosti tvorby takové aplikace byl vypracován časový odhad jednotlivých dílčích činností a pomocí metody PERT vizuálně znázorněna důležitá část, kde se jednotlivé dílčí úkoly plní po určitou dobu nezávisle na sobě.

Práce by měla v budoucnu sloužit jako počáteční dokument pro tvorbu mobilních aplikací, a to hlavně vývojářům, kteří s vývojem pro mobilních aplikace nemají velké zkušenosti. Také by měla informovat o základních stavebních kamenech každé aplikace, jak správně postupovat při jejím návrhu a mockupu. Po dokončení implementace a důkladném otestování by práce měla vysvětlovat základní možnosti monetizace, které v současné době jsou pro mobilní aplikace populární.

Literatura

- [1] Ad Networks *Singular releases ROI Index for top mobile ad networks* [online]. [cit. 2020-05-03]. Dostupné z: <https://www.businessofapps.com/news/singular-releases-roi-index-for-top-mobile-ad-networks/>
- [2] AdMob *Earn more with your apps*. [online]. [cit. 2020-05-03]. Dostupné z: <https://admob.google.com/home/>
- [3] AdMob works *How AdMob works* [online]. [cit. 2020-05-03]. Dostupné z: <https://support.google.com/admob/answer/7356092?hl=en>
- [4] Android One *Program Android One* [online]. [cit. 2020-05-03]. Dostupné z: <https://www.android.com/one/>
- [5] Cocos2d-x *Cocos2d-x informations* [online]. [cit. 2020-05-03]. Dostupné z: <https://www.cocos.com/en/>
- [6] Dactyl Group. *Definice aplikace* [online]. [cit. 2020-05-03]. Dostupné z: <https://www.dactylgroup.com/cs>
- [7] Genadinik, A.: *MobileApp Marketing and Monetization*. CreateSpace Independent Publishing Platform, 2014. 149 p. ISBN 978-1502383822.
- [8] Godot *Godot informations* [online]. [cit. 2020-05-03]. Dostupné z: <https://godotengine.org/>
- [9] Google Developers. *Activity lifecycle* [online]. [cit. 2020-05-03]. Dostupné z: <https://developer.android.com/guide/components/activities/activity-lifecycle>
- [10] Google Developers. *Platform Android* [online]. [cit. 2020-05-03]. Dostupné z: <https://developer.android.com/about>
- [11] Google Developers. *Platform Architecture* [online]. Poslední změna 2019-12-27 [cit. 2020-05-03]. Dostupné z: <https://developer.android.com/guide/platform>
- [12] Google Developers. *Power management* [online]. Poslední změna 2020-02-27 [cit. 2020-05-03]. Dostupné z: <https://developer.android.com/about/versions/pie/power>
- [13] Interaction Design Foundation *UE and UX Design* [online]. [cit. 2020-05-03]. Dostupné z: <https://www.interaction-design.org/literature/topics/ux-design>
- [14] Jana Dannhoferová: *Velká kniha barev: Kompletní průvodce pro grafiky, fotografy a designéry*. Computer Press, 2015. 352 p. ISBN 978-80-251-3785-7.
- [15] Jeffrey Hughes: *Android Apps Secrets to Selling Your Android App Marketing*. Greg Wiegand, 2015. 465 p. ISBN 978-0-7897-4633-7.
- [16] John Horton: *Android Game Programming by Example*. Packt Publishing Ltd., 2015. 352 p. ISBN 978-1-78528-012-2.

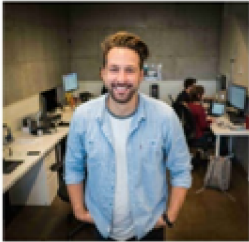
- [17] Laudon, K. C. and C. G. Traver: *E-Commerce 2016: Business, Technology, Society. 12th ed.*. New Jersey: Pearson, 2016. 912 p. ISBN 978-0-133-93895-1.
- [18] Luboslav Lacko: *Mistrouství - Android*. Albatros Media a.s., 2017. 648 p. ISBN 978-80-251-4875-4.
- [19] Luboslav Lacko: *Vývoj aplikací pro Android*. Albatros Media a.s., 2015. 472 p. ISBN 978-80-251-4347-6.
- [20] Media.net *Media.net* [online]. [cit. 2020-05-03]. Dostupné z: <https://www.media.net/>
- [21] Michelle Menard and Bryan Wagstaff: *Game Development with Unity*. Stacy L. Hiquet, 2015. 465 p. ISBN 978-1-305-11054-0.
- [22] Mobile app *Definice mobilní aplikace* [online]. [cit. 2020-05-03]. Dostupné z: https://en.wikipedia.org/wiki/Mobile_app
- [23] Mobile game *Definice mobilní hry* [online]. [cit. 2020-05-03]. Dostupné z: https://en.wikipedia.org/wiki/Mobile_game
- [24] Operating system Android *Informace o systému Android* [online]. [cit. 2020-05-03]. Dostupné z: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [25] Portales, R.: *Mastering Android Game Development. 1st ed.*. Birmingham: Packt Publishing Ltd., 2015. 498 p. ISBN 978-1-78355-177-4.
- [26] Stackoverflow *Does AdMob pay for impressions or only for clicks?* [online]. [cit. 2020-05-03]. Dostupné z: <https://stackoverflow.com/questions/31947763/does-admob-pay-for-impressions-or-only-for-clicks>
- [27] Světelný rozklad *Obrázek světelného rozkladu barev* [online]. [cit. 2020-05-03]. Dostupné z: https://i2.wp.com/krasajachtingu.cz/wp-content/uploads/2012/09/rom_svetlo_1_02.gif
- [28] Top 9 Ad Networks *Top 9 Mobile Ad Networks for Publishers* [online]. [cit. 2020-05-03]. Dostupné z: <https://www.adpushup.com/blog/top-8-best-paying-mobile-ad-networks-you-should-try/>
- [29] Twitter oznámení. *Potvrzení 2,5 miliardy aktivních zařízení* [online]. [cit. 2020-05-03]. Dostupné z: https://twitter.com/Android/status/1125822326183014401?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Cwterm%5E1125822326183014401&ref_url=https%3A%2F%2Fwww.androidpolice.com%2F2019%2F05%2F07%2Fthere-are-now-more-than-2-5-billion-active-android-devices%2F
- [30] Unity *Unity informations* [online]. [cit. 2020-05-03]. Dostupné z: <https://unity.com/>
- [31] Zastoupení jednotlivých verzí systému Android *Operating System Share by Version* [online]. [cit. 2020-05-03]. Dostupné z: <https://netmarketshare.com/>

9. Seznam použitých zkratek a symbolů

| | |
|-------------|--|
| UI | User Interface - Uživatelské rozhraní |
| UX | User Experience - Uživatelská zkušenost |
| GUI | Graphic User Interface - Grafické uživatelské rozhraní |
| GSM | Groupe Spécial Mobile - Systém pro mobilní komunikace |
| GPS | Global Positioning System - Globální polohový systém |
| 4G | mobilní telekomunikační technologie 4.generace |
| Wi-Fi | bezdrátová komunikace v počítačové síti |
| USB | univerzální sériová sběrnice |
| HAL | Hardware abstraction layer - hardwarová abstraktní vrstva |
| DEX | formát souborů se použitím k inicializaci a spouštění aplikací |
| CLASS | kompilované Java soubory |
| JAR | soubor Java Archive |
| API | Application Programming Interface - rozhraní pro programování aplikací |
| XML | eXtensible Markup Language - značkovací jazyk |
| Android SDK | Studio pro vývoj Android aplikací |
| APK | formát souborů aplikací Android |
| SSE2 | Streaming SIMD Extensions 2 - instrukční sada procesorů AMD |
| DX10 | Microsoft DirectX verze 10 - instrukční sada procesorů Intel |
| MIT | licence vzniklá na Massachusettském technologickém institutu |
| AMD | Advanced Micro Devices - výrobce procesorů |
| RAM | polovodičové operační paměti |
| PERT | metoda pro výpočet kritické cesty |

10. Seznam příloh

Jakub Vozka



Věk: 32
Práce: Project manager v IT společnosti
Rodina: Ženatý
Adresa: Olomouc, ČR

Společenský Cílevědomý Spolehlivý

Životopis

Jakub pracuje jako project manager v IT firmě v Olomouci. Má pod sebou tým zkušených programátorů a grafiků. Volný čas nejvíce tráví v rodinném kruhu a zajímá se o nové IT trendy. Mezi jeho oblíbené činnosti patří i hraní her a sledování fantasy filmů.

Osobnost

| | |
|-----------|-----------|
| Introvert | Extrovert |
| Myšlení | Cit |
| Vnímání | Intuice |
| Pasivní | Aktivní |

Informační prostředky

Mobil
Notebook
Televize

Oblíbené aplikace

Instagram Facebook

Motivace

Pobídnutí
Strach
Růst
Síla
Společnost

Cíle

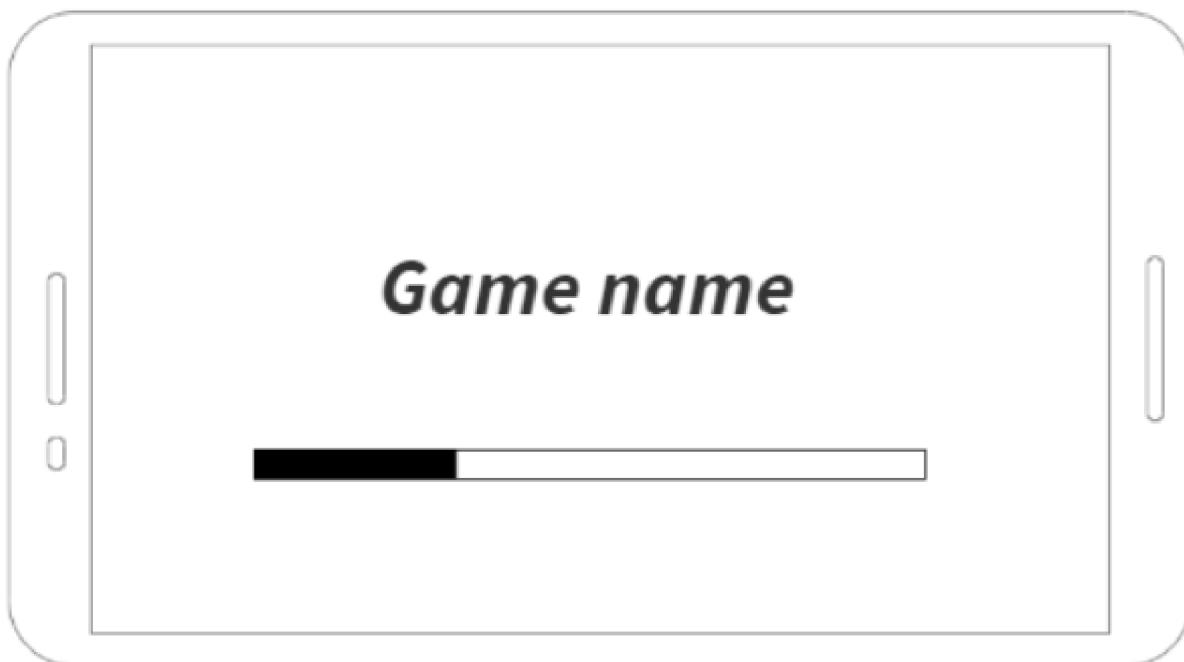
- Postarat se o rodinu
- Být přínosem
- Cestovat

Obavy

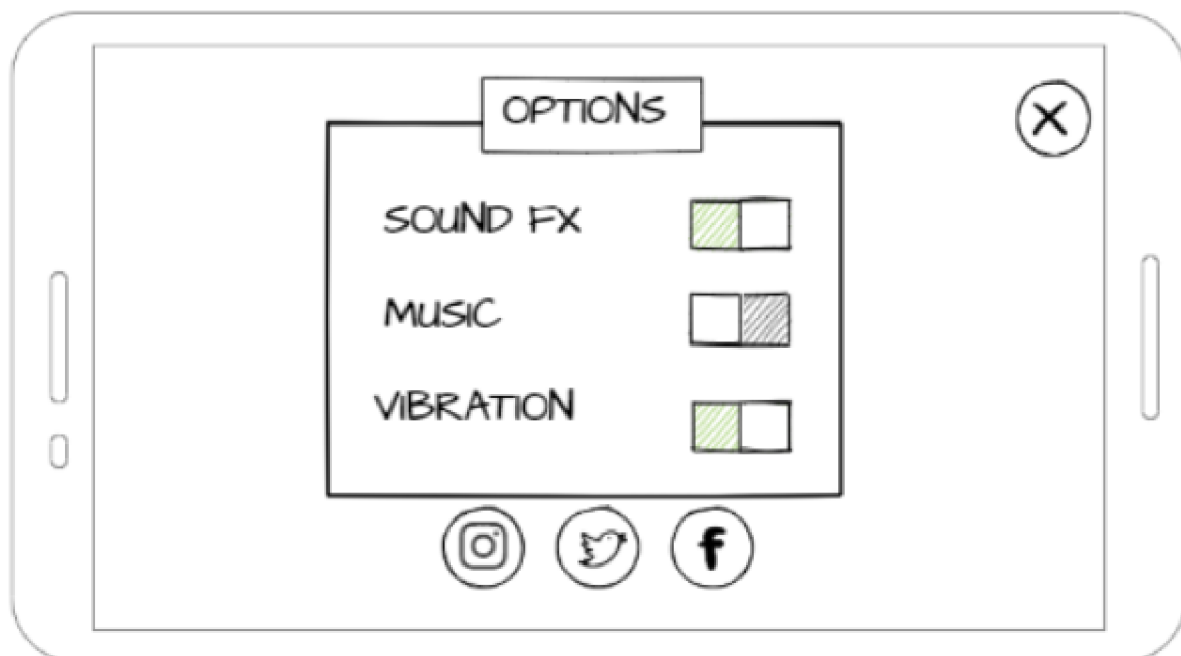
- Strach z neúspěchu
- Neoblíbenost

„S poctivostí nejdál dojdeš.“

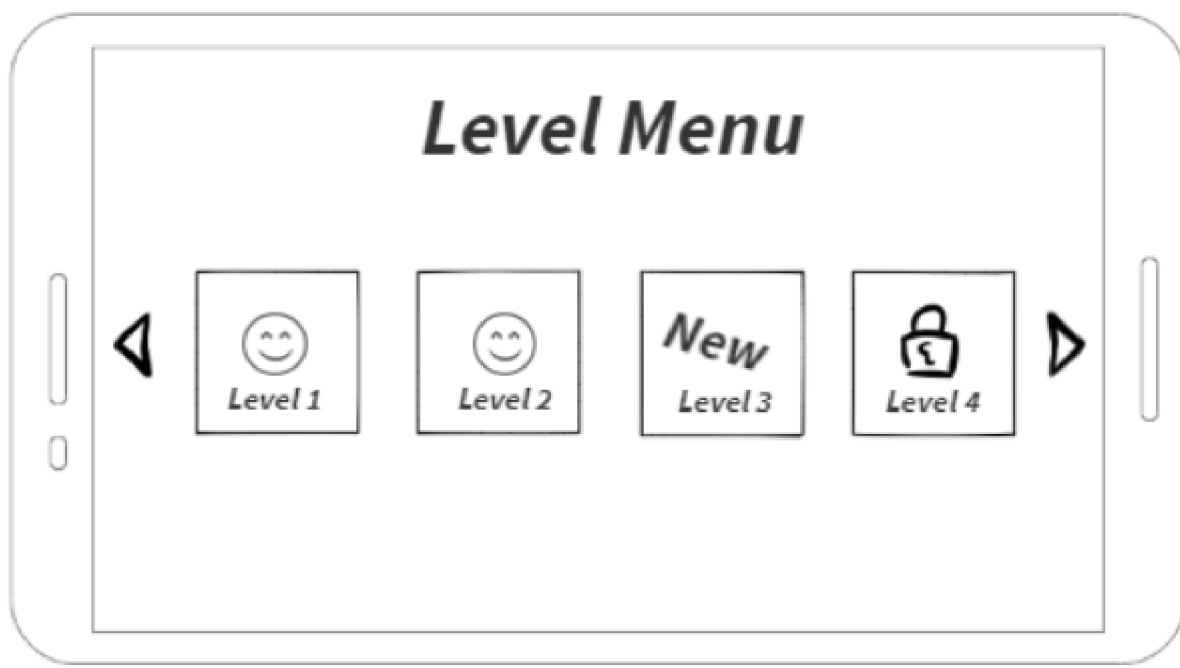
Obrázek 10.1: Druhá persona



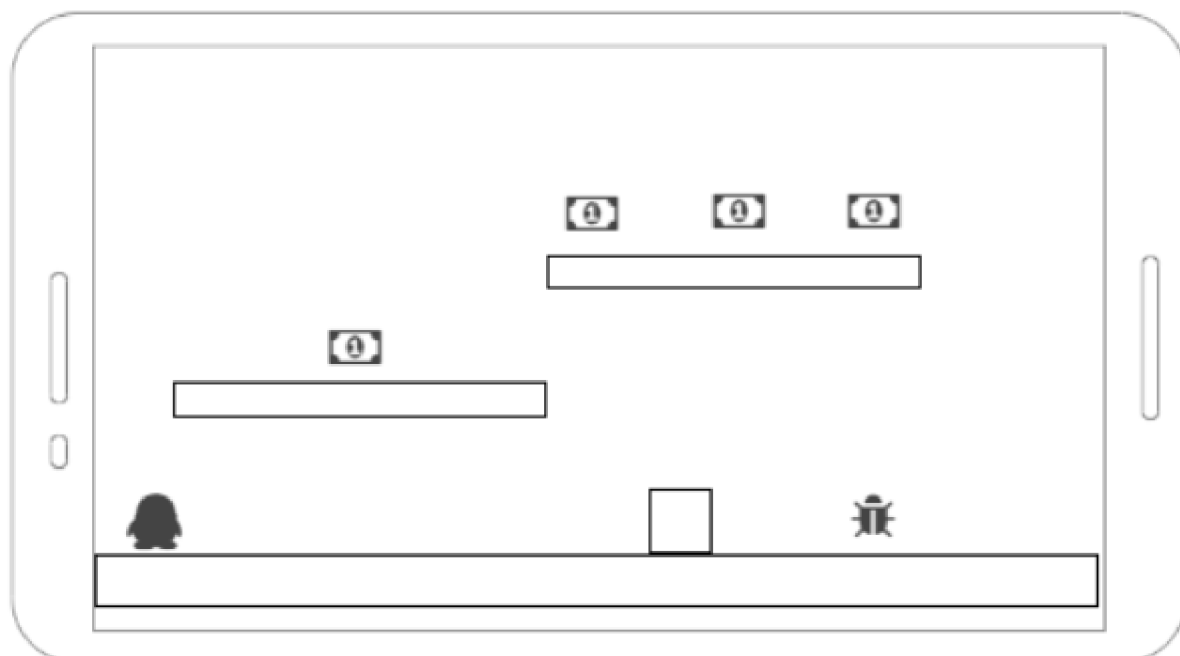
Obrázek 10.2: Návrh úvodní obrazovky aplikace



Obrázek 10.3: Návrh nastavení aplikace



Obrázek 10.4: Návrh výběru úrovní aplikace



Obrázek 10.5: Návrh úrovně aplikace



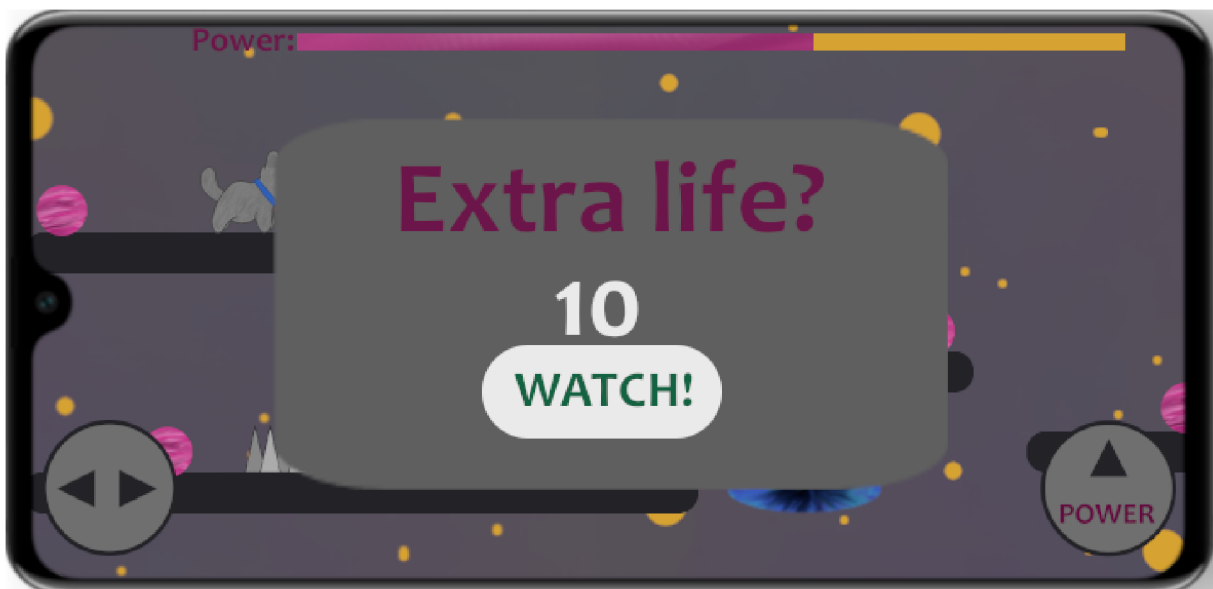
Obrázek 10.6: Mockup úvodní obrazovky aplikace



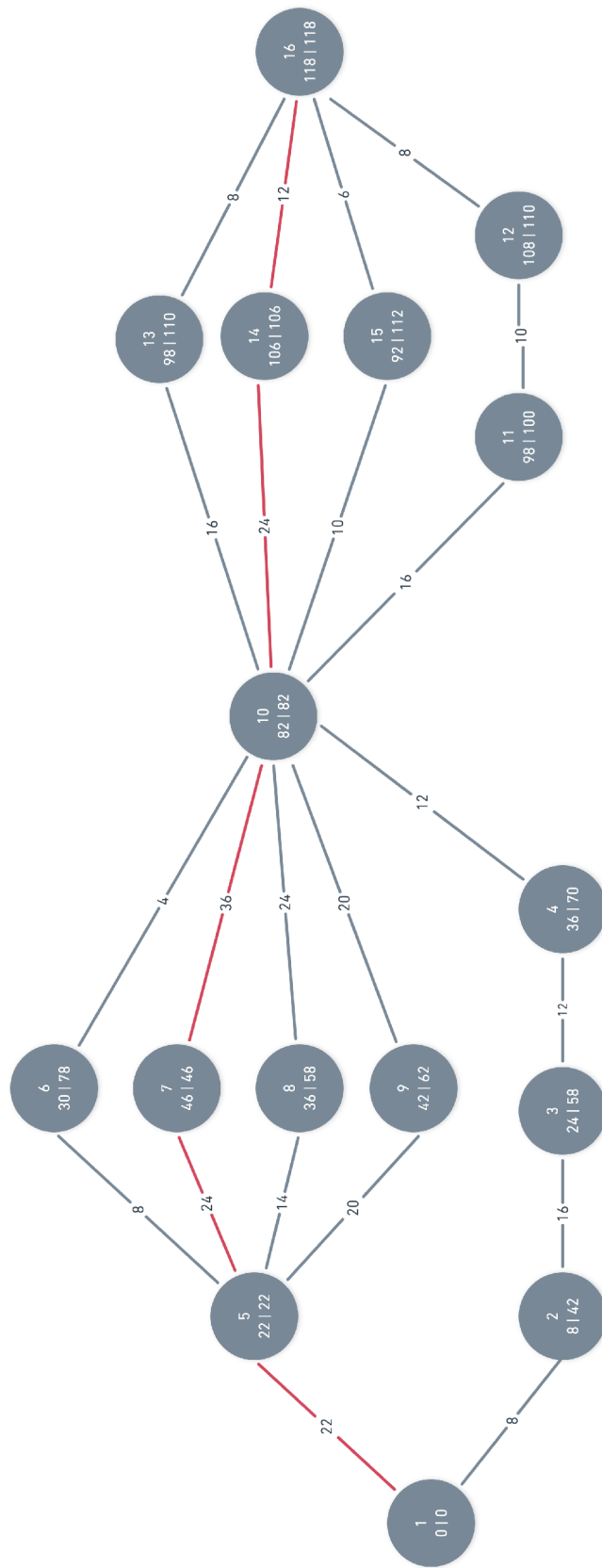
Obrázek 10.7: Mockup menu aplikace



Obrázek 10.8: Mockup úrovně aplikace



Obrázek 10.9: Mockup návratu do hry přes reklamu



Obrázek 10.10: Znázornění časové náročnosti aplikace pomocí metody PERT

| Uzel | Popis |
|-------------|---------------------------|
| <i>1</i> | Začátek |
| <i>2</i> | Základní grafika |
| <i>3</i> | Pohyb postavy |
| <i>4</i> | Pohyb nepřátel |
| <i>5</i> | Grafický návrh |
| <i>6</i> | Grafika pozadí |
| <i>7</i> | Grafika objektů |
| <i>8</i> | Grafika postavy |
| <i>9</i> | Grafika nepřátel |
| <i>10</i> | Vytvoření kolizí |
| <i>11</i> | Vytvoření ukázkové úrovně |
| <i>12</i> | Úprava ukázkové úrovně |
| <i>13</i> | Obrazovka Menu |
| <i>14</i> | Obrazovka Shop |
| <i>15</i> | Obrazovka Nastavení |
| <i>16</i> | Konec |

Tabulka 10.1: Popis činností v metodě PERT

| Přechod | Popis | Přechod | Popis |
|-------------------|--------------|--------------------|--------------|
| <i>1 -> 2</i> | 8 | <i>8 -> 10</i> | 24 |
| <i>1 -> 5</i> | 22 | <i>9 -> 10</i> | 20 |
| <i>2 -> 3</i> | 16 | <i>10 -> 11</i> | 16 |
| <i>3 -> 4</i> | 12 | <i>10 -> 13</i> | 16 |
| <i>4 -> 10</i> | 12 | <i>10 -> 14</i> | 24 |
| <i>5 -> 6</i> | 8 | <i>10 -> 15</i> | 10 |
| <i>5 -> 7</i> | 24 | <i>11 -> 12</i> | 10 |
| <i>5 -> 8</i> | 14 | <i>12 -> 16</i> | 8 |
| <i>5 -> 9</i> | 20 | <i>13 -> 16</i> | 8 |
| <i>6 -> 10</i> | 4 | <i>14 -> 16</i> | 12 |
| <i>7 -> 10</i> | 36 | <i>15 -> 16</i> | 6 |

Tabulka 10.2: Popis dob trvání mezi jednotlivými uzly v metodě PERT