



Ekonomická
fakulta
Faculty
of Economics

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích
Ekonomická fakulta
Katedra aplikované matematiky a informatiky

Bakalářská práce

Vývoj mobilní herní aplikace v prostředí Unity

Vypracoval: Petr Grimm
Vedoucí práce: Mgr. Radim Remeš

České Budějovice 2019

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH
Ekonomická fakulta
Akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr GRIMM**
Osobní číslo: **E16535**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **Ekonomická informatika**
Název tématu: **Vývoj mobilní herní aplikace v prostředí Unity**
Zadávací katedra: **Katedra aplikované matematiky a informatiky**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je vytvořit aplikaci pro mobilní zařízení pomocí herního engine Unity. Hra bude ve stylu deskové hry, ve hře bude uživatel pohybovat postavou po hrací ploše s mapou. Na mapě budou umístěny věže s danými parametry, protivníci a další objekty, které budou ovlivňovat průběh hry. Hra bude obsahovat alespoň 10 úrovní.

Metodický postup:

1. Studium odborné literatury.
2. Publikace výsledků rešerše.
3. Návrh a popis vývoje a implementace výsledné aplikace.
4. Zhodnocení hratelnosti aplikace ostatními uživateli, vypracování doporučení a závěrů.

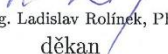
Rozsah grafických prací: **dle potřeby**
Rozsah pracovní zprávy: **40 - 50 stran**
Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:


1. DaGraca, M., & Lukosek, G. (2018). *Learning C# 7 By Developing Games with Unity 2017*. (3rd ed.). Birmingham, UK: Packt.
2. Doran, J. P. (2017). *Unity 2017 Mobile Game Development*. Birmingham, UK: Packt.
3. Hocking, J. (2015). *Unity in Action: Multiplatform game development in C# with Unity 5.* Shelter Island, NY: Manning Publications.
4. Lavieri, E. (2018). *Getting Started with Unity 2018: A Beginner's Guide to 2D and 3D game development with Unity*. (3rd ed.). Birmingham, UK: Packt.
5. Lintrami, T. (2018). *Unity 2017 Game Development Essentials*. (3rd ed.). Birmingham, UK: Packt.
6. Price, M. J. (2017). *C# 7.1 and .NET Core 2.0: Modern Cross-Platform Development*. (3rd ed.). Birmingham, UK: Packt Publishing.
7. Sapio, F., & Ferro, L. S. (2018). *Unity 2017 2D Game Development Projects*. Birmingham, UK: Packt.
8. Thorn, A. (2017). *Mastering Unity 2017 Game Development with C#*. (2nd ed.). Birmingham, UK: Packt.
9. Troelsen, A., DaGraca, M., & Lukosek, G. (2018). *Learning C# 7 By Developing Games with Unity 2017*. (3rd ed.). Birmingham, UK: Packt.

Vedoucí bakalářské práce: **Mgr. Radim Remeš**
Katedra aplikované matematiky a informatiky

Datum zadání bakalářské práce: **19. ledna 2018**
Termín odevzdání bakalářské práce: **12. dubna 2019**


doc. Ing. Ladislav Rolínek, Ph.D.
děkan

JIHOČESKÁ UNIVERZITA
V ČESKÝCH BUDĚJOVICÍCH
EKONOMICKÁ FAKULTA
Studentská 13 (2e)
370 05 České Budějovice


doc. RNDr. Jana Klícnarová, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 29. března 2018

Prohlašuji, že svoji bakalářskou práci „Vývoj mobilní herní aplikace v prostředí Unity“ jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47 zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to - v nezkrácené podobě/v úpravě vzniklé vypuštěním vyznačených částí archivovaných Ekonomickou fakultou - elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

.....
Datum

.....
Podpis studenta

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé bakalářské práce Mgr. Radimu Remešovi, za veškeré cenné rady, trpělivost a schovívavost. Dále bych chtěl poděkovat mé rodině za jejich podporu během studia.

Obsah

1	Úvod.....	11
2	Základní pojmy	12
2.1	Mobilní operační systém	12
2.2	Bootloader	12
2.3	ROM.....	12
2.4	Kernel.....	12
2.5	Knihovny.....	13
2.6	Game engine.....	13
2.7	Skriptování	13
2.8	Framework	14
2.9	Programovací jazyk.....	14
2.10	Mobilní Aplikace.....	15
2.11	C Sharp	15
3	Operační systémy pro mobilní zařízení	16
3.1	Podíl operačních systémů na trhu	16
3.2	Android	16
3.3	iOS.....	17
3.4	Windows Phone	17
4	Herní enginy	18
4.1	Unity.....	18
4.1.1	Unity Asset Store	20
4.1.2	Služby	20
4.1.3	Dostupné verze Unity	23

4.1.4	Hry vyvinuté v Unity	24
4.1.5	Unity editor	24
4.2	Unreal engine	26
4.3	Grafika.....	27
4.3.1	2D grafika	28
4.3.2	3D grafika	28
4.3.3	Modelování	28
4.3.4	Grafické modelovací nástroje	29
5	Vývoj mobilní aplikace.....	30
5.1	Vývojové nástroje	30
5.1.1	Software development kit	30
5.1.2	iOS SDK	30
5.1.3	Java development kit	31
5.1.4	Android Debug Bridge.....	31
5.2	Mobilní hry.....	31
5.2.1	Desková hra	32
5.2.2	Věžovky	32
6	Vytváření aplikace – praktická část	33
6.1	Základní informace o hře a vývoji	33
6.2	Debuggování a zkoušení hry pomocí Unity remote.....	33
6.3	Nahrání hry na Unity cloud.....	34
6.4	Tvorba modelů	35
6.5	Vytváření prostředí – levelů.....	35

6.6	Pohyb kamery.....	36
6.7	Nepřátelé	37
6.7.1	Pohyb nepřátel	37
6.8	Věže.....	39
6.8.1	Druhy věží.....	39
6.9	Uživatelské rozhraní.....	42
6.9.1	Hlavní menu.....	42
6.9.2	Zvolení úrovně	43
6.9.3	Stavění věží.....	44
6.9.4	Úroveň hotová.....	44
6.9.5	Konec hry.....	45
6.9.6	Pauza.....	45
6.9.7	Životy a odpočet	45
6.10	Kompilace hry	45
6.11	Problémy během testování	46
7	Závěr	48
8	Summary.....	49
9	Bibliografie	50
10	Seznam obrázků.....	55
11	Seznam tabulek	57
12	Seznam příloh	i

1 Úvod

Mobilní zařízení se neustále vyvíjejí a mnoho uživatelů přechází ze stolních počítačů a notebooků na tablety a telefony, tato situace ale nenastává u vývojářů, ty vyvíjejí stále hlavně hry na počítače. Vývoj takové hry může trvat i několik let a stále jí to nezaručí úspěch. Dnes lidé stále někam cestují a většina z nich si v ten moment vyndá telefon a začne si číst novinky, psát s kamarády anebo hrát hry. Již v minulosti bylo několik hitů s extrémně jednoduchým konceptem, které vydělali miliardy. Historicky je možné zmínit tři takové tituly, a to Flappy bird, tato hra vydělávala 50 tisíc dolarů denně, a to pouze z reklam, tato hra byla vyvinuta pouze jedním vývojářem. Candy crush je hra už od velké společnosti, ale tato hra vydělávala v dnech své největší slávy 630 tisíc dolarů. Posledním titulem je hra Pokémon Go, tato hra vydělávala denně 1,8 milionu dolarů po přidání obchodování mezi hráči. (Spence, 2014). (Fogel, 2019), (Crecente, 2018)

Cílem této bakalářské práce je čtenáře seznámit co je to engine, jaké vývojové nástroje se využívají pro tvorbu aplikací, jaké herní enginy se ještě využívají a poté přímo s vývojovým prostředím Unity 3D, jeho historií a jeho prvky.

V praktické části bude čtenář seznámen následným vývojem herní aplikace zaměřené pro mobilní zařízení v programovacím jazyce C#. Výsledek této práce je desková hra typu věžovek.

2 Základní pojmy

2.1 Mobilní operační systém

Jedná se o komplexní software, který je navržen pro mobilní zařízení typu tablet, mobilní telefon, čtečky a jiná zařízení. Operační systém je navržen tak aby zajistil chod aplikací a jiných funkcí telefonu. Zároveň určuje, jaké aplikace mohou na zařízení být spuštěny. (Beal, 2011)

Mezi jeho základní funkce patří umožnit uživateli ovládat mobilní zařízení, možnost spouštět aplikace a těmto aplikacím přiřazovat potřebné zdroje pro chod. (Beal, 2011)

2.2 Bootloader

Jedná se o samostatný program, který se spouští při zapnutí mobilního zařízení a spouští potřebné procesy, které jsou třeba pro zapnutí operačního systému. Tento program umožňuje i další funkce například různé testy, restartování mobilního zařízení do továrního nastavení, instalace různých aplikací, aktualizace operačního systému z paměťové karty atd. U většiny telefonu je tato funkce skryta a lze ji zpřístupnit pouze kombinací určitých tlačítek, která se u každého telefonu liší. (Serych, 2007), (Techopedia, 2013)

2.3 ROM

Read-only memory jedná se speciální paměť elektronických zařízení, do které je možné zapisovat pouze velice obtížně, a to ve speciálním režimu anebo vůbec. V této paměti je například operační systém nebo firmware. ROM může dále obsahovat speciální verzi paměti Radio ROM, v této paměti jsou uloženy například informace o sim v případě, že zařízení blokováno pouze na jednoho operátora a veškeré informace ohledně připojení bluetooth a Wi-Fi. Jako další příklad je možné použít EEPROM, v této paměti je uložený BIOS pro počítače. (Computer Hope, 2019)

2.4 Kernel

Jedná se o jádro neboli hlavní modul operačního systému. Tento modul je při načítání systému načtený jako první a zůstává zapnutý až do vypnutí systému, z toho důvodu je

důležité, aby bylo jádro co nejmenší ale zároveň poskytovalo veškeré nezbytné služby, které jsou nutné pro chod operačního systému ale i aplikací. Jádra jsou přidělovány vlastní zdroje, aby nedošlo k nechtěnému přepsání jinými programy/aplikacemi. Jádro je zodpovědné například za správu procesů, paměti, napájení, systémové připojení atd. Jádro v operačním systému Android je speciálně upravené jádro od Linuxu. (AndroidMarket, 2011), (Beal, 2017)

2.5 Knihovny

Jedná se o sady funkcí, zdrojů a modulů, které využívají různé aplikace pro svůj chod. Moduly jsou uloženy ve formátu objektu to je užitečné pro ukládání často používaných modulů, jelikož není nutné je explicitně propojovat s každým programem. U operačního systému Android se například knihovna surface manager stará o to správné zobrazování aplikací a vykreslování aplikací, která běží na rozdílném procesu. Knihovna Open GL slouží pro práci s 3D grafikou knihovna SGL se stará o 2D grafiku. (AndroidMarket, 2011), (Beal, 2001), (Jain, 2014)

2.6 Game engine

Software nebo také v některých případech framework, který pomáhá vývojářům při vývoji a poskytuje jim řadu funkcí, které využijí pro efektivní a rychlé vytváření her. Tento software poskytuje vývojářům řadu možností jako implementaci zvuku, fyzické simulace, správu animací a grafiky ale i správu připojení pro hru pro více hráčů, nejdůležitější část je nicméně logika a knihovny, kterou herní engine používá pro programování. Dříve bylo nutné všechny tyto funkce naprogramovat jednotlivě, to právě díky herním enginům odpadá a vývojáři se mohou soustředit jen na to nejdůležitější. Existuje mnoho druhů herních enginů, nicméně některé enginy jsou lepší na jiné žánry her, například Unreal engine 4 se specializuje spíše na first-person shooter hry zkráceně FPS nebo česky střílečky. Unity na druhou stranu je takový univerzální engine, ve kterém se dají udělat všechny typy her. (Ward, 2008), (Unity, 2018)

2.7 Skriptování

Skriptování je jeden z nejdůležitějších prvků vývoje her, jelikož žádná hra se neobejde bez skriptů. Skripty jsou psány v programovacím jazyce, nejznámější skriptovací jazyk

je C#. Skripty slouží k naprogramování základní logiky hry, implementaci různých knihoven, například knihovny pro základní fyziku, dále slouží k detekci vstupů od uživatele, ale i grafické a zvukové efekty i chování nepřátel. V podstatě vše, co ve hře je, je řešeno prostřednictvím skriptů. (Unity, 2018)

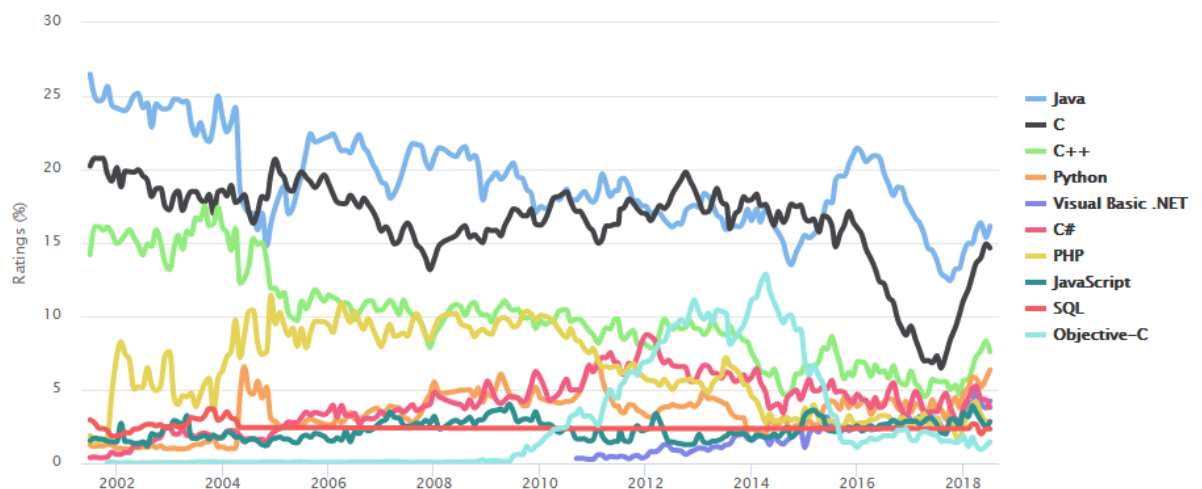
2.8 Framework

Framework je software, případně vývojové prostředí, které má v sobě předdefinované základní knihovny, třídy, funkce, rozhraní a logiku, které mohou být vývojáři rozšířeny o své části kódu a slouží k vývoji softwaru. Celkově tedy zjednodušuje vývojový proces, protože programátoři nemusejí znovu psát celý kód, ale mohou využít již takto předdefinované prostředí. (Techterms, 2013), (Computer Hope, 2017)

2.9 Programovací jazyk

Zjednodušeně programovací jazyk je jazyk pomocí, kterého se vytvářejí základní formy příkazů, které je poté možné implementovat do kódu programu, který následně ovlivňuje chování prostřednictvím různých vstupů počítač. Programovací jazyky lze rozdělit na základní například binární, který je zapsán prostřednictvím 0 a 1, a na složitější typu C, C#, Java, Javascript, Python, Basic atd., tyto jazyky jsou poté překládány kompilátorem. Na (Obrázek 1) lze vidět přehled nejpoužívanějších programovacích jazyků. (Dr. Garbade, 2018)

Obrázek 1: Graf nejpoužívanějších programovacích jazyků



(Dr. Garbade, 2018)

2.10 Mobilní Aplikace

Je program nebo skupina programů, které jsou vyvinuty pro mobilní zařízení. Tyto aplikace poskytují podobné služby jako aplikace pro počítače, jelikož jsou navrženy tak, aby byly malé, mají omezenou funkčnost.

2.11 C Sharp

Za otce programovacího jazyka C# je považovaný Anders Hejlsberg. Ten během vývoje frameworku .NET, který byl původně napsaný v programovacím jazyce SMC (Simple Managed C), vymyslel programovací jazyk, který se v té době pojmenoval jako Cool zkratka pro C object oriented language to v češtině znamená C objektově orientovaný jazyk. Hejlsberg se během vývoje tohoto jazyka řádl základy CLR (Common Language Runtime) Původně Microsoft uvažoval že Cool bude konečný název tohoto jazyka, nicméně se tak nerozhodl, a to z důvodu ochranné známky. Nakonec se jazyk pojmenoval jako C # a byl zveřejněn na konferenci profesionálních vývojářů (Professional Developers Conference) v červenci 2000. V době zveřejnění se o jazyku C sharp říkalo, že to jen kopie programovacího Java. V době vydání C sharp bylo mnoho funkcí podobných Javě, C sharp 1.0 obsahoval jen základní funkce, jako třídy, struktury, atributy, výrazy, události atd. Microsoft pokračoval ve vývoji C sharpu a ten nabíral na popularitě, v dnešní době je už ve vývoji verze C# 8.0 a od původní verze vydané v roce 2000 ušel tento jazyk už velký kus cesty. (Dietrich & Smacchia, 2017)

3 Operační systémy pro mobilní zařízení

3.1 Podíl operačních systémů na trhu

Operační systém Android od června 2012 drží prvenství v podílu na trhu, jeho největší konkurent je iOS. Podíl nejznámějších operačních systémů na trhu v březnu od roku 2013 je zobrazen v tabulce. Z (Tabulka 1) lze vyčíst že Android společně s iOS pokrývají v roce 2019 přibližně 97,5% trhu. (StatCounter, 2017)

Tabulka 1: Podíl operačních systémů na trhu

Rok	Android	iOS	Windows	Symbian	BlackBerry	Neznámý
2013	39.21	24.03	1.59	6.82	3.58	4.29
2014	53.65	23.95	2.35	2.7	1.82	3.2
2015	64.2	20.2	2.29	1.18	1.19	3.16
2016	69.11	19.29	1.75	0.51	0.75	3.44
2017	72.63	19.65	0.89	0.21	0.3	3.73
2018	75.45	20.47	0.47	0.11	0.1	1.94
2019	74.69	22.81	0.29	0.05	0.05	0.39

(StatCounter, 2017)

3.2 Android

Operační systém Android je open source systém založený na Linuxu, který patří mezi nejvíce používané operační systémy pro mobilní zařízení, tento operační systém byl původně vyvinut společností Android Inc., následná spolupráce Open Handset Alliance, kterou vedla společnost Google, vytvořila operační systém Android. Open Handset Alliance je skupina 84 technologických a mobilních firem, které se spojily pod společností Google za účelem rychlejšího vývoje v oblasti mobilních zařízení. První verze Android 1.0 vznikla v roce 2007. (Techopedia, 2018) (openhandsalliance, 2007)

Primárním vývojovým jazykem pro Android aplikace je Java, nicméně Android neobsahuje žádný vývojový nástroj pro Javu. Google tento nástroj nahradil svým vlastním nástrojem Dalvik, který je určen speciálně pro Android. Tento nástroj spouští zkompileovaný kód Javy a převádí ho do bajt kódu Dalvik, ten je navržen tak aby optimalizoval využití baterie a hlavní funkcionalitu mobilního zařízení s limitovaným výkonem procesoru a velikostí paměti. (Techopedia, 2018) (openhandsalliance, 2007)

3.3 iOS

iOS je oproti Androidu uzavřený operační systém, který v roce 2007 vyvinula společnost Apple pro své telefonní iPhone, tablet iPad, televizi Apple TV, hudební přehrávače iPod a další zařízení z periferie produktu Apple. Systém je odvozen ze systému Mac OS X, který je podobný Unixu. (Techopedia, 2011)

3.4 Windows Phone

Windows Phone je operační systém od společností Microsoft, tento je nástupcem systému Windows Mobile. Windows Phone je založen na uživatelském rozhraní „Metro“, které bylo poprvé představeno v operačním systému Windows 8 pro počítače. Tento systém je sestaven ze systému „dlaždic“, každá tato dlaždice představuje odkaz na aplikaci. (Stroud, 2018)

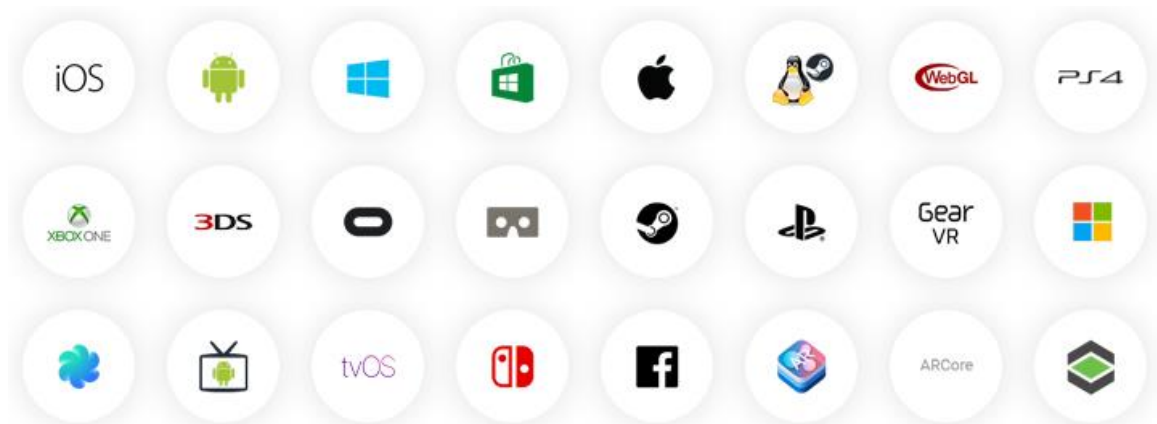
4 Herní enginey

4.1 Unity

Unity je multiplatformní herní engine vyvinut společností Unity Technologies, prvotní plán při vývoji Unity bylo vývojářům poskytnout nástroj, který bude za rozumnou cenu, ale poskytovat profesionální nástroje pro tvorbu her. Bizarní věc na Unity je to, že první několik měsíců byla jediná podporovaná platforma Mac. Ten v té době na tom byl mnohem oproti Windows než teď. První verze Unity byla vydána v roce 2005, několik měsíců poté byla do Unity přidána podpora pro Windows a webové aplikace.

K dnešnímu dni (20.3.2019) lze dělat hry na 29 platformech (Android, iOS, Nintendo 3ds, Nintendo switch atd.), z toho důvodu je to také jeden z nejoblíbenějších herních engineů. Přehled většiny podporovaných platformech je na (Obrázek 2). (Brodkin, 2013), (Unity, 2006)

Obrázek 2: Podporované platformy Unity



(Unity, 2006)

V unity je možné vytvářet 2D i 3D hry s podporou různých funkcí a funkcionalit, ale i filmy, modely pro architekturu, aplikace ve virtuální realitě využitelné při bezpečnostních školení. Pro designéry a umělce jsou zde nástroje, které jsou intuitivní. Funkce Timeline umožňuje vytvářet filmové ukázky v průběhu hry tzv. cinematic nebo také cutscene, prostřednictvím animací, objektů a scén. Kameru má na starosti funkce Cinemachine, jedná se o automatickou kameru, která na základě kompozice scény vytváří co nejlepší záběry. V případě potřeby filtrů je možné využít funkci Post-Processing Stack. Unity podporuje nástroje pro správu hudby, její mixování a případné

efekty. Momentální logo, které unity používá je zobrazeno na (Obrázek 3). (Unity, 2006)

Obrázek 3: Logo Unity



(Unity, 2017)

Unity je dostupné pro operační systém Windows a Mac. Hlavní skriptovacím jazykem je C# s podporou až do verze 7.3. Unity podporuje fyzické engine Box2D a NVIDIA PhysX, fyzické engine slouží například k simulaci gravitace a jiných fyzikálních zákonů. Další výhodou unity je, že lze importovat mnoho různých formátů zvuku, textu, obrázků i videa, přehled všech podporovaných formátů je na (Obrázek 4) (Unity, 2006).

Obrázek 4: Podporované formáty pro import

image formats		
.psd	.jpg	.png
.gif	.bmp	.tga
.tiff	.iff	.pict
.dds		

audio formats		
.mp3	.ogg	.aiff
.wav	.mod	.it
.sm3		

video formats		
.mov	.avi	.asf
.mpg	.mpeg	.mp4

text formats		
.txt	.htm	.html
.xml	.bytes	

(Unity, 2006)

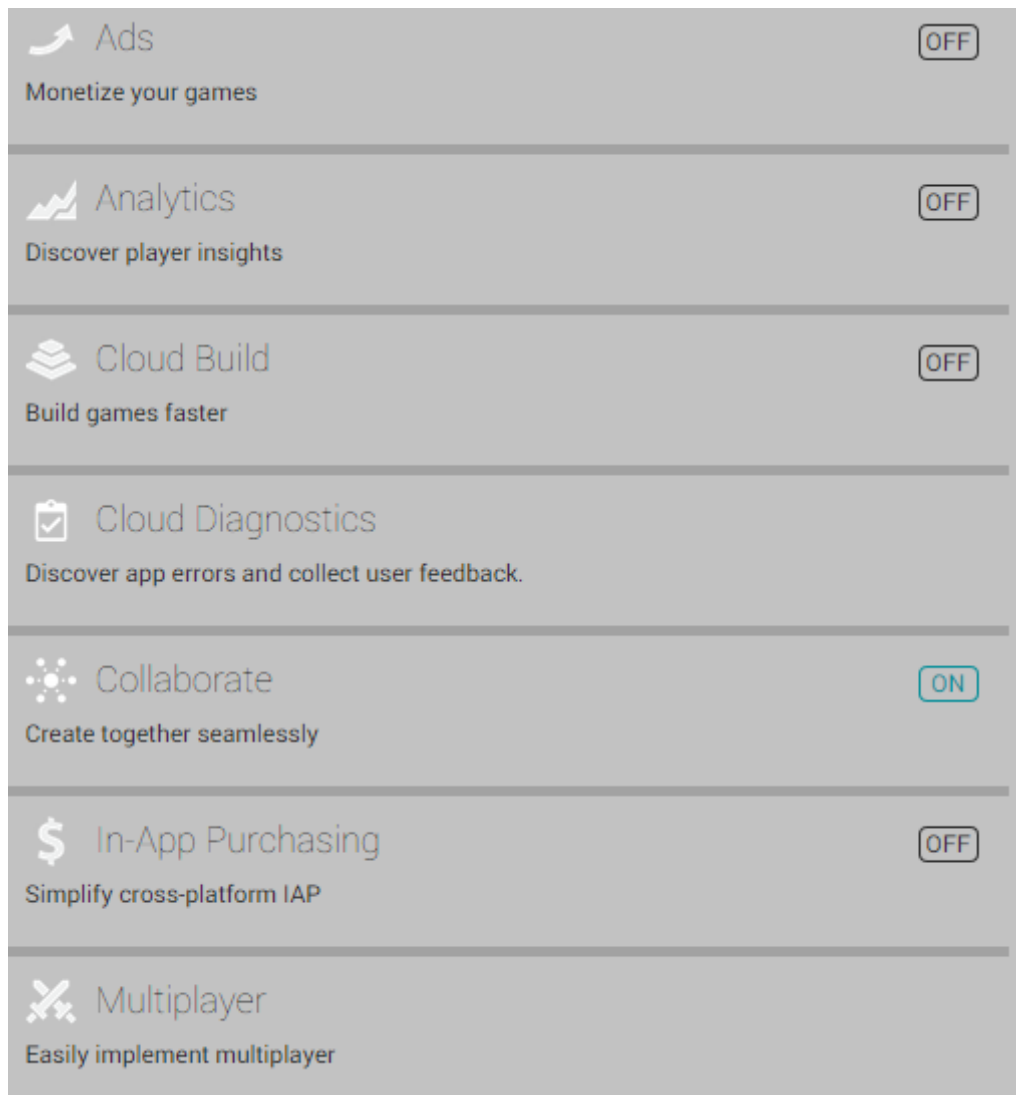
4.1.1 Unity Asset Store

V případě potřeby nástrojů, které Unity neobsahuje je možné je vyvinout dle vlastních představ, případně získat zdarma anebo zakoupit v obchodě Unity Asset Store. V tomto obchodě je možné získat nejen nástroje ale i modely pro 2D a 3D hry, zvuky, šablony, které obsahují od modelů a zvuků až po animace, ale i vizuální efekty. (Unity, 2015)

4.1.2 Služby

Unity poskytuje také mnoho služeb, které umožňují vývojářům získávat statistické informace o hře anebo funkce pro týmy vývojářů, které jim umožňují efektivnější spolupracovat, mezi další služby patří například implementace reklam přímo od Unity nebo multiplayer (hra pro více hráčů), který je na serverech, které spravuje Unity. Dostupné služby pro unity Personal jsou na (Obrázek 5). (Unity, 2015)

Obrázek 5: Přehled dostupných služeb v Unity



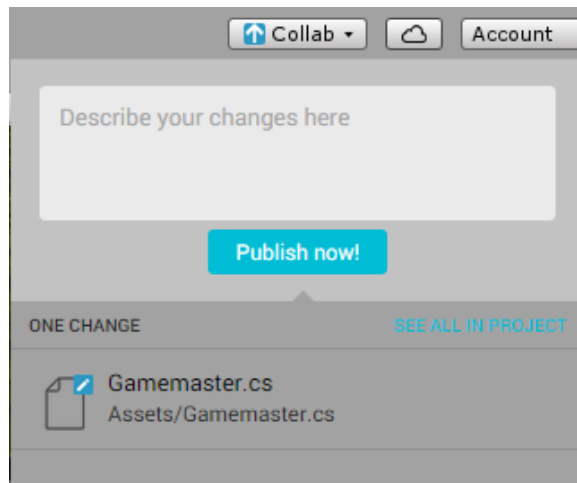
(zdroj: Autor)

4.1.2.1 Unity Cloud a Spolupráce (Collaborate)

Tato funkce umožňuje nahrát celý projekt na Unity Cloud, tato funkce je perfektní pro jakýkoliv tým, umožňuje pracovat několika lidem najednou a pak veškeré změny publikovat na cloud, odkud si je mohou ostatní členové snadno stáhnout. Ke změnám lze napsat komentář, prostřednictvím externích nástroj, je možné zobrazit jednotlivé změny. Tato funkce obsahuje i historii, tudíž je možné zobrazit kdo udělal na čem změnu. Tato funkce je, ale užitečná i pro jednotlivce, a to například pokud pracují na více zařízeních najednou, mohou totiž z jednoho zařízení nahrát změny jednoduchým kliknutím na tlačítko „Publish now!“ a vše se nahraje na cloud, na druhém zařízení je poté možné vše snadno stáhnout a synchronizovat, na (Obrázek 6) je zobrazeno, jak

nahrání na cloud vypadá. Zároveň je možné Unity Cloud považovat za takový bezpečnostní prvek pro zálohu projektu. (Unity, 2015)

Obrázek 6: Okno pro nahrání změn ve službě Collaborate



(zdroj: Autor)

4.1.2.2 Unity Ads a In-App Purchasing

Tyto služby umožňují snadnou implementaci a správu reklam od různých partnerů a případné nákupy v aplikaci. Reklamy a nákupy slouží jako forma monetizace hry a mohou být formou proužkové reklamy, krátkých spotů anebo nákupu měny nebo věcí. Reklamy jsou vybírány podle toho, kdo v ten moment nabízí za přehrání reklamy nejvíce, a to prostřednictvím aukce. Příklad této aukce je zobrazen na (Obrázek 7). Mezi oblíbenou formu reklam patří to, že je uživateli nabídnuto, zda chce reklamu zhlédnout, pokud se pro to rozhodne, je za to nějakým způsobem odměněn ve hře. (Unity, 2015)

Obrázek 7: Způsob, jakým funguje aukce pro reklamy



(Unity, 2015)

4.1.2.3 Multiplayer

Unity multiplayer umožňuje nejsnadnější implementaci této funkce, která je prostřednictvím této služby rychlé na zavedení a zároveň vysoce přizpůsobitelná. Servery jsou poskytovány přímo od Unity zdarma jedině, za co je nutné platit je provoz na tomto serveru, a to v poměru 0,49\$ za 1 GB provozu. Počet hráčů na serveru je odvozen dle verze předplatného Unity. (Unity, 2015)

4.1.2.4 Analytics - Analytické nástroje

Unity analytics (Obrázek 8) jsou velice užitečná služba, pouhou aktivací se začnou shromažďovat informace o chování hráčů, DAU znamená daily active users, to jsou hráči, kteří tuto hru zapínají každý den. MAU znamená monthly active users, aktivní hráči za měsíc, Users znázorňuje celkový počet uživatelů. Revenue ukazuje příjem této hry. Mezi další služby patří sledování, jak dlouho mají uživatelé v průměru zapnutou aplikaci, jaká je nejoblíbenější oblast ve hře prostřednictvím teplotních map a mnoho dalších užitečných údajů pro vývojáře. (Unity, 2015)

Obrázek 8: Statisticky ze služby analytics

DAU	MAU	# Users	Revenue
0	5	5	\$0.00
0%	0%	0%	0%

(zdroj: Autor)

4.1.3 Dostupné verze Unity

Existují 3 běžně dostupné verze Unity a to Personal, které je pro začátečníky a je zdarma, Plus, kteří to myslí už vážněji a Plus, které je určeno pro profesionály. V (Tabulka 2) je uvedeno několik rozdílů mezi jednotlivými verzemi. U Personal verze je úvodní obrazovka vždy Logo unity a popis „Made with Unity“ poté je až možnost vlastní animace a úvodní obrazovky, u ostatních verzí je tato funkce dobrovolná. Verze Plus a Pro podporují tzv. dark theme neboli tmavé schéma, tato funkce je pro uživatele příjemnější při práci a umožňuje jim více se koncentrovat. (Unity, 2006)

Tabulka 2: Přehled dostupných verzí Unity a některé rozdíly

Verze	Školící kurzy	Limit na multiplayer	Cloudové uložení	Prémiová podpora	Limit ročních tržeb	Cena za měsíc
Personal	ne	20 hráčů	1 GB	ne	100 000\$	zdarma
Plus	ano	50 hráčů	25 GB	ne	200 000\$	420\$ za rok nebo 35\$ měsíc
Pro	ano	200 hráčů	neuvedeno	ano	Bez limitu	125\$

(Unity, 2006)

4.1.4 Hry vyvinuté v Unity

Dle informací, které poskytl v rozhovoru John Ricitiello, byla alespoň jednou hra vyvinutá v Unity spuštěná na více než na 3 miliard zařízeních. Polovina celkových her vydaných na Android je právě z Unity a tato situace je i na trzích pro ostatní platformy, na trhu pro virtuální realitu tyto čísla dosahují až 70%. (Dillet, 2018)

Mezi nejznámější hru z unity je možné zařadit multiplatformní karetní hru Hearthstone z roku 2014 založená na příběhu a prostředí Warcraft od společnosti Blizzard Entertainment. V této hře sbírají hráči karty, které poté poskládají do balíčků a hraji proti sobě. Hra má více než 100 miliónů hráčů. Další velice oblíbená hra je Ori and the Blind Forest, jedná se o 2D Metroidvania hra od společnosti Moon Studios z roku 2015, která získala několik ocenění za grafické zpracování. (Unity, 2018)

4.1.5 Unity editor

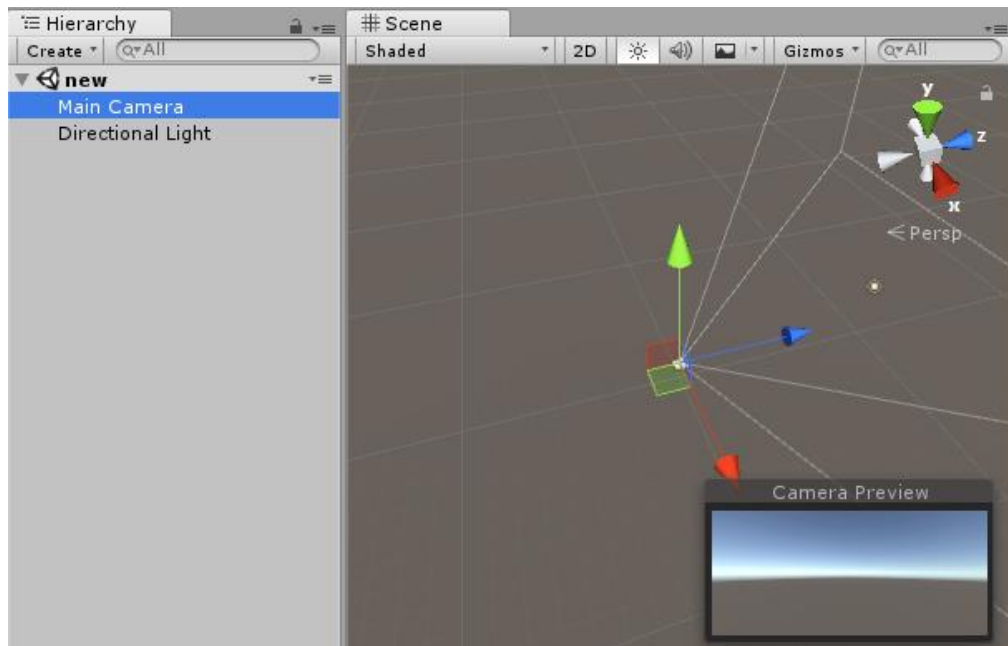
Unity obsahuje mnoho prvků a oken, mezi jeho nejdůležitější okna patří Scene (scéna), Inspektor, Console (konzole), Project, Hierarchy a Game view (hra).

4.1.5.1 Scéna, Hierarchy a Game view

V tomto okně je vidět náhled na scénu, každá scéna je považována za jednu úroveň a slouží k vytvoření této úrovně. Do scény je možné umístit herní objekty, pohybovat

s kamerou a světlem. V hierarchii jsou všechny objekty, které se v dané scéně nacházejí. Prázdnou scénu a hierarchii je možné vidět na (Obrázek 9).

Obrázek 9: Okna pro scénu a hierarchii



(zdroj: Autor)

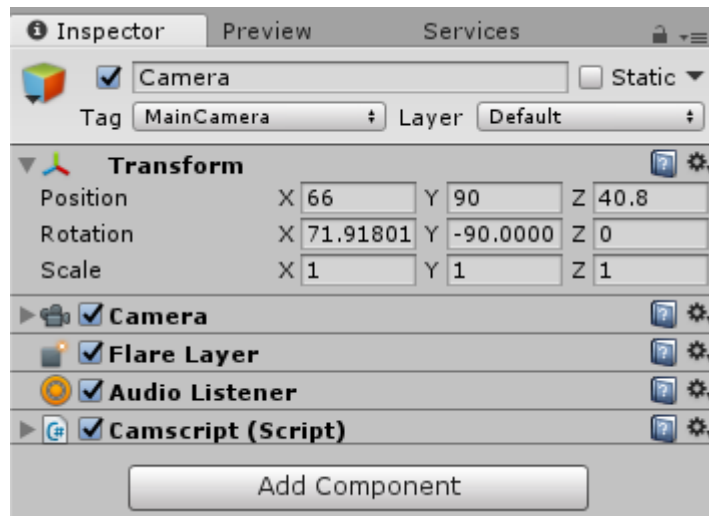
Herní objekty (Game Object) je vše co je umístěno ve scéně a je to hlavní nejdůležitější prvek v celém editoru. Prefab je přednastavený herní objekt, kterému byly již přiřazeny nějaké vlastnosti, hodnoty a komponenty, tyto prefaby je pak možné umisťovat do různých scén bez toho, aby jim bylo nutné nastavovat znova veškeré vlastnosti. Je ale i možné nastavit vlastnosti prefabu individuálně přímo ve scéně, tyto vlastnosti zůstanou pouze u tohoto prefabu, pokud jsou vlastnosti změněny v projektu jsou nastaveny pro všechny prefaby. Game view umožňuje hru spouštět bez nutnosti ji kompilovat, pohled na hru je skrz hlavní kameru, pokud tato funkce spuštěna je možné ve vedlejším okně inspektor upravovat herní objekty. Tyto změny se projeví ihned, po vypnutí se veškeré změny anulují. (Unity, 2018) (Unity, 2018)

4.1.5.2 Inspektor

Inspektor umožňuje nastavování vlastností herních objektů ale i materiálů, zvuků a světla. Přidělovat k objektům různé komponenty jako skripty, zvuky, okna pro uživatelské rozhraní a jeho rozložení. Většina akcí, které se s objekty nechají provést,

jsou právě pomocí inspektoru. Příklad inspektoru v momentě, kdy je označená kamera je možné vidět na (Obrázek 10). (Unity, 2015)

Obrázek 10: Okno inspektor



(zdroj: Autor)

4.2 Unreal engine

Mezi další velice oblíbené herní engine patří Unreal Engine od společnosti Epic Games momentálně je již dostupný Unreal Engine 4, který umožňuje multiplatformní vývoj her, přehled platforem zobrazen na (Obrázek 11). (EPIC GAMES, 2015)

Obrázek 11: Přehled platforem, které podporuje Unreal Engine

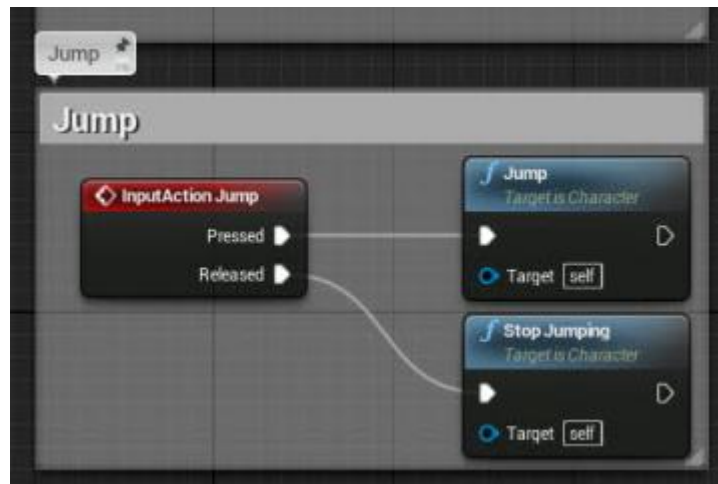


(Denham, 2019)

První verze tohoto enginu vznikla v roce 1998. V tomto enginu je možné používat psát skripty v programovacím jazyce C++ anebo používat tzv. blueprinting. Blueprinting je znázorněn na (Obrázek 12), jedná se o grafický skriptovací systém, který je založen na konceptu využití uzlového rozhraní pro vytváření herních prvků. Jak je tomu u většiny

skriptovacích jazyků, je i tento založen na definici objektově orientovaných tříd a objektů. Tento systém je považován že flexibilní a výkonný, jelikož designérům umožňuje používat nástroje, které jsou určeny spíše pro programátory. Prostřednictvím blueprintingu, je možné ovládat animace, chování postav ovládání Blueprinting a programovací jazyk C++ je možné kombinovat pro vytváření základních systému hry. (EPIC GAMES, 2015) (Denham, 2019)

Obrázek 12: Příklad blueprintingu



(Denham, 2019)

Unreal engine obsahuje i nástroje pro grafiku, které jsou intuitivní tak i fyzický engine PhysX 3.3, zároveň obsahuje nástroje pro správu zvuků a hudby, nástroje pro tvorbu prostředí, grafické nástroje a nástroje pro animaci a videa, nástroje pro uživatelské rozhraní a mnoho dalších. (Denham, 2019) (EPIC GAMES, 2015)

Unreal engine poskytován zdarma se všemi funkcemi a nástroji, pokud ale tržby ze hry budou přesahovat 3 000\$ za čtvrtletí, požaduje Epic Games 5% podíl. . (Denham, 2019)

4.3 Grafika

Grafika je vizuální prezentace objektu nebo obrazu. Grafiku je možné rozdělit do několika kategorií, a to na 2D grafiku, která se dále dělí na vektorovou a rastrovou grafiku a 3D grafiku. (PCMAG, 2013)

4.3.1 2D grafika

2D grafika představuje dvourozměrnou grafiku, ta je orientovaná na osách X, Y, Rastrová grafika je grafika je běžná u fotografií, technických výkresů, typografie a mnoho dalších. Rastrová grafika nebo také bitmapová grafika, která využívá systém mřížek a pixelů, které ve finále sestavují obraz. (PCMAG, 2013)

Vektorová grafika je matematicky složená grafika z řádků s předdefinovaným vzorce. Prostřednictvím této funkce je možné obrázky vytvořené touto grafikou zmenšovat nebo zvětšovat bez ztráty kvality. (PCMAG, 2013)

4.3.2 3D grafika

Trojrozměrná grafika je jedná z kategorií počítačové grafiky. Zatímco dvourozměrná grafika je zaměřená na barvu pixelů, trojrozměrná grafika bere v potaz barvu, jas a polohu. Tyto pixely předávají pozici na ose Z, která v trojrozměrné grafice představuje hloubku. Kombinace těchto pixelů tvoří povrch, který je v některých případech také nazýván jako textura. V trojrozměrné grafice spolu mohou spolupracovat objekty tak, že se překrývají. Některé grafické styly používají různé techniky, jednou z nich je například trasování paprsků, které slouží k vytváření realistických stínů. Konverze informací o trojrozměrných objektech do bitové mapy se nazývá vykreslování. Trojrozměrné modely ve vytvářejí modelováním. Vykreslování se využívá při zobrazování trojrozměrného objektu na obrazovce. (PCMAG, 2013)

4.3.3 Modelování

Pro modelování se využívají speciálně navržené softwary, které umožňují manipulaci s body tzv. vrcholy ve virtuálním prostoru, tak aby byla ve výsledku vytvořena kolekce vrcholů, která vytvoří finální objekt. Objekty mohou být generovány automaticky anebo ruční deformací této sítě. Trojrozměrné modely jsou využívány například ve videohrách, filmech, inženýrství, architektuře a v mnoho dalších odvětvích. (Petty, 2019)

4.3.4 Grafické modelovací nástroje

4.3.4.1 Blender

Blender je volně dostupný open-source program, ve kterém lze vytvářet 3D objekty, animace, filmy ale i vytváření her. Tento program je dodáván s velkým množstvím nástrojů, které mají profesionální úroveň, které je případně možné upravovat pomocí programovacího jazyka Python. (Blender, 2007)

Blender je multiplatformní software, které je možné spouštět na operačních systémech Linux, Windows i Mac. Jeho uživatelské rozhraní využívá formát OpenGL. OpenGL je formát, který specifikuje formát rozhraní pro multiplatformní aplikace. (Blender, 2007)

5 Vývoj mobilní aplikace

Vývoj aplikace je proces, během kterého vytváří počítačový program nebo programy. Pokud se aplikace vyvíjí na zakázku je nutné nejdříve získat veškeré informace, podle těchto informací navrhnout prototyp, tento prototyp testovat, a nakonec implementovat a integrovat.

Dříve pokud vývojář chtěl vyvinout multiplatformní aplikaci, bylo možné použít pouze malou část kódu, jelikož každá platforma využívala svůj vlastní kód. Dnes je to možné díky multiplatformním enginům, frameworkům, nástrojům a programovacím jazykům. Nicméně i tyto kódy je nutné upravovat ale již ne tak velkou část.

(SearchMicroservices, 2017)

5.1 Vývojové nástroje

5.1.1 Software development kit

Software development kit (SDK) je souhrn nástrojů, které se využívají během vývoje aplikace pro konkrétní zařízení, platformu anebo konkrétní funkce. Tyto nástroje obsahují integrované vývojové prostředí, které obsahuje potřebné prostředky a knihovny pro vývojáře. SDK obsahuje například vizuální editor, který umožňuje upravovat grafické uživatelské prostředí aplikace, kompilátor na vytváření aplikace ze zdrojového kódu a mnoho dalších funkcí, které se liší dle specifikací. Různé verze SDK obsahují vzorový kód, který slouží k tomu, aby se mohly vývojáři seznámit s funkcemi daného nástroje ale i případnou dokumentaci, kde je možné dohledat potřebné informace. Vývojové nástroje jsou poskytovány často zdarma, a to z toho důvodu, aby byl zvednut zájem vytváření aplikací pro danou platformu, které tyto nástroje patří.

(Techterms, 2010)

5.1.2 iOS SDK

Vývojové nástroje iOS SDK obsahují rozhraní a nástroje pro vývoj instalací, spouštění a testování. Aplikace lze psát pomocí systémových frameworků iOS anebo také pomocí programovacího jazyka C#. Součástí je i integrované vývojové prostředí pro správu grafiky a projektů, uživatelského rozhraní anebo nástroje pro debugování a ladění.

Další součástí je simulátor iOS, který obsahuje veškeré dokumenty a knihovny, pomocí kterého je možné aplikace testovat na počítačích. (Techopedia, 2014)

5.1.3 Java development kit

Java development kit je od společnosti Oracle, a je pro vývoj Java aplikací. Tento nástroj obsahuje nástroj JRE (Java Runtime Environment), který slouží ke spuštění Java aplikací, dále tento nástroj obsahuje kompilátor Javac, dokumentaci, archivátor a další nástroje které jsou potřebné k vývoji. JDK je dostupný na Windows, Linux a Solaris, každý z těchto operačních systémů má svou verzi nástroje. (Techopedia, 2011)

5.1.4 Android Debug Bridge

ADB je univerzální nástroj příkazového řádku, který umožňuje komunikovat s mobilním zařízením, které má operační systém android a funguje na bázi klient-server. Tento nástroj se používá k instalaci a ladění aplikací a přistupuje k prostředí Unix ve kterém je možné zadávat příkazy pro zařízení android. ADB se skládá ze tří hlavní částí a to klient, server a daemon. Klient se stará o odesílání příkazů. Tato část je spuštěná na počítači a je možné ji spustit příkazem „ADB“ v příkazovém řádku. Daemon spravuje příkazy na zařízení, tento proces je spuštěn na pozadí na všech zařízeních. Poslední částí ADB je server, ten spravuje komunikaci mezi klientem a daemonelem, server je spuštěn na počítači na pozadí. Zařízení a počítač jsou propojeny prostřednictvím USB kabelu, na zařízení je dále nutné mít spuštěný vývojářský režim a v něm funkci debugování pomocí USB. (Developer Android, 2018)

5.2 Mobilní hry

Mobilní hry jsou hry, které se hrají na mobilních telefonech, nejstarší hra na mobilní telefon je varianta Tetrisu z roku 1994, v roce 1997 byla společností Nokia vyvinuta a před instalována do většiny mobilních telefonů hra Had (Snake), ta patří mezi nejnámější mobilní hry na světě. V roce 1999 byla v roce zavedena platforma I-mode od společností NTT DoCoMo, jednalo se o první platformu, odkud bylo možné stahovat hry. Ve zbytku světa stále fungoval režim stahování prostřednictvím telekomunikační sítě. První velká možnost odkud bylo možné hry stahovat byla až v červenci 2008, kdy společnost Apple vydala App Store pro iOS tři měsíce na to byl spuštěn i obchod

Google Play pro mobilní zařízení Android. Nyní bylo možné prostřednictvím těchto obchodů publikovat hry pro širší veřejnost, to vyvolalo konkurenci v herním průmyslu a vedlo k rozvoji zajímavých her a různých žánrů. (Handjam, 2018)

5.2.1 Desková hra

Desková hra je hra, která se odehrává na herní ploše, na kterou lze umisťovat předměty, umístění záleží na pravidlech dané hry může se jednat o náhodné umístění anebo předem definované. Tyto hry jsou založené na strategii nebo pouze o štěstí, často je to kombinace obou těchto faktorů. Cíle bývají porážka protihráčů, dosažení místa na herní ploše. Cíle mohou být ale i jiné, záleží jaký typ hry to je. (MagicBoardGames, 2016)

5.2.2 Věžovky

Věžovka je strategická hra, ve které má hráč za úkol ubránit pevnost, poklad nebo jakékoliv jiné území před nepřáteli. Hráč se brání tím, že staví okolo trasy na předdefinované místo věže, které automaticky zaměřují a střílí na nepřátele. Věže mívají různé vlastnosti jako délku dostřelu, rychlost střelby, poškození atd. Nepřátelé mívají různou rychlost a životy. (GiantBomb, 2013)

6 Vytváření aplikace – praktická část

6.1 Základní informace o hře a vývoji

Hra je ve stylu věžovek, má hráč za cíl ubránit poklad před nepřítelem, k tomu mu pomáhají tři druhy věží, dvě jsou magické a jedna je balista, věže může hráč pokládat pouze na předem dané místo (platformy). Dále hráč může využít postavu, kterou je možné využít jen jednou za kolo, a to pouze na několik sekund, tu je možné umístit pouze na několik míst.

Nepřátelé mají různou rychlost pohybu a životy. Nepřátelé chodí ve vlnách o různých velikostech a pohybují se po předem stanovených trasách. V případě, že hráč nezničí nepřítele, než dojde k pokladu je mu ubrán jeden život, počet životů se mění v závislosti na úrovni.

Hra je zaměřena na mobilní zařízení s operačním systémem Android, hra je i kompatibilní s operačním systémem Windows 10 a Windows 8 pro stolní počítače. Veškeré skripty jsou napsané v objektově orientovaném jazyce C#, defaultní struktura skriptů obsahuje metody start a update. Metoda se spustí v momentě, kdy je načtený kód. Může sloužit k prvotnímu nastavení hodnot. Metoda update se spouští jednou za rámeček a běží nepřetržitě, pokud je kód spuštěn.

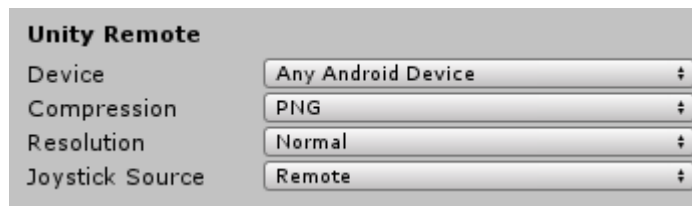
Kód byl psán v softwaru Visual Studio Community 2017 verze 15.9.1 a vývoj hry probíhal ve verzi Unity 2017.4.0f1.

6.2 Debuggování a zkoušení hry pomocí Unity remote

Debuggování větší části hry bylo možné provádět prostřednictvím okna „Game View“ dále jen „Game“ přímo v Unity, nicméně bylo nutné hru průběžně testovat a kontrolovat i na mobilním zařízení. K tomu byla využita aplikace Unity remote 5, která je dostupná v obchodě Google play zdarma.

Po stáhnutí aplikace bylo nutné v telefonu zapnout možnosti pro vývojáře a možnost ladění pomocí USB. Dalším krokem bylo v Unity na stolním počítači nastavit propojení s mobilním zařízením, v tomto nastavení je možné nastavit způsob ovládání a kvalitu, příklad nastavení je možné vidět na (Obrázek 13).

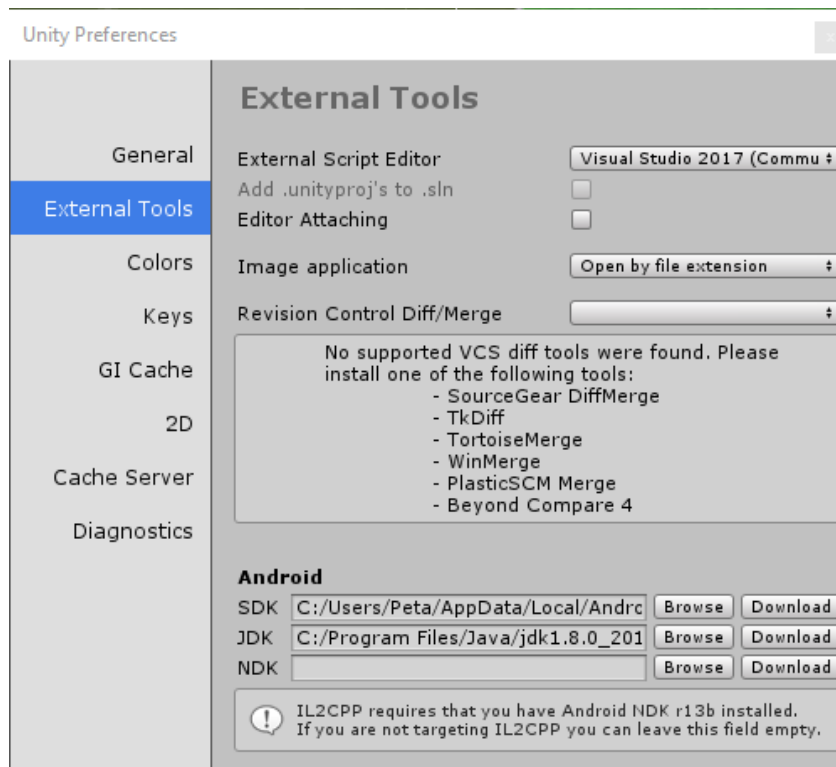
Obrázek 13: Nastavení editoru pro testování pomocí unity remote



(zdroj: Autor)

Dále bylo nutné stáhnout vývojové balíčky Java development kit, Software development kit a Android Debug Bridge. Tyto balíčky je možné stáhnout pomocí odkazů které jsou přímo v Unity. K těmto odkazům možné se dostat přes Edit/Unity Preferences/External Tools. Cesty a zobrazení okna je možné vidět na (Obrázek 14).

Obrázek 14: Náhled okna Unity preferences a cesty pro vývojové nástroje.



(zdroj: Autor)

6.3 Nahrání hry na Unity cloud

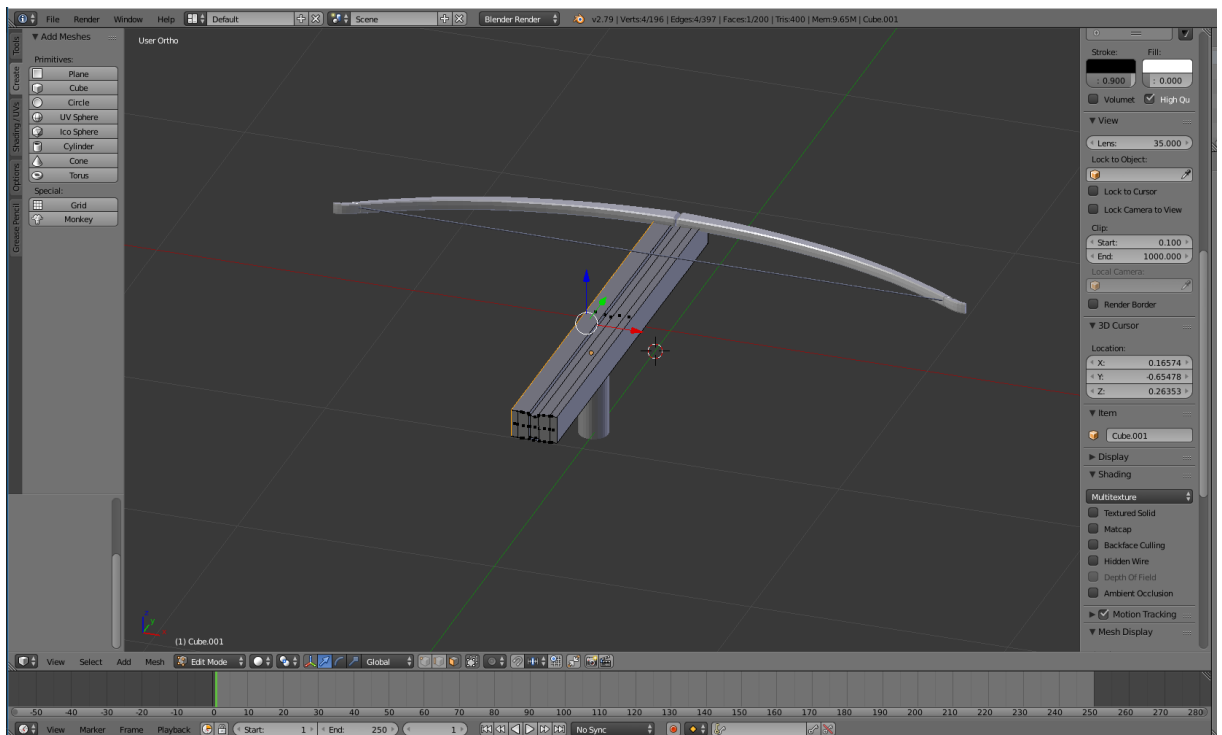
Nahrání projektu Unity cloud byl důležitým krokem, jelikož jsem cestoval mezi domovem a kolejí. Doma jsem pracoval na stolním počítači a na koleji na notebooku. Nahrání na cloud mi velice ulehčilo práci na projektu, z počátku jsem totiž hru měl na USB 3 flash disku, i když měl flash disk čtecí rychlost kolo 90 MB/s a zápisovou

rychlost okolo 50 MB/s byla zde poměrně velká odezva při spuštění. Zároveň cloud fungoval jako forma zálohy. Nahrání projektu na Unity cloud je jednoduché, a to pouhým zapnutím služby Collaborate zmíněnou v páté kapitole.

6.4 Tvorba modelů

Pro hru bylo vytvořeno několik jednoduchých modelů, a to dvě věže a platforma na umístění věží. Modely byly vytvořeny pomocí modelovacího softwaru Blender. Tyto modely bylo poté možné jednoduše importovat do Unity, kde byla nastavena jejich barva a další vlastnosti. Vývoj balisty (Obrázek 15) trval pár hodin, jelikož mé znalosti v tomto programu jsou na úrovni začátečníka.

Obrázek 15: Vývoj modelu pro balistu v modelovacím softwaru Blender.

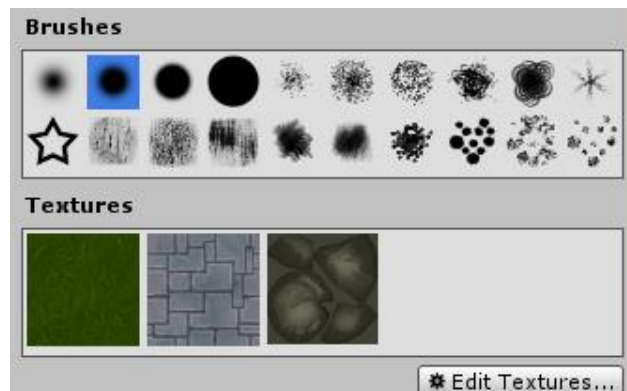


(zdroj: Autor)

6.5 Vytváření prostředí – levelů

Nejdříve je nutné vytvořit herní plochu, kde se budou nepřátelé pohybovat a kam budeme umístit věže. Vytváření prostředí je pouze grafická část. Prostředí bylo navrženo pomocí herního objektu pro terén (Terrain), Pomocí štětce na textury (Obrázek 16) byla vytvořena cesta.

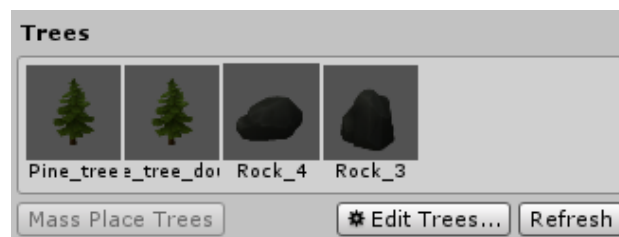
Obrázek 16: Nastavení štětce a volba textur



(zdroj: Autor)

Po vytvoření cesty přišlo na řadu umístění stromů a kamenu, k tomu byla využita funkce pro hromadné umístění stromů (Obrázek 17)

Obrázek 17: Import stromů a kamenů do herního objektu terrain.



(zdroj: Autor)

Další krok bylo umístění vymodelovaných platforem, místa odkud budou vycházet nepřátelé a pokladu, který je nutné ochránit. Náhled na první úroveň je na ()Příloha 2.

6.6 Pohyb kamery

Pro pohyb kamery byl sepsán skript, který detekuje kolik prstů/vstupů je momentálně na dotykovém displeji prostřednictvím funkce `Input.touchCount`. V případě, že funkce detekovala pouze jeden prst, je možné se po herní ploše pohybovat do všech stran. Je k tomu využito pouze dvourozměrný vektor, jelikož vektor do hloubky nás při tomto nezajímá. Pozice prstu je periodicky zaznamenávána pomocí funkce `deltaposition`. V případě, že je jsou detekovány dva prsty, skript uloží pozice obou prstů a podle toho, zda se prsty přibližují od sebe anebo k sobě, je obraz přibližován nebo oddalován. Pro pohyb do stran, přibližování a oddalování jsou nastaveny limity. Příklad kódu, který řeší pohyb do stran a nastavené limity je možné vidět na (Obrázek 18).

Obrázek 18: Část kódu pro pohyb kamery.

```
(Input.touchCount == 1 && Input.GetTouch(0).phase == TouchPhase.Moved)

Vector2 deltatouchmovefirst = Input.GetTouch(0).deltaPosition;
transform.Translate(-deltatouchmovefirst.x * rychlostcamery, -deltatouchmovefirst.y * rychlostcamery, 0);

transform.position = new Vector3 (
    Mathf.Clamp(transform.position.x, 0f, 100f),
    Mathf.Clamp(transform.position.y, 80f, 90f),
    Mathf.Clamp(transform.position.z, 0f, 70f));
```

(zdroj: Autor)

6.7 Nepřátelé

Ve hře jsou tři druhy nepřátel, každý má svou vlastní rychlost a počet životů. Modely nepřátel (Obrázek 19) jsou získané z Unity Asset Store balíčku. Nepřátelé mají své vlastní označení (tag), pomocí kterého jsou vyhledávány věží.

Obrázek 19: Modely nepřátel použité ve hře.



(zdroj: Autor)

6.7.1 Pohyb nepřátel

Nepřátelé se pohybují po mapě pomocí bodů nebo takzvaných „waypointů“ (Obrázek 20). Ty jsou umístěny přímo na trase jsou viditelné pouze ve scéně, na trase jsou umístovány do rohů, jelikož je v tento moment nutné změnit směr.

Obrázek 20: Příklad waypointu, který je umístění na herní ploše.



(zdroj: Autor)

Waypointy jsou ve scéně pod jedním herním objektem, kterému je přiřazen skript, pomocí kterého jsou tyto body načteny do pole. Toto pole je poté využíváno ve skriptu pro nepřátele. V tomto skriptu je řešeno veškeré chování a vlastnosti nepřátel (Obrázek 21). Rychlost představuje rychlost pohybu po scéně, Minrychlost udává minimální rychlost, pod kterou není možné ji snížit. Zivotyemem představují životy nepřátel.

Obrázek 21: Zobrazené vlastnosti nepřátel v okně Inspector.

Rychlost	10
Minrychlost	5
Zivotyemem	20
Text	HPBar (Rect Transform)
Hpbar	HP (Image)

(zdroj: Autor)

Animace po mapě je řešena pomocí funkce LookAt(), kde je jako parametr předáván nejbližší waypoint, ke kterému se nepřítel blíží. Nepřátelé mají přidělené i vizuální zobrazení bodů životů pomocí hitpoints baru (Obrázek 22), ten představuje poměr životů na začátku a momentální stav životů, skript bylo nutné upravit kvůli rotaci, kterou způsobovala funkce LookAt(), v momentě totiž kdy se nepřítel podíval na waypoint byl natočen i hitpoints bar. Červená barva na hitpoints baru představuje ubrané životy. V momentě, kdy životy klesnou na 0 je nepřítel zničen.

Obrázek 22: Zobrazení modelu nepřítele s aktivním hitpoints barem.



(zdroj: Autor)

V případě, že nepřítel dojde až na konec jsou hráči ubráný životy a nepřítel je poté zničen, poslední waypoint představuje poklad, který má své vlastní označení, část kódu pro ubrání životu hráči je na (Obrázek 23). Nepřátelé jsou do hry pomocí vln, v každé vlně je různý počet stejného nepřítele. Mezi vlny je nastavený odpočet. Vlny jsou

nastaveny tak, aby se nespustila další do té doby, než jsou všichni nepřátelé z předchozí vlny zničeni.

Obrázek 23: Náhled na část kódu pro ubraní životů hráči.

```
void DmgToPlayer ()
{
    player.Zivoty--;
    WavesSpawner.Nepratelenazivu--;
    Destroy(gameObject);
}
```

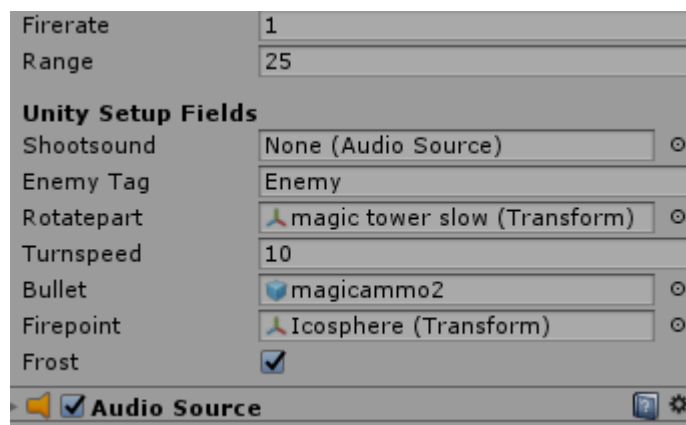
(zdroj: Autor)

6.8 Věže

Ve hře jsou zavedeny tři různé druhy věží. Každá věž má jinou délku dostřelu, rychlost střelby ale i poškození. Na (Obrázek 24) je možné vidět nastavení ledové věže.

Poškození je udáváno podle typu munice. Jeden typ věže je speciální tím, že snižuje rychlost nepřítelům. Věže jsou rozděleny na tři hlavní části. Jedná část je fixní, tato část je jen podstavec. Další část je pohyblivá část, která se otáčí na nepřátele funkcí LookAt(). Třetí část je bod odkud je vystřeleno.

Obrázek 24: Vlastnosti věží zobrazené v okně Inspektor.

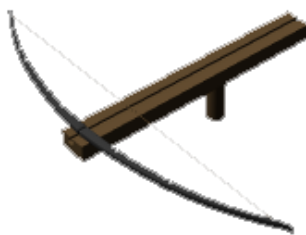


(zdroj: Autor)

6.8.1 Druhy věží

Balista (Obrázek 25) je velká kuše, která má velký dostřel a poškození, její slabinou je ale pomalá rychlost střelby.

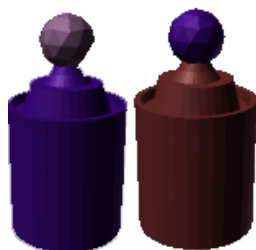
Obrázek 25: Model balisty použitý ve hře.



(zdroj: Autor)

Magická ledová věž a standartní magická věž (Obrázek 26) má speciální vlastnost zpomalení nepřátel, každým výstřelem je snížena rychlost nepřátelům o 1, tuto hodnotu je možné změnit dle potřeby, část kódu na zpomalení je na obrázku (Obrázek 27). Její slabinou je malé poškození, pomalá střelba a malý dostřel. Standartní magická věž má veliké poškození a

Obrázek 26: Magická a standartní magické věže.



(zdroj: Autor)

Obrázek 27: Náhled na kód pro zpomalení nepřátel.

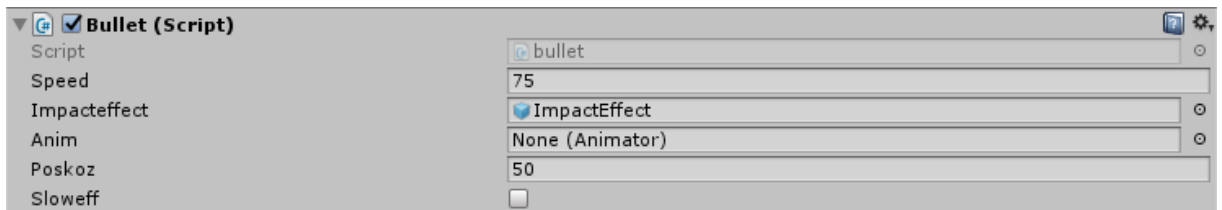
```
nepr.Poskozeni(poskoz);  
if (sloweff)  
{  
    nepr.Zpomaleni(1);  
}
```

(zdroj: Autor)

Věže zaměřují pomocí toho, zda je v jejich dosahu herní objekt, který je označen jako nepřítel. Vzdálenost mezi věží a nepřáteli je vypočítáván převodem hodnot vektoru, tato hodnota je neustále aktualizovaná v momentě kdy se hodnota dostane pod stanovené maximum, věž vystřelí, pokud je nepřítel zničen kód se vrátí na začátek, jelikož se může stát, že se věž zasekne. Věže mají zavedený i zvuk, který je spuštěn v momentě kdy vystřelí.

Vlastnosti munice (Obrázek 28) se nastavují v samostatném skriptu, v tomto skriptu je možné nastavit poškození, rychlost střely, efekt, jaký bude mít střela a zda bude zpomalovat. Poškození je napojené na skript nepřátel, jelikož hodnota života je nastavena jako veřejná, tudíž je možné k této hodnotě přistupovat i z jiných skriptů. Každá věž má svůj vlastní model střely, který je odlišný barvou a ladí k barvě věži. Příklad kódu pro poškození je na obrázku (Obrázek 29).

Obrázek 28: Vlastnosti munice věží.



(zdroj: Autor)

Obrázek 29: Část kódu využitá během střelby.

```
GameObject bulletfire = (GameObject)Instantiate(bullet, firepoint.position, firepoint.rotation);
bullet bulletobj = bulletfire.GetComponent<bullet>();
Shootsound.Play();
```

(zdroj: Autor)

Věže je možné umisťovat na platformy (Obrázek 30, Příloha 3). Pro umisťování věží je skript, ve kterém jsou definované prefaby věží. Původně měla být funkce umisťování věží na platformu pomocí detekce vstupů na dotykovém displeji. V novějších verzích Unity, už je ale možné použít funkci OnMouseDown(), která detekuje levý klik myši stejně jako jednoduché klepnutí na displej. Prostřednictvím této funkce bylo velice usnadněno sepsání kódu pro umístění věže. Během umístění věže bylo také nutné hlídat, zda už není věž na této platformě.

Obrázek 30: Platforma pro umístění věží

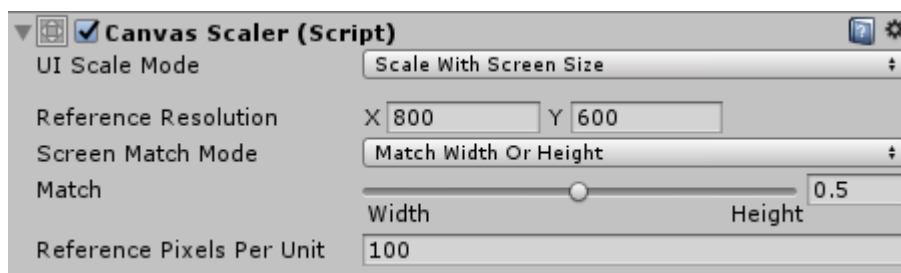


(zdroj: Autor)

6.9 Uživatelské rozhraní

Uživatelské rozhraní se spouští v závislosti na průběhu hry. Uživatelské rozhraní je řešeno pomocí herního objektu Canvas, v tomto objektu jsou poté umístěny všechny prvky uživatelského rozhraní. Uživatelskému rozhraní je možné přidávat různé komponenty a různě nastavovat. Pomocí funkce Canvas Scaler, (Obrázek 31) bylo nastaveno to, aby se uživatelské rozhraní zobrazovalo na zařízeních stejně, další použítá komponenta bylo pro rozložení tlačítek.

Obrázek 31: Komponenty pro správné škálování v závislosti na rozlišení obrazovky.

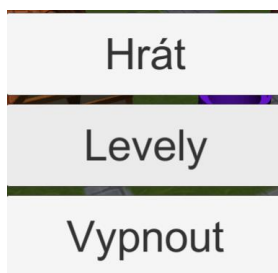


(zdroj: Autor)

6.9.1 Hlavní menu

Hlavní menu (Obrázek 32, Příloha 4) je prvotní scéna při zapnutí hry. V tomto okně může hráč vybrat, zda chce začít hrát, v tomto případě se spustí hra od začátku, případně zda chce vybrat specifickou úroveň anebo zda chce hru úplně vypnout.

Obrázek 32: Část uživatelského rozhraní pro hlavní menu.



(zdroj: Autor)

Funkce tlačítek jsou řešeny samostatným skriptem, pomocí kterého se načítají různá okna a scény, řešení menu je na (Obrázek 33). Načítání scén je možné pomocí indexu scén, který je nastaven v buildovacím okně, nebo přímo jménem scény. Vypnutí aplikace probíhá bez potvrzení.

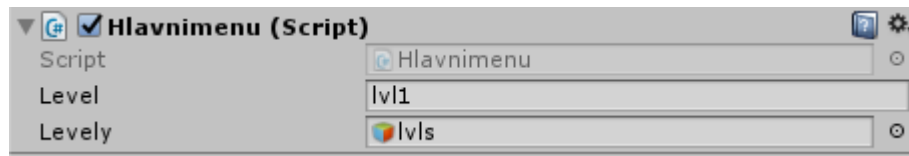
Obrázek 33: Část kódu využitá pro tlačítka uživatelského rozhraní.

```
public void Play()
{
    SceneManager.LoadScene("lv11");
}
public void Levels ()
{
    levely.SetActive(true);
}
public void Exit ()
{
    Application.Quit();
}
```

(zdroj: Autor)

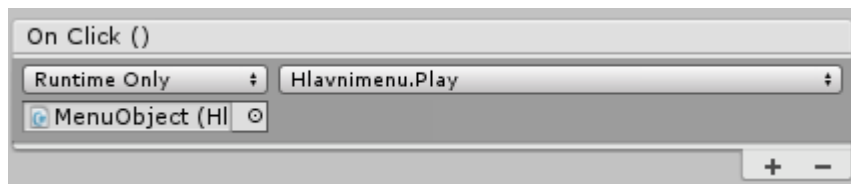
Skript na menu je nutné propojit s herním objektem (Obrázek 34), který menu spustí, poté je nutné hernímu objektu přiřadit canvas a poté tlačítkům přiřadit funkce, které jsou ve skriptu (Obrázek 35)

Obrázek 34: Propojení herního objektu se skriptem pro spuštění Canvasu.



(zdroj: Autor)

Obrázek 35: Nastavení tlačítek a přidělení funkcí, tlačítkům pomocí skriptu.



(zdroj: Autor)

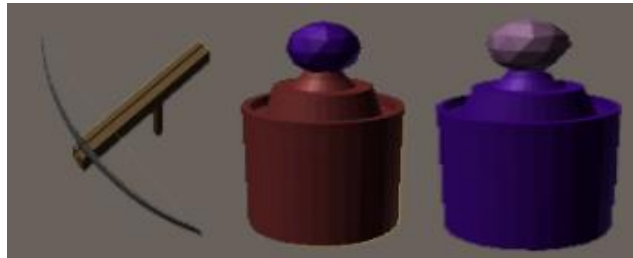
6.9.2 Zvolení úrovně

Do tohoto uživatelského rozhraní se hráč z hlavního menu, zde je možné vybrat různé úrovně.

6.9.3 Stavění věží

Výběr věže může hráč provést pomocí uživatelského rozhraní, které je na spodní části obrazovky, má, zda na výběr ze tří věží, ukázka na (Obrázek 36). Uživatelské rozhraní pro stavění věží je propojené se skriptem, který je spravuje.

Obrázek 36: Uživatelské rozhraní pro výběr věže.



(zdroj: Autor)

6.9.4 Úroveň hotová

Toto uživatelské rozhraní se spustí v případě, že počet zbývajících vln nepřátel klesl na nulu není žádný naživu. V této nabídce má hráč na výběr spuštění hry znovu, návrat do menu anebo další úroveň. Jelikož jdou scény po sobě, je možné spustit další scénu pouze tím, že k indexu momentálně spuštěné scény přidáme hodnotu 1. Náhled kódu je na (Obrázek 37).

Obrázek 37: Náhled na kód, který je využit v rozhraní pro splnění úrovně.

```
public void Reload()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
}

public void Menu()
{
    SceneManager.LoadScene("MenuMain");
}

public void NextLvl()
{
    SceneManager.LoadScene(currentlvl);
}
```

(zdroj: Autor)

6.9.5 Konec hry

Uživatelské rozhraní je napojené na skript, ve kterém jsou uloženy informace o hráči. V momentě, kdy životy hráče klesnou na 0 jsou všechny ostatní skripty vypnuté a hráč může pouze hru spustit znovu nebo se vrátit do hlavního menu

6.9.6 Pauza

Pauza (Příloha 1) je vyvolána tlačítkem na klávesnici „ESC“ nebo pomocí tlačítka zpět na mobilním zařízení. V momentě, kdy je vyvolána pauza, je všechen čas ve hře pozastaven až do jeho obnovení. V uživatelském rozhraní pauza má hráč na výběr buďto navrátit se do hry, restartovat hru, jít do menu anebo hru vypnout. Kód na vypnutí času ve hře je na (Obrázek 38).

Obrázek 38: Náhled na kód pro zastavení času.

```
if (pauzaokno.activeSelf)
{
    Time.timeScale = 0f;
}
else
{
    Time.timeScale = 1f;
}
```

(zdroj: Autor)

6.9.7 Životy a odpočet

V tomto okně je možné vidět životy a zbývající odpočet do nové vlny nepřátel. Příklad zobrazení je na (Obrázek 39)

Obrázek 39: Uživatelské rozhraní pro zobrazení času do další vlny a počet zbývajících životů.

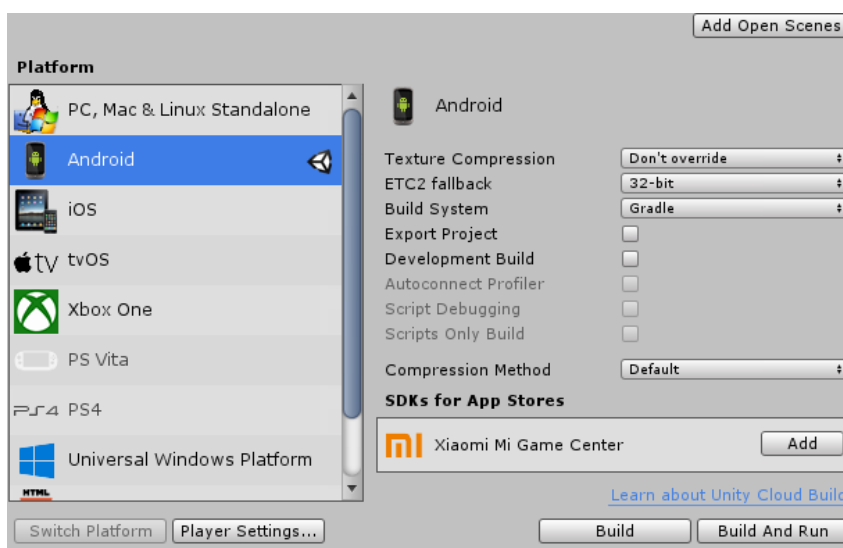


(zdroj: Autor)

6.10 Kompilace hry

Kompilovat hru je možné pomocí okna „Build Setings“ (Obrázek 40) v tomto okně je možné vložit různé scény, které chceme, aby byly ve finální aplikaci. Zvolit platformu kompilace.

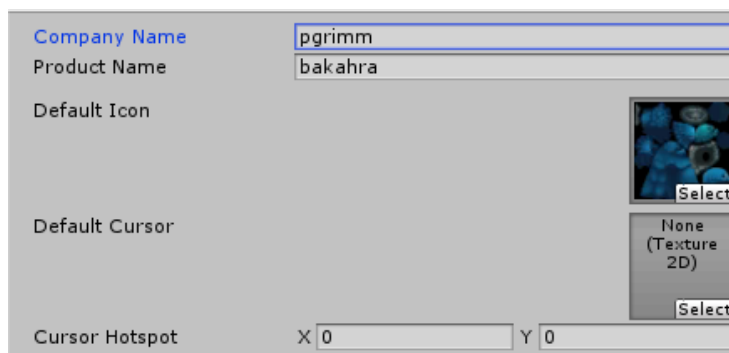
Obrázek 40: Okno s nastavením kompilace.



(zdroj: Autor)

Pomocí tlačítka „player settings“ (Obrázek 41) je možné se dostat do nastavení, kde se nastaví ikona hry, jméno hry, verze, identifikace společnosti, která hru vydává, minimální verze androidu, podporované ovládání a mnoho dalšího.

Obrázek 41: Okno Player settings, kde je možné nastavit jméno aplikace, ikonu a mnoho dalšího.



(zdroj: Autor)

6.11 Problémy během testování

Během testování se vyskytl problém hned na začátku, a to v tom, že na telefonu se nic nezobrazovalo. V Unity není nikde potvrzení, že je telefon správně připojen nebo zda je vůbec připojen. Nastala fáze pokus omyl, na fórech unity bylo několik možných řešení. Žádné z nich ale nefungovalo. Po několika hodinovém trápení se mi podařilo konečně najít řešení. Bylo nutné stáhnout program Android Studio, pomocí tohoto programu

ovladače Google USB. V telefonu přepnout z režimu přenos dat na režim přenos Fotek (PTP) tento krok je ale různých výrobců mobilních zařízení jiný, testovací zařízení bylo Xiaomi Redmi A5 Prime. Hlavní ale bylo udělit výjimku v antivirovém programu Avast pro nástroj ADB, ten byl totiž pokaždé přesunut do virové truhly bez ohlášení. Nakonec se mi podařilo zjistit, že pomocí ADB lze ověřit, zda je připojené zařízení. Příklad zobrazení připojení je možné vidět na (Obrázek 42).

Obrázek 42: Okno s puštěním příkazem ADB, které detekuje připojené zařízení.

```
List of devices attached
adb server version (36) doesn't match this client (40); killing...
* daemon started successfully
d56f44 device
emulator-5554 device
```

(zdroj: Autor)

Během buildování se vyskytlo také několik problémů, první problém byl v tom, že defaultní cesta k software development kitu byla špatná, dalším problémem s SDK bylo to, že nejnovější verze neobsahovala některé nutné nástroje, které potřebuje Unity, to bylo vyřešeno pomocí programu Android Studio, ten má v nastavení funkci pro instalaci SDK a jeho nástrojů. Další problém byl s Java development kitem, jelikož nejnovější verze 11.0.2 také nechtěla správně fungovat, takže dle doporučení bylo nutné stáhnout starší verzi 8u202, která naštěstí fungovala.

7 Závěr

V bakalářské práci byl čtenář seznámen se základními prvky operačního systému pro mobilní zařízení, získal základní informace ohledně různých druhů operačních a byl seznámen s vývojovým prostředím Unity a vývojovými nástroji, které se při vývoji aplikací používají.

V praktické části byl čtenář seznámen s vývojem herní aplikace v prostředí Unity, se zaměřením pro operační systém Android. Hru je možné bez chyb zkompileovat a hrát na mobilních zařízeních se všemi uvedenými funkcemi.

S hrou jsem víceméně spokojený, jsem si vědom několika drobných chyb, které je možné snadno opravit, bohužel k tomu nezbyl čas. Na hře mám v plánu stále pracovat a později ji publikovat na některém obchodě s aplikacemi.

8 Summary

This bachelor thesis focuses on the development of a mobile game application on the Android platform in the Unity 3D development framework with a usage of programming language C sharp. This thesis describes individual elements of a Unity 3D development framework, add-ons that can be obtained in the Unity Asset Store and usage of those add-ons during the development. It also describes the Software Development Kit and Java Development Kit that are needed to create the final Unity application.

The genre of this game is tower defence for single player, where the player must defend treasure using the towers around the route. The player can choose from different types of towers which can be upgraded to stronger towers to stop multiple types/species of enemies. The game includes multiple levels with different difficulties. Gameplay is later tested by a small group of people.

Keywords: development, Unity, Android, game, application, C sharp

9 Bibliografie

- AndroidMarket, . (2011). Jak vypadá Android uvnitř? [Online]. *Androidmarket*.
Androidmarket. Retrieved from <https://androidmarket.cz/android/jak-vypada-android-uvnitř-aneb-co-je-rom-kernel-bootloader-a-dalsi/>
- Beal, V. (2001). What is Library? [Online]. *Webopedia*. Webopedia. Retrieved from <https://www.webopedia.com/TERM/L/library.html>
- Beal, V. (2011). Mobile Operating Systems (Mobile OS) Explained [Online].
webopedia. webopedia. Retrieved from https://www.webopedia.com/DidYouKnow/Hardware_Software/mobile-operating-systems-mobile-os-explained.html
- Beal, V. (2017). What is Kernel? [Online]. *What is Kernel (Operating System Kernel)? Webopedia Definition*. Webopedia. Retrieved from <https://www.webopedia.com/TERM/K/kernel.html>
- Blender, . (2007). About [Online]. *Blender*. Blender. Retrieved from <https://www.blender.org/about/>
- Brodkin, J. (2013). How Unity3D Became a Game-Development Beast [Online].
insights dice. insights dice. Retrieved from <https://insights.dice.com/2013/06/03/how-unity3d-become-a-game-development-beast/>
- Computer Hope, . (2017). What is a Framework [Online]. *Computer Hope*. Computer Hope. Retrieved from <https://www.computerhope.com/jargon/f/framework.htm>
- Computer Hope, . (2019). What is ROM (Read-only Memory)? [Online]. *computerhope*. Computer Hope. Retrieved from <https://www.computerhope.com/jargon/r/rom.htm>
- Crecente, B. (2018). Candy Crush Saga' Earned Just Under \$1 Billion in Past 12 Months [Online]. *Variety*. Variety. Retrieved from <https://variety.com/2018/gaming/news/candy-crush-saga-1-billion-1202908004/>
- Denham, T. (2019). What is Unreal Engine? [Online]. *conceptartempire..* conceptartempire. Retrieved from <https://conceptartempire.com/what-is-unreal-engine/>

Developer Android, . (2018). Android Debug Bridge [Online]. *Android Debug Bridge*. developer android. Retrieved from <https://developer.android.com/studio/command-line/adb>

Dietrich, E., & Smacchia, . (2017). The history of C# [Online]. *Microsoft*. Microsoft. Retrieved from <https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history>

Dillet, R. (2018). Unity CEO says half of all games are built on Unity – TechCrunch [Online]. *TechCrunch*. TechCrunch. Retrieved from https://techcrunch.com/2018/09/05/unity-ceo-says-half-of-all-games-are-built-on-unity/?guccounter=1&guce_referrer_us=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_cs=y17YhNCHnV6Tj-R4uDpnhg

Dr. Garbade, M. (2018). Top 3 most popular programming languages in 2018 (and their annual salaries) [Online]. *hackernoon*. hackernoon. Retrieved from <https://hackernoon.com/top-3-most-popular-programming-languages-in-2018-and-their-annual-salaries-51b4a7354e06>

EPIC GAMES, . (2015). Blueprint Editor Reference [Online]. *unrealengine*. Epic Games. Retrieved from <https://docs.unrealengine.com/en-US/Engine/Blueprints/Editor>

Fogel, S. (2019). Pokémon Go' Global Revenue Grew 37% in 2018 (Analyst) [Online]. *Variety*. Variety. Retrieved from <https://variety.com/2019/gaming/news/pokemon-go-global-revenue-2018-1203098512/>

GiantBomb, . (2013). Tower Defense Compect [Online]. *GiantBomb*. GiantBomab. Retrieved from <https://www.giantbomb.com/tower-defense/3015-413/>

Handjam, . (2018). The History of Mobile Gaming [Online]. *The History of Mobile Gaming*. Handjam. Retrieved from <http://handjam.co.uk/the-history-of-mobile-gaming/>

Jain, S. (2014). Android System Architecture [Online]. *Android Tutorial Horizon*. Androd Tutorial Horizon. Retrieved from <https://android.tutorialhorizon.com/android-system-architecture/>

MagicBoardGames, . (2016). Board Game Introduction - MagicBoardGames.com [Online]. *Board Game Introduction - MagicBoardGames.com*. MagicBoardGames. Retrieved from <http://www.magicboardgames.com/index.php>

openhandsalliance, . (2007). Android Overview | Open Handset Alliance [Online]. *openhandsalliance*. openhandsalliance. Retrieved from http://www.openhandsalliance.com/android_overview.html

PCMAG, . (2013). Graphics [Online]. *graphics Definition from PC Magazine Encyclopedia*. pcmag. Retrieved from <https://www.pcmag.com/encyclopedia/term/43916/graphics>

PCMAG, . (2013). 2D graphics definition [Online]. *pcmag*. pcmag. Retrieved from <https://www.pcmag.com/encyclopedia/term/36990/2d-graphics>

PCMAG, . (2013). 3D graphics Definition [Online]. *pcmag*. 2013. Retrieved from <https://www.pcmag.com/encyclopedia/term/37072/3d-graphics>

Petty, J. (2019). What is 3D Modeling & What's It Used For? [Online]. *conceptartempire*. conceptartempire. Retrieved from <https://conceptartempire.com/what-is-3d-modeling/>

SearchMicroservices, . (2017). What is mobile application development? [Online]. *What is mobile application development?*. SearchMicroservices. Retrieved from <https://searchmicroservices.techtarget.com/definition/mobile-application-development>

Serych, J. (2007). Začínáme s dsPICem (3) - bootloader [Online]. *jakub.serych*. Jakub Serych. Retrieved from <http://jakub.serych.cz/book/export/html/5>

Spence, E. (2014). The Vital And Depressing Lessons Flappy Bird Can Teach Indie Developers [Online]. *Forbes*. Forbes. Retrieved from <https://www.forbes.com/sites/ewanspence/2014/02/18/the-vital-and-depressing-lessons-flappy-bird-can-teach-indie-developers/#173234a43792>

StatCounter, . (2017). Mobile Operating System Market Share Worldwide | StatCounter Global Stats [Online]. *Statcounter*. Statcounter. Retrieved from <http://gs.statcounter.com/os-market-share/mobile/worldwide>

Stroud, F. (2018). What is Windows Phone? Webopedia Definition [Online]. *Webopedia*. Webopedia. Retrieved from https://www.webopedia.com/TERM/W/windows_phone.html

Techopedia, . (2011). What is Java Development Kit (JDK)? - Definition from Techopedia [Online]. *What is Java Development Kit (JDK)? - Definition from Techopedia*. Techopedia. Retrieved from <https://www.techopedia.com/definition/5594/java-development-kit-jdk>

Techopedia, . (2011). What is iOS? [Online]. *What is iOS?*. Techopedia. Retrieved from <https://www.techopedia.com/definition/25206/ios>

Techopedia, . (2013). What is a Boot Loader? [Online]. *techopedia*. 2013: Techopedia. Retrieved from <https://www.techopedia.com/definition/3324/boot-loader>

Techopedia, . (2014). What is the iOS SDK? - Definition from Techopedia [Online]. *What is the iOS SDK? - Definition from Techopedia*. Techopedia. Retrieved from <https://www.techopedia.com/definition/3819/ios-sdk>

Techopedia, . (2018). What is Android? - Definition from Techopedia [Online]. *Techopedia*. Techopedia. Retrieved from <https://www.techopedia.com/definition/5415/android>

Techterms, . (2010). SDK (Software Development kit) [Online]. *SDK (Software Development kit)*. TechTerms. Retrieved from <https://techterms.com/definition/sdk>

Techterms, . (2013). Framework Definition [Online]. *Tech Terms*. TechTerms. Retrieved from <https://techterms.com/definition/framework>

Unity, . (2006). Products - Unity [Online]. *Unity*. Unity. Retrieved from <https://unity3d.com/unity>

Unity, . (2015). Unity - Services [Online]. *Unity- Services*. Retrieved from <https://unity3d.com/services>

Unity, . (2017). Downloads - Public relations - Unity [Online]. *Unity*. Unity. Retrieved from <https://unity3d.com/public-relations/downloads>

Unity, . (2018). Games engines, how do they work? [Online]. *Games engines, how do they work?*. Unity technologies. Retrieved from <https://unity3d.com/what-is-a-game-engine>

Unity, . (2018). Unity - Manual: Scripting [Online]. *Unity manual*. Unity. Retrieved from https://docs.unity3d.com/Manual/ScriptingSection.html?_ga=2.173988512.281247759.1553525530-453986519.1553525530

Unity, . (2018). Unity - Manual: Game Objects - Unity [Online]. *Unity*. Unity. Retrieved from <https://docs.unity3d.com/Manual/GameObjects.html>

Unity, . (2018). Unity - Manual: Scenes [Online]. *Unity*. Unity. Retrieved from <https://docs.unity3d.com/Manual/CreatingScenes.html>

Unity, . (2018). Games made with the Unity engine - Unity [Online]. *Games made with the Unity engine - Unity*. Unity. Retrieved from <https://unity3d.com/games-made-with-unity>

Ward, J. (2008). What is a Game Engine [Online]. *GameCareerGuide*. gamecareerguide. Retrieved from http://www.gamecareerguide.com/features/529/what_is_a_game_.php?page=2

10 Seznam obrázků

Obrázek 1: Graf nejpoužívanějších programovacích jazyků	14
Obrázek 2: Podporované platformy Unity	18
Obrázek 3: Logo Unity	19
Obrázek 4: Podporované formáty pro import	19
Obrázek 5: Přehled dostupných služeb v Unity	21
Obrázek 6: Okno pro nahrání změn ve službě Collaborate	22
Obrázek 7: Způsob, jakým funguje aukce pro reklamy	22
Obrázek 8: Statisticky ze služby analytics	23
Obrázek 9: Okna pro scénu a hierarchii	25
Obrázek 10: Okno inspektor	26
Obrázek 11: Přehled platform, které podporuje Unreal Engine	26
Obrázek 12: Příklad blueprintingu	27
Obrázek 13: Nastavení editoru pro testování pomocí unity remote	34
Obrázek 14: Náhled okno Unity preferences a cesty pro vývojové nástroje	34
Obrázek 15: Vývoj modelu pro balistu v modelovacím softwaru Blender.	35
Obrázek 16: Nastavení štětce a volba textur	36
Obrázek 17: Import stromů a kamenů do herního objektu terrain.	36
Obrázek 18: Část kódu pro pohyb kamery.	37
Obrázek 19: Modely nepřátel použité ve hře	37
Obrázek 20: Příklad waypointu, který je umístění na herní ploše.	37
Obrázek 21: Zobrazené vlastnosti nepřátel v okně Inspector.	38
Obrázek 22: Zobrazení modelu nepřátele s aktivním hitpoints barem.	38

Obrázek 23: Náhled na část kódu pro ubraní životů hráči.....	39
Obrázek 24: Vlastnosti věží zobrazené v okně Inspektor.....	39
Obrázek 25: Model balisty použitý ve hře.....	40
Obrázek 26: Magická a standartní magické věže.....	40
Obrázek 27: Náhled na kód pro zpomalení nepřátel.....	40
Obrázek 28: Vlastnosti munice věží.....	41
Obrázek 29: Část kódu využitá během střelby.....	41
Obrázek 30: Platforma pro umístění věží.....	41
Obrázek 31: Komponenty pro správné škálování v závislosti na rozlišení obrazovky.....	42
Obrázek 32: Část uživatelského rozhraní pro hlavní menu.....	42
Obrázek 33: Část kódu využitá pro tlačítka uživatelského rozhraní.....	43
Obrázek 34: Propojení herního objektu se skriptem pro spuštění Canvasu.....	43
Obrázek 35: Nastavení tlačítek a přidělení funkcí, tlačítkům pomocí skriptu.....	43
Obrázek 36: Uživatelské rozhraní pro výběr věže.....	44
Obrázek 37: Náhled na kód, který je využit v rozhraní pro splnění úrovně.....	44
Obrázek 38: Náhled na kód pro zastavení času.....	45
Obrázek 39: Uživatelské rozhraní pro zobrazení času do další vlny a počet zbývajících životů.....	45
Obrázek 40: Okno s nastavením kompilace.....	46
Obrázek 41: Okno Player settings, kde je možné nastavit jméno aplikace, ikonu a mnoho dalšího.....	46
Obrázek 42: Okno s puštěním příkazem ADB, které detekuje připojené zařízení.....	47

11 Seznam tabulek

Tabulka 1: Podíl operačních systémů na trhu 16

Tabulka 2: Přehled dostupných verzí Unity a některé rozdíly..... 24

12 Seznam příloh

Příloha 1: Uživatelské rozhraní pro pauzu	ii
Příloha 2 Náhled na level	ii
Příloha 3: Pohled na scénu s označenými objekty	iii
Příloha 4: Hlavní menu	iii

Příloha 1: Uživatelské rozhraní pro pauzu



(zdroj: Autor)

Příloha 2 Náhled na level



(zdroj: Autor)

Příloha 3: Pohled na scénu s označenými objekty



(zdroj: Autor)

Příloha 4: Hlavní menu



(zdroj: Autor)