

Návrh a implementace způsobu pohybu v mobilní aplikaci pro virtuální realitu

Diplomová práce

Vedoucí práce:

Ing. Jaromír Landa, Ph.D.

Bc. Petr Jaroš

Brno 2017

Rád bych na tomto místě poděkoval Ing. Jaromíru Landovi, Ph.D. za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování diplomové práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Návrh a implementace způsobu pohybu v mobilní aplikaci pro virtuální realitu**

vypracoval/a samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom/a, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmetná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 22. května 2017

Abstract

Jaroš, P. Design and implementation of movement in mobile application for virtual reality. Bachelor thesis. Brno: Mendel University, 2017.

The diploma thesis focuses on the issue of movement and interaction in virtual reality and then shows the design and concrete example of the application.

Keywords

Android, application development, virtual reality, Google Cardboard, Unity.

Abstrakt

Jaroš, P. Návrh a implementace způsobu pohybu v mobilní aplikaci pro virtuální realitu. Diplomová práce. Brno: Mendelova univerzita v Brně, 2017.

Diplomová práce se zaměřuje na problematiku pohybu a interakce ve virtuální realitě a následně ukazuje návrh a konkrétní příklad aplikace.

Klíčová slova

Android, vývoj aplikací, virtuální realita, Google Cardboard, Unity.

Obsah

1	Úvod a cíl práce	13
1.1	Úvod	13
1.2	Cíl práce	13
2	Vývoj mobilních aplikací	14
2.1	Mobilní operační systémy.....	15
2.1.1	Windows Phone	16
2.1.2	iOS.....	17
2.1.3	Android.....	19
2.1.4	Ostatní systémy.....	21
2.2	Vývojová prostředí	22
2.2.1	Android Studio	23
2.2.2	XCode.....	23
2.2.3	Unity.....	24
2.3	Engine.....	24
2.3.1	Unity.....	25
2.3.2	Cocos2D.....	25
2.3.3	Corona.....	26
2.3.4	Unreal.....	26
3	Unity	28
3.1	Uživatelská přívětivost	28
3.2	Podpora platforem.....	28
3.2.1	Unity for mobile.....	29
3.2.2	Unity for VR and AR	29
3.2.3	Unity for desktop.....	29
3.2.4	Unity for console	30
3.2.5	Unity for web and Smart TVs.....	30
3.3	Vytváření aplikací v Unity.....	30
3.4	Programovací jazyky.....	31

3.5	Asset Store.....	31
3.6	Cena.....	32
4	Virtuální realita	33
4.1	Vývoj virtuální reality.....	33
4.1.1	První pokusy o virtuální realitu.....	33
4.1.2	Stereoskopické fotografie a prohlížeče	33
4.1.3	Sensorama.....	34
4.1.4	První VR displej umístěný na hlavě.....	34
4.1.5	První headset se sledováním pohybu.....	34
4.1.6	Damoklův meč.....	35
4.1.7	Název virtuální realita.....	35
4.1.8	Arkádové videohry s VR.....	35
4.1.9	Nintendo Virtual Boy.....	36
4.2	Virtuální realita v současnosti	36
4.2.1	Google Cardboard	37
4.2.2	Google Daydream.....	38
4.2.3	HTC Vive.....	39
4.2.4	Oculus Rift.....	40
4.2.5	Sony PlayStation VR.....	40
4.3	Pohyb ve virtuální realitě.....	41
4.3.1	Existující řešení pohybu ve virtuální scéně	41
5	Metodika práce	43
5.1	Motivace	43
5.2	Návrh řešení	43
5.3	Engine.....	43
5.4	Vývojové prostředí	44
5.5	Operační systém.....	44
5.6	Programovací jazyk.....	44
6	Implementace aplikace	45
6.1	Příprava	45

6.2	Nastavení	45
6.3	Použité knihovny a rozšíření.....	47
6.3.1	Standard Assets	47
6.3.2	Network Lobby	47
6.3.3	CNControls	47
6.3.4	GoogleVR.....	48
6.4	Prefabs	48
6.4.1	Player.....	48
6.4.2	Ghost.....	49
6.4.3	LevelArt.....	51
6.5	Scény	51
6.5.1	Main	52
6.5.2	3D Scene.....	53
6.5.3	Controller	54
6.6	Sestavení aplikace	55
6.7	Návod k použití	57
7	Testování	58
7.1	Uživatelská přívětivost	58
7.2	Bugy.....	58
7.2.1	Rozlišení hosta a klienta	59
7.2.2	Přidání interakce s okolím	59
7.2.3	Procházení objekty	60
8	Diskuze	61
9	Závěr	62
10	Literatura	63
11	Seznam obrázků	66

1 Úvod a cíl práce

1.1 Úvod

Virtuální realita byla v posledních letech hodně zpopularizována a to i díky neustálému zvyšování výkonu mobilních zařízení. Svůj podíl na tom mají také společnosti, které investují velké peníze do vývoje. Zájem mnoha lidí o virtuální realitu vzbudila společnost Google, když přišla na trh s velmi levným řešením Google Cardboard. Obyčejný karton a dvě speciální čočky stačí k tomu, aby se chytrý telefon s gyroskopem proměnil v systém, který vtáhne diváka do virtuální reality.

Tato práce se zabývá návrhem způsobu pohybu ve virtuální realitě pro mobilní telefony s operačním systémem Android. Ten je momentálně nejrozšířenějším mobilním operačním systémem, což jej činí nejvíce dostupným. Systém je vybrán právě z důvodu zasažení co největšího množství mobilních zařízení, které by potenciálně mohli aplikaci využít. Práce je spíše než aplikace pro koncové uživatele návrhem a ukázkou pro vývojáře. Ti si z práce mohou odnést nápad na vytvoření vlastního řešení pohybu virtuální scénou, nebo použít přímo toto řešení a dále ho rozšířit.

Celá tato práce bude vytvářena ve vývojovém prostředí Unity, které představuje nejpoužívanější řešení při tvoření mobilních aplikací. Přináší vývojářům plnou podporu při psaní aplikací pro Android a také v sobě má nativní podporu pro Google Cardboard. To z něj činí nejpoužívanější nástroj při vytváření aplikací a her pro virtuální realitu pod systémem Android.

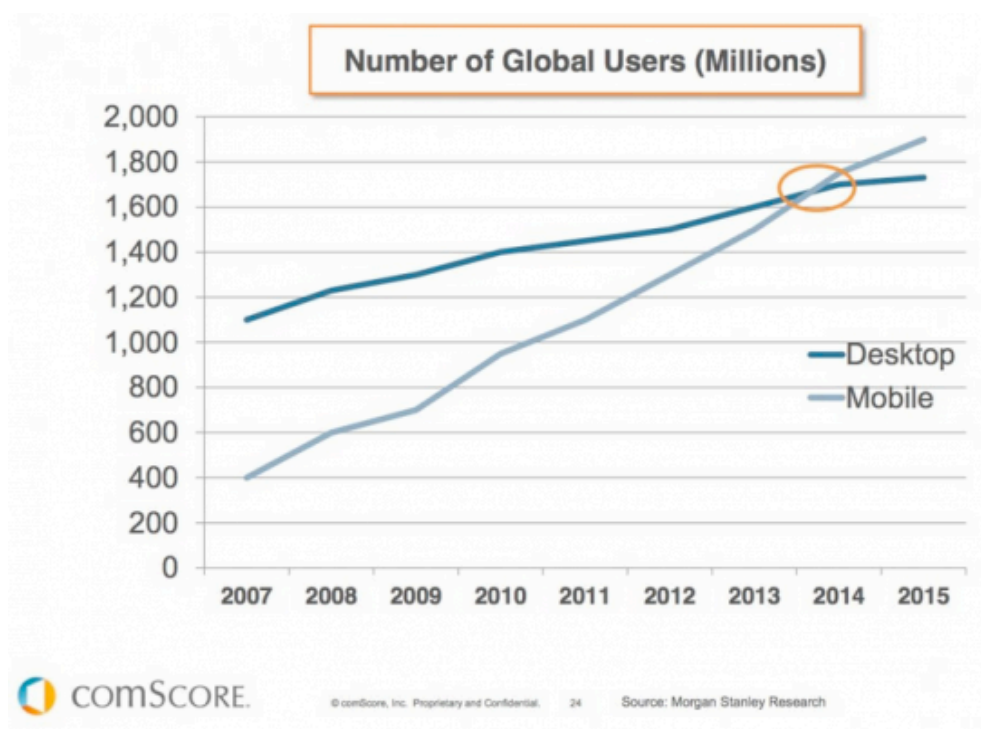
1.2 Cíl práce

Cílem této práce je udělat zážitek z virtuální reality o něco zajímavější. Nejdříve se práce zaměřuje na aktuální situaci ve světě vývoje aplikací, zvláště těch pro virtuální realitu. Poté je navrženo vlastní řešení pohybu a různých interakcí a následně je tento návrh implementován a otestován. Práce se zaměřuje hlavně na dostupnost a uživatelskou přívětivost.

2 Vývoj mobilních aplikací

Vývoj aplikací na mobilní zařízení s sebou nese různá omezení, týkající se menšího výkonu a také malé výdrže baterie. Problém s menším výkonem se každý rok zmenšuje, díky výkonnějším komponentám. Problém s bateriemi však zůstává, možná se dokonce i zhoršuje. Je to dáno tím, že mobilní zařízení dostávají stále více různých funkcí a senzorů, které spotřebovávají určitou energii, avšak kapacita baterií se nijak dramaticky nezvyšuje.

Oproti stolním počítačům mobilní zařízení kupujeme častěji. Je to dáno rychlejším vývojem tohoto odvětví a působí na to i řada dalších faktorů. Jedním z nich například může být to, že mobilní zařízení nosíme stále s sebou, ať už děláme cokoli a tak je zde mnohem vyšší riziko zničení přístroje oproti takovému stolnímu počítači, u kterého je riziko poškození ve srovnání minimální. Díky vyšším prodejm se mobilnímu odvětví daří a je to vidět i na grafu níže, kde je zřejmé, že počet uživatelů mobilních telefonů a tabletů již před několika lety překonal počet uživatelů stolních počítačů. (SmartInsights.com, 2017)



Obr. 1 Počet uživatelů mobilních zařízení a stolních počítačů.
Zdroj: SmartInsights.com (2017)

Mobilní aplikace jsou vyvíjeny s ohledem na výše uvedené nedostatky a proto je zde kladen velký důraz na optimalizaci kódu. Vývojář by se měl snažit, aby se provádělo co nejméně výpočtů jak v procesoru, tak v grafickém jádře. Vyšší nároky jsou také kladeny na využití menšího množství RAM paměti. Čím méně náročná je

aplikace na tyto prostředky, tím širší spektrum mobilních zařízení může oslovit. (SmartInsights.com, 2017)

2.1 Mobilní operační systémy

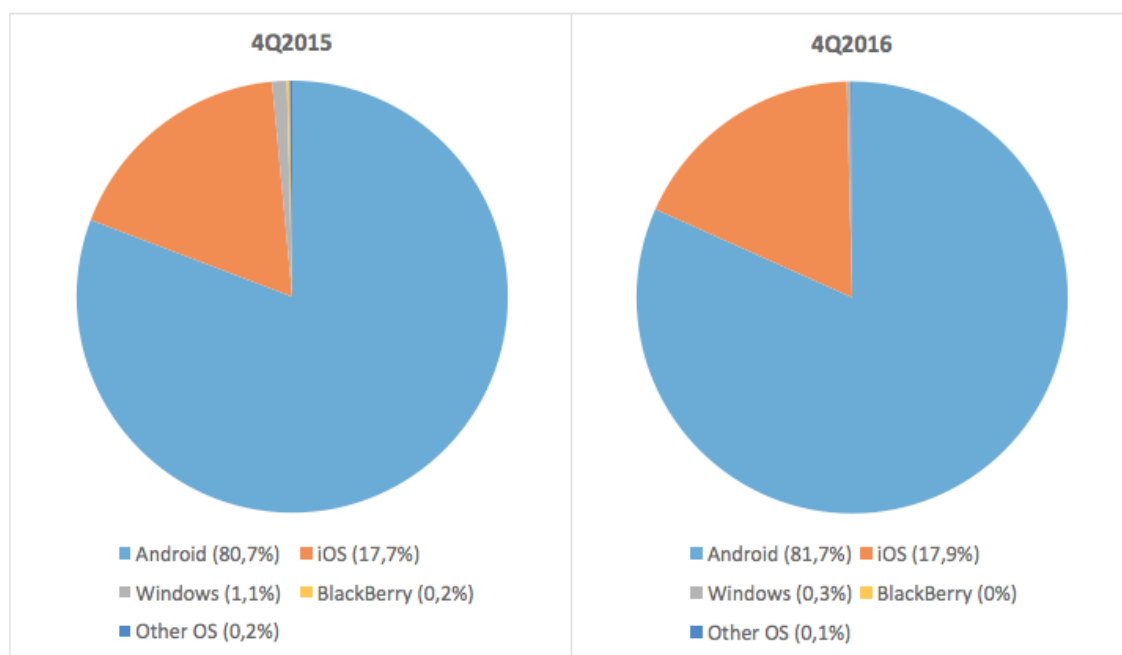
Operační systémy pro mobilní zařízení se od základu liší oproti systémům na stolní počítače. Velký důraz je kladen na nižší energetickou náročnost, která nejvíc ovlivňuje všechny části systému. Dalším důležitým rozdílem je menší výkon mobilních procesorů a grafických čipů. V dnešní době se však výkon neustále zvyšuje a komponenty jsou čím dál menší. Díky tomu by bylo možné na mobilních zařízeních provozovat (po výkonové stránce) i desktopové systémy. Microsoft se už nyní pokouší smazat rozdíly mezi desktop a mobilním systémem. Hlavním důvodem proč vůbec instalovat do mobilů jiné systémy je tak náročnost na spotřebu a v neposlední řadě také uživatelská přívětivost a ovládání. Je nutné se vypořádat se skutečností, že mobilní telefony mají oproti klasickým monitorům velmi malý display. V potaz se také musí brát ovládací zařízení. Klasická myš nebo touchpad ukazují přesně na jeden pixel, kdežto dotyk prstem na několikapalcovou obrazovku zasáhne desítky až stovky pixelů naráz. Ovládací prvky proto musí být dále od sebe a musí mít přizpůsobenou větší velikost. (Wikipedia, 2017)

Tab. 1 Statistiky prodeje smartphonů podle operačního systému ve 4Q16 a 4Q15

Operační systém	4Q16 (tisíce kusů)	4Q16 Podíl (%)	4Q15 (tisíce kusů)	4Q15 Podíl (%)
Android	352.669.9	81.7	325,394.4	80.7
iOS	77,038.9	17.9	71,525.9	17.7
Windows	1,092.2	0.3	4,395.0	1.1
BlackBerry	207.9	0.0	906.9	0.2
Other OS	530.4	0.1	887.3	0.2
Total	431,539.3	100.0	403,109.4	100.0

Zdroj: Gartner, 2017.

V tabulce jsou zobrazeny statistiky prodeje nových mobilních telefonů za období posledního kvartálu roku 2016 a pak posledního roku 2015. Pro lepší vizualizaci je z těchto dat vytvořen následující graf.



Obr. 2 Grafy prodeje mobilních telefonů podle operačního systému za 4Q15 a 4Q16.

Z grafu je patrné, že rozdělení trhu mobilních zařízení podle operačního systému je velmi stabilní a meziročně nedochází k velkým výkyvům. Z různých důvodů se lidé postupně přesouvají z méně zastoupených operačních systémů na hlavní dva protivníky Android a iOS. V posledním kvartálu roku 2015 bylo ještě celkem 1,5 % uživatelů na ostatních systémech, v posledním kvartálu roku 2016 už jich bylo pouze 0,4 % celkem. Nejvíce přístrojů bylo tedy prodáno se systémem Android (přes 80 %) a na druhém místě se drží systém iOS.

2.1.1 Windows Phone

Windows Phone je operační systém od společnosti Microsoft, který je nástupcem starší verze Windows Mobile. Jedná se však o zcela novou verzi, která není vzájemně kompatibilní. První verze byla vydána v roce 2010. (Wikipedia.org, 2017)

Tento systém se snaží využít popularity a rozšířenosti stolní verze operačního systému od firmy Microsoft a to řady systémů Windows (7, 8, 10). Stejné číslování jako u stolních verzí používá i pro mobilní systémy. Mezi jednotlivými verzemi jsou velké rozdíly, např. v samotném jádře, které je založené na různých platformách (Windows CE a novější verze jádro z řady Windows NT). (Wikipedia.org, 2017)

Uživatelské rozhraní využívá systém plochých dlaždic různé velikosti nazývaný Metro. Jedná se o velmi podobné prostředí, které začal Microsoft používat ve svém desktop systému Windows 8. Jednotlivé dlaždice jsou odkazy, které mohou spouštět aplikace, zobrazovat informace např. o počasí, akcích apod., nebo slouží pro rychlé volání nastaveného kontaktu atd. (Wikipedia, 2017)



Obr. 3 Uživatelské prostředí Windows 10.
Zdroj: techglobex.net (2015)

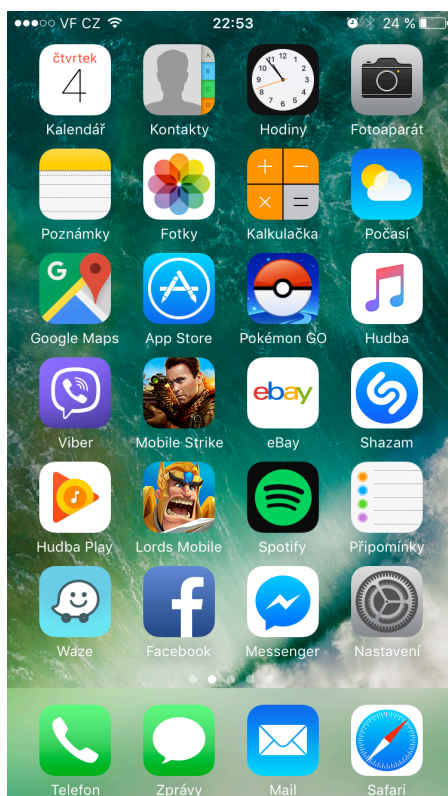
Aplikace se v tomto systému stahují přes Store. Lze zde zakoupit nebo zdarma stáhnout pouze aplikace a hry. Hudbu, videa a další multimediální obsah nabízí Microsoft zvlášť v jiné aplikaci. Předchůdcem obchodu Store na této platformě je Marketplace. Ten figuroval ve verzích systému před verzí 8 a obsahoval jak aplikace a hry, tak hudbu, videa apod. (Wikipedia, 2017)

Od verze 10 je vynecháno označení Phone, protože se podle Microsoftu jedná o univerzální systém, jak pro desktop, tak pro mobilní zařízení. Rozšíření názvu o slovo Phone se přidává v případě, že se chce v textu jasně naznačit, že se jedná o mobilní verzi tohoto operačního systému. (Wikipedia, 2017)

2.1.2 iOS

Operační systém od firmy Apple se nazývá iOS, což je zkratka pro iPhone Operation System. Tento systém je uzavřený a používán výhradně pro telefony iPhone a tablety iPad. Apple nelicencuje žádný ze svých operačních systémů jiným výrobcům hardwaru. Tím se liší od své největší konkurence systému Android. V iOS je kladen velký důraz na soukromí uživatele a bezpečnost. Systém je založený na Darwinu, což je open-source Unix systém vydaný Apple v roce 2000. První iOS byl vydaný v roce 2007 společně s prvním iPhone. (Wikipedia, 2017)

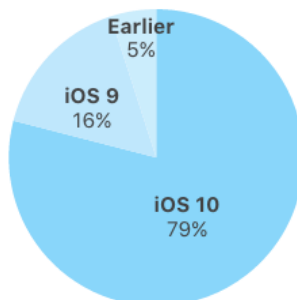
Uživatelské rozhraní se nazývá Cocoa Touch a používá systému ikon aplikací a odkazů seřazených do mřížky bez možnosti zvětšovat nebo zmenšovat jednotlivé ikony, jako tomu je u systému Windows Phone. Mezi sadami ikon lze listovat přejetím doprava nebo doleva a nejspodnější řada ikon zůstává na svém místě i při tomto posouvání. Aplikace lze slučovat do skupin, které si lze libovolně pojmenovat, kvůli lepší organizaci nebo většímu počtu nainstalovaných aplikací. (Wikipedia.org, 2017)



Obr. 4 Uživatelské prostředí iOS 10.

Aplikace do tohoto systému nelze stáhnout jiným způsobem než přes obchod s aplikacemi App Store. Ten obsahuje nabídku placených aplikací, i těch zdarma. Aby mohl někdo nasadit svou aplikaci do App Store, musí si nejdříve zařídit vývojářský účet a zařadit se do developer programu, za který se platí roční poplatek. Je to jeden ze způsobů jak se Apple brání před nekvalitními aplikacemi ve svém obchodě. Některé uživatele, kteří by si chtěli jen tak vyzkoušet nasadit zde svou aplikaci pro přátele toto může odradit. To však není jediná překážka, kterou je nutno překonat. Apple má velice přísná a specifická pravidla pro to, jak má vypadat aplikace a co musí splňovat, aby mohla být do obchodu vložena. Každá aplikace je před nasazením zkontrolována a pokud nesplňuje požadavky, je zamítnuta. Díky tomu se v obchodě nacházejí kvalitnější aplikace, které jsou lépe prověřovány. Tato skutečnost může mít vliv na to, že uživatelé iOS utrácejí za in-app nákupy průměrně dva a půl krát více než uživatelé Androidu. (Petr Mára, 2017)

Výhodou iOS je okamžité šíření aktualizací. V momentě kdy Apple vydá novou aktualizaci, tak si ji může okamžitě každý uživatel nezastaralého zařízení nainstalovat. Díky tomu je zde velký počet zařízení, které mají nainstalovanou poslední verzi systému. To je výhodné jak pro uživatele z důvodu bezpečnosti, tak pro vývojáře, kteří nemusí řešit podporu starších verzí systému. (Petr Mára, 2017)



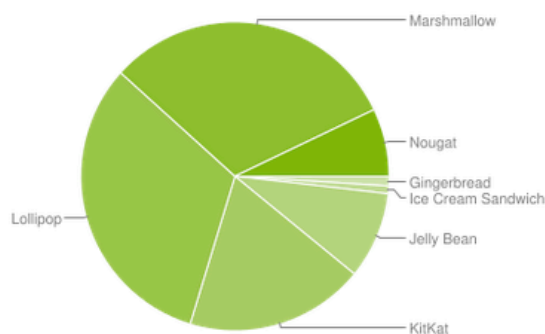
Obr. 5 Fragmentace verzí systému iOS.
Zdroj: Apple.com (2017)

Apple nepoužívá žádné centrální úložiště, ale každá aplikace pracuje ve svém prostoru a nemůže ovlivňovat další aplikace. Nevýhodou je, že nelze stahovat do zařízení dokumenty a poté k nim z různých aplikací přistupovat. Výhodou je vyšší bezpečnost, protože potenciální virus se nemůže snadno šířit dále. (Petr Mára, 2017)

2.1.3 Android

Android je mobilní operační systém vyvíjený společností Google. Je založený na linuxovém jádře a první verze byla publikována v roce 2008. Android byl původně vyvíjen firmou Android Inc., kterou Google v roce 2005 zakoupil. Na rozdíl od konkurenčního iOS, je Android otevřený systém a Google ho licencuje ostatním výrobcům telefonů. Systém je dodáván výrobcům hardware, kteří si jej dále upravují, před instalují své aplikace, přizpůsobují uživatelské prostředí atd. Díky tomu může stejná verze Androidu vypadat na zařízeních od různých výrobců jinak. Bude mít jiný vzhled a jiné před instalované aplikace. Spolu s touto skutečností vzniká jeden velký problém. (Wikipedia, 2017)

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.8%
4.1.x	Jelly Bean	16	3.2%
4.2.x		17	4.6%
4.3		18	1.3%
4.4	KitKat	19	18.8%
5.0	Lollipop	21	8.7%
5.1		22	23.3%
6.0	Marshmallow	23	31.2%
7.0	Nougat	24	6.6%
7.1		25	0.5%



Obr. 6 Fragmentace verzí systému Android.

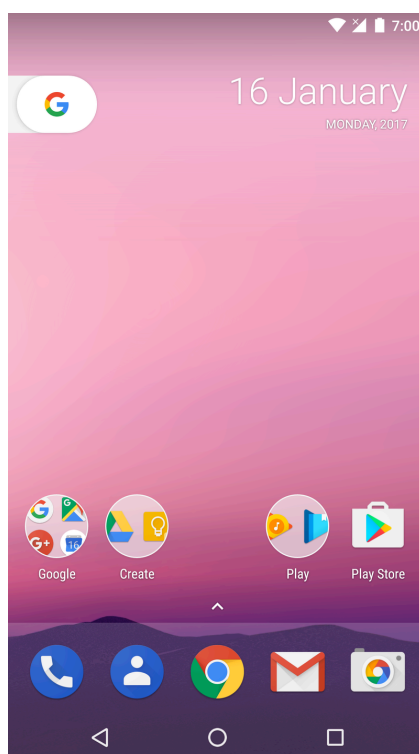
Zdroj: Android.com (2017)

Jelikož je na každém systému přidána upravená vrstva od konkrétního výrobce, tak nelze šířit aktualizace a bezpečnostní záplaty tak snadno jako u systému iOS. Ve chvíli, kdy vyjde nová verze Androidu ji musí výrobci opět upravit a poskytnout svým zákazníkům. Nevýhodou je, že tento proces nějakou dobu trvá a pokud jde o starší zařízení, tak se uživatel aktualizace nemusí dočkat vůbec. Nejen že to je nepříjemné, protože se nedostane k nejnovějším funkcím, ale navíc je zde velké bezpečnostní riziko. Pokud totiž Google zveřejní seznam chyb, které aktualizace opravuje, tak se na tyto chyby v blízké době zaměří velké množství útočníků. Těm pak stačí využít toho, že oprava těchto chyb pro uživatele nějakou dobu jednoduše neexistuje a mohou slabého místa zneužít. Google vyrábí i své vlastní telefony a tablety, ve kterých je čistý Android a tuto nevýhodu nemají, aktualizace lze stáhnout bezprostředně po jejich vydání. (Petr Mára, 2017)

Dalším slabým místem v bezpečnosti Androidu je společné úložiště. Díky tomu si ale tento systém získal na velké popularitě, protože se v tomto podobá nejrozšířenějšímu desktopovému operačnímu systému Microsoft Windows. Jakákoliv aplikace může stahovat data do společného úložiště, ať už dokumenty, hudbu, videa nebo jakékoliv další souborové formáty. Tyto soubory mohou být infikovány virem a snadněji se mohou šířit. Díky tomu, že Android je nejpoužívanější mobilní operační systém, jsou případné chyby odhaleny mnohem dříve. (Petr Mára, 2017)

Pokud se podíváme na uživatelské prostředí, zjistíme, že se od předchozích systémů liší. Podobně používá systém ikon odkazujících na jednotlivé aplikace nebo skupiny aplikací, ale na obrazovku lze umístit také různé gadgety. Ty umožňují hodně si vzhled telefonu přizpůsobit svým potřebám. Lze použít vlastní způ-

sob zobrazování data, času, počasí, akcí, zpráv atd. přímo z gadgetu na ploše. Tuto možnost konkurenční iOS nemá. (Petr Mára, 2017)



Obr. 7 Uživatelské prostředí Android 7.1 Nougat.
Zdroj: Calvin Hogg (2017)

Obchod pro nakupování aplikací se v systému Android jmenuje Google Play. Na rozdíl od ostatních systémů, lze zde zakoupit nejen aplikace a hry, ale také multi-mediální obsah jako je hudba, filmy, podcasty a další. Veškerý dostupný obsah tak lze sehnat na jednom místě. Aplikace do mobilních zařízení se systémem Android lze instalovat i bez Google Play a to přímo z balíčků APK. To s sebou nese výhody jednoduché distribuce programů a her, ale také představuje bezpečnostní riziko. Lze tak totiž do systému nainstalovat libovolnou aplikaci, která může být potenciální hrozbou bezpečnosti. Na rozdíl od iOS lze aplikace do Google Play přidávat bez speciálního placeného účtu. Android je v tomto ohledu mnohem přístupnější pro začínající vývojáře. (Google Play, 2017)

2.1.4 Ostatní systémy

Zastoupení systémů jiných než tři výše jmenovaných je velmi malé. Dle statistik je to kolem 0,1 % a navíc toto číslo každoročně klesá. Hlavním důvodem může být dostupnost aplikací v obchodech, které má každý systém vlastní. Hlavním důvodem malého používání těchto systémů jsou vývojáři, kteří, když už si musí vybrat, pro kterou platformu budou programovat, volí raději nejrozšířenější systémy. Je to

z důvodu vyššího počtu uživatelů a tím pádem i možnosti vyšších zisků o které hlavně jde. (ShoutMeLoud, 2017)

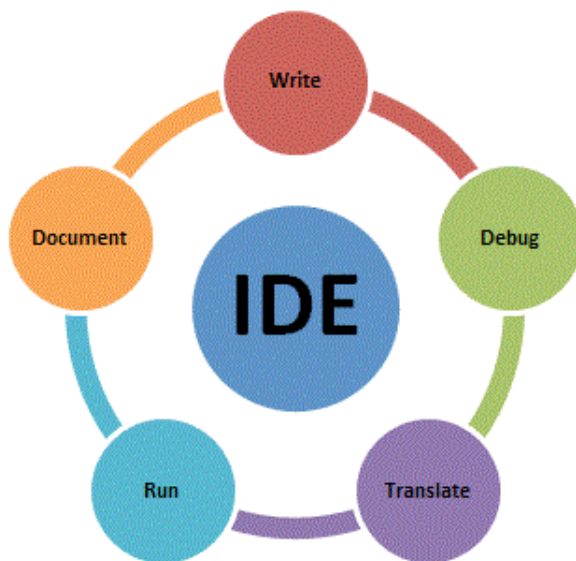
Jeden z alternativních systémů je Symbian. Je to operační systém firmy Nokia, která vyrábí mobilní telefony. Nokia však byla odkoupena společností Microsoft a ten ve vývoji Symbianu nepokračoval a nasadil vlastní systém. Nokia je v současnosti zpět na trhu, ale Symbian je již zastaralý a dále nevyvíjený. (ShoutMeLoud, 2017)

Dalším alternativním systémem je Blackberry OS, které je stejně jako systém iOS uzavřený. Velký důraz je zde kladen na bezpečnost a šifrování uživatelských dat. Tento operační systém firma vyvíjí pouze na své vlastní telefony a cílí s nimi hlavně na firemní zákazníky. Firma se v současnosti potýká s krizí, právě kvůli odlivu zákazníků ke konkurenci. (ShoutMeLoud, 2017)

Firefox OS je open-source mobilní operační systém od společnosti Mozilla. Je navržený speciálně pro HTML5 aplikace, které komunikují rovnou s hardwarem zařízení. Mozilla si slibuje velké množství vývojářů aplikací, z řad těch, kteří již ovládají HTML5. (ShoutMeLoud, 2017)

Existují i další systémy, které jsou však v současnosti již zastaralé a nevyvíjené, nebo jsou velmi málo používané.

2.2 Vývojová prostředí



Obr. 8 Funkce IDE.
Zdroj: Murtaxa, 2016

K vývoji mobilních aplikací se používají vývojová prostředí, které usnadňují práci programátorům. Vývojové prostředí obsahuje editor kódu, našeptávač kódu, debugger, kompilátor atd. Některá prostředí obsahují také nástroj pro návrh a vývoj vizuálního grafického návrhu uživatelského rozhraní. Vývojová prostředí slouží ke

zvýšení produktivity programátorů. Jelikož tato prostředí obsahují spoustu funkcí, projeví se zvýšení produktivity až po určité době, po které se programátor s prostředím pečlivě seznámí a sžije se s ním. Zpravidla jsou tyto prostředí přizpůsobena pro jeden konkrétní programovací jazyk, avšak existuje i mnoho více jazykových prostředí. (Wikipedia, 2017)

2.2.1 Android Studio

Android Studio je oficiální vývojové prostředí pro Android přímo od firmy Google. Umožňuje instantní spuštění upravené aplikace na připojeném zařízení, nebo spuštění v Android emulátoru. Emulátor dokáže instalovat a spouštět aplikace rychleji než klasické fyzické zařízení. Android Studio obsahuje také inteligentní editor kódu, který podle Googlu umožňuje rychlejší a více produktivní práci v každém kroku. Android Studio je založené na architektuře IntelliJ a je schopno pokročilého dokončování kódu, refaktorování a analýzy kódu. Podporuje také kódové šablony a má integrovanou podporu sdílení kódu přes GitHub. Tím ještě více usnadňuje práci začátečníkům. Zároveň však díky integraci GitHubu podporuje i velké firmy, které potřebují distribuovat kód mezi všemi pracovníky. Android Studio poskytuje jednotné prostředí pro vývoj aplikací na telefony se systémem Android, tablety, Android TV, Android Auto a Android nositelné doplňky, jako například chytré hodinky. Toto vývojové prostředí je multiplatformní a lze jej nainstalovat na Windows, Mac i Linux. (Google Developer, 2017)

2.2.2 XCode

Oficiální vývojové prostředí pro systémy od firmy Apple se nazývá XCode. Umožňuje vytvářet aplikace pro iPhone, iPad, Mac, Apple Watch a Apple TV. Poslední verze s pořadovým číslem 8 obsahuje radikálně rychlejší vývojové prostředí s editorem, který lze plně přizpůsobit pomocí instalovatelných rozšíření. Nový systém dokáže upozornit na běhové chyby, na skryté bugy, které ukazují mimo paměť a také obsahuje nový Memory Debugger, který se zanořuje mnohem hlouběji, při hledání chyb s paměť. (Apple Developer, 2017)

Nový Interface Builder poskytuje lepší kontrolu nad vytvářením uživatelského prostředí v aplikacích. Dokáže zobrazovat živý náhled podoby aplikace na různých zařízeních, ať už v horizontální nebo vertikální poloze. Dokáže také zobrazit, jak bude aplikace vypadat, pokud bude uživatelem na zařízení spuštěna v módu dvou aplikací na jedné obrazovce. Interface Builder obsahuje lepší nastavení a přiřazení zobrazení v rámci responsibility. (Apple Developer, 2017)

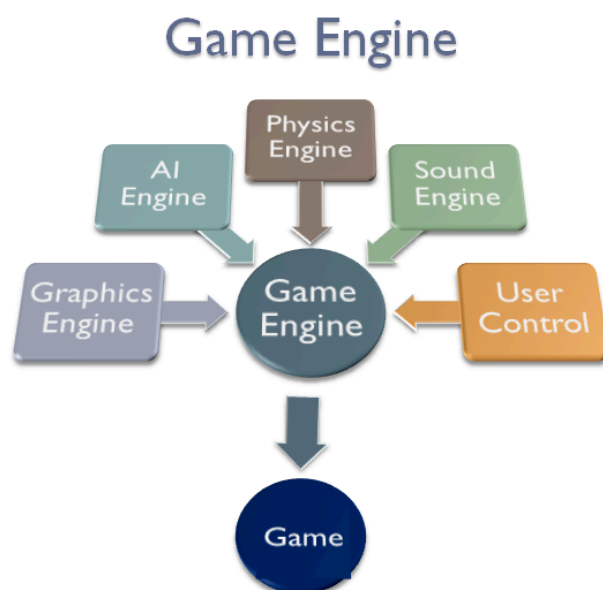
Apple pro vývoj aplikací navrhl svůj vlastní programovací jazyk jménem Swift. V poslední verzi 3 slibuje více konzistentní a přirozené pojmenování API. Swift je určen jako náhrada jazyků C (C, C++ a Objective C). Jelikož je tento jazyk vcelku nový (poprvé se objevil v roce 2014), musí se snažit být srovnatelný s těmito jazyky nebo ještě lepší, aby přesvědčil nové i zkušené programátory, aby se tento jazyk naučili a používali jej. (Apple Developer, 2017)

2.2.3 Unity

Obě předchozí vývojová prostředí se zaměřují pouze na jednu mobilní platformu, podle jejího výrobce. Tedy v XCode nelze vytvářet aplikace pro Android a v Android Studiu zase nelze tvořit aplikace pro iOS. Existuje však vývojové prostředí Unity, ve kterém lze vytvářet aplikace pro Android, iOS, Windows Phone, Windows, Mac, Linux, WebGL, PlayStation 4, Xbox One a spoustu dalších platform. V tomto ohledu lze nazvat Unity nejvíce multiplatformním vývojovým prostředím. Unity je v této práci věnovaná samostatná kapitola. (Unity, 2017)

2.3 Engine

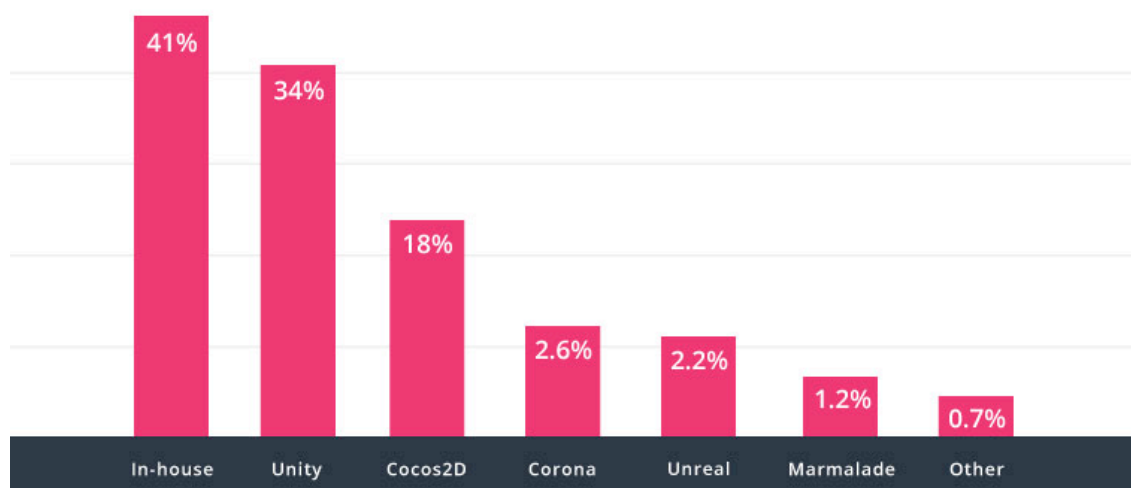
Engine, někdy nazývaný také game engine, je softwarový framework používaný při návrhu a vytváření her. Samozřejmě se používá i pro všechny ostatní typy softwaru, ne jen pro hry. Engine tvoří jádro celého programu a jeho vývoj je velmi náročný. Často je tedy z ekonomických důvodů výhodnější zaplatit za využití nějakého hotového engine, než platit vývojářům za spoustu času stráveného vývojem vlastního engine. Dělat vlastní engine má smysl v případě, že se jedná o firmu, která má v plánu vydat velké množství her a systém provizí a placení za engine je v tomto případě nevyhovující. Proto hodně vývojářů používá některý z hotových engineů. Každý z nich má jiné podmínky použití, jiné možnosti a také jinou cenovou politiku. (Factions By Pidbaq, 2017)



Obr. 9 Z jakých částí se skládá game engine.
Zdroj: Factions by Pidbaq, 2016

Engine často poskytuje služby jako je renderování 2D a 3D grafiky, engine fyziky, detekci kolizí, zvuky, skriptování, animace, umělou inteligenci, síťové nastavení,

system správy paměti, system překladač a mnoho dalšího. Tyto služby však nemusí být poskytovány pouze v rámci celého balíčku, ale mohou být využívány pouze některé části, které lze nazvat middleware. Samotný game engine je sám o sobě také middleware. Může se tak zkombinovat vytvoření zcela vlastního engine, ale na některé části použít hry nebo programu použít právě tento doplněk. (Factions By Pidbaq, 2017)



Obr. 10 Podíl engineů nejúspěšnějších 1000 her zdarma v prvním kvartále roku 2016.
Zdroj: Unity3D, 2016

Graf na předchozím obrázku zobrazuje statistiky o nejvíce používaných game enginech. Tato data jsou sledována u 1000 nejúspěšnějších mobilních her, které byly nabídnuty ke stažení zdarma v období prvního kvartálu roku 2016. Z dat jasně vyplývá, že 41 % her využívá vlastní engine dané společnosti. Lze se tak domnívat, že se jedná o společnosti, které vydávají větší množství her a tak se jim vývoj vlastního engine vyplatil. Mezi ostatními komerčními enginey se jako první umístil Unity a poté Cocos2D. Dalšími v pořadí je Corona a Unreal, kteří mají podíl menší než 3 % a Marmalade, která má pouze 1,2 %. Ostatní enginey jsou velmi málo využívané a dohromady mají pouze 0,7 % trhu.

2.3.1 Unity

Engine od Unity podporuje všechny základní operační systémy jak mobilní, tak stolní. Nabízí podporu 3D a virtuální reality. Podporovány jsou skripty v jazycích C# a Javascript. Unity je nejrozšířenější engine pro vývoj mobilních aplikací a v této práci je mu věnovaná samostatná kapitola. (Unity, 2017)

2.3.2 Cocos2D

Cocos2D je open-source balík pro vývoj her. Největší výhodou tohoto balíku, a také zároveň i důvod jeho popularity, je to, že je poskytován zdarma. Podporuje pro-

gramovací jazyky C++, Lua a Javascript. Cocos2D podporuje vývoj aplikací a her na mnoho platform jak mobilních, tak stolních. Konkrétně to jsou mobilní operační systémy iOS, Android, Windows Phone a počítačové systémy Windows, Linux a Mac OS X. Zahrnuje tak všechny nepoužívanější systémy z obou kategorií. Tento engine používají vývojáři na volné noze i velké společnosti a jeho využití je zdarma i pro komerční účely, tedy za účelem zisku. Ačkoliv to z názvu nemusí být jasné, Cocos2D neslouží pouze k vývoji 2D her. Od verze 3 je podporována také 3D grafika s ní spojené funkce jako například prostorové umístění kamery nebo přidávání světelných zdrojů. Cocos2D také podporuje virtuální realitu, ale zatím v celkem omezené míře. (Cocos2d-x, 2017)

2.3.3 Corona

Dalším nástrojem pro vytváření aplikací a her je engine Corona. Stejně jako předchozí engine, je Corona také k dispozici zdarma. Corona na svých stránkách uvádí několik přinejmenším zavádějících informací. Hned na úvodním banneru stránky se píše, že s tímto enginem lze vytvářet aplikace desetkrát rychleji. Bohužel zde není uveden zdroj nebo odkaz na ověření této skutečnosti. Vyšší rychlost vývoje má být zaručena snadno naučitelným jazykem Lua, ten však podporuje i výše uvedený engine Cocos2D. Další informace hned na úvodní stránce je, že Corona je 100 % zdarma. Na stránce produktu se však dozvíme, že existuje i placená verze. Logicky tak musí být o některé funkce verze zdarma ochuzena, jinak by nemělo smysl kupovat placenou verzi. Poslední zavádějící informace je uvedena v ceníku, kde stojí, že Corona SDK je vedoucí mobilní vývojová platforma. S podílem 2,6 % na trhu to však není možné. (Corona Labs, 2017)

Odhlédneme-li od těchto nepřesných nebo nepravdivých informací, tak Corona podporuje všechny tři nejpoužívanější mobilní systémy a kromě Linuxu i stolní systémy. Podporuje větší množství programovacích jazyků než konkurence a to Obj-C, C, Java, C++, C# a Lua. Základní engine lze rozšířit pomocí pluginů a assetů, které jsou k dispozici placené nebo zdarma v obchodě Corona Marketplace. (Corona Labs, 2017)

Zajímavou funkcí, kterou Corona má, je možnost takzvaných živých buildů. Pomocí několika kliknutí stačí vytvořit live build, nainstalovat zkompilovanou aplikaci do jednoho nebo více zařízení a připojit je do lokální sítě. Jakékoliv změny kódu se pak projeví na zařízeních téměř okamžitě, což by mělo dost urychlovat testování funkčnosti aplikace. O podpoře virtuální reality se na stránkách nic nepíše, z čehož lze usuzovat, že virtuální realitu nepodporuje. (Corona Labs, 2017)

2.3.4 Unreal

Další engine na trhu je Unreal od společnosti Epic Games, Inc. Cenová politika této firmy je jasná a srozumitelná. Po registraci poskytnou všechny zdrojové kódy, nástroje, ukázkové projekty, aktualizace atd. zdarma a vývojář zaplatí až v momentě, kdy se hra začne prodávat 5 % svých zisků z takto vytvořené hry. Unreal nabízí pokročilou grafiku a plnou podporu virtuální reality. Dokonce lze scény pro virtu-

ální realitu upravovat přímo v ní, pomocí speciálního editoru s nasazenými brýlemi. Unreal má také speciální renderovací engine pro virtuální realitu, který pomocí dopředného renderování dokáže produkovat kvalitní grafiku při rychlosti vykreslení až 90 snímků za vteřinu. (Unreal, 2017)

Unreal podporuje standartní platformy iOS, Android a Windows Phone. Podporuje také všechny tři základní počítačové systémy a navíc nabízí přímou podporu pro řadu řešení virtuální reality jako je Oculus, Playstation VR, Samsung Gear VR, Google Daydream a Viveport. Má také vlastní Marketplace s pluginy a rozšířeními, kde mohou obsah nabízet samotní uživatelé. Engine Unreal podporuje pouze programovací jazyk C++. (Unreal, 2017)

3 Unity

Unity je v současnosti nejpoužívanější nástroj pro vývoj mobilních aplikací a her. Nabízí jednoduché a uživatelsky přívětivé prostředí, silnou komunitu a velkou podporu pro začínající uživatele. Podporuje všechny nejběžnější platformy, mezi kterými lze snadno přepínat. Ačkoliv je kladen velký důraz na jednoduchost pro začínající vývojáře, tak je Unity profesionální nástroj, který používají i velké firmy zabývající se vývojem aplikací a her. Podporuje vývoj 2D a 3D her a také poskytuje podporu pro vývoj aplikací pro virtuální realitu a také rozšířenou realitu. (Unity, 2017)

3.1 Uživatelská přívětivost

Uživatelskou přívětivostí v tomto případě není myšlen uživatel programu vytvořeného v Unity, ale uživatelská přívětivost Unity pro samotné vývojáře. Unity hodně myslí na nováčky v oblasti programování aplikací. Snaží se jim co nejvíce usnadnit start vývojářské kariéry a tak získává nové zákazníky. Snaží se pro ně dělat tutoriály, kde se dozví jak udělat v Unity celou hru od začátku do konce nebo i návody zaměřené na konkrétní problémy z oblasti vývoje. Existují tak tutoriály které ukazují jak vyřešit pohyb uživatele, jak se přepínat mezi scénami. Unity se také dost přizpůsobuje lidem, kteří vůbec neumí programovat a chtěli by i přes to vyvíjet hry. Pomáhá v tom editor, kde se místo psaní kódu vybírá hodně vlastností přes grafické prostředí metodou táhni a pusť. Z Asset Store lze stáhnout hotové kódy, které řeší nějaký problém bez nutnosti jeho znalosti. (Unity, 2017)

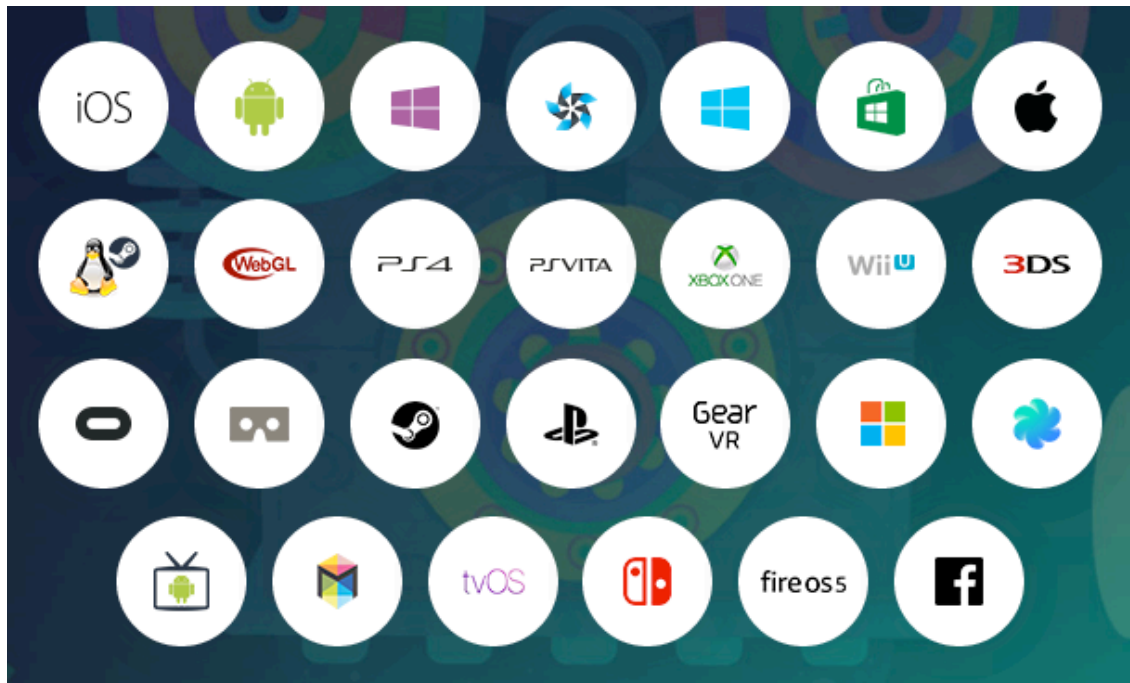
Velmi dobře zpracovaná je dokumentace, která je dostupná online. Lze v ní najít veškerý popis komponent a procházet API, jeho popis i s ukázkami použití. Vše je provázané odkazy, což usnadňuje pohyb v dokumentaci a její porozumění. Pokud uživatel nepochopí jak daná komponenta funguje, může se podívat na některý z tutoriálů, kde je vysvětleno a ukázáno, jak se s danou komponentou pracuje. (Unity, 2017)

Okolo Unity existuje velká komunita vývojářů, kteří si vzájemně radí a pomáhají na fóru. Stačí se registrovat, přečíst pravidla fóra a potom zde lze pokládat dotazy zkušenějším vývojářům, kteří rádi poradí. Spousta problémů je zde už dotázána, takže není velká potřeba zakládat nové otázky, ty lze najít v už existujících diskuzích. V případě nějaké otázky lze také nahlédnout do znalostní báze, která obsahuje nejčastější dotazy rozdělené do sekcí. (Unity, 2017)

3.2 Podpora platforem

Velkou výhodou Unity je, že podporuje velmi široké spektrum platforem. Aplikaci tak zde lze vytvořit pro mobilní zařízení, ale také stolní systémy nebo i herní konzole. Navíc tvorba multiplatformních her je zde velmi jednoduchá. Stačí hru udělat pouze jednou, udělat deploy pro danou platformu, poté jednoduše přepnout na

jinou platformu a udělat znovu deploy. Odpadá tak vytváření stejné aplikace ve dvou různých vývojových prostředích pro různé systémy, což přináší obrovskou časovou úsporu a menší náročnost na počet vývojářů. (Unity, 2017)



Obr. 11 Podporované platformy Unity.
Zdroj: Unity, 2017

3.2.1 Unity for mobile

Jeden herní engine pro mobilní zařízení podporuje pět platforem. Nejpoužívanější Android, pak iOS a Windows Phone. K těmto třem je přidána podpora mobilního systému Tizen a také Fire OS od společnosti Mozilla. Pro všechny tyto systémy je nabízeno sestavení pomocí jednoho kliknutí. (Unity, 2017)

3.2.2 Unity for VR and AR

V Unity vidí ve virtuální realitě budoucnost a tak ji široce podporují. Velký důraz je kladen na schopnost zobrazení velkého počtu snímků za vteřinu, kterého lze dosáhnout díky dobré optimalizaci. Společně s virtuální realitou je podporována i rozšířená realita. Unity už je tak daleko, že zabudovali nativní podporu pro následující systémy: Oculus Rift, Gear VR, PlayStation VR, Microsoft HoloLens, Steam VR/Vive, Google Daydream a Cardboard. (Unity, 2017)

3.2.3 Unity for desktop

Podpora her pro stolní systémy je v Unity samozřejmostí. Hry lze vytvářet pro všechny tři nejpoužívanější operační systémy macOS, Windows i Linux. Pro systém Windows lze vytvářet aplikace zařaditelné do obchodu Windows Store. Mimo tyto

základní systémy je zde podpora i pro Steam OS a také Facebook Gameroom. (Unity, 2017)

3.2.4 Unity for console

Vývoj her na konzole je s Unity jednodušší než kdy předtím. V Unity lze vytvářet hry pro PlayStation 4, Xbox One, PlayStation Mobile, PlayStation Vita a Wii U bez žádných dalších poplatků. Každá z těchto konzolí má svůj schvalovací proces a je nutné si zjistit bližší informace přímo u ní. (Unity, 2017)

3.2.5 Unity for web and Smart TVs

S Unity lze vytvářet i webové hry, které ke svému běhu nepotřebují žádný dodatečný plugin a dosahují dobrých výkonnostních výsledků. Zastoupen je zde i rychle rostoucí segment aplikací a her pro chytré televize, které se stávají čím dál více dostupnějšími a tím pádem i rozšířenějšími. (Unity, 2017)

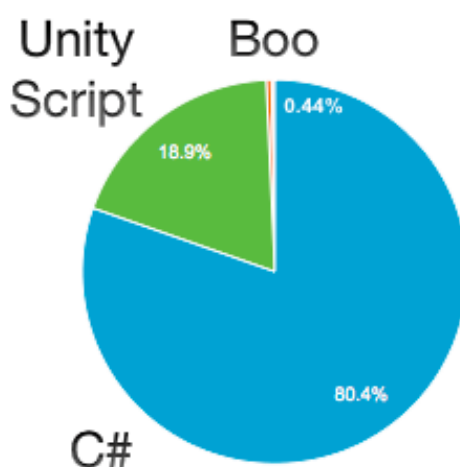
3.3 Vytváření aplikací v Unity

Při vytváření her se používají scény, které v sobě obsahují hierarchii objektů. Mezi jednotlivými scénami se lze přepínat pomocí skriptů. Příkladem scény může být uživatelské menu, ze kterého se spouští hra. Spuštění hry pak probíhá jako načtení jiné scény. Ve scéně je možné mít různé typy objektů. Vždy zde musí být kamera, která je zaměřená na určitou část scény, která je zobrazována. Dále mohou být ve scéně různé typy světél, 3D i 2D objektů, zdroj zvuku a další prvky uživatelského rozhraní. Pro tvorbu tlačítek a formulářů se používá plátno, na které jsou prvky umisťovány. Každý objekt má ve scéně svou pozici, rotaci a měřítko. Speciálním typem objektu, je prázdný objekt, který nemá viditelnou formu, ale používá se jako nosič skriptů a vlastností. Na objekty lze připínat skripty, pluginy nebo nějaké vlastnosti, které je ovlivňují. Objekty lze do sebe zanořovat ve formě stromu. Každý objekt může mít potomky a předchůdce, nebo také může být ve scéně samostatně. Složitější objekty složené z více objektů tak lze seskupovat do jednoho a snáze s nimi pracovat. Z libovolného objektu lze vytvořit prefabrikát, který lze opakovaně použít. Výhodou prefabrikátu je, že z něj můžeme dělat unikátní instance, ale také lze použít hromadnou změnu všech instancí. Do aplikace lze importovat balíky s rozšířeními, které umožňují rozšířit hru například o virtuální realitu, nebo přidává nový způsob ovládání. Přímo v Unity editoru lze aplikaci spustit a vyzkoušet, jak bude hotový program fungovat. Není tak nutné po každé úpravě nahrávat demo do zařízení, pro které je aplikace vyvíjena. Veškeré změny, kterou jsou provedeny při spuštění aplikace jsou po ukončení běhu stornovány. Výhodou je, že lze za běhu zkusit měnit různé parametry a sledovat co se bude dít, bez poškození funkčnosti. Nevýhodou je, že všechny změny jsou zahozeny a musí se provést znovu v editačním módu. V nastavení „Build Settings“ se volí platforma, architektura a podobné vlastnosti. K dispozici je zde také odkaz na nastavení vlastností projektu

pro jednotlivé platformy jako například název vydavatele, číslo verze, výchozí rozlišení atd. (Unity Manual, 2017)

3.4 Programovací jazyky

V Unity lze vytvářet skripty, které potom lze připnout na objekty ve scéně. Tyto skripty potom ovlivňují chování tohoto objektu. Pokud je potřeba nějaký skript, který se nevztahuje ke konkrétnímu objektu, řeší se to tak, že se vytvoří prázdný game object a na ten se potom skript připojí. Podle toho co skript řeší se pojmenuje i samotný objekt. (Unity Manual, 2017)



Obr. 12 Podíl skriptů vytvořených ve třech dostupných jazycích v Unity.
Zdroj: Unity, 2014

V Unity lze používat v současnosti dva typy skriptů. V programovacím jazyce C# lze psát C# skripty nebo lze použít Javascript. Ve většině tutoriálů je používán programovací jazyk C# a to je zároveň dle statistik uživateli nejpoužívanější jazyk. Jediné statistiky, které Unity zveřejnilo jsou z roku 2014, takže se jedná o staré informace, ale už v té době používalo C# přes 80 % uživatelů. Unity Script je v tomto kontextu Javascript s drobnými úpravami od Unity. Boo skripty byly úplně zrušeny a nadále nejsou podporovány. Podle doporučení od uživatelů na fórech Unity a také podle používanosti v tutoriálech a v Asset Store, je v současnosti C# nejpoužívanější jazyk pro skriptování. (Unity Manual, 2017)

3.5 Asset Store

Obchod s různými doplňky a funkcemi se v Unity nazývá Asset Store. Tento obchod má za cíl usnadnit práci vývojářům, kteří si zde mohou zakoupit některé rozšíření do své hry a nemusí ho tak vytvářet sami. V některých případech toto řešení ušetří mnoho času a tím i peníze. Záleží pak vždy na ceně daného doplňku. Pokud náklady

na celkový čas potřebný k vytvoření daného doplňku jsou vyšší než cena za doplněk a čas potřebný k úpravě pro konkrétní použití, tak se zakoupení vyplatí. Je nutné se také zamyslet nad časovým urychlením, pokud se daná věc nemusí vyvíjet a testovat na vlastní pěst. (Unity, 2017)

Doplňky v obchodě jsou pro zpřehlednění rozděleny do několika kategorií a seřazeny podle abecedy. Spousta modelů, skriptů a dalších věcí je zde nabízena zdarma, stačí vyfiltrovat výsledky dle ceny. Filtrovat lze také podle velikosti, minimálního hodnocení, podle doby od poslední aktualizace a doby od vydání nebo také vyfiltrovat jen výsledky, které jsou použitelné pro konkrétní verzi. (Unity, 2017)

3.6 Cena

Unity nabízí několik cenových variant. Lze zakoupit licenci za pevnou cenu, výběrem jednoho ze tří balíčků, nebo pro velké firemní zákazníky Unity nabízí možnost individuálního nacenění. Všechny verze od neplacené, po tu nejdražší obsahují následující funkce a vlastnosti: všechny funkce engine, možnost tvořit aplikace pro všechny dostupné platformy, přístup k posledním aktualizacím Unity, neplatí se žádné podíly ze zisku, engine in-app nákupů, Unity Ads a přístup k beta verzím Unity. I přesto, že jsou na trhu i řešení zdarma, používá Unity největší procento vývojářů. Bližší podmínky a informace o ceně jsou dostupné na stránkách Unity. (Unity, 2017)

Plan	Description	Price	Action
Personal	All the features for beginners & hobbyists to get started. Free No credit card required	Free	Try Personal
Plus	For serious creators looking to bring their vision to life.	\$35 per seat/month	Get Plus
Pro	For professionals looking to profit from advanced customization and complete flexibility.	\$125 per seat/month	Go Pro
Enterprise	A tailored solution to suit your organization's creative goals.	Contact us	Contact us

Until May 31st. Get top-selling Assets for free. [Learn more](#)

Also included with Pro

Obr. 13 Cenové plány Unity.
Zdroj: Unity, 2017

4 Virtuální realita

S vývojem počítačových systémů, nárůstem výkonu i grafických možností se stává virtuální realita stále dostupnější. Do vývoje vlastních řešení se zapojuje čím dál více společností, přibližují ji více lidem a tak je o ni stále větší zájem. Virtuální realita umožňuje procházet 3D scény zcela novým způsobem. Nejčastěji se obraz promítá do zobrazovacího zařízení ve formě brýlí, které si uživatel nasadí na hlavu. Místo sledování obrazovky a prohlížení modelů pomocí klávesnice a myši nebo touchpadu, je obraz promítán přímo před oči. Díky tomu, že brýle zakrývají okolí, je hráči umožněno lepší vycítění do hry a není ničím rušen. (Virtual Reality Society, 2017)

Technicky by se dala virtuální realita popsat jako trojrozměrné prostředí, vygenerované počítačem. Cílem samozřejmě je, aby generovaná realita působila co nejvíce realisticky na naše smysly. Mezi naše vnímání okolí pomocí zraku je vložen mezičlánek, který simuluje umělé prostředí. Zvukové vjemy okolí jsou simulovány reproduktory nebo sluchátky. Stačí uměle vybudit tyto naše dva smysly a náš mozek je tím dostatečně ošálen. Virtuální realita tak může ovlivňovat náš smysl pro orientaci a díky tomu můžeme mít například pocit že padáme i když stojíme pevně na zemi. Tohoto jevu využívají hry simulující jízdu na horské dráze a lidé se opravdu cítí jako kdyby na dráze jelo a někdy se jim i udělá špatně, jako by vše bylo skutečné. (Virtual Reality Society, 2017)

4.1 Vývoj virtuální reality

Ačkoliv dnešní virtuální realita je zprostředkovávána většinou pomocí počítače, tak historie virtuální reality sahá do dob před počítači, jak je známe dnes. Dříve se používali jiné systémy pro stimulaci našich smyslů za účelem vytvoření iluzionární reality. (Virtual Reality Society, 2017)

4.1.1 První pokusy o virtuální realitu

Pokud se podíváme na virtuální realitu jako na pokus o ošálení našich smyslů, které nás má přesvědčit, že se vyskytujeme na jiném místě než jsme, tak lze jako první pokusy o virtuální realitu označit panoramatické nástěnné malby z devatenáctého století. Většinou zobrazovaly historické výjevy a jejich účel byl vyplnit celé zorné pole diváka a tím ho více vtáhnout do svého obsahu. To mělo způsobit, že se divák cítil, jako by na kresleném místě stál a rozhlížel se. (Virtual Reality Society, 2017)

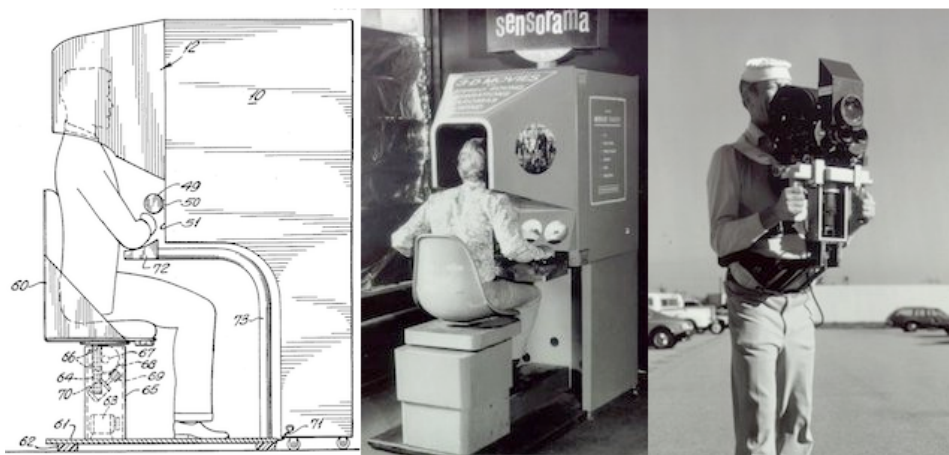
4.1.2 Stereoskopické fotografie a prohlížeče

V roce 1838 výzkum Charlese Wheatstona ukázal, že mozek zpracovává dva různé dvoudimenzionální obrazy z každého oka do jednoho třídimenzionálního. Prohlížení obrázků umístěných vedle sebe ve stereoskopu dává uživateli pocit hloubky a ponoření do obrazu. Lidé časem pomalu, ale jistě vytváří stále efektivnější způsoby, jak stimulovat lidské smysly. Vědci opravdu pokročili v tomto odvětví ve

20. století s příchodem elektroniky a výpočetní techniky. (Virtual Reality Society, 2017)

4.1.3 Sensorama

V polovině padesátých let kameraman Morton Heilig vyvinul stroj nazývaný „Sensorama“. Bylo to malé osobní kino, které stimulovalo všechny smysly a to nejen zrak a sluch. Obsahovalo stereofonní reproduktory, stereoskopický 3D displej, generátor vůni, systém ventilátorů a vibrační židli. Sensorama byl určen k úplnému ponoření uživatele do filmu. Pro svůj vynález natočil také šest krátkých filmů, které sám produkoval a upravoval. (Virtual Reality Society, 2017)



Obr. 14 Sensorama.

Zdroj: Virtual Reality Society, 2016

4.1.4 První VR displej umístěný na hlavě

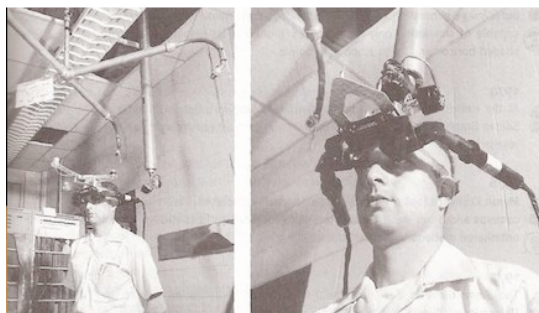
V roce 1960 byl patentovaný první příklad displeje, který byl umístěn v přístroji podobném brýlím. Ten se umísťoval, podobně jako brýle, na hlavu a umožňoval stereoskopické zobrazení 3D obrazu se širokouhlým viděním a stereofonním zvukem. Vynález od firmy Morton Heilig se nazýval „Telesphere Mask“. (Virtual Reality Society, 2017)

4.1.5 První headset se sledováním pohybu

V roce 1961 vyvinuli dva inženýři ze společnosti Philco Corporation prvního předchůdce headsetu pro virtuální realitu, jak je známe dnes. Headset obsahoval video obrazovku pro každé oko zvlášť a magnetický systém sledování pohybu, který byl spojen s kamerou s uzavřeným obvodem. Zařízení se nazývalo „Headsight“ a ve skutečnosti nebylo vytvořeno pro virtuální realitu. Tento termín totiž v této době ještě neexistoval. Headsight byl určen pro armádu, která pomocí něj na dálku ovládala pohyb kamery přirozeným pohybem hlavou. Jednalo se o první krok ve vývoji displeje pro VR, ale chyběla integrace počítačové a obrazové generace. (Virtual Reality Society, 2017)

4.1.6 Damoklův meč

V roce 1968 vytvořil Ivan Sutherland a jeho student Bob Sproull první display pro VR/AR a nazvali jej „Damoklův meč“. Byl připojen k počítači, nikoliv k fotoaparátu. Byl velmi velký, těžký, vypadal strašidelně a pro pohodlné nošení byl zavěšen ze stropě (odtud název). Uživatel musel být k zařízení připojený a počítačová grafika generovala pouze velmi primitivní drátěné modely a objekty. (Virtual Reality Society, 2017)



Obr. 15 Zařízení „Damoklův meč“.
Zdroj: Virtual Reality Society, 2016

4.1.7 Název virtuální realita

Dokonce i po tomto vývoji neexistoval obsáhlý pojem, který by popisoval toto pole působnosti. To se změnilo v roce 1987, kdy Jaron Lanier vytvořil termín „virtuální realita“. Výzkumná oblast nyní měla své jméno. Prostřednictvím své společnosti VPL vyvinul řadu zařízení pro virtuální realitu, včetně rukavic Dataglove a EyePhone brýlí. Byla to první společnost, která prodávala brýle a rukavice pro virtuální realitu. (Virtual Reality Society, 2017)



Obr. 16 Dataglove a EyePhone
Zdroj: Virtual Reality Society, 2016

4.1.8 Arkádové videohry s VR

V roce 1991 se začali vyrábět zařízení pro virtuální realitu, která ji zpřístupňovala široké veřejnosti, přestože vlastnit špičkové zařízení pro VR bylo velmi drahé. Spo-

lečnost Virtuality Group spustila výrobu řady arkádových her a strojů. Hráči si nasadili brýle a hráli na hracích automatech. Některé stroje byli dokonce propojeny a umožňovali zážitek z hraní více hráčů v jedné hře. (Virtual Reality Society, 2017)



Obr. 17 Arkádové hry s podporou VR.
Zdroj: Virtual Reality Society, 2016

4.1.9 Nintendo Virtual Boy

Herní 3D konzola Nintendo Virtual Boy z roku 1995, je považována za první přenosnou konzoli, která dokáže zobrazit skutečnou 3D grafiku. Byl to komerční neúspěch i přes výrazné snížení ceny. Důvody neúspěchu byly nedostatek barev (hry byly pouze červeno-černé), nedostatek softwarové podpory a také nepohodlnost. (Virtual Reality Society, 2017)



Obr. 18 Nintendo Virtual Boy.
Zdroj: Virtual Reality Society, 2016

4.2 Virtuální realita v současnosti

V 21. století zažívá virtuální realita velký rozmach a velmi rychlý vývoj. Je to zásluhou rozvíjení počítačových technologií a hlavně díky výkonným mobilním zařízením. K rozvoji přispívá i vývoj zobrazovacích panelů, které mají čím dál jemnější zobrazení a dokáží vtěsnat větší počet pixelů na palec. To je velmi důležité, protože obrazovka v brýlích pro virtuální realitu je jen pár centimetrů od očí a tak lze snadno pozorovat hustotu zobrazených bodů. Do mobilních zařízení se zabudovává čím dále více senzorů, které lze pro virtuální realitu použít. Fotoaparáty s vysokým rozlišením, hloubkový senzor, senzory pohybu a zrychlení jsou dnes již

standardem. V současnosti existuje na trhu několik řešení virtuální reality od společností Google, Samsung a dalších. (Virtual Reality Society, 2017)

4.2.1 Google Cardboard

Společnost Google přišla s vlastním nástrojem pro virtuální realitu v roce 2014. Pod jménem Google Cardboard se začaly prodávat kartonové brýle, do kterých se vloží chytrý telefon s proximity senzorem a díky němu si lze prohlížet virtuální realitu. Google dokonce uvolnil schéma Cardboard na internet a tak si je může kdokoliv vyrobit doma. Jak vyplývá už z názvu, brýle jsou vyrobené z kartonu s použitím dvou přibližovacích čoček a volitelně jsou zde také samolepící suché zipy, které drží konstrukci pohromadě. Další volitelná výbava brýlí je magnet v levé části nebo tlačítko, které se po stisknutí dotkne displeje a vyvolá nějakou akci, zpravidla pohyb dopředu. (Google Developer, 2017)



Obr. 19 Google Cardboard.
Zdroj: Google, 2017

Konstrukce se složí dle návodu a poté se vloží telefon tak, aby byla středová čára na displeji přesně uprostřed. Brýle se potom nasadí na hlavu a už by mělo fungovat rozhlížení okolo. Požadavky na mobilní zařízení jsou operační systém Android ve verzi 4.2.1 a vyšší a u systému iOS verze 8.0 a vyšší. Windows Phone není podporován. Telefon musí být také vybaven gyroskopem a musí být použit správný formát Cardboard podle velikosti telefonu. (Google Developer, 2017)

Nejnižší ceny lze dosáhnout při výrobě doma z přebytečného kartonu. V případě zakoupení komerčně vyrobených Cardboard se cena v českých obchodech pohybuje od dedadesáti devíti korun až po pět set korun. Cena pět set korun je značně přehnaná, když vezmeme v potaz že se jedná o kousek krabice, gumičku a dvě sklíčka. Zvláště v tomto případě, kdy je plán na brýle dostupný na internetu zdarma. Brýle lze získat více způsoby, jeden z těch nejlevnějších je objednat je, ne-

bo jen jejich část, na zahraničním serveru eBay.com. V květnu 2017 se zde dají sehnat brýle za nejnižší cenu v přepočtu cca 24 Kč s dopravou do České republiky zdarma. Balení obsahuje Cardboard, gumičku, dvě přibližovací sklíčka a dva páry samolepících suchých zipů. Guma na přichycení k hlavě bez nutnosti držení je zde k sehnání od cca 18 Kč a pár sklíček také za cca 18 Kč. Nevýhoda tohoto nákupu je v době dodání, které průměrně dosahuje čtyř až osmi týdnů. (Google Developer, 2017)

4.2.2 Google Daydream

Jak rychle se vyvíjí obor virtuální reality lze demonstrovat na Google Daydream. Na počátku této práce v květnu 2016 ještě tento nástroj nebyl představen a od listopadu roku 2016 už je nabízen k prodeji. Oproti předchozímu Google Cardboard má Daydream zabudovanou podporu přímo v operačním systému Android od verze 7.1 Nougat. Díky větším nárokům tento headset nepodporuje tak velké množství mobilních zařízení. Aktuálně to jsou pouze tyto telefony: Huawei Mate 9 Pro, ZTE Axon 7, Google Pixel, Motorola Moto Z a v brzké době má přibýt také Asus ZenFone AR. Daydream tak oproti Cardboard nemá za cíl představit virtuální realitu širokému spektru uživatelů. (Google Developer, 2017)



Obr. 20 Google Daydream.
Zdroj: Google, 2017

Cena oproti Cardboard je vyšší, zakoupit jej lze za 79 USD, v českých obchodech cena startuje na dvou a půl tisících. Co tedy nabízí Daydream navíc oproti Cardboard, že stojí o tolik více? Měl by být lehký a hlavně mnohem více komfortní na nošení, díky prodyšnému příjemnému materiálu a také má být vodou omytelný. Hlavní výhodou však tvoří speciální ovladač, který má intuitivní ovládání a umožňuje uživateli interakci s virtuální scénou. Ovladač obsahuje pouze několik tlačítek,

právě z důvodu, že když má uživatel nasazeny brýle, tak na ovladač nevidí. Musí tak být schopný ovládat prvky nahmatat poslepu. Ovladač obsahuje chytré senzory, které zpracovávají pohyb a gesta hráče. (Google Developer, 2017)

Díky možnosti interakce se scénou, posouvá Daydream virtuální realitu na další úroveň. Ovladač přináší zcela nové možnosti a otevírá vývojářům dosud nemožné řešení. Nevýhodou ovšem je podpora velmi malého počtu telefonů, která se snad časem zlepši. (Google Developer, 2017)

4.2.3 HTC Vive

Jedno z nejlepších řešení virtuální reality v současnosti je systém HTC Vive, který je vyvíjen ve spolupráci s herní společností Valve. Zakoupit jej lze od dubna 2016. Displej v brýlích má velmi vysoké rozlišení 2160 x 1200 pixelů, speciální senzory pro snímání pohybu uživatele také malé plošky, které jsou sledovány infračervenými snímači rozmístěnými v místnosti. Díky tomu se lze během hraní pohybovat na ploše až pět krát pět metrů. (Martin Pultzner, 2017)



Obr. 21 HTC Vive.
Zdroj: Forbes.cz, 2017

K helmě jsou dodávány dva speciální ovladače, kterými lze interagovat s předměty ve virtuální scéně. Jsou to právě tyto ovladače, které posouvají zážitek na novou úroveň. Headset obsahuje sám o sobě pouze displej. Programy a hry je nutné spouštět na počítači, který do brýlí přenáší obraz. Nároky na výpočetní výkon počítače jsou velmi vysoké a jedná se tak o poněkud dražší řešení. Sada HTC Vive sama o sobě stojí cca 26 tisíc Kč a obsahuje brýle, dva speciální ovladače a také dvojici infračervených snímačů. (Martin Pultzner, 2017)

4.2.4 Oculus Rift

Přímou konkurenci pro HTC Vive představuje Oculus Rift. Tento systém původně začínal jako Kickstarter projekt, který byl později odkoupen společností Facebook. Mark Zuckerberg ze společnosti Facebook koupil tento projekt za zhruba dvě miliardy dolarů. Na headsetu se pracuje od roku 2014 a finální verze šla do prodeje na jaře roku 2016 za cenu kolem 20 tisíc Kč. Oculus nabízí o trochu lepší kvalitu obrazu, díky kvalitnějším čočkám pro zobrazení 3D. Brýle v sobě mají integrované reproduktory pro lepší zážitek i po zvukové stránce. K Oculus Rift jsou také dodávány dva ovladače, podobné těm od HTC Vive. K systému patří také externí infračervený snímač, ten však nedosahuje stejných výsledků přesnosti jako HTC Vive. Stejně jako předchozí systém i Oculus vyžaduje pro svůj běh výkonný počítač. (Martin Pultzner, 2017)



Obr. 22 Oculus Rift.
Zdroj: Forbes.cz, 2017

4.2.5 Sony PlayStation VR

Dalším zajímavým řešením virtuální reality je PlayStation VR od firmy Sony. Na rozdíl od předchozích dvou řešení zde hry nezpracovává připojený počítač, ale herní konzole PlayStation 4. Ke konci roku 2016 již bylo k dispozici 50 her pro toto herní zařízení. Pomocí brýlí lze hrát i hry, které nejsou určeny pro virtuální realitu. Brýle v tom případě slouží jako zobrazovací zařízení, které poskytne pocitově větší zobrazovací plochu. Playstation VR nabízí, stejně jako konkurence, zachycení pohybu hráče pomocí infračervených senzorů. Zabudované reproduktory zde chybí, ale je k dispozici standardní 3,5 mm jack konektor, pro připojení libovolných sluchátek. (Martin Pultzner, 2017)

Sony se snaží o rozšíření virtuální reality o hru více hráčů, kteří společně sdílí zážitky ze hry. Jednou z možností jak toho docílit je současné zobrazení hry v brýlích a v televizní obrazovce. Jeden hráč tak může hru hrát s brýlemi a druhý klasicky s ovladačem před televizí. Tímto způsobem mohou také pozorovatelé sledovat hru hráče s brýlemi. Oproti konkurenčním systémům je PlayStation VR ochuzen o možnost pohybu po místnosti a pokročilé ovladače, takže neposkytuje tak kvalitní zážitek. Nabízí však nižší cenu, která se v České republice pohybuje okolo jedenácti tisíc. Je nutné však mít na mysli, že se jedná o doplněk k PlayStation 4, bez něj je PlayStation VR nepoužitelný. (Martin Pultzner, 2017)



Obr. 23 Sony PlayStation VR.
Zdroj: Forbes.cz, 2017

4.3 Pohyb ve virtuální realitě

Z předchozího textu jasně vyplývá, že řešení pohybu ve virtuální scéně je velmi důležité, protože umocňuje zážitek ze hry. Možnost interagovat s okolními objekty posouvá hranice možností o kus dál.

4.3.1 Existující řešení pohybu ve virtuální scéně

Ze současných systémů mají všechny svá specifická řešení pohybu, většinou pomocí speciálních ovladačů. Ze tří systémů HTC Vive, Oculus Rift a PlayStation VR vychází dle hodnocení nejhůře PlayStation VR, právě díky nejméně dokonalému řešení pohybu a interakce ve scéně. (Martin Pultzner, 2017)

Jediný systém pro virtuální realitu s velmi špatným řešením pohybu a interakce z výše zmíněných je Google Cardboard. V nejlevnější verzi neumožňuje žádnou interakci, kromě softwarových řešení. To spočívá například ve sledování určitého objektu po určitou dobu (zpravidla několik vteřin) a tím je potvrzena daná akce.

Pro lepší uživatelskou přívětivost se zobrazuje kolem kurzoru odpočítadlo, které ukazuje dobu potřebnou k potvrzení. Pokud jsou brýle doplněny o magnet, který ovlivňuje NFC nebo tlačítko, které se po zmáčknutí dotkne displeje, získá se tak jedna akční klávesa. Ta je využívána různě, ale ve většině her, kde se lze pohybovat, je určena k pohybu dopředu. Směr pohybu je na pevně dán směrem, kterým se hráč dívá. (Google Developer, 2017)

Existují také různé ovladače, které lze dokoupit a připojit je k telefonu pomocí bluetooth. Cena těchto zařízení je v rámci stovek až tisíců korun. Výhoda těchto zařízení je v tom, že bluetooth je na všech chytrých mobilech standardem, nevýhodou je, že ne každý si takové zařízení koupí a nosí jej u sebe. (Unofficial Cardboard, 2017)

5 Metodika práce

5.1 Motivace

Systémy pro objevování virtuální reality byly dlouhou dobu finančně náročné a nebyly tudíž příliš dostupné. To se změnilo, když Google přišel s levným řešením Cardboard, které zpropagovalo virtuální realitu a vzbudilo zájem široké veřejnosti. Problémem však zůstává řešení pohybu a interakce ve scéně, které mají dobře vyřešené až dražší systémy. Po vyzkoušení mnoha aplikací a her pro Cardboard je jasně viditelné omezení nemožnosti interakce. Je zde možnost pouze jednoho akčního tlačítka, což je nedostačující. V případě, že ke Cardboard není zakoupen magnet, který využívá NFC nebo dotekové tlačítko, tak je jakákoliv interakce nemožná. Tento fakt do značné míry omezuje zážitek ze hry a to je velká škoda.

5.2 Návrh řešení

Tato práce se zabývá návrhem řešení způsobu pohybu v mobilních aplikacích pro virtuální realitu. Jelikož nejhůře vyřešený způsob interakce má jednoznačně Cardboard, zaměří se tato práce právě na něj. Dalším důvodem, proč zrovna Cardboard, je nízká cena, která zpřístupňuje virtuální realitu každému, kdo vlastní chytrý telefon s gyroskopem. Cílem je vylepšit zážitek a tím i uživatelskou zkušenost pro hráče. Jelikož hlavním kritériem je dostupnost pro širokou veřejnost, řešení pohybu bude softwarové. Jelikož má již dnes skoro každý v kapse chytrý telefon, řešení bude spočívat v ovládní pohybu pomocí druhého telefonu, který bude s telefonem vloženým do Cardboard komunikovat bezdrátově. Velký důraz také musí být kladen i na jednoduchost a intuitivnost. Pokud má totiž uživatel nasazené na hlavě brýle, tak nevidí na ovládací prvky na druhém telefonu.

5.3 Engine

Pro vývoj aplikace, která bude sloužit jako vzorové řešení problému pohybu ve virtuální scéně bude použit engine Unity. V době započtení této práce byla jediná možnost použít tento engine, protože Google vydal Cardboard SDK právě pro něj. V současnosti už jsou ke stažení SDK i pro Android/NDK, iOS a Unreal. Dalším důvodem pro výběr Unity je to, že je stále nejpoužívanější engine a na internetu se dá najít spousta tutoriálů, rad a návodů. Unity podporuje široké spektrum platforem, takže je zde možnost potenciálního rozšíření na další platformy. Unity má také zabudované knihovny pro práci se sítí, která bude v této práci potřeba na propojení obou mobilních zařízení.

5.4 Vývojové prostředí

Jelikož bude na tuto práci používán engine Unity, používání jakéhokoliv jiného vývojového prostředí než samotného Unity editoru nepřipadá v úvahu. Unity má v instalačním balíku předinstalovaný program MonoDevelop, což je samostatné vývojové prostředí pro psaní skriptů. Toto prostředí lze volitelně změnit a v různých video návodech tak uživatelé učinili. Pokud někdo používá jiné prostředí než MonoDevelop, tak je to zpravidla Visual Studio od společnosti Microsoft. Záleží tak jen na vývojáři, na které prostředí je zvyklý a víc mu vyhovuje. V této práci bude používán výchozí editor MonoDevelop.

5.5 Operační systém

Jako cílový mobilní operační systém pro tuto práci byl zvolen Android. Z pohledu dostupnosti jde o nejrozšířenější mobilní systém. Nejdostupnější je i co se týče ceny ve srovnání s mobilními telefony značky Apple, které používají systém iOS. Ten je totiž společně s Androidem jediným podporovaným systémem pro Cardboard. Windows Phone ani jiné systémy zatím podporu nemají a podle dostupných informací se ani nechystá.

Pro vytváření Android aplikací v Unity je potřeba nainstalovat Android Studio, v něm stáhnout Android SDK Tools a kompilátor pro Android. Dříve Google umožňoval stažení pouze SDK tools a kompilátoru samostatně, nyní se však snaží donutit vývojáře stáhnout a nainstalovat Android Studio a tak neposkytují poslední verzi samostatných nástrojů ke stažení. Samostatné nástroje lze stáhnout pouze několik verzí staré.

5.6 Programovací jazyk

Unity podporuje pouze dva programovací jazyky a těmi jsou C# a Javascript. Statisticky je mnohem více používanější C# viz kapitola o Unity. Z toho důvodu bude použit i v této práci. Výhoda použití tohoto jazyka plyne z jeho využívání vývojáři. C# je použit v mnohem více příkladech a tutoriálech. Někteří vývojáři na fórech Unity doporučují C# protože je mnohem robustnější i když je oproti Javascriptu těžší na naučení.

6 Implementace aplikace

6.1 Příprava

K implementaci této aplikace bude zapotřebí nainstalovat nejdřív potřebný software. Na stránkách Unity lze stáhnout poslední verzi Unity Personal zdarma. Instalace probíhá standardně dle operačního systému (operační systém použitý při tvorbě této práce je macOS Sierra).

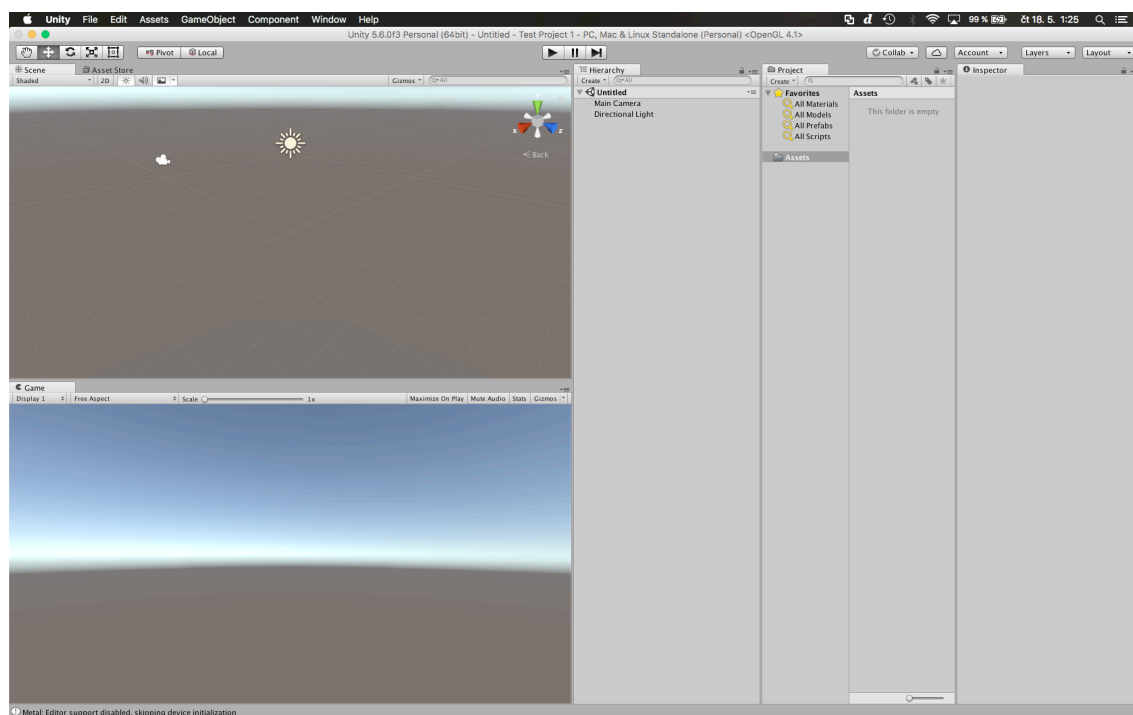
Pro vytváření aplikací pro mobilní systém Android je potřeba Android SDK, který lze stáhnout pomocí Android Studio. Instalátor pro macOS lze stáhnout na vývojářských stránkách tohoto vývojového prostředí. Po instalaci je třeba jít do nastavení, do SDK manažera. Zde se vyberou verze systému Android, pro které je vyžadováno vytváření aplikací. Ve stejné obrazovce v panelu SDK Tools je třeba zaškrtnout Android SDK Tools a Android SDK Platform-Tools. Posledním balíčkem k zaškrtnutí je NDK. Po potvrzení tlačítkem OK, se spustí stahování a instalace vybraných balíčků. Po úspěšné instalaci lze z této stránky zkopírovat nebo opsat cestu k těmto službám a poté je možné Android Studio ukončit.

Nyní je třeba stáhnout balíček Google Cardboard SDK. Nejsnáze ho lze nalézt zadáním tohoto výrazu do vyhledávače Google. Hned přes první vrácený odkaz se dostaneme do vývojářské sekce Cardboardu a stáhneme „Google VR SDK for Unity“. Tento balíček budeme potřebovat až později v Unity, ale je dobré ho mít připravený.

6.2 Nastavení

Při spuštění Unity se zobrazí uvítací obrazovka, která obsahuje seznam projektů, cestu k nim a verzi, ve které byl projekt vytvořen. Jsou zde také tlačítka na vytvoření nového projektu, otevření projektu, který není v seznamu a přihlášení. Po přihlášení lze otevírat i projekty uložené v cloudu. Při vytváření nového projektu je nutné vyplnit název, umístění a zvolit typ grafiky. Lze vybírat buď z 2D nebo 3D aplikace. V této chvíli lze také zvolit, jaké balíčky budou do projektu už v úvodní části naimportovány.

Po vytvoření nového 3D projektu se zobrazí prostředí, ve kterém se aplikace vyvíjí. Ta je rozdělena na několik částí. V levé části jsou zobrazeny náhledy. Spodní náhled zobrazuje pohled na scénu tak, jak se bude zobrazovat přímo v zařízení. Horní náhled slouží pro zobrazení scény, jedná se spíše o pracovní náhled. Oproti spodnímu se v něm lze posouvat, rotovat, přibližovat a oddalovat. Stejně úkony zde lze provádět i s objekty ve scéně. Přepínání mezi posuven, rotací a dalšími úkony je umístěno vlevo nad náhledy.



Obr. 24 Vývojové prostředí Unity.

Pravá část obsahuje tři sloupce. V prvním sloupci je zobrazen seznam všech objektů ve scéně, včetně jejich umístění v hierarchii. Druhý sloupec obsahuje seznam oblíbených položek a složku Assets. Ta je umístěna v adresáři s vytvořenou aplikací a obsahuje všechny entity, které je potřeba uložit do souboru. Kvůli lepší přehlednosti je vhodné zde vytvořit složky, do kterých se budou ukládat vytvořené soubory podle jejich typu.

Prefabs – Do této složky se budou ukládat pro lepší přehlednost všechny prefabrikáty. Jsou to znovupoužitelné netriviální objekty, které se ve hře vyskytují vícekrát nebo je potřeba se na ně odkazovat ve skriptech.

Scenes – Zde se budou ukládat veškeré vytvořené scény. Jelikož je při vývoji potřeba se často mezi scénami přepínat, je lepší mít je všechny v jedné složce. To umožňuje rychlejší nalezení a zefektivňuje práci.

Scripts – Tato složka bude obsahovat všechny vytvořené skripty. Pokud by bylo ve hře větší množství skriptů, tak lze vytvořit další podsložky a třídit je dle různých kritérií. V této ukázkové aplikaci bude stačit jen složka Scripts.

Další složky prozatím nebudou potřeba, vytvoří se samy při importu dalších knihoven, které jsou v této práci použity.

V posledním sloupci se nachází inspektor. Zobrazuje vždy informace o vybraném objektu. Pokud je vybrán nějaký game object, tak v inspektoru můžeme měnit jeho vlastnosti jako je pozice, rotace a měřítko. Zobrazují se zde také skripty a vlastnosti připnuté k tomuto objektu a jejich veřejné rozhraní, kterým lze připnutou vlastnost měnit.

Tlačítka play a pause se zapíná a pauzuje aplikace v prostředí editoru.

6.3 Použité knihovny a rozšíření

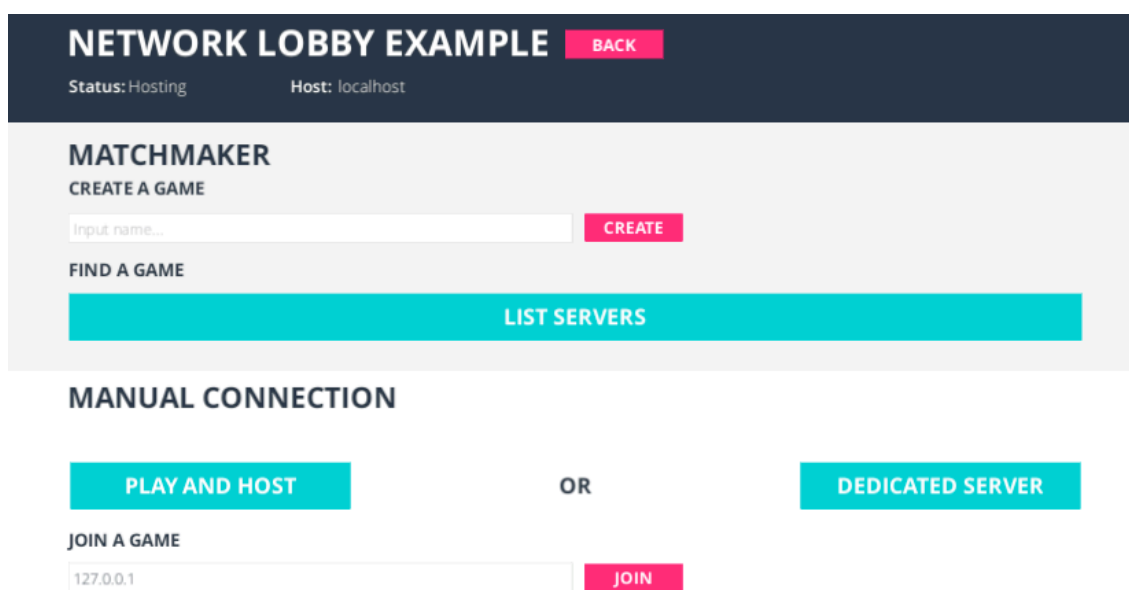
V této části práce se nachází seznam použitých knihoven s popisem jejich funkce.

6.3.1 Standard Assets

V Unity existuje sada zabudovaných rozšíření, které lze snadno do vlastního projektu připojit. Tyto rozšíření jsou široce využívána uživateli a proto jsou zde již v základu. Mezi standartní assety patří 2D, Cameras, Characters, CrossPlatformInput, Effects, Environment, ParticleSystems, Prototyping, Utility a Vehicles.

6.3.2 Network Lobby

Toto rozšíření umožňuje snadné vytvoření lobby pro hru více hráčů a lze jej stáhnout v Asset Store. Má kompletně zpracovaný responzivní design, nastýlovaný vzhled tlačítek, nápisů a zadávacích polí. Umožňuje využití Matchmakeru, dedikovaného serveru nebo přímého propojení. V této práci bude použito přímé propojení. Hostem bude mobilní zařízení, které bude vloženo do Google Cardboard a klientem bude ovládací zařízení.



Obr. 25 Network Lobby manager.

Kromě toho, že tento plugin obstarává funkci lobby, je také zároveň nadstavbou Network manageru. To poskytuje high level API pro veškeré síťové funkce. Zajišťuje jednoduché rozhraní pro vytváření online multiplayer her.

6.3.3 CNControls

Rozšíření CNControls, které je zdarma ke stažení v Asset Storů zajišťuje pokročilé ovládací prvky. K tomuto rozšíření je připojen jednoduchý manuál a video, které

pomůže s nastavením. Nasazení do projektu je velmi jednoduché. Stačí asset importovat, nahrát do scény kreslící plátno Canvas a poté na něj přetáhnout vybraný typ ovládacího prvku. Ve skriptech stačí použít místo třídy Input třídu CnInputManger. Ostatní zůstává stejné oproti využití klasického vstupu. Toto rozšíření je v této práci použito hlavně kvůli joysticku s nastavitelnou křivkou záběru. Tento joystick je inspirován joysticky z ovladačů herních konzolí a poskytuje tak uživateli známý způsob ovládání.

6.3.4 GoogleVR

Plugin který zajišťuje API pro Google Cardboard a Google Daydream se nazývá GoogleVR. Umožňuje do aplikace vložit speciální typ stereoskopické kamery se všemi potřebnými skripty, které zajišťují výpočty. Od posledních verzí se Unity snaží tento plugin do sebe začlenit a nabízet tak nativní podporu. Použití je velmi jednoduché. Stačí knihovnu naimportovat, poté je nutné jít do nastavení projektu, zapnout podporu virtuální reality a vybrat jako cílové zařízení Google Cardboard. Dříve se musel přidávat do scény prefab virtuální kamery, nyní už to není nutné díky integraci přímo do Unity. Po zapnutí podpory virtuální reality v nastavení, se zapne stereoskopická kamera pro všechny scény. V této práci je však potřeba virtuální kamera pouze ve scéně 3D Scene. Proto je třeba vytvořit dva skripty, které se připojí na nějaký objekt ve scéně, třeba na hlavní kameru. Poté se do metody Start vloží kód, který vypne nebo zapne mód virtuální reality. Problém také nastává s orientací displeje, který je ve virtuálním módu vždy na ležato vlevo. Aby byla orientace ve všech scénách stejná, stačí přidat do stejného skriptu vynucení orientace pomocí speciálního příkazu.

6.4 Prefabs

V aplikaci je vytvořeno a použito několik prefabrikátů. Jedná se o objekty, které jsou uloženy v souboru a lze je opakovaně ve scéně používat. Úprava prefabrikátu se projeví na všech jeho instancích. Pokud nějakou instanci ve scéně změním, už se na ni nevztahuje aktualizace změn z hlavního předka.

6.4.1 Player

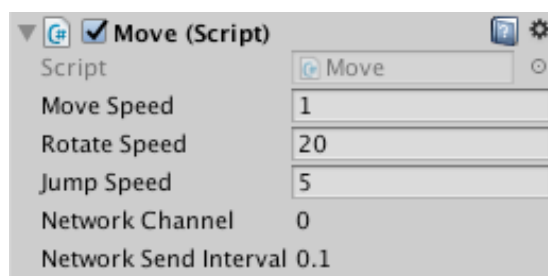
Prefab s názvem Player označuje povinný objekt, vyžadovaný Network managem, který ve scéně zastupuje instanci každého připojeného hráče. Běžně se na tento objekt připojují různé ovládací skripty, které snímají vstupy od hráče a poté pohybují, či jinak interagují s objekty ve hře. Každý player prefab na sobě musí mít skript Network Identity, který mu přiřazuje jedinečnou identitu v síti. Také je mu nastavený příznak Local Player Authority, což určuje, že daná instance je vlastněna lokálním hráčem.

Hráčské prefaby se do hry vkládají pomocí spawn metody Round Robin. Ta najde v scéně objekty, které na sobě mají připnutou vlastnost NetworkStartPosition. Mohou to být prázdné game objekty, pouze s tímto příznakem. Přesná pozice

a rotace tohoto bodu pak označuje bod, na kterém bude hráčův klon vytvořen. Důležitá je taková věc, že různé instance hráčů se nebudou vytvářet na stejném místě, což by způsobilo různé problémy, například s kolizí objektů.

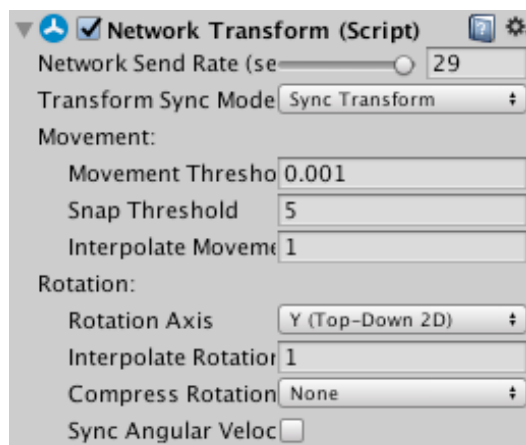
6.4.2 Ghost

Tento stěžejní objekt aplikace má několik speciálních funkcí. Je to společný objekt pro hosta i klienta, který pomocí skriptu snímá vstup uživatele a podle toho se pohybuje scénou. V každé scéně, kde je tento objekt umístěn se vyskytuje i skript přiřazený na kameru, který synchronizuje pohyb ducha s kamerou. Duchem je nazýván proto, že nemá žádný fyzický tvar, je to prázdný game object.



Obr. 26 Skript Move a jeho nastavení.

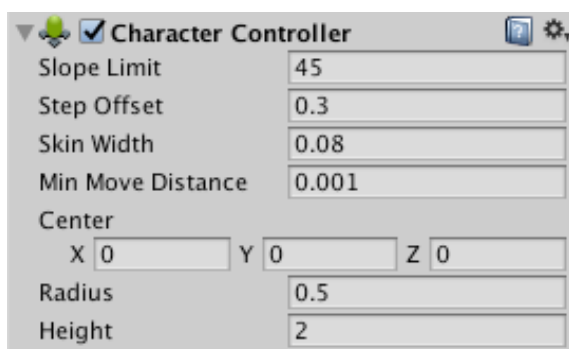
Skript, pomocí kterého se duch pohybuje, obsahuje nastavitelné parametry. Ty lze měnit z inspektoru v editoru. Nastavovat lze rychlost pohybu, rychlost rotace a sílu s jakou hráč vyskočí, při stisku Jump tlačítka. Tyto vlastnosti lze pomocí těchto parametrů měnit za běhu bez nutnosti úpravy nebo znalosti skriptu.



Obr. 27 Network Transform skript.

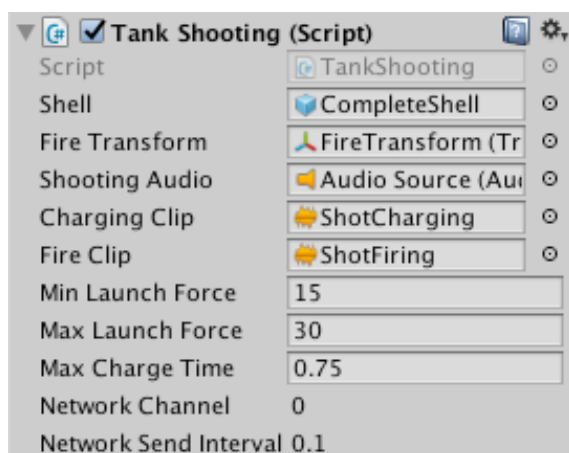
Pomocí skriptu Network Transform se přenáší pohyb objektu ke všem připojeným klientům. To způsobí, že pokud se pohne duch na klientském zařízení, tak se pohne i duch v hlavním zařízení v Cardboard. Jelikož je pohyb ducha sledován kamerou, tak to vytvoří dojem chůze pro uživatele s nasazenými Cardboard. Jelikož je potřeba

ba co nejplynulejší přenesení pohybu, zvýší se Network Send Rate na nejvyšší možnou hodnotu a to dvacet devět. Hodnoty pohybu zůstanou nezměněné. U rotace zapneme přenos pouze v ose Y, jelikož nás ostatní osy nezajímají a přenášení pohybu je výpočetně náročné a zpomaluje aplikaci. Při testování komprese rotace nebylo dosaženo dobrých výsledků. Aplikace se potom vizuálně trhala a nepůsobilo to dobře. Proto je komprese vypnutá.



Obr. 28 Nastavení Character Controller.

Důležitá komponenta na tomto objektu se nazývá Character Controller. Ta umožňuje nastavit objektu vlastní gravitaci, pohyb i možnost skoku. Vytvoří kolizní plášť ve tvaru kapse o výšce dva a průměru nula celá pět. Duch je tak díky tomu sice stále neviditelný, ale má už i tvar a nemůže nadále procházet objekty, které také mají nastavenou kolizní síť. V praxi to znamená, že se duch nepropadne podlahou scény a vráží do objektů ve scéně, jako je tomu u většiny her. Slope limit udává, jaký maximální úhel je schopen duch přejít. S tímto nastavením je duch schopný pokračovat v chůzi po objektech až do sklonu čtyřicet pět stupňů. Ostatní nastavení je ponecháno výchozí.



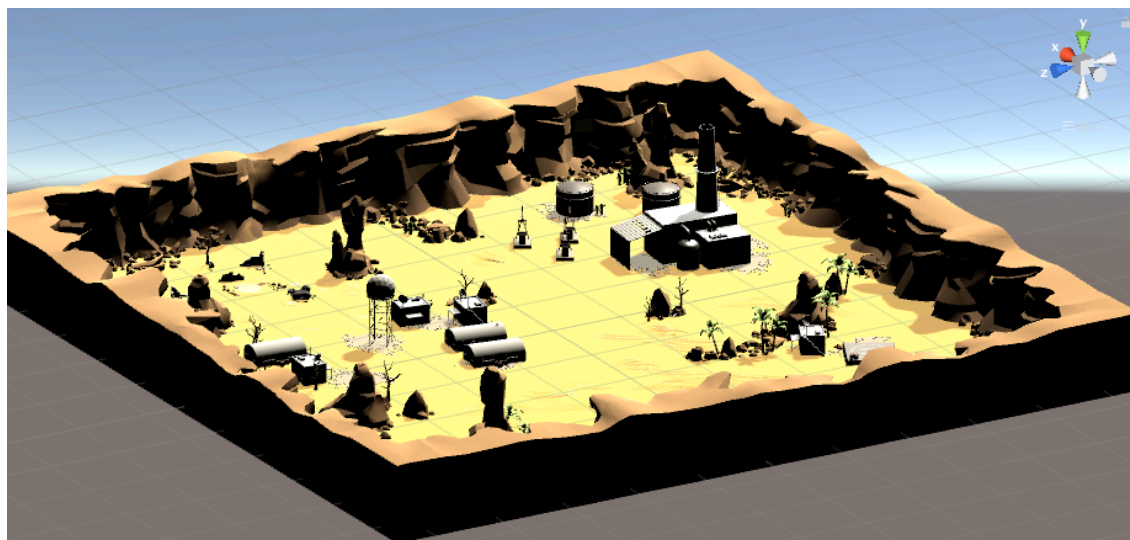
Obr. 29 Tank Shooting skript

Jak bude popsáno dále v práci, tak ovládací prvky budou umožňovat chůzi vpřed a vzad, rotaci, skok a také je zde přidána střelba. Tu řeší skript Tank Shooting. Jed-

ná se o převzatý skript z tutoriálu „Tanks tutorial“, který byl původně předveden na přednášce „Unite Boston 2015“. (Unity, 2017) Skript je upraven pro potřeby této práce. Původně umožňoval i správu života tanků a ubírání pomocí výbuchů. To však v této práci není potřeba. Do veřejné části inspektoru se vloží prefab střely, pozice ze které bude střela vypouštěna, poté zvuky střelení, nabíjení a výbuchu. Síla střelby je dána pomocí doby, po kterou je tlačítko pro střelbu stisknuté. Z inspektoru lze nastavit minimální sílu střely, aby se zabránilo spadnutí střely hned pod hráče, maximální možnou sílu střely a také maximální dobu nabíjení. Střelba je zde pouze z testovacích důvodů, aby se po stisknutí tlačítka dalo snadno ověřit, zda bylo skutečně stisknuto a vyvolalo danou akci.

6.4.3 LevelArt

Tento model scény je převzatý z výše zmíněného tutoriálu o tvorbě hry se střílejšími tanky pro více hráčů. Model je zde místo holé podlahy pro lepší vizuální představu o pohybu hráče. Na jednobarevné podlaze se hůře rozlišuje, jestli se hráč vůbec hýbe a rotuje. V této scéně je pohyb jasně patrný díky okolním objektům. Skládá se z pouště ohraničené skálami, palem a další vegetace, několika vojenských a průmyslových budov a také náhodně se povalujících kamenů a skalek.



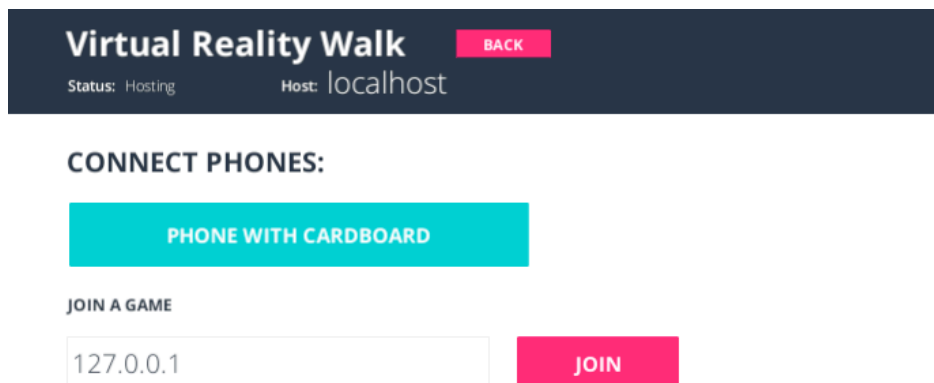
Obr. 30 LeverArt scéna z tutoriálu o tvorbě hry s tanky.

6.5 Scény

Celá aplikace se skládá ze tří scén. Scéna Main je načtena jako první při spuštění aplikace. Další dvě scény 3D Scene a Controller jsou načteny, na každém ze dvou zařízení je jedna z nich.

6.5.1 Main

Main je výchozí scénou celé aplikace a načítá se jako první po přehrání animace „Made with Unity“.



Obr. 31 Scéna Main.

Využívá se zde vzhledu Lobby Manageru, který je upraven pro potřeby této aplikace. Nahoře se nachází název aplikace, status zařízení a IP adresa. V případě zanoření, se zde skriptem zobrazí tlačítko pro navigaci zpět. Rozhraní je velmi jednoduché. Na mobilním zařízení, které bude vloženo do Google Cardboard se zvolí možnost „Phone with Cardboard“.



Obr. 32 Lobby.

Následuje přesun uživatele do lobby, kde se čeká na připojení druhého mobilu. Zobrazí se informace o tom, že se jedná o hru s přímým připojením a je potřeba být

s oběma zařízeními ve stejné síti a použít IP adresu zobrazenou nahoře v aplikaci pro vytvoření síťového spojení.

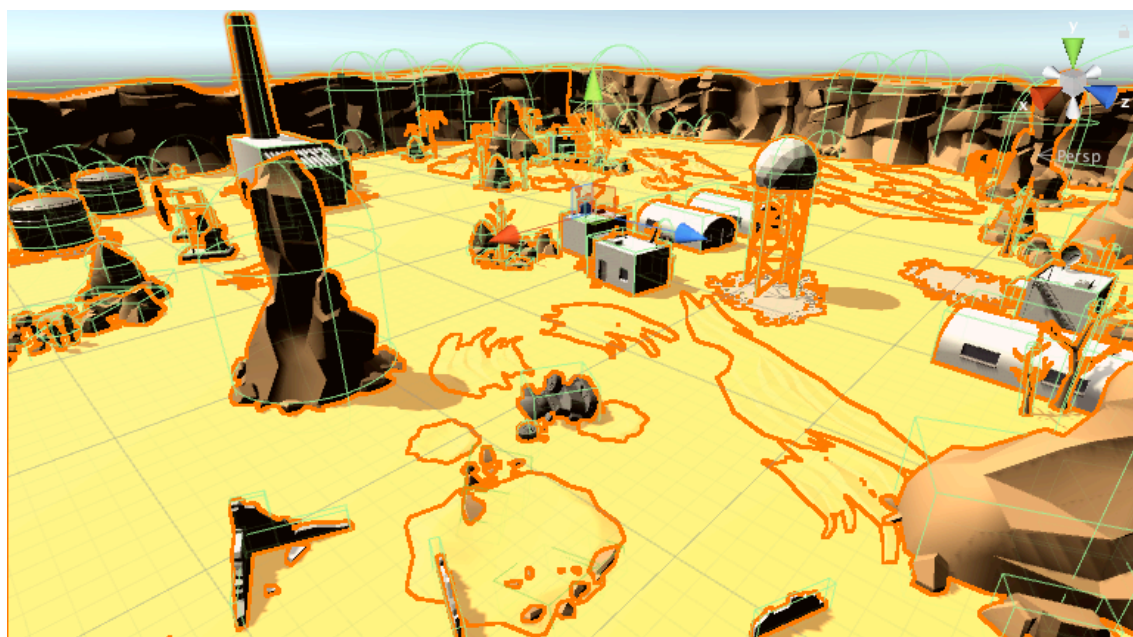


Obr. 33 Připojení obou mobilních zařízení.

Po zadání IP adresy z hlavního zařízení a připojení, se zobrazí druhý telefon v seznamu. V tomto okamžiku jsou obě zařízení připravená na další krok. Nyní je třeba na obou zařízeních potvrdit spojení a vytvořit tak požadavek na spuštění aplikace. To se provede stisknutím tlačítka s nápisem „Join“. Skript obstarávající síťové protokoly vytvoří spojení ve kterém je telefon s Cardboard zároveň hostem i klientem a druhý telefon je klientem. Dle nastavení Lobby manažera se po ustálení spojení změní scéna na scénu Controller. Díky úpravě se v zařízení, které je zároveň hostem změní scéna na 3D scénu. Nyní je na každém zařízení načtena správná scéna a zařízení spolu mohou komunikovat po síti.

6.5.2 3D Scene

Tato scéna je určena pro mobil s Google Cardboard a proto je v ní, jako v jediné, skriptem zapnuta podpora virtuální reality. Ten je připnutý na hlavní kameru a ta obsahuje ještě jeden skript. Ten obstarává připnutí kamery na pozici ducha. Přesněji je to prováděno tak, že v každém snímku se vezme pozice a rotace ducha a na stejného hodnoty se nastaví i kamera. Tento pohyb není přímo na kameře z důvodu, že Unity neumožňuje, aby kamera byla síťový prvek. To je ale potřeba, pokud chceme pohyb synchronizovat po síti, proto je tu řešení s duchem jako dalším objektem, který už síťovou identitu má. Nachází se zde také 3D objekty, tvořící celou scénu, kterou lze procházet.



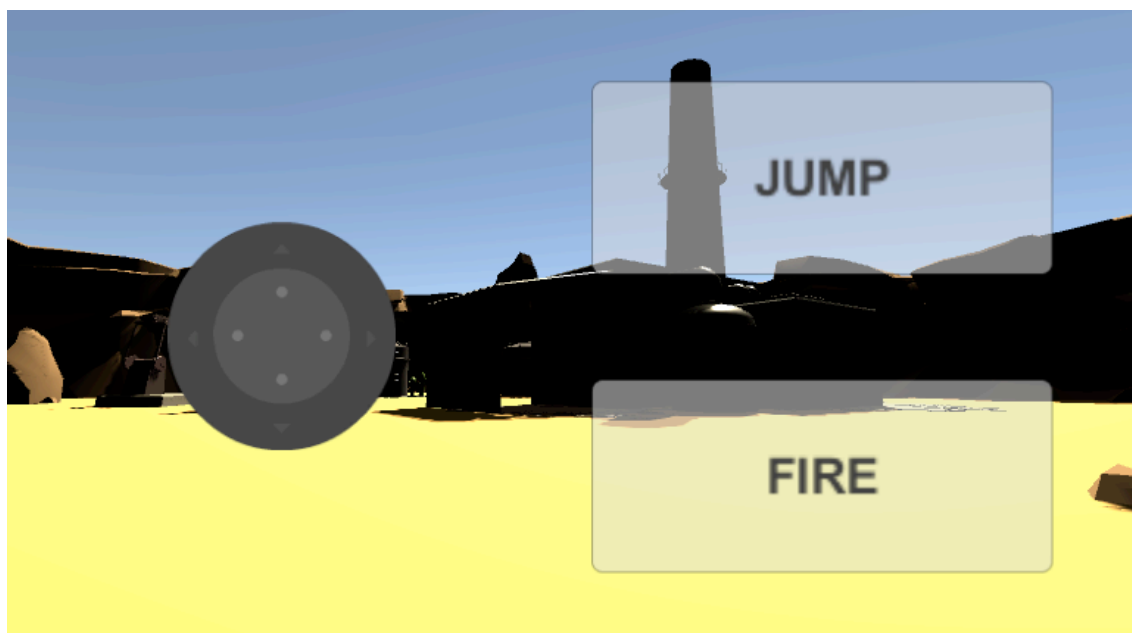
Obr. 34 3D scéna.

Na obrázku je vidět podoba 3D scény. Scéna byla převzata z Unity tutoriálu „Tanks tutorial“. Oranžové křivky označují hranice jednotlivých objektů a tenčí zelené čáry označují jednoduché kolizní objekty. Ty tvoří samostatnou neviditelnou vrstvu, která slouží k tomu, aby hráč při pohybování ve scéně neprocházel skrz objekty, ale vrazil do nich. Hráč sám o sobě má kolem sebe také kolizní objekt, proto nemůže projít skrz. Kolizní objekty se nazývají jednoduchými, protože nemají stejné hranice jako objekty, které zastupují. Jako kolizní objekty se používají hlavně kvádry, koule a kapsle. Výpočty kolizí jsou výpočetně náročné a to je důvod, proč se nepoužívají samotné objekty, ale jsou obaleny zjednodušenými kolizními objekty.

První verze této aplikace neobsahovala kolizní objekty a nebyla zde gravitace. Místo tlačítek pro střelbu a skok zde byl druhý joystick. Tato verze by mohla sloužit pro prohlížení 3D scény s volným pohybem.

6.5.3 Controller

Scéna s názvem Controller je navržena jako ovladač 3D scény. Obsahuje kameru s připnutými skripty na vypnutí módu virtuální reality a také skript na sledování ducha kamerou. Pokud hraje pouze jeden hráč, tak tuto funkci nevyužije, protože na ovládací mobil nevidí, ale představíme-li si například scénář, ve kterém někdo prezentuje nějaké modely ve virtuální scéně, může se místo zákazníka pohybovat a uvidí kam s ním jde. Scéna obsahuje také 3D model základny, takže je vidět za ovládacími prvky stejný výhled, jaký je v Cardboard.



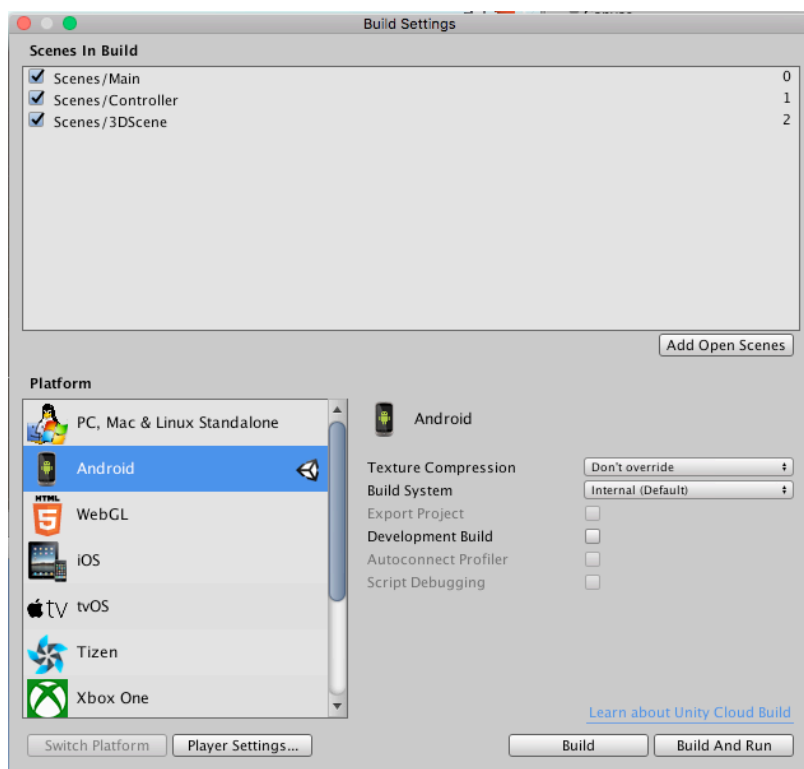
Obr. 35 Scéna s ovladačem.

Ovládací prvky jsou umístěny na plátně, které je vždy zobrazeno před kamerou. Díky tomu je ovládání pořád na stejném místě i když se kamera pohybuje. Joystick má nastavitelnou citlivost a jako výchozí je nastavena lineární. Tlačítka jsou navržena dostatečně velká, aby nedošlo k záměně při používání s brýlemi a zároveň nejsou přes celou obrazovku, aby šlo dobře vidět scénu za nimi. Původně je barva pozadí tlačítek šedá neprůhledná, protože průhlednost nelze nastavit z inspektoru. Průhlednost je proto dodatečně přidána skriptem, do podoby jak je vidět na obrázku nad tímto textem. Díky tomu je lépe vidět na scénu v pozadí, při zachování dobré čitelnosti.

Ve verzi s volným pohybem tato scéna obsahovala místo 3D modelů jednobarevné pozadí a dva joysticky. Levý joystick funguje stejně jako v aktuální verzi a druhý nahrazuje tlačítka pro skok a výstřel. Pravý joystick snímá pouze vertikální osu a pohybuje s hráčem nahoru a dolů. Jelikož už byla vstupní vertikální osa používána, bylo nutné vytvořit novou v nastavení Input Manager.

6.6 Sestavení aplikace

Vývoj této aplikace probíhal po většinu času pro operační systém macOS. Jelikož byla tato práce psána na počítači s tímto systémem, přináší to značnou úsporu času. Při každém testování se nemusí zdlouhavě nahrávat aplikace do mobilů, ale jednoduše se vytvoří sestavení pro daný systém a druhá kopie aplikace se spustí přímo v editoru Unity.



Obr. 36 Build Settings.

Změna platformy na Android je v Unity velmi jednoduchá. Stačí v nastavení projektu vložit všechny použité scény do seznamu scén, vybrat požadovanou platformu a stisknout „Switch Platform“. Důležité je pořadí scén v seznamu. Každá scéna má přiděleno své číslo a nultá je výchozí scéna, která se spustí jako první. Přepínání scén pomocí skriptu může být provedeno zadáním názvu scény, nebo jeho čísla. Proto je vhodné během práce neměnit pořadí scén. Pokud je to nutné, musí se zkontrolovat skripty, jestli nenastal kvůli změně pořadí nějaký problém.

Z obrazovky Build Setting se lze dostat tlačítkem do Player Settings. Tady je potřeba nastavit jméno společnosti a produktu. Potom je důležité se přepnout do výběru nastavení správné platformy. Android zde má na záložce malou ikonku robota. V první sekci „Resolution and Presentation“ lze nastavit, jaké rozložení obrazovky bude povolené. Tady nemá smysl cokoli měnit, protože pozdější zapnutí módu virtuální reality toto nastavení upraví bez možnosti změn. V další sekci lze měnit ikonu aplikace. Sekce „Splash Image“ umožňuje nastavit úvodní logo a animaci. Následuje sekce „Other Settings“. Zde je nutné najít položku „Virtual Reality Supported“ a zaškrtnout ji. Následně se zobrazí výběrové menu, ve kterém je třeba vybrat Cardboard. Dále v položce „Package Name“ je třeba upravit název do tvaru `com.NázevOrganizace.NázevAplikace` bez mezer. Poslední položka, kterou je třeba upravit je „Minimum API Level“ ve které vybereme Android 4.4. „Kit Kat“ (API Level 19). To je minimální vyžadované API s podporou Cardboard.

6.7 Návod k použití

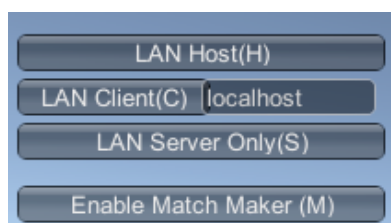
Vezmou se dvě mobilní zařízení s operačním systémem Android 4.4 a vyšší. Minimálně jedno ze zařízení musí obsahovat gyroskop. Telefon s gyroskopem bude ten, který se bude vkládat do Cardboard. Oba mobilní telefony musí být připojené ke stejné wifi síti. Nainstaluje a zapne se aplikace na obou zařízeních. Stiskneme tlačítko „Phone with Cardboard“ na tom telefonu, který přijde do Cardboard. Načte se lobby a v horní části se zobrazí IP adresa. Na druhém telefonu zadáme tuto adresu do políčka „Join a game“ a stiskneme „Join“. Oba telefony se nyní nachází v lobby a navzájem se vidí. Potvrdíme na obou telefonech tlačítkem „Join“, že jsou oba přístroje připravené a poté se na každém zařízení načte daná scéna. Na hostitelském telefonu, který přijde do Cardboard se zobrazí 3D scéna se zapnutou stereoskopickou kamerou. Nyní se tento telefon vloží do brýlí a nasadí na hlavu. Pro lepší zorientování se doporučuje, nejdřív si vyzkoušet ovládání bez nasazených brýlí, zjistit jak joystick a tlačítka reagují. Po nasazení Cardboard s telefonem na hlavu se zařízení pevně utáhne, aby nespadlo a zároveň netlačilo uživatele. Nyní stačí vzít do ruky ovládací telefon, pohybovat se scénou, skákat a střílet.

7 Testování

Při tvorbě této aplikace bylo použito uživatelské testování. To probíhalo s několika potencionálními uživateli. Testovala se uživatelská přívětivost a hledaly se případné chyby, které mohou nastat ať už ve standardním nebo nestandardním způsobu použití aplikace.

7.1 Uživatelská přívětivost

Testování uživatelské přívětivosti probíhalo tak, že osoba dostala dvě dva telefony se zapnutou aplikací. Nedostala však žádné instrukce nebo návod k použití. Pouze jí bylo sděleno k čemu aplikace slouží a co by měla dělat velice stručně. Osoby potom byly sledovány a při nastavování a používání popisovaly svůj zážitek.



Obr. 37 Network Manager HUD.

Všechny osoby si při prvním testování stěžovaly na složitý systém nastavování spojení. V původní verzi totiž ještě nebyl Lobby Manager a místo něj zde byl pouze Network Manager HUD. Jak je vidět z obrázku výše, jedná se spíše o rozhraní pro vývojáře na testování. Stížnosti byly vzaty v potaz a HUD byl nahrazen komplexnějším Lobby Managerem. Díky dalšímu testování a připomínkám bylo lobby upraveno tak, aby obsahovalo jen nutné prvky. Ty byly navíc zvětšeny pro lepší čitelnost. V původních verzích bylo nutné zjistit si IP adresu zařízení mimo aplikaci a poté ji zadávat. To označili testeři jako nepohodlné a tak byla obrazovka upravena tak, aby zobrazovala adresu přímo po vytvoření lobby. Po dalších konzultacích byla do lobby přidána informace o tom, že zařízení musí být ve stejné síti a celkově byly upraveny zobrazené texty, aby lépe navigovaly uživatele. Ovládání pomocí virtuálních joysticků bylo pro všechny uživatele intuitivní a pochopitelné hned na poprvé.

7.2 Buggy

Při vývoji této aplikace se vyskytlo několik problémů, které nastaly při testování uživateli. Ty zdržely vývoj a bylo nutné je nějak vyřešit. Díky přítomnosti u uživatelských testů, bylo snadné vysledovat, jaké kroky chybu spustily a reprodukovat je. V následujícím textu jsou vyjmenovány nejdůležitější problémy, co je způsobilo, jak byly vyřešeny.

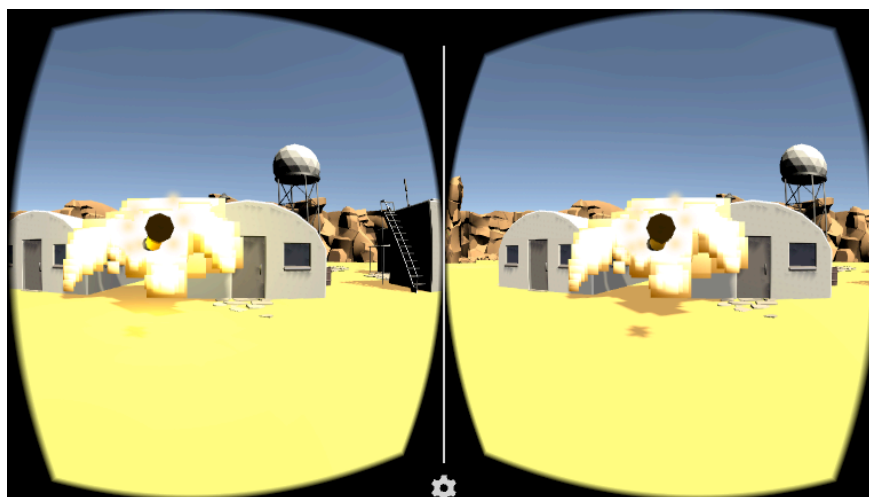
7.2.1 Rozlišení hosta a klienta

Pokud se uživatel připojil do hry hned napoprvé, tak vše fungovalo v pořádku. Problém nastal ve chvíli, kdy uživatel vytvořil lobby, pak z něj odešel a znovu ho vytvořil. Později bylo zjištěno, že tato chyba je způsobena špatným způsobem rozlišení hosta od klienta. Každý síťový objekt dostane po vytvoření svůj vlastní unikátní identifikátor nazývaný netID. V ideálním případě dostane první netID lobby prefab hosta, druhé lobby prefab klienta. Třetí netID dostane player prefab hosta a čtvrté player prefab klienta. Na player prefab tak byl umístěn skript, který pro netID s číslem tři načte scénu 3DScene a objekt s netID čtyři načte scénu Controller a poslal na server příkaz na vytvoření prefabu Ghost. Pokud však uživatel vytvoří lobby, potom z něj odejde a vrátí se, tak je první lobby prefab zničen a s ním i jeho netID a je vytvořen nový. Tento problém posune číslování, player prefab nikdy nedostane číslo tři a tak není přesměrován do správné scény.

Řešení tohoto problému spočívá ve speciální síťové funkci. Funkce se nazývá isServer, případně isClient, a vrací true v případě, že objekt, na který je skript připnutý patří serveru a false pokud ne. Problém s funkcí isClient je ten, že host je zároveň i klientem. Tudíž by na obou zařízeních vracel true. S využitím tohoto postupu se lze nyní v lobby libovolně připojovat a odpojovat a scény jsou načteny korektně.

7.2.2 Přidání interakce s okolím

V původní verzi se dalo volně pohybovat všemi směry. V pozdější verzi byla použita jiná scéna (ta aktuální) a uživatelé při testování sdělili, že by rádi měly ve hře další funkce. Místo pohybu nahoru a dolů by ocenily spíše skákání a nějaké akční tlačítko, například střelbu.



Obr. 38 Střílení s výbuchem projektilu.

Střílení bylo převzato z tutoriálu Tanks tutorial, skripty byly upraveny tak, aby fungovaly v této aplikaci. Problém však byl se skákáním. Aby mohl hráč skočit

a dopadnout zpět do původní hladiny, musí být nějakým způsobem vyřešena gravitace. Ta byla původně zapnuta pomocí prvku RigidBody, ale ukázalo se to jako slepá cesta. Hráč se hned po vytvoření začal propadat podlahou a stále padal. Řešení se nakonec povedlo najít za pomoci Character Controlleru. Ten přináší snadné řešení pro pohyb a skákání.

7.2.3 Procházení objekty

Zároveň s předchozím problémem byl řešen další návrh od uživatelů a to procházení objekty. Ve většině her jsou ve scéně objekty, kterými procházet lze (křoví, voda, kouř atd.) a ty kterými procházet nelze (domy, stromy, zdi atd.). Ve starší verze aplikace se však dalo procházet všemi objekty a to nepůsobilo přirozeně. Řešení nakonec bylo nalezeno v Character Controlleru, který zároveň vyřešil i předchozí problém. Ten totiž obsahuje i kolizní kapsli hráče a díky ní se hráč o zed' opatřenou kolizním objektem zarazí a neprojde dál.

8 Diskuze

Na konci této práce je hotová funkční aplikace pro mobilní operační systém Android s podporou Google Cardboard. Aplikace slouží k prezentaci možného řešení pohybu spíše vývojářům než veřejnosti. V tomto ohledu je zadání splněno, avšak vždy je prostor k dalšímu vylepšování. Způsobů jak vyřešit daný problém je více.

Jedním z možných rozšíření aplikace by mohla být sada různých uživatelských prvků. Současný návrh počítá s řešením pohybu a rotace na jedné půlce displeje a dvěma akčními tlačítky na druhé půlce displeje. To se může hodit pro určitý typ her střileček z prvního pohledu. Pro některé logické hry by se spíš hodilo více tlačítek bez možnosti pohybového joysticku. Jinde by se zase hodila původní verze aplikace, která umožňovala volný pohyb scénou bez zapnuté gravitace a s procházením objekty. Do Unity lze importovat 3D grafiku z různých grafických programů jako například Cinema 4D. Tohle řešení by tak šlo upravit na různé prezentace, kde by osoby v místnosti (jedna nebo více) měla na hlavě Google Cardboard a prezentující by jim pohybovat s postavou a popisoval scénu kterou vidí. Tímto způsobem by šly nízkonákladově prezentovat domy, auta a další produkty. Při tomto způsobu prezentace se ušetří za náklady při cestování a výhodou je, že tak lze prezentovat ještě nevyrobené produkty zajímavějším způsobem než pouhými fotografiemi nebo videem. Způsobů využití je spousta a vždy by bylo třeba ovládnutí upravit konkrétnímu použití.

Momentálně jsou v aplikaci zvukové efekty pouze u výstřelu a výbuchu. Dal by se dodělat kompletní systém zvuků chůze, zpěvu ptáků a hudby na pozadí. Takové věci dokáží zážitek ze hry zase o něco zpříjemnit.

Aplikace by se dala změnit dodáním nějakých úkolů nebo cílů ke zničení. Hráč by sbíral nějakou formu bodů. Ve hře jednoho hráče by mohli být nepřátelé ovládané umělou inteligencí nebo by se mohla hra předělat a uzpůsobit pro více hráčů. Bylo by pak nutné rozlišit od sebe zařízení pro zobrazení scény a ovládací. Také by bylo nutné vytvořit systém spárování dvojic zařízení, aby bylo vždy jasné, který mobilní přístroj ovládá daný telefon s Cardboard.

Dalším zajímavým nápadem je využití multi platformního Unity a vydat aplikaci nejen na Android, ale i na ostatní systémy. Samozřejmě ty mobilní, ale využití by se našlo i u desktopových systémů. Pokud by nebylo k dispozici druhé mobilní zařízení, mohl by si pohyb hráč ovládat přes klávesnici počítače. Řešení je už zahrnuto v práci, protože jsou ovládací prvky namapovány na standardní vstupní klávesy. Tlačítko pro skok tak lze na stolním počítači nebo notebooku nahradit stisknutím mezerníku, klávesy WASD slouží k pohybu a podobně na tom jsou i další ovládací prvky.

Kombinovat lze klasické zobrazení na monitoru, televizi nebo obyčejný display telefonu nebo tabletu, se stereoskopického zobrazením s Cardboard. Jednoduše by se přidalo okno s nastavením, kde by se vybíralo jestli bude mód virtuální reality aktivovaný či nikoliv. Možností využití a úprav je spousta a záleží jen na představivosti vývojářů a požadavcích klientů.

9 Závěr

Hlavní myšlenkou této práce je navrhnout a vytvořit řešení pro pohyb ve virtuální realitě. Vytvořená aplikace může sloužit jako inspirace pro vývojáře mobilních her pro Google Cardboard. Principy ovládání pohybu ve scéně lze z této práce aplikovat do libovolné hry. Aplikace sama o sobě není pro běžné uživatele moc zajímavá. Nabízí sice možnost procházení virtuální scény, skákání a střelení, ovšem uživatel je ve hře sám, není do koho střílet a tak hraní po chvíli omrzí. To je však problém většiny her pro Cardboard. Hry pro něj vytvořené nejsou určeny pro několika hodinovou zábavu a pravidelné hraní. Spíše slouží na ukázkou toho, jak virtuální realita vypadá. Cardboard může sloužit jako pomyslný první krok, který ukáže virtuální realitu za dostupnou cenu. Pro ty, které VR nadchne, jsou na trhu systémy lépe propracované, ovšem mnohem dražší řešení. Hlavní přínos Cardboard tak je ve zpřístupnění VR veřejnosti. Tato práce přináší návrh řešení, jak zůstat finančně přístupný a přitom dát uživateli o něco lepší zážitek díky možnosti pohybu ve scéně, případně různé interakce.

10 Literatura

- Android (operating system). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-05-20]. Dostupné z: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- Android Studio. *Google Developer* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://developer.android.com/studio/index.html>
- App Store: Support. *Apple Developer* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://developer.apple.com/support/app-store/>
- Cross-Platform Game/App Development Toolset. *Corona Labs* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://coronalabs.com/>
- Dashboards. *Android Developers* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://developer.android.com/about/dashboards/index.html>
- Daydream: Introducing Daydream. *Google Developer* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://vr.google.com/daydream/>
- DIY Google Cardboard Virtual Reality 3D Glasses. *EBay.com* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <http://www.ebay.com/itm/DIY-Google-Cardboard-Virtual-Reality-3D-Glasses-for-iPhone-Samsung-Phones-WSx-322483428181?hash=item4b15829b55:g:JT4AAOSwM4xXYD84>
- Download & Install Windows 10 Mobile Preview 15215 for Phones & Tablets. *TechGlobex.net* [online]. 2015 [cit. 2017-05-20]. Dostupné z: <http://www.techglobex.net/2015/02/windows-10-technical-preview-for-phones.html>
- Firefox OS. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-05-20]. Dostupné z: https://cs.wikipedia.org/wiki/Firefox_OS
- Gartner Says Worldwide Sales of Smartphones Grew 7 Percent in the Fourth Quarter of 2016. *Gartner.com* [online]. Egham, U.K., 2017 [cit. 2017-05-20]. Dostupné z: <http://www.gartner.com/newsroom/id/3609817>
- GIBSON, Jeremy. Introduction to game design, prototyping, and development: from concept to playable game-with Unity® and C#. ISBN 03-219-3316-8.
- Google Cardboard. *Google VR* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://vr.google.com/cardboard/>
- Google Play* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://play.google.com/store>
- Google VR overview. *Google Developers* [online]. 2016 [cit. 2017-05-20]. Dostupné z: <https://developers.google.com/cardboard/overview>
- LACKO, Ľuboslav. Vývoj aplikací pro Android. Brno: Computer Press, 2015. ISBN 978-80-251-4347-6.

- MÁRA, Petr. V čem se liší filozofický přístup iOS a Androidu? *Youtube.com* [online]. 2017 [cit. 2017-05-21]. Dostupné z: <https://www.youtube.com/watch?v=q8WpMDsGLUU>
- Mobile Marketing Statistics compilation. *SmartInsights.com* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>
- Mobile operating system. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-05-20]. Dostupné z: https://en.wikipedia.org/wiki/Mobile_operating_system
- PULTZNER, Martin. Pohled do smyšleného světa: Tohle jsou 3 nejlepší virtuální reality. *Forbes.cz* [online]. 2017 [cit. 2017-05-21]. Dostupné z: <http://www.forbes.cz/tohle-jsou-3-nejlepsi-vr/>
- State of IAPs: iOS users spend 2.5x more on in-app purchases than Android users. *AndroidAuthority.com* [online]. 2016 [cit. 2017-05-20]. Dostupné z: <http://www.androidauthority.com/new-report-reveals-that-ios-users-spend-2-5-more-on-in-app-purchases-than-android-users-700983/>
- Swift: About. *Swift.org* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://swift.org/about/>
- Tips for using Google Cardboard. *Unofficial Cardboard* [online]. 2017 [cit. 2017-05-21]. Dostupné z: <https://www.unofficialcardboard.com/pages/tips>
- Top 10 Mobile Phones Operating Systems. *ShoutMeLoud.com* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://www.shoutmeloud.com/top-mobile-os-overview.html>
- Tutorials: Tanks tutorial. *Unity* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://unity3d.com/learn/tutorials/projects/tanks-tutorial>
- UJBÁNYAI, Miroslav. Programujeme pro Android. Praha: Grada, 2012. Průvodce (Grada). ISBN 978-80-247-3995-3.
- Unity Manual. *Unity* [online]. Unity Technologies, 2015 [cit. 2017-05-20]. Dostupné z: <https://docs.unity3d.com/Manual/index.html>
- Unity: Game Engine. *Unity Technologies* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://unity3d.com/>
- Virtual Reality. *Virtual Reality Society* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://www.vrs.org.uk/>
- What is Cocos2d-x? *Cocos2d-x: World's #1 Open-Source Game Development Platform* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <http://www.cocos2d-x.org/>
- What is Unreal Engine 4. *Unreal Engine* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://www.unrealengine.com/what-is-unreal-engine-4>
- What's a good game engine? *Factions by Pidbaq* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <http://factions.pidbaq.com/whats-a-good-game-engine/>

-
- Windows Phone. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-05-20]. Dostupné z: https://cs.wikipedia.org/wiki/Windows_Phone
- XCode. *Apple Developer* [online]. 2017 [cit. 2017-05-20]. Dostupné z: <https://developer.apple.com/xcode/>

11 Seznam obrázků

Obr. 1	Počet uživatelů mobilních zařízení a stolních počítačů. Zdroj: SmartInsights.com (2017)	14
Obr. 2	Grafy prodeje mobilních telefonů podle operačního systému za 4Q15 a 4Q16.	16
Obr. 3	Uživatelské prostředí Windows 10. Zdroj: techglobex.net (2015)	17
Obr. 4	Uživatelské prostředí iOS 10.	18
Obr. 5	Fragmentace verzí systému iOS. Zdroj: Apple.com (2017)	19
Obr. 6	Fragmentace verzí systému Android. Zdroj: Android.com (2017)	20
Obr. 7	Uživatelské prostředí Android 7.1 Nougat. Zdroj: Calvin Hogg (2017)	21
Obr. 8	Funkce IDE. Zdroj: Murtaxa, 2016	22
Obr. 9	Z jakých částí se skládá game engine. Zdroj: Factions by Pidbaq, 2016	24
Obr. 10	Podíl enginů nejúspěšnějších 1000 her zdarma v prvním kvartále roku 2016. Zdroj: Unity3D, 2016	25
Obr. 11	Podporované platformy Unity. Zdroj: Unity, 2017	29
Obr. 12	Podíl skriptů vytvořených ve třech dostupných jazycích v Unity. Zdroj: Unity, 2014	31
Obr. 13	Cenové plány Unity. Zdroj: Unity, 2017	32
Obr. 14	Sensorama. Zdroj: Virtual Reality Society, 2016	34
Obr. 15	Zařízení „Damoklův meč“. Zdroj: Virtual Reality Society, 2016	35
Obr. 16	Dataglove a EyePhone Zdroj: Virtual Reality Society, 2016	35
Obr. 17	Arkádové hry s podporou VR. Zdroj: Virtual Reality Society, 2016	36

Obr. 18	Nintendo Virtual Boy. Zdroj: Virtual Reality Society, 2016	36
Obr. 19	Google Cardboard. Zdroj: Google, 2017	37
Obr. 20	Google Daydream. Zdroj: Google, 2017	38
Obr. 21	HTC Vive. Zdroj: Forbes.cz, 2017	39
Obr. 22	Oculus Rift. Zdroj: Forbes.cz, 2017	40
Obr. 23	Sony PlayStation VR. Zdroj: Forbes.cz, 2017	41
Obr. 24	Vývojové prostředí Unity.	46
Obr. 25	Network Lobby manager.	47
Obr. 26	Skript Move a jeho nastavení.	49
Obr. 27	Network Transform skript.	49
Obr. 28	Nastavení Character Controller.	50
Obr. 29	Tank Shooting skript	50
Obr. 30	LeverArt scéna z tutoriálu o tvorbě hry s tanky.	51
Obr. 31	Scéna Main.	52
Obr. 32	Lobby.	52
Obr. 33	Připojení obou mobilních zařízení.	53
Obr. 34	3D scéna.	54
Obr. 35	Scéna s ovladačem.	55
Obr. 36	Build Settings.	56
Obr. 37	Network Manager HUD.	58
Obr. 38	Střílení s výbuchem projektilu.	59