



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**ZÁBAVNÉ MINIHRY PRO VÍCE HRÁČŮ VE VIRTUÁLNÍ
REALITĚ**

MULTIPLAYER MINIGAMES IN VIRTUAL REALITY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DUŠAN ČUBÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. DANIEL BAMBUŠEK

BRNO 2022

Zadání bakalářské práce



Student: **Čubík Dušan**
Program: Informační technologie
Název: **Zábavné minihry pro více hráčů ve virtuální realitě**
Multiplayer Minigames in Virtual Reality
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte koncept virtuální reality a její využití pro hry více hráčů. Seznamte se s aktuálními síťovými přístupy a nástroji pro hry více hráčů.
2. Vyberte vhodné metody a nástroje a navrhnete sérii zábavných miniher pro více hráčů ve virtuální realitě.
3. Navržený balíček miniher implementujte.
4. Proveďte uživatelské experimenty a vyhodnoťte vlastnosti výsledného řešení.
5. Vytvořte video prezentující klíčové vlastnosti výsledného řešení.

Literatura:

- JERALD Jason. *The VR book: human-centered design for virtual reality*. San Rafael, CA: Morgan & Claypool, 2016. ISBN 978-1-97000-115-0.
- LINOWES Jonathan. *Unity Virtual Reality Projects: Learn Virtual Reality by developing more than 10 engaging projects with Unity 2018*, 2nd Edition, Packt Publishing, 2018.
- HARTSON Rex. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. 2012. ISBN 9780123852427.
- Dále dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a rozpracovaný bod 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Bambušek Daniel, Ing.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 1. listopadu 2021

Abstrakt

Tato práce se zabývá návrhem a implementací hry pro virtuální realitu pro více hráčů. Konkrétně se jedná o sérii zábavných miniher, ve kterých mezi sebou hráči soutěží. Výsledkem práce je klientská aplikace s dvěma implementovanými minihrami pro zařízení Oculus Quest a Quest 2 a serverová aplikace. Shooting Range je minihra pro soutěž ve střelbě. Towers je minihra, ve které si hráči vzájemně ničí své pevnosti. Minihry otestovali uživatelé. Pro implementaci klientské aplikace byl použit herní engine Unity. K implementaci virtuální reality slouží knihovna XR Interaction Toolkit. Síťovou komunikaci na straně klienta realizuje Photon PUN. Implementace serverové části zajišťuje Photon Server SDK.

Abstract

This thesis deals with the design and implementation of a multiplayer virtual reality game. Specifically, it is a series of fun mini-games in which players compete against each other. The result of the work is a client application with two implemented minigames for Oculus Quest and Quest 2 devices and a server application. Shooting Range is a minigame for shooting competition. Towers is a minigame in which players destroy each other's forts. The minigames have been tested by users. The Unity game engine was used to implement the client application. The XR Interaction Toolkit library was used to implement the virtual reality. The network communication on the client side is implemented by Photon PUN. The server side implementation is provided by Photon Server SDK.

Klíčová slova

virtuální realita, hra pro více hráčů, herní design

Keywords

virtual reality, multiplayer game, game design

Citace

ČUBÍK, Dušan. *Zábavné minihry pro více hráčů ve virtuální realitě*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Daniel Bambušek

Zábavné minihry pro více hráčů ve virtuální realitě

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana inženýra Daniela Bambuška. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Dušan Čubík
11. května 2022

Poděkování

Děkuji panu inženýrovi Danielovi Bambuškovvi za odborné vedení práce a za cenné rady při implementaci a psaní této práce. Rovněž děkuji Fakultě Informačních Technologií za poskytnutí zařízení Oculus Quest.

Obsah

1	Úvod	3
2	Virtuální realita	4
2.1	Virtuální realita obecně	4
2.2	Historie až dodnes	4
2.3	Zařízení	5
2.4	Rozdíl mezi virtuální realitou a rozšířenou realitou	6
2.5	Využití virtuální reality	7
3	Hry ve virtuální realitě	11
3.1	Populární a úspěšné tituly na VR	11
3.2	Nástroje pro vývoj	12
3.3	Rizika při vývoji	13
3.4	Platformy pro publikování	13
4	Návrh aplikace	15
4.1	Základní popis	15
4.2	Návrh struktury aplikace	15
4.3	Návrh miniher	16
4.4	Uživatelské rozhraní	23
4.5	Základní ovládání	23
4.6	Volba síťové infrastruktury	24
5	Implementace	26
5.1	Oculus Quest/Quest 2	26
5.2	Unity3D	27
5.3	Photon PUN	30
5.4	Network Player	31
5.5	LoginSceneBP	33
5.6	HomeSceneBP	35
5.7	Čekací místnost	36
5.8	Minihry Shooting Range a Towers	38
5.9	Implementace Serverové části	46
6	Testování	50
6.1	Demografický dotazník	50
6.2	Průběh testování	51
6.3	Dotazník měřící uživatelskou zkušenost (UEQ)	51

6.4	Vlastní dotazník a poznámky	53
6.5	Technické chyby	56
7	Závěr	57
	Literatura	58

Kapitola 1

Úvod

Za poslední dekádu virtuální realita zažila obrovský vzestup. Jedním z důvodů proč se virtuální realita stává více oblíbenou jsou hry. Hardware pro zprostředkování virtuální reality se stává dostupnější pro běžné lidi a jejich motivace si takový hardware koupit je zábava. Jinými slovy už to nejsou jen univerzity nebo firmy, které využijí virtuální realitu pro profesionální úkony.

Hry ve virtuální realitě ještě do nedávna tvořily zejména menší studia a firmy tvořící velkorozpočtové tituly se tomuto sektoru vyhýbaly. Zřejmě kvůli velikosti a rizikům trhu. Dnes už známé herní značky vycházejí na platformy virtuální reality, jako například Half-Life: Alyx, The Elder Scrolls V: Skyrim VR, Medal of Honor: Above and Beyond. Ať už jen jako port z PC na VR nebo titul tvořený výhradně pro virtuální realitu.

Chtěl jsem se zaměřit na práci, jejíž výstup využijí i běžní uživatelé (tj. ne jen odborníci v oboru). Já sám se věnuji vývoji her. Ale s vývojem hry pro více hráčů jsem se nesetkal a tvorba pro virtuální realitu je také pro mě nová. S ohledem na stoupající trend virtuální reality a metaverse¹ byla kombinace těchto dvou aspektů jasný signál se těmto tématům začít věnovat.

Cílem této práce je navrhnout a implementovat hru pro virtuální realitu pro více hráčů pro zařízení Oculus Quest/Quest2. Hra bude obsahovat sérii zábavných miniher, ve kterých hráči mezi sebou budou soutěžit. Hra je cílena pro příležitostné hráče, tudíž jednotlivé minihry musí být navrženy tak, aby jejich princip každý intuitivně a rychle pochopil. Implementovaná hra bude základní verze, jenž bude možné v budoucnu rozšířit a potencionálně z ní vytvořit finální produkt pro trh.

Druhá kapitola uvádí čtenáře do problematiky virtuální reality a jejímu přínosu. V třetí kapitole zkoumá hry ve virtuální realitě a jejich rozdíly při vývoji oproti hrám na jiných platformách a možnosti distribuce aplikací ve virtuální realitě. Čtvrtá kapitola představuje základní myšlenku hry. Je rozvrženo několik možných miniher – jejich prvky, prostředí, smysl, ale i rizika, která mohou nastat při implementaci. Následně jsou rozebrány možnosti základního ovládání, interakce s prostředím a uživatelským rozhraním. Nakonec i byly otestovány a zhodnoceny různá řešení pro síťovou komunikaci. Pátá kapitola popisuje výslednou aplikaci a způsob její implementace. Poslední kapitola hodnotí výslednou aplikaci pomocí testování.

¹Metaverse – trvalé a perzistentní prostředí slučující fyzickou realitu s digitální realitou. Účastníci jsou ztělesněni ve virtuálním světě, komunikují mezi sebou v reálném čase a interagují s digitálními artefakty [12].

Kapitola 2

Virtuální realita

Tato kapitola uvádí do základní problematiky virtuální reality.

2.1 Virtuální realita obecně

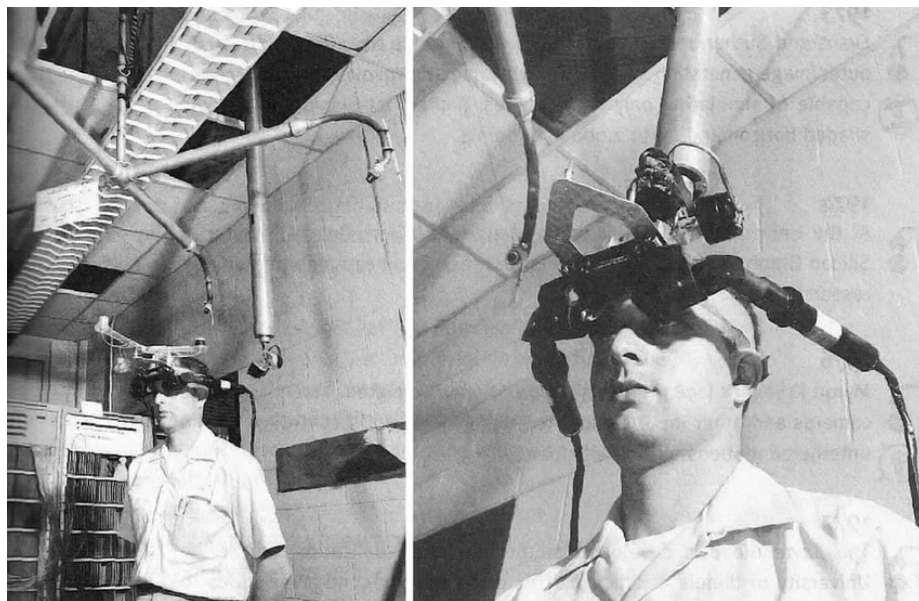
Virtuální realita je digitální prostředí vytvořené počítačem, kdy účastníci mohou svými činy částečně ovlivnit chování této reality. Prostor se jeví jako reálné, tj. účastníci mohou se v něm pohybovat a dotýkat předmětů tak jako v reálném světě [9]. Ideální virtuální realita, kterou není možné rozeznat od naší reality, zatím neexistuje. Například není možné dělat kotrmelce. Vnímání virtuální reality je omezeno podle výstupů zařízení.

2.2 Historie až dodnes

Jako jeden z prvních předchůdců dnešních zařízení pro virtuální realitu se dá považovat stereoskop z roku 1832 od Sir Charles Wheatstone. Divák pozoroval fotografie či malby z různých úhlů a objekt působil trojrozměrně. Poté se začali vyrábět různé varianty těchto zařízení pro běžné spotřebitele. V 1862 bylo prodáno milion kusů zařízení, které fungovalo na podobném principu jako Google Cardboard.

V roce 1961 zaměstnanci Philco Corporation vyvinuli první zařízení, které snímalo pohyb hlavy. Když uživatel pohnul hlavou, tak kamera, ze které se přenášel obraz do zařízení, se otočila ve stejném směru. O rok později IBM získalo patent pro první zařízení, které umožňovalo vstup – rukavice. Uživatel měl k dispozici dvě rukavice a každý prst byl snímán čtyřmi senzory, což umožňovalo 1 048 575 vstupních kombinací. V roce 1965 Ivan Sutherland jako první demonstroval zařízení „na hlavě“ zobrazující obraz generovaný počítačem. Mezitím americká armáda vyvíjela podobné zařízení pro trénink pilotů.

Tým, vedený Alan Kay, v Atari Research v roce 1981 hledalo nové možnosti pro zábavní průmysl a navrhli technologie, které byly klíčové pro komerční využití zařízení virtuální reality. O čtyři roky později tým opustili Jaron Lanier a Thomas Zimmerman a založili VPL Research pro vývoj komerčních VR zařízení. I když tato společnost na začátku 90. let zkrachovala, následující dekádu průmysl virtuální reality zachvátil „boom“. Komerční společnosti, univerzity a armádní složky začali experimentovat s virtuální realitou a dostávala se mezi veřejnou společnost. Bohužel v roce 1996 průmysl virtuální reality dosáhl svého vrcholu a její popularita začala upadat. Až do roku 2012 o ní sotva padaly zmínky, kdy byla založena společnost Oculus Rift a zahájila Kickstarter kampaň pro získání prostředků na vývoj zařízení pro virtuální realitu [10].



Obrázek 2.1: Zařízení Ivana Sutherlanda pro virtuální realitu [?]

2.3 Zařízení

Dnes se virtuální realita nejčastěji zobrazuje pomocí headsetů a jako vstup se využívají ovladače do ruky, jejichž možnosti vstupu se můžou lišit.

- PCVR/ConsoleVR – výpočty probíhají na počítači (na herní konzoli) a obraz se promítá do headsetu (např. kabelem)
 - Výhody: kvalitní zobrazení obrazu (záleží na grafické kartě).
 - Nevýhody: drahé – je potřeba mít počítač se vysoce výkonnou grafickou kartou (herní konzoli) a zároveň headset, kabel je zapojený pro přenos obrazu a to není pohodlné.
 - Příklad: Valve Index (viz obrázek 2.2), PlaystationVR



Obrázek 2.2: Valve Index

Zdroj: <https://www.valvesoftware.com/>

- Standalone – samostatné zařízení, které je v podstatě mobilní telefon s jinými vstupy
 - Výhody: není potřeba mít např. kabel, cenově dostupnější než PCVR
 - Nevýhody: méně kvalitní zobrazení obrazu, délka hratelnosti(používání) je omezena výdrží baterky
 - Příklad: Oculus Quest 2
- Chytrý telefon – samotný headset není většinou elektronické zařízení. Je to „krabice“ a do ní se vloží mobilní telefon, který slouží k zobrazení obrazu. Slouží spíše k přehrávání 360° videí než k používání interaktivních aplikací.
 - Výhody: velice levné
 - Nevýhody: většinou velmi omezené uživatelské vstupy(často bez ovladačů), malé množství kvalitních aplikací, mobilní telefony nemusí být navrženy pro tento účel, tudíž vizuální kvalita může být špatná i baterie se může vybíjet rychleji
 - Příklad: Samsung VR

2.4 Rozdíl mezi virtuální realitou a rozšířenou realitou

Virtuální realita a rozšířená realita se často zaměňují. Sdílejí některé klíčové technologie, často se však jedná o rozdílné zážitky. Ve virtuální realitě se uživatel ocitne v počítačem vygenerovaném prostoru izolovaném od reálného světa. Zatímco rozšířená realita obohacuje reálný svět o vizuální (virtuální) prvky.

Z hlediska počítačové grafiky, rozšířená realita a virtuální realita sdílí nemálo aspektů. Avšak u rozšířené reality uživatel vidí jak realitu tak i vykreslené překrytí na transparentním displeji. Hardware i softwarové nástroje jsou si také podobné. Ale rozšířená realita vyžaduje práci optikou a detekcí.

Rozšířená realita překrývá zorné pole uživatele počítačově vygenerovanými objekty a daty. Virtuální realita vytváří umělé prostředí a úplně skrývá svět kolem. V průmyslu se obě technologie využívají jako trenažéry, ke vzdělání nebo pro zábavu. Rozšířená realita

zobrazuje například instrukce a může sloužit jako asistent při plnění úkolu. U virtuální reality je možnost nacvičit situace, které mohou být nebezpečné.

Virtuální realitu lze zobrazit pouze pomocí headsetu, zatímco rozšířená realita má více možností – chytrý telefon, helma, tablet (viz obrázek 2.3), počítač nebo brýle.

Požadavek u rozšířené reality je detekovat i reálné objekty. Například stůl musí být detekován, aby se renderovaný objekt správně vykreslil (na stole). Důležitá je informace o pozici pro zachování perspektivy. Když se uživatel (kamera) bude pohybovat, tak renderovaný objekt musí například změnit velikost nebo vykreslit se pod jiným úhlem.

Klíčové je správné splnutí virtuální a reálné scény. Pokud se uživatel soustředí na virtuální objekt, a poté se podívá na objekt v místnosti, může být narušena iluze a uživateli to nebude příjemné. Na rozdíl od virtuální reality uživatelé vidí své ruce, nohy a jiné části těla. To může být vysvětlení, proč uživatelé netrpí nevolností při používání rozšířené reality [14].



Obrázek 2.3: Příklad využití rozšířené reality – medicína.

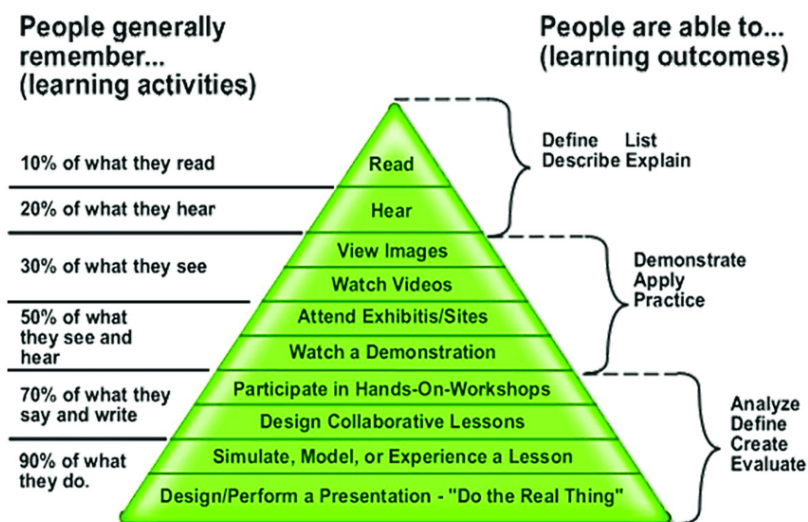
Zdroj: <https://jasoren.com/capabilities-and-limitations-of-apple-arkit-and-google-arcore/>

2.5 Využití virtuální reality

Virtuální realita nejvíce vzrůstá v zábavním průmyslu. Ovšem potenciál virtuální reality je mnohem vyšší. Již několik let virtuální realita zasahuje do jiných odvětví průmyslu. Mezi úspěšné využití virtuální reality patří vědecké vizualizace, architektura, letecké simulátory, terapie, vzdělání, vojenský výcvik a další тренаžéry pro určité profese. Využití virtuální reality snižuje náklady tím, že redukuje chybovost návrhů před výrobou produktu. Poskytuje bezpečné prostředí pro nácvik nebezpečných situací a snižuje PTSD. Pomáhá lépe vizuálně zobrazit data [9].

Americký vzdělavatel Edgar Dale vyvinul „Cone of Experience (viz obrázek 2.4)“. Má formu pyramidy a popisuje způsoby zapamatuje učení nových informací a jejich úroveň abstrakce. Základ pyramidy tvoří reálné zkušenosti a směrem nahoru se zvyšuje abstrakce dané informace pro studenta. Vrchol pyramidy obsahuje nejvíce abstraktní vnímání nové

informace – reprezentace pomocí znaků (čtení) anebo poslouchání. Střed pyramidy tvoří způsoby, kdy student novou informaci pozoruje. Například vidí ji na obrázku, ve videu či je mu přímo demonstrována. Spodní část se soustředí na činnosti nebo aktivní práci s danou informací, tj. kdy si student danou věc opravdu vykouší. Je tedy tvrzeno, že člověk se nejlépe naučí novým věcem, když si je přímo zkouší [1]. A právě virtuální realita umožňuje různé situace skutečně prožít a může nabízet obrovský potenciál ve vzdělání.



Obrázek 2.4: „Cone of Experience“

Zdroj: https://www.researchgate.net/figure/Edgar-Dales-Cone-of-Experience-shows-that-after-two-weeks-we-mainly-remember-information_fig1_336935595/actions#reference

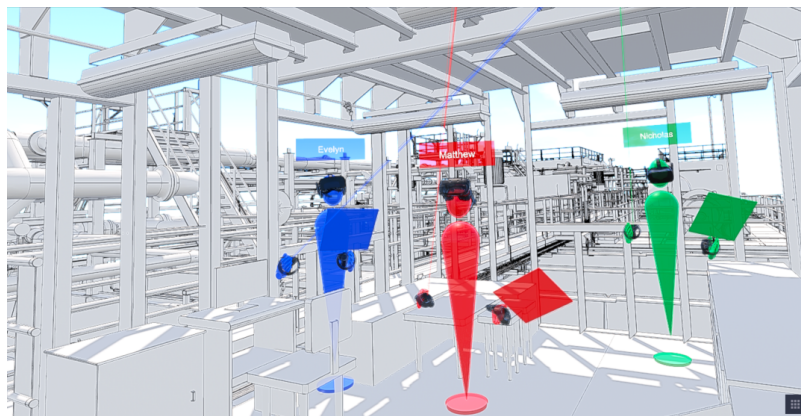
Jak už bylo zmíněno, tak virtuální realita už objevila své uplatnění v průmyslu. Zde jsou konkrétní příklady využití virtuální reality:

- IrisVR¹ – nástroj pro stavební průmysl. Umožňuje vizualizaci objektů, odhalení chyb v návrhu a kolaboraci s více uživateli, kterou lze vidět na obrázku 2.5. Společnost Autodesk, Inc.² nedávno odkoupila tento produkt³.

¹IrisVR, dostupné z <https://irisvr.com/>

²Autodesk, Inc. – softwarová firma zaměřující se na 3D grafiku, vizuální postprodukci, dostupné z <https://www.autodesk.com/>

³Akvizice IrisVR společností Autodesk, Inc., dostupné z <https://blog.irisvr.com/irisvr-joins-autodesk>



Obrázek 2.5: Kolegové si spolu prohlíží návrh v IrisVR.

Zdroj: <https://blog.irisvr.com/human-centered-design-how-to-empower-stakeholders-with-vr-pre-occupancy-evaluations>

- Apex Officer – policejní тренаžér využíván oficiálními policejními složkami v USA ⁴.
- OssoVR – тренаžér pro provádění operací⁵.
- ClassVR – tato služba poskytuje hardware do škol. Součástí je také software pro učitele pro tvorbu a řízení lekcí se studenty⁶ (schéma řízení lekce na obrázku 2.6).

⁴Apex Officer, dostupné z <https://www.apexofficer.com/resources/police-departments-are-implementing-vr-training-technologies>

⁵OssoVR, dostupné z <https://www.ossovr.com/>

⁶ClassVR pro školy, dostupné z <https://www.classvr.com/classroom-virtual-reality-lessons/>



Obrázek 2.6: Schéma režie hodiny učitelem při použití ClassVR.

Zdroj: <https://www.classvr.com/classroom-virtual-reality-lessons/teacher-headset-control-management/>

Kapitola 3

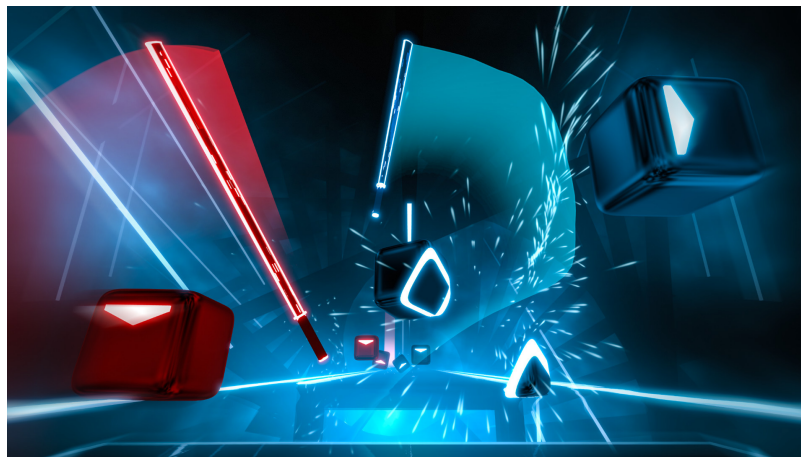
Hry ve virtuální realitě

Podle *Fortune Business Insights* v roce 2020 globální trh her ve virtuální realitě přesáhl hodnotu 6,26 miliard amerických dolarů. Což je o 25,7% více než v předchozích letech. Je nutné brát v potaz COVID-19, kdy v důsledku protiepidemických opatření trávili více volného času doma [8].

3.1 Populární a úspěšné tituly na VR

Beat Saber

Jedna z nejuspěšnějších her pro virtuální realitu. Hráč má v ruce dva meče a v rytmu hudby seká kostky (viz obrázek 3.1). Hra byla vyvinuta pražskou firmou a v roce 2019 byla odkoupena společností Facebook, Inc. (dnes Meta, Inc.).



Obrázek 3.1: Úkázka ze hry Beat Saber.

Zdroj: https://store.steampowered.com/app/620980/Beat_Saber/

Pavlov VR

Střílečka z první osoby připomínající sérii Counter-Strike. Hráči po sobě střílí na menších nebo středně velkých mapách (viz. obrázek 3.2). Jeden z příkladů, kde hráč realisticky manipuluje se zbraněmi.



Obrázek 3.2: Ukázka ze hry Pavlov VR.

Zdroj: https://store.steampowered.com/app/555160/Pavlov_VR/

Half-Life: Alyx

Jedna z prvních vysoko rozpočtových her hra pro virtuální realitu od společnosti Valve Corporation¹. Zasažena do prostředí herní série Half-Life (viz obrázek 3.3). Oceňována pro umělecké zpracování, herní design a intuitivní ovládání pro hru takového měřítka. Po úspěchu Half-Life: Alyx začali i ostatní velké společnosti experimentovat s hrami pro virtuální realitu.



Obrázek 3.3: Half-Life: Alyx

Zdroj: https://store.steampowered.com/app/546560/HalfLife_Alyx/

3.2 Nástroje pro vývoj

Unity

Multiplatformní herní engine. Velice populární mezi nezávislými vyvojáři, ale i většími společnostmi. Podporuje tvorbu 2D her, 3D her či VR/AR. Původně určen pro mobilní hry. Tento engine je podrobněji popsán v sekci 5.2.

¹Valve Corporation, dostupné z <https://www.valvesoftware.com/>

Unreal Engine

Zkušenější vývojáři, kteří vyvíjí hry velkých rozměrů volí většinou Unreal Engine ². Oproti Unity je více modifikovatelný – zdrojový kód je přístupný, takže se nabízí pokročilejší optimalizace hry. Ovšem vývoj menších her pomocí Unreal Engine může být na druhou stranu „kanón na vrabce“.

Amazon Sumerian

Amazon Sumerian³ slouží k vytváření interaktivních 3D scén pro virtuální realitu (nebo rozšířenou realitu). Software můžou používat i lidé bez znalosti programování.

3.3 Rizika při vývoji

Vývoj her pro virtuální realitu se oproti jiným platformám liší. Musí se zvážit nové faktory.

Prostředí

Pro intenzivní zážitek musí vývojář zajistit věrohodnost prostředí, ve kterém se hráč bude pohybovat a interagovat. Například musí velikosti objektů odpovídat velikostem z reálného světa. To stejné platí pro gravitaci a fyziku předmětů. Také záleží na žánru hry. Samozřejmě výjimky jsou hry, které nesimulují běžné podmínky.

Kinetóza

Je nepříjemný stav vyvolaný přesunem z místa na místo jinak než chůzí či během – například rychlá jízda autem. Při používání virtuální reality je tento jev běžný. Mozek je zmatený, když vidí, že se tělo pohybuje, ale nohy zůstávají stát na zemi. Omezením opakování grafických vzorů – například čtvercová podlaha. A hlavně zajistit dostatečný počet snímků za sekundu během hraní [15].

Celková pohodlnost

Zážitek z hraní ovlivňuje také komfort při používání aplikace. V některých hrách hráči přirozeně interagují s prostředím – například sebrání kostky. V jiných jsou nuceni například pro manipulaci se zbraní udělat několik úkonů, které jsou namáhavé – vytáhnout zbraň, nabít zbraň, vystřelit, vyměnit zbraň atd. mnohdy vyžaduje používání jiných tlačítek. V nepohodlné ovládání v kombinaci s kinetózou může významně ovlivnit herní zážitek.

3.4 Platformy pro publikování

Oculus Store

Pro distribuci aplikací virtuální reality pro zařízení od firmy Oculus. Ovšem pro publikování aplikace na Oculus Store musí aplikace projít náročným výběrovým řízením [6] a musí dosahovat prvotřídních kvalit.

²Unreal Engine, dostupné z <https://www.unrealengine.com/>

³Amazon Sumerian, dostupné z <https://aws.amazon.com/sumerian/>

App Lab

Alternativou oficiálního publikování aplikací na platformu Oculus je App Lab. Stačí splnit technické požadavky. Vhodné pro testovací verze aplikací, experimenty [13].

Steam

Služba pro digitální distribuci her a aplikací, primárně počítačové hry. V případě virtuální reality se jedná o PCVR.

Playstation Store

Pro distribuci aplikací virtuální reality pro platformu PlaystationVR.

Google Play a Apple App Store

Pro distribuci aplikací virtuální reality pro mobilní telefony slouží Google Play a Apple App Store.

Kapitola 4

Návrh aplikace

V herním průmyslu existuje kategorie párty her. Jedná se o typ her, kterou si může skupina kamarádů zahrát večer po náročném dni a soutěžit proti sobě. Hlavně tedy relaxovat a zasmát se.

Cílem této práce je vytvořit prototyp herní aplikace, který by implementoval základní minihry s možností rozšíření v budoucnu.

Tomuto žánru se ve virtuální realitě blíží Rec Room¹ a VRChat². V těchto hrách hráči mají možnost tvořit své vlastní zážitky a sdílet je s ostatními hráči. Jedná se spíše o sociální hry a uživatelé tvoří jakékoliv zážitky – například jako virtuální divadlo, karaoke, ale i minihry patřící do zmíněné kategorie.

4.1 Základní popis

Jedná se o párty hru s balíčkem miniher pro více hráčů. Základní myšlenka je připojení 2-4 hráčů do společné místnosti. Zakládající hráč určí minihru. V momentě, kdy jsou všichni hráči připraveni, spustí se první minihra. Hráči v rámci minihry soutěží mezi sebou. Každá minihra se liší od ostatních svými pravidly a cílem, zároveň spolu sdílí některé prvky. Liší se podmínky vítězství, po jejichž splnění se oznámí vítěz a hráči se vrací do čekací místnosti.

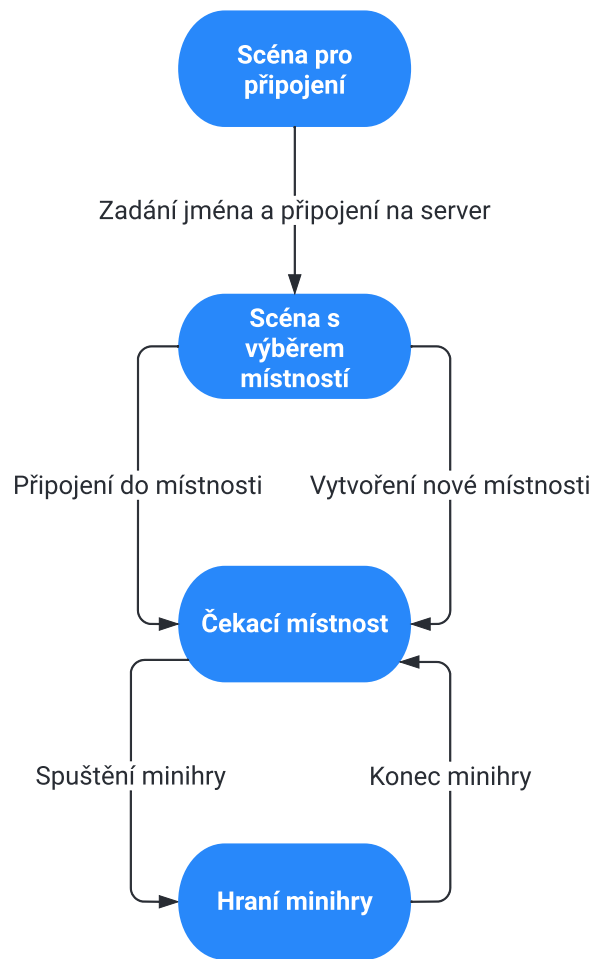
Minihry se zaměří na vtipnou soutěž s přáteli. Cílem je, aby se hráči především bavili soutěžením s přáteli a mohli u hry relaxovat.

4.2 Návrh struktury aplikace

Po spuštění aplikace by měl mít hráč možnost přihlásit se. V hlavním menu bude mít hráč možnost vytvořit novou místnost pro sebe a ostatní hráče, anebo se připojit do již existující místnosti vytvořené jiným hráčem. V místnosti se sejdou hráči. Jakmile jsou všichni hráči připraveni, tak se spustí minihra. Po odehrání minihry se všichni hráči vrací zpět do počáteční místnosti a mohou začít hrát znovu. Základní schéma aplikace lze vidět na obrázku 4.1.

¹Rec Room: <https://recroom.com/>

²VRChat: <https://store.steampowered.com/app/438100/VRChat/>



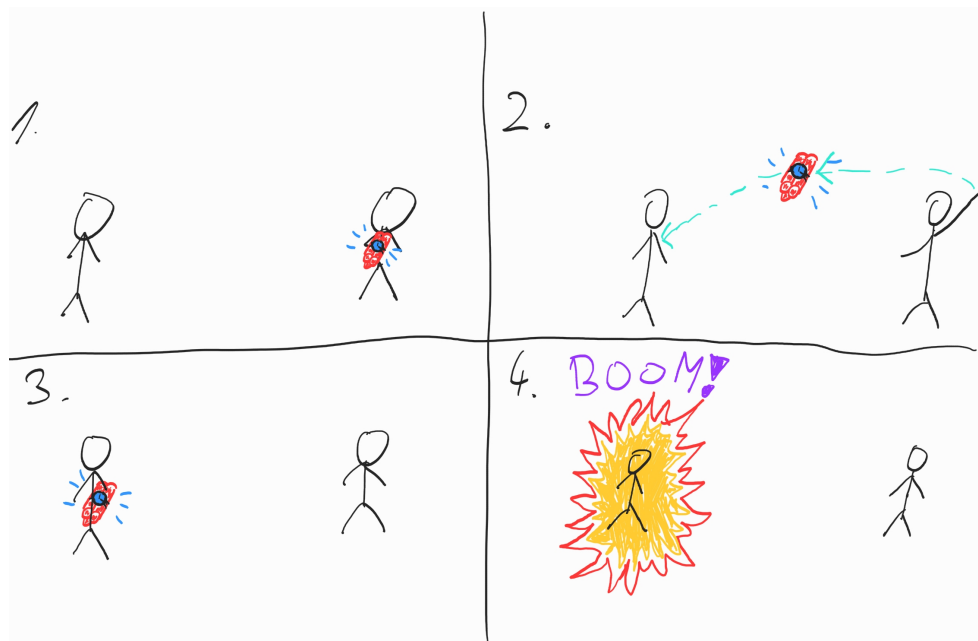
Obrázek 4.1: Schéma struktury aplikace.

4.3 Návrh miniher

Navrhl jsem několik miniher, ve kterých budou hráči soutěžit.

Hra s bombou

V místnosti se objeví časovaná bomba. Když vybuchne zabije blízké hráče. Bomba se „přilepí“ na hráče, ten ji musí odhodit na někoho jiného (viz obrázek 4.2). Poslední vyhrává.



Obrázek 4.2: Ukázka principu hry s bombou

Prvky

- Bomba – objeví se na náhodně vybraném hráči a spustí se odpočet pro explozi. V rytmu tikání problikává dioda. Exploze zabíjí hráče v dosahu.
- Přilepení na tělo – bomba v extrémní blízkosti hráče se mu usadí na hrudníku. Hráč při pohledu dolů uvidí tikající bombu. Bude muset ji sejmout ze svého těla a hodit.
- Bonusy – v aréně se na předem určených místech objeví náhodná krátkodobá vylepšení, jako například zvýšená rychlost, jednorázový teleport, nesmrtelnost nebo přesný hod.

Cíl minihry

Zůstat poslední na živu. Hráč „usmrcený“ explozí je vyřazen z minihry a dál se neúčastní. Minihru vyhrává hráč, který jako jediný nebyl vyřazen.

Prostředí

Minihra se odehrává v aréně připomínající paintballové hřiště a hráči sdílí společný prostor. Rozloha je středně velká, tj. hráči se mohou kolem sebe pohybovat, ale i být vzdáleni na bezpečném místě. Aréna má překážky, za které je možné se schovat před vhozenou bombou. V aréně se nachází místa, kde se objevují zmíněná vylepšení.

Rizika

Jedním z rizik je situace, kdy po hodů bomby by zůstala neaktivně ležet na zemi (na žádného hráče by se nepřilepila). Ani jeden hráč by nebyl ochotný bombu vzít do ruky a riskovat explozi (vyřazení). Možná řešení:

1. Let k nejbližšímu hráči – bomba vzlétne a rychle poletí k nejbližšímu nebo náhodně vybranému hráči až se na něho přilepí.
2. Bonus pro odvážného hráče – hráč, který vezme neaktivní bombu bude odměněn. Odměna je v podobě zmíněných bonusů.

Střílení kachen

Hráči se ocitnou na střelnici a střílí na terče. Za zásah terče dostanou body. Hráči si během soutěže mohou navzájem škodit. Ve scéně jsou k dispozici i nepodstatné „vtipné“ předměty (například pánev), kterým může hráč svého protivníka udeřit. Takový úder je pro pobavení, body jsou zásahy terčů.

Prvky

- Terč – obyčejné terče přidávají pouze body. Součástí minihry jsou i speciální terče, které se vizuálně liší a po zásahu vyvolají speciální události.
- Zbraně – základní zbraní je pistol do jedné ruky, protože je velice jednoduchá na ovládnutí. Přístup k dalším zbraním hráči získají po vyvolání jedné ze speciálních událostí. Může se jednat o luk, vzduchovku nebo kámen.
- Speciální události – krátkodobé změny pravidel v minihře. Může je však i vyvolat skrytá zábavná akce, jako například hráč udeří pálkou třikrát každého hráče. Příklady speciálních akcí:
 1. Změna zbraně – hráči anebo všem se změní zbraň
 2. Oslepení – ostatní hráči nevidí terče

Prostředí

Hráči sdílí menší ohraničený prostor. V okolí prostoru (360 stupňů) se objevují terče na náhodných místech.

Rizika

Špatně viditelné terče by mohly celou hru sabotovat. Vhodné řešení pro případ splnutí objektu s prostředím je zvýraznění obrysu objektu výraznou barvou.

Klasický „Gungame“

„Gungame“ je poměrně známý herní mód ve střílečkách. Je definována posloupnost zbraní. Hráči začínají minihry s první zbraní v pořadí. Za zabití protivníka hráč získává další zbraň v pořadí.

Prvky

1. Velké množství zbraní – aby hra neskončila rychle. Zároveň je důležité se zaměřit na typy zbraní, díky kterým virtuální realita využije svůj potenciál. Například u luku se musí natáhnout tětíva, u vzduchovky „zlomit“ hlaveň a vložit brok anebo odjistit granát a hodit.

Cíl minihry

Mít poslední zbraň v pořadí a „zabít“ hráče.

Prostředí

Podobně jako u „Hry s bombou“ je aréna inspirovaná paintballovým hřištěm dobrou volbou kvůli možnostem krytí.

Rizika

Manipulace s některými typy zbraní mohou být pro hráče až moc náročné, že minihra přestane být záživná.

Badminton s výbušninami

Nad hlavami hráčů se objeví pár menších výbušnin (kulatého tvaru). Hráč musí tyto výbušniny odpálit pryč, aby přežil a nejlépe tak, aby výbušnina dopadla do prostoru jiného hráče a ohrozila ho.

Prvky

- Odpalování výbušnin – hráč drží v ruce badmintonovou raketu. Z prostoru protihráče na něj letí výbušnina, tak hráč ji musí raketou odpálit na jinou stranu.
- Štít – hráč může aktivovat energetický štít, který ho chrání před výbuchem. Štít se po použití musí dobít, takže ochrání pouze před jedním výbuchem.

Cíl minihry

Stejně jako u předchozí minihry s bombou „Zůstat poslední na živu“.

Prostředí

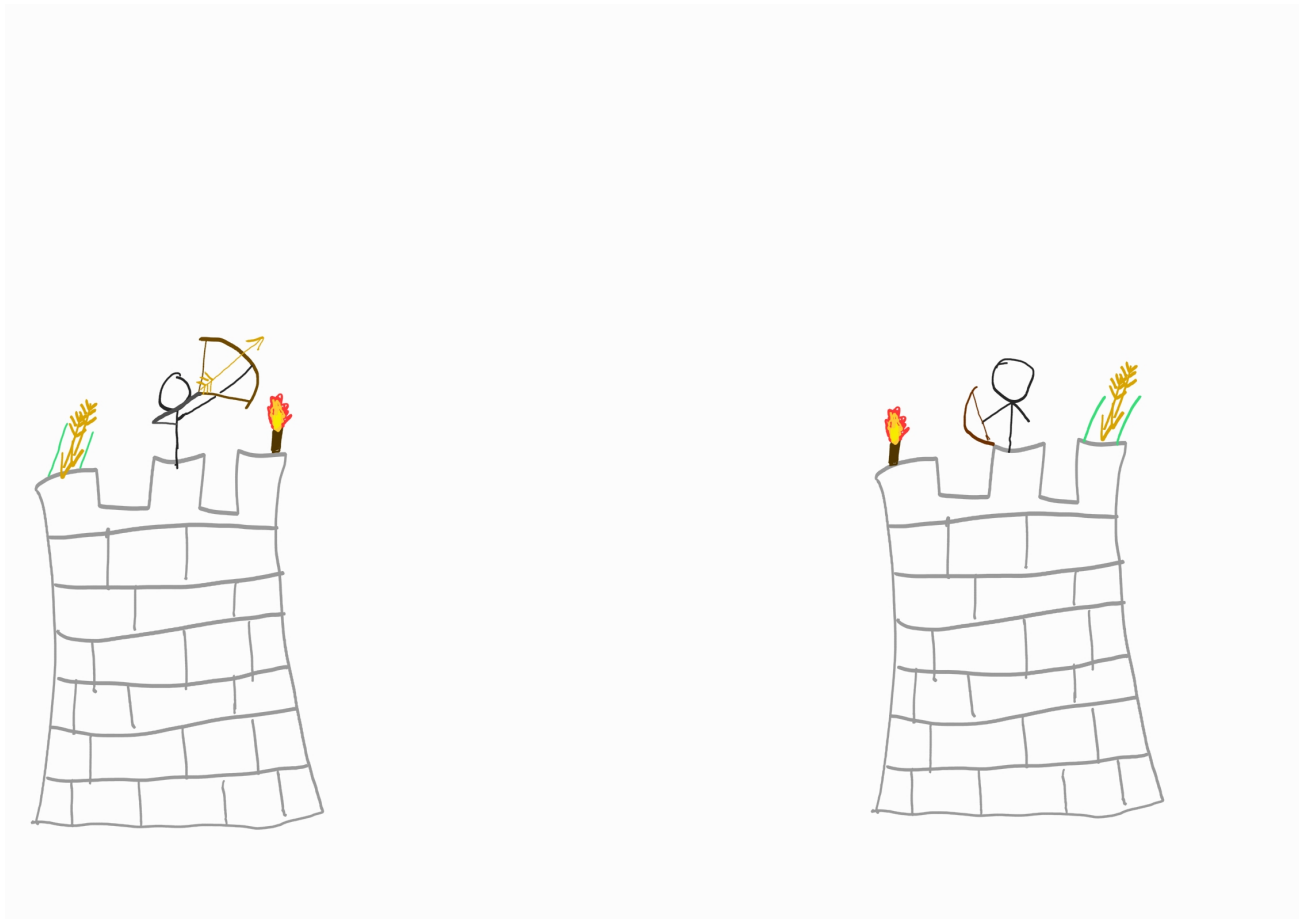
Hráči jako při volejbalu mají svůj vlastní prostor. Tyto prostory jsou odděleny vysokou sítí nebo průhlednou zdí, přes kterou se výbušniny „přepinkávají“.

Rizika

Pokud bude výbušnin málo, tak může nastat situace, kdy budou zapojeni jen dva hráči a ostatní se budou jen dívat bez možnosti se aktivně přidat. Přidáním dalších výbušnin se musí brát ohled na možné příliš rychlé ukončení hry.

Ničení pevností

Každý hráči bude mít vlastní pevnost. Pevnosti jsou od sebe vzdáleny na střední vzdálenost (viz obrázek 5.8). Hráči si vzájemně postupně poškozuji své pevnosti pomocí různých nástrojů.



Obrázek 4.3: Náčrt minihry „Ničení pevností“.

Prvky

- Pevnost – skládá se ze zničitelných částí, aby bylo poškození viditelné a hráči měli představu v jakém stavu je pevnost
- Dělo – součástí pevnosti je dělo. Hráč musí dělo manuálně nabít (např. koule). Poté nastavit
- Opravy – hráč poškozenou část opraví a stav pevnosti se zlepší.
- „Zabití“ protivníka – hráč má k dispozici luk s šípy. Vystřelený šíp pevnost nepoškodí, ale slouží přímo k „zabití“ protivníka. Po zásahu hráč nemůže opravovat svoji pevnost ani střílet z děla. Musí počkat na oživení.

Cíl minihry

Zničit pevnosti ostatních hráčů. Hráč s poslední stojící pevností vyhrává minihru.

Prostředí

Pevnosti budou vysoké a nacházet se v otevřené krajině.

Rizika

Hráč by mohl najít jeden správný úhel, jak správně střílet z luku nebo děla. Pokud by nebylo možné hráče nijak „vyrušit“, tak by minihru mohl vyhrát rychle a jednoduše. Další riziko je pro hráče, kteří trpí strachem z výšek – pevnosti budou vysoko.

Systém bodování

Za odehranou minihru hráč získá body. Tyto body se sčítají a hráči je mohou porovnávat po odehrání několika miniher. Navrhl jsem a zhodnotil tři systémy bodování.

1. Všichni dostanou body – každý hráč za umístění v minihře dostane určitý počet bodů. Za umístění na prvním místě 5 bodů, na druhém 3 body, na třetím 2 body a poslední 1 bod.
 - Výhody: jednoduchá implementace. Když hráč nevyhraje, tak získá „alespoň něco“.
 - Nevýhody: musí být větší počet miniher.
2. Zasloužené body – pořadí v minihře není až tak důležité. Kolik bodů hráč dostane záleží přesně na tom, jak moc bude plnit cíl minihry. Například kdyby se soutěžilo v chytání ryb, tak hráč získá tolik bodů, kolik chytil ryb.
 - Výhody: hráči jsou odměněni za všechny úspěchy ve hře.
 - Nevýhody: náročnost miniher bude různá a je nutné vybalancovat počty získaných bodů. Tj. aby si jeden hráč poctivě nevydřel body ve dvou minihrách, ale jiný hráč ho dožene po jedné lehčí minihře.
3. 1 bod nebo 0 bodů – hráč za umístění na prvním místě získává 1 bod, ostatní 0 bodů.
 - Výhody: jednoduchá implementace
 - Nevýhody: ostatní hráči, kteří také vynaloží nadměrný výkon, nejsou odměněni.

Pohyb

Pohyb ve virtuálně realitě je stále nedokonalý. Virtuální realita dokáže simulovat pohyb po povrchu, ale parkourové triky si uživatel neužije. Dalším problémem je kinetóza (viz sekce 3.3).

- Joystick – nejčastější řešení. Hráč se pomocí jednoho joysticku pohybuje ve směru stisku, tzv. „Artificial Locomotion“³ – může způsobovat nevolnost. Pro uživatele je také namáhavé se neustále točit kolem sebe. To řeší druhý joystick, který rotuje hráče ve scéně. Toto řešení je jednoduché na implementaci a ocení ho uživatelé, kteří nemají prostorný pokoj, tudíž se nemohou volně pohybovat.

³Artificial Locomotion – pohyb ve virtuální realitě, který neodpovídá fyzickému pohybu [7]

- „Na vlastních nohou“ – Uživatel se pohybuje ve virtuální realitě fyzickým pohybem. Uživatelé jsou omezeni velikostí místnosti, takže uživatelé se můžou pohybovat i kombinací joysticku a vlastního pohybu.
- „Treadmill“ – zařízení umožňuje uživateli fyzicky „chodit na místě“, podobně jako na běžícím pásu (viz obrázek 4.4). Zařízení se využívá e-sportových hrách, protože je možné i sprintovat. Nevýhodou je vyšší cena. A zařízení tohoto typu nejsou tak rozšířená, takže se zatím nemusí vyplatit tyto zařízení podporovat.



Obrázek 4.4: Virtuix Omni VR Treadmill

Zdroj: https://omni.virtuix.com/images/omnion_e_jump_left.jpg

- Teleport – hráč si pomocí joysticku vybere místo, kam se chce přesunout a po uvolnění se objeví na místě. Ve této aplikaci tento způsob nemůže být primární, protože by to narušilo základní herní mechaniky. Vhodnější je použití teleportu k implementování speciálních mechanik, jako například zmiňovaný jednorázový teleport v minihře „Hra s bombou“ v sekci 4.3.
- Lezení – způsob vertikálního pohybu. Hráč se ovladači uchopí určitých míst na zdi a leze nahoru (jako na horolezecké zdi). Je to náročnější na implementaci. Existuje riziko způsobení nevolnosti – hráč leze na zdi, ale nohy jsou fyzicky na zemi. Do navržených miniher není potřeba toto řešení implementovat. Ovšem pro návrh nových miniher v budoucnu je dobré tuto mechaniku pohybu neopomenout.

Uchopení objektů

Hráč bude mít možnost interagovat s předměty ve hře. Ovladače budou ve hře reprezentovány jako modely dlaní. Hráč bude moci uchopit předmět do dlaně, jakmile bude ovladač v blízkosti předmětu. Podržením „grab“ tlačítka (viz obrázek 4.5) zůstane předmět v dlaní. V momentě přerušení stisku bude předmět upuštěn.

Speciální akce předmětů

Uchopený předmět bude možné i použít (aktivovat). Typický příklad jsou zbraně. Tlačítko „trigger“ (viz obrázek 4.5) k aktivaci schopnosti (u zbraní střelba).

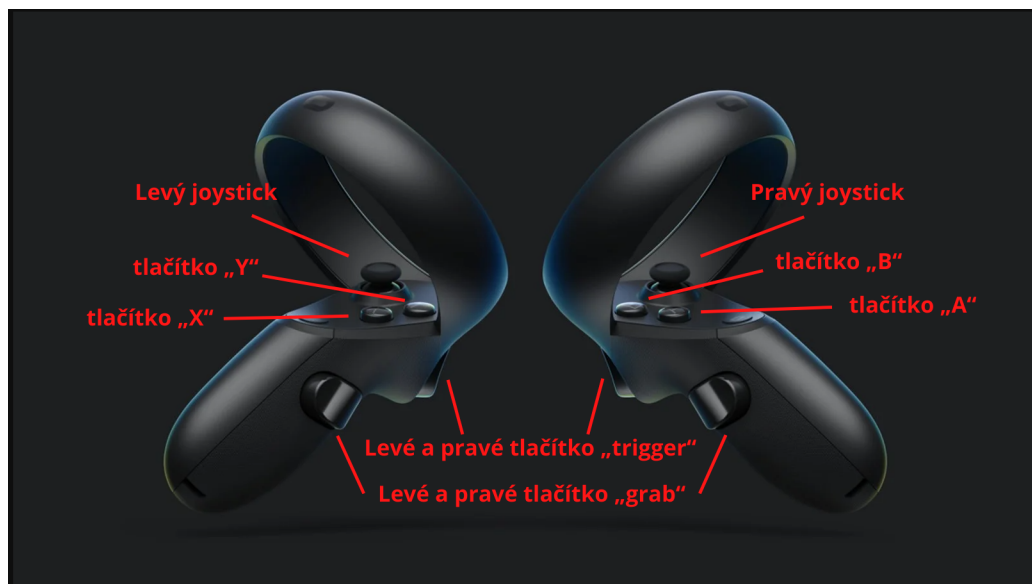
4.4 Uživatelské rozhraní

Drtivá většina zážitků ve virtuální realitě zobrazuje ovladače, které hráč drží, jako 3D modely rukou. To pomáhá uživateli se lépe orientovat a zvyšuje věrohodnost prostředí.

- „Point-and-Click“ – před uživatelem se nachází plátno a pomocí paprsku (laseru) interaguje s prvky. Toto řešení je jednoduché jak na implementaci. Pro uživatele je ovládání svižné a pochopitelné.
- Přímá interakce – prvky uživatelského rozhraní jsou aktivována pomocí přímého dotyku ovladače [11].

4.5 Základní ovládání

Minihry sdílí základní ovládání. Není rozumné měnit většinu ovládání každou minihru. Je důležité zachovat konzistenci, jinak by byl hráč zmatený. Na obrázku 4.5 jsou ovladače zařízení Oculus Quest 2 a jsou pojmenovány vstupy.



Obrázek 4.5: Rozvržení ovládání na ovladačích Oculus Quest 2.

Návrh uživatelského ovládání aplikace:

- Levý joystick – hráč se pohybuje pomocí „Locomotion“ (viz sekce 4.3).
- Pravý joystick – otočení ve směru stisku.
- Levé a pravé tlačítko „grab“ – uchopení předmětu do příslušné dlaně.
- tlačítko „X“ – přepínání činnosti vstupů tlačítka „trigger“ – pro interakci s UI nebo speciální akce předmětu
- tlačítko „A“ – teleport v některých minihrách
- Levé a pravé tlačítko „trigger“
 - Aktivace speciální akce 4.3 předmětu při uchopení
 - Interakce s UI

4.6 Volba síťové infrastruktury

Byla provedena analýza a testování některých existujících nástrojů(knihoven) pro vývoj her pro více hráčů. Předpokládá, že v minihry budou pro 2-4 hráčů. Tudíž požadavkem na síťové řešení je možnost vytvořit krátkodobé místnosti, kde se sejdou 2 hráči, případně až 4.

Photon PUN

Photon PUN⁴ je komerční služba poskytující infrastrukturu pro vývojáře her pro více hráčů. Je s ní poměrně jednoduché začít vyvíjet. Na druhou stranu programátor nemá vždy úplný přehled, co se děje v pozadí a řešení vážnějších problémů může být obtížné. Architektura se jeví jako „peer-to-peer“ – server vývojář neimplementuje a veškerá logika je implementována na klientech. Avšak komunikace mezi klienty probíhá přes server Photon Cloud, který je vývojáři uzavřený.. Photon PUN je více popsán v sekci 5.3.

- Výhody: již vybudovaná síťová infrastruktura, vhodné pro začátečníky, systém místností.
- Nevýhody: uzavřená platforma, komerční (zkušební verze pro omezený počet lidí apod.), v případě použití Photon PUN není autorativní server.

Darkrift 2 Networking

Darkrift 2 Networking⁵ je open source knihovna vyvinuta jednou osobou. Programátor musí vybudovat celou síťovou infrastrukturu sám a vývoj celé aplikace je časově náročnější. Díky tomu má ale programátor přehled, co vše se v aplikaci děje a může ji, co nejlépe optimalizovat. Jedná se o server-klient architekturu.

- Výhody: open source, programátor má plnou kontrolu, co se bude dít a dokáže to přesně určit, velice rychlý, autorativní server
- Nevýhody: náročnější implementace – programátor musí celou síťovou infrastrukturu vybudovat z ničeho

⁴Photon PUN: <https://www.photonengine.com/en/PUN>

⁵Darkrift 2 Networking, dostupné z <https://www.darkriftnetworking.com/>

Photon Server

Pro vývojáře, kterým nestačí Photon PUN a mají zájem o implementaci vlastního autorativního serveru, je určen Photon Server SDK. Jedná se o nádstavbu Photon Cloud, kdy vyvojáři na infrastruktuře staví logiku hry na straně serveru. Musí se použít v kombinaci Photon PUN nebo jiným produktem na straně klienta od společnosti Exit Games. Photon Server je více popsán v sekci 5.9.

- Výhody: možnost implementace autorativního serveru.
- Nevýhody: nedostatečná dokumentace/příklady implementace.

Ostatní síťové nástroje

Za zmínku stojí také Mirror⁶. Tato open-source knihovna nebyla v rámci této práce nijak testována, ale je pozitivně hodnocena vývojářskou komunitou a známé tituly na něm fungují⁷.

⁶Mirror, dostupné z <https://github.com/vis2k/Mirror>

⁷Příklady použití Mirror: <https://mirror-networking.com/showcase/>

Kapitola 5

Implementace

Původně byla uvažována pouze implementace klientské části bez vlastního serveru (s využitím Photon Cloud). Ale po zvážení výhod využití serveru, byl implementován i server pro zpracování režijní logiky hry. Než jsem začal vlastní implementaci naučil jsem se základní práci s XR Interaction Toolkit a Photon PUN z kurzu „Multiplayer Virtual Reality (VR) Development With Unity“¹ na vzdělávacím portálu Udemy², dále jen „kurz“. Aplikace vychází z návrhu struktury aplikace v sekci 4.2.

5.1 Oculus Quest/Quest 2

Oculus Quest a Quest 2 (viz obrázek 5.1) jsou zařízení vyvíjeny společností Oculus, která byla odkoupena společností Facebook Inc., dnes Meta Inc. Jedná se o standalone headsety pro virtuální realitu. Operační systém je založen na systému Android. Oproti PCVR jedná se o samostatné zařízení, pro jehož používání není potřeba vysoce výkonný stolní počítač. Nevýhoda je však méně kvalitní grafické zpracování. Díky Oculus Link je možné právě využít i funkce PCVR, kdy výpočetní výkon běží na připojeném počítači a obraz se promítá do zařízení [23].

¹Kurz „Multiplayer Virtual Reality (VR) Development With Unity“, dostupné z <https://www.udemy.com/course/multiplayer-virtual-reality-vr-development-with-unity/>

²Udemy, dostupné z <https://www.udemy.com>



Obrázek 5.1: Oculus Quest 2.

Vybral jsem jako cílovou platformu tyto zařízení kvůli jejich konzistenci při testování. Jedná se pouze o dvě zařízení, na které se bude vyvíjet. PCVR zařízení existuje mnoho a každé má jiný systém vstupů. V rámci této práce není prostor testovat takové množství zařízení.

5.2 Unity3D

Multiplatformní herní engine primárně pro vývoj 2D a 3D her. Unity také vyvíjí nástroje i pro ostatní odvětví průmyslu. Systém animací a efektů pro filmový průmysl, vizualizační nástroje pro architekturu, rozšířená realita například pro manufaktury [16], či тренаžéry pro konkrétní obory pro nácvik různých scénářů.

Herní engine Unity poskytuje herním vývojářům nástroje pro renderování scén s objekty s texturami a nasvícení. Pro pokročilejší uživatele je možnost dané nástroje optimalizovat.

Software také dokáže nasimulovat fyzikální z reálného světa – gravitace, působení síly, pohyb [18]. Což je důležité právě pro aplikace virtuální reality, aby hráč cítil správnou atmosféru.

Samotná implementace herních mechanik se realizují pomocí knihovny UnityEngine a programovacího jazyka C#. Zajišťuje mapování vstupů, funkce pro fyziku, úprava vizuálních prvků, herní logika, matematické funkce.

Důležitou částí Unity ekosystému je „Unity Asset Store“. Místo, kde herní vývojáři mohou publikovat „assety“ pro jiné vývojáře, i za poplatek. Může se jednat o knihovny, modely, zvuky či jiná rozšíření pro herní engine Unity [17]. Lze tedy tuto službu využít pro komerční účely.

Jedna z dalších výhod je obrovské množství návodů a tutoriálů pro vývoj her v Unity. Pro začátečníka je jednodušší začít s vývojem her právě v Unity než v konkurenčním Unreal Engine. Používání Unity je zdarma pro společnosti s obratem 100 000\$.

GameObject a komponenty

Základní element v Unity je GameObject. GameObject reprezentuje danou entitu ve scéně (viz sekce 5.2), jako například hráče nebo předměty. Skládá se z komponent, které popisují chování objektu a každá vykonává vlastní přiřazenou funkci [19]. Příklady komponent:

- **Transform** – popisuje pozici, rotaci a velikost ve scéně
- **Renderer** – zajišťuje render modelu, aby byl viditelný ve scéně
- **Rigidbody** – GameObject se bude chovat podle zákonů fyziku
- **Collider** – zajišťuje kolize s jinými elementy, které jsou vybaveni komponentou Collider
- **Camera** – zajišťuje přenos obrazu na výstupní zařízení uživatele

Unity povoluje vývojářům vytvářet vlastní komponenty, tzv. „scripts“ [20] – skripty. Tyto skripty jsou programovatelné v jazyce C#. Jeden skript odpovídá jedné třídě a dědí z `MonoBehaviour` [21].

Scéna

Scéna je 2D nebo 3D prostředí, které obsahuje objekty hry a které lze zobrazit uživateli pomocí kamery. Hra se skládá s jedné nebo více scén. Zpravidla se použití více scén používá na vytvoření různých odlišných prostředí – menu, úroveň ve městě, úroveň v lese.

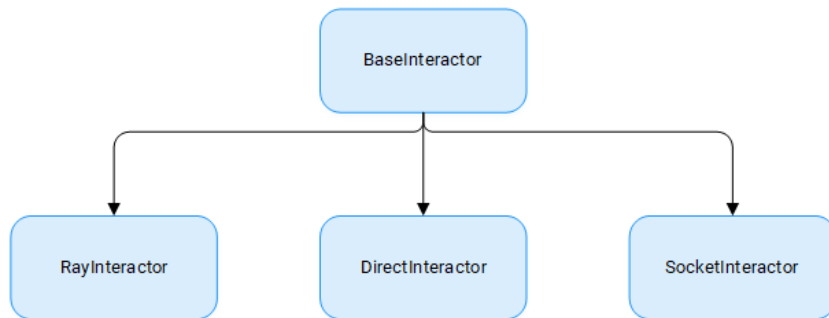
Knihovna UnityEditor

Herní engine Unity také disponuje knihovnou Unity Editor, která slouží k úpravě enginu k vlastním potřebám při vývoje hry. Knihovna využívá například pro příjemnější testování nově implementovaných funkcí vývojářem. V případě této práce implementací tlačítek do editoru, která budou simulovat interakci s uživatelským rozhraním, konkrétně s tlačítky pro připojení na server, připojení do místnosti, zahájení minihry. Tato implementace rychle umožní vývojáři otestovat například, zda se hráč správně připojuje do hry, jelikož si nemusí nasazovat headset pro virtuální realitu a manuálně klikat na tlačítka.

XR Interaction Toolkit

Pro vývoj software pro virtuální realitu v Unity slouží XR Interaction Toolkit. Jedná se o vysoko-úrovňovou knihovnu, která pomocí vstupů ze zařízení zajišťuje interakci s objekty ve scéně [22]. Základními prvky jsou Interactor a Interactable.

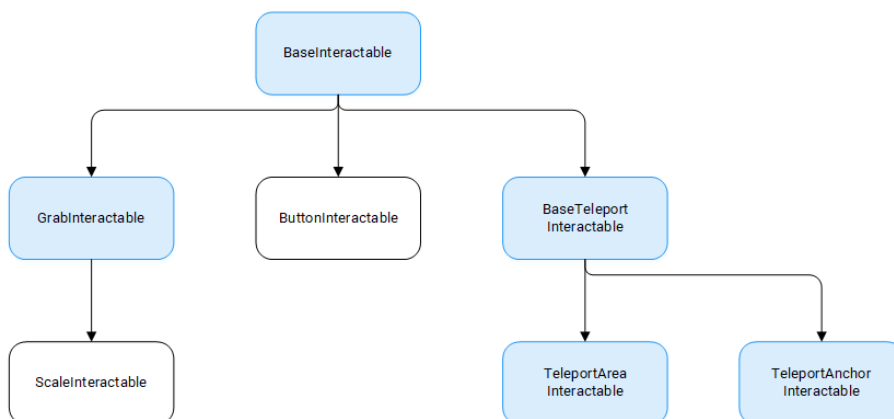
Interactor (viz obrázek 5.2) – jsou zodpovědní za akce jako sebrání objektů ve scéně, interakce probíhá s objekty Interactable. Například také získávají vstup z ovladačů v situaci, kdy hráč chce uchopit předmět. Na těchto objektech se také implementují další akce, když nastanou události jako – uchopení objektu, upuštění objektu atd.



Obrázek 5.2: Schéma třídy Interactor

Zdroj: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.9/manual/index.html>

Interactable (viz obrázek 5.3) – jedná se o objekty ve scéně, které mohou být sebrány objekty Interactor. Pro představu – může se jednat například o knihu na stole, kterou hráč může sebrat do ruky



Obrázek 5.3: Schéma třídy Interactable

Zdroj: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.9/manual/index.html>

Vstupy

Knihovna poskytuje základní systém pro pohyb ve virtuální realitě. **Locomotion System** [7]. Toto řešení může způsobovat nevolnost pro uživatele s menší zkušeností s virtuální realitou. Je k dispozici také možnost implementace teleportu. Vstupy jsou popsány v sekci 4.5 a jejich použití v sekci 5.4.

XR Rig

Knihovna poskytuje základní `GameObject`, obsahující kameru a dva objekty `XR Controller`, jako abstrakci rukou. Kamera se ve scéně otáčí podle hráčova fyzického pohybu hlavou. Oba `XR Controller` zase kopírují fyzický pohyb rukou.

5.3 Photon PUN

Síťová infrastruktura na straně klienta je implementována pomocí Photon PUN³. Běžně komunikuje s Photon Cloud, ale v této aplikaci byl implementován vlastní server pomocí Photon Server SDK (viz sekce 5.9). Část síťové herní logiky je implementována na straně klienta, jako například společné volání metod nebo synchronizace pohybu. Režie hry je implementována na straně serveru. Photon PUN splňuje požadavky z návrhu 4.6 na „systém místností“. V rámci `Game Server` je možnost vytvořit několik na sobě nezávislých místností.

Komunikace s ostatními klienty a serverem

Photon PUN a Server využívají ke komunikaci `RaiseEvent` metody pro posílání zpráv pomocí. `RPC`⁴ metody jsou metody, které zavolá jeden hráč a daná metoda se provede také na ostatních klientech, jedná se o způsob synchronizace událostí. Zpráva oznamující volání `RPC` metody se první pošle na server a poté ostatním klientům, v jádru se však stále jedná o `RaiseEvent`. Vývojář tuto komunikaci na serveru již nijak neimplementuje, server není příjemce. Je možné však v rámci implementace byznysové logiky tuto `RPC` metodu v síťovém toku identifikovat.

Pro přímé poslání zpráv `RaiseEvent` se používá metoda `PhotonNetwork.RaiseEvent` [3]. Jedná se o zprávy, které mají přiřazený tzv. `TAG`, který zprávě přiloží odesílatel a příjemce je schopen určit účel zprávy a správně ji zpracovat. Tagy jsou definovány v třídě `EventCodes`.

Photon View

Komponenta, kterou musí mít každý `GameObject`, co bude jakkoliv komunikovat po síti. Komponenta monitoruje a synchronizuje potřebné proměnné mezi ostatními klienty. Photon View obsahuje atribut `viewID`, který identifikuje objekt v síťové komunikaci.

Ownership a Photon.Instantiate

Každá komponenta `PhotonView` má vlastníka, což je jeden z klientů. Vlastnictví může například určovat, kdo může s daným objektem manipulovat. Vlastnictví je možné měnit podle typu vlastnictví. Ve Photon PUN jsou tři typy vlastnictví [4]:

- **Fixed** – vlastnictví není možné měnit.
- **Request** – klient musí požádat vlastníka o převzetí vlastnictví, žádosti je vyhověno na základě implementovaných herních mechanik.
- **Takeover** – klient nemusí žádat vlastníka o převzetí vlastnictví a stává se vlastníkem ihned.

³Photon Pun, dostupný z <https://www.photonengine.com/en/PUN> a <https://assetstore.unity.com/packages/tools/network/photon-pun-2-120838>

⁴Remote Procedure Calls

Ve většině případů je první vlastník ten kdo objekt vytvořil. Pro správnou inicializaci po síti se používá metoda `PhotonNetwork.Instantiate` [5], čímž se také přiřadí `viewID`. Pro správnou destrukci `PhotonNetwork.Destroy`.

Smooth Sync PUN2

Pro plynulou synchronizaci je použit skript `Smooth Sync`⁵, dostupný na Unity Asset Store. Skript garantuje eliminaci zbytečného síťového provozu. Tento skript je využit primárně pro synchronizaci pozice a rotace objektů.

5.4 Network Player

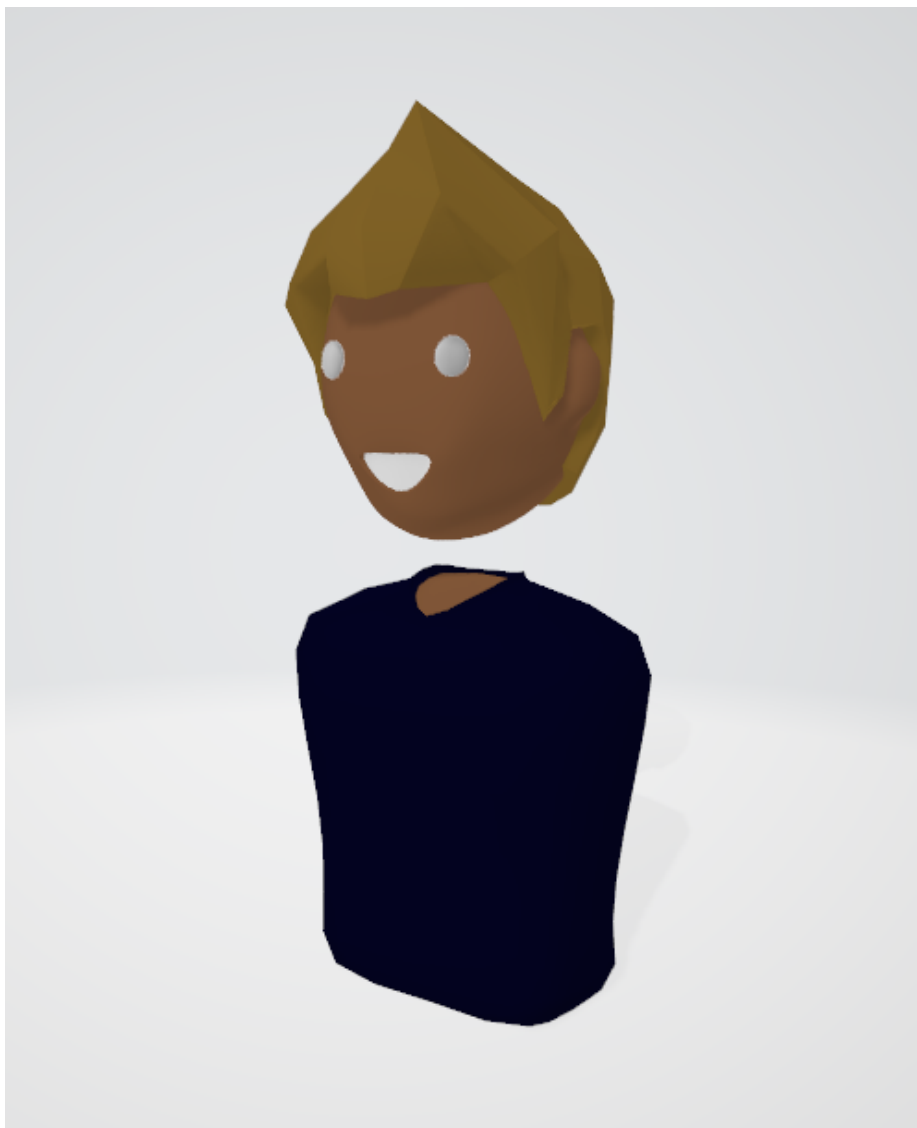
Každého klienta reprezentuje `GameObject Network Player`, který může být buď lokální anebo vzdálený. Skript `PlayerNetworkSetup` má na starosti zobrazení těla a hlavy, zda se jedná o lokálního nebo vzdáleného `Network Player`. Tento skript je převzat z kurzu. Skript `MultiplayerVRSynchronization` byl převzat z kurzu – slouží pro plynulou synchronizaci částí těla při pohybu s ostatními klienty. Skript `LocalPlayerNetworkController` je vytvořen pouze pro lokálního hráče a přijímá zprávy ze serveru.

Avatar

Pro vizuální podobu hráče byl použit 3D model a textura od Emma Kohncke⁶. Avatar lze vidět na obrázku 5.4.

⁵Smooth Sync, dostupný z <https://assetstore.unity.com/packages/tools/network/smooth-sync-96925>

⁶3D model a textura reprezentující hráče, dostupné z <https://sketchfab.com/3d-models/low-poly-male-avatar-02-vr-87bb06b3d4ac48c0ae8d5d76bf0d6211>



Obrázek 5.4: Avatar reprezentující hráče.

Ovládání

Na základě návrhu ovládání v sekci 4.5 bylo implementováno ovládání objektu `NetworkPlayer`. Teleport implementován není, protože nebyla implementována minihra, která by tuto mechaniku využívala.

- Levý joystick – hráč se pohybuje pomocí „Locomotion“ [7].
- Pravý joystick – otočení o 30 stupňů ve směru stisku
- Levé a pravé tlačítko „grab“ – uchopení předmětu do příslušné dlaně.
- tlačítko „X“ – přepínání činnosti vstupů tlačítek „trigger“ – pro interakci s UI nebo speciální akce předmětu
- Levé a pravé tlačítko „trigger“

- Aktivace speciální akce (viz sekce 4.3) předmětu
- Interakce s UI

Lokální UI

Každý hráč má své vlastní lokální uživatelské rozhraní, které vidí pouze daný uživatel. Toto uživatelské rozhraní se nachází v kameře pohledu hráče. Jedná se o:

- Notifikace – textové slovní spojení umístěné v horní části rozhraní na krátkou omezenou dobu. Například se jedná o upozornění pro zakladatele místnosti, když se pokusí zahájit hru v momentě, kdy nejsou všichni připraveni.
- Lokální menu – po stisku tlačítka „X“ se zobrazí menu s možností se odpojit ze místnosti.
- Vítězná tabulka – po odehrání minihry se hráči zobrazí tabulka s výsledky. V případě minihry Shooting Range (viz sekce 5.8) se zobrazí seznam hráčů a jejich skóre. V případě minihry Towers (viz sekce 5.8) se zobrazí jen jeden hráč, který se stal vítězem.

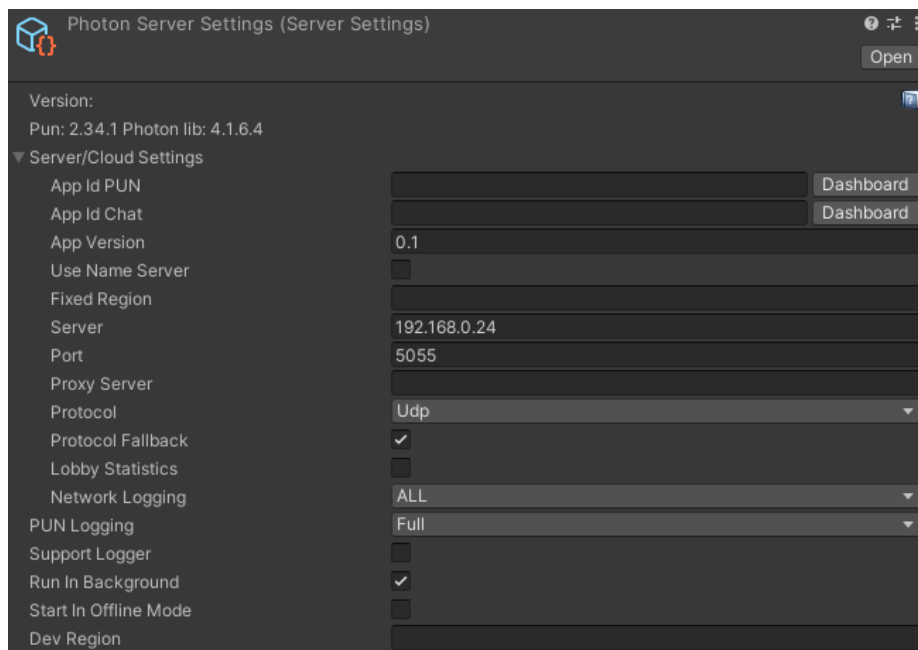
5.5 LoginSceneBP

První scéna, která se načte po spuštění aplikace je `LoginSceneBP`. Hráč vyzván pro zadání přezdívky, která ho bude identifikovat ve hře. Před hráčem jen zobrazena klávesnice jako součást uživatelského rozhraní.

Login Manager skript a připojení na master server

Tento skript připojuje hráče na server pomocí metody `PhotonNetwork.ConnectUsingSettings`. Metoda připojuje klienta na základě konfigurace v souboru `PhotonServerSettings.asset` (viz obrázek 5.5). Pokud se klient úspěšně připojí na `master server`, tak se načte scéna `HomeSceneBP`. Nejdůležitějšími prvky jsou:

- `App Id PUN` – tato položka se používá pouze v případě využití `Photon Cloud`, jedná se o identifikátor aplikace
- `App Version` – určení verze aplikace, pro kontrolu správné verze klienta
- `Server` – IP adresa `master serveru`
- `Port` – Port

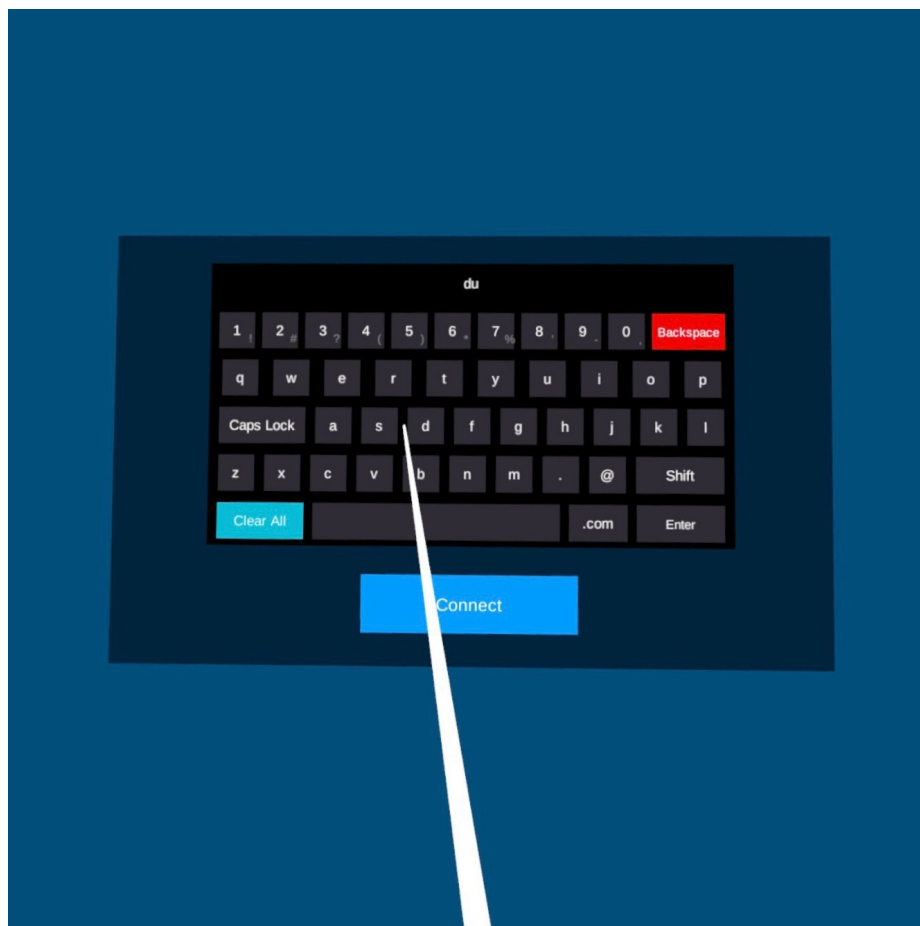


Obrázek 5.5: Photon Server Settings.

VR Klávesnice a uživatelské rozhraní

Při připojení hráče na server je důležité, aby hráč zadal svoji přezdívku. Rozhodl jsem se pro použití jednoduchého panelu s virtuální QWERTY klávesnicí⁷ převzatou z Unity Asset Store. Skript `LoginUIManager` čeká na `Callback` při stisknutí tlačítka „Connect“, poté informuje o stisku instanci třídy `LoginManager`. Uživatel interaguje pomocí `GameObject XRRIg`, jehož kontrolery pro ruce obsahují `XRRayInteractor`, tj. z dlaní se promítá laser, kterým uživatel provádí interakci s uživatelským rozhraním (viz obrázek 5.6).

⁷VR Keyboard, dostupné z <https://assetstore.unity.com/packages/tools/input-management/vr-keyboard-95936>



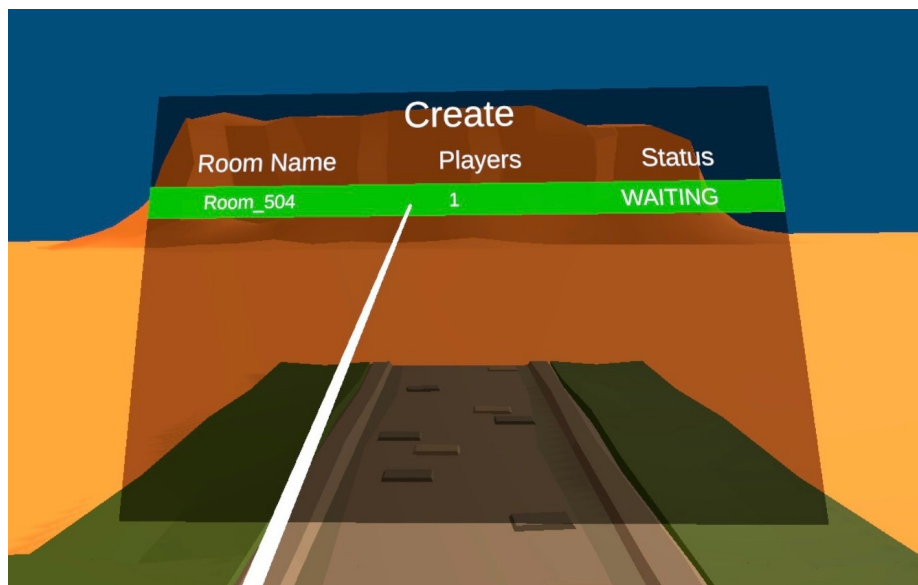
Obrázek 5.6: VR Klávesnice.

5.6 HomeSceneBP

Potvrzením jména se hráč přesune do scény výběru místnosti. Tato scéna obsahuje seznam všech dostupných místností, kam se hráč může připojit. Každá místnost tvoří jeden prvek uživatelského rozhraní v seznamu. Obsahuje informace jako jméno místnosti, počet hráčů a status. Kliknutím na prvek se hráč připojí do zvolené místnosti. Uživatel má také možnost založit vlastní místnost, ta se umístí do seznamu mezi ostatní místnosti.

RoomManager skript

Když je uživatel připojen na serveru, chce se spojit s ostatními hráči připojením do místnosti. Musí mít před sebou plný seznam dostupných místností, kam se může připojit. Implementoval jsem tabulku pro seznam místností (viz obrázek 5.7). Hlavičku tvoří tři sloupce, které prozrazují základní informace o místnosti – jméno, aktuální počet připojených hráčů a status místnosti.



Obrázek 5.7: Seznam dostupných místností.

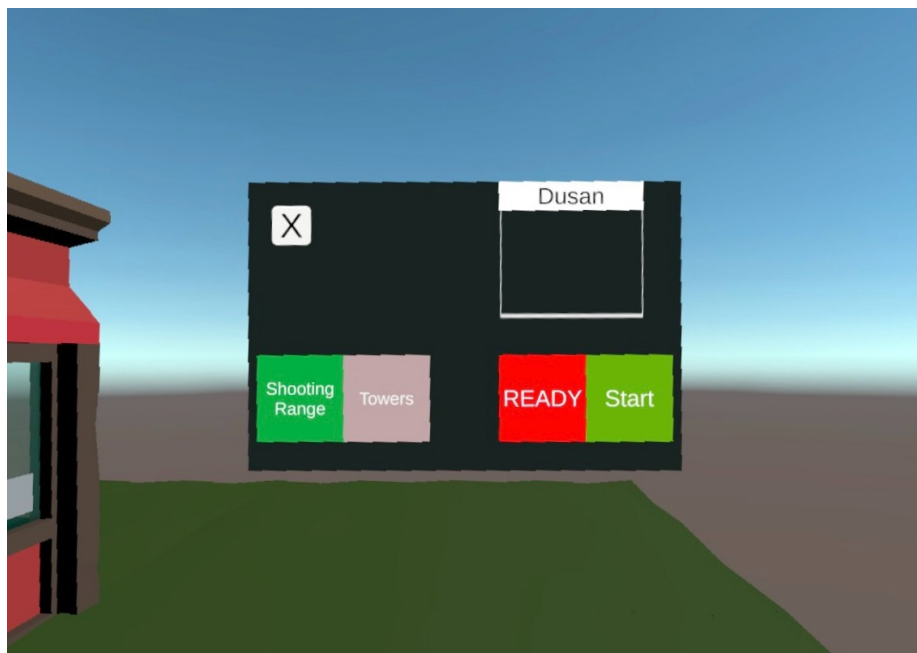
Přepsaná metoda `OnRoomListUpdate(List<RoomInfo> roomList)` třídy `RoomManager`. `List<RoomInfo> roomList` uchovává informace o všech místnostech. Pro každou místnost je přidán prvek, který má charakteristiku tlačítka, do tabulky se seznamem všech her. Prvky se řadí vertikálně. Po kliknutí na zvolenou místnost se v `RoomManager` volá `PhotonNetwork.JoinRoom(roomId)` pro připojení do místnosti a načítá se scéna `Room` (čekačí místnost).

V případě, že hráč chce vytvořit úplně novou vlastní místnost, se nachází nad tabulkou tlačítka „CREATE“ (viz obrázek 5.7). Po kliknutí se zavolá metoda `CreateAndJoinRoom` a uživatel vytvoří novou místnost, do které se mohou ostatní hráči připojit. Rozhodl jsem se umístit toto tlačítka přímo nad tabulku, protože do pravého spodního rohu bych očekával jiné tlačítka (například tlačítka „JOIN“ pro připojení do již existující místnosti, což by některé hráče mohlo zmást).

5.7 Čekací místnost

Před minihrou se hráči schází v čekačí místnosti (scéna `Room`). Každého pro každého hráče je vytvořen pomocí `PhotonNetwork.Instantiate` `GameObject` `NetworkPlayer` (viz sekce 5.4). Ve scéně se nachází společné okno 5.8 uživatelského rozhraní pro připojené uživatele. Okno se rozděluje na dvě části. Levá část slouží pro herní nastavení. Pouze zakladatel (neboli `master client`) místnosti je oprávněn tato nastavení měnit. Nyní se nachází v herním nastavení možnost zvolit následující minihru v levé spodní části. Každá minihra je reprezentována čtvercovým tlačítkem s názvem minihry. Tlačítka právě zvolené minihry má zelenou barvu, aby přítomní hráči rozpoznali, jaká minihra se bude hrát. Pravá polovina obsahuje informace a možnosti interakce týkající se režie místnosti. V pravém horním rohu se nachází vertikální seznam přítomných hráčů. Hráči mohou ze seznamu vyčistit, kdo z hráčů je připraven. Připravený hráč má zelenou přezdívku. Tato informace je důležitá hlavně pro zakladatele místnosti, který může zahájit hru pouze v případě, kdy jsou všichni přítomní v místnosti připraveni. Každý hráč má možnost nastavit, zda je připraven, kliknutím na levé tlačítka „READY“ nacházející se ve spodní části, poté se barva jména změní

na zelenou. Tato změna je viditelná v seznamu u všech hráčů. V momentě, kdy zakladatel místnosti vidí všechny hráče připravené, chápe to jako signál pro zahájení minihry. Minihru zahájí zakladatel místnosti stiskem tlačítka „START“.



Obrázek 5.8: Okno čekací místnosti.

Modely a textury prostředí jsou převzaty z POLYGON City – Low Poly 3D Art by Synty⁸.

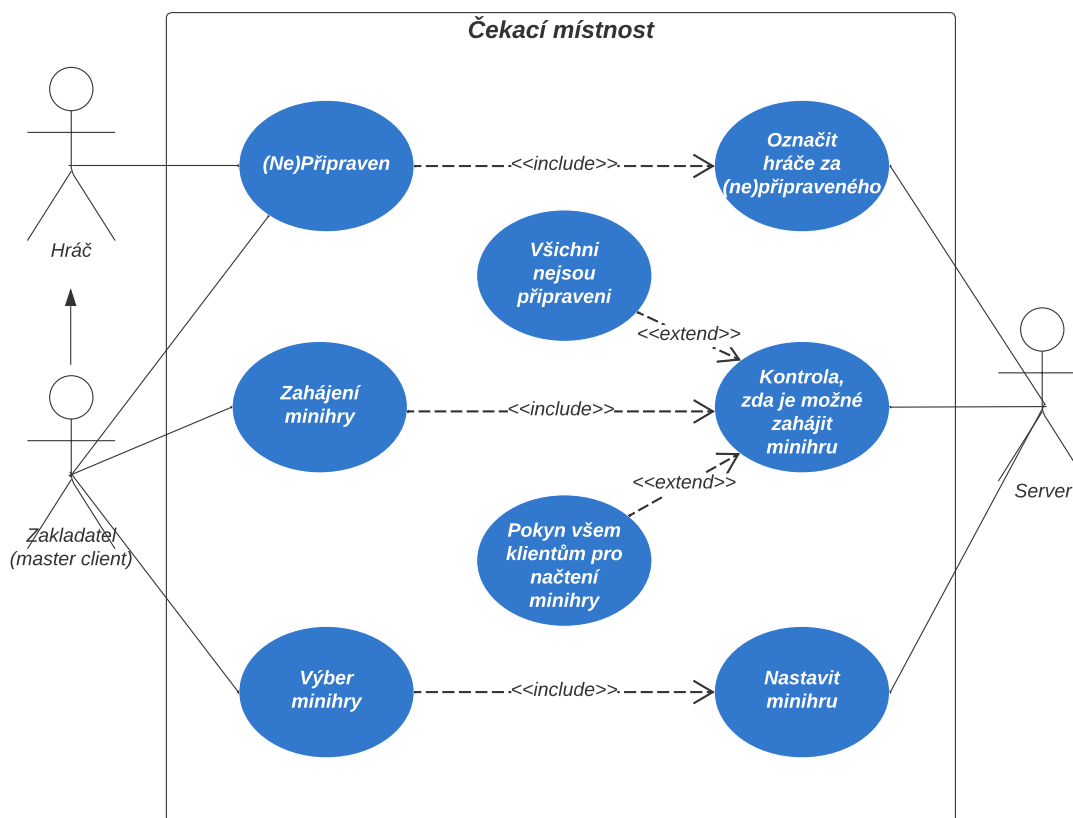
WaitRoomManager

Třída `WaitRoomManager` zajišťuje komunikaci se serverem. V této scéně již probíhá komunikace s částí serveru (viz obrázek 5.9), která byla implementována v rámci této práce a je popsána v sekci 5.9. Jedná se o základní režii místnosti pro nastavení a zahájení minihry. Mezi klienty a serverem se posílají zprávy s vlastně definovány tagy.

- `SET_MINIGAME_TO_SERVER` – odesílaná zpráva předává informaci, jakou minihru `master client` vybral
- `SET_MINIGAME_TO_CLIENTT` – přijímaná zpráva oznamující ostatním klientům změnu minihry, tlačítka minihry změní barvu
- `READY_CLICKED_TO_SERVER` – odesílaná zpráva oznamuje, že daný klient stiskl tlačítko pro přípravu
- `START_CLICKED_TO_SERVER` – odesílaná zpráva oznamuje, že daný `master client` stiskl tlačítko pro zahájení minihry

⁸POLYGON City – Low Poly 3D Art by Synty, dostupné z <https://assetstore.unity.com/packages/3d/environments/urban/polygon-city-low-poly-3d-art-by-synty-95214>. Licence – dostupná z https://unity3d.com/legal/as_terms

- **START_FAIL_TO_CLIENT** – přijímaná zpráva upozorňuje **master client**, že není možné zahájit minihru, protože nejsou všichni účastníci připraveni (tj. neodeslali zprávu **READY_CLICKED_TO** na server)
- **LOAD_MINIGAME_TO_CLIENT** – přijímaná zpráva s parametrem příkazující klientům, aby načetli scénu dané minihry



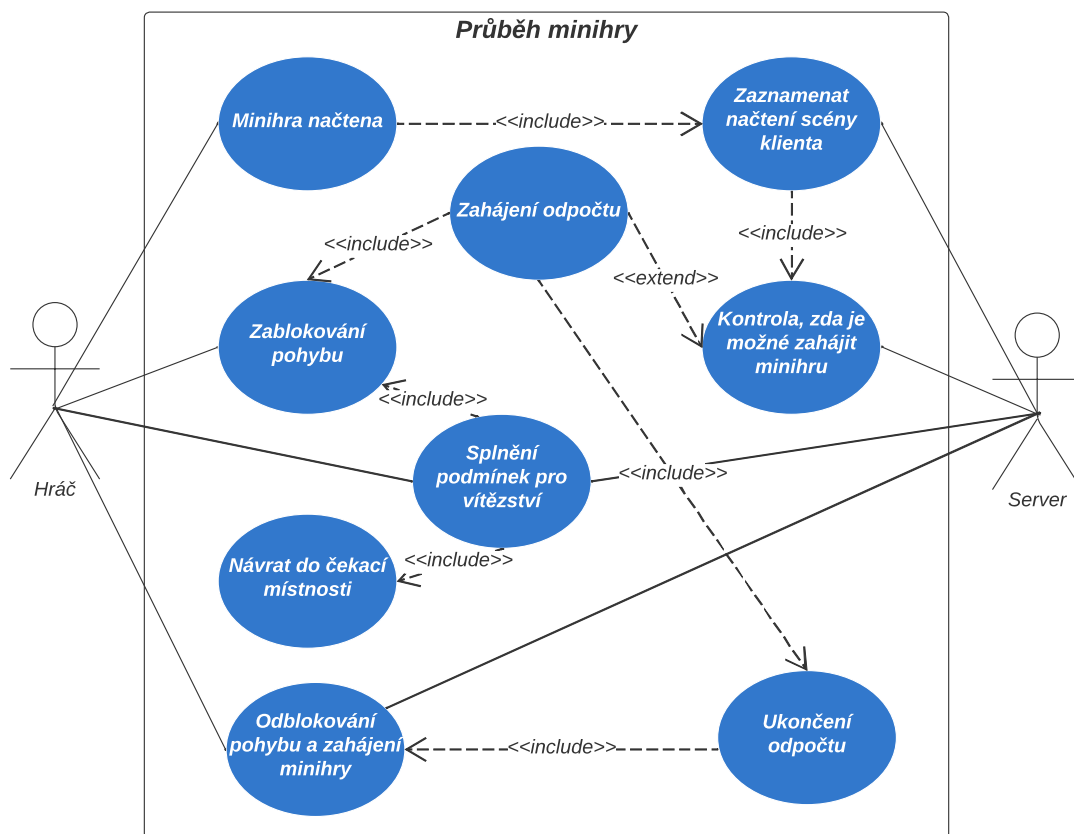
Obrázek 5.9: Schéma čekací místnosti.

5.8 Minihry Shooting Range a Towers

Jakmile se hráči načte scéna pro danou minihru, tak soutěžit se ihned nezačne. Musí se počkat na všechny hráče až budou mít načtenou scénu. Důvod je synchronizace. Je základní eliminovat situace, kdy by zařízení jednoho hráče načetlo scénu, začal přijímat síťové zprávy a už by začal soutěžit. Zatímco druhému se scéna stále nenačetla a měl by značnou nevýhodu. Jakmile tedy zařízení všech hráčů načte scénu minihry, tak se všem zakáže možnost pohybu, přesunou se na danou pozici, a spustí se odpočet do startu minihry. Po odpočtu je pohyb uvolněn a hráči soutěží mezi sebou (viz. obrázek 5.10). Po splnění vítězných podmínek dané minihry, se opět pohyb znemožní. Každému hráči se lokálně objeví vítězná tabulka popisující výsledek minihry. Pod vítěznou tabulkou se nachází tlačítko „Return to waiting room“, po jehož stisknutí se hráč přesune zpět do čekací místnosti.

- **MINIGAME_LOADED_TO_SERVER** – odesílaná zpráva oznamující, že klient načetl scénu

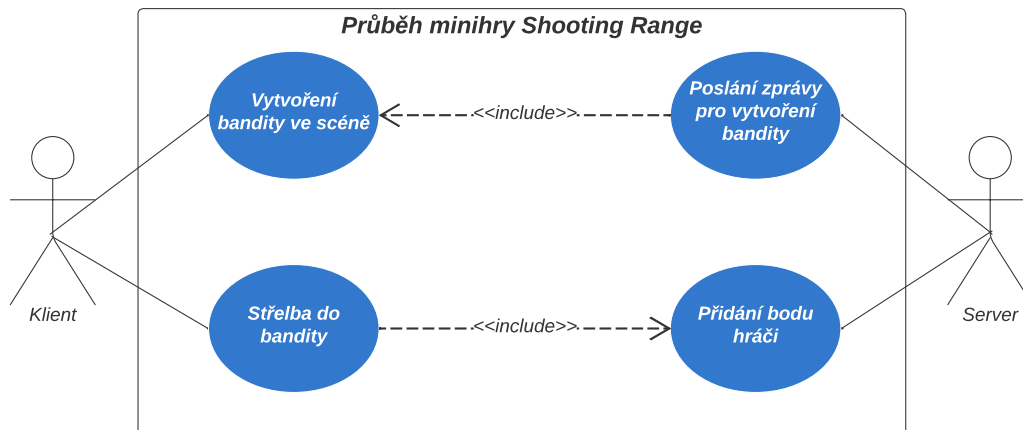
- MINIGAME_COUNTDOWN_TO_CLIENT – přijímaná zpráva zahajuje odpočet před startem mini hry
- MINIGAME_START_TO_CLIENT – přijímaná zpráva zahajuje minihru
- MINIGAME_END_TO_CLIENT – přijímaná zpráva ukončuje minihru



Obrázek 5.10: Schéma obecného průběhu minihry.

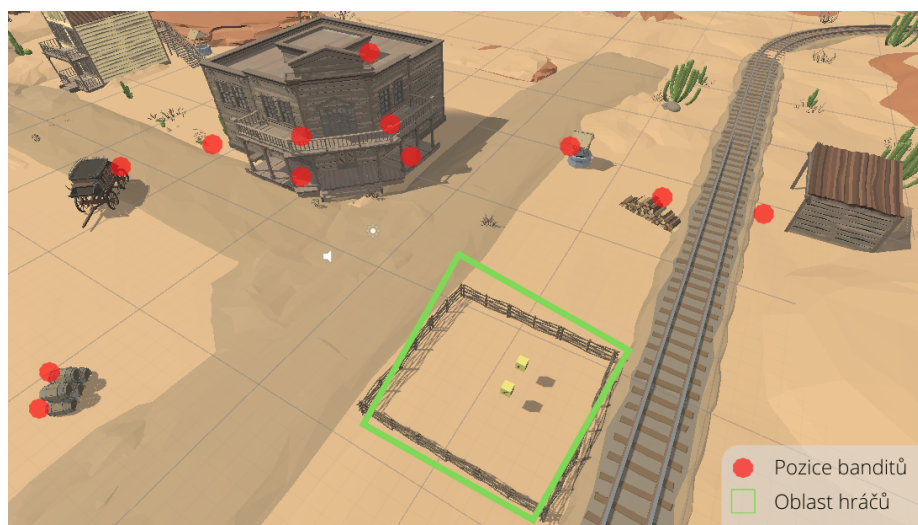
Shooting Range

Jako první minihru jsem zvolil soutěž ve střelbě mezi hráči. Vycházel jsem z návrhu minihry „Střílení kachen“ v sekci 4.3. Ve westernovém prostředí se objevují bandité jako terče. Hráč, který zasáhne nejvíce banditů vyhrává minihru. Aplikace na serveru režíruje události v minihře. Posílá zprávy `SHOOTING_RANGE_SPAWN_TARGET_TO_CLIENT` pro vytvoření bandity a zpracovává událost, kdy je bandita střelen (`SHOOTING_RANGE_ON_HIT_TO_SERVER`) (viz obrázek 5.11). Server uchovává počet zásahů každého hráče. Po uplynutí časového intervalu hraní minihry se zobrazí výsledná tabulka s konečným skóre.



Obrázek 5.11: Schéma průběhu minihry Shooting Range.

Hráči se nachází v ohraničeném prostoru na okraji města. Jedním směrem hráč nahlíží do města, po otočení přes koleje na hory. To může na hráče působit lépe, než uprostřed města, kde by každým směrem scéna vypadala stejně. V plném okruhu od hráčů jsou přednastavené pozice, na kterých se budou bandité objevovat (viz obrázek 5.12). Všechny 3D modely z této scény jsou převzaty ze zakoupeného balíku POLYGON Western – Low Poly 3D Art by Synty⁹.



Obrázek 5.12: Prostředí minihry Shooting Range.

Implementace bandity

Banditu vizuálně reprezentuje model z POLYGON Western – Low Poly 3D Art by Synty (viz obrázek 5.13). `TargetSpawnManager` vytvoří banditu a podle získaného indexu pozice ze

⁹POLYGON Western – Low Poly 3D Art by Synty, dostupné z <https://assetstore.unity.com/packages/3d/environments/historic/polygon-western-low-poly-3d-art-by-synty-112212>. Licence – dostupná z https://unity3d.com/legal/as_terms

serveru nastaví pozici. Bandita má červený obrys kvůli přehlednosti, na střední vzdálenost stále není bandita plně rozeznatelný a červený obrys tuto nedokonalost kompenzuje. Pro vytvoření obrysu byl použit skript `Outline`¹⁰, který byl převzat z Unity Asset Store.



Obrázek 5.13: Bandita.

Na `GameObject` bandity je skript `ShootingTarget` (implementuje `Hittable`) popisuje chování bandity jako terče. Metoda `OnHit` se volá, když hráč zasáhne střelbou banditu. Po zásahu metoda posílá na server zprávu s tagem `SHOOTING_RANGE_ON_HIT_TO_SERVER` (zprávu posílá pouze hráč, který zasáhl cíl, aby server poznal, kdo terč zasáhl) a volá RPC metodu `RPC_GotHit`. Jelikož se jedná o RPC metodu, tak se zavolá na všech klientech. Tato metoda zajišťuje změnu textury model na červenou barvu a přehrání zvuku zranění.

Skript `ShootingTarget` také obsahuje dvě pole prvků `AudioClips`. Pole `clips` obsahuje zvukové stopy, které se přehrají po vytvoření bandity. Pole `deathClips` obsahuje zvukové stopy, které se přehrají po zásahu. Zvuky se přehrávání pomocí komponenty `AudioSource`. Klíčové je nastavení parametru `Spatial Blend` na hodnotu 1. Toto nastavení zaručí přehrání zvuku prostorově.

¹⁰`Outline`, dostupné z <https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>

Revolver

Byla naimplementována jedna zbraň, kterou hráči zasahují bandity (viz obrázek 5.14). Revolver vizuálně reprezentuje model z POLYGON Western - Low Poly 3D Art by Synty. Uchopení zbraně ve virtuální realitě zajišťuje komponenta XRGrabInteractable a skript NetworkGrabbing pro sebrání a umístění do dlaně hráče.



Obrázek 5.14: Hráč drží revolver během minihry.

Samotnou střelbu vykonává skript `Revolver`. Pro věrohodnost revolver nevystřelí ihned po stisku tlačítka. Silnějším stiskem se natahuje úderník a rotuje válcový zásobník tak jako u reálného revolveru. Maximálním stiskem tlačítka revolver vystřelí. Výstřelem udeří úderník a zásobník zaujme novou pozici. Střelba se provede zavoláním RPC metody `RPC_Shoot`. Vytvoří se instance třídy `Ray`, v pozici hlavně revolveru vznikne paprsek mířící vpřed. A použitím metody `Physics.Raycast` se paprsek vygeneruje ve scéně. Pokud paprsek má kolizi s `Hittable`, tak zavolá i jeho metodu `OnHit`. Součástí metody je také přehrání zvuku výstřelu.

Zvukové efekty a hudba

Důležitým faktorem her je zvuk, hráčům to pomáhá ponořit se do atmosféry. Zvukové efekty jsou zakoupeny¹¹ z Epidemic Sound¹².

- Střelba – po stisknutí tlačítka pro střelbu revolver vydá zvuk střelby. Tento zvuk je důležitý pro uživatelský zážitek, aby pochopil, že právě vystřelil z revolveru a měl zpětnou vazbu.
- Bandita – Při vytvoření bandita vykřikne hlášku, aby na sebe upozornil. Zajímavým prvkem je použití 3D zvuku. Při vývoji jsem si všiml, když se objeví nový bandita mimo aktuální rozhled hráče, tak hráč ho bude chtít, co nejdříve najít a střelit. Aby se otočil na správné místo, musí být implementován prostorový zvuk, díky kterému se hráč lépe zorientuje, kde se nový bandita objevil. Po zásahu zazní zvuk zranění. Jeho účel slouží k lepší zpětné vazbě pro hráče, že terč byl zasažen.
- Hudba – Vybral jsem dynamickou hudbu, která přináší pocit rychlé soutěže a zábavy (Jeremy Korpas – Finding Synergy¹³).

Towers

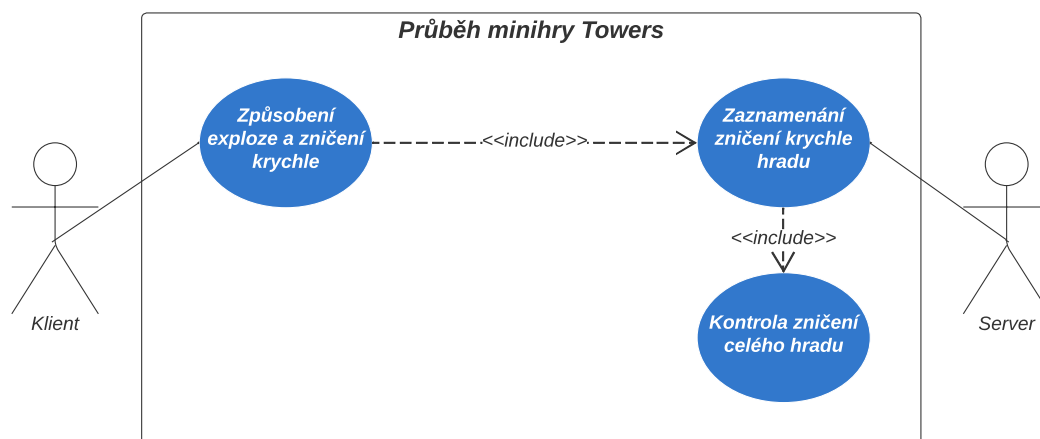
V druhé minihře má každý hráč svoji vlastní pevnost. Tentokrát jsem postupoval podle návrhu minihry „Ničení pevností“ v sekci 4.3. Minihra je implementována pro dva hráče. Oproti návrhu se liší způsob ničení pevností – je použit luk s výbušnými šípy. Nejsou implementovány opravy a „zabití“ protivníka. Pevnost se skládá z malých krychlí, ze kterých se sestaví pevnost. Při zničení jednotlivé krychle pevnosti se odešle na server zpráva informující tuto událost (viz obrázek 5.15). V momentě zničení určitého počtu krychlí v pevnosti, pevnost považuje za zničenou a minihra se ukončí. Jméno vítězného hráče je zobrazeno v tabulce. Prostředí (země a hory) je převzato z POLYGON Knights – Low Poly 3D Art by Synty¹⁴.

¹¹Získaná licence je k dispozici na přiloženém paměťovém médiu.

¹²Epidemic Sound, dostupné z <https://www.epidemicsound.com/>

¹³Jeremy Korpas – Finding Synergy, Licence Creative Commons, dostupné z <https://www.youtube.com/watch?v=G-iNdHnDYWg>

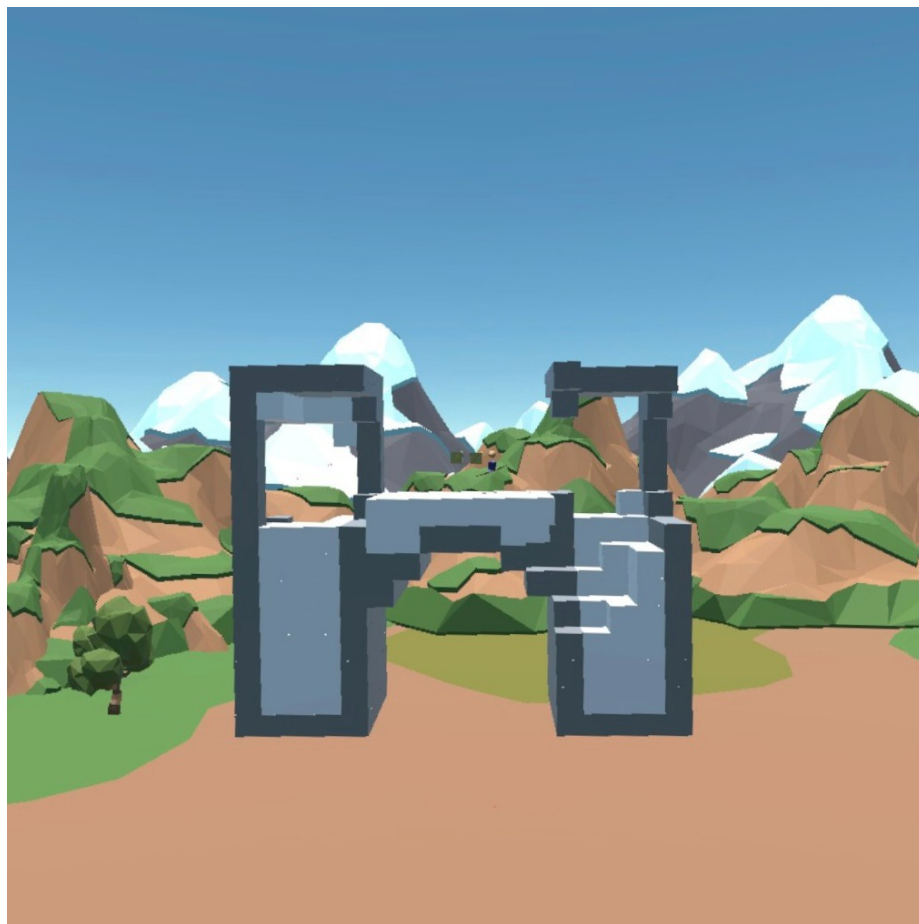
¹⁴POLYGON Knights – Low Poly 3D Art by Synty, dostupné z <https://assetstore.unity.com/packages/3d/environments/fantasy/polygon-knights-low-poly-3d-art-by-synty-83694>. Licence – dostupná z https://unity3d.com/legal/as_terms



Obrázek 5.15: Schéma průběhu minihry Towers.

Sestavení pevnosti

Samotná pevnost se skládá z menších krychlí (GameObject `Brick` a `EdgeBrick`), které jsou společně postaveny do hradu se dvěma věžemi pomocí skriptu `Brick`. Hráč se pohybuje pouze v prostoru nad bránou, tj. na hradbě. Pokud v blízkosti krychlí nastane exploze ze šípu, tak jsou smazány. Tato mechanika tvoří věrohodný vizuální efekt postupně bořícího se hradu, což lze vidět na obrázku 5.16. Všechny krychle jsou zničitelné po explozi ze šípu, kromě plochy nad bránou, kde se pohybuje hráč. Důvodem nezničitelnosti této části je zamezení pádu z pevnosti, případně složité umístění zpět na „polozničenou“ pevnost.



Obrázek 5.16: Poškozená pevnost po zásahu šípy.

Střelba z luku

Hráč vystřeluje výbušné šípy pomocí luku, který simuluje luk z reálného světa, např. padnutí šípu do říhy, určení síly odpalu podle natáhnutí tětiny. Šíp po výstřelu letí po své trajektorii. Za šípem vzniká vizuální efekt „ocasu“ pro lepší pozorovatelnost šípu během letu. Po zásahu krychle pevnosti vznikne `GameObject Explosion`. Luk byl naimplementován pomocí tutoriálu „VR Bow and Arrow for Unity“¹⁵ na Youtube kanále VR with Andrew¹⁶. Model a textura luku je převzata z balíku POLYGON – Bow and Crossbow Pack¹⁷.

Exploze

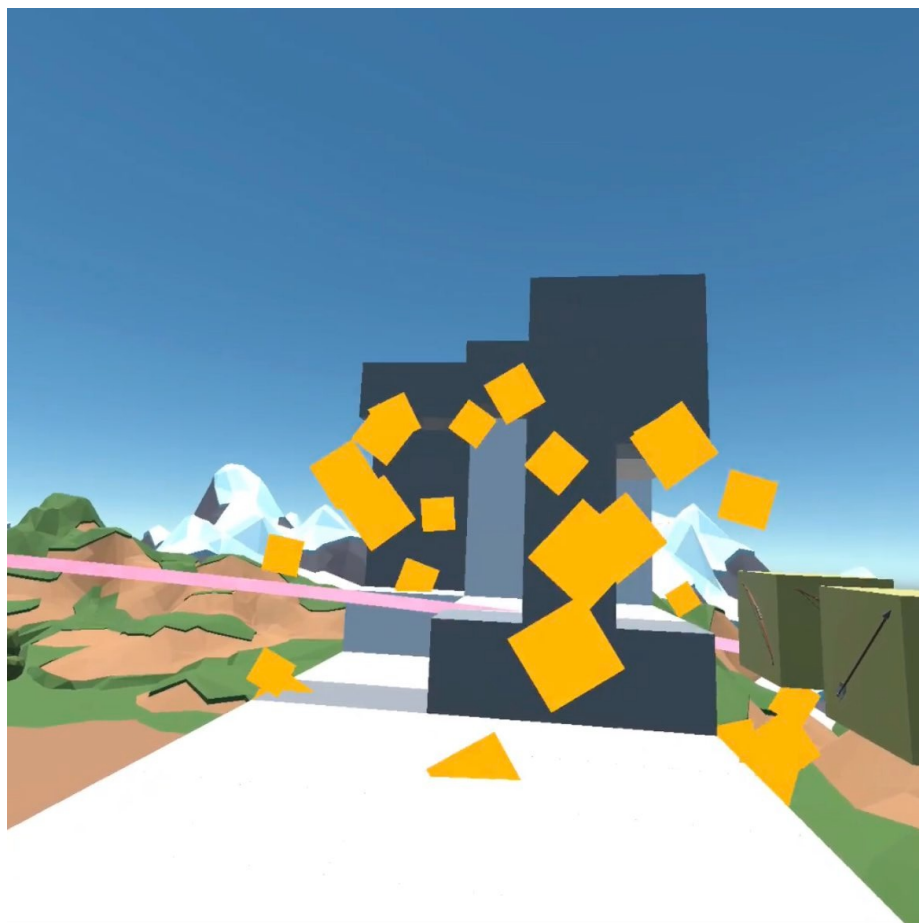
Explozi reprezentuje `GameObject Explosion`. Stejnomený skript vytvoří vizuální efekt připomínající reálnou explozi a přehraje zvukový efekt výbuchu (viz obrázek 5.17). Metoda

¹⁵VR Bow and Arrow: Part 1, dostupné z <https://www.youtube.com/watch?v=ukRVW1fbwtc>, Part 2 dostupné z <https://www.youtube.com/watch?v=QWxhNjF2eiM>, Part 3 dostupné z <https://www.youtube.com/watch?v=ZJWbu-QhT9g>, Update dostupné z <https://www.youtube.com/watch?v=H0xTz4JtWiI>

¹⁶VR with Andrew, dostupné z <https://www.youtube.com/c/VRwithAndrew>

¹⁷POLYGON – Bow and Crossbow Pack, dostupné z https://syntystore.com/pages/free-assets?_pos=2&_sid=8ec730d3c&_ss=r.Licence – dostupná z <https://syntystore.com/pages/end-user-licence-agreement>

`Physics.OverlapSphere` vyhledá všechny krychle v dosahu. Za každý zničený `GameObject Brick` (nebo `EdgeBrick`) se na server odesílá zpráva `TOWERS_BRICK_DESTROYED_TO_SERVER`.



Obrázek 5.17: Exploze při dopadu šípu.

Zvukové efekty

Zvukové efekty jsou také zakoupeny z Epidemic Sound¹⁸. V této minihře nebyl zvuk tak důležitý pro herní výkon tak jako v předchozí minihře. Avšak bylo potřeba implementovat alespoň zvukové efekty exploze a šípu.

- Exploze – V kombinaci se zničenými krychlemi tento zvuk vytváří lepší atmosféru souboje.
- Vystřelení šípu – pro lepší zpětnou vazbu uživateli, že vystřelil šíp.

5.9 Implementace Serverové části

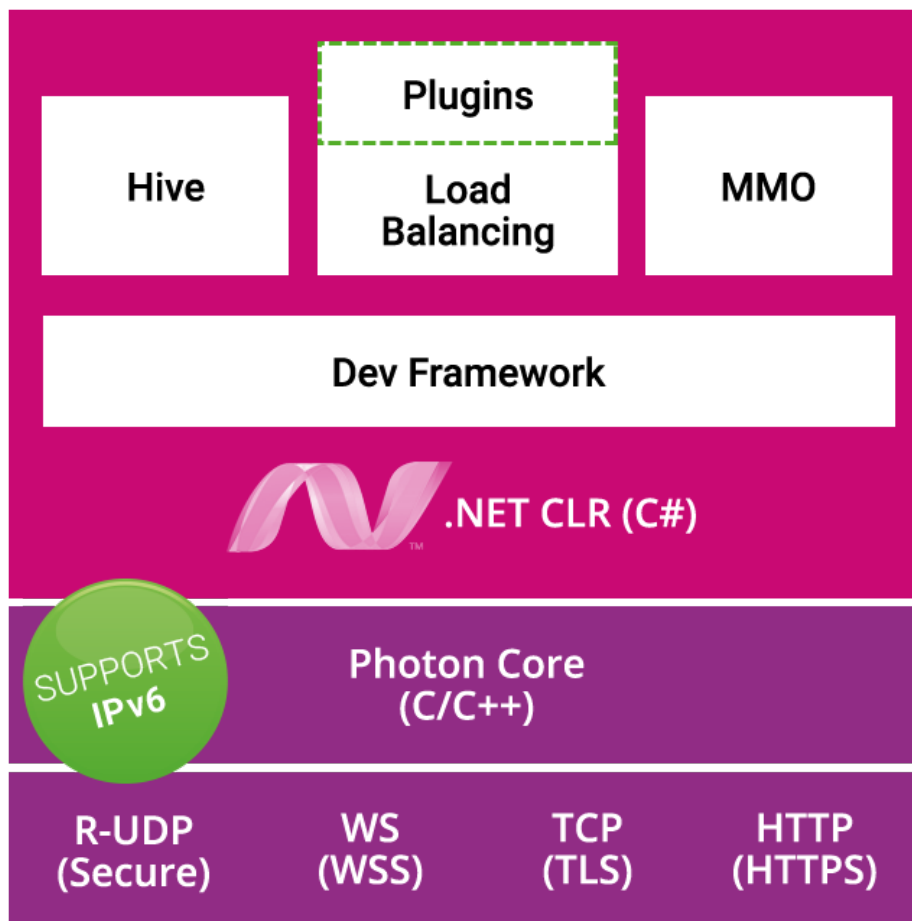
Původně byla zamýšlena pouze klientská část za použití Photon PUN s využitím Photon Cloud. Nicméně jsem se rozhodl pro implementaci i (částečně) autorativního serveru, který by režíroval deterministické stavy. Motivací bylo nespřátovat veškerou autoritu klientovi a

¹⁸Získaná licence je k dispozici na přiloženém paměťovém médiu.

také se připravit na implementaci plně autorativního server (například včetně simulace pohybu na straně serveru) v budoucnu.

Photon Server SDK

Jedná se o síťové řešení pro hostování serveru na vlastním zařízení a implementací s vlastní logikou aplikace. Jedná se o nástavbu Photon Cloud, kterou vývojář rozšiřuje. Jádru SDK je napsáno v C++ a je poskytováno .NET C# API pro implementaci byznysové logiky vývojáři (viz obrázek 5.18).



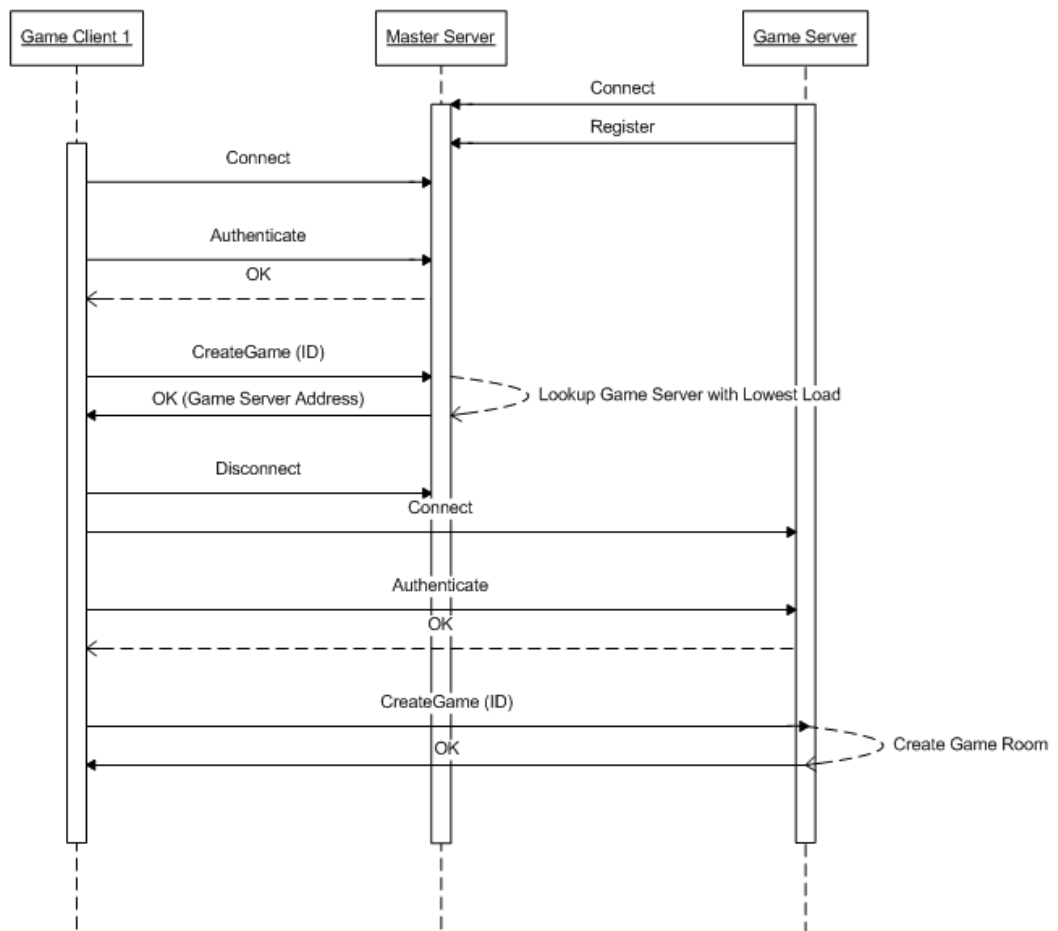
Obrázek 5.18: Architektura Photon Server.

Zdroj: <https://www.photonengine.com/Content/img/photon-server-architecture.png>

Důležitou výhodou Photon Server je vysoká škálovatelnost. **Master Server**¹⁹ monitoruje stavy všech **Game Server**²⁰ a podle jejich vytíženosti se vybírá **Game Server** pro hostování nově vytvořené místnosti [2] (viz obrázek 5.19). V případě této práce je **Game Server** jen jeden, protože při implementaci mechanik hry to zatím je dostačující.

¹⁹Hlavní server, kam se připojuje hráč jako první

²⁰Server, který spravuje místnosti pro hráče. Může být na stejném zařízení jako **Master Server** a může jich být více.



Obrázek 5.19: Schéma klienta, Master Server a Game Server
 Zdroj: <https://doc.photonengine.com/docs/img/LoadbalancingFlowchart.png>

MinigamesServerPlugin

Pro každou nově vytvořenou místnost se inicializuje `MinigamesServerPlugin`, tj. jedna místnost = jedna instance třídy `MinigamesServerPlugin`. Tato třída implementuje metody pro zpracování událostí, které nastanou v místnosti, jako například:

- Vytvoření místnosti – metoda `OnCreateGame`
- Připojení hráče – metoda `OnJoin`
- Odpojení hráče – metoda `OnLeave`
- Vlastní události – metoda `OnRaiseEvent`

NetworkPlayer a CustomProperties

Třída `NetworkPlayer` popisuje klienta na straně serveru. Uchovává jeho stav, například zda klient načel scénou minihry, tak atribut `loaded` je nastaven na `true`. Důležitý atribut je `customProperties` typu `CustomProperties`, jejíž potomci (`ShootingRangeCustomProperties` a `TowersCustomProperties`) uchovávají atributy a stav vážící se ke konkrétní minihře.

GameManager a minihry

Jádrem režie místnosti je třída `GameManager`. Třída logicky vyhodnocuje zprávy od klienta. Při připojení nového hráče vytvoří novou instanci `NetworkPlayer`. Předává instanci třídy `WaitingRoom` informace o akcích hráčů při čekání na zahájení minihry, ta zase uchovává jejich stav a povoluje zahájení minihry v případě připravenosti všech hráčů. Na začátku minihry se vytvoří instance třídy pro aktuální minihru – `Shooting Range` (viz 5.9) nebo `Towers` (viz 5.9). V průběhu minihry jsou předávány této instance zprávy závislé na minihře. Po ukončení minihry je volána metoda `OnMinigameEnded`. Hráčům jsou poslány výsledky zprávou `MINIGAME_END_TO_CLIENT`. *`Shooting Range` Tato třída reprezentuje minihru „`Shooting Range`“ (viz sekce 5.8) na serveru. Po inicializaci je spuštěn časovač, který po vypšení ukončí minihru, a časovač `spawnTargetTimer`, který každé 1.2s náhodně vybere pozici příštího terče a klientovi posílá pokyn `SHOOTING_RANGE_SPAWN_TARGET_TO_CLIENT` pro vytvoření terče ve scéně.

Třída je příjemcem zprávy `SHOOTING_RANGE_ON_HIT_TO_SERVER`, která hlásí zásah terče hráčem. Následně je upraven atribut, který uchovává počet zásahů, v `ShootinRangeCustomProperties` daného hráče.

*`Towers` Tato třída reprezentuje minihru „`Towers`“ (viz sekce 5.8) na serveru. Třída přijímá zprávu `TOWERS_BRICK_DESTROYED_TO_SERVER` za každou zničenou krychli. `TowersCustomProperties` uchovává stav zničených krychlí. V případě, že hráč zničil více jak 350 krychlí pevnosti, tak je minihra ukončena.

Hostování serveru na AWS Cloud

Posledním krokem k plně funkční hře bylo umístění na veřejně přístupný server, na který se mohou klienti připojit. Zvolil jsem AWS Cloud²¹ od společnosti Amazon.com, Inc. kvůli dostupnosti. Po zakoupení služby bylo potřeba ještě server správně konfigurovat. Řídil jsem se návodem²² Museli se otevřít všechny potřebné porty pro síťovou komunikaci.

Všechny soubory a spustitelný program se přesunou na virtuální stroj. V konfiguračním XML souboru na straně serveru bylo potřeba nastavit správné IP adresy pro připojení z `Master Server` na `Game Server`. V poslední řadě je potřeba na straně klienta v souboru `Photon Settings Server` (viz obrázek 5.5) zadat IP adresu, na kterou se bude klient připojovat.

²¹AWS Cloud, dostupné z <https://aws.amazon.com/>

²²Návod pro hostování serveru na AWS, dostupné z https://doc.photonengine.com/en-us/server/current/reference/faq#how_to_host_photon_server_on_amazon_

Kapitola 6

Testování

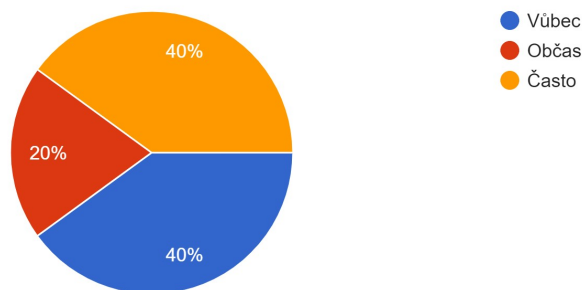
Bylo provedeno uživatelské testování aplikace. Aplikaci vyzkoušelo 5 účastníků.

6.1 Demografický dotazník

Před samotným testováním účastníci vyplnili demografický dotazník¹ za účelem získání informací o účastnících. Předpokládalo se, že skupina účastníků bude různorodá a to se i potvrdilo. Dotazy zaměřeny na zkušenosti s hrami 6.1 a virtuální realitou 6.2 odhalily spojitosti, jak určité typy uživatelů hodnotily jednotlivé minihry (viz obrázek 6.4 a obrázek 6.6).

Hrajete hry?

5 odpovědí

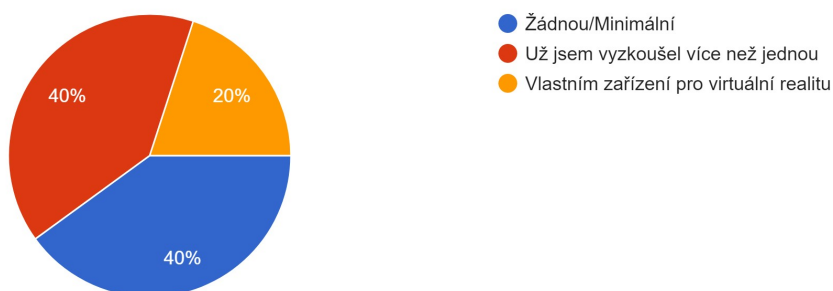


Obrázek 6.1: Otázka zaměřená zkušenost s hrami.

¹Kompletní výsledky demografického dotazníku se nachází na přiloženém paměťovém médiu.

Jakou máte zkušenost s používáním virtuální reality?

5 odpovědí



Obrázek 6.2: Otázka zaměřená zkušenost s virtuální realitou.

6.2 Průběh testování

Účastníci vytvořili místnost nebo se připojili do místnosti. První se testovala Shooting Range (viz sekce 5.8), poté minihra Towers (viz sekce 5.8). Během testování byly zaznamenány připomínky a projeví se i technické nedostatky implementace, které jsou popsány v sekci 6.5. Po odehrání každé minihry účastník vyplnil obecný dotazník pro hodnocení softwarového produktu 6.3.. Po testování účastníci vyplnili i dotazník, který se kladl důraz na konkrétní mechaniky a situace v minihrách (viz sekce 6.4).

6.3 Dotazník měřící uživatelskou zkušenost (UEQ)

Dotazník UEQ² se skládá z 26 dvojic protikladů a dotazovaný volí hodnotu na škále 1 až 7. Hodnotí se 6 aspektů – Atraktivita, Přehlednost, Účinnost, Spolehlivost, Stimulace a Originalita, jejichž výsledek nabývá hodnot v rozsahu –3 až 3.

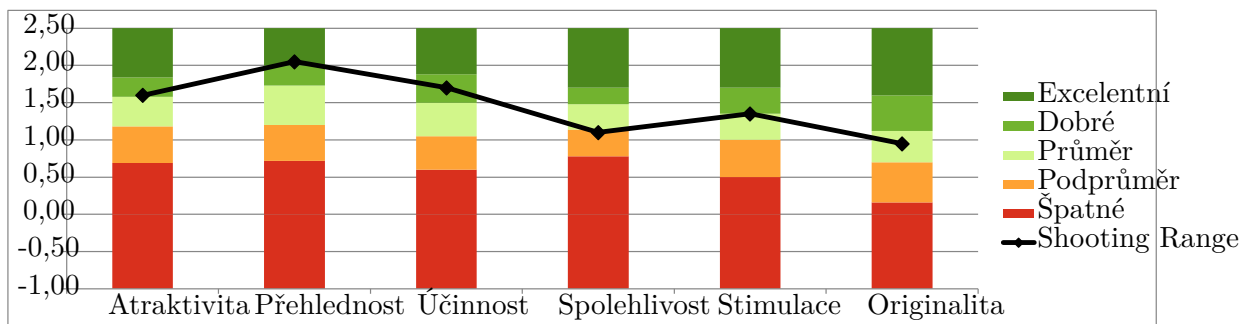
Účastník vyplnil jeden dotazník pro každou minihru. Součástí UEQ jsou i analytické nástroje, které pomáhají získaná data zpracovat. Byly získány následující informace.

- Výsledek hodnocení a porovnání zmíněných aspektů s 468 produkty (například software, webové stránky, sociální sítě)
- Souvislosti s demografií účastníků – hledání zajímavých souvislostí mezi hodnocením a demografií účastníků

Hodnocení Shooting Range

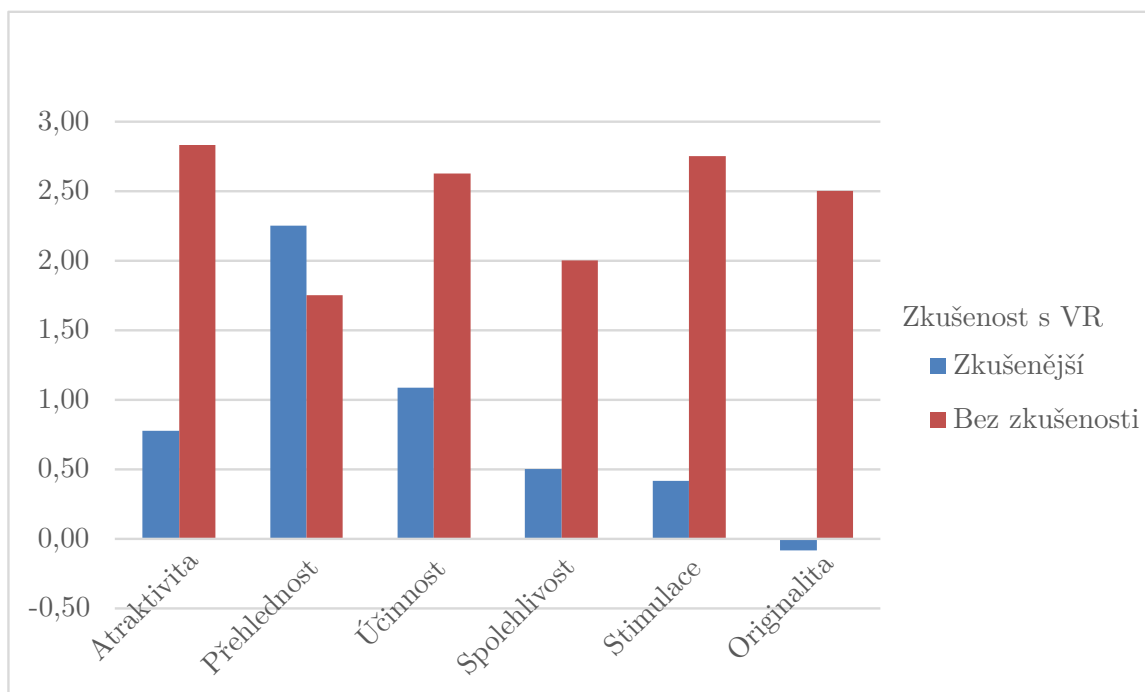
V porovnání s ostatními produkty vedl v Přehlednosti se umístil mezi 10%, zatímco Spolehlivost dosáhla podprůměrného hodnocení (viz obrázek 6.3).

²User Experience Questionnaire – volně dostupný dotazník pro uživatelské hodnocení software, dostupný z: <https://www.ueq-online.org/>



Obrázek 6.3: Výsledek hodnocení minihry Shooting Range v porovnání s ostatními produkty

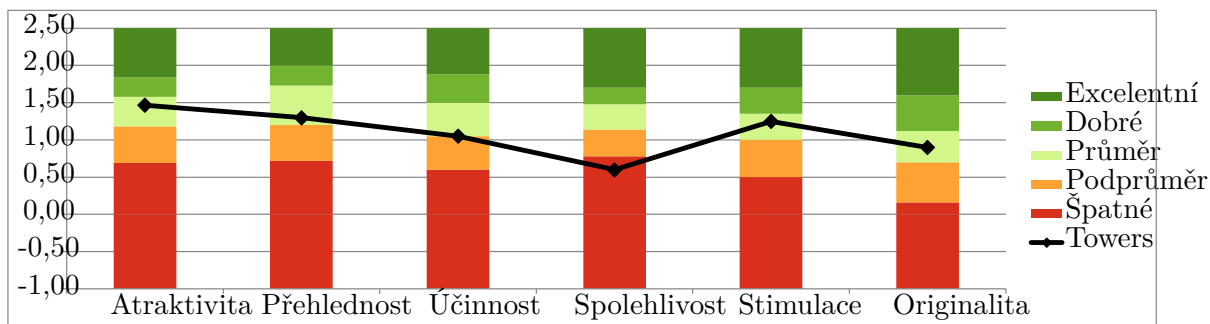
Testování se účastnili dva účastníci, kteří vyzkoušeli virtuální realitu poprvé a jejich odpovědi se oproti ostatním třem účastníkům v některých ohledech liší. Všechny aspekty kromě Přehlednosti byly hodnoceny významně kladněji (viz 6.4).



Obrázek 6.4: Porovnání hodnocení hráčů na základě zkušenosti s VR

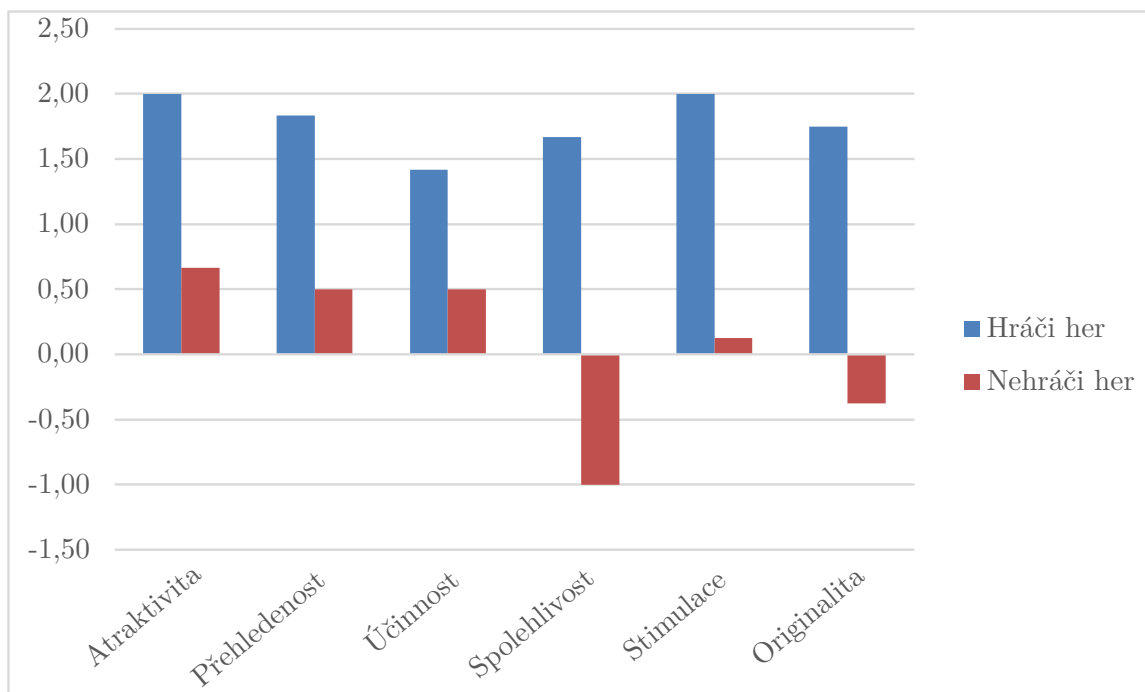
Hodnocení Towers

Minihra Towers už tak úspěšná nebyla. Aspekty dosáhly maximálně průměrného hodnocení v porovnání s ostatními produkty. A dokonce spolehlivost se ani nezařadila do podprůměru (viz obrázek 6.5).



Obrázek 6.5: Výsledek hodnocení minihry Towers v porovnání s ostatními produkty

Účastníci, kteří nehrají hry, tuto minihru nehodnotili tak kladně jako hráči her (viz 6.6). Například jeden z těchto účastníků měl opravdu problém pochopit, jak efektivně vystřelit z luku.



Obrázek 6.6: Porovnání hodnocení minihry Towers mezi hráči a nehráči her

6.4 Vlastní dotazník a poznámky

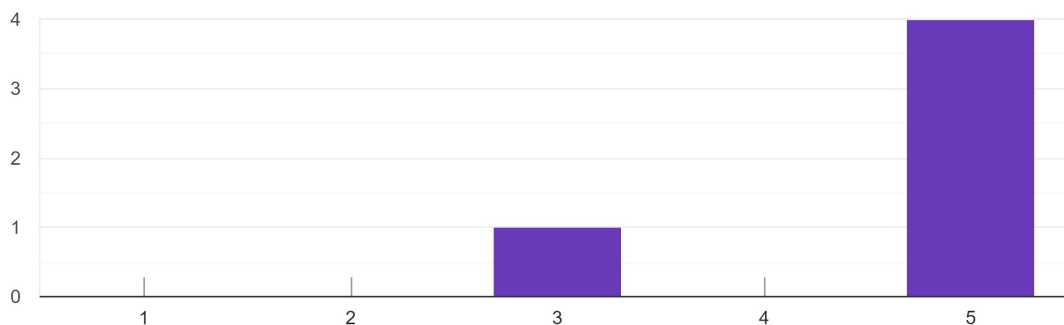
Po odehrání miniher účastníci vyplnili dotazník, který se zaměřil na konkrétní aspekty miniher³. V následujících sekcích se nachází některé dotazy a odpovědi (Na ose Y je počet účastníků, co volilo danou odpověď. Osa X popsána vždy v popisku).

³Kompletní výsledky vlastního dotazníku po odehrání se nachází na přiloženém paměťovém médiu.

Shooting Range

Téměř všichni účastníci se shodli, že nebylo důležité sledovat druhého hráče při hraní minihry (viz obrázek 6.7).

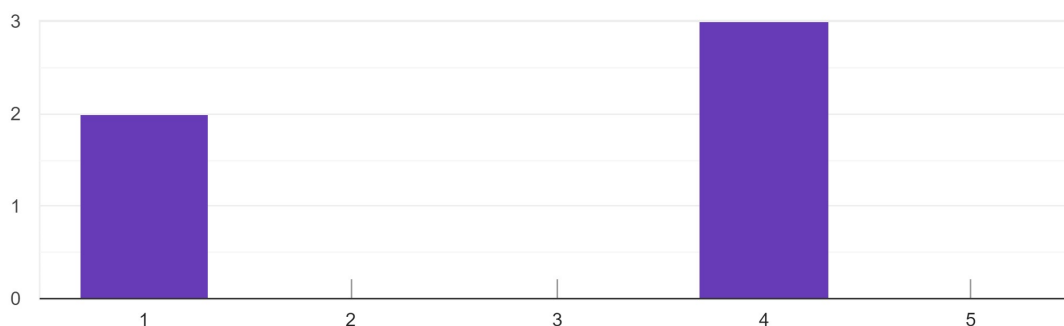
Sledovat a vnímat druhého hráče bylo pro výhru důležité
5 odpovědí



Obrázek 6.7: Graf odpovědí ohledně vnímání přítomnosti druhého hráče v minihře. (1 – Souhlasím, 5 – Nesouhlasím)

Naopak účastníci se rozdělili na (necelé) poloviny v dotahu ohledně orientace pomocí zvukových efektů banditů (viz obrázek 6.8).

Zvukové efekty banditů mi pomáhaly se prostorově orientovat
5 odpovědí



Obrázek 6.8: Graf odpovědí ohledně orientace pomocí zvukových efektů. (1 – Souhlasím, 5 – Nesouhlasím)

Poznámky účastníků:

- „chybí vizuální efekt střelby, aby bylo poznat, kam hráč střílil“
- „krabice pro sebrání revolveru bránily výhledu“

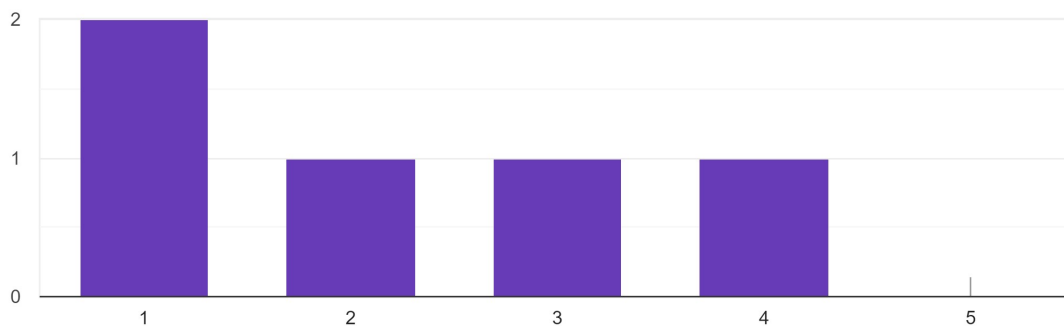
- „nebylo možné rozpoznat, kdo z hráčů trefil banditu“

Towers

Účastníci většinou neměli problém zasahovat pevnost. Avšak pro jednoho účastníka bylo náročné na pevnost dostřelit.

Jak hodnotíte obtížnost trefit se do protivníkovra hradu?

5 odpovědí

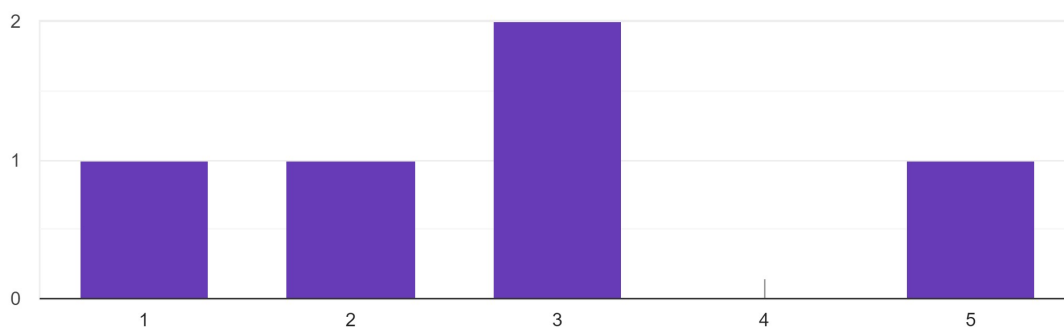


Obrázek 6.9: Graf odpovědí ohledně obtížnosti zásahu pevnosti. (1 – Lehká, 5 – Těžká)

Nejvíce nerozhodní byli účastníci u dotazu, zda střelba z luku odpovídala realitě. Účastník, který měl největší zkušenost se střelbou z luku, hodnotil, že střelba z luku odpovídala realitě. Naopak zmíněný účastník, který měl se střelou problém, s tímto tvrzením nesouhlasil.

Střelba z luku odpovídala realitě

5 odpovědí



Obrázek 6.10: Graf odpovědí ohledně realističnosti luku. (1 – Souhlasím, 5 – Nesouhlasím)

Jeden účastník měl v době testování pomalé internetové připojení. Pevnost se nevytvořila ihned, ale kvůli opožděnému přenosu zpráv šlo vidět postupnou inicializaci krychlí pevnosti, což tvořilo efekt stavby.

Poznámky účastníků:

- „bylo by lepší více vizuálně natáhnout šíp“

6.5 Technické chyby

Při testování se projeví i technické nedostatky implementace.

- Vytvoření revolveru – někdy nefunguje vytvoření revolveru.
 - Příčina – pravděpodobně špatná destrukce objektu `XRInteractionToolkit`.
 - Řešení – zatím neznámé.
- Střelba z luku – nejde vystřelit z luku, pokud se předtím hrála minihra Shooting Range.
 - Příčina – pravděpodobně špatná destrukce `XRInteractionToolkit` (pravděpodobně souvisí s předchozím problémem).
 - Řešení – zatím neznámé.
- Natáhnutí šípu u Oculus Quest 1 – nejde natáhnout šíp při použití zařízení Oculus Quest 1.
 - Příčina – zatím neznámá.
 - Řešení – zatím neznámé. Problém je však možné obejít, pokud se šíp natáhne pod určitým úhlem a pod úrovní pasu.
- Opětovné zničení krychlí – hráč s pomalým síťovým připojením zničil krychle pevnosti. Na server se odeslala zpráva o zničení krychlí. Avšak krychle se kvůli pomalému připojení nestihly zničit ve scéně. Hráč mohl střílet do těchto krychlí znovu a na server se odeslaly zprávy za již zničené krychle.
 - Příčina – na serveru se uchovává počet zničených krychlí hráčem.
 - Řešení – při vytvoření krychle poslat `viewID` na server. Uchováním tohoto atributu bude lépe možné kontrolovat, zda byla krychle již zničena.
- Výhra obou hráčů – oba hráči se blížili k vítězství a každý vystřelil svůj šíp naráz. Oba se objevili v tabulce vítězů.
 - Příčina – špatné ošetření stavu, kdy minihra již byla ukončena.
 - Řešení – správně ošetřit stav po ukončení minihry.

Kapitola 7

Závěr

Cílem práce bylo navrhnout a implementovat balíček miniher pro více hráčů ve virtuální realitě. U návrhu miniher byly rozvedeny klíčové prvky s ohledem na případná rizika. Návrh obsahoval celkem pět miniher a z důvodu vysoké časové náročnosti jsem se soustředil na implementaci dvou vybraných – Shooting Range a Towers. V Shooting Range hráči soutěží v počtu zásahů banditů na westernu. V Towers hráč pomocí luku ničí protivníkovu pevnost. I tak ale vznikla hra, na které lze stavět nové minihry a postupně ji rozšiřovat. Součástí implementace je také serverová část pro vlastní herní logiku na straně serveru. Uživatelským testováním výsledné aplikace jsem zhodnotil silné a slabé stránky obou miniher. Minihra Shooting Range byla dynamičtější, ale chyběla tam větší interakce s druhým hráčem, což by mohla vylepšit implementace speciálních událostí. V Towers hráči soutěžili více na přímo, ale po krátkém čase bylo jednoduché zasáhnout protivníkovu pevnost. Byly zaznamenány technické vady, které se projevily při testování. Nejzávažnější byly situace, kdy hráč nemohl vzít z krabice revolver. Naopak výhra obou hráčů pobavila a dala by se brát jako remíza.

Z hlediska implementace se herní engine Unity a knihovna XR Interaction Toolkit se osvědčily jako skvělé nástroje pro tvorbu zážitků ve virtuální realitě. Photon PUN hodnotím kladně kvůli poměrně rychlé možnosti implementace síťového řešení, avšak jedná se o uzavřený komerční systém, a proto bych příště volil spíše open source řešení. V případě serveru použití Photon Server vyžadovalo někdy hluboké bádání kvůli informacím, které chyběly v technické dokumentaci knihovny.

V dalším vývoji hry je nutné rozšířit stávající minihry, opravit chyby a přidání nových miniher na základě původního návrhu. Dalším důležitým krokem je implementace plně autoritativního serveru (například kvůli vyšší bezpečnosti). Pro přechod do produkční fáze je potřeba integrovat Oculus přihlášení a větší možnosti nastavení aplikace.

Literatura

- [1] DAVIS, B. a SUMMERS, M. Applying Dale's Cone of Experience to increase learning and retention: A study of student learning in a foundational leadership course. *QScience Proceedings*. Červen 2015, sv. 2015, s. 6. DOI: 10.5339/qproc.2015.wcee2014.6.
- [2] EXIT GAMES INC.. *LoadBalancing Application*. Dostupné z: <https://doc.photonengine.com/en-us/server/current/applications/loadbalancing/application>.
- [3] EXIT GAMES INC.. *Photon Documentation: RPCs and RaiseEvent*. Dostupné z: <https://doc.photonengine.com/en-us/pun/current/gameplay/rpcsandraiseevent>.
- [4] EXIT GAMES INC.. *Photon Documentation: Ownership Control*. Dostupné z: <https://doc.photonengine.com/en-us/pun/current/gameplay/ownershipandcontrol>.
- [5] EXIT GAMES INC.. *Photon Documentation: Instantiation*. Dostupné z: <https://doc.photonengine.com/en-us/pun/current/gameplay/instantiation>.
- [6] FACEBOOK TECHNOLOGIES, LLC. *App Upload and Distribution with the Oculus Store*. Dostupné z: <https://developer.oculus.com/resources/publish-intro/>.
- [7] FACEBOOK TECHNOLOGIES, LLC. *VR Locomotion Design Guide*. Dostupné z: <https://developer.oculus.com/resources/bp-locomotion/>.
- [8] FORTUNE BUSINESS INSIGHTS. *Virtual Reality (VR) in Gaming Market Size, Share COVID-19 Impact Analysis, By Component (Hardware, Software, and Content), By Device (Mobile, Console/PC, and Standalone), and Regional Forecast, 2021-2028* [online]. Fortune Business Insights, září 2021. Dostupné z: <https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-gaming-market-100271>.
- [9] JERALD, J. The VR Book: Human-Centered Design for Virtual Reality. In: Morgan Claypool, 2016, kap. What Is Virtual Reality?, s. 9–13. ISBN 978-1-97000-115-0.
- [10] JERALD, J. The VR Book: Human-Centered Design for Virtual Reality. In: Morgan Claypool, 2016, kap. History of VR, s. 15–27. ISBN 978-1-97000-115-0.
- [11] KHAROUB, H., LATAIFEH, M. a AHMED, N. 3D User Interface Design and Usability for Immersive VR. *Applied Sciences*. 2019, sv. 9, č. 22. DOI: 10.3390/app9224861. ISSN 2076-3417. Dostupné z: <https://www.mdpi.com/2076-3417/9/22/4861>.
- [12] MYSTAKIDIS, S. Metaverse. *Encyclopedia*. Únor 2022, sv. 2, s. 486–497. DOI: 10.3390/encyclopedia2010031.

- [13] OCULUS VR. *How a developer can reduce motion sickness in VR games* [online]. Oculus VR, únor 2021. Dostupné z: <https://developer.oculus.com/blog/introducing-app-lab-a-new-way-to-distribute-oculus-quest-apps/>.
- [14] PEDDIE, J. *Augmented Reality: Where We Will All Live*. 1st ed. 2017. Springer, 2017. ISBN 978-3-319-54501-1.
- [15] SKARREDGHOST. *How a developer can reduce motion sickness in VR games* [online]. Skarredghost, listopad 2016. Dostupné z: <https://skarredghost.com/2016/11/28/how-a-developer-can-reduce-motion-sickness/>.
- [16] UNITY SOFTWARE INC.. *Automotive, Transportation Manufacturing* [online]. Dostupné z: <https://unity.com/solutions/automotive-transportation-manufacturing>.
- [17] UNITY SOFTWARE INC.. *Quick guide to the Unity Asset Store* [online]. Dostupné z: <https://unity3d.com/quick-guide-to-unity-asset-store>.
- [18] UNITY SOFTWARE INC.. *Unity Manual: Physics*. Dostupné z: <https://docs.unity3d.com/Manual/PhysicsSection.html>.
- [19] UNITY SOFTWARE INC.. *Unity Manual: GameObjects*. Dostupné z: <https://docs.unity3d.com/Manual/GameObjects.html>.
- [20] UNITY SOFTWARE INC.. *Unity Manual: Creating and Using Scripts*. Dostupné z: <https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>.
- [21] UNITY SOFTWARE INC.. *Unity Manual: MonoBehaviour*. Dostupné z: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>.
- [22] UNITY SOFTWARE INC.. *XR Interaction Toolkit*. Dostupné z: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.9/manual/index.html>.
- [23] VOLGA AKSOY, A. W. *How does Oculus Link Work? The Architecture, Pipeline and AADT Explained* [online]. Oculus VR, listopad 2019. Dostupné z: https://developer.oculus.com/blog/how-does-oculus-link-work-the-architecture-pipeline-and-aadt-explained/?locale=en_US.