VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

Fakulta informačních technologií
Faculty of Information Technology

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

Brno, 2017 Bc. Miroslav Novotný

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

VR INTERAKTIVNÍ APLIKACE
VR INTERACTIVE APPLICATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Miroslav Novotný**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Tomáš Milet**
SUPERVISOR

BRNO 2017

Abstrakt

Cílem této práce je nastudovat virtuální realitu a aspekty vývoje pro tuto platformu. Dále také vybrat a nastudovat vhodné prostředí pro vývoj aplikace a zjistit a vyzkoušet možnosti a omezení, které s touto platformou a prostředím souvisí. Závěrem pak navrhnout aplikaci s využitím nabytých znalostí. Navrženou aplikaci posléze implementovat a otestovat.

Abstract

The goal of this thesis is to learn about virtual reality and the aspects of the development for this platform. The next goal is to find well suited development environment for the application and test the capabilities and limits of them both. The thesis will be completed by designing, implementing and then testing the aplication using knowledge learnt.

Klíčová slova

Virtuální realita, Unity, Nevolnost z virtuální reality, Renderování, HTC Vive, Pohyb po scéně

Keywords

Virtual reality, Unity, Virtual reality sickness, rendering, HTC Vive, In-scene movement

Citace

Novotný Miroslav: VR Interaktivní aplikace, diplomová práce, Brno, FIT VUT v Brně, 2017

VR Interaktivní aplikace

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Tomáše Mileta, a že jsem uvedl všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Miroslav Novotný
24. 5. 2017

Poděkování

Děkuji panu Ing. Tomáši Miletovi za odbornou pomoc při řešení diplomové práce.

© Miroslav Novotný, 2017
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah....1
1 Úvod....3
2 Unity....4
2.1 Základy enginu....4
2.2 Rendering....4
2.3 Skriptování....5
2.4 Fyzika....5
2.5 Animace....6
2.6 Unity pro VR....7
2.7 Unity Editor....7
2.8 Asset store, standard assets....8
3 Virtuální realita....9
3.1 Přístupy....9
3.2 Interakce s virtuální realitou a její problémy....10
3.3 Potenciál a využití virtuální reality....13
3.4 Člověk ve virtuální realitě....14
4 Návrh aplikace....18
4.1 Platforma....18
4.2 Engine....18
4.3 Přirozenost ovládání....19
4.4 Obecné problémy a řešení....19
4.5 Cíle vývoje....19
4.6 Aspekty aplikace....19
4.7 Možnosti pohybu po scéně....20
5 Implementace....22
5.1 Obecný návrh....22
5.2 Důležité třídy a rozhraní....22
5.3 Důležité aspekty vývoje pro VR....23
5.4 Finální scény....23
5.5 Možnosti rozšíření....23
5.6 Využití jako základ....24
5.7 Implementované techniky cestování....24
6 Interaktivní objekty a testování....26
6.1 Cíle testů....26

6.2 Prostředky a postup testování....26
6.3 Testování VR mimo hlavní aplikaci....27
6.4 Interaktivní objekty a mechaniky....28
6.5 Závěry testování....33
7 Závěr....34

1 Úvod

Virtuální realita (VR) je oblast s velkým potenciálem a v posledních letech dosahuje značného růstu jednak v oblasti samotného zájmu lidí, ale také v množství prostředků do ní investovaných [1]. V této práci představím, co se vlastně skrývá pod pojmem virtuální realita. Dále nastíním možnosti vývoje pro tuto platformu, případné problémy a jejich vliv. Cílem této práce je také nastudovat vhodné prostředí pro vývoj aplikace využívající virtuální reality a v závěru implementace takovéto aplikace a to konkrétně pro htc Vive. Jako prostředí jsem si zvolil herní engine Unity, který je, jak později detailněji rozeberu, velmi vhodným nástrojem pro vývoj takovéto aplikace. Jedním z klíčových zátarasů v aplikacích ve VR je pak volný pohyb po scéně, kterému budu věnovat větší úsilí než ostatním typům interakce. Výsledný produkt by pak měl ležet někde na pomezí testování a implementace hry. Tento fakt bude také reprezentován rozdělením aplikace do několika částí.
Návrh jak testovacích tak herních částí pak bude ležet částečně na teoreticky zjištěných poznatcích, vyčtených z internetových zdrojů nebo knih, tak na vlastních zkušenostech nabytých při předchozím testováním. Součástí práce jsou pak i výsledky testování aplikací, které byly vyvinuty jako součásti jiných projektů, ale závěry z nich zjištěné byly pro tuto práci velmi zajímavé.
Na konec by součástí aplikace měla být minimálně jedna scéna, kde se bude uživatel snažit dosáhnout nějakého cíle a, nehledě na testování se snažit, využít dostupných prostředků k jeho dosažení.

2 Unity

Ilustrace 1: Unity

Unity je herní engine vyvinut firmou *Unity Technologies* [2]. Mezi jeho hlavní přednosti patří například multiplatformnost (z unity je v současné době možné vytvářet aplikace pro více než 20 platforem). Unity zároveň poskytuje vysokou míru optimalizace pro každou z podporovaných platforem. Engine dále obsahuje kompletní renderovací engine. Mimo podporu pro physically based shading také, kromě tradičního osvětlovacího modelu, podporuje Deferred Shading. Krom již hotových assetů unity umožňuje tvorbu a import vlastních, do toho je také zahrnuta podpora grafických (2D i 3D) i zvukových formátů a vlastní scriptování v C# nebo unityScript (podobné javascriptu). Další velmi pozitivní vlastností je silná a částečně nativní podpora systémů pro virtuální a rozšířenou realitu. Unity je nyní rozšířený a poměrně používaný engine, který se může pochlubit aktivní komunitou a s tím spojeným obchodem s assety. To je podporováno kvalitními možnostmi pro tvorbu knihoven třetí strany.

2.1 Základy enginu

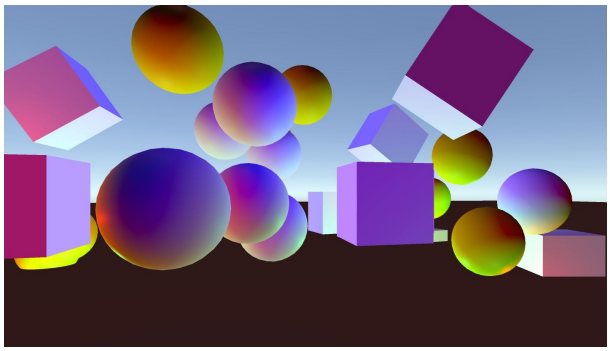
Zde jsou některé kategorie tak, jak jsou uvedené v dokumentaci [3]. Nejdůležitějším typem objektu v Unity je *GameObject*. Každý objekt v projektu je GameObject. Samotný nemá žádné vlastnosti, tvoří tedy kontejner pro vlastnosti. Taková vlastnost se v Unity nazývá *Component* (Komponenta). GameObject vždy obsahuje komponentu transform, která reprezentuje jeho pozici, rotaci a velikost. Engine přichází se značným množstvím již hotových komponent, jako například světlo nebo kamera a samozřejmě umožňuje tvorbu vlastních komponent pomocí skriptování. GameObject také obsahuje vlastnosti *Tag* a *Layer*, neboli značka a vrstva. Značka je často používaná k efektivnímu vyhledávání a srovnávání objektů. Vrstva je většinou používaná na maskování různých operací, typicky světel, kamery nebo kolizí. V rámci projektu je možné tvořit takzvané *Prefaby*. Tyto objekty slouží jako mimo scénu uložené GameObjecty, ze kterých se dá v případě potřeby instanciovat. Nově vytvořený GameObject se stává klonem prefabu a přejímá všechny jeho vlastnosti.

2.2 Rendering

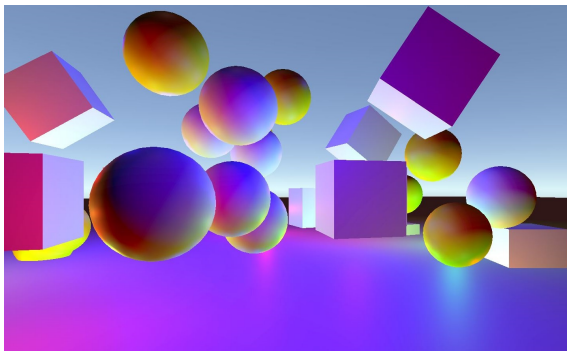
Moderní herní enginy (UE, Unity, CryEngine) řeší rendering primárně dvěma způsoby, pomocí forward renderingu (Unity, UE3, volitelné v UE4) nebo deferred renderingu (UE4, CryEngine, volitelné v Unity) [4].
Dopředné renderování pracuje na principu rasterování každého objektu ve scéně. Během procesu stínování dochází pro každé světlo k výpočtu, jak má být daný objekt osvětlen. Každý objekt musí být tedy zvážen vůči každému světlu. Na tento postup existuje množství optimalizací snižující počet výpočtů, například výpočty viditelnosti a vynechávání případů, kdy světlo nenasvětluje objekt. Často se také používá metoda omezení počtu světel, které mohou najednou nasvětlovat jeden grafický objekt. Velký počet světel v tomto systému způsobí velký dopad na rychlost výpočtu dynamického osvětlení.

Deferred Shading na druhou stranu funguje tak, že nejprve rasterizuje celou scénu bez světel a zároveň vytvoří sérii 2D bufferů, do kterých uloží geometrické informace, nutné k pozdějšímu výpočtu osvětlení. Mezi tyto informace patří například hloubka z pohledu kamery, povrchové normály nebo difúzní a spekulární barvy. Pro každý bod obrazu je pak vypočten světelný model za použití uložených informací. Výhoda tohoto přístupu je hlavně v mnohem nižší přidané náročnosti s každým dalším světelným zdrojem [5]. Tato vlastnost umožňuje například použití emisivních textur, které opravdu nasvětlují scénu, bez zásadního dopadu na výkonnost. S tímto přístupem je možné použít i o řád více světel, než u dopředného renderování. Tento typ renderingu s sebou ale nese problémy, například s průhlednými objekty, nebo s různými světelnými modely pro různá světla a objekty. Existují ale postupy, které obcházejí tyto omezení, například vykreslení průhledných objektů dopředným renderováním [6].

Ilustrace 2: Forward v Unity



Ilustrace 3: Deffered v Unity



2.3 Skriptování

Důležitou součástí herního enginu je prostředek, který umožňuje vývojáři tvořit prvky, jenž se nedají snadno vyrobit pomocí základních nástrojů. *Script* (Skript) někdy taky *Behaviour* (Chování) funguje jako vývojářem definovaná komponenta. V rámci hry obvykle slouží ke zpracovávání vstupů hráče a dalších herních událostí, řízení efektů nebo i chování entit a mnoho, mnoho dalšího.
Skriptování v Unity funguje jako klasické imperativní programování v jazyce C# (nebo unityScriptu), k enginu se zde přistupuje pomocí importování jeho částí jako knihoven. Většina skriptů tedy začíná řádkem "using UnityEngine;" popřípadě ekvivalentem.
Unity obsahuje prostředek *Event System* (Systém událostí), který zajišťuje adekvátní zasílání událostí, jako například vstupy, do skriptu.

2.4 Fyzika

Unity engine obsahuje komponenty, které zařizují fyzikální simulaci. Snaha je, aby vývojář v Unity pouze parametrizoval systém a tím docílil realistického chování objektů. Unity obsahuje dva separátní fyzikální enginy pro 2D a 3D simulaci, tyto systémy jsou téměř identické, 2D verze má pouze odstraněnou jednu dimenzi.
Hlavní komponentou fyzikálního chování v Unity je *Rigidbody* (Zde je možné použít název tuhé těleso nebo přejmout anglický termín). Od momentu, kdy je tato komponenta přidána k objektu typu GameObject, převezme řízení co se týče pohybu objektu. Rigidbody do sebe zahrnuje fyzikální vlastnosti jako jsou rychlost, zrychlení, hmotnost, síla anebo rotační pohyby. Nehybný objekt je automaticky uspán, aby byl ušetřen simulační výkon. Tato komponenta je také nutná, aby mohl v rámci skriptování vývojář zpracovávat kolize objektu. V případě, že je potřeba pouze zpracovávat

kolize, je v Rigidbody možné nastavit hodnotu *isKinematic* (objekt není dynamický, ale kinematický), Unity pak vynechá simulaci pohybu a zpracovává pouze kolize.
Samotný kolizní systém, který je součástí fyzikálního enginu, závisí na komponentách typu *Collider*. Tyto komponenty definují tvar, který často z optimalizačních důvodů nemusí plně odpovídat tvaru objektu. Unity nabízí několik základních tvarů, pro které má engine optimalizovanou detekci a ty jsou *Box Collider* (Kvádr), *Sphere Collider* (Koule) a *Capsule Collider* (Kapsle). Jako netriviální typ je zde *Mesh Collider*, jehož tvar závisí na geometrii vybraného modelu. Posledně zmíněný je však zdaleka výpočetně nejnáročnější, tento typ musí být konvexní. Unity také nabízí možnost automatického výpočtu konvexní obálky z nekonvexního modelu. Ta často poměrně kvalitně aproximuje tvar objektu. Automatický přístup má samozřejmě problémy s nevhodnými tvary. Součástí kolizních komponent také mohou být fyzikální materiály, v nichž se nastavují vlastnosti povrchu objektu, jako například tření a odrazivost. Collider může být také nastaven do módu Trigger (Spínač, Spouštěč), ve kterém se jeho kolize nepřenáší do fyzikální simulace, pouze skript objektu dostane informaci o tom, že kolize nastala. Tento typ je optimalizovaný na co nejrychlejší detekci a nepočítá přesné kolizní body, rychlosti srážek a další fyzikální vlastnosti kolize.
Dalším typem fyzikálních komponent v Unity je typ *Joint* (Kloub), jeho využitím je možné propojit dva různé objekty typu GameObject mající komponentu Rigidbody. Klouby fungují jako efektory a působí silou na připojené objekty, například *Spring* (Pružina) nebo *Hinge* (Záves, Pant) s různými vlastnostmi. Unity obsahuje také plně nastavitelný *Configurable Joint* (Konfigurovatelný kloub), který nabízi velké množství nastavení co se týče omezení pohybu a působení sil.
Poslední hlavní částí fyzikální simulace je *Character Controller* (Kontroler/Ovladač Postav), tato komponenta řeší problém, že často není vhodné, aby se hráčem ovládaná postava chovala přesně dle fyzikálních zákonů. Například příliš vysoká hybnost omezuje responzibilitu ovládání. Tento systém poskytuje základní prostředky pro vytvoření hráčem ovládané postavy reagující ve fyzikálním světe. Nedovoluje projít zdí a podobně, dále také usnadňuje manipulaci s hráčem z hlediska skriptu.

2.5 Animace

Animační systém v Unity se nazývá *Mecanim* a mezi jeho hlavní účely patří správa a řízení animací. Mezi jeho hlavní funkcionality patří například *Animation Retargeting*, neboli možnost použití stejných animací na různé objekty se stejnou kostrou, dále plná kontrola animačních vah za běhu programu nebo tvorba událostí následkem přehrávání animace. Engine také obsahuje takzvané *Blend Shapes*, což je animační technika, ve které se ukládají deformované části meshe, vhodná například pro animace mimických svalů a podobně. Další s velkých částí tohoto systému je hierarchický stavový automat. Engine obsahuje navíc ještě další nástroje pro správu a tvorbu animací.

V Unity existuje komponenta *Animator* a ta v sobě zaštiťuje administraci výše uvedených. Vývoj v této komponentě je umožněn pomocí vizuálního programovacího nástroje, jde o speciální, rozšířený typ konečného automatu. Animátor řídí animace v rámci jednoho GameObjectu, může obsahovat několik vrstev, která každá v základu funguje jako samostatný konečný automat. Přechody mezi stavy vrstvy mohou být buď na základě změny parametrů nebo časované. Například přechod do stavu *běh* ze stavu *nehybný,* může nastat, pokud je rychlost větší než nulová.
Právě tyto proměnné jsou pak hlavním, ale zdaleka ne jediným, prostředkem pro řízení animace ze skriptu. Ten jako takový má přístup k většině vlastností Animátoru a může je přímo ovlivňovat nebo modifikovat, doporučený je ovšem spíše přístup přes parametry pro snazší udržení konzistence animace vůči světu. Také je tu možnost úplně vynechat celý automat a rovnou řídit animaci ze skriptu, tento způsob umožňuje často snazší spolupráci s nějakou netriviální mechanikou, vytvořenou v onom skriptu, ale může způsobit problémy, pokud bychom jej chtěli kombinovat s vestavěnými funkcemi animátoru.

2.6 Unity pro VR

Unity je nyní velmi vhodným enginem pro tvorbu VR aplikací. Engine obsahuje vestavěnou podporu pro nejrozšířenější VR headsety a je pravděpodobné, že počet VR platforem, které Unity podporuje se bude v blízké době rozšiřovat. Nynější podporované platformy jsou Oculus, OpenVR, Playstation VR a Gear VR. HTC Vive, který je využit v rámci tohoto projektu, spadá pod Open VR, na kterém staví knihovna Steam VR. Nativní podpora v sobě zahrnuje automatické nastavení renderovacího cíle do VR headsetu a také příjem základního vstupu ze strany VR, jako například natočení kamery pomocí pohybu hlavy [7].

V mojí aplikaci je použita knihovna SteamVR, která tvoří základní rozhraní pro práci s VR systémy na úrovni zpracování vstupů a výstupů. Pro usnadnění práce s virtuálními objekty je použita knihovna VRTK (zkratka z Virtual Reality Tool kit). Vývoj pak neprobíhá přímo nad SteamVR, protože VRTK nad ním tvoří kvalitní nástavbu [8].

Samotné VRTK je tvořeno z několika částí. Dle oficiálního zdroje je rozděleno na prefaby, skripty a příklady. Zatímco příklady jsou vhodné hlavně k naučení možností této knihovny a prefaby se dají použít v typizovaných případech, skripty obsahují třídy, ze kterých je možno dědit užitečnou funkcionalitu a postavit na nich většinu VR prvků ve hře.
Mezi dobře použitelnými prefaby je například vizualizace herních ovladačů nebo *CameraRig* (základní CameraRig je jako prefab již ve steamVR, VRTK jej různými způsoby rozšiřuje), což je GameObject ve scéně představující kameru i s omezením herní místnosti odpovídající skutečné místnosti. Skriptů je v knihovně několik desítek [9].

2.7 Unity Editor

Jednou z nejsilnějších stránek Unity je mocné vývojové prostředí zvané *Unity Editor*. Tento program nabízí pohodlný a přehledný přístup ke většině aspektů vyvíjeného programu a zároveň udržuje přehledný systém pro organizaci celého projektu. Editor, jako takový, je také vyvíjen v samotném Unity, a proto stejným způsobem, jako probíhá vývoj samotné aplikace, je možné upravovat editor. Unity nabízí rozsáhlé možnosti pro takzvaný *in-editor scripting*, tedy skripty běžící za běhu editoru a nikoliv pouze za běhu programu. Editor se dělí na několik částí, nejdůležitější z nich jsou *Inspector*, který zobrazuje a umožňuje modifikovat vlastnosti vybrané entity (tou může být GameObject nebo importovaný zdrojový soubor, jako zvuk, model nebo skript). Okno *Project* umožňuje procházet soubory celého vyvíjeného projektu a v podstatě funguje jako manažer souborového systému v rozsahu složek projektu. Okna *Scene* a *Hiearchy*, jsou důležitá pro zobrazení, tvorbu a úpravu scény. Scéna je speciální typ souboru v Unity, který tvoří souhrn GameObjectů a specifického nastavení, se kterým se dá jednotně pracovat. V programu obvykle bývá scéna pro hlavní menu nebo pro každý level případné hry. Unity Editor umožňuje editovat najednou právě jednu scénu. Samotné okno Scene pak graficky zobrazuje právě editovanou scénu a okno Hierarchy zobrazuje hierarchii objektů ve scéně. V neposlední řádě je tu Okno *Console*, které funguje často jako primární debugovací nástroj. Do něj je pak ve výchozím nastavení přesměrován chybový i standardní výstup programu. Ještě zmíním okno *Game*, ve kterém běží v případě spuštění právě vyvíjená aplikace. Unity Editor v základu obsahuje přidané textové programovací prostředí *Mono Develop*, editor ale umožňuje vývojáři nastavit preferované vývojářské prostředí.

2.8 Asset store, standard assets

Unity již v základu přichází s velkým množstvím standardních knihoven, mezi tyto knihovny patří kamery, postavy, efekty, prostředí, částicové efekty, vozidla a další. Tyto jsou při vývoji s Unity široce používané. Knihovny jsou velmi vhodné zejména pro rychlé prototypování a testování, kde na místo tvoření vlastních assetů (obecný název pro relativně malou a nezávislou část aplikace), může vývojář použít standardní z Unity a v případě dalšího vývoje je nahradit svými. Některé z assetů se pak můžou hodit po celou dobu vývoje a to například některé grafické efekty.
Součástí servisu Unity je také tak zvaný *Asset store* [10]. Je to obchod s obsahem přímo stažitelným do Unity editoru. V rámci této služby mohou vývojáři nabízet a nebo stahovat obsah, zdarma nebo za určené částky. Asset store dělí stažitelný obsah do několika kategorii a v této době je zde několik desítek tisíc položek počínaje drobnými prvky konče celými projekty.

3 Virtuální realita

Definice termínu virtuální realita vychází z přirozených definic obou slov. Tento pojem je poté značně široký, ale většinou bývá interpretován jako specifický typ emulované reality.
Lidé vnímají svět na základě jejich smyslů, tyto vjemy společně s naším mozkem zajišťují proud informací, který nám dává povědomí o prostředí kolem nás. Všechno naše poznání světa pochází od našich smyslů, jinými slovy celá realita je jen kombinací senzorických informací a snahy našeho mozku smysluplně rozpoznat přijaté vjemy. Je tedy reálné předpokládat, že pokud našim smyslům prezentujeme vytvořené informace, bude naše vnímání reality změněno odpovídajícím způsobem. Budeme poznávat realitu, která ve skutečnosti neexistuje, ale budeme ji vnímat jako skutečnou. Tento jev nazýváme Virtuální Realitou [12].
V této době se často za virtuální realitu považuje trojrozměrné, počítačem vygenerované prostředí, které může být prozkoumáváno člověkem, popřípadě je s ním možné i interagovat. Osoba se pak stává součástí virtuální reality a může manipulovat se světem kolem sebe a provádět různé další akce.

3.1 Přístupy

V dnešní době existuje několik různých systémů používaných k implementaci virtuální reality jako například headsety, všesměrové běžecké pásy nebo speciální rukavice, tyto systémy jsou z pravidla řízeny pomocí počítače. Obvykle jsou použity ke stimulací více smyslů společně pro vytvoření lepší iluze reality. Obecně je dosažení kvalitní virtuální reality poměrně obtížné, naše smysly a mozek se vyvinuly tak, aby nám vjemy zprostředkovávaly naprosto synchronně. Pokud je tedy nějaký z našich vjemů v rozporu s ostatními nebo prostě jen neodpovídající, i jen nepatrně, dokážeme to poznat. Tyto problémy v závěru mívají řadu následků jako například ztráta imerze, přesvědčivosti a vůbec pozitivní zkušenosti z virtuální reality. Při závažnějších problémech pak dochází k jevu zvaném *Virtual reality sickness*, nebo také nevolnost z virtuální reality. Tyto problémy je pak nutné řešit jak na straně hardwarové tak softwarové, jak na straně technické tak koncepční. Je nutné, aby technologie pro virtuální realitu fungovala podle lidské fyziologie. Například lidské zorné pole má okolo 180 stupňů, přičemž něco přes 100 stupňů vidění periferní [11]. Většina lidí si neuvědomuje své periferní vidění do té doby než je ztratí (Ztráta periferního vidění je známá pod jménem *Tunnel Vision*). Tento jev je spojen s kvalitou vizuálních vjemů poskytovaných systémem virtuální reality. Příklad nesouhlasného vnímaní mezi středním uchem a zrakovým vjemem je podobný jako například čtení v autě nebo na lodi. Tento rozpor může způsobit nevolnosti (*motion sickness*).
Pokud se v rámci implementace virtuální reality podaří dosáhnout správné synchronizace, je možné dosáhnout stavu, ve kterém daná osoba opravdu cítí, že se nachází v daném prostředí (anglický výraz *sense of presence*, tento pojem úzce souvisí s pojmem imerze).
Ač se stále objevují nové způsoby využití virtuální reality, její velký potenciál je již nyní zřejmý. Nejzjevnějšími případy jejího užití jsou například imerzivní filmy nebo video hry. Zábavní průmysl má sice multi-miliardový rozpočet, ale to ovšem neznamená, že virtuální realita nemá žádná další využití. Příklady aplikace můžeme sledovat například pro architekturu, sport, medicínu, umění a další. Často se najdou případy, kdy je příliš nebezpečné nebo nákladné provést něco ve skutečném světě, v takovém případě pak nastupuje virtuální realita. Od trénování pilotů nebo chirurgů nám ukazuje, že znalosti a zkušenosti v ní nabité se přenášejí do skutečného světa. A jak se cena virtuální reality snižuje, a ta se stává čím dál více populární a rozšířenější, vidíme také větší a větší využití mimo zábavní průmysl.

3.2 Interakce s virtuální realitou a její problémy

Během historie vývoje systémů pro virtuální realitu se vystřídalo již několik různých postupů k jejímu dosažení. Tato kapitola se věnuje některým z nich. Tato kapitola se také bude týkat *virtual reality sickness*, což je obávaný problém, kterého se však lidé často bojí zbytečně. V podkapitolách bude rozebráno, jak vzniká, jak se mu dá předcházet a jaký je jeho vliv na virtuální realitu. Jedna z prvních věcí, co většinu lidí napadne při prvním zmínění virtuální reality, je právě ona případná nevolnost. Strach je ovšem často přeceněný, jelikož lidé soudí dříve než mají jakoukoliv zkušenost a vychází například z předpokladu, že to bude jako čtení v autě. A jelikož tento jev opravdu může nastat, je třeba si v rámci návrhu virtuálního zážitku dát pozor na případné spouštěče této reakce. Existence tohoto jevu vychází z vlastnosti lidského organismu vyhodnocovat nesouhlasné vjemy jako příznak otravy a pokusit se o zbavení se případného jedu.
Obecně můžeme dělit problémy s nevolností ve virtuální realitě do tří kategorií. Dělení je založeno na původci nepříjemného zážitku, ale také na postupech při jejich řešení. Ač spolu do jisté míry souvisí, každá má své vlastní způsoby řešení nebo předcházení problémů. Kategorie jsou hardwarové, softwarové a individuální. Jako poslední podkapitolu uvedu možnosti a problémy s pohybem po virtuální scéně, protože to jako nedílná součást většiny aplikací inherentně patří do interakce s virtuální realitou.

3.2.1 Hardware

První zásadní problém bývá způsoben nedokonalostí techniky. Problémy v této kategorii vždy závisely na konkrétní implementaci navození virtuální reality.
Nejdominantnější typ hardwaru této doby je pravděpodobně populární headset [13], tedy systém přenosný na hlavě. (Nebo také HMD – *Head Mounted Display.*) Toto řešení v sobě skloubilo cenovou dostupnost a kvalitu i s vývojářskou přístupností. Tyto zařízení se obecně dělí do dvou typů, systém jako například htc vive nebo occulus rift jsou ve své podsatě speciální vstupně/výstupní periferie. Výrobky jako google cardboard nebo samsung gearvr pak vyžadují jiný přístroj, který funguje jako výpočetní jednotka, display a poskytuje i prostředky pro vstup, v těchto případech je jím mobilní zařízení. Samotný headset je pouze jakési pouzdro s čočkami zakotvující obraz z displaye pro trojrozměrné vidění. Tyto zařízení poskytují primárně vizuální a zvukové vjemy. Headsety obvykle snímají pohyb hlavy pomocí akcelerometrů a gyroskopů (ale také například pomocí optických snímačů pozice) a pak tuto informaci používají ke správné aktualizaci obrazu. Objevuje se také sledování pozice postavy nebo rukou v prostoru, existují ale i snímače pro celé tělo [14]. Ve snaze získat konkurenční výhodu se firmy snaží o co nejpřesnější a nejúplnější získávání informací pro dosažení imerze ve virtuální realitě. Z těchto postupů celkem přirozeně vyplývají jejich nevýhody, lidský mozek si uvědomuje pozice jeho končetin pomocí několika různých technik, například pomocí paměti, hmatu nebo napětí ve svalech. Toto uvědomění bývá někdy považováno za jeden z lidských smyslů. Při nesouhlasu smyslových vjemů obvykle vyhrává vizuální vstup, pokud ale například pozice neodpovídá příliš zjevně nebo nesouhlasí přímo pohyb končetiny, nastává velmi nepříjemný pocit, který může vyvrcholit až v samotnou nevolnost. Jako příklad uvedu jev známý z htc vive, v případě špatného snímání pozice ovladače v ruce pomocí vizuálního sledování (systém *Lighthouse*), vive přechází na dopočítání potřebných informací z akcelerometru v ovladači. Pak může nastat problém, že ve virtuální realitě ruka "odjíždí" nějakým směrem, ačkoliv ve skutečném světe se nehýbe což způsobuje poměrně nepříjemný pocit. Většina senzorů například také nesleduje spodní polovinu těla, což vnáší do vývoje jakési dilema. První možnost je pokusit se pozici nohou odhadnout a riskovat nepřesnou pozici, druhá, častěji používaná varianta je nezobrazovat tělo uživatele a obejít tím tento problém.
Obecně by se dalo říct, že většina těchto problémů je způsobena nespolehlivostí a nepřesností sledovacího systému nebo jeho prostou nedokonalostí.

3.2.2 Software

Existuje také řada problémů ležících na pomezí mezi softwarovými a hardwarovými. Jelikož výkon žádného hardwaru není nekonečný, může kombinace jeho omezením se špatnou optimalizací aplikace způsobit pokles obnovovací frekvence obrazu, nízký počet snímků za vteřinu (*fps*), vede k trhavému zpoždění vnímaného uživatelem, tento jev je velice nepříjemný a u většiny lidí rychle vyvolává nepříjemné nevolnosti. Z tohoto důvodu je již během vývoje nutno uvažovat vhodnost aplikace pro danou platformu a samozřejmě optimalizace.
Přední výrobci VR brýlí prosazují obnovovací frekvenci 90 snímků za sekundu [15]. Cokoliv pod tuto hodnotu může působit nepohodlně, nepřirozeně nebo ohrožovat uživatele VR nevolností (v tomto případě někdy nazývanou *Simulation sickness*, tedy simulační nevolností). Proto je ve VR důležité udržovat jak stabilní výpočetní náročnost a zabránit softwarovému kolísání obnovovací frekvence, tak také používat dostatečně kvalitní display. Systémy využívající desktopy a jiné výkonné stroje tady naráží na problémy méně často než například mobilní VR.

Softwarové problémy úzce souvisí s návrhem aplikace, kde většinou leží i jejich případné řešení.
Typický problém s nesouhlasnými vjemy může být způsoben prostým pohybem ve virtuální realitě, zatímco ve skutečnosti stojíme. Rychlost pohybu naštěstí mozek převážně vnímá pomocí očí, mnohem větší problém tedy je zrychlení. Vjem středního ucha, které nevnímá žádnou změnu rychlosti, nesouhlasí s naší vizí a naopak. To může způsobit nepříjemný pocit podobný nevolnosti z auta nebo mořské nemoci. Tento jev může být oslaben (nebo zesílen) případnou očekávávaností takového zrychlení. Problémy způsobené očekávaným odstředivým zrychlením se dají odlehčit například vodící čarou, plynulostí pohybu nebo nějakou z dalších technik pro snížení pohybové nevolnosti (*motion sickness*) [16]. Nejjednodušším řešením pak je navrhovat aplikaci s co nejmenším množstvím pohybu, který neodpovídá pohybu uživatele ve skutečnosti. Tento postup však nemusí být vždy možný, proto je vhodné se omezit například na to, aby byl veškerý neskutečný pohyb alespoň vyvolaný uživatelem nebo ve směru jeho pohledu. Vyhýbání se prudkému zrychlení nebo zpomalení je také vhodné. Pro vnímaní člověka mohou být také nepříjemné rychle se míhající objekty v jeho periferním vidění. Některé pokusy ukázaly, že velkému množství lidí pomáhalo mít v zorném poli stacionární předmět, například kokpit letounu nebo stacionární mřížku. Další možností je například snížit při vyšší rychlostí zorné pole (*Field of view*) [17].
Velmi nepříjemně také působí otáčení kamerou způsobené aplikací. Jev ekvivalentní rozhlížení se po scéně, který ale není vyvolán samotným otáčení hlavy. Podobný vjem může způsobit například převrácení uživatelem řízeného vozidla. Je to podobný efekt jako v případě nepřesného sledování pozice nebo natočení. Ve snaze vývojáře by tedy mělo být, aby byl pohyb co nejpředvídatelnější a nejočekávatelnější. Obvykle je velmi nepohodlné, pokud člověk nechápe svůj vlastní pohyb, toto pak vede k dezorientaci a proto by aplikace neměla tento jev vytvářet a cílit raději na pohodlí uživatele.
Může se ovšem stát, že bude cílem aplikace navodit právě takovéto stavy a vjemy, například pro nějakou hororovou hru nebo pro nějaký interaktivní zážitek. Pak je naopak tyto poznatky možné použít k efektivnímu dosažení cíle aplikace.
Pokusy ukázaly, že efekty způsobující zkreslení obrazu, které neměli žádný translační nebo rotační efekt nezpůsobovali zásadní problémy téměř nikomu. Například rozmazání obrazu, vinětace, efekt čočky nebo i prosté snížení rozlišení. Tyto efekty začaly způsobovat nepříjemnosti až při delším vystavení [18].

3.2.3 Pohyb po scéně

Již zmíněný vive také disponuje možností sledovat pohyb hráče po celé místnosti a přenášet jeho pozici do virtuální scény. Tato vlastnost přináší velmi kvalitní vcítění hráče do prostředí a odemyká velké možnosti pro aplikace, které jsou schopny této vlastnosti využít. Jako příklad zde můžu uvést virtuální muzeum, s možností obejít exponát kolem dokola, jako by to bylo možné provést ve skutečném světě. S touto výhodou však přichází i velká nevýhoda, jakýsi virtuální paradox. Protože Snímače jsou jen několik metrů od sebe, tvoří místnost typicky o velikosti tři až pět metrů na

délku a na šířku. Pohyb je tedy volný nic méně omezen na poměrně malý prostor. Vývojář má sice otevřené dveře k volnému pohybu, ale nakonec stejně tvrdě narazí na tohle omezení. Existuje několik postupů, jak virtuálně rozšířit prostor některé z nich jsem ve své práci vyzkoušel a otestoval, o nich později.
Existuje také technika jak nekonečně rozšířit prostor, bez nároku na vývojáře aplikace. Tohoto je možné dosáhnout, za použití všesměrového běžícího pásu ať už aktivně pohybujícího se proti uživateli, popřípadě vytvořeného ze zešikmeného kluzkého povrchu a speciálních bot.
Jako zástupce zde uvádím Virtuix Omni, který jde cestou speciálního povrchu, ve kterém při každém kroku noha sjede po kluzké podložce zpět do výchozí pozice. Hráč je v něm ukotven ve speciálním rámu. Ten je umístěn přibližně ve výšce pasu a hráč je k němu pevně připnut. Toto zařízení je navrženo pro použití pro „obyčejného“ hráče. Se snahou snížit jeho cenu na rozumnou mez a velikost celého systému tak, aby se dal umístit do případného obývacího pokoje. Podle recenzí je však nutné si na používání Virtuixu chvíli zvykat a pohyb není tak přirozený a intuitivní jako ve skutečnosti. Nejlépe prý systém funguje při chůzi vpřed, kroky vzad a do stran jsou proveditelné, ale náročnější. Kvůli principiálně nízkému tření s podložkou je pak téměř nemožné otáčet se na místě [27].
Druhý případ jiného přístupu je Infinadeck, tento poměrně velký přístroj slouží jako běžící pás, který dokáže rotovat ve dvou osách. Dále je k uživateli připnut snímač směru pohybu, podle kterého celý systém reaguje. Tento přístup umožňuje přirozený pohyb, jako ve skutečném světě. Tento systém umožňuje všechny prostředky normálního běžného pohybu, jako couvání, otáčení se na místě a podobně. Zařízení je však stále ve vývoji. Maximální rychlost běhu je omezena na relativně volný klus a kotva držící hráče neumožňuje skoky a podobné aktivity [28].

Pozn. Dle recenzí které jsem našel je srovnání těchto dvou periferií celkem očekávávané, Infinadeck je skvělý, ale velký a drahý. Virtuix je znatelně horší nicméně mnohem levnější a menší.

3.2.4 Individualita uživatelů

Každý člověk je jiný a stejně tak je vnímání virtuální reality odlišné uživatel od uživatele. Někteří lidé bývají odolnější nebo zvyklejší na jisté vjemy než jiní a naopak. Lišit se také bude rychlost, s jakou si uživatelé na samotnou imerzi zvyknou. Do této kategorie spadá nejen vliv negativních efektů na člověka, ale také například jeho schopnost efektivně využívat nástroje ve virtuální realitě nebo odhadovat ze znalostí mechanik aplikace. K tomuto jevu se dá přistoupit i z pohledu vývojáře. Aplikace může obsahovat nastavení, které v případě problémů zamezí nevolnostem, například již zmíněná stacionární mřížka může být prvkem v možnostech programu. Uživatel, který bude mít problém si ji poté nakonec může vypnout, jakmile si zvykne na ony netypické vjemy. Zde se navíc může objevit jev, že pokud uživatel na nějakou činnost není zvyklý nebo ji neumí ve skutečném světě, bude ji mít problém zvládnout i v tom virtuálním a naopak. Typicky činnosti jako je kreslení nebo tvorba hudby, ale i jednodušší věci jako házení předmětů, budou silně ovlivněny zkušenostmi a schopnostmi člověka, nehledě na návrh aplikace. V této fázi přichází možnost uzpůsobit aplikaci více tak, aby mohla sloužit k výuce dané činnosti (výuka může být i součástí hry dokonce i mechanik). Z toho také plyne možnost učit se ve virtuální realitě a poté přenést naučené dovednosti do skutečnosti. Občas se navíc objevují velmi citliví lidé, kteří opravdu mají problémy už se samotným vizuálním vjemem.

Problémy zmíněné v předchozích odstavcích bývají často předkládány jako závažné chyby celého konceptu virtuální reality a často bývají jádrem předsudků u mnohých lidí.

3.3 Potenciál a využití virtuální reality

Ač potenciál pro aplikaci Virtuální reality v herním průmyslu je obrovský, není to zdaleka jediné odvětví, ve kterém se najde pro tuto technologii použití.

3.3.1 Virtuální terapie

Skutečný svět je nahrazen počítačem generovaným a uživatel najednou stojí na mostě přes propast nebo na vrcholku hory. Podle studie byli pacienti trpící strachem z výšek schopni snáze konfrontovat jejich fóbii ve virtuální realitě. Po skončení léčby, dokázalo více než 90% pacientů překonat jízdu ve skleněném výtahu do 15-tého patra. Výsledky podle průzkumu přetrvávají i měsíce po skončení terapie. Tyto úspěchy značí možnost širšího nasazení této technologie při léčbě nejen strachu z výšek, ale i dalších fóbií a nervových potíží [19].
Všechny takovéto studie ukazují provázanost mezi vnímáním ve virtuální realitě a skutečnosti. Oklamání uživatele, tady bráno v dobrém slova smyslu, je již na dostatečné úrovni, abychom dokázali využít jeho vlastností k léčbě psychiky, ale pravděpodobně také mnoho dalšího.

3.3.2 Výuka, trénink a výcvik

Výzkum na poli chirurgie ukázal, že použití simulace ve virtuální realitě k dosažení určitého cíle, značně vylepšil výkony lékařu během operace. Cílem tohoto experimentu bylo dokázat, že se dovednosti získané ve virtuální realitě přenáší do skutečného operačního sálu. Cílem byl trénink, snížení chybovosti a certifikace chirurgů [21].
V armádě, námořnictvu či letectvu hrál výcvik vždy zásadní roli. Dovednost vykonávat rozkazy bez pochybností a reagovat bez prodlevy v rámci situace byli vždy důležitou součástí bojového výcviku. Společně s týmovou prací je možné všechny tyto dovednosti trénovat i ve virtuální realitě. Vojáci se učí jak udržet chladnou hlavu a koordinovaně postupovat se zbytkem své jednotky [22]. Existuje několik cest, jakými je k tomuto možné přistupovat. Již samotné hry mohou nabízet rozsáhlé virtuální prostředí a jsou vhodné pro začínající rekruty. Pokročilejší systému jako například CAVE [20] nabízí prostředky pro výcvik celých skupin v různých bojových situacích. Tato technologie je založena na promítání obrazu do okolí uživatele. K aktualizaci obrazu pak také dochází pomocí sledování jeho pozice. Uživatel dále používá 3D brýle k dosažení vjemu prostorového obrazu.
Podobným způsobem, který byl zmíněn v předchozím odstavci je také možné trénovat vojenské mediky.

3.3.3 Architektura, Virtuální prohlídky a Design

Dalo by se říct, že kdekoliv, kde je možné použít působivý trojrozměrný vizuální vjem, je vhodné a výhodné využit virtuální realitu. Toto tvrzení sebou nese jistá omezení na virtuální prohlídky nedosažitelných nebo neexistujících prostor je virtuální realita dosti bezkonkurenční technologie.
Od rekonstrukce antického města po stavbu nových budov si virtuální realita nachází využití i v architektuře a s ní spojených disciplínách [23]. Nynější vývoj v oblasti VR také znamená snížení pořizovací ceny pro danou firmu, investice do ní je pak menším riskem.

3.3.4 Filmy

Společně s 360 stupňovými videi si i filmy začínají razit svoji cestu na pole virtuální reality. Filmy mohou nabízet částečně interaktivní pohled na děj s několika perspektiv. Ač filmový VR obsah vypadá působivě, jeho produkce je stále poměrně problematická a nákladná [24]. Na druhou stranu, tento technologický vývoj přispívá modernímu trendu, kdy se úspěšná vývojářská studia na poli počítačových her snaží o svoji vlastní filmovou tvorbu. Toto odvětví navíc naráží na jeden vcelku nečekaný a zatím ještě ne zcela prostudovaný jev. Výhodná cílová skupina pro VR jsou hráči právě počítačových her a jeden z důvodů proto je, že jsou již zvyklí na pohled z první osoby. Snaha

nárazově přivést větší skupiny lidí k VR má zvýšené nároky na vývoj produktu z tohoto média a to právě proto, že daná cílová může být v průměru mnohem snáze zasažena některým z vedlejších účinků. Nekvalitní zkušenost pak může navíc způsobit špatný pohled na platformu jako takovou a to právě u uživatele, který si právě tvoří první dojem. Více v kapitole zvyk na VR.

3.3.5 Hry

Ač je termín hra překvapivě špatně definován, v tomto případě můžeme použít název *interactive experience* (interatkivní zkušenost/zážitek). Virtuální realita tento obor rozšiřuje o imerzivní trojrozměrný zážitek a často i o netypickou skupinu vstupů. Průmysl počítačových her se stal jedním z hlavních ekonomických tahounů soudobé, uživatelsky dostupné, virtuální reality. Velké hardwarové, ale a softwarové firmy, jako například facebook, samsung, valve nebo htc investují do rozvoje virtuální reality. Tato silná podpora na jednu stranu urychluje vývoj této platformy, na druhou stranu, ale také způsobuje nutnost výběru konkrétní platformy pro vývoj aplikace. Ač se displaye a zobrazování jako takové příliš neliší, vývoj vstupních zařízení je odlišný a například mobilní zařízení poskytují podstatně menší výpočetní výkon. Tyto a další problémy často nutí vývojáře vybrat si konkrétní typ systému pro virtuální realitu. Do budoucna také hrozí nedostatečná zpětná kompatibilita mezi jednotlivými verzemi zařízení, ale je taky možné, že tento problém bude vyřešen uspokojivě [25][26].

3.4 Člověk ve virtuální realitě

Dosud jsme se věnovali virtuální realitě téměř výlučně z pohledu nových technologií a nových přístupů, využívajících specializovaného hardwaru. Můžeme se ale podívat na Virtuální realitu v poněkud obecnějším měřítku. Ač samotný pojem realita je obvykle využíván ve významu stavů věcí, které reálně existují, lidské vnímání a myšlení je právě to co pro nás definuje svět, ve kterém žijeme. Pak akt tvoření virtuální reality pro daného člověka je vytvoření prostředí, které jeho smysly a rozum přijmou jako skutečnost na podvědomé úrovni. Člověk disponuje schopností vnořit se do systému a přemýšlet v jeho kontextu jako by nahradil reality, ve které skutečně existuje. Na podvědomé úrovni si každý z nás vytváří mentální model reality kolem sebe, který mu pomáhá v rámci jeho života. Zároveň jsme také schopni adaptovat svůj vnitřní model dle daného virtuálního světa a využívat této zručnosti například k předvídání výsledků našich akcí. Tento jev platí ve skutečnosti a stejně tak platí ve virtuální realitě, pokud obsahuje pevně daná a pochopitelná pravidla.

3.4.1 Imerze

Obecně tento pojem znamená ponoření nebo vtažení. U lidí hovoříme obvykle o imerzi do určité situace nebo stavu. Jedná se o specifický psychologický zážitek, který přináší odlišné vnímání, participaci, ale i identifikaci, například dějovou. Pojem imerze se používá v různých dalších uměních (kinematografie, ale i výtvarné umění), ale v kontextu této práce budeme vycházet z imerze v počítačových her a hlavně ve spojitosti s virtuální realitou.
Samotný termín je spřízněný s pojmem teleprezence (nebo jen prezence) a to ve vztahu „pocit býti tam“ Tento jev byl v posledních letech studován a například v knize [29], si autoři všímají spojitosti s šesti hlavními rysy prezence. Tyto rysy jsou, za prvé kvalita sociální interakce, ve smyslu rozšíření a zlepšení této pomocí prezence. Za druhé, je prezence založena na vysoké míře realismu (zobrazovaného) prostředí, dále přináší efekt přenesení někam jinam. Čtvrtým rysem je ponoření našeho těla do virtuálního prostředí (percepční a psychologická imerze). Za páté je možnost uživatele netriviálním způsobem ovlivňovat okolní prostředí a za šesté vnímat počítač samotný jako inteligentního sociální agenta. Tento koncept byl převzat jako klíčový aspekt počítačových her.

Je uváděno, že právě těsný vztah mezi vizuálními, kinestetickými a zvukovými vjemy vede právě k pocitu imerze, který se snaží vytvářet mnoho počítačových her, simulací samozřejmě systémů virtuální reality [31].
Míra subjektivity tohoto fenoménu vede na problémy jej objektivně a vědecky změřit, mimo dotazníky byly i pokusy o měření konkrétních tělesných znaků (například rozrušení) a chování. Tyto metody nevyrušují uživatele od zážitku a nejsou ovlivněny zprostředkováním jejich pocitů.
Objevuje se i kritika imerze a to takové, že samotné pohlcení do fantazijního světa brání kritické distanci a myšlení o daném díle.

Největší problém však pravděpodobně spočívá ve špatně standardizovaném používání konceptu. Jedná se o poměrně často (v relevantních kruzích) používaný termín a je často využíván v odlišných definicích a významových použitích [30].

3.4.2 Dosažení imerze

Nehledě na kritiku se zde budeme věnovat příkladům umožňující zlepšení imerzivního pocitu, ve smyslu vcítění nebo vnoření se do předkládané virtuální reality. Uváděné postupy jsou obecné nemusí se tedy týkat výlučně typu virtuální reality získané vizuálním 3D prostředím.
Obvykle se dá seskupit hry dosahující imerze do dvou skupin, první kategorie vytváří bohatý myšlenkový model herního prostředí a druhá konzistenci mezi věcmi v rámci prostředí. Nejprv k první kategorii, ta často čerpá z konceptu bohatosti vjemů. Dosažení imerze tedy čerpá například z mnoha různých smyslových vjemů představující dané prostředí nebo lépe kompletnímu nahrazení vjemů podněty z daného prostředí. Dále je tu možnost zaměstnání uživatelova mozku nutností velkého soustředění na nějaký aspekt v prostředí. Nenechávaje prostor pro povšimnutí aspektů, které by mohly negativně ovlivnit imerzi. Nakonec je to mužnost upoutání příběhem, pokud je kvalitní dokáže často sám upoutat pozornost a vytvořit imerzivní zážitek. Poslední uvedený případ je více méně jediná možnost, jakou mají knihy k dosažení tohoto. Vrátíme-li se zpět do světa počítačových her, můžeme dát několik konzistentních znaků ve vztahu ke kvalitnímu imerzivnímu zážitku.
Odstranění nepatřičných vjemů, v herním světě zabraňuje nevhodnému vytržení z pocitu. Tedy zabránění momentu, ve kterém si člověk aktivně uvědomuje neexistenci prostředí. Konzistentní chování věcí v prostředí také pomáhá působením na naši představu, že „všechno dává smysl“. Do této kategorie také spadá uvěřitelná interakce se všemi entitami, které se ve hře vyskytují. Obzvláště u interakci s lidmi, ale i s objekty nebo zvířaty může snadno nastat ono vytržení způsobené tím, že se umělé entity chovají proti naším zkušenostem. Dále nepřerušovanost a čistota smyslových vjemů. Tento aspekt je pravděpodobně nejsilnější ve vizuální oblasti, ale zdaleka na ni není omezený. Typicky špatně provedené přechody mezi scénami mohou zničit zážitek z dané hry. Nakonec samotná interakce s prostředím, tento bod částečně zapadá pod všechny předchozí (které se sami částečně překrývají), ale je zde nutné si uvědomit, že zpětná odezva aplikace by měla být ideálně na co největší rozsah akcí uživatele. Toto zahrnuje velký počet interaktivních objektů, entity aktivně reagující na hráčovo chování popřípadě konzistentní fyzikální systém dávající pocit skutečného světa (i když ne nutně našeho). Dá se říct, že média nebo umění poskytují zážitek nehledě na to, jestli jsou imerzivní nebo ne, nutno dodat, že pro některé typy se nutně nemusí imerzivita hodit a tedy ne všechny zážitky musí být imerzivní. Nicméně tento přístup dává vývojářům další nástroj do repertoáru pro tvorbu kvalitního zážitku [32].

3.4.3 Vnímání sebe sama ve virtuální realitě

Jak ukazují experimenty, vnímání vlastního těla je silně založeno na smyslových vjemech. Tento jev ukazují například známé experimenty s vyměněním tělo nebo adoptováním cizí ruky. Ve zkratce dochází ke stimulování více smyslů současně stejným způsobem. Dobrovolníkovi je zakryta jeho skutečná ruka, zatímco na její místo se položí ruka uměla. Poté se experimentátor dotýká zároveň skutečné ruky, ale i umělé na stejném místě, tento postup vytvoří dojem, že umělá ruka náleží k tělu. Podobného jevu využívá za pomocí VR experiment, který provádí výměnu celého těla za tělo

plastového manekýna. Člověk má na očích nějaký VR headset jehož vize koresponduje s pohledem coby očima manekýna. Poté jsou opět vytvářeny hmatové pocity na na stejném místě u skutečného a umělého těla [33]. Výzkumníci dosahují něčeho co se obvykle nazývá *out-of-body experience* neboli zkušenosti mimo tělo. Tímto postupem bylo zjištěno, že se člověk dokáže takto vcítit do těla jiného člověka dokonce i opačného pohlaví. A ukazuje relativní jednoduchost změny toho co mozek vnímá jako sebe. V rámci experimentu se nepodařilo přesvědčit žádného z dobrovolníků, aby sebe identifikoval jako nějaký neživý objekt, například krabici nebo stůl. Vědec Kynan Eng, který prováděl tento experiment, ale předpokládá, že při delším časovém horizontu a s vhodnými nástroji by se dal tento jev rozšířit na jakýkoliv skutečný nebo i virtuální objekt. Tento jev také pravděpodobně souvisí se schopností člověka osvojit si libovolný nástroj jako rozšíření vlastního těla [34]. Tato schopnost pak člověku umožňuje bez přidané námahy a pozornosti (alespoň na vědomé úrovni) manipulovat kvalitně i se složitými nástroji (jako příklady se dají použít třeba kartáček na zuby nebo auto).
Všechny tyto vlastnosti lidského mozku potažmo jeho vnímání a představy souvisí s vnímáním virtuální reality. Zde narážíme na dříve zmíněný problém, kde část těla uživatele ve virtuální realitě neodpovídá pohybem nebo pozicí jeho skutečnému tělu, přičemž vyvolává nepříjemné pocity. Tento jev nám dává příklad toho typu, že nemusí tolik záležet na tom, jestli to co smysly vnímají odpovídá skutečnému stavu, ale hlavně musí smyslové vjemy odpovídat sobě navzájem. Například oči vidí pohyb vlastního těla, ale hmat ani žádný jiný vjem něco podobného nenaznačuje vyvolává rozpor, snad ještě větší problém vyvolá postup, kdy uživatel hýbe nějakou částí těla a viditelná změna pozice je znatelně v rozporu. Tento jev bývá zesílen v případě, že je navíc pohyb očividně nereálný.

3.4.4 Pro a proti virtuálního těla

V našem reálném světě jsme téměř stále zvyklí vidět nějakou část svého těla. A ač by se dalo čekat, že vize vlastního těla bude nutná záležitost, testy ukazují, že tomu tak není. Již nějakou dobu se ví, že v tradičních hrách, kde uživatel sleduje plochou obrazovku, není nutné, aby bylo tělo vidět. Tato vlastnost v aplikaci může být výhodou například pro zlepšení imerze, obvykle však vývojářům nestojí za námahu části těla, které ve hře nejsou aktivně využívané, nebo alespoň hráč nepotřebuje žádnou informaci o jejich stavu, neimplementovat. Toto tedy často vedlo k tomu, že byly vidět pouze ruce a to kvůli manipulaci z virtuálním světem. Jako příklad můžu uvést téměř libovolnou akční hru viděnou očima hlavní postavy. Kde ruce interagují v závislosti na hráčových příkazech se světem (typicky se zbraní) a zbytek těla je ignorován. Dokonce se ukázalo, že není přímo nutné, aby hráč viděl i samotné ruce, mozek se adaptuje na interakci přímo s předměty, pouze postačí pokud má na takovou manipulaci dostatek informací. Tyto vlastnosti pak splňuje i prostředí VR. Zde se opět ukázalo, že člověk potřebuje primárně informace k tomu, aby dokázal působit v nějakém prostředí a výsledky jsou dosti podobné těm desktopovým. Dalo by se však říct, že využití vlastního těla má ve VR větší prostor k vyvolání nějakého pocitu. Viz Ilustrace 4, ve které vývojáři velmi intenzivně využívají uvědomění si vlastních končetin. Zde se pak objevují dvě otázky, v jakém případě je výhodné tuto techniku použít, a jakým způsobem. Odpověď na první otázku dosti záleží na konkrétní aplikaci, obecně se dá říct, že pokud chceme, aby si uživatel své tělo uvědomoval ve hře, musíme mu jej ukázat. Dá se také čekat, že se do budoucna objeví další vhodné způsoby použití této techniky. Způsoby prezentace člověka ve virtuální realitě se navíc překrývají u samotného uživatele se všemi dalšími lidmi, na které v prostředí narazí. Tyto ostatní lidé mohou být samozřejmě také virtuální, mohou to ale také být postavy představující jiné uživatele využívající technologie VR ve stejné době.

Ilustrace 4: Využití vlastního těla ve VR

3.4.5 Implementace virtuálního těla

Existují různé postupy jak reprezentovat uživatele v rámci aplikace. Za roky existence počítačových her bylo vyzkoušeno spoustu různých variant. Některé z nejpoužívanějších technik v desktopových a konzolových aplikacích nejsou pro VR headsety vhodné. Například zobrazení rukou uživatele pomocí dvojrozměrných obrazů umístěných do souřadného systému kamery je pro dvoj-kamerový systém naprosto nepoužitelný.

Jednou z možností je použít 3D model s inverzní kinematikou pro všechny části těla, pro které není konkrétní pozice a natočení dané ze vstupu dané platformy. Inverzní kinematika (IK) je technika běžně využívaná pro řízení pohybu skupiny tuhých těles [36]. Před IK je postaven takový problém, že známe specifické body, takzvané koncové efektory, v nějakém translačním systému založeného na hierarchicky postavených kloubech, a chceme vypočítat konfiguraci ostatních bodů v řetězci tak, aby výsledek odpovídal. Jedna se o opak takzvané dopředné kinematiky, která je běžně používaná v počítačových animacích, kde zase známe konfigurace všech kloubů a hledáme jejich výslednou pozici. Nutno dodat, že inversní kinematika je mnohem složitějším problémem nežli dopředná. Metody používané pro výpočet IK jsou velmi podobné metodám používaných v robotice pro výpočty při ovládání robotických paží a existuje jich poměrně velké množství.

Před definované animace jsou další z možností jak se vypořádat s reprezentací hráčova těla, často používaná v počítačových hrách. Do VR se ovšem často zdaleka tolik nehodí právě kvůli mnohem lepší specifikaci pohybu hráče dané vstupními prvky platformy. Pro takové VR zařízení, které ovšem nemají přímé pohybové vstupy a jsou například pouze vizuální, se ovšem dá poměrně rozumně použít. Obecně se předpokládá, že zvýšené požadavky na plynulost obrazu ve VR pak kladou větší nároky na samotnou animací, ať už co se týče jejich požadavků na kvalitu provedení, tak i na její přirozenost. V trojrozměrném prostředí člověk snáze podvědomě odhalí případné chyby a je pak snadné, aby animace vypadala „levně“ popřípadě nekvalitně.

Pokud je to do aplikace vhodné anebo není rozsah dostatečně velký a není možné strávit velké množství času na implementaci IK nebo vytváření kvalitních animací, existuje několik jednodušších metod jak podat uživateli dostatečné informace o jeho těle. Jedním z přístupů je zobrazovat pouze konce končetin, respektive navázat pouze body, pro které existuje přímý vstup. Tento přístup nevyžaduje žádné složité výpočty, protože pouze umisťuje a natáčí nějakou 3D reprezentaci části těla do bodů jejichž vlastnosti jsou známé ze vstupního zařízení. Tento postup odpovídá, v případě htc vive, zobrazení pouze rukou, od zápěstí dále, a popřípadě hlavy uživatele. Tento přístup jsem částečně využil.

Nejjednodušší možností je pak zobrazit pouze manipulovatelné předměty popřípadě nějaký jednoduchý objekt, který reprezentuje stav uživatelových končetin ve virtuální prostředí. Jelikož knihovna patřící k htc vive již obsahuje funkční reprezentaci pomocí virtuálních ovladačů, které co nejpřesněji kopírují stav skutečných ovladačů, využívám tohoto přístupu nejvíce. Příklad využití této metody je pak zobrazení pouze předmětu, kterým hráč výlučně manipuluje (například zbraň v akční hře).

Více méně všechny tyto postupy se vztahují v podobné míře na reprezentaci ostatních hráčů z pohledu uživatele (Tedy jak vidí ostatní a nikoliv sebe). Hlavní rozdíly jsou poté v tom, že nároky na přesnost a responzivnost nebývají pro reprezentaci ostatních uživatelů tak vysoké, jelikož jejich skutečnou pozici neznáme. Stačí tedy, aby byly dostatečně odpovídající z pohledu aplikace, ve které se pohybujeme.

4 Návrh aplikace

Tato aplikace si klade několik cílů, těmto pak musí odpovídat její návrh. Testování je založeno na teoretickém podkladu a vlastních experimentech a vedle toho je snaha o zábavnost, která je založena na výsledcích testování a osobních zkušenostech.

4.1 Platforma

Platforma, na kterou je vyvíjeno, dost silně ovlivňuje ráz dané aplikace. Htc Vive disponuje displayem a sluchátky, pro vizuální a zvukové vjemy, jako výstupem. Tyto vlastnosti se silně podobají typickému desktopovému zařízení, kde pouze vizuální zážitek není zdaleka tolik imerzivní. Na druhou stranu, tato platforma disponuje několika vstupy, které umožňují (ale také vynucují) velmi odlišný návrh oproti klasickému osobnímu počítači. Systéme Lighthouse poskytuje přesné snímání pozice a pohybu uživatele po místnosti na prostoru několika čtverečních metrů a zároveň snímá pozici ovladačů a skrze ně umístění rukou v šesti osách volnosti (*six degrees of freedom*). Na ovladačích se také nachází několik různých tlačítek, vhodných pro rozdílné typy uživatelského vstupu. Dost možná nejvýznamnější tlačítko na ovladači je spoušť neboli trigger, které leží pod místem, kde má většina uživatelů (pokud používají ovladač jak byl navržen) ukazováček. Toto tlačítko snímá míru stisku a tuto hodnotu vrací do programu s poměrně vysokou přesností. Další také důležitý ovládací prvek na ovladači je grip, toto tlačítko se nachází z obou stran rukojeti ovladače a obvykle sedí v dlani uživatele. Nehledě na to, jestli je zmáčknuto z levé nebo pravé strany výsledek je pouze binární a to jestli je nebo není stisk. Ovladač dále obsahuje dotykové tlačítko, které krom stisku dokáže zjistit i místo stisku a do jistých mezí i míru stisku (odlišuje pouze varianty žádná interakce, dotyk a stisk).

Ilustrace 5: Htc Vive

4.2 Engine

Jako základ pro tuto aplikaci jsem vybral engine Unity3D a to hned z několika důvodů. Unity má zdarma použitelnou verzi obsahující všechny potřebné prvky k vývoji. Dále je to z hlediska VR jeden z nejflexibilnějších nástrojů. Má aktivní komunitu a kvalitní dokumentaci, usnadňující řešení většiny problémů. Při vývoji aplikace Unity umožňuje abstraktní přístup, základna enginu je však otevřená a je možné sestoupit na nižší úroveň v případě potřeby.
Ze dvou poskytovaných variant renderování vybírám deferred shading a to díky možnosti využít velké množství světelných zdrojů ve formě emisivních povrchů.

4.3 Přirozenost ovládání

Cílem návrhu je umožňovat uživateli co nejpřirozenější manipulaci s okolním prostředím. Tento postup vychází z předpokladu, že příjemnější a intuitivnější bude taková interakce, která se více podobá svému ekvivalentu ve skutečném světě. Například sebrání předmětu fyzickým stisknutím ruky, na místo výběru v menu. Součástí tohoto implementovat mechaniku, umožňující uživateli odkládat předměty do jeho vybavení, ekvivalentně s úschovou věci do kapsy na opasek či popruh.
Z ovládání aplikace vychází několik základních událostí, které můžou nastav pro téměř jakýkoliv interaktivní objekt ve scéně. První z nich je událost použít (use), ta je v aplikaci iniciována stiskem spouště s tím, že se ovladač musí nacházet v těsné blízkosti používaného objektu. Další interakce je sebrání (grab), které je založené na stisku bočních tlačítek na ovladači a také je u něj požadavek blízkosti k chycenému objektu. Upuštění předmětu probíhá stejným způsobem jako chycení a to stiskem tlačítka sebrání, tentokrát je umístění v nebo u objektu implicitní, protože je v držení ovladače. Ke speciálním účelům také využívám stisknutí touchpadu. Je zde také jedna výjimka a tou jsou primárně vrhací předměty. Poslední dva případy budou důkladně popsány později.

4.4 Obecné problémy a řešení

Jeden z největších problémů při vývoji této aplikace je omezený prostor, po kterém je hráči umožněn pohyb ve skutečném světě. Ač můžou být virtuální světy obrovské a sledování pohybu hráče precizně přesné, pokud uživatel nemůže udělat fyzicky krok ve skutečném světě, nebude jej schopen udělat ani ve virtuálním. K tomuto jsem se pokusil přistoupit hned z několika úhlu. První řešení leží v samotném návrhu aplikace a potažmo návrhu prostředí, ve kterém se bude uživatel pohybovat. Tedy omezení světa na různé pohyblivé platformy, výtahy a podobně. Druhá metoda je použití způsobu pohybu, který se odvíjí od fyzického pohybu hráče, ale nikoliv pomocí reálného chození. Možné je například použít šplhání, kde hráč pomocí rukou ručkuje po virtuálním laně či zdi. Poslední přístup je umožňovat více různých módů pohybu, kde může, vedle klasického pohybu chozením po místnosti, uživatel také překonávat větší vzdálenosti pomocí nějaké netypické metody. Například teleportační mechanika nebo systém s využitím létání, kde ruce slouží k navigaci a řízení směru letu.

4.5 Cíle vývoje

Cílem je pak vyvinout imerzivní a zábavnou aplikaci s pokud možno co nejnižším, nebo ideálně žádným, vlivem nevolnosti z virtuální reality. Dále také vyzkoušet možnosti, které tato platforma nabízí. V neposlední řadě otestovat možnosti pohybu a další interakce s prostředím a použít tyto zjištění k vylepšení zábavné části aplikace.

4.6 Aspekty aplikace

Jeden s dílčích prvků hry je předmět s pracovním názvem lampa, název zhruba odpovídá vzhledu předmětu, jeho účel je však dosti odlišný. Lampa má fungovat jako hlavní pohybový nástroj, umožňující překonat bariéru velikosti skutečné místnosti, ve které uživatel stojí. Idea je taková, že hráč dojde k lampě a interakcí s ní změní svůj pohybový mód, lampa pak umožní hráči létat prostorem, za použití ovladačů k řízení směru letu.
Součástí aplikace budou také cílové předměty, v rámci dosažení cílového předmětu bude uživatel muset splnit úkol s různě velkou obtížnosti. Cílem je zároveň s vývojem aplikace experimentovat s dobrými vývojářskými technikami co se týče příjemných a zábavných interakcí ve virtuální realitě a poté tyto aspekty zahrnout do úkolů k získání cílových předmětů. Mezi již vymyšlené zábavné akce proveditelné ve virtuální realitě patří například vrhání předmětů nebo šplhání prostředím.

4.7 Možnosti pohybu po scéně

Nejprve bude představeno rozdělení pohybových možností a následně budou představeny techniky pomocí nichž je možné tuto činnost provádět. S tohoto pak vyjdou návrhy a způsoby provedení přesunu po scéně použité v aplikaci.

4.7.1 Klasifikace

Kniha 3D User Interfaces Theory and Practice [35] definuje tři odlišné dělení způsobů pohybu po scéně. První z nich je zda je technika aktivní nebo pasivní. Zde jde o otázku zda-li uživatel ovládá svoji trajektorii po scéně nebo nikoliv. Tato práce se (podobně jako kniha) zabývá hlavně aktivním přístupem ve snaze o co největší interaktivitu. Kniha také zmiňuje typ pohybu, který je aktivní i pasivní zároveň. Další způsob klasifikace je založen na tom, jestli uživatel ke změně pozice ve scéně využívá skutečného pohybu popřípadě jestli fyzické tělo uživatele zůstává stacionární zatímco se virtuálně hýbe. Kde typicky desktopové aplikace používají pouze virtuální pohybu. V aplikaci, která je součástí této práce je k dispozici několik různých způsobu sledování pohybu, translačního i rotačního. Toto nám umožňuje využívat i fyzického pohybu a nebo dokonce kombinaci obou přístupů. Třetí možný přístup ke klasifikaci je podle dekompozice úkolu. Tento dělí pohyb do tří komponent, prvním z nich je podle výběru směru nebo cíle, kdy uživatel vybírá jak a kam se bude pohybovat. Druhým je pak systém, ve kterém uživatel ovládá svoji rychlost popřípadě zrychlení. Poslední podúkol je takový, že uživatel ovládá jak je cesta započata, jak pokračuje a jak je ukončena. Tedy podle toho jakým způsobem vstup ovlivňuje pohyb, například jestli vstup pouze iniciuje nebo je poskytování konstantně. Posledním typem klasifikace je podle metafory. Tento typ klasifikace by měl být nejsnáze pochopitelný z pohledu koncového uživatele a jedná se o popis pomocí přirovnání. Je tedy možné popsat pohyb jako, za pomocí „létajícího koberce“, výhoda tohoto typu klasifikace je také v její snadné rozšířitelnosti.

4.7.2 Techniky

Existují tyto hlavní techniky pro pohyb uživatele (zastoupeného kamerou ve scéně).

1. Techniky využívající fyzický pohyb - vhodné pro zlepšení imerze či vynucení fyzického zapojení uživatele.
    - Chůze v prostoru
    - Chůze na místě
    - Simulované chozením
    - Kola

2. Technika natáčení – tyto umožňují uživateli bez námahy (například v sedě) překonávat značné vzdálenosti, jsou poměrně univerzální a dají se použít s různými vstupy.
    - Řízení směrem pohledu
    - Ukazování
    - Řízení tělem

3. Technika kamery na dlani – umožňuje uživateli navádět pozici kamery, kterou sleduje scénu, pomocí různých ovládacích prvků.
    - Fyzické zatáčení nástrojem
    - Virtuální ovladač pohybu
    - Poloautomatické zatáčení

4. Technika plánování cesty – tato umožňuje uživateli přistoupit ke scéně jako celku a manipulovat v ní sebou k dosažení změny pozice.
    - Kreslení cesty
    - Označování bodů podél cesty
    - Manipulace s reprezentací uživatele

5. Techniky založené na cíli – tato skupina metod je založená na manipulaci objektem ve scéně.
    - Mapová nebo WIM specifikace cíle (World in miniature)
    - Technika zpětného přiblížení
    - Chytání se vzduchu
    - Manipulace s nehybným objektem

6. Techniky cestování pomocí změny měřítka – často ve spojení s jinými metodami

7. Techniky změny orientace pohledu – tyto metody mění hlavně směr a natočení kamery
    - Sledování natočení hlavy
    - Orbitální sledování (Uživatel otáčením krouží kolem cíle)
    - Neisomorfická rotace (Používané v případě chybějící části v zorném poli)
    - Virtuální sféra

8. Techniky specifikace rychlosti

4.7.3 Designové rady

Zde jsou tedy doporučené postupy při vývoji pohybového systém ve virtuální realitě:

- Technika cestování musí odpovídat dané aplikaci.
- Je možné využívat přirozené i „magické“ techniky
- Může být vhodné využívat kombinaci technik, vstupů a výstupů
- Pohybové techniky by měli být jednoduše integrovatelné s ostatními prvky interakce v aplikaci
- Různé cíle cestování by měly být podpořeny různými technikami
- Pro jednoduché cestování je vhodné použít techniky založené na cíli, a řízení směru pro průzkum a hledání
- Fyzické metody pohybu jsou v hodné pokud je vyžadována fyzická námaha uživatele nebo k dosažení lepší přirozenosti
- Pokud je důležitý kontext je lepší použít elegantní metody pohybu (například představení okolí pomocí postupné cesty)
- Je výhodné cvičit uživatele k použití sofistikovaných strategií ke zlepšení přehledu
- Může být vhodné využít stejný typ interakce pro více úloh, obzvláště pokud je cestování využito v kontextu manipulace

4.7.4 Výsledný návrh pohybu

Součástí celé aplikace by ideálně mělo být několik různých a nezávislých způsobů, jak docílit přesunu, nejen hráče, v prostředí. Několik z těchto technik poté bude vybráno do největší scény, která bude zároveň sloužit jako herní. Další aspekt je pokrytí několika různých typů pohybu, z hlediska klasifikace. To znamená využití postupů reálné pohybu, ale také kompletně virtuálních, zároveň též metody přirozené, ale i magické. Dále také technika založena na cíli a technika specifikace rychlosti.

5 Implementace

Jak již bylo zmíněno, celá aplikace je implementována v herním enginu Unity3D, za užití programovacího jazyka C#.

5.1 Obecný návrh

V rámci implementace jsem postupoval ve dvou hlavních krocích. První fáze se zabývala zejména přípravou na vytvoření hlavní části programu. Sestávala se z testovacích prototypů, které sloužily téměř výlučně k získávání informací o platformě jako takové a její spolupráci a spolu fungování s Unity. Součástí prvotní implementace byla jedna velká testovací scéna, do které byl postupně přidán každý další objekt jehož fungování bylo třeba vyzkoušet. Úkolem této fáze bylo zjistit všechny potřebné informace k implementaci druhé, finální, fáze. Zde bylo také vytvořeno několik objektů, které byly později z různých důvodů zahozeny a nejsou v konečném produktu, z uživatelského pohledu, dostupné. Tyto objekty však zůstávají uloženy ve složkách projektu a v případě potřeby by je bylo možné znovu zprovoznit.
Druhá část tvorby programu byla implementace samotných finálních scén, poučení se ze všech dosud zjištěných informací a vytvoření kompletního testovacího prostředí. V této části již program nebyl agregován do jedné testovací scény, ale byl sémanticky rozdělen do několika různých. Základem zde bylo vytvořit pro každý test vhodnou scénu, ve které by bylo možné co nejlépe otestovat danou funkcionalitu. Další cíl bylo mít alespoň jednu scénu, která bude působit jako herní, kde bude mít uživatel nejlépe nějaký cíl a bude nucen využít dostupných nástrojů k jeho dosažení. Ideální stav je pak ten, že k dosažení daného cíle nevede pouze jedna cesta. Zmíněná scéna se pak nesoustředí na izolované otestování nějaké mechaniky, ale naopak o její napojení a využití v komplexnějším prostředí.

5.2 Důležité třídy a rozhraní

Z pohledu implementace existují dva typy interaktivních objektů. Složitější s více různými třídami obsluhujícími jejich chování a jednodušší, kde je pouze jedna. Tento fakt je důležitý neboť u první varianty je nutné správně rozesílat události, ke třídě, která je potřebuje ke zpracování. Například událost typu kolize je v Unity zaslána prvnímu objektu (zde je objekt ve významu instance třídy), který ji dokáže přijmout v daném kontextu. Tento se pak musí postarat o to, aby se tato událost dostala kam patří.

5.2.1 Knihovní

VRTK_InteractableObject tvoří základ pro většinu interaktivních objektů a stará se o rozesílání událostí vytvořených uživatelem k daným entitám, pro zpracování z pohledu programu.
CameraRig zastupuje skupinu tříd, které se starají o základní funkcionalitu VR jako například renderování správného obratu do headsetu. Dále také tvoří rozhraní mezi hardwarem a softwarem.
Balíček VRTK tříd reprezentující hráče jako jsou *VRTK_Player_Object* nebo *VRTK_Headset_Collision*. Tyto jsou povětšinou poměrně málo rozsáhle třídy a každá z nich se stará o nějakou vlastnost hráče, jako například vytvoření události kolize hlavy uživatele s virtuálním objektem.

5.2.2 Vlastní

EntityInterface je rozhraní, které zajišťuje, aby daná třída, která jej splňuje dokázala přijímat události rozesílané z třídy, která se stará o tuto aktivitu.
Třída *Throwable* dědí ze třídy *VRTK_InteractableObject* a splňuje rozhraní *EntityInterface*, ale zároveň také slouží jako základní třída pro složitější objekty, tam pak rozesílá události.

BasicEntity je nejjednodušší třída, splňující *EntityInterface* tato třída je abstraktní, a má několik virtuálních metod.
GuidedMovement je třída, která dokáže řídit pohyb objektu po scéně pomocí interpolace pozice a rotace mezi body. K nastavení těchto bodů slouží rozšíření editoru *GuidedMovementEditor*, které umožňuje tyto body nastavovat přímo v grafické reprezentaci scény. V každém z řídících bodů je nastavena pozice, rotace a doba přechodu od posledního bodu (a pomocný údaj jestli pozici bodu zobrazovat, pro větší přehlednost).
MovingPlatformControll, MovingPlatformMotor, PlatformTracker je trojice tříd obsluhující pohybové chování objektu pohyblivé platformy. Motor se stará pouze o provedení dané translace pro samotnou platformu a objekty k ní přichycené. Control řídí rychlost a směr a také má na starost seznam objektů přichycených k platformě. Tato třída nic z této funkcionality nedělá aktivně, ale čeká na vstup. PlatformTracker má pak instanci na každém objektu, u kterého je výchozí chování přichytit se k platformě. Tento pak pro dosažení přichycení komunikuje s Control.

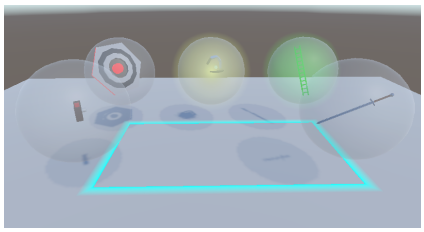
5.3 Důležité aspekty vývoje pro VR

Vývoj po VR sebou nese několik požadavků, které nejsou přítomné u ostatních platforem. Jeden z nich je různá velikost reálné místnosti. Stejně jako je problém samotný fakt omezené velikosti, další komplikaci přináší vlastnost, že každý uživatel může mít místnost různě velikou. K tomuto problému jsem přistoupil speciální chováním objektů, umožňující přizpůsobit svoji velikost vůči velikosti místnosti. To je pak například zřejmé u scény s výtahem, kde ten je vždy svojí velikost přizpůsoben na velikost skutečné místnosti. Tato obecnost sebou nese řadu dalších problémů jako například umístění objektů ve scéně na relativní pozice vzhledem k velikosti místnosti. Této vlastnosti je dosaženo pomocí umístěných kotev, jejichž pozice je nastavena v závislosti na měřítku předka kotvy. Objekt ve scéně je pak umístěn do pozice kotvy bez nutnosti dalších změn velikosti (Velikost samotných objektů pak vytváří hranici minimální velikosti skutečné místnosti).
Dalším omezením je potřeba vysoké obnovovací frekvence a tedy nutnost snížení náročnosti aplikace, vzhledem k nižší grafické náročnosti nebylo nutné v tomto případě tento problém řešit.

5.4 Finální scény

V závěrečné aplikaci se nachází pět herně-testovacích scén, dostupných z šesté scény, která funguje jako jakýsi cestovní uzel nebo rozcestí. Scény jsou, střelnice, v níž uživatel může střílet do terčů z kuše a házet nože, přičemž stojí na pohyblivé platformě. Dále temná scéna, ve které je opět kuše, ale hlavně lampa umožňující létání. Následně scéna jejíž jádro leží ve šplhání, ale obsahuje také teleport a poslední variantu kuše. Předposlední scénou je scéna bojová, která zkouší kombinaci netriviální činnosti, společně s rychlým pohybem v prostoru. Nakonec pak scéna s výtahem, která slouží k předvedení jiného typu řízení platformy a ukázky dalších objektů co se do ostatních scén nevešly.

Ilustrace 6: Scéna "rozcestí"



5.5 Možnosti rozšíření

VR v nynějším duchu, co se týče masového nasazení kvalitních headsetů je poměrně nová záležitost. Součástí nové technologie, nebo rozšíření jejího využití se nese i jistá neprozkoumanost možností. K tomu samotná virtuální realita může být využita k mnoha účelům sama o sobě. Jak jsme však viděli u příkladu pohybu po scéně velké množství návrhu již bylo vytvořeno před značnou dobou a i všechny nové možnosti stále bez problému spadají do již zavedených klasifikací.
Co se tedy týče testování samotného je zde spoustu ideí, které by bylo vhodné také vyzkoušet a implementace se zase nabízí jako vhodný základ pro další takové testování.

5.6 Využití jako základ

Výsledný kód je záměrně napsán v co nejmodulárnějším a nejobecnějším stylu. Součástí tohoto designu je snadná rozšířitelnost, ale také upravitelnost, co se týče již hotových objektů. Jádro implementované aplikace (společně s informacemi při tom nabytými) je možné využít jako základ pro hru nebo jiný druh aplikace využívající vstup výstupního zařízení htc vive.

5.7 Implementované techniky cestování

Zde je souhrn technik cestování použitých v aplikace.

Načtení scény

První metodou, která nutně nemusí být označena za čistě cestovní, ale částečně spadá do této kategorie, je technika umožňující kompletně změnit scénu. Ta se dá částečně přirovnat k technikám okamžitého přesunu, ne nepodobné teleportování. Ač je využita pouze v výchozí scéně, ve které si uživatel vybírá, kterou další scénu otevře, hraje zde klíčovou roli k nahrazení tradičního uživatelského rozhraní.

Létání

Létání je mechanika, kterou téměř nebylo možné vynechat, už kvůli jejímu subjektivnímu dopadu. Zde je implementována poměrně jednoduchým způsobem, kde uživatel pozicí ruky mění směr pohybu.

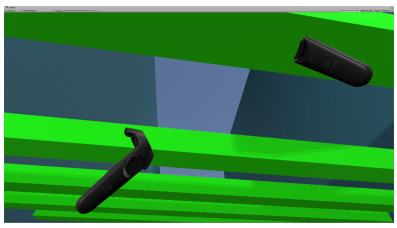
Teleportování

Pomineme-li načtení scény, teleportování se dá jako cestovní mechanika využít i v rámci změny pozice mezi dvěma, ne příliš vzdálenými, body. Zde bylo nutné prozkoumat, jakým způsobem udělat takovou náhlou změnu pozice, aby její použití nemělo na uživatele žádný negativní dopad. Zde se nabízí několik možností jak na poli samotné interakce tak co se týče grafického zpracování.

Šplhání

Šplhání je metoda, která částečně využívá metod fyzického pohybu, primárně je to ale technika založená na pohybu pomocí manipulace z nehybným objektem. Tato technika již byla implementována v knihovně VRTK, v rámci práce jsem jen vytvářel vhodné prostředí, ve které by se dala adekvátně otestovat a pokud možno zábavně využít.

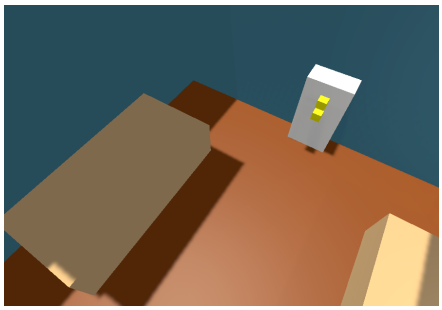
Ilustrace 7: Šplhání



Pohyblivá platforma

Další možnou metodou pohybu po scéně je pak připojit uživatele k nějakému objektu a hýbat poté tímto objektem. Na tomto principu pracují teoreticky všechny dopravní prostředky jako příklad se dá použít také výtah nebo obecně pohyblivá platforma. Aplikace navíc představuje i řiditelnou variantu, která se dá navigovat pomocí pák.

Ilustrace 8: Výtah



Lokální techniky pohybu a mrtvé zóny

Všechny výše zmíněné techniky obsahují jeden zásadní problém, a to je lokální podmínka nutná k jejich fungování. Jelikož všechny pohybové prvky aplikace (s výjimkou fyzického chození) vychází z funkcionality nějakého objektu ve scéně, může se uživatel dostat do problému, že žádná z nich nemusí být fyzicky na dosah a on nemá žádný způsob jak se k ní dostat, jelikož je na fyzický přechod příliš daleko. Tyto oblasti budu nazývati mrtvými zónami. Idea tedy je vytvořit nějaké záchranné mechaniky, které dávají hráči možnost dostat se z obtížné situace a nebo z mrtvé zóny.
Z důvodu existence mrtvých zón je pak vhodné, aby existovala nějaká záchranná mechanika, bez silného lokálního omezení, která by uživateli umožnila dostat se z problematického místa. Použití těchto technik je v aplikaci namapováno na stisk touchpadu.

Přitahování

Tato metoda vyžaduje konkrétní bod, který je umístěn někde ve scéně, nicméně k jeho využití nepotřebuje hráč být ve fyzickém dosahu, ale stačí mu tento bod vidět. Samotné použití je pak jednoduché stačí namířit ovladač na tento přitahovací bod a stisknout touchpad a uživatel je přímou cestou přitažen k tomuto objektu.

6 Interaktivní objekty a testování

Testování této aplikace slouží jak k jejímu vylepšení, tak k získání teoretických informací o daných subjektech. Hlavní a závěrečnou aplikaci pak vyzkoušelo přibližně dvacet lidí.

6.1 Cíle testů

Hlavním cílem testů je za prvé ověřit navrhované postupy a za druhé srovnat tyto postupy s jejich případnými ekvivalenty. Další úlohou tohoto testování je také získat obecný pohled na vlastnosti platformy htc vive a skrz ni na interaktivní aplikace ve virtuální realitě jako takové.
Konkrétně testování zahrnuje tři nejdůležitější aspekty. Tyto jsou možnosti přirozeného pohybu po scéně, intuitivní interaktivní ovládání a subjektivní hodnocení kvality jednotlivých interakcí. Jelikož získané informace, ale i samotná implementace mohou sloužit jako základ pro vývoj počítačové hry, mohou se ukázat právě subjektivní pocity z jednotlivých objektů jako nejdůležitější část zpětné vazby. Nebo jako prostředek pro poučení do budoucna, ať už jako ukázka úspěšných principů, které je možné využít, ale také upřesnění informace o tom, kterým se raději vyhnout.
Co se pohybových možností týče bylo účelem testování prozkoumat jednak, jak tyto typy vyhovují této platformě, ale také jak kvalitně pracuje tato konkrétní implementace těchto technik. U cestovních metod byla také snaha využít testování k jejich zlepšení nebo najití lepších metod.

VR za použití headsetu htc vive se vyznačuje možností chodit po místnosti, tato vlastnost ač přináší na první pohled možnost volného pohybu vytváří omezení odpovídající velikosti fyzické místnosti, ve které se hráč skutečně nachází. Tento problém se aplikace snaží několika způsoby řešit.
Výsledky testů jednotlivých interaktivních objektů a mechanik proberu v jiné kapitole.

Na většinu kladených otázek i v rámci subjektivních odpovědí testeři odpovídali stejně nebo velmi podobně. V textu tedy budou uvedeny zejména výjimečné případy nebo zajímavosti, vedle samotných zjištěných informací.

6.2 Prostředky a postup testování

Jak již bylo zmíněno, vyvíjená aplikace používá jako vstup výstupní zařízení htc vive. Testování probíhalo v místnosti o velikosti přibližně 2,5 metrů na šířku a 4,5 metrů na délku. Výsledný prostor pak byl, z bezpečnostních důvodů, o něco menší, abychom se vyhnuli případným konfliktům s nábytkem okolo. Bylo pak také nutné z prostoru odstranit všechny případné překážky jako například židle nebo batohy, o jejíž existenci pak uživatel ve virtuální realitě nemá potuchy. Dále bylo potřeba také informovat zrovna netestující přihlížející, aby dávali pozor, kde stojí, a nevstupovali nynějšímu hráči do cesty. Tento jev byl překvapivě častý, pravděpodobně proto, že lidé nejsou zvyklí na člověka, který se pohybuje po místnosti a přitom je nevidí. Při a po zkoušení aspektů aplikace bylo testerům položeno několik otázek, jejichž shrnutí bude tvořit základ následující kapitoly. Otázky se povětšinou týkali přesně cílů testování, přičemž ještě přibyla téměř obligátní otázka, zda-li se testujícímu uživateli nedělá nevolno. Dále bylo snahou zjistit, zda je uživatel schopen efektivně použít interakci s prostředím k dosažení svého cíle, neboli jestli dokáže techniky a mechaniky prezentované aplikací aktivně používat a využít jejich potenciálu.

6.3 Testování VR mimo hlavní aplikaci

Součástí práce jsou i testy, které nebyly součástí implementace v diplomové práci, jejich testování však zahrnuto je. Tyto testy byly provedeny na několika desítkách lidí různých věkových kategorií.

6.3.1 Let stíhačkou

Tento test vycházel z jednoduchého nápadu, hráč ovládá stíhačku, která letí ve směru pohledu. Samotná implementace této aplikace nebyla součástí diplomové práce, ale byla zhotovena jako semestrální projekt předmětu mobilní aplikace. Součástí této práce je tedy pouze testování a jeho výsledky. Program byl spouštěn na mobilním zařízení Samsung S7 Edge společně s headsetem Gear VR. Díky snadné přenositelnosti se naskytla možnost otestovat aplikaci na různých skupinách uživatelů. Cílem hry bylo uletět co největší vzdálenost, přičemž se uživateli stavěli do cesty různé překážky, kterým se musel vyhnout nebo je zničit.
Testování bylo prováděno velmi jednoduchým způsobem, ve kterém byli dobrovolníci tázáni na jejich subjektivní pocit ze hry, ale také jestli se u nich během letu neprojevily žádné znaky nevolnosti. Výsledek celkem přesvědčivě utvrdil, že celé obavy z přehnaných žaludečních potíží jsou mylné. U žádného z uživatelů, který hru hrál se nedostavily žádné problémy. A víceméně jediné stížnosti, které se objevovaly (mimo připomínky, jak by se hra dala vylepšit) se týkaly obtížnosti. Tento přístup byl ale částečně předpokládaný, jelikož hra byla předložena širokému spektru uživatelů od malých dětí až po důchodce.
Pouze v rámci jednoho z testů se za důsledku chyby v programu stalo to, že bod otáčení se posunul ze středu stíhačky (a potažmo středu hlavy uživatele) o celkem výrazný kus do strany a celá rotace při otočení hlavy pak způsobila prudké a nečekané otočení kamery kolem onoho vzdáleného bodu.

Jak se ukázalo, tuto chybu nesl tester velmi těžce. Téměř ihned po zjevení problému urychleně sejmul headset z hlavy.
Na druhou stranu, tento jev ukázal, že virtuální realita je oblast, ve které je třeba dát si o to větší pozor na případné chyby, jelikož taková chyba může vést pro člověka k dosti nepříjemnému zážitku. Tenhle fakt také více upevňuje důvody k testování aplikací a pojištění si, aby žádný takový problém nenastal.

Ilustrace 9: Let stíhačkou

6.3.2 Simulátor „opilosti“

Tato aplikace byla vyvíjena během akce VR Hackathon, součástí této práce je tedy opět pouze její testování (díky soutěži jsme měli možnost dát aplikaci vyzkoušet několika desítkám lidí). Hráč se ve hře nachází ve virtuální hospodě, s nekonečnou zásobou piva a vrhacích šipek na stole. Samotné házení je v testu jen jako referenční činnost, která dává hráči cíl činnosti v různé míře virtuální opilosti. S každým další pivem se obraz více rozmazává a rozhoupává a zkresluje. V této aplikaci byla snaha se opravdu posunout na hranu snesitelnosti vizuálního vjemu. Od jistého stupně zkreslení již měla většina uživatelů problém udělat rovný krok dopředu. Také kvůli obavám ze ztráty rovnováhy často přecházeli do stabilnějšího postoje. Proti našemu veškerému očekávání se však drtivé většině testerů nedělalo nevolno a to ani při největším stupni opilosti při kterém se málokdo vůbec pokoušel po scéně nějak pohybovat.
Součástí této scény byl ještě dalekohled, jehož původním cílem bylo pozorovat trefené terče. Ukázalo se však, že minimálně naivním přístupem nebude možné s nynější technologií vr headsetů věci podobného typu kvalitně implementovat. Snaha podívat se okulárem na nějakou část scény vede k velkému soustředění uživatelem podívat se jedním okem relativně malou části fyzického displaye. Tento postup zesílí vizuální nedokonalosti, kterých by si jinak uživatel nevšímal, jako jsou nedostatečné rozlišení obrazovky, nedokonalost čočky nebo jen grafické vady v samotné aplikaci.

Dokud nebudou tyto vady odstraněny, bude nutné podobné objekt zpracovávat jiným způsobem. Nutno dodat, že objekt jako takový se většině uživatelů subjektivně líbil, jelikož zmíněná vada se projevila jen při snaze jej efektivně využít (o což se většina lidí nepokoušela).

6.3.3 Zvyk na VR

Při používání VR headsetů začíná být vidět jev, který se může do budoucna ukázat jako velmi důležitý aspekt celé platformy. Tento jev je známý například i u cestovní nemoci z auta, takzvané kinetózy. Člověk se adaptuje. Zvyk je dost možná nejúčinnější způsob, jakým se trvale zbavit problémů s nevolností. Existuje řada lidí, kteří jsou přirozeně odolní a u nichž nedojde k tomuto úkazu na prvním místě. Dále jsou tu lidé, kteří si již jinou činností podobného rázu zvykli a potom nemají žádné problémy. Tato vlastnost také vytváří v testování celkem jednoznačný výsledek, a ten je, že lidé přicházející k VR po značných zkušenostech s hrami z jiné platformy mnohem méně trpí jakýmikoliv problémy. Tento jev by bylo vhodné dále prostudovat a otestovat. Je možné, že by se dalo využít VR na pomoc lidem, kteří trpí různými typy těchto nemocí.
Zvyk na VR je navíc velmi kvalitním argumentem proti komukoliv, kdo nevěří, že lidé nejsou na zkušenosti typu VR připraveni, potažmo není připravena tato technologie pro lidi.

6.4 Interaktivní objekty a mechaniky

Fyzická presence ve VR nám dává možnost intuitivně interagovat s okolním prostředím. Velká část práce se pak zabývá snahou nepoužívat abstraktní prvky rozhraní, ale místo toho manipulovat přímo s objekty.

6.4.1 Obecný objekt ve scéně

Základní idea intuitivní interakce s prostředím je určit jednoznačná pravidla manipulace a poté dodržet tato pravidla v rámci jakéhokoliv využívání libovolného objektu. Většina věcí ve scéně spadá typicky do dvou kategorií, držitelné a ne nedržitelné, hlavní rozdíl v tom jestli je hráč schopen takový objekt uchopit a přenést, případně používat nebo upustit. Držitelné objekty po uchopení zůstávají přichyceny k „ruce“ uživatele, zastoupené ovladačem v aplikaci. S nedržitelnými objekty je také možno manipulovat například použitím (zde jako příklad můžu uvést výběr scény, ve kterém je zvolená scéna označena „použitím“ objektu, který ji reprezentuje). Zvláštním případem objektů jsou takové, po kterých je možné šplhat, tedy objekt zůstává na místě, ale pohybuje se hráč.
Sebrání objektu je pak provedeno tlačítkem grip, kde na následující držení objektu již není nutno dělat nic. Upuštění drženého objektu je pak docíleno opětovným stiskem tohoto tlačítka.

6.4.2 Interakce pomocí mechanik

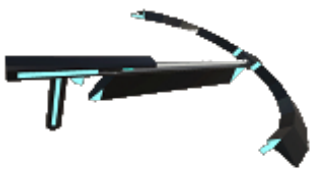
Důležitý aspekt intuitivního ovládání jsou co nejjednodušší a nejpřirozenější mechaniky. S co největší snahou přiblížit se skutečném použití předmětu v rámci možností platformy. Do této kategorie spadá veškerá obecná interakce, která nevychází z použití jedné konkrétní vlastnosti objektu a patří k nim hlavně šplhání a házení předmětů.

6.4.3 Seznam nejdůležitějších objektů

Níže je prezentován neúplný seznam entit, se kterými může uživatel v rámci aplikace interagovat. Některé byly z prezentačních důvodu v tomto seznamu vynechány.

Kuše a šipky

Ilustrace 10: Kuše



Cílem kuše je otestovat různé vlastnosti interakce s objektem. První důležitý aspekt je samotná schopnost uživatele interagovat s objektem, veškerá manipulace s objektem by měla být pokud možno co nejintuitivnější

a nejplynulejší, jelikož hráčova pozornost musí být věnována na jiné aspekty zacházení s objektem, například míření.
Kuše spadá pod držitelný typ objektů, je možné ji přenášet a popřípadě hodit. Její chování v takovém případě následuje fyzikální principy prostředí (například padá pod vlivem gravitace). Tato vlastnost byla podobně testována i u ostatních objektů stejného typu. Další možnost interakce s kuší je vystřelení šipky, mechaniku přebíjení nebo natahování jsem pro jednoduchost vynechal.
Kuše v různých scénách střílí různé šipky, na střelnici klasické, v lezecké scéně pak šipky, po kterých se dá šplhat a v tmavé scéně zase šipky, které fungují jako zdroj světla.

Kuše byla v testování pokaždé jeden z prvních nebo přímo první představovaný objekt. Velmi snadno se na ní dalo ukázat základní ovládání typu sebrat a použít. A v rozsahu testovaných osob se neobjevil nikdo, kdo by měl s ovládáním kuše jakékoliv problémy.
Zábavnost používání, ale i schopnost testerů kvalitně mířit, se dosti lišila, člověk od člověka. Nakonec však převažovaly pozitivní ohlasy, i když přesné míření se ukázalo poměrně obtížným.

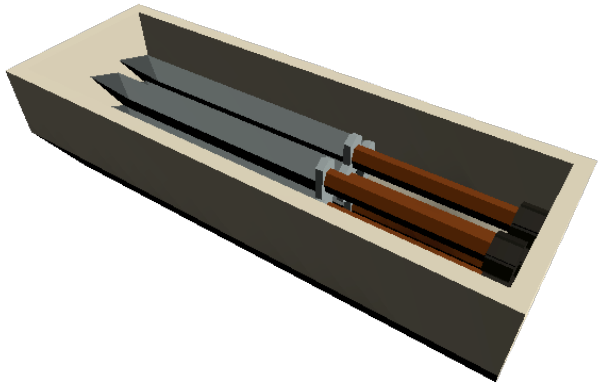
Lampa a let

Tento objekt je reprezentací pohybové techniky létání, jelikož tato aplikace představuje naprostou většinu mechanik reprezentovanými nějakou virtuální věcí, bylo nutné vymyslet objekt, jehož použití vyústí v samotný pohyb po scéně. Směr letu je řízen pozicí lampy oproti trupu uživatele. Lampa vedle toho slouží i jako případný zdroj světla. Při létání pouze jeden z testerů oznámil, že se mu dělá trošku špatně, nebylo to nicméně natolik závažné, aby činnost přerušil. Zbytek testovací skupiny nenarazil na žádné problémy.

Generátor interaktivních objektů

Použití tohoto objektu uživateli vloží do ruky objekt, který odpovídá typu nastavenému danému generátoru. Generátor také může aplikovat vynucené chycení objektu následně po jeho vytvoření. Spuštění generace se provede iniciací události použít na generátor. Typickým příkladem je generátor vrhacích nožů, který po použití dá nůž přímo do ruky uživateli a jelikož je tento akt na stejném tlačítku, nůž zůstane v uživatelově ruce do té doby než není toto tlačítko puštěno.

Ilustrace 11: "Generátor" nožů



Poté, co bylo vysvětleno ovládání programu se u žádného uživatele neobjevil žádný problém týkající se použití generátoru. Občas se stávalo, že někdo na první pokus nepochopil systém automatické chycení objektu při jeho vytvoření a okamžitě jej pustil, tuto chybu pak nicméně nikdo neopakoval.

Házení

V rámci implementace házení se naskytl problém s unifikací jejího ovládání. Existují předměty, které jsou primárně určené k házení, pro ty pak musí být tento úkon co nejjednodušší a pokud možno nejpřesnější. Dále ale existují předměty, u kterých je možnost je hodit pouze vedlejší efekt toho, že je možné je držet. Zde jsou priority, aby uživatel nemusel vykonávat po sebrání předmětu žádnou další činnost na jeho držení, což usnadňuje dlouhodobou manipulaci. Jak se ukázalo, tyto priority jsou v rozporu. Nejsnazší způsob, jak něco co nejpřesněji hodit, je pomocí tlačítka na ovladači jménem trigger. To nijak neomezuje pohyb ruky jako takové a navíc je podobné skutečnému hodu objektu, kdy zakončení vrhu je učiněno rozevřením ukazováčku a palce. Vržený objekt pak získává rychlost,

jakou měl ovladač v době puštění spouště. Tento postup je vhodný pro přesné házení objektů jako jsou šipky nebo nože a navíc u něj nehrozí neúmyslné upuštění ovladače. Druhá možnost je vržení obecného předmětu tak, že jej gripem upustíme a zároveň u toho pohybujeme rukou stejně jako u minulého způsobu. Tento přístup však není zdaleka tak pohodlný ani přesný, jelikož uživatel musí provést poměrně neintuitivní sevření ruky během švihu rukou, což jde proti zvyklostem uvolnění sevření u reálného vrhu předmětem. V programu jsou rozděleny tyto objekty do dvou kategorií, primárně vrhací a ostatní objekty, u kterých je hod pouze vedlejším efektem faktu, že mohou být drženy.

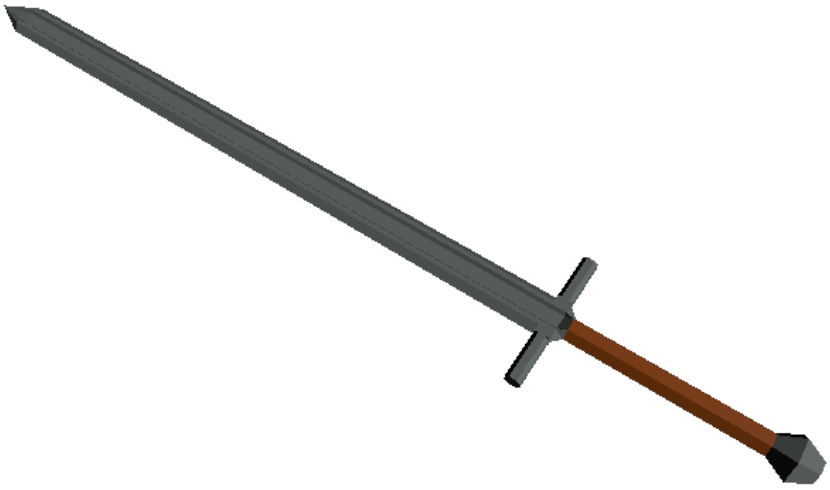
Ilustrace 12: Oštěp

Vrhání objektů je jedna z činností, která umí být ve VR velmi podobná svému reálnému ekvivalentu. Obecně platilo, že čím umí člověk lépe házet ve skutečnosti, tím dříve se adaptoval na VR a tomu odpovídala i jeho přesnost. Objevili se však uživatelé, kteří měli problémy s načasováním a často zastavili pohyb ruky dříve než pustili tlačítko na ovladači a vržený objekt nedostal žádnou rychlost. Toto chování vykazoval přibližně jeden z deseti uživatelů přičemž až na naprosté výjimky si dokázali rychle zvyknout.

Co se dá uchopit, dá se hodit

Jak již bylo zmíně v práci je snaha sjednotit ovládání a mechaniky k docílení intuitivního ovládání. Jedna z klíčových záležitostí pak je, že každý objekt ve scéně, který se dá sebrat a je možné s ním manipulovat, se pak dá upustit nebo hodit.

Ilustrace 13: Meč

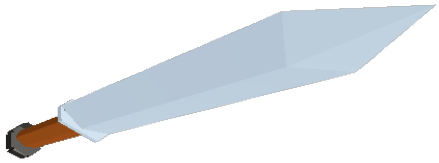


Tato mechanika umožňovala představit poměrně krátce ovládání programu, které se pak dalo aplikovat na většinu prvků v něm. Na druhou stranu, s objekty jako kuše nikdo obvykle házet nezkoušel.

Vrhací nůž

Jednoduchý příklad objektu, jehož jediným účelem je býti hozen. Technicky vzato se nůž liší od ostatních objektů pouze grafikou a také tím, že nemá žádnou vlastnost, která by se aplikovala jeho použitím (použitím v kontextu herní události). K výraznému zvýšení pohodlí je u něj však změněn způsob jak sebrání tak hození. Ovládání nože je přesunuto na spoušť jako by bylo obvykle použití objektu. Tato změna nám přináší již zmíněnou výhodu v podobnosti ovládání se skutečností, ale také nám umožňuje pohodlně držet stisk celou dobu, dokud není nůž hozen. Tato změna také umožňuje chování zmíněné v generátoru objektu, které umožňuje snadné a rychlé házení dalších nožů.

Ilustrace 14: Vrhací nůž



Platforma

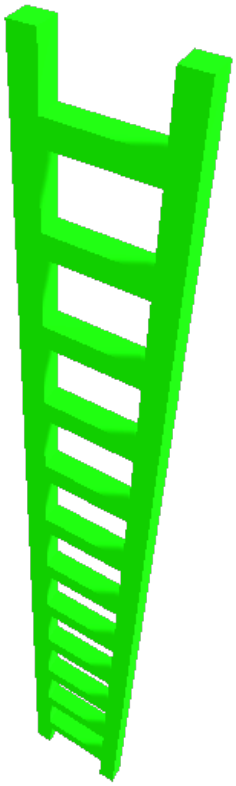
Objekt platforma je fyzickou reprezentací stejnojmenného způsobu pohybu po scéně zmíněného v předchozí kapitole. Samotný pohyb platformy nikomu z testovaných problém nedělal.

Řízení

Řízení platformy je v scéně realizováno pomocí páček nebo tlačítek. Obvykle trvalo trochu déle než si uživatel zvyknul na ovládání. Tento problém, ale pravděpodobně vychází ze špatného designu samotných páček. Jediné subjektivně nepříjemné pak bylo, když uživatel narazil do nějaké překážky (dalo by se polemizovat, jestli to potom není objektivně nepříjemné).

Šplhání

Ilustrace 15: Žebřík



Šplhání jako metoda pohybu již byla probrána v předchozí kapitole a slouží jako primární pohybová technika v největší ze scén v aplikaci. Tato technika až překvapila tím, jak naprosté většině lidí připadala zábavná (což byl také důvod, proč jsem kolem ní postavil největší scénu). Zároveň umožňovala poměrně kvalitní přesun po mapě. Jediný problém byl s mrtvými zónami, do kterých se uživatelé čas od času dostávali. Ze samotného lezení se pak nikomu špatně nedělalo. Nepříjemné bylo pak pouze selhání následované pádem. Na druhou stranu i na pád z výšky ve VR si několik lidí bez problému zvyklo.
Zde se také ukázalo, že pokud má osoba skutečný velký strach z výšek, bude mít pak strach i ve VR. (Zde jsem měl to štěstí takovou osobu sehnat a přesvědčit k otestování aplikace). Nutno dodat, že daná osoba byla první, a jediná, které se povedlo přelézt první část scény na první pokus bez pádu. Zajímavé bylo, že po prvním pádu, ve druhé části, se pak bála méně.

Teleporter a načítání scény

Ilustrace 16: Teleporter

Tyto dva objekty byly již také zmíněny v možnostech pohybu, hráč použije objekt, který teleportér reprezentuje a po chvíli a zčernání obrazovky se přenese na jiné místo. Rozdíl je pouze v tom, jestli je hráč přenesen do jiné scény nebo zůstává v téže scéně.
Kvůli přirozenějšímu přechodu dojde na nějaký, poměrně krátký čas, k vytvoření černé masky, která zakryje většinu uživatelovi vize.

Během testování ani teleportery ani načítání scény nedělaly nikomu žádné faktické problémy, jako jediný mírný zádrhel se ukázalo, že když byla doba načítání, a tím i doba zčernalé obrazovky, delší než je nutné, lidé ztráceli trpělivost a nevědomky přechod do jiné scény, nebo k jinému přenašeči rušili. Tenhle detail byl opraven zkrácením doby nutné k přesunu (doba načítání scény, jako takové, je velmi krátká, chyba byla ve zbytečně dlouhé době tmavnutí obrazovky).

Přitahování

Tato technika byla použita ze dvou důvodů. Jednak jako primární řešení mrtvé zóny, (vhodné díky nelokálnosti), ale také kvůli zjištění jakým způsobem bude takto hrubá mechanika působit na uživatele. Jak se ukázalo, nikdo neměl žádné nepříjemné vedlejší následky této metody. Jako nevýhoda tohoto přístupu se zdálo porušení základního konceptu uniformního ovládání. Jakémukoliv uživateli však stačilo jedenkrát zmínit tuto záchranou mechaniku k jejímu bezproblémovému používání. (Implementováno tak, že uživatel musí namířit na *Puller* a stisknout spoušť, kvůli velké obtížnosti míření z dálky je pak možné stiskem touchpadu zobrazit paprsek, který z namíření udělá velmi snadný úkol)

Fotoaparát

Tento objekt nemá v aplikaci zásadní roli, ani konkrétní důvod, ukazuje pouze možnost práce s kamerou a také je na něm vidět subjektivně zajímavý pohled na VR. Použití fotoaparátu vytvoří fotku, tak jak by se dalo čekat o skutečného polaroidu. Nutno dodat, že tento předmět připadal uživatelům subjektivně poměrně zajímavý.

6.5 Závěry testování

Testy z větší části následovaly očekávání vytvořené vstupní teorií, objevilo se tu však několik rozdílů. Pravděpodobně nejdůležitější závěr je, že velký strach z nevolností je s nynější technikou a alespoň rozumným návrhem, opravdu na hranici mýtu. Ani u testů, u kterých byla záměrně vytvořena poněkud těžší situace, se nevyskytovaly závažné problémy. Ano, padání dokáže být nepříjemné, ale testeři tento pocit popisovali spíše jako zvláštní a občas jako nepříjemný. Nestalo se však, že by se během testů (bráno v potaz, že testy fungovaly, jak měly) udělalo někomu špatně.

Moje osobní teorie na nepříjemnost případného pohybu tedy úzce souvisí z pozorností člověka. Znovu zde jde o informace, které má daný uživatel k dispozici, i velmi stroze udělaný pohyb, který je přesně řízený ze strany testera, který jej může kdykoliv přerušit nebo změnit, nezpůsoboval téměř žádné problémy (zde mluvím o většinovém výsledku, nikoliv absolutním). Na druhou stranu plynulý pohyb s pozvolným zrychlením, který jde proti vůli uživatele, je neovladatelný a nebo začne nečekaně, způsobuje nějaké problémy. Spojením těchto dvou modelů dohromady jsme pak schopni vyvolat u většiny uživatelů podobné reakce na stejný typ pohybu.

Celkem důležitý rozdíl působila rychlost, přičemž ta zvýrazňovala vlastnosti případného pohybu. Létaní tedy většinou nezpůsobí problém, rychlé létání může způsobovat malé problémy (spíše, ale v blízkosti objektů, pokud je let vysoko nad zemí a uživatel nemá referenční body, sama rychlost pak připadá nižší). Prudké zastavení je pak nepříjemné, zastavení nárazem do objektu je poté velmi nepříjemné. Stejně tak ztráta kontroly, pokud řízený let najednou přejde ve volný pád můžeme očekávat nepříjemnosti.

V rámci testování se také objevuje jev, že při zvyku na VR, kdy byly vjemy výrazně odlišné, než na které jsme normálně zvyklí, dochází k nutnosti opětovného zvykání si na skutečnost. Může se pak stát, že se člověku po sejmutí headsetu začne točit hlava nebo bude chvíli špatně ostřit. Tento jev nebyl příliš častý a vždy rychle odezněl.

7 Závěr

V první části práce jsem nastudoval vývojové prostředí Unity, jeho silné stránky a jeho využití k vývoji aplikace pro virtuální realitu. Dále jsem prozkoumal možnosti systému htc Vive a za pomoci několika menších experimentů, které jsem měl možnost vyzkoušet asi na dvaceti lidech, jsem si prakticky zjistil některé přednosti a omezení při vývoji pro virtuální realitu. Provedené experimenty odhalili několik zajímavých informací, hlavně co se týče nevolnosti z virtuální reality a různého působení vjemů na uživatele různých věkových a zájmových skupin.

Tato část práce pak také implementuje kostru programu spustitelnou na PC se zapojeným Vivem. Účelem této kostry bylo zatím hlavně vyzkoušet nabyté znalosti a postavit základ pro další vývoj aplikace. Vyzkoušel jsem několik různých interakcí s prostředím ve virtuální realitě a promyslel jejich využití do cílové aplikace. Testování mi také umožnilo ověřit si výkonnostní limit zařízení a případné hardwarové omezení, které bude nutné brát v potaz do výsledného programu. Tento limit se nakonec neukázal jako žádný problém, vzhledem ke grafické jednoduchosti aplikace, byla stále velká rezerva co se výkonu týče.

Finální aplikace pak obsahuje několik scén. Cílem některých scén bylo pouze otestování nějakých mechanik, jiné scény si však dávaly za úkol uživatele pobavit (tato vlastnost by také mohla být brána jako test zábavnosti). S touto aplikací bylo poté znovu provedeno testování, tentokrát už cíleněji na konkrétní mechaniky, zejména pohybové. Některé s poznatků pak byly rovnou aplikovaný na okamžité vylepšení dané aplikace jiné posloužili pouze jako podklad pro závěry v této práci.

Samotné testování pak podle mého názoru přineslo dostatečné množství informací, ze kterých bylo možné vyvést závěry.

Literatura

[1] *AR/VR investment hits $1.7 billion in last 12 months* [online]. [cit. 2017-01-09]. Dostupné z: http://www.digi-capital.com/news/2016/04/arvr-investment-hits-1-7-billion-in-last-12-months/#.WG57gS9X-Hs

[2] *Unity3D* [online]. [cit. 2017-01-09]. Dostupné z: https://unity3d.com

[3] *Unity User Manual* [online]. [cit. 2017-01-09]. Dostupné z: https://docs.unity3d.com/Manual/index.html

[4] *Unreal engine Rendering Overview* [online]. [cit. 2017-01-09]. Dostupné z: https://docs.unrealengine.com/latest/INT/Engine/Rendering/Overview/

[5] *Forward Rendering vs. Deferred Rendering* [online]. [cit. 2017-01-09]. Dostupné z: https://gamedevelopment.tutsplus.com/articles/forward-rendering-vs-deferred-rendering—gamedev-12342

[6] *Forward vs Deferred vs Forward+ Rendering with DirectX 11* [online]. [cit. 2017-01-09]. Dostupné z: http://www.3dgep.com/forward-plus/

[7] *SteamVR Plugin* [online]. [cit. 2017-01-09]. Dostupné z: https://www.assetstore.unity3d.com/en/#!/content/32647

[8] *VRTK - SteamVR Unity Toolkit* [online]. [cit. 2017-01-09]. Dostupné z: https://www.assetstore.unity3d.com/en/#!/content/64131

[9] *VRTK - Virtual Reality Toolkit* [online]. [cit. 2017-01-09]. Dostupné z: https://vrtoolkit.readme.io

[10] *Unity Asset Store* [online]. [cit. 2017-01-09]. Dostupné z: https://www.assetstore.unity3d.com/en/

[11] *A Guide to Understanding Your Peripheral Vision* [online]. [cit. 2017-01-09]. Dostupné z: http://www.eyehealthweb.com/peripheral-vision/#

[12] *What is Virtual Reality?* [online]. [cit. 2017-01-09]. Dostupné z: http://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html

[13] *The best VR headsets* [online]. [cit. 2017-01-09]. Dostupné z: https://www.wareable.com/headgear/the-best-ar-and-vr-headsets

[14] *Virtual Reality Motion Tracking* [online]. [cit. 2017-01-09]. Dostupné z: http://www.vrs.org.uk/virtual-reality-gear/motion-tracking/

[15] *Virtual Reality, Frame Rate* [online]. [cit. 2017-01-09]. Dostupné z: http://www.playin4k.co.uk/blog/files/frame-rate-is-key-to-avoiding-vr-sickness.html

[16] *How to Avoid the Effect of Motion Sickness in VR* [online]. [cit. 2017-01-09]. Dostupné z: http://vrscout.com/news/avoid-motion-sickness-developing-for-vr/

[17] *SCIENTISTS THINK THEY'VE FOUND A WAY TO ELIMINATE VIRTUAL REALITY SICKNESS* [online]. [cit. 2017-01-09]. Dostupné z: http://www.popsci.com/scientists-think-theyve-found-way-to-eliminate-virtual-reality-sickness

[18] TED. FAABORG, Alex. *Designing for virtual reality and the impact on education* [Video file]. In: . [cit. 2017-01-04]. Dostupné z: https://www.youtube.com/watch?v=DQMA5NNhN58

[19] *Virtual therapy. In: Psychology Today* [online]. [cit. 2017-01-05]. Dostupné z: https://www.psychologytoday.com/articles/199411/virtual-therapy

[20] *CAVE Automatic Virtual Environment* [online]. [cit. 2017-01-09]. Dostupné z: http://www.visbox.com/products/cave/

[21] *Virtual Reality Training Improves Operating Room Performance.* In: Annals of Surgery [online]. [cit. 2017-01-05]. Dostupné z: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1422600/

[22] *Virtual reality combat training*. In: Virtual Reality Society [online]. [cit. 2017-01-05]. Dostupné z: http://www.vrs.org.uk/virtual-reality-military/combat-training.html

[23] *Architecture's virtual shake-up.* In: BBC news [online]. [cit. 2017-01-05]. Dostupné z: http://news.bbc.co.uk/2/hi/programmes/click_online/4385006.stm

[24] *'Gone' is a VR thriller.* In: Engadget [online]. [cit. 2017-01-05]. Dostupné z: https://www.engadget.com/2015/12/04/gone-vr-thriller/

[25] *Virtual Reality Gaming* [online]. [cit. 2017-01-09]. Dostupné z: http://www.game.co.uk/webapp/wcs/stores/servlet/HubArticleView?hubId=663753&articleId=663754&storeId=10151

[26] *What is Virtual Reality Gaming* [online]. [cit. 2017-01-09]. Dostupné z: http://www.vrs.org.uk/virtual-reality-games/what-is-vr-gaming.html

[27] *Virtuix Omni* [online]. [cit. 2017-05-18]. Dostupné z: http://www.virtuix.com

[28] *Infinadeck* [online]. [cit. 2017-05-18]. Dostupné z: http://www.infinadeck.com

[29] LOMBARD, Matthew a Theresa DITTON. At the Heart of It All: The Concept of Presence. *Journal of Computer-Mediated Communication* [online]. 1997, 3(2), 0-0 [cit. 2017-05-11]. DOI: 10.1111/j.1083-6101.1997.tb00072.x. ISSN 10836101. Dostupné z: http://doi.wiley.com/10.1111/j.1083-6101.1997.tb00072.x

[30] Imerze. *Game art* [online]. [cit. 2017-05-11]. Dostupné z: http://cas.famu.cz/gameart/page.php?page=7

[31] GRIMSHAW, Mark. *The Oxford handbook of virtuality*. New York: Oxford University Press, 2014. ISBN 9780199826162.

[32] The Psychology of Immersion in Video Games. *The Psychology of Video Games* [online]. [cit. 2017-05-11]. Dostupné z: http://www.psychologyofgames.com/2010/07/the-psychology-of-immersion-in-video-games/

[33] Swapping your body becomes a virtual reality. *New Scientist* [online]. [cit. 2017-05- 11]. Dostupné z: https://www.newscientist.com/article/dn16180-swapping-your-body-becomes-a-virtual-reality/

[34] Brain Sees Tools as Extensions of Body. *Live science* [online]. [cit. 2017-05-11]. Dostupné z: http://www.livescience.com/9664-brain-sees-tools-extensions-body.html

[35] DOUG A. BOWMAN .. [ET AL.]. *3d user interfaces: theory and practice*. Pbk. ed. 2005. S.l.: Addison-Wesley Educationa, 2014. ISBN 9780321980045.

[36] R. BUSS, Samuel. *Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods*. San Diego La Jolla, 2009. Department of Mathematics University of California.

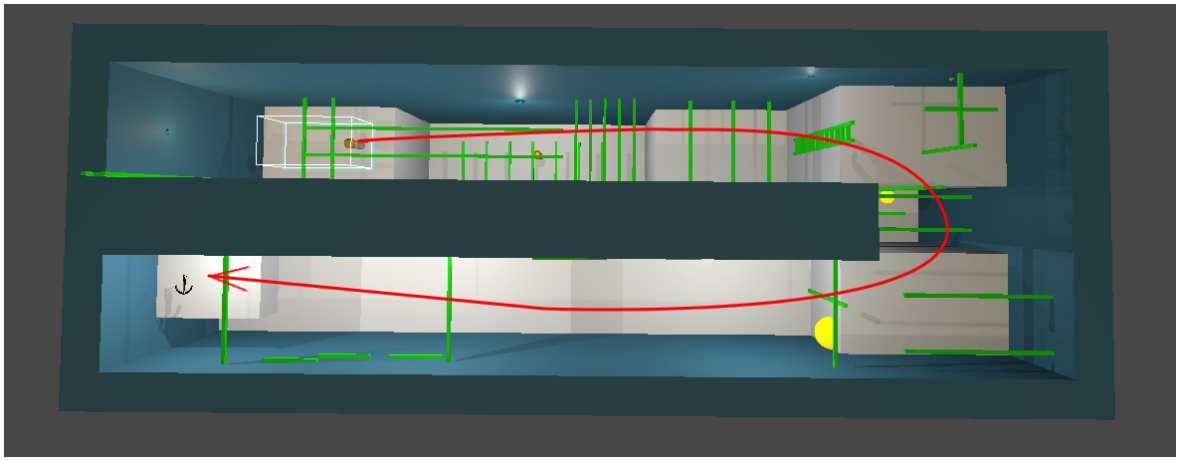
Manuál

Většina nutného ovládání je již popsána v rámci textu diplomové práce, zde nicméně zopakuji základní aspekty a popíšu způsob postupu největší ze scén obsažených v aplikaci.

Základní interakce probíhá pomocí tlačítka "grab", které je umístěno po bocích ovladače a odpovídá stisku celé dlaně. Toto tlačítko je pak použito jak při šplhání tak při sebrání kuše ze země.
Použití věci, což v této scéně odpovídá použití teleportéru nebo vystřelení z kuše, odpovídá stisku tlačítka "trigger" neboli spouště. Toto tlačítko je umístěné na spodku ovladače, na kterém se při běžném držení nachází ukazováček. K otevření scény dojde vložením ovladače do objektu reprezentujícího danou scénu (v případě scény se šplháním je to žebřík) a stiskem spouště. V případě teleportéru a načtení scény je třeba spoušť chvíli držet, jelikož teleportování chvíli trvá (přibližně jednu vteřinu). Cesta scénou odpovídá šipce na obrázku níže Ilustrace 17: Mapa scény. Co se samotného šplhání týče, nehybné objekty, které toto umožňují jsou znázorněny zelenou barvou. Šplhání je pak provedeno vložením ovladače to objektu, stisknutím tlačítka "grab" a přitažením se k nebo od objektu.
Jako záchranná mechanika se v této scéně vyskytují "přitahovače", které vypadají jako velké zlaté sféry. Po namíření ovladače na tuto sféru pak tlačítkem použíti (spoušť) dojde k přitažení uživatele k cílovému body. Pro namíření je třeba zapnout paprsek z ovladače. Zapnutí se provede stiskem touchpadu na vrchní části ovladače, obvykle pod pozicí palce při standardním držení.

Kuše na konci scény je pak cílem bez toho, že by sama měla výraznější roli, šipky které zní lze střílet však můžou sloužit také jako nehybný objekt pro šplhání.

Ilustrace 17: Mapa scény

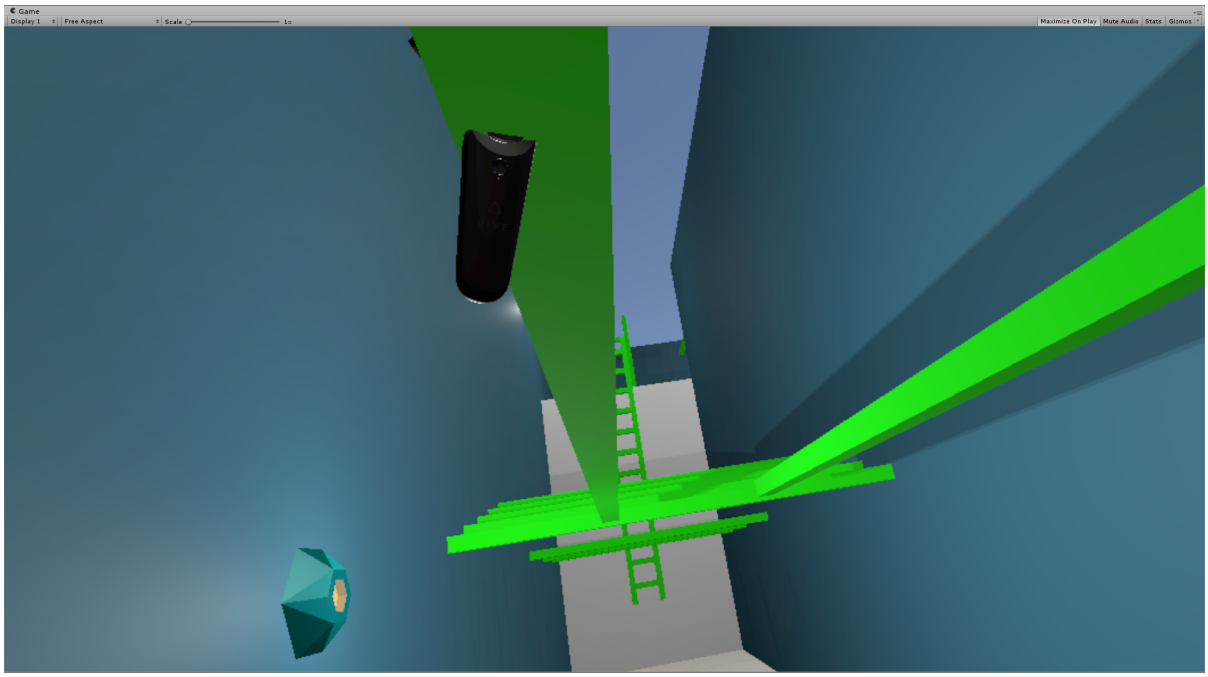


Obsah DVD

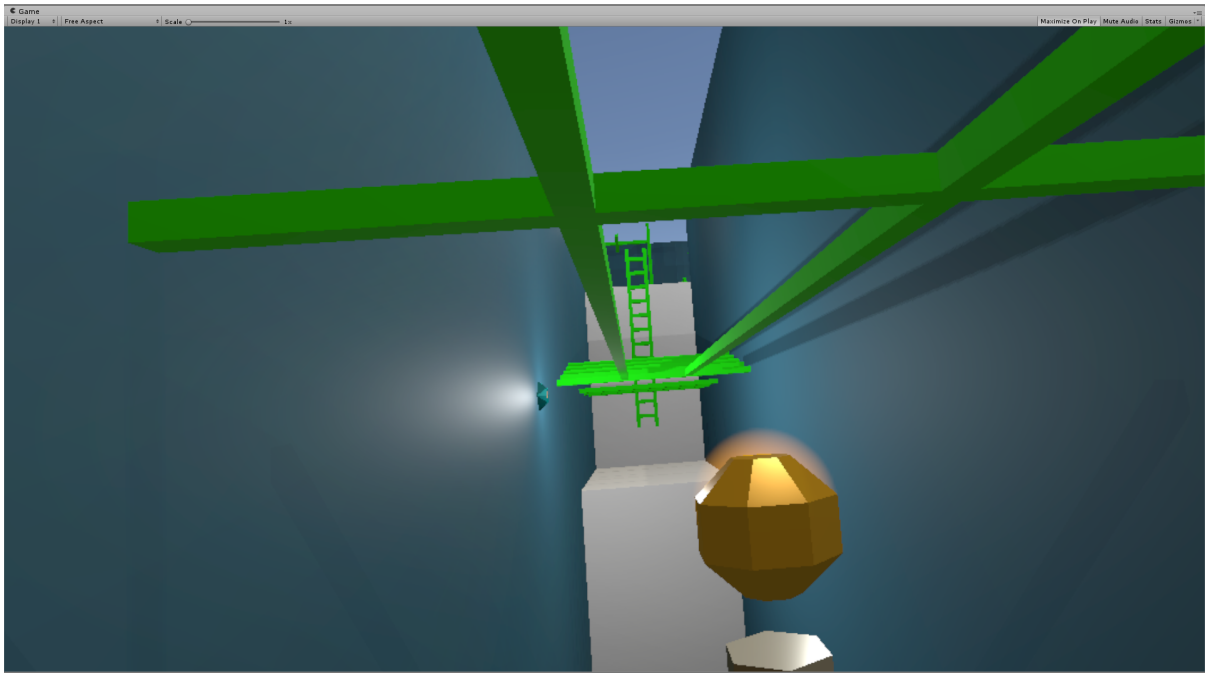
- Celá složka projektu (stejná jako obsah repozitáře)
- Build aplikace ve formátu .exe
- Natočené video obsahující záznam z projití celé aplikace
- Text diplomové práce ve formátech .odt a .pdf
- Manuál, stejný jako je součástí tohoto textu
- Obsah DVD, tento text

Přílohy

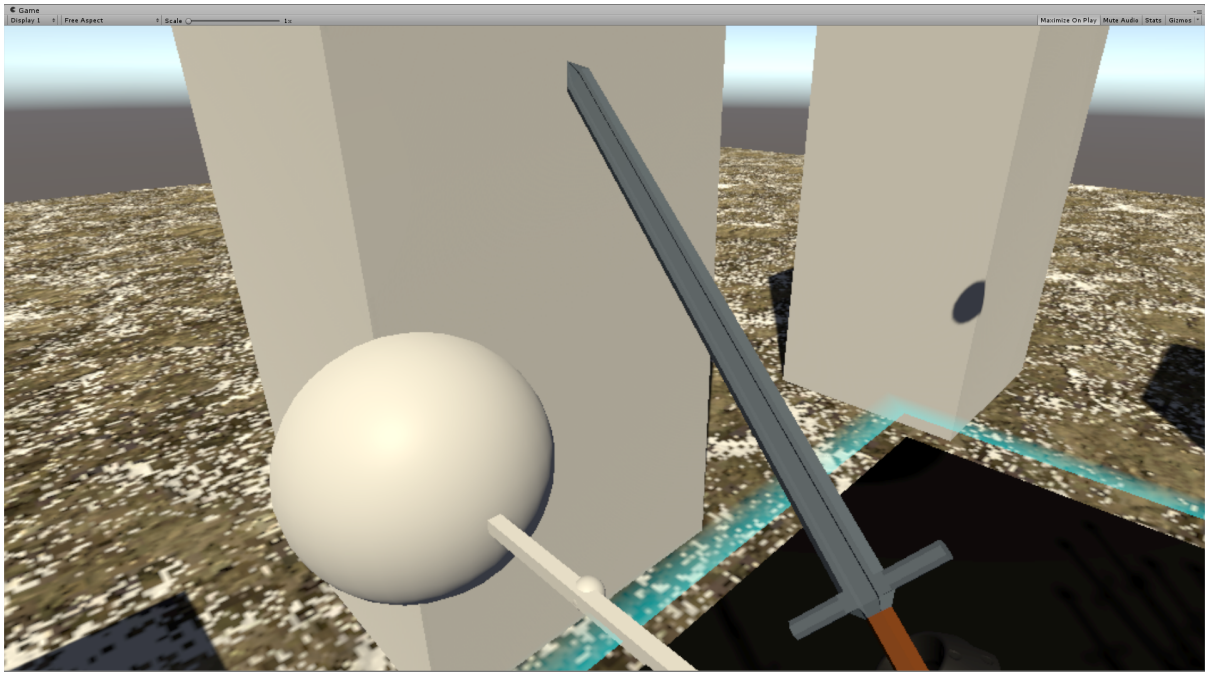
Šplhací scéna, šplhání



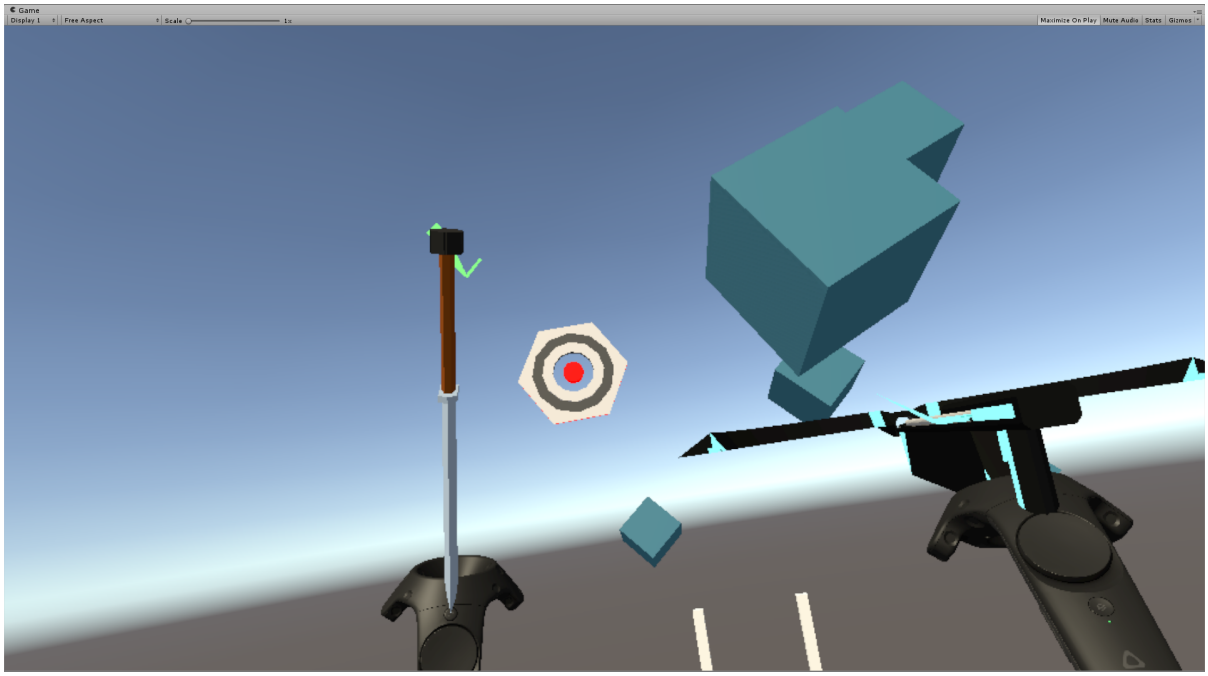
Šplhací scéna, teleportér



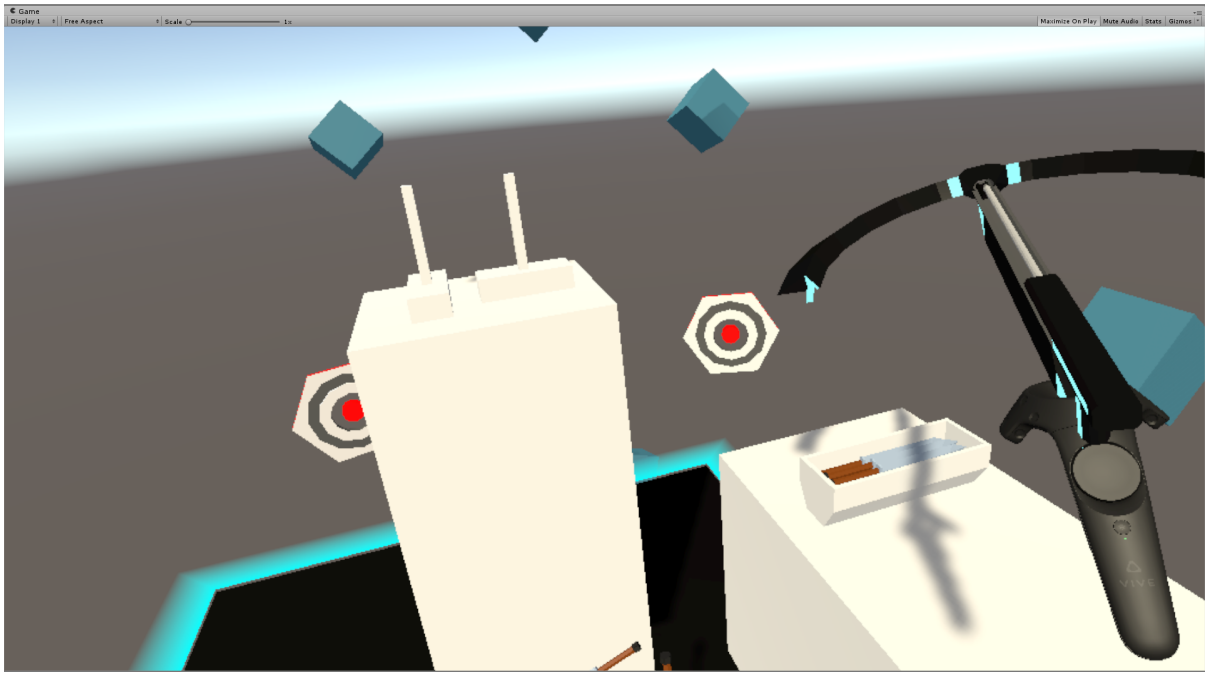
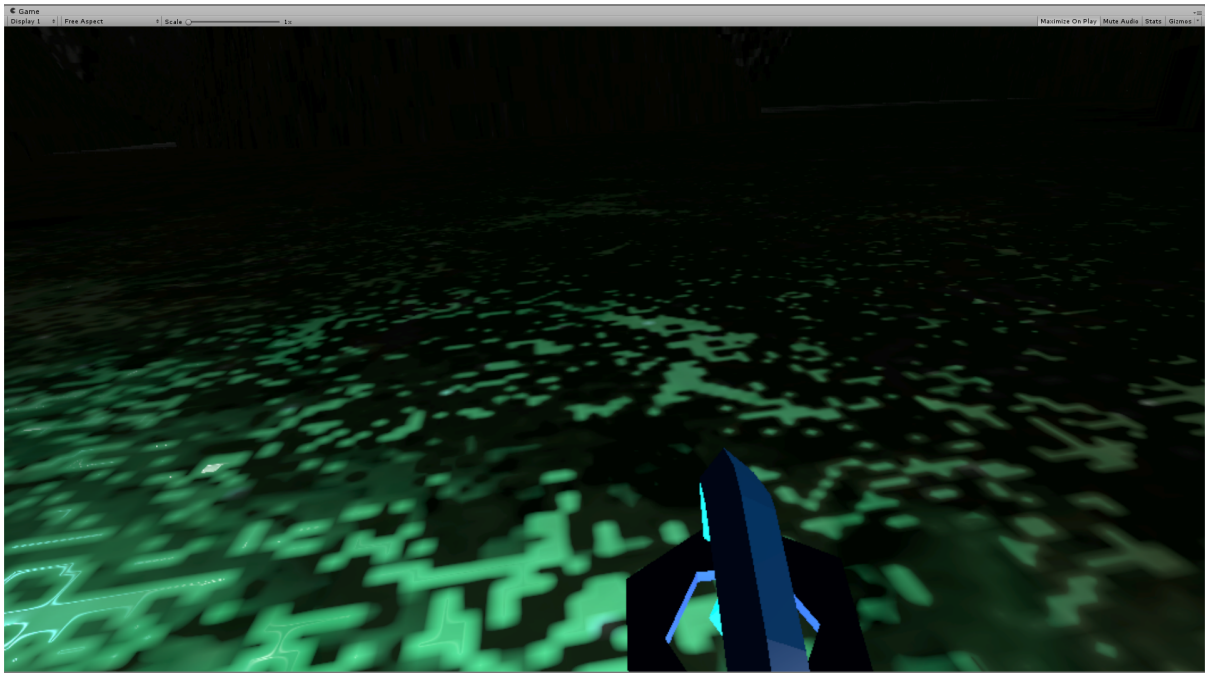
Bojová scéna



Střelnice, vrhací nůž, kuše



Střelnice, kuše, ovládání platformy



Let temnou scénou