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

INOVATIVNÍ HERNÍ VR DEMO V UE4

VR INNOVATIVE GAME DEMO IN UE4

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JOSEF DOLEŽAL**
AUTHOR

VEDOUCÍ PRÁCE **Ing. TOMÁŠ MILET,**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání bakalářské práce

Student:	**Doležal Josef**
Program:	Informační technologie
Název:	**Inovativní herní VR demo v UE4**
	VR Innovative Game Demo in UE4
Kategorie:	Počítačová grafika

Zadání:

1. Nastudujte Unreal Engine 4 a grafické efekty.
2. Navrhněte inovativní herní mechaniky a interaktivní aplikaci, kde mechaniky demonstrujete.
3. Naimplementujte navrženou aplikaci a vytvořte sadu úrovní.
4. Zhodnoťte a proměřte.
5. Vytvořte demonstrační video.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce:	**Milet Tomáš, Ing.**
Vedoucí ústavu:	Černocký Jan, doc. Dr. Ing.
Datum zadání:	1. listopadu 2018
Datum odevzdání:	15. května 2019
Datum schválení:	13. května 2019

Abstrakt

Cílem práce je vytvořit demo aplikaci pro virtuální realitu s použitím Unreal engine 4. Hlavními mechanikami vytvořenými v této práci jsou gravitační pole, simulující vlastní gravitaci a zrcadlové teleportály schopné teleportovat hráče na stejné, nebo odlišné místo na které se teleportovali předtím. Vytvořené prefabrikáty jsou dále volně použitelné v ostatních projektech v Unreal engine 4

Abstract

The task of this bachelor thesis is to create a demo application for virtual reality using Unreal engine 4. Main mechanics are gravity sfere, simulating custom gravity and mirror portals with an ability to teleport a player to a previous or any other destination. All prefabs are free to be used in another Unreal engine 4 projects.

Klíčová slova

Unreal Engine, hra, virtuální realita, VIVE, Blueprint, gravitace, portál

Keywords

Unreal Engine, game, virtual reality, VIVE, Blueprint, gravity, portal

Citace

DOLEŽAL, Josef. *Inovativní herní VR demo v UE4*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Milet,

Inovativní herní VR demo v UE4

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana inženýra Tomáše Mileta Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Josef Doležal
13. května 2019

Poděkování

Tímto bych chtěl poděkovat panu inženýru Miletovi za jeho cenné rady, inspiraci a celkové vedení této práce.

Obsah

1 Úvod **3**

2 Virtuální realita **4**
2.1 Co je to virtuální realita 4
2.2 Technologie 5
2.2.1 VIVE HMD 5
2.2.2 Kontroléry 6
2.2.3 Základové stanice 6
2.3 Vidění ve VR 7
2.3.1 Geometrie 8
2.3.2 Renderování 8
2.4 Využití 9

3 Unreal Engine 4 **12**
3.1 Fyzika 12
3.2 Blueprint 12
3.3 Typy, funkce, aj. 13

4 O hře **14**

5 Prefabrikáty **16**
5.1 Zrcadlový portál 16
5.1.1 Základní zrcadlo 16
5.1.2 Dvou lokační zrcadlo 16
5.1.3 Zrcadlo pro přenos objektů 17
5.1.4 Textura pro zrcadlový portál 17
5.2 Gravitační pole 18
5.2.1 Složení 18
5.2.2 Implementace 19
5.3 Fadebox 20
5.3.1 Složení 20
5.3.2 Implementace 20
5.4 Tlačítka 20
5.4.1 Složení 20
5.4.2 Implementace 21

6 Úrovně **22**
6.1 Level 01 22

6.1.1 Rozložení . . . 22
6.1.2 Průchod-návod na úroveň . . . 22
6.2 Level 02 . . . 23
6.2.1 Rozložení . . . 23
6.2.2 Průchod-návod na úroveň . . . 23
6.3 Level 03 . . . 23
6.3.1 Rozložení . . . 23
6.3.2 Průchod-návod na úroveň . . . 24
6.4 Level 1 . . . 24
6.4.1 Rozložení . . . 25
6.4.2 Průchod-návod na úroveň . . . 25
6.5 Level 2 . . . 25
6.5.1 Rozložení . . . 25
6.5.2 Průchod-návod na úroveň . . . 26

7 Testování **27**

8 Závěr **28**

Literatura **29**

A Obsah přiložené SD karty **30**

B Návod na práci s aplikací **31**

C Obrázky **32**

Kapitola 1

Úvod

V dnešní době se hry stávají běžnou součástí denního života, a tak vzniká nepřeberné množství her, mířených na co největší skupinu lidí s cílem co největšího zisku. Vytvářet inovativní produkt, který je odlišný od všeho co je na trhu je téměř nemožné a lze tento trend pozorovat u mnoha her, kdy se tituly opakují a vypadají jako vy byli zkopírovány a vloženy s jiným grafickým kabátem a zasazením do jiného prostředí. Do toho všeho dění vstoupila nová možnost a tou je virtuální realita, která nabídla nový způsob jakým se lze do světa her ponořit. Ovšem trh je neúprosný a tak se i pro tuto platformu začalo hrnout plno titulů, které měly, či neměly úspěch. Obecně se drží trend, že hra musí být akční, rychlá a ne příliš těžká, aby padla na milost co nejvíce hráčům. Logické hry začaly pomalu upadat a dny slávy Portálu 2 jsou téměř ty tam a zrovna nový vítr jako virtuální realita by mohla být pro tento typ her naprosto dokonalá.

Zadáním této bakalářské práce je vytvořit demo aplikaci pro virtuální realitu s využitím Unreal engine 4. Zvolená aplikace je logická hra, která manipuluje s gravitací a používá teleportační zrcadla. Dále všechny prefabrikáty vytvořené v této práci budou znovupoužitelné pro jiné Unreal engine 4 projekty. Zvláště pak gravitační pole, protože gravitace v Unreal engine funguje pouze na ose ose Z s tím, že lze její hodnotu nastavit jak v kladných tak i v záporných hodnotách.

Kapitola 2

Virtuální realita

První zmínka o virtuální realitě se objevila v roce 1965 v článku "The Ultimate Display"od Ivana Suterlanda, kde se zmiňuje, že display, který tvoří zrcadlo zobrazující matematický svět, který je uložený v paměti počítače, by měl působit na co nejvíce smyslů [8]. V této době displeje byly schopny zobrazovat informace, které působily na lidský zrak a byly schopné vydávat audio signál, který působil na lidský sluch. Vizí pana Suterlanda byla možnost, že by displej dokázal více, jako například práce s pozicí uživatele, nebo simulovat síly působící na uživatele (například přetížení při letu letadlem), nebo aby interakce s počítačem byla podle polohy rukou, nohou, nebo očí. Na konci článku je uvedeno, že takový displeje by nemusel být nutně monitor, ale ku příkladu celá místnost.

V dnešní době je více možností a je možné tyto věci řešit, a už existují podobné displeje, které dokážou výše uvedené věci.

2.1 Co je to virtuální realita

K pochopení pojmu virtuální realita, je zapotřebí vědět co znamenají její klíčová slova a to virtuální a realita. Slovo virtuální by šlo označit za nehmatatelné a vytvořené v paměti nějakého počítače. Pro slovo realita nastává větší problém. Co je to realita? Jak lze určit, že je něco reálné a nereálné.Na příklad při procházce do lesa, kdy na cestě uvidím divoké prase. Je to prase reálné, když ho vidím? Začne být reálné, když ho uslyším a nebo začne být reálné, až se proti mě rozeběhne a narazí do mě. Prase ovšem nemusí být reálné a může to být pouze výplod fantazie. S přesností nelze říci co slovo realita znamená, z technického hlediska ho ani nelze definovat a jiné znění je lepší ponechat na filozofech a lidech zaobírajících se touto tématikou.

S pomocí klíčových slov nelze přesně definovat pojem virtuální realita. Existuje i jiná definice a tu v roce 1989 zvedl Jaron Lanier, zakladatel firmy VPL Research a výzkumník v oblasti virtuální reality, který to tento pojem chápal jako, „počítačem vytvořené interaktivní trojrozměrné prostředí, do něhož se člověk totálně ponoří“ jak je uvedeno v knize Reálně o virtuální realitě [7].

K lepšímu porozumění co virtuální realita obnáší, je dobré zmínit pohled autorů knihy Reálně o virtuální realitě [7], kteří v tomto díle popsali virtuální realitu ve třech stupních pasivní, aktivní a interaktivní.

- Pasivní je stupeň, ve kterém se nacházíme v prostředí, ve kterém můžeme věci okolo nás pozorovat, poslouchat a snad i hmatat, ale na takové prostředí nemáme žádný vliv. Průchod tímto prostředím lze simulovat změnou tohoto prostředí. Dalo by se to přirovnat k místnosti, kdy uživatel stojí uprostřed a může vidět nábytek, nebo slyšet hluk za oknem.

- Aktivní je stupeň, ve kterém přibývá možnost pohybu. Je tedy možné chodit po místnosti, podívat se z okna co způsobuje daný hluk, nebo místnost opustit.

- Interaktivní je stupeň, ve kterém je krom pohybu také možnost prostředí ovlivňovat. Je možné si otevřít okno, aby se dalo lépe podívat na zdroj hluku, nebo je možné volně přesunout židli, či uchopit knihu z poličky.

Jak je vidět nejintenzivnější stupeň je interaktivní, kdy je uživatel téměř neomezený v tom co lze v dané simulaci dělat, ovšem tento stupeň vyžaduje složitější technologie než stupeň pasivní, který lze vnímat jako speciální trojrozměrný film.

2.2 Technologie

Dnes může mít přístup k virtuální realitě každý. Stačí si pořídit přístroj, který umožní přístup do tohoto světa a pořizovací cena není nikterak ohromná. na trhu se můžeme setkat s několika produkty, jedním z nich je také zařízení VIVE od firmy HTC. Toto zařízení nabízí HMD, dva kontroléry, dvě základní stanice pro volný pohyb ve virtuální realitě a linkbox sloužící pro propojení HMD a počítače. VIVE také nabízí možnost bezdrátového připojení k počítači za použiti bezdrátového adaptéru, který se připevní k HMD. Dalšími prvky na trhu jsou brýle Hololens od firmy Microsoft, které nepotřebují žádné kontroléry, ani základní stanice pro vymezení prostoru. Brýle Hololens vytváří projekci aplikace přímo do místnosti uživatele, nejde tedy přímo o virtuální realitu, ale pouze o rozšířenou realitu. Konkurencí pro VIVE je zařízení Oculus Rift a nově Oculus Quest. Ty nabízejí podobnou sestavu jako VIVE, což znamená vlastní HMD, dva kontroléry a dvě základové stanice. V neposlední řadě firma Valve vytváří vlastní HMD pro virtuální realitu zvanou Valve INDEX, kdy se jedná o produkt podobný, ne-li téměř identický VIVE a Oculus Rift s rozdílem, že nabízí větší dosah a lepší zorné pole než samotné VIVE [9]. Pro vlastníky herních konzolí je na trhu možnost využiti zařízení PlayStation VR. To potřebuje ke svému fungování herní konzoly PlayStation 4 se speciální kamerou, která je k dostání v sadě pro PlayStation VR.

2.2.1 VIVE HMD

Firma HTC momentálně nabízí tři druhy HMD zařízení. Prvním je klasické zařízení VIVE, které dává možnost využití virtuální reality široké veřejnosti. Dále je tu zařízení VIVE Pro, které zvyšuje kvalitu obrazu zvětšením hodnoty PPI displeje. Poslední zařízení je VIVE Focus, které slouží jako all-in zařízení, což znamená, že ke svému fungováni nepotřebuje žádný připojený počítač a ani žádné základové stanice. Dále firma HTC oznámila nové zařízení VIVE Cosmos, které by mělo nabízet špičkový grafický zážitek a jako VIVE Focus nepotřebuje žádné základové stanice [1].

Obrázek 2.1: HTC VIVE HMD (zdroj: `https://realidadvirtualcastellon.com/wp-content/uploads/2017/06/99HAHZ046-00.png`)

2.2.2 Kontroléry

Kontroléry slouží jako vstupní zařízení pro virtuální realitu, kdy pomocí nich interagujeme se spuštěnou aplikací. Každý HMD má specifické kontroléry[1].

Obrázek 2.2: HTC VIVE kontrolér (zdroj:`https://www.vive.com/media/filer_public/ac/85/ac8560e4-8d7f-42b6-9394-8fa6d5064b4e/controller_01.jpg`)

2.2.3 Základové stanice

Základové stanice slouží pro vymezeni prostoru ve kterém se člověk může pohybovat a kde se snímá pohyb HMD a kontrolérů. Základové stanice využívají zařízení HTC VIVE, HTC VIVE Pro a PlayStation VR[2].

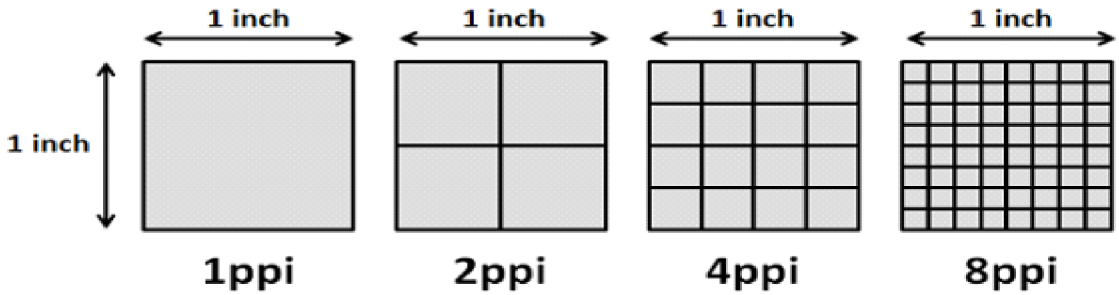
[1] výjimkou VIVE a VIVE Pro, které používají identické kontroléry a zařízení Hololens je možné ovládá pomocí gest

[2] pro tento účel souží kamera pro PlayStation 4

Zařízení VIVE používá základové stanice SteamVR ve verzi 1.0 nebo 2.0. Pro použití jsou potřeba dvě stanice které jsou umístěny naproti sobě a tím vytvoří prostor pro sledování kontroléru a HMD. Verze 1.0 nabízí možnost umístění stanic ve vzdálenosti až 5 metrů a s 120° zorným polem. Stanice ve verzi 2.0 mají maximální vzdálenost 6 metrů a zorným polem 150° [1].

2.3 Vidění ve VR

Pro vidění ve virtuální realitě slouží displej, který je umístěný v určité vzdálenosti od očí uživatele.

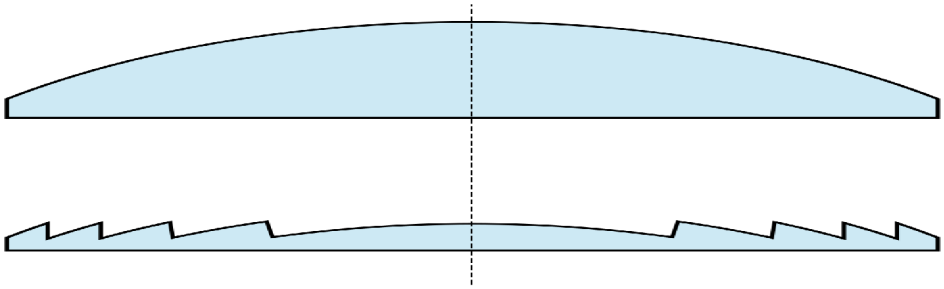


Obrázek 2.3: Počet pixelů v jednom čtverečním palci v závislosti na velikosti PPI./https://www.giffgaff.com/blog/pixel-density-how-to-calculate-ppi/pixel-density-explained.png/

Důležitým faktorem pro dosažení téměř dokonalé reality z hlediska obrazu je potřebná hustota pixelů, která nám určí počet pixelů v jednom palci čtverečním[3]. Jednotka hustoty pixelů se udává v PPI (pixel per inch) viz obrázek 2.3. Tato hodnota je důležitá proto, aby nevznikal aliasing, tedy aby diagonální hrany zůstávaly ostré i při oddalování vzdálenosti. Ideální hodnota hustoty pixelů je závislá na vzdálenosti displeje od očí uživatele. Přičemž čím blíže je displej, roste požadovaná hodnota hustoty pixelů. Příklad uvedený v online knize Virtual reality od Steven M. LaValle [5] názorně ukazuje výpočet požadované hodnoty pro display chytrého telefonu umístění 12 palců od očí uživatele, kdy požadovaná minimální hodnota je kolem 287 PPI. Pro VR headsety se ovšem používají čočky, které umožní displej přiblížit mnohem více k očím uživatele, aby se docílilo navození pocitu, že uživatel je součástí scény. To ovšem sebou přináší větší náročnost na displej který by musel mít nejméně 2292 PPI, což je s momentální technologií nedosažitelné. Částečně se tento problém snaží výrobci zmírnit použitím členěných čoček, kdy dosahují větší ostrosti na úkor ztráty světla [4]. Tyto čočky jsou také lehčí a užší něž klasické čočky, jak lze vidět na obrázku 2.4 [11].

Dalším faktorem je snímková frekvence, která je zodpovědná za plynulý obraz při pohybu. Aby člověk vnímal obraz jako plynulý musí být snímková frekvence výší než 23 snímků za vteřinu. Pro televizi je snímková frekvence mezi 25 až 30 snímky za sekundu. Pro počítačové hry je ideální frekvence mezi 60 až 120 snímky za vteřinu. Pro 360° video je ideální frekvence 60 snímků za vteřinu. Pro videa ve virtuální realitě by měla být stejná jako pro 360° video, neboť se jedná o stejnou technologii. Problém nastává při určování snímkové frekvence pro hru ve virtuální realitě, kdy se přidává pohyb po prostoru. Mnoho hráčů ta-

[3] 1 palec čtvereční je roven 6,4516 centimetrům čtverečním



Obrázek 2.4: Klasická čočka v porovnání s Fresnelovou čočkou použitou v HTC Vive (zdroj: `https://cs.wikipedia.org/wiki/Fresnelova_čočka`)

kovýchto her je přesvědčené o hodnotě 90 snímků za vteřinu. Tato hodnota je také uváděná ve specifikacích HTC VIVE.

2.3.1 Geometrie

3D prostoje je definován třemi osami, které určují délku, šířku a výšku. Každý objekt v tomto prostoru lze popsat body, které mají různé hodnoty souřadnic v zavilosti jak daleko se nacházejí od počátku souřadného systému. Reálně je těchto bodů nekonečně mnoho, což pro počítačový svět znamená problém, neboť zde se vždy pracuje s konečným počtem hodnot. Řešením tohoto problému je utváření takzvaných primitiv[4]. Nejvíce užitečné primitivum pro tento účel je trojúhelník, který se skládá spojením tří bodů v souřadném systému 2.1. Tyto hodnoty lze jednoduše uložit jako vektor hodnot. Spojením několika takovýchto trojúhelníků, lze vytvořit síť modelu. S větším počtem trojúhelníků narůstá paměťová náročnost, ale také roste kvalita modelu jak je možné vidět na obrázku 2.5.

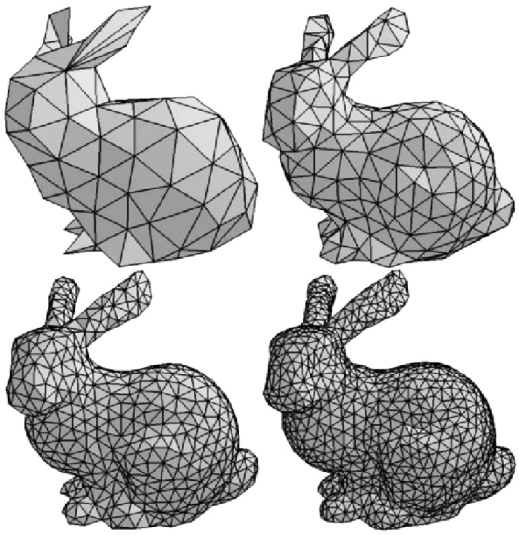
$$([x_1, y_1, z_1], [x_2, y_2, z_2], [x_3, y_3, z_3]) \tag{2.1}$$

Další nespornou výhodu trojúhelníku je malá náročnost na výpočetní algoritmy než by tomu bylo u jiných primitiv jako například u křivky.

2.3.2 Renderování

Za předpokladu použití trojúhelníku jako primitiva, můžeme použít pro renderování proces zvaný rasterizace. Rasterizace je proces převodu 3D scény (herního světa) do 2D scény (obrazovka displeje). Při tomto přechodu se ovšem ztratí jedna důležitá informace a tou je vzdálenost objektů od pozorovatele. Řešením tohoto problému je uložení si těchto hodnot do takzvaného Z-bufferu. Tyto uložené hodnoty poslouží při stínování scény, kdy z hodnot uložených v Z-bufferu a pozice zdroje světla se vypočítá, jaký stín vrhá daný objekt na ostatní objekty ve scéně. Dále se hodnoty využijí k určení které objekty není nutné vykreslovat, neboť z daného pohledu jsou schovány za jiným objektem. Tím se nesporně sníží náročnost na výpočet a zvýšení výkonu aplikace. Rasterizace také napomůže ulehčení zátěži CPU neboť všechny výpočty probíhají na GPU a moderní grafické karty jsou na tyto výpočty stavěny a optimalizovány [5].

[4] jednoduchý objekt jako například přímka, křivka, nebo mnohoúhelník



Obrázek 2.5: Kvalita modelu v závislosti na počtu trojúhelníků (Zdroj: `https://cathyatseneca.gitbooks.io/3d-modelling-for-programmers/content/3ds_max_basics/3d_ representation.html`

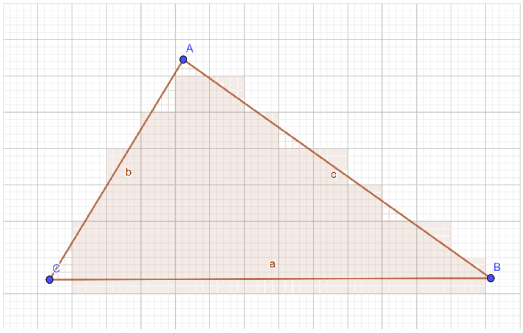
Rasterizace sebou ovšem přináší i záporné věci a jednou z nich je aliasing. Ten způsobuje zubaté hrany na přechodu a snižuje tak kvalitu obrazu, navíc ve virtuální realitě se pozice pozorovatele neustále mění, neboť HMD, které je na hlavě uživatele, je neustále v pohybu. Problém je řešitelný pomocí technologie zvané antialiasing v případě virtuální reality se používá 4xMSAA[5] nebo 8xMSAA, který vykreslí obraz několikanásobně a pak s minimálním posunutím tyto obrazy prolne. Bez použití antialiasingu by byl obraz nepoužitelný a multisampling není přílišná zátěž pro dostatečný výkon aplikace [10].

Další problém způsobují samotné čočky. Ty způsobuji sférická aberace, což znamená že paprsky jdoucí od okraje čočky se lámou dříve než paprsky jsoucí ze středu [12]. Tento efekt je minimální, když je zornice oka přímo naproti středu čočky. Bohužel zařízení jako HTC VIVE má tyto čočky pevně usazené a s jejich pozicí nelze hýbat. Proto se tento problém řeší pomocí transformace obrazu, která obraz na okrajích rozostří a vytvoří tubusový obraz. Jak je možné vidět na obrázku 2.7 výsledný obraz zobrazený na monitoru je odlišný od obrazu zobrazeného na displeji HMD. Moderní grafické procesory v sobě mají zabudované funkce, které sníží náročnost transformační operace.

2.4 Využití

Virtuální realita má obrovský potenciál využití a to nejen pro zábavu. Možnosti, které nabízí by byli skvělé pro studenty technických oborů, kdy by byla možnost prohlížení strojních modelů ze všech úhlů, a možnost dostat se až do jádra mechanizmu. Další uplatnění by bylo v architektuře, kdy návrhář může svůj výtvor jednoduše procházet, a vidět věci, které by na papíře a ani na monitoru nebyly možné, nebo zbytečně složité. V neposlední řadě studenti medicíny by měli jednoduší studium, kdy pohodlně a kdykoliv mohou studovat lidskou anatomii, nebo simulovat operaci.

[5] Multisample antialiasing



Obrázek 2.6: Aliasing při rasterizaci trojúhelníku (nalevo) Porovnání několikanásobného multisample antialiasingu (napravo, zdroj: https://mynameismjp.files.wordpress.com/2012/10/msaa-edges.png)

Obrázek 2.7: Obraz zobrazený na monitoru vs obraz zobrazený na displeji HMD (Zdroj: `https://medium.com/aol-alpha/how-to-design-vr-skyboxes-d460e9eb5a75`)

Dalším aspektem je boření bariér, kdy se lidé mohou setkávat volně s ostatními lidmi z různých koutů planety ve formě určité interakce. Může jít o nové způsoby tvoření konferencí, kdy se klasické videokonference nahradí těmi ve virtuální realitě. To otevře další možnosti jak prezentovat svoje výrobky, kdy kupující uvidí virtuální interaktivní model produktu, který umožní lepší demonstraci. Příkladem v automobilovém průmyslu, prezentace nového vozidla. Nemusí se však jednat jen o konference, ale i o volno časovou interakci. Vhodným příkladem je existující aplikace VRChat, která je volně přístupná a od svého vydání v roce 2017 si vytvořila obrovskou komunitu lidí. Zde mohou lidé volně komunikovat a interagovat s ostatními uživateli této aplikace.

Pro stoprocentní využití bude ovšem zapotřebí udělat posun v technologii a to už hardwarového rázu tak i softwarového s velkým zaměřením se na bezpečnost.

Obrázek 2.8: Aplikace VRChat (zdroj:`https://apkpure.com/es/vr-chat-game-anime-avatars/com.vravatars.anime`)

Kapitola 3

Unreal Engine 4

Unreal engine 4 je open source herní engine, vytvořený firmou Epic Games. Celý engine je napsán v jazyce C++ a je designován pro nejlepší výkon. Nabízí profilování ať už CPU tak GPU. Podporuje většinu primárních platforem ať už se jedná o všechny operační systém na počítačích, tak i konzole a mobilní zařízení. V neposlední řadě nabízí dech beroucí grafiku a propracované vnitřní systémy, jako je například široká možnost tvorby vozidel, nebo nástroje pro práci s oblečením.

3.1 Fyzika

Fyziku v Unreal enginu obstařává PhysX 3.3 engine od firmy NVIDIA. Ten se stará o simulace a výpočty pro všechny objekty v scéně. jedná se o komplexní engine, který počítá s časem, prostorem, energií, silami a momenty. Počítání probíhá v trojrozměrném prostoru kdy vektorové veličiny jsou vyjádřeny jako vektory s hodnotami v plovoucí desetinné čárce a hodnota jako je třeba čas se uvádí v podobě celých čísel. Objekty na něž jsou aplikovány fyzikální zákony a detekování kolize se nazývají Actors [6]. Toto označení se přenáší i do Unreal engine, kde vyjadřuje abstrakci chování určité komponenty (UActorComponent).

Komponenty jsou speciální typ objektu kdy Actors těchto komponent jsou spolu navzájem vázány. Pomocí toho lze jednoduše definovat chování celého objektu ve scéně. Samotná komponenta se skládá ze tří tříd. První je již zmiňovaná Actor komponenta (UActorComponent), která má na starosti popis chování. Dále je scénová komponenta (USceneComponenet), která má za učel tvořit poziční vyjádření komponenty. Poslední je Primitivní komponenta (UPrimitiveComponent), ta má za účel geometrickou reprezentaci a tím i vytváření kolizí [3].

3.2 Blueprint

Unreal enngine nabízí takvaný Blueprints Visual Scripting, což je skriptovací systém, který umožňuje přímo tvořit herní elementy v samotném editoru. Programování probíhá propojováním uzlů mezi sebou a tím tvoření grafů, které se vykonávají na základě nějaké události, která s daným grafem spojena. Dále je zde možnost práce s proměnnými, vytváření vlastních funkcí a maker, které lze použít jako uzel v dalších blueprintech [2].

Použití tohoto systému je velice široké. Lze ho využít pro vytvoření základu hry, kdy se zadefinuje potřebná kostra a samotná funkčnost se naprogramuje pomocí psaní kódu v C++, nebo lze krom kostry vytvořit i základní chování pomocí blueprintu a složitější chování

dodefinovat pomocí psaného kódu, či celou hru postavit pouze za použití blueprintu [2]. V této práci byl využit poslední způsob programování a to z důvodu, že samotný blueprint nabízí dostatek nástrojů a typů, které stačí k vypracování herní logiky i k vytvoření plně funkčních prefabrikátů. V neposlední řadě také dovoluje programovat i za běhu samotné hry v editoru, což se skvěle hodí k ověřování, které řešení problému je funkční, či efektivnější. Blueprint lze spustit i v simulačním módu, kdy programátor může sledovat jaké uzly a v jakém pořadí jsou prováděny.

Příklad blueprintu je uveden na obrázku 5.6.

3.3 Typy, funkce, aj.

Blueprinty jsou několika typů, které mají své specifické využití.

- **Blueprint Class** je asset dovolující programovat nové prvky typu Actor, nebo stavět už nad existujícími prvky.
- **Level Blueprint** je specifický typ, který se vztahuje na vše v rámci dané úrovně. Každá nově vytvořená úroveň si vytváří tento typ blueprint automaticky, nelze je jiným způsobem vytvářet. Výhodou tohoto typu blueprintu je možnost odkazovat se na objekty umístěné přímo v úrovni a tímto způsobem skriptovat průběh celé úrovně. Nevýhodou je že z tohoto typu blueprintu, nelze referencovat objekty do ostatních blueprintů.
- **Data-Only Blueprint** je typ blueprintu, který obsahuje pouze kód v podobě vytvořeného grafu z uzlů.
- **Blueprint Interface** je typ blueprintu, který vytváří rozhraní pro blueprinty, obsahuje pouze názvy funkcí bez jejich implementace. Slouží pro vzájemnou komunikaci mezi blueprinty. Tento typ však přichází s omezeními, kdy do tohoto blueprintu nemůžeme přidávat žádné komponenty, vytvářet nové proměnné a editovat grafy.
- **Blueprint Utilities** je typ blueprintu, který nám dovoluje vytvářet vlastní události.

[2]

Pro tuto práci byli použity první dva typy blueprintů. Blueprint Class se využil při tvorbě prefabrikátů kdy pomocí komponent byl složen celý Actor, a následně pomocí grafů popsáno jeho chování. Level Blueprint se využil pro skriptování úrovní, a definici tlačítek. Více o tlačítkách 5.4.

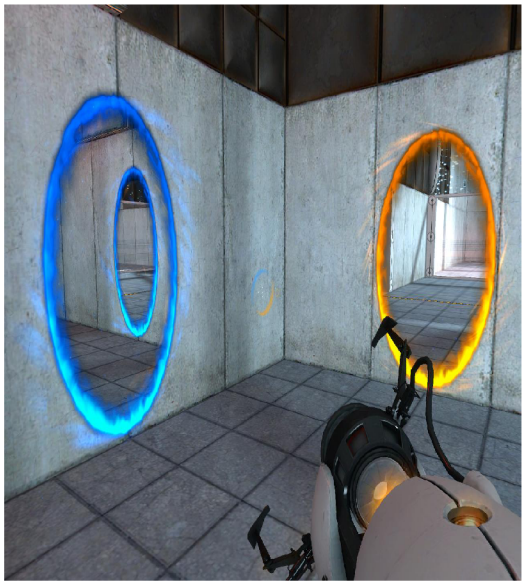
Pro samotné programování v C++ v Unreal engine je potřeba znát třídy, které Unreal engine používá a dodržovaní pravidel pro programování, z důvodu čitelnosti a znovupoužitelnosti kódu. Pro programování pomocí blueprintů je dobré se s několika třídami a typy seznám z důvodu jejich omezení. Příkladem může být, že fyzikální vlastnosti lze využít pouze na komponenty typu UPrimitiveComponent.

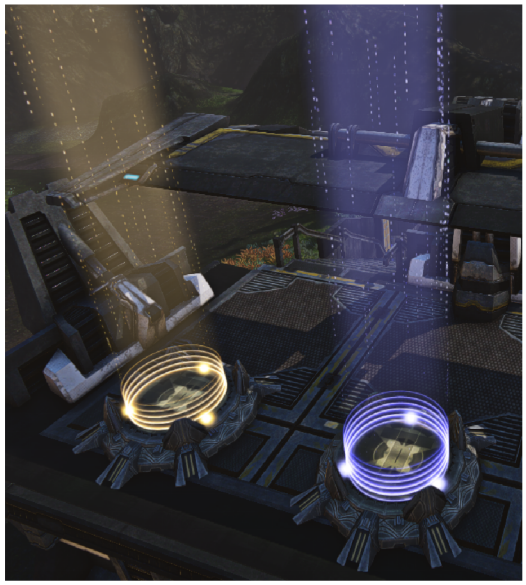
Kapitola 4

O hře

Jedná se o logickou hru, ve které hráč musí vyřešit nějaký problém, aby se mohl dostat o úroveň výše a tím se dostat do kýženého cíle a utéct z bludiště ztraceného v prostoru a čase. Stěžejní herní mechaniky jsou zrcadlové portály 5.1 a gravitační pole 5.2.

Nejedná se o zcela nové mechaniky, neboť teleportály v různých variacích jsou součástí většiny logických her. Například hra Portál používá dva odlišné portály, které hráč umisťuje na vybraná místa a pomocí nich řeší úlohy, které mu hra předloží. Jiné hry používají portály jako způsob rychlého pohybu hráče po úrovních, kdy se zde vkládají určité podmínky, jako například hráč musí vlastnit určitý předmět, aby mohl projít tím a tím portálem. Gravitační pole jsou součásti mnoha her a nemusí se jednat o logické hry. Často se tyto prvky objevují v FPS hrách[1] kde slouží jako rychlí způsob jak hráče dostat do výše položených částí hry. Příkladem lze ukázat pady ze hry Planetside 2.





Obrázek 4.1: Portály ze hry Portál 2 (nalevo, zdroj: http://csko-valve-atd.bloger.cz/obrazky/csko-valve-atd.bloger.cz/2-portaly.jpg) a gravitační pole v Planetside 2 (napravo, zdroj: http://iridar.net/wp-content/uploads/2017/08/begin-grav-lift.png)

[1]FPS-First person shooter hry ve kterých se střílí z první osoby

Inovativní na této hře je právě kombinace těchto prvků, vložená do virtuální reality, kdy pole mají různé směry gravitace a portály se dělí na dva různé druhy. Důvodem proč primárně použít portály byl fakt, že v celé hře se používá chůze pomocí touchpadu kontroléru, namísto použití teleportu, který se využívá v mnoha hrách ve virtuální realitě. Důvod gravitačních polí je pokořit problém Unreal enginu, kdy gravitace je nastavitelné pouze na jedné ose a znovu použitelný prefabrikát by mohl být užitečný i pro jiné programátory. Základní herní mechaniky byly vytvářeny v prostředí pro první osobu, neboť se nejvíce blíží pohledu ve virtuální realitě a byla možnost tyto mechaniky testovat i bez potřebného vybavení jako je HMD s kontroléry pro virtuální realitu. Výsledkem bylo vytvoření fungujícího gravitačního pole a portálů. Při tomto způsobu bylo potřeba vytvořit samotný herní charakter pro první osobu, který simuloval osobu pohybující se ve virtuální realitě. Nevýhodou tohoto přístupu je, že tento blueprint je pro finální hru nepoužitelný, protože hra se ve výsledku hraje ve virtuální realitě. Další problém nastal, při převodu celé scény do virtuální reality, kdy kolize charakteru ve virtuální realitě a charakteru z první osoby měly naprosto odlišné chování. Problém s kolizí u virtuálního charakteru byl vyřešen pomocí fadeboxů 5.3, které částečně tento problém řeší.

Kapitola 5

Prefabrikáty

5.1 Zrcadlový portál

Zrcadlový portál slouží jako jedna možnost pohybu v rámci úrovně. Pomocí těchto portálů je možné přemisťovat hráče a speciální objekty, které se ve hře vyskytují. V zrcadlech se zobrazuje pozice na kterou se hráč, či daný objekt přemístí. Ve hře se vyskytují tři druhy portálů. Prvním je základní portál, který přenáší hráčský charakter na předem definované místo. Dalším typem portálu je zrcadlo, které dokáže přenést hráče na dvě různé lokace, kde mezi nimi lze přepínat. Posledním typem je portál který je schopný přenést objekt na požadovanou pozici v úrovni.

5.1.1 Základní zrcadlo

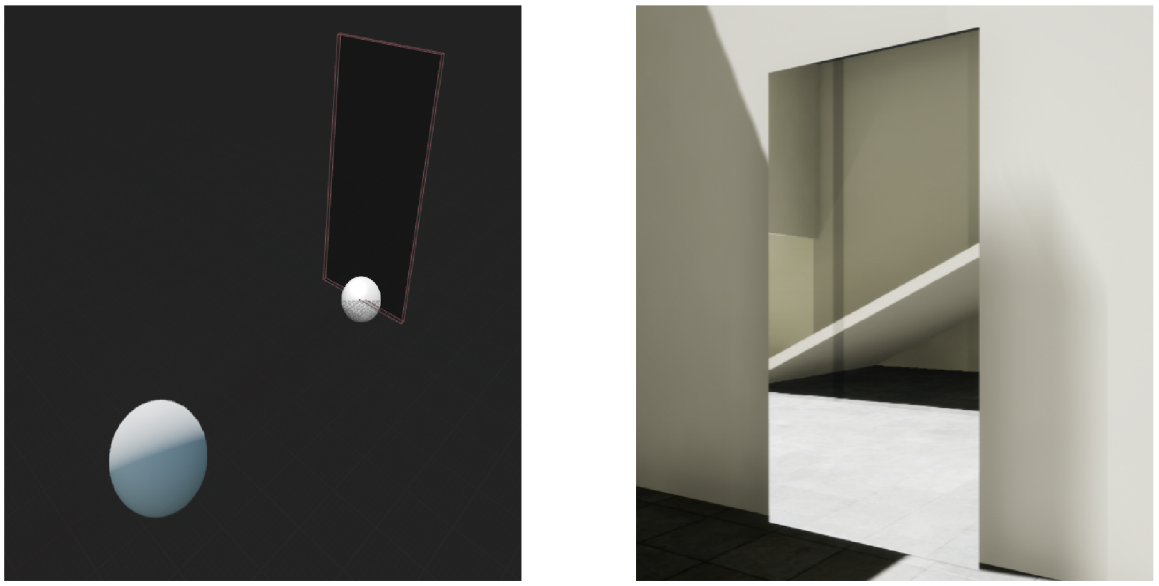
Základní zrcadlo *Mirror* se skládá ze čtyřech komponent. První je výchozí kořen scény (*DefaultSceneRoot*), na který je navázána komponenta (*StaticMeshComponent*) *Mirr*. Ta slouží jako tělo zrcadla na které se vykresluje textura, zobrazující cílovou lokaci daného zrcadlového portálu. Okolo *Mirr* se nachází kolizní box (*Box Collision*) *PortTrigger*, který slouží spuštění procesu portálu. Poslední komponentou je komponenta (*StaticMeshComponent*) *PortInto*, která určuje místo kam se bude hráč teleportovat při spuštění portálu viz obrázek 5.1. *PortInto* je možné při dosazení do úrovně volně přemisťovat nezávisle na umístění ostatních komponent portálu. Tato komponenta se ve hře nerenderuje.

Při překrytí kolizního boxu *PortTriger* nějakým objektem se provede kontrola, zda objektem je hráčský charakter pokud není neprovede se žádná operace. V opačném případě se využije funkce *Teleport* kdy cílem hráčský charakter a cílovou destinací je pozice komponenty *PortInto*, která se získá funkcí *GetWorldLocation*.

5.1.2 Dvou lokační zrcadlo

Dvou lokační zrcadlo *MirrorTwoEnded* se skládá z pěti komponent. Od *Mirror* se liší v komponentě *PortInto* a to tak že se zde tyto komponenty nacházejí dvakrát. První je komponenta *PortIntoWhite* a druhá komponenta je *PortIntoBlack*. Tyto komponenty jsou rozlišeny barvou pro přehlednost, aby se dalo rozlišit na kterou se budeme přemisťovat v rámci úrovně. *PortIntoWhite* je tvořena materiálem bíle barvy, která je jako barva základního materiálu a *PortIntoBlack* je tvořena materiálem černé barvy viz obrázek 5.2.

Implementace je také velice podobná. Přibývá proměnná typu Bool *PortOP*, která udává cílovou destinaci a proměnný typu *StaticMeshComponent End*, ze které se získává pozice



Obrázek 5.1: Model zrcadla (nalevo) a zrcadlo ve hře (napravo)

pro teleportaci. False hodnota *PortOP* je určené pro pozici *PortIntoBlack* a True hodnota je pro pozici *PortIntoWhite*. Implicitní hodnota je True. Pro nastavování této proměnné slouží dvě funkce. *Porttowhite* nastavuje hodnotu *PortOP* na True a Funkce *Porttoblack* nastavuje hodnotu PortOP na False. Test na tuto proměnou se vkládá mezi testem zda portál spouští hráčský charakter a mezi samotnou teleportaci. Poté se nastaví hodnota správná hodnota proměnné *End* ze které se získá lokace pro funkci Teleport.

5.1.3 Zrcadlo pro přenos objektů

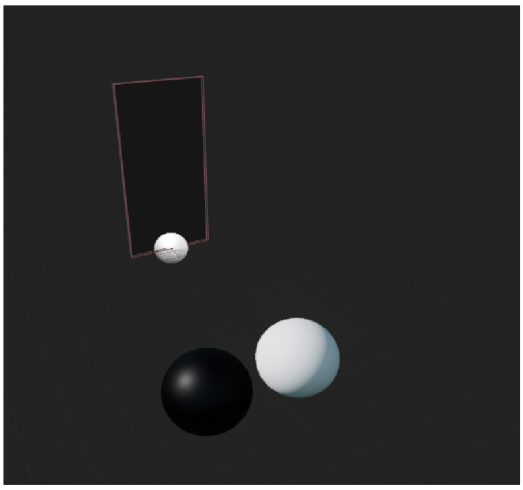
Zrcadlo pro přenos objektů je speciální případ nacházející se v první úrovni a je implementováno přímo v Level blueprintu této úrovně. Hlavním důvodem byla nutnost odstranění objektu (v tomto případě modrého míčku) z konkrétního gravitačního pole, které se nachází v dané úrovni.

Tento portál nevyužívá funkci Teleport. Přemístění objektu je založeno na odstranění objektu ze hry pomocí funkce DestroyActor a následně vytvořením nového objektu pomocí funkce SpawnActor <*ActorClass*> která vytvoří objekt na požadovanou pozici.

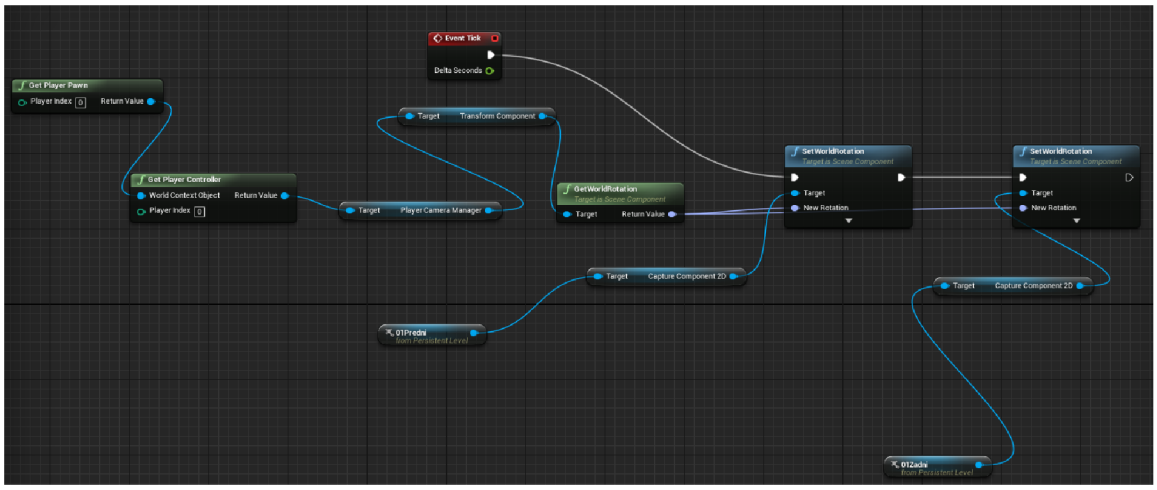
5.1.4 Textura pro zrcadlový portál

Textura pro zrcadlový portál se vytváří zvlášť pro každé zrcadlo. Prvně se pomocí komponenty *CaptureScreen2D* zachytí příslušný obraz, ten poslouží jako vzorek pro texturu. Tato textura se poté aplikuje na prvek zrcadla nazvaný *Mirr*. Základní hodnota je nastavena na texturu *Glass*, které představuje sklo a tvoří zrcadlo průhledným. Pri aplikaci naší textury je zrcadlo průhledné pouze ze zadní strany.

Pro optimální zobrazení obrazu v zrcadle je nutné nastavit spolupráci HMD a komponenty *CaptureScreen2D*, která se chová jako kamera. Docílíme toho, že přidáváme rotaci této kameře na základě rotace hráčova HMD. Toto je nutné udělat přímo v Level blueprintu úrovně viz obrázek 5.3.



Obrázek 5.2: Model dvoulokačního zrcadla.



Obrázek 5.3: Graf pro rotaci *CaptureScreen2D* kamer.

5.2 Gravitační pole

Gravitační pole slouží k přemisťování objektů v úrovni. Umožňují ovlivnění gravitace ve všech osách souřadnicového systému a je možné směr této gravitace v průběhu hry měnit. Dále je možné pole natáčet ve zvoleném směru, či úplné pole deaktivovat, nebo znova aktivovat.

5.2.1 Složení

Gravitační pole se skládá ze dvou komponent. první komponentou je výchozí kořen scény, Kolem kořene se nachází triggerbox, který spouští samotné funkce tohoto pole. Ohraničení tohoto trigerboxu je renderováno takže je viditelné hráči ve hře.

Jako další doplňující komponenty jsou efekty vytvořené pomocí kaskád (partical system) které tvoří grafický element tohoto prvku. Jedná se o částice které plynou ze středu pole ve směru příslušné gravitace. Při vypnutí pole je tento systém odstraněn a při zapnutí se opět vytvoří nový.

Obrázek 5.4: Využití gravitačního pole ve hře

5.2.2 Implementace

Blueprit obsahuje tři zásadní proměnné a jedno pole objektů. První proměnná typu Bool nazvaná *On/Off* se stará o řízení pole, určuje zda je pole aktivní nebo není. Tato proměnná je veřejná a lze ji na začátku inicializovat dle potřeby. Další proměnou je proměnná *Option Axis*, ta je typu Int a určuje samotný směr gravitace. Nabývá hodnoty od nuly do pěti. Tato proměnná je opět veřejná a lze ji na začátku inicializovat dle potřeby. Další důležitou proměnnou je proměnná *Power* která je typu Vector. Ta se stará o uložení hodnoty vektoru síly, které se bude aplikovat na tělesa v daném poli. Tato proměnná je interní a nelze do její hodnoty nijak zasahovat z venčí. Poslední je pole objektů *InActor*, do kterého se ukládají všechny komponenty typu *Actor*, které jsou přítomny v daném poli. Do tohoto pole se nevkládá pouze hráčský *Pawn*. Pole je opět interní a nelze do něj zasahovat z venčí.

Při spuštění úrovně se pro všechna aktivní pole (rozhoduje o tom hodnota proměnné *On/Off*) vytvoří příslušný *partial system*, podle hodnoty proměnné *Option Axis* a na tento systém je vytvořena reference. Následně pole reaguje na vstup, nebo výstup objektu z, nebo do *triggerboxu*. Při vstupu do pole se ověří aktivita pole. Pokud je pole aktivní projdou se všechny objekty typu *Actor* a třídou nastavenou na *BlueBall*, které se nacházejí v úrovni a porovnají se s objektem který vstupuje do pole. Pokud se najde shoda je tomuto objektu vypnuta gravitace, která je simulována samotný enginem a objekt je uložen do pole *InActors*. Při opouštění objektu z pole je provedená opačná procedura, kdy opět dojde k porovnání, zda je objekt správného typu a třídy, když je nalezena shoda, je tento objekt odstraněn z pole *InActors* a je mu znova aktivována gravitace simulovaná samotným enginem. Je nutno podotknout, že ostatní fyzické vlastnosti objektu zůstávají zachovány, neboť je jich třeba k výpočtu síly. S každým nově načteným framem se projde pole *InActors* a na každý prvek v tomto poli se aplikuje síla, která je dána ve vektoru *Power*. Tento vektor je aktualizován na základě změny hodnoty *OptionAxis*. Velikost sílz se vypočítá z hmotnosti daného objektu a gravitačního zrychlení, které je jako interní proměnná. Hodnota je posl0ze aplikována do vektoru, kdy hodnoty os jsou závislé na již zmíněné proměnné *OptionAxis*.

Tento prefabrikát nabízí funkce, které jsou schopny pole natáčet po směru všech os v kladném i záporném směru. Kdy jejich implementace pracuje s pozicí pole v dané úrovni. Dále se zde nacházejí dvě funkce, které jsou schopny zapínat a vypínat pole za běhu samotné aplikace. Funkce *OffGravity* nastaví hodnotu promenné *On/Off* na False. Poté se projde

pole *InActors* a pro každý objekt v tomto poli je zpět aktivována gravitace samotného enginu, při tom jsou tyto prvky odstraněny z daného pole. Nakonec se deaktivuje *partial systém* na který byla vytvořená reference. Funkce *OnGravity* nastaví hodnotu proměnné *On/Off* na True, posléze projde všechny objekty typu *Actor* se třídou *Blueball* a zkontroluje zda daný objekt se nekryje s zapínaným gravitačním polem. Pokud se kryje je tomuto objektu deaktivována gravitace enginu a je přidán do pole *InActors*. Dále se vytvoří nový *partial system* dle hodnoty proměnné *OprionAxis* a vytvoří se na něj nová reference. Poslední sada funkcí jsou funkce které jsou schopny za běhu aplikace měnit směr působení gravitace v jakémkoliv směru. Ty fungují tak, že nastaví příslušnou hodnotu promenné *OptionAxis*. Dále odstraní *pratial system*, na který má dané pole referenci a ihned se vytvoří nový *partial systém*, který zobrazí správný směr a je na něj opět uložena nová reference.

5.3 Fadebox

Fadebox slouží jako náhrada za kolizi ve hře. Jelikož hráčský *VRpawn* neobsahuje kolizní rozměry je nutné zajistit, aby se hráč nedostával do pozic ve, kterých by neměl být, tím je myšleno procházení zdí, nebo obcházení herních mechanik. Hlavní funkcí je ztmaveni displeje, kdy hráč ni nevidí a tím pádem by měl nabít dojmu, že se ocitl na místě ve kterém by se nacházet neměl.

5.3.1 Složení

Fadebox se skládá z výchozího kořene scény a z *triggerboxu*, který tento kořen obaluje. V základu je tento box velice úzký, aby se dal použít na stěny, které od sebe oddělují dvě místnosti.

5.3.2 Implementace

Fadebox reaguje na vstup či výstup hráče do pole. Při vstup hráče se zkontroluje zda se jedná o hráčský charakter a pokud ano, získá se reference na hráčský HMD a pomocí funkce *Start Camera Fade* se začne hráči obraz postupně ztmavovat do černa. Obraz ze opět vrátí potom co hráč opustí toto pole.

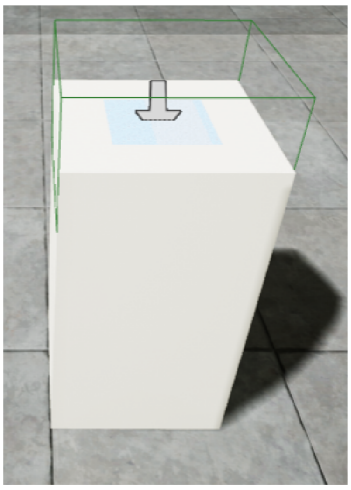
5.4 Tlačítka

Tlačítka v tomto projektu nejsou řešená jako prefabrikát. Proto se musí zvláště modelovat každé tlačítko a následně mu implementovat funkčnost. Důvodem je, že tlačítka přímo interagují s objekty, které jsou vloženy v úrovni a chování tlačítek se mění.

5.4.1 Složení

Tlačítko se skládá z kvádru, na kterém je umístěný panel kolem kterého je přidán *triggerbox*. Ten při kolizi s jiným objektem spustí naprogramovanou akci. Akce, kterými tlačítka mohou ovlivňovat ostatní prefabrikáty, jsou.

- Vypnutí a zapnutí gravitačního pole
- Změna směru gravitačního pole

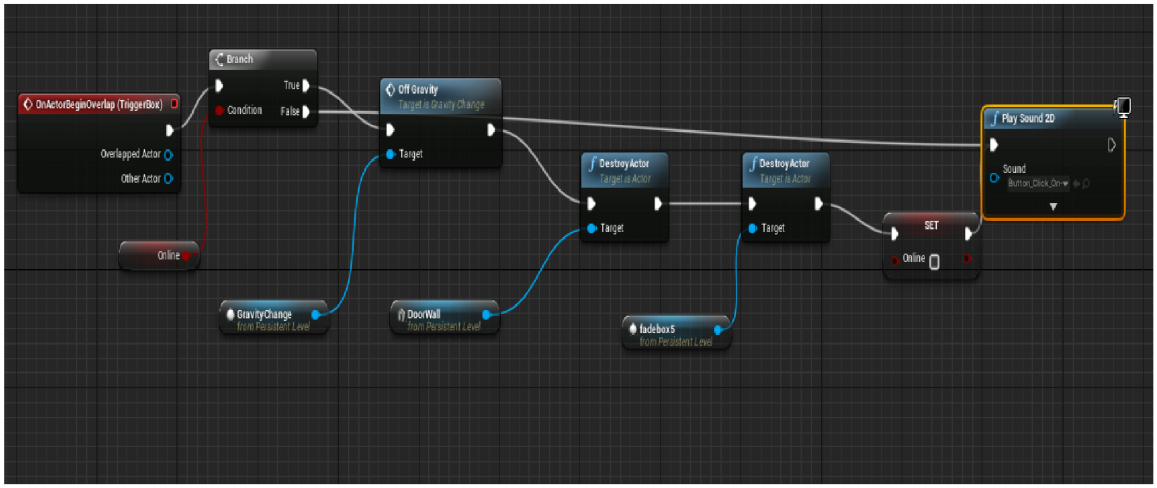


Obrázek 5.5: Tlačítka umístěný v úrovni

- Natáčeni gravitačního pole
- Změna cílové destinace u dvoulokačního zrcadla
- Odstranění dveří

5.4.2 Implementace

Tlačítko se posléze implementuje uvnitř daného Level blueprintu který patří úrovni ve které je tlačítko umístěno. Příklad funkce, kdy tlačítko vypne gravitační pole a odstraní dveře s fadeboxem, je vidět na obrázku 5.6.



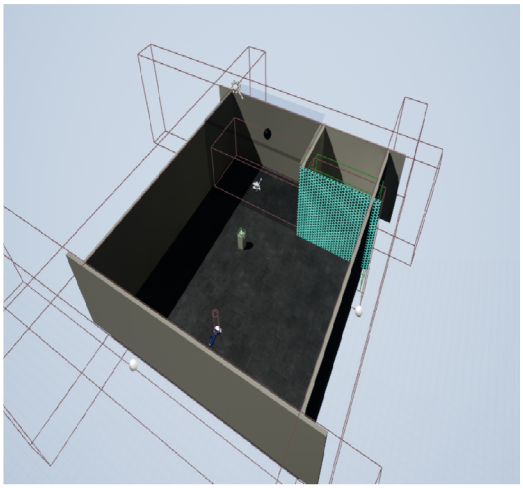
Obrázek 5.6: Graf pro funkčnost tlačítka v první úrovni

Kapitola 6

Úrovně

6.1 Level 01

První úroveň seznamuje hráče s prostředím hry. Také se zde setká s první mechanikou, která je gravitační pole.



Obrázek 6.1: Nadhled na první úroveň

6.1.1 Rozložení

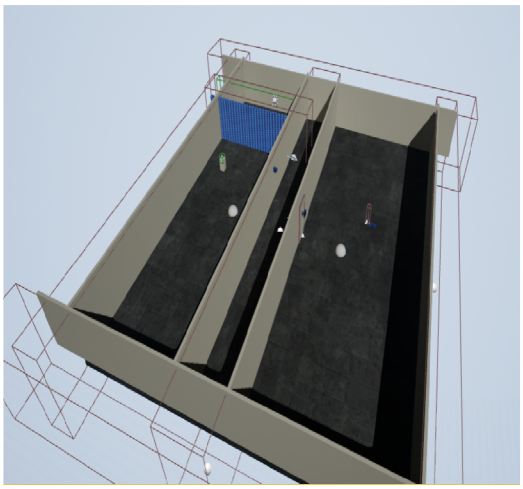
Úroveň se skládá z jedné menší platformy, která je dokola obehnána zdmi. Dále se zde za sklem nachází jedno gravitační pole, ve kterém je míček tlačen nahoru. Dále se zde nachází tlačítko, které ovládá gravitační pole. Nakonec se zde nacházejí dveře, za kterými je průchod do další úrovně.

6.1.2 Průchod-návod na úroveň

Hráč začíná, na kraji místnosti, a proti sobě má celou scénu. Jeho úkolem je dostat se k tlačítku a vypnout gravitační pole. Poté co hráč vypne gravitační pole se míček dostane na zem a odstraní dveře a může projít do další úrovně.

6.2 Level 02

Druhá úroveň stále pouze hráči předvádí další mechaniku, kterou je jednoduchý zrcadlový portál. Hráčův cíl je opět se dostat za dveře, které jemu blokují průchod do další úrovně.



Obrázek 6.2: Nadhled na druhou úroveň

6.2.1 Rozložení

Úroveň se skládá z jedné větší platformy, která je rozdělená na tří části. První je startovní pozice, kde se na zdi nachází zrcadlový portál. Druhou částí je výplňová část, do které hráč nemá přístup a slouží jako ochrana proti přeskočení mechaniky portálu. Zde se nachází fadebox. Poslední část je část kde se nachází tlačítko a další portál. V této místnosti jsou také dveře, které po stisknutí tlačítka zmizí. Celá úroveň je opět obehnána zdmi.

6.2.2 Průchod-návod na úroveň

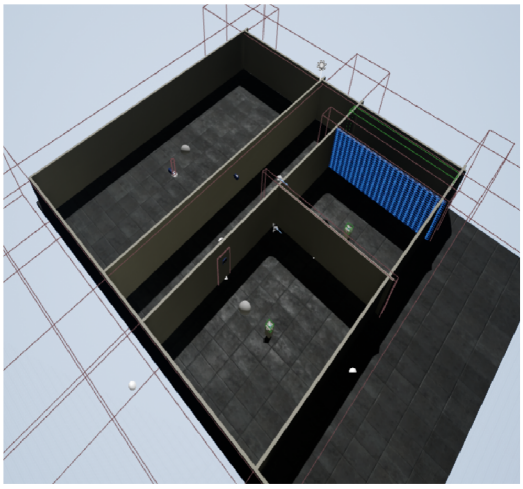
Hráč začíná v místnosti s jedním zrcadlem, kde má hráč za úkol jím projít. To ho teleportuje do místnosti s tlačítkem, kde po stisknutí tohoto tlačítka odstraní dveře a volně projde do další úrovně.

6.3 Level 03

Třetí úroveň a zároveň poslední úvodní úroveň, hráči představí poslední mechaniku, kterou je dvoulokační zrcadlo. Hráčův cíl je odstranit dveře a dostat se tak do další úrovně.

6.3.1 Rozložení

Úroveň se skládá z jedné větší platformy, která je rozdělena na čtyři části. První část je podobná druhé úrovni a to tak, že se zde nachází jedno zrcadlo. Účelem je ukázat, že ne každé zrcadlo se musí chovat stejným způsobem. Další částí je opět výplňová část pro umístění fadeboxů, které zabraňuji přeskakování mechaniky. Třetí částí je část s tlačítkem a dalším zrcadlem, kdy tlačítko ovládá portál v první místnosti a překalibruje jeho cílovou destinaci. V poslední části se nachází zrcadlo, tlačítko a samotné dveře, bránící hráči v



Obrázek 6.3: Nadhled na třetí úroveň

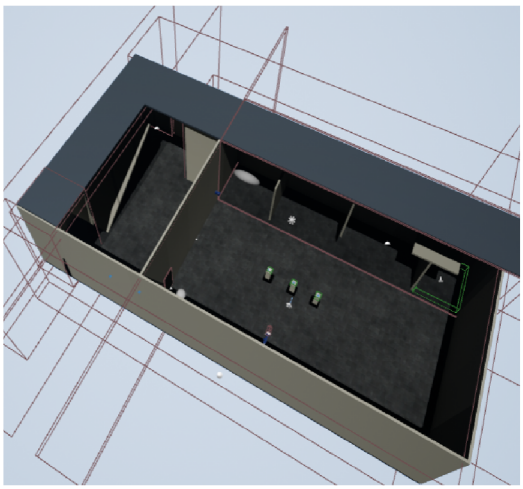
průchodu. Pomocí tlačítka si hráč může otevřít dveře za kterými se nachází průchod do další úrovně. Vše je opět obehnáno zdmi.

6.3.2 Průchod-návod na úroveň

Hráč podobně jako v druhé úrovni začíná v místnosti s jedním zrcadlem. Po průchodu tímto zrcadlem, pomocí tlačítka, překalibruje portál v první místnosti. Pomocí zrcadla, které se nachází ve stejné místnosti jako hráč, se dostane zpět do první místnosti. Po opětovném průchodu se hráč dostane do poslední místnosti, kde pomocí tlačítka odstraní dveře a tak bude mít volný průchod do další úrovně.

6.4 Level 1

Čtvrtá úroveň je zároveň první úroveň, ve které hráč začne řešit první úkoly, aby se mohl dostat do další úrovně. Hráčův cíl je dostat míč přes bludiště.



Obrázek 6.4: Nadhled na čtvrtou úroveň

6.4.1 Rozložení

Úroveň se skládá z jedné velké platformy, která je je obehnána zdmi a rozdělena na dvě místnosti. První je startovní místnost, ve které se nachází tří tlačítka s bludištěm a zrcadlový portál, sloužící pro průchod do druhé místnosti. v druhé místnosti se nachází zrcadlový portál, který dostane hráče zpět do místnosti s bludištěm, dále dvě tlačítka pro ovládání polí v místnosti a překážková dráha ve které je umístěn míček. Za překážkou je zrcadlo, které dostane míček do bludiště v první místnosti.

6.4.2 Průchod-návod na úroveň

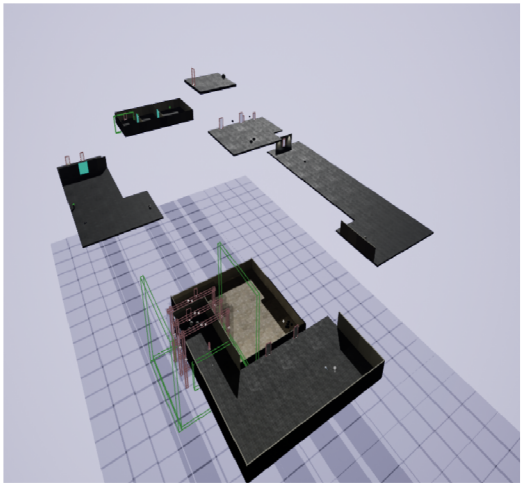
Hráč začíná v místnosti s bludištěm. Jeho prvním úkolem je získat míček, který následně bude navigovat skrz toto bludiště. Aby hráč dostal míček musí projít zrcadlovým portálem. Za ním se nachází místnost s překážkou. Pomocí tlačítek hráč ovládá pole, které mu pomohou míček dostat k zrcadlu, které se nachází za překážkou. Tlačítko na levé straně vypíná a zapíná pole které tlačí míček směrem nahoru. Tlačítko napravo zase ovládá pole které tlačí míček směrem vpravo. Pomocí vypnutého pole vzhůru a zapnutého pole směrem doprava se míček dostane přes překážku a teleportuje se do bludiště. Poté se hráč pomocí zrcadla vrátí zpět do místnosti, ve které začínal. Zde pomocí tří tlačítek naviguje míč přes bludiště tak, že v bludišti se po celém prostoru nacházejí tři pole, které dokáží míček tlačit směrem vzhůru, doprava či doleva. Tlačítka mají za úkol tyto pole zapínat a vypínat tak, že tlačítko vpravo ovládá pole, které tlačí míček směrem doprava. Tlačítko uprostřed ovládá pole, které tlačí míček směrem vzhůru a poslední tlačítko vlevo ovládá pole, které tlačí míček doleva. Hráč si muže předtím, než si teleportuje míček do tohoto bludiště, tyto pole přednastavit a tím získat výhodu. Když míček dosáhne země na druhé straně bludiště je hráč okamžitě teleportován do druhé úrovně.

6.5 Level 2

Pátá úroveň a zároveň poslední úroveň této demo aplikace, je koncipována jako zrcadlové bludiště, kdy má hráč za úkol správně projít zrcadly, aby se dostal na konec úrovně. Tato úroveň byla inspirována částí raidu ze hry Guild Wars 2 zvaná Twisted Castle (ilustrace na obrázku v příloze C.1), kdy má skupina deseti lidí za úkol projít bludištěm portálů a pomocí nášlapných desek si otevřít cestu do další části raidu.

6.5.1 Rozložení

Úroveň se skládá celkem z osmy platforem, které se nacházejí v několika výškových úrovních. První startovní platforma má tři zrcadla, jedno je klasické a dvě další jsou dvoulokační. Dále jsou zde tři uzavřené roury tvořící dráhu pro míček. Ty slouží jako nápověda hráči. Pro pohyb míčku uvnitř rour jsou umístěny gravitační pole. Další platformou je dlouhá chodba která má na konci zrcadlo. Tato platforma není obehnaná zdmi a hráč má výhled po celé úrovni. Další dvě platformy jsou malinké platformy kde se nacházejí dvě dvoulokační zrcadla, kterými má hráč za úkol projít. Dále je tu velká platforma která je obehnaná zdmi, ve které se nachází dvě zrcadla. Tato místnost slouží jako potrestání hráče, že zvolil špatné zrcadlo. Na této platformě je také mechanika, která odstraňuje část podlahy. Ta je ovšem pouze grafická a nemá žádný efekt na samotný průběh hry. Předposlední platforma je další platforma v podobě chodby, která skrývá jedno tlačítko a dveře, za kterými se skrývají další



Obrázek 6.5: Nadhled na pátou úroveň

dvě dvoulokační zrcadla. Polední platforma je finální platforma skrývající dvoje dveře se sadou dvou tlačítek a průchod do další úrovně.

6.5.2 Průchod-návod na úroveň

Hráč začíná na startovní platformě přesné pod rourami. Jakmile hráč opustí prostor pod rourami spustí mechanismus, který aktivuje gravitační pole v těchto rourách a míčky se začnou pohybovat ve směru hodinových ručiček, nebo v protisměru. Směr, kterým se tytou míčky pohybují určí jaké zrcadlo si má hráč vybrat, aby se nedostal na spodní platformu. Platí pravidlo, že roura, která je nejblíže hráči symbolizuje první sadu dvoulokačních zrcadel[1], druhá roura druhou sadu a třetí roura poslední sadu. První dvě sady zrcadel se nacházejí na malých platformách a třetí sada zrcadel se nachází za dveřmi v chodbovité platformě. Pokud de o směr kterým se míčky hýbou, tak pohyb míčku po směru hodinových ručiček značí zrcadlo vpravo, zatímco pohyb míčku v protisměru značí, že správné zrcadlo se nachází nalevo. Tlačítka na mapách postupně otvírají dveře, avšak při jejich stisknutí je hráč teleportován zpět na startovní platformu a dvoulokační zrcadla projdou rekalibrační fází, čímž se změní také směr míčku v rourách. To jakým způsobem je nutné procházet zrcadla je zobrazeno v tabulce 6.1. K dokončení úrovně je zapotřebí projít bludištěm alespoň třikrát. Po dokončení posledního průchodu se hráč dostane na konec herního dema. To však neznamená konec hry. Hráč se dostane do další úrovně, kde bude mít možnost vyzkoušet si různé věci a mechaniky, které se nacházejí v ostatních úrovních.

Běh	**První sada**	**Druhá sada**	**Třetí sada**
První	Vpravo	Vlevo	Vlevo
Druhý	Vlevo	Vlevo	Vpravo
Třetí	Vpravo	Vpravo	Vpravo

Tabulka 6.1: Tabulka pro správný průchod zrcadli

[1] Zrcadla na startovní platformě se nepočítají jako sada, zrcadlo vlevo teleportuje na konec chodby takže hráči ušetří čas

Kapitola 7

Testování

Celkem byly na testy přizvány tři osoby. Žádná s těchto osob neměla žádnou zkušenost s použitím virtuální reality. Všechny osoby byli studenty VUT na fakultě informačních technologií ve věku mezi 22 a 23 lety.

Při testování zda je dostatečně vše ukázáno a mechaniky jsou dostatečně srozumitelné, nevznikaly žádné vážné problémy a testované osoby pochopily co daná mechanika dělá. Se samotnou hratelností hry se problémy objevily. První problém, který při testování nastal byl s kalibrací rychlosti pohybu, kdy se postupně musel upravit koeficient rychlosti pohybu tak, aby to hráčům nebylo nepříjemné. Další zásadní věcí, která se naskytla byl problém se zobrazováním textury v zrcadlových portálech, kdy na obrazovce monitoru bylo zobrazení správné, ale hráč na svém displeji u HMD měl obraz rozdvojený. Tento problém se bohužel nepodařilo vyřešit. Poslední věcí, která se vyskytla při testování, byl problém, kdy hráč opustil platformu v Level 2 a volně chodil ve vzduchu. Tento problém se částečně vyřešil opět pomocí fadeboxů, které poukázali na skutečnost, že hráč by v takovéto pozici být neměl. V průběhu testování nevznikly žádné vážné příznaky nemoci z virtuální reality.

Kapitola 8

Závěr

Cílem bylo vytvořit demo aplikaci pro virtuální realitu. Výsledná hra obsahuje celkem šest úrovní z kterých dvě jsou rozsáhlejší. Všechny mechaniky jsou postupně ukázány v prvních úrovních. Všechny prefabrikáty byly vytvořeny v prostředí Unreal engine 4 se kterým jsem neměl žádné zkušenosti, stejně jako s vývojem nějaké hry či prací s virtuální realitou. Ovšem vše pro mě byl přínos částečně proniknout do každé z problematik a i přes technické problémy se seznámit s prací s prostředky pro virtuální realitu tak i s prostředím Unreal engine a v neposlední řadě kolik práce a času zabere vytvoření nějaké mechaniky a následné propojení více mechanik do srozumitelného celku.

Hra by šla dále vyvíjet a zdokonalovat. Primárně by se museli odstranit chyby jako je špatná kolize a zobrazování textur v zrcadlech. Dále by se přidali další mechaniky, například interakce s časem, nebo deformace prostoru. Zlepšit kvalitu grafických prvků jako například identifikátor směru gravitace v gravitačním poli.

Přínosem této práce je i možnost využití prefabrikátů, které mohou být použity a zdokonaleny v dalších projektech.

Literatura

[1] Corporation, H.: *VIVE*. [Online; navštíveno 4.5.2019].
URL https://enterprise.vive.com/eu/support/

[2] EpicGames: *Blueprints Overview-UE4 Documentation*. [Online; navštíveno 10.5.2019].
URL https://docs.unrealengine.com/en-US/Engine/Blueprints/Overview

[3] EpicGames: *Components-UE4 Documentation*. [Online; navštíveno 8.5.2019].
URL https://docs.unrealengine.com/en-US/Programming/UnrealArchitecture/Actors/Components

[4] Jay: *How Lenses for Virtual Reality Headsets Work*. [Online; navštíveno 7.4.2019].
URL https://vr-lens-lab.com/lenses-for-virtual-reality-headsets/

[5] LaValle, S. M.: *Virtual Reality*. [Online; navštíveno 3.4.2019].
URL http://vr.cs.uiuc.edu/vrbook.pdf

[6] NVIDIA: *Welcome to PhysX*. [Online; navštíveno 8.5.2019].
URL https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/3.3.4/Manual/Introduction.html#physics-vs-physx

[7] Steave Aukstakalnis, D. B.: *Reálně o virtuální realitě*. JOTA, 1994, ISBN 80-85617-41-2.

[8] Sutherland, I. E.: *The Ultimate Display* . [Online; navštíveno 27.3.2019].
URL
http://worrydream.com/refs/Sutherland%20-%20The%20Ultimate%20Display.pdf

[9] Valve: *Valve Index*. [Online; navštíveno 4.5.2019].
URL https://www.valvesoftware.com/cs/index

[10] Vlachos, A.: *Advanced VR Rendering GDC2015*. [Online; navštíveno 4.5.2019].
URL http://alex.vlachos.com/graphics/Alex_Vlachos_Advanced_VR_Rendering_GDC2015.pdf

[11] Wikipedie: *Fresnelova čočka*. [Online; navštíveno 4.5.2019].
URL https://cs.wikipedia.org/wiki/Fresnelova_čočka

[12] Wikipedie: *Spherical aberration*. [Online; navštíveno 5.5.2019].
URL https://en.wikipedia.org/wiki/Spherical_aberration

Příloha A

Obsah přiložené SD karty

Přiložená sd karta obsahuje krátké video s názvem *MirageIntro* a složku *Mirage* obsahující celý projekt i se spustitelnou aplikací. Složka Mirage má následující strukturu.

Build Obsahuje logy z buildu aplikace.

Config Obsahuje konfigurační soubory pro spuštění UE projektu.

Content Obsahuje všechny zdrojové soubory, kdy složky *Blueprints* a *Levels* obsahují vytvořené blueprinty.

Intermediate Obsahuje konfigurační soubory pro VisualStudio a Operační systémy.

Saves Obsahuje uložené soubory pro zálohu a chod UE projektu.

Video Obsahuje plné video průchodu do začátku posledního levelu.

WindowsNoEditor Obsahuje spustitelnou aplikaci s názvem *MIRAGE.exe* a potřebné soubory pro běh hry.

Mirage.uproject Spuštění projektu v Unreal Engine 4 ve verzi 4.16.2.

Příloha B

Návod na práci s aplikací

Aplikace se spouští pomocí *MIRAGE.exe*. K hraní je zapotřebí OS Windows 64bit, HTC VIVE, nebo HTC VIVE Pro HMD, Steam VR a dva kontroléry. Pohyb po herním světě je možný pomocí touchpadu na levém kontroléru, nebo samotným pohybem uvnitř vymezeného prostoru mezi základovými stanicemi. K vypnuti aplikace slouží klávesová zkratka Alt+F4.

Příloha C

Obrázky

Obrázek C.1: Twisted Castle ze hry Guild Wars 2. (zdroj: https://wiki.guildwars2.com/images/8/8d/Twisted_Castle.jpg)