

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Virtuální realita v prostředí Blender VR

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 4. května 2016

Václav Mareš

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce Ing. Petru Vaněčkovi, PhD. za cenné rady a připomínky, které mi pomohly tuto práci dokončit. Také bych rád poděkoval členům BlenderVR týmu, zejména pak Dr. Davidu Poirier-Quinotovi, za ochotu a spolupráci.

Abstract

The purpose of this work was to learn about a project BlenderVR thanks to which should be possible to use applications created in modeling tool Blender in virtual reality. Then get information about a support of VRPN protocol (Virtual Reality Peripheral Network) which enables communication with peripheral devices. Next goal was to find options of connecting chosen devices (Oculus Rift DK2, WiiMote, TrackIR 5, mouse and keyboard) with possible usage of VRPN. For each supported device create a simple demo application and then create a nontrivial virtual reality application. As a part of this work there also will be a description of configuration and usage of BlenderVR and VRPN in Windows environment.

Abstrakt

Cílem této práce bylo seznámit se s projektem BlenderVR, díky kterému by mělo být možné použít aplikace vytvořené modelovacím nástrojem Blender ve virtuální realitě. Dále bylo třeba získat informace o podpoře protokolu VRPN (Virtual Reality Peripheral Network), který dokáže zprostředkovat komunikaci s periferními zařízeními. Dalším bodem bylo zjistit možnosti připojení vybraných zařízení (Oculus Rift DK2, WiiMote, TrackIR 5, myš a klávesnice) s případným využitím VRPN. Pro každé podporované zařízení vytvořit ukázkovou aplikaci a závěrem vytvořit netriviální aplikaci virtuální reality. Součástí práce bude také popis konfigurace i obsluhy BlenderVR a VRPN v prostředí Windows.

Obsah

1	Úvod	1
2	Virtuální realita	2
2.1	Obecně	2
2.2	CAVE	3
3	Blender	4
3.1	Obecně	4
3.2	Blender Game Engine	4
3.3	Blender Player	5
4	BlenderVR	6
5	VRPN	8
5.1	Obecně	8
5.2	Architektura	9
5.3	Server	9
5.4	Klient	10
5.5	Vlastní modul	11
5.5.1	Dostupné moduly	11
6	Použitá zařízení	14
6.1	Myš a klávesnice	14
6.2	TrackIR 5 v kombinaci s TrackClip	14
6.2.1	Obecně	14
6.2.2	VRPN	15
6.3	Ovladač WiiMote	15
6.3.1	Obecně	15
6.3.2	VRPN	16
6.4	Oculus Rift DK2	17
6.4.1	Obecně	17

6.4.2	VRPN a BlenderVR	18
7	Instalace prostředí	19
7.1	BlenderVR	19
7.1.1	Součásti instalace	19
7.1.2	Dokončení	20
7.2	VRPN server	21
7.3	Zařízení	21
7.3.1	WiiMote	21
7.3.2	Oculus Rift DK2	22
8	Konfigurace a funkční kód	23
8.1	Rozbor konfiguračního souboru *.xml	23
8.1.1	Sekce starter	23
8.1.2	Sekce users	24
8.1.3	Sekce computers	24
8.1.4	Sekce screens	25
8.1.5	Sekce plugins	29
8.2	Soubor *.processor.py	30
8.3	Konfigurace VRPN	32
8.4	Konfigurace síťového propojení	33
9	Závěr	35
A	Spouštění a obsluha	40
A.1	BlenderVR	40
A.1.1	Spuštění aplikace	40
A.1.2	Obsluha	40
A.1.3	Řešení problémů	41
A.2	VRPN Server	44
A.3	Oculus Rift	45
B	Popis ukázek	47
B.1	Jednoduchá ukázka	47
B.2	Ukázka s myší a klávesnicí	47
B.3	Ukázka s WiiMote	48
B.4	Ukázka s Oculus Rift DK2	49
B.5	Demonstrační hra	50
B.5.1	Obecně	50
B.5.2	Zobrazení	50
B.5.3	Ovládání	52

B.5.4 Implementace	53
B.5.5 Závěr	53
C Obsah DVD	54

1 Úvod

Cílem této práce bylo seznámit se s projektem BlenderVR, což je software, který umožňuje modely, animace či hry vytvořené pomocí modelovacího nástroje Blender upravit pro virtuální realitu. Je tedy možné připojit různá periferní zařízení, jakým je např. Wii ovladač, nebo změnit standardní zobrazení nad rámce klasického Blender Playeru.

Klíčovým prvkem pro připojení většiny periferních zařízení je VRPN, tedy Virtual Reality Peripheral Network. Jedná se o protokol, který umožňuje komunikaci s řadou zařízení používaných ve spojení s virtuální realitou. Jedním z důležitých bodů této práce bylo zjistit jeho podporu pro vybraná zařízení a zajistit správnou konfiguraci a obsluhu VRPN serveru v souvislosti s BlenderVR.

Ukázalo se, že propojení BlenderVR s VRPN lze uskutečnit, a proto byly vytvořeny jednoduché ukázkové aplikace, které demonstrují funkčnost jednotlivých zařízení. S využitím získaných znalostí byla také vytvořena netriviální aplikace virtuální reality, která kombinuje prvky použité v demonstračních ukázkách.

V průběhu práce bude čtenář informován o tom, jak správně nakonfigurovat BlenderVR, aby jej bylo možné použít pro vlastní aplikace. Součástí bude také popis jednotlivých zařízení a jejich ukázek. Vše bude zaměřeno na práci s operačním systémem Windows.

2 Virtuální realita

2.1 Obecně

Virtuální realita (VR) je počítačem vytvořené prostředí, které uživateli dává pocit reálného světa. Principem je zmást lidské smysly tak, aby si mozek myslel, že jde o realitu. Tohoto efektu může být dosaženo řadou různých způsobů a zařízení. Velký důraz se klade zejména na vizuální iluze, protože zrak je hlavní složkou vnímání. Zařízení, která se zaměřují právě na zrak, a v poslední době se značně rozšířila, se nazývají HMD neboli Head-mounted Display. HMD se dá volně přeložit jako „Displej, který se dá připevnit na hlavu“ a typicky se jedná o určité typy brýlí s projekty. Uživateli je promítána scéna přímo před očima a okolí je odstíněno. Díky tomu se může uživatel plně soustředit na projekci. Častým prvkem bývá pohybové čidlo, které snímá pohyb a náklony hlavy a upravuje podle toho projekci. Výhodou těchto zařízení je jejich snadná přenositelnost. Zároveň není třeba upravovat celou místnost (uživatel ji nevidí), což šetří čas a prostředky. Příkladem těchto zařízení může být Oculus Rift, HTC Vive nebo Playstation VR.

Dalším způsobem navození virtuální reality může být např. CAVE systém (viz níže). K celkovému dojmu značně pomáhá kvalitní prostorový zvuk, který dobře doplňuje scénu. V případě, že se jedná o interaktivní scénu, často se používají různé ovladače, které reagují na pohyb, jakým je třeba Wii ovladač.

VR má řadu využití. Často bývá použita jako trenažér pro různé aktivity, které z nějakého důvodu není možné snadno zkusit v reálném světě. Může se tak jednat například o různé letové nebo vojenské simulátory, trenažéry pro přežití přírodních katastrof, sportovní simulátory, výukové simulátory pro lékaře atd. S VR se také můžeme setkat při prezentaci nejrůznějších designů od domů, aut až po návrhy elektroniky. Jednou z nejširších oblastí použití je zábavní průmysl, často tak narazíme na VR ve spojitosti s filmy nebo hrami.[1]

2.2 CAVE

CAVE (Cave Automatic Virtual Environment) je prostředí pro virtuální realitu, které bylo vynalezeno skupinou vědců v roce 1992. Snahou bylo vytvořit vizualizační nástroj, který využívá velkých projekčních stěn.

CAVE je obvykle tmavá čtvercová místnost, na jejíž stěny je promítáno pomocí zadní projekce¹. Díky tomuto systému umožňují moderní CAVE promítání na všech šest stěn, což zajišťuje velice dobrý dojem z celé virtuální reality. Tato projekce s sebou však nese také řadu nevýhod, jakou je například nutnost vyvýšení místnosti kvůli stropní projekci, komplikace se vstupem do místnosti, dezorientace atd. Proto se tedy často, zejména z finančních důvodů, od CAVE systému složeného z šesti stěn upouští.[2]

¹Zadní projekce je termín, kdy je projektor umístěn za plátnem. Díky tomu není uživatel rušen světlem projektoru a projektor nestíní výhled.

3 Blender

3.1 Obecně

Blender je software, který umožňuje modelování 3D grafiky. Je zde k dispozici řada pokročilých funkcí, jakými jsou například: vytváření animací a počítačových her, simulace proudění tekutin, simulace pohybu těla a částic a mnoho dalších.

Hlavní výhodou Blenderu je, že se jedná o open-source software a může jej tak zdarma používat, a dle svých představ upravovat každý, za předpokladu, že dodrží GNU General Public License, tedy vytvořený software také musí spadat pod GPL licenci. Další z jeho velkých předností je to, že se jedná o multiplatformní nástroj, jinými slovy, dá se používat na různých operačních systémech. Díky tomu Blender využívá řada uživatelů od amatérů až po profesionály. Tito uživatelé mezi sebou sdílejí své zkušenosti a dávají tím cenné rady a tipy nováčkům v této oblasti. Proto se vývoj Blenderu stále posouvá kupředu a nezůstává pozadu za komerčními vývojovými nástroji.[3][4]

3.2 Blender Game Engine

„Blender má svůj vlastní vestavěný herní engine^a, který umožňuje vytvářet interaktivní 3D aplikace nebo simulace. Hlavní rozdíl mezi Game Engine a tradičním Blender systémem je v procesu renderování. V klasickém Blenderu jsou obrázky a animace vytvořeny offline – jakmile se vyrenderují, nemohou být změněny. Naopak, Blender Game Engine renderuje scénu průběžně v reálném čase a jeho součástí jsou nástroje pro uživatelskou interakci během renderovacího procesu.“[5]

^aHerní engine (Game Engine) je jádro celé hry. Jedná se o centrálu, které řídí zbylé součásti aplikace.

Blender Game Engine (BGE) je užitečný doplněk, který přidává možnosti nad rámec klasického modelovacího nástroje. Renderování v reálném čase je možné využít k vytvoření interaktivních animací či přímo her. Navzdory

svým širokým možnostem, není práce s BGE příliš složitá, nicméně je třeba naučit se s tímto nástrojem pracovat. Klíčem práce s BGE je logický editor, který umožňuje připojit k jednotlivým objektům posluchače. Ty se spustí na základě uživatelem definované události, kterými se může myslet např. nějaký typ kolize, stisknutí tlačítka nebo pohyb myši. Poté uživatel také musí definovat, co se bude dít, jakmile daná událost nastane. Může se jednat např. o pohyb nebo deformaci objektu, ale také jeho odstranění či přidání. Události se nastaví pomocí konfigurace tzv. logických bloků, kde uživatel klikáním navolí požadované možnosti. Objektům je také možné přiřazovat tzv. vlastnosti (*properties*). Ty můžeme chápat jako proměnné v programovacích jazycích. Uchovávají hodnotu různého datového typu (např. číslo, text), se kterou je možné dále pracovat.

K posluchači je také možné připojit skript v jazyce Python, který bude vykonáván. Pomocí těchto skriptů můžeme vytvářet komplexní logické operace, které bychom pomocí logických bloků vytvářeli obtížně, nebo by nebyly vůbec možné. Tato varianta také může vyhovovat uživatelům, kteří jsou na programování zvyklí a preferují procedurální přístup.

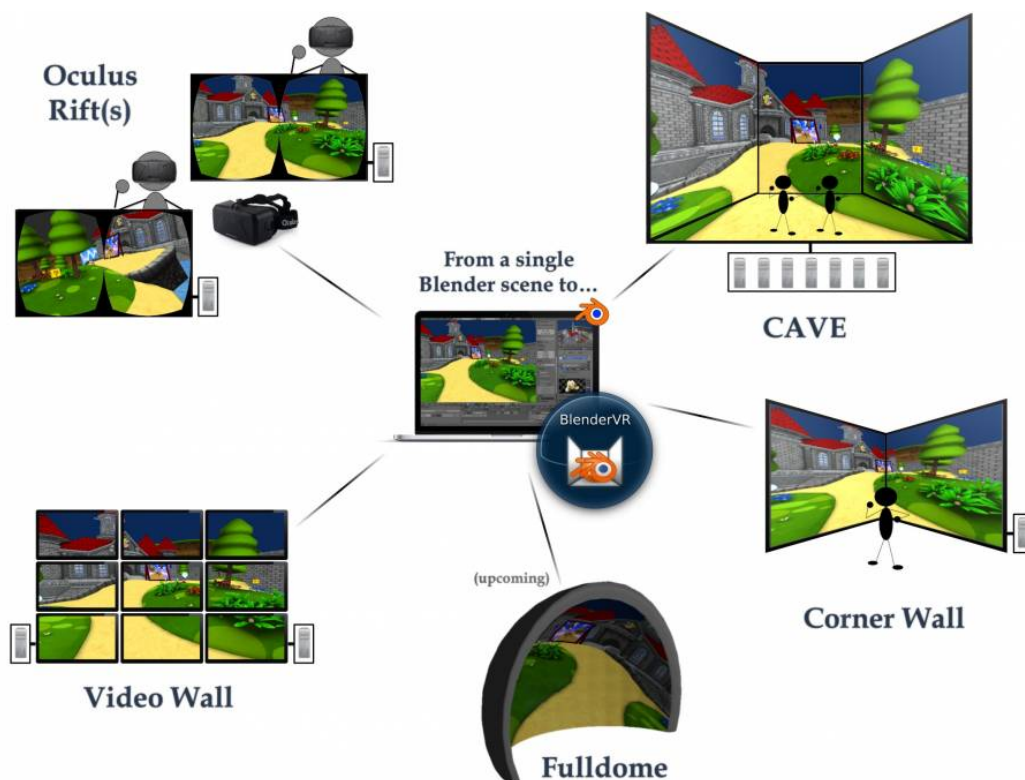
3.3 Blender Player

Vytvořené projekty je možné spouštět pomocí přehrávače Blenderu, tedy Blender Playeru. Je to standardní součást Blenderu, nicméně je možné jej spouštět také samostatně, což umožňuje uživatelům distribuovat jejich práce bez nutnosti spouštění samotného Blenderu. Přehrávač lze nastavit dle našich potřeb. Můžeme tak např. změnit velikost okna, upravit rozlišení, zobrazit počet snímků za sekundu atd.

4 BlenderVR

BlenderVR je projekt vytvořený skupinou vývojářů z Francie, jejichž cílem bylo spojit herní engine programu Blender s virtuálním prostředím. Díky tomuto projektu je tedy možné vytvořit hru nebo simulaci pomocí 3D modelovacího nástroje Blender a poté ji spustit ve virtuální realitě.

Z počátku byl projekt zaměřen pouze na CAVE systém, a proto se dříve jmenoval Blender CAVE. Vzhledem ke své expanzi byl projekt přejmenován na obecnější BlenderVR. Nyní je možné zobrazení na různá HMD zařízení, jako je např. Oculus Rift, nebo na stěnu složenou ze soustavy monitorů. Možnosti zobrazení jsou graficky znázorněny na obrázku 4.1. Také je možné využití OSC, neboli Open Sound Control, což je protokol pro synchronizaci zvuku.[8]



Obrázek 4.1: Grafické znázornění možností BlenderVR

BlenderVR disponuje širokou škálou nastavení. Díky tomu je možné si adaptovat zobrazení na řadu zařízení s různým počtem obrazovek. Je také

možné připojit různá periferní zařízení, používaná ve virtuální realitě. Může se jednat o pohybová čidla, kamery, ovladače atd. Tato zařízení můžeme se systémem spojit pomocí protokolu VRPN.

Většina nastavení probíhá v konfiguračním souboru typu `xml`, se kterým aplikace pracuje. BlenderVR je napsán v programovacím jazyce Python, který je také třeba k obsluze logiky ve vytvořených projektech, stejně jako v Blenderu.

Vývojáři se nesnaží jen opravit a zdokonalit stávající funkce, ale přirozeně projekt také rozšiřují o nové. Jednou z připravovaných novinek je podpora tzv. `fulldome`, což je kopule, do které se promítá. Častým využitím tohoto dómu jsou planetární projekce, kdy uživatel kouká nahoru do kopule, kde je mu promítán vesmír.

Projekt BlenderVR stále prochází velkým vývojem. Členové týmu projekt vylepšují a dělají ho tak přístupnější pro širší veřejnost. BlenderVR je možné spustit na různých operačních systémech. Je zde podpora pro Windows, Linux i Mac OS.[6][7]

Možnosti BlenderVR jsou podrobněji rozebrány v kapitole Konfigurace, kde jsou doplněny o konkrétní příklady.

5 VRPN

5.1 Obecně

Virtual Reality Peripheral Network neboli VRPN je sada pravidel navržených tak, aby dokázala implementovat taková síťová rozhraní, která umožní komunikaci mezi počítačem a fyzickými zařízeními použitými ve virtuální realitě. Těmito zařízeními se typicky rozumí různá pohybová čidla, ovladače, vysokorychlostní kamery atd. Cílem je vytvořit takové prostředí, kdy aplikace nebude vědět o síťové vrstvě a nebude závislá na dané konfiguraci. Stejně tak síťové vrstvě nebude záležet na počítači nebo aplikaci, která ji obsluhuje. Bude se tedy jednat o využívání a nabízení služeb, které je běžné například v OSI/ISO modelu v základní myšlence správy sítí. Tímto způsobem můžeme na všechna zařízení zmíněného typu koukat jako na univerzální zařízení a vyšší vrstvy tak nabízejí mnohem větší variabilitu za stejné konfigurace, tedy nemusíme měnit naši aplikaci proto, abychom mohli implementovat jiné zařízení.

VRPN protokol byl vyvíjen se snahou o co největší rychlost, což je samozřejmě, neboť ve virtuální realitě je velmi důležité, aby výsledná aplikace měla co nejmenší zpoždění reakcí. Další snahou bylo také učinit vše co nejjednodušeji pro klienta. Zatímco server je možné široce nastavovat a vytvářet podporu pro vlastní zařízení, každý klient pracuje s VRPN podobným způsobem. Problémem při vývoji VRPN bylo najít mez, kdy rozhraní bude dost jednoduché na to, aby podporovalo co nejvíce zařízení a zároveň dostatečně složité na to, aby implementovalo co nejvíce funkcí určitého zařízení. Je řada způsobů, jak se s tím protokol vypořádává, jedním z těch přímočarých je například ignorování zpráv od nepodporovaných funkcí. Pro zvýšení stability VRPN se používá časová synchronizace mezi klienty a servery. Také dochází k automatickému obnovení spojení v případě poruchy. VRPN podporuje řadu různých zařízení a jejich počet se s rozvojem této sféry stále zvyšuje.[9][10][11]

5.2 Architektura

VRPN rozlišuje zařízení dle typu, což určuje strukturu přenášených zpráv. Je to z toho důvodu, že různá zařízení slouží k různým účelům a je tedy třeba zasílat odlišené informace. Např. tlačítkový joystick bude odesílat podobná data jako klávesnice – číslo tlačítka, které bylo stisknuto. Budou tedy stejného typu – `vrpn.Button`. Oproti tomu pohybové čidlo typu `vrpn.Tracker` bude zasílat souřadnice, což vyžaduje jinou strukturu zprávy. Se zvyšujícím se počtem zařízení vznikají nové typy a ty již existující se upravují pro potřeby nových zařízení.

Samozřejmě je stále možné využívat dodatečných funkcí určitého zařízení, i přes to, že ostatní zařízení stejného typu touto funkcí nedisponují (např. pokročilejší čidla snímají více dimenzí dat, klávesnice používá více tlačítek než joystick atd.). Zařízení, která tyto nadstandardní funkce nepodporují je jednoduše ignorují, díky tomu je možné snadno vidět základní i pokročilá zařízení stejným způsobem, což značně usnadní práci s nimi.

5.3 Server

VRPN server má široké možnosti nastavení a je tak možné si jej upravit dle vlastních představ. Bohužel je nutné většinu funkcí nastavit již při kompilaci, poté je konfigurace značně omezena. VRPN je napsáno v programovacím jazyce C++ a je možné jej tedy zkompilovat např. pomocí programu CMake, Visual Studio nebo dalších nástrojů. Během kompilace můžeme ovlivnit např. podporovaný operační systém, podporovaná zařízení a mnoho dalších. Za předpokladu, že potřebujeme, aby náš server podporoval něco nestandardního, např. podporu jazyka Python, kterou vyžaduje BlenderVR, musíme si jej s velkou pravděpodobností zkompilovat sami, nicméně často se setkáme s tím, že VRPN server, který splňuje naše představy, je k dispozici ke stažení již zkompilovaný. To může velice usnadnit práci zejména uživatelům, kteří v této oblasti nemají velké zkušenosti. Zařízení, která server poslouchá, je třeba nastavit v konfiguračním souboru. Server může poslouchat více zařízení najednou a k danému serveru se může připojit více klientů.

5.4 Klient

V případě klienta je potřeba definovat dvě hlavní informace: jméno zařízení, které posloucháme a IP adresu nebo hostname serveru. Pokud tedy chceme pracovat s akcemi myši pomocí VRPN na tomto počítači, je třeba uvést jméno (standardně např. `Mouse0`) a jako hostname serveru: `localhost` (jedná se o tento počítač). Zprávy, které klient obdrží od serveru jsou v rámci daného zařízení ještě rozděleny na typy. V příkladu s myši tedy obdržíme dva proudy dat, jeden pro klikání (`vrpn_Button`) a druhý pro pohyb (`vrpn_Analog`). Díky tomu můžeme obsluhovat zvlášť různé akce daného zařízení, případně některé zprávy ignorovat.

Informace obsažené ve zprávě závisí na konkrétním zařízení a jeho typu. Zprávy jednotlivých tříd zařízení jsou reprezentovány strukturami. Jedinou informací, kterou přenáší všechny třídy zařízení, je čas, kdy daná akce nastala. Tato informace je reprezentována strukturou `timeval`, která ve své výchozí podobě není pro člověka příliš čitelná, nicméně je možné ji převést. Jako příklad si rozeberme tlačítková zařízení, která jsou zastoupena strukturou `vrpn_BUTTONCB`. Kromě zmíněného času tato struktura obsahuje číslo tlačítka, které bylo stisknuto (`int button`) a stav, který určuje, zda bylo stisknuto, či uvolněno (`int state`). Zatím co `button` nabývá hodnot dle počtu tlačítek (např. v případě myši 3, v případě klávesnice 256), `state` rozeznává pouze dva stavy 0 a 1.

Zde je zobrazen příklad zprávy po stisknutí pravého tlačítka myši:

Výčet hodnot

```
[1460237482:203380, 2, 1]
```

Výstup programu vrpn_print_device

```
Button Mouse0@localhost, number 2 was just pressed
```

Výstup programu vrpn_print_messages

```
Time: 1460237482:203380, Sender: Mouse0, Type vrpn_Button  
Change, Length 8
```

Jedná se o různé interpretace získané z přenášené struktury. Obdobně vypadají i ostatní typy zařízení, ale samozřejmě mají jiný formát.

5.5 Vlastní modul

Zkušenější uživatelé si mohou upravit části kódu serveru, aby jej rozšířili o své požadavky. Typickým důvodem je snaha o zajištění podpory pro zatím nepodporované zařízení, tedy tvorba modulu. Je možné vytvořit úplně novou třídu zařízení nebo navázat na nějakou již existující, záleží na tom, jaký modul vytváříme. Podrobný návod včetně specifikací je dostupný na oficiálních stránkách VRPN.[12][13]

5.5.1 Dostupné moduly

V současné době (10. 3. 2016) existují tyto oficiální třídy zařízení:

`vrpn_Tracker_Remote`

Trackery slouží jako senzory pohybu, které se typicky připevní na uživatele nebo je uživatel uchopí. Poté zaznamenávají jeho pozici či pohyb částí těla. Příklady zařízení, která využívají tento modul, mohou být WiiMote, Razer Hydra nebo Microsoft Kinect.

`vrpn_Button_Remote`

Tento modul používá každé zařízení, které disponuje tlačítky. Jedná se o třídu zařízení, která zaznamenává, zda bylo tlačítko stisknuto či uvolněno a o jaké tlačítko se jednalo. Modul využívá např. klávesnice, různé druhy ovladačů, myš atd.

`vrpn_Analog_Remote`

Tato třída zařízení umožňuje přenos analogových dat, která vysílají např. joysticky, posuvníky, ale i třeba myš či WiiMote. Informace jsou přenášeny polem hodnot různých rozměrů, které se definuje na straně serveru.

vrpn_Analog_Output_Remote

Jedná se o stejnou třídu zařízení jakou je `vrpn_Analog_Remote` s tím rozdílem, že zde klient specifikuje strukturu informací pro server.

vrpn_Dial_Remote

Modul `vrpn_Dial_Remote` umožňuje pracovat s množinou dat, jako by to byly ciferníky. Předmětem zprávy je vždy označení ciferníku (reprezentováno celým číslem) a o kolik by otočen (zlomek formou desetinného čísla). Jedná se o obdobu `vrpn_Analog_Remote` a tyto třídy se dají zaměňovat.

vrpn_ForceDevice_Remote

Zařízení využívající tuto třídu umožňují zaznamenávat sílu, kterou uživatel vyvíjí. Často se používají v kombinaci s třídami `vrpn_Button_Remote` a `vrpn_Tracker_Remote`, které obsluhují ostatní části daného přístroje. Příkladem může být PHANTOM OMNI od společnosti Sensable. Jedná se o přístroj, který má rameno se dvěma klouby na jehož konci je úchyt podobný tužce, se kterým uživatel může hýbat. Jsou zde také umístěna tlačítka.[14]

vrpn_Imager_Remote

Tento modul slouží k přenosu obrazů nebo videa. Informace jsou přenášeny proudem dat rozděleném po obrazech, kde každý obraz je oddělen počáteční a koncovou značkou. Mezi nimi je výpis jednotlivých obrazových bodů, které jsou reprezentovány hodnotami RGB.

vrpn_Sound_Remote

Třída zařízení `vrpn_Sound_Remote` umožňuje práci se zvukem.

vrpn_Poser_Remote

Tento modul umožňuje pracovat s pohyblivými platformami, jejichž funkcí je uzpůsobit svou pozici a náklon dle potřeby. Tuto třídu zařízení můžeme považovat za opak `vrpn_Tracker_Remote`. Místo toho, abychom detekovali pozici objektu, definujeme, kam se má posunout.

6 Použitá zařízení

6.1 Myš a klávesnice

I přesto, že se nejedná o zařízení navržená pro virtuální realitu, VRPN dokáže pracovat i s daty z myši a klávesnice. Využití těchto zařízení dlí především v jejich snadném testování. Také se často používají jako doplněk pokročilejších zařízení.

VRPN používá pro myš třídu `vrpn_Mouse`, pro klávesnici `vrpn_Keyboard`. Ani jedno ze zařízení neobsahuje speciální konfigurace a jediné, co lze nastavit, jsou jejich názvy. Výchozí hodnoty jsou `Mouse0`, respektive `Keyboard0`. Klávesnice je tlačítkové zařízení, které rozeznává 264 různých kláves. Stejně jako u všech tlačítkových zařízení je zaznamenáno, kdy bylo tlačítko stisknuto, ale také kdy bylo uvolněno. Myš je tlačítkové i analogové zařízení. Rozeznává tři standardní tlačítka: levé, pravé a prostřední (stisk kolečka). Analogový signál zaznamenává pohyb myši. Je reprezentován dvěma desetinnými hodnotami od 0 do 1, které představují souřadnice na obrazovce, kde souřadnice $[0,0]$ je v levém horním rohu.

6.2 TrackIR 5 v kombinaci s TrackClip

6.2.1 Obecně

Zařízení TrackIR 5 od výrobce NaturalPoint je snímač pohybu hlavy. TrackIR se připevní na monitor a nasměruje na hlavu uživatele. Ten má na sobě kšilt s připnutým TrackClip, který disponuje třemi LED diodami. Ty diody pak zachycují infračervený senzor TrackIR.

TrackIR je primárně zaměřen na připevnění na monitor, což díky nastavitelnému stojánku jde velice dobře bez ohledu na typ monitoru. K počítači je připojen standardně pomocí USB kabelu. Díky podpoře šesti stupňů volnosti (6DOF), dokáže TrackIR 5 snímat velké detaily. Snímá každý pohyb hlavy, tedy naklánění, otáčení atd. Rozsah snímače je $51,7^\circ$, což je naprosto dostačující pro své použití. Výhodou TrackClip je, že je možné jej snadno připevnit na libovolný běžný kšilt, uživatel tedy nepotřebuje připlácet za

další zařízení.

TrackIR 5 se používá v oblasti herních simulací zejména pak těch leteckých. Uživatel se může rozhlížet hlavou a nemusí k tomu používat dodatečný joystick nebo jiné zařízení, což umožní další interakci a lepší celkový dojem ze hry.[15][16]

6.2.2 VRPN

Podle dostupných informací bohužel zařízení TrackIR 5 nemá oficiální podporu protokolu VRPN. Jednou z možností, jak této podpory docílit, je vlastnoručně si ji naimplementovat pomocí TrackIR SDK, případně se tuto implementaci pokusit získat od neoficiálního vývojáře. Vzhledem k tomu, že tvorba vlastního modulu překračuje rozsah práce, od použití zařízení TrackIR 5 jsem upustil.[17]

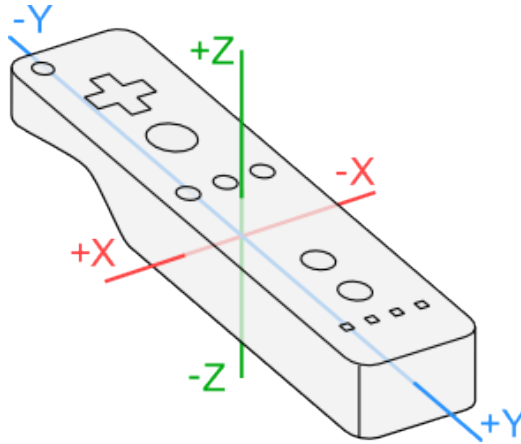
6.3 Ovladač WiiMote

6.3.1 Obecně

WiiMote neboli Wii Remote, tedy Wii ovladač, je zařízení od společnosti Nintendo. Oproti klasickému ovladači, WiiMote nereaguje nejen na stisky tlačítek, ale také na pohyb. Toto zařízení bylo navrženo jako součást Wii konzole a právě díky tomuto ovladači se konzole stala rychle populární. Bylo to proto, že reakce na pohyb dávala uživateli pocit reálné situace a unikátní požitek z každé hry, které byly většinou zaměřeny na sport. WiiMote bylo postupně vylepšováno a vznikly tak různé pokročilejší verze a dodatečná vylepšení. Jednou z těch známějších je např. Nunchuk. Díky svému potenciálu se zařízení rozšířilo mimo působnost Wii konzole.

WiiMote komunikuje s konzolí, nebo případným počítačem, pomocí technologie Bluetooth. Aby bylo možné určit náklon WiiMote, je v zařízení umístěn akcelerometr, který v radiánech měří úhel náklonu ve všech třech osách. Rozložení os je vidět na obrázku 6.1. Vpředu na ovladači je také kamera, která reaguje na infračervené záření a zaznamenává polohu WiiMote vzhledem ke zdroji. Díky tomu jsme schopni zjistit kam a jakým způsobem

uživatel zařízením pohnul. Zatímco měření náklonu je funkční standardně, pro měření pozice musíme mít k dispozici infračervený vysílač.[18][19][20]



Obrázek 6.1: Souřadnicový systém, který používá WiiMote

6.3.2 VRPN

VRPN nepodporuje WiiMote nativně jako například myš nebo klávesnici, ale je třeba přidat knihovnu zvanou `wiiuse.dll`. To se musí provést během kompilace serveru. I přesto, že to není obtížný krok, může se značně zkomplikovat např. kvůli nevhodné verzi operačního systému nebo kolizi s jinou komponentou VRPN serveru. Jakmile je VRPN server zkompilován s knihovnou `Wiiuse`, podléhá licenci GPL.[21]

Z hlediska VRPN je WiiMote ovladač považován za tlačítkové i analogové zařízení, kde analogová část snímá náklon i pozici zařízení. Zatímco se tlačítka týče, vrací VRPN server pouze jeho ID, analogová část vrací velké množství informací. Ty se skládají z hodnot udávajících náklon a pozici zařízení, stav baterie a také detaily o doplňku Nunchuck. Používá se třída `vrpn_WiiMote` a její výchozí název je `WiiMote0`. Na rozdíl od myši a klávesnice se dá Wii ovladač dále konfigurovat. Kromě jména má další parametry, které je možné nastavit. Můžeme tak specifikovat pořadové číslo hráče nebo např. přeorganizovat identifikační čísla tlačítek. Podrobnosti jsou uvedeny v konfigurační sekci.[22]

Bohužel se mi nepodařilo zkompilovat takový VRPN server, který by umožňoval zaznamenávat analogické hodnoty z WiiMote. Problémy s kompilací byly způsobeny nutností použití 64-bitové verze, která nespolečně pracovala

dobře. Z tohoto důvodu je WiiMote v ukázkách použito pouze jako tlačítkové zařízení.

6.4 Oculus Rift DK2

6.4.1 Obecně

Oculus Rift DK2 od společnosti Oculus VR patří do série zařízení HMD, zmíněných výše. Oculus Rift se skládá ze dvou částí. Tou hlavní jsou projekční brýle, které si uživatel umístí na hlavu a přichytí pomocí připojených popruhů. Díky odstínění uživatel nevidí své okolí, ale vnímá pouze projekci. Druhou částí je pohybový senzor, který se typicky umísťuje na monitor a zaznamenává pohyb brýlí – tedy uživateli hlavy. To, jak zařízení vypadá, je vidět na obrázku 6.2. Přístroj je k počítači připojen pomocí USB a HDMI.

Oculus Rift se dá poměrně snadno nakonfigurovat a jeho spuštění zvládne i méně zkušený uživatel. Jsou dva způsoby, pomocí kterých je možné Oculus Rift propojit s počítačem. Prvním z nich je tzv. Extended Display. V tomto režimu se Oculus Rift chová podobně jako druhý monitor a ve Windows tedy musíme mít nastaveno rozšířené zobrazení. Tento způsob je poměrně jednoduchý, ale nese sebou také řadu nevýhod, a proto se od jeho vývoje upustilo. Poslední verzi SDK, které tento režim podporuje je verze 0.6. Druhým způsobem je tzv. Direct Display, kde je aplikace přímo navázána na Oculus Rift. Díky tomu se vyhneme zbytečné práci s obrazovkami, nicméně některé starší aplikace tento mód nepodporují. V současné době je tento režim preferovaný.

Výsledná projekce v Oculus Rift DK2 má rozlišení 1920 x 1080 s frekvencí 75hz. To je značné zlepšení oproti předchozí verzi DK1, nicméně díky tomu také stoupají nároky na hardware. Pro plynulé zobrazení, což nejen zlepšuje dojem z virtuální reality, ale také je vhodnější pro zdravotní stav uživatele, je třeba mít dostatečné FPS, tedy počet snímků za sekundu. Doporučená hodnota je 75 FPS. Díky této skutečnosti mnoho přenosných počítačů a starších stolních počítačů Oculus Rift nepodporuje.[23][24][25]



Obrázek 6.2: Oculus Rift DK2

6.4.2 VRPN a BlenderVR

Oculus Rift nemá oficiální podporu ze strany VRPN, nicméně je k dispozici neoficiální software, který se chová jako VRPN server a umožňuje tak přenos dat ze zařízení na VRPN klienta. Přenášenými daty jsou souřadnice s pozicí zařízení vzhledem k čidlu a také hodnoty, udávající náklon brýlí. Nevýhodou tohoto software je, že se jedná již o předkompilovaný server, který není možné nijak konfigurovat ani upravovat. [26]

Použití VRPN však není nutné, protože BlenderVR zajišťuje vlastní podporou pro Oculus Rift DK1 i DK2. BlenderVR umožňuje velice pohodlnou komunikaci s tímto zařízením a díky tomu uživatel není nucen spouštět žádný dodatečný software mimo standardního runtime prostředí pro Oculus Rift. V současné době (k 15. 2. 2016) však BlenderVR nepodporuje Direct Display a proto je nutné použití starší verzi Oculus Rift SDK, která stále podporuje Extended Display. Členové týmu BlenderVR na zajištění podpory pro Direct Display pracují, mělo by se tak jednat pouze o dočasné řešení.

7 Instalace prostředí

Instalace bude popsána pro 64 bitový operační systém Windows. Já jsem vše úspěšně nainstaloval na systému Windows 7 i 8.1. V případě instalace prostředí na jiný operační systém, doporučuji prostudovat návody dostupné na stránkách projektu.

7.1 BlenderVR

Práce s BlenderVR bude popisována pro verzi 1.0 staženou 27. února 2016.

Spolu s vývojem BlenderVR se jeho instalace značně usnadnila. Dříve bylo nutné celý projekt ručně kompilovat a nyní stačí pouze projít jednoduchým instalačním průvodcem, se kterým si poradí i méně zkušený uživatel.

Nejprve je nutné si obstarat instalační soubor. Ten je k dispozici na stránkách BlenderVR a také na přiloženém DVD. Cesta k instalačnímu souboru na DVD:

```
installation/BlenderVR_setup_Win64.exe
```

Celá instalace zabere pár minut a vyžaduje občasné zásahy uživatele. Po jejím dokončení se na ploše objeví zástupce BlenderVR, pomocí něhož můžeme aplikaci spouštět. K instalaci je třeba internetové připojení.

Během práce s BlenderVR se vyvarujte používání mezer, symbolů, diakritiky atp. v názvech souborů a složek. Může to způsobovat špatně odhalitelné chyby.

7.1.1 Součásti instalace

Při instalaci BlenderVR se také spustí instalace dalších programů, které jsou nezbytné pro funkčnost BlenderVR.

Python

Python je skriptovací jazyk, ve kterém je BlenderVR napsaný. Během instalace je třeba označit, že chceme nainstalovat rozšíření pip a přidat Python do systémové proměnné `PATH`. Pokud to neuděláme během instalace, bude třeba tyto věci učinit později.

QT

Dále bude třeba nainstalovat software Qt, což je nástroj pro vytváření GUI. Jeho instalace se opět spustí automaticky.

Blender

Součástí instalace je také Blender 2.75. Po instalaci se bude nacházet v instalačním adresáři ve složce `blender`.

Ukázky

Spolu s BlenderVR získáme také řadu ukázkových aplikací a konfigurací, které nám umožní snadno otestovat většinu funkcí, které BlenderVR nabízí. Ukázky se budou nacházet ve složce `samples` a konfigurační soubory ve složce `configuration files`. Tato část instalace je volitelná a zabírá necelých 100MB na pevném disku.

7.1.2 Dokončení

Na konci instalace je třeba povolit spuštění skriptu `BlenderVR-install.bat`, který provede konfigurace. Nepovolení tohoto skriptu by vedlo ke zbytečným komplikacím.

7.2 VRPN server

VRPN server není součástí instalace BlenderVR. Je tedy třeba jej zkopírovat z DVD, kde se nachází v adresáři `installation/vrpn`. Standardní místo, pro umístění VRPN je adresář `plugins` ve složce s projektem, nicméně to není povinné.

Zkušenější uživatelé si mohou VRPN zkompilevat sami. Ve spojení s 64-bit systémem Windows se jedná se o komplikovaný a zdlouhavý proces, kterému bych se doporučil vyhnout. Soubory a programy potřebné pro kompilaci serveru jsou dostupné na DVD:

```
installation/vrpn_compile
```

7.3 Zařízení

Pro funkčnost celého projektu je také třeba připravit a nainstalovat zařízení, která se v aplikacích používají.

7.3.1 WiiMote

Ovladač Wii komunikuje pomocí Bluetooth. Pokud počítač standardně Bluetooth nepodporuje, je možné použít Bluetooth adaptér. Ve chvíli, kdy máme Bluetooth připraveno, zvolíme možnost *Přidat zařízení*. Nyní na Wiimote stiskneme tlačítka *A* a *B* současně. Tím se spustí inicializace a zařízení by se mělo objevit v nabídce. Alternativně je možné stisknout malý červený knoflík, který je umístěn pod krytem. Vybereme zařízení a přidáme jej, nyní by mělo komunikovat s počítačem.

Pro otestování funkčnosti doporučuji použít software zvaný WiimoteLib, který je k dispozici na DVD:

```
installation/WiimoteLib/WiiTest.exe
```

Pokud vše proběhlo v pořádku, po spuštění se objeví rozhraní, které bude zobrazovat jaké tlačítko bylo stisknuto, jak je zařízení nakloněno, stav baterie atd. Software nejenže otestuje funkčnost, ale také zařízení zinicizuje

a připraví pro další použití.

Občas, zejména při nečinnosti nebo po vypnutí, se stane, že zařízení přestane reagovat. V tom případě doporučuji zařízení odebrat a celý proces zopakovat.

7.3.2 Oculus Rift DK2

Nejprve propojíme pohybový senzor s brýlemi a umístíme jej na monitor. Oculus Rift připojíme k počítači pomocí USB a HDMI kabelu, senzor pouze pomocí USB kabelu. Po připojení by se na brýlích měla rozsvítit oranžová kontrolka. Poté je potřeba nainstalovat software, který je k dispozici na stránkách Oculus nebo na DVD. Je nutné použít verzi 0.6, která stále podporuje režim Extended Display. V případě nefunkčnosti je možné zkusit použít starší verzi 0.5, nicméně starší verze by již mohly způsobovat problémy. Umístění na DVD:

```
installation/oculus_runtime_sdk_0.6.0.1_win.exe
```

Pro úspěšnou instalaci a následné zprovoznění Riftu je třeba mít aktualizovaný systém Windows a ovladače grafické karty¹. V případě problémů doporučuji provést tyto aktualizace a také zkontrolovat, zda grafická karta Oculus Rift podporuje.

V případě úspěšné instalace se dole na panelu objeví ikona Oculus Riftu, pomocí níž je možné otevřít rozhraní. Zde by měl být text *Oculus Rift DK2: Ready*, pokud ne, je třeba zkontrolovat správnost instalace a zapojení. Pro pokračování je třeba přidat uživatele, nastavit jeho výšku a také je doporučeno nastavit ostatní parametry jako je vzdálenost očí od krku atd. pro co nejlepší výsledek.

Poté je možné stisknout *Show Demo Scene* pro spuštění ukázky. Pokud vše funguje správně, měla by se již zahájit projekce do zařízení.

¹Pro Windows 7 je vyžadován Service Pack 1, které je možné stáhnout na oficiálních stránkách Microsoft.

8 Konfigurace a funkční kód

8.1 Rozbor konfiguračního souboru *.xml

Konfigurace BlenderVR spočívá v editaci konfiguračního souboru ve formátu XML. Konfigurační soubory jsou připraveny pro jednotlivé ukázky, nicméně i tak je nutné je editovat, protože je třeba změnit cesty dle vlastní instalace, případně měnit nastavení zobrazení dle dostupných možností.

V této sekci bude podrobně popsán celý konfigurační soubor. Pomocí něj je možné změnit zobrazení scény, přidávat počítače a uživatele nebo přidávat posluchače periferních zařízení.

Celá konfigurace je uzavřena v sekci `blendervr`

```
<blendervr>
```

8.1.1 Sekce starter

První z podsekcí je sekce `starter`. V záhlaví je uvedena cesta ke spouštěcímu souboru aplikace Blender, tedy cesta k `blender.exe`. Při zachování standardní souborové struktury, se bude tento soubor nacházet v instalačním adresáři ve složce `blender`. Výsledek může vypadat takto:

```
<starter blender="D:/BlenderVR/blender/blender.exe">
```

Následuje seznam použitelných zobrazení. Díky tomu můžeme měnit zobrazení naší aplikace. Můžeme například specifikovat rozložení na několik monitorů, zobrazení na Oculus Rift, malé okno pro testovací účely atd. Do atributu `name` uložíme jméno, které se bude zobrazovat v aplikaci a mezi značky `config` vložíme seznam obrazovek (`screen`), které jsou v daném zobrazení použity. Parametry těchto obrazovek definujeme dále.

```
<config name="single"> single </config>  
<config name="dual"> wallLeft, wallRight </config>  
</starter>
```

8.1.2 Sekce users

V sekci `users` je seznam všech uživatelů, kteří v aplikaci figurují. Každý uživatel může mít nastavené jiné zobrazení a také může být asociován s jiným periferním zařízením. Díky tomu může každý uživatel používat jiný ovladač, být připojen na samostatný tracker, používat vlastní Oculus Rift atd. Takto je možné vytvořit hru či simulaci pro více hráčů, kteří spolu mohou interagovat, což značně rozšiřuje naše možnosti. Často je však potřeba propojení více počítačů, aby bylo možné obsloužit větší množství zařízení. Pro naše potřeby si ale vystačíme s jedním uživatelem, kterého symbolicky nazveme *user A*.

V sekci `users` je také nepovinná sekce `behavior`, která umožňuje nastavit pozici uživatelů a jejich vzdálenost očí. Výchozí hodnotou pro vzdálenost očí je 0.06 m a pro pozici uživatele to pak je (0.0, 0.0, 0.0). Nastavení vzdálenosti očí může být důležité pro některá 3D zobrazení, která se specifikují v sekci `screens` (viz dále), obvykle si ale vystačíme s výchozí hodnotou. Tyto hodnoty mění nastavení pro všechny uživatele, rozdílné specifikace jednotlivých uživatelů nejsou dostupné.

```
<users>
  <behavior eye_separation='0.05'>
    <default_position>1.0, 0.0, 0.0</default_position>
  </behavior>
  <user name="user A"/>
</users>
```

8.1.3 Sekce computers

Chování jednotlivých počítačů musí být definováno v sekci `computers`. Podobně jako v případě uživatelů, budeme používat pouze jeden počítač. Také se zde nachází nastavení `system`, kde je nutné v sekci `daemon` specifikovat cestu k operačnímu systému Windows.

```
<computers>
  <system>
    <daemon>
      <environment>SystemRoot=C:/Windows</environment>
    </daemon>
  </system>
</computers>
```

Také je třeba uvést cestu k `blenderplayer.exe`, nachází se typicky na stejném místě jako `blender.exe`. V případě použití VRPN je poté třeba definovat knihovnu `vrpn.pyd`, která se nachází ve složce `python` v adresáři s VRPN. Stejně jako v případě `blender.exe`, uvedení neplatných cest povede k chybě při spouštění programu.

```
<blenderplayer executable="D:/BlenderVR/blender/
  blenderplayer.exe"/>
<library path="D:/BlenderVR/plugins/vrpn/python"/>
</system>
```

Nyní můžeme vyjmenovat počítače. Jak již bylo zmíněno, pro naše účely si vystačíme s jedním, jedná se tedy pouze o formalitu.

```
<computer name="Any" hostname="*" />
</computers>
```

8.1.4 Sekce screens

Následuje sekce `screens`, tedy sekce obrazovek. Této části je velmi důležité porozumět, protože značně ovlivní celé zobrazení a bude třeba ji upravovat dle vlastních možností zobrazení.

```
<screens>
```

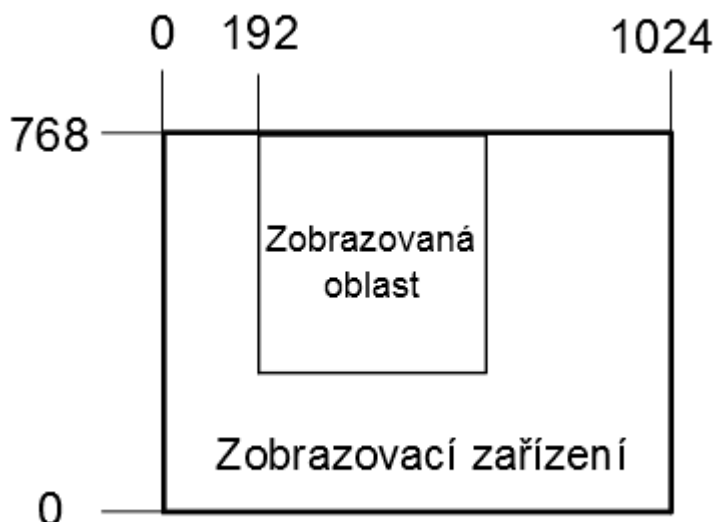
Každá obrazovka musí mít vlastní jméno a počítač, na který se váže. Tyto hodnoty samozřejmě musí být v souladu s hodnotami uvedenými v sekci `starter` a `computers`. Ve zbytku sekce je poté detailní nastavení zobrazení. Celá sekce se jmenuje `screens`, jednotlivé obrazovky pak mají samostatné podsekce `screen`.

```
<screen name="wallLeft" computer="Any">
```

V podsekci `display` je atribut `options`, kde je možné specifikovat nastavení zobrazení. Jedním z použitelných parametrů je `-w`, takto můžeme specifikovat rozlišení a posun okna aplikace. Předpokládejme tedy například, že máme obrazovku s rozlišením 1024x768 a chceme zobrazit okno naší aplikace o velikosti 600x480. Chceme, aby se okno nacházelo ve vrchní části displeje a bylo zleva odsazené o 192 pixelů. Abychom tohoto dosáhli, použijeme následující nastavení:


```
<display options = "-w 600 480 192 768">
```

První dvě hodnoty udávají rozlišení zobrazovaného okna, druhé pak odsazení od levého horního rohu. Výsledek je znázorněn na obrázku 8.1.



Obrázek 8.1: Ukázka zobrazení dané konfigurace s použitím parametru -w

Pokud bychom chtěli aplikaci zobrazit na celou obrazovku, musíme použít parametr -f, za ten je také možné uvést rozlišení.

```
<display options = "-f">
```

Další možností je parametr -s, který slouží k přepnutí zobrazení do Sterea 3D. Je zde možnost výběru z několika módů, které upravují zobrazení typicky pro různé druhy 3D brýlí. Jedná se např. o -s anaglyph, který upraví zobrazení pro červeno-modré 3D brýle, -s sidebyside, který zobrazí levou i pravou stranu pro HMD zařízení atd. Toto nastavení však musí korespondovat s grafickým bufferem (viz níže).

```
<display options="-f 1920 1080 -s sidebyside">
```

Možností nastavení je celá řada. Je možné nastavit anti-aliasing, zobrazit různé parametry z BGE, natočit obraz atd. Pro více podrobností o nastavení zobrazení je možné zapnout nápovědu pro Blender Player, tedy jej spustit s parametrem -h (příkazem blenderplayer.exe -h v adresáři blender).

Dále je třeba definovat grafický buffer, který obsahuje tři parametry: `buffer`, `user` a `eye`. Uživatele `user` nastavíme dle uživatele, o kterého se jedná, v našem případě `user A`. Zbylé dva parametry závisí na tom, zda jsme použili 3D stereo zobrazení. Pokud ne, `buffer` nastavíme na hodnotu `mono` a `eye` na `middle`.

```
<graphic_buffer buffer="mono" user="user A" eye="middle"/>
```

Vzhledem k tomu, že se jedná o výchozí hodnoty, je v tomto případě možné tyto parametry vynechat.

```
<graphic_buffer user="user A"/>
```

Pokud používáme Stereo 3D, je třeba použít dva grafické buffery. U jednoho z nich se `buffer` i `eye` bude rovnat `left` a u druhého `right`.

```
<graphic_buffer buffer="left" user="user A" eye="left"/>
<graphic_buffer buffer="right" user="user A" eye="right"/>
</display>
```

Tím podsekcce `display` končí a následuje podsekcce `wall`, kde nastavíme pozici a velikost dané obrazovky. Zde jsou položky `corner`, kde každá z nich popisuje jeden roh obrazovky. Rohy jsou definovány pouze tři, čtvrtý již není nutný. Každý roh je určen souřadnicemi a vychází se ze skutečnosti, že se uživatel nachází na pozici (0.0, 0.0, 0.0). Je nutné podotknout, že první hodnota je šířka, druhá výška a třetí hloubka, tedy vzdálenost od obrazovky.

Zadáním těchto souřadnic určíme i velikost obrazovek. Tyto hodnoty musejí být nastaveny tak, aby korespondovaly s jednotkami v Blenderu. Pokud tedy chceme reálné zobrazení, což je typicky náš cíl ve virtuální realitě, měli bychom mít scénu v Blenderu udělanou tak, aby její hodnoty odpovídaly skutečnosti, tedy hráč je např. 1,8 m vysoký, dveře 0,6 m široké atd. Pokud toto naše scéna splňuje, můžeme zadat skutečné rozměry našeho zobrazovacího zařízení. Jestliže rozměry ve scéně skutečnosti neodpovídají, musíme spočítat poměr a podle něj upravit zadané rozměry.

Princip by měl být patrný z následujícího příkladu. Pokud bychom měli dvě čtvercové obrazovky o velikosti 1 m a chtěli je zobrazit vedle sebe s tím, že by uživatel byl v jejich středu vzdálený 70 cm, konfigurace by vypadala takto:

```
<wall>
```

```
<corner name="topRightCorner">0.0, 0.5, -0.7</corner>
<corner name="topLeftCorner">-1.0, 0.5, -0.7</corner>
<corner name="bottomRightCorner">0.0, -0.5, -0.7</corner>
</wall>
```

Toto konkrétní nastavení by odpovídalo levé obrazovce. Její pravá strana by tedy byla ve středu oblasti. Pravá obrazovka by poté vypadala takto:

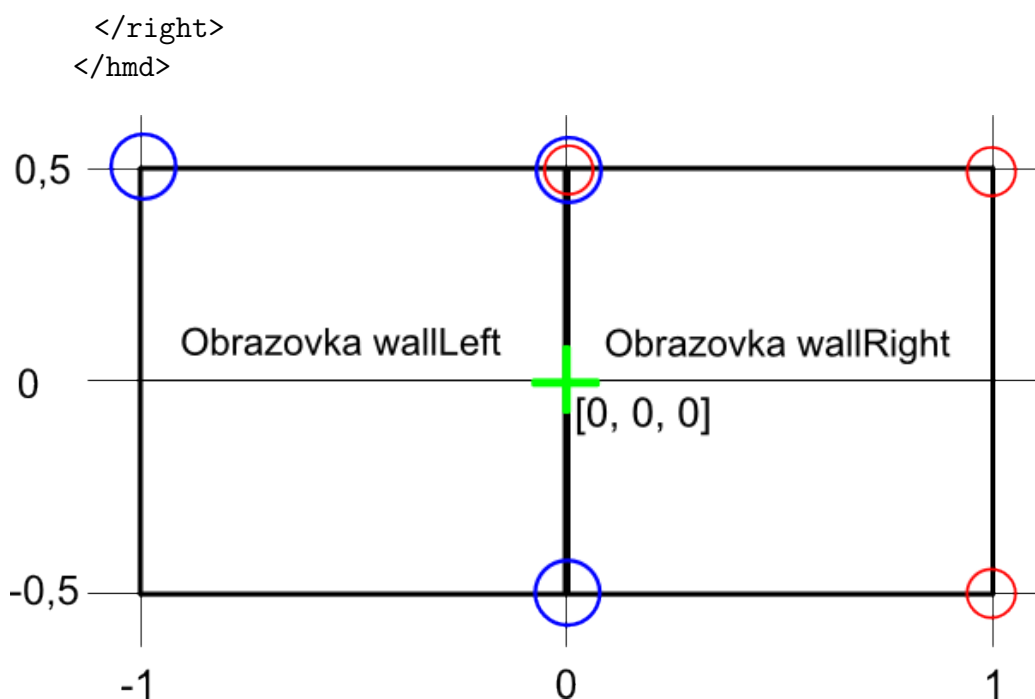
```
<wall>
  <corner name="topRightCorner">1.0, 0.5, -0.7</corner>
  <corner name="topLeftCorner">0.0, 0.5, -0.7</corner>
  <corner name="bottomRightCorner">1.0, -0.5, -0.7</corner>
</wall>
```

Tato sekce `wall` by samozřejmě byla v jiné sekci `screen`, v každé sekci je pouze jedna. Zde je ukázána pro přehlednost, aby byla obě nastavení v dokumentu u sebe.

Z obrázku 8.2 by měl být patrný význam jednotlivých hodnot a jejich projevení se na výsledném zobrazení. Střed zobrazované oblasti je zvýrazněn zeleným křížem, tam se nachází také hlava uživatele (vzdálená 70 cm). Rohy levé obrazovky jsou zvýrazněny modře. Rohy pravé obrazovky jsou poté červené. Celá sestava by tedy byla 2x1 m velká. Pohled na scénu je určen umístěním kamery v našem modelu (souboru *.blend). Je důležitá její pozice a také její natočení, nicméně její velikost nemá na zobrazení žádný vliv. V editačním režimu v Blendru je kamera zastoupena 3D objektem kamery, pomocí něhož můžeme lépe určit vhodný úhel náklonu, ale je třeba brát zřetel na to, že její pozice je určena referenčním bodem nikoliv její animovanou obrazovkou.

V případě, že bychom chtěli použít nějaké HMD zařízení, např. Oculus Rift, dané sekce by nebyly nazvány `wall`, ale `hmd`. Také by bylo třeba uvést model zařízení, se kterým pracujeme. Vzhledem k podstatě těchto zařízení, obsahuje každá sekce `hmd` podsekcí pro levé a pravé oko, tedy `left` a `right`. Je také potřeba využít 3D stereo buffer s režimem Side by Side.

```
<hmd model="oculus_dk2">
  <left>
    standardní definice tří rohů popsána výše ...
  </left>
  <right>
    standardní definice tří rohů popsána výše ...
```



Obrázek 8.2: Ukázka konfigurace dvou obrazovek popsaná v sekcích `wall`

Takto můžeme mít definováno různé množství obrazovek, ale každá z nich musí mít zmíněné náležitosti. Pokud již máme definovány všechny obrazovky, končí i celá sekce `screens`.

```

  </screen>
</screens>

```

Je vhodné do jednoho konfiguračního souboru zahrnout více zobrazení (např. jedna obrazovka, 3 obrazovky, Oculus Rift atd.), abychom zbytečně nemuseli vytvářet a vždy otevírat jiný konfigurační soubor.

8.1.5 Sekce `plugins`

Pokud jsme spokojeni s nastavením našich obrazovek následuje nepovinná sekce `plugins`. Sem můžeme přidat různé nadstavby. Jedná se o funkce, které umožní přidání speciálních zařízení, která se používají ve virtuální realitě. Především nás bude zajímat použití VRPN a Oculus Rift DK2.

```

<plugins>

```

V případě VRPN je nutné specifikovat jednotlivé třídy zařízení a nastavit jméno funkce, která je bude spravovat v souboru *.processor.py. V případě síťové konfigurace také IP adresu nebo hostname serveru.

```
<vrpn>
  <analog device="Mouse0" host="localhost" processor_method
    ="mouseAnalog"/>
  <button device="Mouse0" host="localhost" processor_method
    ="mouseButton"/>
  <button device="Keyboard0" host="localhost" processor
    method="keyboard"/>
</vrpn>
```

Takto je nastaveno, že veškerá analogová data, která klient obdrží od myši, bude zpracovávat funkce mouseAnalog(), tlačítka mouseButton() atd. Názvy zařízení musí korespondovat s nastavením vrpn.cfg popsáném níže.

Konfigurace Oculus Rift je poměrně triviální, protože toto řeší interně BlenderVR pomocí funkce user_position. Stačí tedy přidat jméno uživatele i počítače a nastavit zmíněnou funkci.

```
<oculus_dk2>
  <user viewer="user A" computer="Any" processor_method
    ="user_position"/>
</oculus_dk2>
</plugins>
```

Konfiguraci ukončíme uzavřením sekce blendervr.

```
</blendervr>
```

8.2 Soubor *.processor.py

Funkční kód dané aplikace se nachází v souboru *.processor.py. Jedná se o skript napsaný v jazyce Python, kde obvykle bývá vyřešena interakce uživatele s aplikací pomocí periferních zařízení. Tento soubor mává zpravidla stejný název jako simulační soubor. Pokud se tedy simulace bude jmenovat simple.blend, procesor se bude jmenovat simple.processor.py.

Skript importuje knihovnu zvanou bge. Díky tomu je možné psát stejný

kód, jaký bychom psali v BGE. Je tedy možné získat odkaz na scénu a její objekty a dále s nimi pracovat pomocí kódu.

Hlavní složkou celého skriptu je třída `Processor`. Nachází se zde funkce `__init__`, což je prakticky konstruktor a má tedy na starosti inicializační práce. Typicky zde tedy získáváme zmíněné odkazy na scénu a objekty a provádíme další přípravné akce.

Dále se zde musí nacházet funkce použité v sekci `plugins` v konfiguračním souboru. Jedná se o obslužné funkce, které se spustí v případě, že aplikace získá data od nějakého zařízení. Pokud jsme tedy například nastavili, že analogový signál myši bude zpracovávat funkce zvaná `mouseAnalog()`, musíme ji zde specifikovat. Prvním parametrem funkce je tzv. `self`, což je objekt umožňující práci s atributy instance. Druhým parametrem je asociativní pole, které obsahuje informace obdržené od VRPN serveru.

V následujícím kousku kódu je vidět obsah jednoduché třídy `Processor`, díky které je možné pomocí myši hýbat kuličkou na scéně skrze VRPN server. Během inicializace se získá odkaz na objekt "Sphere" ze scény a vypíše se do konzole BlenderVR řetězec: "Processor - init". Následuje funkce `mouseAnalog()`, která se spustí vždy, když VRPN obdrží analogový signál od myši.

Redukovaná část kódu ze skriptu použitého v ukázce s myší a klávesnicí
- `kmDemo.processor.py`

```
class Processor(blendervr.processor.getProcessor()):
    def __init__(self, parent):
        super(Processor, self).__init__(parent)

    if self.BlenderVR.isMaster():
        # This line enables synchronization with the master
        # window in case of multiple windows
        sync = self.BlenderVR.getSceneSynchronizer()
        sync.getItem(bge.logic).activate(True, True)

    # BlenderVR console log
    self.logger.info("Processor - init")

    # get reference to Blender scene
    self._scene = bge.logic.getCurrentScene()

    # get refernce to Blender object Sphere
```

```
self._sphere = scene.objects["Sphere"]

# sensitivity / range of a mouse movement
self._s = 10.0

def mouseAnalog(self, info):
    # get coordinates and save them to associative array [x,y]
    raw_data = info['channel']
    data = {'x' : raw_data[0], 'y' : raw_data[1]}

    # each move of a mouse will change position of the sphere
    # sphere will move on screen up, down, left or right
    # speed of a movement is based on the given sensitivity
    self._sphere.localPosition[0] = (data['x'] - 0.5) * s
    self._sphere.localPosition[2] = -((data['y'] - 0.5) * s)
```

Nyní, pokud spustíme server a poté standardním způsobem spustíme náš projekt (viz dále), po spuštění se do konzole vypíše `Processor - init` a každý pohyb myši pohne s kuličkou na scéně.

Tímto způsobem je možné vytvářet funkcionalitu libovolných rozměrů. Mimo znalosti programovacího jazyka Python je také třeba být obeznámen s funkcemi BGE, které je možné nalézt v dokumentaci Blenderu. S celým skriptem je samozřejmě možné pracovat standardním způsobem a vytvářet tak další funkce či třídy dle potřeby pro zpřehlednění celého kódu. Funkcionalitu, která nesouvisí s externím zařízením, je možné zahrnout přímo do Blender scény.

8.3 Konfigurace VRPN

VRPN server se z větší části nastaví v průběhu kompilace, tam definujeme podporu zařízení a různých funkcí. Jakmile již máme server zkompileovaný, je třeba pouze měnit soubor `vrpn/server/vrpn.cfg`, ve kterém je nastaveno, která zařízení server poslouchá. Tento soubor standardně obsahuje všechna podporovaná zařízení, nicméně součástí této práce je zkrácený konfigurační soubor, který obsahuje pouze zařízení, která byla použita.

Každý řádek souboru představuje jedno zařízení. V případě, že chceme dané zařízení povolit nebo naopak zakázat, je třeba daný řádek odkomento-

vat, respektive zakomentovat (přidáním nebo odebráním znaku #). V řádce je uvedena třída daného zařízení (např. `vrpn_Mouse`), název zařízení, který je typicky odvozen z názvu třídy (např. `Mouse0`), a poté z různého počtu parametrů, které závisí na daném zařízení. Význam parametrů je popsán v komentářích u jednotlivých zařízení.

Příkladem může být konfigurace `WiiMote`. Prvním parametrem, stejně jako u všech zařízení, je název, v tomto případě `WiiMote0`. Následuje číselná hodnota, která určuje číslo hráče, v našem případě se tedy jedná o prvního hráče. Dále zde jsou tři logické hodnoty, které nabývají hodnot 0 (`false`) a 1 (`true`). První z nich je tzv. `useMotionSensing`, který umožňuje zjišťování pozice zařízení. Dalším je `useIR`, který určuje, zda se má použít infračervený senzor, který se používá k detekci pohybu. Vzhledem k tomu, že v této práci pozici ovladače zjišťovat nepotřebujeme, nastavíme obě hodnoty na 0, tím tyto funkce vypneme. Další hodnota určuje, zda chceme změnit ID jednotlivých tlačítek tak, aby dávala pro uživatele větší smysl. Jedná se o užitečnou funkci, proto je zde povolena. Poslední parametr je nepovinný a obsahuje fyzickou adresu našeho Bluetooth adaptéru. Toto nastavení je možné použít pouze v unixových systémech, proto jej zde nepoužijeme.

Příklad : `vrpn_WiiMote WiiMote0 1 0 0 1`

8.4 Konfigurace síťového propojení

V praxi se často setkáme se situací, kdy není možné mít všechna zobrazovací zařízení připojena k jednomu počítači. Důvodem bývá nedostatečný počet konektorů, příliš velká vzdálenost jednotlivých zařízení od počítače atd. Aby bylo možné se s touto situací vypořádat, BlenderVR nabízí možnost síťového propojení. I přesto, že se práce touto konfigurací nebude příliš zabývat, je zde lehce celá situace nastíněna. Podrobnější informace o této konfiguraci včetně příkladu jsou dostupné na stránkách BlenderVR.

Celá konfigurace funguje na principu *master – slave*. Jeden z počítačů je *master* – hlavní a ostatní jsou *slave* – vedlejší. V konfiguraci bude třeba přidat několik věcí, jednou z nich je přidání počítače do sekce `computers` pro každý obsluhovaný počítač. Jednotlivé počítače se již nebudou jmenovat "Any", ale nějakým praktickým jménem např. "Left". Také je třeba definovat IP adresu daného stroje. Sekce `<system>` je rozšířena o parametr `root`, který obsahuje cestu k BlenderVR, a také o parametr `anchor`, kde je specifikována cesta

k aplikacím. Dále přibude sekce `login`, která řídí připojení pomocí SSH. Bude zde také uvedeno, zda je dané zařízení *master* nebo *slave* a umístění Pythonu.

```
<computer name="Left" hostname="192.168.0.1">
  <system root="Z:/BlendVR/source" anchor="Z:/BlendVR/demos">
    ...
    <login remote_command="ssh master@192.168.0.1" python=
      "C:/PY/Python.exe"/>
  </system>
</computer>
```

Dle mého názoru BlenderVR řeší tuto situaci poměrně elegantně a není potřeba příliš měnit náš konfigurační soubor, což je samozřejmě velká výhoda.

9 Závěr

Ukázalo se, že BlenderVR je zajímavý software disponující různými funkcemi s širokými možnostmi nastavení. Ovládání programu je poměrně jednoduché, avšak uvítal bych zjednodušení konfiguračních souborů a jejich integraci do samotného programu např. formou nastavení. Projekt stále obsahuje řadu chyb a nedokonalostí, což se u začínajícího nekomerčního nástroje dá očekávat, autoři jsou však ochotni s případnými problémy pomoci.

Práce s VRPN v kombinaci s 64-bit systémem Windows a BlenderVR nebyla snadná, jelikož bylo třeba pracovat s velice konkrétním nastavením, se kterým neměli zkušenosti ani členové BlenderVR týmu. Spojení se nakonec podařilo uskutečnit, bohužel bez možnosti přenosu analogových informací z WiiMote. Zařízení se tak chovalo jen jako běžný tlačítkový ovladač. Pro zařízení TrackIR 5 bohužel není oficiální podpora pro VRPN, proto jsem od jeho použití upustil.

Oculus Rift DK2 naopak fungoval dobře a díky tomu, že BlenderVR poskytuje pro HMD zařízení zvláštní podporu bez nutnosti použití VRPN protokolu, jej nebylo složité do projektu zahrnout. Nevýhodou však zůstává nutnost použití zastaralého Extended Mode a s ním spojené potíže s nastavením cílové obrazovky. Menším nedostatkem je také nižší kvalita zobrazení, která je způsobena zkreslením. Vývojáři o těchto problémech ví a pracují na jejich vyřešení, jejich trvání by tak mělo být jen dočasné.

Jednou z nejtěžších věcí na této práci bylo vyrovnat se s vývojem BlenderVR. Často jsem se tak dostával do situace, kdy jsem investoval čas do řešení určitého problému, který byl posléze opraven či byl na stránky projektu přidán přehledný návod pro jeho vyřešení. Naopak příjemnější stránkou věci bylo vidět dopad mé činnosti na vývoj celého projektu. Dané opravy či návody se často objevovaly na základě mých otázek a připomínek. Dle mého názoru se tak zlepšila podpora pro Windows, která byla původně jen okrajová.

Přehled zkratk

6DOF Six Degrees of Freedom (Šest stupňů volnosti)

BGE Blender Game Engine (Část modelovacího nástroje Blender, která umožňuje renderování v reálném čase)

CAVE Cave Automatic Virtual Environment (Prostředí pro virtuální realitu složené z několika stěn)

FPS Frames Per Second (Počet snímků za vteřinu)

GPL General Public License (Všeobecná veřejná licence)

HMD Head-Mounted Devices (Zařízení, která se připevňují na hlavu, např. Oculus Rift)

VR Virtual Reality (Virtuální realita)

VRPN Virtual Reality Peripheral Network (Protokol pro komunikaci s periferními zařízeními ve virtuální realitě)

Literatura

- [1] *Virtual Reality Society* [online]. [cit. 2016-02-29]. <<http://www.vrs.org.uk/>>
- [2] Muhanna A. Muhanna: *Virtual reality and the CAVE: Taxonomy, interaction challenges and research directions*. Journal of King Saud University - Computer and Information Sciences, Volume 27, Issue 3, July 2015, Pages 344-361, ISSN 1319-1578
- [3] *Blender Manual* [online]. [cit. 2016-03-03]. <<https://www.blender.org/manual/>>
- [4] *The GNU General Public License* [online]. [cit. 2016-03-03]. <<http://www.gnu.org/copyleft/gpl.html>>
- [5] *Blender Game Engine (Vlastní překlad)* [online]. [cit. 2016-03-03]. <https://www.blender.org/manual/game_engine/introduction.html>
- [6] *BlenderVR* [online]. [cit. 2016-03-15]. <<http://blendervr.limsi.fr>>
- [7] Brian F.G. Katz, Dalai Q. Felinto, Damien Touraine, David Poirier-Quinot, Patrick Bourdot: *BlenderVR: Open-source Framework for Interactive and Immersive VR*. LIMSI-CNRS, Campus Universitaire d'Orsay, Orsay, France, IEEEVR 2015
- [8] *OSC* [online]. [cit. 2016-03-30]. <<http://opensoundcontrol.org/introduction-osc>>
- [9] *VRPN* [online]. [cit. 2016-02-10]. <http://www.cs.unc.edu/~taylor/cyberinfrastructure/vrpn_html/index.html>
- [10] *VRPN GitHub Wiki* [online]. [cit. 2016-02-10]. <<https://github.com/vrpn/vrpn/wiki>>

- [11] Taylor,II, Russell M. and Hudson, Thomas C. and Seeger, Adam and Weber, Hans and Juliano, Jeffrey and Helser, Aron T.: *VRPN: A Device-independent, Network-transparent VR Peripheral System*. Procedia Technology, Elsevier, 2012
- [12] *VRPN - New Device class* [online]. [cit. 2016-02-10]. <http://www.cs.unc.edu/~taylorr/cyberinfrastructure/vrpn_html/new_Device_class.html>
- [13] *Client side VRPN Devices* [online]. [cit. 2016-02-10]. <<https://github.com/vrpn/vrpn/wiki/Client-side-VRPN-Devices>>
- [14] *PHANTOM Omni Haptic Device* [online]. [cit. 2016-03-12]. <<http://www.dentsable.com/haptic-phantom-omni.htm>>
- [15] *TrackIR 5* [online]. [cit. 2016-03-02]. <<http://www.naturalpoint.com/trackir/products/trackir5/>>
- [16] *TrackIR 5 manual* [online]. [cit. 2016-03-02]. <http://www.naturalpoint.com/trackir/documents/TrackIR-software-5_4_0-Manual.pdf>
- [17] *NaturalPoint Forum - VRPN Support* [online]. [cit. 2016-03-02]. <<https://forums.naturalpoint.com/viewtopic.php?p=46886>>
- [18] *WiiMote* [online]. [cit. 2016-03-10]. <<http://wiibrew.org/wiki/Wiimote>>
- [19] *WiiMote - Nunchuck* [online]. [cit. 2016-03-10]. <http://wiibrew.org/wiki/Wiimote/Extension_Controllers/Nunchuck>
- [20] *WiiMote axis - Image* [online]. [cit. 2016-03-10]. <http://wiibrew.org/wiki/File:Wiimote_axis2.png#filelinks>
- [21] *WiiUse* [online]. [cit. 2016-03-10]. <<http://public.vrac.iastate.edu/vancegroup/docs/wiiuse/index.html>>
- [22] *Tutorial - use VRPN* [online]. [cit. 2016-03-10]. <<http://www.vrgeeks.org/vrpn/tutorial---use-vrpn>>
- [23] *Oculus Rift - User Documentation* [online]. [cit. 2016-03-31]. <<https://product-guides.oculus.com/en-us/documentation/rift/latest/>>

-
- [24] *Oculus Rift - Developer Documentation* [online]. [cit. 2016-03-31]. <<https://developer.oculus.com/documentation/intro-vr/latest/concepts/book-bp/>>
- [25] *Oculus Rift - Image* [online]. [cit. 2016-03-31]. <<https://dbvc4uanumi2d.cloudfront.net/cdn/4.5.24/wp-content/themes/oculus/img/order/dk2-product.jpg>>
- [26] *Unofficial Rift To VRPN Server* [online]. [cit. 2016-03-31]. <<http://projects.ict.usc.edu/mxr/diy/vrpn/>>
- [27] *Tutorials for Blender 3D - Mouse Look* [online]. [cit. 2016-04-15]. <http://www.tutorialsforblender3d.com/Game_Engine/MouseLook/MouseLook_First_1.html>

A Spouštění a obsluha

A.1 BlenderVR

A.1.1 Spuštění aplikace

Aplikaci je možné pohodlně spouštět pomocí ikony na ploše, respektive souborem `BlenderVR.bat` nacházejícím se v instalačním adresáři.

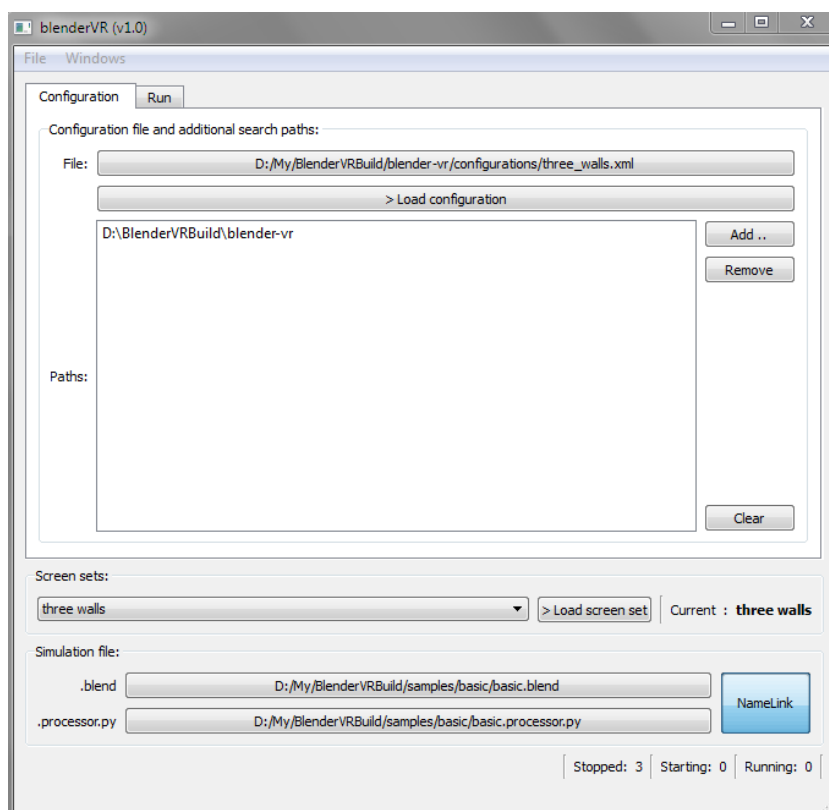
A.1.2 Obsluha

Po spuštění programu, se již navigujeme pomocí uživatelského rozhraní, které je vidět na Obrázku A.1.

Vybereme záložku **Configuration** a položku **File**, tím nastavíme konfigurační soubor. Poté klikneme na tlačítko **Load Configuration**, aby se data uložila do vnitřní paměti aplikace. Abychom otestovali, zda je konfigurace správná, přepneme se na záložku **Run**. Pokud v našem souboru byly nějaké chyby, program nás na to upozorní (informace se objeví ve výpisu ve střední části). Jestliže je vše v pořádku, můžeme pokračovat dál.

Nyní se zaměříme na nastavení obrazovek **Screen sets**. Nabídka tohoto seznamu přímo koresponduje s nastavením v konfiguraci. Vybereme tedy nějaké nastavení (pro začátek doporučuji konfiguraci s jednou obrazovkou např. `basic`) a klikneme na tlačítko **Load screen set**, aby se nastavení načetlo. Nyní byste již místo **Current: None** měli vidět např. **Current: basic**.

Ještě nám zbývá vybrat simulační soubor vytvořený v Blenderu. Na DVD se všechny ukázky nacházejí ve složce `demos`, doporučuji vyzkoušet ukázkou zvanou `basic`. V části **Simulation file** tedy vybereme nějaký soubor `*.blend`, např. `basic.blend` ze složky `basic`. Pokud je vybrána možnost **NameLink**, měl by se automaticky připojit i soubor se skriptem (pokud se jmenují shodně). Jinak můžeme kliknout na nejspodnější tlačítko a vybrat soubor se stejným jménem, který má ale koncovku `.processor.py`, tedy `basic.processor.py`.



Obrázek A.1: Uživatelské rozhraní BlenderVR

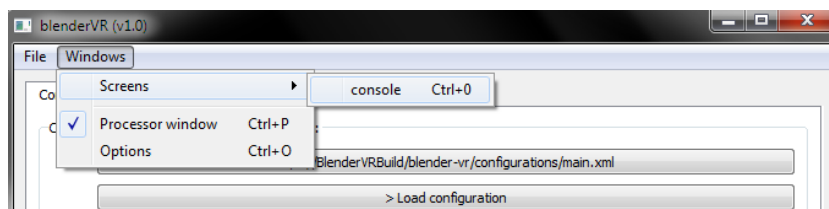
Teď můžeme náš projekt spustit. Uděláme to stisknutím tlačítka **Start** (tlačítkem **Stop** se simulace vypne). Pokud vše proběhlo v pořádku, naběhne okno Blender Player a naše aplikace se spustí. Details o této ukázce jsou dostupné v příloze s ukázkami.

A.1.3 Řešení problémů

Pokud se stane, že i přesto, že jste editovali model *.blend, se v BlenderVR stále nic nemění, smažte dočasný soubor *.vr.blend, který má stejné jméno jako název použité simulace a nachází se na témže místě. Při editaci konfiguračního souboru je třeba opět kliknout na **Load Configuration** (pozor na změny obrazovek, pak je třeba stisknout i **Load screen set**).

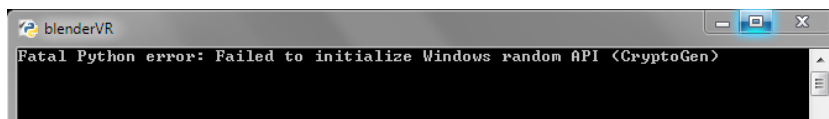
Jestliže se z nějakého důvodu nepodaří aplikaci vůbec spustit, doporučuji vyzkoušet tyto možnosti:

- Zkontrolujte, zda jsou v konfiguračním správně uvedeny cesty k souborům `blender.exe` a `blenderplayer.exe`. V případě, že používáte VRPN, tak také zkontrolujte cestu k adresáři `python`, kde se nachází soubor `vrpn.pyd`.
- Problém může způsobovat mezera nebo neobvyklý znak v cestě nebo názvu souboru použitým v aplikaci, proto se jím snažte vyvarovat.
- Zkuste jinou simulaci, jiný konfigurační soubor nebo jiné nastavení obrazovek.
- Zjistěte, zda je v pořádku Váš konfigurační soubor, zejména ověřte, že máte správně uvedenou cestu k `blenderplayer.exe`, `blender.exe` a také k Windows.
- Při uvádění cest zkuste změnit jednoduchá lomítka na zpětná lomítka nebo naopak, případně použít dvojitá lomítka.
- Vypněte a znovu zapněte aplikaci, případně i restartujte počítač.
- Zkontrolujte, zda je vše korektně nainstalováno, zejména Python a jeho součásti. Případně zkuste celou instalaci zopakovat.
- Zkontrolujte výpisy aplikace. Je zde výpis v hlavním okně, výpis v příkazové řádce, která se otevřela společně s aplikací (případně z ní byla aplikace spuštěna) a také, pokud kliknete na záložku Windows - Screens a vyberete danou obrazovku (např. `console`), otevře se další výpis (viz Obrázek A.2). Pokud zde bude nějaký podezřelý text, ale Vy si s ním nebudete vědět rady, doporučuji se pokusit vyhledat daný text na internetu zejména pak na stránkách BlenderVR projektu, případně kontaktovat autory a poslat jim Vaše výpisy. Pozor, toto okno je třeba mít otevřené před tím, než se stiskne tlačítko **Start**, jinak bude okno prázdné.



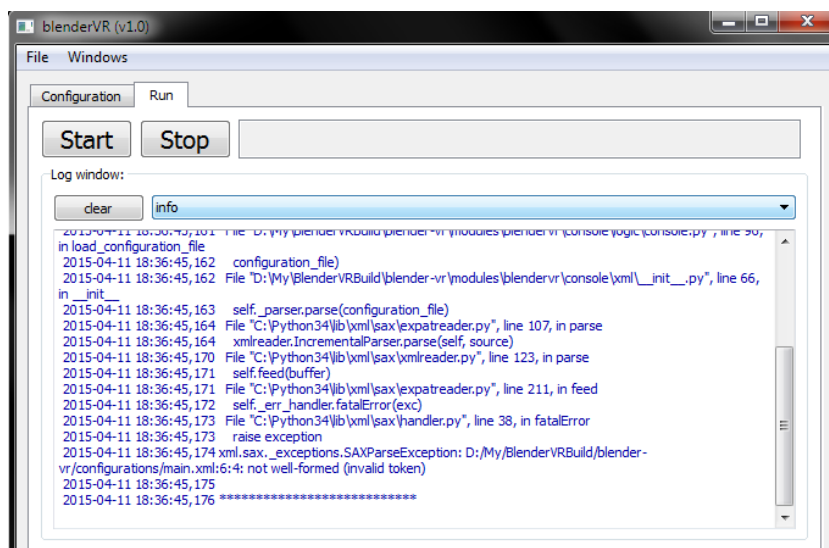
Obrázek A.2: Otevření výpisu dané obrazovky

Takto (viz Obrázek A.3) například vypadá výpis v příkazové řádce v případě, že zapomeneme přidat do konfiguračního souboru informace o cestě k Windows. Je vidět, že chyba sama o sobě příliš neřekne, proto je třeba dávat na tyto detaily pozor.



Obrázek A.3: Chybový výpis z příkazové řádky.

V případě, že uděláme syntaktickou chybu při psaní konfiguračního souboru, vypadá chybový výpis v hlavním okně tímto způsobem (Obrázek A.4):



Obrázek A.4: Chybový výpis při nevyhovujícím konfiguračním souboru.

Když sjedeme ve výpisu až dolů, vidíme, že nás aplikace upozorňuje, že v souboru `main.xml` na 6. řádce je chyba (`not well formed (invalid token)`). Zde již je výpis přehlednější a tak si s touto chybou může poradit i méně zkušený uživatel. V tomto případě se jednalo o vynechání koncové závorky u sekce `starter` (`<starter>`).

Je třeba brát v potaz, že se celá aplikace stále nachází ve fázi vývoje, a tak se stále můžeme setkat s různými chybami. Občas stane, že aplikace bezdůvodně přestane pracovat, konzoli zaplní chybový výpis, který se za stejné konfigurace dříve neobjevil atd. Pokud si jste jisti, že se nejedná

o chybu z Vaší strany, nýbrž se strany aplikace, je možné tuto chybu nahlásit týmu BlenderVR a pomoci tak s vývojem.

A.2 VRPN Server

VRPN server se nachází na místě, kam byl nakopírován, standardně v adresáři s BlenderVR v `plugins/vrpn`. Spouští se souborem `vrpn_server.exe`, který se nachází ve složce `server`. Program doporučuji spouštět v příkazové řádce s parametrem `-v`, díky čemuž získáme dodatečné informace, které mohou pomoci odhalit případné chyby. Před spuštěním samozřejmě musí být správně nakonfigurován soubor `vrpn.conf`, který se nachází na témže místě. Pro začátek doporučuji otestovat např. klávesnici, tedy odkomentovat řádek začínající na `vrpn.Keyboard` a ostatní zakomentovat.

Po spuštění doporučeným způsobem by měl server vypsat tyto řádky:

```
Reading from config file vrpn.cfg
Opening vrpn.Keyboard Keyboard0
```

Nyní je třeba otestovat klientskou část. K tomu nám poslouží program `vrpn_print_devices.exe`, který se nachází ve složce `vrpn/client`. Tato aplikace vypisuje všechna data získaná z poslouchaného zařízení. Je třeba ji spustit v příkazové řádce a jako parametr uvést poslouchané zařízení se jménem nebo adresou hostitele serveru. Vzhledem k tomu, že server běží na témže počítači, bude hostitel `localhost`. Spuštění tedy vypadá takto:

```
vrpn_print_devices.exe Keyboard0@localhost
```

Pokud vše funguje správně, stisky a následné uvolnění tlačítek klávesnice se budou vypisovat v terminálu.

Poslední test funkčnosti VRPN a zároveň Pythonu, je otestování knihovny `vrpn.pyd`. Nejprve je třeba upravit skript `vrpn_pyd_test.py`, který se na DVD nachází v `installation/vrpn`. Na čtvrtém řádku skriptu, který začíná `sys.path.append`, je nutné změnit cestu odpovídající lokaci `vrpn`. Poté stačí skript spustit dvojklikem nebo v příkazové řádce příkazem:

```
python vrpn_pyd_test.py
```

Pokud posledním řádkem výpisu bude: `***Success***`, vše je v pořádku.

V opačném případě doporučuji zkontrolovat cestu uvedenou ve skriptu, zda je správně nainstalován Python, případně znovu zkopírovat nebo zkompileovat VPRN.

A.3 Oculus Rift

V případě, že byl Oculus Rift nainstalován a nastaven dle návodu, první věcí, kterou je třeba učinit, je přepnutí režimu na *Extended Display*. To se provede kliknutím na ikonu Riftu na panelu Windows, kde zvolíme možnost **Rift Display Mode**. Nyní by se měla otevřít tabulka umožňující nastavení zobrazení, kde je třeba označit možnost **Extend Desktop to the HMD**. Alternativně je možné spustit **Oculus Configuration Utility** a zde v záložce **Tools** opět zvolit možnost **Rift Display Mode**. Pokud tato nastavení nevidíte, doporučuji zkontrolovat, zda používáte správnou verzi software, tedy Runtime verze 0.6 nebo 0.5 (starší bych již nedoporučoval). Pokud je Oculus zapnutý, měla by po změně nastavení svítit modrá kontrolka.

Nyní je třeba přepnout typ zobrazení druhé obrazovky ve Windows, což se dělá v místě, kde se nastavuje rozlišení obrazovky. Oculus by zde měl být zobrazen jako svislá podlouhlá obrazovka, což je špatně, protože výsledná projekce bude překlopená. Abychom tento problém vyřešili, stačí změnit orientaci na výšku. Takto je Oculus připraven k použití v BlenderVR.

Bohužel BlenderVR ve Windows neumožňuje určit obrazovku, na které se má aplikace spustit. To značně komplikuje situaci, zejména pak při použití zobrazení na celou obrazovku. Jsou ale způsoby, jak si s tímto problémem poradit. Prvním řešením je spustit aplikaci standardně na celou obrazovku a stiskem kláves *Shift + Windows + šipka doleva* nebo *doprava* přesunout okno na druhou obrazovku, tedy na Oculus. Toto řešení je nejlepší, nicméně reaguje špatně v případě, že má hlavní obrazovka jiný poměr stran než Oculus Rift, což je 16:9. Další možností je spustit aplikaci v režimu v okně. Toto okno pak přesunout na obrazovku Riftu a poté jej maximalizovat. Zde může nastat problém, kdy jsou vidět okraje okna, což ruší celkový dojem z dané aplikace virtuální reality. Posledním, mě známým řešením, jak tento problém vyřešit, je nastavit Oculus Rift jako hlavní obrazovku. Poté stačí spustit aplikaci v zobrazení na celou obrazovku a ona se automaticky zobrazí v Oculus Riftu. Tento způsob mi fungoval vždy, nicméně není příliš pohodlný a často je třeba nešikovně přesouvat okna či klikat v projekci.

Pro otestování funkčnosti Oculus Riftu ve spojitosti s BlenderVR doporučuji zkusit ukázkou `oculusDemo`, která je pro tyto účely určena. Detaily o této ukázce jsou dostupné v příloze s ukázkami.

B Popis ukázek

Všechny ukázky se spouštějí standardním způsobem, který byl vysvětlen výše. Každá z ukázek se skládá ze souboru `*.blend`, `*.processor.py` a konfiguračního souboru `*.xml`. Tyto soubory sdílejí název se jménem adresáře, kde se daná ukázka nachází. Aplikace, které využívají VRPN také obsahují soubor `vrpn.cfg` upravený pro danou ukázku, kterým je třeba nahradit `vrpn.cfg` na místě se serverem. Nicméně uživatel také může snadno změnit potřebné řádky sám. Jednotlivé ukázky jsou na DVD umístěny ve složce `demos`, kde má každá ukázka svůj adresář.

Některé ukázky také využívají skripty napsané v jazyce Python, které jsou integrovány uvnitř souboru `*.blend`. Tyto skripty jsou dostupné při otevření daného souboru v Blenderu v pohledu *Scripting* nebo *Game Logic*. Jejich kopie jsou také umístěny na DVD v adresáři `source`.

B.1 Jednoduchá ukázka

Adresář: `demos/basic`

Tato ukázka slouží k otestování funkčnosti BlenderVR ve své nejjednodušší podobě. Po spuštění se zobrazí scéna s kuličkou, kde je pomocí šipek možné s kuličkou hýbat. Tato ukázka nevyužívá VRPN, ovládání je implementováno pomocí BGE. Nachází se zde pouze jedno zobrazení zvané `basic`, které spustí aplikaci v okně velikosti 640 x 480px.

B.2 Ukázka s myší a klávesnicí

Adresář: `demos/kmDemo`

Cílem této ukázky je demonstrovat spolupráci VRPN a BlenderVR. Proto je zde ovládání implementováno v souboru `*.processor.py`, i když by bylo možné, a také jednodušší, implementovat tyto funkce přímo v BGE.

Opět se jedná o jednoduchou scénu, kde se nachází pouze kulička na

plošině. S touto kuličkou je možné posouvat po scéně pohybem myši. Tlačítka myši slouží k obsluze kamery ve scéně. levým tlačítkem se kamera přiblíží, pravým se naopak oddálí. Prostředním tlačítkem myši je možné resetovat kameru do původní pozice.

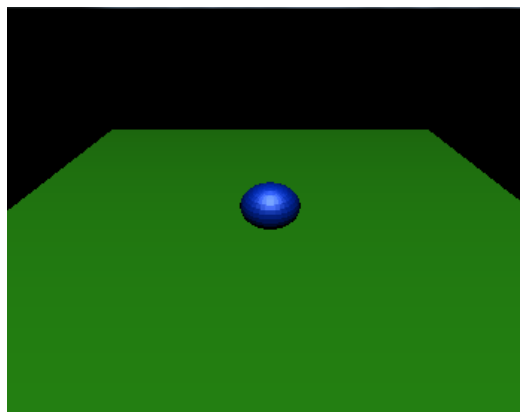
Pomocí klávesnice můžeme s kuličkou vyskočit, konkrétně šipkou nahoru. Při podržení šipky dolů, kulička dočasně zmizí, po jejím uvolnění se opět objeví. Šipky doleva a doprava vypisují text o jejich stisknutí do terminálu. Tlačítkem Enter je možné resetovat kuličku na původní pozici.

Ukázka obsahuje několik zobrazení. Stejně jako ta předchozí, i tato obsahuje zobrazení `basic`. Dále se zde nachází `dua1`, které spustí dvě okna vedle sebe, každé o velikosti 320 x 480px. Posledním zobrazením je `fullscreen`, které aplikaci spustí přes celou obrazovku. Ukázka je vidět na obrázku B.1.

Před spuštěním aplikace je třeba zapnout VRPN server s konfigurací pro myš a klávesnici, soubor `vrpn.cfg` tedy musí mít odkomentovány tyto řádky:

```
vrpn_Mouse Mouse0  
vrpn_Keyboard Keyboard0
```

Nebo je možné použít `vrpn.cfg`, který se nachází v adresáři s ukázkou.



Obrázek B.1: Ukázka s klávesnicí a myší

B.3 Ukázka s WiiMote

Adresář: `demos/wiiDemo`

Tato ukázka slouží k demonstraci a vyzkoušení funkčnosti Wii ovladače. Bohužel se mi nepodařilo zkompileovat VRPN server takovým způsobem, který by umožnil zaznamenávat analogický pohyb zařízení, proto je v ukázce možné pracovat pouze s tlačítky. V případě, že by ale analogový pohyb byl zaznamenán, vypíše se do terminálu text: "**Analaog movement recorded**" a vypíše se konkrétní hodnoty.

Ve scéně se opět nachází kulička, která umožňuje stejné funkce jako v předchozí ukázce. Pomocí šipek na ovladači je možné s kuličkou pohybovat. Průhledný knoflík s nápisem 'A' umožňuje resetovat pozici kuličky. Pomocí tlačítek plus a minus je možné přibližovat, respektive oddalovat kameru. Tlačítko s domečkem 'HOME' vrátí kameru do výchozí pozice. Tlačítko s číslem '1' dočasně skryje kuličku, zatímco tlačítko '2' vypíše do terminálu zprávu. Poslední tlačítko v přední části vespod ovladače, označené jako 'B', umožňuje skoky kuličky.

Opět je třeba spustit VRPN server. Konfigurační soubor `vrpn.cfg` musí mít odkomentován tento řádek:

```
vrpn.WiiMote WiiMote0 1 0 0 1
```

K dispozici jsou stejná zobrazení jako v ukázce s myší a klávesnicí, tedy `basic`, `dual` a `fullscreen`.

Skript, který umožňuje plynulý pohyb kuličky pomocí VRPN se jmenuje `Movement.py`. Jedná se o stejný princip, který je použit i v demonstrační hře (více informací dále).

B.4 Ukázka s Oculus Rift DK2

Adresář: `demos/oculusDemo`

Stejně jako předchozí příklad sloužil k demonstraci WiiMote, tento slouží k vyzkoušení zařízení Oculus Rift DK2. Jedná se o jednoduchou ukázkou, kde kolem uživatele rotuje řada barevných koulí. Uživatel nemůže prostředí nijak měnit, může se pouze rozhlížet s použitím Oculus Riftu.

Opět je zde zobrazení `basic`, které zobrazí scénu bez použití Riftu. Dále jsou zde dvě zobrazení určené pro Oculus, jedná se o `oculus window`, které spustí aplikaci v okně s rozlišením 960 x 540px a poté `oculus full`, které

aplikaci spustí v režimu na celou obrazovku.

Skript použitý pro rotaci objektů je pojmenován `Rotation.py`.

B.5 Demonstrační hra

Adresář: `demos/game`

B.5.1 Obecně

Jedná se o nenáročnou hru, která obsahuje prvky použité v jednotlivých ukázkách. Hra je prezentována v první osobě, tedy z pohledu hráče. Ten se zde ocitá v roli vědce, který skončil na kolečkovém křesle a nyní prozkoumává opuštěné laboratorní zařízení. Invalidní vozík jsem zvolil z důvodů použití Oculus Riftu, protože hráč sedí na židli a mačká tlačítka, a tak se dle mého názoru se hrou, kde se nachází ve stejné pozici, lépe ztotožní, což zlepší dojem celé virtuální reality.

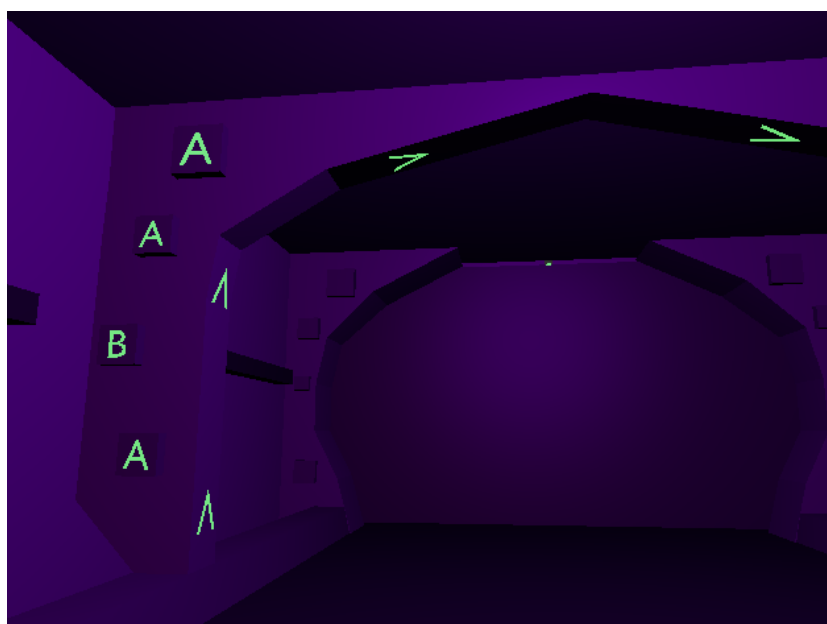
Hra se skládá z několika místností a hráčovým úkolem je přijít na to, jak otevřít dveře v každé z nich a dostat se tak dál. K otevření dveří je třeba vyřešit jednoduchý úkol, který obvykle vyžaduje dobře sledovat své okolí. Ukázky ze hry jsou vidět na obrázcích B.2 a B.3.

B.5.2 Zobrazení

K dispozici jsou zobrazení použitá v předchozích ukázkách. Je zde tedy `basic`, pro testovací účely a `fullscreen`, které spustí hru na celou obrazovku. Hra podporuje Oculus Rift, proto tu jsou i zobrazení `oculus window`, `oculus full`. V případě použití Oculus Riftu je vhodné se správně usadit, aby se hráč cítil co nejpřirozeněji vzhledem k zobrazení křesla ve hře.



Obrázek B.2: Ukázka ze hry z 2. místnosti



Obrázek B.3: Ukázka ze hry z 3. místnosti

B.5.3 Ovládání

Hru lze ovládat několika způsoby. Uživatel se může rozhlížet pomocí Oculus Riftu, případně je možné použít myš. Vzhledem k tomu, že myš kromě rozhlížení nemá implementovanou žádnou jinou funkci, nedoporučuji používat tato zařízení současně. Rozhlížení pomocí myši je po spuštění aplikace vypnuto a v případě potřeby je třeba jej aktivovat (viz dále). Pro pohyb a akce je třeba použít klávesnici nebo WiiMote. S klávesnicí se hráč pohybuje pomocí šipek nebo kláves W, S, A, D. Poté je zde akce A, kterou lze vyvolat klávesou CTRL a akce B, kterou zastupuje mezerník. Rozhlížení myší je možné aktivovat klávesou ENTER. WiiMote má ovládání obdobné. Hráč se pohybuje s použitím šipek, akci A vyvolá tlačítkem 'A' (větší kulaté tlačítko) a akci B tlačítkem 'B' (tlačítko vespod ovladače), myš se aktivuje tlačítkem '2'. Role daných akcí nejsou specificky určeny, záleží na konkrétní místnosti. Ve chvíli, kdy je aktivováno rozhlížení pomocí myši, přepne se způsob pohybu hráče. Místo, aby se hráč pomocí levé a pravé šipky otáčel, dělá nyní úkroky, jako obvyklé u většiny her z první osoby. Po vypnutí myši se ovládání vrátí do původního stavu.

Aplikace se spouští standardním způsobem a je třeba také spustit VRPN server, který poslouchá WiiMote a klávesnici, tedy s těmito odkomentovanými řádky:

```
vrpn_Keyboard Keyboard0
vrpn_WiiMote WiiMote0 1 0 0 1
```

V případě, že nepoužijete WiiMote a zařízení bude vypnuté, je možné jej vyjmout z konfiguračního souboru VRPN. V opačném případě bude server periodicky vypisovat chybu při snaze spojit se s WiiMote. I přesto, že by tato skutečnost nijak neměla ovlivnit funkčnost programu, je dobré o ní vědět. V případě, že je aktivováno rozhlížení myší, ukazatel se zafixuje v okně aplikace, se kterým se poté hůře pohybuje. To způsobuje problémy zejména v případě použití Oculus Riftu. Z tohoto důvodu je tato funkce z počátku vypnuta, uživatel si ji může snadno zapnout, jakmile je spokojen s nastavením okna aplikace.

B.5.4 Implementace

Klávesnice a WiiMote využívají komunikaci pomocí VRPN. V souboru `game.processor.py` je zachyceno stisknutí používaných kláves. Stisk dané klávesy přepne logickou proměnnou, která je uchována jako parametr hráče. Funkce dané klávesy je pak zpracována v BGE skriptem `Enviroment.py`, který zajišťuje interakci se scénou. Oculus Rift je také zpracován v souboru `game.processor.py`, je zde obslužen interní funkcí BlenderVR `user_position()`.

Myš je zajištěna skriptem `MouseLook.py` v BGE, který není mé tvorby, ale byl získán ze stránek *Tutorials for Blender 3D*[27]. Implementace rozhlížení myši není triviální záležitost a vzhledem k tomu, že je hra zaměřena na Oculus Rift, byla přidána pouze jako druhá možnost. Z těchto důvodů jsem zvolil využití cizí implementace.

B.5.5 Závěr

Jedná se o ukázkovou hru, a proto je zde velký prostor pro zlepšení. Hře by prospělo např. lepší grafické zpracování nebo rozšíření obsahu. Velkým plusem by také byly prostorové zvukové efekty. Značně se zde projevuje chyba BlenderVR, která způsobuje zkreslení při použití Oculus Riftu. Tato chyba negativně ovlivňuje zejména text a drobné detaily, které uživatel vidí rozmazaně. S další verzí BlenderVR by již tento problém měl být opraven a dojem z celé hry by se tak měl zlepšit.

WiiMote zde také působí poněkud nešikovně. To je z důvodu, že tento ovladač je primárně určen na práci s analogovými částmi, která je zde bohužel nefunkční. Z toho důvodu by se zde více hodil klasický gamepad, se kterým by ovládání hry tohoto typu bylo přirozenější.

C Obsah DVD

Součástí práce je přiložené DVD, které obsahuje:

- Tento dokument ve formátu PDF i jeho zdroje v adresáři `document`
- Ukázkové aplikace v adresáři `demos`
- Instalační soubory v adresáři `installation`
- Skripty použité v souborech `*.blend` v adresáři `source`