



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

VIZUALIZACE MODELU VE VIRTUÁLNÍ REALITĚ

MODEL VISUALIZATION IN VIRTUAL REALITY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Ondřej Cmajdálka

VEDOUcí PRÁCE

SUPERVISOR

Ing. Tomáš Hůlka

BRNO 2019

Zadání bakalářské práce

Ústav: Ústav automatizace a informatiky
Student: **Ondřej Cmajdálka**
Studijní program: Strojírenství
Studijní obor: Základy strojního inženýrství
Vedoucí práce: **Ing. Tomáš Hůlka**
Akademický rok: 2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Vizualizace modelu ve virtuální realitě

Stručná charakteristika problematiky úkolu:

Možností, jak zobrazit model ve virtuální realitě, je mnoho. Úkolem studenta bude nalézt tu nejvhodnější pro zobrazení náročnějších modelů a s její pomocí vytvořit vizualizaci vhodného objektu (např. robota).

Cíle bakalářské práce:

Stručná rešerše problematiky.
Vytvoření VR modelu dle předlohy.

Seznam doporučené literatury:

WANG, S., MAO, Z., ZENG, C., GONG, H., LI, S. and CHEN, B.: A new method of virtual reality based on Unity3D, 2010 18th International Conference on Geoinformatics, Beijing, 2010, pp. 1-5.

TRENHOLME, D., SMITH, S.P.: Computer game engines for developing first-person virtual environments, Virtual Reality (2008) 12: 181.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Tato práce se zabývá fenoménem zvaným virtuální realita, jeho historickým vývojem a v dnešní době používanými technologiemi. Dále se věnuje herním enginům, jejich historii a struktuře. Praktická část práce popisuje návrh a tvorbu prostředí pro virtuální realitu. Navržené prostředí demonstruje vybrané vlastnosti herního enginu, které z něj činí vhodný nástroj k vývoji prostředí pro virtuální realitu.

ABSTRACT

This thesis deals with phenomenon called virtual reality, its historical development and nowadays used technologies. The next part is dedicated to game engines, their history and structure. In practical part a proposal and development of virtual reality environment is described. The proposed environment demonstrates selected features of the game engine that make it a suitable tool for developing virtual reality environments.

KLÍČOVÁ SLOVA

Virtuální realita, Snímače polohy, Stereoskopické displeje, HTC Vive, Herní engine, Unity, 3D modely

KEYWORDS

Virtual reality, Tracking devices, Stereoscopic display, HTC Vive, Game engine, Unity, 3D models

BIBLIOGRAFICKÁ CITACE

CMAJDÁLKA, Ondřej. Vizualizace modelu ve virtuální realitě [online]. Brno, 2019 [cit. 2019-05-23]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/117467>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Tomáš Hůlka.

PODĚKOVÁNÍ

Rád bych zde poděkoval zejména panu Ing. Tomáši Hůlkovi za přátelské a vlídné vedení při tvorbě práce a za poskytnutí cenných rad. Dále bych chtěl poděkovat své rodině a přátelům za podporu.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Tomáše Hůlky a s použitím literatury uvedené v seznamu literatury.

V Brně dne 24. 5. 2019

.....

Ondřej Cmajdálka

OBSAH

ÚVOD.....	15
1 VIRTUÁLNÍ REALITA	17
1.1 Definice virtuální reality.....	17
1.2 Historie VR.....	18
1.3 Aplikace.....	20
1.4 Technologie	21
1.4.1 Vstupní zařízení.....	22
1.4.2 Výstupní zařízení.....	24
2 HERNÍ ENGINEY	29
2.1 Definice	29
2.2 Historie	29
2.3 Komponenty	30
2.4 Přehled současných enginů.....	31
2.4.1 Unity.....	32
2.4.2 Unreal Engine 4	33
2.4.3 CryEngine	34
3 PRAKTICKÁ ČÁST	35
3.1 Uživatelské rozhraní v Unity.....	35
3.2 Podpora VR v Unity	36
3.3 Návrh scény	38
3.3.1 Modelování.....	38
3.3.2 Pohyb	39
3.3.3 Kompozice.....	39
3.4 Samotná tvorba	40
3.4.1 Basketbal	41
3.4.2 Bowling	43
3.4.3 Galerie	47
3.4.4 Střelnice	49
3.4.5 Spojovací místnost.....	53
ZÁVĚR	57
SEZNAM POUŽITÉ LITERATURY.....	59
SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ	63

ÚVOD

Snad každý někdy snil o tom, že bude špičkový závodník, pilot, nebo třeba kosmonaut. Jenom malé části z nás se tyto sny skutečně vyplní. Ostatním nezbyvá než se spokojit s různými simulacemi. Simulace se více či méně blíží v poskytovaných podnětech našim smyslům těm skutečným, a tak navozují alespoň iluzi rychlé jízdy, letu letadlem nebo obletu měsíce. Nejdokonalejší simulaci reality představuje virtuální realita, která má za cíl poskytovat vjemy tak věrné, že si uživatel bude připadat, jako by se na chvíli opravdu stal třeba oním pilotem.

Odvětví virtuální reality se neustále vyvíjí a stává se čím dál tím dostupnějším pro širokou veřejnost. Jakožto nástroj, který nachází své uplatnění i v jiných oblastech, než je zábavní průmysl, si jistě zaslouží naši pozornost.

V první kapitole se virtuální realitě bude podrobně věnováno. Bude zmíněna její historie, možné uplatnění a technologie, které ke svému fungování využívá. Podrobněji bude nahlédnuto na způsob, jakým lze zaznamenávat pohyb a taky, jak lze promítanému obrazu dodat potřebnou hloubku.

V druhé kapitole bude zblízka pohlédnuto na herní enginy, které představují velice mocný nástroj v oblasti vytváření virtuálních prostředí. Jejich fyzikální enginy dovedou věrně simulovat působení sil a docílit tak realisticky vypadajících pohybů. Jejich pokročilé vykreslovací systémy dokážou tak věrně podat strukturu a odlesky povrchu vložených 3D modelů, že má člověk až potřebu si na ně sáhnout. Proto jim bude věnována celá druhá kapitola této práce. Na konci kapitoly budou zmíněny jedny z nejznámějších enginů a budou krátce porovnány.

V praktické části práce bude popsán vývoj prostředí pro virtuální realitu. Scéna bude navržena tak, aby demonstrovala vlastnosti vývojového prostředí. Vybraný herní engine bude podrobněji popsán včetně uživatelského rozhraní a kroků, které je třeba udělat, aby byl výsledný projekt spustitelný ve virtuální realitě. Vývoj prostředí bude podrobně popsán od samotného záměru demonstrace určité vlastnosti, přes návrh, až po celkové vymodelování a aplikaci funkčních interakcí.

Dosažené výsledky budou shrnuty v závěrečné části práce.

1 VIRTUÁLNÍ REALITA

Pojem *virtuální realita* je široce zpopularizován. Často ale dochází k jeho mylnému výkladu. Proto je v této kapitole rozebráno, co přesně pojem VR znamená. Dále pak je zde uveden historický vývoj, kterým si tohle odvětví prošlo. Na závěr kapitoly jsou pak zmíněny některé přístroje mající funkci vtáhnout uživatele do virtuálního prostředí.

1.1 Definice virtuální reality

Jednou z možných definic pojmu VR je interaktivní, pohlcující, divácky zaměřené, multismyslové, trojrozměrné, počítačem generované prostředí a technologie, které tohle prostředí vytváří [1]. Tahle definice, stejně jako mnoho dalších používaných definic pojmu VR, není jednoznačná. Např. technologií sloužících k vytvoření VR a vtažení uživatele do tohoto prostředí existuje velké množství.

Jedná se tedy o umělé, počítačem vytvořené prostředí. Uživatel s ním interaguje pomocí zařízení, které poskytují jeho smyslům zpětnou vazbu. Uživatel má tak pocit, že se nachází uvnitř uměle vytvořeného světa, který jej zcela pohlcuje a který může plně ovlivňovat svým působením.

V souvislosti s pojmem VR je potřeba zmínit pojmy *telepresence* a *kyberprostor*. Ty jsou blízce spojené s VR, avšak mají trochu jiný význam [2].

- Telepresence – pojem zaveden Marvinem Minsky (1980) odkazující na teleoperační systémy pro dálkové ovládání manipulace s fyzickými položkami. Je to druh VR, který ale místo virtuálního prostředí přenáší uživatele do reálného vzdáleného prostředí.
- Kyberprostor – byl vynalezen a definován Williamem Gibson jako: „Konsenzuální halucinace prožívaná denně miliardami legitimních operátorů, v každém národě, dětmi, které se učí matematickým pojmům... grafické zobrazení dat abstrahovaných z pamětí každého počítače v lidské společnosti. Nepředstavitelná komplexita. Linie světla rozprostírající se v neprostoru mysli, klastry a konstelace dat. Jako světla velkoměsta, vzdalující se..." [3].

Dnes je pojmem označován *World Wide Web (Internet)*.

Od VR je potřeba také oddělit *rozšířenou realitu* (AR podle anglického výrazu *augmented reality*). Ta využívá podobně jako VR náhlavních displejů, jedná se ale o displeje průhledové. Kombinuje tedy reálný svět s počítačem generovanými grafickými prvky. Jako příklad je možné uvést Google ARCore pro android nebo Microsoft HoloLens (obr. 1) [4].



Obr. 1: Microsoft HoLolens 2 [5]

1.2 Historie VR

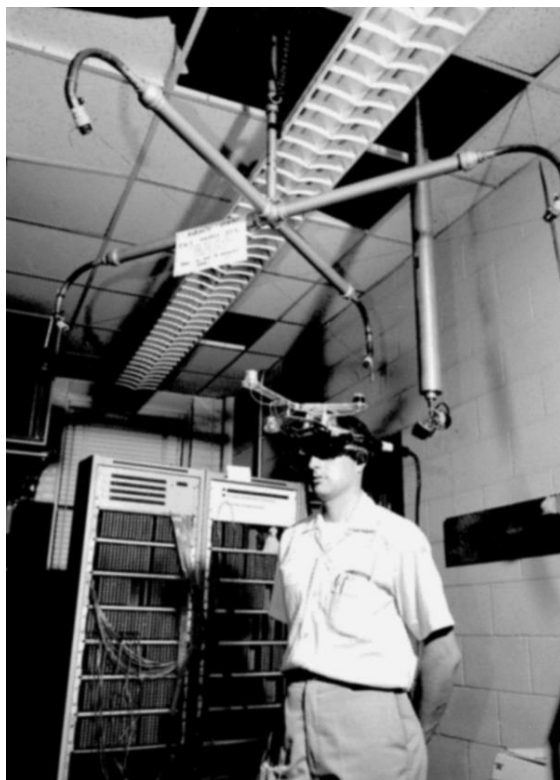
Prvním přístrojem pro VR byla *Sensorama* navržena Mortonem Heilig v 50. letech (obr. 2). Prototyp byl sestaven v roce 1962. Přístroj využíval 3D vizuální, zvukové, haptické a čichové podměty a dokonce i vítr, aby zajistil dokonale pohlcující zážitek. Promítal film znázorňující jízdu na motocyklu Brooklynem. Nebyla zde však žádná možnost interakce s filmem, jako např. změnit trasu jízdy [6].



Obr. 2: Sensorama [7]

Nejběžněji se ve VR využívají tzv. náhlavní displeje (HMD – *Head Mounted Display*). První HMD byl vytvořen společností Philco v roce 1961 a jmenoval se *Headsight*. Byla to helma s CRT (Cathode Ray Tube) obrazovkou a pohyb sledujícím zařízením k identifikování pozice hlavy [6].

Jedním z nejznámějších HMD, nebo lépe BOOM (Binocular Omni Orientation Monitor), je „*Sword of Damocles*“ sestaven Ivanem Sutherland v roce 1968 (obr. 3). Přístroj byl příliš těžký na pohodlné nošení, a tak musel být zavěšen na strop (běžné pro BOOM). Odtud vzešel název přístroje. Souprava byla schopna sledovat polohu uživatele i jeho očí a přizpůsobit promítaný obraz těmto informacím [6].



Obr. 3: Sword of Damocles [8]

Dalším zařízením spojeným s technologií VR jsou snímací rukavice. První taková rukavice byla „*Sayre Glove*“ vytvořena v roce 1977 Tomem DeFanti a Danielem J. Sandin. Fungovala pomocí světelných senzorů. Ty byli spojeny se zdrojem světla přes ohebné trubice, které byly připojené ke všem prstům. Při ohnutí prstu došlo ke změně množství světla dopadajícího na senzor. Tuhle změnu bylo možné detekovat. Další rukavicí je „*Power Glove*“ vyráběné firmou Mattel pro Nintendo Entertainment System. Ta kromě snímání ohybu prstů snímala i polohu rukavice v prostoru pomocí ultrazvukových snímačů. Přestože byla mnohem levnější a zpočátku se prodávala poměrně dobře, tak kvůli nedostatku her a kritice na její nepřesné a obtížné ovládání se produkce zastavila [6].

V minulosti zde byly pokusy uvést další produkty související s virtuální realitou pro širokou veřejnost. Z důvodu vysoké ceny se však zatím tyto technologie převážně uplatňují v armádním a akademickém výzkumu [6].

1.3 Aplikace

VR nabízí velké výhody v mnoha aplikačních oblastech. Poskytuje snadný a intuitivní způsob interakce mezi člověkem a počítačem. Uživatelé je umožněno sledovat a manipulovat s počítačem vytvořeným prostředím stejně, jako by byl v reálném světě. Odpadá tedy nutnost učením se mnohdy obtížné interakce s uživatelským prostředím. Díky tomu VR nachází uplatnění v leteckých simulátorech, prohlídkách architektonických návrhů, vizualizace dat, teleoperačních systémech a v neposlední řadě v zábavním průmyslu [1].

Architektonické prohlídky byly první pokusy využívající k vizualizaci VR. Právě díky VR lze docílit pocitu přítomnosti a prostoru ve virtuální budově. Ten nelze zprostředkovat ani pomocí nejrealističtějších obrazů nebo animací promítaných na monitor [1].

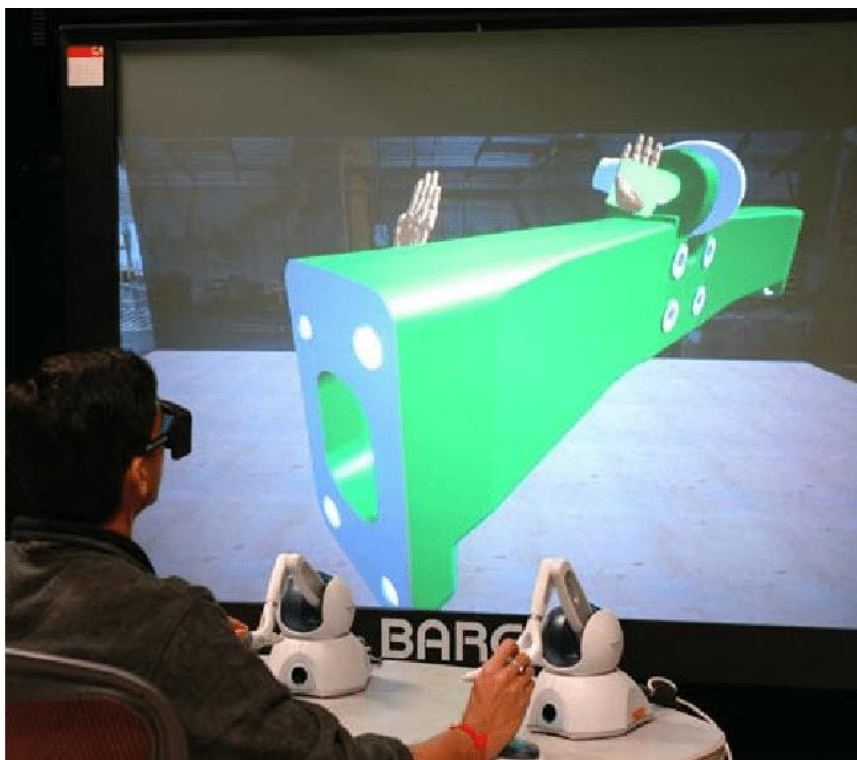
VR je velmi výhodně využívána v letectví při výcviku pilotů, a to jak vojenských, tak i civilních. Výcvik na leteckém simulátoru (obr. 4) je daleko levnější než na skutečném letadle. Navíc je nesrovnatelně více bezpečný. Nabízí také možnost simulace nebezpečných situací, které např. nastaly v minulosti a mnohdy skončily katastrofálně. Ty by z pochopitelných důvodů nebylo možné vyzkoušet se skutečným letadlem [9].



Obr. 4: Letecký plně pohyblivý simulátor [10]

Další oblastí, kde se začíná VR používat, je simulování montážních metod. Návrh montážního postupu je jednou z nejnákladnějších částí vývoje nového produktu. Dobře navržený montážní proces musí brát v úvahu mnoho faktorů jako např. optimální doba nutná k sestavení, v jakém pořadí se díly budou sestavovat, jaké nástroje budou potřeba k sestavení atd. Nemělo by být zapomenuto na ergonomii, bezpečnost pracovníků a přístupnost jednotlivých komponent. V současné době návrháři využívají 3D CAD systémy. Výsledné sestavení je však zobrazené na 2D monitoru počítače. Následuje

zhotovení fyzického prototypu, který je poté sestaven pracovníky. Až oni zjistí, zda je možné produkt skutečně sestavit a jak dobře jsou jednotlivé části přístupné. Právě zhotovení fyzického modelu je ale velmi nákladné a zdlouhavé. Proto v současnosti probíhá snaha o vytvoření VR aplikací, ve kterých by bylo možné sestavit výsledný produkt bez nutnosti výroby fyzického prototypu. Nejnovější vývoj se pak do takové aplikace snaží zahrnout i co nejrealističtější hmatovou a zvukovou zpětnou vazbu. Vývoj takové aplikace čelí několika problémům jako jsou optimalizace detekce kolizí a simulace silového působení mezi sestavovanými částmi [11].



Obr. 5: Obouruční haptické rozhraní SHARP [11]

1.4 Technologie

Pod pojmem VR jsou v očích široké veřejnosti rozuměny převážně náhlavní displeje (HMD). To ale zdaleka nejsou všechny technologie, které do toho odvětví spadají. Nejedná se dokonce ani o jedinou používanou zobrazovací techniku. VR zahrnuje velké množství zobrazovacích, pohyb snímajících a softwarových technologií. V následující podkapitole je pohlédnuto na některé z nich.

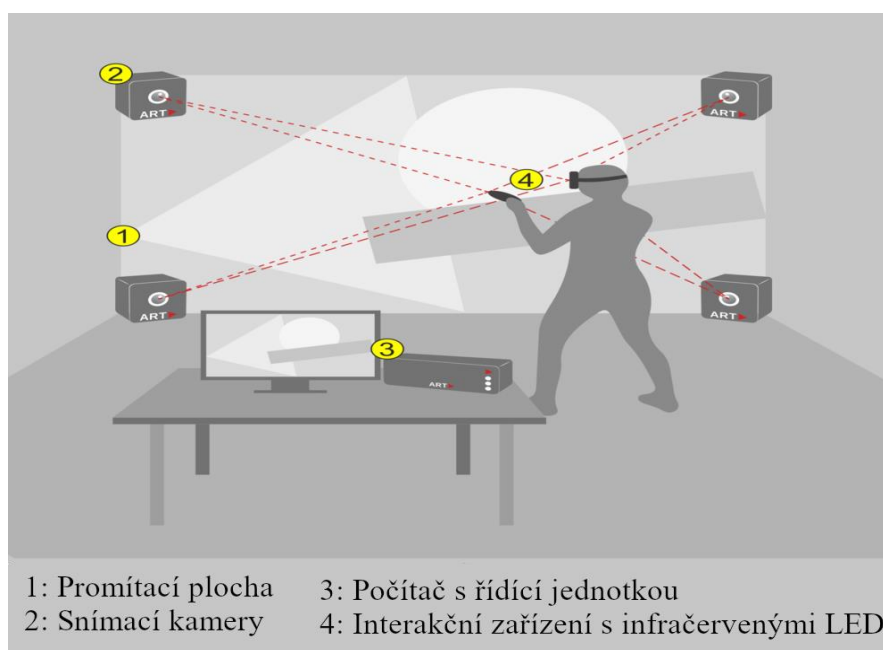
1.4.1 Vstupní zařízení

Pomocí vstupních zařízení je zprostředkován vliv uživatele na virtuální prostředí. Pohyb a vykonávání úkonů ve virtuálním světě by měly být co možná nejpřirozenější a nejintuitivnější. Stále není k dispozici technologie, která by to plně umožňovala [1].

Úplným základem pro pohlcující VR systémy je snímání pozice a orientace hlavy, které jsou nutné pro správné vykreslování. Dále pak jsou obvykle snímány pozice rukou, popřípadě i dalších částí těla. Jelikož žijeme v trojrozměrném prostoru, mají všechny tyto objekty šest stupňů volnosti (DOF – *degrees of freedom*). Jedná se o posunutí podél tří souřadnicových os a rotací okolo těchto os. Zařízení snímající tyto hodnoty se nazývají *trackery* (obr. 6) [1].

Nejdůležitějšími vlastnostmi zvažovanými při výběru vhodných trackerů pro jednotlivé aplikace podle [1] jsou:

- *obnovovací frekvence* – je definovaná jako počet měření poloh na jednotlivých souřadnicích za jednu sekundu. S rostoucí frekvencí roste plynulost snímání pohybu, ale rostou nároky na výpočetní výkon.
- *odezva* – je uváděna jako čas obvykle měřený v milisekundách mezi akcí uživatele v reálném světě a odesláním informace o této akci trackerem. Snaha je samozřejmě docílit co možná nejmenší hodnoty.
- *přesnost* – je představována chybou v naměřených hodnotách jednotlivých souřadnic. Nejčastěji se udává v případě souřadnic v milimetrech a u rotací v úhlových stupních. Opět menší hodnota znamená kvalitnější tracker.
- *rozlišení* – nejmenší změna polohy a rotace, kterou je tracker schopen detekovat. I zde je cílem dosáhnout co nejmenších hodnot.
- *rozsah* – je definován jako pracovní oblast, ve které je tracker schopný zaznamenávat pohyby se specifikovanou přesností a rozlišením.



Obr. 6: Příklad optického trackovacího systému společnosti ART. Upraveno z [12]

Dalšími nezanedbatelnými vlastnostmi ovlivňujícími použití trackerů jsou jejich hmotnost, velikost, komfort při používání atd. Bylo vyvinuto mnoho způsobů detekce pohybu, které jsou v [1] popsány následovně:

- Magnetické trackery – patří k nejpoužívanějším. Celé zařízení se skládá ze statické části (*vysílač*, taky nazývaný zdroj), pohyblivých částí (*přijímač*, nebo taky senzor) a kontrolní stanice. Obě části mají stejnou konstrukci. Obsahují tři na sebe vzájemně kolmé antény. Ty mění elektrický proud na magnetické pole a opačně. Výsledný signál je přiveden do kontrolní stanice, která na jeho základě spočítá polohu a orientaci senzorů.

Je možné použít jak střídavý, tak i stejnosměrný proud. Řešení pomocí stejnosměrného proudu se používá za účelem eliminovat vznikání vířivých proudů, které způsobují rušení. I tak jsou takové trackery ale velmi náchylné k rušení způsobenému feromagnetickými materiály.

Mezi výhody magnetických trackerů patří malá velikost a nízká hmotnost senzorů. Dále u nich není vyžadováno, aby mezi vysílačem a přijímačem byla nepřerušená linie pohledu (LOS – *line of sight*), nejsou náchylné k akustickému rušení, vysoká obnovovací frekvence, nízká odezva a dobrá dostupnost.

Nevýhody pak jsou silná závislost kvality měření polohy na vzdálenosti přijímače od emitoru. Je to důsledek chování magnetické vlny, kdy její amplituda klesá s rostoucí vzdáleností od zdroje. A velké množství zdrojů rušení.

- Akustické trackery – využívají ultrazvukové vlny (frekvence nad 20 kHz). Vysílač i přijímač se typicky skládají ze tří čidel s pevnou prostorovou konfigurací. Měření je čas, který zabere zvukové vlně dorazit k přijímači, nebo koherence vlny. Metoda poskytuje relativní data. To znamená, že informuje pouze o velikosti změny měřené hodnoty. Proto chyby v měření narůstají s časem. Parametry nemohou konkurovat magnetickým trackerům.

Mezi výhody patří levná pořizovací cena, nízká hmotnost a malá velikost. Oproti magnetickým trackerům netrpí rušením magnetickou interferencí.

Nevýhody jsou nutnost zajištění přímé linie pohledu mezi senzory, nepřesnosti v měření způsobené akustickou interferencí (akustický šum a ozvěny).

- Optické trackery – existuje jich několik druhů. K měření prostorové konfigurace jim slouží kamery, které zachycují světelnou mapu ze série světelných zdrojů (např. LED). Jiný typ využívá laserového světla. To nejprve projde difrakční mřížkou a je promítnuto na snímaný objekt. Senzor pak vyhodnotí vzniklý obrazec a vypočítá orientaci v prostoru. Poslední možností jsou pak systémy pracující se vzory. Porovnávají snímané vzory s předem nadefinovanými a podle těchto porovnání určují natočení a polohu snímaného objektu. Stejně jako u magnetických trackerů i zde s rostoucí vzdáleností mezi senzorem a snímaným objektem klesá velmi výrazně přesnost měření.

Výhodou je velmi rychlá obnovovací frekvence, možné pokrytí velkých pracovních objemů, nedochází k rušení, jako u předchozích metod. Dosahují i velmi dobré přesnosti.

Nevýhodou jsou nutnost zajištění přímé linie pohledu, okolní světlo a infračervené záření mohou negativně ovlivnit přesnost, vysoká pořizovací cena a problémy se sledováním více než jednoho objektu.

- Mechanické trackery – jedná se o spojení několika tuhých nosníků propojených klouby. Ze znalosti místa ukotvení počátku, délek nosníků a úhlů natočení v jednotlivých kloubech měřených např. pomocí potenciometrů lze určit polohu volného konce.

Výhodou je vysoká přesnost, nepodléhají žádnému rušení, mají vysoké obnovovací frekvence a mohou podporovat silovou zpětnou vazbu.

Nevýhody pak jsou velice omezená svoboda pohybu vlivem mechanického spojení, malý pracovní prostor a lze sledovat pouze jeden objekt.

1.4.2 Výstupní zařízení

Výstupní zařízení poskytují smyslové vjemy uživateli a pomocí nich se jej snaží vtáhnout do virtuálního prostředí. Patří sem zařízení produkující obraz, prostorový zvuk, a další. Ne všechny smysly jsou rovnocenné v množství informací předávaných do mozku. Naopak zde existují obrovské rozdíly. Podle [1] je podíl vnímání člověka rozdělen mezi jednotlivé smysly následovně:

- zrak70%
- sluch.....20%
- čich5%
- hmat4%
- chuť.....1%

Z čehož je patrné, že největší vliv na naše vnímání má zrak. Z tohoto důvodu je následující text zaměřen na obrazová zařízení zprostředkovávající vizuální vjemy.

Ke vtáhnutí uživatele do virtuálního prostředí je třeba monitor, který je schopný zobrazit hloubku prostoru. To je možné u *stereoskopického zobrazení*, které je podrobně popsáno podle [13].

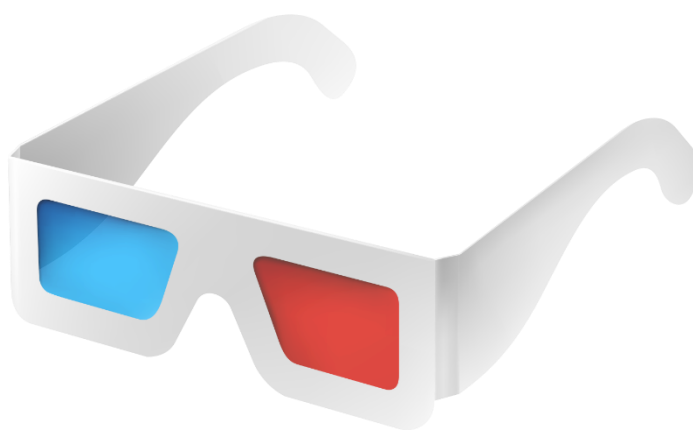
1.4.2.1 Stereoskopické zobrazení

Člověk vnímá hloubku (vzdálenost) objektů pomocí jemných rozdílů v zobrazení bodů na sítnici očí. Díky dvěma očím, které máme umístěny přibližně sedm centimetrů od sebe, dodává každé z očí do mozku mírně odlišný obraz. Stereoskopické zobrazení tedy vytváří dva obrazy, jeden pro každé z našich očí. To sebou přináší zvýšené nároky na výpočetní výkon, protože je nutné zdvojnásobit obnovovací frekvenci. Složitějším problémem je nutnost tyto obrazy od sebe oddělit, nebo je zobrazovat odděleně. Obrazy lze oddělit v čase, nebo v prostoru.

Na principu odděleného zobrazení v prostoru pracují HMD. Obrazy pro obě oči jsou přeneseny do zařízení a promítány na dvou displejích, které jsou umístěny tak, že se před každým okem nachází jeden displej. Obraz pro pravé oko je zobrazen na pravém a pro levé na levém displeji.

Druhou možností, jak oddělit obrazy, je jejich oddělení v čase. Na jednom zobrazovacím zařízení dochází k pravidelnému střídání obrazů pro každé z očí. Uživatel musí mít na očích tzv. *zatmívací brýle* (*shutter glasses*, *flicker lens*). Brýle mají polopropustné zorníky, které se střídavě zatmaví tak, aby propustily obraz pro pravé oko pouze do pravého oka a naopak. Brýle jsou synchronizovány s monitorem (pomocí infračervených paprsků).

Obrazy prezentované zároveň na jednom displeji lze od sebe oddělit optickými filtry. Nejjednodušší a nejlevnější metodou je využití barevných filtrů. Obraz pro jedno z očí vykreslíme v odstínech červené, pro druhé pak v modré barvě. Za pomoci všem známých modročervených brýlí (obr. 7) pak zajistíme viditelnost obrazů pouze v oku, pro které jsou určeny. Nevýhodou je omezení viditelnosti na odstíny dvou barev i když je obraz vykreslen v libovolných barvách. Další, pokročilejší metodou je využití polarizace světla. Polarizací lze dosáhnout dokonalého oddělení obrazů. Světelná vlna obou obrazů je polarizována. Roviny polarizace vln obou obrazů jsou na sebe navzájem kolmé. Brýle mají v zornících polarizační filtry. Filtry mají taktéž polarizaci kolmou jeden na druhý a propustí tak pouze požadovaný obraz pro každé z očí. Nevýhodou pak je zcela zřejmá náchylnost na polohu brýlí, a tedy i uživatelské hlavy. Při jakémkoliv natočení propustí filtry kromě jím určeného i část druhého obrazu. Při pootočení o devadesát úhlových stupňů se dokonce obrazy zcela vymění a pravé oko uvidí obraz určený pro levé.



Obr. 7: Papírové 3D brýle [14]

1.4.2.2 Náhlavní displeje – HMD

Jak už bylo zmíněno, HMD (obr. 8) jsou nejznámější a nejvíce rozšířenou technologií z oblasti VR. Jak už název vypovídá, jde o nositelnou elektroniku, kterou si uživatel nasazuje přímo na hlavu. Mají velmi široké spektrum využití od hraní her až po aplikace v armádě, medicíně a strojírenství.

V [15] je tahle technologie popsána následovně.

Nejdůležitější částí HMD je pochopitelně samotný displej. Nejběžněji se jedná o LCD (*liquid crystal display*), které jsou v novějších zařízeních nahrazovány panely OLED (*organic light-emitting diode*).

Základním parametrem u takových displejů je jejich rozlišení. Udává se jako počet obrazových bodů na délku palce. Dnešní panely dosahují hodnot kolem 500 ppi.

Obnovovací frekvence je též velmi důležitá. Ovlivňuje plynulost pohybu a má nezanedbatelný vliv na úroveň realističnosti, jaké lze s HMD dosáhnout. Je definována jako počet snímků, které zvládne displej promítnout za jednu sekundu. Běžné monitory dnes dosahují 60 fps (*frames per second*), ale ani monitory s dvojnásobnou a lepší obnovovací frekvencí nejsou vzácností.

Neméně důležitým faktorem je odezva. Může být vysvětlena jako čas potřebný k přizpůsobení obrazu virtuálního světa vůči nové poloze uživatelské hlavy. Jelikož se VR snaží co nejlépe přiblížit skutečnosti, je nutné dosáhnout co možná nejnižších hodnot. Zařízení by měla dosahovat odezvy maximálně 20 milisekund.

Optické čočky jsou další součástí, bez které by HMD nemohly fungovat a která velice ovlivňuje jejich výslednou kvalitu. Přestože se obrazový panel v HMD nachází blízko očí, stále by bez vhodných čoček nepokryl celé zorné pole uživatele. Pomocí nich lze dosáhnout FOV (*field of view*) alespoň mezi 90 až 100 stupni, které je potřebné pro přesvědčivý VR obraz.

HMD též obsahují trackery snímající polohu uživatelské hlavy.



Obr. 8: HTC VIVE PRO s ovladači a snímači [16]

1.4.2.3 CAVE – cave automatic virtual environment

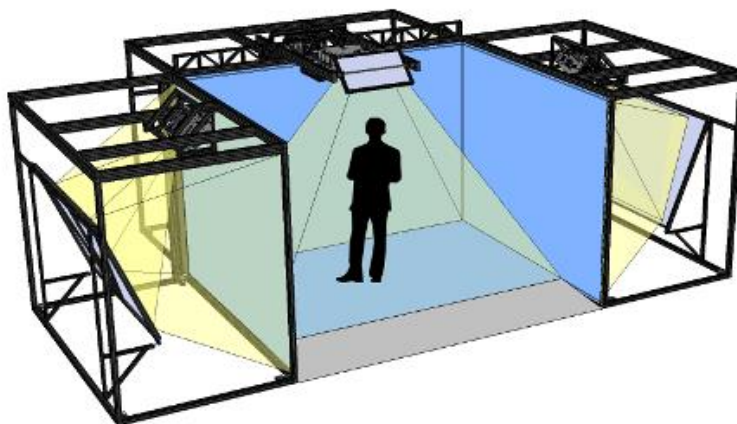
Nazýváme tak plně imerzní prostředí tvořené místností na jejíž stěny včetně podlah a stropů jsou promítány obrazy sérií projektorů (obr. 9). Typické vybavení takové místnosti je následující:

- promítací stěny
- projekční podlaha
- reproduktory rozmístěné po celém prostoru
- trackovací senzory ve stěnách
- mediální soubory (zvuk a video)

Obrazy promítané na stěnách jsou stereoskopické. Osoba uvnitř CAVE tedy musí mít na očích brýle kompatibilní s danou technologií stereoskopického obrazu. Obrazy jsou generované velmi výkonnými počítači, které zpracovávají signály ze senzorů zaznamenávajících pozici uživatele a vykreslují obrazy promítané na stěny [17].

Interakce uživatele s VR prostředím uvnitř CAVE může probíhat několika způsoby. Podle [18] to jsou následující:

- přímá – jak název vypovídá uživatel interaguje se systémem přímo. Příkladem může být zvednutí krabice.
- fyzická – k interakci dochází pomocí fyzického zařízení. Např. pomocí volantu, rukavic atd. Příkladem může být nastartování motoru kliknutím na tlačítko na volantu.
- virtuální – je velice podobná fyzické interakci až na to že ovladač je pouze reprezentován ve virtuálním prostředí. Jako příklad může posloužit zapnutí světel kliknutím na virtuální vypínač.
- pomocí operátora – operátor je fyzická osoba nebo program, který provádí změny v prostředí na základě požadavků uživatele.



Obr. 9: Jedna z možných realizací CAVE systému [19]

2 HERNÍ ENGINY

Vývoj her dříve probíhal jiným způsobem než dnes. Pokud chtěli vývojáři vyvinout novou hru, museli kromě jednotlivých modelů, textur, levelů a dalších součástí nově vytvořit i věci jako je herní fyzika, scripty pro načítání a vykreslování obsahu a kolize objektů. Ty jsou ale u her stejného typu velice podobné a souhrnně se nazývají *herní engine*. V následující kapitole je vysvětleno, co pojem herní engine znamená, jaké části nemohou takovému enginu chybět, krátce rozebrána historie vývoje enginů a zmínění nejznámější zástupci.

2.1 Definice

Pojem herní engine není snadné přesně definovat. Nevztahuje se totiž k ničemu konkrétnímu. Mnohem snazší je definovat jeho funkci. Jedná se o nástroj, který slouží k provádění úkolů, mezi které patří vykreslování, fyzika, vstupy a další, aby se vývojáři mohli zaměřit na detaily své hry [20].

K pochopení, jak to celé funguje, je vhodné si nejprve uvědomit, jak fungují počítače samotné. Vytvořit jakýkoliv program by bylo neuvěřitelně komplikované, pokud bychom museli neustále myslet na to, že počítač používá kolísání elektrického signálu pro provádění aritmetických operací. Mnohem jednodušší je s počítačem komunikovat pomocí některého z programovacích jazyků, který převede kombinace nul a jedniček na srozumitelné příkazy. Herní engine tedy umožňuje vykreslit požadovanou scénu (např. avatar uvnitř místnosti) bez toho, aniž bychom se museli zabývat jaký elektrický impuls je potřeba poslat do monitoru k rozsvícení správného pixelu [21].

2.2 Historie

Jak už bylo zmíněno, herní engine je součástí každé hry. S rostoucími požadavky na zefektivnění vývoje nových her ale přestalo být ekonomicky únosné tvořit originální engine pro každou novou hru. Větší společnosti tedy začaly své enginey prodávat. V poslední době pak poskytovat zadarmo, kdy vývojář používající cizí engine začne platit, až pokud jím vytvořený projekt dosáhne určitého výdělku. Firmy se dnes tedy mohou rozhodnout, zda se jim oplatí vyvinout vlastní engine, nebo využijí části, popřípadě celý, již funkční a vyzkoušený engine.

První enginey vznikaly v devadesátých letech minulého století. Ty tvořily trojrozměrné objekty pouze pomocí dvourozměrných tvarů. Patří sem např. *Doom engine* (1993) společnosti ID software nebo *Voxel engine* (1992) společnosti NovaLogic. Prvním skutečně 3D enginem se stal *XnGine* (1995) společnosti Bethesda Softworks. Prvním herním 3D enginem pak *Quake engine* (1996) společnosti ID software. Ten využíval např. funkce Z-buffering. Tahle funkce vypočítává, která část scény je viditelná, a tedy zpracovávána enginem [22].

V následujících letech vzniklo mnoho enginů, většinou vyvinutých velkými herními společnostmi jako jsou např. *CryEngine* (2004), *Source engine* (2005), *Unity* (2005), *Unreal engine* (2008). Ty úspěšné, jako jsou třeba právě Unity a Unreal engine, byly vydány již v několika verzích a jejich vývoj pokračuje do současnosti. S každou novou verzí přináší nové technologie v oblasti vykreslování, práce se soubory, zvukem, rozšíření podporovaných platforem a v neposlední řadě ve vytváření co možná nejprívětivějšího uživatelského rozhraní [23].

V dnešní době jsou herní enginey natolik vyspělé systémy, že už se dávno nepoužívají jen pro tvorbu her. Nalezly uplatnění i v oblastech jako jsou filmy, vývoj ve strojním průmyslu, nebo vytváření virtuálních prostředí a simulací [24].

2.3 Komponenty

Herní enginey tvoří velice různorodou skupinu. Existují enginey vhodné pro závodní hry, střílečky z první osoby, strategie atd. Ty se všechny fundamentálně liší, a tedy lze očekávat i velké rozdíly v enginech, na kterých dané hry běží. Přesto je několik částí, které by měl každý engine obsahovat. Podle [25] to jsou:

- zdrojový manažer – každá hra obsahuje velké množství zdrojových dat. Patří sem data dvou typů: *mediální* a *behaviorální* (týkající se chování). Mediální data obsahují obrázky, zvuky a animace. Behaviorální pak např. data obsahující informace o nastavení hry. Těchhle dat může být opravdu velký objem, a tudíž je potřeba oddíl, který je spravuje. Jeho hlavními funkcemi jsou: identifikace a rozlišení jednotlivých souborů, jejich načítání a ukládání do mezipaměti a ujišťování se, zda se v souborech hry nevyskytuje více souborů se shodným jménem.
- vykreslovací manažer – má na starosti vykreslení načtených grafických souborů na obrazovku. Toho je dosaženo pomocí grafických vykreslovacích knihoven jako jsou např. OpenGL, DirectX, nebo SDL. Jeho hlavními úkoly jsou: efektivní komunikace mezi enginem a grafickým hardwarem a vykreslování obrazových souborů na monitor v nastaveném rozlišení.
- vstupní manažer – zpracovává data ze vstupních zařízení ať už to je klávesnice a myš, nebo různé ovladače používané při hraní na konzolích. Hlavními funkcemi jsou: zpracovávat vstupní data z celého spektra podporovaných zařízení, převést signál na standardní signál nezávislý na vstupním zařízení.
- zvukový manažer – má prakticky stejnou funkci jako vykreslovací manažer, jen se zabývá zvukovými soubory. Stará se jak o podkladovou hudbu, tak o zvukové efekty jako např. střelbu, kroky aj. Jeho funkce jsou: komunikace mezi hrou a zvukovým hardwarem, nastavování hlasitosti,

aplikace efektů ke zvukům jako jsou např. ozvěna a ztlumení se vzdáleností.

- chybový manažer – každý engine je před svým vydáním testovaný, zda neobsahuje chyby. Vzhledem k obrovskému množství možných scénářů je ale nemožné všechny scénáře otestovat, a proto každý engine nějaké chyby obsahuje. Je tedy potřeba chybový manager. Jeho funkce je detekce chyb, zabránění náhlému vypnutí engineu při výskytu chyby, zobrazení chybového hlášení, uložení hlášení o chybě v čitelné formě a označit chybu číslem, které pomáhá k lepší klasifikaci chyby.
- scénový manažer – koordinuje vykreslovacího a zvukového manažera tak, aby jejich výstupy vytvořily smysluplnou scénu. K tomuto účelu musí zvládat několik funkcí, k nimž patří: komunikace mezi vykreslovacím, zvukovým, zdrojovým a vstupním manažerem, držení časové linie za účelem spouštění eventů a koordinace pohybu objektů, správně rozlišit objekty na scéně a umožnit tak vývojáři s nimi pracovat.
- fyzikální manažer – stará se o aplikování fyzikálních zákonů na herní objekty, přesněji síly a točivého momentu. Má tedy podobnou funkci jako scénový manažer, který na objekty aplikuje herní pravidla.
- skriptovací manažer – herní engine je vytvořen pomocí kompilovaného jazyka jako je C++ nebo C#. Použití takového jazyka zlepšuje výkonnost engineu. Má ale též za následek obtížnější modifikace jeho vlastností. Modifikace se provádí pomocí externích souborů psaných skriptovacími jazyky (*Python*, *Lua*) nebo angličtinou. Nezávisle na výběru jazyka je potřeba se postarat o jeho podporu a interakci s engineem. Hlavní funkce: odkrýt část složek a funkcí engineu a umožnit k nim přístup pomocí skriptovacího jazyka, přeložit a správně zpracovat instrukce v souborech.

2.4 Přehled současných engineů

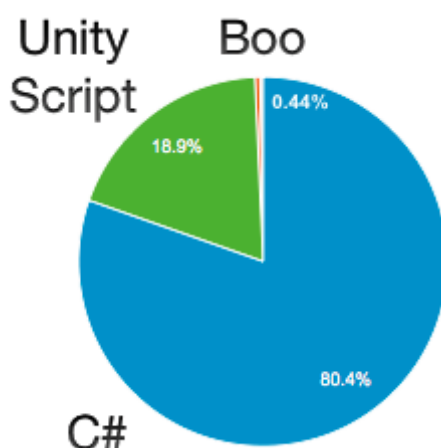
Herních engineů existuje obrovské množství. Jako všechno ostatní i jednotlivé enginey mají své výhody a nevýhody. Nelze jednoznačně určit, který z nich je celkově nejlepší. Nicméně každý se hodí lépe na jinou aplikaci, proto je nutné výběru vhodného engineu věnovat dostatečnou pozornost. Práce je zaměřena na vytvoření prostředí pro virtuální realitu. K tomuto účelu jsou nejvhodnější herní enginey primárně určeny k vytváření her typu FPS (*first-person shooter* – střílečka z první osoby). V následující části kapitoly je uvedeno několik nejznámějších engineů [26].

2.4.1 Unity

Unity [27] je herní engine vyvíjený společností Unity Technologies. Společnost sama hry nevyvíjí, zaměřuje se pouze na zdokonalování svého enginu. Možná právě proto se jedná o jeden z nejvšestrannějších dnes dostupných enginů. Lze jej s výhodou použít na 2D i 3D projekty, kdy pro 2D projekty se jedná o jeden z nejlepších vývojářských nástrojů vůbec. Přestože Unity nenabízí tak pokročilé funkce jako nejlepší konkurenční engine, pro přívětivé uživatelské rozhraní a dobře nastavenou cenovou politiku, je využíván mnoha vývojářskými studiemi. V Unity byla vytvořena více jak polovina všech mobilních her a také obsahu pro rozšířenou a virtuální realitu. Pro svou jednoduchost a přehlednost je také vhodný pro začínající vývojáře.

Unity je multiplatformní herní engine, v současné době podporuje 27 různých platforem včetně mobilních telefonů, stolních počítačů, herních konzolí a zařízení pro virtuální a rozšířenou realitu. Tak široké uplatnění nenabízí žádný jiný engine.

Ke skriptování v Unity slouží jazyky C# a *UnityScript* (také známý jako *JavaScript* pro Unity). Dříve byl podporován ještě jazyk *Boo*, ten ale již není podporován kvůli nízké oblíbenosti (obr. 10) [28].



Obr. 10: Zastoupení skriptovacích jazyků ve skriptech vytvořených v Unity [28].

Unity je zdarma v odlehčené verzi pro osobní využití nebo pro malé společnosti, jejichž roční příjem nepřekročí 100 000 dolarů. Pro větší společnosti je pak zpoplatněno tarifně. Společnosti vydělávající do 200 000 dolarů ročně zaplatí 35 dolarů za měsíc, úspěšnější pak 125 dolarů.

Mezi hry vytvořené v Unity patří např.: *Escape from Tarkov*, *Inside*, *Ori and the Blind Forest*, *Hearthstone: Heroes of Warcraft*, *Superhot* a mnoho dalších [29].

2.4.2 Unreal Engine 4

Unreal engine 4 [30] je poslední verzí známého herního enginu společnosti Epic Games. Jedná se o v současnosti nejpokročilejší herní engine, který je schopen produkovat úžasné grafické scény obzvláště z futuristického prostředí. Je primárně určený pro FPS hry, má ale vše potřebné pro tvorbu her jakéhokoliv žánru. Další možné využití je pro tvorbu filmových klipů, virtuálního a rozšířeného prostředí, kdy je možné spustit Unreal Editor přímo ve virtuální realitě a tvořit tak uvnitř virtuálního prostředí. Díky své komplexnosti a programování v jazyce C++, který je složitější než C#, je obtížnější na zvládnutí než Unity, a proto se nedoporučuje začátečníkům.

Stejně jako Unity je i zde podporována většina dostupných platform užívaných na mobilních zařízeních, počítačích i konzolách a zařízeních pro virtuální realitu.

Unreal Engine je zcela zdarma. Epic Games si účtují pouze 5 % z výdělku, pokud čtvrtletní tržby z vydaného produktu překročí 3 000 dolarů. Epic Games také uvolnilo 100 miliónů dolarů, ze kterých sponzoruje nadějně projekty částkou 5 až 500 tisíc dolarů. Navíc je engine poskytován v otevřené formě a vývojář tedy může zasahovat do zdrojového kódu enginu napsaného jazykem C++.

V předešlých verzích enginu se používal *UnrealScript* podobný jazyku Java. Ke skriptování se v poslední verzi enginu používá výhradně jazyk C++. Druhou možností vytváření funkčních interakcí je pomocí tzv. *Blueprintů*. Ty umožňují vytvářet funkční interakce (např. interaktivní objekty – obr. 11) pomocí grafického spojování uzlů představujících jednotlivé funkce a objekty. Dělalí tak programování mnohem jednodušší a názornější. To je obrovská výhoda obzvláště pro lidi, kteří se programováním přímo nezabývají. Umožňují tak např. daleko lepší spolupráci mezi programátory a ostatními členy týmu.



Obr. 11: Příklad využití Blueprintů k vytvoření interaktivního objektu [31].

Při výčtu nejznámějších her jejichž základ tvoří Unreal Engine nesmí být zapomenuto na Unreal Tournament, který vyšel v několika verzích vždy společně s novou verzí enginu. Dalšími pak jsou: Ark: Survival Evolved, Dead Island 2, Sea of Thieves, Ace Combat 7: Skies Unknown, Fortnite: Save the World a mnoho dalších [32].

2.4.3 CryEngine

CryEngine je produktem německé firmy CryTek [33]. Ta jej používá ve všech svých hrách, mezi které patří např. hra Crysis. Jedná se o velice výkonný engine, který vyniká především ve vytváření realistických grafických scén, kdy ve ztvárnění přírody je nejlepší ze všech. K vytváření levelů se používá tzv. *Sandbox* (obr. 12). Ten je koncipován podle hesla: WYSIWYG (What You See Is What You Get). Nabízí soubor nástrojů umožňující tvorbu a úpravy v reálném čase. Jedná se o engine určený primárně pro AAA (označení pro hry s nejvyššími rozpočty na vývoj a reklamu) FPS hry. K užití v jiných typech her se příliš nedoporučuje [34].

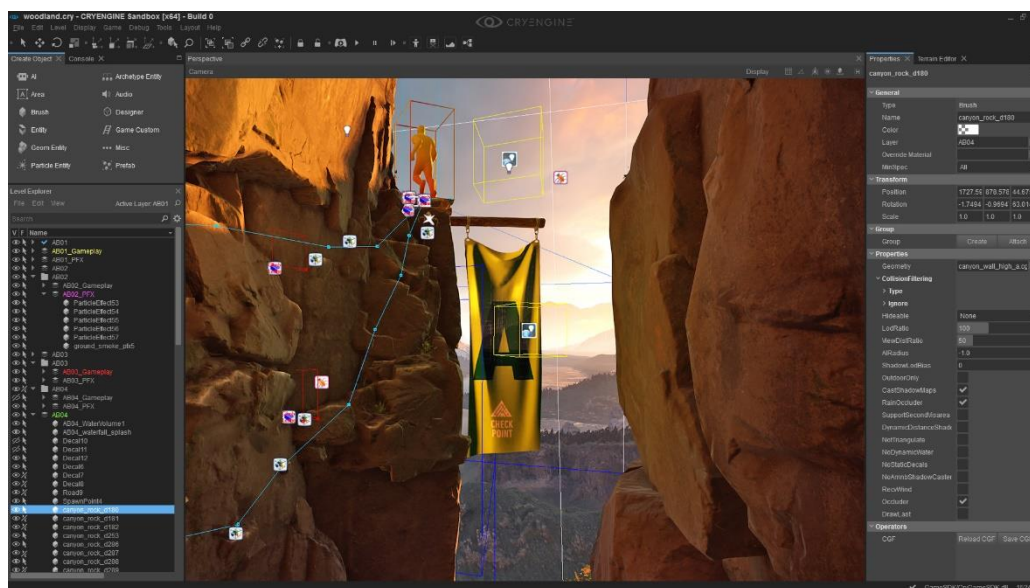
Finanční problémy, které firmu CryTek sužují od roku 2014 společně s velice úzkým zaměřením enginu neblaze ovlivňují četnost jeho užití. CryEngine není zdaleka tak rozšířený jako třeba konkurenční Unreal Engine nebo Unity [35] [36].

Ke skriptování slouží jazyky Lua, C++ a v poslední verzi také C#. Dále je možnost stejně jako v Unreal Enginu tvořit skripty pomocí grafického rozhraní zde nazvaného Flow Graph.

Mezi podporované platformy patří Oculus Rift, Windows PC a z konzolí to jsou Xbox One a Playstation 4. To představuje pouze zlomek platforem podporovaných výše zmíněnými enginy. Podpora mobilních zařízení tady úplně chybí.

CryEngine je zcela zdarma. Při použití pro komerční účely činí poplatek 5 % z tržby za produkt. Na tržby do 5 000 dolarů za produkt během jednoho roku se poplatek nevztahuje. Lze využívat i pro akademické účely, vytvořené projekty však nesmí být zpeněženy bez souhlasu společnosti CryTek.

Mezi hry vytvořené tímto enginem patří Crysis (CE2 – CryEngine 2), první díl série Far Cry (CE 1), Warface (CE 3), Snow (CE5) nebo česká hra Kingdom Come: Deliverance (CE 3.6-4) [37].



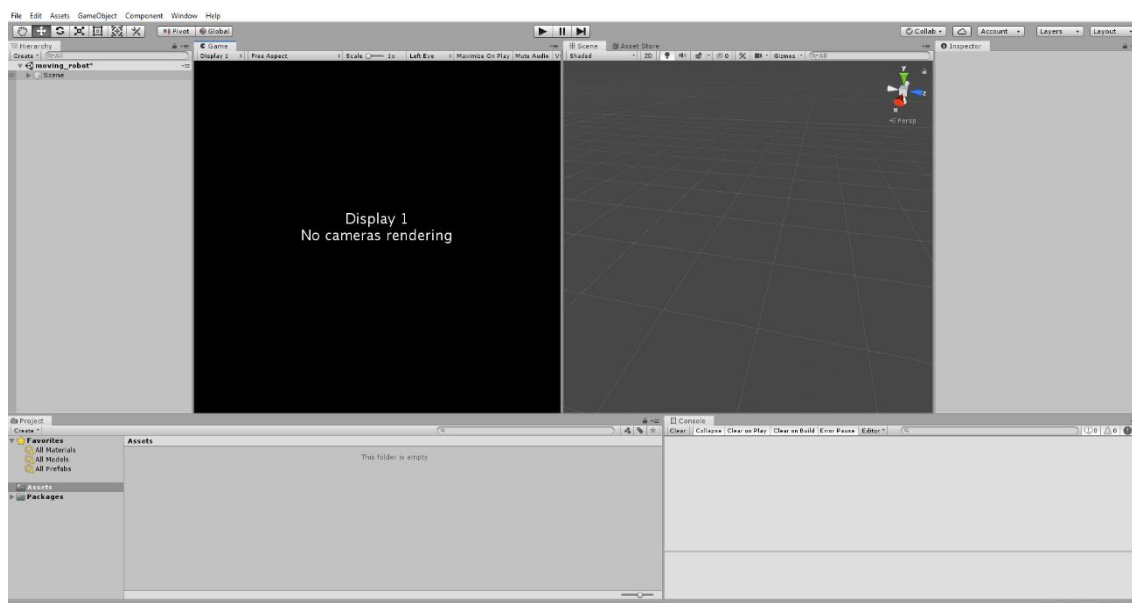
Obr. 12: CryEngine Sandbox [33]

3 PRAKTICKÁ ČÁST

Praktická část práce představuje tvorbu prostředí pro virtuální realitu. Podrobný popis prostředí pro VR je uveden v následující kapitole. K tvorbě scény byl zvolen herní engine Unity. Představuje totiž v současnosti nejlepší alternativu pro začínajícího vývojáře. Dále je též vhodný pro tvorbu VR prostředí. Testování probíhalo na zařízení *HTC Vive Pro*, které je dostupné na Ústavu automatizace a informatiky VUT v Brně.

3.1 Uživatelské rozhraní v Unity

Standardní uživatelské rozhraní vývojového prostředí engine Unity je zobrazeno na obr. 13.



Obr. 13: Uživatelské rozhraní engine Unity

Uživatelské rozhraní lze libovolně měnit. Je možné měnit velikost jednotlivých oken přesouváním jejich hranic. Okna se taky dají jakkoliv přeskládat, dokovat do sebe, nebo vyjmout ze struktury a vytvořit tak samostatné okno.

V levém horním rohu se nachází lišta záložek. V liště se nachází záložky, pomocí kterých lze scénu nebo celý projekt uložit, vytvářet jednotlivé části jako jsou *assets*, herní objekty (*GameObject*) a komponenty (*Component*).

Pod záložkami se nachází lišta s přepínači transformací. Zde je možné vybrat, zda se označeným objektem bude posouvat, rotovat s ním, nebo měnit jeho velikost. Dále se za touto lištou nachází dva přepínače, první mění začátek souřadnicového systému u objektu mezi jeho středem a zvoleným *pivotem*. Druhý pak určuje, zda se bude souřadnicový systém otáčet společně s objektem, nebo zda bude udržovat směr kartézských souřadnicových os. Dále se nachází za těmito přepínači uprostřed obrazovky

tři tlačítka. První tlačítko spouští simulaci vytvářené hry. Po spuštění jde simulaci tímto tlačítkem opět vypnout. Druhé tlačítko simulaci pozastaví. Třetím tlačítkem se lze při pozastavené simulaci posouvat o jednotlivé snímky dopředu.

Na zbytku plochy se nacházejí jednotlivá okna. Jak již bylo zmíněno, jejich konfiguraci lze libovolně měnit. V zásadě jsou zde přítomna tato okna:

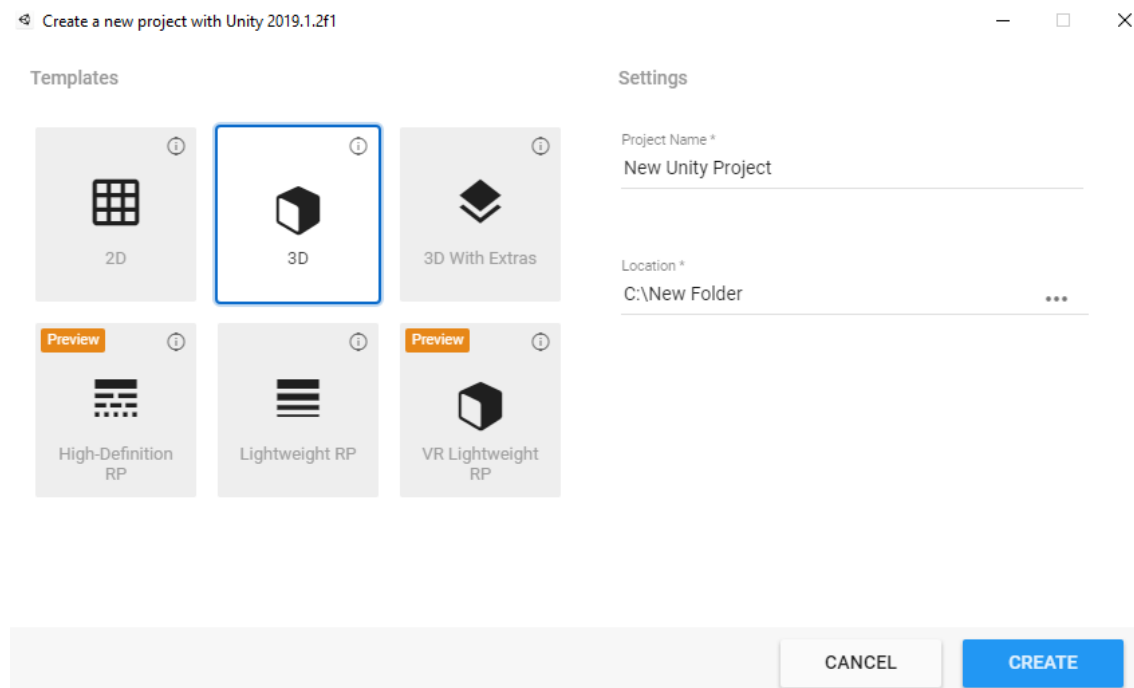
- Hierarchie (*Hierarchy*) – je zde zobrazena složka scény a všechny herní objekty, které se v ní nacházejí.
- Projekt (*Project*) – jsou zde zobrazeny všechny soubory importované v projektu včetně skriptů, materiálů, textur atd. Z okna projektu je možné objekty přesouvat na scénu.
- Konzole (*Console*) – v konzoli dochází k vypisování chyb. Taky je zde možnost, si do konzole nechat vypsát údaje pomocí skriptů přidaných k objektu. Jedná se např. o souřadnici objektu, nebo o hodnoty různých proměnných.
- Inspektor (*Inspector*) – zobrazují se zde vlastnosti právě vybraného herního objektu. Jeho transformace, přiřazené skripty a komponenty, které zde lze k objektu přidávat.
- Scéna (*Scene*) – přetažením objektu z okna Projekt do okna Scéna ho umístíme na scénu. Lze zde s objekty libovolně přesouvat. Jedná se o hlavní pracovní prostor.
- Hra (*Game*) – je okno sloužící pro simulace. Při spuštěné simulaci je zde vidět scéna pohledem hráče. Dochází zde k testování vytvořené scény.

3.2 Podpora VR v Unity

Aby bylo možné se ve scéně pohybovat jako ve VR prostředí, je nutné nejdříve provést několik kroků.

Počátečním krokem je vytvoření projektu v programu *Unity Hub* (verze 2.0.0), který slouží jako instalátor a spouštěcí program enginu. V něm, po kliknutí na tlačítko *New*, vyskočí následující okno (obr. 14), ve kterém se zvolí jméno projektu a složka, do

které se bude projekt ukládat. Dále se zde provádí volba optimalizace pro 2D nebo 3D prostředí a několik dalších specializovaných možností. V této práci byl zvolen 3D projekt.

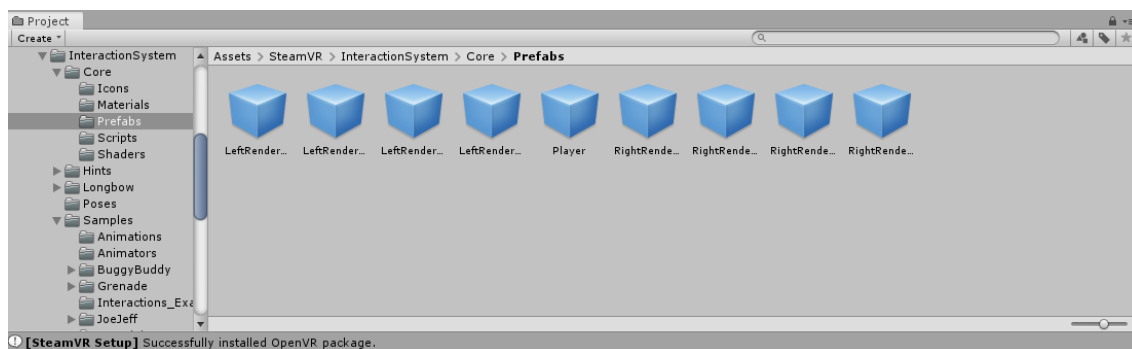


Obr. 14: Tvorba nového projektu v Unity Hub

Dále je do vytvořeného projektu nutné importovat *plugin SteamVR* z Asset Store, který lze otevřít přímo z Unity v záložce Okno (*Window*). *Plugin* obsahuje velké množství předpřipravených skriptů a prefabrikátů, které velice usnadňují práci.

Posledním krokem je přetažení prefabrikátu Hráč (*Player*) (obr. 15) na scénu, jenž lze nalézt pomocí vyhledávání ve složce Projekt. Jeho pozici lze upravovat přímo přesouváním v okně scéna nebo pomocí zadání souřadnic v okně Inspektor.

Tento prefabrikát zajišťuje jak správné fungování HMD, tak i obou ovladačů. Tyto ovladače jsou vykreslovány na scéně v pozici a orientaci, které zaujímají ve skutečnosti a mají i namapována jednotlivá tlačítka.



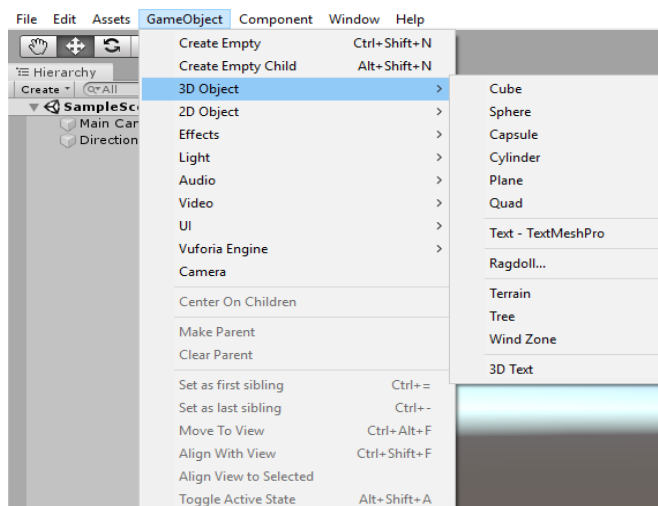
Obr. 15: Prefabrikát Hráč

3.3 Návrh scény

Práce je koncipována tak, aby demonstrovala základní vlastnosti engine. Za tímto účelem bylo vytvořeno prostředí, ve kterém se nachází basketbal, bowling, střelnice a galerie.

3.3.1 Modelování

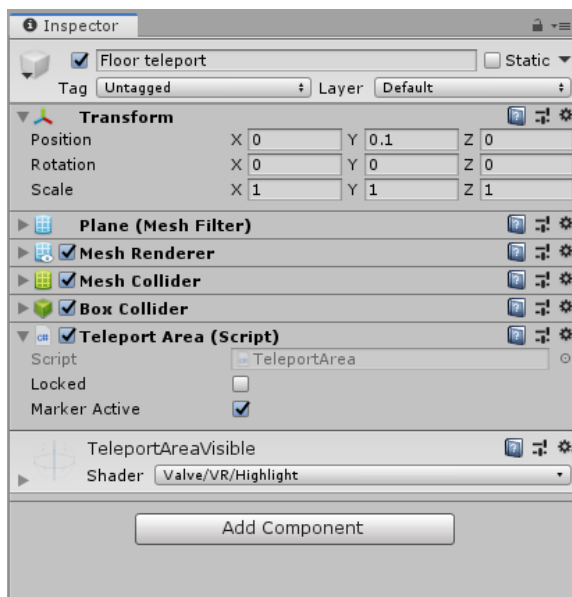
Modelovat ve 3D projektu engine Unity lze pouze přidáváním jednoduchých tvarů, jako jsou kostky, koule, válce atd. (obr. 16). Jejich rozměry a rotace lze libovolně volit ve všech souřadnicových osách kartézského systému. Vytvořit jednoduché objekty, jako jsou místnosti nebo stoly lze relativně lehce. Není však umožněno materiál odebírat, proto k vytvoření např. otvoru pro dveře je nutné stěnu rozdělit na více dílů a ty seskládat tak, aby vznikl požadovaný otvor. K modelování složitějších objektů pak engine není vhodný vůbec. Je nutné využít některý z dostupných modelovacích softwarů jako je *Blender* nebo *Autodesk Maya*. Modely pak lze importovat do engine ve formátu *.FBX* nebo *.OBJ*.



Obr. 16: Vytváření 3D objektů v Unity

3.3.2 Pohyb

V důsledku zvolených rozměrů scény, bylo nutno po ní zajistit efektivní pohyb hráče. S vyřešením tohoto problému opět pomohl plugin SteamVR. Ten obsahuje prefabrikát *Teleporting*. Jeho přetažením do okna Hierarchie je umožněna teleportace hráče stiskem tlačítka *Touchpad* na ovladači. To ale k úspěšnému přemísťování nestačí. Povrchy a body vhodné k teleportování je nutné na scénu nejprve přidat. Lze tak učinit přidáním objektu *TeleportPoint* nebo skriptu *Teleport Area*, který je třeba aplikovat přímo na Herní Objekt (např. na rovinu představující podlahu) pomocí tlačítka *Add Component* (obr. 17).



Obr. 17: Teleport Area (Script)

3.3.3 Kompozice

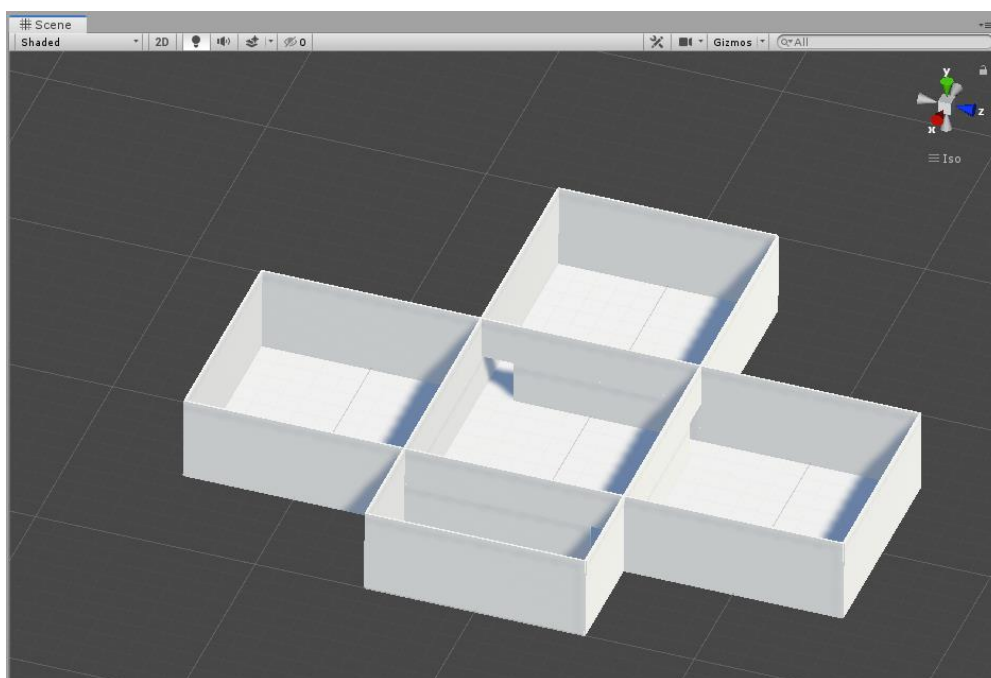
Scéna je rozdělena na pět menších scén. Jednotlivé scény mají tvar místností o čtvercovém a obdélníkovém půdorysu, centrální místnost slouží jako spojovací chodba. Ve čtyřech bočních místnostech jsou demonstrovány následující vlastnosti engine:

- Fyzikální model – fyzikální model je demonstrován na hře basketbal. Zobrazuje se zde především působení gravitační síly na objekty letící vzduchem.
- Fyzická interakce mezi objekty – je demonstrována na hře bowling. Znázorňuje chování objektů při vzájemném dotyku. Dochází zde ke tření mezi koulí a dráhou a předání energie při nárazu koule do kuželek.
- Importování 3D modelů – možnost importovat modely z externích zdrojů je demonstrována na místnosti s galerií. Nachází se zde dvě robotická ramena a jeden model humanoida. Modely jsou rozhýbány přidáním skriptů. Je zde znázorněna hlavně realističnost, jaké lze ve virtuální realitě dosáhnout.
- Složitější interakce mezi objekty – jsou demonstrovány na střelnici. Zde je možné pomocí pistole sestřelovat terče. K zachycení správné funkce střelby je třeba větší počet složitějších skriptů.

3.4 Samotná tvorba

Po vytvoření nového projektu a přidání prefabrikátů Hráč a Teleporting způsobem popsáným výše, byla započata práce na samotném modelování. Počátečním krokem bylo vytvoření podlahy a stěn.

Podlaha je tvořena několika vodorovnými plochami. Podlaha prostřední místnosti, stejně jako třech bočních místností, je tvořena čtvercem o straně délky deset metrů. Čtvrtá boční místnost na bowling má tvar obdélníku. Strany kolmé na centrální místnost mají délku pouze pět metrů. Stěny jsou pak vytvořeny protáhnutím krychle. Jejich šířka je nastavena na desetinu metru a výška na čtyři metry. Průchody do jednotlivých místností si vyžádaly rozdělení všech vnitřních stěn na dva díly, jak lze vidět na obr. 18.



Obr.18: Vymodelované podlahy a stěny

Podlaha, stejně jako všechny stěny, mají přiřazeny komponenty typu *collider*. Komponent lze přidat pomocí tlačítka Add Component. Collider zamezí propadnutí hmotných objektů. Hmotný objekt musí mít též přiřazen collider.

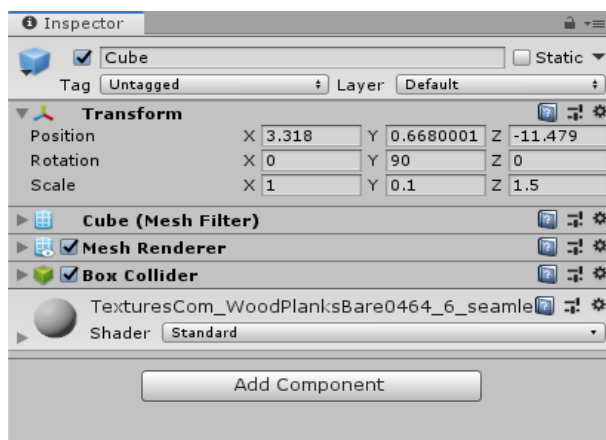
Dále následovalo vytvoření ploch sloužících k teleportování. Po aplikaci skriptu Teleport Area na objekt se tento objekt stane ve hře neviditelným. Bylo tedy nezbytné vytvořit nové plochy, které jsou mírně vyvýšeny oproti plochám podlahy a na ty následně aplikovat zmíněný skript.

Dalším krokem bylo modelování jednotlivých scén. Postup jejich tvorby je popsán v následujících podkapitolách.

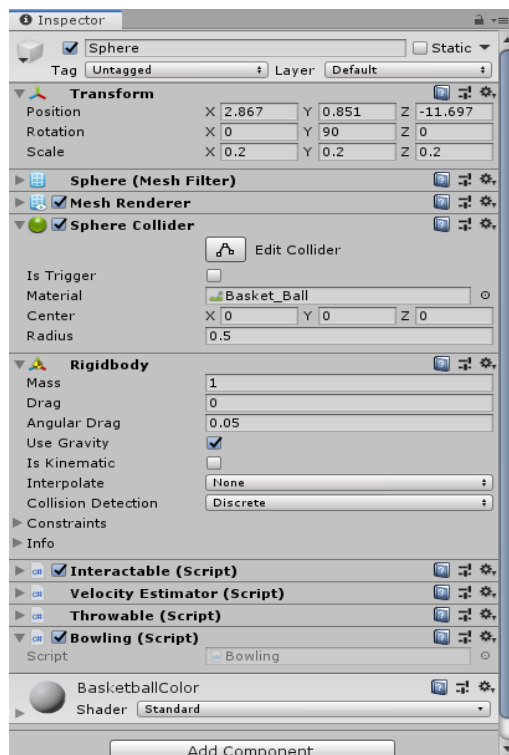
3.4.1 Basketbal

Scéna pro hru basketbal obsahuje koš, stůl a několik basketbalových balónů. Jak již bylo zmíněno, vymodelovat koš je přímo v unity prakticky nemožné. Bylo tedy rozhodnuto o importování koše z volně dostupného internetového zdroje [38].

Stůl byl vytvořen pomocí čtyř válečků a na nich umístěné desky, která vznikla opět deformováním krychle. Deska stolu má, stejně jako podlaha a stěny, přiřazen Collider (obr. 19 a). Míče byly vytvořeny pomocí 3D objektu koule. Byla jim kromě Sphere Collider přiřazena i komponenta *RigidBody*. V *RigidBody* lze objektu přiřadit hmotnost a zapnout působení gravitace. Dále zde byly přidány i skripty *Throwable* a *Interactable*. Oba skripty jsou dostupné ve SteamVR a zajišťují funkční interakci s ovladači headsetu (obr. 19 b).



Obr. 19 a: Komponenty desky stolu



Obr. 19 b: Komponenty basketbalového míče

Dalším krokem pak bylo rozdělení podlahy na dvě plochy. K tomuto kroku bylo přistoupeno kvůli následné snadnější aplikaci textur, kdy část u vstupu má jinou texturu než zbytek místnosti tvořící hrací plochu.

Posledním krokem pak byla aplikace textur. V projektu byly použity textury volně dostupné na internetu, např. z [39]. Obrázek s texturou lze jednoduše přetáhnout do složky v projektu (obr. 20) a následně ze složky na požadovaný objekt.



Obr. 20: Složka Materiály (*Materials*) obsahující vložené textury

Finální pohled na scénu se hrou basketbal po přidání všech objektů a textur je zobrazen na obr. 21.



Obr. 21: Pohled do scény se hrou basketbal

Konečnou fází vývoje VR prostředí bylo testování pomocí headsetu. Při testování byly pořízeny snímky obrazovky na připojeném počítači. Výsledný snímek je uveden jako obr. 22.



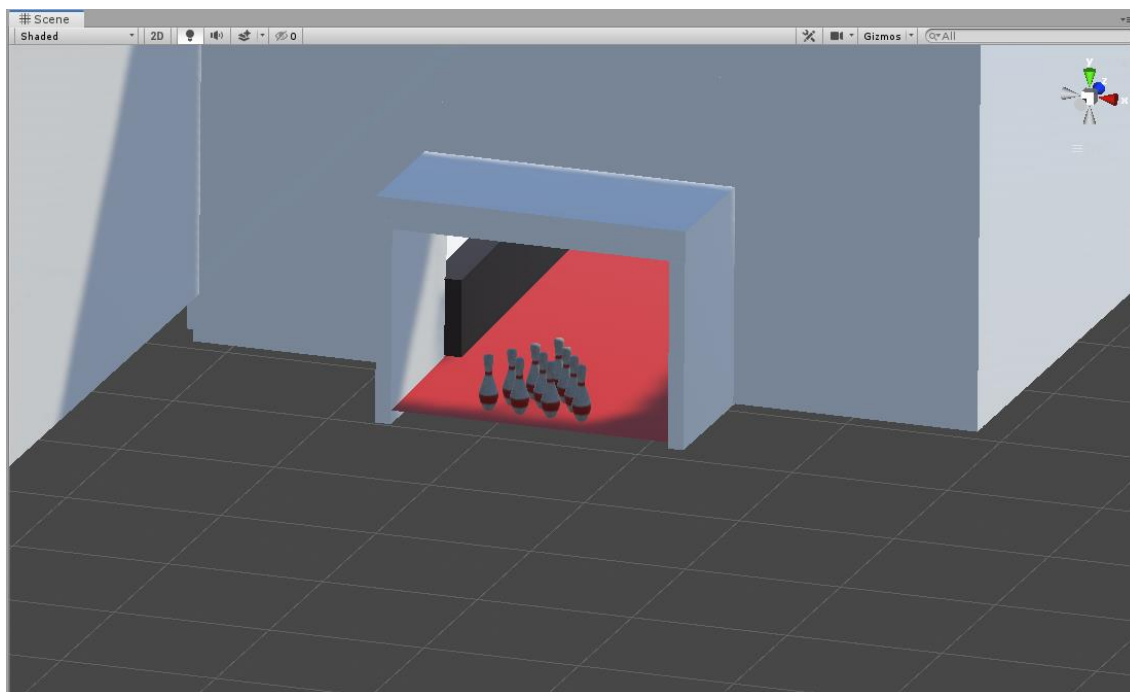
Obr. 22: Snímek obrazovky pořízený přímo z VR ve scéně se hrou basketbal

3.4.2 Bowling

Druhou scénou v pořadí je místnost se hrou bowling. Scéna obsahuje jednu dráhu na bowling, deset kuželek, tři kuželkové koule a podavač koulí. Modely kuželek i koulí byly opět importovány [40].

Přestože je místnost menší než místnost s basketbalem, je mnohem členitější. Podlaha byla rozdělena na čtyři plochy. První plocha tvoří podlahu u vchodu do místnosti a je též jediná, na kterou byl umožněn teleport. Zbylé tři plochy jsou sklopeny dolů pod úhlem dvou stupňů ve směru dráhy. Dráha je tvořena prostřední plochou. Po jejích stranách jsou zátarasy bránící nepřesně hozené kouli opustit dráhu. Zátarasy byly vytvořeny protáhnutím 3D objektu krychle. Prostor okolo dráhy vyplňují zbylé dva segmenty podlahy.

Část podlahy tvořící dráhu je delší než okolní části podlahy a je vysunuta otvorem v zadní stěně místnosti. Kuželky jsou postaveny na úplný konec dráhy. Po zásahu kuželek koulí dojde ke shoení kuželek ze scény (obr. 23).

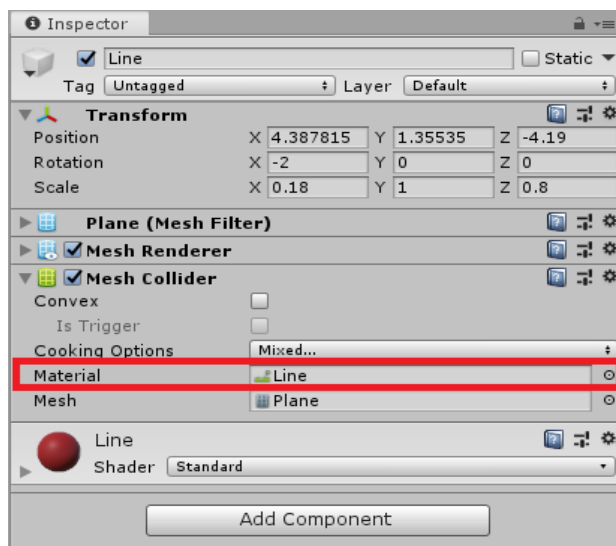


Obr. 23: Umístění kuželek na dráze

Z obrázku je patrné, že i koule po dojetí na konec dráhy spadne ze scény. Přidáním jednoduchého skriptu k objektu bowlingové koule bylo dosaženo jejího vrácení zpátky na scénu. Skript kontroluje vertikální souřadnici, na které se koule nachází. Klesne-li hodnota souřadnice pod definovanou hodnotu, je pozice koule resetována na pozici při spuštění hry. Obdobný, ale mírně složitější skript, je aplikován i na kuželky. Ten kromě sledování vertikální souřadnice zároveň počítá i spadlé kuželky a k resetování dojde až po shoení všech kuželek.

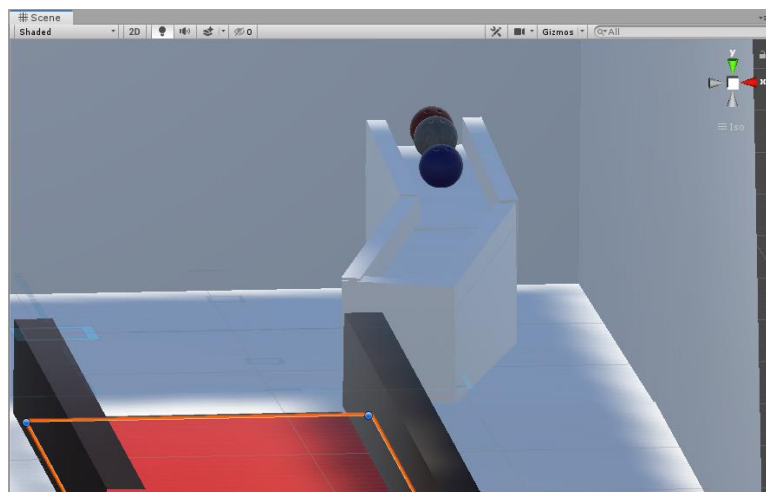
Vlastnosti chování materiálů při vzájemném styku se určují pomocí fyzikálních materiálů. Vytvořený fyzikální materiál se vloží do komponentu *Mesh Collider* (obr. 24). Fyzikální materiál lze vytvořit příkazem *Create* v záložce *Assets* a následným výběrem možnosti *Fyzikální Materiál (Physic Material)*. Ve vlastnostech materiálu lze nastavit statické i dynamické tření, odskakování objektů při dopadu a jejich násobiče.

U dráhy i koulí byl zvolen co nejnížší koeficient dynamického tření. Statické tření bylo rovněž sníženo z původní předdefinované hodnoty 0.6 na 0.1 u dráhy a 0.01 u koulí. Odskakování je u koulí i dráhy nastaveno na nejnížší, tedy nulovou hodnotu. Naopak u kuželek je hodnota nastavená na maximální, tedy jednotkovou hodnotu. Tímto je dosaženo lepšího efektu při zásahu kuželek koulí.



Obr. 24: Vložení fyzického materiálu

Podavač koulí byl vymodelován opět pomocí krychlí. Celkem se skládá ze sedmi objektů. Je také sklopen pod úhlem, tentokrát pěti stupňů. Je na něj taktéž aplikován fyzikální materiál dráhy, aby docházelo k co možná nejmenšímu tření mezi jeho povrchem a povrchem koulí. Podavač koulí koule dopravuje z jejich startovní pozice k začátku bowlingové dráhy tak, aby bylo možno pohodlně házet (obr. 25).



Obr. 25: Podavač na koule

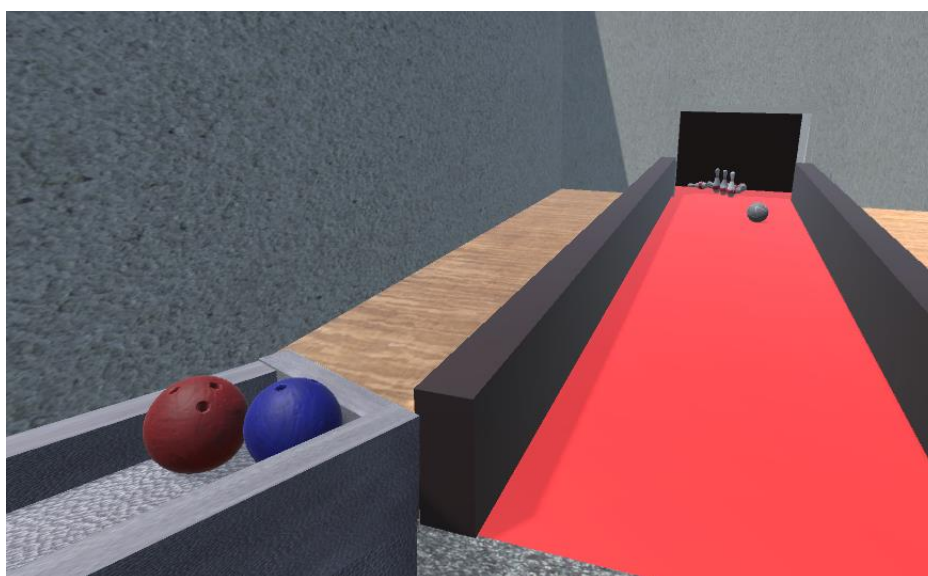
Koule jsou přítomny celkem tři. Každá z nich má jinou hmotnost. Hmotnost koulí se volí v komponentu RigidBody. Koule mají hmotnost nastavenou na 13, 15 a 17 jednotek. To odpovídá stejnému počtu kilogramů. Kuželky pak mají hmotnost nastavenou na 1.6 kg. Stejně jako basketbalový míč i koule musí mít komponenty Throwable a Interactable, aby s nimi bylo možné házet pomocí ovladačů.

Posledním krokem bylo opět přidání textur na všechny objekty. Většina textur byla importována ve formě obrázku z internetu [39] zbylé, např. textura dráhy a zátarasů, byly vytvořeny přímo v Unity v záložce Assets příkazem Create a výběrem možnosti Material. U vytvořeného materiálu lze měnit barvu, odlesky atd. Výsledný pohled na scénu je vyobrazen na obr. 26.



Obr. 26: Pohled do scény se hrou bowling

Při závěrečném testování funkčnosti aplikace byl pořízen následující snímek (obr. 27).



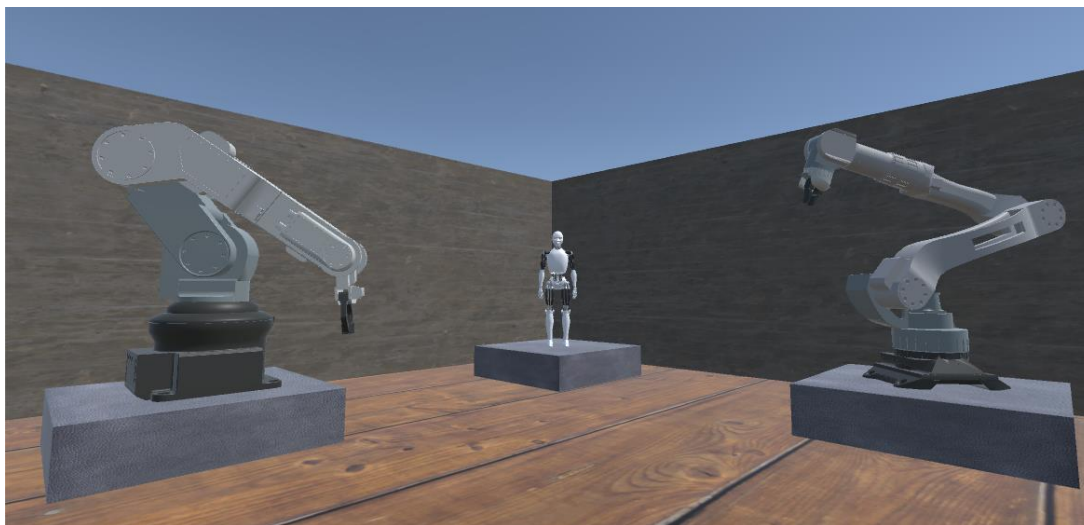
Obr. 27: Snímek z testování hry bowling

3.4.3 Galerie

V další místnosti je umístěna třetí scéna nesoucí název galerie. Nacházejí se v ní tři importované 3D modely. Modely byly staženy ze stránky free3D.com [38]. Modely jsou na scéně umístěny na vytvořených podstavcích.

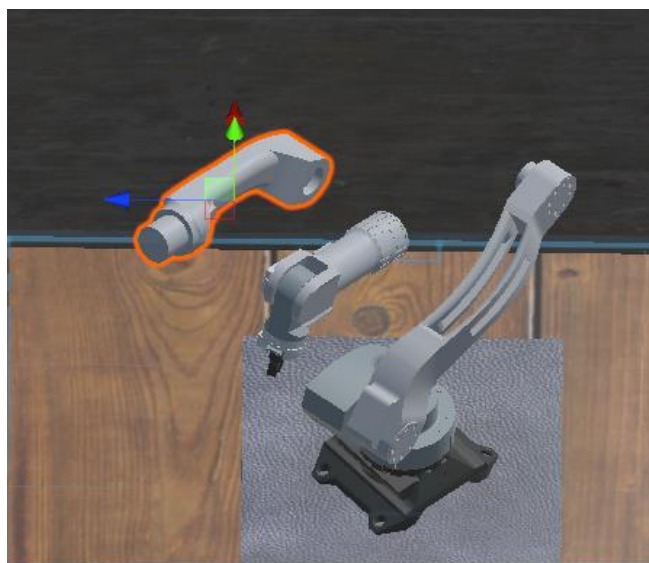
Při importování modelů bylo zjištěno, že do Unity nelze kopírovat soubory přímo z komprimovaných složek, jako jsou např. složky formátu .zip. Nejprve je nutné soubory rozbalit a až následně vložit do složek projektu.

Po aplikování textur na všechny objekty v místnosti bylo docíleno vizuálu zobrazeného na obr. 28.



Obr. 28: Pohled do scény s galerií

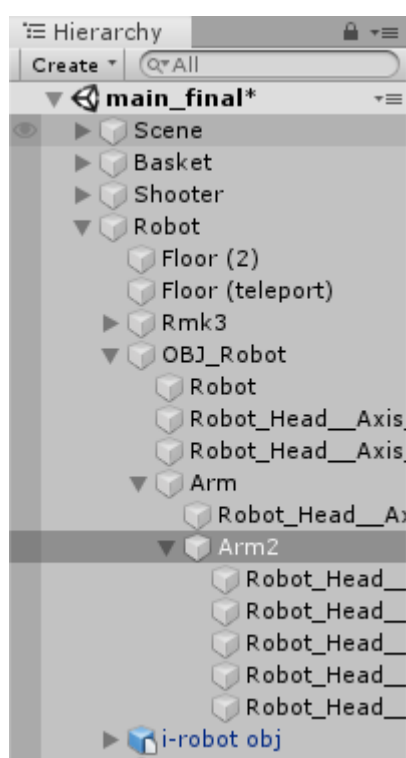
Objekty v Unity je možné rozpohybovat několika způsoby. Vše je řízeno skripty, které k objektu přiřadíme. Mohou reagovat na různé podněty anebo se hýbat naprosto mimovolně. Všechny modely importované do Unity jsou poskládány z několika součástí. Součástmi lze pohybovat nezávisle na ostatních součástech, jak je zobrazeno na obr. 29.



Obr. 29: Posunutí součásti robota

Všechny součásti robotu tedy tvoří samostatné Herní Objekty. V okně Hierarchie je robot reprezentován jako složka. Ta má podsložky představující jednotlivé části. Jednotlivé části jsou tzv. děti (anglické označení: *Children*) herního objektu Robot. Podsložky lze libovolně přesouvat.

Vytvořením nové prázdné podsložky ve složce robotu lze tedy vytvořit nový, robotu podřazený objekt. Novou podsložku lze vytvořit kliknutím pravým tlačítkem na složku robotu a vybráním možnosti *Create Empty* nebo při označení složky robotu a výběrem možnosti *Create Empty Child* v záložce Herní Objekt. Přesunutí součástí do této složky je učiní Child objekty námi vytvořeného objektu. Přidáním skriptu na nově vytvořený objekt je aplikována jeho funkce na všechny podřazené části. Vytvořením vhodné struktury (obr. 30) tak lze relativně lehce dosáhnout i složených pohybů.



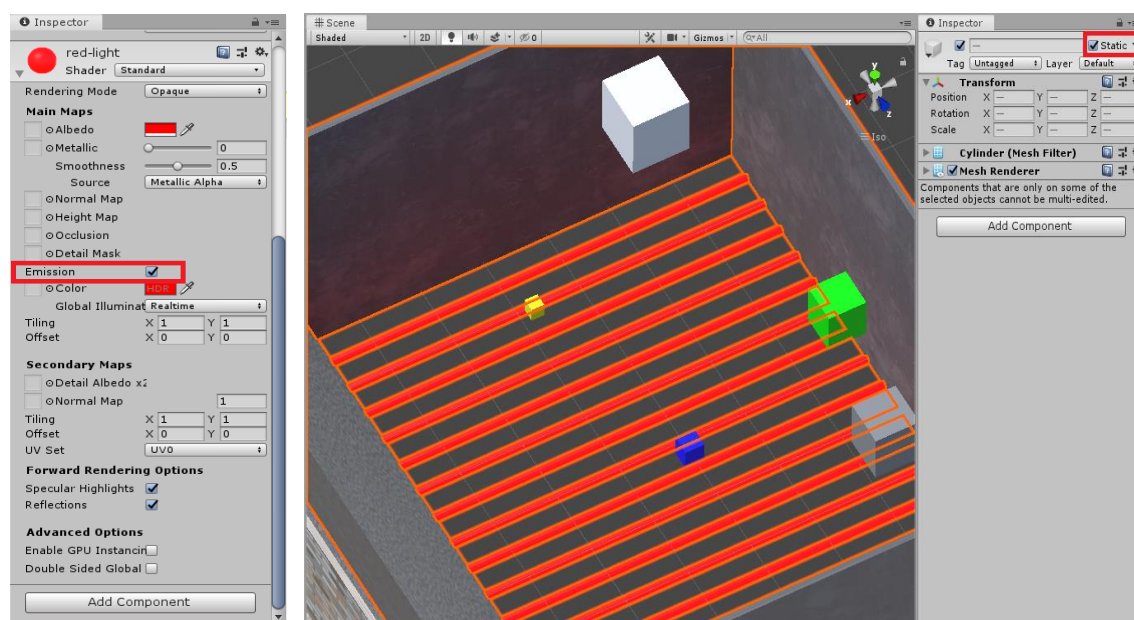
Obr. 30: Struktura Child objektů robotu

Přes svou jednoduchost scéna představuje největší potenciál nabízený VR. Hlavní potenciál spočívá v naprosto realistickém zobrazení libovolných objektů, kterého nelze dosáhnout jinou zobrazovací technikou. Importováním velice kvalitního modelu a vytvořením věrného okolního prostředí lze vyprodukovat téměř stejně kvalitní vizuální vjem, jaký by byl spatřen při pohledu na objekt ve skutečnosti.

3.4.4 Střelnice

Poslední z bočních místností je tvořena scénou se střelnicí. Obsahuje nejsložitější skripty, řídící interakce mezi objekty, ze všech zbylých scén. Scéna obsahuje model zbraně, který byl importován do projektu z [38]. Dále se zde nachází šest terčů v podobě barevných krychlí různých velikostí a pult zpoza kterého se střílí.

Podlaha je opět rozdělena, tentokrát na dvě části. Část podlahy u vchodu je stejně jako u bowlingu jediná, na kterou je umožněn teleport. Zbytek podlahy vyplňuje prostor pod terči a je tvořen „*laserovou mříží*“. Mříž byla vytvořena několika protáhlými válci. Válcům je přiřazen zářící červený materiál. Vyzařování bylo u materiálu docíleno označením políčka *Emission* ve vlastnostech materiálu (zvýrazněno na obr. 31 a). Aby zářící předmět vrhal světlo i na ostatní předměty, je nutno vygenerovat světelnou mapu. Tu Unity vygeneruje po označení zářících objektů společně s okolními objekty a jejich nastavením jako statické (označeno na obr. 31 b).



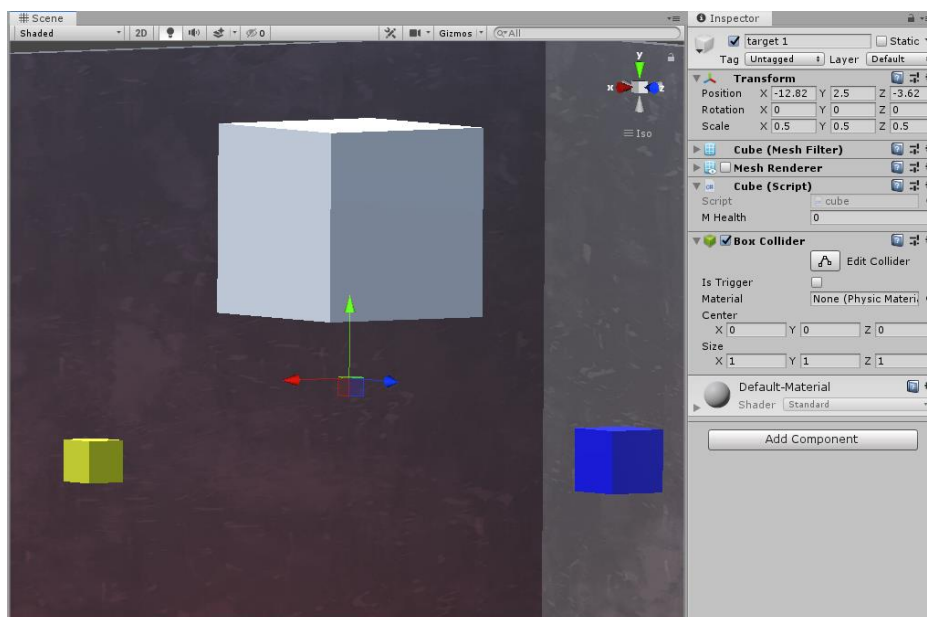
a)

b)

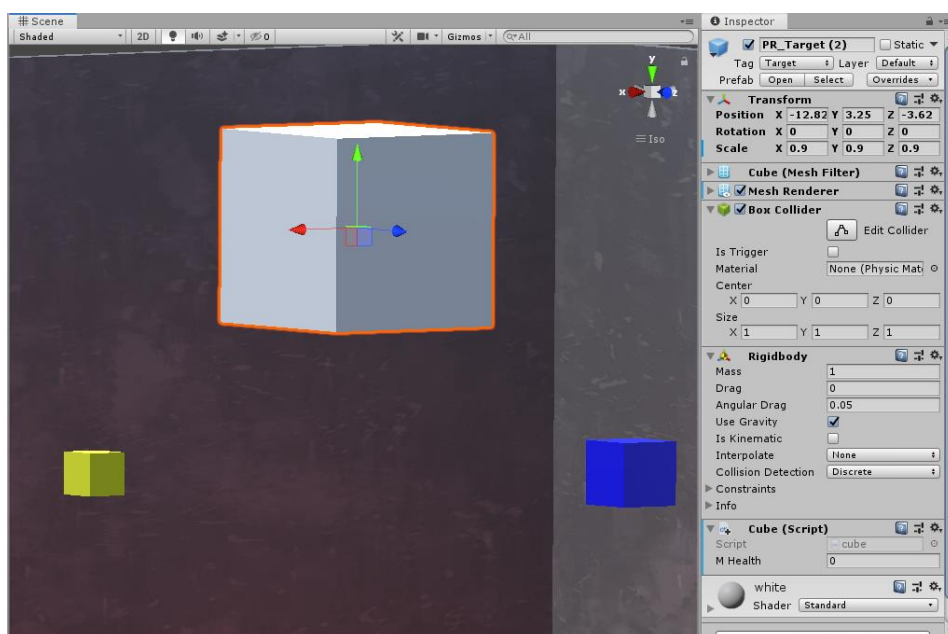
Obr. 31: a) Přiřazení emise k materiálu, b) Vygenerování světelné mapy

Terče na střelnici jsou reprezentovány šesti krychlemi. Krychle se vznášejí ve vzduchu v prostoru střelnice. Vznášení krychlí s přidáním komponentem *RigidBody*. Ve kterém byla nastavena hmotnost a zapnuté působení gravitace, bylo docíleno umístěním krychle na druhou krychli. U druhé krychle bylo vypnuto vykreslování a je tudíž ve scéně neviditelná. Tato průhledná krychle taky nemá komponent *RigidBody*, díky čemuž na ni nepůsobí gravitace. Obě krychle mají přidáný *Box Collider*, který zabraňuje, aby hmotná

krychle propadla přes neviditelnou krychli pod ní. Detail obou objektů je na obr. 32 a) a 32 b).



a)



b)

Obr. 32: a) neviditelná podpůrná krychle s vypnutým komponentem *Mesh Renderer*,
b) krychle představující cíl

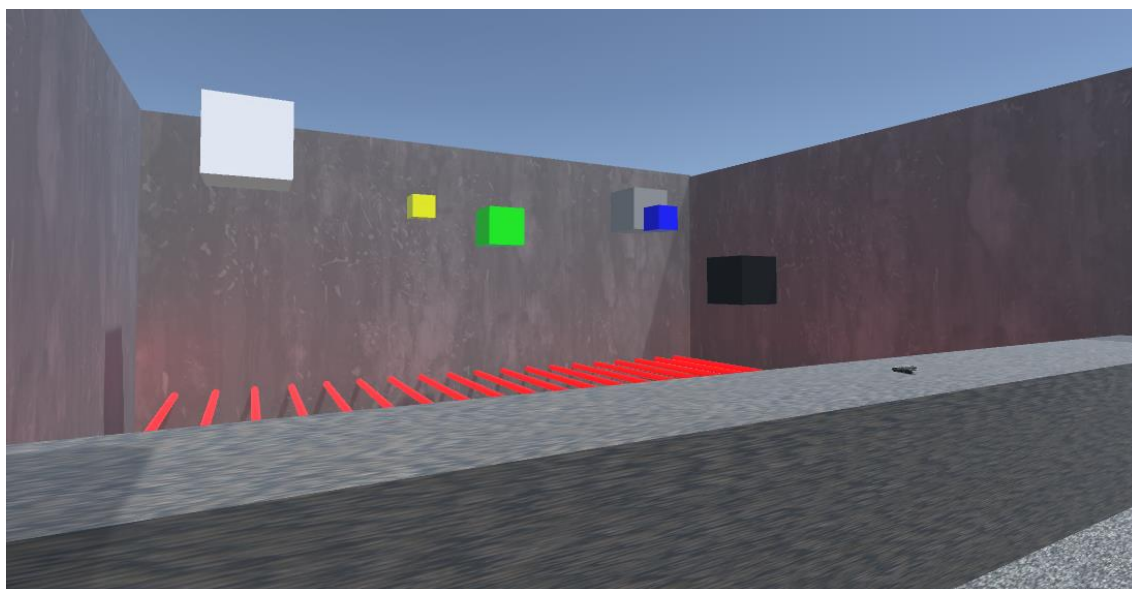
Na obr. 32 b) si lze povšimnout, že má krychle, v kolonce *Tag* umístěné v horní části okna Inspektor, nastaveno *Target*. Je to vytvořená skupina obsahující Herní objekty, na které jsou aplikovány zásahové efekty, v případě zasažení výstřelem z pistole.

Pomocí přiloženého skriptu nazvaného *Cube* (lze vidět na obr. 32 b) byl zásahový efekt definován jako krátká změna barvy krychle na červenou. Dojde též k jejímu posunutí o malý kousek v záporném směru osy x, která je kolmá na zadní stěnu místnosti. Díky tomu krychle spadne z její podpory a propadnutím přes laserovou mříž (válce tvořící mříž nemají Collider) zmizí ze scény.

Skript přidáný do komponentů zbraně definuje proměnnou uchovávající informaci o pozici a natočení ústí hlavně. Dále je zde definována hodnota poškození, které zbraň uděluje. Další část skriptu definuje funkci výstřelu. Ta je zavolána ze skriptu přidáného do komponentů ovladače, pokud dojde ke stlačení tlačítka spouště. Funkce výstřelu vygeneruje paprsek ve směru, kterým míří hlaveň pistole. Pokud tento paprsek zasáhne některý z terčů, jsou na něj aplikovány zásahové efekty.

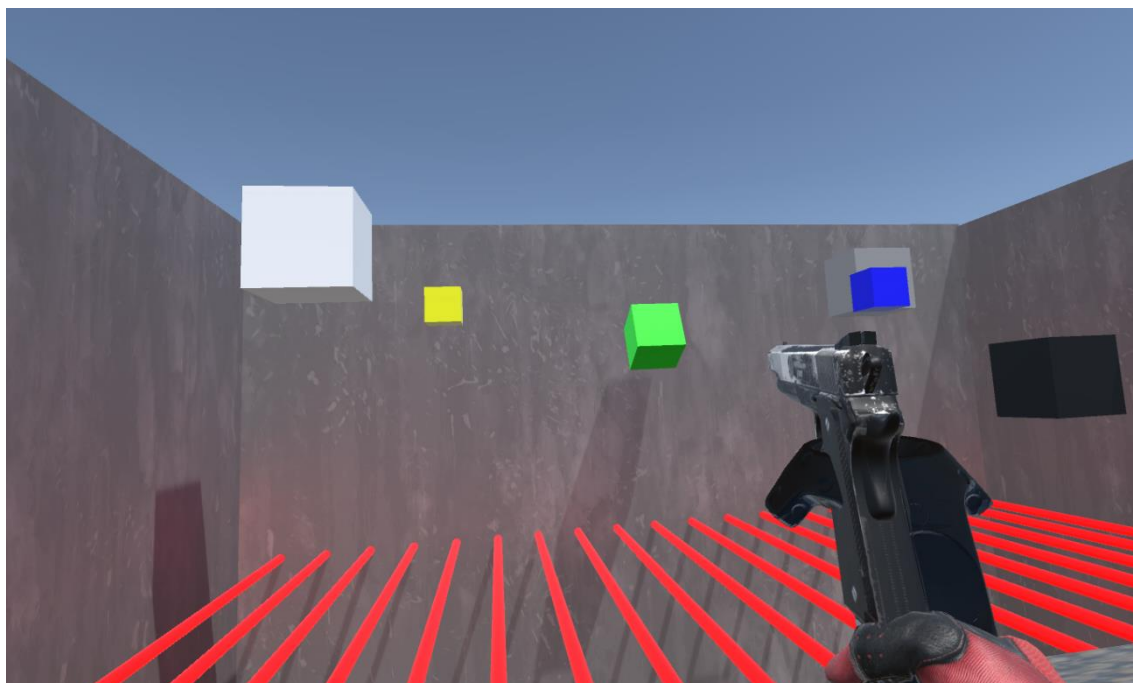
Pistol je ovládána jedním z ovladačů. Je na ovladači zafixována a pohybují se ve virtuálním prostředí společně. Zafixovaná poloha byla zvolena tak, aby se z pistole dalo střílet bez nutnosti nepřirozeného vytáčení ovladače.

Po přidání všech textur dostala scéna následující podobu vyobrazenou na obr 33.

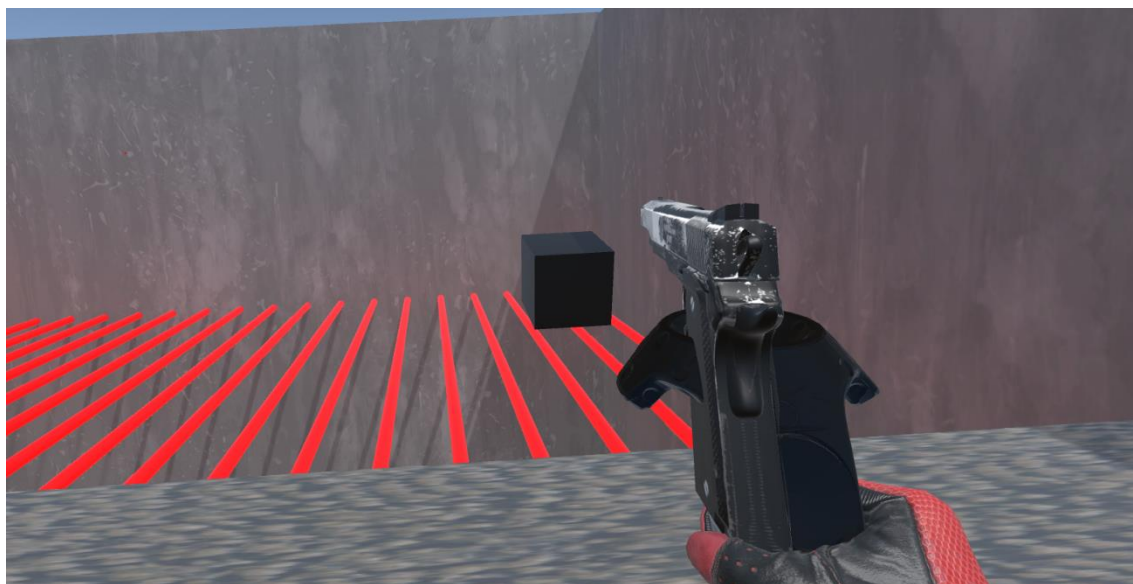


Obr. 33: Náhled do scény se střelnici

I scéna se střelnici prošla testováním při použití VR headsetu. Následují snímky obrazovky zachycených při tomto testování. Obrázky 34 a) a b) zobrazují padající terče zasažené výstřelem z pistole.



a)



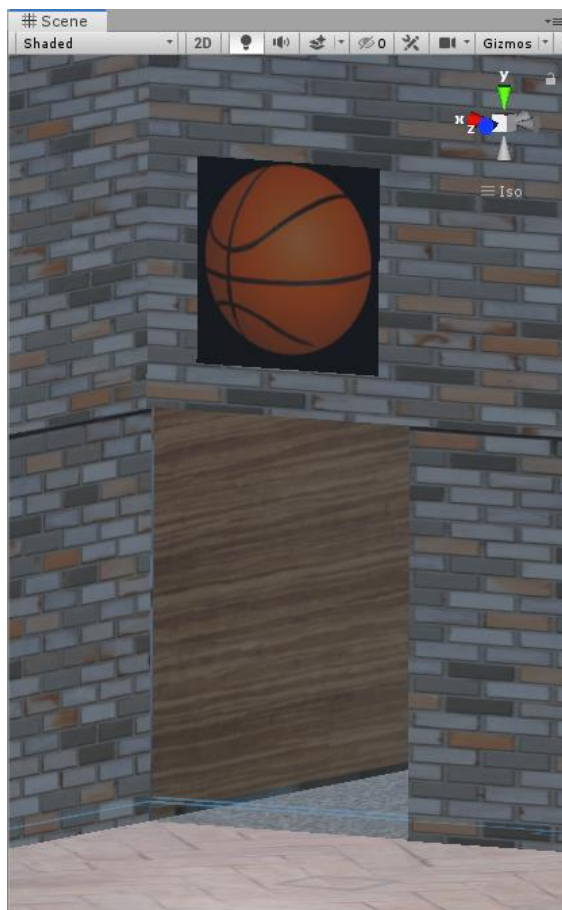
b)

Obr. 34: Snímky obrazovky pořízené při testování střelby na terče, na obrázku a) je padající zelená kostka, obrázek b) zobrazuje padající černou kostku v době, kdy všechny okolní kostky již byly shozeny

3.4.5 Spojovací místnost

Poslední z vymodelovaných místností se nachází v samotném středu celé scény. Jejím hlavním účelem je spojení všech zbývajících místností. Nachází se zde schody k vyvýšeným plošinám a stůl s pyramidou z kostek.

K zajištění lepší orientace byla nad každý vchod do postranní místnosti umístěna deska. Desky jsou tvořeny 3D objektem plocha. Na jednotlivé desky byla umístěna textura s obrázkem, představujícím účel místnosti, do níž lze daným vchodem vstoupit. Na obr. 35 je pohled na desku nad vchodem do místnosti se hrou basketbal.



Obr. 35: Deska u vstupu do místnosti se hrou basketbal

Ve spojovací místnosti se dále nacházejí schody vedoucí k vyhlídkovým plošinám (obr. 36). Schody, stejně jako plošiny, byly vytvořeny úpravou 3D objektů krychlí. Na prvním, prostředním a posledním schodu jsou umístěny *TeleportPointy*. Plošiny byly duplikovány a z jejich duplikátů byly vytvořeny teleportační plochy. Plošiny jsou umístěny zároveň s vrchním okrajem stěn, kdy nad každým vchodem do postranní místnosti se nachází jedna z nich. Díky plošinám je hráči nabídnuta možnost si jednotlivé scény prohlédnout z jiné, vyvýšené perspektivy. Na obr. 37 je znázorněn takový pohled na scénu se hrou basketbal.



Obr. 36: Schody a vyhlídkové plošiny



Obr. 37: Pohled z plošiny na scénu se hrou basketbal

V místnosti je také umístěn stůl. Na stole stojí pyramida postavená z kostek těsně naskládaných k sobě, jak lze spatřit na obr. 36. Kostky mají přidán komponent *Throwable*. Při jeho přidání jsou automaticky přidány komponenty *RigidBody*, *Interactable* a *Velocity Estimator*, které nelze následně odstranit, protože je na nich komponent *Throwable* závislý. O přidání pyramidy z kostek bylo rozhodnuto kvůli otestování problému, který se vyskytl v projektu vytvořeném pod starší verzí enginu Unity.

Jednalo se o verzi 2018.3.0f2. Při stejné postavené konfiguraci kostek došlo po uchopení ovladačem při pokusu o vytažení spodní kostky k jejich nepřírodným deformacím. Při úplném vytažení kostky ze struktury se vrchní kostky přeměnily na prkna, jak dokládá obr. 38 a) a b). Jak je vidět na obr. 39 ve verzi 2019.1.2f1 se kostky chovají zcela přirozeně a problém byl vývojáři engine Unity odstraněn.



a)



b)

Obr. 38: a) Deformující se kostky, b) Kostky přetvořené na desky



Obr. 39: Chování kostek v nové verzi programu Unity

ZÁVĚR

Cílem bakalářské práce bylo provést rešerši na téma virtuální realita a dostupných možností na tvorbu prostředí vhodného pro VR. Po prostudování zdrojů bylo dalším úkolem vymodelovat scénu pro VR, která by demonstrovala základní vlastnosti vývojového prostředí.

V teoretické části práce je VR definována, stručně shrnuta její historie a možné aplikace. Dále jsou zde uvedeny technologie, které se v souvislosti s VR používají nebo vyvíjejí. Pro vývoj VR prostředí se jeví jako nejvhodnější herní engine, kterým je věnován zbytek teoretické části práce. V této části byl pojem herní engine definován, byla popsána historie engineů a jejich nepostradatelné části. V závěru kapitoly je pak popsáno několik vybraných zástupců.

K tvorbě VR prostředí v praktické části práce byl vybrán herní engine Unity. K demonstraci vlastností engineu byly zvoleny scény s galerií, střelnicí a hrami basketbal a bowling. Dále bylo otestováno, zda přetrvává problém s chybnou prací fyzikálního engineu při vzájemném smýkání těles po sobě, který se vyskytl v jedné z předchozích verzí engineu.

K modelování prostředí byl použit výhradně modelovací systém zabudovaný uvnitř v engineu. Složitější modely byly importovány z volně přístupných zdrojů. Funkční interakce pak byly opět přidány v engineu pomocí skriptů. Bylo tak docíleno např. umísťování kuželek zpět na scénu, rozpohybování robotických ramen nebo shazování terčů výstřelem z pistole.

Vytvořené scény jsou plně funkční a vhodně demonstují zamýšlené vlastnosti engineu. Každou místnost vytvořené scény je možno v budoucnu rozvíjet a vylepšit. Zajímavým zlepšením by mohlo být přidání modelu ovladače do scény s roboty, kterým by se dal robot přímo ovládat. Hratelnost hry basketbal by zlepšilo přidání podobného skriptu na vrácení balónů po hodu na koš, jako se nachází u bowlingových koulí. Hru bowling vylepšením skriptu vrácení kuželek tak, aby byly započítávány i kuželky shozené na bok. Střelnice pak nabízí celou řadu potenciálních vylepšení přidáním nových zbraní nebo cílů.

VR i herní enginey jsou slibně se rozvíjející odvětví, která mají enormní potenciál. Pomocí herních engineů a vložením kvalitních 3D modelů z modelovacích programů, lze vytvořit realisticky vyhlížející prostředí pro VR, jak bylo předvedeno v této práci a bylo tak dosaženo stanovených cílů.

SEZNAM POUŽITÉ LITERATURY

- [1] MAZURYK, Tomasz a Michael GERVAUTZ. *Virtual Reality – History, Applications, Technology and Future*. Institute of Computer Graphics, Vienna University of Technology, Austria, 1996.
- [2] Akshay Bhardwaj, Minakshi Bhardwaj, and Anu Gaur, “Virtual reality: An overview,” *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 4, pp. 159-164, 2016.
- [3] *Kyberprostor* [online]. c2017 [citováno 23. 05. 2019]. Dostupný z: <https://wikisofia.cz/index.php?title=Kyberprostor&oldid=47279>
- [4] LINOWES, Jonathan. *Unity Virtual Reality Projects: Explore the world of virtual reality by building immersive and fun VR projects using Unity 3D*. 2nd Edition. Birmingham: Packt Publishing, 2018. ISBN 9781788477185.
- [5] LANG, Ben. Report: Microsoft Holding a HoloLens 2 Press Event Next Month. *Road to VR* [online]. 17.1.2019 [cit. 2019-05-23]. Dostupné z: <https://www.roadtovr.com/report-microsoft-holding-a-hololens-2-press-event-next-month/>.
- [6] BOAS, Y. A. G. V. Overview of virtual reality technologies. In: *Interactive Multimedia Conference*. 2013.
- [7] SHARMA, Mohit. Third Dimension Knowledge. *Blogger.com* [online]. 24.1.2011 [cit. 2019-05-23]. Dostupné z: <http://thirddimensionknowledge.blogspot.com/2011/01/father-of-virtual-reality.html>
- [8] LEWIS, Richard. Father Ivan and his Children. *Lewis art cafe* [online]. 17.6.2018 [cit. 2019-05-23]. Dostupné z: <http://lewisartcafe.com/father-ivan-and-his-children/>
- [9] EARNSHAW, R. A. *Virtual Reality Systems*. Cambridge: Academic Press, 1993. ISBN 9781483296579.
- [10] Letecký plně pohyblivý simulátor. In: *FLY FOR FUN* [online]. c2019 [cit. 2019-05-23]. Dostupné z: <http://www.flyforfun.cz/nase-sluzby/vycvik-pilotu/letecky-simulator-6/letecky-plne-pohyblivy-simulator-34/>
- [11] SETH, Abhishek, Judy M. VANCE a James H. OLIVER. Virtual reality for assembly methods prototyping: a review. *Virtual Reality* [online]. 2011, 15(1), 5-20 [cit. 2019-05-23]. DOI: 10.1007/s10055-009-0153-y. ISSN 1359-4338. Dostupné z: <http://link.springer.com/10.1007/s10055-009-0153-y>
- [12] *What is motion tracking?* [online]. Advanced Realtime Tracking spol s.r.o., 2017 [cit. 2019-05-23]. Dostupné z: <https://ar-tracking.com/technology/motion-tracking/>
- [13] ŽÁRA, Jiří, Bedřich BENEŠ, Jiří SOCHOR a Petr FELKEL. *Moderní počítačová grafika*. Druhé vydání. Brno: Computer Press, 2004. ISBN 80-251-0454-0.
- [14] 3d-Glasses PNG. In: *Searchpng* [online]. 13.1.2019 [cit. 2019-05-23]. Dostupné z: <https://www.searchpng.com/2019/01/13/3d-glasses-png-free-download/>
- [15] *Head-mounted Displays (HMDs)* [online]. Virtual Reality Society, 2017 [cit. 2019-05-23]. Dostupné z: <https://www.vrs.org.uk/virtual-reality-gear/head-mounted-displays/>
- [16] VAN CAMP, JEFFREY. REVIEW: HTC VIVE PRO. *WIRED* [online]. c2018 [cit. 2019-05-23]. Dostupné z: <https://www.wired.com/review/review-htc-vive-pro/>
- [17] *CAVE Fully Immersive Virtual Reality* [online]. Virtual Reality Society, 2017 [cit. 2019-05-23]. Dostupné z: <https://www.vrs.org.uk/virtual-reality-environments/cave.html>

- [18] *Journal of King Saud University – Computer and Information Sciences* [online]. 2015, 27(3) [cit. 2019-05-23]. ISSN 13191578. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S1319157815000439>
- [19] *CAVE Automatic Virtual Environment* [online]. Visbox, Inc., c2016 [cit. 2019-05-23]. Dostupné z: <http://www.visbox.com/products/cave/>
- [20] WARD, Jeff. *What is a Game Engine?* [online]. Informa PLC, 2008 [cit. 2019-05-23]. Dostupné z: http://www.gamecareerguide.com/features/529/what_is_a_game_.php?page=2
- [21] ENGER, Michael. *Game Engines: How do they work? GIANT BOMB* [online]. CBS Interactive Inc, 2013 [cit. 2019-05-23]. Dostupné z: <http://www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-theywork/101529/>
- [22] PAUL, P S, G. SURAJIT a A. BHATTACHARYA. History and comparative study of modern game engines. *International Journal of Advanced Computer and Mathematical Sciences* [online]. 2012, 245-249 [cit. 2019-05-23]. ISSN 2230-9624. Dostupné z: https://www.researchgate.net/publication/259496289_History_and_comparative_study_of_modern_game_engines
- [23] HAAS, John.: *A History of the Unity Game Engine* [online]. Worcester, 2013 [cit. 2019-05-23]. Dostupné z: https://web.wpi.edu/Pubs/E-project/Available/E-project-030614-143124/unrestricted/Haas_IQP_Final.pdf. An Interactive Qualifying Project. Faculty of WORCESTER POLYTECHNIC INSTITUTE
- [24] WANG, S., MAO, Z., ZENG, C., GONG, H., LI, S. and CHEN, B.: *A new method of virtual reality based on Unity3D*, 2010 18th International Conference on Geoinformatics, Beijing, 2010, pp. 1-5.
- [25] THORNE, Alan. *Game engine design and implementation: Foundations of game development*. Sudbury: Jones & Bartlett Learning, 2011. ISBN 9781449656485.
- [26] TRENHOLME, David a Shamus P. SMITH. Computer game engines for developing first-person virtual environments. *Virtual Reality* [online]. 2008, 12(3), 181-187 [cit. 2019-05-23]. DOI: 10.1007/s10055-008-0092-z. ISSN 1359-4338. Dostupné z: <http://link.springer.com/10.1007/s10055-008-0092-z>
- [27] [online]. Unity Technologies, c2019 [cit. 2019-05-23]. Dostupné z: <https://unity.com/>
- [28] ALEKSANDR. *Documentation, Unity scripting languages and you* [online]. Unity Technologies ApS, 2014 [cit. 2019-05-23]. Dostupné z: <https://blogs.unity3d.com/2014/09/03/documentation-unity-scripting-languages-and-you/>
- [29] List of Unity games. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-05-23]. Dostupné z: https://en.wikipedia.org/wiki/List_of_Unity_games
- [30] [online]. Epic Games, Inc., 2019 [cit. 2019-05-23]. Dostupné z: <https://www.unrealengine.com/en-US/>
- [31] BROWETT, Thomas. *What can you do with Blueprint?* [online]. Epic Games, Inc., 2014 [cit. 2019-05-23]. Dostupné z: <https://www.unrealengine.com/en-US/blog/what-can-you-do-with-blueprints>
- [32] List of Unreal Engine games. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-05-23]. Dostupné z: https://en.wikipedia.org/wiki/List_of_Unreal_Engine_games
- [33] [online]. Crytek GmbH., c2019 [cit. 2019-05-23]. Dostupné z: <https://www.cryengine.com/>

- [34] KLEKNER, Martin. SVĚT REALTIME RENDER ENGINŮ. *VFXcz* [online]. Crytek GmbH., 2019 [cit. 2019-05-23]. Dostupné z: <http://vizualniefekty.cz/svet-realtime-render-enginu/>
- [35] Most used Engines. *Itch.io* [online]. itch corp, 2019 [cit. 2019-05-23]. Dostupné z: <https://itch.io/game-development/engines/most-projects>
- [36] WILSON, Matthew. *Source: Crytek is sinking, wages are unpaid, talent leaving on a daily basis* [online]. Kitguru.net, 2016 [cit. 2019-05-23]. Dostupné z: <https://itch.io/game-development/engines/most-projectshttps://www.kitguru.net/gaming/matthew-wilson/source-crytek-is-sinking-wages-are-unpaid-talent-leaving-on-a-daily-basis/>
- [37] List of CryEngine games. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-05-23]. Dostupné z: https://en.wikipedia.org/wiki/List_of_CryEngine_games
- [38] *Free3D* [online]. [cit. 2019-05-23]. Dostupné z: <https://free3d.com/>
- [39] *Textures.com* [online]. 2019 [cit. 2019-05-23]. Dostupné z: <https://www.textures.com/>
- [40] *Bowling assets* [online]. Dostupné z: https://fusedvr.com/buddydrive/file/bowling_assets/

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

VR.....	Virtuální Realita (z anglického Virtual Reality)
AR.....	rozšířená realita (Augmented Reality)
OBR	obrázek
NAPŘ.....	například
TZV.....	takzvaný
ATD	a tak dále
HMD	náhlavní displej (Head Mounted Display)
CRT.....	katodová trubice (Cathode Ray Tube)
BOOM	Binocular Omni Orientation Monitor
CAD	Počítačem podporovaná Kreslení (Computer Aided Design)
DOF	Stupně volnosti (Degrees of Freedom)
LOS.....	Linie pohledu (Line of Sight)
LED.....	Světlo emitující dioda (Light Emiting Diode)
LCD	Displej z tekutých krystalů (Liquid Crystal Display)
OLED.....	Organická světlo emitující dioda (Organic Light-Emitting Diode)
FOV	Zorné pole (Field of View)
CAVE.....	Virtuální jeskyně (Cave Automatic Virtual Environment)
SDL.....	Simple DirectMedia Layer
AJ	a jiné
FPS.....	Střílečka z první osoby (First Person Shooter)