



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

PROPOJENÍ VIRTUÁLNÍ REALITY S EXTERNÍM ZDROJEM DAT

ON-LINE CONNECTION OF VIRTUAL REALITY WITH THE EXTERNAL DATA SOURCE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DANIEL KLÍČ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ KOVÁŘ, Ph.D.

BRNO 2018

Abstrakt

Bakalářská práce se zaměřuje na problematiku propojení virtuální reality s on-line externím zdrojem dat a následném návrhu aplikace v prostředí Unity a hardwarem HTC Vive.

Summary

Bachelor thesis deals with on-line connection of virtual reality with the external data source and creation of application in IDE Unity for HTC Vive hardware.

Klíčová slova

Virtuální realita, UNITY, headset, C#, SQL, HTC Vive, externí zdroj dat, IDE

Keywords

Virtual reality, UNITY, headset, C#, SQL, HTC Vive, external data source, IDE

KLÍČ, D. *Propojení virtuální reality s externím zdrojem dat*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 33 s. Vedoucí Ing. Jiří Kovář, Ph.D.

Prohlašuji, že svou bakalářskou práci na téma Propojení virtuální reality s externím zdrojem jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Daniel Klíč

Děkuji panu Ing. Jiří Kovář, Ph.D. za poskytnuté informace, věnovaný čas, ochotu a rady, které mi pomohli při tvorbě mé práce a také mojí rodině za podporu a motivaci, bez které by psaní práce bylo obtížnějším úkonem.

Daniel Klíč

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 2 |
| 2 | Virtuální realita | 3 |
| 2.1 | Historie virtuální reality | 3 |
| 2.2 | Virtuální realita v současnosti | 4 |
| 2.3 | Typy virtuální reality | 4 |
| 2.3.1 | Desktopová virtuální realita | 4 |
| 2.3.2 | Mobilní virtuální realita | 4 |
| 2.4 | Technologie používané při tvorbě virtuální reality | 5 |
| 2.5 | Zařízení pro virtuální realitu | 5 |
| 2.5.1 | Oculus Rift | 5 |
| 2.5.2 | HTC Vive | 8 |
| 2.5.3 | Sony Playstation VR | 10 |
| 2.5.4 | Samsung Gear VR | 11 |
| 3 | Prostředí Unity3D a Blender | 13 |
| 3.1 | Co je to Unity3D | 13 |
| 3.2 | Herní engine Unity | 13 |
| 3.3 | Unity Editor | 14 |
| 3.4 | Tvorba skriptů | 15 |
| 3.5 | Blender | 16 |
| 4 | Návrh aplikace | 17 |
| 4.1 | Výběr vhodného typu propojení | 17 |
| 4.1.1 | SQL Server Management Studio | 17 |
| 4.1.2 | Modul WWW | 18 |
| 4.2 | Zpracování dat | 19 |
| 4.2.1 | Jazyk SQL | 19 |
| 4.2.2 | Jazyk XML | 19 |
| 5 | Tvorba aplikace a jejích komponent | 21 |
| 5.1 | Tvorba grafů | 21 |
| 5.1.1 | Grafické zpracování | 21 |
| 5.1.2 | Popis kódu | 22 |
| 5.2 | Tvorba scény | 23 |
| 5.3 | Ovládání | 24 |
| 5.4 | Realizace propojení | 24 |
| 6 | Závěr | 27 |
| 7 | Seznam použitých zkratk a symbolů | 31 |
| 8 | Seznam příloh | 32 |

1 Úvod

Virtuální realita se v posledních letech dostala do podvědomí veřejnosti a v nemalém množství i na trh a do domácností v podobě VR headsetů. Za pomoci těchto headsetů si lidé mohou zahrát hry s pocitem, jako by sami byli účastníky dění. Takovéto hry byly i prvotním účelem pro tvorbu VR headsetů a podnětem pro jejich vývoj. Dalším využitím VR headsetů a virtuální reality se v poslední době stal strojní průmysl, kde se virtuální realita používá při návrhu různých strojů a testování jejich funkčnosti bez potřebného sestavení stroje. Problém nastává v tom, že i když jsme vlastníkem VR headsetu, tak to není vše, co potřebujeme pro správnou funkčnost. Na prvním místě je potřeba velmi výkonný počítač s výkonným grafickým procesorem a grafickou kartou a také mnoho paměti z důvodu velikosti her a aplikací pro virtuální realitu.

To nás přivádí k uvažování nad možností ukládání dat on-line, které že nejen razantně zmírní nároky na velikost paměti počítače, ale také umožní a usnadní sdílení dat mezi vývojáři při tvorbě strojů a různých mechanismů.

Tato bakalářská práce se zabývá problematikou propojení virtuální reality s externím zdrojem dat on-line IDE Unity za použití programovacího jazyka C#. V práci se čtenář prvotně seznámí obecně s virtuální realitou a poté s vybranými VR headsety od různých společností, jejich vývojem a specifikacemi. Poté bude čtenáři popsáno IDE Unity, práce s ním a tvorba aplikací a her. Dále bude v práci rozebrán samotný vývoj aplikace pro propojení VR s externím zdrojem, problematika, možná vylepšení aplikace a samotný program.

2 Virtuální realita

Tato kapitola pojednává o historii virtuální reality, která sahá už do 1. poloviny 20. století. Dále zde bude probírána virtuální realita nyní, typy virtuální reality a různé druhy moderních headsetů a jejich vlastnosti.

2.1 Historie virtuální reality

První fiktivní model, který se dá považovat za virtuální realitu, byl v krátké povídce Pygmalion's Spectacles napsané americkým sci-fi spisovatelem Stanley G. Weinbaumem roku 1935. V této povídce vytvořila jedna z postav brýle, pomocí kterých se mohl uživatel ponořit do virtuální reality, ve které byla zakomponována i simulace čichu, citu a chuti [1].

Skutečný, nikoli už fiktivní vývoj tzv. headsetů, začal až roku 1965, což je 30 let po Weinbaumově fiktivní představě, kdy Ivann Sutherland vytvořil historicky první VR headset pro vojenské využití. Za pomoci speciálního armádního softwaru a speciálních platform na řízení pohybů (motion control platform), byl tento první VR headset navržen pro tréninkové účely. V dnešní době je VR trénink standardem pro armádní odvětví, kde se používá pro simulaci např. letová cvičení, různé bojové situace atd., pro vojáky ještě předtím, než nastoupí do skutečného boje.

První komerční VR headsety se objevily až v 90. letech 20. století. Tyto VR headsety ale nesklidily velký úspěch, nýbrž nebyly pohodlné, graficky nebyly vůbec uspokojivé a byly schopny pouze dvou barev. Mezi takovéto první VR headsety se řadí Virtual Boy od společnosti Nintendo z roku 1993, jehož hry byly tvořeny pouze z černé a červené barvy, ale vzhledem k neúspěchu jich mnoho také nevyšlo [2].



Obrázek 2.1: Nintendo Virtual Boy (1993) [3]

2.2 Virtuální realita v současnosti

V současné nejsou VR headsety ničím neobvyklým, ačkoliv se jedná o velmi pokročilou technologii. Za první moderní VR headset lze považovat Oculus Rift od firmy Oculus VR, který se na trhu objevil roku 2014, ale dal o sobě znát už roku 2012, kdy společnost založila svou stránku na Kickstarteru, aby získala dostatečný kapitál na čtenější výrobu těchto VR headsetů.

Později roku 2015 se firma Oculus spojila se společností Samsung a podílela se na vývoji Samsung Gear VR. Dá se říct, že se VR headsety staly fenoménem a vznikla i výrazná konkurence mezi společnostmi, které tyto headsety vyrábějí. Za zmínku stojí HTC Vive, který je velkým konkurentem zmíněnému Oculus Rift a dále také Playstation VR.

S přibývajícím počtem novějších VR headsetů, které se předhánějí kvalitou displejů, přibývajícím počtem různých aplikací a lepšími možnostmi napájení se zvyšuje míra využití VR headsetů. Dále lze také stojí za zmínku použití různých pohybových senzorů rozmístěných různě po těle pro ještě větší zážitek a širší škálu využití [2].

2.3 Typy virtuální reality

Virtuální realitu lze rozdělit na dva druhy. Každý druh je specifický svými vlastnostmi a typem použití.

2.3.1 Desktopová virtuální realita

U desktopové VR slouží headset jako periferní zařízení zatímco velmi výkonný počítač zpracovává náročnou grafiku, která je k správné funkci headsetu zapotřebí. Nejčastěji je headset propojen s počítačem či herní konzolí kabely. Naše aplikace či hra poté běží na vzdáleném zařízení a headset je naším periferním displejem, které také snímá naše pohyby hlavy a celkovou pozici. Příkladem takovýchto zařízení jsou Oculus Rift, Playstation VR, HTC Vive a jiné [4].

2.3.2 Mobilní virtuální realita

Pod pojmem mobilní VR si lze představit Google Cardboard, tudíž naším periferním zařízením i výpočetní zařízením na zpracování dat je mobilní telefon, který se uloží do speciálního pouzdra. Display telefonu promítá dvojitý stereografický obraz. Dále zařízená pomocí svých senzorů snímá pohyby hlavy, ale ne celkovou pozici. Nevýhodou oproti desktopové VR je, že procesor mobilního zařízení není tolik výkonný, tudíž nemůžeme čekat stejně kvalitní výstup a četnost možností využití. Zde je příkladem zmíněný Google Cardboard, Samsung gear VR či Zeiss VR One [4].

2.4 Technologie používané při tvorbě virtuální reality

Při tvorbě zařízení pro virtuální realitu se setkáme s těmito technologiemi a pojmy:

- Stereoskopický displej: Známý také jako 3D displej nebo „head mounted displej“. Tyto displeje používají kombinaci mnoha obrázků, realistických optických zkruslení, a speciálních čoček, aby vytvořili stereofonní obraz, který naše oči interpretují jako třídimenzionální hloubku.
- Hardware na snímání pohybu: Gyroskopy, akcelerometry a jiné nenákladné komponenty jsou používány k tvorbě VR headsetů, aby snímaly pohyby těla a natočení hlavy a data z těchto pohybů poslali do aplikace, která je poté zpracovává a nakládá s nimi.
- Vstupní zařízení: Virtuální realita vytváří potřebu pro vývoj nových typů zařízení, jako jsou zařízení na snímání pohybů těla či rukou, nové gamepady či klávesnice.
- Desktopové a mobilní platformy: Tento pojem obsahuje počítačový hardware, operační systémy, software s určitým prostředím pro zařízení, engine a framework pro spuštění a funkci aplikace a softwarové nástroje pro jejich vytvoření.

„Bez těchto čtyř komponentů je těžké získat zážitek z virtuální reality, který nás zcela pohltí.“ (Parisi 2015, s. 9) [5]

2.5 Zařízení pro virtuální realitu

V této kapitole budou blíže popsány a specifikovány vybrané zařízení pro virtuální realitu.

2.5.1 Oculus Rift

Oculus Rift od firmy Oculus VR byl prvním volně dostupným VR zařízením, které se dostalo do podvědomí širší veřejnosti. Tomuto také přispěla kampaň na stránkách Kickstarter. Firmu si povšimnul zakladatel Facebooku Mark Zuckerberg, který společnost Oculus VR koupil za ohromnou částku dvou miliard korun roku a nastartoval tak pořádný vývoj tohoto headsetu [6][7].

Model DK1

Originální Oculus Rift DK1, což je zkratka pro Developer Kit, byl velkou věcí, protože oproti konkurenci představoval obrovský pokrok. Napříč tomu, že byl lepší než konkurenční přístroje, Oculus Rift DK1 nebyl příliš komfortní a k dnešním poměrům měl malé rozlišení 1280x800 pixelů. Další problém nastával při používání, kdy uživatelé často zažívali nevolnosti (motion sickness) i navzdory poměrně rychlému snímání natočení hlavy (head trackingu) okolo 250Hz. U tohoto modelu bylo možné pouze sledování okolí, protože tento model neobsahoval senzor na snímání polohy hlavy, ale pouze na natáčení v třech směrech (osách x, y, z). Zorné pole tohoto modelu bylo přes 90°, což je velmi důležité pro přiblížení požitku reality. Tento model roku 2012, po úspěšném prodeji všech headsetů, nastartoval firmu Oculus VR a podnítil tak vývoj další verze Oculus Rift DK2 [5].

2.5. ZAŘÍZENÍ PRO VIRTUÁLNÍ REALITU



Obrázek 2.2: Originální Oculus Rift DK1 (Developer Kit 1) [8]

| Parametr | Specifikace |
|------------------------|--|
| Váha | 380 g |
| Zorné pole | 110° |
| Rozlišení | 1280x800px |
| Odezva snímání polohy | 2 ms |
| Snímání polohy | 3 stupně volnosti (3 degrees of freedom) |
| Perzistence | 3 ms |
| Zpoždění | 50-60 ms |
| Obnovovací frekvence | 60 Hz |
| Aktualizační frekvence | Rotační: 1000 Hz |
| Display | LCD 7 palců |
| Senzory | Gyroskop, Akcelerometr, Magnetometr |
| Propojení | USB. HDMI |

Tabulka 2.1: tabulka specifikací modelu DK1

Model DK2

Na jaře roku 2014 firma Oculus Rift vydala svůj druhý model headsetu Oculus Rift DK2. Oproti prvnímu modelu DK1 je tento model svým vzhledem elegantnější. Pořád je poměrně větších rozměrů, ale je lehčí. Dále má oproti svému předchůdci lepší rozlišení 1920x1080 pixelů, namísto LED displeje je displej nahrazen technologií OLED, kvůli rozmazávání obrazu při pohybu (motion blur) a obsahuje sensor na sledování polohy hlavy (positional head tracking), což u předchozího modelu chybělo. Tato technologie nejen umožňuje uživateli sledování okolí natáčením hlavy, ale také umožňuje pohyb po virtuální scéně. Nicméně k tomuto senzoru bylo zapotřebí mít před sebou umístěnou speciální kameru, která tuto polohu snímala, což znamenalo potřebu být v blízké vzdálenosti od počítače a kamery. Dalším problémem byla nutnost směřování headsetu (facing) přímo na kameru, aby tato technologie fungovala [5].



Obrázek 2.3: Originální Oculus Rift DK2 (Developer Kit 2) s kamerou pro snímání pohybů [9]

| Parametr | Specifikace |
|------------------------|---|
| Váha | 440 g |
| Zorné pole | 100° |
| Rozlišení | 1920x1080 |
| Odezva snímání polohy | 2 ms, 3 ms, plná |
| Snímání polohy | 6 stupňů volnosti (6 degrees of freedom) |
| Perzistence | 3 ms |
| Zpoždění | 50-60 ms |
| Obnovovací frekvence | 75 Hz, 72 Hz, 60 Hz |
| Aktualizační frekvence | Rotační: 1000 Hz Poziční: 60 Hz |
| Display | OLED 5,7 palců |
| Senzory | Gyroskop, Akcelerometr, Magnetometr |
| Propojení | USB. HDMI |
| Poziční kamera | Samostatná kamera Near Infrared CMOS Sensor |

Tabulka 2.2: tabulka specifikací modelu DK2

Prototyp „Crescent Bay“

Třetí hlavní vývojovou verzí Oculus Rift je přístroj s přezdívkou Crescent Bay. Tato verze má mnoho lepších vlastností než předchozí verze DK2 a také udělala velký pokrok v designu. Display má ještě větší rozlišení a model obsahuje 360° snímání hlavy, což odstranilo potřebu přídatné kamery na snímání polohy hlavy. Dále jsou zde také poziční senzory (positional head tracking) umístěny na zadní straně headsetu [5].

Model CV1

Z prvopočátku byl Oculus Rift pouze headset bez dalšího hardwaru, ale vzhledem ke konkurenci a konkurenčnímu HTC Vive, které ke svému headsetu vyvinulo také ovladače, tak firma Oculus VR přidala také ovladače nazývané Oculus Touch. Tyto ovladače poté byly součástí první komerční verze Oculus Rift CV1, která se dostala na trh v roce 2016 a je do teď nejnovější verzí [5].

2.5. ZAŘÍZENÍ PRO VIRTUÁLNÍ REALITU



Obrázek 2.4: Sestava setu Oculus Rift CV1 s ovladači Oculus Touch [10][11]

| Parametr | Specifikace |
|------------------------|--|
| Váha | 360 g |
| Zorné pole | 110° |
| Rozlišení | 2160x1200px |
| Snímací rozmezí | 100°H x 70°V |
| Snímání polohy | 6 stupňů volnosti (6 degrees of freedom) |
| Snímání pozice | Oculus Sensor |
| Perzistence | Nízká |
| Obnovovací frekvence | 90 Hz |
| Aktualizační frekvence | Rotační: 1000 Hz Poziční: 60 Hz |
| Display | Dual OLED Panel |
| Senzory | Gyroskop, Akcelerometr, Magnetometr |
| Propojení | USB 2.0, USB 3.0, HDMI |

Tabulka 2.3: tabulka specifikací modelu CV1

2.5.2 HTC Vive

HTC Vive je hlavním konkurentem Oculus Rift a vznikl ve spolupráci výrobce mobilních telefonů HTC a softwarové společnosti Valve, která vytvořila herní klient Steam. Tento VR headset byl představen v roce 2015 a sklídl úspěch hlavně tím, že nabízel nejen headset, ale také pár speciálních ovladačů, které v té době Oculus Rift neměl [12].



Obrázek 2.5: Headset HTC Vive [13]

Vývoj HTC Vive

Do vývoje se jako první pustila skupina vývojářů z Valve roku 2012, kdy vytvořili prototyp systému s jednoduchým displejem upevněným k hlavě (head-mounted display), kamerou a QR kódy (AprilTags) na určování pozice hlavy (natočení). Během testů Valve přišlo na to, že musejí použít nízkoperzistentní displej, vzhledem k záměrům používat headset na hraní her. Při dalším testování se Valve rozhodlo zaměřit na volnost pohybu při ovládání ve virtuální realitě a vymyslelo koncept nazvaný Room-Scale VR.

Na začátku roku 2013 se k společnosti Valve připojilo HTC, protože vidělo potenciál ve VR a v roce 2014 podepsalo s Valve smlouvu o spolupráci. Poté se rozjel vývoj Developer Edition. Prvním krokem byl přechod z metody AprilTags na metodu laser-tracking. Tato metoda je velmi přesná, ale není bezchybná. Je založena na systému velmi rychle rotující základně, která vyzařuje laserový paprsek do prostoru pokoje a blikajících LED diod, kde pozice je poté počítána na základě odezvy mezi laserem a bliknutím diody. Pro správnou funkčnost je proto headset pokryt velkým množstvím senzorů nasměrovaných do mnoha směru.

Další nedílnou součástí jsou speciálně vyvinuté ovladače, které se inspirovaly ze dvou produktů: Playstation Move od společnosti Sony a ovladače Steam od samotného Valve. Do ovladače bylo nutné také zakomponovat sensory na snímání laseru (laser tracking sensor), pro určování polohy [12].



Obrázek 2.6: Ovladače pro HTC Vive [14]

| Parametr | Specifikace |
|------------------------|---|
| Váha | 468 g (model 2017) |
| Zorné pole | 110° |
| Rozlišení | 2160x1200px |
| Snímací rozmezí | 120°H x 120°V |
| Snímání polohy | 6 stupňů volnosti (6 degrees of freedom) |
| Snímání pozice | Základna (Base Station) |
| Perzistence | Nízká |
| Obnovovací frekvence | 90 Hz |
| Aktualizační frekvence | Rotační: 1000 Hz Poziční: 60 Hz |
| Display | Dual OLED Panel |
| Senzory | Gyroskop, Akcelerometr, Magnetometr, Sensor na snímání laseru, kamera |
| Propojení | USB 2.0, USB 3.0, HDMI |

Tabulka 2.4: tabulka specifikací headsetu HTC Vive

2.5. ZAŘÍZENÍ PRO VIRTUÁLNÍ REALITU

2.5.3 Sony Playstation VR

Tento headset byl vytvořen společností Sony Interactive Entertainment pro herní konzoli Sony Playstation 4/ Sony Playstation 4 Pro, pro zlepšení herních zážitku při hraní her.



Obrázek 2.7: Playstation VR headset [15]

Vývoj PlayStation VR

První myšlenka na VR headset vznikla už při vývoji Playstation Move v roce 2009. Tento ovladač se objevil na trhu v roce 2010 a sloužil k hraní speciálně navržených her, které byly ovládány pohyby těla, kdy hráč držel ovladače v rukou a ty byly snímány kamerou Playstation Eye a tím byla získávána informace o poloze.

Prvotní prototypy VR headsetů byly vyvíjeny od roku 2012, ale vývoj samostatného Playstation VR byl zahájen v roce 2014 pod krycím názvem Project Morfeus . Samotný název Playstation VR byl headsetu udělen v roce 2015 a do dnešní doby nebyl změněn. Konečná verze Playstation VR vyšla na trh v druhé polovině roku 2016 a sklídila obrovský úspěch. K ovládání headsetu je možno použít 3 zařízení: Ovladač Playstation Move, DualShock 4 a Playstation Aim Controller. DualShock 4 je typický bezdrátový gamepad vyvinut pro herní konzole Playstation 4. Je vybaven zabudovaným reproduktorem, touchpadem, akcelerometrem a gyroskopem pro snímání pohybu a možným ovládním nakláněním samotného ovladače.

Playstation Aim Controller je ovladač ve tvaru zbraně, který pracuje na podobné bázi jako ovladač Playstation Move a také na principu ovladače DualShock 4, na kterém je také založen. Společnost Sony vyvíjí dále nové možnosti ovládním např. pomocí pohybu oka (Eye Tracking) a také hledají možnosti, jak zapojit při hraní her celé tělo pro ovládním ve virtuální realitě. Také se budou snažit zjednodušit a zpříjemnit používání samotného headsetu odlehčením a úpravou na bezdrátovou verzi [16][17].



©2016 Sony Interactive Entertainment Inc. All rights reserved.
Design and specifications are subject to change without notice.

Obrázek 2.8: Ovladač Playstation Aim Controller [18]

| Parametr | Specifikace |
|------------------------|---|
| Váha | 610 g (model 2017) |
| Zorné pole | 100° |
| Rozlišení | 1920x1080px |
| Snímací rozmezí | 75°H x 45°V |
| Snímání polohy | 6 stupňů volnosti (6 degrees of freedom) |
| Snímání pozice | Playstation Camera a 9 LED diod na displeji |
| Perzistence | Nízká |
| Obnovovací frekvence | 120 Hz, 90 Hz |
| Aktualizační frekvence | Rotační: 1000 Hz Poziční: 60 Hz |
| Display | OLED displej 5,7 palce |
| Senzory | Gyroskop, Akcelerometr |
| Propojení | USB 2.0, USB 3.0, HDMI |

Tabulka 2.5: tabulka specifikací Playstation VR

2.5.4 Samsung Gear VR

Samsung Gear VR headset slouží k propojení s mobilním telefonem a spadá do odvětví mobilní virtuální reality. Byl vyvinut společností Samsung za pomoci společnosti Oculus VR, jako odpověď na velký vývoj desktopové virtuální reality.

Vývoj Samsung Gear VR

Společnost začala mluvit o VR headsetem už v roce 2005, kdy si nechal patentovat headset tvaru písmena V (clamshell headset). Tvar písmena V nebyl náhodou, protože již v tomhle roce, kdy byla zrovna vlna mobilních telefonů tzv. Věček, společnost pomýšlela na použití mobilního telefonu jako součástí headsetu. V tomto roce byl ale problémem nejen displej, ale také výkon tehdejších telefonů.

Samsung dále pokračoval ve vývoji a první pozitivnější výsledky přišli až v roce 2013, kdy také vyšla řada mobilních telefonů Galaxy S4. Bohužel i u tohoto modelu nebyla stále dostačující kvalita displeje, ale Samsung se poučil v jiných stránkách věci: rychlosti a přesnosti snímání natáčení hlavy, GPU optimalizací a zdokonalení displeje se super rychlými reakcemi. Poté se Samsung zaměřil na vývoj mobilního telefonu splňujícího jejich náročné požadavky na kvalitu displeje a vznikl Galaxy Note 4 s QHD displejem. Vytvoření tohoto mobilního telefonu a spolupráci s firmou Oculus vedlo k první verzi komerčně prodejného headsetu roku 2014 [5].



Obrázek 2.9: Samsung Gear VR (2017) s ovladačem [19]

2.5. ZAŘÍZENÍ PRO VIRTUÁLNÍ REALITU

| Verze | Popis |
|---------|---|
| SM-R320 | První komerční verze z roku 2014 kompatibilní s telefony Samsung Galaxy Note 4. |
| SM-R321 | První komerční verze kompatibilní z roku 2016 s telefony Samsung Galaxy S6 a S6 Edge. |
| SM-R322 | První komerční verze kompatibilní z roku 2015 s telefony Samsung Galaxy S6, S6 Edge, S6 Edge +, Galaxy Note 5, S7 a S7 Edge. |
| SM-R323 | První komerční verze kompatibilní z roku 2016 s telefony Samsung Galaxy S6, S6 Edge, S6 Edge +, Galaxy Note 5, S7 a S7 Edge, Galaxy Note 7. |
| SM-R324 | První komerční verze kompatibilní z roku 2017 s telefony Samsung Galaxy S6, S6 Edge, S6 Edge +, Galaxy Note 5, S7 a S7 Edge, Galaxy S8, Galaxy S8 +. Verze obsahuje i speciální ovladač. |
| SM-R325 | První komerční verze kompatibilní z roku 2017 s telefony Samsung Galaxy S6, S6 Edge, S6 Edge +, Galaxy Note 5, S7 a S7 Edge, Galaxy S8, Galaxy S8 +, Galaxy Note 8. Verze obsahuje i speciální ovladač. |

Tabulka 2.6: Seznam modelů Samsung Gear VR a podporované mobilní telefony

Specifikace modelu

Specifikace modelu závisí na použitém mobilním telefonu od společnosti Samsung a na jeho specifikacích.

3 Prostředí Unity3D a Blender

Tato kapitola pojednává o herním engine Unity a uživatelském prostředí, pro tvorbu aplikací a her. Bude zde popsány jednotlivé verze engine a také jednotlivé části uživatelského rozhraní, potřebné pro orientaci v prostředí.

3.1 Co je to Unity3D

Unity3D je výkonný multiplatformní herní 3D engine a uživatelsky přátelské vývojové prostředí vyvinuté společností Unity Technologies. Pro začátečníky je velmi jednoduché na naučení a pro experty je velice výkonné a splňuje většinu náročných požadavků na tvorbu především her, ale také 2D a 3D aplikací na různé platformy.

3.2 Herní engine Unity

Původní účel pro vytvoření engine Unity byla tvorba mobilních a webových her. Engine vznikl v roce 2005 pro tvorbu her na platformu OS X. Tato verze 1.0 sklídila velký úspěch, a to také díky tomu, že byla jak placená, tak i bezplatná verze, tudíž podporovala tvorbu pro osobní používání aplikací. Společnost vyvíjet verzi 1.1, která také podporovala nejen zmiňovaný OS X, ale také Microsoft Windows.

Unity 2.0 bylo zaměřeno ještě více na tvorbu v Microsoft Windows a na vylepšení kompatibility webových prohlížečů napříč všemi platformami. Tohoto bylo dosaženo za pomoci přidání podpory Microsoft DirectX. Další hlavní verzí byla verze Unity 3.0, který vyšla v roce 2010, představující low lever debugging, možnost zastavení renderování objektů mimo kameru, lightmapping a také novinku nazvanou Assets Store, což je digitální obchod, ve kterém si vývojáři mohli buď zakoupit nebo zdarma stáhnout různé modely, nastavení pro aplikace a mnoho dalšího.

Verze Unity 4.0 přinesla uživatelům tvorbu i 2D her a aplikací, která v předešlých verzích nebyla podporována. Dále přinesla Mecanim animace, Shuriken částicový systém a podporu Adobe Flash a Linux. Unity 5.0 přinesla podporu WebGL, Real-time Global Illumination osvětlování, Physical Shaders engine, vylepšení odrazu obrazů od předmětů, nový Physics Engine, klient na tvorbu animací, Audio Mixer na mixování skladeb a zvuků a 64-bitový editor.

Nejnovější verzí je Unity 2017, které přineslo nové nástroje pro tvorbu designu, Timeline časový editor, lepší Post-processing, vylepšení 2D tvorby, vylepšení samotného editoru, podpory Visual Studio, použití nejnovější verze C#6 & .NET 4.6, podporu PS4, NVIDIA VRWorks a Universal Windows Platform [20].

3.3 Unity Editor

V současnosti jde o nejpoužívanější nástroj pro tvorbu her a aplikací na různé platformy. Tento editor nabízí jednoduché a uživatelsky přívětivé prostředí se silnou komunitou. I když je především kladen důraz na začínající vývojáře, neznamená to, že by Unity3D nebylo plnohodnotným profesionálním vývojovým prostředím, což dokazuje skutečnost, že jej používají pro vývoj i velké firmy, které se zabývají tvorbou her.

Celé rozhraní je rozděleno do základních částí: okna projektu, okna scény, okna hierarchie, oknem inspektoru a Toolbarem (oknem s nástroji). Po sestavení hry či aplikace se poté musí projekt „Sestavit“ v nastavení „Build Settings“, kde si vybereme platformu, na které chceme aplikaci či hru používat a sestavíme ji [21].

Pro práci na této práci byl původně zvolen Unity3D ve verzi 2017.4.0f2 jelikož se jedná o poslední verzi se všemi novými vylepšeními.

Okno projektu

V tomto okně máme přístup k nastavením, použitým objektům a skriptům, které náleží do projektu. Nalevo máme panel se strukturou složek projektu. Po rozkliknutí složky se nám zobrazí v pravém panelu obsah složky jako ikony, které reprezentují svůj typ souboru. Pod seznamem struktury projektu najdeme sekci Oblíbené, kde můžeme ukládat své často používané objekty, pro lepší přístup k nim samotným.

Podél horní hrany celého okna je vyhledávací okénko pro snazší vyhledávání a nalevo tohohle pole je tlačítko Vytvoř, které nám umožňuje vložit nové objekty či podsložky. Při vyhledávání si můžeme také nastavit, jaké typy souborů přesně hledáme (skripty, objekty, hudební soubory...).

Okno scény

Okno scény je interaktivní náhled do světa, který zrovna vytváříte. V tomto okně se určují pozice postav, kamer, světla a dalších herních objektů. Toto okno se dá také přepnout na Herní okno, které nám ukazuje, jak hra či aplikace ve výsledku vypadá [22].

Okno hierarchie

Okno hierarchie obsahuje seznam každého herního objektu GameObject ve zvolené scéně. Některé herní objekty jsou pouze pro určitou scénu a některé jsou tzv. Prefab objekty, které tvoří většinu hry a nemízí mezi jednotlivými scénami. Tyto Prefab objekty poznáme modrým zabarvením textu. Objekty v tomto okně jsou řazeny na „rodičovské“ a „dětské“ objekty, jak je zvykem v objektovém programování [23].

Okno inspektoru

Projekty v Unity Editoru jsou tvořeny z mnoha herních objektů (GameObjects), které obsahují skripty, zvuky a další grafické elementy. Okno inspektoru zobrazuje detailní informace o těchto objektech a jejich celém obsahu. V tomto okně tyto atributy objektů můžeme různě upravovat či tvořit [24].

Toolbar

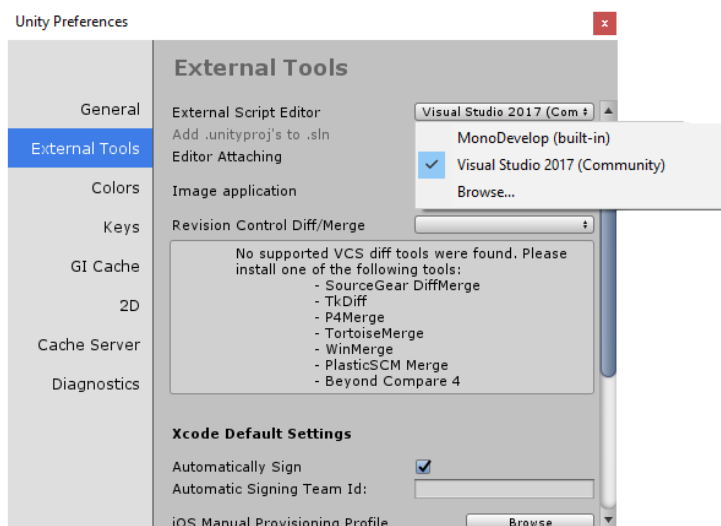
Toolbar jak už název napovídá obsahuje několik základních ovládaní, které jsou zapotřebí pro editaci a tvorbu. Je rozdělen na 7 částí: Transformační nástroje, transformace okna scény, Play/Pause/Step tlačítko pro ovládaní v herním okně, tlačítko Cloudu, tlačítko Účet, pro nastavování účtu, tlačítko vrstev a tlačítko rozložení [25].

Assets Store

Unity Asset Store je název pro obchod s různými herními doplňky a jeho úkolem je usnadnit práci vývojářům, kteří jsou například zbláhli ve skriptování, ale ne designu, anebo naopak. Jsou zde doplňky jak placené, tak i zdarma. Vše je tu přehledně rozděleno a nechybí i recenze ostatních uživatelů a jejich hodnocení.

3.4 Tvorba skriptů

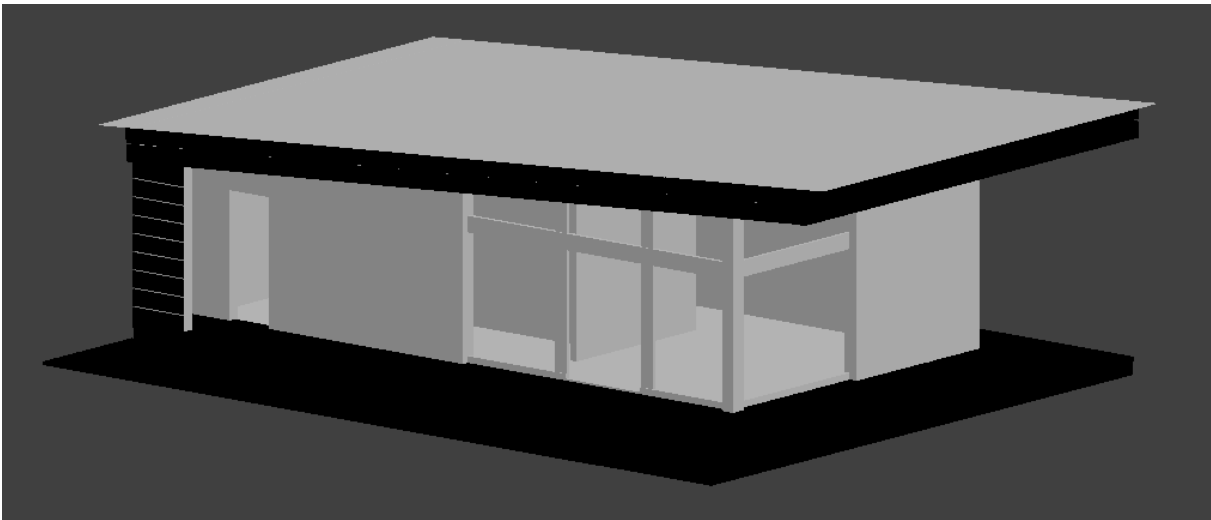
Skripty se v Unity používají k určování vlastností či chování určitých objektů, ke kterým se dají přiřazovat stejně, jako ostatní komponenty přetažením na objekt. Skripty jsou nedílnou součástí tvorby jakékoliv aplikace v Unity. Unity nabízí na výběr ze dvou programovacích jazyků, ve kterých lze skript psát. Těmito jazyky jsou Javascript a C#. Oba dva jazyky jsou tzv. objektově orientované jazyky. Pro každý z těchto jazyků lze na příslušných stránkách najít dokumentaci pro jednotlivé funkce a vlastnosti z knihoven. Pro tvorbu skriptů si lze vybrat ze dvou editorů: MonoDevelop, Microsoft Visual Studio. Microsoft Visual Studio již obsahuje volně dostupný plugin pro propojení s Unity. Výhodou používání Visual Studia je jeho funkce IntelliSense a navigace kódu pro C#, což zrychluje samotnou tvorbu kódu. MonoDevelop od firmy Xamarin je volně dostupné vývojové prostředí, které je kompatibilní s 11 programovacími jazyky a pro programování využívá Mono a .NET framework. MonoDevelop je také propojen s Unity [26].



Obrázek 3.1: Výběr editoru pro skriptování

3.5 Blender

Blender je open source uživatelské rozhraní vytvořené společností Blender Foundation a je určený především pro tvorbu 3D počítačové grafiky a animací. Lze v něm modelovat, vytvářet sítě k tvorbě charakterů, animace, renderovat a také editovat videa. Pro pokročilejší uživatele je možnost vytvořit si vlastní verzi aplikace či nástrojů pomocí jazyka Python. Blender je takzvaná cross-platform aplikace, což znamená, že funguje na více platformách (Windows, Linux, Macintosh), k čemuž používá OpenGL grafickou knihovnu. Pro modelování v Blenderu je použita metoda vynášení bodů a jejich spojování, tudíž modelované objekty jsou reprezentovány jako takto vzniklé plochy. Nespornou výhodou Blenderu je možnost přizpůsobit si uživatelské prostředí podle vlastních představ, což napomáhá k lepší orientaci uživatele v samotném prostředí programu. Editor Blender byl v práci použit na tvorbu několika modelů a různé vizuální úpravy. Hlavním vytvořeným modelem za pomoci tohoto nástroje byl model haly, který byl použit pro tvorbu důvěryhodného prostředí [27].



Obrázek 3.2: Model domu v prostředí Blender

4 Návrh aplikace

V této kapitole budete seznámeni se samotným procesem návrhu řešení propojení se serverem, jednotlivými možnostmi propojení, typy použitých dat a realizací. První část se zabývá výběrem možného propojení se serverem dle požadavků a popisem těchto možností. Ve druhé části jsou popsány typy použitých jazyků pro získávání dat a principy používání těchto jazyků.

4.1 Výběr vhodného typu propojení

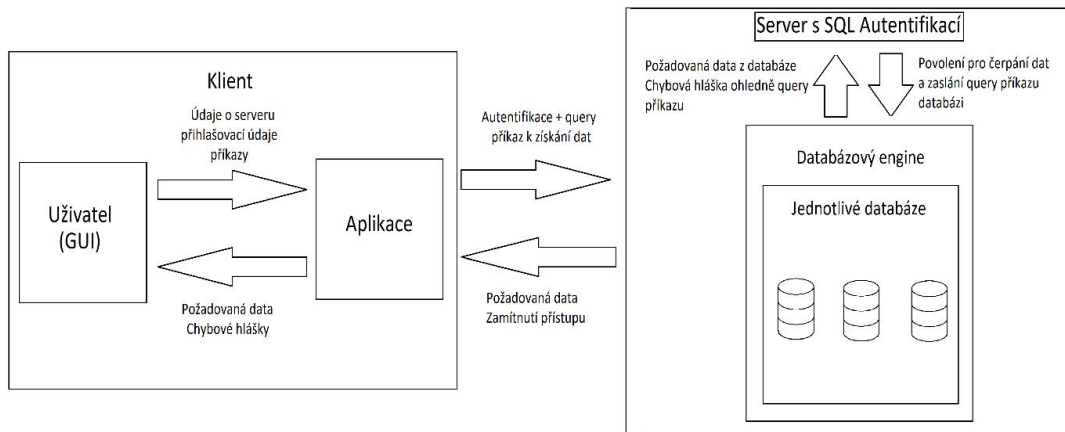
Prvotním úkolem při návrhu byl výběr správného typu propojení klienta k serveru, které by zaštiťovalo všechny zadané předpoklady. Tyto předpoklady byly, aby toto propojení dokázalo brát data z databáze umístěné na daném serveru tak, aby se dali dále zpracovat např. vynést do grafu či vypsat do textového pole. Na výběr bylo z několika druhů databázových serverů a propojení s nimi, ale pro tuto práci byl vybrán SQL Server Management Studio, dále jen SSMS, především kvůli předešlým znalostem práce s tímto databázovým systémem, jednoduchosti vytvoření serveru pro práci a podpoře propojení za pomoci .NET knihovny System.Data.SqlClient, která byla použita, jelikož je podporována i v rozhraní Mono, využívané v Unity3D.

Dalším požadavkem bylo propojení, které by dokázalo zpracovávat data uložená ve formátu XML na webovém serveru. K tomuto propojení a zpracování dat byl využit modul WWW, který je součástí základních knihoven Unity3D a na zpracování XML souboru bylo využito knihoven jazyka C# pocházejících také z rozhraní Mono.

4.1.1 SQL Server Management Studio

Jedná se o software, který je určen k řízení, upravování, administrativě a vývoji komponentů z SQL Serveru, Azure SQL Databáze a dalších. Tento nástroj obsahuje jak editor pro psaní kódů, tak i grafický editor pracující s objekty a funkcemi serveru. V softwaru nalezneme tři druhy průzkumníků. Prvním z nich je objektový průzkumník, který používá prostředí na bázi hierarchií, pro náhled a úpravu objektů z každé instance SQL Serveru. Průzkumník obsahuje tabulkový pohled na objekty instancí a také schopnost vyhledávání konkrétních objektů. Dalším druhem průzkumníka je průzkumník šablon. SQL Server nabízí několik šablon, což jsou soubory obsahující SQL skripty, které ulehčují tvorbu objektů v databázích. Tyto šablony jsou dostupné pro jednotlivá řešení a projekty. Také jsou dostupné pro tvorbu databází, tabulek, indexů, statistik a funkcí. Je možnost i tvorby vlastní šablony. Posledním průzkumníkem je průzkumník řešení, který obsahuje položky nazvané projekty, za pomoci kterých můžeme upravovat a řídit skripty databází, query soubory, data propojení a soubory. V řešení můžeme mít více projektů a také soubory či metadata, které jednotlivé řešení definují. Pro propojení s SSMS se využívá propojení s autorizací SQL Authentication, Windows Authentication a nebo Active Directory. Dále se volí mezi typem serveru, který může být Database Engine, Analysis Services, Integration Services, Reporting Services a Azure Storage. Pro naši aplikaci byl vybrán typ serveru Database Engine, vzhledem k práci s databázemi [28].

4.1. VÝBĚR VHODNÉHO TYPU PROPOJENÍ



Obrázek 4.1: Schéma propojení s SQL Serverem

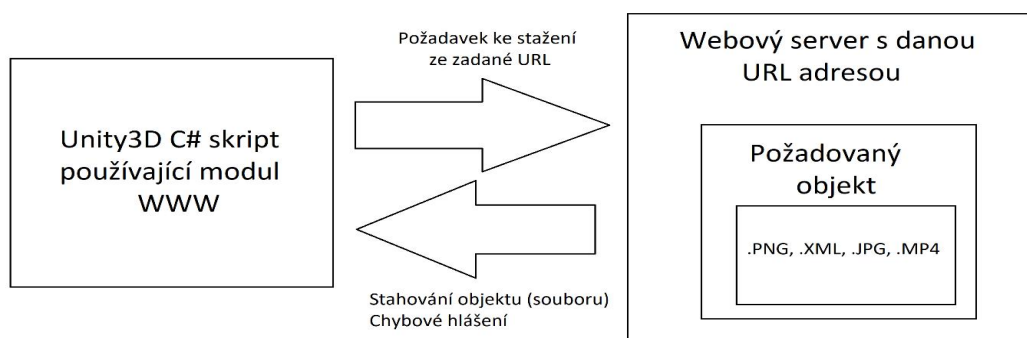
4.1.2 Modul WWW

Modul WWW se používá pro jednoduchý přístup k webovým stránkám a datům uložených na nich. Tento modul funguje tak, že po jeho zavolání začne stahování dat na pozadí, které vytvoří nový WWW objekt obsahující data z určité URL adresy. Na ukázce níže můžeme vidět jednoduchý případ získání dat z URL adresy za pomoci WWW modulu. V ukázce byla vytvořena proměnná url, která je typem String a obsahuje naši URL adresu. Dále je pomocí této proměnné vytvořena proměnná www typu WWW, která obsahuje data z URL adresy. Ukázku použití modulu WWW můžete vidět zde:

```
using UnityEngine;
using System.Collections;

public class Class1 : MonoBehaviour
{
    public string url="http://totojесmyslenaadresa.cz/" +
        "adresar/pripadnyobrazek.jpg";
    IEnumerator Start()
    {
        using (WWW www = new WWW(url))
        {
            yield return www;
        }
    }
}
```

Třidu WWW je možné použít jak pro získání, tak i posílání požadavků na server. Nejčastější využití tohoto modulu je pro čtení či zápis nejvyšších skóre na webový server, jelikož Unity3D je především určeno na vývoj her [29].



Obrázek 4.2: Schéma propojení za pomoci modulu WWW

4.2 Zpracování dat

V projektu jsou data uložena ve dvou formátech. V databázové tabulce, ze které se data získávají za pomoci příkazů ve formátu SQL jazyka a ve formátu XML, ke kterému jsou dostupné .NET knihovny pro výběr a zpracování dat. Těmito daty se rozumí údaje o teplotách, které hrají velkou roli ve strojírenství, kdy se mění vlastnosti výrobních strojů či materiálů při určitých teplotách, a proto je nutné tyto teploty sledovat. Data je možné zpracovávat i z jiných než uvedených jazyků, ale pro tuto práci byly vybrány tyto dva jazyky.

4.2.1 Jazyk SQL

SQL je zkratka pro tzv. strukturovaný dotazovaný jazyk (Structured Query Language), který se používá při programování a práci s databázemi. Tento jazyk je náhradou jazyka SEQUELE, který pochází z 70. let 20. století a byl vyvinut firmou IBM pro práci s relačními databázemi [30].

Příkazy v jazyce SQL lze rozdělit na čtyři základní skupiny:

- Manipulace dat (příkazy: SELECT, UPDATE, DELETE ...)
- Definice dat (CREATE, ALTER ...)
- Řízení přístupových práv (GRANT, REVOKE ...)
- Řízení transakcí (COMMIT, ROLLBACK, START TRANSACTION ...)

Zdemůžete vidět ukázkou jednoduchého příkazu pro získání položky nazvané TV z databázové tabulky UT:

```
SELECT TV FROM UT
```

4.2.2 Jazyk XML

Data uložena ve formátu XML v projektu zastupují data získaná z URL adresy pomocí modulu WWW. Název XML je zkratka z anglického Extensible Markup Language. Jedná

4.2. ZPRACOVÁNÍ DAT

se o obecný značkovací jazyk podobný jazyku HTML, tudíž je také rozdělen do různých entit, které mohou obsahovat další entity. Jazyk je určen pro výměnu dat mezi aplikacemi a publikaci dokumentů [31]. Níže můžeme vidět ukázkou jazyka XML, která obsahuje data o předpovědi počasí ze dne 4.5.2018. Základní entitou je entita Weatherdata, která obsahuje všechna data a další entity. První sub-entitou je entita Location, která obsahuje informace o poloze místa, pro které chceme znát předpověď počasí. Další sub-entitou je entita Sun obsahující časy východu a západu slunce. Jako poslední je tu sub-entita Forecast, ve které nalezneme předpověď počasí pro naše hledané datum, které je v tomto případě 4.5.2018 a platí mezi časy 9:00 až 12:00:

```
<weatherdata>
  <location>
    <name>Brno</name>
    <type/>
    <country>CZ</country>
    <timezone/>
    <location altitude="0" latitude="49.1922" longitude="16.6113"
      geobase="geonames" geobaseid="3078610"/>
  </location>
  <credit/>
  <meta>
    <lastupdate/>
    <calctime>0.1031</calctime>
    <nextupdate/>
  </meta>
  <sun rise="2018-05-04T03:26:54" set="2018-05-04T18:14:41"/>
  <forecast>
    <time from="2018-05-04T09:00:00" to="2018-05-04T12:00:00">
      <symbol number="800" name="jasno" var="01d"/>
      <precipitation/>
      <windDirection deg="41.5051" code="NE" name="NorthEast"/>
      <windSpeed mps="5.56" name="Moderate breeze"/>
      <temperature unit="celsius" value="21.85" min="21.09"
        max="21.85"/>
      <pressure unit="hPa" value="991.9"/>
      <humidity value="68" unit=""/>
      <clouds value="jasno" all="0" unit=""/>
    </time>
  </forecast>
</weatherdata>
```

5 Tvorba aplikace a jejích komponent

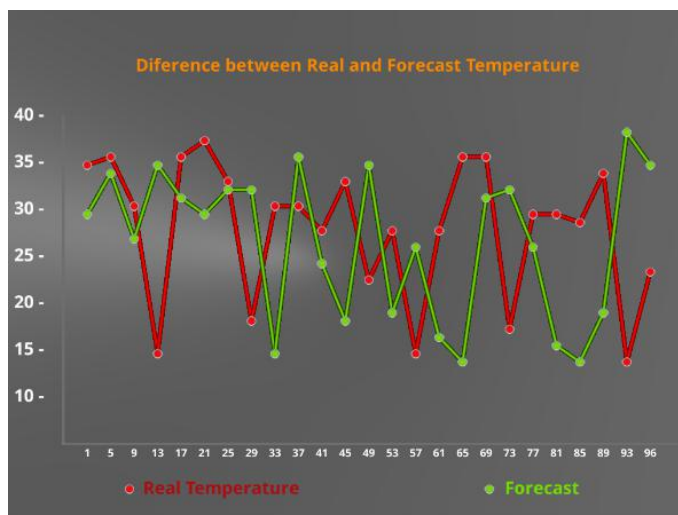
Tato kapitola pojednává o samotné tvorbě aplikace, realizaci propojení se serverem a o tvorbě prostředí v aplikaci Unity3D. Nejprve zde bude rozebrána tvorba grafů, způsob jejich vykreslování a jak byly naprogramovány. V další části bude popsána tvorba scény, umístění komponent v ní a celkově její obsah. Dále zde bude rozebráno samotná realizace propojení.

5.1 Tvorba grafů

Jedním z požadavků na bakalářskou práci bylo také vytvoření ukázky funkčnosti propojení, ve které bude obsaženo zpracování dat do grafů. Z počátku se tento požadavek jevil jako velmi závažný problém, jelikož grafické zpracování dat v Unity3D je nutno vytvářet přes objekty, které musí obsahovat další jednotlivé herní objekty, které tvoří samotný graf. Dalším problémem byla nedostatečná znalost práce s těmito objekty. Napříč těmito komplikacím bylo vykreslování grafu vytvořeno za pomoci skriptů sdílených na internetovém fóru *StackOverFlow*¹.

5.1.1 Grafické zpracování

Základem celého grafu je objekt typu Canvas obsahující pozadí, samotné osy grafu, jejich popis, název grafu, legendu a referenční nulový bod. Po spuštění vykreslení grafu se nám objeví objekt s názvem Holder, ve kterém se postupně začnou objevovat jednotlivé body, které mají přepočtenou polohu tak, aby odpovídali osám grafu. Tento objekt je po každém vypnutí či stisku tlačítka smazán a s ním i jednotlivé body a propojení těchto bodů. K propojení bodů je použita funkce Line Renderer, která propojí vždy určitou dvojici bodů.



Obrázek 5.1: Ukázka vytvořeného grafu

¹Fórum je dostupné z: <https://www.stackoverflow.com>

5.1. TVORBA GRAFŮ

5.1.2 Popis kódu

Celý skript je rozdělen do 5 základních částí, které tvoří definice proměnných a jednotlivé metody BarGraphGreen, BarGraphRed, ShowData a ShowGraph. Metody BarGraphGreen a BarGraphRed jsou typu IEnumerator a metody ShowData a ShowGraph jsou typu Void. Jako první jsou zdefinovány globální proměnné, které získávají své hodnoty z jednotlivých prefab objektů.

Metoda BarGraphGreen a BarGraphRed

Vstupem do obou metod je typ GraphData obsahující řetězec hodnot a typ float, který udává velikost mezery mezi body. Metoda vytvoří nejprve vektor, obsahující souřadnice pro počáteční bod nazvaný startPoint. Dále následuje smyčka WHILE, jejíž trvání je podmíněno délkou vstupního řetězce s daty. V této smyčce vytváří jednotlivé další body a ty také propojí za pomoci Line Renderu.

Metoda ShowData

Vstupem do této metody je typ GraphData obsahující řetězec hodnot, typ integer obsahující číslo určující, o které počasí se jedná a typ float, udávající velikost mezery mezi body. Nejprve jsou hodnoty řetězce přepočteny na velikost samotného grafu pomocí smyčky FOR. Poté je metodou rozhodnuto, která metoda pro vykreslení hodnot (buď BarGraphGreen nebo BarGraphRed) bude spuštěna. Níže je uveden kód celé metody:

```
public void ShowData(GraphData[] gdlist,int weatherNum,float gap) {  
  
    for(int i = 0; i < gdlist.Length; i++)  
    {  
        // jelikož je osa Y od 0 do 7 dílků, musíme ji vydělit bodem  
        // s největší hodnotou, abys jsme dostali hodnotu rovnu či menší 1 poté  
        // ji můžeme vynásobit číslem 7, aby nám seděla osa do grafu.  
        // příklad: marbles = 90, highest = 90 so 90/90 = 1 poté 1*7 = 7  
        // takže pro 90, Y = 7  
        gdlist[i].marbles = (gdlist[i].marbles/highestValue)*7;  
    }  
    if(weatherNum == 1)  
        StartCoroutine(BarGraphRed(gdlist,gap));  
    else if(weatherNum == 2)  
        StartCoroutine(BarGraphGreen(gdlist,gap));  
}
```

Metoda ShowGraph

Nejprve je graf vyčištěn pomocí funkce ClearGraph(), která vymaže bývalé hodnoty z grafu, pokud jsou přítomny. Poté se podmínkou IF zjistí, zdali řetězce nejsou prázdné a po splnění podmínky začne vykreslovat jednotlivé body do grafu ve scéně. Níže lze vidět ukázkou kódu pro inicializaci objektu holder, což je objekt, nesoucí body grafu a objekty propojení těchto bodů a vyplnění polí získanými hodnotami:

```

holder = Instantiate(HolderPrefb,Vector3.zero,Quaternion.identity) as GameObject;
holder.name = "h2";

GraphData[] gd1 = new GraphData[graphDataReal.Count];
GraphData[] gd2 = new GraphData[graphDataForecast.Count];

for(int i = 0; i < graphDataReal.Count; i++){
    GraphData gd = new GraphData();
    gd.marbles = graphDataReal[i].marbles;
    gd1[i] = gd;
}
for(int i = 0; i < graphDataForecast.Count; i++){
    GraphData gd = new GraphData();
    gd.marbles = graphDataForecast[i].marbles;
    gd2[i] = gd;
}

```

5.2 Tvorba scény

Scéna projektu je navržena takovým způsobem, aby bylo možné ukázat propojení se serverem a zpracování dat do grafů a textů stylem připomínající realitu. Byl vytvořen model domu, který má simulovat reálné prostředí, ve kterém se lze pohybovat. Scéna je obohacena také o interaktivní akci, kdy při kliknutí na tlačítko se zobrazí obsah grafu se stejnými daty. Základní prostředí celé scény je vytvořeno pomocí funkce Terrain, díky které lze vytvořit terén, ve kterém se dá pohybovat. Vytvořený terén je ohraničen simulací hor, aby uživatel nemohl vyskočit ven z prostředí, což by způsobilo nutnost zastavení aplikace, jelikož by se uživatel nacházel v nekonečném prostoru. Základním bodem tohoto prostředí je dům, ve kterém se uživatel objeví při spuštění aplikace.

Samotný model domu byl vytvořen v programu Blender a skládá se z těchto částí: vnitřní stěny, vnější stěny, podbití, podlaha, strop, střecha, venkovní obložení domu a dřevěné desky kolem celého domu pro vytvoření terasy. Tyto jednotlivé části jsou tvořeny z objektu krychle, který byl dále vytvářován do potřebného tvaru. Pro ještě realističtější pocit na jednotlivé části byly použity volně dostupné grafiky a textury jednotlivých materiálů např. dřevěná textura, černý mramor a další. Stěn jsou stěny udělané tak, aby nebyly průchozí, čehož bylo docíleno za pomoci tzv. box colliderů, což jsou speciální objekty v prostředí Unity, které vytvoří neprůchozí objekt, kolem zdí. Pro větší volnost nejsou do oken umístěny objekty, které by měly průhlednou texturu simulující sklo.

Dům je vybaven objekty ve tvaru nábytku, který byl také vymodelován v prostředí Blender a dále na něj byly použity volně dostupné textury materiálů. Důležitým předmětem mezi vybavením domu je laptop ležící na stole, u kterého se uživatel po spuštění objeví, jelikož tento laptop slouží jako interaktivní předmět, který zobrazuje na své obrazovce aktuální předpověď počasí a po kliknutí na něj také zobrazí samotný graf na stěně za laptopem. Tento laptop obsahuje objekt typu Canvas, do kterého byl vložen textový objekt. V tomto textovém poli je vložen další objekt typu Panel obsahující posuvné pole ScrollRect s textem. Tento text je řízen skriptem, kdy po spuštění aplikace je spuštěn tento skript, který se připojí na samotný server s předpovědí počasí a ze souboru typu XML vybere data, převede je na typ String a tyto data vytiskne do tohoto textového pole. Tudíž už po zapnutí aplikace uživatel vidí předpověď počasí na samotném monitoru tohoto laptopu. Model laptopu byl stažen z volně dostupných balíčků z Asset Store.

5.3. OVLÁDÁNÍ

Dalším objektem v této scéně je červené tlačítko, které je zde z důvodu interaktivního zobrazení dat do přítomného grafu. Po stisku tohoto tlačítka je spuštěn skript, který se připojí jak na server obsahující databázi se simulací naměřených hodnot počasí, tak i na server obsahující obecná data o teplotách za poslední dny na námi zadané poloze. Tvorba grafu je popsána v kapitole 5.1. Do scény je také vloženo osvětlení za účelem zlepšení viditelnosti interiéru. Toto osvětlení má simulovat klasická světla, která najdeme v domácnosti. Bez tohoto osvětlení by byl interiér temný, jelikož je zde přítomno světlo simulující svit slunce, které způsobuje v interiéru stín.

5.3 Ovládání

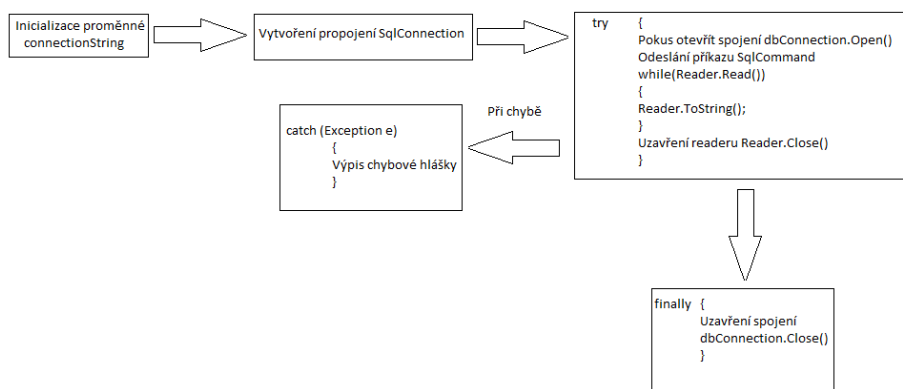
Důležitým prvkem aplikace je samotné její ovládání a navigace po vytvořené scéně. Předpokladem aplikace byla kompatibilita s VR headsetem od společnosti HTC Vive, tudíž byl použit speciální plugin pro VR od nazvaný VRTK. Tento plugin je možné najít v Asset Store přímo v Unity a je volně dostupný. Plugin obsahuje speciálně upravenou kameru, která je kompatibilní s připojeným headsetem, objekty pro ovládání pomocí ovladačů HTC Vive, teleporter a další. Pro pohyb byl využit skript z této knihovny, který je zde předvytvořen a samotná chůze je prováděna pomocí touchpadu na ovladačích. Pro určení směru chůze stačí, aby se uživatel podíval do směru, kam chce jít a zmáčkl již zmíněný touchpad. Vzhledem k náročnosti používání VR headsetu s počítačem, jelikož je headset náročný na hardware, byla do scény vložena také možnost pohybu pomocí klávesnice a rozhlížení pomocí myši. Toto ovládání bylo obsaženo jako standartní skript v balíčku Standard Assets přímo vytvořeného společností Unity dostupného z Asset Store.

Pro rozhlížení a otáčení po okolí scény je použit také skript z této knihovny, který získává informace o poloze a natočení headsetu a určuje pomocí nich souřadnice, na které se má kamera natočit. Pro interakci z interaktivními předměty ve scéně, jako je laptop či tlačítko, je použito tlačítko na ovladači HTC Vive, které je nazvané HairTrigger. Na model ovladače, který je objeven ve scéně je přidán Collider, pomocí kterého je zjištěna interakce s předměty.

5.4 Realizace propojení

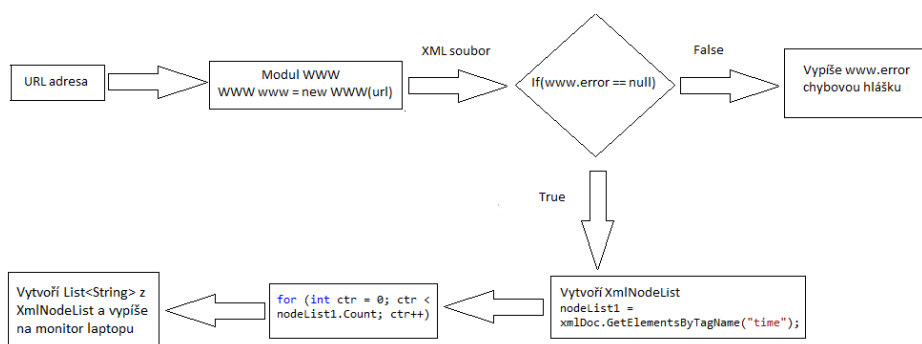
Propojení bylo realizováno dvěma způsoby, kde jako první bylo řešeno propojení pomocí SqlConnection knihovny s SSMS. Nejprve bylo nutné zadefinovat tzv. connection string, díky kterému bylo možné navázat spojení s databází na serveru. Tento řetězec obsahuje název serveru či číslo portu, název databáze, ke které se chceme připojit, uživatelské jméno a heslo. Tímto řetězcem je vytvořeno spojení k serveru, které nám umožní čerpat potřebná data z databáze. Aplikace obsahuje smyčku, která nám kontroluje, zdali se připojení zdařilo, či nezdařilo. Pokud se spojení nezdařilo navázat, je nám do konzole vytisknuta chybová hláška, kterou většinou způsobují špatně zadané údaje do řetězce. Tato smyčka je tvořena za pomoci funkcí try, catch, finally.

Pokud jsou všechny údaje správné, naváže se spojení se serverem, vytvoří se DataReader a odešle se příkaz typu SqlCommand psaný jazykem SQL. Provedení tohoto příkazu je dále uloženo do předem vytvořeného DataReaderu, jehož obsah je poté možno zpracovat jako typ string a uložit do proměnné. Tuto proměnnou můžeme použít jako nositele vstupních hodnot k vytvoření grafu s požadovanými hodnotami.



Obrázek 5.2: Schéma propojení se serverem SqlServerClient

Dalším typem použitého propojení bylo propojení modulem WWW. Vstupním bodem pro připojení je URL adresa, kterou byla zvolena volně dostupná adresa `www.api.openweathermap.org` obsahující data o předpovědi počasí na následujících pět dní od data spuštění aplikace. Modulem WWW se vytvoří propojení se serverem a následně je ověřeno funkcí `if`, zdali nenastal problém s komunikací. Pokud nenastane problém, funkce vytiskne na konzoli celý obsah souboru XML z webové stránky a spustí se filtrace dat z XML souboru. Tato filtrace je provedena funkcemi z knihovny `System.Xml`, což je základní .NET knihovna pro práci s XML soubory. Nejprve jsou vytvořeny seznamy, které obsahují informace o časech, počasí a teplotách v následujících dnech. Tyto seznamy jsou typu `XmlNodeList` a je potřeba je přetypovat do formy seznamu typu `String`. K tomuto účelu slouží cyklus `for`, který ukládá do seznamu typu `string` informace o času, teplotě a počasí od počáteční nulové hodnoty až do poslední hodnoty ze seznamu `XmlNodeList`, kterou zjistí za pomoci funkce `Count`. Tyto hodnoty teplot uložené v řetězci `String` lze dále zpracovat do řetězce typu `Integer` a tento řetězec vložit jako vstupní hodnoty k vytvoření grafu nebo je ve formátu `String` vytisknout do konzole či zvoleného textového pole vloženého do objektu.



Obrázek 5.3: Schéma propojení se serverem pomocí modulu WWW

5.4. REALIZACE PROPOJENÍ

Další vytvořenou metodou propojení je propojení za pomoci Windows Communication Foundation, zkráceně WCF. Nejprve byla vytvořena serverová služba (z ang. Server service) ve vývojovém prostředí Visual Studio, která byla automaticky vygenerována se skripty IService.cs, který slouží jako rozhraní a Service.cs. Ve skriptu Service1.cs byl původní obsah změněn následujícím způsobem:

```
namespace UnityService
{
    public class Service1 : IService1
    {
        public string GetData(int value)
        {
            return string.Format("You entered: {0}", value);
        }
    }
}
```

Tato úprava nám určuje, že při komunikaci, kdy mu pošleme číslo typu Integer, nám serverem bude vrácen text s hodnotou, kterou jsme na server poslali. Dále byl přidán další projekt do řešení nazvaný UnityWCF, který je konzolovou aplikací. V tomto projektu byl upraven konfigurační soubor app.config, aby bylo možné se připojit na server typu Localhost. Tato konzolová aplikace musí být spuštěna ještě před samotným programem v Unity, jelikož nám obstarává propojení se serverem. Aby bylo možné volání vytvořené WCF konzolové aplikace, je nejprve zapotřebí vytvořit proxy třídu. Tato proxy třída byla vygenerována za pomoci nástroje svcutil. Do příkazového řádku systému Windows bylo nutné zapsat cestu do adresáře Unity3D na disku a poté spustit příkaz, který vygeneruje potřebnou proxy třídu do souboru s názvem UnityWCFServerClient. V samotném editoru Unity3d byla vytvořena složka Plugins, do které byl vložen soubor knihovny System.ServiceModel.dll kompatibilní s prostředím Mono. Do složky Scripts byl umístěn vygenerovaný soubor UnityWCFServerClient a poté vytvořen nový skript s názvem ServerClient, při jehož zavolání byla navázána komunikace se serverem a spuštění funkce, která na něm byla umístěna. V případě této práce server vracel řetězec typu String a odeslané číslo. Níže je uveden hlavní metodu skriptu ServerClient obstarávající toto propojení a volání funkce:

```
public class ServerClient : MonoBehaviour {

    // Use this for initialization
    public void Start()
    {
        Service1Client client = new Service1Client(new BasicHttpBinding(),
            new EndpointAddress("http://localhost:8080/UnityService"));

        string receivedString = client.GetData(10);
        Debug.Log(receivedString);
    }
}
```

6 Závěr

V této práci byla nejprve rozebrána v rešeršní části samotná historie virtuální reality, která sahá až do první poloviny 20. století. Dále na toto téma navázala část popisující virtuální realitu v současnosti, a to z toho důvodu, aby si čtenář mohl představit pokrok, kterým tato technologie prošla za poslední půlku století. Byly popsány jednotlivé veřejně dostupné headsety pro virtuální realitu, jejich vlastnosti a použití. Cílem této práce bylo vytvoření aplikace, která by byla schopna implementovat propojení s externím zdrojem dat online do virtuální reality. Tímto propojením se rozumí získávání informací z databází a jiných typů dat. Před samotnou tvorbou aplikace bylo nutné seznámit se s editorem Unity3D, který je blíže popsán v kapitole 3 a také v prostředí aplikace Blender, pro vytváření jednotlivých modelů pro ukázkou funkčnosti propojení. Největším problémem při návrhu aplikace bylo vybrání vhodného typu propojení, které by zaštiťovalo všechny požadavky a také vytvoření grafu ze získaných dat. Propojení, které bylo vybráno po návrh aplikace je uvedeno v kapitole 4, kde jsou samotné typy propojení také podrobněji rozebrány. V poslední 5. kapitole je popsána samotná funkce jednotlivých komponent aplikace a jak jednotlivé tyto komponenty pracují.

Literatura

- [1] "Pygmalion's Spectacles," *Probably the First Comprehensive and Specific Fictional Model for Virtual Reality (1935)*. HistoryofInformation.com [online]. 2004- [cit. 2018-02-18]. Dostupné z: <http://www.historyofinformation.com/expanded.php?id=4543>
- [2] BROWN, Liza. A Brief History of Virtual Reality. In: *Filmora* [online]. [cit. 2018-05-21]. Dostupné z: <https://filmora.wondershare.com/virtual-reality/history-of-vr.html>
- [3] Nintendo Virtual Boy. In: *Techspot* [online]. 1998- [cit. 2018-05-23]. Dostupné z: <https://static.techspot.com/articles-info/1085/images/2015-12-04-image.png>
- [4] LINOWES, Jonathan.: *Unity Virtual Reality Projects: Explore the world of virtual reality by building immersive and fun VR projects using Unity 3D*. Birmingham, UK: Packt Publishing, 2015, 286 s. ISBN 978-1-78398-855-6.
- [5] PARISI, Tony.: *Learning Virtual Reality: developing immersive experiences and applications for desktop, web, and mobile*. Sebastopol, CA: O'Reilly Media, 2015, 127 s. First Edition. ISBN 978-1-49192-283-5.
- [6] CHAFKIN, Max. Why Facebook's \$2 billion bet on Oculus Rift might one day connect everyone on earth. In: *Vanity Fair* [online]. 2014- [cit. 2018-05-21]. Dostupné z: <https://www.vanityfair.com/news/2015/09/oculus-rift-mark-zuckerberg-cover-story-palmer-luckey>
- [7] KUMPARAK, Greg. A Brief History of Oculus, from Day Zero to Day One. In: *Medium* [online]. 2012- [cit. 2018-05-21]. Dostupné z: <https://medium.com/kickstarter/a-brief-history-of-oculus-from-day-zero-to-day-one-8878aae002f8>
- [8] The Development Kit 1. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-23]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/a/ae/Oculus_Rift_-_Developer_Version_-_Front.jpg
- [9] Developer Kit 2. In: *Techhive* [online]. San Francisco (CA), 2012- [cit. 2018-05-23]. Dostupné z: <https://images.techhive.com/images/article/2014/03/oculus-rift-dev-kit-2-100250984-orig.jpg>
- [10] The Oculus CV1. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-23]. Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/0/09/Oculus-Rift-CV1-Headset-Front.jpg>
- [11] The Oculus Touch. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-23]. Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/d/d8/Oculus-Rift-Touch-Controllers-Pair.jpg>
- [12] SOUPPOURIS, Aaron. How HTC and Valve built the Vive. In: *Engadget* [online]. 2017- [cit. 2018-05-21]. Dostupné z: <https://www.engadget.com/2016/03/18/htc-vive-an-oral-history/>

- [13] HTC Vive. In: *CZC.cz* [online]. Praha, CZ, 1998- [cit. 2018-05-23]. Dostupné z: https://iczc.cz/0k04k94hkshfqbvt01sb5b21b1_7/obrazek
- [14] HTC Controller. In: *VR Junkies* [online]. Orem, UT, 2016 [cit. 2018-05-23]. Dostupné z: <https://vrjunkies.com/wordpress/wp-content/uploads/2016/08/htc2.jpg>
- [15] PlayStation VR. In: *XinReality* [online]. 2015- [cit. 2018-05-21]. Dostupné z: https://xinreality.com/mediawiki/images/9/95/Project_morpheus1.jpg
- [16] LEONE, Matt. The making of Playstation VR. In: *Polygon* [online]. Vox Media, 2012- [cit. 2018-05-21]. Dostupné z: <https://www.polygon.com/2016/3/9/11174194/the-making-of-playstation-vr>
- [17] PlayStation VR. In: *XinReality* [online]. 2015- [cit. 2018-05-21]. Dostupné z: https://xinreality.com/wiki/PlayStation_VR#Developer
- [18] PlayStation Aim. In: *Flickr* [online]. 2004 [cit. 2018-05-23]. Dostupné z: https://c1.staticflickr.com/1/753/33195728041_c0b83f3244_o.png
- [19] Samsung Gear VR. In: *Samsung* [online]. [cit. 2018-05-23]. Dostupné z: [http://images.samsung.com/is/image/samsung/global-mkt-gear-vr-2016-gear-vr_supreme_vr?\\$Download-Source\\$](http://images.samsung.com/is/image/samsung/global-mkt-gear-vr-2016-gear-vr_supreme_vr?$Download-Source$)
- [20] HAAS, John.: *A History of the Unity Game Engine* [online]. Worcester, 2013 [cit. 2018-05-21]. 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
- [21] Unity - Fast Facts. In: *Unity* [online]. 2015- [cit. 2018-05-21]. Dostupné z: <https://unity3d.com/public-relations>
- [22] Unity - Manual: Creating Scenes. In: *Unity Documentation* [online]. 2015- [cit. 2018-05-21]. Dostupné z: <https://docs.unity3d.com/Manual/CreatingScenes.html>
- [23] Unity - Manual: Hierarchy window. In: *Unity Documentation* [online]. 2015- [cit. 2018-05-21]. Dostupné z: <https://docs.unity3d.com/Manual/Hierarchy.html>
- [24] Unity - Manual: Inspector window. In: *Unity Documentation* [online]. 2015- [cit. 2018-05-21]. Dostupné z: <https://docs.unity3d.com/Manual/UsingTheInspector.html>
- [25] Unity - Manual: Toolbar. In: *Unity Documentation* [online]. 2015- [cit. 2018-05-21]. Dostupné z: <https://docs.unity3d.com/Manual/Toolbar.html>
- [26] Unity - Manual: Creating and Using Scripts. In: *Unity Documentation* [online]. 2015- [cit. 2018-05-21]. Dostupné z: <http://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>
- [27] VILLAR, Oliver.: *Learning Blender: a hands-on guide to creating 3D animated characters*. Upper Saddle River, New Jersey: Addison-Wesley, 2015. ISBN 978-013-388-617-7.

LITERATURA

- [28] SQL Server Management Studio. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-21]. Dostupné z: https://en.wikipedia.org/wiki/SQL_Server_Management_Studio
- [29] Unity - Manual: WWW. In: *Unity Documentation* [online]. 2015- [cit. 2018-05-21]. Dostupné z: <https://docs.unity3d.com/ScriptReference/WWW.html>
- [30] SQL. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-21]. Dostupné z: <https://en.wikipedia.org/wiki/SQL>
- [31] Extensible Markup Language (XML). In: *World Wide Web Consortium (W3C)* [online]. [cit. 2018-05-21]. Dostupné z: <https://www.w3.org/XML/>
- [32] BALCÁREK, D.: *Virtuální svět*. [Diplomová práce]. Brno: VUT, FIT, 2015 [cit. 2018-05-21]. 64 s.
- [33] ROBINSON, Simon.: *C#: programujeme profesionálně*. Brno: Computer Press, 2003, 1300 s. Programmer to programmer. ISBN 80-251-0085-5.
- [34] ALBAHARI, Joseph a Ben ALBAHARI.: *C# 6.0 in a nutshell: the definitive reference*. Sebastopol, CA: O'Reilly Media, 2015. In a nutshell (O'Reilly & Associates). Sixth edition. 1114 s. ISBN 978-1-491-92706-9.

7 Seznam použitých zkratek a symbolů

| | |
|------|--|
| VR | virtuální realita |
| IDE | vývojové prostředí, z anglického <i>Integrated Development Environment</i> |
| SSMS | SQL Server Management Studio |

8 Seznam příloh

A Obsah přiloženého CD

1. Písemná zpráva ve formátu PDF
2. Zdrojové kódy
3. Vytvořená aplikace
4. Zdrojový tvar práce

B Obrázková příloha



