

Využití virtuální reality pro potřeby zaučování nových zaměstnanců v prostředí výrobních společností

Bc. Michal Bírošík

Diplomová práce
2019



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2018/2019

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Michal Bírošik**
Osobní číslo: **A17275**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **kombinovaná**

Téma práce: **Využití virtuální reality pro potřeby zaučování nových zaměstnanců v prostředí výrobních společností**

Téma anglicky: **The Use of Virtual Reality for the Training Process of New Employees in the Manufacturing Companies Environment**

Zásady pro vypracování:

1. Vytvořte literární rešerši o technologii virtuální reality a jejím praktickém využití.
2. Provedte důkladnou analýzu části výrobního procesu vhodného na vytvoření a nasazení aplikace virtuální reality.
3. Popište aktuální verze programů Blender a Unity a možnosti jejich využití v oblasti virtuální reality.
4. V programu Blender vytvořte komplexní 3D model vybrané části výrobního prostředí.
5. 3D model implementujte ve vývojovém prostředí Unity a vytvořte aplikaci virtuální reality tak, aby ji bylo možné reálně použít ve vybrané výrobní společnosti.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. MIHELJ, Matjaž, Domen NOVAK a Samo BEGUŠ. **Virtual Reality Technology and Applications**. Dordrecht [Netherlands]: Springer, 2014, 231 s. ISBN 978-94-007-6910-6.
2. PARISI, Tony. **Learning Virtual Reality**. Sebastopol: O'Reilly Media, 2015, 172 s. ISBN 063-6-920-03846-7.
3. BRITO, Allan. **Blender Quick Start Guide**. Birmingham [UK]: Packt Publishing, 2018, 194 s. ISBN 978-1-78961-947-8.
4. BLAIN, John M. **The complete guide to Blender graphics: computer modeling & animation**. Fourth edition. Boca Raton, FL: CRC Press, 2017, 697 s. ISBN 978-1138081914.
5. OKITA, Alex. **Learning C# programming with Unity 3D**. Boca Raton: CRC Press, 2015, 690 s. ISBN-13: 978-1-4665-8653-6.
6. LINOWES, Jonathan. **Unity Virtual Reality Projects**. Second Edition. Birmingham [UK]: Packt Publishing, 2018, 492 s. ISBN 978-1-78847-880-9.
7. **Blender 2.79 Reference Manual**. [online]. BLENDER Foundation, 2018 [cit.2018-11-26]. Dostupné z: <https://docs.blender.org/manual/en/latest/>
8. **Unity User Manual**. [online]. UNITY Technologies, 2018 [cit.2018-11-26]. Dostupné z: <https://docs.unity3d.com/Manual/index.html>

Vedoucí diplomové práce:

Ing. Pavel Pokorný, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

3. prosince 2018

Termín odevzdání diplomové práce:

15. května 2019

Ve Zlině dne 7. prosince 2018

doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Mgr. Roman Jašek, Ph.D.
garant oboru

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 15.05.2019

Michal Bírošík, v.r.

.....
podpis diplomanta

ABSTRAKT

Diplomová práce se zabývá technologií virtuální reality. Teoretická část práce obsahuje definici technologie, porovnání s rozšířenou realitou, její zařazení v konceptu smíšené reality, stručnou historii vývoje technologie a její současný stav. Práce nabízí rozbor technologie z hlediska principů funkčnosti a hardwaru, popisuje jednotlivé formy technologie a nástrojů potřebných k její využívání včetně softwarových a vývojářských řešení. Popsané a rozebrány jsou i hlavní nedostatky technologie a způsoby jejího využití. Součástí práce jsou také analýza a rozbor vybrané výrobní společnosti a výrobního procesu s postupem a popis nástrojů pro tvorbu navržené VR aplikace. V praktické části práce nabízí návrh řešení a zahrnuje celý postup vývoje VR aplikace pomocí zvolených nástrojů od tvorby 3D modelů, UV map, textur až po samotné vytvoření, sestavení a odzkoušení funkčnosti vytvořené aplikace. V závěru práce je aplikace zhodnocena.

Klíčová slova: Virtuálna realita, desktopová forma, HMD, headset, VR aplikácia.

ABSTRACT

Diploma thesis deals with the technology of virtual reality. The theoretical part of the thesis contains definition of technology, comparison with the augmented reality, its inclusion in the concept of mixed reality, a brief history of technology development and its current status. The thesis offers an analysis of the technology in terms of principles of functionality and hardware, it describes the various forms of technology and tools necessary for its use including software and development solutions. The main shortcomings of the technology and its utilization are described and discussed as well. The work also includes an analysis of a selected manufacturing company and the production process with the procedure and the description of tools used for creating the designed VR application. In the practical part, the thesis offers a solution design and it includes the whole VR application development process using selected tools, from creation of 3D models, UV maps, textures to the creation of the application itself, assembly and testing of the created application. At the end of the work the application is evaluated.

Keywords: Virtual reality, desktop form, HMD, headset, VR application.

Chcem sa poďakovať vedúcemu diplomovej práce Ing. Pavlovi Pokornému, Ph.D., za odborné rady, ktoré mi poskytol pri našej komunikácii a konzultáciách. Zároveň sa chcem poďakovať spoločnosti PRIA SYSTEM s.r.o. a jej pracovníkom, ktorí boli zadávateľmi témy a projektu diplomovej práce a ktorí boli vždy ochotní mi v prípade potreby poskytnúť odborné konzultácie alebo dohodnúť potrebné stretnutia. Rovnako im ďakujem za ich odborné rady pri vývoji. Tiež sa chcem veľmi poďakovať mojej rodine, ktorá ma podporovala a stála pri mne počas celého štúdia.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	10
I TEORETICKÁ ČASŤ	12
1 DEFINOVANIE A CHARAKTERISTIKA VIRTUÁLNEJ REALITY	13
1.1 DEFINÍCIA VIRTUÁLNEJ REALITY	13
1.1.1 Virtuálne prostredie	14
1.1.2 Virtuálna prítomnosť	14
1.1.3 Senzorická spätná väzba	15
1.1.4 Interaktivita	15
1.1.5 Čo virtuálna realita nie je	16
1.1.6 Koncept zmiešanej reality	17
1.2 HISTÓRIA VÝVOJA VIRTUÁLNEJ REALITY A JEJ SÚČASNOSŤ	18
1.2.1.1 Sensorama	18
1.2.1.2 Philco Headsight.....	19
1.2.1.3 The Ultimate Display.....	19
1.2.1.4 Sword of Damokles	19
1.2.1.5 Super Cockpit.....	20
1.2.1.6 Videoplance	21
1.2.2 Virtuálna realita v súčasnosti	21
2 ROZBOR TECHNOLOGIE VIRTUÁLNEJ REALITY	22
2.1 AKO TECHNOLOGIA PRACUJE	22
2.1.1 Stereoskopické displeje.....	22
2.1.1.1 Fresnelové šošovky	22
2.1.1.2 Parallax	23
2.1.1.3 Barelové skreslenie.....	24
2.1.2 Hardvér na sledovanie pohybu	24
2.1.2.1 Elektromagnetické sledovanie	26
2.1.2.2 Akustické (ultrazvukové) sledovanie	27
2.1.2.3 Optické sledovanie	27
2.1.2.4 Mechanické sledovanie.....	28
2.1.3 Vstupné zariadenia	28
2.1.4 Výpočtové platformy	29
2.2 VÝPOČTOVÉ PLATFORMY, ROZDELENIE TECHNOLOGIÍ A NÁSTROJOV VIRTUÁLNEJ REALITY	29
2.2.1 Desktopová virtuálna realita	30
2.2.2 Mobilná virtuálna realita.....	31
2.2.3 Konzolová virtuálna realita	32
2.2.4 Standalone VR.....	32
2.2.5 Natívne súbory na vývoj softvéru (SDK).....	33
2.2.6 Herné enginy a frameworky.....	33
2.2.7 Webové prehliadače.....	34
2.3 VIRTUÁLNA REALITA A JEJ HLAVNÉ NEDOSTATKY A OBMEDZENIA.....	34
2.4 SPÔSOBY VYUŽITIA VIRTUÁLNEJ REALITY	36
3 ROZBOR A ANALÝZA VÝROBNÉHO PROSTREDIA, ANALÝZA VÝROBNÉHO PROCESU	38

3.1	ROZBOR A ANALÝZA VÝROBNÉHO PROSTREDIA	38
3.1.1	SWOT analýza spoločnosti.....	39
3.1.2	Voľba vhodnej technológie a zariadení	42
3.2	ANALÝZA A ROZBOR VÝROBNÉHO PROCESU A POSTUPU VHODNÉHO NA NASADENIE APLIKÁCIE VR	42
3.2.1	Analýza postupu výrobného procesu.....	43
4	NÁSTROJE NA TVORBU NAVRHNUTEJ VR APLIKÁCIE	47
4.1	BLENDER VER. 2.79.....	47
4.1.1	Substance Painter ver. 2018.1.0	49
4.2	UNITY 3D VER. 2018.3.3F1 (64-BIT)	49
4.2.1	Microsoft Visual Studio 2017	51
II	PRAKTICKÁ ČASŤ	52
5	NÁVRH RIEŠENIA, MODELOVANIE, TEXTUROVANIE.....	53
5.1	NÁVRH RIEŠENIA.....	53
5.1.1	Výber formy VR a zariadenia VR.....	53
5.1.1.1	Porovnanie Oculus Rift a HTC Vive, návrh riešenia	54
5.1.2	Výber nástroja pre modelovanie 3D.....	57
5.1.3	Výber nástroja na textúrovanie	57
5.1.4	Výber nástroja pre tvorbu VR aplikácie	57
5.1.5	Výber IDE a programovacieho jazyka pre tvorbu skriptov	58
5.1.6	Návrh simulácie výrobného procesu a postup tvorby VR aplikácie	58
5.2	TVORBA, ÚPRAVA A EXPORT 3D MODELOV PROSTREDIA.....	60
5.2.1	Modelovanie, príprava.....	60
5.2.2	Tvorba 3D modelu.....	61
5.2.3	Úprava vytvorených 3D modelov	66
5.3	TEXTUROVANIE VYTVORENÝCH 3D MODELOV	67
5.3.1	Tvorba UV máp.....	67
5.3.2	Vytvorenie UV máp modelu	68
5.3.3	Textúrovanie	71
5.3.4	Úprava materiálov, tvorba, nasadenie a export textúr na vymodelované 3D modely	72
6	VYTVORENIE APLIKÁCIE S PODPOROU VIRTUÁLNEJ REALITY PRE ÚČELY ZAÚČANIA NOVÝCH ZAMESTANCOV VO VÝROBNOM PROCESE	78
6.1	VYTVORENIE VR APLIKÁCIE.....	78
6.1.1	Inštalácia, nastavenie VR hardvéru a vytvorenie nového projektu	78
6.1.2	Implementácia vymodelovaného 3D prostredia a jeho komponentov, úprava.....	80
6.1.3	Tvorba UI a pomocných obsahov.....	90
6.2	TVORBA APLIKAČNEJ LOGIKY PRE VYTVORENIE POŽADOVANEJ SIMULÁCIE	94
	ZÁVER	101
	ZOZNAM POUŽITEJ LITERATÚRY.....	103
	ZOZNAM POUŽITÝCH SYMBOLOV A SKRATEK	108
	ZOZNAM OBRÁZKOV	110
	ZOZNAM TABULIEK	113

ZOZNAM PRÍLOH	114
----------------------------	------------

ÚVOD

Súčasná doba výkonného hardvéru a pokroku vo vývoji veľkého spektra technológií a v oblasti softvérového a aplikačného vývoja umožňujú používať, nasadzovať a aplikovať technológiu virtuálnej reality na rôznych úrovniach využitia a na vybrané účely. Realizácia je na stále vyššej úrovni a vo forme bližšej reality a uveriteľnosti, ktorá je pre technológiu virtuálnej reality jedným z najdôležitejších faktorov. Z tohto dôvodu sme sa v diplomovej práci rozhodli venovať práve technológií virtuálnej reality a jej praktickému využitiu ako prostriedku na začatie nových alebo existujúcich zamestnancov vo výrobných spoločnostiach vo forme aplikácie virtuálnej reality, simulujúcej vybraný výrobný proces a jeho postup. Prostredníctvom vytvorenej aplikácie bude vybraná výrobná spoločnosť začať svojich zamestnancov.

V teoretickej časti práce sa zameriame na vysvetlenie, definovanie a vymedzenie pojmu virtuálna realita, porovnanie s rozšírenou realitou a zaradenie technológie v kontinuu zmiešanej reality. Následne stručne popíšeme históriu vývoja technológie, ktorá predchádzala vývoju technológie do súčasnej podoby, a prejdeme si vývoj technológie za posledné roky, ktoré vo veľkej miere prispeli k rozvoju komerčných verzií zariadení HMD. Následne vykonáme rozbor technológie na základe odbornej literatúry a odborných článkov, zaoberajúcich sa technológiou. Vysvetlíme, na akých princípoch technológia pracuje, a spojením akých technológií vytvára efekt virtuality a pohltienia. Charakterizujeme hlavné hardvérové a softvérové komponenty, prostredníctvom ktorých virtuálnu realitu využívame. Keďže technológia má svoje nedostatky a obmedzenia, popíšeme aj tie. V teoretickej časti práce stručne rozoberieme oblasti využitia virtuálnej reality. Súčasťou teoretickej časti bude analýza a rozbor vybranej výrobnéj spoločnosti a vhodnosti použitia technológie VR na nami zvolený účel, ako aj analýza a rozbor samotného výrobného procesu a jeho postupu, ktoré sú nevyhnutne potrebné pre návrh a realizáciu koncovej VR aplikácie. Stručne popíšeme aj nástroje na tvorbu samotnej VR aplikácie. V praktickej časti práce predstavíme návrh riešenia koncovej VR aplikácie, pričom zadefinujeme hardvérové a softvérové riešenia, ktoré sme sa rozhodli použiť. Voľbu jednotlivých hardvérových a softvérových riešení, ktoré použijeme na tvorbu koncovej VR aplikácie, odôvodníme. Po návrhu prejdeme k modelovaniu 3D prostredia, textúrovaniu vytvorených 3D modelov a samotnej tvorbe VR aplikácie. Výstupom praktickej časti bude aplikácia virtuálnej reality, vytvorená pre potrebu vybranej výrobnéj spoločnosti so zameraním na spôsob využitia formou simulácie výroby.

Cieľom práce je ukázať a využiť potenciál technológie virtuálnej reality vo zvolenej forme prostredníctvom vybraných komponentov a nástrojov ako prostriedku na zaučovanie. Vybraná forma využitia technológie môže byť v mnohých smeroch praktickou pomôckou pre vybranú spoločnosť, ale aj príkladom pre iné spoločnosti. Práca môže na základe teoretickej časti zároveň slúžiť aj ako vysvetlenie a ozrejenie technológie virtuálnej reality, jej definícií, princípov a spôsobov fungovania a možností jej využitia.

I. TEORETICKÁ ČASŤ

1 DEFINOVANIE A CHARAKTERISTIKA VIRTUÁLNEJ REALITY

Mnoho odborníkov definuje pojem virtuálna realita rôznymi spôsobmi. V predkladanej práci sa zameriame na definície, dôležité z pohľadu témy a zamerania predmetnej práce, ktoré podľa nášho názoru technológiu najviac vystihujú v jej podstate, spôsobe využitia a jej spôsobe fungovania.

V nasledujúcej časti práce si pre potrebu pochopenia technológie zadefinujeme pojem virtuálna realita, porovnáme ju s technológiou rozšírenej reality a presne zadefinujeme, kde sa nachádza virtuálna realita na kontinuu zmiešanej reality.

1.1 Definícia virtuálnej reality

„Realita je len ilúzia, hoci veľmi pretrvávajúca.“

-Albert Einstein

Vo všeobecnosti je virtuálna realita počítačovo generovaná simulácia 3D prostredia, ktoré sa javí ako veľmi reálne pre osoby, ktoré ju priamo zažívajú prostredníctvom použitia špeciálnych elektronických zariadení. Hlavným cieľom je dosiahnuť silný pocit prítomnosti vo virtuálnom prostredí [6].

Podľa inej definície možno virtuálnu realitu ponímať ako vysoko interaktívne, počítačovo založené multimedialne prostredie, v ktorom sa používateľ stáva účastníkom v počítačovo generovanom svete. Ide o simuláciu skutočného alebo predstaveného prostredia, ktoré možno vizuálne preskúmať v troch dimenziách, a to v šírke, výške a hĺbke, a ktoré môže zároveň poskytovať vizuálne interaktívny zážitok pohybom v plne reálnom čase so zvukom a s možnosťou hmatových a iných foriem spätnej väzby. Virtuálna realita je pre ľudí spôsob, ako možno vizualizovať, manipulovať a komunikovať s počítačom a mimoriadne zložitými dátami. Technológia ruší bariéry medzi ľuďmi a počítačmi. Virtuálna realita vytvára simuláciu reálnych životných situácií [9].

Ďalšia definícia od odborníkov na technológiu virtuálnej reality a jej aplikácie hovorí, že virtuálna realita sa skladá z interaktívnej počítačovej simulácie, ktorá vníma užívateľov stav a prevádzku, a nahrádza alebo rozširuje informácie senzorickej spätnej väzby do jedného alebo viacerých zmyslov, a to v takom zmysle, že používateľ má pocit, že je pono-

rený do simulácie (virtuálneho prostredia). Môžeme teda identifikovať štyri základné elementy virtuálnej reality:

- 1) Virtuálne prostredie.
- 2) Virtuálnu prítomnosť.
- 3) Senzorickú spätnú väzbu (ako odpoveď na akcie používateľa).
- 4) Interaktivitu [1].

1.1.1 Virtuálne prostredie

Počítačovo generované virtualizované prostredie predstavuje popisy objektov v rámci simulácie a pravidiel, ako aj vzťahy, ktoré riadia tieto objekty. Virtuálna realita je pozorovanie virtuálneho prostredia prostredníctvom systému, ktorý zobrazuje objekty a umožňuje interakciu, čím vytvára virtuálnu prítomnosť. Virtuálne prostredie je určené jeho obsahom (objekty a znaky). Obsah sa zobrazuje rôznymi spôsobmi (vizuálnymi, zvukovými a hmatovými), pričom je používateľom vnímaný prostredníctvom videnia, počutia a dotyku [1].

1.1.2 Virtuálna prítomnosť

Môže byť rozdelená na fyzickú (zmyslovú) a duševnú prítomnosť. Predstavuje pocit, že používateľ v skutočnosti „žije“ v prostredí. Môže to byť buď úplne psychologický stav, alebo dosiahnutý stav prostredníctvom niektorého fyzického média.

- 1) Fyzická virtuálna prítomnosť je základnou charakteristikou virtuálnej reality a reprezentuje telo používateľa fyzicky vstupujúce na médium. Definuje virtuálnu realitu a súčasne ju oddeľuje od iných médií. Virtuálna prítomnosť sa dosiahne predstavením virtuálneho sveta používateľovi so syntézou stimulov k jednému alebo viacerým zmyslom, v reakcii na polohu a činnosti používateľa. Syntetické podnety sú umelo vytvorené a prezentované používateľovi, ale nie je potrebné ovplyvňovať všetky zmysly alebo zahŕňať celé ľudské telo [1].
- 2) Duševná virtuálna prítomnosť predstavuje stav tranzu, nadšenia (angažovanosti a očakávaní, pocitu, že používateľ je súčasťou virtuálnych svetov). Úroveň požadovanej duševnej virtuálnej prítomnosti závisí od cieľových aplikácií virtuálnej reality. U užívateľa môže mať rôznu intenzitu:
 - Užívateľia môžu vnímať spojenie s počítačom.

- Uživatelé môžu ignorovať skutočný svet a zamerať sa na interakciu s virtuálnym svetom, pričom stále poznajú rozdiel medzi reálnym a virtuálnym svetom.
- Uživatelé môžu byť tak ponorení do virtuálneho prostredia, že zabudnú, že je virtuálne [1].

1.1.3 Senzorická spätná väzba

Je kľúčovou súčasťou virtuálnej reality. Systém virtuálnej reality poskytuje priamu senzorickú spätnú väzbu používateľom podľa ich fyzickej polohy. Všeobecne platí, že väčšina spätných väzieb je poskytovaná prostredníctvom vizuálnych informácií, aj keď niektoré prostredia používajú iba hmatové informácie. Samozrejmosťou je potrebné sledovanie používateľovej polohy s cieľom poskytnutia vhodnej spätnej väzby. Systém musí mať teda schopnosť automaticky merať polohu a orientáciu objektov v reálnom prostredí [1].

1.1.4 Interaktivita

Ak má byť virtuálna realita realistická, musí reagovať na akcie používateľov. Inými slovami, musí byť interaktívna. Schopnosť užívateľa ovplyvniť počítačovo generované prostredie predstavuje jednu z najpodstatnejších foriem interakcie. Ďalšou možnosťou je zmeniť umiestnenie a pozorovací uhol, z ktorého používateľ prostredie sleduje [1].

Na základe predchádzajúcich definícií a množstva ďalších definícií od odborníkov z oblasti virtuálnej reality, ktoré sme pri písaní predmetnej práce preštudovali, sme zistili, že v hlavných aspektoch definícií sa vyskytovali rovnaké kľúčové vlastnosti, na základe čoho môžeme virtuálnu realitu v skratke definovať ako:

Počítačom generované trojdimenzionálne prostredie, ktoré môže užívateľ preskúmať, vytvárať s ním interakcie a spolupracovať v reálnom čase, pričom sa zároveň stáva súčasťou virtuálneho sveta, a teda je do generovaného prostredia plne ponorený. Technológia virtuálnej reality by mala spĺňať nasledovné kľúčové vlastnosti, ktoré sú zároveň aj jej požiadavkami:

- 1) Je registrovaná v počítačovo generovanom trojdimenzionálnom prostredí (3D) alebo vytvára dojem trojdimenzionality.
- 2) Celá simulácia prebieha v reálnom čase a je dynamického charakteru.
- 3) Poskytuje užívateľom interakciu, pričom je interaktívna v reálnom čase.
- 4) Má pohlcujúci charakter.

- 5) Je prezentovaná a poskytovaná užívateľom prostredníctvom zmyslov, za pomoci použitia špeciálneho hardvéru [1][2][6][9][10][11].

1.1.5 Čo virtuálna realita nie je

Mnoho používateľov si ešte stále aj napriek pomerne veľkej popularizácii a rozšírení dvoch rozdielnych technológií ako je virtuálna realita a rozšírená realita, tieto technológie mýli, zamieňa alebo spája do jednej. V tejto časti práce zdefinujeme, čo virtuálna realita nie je, a aké sú hlavné rozdiely medzi spomenutými technológiami.

Virtuálna realita nie je rozšírená realita. Podľa jednej z definícií rozšírenej reality môžeme identifikovať, že ide o technológiu zameriavajúcu sa na integráciu a miešanie počítačom generovaných dát v rôznych formách, ako napríklad 3D modely, zvuky, videá, popisy, pričom takto generovaným obsahom následne podľa potreby a použitia možno obohacovať vrstvu fyzickej reality. Rozširujeme teda realitu o digitalizovanú informačnú vrstvu, prebiehajúcu v reálnom čase. Podobne ako virtuálna realita má kľúčové vlastnosti, ktoré by mala technológia spĺňať:

- 1) Technológia obohacuje realitu o virtuálny obsah.
- 2) Je interaktívna v reálnom čase.
- 3) Je registrovaná v trojdimenzionálnom prostredí [13][31].

Najväčším rozdielom oproti technológií virtuálnej reality je, že rozšírená realita obohacuje realitu o virtuálny obsah, no nenahrádza ju kompletne počítačovo generovaným prostredím, ako je to v prípade virtuálnej reality. Pre lepšie vymedzenie a charakterizovanie porovnáme hlavné vlastnosti, ktoré spomínané technológie odlišujú a rozdeľujú.

Virtuálna realita je plne digitalizované, počítačom generované trojdimenzionálne prostredie, využívajúce softvérové a hardvérové prostriedky na dosiahnutie čo najrealistickejšieho vzhľadu generovaného prostredia, zobrazovaného a poskytovaného používateľovi. Hlavným cieľom je teda nahradiť realitu za vygenerované, plne pohlcujúce virtuálne prostredie. Pri využívaní technologických prostriedkov pre zobrazovanie virtuálneho obsahu je zabezpečené čo najpohlcujúcejšie zobrazenie, a teda, aby užívateľ nevidel a neregistroval reálny svet alebo ho vnímal v čo najmenšom rozsahu. Hardvérové komponenty slúžiace na zobrazovanie virtuálneho obsahu majú takzvaný uhol pohľadu (FOV, angl. field of view), ktorý je používateľovi zobrazovaný. V prípade virtuálnej reality je

FOV zariadení široký, z čoho vyplýva, že sledovacie hľadisko systémov vo vzťahu k reálnemu svetu nie je a nemusí byť presné. Na druhej strane pre technológiu rozšírenej reality je hlavným cieľom rozšíriť vrstvu fyzickej reality o počítačom generovaný digitalizovaný obsah v rôznych formách v nie úplne pohlcujúcom prostredí. Pre zobrazovanie obsahu rozšírenej reality zariadeniami je FOV malý. Naopak, čo sa týka sledovacieho hľadiska vo vzťahu k reálnemu svetu, musí byť čo najpresnejšie, a teda s maximálne možnou presnosťou z dôvodu úplného, čo najdokonalejšieho vytvorenia požadovanej ilúzie, v ktorej je digitalizovaný obsah fixovaný na reálny svet. Výpis hlavných rozdielnych vlastností technológií virtuálnej a rozšírenej reality uvádzame v tabuľke č.1 [12][13].

Tab. 1. Rozdielne vlastnosti medzi technológiami virtuálnej a rozšírenej reality[12].

	Virtuálna Realita	Rozšírená realita
Sledovacie hľadisko	Nízka presnosť	Nutná čo najvyššia presnosť
Generované prostredie, obsah	Nutná vysoká reálnosť	Minimálne rederovanie
Zobrazovacie zariadenia	Plne pohlcujúce	Minimálne pohlcujúce
FOV zariadení	Široký	Malý

1.1.6 Koncept zmiešanej reality

Koncept zmiešanej reality predstavili Milgram a Khisino. Jedná sa o konkrétnu podmnožinu virtuálnej reality a súvisiacich technológií, ktoré zahŕňajú zlúčenie reálneho a virtuálneho sveta niekde na virtuálnom kontinuu, ktoré spája kompletne reálne prostredia s úplne virtuálnymi prostrediami (obrázok č.1) [14][15].



Obr. 1. Koncept kontinua zmiešanej reality [15].

Koncept virtuálneho kontinua sa vzťahuje na zmes tried predmetov, prezentovaných na konkrétne situácie zobrazenia, ako je to znázornené na obrázku č.1, kde reálne prostredia (RP) sú zobrazené na ľavom konci kontinua a virtuálne prostredia (VP) na pravom konci kontinua. Prvý prípad vľavo definuje prostredia, pozostávajúce výlučne z reálnych objektov, a zahŕňa napríklad to, čo sa pozoruje prostredníctvom konvenčného zobrazenia reálnej scény. Druhý prípad od RP definujeme ako rozšírenú realitu a zahŕňa priame prezeranie reálnej scény, ale prostredníctvom špeciálnych elektronických zobrazovacích zariadení a systémov, pomocou ktorých sa renderuje virtuálny obsah do reálneho prostredia, pričom ostáva zachované, že reálne prostredie je doplnené o minimálny virtuálny obsah. Posledný prípad vpravo definuje prostredie pozostávajúce výlučne z virtuálnych objektov. Príkladom by mohla byť konvenčná počítačová grafická simulácia. Môžeme povedať, že reálne prostredie je plne nahradené počítačom generovanou grafikou. Ako posledný je definovaný prípad medzi rozšírenou realitou a virtuálnym prostredím, ktorý nazývame rozšírenou virtualitou. Jedná sa o prípad, kedy väčšina užívateľovho pohľadu je nahradená digitalizovaným virtuálnym prostredím a jeho obsahom, no napriek tomu je užívateľovi umožnené prehliadať aj malú časť reálneho prostredia. Z taxonómie konceptu zmiešanej reality nám vyplýva, že technológie rozšírenej a virtuálnej reality sú si blízke a mimo hardvérových komponentov, slúžiacich na vykonávanie predmetných technológií, sa rozlišujú na základe toho, aké množstvo virtuálneho obsahu použijeme [12][14][15].

1.2 História vývoja virtuálnej reality a jej súčasnosť

Pre lepšie poznanie a pochopenie technológie virtuálnej reality je dôležité poznanie histórie jej vývoja, ktorá predchádzala vývoju technológie do súčasnej podoby.

1.2.1.1 *Sensorama*

V roku 1957 Morton Heilig vyvinul a patentoval *Sensoramu*. Funkčný mechanický prototyp bol zostrojený v roku 1962. Bol to stroj, ktorý ponúkal zážitok a skúsenosti s virtuálnou jazdou na bicykli. Používateľ sedel v prístroji s trojdimenzionálnym zobrazením. Počas simulácie počul zvuky mesta, pociťoval vietor a vibrácie sedadla a riadidiel, dokonca cítil určité vône. Stroj poskytoval vysokú spätnú väzbu, aby vytvoril pocit, že sa užívateľ naozaj nachádza v simulácii. Neexistovala však žiadna interakcia s premietaným filmom. *Sensorama* bola prvým krokom v skutočnom vývoji virtuálnej reality (obrázok č.2) [1][16][17].



Obr. 2. Sensorama simulátor [16].



Obr. 3. Philco Headsight HMD [19].

1.2.1.2 *Philco Headsight*

V roku 1961 bol vo Philco Corporation vyvinutý prvý HMD displej (angl. Head-Mounted Display) s názvom *Headsight*. Jednalo sa o prilbu, ktorú tvoril displej s obrazovkou s katódovým lúčom a sledovacím systémom, čo umožnilo pomocou pohybov hlavy identifikovať jej pozíciu, a tým prístup do umelého prostredia vylepšeného sledovacím systémom. *Headsight* neposkytoval pohľad do virtuálneho prostredia, ale zobrazoval video zo skutočného, vzdialeného miesta. Možno to považovať za prvý príklad teleprezencie, aplikáciu virtuálnej reality, ktorá je dnes stále populárna. *Headsight* bol prvým skutočným vynálezom HMD (obrázok č.3) [1][16][17].

1.2.1.3 *The Ultimate Display*

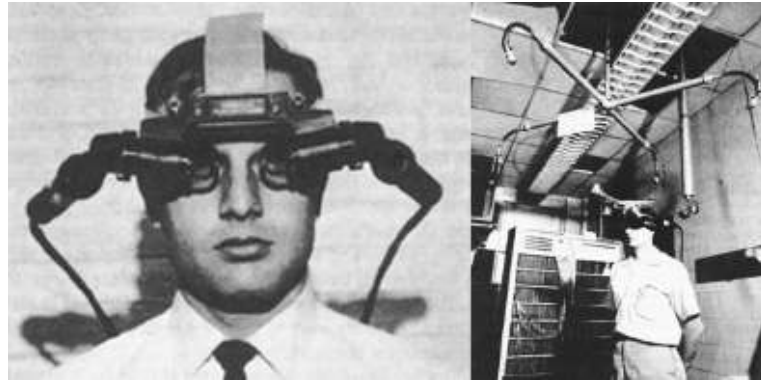
V roku 1965 napísal Ivan Sutherland dokument s názvom *The Ultimate Display*, ktorý predpovedal vzostup fantastických svetov viditeľných v dnešných počítačových hrách:

„Neexistuje žiadny dôvod, prečo by sa počítačom zobrazované objekty mali riadiť bežnými pravidlami fyzickej reality, s ktorou sme oboznámení“[1].

1.2.1.4 *Sword of Damokles*

Ivan Sutherland, priekopník v oblasti počítačovej grafiky, vynašiel a vytvoril v roku 1968 prvý skutočný počítačom sprostredkovaný systém virtuálnej reality nazývaný Damoklov meč. Bol to prvý BOOM display (Binokulárny Omni Orientačný Monitor). Systém pozostával z okuliarov s dvomi malými obrazovkami, jedna pre každé oko, ktoré pri spoločnom zobrazovaní vytvárali ilúziu trojdimenzionálnej vize. Bol pripevnený do stropu a presúval sa cez dlhý, výškovo nastaviteľný pól, do ktorého mohla byť vsunutá a pripevnená hlava užívateľa. Displej sledoval polohu oboch očí. Umožnil užívateľovi otočiť sa okolo Z osi o 360 stupňov, pričom zároveň sledoval orientáciu a polohu hlavy používateľa. Systém

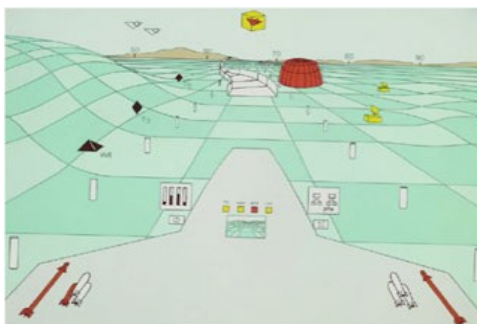
nebol plne pohlcujúci, nakoľko umožňoval užívateľovi vidieť časť reálneho prostredia cez priehľadné prvky hardvéru. Damoklov meč sa považuje za prvý prostriedok pre rozšírenú realitu - syntetické podnety sú nadradené stimulom zo skutočného prostredia. Damoklov meč bol predchodcom všetkých digitálnych okuliarov a aplikácií virtuálnej reality. Cieľom vynálezu bolo obklopiť používateľa zobrazenými trojdimenzionálnymi informáciami, ktoré sa pri pohybe menia (obrázok č.4) [1][16][17].



Obr. 4. Ivan Sutherland a HMD Damoklov meč [16].

1.2.1.5 *Super Cockpit*

V roku 1968 Tomas A. Furness III zaviedol technológiu virtuálnej reality do vzdušných síl USA. Navrhol a zostrojil vizuálne zobrazovacie systémy pre kokpity bojových lietadiel a vyvinul jeden z prvých prototypov letových simulátorov, založených na technológií virtuálnej reality. Vyvinuté HMD umožňovali užívateľovi po nasadení za zlých podmienok nahradiť pohľad virtualizovaným prostredím, ktoré bolo generované na základe vonkajších kamier a senzorov. Projekt sa stal známy pod názvom *Super Cockpit* (obrazok č.5) [12][19].



Obr. 5. Obraz zo *Super kokpitu* [19].



Obr. 6. Projekt *Videoplace* [19].

1.2.1.6 Videoplace

Prvé prostredia, ktoré reagovali na akcie používateľa, boli vyvinuté okolo roku 1970 Myrom Kruegerom. Použitím rôznych senzorov od videokamier až po tlakové snímače, systém VR mohol rozpoznať aktivity používateľov. Virtuálne objekty reagovali a fungovali ako skutočné. Keďže viacerí užívatelia mohli súčasne interagovať s virtuálnym prostredím, bol to aj prvý príklad viac užívateľského prostredia. Krugerovou najznámejšou tvorbou bolo prostredie *Videoplace*, ktoré zahŕňalo umelecké aktivity ako napríklad kreslenie na virtuálnych objektoch (obrázok č.6). Kruger tiež vytvoril pojem *Umelá realita*, ktorý popisuje rozpoznávanie užívateľských aktivít a generovanie spätnej väzby, ktorá posilňuje ilúziu aktivít vo virtuálnom prostredí. Účastníci boli schopní vytvárať interakciu jeden s druhým vďaka technikám spracovania obrazu, ktoré určovali ich pozície v 2D priestore [1][18][19].

1.2.2 Virtuálna realita v súčasnosti

Míľnikom, ktorý môžeme považovať za odštartovanie revolučného vývoja technológie VR, je úspech mladého vývojára Palmera Luckeyho, ktorý vo svojich 17 rokoch začal s experimentovaním vlastného VR headsetu. K vývoju sa pridala aj jeden z najlepších programátorov John Cormack. Spojenie Cormackových programátorských schopností s prototypom VR headsetu s názvom Rift, ktorý vyvinul Luckey, vyústilo v roku 2012 do vzniku startup projektu Oculus. Technológia sa stala súčasťou Kickstarter kampane. Po kampani sa postupne projekt Oculus Rift stal úspešným, čo viedlo k vzniku nezávislej spoločnosti, ktorú v roku 2014 kúpila spoločnosť Facebook za 1,91 miliárd eur. V roku 2013 spoločnosť Valve zdieľala technológiu umožňujúcu vznik obsahov pre virtuálnu realitu, ktoré na headsetoch nesekajú a zároveň poskytujú primeranú kvalitu. Technológia bola implementovaná Luckeyom vo vyvíjanom headsete Oculus Rift. V roku 2015 VR divízia spoločnosti Valve priniesla v spolupráci so spoločnosťou HTC vlastnú verziu headsetu s názvom HTC Vive, ktorá bola určená na využívanie VR v priestore. V roku 2015 nastal rapídny nárast vývoja technológií VR pre rôzne platformy, spôsoby použitia, ceny a kvality. V nasledujúcom roku sa technológii a jej vývoju venovalo viac ako 230 spoločností z celého sveta. Dnes si konkurujú produkty od firiem Oculus, Valve (zamerané na dektopovú a standalone formu VR), Microsoft (kombinácia VR a AR) a Sony Computer Entertainment (konzolová verzia VR). Spoločnosti Google, Samsung a iné sa zamerali na vývoj mobilnej a standalone verzie virtuálnej reality [11][19][24].

2 ROZBOR TECHNOLOGIE VIRTUÁLNEJ REALITY

Technológia virtuálnej reality vyžaduje ku vytvoreniu, nasadeniu a fungovaniu nutne potrebné hardvérové a softvérové komponenty, ktoré majú na starosti celý proces správneho fungovania na zvolených typoch zariadení a platform. Pre potreby diplomovej práce rozoberieme a popíšeme jednotlivé softvérové aj hardvérové technológie, komponenty a riešenia. Rozoberieme aj princíp fungovania, nedostatky technológie a možnosti využitia v rôznych odvetviach pre rôzne účely.

2.1 Ako technológia pracuje

Virtuálna realita má jeden cieľ - presvedčiť vás, že sa nachádzate niekde inde. Funguje na princípe oklamania ľudského mozgu najmä prostredníctvom zrakovej kôry a častí mozgu, ktoré vnímajú pohyb. Na vytvorenie ilúzie je nutné spojenie množstva technológií. Medzi technológie, ktorých spojením vytvárame ilúziu, a teda virtuálnu realitu patria hlavne:

- 1) Stereoskopické displeje.
- 2) Hardvér na sledovanie pohybu.
- 3) Vstupné zariadenia.
- 4) Výpočtové platformy [2].

2.1.1 Stereoskopické displeje

Známe ako trojdimenzionálne (3D) stereoskopické displeje alebo displeje umiestniteľné na hlave (HMD, angl. Head Mounted Displays), tiež známe ako headsety. Zobrazovacie zariadenia používajú kombináciu viacerých obrazov, realistické optické skreslenie a špeciálne napríklad Fresnelové šošovky pre vytvorenie stereofónneho obrazu, ktorý naše oči interpretujú ako s trojdimenzionálnou hĺbkou. Aby sme vytvorili ilúziu hĺbky, potrebujeme vygenerovať samostatný obraz pre každé oko zvlášť, jeden mierne posunutý od druhého, aby boli schopné simulovať tzv. Parallax [2][26].

2.1.1.1 Fresnelové šošovky

Typ šošoviek, ktorý bol vynájdený takmer pred 200 rokmi pre použitie v majákoch. Segmentuje štandardné zakrivené šošovky do malých zakrivených sekcií. Pri pohľade z prednej strany šošovky, ako je znázornené na obrázku č.7, to vyzerá, že na povrchu šošovky sú koncentrované kruhové hrebene. Fresnelové šošovky môžu byť veľmi veľké s krátkou ohniskovou vzdialenosťou, pričom sú stále tenké a ľahké, čo je ideálne na použitie v zaria-

deniach virtuálnej reality. Fresnelové šošovky tiež produkujú optickú difúziu, ktorá pomáha minimalizovať tzv. screen-door effect [26].

- screen-door effect - v zariadeniach virtuálnej reality je screen-door efekt vizuálny artefakt spôsobený displejom vo vnútri headsetu. Moderné ploché panelové displeje používajú pixely, čo sú v podstate malé samostatné zobrazovacie body rozložené na panely. Medzi jednotlivými pixelmi je malý úsek voľného miesta, ktoré nesvieti a je čierne, čo vedie k tomu, že vidíte v renderovanom obraze čiernu vizuálnu mriežku [25].



Obr. 7. Klasická Fresnelová šošovka pri pohľade z prednej strany použitá v zariadení HMD HTC Vive [26].

Obr. 8. Pohľad z prednej strany na hybridnú Fresnelovú šošovku použitú v zariadení HMD Oculus Rift [26].

Okrem klasickej Fresnelovej šošovky existujú aj hybridné fresnelové šošovky s veľmi jemnými kruhovými hrebeňmi v kombinácii s bežnými konvexnými šošovkami, vďaka čomu je veľmi ťažké hrebene vidieť. Z dôvodu použitej kombinácie je screen-door efekt u hybridnej Fresnelovej šošovky sotva zistiteľný. Hybridné šošovky majú tiež dôslednejšie zameranie cez zorné pole, čo znamená, že sú viac zhovievavejšie k strate pocitu, aká je pozícia užívateľových očí v postavení oproti HMD. Hybridné Fresnelové šošovky sú využívané v zariadení HMD Oculus Rift (obrázok č.8) [25][26].

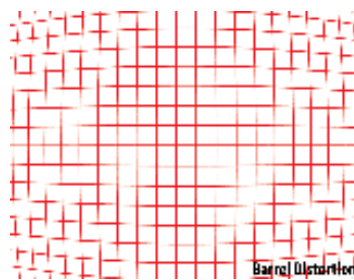
2.1.1.2 Parallax

Reprezentuje zmenu alebo rozdiel v zjavnej polohe objektu, pozorovaného pozdĺž dvoch rozdielnych zorných uhlov, a je meraný uhlom alebo polomerom sklonu medzi dvoma pozorovacími líniami. Parallax je vizuálny fenomén, pri ktorom náš mozog vníma hĺbku na základe rozdielu v zdanlivej polohe renderovaného prostredia a jeho objektov. Je to dosiahnuté vďaka vzájomne rozdielnej, posunutej polohe našich očí, ktoré predstavujú súčasne rôzne pohľady vďaka ich posunu, čo je aj základom stereopsie (procesu, pri ktorom

mozog využíva parallax z dôvodu rozdielnych pohľadov očí, aby bolo dosiahnuté hĺbkové vnímanie a odhad vzdialenosti od objektov) [2][21].

2.1.1.3 Barelové skreslenie

Pre vytvorenie naozaj dobrej ilúzie je potrebné skresliť obraz do lepšej emulácie sférického tvaru oka použitím techniky známej ako Barrel Distortion (barelové skreslenie). Je to stav, keď sú priamky zakrivené do vnútra v tvare barelu. Často videný jav na širokohlých šošovkách z dôvodu, že zorné pole šošovky (objektívu) je oveľa širšie ako veľkosť snímača obrazu, a preto potrebuje byť stlačená, aby sa zmestili. V dôsledku toho sú priame čiary viditeľne zakrivené smerom do vnútra, najmä smerom k extrémnym hranám okraju rámu. Čiary sa zobrazujú priamo v strede rámčeka a začínú sa ohýbať len od stredu. Je to z dôvodu, že obraz je rovnaký v optickej osi (stred objektívu), ale jeho zväčšenie klesá smerom k rohom (obrázok č.9) [2][20].



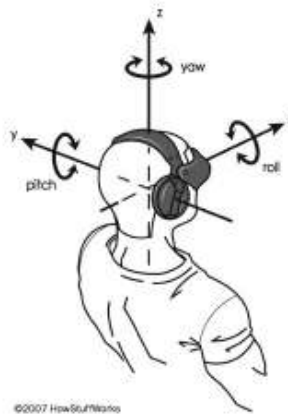
Obr. 9. Barelové skreslenie [20].

Spojením uvedených techník dostaneme vysoko kvalitné zobrazenie virtuálneho obsahu. Zo softvérového hľadiska je úlohou aplikácie renderovať snímky minimálne 60- a ideálne 120-krát za sekundu (označované tiež ako FPS, angl. frame per second), aby sa zabránilo vnímaniu akéhokoľvek oneskorenia, ktoré by mohlo narušiť ilúziu, alebo viesť k horším stavom ako je nevoľnosť, čo býva stav často spojený so slabou výkonnosťou virtuálnej reality. Výborným príkladom je výkonný hardvér VR Oculus Rift CV (Custom Version). Oculus Rift v sebe spája množstvo pokročilých technológií, slúžiacich na kvalitné zobrazovanie virtuálnych obsahov, vrátane vyššie spomínaných techník parallaxu a barelového skreslenia [2][11].

2.1.2 Hardvér na sledovanie pohybu

Ďalším dôležitým krokom, ako presvedčiť mozog, že sa nachádzame na inom mieste, je sledovanie pohybov hlavy a ďalších častí tela ako ruky, nohy alebo hrudník a na základe zmien polôh následné aktualizovanie renderovanej scény v reálnom čase. Keď užívateľ

vykoná akciu ako otočenie hlavy, zmenu pozorovacieho uhla alebo iný pohyb, dáta popisujúce predmetné správanie reprezentujú vstup pre počítač. Počítačové ovládače spracovávajú informácie a vytvárajú vhodnú spätnú väzbu. Výstupné zariadenie následne odovzdá spätnú väzbu ako výstup pre užívateľa. Môžeme povedať, že je to napodobňovanie reálneho stavu, ktorý nastane, keď sa rozhliadneme a pohybujeme v reálnom svete. Všetky trojdimenzionálne objekty majú tzv. 6DOF (six degrees of freedom, šesť stupňov slobody koordináty). Tri DOF sú určené pre pozíciu koordinátou X, Y, a Z offsetov a tri DOF na orientáciu, čiže zmeny v smere otáčania (obrázok č.10) [2][22][23].



Obr. 10. Zobrazenie 6DOF [23].

Vo všeobecnosti existujú dva typy systémov sledovania polôh a pohybov. Prvý dodáva absolútne dáta (celkové polohy/orientačné hodnoty) a druhý dodáva relatívne dáta (zmena dát od posledného stavu). Všetky systémy, určené na sledovanie polôh a pohybov, musia podporovať relatívne alebo absolútne dáta alebo ich podmnožiny. Pri sledovacích systémoch sa berú do úvahy nasledovné činitele:

- 1) Rozsah – pracovný priestor, v rámci ktorého môže sledovací systém merať polohu a orientáciu so špecifikovanou presnosťou, rozlíšením a uhlovou konvergenciou systému.
- 2) Rozlíšenie – je najmenšou zmenou polohy a orientácie, ktorú dokáže sledovací systém rozpoznať. Najmenšie možné hodnoty znamenajú lepší výkon.
- 3) Presnosť – je meranie chyby v zistenej polohe a orientácii. Obvykle je udávaná v absolútnych hodnotách, pre polohu v milimetroch alebo v stupňoch pre orientáciu. Menšie hodnoty znamenajú lepšiu presnosť.

- 4) Latencia – čas medzi fyzickou akciou užívateľa a začiatkom prenosu správy, ktorá reprezentuje akciu užívateľa. Je meraná v milisekundách, nižšie hodnoty latencie prispievajú k vysokému výkonu.
- 5) Rýchlosť aktualizácie – definuje počet meraní za sekundu, udáva sa v Hz (Hertzoch). Vyššia rýchlosť aktualizácie podporuje plynulé sledovanie pohybov, ale zároveň vyžaduje väčšie množstvo výpočtového času a spracovania [22][23].

Sledovanie polohy hlavy je rovnako dôležité, ako kvalita stereo renderovania. Naše vnemové systémy sú veľmi citlivé na vnímanie pohybu, z čoho vyplýva, že vysoká latencia pri sledovaní polohy hlavy môže prerušiť pocit pohltienia alebo spôsobiť nevoľnosť rovnako ako pri oneskorení v stereo renderovaní. Hardvérové jednotky pre technológiu virtuálnej reality musia sledovať pohyby hlavy čo možno najrýchlejšie, a zároveň softvér na spracovanie dát, získaný z jednotiek, musí s technikou držať krok. Súčasný hardvérový prostriedok ako HMD, vyžadujúce rýchle sledovanie pohybov hlavy napr. pomocou vysokorýchlostných inerciálnych meracích jednotiek (IMU), slúžiacich na sledovanie hlavy, kombinujú gyroskop, akcelerometer a hardvér magnetometra, aby presne merali zmeny v polohe a rotácií. Systémy hardvéru virtuálnej reality používajú rôzne konfigurácie IMU jednotiek. Medzi ďalšie hlavné typy sledovacích systémov môžeme zaradiť systémy na báze elektromagnetického sledovania, akustického (ultrazvukového) sledovania, optického sledovania alebo mechanického sledovania [22][23].

2.1.2.1 Elektromagnetické sledovanie

Pohlucujúce systémy často využívajú elektromagnetické sledovanie. Sledovače sa skladajú zo statickej časti (vysielač alebo zdroj), pohyblivej časti (prijímače alebo snímače) a riadiacej jednotky. Zostavy vysielača a prijímača sú podobné, nakoľko obe obsahujú tri vzájomne kolmé antény. Každá anténa pracuje na jednosmernom alebo striedavom prúde, generuje magnetické polia (komunikačné médium) podľa meraní. Prijímače zhromažďujú linky generovaných magnetických polí a posielajú získané dáta do riadiacej jednotky na výpočet polohy a sledovania. Kvalitné elektromagnetické sledovacie systémy majú veľmi vysokú citlivosť a nízku úroveň latencie. Nevýhodou systémov je, že čokoľvek, čo môže generovať magnetické pole, môže zasahovať do signálov vysielaných z vysielača do prijímača [18][22][23].

2.1.2.2 Akustické (ultrazvukové) sledovanie

Zvuk umožňuje určenie relatívnej vzdialenosti medzi dvoma bodmi pomocou použitia zostavy vysielačov a prijímačov. Na určenie polohy a orientácie objektu v priestore meraním relatívnej vzdialenosti používajú akustické sledovače ultrazvukové vlny v hodnote nad 20 kHz. Meria sa čas potrebný na to, aby ultrazvukové vlny z vysielača dosiahli k prijímaču. Systém vypočítava polohu a orientáciu objektu na základe času, kedy sa zvuk dostal k prijímaču. Na meranie sa používa metóda merania času letu (TOF, angl. Time of Flight) alebo koherentnej fázy (PC, angl. Phase Coherent). TOF zvyčajne poskytuje relatívne dáta s tendenciou zavádzania chyby. Akustické sledovacie systémy majú veľa nevýhod. Zvuk sa pohybuje relatívne pomaly, čiže rýchlosť aktualizácie cieľovej polohy objektu je podobne pomalá. Prostredie môže tiež nepriaznivo vplývať na efektivitu sledovacieho systému, nakoľko rýchlosť šírenia zvuku vzduchom sa môže meniť v závislosti od teploty, vlhkosti alebo barometrického tlaku v prostredí [18][22][23].

2.1.2.3 Optické sledovanie

Optické sledovače používajú rôzne konfigurácie a metodológie. Explicitne ich môžeme klasifikovať do troch hlavných kategórií:

- 1) Laserový rozsah – systémy na objekty prenášajú laserové svetlo, ktoré prechádza difrakčnou mriežkou. Následne senzor systému analyzuje difrakčný vzor na povrchu objektu pre výpočet jeho polohy a orientácie.
- 2) Rozpoznanie vzoru – na určenie polohy a orientácie objektu, sledovače porovnávajú známe vzory so vzormi snímanými.
- 3) Majákové sledovanie – metodika používa ako vysielač signálu v optickom zariadení tzv. súbor majákov (pozostáva zo sady infračervených LED) a snímačmi je rad kamier zachytávajúcich vyžarované infračervené svetlo z vysielačov, čiže obraz majákového vzoru. LED diódy svietia v postupných impulzoch. Kamery zaznamenávajú vysielané impulzné signály a odosielajú získané informácie do riadiacej jednotky systému. Riadiaca jednotka následne extrapoluje dáta na určenie polohy a orientácie objektu. Použitím známych geometrií majákov môžeme odvodiť polohu a orientáciu sledovaného objektu [18][22][23].

Systémy na báze optického sledovania majú vysokú rýchlosť aktualizácie, z čoho vyplýva, že problémy s latenciou sú minimalizované. Nevýhodou systému je, že línia zorného uhla medzi kamerami a LED diódami môže byť zakrytá, čo vplýva a zasahuje do

procesu sledovania. Obyčajné svetlo alebo infračervené žiarenie môže tiež spôsobiť, že systém sledovania je menej účinný [18][22][23].

2.1.2.4 *Mechanické sledovanie*

Používajú fyzické spojenie medzi cieľom a pevným referenčným bodom. HMD je namontovaný na konci mechanického ramena, ktoré sa skladá z kĺbových spojení. Systém rozpoznáva polohu a orientáciu pomocou určovania uhlov na spojoch ramena prostredníctvom zabudovaných potenciometrov a ozubených kolies. Rýchlosť aktualizácie je v prípade mechanického sledovacieho systému veľmi vysoká. Nevýhodou systému je, že mechanické rameno ku ktorému je HMD pripevnený, obmedzuje rozsah pohybu užívateľa [22][23].

Gyroskop, akcelerometer, ultrazvuk, optika, elektromagnet a ďalšie techniky a komponenty sa používajú v hardvéri virtuálnej reality, aby zisťovali, kedy sa naše telá pohybujú, naše hlavy otočia a na základe detekovaných zmien polôh a pohybov môže následne aplikácia aktualizovať náš pohľad v 3D virtualizovanej scéne v reálnom čase. Ak je stereo renderovanie, sledovanie pohybov hlavy a častí tela správne skombinované a aktualizované s dostatočnou frekvenciou, sme stále bližšie k nadobudnutiu skutočného pocitu ponorenia sa do skúseností a zážitkov vo virtuálnom svete [2][18][22][23].

2.1.3 *Vstupné zariadenia*

Pre vytvorenie presvedčivého pocitu pohltienia vytvára virtuálna realita potrebu nových typov a druhov vstupných zariadení mimo klávesnice a myši. Spomínanými vstupnými zariadeniami pre technológiu virtuálnej reality môžu byť:

- Herné ovládače - pripojené priamo k počítaču za účelom vytvárania interakcie s prostrediami virtuálnej reality. Pozostávajú z dotykovej (touchpad) plochy, ovládacích džojstikov a tlačidiel. Pripojené sú buď prostredníctvom usb kábla, prepojeného medzi portami zariadenia a PC, alebo použitím wifi a bluetooth technológií. Ovládače dávajú spätnú väzbu užívateľovi v podobe vibrácií. Niektoré typy majú v sebe zabudovaný aj gyroskop, a tak je možné určité zmeny polôh a pohyby simulovať pomocou náklonov zariadenia. Príkladom sú ovládače *Gamepad Microsoft Xbox* alebo *Sony DualShock 4*.
- Ručné sledovacie senzory - konzoly alebo zariadenia so zabudovanými senzormi, prostredníctvom ktorých sa sledujú pohyby rúk a prstov a rozpoznávajú gestá bez nutnosti dotyku alebo kontaktu so vstupným alebo iným zariadením.

Fungujú na princípe infračervených kamier. Poskytujú vyššiu formu pohľtenia, nakoľko užívateľ tak povediac vidí a používa na interakciu vo virtuálnom prostredí vlastné ruky. Nevýhodou je však absencia inej spätnej väzby ako vizuálnej. Príkladom môže byť technológia *Leap Motion*.

- Ručné a celotelové sledovacie systémy - zariadenia kombinujúce snímanie pohybov rúk s ovládacími prvkami, ktoré obsahujú herné ovládače. Najkomplexnejšia forma vstupných zariadení využívaných pre interakciu vo virtuálnom prostredí. Spätná väzba poskytovaná užívateľom je aj v podobe vibrácií, ktoré je možné podľa vlastných potrieb nakonfigurovať. Zariadenia spolupracujú so senzormi fungujúcimi na princípoch majákového sledovania alebo infračervených kamier. Senzory sú pripojené pomocou usb s PC. Príkladom je aj technológia *Oculus Touch* [2][22].

2.1.4 Výpočtové platformy

Zahŕňajú prevádzkový počítačový hardvér, operačný systém, softvér na komunikáciu s rozhraniami zariadení, frameworky a enginy, ktoré zabezpečujú beh aplikácií a softvérové a vývojárske nástroje na ich budovanie [2].

Bez všetkých vyššie spomenutých komponentov je ťažké dosiahnuť kompletne a plne hodnotné skúsenosti s virtuálnou realitou [2].

Rozdelenie výpočtových platforiem, čiže technológie virtuálnej reality podľa hardvéru, prostredníctvom ktorého VR aplikujeme a využívame spoločne so softvérovými riešeniami, rozdelíme a pre nás podstatné komponenty aj rozoberieme a popíšeme v nasledujúcej časti práce, a to z dôvodu využitia zvolených technológií pre riešenie zadaného projektu.

2.2 Výpočtové platformy, rozdelenie technológií a nástrojov virtuálnej reality

Virtuálna realita obsahuje a ponúka v súčasnosti pomerne širokú zbierku technológií od 3D displejov, hardvéru na sledovanie polohy a pohybov užívateľa, vstupných zariadení, softvérových riešení ako frameworky, modelovacie a vývojárske nástroje. Veľa aplikácií virtuálnej reality bude v súčasnej dobe fungovať na množstve existujúcich stolových, prenosných počítačoch, konzolách alebo mobilných telefónoch [2][6].

Tým sa dostávame k bodu, že technologický pokrok umožnil rozšírenie a rozdelenie virtuálnej reality na štyri hlavné kategórie podľa hardvérových zariadení a komponentov, prostredníctvom ktorých virtuálnu realitu vykonávame:

- 1) Desktopová virtuálna realita.
- 2) Mobilná virtuálna realita.
- 3) Konzolová virtuálna realita.
- 4) Standalone virtuálna realita[2][11].

Desktopovú formu virtuálnej reality si pre potrebu našej práce a jej využitia pre praktickú časť stručne popíšeme a charakterizujeme.

2.2.1 Desktopová virtuálna realita

Desktopová virtuálna realita je jedna z najvýkonnejších foriem používania technológie virtuálnej reality. Umožňuje ponoriť sa do plne pohlcujúceho virtuálneho prostredia a vykonávanie interakcií v reálnom čase, a to buď v takzvanej statickej, stacionárnej virtuálnej realite (užívateľ sedí na stoličke a na interakciu využíva herné ovládače, pričom je sledovaná len zmena polohy hlavy prostredníctvom HMD a poloha ovládačov, ale nie je sledovaná zmena polohy užívateľa v priestore), alebo v priestorovej virtuálnej realite nazývanej aj room-scale (je sledovaný pohyb užívateľa v priestore, čo znamená, že užívateľ si na základe možností použitej technológie môže vyhradiť tzv. pracovný priestor, v ktorom sa môže pohybovať. Pohyb je senzormi zaznamenávaný, zmeny aplikované a renderované do virtuálneho prostredia). Desktopovú formu virtuálnej reality vykonávame prostredníctvom nasledujúcich komponentov:

- 1) Výkonný stolový PC alebo notebook.
- 2) VR headset.
- 3) Vstupné zariadenia (ovládače).
- 4) Senzory.
- 5) Softvérové vybavenie (OS PC, programové ovládače pre VR hardvér, aplikácie) [2][6][32][33].

Využívanie technológie prebieha prostredníctvom nasadeného HMD, ktorý je pripojený pomocou USB káblov a HDMI káblom k zodpovedajúcim portom PC alebo pomocou prídavných bezdrôtových wifi adaptérov, ktoré zabezpečujú bezdrôtovú formu pripojenia HMD k PC. Bezdrôtová forma pripojenia HMD je výborným pokrokom vo vývoji

desktopovej formy VR využívanej v priestore, nakoľko eliminuje obmedzenie pohyblivosti užívateľa v jeho pracovnom priestore z dôvodu krátkosti káblových prepojení. Zariadenie HMD je tiež pomocou prídavného adaptéru napájané (obsahuje vymeniteľnú nabíjateľnú batériu). K PC v závislosti od použitej technológie môžu byť pripojené aj senzory, slúžiace na snímanie a rozpoznanie polohy a pohybov užívateľa a vstupných zariadení, ktoré môžu byť reprezentované špeciálnymi ovládačmi, ale aj samotnými rukami. Senzory sú rovnako pripojené prostredníctvom USB portov k PC. Veľmi dôležitou súčasťou desktopovej formy VR je výpočtové zariadenie, a teda stolový počítač alebo výkonný notebook, ktorý spája všetky potrebné komponenty, zabezpečuje komunikáciu medzi nimi a v neposlednom rade vykonáva všetky potrebné výpočtové, pamäťové operácie vrátane samotného behu koncových VR aplikácií. Aby bola celá komunikácia zariadení plne funkčná, je potrebné softvérové vybavenie PC počnúc operačným systémom, ovládačmi zabezpečujúcimi komunikáciu medzi jednotlivými hardvérovými komponentmi a PC a samozrejme aj programové vybavenie, ako aplikácie umožňujúce spustenie a beh VR zariadení a hotové VR aplikácie, simulácie, hry a podobne. V prípade tvorby VR aplikácií a obsahov sú potrebné vývojárske nástroje a frameworky, IDE, modelovacie nástroje. Požiadavky na výkon PC sú udávané v závislosti od použitého VR hardvéru. Každý výrobca a zariadenie má iné požiadavky na spúšťanie a bezproblémové využívanie zariadení virtuálnej reality, avšak môžeme s určitosťou povedať, že spoločnou požiadavkou pre zariadenia desktopovej formy virtuálnej reality je vysoký výkon výpočtového zariadenia. Reprezentantmi desktopovej formy technológie virtuálnej reality sú napríklad zariadenia Oculus Rift s príslušenstvom alebo HTC Vive, Vive Pro s príslušenstvom [2][6][30][32].

2.2.2 Mobilná virtuálna realita

V súčasnosti patrí k najrozšírenejšej forme využívania technológie virtuálnej reality z dôvodu, že veľa používateľov vlastní výkonné smart mobilné zariadenie, ktoré je pre mobilnú formu VR nevyhnutné, samozrejme v kombinácii s doplnkovým hardvérom, určeným na mobilnú VR. Môžeme teda povedať, že je to virtuálna realita kdekoľvek a kedykoľvek. Požiadavky na beh mobilnej VR sú:

- 1) Smartfón s operačným systémom Android s minimálnou verziou API 4.1 a vyššie alebo s operačným systémom iOS s minimálnou verziou API 8.0 a vyššie.

Požiadavky na smartfón:

- uhlopriečka zariadenia v rozsahu 4,7“(palcov) až 6“(palcov),
- displej zariadenia s čo najväčším rozlíšením (HD, 4K),
- zariadenie musí mať vstavané senzory akcelerometra a gyroskopu,
- výkonný CPU, odporúčané sú 4 jadrá,
- pamäť RAM 3GB,
- samostatný výkonný grafický procesor.

2) Headset určený pre mobilnú VR [2][6][32].

2.2.3 Konzolová virtuálna realita

Forma virtuálnej reality, ktorá beží na hracej konzole a práve konzola slúži ako hlavné výpočtové zariadenie. Hracie konzoly sú navrhnuté pre beh výkonovo a graficky náročných aplikácií a hier, čo znamená výkonné, kvalitné a pohlcujúce VR obsahy. Pre VR je potrebný headset vyvinutý na konzolovú verziu. Výkonovo a súborom implementovaných technológií sa headset približuje zariadeniam desktopovej formy VR.

2.2.4 Standalone VR

V prípade standalone formy je virtuálna realita vykonávaná prostredníctvom samotných headsetov. Nie je potrebné samostatné výpočtové zariadenie, nakoľko headset sám je výpočtovým zariadením s implementovanými a vstavanými všetkými potrebnými a dôležitými komponentmi. Standalone formu VR môžeme tiež nazývať ako All-in-One VR. Najnovšie verzie sa výkonovo približujú, možno aj rovnajú výkonu niektorých desktopových zariadení. Sledovanie polohy a zmeny polohy headsetu a ovládačov v priestore je prelomové, vykonávané prostredníctvom technológie obráteného sledovania. Jedná sa o 6DOF sledovanie, založené na špeciálnych ultraširokoúhlých senzoroch a algoritmoch počítačového videnia, ktoré vykonávajú a vypočítavajú presnú polohu v reálnom čase. Príkladom najnovšej high-end standalone formy VR je zariadenie Oculus Quest [6][29][32].

Pri výkonnosti jednotlivých foriem VR musíme podotknúť, že v prípade desktopovej formy virtuálnej reality platí pravidlo „čím väčší výkon výpočtového zariadenia, tým lepší zážitok“ dvojnásobne.

Keďže výpočtové platformy zahŕňajú aj softvér pre vytváranie aplikácií, a to v niekoľkých rôznych variantoch, jednotlivé varianty popíšeme. Vybrané nástroje využijeme v praktickej časti predmetnej práce, pričom v časti návrhu riešenia odôvodníme voľbu vybraných nástrojov.

Softvér na vytváranie aplikácií virtuálnej reality môžeme teda rozdeliť na:

- 1) Nativné súpravy na vývoj softvéru (SDK, angl. Software Development Kits).
- 2) Herné enginy a frameworky.
- 3) Moderné webové prehliadače [2].

2.2.5 Nativné súpravy na vývoj softvéru (SDK)

Sú ovládače zariadení a softvérové knižnice, používané v spojení s hosťateľským operačným systémom PC. Na OS Windows to môžu byť knižnice Win32 používané v C++ aplikáciách. Na systéme Android to sú knižnice Java a podobne. SDK sú zvyčajne určené pre špecifické frameworky a hardvér. VR SDK poskytujú základné nástroje na navrhovanie, vytváranie a testovanie obsahov, aplikácií a VR skúseností. Sú stavebnými kameňmi na tvorbu VR vo forme mobilných aplikácií, simulácií a ďalších obsahov podľa potreby a použitia. Ponúkajú nástroje na vykonávanie rôznych druhov funkcií od pridávania a duplikovania obsahov, až po pohybovanie 3D objektami. Nástroje môžu byť v prípade potreby upravené a prispôsobené prostredníctvom použitia API. Vývojári môžu budovať natívne aplikácie jednoducho použitím SDK a tvorbou vlastných pluginov s ohľadom na všetko ako 3D grafiku, herné správanie a podobne [2][34].

2.2.6 Herné enginy a frameworky

Pri použití herných enginov sa nemusíme zaoberať natívnym SDK. Knižnice herných enginov, známe tiež ako middleware, sa starajú o nízkoúrovňové detaily 3D renderovania, fyziku, herné správanie a rozhranie so zariadeniami. V herných enginoch je grafika aplikácií kompletne syntetická, založená na ručne vytvorených, vymodelovaných 3D modeloch, animáciách, pozadí a podobne. V súčasnosti využíva väčšina VR vývojárov herné enginy pre vývoj aplikácií, keďže majú veľkú podporu obsahov, pripravených práve pre technológiu a aplikáciu VR. Množstvo herných enginov má tiež silnú podporu pre cross platform integráciu, čo umožňuje písať zdrojový kód len raz (alebo väčšinu kódu s minimálnymi úpravami) a zacieliť ho na viaceré, či už desktopové, mobilné, konzolové alebo standalone platformy. Herné enginy často prichádzajú s výkonnými súbormi nástrojov, známymi ako level editory alebo integrované vývojové prostredia (IDEs, angl. Integrated Development Environments) [2].

2.2.7 Webové prehliadače

Tvorcovia prehliadačov rýchlo nasledujú vývoj virtuálnej reality. V prípade prijatia funkcií VR do webového prehliadača to vyzerá, že výsledkom prijatia budú dva hlavné vplyvy:

- 1) Znamená, že môžeme používať webové technológie ako HTML5, WebGL a JavaScript pre vytvorenie vlastných aplikácií, čo znamená rýchlejšie kódovanie a možnosť aplikácie pre viac platforiem.
- 2) Poskytnutie prístupu k všetkým existujúcim infraštruktúram, ktoré web ponúka, napríklad hyperlink medzi skúsenosťami s virtuálnou realitou, hosťovaním obsahov na klaudoch, rozvojom zdieľaných skúseností s viacerými užívateľmi a integráciou webových dát priamo do aplikácií virtuálnej reality[2].

Pre projekt predmetnej práce, a teda vývoj VR aplikácie, simulujúcej časť výrobného procesu a postupu, využijeme možnosť tvorby prostredníctvom herného enginu. Herný engine popíšeme a vyzdvihneme jeho kľúčové vlastnosti na konci teoretickej časti predmetnej práce. Následne na začiatku praktickej časti práce v časti návrhu riešenia na základe zváženia viacerých aspektov a potrieb odôvodníme voľbu herného enginu, v ktorom budeme vyvíjať našu výslednú VR aplikáciu.

Veľmi dôležitou súčasťou VR, čo v našom prípade predstavuje tvorba aplikácie VR, sú z dôvodu úplne syntetických prostredí nutne potrebné programy, určené na modelovanie 3D virtuálneho prostredia a jeho objektov, a tiež nástroje na tvorbu, aplikáciu textúr. Zároveň budeme potrebovať IDE na písanie skriptov, prostredníctvom ktorých budeme vytvárať v našej aplikácii interakcie a simulácie. Vykonáme popis programov a nástrojov na modelovanie, textúrovanie a skriptovanie, rovnako ako aj popis herného enginu, a to na konci teoretickej časti práce v časti nástroje na tvorbu navrhutej VR aplikácie a v praktickej časti práce v časti návrhu riešenia. Odôvodníme aj voľbu nástrojov.

2.3 Virtuálna realita a jej hlavné nedostatky a obmedzenia

Virtuálna realita nie je dokonalá technológia. Ako aj všetky iné technológie, aj VR má množstvo nedostatkov, pre ktoré „trpí“, a ktoré ju limitujú ešte aj v dnešnej dobe technologického pokroku. Medzi hlavné nedostatky a obmedzenia technológie VR patria:

- 1) Rozlíšenie obrazovky respektíve PPI/DPI, ktoré je považované za najväčší nedostatok. Nejde ani tak o rozlíšenie, ako o počet pixelov na jednotku plochy uvádzanú v palcoch (PPI, angl. Pixels Per Inch), alebo tiež udávanú ako počet bodov na jed-

notku plochy (DPI, angl. Dots Per Inch). Ľudské oko dokáže na vzdialenosť oka od zobrazovacieho zariadenia 100mm pri ostrosti zraku 1 uhlová minúta (,), čo je priemerná ostrosť zraku u dospelých ľudí, rozpoznať 876 PPI/DPI, čiže vidieť jednotlivé pixely/body obrazovky. To znamená, že pri vzdialenosti približne 10mm, čo je vzdialenosť oka od šošovky v súčasných headsetoch, by musel mať displej zariadenia približne 1200 PPI/DPI pri priemernej ostrosti zraku, aby sa jednotlivé pixely/body stali pre ľudské oko nerozpoznateľnými. Najmenšou možnou hodnotou ostrosti zraku je však 0,4 uhlovej minúty (reprezentuje najlepšiu možnú ostrosť zraku človeka), a teda pri vzdialenosti 100 mm oka od obrazovky by dokázalo oko rozpoznať až 2190 PPI/DPI. Môžeme teda tvrdiť, že pri zachovaní veľkosti zobrazovacieho displeja, no pri zväčšovaní jeho rozlíšenia, čiže zvyšovaní hustoty pixelov/bodov, je rozpoznateľnosť jednotlivých pixelov/bodov stále menšia. Rozlíšenie má teda vplyv na rozpoznateľnosť pixelov/bodov ľudským okom a jeho zvyšovaním sa postupne eliminuje najväčší nedostatok technológie [2][35].

- 2) Screen-door efekt – je viditeľnosť čiernej vizuálnej mriežky v renderovanom obraze zobrazovacieho zariadenia, spôsobené malými úsekmi voľného neosvetleného miesta medzi jednotlivými pixelmi displeja. Efekt môže byť minimalizovaný použitím vhodných šošoviek alebo použitím RGB subpixelovej matice [2][25].
- 3) FOV – problém s uhlom pohľadu je už takmer vyriešený, keďže súčasné HMD poskytujú FOV 110 stupňov, čo je takmer na hodnote možností ľudského oka, ktoré má FOV 120 stupňov [26].
- 4) Latencia – pri komunikácii medzi jednotlivými zariadeniami virtuálnej reality, pri behu samotnej aplikácie a renderovaní virtuálneho prostredia, ale aj pri zisťovaní polohy a zmeny polohy užívateľa v priestore a tiež pri interakciách vo virtuálnom prostredí. Na minimalizovanie latencie je potrebný veľmi výkonný hardvér výpočtového zariadenia, ale aj jednotlivých hardvérových komponentov virtuálnej reality [2][18][22][23].
- 5) Rozsah – maximálna možná veľkosť vymedzeného pracovného (hracieho) priestoru, v ktorom je možné sa pohybovať, a teda využívať technológiu virtuálnej reality, aby všetky jej komponenty pracovali správne. Rozsah je určený na základe možností senzorov, ktoré snímajú polohu a pohyb užívateľa v priestore. Obmedzením rozsahu trpí hlavne desktopová a konzolová forma virtuálnej reality. Technológia pokročila a začína sa testovať a implementovať nová forma sledovania polohy a

pohybov, ktorá nepotrebuje prídavné senzory, čiže rozsah priestoru použitia technológie by mal byť obrovský (ak vôbec limitovaný). Spomenutá technológia sa začína implementovať do standalone zariadení virtuálnej reality [22][29].

- 6) Káble – problém, ktorý obmedzuje pohyblivosť pri využívaní technológie VR. Problémom trpia headsety, ktoré zatiaľ nemajú implementované vlastné wifi rozhrania, prostredníctvom ktorých by mohli komunikovať s výpočtovým zariadením bez nutnosti pripojenia headsetu káblom. Problém s káblami tiež rieši spoločnosť zaoberajúca sa výrobou wifi adaptérov, ktoré je možné k headsetom pripojiť, a tak eliminovať problém s káblami. Napriek tomu potrebujú zariadenia aj napájanie, a to je riešené batériami s výdržou približne 2-3 hodiny, takže po uplynutí doby je batérie nutné vymeniť alebo pripojiť headset opäť pomocou káblov [26][30].
- 7) Zdravotné problémy – používanie technológie virtuálnej reality môže niektorým užívateľom spôsobovať zdravotné problémy ako nevoľnosť, točenie, bolesť hlavy alebo strata rovnováhy. V minulosti boli zdravotné problémy pri využívaní technológie oveľa častejšie z dôvodu jej zlej kvality, no zlepšovaním kvality technológie sa spomínané zdravotné problémy postupne eliminujú [9].

Napriek všetkým nedostatkom a obmedzeniam musíme však uviesť, že technológia v priebehu pár posledných rokov prešla zásadnými zmenami, úpravami a vylepšeniami, ktoré nám vo svojej podstate umožňujú používať technológiu na tak vysokej úrovni a vo formách použitia, ako ich dnes poznáme. Množstvo nedostatkov, na ktoré technológia „trpela“ v minulosti, bolo práve z dôvodu technologického pokroku odstránených alebo minimalizovaných.

2.4 Spôsoby využitia virtuálnej reality

Technológia virtuálnej reality našla svoje uplatnenie v mnohých odvetviach, v rôznych formách a spôsoboch využitia. Jedným z najväčších pozitív je, že technológia neslúži len na zábavu, ale uplatnila svoj potenciál aj v smere využitia technológie pre uľahčenie rôznych životných situácií, v rozvoji ľudského potenciálu, poznávaní, kreativite, ale aj pri riešení reálnych problémov. Môžeme teda tvrdiť, že sa stala nástrojom pre uľahčenie života. Spôsoby využitia technológie virtuálnej reality by sme mohli rozdeliť pre účely:

- vzdelávania,
- simulácie,
- vedy, výskumu,

- herného odvetvia,
- marketingu [2][6][9].

Pre predkladanú prácu je dôležitá možnosť využitia obrovského potenciálu technológie na účel simulátora. Technológia nám umožňuje vytvorenie rôznych druhov simulátorov, kde sa jednotlivci, či už vodiči, piloti alebo odborná obsluha, vzdelávajú, zaškoľujú a zlepšujú vo svojich oblastiach pôsobenia bez nutnosti zničenia drahého stroja, jeho neodbornou manipuláciou alebo dokonca ohrozením svojich spolupracovníkov alebo zákazníkov. Zamestnanca, ktorý obsluhuje jeden výrobný stroj, vieme bez problémov prostredníctvom virtuálnej reality zaškoliť na ďalšie stroje, ktoré dokáže následne obsluhovať v prípade potreby. Obrovskou výhodou je možnosť nasimulovania rôznych takzvaných „životných situácií“, ktoré môžu pri činnostiach nastať, a tak jednotliví obsluhujúci pracovníci nebudú pri ich skutočnom výskyte zaskočení, ale budú pripravení reagovať tak, aby vznikli napr. čo najnižšie alebo žiadne škody. Spôsob využitia na simulovanie vieme doplniť aj napríklad o virtuálny manuál, ktorý by reprezentoval účel vzdelávania, pričom užívateľ nemusí výrobný stroj len obsluhovať, ale doplnený virtuálny obsah mu umožní zobraziť rozbor celého stroja alebo si prezerat' jednotlivé časti strojov, zobrazovať k nim popisy, princípy fungovania. Virtuálny manuál sa môže stať výbornou pomôckou pri riešení problémov a chýb strojov. Tak sa môže pracovník vzdelávať v podstate bez nutnosti reálnej prehliadky výroby a rozoberania častí strojov alebo nutnosti absolvovania drahých odborných školení. Obrovskou výhodou použitia VR pre účel simulátorov a vzdelávania je aj šetrenie finančných prostriedkov [6][36].

Medzi hlavné praktické využitia technológie virtuálnej reality na účel simulovania môžeme teda zaradiť:

- 1) Školenia nových zamestnancov.
- 2) Krízové scenáre.
- 3) Diagnostika prvej úrovne.
- 4) Vizualizácia „za roh“ [37].

Ako sme už v práci spomenuli, zameriame sa na využitie technológie spôsobom simulátora, čiže vytvoríme aplikáciu virtuálnej reality, ktorá bude slúžiť ako prostriedok pre zaúčanie nových alebo aj existujúcich zamestnancov v prostredí výrobnej spoločnosti. Aplikácia bude prostredníctvom virtuálnej reality simulovať jeden výrobný postup, proces vybranej spoločnosti.

3 ROZBOR A ANALÝZA VÝROBNÉHO PROSTREDIA, ANALÝZA VÝROBNÉHO PROCESU

V nasledujúcej časti práce sa zameriame na rozbor, analýzu a zdokumentovanie výrobného prostredia, rozbor a analýzu výrobného procesu a jeho postupu, ktoré budú predmetom modelovania prostredia a simulovania výroby v našej VR aplikácií.

Súčasťou zdokumentovania bude aj dokumentácia na základe rozhovorov, fotodokumentácia a videodokumentácia výrobného prostredia a postupu, ktoré nám budú slúžiť ako podklad pre modelovanie samotného výrobného prostredia, jeho jednotlivých súčastí a komponentov a pri návrhu simulácie procesu výroby.

Nakoľko nám vybraná spoločnosť neposkytla žiadne reálne štatistické dáta, datase-ty alebo výstupy z predchádzajúcich analýz či už ekonomických, technologických, časových alebo iných štatistických dát, budeme pri rozbere a analýze spoločnosti, výrobného procesu, postupu a zhodnotení vhodnosti použitia technológie virtuálnej reality na zadaný účel pre vybranú spoločnosť vychádzať z informácií získaných na základe rozhovorov s členom predstavenstva spoločnosti, rozhovorov s človekom z výroby spoločnosti a tiež z dát a informácií získaných z internetových zdrojov.

3.1 Rozbor a analýza výrobného prostredia

Cieľom našej práce je využitie technológie virtuálnej reality ako prostriedku pre zaučovanie zamestnancov v prostredí výrobných spoločností. V našom prípade sa jedná o návrh a vývoj aplikácie virtuálnej reality, ktorá simuluje vybraný výrobný proces, realizovaný v danej spoločnosti. Pre zvolené riešenie bolo potrebné uskutočniť rozbor a analýzu vybranej výrobnéj spoločnosti, keďže budeme simulovať práve jeden z výrobných postupov, používaný na vybranej špecifickej technológii.

Predmetnou výrobnou spoločnosťou je tlačiarenská spoločnosť. Spoločnosť produkuje výhradne samolepiace etikety, pričom využíva najlepšie skúsenosti partnerov a dodávateľov, ktorí sú svetovými špičkami v odbore. Používané materiály, nástroje, technológie a technika sú v kombinácii s vysokou odbornosťou pracovníkov zárukou toho, že výsledné produkty spĺňajú najprísnejšie kritéria. Spoločnosť je zároveň aktívna v implementácii a nasadzovaní rôznych moderných technológií, zariadení a postupov, čo je plusovým faktorom aj pre náš projekt.

Z dôvodu zistenia vhodnosti použitia VR technológie pre účel zaučovania zamestnancov vo vybranej spoločnosti vykonáme SWOT analýzu spoločnosti s dôrazom na účel použitia technológie VR ako prostriedku pre zaučovanie a jej možných prínosov.

3.1.1 SWOT analýza spoločnosti

SWOT analýza je založená na zvažovaní vnútorných faktorov spoločnosti a faktorov prostredia. Do vnútorných faktorov spoločnosti zaraďujeme silné a slabé stránky spoločnosti a do faktorov prostredia zaraďujeme príležitosti a hrozby. Výhoda zobrazenia SWOT analýzy ako strategického nástroja spočíva v tom, že komplikovaná situácia je zúžená len na dve podstatné zložky, čo napomáha ľahšie prezentovať zložité súvislosti [38].

Cieľom SWOT analýzy je posúdenie vnútorných predpokladov spoločnosti k uskutočneniu určitého zámeru. V našom prípade nám SWOT analýza spoločnosti napomôže v riešení otázky vhodnosti využitia technológie virtuálnej reality pre proces zaučovania jej zamestnancov.

Treba však brať na vedomie aj fakt, ktorým je obmedzený prístup k zdrojom informácií, na základe ktorých sme pri analýze vychádzali. Pre dôveryhodnosť a overenie správnosti definovania a zaradenia jednotlivých vnútorných a vonkajších faktorov bola analýza konzultovaná a odobrená členom predstavenstva spoločnosti, z ktorého informácií sme vychádzali.

Tab. 2. SWOT analýza.

<i>Silné stránky – vnútorné interné prostredie</i>	<i>Slabé stránky – vnútorné interné stránky</i>
Zabezpečená surovinová základňa	Vysoká finančná zaťaženosť
Dlhodobá skúsenosť zamestnancov s výrobou a predajom	Vysoká materiálová náročnosť
Silná technická základňa	Nízky stupeň angažovanosti zamestnancov
Vysokokvalifikovaní technickí pracovníci	Fluktuácia zamestnancov
Vnímanie a implementácia európskych trendov a politiky riadenia firmy	Vysoké náklady na zaškolenie zamestnancov
Sociálna stabilizácia zamestnancov	Úrazovosť neodbornou manipuláciou

Pozitívne finančné ukazovatele spoločnosti	Porucha výrobných strojov, a tým zníženie výnosov a ziskov
Efektívne stanovenie poradia dôležitosti pri rozhodovaní o investíciách	Výpadok výrobných strojov z dôvodu potreby zaškolenia – zníženie ziskov
Ponuka pracovných miest	
<i>Príležitosti – vonkajšie externé prostredie</i>	<i>Hrozby – vonkajšie externé prostredie</i>
Rozšírenie sortimentu	Inflačný vývoj cien
Skvalitnenie výrobkov	Vývoj hospodárskej politiky
Tvorba nových pracovných podmienok	Rast cien energií, vody a stočného
Zvýšený rozvoj priemyslu v danom regióne	Cenová konkurencia
Možnosť kooperácie s vlastnými dodávateľmi	Prienik zahraničných firiem na trh, konkurencia
Zvýšenie konkurencie schopnosti firmy	Konkurencia v oblasti zamestnávania
Možnosť prieniku na iné trhy	
Rýchly rozvoj polygrafického priemyslu	
Inovácia vzdelávacieho a zaškolovacieho systému prostredníctvom využitia VR	

Na základe vykonanej analýzy môžeme povedať, že nedostatky sa vyskytujú v procese vzdelávania a zaškolenia nových pracovníkov.

Pracovníci sú jedným z kľúčových faktorov rozvoja spoločnosti, a preto je dôležité zdokonaľovanie ich zručností a ich rozvoj, aby zaisťovali chod efektívneho výrobného systému. Zamestnanci očakávajú zaškolenie, preškolenie a tréningy ohľadom používaných technológií a výrobných postupov, aby pochopili systém, a zároveň získali praktické skúsenosti. Vedenie firmy očakáva rýchle a efektívne zaškolenie, ktoré nebude zbytočne nákladné a nebude znižovať rentabilnosť, efektívnosť a ziskovosť produkcie. Veľa školení nie je efektívnych, je nudných, pričom nespĺňa účel zaškolenia. Po nasadení zamestnanca do výroby po zlom školení hrozí poškodenie drahých zariadení neodbornou manipuláciou,

ohrozenie zdravia a bezpečnosti, ako aj ďalšie možné negatívne vplyvy poškodzujúce chod firmy. Veľa školení je náročných na materiálne zabezpečenie, priestor a čas. Z vymenovaných dôvodov sa v súčasnosti začína klásť dôraz na prepojenie teórie s praxou, a to spôsobom precvičovania a zaškolenia formou simulácie. Hľadajú sa riešenia, ktoré zaškolenie formou simulácie ponúkajú.

Technológia virtuálnej reality nám poskytuje možnosť využiť ju na účel zaškolenia zamestnancov prostredníctvom simulácie.

Na základe vykonanej analýzy môžeme zhodnotiť, že vhodnosť použitia technológie virtuálnej reality pre zaučovanie zamestnancov vybranej výrobnéj spoločnosti je opodstatnená a adekvátne. Použitie technológie eliminuje hlavné nedostatky a nevýhody pri klasickej forme vzdelávania, medzi ktoré patria:

- citlivosť na poruchy,
- nízka pružnosť pri výrobných zmenách,
- produkovanie väčšieho množstva chýb,
- zastavenie celej výrobnéj linky,
- znižovanie využívania výrobnéj kapacity a rentability výroby,
- vysoké náklady na zaškolenie,
- nudnosť, neefektívnosť.

Medzi hlavné prínosy a výhody technológie virtuálnej reality v procese zaškolenia patria:

- 1) Nižšie náklady – nie je potrebná častá a nákladná odstavka výroby z dôvodu zaškolenia zamestnancov.
- 2) Zrýchlenie – zaškolenie a zácvičenie s použitím zintenzívnenia výučby (nie je potrebné čakať, kým sa uvoľní technológia k tréningovým účelom).
- 3) Skvalitnenie – je možné učiť sa prirodzenou cestou pokus/omyl, čo v reálnej situácii pri bežnej prevádzke nie je možné.
- 4) Zatraktívnenie – prostredie a spôsob zaškolenia zvýši záujem a pozornosť pracovníkov pri zaškolení a zároveň je atraktívnou témou pre PR.
- 5) Zjednodušenie – zníženie kognitívnej náročnosti zaškolenia (maximálna názornosť).
- 6) Zefektívnenie – možnosť vytvorenia simulácie a zaškolenia podľa vlastných predstáv a potrieb, možné rozšírenie, úprava existujúcich simulácií.

- 7) Interná skúsenosť – skúsenosť s využitím VR technológie ako odrazový mostík pre ďalšie aplikácie a rozšírenia v podobe VR zaškolovacích projektov.

3.1.2 Voľba vhodnej technológie a zariadení

Spoločnosť disponuje rôznymi technológiami, technikami a postupmi tlače, ktoré vykonáva prostredníctvom veľkého množstva zariadení a strojov. Medzi najvyužívanejšie zariadenia a technológie môžeme zaradiť:

- 1) Sieťotlač na sieťotlačovej linke SMAG.
- 2) Flexotlač prostredníctvom flexotlačiarenských strojov GALLUS, GALLUS EM280 a GALLUS RCS 330.
- 3) Ofsetová tlačiarenská technológia kombinovaná s rotačnou sieťotlačou a možnosťou horúcej razby či embosingu na stroji GALLUS.
- 4) Digitálna tlač pomocou tlačiarenského stroja XEIKON 3000[39].

Hlavnou, najpoužívanejšou a jednoznačne najvyťaženejšou technológiou tlače je v predmetnej spoločnosti flexotlač. Flexotlač je zároveň aj najmodernejšou technológiou spoločnosti, ktorá obsahuje kvalitu tlače porovnateľnú s ofsetom. Špeciálnou vlastnosťou tejto technológie je možnosť kombinácie rôznych tlačových technológií naprieč celým portfóliom. Zariadením, poskytujúcim vymenované, je flexotlačová linka GALLUS RCS300, umožňujúca navyše možnosť použitia rotačného sita, horúcej a studenej tlače, laminovania, výroby sendvičových etikiet a veľa ďalších možností. Stroj je schopný tlačiť aj na ťažko spracovateľný potlačový materiál ako tubalaminátové alebo BOPP fólie [39].

Tlačový stroj GALLUS RCS300 je na základe získaných informácií a konzultácie s predstaviteľmi spoločnosti vhodnou voľbou na tvorbu a nasadenie VR aplikácie, simulujúcej jeden z najpoužívanejších výrobných procesov a postupov, ktoré sa na predmetnom stroji vykonávajú.

3.2 Analýza a rozbor výrobného procesu a postupu vhodného na nasadenie aplikácie VR

Na základe overenia vhodnosti nasadenia aplikácie VR ako zaškolovacieho prostriedku a voľbe vhodného zariadenia sme prešli na výber a analýzu výrobného procesu a postupu vykonávanom na zvolenom zariadení, ktorý bude predmetom navrhutej a vytvorenej simulácie.

Keďže na zvolenom zariadení je možnosť použitia širokého spektra tlačových techník, bol vybraný proces a postup, ktorý je najčastejšie používaný a vykonávaný, a ktorý je zároveň vhodný svojou obtiažnosťou a náročnosťou na simulovanie vo vyvinutej VR aplikácii. Výber bol vykonaný na základe komunikácie s predstaviteľmi spoločnosti.

3.2.1 Analýza postupu výrobného procesu

Výrobným procesom s postupom je flexotlač na tlačiarenskom stroji GALLUS RCS330. Okrem samotnej obsluhy tlačiarenskeho stroja vyžaduje kompletný výrobný proces aj obsluhu ďalších komponentov a zariadení. Proces a postup zároveň vyžaduje vykonanie potrebných rozhodnutí a krokov od prípravy na tlač na základe definovanej zákazky až po nastavenie tlačového stroja do koncového stavu nutného pre úspešné zvládnutie tlače.

Výrobný proces začína prípravou. Tá spočíva v procese rozhodovania o použitých komponentoch ako je vhodná veľkosť tlačových valcov. Operátor výroby prejde k prvému zariadeniu, stojanu na tlačové valce. Stojan je pomôcka na uskladnenie tlačových valcov. Tlačové valce sú na stojane uložené a zoradené podľa ich veľkosti. Valce sa líšia svojím obvodom – tzv. raportom. Voľba veľkosti valcov je na operátorovi výroby. Tlačové valce s veľkým raportom sa hodia na veľké zákazky. Je to z dôvodu, že jediným otočením valec potlačí väčšiu plochu potlačového materiálu. Operátor výroby musí však brať do úvahy aj fakt, že tlačové valce s väčším raportom vyžadujú väčšie a drahšie tlačové polyméry. Operátor výroby zoberie vhodný tlačový valec a prejde s ním k ďalšiemu zariadeniu, stroju.

S vhodne zvoleným tlačovým valcom prichádza operátor výroby k stroju slúžiacemu na lepenie tlačových polymérov. Tlačové polyméry sa lepia na tlačové valce, pričom ich podstatnou úlohou v procese tlače je prenášanie farby na potlačový materiál v tlačiarenskom stroji. Veľkosť tlačového polyméru je volená operátorom výroby na základe veľkosti zákazky a veľkosti tlačových valcov. Od veľkosti polyméru sa odvíja aj jeho cena, čo je faktor, ktorý treba brať do úvahy pri voľbe vhodného polyméru. Zariadenie, slúžiace na lepenie polyméru, funguje na princípe vloženia tlačového valca do priestoru medzi dvoma ramenami. Následným stlačením tlačidla (pedálu) sa aktivuje hydraulický piest ľavého ramena a ten posuvom vložený valec medzi ramenami uzavrie. Po uzavretí tlačového valca medzi ramenami aktivuje operátor stlačením tlačidla, prepínača uzamknutie vloženého valca. Vo vrchnej časti zariadenia sa nachádzajú dva digitálne laserové mikroskopy. Obidva mikroskopy sú nastavované odborníkom, ktorý má schopnosti a vedomosti, ako správne mikroskopy nastaviť. V prípade vedomostí a znalostí o správnom

nastavovaní mikroskopov, môžu byť tieto mikroskopy nastavované aj operátorom výroby. Mikroskopy zohrávajú pri lepení polyméru dôležitú úlohu. Slúžia na kontrolu správnosti lepenia a nalepenia tlačového polyméru na tlačový valec, ktorý musí byť pre bezchybnú a vysokokvalitnú tlač nalepený čo najpresnejšie. Detailné priblíženia na časti a hrany polyméru sú zobrazované prostredníctvom obrazovky monitoru, ktorý je súčasťou zariadenia. Postup lepenia polyméru prebieha postupne od úvodného priloženia, zarovnania a nalepenia vrchnej začiatkovej strany (hrany) polyméru. Operátor skontroluje na zobrazenom priblížení vhodnosť úvodného nalepenia. Ak nie je polymér nalepený správne, nalepenú časť odlepí a postup opakuje pokiaľ nebude polymér nalepený správne a vyhovujúco. Ak je polymér nalepený správne, prejde operátor k stlačeniu tlačidla (spínača), ktorý spustí rotáciu ramien (pravé rameno obsahuje motorček, ktorý zabezpečuje rotáciu), prichytávacích vložený a zaistený tlačový valec. S ramenami rotuje aj vložený valec. Rotácia zabezpečuje dolepenie zvyšnej časti tlačového polyméru. Po úspešnom nalepení celého polyméru operátor skontroluje správnosť nalepenia prostredníctvom priblíženia z mikroskopov. Hotový tlačový valec s nalepeným tlačovým polymérom operátor stlačením tlačidla na odistenie odistí a následným stlačením tlačidla (pedálu) uvoľní z ramien piestov. Operátor výroby zoberie hotový tlačový valec s nalepeným polymérom a prejde s ním k ďalšiemu zariadeniu, stroju.

Operátor výroby prichádza s hotovým valcom k tlačiarenskému stroju. Tlačiarenský stroj slúži na samotnú tlač tlačových vzorov na potlačový materiál. Tlačiarenský stroj sa skladá z hlavných častí, ktorými sú:

- vstupný zásobník a podávače,
- jednotlivé tlačové jednotky tlačiarenskeho stroja,
- výstupný zásobník a podávače.

Vo vstupnom zásobníku je umiestnený potlačový materiál, ktorý po zahájení tlače putuje cez jednotlivé tlačové jednotky až na výstup tlačiarenskeho stroja a jeho zásobník, kde sa potlačený materiál opäť navíja do hotového balu. Tlačové jednotky zabezpečujú tlač častí vzoru koncovej tlače, a to v jednej konkrétnej farbe. Každá tlačová jednotka sa skladá z troch hlavných samostatných častí, a to:

- farebníková (rastrová) časť,
- tlačová časť (časť s tlačovým valcom),
- transportná časť.

Vo farebníkovej časti je umiestnený farebníkový zásobník a tzv. rastrový valec, ktorý má za úlohu prenášanie farby z vloženého farebníkového zásobníka na tlačový valec. Na tlačovom polyméry, nalepenom na tlačovom valci, vytvára rastrový valec tzv. raster (sieť drobných bodov farby, pričom veľkosť bodov odpovedá tzv. sile rastrového valca). Sila rastrového valca vyplýva z veľkosti „pórov“ na rastrovom valci, ktoré prenášajú buď viac alebo menej farby. Súčasťou farebníkovej (rastrovej) časti tlačovej jednotky sú aj záchytné nádoby, v ktorých sa zachytáva odpadová farba zo zásobníkov. Tlačový valec s nalepeným tlačovým polymérom sa vkladá na rotačné rameno tlačovej časti jednotky a zabezpečuje prenos farby na potlačový materiál. Transportná časť tlačovej jednotky zabezpečuje presun potlačového materiálu medzi tlačovými jednotkami od začiatku až po výstup tlačového stroja. Jednotlivé rotačné časti tlačových jednotiek sú poháňané samostatnými servomotormi, uloženými v kapotovanej časti každej tlačovej jednotky, v ktorej sa nachádzajú aj riadiace jednotky tlačových jednotiek. V spodnej časti každej tlačovej jednotky je umiestnená UV lampa, ktorá zabezpečuje vysušenie potlačeného vzoru medzi tlačovými jednotkami. Postup úkonov a krokov, ktoré musí operátor po príchode k tlačiarenskému stroju vykonať, je nasledovný. Operátor musí odistiť dve istiace skrutky (spodnú a vrchnú) na tlačovej časti tlačovej jednotky. Vrchná skrutka istí vyklápací panel tlačovej časti k rotačnému ramenu, spodná istí samotný vyklápací panel k posuvnej časti. Po odistení skrutiek operátor vysunie a vyklopí panel tlačovej časti. Na rotačnom ramene je umiestnená vložka tlačového ramena a valca. Vložka slúži na zaistenie tlačového valca na rotačnom ramene tlačovej časti. Operátor vložku vyberie, nasunie tlačový valec s polymérom na rotačné rameno tlačovej časti a následne vsunie vložku naspäť na rameno do časti, ktorá môže zároveň pokrývať časť vnútornej strany tlačového valca. Ak sú valec a vložka zasunuté na svojich miestach, operátor zaklopí a zasunie vyklápací panel tlačovej časti na pôvodné miesto a zaistí vrchné a spodné istiace skrutky. Rotačné rameno rastrovej a tlačovej časti jednotky obsahujú istiace piny, ktoré zaistia rastrový valec a tlačový valec s vložkou na mieste a zabránia ich posuvu. Aktivácia istiacich pinov je nevyhnutná pre spustenie procesu tlače, preto operátor výroby istiace piny aktivuje. Tlačiarenský stroj je takmer pripravený na spustenie tlače. Operátor musí ešte zabezpečiť výber farebníkového zásobníka a jeho vloženie do farebníkovej (rastrovej) časti tlačovej jednotky, na jemu určené miesto. Operátor výroby prejde k regálu slúžiacemu na uskladnenie naplnených a pripravených farebníkových zásobníkov.

Farebníkový zásobník je súčasťou každej tlačovej jednotky. Každá tlačová jednotka tlačí jednu konkrétnu farbu na potlačový materiál. Na tlač sa používa buď tzv. priama farba (konkrétny odtieň podľa Pantone vzorkovníku) alebo farba ako jeden kanál CMYK (Cyan Magenta Yellow Black) procesu. Vhodnú farbu mieša a plní do farebníkových zásobníkov grafik, vyškolený a zaučený na miešanie a dopĺňanie farieb. Farby sú miešané na základe požiadavky, ktorá vyplýva zo zadania zákazky. Operátor výroby vyberá vhodný pripravený a naplnený farebníkový zásobník, s ktorým následne prejde naspäť k tlačiarenskému stroju.

Po príchode k tlačiarenskému stroju a vybranej tlačovej jednotke, vloží naplnený farebníkový zásobník na miesto pre neho určené, ktoré sa nachádza vo farebníkovej (rastrovej) časti tlačovej jednotky. S úspešne vloženým zásobníkom môže operátor výroby prejsť ku konečnej fáze krokov, ktoré predchádzajú úspešnému spusteniu tlače. Pred spustením tlače musí byť do vstupnej časti tlačiarenskeho stroja vložený potlačový materiál.

Vloženie potlačového materiálu zabezpečuje vyškolený zamestnanec, ktorý má na starosti potlačové materiály (vrátane uskladenia), nakoľko na vloženie a zapracovanie materiálu do mechanizmu tlačiarenskeho stroja je potrebné vyškolenie a skúsenosť. Typ a veľkosť potlačového materiálu sú zvolené na základe požiadavky v zákazke. Operátor vizuálne skontroluje prítomnosť potlačového materiálu a prejde ku konečnej fáze zahájenia tlače - prejde k ovládacej jednotke tlačiarenskeho stroja.

Operátor aktivuje tlačovú časť a následne farebníkovú (rastrovú) časť tlačovej jednotky, ktoré po aktivácii presunú svoje polohy tak, aby sa dotlačili jednotlivé valce na seba. Operátor ešte raz skontroluje zariadenie a overí úplnosť všetkých krokov výrobného procesu, jeho postupu a spustí tlač.

Analyzovaný výrobný proces a jeho postup sú vhodné pre nasadenie do VR aplikácie. Na základe analyzovaného výrobného procesu a jeho postupu bude navrhnutá aplikácia virtuálnej reality, simulujúca vybraný proces a postup pre účel zaučovania zamestnancov.

Pri návrhu a tvorbe aplikácie budeme vychádzať zároveň z informácií, získaných na základe rozhovorov s predstaviteľmi a zamestnancami spoločnosti, fotodokumentácie a videodokumentácie, ktoré boli zrealizované z dôvodu ich nevyhnutnosti pre projekt.

4 NÁSTROJE NA TVORBU NAVRHNUTEJ VR APLIKÁCIE

Tvorba VR aplikácie si vyžaduje použitie viacerých nástrojov ako sú modelovací program na tvorbu a modelovanie 3D objektov, komponentov, prostredia. V našom prípade bude nástrojom Blender ver. 2.79. Textúrovanie vymodelovaných modelov bude kombináciou práce v modelovacom programe Blender a textúrovacom programe Substance Painter, plus zdroje z internetu. Nakoniec na samotnú tvorbu VR aplikácie budeme potrebovať nástroj na implementáciu vytvorených 3D modelov. Naším nástrojom bude game engine Unity 3D. Funkčnú stránku aplikácie budeme riešiť pomocou písania skriptov v programovacom jazyku C# v IDE Microsoft Visual Studio 2017, ktoré game engine Unity priamo podporuje a ponúka.

V nasledujúcej časti práce si stručne popíšeme jednotlivé nástroje, potrebné pre tvorbu našej VR aplikácie.

4.1 Blender ver. 2.79

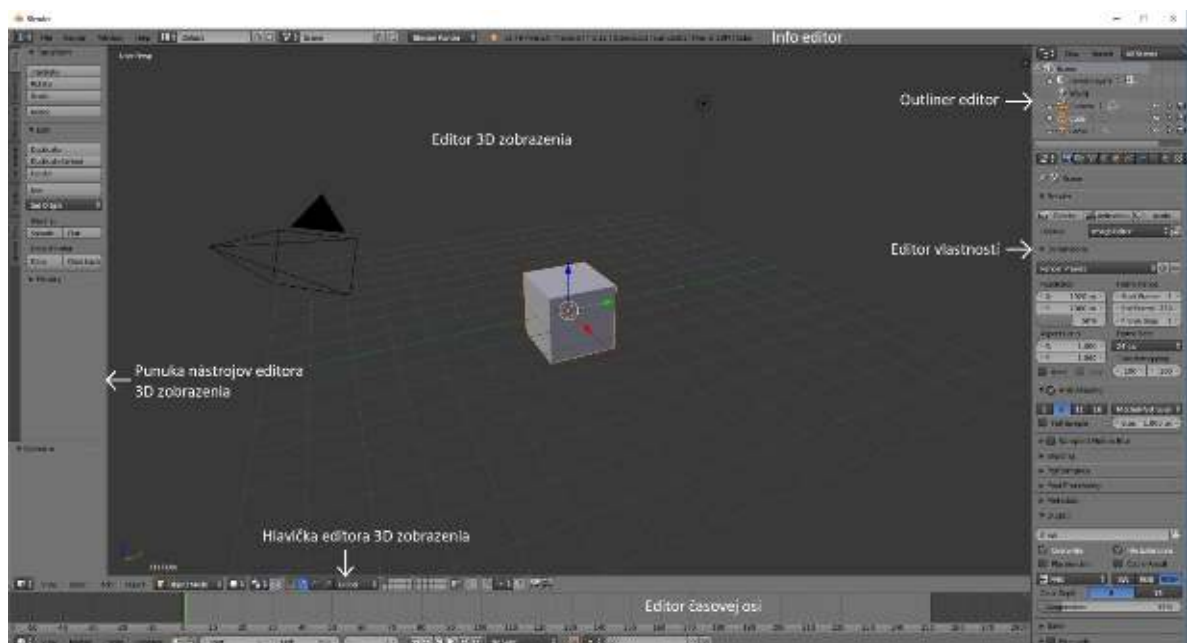
Nástroj na modelovanie je nevyhnutnou súčasťou pri tvorbe VR aplikácie, nakoľko celé prostredie, vrátane jednotlivých strojov, zariadení, súčiastok a komponentov musíme počítačovo vymodelovať do 3D podoby. Nástroj by mal ponúkať prehľadné a user friendly užívateľské rozhranie. Mal by byť rýchly s dobrou ponukou základných modelovacích primitív a ponukou funkcií, nástrojov a možností práce s grafikou, ktoré nám modelovanie uľahčia. Potrebné budú aj nástroje na úpravu a možnosti úprav vymodelovaných 3D modelov. Nástroj nemusí byť robustný, nakoľko pre modelovanie použijeme základné modelovacie techniky a niekoľko doplnkových nástrojov. Modelovací program by mal mať dobrú podporu UV unwrappingu a tvorby UV máp, ak možno automatizované generovanie UV máp. Nesmie chýbať tvorba základných textúr a materiálov. Nutnosťou je podpora programu pre viaceré formáty 3D modelov a možnosti ich exportu do formátov, ktoré podporujú vývojové prostredie. Keďže sa jedná o menší projekt, nástroj by nemal byť veľmi finančne náročný (predpokladáme minimálne resp. žiadne finančné prostriedky). V prípade rozšírenia projektu do budúcnosti o ďalšie 3D modely je potrebná okamžitá dostupnosť nástroja bez nutnosti ďalších investícií za nástroj.

Nástrojom, ktorý spĺňa všetky naše požiadavky, je aplikácia na prácu s grafikou a modelovanie Blender ver. 2.79. Blender je bezplatný open source grafický softvér pre počítačové modelovanie a animáciu, obsahujúci a ponúkajúci kľúčové vlastnosti ako mo-

delovanie, fotorealistické renderovanie, GPU renderovanie, nástroje animácie, realistické materiály, vizuálne efekty, charakter rigging, UV unwrapping, možnosť kompozície, simulácie, maskovanie, kamerové sledovanie, objektové sledovanie, tvorbu obsahov do 3D aplikácií a hier, tvorbu hier, úpravu videa, môže byť integrovaný s pipelined nástrojmi a ponúka prispôsobiteľné užívateľské rozhranie [3][4][40].

Voľba programu Blender ver. 2.79 pre náš projekt vývoja VR aplikácie je opodstatnená a viac ako vhodná. Program je komplexný, rozhranie je rozsiahle, kreatívne skúsenosti sú na vysokej úrovni.

Základný popis užívateľského rozhrania programu Blender ver. 2.79 a screenshot rozhrania prezentujeme prostredníctvom obrázku č. 11.



Obr. 11. Screenshot základného užívateľského rozhrania programu Blender.

Základné užívateľské rozhranie Blenderu sa skladá z niekoľkých neprekrývajúcich sa okien. Ich usporiadanie je prispôsobiteľné podľa našich potrieb. Rozhranie Blenderu môže pôsobiť mierne komplikovane, nakoľko integruje množstvo rôznych typov editorov, ktoré možno otvoriť súčasne v ich vlastných paneloch. Pri predvolenom zobrazení (obrázok č.11) si môžeme uvedomiť, že obsahuje päť rôznych editorov. Najzreteľnejším editorom je veľké 3D zobrazenie. Je to priestor, v ktorom môžeme prezerať, posúvať a organizovať objekty na našej Blender scéne. Ako každý editor, Blender môže mať viac panelov. V spodnej časti editora 3D zobrazenia sa nachádza hlavička editora. Hlavička je rad ponúk a nástrojov, ktoré poskytujú výbornú kontrolu nad editorom, vrátane selektorov zobrazení, režimov úprav, manipulátorov transformácií a manažmentu vrstiev. Na ľavej

strane editora je ponuka nástrojov, obsahujúca rôzne nástroje na úpravu, ktoré môžu byť aplikované na aktuálne vybraný objekt na scéne. Zobrazenie ponuky nástrojov je v réžii užívateľa. Editor má rovnako možnosť zobrazenia panela vlastností, ktorý poskytuje nastavenia vlastností pre aktuálne vybraný objekt. Ďalšími otvorenými editormi sú:

- 1) Info editor - má globálny charakter menu aplikácie a obsahuje informácie o aplikácii.
- 2) Editor časovej osi - je určený pre animácie.
- 3) Outliner editor - ponúkajúci hierarchický pohľad na všetky objekty na našej scéne.
- 4) Editor vlastností - je výkonný panel umožňujúci vidieť a upravovať veľké množstvo vlastností objektov na našej scéne [3][4][6].

4.1.1 Substance Painter ver. 2018.1.0

Substance Painter je 3D referenčná textúrovacia aplikácia, ktorá umožňuje textúrovať a vykreslovať 3D mriežky. Ponúka vysokú rýchlosť, flexibilitu a vizuálnu kvalitu. Používa inteligentné materiály, poskytuje automatické nasadenie textúr na modely a mnoho ďalších nástrojov a možností na tvorbu a nasadenie textúr. Je kompatibilný s game enginom Unity a ponúka možnosť konvertovania a exportu textúr do potrebných formátov a kvality [41].

Na textúrovanie využijeme nástroj Substance Painter ver. 2018.1.0, v prípade potreby ďalších textúr a materiálov zdroje z internetu.

4.2 Unity 3D ver. 2018.3.3f1 (64-bit)

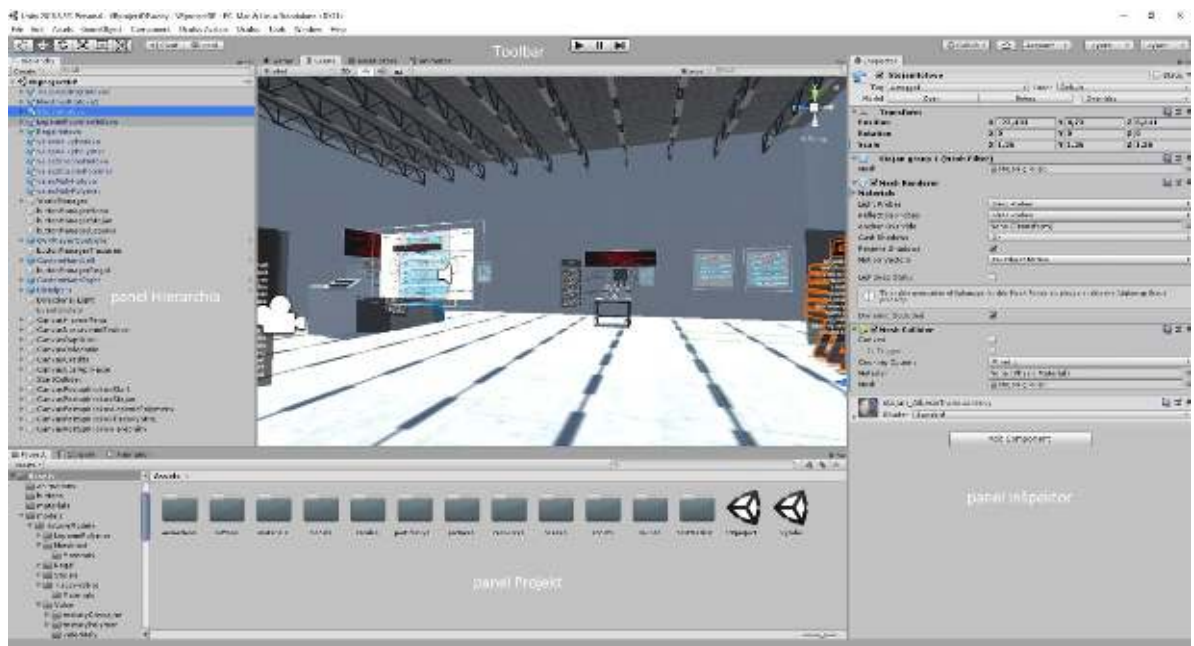
Ako už bolo spomenuté v úvode tejto kapitoly, pre vytvorenie funkčnej VR aplikácie, v našom prípade simulátoru výrobného procesu a jeho postupu, potrebujeme vývojové prostredie, ktoré nám umožní implementovať vytvorené 3D modely komponentov, súčastí a prostredia, samozrejme vrátane vytvorených a vygenerovaných textúr. Zároveň nám prostredie musí ponúkať podporu technológie virtuálnej reality, keďže vyvíjame VR aplikáciu. Výhodou vývojového prostredia je to, že podporuje čo najväčšie množstvo VR hardvéru, ktorý bude súčasťou nášho projektu. Prostredie by malo podporovať tvorbu skriptov slúžiacich na tvorbu logiky, interakcie a v podstate prostredníctvom skriptov by malo umožniť naprogramovať tak povediac čokoľvek, čo potrebujeme. Výhodou je, ak prostredie umožňuje písanie skriptov vo viacerých programovacích jazykoch, medzi ktorými by v prípade potreby nebol problémový prechod. Keďže chceme vyvíjať aplikáciu, ktorá nie je robustná a rozsiahla, ale je skôr menšieho rozsahu a nevyžaduje robustné nástroje na

tvorbu, môžeme zvoliť nástroj, ktorý je rýchly a spoľahlivý. Cross platform integráciou dosiahneme možnosť vybudovania a použiteľnosť aplikácie na rôzne platformy bez nutnosti prekódovania aplikácie.

Nástrojom pre vývoj aplikácie, ktorý spĺňa všetky naše požiadavky, je game engine Unity 3D vo verzii 2018.3.3f1 (64-bit). Profesionálne game enginey sa stali nástrojmi výbehu pre vývoj aplikácií virtuálnej reality z dôvodu blízkeho nasadenia medzi schopnosťami a možnosťami enginov a požiadavkami na tvorbu kvalitného VR. Game engine Unity3D je výborné riešenie pre tvorbu VR. Unity sa ukázalo ako uprednostňovaný nástroj pre vývojárov menších, stredných projektov a príležitostných vývojárov, vďaka svojej kombinácii sily a prístupnosti za skvelú cenu. Medzi najvýznamnejšie vlastnosti Unity3D patria:

- 1) Sila – runtime prehrávača Unity poskytuje mnoho dôležitých grafických funkcií, bohatý materiálny systém s fyzicky založeným renderovaním, osvetlením v reálnom čase, fyziky a skriptovania.
- 2) Rozšíriteľnosť – Unity je založený na modeli entít a komponentov, ktorý robí systém flexibilný, rozšíriteľný a konfigurovateľný prostredníctvom užívateľských skriptov. Môžu byť dokonca prepísané funkcie editora, čo poskytuje možnosť tvorby vlastných nástrojov na úpravu.
- 3) WYSIWYG editor – editor Unity sa ľahko používa, podporuje mnoho funkcií produktivity a pracovných tokov. Umožňuje import modelov a scén vytvorených prostredníctvom profesionálnych modelovacích nástrojov ako sú Blender, 3ds Max.
- 4) Portable – runtime prehrávača podporuje natívne desktop (OS X, Windows, Linux), populárne herné konzoly Xbox a PlayStation, mobilné operačné systémy vrátane iOS a Android, web prostredníctvom plugin prehrávača a experimentálnej podpory pre WebGL.
- 5) Cenová dostupnosť – bezplatná verzia Unity je plne funkčná bez obmedzení. Pre komerčné použitie spoločnosť ponúka primerané licenčné podmienky, ktoré zahŕňajú dostupný mesačný poplatok a skromný honorár (ale to len v prípade, ak sú z aplikácie poberané zisky).
- 6) Bohatý ekosystém – Unity má on-line Asset Store s nespočetným množstvom dostupných 3D modelov, animácií, balíkov hotových kódov a utilít.
- 7) Pripravenosť na VR – Oculus SDK je dodávaný s Unity integration balíkom, ktorý poskytuje podporu pre kľúčové programové konštrukty virtuálnej reality, čo uľahčuje prácu s VR [2][5].

Základný popis užívateľského rozhrania prostredia Unity3D ver. 2018.3.3f1(64-bit) a screenshot rozhrania možno vidieť na obrázku č. 12.



Obr. 12. Screenshot užívateľského rozhrania game engine Unity3D.

Zobrazenie hlavnej scény môžeme konfigurovať do štvorrozmerného rozloženia (ľavé, horné, pravé a perspektívne), ale môžeme použiť aj iné typy a schémy rozloženia. Panel Hierarchia umožňuje prehliadať objekty, ktoré sa nachádzajú v aktuálnej scéne. Na paneli Projekt vidíme a môžeme prehliadaním hierarchie zložiek a súborov zobrazovať prehľad všetkých aktuálnych súčastí v našom projekte (tieto sa môžu alebo nemusia načítať do aktuálnej scény). Panel Inšpektor poskytuje podrobné informácie o vlastnostiach a poskytuje možnosť úprav vlastností pre aktuálne vybraný objekt. Panel Toolbar nám poskytuje prepnutie na zobrazenie režimu Game, Scene, Asset Store alebo Animation, zároveň poskytuje možnosť spustenia, zastavenia behu aplikácie a ponúka nástroje na pohyb a navigáciu po scéne [2][5].

4.2.1 Microsoft Visual Studio 2017

Unity3D má podporu programovacích jazykov C# a JavaScript. Pre predmetný projekt použijeme programovací jazyk C#. Keďže OS, v ktorom našu VR aplikáciu budeme vyvíjať, je Microsoft Windows 10, budeme na tvorbu skriptov používať IDE Microsoft Visual Studio 2017, ktoré priamo podporuje jazyk C#. Inštalácia IDE MS Visual Studio je nám ponúknutá spoločne s inštaláciou game engine Unity [5].

II. PRAKTICKÁ ČASŤ

5 NÁVRH RIEŠENIA, MODELOVANIE, TEXTUROVANIE

V tejto kapitole práce predstavíme návrh samotného riešenia, ktorého súčasťou bude výber jednotlivých hardvérových a softvérových riešení, potrebných pre tvorbu a beh VR aplikácie. Dôraz budeme klásť na výber VR hardvéru, nakoľko výkon a kvalita VR sú podstatné pre konečnú VR aplikáciu a jej využívanie. Voľbu nástrojov odôvodníme. Po voľbe nástrojov a zariadení prejdeme k návrhu simulácie a postupu jej tvorby. Tým splníme časť návrhu riešenia a prejdeme k modelovaniu 3D prostredia, zariadení a komponentov na základe videodokumentácie a fotodokumentácie, ktoré sme v predmetnej spoločnosti vykonali. Ďalším krokom po vymodelovaní prostredia bude vytvorenie UV máp jednotlivých častí 3D modelov. Kapitulu ukončíme vytvorením, nasadením a exportom textúr do požadovaného formátu a kvality, rovnako aj exportom vytvorených 3D modelov.

5.1 Návrh riešenia

V časti návrh riešenia navrhne spôsob riešenia koncovej VR aplikácie vrátane výberu formy VR, VR hardvéru, nástrojov na tvorbu aplikácie a návrhu simulovaného výrobného procesu a postupu. Jednotlivé voľby odôvodníme.

5.1.1 Výber formy VR a zariadenia VR

Pre riešenie výslednej aplikácie s podporou virtuálnej reality, ktorá bude výstupom našej práce, sme sa rozhodli použiť desktopovú formu virtuálnej reality, a to z týchto dôvodov:

- Je najvýkonnejšia forma virtuálnej reality. Mobilná ani standalone forma virtuálnej reality, čo sa týka kvality a výkonu, na desktopovú formu nestačia.
- Do úvahy sme brali aj obstarávaciu cenu potrebných hardvérových komponentov na VR a výpočtového zariadenia s potrebným výkonom, ktorá je v prípade desktopovej formy VR väčšia (celá zostava vrátane PC a kompletnej sady zariadení VR je cca 1500 Euro). V našom prípade je veľkou výhodou fakt, že spoločnosť, pre ktorú VR aplikáciu vyvíjame pracuje s grafikou, pričom na prácu s grafikou sa používajú vysokovýkonné PC, ktoré majú dostatočný výkon na beh VR aplikácie. Tým odpadá nutnosť obstarania výkonného výpočtového zariadenia a náklady na beh VR sa znížia iba na cenu za VR hardvér (cca 600 Euro).
- Ďalším pozitívom, braným do úvahy pri rozhodovaní, je efektívnosť využitia hardvérových prostriedkov a fakt, že v prípade obstarania nového výpočtového za-

riadenia, či už výkonného PC alebo notebooku, môže byť toto zariadenie využité aj na iné účely, ako na beh virtuálnej reality.

- Jedným z hlavných dôvodov pri rozhodovaní bol typ a hlavný účel výslednej aplikácie. Tým je v prípade našej VR aplikácie simulácia výrobného procesu a jeho postupu, čo vyžaduje vyšší výkon, väčšiu mieru manipulácie a interakcie s objektmi virtuálneho prostredia a na úrovni, ktorú mobilná ani standalone forma neposkytujú.

Na základe výberu typu formy virtuálnej reality sme prešli k výberu hardvéru a komponentov VR, ktoré pre náš projekt použijeme.

Vývojárskych a výrobných spoločností, zaoberajúcich sa práve desktopovou formou technológie virtuálnej reality, je mnoho. Pre potrebu návrhu riešenia a z dôvodu využitia tejto formy VR zariadení v praktickej časti porovnáme produkty dvoch hlavných hráčov a konkurentov, ktorí ponúkajú najkvalitnejšie a najvýkonnejšie hardvérové prostriedky desktopovej VR. Po porovnaní odôvodníme našu voľbu s ohľadom na kladené požiadavky.

5.1.1.1 Porovnanie Oculus Rift a HTC Vive, návrh riešenia

Porovnanie zariadení sme vykonali na základe informácií získaných z literárnych zdrojov a webových portálov, medzi ktorými bol aj portál tom's Hardware, ktorý sa podrobne zaoberal rozborom oboch zariadení. Porovnaním zariadení zistíme, že majú veľa podobných vlastností, technológií a možností, alebo medzi ktorými sú minimálne rozdiely. Nájdu sa však aj použité rozdielne technológie. Hlavnými rozdielmi sú:

Tab. 3. Porovnanie VR zariadení HTC Vive a Oculus Rift vrátane príslušenstva [26][27][28][30].

Použitá technológia	Oculus Rift	HTC Vive
Veľkosť použitých displejov	90 mm x 2, 456 ppi pre každý displej	91,9 mm x 2, 447 ppi pre každý displej
FOV udávaný výrobcom	110V x 110Š stupňov	110V x 110Š stupňov
FOV na základe testov	94V x 93Š stupňov	110V x 113Š stupňov
Ideálna vzdialenosť oka od šošovky zariadenia	12 mm, vzdialenosť nastaviteľná	8 mm, vzdialenosť pevne daná
Použité šošovky	hybridné Fresnelové šošovky	klasické Fresnelové šošovky
Nastavenie objektívu IPD	58 – 72 mm	60,8 – 70,6 mm

Sledovanie polohy hlavy	gyroskop, akcelerometer a magnetomer, 44x IR LED (pokrytie celého headsetu)	gyroskop, akcelerometer, 32x fotodiódové senzory (predná strana headsetu)
Sledovacia technológia	konštalácia 6DOF infračervených optických kamier s 360 stupňovým sledovaním	na báze 360 stupňového 6DOF laserového sledovania prostredníctvom majákových sledovacích základní
Sledovanie vymedzeného priestoru	softvérové	softvérové, kamera
Zvuk zariadenia	3D priestorové audio slúchadlá, audio jack pre 3,5 mm konektor	žiadne, audio jack pre 3,5 mm konektor
Vstupné zariadenie	Oculus Touch, senzory na snímanie polohy ovládačov a prstov, konfigurovateľné vibrácie, tlačidlá, džojstik, trackpad	Vive Controllers, senzory na snímanie polohy ovládačov, vibrácie, tlačidlá, trackpad
Podpora priestorového VR	áno, room-scale VR, max. pracovný priestor 6 x 6 m	áno, room-scale VR, max. pracovný priestor 3 x 4 m
Minimálna Pamäť	8 GB	4 GB

Porovnaním sme zistili, že headset Oculus Rift, ako aj HTC Vive v sebe spájajú kombináciu kvalitných a výkonných technológií, ktoré sľubujú VR na vysokej úrovni. Použité displeje Riftu ponúkajú 456 ppi pre každý, čo je o niečo málo viac ako u Vive. Oba headsety na základe tvrdení výrobcov poskytujú FOV 110 stupňov, ale na základe testov portálu tom's Hardware má Rift priemerný FOV na úrovni 95 stupňov oproti 110 stupňov ponúkaných Vive. Tento deficit Riftu je však pre ľudské oko a vnímanie nepozorovateľný. Je to spôsobené aj skutočnosťou, že Rift používa kvalitnejšie hybridné Fresnelové šošovky, ktoré majú dôslednejšie zameranie cez zorné pole, aplikujú barelové skreslenie a zároveň kombináciou použitých technológií spôsobujú, že screen-door efekt je na displeji sotva zistiteľný. Na základe týchto technológií je obraz Riftu lepší ako u Vive. Rift ďalej ponúka možnosť nastavenia vzdialenosti oka od šošovky prostredníctvom voliteľnej rozpiery okuliarov, čo Vive neumožňuje. Sledovanie polohy hlavy a ovládačov je pri zariadeniach

od Oculus zabezpečené gyroskopom, akcelerometrom a magnetomerom v kombinácii so 44 IR LED, nachádzajúcich sa na celom headsete vrátane zadnej a vrchnej strany a ďalších senzorov, nachádzajúcich sa na ovládačoch. Sensory na Vive sú umiestnené len na prednej strane zariadenia. Vive oproti Riftu má okrem softvérového sledovania vymedzeného priestoru aj integrovanú kameru, ktorá pri prekročení vymedzeného priestoru vypne zobrazovaný VR obsah a zobrazí nám obraz z kamery. Rift ponúka 3D priestorové audio slúchadlá, poskytujúce kvalitný 3D zvuk. Rift má možnosť výmeny slúchadiel za vlastné. Vive žiadne vstavané slúchadlá nemá takže je nutné si pripojiť vlastné. Priestorová VR, a teda room-scale, je podporovaná na oboch zariadeniach s rozdielnym maximálnym rozsahom (priestorom) použitia. Rift ponúka maximálny rozsah (priestor) 6 x 6 m a Vive ponúka 3 x 4 m. Vive má minimálne požiadavky na pamäť výpočtového zariadenia 4 GB oproti 8 GB, ktoré požaduje Rift. Napriek tomu je pamäť 4 GB pre VR nedostatočná a slabý výkon je tak viditeľný (odporúčaná, nie minimálna pamäť pre Vive je 8GB) [26][27][28][30].

Na základe porovnania technológií zariadení Oculus Rift a HTC Vive hodnotíme, že Oculus Rift ponúka o niečo lepšie technológie a vlastnosti pre VR.

Keďže sú si headsety veľmi podobné, chceli sme napriek použitým technológiám a papierovým hodnotám vyskúšať oba headsety a zhodnotiť každý z nich na základe vlastnej skúsenosti.

Do úvahy sme brali pohodlie pri používaní, pričom pri Rifte a ovládačoch sme za celú dobu používania nepocíťovali nepohodlie a boli sme do VR obsahov plne ponorený. Pri Vive nám po dlhšom používaní začala prekážať hmotnosť zariadenia a jej pocit na hlave, čo nás miestami z úplného pohltenia vytrhlo. Obraz obidvoch zariadení bol kvalitný bez tvorby latencie, duchov a iných obrazových nedostatkov. Na Rifte bol z nášho pohľadu obraz zariadenia mierne kvalitnejší a neuvedomovali sme si tak screen-door efekt, ako pri Vive. Zvuk Riftu bol kvalitný, pri Vive záležal od použitých vlastných slúchadiel. Odozva a spolupráca zariadení pri behu VR obsahov od oboch výrobcov bola na vynikajúcej úrovni. Môžeme povedať, že reakcie na zmeny alebo interakciu boli vykonávané a renderované v reálnom čase. Ovládače Oculus Touch nás prekvapili a presvedčili, že Oculus je správna voľba. Keďže chceme vytvárať VR aplikáciu na simulovanie výrobného procesu a jeho postupu a aplikácia bude vyžadovať vysokú mieru interakcie, sú ovládače od Oculusu výbornou voľbou. Vhodne „padnú“ do ruky, ovládanie je intuitívne, snímanie polohy prstov je výborná vlastnosť. Vibračné odozvy boli na vysokej úrovni. Ovládače od HTC nám pri-

šli trochu ťažkopádne. V tomto smere boli pre nás jednoznačným „vítázom“ Oculus Touch. Po vyskúšaní oboch headsetov aj s komponentmi hodnotíme, že pre nás boli zariadenia Oculus Rift, Oculus Touch a senzory na snímanie polohy vyhovujúcejšie.

Do úvahy sme brali aj obstarávaciu cenu za VR zariadenia. Cena Oculus Rift, Oculus Touch a senzorov je približne 560 Euro. Pridaním bezdrôtovej jednotky a napájania od spoločnosti TPCast nás celá zostava bude stáť približne 670 Euro. Cena HTC Vive, Vive Controllers a senzorov je približne 630 Euro, po pridaní bezdrôtovej jednotky s napájaním od TPCast zaplatíme približne 740 Euro.

Na základe predchádzajúceho rozboru a odskúšaní VR zariadení Oculus Rift s príslušenstvom a HTC Vive s príslušenstvom sme dospeli k rozhodnutiu, že VR zariadenia od Oculus sú pre náš projekt správnu voľbou.

5.1.2 Výber nástroja pre modelovanie 3D

Zvoleným nástrojom pre modelovanie je aplikácia Blender ver. 2.79. Voľbu nástroja sme hodnotili na základe viacerých hľadísk, ktorými sú:

- 1) Splnil všetky naše požiadavky na požadovaný nástroj (požiadavky sme definovali pri popise nástrojov na tvorbu).
- 2) Voľba nástroja pre modelovanie 3D bola konzultovaná s odborníkmi z oblasti vývoja VR aplikácií (nakoľko sa nejedná o rozsiahly projekt s veľkým množstvom 3D modelov, bolo odporúčané riešenie pre náš projekt Blender).
- 3) Odporúčanie na základe odborných článkov [40][42].
- 4) Faktorom, braným do úvahy pri voľbe nástroja, bola aj predchádzajúca skúsenosť s 3D modelovaním v aplikácií Blender.

5.1.3 Výber nástroja na textúrovanie

Zvoleným nástrojom na textúrovanie je aplikácia Substance Painter ver. 2018.3.2. Nástroj nám odporučili a poskytli (vrátane licencie a potrebných materiálov) odborníci z oblasti vývoja VR aplikácií.

5.1.4 Výber nástroja pre tvorbu VR aplikácie

Zvoleným nástrojom pre tvorbu samotnej VR aplikácie je herný engine Unity 3D ver. 2018.3.3f1 (64-bit). Voľbu nástroja sme hodnotili na základe viacerých hľadísk, ktorými sú:

- 1) Splnil všetky naše požiadavky na požadovaný nástroj (požiadavky sme definovali pri popise nástrojov na tvorbu).
- 2) Voľba nástroja pre tvorbu VR aplikácie bola konzultovaná s odborníkmi z oblasti vývoja VR aplikácií (nakoľko sa nejedná o rozsiahly projekt, ale o menší projekt VR, bolo odporučené riešenie pre tvorbu aplikácie herný engine Unity).
- 3) Odporúčanie na základe publikácií a odborných článkov [2][6][43].
- 4) Faktorom, braným do úvahy pri voľbe nástroja, bola aj predchádzajúca skúsenosť s vývojom aplikácií v hernom engine Unity 3D.

5.1.5 Výber IDE a programovacieho jazyka pre tvorbu skriptov

Vybrali sme si herný engine Unity, keďže priamo podporuje písanie skriptov v programovacích jazykoch C# a JavaScript. Pre náš projekt sme si zvolili jazyk C#. Dôvodom sú predchádzajúce skúsenosti s programovaním v jazyku C#. Našu VR aplikáciu vyvíjame prostredníctvom OS Windows, a preto sme pre tvorbu skriptov zvolili IDE Microsoft Visual Studio 2017, ktoré ponúka Unity pri inštalácii herného enginu.

Voľbou vymenovaných VR zariadení, komponentov a nástrojov pre tvorbu máme všetky potrebné súčasti na vytvorenie VR aplikácie, simulujúcej výrobný proces a postup vybranej výrobnéj spoločnosti. Posledným krokom pred vývojom aplikácie je návrh samotnej simulácie a postup jej tvorby.

5.1.6 Návrh simulácie výrobného procesu a postup tvorby VR aplikácie

Simulácia výrobného procesu a jeho postupu bude vykonaná na základe vykonanej analýzy a rozboru zvoleného výrobného procesu a postupu. Vytvorená simulácia nebude 100 % kópiou skutočného výrobného procesu, nakoľko niektoré procesy a postupy sú len ťažko a ak vôbec simulovateľné. Takéto procesy a postupy nebudú z aplikácie vynechané z dôvodu zachovania dôveryhodnosti a reálnosti, ale bude zvolená forma simulácie procesu a postupu, ktorá najviac približuje ich komplikované vykonanie. Takýchto procesov bude v našej VR aplikácií minimum. Predmetné procesy a postupy s návrhom ich riešenia sú:

- 1) Stlačenie pedálu na uzavretie tlačového valca medzi rotačnými ramenami piestov na stroji určenom pre lepenie tlačového polyméru. Na interakciu s VR prostredím, jeho prvkami a objektmi používame ovládače a ruky, stlačenie pedálu nohou je v našej aplikácií nemožné nasimulovať. Riešením bude stlačenie tlačidla na ovládacom paneli zariadenia rukou, pričom tlačidlo bude plniť úlohu pedálu.

- 2) Lepenie polyméru na tlačový valec. Je veľmi komplikované nasimulovať prilepenie úvodnej vrchnej hrany polyméru, ktorý je z mäkkých umelých materiálov na povrch tlačového valca. Preto je tento proces v našej aplikácii zakomponovaný a automatizovaný v doleповacom procese prostredníctvom formy s polymérom, ktorý je aktivovaný na základe stlačenia tlačidla na ovládacom paneli zariadenia, ktoré slúži na nalepenie polyméru. Z tohto dôvodu je vynechaný krok kontroly vhodnosti nalepenia polyméru prostredníctvom zobrazenia elektronických mikroskopov na displeji zariadenia. Pri simulácii ide o dodržanie postupu krokov pre úspešné nalepenie polyméru, a to je v aplikácii dodržané a zachované. Ak je v simulácii postup dodržaný, polymér je úspešne nalepený.

Ostatná časť analyzovaného výrobného procesu a jeho postupu bude v našej VR aplikácii dodržaná a zachovaná.

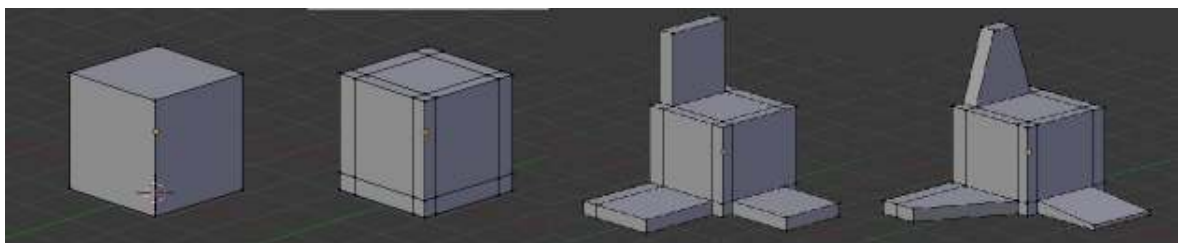
Postup tvorby navrhutej VR aplikácie, simulujúcej vybraný výrobný proces bude spočívať v tvorbe, modelovaní 3D prostredia, jeho súčastí a komponentov, medzi ktoré patria jednotlivé výrobné zariadenia, stroje, súčastky, hala a všetky prvky potrebné pre simuláciu výroby. Jednotlivé stroje, súčasti a komponenty budú modelované na základe videodokumentácie a fotodokumentácie (referenčné foto a video prostredia a zariadení), ktoré boli vo vybranej spoločnosti pre náš účel vykonané. Po vymodelovaní všetkých 3D objektov prejdeme k tvorbe UV máp jednotlivých modelov a ich častí. Po vytvorení UV máp vytvoríme alebo upravíme existujúce materiály a textúry. Textúry vytvoríme a nasadíme na základe vytvorených UV máp. Ako vzor pri textúrovaní nám poslúžia referenčné foto a video materiály rovnako, ako aj pri modelovaní. Vytvorené modely a textúry z jednotlivých nástrojov exportujeme do požadovaných formátov a v prípade textúr do požadovanej kvality. Následne môžeme importovať vytvorené a exportované modely, textúry a materiály do vývojového prostredia a začať s tvorbou VR aplikácie. Tvorba aplikácie bude obnášať implementovanie vytvorených 3D modelov, textúr a materiálov, implementovanie balíka od spoločnosti Oculus, pripraveného pre podporu zariadení VR a ponúkajúceho základné skripty, riešiacie lowlevel programovanie pre základnú funkčnosť headsetu, ovládačov a senzorov. Potrebná bude tvorba rôznych objektov vrátane UI prvkov, zvukových objektov a ďalších potrebných objektov pre zabezpečenie čo naj dôveryhodnejšej simulácie výrobného procesu. Najdôležitejšie bude programovanie, a teda tvorba skriptov, prostredníctvom ktorých vytvoríme interakciu, všetky simulačné procesy a celú

aplikačnú logiku pre našu VR aplikáciu. Hotovú VR aplikáciu zostavíme a vybudujeme pre platformy Windows, Mac, Linux. Otestujeme funkčnosť vyvinutej aplikácie.

5.2 Tvorba, úprava a export 3D modelov prostredia

5.2.1 Modelovanie, príprava

Pri modelovaní sme sa snažili držať a riadiť základnými pravidlami pre správne modelovanie, vzhľadom na typ a použitie jednotlivých 3D modelov v aplikácií. Na modelovanie sme používali spôsob tzv. box modelingu, kedy sme začínali modelovať z obyčajnej kocky alebo valca (podľa modelu a potreby), ktoré sme postupne rozmodelovali použitím pridávania edge loopov, rozšírení oblasti, transformačných metód a nástrojov ako preklad, škálovanie a rotácia vrcholov, hrán a tvárí, ale aj ďalších techník a metód podľa potreby (obrázok č.13). V niektorých prípadoch sme použili aj pridanie kruhu alebo plochy, ktoré sme následne použitím vymenovaných techník a ďalších modelovacích metód a techník rozmodelovali do požadovaného tvaru a stavu. Pred začatím modelovania bolo dôležitým faktorom stanoviť si topológiu modelovaných objektov. Príkladom topológií pri modelovaní je použitie trianglov (trojstranový polygón), quadov (štvorstranový polygón) alebo N-gonov (polygónov, ktoré majú viac ako štyri strany). Tá sa stanovuje v závislosti na účele modelu, pričom účelom modelu rozumieme, či bude model využívaný k animácií/deformácií (preferuje sa použitie quadov), alebo sa bude jednať o statický model (možnosť použitia trianglov). Používaniu N-gonov sa vyhýbame, nakoľko môžu v budúcnosti spôsobovať technické komplikácie, napríklad pri exportovaní z jedného softvéru do druhého, alebo pri subdividingu (rozdeľovaní). Keďže budeme modely a ich časti využívať na interakciu, animácie a transformácie ich jednotlivých častí na základe vykonanej interakcie užívateľa, pri modelovaní sme využívali topológiu quadov [3][4][7][44].



Obr. 13. Metóda modelovania box modeling.

Pre vymodelovanie všetkých 3D modelov v našej aplikácií sme používali modelovacie techniky, nástroje a spôsoby, ktorých použitie sme v prípade nevedomosti a potreby

čerpali z publikácií a manuálov určených pre nami zvolený modelovací nástroj Blender ver. 2.79 [3][4][7].

Pre modelovanie 3D modelov prostredia a jeho komponentov sme použili aplikáciu Blender ver.2.79. Po spustení aplikácie nás uvítalo základné užívateľské rozhranie s úvodným oknom, ktoré nám ponúkalo možnosť voľby interakcie (Blender, 3ds Max, Maya). Ponechali sme primárne nastavený Blender a okno sme zavreli. Tým sme sa v podstate dostali k tvorbe samotného 3D modelu. Primárne nastavenie UI Blenderu nám vyhovovalo. Do zobrazenia sme si pridali stlačením klávesy *T* ponuku nástrojov editora 3D zobrazenia, ktorý nám ponúkal sériu upravovacích nástrojov ako napríklad transformácie, preklad, rotácia, škálovanie alebo pridávanie rozšírenia, tvárí, duplikovanie, ďalšie delenie častí objektu a mnoho ďalších nástrojov, ktoré sme pri modelovaní našich 3D objektov používali, a ktoré boli aplikovateľné pre aktuálne zvolený objekt na našej scéne. Pridali sme si aj panel vlastností editora 3D zobrazenia stlačením klávesy *N*, ktorý nám ponúkal nastavenie vlastností aktuálne zvoleného objektu. Tým sme ukončili nastavenie nášho UI. Na scéne sme mali vložené tri základné objekty, ktoré sme pomocou pravého tlačidla myši postupne označili a odstránili stlačením klávesy *X*, voľba možnosti delete. Tým sme si vyčistili a pripravili modelovací priestor (scénu) na tvorbu nášho 3D modelu.

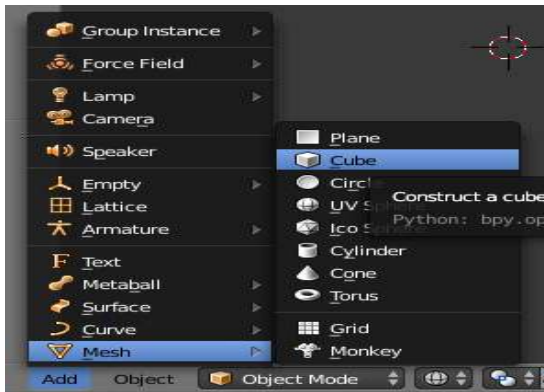
Pred začatím modelovania sme si ešte zmenili zobrazenie scény z perspektívneho (Persp) na ortografický (kolmé premietanie referenčnej sféry do roviny, Ortho) stlačením klávesy *numpad5*.

Pri modelovaní sme používali klávesové skratky, ktoré nám umožňovali vyvolať jednotlivé ponuky, možnosti a nástroje bez nutnosti ich hľadania na paneloch a preklikávania sa medzi nimi. Tento postup nám modelovanie rapídne urýchlil. V našej práci vysvetlíme skratky a budeme ich aj používať.

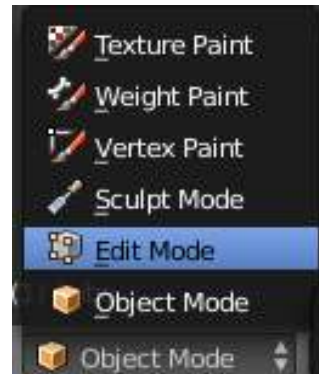
5.2.2 Tvorba 3D modelu

Začali sme pridaním (Add, klávesová skratka *Shift+A*) nového základného objektu kocka (Cube) typu mriežka (Mesh) (obrázok č. 14). Pridaný objekt sme si označili a prepli sme sa z režimu objektového módu (Object Mode) do editačného módu (Edit Mode) voľbou mož-

nosti Edit Mode v ponuke hlavičky 3D editora alebo klávesou *Tab* (obrázok č. 15).

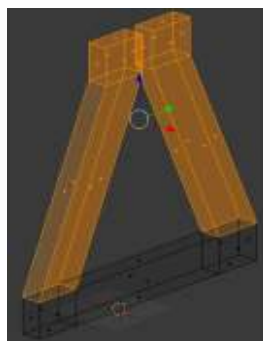


Obr. 14. Pridanie nového objektu kocka.



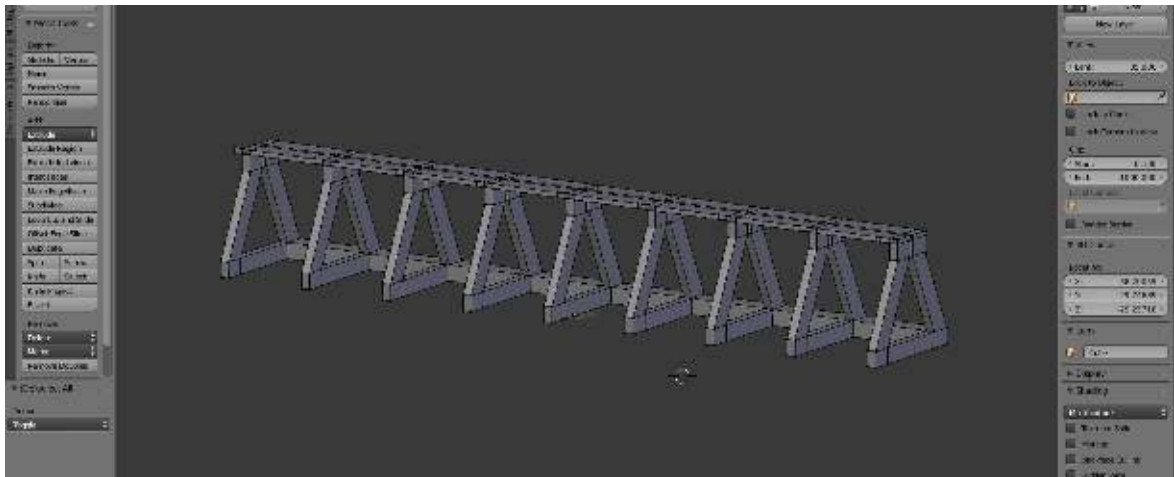
Obr. 15. Prechod do edit módu.

Stlačením klávesy *Z* sme skryli zobrazenie plôch objektu, aby sme mohli vidieť jednotlivé vrcholy, hrany alebo tváre (Vertex, Edge, Face). Pre voľbu práce s vrcholmi, hranami alebo tvármi sme zvolili možnosť výber módu mriežky (Mesh Select Mode, klávesová skratka *Ctrl+Tab*). Zvolili sme prácu s vrcholmi (Vertex). Zrušenie označenia alebo označenie celého objektu sme prepínali pomocou klávesy *A*. Pre označenie určitej oblasti, časti objektu sme použili klávesu *B* (označenie oblasti myšou) alebo klávesu *C* (kruhovú oblasť označovanú myšou). Označili sme si štyri bočné vrcholy objektu kocky a použili sme transformáciu preklad (Translate, kláves *G*) v osi X (kláves *X* po voľbe nástroja) z ponuky nástrojov. Zúžili sme kocku na polovičnú veľkosť. Označili sme si zadné vrcholy kvádra, použili transformáciu preklad (kláves *G*, kláves *Y*) a predĺžili sme dĺžku kvádra v osi Y. Po transformácii sme ponechali označené vrcholy, ktoré sme prekladali. Zvolili sme nástroj rozšíriť oblasť (Extrude Region, kláves *E*), prostredníctvom ktorého sme rozšírili objekt o ďalšiu časť rovnakého tvaru a veľkosti, ale rozdielnej dĺžky v osi Y. Po potvrdení sme rovnakú akciu vykonali ešte raz, pričom sme vytvorili ďalšiu rovnako dlhú časť, ako prvá vytvorená časť (kváder). Vytvorili sme objekt skladajúci sa z troch častí (obrázok č. 16, nevysvietená spodná časť objektu).



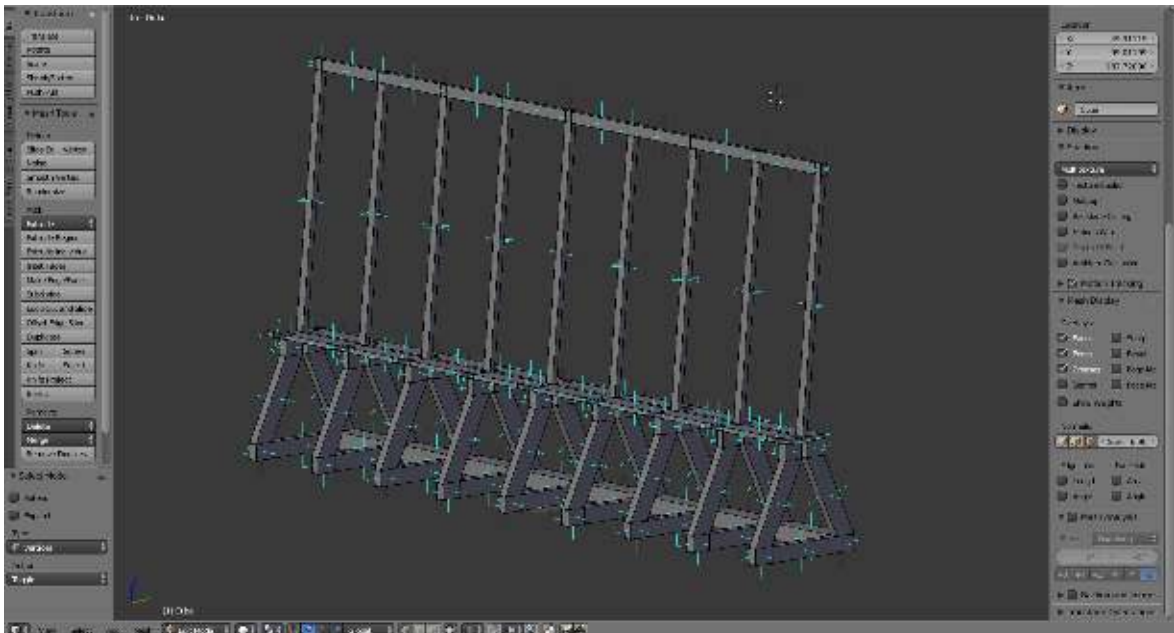
Obr. 16. Vymodelovaná úvodná časť nohy stojana 3D modelu.

Prepli sme na prácu s tvármi (*Ctrl+Tab*, voľba Face). Označili sme si dve vrchné tváre vytvorených krátkych častí objektu. Zvolili sme nástroj rozšíriť oblasť (kláves *E*), ktorým sme objekt rozšírili o ďalšie kvádre. Kláves *A*, zrušili sme označenie častí objektu a označili sme si jednu z vytvorených vrchných tvári. Prepli sme si zobrazenie na pohľad z boku (*numpad3*). Použili sme transformáciu preklad v osi Y (kláves *G*, kláves *Y*) a presunuli sme vrchnú časť tváre bližšie ku stredu vytvoreného objektu, čo zároveň spôsobilo aj posun hrán patriacich k zvolenej tvári v rovnakom smere. Rovnakú transformáciu sme vykonali aj na vrchnej tvári druhého rozšírenia. Zvoli sme vrchné tváre oboch vytvorených kvádrov a nástrojom rozšíriť oblasť v osi Z (klávesa *E*, klávesa *Z*) sme vytvorili ďalšie rozšírenia objektu (obrázok č.16, vyznačená oblasť). Ďalším krokom bola zmena práce s vrcholmi (*Ctrl+Tab*, Vertex). Označili sme si dva vrchné vnútorné vrcholy každého z najvyššie položených kvádrov a voľbou nástroja doplniť tvár (Add Face, klávesa *F*) sme medzi vytvorenými hranolmi vytvorili tvár, ktorá obe časti spájala. Rovnakým postupom ako pri modelovaní prvej časti nohy stojana sme postupovali pri domodelovaní jej zvyšku. Keď sme mali nohu stojana vymodelovanú, označili sme si celý vymodelovaný objekt (kláves *A*) a použili sme možnosť špeciálneho nástroja odstrániť zdvojenia (Remove Doubles), ktoré sa vytvárajú automaticky pri modelovaní a postupne by mohli spôsobovať problém pri ďalšom modelovaní. Zvolenú súčasť sme mali čistú a prešli sme k použitiu nástroja pridať duplicitu (Add Duplicate, *Shift+D*), ktorý zdublikuje zvolenú mriežku a umožní nám ju umiestniť podľa potreby. Zmenili sme zobrazenie na pohľad z prednej strany (*numpad1*), pridať duplicitu (*Shift+D*), umiestni v ose X (kláves *X*) posunutú o hodnotu 43 v X a potvrdenie umiestnenia. Takýchto duplicit sme rovnakým spôsobom vytvorili 8 pričom polovica bola od predchádzajúcej posunutá v osi X o hodnotu približne 43 a druhá polovica v osi X o hodnotu približne -43. Chceli sme zachovať prvú vymodelovanú časť na pôvodnom mieste. Prepli sme si na prácu s vrcholmi (*Ctrl+Tab*, Vertex). Prešli sme k domodelovaniu spodnej časti nôh stojana, pričom sme išli pracovať so zadnými spodnými kvádrami a ich vrcholmi. Označili sme si vždy štyri vnútorné vrcholy susedných dvojíc nôh stojana a použili sme špeciálny nástroj premostiť hrany slučiek (kláves *W*, Bridge Edge Loops), ktorý vytvorí premostenie medzi dvoma označenými slučkami vrcholov. Pri použití nástroja treba dať pozor, aby bol počet vrcholov každej slučky rovnaký. Rovnakým spôsobom sme vytvorili premostenia medzi jednotlivými časťami (nohami) stojana vrátane vrchných vnútorných častí nôh stojana, a tým sme vymodelovali spodnú stojanovú základňu (obrázok č.17).



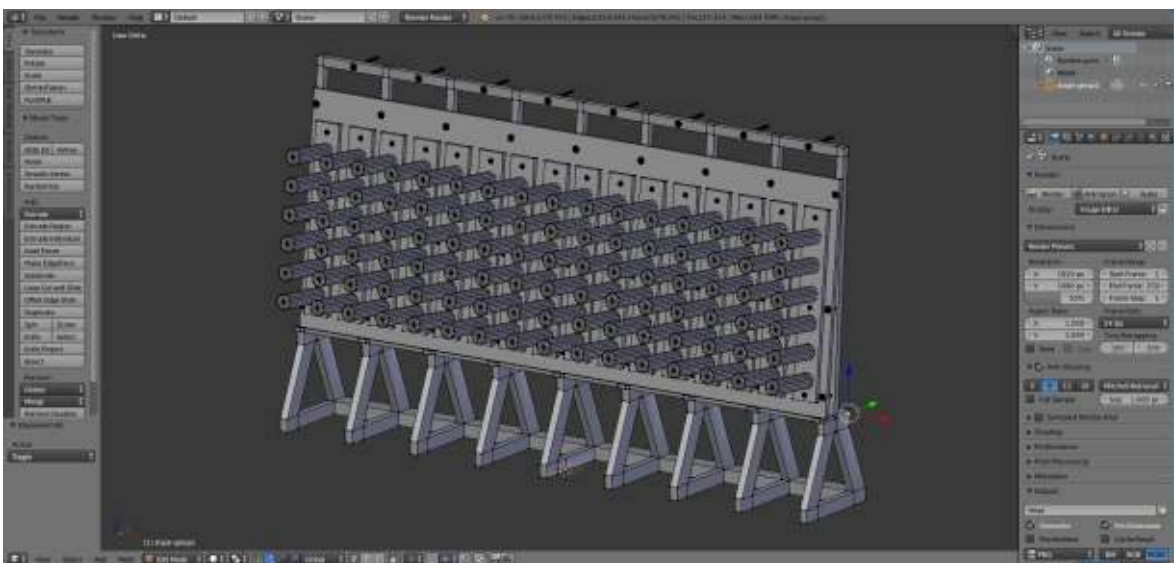
Obr. 17. Vymodelovaná spodná základňa stojana na tlačové valce.

Prepli sme na prácu s tvármi (*Ctrl+Tab*, *Face*) a spolu so stlačenou klávesou *Shift* sme postupne označovali vrchné tváre stredných častí jednotlivých nôh, nachádzajúcich sa na vrchných častiach stojanovej základne. Voľbou nástroja rozšíriť oblasť (kláves *E*) v osi *Z* (kláves *Z*) sme rozšírili spodnú časť základne stojana o stĺpiky (kvádre) smerujúce do vrchnej časti stojana. Zmenili sme zobrazenie na pohľad zo strany (*numpad3*) a na stále označené tváre vrchných strán pridaných kvádrov sme aplikovali transformáciu preklad (kláves *G*) v osi *Y* (kláves *Y*) a objekty aj s hranami sme mierne posunuli, aby boli v naklonenej polohe. Na stále označené vrchné tváre stĺpikov sme ešte raz použili nástroj rozšíriť oblasť (kláves *E*) v osi *Z* (kláves *Z*) a rozšírili sme ich o koncové vrchné časti jednotlivých stĺpikov. Prepli sme na prácu s vrcholmi (*Ctrl+Tab*, *Vertex*) a postupne sme označovali vnútorné slučky hrán susedných koncových častí stĺpikov, aby sme na ne použili špeciálny nástroj premostiť hrany slučiek (kláves *W*, *Bridge Edge Loops*) a vytvorili tak prepojenie medzi jednotlivými stĺpikmi stojana, ako to bolo aj v prípade spodnej základne stojana. Po vytvorení prepojenia sme si označili celý model (kláves *A*) a použili nástroj odstrániť zdvojenia, ktorý nám odstránil vytvorené zdvojenia. Pri kontrole modelu sme si všimli, že niektoré tváre majú opačne prepočítané normály (vonkajšie strany tvári sú otočené do vnútornej strany). Na opravu zisteného problému sme si preto zobrazili normály tvári celého modelu, kde sa nám naše podozrenie potvrdilo. Na odstránenie chyby sme použili nástroj Mriežky Normály a zvolili sme možnosť prepočítať normály do vonkajšej strany (*Recalculate Outside*, *Ctrl+N*). Použitie nástroja sme overili kontrolou správneho smeru jednotlivých normál modelu (smer normál je zobrazený modrými čiarami a smerujú od častí modelu smerom von) (obrázok č.18). Tým sme mali opravenú a hotovú časť celej konštrukcie modelu stojana pre tlačové valce (obrázok č.18).



Obr. 18. Vymodelovaná konštrukcia stojana pre tlačové valce so zobrazením normál.

Použitím rovnakých a ďalších modelovacích postupov, metód a techník ako pri modelovaní konštrukcie stojanu pre tlačové valce sme vymodelovali podkladovú dosku stojanu, držiaky pre tlačové valce s podkladovými doskami a skrutky. Vymenované časti sme modelovali postupne ako samostatné objekty, pričom po vymodelovaní sme ich priradili k hlavnému objektu stojana (pre priradenie sme museli prejsť do objektového módu (kláves *Tab*), označiť požadované časti (*Shift*+pravé tlačidlo myši) a nástrojom spojiť (*Join*, *Ctrl+J*) sme samostatné vymodelované časti spojili). Týmto spôsobom sme vytvorili kompletný 3D model stojana pre tlačové valce (obrázok č.19).



Obr. 19. 3D model stojana pre tlačové valce.

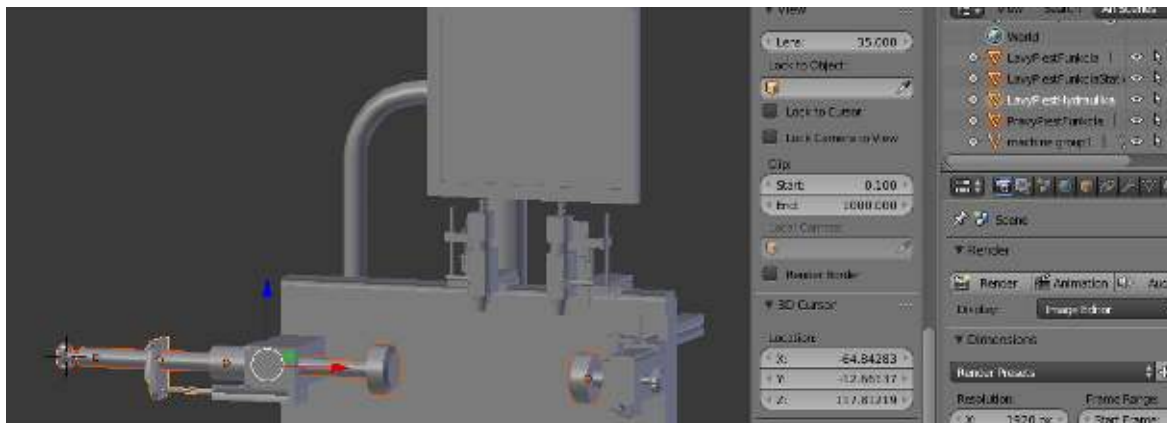
Pre modelovanie všetkých 3D modelov pre náš projekt sme používali rovnaké a ďalšie modelovacie postupy, metódy a techniky ako pri modelovaní stojanu pre tlačové valce. V prípade potreby doplnenia vedomosti sme tieto čerpali z publikácií a manuálov.

5.2.3 Úprava vytvorených 3D modelov

Pri modelovaní jednotlivých 3D modelov a ich častí sme sa snažili pristupovať od začiatku tak, aby obsahovali čo najmenšie množstvo chýb a nepresností, samozrejme v rámci našich modelovacích schopností. To sme sa snažili dosiahnuť, ako bolo možno vidieť aj pri tvorbe modelu stojana na tlačové valce, priebežným používaním nástrojov a techník na úpravu chýb a nepresností z ponuky nástrojov na čistenie ako napríklad odstránenie duplicit, vymazanie stratených a neprepojených vrcholov, hrán a tvárí, zaplnenie otvorov, vytvorenie a doplnenie chýbajúcich prepojení alebo používaním nástrojov na prácu s normálami, ako prepočítanie normál do nami požadovaného smeru. V prípade zle definovaných normál by sme mohli mať problémy po implementácii vytvorených 3D modelov do vývojového prostredia, kde by sa nám zle prepočítané normály zobrazovali/nezobrazovali (opačne prepočítané normály by boli prázdny, chýbajúcim miestom alebo časťou modelu). V prípade našej nevedomosti a potreby využitia upravovacích a opravovacích metód a techník sme čerpali vedomosti z publikácií a manuálov [3][4][7].

V prípade modelovania našich 3D modelov je potrebné dodať, že nakoľko sa jedná o simuláciu výrobného procesu a postupu, budú v aplikácii vytvárané rôzne interakcie užívateľa s prostredím, súčiastkami a komponentmi, pričom samotná simulácia bude spočívať v obsluhu jednotlivých strojov, zariadení a súčastí. Interaktívne časti strojov sme museli v jednotlivých modeloch oddeliť a definovať ako samostatné časti (objekty). To sme vykonávali označením potrebnej časti modelu a jej následným oddelením od celku použitím nástroja oddeliť (kláves *P*, *Separate Selection*). Oddelené časti sme si z dôvodu prehľadnosti nazvali podľa ich charakteristiky a využitia v našej VR aplikácii. Príkladom môže byť, vytvorenie samostatných častí stroja pri 3D modeli zariadenia, slúžiaceho na lepenie tlačového polyméru (obrázok č.20).

V prípade vyselektovaných častí, ktorými boli jednotlivé valce zariadení, sme museli ešte nastaviť pivotov do stredu objektov (*Set Origins*, *Shift+Ctrl+Alt+C*, voľba možnosti *Origin to Center of Mass Volume*). Bolo to nevyhnutné, pretože keď budeme v konečnej aplikácii programovať transformáciu rotácie, budeme objekty rotovať okolo ich vlastných pivotov.



Obr. 20. Oddelenie funkčných častí modelu zariadenia do samostatných objektov.

Oddelenie potrebných funkčných častí sme vykonali ako posledný krok pred exportom modelov do potrebného formátu a ich importom do vývojového prostredia (oddeleniu častí predchádzala tvorba UV máp modelov). Po vykonaní všetkých potrebných úprav a opráv jednotlivých 3D modelov sme mohli prejsť k tvorbe UV máp a textúrovaniu.

5.3 Texturovanie vytvorených 3D modelov

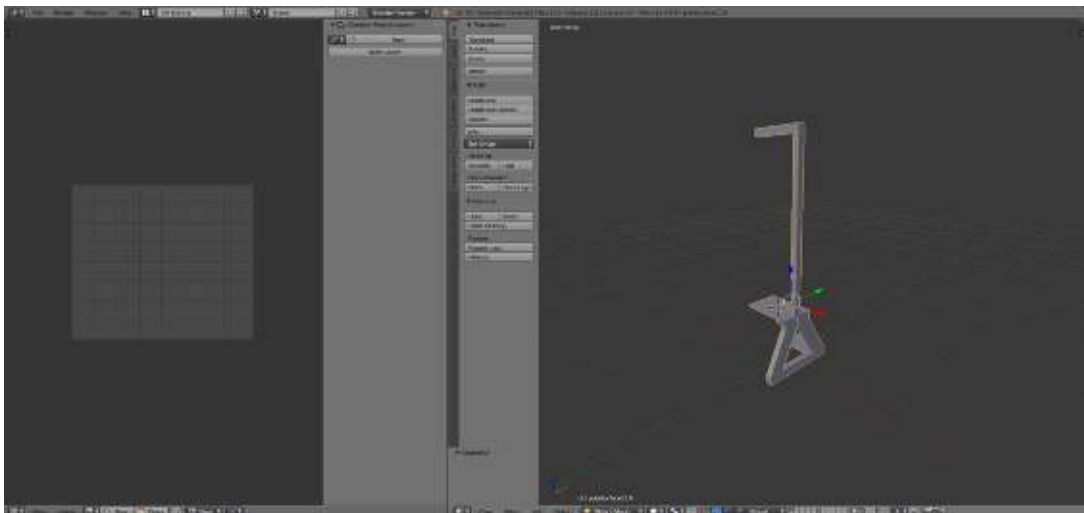
5.3.1 Tvorba UV máp

Samotnému textúrovaniu predchádzala tvorba UV máp pre jednotlivé modely a ich časti. UV mapovanie nám definuje spôsob zobrazenia akéhokoľvek obrazového súboru, ktorý chceme zobrazit' na aktuálnom 3D povrchu. UV mapovanie je proces, pri ktorom akoby prevádzame 3D model alebo jeho označené a špecifikované časti do 2D podoby za účelom textúrovania. Poznáme viac spôsobov, ktoré nám umožňujú tvorbu UV máp. Pre náš projekt sme použili spôsob formou vyselektovania (označenia) požadovanej časti spôsobom vyznačovania, určovania UV seamov (miesta, kde dôjde k rozloženiu modelu) a následným unwrappingom, ktorý nám nami definovanú časť modelu rozložil do požadovaného 2D rozloženia (vytvoril tzv. UV shell). 2D rozloženie nám zobrazil na UV Gride (UV mriežke), slúžiacej na tvorbu a prácu s UV mapami. Tiež sme použili spôsob, pri ktorom sme požadované časti oddelili do samostatných častí (objektov), oddelené sme si ich označili, definovali seamy (Mark Seam) a následne Unwrap (postup robený z dôvodu lepšej prehľadnosti pri vyznačovaní seamov a tvorbe UV shellov jednotlivých častí), alebo sme využili nástroj na automatické vygenerovanie UV mapy vyznačenej časti, objektu (Smart UV Project), ktorý nám automaticky UV mapu vygeneroval. Rozdielom v použití automatickej mapy oproti určovaní seamov a definovaní shellov ručne bol v tom, že nástroj na automatickú tvorbu niekedy zle vyznačil seam, takže po vygenerovaní UV mapy sme si

ich museli skontrolovať. Pri použití metódy oddelenia do samostatných častí, sme po vytvorení UV mapy časti objektu tento napäť spojili (pridali) k pôvodnému modelu (vytvorená UV mapa sa automaticky preniesla s ním a pridala sa do hlavnej UV mriežky). Pri tvorbe UV máp, aby sme ušetrili množstvo práce a času, sme využili to, že môžeme vytvoriť UV mapu len jednej časti modelu, ktorá sa vo výslednom modeli opakuje, a túto mapu aplikovať aj na všetky ostatné rovnaké (duplicitné) časti modelu. Dôležitou časťou UV mapovania bolo dobré rozloženie UV shellov v priestore UV mriežky a to tak, aby bol priestor mriežky využitý čo najviac, teda aby textúry mali na povrchu modelu čo najväčšie rozlíšenie (kvalitu).

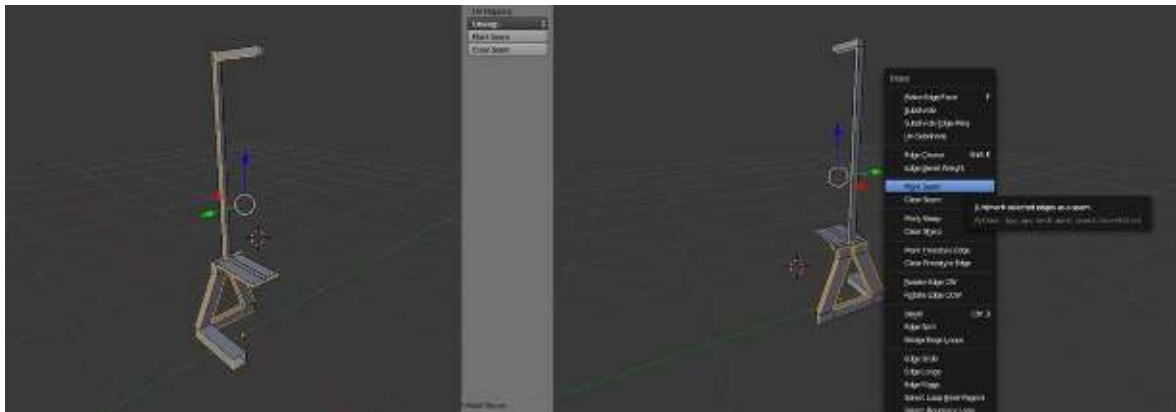
5.3.2 Vytvorenie UV máp modelu

Postup sme začali výberom požadovanej časti 3D modelu stojana na tlačové valce a jej oddelením výberu od celého modelu (Separate Selection, kláves *P*). Vybrali sme časť nohy stojana s konštrukciou, ktorá sa nám v modeli viackrát opakuje. Zvyšnú časť 3D modelu sme skryli funkciou Hide (kláves *H*). Na ľavej strane aplikácie sme si pridali ďalší editor a zvolili sme na ňom zobrazenie UV/Image editor (obrázok č.21).



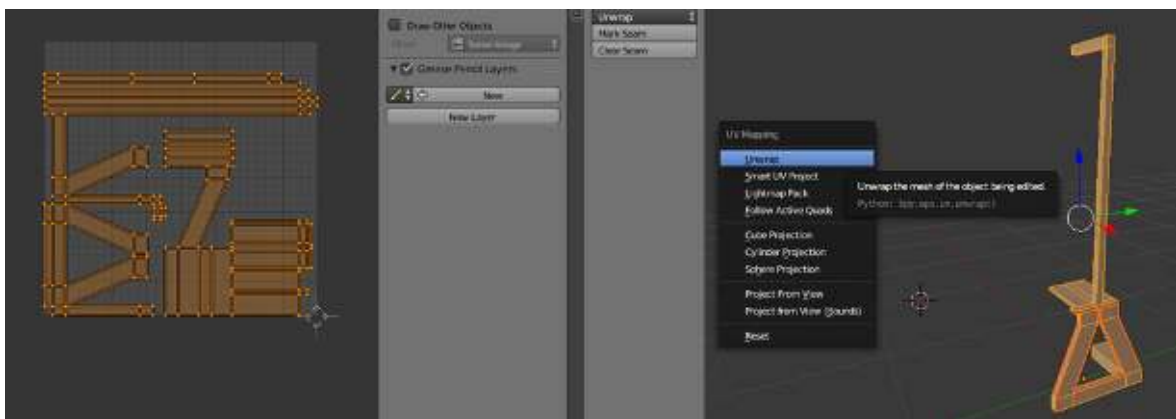
Obr. 21. Zobrazená vyselektovaná časť noha stojana s konštrukciou a UV editor.

Prešli sme z objektového módu do editačného módu (kláves *Tab*). Zvolili sme výber práce s hranami (*Ctrl+Tab*, Edge), ktoré nám následne poslúžili ako UV Seamy, teda miesta kde dôjde k rozloženiu modelu. Jednotlivé požadované hrany sme postupne označovali prostredníctvom stlačeného *Shift* a ľavého tlačidla myši. Pri označených hranách sme použili nástroj označiť švy (*Ctrl+C*, Mark Seam) (obrázok č.22).



Obr. 22. Výber hrán a označenie Seamov.

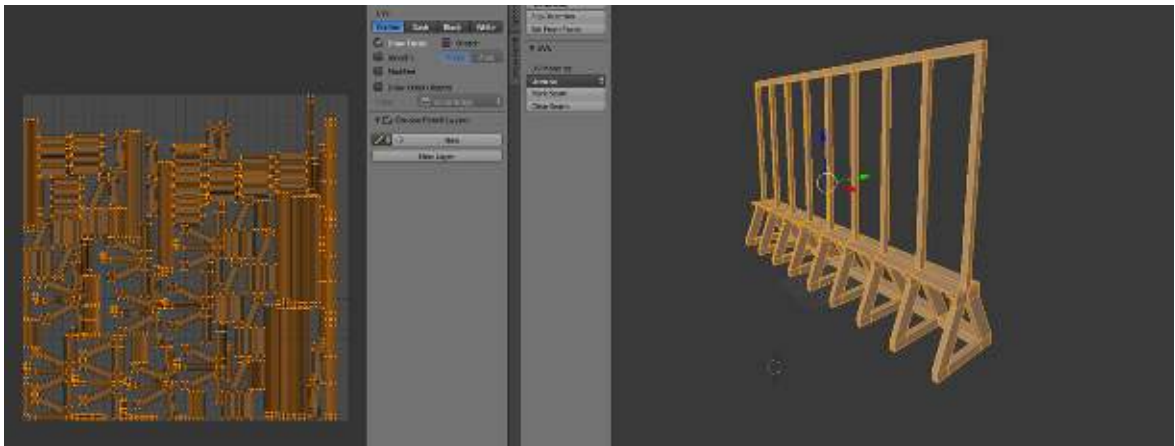
Označili sme celý model (kláves *A*), použili nástroj na prácu s UV mapami (kláves *U*) a zvolili možnosť rozbaľiť (Unwrap). Vytvorili sme UV mapu vybranej časti modelu. UV mapa (rozloženie) sa nám po použití nástroja rozbaľiť (Unwrap) zobrazila na UV mriežke v UV editore (obrázok č.23).



Obr. 23. UV mapa vybranej časti stojana na tlačové valce.

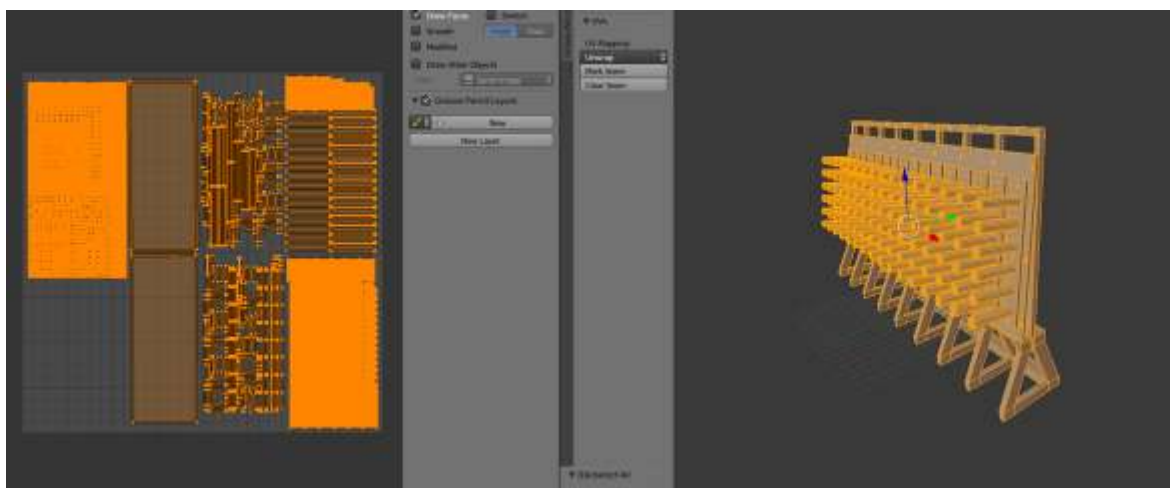
Následne, aby sme si ušetrili veľké množstvo práce a času, sme využili to, že sme mali UV mapu časti modelu, ktorá sa nám vo výslednom 3D modeli viackrát opakovala. Blender nám ponúka možnosť využitia nástroja vytvoriť spojenia (*Ctrl+L*, Make Links) a voľbou možnosti previesť UV mapy (Transfer UV Maps), a tak transformovať UV mapy z aktívneho na vybraný objekt, pričom musíme dať pozor a dodržať pravidlo, že objekt, na ktorý chceme mapu preniesť, musí mať zhodnú geometriu s objektom aktívnym (objektom s vytvorenou UV mapou). Takto môžeme označiť aj celú množinu zhodných objektov a transformovať UV mapy v jednom kroku na väčšie množstvo zhodných objektov. Zobrazili sme si skryté časti stojana voľbou možnosti zobraz skryté (Show Hidden, *Alt+H*) a vyseletovali sme si z modelu potrebné rovnaké časti ako bola časť stojana, ktorej sme vytvorili UV mapu. Vybrané časti sme od celkového modelu oddelili (kláves *P*, Selection) a vytvorili sme tak zhodné časti prvého objektu. Zvyšné časti modelu sme opäť skryli

(označenie a kláves *H*). Jednotlivé vybrané objekty sme si v objektovom móde označili s tým, že ako primárny objekt, z ktorého sa bude UV mapa transformovať bola časť s vytvorenou UV mapou. Použitie nástroja vytvoriť spojenia (*Ctrl+L*, Transfer UV Maps). Vytvorili sme UV mapy pre všetky časti stojana, ktoré môžeme následne spojiť (*Ctrl+J*, Join) do jedného objektu, pričom UV mapy jednotlivých častí sa nám automaticky preniesli do primárnej UV mapy (obrázok č.24).



Obr. 24. Vytvorená UV mapa konštrukcie stojana na tlačové valce.

Rovnakým postupom sme pracovali na všetkých častiach 3D modelu. Po vytvorení všetkých UV máp (jednotlivých UV shellov) sme tieto ešte čo najefektívnejšie rozložili na UV mriežke (obrázok č.25).



Obr. 25. Hotová UV mapa 3D modelu stojana pre tlačové valce.

Proces pri použití nástroja na automatické generovanie UV máp (Smart UV Project) fungoval na rovnakom princípe, až na rozdiel, že v prípade Smart UV, ak máme dobre vymodelovaný 3D model, si môžeme rovno označiť a vyselektovať napríklad celú konštrukciu, aplikovať nástroj a ten nám zabezpečí vytvorenie seamov a UV máp. Rozdiel je

aj v použití funkcie rozbalenia (Unwrap), tú použijeme pred nástrojom Smart UV Project. Správnosť vygenerovanej UV mapy treba skontrolovať (môže obsahovať chyby, ktoré sú pozorovateľné na UV mape rozloženej na UV mriežke).

Po vytvorení všetkých UV máp sme jednotlivé 3D modely exportovali a mohli sme prejsť k tvorbe, úprave a nasadeniu textúr. Všetky modely sme exportovali do formátu FBX postupom, voľba Súbor (File) výber Exportovať (Export), formát FBX a nastavili sme názov a umiestnenie exportovaného modelu.

5.3.3 Textúrovanie

Textúrovanie by sme mohli nazvať ako realizmus hotového 3D modelu, ktorý závisí od textúr. Inak povedané, cieľom použitia a nasadenia textúr je ukázať materiál, z ktorého je vyrobený a tvorený skutočný objekt. Textúra je pixelový obraz, ktorý je položený na pripravený polygonálny model (na UV mapu), vytvorený v grafickom programe s možnosťou dať mu vlastnosti ako farbu, parametre odrazu, parametre lomu, špecifikovať index lomu alebo pridať reliéf, falošný reliéf (efekt reliéfu) a ďalšie vlastnosti. Keď hovoríme o pridávaní či úprave vlastností, hovoríme o mapovaní textúr (texture mapping). V práci sme používali typ textúrového mapovania Reliéfne mapovanie (Relief mapping), pod ktoré patrí používanie rôznych nástrojov a techník ako Projekčné maľovanie (Projection Painting, použitie obrázku ako spôsob pridania detailu skutočného sveta do našich textúr), pridanie, úprava Farebných máp (Color Maps, niekedy označované ako difúzne mapy, definujú všetku povrchovú farbu materiálu, bez ďalších informácií o osvetlení alebo tieňovaní). Využili sme farebné mapy ako východiskový bod pre ďalšie typy textúr. Ďalšou možnosťou, ktorú sme využívali v našom projekte, je pridanie a úprava Bump máp (Bump mapa je grayscale image, obrázok v odtieňoch šedej, ktorý funguje ako mapa pseudo-výšky, nevytvára ďalšiu geometriu, ale vytvára ilúzie zvýšených alebo znížených detailov na povrchu modelu). Alternatívou použitia Bump máp môže byť použitie Normálových Máp (Normal Maps sú mapy smeru normál geometrie modelu (objektu), ktoré umožňujú tvorbu ilúzie falošnej výšky alebo ovplyvnenia modelu osvetlením a ďalších efektov). V prípade potreby kombinovania viacerých materiálov alebo textúr na jeden objekt máme k dispozícii Mapy masiek (Mask Maps umožňujú definovať pomocou čiernej alebo bielej hodnoty mapy, ktorá vrstva textúry sa zobrazí na ktorej vrstve objektu. Vo väčšine prípadov biela maska definuje oblasti pre textúru, ktorá bude viditeľná, zatiaľ čo čierna maska zobrazí ostatné vrstvy). Použitie máp masiek je veľmi flexibilný a nedeštruktívny spôsob, ako pracovať pri

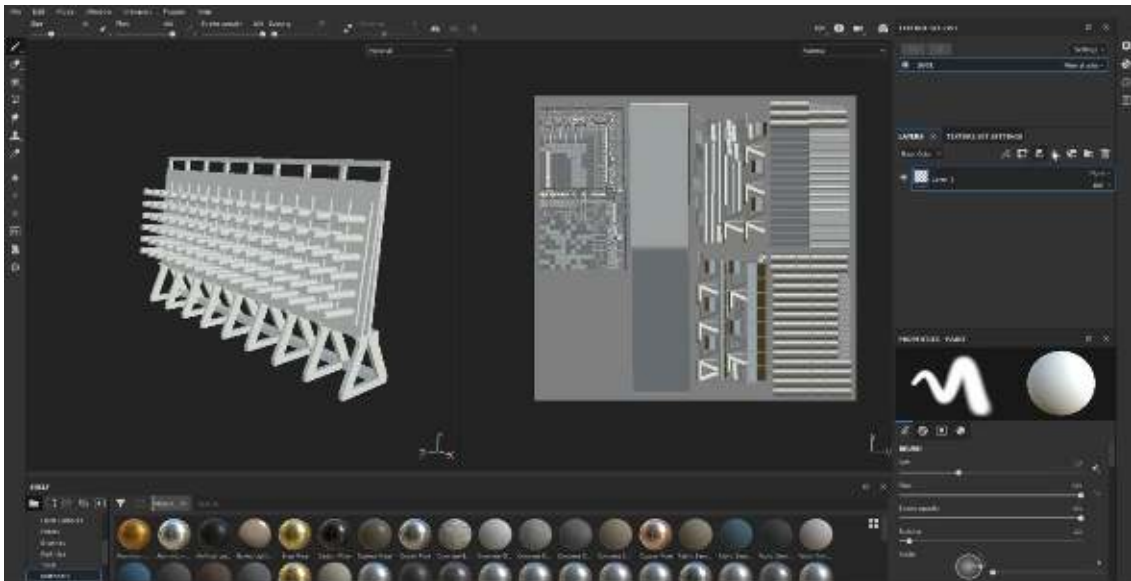
textúrovaní, pretože nám umožňuje zmeniť, kde sa textúry objavia bez potreby skutočného dotyku samotných máp. V našom projekte sme mapy masiek využívali pri takmer všetkých objektoch, vrstvách, textúrach a UV mapách. Textúrovanie je jedným z hlavných štádií tvorby 3D modelov, pri ktorom by mal modelovaný objekt získať určité špecifické vlastnosti, aby bol model realistickejší. Kvalita povrchového textúrovania je definovaná množstvom texelov (počet pixelov na jednej minimálnej jednotke textúry). Určujúcimi faktormi hotovej 3D grafiky (textúrovania) sú konečný formát textúrovaného obrazu a rozlíšenie textúr [41][45][46].

Všetky textúry boli robené na základe podobnosti materiálov z referenčných fotografií a video záznamov, ktoré boli súčasťou foto a video dokumentácie vykonaných vo vybranej výrobnéj spoločnosti. Textúrovacie techniky, metódy, postupy a spôsoby ich využitia boli v prípade potreby čerpané z manuálov a tutoriálov, určených pre textúrovaciu aplikáciu Substance Painter [41].

V prípade potreby použitia ďalších textúr, ktoré aplikácia na tvorbu a prácu s textúrami neobsahovala, boli tieto stiahnuté z internetových zdrojov [47][48].

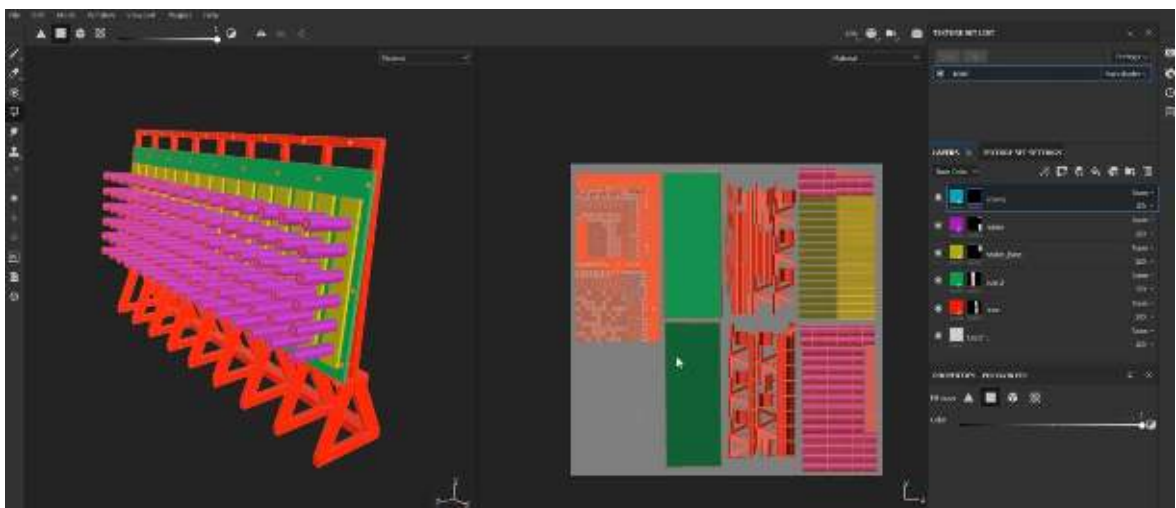
5.3.4 Úprava materiálov, tvorba, nasadenie a export textúr na vymodelované 3D modely

Pre tvorbu, úpravu a nasadenie textúr sme pre náš projekt zvolili aplikáciu pre tvorbu a prácu s textúrami Substance Painter ver.2018.1.0. Postup tvorby textúr sme začali importovaním 3D modelu s vytvorenými UV mapami do aplikácie. Zvoli sme možnosť súbor, nový projekt, vybrali sme nami exportovaný 3D model, formát normálovej mapy (Normal map format) sme ponechali na nastavení OpenGL a rozlíšenie dokumentu sme nastavili na hodnotu 1024 (neskôr pri práci s textúrami alebo pri konečnom exporte hotových textúr je možnosť zmeny nastavenia požadovaného konečného rozlíšenia bez akýchkoľvek strát). Nami importovaný model aj s UV mapou sa nám zobrazil v rozdelenom zobrazení. Na ľavej strane 3D model a na pravej strane 2D UV mapa modelu (obrázok č.26). Na pravej strane a spodnej strane aplikácie je zobrazené užívateľské rozhranie, prostredníctvom ktorého sme robili väčšinu úkonov a postupov pri tvorbe textúr. Na ľavej strane máme ponuku nástrojov na úpravy, zobrazovanie a ďalšie (obrázok č.26).



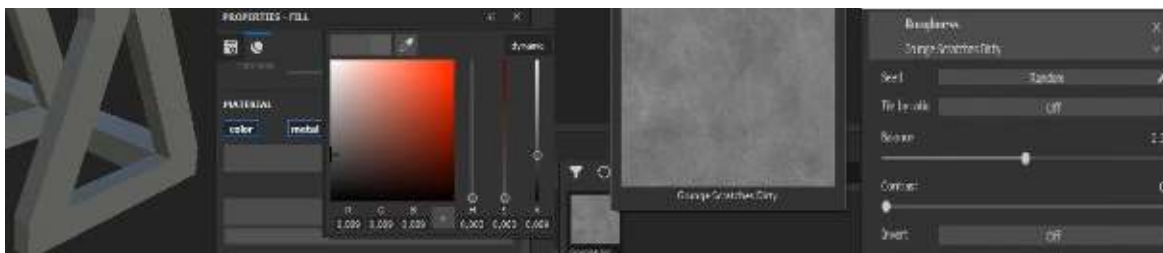
Obr. 26. Importovaný 3D model stojana s vytvorenou UV mapou a UI aplikácie.

Textúrovanie sme začali rozdelením jednotlivých častí modelu do vrstiev a tieto sme si pomenovali. Vrstvy sme pridávali voľbou Pridať vrstvu výplne (Add a fill layer), vrstvu sme pomenovali a v ponuke možností nastavenia máp sme nastavili jej Základnú farbu (Base Color) na červenú (nastavenie podľa ľubovoľnej voľby). Pridali sme do vytvorenej vrstvy čiernu masku (označíme vrstvu, pravý klik myši, Add black mask) a pridanú čiernu masku sme si následne označili (pridaný čierny štvorec vo vrstve). V ľavom stĺpci nástrojov sme prešli na nástroj Výplň polygónu (Polygon fill) a v UV mriežke sme označili časti modelu, pre ktoré sme chceli definovať pridanú masku (z tohto dôvodu je dôležité dbať na dobré rozloženie UV mapy už pri UV mappingu, pretože zvyšuje efektivitu práce pri textúrovaní). Rovnakým spôsobom sme pridali potrebné vrstvy a časti modelu tak rozdelili z dôvodu lepšej prehľadnosti počas textúrovania (obrázok č.27).



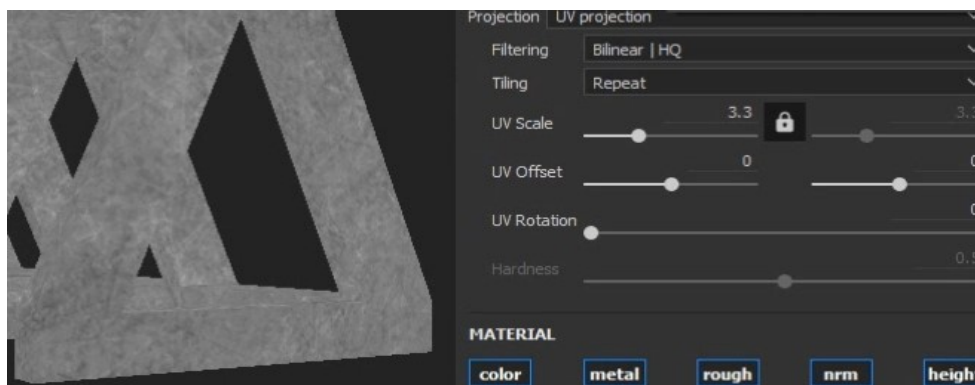
Obr. 27. Vytvorené vrstvy jednotlivých súčastí s ich farebným rozlíšením.

Následne sme začali s textúrovaním *base* vrstvy. Zmenili sme Základnú farbu (Base Color) na šedú, ktorá je podobná šedej farbe z referenčného obrázku stojana pre tlačové valce. V ďalšom kroku sme do Roughness slotu materiálu (vrstvy) pridali textúru (Grunge Scratches Dirty), ktorú nám aplikácia ponúkala (obrázok č.28). Roughness je mapa textúry, ktorá udáva ako veľmi ostré sú odrazy svetla (okolía) na povrchu objektu.



Obr. 28. Nastavenie Base Color, pridanie textúry do Roughness slotu.

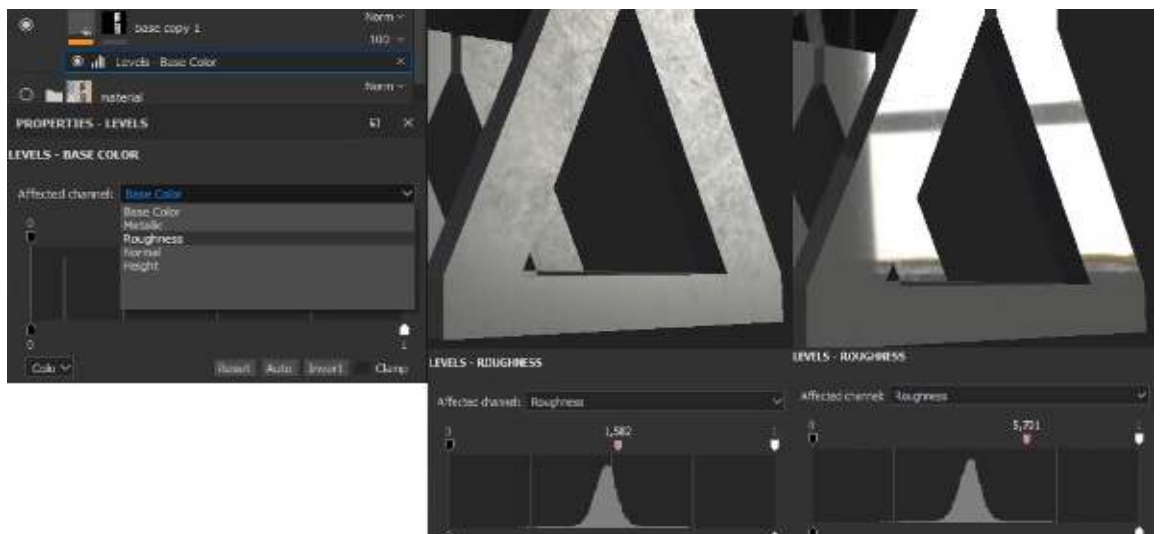
V nastavení vrstvy sme zmenili UV scale hodnotu (udáva rozloženie textúry, respektíve veľkosť textúry, ktorá je projektovaná na model). Pre lepšiu viditeľnosť a zobrazenie sme prepli na režim zobrazenia modelu Roughness v izolovanom režime (kláves C) (obrázok č.29). Návrat do materiálového módu sme vykonali stlačením klávesy M.



Obr. 29. Nastavenie hodnoty UV scale vybranej vrstvy v izolovanom režime.

Ako ďalšie sme pridali do *base* vrstvy Úrovne (Levels), pomocou ktorých môžeme ovplyvňovať textúru v Roughness vstupe (označenie vrstvy, pravé tlačidlo myši, Add levels). V pridanom Levels sme ešte museli špecifikovať Ovplyvnený kanál (Affected channel), ktorým bol v tomto prípade Roughness (obrázok č.30). V prípade ak sme chceli pracovať napríklad s mapou textúry Výšky (Height), zvolili sme hodnotu pre ovplyvnený kanál Height. Hodnotu ovplyvneného kanálu pri Levels sme volili vždy na základe mapy, s ktorou sme išli pracovať. Zmenou nastavenia v ovládaní Roughness Leves sme mohli meniť ostrosť odrazov svetla (okolía) na objekte. Pri nastavení vyšších hodnôt boli odrazy

pomerne ostré, zatiaľ čo pri nízkych hodnotách boli odrazy značne rozostrené (obrázok č.30).



Obr. 30. Pridanie úrovne (Levels) pre Roughness, zmena nastavenia hodnôt.

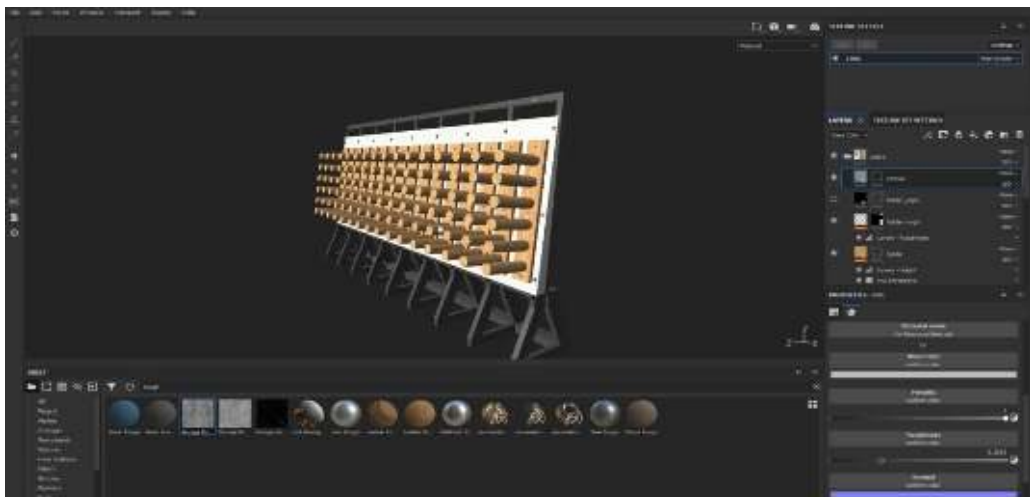
Po nastavení Roughness na požadovanú hodnotu sme ešte nastavili hodnotu Metallic, aby sme objektu *base* pridali kovový efekt. Ďalším krokom bolo pridanie Bump (Height) mapy (dodáva objektu ilúziu nerovnosti povrchu bez toho, aby dochádzalo ku zmene topológie objektu samotného. Mení spôsob lámania lúčov svetla na objekte, čo vytvára ilúziu nerovnosti). V našom prípade sme ju urobili v ďalšej vrstve, ktorú sme vytvorili formou duplikovania (Ctrl+D) *base* vrstvy a nazvali sme ju *base bump*. Dôvodom vytvorenia novej vrstvy bolo, že v Height slotu vrstvy *base bump* sme chceli použiť inú textúru, ako vo vrstve *base*, a teda, keby sme chceli meniť UV scale, tak by sme ho menili na dvoch rôznych textúrach súčasne (rovnaká hodnota UV scale pre dve rôzne textúry), a to by zabraňovalo využitiu dvoch rôznych UV scale hodnôt. Vo vytvorenej vrstve *base bump* sme v sekcii MATERIAL nechali označenú len *height* a to z dôvodu, že keby sme to neurobili, tak zvyšné kanály *base bump* by ovplyvňovali kanály vo vrstve *base* (je to z dôvodu hierarchickej logiky, vyššie vrstvy ovplyvňujú vrstvy nižšie a v našom prípade je vrstva *base* pod vrstvou *base bump*). Následne sme do slotu Height vrstvy *base bump* vložili textúru Grunge Roughcast, nastavili sme hodnotu UV scale a pridali sme do vrstvy Leves (Add Leves) s ovplyvneným kanálom nastaveným na Height. Zmenou hodnôt cez Levels sme nastavili Height textúry do požadovanej podoby (obrázok č.31). Hodnoty Levels sme pre konečnú podobu textúry ešte upravovali.



Obr. 31. Bump (Height) mapa v samostatnej vrstve s vloženou textúrou a nastavením.

Úspešne sme vytvorili textúru na vybranú časť 3D modelu. Popísaným spôsobom sme postupovali a pokračovali s tvorbou textúr aj vo zvyšných vrstvách modelu až pokiaľ sme nevytvorili textúry pre všetky vrstvy (obrázok č.32).

Pri tvorbe textúr na ďalšie časti modelu, ale aj všetky zvyšné modely, sme postupovali rovnakými alebo podobnými metódami a postupmi. Ďalšími používanými postupmi boli napríklad pridanie filtrov do vrstiev (Add filter) a pridanie HSL Perception filtra, kde sme pomocou nastavenia hodnôt odtieňu, sýtosti a svetlosti (Hue, Saturation, Lightness) nastavovali farebné kanály objektov. V prípade potreby použitia rôznych textúr pre jednotlivé kanály objektu sme tieto kanály rozdelili do samostatných vrstiev, a tak sme mohli bez problémov textúry, ich efekty a vlastnosti kombinovať. Pri potrebe vloženia špecifickej textúry do Base Color slotu alebo iných slotov, ako to bolo potrebné pri textúrovaní častí *holder* a *holder_base* vrstiev modelu stojana, a Substance Painter požadovanú textúru neobsahoval, museli sme textúru stiahnuť z internetových zdrojov, importovať ju a použiť.



Obr. 32. Hotové textúry na 3D model stojana pre tlačové valce.

Po vytvorení textúr na 3D model nám ostával posledný krok, ktorým bol export vytvorených textúr. Ešte pred samotným exportom sme zmenili rozlíšenie textúr, ktoré sme na začiatku nastavili na hodnotu 1024 x 1024 na hodnotu 4096 x 4096. Zmenu sme vykonali v záložke TEXTURE SET SETTINGS a zmenou na požadovanú hodnotu v poli veľkosť (size). Export sme vykonali zobrazením ponuky Súbor (File) a voľbou možnosti Exportovať textúry (Export textures). Zobrazila sa nám ponuka, v ktorej sme zadali typ konfigurácie (špecifická konfigurácia pre export textúr), v našom prípade sme vybrali možnosť Unity 5 (Standard Metallic), čo je nastavenie textúr pre nami zvolené vývojové prostredie. Formát textúr sme zvolili tiff a veľkosť dokumentu (document size) sme nastavili na hodnotu 4096 x 4096. Vybrali sme cestu pre uloženie exportovaných textúr a zvolili možnosť Exportovať (Export). Pri exporte sa Height mapa konvertovala na Normálovú mapu (konfigurácia Unity 5). Rovnakým postupom sme vyexportovali všetky vytvorené textúry.

6 VYTVORENIE APLIKÁCIE S PODPOROU VIRTUÁLNEJ REALITY PRE ÚČELY ZAÚČANIA NOVÝCH ZAMESTANCOV VO VÝROBNOM PROCESE

Po vymodelovaní všetkých 3D modelov, vytvorení UV máp a tvorbe textúr pre jednotlivé modely sme mohli prejsť k samotnej tvorbe VR aplikácie, simulujúcej vybraný výrobný proces a jeho postup.

Pri vývoji a tvorbe VR aplikácie sme v prípade potreby čerpali informácie z publikácií a manuálov určených pre vývoj v nami zvolených vývojárskych nástrojoch [2][5][6][8].

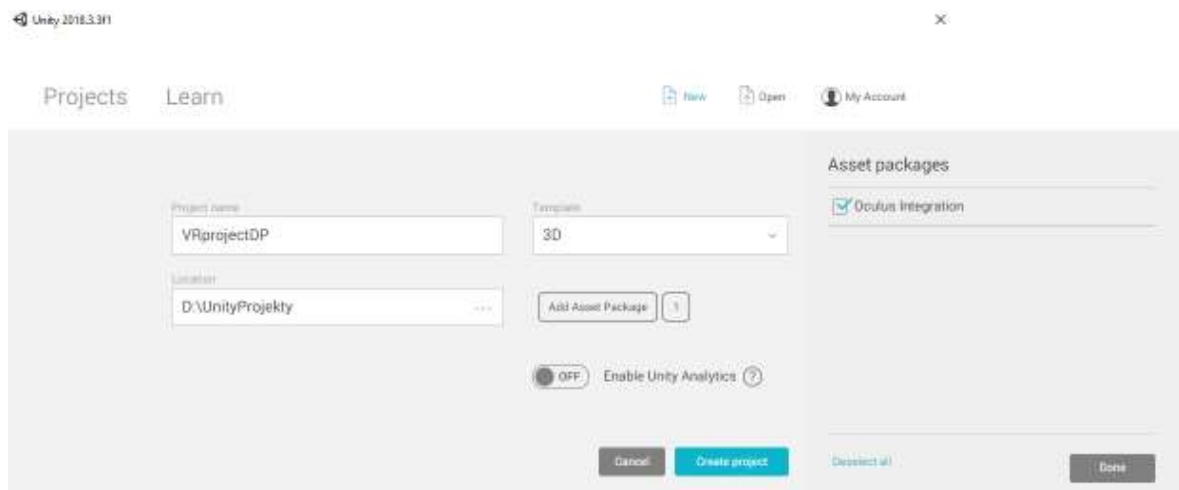
6.1 Vytvorenie VR aplikácie

6.1.1 Inštalácia, nastavenie VR hardvéru a vytvorenie nového projektu

Prvým krokom po exporte vytvorených 3D modelov a ich textúr bola potreba nainštalovania ovládačov a softvéru pre zvolený VR hardvér Oculus Rift, Oculus Touch a senzory. Potrebný inštalčný balík Oculus Rift Software sme si stiahli z webovej stránky výrobcu zo sekcie Setup. Po stiahnutí sme inštalčný balík nainštalovali. Softvér sa po úspešnej inštalácii automaticky spustil a vyžadoval vytvorenie užívateľského konta a prihlásenie sa do Oculus prostredia (mali sme možnosť prihlásiť sa aj prostredníctvom Facebook konta). Potrebné kroky sme vykonali. Po prihlásení sme prešli sériou základných nastavení vytvoreného konta, ale aj samotného VR hardvéru. Sprievodca nastavením VR hardvéru nás vyzval na pripojenie všetkých jeho komponentov, ktoré sme po rozpoznaní museli pre správnu funkčnosť nakonfigurovať. Správnu konfiguráciu VR zariadení sme otestovali formou skúšobnej VR aplikácie od spoločnosti Oculus. Následne sme mohli prejsť na vývoj našej VR aplikácie.

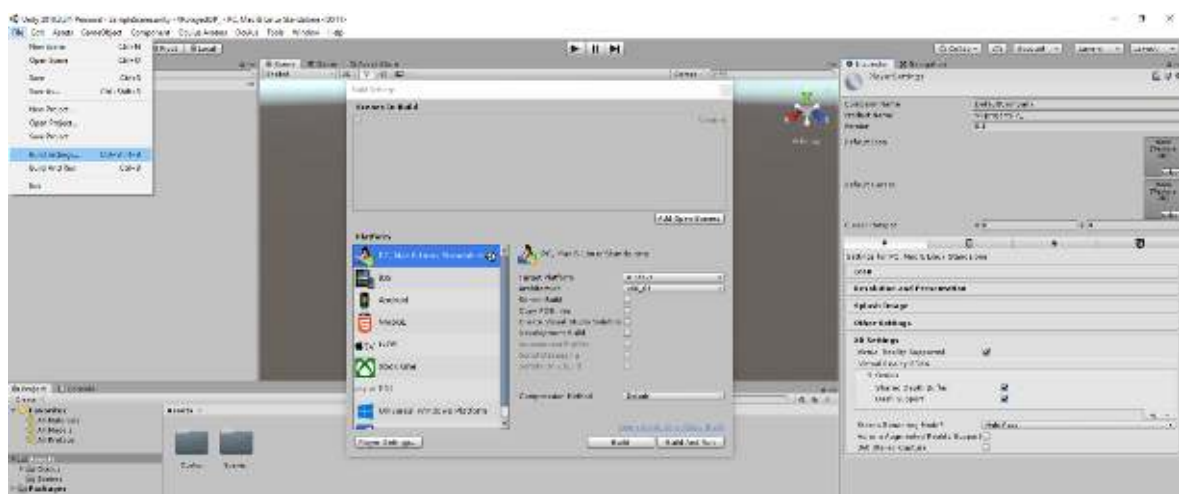
Pre vývoj VR aplikácie sme zvolili herný engine Unity 3D ver. 2018.3.3f1 (64-bit) a na tvorbu skriptov Microsoft Visual Studio 2017 (programovací jazyk C#). Po spustení herného enginu Unity sa nám zobrazila úvodná obrazovka na prihlásenie užívateľa prostredníctvom Unity konta. Konto na Unity sme už mali vytvorené, takže sme ho použili na prihlásenie sa. Po prihlásení nás čakala ponuka s možnosťami tvorby nového projektu na lokálnom disku alebo v Cloude a možnosť otvorenia už existujúceho projektu. Vybrali sme možnosť vytvorenia nového projektu na lokálnom disku (New, On Disk). Meno projektu

sme zadali „VRprojectDP“, v kolónke Template sme zvolili, že sa jedná o 3D projekt, vybrali sme umiestnenie projektu a v možnosti Pridať Asset Balík (Add Asset Package) sme vybrali možnosť Oculus Integration (je potrebný balík pre podporu Oculus zariadení obsahujúci preddefinované vytvorené objekty a skripty zabezpečujúce základné funkcie VR zariadení) (obrázok č.33).



Obr. 33. Vytvorenie VR 3D projektu „VRprojectDP“ s pridaným Oculus Integration balíkom.

Po vytvorení projektu nám herný engine otvoril vývojové prostredie, ktoré sme si popísali v časti práce, ktorá sa zaoberala popisom vývojových nástrojov. Aby sme mohli začať vyvíjať VR aplikáciu, museli sme cez voľbu možností Súbor (File), Nastavenie budovania (Building Settings) a možnosť Nastavenia prehrávača (Player Settings) povoliť v záložke inšpektora XR Nastavenia (XR Settings) možnosť Podporovaná virtuálna realita (VR Supported), v ponuke VR SDKs sme ponechali len možnosť Oculus (obrázok č.34). Vytvorený projekt sme ešte uložili, čo nám zmenilo názov hlavnej scény projektu.

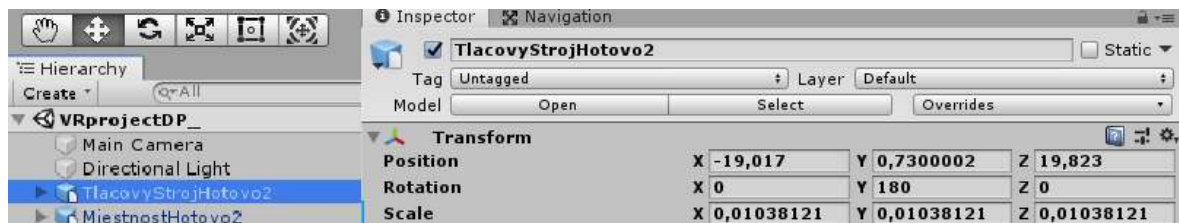


Obr. 34. Nastavenie herného engine Unity pre podporu vývoja VR Oculus.

Prostredie bolo pripravené na vývoj našej VR aplikácie. Prešli sme k importu vytvorených 3D modelov a ich textúr.

6.1.2 Implementácia vymodelovaného 3D prostredia a jeho komponentov, úprava

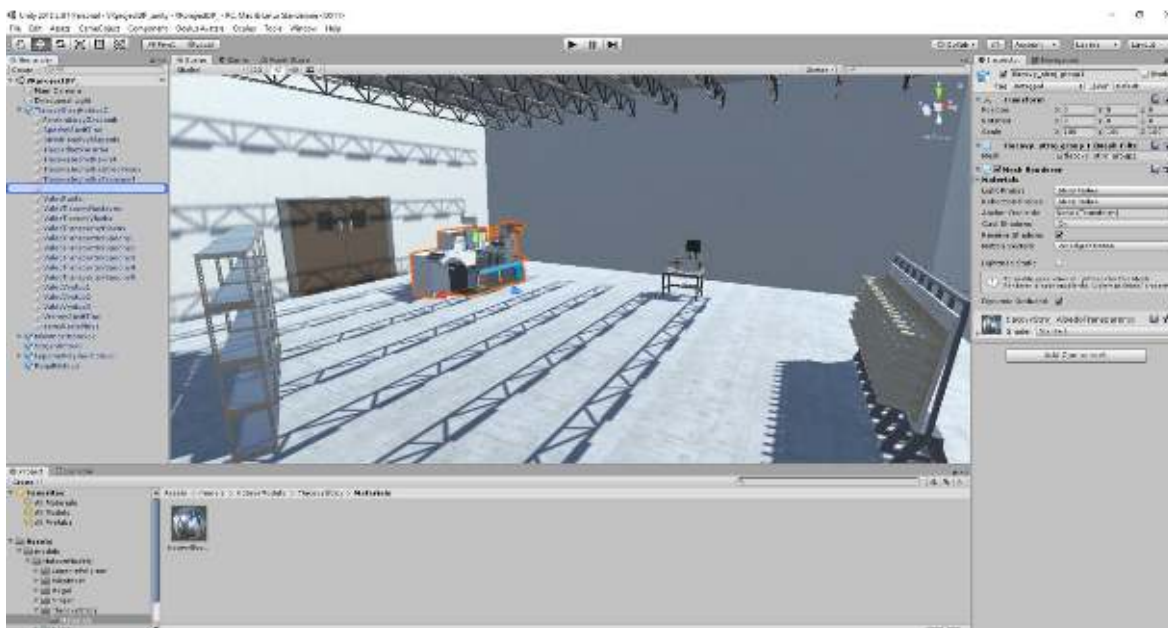
V paneli Project sme vytvorili novú zložku (pravé tlačidlo myši, Create, Folder) a nazvali sme ju models. Zložku sme otvorili a spôsobom drag-and-drop sme do nej prekopírovali pripravenú zložku z lokálneho disku nášho PC s vyexportovanými 3D modelmi a textúrami, ktoré sme si z dôvodu prehľadnosti usporiadali podľa ich príslušnosti. Keď bol import modelov a textúr hotový, postupne sme začali spôsobom drag-and-drop implementovať jednotlivé modely na našu scénu (objekty sme pridávali do Hierarchy panelu). Po pridaní všetkých 3D modelov sme museli modely na scéne rozložiť, usporiadať, nastaviť im veľkosť a rotáciu, všetko podľa našej potreby a predstavy. Potrebné transformačné zmeny jednotlivých modelov sme robili prostredníctvom nástrojov na pohyb (presun), rotáciu a škálovanie (Move Tool, Rotate Tool a Scale Tool)(obrázok č.35), alebo priamym nastavením hodnôt jednotlivých modelov a ich súčastí cez Inspector panel a Transform hodnoty (Position, Rotation, Scale) pričom hodnoty každého parametra sa dali nastavovať v troch rozmeroch (v osi X, Y a Z) (obrázok č.35).



Obr. 35. Nastavenie umiestnenia, rotácie a rozmerov implementovaných objektov.

Po rozložení, usporiadaní, rotácií a nastavení požadovaných veľkostí všetkých pridaných 3D modelov sme mohli prejsť k nasadeniu ich textúr. V zložkách jednotlivých modelov sme mali aj ich textúry. Exportované textúry sa skladali z troch samostatných máp, kanálov (AlbedoTransparency, MetallicSmoothness a Normal). Po ich importe do herného enginu sme na základe kanálov textúr vytvorili v ich umiestnení zložku s názvom Materials, v ktorej nám po použití nástroja Vytvoriť Prednastavený Materiál (pravé tlačidlo myši na kanál textúry, Create Material Preset) Unity vytvoril z jednotlivých kanálov hotový materiál. Postup sme opakovali pre textúry všetkých modelov. Vytvorené preddefinované materiály sme mohli nasadiť na príslušné modely spôsobom drag-and-drop, pričom cieľom drop boli pridané modely a ich súčasti v Hierarchy paneli. Ak sa model skladal len z jednej časti (napríklad Regál), vytvorený materiál sme položili na tento objekt. Ak sa

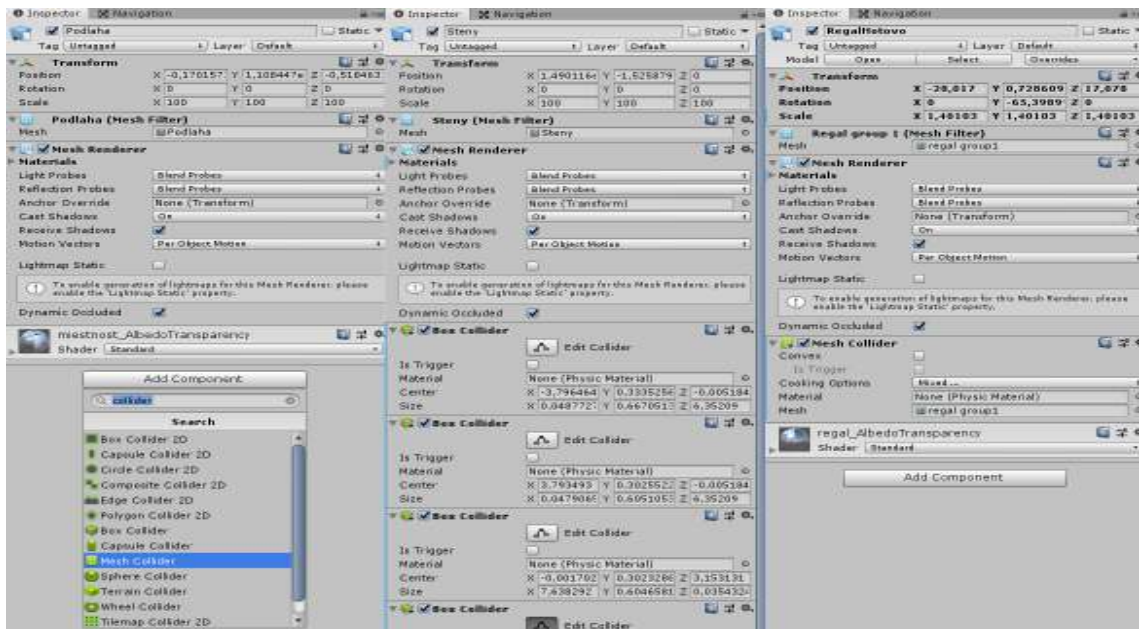
modely skladali z viacerých samostatných častí (z dôvodu interakcií, manipulácií a transformácií s oddelenými časťami), hotový materiál sme museli pokladať na jednotlivé časti zvlášť (výsledný efekt bol rovnaký, ako keby bol materiál nasadený na jeden objekt). Rovnakým spôsobom sme nasadili textúry na všetky implementované 3D modely (obrázok č.36).



Obr. 36. Implementované 3D modely s nasadenými textúrami.

Mali sme úspešne implementované modely s nasadenými textúrami. Nasledovalo pridanie koliderov (colliders) z dôvodu, že pridané objekty obsahovali síce vo svojej štruktúre Mesh Renderers, ktoré zabezpečovali objektom vykresľovanie ich mriežok (meshov) ale neobsahovali žiadne komponenty, ktoré by riešili ich fyziku (bez pridanej fyziky mohli jednotlivé objekty medzi sebou prechádzať, ako keby boli len vykreslené, ale nemali žiadnu fyzickú štruktúru). Pre tento účel nám Unity ponúkalo kolider komponenty (collider components). Ich účelom bolo vytvorenie akoby fyzickej vrstvy, ktorá v prípade stretu dvoch koliderov zabraňovala pohybu medzi objektmi. K dispozícii sme mali viac druhov koliderov, pričom každý vytváral fyzickú štruktúru pre rôzne objekty (podľa typu a tvaru objektu, do ktorého sme komponent chceli pridať) (obrázok č.37). Vo väčšine prípadov sme používali pridanie Mesh Collider (vytváral štruktúru na základe meshu objektu), ale niekedy sme napríklad potrebovali pridať do jedného objektu aj viac Box Colliders, ktoré sme mohli podľa potreby upravovať transformačnou metódou Scale (zmenou hodnoty škály). Kolidery sme pridávali postupom označenia požadovaného objektu v Hierarchy paneli, v Inspector paneli sa nám zobrazili komponenty, ktoré zvolený objekt obsahoval. Na konci aktuálnych komponentov sme mali možnosť Pridať Komponent (Add Component), po

ktorého voľbe sa nám zobrazilo vyhľadávacie pole, do ktorého sme zadali „collider“ a Inšpektor nám na základe zadanej hodnoty zobrazil všetky komponenty, obsahujúce v názve zadané slovo (v tomto prípade všetky kolidery) (obrázok č.37). Z ponuky sme vybrali kolider, a ten sa nám pridal medzi aktívne komponenty objektu (obrázok č.37).



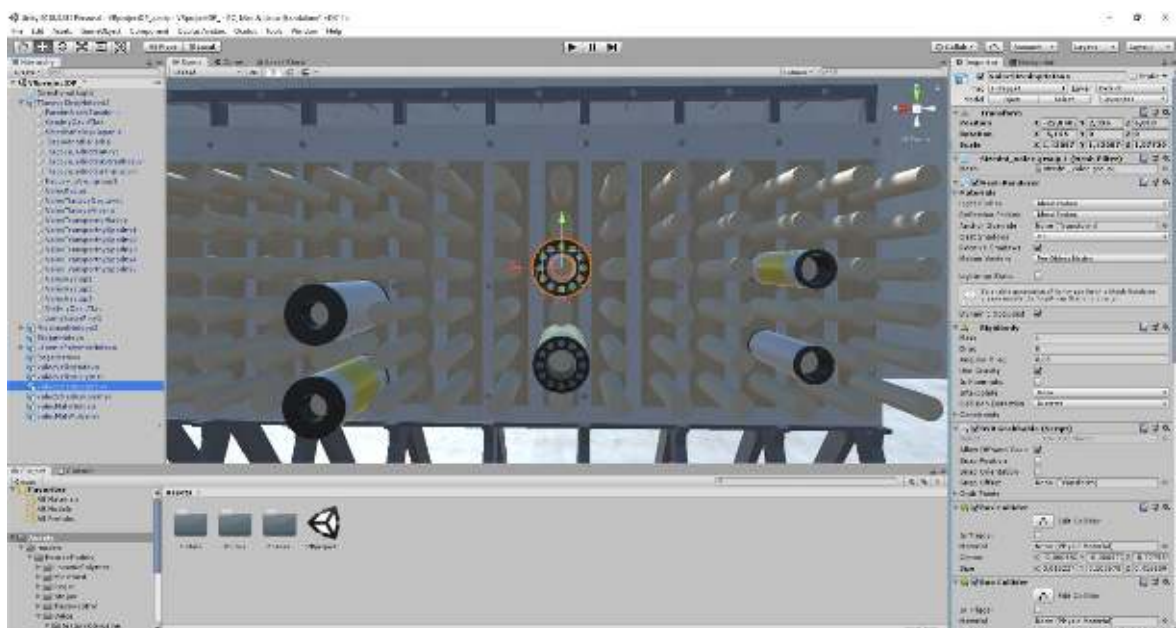
Obr. 37. Ponuka možností pridania koliderov, príklady pridania Mesh a Box koliderov.

Rovnakým spôsobom sme pridali požadované komponenty koliderov do všetkých pridávaných objektov (modelov) a ich jednotlivých súčastí. Do prostredia sme potrebovali implementovať aj kolidery, ktoré neslúžili na pridanie fyziky, ale ako spúšťače (senzory), ktoré pri kolízii s iným koliderom v prostredí vykonajú naprogramované funkcie. V našom prípade sme potrebovali sledovať vstup do takýchto koliderov objektom charakteru (koliderom objektu). Po kolízii sa zobrazilo UI daného objektu a na základe výberu simulácie v hlavnej ponuke aj pomocný obsah s postupom krokov, ktoré treba vykonať. Spúšťače koliderov sme potrebovali pridať a nastaviť do objektov stojana na tlačové valce, zariadenia na lepenie polyméru, tlačiarenského stroja a regálu na farebníkové zásobníky. Aby bol pridávaný kolider spúšťačím v nastavení Box Collideru, povolili sme atribút Is Trigger (je spúšťač) (obrázok č. 38). Rovnakým spôsobom sme pridali a nastavili spúšťače koliderov aj do zvyšných vymenovaných objektov. Zariadeniu na lepenie polymérov sme pridali navyše Capsule kolider.



Obr. 38. Spúšťací (Trigger) Box Collider objektu tlačiarenskeho stroja.

Ďalším špeciálnym prípadom boli tlačové valce, v ktorých sme potrebovali z dôvodu ich interakcie s užívateľom, ale aj s prostredím, prispôbiť pridané Box kolideru na požadovaný tvar (obrázok č.39). Zároveň sme objekty valcov potrebovali doplniť o ďalšie komponenty a skripty, ktoré by im pridali ďalšiu fyziku a umožnili im byť v prostredí hmatateľné (možnosť objekt vziať a manipulovať s ním). Pre dosiahnutie požadovaných vlastností sme do objektov valcov pridali komponent Rigidbody (Pevné teleso), spôsobom označenia všetkých valcov v Hierarchy paneli, v Inspector paneli sme zvolili možnosť Add Component, zadali sme názov Rigidbody a nájdený komponent pridali. Komponent pridával objektu potrebnú fyziku vrátane možnosti nastavenia gravitácie (gravitáciu sme povolili). Pre umožnenie objektom byť hmatateľnými sme rovnakým spôsobom pridali skript s názvom OVR Grabbable (súčasťou Integration balíku), ktorý schopnosť zabezpečoval (obrázok č.39). Ďalšími objektmi v našej aplikácii, ktorými sme potrebovali špeciálne upraviť pridané Box kolideru, pridať komponenty Rigidbody a skripty OVR Grabbable, boli *ValecTlacovyVlozka* a všetky *FarebnikoveZasobniky* (objekty boli samostatnými časťami modelu tlačového stroja).



Obr. 39. Tlačové valce s pridanými Box kolidermi, Rigidbody a OVR Grabbable.

Z hľadiska vytvoreného výrobného prostredia, jeho objektov a komponentov, nám zostávalo vytvoriť a upraviť pár samostatných podobjektov tlačiarenskeho stroja a zariadenia na lepenie tlačových polymérov. Začali sme pridaním a úpravou podobjektov tlačiarenskeho stroja. Keďže časťou výrobného procesu bolo vzatie a vloženie pripravených farebníkových zásobníkov do rastrovej časti tlačiarenskeho stroja, potrebovali sme tieto zásobníky vytvoriť a umiestniť na potrebné miesto (regál na uloženie pripravených farebníkových zásobníkov). Vytvorenie sme riešili spôsobom zduplikovania pôvodného vymodelovaného zásobníka (objektu tlačiarenskeho stroja s názvom *FarebnikovyZasobnik*. Všetky zásobníky sme pomenovali (pomenované farbami, ktoré reprezentovali ich náplň) a umiestnili na regál (obrázok č.40).



Obr. 40. Vytvorené a umiestnené farebníkové zásobníky.

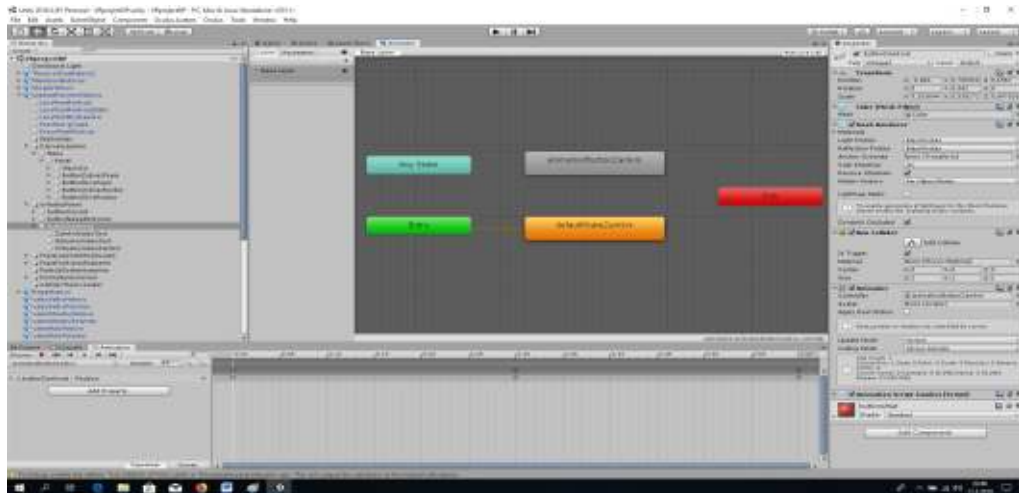
Z dôvodu zjednodušenia aplikačnej logiky a funkčnosti tlačiarenskeho stroja sme potrebovali vykonať malé úpravy jeho podobjektov s názvami *VrchnyZavitTlac*, *SpodnyZavitTlac* a *ZamykaciePiny1*. Všetky z menovaných objektov sme v Inspector paneli nastavili ako Disabled (Vypnuté). Objekt *VrchnyZavitTlac* sme zduplikovali a presunuli do objektu *StredPanelVyklapanie* a objekt *SpodnyZavitTlac* sme zduplikovali a presunuli do objektu *TlacovaJednotkaStredPosuv*. Objekt *ZamykaciePiny1* sme zduplikovali dvakrát. Jeden duplikát sme presunuli do objektu *ValecRaster* a druhý do *ValecTlacovyNastavec*. Zduplikované piny sme museli ešte umiestniť na potrebné miesta. Všetky presuny sme robili spôsobom drag-and-drop. Nakoniec sme všetky zduplikované objekty nastavili ako Enabled (Zapnuté).

Potrebovali sme ešte pridať objekt, ktorý by nám na základe voľby užívateľa umožnil zobrazit' (zvýrazniť) požadovaný priestor alebo objekt na scéne. Na tento účel sme využili prefab objekt *ParticleSystem* (obrázok č.41). Objekt sme pridali, pomenovali, upravili veľkosť a umiestnili do objektov stojana pre tlačové valce zariadenia na lepenie polymérov a regál pre farebníkové zásobníky.



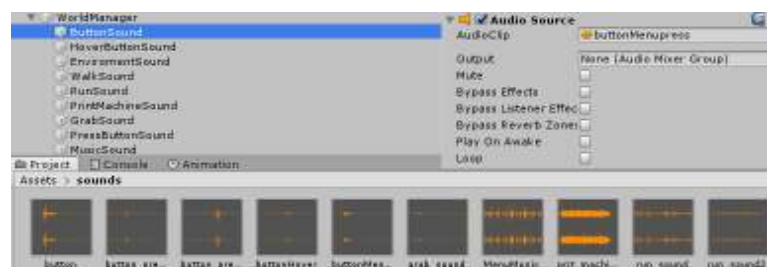
Obr. 41. ParticleSystem zariadenia na lepenie tlačových polymérov.

V objekte zariadenia na lepenie tlačových polymérov sme potrebovali vytvoriť ovládací panel s tlačidlami na obsluhu zariadenia a formu na tlačový polymér. Všetky objekty a ich súčasti sme vytvorili, umiestnili a nastavili takým spôsobom, aký sme používali pri tvorbe a nastavení objektov počas tvorby aplikácie. Pre vytvorené objekty tlačidiel sme potrebovali vytvoriť a zakomponovať animácie, ktoré by simulovali pohyb stlačenia tlačidiel. Označili sme si objekt tlačidla *buttonZamknut*, v Inspector paneli sme mu pridali komponent Animator. V Project paneli sme si vytvorili zložku s názvom *animations*. Do zložky sme pridali objekty Animator Controller a Animation (pravé tlačidlo myši, Create, Animator Controller a Animation). Oba objekty sme si pomenovali. Objekt Animator Controller sme vložili spôsobom drag-and-drop do Controller atribútu komponentu Animator objektu *buttonZamknut*. V hlavnej ponuke Unity v záložke Windows sme vybrali možnosť Animation a zvolili sme možnosť Animátor, čím sme pridali do Toolbaru editor Animátora. Tiež sme zvolili možnosť Animation, čím sme pridali Animation editor do záložiek Project panelu. V Project paneli zložke *animations* sme dvojklikom vybrali objekt Animation Controller *animationButtonZamkni*. Otvoril sa nám pridaný Animator editor so základnými stavmi a animáciami označeného kontroleru. Spôsobom drag-and-drop sme pridali do kontroleru objekt Animation *animationButtonZamkni*. V kontroleri sme vytvorili ďalší stav (pravé tlačidlo myši, Create State, Empty), následne sme stav označili a definovali sme mu Set as Layer Default State (vytvorený stav sme definovali ako základný stav, v ktorom sa kontroler nachádzal). Jednotlivé animácie a stavy sme si podľa potreby pomenovali (obrázok č.42). Ostávalo nám vytvoriť animáciu. Otvorili sme nástroj na tvorbu animácií, ktorý sme pridali do záložiek Project panelu. Označili sme v Hierarchy paneli objekt *buttonZamkni*, pre ktorý sme animáciu vytvárali. V ponuke nástroja sme stlačili tlačidlo na nahrávanie (červený kruh). Na časovej osi nástroja sme postupne posúvali časovú hranicu, pričom v požadovaných časoch sme objekt *buttonZamkni* posúvali na scéne pomocou transformačných metód. Nástroj na tvorbu animácie si pre vybraný čas vždy uložil všetky zmeny v transformácií vybraného objektu (obrázok č.42). Týmto spôsobom sme vytvorili animáciu. Rovnakým spôsobom sme postupovali pre tvorbu a implementáciu animácií zvyšných tlačidiel.



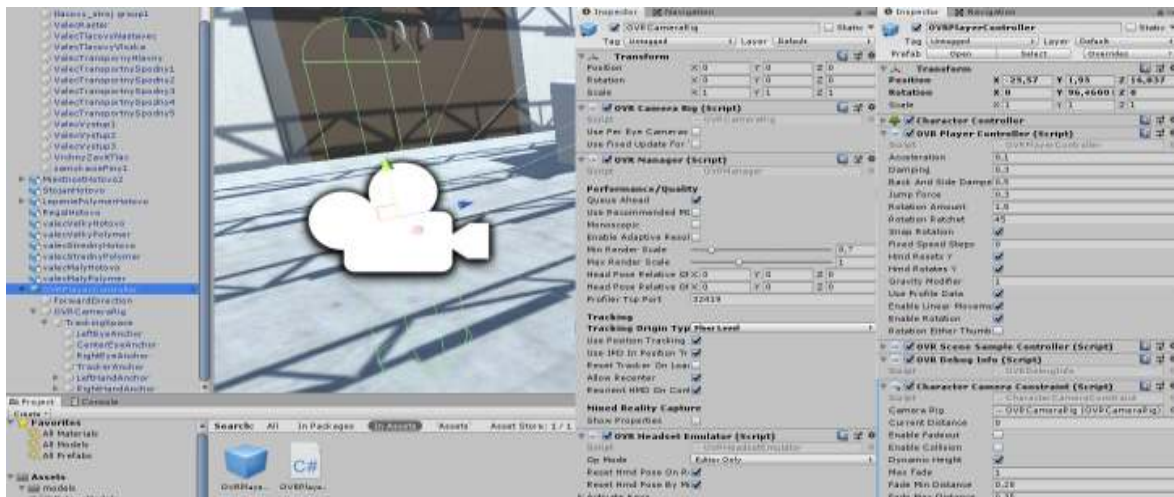
Obr. 42. Animation editor, Animator editor, vytvorenie animácie pre objekt *buttonZamkni*.

Keďže sme v našej VR aplikácii chceli implementovať aj zvuk, potrebovali sme stiahnuť a importovať požadované zvukové súbory. Zvukové súbory sme stiahli z webovej stránky www.freesound.org vo formáte .mp3 (stiahnuté a použité zvukové súbory podliehali voľne použiteľnej licencií pre akýkoľvek účel, vrátane použitia pre vývoj aplikácií). Plánovali sme použiť zvukové efekty pre UI (zvučka hlavná ponuka, zvuk stlačenia tlačidla ponuky UI, hover tlačidiel UI), ďalej zvuky pre chôdzu, beh, uchopenie, pustenie objektu a zvuk tlačiarenského stroja po spustení tlače. Po stiahnutí zvukových súborov sme si vytvorili v Project paneli zložku s názvom *sounds* a stiahnuté súbory sme spôsobom drag-and-drop do zložky importovali. V Hierarchy paneli sme vytvorili prázdny objekt s názvom *WorldManager*, do ktorého sme pridali podobjekt typu Audio, Audio Source (zvukový zdroj). Objekt sme pomenovali podľa zvuku, ktorý mu bol implementovaný, a ktorý reprezentoval. Do slotu atribútu AudioClip, komponentu Audio Source sme vložili spôsobom drag-and-drop požadovaný zvukový súbor zo zložky sounds. Podľa povahy zvuku sme ešte upravili nastavenie atribútov Play On Awake (hraj pri spustení aplikácie) a Loop (opakuj prehrávanie v slučke). Rovnakým spôsobom sme pridali a priradili všetky zvukové súbory (obrázok č.43).



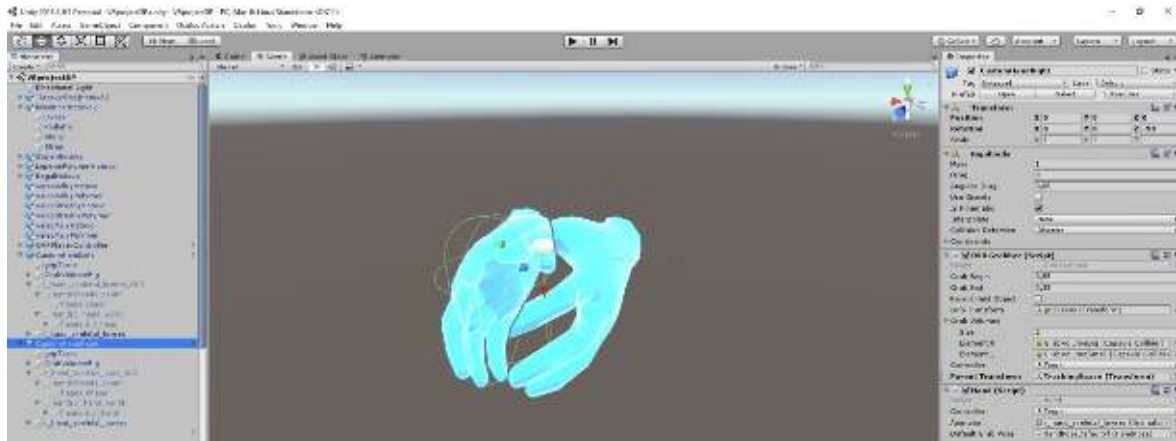
Obr. 43. Pridanie zvukových objektov, nastavenie.

Mali sme pridané a upravené všetky potrebné objekty a komponenty a vytvorenú základnú fyziku výrobného prostredia. Pred prvým testom našej aktuálnej VR aplikácie sme ešte potrebovali pridať do Hierarchy panelu najdôležitejšie objekty pre VR. Prvým objektom bol tzv. prefab OVRPlayerController, ktorý bol súčasťou balíka Oculus Integration. OVRPlayerController zabezpečoval funkčnosť všetkých VR zariadení. Spájal v sebe objekty, komponenty a skripty, prostredníctvom ktorých zabezpečoval napríklad snímanie a sledovanie polohy headsetu, renderovanie obrazu a 3D obsahov do headsetu, snímanie ovládačov, ovládanie pohybu charakteru, plnil funkcie avatara a hlavnej (Main) kamery, ktorú nahrádzal. Main kameru sme pred pridaním objektu z Hierarchy panelu vymazali (ľavé tlačidlo myši, Delete). Objekt OVRPlayerController sme vyhľadali zadaním názvu do vyhľadávacieho pola Project panelu. Po zobrazení objektu sme ho drag-and-drop spôsobom pridali do Hierarchy panelu (obrázok č.44). Po pridaní do Hierarchy panelu sme pre správnu funkčnosť objektu vykonali sériu nastavení. Nastavili sme si polohu umiestnenia objektu na scéne a pre parameter Radius sme nastavili hodnotu 0,2 (atribút Radius nastavoval veľkosť kolideru). V objekte s názvom OVRCameraRig, ktorý bol súčasťou OVRPlayerControlleru sme nastavili v atribúte Tracking Origin Type (definuje aktuálny typ pôvodu sledovania) možnosť Floor Level (Úroveň zeme) (obrázok č.44), čo nám umožní pohybovať sa v našej aplikácii pomocou joystiku ľavého ovládača (pohyb nie je zatiaľ správny, pretože nastavenie nám umožnilo pohybovať len kamerou (renderovaným obrazom, ale nie samotným charakterom). Pre správne nastavenie Charakter kontroleru a pohybu po scéne sme museli ešte pridať do objektu OVRPlayerController skript z Integration balíka s názvom Character Camera Constraint (Add Component, názov komponentu, pridať) a nastaviť do parametra skriptu Camera Rig spôsobom drag-and-drop objekt OVRCameraRig. Tiež sme museli povoliť parameter skriptu na dynamickú výšku (Dynamic Height) (obrázok č.44), čo aktivovalo sledovanie a zmenu polohy charakteru v osi Z. Teraz sme mali korektne nastavený OVRPlayerController. Správne nastavenie a funkčnosť sme otestovali v Run móde herného enginu (funkčnosť bola správna, pri pohybe sa menila pozícia celého charakteru, vrátane zmeny výšky).



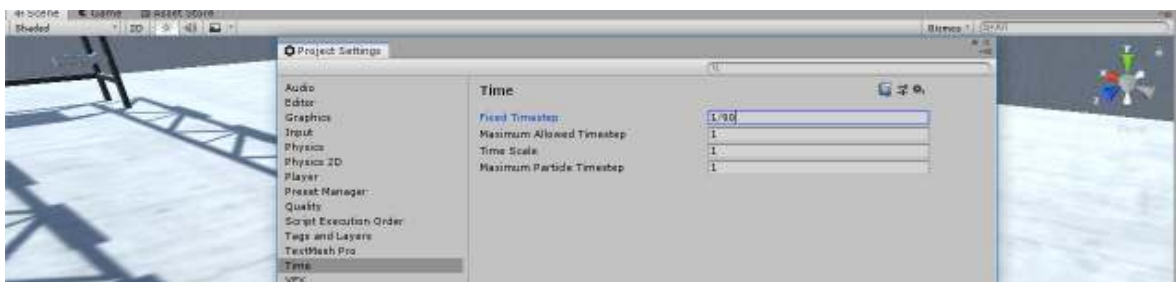
Obr. 44. Pridanie objektu OVRPlayerController, skriptu Character Camera Constraint a nastavenie objektu a jeho súčastí.

Druhými potrebnými objektmi boli prefab z Integration balíka s názvami CustomHandLeft a CustomHandRight. Reprezentovali virtuálne ruky, ktoré boli našimi nástrojmi na interakciu s prostredím. Objekty sme pridali do Hierarchy panelu spôsobom drag-and-drop. Oba boli vytvorené a pripravené na úpravu a nastavenie podľa potreby vývojára. Obsahovali skripty s názvom OVR Grabber, ktoré zabezpečovali funkcionlitu grabera (objektu, ktorý môže v prostredí manipulovať s objektmi ktoré sú grabbable, v našom prípade tlačové valce, farebníkové zásobníky a vložka tlačového valca). Prešli sme k ich nastaveniu a prispôsobeniu. Potrebovali sme rukám definovať ich Parent Transform (objekt ku ktorému sa budú vzťahovať vykonané transformácie). Tento bol pre nás pripravený ako súčasť OVRPlayerControlleru (podobjekt TrackingSpace). Pridali sme ho do atribútu oboch rúk Parent Transform v skripte OVR Grabber spôsobom drag-and-drop (obrázok č.45). V podobjekte GrabVolumeSmall sme odstránili vložený kolider a v podobjekte GrabVolumeBig sme veľkosť a umiestnenie kolideru nastavili podľa potreby. Úpravu sme vykonali z dôvodu interakcií s inými objektmi (napríklad tlačidlo, by pri stlačení registrovalo dva rôzne kolidery a teda reagovalo by, ako keby bolo dvakrát stlačené). Nakoniec sme v podobjekte hands:Lhand nastavili vlastnosti materiálu ruky. Rovnaký postup sme vykonali pre objekty oboch rúk (obrázok č.45).



Obr. 45. Definovanie parametra Parent Transform, nastavenie a prispôsobenie objektov.

Prostredie sme mali kompletne pripravené na tvorbu aplikačnej logiky. Pred začatím tvorby UI a písaním vlastných skriptov sme ešte otestovali funkčnosť vytvorenej aplikácie. Pre otestovanie chovania priebežného vývoja aplikácie sme prechádzali do Run módu herného engine. Run mód sme používali vždy po pridaní a aplikovaní nových objektov, elementov a funkčností. V ďalšom vývoji sme nepokračovali dovedy, pokiaľ sme nedosiahli požadované správanie našej aplikácie. Po otestovaní vytvoreného prostredia sme zistili chybné, nedostatočné renderovanie objektov rúk pri pohybe prostredím. To bolo spôsobené faktom, že Unity má primárne nastavené kontrolovanie (overovanie) fyziky 60FPS (rámcov za sekundu) a nie 90FPS, čo je potrebné pre virtuálnu realitu. Cez ponuku Edit a voľbu možnosti Project Setting sme sa dostali do okna, kde sme v ľavej časti vybrali možnosť Time. Do kolónky Fixed Timestep sme pôvodnú hodnotu 0,02 nahradili hodnotou 1/90, čo nám zabezpečilo sledovanie fyziky 90FPS, a tým sme odstránili vzniknutý problém (obrázok č.46).



Obr. 46. Nastavenie kontroly fyziky na hodnotu 90FPS.

Po úspešnom otestovaní jednotlivých funkčností sme mohli prejsť na vývoj aplikačnej logiky.

6.1.3 Tvorba UI a pomocných obsahov

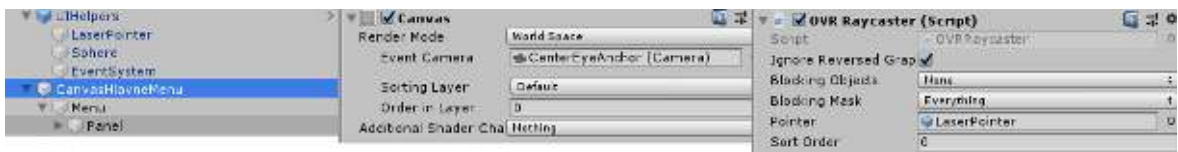
V predchádzajúcej časti sme podrobne popísali funkcie a postupy importu, pridávania objektov, pridávania komponentov, vkladania objektov do atribútov a ich slotov. Vymenované úkony sú pri tvorbe našej VR aplikácie vždy rovnaké, preto z dôvodu urýchlenia, skrátenia a zjednodušenia písania postupu tvorby aplikácie budeme používať nasledovné:

- 1) Import súborov - budeme používať import súborov do zložky (názov zložky).
- 2) Pridanie objektu na scénu - budeme používať pridanie objektu do hierarchie.
- 3) Pridávanie komponentov do štruktúry objektu - budeme používať pridanie komponentu a pridanie skriptu.
- 4) Vkladanie objektov, súborov alebo skriptov do požadovaných slotov atribútov - budeme používať vloženie objektu, vloženie súboru a vloženie skriptu.
- 5) Keď hovoríme o nastavení atribútov komponentov, vždy je to v Inspector paneli zvoleného objektu.
- 6) Keď hovoríme o vytvorení zložky alebo skriptu, je to vždy v štruktúre Project panela nášho projektu spôsobom pravé tlačidlo myši, Create, Folder alebo C# Script.

Ak použijeme v práci postup, ktorý sme ešte nepoužili, tak ho popíšeme podrobnejšie.

Začali sme vytvorením užívateľského rozhrania pre virtuálnu realitu. Do hierarchie sme pridali Canvas (Plátno), ktoré je objektom UI. Po pridaní Canvasu sa nám automaticky do hierarchie pridal aj EventSystem (systém slúžiaci na interakciu a obsluhu prvkov užívateľského rozhrania). Pridaný EventSystem neobsahoval komponenty a skripty na obsluhu VR užívateľského rozhrania, takže sme ho odstránili. Vytvorili sme si hierarchickú štruktúru objektu Canvas, ktorá sa skladala z podobjektov Quad, Panel a podľa ponuky možností Buttons. Objekty sme pridávali do Canvas ako jeho podobjekty. Najskôr sme nastavili hlavný objekt Canvas. Pre atribút Render Mode (zobrazovaný mód) sme nastavili možnosť World Space (umožnilo nám pracovať s Canvasom ako s akýmkoľvek iným pridaným 3D objektom, možnosť vykonania transformácií position, rotation, scale). Tým sme mohli objekt Canvasu umiestniť podľa potreby do priestoru výrobného prostredia. Ako druhé sme museli nastaviť atribút Event Camera (definoval kameru, cez ktorú sa spúšťali udalosti. V režime World Space sa používal na určenie polohy kliknutia a prekrytia tlačidiel UI). V našom prípade sme do slotu vložili objekt CenterEyeAnchor, ktorý bol súčasťou OVRPlayerControlleru. Poslednou vecou, aby sme mohli vytvárať s Canvasom a jeho ob-

jektmi interakciu, bolo odstránenie skriptu Graphic Raycaster z jeho štruktúry a nahradenie ho skriptom OVR Raycaster. Následne sme ešte do hierarchie pridali prefab objekt UIHelpers (súčasť Integration balíku). UIHelper obsahoval okrem LaserPointer a Sphere objektov aj EventSystem, ktorým sme nahradili pôvodný, už vymazaný EventSystem. Objekt LaserPointer sme vložili do slotu Canvas atribútu Pointer, ktorý obsahoval pridaný OVR Raycaster skript. Pre lepšiu viditeľnosť ukazovateľa LaserPointer sme upravili nastavenia objektu Sphere, ktorý plnil úlohu zakončenia ukazovateľa. Mali sme nastavený Canvas (obrázok č.47).



Obr. 47. Nastavenie Canvas a OVR Raycaster (Script).

Objekt Quad nám slúžil ako transparentná podkladová doska pre pridaný Panel. Nepotrebné komponenty, ktoré obsahoval, sme odstránili. Do nastavenia Quadu Anchor Presets sme nastavili middle, centre (prostredný, stred). Objekt Panel slúžil ako pozadie pre Canvas. Nastavili sme bielu farbu pozadia s alfa kanálom na hodnote sto. Do Panel sme pridali ešte jeden Panel (slúžil ako hlavička aktuálneho UI) a potrebné množstvo Button (tlačidlo) objektov, ktoré reprezentovali už jednotlivé tlačidlá ponuky vo vytváranom UI. Počet tlačidiel sme volili vždy podľa obsahu UI, ktorý sme vytvárali. V aplikácii Adobe Photoshop CS6 sme si vytvorili úplne jednoduché obrázky, ktoré sme použili ako Source Image (zdrojový obrázok) pre hlavičku a tlačidlá (obrázok č.48). Vytvorili sme novú zložku s názvom buttons a importovali sme do nej vytvorené obrázky. Keď sme ich chceli použiť ako Source Image pre hlavičku a tlačidlá, museli sme im definovať Texture Type ako Sprite (2D and UI), čo sme aj urobili. Obrázok *panel header* sme vložili do slotu Source Image hlavičky a obrázok *tlačitka* sme vložili do slotov Source Image jednotlivých tlačidiel. Jednotlivé tlačidlá a hlavičku sme si v Canvase usporiadali.



Obr. 48. Zdrojové obrázky pre hlavičku a tlačidlá UI.

Do objektu hlavičky a všetkých tlačidiel sme pridali podobjekt Text. Hlavičke a tlačidlám sme nastavovali text vždy podľa účelu, na ktorý mali slúžiť. Keďže sme ako prvé UI tvorili hlavnú ponuku nazvali sme hlavičku *HLAVNÍ NABÍDKA* (všetky zobrazované texty vo vyvíjanej VR aplikácii boli písané v českom jazyku, nakoľko bola vyvíjaná

pre českú výrobnú spoločnosť). Hlavičky sme nastavili atribúty Font Style na Bolt And Italic, Font Size 14, Alignment stred, stred a Color v hexadecimálnom tvare na FF0404 (červená). Mali sme nastavený objekt hlavičky. Prešli sme k nastaveniu tlačidiel. Objekty Text sme nechali nastavené na základných hodnotách, len sme zadali zobrazovaný text. V prípade rovnakého nastavenia pre všetky objekty Button, ako to bolo aj v tomto prípade, sme ich všetky označili a nastavili sme atribúty Highlighted Color na B1FFCD a Pressed Color na 06FF00. Keďže sme vedeli, že budeme implementovať aj funkcie a zvuky, pripravili sme si objekty tlačidiel pridaním dvoch slotov v Event systéme On Click(). Jeden slúžil na volanie funkcie zo skriptu a druhý pre zvukový zdroj. Pridali sme si komponent Event Trigger (Script), v ktorom sme zvolili Add New Event Type, a vybrali možnosť Pointer Enter (pridávalo nám to funkcionality, že keď sme na tlačidlo ukázali prostredníctvom LaserPointer, tak sa vykonala nami definovaná funkcionality, v tomto prípade sme definovali rovnako ako pri On Click, že sa jedná o zvukový zdroj). UI hlavnej ponuky sa skladalo z viacerých možností voľby, ktorými boli Spustiť simuláciu, Volný pohyb, Nastavenie zvuku, O aplikácii a Ukončiť aplikáciu. Pre každú z vymenovaných voľieb sme museli rovnako, ako v prípade hlavnej ponuky, vytvoriť samostatné UI. To sme vykonali rovnakým spôsobom, ako v prípade hlavnej ponuky s pár rozdielmi, kedy sme si UI upravili podľa našej potreby. Napríklad, keď sme chceli v UI zobraziť text, tak sme pridali objekt Quad a do neho Text, v ktorom sme zobrazovaný text zadali alebo sme pridali obrázok ako pozadie UI, alebo ako v prípade UI pre nastavenie zvukov, sme pridali objekty Slider a podobne. Všetky pridané objekty a podobjekty UI sme účelovo a zrozumiteľne nazvali, aby sme mohli pri tvorbe funkcií (kódovaní) k nim prostredníctvom mien pristupovať a pracovať s nimi. Vytvorili sme UI hlavnej ponuky (obrázok č.49). Čo však musíme podotknúť je, že rozhranie sme vytvorili zatiaľ iba z objektivej a vizuálnej stránky, nie však zo stránky funkčnej. Funkčnú stránku jednotlivých UI a ďalších objektových a grafických súčastí a elementov vyriešime v časti tvorby aplikačnej logiky spolu s celou aplikačnou logikou našej VR aplikácie.

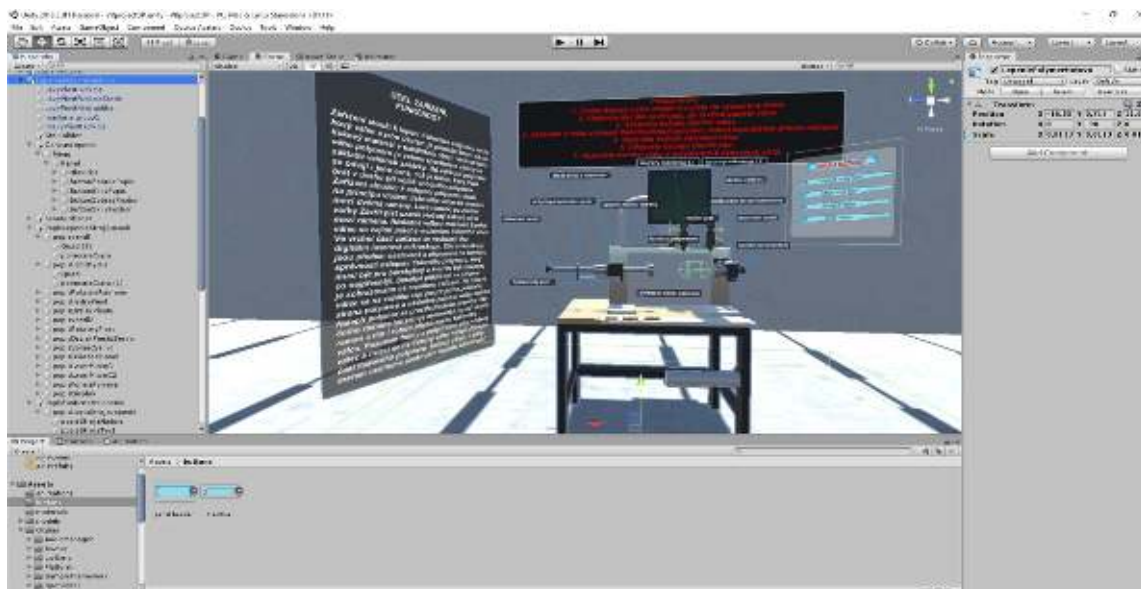


Obr. 49. Vytvorené VR UI hlavnej ponuky (objektová a vizuálna stránka).

Keďže naša VR aplikácia je zameraná na simulovanie výrobného procesu a jeho postupu a bude slúžiť na zaškoľovanie zamestnancov, ktorého súčasťou je aj vzdelávanie, rozhodli sme sa do riešenia simulácie pridať pomocné objekty a elementy (obsahy), ktoré budú slúžiť k lepšiemu pochopeniu fungovania zariadení, strojov a celého výrobného procesu. Zároveň budú slúžiť ako vzdelávacie prostriedok pre užívateľa. Spomínanými objektmi budú v našej aplikácii zobrazovacie pomocné popisy jednotlivých súčastí a komponentov, z ktorých sa zariadenia a stroje skladajú, zobrazovacie okná s popismi účelov a princípami funkčnosti jednotlivých zariadení a zobrazovacie okná s popismi krokov, ktoré treba vykonať pre úspešné zvládnutie výrobného procesu. Pomocné objekty (obsahy) sa budú zobrazovať po voľbe možností ich zobrazenia prostredníctvom UI príslušných hlavných zariadení. V prípade pomocného obsahu s postupom krokov sa bude tento zobrazovať na základe voľby typu simulácie v hlavnej ponuke.

Prešli sme na tvorbu štyroch UI a pomocných obsahov priradených k jednotlivým hlavným objektom, ktorými boli v našej aplikácii stroje a zariadenia, reprezentujúce stanoviská výrobného procesu a postupu (stojan na tlačové valce, stroj na lepenie tlačového polyméru, tlačiarenský stroj a regál na farebníkové zásobníky). Aby sme si uľahčili prácu, zdublikovali sme už vytvorený UI s názvom *CanvasHlavneMenu*. Upravili a umiestnili sme ho ku všetkým vymenovaným objektom podľa našej potreby a jeho využitia. Prešli sme na pridanie pomocných obsahov. V hierarchii sme označili objekt zariadenia na lepenie tlačových polymérov s názvom *LepeniePolymerHotovo*. Pridali sme dva prázdne podobjekty, ktoré sme nazvali *PopisFunkčnostiLepenie* a *PopisLepenieStrojSucasti*. Označili sme si prázdny objekt *PopisFunkčnostiLepenie* a pridali sme mu podobjekty Quad (slúžil ako podkladová doska pre text) a Text (zadanie zobrazovaného textu s popisom funkčnosti zariadenia). Označili sme si objekt *PopisLepenieStrojSucasti* a pridali sme mu rovnako podobjekty Quad (podklad), Text (popis súčasti) a navyše Plane (tvar sme upravili transformáciami, aby vyzeral ako dlhá čiara, a farbu sme zmenou nastavenie atribútu Albedo nastavili na hodnotu 00FFFF, tyrkysová). Veľkosť a umiestnenie sme upravili tak, aby jeden koniec vytvorenej čiary (Plane) ukazoval na časť popisovaného objektu, a druhý koniec ukazoval na vytvorený príslušný textový popis súčasti (Quad+Text). Objekty sme pomenovali podľa súčastí na ktoré ukazovali. V objekte *PopisLepenieStrojSucasti* sme spôsobom duplikovania vytvorili potrebný počet popisov, ktoré sme pomenovali a umiestnili tak, aby boli týmto spôsobom popísané všetky dôležité časti zariadenia. Rovnaký alebo podobný postup sme použili aj pri tvorbe pomocného obsahu s popismi krokov,

ktoré treba vykonať. Rovnaký postup tvorby pomocných obsahov sme použili pre všetky hlavné zariadenia a ich stanoviská. Mali sme vytvorené UI aj s pomocnými obsahmi (obrázok č.50).



Obr. 50. UI s pomocnými obsahmi pre zariadenie na lepenie tlačových polymérov.

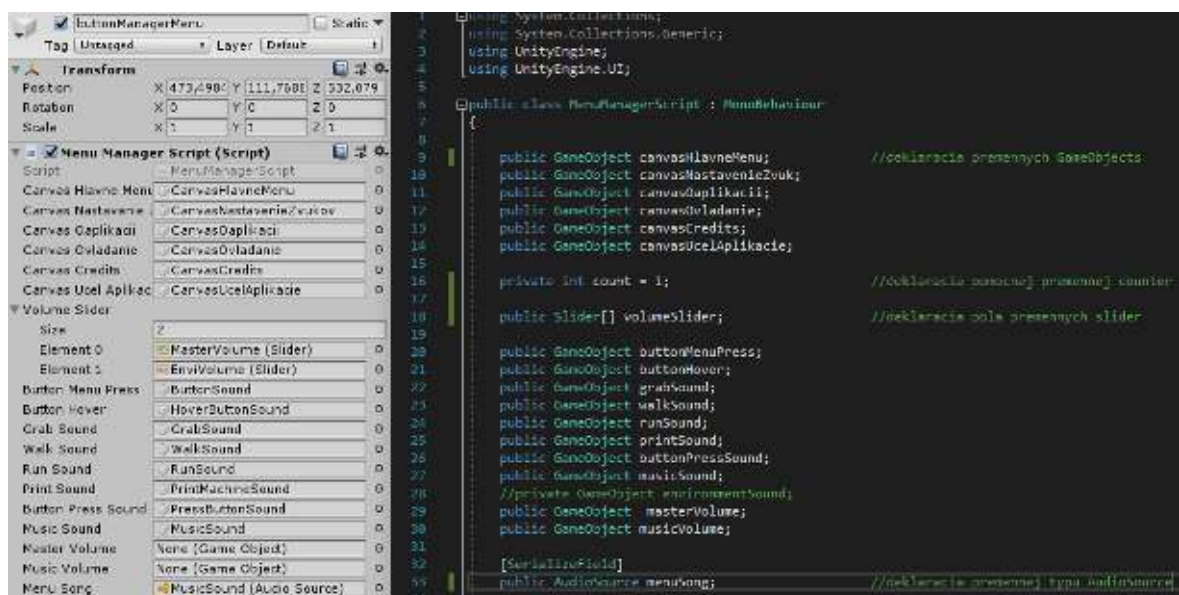
6.2 Tvorba aplikačnej logiky pre vytvorenie požadovanej simulácie

Mali sme vytvorené prostredie, všetky jeho súčasti a potrebné komponenty vrátane UI. Aplikáciu sme mali pripravenú na tvorbu aplikačnej logiky. Pod aplikačnou logikou rozumieme vytvorenie skriptov a pomocou programovania v zvolenom IDE vytvoriť funkčnosti jednotlivých zariadení, interakcií užívateľa s prostredím a podobne. Skripty sme vytvárali priamo v hernom engine Unity a kódy sme písali v IDE Microsoft Visual Studio 2017 v programovacom jazyku C#.

Pri vývoji a tvorbe VR aplikácie sme v prípade potreby čerpali informácie z publikácií a manuálov určených pre vývoj v nami zvolených vývojárskych nástrojoch [2][5][6][8].

Začneme postupom základného princípu vytvárania logiky v Unity. Tieto princípy sa v našej práci opakovali. Začali sme vytvorením funkčnosti hlavnej ponuky. Systém Unity fungoval tak, že sme si vytvorili nový prázdny objekt v hierarchii, nazvali sme ho *buttonManagerMenu*. V Project panely sme vytvorili zložku s názvom *scripts*. V zložke sme pridali nový C# skript, ktorý sme pomenovali *MenuMangerScript*. V Inspector panely objektu *buttonManagerMenu* sme pridali vytvorený skript *MenuMangerScript*. Po dvojkliku na pridaný skript sa nám otvorilo IDE MS Visual Studio (ďalej v práci budeme používať

skratku IDE) so základnou triedou podľa názvu skriptu a funkciami Start() a Update(). Obe funkcie sme v našej práci využívali. Funkcia Start() inicializovala naprogramované funkcie, objekty a hodnoty pri štarte aplikácie ak bol skript implementovaný v nejakom objekte a funkcia Update() bola volaná raz za jeden rámeček, počas celého behu aplikácie (využívali sme keď sme potrebovali sledovať hodnotu vybraných premenných a na základe zmeny hodnoty vykonať definovanú funkciu). Skriptom *MenuMangerScript* sme chceli zabezpečiť funkčnosť pre jednotlivé UI hlavnej ponuky, vytvorenia funkcií pre použitie zvukových záznamov vrátane nastavenia a ovládania hlasitosti zvukov, zamedzenie pohybu pred voľbou *Spustit simulaci* alebo *Volný pohyb*, zobrazovanie hlavnej ponuky po stlačení definovaného tlačidla na ovládači. Pre ovládanie objektov z hierarchie nášho projektu sme v triede skriptu vytvorili verejnú (public) premennú *GameObject*, ktorá sa nám vo vloženom skripte *MenuMangerScript* pridala ako atribút (premenná), do ktorého môžeme vložiť nami zvolený objekt z hierarchie projektu a tak definovať pri programovaní funkcií v skripte o aký objekt sa jedná. Týmto spôsobom si vieme pridať objekty modelov, komponentov, súčastí, zvukových objektov, objektov UI a ďalších. Ak by sme však deklarovali *GameObject* ako *private*, nezobrazil by sa nám ako dostupný atribút (premenná) a priradenie objektu z hierarchie by sme museli spraviť kódovo v skripte (*private GameObject priklad = GameObject.Find(“názov objektu z hierarchie”);*). V skripte *MenuMangerScript* sme vytvorili public *GameObject* pre jednotlivé vytvorené UI (Canvas) hlavnej ponuky, objekty zvukových súborov a UI sliderov na ovládanie zvukov (obrázok č.51).



Obr. 51. Deklarácia požadovaných public premenných v skripte a vloženie objektov do jednotlivých slotov atribútov (premenných).

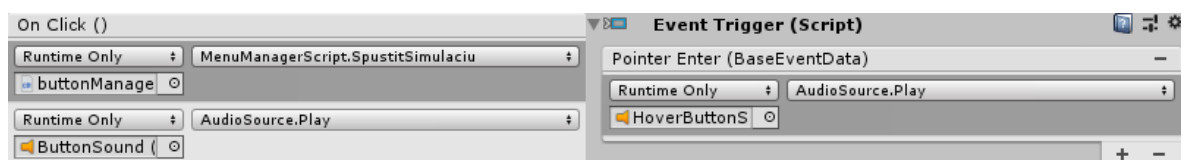
Funkčnosť jednotlivých UI a teda ich ovládanie, preklikávanie a zobrazovanie funguje na jednoduchom kódovom princípe. Vo funkcií Start() volaním funkcie (*canvasHlavnéMenu.SetActive(true);*) sme nastavili menovanému objektu aby bol pri štarte aplikácie zapnutý. Všetky zvyšné Canvas objekty sme nastavili pri štarte ako vypnuté (*false*). Ako ďalšie sme v skripte definovali jednoduché verejné funkcie bez návratovej hodnoty a vstupnej premennej, v ktorých sme len prostredníctvom volania funkcie *SetActive* a zmenou hodnôt medzi *true/false* vytvorili funkcie pre zobrazovanie a skrývanie požadovaných Canvas objektov (obrázok č.52). Týmto spôsobom sme vytvorili funkcie pre všetky tlačidlá, ktoré sa v Canvas objektoch UI hlavnej ponuky nachádzali.

```
void Start()
{
    canvasHlavneMenu.SetActive(true);
    canvasNastavenieZvuk.SetActive(false);
    canvasOplikacii.SetActive(false);
    canvasOvladanie.SetActive(false);
    canvasCredits.SetActive(false);
    canvasUcelAplikacie.SetActive(false);
}

public void Oplikacii()
{
    canvasOplikacii.SetActive(true);
    canvasHlavneMenu.SetActive(false);
}
```

Obr. 52. Volanie funkcií *SetActive* pre všetky objekty Canvas v hlavnej ponuke vo funkcií Start() a definícia funkcie na prechod medzi ponukami.

Keď sme chceli vytvorené funkcie použiť pre jednotlivé tlačidlá Canvasov, označili sme si napríklad v hierarchii objekt tlačidla O aplikácii (*ButtonOplikacii*) a v Inspector paneli, v pridanom On Click() Evente sme do voľného atribútu vložili objekt *buttonManagerMenu* (ktorý mal vo svojej štruktúre náš *MenuMangerScript*). Následne sme mohli po kliknutí vo vedľajšej ponuke vybrať *MenuManagerScript* a vybrať funkciu *Oplikacii()* (obrázok č.53). Týmto spôsobom sme pridali funkcie všetkým UI (ich tlačidlám) v celej našej aplikácii. Rovnakým spôsobom sme pridali aj zvuky do všetkých tlačidiel v aplikácii s rozdielom, že do slotov Eventov (On Click(), Pointer Enter()) sme vkladali objekty požadovaných zvukov. Funkcie pre prehrávanie zvukov sme písať nemuseli, pretože objekty sme vytvorili ako *Audio Source* a tie už majú funkcie prehrávania, zastavenia implementované (obrázok č.53).



Obr. 53. Nastavenie vytvorenej funkcie objektu tlačidla O aplikácii a pridanie zvukov.

Pehrávanie ďalších zvukov sme vyriešili jednoduchým spôsobom deklarácie *public AudioSource* premennej vo vybranom skripte, vložení do slotu požadovaný zvuk

a následne v skripte volaním funkcií *premenna.Play()*; a *premenna.Stop()*; sme zvuky púšťali a zastavovali. Týmto spôsobom fungovalo aj prehrávanie vytvorených animácií, ktoré sme použili pre stlačenie tlačidiel panela obsluhy stroja na lepenie tlačových polymérov s rozdielom, že sme nedeklarovali premennú typu Audio Source ale Animator.

Pokračovali sme v riešení funkcií hlavnej ponuky. Prvé dve možnosti v ponuke mali iné funkcie ako ostatné. Pre obe možnosti *Spustit simulaci* a *Volný pohyb* sme potrebovali nastaviť aby pred ich voľbou nebol možný pohyb užívateľa prostredím. To sme mohli spraviť úpravou prefab skriptu *OVRPlayerController*, ktorý bol implementovaný ako súčasť štruktúry prefab objektu *OVRPalyerController* (pridaný v hierarchii). Aby sme mohli úpravu spraviť, potrebovali sme vytvoriť špeciálnu premennú, ktorá bola viditeľná v rámci celého projektu a dostupná z akéhokoľvek skriptu implementovaného v aplikácii. Na tento účel sme mali triedu *PlayerPrefs* aj s funkciami. Prostredníctvom použitia tejto metódy sme vedeli vytvoriť premennú (uloženú v externom súbore) do ktorej sme pomocou funkcie *Set(Int/Float)* vedeli uložiť hodnoty typu int, float a pomocou funkcie *Get(Int/Float)* nastavené hodnoty kedykoľvek získať. Najdôležitejšie bolo deklarovanie premennej ktoré sa robilo na základe zadaného (string) mena. Premennú sme vytvorili v skripte *MenuManagerScript*, funkcií *Start()* s integer hodnotou nastavenou na nula (obrázok č.54). Do funkcie tlačidla *SpustitSimulaciu()* sme pridali *PlayerPref*, ktorý vytvorenej premennej po voľbe tlačidla nastavoval novú int hodnotu (jedna) (obrázok č.54).

```
void Start()
{
    PlayerPrefs.SetInt("ZvolenaSimulacia", 1);
    PlayerPrefs.SetInt("PohybMozny", 0);
}

public void SpustitSimulaciu()
{
    canvasHlavneMenu.SetActive(false);
    PlayerPrefs.SetInt("ZvolenaSimulacia", 0);
    PlayerPrefs.SetInt("PohybMozny", 1);
}
```

Obr. 54. Vytvorenie a nastavenie špecialnej premennej *PlayerPref* s názvom *PohybMozny*.

Následne sme si v skripte *OVRPlayerController* deklarovali premennú typu bool s názvom *pohybMozny* a vo funkcií *Update()* sme premennej nastavovali hodnotu na základe príkazu (*pohybMozny = PlayerPrefs.GetInt("PohybMozny") == 1;*), čo bolo v podstate porovnávanie hodnoty (overovanie pravdivosti) v premennej *PohybMozny* a hodnoty jedna, pričom ak boli hodnoty rovnaké do bool lokálnej premennej sa zapísala hodnota *true* inak *false*. Nakoniec sme v skripte našli funkciu na pohyb užívateľa a tú sme zavreli do podmienky *if (pohybMozny)* ktorá v prípade *true* pohyb umožnila, inak nie. Rovnakým spôsobom sme v celom projekte riešili podobné situácie, keď sme potrebovali povoliť/zakázať funkčnosť objektov v rámci viacerých skriptov, alebo overovať výber a voľbu užívateľa naprieč simuláciou, ale aj ďalšie iné situácie. Príkladom môže byť zobrazovanie pomocné-

ho obsahu s postupom krokov na základe voľby *Spustit simulaci* (obsah sa zobrazuje) alebo *Volný pohyb* (obsah sa nezobrazuje). Triedu s funkciami sme používali aj keď sme potrebovali uložiť a načítať polohu a rotáciu objektov medzi viacerými skriptami.

Funkciu na zobrazenie hlavnej ponuky na základe stlačenia definovaného tlačidla ovládača sme vyriešili vo funkcii `Update()` skriptu *MenuMangerScript*. Použili sme triedu a funkciu na sledovanie stlačených tlačidiel ovládača *OVRInput.GetUp(OVRInput.Button.Two)* pričom sme definovali tlačidlo dva. Zobrazovanie UI sme vyriešili použitím premennej counteru, ktorej hodnotu sme pri každom stlačení tlačidla navyšovali o jedna a následne sme delili modulo dva. Na základe zostatku sme vykonávali zobrazovanie a skrývanie UI (obrázok č.55).

```

if (OVRInput.GetUp(OVRInput.Button.Two))
{
    count += 1;
    if (count % 2 == 0)
    {
        canvasHlavneMenu.SetActive(true);
    }
    else
    {
        canvasHlavneMenu.SetActive(false);
    }
}

```

Obr. 55. Zobrazovanie hlavnej ponuky.

Nastavenie zvukov sme vyriešili pomocou použitia tried, funkcií a premených typu `AudioSource`, `PlayerPrefs` a `Slider` v skripte *MenuMangerScript* (obrázok č.56). Po nakódovaní sme v hierarchii označili podobjekty typu `Slider` *MasterVolume* a *EnvVolume*, ktoré patrili pod *CanvasNastavenieZvukov*. Do Eventov (`On Value Changed()`) sme vložili objekt `buttonManagerMenu` a vybrali vytvorené funkcie *SetMasterVolume* a *SetMusicVolume* (typu `Dynamic float`). To nám umožnilo nastavovať hlasitosť jednotlivých zvukov v našej aplikácii.

```

public void NastavnieZvukov()
{
    canvasNastavenieZvukov.SetActive(true);
    canvasHlavneMenu.SetActive(false);

    if (masterVolume != null)
    {
        masterVolume.GetComponent<Slider>().normalizedValue = PlayerPrefs.GetFloat("MasterVolume", 1.0f);
    }
    else
    {
        PlayerPrefs.SetFloat("MasterVolume", 1.0f);
    }

    if (musicVolume != null)
    {
        musicVolume.GetComponent<Slider>().normalizedValue = PlayerPrefs.GetFloat("MusicVolume", 1.0f);
    }
    else
    {
        PlayerPrefs.SetFloat("MusicVolume", 1.0f);
    }
}

public void SetMasterVolume(float value)
{
    PlayerPrefs.SetFloat("MasterVolume", value);
    UpdateSoundLevel();
}

public void SetMusicVolume(float value)
{
    PlayerPrefs.SetFloat("MusicVolume", value);
    UpdateSoundLevel();
}

public void UpdateSoundLevel()
{
    buttonManagerMenu.GetComponent<AudioSource>().volume = PlayerPrefs.GetFloat("MasterVolume", 1.0f);
    buttonManagerMenu.GetComponent<AudioSource>().volume = PlayerPrefs.GetFloat("MusicVolume", 1.0f);
    grassSound.GetComponent<AudioSource>().volume = PlayerPrefs.GetFloat("MasterVolume", 1.0f);
    wallSound.GetComponent<AudioSource>().volume = PlayerPrefs.GetFloat("MasterVolume", 1.0f);
    rainSound.GetComponent<AudioSource>().volume = PlayerPrefs.GetFloat("MasterVolume", 1.0f);
    fireSound.GetComponent<AudioSource>().volume = PlayerPrefs.GetFloat("MasterVolume", 1.0f);
    buttonManagerMenu.GetComponent<AudioSource>().volume = PlayerPrefs.GetFloat("MasterVolume", 1.0f);
    musicSound.GetComponent<AudioSource>().volume = PlayerPrefs.GetFloat("MusicVolume", 1.0f);
    //buttonManagerMenu.GetComponent<AudioSource>().volume = PlayerPrefs.GetFloat("MasterVolume", 1.0f);
}

```

Obr. 56. kódy pre nastavenia zvukov použitých v aplikácii.

Prešli sme na riešenie ďalšej funkcionality, ktorou bolo zobrazovanie jednotlivých UI hlavných zariadení na základe vstupu a výstupu do ich spúšťačích koliderov. Pre roz-

poznanie vstupu, výstupu alebo zotrvaníu v kolideroch sme mali k dispozícii funkcie zakomponované v štruktúre koliderov. Vstup funkcia (*OnTriggerEnter(Collider other)*), Zotrvanie funkcia (*OnTriggerStay(Collider other)*) a výstup funkcia (*OnTriggerExit(Collider other)*). Využitím funkcie *SetActive()* vnorenej vo vymenovaných funkciách koliderov sme dosiahli zobrazovanie a skrývanie UI (obrázok č.57). Rovnaký spôsob implementácie sme použili pre všetky UI hlavných objektov.

```

void OnTriggerEnter(Collider other)
{
    zvolenaSimulacia = PlayerPrefs.GetInt("ZvolenaSimulacia") == 0;
    canvasToolStojan.SetActive(true);
    if (zvolenaSimulacia)
    {
        canvasPostupKrokovStojan.SetActive(true);
    }
}

void OnTriggerExit(Collider other)
{
    canvasToolStojan.SetActive(false);
    if (zvolenaSimulacia)
    {
        canvasPostupKrokovStojan.SetActive(false);
    }
}

```

Obr. 57. Použitie funkcií kolideru na zobrazovanie a skrývanie UI (Canvas).

Aktiváciu a zobrazovanie objektov *ParticleSystem* slúžiacich na zvýraznenie voľby užívateľa sme robili už spomenutými postupmi funkciou *SetActive()* a pre uloženie voľby a jej ďalšie využitie postupom uloženia hodnôt a stavov do premenných *PlayerPrefs*.

Keďže v aplikácii vytvárame množstvo rôznych transformácií na základe interakcie užívateľa s prostredím, ako napríklad voľba možností z ponuky UI pre obsluhu tlačiarenského stroja, tak si ako poslednú časť vývoja aplikačnej logiky v našej práci popíšeme spôsob vytvorenia transformácie vybraného objektu. Spôsob tvorby skriptu vrátane deklarácií premenných a priradenia objektov premenných je rovnaký, ako postup, ktorý sme v práci popísali a použili. Ako príklad použijeme tvorbu transformácie pre istiaci stredový panel. Objekt panelu z hierarchie sme mali pridaný do premennej *GameObject strednýIstiaciPanelVyklápací*. Panelu sme potrebovali naprogramovať dve na seba nadväzujúce transformácie (zmena polohy objektu a následná rotácia). Vytvorili sme ďalšie premenné, ale typu *private Vector3 poziciaIstiacehoPanelu* a *rotaciaIstiacehoPanelu* (premenná pre pozíciu slúžila na uchovanie x,y,z hodnôt position objektu na scéne a premenná pre rotáciu na uchovanie x,y,z hodnôt rotation). Doplňujúcim objektom bol objekt *Capsule*, ktorý bol podobjektom *strednýIstiaciPanelVyklápací* a bol deklarovaný ako objekt typu *Transform*. Pridaný objekt slúžil ako pomocný bod okolo ktorého objekt panelu rotoval. Jednotlivé transformácie sme programovali prostredníctvom využitia vymenovaných vektorových premenných a pomocného objektu *Capsule*. Obe transformácie sme chceli vykonať iba do určitého stavu (pozície). Preto sme si vytvorili pre každú transformáciu dve pomocné premenné typu *float*, do ktorých sme zadali maximalne možné hodnoty jednotlivých polôh pa-

nela (rotačných a pozičných). Hodnoty požadovaných polôh sme zistili na základe použitia funkcie DebugLog() pre vybraný objekt. Tiež sme vytvorili dve pomocné premenné typu bool aby sme vedeli kedy panel otvoriť a kedy zavrieť. Pomocou deklarovaných premenných sme transformácie nakódovali (vysvetlenie kódu v komentároch)(obrázok č.58).

```

void Start()
{
    // Nastavenie pozície a rotácie objektu
    // Pozícia objektu
    transform.position = new Vector3(0, 0, 0);
    // Rotácia objektu
    transform.rotation = Quaternion.Euler(0, 0, 0);
}

// Metóda na otvorenie panelu
void OpenPanel()
{
    // Nastavenie pozície objektu
    // Pozícia objektu
    transform.position = new Vector3(0, 0, 0);
    // Rotácia objektu
    transform.rotation = Quaternion.Euler(0, 0, 0);
}

// Metóda na zatvorenie panelu
void ClosePanel()
{
    // Nastavenie pozície objektu
    // Pozícia objektu
    transform.position = new Vector3(0, 0, 0);
    // Rotácia objektu
    transform.rotation = Quaternion.Euler(0, 0, 0);
}

```

Obr. 58. Naprogramovanie transformácií pozície a rotácie objektu stredového panelu.

Na tvorbu všetkých transformácií sme používali rovnaké alebo podobné postupy (podľa potreby), ako v prípade transformácií pozície a rotácie objektu stredového panelu.

Použitím vymenovaných ale aj ďalších programovacích postupov, metód a techník sme postupovali pri tvorbe celej funkčnej stránky našej VR aplikácie. V prípade potreby použitia neznámych funkcií a postupov sme čerpali informácie z publikácií a manuálov určených pre nami zvolené vývojové prostredie a programovací jazyk [2][5][6][8].

Po vytvorení všetkých častí našej aplikácie vrátane aplikačnej logiky a otestovaní funkčností sme mohli prejsť na posledný krok a to vybudovanie koncovej VR aplikácie. To sme spravili jednoduchým postupom voľby možnosti File (Súbor) z hlavnej ponuky Unity a výberom možnosti Build Settings (nastavenia budovania). Do pola Scenes In Build (Scény v BUILDE) sme spôsobom drag-and-drop presunuli našu hlavnú scénu z Hierarchy panelu s názvom VRprojectDP. Po pridaní scény do buildu sme ju ešte potvrdili (zaklikli štvorček, aby v ňom bol potvrdzujúci znak). Platformu sme ponechali na PC, Mac & Linux Standalone, Target Platform (cieľová platforma) Windows a Architecture (architektúra) x86_64. Potvrdili sme možnosť Build (budovať).

Vybudovanú VR aplikáciu sme ešte z dôvodu overenia správnej funkčnosti odskúšali. Aplikácia je plne funkčná a pripravená na nasadenie do reálnej prevádzky.

ZÁVER

Cieľom diplomovej práce bolo oboznámenie sa s technológiou virtuálnej reality, definovaním a rozborom technológie, princípmi jej fungovania na základe použitého hardvéru, foriem VR a nástrojov, potrebných pre jej tvorbu a používanie. Cieľom bolo aj oboznámenie sa s nedostatkami technológie a spôsobmi a možnosťami jej využitia. Následne na základe získaných vedomostí a vybraných nástrojov bolo cieľom práce vytvoriť funkčnú VR aplikáciu pre potreby zaučovania nových zamestancov v prostredí vybranej výrobnéj spoločnosti.

V úvode práce sme sa oboznámili s technológiou virtuálnej reality. Na základe literárnych zdrojov sme ju definovali, porovnali s rozšírenou realitou, určili jej miesto v koncepte zmiešanej reality, poskytli prehľad stručnej histórie vývoja technológie a dostali sa až k súčasnému stavu. Súčasťou oboznámenia sa bola potreba pochopenia princípu funkčnosti technológie, hardvérových a softvérových prostriedkov pre beh a tvorbu VR, rozdelenia foriem VR až po pochopenie jej nedostatkov a možností jej využitia, čo sme tiež na základe štúdia literárnych zdrojov uskotočnili. Pre úspešné splnenie hlavného cieľa, ktorým bolo vytvorenie funkčnej VR aplikácie, simulujúcej vybraný výrobný proces, sme na základe vykonania analýzy potrebovali vybrať vhodnú výrobnú spoločnosť pre nasadenie aplikácie. Pre tento účel a overenie vhodnosti voľby vybranej výrobnéj spoločnosti sme vykonali SWOT analýzu, ktorá nám vhodnosť nasadenia potvrdila. Následne sme vykonali výber, analýzu a rozbor vhodného výrobného procesu a jeho postupu, ktorý bol konzultovaný a одобrený predstaviteľmi vybranej spoločnosti. Z dôvodu tvorby aplikácie sme potrebovali vybrať a popísať vývojové nástroje, prostredníctvom ktorých sme VR aplikáciu vytvorili. Prešli sme k praktickej časti práce, pričom sme začali návrhom riešenia, v ktorom sme zvolili formu VR, porovnali a na základe porovnania a našich požiadaviek vybrali VR zariadenia, bez ktorých by vývoj ani beh aplikácie nebol možný. Odôvodnili sme voľbu jednotlivých vybraných nástrojov pre tvorbu a navrhli sme simuláciu samotného výrobného procesu vo vyvinutej aplikácii. Mali sme pripravené všetko potrebné pre úspešné zvládnutie hlavného cieľa. Pomocou zvolených vývojových nástrojov sme začali s tvorbou 3D modelov zariadení, komponentov, súčastí a prostredia. Po vymodelovaní sme prešli k textúrovaniu vytvorených 3D objektov, ktorého súčasťou bola aj tvorba UV máp. Všetky vytvorené súčasti sme implementovali do vývojového prostredia, v ktorom sme simuláciu výrobného procesu vytvorili a naprogramovali. Aplikáciu a jej funkčnosť sme testovali priebežne počas celého vývoja, čím sme sa snažili aj progra-

movo ošetrovať možné neželané stavy a scénare, ktoré by počas používania mohli nastať. Všetky stavy, ktoré môžu nastať sme neošetrili, keďže na časť z nich sa príjde až samotným nasadením a používaním aplikácie v reálnej prevádzke. Ďalším nedostatkom boli chyby na textúrach, modely mali tiež svoje chyby. Z hľadiska splnenia hlavného cieľa, a to vytvorenia VR aplikácie, simulujúcej vybraný výrobný proces a jeho postup, môžeme zhodnotiť, že aplikácia je aj napriek pár nedostatkom funkčná a schopná nasadenia do prevádzky.

Najťažšou a najdlhšou časťou práce bolo pri samotnej tvorbe VR aplikácie čo najpresnejšie modelovanie všetkých 3D modelov, tvorba UV máp a ich textúrovanie. Veľkým plusom pri jednotlivých úkonoch bola naša predchádzajúca skúsenosť s menovanými činnosťami, metódami a postupmi v rámci vybraných nástrojov. Do budúcnosti by bolo možné aplikáciu rozšíriť o detailné rozloženie a rozbor jednotlivých strojov a ich častí s doplnením animácií a informácií o najčastejších závadách, čo by vo svojej podstate dopĺňalo aplikáciu o ďalší spôsob využitia VR, a to ako servisný manuál. Aplikácia by sa tiež mohla rozšíriť o simulácie ďalších výrobných procesov a ich postupov vybranej výrobnéj spoločnosti. Vysvetlenie prekročenia maximálneho stanoveného rozsahu pre diplomovú prácu: Maximálny stanovený rozsah našej diplomovej práce sme prekročili z dôvodu, že okrem literárnej rešerše technológie sme museli v práci vykonať dôkladné analýzy a rozbor výrobného prostredia, procesu a jeho postupu a tiež sme musely uskutočniť popis jednotlivých vybraných vývojových nástrojov. Dôvodom prekročenia rozsahu práce bol aj fakt, že praktická časť práce sa skladala z viacerých samostatných častí vývoja konečnej aplikácie a to tvorby 3D modelov, tvorby UV máp, textúrovania a nakoniec implementácie vytvorených obsahov a vývojom konečnej aplikácie. V jednotlivých častiach sme sa snažili postupy a procesy tvorby popísať stručne, ale zároveň tak, aby splnili požadovanú odbornosť pre diplomovú prácu. Posledný dôvodom je množstvo obrázkov (screenshotov), ktoré práca obsahuje (sú grafickou reprezentáciou vykonávaných postupov).

ZOZNAM POUŽITEJ LITERATÚRY

- [1] MIHELJ, Matjaž, Domen NOVAK a Samo BEGUŠ. *Virtual Reality Technology and Applications*. Dordrecht [Netherlands]: Springer, 2014, 231 s. ISBN 978-94-007-6910-6.
- [2] PARISI, Tony. *Learning Virtual Reality*. Sebastopol: O'Reilly Media, 2015, 172 s. ISBN 063-6-920-03846-7.
- [3] BRITO, Allan. *Blender Quick Start Guide*. Birmingham [UK]: Packt Publishing, 2018, 194 s. ISBN 978-1-78961-947-8.
- [4] BLAIN, John M. *The complete guide to Blender graphics: computer modeling & animation*. Fourth edition. Boca Raton, FL: CRC Press, 2017, 697 s. ISBN 978-1138081914.
- [5] OKITA, Alex. *Learning C# programming with Unity 3D*. Boca Raton: CRC Press, 2015, 690 s. ISBN-13: 978-1-4665-8653-6.
- [6] LINOWES, Jonathan. *Unity Virtual Reality Projects*. Second Edition. Birmingham [UK]: Packt Publishing, 2018, 492 s. ISBN 978-1-78847-880-9.
- [7] Blender 2.79 Reference Manual. [online]. *BLENDER Foundation*, 2018 [cit.2018-11-26]. Dostupné z: <https://docs.blender.org/manual/en/latest/>
- [8] Unity User Manual. [online]. *UNITY Technologies*, 2018 [cit.2018-11-26]. Dostupné z: <https://docs.unity3d.com/Manual/index.html>
- [9] OKECHUKWU, Moses a Felista UDOKA. *Understanding Virtual Reality Technology: Advances and Applications*. SCHMIDT, Matthias, ed. *Advances in Computer Science and Engineering* [online]. InTech, 2011, 2011-03-22 [cit. 2019-02-27]. DOI: 10.5772/15529. ISBN 978-953-307-173-2. Dostupné z: <http://www.intechopen.com/books/advances-in-computer-science-and-engineering/understanding-virtual-reality-technology-advances-and-applications>
- [10] What is Virtual Reality. [online]. *Virtual Reality Society*, 2017 [cit.2019-02-27]. Dostupné z: <https://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html>
- [11] ŠUBÁK, Michael. ČO JE VIRTUÁLNA REALITA. [online]. *VIRTÁLNAREALITA.EU*, 2016 [cit.2019-02-27]. Dostupné z: <https://virtualnarealita.eu/co-je-virtualna-realita/>

- [12] BILLINGHURST, Mark, CLARK, Adrian, LEE, Gun. A Survey of Augmented Reality. [New Zealand]: *Foundations and Trends® in Human-Computer Interaction* 2015, **8**(2-3), 73-272. DOI: 10.1561/11000000049.
- [13] CRAIG, Alan B. *Understanding augmented reality: concepts and applications*. Amsterdam: Morgan Kaufmann, [2013]. ISBN 978-0-240-82408-6.
- [14] MILGRAM, Paul, KISHINO, Fumio. A Taxonomy of Mixed Reality Visual Displays. [online]. *IEICE Transactions on Information Systems*, 1994 [cit. 2019-02-27]. Dostupné z: [https://www.researchgate.net/profile/Paul_Milgram/publication/231514051_A_Taxonomy-of-Mixed-Reality-Visual-Displays.pdf?origin=publication_detail](https://www.researchgate.net/profile/Paul_Milgram/publication/231514051_A_Taxonomy_of_Mixed_Reality_Visual_Displays/links/02e7e52ade5e1713ea000000/A-Taxonomy-of-Mixed-Reality-Visual-Displays.pdf?origin=publication_detail)
- [15] SEREIKA, Markus. Mixed Reality vs. Mixing Realities. [online]. *Mixedrealities*, 2008 [cit.2019-02-27]. Dostupné z: <http://www.mixedrealities.de/>
- [16] MELOV, Gonzalo. HMD – History and Objectivities of Inventions. [online]. *GALSSDEVELOPMENT WORDPRESS*, 2014 [cit.2019-02-28]. Dostupné z: <https://glassdevelopment.wordpress.com/2014/04/17/hmd-history-and-objectives-of-inventions/>
- [17] BOAS, Yuri, Antonio, Goncalves, Vilas. Overview of Virtual Reality Technologies. [online]. *School of Electronic and Computer Science, Semantic scholar*, 2013 [cit.2019-02-28]. Dostupné z: <https://pdfs.semanticscholar.org/4214/cb09e29795f5363e5e3b545750dce027b668.pdf>
- [18] MAZURYK, Tomasz, GERVAUTZ, Michael. Virtual Reality – History, Applications, Technology and Future. [online]. *Institute of Computer Graphics, Vienna University of Technology*, 1996 [cit.2019-02-28]. Dostupné z: <https://www.cg.tuwien.ac.at/research/publications/1996/mazuryk-1996-VRH/TR-186-2-96-06Paper.pdf>
- [19] STEINICKE, Frank. *Being Really Virtual: Immersive Natives and the Future of Virtual Reality*. [online]. New York, NY: Springer Berlin Heidelberg, 2016. [cit.2019-03-04]. Dostupné z:

- https://www.springer.com/cda/content/document/cda_downloaddocument/9783319430768-c2.pdf?SGWID=0-0-45-1593315-p180166446.
- [20] MANSUROV, Nasim. What is Distortion?. [online]. *Photography Life*, 2013 [cit.2019-03-05]. Dostupné z: <https://photographylife.com/what-is-distortion>
- [21] Wikipedia. Parallax. [online]. *Wikipedia*, 2017 [cit.2019-02-27]. Dostupné z: <https://en.wikipedia.org/wiki/Parallax>
- [22] SHAFQAT, Saad, Ch. Input Devices for Virtual Reality (VR) Systems. [online]. *Techooid*, 2018 [cit.2019-03-07]. Dostupné z: <https://techooid.com/input-devices-vr>
- [23] STRICKLAND, Jonathan. How Virtual Reality Gear Works. [online]. *Electronics, Howstuffworks*, 2007 [cit.2019-03-07]. Dostupné z: <https://electronics.howstuffworks.com/gadgets/other-gadgets/VR-gear6.htm>
- [24] BIZNÁR, Michal. História virtuálnej reality – od fantastických snov po únik zo skutočnosti. [online]. *TECHBOX*, 2018 [cit.2019-02-27]. Dostupné z: <https://techbox.dennikn.sk/temy/historia-virtualnej-reality-od-fantastickyh-snov-po-unik-zo-skutocnosti/>
- [25] HOFFMAN, Chris. What Is the “Screen Door Effect“ in VR?. [online]. *How-To Geek*, 2019 [cit.2019-03-07]. Dostupné z: <https://www.howtogeek.com/404491/what-is-the-screen-door-effect-in-vr/>
- [26] DAVIES, Alex. Oculus Rift Vs. HTC Vive Vs. PlayStation VR, HMD Specifications, Displays, Lenses And FoV. [online]. *Tom'sHARDWARE*, 2016 [cit.2019-03-07]. Dostupné z: <https://www.tomshardware.co.uk/vive-rift-playstation-vr-comparison,review-33556-3.html>
- [27] OCULUS. Oculus Rift. [online]. *Facebook Technologies*, 2016 [cit.2019-03-05]. Dostupné z: <https://www.oculus.com/rift/#oui-csl-rift-games=mages-tale>
- [28] VIVE. VIVE VR SYSTEM. [online]. *HTC Corporation*, 2019 [cit.2019-03-05]. Dostupné z: <https://www.vive.com/us/product/vive-virtual-reality-system/>
- [29] OCULUS, VR. Introducing Oculus Quest. [online]. *Facebook Technologies*, 2018 [cit.2019-03-05]. Dostupné z: <https://www.oculus.com/blog/introducing-oculus-quest-our-first-6dof-all-in-one-vr-system-launching-spring-2019/>
- [30] TPCAST. TPCAST Wireless Adapter for Oculus Rift. [online]. *TPcast U.S.*, 2018 [cit.2019-03-05]. Dostupné z: <https://www.tpcastvr.com/product-rift>

- [32] MULLEN, Tony. Prototyping Augmented Reality. Indianapolis, Indiana: John Wiley & Sons, 2011, 272 s. ISBN: 978-1-118-03663-1.
- [32] ŠUBÁK, Michael. AKO SA DOSTAŤ DO VIRTUÁLNEJ REALITY. [online]. *VIRTUÁLNA REALITA.EU*, 2016 [cit.2019-04-16]. Dostupné z: <https://virtualnarealita.eu/vseobecne-virtualna-realita/ako-sa-dostat-do-vr/#pcvirtulnarealita>
- [33] BORROWED, LIGHT. Room Scale vs. Seated VR. [online]. *Borrowed Light Studio*, 2016 [cit.2019-04-16]. Dostupné z: <http://www.borrowedlightvr.com/2016/02/29/room-scale-vs-seated-vr/>
- [34] G2 CROWD. Best VR SDK Software. [online]. *G2 Crowd, Inc.*, 2017 [cit.2019-04-16]. Dostupné z: <https://www.g2.com/categories/vr-sdk>
- [35] SUDHAKARAN, Sareesh. Notes by Dr. Optoglass: The Resolution of the Human Eye. [online]. *Wolfcrow*, 2012 [cit.2019-04-16]. Dostupné z: <https://wolfcrow.com/notes-by-dr-optoglass-the-resolution-of-the-human-eye/>
- [36] RUBIO-TAMAYO, Jose, Manuel GERTRUDIX BARRIO a Francisco GARCÍA GARCÍA. Immersive Environments and Virtual Reality: Systematic Review and Advances in Communication, Interaction and Simulation. *Multimodal Technologies and Interaction* [online]. 2017, 1(4) [cit. 2019-04-24]. DOI: 10.3390/mti1040021. ISSN 2414-4088. Dostupné z: <http://www.mdpi.com/2414-4088/1/4/21>
- [37] SFÉRA. Virtuálna realita. [online]. *Sféra, Grafické Informačné systémy*, 2018 [cit.2019-04-17]. Dostupné z: <https://www.sfera.sk/sk/sluzby/virtualna-realita/>
- [38] FAP. SWOT analýza. [online]. *FAP.SK*, 2019 [cit.2019-04-18]. Dostupné z: <http://www.fap.sk/Analyza-ostatne/SWOT-ANALYZA-FIRMY.htm>
- [39] OSMA. Profil firmy. [online]. *Agentúra Osma a.s.*, 2006 [cit.2019-02-20]. Dostupné z: <http://www.osma.cz/profil.htm>
- [40] EDUCBA. 3DS Max vs Blender. [online]. *EDUCBA.COM*, 2019 [cit.2019-04-20]. Dostupné z: <https://www.educba.com/3ds-max-vs-blender/>
- [41] ALLEGORITHMIC. Substance Painter. [online]. *ADOBE*, 2019 [cit.2019-04-20]. Dostupné z: <https://support.allegorithmic.com/documentation/spdoc/substance-painter-20316164.html>

- [42] PLURALSIGHT. 3ds Max, Maya LT or Blender – Which 3D Software Should I Choose for Asset Creation?. [online]. *Pluralsight*, 2015 [cit.2019-04-25]. Dostupné z: <https://www.pluralsight.com/blog/film-games/3ds-max-maya-lt-blender-3d-software-choose-asset-creation>
- [43] THINKWIK. CryEngine vs Unreal vs Unity: Select the Best Game Engine. [online]. *MEDIUM.com*, 2018 [cit.2019-04-25]. Dostupné z: <https://medium.com/@thinkwik/cryengine-vs-unreal-vs-unity-select-the-best-game-engine-eaca64c60e3e>
- [44] TURBOSQUID. Tris, Quads & N-Gons. [online]. *TurboSquid 3D Resources*, 2017 [cit.2019-05-01]. Dostupné z: <https://www.turbosquid.com/3d-modeling/training/modeling/tris-quads-n-gons/>
- [45] PLURALSIGHT. Understanding the Difference between Texture Maps. [online]. *Pluralsight*, 2014 [cit.2019-05-03]. Dostupné z: <https://www-stage.pluralsight.com/blog/film-games/understanding-difference-texture-maps>
- [46] WOW-HOW. 3D TEXTURING – ONE MORE STEP TO REALISM. [online]. *Wow-How Studio*, 2018 [cit.2019-05-03]. Dostupné z: <https://wow-how.com/3d-texturing>
- [47] CC0 TEXTURES. 521 public domain PBR textures. [online]. *Struffel Production (Lennart Demes)*, 2017 [cit.2019-05-03]. Dostupné z: <https://www.cc0textures.com/>
- [48] TEXTURES.COM. Textures for 3D, graphic design and Photoshop!. [online]. *Textures.com*, 2005 [cit.2019-05-03]. Dostupné z: <https://www.textures.com/>

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATEK

2D	Two-Dimensional space – Dvojmerný priestor, objekt.
3D	Three-Dimensional space - Trojrozmerný priestor, objekt.
6DOF	Six-Degrees-of-freedom – Voľnosť pohybu v 3D priestore.
API	Application Programming Interface – rozhranie pre programovanie aplikácií.
AR	Augmented Reality – rozšírená realita.
BOOM	Binocular Omni Orientation Monitor – binokulárny monitor.
CMYK	Cyan Magenta Yellow Black – farebný kanál pre tlač.
CPU	Central Processing Unit – centrálna výpočtová jednotka, procesor.
CV	Custom Version – zákaznícka verzia (Oculus Rift).
DPI	Dots Per Inch – počet bodov na jednotku plochy.
FPS	Frame Per Second – rámeček za sekundu.
FOV	Field of View – uhol pohľadu (udávaný v stupňoch).
GB	GigaByte – jednotka veľkosti pamäte.
GPU	Graphic Processing Unit – grafický procesor.
HMD	Head Mounted Display – headup display, zariadenie pre zobrazovanie VR.
Hz	Herz – jednotka frekvencie.
IDE	Integrated Development Environment – integrované vývojové prostredie.
IMU	Inertial Measuring Unit - vysokorýchlostné inerciálne meracie jednotky.
LED	Light Emitting Diode – elektroluminiscenčná dióda.
OS	Operation System – operačný systém PC.
PC	Personal Computer – osobný počítač.
PC	Phase Coherent – metóda merania času koherentná fáza.
PR	Public Relations – práca s verejnosťou.
PPI	Pixel Per Inch – počet pixelov na jednotku plochy, udávanú v palcoch.

RAM	Random Access Memory – paměť s náhodným přístupem.
SDK	Software Development Kits – nativní sady na vývoj softvéru.
SWOT	Analitická metoda slabých, silných stránek, příležitostí a rizik.
TOF	Time of Flight – metoda měření času letu.
UI	User Interface – uživatelské rozhraní.
USB	Univesal Serial Bus – univerzální sériová zbernica.
VR	Virtual Reality – virtuální realita.
VP	Virtuálně prostředí.

ZOZNAM OBRÁZKOV

Obrázok č.1 – Koncept kontinua zmiešanej reality.

Obrázok č.2 – Sensorama simulátor.

Obrázok č.3 – Philco Headsight HMD.

Obrázok č.4 – Ivan Shuterland a HMD Damoklov meč.

Obrázok č.5 – Obraz zo *Super kokpitu*.

Obrázok č.6 – Projekt *Videoplace*.

Obrázok č.7 – Klasická Fresnelová šošovka pri pohľade z prednej strany použitá v zariadení HMD HTC Vive.

Obrázok č.8 – Pohľad z prednej strany na hybridnú Fresnelovú šošovku použitú v zariadení HMD Oculus Rift.

Obrázok č.9 – Barelové skreslenie.

Obrázok č.10 – Zobrazenie 6DOF.

Obrázok č.11 – Screenshot základného užívateľského rozhrania programu Blender.

Obrázok č. 12 – Screenshot užívateľského rozhrania game enginu Unity3D.

Obrázok č.13 - Metóda modelovania box modeling.

Obrázok č.14 – Pridanie nového objektu kocka.

Obrázok č.15 – Prechod do edit módu.

Obrázok č.16 – Vymodelovaná úvodná časť nohy stojana 3D modelu.

Obrázok č.17 – Vymodelovaná spodná základňa stojana na tlačové valce.

Obrázok č.18 – Vymodelovaná konštrukcia stojana pre tlačové valce so zobrazením normál.

Obrázok č.19 – 3D model stojana pre tlačové valce.

Obrázok č.20 – Oddelenie funkčných častí modelu zariadenia do samostatných objektov.

Obrázok č.21 – Zobrazená vyselektovaná časť noha stojana s konštrukciou a UV editor.

Obrázok č.22 – Výber hrán a označenie Seamov.

Obrázok č. 23 – UV mapa vybranej časti stojana na tlačové valce.

Obrázok č. 24 – Vytvorená UV mapa konštrukcie stojana na tlačové valce.

Obrázok č. 25 – Hotová UV mapa 3D modelu stojana pre tlačové valce.

Obrázok č.26 – Importovaný 3D model stojana s vytvorenou UV mapou a UI aplikácie.

Obrázok č.27 – Vytvorené vrstvy jednotlivých súčastí s ich farebným rozlíšením.

Obrázok č.28 – Nastavenie Base Color, pridanie textúry do Roughness slotu.

Obrázok č.29 – Nastavenie hodnoty UV scale vybranej vrstvy v izolovanom režime.

Obrázok č.30 – Pridanie úrovne (Levels) pre Roughness, zmena nastavenia hodnôt.

Obrázok č.31 – Bump (Height) mapa v samostatnej vrstve s vloženou textúrou a nastavením.

Obrázok č. 32 – Hotové textúry na 3D model stojana pre tlačové valce.

Obrázok č.33 – Vytvorenie VR 3D projektu „VRprojectDP“ s pridaným Oculus Integration balíkom.

Obrázok č.34 – Nastavenie herného enginu Unity pre podporu vývoja VR Oculus.

Obrázok č.35 – Nastavenie umiestnenia, rotácie a rozmerov implementovaných objektov.

Obrázok č.36 – Implementované 3D modely s nasadenými textúrami.

Obrázok č.37 – Ponuka možností pridania koliderov, príklady pridania Mesh a Box koliderov.

Obrázok č.38 – Spúšťačí (Trigger) Box Collider objektu tlačiarenskeho stroja.

Obrázok č.39 – Tlačové valce s pridanými Box kolidermi, Rigidbody a OVR Grabbable.

Obrázok č.40 – Vytvorené a umiestnené farebníkové zásobníky.

Obrázok č.41 – ParticleSystem zariadenia na lepenie tlačových polymérov.

Obrázok č.42 – Animation editor, Animator editor, vytvorenie animácie pre objekt *button-Zamkni*.

Obrázok č.43 – Pridanie zvukových objektov, nastavenie.

Obrázok č.44 – Pridanie objektu OVRPalyerController, skriptu Character Camera Constraint a nastavenie objektu a jeho súčastí.

Obrázok č. 45 – Definovanie parametra Parent Transform, nastavenie a prispôsobenie objektov.

Obrázok č.46 – Nastavenie kontroly fyziky na hodnotu 90FPS.

Obrázok č.47 – Nastavenie Canvas a OVR Raycaster (Script).

Obrázok č.48 – Zdrojové obrázky pre hlavičku a tlačidlá UI.

Obrázok č. 49 – Vytvorené VR UI hlavnej ponuky (objektová a vizuálna stránka).

Obrázok č.50 – UI s pomocnými obsahmi pre zariadenie na lepenie tlačových polymérov.

Obrázok č.51 – Deklarácia požadovaných public premenných v skripte a vloženie objektov do jednotlivých slotov atribútov (premenných).

Obrázok č.52 - Volanie funkcií *SetActive* pre všetky objekty Canvas v hlavnej ponuke vo funkcií *Start()* a definícia funkcie na prechod medzi ponukami.

Obrázok č.53 – Nastavenie vytvorenej funkcie objektu tlačidla O aplikácii a pridanie zvukov.

Obrázok č.54 – Vytvorenie a nastavenie špeciálnej premennej PlayerPrefs s názvom PohybMozny.

Obrázok č.55 – Zobrazovanie hlavnej ponuky.

Obrázok č.56 – kódy pre nastavenia zvukov použitých v aplikácii.

Obrázok č.57 – Použitie funkcií kolideru na zobrazovanie a skrývanie UI (Canvas).

Obrázok č.58 – Naprogramovanie transformácií pozície a rotácie objektu stredového panelu.

ZOZNAM TABULIEK

Tabuľka č.1 – Rozdielne vlastnosti medzi technológiami virtuálnej a rozšírenej reality.

Tabuľka č.2 – SWOT analýza.

Tabuľka č.3 – Porovnanie VR zariadení HTC Vive a Oculus Rift vrátane príslušenstva.

ZOZNAM PRÍLOH

Príloha číslo 1

DVD s elektronickou verziou diplomovej práce a s prílohami 2 – 6 zozipovanými v jednom spoločnom súbore s názvom „Prílohy Praktická Časť“.

Príloha číslo 2

Kompletný vývojársky projekt VRprojectDP pre herný engine Unity 3D – obsahuje všetky zdrojové kódy, modely, textúry, zvuky, obrázky a všetky komponenty a súčasti vrátane implementácií, ktoré boli použité pri tvorbe našej VR aplikácie.

Príloha číslo 3

Zostavená (vybudovaná) konečná VR aplikácia pripravená na nasadenie do prevádzky.

Príloha číslo 4

Fotodokumentácia zariadení výrobnjej spoločnosti.

Príloha číslo 5

Videodokumentácia zariadení výrobnjej spoločnosti.

Príloha číslo 6

Video ukážka zo simulácie výrobného procesu a jeho postupu z vytvorenej VR aplikácie.

