

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ

Studijní program: B 2301 Strojní inženýrství
Studijní zaměření: Průmyslové inženýrství a management

BAKALÁŘSKÁ PRÁCE

Tvorba interaktivních modulárních komponent pro virtuální prostředí v
rámci konceptu Digitální továrny

Autor: **Lukáš HANZÍK**

Vedoucí práce: **Ing. Petr HOŘEJŠÍ, Ph.D.**

Akademický rok 2016/2017

Prohlášení o autorství

Předkládám tímto k posouzení a obhajobě bakalářskou/diplomovou práci, zpracovanou na závěr studia na Fakultě strojní Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů, uvedených v seznamu, který je součástí této bakalářské/diplomové práce.

V Plzni dne:
1. 4. 2017

.....
podpis autora

ANOTAČNÍ LIST BAKALÁŘSKÉ PRÁCE

AUTOR	Příjmení HANZÍK	Jméno Lukáš		
STUDIJNÍ OBOR	2301T007 Průmyslové inženýrství a management			
VEDOUCÍ PRÁCE	Příjmení (včetně titulů) Ing. HOŘEJŠÍ, Ph.D.	Jméno Petr		
PRACOVIŠTĚ	ZČU - FST - KPV			
DRUH PRÁCE	DIPLOMOVÁ	BAKALÁŘSKÁ		Nehodící se škrtněte
NÁZEV PRÁCE	Tvorba interaktivních modulárních komponent pro virtuální prostředí v rámci konceptu Digitální továrny			

FAKULTA	strojní	KATEDRA	KPV	ROK ODEVZD.	2017
----------------	---------	----------------	-----	--------------------	------

POČET STRAN (A4 a ekvivalentů A4)

CELKEM	42s.	TEXTOVÁ ČÁST	---	GRAFICKÁ ČÁST	---
---------------	------	---------------------	-----	----------------------	-----

<p style="text-align: center;">STRUČNÝ POPIS (MAX 10 ŘÁDEK)</p> <p>ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY</p>	<p>V této bakalářské práci je ukázána tvorba digitálních modelů několika strojů a inventárního vybavení, následně vytvořeny programové kódy, které popisují pohyby, funkce a interakce, které tyto modely budou schopné vykonávat. Modely včetně přiřazených skriptů jsou uloženy do knihovny souborů takovým způsobem, aby bylo umožněno jejich využití při tvorbě digitálních továren a jiných projektů ve vývojovém prostředí Unity 3D. Na závěr je zde vytvořeno ukázkové virtuální prostředí, zobrazující možné budoucí využití výše zmíněné digitální knihovny. Jsou zde také ve zkratce popsány využití SW nástroje Unity 3D a Blender 3D.</p>
<p style="text-align: center;">KLÍČOVÁ SLOVA</p> <p style="text-align: center;">ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE</p>	<p style="text-align: center;">Digitální továrna, Tvorba digitálních modelů, Blender 3D, Unity 3D, Interaktivní prostředí</p>

SUMMARY OF BACHELOR SHEET

AUTHOR	Surname HANZÍK	Name Lukáš	
FIELD OF STUDY	2301T007 Industrial Engineering and Management		
SUPERVISOR	Surname (Inclusive of Degrees) Ing. HOŘEJŠÍ, Ph.D.	Name Petr	
INSTITUTION	ZČU - FST - KPV		
TYPE OF WORK	DIPLÓMA	BACHELOR	Delete when not applicable
TITLE OF THE WORK	Development of Interactive Modular Components for Virtual Environment in Terms of Digital Factory Concept		

FACULTY	Mechanical Engineering	DEPARTMENT	Industrial Engineering and Management	SUBMITTED IN	2016
----------------	------------------------	-------------------	---------------------------------------	---------------------	------

NUMBER OF PAGES (A4 and eq. A4)

TOTALLY	42p.	TEXT PART	---	GRAPHICAL PART	---
----------------	------	------------------	-----	-----------------------	-----

BRIEF DESCRIPTION TOPIC, GOAL, RESULTS AND CONTRIBUTIONS	<p>This bachelor thesis describes the development of digital models of several machines and inventory equipment and making of source codes which describe functions and interactions of these models. Models and scripts are saved as a file library in such manner to allow them to be used when creating digital factories' interfaces and other projects in Unity 3D. Finally there has been made a virtual interface which shows possible usage of previously mentioned digital library. There are also shortly described programs Unity 3D and Blender.</p>
KEY WORDS	Digital Factory, Development of Digital Models, Blender 3D, Unity 3D, Interactive Environment

Obsah

Přehled použitých zkratk a symbolů.....	2
Seznam obrázků	2
1 Úvod	3
1.1 Základní pojmy.....	3
2 Analýza současného stavu	6
2.1 DIGITOV	6
2.2 Siemens DF&PD	7
2.3 Zobrazovací metody a technologie.....	8
2.3.1 Metody.....	8
2.3.2 Zobrazovací zařízení	9
3 Analýza možností SW nástrojů Unity a Blender	11
3.1 Unity 3D.....	11
3.1.1 Layouts	11
3.1.2 Views.....	12
3.1.3 TransformTools	12
3.1.4 Cameras	12
3.1.5 Assets.....	12
3.1.6 Skriptování	13
3.1.7 Možnosti využití Unity.....	14
3.2 Blender	19
3.2.1 Módy	19
3.2.2 Řídící bod	21
3.2.3 Modelování.....	21
3.2.4 UV mapping	23
3.2.5 Možnosti využití Blenderu	24
4 Tvorba a popis modelů (komponent).....	26
4.1 Modely.....	27
4.1.1 Model dveří	27
4.1.2 Model frézky	29
4.1.3 Dopravníkový pás.....	32
4.1.4 Ovládací panel	33
4.1.5 Ostatní modely	34
5 Tvorba a popis interakcí	35
5.1 Algoritmus rotace nástrojů	35
5.2 Algoritmus světelného spínače.....	36
5.3 Algoritmus viditelného kurzoru myši (visibleCursor.cs.cs).....	36
5.4 Algoritmus restartování levelu (restart.cs)	36
5.5 Algoritmus ukončení programu (escapeQuit.cs).....	36
5.6 Algoritmus povolení/zakázání komponentu (MD_dis.cs).....	37
5.7 Ukázkové prostředí a ovladač FPS.....	37
5.8 Algoritmus lineární trajektorie	37
5.9 Animace.....	38
5.9.1 Spuštění animace přiblížením nebo vzdálením FPS ovladače (fireAnim_enterExit.cs).....	38
5.9.2 Spuštění animace stisknutím myši (fireAnim1.cs)	38
6 Závěr.....	39
Zdroje	40

Přehled použitých zkratk a symbolů

2D	dvourozměrný
3D	trojrozměrný
AR	Rozšířená realita (anglicky Augmented Reality = AR)
CAD	computer aided design
CAM	computer aided manufacturing
CAVE	Computer Aided Virtual Environment
DP	Digitální podnik
HDMI	High-Definition Multimedia Interface
HMD	Head Mounted Display
KPV	Katedra průmyslového inženýrství a managementu
MH	Microsoft Hololens
NASA	Národní úřad pro letectví a kosmonautiku
PC	Osobní počítač
SW	Software
VR	Virtuální realita

Seznam obrázků

Obrázek 1– DIGITOV [22]	6
Obrázek 2 - prostředí sw SIRIUS [45]	7
Obrázek 3 - anaglyf[24]	9
Obrázek 4 - Tanks![25]	14
Obrázek 5 - VR smaples[26]	15
Obrázek 6 - Virtual Car[27]	16
Obrázek 7-Ferrari AR showroom[29]	16
Obrázek 8-Oshkosh HEMTT VirtualTraining[31]	17
Obrázek 9-Gunstriction[32]	17
Obrázek 10-Bike Config[40]	18
Obrázek 11 - Příklad vzniku meta objektu[33]	22
Obrázek 12-Cloth Modifier[20]	23
Obrázek 13-Liquid Modifier[21]	23
Obrázek 14-texturování pomocí UV mapování [34]	23
Obrázek 15-Elephants Dream[35]	24
Obrázek 16-Big Buck Bunny[36]	24
Obrázek 17-BMW Promo Blender[38]	25
Obrázek 18-možnosti Blenderu[39]	25
Obrázek 19 - dveře	28
Obrázek 20-frézka	31
Obrázek 21-Pásový dopravník	33
Obrázek 22-Ovládací panel	34

1 Úvod

Na Katedře průmyslového inženýrství a managementu probíhalo nebo dosud probíhá několik projektů zaměřených na vytvoření levnější varianty (k zatím dostupným programům) softwarového prostředí, v němž by bylo možné různými způsoby simulovat chod digitálních podniků a továren. V této práci je popsáno několik z mnoha dílčích kroků nezbytných při vývoji takovýchto virtuálních prostředí.

Je zde ukázána tvorba digitálních modelů několika strojů a inventárního vybavení, následně vytvořeny programové kódy, které popisují pohyby, funkce a interakce, které tyto modely budou schopné vykonávat. Modely včetně přiřazených skriptů budou uloženy do knihovny souborů takovým způsobem, aby bylo umožněno jejich využití při tvorbě digitálních továren. Na závěr je zde vytvořeno ukázkové virtuální prostředí, zobrazující možné budoucí využití výše zmíněné digitální knihovny. Jsou zde také ve zkratce popsány využití SW nástroje Unity 3D a Blender 3D. V této úvodní kapitole je ještě uvedeno několik základních pojmů týkajících se rozebírané problematiky.

1.1 Základní pojmy

Oblasti světového obchodu i globální výroby se neustále vyvíjejí a vytvářejí nové trendy, které je nutné pro optimální fungování podniků dodržovat. Rychlý rozvoj automatizace, výpočetních a komunikačních technologií klade na konkurenceschopnost firem stále silnější nároky. Čím dál důležitějším faktorem se stává rychlost, kterou se nové produkty (myšlenky, výrobky, inovace...) dostávají na trh. Snížení časové potřeby pro předvýrobní etapy může přispět ke značnému zefektivnění například v oblasti nákladů nebo k delšímu setrvání produktů na trhu. V současné době se zdá téměř nemožné vyhovět všem výše zmíněným požadavkům bez využití alespoň nějakého stupně digitalizace [4], [5].

„Digitalizace je převod vybraných měřitelných fyzikálních veličin digitalizovaného objektu do numerických hodnot, jejich kódování a uložení za účelem pozdějšího vygenerování jiných fyzikálních veličin...“ [3].

Digitalizace a počítačová podpora umožňuje zkvalitnění a zrychlení vývojových a předvýrobních etap, stejně tak i fází týkajících se výroby. Její nástup znatelně změnil metody přístupu k inženýrské práci a to ve všech částech životního cyklu výrobku. Dnes je jen těžko představitelný jakýkoli postup v oblasti konstrukce (a následně výroby) bez technologií CAD/CAM, se kterými je možné produkt vytvářet od prvního návrhu, přes simulování případných zatížení během provozu, naprogramování chodu výrobních strojů až po vizuální prezentaci. V řízení podniku se využívají systémy digitálního plánování.

Plné integrování digitálního zpracování a sdílení toku informací do všech nebo některých oblastí výrobního procesu shrnujeme pod pojmem **digitální továrna/digitální podnik** (DP) [4], [5].

*„Pod pojmem **Digitální podnik** rozumíme zpravidla, tak jak bylo již ukázáno, především postupný projektový proces, který je podporován velmi různorodými, ale navzájem kompatibilními softwarovými nástroji. Cílem je digitálně zachytit a utvářet pomocí podpůrných softwarových nástrojů část životního cyklu výrobku, přičemž postupujeme od počátečních koncepčních stádií až k simulaci precizně navrženého dílenského pracoviště, hnízda nebo střediska. Pojem Digitální podnik lze tedy dobře chápat především ve filozofické*

rovině určitého přístupu, kdy všechny informace přes dlouhý řetězec činností tvorby a života výrobku jsou zpracovávány, přednášeny a většinou i realizovány v digitální podobě “ [4].

DP je tedy jakýsi nadřazený pojem pro kombinaci metod, modelů a nástrojů, které jsou použity v reálných firmách a mají za cíl zajistit efektivní využití všech jejich zdrojů. V této práci bude definice pojmu digitální podnik omezena pouze na část týkající se zejména simulace výrobních procesů.

Používání principů DP má jistě svá úskalí i pozitiva. „*Zavedení principů a systémů digitálního podniku se přímo projevuje na ekonomických a výrobních ukazatelích firmy, protože každá drobná úspora realizovaná v etapě plánování se po zahájení sériové výroby mnohokrát znásobí. Díky tomu je návratnost investic do systému digitální továrny především u velkých podniků poměrně krátká...*“ [5].

V první řadě je možné zmínit snížení potřeby času a to hned v několika oblastech. Ve fázi vývojové, odpadá značná část časových nákladů, zejména na tvorbu různých prototypů a testování jejich funkce a provozu. Digitalizace také značně zrychluje komunikaci mezi jednotlivými subjekty vývojového řetězce, usnadňuje sdílení potřebných dat a velice omezuje pracnost rutinních nebo jinak se opakujících operací. Těmito kroky lze velice urychlit rozběhnutí výroby a dosažení naplánovaných optimálních objemů výroby. Celkově se tedy jedná o značné zkrácení předvýrobních stádií a zvýšení produktivity. Z hlediska finančních nákladů jde také dosáhnout určitého zefektivnění, většina z nich vyplývá z výše uvedených úspor času. U výroby prototypů také významně klesá jejich počet (virtuální modely lze snadno duplikovat a následně měnit některé parametry, což značně zrychlí například porovnávání jednotlivých funkcí, rozměrů a tvarů prototypů). Digitalizace usnadňuje organizaci potřebné dokumentace, při správném zacházení téměř vylučuje opakování potřebných operací, případnou duplikaci výrobních postupů nebo vznik více stejných verzí dokumentů a tím velmi usnadňuje například i správné naplánování, což také vede ke snižování investičních nákladů, zmetkovitosti nebo odchylek od požadovaných množství výstupů. Celkové využití zdrojů je díky správnému naplánování efektivnější. Simulovat lze i různé postupy výroby (i bez nakupování příslušných strojů nebo lidského kapitálu). Náklady klesají i při uchovávání dat.

Z výše zmíněných přínosů jde také odvodit značné zvýšení míry flexibility podniku. Již naznačená variabilita při návrhu konstrukcí (prototypy, zvyšování modularity a pokles náročnosti na smontovatelnost) nebo při volbě materiálových nebo datových toků [4], [5].

Zřejmým nedostatkem DP by mohlo být poměrně rychlé navýšení vstupních nákladů zejména na využívaný SW a odbornou práci potřebnou pro jeho zavedení a údržbu [4], [5]. Ohled je nutné brát i na fakt, že například při digitálním testování dochází pouze k simulaci pomocí určitých matematických modelů, které se realitě mohou velmi přibližovat, ale v mezních případech ji nemusí popisovat zcela dokonale a to může vést k nežádoucím výsledkům při převedení do skutečné výroby.

Prostředí související s využíváním digitalizace a DP shrnujeme pod pojmem **virtuální realita** (VR). Virtuální jevy dnes jednoznačně patří ke všednímu životu. Mohou mít původ v reálném světě (tvary, barvy, vzájemné vazby), ovšem se od něj mohou i diametrálně lišit (např. absence základních fyzikálních zákonů a charakteristik). Základní vlastností virtuálních jevů je zcela nemateriální povaha a značný stupeň abstrakce, které dovolují velmi široké

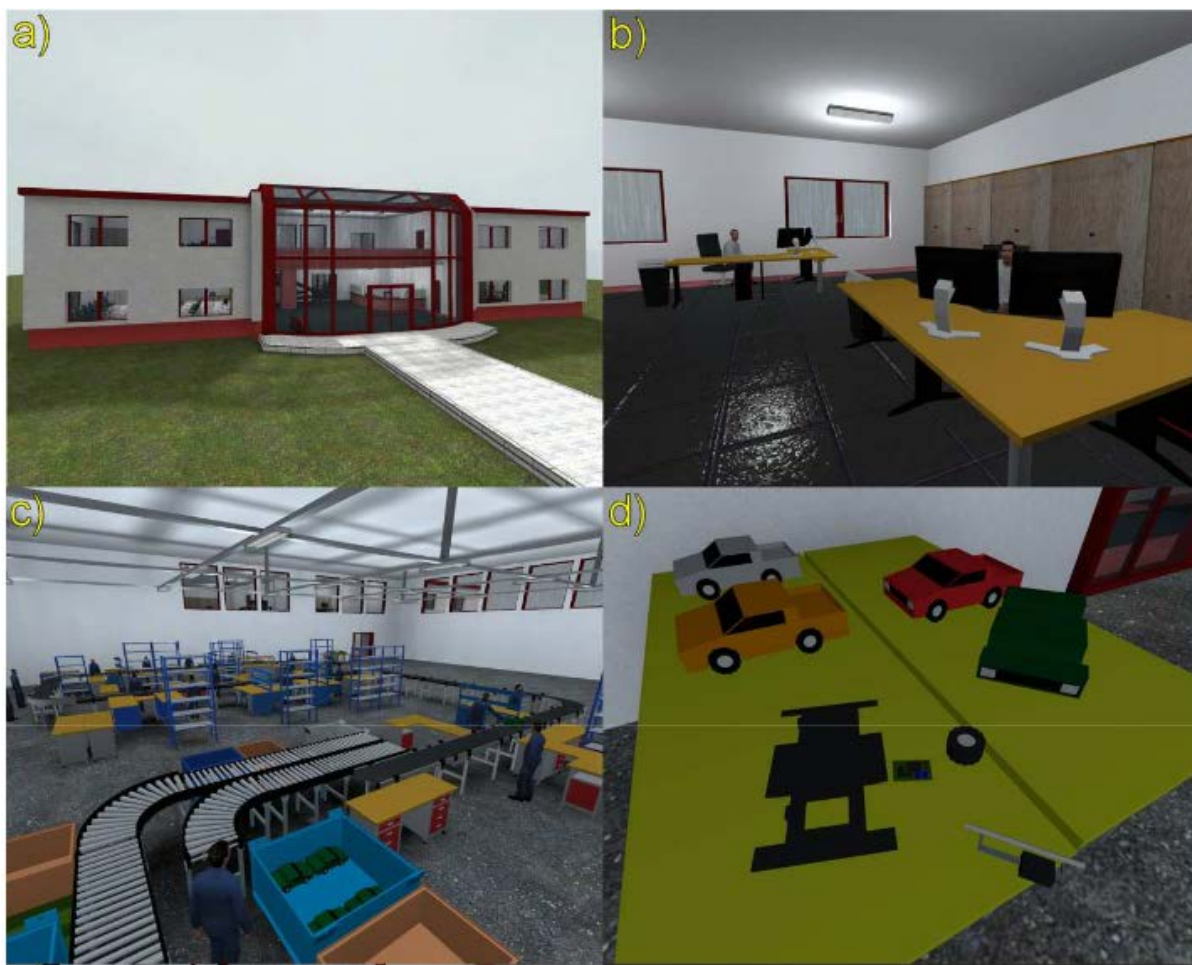
využití (lékařství, letectví, strojírenství, zábavní průmysl...). Práci s nimi také zjednodušuje jejich možnost je archivovat a reprodukovat, nebo možnost rozšiřovat nebo naopak redukovat jejich obsah a parametry. Umožňují například simultánní přístup více uživatelů z různých míst, což může mít za následek značné snížení časových nároků na provádění určitých činností, nebo na jejich organizaci [4], [5]. Virtuální realitou můžeme rozumět technologie, které za pomoci počítačů vytvářejí simulace určitého prostředí a využitím různých nástrojů zprostředkovávají komunikaci mezi tímto prostředím a uživatelem (tzn., zajišťují kontakt uživatele a virtuálních jevů). Lze také říci, že se jedná o metody zobrazení virtuálního prostoru, při snaze zajistit požadovaný stupeň imerze (z lat. *immersio*), tzn. maximálního pocitu „ponoření“ neboli navození pocitu reálnosti dané simulované situace. Imerze závisí zejména na kvalitě použitých technologií a lze navodit stimulací různých smyslů (nejčastěji zrak, sluch a hmat) [4], [5].

„Těsný vztah mezi vizuálními, kinestetickými a zvukovými modalitami je klíčem k pocitu imerze, který vytváří mnoho počítačových her, simulací a systémů virtuální reality“ [2].

Její míru je možné posuzovat podle dosažené vizualizace (např. snaha o co nejvěrnější napodobení reality) či interaktivity (VR reaguje na vnější podněty uživatele; uživatel je schopen vnímat např. zpětnou reakci prostředí) a dále ji zvyšovat za pomoci např. 3D stereoskopie, haptických pomůcek apod. Jednoduše lze VR chápat, jako jakékoli prostředí (např. uživatelské nebo i jiné) vytvářené a zobrazované výpočetní technikou. Z úrovně jaké dosahují dnešní technologie je zřejmé, že zatím není možné vytvořit dokonalý virtuální obraz reality, nicméně pro oblasti průmyslu výrobního nebo například také zábavního je výsledná kvalita dostačující.

Pokud zobrazíme reálný svět ve virtuálním prostředí, je možné ho následně rozšířit o počítačově vytvořené prvky nebo objekty. Takovéto zobrazení je nazýváno **rozšířená realita** (Augmented Reality - AR). Při prvním pohledu na takto definovaný pojem se může zdát, že se jedná o něco složitějšího, nicméně při bližším pohledu je jasné, že alespoň s nějakým druhem AR je možné se setkat v každodenním životě velmi snadno (např. televizní přenosy doplněné informačními tabulkami, fotografie s popisem data a místa pořízení, titulky ve filmech, ale i trojrozměrné modely umístěné přímo do scén nebo technologie polopropustných brýlí...). AR je pojem ve vztahu k VR jaksi podřízeným (lze považovat za jednu z technologií), nicméně určitý rozdíl mezi virtuální a rozšířenou realitou je možné pochopit jako míru dosažitelné imerze. VR si klade za cíl uživatele co nejdokonaleji „pohltnout“, naproti tomu AR reálný svět „nezakrývá“, ale pouze doplňuje. Tato práce si neklade za cíl vytvořit prostředí AR, nicméně tento pojem je zde dobré zmínit pro upřesnění výše uvedených pojmů.

Z výše zmíněných informací vychází potřeba tvorby interaktivních modelů podniků, a to z důvodů simulace provozu, prezentace produktů, nebo vzdělávání personálu. Pro jejich tvorbu a provoz existuje určité množství různě vyvinutých SW, které často nemusí splňovat kritéria úspor času (jsou příliš komplikované) nebo nejsou zcela výhodné po stránce finanční [22]. Tyto důvody vedou k myšlence vytvoření substitučního modelovacího prostředí, jehož výstupy by se vyrovnaly běžně komerčně využívaným programům a zároveň jeho ovládání bylo maximálně intuitivní a náklady na jeho tvorbu a provoz nedosahovaly příliš vysokých částek. Tato práce se věnuje zejména takto zvolenému řešení a částečně se snaží navázat na myšlenky projektu DIGITOV (viz Obrázek 1), který je popsán v následující kapitole.



Obrázek 1– DIGITOV [22]

2 Analýza současného stavu

Při současném stavu technologií již není nutné VR a další nástroje považovat pouze za technologii zítřka. Využívána bývá v různých odvětvích a to od simulování lékařských zákroků, přes analýzu pohybových možností sportovců nebo pracovníků v různých oborech, výcvikové aplikace (ve vojenském i civilním letectví, návody pro montáž a servis nebo orientaci v prostoru) až po oblasti průmyslové výroby, jako jsou například tvorba technických výkresů a trojrozměrných modelů při konstrukci apod. Značného uplatnění našla také v automobilovém průmyslu.

V rámci této kapitoly je také nutné zmínit, že obor virtuální reality je značně dynamický (vývoj a změny probíhají často v řádu pouhých několika měsíců, nebo dokonce týdnů), a proto lze očekávat, že níže popsané ukázky budou v čase zastarávat. Uvedeme zde tedy pouze určitý nástin možných variant. Pro zjištění skutečně aktuálního stavu je nutné sledovat konkrétní technologie online.

2.1 DIGITOV

DIGITOV je softwarový balíček vytvořený pro výuku předmětu KPV/DPVR na katedře KPV na Fakultě strojní, Západočeské univerzity v Plzni. Tento nástroj je založen na Source Engine (a s ním dodávané sady Source SDK), vyprodukované firmou Valve Corporation, který byl použit v počítačové hře Half-Life 2. Původním záměrem bylo vytvořit levnější alternativu k moderním komerčním programům pro tvorbu prostorových vizualizací [7].

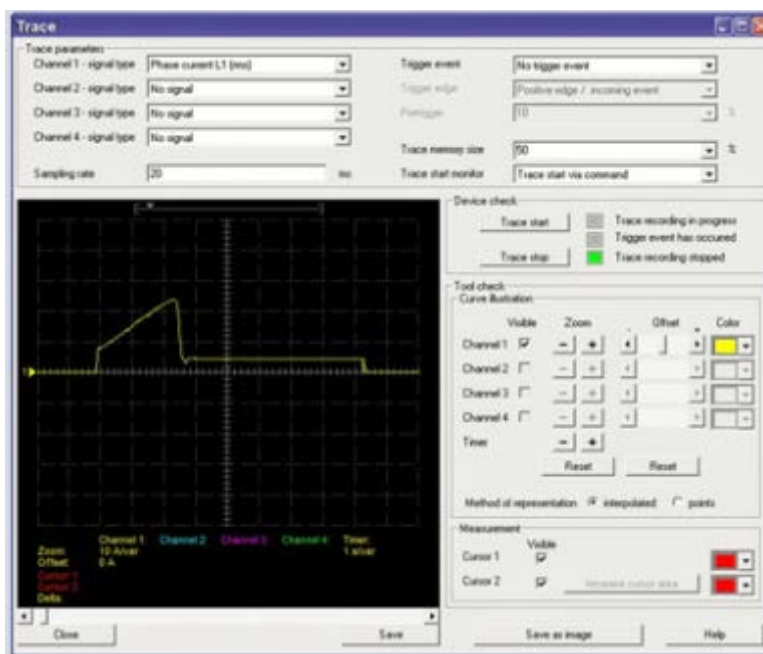
Balíček obsahuje data doplňující Source Engine, která by měla sloužit pro vytváření prostředí virtuálních podniků. Byly pro něj vytvořeny, nebo z volně dostupných zdrojů importovány textury, statické, dynamické i fyzikální modely (strojů, dopravníků, nábytkového vybavení apod.). Dále balíček obsahuje i data pro řídicí logiku některých virtuálních objektů a modelů. Modely vytvářené v DIGITOV mohou být také kompatibilní s technologií CAVE [7].

DIGITOV je na první pohled (např. Obrázek 1) vhodnou alternativou k běžnému komerčně využívanému SW virtuální reality. V rámci digitální továrny umožní simulovat celý proces výroby nebo jen jeho jednotlivé části. V průběhu jeho testování, zdokonalování a provozování se však ukázalo jako nepraktické a finančně neefektivní využití licence Source Engine pro jiné než výukové účely, a proto se od jeho dalšího vývoje začíná upouštět [7].

Lépe se z hlediska licenčního využití (ale i lepší multiplatformity, vzhledu, funkčnosti atd. viz následující kapitola o SW) jeví vývojové prostředí Unity, do kterého se v současnosti převádí stávající data z DIGITOVU.

2.2 Siemens DF&PD

Jedná se o divizi Digital Factory and Process Industries and Drives, která patří pod společnost Siemens. V produktovém portfoliu nalezneme SW i HW nástroje pro automatizaci, techniku pohonů, spínací techniku, rozvody energie, procesní instrumentace a komunikace nebo systémy pro ovládání a vizualizaci. Detailní popis jednotlivých produktů by vydal na několik bakalářských prací. Tématu této práce nejlépe odpovídají nástroje pro parametrizaci, konfiguraci a hlavně vizualizaci s názvem SIRIUS (viz Obrázek 2). Tento název v sobě zahrnuje SW i HW pro standardní panely pro ovládání strojů (např. WinCC, TIA), řešení pro využívání průmyslových PC a nejrůznější LCD monitory [43], [44].



Obrázek 2- prostředí sw SIRIUS [45]

Kromě společnosti Siemens využívají nástroje VR a DP například společnosti Škoda, Ford aj. Konkrétní případy byly popsány v jiných pracích (viz přiložený seznam literatury).

2.3 Zobrazovací metody a technologie

Tato práce se přímo nezabývá napojením virtuálního prostředí na níže popsané technologické prostředky; nicméně zde vytvářená knihovna bude umožňovat budoucí kompatibilitu s podobnými zobrazovacími zařízeními. Jelikož mají zobrazovací zařízení v oblasti virtuálních prostředí velkou budoucnost, jsou některé z nich v této podkapitole zmíněny.

Pocit obklopení virtuálním prostorem bude zřejmě nejsilněji navozovat stimul vnímaný zrakem (vizuální vjem snadno upoutá pozornost a ve většině případů tvoří první a velmi silný kontakt člověka s prostředím), proto je ve většině případů kladen značný důraz na vytvoření dostatečně kvalitního 3D prostoru a jeho projekci. Je zřejmé, že pokud je zobrazován trojrozměrný prostor do prostoru dvojrozměrného (většina současných technologií - monitory, projektory...), je nutné využít různé postupy pro dosažení požadovaného prostorového efektu na výstupu. Zobrazované objekty jsou často vykreslovány pomocí různých aproximací povrchu nebo transformací perspektivy, uplatňuje se využívání textur a simulace osvětlení scén a optických vlastností povrchů objektů. Pro zobrazování výpočetní technikou je také nutné uvážit viditelnost objektů z daného pohledu, kvůli snížení výpočtové náročnosti [4], [5].

Aproximace povrchu se provádějí nejčastěji dekompozicí na polygony nejlépe na trojúhelníky, které svou jednoduchostí umožňují značně vylepšit nároky na výpočet (při aproximaci polygony stejně dochází k rozdělení na trojúhelníky). Podle požadavků na kvalitu výstupu, nebo na jeho rychlost, se zvolí jakým počtem (tedy jakou jemností/přesností) bude povrch zobrazen. Datové uložení může pak probíhat například ukládáním souřadnic jednotlivých vertexů (bodů/vrcholů) do tabulek. Dále je nutné uložit i souřadnice ploch, neboli hran mezi vrcholy, které následně tvoří face (plochu). Reprezentování ploch také často probíhá pomocí normál. Aproximace mohou být samozřejmě různého druhu a záleží jen na programátorovi, zda zvolí vhodný typ pro konkrétní zobrazovaný objekt. V některých aplikacích je také nutné dodržet sled kroků, ve kterých bylo těleso vytvořeno (např. konstrukční systémy CAM). K tomu se může využít abstraktních datových struktur, nejčastěji stromů [4], [5].

Dalším důležitým aspektem při tvorbě 3D prostředí je simulace světla. Tento faktor bývá často považován za nejsložitější a má rozhodně nejzásadnější vliv na celkovou kvalitu zobrazení (bez světla není nic vidět). Dnes už jsou vytvořeny dostatečné algoritmy pro simulaci světla, ve většině dostupných programů je velmi snadno zvolitelné základní přednastavení, které značně urychlí přípravu výsledných vzhledů vytvářených projektů. Světlo a optické vlastnosti povrchů materiálů jsou reprezentovány pomocí vektorů, které určují směry dopadu a odrazu světla, normálami povrchu osvětlovaných objektů a dále například i směry, ze kterých se na objekty pohlíží a ve kterých jsou následně renderovány [4], [5]. Obecně jsou rozlišovány i různé druhy zdrojů světla, cílem této práce není je popisovat.

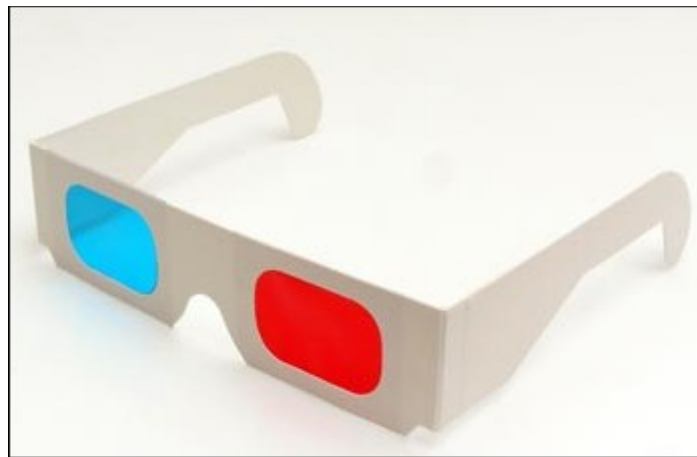
2.3.1 Metody

Před popisem vybraných zobrazovacích technologií jsou zde zmíněny některé metody pro dosažení trojrozměrnosti zobrazení.

Nejčastějšími technikami využívanými pro zobrazování ve třech dimenzích jsou různé druhy stereoskopické projekce. Zde se prostorový efekt získává promítáním dvou téměř se

překrývajících obrazů, které jsou nasnímány příslušnými zařízeními tak, aby byla napodobena vzdálenost lidských očí. Různými metodami (popsanými níže) se dá docílit toho, aby divák vnímal obrazy pro levé oko levým a pro pravé pravým okem, čímž je dosaženo relativního vnímání hloubky jako třetího rozměru obrazu [4], [5].

Anaglyf je pravděpodobně nejjednodušší a nejlevnější způsob zobrazování ve třech rozměrech. Sledovaný obraz je tvořen dvěma, vůči sobě posunutými, vrstvami určitých barev (nejčastěji modrá-červená). Skrz brýle (viz Obrázek 3) s příslušným filtrem prochází do každého oka pouze příslušné spektrum barev, tím je dosaženo 3D efektu. Za určitý nedostatek lze považovat ztrátu barevných informací. Výhodou je nepotřebnost speciální technologie pro projekci [4], [5].



Obrázek 3 - anaglyf[24]

U metody zvané **pasivní stereoskopie** se trojrozměrnosti dosahuje rozložením obrazu do horizontálních a vertikálních rovin (polarizace světla), využitím promítací plochy, která zachovává polarizaci a brýlí s příslušnými polarizačními filtry pro každé oko. Nutné je promítat ze dvou projektorů opatřených správnými polarizačními filtry. Při současném stavu technologií se nejedná o finančně nedostupnou metodu a v porovnání s anaglyfem jde o projekci bez omezení zobrazování barev. Naopak metoda **aktivní stereoskopie** využívá pouze jeden projektor, který promítá obraz s dvojnásobnou snímkovou frekvencí (postupně jeden obraz pro pravé následně pro levé oko). Brýle jsou elektronické a dálkově synchronizované se zdrojem vysílání tak, aby zatmíváním hledí propouštěly pouze obraz pro příslušné oko. Při dostatečné frekvenci promítání je lidský mozek schopen vnímat projekci jako jednotný trojrozměrný obraz. Plátno nemusí mít žádné výjimečné vlastnosti (v porovnání s pasivní stereoskopií).

2.3.2 Zobrazovací zařízení

Konečnou podobu zobrazení nejvíce ovlivňuje zvolená technika. Na výběr je velká řada zařízení, která svými vlastnostmi zajistí požadovanou kvalitu výstupu (samozřejmě s přihlédnutím k finančním možnostem uživatele). Kromě monitorů a projektorů je možné používat i speciálních přileb nebo brýlí, které mají integrované displeje. Další možností je využití zařízení typu CAVE (Computer Aided Virtual Environment).

Oculus rift

Rift je technologie virtuální reality. Jedná se o nepropustný HMD display (zkr. z angl. Head Mounted Display), ovládání je řešeno externími zařízeními bez snímačů gest nebo prostoru (klávesnice, myš, speciální ovladače), zobrazení pak může interagovat se snímačem

polohy brýlí. Informace o poloze a rotaci zařízení jsou snímány infračerveným senzorem. Na rozdíl od některých ostatních zařízení je nutné Oculus Rift propojit kabelem s počítačem, který zajišťuje potřebné kapacity pro výpočet grafiky a dalších funkcí [14].

Microsoft Hololens

Microsoft Hololens je technologie rozšířené reality, která umožňuje projekci trojrozměrných objektů do reálného prostoru. Oficiální vydání verze pro vývojáře (zatím pouze pro účastníky programu Windows Insider v Kanadě a USA) proběhlo v roce 2016. Primárně by tato technologie měla být využívána v sektoru konstrukce, k vizualizaci navrhovaných modelů. Její další vývoj by měl směřovat i na využití v běžném životě a to nejen v zábavním průmyslu, ale také pro praktičtější využití (například návody na montáž, instalaci, údržbu...). Projekce je vytvářena speciálními brýlemi, ve kterých jsou umístěny snímače okolí a pohybu uživatele. Uživatel pak gesty rukou ovládá jednotlivé funkce brýlí a rozmísťuje hologramy trojrozměrných objektů do prostoru kolem sebe [8], [9], [10], [11].

Tuto technologii používá při prezentacích svých produktů například automobilka Volvo. „Hologramy“ umožňují zobrazit modely vozů a pomocí několika příkazů (nejčastěji gesta ruky snímaná zařízením umístěným ve speciálních brýlích) provádět různé funkce a vizuální efekty. Například je možné prohlížet interiéry vozů nebo „zneviditelnit“ karoserii a prohlížet si technologické uspořádání jednotlivých komponent vozu (podvozek, motor...). Dále lze vizualizovat vlastnosti vozů při běžném provozu nebo v nestandardních situacích (obtížné předjíždění, reakce bezpečnostních prvků a systémů bránících haváriím...) [8], [9], [10], [11].

Jedním z nejsoučasnejších projektů je například propojování technologie Microsoft Hololens a modelářského prostředí Autodesk Fusion 360. Cílem je zvýšit stupeň možnosti komunikace a spolupráce jednotlivých pracovišť v procesu vývoje produktů (zejména designérů a technických pracovišť) [8], [9], [10], [11].

Několik projektů je vytvářeno i ve spolupráci se vzdělávacími institucemi. Například na univerzitě Cleveland Clinic se snaží nahrazovat klasickou výuku lidské anatomie (založené na knižních ilustracích) zobrazováním prostorových modelů, které zobrazují velmi detailně uspořádání lidského těla. Přístupné jsou vizualizace svalů, kostí nebo oběhového či nervového systému, které díky své trojrozměrnosti a digitálnímu zpracování umožňují doposud velmi neobvyklý způsob zkoumání těchto soustav. V jiných oborech než ve zdravotnictví je technologie MH uplatňována například na univerzitě Case Western Reserve. Zde se s ní například vizualizují archeologické nálezy, na kterých je možné provádět digitální restaurátorskou práci a tím dosáhnout jejich předpokládaného původního vzhledu [8], [9], [10], [11].

V oblasti architektury a konstrukce využívá tento SW také společnost Trimble. Pro prezentaci připravovaných projektů se například využívá umístování „hologramů“ budov do prostředí, ve kterém by měli stát. Lze tak zkoumat celkové začlenění do prostoru, nebo upravovat některé parametry staveb. Zajímavým nástrojem je možnost zobrazovat stavbu ve skutečném měřítku (a nechat se jí „obklopit“), což například zjednoduší navrhování interiérů, nebo jejich následné předvádění. Toto zobrazení ve skutečné velikosti lze také uplatnit pro kontrolu již prováděných stavebních prací, nebo úpravu některých detailů a to v reálném čase přímo na místě během provozu [8], [9], [10], [11].

Laboratoř pro pohonné motory Národního úřadu pro letectví a kosmonautiku (NASA) připravuje ve spolupráci s Microsoft Hololens projekt Onsight. Mělo by jít o virtuální zobrazení povrchu planety Mars, jehož zkoumáním by mohli vědečtí pracovníci NASA vyhodnocovat a plánovat trasu a činnosti vesmírného vozítka Curiosity. Pro tvorbu virtuálního prostředí budou použity skutečné snímky okolí pořízené roverem Curiosity přímo na místě. Podle vedoucích pracovníků projektu by mělo dojít k přechodu vědců od monitorů k práci ve virtuálním prostoru až do té míry, že bude připomínat práci geologa na zemi. Zatím je vše v průběhu testování, rozběhnutí aplikace je zatím naplánováno na rok 2020 [8], [9], [10], [11].

Vuzix

V oboru virtuální reality vyvíjí společnost Vuzix například technologii nazývanou iWear Video Headphones. Z názvu je patrné, že se jedná o zařízení pro projekci obrazu a přehrávání zvuku. iWear Video Headphones jsou sluchátka s namontovanými displeji. Podporovány jsou i HDMI vstupy, tzn. je možné připojení zařízení k různým přehrávačům videa, herním konzolám a počítačům, dokonce i k některým mobilním telefonům. Produkt je zaměřen především pro využití v zábavním průmyslu, v konceptu digitální továrny není předpokládáno významné podpory. Na stránkách výrobce zatím není uveden datum uvedení na trh [23].

Co se týká rozšířené reality, ve stádiu vývoje je například zařízení M2000AR (opět se jedná o polopropustné brýle), jehož úkolem by mělo být zprostředkovat bezdrátový přístup k informacím a obsahu RR v oblasti průmyslové výroby. Primární využití by měl nalézt při vývoji prostředí RR, výcvikových aplikací zaměřených na údržbu a opravy strojů, případně pro výcvik personálu tyto stroje obsluhujícího a provozujícího. I zde je podporováno připojení HDMI kompatibilní se všemi běžnými platformami (PC apod.) [23].

3 Analýza možností SW nástrojů Unity a Blender

Software byl vybírán zejména na základě možnosti využití pod volnou licencí a po doporučení vedoucího práce a odborného konzultanta. Zde jsou uvedeny některé vlastnosti a možnosti použitých programů. Tento text neslouží jako návod na používání uvedených programů.

3.1 Unity 3D

Pro tvorbu prostředí DP, tvorbu skriptů a výsledné vizualizace bylo zvoleno vývojové prostředí Unity. Důvodem byla možnost kvalitního grafického zpracování (podpora 2D i 3D), velká možnost kompatibility s většinou známých druhů operačních systémů, webových prohlížečů a se zařízeními používanými v oblasti virtuální reality, široká podpora různých skriptovacích jazyků a poměrná nenáročnost na ovládání tohoto počítačového programu. Další výhodou je existence nástrojů pro testování. Například možnost prohlížet již vytvořené vizualizace bez nutnosti vytvářet build (zkompileovaná verze vyvíjeného software) nebo provádět změny ve zrovna spuštěném programu [1]. Zde jsou popsány některé základní nástroje vhodné pro tvorbu připravovaného projektu a orientaci ve vybraném vývojovém prostředí. Unity lze po registraci využívat zdarma pouze pro osobní využití.

3.1.1 Layouts

Unity umožňuje různými způsoby upravovat rozložení pracovní plochy a tím optimalizovat pracovní podmínky vývojáře. Na výběr je několik předdefinovaných variant (default, 2 by 3,

4split, tall, wide), každá s různými možnostmi zobrazení scény a karet nástrojů. Není nutné se držet přednastavených hodnot, rozložení lze také podle potřeby modifikovat posouváním jednotlivých propriet [1].

3.1.2 Views

Jedná se o možnosti zobrazení (různé náhledy) specifických komponent projektu. Základním pohledem v Unity je scéna (**SceneView**). Do ní se umísťují a dále upravují všechny objekty, které následně vytvářejí projekt. Objekty stačí pouze tažením myši přesunout například z knihovny assetů (Assets) na plochu scény a následně jim přiřadit požadovanou polohu, velikost a další parametry. K zobrazení hierarchických vazeb mezi objekty slouží **Hierarchy View**. Zobrazují se zde automaticky názvy všech objektů umístěných na scéně a opět pouhým přetahováním jejich polohy lze nastavovat různou podřízenost mezi nimi. Při výběru objektu v tomto seznamu se pohled scény přizpůsobí tak, aby byl zvolený objekt umístěn ve středu. Tento nástroj může značně usnadnit práci s vysokým počtem objektů. Dalším oknem, které zobrazuje všechny komponenty projektu (objekty, textury, skripty, ...) rozdělené do příslušných složek je **Project View**. Tato položka značně zpřehledňuje práci v rozsáhlejších projektech. Pro zobrazení detailů jednotlivých objektů a nastavování jejich vlastností slouží **Inspector View**. Po poklepání myši na daný objekt se zobrazí možnosti nastavení vždy v závislosti na konkrétním typu objektu. Typicky lze nastavovat souřadnice polohy, přizpůsobení rozměrů, vlastnosti vzhledu apod. U objektů typu kamera lze pak upravit vlastnosti projekce, perspektivu a jiné propriety pro renderování. Pohled hlavní kamery nazvaný **Game View** se spustí poklepáním na tlačítko přehrávání, které je standardně umístěné v horní části projektového okna. Využívá se s výhodou zejména při testování funkcí vytvářeného projektu, kdy lze provádět změny rovnou v běžícím programu, které se projeví pouze v Game View a nemají vliv na uložený projekt [1].

3.1.3 TransformTools

Transform tools je sada nástrojů používaná k interakci s pohledem scény. Je tvořena několika tlačítky s příslušnými funkcemi. Umožňuje pohybovat celou scénou, posouvat nebo rotovat pouze jednotlivými objekty, nebo provádět na těchto objektech různé modifikace (např. změna rozměrů) [1]. Celé ovládání je ovšem velmi intuitivně laděné, a proto lze TransformTools velmi snadno nahradit pouze tlačítky myši.

3.1.4 Cameras

Kamery jsou využívány pro renderování (neboli tvorbu reálného obrazu na základě počítačového modelu) scén. Z jejich vlastností lze například nastavit perspektivu a tím ovlivňovat výsledný grafický výstup. Při označení kamery se také zobrazí malé okno (tzv. Camera Preview) v rohu obrazovky, které ve snížené kvalitě zobrazuje pohled kamery, čímž umožňuje vyladit její polohu před výsledným renderováním a ušetřit tak výpočetní čas.

3.1.5 Assets

Assety (z angl. výrazu assets pro užitečné nebo hodnotné věci) je označení pro objekty ve vývojovém prostředí Unity. Je nutné je nechávat pouze jako „fyzické“ objekty, ale také jako skripty, shadery, materiály a další propriety potřebné pro vytváření virtuálního prostředí. Můžeme rozlišit několik kategorií assetů. Některé jsou přímo součástí základního

programového vybavení Unity 3D. Přednastavené assety jsou často tvořeny knihovnami objektů základních tvarů (koule, kvádr...) a umožňují zrychlit části práce, ve kterých nejsou kladeny příliš vysoké požadavky na tvarovou složitost objektů (stačí definovat pouze základní rozměry a umístění). Tyto nejjednodušší objekty se také s výhodou využívají při výuce s programem Unity 3D (snadnější pochopení jednotlivých funkcí, opět bez nutnosti složitého nastavování nebo modelování) nebo testování skriptů atd. Lze je i dále upravovat přímo v Unity, ale často se pro tvorbu vlastních tvarů využívá jiný software (v této práci například Blender). Druhou, o stupeň složitější skupinou jsou assety vytvořené třetími stranami. Tyto assety je možné stáhnout z webových stránek (Unity Asset Store) a do programu je následně importovat. Lze je využít pro pouhé doplnění zobrazovaného prostoru (nábytek, automobily...) i jako hlavní součásti celého projektu. U propracovanějších a komplikovanějších assetů je nutné počítat s tím, že se pravděpodobně nebudou vyskytovat ve formátech volné licence. Také je nutné dbát na to, zda máme dostatečné oprávnění využívat assety třetích stran pro zamýšlené účely. Třetí a poslední skupina jsou assety přímo vytvořené uživatelem. Je možné je vytvářet kombinací výše zmíněných dvou skupin, nebo rovnou definováním vlastních tvarů (nejčastěji v jiném prostředí než Unity – např. Blender, Sketch Draw...). Vývojář je zde omezen pouze vlastnostmi programu, je ovšem nutné brát ohledy na náročnost časovou i výpočetní (např. drobné detaily na relativně velkém objektu není nutné modelovat, ale využít s výhodou textury apod.). Přístup ke knihovnám assetů se také značně liší podle toho, zda je využívána placená nebo volná verze programu. V programu Unity 3D se s assety pracuje pomocí rozbalovacího menu umístěnou pod tlačítkem Assets nebo Game Object a zvolením příslušných funkcí (např. zvolením GameObject dále 3D object otevře knihovnu assetů základních tvarů). Assety jsou obvykle sdružovány do určitých celků, často podle společných tvarů nebo funkcí, nazývaných knihovny, balíky apod. [1].

3.1.6 Skriptování

Skriptování znamená psaní zdrojového kódu, který obsahuje a popisuje informace, data a funkce jednotlivých objektů a vztahy mezi nimi. Pro tuto práci bude zvolený programovací jazyk C#, který je podporován zvoleným vývojovým prostředím, umožňuje využívat funkce (tzn. částí programu popisujících „činnosti“) a který je také objektově orientovaný (tzn., umožňuje tvorbu tříd, ve kterých se mohou ukládat informace o objektech). Cílem této práce není seznamovat čtenáře s detaily tohoto jazyka. Součástí programu Unity je prostředí zvané Mono Develop, které umožňuje psaní kódu Přímou v Unity ve všech podporovaných jazycích. Jak již bylo výše zmíněno i skripty patří mezi assety. Tedy tvorba nového skriptu se provádí obdobně jako u assetů, lze je i získávat od třetích stran (např. Unity Store). Skriptování může být i značně komplikovaná záležitost. Před tvorbou projektu by měli být naplánovány struktury jednotlivých funkcí i požadavky na to jakými daty budou objekty reprezentovány (samozřejmě s přihlédnutím k rozsahu řešeného problému) [1]. Aby se funkce jednotlivých skriptů v prostředí nějak projevila, je nutné k nim přiřadit příslušné objekty (nejméně přetažením skriptu v Project View nebo v Hierarchy View na zvolený objekt).

3.1.7 Možnosti využití Unity

Pokud jsou výše zmíněné nástroje vývojového prostředí vhodně využity, mohou mít následné výstupy velmi vysokou kvalitu a to jak z hlediska funkčnosti, tak z hlediska vzhledu. Nejčastěji Unity je využíváno pro tvorbu počítačových her, nicméně z hlediska virtuální reality je i u takovýchto projektů určitá podobnost s digitální továrnou.

Tutoriály a vzorové projekty

Prvními projekty, se kterými se může běžný uživatel Unity setkat, jsou volně dostupné výukové verze z Unity Store. Tyto projekty se často vyznačují velmi základním, ale přesto poměrně kvalitním zpracováním. Jde o nepříliš komplikované hry, které mají sloužit k výuce základních postupů tvorby v tomto vývojovém prostředí, např. modelování, práci se zvukem, fyziku objektů nebo herní architekturu. Mezi nejvíce stahované patří například tutoriál Tanks! (viz Obrázek 4) nebo VR samples od Unity Technologies (viz Obrázek 5).



Obrázek 4 - Tanks![25]

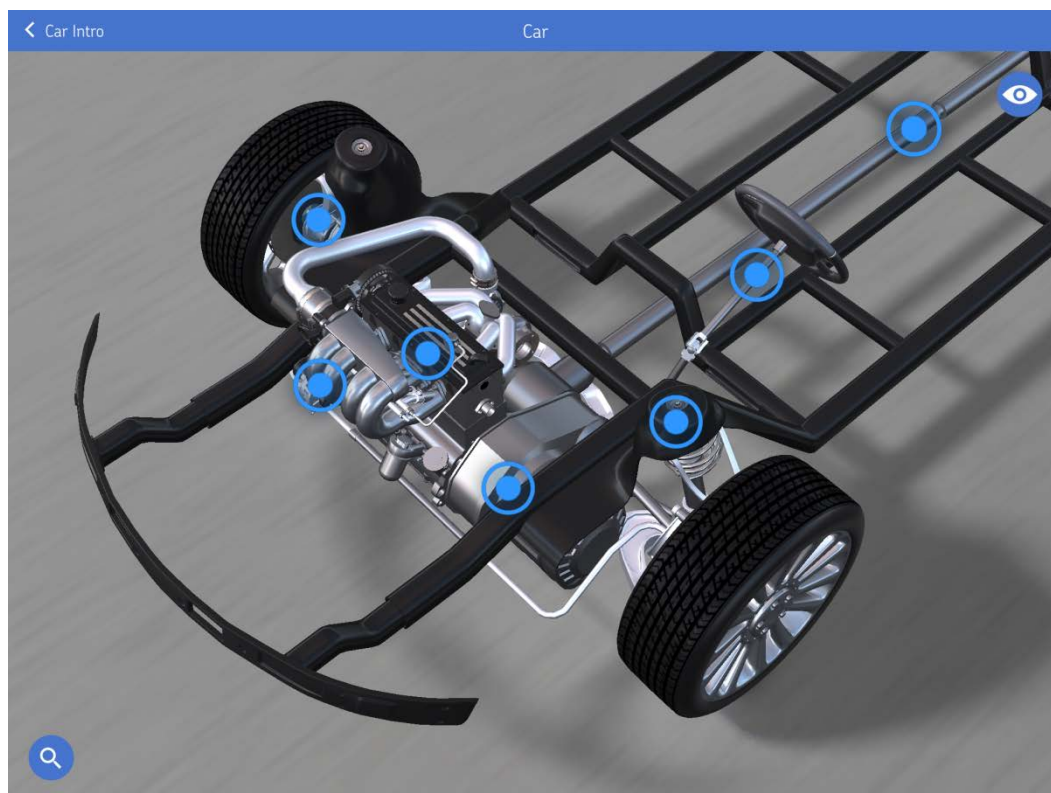


Obrázek 5 - VR smaples[26]

Non-games

V Unity se samozřejmě nevytváří pouze herní SW. Některé aplikace, zvláště z oblasti strojírenství stojí za prozkoumání.

Jedním z programů je například projekt **Virtual Car** (viz Obrázek 6) od Animech Technologies, zaštitěný firmou SKF. Jedná se o interaktivní 3D aplikaci, která umožňuje prohlížet a ovládat speciálně navržené automobily. Největší zajímavostí je velmi precizně zpracovaný systém jednotlivých dílů, které jsou velmi detailně vymodelované a animované, včetně velmi malých součástí (např. ložiska apod.) [28]. Původním záměrem je poukázat na části, které se při běžném provozu nacházejí pod povrchem, a proto příliš nevynikají a potřeba jejich kvalitního zpracování nemusí být na první pohled zřejmá. Aplikace by měla sloužit pro prezentování těchto součástí a jejich provozu zákazníkům SKF, a to i těm, kteří nejsou vzděláni v oboru strojírenského inženýrství. Tím by mělo být dosaženo časové úspory při uzavírání obchodních smluv [27].



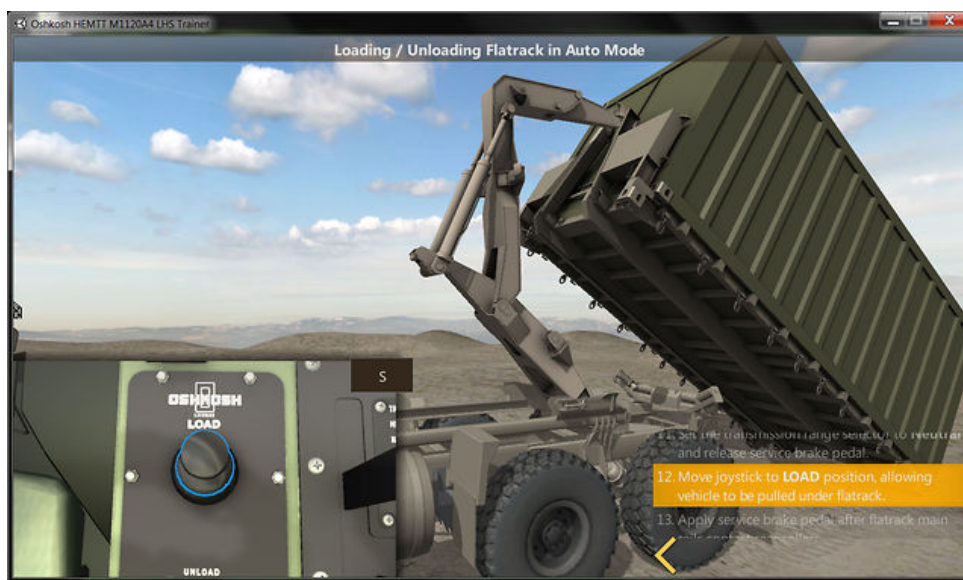
Obrázek 6 - Virtual Car[27]

Další aplikací, která stojí za zmínku, je **Ferrari ARShowroom** (viz Obrázek 7) od Pleribus. Byla optimalizována pro zařízení iPad a klade si za cíl vytvořit prostředí rozšířené reality, které slouží pro prezentaci vozů Ferrari, včetně jejich technických parametrů. Aplikace využívá snímač v zařízení a SW pro rozpoznávání tvarů jednotlivých vozů a následně se na jejich snímcích zobrazuje doplňkový obsah s náležitými informacemi; uživateli je umožněno dále u modelu měnit některé parametry týkající se vzhledu a technologického vybavení a prohlížet je [29].



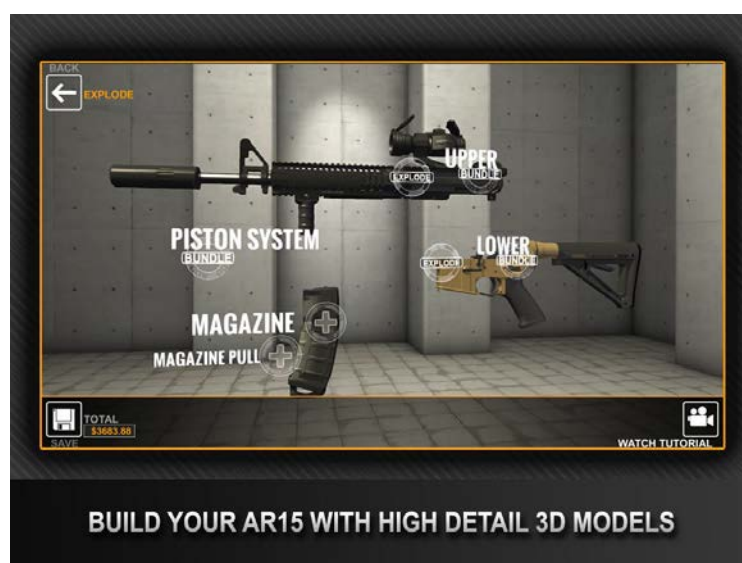
Obrázek 7-Ferrari AR showroom[29]

Virtuálním vzděláváním a tréninkem v oblasti ovládání těžké techniky se zabývá například program **Oshkosh HEMTT VirtualTraining** (viz Obrázek 8) od vývojářů Heartwood. Hlavní myšlenku zde tvoří potřeba nahradit zastaralé výukové nástroje, jejichž vlastnosti jsou pro dnešní dobu nedostatečné. Snahou je zde učinit ovládání více intuitivní a hlavně zobrazované modely opatřit lepší zpětnou vazbou a animacemi a tím umožnit výuku ve smyslu Learn-by-doing (výuka praxí). Užívání aplikace by mělo vést na snížení časové potřeby pro trénink a tím na úspory s ním spojených nákladů [31]. Aplikace umožňuje zobrazit například panely ovládání v interiéru vozu, nebo dále i vnější ovladače.



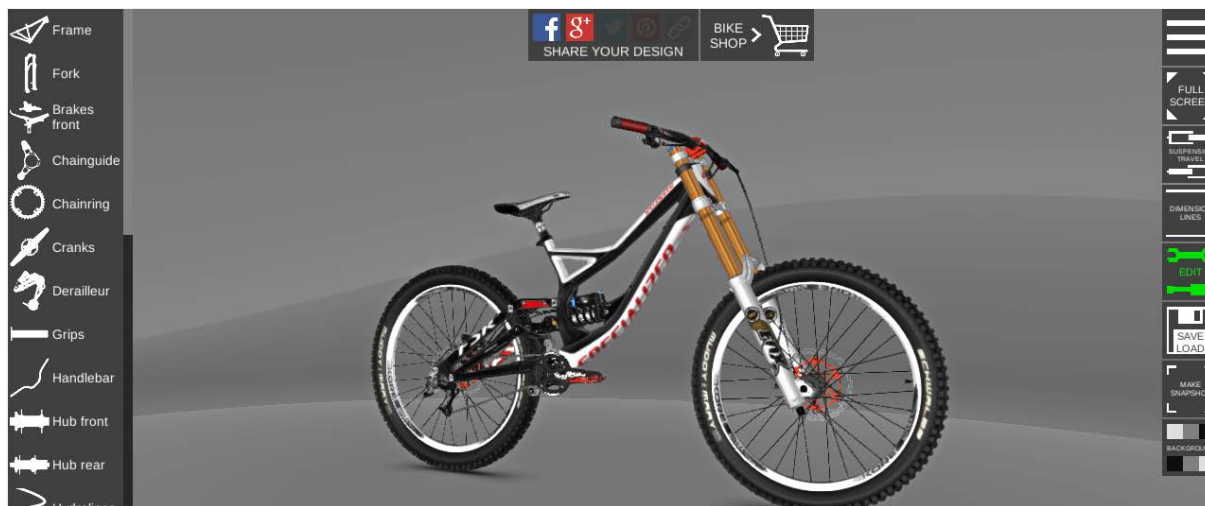
Obrázek 8-Oshkosh HEMTT VirtualTraining[31]

V Unity byla také vytvořena webová aplikace Gunstriction (viz Obrázek 9), která se zabývá konstrukcí dílů různých střelných zbraní a jejich montáží. V současné době pro ni zatím nebylo nalezeno uplatnění v praxi a slouží pouze pro umístování reklam třetích stran [32]. Z hlediska prezentace možností vývojového prostředí Unity však stojí i tento program za zmínku.



Obrázek 9-Gunstriction[32]

Webová aplikace Bike Config (viz Obrázek 10), jak vyplývá již z názvu, se zabývá úpravou, montáží a konfigurací součástí jízdních kol. Slouží pro prezentaci možností pro zákazníky, kterým je umožněno si nejen vybrat vlastní sestavu, ale následně také upravovat vzhled jednotlivých dílů [40].



Obrázek 10-Bike Config[40]

3.2 Blender

Pro tvorbu vlastních objektů virtuálního prostředí je možné použít různé softwarové nástroje. V této práci byl zvolen program Blender (verze 2.76). Hlavním důvodem je mnoho funkcí (modelování, UV mapování, texturování...) vhodných pro tvorbu modelů v požadované kvalitě a následná možnost jejich exportování (např. pouhým přetažením zvoleného objektu myši do knihovny Assetů) do potřebného formátu, vše pod licencemi freeware/opensource. Vhodná je i kompatibilita vybraného SW s téměř všemi běžně dostupnými operačními systémy.

Uživatelské prostředí programu Blender umožňuje při správném nastavení velmi intuitivní přístup ke všem funkcím přímo z úvodní obrazovky, bez nutnosti složitého přepínání mezi okny jednotlivých nástrojů. Okno programu je členěno na několik částí, mezi nimiž je určitá hierarchie. Editory (**Editors**) jsou nástroje, pro zobrazování a úpravu různých typů dat, jsou rozdělené podle jednotlivých vykonávaných funkcí. Okna jednotlivých editorů obsahují komponenty a nástroje členěné do tzv. regionů (**Regions**). Ty umožňují zobrazit komponenty editoru pro specifický úkol a lze mezi nimi vybírat přepínáním příslušných nástrojových karet tzv. **Headers**. Nejmenšími organizačními prvky jsou pak panely (**Panels**). Jsou to rozbalovací prvky, které obsahují tlačítka pro další funkce editorů. Pracovní plochu a ostatní okna editorů lze snadno rozdělit na více částí a usnadnit tak orientaci při tvorbě složitějších objektů (např. různé pohledy v různých oknech) [16], [17]. Níže jsou popsány některé nástroje vhodné pro tvorbu připravovaného projektu a orientaci ve zvoleném vývojovém prostředí.

3.2.1 Módy

Program Blender nabízí na výběr různé módy (**Modes**), které zpřístupňují různé funkce pro konkrétní práci s vybranými typy objektů. To v jakém módu se pohybujeme, ovlivňuje zobrazování ovládacích panelů a kontrolních tlačítek, mění chování objektů v různých pohledech nebo mění dostupnost jednotlivých funkcí. Nejzřetelněji se změna módu projeví různými vzhledy pracovní plochy. Zde jsou popsány základní z nich, u kterých je předpoklad, že budou využity pro tvorbu projektu k bakalářské práci [16], [17].

Objektový mód

Objektový mód (**Object mode**) je nastaven jako výchozí po spuštění programu. Jako jediný umožňuje přepínání výběru mezi objekty všech typů a další práci s nimi (posun, rotace, translace, změna velikosti...) [16], [17]. Základním vzhledem pracovní plochy v tomto módu je tzv. solid, ve kterém jsou zobrazeny pouze vnější obrysy jednotlivých tvarů a základní barva materiálu přiděleného různým objektům (tzn. bez textur apod.). V dolní liště lze přepnout do několika dalších zobrazení, které se odlišují vnějším vzhledem nebo například průhledností zobrazených objektů a tím usnadňují orientaci mezi větším počtem zobrazených těles, nebo například šetří čas pro výpočet konečného renderu. Na výběr jsou například nastavení Rendered, Material a Texture, které, jak již z jejich názvů vyplývá, zobrazují v různých stupních kvality i materiál (rozumějme vnější vzhled) objektů a usnadňují představu o výsledném zevnějšku bez nutnosti renderovat každou scénu, kterou potřebuje autor zkontrolovat nebo v ní doladit detaily. Protikladem k nastavení solid je pak tzv. Wireframe, jehož název opět napovídá, že se bude jednat o zobrazení „rámu“, neboli vnějších

obrysů těles. Poslední v seznamu je pak možnost Bounding box, po jejímž zvolení se zobrazí pouze vnější obrysy krychle opsané danému objektu. Tohoto nastavení se využívá například pro nastavení vhodné polohy pivotního bodu (např. geometrie krychle je lépe použitelná než komplikovaný objekt).

Editační mód

Editační mód (Edit mode) představuje prostředí pracovní plochy, ve kterém lze zpracovávat detailní části modelů. Při přepnutí do Edit Mode se zobrazí základní prvky pro práci s konkrétním typem objektu. Upravují se tak buď sítě vrcholů, hran a stěn (u primitivních Mesh objektů), nebo například kontrolní body křivek (u modelů jimi vytvářenými) [16][17]. Na spodní liště se po zvolení editačního módu opět zobrazí charakteristické ikony pro práci s objekty. Zvolením ikon lze vybírat mezi typem výběru vrcholů, hran nebo ploch (při práci s Mesh objekty) což určitým způsobem mění postup při modelování např. metodou Extrude. Dále se zde nachází tlačítko pro nastavení průhlednosti, které umožňuje přepnout mezi dvěma polohami, kde v jedné lze vybírat tvarové prvky pouze viditelné ze směru pohledu uživatele a ve druhé vybírat například vrcholy v celém objemu tělesa. Další důležitou funkcí v Edit Mode je rozbalovací menu Proportional Editing, jež umožňuje nastavení několika stupňů jakéhosi silového pole okolo zvoleného editačního prvku (Vertex, hrana, plocha), které ovlivňuje i nezvolené prvky v okolí a umožňuje tak dosáhnout značně komplikovaných tvarů, kterých by bez tohoto nástroje bylo velmi obtížné dosáhnout.

Vhodné je také zmínit, že změny provedené v Edit Mode (například změna polohy vůči pivotnímu bodu, změna rozměrů apod.) mají vliv na číselně vyjádřené hodnoty v dalších módech. To znamená, že pokud bychom měli například krychli s počátečním nastavením dimenze v každé ose na hodnotu 2 a následně ji v Edit mode zmenšili na polovinu, projeví se to přepočtem i v jiných módech.

Částicový mód

Částicový mód (Particle Mode) je určen pouze pro práci s objekty typu Mesh a je věnuje se zejména nastavování parametrů pro generování částicových systémů. Částic se využívá hlavně při tvorbě simulací dynamických objektů (jiskry, plamen, déšť...) [16], [17].

3.2.2 Řídící bod

Řídící bod (Pivot Point) reprezentuje střed všech rotací, změn měřítka, zrcadlení a dalších transformací objektů [16]. Na výběr je z několika možností nastavení jeho polohy, které jsou umístěny v rozbalovacím menu v levém panelu ovládacích prvků (zobrazí se ovšem pouze v Object Mode). Provést lze například nastavení přesných polohových souřadnic řídicího bodu a pak jim přizpůsobit geometrii objektu, nebo naopak vycentrovat pivot podle rozměrů modelovaného tělesa. Přesné polohy lze dosáhnout i „odesláním“ řídicího bodu do 3D kursoru (např. 3D kursor umístit na vybraný vertex a následně na tuto polohu přiřadit pivot). Zvolení možnosti Centre of Mass pak způsobí, že pivot reprezentuje střed hmotnosti, což je také důležité pro některé aplikace.

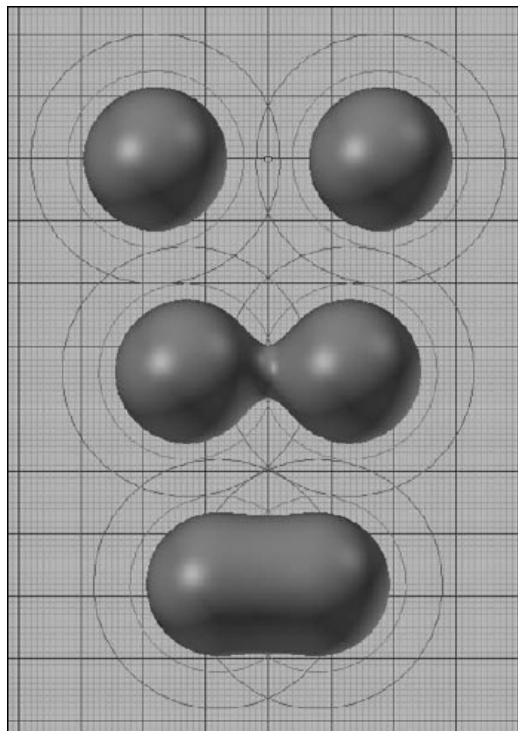
3.2.3 Modelování

Pro tvorbu prostorových virtuálních objektů je zapotřebí vhodně zkombinovat komponenty jako modely, materiály modelů a prostředí a světlo. Zaměříme se nejprve na část modelovací.

Výše zmíněný program Blender umožňuje zvolit mezi několika různými metodami modelování. Prvním druhem je skládání povrchové sítě objektů pomocí vrcholů (Vertex, Verticies), hran (Edges) a stěn (Faces) tzv. **Mesh Modeling**, který umožňuje vytváření tzv. primitivních objektů (**MeshPrimitives**; jejichž povrch je složen např. z trojúhelníků nebo jiných polygonů). Základní prvky (vrcholy, hrany, plochy) jsou popisovány metodami analytické geometrie (např. souřadnice vrcholů) a práce s nimi je poměrně jednoduchá. Vytváření nových primitivních objektů se provádí v Object Mode, jejich následné úpravy v Edit Mode nebo Sculpt Mode. Základním nástrojem pro úpravu tvarů primitivních objektů je vytahování základních prvků (vrcholů, hran, stěn), tzv. **Extrude**. Druhým typem je modelování pomocí křivek (**Curve Modeling**), kde se využívá různých typů křivek (definovaných kontrolními body), po kterých se následně vytahují požadované profily a tvary objektů. Výsledné tvary je možné snadno převést na objekty typu Mesh, ovšem často je výhodnější pro výpočtovou rychlost toto nečinit. Modelování povrchů (**Surface modeling**) je podobné křivkovému, ovšem s tím rozdílem, že je omezeno pouze na lineární tvary. Modelování za pomoci meta objektů (**Meta Object Modeling** – viz Obrázek 11) se podobá práci s Mesh Primitives, ale místo práce s vrcholy, hranami a stěnami se navíc využívá jejich vlastnosti podobné kapalinám (liquid-like quality), která umožňuje například částečné splývání objektů po jejich přiblížení a tím dosažení velmi komplexních tvarů, které by šlo jinými metodami jen velmi těžko vymodelovat. S výhodou se používá i při simulování chování tekutin. I zde je možné objekty následně konvertovat na typ Mesh. Dále se v Blenderu uplatňuje i nástroj pro modelování textu, který umožňuje jednotlivým písmenům přidat třetí dimenzi. Pro modelování lze použít i některé skripty (Blender podporuje programovací jazyk Python). S výhodou se využívají nejen pro navrhování vlastních Mesh objektů, ale například i pro simulování vlastností kapalin, textilních látek apod. Tvorba samotných zdrojových kódů může být velmi složitá, proto jsou na internetu dostupné knihovny skriptů, které lze do programu snadno importovat [16], [17].

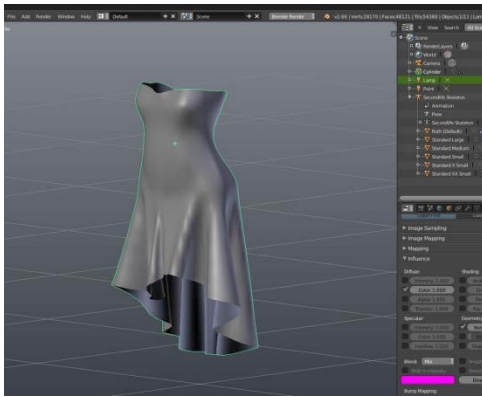
S výhodou se také využívají prázdné objekty (tzv. **Empties**), které se nezobrazují při výsledném renderování, ale slouží jako reference pro vazby mezi objekty (např. Rodičovský prvek skupiny objektů), nebo středy různých pohybů apod.

Dalším vhodným nástrojem pro úpravu tvaru objektů (modelování) jsou modifikátory (**Modifiers**). Přiřazením modifikátoru k daným objektům lze automaticky upravovat jejich vzhled a tvar, aniž by docházelo k faktické deformaci výchozí podoby. Provedené změny se mohou projevovat pouze při renderování, nebo v pracovním okně náhledu (pokud není nastaveno jinak). Opět zde lze dosáhnout i tvarů, jinak značně náročných na pouhé manuální modelování a to bez ovlivnění základní geometrie objektu. Na objekty je možné modifikátory aplikovat „ve vrstvách“, tzn. modifikátor, který je nastaven jako první, ovlivňuje vzhled po nastavení jemu podřízených. Na výběr je z mnoha možností a typů modifikátorů. Zde uvedeme jen několik nejdůležitějších. První skupinou modifikátorů je

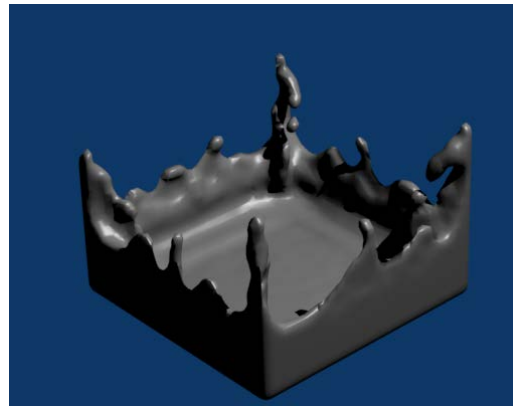


Obrázek 11- Příklad vzniku meta objektu[33]

Generate, která, jak již z názvu vyplývá, umožňuje generovat různé prvky, nebo skupiny prvků. Typickým představitelem této skupiny je vytvoření pole objektů, které se používá zejména pro tvorbu periodicky se opakujících tvarů, bez nutnosti každý definovat zvlášť. Je zde i funkce pro vyjmutí konkrétního objektu z pole, což umožní tvorbu modelů, které jsou si vzájemně velmi podobné, ale liší se pouze v drobných detailech. Dále do této skupiny patří například Edge Split, umožňující rozdělování hran požadovaným počtem vrcholů, což může sloužit třeba pro oddělování dalšího profilu pro tažení apod. Dobře využitelná je i funkce zrcadlení (Mirror), která se s výhodou využívá pro tvorbu osově souměrných modelů. Často se také využívá modifikátor Subdivision Surface, po jehož aplikování se převede povrch tělesa na jemnější polygonální síť a tím se odstraní ostré přechody a těleso se jeví jako oblé, to vše proběhne beze změny původní geometrie (tzn. krychli lze editovat stále jako krychli, přestože se jeví jako koule – vertexy krychle slouží jako řídicí body). Druhou skupinou modifikátorů je tzv. Deform, která pomáhá napodobovat různé deformace tvaru. Nejčastěji využívaným zástupcem této skupiny je Smooth modifier, který provádí výpočet povrchu tak, aby se jevil jako hladký. Za zmínku stojí i skupina Simulate, používaná při simulaci fyzikálních vlastností látek (kapaliny, textilie apod.). Obrázek 12 ukazuje příklad využití modifikátoru Cloth Modifier, Obrázek 13 pak Liquid Modifier. Je patrné, že bez použití modifikace, by bylo dosažení takového tvaru při nejmenším velmi pracné.



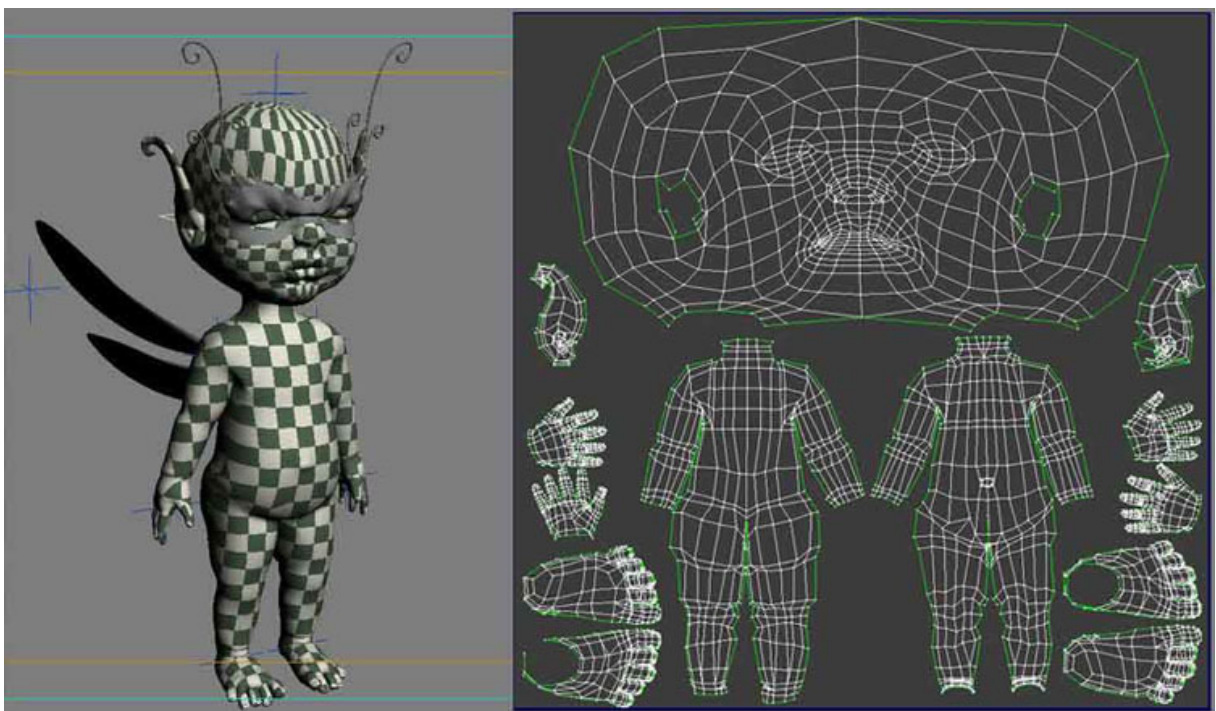
Obrázek 12-Cloth Modifier[20]



Obrázek 13-Liquid Modifier[21]

3.2.4 UV mapping

UV mapování (UV mapping) je technika používaná pro tvorbu textur „na míru“ daným objektům. Je určena pro práci s objekty typu MESH. Vhodným nastavením hran, podle kterých je těleso „rozřezáno“, je umožněna dekompozice povrchu objektu do roviny, jeho následné převedení na požadovaný obrázkový formát, který se následně vyplní požadovanými texturami, barvami apod. a dále je zpět „nabalen“ (funkce wrap) do trojrozměrného prostředí. S výhodou je možné toto použít pro velmi jednoduchá tělesa i objekty značně složitých tvarů a tím dosáhnout požadovaných vzhledů. Příklad této techniky viz Obrázek 14.



Obrázek 14-texturování pomocí UV mapování [34]

3.2.5 Možnosti využití Blenderu

Jak již bylo zmíněno výše, nabízí Blender díky svým funkcím poměrně široké možnosti využití. Při správné aplikaci funkcí a postupů se dá dosáhnout skutečně kvalitních výsledků po stránce grafické i funkční. V tomto oddíle budou popsány možnosti využití tohoto programu v praxi. Blender je často využíván k tvorbě krátkých animovaných videí i celovečerních filmů i seriálů; vytváří se v něm grafické podklady k počítačovým hrám, digitálním modelům apod.

V roce 2010 byl například zveřejněn krátký film Elephants Dream, který byl mimo jiné renderován pro trojrozměrnou projekci [35]. Přestože je celý tento projekt licencován jako Creative Common Attribution (obdoba volné licence), je velmi kvalitně zpracován. Velice zde kupříkladu vyniká, jakých výsledků lze v Blenderu dosáhnout na poli modelování „živého“ materiálu (viz Obrázek 15) a jeho následného rozpohybování pomocí animací.



Obrázek 15-Elephants Dream[35]

Další ukázkou modelování živých objektů a zejména pak využití částicových systémů aplikovaných při simulování nejen vzhledu jednotlivých postav, ale i efektů okolního prostředí, je Nizozemský filmový projekt Big Buck Bunny. Z obrázku 16 je zřejmé, že výsledný efekt dosahuje značně vysoké kvality.



Obrázek 16-Big Buck Bunny[36]

Blender byl využit i při tvorbě promočního videa (viz Obrázek 17), jejímž zadavatelem byla automobilka BMW [37]. Tento projekt dobře prezentuje, že v tomto grafickém editoru je možné vytvořit i modely strojních zařízení s vysokou přesností a kvalitě odpovídající i těm nejvyšším požadavkům.



Obrázek 17-BMW Promo Blender[38]

Další ukázky práce a možnosti využití Blenderu (viz Obrázek 18) je možné najít v nepřeberném množství na různých internetových fórech (např. blender.org) věnovaných právě tomuto programu.



Obrázek 18-možnosti Blenderu[39]

4 Tvorba a popis modelů (komponent)

Jak bylo řečeno v úvodu, bude v této práci kladen důraz na vytvoření knihovny modelů použitelnou v programu Unity 3D. Knihovna bude obsahovat modely různých strojních zařízení, inventárního vybavení a jiných objektů, které bude následně možné využít pro tvorbu interaktivního prostředí simulujícího chod digitálního podniku. V této kapitole budou popsány jednotlivé modely vytvořené autorem práce, dále postupy, které byly při jejich modelování v programu Blender použity a následně bude vysvětleno, jakým způsobem byly modely převáděny do prostředí Unity a způsoby řešení některých komplikací, které se během tohoto importování vyskytovaly.

Úvodem je třeba zmínit několik zásad, které se obecně týkají každého vytvářeného modelu.

Kromě vhodného aplikování jednotlivých SW nástrojů a modelovacích metod, je také u každého modelu nutné dbát na smysluplné pojmenování jednotlivých částí, jehož správná struktura značným způsobem usnadní a zpřehlední budoucí práci s modely, zejména při přiřazování funkcí (např. pomocí skriptů) jednotlivým prvkům. Jistě se při programování kódu, nebo při doladění konečného vzhledu snadněji zorientujeme, pokud hledáme mezi názvy například VrtakUlozeni (podle programátorské konvence oddělujeme jednotlivá slova názvu pouze velkými písmeny), než součást označenou například defaultně circle.001.

Přestože během modelování nebudou explicitně nastavovány všechny rodičovské vazby mezi jednotlivými prvky, neznamená to, že na ně nebude vůbec brán zřetel. U modelů vytvářených v této práci předpokládáme jejich importování do Unity, kde jim budou následně přiděleny některé funkce, kterým by měla být přizpůsobena i hierarchie prvků. Je proto nutné nahlížet na objekty v tom smyslu, že nejsou statické, ale s největší pravděpodobností budou v budoucnosti vykonávat nějaký pohyb a tomuto faktu přizpůsobit jednotlivé modelovací postupy, uspořádání a propojenost jednotlivých částí modelů.

Důležitým parametrem zde bude poloha pivotního prvku, který reprezentuje střed hmotnosti nebo geometrie a zejména střed pohybu, translací, rotací a transformací každé součásti. Nejčastěji se jeho poloha nastavuje zvolením správné základny (např. označením objektu, nebo skupiny vertexů), do které se umístí 3D kurzor a v Object módu na panelu tools se do něj nastaví poloha pivotu.

4.1 Modely

Zde se konkrétněji podíváme na proces, ve kterém jednotlivé objekty obsažené ve vytvářené knihovně vznikaly. Nejprve je vždy zmíněn důvod, proč byl daný model vybrán. Dále je třeba analyzovat a popsat důležité rozměry jednotlivých částí a tím stanovit nejlepší sled modelovacích technik pro dosažení požadovaných tvarů. Podrobnější popis postupu modelování je uveden vždy pod samostatným nadpisem; důraz byl zejména kladen na ty techniky, které jsou při tvorbě daného modelu něčím specifické (např. modelování dopravníkového pásu je popsáno podrobněji, než pouhé upravování rozměrů primitivních objektů). V poslední řadě jsou konstatovány některé nepřesnosti při importu do Unity a jejich odstranění.

Je dobré si uvědomit, že tato kapitola neslouží jako podrobný návod na modelování, ale pouze jako přehled použitých postupů. V současné době se jako nejlepší formát podrobných návodů jeví krátká instruktážní videa, která jsou běžně dostupná na internetu (např. viz seznam zdrojů). Z časových a technologických důvodů do této práce nebyla přidána žádná videa s návody. Některé detaily postupů se mohou lišit v závislosti na zvolené verzi programu.

4.1.1 Model dveří

Za předpokladu, že by knihovna modelů měla sloužit k simulaci provozu digitální továrny, je jasné, že by její součástí měl být i tak základní objekt jakým jsou dveře. Přestože se jejich tvorba na první pohled jeví jako velice triviální záležitost, při zjištění počtu jednotlivých součástí a požadavků na jejich přesnost se toto zdání poněkud změní.

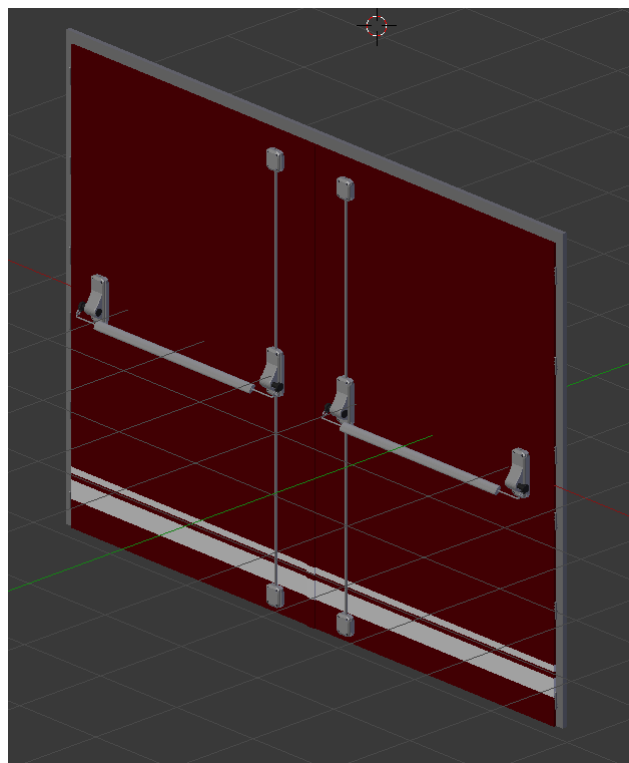
Poměrně přesné musí být nejen uložení dveří v pantech, ale i rozměry jednotlivých, na sebe dosedajících křídel, včetně rozměru obvodového, který přiléhá k obložení dveří. Díky využití funkce *Separate*, která umožňuje v editačním módu ze kteréhokoliv mesh objektu oddělit vybraný oddíl vertexů, hran či ploch je dodržení této přesnosti (u na sebe navazujících součástí) poměrně nekomplikovanou záležitostí. Přesnější postup modelování bude popsán níže.

Tvaru jednotlivých prvků je dosaženo vytahováním profilů pomocí metody *extrude* a vhodným aplikováním některých modifikátorů (*Subdivisoin Surface*, *Mirror*, aj.) jejichž fungování je popsáno v kapitole 3.2. Z hlediska dynamiky tohoto objektu je nutné zohlednit umístění pivotních prvků (*Pivot Point*) některých součástí, které zastupují střed hmotnosti a zejména střed pohybu každého tělesa. Při dodržení správné hierarchie rodičovských vazeb mezi prvky bude stačit provést vhodné nastavení pouze u objektu reprezentujícího dveřní panty a uložení otevíracího mechanismu dveří.

Postup tvorby

Jako nejjednodušší se jeví začít vymodelováním jednoho z dveřních křídel a na ně následně „nabalit“ ostatní části. Vložíme tedy základní krychli (*Shift+A* → *Mesh* → *cube*), jejíž dimenze upravíme do požadovaného obrysu křídla. V editačním módu označíme jednu z čelních stěn a metodou *extrude* ji roztáhneme do požadovaného rozměru. Využitím funkce *Separate* kopírujeme potřebné rozměry, ze kterých následně extrudujeme tvar navazujícího křídla a dalších součástí. Tvar pantů extrudujeme ze základní kružnice (*Shift+A* → *Mesh* → *circle*),

pro reprezentaci uchycení ke stěnám a křídům postačí vložení základního válce (Shift+A → Mesh → cylinder). Vzhledem zanedbatelným rozměrům pantů k celému objektu není nutné nastavovat modifikátor, ani shader pro zaoblený vzhled. Vhodně aplikujeme modifikátory pole prvků (Array) a zrcadlení (Mirror), čímž nastavíme požadovaný počet i rozmístění dveřních pantů, bez nutnosti několikanásobného manuálního duplikování/kopírování. Protože panty na jednotlivých stranách dveří na sobě nejsou závislé, ani netvoří jednotlivý celek, je nutné ještě využitím funkce Separate oddělit (tzn. vyčlenit z pole) alespoň jednotlivé segmenty na levé a pravé straně. Po oddělení dbáme na smysluplné pojmenování (např. s indexem P/L pro jednotlivé strany). Řídicí prvek nastavíme označením kružnice, která ohraničuje stykovou plochu mezi horním a spodním dílem pantu, umístíme do ní 3D kursor a následně do něj pivot. Tím vznikne vhodný základ pro rotaci kolem osy z (což je předpokládaný pohyb pantu). Základ uložení kliky vytváříme ze základní krychle, ve které nejprve, v Edit Mode, přidáme do stěny v čelním pohledu užitím funkce Face Loop Cut 2 nové hrany a to jak ve směru horizontálním tak směru vertikálním. Zvětšíme jejich rozměr podle příslušných os, čímž vytvoříme sražení obvodových hran. Upravíme dimenze krychle do požadovaného tvaru. Funkcí Separate oddělíme přední čelní plochu, kterou následně mírně zmenšíme a extrudujeme do tvaru krycího víka uložení kliky. Vložíme základní válec (bude sloužit jako reference pro díry), jako samostatný objekt, upravíme jeho rozměry do tvaru díry pro šroub ve víku, umístíme jej do levého horního rohu víka a nastavíme jeho řídicí bod shodně s pivotem víka. Aplikujeme zrcadlový modifikátor takovým způsobem, aby výsledné kopie válce vyplnily zbylé 3 rohy víka. Nakonec na objekt víka aplikujeme modifikátor Boolean a jako referenční objekt nastavíme výše zmíněný válec. V poli Operation nastavíme hodnotu Difference, tím vznikne zamýšlený tvar děr pro šrouby. Po kliknutí na tlačítko Apply lze smazat referenční válec a dále pracovat s objektem víka, ve kterém zůstaly požadované otvory. Samotné uložení kliky modelujeme ze základní kružnice, ze které smažeme vhodný počet vertexů a doextrudujeme do výsledného tvaru. Profil uzavřeme, vytvořením hrany mezi krajními vertexy (stisknutím F) a následně jej extrudujeme pro dosažení požadovaného rozměru. Provedeme sražení vnějších hran podobně, jak již bylo popsáno výše. Vložením válce a nastavením modifikátoru Boolean, „vyvrtáme“ otvor pro kliku (podobným postupem jako u víka). Tyto 3 objekty (základ, víko, uložení) spojíme pro lepší manipulaci do jednoho dílu (označení všech 3 a zkratka Ctrl+J). Konečnou podobu kliky vytvoříme extrudováním základní kružnice, zahnutí vytváříme využitím funkce Spin. Oválný průřez madla kliky vytvoříme změnou dimenze



Obrázek 15 - dveře

základní kružnice v požadovaném směru jedné ze souřadnicových os. Šrouby vkládáme z knihovny základních objektů v blenderu (pokud není aktivovaná, lze její zobrazování nastavit v menu User Preferences). Víko pro zarážku dveří vytvoříme pouhým duplikováním tvaru pro uložení kliky a úpravou jeho dimenzí (v Object Mode). Tyč tvořící zarážku pak pouhým vytažením základního válce. Takto vzniklou konstrukci otevíracího mechanismu dveří pak 4x duplikujeme a nastavíme polohu duplikátů tak, aby byl otevírací mechanismus vhodně umístěn na obou dveřních křídlech. Jako materiál použijeme základní barvy, bílé pruhy na křídlech vytvoříme přidáním a vhodným umístěním hran (funkcí Face LoopCut) a následným přiřazením druhého materiálu plochám mezi těmito hranami. Výsledný objekt je vidět na obrázku 19.

Díky zvolenému postupu při modelování nevznikly u tohoto modelu žádné problémy při importování do knihovny assetů v Unity.

4.1.2 Model frézky

Frézka patří k často využívaným strojům v podnicích zaměřených na výrobu. Při simulaci prostředí digitální továrny proto bude vhodné jí do knihovny navrhovaných modelů zahrnout. V porovnání s objektem dveří je zřejmé, že tvorba tohoto modelu bude o něco náročnější. Pokud ovšem zvolíme vhodné modelovací techniky, mělo by být dosaženo dostatečně kvalitního výsledku bez zbytečných komplikací.

Ze základních rozměrů je třeba si všimnout zejména návaznosti nosného ramene a vodícího sloupu; dále pak dosednutí pojezdového mechanismu nesoucím obráběcí nástroje a další příslušenství. Využitím funkce Separate by opět měla být dodržena dostatečná přesnost navazujících ploch.

Z modelovacích postupů opět nejčastěji využijeme metodu Extrude. Zvláštní pozornost pak věnujeme modelování obráběcího nástroje (frézy), kde využijeme modifikátor pro tvorbu šroubových ploch (ScrewModifier) a modelování pomocí Bezierových křivek.

Postup tvorby

Nejlepší bude začít modelováním základní desky, kterou „vytáhneme“ ze základní krychle. Hrany krychle rozdělíme metodou Face Loop Cut, vhodně rozmístíme nově vzniklé hrany změnou rozměru obvodových ploch, vytvoříme sražení hran a dále extrudujeme nohy podstavy a prohlubeň pro upevňovací lišty. Upevňovací lišty vytvoříme upravením rozměru základní krychle a aplikací modifikátoru Array. Trojúhelníkový půdorys základny pro vodící sloup vytvoříme tak, že na vhodně rozměrově upravenou polovinu základní kružnice aplikujeme funkci Spin, odstraníme překrývající se vertexy (funkce RemoveDoubles) a uzavřeme profil vytvořením hran mezi krajními body polokružnic (zvolit vertexy a stisknout klávesu F). Takto vzniklý tvar extrudujeme do konečné podoby. Vložení vhodně upravených základních válců a nastavením modifikátoru Boolean vytvoříme díry pro šrouby a následně otvor pro vodící sloup (postup je obdobný jako u modelování dř u výše zmíněných dveří). Tvar vodícího sloupu vytvoříme extrudováním základní kružnice. Převodovou skříň vytvoříme spojením základního válce a mírně upravené základní krychle, otvory pro šrouby vzniknou již několikrát výše zmíněným postupem (upravený válec + modifikátor Boolean).

Rozměry horního víka převodové skříně dostaneme využitím funkce Separate a extrudováním do požadovaného tvaru.

Dosažení tvaru nosného ramene není v porovnání s dosud popsány objekty tolik triviální, pojďme ho proto popsat trochu blíže. Začneme oddělením (funkce Separate) spodního průměru nosného válce, z něhož následně vytvoříme polokružnici a vyextrudujeme základní objímku sloupu, z jednoho okraje pak vytáhneme plochu pro spojení šrouby. Na takto vzniklý objekt aplikujeme modifikátor Subdivision Surface. Vzniklý tvar neodpovídá požadovanému (je příliš oblý a deformovaný), proto musíme nastavit váhu hran, u kterých chceme dosáhnout ostrosti. To provedeme zvolením požadovaných hran a spuštěním funkce Crease Subsurf Edge (Shift+E) a nastavením hodnoty váhy na maximum (buď tažením myši do krajní polohy, nebo číselně +1). Změna váhy se v Edit Mode projeví barevným zvýrazněním, v Object Mode pak dosažením kýženého tvaru. Pro další práci je nutné stisknout tlačítko Apply v menu modifikátorů (tím se zruší původní geometrie tělesa a převede se na tvar vzniklý zapnutím modifikátoru). V editačním módu přepneme na výběr ploch (Face Selection) a označíme tvar, který zhruba odpovídá průřezu ramene. Extrudujeme ho do zvolené délky a následně upravíme jeho rozměr v normálové ose na 0 (z válcové plochy dostaneme rovinnou). Označíme spodní část profilu ramene a tažením vzhůru vytvoříme skosený bokorys ramene. Následně vytvoříme odlehčovací otvory v rameni a díry pro šrouby, stejně jako zde již bylo vícekrát uvedeno (válec+Boolean). Nakonec nastavíme modifikátor Mirror a konečný tvar ramene je hotov.

Kolejnici vedení vytvoříme separací (funkce Separate) vertexů v jednom zvoleném rohu nosného ramene a jejich následnou extrudací do požadovaného tvaru. Abychom nemuseli kvůli vytvoření ostatních kolejníc několikrát duplikovat a nastavovat polohu, využijeme modifikátor Mirror. Horní vedení vytvoříme roztažením základního válce a jeho uchycení vymodelujeme již známými postupy ze základní krychle. Vytvoření tělesa pojezdu je poměrně triviální záležitost a nebudeme ji zde podrobně popisovat. Za zmínku stojí už jen řídicí kolo mechanismu pojezdu, které je tvořeno základním objektem Torus; jeho ramena pak vytažena ze základní krychle a aplikováním funkce Spin. Z příslušenství kolem obráběcího nástroje popíšeme trysku na chladicí kapalinu, která vznikla extrudováním ze základní kružnice a opět aplikací funkce Spin. Jiná možnost pro její tvorbu by byla například modelování tažením profilu po křivce, kde by bylo dosaženo jemnějšího tvaru; nicméně vzhledem k zanedbatelnosti rozměru této součásti vzhledem k celému modelu byla zvolena metoda první, která je méně náročná na výpočet (Při převodu křivky na Mesh by objekt obsahoval mnohem více vertexů). Kolo pro nastavení polohy obráběcího nástroje bylo do tohoto modelu importováno z modelu soustruhu, který bude popsán níže. Šrouby opět vložíme z knihovny základních objektů.

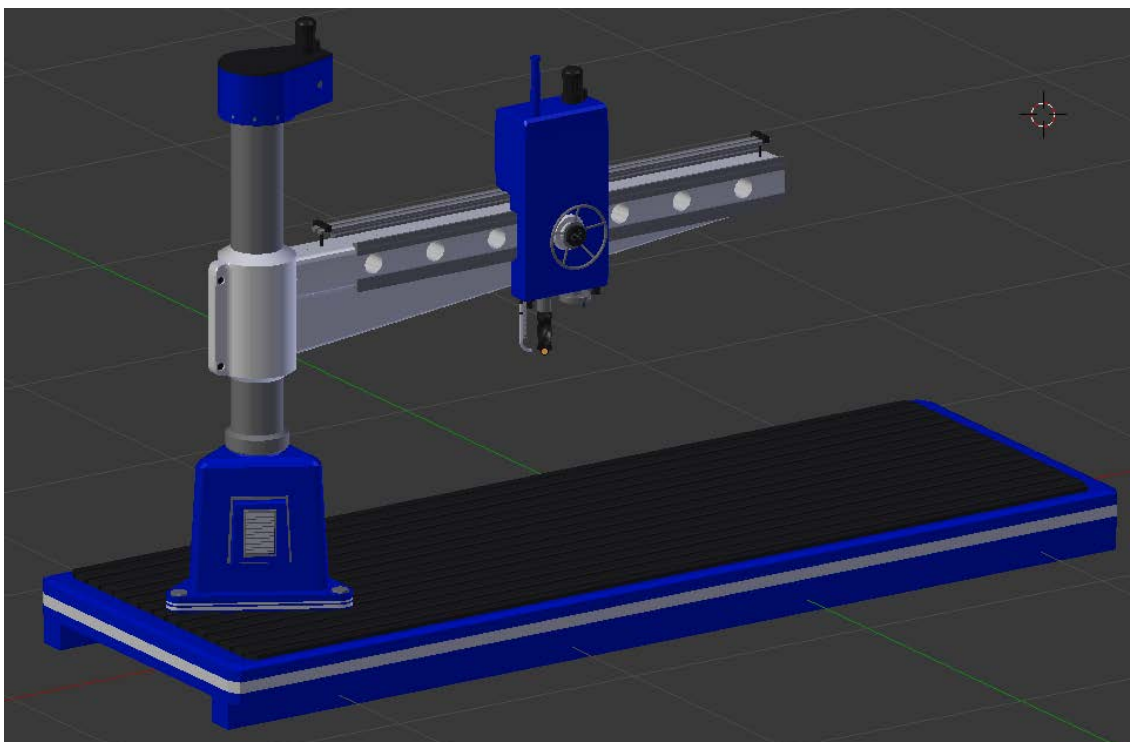
Tvorba obráběcího nástroje (zde zvolen vrták) je opět poněkud jiná než u ostatních nástrojů, popíšeme ji zde proto podrobněji. Začneme vložením Bezierova kruhu, což je uzavřená varianta Bezierových křivek. V Edit Mode upravíme tvar do požadovaného průřezového profilu těla vrtáku (pro jednoduchost může být zvolen obvod zploštělé osmičky), následně aplikujeme modifikátor Screw. Základně je výška požadované šroubovice nastavena na nulový rozměr, pro dosažení správného tvaru je nutné zvýšit parametr Screw na

požadovanou hodnotu. Pro další práci na objektu je nutné převést křivku na objekt typu Mesh. To provedeme spuštěním funkce konvertování (Alt+C) a zvolením položky MeshfromCurve. Špičku vrtáku zhotovíme pouze označením spodní hrany výsledného tvaru a jejím zmenšením na rozměr 0 (je následně vhodné odstranit překrývající se vertexy – funkcí Remove Doubles). Stopku vrtáku musíme vytvořit vložení základního válce (v Edit Mode!), u kterého nastavíme stejný počet vertexů jako u horní hrany šroubovice, následně jej umísíme mírně nad šroubovici. Následně označíme spodní hranu válce a horní hranu šroubovice a aplikujeme skript BridgeEdgeLoops (mezerník → BridgeEdgeLoops), který provede vyplnění a spojení zvolených vertexů plochami [42]. Je třeba si povšimnout faktu, že takto vymodelovaný obráběcí nástroj by nebyl pro strojařské praxe vhodný z hlediska geometrie břitů apod. Nicméně cílem této práce není zabývat se navrhováním obráběcí techniky a z hlediska grafické reprezentace strojního zařízení je tento postup zcela dostatečný.

Jako materiály vzhledu použijeme opět základní barvy. Pruhy provedeme stejnou technikou jako u modelu popsaném výše.

Před importováním bylo nutné, zejména u objektu reprezentujícího obráběcí nástroj, provést rekalkulaci normál na vnější směr. Bez tohoto postupu by se mohli po importu jevit některá místa na součástech jako průhledná, či neviditelná. Směr normál je totiž základně stanovován automatickým výpočtem během modelování a občas (hlavně při aplikaci složitějších modifikátorů) se stane, že některé plochy jsou vzhledem ke zbytku modelu inverzní, což má na jejich vzhled negativní dopad.

Výsledek modelování je vidět na obrázku 20.



Obrázek 20-frézka

4.1.3 Dopravníkový pás

Dopravníkový pás je také velice často využívaným vybavením továren, byl proto zařazen i do této knihovny modelů. Je zřejmé, že tvorba většiny jeho částí bude po vysvětlení postupů použitých u výše zmíněných modelů poměrně jednoduchou záležitostí. Tvorba konečného tvaru pásoviny ovšem nemusí být vždy zcela triviální.

Pozornost je třeba věnovat zejména rozměrům válcových ploch, na které dosedá pás. Přesnosti uložení pohybových válců dosáhneme opět aplikováním funkce Separate. Tvar pásu je pak nutné přizpůsobit trase, kterou bude výsledný model vykonávat. K tomu využijeme s výhodou modifikování podle Beziérových křivek.

Postup tvorby

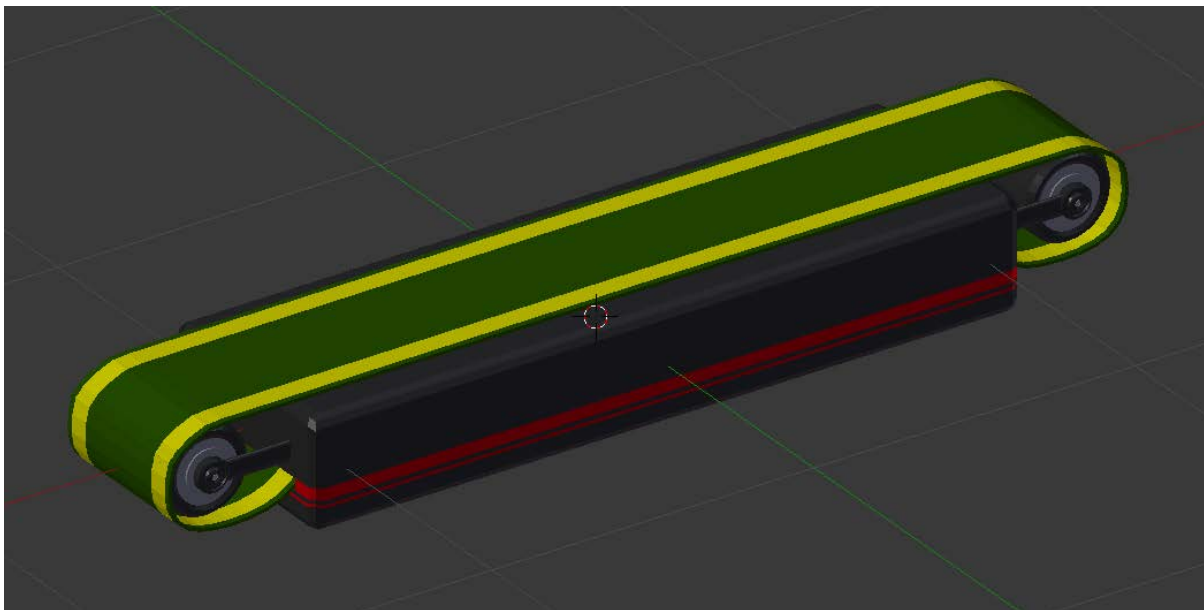
Zmíníme zde pouze ty prvky, u nichž byly použity postupy, které se nevyskytovaly v předchozích kapitolách. Nejprve tedy popíšeme vznik pásu; předpokládáme, že tělo dopravníku, včetně pohybových válců již byly dokončeny (využito bylo pouze metody Extrude a aplikování některých modifikátorů).

Tvořícím prvkem „pásoviny“ bude základní krychle, jejíž dimenze upravíme do hlavních rozměrů pásu – tzn. požadovaná šířka a výška; délku nastavíme v porovnání se šířkou poměrně menší (bude ovlivňovat zejména kvalitu zobrazení v místech ohybu pásu – obecně platí, že čím kratší délka, tím větší je kvalita). Následně vložíme základní Beziérovu křivku (Shift+A→Curve→BezierCurve), nastavíme jí v menu vlastností pouze 2 rozměrné chování a nakonec její tvar vymodelujeme (opět Extrude) tak, aby reprezentoval trasu, po které se bude pás pohybovat (krajní body spojíme – klávesa F). Soustředíme se na správné dosednutí na válcové plochy (většinou stačí správně nastavit a upravovat polohu, škálu a rotaci kontrolních bodů) a přesný průchod otvorem v tělese dopravníku. Protože je v tomto případě trasa symetrická, postačí vymodelovat jednu stranu, poté zapnout modifikátor Mirror a ten následně sjednotit s původním tvarem (tlačítko Apply). Tvar pásu dostaneme aplikováním modifikátoru Array na výše zmiňovanou upravenou základní krychli. Změníme nastavení položky Fit Type na Fit Curve a jako parametr nastavíme námi již definovaný tvar pásu (Bezierova křivka). Tím se automaticky vypočte počet prvků pole tak, aby vyplnily délku křivky bez mezery nebo zbytečného přesahu. Dále nastavíme na upravenou krychli modifikátor Curve a jako parametr opět nastavíme objekt křivky, představující cestu pásu.

Jako materiály vzhledu použijeme opět základní barvy. Pruhy provedeme stejnou technikou jako v modelech popsaných výše.

Při takto zvoleném postupu modelování nevznikly při importování žádné komplikace. Bylo pouze nutné u některých ploch provést rekalkulaci směru normály do požadovaného směru.

Výsledek modelování je vidět na obrázku 21.



Obrázek 21-Pásový dopravník

4.1.4 Ovládací panel

Pro účely ovládání některých funkcí modelů, které budou popsány v následující kapitole, byl vytvořen univerzální ovládací panel. Obsahuje sady různých tlačítek, jejichž počet a poloha půjde měnit podle požadavků pro konkrétní typ stroje. Umístěny zde budou přepínače typu zapnout/vypnout, nouzový vypínač, regulátor řízení otáček/rychlosti a spínač pro výběr různých poloh. Na přesnou návaznost rozměrů zde není brán takový zřetel jako u ostatních modelů. Nicméně pro vytvoření požadovaného vzhledu jednotlivých tlačítek bude nutné správným způsobem aplikovat metodu UV mapování.

Tvar základní desky i tlačítek vymodelujeme použitím postupů, podobných postupům popsaným výše, dále se budeme věnovat pouze vytvoření obrázkových textur a správnému „rozbalení“ geometrie tělesa při UV mapování.

Při troše aproximace lze na většinu zde použitých tvarů pohlížet jako na tělesa se dvěma podstavami (i když ne vždy rovinnými) a pláštěm a tomu pak přizpůsobit výběr hran, které budou následně představovat švy pro rozbalení povrchu těles. Označujeme tedy (v Edit Mode) vždy obvody horní a spodní podstavy a jednu hranu, která je spojuje (podobně jako bychom například rozřezávali povrch válce). Když jsou hrany vhodně vybrány, nastavíme jim v postranním menu na kartě Shading/UVs vlastnost Mark Seam, čímž je označíme jako švy (projeví se to změnou barvy). Dále vybereme všechny vertexy daného objektu a v té samé kartě vybereme v rozbalovacím menu Unwrap, čímž provedeme doslova rozbalení povrchu tělesa podle dříve definovaných švů. Výsledek Unwrapu se zobrazí po přepnutí do UV/Image editoru. V UV/Image editoru vybereme předem připravenou texturu (zde použity pouze základní nápisy a tvary vytvořené v programu Malování, nebo stažené z volně dostupných zdrojů) a následně upravíme polohu rozbalených vertexů tak, aby umístění obrázku (např. textu) odpovídalo požadovanému vzhledu tlačítka. Při této technice texturování je dobré mít pracovní plochu rozdělenou do více oken a v jednom mít UV/Image editor a ve druhém 3D view se zapnutým zobrazováním textur. Z rozbaleného povrchu nemusí být například vždy patrné, která podstava je horní a která spodní; využitím více oken se pak správné umístění

textury stává celkem snadno proveditelnou záležitostí. V novějších verzích Blenderu již pak ani není nutné obrázek na objekt „nabalovat zpět“; změny textury jsou patrné okamžitě.

Jediný problém při takto zvoleném nastavování vzhledu vzniká při importování do Unity, kdy nedochází k přenesení zvolených obrázků, které je nutné samostatně vložit do složky s materiály v knihovně assetů a následně materiály manuálně přiřadit jednotlivým objektům.

Výsledek modelování je vidět na obrázku 22 (při pohledu shora).



Obrázek 22-Ovládací panel

4.1.5 Ostatní modely

Postupy, kterými vznikaly ostatní modely obsažené ve vytvářené knihovně, by nemělo smysl zde podrobně popisovat, protože by se od výše zmíněného lišili jen v nepatrných detailech. Důvody, proč zde byli umístěny, jsou také velmi podobné výše zmiňovaným. Jejich seznam a podobu lze sledovat například v prezentaci přiložené k BP nebo nejlépe v knihovně souborů, která je součástí příloh k bakalářské práci (viz CD ROM). Byly také použity modely třetích stran, stažené ze stránek <https://3dwarehouse.sketchup.com>.

5 Tvorba a popis interakcí

V této kapitole popíšeme části programového kódu, které byly vytvořeny za účelem „rozpohybování“ a další funkčnosti některých částí vytvářených modelů. K programování byl využit jazyk C sharp a vývojové prostředí Monodevelop (vše kompatibilní s programem Unity). Jedná se o poměrně nekomplikované skripty, sloužící pouze pro ilustraci některých základních pohybů (např. rotace obráběcích nástrojů) a funkčních vztahů mezi jednotlivými komponenty (např. tlačítka ovládacího panelu). Cílem této práce není vytvořit komplexní sadu programů pro ovládání vytvořených modelů.

Funkce strojů budou ve většině případů spouštěny pomocí objektů zvaných Collider, které budou podrobněji popsány níže. Spuštění bude probíhat buď kliknutím myši, nebo v jiných případech vstoupením FPS ovladače do určité zóny (také vymezené Colliderem). Objekty typu Collider budou většinou umístěny na ovládacím panelu popsaném v kapitole 4.1.4. V některých případech bude zapotřebí vytvořit některé doplňkové funkce jako například pohyb tlačítek panelu apod. Vše bude uvedeno v následujících podkapitolách.

Skripty v Unity se uloží do knihovny Assetů a pracuje se s nimi podobně jako s ostatními modely (více viz kapitola 3). Pro správnou funkci stačí, tažením myši, jednotlivé skripty přiřadit požadovaným modelům a následně vhodně nastavit vstupní parametry (bude popsáno níže). Pro lepší pochopení vysvětlené funkčnosti skriptů, je nutné nahlédnout do zdrojových kódů obsažených v příložené knihovně Assetů. Pokud by čtenáři nebyl jasný způsob fungování jednotlivých volaných funkcí, nechť si je prostuduje například na webu unity3d.com.

Nakonec je dobré zmínit, že uvedené skripty neobsahují ošetření proti zadávání nekorektních hodnot, je tedy nutné dbát na správný postup při jejich aplikování. Nesprávné vstupní parametry ovšem nebudou mít v zásadě negativní dopad. Jediné, k čemu může dojít je, že se po aplikaci skriptu nestane nic, nebo například požadovaný pohyb bude probíhat v jiném směru.

5.1 Algoritmus rotace nástrojů

Vstupní parametry: desetinné číslo: **speed**, Game Object: **GO**, textový řetězec: **osa**

Výstup: Rotace podle příslušné osy, příslušnou rychlostí

Tento skript (v příložené knihovně pojmenovaný **Toceni.cs**) byl vytvořen pro popis rotace obráběcích nástrojů, upnutí soustruhu a některých dalších komponent. Veřejně přístupné proměnné (public) zde slouží jako vstupní parametry. Nastavit lze rychlost otáčení (parametr speed), nastavit která část modelu má rotovat (parametr GO) a následně směr a orientaci rotace (parametr osa – např. [y] = rotace podle lokální osy y proti směru hodinových ručiček a [-y] po směru).

V jednotlivých instancích kódu pak dochází k porovnání požadovaných parametrů se vstupními hodnotami a při shodě je inicializována funkce **transform.Rotate** jejímiž parametry jsou příslušné směry a orientace rotace (vyjádřené pomocí funkce **Vector3**) a počet otáček, který je vyjádřen voláním funkce **Time.deltaTime** (zjednodušeně otáčky = rychlost*čas). Pro bližší pochopení je nutné nahlédnout do přiložené knihovny a přečíst si příslušný zdrojový kód.

5.2 Algoritmus světelného spínače

Vstupní parametry: pole objektů (světla): **svetla**, GameObject: **initial**, desetinné číslo: **z**, desetinné číslo: **porovnat**, celé číslo **otocit**

Výstup: rozsvícení/zhasnutí světla a pohyb spínače

Tento skript (**LightSwich2.cs**) byl vytvořen pro ovládání osvětlení. Veřejně přístupné proměnné (**public**) zde slouží jako vstupní parametry. Nastavit lze počet a typ ovládaných světel (parametr **svetla**), objekt reprezentující spínač (**initial**) a o kolik stupnu se má zvolený spínač pootočit (**otocit**).

Po inicializaci všech objektů scény, se do proměnných **z** a **porovnat** přiřadí souřadnice reprezentující rotaci kolem osy **z**, objektu reprezentujícího spínač. Pokud je na spínač kliknuto myší, rozsvítí se nebo zhasnou světla a vypínač se přemístí do příslušné pozice. Pro bližší pochopení je nutné nahlédnout do přiložené knihovny a přečíst si příslušný zdrojový kód.

5.3 Algoritmus viditelného kurzoru myši (**visibleCursor.cs.cs**)

Skript bez vstupních parametrů, jehož výsledkem je zobrazení kurzoru myši. Kurzor je následně využíván pro ovládání funkcí programu a orientaci v prostoru. Byl přiřazen objektu reprezentujícímu ovladač první osoby (**FPS ovladač** – pospán níže). V souvislosti s tímto skriptem byla vytvořena textura, která se bude zobrazovat jako kurzor. Bez takto zvolené textury, by se po aplikování skriptu, zobrazoval pouze základní kurzor šipky, který je součástí operačního systému počítače. Pro bližší pochopení je nutné nahlédnout do přiložené knihovny a přečíst si příslušný zdrojový kód.

5.4 Algoritmus restartování levelu (**restart.cs**)

Tento skript je bez vstupních parametrů a je určen pro uvedení programu do počátečních podmínek. Pokud je kdykoli v průběhu spuštěného programu stisknuta klávesa **r**, dojde k resetování ukázkového prostředí do počátku. Pro bližší pochopení je nutné nahlédnout do přiložené knihovny a přečíst si příslušný zdrojový kód.

5.5 Algoritmus ukončení programu (**escapeQuit.cs**)

Tento skript je bez vstupních parametrů a zde je určen pro přiřazení k ovladači z první osoby (**FPS** – first person controller). Pokud provedeme **build** (sestavení spustitelného programu) testované scény, je díky tomuto skriptu možné takovouto aplikaci ukončit stisknutím klávesy **escape**. Pro bližší pochopení je nutné nahlédnout do přiložené knihovny a přečíst si příslušný zdrojový kód.

5.6 Algoritmus povolení/zakázání komponentu (MD_dis.cs)

Vstupní parametry: GameObject: **GO**

Výstup: povolení nebo zakázání určitého parametru

Skript použitý pro aktivaci nebo deaktivaci rotace nástroje po stisknutí tlačítka myši. Pro bližší pochopení je nutné nahlédnout do přiložené knihovny a přečíst si příslušný zdrojový kód.

5.7 Ukázkové prostředí a ovladač FPS

Knihovna také obsahuje ukázkovou aplikaci (vytvořenou pomocí Unity 3D), která ilustruje možné využití knihovny modelů v praxi. Jedná se o hrubý návrh rozmístění strojů ve výrobní hale. Mezi stroji je možné se pohybovat za využití FPS ovladače (pohled první osoby, ovládání myši + WSAD), který je součástí základních assetů vývojového prostředí Unity 3D. Pro správné chování z hlediska fyzikálních zákonů je nutné, aby FPS ovladači nebyl umožněn průchod jednotlivými stroji nebo například stěnami. Tato vlastnost se nastavuje využitím tzv. kolizních objektů (Collider), které při správné aplikaci programu sdělují informace o „nepřechodnosti“ určitých míst. Zprovoznění se provádí nejčastěji jako vložení nového komponentu zvoleného objektu (např. menu Component→Physics→Box Collider). Kolizní objekty mohou být různě složité; mohou mít pouze tvar krychle opsané základnímu objektu (Box Collider), nebo mohou tvar zvoleného objektu detailně „opisovat“ (Mesh Collider), ovšem za předpokladu vyšších nároků na výpočetní kapacitu počítače. V této práci byly využity pouze kolizní objekty typu Box Collider, z důvodu snížení výpočetní složitosti. Pokud by bylo v budoucnosti nutné, například z důvodu komplexnějšího využití modelů, využít detailnější kolizní objekty, současné uspořádání by nemělo nijak překážet případné úpravě nebo změně (uživatel knihovny může jednoduše smazat komponent Box Collider a využít některou z jiných možností).

Jak již bylo uvedeno výše, kolizní objekty mohou sloužit i jako spouštěče různých funkcí programu. Postup přiřazení **Collideru** k jednotlivým objektům je stejný, pouze se musí v menu **Components** zvolit možnost **IsTrigger**. Pro bližší pochopení je nutné nahlédnout do přiložené knihovny a přečíst si příslušný zdrojový kód.

Pro usnadnění orientace v ukázkovém prostředí byl vytvořen dokument s návodem, který je nahrán na přiloženém CD.

5.8 Algoritmus lineární trajektorie

Pro pohyb vysokozdvížných vozíků, byly využity algoritmy TrajectoryLinear.cs a FireTraj.cs. Je nejprve zapotřebí definovat body vykonávané trasy, specifikovat objekt, který se bude po trase pohybovat a trajektorii mu přiřadit. Pro bližší pochopení je nutné nahlédnout do přiložené knihovny a přečíst si příslušný zdrojový kód. Tvůrcem obou algoritmů je ing. Jiří Polcar, který je zároveň odborným konzultantem této práce.

5.9 Animace

V této podkapitole budou popsány skripty, které slouží k práci s animacemi. V této souvislosti bylo v programu Unity vytvořeno několik animací, které budou ukazovat některé pohyby strojů a ovládacích prvků. Animace se vytváří pomocí klíčových snímků (tzv. key frames) a následně se přiřazují jednotlivým objektům a jejich přehrávání může probíhat automaticky nebo pomocí skriptů. Z časových důvodů a kvůli redukování rozsahu práce zde postup jejich tvorby nebude podrobněji popsán. Pro bližší pochopení problematiky viz příložené zdroje např. [46]. Podrobný seznam animací lze sledovat v příložené knihovně assetů, nebo v ukázkové aplikaci.

5.9.1 Spuštění animace přiblížením nebo vzdálením FPS ovladače (fireAnim_enterExit.cs)

Vstupní parametry: GameObject: **GO**, textový řetězec **prvni**, textový řetězec **druha**

Výstup: Spuštění animace přiblížením nebo vzdálením FPS ovladače

Pokud FPS ovladačem vejde nebo vyjde ze zóny vyhrazené colliderem, který slouží jako spouštěč skriptu, spustí se animace. Tento skript byl využit např. pro otevírání a zavírání dveří. Veřejně přístupné proměnné slouží pro zadání názvů soustředěných animací. Parametr GO slouží pro reprezentaci objektu, ke kterému jsou přiřazeny spouštěné animace. Pro bližší pochopení je nutné nahlédnout do příložené knihovny a přečíst si příslušný zdrojový kód

5.9.2 Spuštění animace stisknutím myši (fireAnim1.cs)

Vstupní parametry: GameObject: **GO**, boolean typ **nahore**, textový řetězec **první**, **druha**

Výstup: Spuštění animace kliknutím myši

Funguje podobně jako skript v kapitole 5.7.1, pouze s rozdílným spouštěcím mechanismem.

6 Závěr

V této práci byly zpracovány tyto informace:

- Byly zde zmíněny některé základní pojmy, které se týkají problematiky digitálního podniku a digitální továrny.
- Byl zde popsán současný stav zobrazovacích technologií, využívaných v rámci konceptu virtuální reality a několik projektů zabývajících se problematikou virtuální továrny.
- Byly analyzovány a popsány možnosti a nástroje zvolených vývojových prostředí Unity 3D a Blender 3D.
- Bylo vytvořeno (v programu Blender) a následně popsáno několik modelů strojů a inventárního vybavení, které bude možné využít pro tvorbu interaktivních virtuálních prostředí v rámci konceptu digitální továrny.
- V prostředí Monodevelop (Unity 3D) byly vytvořeny programové kódy pro ovládání funkcí některých výše zmíněných modelů.
- Byl vytvořen ukázkový program, který ilustruje možnost využití knihovny modelů. Pro usnadnění orientace v ukázkovém prostředí byl vytvořen dokument s návodem.

Výsledkem práce je tedy zejména ukázka možností SW využitého pro modelování a přípravu virtuálního prostředí. Program Blender je dostupný pod volnou licenci, a přesto jsou dosahované výsledky, ať už po stránce vizuální nebo technické (např. návaznost, modularita a přesnost jednotlivých součástí) poměrně kvalitní a v některých případech minimálně srovnatelné s komerčně nabízenými programy. Vývojové prostředí Unity také umožňuje dosáhnout značně kvalitního vzhledu i funkčnosti a to i v případě využití verze programu nabízené jako freeware. Vhodnost využití těchto nástrojů v projektech, zabývajících se problematikou digitální továrny, byla vyhodnocena jako vysoká.

Z hlediska technologií je pro tento a podobné projekty předpokládáno, postupné zvyšování provázanosti s projekčními zařízeními maximalizujícími stupeň imerze. I když v této práci nebyl kladen přímý důraz na tyto technologie, existuje určitá pravděpodobnost, že pokud by v budoucnosti došlo k návazné práci na tento projekt (např. diplomová práce), byla by detailněji zpracována i problematika přímého propojení s výpočetní a zobrazovací technikou.

Zdroje

- [1] LAVIERI, E. *GettingStartedwith UNITY 5*. 1. Birmingham: PacktPublishing Ltd., 2015. ISBN 978-1-78439-831-6.
- [2] BENDOVIÁ, Helena: Imerze, Game Art 2013. Dostupné online:
<<http://cas.famu.cz/gameart/page.php?page=7>> [cit. 13. 10. 2015].
- [3] PSOHLAVEC, Stanislav: Digitalizace – co tím myslíte, Ikaros. Dostupné online:
<<http://ikaros.cz/digitalizace-%E2%80%93-co-tim-myslite>> [cit. 13. 10. 2015].
- [4] HOŘEJŠÍ, P., T. GÖRNER a O. KURKIN. *Virtuální realita : základní úroveň, e-book* [online]. 1. Plzeň: ZČU, 2012 [cit. 2015-12-13]. ISBN 978-80-87539-07. Dostupné z:
<<https://portal.zcu.cz/portal/studium/courseware/kpv/dpvr/stazeni.html>>
- [5] LEEDER, E. a M. BUREŠ. *Digitální podnik* [online]. 1. Plzeň: ZČU, 2012 [cit. 2015-12-13]. Dostupné z: <<https://portal.zcu.cz/portal/studium/courseware/kpv/dpvr/stazeni.html>>
- [6] TOMÁNEK, Tomáš: Imerze, Nová média, web studentů SNM. Dostupné online:
<<http://novamedia.xf.cz/haptic.html>> [cit. 15. 10. 2015].
- [7] POLCAR, Jiří. *BAKALÁŘSKÁ PRÁCE: Využití Source Engine pro vizualizaci a interakci v prostředí digitální továrny a tvorba studijních podkladů* [online]. Plzeň, 2012 [cit. 2015-12-13]. Dostupné z:< <https://otik.uk.zcu.cz/bitstream/handle/11025/4158/BP.pdf?sequence=1>>. ZČU v Plzni. Vedoucí práce Petr HOŘEJŠÍ.
- [8] *Microsoft.com* [online]. 2015 [cit. 2015-12-13]. Dostupné z:
<<https://www.microsoft.com/microsoft-hololens/en-us>>
- [9] *Microsoft.com* [online]. 2015 [cit. 2015-12-13]. Dostupné z:
<<https://www.microsoft.com/microsoft-hololens/en-us/commercial#autodesk>>
- [10] *Microsoft.com* [online]. 2015 [cit. 2015-12-13]. Dostupné z:
<<https://www.microsoft.com/microsoft-hololens/en-us/experience>>
- [11] *Microsoft.com* [online]. 2015 [cit. 2015-12-13]. Dostupné z:
<<https://www.microsoft.com/microsoft-hololens/en-us/hardware>>
- [12] *Jet Propulsion Laboratory: California Institute of Technology* [online]. 2015 [cit. 2015-12-13].
<Dostupné z: <http://www.jpl.nasa.gov/news/news.php?feature=4451>>
- [13] *Google Developers* [online]. 2015 [cit. 2015-12-13]. Dostupné z:
<<https://developers.google.com/glass/develop/gdk/quick-start>>
- [14] *Wikipedia, the free encyclopedia* [online]. 2015 [cit. 2015-12-13]. Dostupné z:
<https://en.wikipedia.org/wiki/Oculus-Rift>
- [15] POLCAR, Jiří. *NEKONVENČNÍ NÁSTROJ K VIZUALIZACI A INTE RAKCI VE VIRTUÁLNÍM PODNIKU SV OČ – FST 2014* [online]. : 7 [cit. 2015-12-13]. Dostupné z: <<http://old.fst.zcu.cz/-files-web-FST/-SP-FST%28SVOC%29/-2014/-sbornik/PapersPdf/Mgr/Polcar-J.pdf>>
- [16] Blender Reference Manual. *Blender.org* [online]. 2015 [cit. 2015-12-13]. Dostupné z:
<<https://www.blender.org/manual/>>
- [17] Blender Reference Manual. *Blender.org* [online]. 2015 [cit. 2015-12-13]. Dostupné z:
<<https://www.blender.org/manual/interface/introduction.html>>

- [18]Vrcholy, hrany a stěny. *Wiki.Blender.org* [online]. 2013 [cit. 2015-12-13]. Dostupné z: <<http://wiki.blender.org/index.php/Doc:CZ/2.6/Manual/Modeling/Meshes/Mesh-Structures>>
- [19]Metaobjekty. *Wiki.Blender.org* [online]. 2013 [cit. 2015-12-13]. Dostupné z: <<http://wiki.blender.org/index.php/Doc:CZ/2.6/Manual/Modeling/Metas>>
- [20]BlenderDressCloth Test [obrázek]. *Joszuk.files.wordpress.com* [online]. [cit. 2015-12-13]. Dostupné z: <https://joszuk.files.wordpress.com/2013/03/dress-blender.jpg>
- [21]1936-Fluid-Simulation---Tutorial-1 [obrázek]. *Cgtutorials.com* [online]. [cit. 2015-12-13]. Dostupné z: <http://www.cgtutorials.com/oneadmin/-files/linksdire/1936-Fluid-Simulation---Tutorial-1--Very-Basic-Introduction-.png>
- [22]POLCAR, Jiří. *Využití balíku DIGITOV pro praktické účely* [online]. ZČU, Plzeň, 2014 [cit. 2016-03-26]. Dostupné z: <https://otik.uk.zcu.cz/bitstream/handle/11025/12270/DP-Jiri-Polcar.pdf?sequence=1>
- [23]Vuzix: ViewTheFuture. *Www.vuzix.com* [online]. [cit. 2016-03-26]. Dostupné z: <https://www.vuzix.com/Products>
- [24]AUTOR NEUVEDEN[obrázek]. *webnode.cz/* [online]. [cit. 26.3.2016]. Dostupný na WWW: <http://files.zskaznejov.webnode.cz/200017615-291e729feb/3D%20br%C3%BDle%20-%20pap%C3%ADrov%C3%A9.jpg>
- [25]Unity [obrázek]. *Assetstore.unity3d.com* [online]. [cit. 2016-03-26]. Dostupné z: <https://www.assetstore.unity3d.com/en/#!/content/46209>
- [26]Unity [obrázek]. *Assetstore.unity3d.com* [online]. [cit. 2016-03-26]. Dostupné z: <https://www.assetstore.unity3d.com/en/#!/content/51519>
- [27]Animech Technologies. *Animechtechnologies.com* [online]. [cit. 2016-03-26]. Dostupné z: <http://www.animechtechnologies.com/showcase/virtual-car-2/>
- [28]Unity. *Unity3d.com* [online]. [cit. 2016-03-26]. Dostupné z: <https://unity3d.com/showcase/gallery/non-games>
- [29]ClintHannaFord - Portfolio. *Pleribus.com* [online]. [cit. 2016-03-26]. Dostupné z: <http://www.pleribus.com/portfolio-item/ferrari-augmented-reality-ipad-app/>
- [30]Unity - Showcase - Gallery. *Unity3d.com* [online]. [cit. 2016-03-28]. Dostupné z: <https://unity3d.com/showcase/gallery/non-games>
- [31]Oshkosh HEMTT VirtualTraining. *Http://hwd3d.com* [online]. [cit. 2016-03-28]. Dostupné z: <http://hwd3d.com/vimeo-video/oshkosh-hemtt-virtual-training/>
- [32]Gunstruction. *Gunstruction.net* [online]. [cit. 2016-03-28]. Dostupné z: <http://www.gunstruction.net/news/>
- [33]Using Meta Objects. *Safaribooksonline.com* [online]. [cit. 2016-03-28]. Dostupné z: <https://www.safaribooksonline.com/library/view/blender-for-dummies/9780470584460/ch006-sec010.html>
- [34]UV unwrap. *Goanna.cs.rmit.edu.au* [online]. [cit. 2016-03-28]. Dostupné z: <http://goanna.cs.rmit.edu.au/~gl/teaching/Interactive3D/2012/images/uv-unwrap.jpg>
- [35]Do ElephantsDreamofClockworkAndroids? *Farlane.com* [online]. [cit. 2016-03-28]. Dostupné z: <http://farlane.com/2006/05/23/do-elephants-dream-of-clockwork-androids/>

- [36]SfeMovie Project [obrázek]. *Sfemovie.yalir.org* [online]. [cit. 2016-03-28]. Dostupné z: <http://sfemovie.yalir.org/1.0/screenshots/big-buck-bunny1.jpg>
- [37]27 InspiringBlenderAnimationsThatWill Make Your Jaw Drop. *Blenderguru.com* [online]. [cit. 2016-03-28]. Dostupné z: <http://www.blenderguru.com/articles/27-inspiring-blender-animations-that-will-make-your-jaw-drop/>
- [38]BMW 3 PromoBlender [obrázek]. *Http://1080.plus* [online]. [cit. 2016-03-28]. Dostupné z: <http://i4.ytimg.com/vi/J5Xf7XrNWnA/maxresdefault.jpg>
- [39]Art Gallery [obrázek]. *Blender.org* [online]. [cit. 2016-03-28]. Dostupné z: <http://archive.blender.org/uploads/pics/3D-Watch-Olivier-Amrein-December2007.jpg>
- [40]BikeConfig. *Http://www.bikeconfig.com/* [online]. [cit. 2016-03-28]. Dostupné z: <http://www.bikeconfig.com/>
- [41]<http://answers.unity3d.com/questions/34572/blender-mesh-import-issue.html>
- [42]CG-Tutorials. Blender 3D tutorial - How to model a drill bit (BezierCircle, Screw, Bridge). In: *Youtube* [online]. Zveřejněno 24. 12. 2014 [vid. 2016-4-6]. Dostupné z: <https://www.youtube.com/watch?v=bIRY60eeNYU>
- [43]Informace o divizi DF&PD. *Siemens.cz* [online]. [cit. 2017-03-19]. Dostupné z: <http://www1.siemens.cz/ad/current/index.php?ctxnh=635babf0f2&ctxp=home>
- [44]Systémy pro ovládání a vizualizaci. *Siemens.cz* [online]. [cit. 2017-03-19]. Dostupné z: <http://www1.siemens.cz/ad/current/index.php?vw=0&ctxnh=c9f2e370df&ctxp=home>
- [45]SIRIUS Industrial Controls. *Siemens.cz* [online]. [cit. 2017-03-19]. Dostupné z: <http://www1.siemens.cz/ad/current/content/data-files/katalogy/ic10/cat-ic-10-2017-en.pdf>
- [46]Unity tutorials. *Unity3d.com* [online]. [cit. 2017-05-23]. Dostupné z: <https://unity3d.com/learn/tutorials/topics/animation>