

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ

Studijní program: B0715A270013 – Strojní inženýrství
Studijní specializace: Průmyslové inženýrství a management

BAKALÁŘSKÁ PRÁCE

**Tvorba univerzálního prostředí ve virtuální realitě pro interakci
s průmyslovými 3D modely**

Autor: Filip KRÁL
Vedoucí práce: Ing. Matěj Dvořák

Akademický rok 2023/2024

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta strojní

Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Filip KRÁL**
Osobní číslo: **S20B0172P**
Studijní program: **B0715A270013 Strojní inženýrství**
Specializace: **Průmyslové inženýrství a management**
Téma práce: **Tvorba univerzálního prostředí ve virtuální realitě pro interakci s průmyslovými 3D modely**
Zadávající katedra: **Katedra průmyslového inženýrství a managementu**

Zásady pro vypracování

- Úvod
- Analýza současných technologických možností
- Analýza možností implementace 3D modelů do VR
- Rešerše dostupných softwarů pro vývoj a tvorbu modelů
- Popis realizace a návrh vlastní implementace
- Zhodnocení a závěr

Rozsah bakalářské práce: **40 až 50 stran**
Rozsah grafických prací: **–**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. Linowes, J., Unity 2020 Virtual Reality Projects: Learn VR development by building immersive applications and games with Unity 2019.4 and later versions, 3rd Edition, Packt Publishing 2020, ISBN 978-1839217333
2. Sung, K., Smith, G., Basic Math for Game Development with Unity 3D: A Beginner's Guide to Mathematical Foundations, Apress 2019, ISBN 978-1484254424
3. Okita, A., Learning C# Programming with Unity 3D, second edition, Routledge 2019, ISBN-13: 978-1138336810
4. LaValle, S. M., Virtual Reality, Cambridge University Press 2023, dostupné online na <http://lavalle.pl/vr/>
5. Oficiální Unity3D návody dostupné na <https://learn.unity.com/>

Vedoucí bakalářské práce: **Ing. Matěj Dvořák**
Katedra průmyslového inženýrství a managementu

Konzultant bakalářské práce: **Ing. Miroslav Bednář**
Katedra průmyslového inženýrství a managementu

Datum zadání bakalářské práce: **16. října 2023**
Termín odevzdání bakalářské práce: **24. května 2024**

L.S.

Doc. Ing. Vladimír Duchek, Ph.D.
děkan

Doc. Ing. Michal Šimon, Ph.D.
vedoucí katedry

Prohlášení o autorství

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

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

V Plzni dne:

.....

podpis autora

Poděkování

Rád bych tímto vyjádřil svou vděčnost vedoucímu mé bakalářské práce panu Ing. Matějovi Dvořákovi. Především za jeho rady, trpělivost a čas, který mi věnoval.

Také chci poděkovat mojí matce, která mě po celý průběh studia podporovala.

ANOTAČNÍ LIST BAKALÁŘSKÉ PRÁCE

AUTOR	Příjmení Král	Jméno Filip	
STUDIJNÍ PROGRAM	B0715A270013 Strojní inženýrství		
VEDOUCÍ PRÁCE	Příjmení (včetně titulů) Ing. Dvořák	Jméno Matěj	
PRACOVISŤE	ZČU - FST – KPV		
DRUH PRÁCE	DIPLOMOVÁ	BAKALÁŘSKÁ	Nehodící se škrtněte
NÁZEV PRÁCE	Tvorba univerzálního prostředí ve virtuální realitě pro interakci s průmyslovými 3D modely		

FAKULTA	strojní	KATEDRA	KPV	ROK ODEVZD.	2024
---------	---------	---------	-----	-------------	------

POČET STRAN (A4 a ekvivalentů A4)

CELKEM	69	TEXTOVÁ ČÁST	63	GRAFICKÁ ČÁST	6
--------	----	--------------	----	---------------	---

STRUČNÝ POPIS (MAX 10 ŘÁDEK) ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY	Bakalářská práce obsahuje rešerši technologie virtuální reality a používaných softwarů pro tvorbu aplikací. Cílem této práce je vytvořit aplikaci ve virtuální realitě, ve které budou prezentovány různé způsoby interogování s průmyslovými 3D modely. Tato aplikace má potencionální využití ve výuce a v podnicích za účelem edukace.
KLÍČOVÁ SLOVA ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE	Virtuální realita, Unity 3D, Blender, aplikace, 3D model, interakce

SUMMARY OF BACHELOR SHEET

AUTHOR	Surname Král	Name Filip		
STUDY PROGRAMME	B0715A270013 Mechanical Engineering			
SUPERVISOR	Surname (Inclusive of Degrees) Ing. Dvořák	Name Matěj		
INSTITUTION	ZČU - FST - KPV			
TYPE OF WORK	DIPLOMA	BACHELOR	Delete when not applicable	
TITLE OF THE WORK	Creating a universal virtual reality environment for interacting with industrial 3D models			

FACULTY	Mechanical Engineering	DEPARTMENT	KPV	SUBMITTED IN	2024
----------------	------------------------	-------------------	-----	---------------------	------

NUMBER OF PAGES (A4 and eq. A4)

TOTALLY	69	TEXT PART	63	GRAPHICAL PART	6
----------------	----	------------------	----	-----------------------	---

BRIEF DESCRIPTION TOPIC, GOAL, RESULTS AND CONTRIBUTIONS	<p>The bachelor thesis contains a research of virtual reality technology and software used for creating applications. The aim of this thesis is to create a virtual reality application in which different ways of interacting with industrial 3D models will be presented. This application has potential use in education and in companies for educational purposes.</p>
KEY WORDS	<p>Virtual reality, Unity 3D, Blender, application, 3D model, interaction</p>

Obsah

Úvod.....	14
1 Technologické možnosti	15
1.1 Navigace ve VR	15
1.1.1 Sledování pohybu.....	16
1.1.2 Locomotion	16
1.1.3 Ruční ovládání ve VR	18
1.2 Hardware	18
1.2.1 Volba VR headsetu.....	19
1.2.2 Budoucnost VR hardwaru	20
2 Využití virtuální reality v praxi.....	22
2.1 Virtuální realita v průmyslu 4.0	23
2.1.1 Virtuální realita pro chytré továrny.....	23
2.1.2 Virtuální realita v internetu věcí.....	23
2.2 Virtuální realita ve strojírenství.....	24
2.3 Virtuální realita ve školství	25
3 Možnosti implementace 3D modelů do VR.....	26
3.1 Analýza technických znalostí pro VR.....	26
3.2 Optimalizace modelů.....	27
3.3 Interakce ve VR.....	27
3.3.1 Uživatelská rozhraní.....	27
3.3.2 Manipulace s 3D modely.....	28
3.3.3 Zhodnocení možností implementace modelu ve VR	29
4 Software pro vývoj aplikace a tvorbu modelů	30
4.1 Modelovací software	31
4.1.1 SketchUp	31
4.1.2 Blender	31
4.1.3 Autodesk Maya	32
4.1.4 ZBrush.....	33
4.1.5 Volba modelovacího softwaru	33
4.2 Vývojový software	34
4.2.1 Unity.....	35
4.2.2 Unreal Engine.....	35
4.2.3 Godot Engine.....	36
4.2.4 CryEngine.....	37

4.2.5	Volba vývojového softwaru	38
5	Blender	39
5.1	Uživatelské rozhraní Blenderu	39
5.2	Modelování v Blenderu	40
5.2.1	Animace průmyslových modelů.....	45
5.2.2	Export modelů a animací.....	46
6	Doprovodný scénář aplikace	48
7	Unity.....	50
7.1	Příprava projektu v Unity	50
7.1.1	Instalace a stažení Unity.....	50
7.1.2	Uživatelské rozhraní Unity.....	50
7.1.3	Nastavení projektu.....	51
7.1.4	Prostředí aplikace	52
7.2	Funkcionalita aplikace.....	53
7.2.1	Tvorba skriptů	53
7.2.2	Spouštění animace.....	56
7.2.3	Spouštění aplikace.....	58
8	Závěr.....	59
9	Seznam použitých zdrojů	60

Přehled použitých zkratek a symbolů

VR	Virtual reality – Virtuální realita
AR	Augmented reality – Rozšířená realita
URP	Universal render pipeline
LTS	Long term support – Dlouhodobá podpora
3D	Three-dimensional – Trojrozměrný
2D	Two-dimensional – Dvojrozměrný
SW	Software
HW	Hardware
IoT	Internet of things – Internet věcí
6DoF	6 Degrees of freedom – 6 Stupňů volnosti

Seznam obrázků

Obr. 1 Ukázka VR headsetu v počátcích [1].....	15
Obr. 2 Ukázka kartézských souřadnic [3].....	16
Obr. 3 Možnosti pohybu ve VR [5].....	17
Obr. 4 Dotazník pro pohyb ve VR [5].....	17
Obr. 5 Ruční ovládání ve VR [4]	18
Obr. 6 Porovnávané headsety [8]	20
Obr. 7 Přístup firem k implementaci AR/VR technologií [9].....	22
Obr. 8 Předpověď Amerického trhu pro VR/AR v roce 2025 [9].....	22
Obr. 9 Reálná a virtuální chytrá továrna [10].....	23
Obr. 10 Virtuální realita v internetu věcí [11].....	24
Obr. 11 Kooperace ve VR [13]	24
Obr. 12 Výuka ve VR [14].....	25
Obr. 13 Oblasti tvořící VR aplikaci [1].....	26
Obr. 14 Vliv počtu polygonů na model [16]	27
Obr. 15 Způsob ulehčení umístění objektů při manipulaci [6]	29
Obr. 16 Vývojový diagram pro 3D VR aplikaci [18]	30
Obr. 17 Ukázka projektu v SketchUp [20].....	31
Obr. 18 Ukázka projektu v Blender [21].....	32
Obr. 19 Ukázka projektu v Autodesk Maya [22]	32
Obr. 20 Ukázka projektu v ZBrush [23]	33
Obr. 21 Ukázka projektu v Unity [26]	35
Obr. 22 Ukázka projektu v Unreal Engine [26]	36
Obr. 23 Ukázka projektu v Godot Engine [27]	37
Obr. 24 Ukázka projektu v CryEngine [29]	37
Obr. 25 Uživatelské rozhraní v Blenderu.....	40
Obr. 26 Model stolu	41
Obr. 27 Model destičky	41
Obr. 28 Model šroubu	42
Obr. 29 Model kuličkového ložiska	42
Obr. 30 Model válečkového řetězu	43
Obr. 31 Model větrné turbíny.....	44
Obr. 32 Model diferenciálu	44
Obr. 33 Ukázka projektu po rozstřelu diferenciálu	45
Obr. 34 Modely po rozstřelu	46

Obr. 35 Ukázka vybraných možností při exportu z Blenderu.....	47
Obr. 36 Speech Studio.....	48
Obr. 37 Uživatelské rozhraní Unity	51
Obr. 38 Prostředí aplikace	52
Obr. 39 Obrázky v prostředí [30]	53
Obr. 40 Skript pro tabulky a spawn.....	53
Obr. 41 Skript metody GenerateObj(int i)	54
Obr. 42 Skript metod RotationObjx/y(float i).....	54
Obr. 43 Skript metody ScaleObj(float i)	55
Obr. 44 Objekt se skriptem Model Selector	55
Obr. 45 Animator Unity	56
Obr. 46 Skript s poli pro animace	56
Obr. 47 Skript s metodou StartRozpadReverse()	57
Obr. 48 Skript s coroutine	57
Obr. 49 Použitá tlačítka.....	58

Seznam tabulek

Tab. 1: Tabulka s druhy headsetů [7]	19
Tab. 2 Tabulka porovnávaných headsetů [8]	19
Tab. 3 Tabulka porovnání modelovacích softwarů	33
Tab. 4 Tabulka porovnání vývojových softwarů	38
Tab. 5 Testování aplikace.....	58

Úvod

Virtuální realita je dynamicky se vyvíjející technologie, která nachází stále širší uplatnění ve strojírenském průmyslu. Tento technologický pokrok umožňuje vznik nových projektů, které využívají širokých možností a potenciálu virtuální reality pro zlepšení kvality výuky nebo zaškolování zaměstnanců. Největší výhodou virtuální reality je její schopnost přizpůsobit se specifickým požadavkům aplikací, které realizuje.

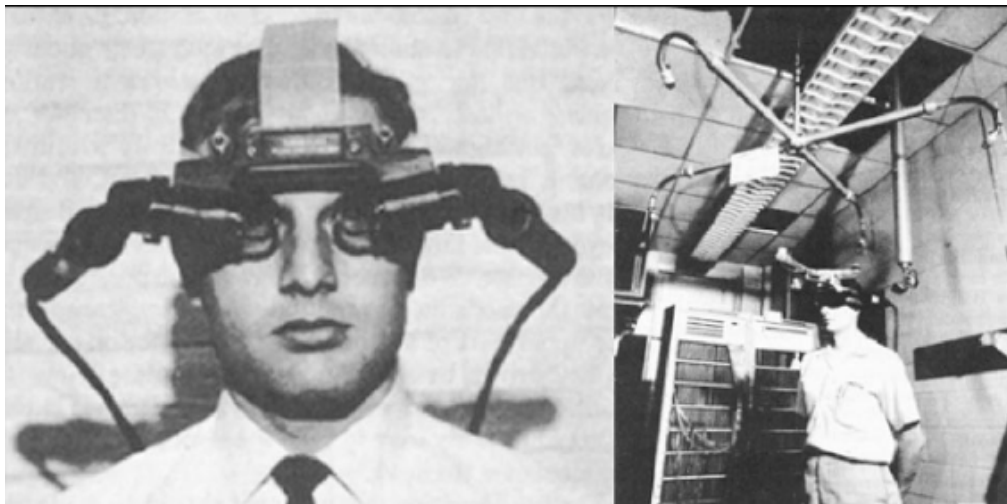
Cílem této bakalářské práce je vytvořit aplikaci, ve které budou prezentovány různé způsoby interakce s průmyslovými 3D modely ve virtuální realitě. Součástí práce je také tvorba realistických digitálních modelů určených pro virtuální prostředí. Aplikace bude obsahovat edukativní prvky, které umožní uživatelům získávat nové znalosti prostřednictvím manipulace a interakce s modely.

Hlavním přínosem této aplikace je rozšíření znalostí uživatelů, čímž se stane užitečným nástrojem pro vzdělávací účely ve školách i pro školení zaměstnanců v průmyslových odvětvích. Díky interaktivním prvkům si uživatelé budou moci modely detailně prohlédnout a lépe porozumět jejich fungování.

1 Technologické možnosti

Ačkoli se může zdát VR jako futuristická technologie, která ještě nedosáhla svého potenciálu, tak v dnešní době můžeme být svědky stále častější implementace VR do života. Od této technologie jsou očekávané velké věci a slibuje si, že změní způsob interakce mezi lidmi a s informacemi. Důvodem rostoucího množství výskytů VR jsou hlavně nové inovativní technologie čoček, senzorů, nové řešení VR headsetů¹ a zvyšování jejich výkonu, vývojem nových technologií vzniká prostor pro tvorbu aplikací, které jsou víc realistické. Dalším důvodem rozšíření této technologie je zlevňování HW² a SW³. Jak HW, tak SW musejí vycházet z možností, které nabízí.

Dohromady vzniká 3D zážitek v digitálním světě. Virtual reality neboli virtuální realita se dá jednoduše popsat jako pohlcující umělý svět, se kterým lze interagovat. Po nasazení VR headsetu a spuštění aplikace dochází k přemístění uživatele do kompletně digitálního světa. Přes to, že největší růst tato technologie zažívá až v posledních pěti letech, tak pokusy o proražení na trh s touto technologií probíhají desítky let. Obr. 1 je ukázka prvního headsetu, který vynalezl Ivan Sutherland v roce 1965. V pravé části obrázku je vidět, že brýle byly připojeny k železné konstrukci s trubkami. [1] [2]



1.1 Navigace ve VR

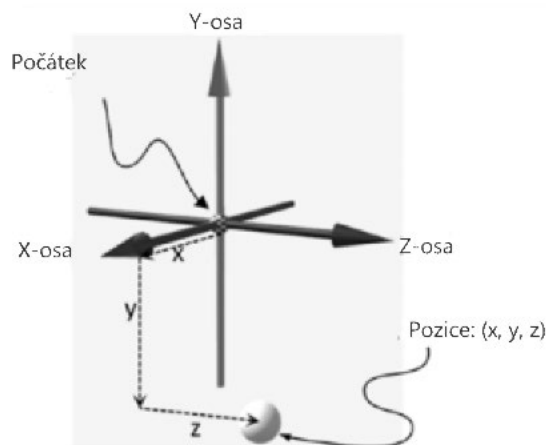
Jelikož je virtuální realita pouze digitální svět, tak veškeré dění v tomto světě je tvořené matematickými algoritmy a výpočty. Pohyb je základní funkcí interaktivního prostředí. Veškerý pohyb ve VR je vyměřen přes kartézské souřadnice, které sledují pozici a orientaci VR headsetu, ve kterém jsou zabudované senzory, gyroskopy, akcelerometry a spousta dalších senzorů, které vyměřují pozice objektů ve skutečném světě a převádějí je na kartézské souřadnice. Dalšími příklady kartézských souřadnic ve VR je sledování rukou nebo ovladačů, které umožňují další pohyb, interakce s virtuálními objekty a prostředím, které vypočítává vzdálenost mezi uživatelem a cílem, při dostatečné vzdálenosti může s virtuálním světem interagovat. Tyto polohy jsou vyměřované ve skutečném čase (real-time). Kartézské souřadnice

¹ VR headset – zařízení na hlavě, které obsahuje display, reproduktory, sledovací systém a ovládací prvky

² Hardware – hmatatelné vybavení počítače

³ Software – programy v počítačích

pro pohyb ve 3D je tvořen třemi proměnnými hodnoty (x , y , z). Tyto proměnné hodnoty se neustále mění podle pozice (viz Obr. 2). [3]



Obr. 2 Ukázka kartézských souřadnic [3]

1.1.1 Sledování pohybu

Pohybování ve VR umožňuje uživateli zažívat imerzní zážitek, ten musí probíhat v reálném čase. Pohyb v reálném čase je hlavně důležitý kvůli odezvě na pohyb uživatele. Při pomalé odezvě dochází k nevolnostem a špatnému uživatelskému zážitku. Prvním způsobem sledování pohybu uživatele bylo pomocí zevnějšku (outside-in), trend ale ukazuje, že efektivnějším řešením pro uživatele je vnitřní (inside-out) sledování. To na rozdíl od outside-in má zabudované kamery a světla přímo v headsetu, tudíž veškeré sledování prostoru a uživatele probíhá z něj.

Nové modely headsetů také umožňují pomocí průchodného obrazu nastavit hranice, které vytyčují, kam až bude mít uživatel možnost pohybu. Outside-in technologie využívá rozmístěných kamer v prostředí. Ty následovně sledují uživatele a jeho pohyb. Ačkoli je toto řešení přesnější a rychlejší na odezvu, výrobci se zaměřují na to, aby tyto nedostatky u inside-out technologie zlepšili a umožnili uživateli méně práce s přípravou využívání VR headsetu. Sledování prostoru slouží pro získání digitálních informací o 3D prostoru. Tím dokáže headset sledovat pohyb uživatele v šesti stupních volnosti (6DoF⁴). [4] Sledování pohybu se používá u locomotion (viz 1.1.2) a sledování ovladačů a rukou (viz 1.1.3).

1.1.2 Locomotion

Locomotion je základní funkcí pro interakci nejen ve virtuálním prostředí, ale také všude kde se vyskytuje HCI⁵. Locomotion znamená možnost pohybu v prostředí a stejně jako u HW (viz 1.2) dochází také na zlepšování uživatelské přívětivosti a dochází k vzniku nových lokomočních technik nebo zlepšování těch existujících. Na lokomoci vzniká specializovaný HW, který podporuje pohyb ve skutečném světě a zlepšuje jeho podmínky, příkladem může být chodící pás. Pohyby se mohou lišit, ať už jde o teleportaci nebo chození, tak také může být pohyb šplhání. Při špatném řešení locomotion dochází k vzniku motion sickness⁶ a dalších zdravotních problémů.

⁴ 6Dof (6 Degree of Freedom) – umožňuje pohyb v osách x , y , z a rotaci kolem nich

⁵ HCI – Human Computer Interaction, neboli interakce člověka s počítačem

⁶ Motion sickness – nevolnost z pohybu, vzniká při špatně navrženém pohybu ve VR, který neodpovídá očekávání uživatele

Převládající techniky locomotion v odborné literatuře jsou:

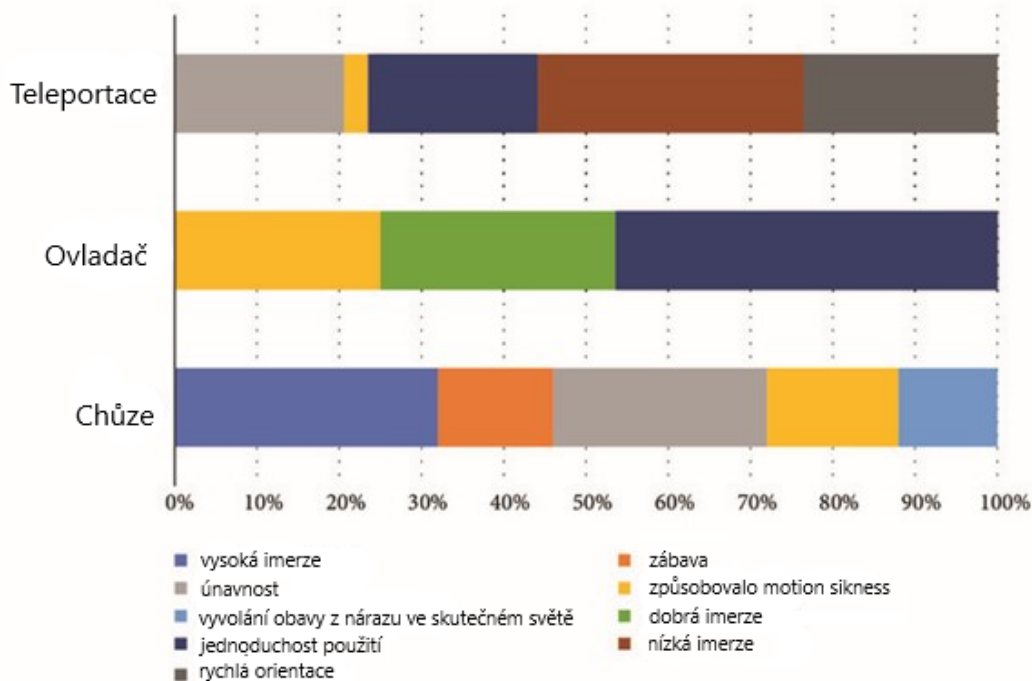
- **Fyzickým pohybem** – Pohyb, který dělá uživatel v reálném světě, se odráží na pohybu ve VR. Jde o kontinuální pohyb, který se dále dělí na možnost chození po místnosti nebo pohyb nohou v sedě, popřípadě chůze na místě.
- **Ovladači** – Ovládání pohybu je dáno joysticky na ovladači, které pohybují uživatele ve VR neomezeně na prostoru ve skutečném světě. Tento pohyb je také kontinuální.
- **Teleportací** – Pohyb je tvořen skoky na pozice ve VR. Portace je vyvolána umělou interakcí ve VR a je nekontinuální. [5] [6]



Obr. 3 Možnosti pohybu ve VR [5]

Volba pohybu je závislá primárně na možnostech prostoru, HW, vhodnosti pro aplikaci a její stavbou. Pohyb joystickem a teleportací je obzvláště výhodný pro pohyb na velké vzdálenosti s omezeným prostorem, zatímco chůze spíše pro vyšší úroveň imerze a zábavy.

Na Obr. 4 je vidět studie na 26 lidech, kteří používali pohyb pro hru ve VR, po dokončení hry vyplnili dotazník, ze kterého lze říct, že nejvíce zábavný a imerzní pocit měli z fyzického pohybu. Docházelo ale také k strachu z nárazu, motion sickness a únavy, která vzniká při dlouhém používání VR. Nejjednodušším pohybem byl pohyb joystickem, na který mohou být uživatelé zvyklí z konzolí, také měl dobrou imerzi, ale docházelo k vysokému výskytu motion sickness, který vznikal rozdílem očekávaného a skutečného pohybu ovladačem. Teleportování v lidech vyvolávalo únavu a nízkou imerzi, naopak uživatelům nabídla rychlou navigaci po prostředí a jednoduchost. [5]

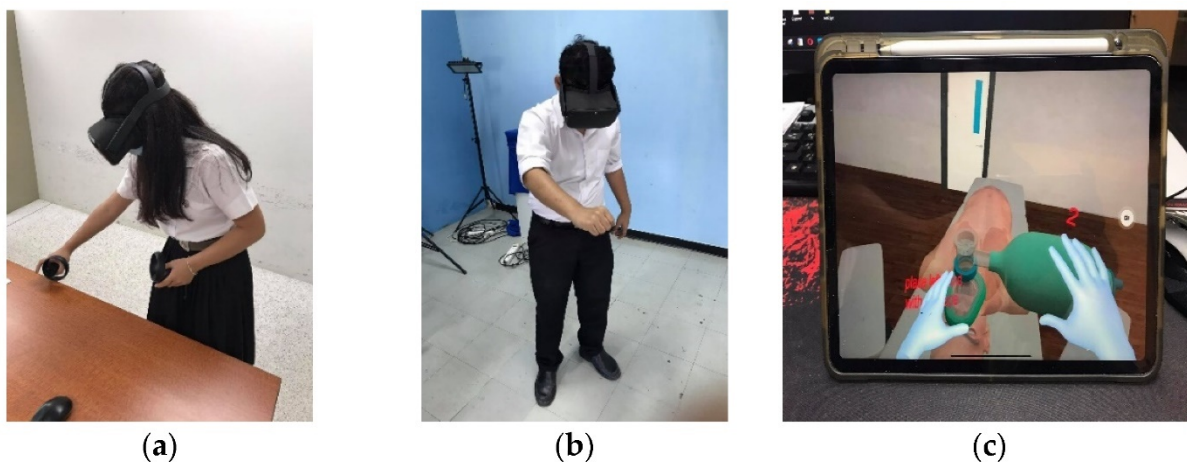


Obr. 4 Dotazník pro pohyb ve VR [5]

1.1.3 Ruční ovládání ve VR

Existují dva způsoby ovládání pomocí rukou ve VR. Jedním z nich je ovládání pomocí ovladačů a druhým je tzv. handtracking, který vznikl za účelem vytvoření ještě více imerzního světa. Ruce jsou sledovány headsetem, který má zabudované kamery, ty sledují polohu ovladačů či rukou a následně je vykreslují digitálně ve VR. Jak pohyb rukou, tak ovladače mají své možnosti využití, které se odlišují podle aplikace a je třeba jejich možnosti zohlednit za účelem lepšího řešení uživatelského zážitku.

Ovladače jsou hlavně bezdrátové a sledované algoritmem Simultaneous Localization And Mapping (SLAM⁷). LED diody na ovladači jsou sledovány kamerami a umožňují vysokou přesnost sledování v 6DoF. Ovládání prostředí je pomocí tlačítek na ovladači. Sledování rukou je nejčastěji řešeno pomocí infračervených kamer, ale existují i další řešení sledování např. ultrazvukem nebo rukavicemi pomocí RGB. Infračervené kamery jsou buďto v brýlích (inside-out) anebo v prostředí (outside-in), při ručním ovládání není potřeba dalšího příslušenství. Ruce jsou vykreslovány algoritmem v počítači, který sleduje pohyb rukou a gesta, pomocí kterých se ovládá VR. Na Obr. 5. lze vidět jednotlivé způsoby ovládání: (a) s ovladači; (b) pomocí handtrackingu; (c) ukázka ovládané aplikace. [4]



Obr. 5 Ruční ovládání ve VR [4]

Ovládání ovladači je vhodnou volbou pro aplikace, které vyžadují velmi přesného ovládání, zatímco handtracking nabízí vyšší stupeň imerze a zábavy. Joysticky také často mají víc tlačítek, které mají různé funkce a musí se s nimi noví uživatelé VR seznamovat, což může být časově náročné.

1.2 Hardware

Velkým skokem dopředu v technologii, který oživil VR byl v HW, starší technologie vyžadovali výkonné a pro tu dobu drahé počítače, které byly pro širokou veřejnost nedostupné. K nim byl připojen headset se senzory, ovladači, reproduktory a dalšími zařízeními pro správné fungování, takže docházelo kvůli kabelům k omezení pohybu. Nyní jsou tato zařízení přímo umístěna v headsetech, popřípadě napojena pomocí Bluetooth⁸ či Wi-Fi⁹. [2]

⁷ SLAM – využívá se k vytvoření mapy prostředí, která si získá sběrem dat několika sensorů, následně se v této mapě zjišťuje chtěná pozice

⁸ Bluetooth, Wi-Fi – bezdrátové signály pro přenos dat mezi zařízeními

Headsety lze rozdělit podle více měřítek, ale hlavním kritériem je jaký má zdroj:

- **Tethered** – headset musí být fyzicky připojen na vnější zdroj. Tím zdrojem nejčastěji bývá počítač nebo herní konzole, který musí mít dostatečný výkon na požadované použití.
- **Standalone** – tato kategorie vyžaduje nejméně vnějších interakcí, protože na rozdíl od tethered a smartphone mají vlastní procesor, snímač, baterii, display, úložiště a reproduktor. Občas jsou také nazývány all-in-one.
- **Smartphone** – stejně jako tethered potřebují vnější zdroj. Zdrojem je zpravidla smartphone, který je zasunut do náhlavní soupravy. [7]

Tab. 1: Tabulka s druhy headsetů [7]

Druhy headsetů				
Druh	Výhody	Nevýhody	Použití	Hlavní výrobci
Tethered	Vysoce výkonné, realistické aplikace, netřeba dbát baterie	Omezený pohyb kabelem, nároky na prostor, vysoké náklady na výkonný počítač	Umění, hry, turismus, profesionální aplikace	PlayStation, Meta, HTC, Valve, HP
Standalone	Nízké nároky na prostor, netřeba externího zařízení, nižší náklady, schopnosti tethered	Omezený výkon, baterie	Sledování filmů, jednoduché hry, sociální VR, výukové aplikace	Meta, Pico, XRSpace, HP, DPVR, Skyworth
Smartphone	Dostupnost, jednoduchost použití	Kompatibilita mobilu a brýlí, omezený výkon	Sledování filmů, jednoduché hry, sociální VR	Zeiss, Google, Samsung

1.2.1 Volba VR headsetu

Trh s VR headsety nabízí velké množství kvalitních headsetů od různých výrobců. Pro porovnání produktů byly zvoleny 2 standalone a 2 tethered headsety, protože smartphone headsety mají omezený výkon, který může být v průmyslových aplikacích překážkou. Standalone produkty jsou Pico Neo 3 a nejprodávanější headset Oculus Quest 2, tethered produkty HTC Vive Pro 2 a HP Reverb G2.

Tab. 2 Tabulka porovnávaných headsetů [8]

Porovnávané produkty								
Produkt	Druh	Cena [Š]	Ovládání	GPU	RAM [GB]	Rozlišení (na oko)	Sledování prostředí	Hmotnost [g]
Oculus Quest 2	Standalone	299	Handtracking Ovladače	Qualcomm Snapdragon XR2	6	1832x1920	Inside-out	503
Pico Neo 3	Standalone	390	Ovladače	Qualcomm Snapdragon XR2	6	1832x1920	Inside-out	610
HTC VivePro 2	Tethered	1399	Ovladače	Zdroj	Zdroj	2448x2448	Outside-in	850
HP Reverb G2	Tethered	599	Ovladače	Zdroj	Zdroj	2160x2160	Inside-out	498

Specifikace porovnávaných produktů lze vidět v Tab. 2. Headsety jsou si, co se týká vzhledu, velmi podobné, jednotlivé produkty lze vidět na Obr. 6. Všechny mají HW umístěný pouze v přední části brýlí. Specifikace headsetu jako výdrž baterie, GPU¹⁰, CPU¹¹, úložiště, RAM¹² jsou rozdílné pro standalone a tethered, protože u standalone brýlí jsou součástí headsetu, zatímco tyto parametry získává tethered headset podle zdroje, ke kterému jsou napojené. Standalone headsety jsou cenově o dost méně nákladné než tethered, které navíc vyžadují neustálé připojení ke zdroji a dostatečně výkonný počítač. Tethered headsety mají lepší rozlišení na oko a pole pohledu. Za povšimnutí také stojí hmotnost HTC Vive Pro 2, která může být při dlouhé době používání překážkou. Zároveň je tento headset outside-in, tudíž uživatelé musí rozmířovat sledovací zařízení po místnosti. Produkt od Mety Oculus Quest 2 umožňuje jak ovládání ovladačem, tak handtracking. [8]



Obr. 6 Porovnávané headsety [8]

Výběr headsetu podle parametrů je dán tím, jakým způsobem bude headset používán a jaké jsou vstupní náklady. Existuje velmi široké pásmo použití (viz 2 Využití virtuální reality v praxi). Tethered headsety jsou cenově násobně dražší. Také neumožňují handtracking jako Oculus Quest 2. Přes to, že umožňují kvalitnější obraz a s výkonnějším zdrojem plynulejší chod, tak pro obvyklé průmyslové aplikace bude výkon standalone headsetů dostatečný. Když se porovnají pouze standalone headsety, tak Oculus Quest 2 poráží parametry produkt Pico Neo 3. Jako jediný také umožňuje volbu mezi ručním ovládaním, což lze využít při tvorbě více imerzních aplikací. Jde také o téměř nejlehčí headset. Volen je headset Oculus Quest 2 od společnosti Meta.

1.2.2 Budoucnost VR hardwaru

I přes neustálý vývoj má nynější VR omezení. Hlavními problémy jsou spojené s výkonem a hmotností HW. Na těchto omezeních již firmy pracují ve výzkumu nových materiálů a výkonnějšího HW. Vzhled bude směřovat na brýle, překážkou jsou plastové nebo skleněné čočky, které zabírají velký prostor a tím určují vzhled brýlí. Řešením se zabývali Andrew Maimone a Jun-Ren Wang z laboratoří Facebook Reality Labs, kteří ve výzkumu řešili možnost holografické optiky.

Použitím moderních optických a konstrukčních možností bude docházet k zmenšování hmotnosti, tím dosáhnou lepší manipulace s brýlemi a možná ani brýle nebudou potřeba, stačit budou čočky, které dají do očí. Samotné VR nebude dostačující a firmy se budou zaměřovat

¹⁰ GPU – Graphics Processing Unit neboli grafická jednotka vykonává graficky zaměřené kalkulace, které jsou následně viditelné na obrazu

¹¹ CPU – Central Processing Unit neboli centrální procesová jednotka vykonává pomocí operací přeměnu vstupních dat na více srozumitelné výstupní informace

¹² RAM – Random Access Memory neboli paměť s náhodným přístupem, jejíž funkce je ukládání krátkodobých informací pro právě běžící program

na kombinaci světů AR¹³ a VR, tedy MR¹⁴. Některé headsety to umožňují už nyní. Zvyšování výkonu nebude potřeba, VR zařízení se připojí přes síť na externí zdroj, čímž se uvolní prostor pro HW. [2] V blízké budoucnosti je očekávaný vstup produktu Apple Vision Pro od giganta Apple na trh, který bude mít prodejní cenu od 3 499 \$, což je 79 495 Kč. Tento produkt bude k prodeji od počátku roku 2024.

¹³ AR – Augmented reality neboli rozšířená realita má hlavně funkci nástroje pro reálný svět, pomocí chytrého zařízení a kamery lze v zařízení naimportovat objekty do prostředí na obrazovce zařízení

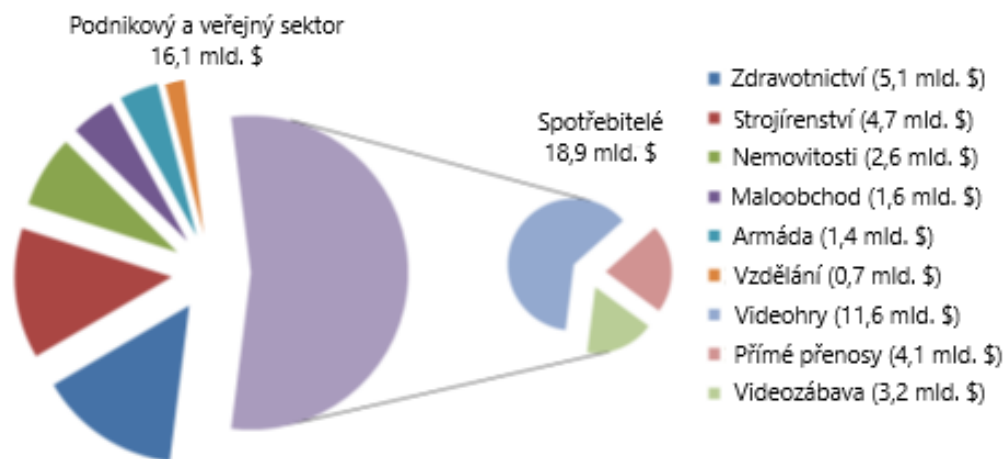
¹⁴ MR – Mixed reality neboli mixovaná realita je kombinací VR a AR, zároveň umožňuje přepínání mezi těmito realitami. V mixovaném prostředí může headset promítat obraz před námi pomocí kamer a zároveň naimportovat části virtuálního světa, se kterými budeme interagovat

2 Využití virtuální reality v praxi

Za účelem konkurenceschopnosti by se společnosti měly přizpůsobovat novým technologiím, Obr. 7 představuje, jak je implementována VR/AR technologie v USA. Světle modré pole (33,3 %) popisuje firmy, které nemají v plánu tyto technologie využívat. Důvodů pro to může být více firmy, například nemají dostatek zdrojů na investice. Investice není tvořena pouze HW, ale jsou s ní spojeny i náklady na SW a školení zaměstnanců a změna infrastruktury podniku. Fialové pole (30,8 %) představuje firmy, které měly možnost implementace, ale nepřijaly jej, tudíž jim nevyhovovaly nabídky, které jim daly specializované firmy. Zelené pole (15,8 %) jsou firmy, které tvrdí, že plánují tuto technologii implementovat do tří let. Odklad může být způsoben přípravou na investici a infrastrukturu. Červené pole (7,5 %) reprezentuje firmy, které plánují implementovat do roka a tmavě modré pole (12,5 %) společnosti, které ji již využívá. Tento obrázek průzkumu trhu představuje vysoký potenciál pro trh s VR. Plného rozsahu využije pouze 2,5 % firem. Nejčastěji to jsou nové moderní firmy, které vznikaly už při rozšiřování VR a zaměřují se na využití výhod průmyslu 4.0, také jsou tyto firmy nazývané chytré továrny budoucnosti. [9]



Obr. 7 Přístup firem k implementaci AR/VR technologií [9]



Obr. 8 Předpověď Amerického trhu pro VR/AR v roce 2025 [9]

2.1 Virtuální realita v průmyslu 4.0

Firmy procházejí neustálou modernizací za cílem zvýšení efektivity výroby. To Průmysl 4.0 umožňuje pomocí informačních a technologických inovativních řešení. Zaváděné jsou chytré IT systémy, které umožňují vytvoření chytré továrny. Implementací těchto technologií dochází ke zlepšení organizace, práce s produkty, informacemi či daty a výsledkem je snížení časových a pracovních nároků na zaměstnance. Zde se našlo i uplatnění pro VR, které pomáhá při automatizaci a digitalizaci složitých výrobních procesů. [9]

2.1.1 Virtuální realita pro chytré továrny

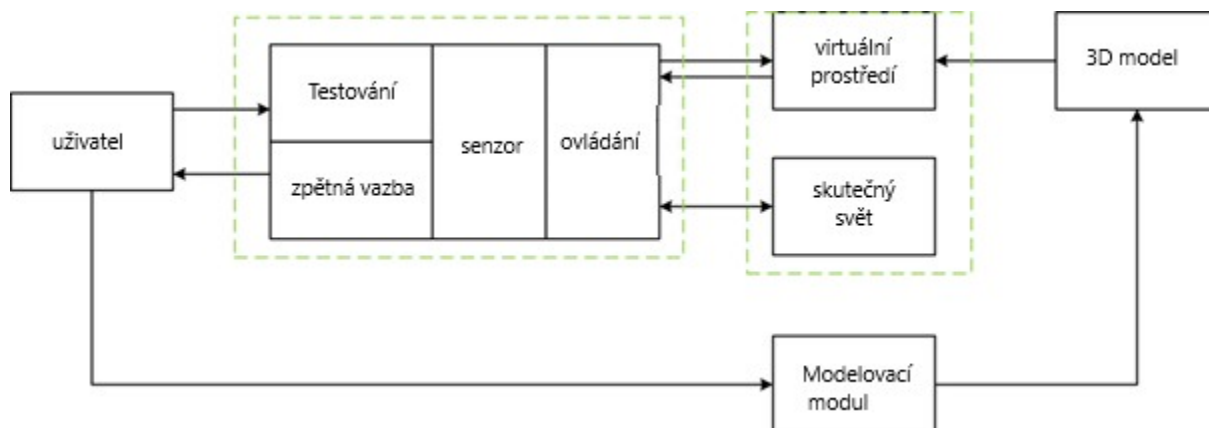
Chytré továrny používáním digitalizace, nových technologií a zaměřením na sledování procesu, umožňují získávání důležitých informací pro aplikování pro skutečnou výrobu. Nové technologie VR se stále častěji vyskytují v podnicích za účelem doplňování inteligentních výrobních systémů. Umožňují propojení kybernetického světa, který je tvořen daty z procesu a přenášejí jej do fyzického světa, který je lépe srozumitelný pro uživatele. VR se v podnicích nejčastěji používá pro zaučování zaměstnanců na montážní, demontážní práce. Dále jej lze využít pro simulaci výrobního cyklu, testování výrobku. Využívá se také pružnosti, kterou výrobní průmysl potřebuje a tu schopnost VR má, lze velmi rychle vymýšlet a měnit scénáře, které mohou nastat. [10]



Obr. 9 Reálná a virtuální chytrá továrna [10]

2.1.2 Virtuální realita v internetu věcí

Hlavní změnou, kterou přináší VR je pohlcující zážitek oproti videím, obrázkům a zvukům, které byly dosud využívány. Vzniká tedy nový způsob komunikace mezi člověkem a strojem, který je více efektivní. Toho se využívá v IoT, které pomocí internetu umožňují komunikaci stroj-stroj, stroj-člověk a člověk-člověk v reálném čase. Interakce ve VR se odráží do skutečného světa okamžitě, kdy jeden uživatel může být ve VR a druhý ve skutečném světě a dávat si zpětnou vazbu. Sběrem dat senzory se následně nahrávají na server, ze kterého lze data dále využívat pro komunikaci mezi stroji a lidmi. Možnost těchto technologií je velká univerzálnost použití, které nevyžaduje přívětivé podmínky. Rozrůstáním trojrozměrných technologií a grafů dochází k větší realističnosti, a tím lepší interpretace dat a skutečného stavu. [11]



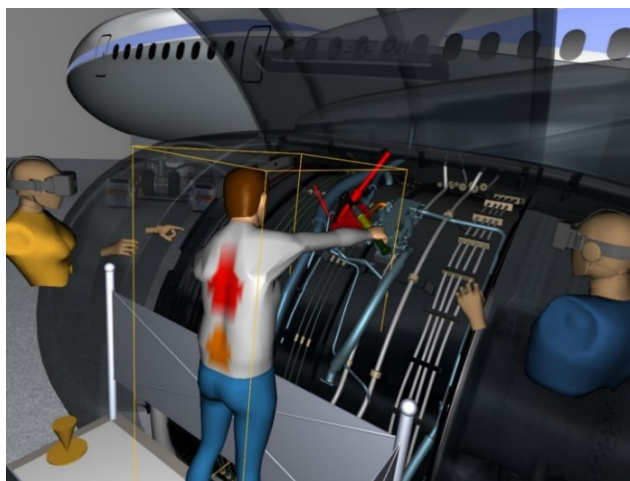
Obr. 10 Virtuální realita v internetu věcí [11]

2.2 Virtuální realita ve strojírenství

Hlavním důvodem implementace VR do strojírenství je jeho návratnost investic (ROI). Pomocí VR lze produkt testovat přímo v umělém prostředí a dělat v něm úpravy a interagovat s ním. Toto je velká úspora času a nákladů. Možnost 3D vizualizace ulehčuje pochopení problematiky, zároveň vzniká prostor pro nové nápady. Virtuální prostředí není závislé na místě, to znamená, že lze pracovat na jednom projektu s teamem, který se nachází jinde. Virtuální realita tedy přináší vysoké úspory, které lze rozdělit na hmotné a nehmotné. Hmotnými úspory jsou šetření nákladů a zlepšení produktu. Úspora nákladu vzniká virtuálním testováním a spoluprací. Nehmotné přínosy jsou zajímavější a interaktivnější pracovní prostředí pro zaměstnance, prostor pro kreativitu, zrychlení procesu vývoje a lepší komunikace nad projektem.

Zde jsou některá odvětví ve strojírenství a možnost použití:

- **Optimalizace výroby** – efektivní navržení haly nebo skladu
- **Lean manufacturing** – efektivní řešení zásob pro výrobu
- **Training** – zaučování budoucích pracovníků (svařování, obrábění, montáž, ...)
- **Konstruování** – import CAD projektů do VR
- **Automobilový design** – návrh a prohlížení interiéru i exteriéru
- **Hornictví** – tvorba efektivního řešení postupu těžby
- **Stavebnictví** – tvorba efektivního řešení staveb
- **BOZP** – školení [12]



Obr. 11 Kooperace ve VR [13]

2.3 Virtuální realita ve školství

S rozšířením technologie do širokého spektra použití došlo také na vzdělávání. Je důležité využít potenciálu interaktivních technologií pro výuku, technologie používané pro nynější vzdělávání bývají nejčastěji prezentace na plátnu a výuka na počítačích. Technologie ovšem představují také výzvu ve školství, studenti tráví stále více času na mobilech, a to i při hodinách, tudíž se nesoustředí. Toto je prostor pro implementaci virtuální reality, která získá od studenta stoprocentní soustředění. Student má nasazené brýle a je obklopen virtuálním světem, který s ním interaguje, mluví na něj a vyžaduje od něj zapojení.

Aplikace se odvíjí podle toho, jak student plní zadání, čímž dochází k přizpůsobení na rychlost jednotlivců. Výhodou je také spojení teorie i praxe, což může být pro školy v reálném světě nákladné a rizikové pro bezpečí studentů. Průběh výuky může vypadat tak, že učitel se seznámí s technologií a náročností aplikace pro studenty. Výhodou VR je také, že si může nechat udělat aplikaci podle svých představ a výuku si kolem toho přizpůsobit. Učitel má schopnost spouštět a vypínat aplikaci. Před spuštěním dochází k seznámení studentů s technologií a způsobem používání. Veškeré dění, které se odehrává v brýlích může sledovat přes chytré zařízení jako je mobilní telefon, tablet nebo počítač. [14]



Obr. 12 Výuka ve VR [14]

3 Možnosti implementace 3D modelů do VR

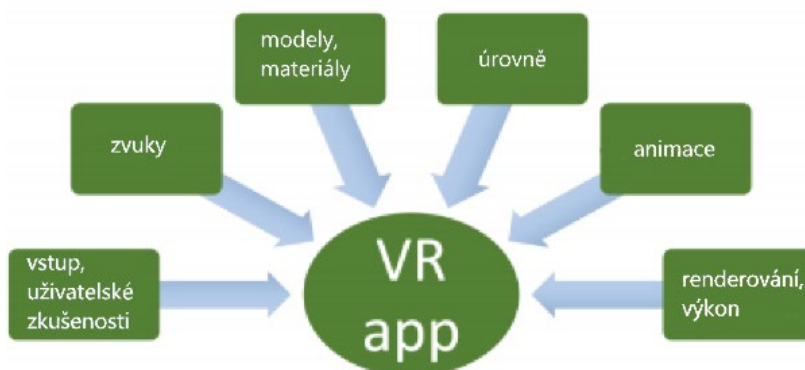
Analýza možností implementace 3D modelů má velký význam při vývoji VR. V technologických možnostech je uvedeno, jak může být VR využita. Je třeba si položit otázku, co se od VR očekává, jak bude používána a jaké budou možnosti ovládání, konkrétně co bude možné s 3D modely dělat. Zároveň je třeba mít na mysli, že často platí čím kvalitnější modely jsou, tak přes lépe vypadající aplikaci dochází zároveň k vyšším nárokům na vykreslování a výkon VR zařízení. Postup je takový, že prvně je požadovaný model potřeba vytvořit, a to pomocí CAD SW anebo alternativních modelovacích SW (viz 4 Software pro vývoj aplikace a tvorbu modelů). V těch se následně vykresluje 3D geometrie objektů, po dokončení modelování dochází k exportu modelu. Export modelu je třeba udělat v takovém formátu, aby byl kompatibilní s VR. Posledním krokem je nastavení interakcí s modelem. [15]

3.1 Analýza technických znalostí pro VR

Tvorba jedné aplikace pro VR vyžaduje uplatnění mnoha oblastí, které jsou vidět na Obr. 13. Uplatnění pro tvorbu aplikací lze vidět u různých profesí, VR je multimediální platforma, která potřebuje 3D design, animace a hudbu. Softwarové inženýrství, kde je potřeba platformových¹⁵ operací, debuggování¹⁶ a programování¹⁷. Architektura a modelování se využívá u návrhu úrovní, prostředí a tvorbě 3D modelů, také se využívá u scénáristů pro odpovídající děj, osvětlování a další.

Při vývoji aplikace se využívá těchto oblastí:

- **Velikost prostředí** – rozměry by měly odpovídat reálným
- **Ovládání z první osoby** – ovládání rukami/ovladačem, pohyb po prostředí
- **Nástroje uživatelského rozhraní** – možnosti interakce s prostředím
- **Osvětlení a renderování** – volba správného řešení světla, grafiky a efektů
- **Fyzika a gravitace** – zvláště důležité pro větší pocit imerze
- **Animace** – pohyby ve scéně, které mohou být předdefinované nebo interaktivní
- **Tvorba, spuštění a optimalizace aplikace** – různé headsety mají různé použité nástroje na spuštění aplikace a optimalizace zlepšuje chod
- **Programování** – umožňuje dát interaktivní vlastnosti prostředí [1]



Obr. 13 Oblasti tvořící VR aplikaci [1]

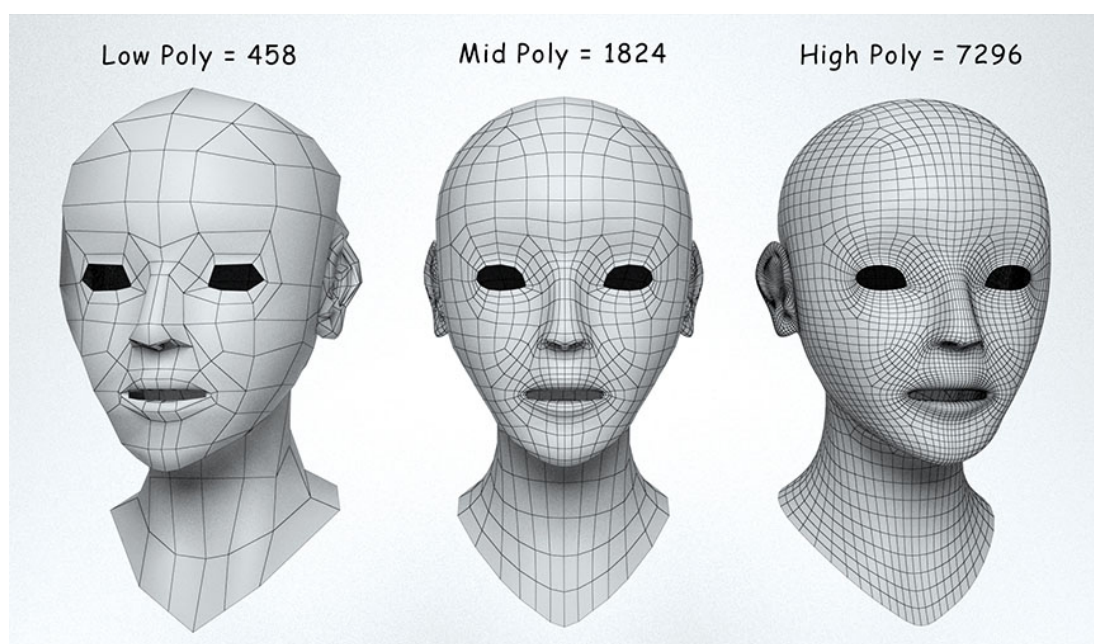
¹⁵ Platforma – kombinace hardwaru a operačního systému potřebná pro správný chod softwaru

¹⁶ Debuggování – opravování chyb, které vznikají v softwaru při vývoji

¹⁷ Programování – tvorba zdrojového kódu v programovacím jazyce

3.2 Optimalizace modelů

Standalone headsety mohou mít u náročných aplikací problémy s výkonem. Ten lze ušpóřit při vykreslování modelů pomocí optimalizace. Modely jsou tvořeny polygony, což jsou mnohoúhelníky. Poskládáním těchto polygonů dohromady vzniká polygonová síť neboli chtěný model. Jejich počet určuje kvalitu modelu a čím nižší počet polygonů tzv. low-poly tím nižší nároky na výkon jsou. Dochází ovšem také ke zhoršení vzhledu modelu, protože větším množstvím ploch lze kvalitněji vykreslit. Lze si to představit jako u pixelu na obrazovkách. Ušetření výkonu na modelech umožňuje plynulejší zážitek ve VR. Existují také mid-poly se střední náročností a high-poly modely, ty jsou vysoce náročné na výkon a pro VR jsou téměř nepoužitelné. [15] Mezi počtem polygonů lze pomocí algoritmů v SW přepínat a vybrat tím vhodné množství, vliv počtu polygonů na model je vidět na Obr. 14.



Obr. 14 Vliv počtu polygonů na model [16]

3.3 Interakce ve VR

Součástí uživatelského rozhraní jsou možnosti, jakými lze ve VR interagovat. Mezi základní se řadí možnost pohybu, ovládání aplikace výběrem a manipulací s objekty a samotné ovládání platformy nezávisle na aplikaci. Celková interakce lze pak rozdělit na realistickou anebo magickou, záleží, jaká míra přirozenosti odpovídá pro aplikaci nejlépe. V realistických interakcích lze očekávat pohyb člověka, uchopování věcí pomocí rukou, zatímco v magických interakcích se využívá pohyb teleportací a manipulace s místem a rozměry modelu na dálku. Volba přístupu k aplikaci by měla odrážet co nejlepší efektivitu aplikace a používají se i kombinace těchto možností. Obecně se pro průmyslové použití využívají realistických prostředí, která více napodobují praxi a jsou také vhodná pro herní průmysl magických interakcí. [17]

3.3.1 Uživatelská rozhraní

Zde má uživatel možnost ovládat dění aplikace například změnu scény a nastavení. Ovládání se částečně dá převzít z 2D obrazovek, je možné používat ovládání pomocí menu, tlačítek, dvojkliku, přetáhnutí objektu, ale také hlasové příkazy a gest. Nevýhodou uživatelských rozhraní je, že je velmi náročné udělat takové, které by odpovídalo realitě 1:1 a zároveň bylo

uživatelsky přívětivé. Uživatelská rozhraní je možné kombinovat, ale musí zůstat dostatečně jednoduché pro nového uživatele.

Možnosti rozhraní jsou:

- **Menu** – Tato technika je nejrozšířenější, její výhodou je univerzálnost použití. Je možné pracovat se základními vlastnostmi jako je počet položek na výběr, tvar, barva, umístění, možnost ovládání, interakce, které menu bude vytvářet, zvuky výběru. Menu lze dále využít pouze jako pevně umístěnou nabídku nebo s kontextem přiřazené k určitému objektu, také lze sledovat pohled uživatele a být neustále na očích.
- **3D widgety** – Jsou přímo spojené s interakcí uživatele ve VR prostředí. Jsou to 3D objekty, jejichž funkcí je doplnění VR. Dávají uživateli informace o objektech, interakcích, akcích, možnostech ovládání a využívá se primárně jako informační prostředek.
- **Tangibles** – Jsou to rekvizity a zpravidla objekty. Tyto objekty mají funkci nástroje. Uživateli se po uchopení takového objektu změni způsob ovládání. Umožňují efektivní řešení pro aplikace s vyšším počtem možností ovládání.
- **Hlasové příkazy** – Hlasové příkazy jsou ideálním řešením pro spojení víc způsobů ovládání najednou, protože nevyžadují ovládání rukou. Tato technika má své výhody a nevýhody, nevýhodami je nutnost se naučit příkazy, zároveň mluvením může narušovat klid pracovního prostředí a kdyby bylo více lidí používajících těchto příkazů nebo lidí procházejících, mohlo by docházet k nechtěným interakcím. Výhodou je možnost ovládat aplikaci a její objekty bez nutnosti věc vidět nebo na ní dosáhnout.
- **Gesta** – Stejně jako hlasové příkazy je potřeba se gesta naučit. Pro uživatele není problém vyvolat akci, i když objekt nevidí a pomocí gest jej vyvolá, nevýhodou gest je, že jsou pro uživatele nepřírozená a není zvyklí využívat vyšší počet gest. [17]

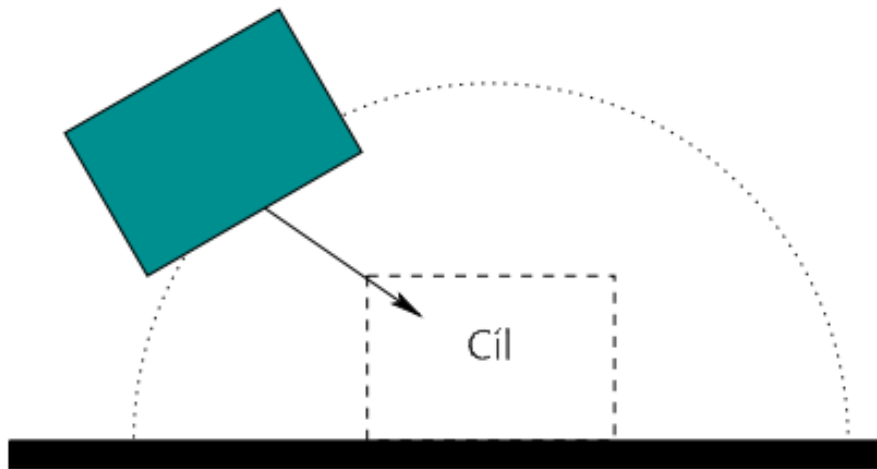
3.3.2 Manipulace s 3D modely

Jednou ze základních možností interakce ve VR je uchopení modelu a manipulace s ním. Manipulace s modelem umožňuje uživateli měnit základní vlastnosti modelu, které nemusí být možné v reálném světě zopakovat. Uživatel může měnit polohu, natočení, rozměry, tvar, barvu a další. To, jaké interakce jsou zvoleny, je dáno účelem aplikace. Stejně možnosti práce se skutečnými objekty a těmi virtuálními se využívá při simulačních aplikacích používané pro zaučování zaměstnanců, například při montáži, ve které mají modely fyzické vlastnosti jako ty reálné. Mohou například po puštění ve vzduchu spadnout vlivem gravitace. Manipulace lze rozdělit na egocentrickou a excentrickou. Egocentrická manipulace se využívá při cílení na co nejvyšší pocit imerze. Uživatel vidí svět v prvním pohledu a umožňuje mu ovládání objektů které vidí před sebou ve skoro realistických rozměrech. Excentrická manipulace se využívá pro jednoduché manipulování objekty u komplexních prostorů. Uživatel se necítí přímo součástí a má na prostor pohled zvenčí. Objekty nemusí být vykreslovány realisticky, ani to není pro komplexní prostory vhodné z hlediska přehlednosti.

Technikami manipulace jsou:

- **Arčball** – Tato technika se využívá výhradně pro manipulaci s orientací objektů. Kolem objektu může být buď viditelná nebo neviditelná koule, která působí jako obal, se kterým má uživatel možnost otáčet. Ten poté rotuje s objektem uvnitř kolem jeho středu a otáčí ním.
- **Virtuální ruce** – tato technika se využívá primárně u simulačních aplikacích za účelem ovládní odpovídajícího reálnému světu. Výhodou je, že je to pro všechny uživatele víc přirozené a není třeba se učit orientaci ovládní jako u ovladačů nebo myši s klávesnicí.
- **Ukazování** – Ukazovací gesta se využívají nejen pro výběr, ale také pro manipulaci s objekty. Objekty mohou být vzdálené a pomocí gest jako je paprsek, nejčastěji vztyčením ukazováčku může vybrat objekt a manipulovat s ním.
- **Voodoo panenky** – Tato excentrická technika ovládní je založena na změně rozměrů objektů. To umožňuje uživateli se v komplexních prostředích lépe orientovat a rozeznávat objekty. [17]

Při manipulativní práci s objekty, může začít docházet k únavě rukou, která se nazývá „gorilla arms“. Ta vzniká při dlouhodobém natažení rukou, čímž si ruce uživatel brzy unaví. Řešením je úprava aplikace, aby k těmto problémům nedocházelo. Využívá se například toho, že pokud uživatel musí přenášet objekt na dlouhé vzdálenosti, tak je pouze jednou uchopí a nemusí celou dobu manipulace držet zmáčkнутý ovladač nebo gesto, pro puštění objektu toto gesto zopakuje. Dalším způsobem ulehčení pohybu je umístění objektu do gravitačního pole, kde se objekt již sám přemístí do cílové pozice (viz Obr. 15). [6]



Obr. 15 Způsob ulehčení umístění objektů při manipulaci [6]

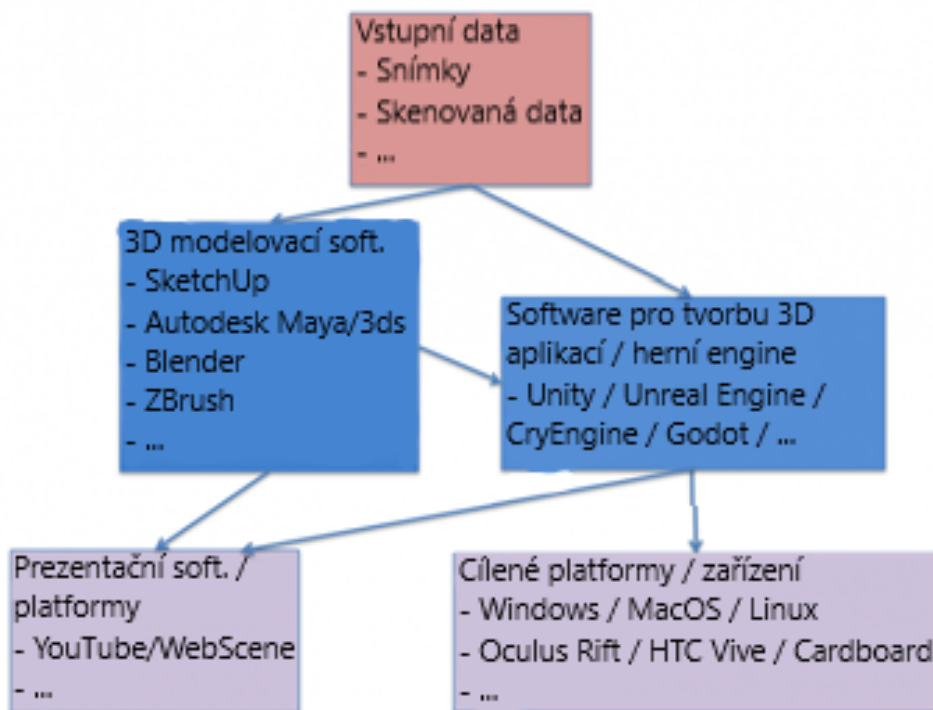
3.3.3 Zhodnocení možností implementace modelu ve VR

Možnosti interakce ve VR jsou široké a je třeba zvolit vhodný způsob ovládní pro interakci s průmyslovými 3D modely. V aplikaci bude možné uživatelem si naimportovat model, uchopit jej do ruky a manipulovat s ním, to znamená, že uživatel uchopením modelu do ruky ho může přemístit a otáčet. Dále bude aplikace mít možnost tlačítek, kterými také bude možné s modely pracovat. Tlačítka budou sloužit jako dálkové ovládní modelu. Uživatel bude pomocí nich mít možnost s modelem otáčet, zvětšovat a zmenšovat. Tlačítka budou ručně ovládná uživatelem. Využije se menu, ve kterém bude nabídka možných modelů, kterými bude moc následovně naimportovat. Další možností interakce by mohl být rozstřel modelu na jednotlivé části, doplnění 3D widgety s informacemi o průmyslovém modelu.

4 Software pro vývoj aplikace a tvorbu modelů

Vývoj plnohodnotné aplikace ve VR je složitý proces, který je výhodné pro přehlednost rozdělit na jednotlivé kroky. V jednotlivých krocích je potřeba zvolit SW, někdy dochází k volbě více SW najednou. Kroky, které jsou potřebné k vytvoření aplikace od počátku, jsou vidět ve vývojovém diagramu Obr. 16. Ačkoli postup tvorby není jednotný pro všechna použití, pokud vzniká aplikace od počátku, tak může sloužit jako plán postupu. Často se také stává, že je třeba se mezi jednotlivými kroky vracet a přeskakovat podle vývoje aplikace. První jsou vstupní data, to může být již určitá forma zadání, informace nebo náčrty, které jsou obdrženy při specifikaci nebo to také může být představa, kterou tvůrce aplikace má. [18] V této práci jsou vstupními daty informace ze zadání, které je při konzultacích s vedoucím práce upřesňováno za účelem přesněji odpovídajícího řešení, představivost a vyhodnocování ze získaných znalostí.

Se vstupními daty je možné pokračovat volbou 3D modelovacích SW popřípadě rovnou do 3D vývojových aplikací. Úkolem modelovacích SW je hlavně tvorba, úprava a práce na modelech. To umožňují buď manuálním modelováním nebo automatickým, popřípadě lze využít kombinace. Ačkoli modelovací SW umožňuje nějaké animace, jsou tyto možnosti velmi omezené, proto se využívají vývojové aplikace. Modely jsou následně importovány do vývojových aplikací, které jim dávají funkci a zapojují je do scény. Vznikají z toho aplikace, hry a filmy. Modely a aplikace mohou být samostatně prezentovány na platformách jako jsou YouTube, kde je jejich funkce spíše stacionární. Pro využití ve VR je potřeba vzniklou aplikaci převést na platformu v headsetu, aby byl schopný ji zapnout. Těchto platform, na které jsou aplikace a modely převedeny, může být více najednou, záleží na schopnostech exportu daného SW. [18]



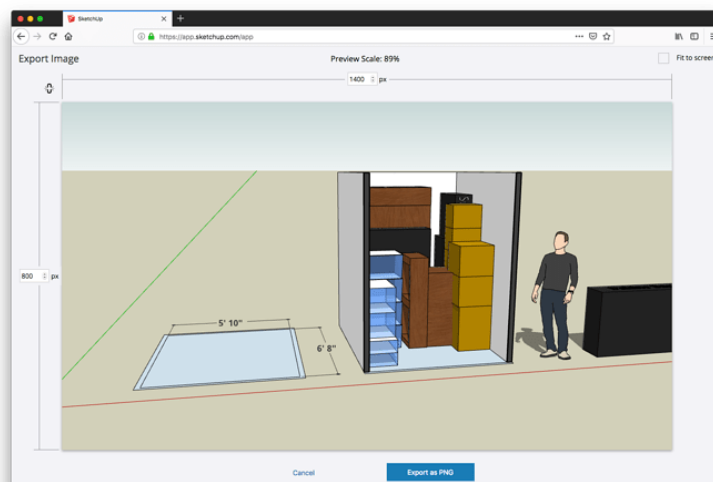
Obr. 16 Vývojový diagram pro 3D VR aplikaci [18]

4.1 Modelovací software

Prostředí a modely jsou základem každé aplikace. Vznikají primárně modelovacích SW, které umožňují tvůrcům jednoduše tvorbu pomocí nástrojů. Tyto nástroje modelovacích SW se mohou lišit podle zaměření, některé průmyslová odvětví jako jsou 3D tisk, strojírenství a zdravotnictví vyžadují tvorbu realistických modelů, zatímco ve filmářském a herním průmyslu se využívají hlavně alternativní styly. Volba SW, se kterým jsou modely tvořeny, také záleží na schopnostech uživatele. Na trhu je možné nalézt velké množství takovýchto modelovacích SW, které odpovídají požadavkům uživatele. [19] Další možností pro tvorbu modelů jsou CAD SW, který však většinou není vhodnou volbou pro aplikaci ve VR (viz 3 Možnosti implementace 3D modelů do VR).

4.1.1 SketchUp

SketchUp si získal v modelářské komunitě jméno svou jednoduchostí, která umožňuje novým uživatelům začít tvořit bez vstupních znalostí. Má velmi přehledné uživatelské prostředí a využívá se v širokém spektru pozic. Hlavní využití se nalézá u přesných modelů, které vznikají od nuly bez předloh, takže se často využívá u modelů pro 3D tisk, strojírenství a design produktů. Orientací na detailní řešení modelů se SketchUp SW primárně zaměřuje na samotné modely, než na prostředí a větší množství objektů. SketchUp komunita zároveň nahrává svoji tvorbu na internet, takže je možné si importovat do vlastního projektu a použít. SketchUp nabízí zdarma tvorbu uživatelům přímo v prohlížeči bez potřeby stahování aplikace. SketchUp dále nabízí placené verze, které se zaměřují na profesionální modelování. [19]



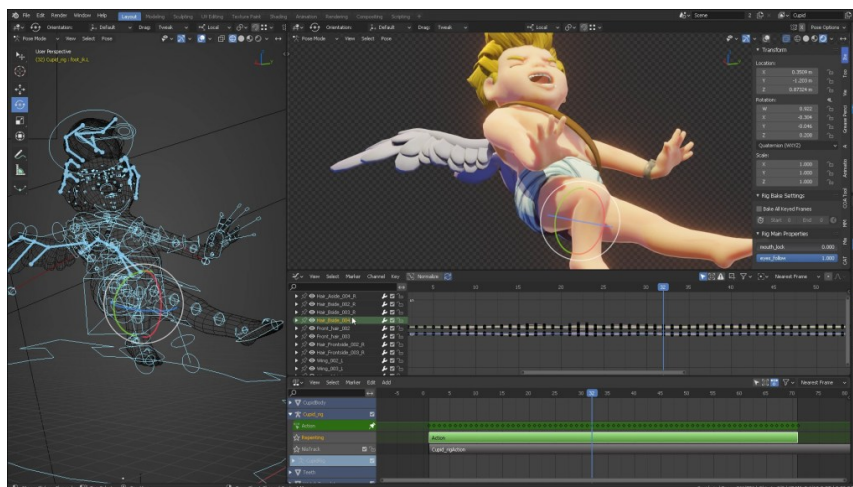
Obr. 17 Ukázka projektu v SketchUp [20]

4.1.2 Blender

Blender je z konkurence modelovacích SW nejvíce adaptabilní. Kromě samotného modelování umožňuje rigging¹⁸, simulace, renderování. Především je považován za nejlepší modelovací SW nejen pro samotnou tvorbu modelů, ale také pro animátory a tvorbu animací, které scéně dávají život. Má velkou a aktivní komunitu, která nahrává výuková videa a modely, čímž umožňuje lepší vstupní podmínky a ušetření času dalším uživatelům. Po vytvoření snímku lze v Blenderu využít i střihu videa. Širokou škálou možností Blender nabízí spolupráci na jednom projektu a tím, že jde o stejný SW, tak je ulehčené sdílení projektů. Blender se neustále rozvíjí,

¹⁸ Rigging – používá se pro animace pohybu postav, objektů a efektů pomocí kostry

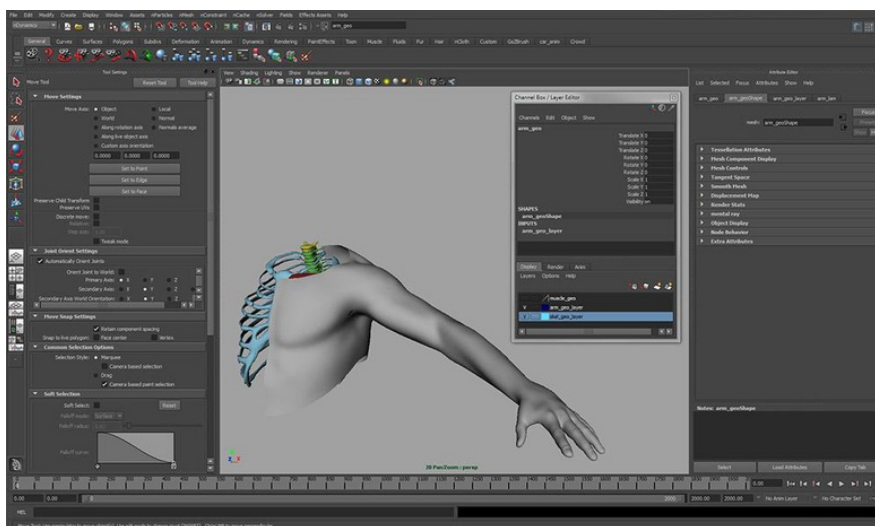
jelikož jde o open source¹⁹, tak neustále vznikají ve spolupráci s uživateli SW nové funkce, například vznik modelů pro 3D tisk. Blender je pro domácí použití zdarma. [19]



Obr. 18 Ukázka projektu v Blender [21]

4.1.3 Autodesk Maya

Autodesk Maya je nejvíce využíván pro profesionální animátory a modeláře, umožňující realistického chování a přesnosti detailu. Jde o první SW, který představil možnost chlupů a vlasů, ale také má přednastavené šablony pro chování vody, ohně a další elementů, čímž se stává často využíván pro filmový průmysl. Přes živé animace a možnosti modelování i renderování má tento SW velmi nepřívětivé uživatelské prostředí, které může být velkou překážkou pro nové uživatele. Vysoce realistickými animacemi a scénami vzniká nárok na výkonný počítač. Autodesk Maya je placený SW, který funguje na předplatném programu, možnostmi je měsíční (235\$), roční (1875\$) a tříroční (5625\$). Čímž se stává hlavně SW pro komerční využívání. [19]

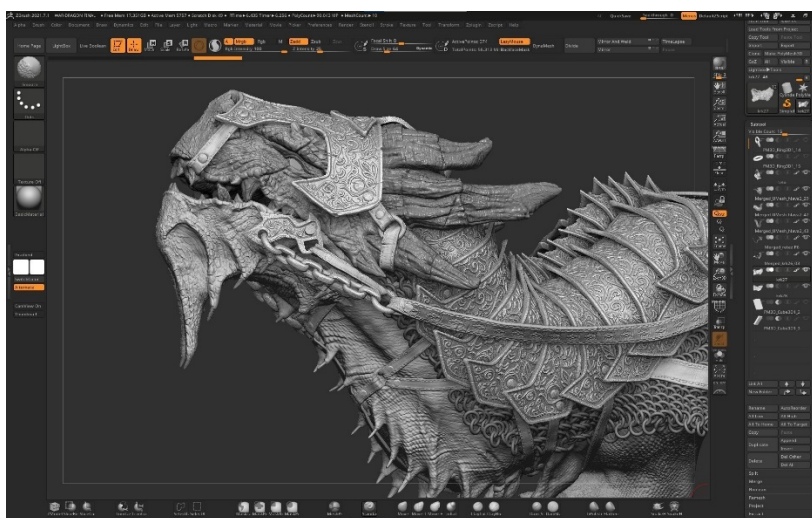


Obr. 19 Ukázka projektu v Autodesk Maya [22]

¹⁹ Open source – uživatelé mají přístup ke zdrojovému kódu a mají možnost jej upravovat

4.1.4 ZBrush

ZBrush využívá především modelování z digitální hlíny. Uživatelé pomocí štětců a dalších nástrojů upravují digitální hlínu, dokud nevymodelují požadovaný tvar. Tento SW se používá především pro detailní úpravy na modelech. Také se dá využít pro znovuvytvoření existujících modelů, které si po vložení do ZBrush a postupnou úpravou hlíny dopracovává. Modely vystupující ze ZBrush, lze přirovnat k sochám, které vyvolávají ručně tvarovaný pocit. To je také důvod proč se stal ZBrush vysoce oblíbeným SW pro 3D tisk. Možnosti upravování modelu lze také využít pomocí grafických tabletů. ZBrush přes možnosti práce na modelech má velmi omezené možnosti animací a je tedy víc orientovaný na samotné modelování. Nevýhodou tohoto SW je náročná orientace v uživatelském rozhraní, které je komplikované. ZBrush je placeným SW s předplatným programem měsíčně (39,95\$), 6 měsíců (179,95\$) a trvalou licenci (895\$). [19]



Obr. 20 Ukázka projektu v ZBrush [23]

4.1.5 Volba modelovacího softwaru

Výběr modelovacího SW a kritérií pro hodnocení je proveden subjektivně autorem podle získaných znalostí k jednotlivým modelovacím SW. Hodnocení je dáno podle školského způsobu známkování. Jednotlivé SW jsou hodnoceny v rozmezí 1-5, kde nejlepší známka je 1 a nejhorší 5. Získané známky se u SW vydělí počtem kritérií a získají výsledný průměr.

Tab. 3 Tabulka porovnání modelovacích softwarů

Volba modelovacího softwaru							
Software	Vstupní znalosti	Vstupní náklady	Modelování	Animace	Komunita a výukové možnosti	Průměr	Umístění
SketchUp	1	1	3	4	2	2,2	2.
Blender	2	1	2	2	1	1,6	1.
Autodesk Maya	5	5	1	1	2	2,8	3.
ZBrush	3	3	3	5	3	3,4	4.

Z Tab. 3 je vidět, že nejlepší hodnocení, a tím nejlepší průměr získal modelovací SW Blender, na druhém místě byl SketchUp, na třetím Autodesk Maya a na čtvrtém ZBrush. Blender získal nejlepší hodnocení, kvůli pokrytí největšího množství funkcí, zároveň jde o SW, který je zdarma a možností lehčího vstupu pro nové uživatele.

4.2 Vývojový software

Volba vývojového SW je pro nový projekt klíčová, protože většina části vývoje aplikace probíhá právě v ní. Nejčastěji se pro tvorbu 3D VR aplikací používají herní enginey, jejichž původním důvodem použití byla tvorba her. Tyto enginey usnadňují vývojářům tvorbu nových produktů předdělanými šablonami, takže si uživatelé nemusí tvořit vlastní SW. Nyní se univerzálnost SW využívá nejen pro hry, ale také aplikace v průmyslu a filmy. Herní engine je celek, který je tvořen více enginey. Pokud by některé nebyly součástí herního engine, musel by si je uživatel sám implementovat, což je náročné. Enginey nejčastěji nahrazují funkce jako jsou zvuky, 3D grafika, fyzika, kolize a animace. [18] [24]

Pro vývoj 3D projektu ve VR existuje velké množství herních engineů, u kterých je hlavní rozdíl v těchto možnostech:

- **Obchod** – Pro vytvoření scény je potřeba mít vhodné objekty. Objekty ve scéně je možné si pomoci modelovacími SW vytvořit. Herní enginey také umožňují tvorbu objektů, ale tento způsob je omezený, protože to není hlavním zaměřením těchto SW. Další možnosti jsou obchody, kde lze najít 2D, 3D objekty, prostředí, zvuky, světla a další potřebné věci pro vytvoření imerzní scény. Do těchto obchodů přispívají uživatelé jednotlivých SW a nacházejí se tam jak placené, tak zdarma předměty, které je možné využít pro vlastní řešení. Nahráním svých předmětů autorem do obchodu umožňuje komukoli využívat jeho materiál a lze je využít i pro komerční aplikace. Větším množstvím předmětů v obchodě se zároveň zvyšuje šance, že některé bude možné použít pro vlastní scénu.
- **Komunita a výukové možnosti** – Jak bylo zmíněno u obchodu, tak se často stává, že otevřením SW pro komunitu uživatelů dochází k rozšiřování možností, které jednotlivé enginey nabízejí. Komunita zároveň může radit na fóru a vytvářet návody pro tvorbu jednotlivých aplikací. Může dávat zpětnou vazbu, podle které se řídí vývojová studia. Samotné enginey vytvářejí výukové kurzy, pro edukaci svých uživatelů, které jsou určené pro různé úrovně znalostí.
- **Programovací jazyk** – Samotná scéna se stacionárními objekty pro průmyslové aplikace je nedostačující, proto se používá programování, které dává scéně život. Programovací jazyků existuje velké množství a nelze vybrat ten nejlepší. Programovací jazyky lze rozdělit podle náročnosti, také se vyskytují vizuální programovací systémy, které umožňují uživateli jednodušší řešení svou přehledností a manipulací.
- **Grafická řešení** – Enginey se orientují na ulehčování možnosti tvorby realistického obsahu a animovaných grafických řešení. To vzniká pomocí světla, objektů, efektů, stínů, zvuků a scén. Více realistické řešení aplikací má zároveň větší nároky na výkon a často je třeba tuto míru omezit.
- **Renderování** – Proces renderování je poslední částí tvorby projektu. Při němž se sjednotí veškeré vytvořené scény a vzniká jednotná aplikace, která je spustitelná na virtuálním zařízení. Rychlost a kvalita renderování se odlišují podle použitých systémů v enginech, které tento proces ovlivňují. [25]

4.2.1 Unity

Nejvíce rozšířenou platformou pro tvorbu 2D a 3D projektů se stal relativně nový herní engine, který byl vydán v roce 2005. Unity bylo použito pro řešení až 60% AR či VR projektů a 50 % her na mobilní telefony. Tento engine si jen za posledních 12 měsíců stáhlo přes 24 milionů uživatelů, kteří pocházeli z různých pozic a zkušeností, od nových uživatelů až po profesionální studia. Unity také nabízí nejširší pokrytí podporovaných platform, celkově jich je 28 nejznámějšími jsou Windows Mixed Reality, Oculus, iOS a Android. Unity se neustále zlepšuje a rozšiřuje, vznikají nové aktualizace a verze za účelem zefektivnění práce v Unity a pokrytí širších možností tvorby. Největším množstvím assetů²⁰ se nachází na Unity Assets Store, kde je přes 50 000 assetů a z toho přes 31 000 3D assetů. Unity je od vzniku otevřený engine, proto vzniklo velké množství článků, návodů a diskusí. Unity nabízí 5500 výukových kurzů. Programovacím jazykem, který Unity využívá je C#, na který už vniklo přes 128 000 vláken na Unity fóru. Z nichž je alespoň 6100 zaměřených na VR. Dalším důvodem, proč je vhodný C# pro tvorbu VR je, že kolem 90 % aplikací vzniklo pomocí tohoto programovacího jazyka, čímž vzniklo víc řešení jednotlivých problémů při tvorbě aplikace. Grafické možnosti v Unity jsou jedny z nejlepších, ale může docházet potřebě úprav v možnostech pro jednotlivé 3D assety. Tento software je pro nekomerční užítky zdarma, pokud však výnosy přesahují 100 000\$ je nutné přejít na roční předplatné, které se liší vydělanou sumou a možnostmi navíc: Plus (399\$), Pro (1800\$) nebo Enterprise (4000\$). [25]



Obr. 21 Ukázka projektu v Unity [26]

4.2.2 Unreal Engine

Druhým nejvíce rozšířeným herním enginem je Unreal Engine. Tento engine vznikl roku 1998 a v průběhu let se stal populární hlavně pro svoje vysoce kvalitní použití ve filmech a hrách na počítači. Engine je možné od roku 2015 získat zadarmo, což způsobilo nárůst popularity a získáním 7 milionu uživatelů. Unreal Engine umožňuje tvořit aplikace na 15 platformách. Ačkoli je Unreal Engine primárně používán pro počítačové hry, studia jej využívají i pro tvorbu

²⁰ Asset - veškeré použitelné zdroje ve VR aplikaci

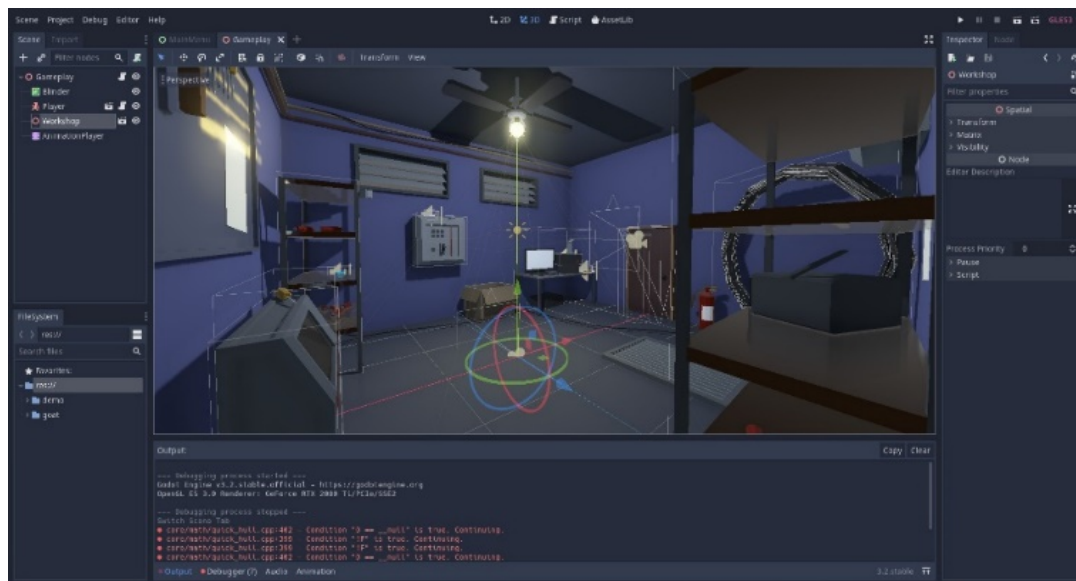
například simulačních aplikací pro VR. Počet assetů, které jsou v obchodu pro Unreal Engine je kolem 10 000. Unreal Engine nabízí přes 2200 výukových kurzů, které uživatele připraví všemi potřebnými znalostmi pro praxi. Programovací jazyk pro Unreal je C++, ke kterému vzniklo na fóru přes 12 000 vláken, z nichž bylo přes 4600 zaměřeno na VR či AR. Přesto, že je velké množství programátorů v C++, tak většina uživatelů využívá vizuální programování. Unreal Engine je tedy rozdělen na dvě části uživatelů, kteří využívají jiný způsob programování. Systém pro vizuální programování Blueprints nabízí lepší řešení pro nové uživatele, kteří nemají s programováním zkušenosti, zatímco programy v C++ nabízí pružnější řešení problémů než to, které umožňuje Blueprints. Co se týká grafické realismu je Unreal Engine oproti konkurenci nejlépe připravený na tvorbu, při které umožňuje plno věcí udělat za vývojáře. Unreal Engine je pro nekomerční používání zdarma, pokud je aplikace monetizována, tak je účtován 5% poplatek ze všech příjmů. [25]



Obr. 22 Ukázka projektu v Unreal Engine [26]

4.2.3 Godot Engine

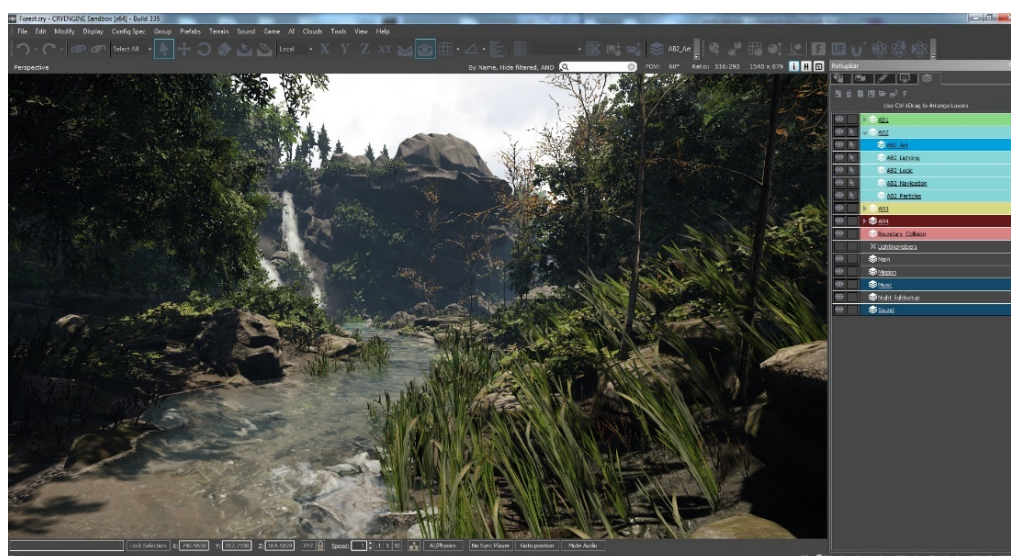
Ačkoli počet uživatelů enginu Godot není takové množství jako u ostatních, umožňuje uživatelům originální možnosti, primárně kvůli tomu, že je open source. Open source znamená, že veškeré používání aplikace je zdarma a komerční zisky nejsou zpoplatňovány. Godot je primárně využíván pro 2D aplikace, ale je stále více využíván i pro 3D. Přes to, že jde o novější herní engine, tak využívá hlavně programování, takže je to ideální SW uživatele, kteří mají zkušenosti s programováním. Godot má originální řešení programovacích jazyků, hlavním jazykem, který se používá je C++, ale pro uživatele, kteří tento jazyk neumí umožňuje programování C#, vizuálního programování a vlastního jazyku GDScript, který vznikl kombinací Pythonu a Lua. Godot má aktivní komunitu, která na fóru pomáhá při dotazech. Aplikace vznikající v tomto enginu je možné tvořit pro všechny hlavní platformy. Počet aktivních uživatelů se odhaduje na 100 000 a komunita uživatelů Godotu napomáhá tvorbě manuálů a řešení problémů. Výjimečný je také v náročnosti na výkon, Godot má velmi malé systémové požadavky, což se následně může odrazit na renderování a kvalitě aplikací v nich. [24]



Obr. 23 Ukázka projektu v Godot Engine [27]

4.2.4 CryEngine

Jde o výkonný neustále rostoucí herní engine, který umožňuje podporu všech základních platform. Je nejvíce známý pro použití v herních studiích k tvorbě kvalitního obsahu. Nabízí velmi jednoduchý editor pro testování vytvářené aplikace, čímž dochází ke zmenšení časové náročnosti. Programovacích jazyků pro CryEngine je více, používá C++, C#, Lua a také umožňuje instalaci doplňku pro vizuální programování pomocí vývojového diagramu. CryEngine nabízí spoustu hotových modelů, které lze využít pro 3D řešení. Čím se liší tento engine proti ostatním je možnost renderu bez připojení internetu, toho se obzvlášť využije při renderování velkých aplikací, které jsou časově náročné a hrozí výpadek připojení. Tento engine využívá menší komunita uživatelů, tudíž je náročnější dohledat řešení v tomto SW. Také má hůř zdokumentované návody než konkurence, což může být pro nové uživatele odrazující. CryEngine je SW, který je pro nekomerční požívání zdarma, po monetizaci aplikace a zisku nad 5000\$ ročně se účtuje poplatek 5 %. [24] [28]



Obr. 24 Ukázka projektu v CryEngine [29]

4.2.5 Volba vývojového softwaru

Výběr modelovacího SW a kritérií pro hodnocení je proveden subjektivně autorem podle získaných znalostí ve 4. části. Hodnocení je dáno podle školního způsobu známkování. Jednotlivé SW jsou hodnoceny v rozmezí 1-5, kde nejlepší známka je 1 a nejhorší 5. Získané známky se u SW vydělí počtem kritérií a získají výsledný průměr.

Tab. 4 Tabulka porovnání vývojových softwarů

Volba vývojového softwaru							
Software	Vstupní znalosti	Vstupní náklady	Komunita a výukové možnosti	Obchod	Programovací jazyk	Průměr	Umístění
Unity	1	1	1	1	2	1,2	1.
Unreal Engine	1	1	3	2	3	2	2.
Godot Engine	3	1	3	5	1	2,6	3.
CryEngine	3	1	4	4	1	2,6	3.

Z Tab. 4 je vidět, že všechny herní engine nabízejí pro nekomerční používání své SW zdarma. Nejlepší hodnocení a tím nejlepší průměr získal herní engine Unity. Na druhém místě se umístil Unreal Engine a na třetím se umístili Godot Engine a CryEngine, které přesto, že měly větší možnosti volby u programovacího jazyka, tak v ostatních kategoriích nebyly v porovnání s konkurencí dost dobré.

5 Blender

Před samotným začátkem tvoření aplikace do VR je nejprve vytvořen obsah, který se následně exportuje do Unity. V této kapitole je popsán způsob od prvního stažení, seznámení s uživatelským prostředím, až po podrobný popis tvorby modelů, se kterými uživatel interaguje. V Blenderu se také využije možnosti tvorby animace. Animací u strojírenských modelů bude rozstřel. Tento typ animace je typický pro strojírenskou praxi a umožňuje uživateli si lépe model prohlížet ve virtuálním prostředí. Vytvořené modely s animacemi lze exportovat do projektu v Unity.

Instalace a stažení Blenderu

Blender nabízí velmi snadný způsob stažení. Přes oficiální stránky Blenderu se na horní liště se vybere tlačítko *Download*, které přeměruje uživatele na možnost výběru svého OS²¹ na zařízení a poté zmáčknout *Download Blender*. Do počítače je stažen *Blender Setup*, pomocí kterého se při následování kroků nainstaluje software Blender do počítače.

5.1 Uživatelské rozhraní Blenderu

V této části je popis základních částí a možností SW Blenderu. Otevřením softwaru Blender v počítači se otevře nový *General* projekt. Uživatelské rozhraní lze rozdělit na 5 částí. Horní lišta, *3D Viewport*, *Outliner*, *Properties* a *Timeline*.

Lišta – V horní liště lze vidět složky s názvy *Layout* a *Modeling*. Jejich funkcí je především přepínat se mezi pracovními prostředí, které slouží pro rychlou změnu scény a orientaci mezi požadovanými akcemi. Při vytvoření nového projektu uživatel vidí právě *Layout* pracovní prostředí. V tomto prostředí se nacházejí nástroje, pomocí kterých si lze pracovní prostředí, ve kterém se uživatel nachází, nastavovat podle potřeby.

3D Viewport – Další částí je pohled na pracovní prostředí nebo také *3D Viewport*, které se skládá z okna s kostkou, levé lišty s nástroji a horní lišty s *Object Mode*. Je zde zobrazen projekt, na kterém se pracuje a kde je možnost přepínání mezi módy²². Ikona s mřížkou a kuličkou *Editor Type* také přepíná pracovní prostředí pro různé potřeby.

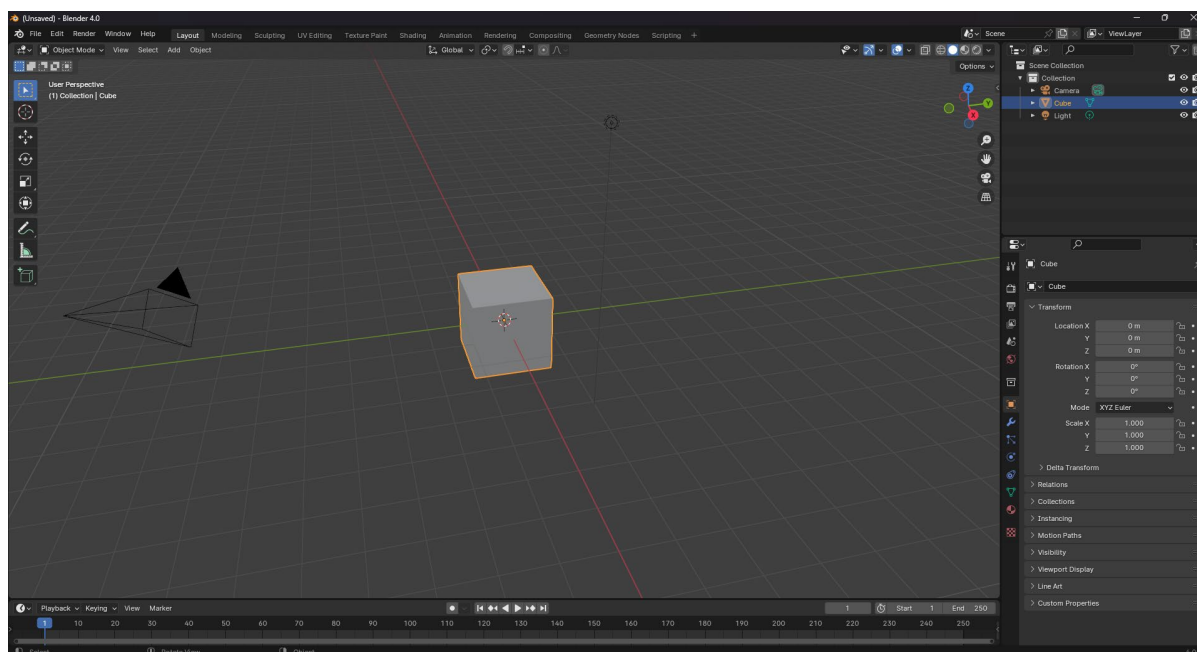
Outliner – V pravé horní části, kde je složka *Scene Collection*, se nachází okno *Outliner*. To má několik funkcí. Především slouží jako seznam vytvořených a existujících objektů v projektu. To je obzvlášť užitečné v projektech s velkým množstvím modelů, kde je náročné se orientovat a manuálně zvolit požadovaný model. Další funkcí je hierarchie, kde se ke každému modelu připisují vlastnosti, které se k němu přidávají například materiál a síť objektu. Lze zde zviditelňovat a skrývat objekty.

Properties – V pravé dolní polovině obrazovky je okno s vlastnostmi. Hlavní funkcí je možnost úpravy scény. Je zde možné měnit vlastnosti modelu jako rozměry, pozice, materiál a vlastnosti ve scéně. Výhodné je především možnost přesných hodnot, které lze zadat například při nastavování rozměrů modelu, které je při alternativě *Scale* nepřesné. Další možností jsou rozšíření, které lze do projektu přidat a tím rozšířit možnosti nástrojů.

²¹ OS – Operační systém, na kterém funguje počítač (Windows, Linux, MacOS)

²² Módy – V Blenderu existuje množství módů, mezi kterými se lze přepnout. Využívají se pro různé způsoby práce s objekty. Často používanými módy jsou *Object Mode*, *Edit Mode*, *Pose Mode* a *Sculpt Mode*.

Timeline – Poslední částí je okno ve spodní části. To se nazývá *Timeline*, neboli časová osa. Jde o okno, které se především používá při tvorbě animací a snímků. Na ose x se nacházejí číselné hodnoty, které označují čas animace. Vytvořením dvou bodů na této ose v různých časech se zapisuje děj, ve kterém zachycují veškeré změny v tomto úseku.



5.2 Modelování v Blenderu

V této práci bylo potřeba nejprve vytvořit model stolu a destičky. Stůl i destička budou využity pro tlačítka, pomocí kterých se bude interagovat s průmyslovými modely. V této práci byly vybrány takové průmyslové modely, které jsou typické pro strojírenskou praxi a je na nich možnost dělat dobře viditelné animace rozpadu. Modelované průmyslové modely jsou:

- Šroub s maticí a podložkou
- Kuličkové ložisko
- Válečkový řetěz
- Větrná elektrárna
- Diferenciál

Stůl

Stůl se skládá z dvou noh a z desky. Deska byla modelována tak, že se v *Object Mode* pomocí funkce *Scale* základní kostky vymodeluje deska, ta má tvar obdélníku. Následně se využije funkce *Loop Cut*, kterou vytvořím nové plochy na desce. Přepnutím do *Edit Mode*²³ se přes *Extrude Region* vysunou nohy stolu. Nohy jsou pomocí *Bevel* neboli sražení sraženy. Nakonec je stolu přidán černý materiál v *Editor Type*.

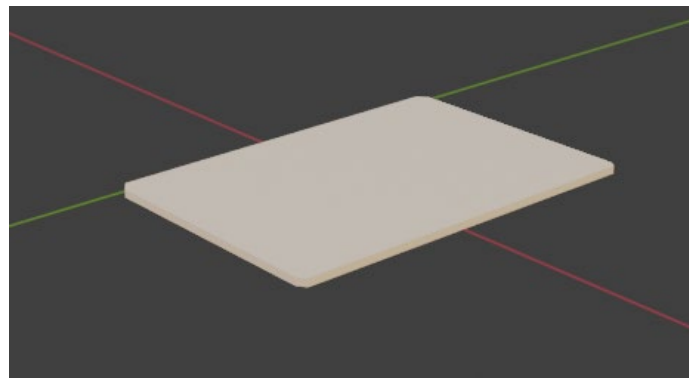
²³ Edit Mode – Tento mód se primárně používá pro úpravu geometrie, například srážení hran, vytahování, tvorbu děr.



Obr. 26 Model stolu

Destička

Tvorba destičky je velmi jednoduchá. Pomocí základní krychle se funkcí *Scale* udělá obdélník, tomu se následně zaoblí hrany *Bevel*, k zaoblení hran je třeba se přepnout do *Edit Mode*.

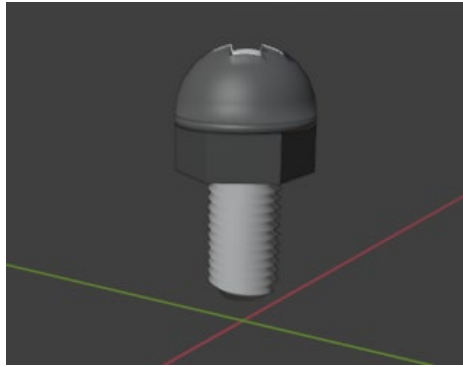


Obr. 27 Model destičky

Šroub s maticí a podložkou

Šroubovice je vytvořena pomocí jedné vertikály, u které se použije *Modifiers* → *Screw*, následně je vytažen a pomocí *Screw* v *Screw Modifiers* se udělá stoupání šroubovice. Následně se přes *Iterations* v *Screw Modifiers* udělá opakované stoupání šroubovice. U rovné plochy se pomocí *Subdivide* vytvoří prostřední vertex a tím následně je získán tvar šroubovice. Výběrem spodních vertexů²⁴ šroubu, vytažením a zarovnáním se vytvoří dolní část šroubu. Horní část se vytvoří pomocí vysunutí ploch a jejich uzavření a následně zaoblení. Šroubovací drážka vzniká pomocí vysunutí ploch v horní části.

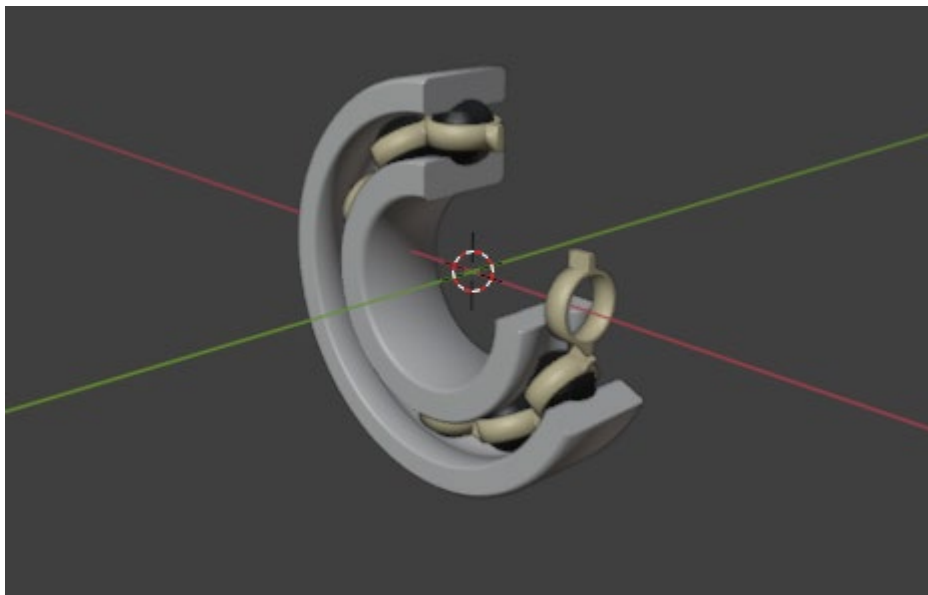
²⁴ Vertex – Používá se v programu Blender jako označení pro vrchol.



Obr. 28 Model šroubu

Kuličkové ložisko

Kuličkové ložisko je tvořeno vnitřním a vnějším kroužkem, kuličky a oddělovače. Vnitřní a vnější kroužek ložiska je vytvořen vysunutím válce v němž se vytvoří kruhová plocha, která je následně odstraněna, kuličky jsou jednoduché objekty pomocí *Mesh* → *UV Sphere*. Přemístěním mezi kroužky je vyresetovaná pozice pro vytvoření kuliček po celém kruhu mezi kroužky. Zkratkou *Ctrl + A* → *Reset All Transforms*. Přidáním *Empty Object*, k němu se vytvoří *Array* → *Add Modifier*, kde se nastaví kulička jako *Empty Anchor*²⁵ nastaví se úhel mezi jednotlivými kuličkami a poté *Count*, pro počet kuliček s tímto úhlem. Pro vytvoření oddělovače je třeba na kuličky vybrat plochy na boku a pomocí *Scale* je vysunout. Stejným způsobem je uděláno spojení mezi kuličkami označením kolmé plochy k vysunutému kroužku a vysutím.

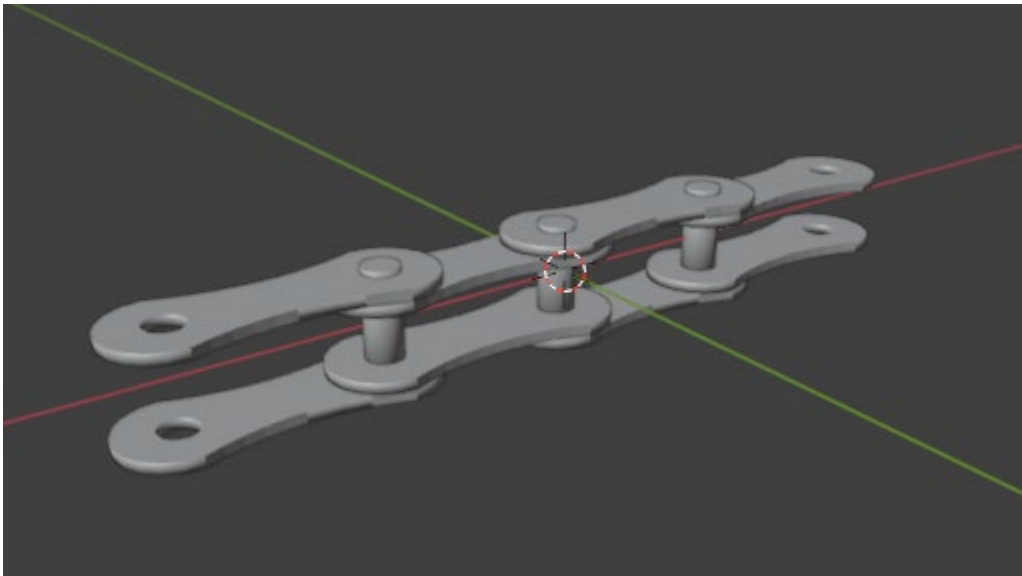


Obr. 29 Model kuličkového ložiska

²⁵ Anchor – V Blenderu se využívá pro označení referenčního bodu ve scéně.

Válečkový řetěz

Válečkový řetěz je vymodelován ze dvou částí. První je čep. Ten se udělá přes kruhový profil *Circle* a vytažením *Extrude*. Druhou částí je vnitřní a vnější destička. Vytažením kruhového profilu *Extrude* do výšky a následným rozdělením na dvě části lze boční body zaoblit a vytvořit tak destičku ve tvaru čísla osm. Díry do destiček jsou vytvořeny novým profilem *Circle* na destičce a odstraněním.



Obr. 30 Model válečkového řetězu

Větrná turbína

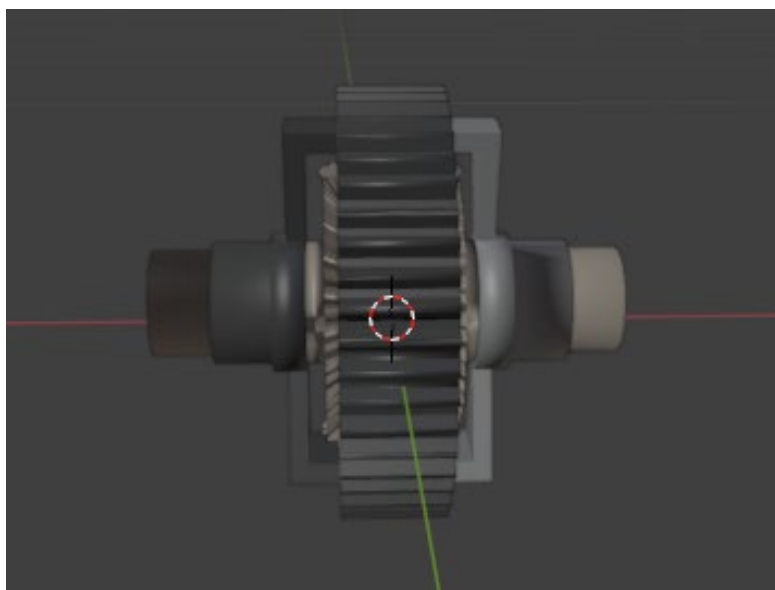
Větrná turbína je tvořena čtyřmi částmi. První je tubus větrné turbíny ten se vytvoří plochou kruhu *Circle* a vytažením *Extrude*. Otočná gondola se modeluje natočením čtvercové plochy *Plane* a vytažením pomocí *Extrude*, následně se zaoblí hrany vytvořeného obdelníku *Bevel*. Další částí je náboj rotoru, ten lze vytvořit opět pomocí kruhového *Circle* profilu a vytažením *Extrude* s následným zaoblením *Bevel* na konci náboje. Poslední částí jsou tvarované listy rotoru. Ze středu rotoru se přes *Extrude* vysune čtvercový profil, tak aby vytvořil dvě části. První slouží jako spojka rotoru a lopatky a druhá je samotná lopatka. V druhé části se udělá několik řezů. Vybráním bodu na stranách obdelníku a zkratkou G se posune bod na požadovanou šířku lopatky. Poté zbývá profil natočit pomocí R. Vybráním celé lopatky a ve vlastnostech se přidáním *Solidify Modifieru* přidá šířka listu. Listy se následně duplikují na počet 3.



Obr. 31 Model větrné turbíny

Diferenciál

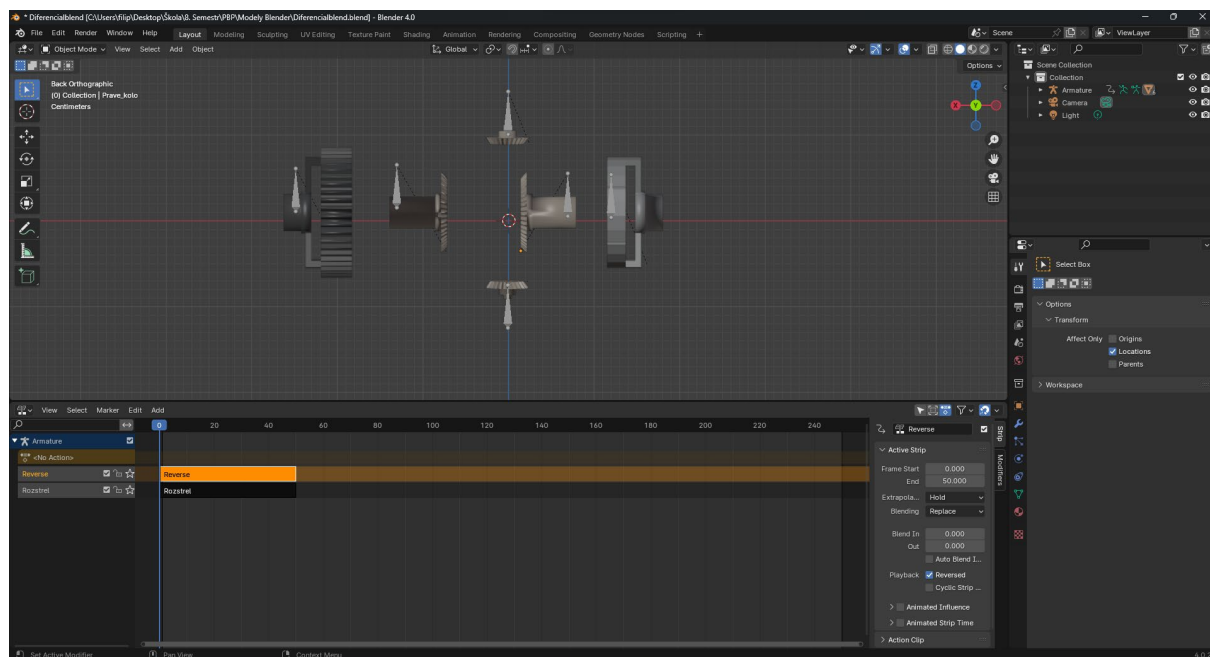
Pro diferenciál se vytvořilo talířové kolo, dvě planetová kola, které mají vstupní hřídele, dvě satelitní kola a klec. Planetová kola a satelitní kola se dělají stejným způsobem. Přidáním kruhu *Shift + A* → *Circle* poté se změni počet hran v *Edit Mode* → *Extrude Along Normals* se vytvoří kruh s rovinami, roviny se následně střídavě vytáhnou a zmenší se délka hrany na konci zubu, čímž se získá tvar zubu. Sražením ploch zubů se vytvoří kuželové kolo. Vstupní hřídel je kruhový profil, který je vytažený. Stejný postup je použit pro zuby talířového kola, kde není sražení zubů. Tvar klece je stejný jako u talířového kola, vytažením kruhu a poté vytvořením ploch na kruhu a vytažením tvarů.



Obr. 32 Model diferenciálu

5.2.1 Animace průmyslových modelů

K hotovým modelům je nyní možné vytvořit animaci rozstřelu. Nejprve je třeba ke každé části modelu přidat armaturu²⁶. Armatura se udělá přes *Shift + A* → *Armature* v *Object Mode*. Následně se pomocí zkratky *Tab* nebo tlačítka pod lištou přepnu do *Edit Mode*. Zde se již přetáhnou armatury na jednotlivé části (viz. Obr. 33). Pro nastavení správných pozic armatur je potřeba použít různé pohledy, protože přesto že v jedné ose armatury mohou být v dobré poloze v jiném pohledu se pozice může výrazně lišit. S kostmi na korektních pozicích se přepne do *Pose Mode*²⁷. Nejprve je označena součást a následně pomocí *Shift + i* armatura k němu. Zkratkou *Ctrl + P* je následně vybrán *Bone*, tím se přiřadí kost k součásti, stejným způsobem se to udělá i u ostatních součástí. S přiřazenými armatury je možné začít tvořit animaci. Na dolní liště je pro zachycení animace použito okno *Nonlinear Animation* v počáteční pozici, tedy v 0 je model sestaven dohromady v celku. Zkratkou *I* a zvolením *Position, Scale, Rotation* se vytvoří stopa v nultém bodě. Nyní se zvolí délka přechodu čím delší je chtěná animace tím vyšší hodnota se volí. Pro tuto aplikaci byly voleny délky animací na hodnotu 50. Na hodnotě 50 se pak armatury společně se součástmi roztáhnou do chtěné pozice, aby bylo přehledně vidět z jakých součástí je model tvořen. Opětovným stlačením zkratky *I* a zvolením *Position, Scale, Rotation* se zapíše stopa na hodnotě 50. Akci je nyní možné pomocí tlačítka *Push Down Action* přesunout pro editaci chtěné animace. Animace je přejmenována na „Rozstrel“ a zpáteční animace se vytvoří pomocí zaškrtnutí *Reversed* a pojmenována na „Reverse“. Tento postup je použit pro všechny průmyslové modely (viz. Obr. 34).



Obr. 33 Ukázka projektu po rozstřelu diferenciálu

²⁶ Armatura – Je nástroj, který se používá při tvorbě animací. Umožňuje manipulaci s kostmi (Bones), které se používají pro rozpohybování objektu.

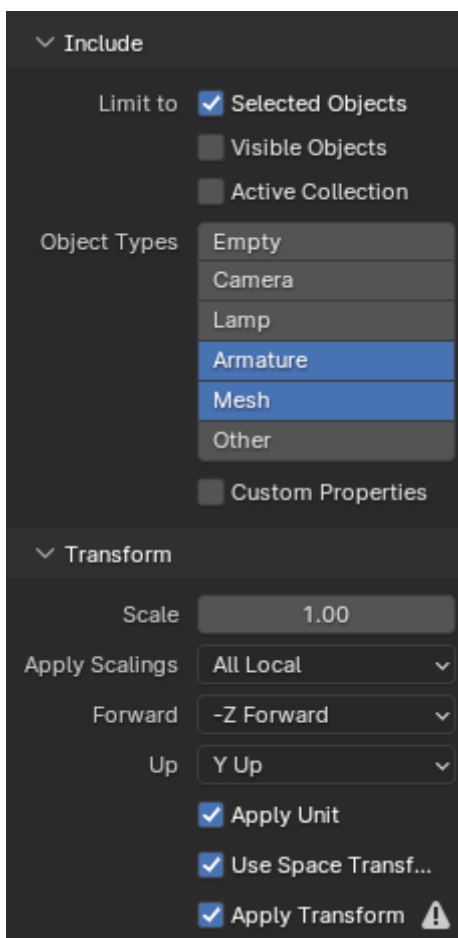
²⁷ Pose Mode – Tento mód se používá pro tvorbu animace pomocí přesunu kostí, zachycují se zde například rotace, poloha a pozice modelu.



Obr. 34 Modely po rozstřelu

5.2.2 Export modelů a animací

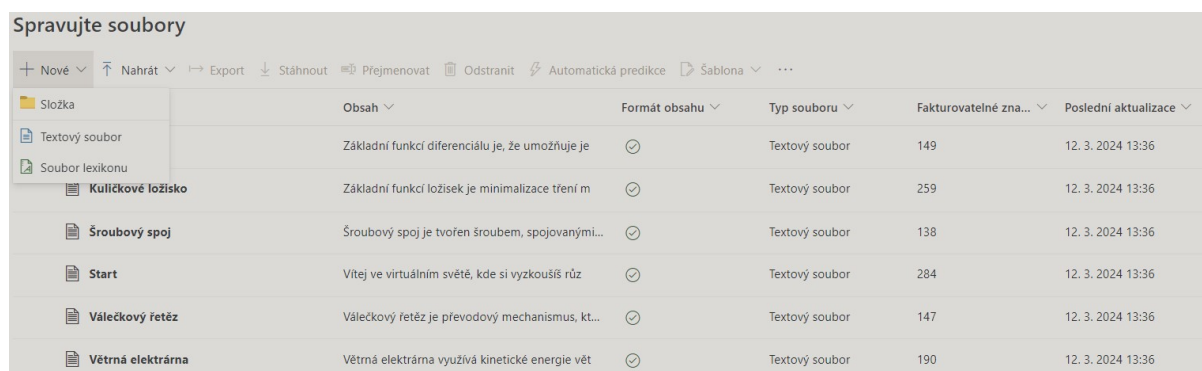
Vytvořené modely spolu s animacemi rozstřelu lze následně exportovat. Blender nabízí export do širokého množství formátů. Nejčastěji používaným formátem pro export do herních enginů je .fbx, neboli *FilmbOX*, který spolehlivě exportuje materiály, animace a textury modelu v jednom balíčku. Prvně je třeba se přepnout u modelu do *Object Mode* a následně označit veškeré armatury a modely viz. obr. dole. Následně se v horní liště vybere *File* → *Export*, kde se vybere formát .fbx. Pro exportování již dříve označených modelů a kostí se vybere možnost *Selected Objects*, exportovat stačí *Armature*, která se využívá pro kosti u animací a *Mesh* pro exportování geometrie modelu. Po zmáčknutí tlačítka *Export FBX* se vytvoří jediný soubor obsahující potřebný model s animacemi. Stejným způsobem byly exportovány ostatní modely (viz. Obr. 35).



Obr. 35 Ukázka vybraných možností při exportu z Blenderu

6 Doprovodný scénář aplikace

Pro VR aplikaci s průmyslovými modely je vhodné udělat alespoň základní seznámení uživatele jak možnostmi aplikace, tak základní informace k modelům. Způsoby podání doprovodných informací uživateli se nejčastěji vyskytují dva. První možností je text, který se může vyskytovat na panelech jako je na obrázcích v místnosti (viz. Obr. v místnosti), nebo volně v prostředí. Další možností, která byla aplikována do této práce je doprovodné informace podávané pomocí zvuku. Unity podporuje audio soubory třetích stran. Pro vytvoření namluvených informací k modelům byl použit Microsoft Azure AI, konkrétně pro převedení písma na slovo se používá *Text to Speech*. Microsoft tuto možnost nabízí zdarma. Nový soubor se vytvoří přes *Nové* → *Textový soubor* (viz. Obr. 36). Do prázdného textového souboru se následně napíše požadovaný text.



Obr. 36 Speech Studio

Úvod do aplikace

„Vítej ve virtuálním světě, kde si vyzkoušíš různé způsoby interakcí s průmyslovými modely. Po levé ruce se nachází tabulka modelů. Výběrem modelu prstem se zobrazí před tebou. Následně s modelem můžeš pomocí tlačítek na stole interagovat. Modelům můžeš měnit velikost, provádět rotaci a udělat rozstřel.“

Při vybrání průmyslového modelu v tabulce se spustí audio soubor s charakteristickým popisem modelu:

Šroub s maticí a podložkou

„Hlavním úkolem šroubu s maticí a podložkou je vytvořit šroubový spoj. K tomu je potřeba navíc spojovaný materiál, který se tímto spojem upevní. Funkce šroubu je přenos tahu a tlaku ve spoji. Matice se používá pro utažení spoje. Podložka se dává pod hlavu šroubu za účelem lepší dotykové plochy a menšího opotřebení ploch. Jde o pevný rozebíratelný spoj.“

Kuličkové ložisko

„Základní funkcí ložisek je minimalizace tření mezi rotujícími součástmi. Toto zatížení je přenášeno kuličkami, které se odvalují po dráze vnitřního a vnějšího kroužku. Oddělovač, který je mezi jednotlivými kuličkami, má funkci rovnoměrného zatížení celého ložiska.“

Válečkový řetěz

„Válečkový řetěz je převodový mechanismus, který je možné vidět například u kola. Řetěz je tvořen vnitřní a vnější destičkou, které jsou spojeny čepy.“

Větrná elektrárna

„Větrná elektrárna využívá kinetické energie větru. Ten naráží na tvarované lopatky, které rotací v rotoru vytvářejí mechanickou energii. Tu následně generátor přemění na elektrickou energii.“

Diferenciál

„Základní funkcí diferenciálu je, že umožňuje jednotlivým kolům vozidla na stejné nápravě mít odlišné rychlosti. Toho se využívá hlavně při zatáčkách.“

Export souboru

Po vytvoření souborů s popisem každého modelu byly následně exportovány pomocí tlačítka *Export* → *Exportovat na místní disk* → *Exportovat*. Při exportování je možnost volby formátu zvukového souboru a kvality zvuku. V této aplikaci byl použit formát 24k, .wav. Tímto způsobem byly vytvořeny zvukové soubory.

7 Unity

Po vytvoření obsahu aplikace, modelů a animací s doprovodným zvukovým souborem, se kterými se bude následně interagovat, je možné v diagramu z Obr. 16 postupovat na herní engine, kterým byl zvolen Unity. Samotná tvorba aplikace byla rozdělena na dvě části. V té první je třeba připravit projekt pro následnou tvorbu aplikace. V té druhé se již začne programovat funkcionality aplikace.

7.1 Příprava projektu v Unity

Prvním krokem je stažení Unity. Po úspěšném stažení je vhodné se seznámit s jeho základními možnostmi ovládání a funkcemi, které tento software nabízí. Při zahájení nového projektu je nezbytné provést určité úpravy v nastavení, aby byla aplikace kompatibilní s požadovaným zařízením Oculus. Teprve poté lze přejít k úpravám ve scéně a tvorbě funkcionality. Postup těchto kroků je podrobně popsán v následujících kapitolách.

7.1.1 Instalace a stažení Unity

Instalace SW se provádí z oficiálních stránek Unity. Tam se vybere tlačítko *Download*, pro OS počítače a následně přes stažený *Unity Setup* se nainstaluje program. Při instalaci byl vybrán *Developer tool Microsoft Visual Studio Community 2022*, to je programovací software s jazykem C#, který se v Unity využívá pro tvorbu skriptů. Následně byla vybrána *Platform Android Build Support*, která se používá při tvorbě aplikací pro androidové OS, což Oculus Quest 2 používá. Do počítače je stažen *Unity Hub*, přes který se přepíná mezi jednotlivými projekty a nainstalovanými verzemi Unity. V *Unity Hub* je také *Learn*, což je školící program od Unity, ve kterém se lze učit základy práce se softwarem a skriptování. Složka *Installs* obsahuje aktuálně stažené verze Unity v počítači a *Community* veškeré novinky a odkazy na komunitní centra. Pro tuto práci byl zvolen při tvorbě nového projektu *URP (Universal Render Pipeline)*. Ten se při vývoji VR aplikací volí hlavně za účelem přednastaveného projektu se zaměřením na optimalizaci výkonu. Verze Unity byla pro tuto práci volena Unity 2021.3.24f, tato verze má zkratku *LTS (Long-term Support)*. To znamená, že Unity tuto verzi doporučuje pro tvorbu nových projektů, kvůli neporuchovosti.

7.1.2 Uživatelské rozhraní Unity

Otevřením projektu *URP* se otevře nová scéna (viz. Obr. 37), do které je tvořena aplikace. Ve větších aplikacích se využívá více scén, mezi kterými se lze naprogramováním přepínat.

Lišta – Horní lišta v novém projektu slouží pro rychle přístupné umístění možností, které lze jinak tvořit jen v určitých částech projektu. Lze zde tvořit scény, importovat nástroje, exportovat projekt atd.

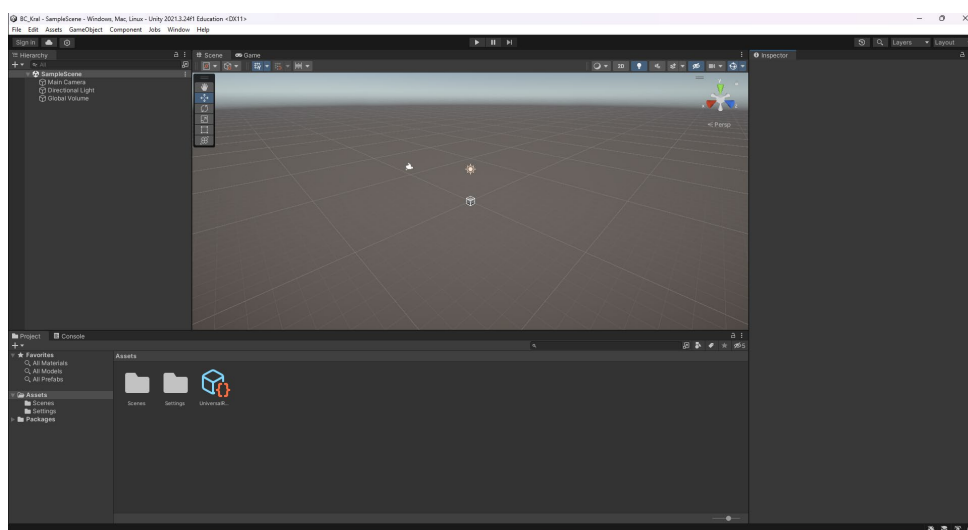
Scene/Game Viewer – V této části je viditelný veškerý tvořený obsah na scéně projektu. Toto okno uprostřed se po spuštění aplikace přepíná ze *Scene* na *Game*, kde je již aktivní scéna a je zde vidět co se v aplikaci děje. V levé horní liště jsou umístěné nástroje pro pohyb po scéně, rotaci, změnu rozměrů. V pravém horním rohu je jakýsi kompas, který se pohybuje spolu s pohledem uživatele. Kompas také umožňuje při kliknutí na jednu z os měnit pohled. Další funkcí je přepínání mezi 2D a 3D, vypínání a zapínání zvuků a kontrolky.

Hierarchy – Podobně jako v Blenderu je i zde určitá hierarchie projektu. V okně se vyskytují veškeré objekty, které byly přidány do scény. Je možné zde pracovat se vztahem *Parent* a *Child*, které má praktické využití. Využívá se například když je třeba manipulovat s více objekty najednou. Tyto objekty neboli *Children* se v hierarchii dají pod *Parent* a stačí následně pouze

pracovat s *Parentem*. Také se využívá pro orientaci ve scéně, která je obsáhlejší. Stačí označit v hierarchii chtěnou položku a dát zkratku *F* pro *Focus* a kamera se v projektu přenesne na chtěnou položku. Vybráním položky v hierarchii lze následně měnit její vlastnosti v *Inspectoru*.

Inspector – Při vybrání položky v projektu se v inspektoru ukážou veškeré vlastnosti položky v projektu. To znamená, že se lze dozvědět polohu, vrstvu, jméno položky a další. V okně je také možnost *Add Component*. Rozkliknutím dostane uživatel možnost rozšiřovat vlastnosti položky, ať už jde o fyzické vlastnosti, zvuk, efekty nebo také skripty, pomocí kterých se programováním přidají další funkce v projektu.

Project/Console Window – Dolní okno je tvořeno dvěma částmi. První částí je *Project*, to je úložiště všech balíčků a assetů přidaných do projektu jako zvukové stopy a modely. Jsou zde všechny položky, které jsou v hierarchii, ale navíc i ty, které ještě nejsou použity. Druhá část je *Console*, ta má funkci především pro zobrazování zpráv a ohlašování poruch aplikace.



Obr. 37 Uživatelské rozhraní Unity

7.1.3 Nastavení projektu

Po vytvoření nového projektu je provedena změna nastavení projektu. Umožní se tím testování aplikace při tvorbě, zároveň by jinak nebylo možné aplikaci na Oculus Quest 2 spustit. Dalším důvodem je optimalizace výkonu. V nastavení jsou možnosti, které se používají pro optimalizaci výkonu, což je pro standalone headset vhodné. Nastavení scény je vhodné před začátkem úprav, protože by jinak kvůli množství změn v projektu a novým souborům vzniklo časově náročné přenastavení projektu.

Build Settings – Nejprve je potřeba změnit platformu, pro kterou je aplikace vytvářena. V horní liště přes *File* se vybere *Build Settings* tam se vybere požadovaná platforma na Android a spustí se *Switch Platform*. Následně se změní *Texture Compression* na *ASTC (Adaptive Scalable Texture Compression)*, tato možnost se volí za snížení velikostí vykreslovaných textur, čímž se dosáhne vyšší plynulosti aplikace a nižší spotřeby baterie.

Player Settings – Z *Build Settings* se následně udělají změny v *Player Settings*. Nejprve je potřeba toto ovládání nainstalovat. To se provede pomocí *Install XR Plug-in Management* poté se vybere ikona androidu a zvolí se Oculus, tím se provede instalace balíčku s pluginy pro Oculus headset. V něm se vybere *Stereo Rendering Mode* na *Multiview*, tím je uspořen výkon, protože se vykresluje scéna pro obě oči současně, *Low Overhead Mode* a *Phase Sync* místo

Optimize Buffer Discards a *Symmetric Projection*. V *Player* se *Configuration* přepne na *IL2CPP* (*Intermediate Language to C++*). To způsobí kompilaci skriptů C# a převedením do C++, který je pro platformu Android vhodný. Touto změnou se otevře možnost *Target Architectures* změnit na *ARM64*, která se využívá u androidových 64bitových zařízení. Odškrne se *Auto Graphics API* a *OpenGL: Profiler GPU Recorders*. Zaškrtně se *Use incremental GC* a místo *ARM v7* se zvolí *ARM64*.

7.1.4 Prostředí aplikace

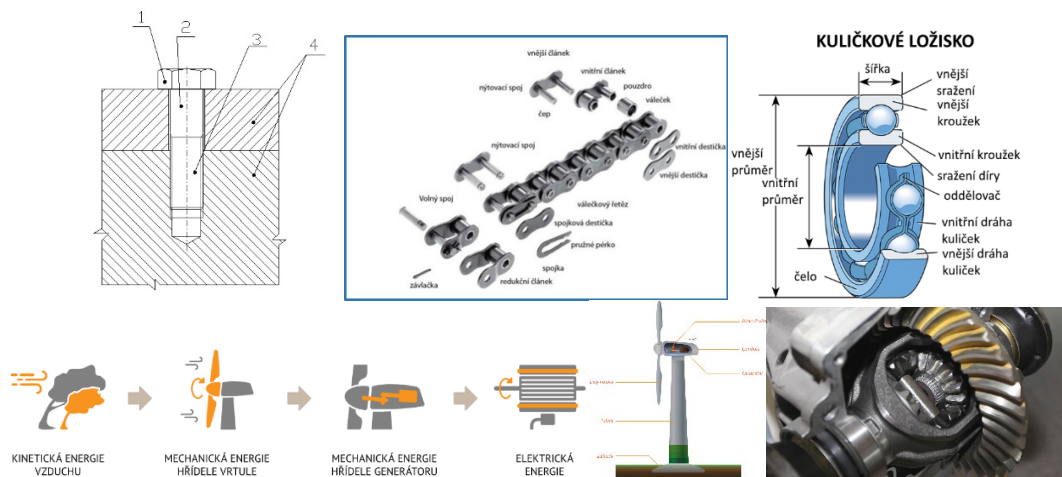
Prostředí ve VR je důležité především pro účel autentičnosti. Prostředí musí odpovídat účelu aplikace a důležitou součástí prostředí ve VR je osvětlení, které je potřebné pro dobrou viditelnost textů a modelů. Prostředí použité do této aplikace bylo nainportováno do projektu z *Asset Store* a poté upraveno. Do *Asset Store* lze přejít v Unity pomocí lišty *Window* → *Asset Store* anebo přes webové stránky <https://assetstore.unity.com/>, odkud bylo použito prostředí *Sci-Fi VR Lab*. Na požadovaném assetu se vybere *Add to My Assets*, čímž se asset přiřadí k přihlášenému účtu v obchodě. Zároveň se zobrazí možnost *Open in Unity*, čímž se asset načte do otevřené scény. Nejdříve se zmáčkne *Download* pro stažení do projektu a poté *Import* pro nahrání do projektu. V další možnosti se ukáží veškeré položky, které se nahrají do projektu, vybráním požadovaných položek a potvrzením tlačítkem *Import* se nahraje složka s prostředím do *Project Window*, ze které je následně nahrána do projektu (viz. Obr. 38).

V prostředí bylo provedeno několik úprav. Byl odstraněn veškerý nábytek a všechny místnosti mimo hlavní místnost. Následně byla odebrány světla, které vytvářela příliš osvětlené a špatně čitelné prostředí.



Obr. 38 Prostředí aplikace

Každému modelu v aplikaci je přidán obrázek na zdi s cílem poskytnout uživatelům lepší představu o struktuře modelu, který si mohou prohlížet v místnosti (viz. Obr. 39).



Obr. 39 Obrázky v prostředí [30]²⁸

7.2 Funkcionalita aplikace

Projekt je připravený pro tvorbu funkcionality. V této druhé části je provedena tvorba interakcí mezi veškerými vytvořenými modely, animacemi k nim a zvukovým souborům. Pro tuto aplikaci to především znamená pomocí skriptů vytvořit interakci mezi tlačítky a animacemi modelů.

7.2.1 Tvorba skriptů

Pro modely bylo nejprve vytvořeno místo kde se budou při importování zobrazovat. K tomu se použije prázdný objekt, který je vytvořen v hierarchii *Create Empty* a se změnou názvu na *ModelSelector*. Tento prázdný objekt je umístěn naproti místu, kde se uživatel objeví při spuštění aplikace. Pozice je vybrána tak, aby zhruba odpovídala úrovni očí. Následně mu byl přiřazen skript *ModelSelector*. Pro tvorbu nového skriptu se zmáčkne pravým tlačítkem v *Project Window* → *Create* → *C# Script*. V prvním řádku skriptu je vytvořeno pole objektů, jelikož je to veřejná metoda, tak je možné přes vlastnosti objektu rozšířit pole na 5, tedy počet vymodelovaných modelů a přetáhnou modely z assetů do pole. V druhém řádku je postup stejný, ale pole je pro zvukové soubory k jednotlivým modelům. Následně bylo vybráno místo, kde se modely zobrazí. Pro to byl vytvořený další prázdný objekt *SpawnPoint*, který je využitý pro modely jako místo, kde se budou při vyvolání zobrazovat. Při importování nového modelu místo původního se musí ten původní nahradit. Pro identifikaci původního modelu je *GameObject oldObj*.

```
public GameObject[] _genObj;

public AudioClip[] _audioObj;

public Transform spawnPoint;

public GameObject oldObj;
```

Obr. 40 Skript pro tabulky a spawn

²⁸ [30] – Pro tuto skupinu obrázků bylo použito jednotné označení [30], které je v seznamu použitých zdrojů jednotlivě ocitováno [30] - [34].

Metoda *GenerateObj(int i)* pracuje s parametrem *i* a celými čísly. Podmínka, kterou obsahuje funguje tak, že pokud není *oldObj == null*, tak je *oldObj* zničen. To znamená, že vybráním nového objektu se stará zničí a naimportuje se nový *_genObj[i]* z pole objektů a vytvoří se pod rodiče *spawnPoint*, vůči kterému má nulovou rotaci a posuv. Poté je vytvořena proměnná *defScale*, která nastaví měřítko objektu ve všech osách na hodnotu *x*. Z tabulky zvuků se nastaví zvukový soubor zvoleného modelu *_audioObj[i]* a následně se přehraje *this.GetComponent<AudioSource>().Play()*.

```
public void GenerateObj(int i)
{
    if (oldObj != null)
    {
        Destroy(oldObj);
    }

    oldObj = Instantiate(_genObj[i]);
    oldObj.transform.parent = spawnPoint;
    oldObj.transform.localPosition = Vector3.zero;
    oldObj.transform.localEulerAngles = Vector3.zero;
    defScale = oldObj.transform.localScale.x;
    this.GetComponent<AudioSource>().clip = _audioObj[i];
    this.GetComponent<AudioSource>().Play();
}
```

Obr. 41 Skript metody *GenerateObj(int i)*

V této metodě se prováděla rotace kolem os *x* a *y*. Obě metody mají stejný princip fungování, jedinou změnou byla osa rotace. Pomocí podmínky *if* se zkontroluje, zda *oldObj* není *null*, pokud není, tak se provede rotace kolem nenulové osy o hodnotu *i* pomocí komponenty *transform*, která se používá pro pohyb a *EulerAngles*, která se používá při změně úhlu. Jelikož jsou metody *public*, tedy veřejné, tak se hodnota rychlosti rotace nastavuje v inspektoru objektu, na kterém tento skript je přiřazen.

```
public void RotationObjx(float i)
{
    if (oldObj != null)
    {
        oldObj.transform.localEulerAngles += new Vector3(i, 0f, 0f);
    }
}
public void RotationObjy(float i)
{
    if (oldObj != null)
    {
        oldObj.transform.localEulerAngles += new Vector3(0f, i, 0f);
    }
}
```

Obr. 42 Skript metod *RotationObjx/y(float i)*

Další metodou je *ScaleObj*, která je použita pro změnu rozměrů. Opět je zde podmínka, která kontroluje, zda *oldObj* není nulový, pokud je splněna podmínka, tak se vynásobí hodnota současného měřítka o hodnotu *i*. Poté je další podmínka, ta je vytvořena, aby nebyla možnost objekt nekonečně zvětšovat. Pokud je nová hodnota měřítka v rozmezí mezi $defScale / 2$ a $defScale * 2$, kde *defScale* je původní měřítko objektu, změní se měřítko objektu podle

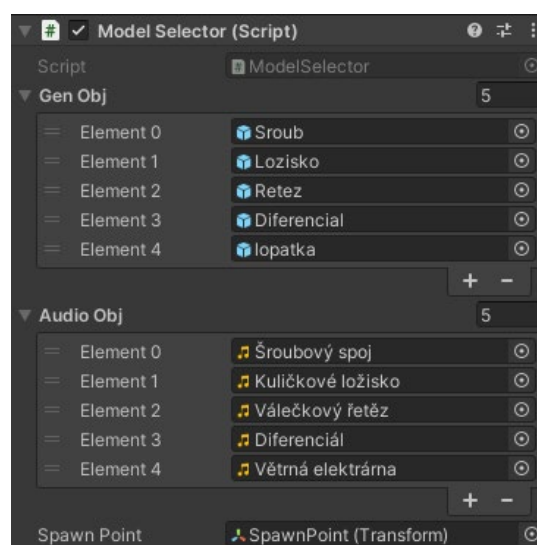
vypočítané hodnoty x . Další podmínka *Else if* se spustí pouze v případě, že hodnota překročila meze $defScale / 2$ a $defScale * 2$ a nelze model dál zvětšovat/zmenšovat, záleží podle toho, která mez se překračuje.

```
public void ScaleObj(float i)
{
    if (oldObj != null)
    {
        float x = oldObj.transform.localScale.x * i;

        if (oldObj.transform.localScale.x <= defScale * 2 && x > 0)
        {
            oldObj.transform.localScale += new Vector3(x, x, x);
        }
        else if (oldObj.transform.localScale.x >= defScale / 2 && x < 0)
        {
            oldObj.transform.localScale += new Vector3(x, x, x);
        }
    }
}
```

Obr. 43 Skript metody ScaleObj(float i)

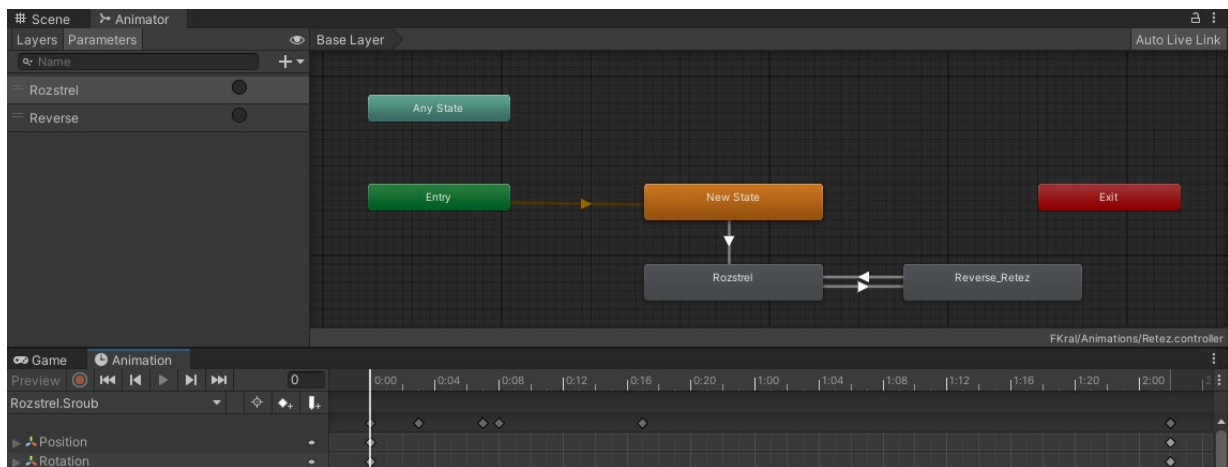
S vytvořeným skriptem *ModelSelector* je možné jej použít na objekt *ModelSelector*. Následně do tabulek přetáhnout modely a zvukové soubory z *Project Window* a *SpawnPoint*. Pro Import modelů a manipulaci s modelem byla použita tlačítka, které jsou již předpřipravená v balíčku Oculus v ukázkových scénách. Pro modely byla použita tabulka, ve které byly změněny názvy a přidány pro jednotlivá tlačítka číselnou hodnotu 0-5 (viz. Obr. 44), čímž se vybere model z tabulky v této pozici a naimportuje. Dále bylo nastaveno, že při zmáčknutí tlačítka se pro *Runtime Only* přidá skript *ModelSelector* → *ModelSelector.RotationObjx/y* a pro tlačítka pro změnu velikostí *ModelSelector.ScaleObj*. U jednotlivých tlačítek pro rotaci a změnu rozměrů bylo potřeba dát číselnou hodnotu i , která byla použita ve skriptech. Pro kladné rotace a zvětšování velikost kladná hodnota a pro opačnou rotaci a zmenšování velikosti záporná.



Obr. 44 Objekt se skriptem Model Selector

7.2.2 Spouštění animace

Animace jsou po přidání do Unity v jedné složce spolu s modelem a materiály jako *Animation Clip*. Animace rozstřelu a sloučení modelu je spustitelná na jednom tlačítku, veškeré práce s animacemi se dělají v *Animator*. V *Parameters* je třeba vytvořit dva parametry (Rozstrel a Reverse), které jsou volány ve skriptu pro animaci (viz. Obr. 46, Obr. 47, Obr. 48). Oba parametry jsou volány pomocí *Trigger*, který se používá pro spouštění animace pomocí spouštěče v této aplikaci tlačítka. Spuštěním animace, tedy stlačením tlačítka pro rozstřel, se spustí *Entry*, což je vstupní stav. Poté se přepne do *New State*, který nejprve navazuje na *Empty State* Rozstrel, která obsahuje rozpad modelu a poté je navázána na opačnou animaci. Pro přechod mezi Rozpad a Reverse je nutné přidat *Conditions*. Tyto podmínky určují, jaký parametr a tedy animace se spustí. Stejné nastavení animací bylo nastaveno u všech modelů.



Obr. 45 Animator Unity

Nyní je možné vytvořit skript, který se použije na tlačítka pro spouštění animace. Je potřeba, aby ve skriptu bylo možné přepínat mezi animací rozpadu a složení. Skript je pojmenován *AnimateObject*. Byla vytvořena dvě veřejná pole *_reverse* pro sloučení modelu a *_activeAnim* při spuštěnou animaci. Tato políčka mohou být, buďto zaškrtnutá nebo ne, tedy *true* nebo *false*. A *_modelSelector*, která je instance třídy *ModelSelector*.

```
public bool _reverse;  
public bool _activeAnim;  
  
public ModelSelector _modelSelector;
```

Obr. 46 Skript s poli pro animace

Po spuštění animace Rozstrel/Reverse se zavolá metoda *StartRozpadReverse()*. Ta se nejprve podmínkou ověří, jestli není aktivní animace *!_activeAnim* a jestli je přítomný model, u kterého se má spouštět animace *_modelSelector.oldObj != null*. Splněním těchto podmínek následuje další podmínka, která říká, že pokud je políčko *_reverse = false*, tak se bude zapínat animace rozstřelu, následně se přepne políčko na *true* a spustí se animace pro rozstřel. Naopak pokud je již políčko *_reverse = true* a opět se aktivuje, tak se přepne zpět na *false* a spustí animaci *Reverse*.


```
public void StartRozpadReverse()
{
    if(!_activeAnim && _modelSelector.oldObj != null)
    {
        if(!_reverse)
        {
            _reverse = true;
            StartCoroutine(StartRozpadReverse_cor("Rozstrel"));
        }
        else
        {
            _reverse = false;
            StartCoroutine(StartRozpadReverse_cor("Reverse"));
        }
    }
}
```

Obr. 47 Skript s metodou StartRozpadReverse()

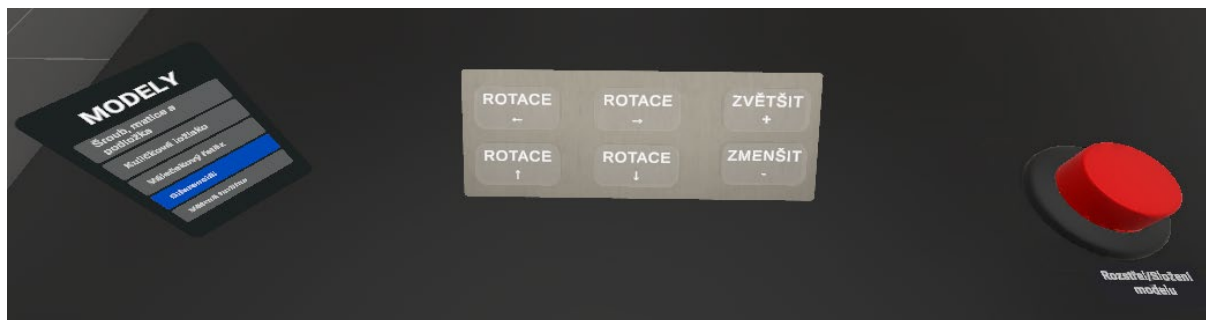
Tato coroutine se používá pro spuštění triggeru animace. Nejprve nastaví aktivní animaci na pravdivou `_activeAnim = true`, tím se dosáhne toho, že nelze provést jinou animaci, dokud ta první je aktivní. Nejprve nastaví `_activeAnim` na `true`, aby zabránila spuštění další animace, dokud tato coroutine neskončí. Poté spustí animaci s triggerem specifikovaným v parametru `animTrigger` pro objekt `_modelSelector.oldObj`. Počká 2.5 sekundy pomocí `yield return new WaitForSeconds(2.5f)`, aby se animace mohla přehrát. Nakonec nastaví `_activeAnim` zpět na `false`, což umožní spuštění další animace.

```
IEnumerator StartRozpadReverse_cor(string animTrigger)
{
    _activeAnim = true;

    _modelSelector.oldObj.GetComponentInChildren<Animator>().SetTrigger(animTrigger);
    yield return new WaitForSeconds(2.5f);
    _activeAnim = false;
}
```

Obr. 48 Skript s coroutine

Stejně jako tlačítka pro rotaci, změnu rozměru a tabulku s modely, tak tlačítko pro vyvolávání bylo importováno z Oculus scény. Bylo třeba k němu přidat funkci při stlačení Runtime Only → `AnimateObject.StartRozpadReverse`. Následně je skript `AnimateObject` přidán na tlačítko `ButtonRozstrel` a do pole `Model Selector` se přidá již vytvořený skript `ModelSelector`. Tlačítka na ovládání scény jsou vidět na Obr. 49.



Obr. 49 Použitá tlačítka

7.2.3 Spouštění aplikace

Následuje vytvoření souboru s aplikací. To se dělá přes *File* → *Build Settings* a *Build*, následně se soubor nahraje do VR headsetu a aplikaci je možné spustit. Při spouštění aplikace byly zjištěny chyby, které byly zapsány (viz. Tab. 5) a poté opraveny.

Tab. 5 Testování aplikace

Verze 1.0.01	Při testování byla zjištěna špatná viditelnost modelů kvůli bílému pozadí za modely. Vyřešeno černou plochou.
Verze 1.0.02	Při přepínání mezi modely se nepřerušoval zvuk, který byl k jednotlivým modelům. Vyřešeno pomocí skriptu.
Verze 1.0.03	Při změně rozměru modelů bylo možné model neomezeně zvětšovat a zmenšovat. Skriptem se změnila meze pro rozměry.
Verze 1.0.04	Špatné textury modelů. Modely měly průhledné prvky. Opraveno změnou nastavení zdroje materiálu.

Po opravě chyb při testování je aplikace připravena pro použití.

8 Závěr

Cílem této bakalářské práce bylo vytvoření aplikace, ve které uživatel interaguje s průmyslovými 3D modely. Práce je rozdělena na teoretickou a praktickou část. V teoretické části byla provedena rešerše odborných publikací. Nejprve byl vysvětlen základní princip fungování technologie virtuální reality. Dále byly popsány základní metody ovládání pohybu ve virtuálním prostředí a jejich vliv na uživatele. Následně bylo popsáno rozdělení virtuálních headsetů a představeny praktické příklady využití virtuální reality. Poté byl proveden průzkum existujících metod interakcí s modely a možností implementace 3D modelů ve virtuální realitě. Na závěr byla provedena analýza a vyhodnocení nejvhodnějších modelovacích softwarů a herních enginů pro vývoj aplikace.

V praktické části byla provedena tvorba finální aplikace. Nejprve byl vytvořen obsah aplikace. Ten tvoří realistické průmyslové 3D modely. Modely byly voleny tak, aby byly typické pro strojírenskou praxi. Tvorba modelů probíhala v modelovacím softwaru Blender. Blender byl následně použit i pro tvorbu animace roztřelu modelů, při kterých se model rozdělí na jednotlivé části. Modely z Blenderu s animacemi byly následně exportovány v podporovaném formátu pro herní engine Unity. Hlavní edukační části aplikace tvořily zvukové soubory, které byly vytvořené pro každý průmyslový model a obsahovaly charakteristický popis. Finálním krokem byla tvorba aplikace v herním enginu Unity. Pro aplikaci bylo naimportováno prostředí. V Unity bylo potřeba pomocí programování v C# vytvořit funkcionalitu aplikace. Interakce s průmyslovými modely byly vyřešeny pomocí tlačítek, díky kterým má uživatel možnost provést rotaci, změnu rozměru a spustit animaci roztřelu a opětovného složení modelu.

Výstupem této bakalářské práce je funkční aplikace pro virtuální headset Oculus Quest 2. V aplikaci má uživatel možnost výběru z šesti realistických průmyslových modelů. Při výběru modelu z tabulky se spustí zvukový soubor, který popisuje charakteristické vlastnosti modelu. S modely může uživatel v univerzálním prostředí manipulovat a zkoumat tak jejich stavbu a charakteristickou funkci. V rámci této aplikace se nabízí možnost tuto práci dále rozvíjet, a to především v oblasti používání dalších typů interakcí jako jsou gesta, hlasové ovládání a rozšíření knihovny modelů. Další možností by bylo přidání testu pro ověření získaných znalostí.

9 Seznam použitých zdrojů

- [1] LINOWES, Jonathan. *Unity 2020 Virtual Reality Projects - Third Edition: Learn VR development by building immersive applications and games with Unity 2019.4 and later versions* [online]. 3rd Edition. Packt Publishing, 2020 [cit. 2023-11-25]. ISBN 978-1839217333.
- [2] KUGLER, Logan. The state of virtual reality hardware. *Communications of the ACM* [online]. 2021, 2021-01-25, 64(2), 15-16 [cit. 2023-11-11]. ISSN 0001-0782. Dostupné z: doi:10.1145/3441290
- [3] SUNG, Kelvin a Gregory SMITH. *Basic Math for Game Development with Unity 3D* [online]. Berkeley, CA: Apress, 2019 [cit. 2023-11-26]. ISBN 978-1-4842-5442-4. Dostupné z: doi:10.1007/978-1-4842-5443-1
- [4] KHUNDAM, Chaowan, Varunyu VORACHART, Patibut PREEYAWONGSAKUL, Witthaya HOSAP a Frédéric NOËL. A Comparative Study of Interaction Time and Usability of Using Controllers and Hand Tracking in Virtual Reality Training. *Informatics* [online]. 2021, 8(3) [cit. 2023-11-11]. ISSN 2227-9709. Dostupné z: doi:10.3390/informatics8030060
- [5] BOLETISIS, Costas a Jarl Erik CEDERGREN. VR Locomotion in the New Era of Virtual Reality: An Empirical Comparison of Prevalent Techniques. *Advances in Human-Computer Interaction* [online]. 2019, 2019-04-01, 2019, 1-15 [cit. 2023-11-10]. ISSN 1687-5893. Dostupné z: doi:10.1155/2019/7420781
- [6] LAVALLE, Steven. *VIRTUAL REALITY* [Cambridge University Press 2023]. [cit. 2023-11-26]. Dostupné z: <http://lavalle.pl/vr>
- [7] ANIWAA. *Types of VR headsets: PC VR, standalone VR, smartphone VR* [online]. ANIWAA TEAM. 2021, 2021-8-6 [cit. 2023-11-03]. Dostupné z: <https://www.aniwaa.com/guide/vr-ar/types-of-vr-headsets/>
- [8] BROWN, Rory. VRCOMPARE. *Oculus Quest 2 vs Pico Neo 3 vs HTC Vive Pro 2 vs HP Reverb G2 (Comparison)* [online]. [cit. 2023-11-16]. Dostupné z: https://vr-compare.com/compare?h1=pDTZ02PkT&h2=TIUMm3G29&h3=mLbW9G7f4&h4=HdfjN_0UIPY
- [9] MACHAŁA, Szymon, Norbert CHAMIER-GLISZCZYŃSKI a Tomasz KRÓLIKOWSKI. Application of AR/VR Technology in Industry 4.0. *Procedia Computer Science* [online]. 2022, 2022(207), 2990-2998 [cit. 2023-11-06]. ISSN 18770509. Dostupné z: doi:10.1016/j.procs.2022.09.357
- [10] ZAWADZKI, Przemyslaw, Krzysztof ZYWICKI, Pawel BUN a Filip GORSKI. Employee Training in an Intelligent Factory Using Virtual Reality. *IEEE Access* [online]. 2020, 8, 135110-135117 [cit. 2023-11-16]. ISSN 2169-3536. Dostupné z: doi:10.1109/ACCESS.2020.3010439

- [11] LV, Zhihan. Virtual reality in the context of Internet of Things. *Neural Computing and Applications* [online]. 2020, 32(13), 9593-9602 [cit. 2023-11-17]. ISSN 0941-0643. Dostupné z: doi:10.1007/s00521-019-04472-7
- [12] METAVERSEVRNOW. Virtual reality in engineering industry: Pros and cons - metaverse VR now. *Virtual reality in engineering industry: Pros and cons - metaverse VR now* [online]. 2022 [cit. 2023-11-05]. Dostupné z: https://metaversevrnow.com/engineering/virtual-reality-in-engineering/#Virtual_Reality_Is_Shaping_Todays_Engineering_Work
- [13] Human-Centric Product & Process Validation Day. *ESI Group* [online]. 2022 [cit. 2023-11-05]. Dostupné z: <https://www.esi-group.com/events/2022/human-centric-product-process-validation-day>
- [14] LARSON, Kate. Teach your students with virtual reality. In: *Ideas & Inspiration from Demco* [online]. 2023 [cit. 2023-10-28]. Dostupné z: <https://ideas.demco.com/blog/teach-with-virtual-reality/>
- [15] BEDNÁŘ, Miroslav, David KRÁKORA a Michal ŠIMON. Optimalizace 3D modelů pro virtuální realitu. In: *Průmyslové inženýrství 2021: Sborník příspěvků* [online]. Západočeská univerzita v Plzni, 2021, s. 15-23 [cit. 2023-11-02]. ISBN 9788026107927. Dostupné z: doi:10.24132/PI.2021.07927.015-023
- [16] animationkolkata. MAAC. ASPECTS OF 3D CHARACTER MODELING. MAAC. *Animationkolkata* [online]. 2021, 2023-01-02 [cit. 2023-11-02]. Dostupné z: <https://www.animationkolkata.com/blog/2021/05/31/aspects-3d-character-modeling/>
- [17] DOERNER, Ralf, Christian GEIGER, Leif OPPERMANN, Volker PAELKE a Steffi BECKHAUS. Interaction in Virtual Worlds. In: DOERNER, Ralf, Wolfgang BROLL, Paul GRIMM a Bernhard JUNG, ed. *Virtual and Augmented Reality (VR/AR)* [online]. Cham: Springer International Publishing, 2022, s. 201-244 [cit. 2023-11-17]. ISBN 978-3-030-79061-5. Dostupné z: doi:10.1007/978-3-030-79062-2_6
- [18] KLIPPEL, Alex, Mahda BAGHER a Jan WALLGRÜN. 2.6 3D and VR Application Building Workflows | *GEOG 497: 3D Modeling and Virtual Reality* [online]. [cit. 2023-11-18]. Dostupné z: <https://www.e-education.psu.edu/geogvr/node/559>
- [19] PICKAVANCE, Mark, John LOEFFLER a Steve CLARK. *The best 3D modeling software in 2023* [online]. [cit. 2023-11-25]. Dostupné z: <https://www.techradar.com/best/best-3d-modelling-software#section-best-3d-modeling-software-for-animators>
- [20] HARRISON, Mark. *Take a .PNG snapshot in SketchUp for Web* [online]. In: . [cit. 2023-11-26]. Dostupné z: <https://forums.sketchup.com/t/take-a-png-snapshot-in-sketchup-for-web/82648>
- [21] *Blender V – Darker UI for Blender to showcase your work*. [online]. In: 3DIRECTOR. [cit. 2023-11-26]. Dostupné z: <https://3dcinetv.com/blender-v-darker-ui-for-blender-to-showcase-your-work/>

- [22] AMIN, Jahirul. *Introduction to Maya: Learn the Maya User Interface (UI)* [online]. In: . [cit. 2023-11-26]. Dostupné z: <https://3dtotal.com/tutorials/t/introduction-to-maya-learn-the-maya-user-interface-ui-jahirul-amin-ui>
- [23] TUREK, Vlado a Vladimír KAFKA. *Základy sculptingu v ZBrush Core Mini* [online]. In: . [cit. 2023-11-26]. Dostupné z: https://blog.prusa3d.com/cs/zaklady-sculptingu-v-zbrush-core-mini_69511/
- [24] VIONIX. Godot vs Cryengine game engine comparison. *VionixStudio* [online]. 2020 [cit. 2023-11-19]. Dostupné z: <https://vionixstudio.com/2020/01/18/godot-vs-cryengine-game-engine-comparison/>
- [25] GAJSEK, Dejan. *Unity vs Unreal Engine for XR Development: Which One Is Better? [2022 Updated]* [online]. 2022 [cit. 2023-11-19]. Dostupné z: <https://circuitstream.com/blog/unity-vs-unreal#z5>
- [26] OKORO, Ikechi. Tools and Platforms for XR Content Creation. In: *Medium* [online]. [cit. 2023-11-19]. Dostupné z: <https://medium.com/imisi3d/tools-and-platforms-for-xr-content-creation-debd35235b06>
- [27] DAVE, Liam. *Want to make a 3D adventure game? There's now a full template project for Godot Engine* [online]. [cit. 2023-11-29]. Dostupné z: <https://www.gamingonlinux.com/2020/02/want-to-make-a-3d-adventure-game-theres-now-a-full-template-project-for-godot-engine/>
- [28] MOZOLEVKSA, Victoria. *Best Game Engines of 2023: Pros, Cons, and Top Picks for Different Types of Games* [online]. [cit. 2023-11-20]. Dostupné z: <https://kevurugames.com/blog/best-game-engines-2022-pros-cons-and-top-picks-for-different-types-of-games/>
- [29] CALLAHAM, John. CryEngine officially joins Unreal Engine 4 as a subscription-based service. In: *Neowin* [online]. [cit. 2023-11-20]. Dostupné z: <https://www.neowin.net/news/cryengine-officially-joins-unreal-engine-4-as-a-subscription-based-service/>
- [30] 1. Šroubové spoje, 2010. In: Maturitní otázky, studium, témata [online]. 2010-22-09 [cit. 2024-04-04]. Dostupné z: <https://strojirenstvi-mechanika.blogspot.com/2010/09/1-sroubove-spoje.html>
- [31] Jaký je rozdíl mezi různými druhy ložisek? [online]. In: . 2015-19-10 [cit. 2024-04-04]. Dostupné z: <https://e-konstrukter.cz/novinka/jaky-je-rozdil-mezi-ruzny-mi-druhy-lozisek>
- [32] PROTIVÍNSKÝ, Luděk. KATEDRA FYZIKY, CHEMIE A ODBORNÉHO VZDĚLÁVÁNÍ PEDAGOGICKÁ FAKULTA MU PŘEDMĚT: UOPK 8009 Části strojů 2 VYUČUJÍCÍ: Mgr. Ing. Kateřina Šmejkalová [online]. [cit. 2024-04-04]. Dostupné z: <https://slideplayer.cz/slide/11784725/>
- [33] Diferenciál - k čemu slouží a jak funguje? [online]. In: . [cit. 2024-04-04]. Dostupné z: <https://www.cebia.cz/novinky/rady-a-tipy/diferencial-k-cemu-slouzi-a-jak-funguje>

[34] VĚTRNÁ ELEKTRÁRNA [online]. [cit. 2024-04-04]. Dostupné z:
<https://www.svetenergie.cz/cz/energetika-zblizka/obnovitelne-zdroje-energie-pro-deti/vetrna-energie-pro-deti/vetrna-elektrarna/jak-funguje>

Seznam příloh

Příloha č. 1: Skript ModelSelector

Příloha č. 2: Skript AnimateObject

PŘÍLOHA č. 1

Skript AnimateObject

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ModelSelector : MonoBehaviour
{
    public GameObject[] _genObj;

    public AudioClip[] _audioObj;

    public Transform spawnPoint;

    public GameObject oldObj;

    private float defScale;

    public void GenerateObj(int i)
    {
        if (oldObj != null)
        {
            Destroy(oldObj);
        }

        oldObj = Instantiate(_genObj[i]);
        oldObj.transform.parent = spawnPoint;
        oldObj.transform.localPosition = Vector3.zero;
        oldObj.transform.localEulerAngles = Vector3.zero;
        defScale = oldObj.transform.localScale.x;
        this.GetComponent().clip = _audioObj[i];
        this.GetComponent().Play();
    }

    public void RotationObjx(float i)
    {
        if (oldObj != null)
        {
            oldObj.transform.localEulerAngles += new Vector3(0f, i, 0f);
        }
    }

    public void RotationObjy(float i)
    {
        if (oldObj != null)
        {
            oldObj.transform.localEulerAngles += new Vector3(i, 0f, 0f);
        }
    }

    public void ScaleObj(float i)
    {
        if (oldObj != null)
        {
            float x = oldObj.transform.localScale.x * i;

            if (oldObj.transform.localScale.x <= defScale * 2 && x > 0)
            {
                oldObj.transform.localScale += new Vector3(x, x, x);
            }
            else if (oldObj.transform.localScale.x >= defScale / 2 && x < 0)
            {

```

```
        oldObj.transform.localScale += new Vector3(x, x, x);
    }
}
}
```

PŘÍLOHA č. 2

Skript AnimateObject

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AnimateObject : MonoBehaviour
{
    public bool _reverse;
    public bool _activeAnim;

    public ModelSelector _modelSelector;

    public void StartRozpadReverse()
    {
        if(!_activeAnim && _modelSelector.oldObj != null)
        {
            if(!_reverse)
            {
                _reverse = true;
                StartCoroutine(StartRozpadReverse_cor("Rozstrel"));
            }
            else
            {
                _reverse = false;
                StartCoroutine(StartRozpadReverse_cor("Reverse"));
            }
        }
    }

    IEnumerator StartRozpadReverse_cor(string animTrigger)
    {
        _activeAnim = true;

        _modelSelector.oldObj.GetComponentInChildren<Animator>().SetTrigger(animTrigger);
        yield return new WaitForSeconds(2.5f);
        _activeAnim = false;
    }
}
```