

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ

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

BAKALÁŘSKÁ PRÁCE

**Využití nástrojů umělé inteligence pro tvorbu aplikací do
průmyslových podniků**

Autor: Šimon Šobr
Vedoucí práce: Ing. Miroslav Bednář
Konzultant 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: **Šimon ŠOBR**
Osobní číslo: **S23B0259P**
Studijní program: **B0715A270013 Strojní inženýrství**
Specializace: **Průmyslové inženýrství a management**
Téma práce: **Využití nástrojů umělé inteligence pro tvorbu aplikací do průmyslových podniků**
Zadávací katedra: **Katedra průmyslového inženýrství a managementu**

Zásady pro vypracování

- Úvod do Industry 4.0
- Současný stav nástrojů umělé inteligence
- Návrh použití umělé inteligence pro tvorbu aplikací do průmyslových podniků
- Tvorba aplikace pomocí nástrojů umělé inteligence
- Závěr a vyhodnocení

Rozsah bakalářské práce: **30-50 stran**
Rozsah grafických prací: **–**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. GREENGARD, Samuel. *Virtual Reality*. Cambridge, MA: The MIT Press, 2019. ISBN 262537524.
2. SHARDA, Ramesh, Dursun DELEN a Efraim TURBAN. *Analytics, data science, & artificial intelligence: systems for decision support*. Eleventh edition, global edition. Harlow: Pearson, 2021. ISBN 978-1-292-34155-2.
3. RUSSELL, Stuart J. a Peter NORVIG. *Artificial intelligence: a modern approach. Fourth edition, global edition*. Harlow: Pearson, 2022. ISBN 978-1-292-40113-3.

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

Konzultant bakalářské práce: **Ing. Matěj Dvořák**
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

Tímto bych chtěl poděkovat vedoucímu mé bakalářské práce, panu Ing. Miroslavu Bednářovi, za jeho cenné rady, čas a ochotu poradit během mého psaní této práce. Také bych chtěl poděkovat konzultantovi práce, panu Ing. Matěji Dvořákovi.

ANOTAČNÍ LIST BAKALÁŘSKÉ PRÁCE

AUTOR	Příjmení Šobr	Jméno Šimon	
STUDIJNÍ PROGRAM	B0715A270013 Strojní inženýrství		
VEDOUCÍ PRÁCE	Příjmení (včetně titulů) Ing. Bednář	Jméno Miroslav	
PRACOVIŠTĚ	ZČU – FST – KPV		
DRUH PRÁCE	DIPLOMOVÁ	BAKALÁŘSKÁ	Nehodící se škrtněte
NÁZEV PRÁCE	Využití nástrojů umělé inteligence pro tvorbu aplikací do průmyslových podniků		

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

POČET STRAN (A4 a ekvivalentů A4)

CELKEM	50	TEXTOVÁ ČÁST	42	GRAFICKÁ ČÁST	8
---------------	----	---------------------	----	----------------------	---

STRUČNÝ POPIS (MAX 10 ŘÁDEK) ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY	Tato bakalářská práce pojednává o možnostech využití nástrojů umělé inteligence v oblasti vývoje softwarových aplikací. Následně popisuje pracovní postup tvorby pracovní návodky ve virtuální realitě a konkrétní použité nástroje umělé inteligence. Na závěr obsahuje porovnání běžného postupu a toho použitého v této práci.
KLÍČOVÁ SLOVA ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE	AI, VR, umělá inteligence, virtuální realita, průmysl 4.0, Unity3D, aplikace, software

SUMMARY OF BACHELOR SHEET

AUTHOR	Surname Šobr	Name Šimon	
STUDY PROGRAMME	B0715A270013 Mechanical Engineering		
SUPERVISOR	Surname (Inclusive of Degrees) Ing. Bednář	Name Miroslav	
INSTITUTION	ZČU - FST - KPV		
TYPE OF WORK	DIPLOMA	BACHELOR	Delete when not applicable
TITLE OF THE WORK	Using artificial intelligence tools to create applications for industrial companies		

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

NUMBER OF PAGES (A4 and eq. A4)

TOTALLY	50	TEXT PART	42	GRAPHICAL PART	8
----------------	----	------------------	----	-----------------------	---

BRIEF DESCRIPTION TOPIC, GOAL, RESULTS AND CONTRIBUTIONS	This bachelor thesis discusses the possible uses of artificial intelligence tools within the field of software development. Furthermore, it details the workflow of creating a training virtual reality application. The final part deals with the comparison of a standard workflow when developing applications and the workflow used in this thesis.
KEY WORDS	AI, VR, Artificial Intelligence, Virtual Reality, Industry 4.0, Unity3D, Application, Software

Obsah

Seznam obrázků	9
Seznam tabulek	10
Seznam zkratk	10
Úvod	11
1 Historické pozadí.....	12
1.1 Průmysl 4.0.....	13
2 Umělá inteligence.....	14
2.1 Rozdělení umělé inteligence	15
2.1.1 Rozdělení na základě schopností.....	15
2.1.2 Rozdělení na základě funkce.....	16
3 Technologie XR	17
3.1 Virtuální realita	17
3.1.1 PC VR	18
3.1.2 Standalone VR.....	19
3.2 Rozšířená realita	20
3.2.1 Head Mounted Display.....	20
3.2.2 Ruční zařízení.....	21
3.2.3 Spatial Augmented Reality.....	21
4 Nástroje využívající umělou inteligenci.....	22
4.1 Nástroje na programování	23
4.2 Nástroje na tvorbu textur.....	24
4.3 Nástroje na tvorbu modelů	25
4.4 Nástroje na tvorbu obrázků	25
5 Návrh aplikace.....	27
5.1 Návrh činnosti	27
5.2 Průběh scénáře ve virtuálním prostředí	29
6 Tvorba aplikace	30
6.1 Tvorba prostředí	30
6.2 Tvorba předmětů v prostředí	33
6.3 Tvorba hlasového popisu postupu.....	36
6.4 Projekt v Unity3D	38
6.5 Skripty pro funkcionalitu aplikace	41
7 Zhodnocení a porovnání pracovního postupu	45
Závěr.....	46
Seznam použitých zdrojů	47

Seznam obrázků

Obr. 1 - Průmyslové revoluce [4].....	12
Obr. 2 - Zařízení Oculus Rift S [23].....	17
Obr. 3 - Příklad zařízení CAVE [52].....	18
Obr. 4 - Způsoby sledování pozice HMD [26].....	19
Obr. 5 - Zařízení Meta Quest 2 [53].....	20
Obr. 6 - Srovnání provedení HMD AR [54]	21
Obr. 7 - Ruční zařízení na rozšířenou realitu [55].....	21
Obr. 8 - Struktura umělé neuronové sítě [34].....	22
Obr. 9 - Princip využití umělé inteligence na programování [36]	23
Obr. 10 - Příklad nástroje na tvorbu textur [40].....	24
Obr. 11 - Modely vygenerované pomocí AI [41] [42].....	25
Obr. 12 - Obrázky vygenerované pomocí AI [45]	26
Obr. 13 - Panoramatický snímek vytvořen pomocí AI [47].....	26
Obr. 14 - Komponenty uvnitř LED televize [56].....	28
Obr. 15 - Připevnění zadního krytu k rámu [57]	29
Obr. 16 - Prvotní podoba prostředí.....	30
Obr. 17 - Textura podlahy [39]	31
Obr. 18 - Textura zdi [39]	31
Obr. 19 - Generování pracovního stolu [48]	31
Obr. 20 - Vygenerovaný model a možnosti [48].....	32
Obr. 21 - Generování prostorového snímku [47]	32
Obr. 22 - Textové pokyny pro generování	33
Obr. 23 - Výsledná podoba vygenerovaného rámu [49]	33
Obr. 24 - Model základní desky [48]	34
Obr. 25 - Model zdrojového modulu [48]	34
Obr. 26 - Model šroubováku [48].....	35
Obr. 27 - Srovnání kvality modelů u funkce Decimate.....	35
Obr. 28 - Nastavení Decimate v Blenderu	35
Obr. 29 - Využití nástroje na generaci hlasového záznamu [50]	36
Obr. 30 - Potřebné pluginy pro VR funkcionalitu.....	38
Obr. 31 - Nastavení Oculus Pluginu.....	38
Obr. 32 - Nastavení XR Pluginu	38
Obr. 33 - Rozmístění prvků osvětlení	39
Obr. 34 - Pracovní plocha s umístěnými komponenty	40

Seznam tabulek

Tabulka 1 - Kategorie definic umělé inteligence [8]..... 14

Seznam zkratk

AI	Artificial Intelligence
IoT	Internet of Things
IoS	Internet of Services
NLP	Natural Language Processing
VR	Virtual Reality
XR	Extended Reality
SAR	Spatial Augmented Reality
HMD	Head Mounted Display
GenAI	Generative Artificial Intelligence
ANN	Artificial Neural Network
GAN	Generative Adversarial Network

Úvod

V současném dynamickém prostředí prochází průmyslová odvětví neustálým procesem transformace, kde se nové technologie stávají hybnou silou efektivnosti a inovace. Koncept Průmyslu 4.0, charakterizovaný digitální transformací a propojením technologií, poskytuje nové perspektivy pro průmyslové podniky. Jedním z klíčových prvků této transformace je využití nástrojů umělé inteligence, které umožňují automatizaci, analýzu dat a vytváření inteligentních systémů.

Tato práce se zaměřuje na důležitý kontext spojený s vývojem aplikací pro průmyslová odvětví pomocí nástrojů využívajících umělou inteligenci. Umělá inteligence, jako klíčový prvek Průmyslu 4.0, přináší schopnost strojového učení, analýzy velkého množství dat a automatizace rozhodovacích procesů. Tato práce se detailně zabývá rozdělením umělé inteligence a zkoumá její různé aspekty, jako jsou konvenční metody učení, neuronové sítě, strojové učení a její funkcionalitu, které jsou klíčovými koncepty pro vytváření inteligentních aplikací v průmyslovém kontextu.

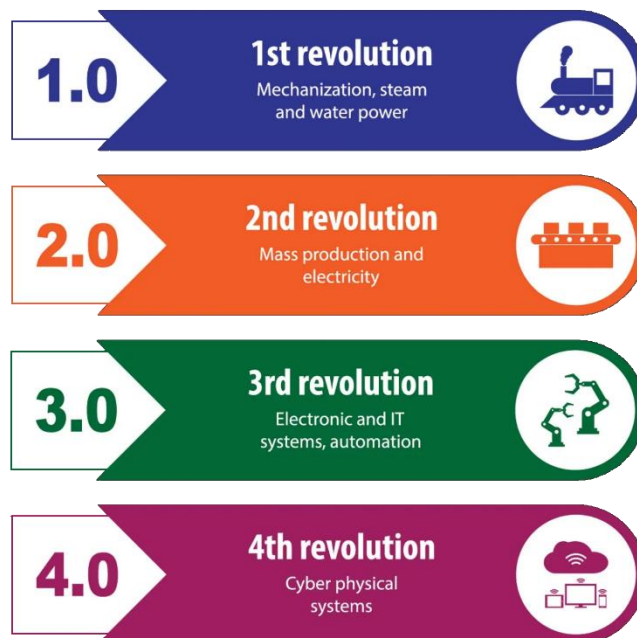
V rámci tohoto hrají také technologie virtuální a rozšířené reality významnou roli. Poskytují nové možnosti pro interakci s daty a procesy v průmyslovém prostředí. Práce analyzuje, jak mohou zmíněné technologie spolupracovat s nástroji umělé inteligence k vytváření inovativních a efektivních průmyslových aplikací.

Dále je součástí této práce zkoumání konkrétních nástrojů využívajících umělou inteligenci, jejichž využití jsou relevantní pro tvorbu aplikací v tomto odvětví. Mezi tyto nástroje se řadí například ty z oblasti programování nebo 3D modelování. Práce analyzuje, jaké nástroje jsou k dispozici, jaké možnosti poskytují a jak mohou být implementovány pro optimalizaci průmyslových procesů pomocí vytvoření aplikací v oblasti průmyslového inženýrství.

Následuje návrh pracovního postupu tvorby aplikace za pomoci umělé inteligence a jeho detailní popis. Tato část také obsahuje výsledky a zhodnocení.

1 Historické pozadí

Moderní doba průmyslu sahá několik set let do minulosti a mezi její nejdůležitější milníky patří tzv. průmyslové revoluce, které značí stěžejní změny v průmyslu, výrobě a společnosti obecně. Definujeme čtyři takováto období.



Obr. 1 - Průmyslové revoluce [4]

První průmyslová revoluce započala ke konci 18. století a pokračovala do 19. století. Byla charakterizována mechanizací výroby v továrnách za využití parního a vodního pohonu strojů.

Počátek druhé průmyslové revoluce se datuje na konec 19. století. Jejími hlavními znaky bylo široké využití elektřiny a také vznik montážních linek, což mělo také za následek vznik hromadné výroby a její větší efektivitu.

Třetí průmyslová revoluce, též známá jako digitální revoluce, se zrodila v polovině 20. století. Bylo to období elektroniky, automatizace a počítačových systémů. Pomocí informačních technologií bylo dosaženo zefektivnění, automatizace a větší míry komunikace ve výrobě [1].

Období počínající začátkem 21. století a to, ve kterém se nacházíme nyní, je čtvrtá průmyslová revoluce. Dalším běžným termínem je také Průmysl 4.0 (Industry 4.0), nebo někdy také 4IR (4th Industrial Revolution) [2]. Jeho charakteristiky jsou integrace digitálních technologií do výroby, vzájemná propojenost strojů a systémů, využití internetu ve všech odvětvích a také prolínání reálného a virtuálního světa v tzv. kyberfyzických systémech [3].

1.1 Průmysl 4.0

Mezi čtvrtou a třetí průmyslovou revolucí je více pozvolný přechod, než jak tomu bylo v předchozích případech. Zastánci tohoto termínu argumentují, že čtvrtá průmyslová revoluce není jen pokračováním té třetí díky jejím následujícím vlastnostem [4].

- **Rychlost** – Míra pokroku a inovací roste exponenciálně, na rozdíl od ostatních období, kdy měla spíše lineární průběh.
- **Rozsah** – Čtvrtá průmyslová revoluce není lokalizovaným fenoménem, ale ovlivňuje takřka všechna průmyslová odvětví po celém světě.
- **Změny hospodářské politiky** – Nové obchodní modely založené na Průmyslu 4.0 a obecná změna paradigmatu v průmyslovém odvětví.

Z tohoto důvodu tak nelze přesně definovat počátek Průmyslu 4.0. Tento termín byl však poprvé použit v Německu v roce 2011 na charakterizování konceptu hospodářské politiky, která využívala nových vyspělých strategií [3]. Jedná se o integrování nejvyspělejších technologií a dosažení ideálního toku informací mezi fyzickým a digitálním světem.

Definice Průmyslu 4.0 je mnoho, ale obecně se shodují na tom, že na jeho středu stojí tzv. kyberfyzické systémy (CPS, z anglického Cyber-Physical System). Jedná se o sofistikované spojení výpočetních algoritmů se samotnými fyzickými procesy za účelem zvýšení efektivity a přesnosti a také většího pochopení komplexních systémů, které přináší lepší možnost řešení náročných problémů. V Průmyslu 4.0 je kladen důraz na propojenost na globálním měřítku. Zařízení a systémy spolu komunikují a tvoří tak síť, která umožňuje přenos informací v reálném čase.

Důležitými pojmy v rámci Průmyslu 4.0 jsou tzv. Internet of Things a Internet of Services. Oba tyto koncepty jsou založeny na nepřetržité internetové propojení pomocí internetu, což umožňuje komunikaci nejen mezi lidmi (C2C – Communication to Communication), ale také mezi stroji navzájem (M2M – Machine to Machine) [5].

- **Internet of Things (IoT)** - Jedná se o síť zařízení, strojů, či jiných fyzických objektů se vzájemnou konektivitou a senzory, které jsou schopny sbírat data v reálném čase. Tyto data jsou pak přeposílána a zanalyzována, čímž se výrazně zlepšuje a zefektivňuje proces rozhodování či monitorování výroby. Výhodou je také větší míra automatizace, dosažená vzájemnou komunikací strojů a tím pádem menší nutností zásahu člověka [6].
- **Internet of Services (IoS)** – Je to model, ve kterém jsou služby poskytovány, organizovány a využívány přes internet. Umožňuje značnou míru flexibility a rozšiřitelnosti infrastruktury. Nejčastěji se jedná o systémy uschování dat, zpracování dat a datové analýzy. Jeho součástí je tzv. cloud computing (vzdálené poskytování služeb servery přístupnými z internetu). Tento více centralizovaný přístup umožňuje jednodušeji spravovat či analyzovat např. data uložená na serveru [7].

Dalším klíčovým rysem je tzv. interoperabilita neboli schopnost rozdílných systémů a jejich komponentů spolu navzájem komunikovat a být kompatibilní, což umožňuje větší míru spolupráce. Interoperability je docíleno např. pomocí využívání standardizovaných datových formátů a komunikačních protokolů (jako například XML nebo SQL).

2 Umělá inteligence

Pojem umělá inteligence, často označována jako AI (Artificial Intelligence), se zrodil v 50. letech 20. století [8] a jeho počátky jsou spjaty s tzv. Turingovým testem vyvinutým Alanem Turingem v roce 1950 [9]. V tomto testu měl lidský porotce přístup k záznamu komunikace mezi člověkem a strojem. Věděl pouze, že jeden z účastníků byl stroj, ale nevěděl který. Pokud nedokázal spolehlivě vyhodnotit, který z účastníků je stroj, pak daný stroj tímto testem prošel a jednalo se o umělou inteligenci.

Turingův přístup k definování umělé inteligence je pouze jedním z mnoha způsobů. Všechny lze obecně kategorizovat do čtyř hlavních skupin, které jsou popsány v Tabulce 1. Ty v horním řádku se zaměřují na myšlení a uvažování, zatímco ty ve spodním řádku na chování. Dále pak definice v levém sloupci jsou zaměřeny na míru podobnosti výkonu stroje k lidskému výkonu a ty v pravém sloupci cílí na porovnání výkonu stroje k nějakému ideálnímu, či „správnému“ výsledku.

Tabulka 1 - Kategorie definic umělé inteligence [8]

Lidské smýšlení	Racionální smýšlení
Lidské chování	Racionální chování

- **Lidské smýšlení** – K porovnání míry podobnosti smýšlení stroje ke smýšlení lidskému je nejdříve nutné určit, jak přesně lidské myšlení funguje. Jedním příkladem řešení, které použili Newell a Simon ve svém projektu GPS (General Problem Solver) v roce 1961 bylo nechat stroj a lidi řešit stejný problém a poté porovnávat jednotlivé kroky a myšlenkové pochody lidí během řešení s těmi, jaké měl stroj [10].
- **Lidské chování** – Do této kategorie spadá dříve zmíněný Turingův test. Aby byl stroj tento komplexní test schopen splnit, musí být schopen mj. přirozené jazykové komunikace, zapamatování si věcí, které vidí a slyší, využívání těchto informací na zodpovídání otázek a řešení problémů a adaptování se na nové podmínky. Existuje však také na něm postavený úplnější test, známý jako tzv. totální Turingův test [11], který se zaměřuje nejen na výše zmíněné vlastnosti, ale také na schopnost vnímání pomocí počítačového vidění a manipulaci s objekty pomocí např. robotických ramen.
- **Racionální smýšlení** – Tento přístup tvoří tzv. „správné myšlení“, tj. nevyvratitelné myšlenkové pochody. Na tomto je založeno odvětví logiky, které je schopné popsat různorodá tvrzení o předmětech a vztahy mezi nimi pomocí přesně stanovené notace.
- **Racionální chování** – Dle této definice by se program měl chovat racionálně, tedy v každé situaci se zachovat tak, aby bylo dosaženo nejlepšího výsledku, nebo v nejisté situaci alespoň nejlepšího očekávaného výsledku. Klade tak důraz na správné závěry, které jsou utvořeny na základě daných podkladů.

2.1 Rozdělení umělé inteligence

Přesné členění umělé inteligence není dané, ale můžeme ji dělit do různých kategorií na základě porovnávání různých vlastností.

2.1.1 Rozdělení na základě schopností

- **Úzká či slabá umělá inteligence (Narrow AI/Weak AI)**

Jedná se o systémy, které jsou vytvořeny a vycvičeny za účelem plnění jednoho specifického úkolu nebo úzkého spektra úkolů. Jsou schopny velmi efektivně vykonávat předem definované činnosti při zadaných parametrech. K tomuto potřebují velké množství dat, která se k jejich funkci vztahují. Jejich největší nevýhodou je omezené využití v situacích, jelikož nejsou schopny zobecňování napříč různorodými problémy.

Úzká umělá inteligence má mnohá použití, kde nejčastější jsou následující:

- **Virtuální asistenti**

Dovolují uživateli hlasem zadat nějaký pokyn, či se zeptat na nějakou informaci a tu mu dále předají [12].

- **Zpracování přirozeného jazyka (NLP – Natural Language Processing)**

Tyto programy jsou schopny pochopit lidský jazyk a také jím odpovídat. Jsou to např. online systémy zákaznické podpory nebo překladače jazyků.

- **Rozpoznávání obrazu a hlasu**

Jsou schopny rozpoznat vizuální a zvukové podměty jako např. u zabezpečovacích systémů rozpoznání obličeje u mobilních zařízení.

- **Doporučovací systémy**

Využívány u mnoha webových stránek na vyhodnocení uživatelského chování a následného doporučení produktu, o který by podle algoritmu měl mít zájem. Nejvíce běžné např. u online obchodů nebo serverů na sledování filmů [13].

- **Systémy detekce podvodu**

Jsou schopny detekovat např. neobvyklé transakce u bankovních účtů a předejít tak možným finančním ztrátám díky podvodu.

- **Obecná či silná umělá inteligence (General AI/Strong AI)**

Na rozdíl od úzké umělé inteligence je tento typ stále pouze teoretický a zatím nebyl realizován. Jeho hlavní charakteristikou je, že má širokou škálu znalostí a tyto znalosti je schopen aplikovat na řešení různorodých problémů. Zároveň je schopen získávat nové zkušenosti, učit se z nich a autonomně se přizpůsobovat nově vzniklým podmínkám. Obecná umělá inteligence by tak byla schopna přemýšlet a řešit problémy na stejné úrovni jako člověk [14].

- **Umělá super-inteligence (ASI)**

Koncept umělé super-inteligence je pouze hypotetický. Jedná se o tak pokročilý systém, že by převyšoval veškeré schopnosti lidí. Jeho schopnost učit se a přizpůsobovat by mu dovozovala neustále rapidně zvyšovat své znalosti a inteligenci [15].

2.1.2 Rozdělení na základě funkce

- **Reactive Machine**

Jedná se o základní úroveň umělé inteligence. Stroje tohoto typu fungují na základě předem stanovených pravidel a algoritmů a jsou navrženy na použití u jednoho specifického úkonu a nemají využití u žádného jiného. Neukládají si do paměti žádná data a nejsou tak schopny využívat předešlé zkušenosti. Z tohoto také vychází jejich název – pouze vnímají momentální situaci a na ni reagují. Z předchozích vlastností také vyplývá, že nemají schopnost učit se a měnit svá rozhodnutí, tedy na specifický vstup mají danou stálou reakci (mají deterministický charakter).

Využití nachází jako tzv. expertní systémy [16], tedy systémy napodobující rozhodovací schopnost lidského experta v daném odvětví. Jsou navrženy na řešení komplexních problémů. Dalším příkladem mohou být dřívější programy schopné hraní šachu.

- **Limited Memory**

Na rozdíl od předešlého typu mají tyto systémy schopnost ukládat si v paměti minulé události. Toto jim dovoluje analyzovat své předešlé zkušenosti a na základě toho dělat nová, více informovaná rozhodnutí. Mají tak určitou míru přizpůsobivosti a jsou schopny měnit své chování na základě pozorovaných vzorců. Jejich učení je ale často stále specifické na jednu užší oblast a nelze tyto naučené znalosti zobecnit pro aplikaci v širším spektru problémů [17].

Příklad použití je např. v tzv. autonomní vozidla nebo samořízená motorová vozidla, která jsou schopna vnímat své prostředí a kontrolovat jízdu vozidla na základě vyhodnocování těchto podnětů. Zároveň pak dokážou přizpůsobovat své reakce s ohledem na své předešlé zkušenosti [18].

- **Theory of Mind**

Tato úroveň umělé inteligence je prozatím pouze teoretická. Je to takový systém, jenž by dokázal porozumět lidským emocím, záměrům, či přesvědčením a dokázal by tedy komunikovat s lidmi s vysokou mírou sociální inteligence. Choval by se pro člověka mnohem více „přirozeně“ a byl by schopen empatie. Dokázal by se učit jak z analyzování dat, ale také chápat komplexní sociální situace a adekvátně se dle nich chovat [19].

Možná navrhovaná využití by mohla být např. virtuální asistenti s vysoce pokročilými sociálními schopnostmi, nebo vyučovací programy ve školství, které by se dokázali adaptovat na individuální potřeby studenta.

- **Self-Aware AI**

Obdobně jako u předešlého druhu umělé inteligence se i zde jedná o zatím čistě teoretický koncept. Z názvu vyplývající základní charakteristika tohoto systému je sebeuvědomění. Takovýto stroj by si byl vědom vlastní existence a měl své vědomí. Dokázal by se reflektovat nad svými zkušenostmi a rozhodnutími a měl by porozumění o svém rozhodovacím procesu. Získal by tak velkou míru samostatnosti, dokázal se zdokonalovat v odlišných oblastech na základě svého širokého pochopení a rozhodoval se s ohledem na své vlastní cíle [20].

3 Technologie XR

Termín Extended Reality, též zkracován na XR, je zastřešující pojem, jenž pod sebou shromažďuje technologie, které rozšiřují, obohacují a obecně zlepšují vnímání reality. Vytváří prostředí, v němž jsou spolu reálný a digitální (či virtuální) svět výrazně propojeny. Toto téma zpravidla dělíme na dvě hlavní oblasti: virtuální realita a rozšířená realita. Také existuje pojem smíšená realita, přičemž se z velké části jedná o synonymum pro rozšířenou realitu.

3.1 Virtuální realita

Jedná z možných definic virtuální reality, také pouze VR (Virtual Reality), je podnícení cíleného chování u subjektu pomocí umělé stimulace smyslů, přičemž subjekt si tohoto zásahu není vůbec, nebo jen málo vědom [21]. Důležité složky této definice jsou:

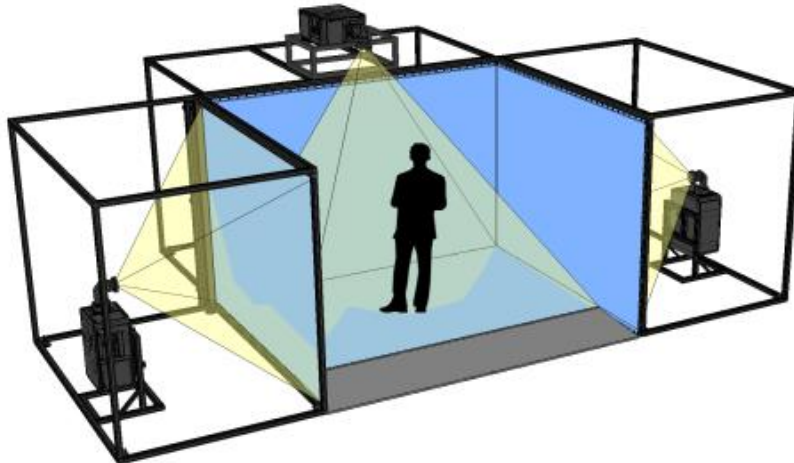
- **Cílené chování**
Subjekt prožívá zážitek, který byl navrhnout nějakým tvůrcem. Může se jednat např. o létání, chození, či sledování filmu.
- **Umělá stimulace smyslů**
Pomocí různých technologií mají jeden nebo více smyslů své běžné vstupy nahrazeny nebo vylepšeny pomocí umělé stimulace.
- **Povědomí**
Během zážitku si subjekt není vědom zásahu do jeho vnímání a připadá si skutečně přítomen v upraveném či plně nahrazeném světě.

Tohoto je ve většině případů dosaženo pomocí upevnění zařízení s obrazovkou (tzv. HMD – Head Mounted Display) k hlavě subjektu [22]. Tyto zařízení mají jednu nebo dvě obrazovky s čočkami v oblasti před očima uživatele.



Obr. 2 - Zařízení Oculus Rift S [23]

Možnou alternativou pro docílení efektu virtuální reality je využití speciálně navržených místností s několika projektory, či velkými obrazovkami, které obklopují subjekt ze všech stran. Takové systémy jsou též známy pod názvem CAVE (Cave Automatic Virtual Environment) [24].



Obr. 3 - Příklad zařízení CAVE [52]

Virtuální realitu s použitím HMD lze rozdělit na základě provedení zařízení, které je k jejímu docílení použito. Mezi dvě nejběžnější se řadí PC VR a Standalone VR.

3.1.1 PC VR

Též nazýván Tethered VR, tedy virtuální realita, u které je během celého procesu zařízení propojeno s počítačem pomocí kabelů. Veškerá výpočetní technika se nachází v tomto počítači a nemusí tak být limitována malými rozměry a určitým váhovým omezením, pokud by se nacházela přímo v HMD. To u PC VR dovoluje používat velmi výkonné počítače a tím pádem může být kvalita a komplexnost prostředí a činností ve virtuální realitě na velmi vysoké úrovni.

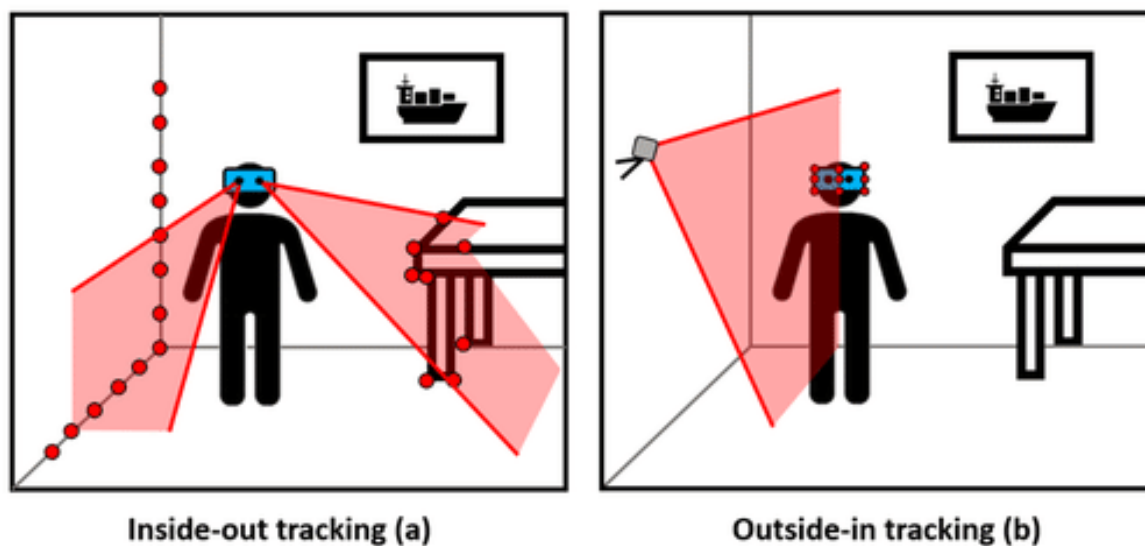
Proces sledování a zaznamenávání pozice a orientace HMD ve trojrozměrném prostoru a jeho následné převedení na pohyb ve virtuálním světě může být realizován dvěma hlavními způsoby [25].

- **Inside-Out Tracking (Integrated Tracking)**

V samotném HMD jsou umístěny senzory, která zachycují okolní prostor a dovolují tak systému sledovat svou pozici a orientaci. Jejich výhodou je jednodušší příprava před použitím a větší přenosnost. Příkladem je např. Oculus Rift S.

- **Outside-In Tracking (External Tracking)**

U tohoto druhu se v HMD nachází např. infračervené vysílače, které jsou snímány pomocí externích senzorů. Výhodou je zpravidla lepší přesnost a odezva. Nutnost umístění samostatných senzorů ale prodlužuje a komplikuje přípravu před použitím, a také je tzv. playspace (oblast ve které lze zařízení využívat) omezen na oblast mezi těmito senzory. Příkladem jsou např. HTC Vive a Valve Index.



Obr. 4 - Způsoby sledování pozice HMD [26]

Neustálá propojenost s počítačem, a tím pádem i zdrojem napájení dovoluje nepřetržitý provoz a zároveň díky ní odpadá nutnost začlenění velké baterie do HMD, čímž je snížena váha a je získán větší prostor pro ostatní komponenty. Samotná nutnost propojení je však i jeho největší nevýhodou, jelikož má za následek menší mobilitu a horší míru komfortu během užívání [27].

3.1.2 Standalone VR

Novější formou virtuální reality je Standalone VR, také známé jako Untethered VR. Využívá bezdrátového HMD, které je schopno fungovat bez nutnosti připojení k externí výpočetní technice. Veškerý potřebný hardware, což mimo jiné zahrnuje centrální a grafický procesor, se nachází přímo uvnitř zařízení. U Standalone VR je využíván výše zmíněný Inside-Out Tracking. Jelikož zařízení není propojeno s žádnými dalšími samostatnými snímači, není Outside-In Tracking možný.

Díky snaze minimalizovat rozměry a váhu těchto zařízení z důvodu komfortnějšího užití, je jejich výkonnost omezena a je tak podstatně nižší než u PC VR. Díky tomu jsou nevhodné pro velmi technicky náročné aplikace. Další podstatnou nevýhodou je omezená doba provozu, která je limitována kapacitou baterie zabudované do HMD, která potřebuje být pravidelně dobíjena a nelze tak zařízení využívat nepřetržitě.

Mezi největší klady Standalone VR patří jejich vysoká přenosnost, jelikož odpadá nutnost přípravy externích snímačů v prostoru jejich použití. Dále již není problémem propojovací kabel, který v mnoha případech může značně omezovat pohyb subjektu. Tyto vlastnosti značně zlepšují ergonomii během provozu [27]. Příkladem Standalone VR zařízení jsou např. Meta Quest 2 (dříve známé jako Oculus Quest 2) a HTC Vive Focus 3.



Obr. 5 - Zařízení Meta Quest 2 [53]

3.2 Rozšířená realita

Také zkracováno na AR (Augmented Reality). Na rozdíl od virtuální reality, při které je vnímání subjektu kompletně nahrazeno, se u rozšířené reality jedná o obohacení vnímání skutečného světa. V obraze, který subjekt vidí, se pak prolíná skutečný svět s tím virtuálním. Počítačově vygenerované objekty překrývající skutečný svět mohou být buď konstruktivní (ke skutečnému světu něco přidávají) nebo destruktivní (ze skutečného světa něco zakrývají) [28].

Existují tři hlavní zobrazovací zařízení, která jsou využívána u rozšířené reality: Head Mounted Displays (HMD), ruční zařízení a Spatial Augmented Reality (SAR) [29].

3.2.1 Head Mounted Display

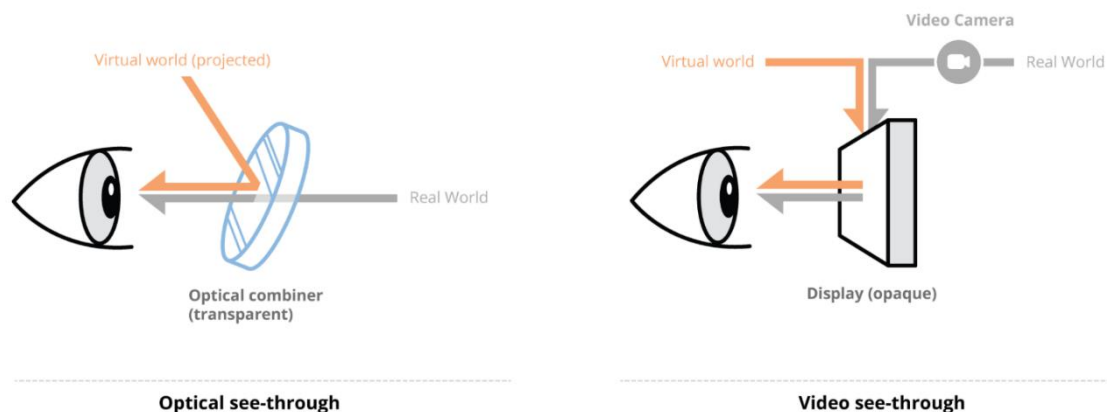
Za použití zařízení upevněného k hlavě uživatele (HMD) se zobrazuje spojení reálného okolního světa a virtuálně přidaných prvků. Tyto zařízení mohou být monokulární nebo binokulární. Tento proces může fungovat na dvou odlišných principech:

- **Optical See-Through**

Skrze průhlednou čočku je uživatel stále schopen přímo vidět skutečný svět, ale k jeho pohledu jsou opticky přidány objekty promítané na čočce. Okolní prostředí je vnímáno více přirozeně než jen pouze přes obrazovku, ale promítané obrazy mohou mírně „zaostávat“ za skutečnými během pohybu kvůli prodlevě způsobené zpracováváním obrazu v procesoru HMD.

- **Video See-Through**

Neprůhledná zařízení, ve kterých se nachází kamery snímající okolní prostor, který je převáděn do digitální formy a poté kombinován s informacemi, které mají být uživateli zobrazeny. Toto je technicky náročnější, ale dovoluje to větší míru kontroly nad načasováním pohybu zobrazených objektů a tím pádem lepší synchronizaci „skutečné“ a přidané složky pohledu.



Obr. 6 - Srovnání provedení HMD AR [54]

3.2.2 Ruční zařízení

Jsou to malá a velmi přenosná zařízení, která obsahují procesor a obrazovku. Fungují na principu Video See-Through a využívají zabudované senzory na sledování vlastního pohybu v prostoru. Nejčastěji využívaná provedení se dají rozdělit do tří kategorií: chytré telefony, PDA (Personal Digital Assistant) a tablety [29].

Moderní chytré telefony s výkonnými procesory a kamerami společně se zabudovaným akcelerometrem jsou velmi vhodné pro využití na zobrazení rozšířené reality, ale jejich relativně malá velikost obrazovky může být limitujícím faktorem pro některé aplikace. PDA zařízení mají v tomto smyslu podobné charakteristiky, akorát jsou to specializovaná zařízení určena pouze na tuto funkci. Tablety jsou zpravidla vybaveny více výkonným hardwarem a také podstatně větší obrazovkou, ale jejich přenosnost je menší než u předešlých dvou druhů.



Obr. 7 - Ruční zařízení na rozšířenou realitu [55]

3.2.3 Spatial Augmented Reality

Za pomoci projektorů, hologramů a optických prvků jsou grafické informace zobrazovány přímo na fyzických předmětech, čímž vzniká interaktivní a dynamické prostředí. Uživatel tedy nemusí držet či mít nasazenu obrazovku [30]. Díky tomuto lze SAR využívat nejenom pro jednotlivce, ale i skupinu současných uživatelů.

K dosažení věrohodného efektu rozšířené reality je často systém vybaven technologiemi na snímání a mapování geometrie předmětů v prostoru, k čemuž je nutná přesná kalibrace jeho senzorů a kamer. Kvůli tomuto jsou často SAT systémy limitovány na své přenosnosti oproti ostatním provedení rozšířené reality.

4 Nástroje využívající umělou inteligenci

Velký pokrok v oblasti umělé inteligence v nedávné době měl za následek její průnik do oblasti vývoje aplikací. Tato technologie našla využití právě zde hlavně kvůli jedné z jejich hlavních vlastností, kterou je všestrannost. Různorodé nástroje založené na umělé inteligenci jsou využívány ve všech fázích vývoje aplikací a tento proces se tak stává rychlejším, efektivnějším a více dynamickým.

Tyto nástroje se dají klasifikovat jako tzv. generativní umělá inteligence (GenAI – Generative Artificial Intelligence), tedy takové programy, jež jsou schopné tvořit text, obrázky a jiná média [31]. Jsou založeny na principu strojového učení (Machine Learning), u kterého dostane model přístup k velkému souboru dat, na kterém je „učí“ a je poté schopen tyto zkušenosti zobecnit a aplikovat na řešení nových úkolů. Obecně lze definovat tři přístupy k Machine Learning [32].

→ **Supervised Learning** (učení s učitelem)

U všech vstupů jsou „učitelem“ definovány žádoucí výstupy a cílem je, aby se model naučil obecné pravidlo, které přiřazuje vstupy k výstupům.

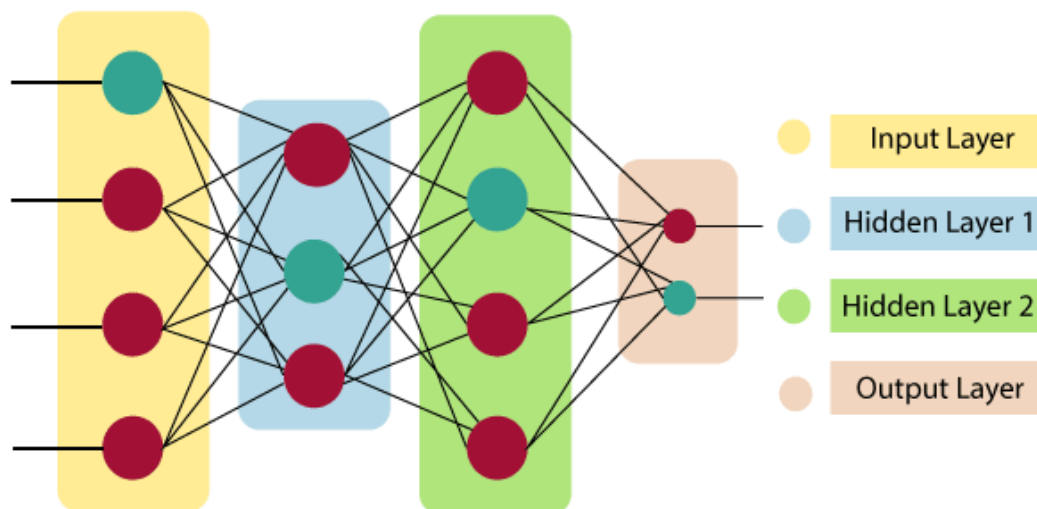
→ **Unsupervised Learning** (učení bez učitele)

Vstupní data nemají přiřazený žádný výstup a cílem je najít určitou strukturu ve všech přijatých vstupech.

→ **Reinforcement Learning** (učení se zpětnou vazbou)

Model interaguje s dynamickým prostředím, ve kterém má za úkol splnit daný úkol. Při provádění tohoto úkolu dostává zpětnou vazbu na způsob odměn a snaží se tyto odměny maximalizovat.

V nedávných letech nejčastěji využívaný výpočetní model je tzv. umělá neuronová síť (ANN – Artificial Neural Network nebo jen NN – Neural Network). Ten dovoluje simulovat komplexní vztahy mezi řadou vstupů a výstupů. Skládá se z neuronů, které jsou navzájem propojeny synaptickými vazbami, pomocí nichž si předávají informace. Ty jsou rozděleny do několika vrstev, přičemž různé vrstvy zpravidla vykonávají jiné transformace s jejich vstupy. Informace jsou přijímány ve vstupní vrstvě, procházejí jednou či více skrytými vrstvami a dostávají se do výstupní vrstvy. Jednotlivé neurony a vazby mají svou váhu, která se může zvětšovat či zmenšovat. Tato váha pak určuje velikost vlivu, jež má jeden článek na jiný [33].



Obr. 8 - Struktura umělé neuronové sítě [34]

4.1 Nástroje na programování

V oblasti programování jsou nástroje umělé inteligence využívány k výpomoci uživateli, urychlení tohoto procesu a automatizaci často se opakujících úkonů. Nejčastěji tomu tedy je u tvorby skriptů (programů napsaných ve skriptovacím jazyce).

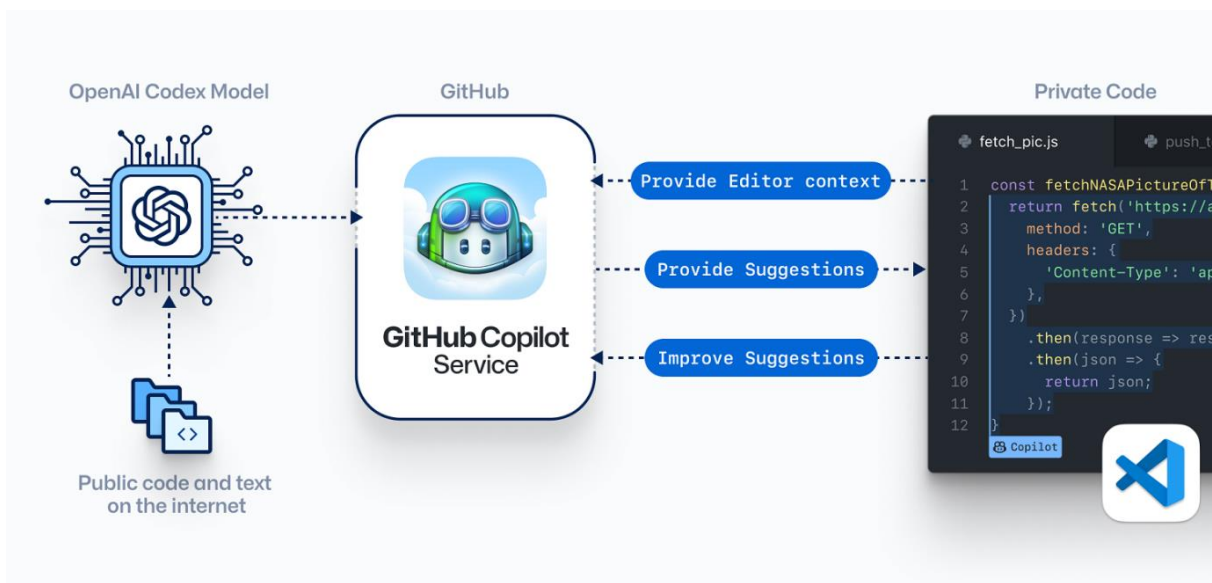
Skriptovací jazyk lze definovat jako takový, který se lze rychle naučit a jde ho jednoduše používat. Skripty jsou často spíše menší a jednodušší a tvoří části většího programu [35]. Jsou používány především na automatizaci úkonů či upravování prostředků již stávajícího systému.

Nástroje na bázi umělé inteligence mají u tohoto procesu široké využití. Nejběžněji to je automatické dokončování kódu. Jedná se o nástroje, které jsou zabudovány přímo do softwaru na tvorbu kódu. Chápe kód, který již byl napsán, a během dalšího psaní pak uživateli v reálném čase doporučují, co by měl dále psát. Jsou schopny se učit a po určitém čase se tak adaptují na styl psaní daného uživatele. Na podobném principu také funguje proces generování kódu, u něhož nástroje dokážou vytvořit i poměrně velké části kódu na základě výzvy od uživatele a případně ho po dalších výzvách upravovat dle jeho požadavků.

Dalším důležitým využitím AI nástrojů je jejich schopnost optimalizovat a refaktorovat (provádět změny ve struktuře kódu bez ovlivnění jeho původní funkcionality) kód. Nástroj nejprve zanalyzuje celý uživatelský kód a poté identifikuje chyby či potenciální problémy a navrhně pro ně řešení nebo alternativní provedení. Dovolují tak více interaktivní a intuitivní přístup k řešení problémů v oblasti programování.

V neposlední řadě lze také tyto nástroje využít k automatickému generování dokumentace pro kód aplikace. Díky interpretaci struktury kódu a kódu samotného jsou schopny navrhnout velmi přehlednou dokumentaci, čímž uživateli ušetří velké množství času.

Příkladem je např. GitHub Copilot, AlphaCode nebo Tabnine.



Obr. 9 - Princip využití umělé inteligence na programování [36]

4.2 Nástroje na tvorbu textur

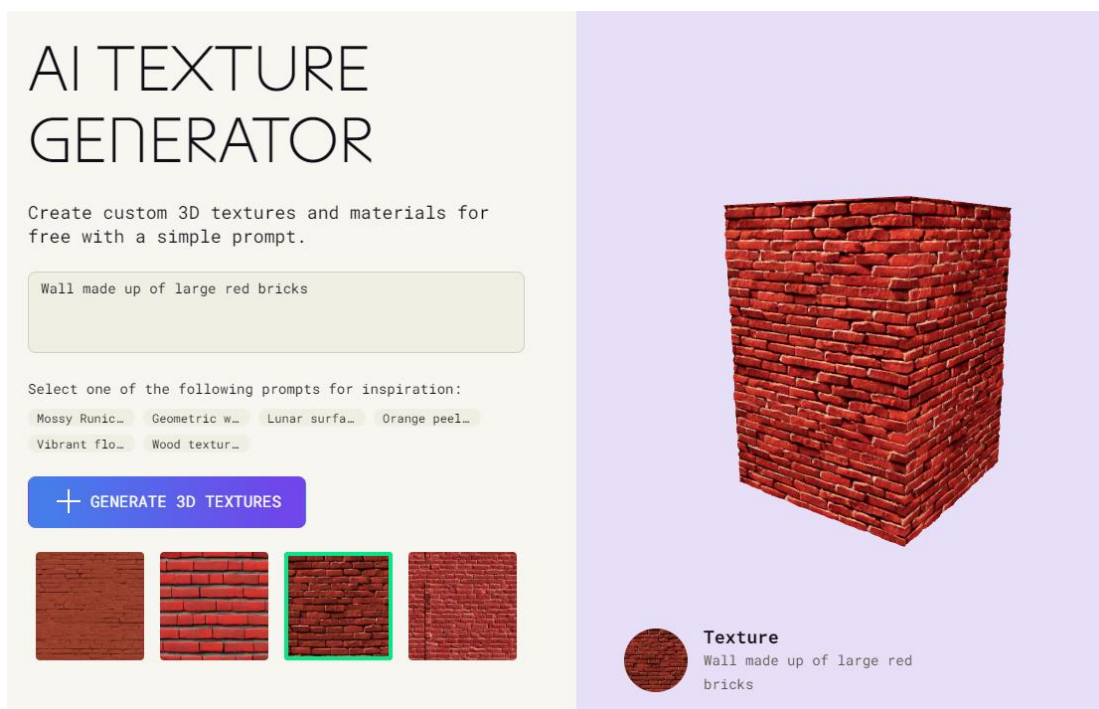
Jedná se o nástroje schopné vytvářet textury (popisuje vizuální vzhled dvourozměrného či trojrozměrného povrchu v počítačovém prostředí, tedy jeho vlastnosti jako např. strukturu a barvu) na základě pokynu, který popisuje, jak by daná textura měla vypadat. Datové soubory, na kterých se modely pro tuto práci určené „učí“ obsahují velké množství obrázků a textur různých materiálů, povrchů či vzorů.

Uživatel je často schopen nastavovat různé parametry vytvářecího procesu a někdy i v reálném čase vidět, jaký specifický efekt tyto parametry mají na výslednou texturu. Běžně je vygenerováno výsledků hned několik a uživatel si může vybrat ten, který nejvíce odpovídá jeho požadavkům. S využitím těchto nástrojů je tak možný více interaktivní přístup k procesu tvorby textur, který je zároveň značně urychlen a zefektivněn.

Dalšími z jejich dalších častých funkcí je schopnost zvětšit rozlišení (upscaling) a obecně vylepšit kvalitu textury, kterou jí uživatel předloží, nebo také schopnost změnit charakteristiky textury tak, aby se podobala určité předloze. Texturovací nástroje jsou běžně integrovány přímo do 3D modelovacích softwarů, což zaručuje větší míru pohodlnosti jejich užívání.

Texturovací nástroje převážně využívají principu tzv. generativních adversiálních sítí (GAN – Generative Adversarial Network). Ty dosahují generování požadované cílové položky s využitím dvou modelů. Prvním modelem je samotný generativní, který na základě zadaného souboru dat vytváří požadované výstupy. Druhý model, tzv. diskriminátor, pak porovnává míru podobnosti položky vygenerované prvním modelem s těmi v původním souboru dat. Iterací tohoto procesu se pak generativní model zlepšuje ve tvorbě požadovaných výstupů a diskriminátor získává schopnost lépe rozpoznávat skutečné položky od těch vygenerovaných modelem umělé inteligence [37].

Dostupné texturovací nástroje jsou např. Poly [38], Texture Lab [39] a Polycam [40].



Obr. 10 - Příklad nástroje na tvorbu textur [40]

4.3 Nástroje na tvorbu modelů

Jsou schopny na základě vstupu, kterým je buď text nebo obrázek, vygenerovat trojrozměrné modely pro nejrůznější použití v počítačovém prostředí. Obdobně jako u texturovacích nástrojů jsou založeny na principu generativní adversiální sítě. Některé generátory modelů mohou i zároveň samotné generátory textur obsahovat, a tedy jejich výstupem bude 3D model i s texturou. K tomuto využívají svou schopnost velmi dobře mapovat textury na model, tedy přiřazovat určitým místům na povrchu modelu správné textury. Další možností některých nástrojů je také tvorba specifické animace pro vygenerovaný model.

Proces tvorby modelu pomocí těchto nástrojů je značně urychlen oproti manuální tvorbě. Díky tomuto mají největší využití např. v oblastech virtuální reality, návrhu produktů či architektury, kde mohou být použity na vytváření prototypů a detailních modelů nebo celých virtuálních prostředí.

Příklady těchto nástrojů jsou např. 3DFY [41], Masterpiece X [42] nebo Point-E [43].



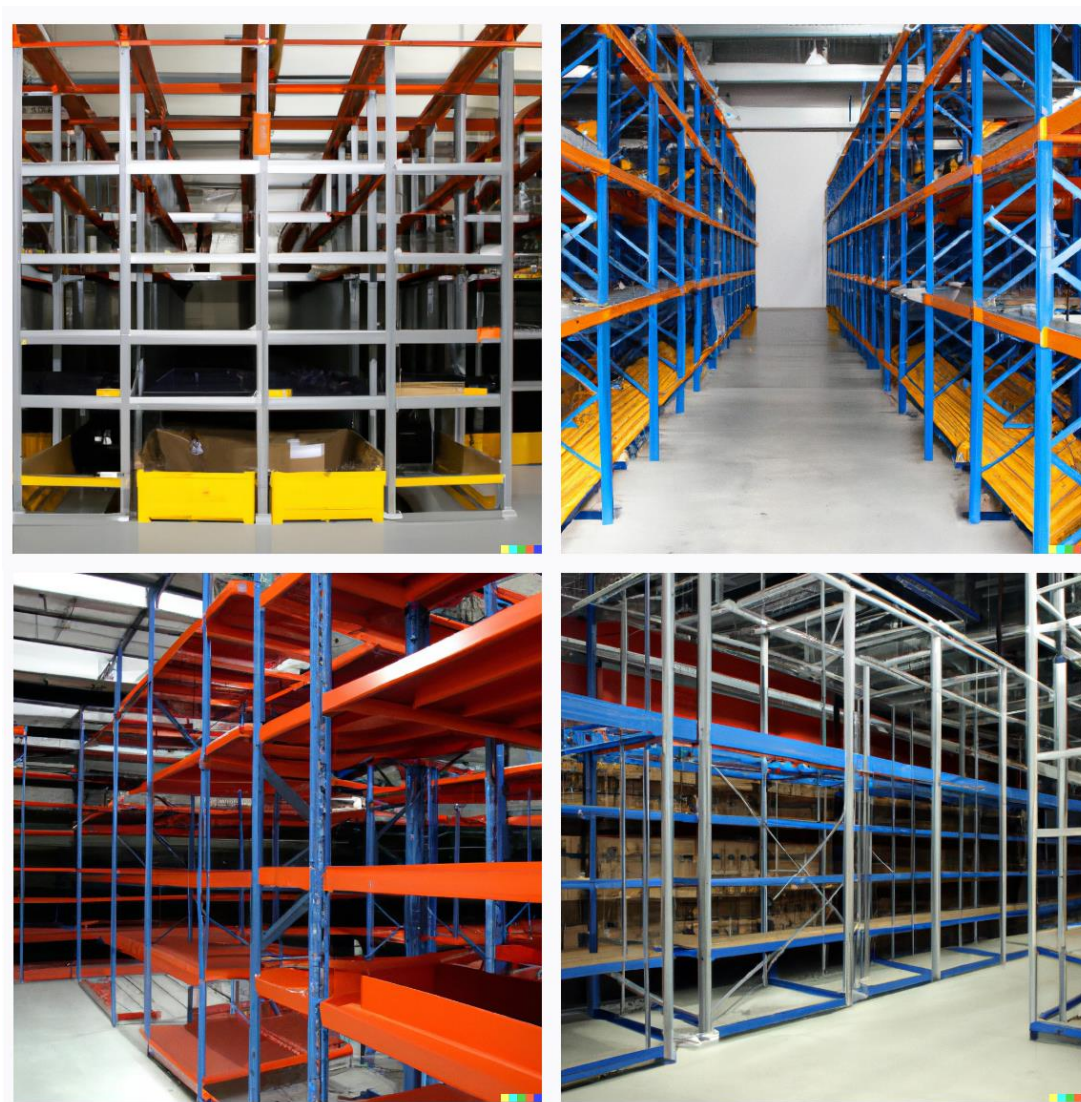
Obr. 11 - Modely vygenerované pomocí AI [41] [42]

4.4 Nástroje na tvorbu obrázků

Tyto nástroje, obdobně jako ty předchozí, využívají GAN na tvorbu 2D, či 2+1D (dvojrozměrný obraz rozšířený o informace o hloubku) obrázků. Je tak možné vytvořit nejen běžné obrazy, ale i panoramatická 360° prostředí, která lze využít například ve virtuální realitě. Jako vstup je přijímán buď text, který obraz popisuje, nebo obrázek jiný, na základě něhož je vytvořen nový.

Dalšími funkcemi je také např. upravování obrázků, což může zahrnovat odstraňování nežádoucích prvků, přidávání nových prvků, zvýšení rozlišení a další. Již zmíněné 360° snímky mohou být buď vytvořeny na základě textového popisu, nebo lze programu zadat řadu fotografií, které má sjednotit dohromady a vytvořit z nich jednu.

Příklady těchto nástrojů jsou Stable Diffusion [44] a DALL-E [45] (pouze 2D obrázky) nebo CopernicAI [46] a Skybox Lab [47] (panoramatické obrázky).



Obr. 12 - Obrázky vygenerované pomocí AI [45]



Obr. 13 - Panoramatický snímek vytvořen pomocí AI [47]

5 Návrh aplikace

V této části práce je vysvětlena navrhovací fáze, ve které byl zvolen specifický proces, který bude modelován. Dále byl vytvořen podrobný scénář, popisující průběh tohoto procesu, a také je popsán samotný proces tvorby aplikace.

Aplikace je vytvořena v softwaru Unity3D, do které budou veškeré ostatní objekty či jiné druhy médií z jiných zdrojů importovány. Ty zahrnují např. 3D modely ze softwaru Blender nebo AI nástrojů na generování modelů a také audio soubory obsahující zvukovou instruktaž, která byla vytvořena s pomocí AI nástroje.

Pracovní postup pro vytvoření celé aplikace byl následující:

- **Volba tématiky** – V této fázi bylo vybráno několik možných činností, které se dají převést do virtuálního prostředí a je možné pro ně vytvořit aplikaci sloužící na zaučení pracovníků. Bylo nutné zvážit komplexnost jednotlivých činností a také užitečnost vyučování ve virtuální realitě oproti tomu ve skutečném světě.
- **Tvorba scénáře** – Po zvolení specifické činnosti byl vytvořen detailní scénář popisující veškeré dílčí úkony, které se v aplikaci nacházejí. Scénář obsahuje popis prostředí, jednotlivé změny v něm, zvukovou instruktaž a požadavky kladené na uživatele.
- **Příprava prostředí a objektů** – Tato část obsahovala návrh vhodného virtuálního prostředí s ohledem na prováděnou činnost a jeho následné vytvoření, včetně všech 3D modelů a jim přidělených textur, které se v aplikaci budou nacházet.
- **Nastavení scény** – Nyní bylo samotné prostředí zkompletováno za pomoci předem vytvořených objektů. Dále v něm bylo nastaveno osvětlení a připraven prostor, ve kterém bude uživatel manipulovat s danými objekty.
- **Tvorba aplikace** – Následující krok je samotné vytvoření aplikace. Zde se především zabezpečuje požadovaná funkcionalita aplikace pomocí vytvoření a implementace skriptů.
- **Testovací fáze** – Na závěr probíhá podrobné testování aplikace, aby se prověřilo, že všechny na ni kladené požadavky jsou splněny.

V následujících podkapitolách jsou detailněji popsány jednotlivé kroky vývojového procesu.

5.1 Návrh činnosti

Prvotním krokem v celém vývojovém cyklu aplikace byl samotný návrh dané činnosti, která bude převedena do virtuálního prostředí. Pro tento účel byla zvolena finální montáž LED televizoru prováděná pracovníkem v prostředí průmyslového podniku.

Jedná se o dobře modelovatelnou činnost, jelikož neobsahuje velké množství komponent a ty, které obsahuje, jsou relativně jednoduššího charakteru. Přínos vyučování pracovníků na této montáži ve virtuální realitě spočívá také v tom, že několik z komponent jsou elektronické komponenty, které jsou náchylné na poškození. Díky prvotnímu procvičení ve virtuálním prostředí se tak minimalizuje možnost nechtěného znehodnocení jedné z montovaných součástí.

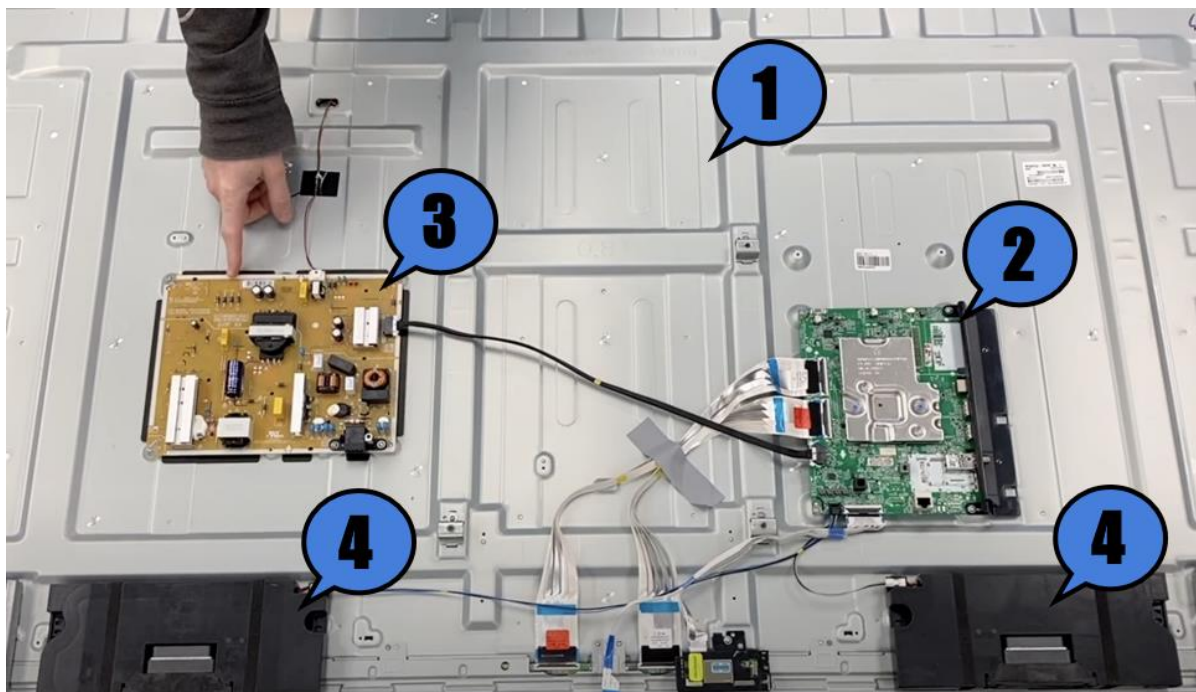
Televizor se skládá z velkého množství komponent a prochází řadou komplexních kroků v průběhu jeho výroby, ale všechny tyto kroky se zpravidla dají rozdělit na několik pracovišť, na kterých se odehrávají. V této práci bude tedy zpracováno poslední pracoviště z pohledu výrobního procesu, na kterém se již provádí kompletace samotného televizoru do finální

podoby. Tuto činnost provádí pracovník v pozici ve stoje u pracovního stolu, na kterém má dostatečný prostor pro vhodnou manipulaci se součástkami.

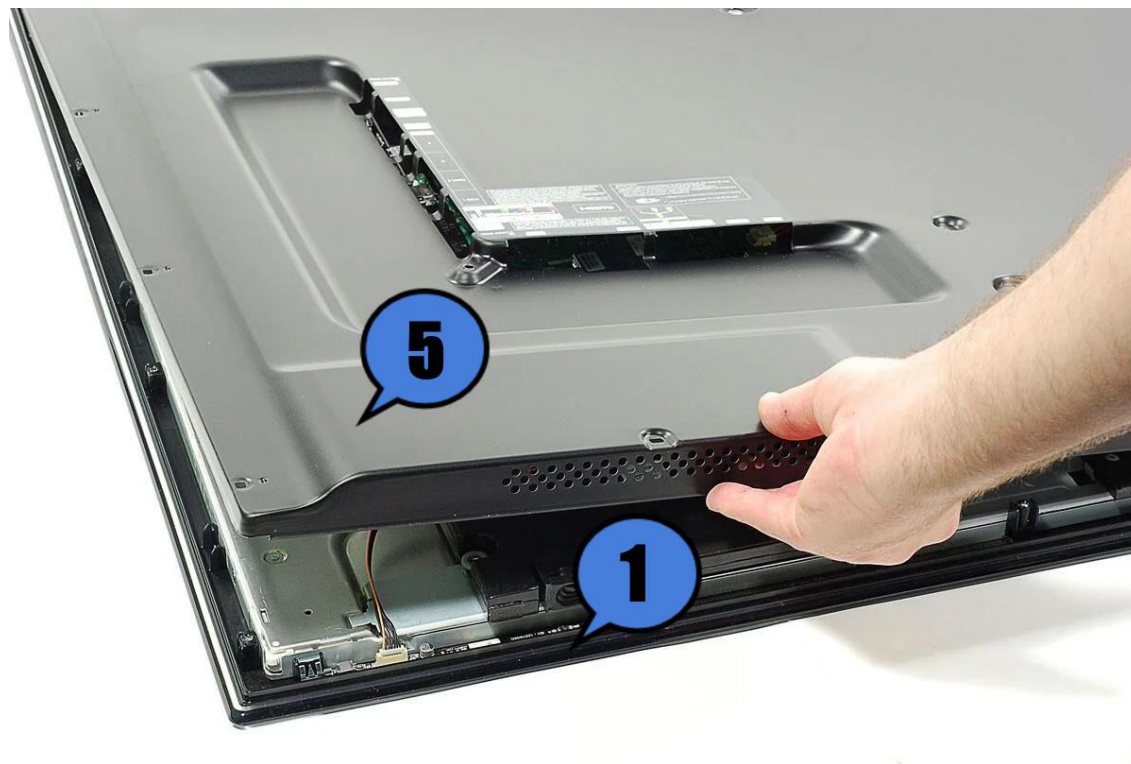
V modelovaném procesu se pracuje s těmito komponenty:

- **Rám (1)** – Připevňují se k němu ostatní komponenty. Zpravidla je zhotoven z tvrdého plastu nebo hliníku, případně kombinace obou těchto materiálů. Je obdélníkového tvaru, z přední strany má připevněnu LED obrazovku a ze zadní strany má připravena místa pro nainstalování ostatních součástí.
- **Základní deska (2)** – Základní část hardwaru televize. Propojuje jednotlivé elektronické součástky a řídí je. Nachází se zde důležité komponenty jako procesor či operační paměť. Zároveň je na ni zhotoven panel s potřebnými porty pro běžné konektory na televizorech jako je HDMI, DisplayPort apod.
- **Zdrojový modul (3)** – Do této komponenty je přiváděn proud z napájecího kabelu a dále je odsud rozváděn do všech ostatních míst celé televize. Je typicky oranžové/bronzové barvy, aby byla lehce odlišitelná od základní desky. Přivedené střídavé napětí z běžné zásuvky je zde přeměněno na stejnosměrné napětí.
- **Reproduktory (4)** – Obecně se nacházejí jeden až dva na každém televizoru, v tomto případě jsou modelovány reproduktory dva. Jsou umístěny na spodní nebo boční části rámu, aby zvuk jimi produkovaný nebyl zakrýván některou z jiných komponent.
- **Zadní krycí panel (5)** – Překrývá všechny vnitřní komponenty, aby byly chráněny od znečištění nebo nežádané manipulace. Zpravidla je také plastový nebo hliníkový a k rámu se připevňuje pomocí několika šroubů.

Výše popsané komponenty jsou označeny na následujících obrázcích:



Obr. 14 - Komponenty uvnitř LED televize [56]



Obr. 15 - Připevnění zadního krytu k rámu [57]

5.2 Průběh scénáře ve virtuálním prostředí

Poté co si uživatel nasadí brýle pro virtuální realitu, ocitne se v prostředí připraveném pro simulovanou montáž. Jedná se o místnost, ve které se nachází pracovní stůl a na něm rozložené komponenty, se kterými bude zacházeno. Přímou nad stolem se nachází tlačítko start, kterým může uživatel celý proces započít.

Po zahájení se spustí hlas, který uživatele uvítá a vysvětlí mu základní poznatky nutné pro orientaci a manipulaci s předměty ve virtuálním prostředí. Tento hlas bude též navádět a provázet uživatele celou montáží. Veškerá hlasová instruktaž je vygenerována pomocí nástroje umělé inteligence.

V návaznosti na prvotní zaškolení musí uživatel provést první krok, kterým je instalace zdrojového modulu a základní desky. Ty jsou na základní rám přiloženy do správné pozice a poté v označených místech přišroubovány. Po jejich úspěšnému přimontování je po uživateli požadováno je propojit kabely.

Následně se upevní dva reproduktory na spodní stranu rámu a taktéž je provedeno jejich připojení kabelem. Reprodukory se pouze umístí do zvýrazněné pozice a s mírným přitlačením se zapasují rámu.

V neposlední řadě je nainstalován zadní krycí panel. Poté je montáž televize dokončena. Uživateli je sděleno, jak dlouho mu celá montáž trvala pomocí zobrazené časomíry před ním na stole. Také může případně celou aplikaci restartovat pomocí nově dostupného tlačítka.

Detailní rozepsání celého scénáře se nachází v kapitole 2.3 Tvorba hlasového popisu postupu.

6 Tvorba aplikace

V této části je vysvětlen proces tvorby aplikace a jsou zde detailně popsány jednotlivé kroky vývoje s důrazem na využití nástrojů umělé inteligence. Jedná se o tvorbu prostředí, ve kterém se uživatel nachází, a také předmětů, se kterými je během montáže manipulováno. To zahrnuje vytvoření 3D modelů a také textur pro dané modely.

Také je zde popsán postup při vytvoření hlasové instruktáže podle scénáře a její následná implementace.

Poté následuje proces dosažení požadované funkcionality, a to zpravidla pomocí programování skriptů pro interakci uživatele s prostředím a jejich následné správné nastavení.

6.1 Tvorba prostředí

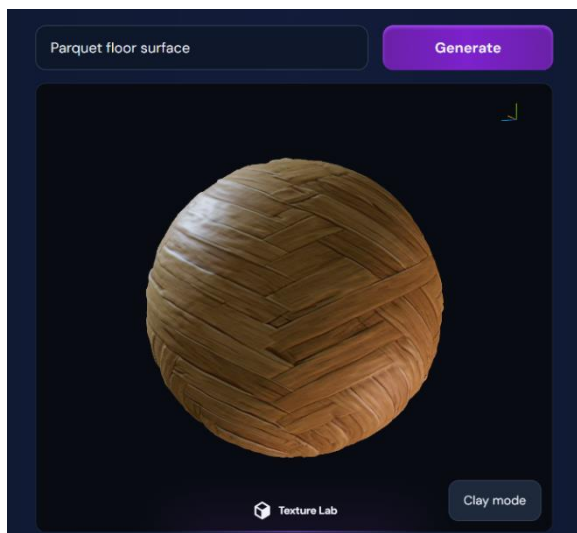
Počátečním krokem je vytvoření celého virtuálního prostoru, ve kterém se bude montáž odehrávat. Jako zasazení byla zvolena menší čtvercová místnost s párem oken, ve které je umístěn pracovní stůl.

V místnosti se na stropě nacházejí čtyři lampy, které ji dodávají potřebné osvětlení, což zabezpečuje, že všechny předměty lze lehce identifikovat a vidět.

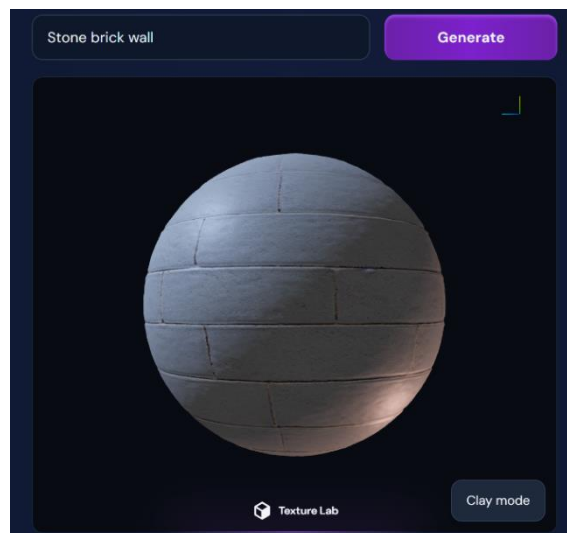


Obr. 16 - Prvotní podoba prostředí

Tvorba textur pro podlahu, stěny a strop byla provedena pomocí nástroje TextureLab [39]. V něm byly specifikovány požadované textury a nástroj je následně vygeneroval. Poté byly ve formátu 2D snímku textury a snímku udávající odražení světla od povrchu importovány do Unity3D. Následně byl vytvořen nový materiál obsahující tyto textury a ten byl přiřazen odpovídajícím objektům.



Obr. 17 - Textura podlahy [39]



Obr. 18 - Textura zdi [39]

Pro podlahu byla vytvořena textura na základě popisu „Parquet floor surface“ (parketová podlaha) a pro zeď podle popisu „Stone brick wall“ (zeď z kamenných kostek).

Dále se v místnosti nachází samotný stůl a také stropní osvětlení. Tyto modely byly vytvořeny přes nástroj LumaLabs [48]. Ten pracuje na principu zadání popisu požadovaného předmětu, ke kterému následně vygeneruje čtyři varianty. Uživatel může následně zvolit variantu, která nejvíce odpovídá jeho požadavkům. V dalším kroku je k dispozici menu, ve kterém lze určit kvalitu modelu a formát, ve kterém bude exportován. Veškeré textury jsou vytvořeny současně a modelům rovnou přiřazeny. Pokud je zvolen formát modelu, do kterého lze textury zasadit, je tak učiněno, a pokud ne, jsou textury vygenerovány zvlášť jako 2D snímek.

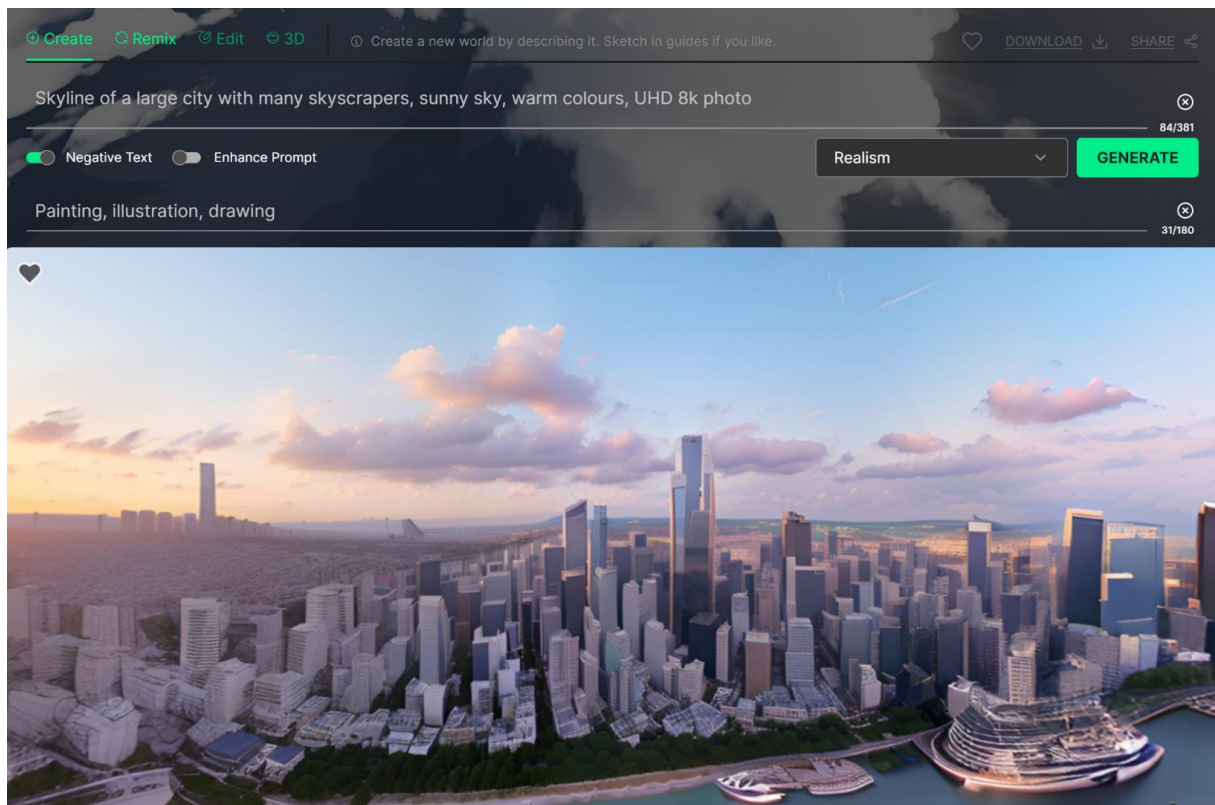


Obr. 19 - Generování pracovního stolu [48]



Obr. 20 - Vygenerovaný model a možnosti [48]

Následně bylo za pomoci nástroje BlockadeLabs Skybox AI [47] vytvořeno okolní prostředí mimo místnost. Konkrétně byl použit výhled na velkoměsto s vysokými budovami. Snímek je vytvořen ve formě ekvidistantní válcové projekce, a je následně importován a přiřazen k materiálu, který je použit na texturování tzv. skyboxu. Tím se vytvoří efekt prostorového snímku.



Obr. 21 - Generování prostorového snímku [47]

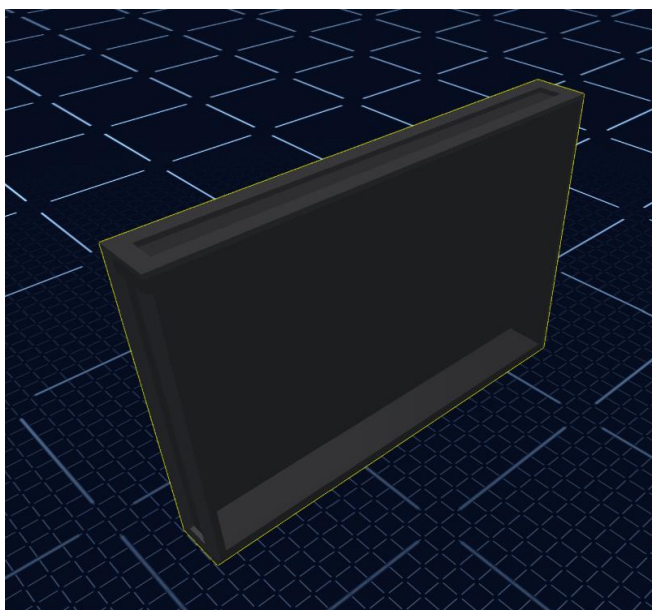
6.2 Tvorba předmětů v prostředí

V této fázi byly vygenerovány veškeré předměty potřebné pro simulovanou montáž. Primárně se jedná o komponenty, které jsou skládány dohromady (rám, základní deska, zdrojový modul, reproduktory, zadní kryt), ale také se jedná o doprovodné předměty jako např. šroubovák potřebný k upevnění jednotlivých desek k rámu a další.

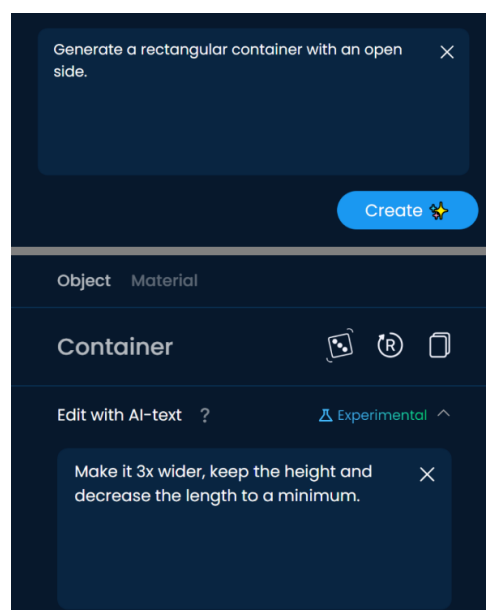
Tyto modely se dají rozdělit do dvou kategorií:

- **Modely s jednoduchou geometrií** – zpravidla jsou to ty, které se skládají pouze z rovných ploch a pravidelných hran
- **Modely se složitější geometrií** – tyto modely mají komplexnější tvar a mnohem větší množství ploch

Jednodušší modely, které zahrnují obdélníkový rám televize nebo obrazovku byly vygenerovány v nástroji SloydAI [49]. Jedná se o nástroj obsahující velkou knihovnu 3D modelů, které mají každý velké množství parametrů pro přizpůsobení požadovanému využití. Ve vyhledávacím okně se zadá popis požadovaného modelu, který je interpretován pomocí AI a následně vhodně zvolen. Poté je možné dalšími pokyny pro AI systém vytvořený model dále modifikovat a zároveň pro něj zvolit vhodnou texturu.



Obr. 22 - Výsledná podoba vygenerovaného rámu [49]

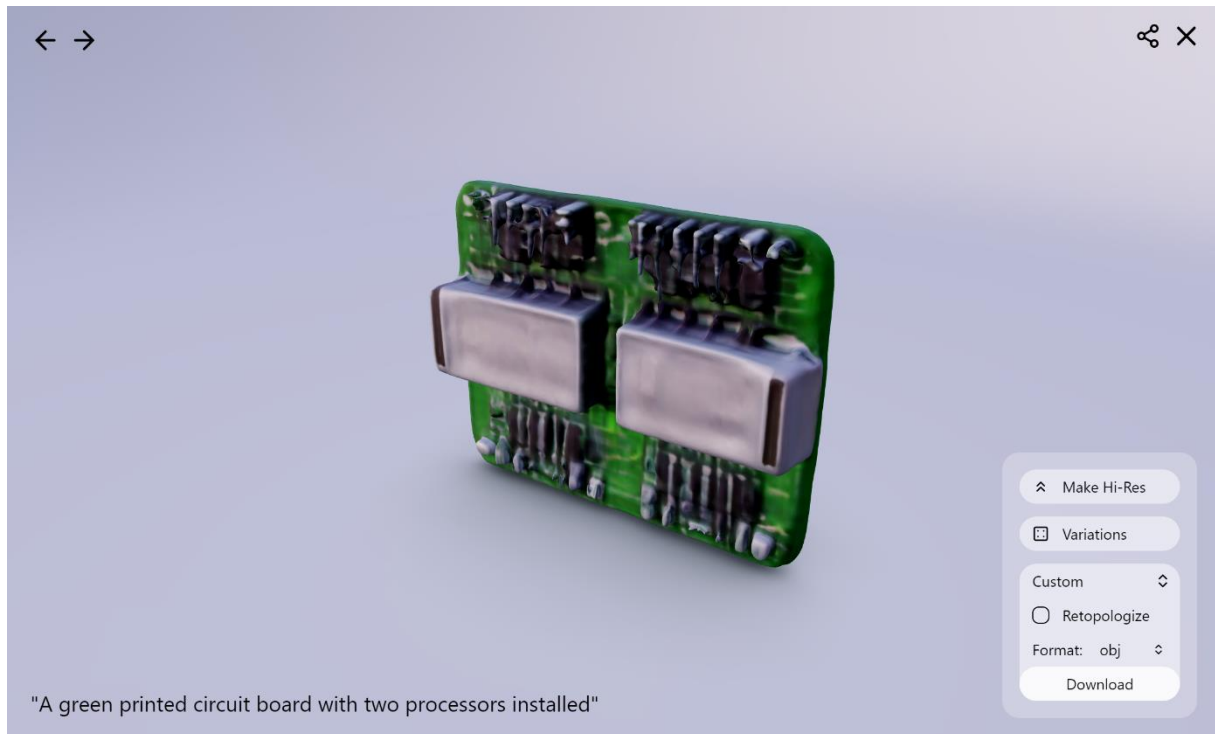


Obr. 23 - Textové pokyny pro generování

Modely složitějších komponentů byly vytvořeny za pomoci nástroje LumaLabs [48]. Nejdříve byly detailně popsány a následně byl zvolen nejvíce vyhovující výsledek po několika iteracích generování.

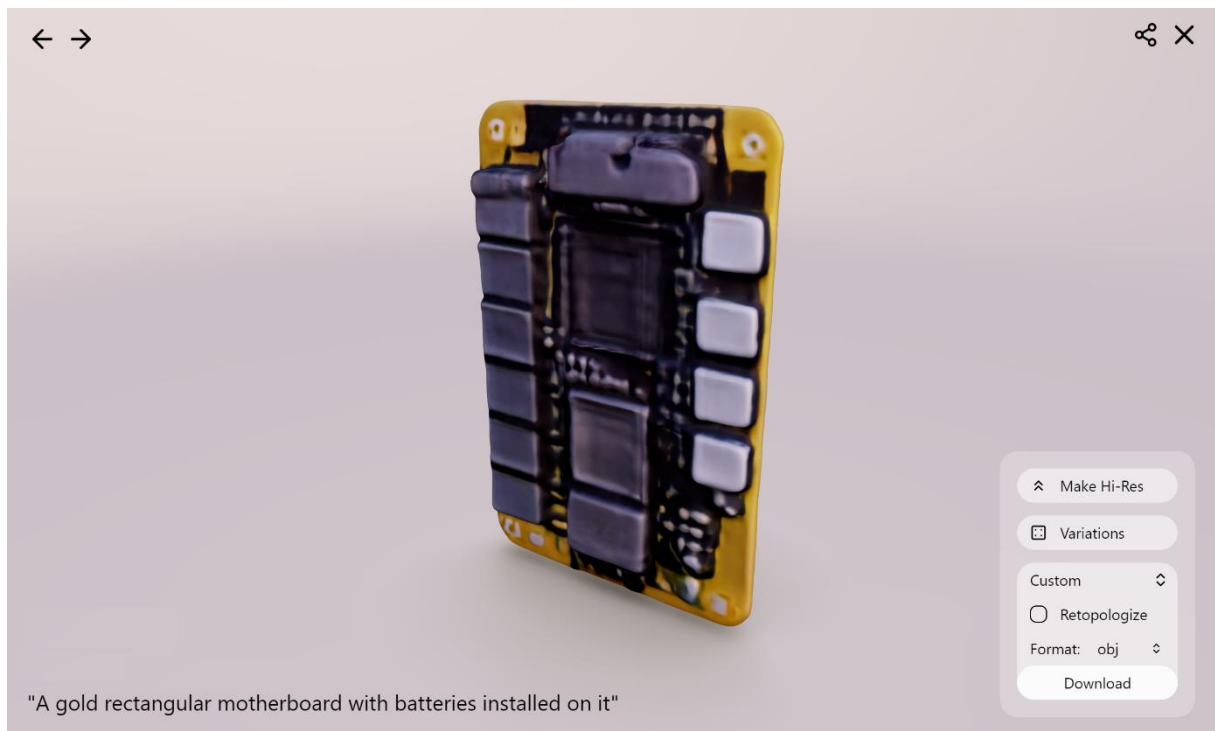
Takto vzniklé modely obsahují velké množství polygonů a pokud by byly rovnou v tomto stavu použity v aplikaci, mělo by to za následek dlouhé načítací časy a také by kladly zbytečně velké nároky na výkon zařízení, ve kterém by aplikace byla přehrávána. Z tohoto důvodu je tedy nejdříve nutné modely optimalizovat.

Nejdříve byla vygenerována základní deska, která je v montážním postupu první komponentou, se kterou bude uživatel manipulovat.



Obr. 24 - Model základní desky [48]

Následně byl vytvořen zdrojový modul, který bude namontován hned po základní desce.



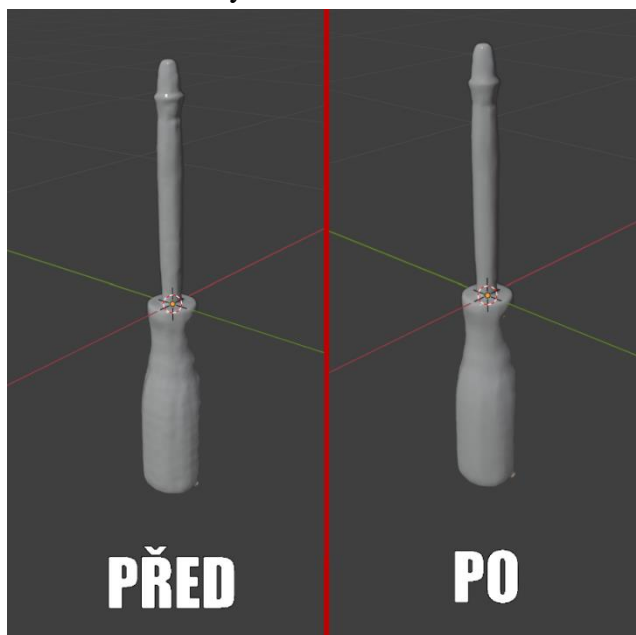
Obr. 25 - Model zdrojového modulu [48]

Dále se pomocí nástroje vygeneroval šroubovák, který bude sloužit jako pomocné nářadí během montážního procesu, konkrétně při upevňování komponent k rámu.

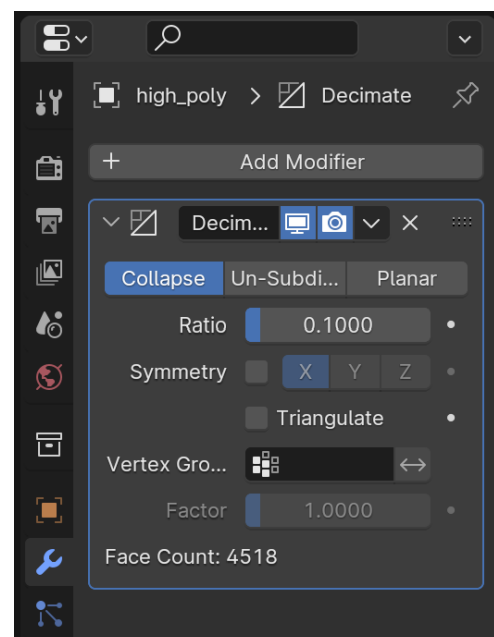


Obr. 26 - Model šroubováku [48]

Optimalizace byla provedena v grafickém softwaru Blender. Nejprve byly vygenerované modely importovány a následně byla použita funkce Decimate, která sníží množství polygonů podle zadaného parametru, kterým je v tomto případě procentuální hodnota původního množství. Tento proces sice sníží kvalitu modelů, ale u této aplikace je tento rozdíl téměř nezaznamatelný.



Obr. 27 - Srovnání kvality modelů u funkce Decimate

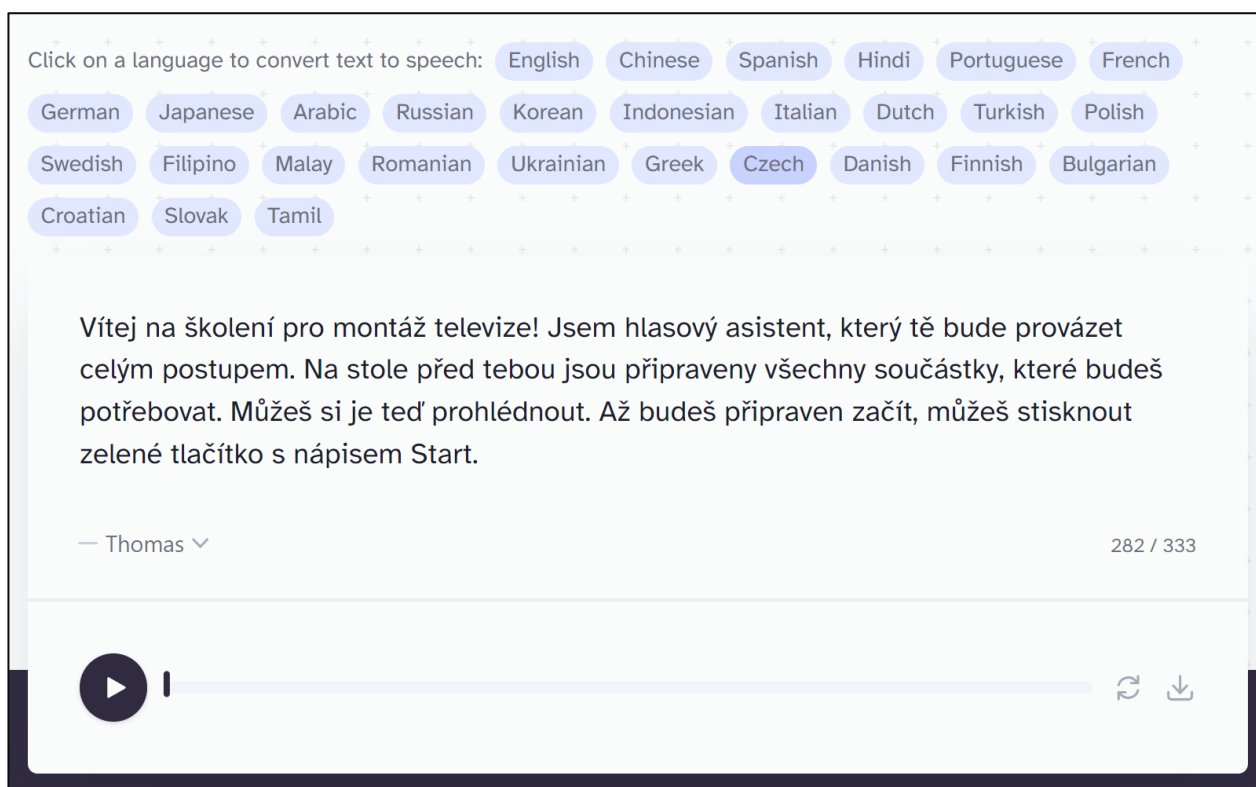


Obr. 28 - Nastavení Decimate v Blenderu

6.3 Tvorba hlasového popisu postupu

Další podstatnou částí je generování všech potřebných audio nahrávek pro hlasové navádění uživatele v průběhu modelované pracovní činnosti. Tohoto bude docíleno využitím nástroje umělé inteligence ElevenLabs [50].

Tento nástroj dokáže z textu vytvořit přirozeně znějící hlasovou nahrávku ve formátu mp3. K dispozici je několik desítek jazyků (mimo jiné angličtina, španělština, němčina nebo francouzština), ale v tomto projektu bude využíván český jazyk. Dále je možnost zvolit jeden z řady druhů mužských i ženských hlasů, které každý znějí odlišně. Pro generování hlasové instruktaže pro tuto aplikaci byl použit hlas označen jako Thomas.



Obr. 29 - Využití nástroje na generaci hlasového záznamu [50]

Podrobný scénář průběhu pracovního postupu (mluvené části vložené do nástroje ElevenLabs jsou označeny kurzívou):

Začátek před pracovním stolem, na kterém jsou připraveny komponenty pro smontování televize.

Vítej na školení pro montáž televize! Jsem hlasový asistent, který tě bude provázet celým postupem. Na stole před tebou jsou připraveny všechny součástky, které budeš potřebovat. Můžeš si je teď prohlédnout. Až budeš připraven začít, můžeš stisknout zelené tlačítko s nápisem „Start“.

Po stisku tlačítka se zvýrazní základní rám televize a základní deska.

Před sebou vidíš zvýrazněný rám televize, do kterého se budou veškeré komponenty montovat. Na pravé straně od rámu je pak základní deska, která se instaluje jako první.

Objeví se ukazatel v místě, kam se má deska umístit.

Ted' můžeš uchopit základní desku a přiložit ji do označeného místa. Pracuj opatrně, deska se může při nesprávném zacházení velmi lehce poškodit!

Po umístění desky do správného prostoru se objeví indikátory v místech, kde má být přišroubována. Zároveň se zvýrazní šroubovák ležící na stole.

Nyní se deska musí připevnit. Vezmi šroubek, dej ho do jednoho z vyznačených míst a následně k němu přilož šroubovák.

Po zkompletování všech čtyř šroubů se zvýrazní zdrojový modul a objeví se ukazatel, kam má být umístěn. Po jeho umístění do této zóny se obdobně jako v minulém kroku vyznačí místa, kde se má přišroubovat a zvýrazní se šroubovák.

Skvěle! Ted' je na řadě připevnění zdrojového modulu. Postup je velmi podobný jako u základní desky. Modul uchop, vlož ho do vyznačené polohy a následně jej přišroubuj v označených místech.

Následně se označí pozice na základní desce a zdrojovém modulu, ve kterých mají být vzájemně propojeny pomocí kabelů.

Ted' je nutné základní desku a zdrojový modul propojit. Před sebou vidíš označená místa, ke kterým musí být připojeny kabely. K jednomu z nich přilož ruku, stiskni a přetáhni ruku až k druhému vyznačenému místu na druhé součásti.

Zvýrazní se dva shodné reproduktory, po uchopení jednoho z nich se objeví označení míst, do kterých patří. Po umístění obou do správných pozic se vyznačí šroubovák a místa, kde se mají přišroubovat.

V tomto kroku musíš přimontovat reproduktory. Jeden z nich uchop, umísti ho do jedné z označených pozic a následně uchop druhý a umísti ho do té zbývající. Poté budeš muset reproduktory připevnit dvěma šroubky na každém z nich.

Zvýrazní se zadní část televize ležící na stole. Po jejích nasazení na rám se označí místa v rozích, ve kterých se musí přišroubovat.

Posledním krokem je připevnění zadní krycí části televize. Tu zvedni a opatrně ji přilož k rámu s přimontovanými součástkami. Poté ji v označených místech přišroubuj.

Po dokončení všech kroků se přehraje zvuk vyznačující dokončení úkolu.

Dobrá práce! To je v rámci tohoto školení vše.

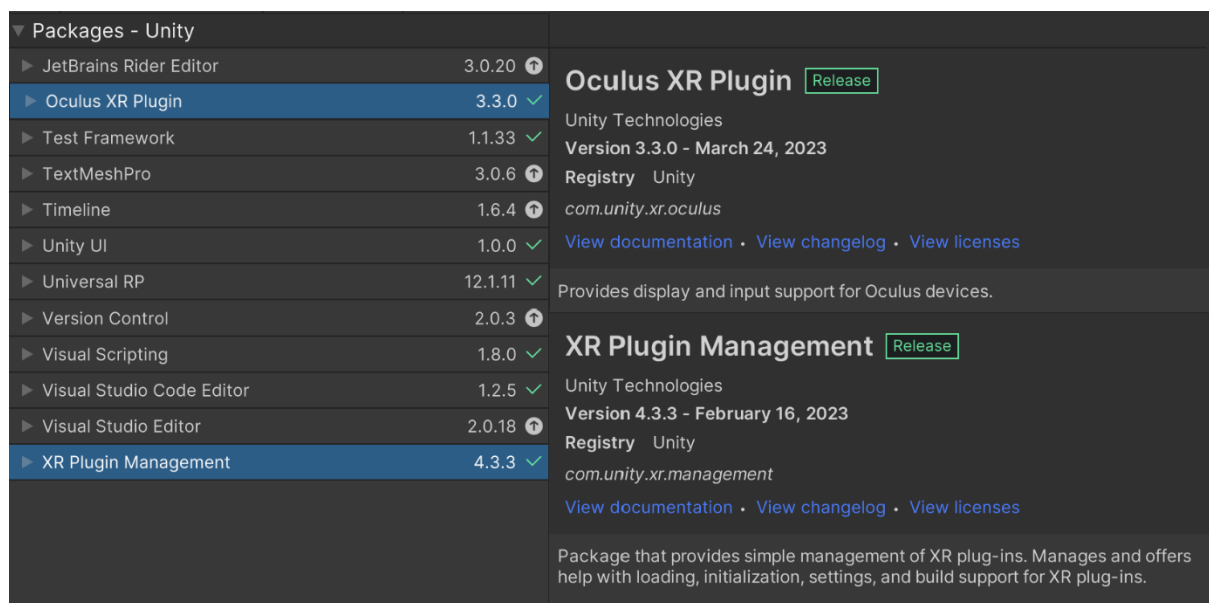
Objeví se koncová grafika obsahující celkovou dobu trvání školení a tlačítko „Restart“, které celý scénář přehraje od začátku.

Konec scénáře.

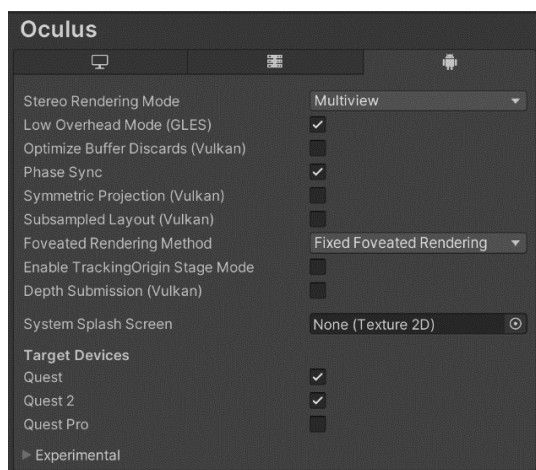
6.4 Projekt v Unity3D

Po vytvoření a připravení veškerých potřebných objektů, textur, zvukových souborů atd. je na čase vytvoření a nastavení scény v programu Unity3D. Projekt je vytvořen na bázi Universal Render Pipeline, který umožňuje rychlejší a více optimalizovaný grafický průběh. Zároveň jsou dalšími jeho výhodami jednoduchost užívání, dobrá možnost skriptování a také velmi dobrá podpora pro řadu platforem, ať už to jsou Windows, Mac, Linux, Android, iOS, či další.

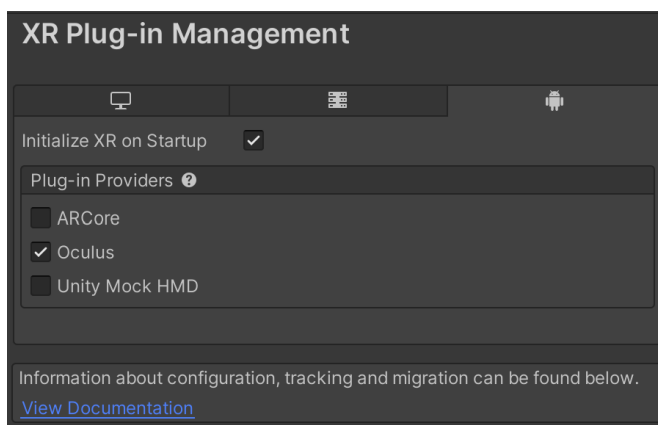
Pro umožnění funkčnosti aplikace u zařízení pro virtuální realitu je nutné nainstalování několika pluginů a balíčků. Konkrétně se jedná o Oculus Integration a následně Oculus XR Plugin a XR Plugin Management. Poté je nutné správné nastavení parametrů těchto pluginů.



Obr. 30 - Potřebné pluginy pro VR funkcionalitu



Obr. 31 - Nastavení Oculus Pluginu



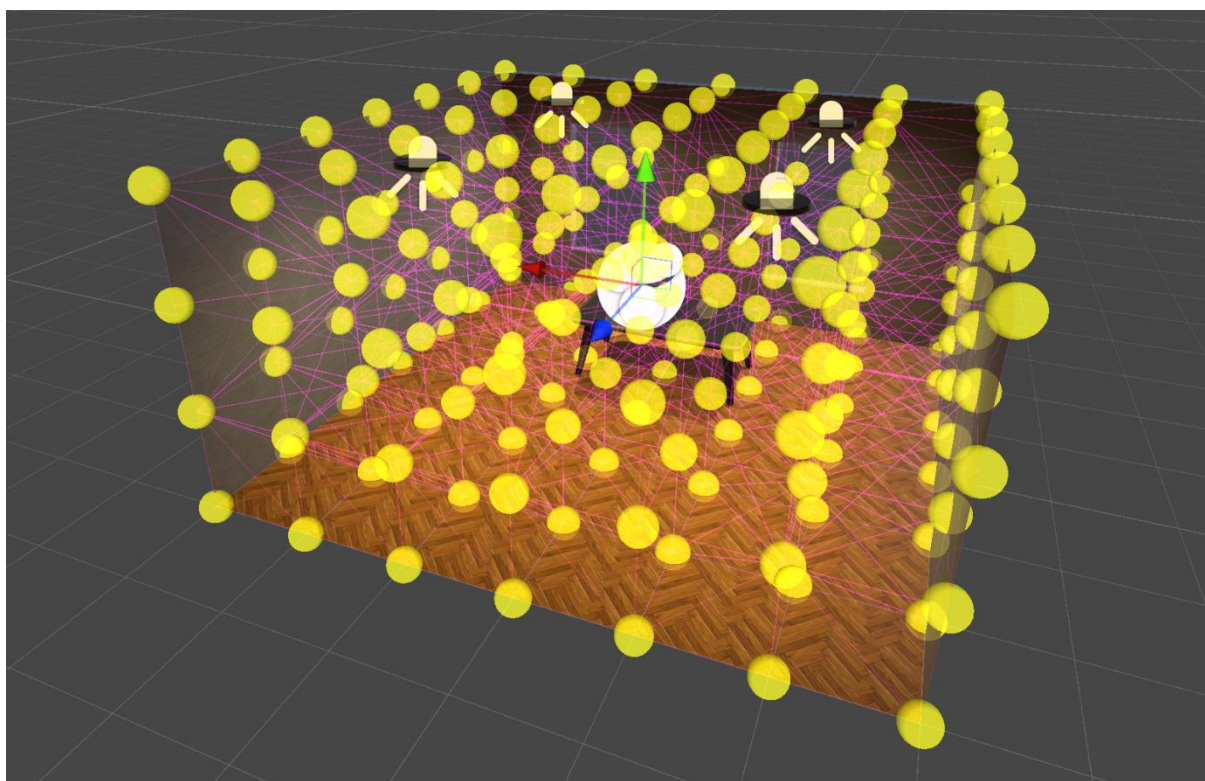
Obr. 32 - Nastavení XR Pluginu

Nedílnou součástí virtuální scény je pak její osvětlení. V prostředí této aplikace se nachází čtyři modely lamp jako součást stropního osvětlení. Do míst, ve kterých se tyto modely nachází, byly poté přidány zdroje světla ve formě Spotlight, u kterých byla poté zvolena odpovídající intenzita a barva světla.

V následujícím kroku byly vytvořeny tzv. Reflection Probes, které do prostředí dodávají odrazy světla a zvyšují tak jeho kvalitu. Jelikož se v tomto projektu jedná o relativně malou místnost pravidelného tvaru, byla umístěna pouze jedna Reflection Probe a to do středu místnosti, zatímco velikost jí ovlivňovanou oblasti byla nastavena tak, aby byla o něco málo větší než místnost samotná, a celou jí tak pokryla.

Posledním potřebným prvkem je tzv. Light Probe Group, která se skládá z několika bodů zkoumajících změnu světla na stín a tím dodávají světlu realistickou podobu. Ty byly rozmístěny po celém objemu místnosti v pravidelných rozestupech.

Na následujícím obrázku je vidět rozmístění těchto popsaných prvků (žluté koule jsou Light Probe Group, bílá koule ve středu je Reflection Probe a čtyři ikony žárovek jsou Spotlights)



Obr. 33 - Rozmístění prvků osvětlení

Pokud by se osvětlení ponechalo takto nastaveno, aplikace by sice fungovala, avšak světla a stíny od světla by se musely neustále dynamicky generovat a vypočítávat, což by zabíralo značnou část výkonu zařízení, na kterém by aplikace běžela. K odstranění tohoto problému se provede tzv. Baking, neboli zapékání světla. Tento proces spočívá v předběžném vypočítání interakce světla s prostředím a následná integrace výsledku do aplikace. Tím se docílí značně plynulejšího chodu aplikace.

Nejprve je nutné objekty nacházející se ve scéně a tedy ty, které se během činnosti nebudou pohybovat, nastavit všechny jako statické. To zahrnuje stůl, stěny, podlahu, strop a světla. Po provedení tohoto kroku je možné přejít do samotného nastavení Baking procesu a jeho následné uskutečnění.

Po zapečení světla mohou být do prostředí vloženy všechny dynamické objekty, tedy ty, se kterými bude uživatel interagovat. Do této kategorie spadají samotné komponenty televizní sestavy, tedy rám, základní deska, zdrojový modul a reproduktory. Všechny tyto předměty se importují do projektu, zasadí do scény a přiřadí se jim jejich odpovídající struktura. Také je důležité správně nastavit jejich rozměry, jelikož modely vygenerované pomocí AI často nemají správnou velikost odpovídající danému předmětu.

Všechny tyto objekty se rozmístí na jejich počáteční pozice, ve kterých se mají při spuštění aplikace nacházet. Toto je znázorněno na následujícím obrázku:



Obr. 34 - Pracovní plocha s umístěnými komponenty

Testování aplikace je možné dvěma způsoby. Prvním z nich je přímo v otevřeném softwaru Unity s pomocí připojení brýlí k počítači, na kterém je projekt zpracováván. Druhý způsob je pak vytvoření souboru (tzv. Building), který je nainstalován do brýlí a umožní jim aplikaci přehrát.

Pro obě tyto alternativy je nezbytné správné nakonfigurování parametrů v nastavení platformy, tedy platformy Windows pro první možnost a platformy Android pro možnost druhou. Pro platformu Android bylo zvoleno nastavení ASTC (Adaptive Scalable Texture Compression), které snižuje hardwarovou zátěž během chodu aplikace.

6.5 Skripty pro funkcionalitu aplikace

Fungování aplikace, ať už se jedná o interakci uživatele s prostředím, či přehrávání hlasové instruktaže v určitém předem stanovém sledu, je zprostředkováno za pomoci skriptů. Tyto části kódu, psané v určitém programovacím jazyce, jsou vloženy do projektu v softwaru Unity a spuštěny během chodu aplikace, jak je potřeba.

Proces tvorby některých skriptů byl založen na sestrojení detailního popisu funkcionality, kterou má skript aplikaci dodávat. Do popisu byla také přidána specifická pojmenování proměnných a metod, aby bylo docíleno určitého sjednocení pojmenování. Pokud by tomu tak nebylo, tak by mohl použitý AI model volit odlišná pojmenování a byla by tak zkomplikována možnost vzájemné provázanosti skriptů.

Následujícím krokem bylo vložení tohoto popisu do nástroje ChatGPT [51] s odpovídajícím požadavkem. Tady je ukázka tohoto pracovního procesu u skriptu s názvem **CameraRecenter**, který zaručuje, že po nastartování aplikace se kamera přesune do správné výchozí pozice, a také umožňuje kameru do této pozice přenést manuálně.

Napiš skript v jazyce C#, který odpovídá následujícímu popisu:

Tento skript je určen pro recenterování kamery ve virtuální realitě (VR) v Unity, který také umožňuje nastavit různé parametry pro recenterování a časování. Zde je podrobný popis jednotlivých částí a jejich funkcionality:

→ **Proměnné:**

- *m_CameraRig*: Transformace kamery v rámci scény.
- *m_CentreEyeAnchor*: Transformace pro umístění kamery
- *centralniKamera*: Reference na centrální kameru ve scéně.
- *cameraDefaultYValue*: Výchozí výška kamery.
- *cameraY*: Aktuální výška kamery.
- *m_OVRCameraRig*: Reference na OVRCameraRig, který obsahuje komponenty kamery pro Oculus VR.
- *degree*: Úhel natočení kamery po recenterování.
- *resetPositionTime*: Časový interval pro zpoždění před recenterováním kamery.
- *playOnAwake*: Určuje, zda má být funkce spuštěna automaticky při startu.

→ **Metoda Awake()**: Nastavuje počáteční hodnoty a hledá referenci na centrální kameru. Pokud je nastavena možnost *playOnAwake*, spouští funkci *ResetVRPosition()* s nastaveným zpožděním a také spouští rutinu *Cekej()*.

→ **Metoda ResetVRPosition()**: Realignuje kameru na výchozí pozici a úhel rotace. Natočí kameru podle zadaného úhlu. Přesune kameru na novou pozici na základě pozice spawn bodu. Upraví výšku kamery.

→ **Metoda recenterDo()**: Spouští rutinu *recenterDoAgain()*.

→ **Metoda recenterDoAgain()**: Spouští animaci při recenterování. Čeká určitý časový interval. Volá metodu *ResetVRPosition()* pro recenterování kamery. Čeká další časový interval. Opětovně spouští animaci.

→ **Metoda ResetVRPositionNoRot()**: Provádí stejné operace jako *ResetVRPosition()*, ale neprovádí žádné otáčení kamery.

→ **Metoda RecenterDoNoRot()**: Spouští rutinu *recenterDoAgainNoRot()*.

→ **Metoda recenterDoAgainNoRot()**: Provádí stejné operace jako *recenterDoAgain()*, ale používá metodu *ResetVRPositionNoRot()*.

→ **Metoda Cekej()**: Čeká určitý časový interval a poté spouští animaci.

Po provedení požadavku je výstupem skript v jazyce C#, který vypadá následovně:

```
1. using System.Collections;
2. using UnityEngine;
3.
4. public class CameraRecenter : MonoBehaviour
5. {
6.     // Proměnné
7.     private Transform m_CameraRig;
8.     private Transform m_CentreEyeAnchor;
9.     private GameObject centralniKamera;
10.    private float cameraDefaultYValue;
11.    private float cameraY = 0f;
12.
13.    public OVRCameraRig m_OVRCameraRig;
14.    public int degree;
15.    public float resetPositionTime = 0.1f;
16.    public bool playOnAwake;
17.
18.    // Metoda Awake
19.    void Awake()
20.    {
21.        centralniKamera = GameObject.Find("CenterEyeAnchor");
22.        m_CentreEyeAnchor = centralniKamera.transform;
23.        m_CameraRig = m_OVRCameraRig.transform;
24.        cameraDefaultYValue = m_CameraRig.position.y;
25.
26.        if (playOnAwake)
27.        {
28.            Invoke("ResetVRPosition", resetPositionTime);
29.            StartCoroutine(Wait());
30.        }
31.    }
32.
33.    // Metoda ResetVRPosition
34.    public void ResetVRPosition()
35.    {
36.        float currentRotY = m_CentreEyeAnchor.eulerAngles.y;
37.        float targetYRotation = 0.0f;
38.        float difference = targetYRotation - currentRotY + degree;
39.        m_CameraRig.Rotate(0, difference, 0);
40.
41.        Vector3 newPos = new Vector3(transform.position.x - m_CentreEyeAnchor.position.x, 0,
transform.position.z - m_CentreEyeAnchor.position.z);
42.        m_CameraRig.transform.position += newPos;
43.
44.        m_CameraRig.transform.parent = transform;
45.        cameraY = cameraDefaultYValue - m_CentreEyeAnchor.localPosition.y;
46.        m_CameraRig.localPosition = new Vector3(m_CameraRig.localPosition.x, cameraY,
m_CameraRig.localPosition.z);
47.    }
48.
49.    // Metoda recenterDo
50.    public void RecenterDo()
51.    {
52.        StartCoroutine(RecenterDoAgain());
53.    }
54.
55.    // Metoda recenterDoAgain
56.    IEnumerator RecenterDoAgain()
57.    {
58.        // Spustit animaci při recenterování
59.        yield return new WaitForSeconds(1.5f);
60.        ResetVRPosition();
61.        yield return new WaitForSeconds(0.1f);
62.    }
63.
```

```
64. // Metoda ResetVRPositionNoRot
65. public void ResetVRPositionNoRot()
66. {
67.     Vector3 newPos = new Vector3(transform.position.x - m_CentreEyeAnchor.position.x, 0,
transform.position.z - m_CentreEyeAnchor.position.z);
68.     m_CameraRig.transform.position += newPos;
69.
70.     m_CameraRig.transform.parent = transform;
71.     cameraY = cameraDefaultYValue - m_CentreEyeAnchor.localPosition.y;
72.     m_CameraRig.localPosition = new Vector3(m_CameraRig.localPosition.x, cameraY,
m_CameraRig.localPosition.z);
73. }
74.
75. // Metoda RecenterDoNoRot
76. public void RecenterDoNoRot()
77. {
78.     StartCoroutine(RecenterDoAgainNoRot());
79. }
80.
81. // Metoda RecenterDoAgainNoRot
82. IEnumerator RecenterDoAgainNoRot()
83. {
84.     // Spustit animaci při recenterování
85.     yield return new WaitForSeconds(1.5f);
86.     ResetVRPositionNoRot();
87.     yield return new WaitForSeconds(0.1f);
88. }
89.
90. // Metoda Cekej
91. IEnumerator Wait()
92. {
93.     yield return new WaitForSeconds(0.3f);
94.     // Spustit animaci
95. }
96. }
```

Dalším velmi důležitým skriptem je ten s názvem **GrabbableObject**. Tento skript přináší komplexní možnosti interakce s objekty ve virtuální realitě a umožňuje detailní ovládání chycení a puštění objektů pomocí rukou hráče. Pracuje s rukama hráče ve virtuální realitě pomocí technologie hand-tracking.

→ Proměnné

- *canGrabLeft* / *canGrabRight*: Tyto proměnné určují, zda může být objekt chycen levou nebo pravou rukou.
- *setupMode*: Určuje, zda je tento objekt přidělen k levé ruce, pravé ruce nebo žádné ruce.
- *rightHand* / *leftHand*: Reference na herní objekty reprezentující pravou a levou ruku.
- *rightHandTracking* / *leftHandTracking*: Reference na skripty pro sledování pohybu pravé a levé ruky.
- *gameDirector*: Reference na skript řídící herní události.
- *kinem*: Reference na Rigidbody objektu.
- *isSnapped*, *snapZone*, *wasSnapped*, *canSnapToSnapzone*: Proměnné pro sledování stavu snapování objektu k ruce nebo snapování do určité zóny.
- *GrabDistance*: Maximální vzdálenost, ve které může být objekt chycen.
- *Pivot*: Referenční bod pro rotaci a pozici objektu.
- Další proměnné pro nastavení chování při chycení a puštění objektu.
- Proměnné pro identifikaci prstů: Tyto proměnné obsahují názvy prstů v hierarchii ruky, což je důležité pro detekci jejich polohy.

- **Metoda Start():** Inicializuje proměnné a získává reference na potřebné herní objekty.
- **Metoda Update():** Aktualizuje stav objektu ve hře.
- **Metody vypnoutPrsty() a zapnoutPrsty():** Zapíná a vypíná kolizní detekci prstů.
- **Metoda setProperties():** Nastavuje vlastnosti objektu na základě interakce s rukama.
- **Metody Grab(), instantGrab(), GrabNone():** Metody pro chycení a puštění objektu.
- **Metody OnTriggerStay() a OnTriggerExit():** Detekuje, zda jsou prsty v dosahu.
- **Metody OnDestroy():** Zajišťuje, že po zničení objektu se ruka uvolní.

U následujících skriptů je z důvodu předejití zbytečné repetice uveden pouze popis jejich funkcionality.

Skript **SnapZone** implementuje funkčnost "snap drop zóny" v herním prostředí Unity. Jedná se o specifikovaný prostor, který může mít libovolně zvolený tvar, který ukazuje uživateli pozici, do které má být jím držený objekt umístěn. Jeho hlavním účelem je tak umožnit hráčům snadnou a intuitivní manipulaci s herními objekty, které mají být připevněny k určenému místu v prostoru hry.

Pomocí skriptu **GameScriptManager** je možné spravovat postup a řízení různých prvků v prostředí aplikace v závislosti na postupu uživatele, což obsahuje například zobrazování a odstraňování objektů, jak je potřeba, nebo spouštění předem určených události při splnění daných požadavků. Zároveň je zde také funkcionality sledování časů jednotlivých kroků a celkového času trvání celého procesu.

7 Zhodnocení a porovnání pracovního postupu

V jejich současném stavu jsou nástroje umělé inteligence schopné výrazně zlepšit efektivitu některých činností v rámci vývoje aplikací. Asi největším využitím je několikanásobné zrychlení jednoduchých a často se opakujících činností, které však často u vývojářů tvoří poměrně velkou část jejich pracovní náplně. Využitím specifických nástrojů tak může tuto časovou zátěž skoro až zcela eliminovat a více se soustředit na komplexnější činnosti, jako například širší strategii během vývoje.

Na počátku vývoje aplikace stojí fáze navrhování a prototypování. S využitím umělé inteligence může vývojář velmi efektivně a rychle navrhnout několik možností a vytvořit pro ně zkušební verze, čímž je značně zefektivněn proces výběru té nejlepší varianty. Běžně by se pouze navrhlo několik verzí, ale nebylo by dostatek času nebo prostředků je všechny testovat a tak by bylo rozhodováno na základě určitých teoretických předpokladů.

Další využití umělé inteligence je u vytvoření určité počáteční verze během fáze vývoje a to primárně v oblasti programování. Uživatel je schopný nástroji definovat funkcionalitu požadovaného kódu, jež je následně velmi rychle vygenerován. Zatímco u standardního postupu by vývojář musel trávit čas psaním kódu od samého počátku, může použít nástroj umělé inteligence a následně rovnou přejít k upravování a zdokonalování programovacího kódu. Do této části patří také využití nástrojů pro generování textur, 3D modelů nebo hlasových nahrávek. S jejich pomocí je vývojář schopen rychle vytvořit prostředí pro testování funkcionality s dostačující mírou kvality. Zároveň je tímto ušetřeno velké množství času a prostředků, které by musely být vynaloženy na jejich nákladnou tvorbu, i přestože eventuálně ve finálním produktu ani nemusí být ani využity.

V samotné části upravování kódu pak vývojář využívá pomocný nástroj zabudovaný do programovacího softwaru, který je schopen mu v reálném čase navrhnout úpravy, či ho hned upozornit na případné nedokonalosti v kódu. Tímto se výrazně zkrátí čas strávený ve fázi hledání a opravování případných chyb, tzv. debugging. Vývojář se tak může více soustředit na komplexní funkcionalitu celého programu.

Častou náplní práce softwarového vývojáře je také zjednodušování kódu a tvorba dokumentace. Pokud je tato činnost prováděna manuálně, může trvat velmi dlouhou dobu. Nástroj umělé inteligence dokáže velmi rychle analyzovat celý kód a některého jeho části zkondenzovat a zjednodušit se současným zachováním funkcionality. Také dokáže převést kód do určité standardizované podoby a docílí tak unifikace struktury napříč různými aplikacemi, čímž se zjednoduší případné budoucí úpravy. V závěrečné fázi dokáže vytvořit dokumentaci a popis kódu pro lepší orientaci v něm.

Závěr

Výsledky této práce ukazují, že umělá inteligence má velký potenciál zefektivnit a zjednodušit proces vývoje aplikací pro virtuální realitu. Při generování kódu pomocí AI došlo k výraznému zkrácení doby potřebné pro napsání základních funkcionalit aplikace, což vývojářům umožňuje věnovat více času na složitější a kreativnější aspekty projektu. AI nástroje pro generování textur a 3D modelů také prokázaly svou užitečnost, především při rychlém vytváření prvků bez nutnosti hlubokých znalostí grafického designu.

Porovnáním jednotlivých přístupů a technik bylo zjištěno, že největší přínos AI spočívá v automatizaci běžných a časově náročných úkolů. Tímto způsobem mohou vývojáři dosáhnout vyšší efektivity a kvality výsledného produktu.

Na základě získaných poznatků lze konstatovat, že integrace umělé inteligence do procesu vývoje softwarových aplikací je nejen možná, ale také velmi přínosná. Při dodržení správných postupů a využití vhodných nástrojů může AI výrazně přispět ke zlepšení celkové kvality a efektivity vývoje projektů ve virtuální realitě. Tento přístup může zkrátit dobu vývoje, snížit náklady a zároveň zvýšit úroveň detailnosti aplikací.

Tento projekt ukazuje, že AI může být účinným nástrojem pro vývoj aplikací, které vyžadují vysokou úroveň detailů a interakce. Výsledná aplikace demonstruje, jak může AI přispět k tvorbě výukových a tréninkových simulací, které jsou efektivní a uživatelsky přívětivé. V budoucnu se očekává další rozvoj těchto technologií, což otevře nové možnosti a aplikace pro široké spektrum uživatelů a vývojářů. Umělá inteligence se pravděpodobně stane standardní součástí vývojářských nástrojů a procesů. Díky neustálému pokroku v oblasti strojového učení a zpracování přirozeného jazyka budou AI systémy schopny lépe porozumět potřebám uživatelů a přizpůsobit aplikace jejich specifickým požadavkům. Tento trend povede ke zvýšení personalizace a přizpůsobivosti aplikací, což dále rozšíří jejich využitelnost v oblasti softwarového vývoje.

Závěrem lze říci, že využití umělé inteligence v procesu vývoje softwarových aplikací pro virtuální realitu představuje významný krok kupředu. Tato technologie nejenže zvyšuje efektivitu a kvalitu vývoje, ale také otevírá dveře k novým a inovativním způsobům, jak mohou být virtuální aplikace využity.

Seznam použitých zdrojů

- [1] LUKAC, Dusko. The fourth ICT-based industrial revolution „Industry 4.0“ — HMI and the case of CAE/CAD innovation with EPLAN P8. In: *2015 23rd Telecommunications Forum Telfor (TELFOR)* [online]. B.m.: IEEE, 2015, s. 835–838. ISBN 978-1-5090-0055-5. Dostupné z: doi:10.1109/TELFOR.2015.7377595
- [2] BAI, Chunguang, Patrick DALLASEGA, Guido ORZES a Joseph SARKIS. Industry 4.0 technologies assessment: A sustainability perspective. *International Journal of Production Economics* [online]. 2020, **229**, 107776. ISSN 0925-5273. Dostupné z: doi: <https://doi.org/10.1016/j.ijpe.2020.107776>
- [3] ROBLEK, Vasja, Maja MEŠKO a Alojz KRAPEŽ. A Complex View of Industry 4.0. *SAGE Open* [online]. 2016, **6**(2), 2158244016653987. Dostupné z: doi:10.1177/2158244016653987
- [4] SCHWAB, Klaus. The Fourth Industrial Revolution. *Encyclopedia Britannica* [online]. 2023. Dostupné z: <https://www.britannica.com/topic/The-Fourth-Industrial-Revolution-2119734>
- [5] COOPER, Joshua a Anne JAMES. Challenges for Database Management in the Internet of Things. *IETE Technical Review* [online]. 2009, **26**(5), 320–329. Dostupné z: doi:10.4103/0256-4602.55275
- [6] ROSE, Karen, Scott ELDRIDGE a Lyman CHAPIN. The internet of things: An overview. *The internet society (ISOC)*. 2015, **80**, 1–50.
- [7] REIS, Jacqueline Zonichenn a Rodrigo Franco GONÇALVES. The Role of Internet of Services (IoS) on Industry 4.0 Through the Service Oriented Architecture (SOA). In: [online]. 2018, s. 20–26. Dostupné z: doi:10.1007/978-3-319-99707-0_3
- [8] RUSSELL, Stuart J a Peter NORVIG. *Artificial intelligence a modern approach*. B.m.: London, 2010.
- [9] TURING, A M. Computing machinery and intelligence. *Mind* [online]. 1950, **59**, 433–460. ISSN 1460-2113(Electronic),0026-4423(Print). Dostupné z: doi:10.1093/mind/LIX.236.433
- [10] NEWELL, Allen a Herbert Alexander SIMON. GPS, a program that simulates human thought. 1961.
- [11] FRENCH, Robert M. The Turing Test: the first 50 years. *Trends in cognitive sciences*. 2000, **4**(3), 115–122.
- [12] CHUNG, Hyunji a Sangjin LEE. Intelligent Virtual Assistant knows Your Life. 2018.
- [13] CHATTERJEE, Gairik. Analysis and impacts of narrow artificial intelligence in global marketing [online]. 2020 [vid. 2023-11-15]. Dostupné z: <http://dspace.dtu.ac.in:8080/jspui/handle/repository/17975>
- [14] SHEVLIN, Henry, Karina VOLD, Matthew CROSBY a Marta HALINA. The limits of machine intelligence: Despite progress in machine intelligence, artificial general intelligence is still a major challenge. *EMBO reports* [online]. 2019, **20**(10), e49177. ISSN 1469-3178. Dostupné z: doi:10.15252/embr.201949177

- [15] BOSTROM, Nick. *Superintelligence: Paths, strategies, dangers*. B.m.: Oxford: Oxford University Press. 2014
- [16] JACKSON, Peter. *Introduction to expert systems*. 1986.
- [17] HINTZE, Arend. Understanding the four types of AI, from reactive robots to self-aware beings. *The Conversation* [online]. 2016 [vid. 2023-11-15]. Dostupné z: <https://theconversation.com/understanding-the-four-types-of-ai-from-reactive-robots-to-self-aware-beings-67616>
- [18] MEYER, Gereon a Sven BEIKER, ed. *Road Vehicle Automation* [online]. Cham: Springer International Publishing, 2014. ISBN 978-3-319-05989-1. Dostupné z: doi:10.1007/978-3-319-05990-7
- [19] WILLIAMS, Jessica, Stephen M FIORE a Florian JENTSCH. Supporting Artificial Social Intelligence With Theory of Mind. *Frontiers in Artificial Intelligence* [online]. 2022, 5. ISSN 2624-8212. Dostupné z: <https://www.frontiersin.org/articles/10.3389/frai.2022.750763>
- [20] KOUNEV, Samuel, Peter LEWIS, Kirstie L BELLMAN, Nelly BENCOMO, Javier CAMARA, Ada DIACONESCU, Lukas ESTERLE, Kurt GEIHS, Holger GIESE, Sebastian GÖTZ, Paola INVERARDI, Jeffrey O KEPHART a Andrea ZISMAN. The Notion of Self-aware Computing. In: Samuel KOUNEV, Jeffrey O KEPHART, Aleksandar MILENKOSKI a Xiaoyun ZHU, ed. *Self-Aware Computing Systems* [online]. Cham: Springer International Publishing, 2017, s. 3–16. ISBN 978-3-319-47474-8. Dostupné z: doi:10.1007/978-3-319-47474-8_1
- [21] LAVALLE, Steven M. *Virtual Reality* [online]. B.m.: Cambridge University Press, 2023. ISBN 9781108182874. Dostupné z: doi:10.1017/9781108182874
- [22] SHIBATA, Takashi. Head mounted display. *Displays* [online]. 2002, 23(1–2), 57–64. ISSN 01419382. Dostupné z: doi:10.1016/S0141-9382(02)00010-0
- [23] *iFixit* [online]. [vid. 2023-11-16]. Dostupné z: https://www.ifixit.com/Device/Valve_Index_Headset
- [24] CRUZ-NEIRA, Carolina, Daniel J. SANDIN, Thomas A. DEFANTI, Robert V. KENYON a John C. HART. The CAVE: audio visual experience automatic virtual environment. *Communications of the ACM* [online]. 1992, 35(6), 64–72. ISSN 0001-0782. Dostupné z: doi:10.1145/129888.129892
- [25] GOURLAY, Michael J. a Robert T. HELD. Head-Mounted-Display Tracking for Augmented and Virtual Reality. *Information Display* [online]. 2017, 33(1), 6–10. ISSN 0362-0972. Dostupné z: doi:10.1002/j.2637-496X.2017.tb00962.x
- [26] *Accuracy Investigation of the Pose Determination of a VR System – Scientific Figure on ResearchGate* [online]. [vid. 2023-11-17]. Dostupné z: https://www.researchgate.net/figure/Different-tracking-procedures-a-b-and-system-components-c_fig1_349628153
- [27] ANGELOV, Vladislav, Emiliyan PETKOV, Georgi SHIPKOVENSKI a Teodor KALUSHKOV. Modern Virtual Reality Headsets. In: *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* [online]. B.m.: IEEE, 2020, s. 1–5. ISBN 978-1-7281-9352-6. Dostupné z: doi:10.1109/HORA49412.2020.9152604

- [28] ROSENBERG, Louis B. *The Use of Virtual Fixtures as Perceptual Overlays to Enhance Operator Performance in Remote Environments*. [online]. 1992 [vid. 2023-11-19]. Dostupné z: <https://apps.dtic.mil/docs/citations/ADA292450>
- [29] CARMIGNIANI, Julie, Borko FURHT, Marco ANISETTI, Paolo CERAVOLO, Ernesto DAMIANI a Misa IVKOVIC. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications* [online]. 2011, **51**(1), 341–377. ISSN 1380-7501. Dostupné z: doi:10.1007/s11042-010-0660-6
- [30] BIMBER, Oliver a Ramesh RASKAR. *Spatial augmented reality: merging real and virtual worlds*. B.m.: CRC press, 2005.
- [31] KARPATY, Andrej, Pieter ABBEEL, Greg BROCKMAN, Peter CHEN, Vicki CHEUNG, Yan DUAN, Ian GOODFELLOW, Durk KINGMA, Jonathan HO, Rein HOUTHOOFT, Tim SALIMANS, John SCHULMAN, Ilya SUTSKEVER a Wojciech ZAREMBA. Generative models. *OpenAI* [online]. 16. červen 2016 [vid. 2023-11-26]. Dostupné z: <https://openai.com/research/generative-models>
- [32] JORDAN, M I a T M MITCHELL. Machine learning: Trends, perspectives, and prospects. *Science* [online]. 2015, **349**(6245), 255–260. Dostupné z: doi:10.1126/science.aaa8415
- [33] SCHMIDHUBER, Jürgen. Deep learning in neural networks: An overview. *Neural Networks* [online]. 2015, **61**, 85–117. ISSN 08936080. Dostupné z: doi:10.1016/j.neunet.2014.09.003
- [34] *Artificial Neural Network Tutorial* [online]. [vid. 2023-11-26]. Dostupné z: <https://www.javatpoint.com/artificial-neural-network>
- [35] HEY, Anthony JG a Gyuri PÁPAY. *The computing universe: a journey through a revolution*. B.m.: Cambridge University Press, 2014.
- [36] GitHub Copilot AI Improved, Offered as API: „A Taste of the Future". *Visual Studio Magazine* [online]. 8. listopad 2021 [vid. 2023-11-25]. Dostupné z: <https://visualstudiomagazine.com/articles/2021/08/11/openai-api.aspx>
- [37] GUI, Jie, Zhenan SUN, Yonggang WEN, Dacheng TAO a Jieping YE. A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *IEEE Transactions on Knowledge and Data Engineering* [online]. 2023, **35**(4), 3313–3332. ISSN 1041-4347. Dostupné z: doi:10.1109/TKDE.2021.3130191
- [38] *Poly* [online]. [vid. 2023-11-26]. Dostupné z: <https://withpoly.com/browse/textures>
- [39] *Texture Lab* [online]. [vid. 2023-11-26]. Dostupné z: <https://www.texturelab.xyz>
- [40] *Polycam* [online]. [vid. 2023-11-26]. Dostupné z: <https://poly.cam/tools/ai-texture-generator>
- [41] *3DFY* [online]. [vid. 2023-11-26]. Dostupné z: <https://3dfy.ai/solutions>
- [42] *Masterpiece X* [online]. [vid. 2023-11-26]. Dostupné z: <https://app.masterpiecex.com>
- [43] *Point-E* [online]. [vid. 2023-11-26]. Dostupné z: <https://openai.com/research/point-e>
- [44] *Stable Diffusion XL* [online]. [vid. 2023-11-27]. Dostupné z: <https://stability.ai/stable-diffusion>
- [45] *DALL-E 3* [online]. [vid. 2023-11-27]. Dostupné z: <https://openai.com/dall-e-3>

- [46] *CopernicAI* [online]. [vid. 2023-11-27]. Dostupné z: <https://copernicai.com>
- [47] *Blockade Labs Skybox* [online]. [vid. 2023-11-27]. Dostupné z: <https://www.blockadelabs.com/#skybox>
- [48] *LumaLabs* [online]. [vid. 2024-04-08]. Dostupné z: <https://lumalabs.ai/genie>
- [49] *SloydAI* [online]. [vid. 2024-04-08]. Dostupné z: <https://www.sloyd.ai>
- [50] *ElevenLabs* [online]. [vid. 2024-04-08]. Dostupné z: <https://elevenlabs.io>
- [51] *OPENAI. ChatGPT* [online]. 2024 [vid. 2024-04-08]. Dostupné z: <https://chat.openai.com>
- [52] *VisBox* [online]. [vid. 2023-11-16]. Dostupné z: <http://www.visbox.com/products/cave/viscube-c4-4ku/>
- [53] CNN. *The Oculus Quest 2, our favorite VR headset, comes with a \$50 gift card for Black Friday* [online]. 24. listopad 2021 [vid. 2023-11-19]. Dostupné z: <https://edition.cnn.com/2021/11/24/cnn-underscored/oculus-quest-2-black-friday-deal-2021>
- [54] *Understand display techniques in augmented reality* [online]. [vid. 2023-11-19]. Dostupné z: <https://niteeshyadav.com/blog/understanding-display-techniques-in-augmented-reality-7485/>
- [55] Trimble's new high-accuracy handheld Augmented Reality system takes data visualization outdoors. *Geospatial World* [online]. 17. září 2019 [vid. 2023-11-25]. Dostupné z: <https://www.geospatialworld.net/news/trimbles-new-high-accuracy-handheld-augmented-reality-system-takes-data-visualization-outdoors/>
- [56] *How to Identify the Parts in an LED (and OLED) TV* [online]. [vid. 2024-04-08]. Dostupné z: <https://blog.shopjimmy.com/how-to-identify-the-parts-in-an-led-and-oled-tv/>
- [57] *Cracking Open the 55" Samsung LED TV* [online]. [vid. 2024-04-08]. Dostupné z: <https://www.techrepublic.com/pictures/cracking-open-the-55-samsung-led-tv-un55d6300sf/?slide-index=11>