



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**PROGRAMOVÁNÍ ROBOTICKÉHO PRACOVISTĚ PO-
MOCÍ MICROSOFT HOLOLENS 2**

ROBOTIC WORKPLACE PROGRAMMING USING MICROSOFT HOLOLENS 2

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. SIMONA HIADLOVSKÁ

VEDOUcí PRÁCE

SUPERVISOR

Ing. DANIEL BAMBUŠEK

BRNO 2022

Zadání diplomové práce



Studentka: **Hiadlovská Simona, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Softwarové inženýrství
Název: **Programování robotického pracoviště pomocí Microsoft HoloLens 2
Robotic Workplace Programming Using Microsoft HoloLens 2**
Kategorie: Uživatelská rozhraní
Zadání:

1. Prostudujte koncept rozšířené reality a její využití v robotice. Seznamte se s brýlemi Microsoft HoloLens 2, fakulní experimentální platformou ARCOR2, jejími možnostmi a aplikačním rozhraním.
2. Vyberte vhodné metody a nástroje a navrhnete uživatelské rozhraní, které umožní pomocí HoloLens skrze platformu ARCOR2 naprogramovat kolaborativního robota.
3. Navrženou aplikaci implementujte.
4. Proveďte experimenty a vyhodnoťte vlastnosti výsledného řešení.
5. Vytvořte plakát nebo video prezentující klíčové vlastnosti výsledného řešení.

Literatura:

- SCHMALSTIEG Dieter, HÖLLERER Tobias. *Augmented Reality: Principles and Practice*. Addison-Wesley, 2016. ISBN 978-0321883575.
- HARTSON Rex. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. 2012. ISBN 9780123852427.
- KAPINUS Michal, BERAN Vítězslav, MATERNA Zdeněk a BAMBUŠEK Daniel. Spatially Situated End-User Robot Programming in Augmented Reality. In: *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. New Delhi: Institute of Electrical and Electronics Engineers, 2019, s. 1-9. ISBN 978-1-7281-2622-7.
- BAMBUŠEK Daniel, MATERNA Zdeněk, KAPINUS Michal, BERAN Vítězslav a SMRŽ Pavel. Combining Interactive Spatial Augmented Reality with Head-Mounted Display for End-User Collaborative Robot Programming. In: *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. New Delhi: Institute of Electrical and Electronics Engineers, 2019, s. 1-9. ISBN 978-1-7281-2622-7.
- Dále dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a rozpracovaný bod 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Bambušek Daniel, Ing.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 18. května 2022

Datum schválení: 1. listopadu 2021

Abstrakt

Táto práca sa zaoberá použitím okuliarov, pre rozšírenú realitu, Microsoft HoloLens 2 pri programovaní robotického pracoviska. Použitie okuliarov je demonštrované vytvorením používateľského rozhrania. Práca nadväzuje na existujúce používateľské rozhranie AREditor pripojené k ARServeru. Rozhranie umožňuje používateľovi pridávať do scény pracoviska 3D objekty robotov a kolíznych objektov a manipulovať s nimi. Následne môže k vytvoreným scénam pridávať konkrétne úlohy, v ktorých môže za pomoci 3D objektov akcií a akčných bodov určovať typ akcie a miesto jej vykonania. Akcie môže spojiť prepojeniami, ktoré určia poradie vykonávania akcií. Všetky funkcie sú k dispozícii v jednoduchom menu, ktoré sa používateľovi zobrazí vždy pri pozretí sa na ruku. Výsledné používateľské rozhranie je otestované pomocou používateľských experimentov, kde si účastníci experimentu v rôznom poradí vyskúšali v jednoduchých úlohách navrhnuté používateľské rozhranie a existujúce rozhranie AREditor.

Abstract

This thesis focuses on usability of mixed reality head-mounted display – Microsoft HoloLens in programming a robotic workplace. Use of the headset is demonstrated by created user interface. The thesis builds on the existing user interface– AREditor connected to ARServer. It allows the user to add and manipulate with 3D objects of robots and collision objects to the workplace scene. Subsequently, users can add specific tasks to the created scenes, in which they can use 3D action objects and action points to determine the type of action and the place of its execution. User can combine actions by links that determine the order in which actions are performed. All functions are available in a simple menu, which is displayed to the user whenever he looks at his hand. The resulting user interface is tested using user experiments, where the participants of the experiment tested the designed user interface and the existing AREditor interface in simple tasks.

Kľúčové slová

rozšírená realita, zmiešaná realita, Microsoft HoloLens 2, robotické pracovisko, AREditor, Unity

Keywords

augmented reality, mixed reality, Microsoft HoloLens 2, robotic workplace, AREditor, Unity

Citácia

HIADLOVSKÁ, Simona. *Programování robotického pracoviště pomocí Microsoft HoloLens 2*. Brno, 2022. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Daniel Bambušek

Programování robotického pracoviště pomocí Microsoft HoloLens 2

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracovala samostatne pod vedením pána Ing. Daniel Bamboušek. Uviedla som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpala.

.....
Simona Hiadlovská
17. mája 2022

Podakovanie

Chcela by som poďakovať vedúcemu mojej práce Ing. Danielovi Bambuškovovi za všetku pomoc a cenné rady, ktoré mi poskytol. Taktiež by som chcela poďakovať za poskytnutie pomoci celému tímu výskumnej skupiny Robo@Fit, ktorý mi pomáhal s prepojením aplikácie so serverom. Rovnako tak chcem poďakovať aj fakulte za požičanie potrebného vybavenia k vývoju a testovaniu aplikácie.

Obsah

1	Úvod	3
2	Rozšírená realita (AR)	4
2.1	Definícia AR	4
2.2	Zmiešaná a virtuálna realita	4
2.3	Zobrazovacie techniky	5
3	Microsoft HoloLens 2	9
3.1	Hardware	9
3.2	Možnosti interakcie	10
3.3	Tvorba aplikácií pre Microsoft HoloLens 2	11
3.4	Prípady použitia	12
4	AREditor	14
4.1	Popis a motivácia	14
4.2	Reprezentácia programu	14
4.3	Interakcia s objektami	17
4.4	Zhrnutie aplikácie	18
5	Užívateľské rozhranie pre robotické pracovisko	19
5.1	Existujúce riešenia	19
5.2	Zhodnotenie a motivácia	20
5.3	Návrh elementov užívateľského rozhrania AREditor pre Microsoft HoloLens 2	21
5.4	Interakcia s robotom v užívateľskom rozhraní AREditor pre Microsoft Holo- Lens 2	22
6	Implementácia	25
6.1	Spojzdenie AREditora	25
6.2	Zobrazenie scény a projektu	28
6.3	Interakcia s objektami	29
6.4	Vizualizácia zoznamov	34
6.5	Prepínanie režimov	35
7	Experimenty	37
7.1	Popis experimentu	37
7.2	Priebeh experimentu	41
7.3	Výsledky experimentu	42
7.4	Nápady do budúca	47

8 Závěr	50
Literatúra	52
A Obsah priloženého pamäťového média	55
B Dotazník User Experience Questionnaire a NASA Task Load Index	56

Kapitola 1

Úvod

Industriálne roboty sú flexibilné stroje, ktoré môžu vykonávať rôzne typy jednoduchých ale aj zložitejších úloh vďaka nainštalovaným senzorm a iných funkcií. Aj vďaka rýchlo narastajúcemu pokroku v oblasti automatizácie sú roboty v súčasnosti čo raz viac inštalované aj do prostredia kde je spolupráca človeka s robotom nevyhnutná. Pracovníci tak zdieľajú pracovisko spolu s kolaboratívnymi robotmi a jednou z jeho úloh je nastavenie robota.

Programovanie priemyselných robotov vo veľkovýrobe, ale aj v malých či stredných podnikoch vyžaduje veľa času. Aj preto sa v oblasti programovania robotického pracoviska začala využívať rozšírená realita. Tá sa už používa pri priemyselných úlohách viac ako 20 rokov a je možné ju použiť napr. na vizualizáciu informácií alebo aj na interakciu.

V súčasnosti sa mnoho výskumníkov a výskumníčok v tejto oblasti venuje využitiu rozšírenej reality pri programovaní robotického pracoviska tvorbou používateľského rozhrania pre ručné displeje. Aj keď táto oblasť ešte stále čelí veľkému záujmu vo výskume, existujú limity, ktoré ručné displeje prinášajú.

Zobrazenie rozšírenej reality v náhlavných displejoch sa stáva čoraz viac dostupným či už cenovo ale aj dizajnom. Aj preto je už dnes vidieť vysoký nárast záujmu vo výskume ich využitia v rôznych oblastiach. Jednou z oblastí je aj programovanie robotického pracoviska.

Táto práca sa práve venuje použitiu rozšírenej reality pri programovaní robotického pracoviska s využitím náhlavného displeja Microsoft HoloLens 2 od spoločnosti Microsoft. V kapitole 2 je opísaná základná definícia rozšírenej reality spolu s jej zaradením v koncepte kontinua zmiešanej reality. Popísané sú aj kategórie zobrazovacích techník a typy displejov v ktorých sa tieto techniky využívajú. Technológií Microsoft HoloLens 2 sa venuje kapitola 3. Okrem technickej špecifikácie sú v tejto kapitole popísané aj základné typy interakcií, ktoré technológia ponúka. V závere kapitoly je predstavených niekoľko typov konkrétnych riešení v rôznych oblastiach. Keďže táto práca nadväzuje na existujúce riešenie rozhrania pre ručný displej, kapitola 4 sa venuje prehľadu tohto riešenia. Kapitola 5 sa venuje ďalším existujúcim riešeniam v oblasti programovania robotického pracoviska a následne predstaveniu návrhu riešenia pre Microsoft HoloLens 2. Implementácia riešenia je popísaná v kapitole 6, kde sú jednotlivé časti rozdelené podľa toho či ide o interakčné prvky alebo vizualizáciu. Výsledné používateľské rozhranie je otestované pomocou používateľských experimentov, ktoré sú popísané a vyhodnotené v predposlednej kapitole 7 a v závere je zhrnutý celkový prínos navrhnutého riešenia.

Kapitola 2

Rozšírená realita (AR)

Kapitola pokrýva základne vlastnosti rozšírenej reality. V prvej sekcii 2.1 je opísaná definícia rozšírenej reality, ktorú ako prvý priniesol Azuma [7] v roku 1997. Okrem rozšírenej reality existuje aj virtuálna a zmiešaná realita. Rozdieli a ich vlastnosti sú popísane v druhej sekcii 2.2. Jeden z hlavných prvkov rozšírenej reality sú zobrazovacie techniky, ktoré sa rozdeľujú do niekoľkých kategórií a sú popísane v sekcii tretej 2.3.

2.1 Definícia AR

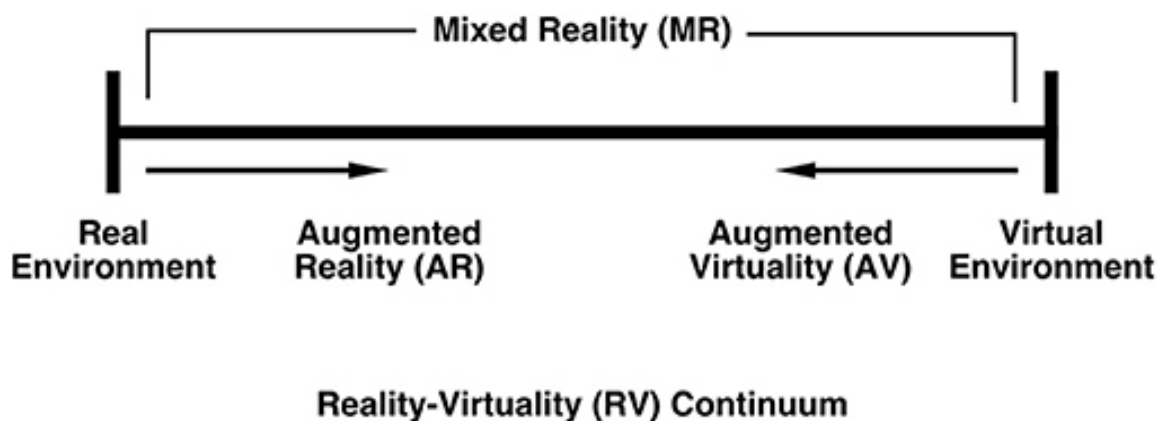
Azuma [7] navrhol prvú definíciu AR identifikáciou troch hlavných prvkov, ktoré ju charakterizujú: kombinácia reálnych a virtuálnych prvkov, interakcia v reálnom čase a reprodukcia obsahu/prvkov v 3D [9, 12].

Tieto tri charakteristiky definujú aj technické požiadavky AR, teda že musí mať displej, ktorý dokáže kombinovať reálne a virtuálne obrazy, počítačový systém, ktorý dokáže generovať interaktívnu grafiku reagujúcu na vstup od užívateľa v reálnom čase, a sledovací systém, ktorý dokáže nájsť pozíciu pohľadu užívateľa a umožní, aby sa virtuálny obraz javil ako statický v reálnom svete.

2.2 Zmiešaná a virtuálna realita

Vo virtuálnej realite má užívateľ na hlave nasadený náhlavný prístroj (displej) a jeho pohľad na skutočný svet je úplne nahradený počítačom generovanou grafikou. V systéme VR je užívateľ úplne oddelený od skutočného sveta a izolovaný iba na pohľad v displeji, takže počítač je pre používateľa opäť neviditeľný [26]. Na rozdiel od toho sú rozhrania AR navrhnuté tak, aby zlepšili interakcie v reálnom svete [11]. Hlavným rozdielom medzi virtuálnou a rozšírenou realitou je ten, že zatiaľ čo cieľom VR je kompletne nahradiť reálny svet tým virtuálnym a poskytnúť tak užívateľovi nové prostredie, cieľom AR je rozšíriť reálny svet digitálnym obsahom.

Koncept zmiešanej reality predstavil Milgram a spol. [20]. Zmiešanú realitu predstavil ako zlúčenie reálneho sveta a virtuálneho sveta. Okrem toho predstavil aj koncept kontinua zmiešanej reality, čo je taxonómia rôznych spôsobov, akým možno kombinovať virtuálne a skutočné prvky (Obrázok 2.1). Na pravej strane je virtuálne prostredie (VE), kde je pohľad používateľa na svet úplne nahradený počítačom generovaným virtuálnym obsahom. Na opačnom ľavom konci je reálne prostredie (RE), kde žiadne zobrazenie užívateľa nie je nahradené virtuálnym obsahom. Približovaním sa k pravej strane (VE) je rozšírená virtu-



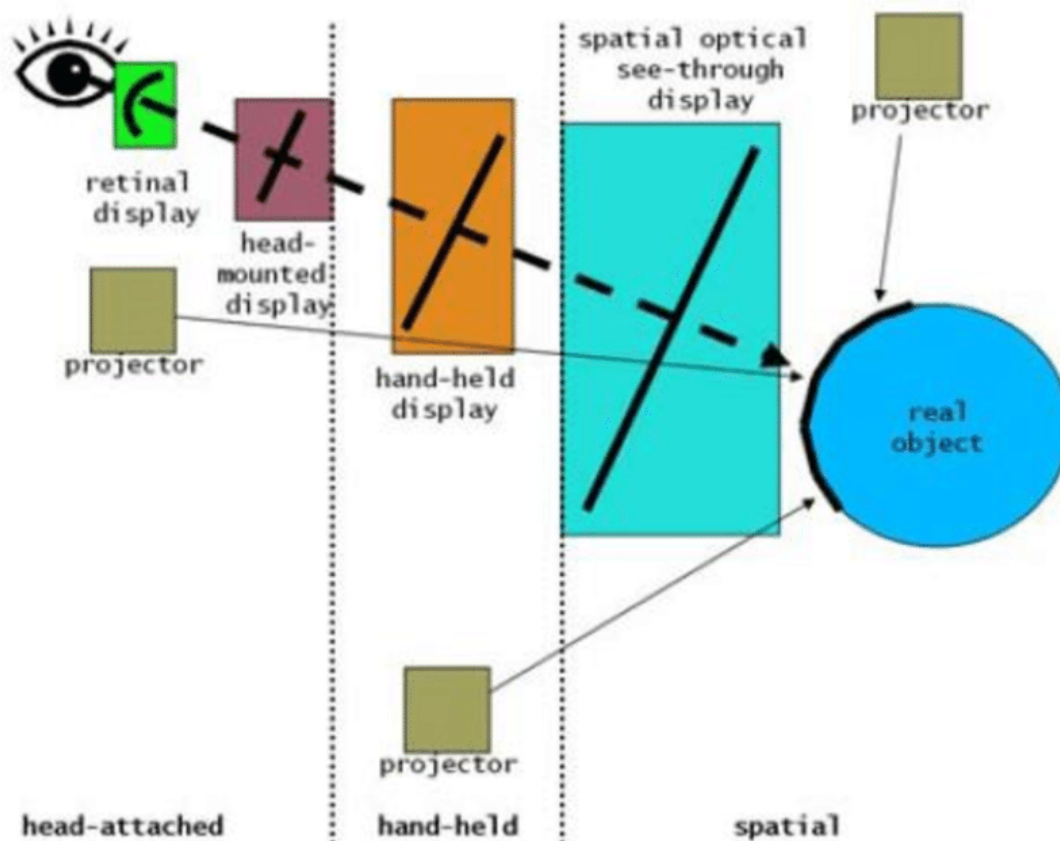
Obr. 2.1: Koncept kontinua rozšírenej reality podľa Milgram [20].

alita, kde väčšinu pohľadu používateľa nahrádza počítačová grafika, užívateľ však stále môže vidieť reálny svet. Približovaním sa k ľavej strane (RE), je rozšírená realita, kde virtuálny obsah zlepšuje pohľad užívateľa na skutočný svet. Ako sa do AR scény pridáva viac alebo menej virtuálneho obsahu, rozhranie sa posúva bližšie alebo ďalej od koncových bodov VE alebo RE. Hlavnou lekciou z tejto taxonómie je, že rozhrania AR neexistujú ako samostatný bod medzi reálnymi a virtuálnymi zážitkami, ale môžu sa objaviť kdekoľvek v rámci kontinua zmiešanej reality [9].

2.3 Zobrazovacie techniky

Podľa toho, ako sa kombinuje zobrazenie virtuálnych prvkov s reálnym svetom, je možné AR zobrazovacie techniky kategorizovať do 4 hlavných typov: (1) *texzobrazenie pomocou videa*, (2) *zobrazenie skrz optické systémy*, (3) *projekcia na fyzickú plochu* a (4) *eye multiplexed* [9].

- **zobrazenie pomocou videa (video based)** – tento typ zobrazenia kombinuje prvky virtuálneho sveta so záberom z videokamery. Do videa sa po nahraní vykreslia virtuálne prvky pomocou techník digitálneho spracovania obrazu. V mnohých prípadoch je videokamera pripevnená na zadnej strane displeja, otočená smerom k reálnej scéne, na ktorú sa používateľ pozerá pri sledovaní displeja. Tieto typy displejov vytvárajú digitálnu ilúziu videnia reálneho sveta "cez" displej, preto sa nazývajú "video priehľadné" displeje.
- **zobrazenie skrz optické systémy (optical see-through)** – využíva optické systémy pre kombinovanie virtuálnych prvkov s reálnym svetom. Optický systém v tomto type AR zobrazenia zvyčajne obsahuje rozdeľovače lúčov (napr. polovičné zrkadlá alebo kombinované hranoly), ktoré kombinujú pohľad na reálny svet videný cez rozdeľovač s odrazom obrazu z video displeja. V tomto prípade je to scéna reálneho sveta, ktorá sa odráža na polovičnom zrkadle, zatiaľ čo užívateľ môže skrz zrkadlo vidieť obraz videa.
- **projekcia na fyzickú plochu** – hlavným rozdielom oproti predchádzajúcim dvom je to že virtuálne prvky sú zobrazené priamo na fyzický objekt/plochu pomocou projektora.



Obr. 2.2: Vizualizácia zobrazovacích techník podľa umiestnenia displeja Bimler a Ras-
 kal [10].

- **eye multiplexed** – virtuálna scéna je reprodukována v reálnom prostredí, ale nezlučená s pohľadom na reálny svet. Na základe toho, že obraz nie je zlúčený s pohľadom na reálny svet, je na používateľovi aby si tieto dva obrazy v mysli spojil. Zobrazenie má menšie nároky na presnosť, keďže je obraz umiestnený vedľa reálneho sveta. Pre zobrazenie virtuálnej scény z pohľadu reálneho sveta je však postačujúca na to aby ju užívateľ vnímal ako rovnaký pohľad.

Okrem rozdelenia zobrazovacích techník podľa toho ako sa kombinuje zobrazenie virtuálnych prvkov s reálnym svetom, ich môžeme kategorizovať aj podľa toho kde je displej umiestnený (Obrázok 2.2).

Náhlavný displej

Náhlavné displeje poskytujú pohľad na virtuálne obrazy priamo pred oči užívateľa. Tento typ displeja patrí medzi najrozšírenejšie displeje v oblasti výskumu rozšírenej reality. Vďaka tomu, že má užívateľ displej umiestnený priamo na hlave, nie je medzi ním a displejom umiestnení už žiadny fyzický objekt, ktorý by virtuálne objekty zakrýval (Obrázok 2.3). Veľkosť môžu predstavovať tvar helmy, ale aj ľahké a tenké okuliare. Z hľadiska predchádzajúceho rozdelenia podľa zobrazenia virtuálnych prvkov sú náhlavné displeje využívané



(a) Náhlavný displej Magic Leap One – zobrazenie skrz optický systém.



(b) Náhlavný displej Google Glass Enterprise Edition 2 – zobrazenie pomocou projektora.



(c) Náhlavný displej MERGE AR/VR Headset – zobrazenie pomocou videa.

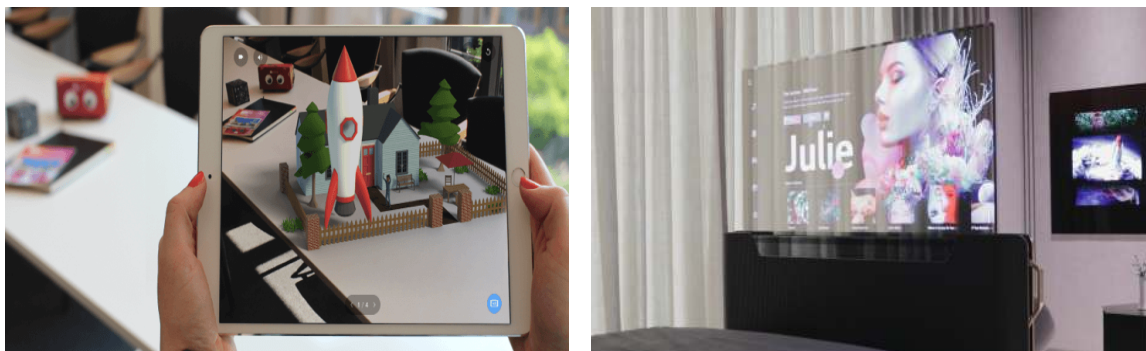
Obr. 2.3: Príklady typov náhlavných displejov podľa zobrazovania virtuálnych objektov¹.

prevažne k zobrazeniu pomocou videa a skrz optické systémy. Okrem toho však existujú aj technológie, ktoré využívajú zobrazenie pomocou projektora. Väčšina náhlavných displejov nie je v priamom kontakte s okom používateľa ale vyvíjajú sa aj priamejšie metódy ako je zobrazenie priamo na sietnicu oka.

Ručný displej

Ručné displeje sú obecné najrozšírenejší typ displeja zobrazovania virtuálnych objektov. Pre ich zobrazenie sa prevažne využíva zobrazenie pomocou videa cez zadnú kameru zariadenia. K typickým príkladom patrí tablet alebo mobilný telefón. Pre získanie presnosti pózy zariadenia sa prevažne používa kamera, ktorej môžu pomôcť aj iné vstavané senzory ako je gyroskop či akcelerometer. Aplikácie s rozšírenou realitou pre mobilné zariadenia sú populárne aj pre bežných užívateľov týchto zariadení. K najznámejším aplikáciám rozhodne patrí hra Pokemon Go, kde užívateľ na rôznych miestach v reálnom svete hľadá a zbiera virtuálne postavičky (Obrázok 2.4).

¹<https://www.softwaretestinghelp.com/best-augmented-reality-glasses>



Obr. 2.4: **V ľavo:** ručný displej – príklad aplikácie CoSpaces Edu². **V pravo:** priestorový displej – príklad transparentnej televízie LG³

Priestorový displej

Oproti náhlavným a ručným displejom, ktorými je možné pohybovať spolu s užívateľom, priestorové displeje sú pevne položené na konkrétnom mieste. Medzi príklady priestorových displejov patria napríklad transparentné LCD alebo OLED obrazovky. Tie zvyknú byť využívané vo vitrínach obchodov ale dnes sú k dispozícii aj transparentné televízie (Obrázok 2.4). Priestorové displeje môžu využívať aj vizualizáciu AR založenú na videu. Stolný monitor spárovaný s videokamerou smerujúcou na stôl pred monitorom je jedným zo široko používaných nastavení, aké Desktop AR alebo Video-Monitor AR využíva pri aplikáciách AR.

²<https://cospaces.io/edu/tech-check/ar-tablet.html>

³<https://edition.cnn.com/videos/business/2021/01/11/lg-transparent-tv-orig.cnn-business>

Kapitola 3

Microsoft HoloLens 2

Microsoft HoloLens 2 je technológia vyvinutá spoločnosťou Microsoft a ako uvádza sám výrobca ide o samostatný holografický počítač, ktorý vylepšuje funkcionality a nástroje svojho predchodcu Microsoft HoloLens. V užšom slova zmysle ide o optický priehľadný náhlavný počítač (HMC - head-mounted computer), vytvorený pre rozšírenú realitu [28]. HoloLens 2 beží na operačnom systéme Windows Holographic OS, ktorý je založený na Windows 10.

Nasledujúca sekcia 3.1 se venuje popisu základných komponentov hardware. V sekcii 3.2 sú vysvetlené možnosti interakcie so zariadením, medzi ktoré patrí ovládanie pomocou rúk, ovládanie hlasom či ovládanie pohľadom. Aj keď toto zariadenie je k dispozícii len niekoľko rokov, výrobca ponúka niekoľko možností a rozšírení pre tvorbu aplikácií v Microsoft HoloLens 2. Tie, ktoré boli využité pri písaní tejto práce sú popísane v sekcii 3.3. V poslednej sekcii 3.4 je uvedených niekoľko príkladov aplikácii, ktoré boli vyvíjané pre HoloLens a sú rozdelené do 4 kategórií: *Medicína a zdravotná starostlivosť*, *Vzdelávanie*, *Priemysel*, *Architektúra a stavebníctvo*.

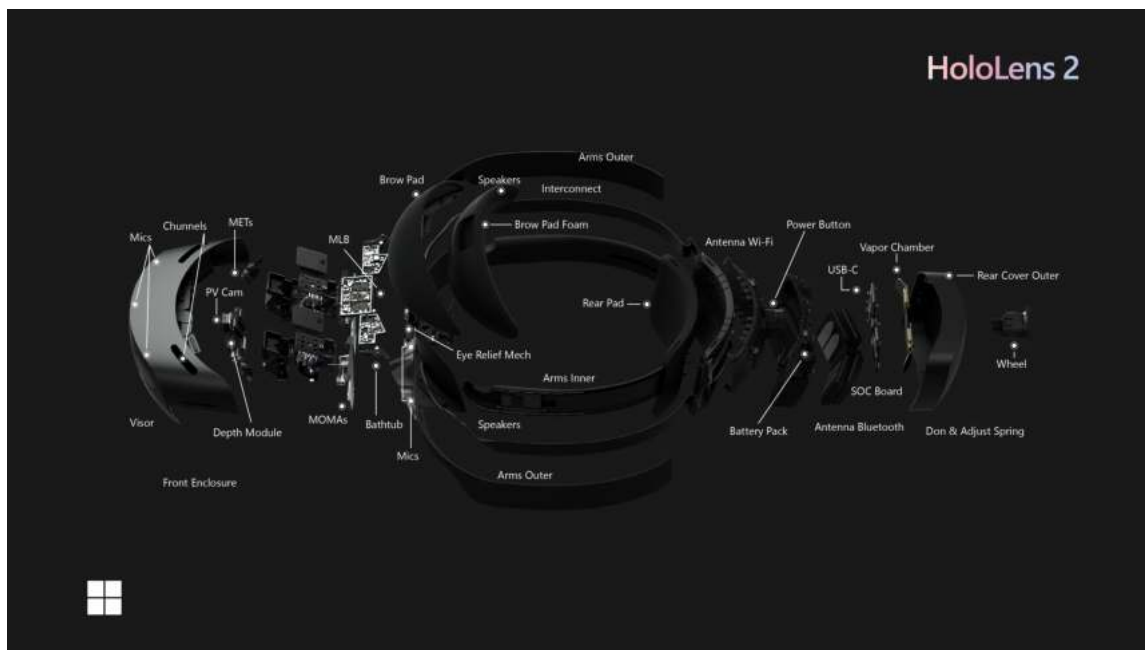
3.1 Hardware

Druhá generácia prináša niekoľko vylepšení oproti prvej, vrátane väčšieho pozorovacieho poľa, samostatného jadra DNN¹, sledovania pohľadu očí či plne sledovanie pohybu rúk. Zariadenie je vybavené druhou generáciou na mieru vyrobenou holografickou jednotkou (HPU² 2.0), ktorá umožňuje nízko energetické počítačové videnie v reálnom čase. Na HPU spúšťa všetky algoritmy spojené s počítačovým videním (sledovanie hlavy, sledovanie rúk, sledovanie pohľadu očí, priestorové mapovanie) a hostí jadro DNN. Nachádza sa na prednej časti zariadenia v blízkosti senzorov. CPU na SoC³ (Qualcomm Snapdragon 850) zostáva plne k dispozícii pre aplikácie. SoC sa nachádza na zadnej strane. Zariadenie je vybavené hĺbkovou a RGB kamerou, štyrmi kamerami a inerciálnou meracou jednotkou (IMU). Zvuk sa zachytáva pomocou mikrofónového poľa (5 kanálov) [27] (Obrázok 3.1).

¹Hlboká neurónová sieť (anglicky Deep Neural Network) – neurónová sieť s určitou úrovňou zložitosti, zvyčajne aspoň dvoma vrstvami. Hlboké siete spracúvajú údaje komplexným spôsobom pomocou sofistikovaného matematického modelovania.

²Holografická procesorová jednotka (HPU) je názov spoločnosti Microsoft pre koprocesor v náhlavnom displeji HoloLens. HPU vykonáva spracovanie, ktoré integruje skutočný svet a údaje pre rozšírenú realitu.

³System na čipe (SoC) je integrovaný obvod (známy aj ako "čip"), ktorý integruje všetky alebo väčšinu komponentov počítača alebo iného elektronického systému.



Obr. 3.1: Rozloženie komponentov Microsoft HoloLens 2.

3.2 Možnosti interakcie

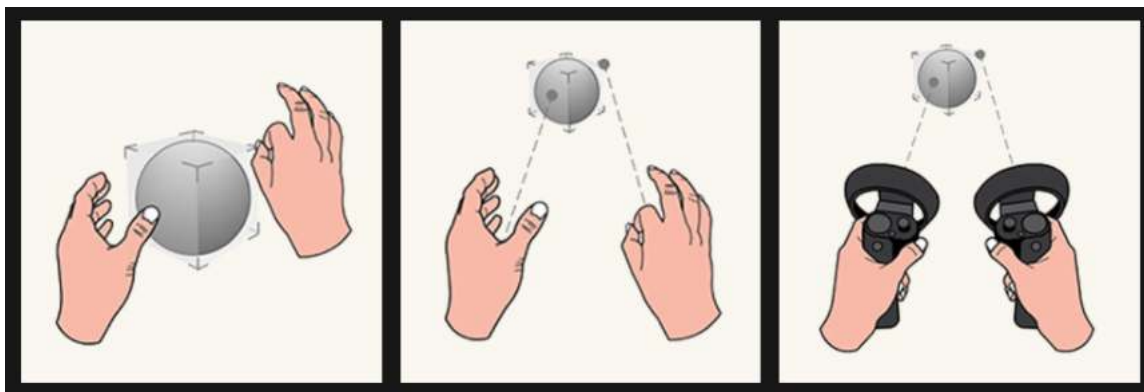
Výrobca sa pri návrhu interakčných modelov pri zmiešanej realite zameral na tri typy, ktoré pospájal do modelov multimodálnej interakcie [3]. Tieto modely interakcie zahŕňajú sledovanie rúk a očí spolu so vstupom prirodzeného jazyka. To, ktorý model je najvhodnejší pre využitie v aplikácii závisí od rôznych faktorov, ktoré výrobca presnejšie špecifikoval pri každom modeli: *Ovládanie pomocou rúk a gest*, *Ovládanie bez rúk*, *Ovládanie pohľadom*.

Ovládanie pomocou rúk a gest

Pomocou rúk je možné s objektami manipulovať priamo, pomocou pointera alebo ovládačmi (Obrázok 3.2).

Priama manipulácia je vstupný model, ktorý zahŕňa dotýkanie sa hologramov priamo rukami. Myšlienkou tohto konceptu je, že predmety sa správajú tak, ako by sa správali v reálnom svete. Tlačidlá sa dajú aktivovať jednoduchým stlačením, predmety sa dajú vyberať uchopením a 2D obsah sa správa ako virtuálna dotyková obrazovka. Priama manipulácia je užívateľsky prívetivá. Používateľ sa nemusí učiť žiadne symbolické gestá. Všetky interakcie sú postavené na vizuálnom prvku, ktorého sa môžete dotknúť alebo ho chytiť. Považuje sa za „blízky“ vstupný model, pretože sa najlepšie používa na interakciu s obsahom na dosah ruky.

Ukázať pointerom a potvrdiť rukami je vstupný model, ktorý používateľom umožňuje zacieliť, vybrať a manipulovať s 2D a 3D obsahom mimo dosahu. Táto technika „ďalekej“ interakcie je jedinečná pre zmiešanú realitu, pretože ľudia takto prirodzene neinteragujú so skutočným svetom. Tento model prelomuje fyzické obmedzenia reálneho sveta čím zefektívňuje a zefektívňuje interakcie používateľov.



Obr. 3.2: Ovládanie pomocou rúk a gest.

Ovládače sú hardvérové príslušenstvo, ktoré používateľom umožňuje vykonávať akcie v zmiešanej realite. Výhodou ovládačov oproti gestám je, že ovládače majú presnú polohu v priestore, čo umožňuje jemnú interakciu s digitálnymi objektmi.

Ovládanie bez rúk

V niektorých prípadoch potrebuje používateľ používať ruky k naplneniu práce v reálnom svete.

K ovládaniu bez rúk patrí ovládanie hlasom, kedy užívateľ môže interagovať s užívateľským prostredím vyslovením názvu tlačidla alebo iných preddefinovaných príkazov. To však nie je vždy možné (napr. hlučné prostredie) a vtedy je možné využiť ovládanie „pozeraním“ sa na hologram a tým aktivuje jeho funkcionality.

Ovládanie pohľadom

Ovládanie pohľadom je kombinácia predchádzajúcich dvoch možností kedy užívateľ môže zamerať objekt hlavou, očami alebo rukou a následne ho aktivuje dlhším pozeraním, gestom, hlasom alebo špeciálnym vstupom.

3.3 Tvorba aplikácií pre Microsoft HoloLens 2

Spoločnosť Microsoft ponúka hneď niekoľko možností či rozšírení, ktoré môžu nejakým spôsobom zlepšiť alebo pomôcť vývojárom pri tvorbe aplikácií pre Microsoft HoloLens 2.

Universal Windows Platform

Universal Windows Platform (UWP) je jedna z viacerých možností pre tvorbu aplikácií, ktoré bežia na Windows 10 alebo Windows 11 a môže byť kombinovaná s inými platformami [4]. Výrobca špecifikuje tieto vlastnosti:

- Zabezpečenie: UWP aplikácie prehlásia, ku ktorým zdrojom a dátam zariadenia majú prístup a používateľ musí prístup autorizovať.
- Možnosť používania spoločnej API na všetkých zariadeniach so systémom Windows 10.

- Schopnosť používať funkcie špecifické pre zariadenie a prispôbiť používateľské rozhranie rôznym veľkostiam obrazovky zariadenia, rozlíšeniam a DPI.
- Dostupné v obchode Microsoft Store na všetkých zariadeniach (alebo iba na tých, ktoré sa určia), ktoré bežia na Windows 10.
- Programovateľné v C#, C++, Visual Basic a Javascript.

Unity

Unity⁴ je nástroj od spoločnosti Unity Technologies pre tvorbu 2D a 3D hier. Logika aplikácii je programovaná v jazyku C#. Ponúka široké množstvo platforiem, pre ktoré je možné hry vyvíjať. Vráťane, stolných PC so systémom Windows, MAC či Linux, mobilné zariadenia Android a iOS, hracie konzoly ako je PlayStation, Xbox, Nintendo Switch či Google Stadia. Okrem toho však Unity podporuje VR či MR technológie vrátane HoloLens. Tvorba aplikácii pre HoloLens je oficiálne podporovaná využitím UWP.

Mixed Reality Toolkit

Mixed Reality Toolkit⁵ (MRTK) je projekt vytvorený spoločnosťou Microsoft a poskytuje súbor komponentov a funkcií, ktoré sa používajú na urýchlenie a zjednodušenie vývoja aplikácii so zmiešanou realitou v nástroji Unity.

Poskytuje napríklad multiplatformový vstupný systém pre priestorovú interakciu a používateľské rozhranie, umožňuje rýchle prototypovanie prostredníctvom simulácií v editore, funguje ako rozširiteľný rámec, ktorý poskytuje vývojárom možnosť vymieňať a rozširovať základné komponenty.

3.4 Prípady použitia

Výskum v oblasti vývoja aplikácii pre rozšírenú, virtuálnu a zmiešanú realitu (skrátene spolu XR) sa za posledné roky niekoľko násobne zvýšil a už dnes je možné sledovať niektoré prípady použitia v praxi. Tejto téme sa podrobnejšie venoval Sebeom Park a spol. [25], ktorý zhrnuli 44 štúdií, ktoré sa zamerali na tvorbu aplikácii pre Microsoft HoloLens.

Medicína a zdravotná starostlivosť

Pre medicínske a chirurgické pomôcky a systémy sa HoloLens využívajú na účely, ako je vizualizácia medicínskych údajov, vyhľadávanie ciev, podpora pre zameranie ihly alebo vrtačky a podpora pre endoskop. Používa sa najmä ako pomocný prostriedok, prostredníctvom ktorého sa poskytujú informácie súvisiace s chirurgickým zákrokom alebo sa zlepšuje presnosť operácie.

Hanna a spol. [16] hodnotili užitočnosť Microsoft HoloLens v klinických a neklinických aplikáciách patológie. Chien a spol. [14] navrhli MR systém na lekárske použitie, ktorý dokáže vizualizovať medicínske údaje na telo pacienta s kombináciou senzora Intel RealSense. Mojica a spol. [21] navrhli prototyp holografického rozhrania, ktorý dokáže vizualizovať 3D dáta magnetickej rezonancie pre plánovanie neurochirurgických zákrokov.

⁴<https://unity.com/>

⁵<https://github.com/microsoft/MixedRealityToolkit-Unity>

Vzdelávanie

V oblasti medicínskeho vzdelania je AR prospešná pre vzdelávanie študentov a rovnako tak je rozsiahle používaná v získaní medicínskej praxe a simulácii. Okrem toho využitie AR v telemedicíne, ktorá poskytuje lekárske služby na diaľku, sa zlepšujú problémy vo vidieckych oblastiach, ktoré nemajú prístup k lekárskeým službám.

Moro a spol. [22] hodnotili a porovnávali efekty učenia pri prednášaní prostredníctvom AR a tabletov. Condino a spol. [15] vyvinuli hybridný tréningový systém pre ortopedickú operáciu a vyhodnotili potenciál MR v tréningovom systéme. Caligiana a spol. [13] vyvinuli aplikáciu, ktorá je užitočná pre úpravu a rezanie virtuálnych objektov a návrh simulácii chirurgickej operácie. Celkovo zosumarizovali takto 5 štúdií.

Priemysel

Keďže sa vo výrobnom priemysle zavádzajú a čoraz viac využívajú priemyselné roboty, je dôležité izolovať a monitorovať pracovisko pracovníkov a robotov. Momentálne sa vo výrobnom priemysle využívajú na údržbu, opravy a dodatočné vybavenie robotov, ako aj na montáž produktov a riadenie výroby.

Hietanen a spol. [17] navrhli AR interaktívne používateľské rozhranie na monitorovanie a zabezpečenie minimálnej ochrannej vzdialenosti medzi robotom a operátorom v oblasti priemyselnej výroby. Vorraber a spol. [29] prezentovali výsledky výskumu v ktorom využili AR výlučne s hlasovou komunikáciou pre vzdialenú údržbu strojov a systémov. Mourtzis a spol. [23] zase navrhli aplikáciu, ktorá načíta a zobrazí plán výroby a spracované údaje pre riadenie výroby.

Architektúra a stavebníctvo

Mnoho miest sa dnes príliš rýchlo transformuje na smart mestá, preto aj v tejto oblasti sa odborníci čo raz viac snažia nájsť uplatnenie AR technológií. V tomto odvetí sa AR technológia využíva pri mapovaní priestorov alebo vizualizovaní dát.

Zhang a spol. [32] navrhol metódu na vizualizáciu 3D modelov miest v Toronte (Kanada) a rôznych iných typov mestských údajov. Bahri a spol. [8] navrhli aplikáciu na rozmiestňovanie nábytku v miestnosti pre vytvorenie systému informačného modelu budov – BIM⁶. Wu a kol. [30] navrhli nový riadiaci systém pre plánovanie ciest všesmerových mobilných robotov.

⁶Informačný model budovy (anglicky Building Information Modelling alebo Building Information Management, skrátene BIM) je proces vytvárania a správy dát o budove počas celého jej životného cyklu. V podstate ide o digitálny model, ktorý reprezentuje fyzický a funkčný objekt s jeho charakteristikami; niečo ako „digitálne dvojča“ [1].

Kapitola 4

AREditor

AREditor je aplikácia, ktorá je súčasťou systému ARCOR2 (Augmented Reality Collaborative Robot) s cieľom zjednodušiť programovanie úloh pre kolaboratívneho robota. Systém bol vytvorený výskumnou skupinou Robo@FIT na univerzite Vysoké učení technické v Brne.

4.1 Popis a motivácia

Ako vo svojej práci tvrdia Kapinus a spol. [19], práca užívateľov, ktorý používajú užívateľské rozhrania pre interakciu s robotom, je často krát ovplyvnená viacerými aspektami: musia niekoľko krát meniť pozíciu práce medzi pracoviskom a počítačom alebo slabá spätná väzba od robota zhoršuje komunikáciu medzi robotom a človekom v prípade slabej vizuálnej alebo zvukovej odozvy. Na druhej strane však v prípade využitia dnešných vyspelejších technológií, ktoré sprístupňujú prirodzenejšiu interakciu medzi človekom a robotom, ich užívateľské rozhranie je stále založené na 50 ročnom starom koncepte.

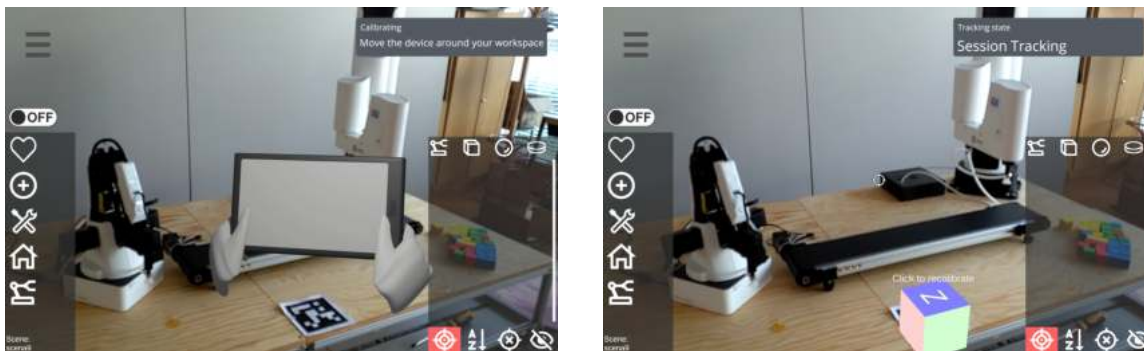
Preto navrhli systém, ktorý má odstrániť nedostatky 2D užívateľského rozhrania a posunúť ho do 3D sveta. AREditor je teda interaktívna mobilná aplikácia, presnejšie užívateľské rozhranie s využitím AR, ktoré vizualizuje 3D objekty scény (viac v sekciách 4.2, 4.3) do reálneho priestoru kde používateľ môže s objektami interagovať a následne tak aj definovať akcie pre nastavenie robota. Zároveň ide o klientsku časť aplikácie ktorá je súčasťou systému ARCOR.

4.2 Reprezentácia programu

Program sa skladá z niekoľkých častí. Aby boli 3D objekty scény vyobrazené na určenom mieste je potrebné najprv scénu ukotviť pomocou kalibračného bodu. Ďalej je potrebné rozlišovať scénu a projekt. Zatiaľ čo scéna definuje rozloženie objektov v scéne, v projekte je možné k objektom scény pridať správanie. Ku každej scéne je možné vytvárať viacero projektov, čo vlastne znamená, že robotické pracovisko môže vykonávať viacero rôznych úloh.

Kalibrácia scény

Aplikácia je navrhnutá tak, aby priestor a súradnicový systém nebol statický, ale bolo ho možné dynamicky meniť na základe kotvy, ktorá reprezentuje začiatok súradnicového sys-



Obr. 4.1: Kalibrácia scény v AREditore pre tablet

tému. K tomu bola využitá, v rozhraní pre tablet, knižnica ARFoundation¹, ktorá sleduje priestor viditeľný kamerou zariadenia a hľadá značku (marker) označujúcu začiatok súradnicového systému scény. Keď značku zaregistruje, je jej priestor vyznačený 3D modelom kocky a na scéne sa zobrazia objekty scény relatívne k značke a ich skutočnej pozícii v scéne (Obrázok 4.1). Po spustení scény alebo projektu je kalibrácia spustená automaticky a je možné scénu kedykoľvek recalibrovať.

Scéna - definovanie objektov pracoviska

Program pozostáva z dvoch kľúčových komponentov. Jednou z nich sú tzv. scény a druhým komponentom sú tzv. projekty. Scéna reprezentuje objekty na pracovisku v reálnom svete, ktoré sú kľúčové pre programovanie robotického pracoviska: roboti, senzory, kolízne objekty (Obrázok 4.2). Okrem kolíznych objektov, ktoré je možné reprezentovať základným 3D telesom (kocka, kváder, guľa) sú zvyšné modely scény uložené na serverovej časti aplikácie. Na klientskej časti je uložený iba základný model robota či senzoru, ktoré sú po načítaní modelu zo servera nahradené reprezentáciou konkrétneho modelu. Spracovanie modelu za behu programu je v nástroji Unity komplikovanejší problém a preto bol v aplikácii využitý balík z tretích strán - TriLib². Aplikácia je teda nezávislá na type objektu a všetku logiku a vlastnosti sa stará server zatiaľ čo klientska časť spracuje model vo glTF³ formáte.

Projekt - definovanie správania sa robotov

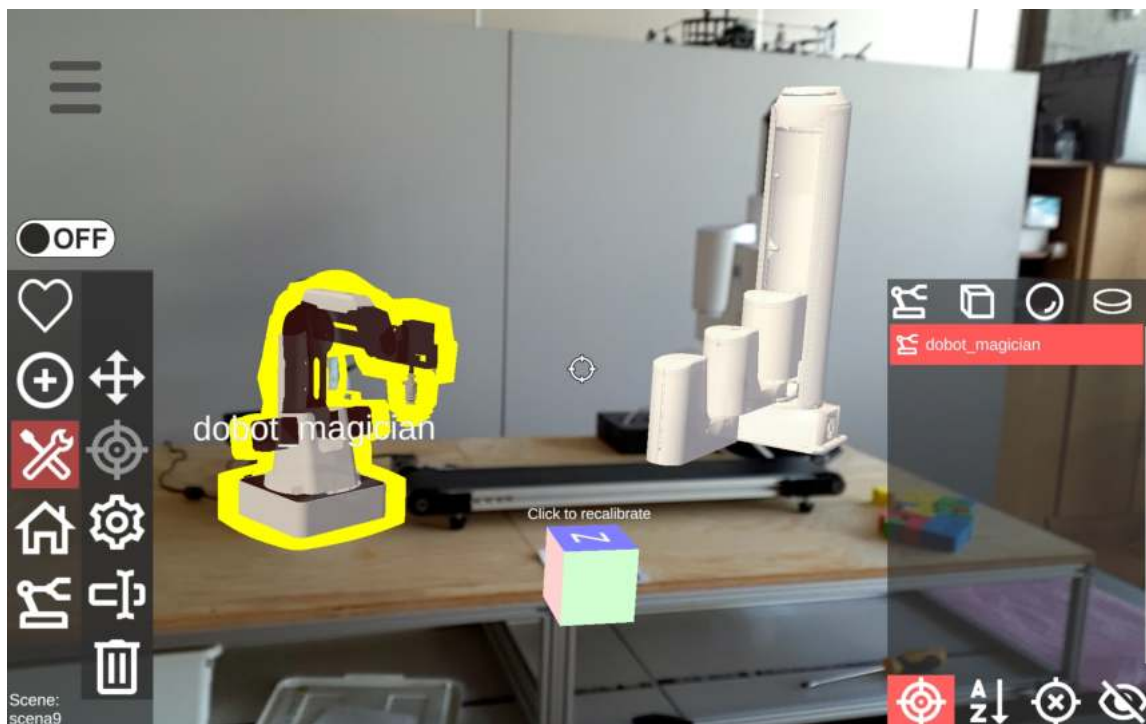
Projekt je súčasťou scény a definuje správanie sa robotov. Ide teda o navrhnutie operácii konkrétneho scenára využitia robota. Každá scéna môže mať vytvorených niekoľko projektov, pričom každý z nich predstavuje iný scenár toho istého prostredia. Reprezentácia operácii bola inšpirovaná vývojovým diagramom v 3D priestore.

Bod v priestore je reprezentovaný tzv. akčnými bodmi, nad ktorými sú definované konkrétne operácie. Tieto akčné body sú reprezentované malou guľou a ich hlavnou úlohou je vizualizovať užívateľovi kde bude operácia/operácie vykonaná/é (Obrázok 4.3). Okrem toho môže reprezentovať aj nastavenie rúk robota.

¹<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.2/manual/index.html>

²<https://assetstore.unity.com/packages/tools/modeling/trilib-2-model-loading-package-157548#description>

³glTF (GL Transmission Format) je formát 3D súboru, ktorý ukladá informácie o 3D modeli vo formáte JSON. Použitím JSON sa minimalizuje veľkosť 3D aktív a runtime spracovanie potrebné na rozbalenie a použitie týchto aktív.[2]

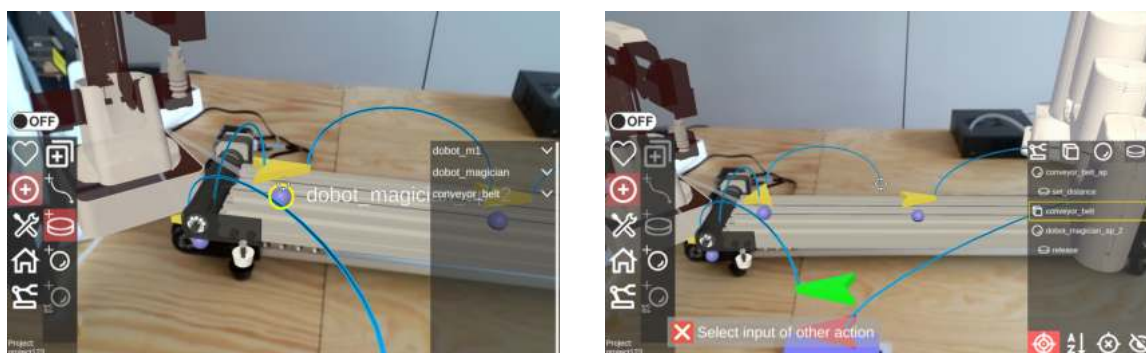


Obr. 4.2: Zobrazenie robota v scéne

Operácia je reprezentovaná šípkami, ktorej smer určuje smer operácii. Konkrétne akcie opäť závisia od typu robota a teda ich zoznam sa získava zo serveru. Medzi základné patrí *pick* (zdvihnutie), *suck* (nasatie), *place* (položenie).

Operácie samé o sebe nezabezpečujú postupnosť krokov. Tá je v aplikácii zabezpečená spojmi, ktoré sú umiestnené medzi výstupom predchádzajúcej operácie a vstupom tej ďalšej. Spoje sú reprezentované modrou krivkou.

Okrem toho sú pri vytváraní nového projektu vždy preddefinované dve operácie, z ktorých jedna reprezentuje štart (zelená šípka) a druhá koniec (červená šípka). Tieto operácie slúžia ako vstupný a výstupný krok scenára a je vždy potrebné aby prvý spoj vychádzal zo štartovacej šípky a ten posledný vstupoval do tej konečnej.



(a) Pridanie akcií k objektu.

(b) Vytvorenie prepojenia medzi akciami.

Obr. 4.3: Zobrazenie akčných bodov a akcií.

Komunikácia so serverom

Ako bolo už spomenuté vyššie AREditor je len klientskou časťou aplikácie a získanie a rozloženie v 3D scéne zabezpečuje server. Preto je potrebné popísať základne funkcionality spojené s komunikáciou. Komunikáciu so serverom zabezpečuje knižnica tretej strany *NativeWebSocket*⁴. Dáta, ktoré prichádzajú zo servera sú rozdelené podľa toho či sa jedná o žiadosť, odpoveď alebo udalosť. Najdôležitejšou súčasťou sú práve udalosti. Tie reagujú a posielajú sa na základe posledného stavu aplikácie a zároveň určujú stav aplikácie. Medzi základné udalosti možno zaradiť otvorenie scény, projektu či balíka kde súčasťou sú aj dáta a informácie o objektoch v danej scéne. Ďalšie udalosti sú napr. otvorenie hlavného menu editora alebo posielanie informácií o zmene stavu scény či projektu. Porozumenie toho, ako jednotlivé udalosti ovplyvňujú stav aplikácie v užívateľskom rozhraní je jeden z prvých hlavných krokov pri tvorbe užívateľského rozhrania v AREditore pre Microsoft Hololens 2.

4.3 Interakcia s objektami

Manipulácia s 3D objektmi je zabezpečená viacerými spôsobmi. V prípade označenia (výberu) nejakého objektu v scéne je buď možné jednoducho zamerať objekt zameriavačom umiestneným v strede obrazovky, alebo v prípade ak je objekt obtiažnejšie zamerať využiť menu, v zozname zobrazených objektov vybrať jeden z objektov a následne s ním interagovať.

Premiestnenie a rotácia

Premiestniť alebo otočiť objekt je možné rolovacím komponentom, kde smer posunu (osa x, osa y, osa z) sa určí jej výberom pomocou zameriavača (Obrázok 4.4). Tento pohyb však môže byť niekedy pomalý a preto je možné využiť tlačidlo s ikonkou ruky. Pri držaní tohto tlačidla je možné jednoducho posúvať zariadením a tým sa pohybuje aj vybraný objekt. Táto možnosť je však nepresnejšia.

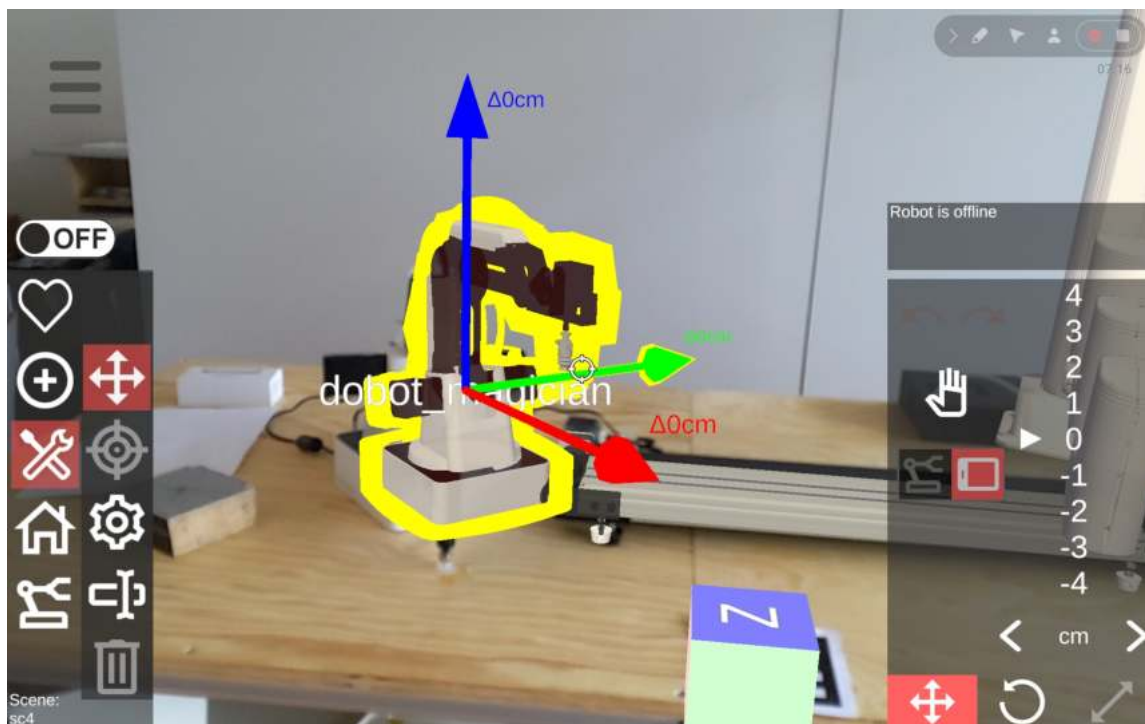
Akčné body a operácie

Akčné body môžu byť definované globálne alebo pridané do hierarchie, čo znamená, že bod má definovaného rodiča a v prípade posunu rodiča je posunutý aj akčný bod. Označením akčného bodu je možné k bodu pridať operáciu zo zoznamu. Operácia je po vytvorení vizualizovaná nad bodom a je možné ju premiestniť. Následne je možné jej označením pridať k nej spoj, kde tento spoj vystupuje z danej akcie a užívateľ si určí kam vstupuje. Jediné obmedzenie je, že spoj nemôže vstupovať do definovanej štartovacej akcie a naopak, vystupovať z definovanej konečnej akcie.

Simulácia scenáru, operácie

V aplikácii je možné nasimulovať vytvorený scenár alebo konkrétnu operáciu (operácie). Scenár je možné nasimulovať buď priamo z editora Projektu, označením štartovacej akcie, alebo vytvorením balíka a následne spustením zo sekcie pre zoznam balíkov. Simulácia dokáže zobraziť pohyb robota a vykonanie akcií.

⁴<https://github.com/endel/NativeWebSocket>



Obr. 4.4: Manipulácia s robotom v scéne.

4.4 Zhrnutie aplikácie

Aplikácia prináša mnoho výhod oproti doteraz zaužívaným užívateľským rozhraniám. Jednou z nich je napríklad vizualizácia objektov pre programovanie robotického pracoviska, kde Kapinus a spol. [19] navrhli riešenie v ktorom presunuli objekty z 2D prostredia do 3D. To umožňuje užívateľovi jednak, lepšie si predstaviť prostredie svojho pracoviska, ale zároveň s ním aj priamo interagovať. Zároveň však tým, že bola odstránená bariéra prepínaní užívateľa medzi pracoviskom a zariadením, sa vytvoril komplexnejší prípad použitia. Užívateľské rozhranie je síce pomerne jednoduché a intuitívne a užívateľ by nemal mať problém sa rýchlo naučiť pracovať so zariadením, ale sú isté limity, ktoré mobilné zariadenia prinášajú. Medzi ne napríklad patrí manipulovanie s objektmi, ktoré je limitované 2D plochou. Ďalej je to veľkosť zariadenia, kde je potrebné zabezpečiť dostatočne veľký priestor pre zobrazenie prostredia ale zároveň vedieť stále jednoducho pracovať aj s užívateľským prostredím. A aj keď prináša výrazne zlepšenie v interakcii s prostredím, užívateľ musí mať stále zamestnané ruky držaním zariadenia, čo môže prinášať diskomfort ale aj úbytok efektívnosti.

Kapitola 5

Užívateľské rozhranie pre robotické pracovisko

Táto kapitola sa venuje existujúcim riešeniam tvorby užívateľského rozhrania pre robotické pracovisko. V sekcii 5.1 je zanalyzovaný súčasný stav. Zhodnotenie stavu a motivácia k tvorbe rozhrania pre programovanie robotického pracoviska v Microsoft HoloLens 2 je opísaná v sekcii 5.2. V sekcii 5.3 bol predstavený návrh základných UI elementov aplikácie a v sekcii 5.4 interakciu s robotom v prostredí Microsoft HoloLens 2.

5.1 Existujúce riešenia

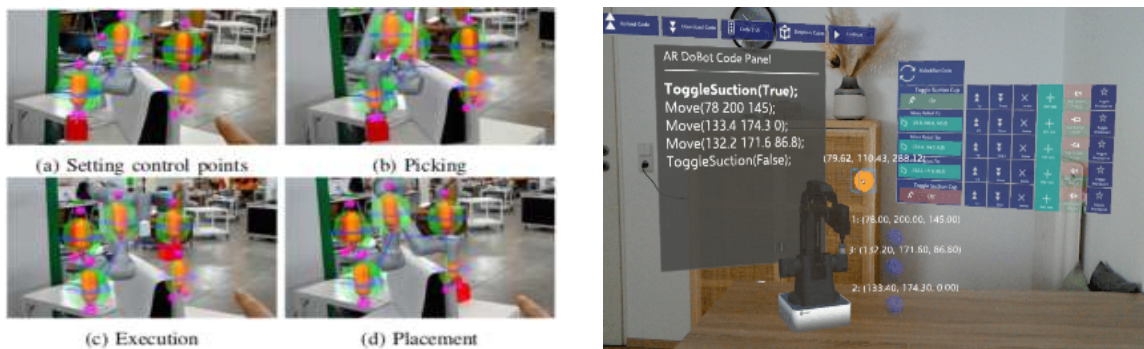
Transformácia výroby k využívaniu robotov, vedie k tomu že narastá dopyt po zefektívnení práce užívateľa s robotom a jeho bezpečné používanie. Preto sa aj čoraz väčšie množstvo odborníkov a výskumníkov zameriava na využitie XR pri práci s robotom. Predchádzajúca kapitola 4 popisovala platformu AREditor¹, kde skupina výskumníkov z Robo@Fit navrhla s využitím AR užívateľské rozhranie pre programovanie robotického pracoviska v mobilnom zariadení. Programy sú reprezentované navzájom prepojenými akciami, ktoré sú viazané k miestu, kde sa majú vykonať.

V ďalšej práci Huang a spol. [18] predstavili systém Code3, navrhnutý pre mobilné zariadenia, ktorý integruje tri základe komponenty, ktoré sú potrebné pre komplexné programovanie úloh. Prvým komponentom je CustomLandmarks, ktorý umožňuje užívateľom vytvárať knižnicu perцепčných orientačných bodov. Druhým je CustomActions, ktorý umožňuje užívateľom vytvárať knižnicu manipulačných akcií. Ide o kinestetický systém programovania. Tretia komponenta, CodeIt, je programovacie rozhranie na báze drag-and-drop operácie, ktorá umožňuje užívateľom definovať logiku toku riadenia.

Technológiu Microsoft HoloLens využili vo svojej práci Enes a spol [31], ktorý navrhli jednoduché programovacie prostredie v rozšírenej realite s rozšíreným ladením (SPREAD²). SPREAD využíva AR pre vizualizovanie robota ako aj pre programovanie prostredia na pracovisku. SPREAD sa skladá z troch hlavných častí. Na vyhodnotenie svojich návrhu si zavolali odborníkov z oblasti priemyselnej automatizácie, robotiky či AR, ktorý zhodnotili že navrhnuté riešenie má potenciál uľahčiť a obohatiť súčasne procesy programovania robotov (Obrázok 5.1).

¹<https://github.com/robofit/arcor2/wiki/Architecture>

²Simple Programming Environment in Augmented Reality with Enhanced Debugging



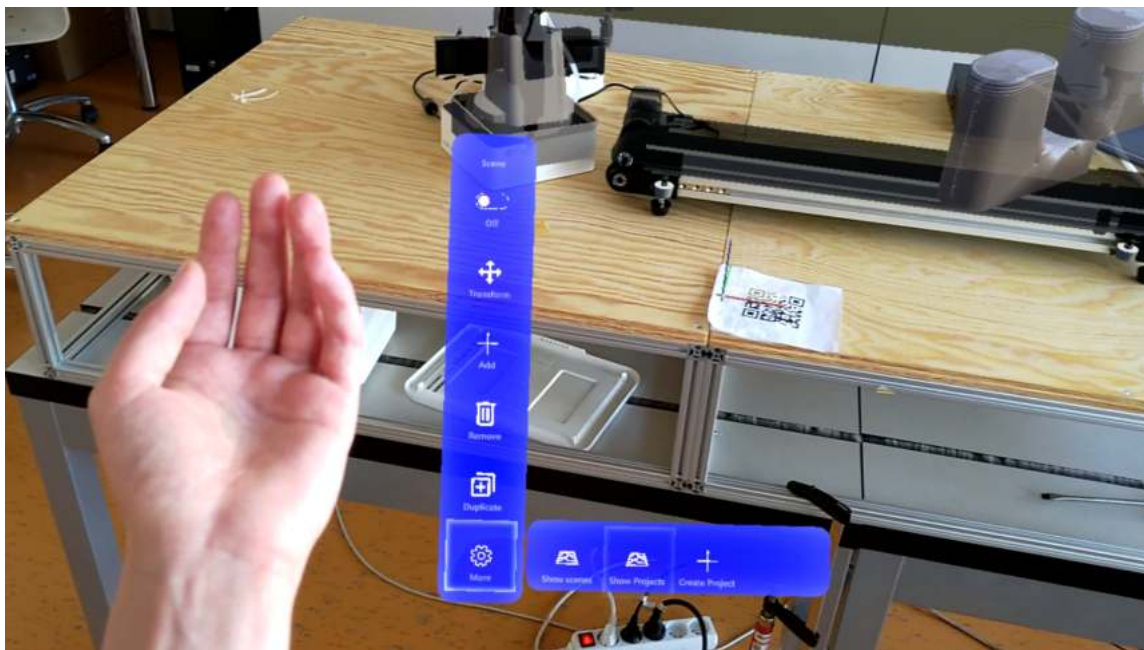
Obr. 5.1: **V ľavo:** Kroky pre vytvorenie úlohy pre robota Ostatnin a spol. [24] . **V pravo:** Program robota, model robota a vytváranie úloh založených na blokoch Enes a spol [31].

V ďalšej práci Ostatnin a spol. [24] v ktorej využili technológiu Microsoft HoloLens, vyvinuli systém, v ktorom sú okuliare pripojené cez ROS³ k nástroju Unity a manipulátoru robota. Systém umožňuje – rozpoznať skutočnú polohu robota analýzou množiny bodov v súradnicovom systéme (point cloud), využiť virtuálne body a menu pre tvorbu úloh, generovanie trajektórie pre tvorbu simulácii alebo spustenie na reálnom stroji. Poskytuje tiež možnosť škálovania virtuálneho a reálneho sveta pre presnejšie plánovanie (Obrázok 5.1).

5.2 Zhodnotenie a motivácia

Predchádzajúca kapitola ponúka širší pohľad na to, akým spôsobom je možné využiť prvky rozšírenej reality pri programovaní robota, špeciálne s použitím okuliarov Microsoft HoloLens 2. Ukazuje sa, že momentálne stále dominujú nástroje, pri ktorých musí užívateľ fyzicky pracovať so zariadením, či už ide o metódy s alebo bez využitia rozšírenej reality. Napriek tomu dnes vidíme veľký potenciál v technológiách ako je napr. Microsoft HoloLens pretože môžu odstrániť niektoré nedostatky tradičných metód. Jednou z hlavných motivácií pre využitie náhlavných zariadení je, že užívateľ lepšie vníma umiestnenie virtuálneho obsahu oproti tabletu vďaka stereoskopii. Tablet má len jeden displej a pre vnímanie hĺbky a správneho umiestnenia virtuálnych prvkov je potrebné sa na neho pozrieť z viacerých uhlov. Keďže u okuliarov sú displeje dva, je možné vidieť stereoskopicky a preto je možné omnoho lepšie vnímať hĺbku a ukotvenie virtuálnych prvkov. Okrem toho sa užívateľovi uvoľnia ruky a prostredie pracoviska má priamo pred sebou vďaka čomu je ušetrený od neustáleho prepínania sa medzi pozeraním na zariadenie a prostredie. Ďalšou možnou výhodou oproti napr. mobilnej verzii AREditor je, že užívateľ by mal vedieť prirodzenejšie manipulovať s objektmi v 3D scéne čo môže ušetriť ďalší čas pri práci. Preto bolo v ďalšej časti diplomovej práce navrhnuté užívateľské rozhranie pre Microsoft HoloLens 2, ktorého využiteľnosť a funkčnosť bola následne otestovaná a porovnaná s mobilnou verzou programu.

³Robotický operačný systém (ROS) je súbor softvérových knižníc a nástrojov, ktoré pomáhajú vytvárať robotické aplikácie.



Obr. 5.2: Zobrazenie menu pri pozeraní sa na dlaň ruky.

5.3 Návrh elementov užívateľského rozhrania AREditor pre Microsoft HoloLens 2

Pri vytváraní elementov a obrazoviek užívateľského rozhrania AREditora v Microsoft HoloLens 2 je potrebné vyriešiť dva na seba nadväzujúce problémy. Prvým z nich je už vyššie spomenutá komunikácia so serverom, kde server určuje nasledujúci stav aplikácie (otvoriť hlavné menu, otvoriť scénu apod.). Zatiaľ čo v mobilnom zariadení je pomerne jednoduché rozlíšiť či užívateľ je v stave vizualizácie prostredia alebo v hlavnom menu pri okuliaroch to môže byť problematickejšie. Ak je užívateľ v menu, vidí iba prvky rozhrania, avšak pri okuliaroch je aj menu umiestnené v prostredí zmiešanej reality.

Umiestnenie grafických prvkov v prostredí zmiešanej reality je však aj výhodou oproti mobilnej verzii, keďže užívateľ sa nemusí prepínať medzi dvoma prostrediami a je priamo schopný interagovať s týmito prvkami.

Návrh menu

Jednou z možností teda je vytvoriť viacero menu komponentov na základe toho, v akom stave (scéna, menu, projekt) sa užívateľ nachádza a využiť možnosť pripnutej obrazovky, ktorá užívateľa nasleduje pri pohybe. Táto možnosť by mohla do istej miery vyriešiť problém súvisiaci s rozlíšením stavu aplikácie avšak na druhej strane nevyužíva potenciál, ktorý zobrazenie AR v okuliaroch ponúka. Rozhodla som sa preto spojiť všetky elementy užívateľského rozhrania do jedného menu a zároveň využiť možnosť okuliarov zobrazovania prvkov na ruke (Obrázok 5.2). Užívateľ tak vyvolá a zobrazí menu, iba v prípade ak to potrebuje, pozeraním sa na svoju dlaň.

Návrh zobrazenia zoznamov

Ďalším elementom užívateľského rozhrania je zobrazenie zoznamov (scény, projekty, objekty, akcie). Zatiaľ čo v mobilnej verzii je prirodzenejšie zobrazovať zoznam vypísaním jednotlivých objektov, v okuliaroch môže byť táto možnosť obmedzujúca. V mojom riešení som preto navrhla zobrazenie pomocou 3D modelov a tak má užívateľ možnosť vidieť ukážku modelu/scény/projektu ešte pred jeho výberom (Obrázok 5.3).



Obr. 5.3: Zobrazenie zoznamu robotov ako 3D objekty.

V prípade akcií toto riešenie nie je úplne možné, keďže nejde o rôzne typy modelov. V tomto prípade som však potrebovala vyriešiť iný problém. Pri pridávaní akcií je potrebné najprv vybrať objekt, ktorý má akciu vykonať, a až následne vybrať jednu z jeho možných akcií. Mobilná verzia aplikácie ponúka riešenie v podobe hierarchického zoznamu, kde sa po výbere objektu vyrolujú jeho akcie. Obdobne je to možné využiť aj v mojom riešení v podobe výberu objektu ako 3D modelu a následne zobrazenia akcií. Druhou možnosťou je výber objektu, ku ktorému sa má akcia priradiť pomocou už vytvorených modelov v scéne, keďže nakoniec ten aj predstavuje objekt akcie. Vzhľadom k tomu, že zobrazenie modelu ako 3D ukážku som navrhla využiť už pri pridávaní objektov, môže byť táto možnosť s rôznym využitím na dvoch miestach pre užívateľa máť. Preto som sa rozhodla využiť druhú možnosť.

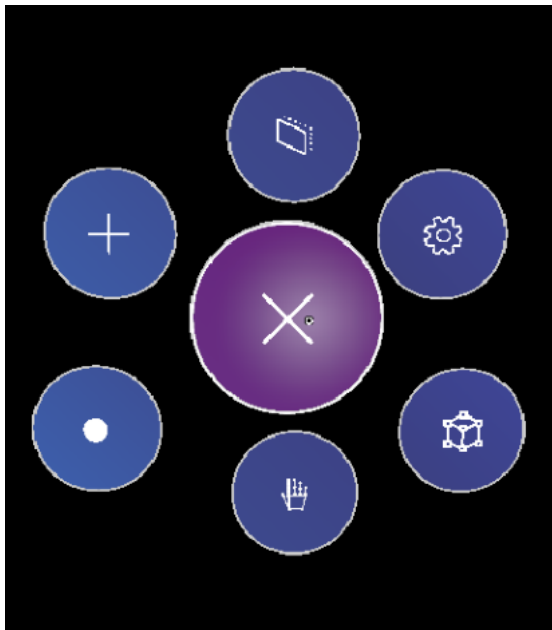
5.4 Interakcia s robotom v užívateľskom rozhraní AREditor pre Microsoft HoloLens 2

Hlavnou funkcionalitou aplikácie je interakcia s robotom, pod ktorú možno zaradiť jednak samotné manipulovanie s robotom ale aj pridávanie a manipulovanie s prvkami, ktoré sú súčasťou zadávania úloh robotovi. Okrem modelu robota či senzora sú modely ostatných objektov uložené v klientskej časti aplikácie.

Interakcia s objektmi je však oproti mobilnej verzii pomerne odlišná a do istej miery jednoduchšia. Zatiaľ čo je užívateľ v mobilnej verzii limitovaný plochou obrazovky, v okuliaroch je možné s objektmi interagovať priamo (pozri sekciu 3.2).

Návrh zobrazenia menu pre interakcie

V prvej fáze riešenia som navrhla možnosť priamej transformácie objektu, čo znamená že užívateľ nemusel explicitne voliť funkciu pre manipulovanie ale objekt v scéne mohol transformovať rovno. Pri hlbšej analýze som však zistila, že táto možnosť nie je vhodná, keďže



(a) Revolverové menu – návrh.



(b) Menu na ruke s otvoreným bočným menu pre pridávanie.

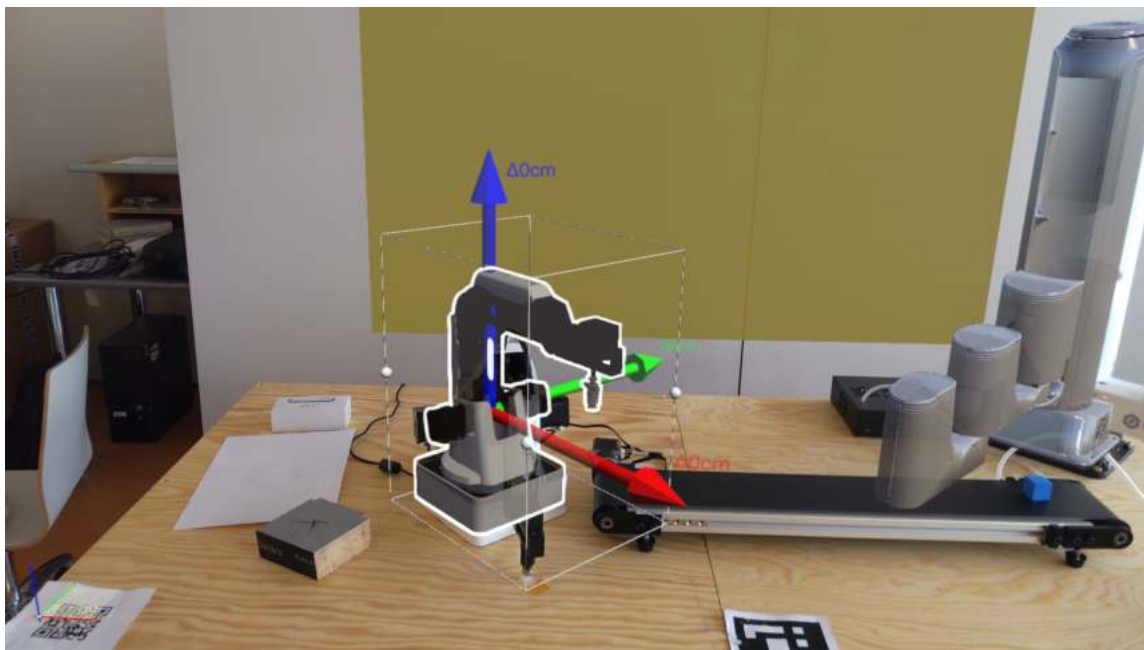
Obr. 5.4: Porovnanie návrhu revolverového menu s menu na ruke.

okrem transformácie objektu by mala aplikácia ponúkať aj inú interakciu s objektom. Či už ide o premenovanie alebo odstránenie objektu ale predovšetkým o pridávanie akcií a prepojení v projekte. Ďalšou možnosťou teda je využiť podobnú logiku ako pri mobilnej verzii. A to tak že, užívateľ najprv vyberie objekt s ktorým chce interagovať a následne funkciu, ktorú chce vykonať. K tomu som navrhla revolverové menu, ktoré sa zobrazí po kliknutí na objekt (Obrázok 5.4). Tento návrh však môže byť pre užívateľa zdĺhavý keďže je zakaždým potrebné spraviť dva kroky. Preto som sa nakoniec rozhodla navrhnúť riešenie v podobe režimov. To znamená, že užívateľ si najprv vyberie funkciu ktorú chce vykonať (režim) a následne môže v tomto režime vykonávať túto funkciu s objektami. Medzi základné režimy patrí: *Transformácia (manipulácia)*, *pridávanie akcií*, *pridávanie prepojení* a *odstránenie*.

Manipulácia

Užívateľ môže umiestniť a presúvať či otáčať objekt pomocou tzv. obalovacích kvádrov. Obalovacia kváder ohraničuje povrch objektu, ktorý interaguje s ovládacími prvkami. Napríklad užívateľ môže pomocou rúk alebo ukazovacieho kurzora označiť kváder a pomocou rôznych gest zväčšovať/zmenšovať, premiestňovať alebo otáčať objekt, ktorý obaluje (Obrázok 5.5). Užívateľ tak jednak nemusí otáčať so zariadením pre označenie objektu ale jednoducho použije prirodzený pohyb rúk keď chce niečo chytiť.

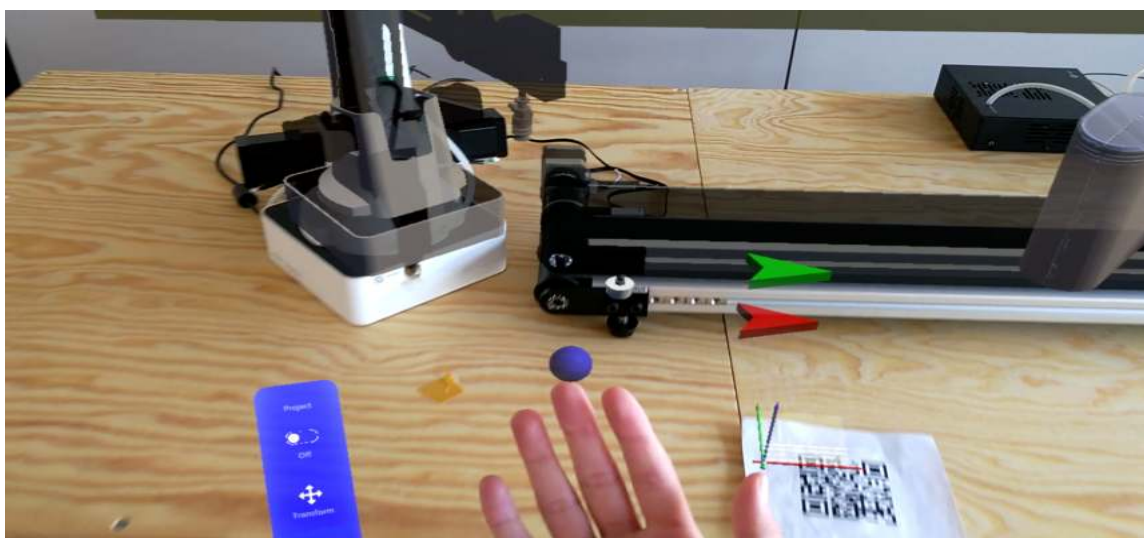
Pri manipulovaní s objektami však môže byť odozva priveľmi intenzívna. Preto som sa rozhodla doplniť možnosť umiestnenia pomocou kvádra aj o možnosť pohybu jednou súradnicovou osou. Inšpirovala som sa návrhom v mobilnej verzii kde je pri objekte vyobrazený model súradnicovej osi a podľa toho, ktorá os je označená sa objekt pohybuje.



Obr. 5.5: Manipulácia s robotom v scéne so zobrazením súradnicovej osi.

Pridávanie akcií a akčných bodov

Pre pridávanie akčných bodov na miesto vykonania akcie som využila možnosť ovládania pomocou gest. Užívateľ môže stlačením prstov pridať na miesto guľôčku, ktorá reprezentuje akčný bod. Pre vizualizáciu, toho že užívateľ umiestňuje akčný bod, som navrhla umiestniť túto guľôčku na prst užívateľa hneď po prepnutí sa do režimu pridávania akcií. Aby užívateľ nemusel explicitne vyberať režim pridania akcie, navrhla som spojiť tieto dve a teda po umiestnení akčného bodu je možné rovno priradiť objektu akciu.



Obr. 5.6: Zobrazenie guľôčky na ruke.

Kapitola 6

Implementácia

Táto kapitola popisuje implementáciu navrhnutého riešenia používateľského rozhrania v Microsoft HoloLens 2. Pred samotným začatím implementácie riešenia bolo potrebné spojazdniť existujúce riešenia na ktoré práca nadväzuje. Spojazdnenie AREditora je popísané v sekcii 6.1. Kapitola sa okrem toho venuje aj základným požiadavkám potrebným pri tvorbe aplikácii pre Microsoft HoloLens 2. Nasledujúce sekcie sú rozdelené podľa typu funkcionality, ktoré má rozhranie poskytovať. V sekcii 6.2 je popísaný spôsob zobrazenia scény a projektu. Implementácia interakčných prvkov, je popísaná v sekcii 6.3. Sekcia je rozdelená podľa jednotlivých režimov, v ktorých užívateľ môže s objektami scény interagovať. Okrem priamej interakcie s objektami, by mal užívateľ vedieť vytvárať nové a upravovať staré scény a projekty. V sekcii 6.4 je preto popísaná implementácia vizualizácie zoznamov scén a zoznamov objektov, ktoré je do scény možné pridávať. Nakoniec, v sekcii 6.5 je detailnejšie popísaná implementácia prepínania režimov či funkcií, keďže niektoré funkcie vyžadujú pred ukončením alebo začatím, vykonať určité akcie.

6.1 Spojazdnenie AREditora

Ako už bolo spomenuté v kapitole 4, táto práca nadväzuje na už existujúci projekt AREditor. Preto bolo potrebné pred začatím implementácie užívateľského rozhrania v Microsoft HoloLens 2 spojazdniť serverovú časť aplikácie a následne aj klientsku časť tabletovej verzie v Unity.

ARServer

ARServer funguje ako centrálny bod pre používateľské rozhrania a sprostredkúva komunikáciu s ostatnými službami. Rozhrania sú pripojené k ARServeru cez Websockets, čo umožňuje obojsmernú komunikáciu. ARServer uchováva stav systému, zatiaľ čo rozhrania s ním môžu manipulovať pomocou sady RPC¹ a prijímať upozornenia na zmeny. Predpokladá sa, že každé pracovisko prevádzkuje svoju vlastnú inštanciu ARServera, a preto server udržiava iba jednu reláciu pre všetkých používateľov: ak jeden používateľ otvorí projekt, rovnaký projekt sa zobrazí aj ostatným pripojeným používateľom (Obrázok 6.1). Na podporu efektívnej a bezpečnej spolupráce medzi užívateľmi je tu uzamykací mechanizmus, ktorý bráni viacerým užívateľom manipulovať s rovnakým prvkom (napr. ovládať robota).

¹Remote procedure call – vzdialený volanie procedúr

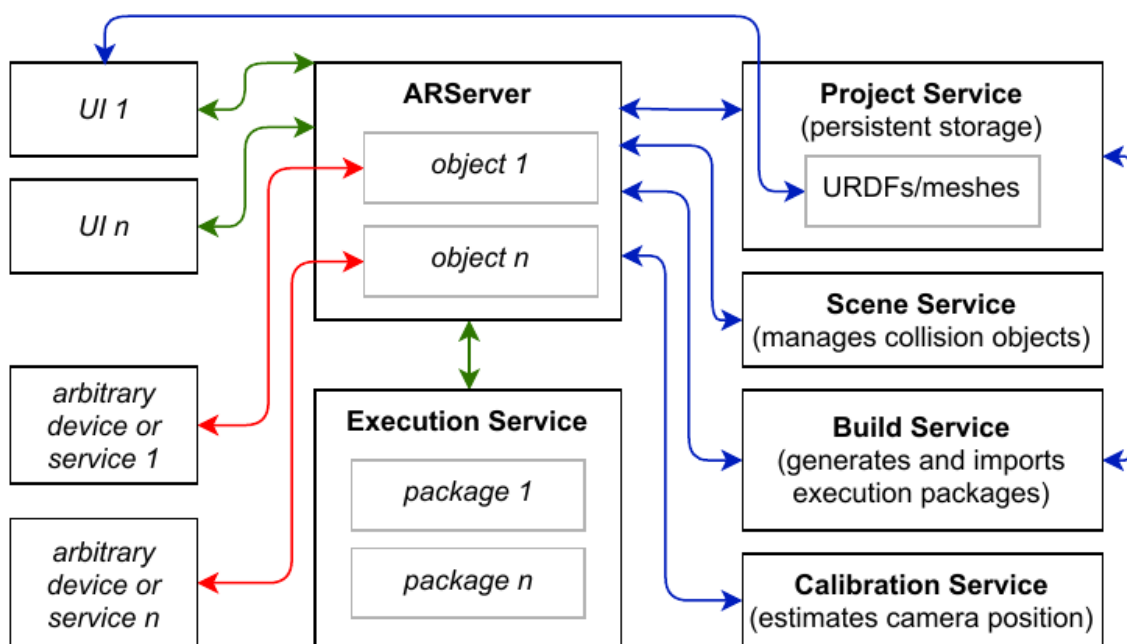
ARServer tiež slúži ako proxy medzi zdrojovým kódom v jazyku Python a prostredím AR. Analyzuje kód typov objektov, aby extrahoval dostupné akcie a ich parametre a vytvára JSON metadáta, ktoré sú dostupné pre používateľské rozhrania². Github repozitár je možné naklonovať a následne podľa inštrukcií spojzdať docker kontajner.

AREditor

AREditor v tablete je zatiaľ jediné používateľské rozhranie pripojené k ARServeru a je implementovaný v Unity. Keďže táto práca práve rozširuje typ používateľského rozhrania pripojeného k tomuto serveru, bolo ideálne využiť tento projekt.

Hlavnou zložkou štruktúry Unity projektu je zložka *Assets*. V zložke *Assets* je zaužívané rozdeľovať projekt do štyroch častí. Skripty, kde sú uložené zdrojové kódy vývojára. *Prefabs*, ktoré slúžia na uchovávanie znovupoužiteľných vývojárom definovaných objektov. A nakoniec modely a samotná scéna. V tomto prípade ide však o niečo komplexnejší projekt, ktorý obsahuje zložku *BASE* so základnými nastaveniami projektu, ďalej zložku *TABLTE*, ktorá obsahuje vyššie spomenutú štruktúru pre toto užívateľské rozhranie. Okrem toho sa v projekte nachádza aj zložka *Submodules*, ktorá slúži pre uchovanie využitých assetov tretích strán. Keďže ide o projekty z tretích strán táto zložka nie je priamo súčasťou projektu a je potrebné si tieto projekty importovať. Rovnako ako pri serveri je možné si projekt naklonovať z git repozitára, ktorý obsahuje aj potrebné inštrukcie k spojzdeniu a zoznam projektov k importovaniu.

O komunikáciu so serverom sa stará trieda *WebsocketManager*, ktorá odosiela a prijíma požiadavky.



Obr. 6.1: Architektúra²prepojenia rozhrania AREditor s ARServer.

²<https://github.com/robofit/arcor2/wiki/Architecture>

Nastavenie projektu pre HoloLens

V sekcii 3.3, bol stručne popísaný nástroj, resp. projekt Mixed Reality Toolkit, ktorý poskytuje súbor komponentov a funkcií pre zjednodušenie a urýchlenie vývoja aplikácií s rozšírenou realitou. MRTK, tvorí aj základnú časť vstupného systému pre interakciu s objektami tejto aplikácie. Okrem toho je využívaný aj v iných častiach aplikácie, ktoré sú popísane nižšie.

Pre komunikáciu so serverom bolo pôvodne plánované využiť rovnakú triedu ako v tabletovej verzii. Avšak vzhľadom k vysokej previazanosti zdrojových kódov s týmto rozhraním to nebolo možné. Refaktor tej časti zdrojových kódov, ktoré by bolo možné využiť aj pri okuliaroch by však mohol byť zdĺhavý a v konečnom dôsledku to nie je ani predmetom práce. Preto som sa rozhodla prevažne prevziať tieto časti zdrojových kódov tak, že v názve každého skriptu, ktorý do istej miery kopíruje rozhranie pre tablet je veľké písmeno „H“ (HoloLens).

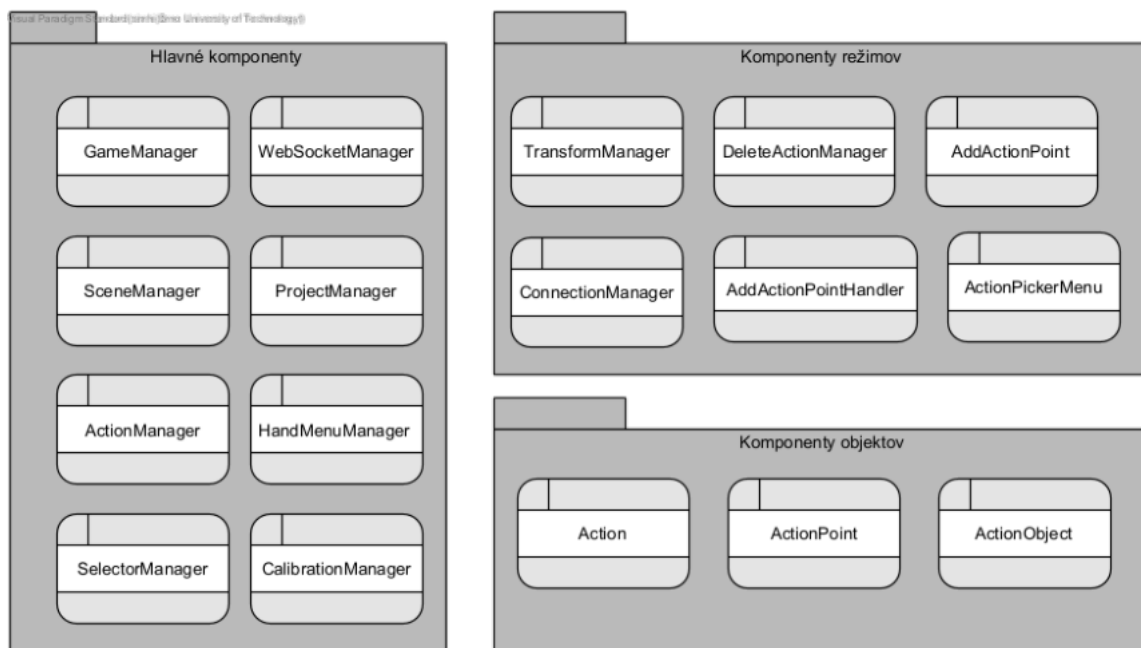
Komponenty aplikácie

K hlavným komponentom aplikácie patria skripty, ktoré manažujú tok programu: *WebSocketManager*, *SceneManager*, *ProjectManager*, *ActionManager*, *GameManager*, *HandMenuManager*, *SelectorManager*.

- **GameManager** – Je hlavným mozgom aplikácie. Manažuje stav aplikácie a stav pripojenia. Spúšťa proces otvorenia/zatvorenia scény a projektu
- **WebSocketManager** – Komunikuje so serverom. Zachytáva odpovede od serveru a posíla naň požiadavky s konvertovanými údajmi v json formáte.
- **ActionManager** – Služi na uchovávanie metadát o všetkých objektoch (robot, dopravníkový pás apod.), ktoré je možné do scény pridávať
- **SceneManager** – Spravuje scénu. Pridáva, upravuje a odstraňuje objekty zo scény
- **ProjectManager** – Spravuje projekt. Pridáva, upravuje a odstraňuje akcie, ktoré vytvárajú konkrétny program pre robotické pracovisko.
- **SelectorManager** – Spravuje akcie režimu. Prepína typ režimu podľa výberu používateľom a následne spúšťa akcie spojené s týmto režimom.
- **HandMenuManager** – Manažuje pripínanie funkcií (pridanie objektu, zoznam scén/projektu a pod.), vrátane režimov. Jeho hlavnou úlohou je vynulovať nastavenia spojené s predchádzajúcou funkciou alebo režimom, ktoré by mohli ovplyvniť korektné správanie sa nasledujúcej funkcie.

Komponenty *SceneManager* a *ProjectManager* vytvárajú 3D objekty. Každý objekt je tvorený komponentom rozšírenia skriptu *InteractiveObject*. Medzi hlavné patrí *ActionObject*, *Action*, *ActionPoint*. Tie sú ďalej rozšírené už podľa konkrétneho typu objektu robota, akcie či akčného bodu.

Medzi hlavné komponenty režimov, ktoré spúšťa *SelectorManager* patria *TransformManager*, *DeleteActionManager*, *AddActionPoint*, *ConnectionManager*, *AddActionPointHandler* a *ActionPickerMenu* (Obrázok 6.2).



Obr. 6.2: Základná architektúra komponentov pre rozhranie Microsoft HoloLens 2.

Kalibrácia

Tak ako pri tabletovej verzii projektu, aj tu je potrebné definovať začiatok súradnicového systému. K tomu som využila projekt tretích strán QRTracking³, ktorý umožňuje naskenovanie QR kódu a následne zobrazenie objektu so súradnicovou osou na jeho pozíciu (Obrázok 6.3). Po kalibrácii je zobrazená scéna relatívne k tomuto objektu.

6.2 Zobrazenie scény a projektu

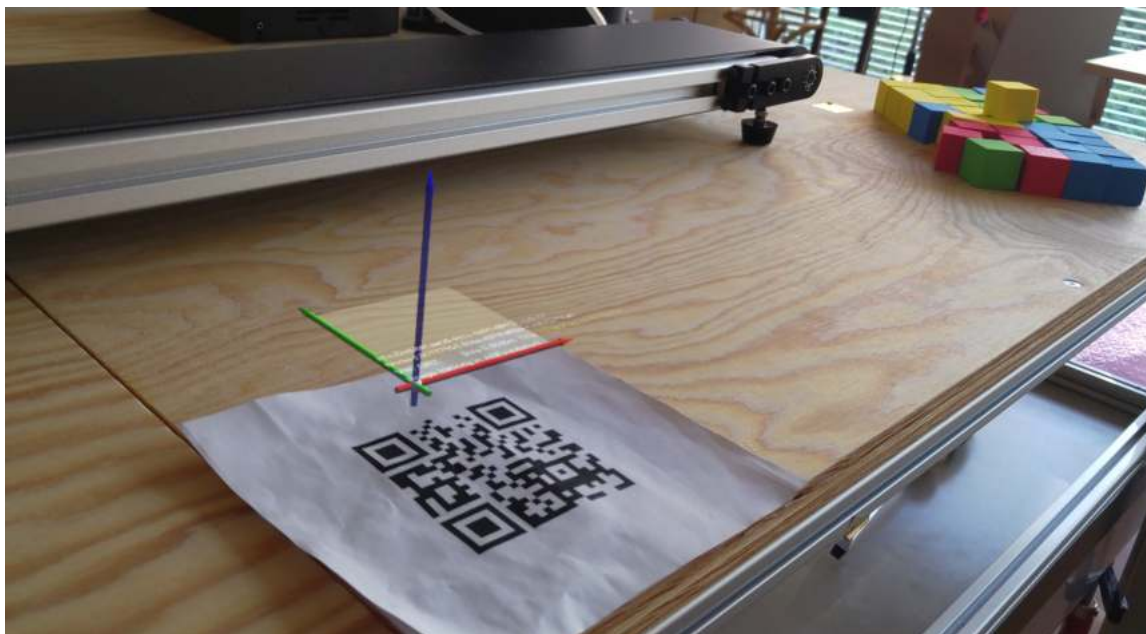
K zobrazeniu scény je potrebné vykonať viacero na seba nadväzujúcich krokov. V prvom rade, sa pošle požiadavka na server s RPC⁴ *OpenProject/OpenScene*, ktoré rozhraniu pošlú informácie o všetkých objektoch k danej scéne. Každý objekt scény má vytvorený svoj vlastný *Prefab*, ktorý využíva komponenty potrebné pre prácu s ním. Napríklad robot má nastavenia jeho efektorov.

Objekty sa pridávajú do scény podľa pozícií získanej zo servera. V prípade robota je však potrebné model načítať osobitne, keďže ten sa priamo v projekte nenachádza. Modely robotov sú uložené v urdf a ako už bolo spomenuté v podkapitole 4.2 v Unity je komplikovanejšie načítanie modelu za behu programu a preto bola k tomu využitá knižnica tretích strán TriLib. MRTK má však aj vlastné spracovanie modelov v gLTF formáte a preto ich nie je možné používať naraz. Pri hľadaní možnosti riešenia som zistila že v MRTK práve kvôli tomuto prípadu pridali možnosť pre vypnutie funkcionality spracovania týchto modelov.

Po zobrazení scény sa pomocou pointeru (viac v sekcii 3.2) ukázaním na niektorý z objektov, daný objekt zvýrazní a zobrazí sa jeho názov. K zvýrazneniu objektu bola použitá

³<https://github.com/yl-msft/QRTracking>

⁴Remote procedure call – vzdialený volanie procedúr



Obr. 6.3: Vizualizácia kalibračného bodu v Microsoft Hololens 2 (pozn. virtuálne prvky sú po odfotohrovaní posunuté).

knížnica tretích strán *Outliner*⁵, ktorá podľa tvaru mriežky vykreslí na povrchu objektu jeho obrys.

6.3 Interakcia s objektami

O interakciu s objektami sa stará jednoduché menu „na ruke“, ktoré sa zobrazí vedľa ruky pri pozeraní sa na dlaň ruky. K tomu zobrazeniu som využila skript vytvorený v MRTK projekte a jednotlivé komponenty (tlačidlá, materiál, rozloženie) upravila podľa potrieb.

Pri interakcii som využila model režimov (viac v sekcii 5.4) a teda po kliknutí na jednu z funkcií, môže užívateľ aplikovať túto funkciu na všetky objekty, ktoré môžu funkciu spracovať. Medzi ne patrí *Transformácia*, *Odstránenie*, *vytvorenie akcií*, *pridanie pripojenia*.

- **Transformácia** – Používateľ môže s objektami manipulovať z blízka ale aj z diaľky pomocou pointera. K upresneniu pozície bola implementovaná aj súradnicová os, vďaka ktorej je možné manipulovať s každou osou zvlášť.
- **Pridávanie akcií** – Používateľ pridáva k akčnému bodu priamo akciu, ktorú má vykonať. Objekt, ku ktorému má byť pridaná akcia si používateľ vyberá kliknutím na jeho model v scéne a zoznam jeho akcií sa zobrazí na mieste, kde má byť akcia umiestnená. Aby mohol používateľ jednoducho umiestniť akčný bod na miesto kde má byť akcia vykonaná, bola implementovaná funkcia, ktorá zachytáva pozíciu ruky a po stlačení prstov umiestni objekt na miesto ich stlačenia.
- **Prepojenia** – Používateľ k určaniu toku vytvorenej úlohy vytvára prepojenia. Najprv si vyberie akciu, z ktorej má prepojenie vychádzať a následnej akciu kam má vchádzať.

⁵<https://assetstore.unity.com/packages/vfx/shaders/fullscreen-camera-effects/outline-effect-78608description>



(a) Zobrazenie robota po označení.



(b) Zobrazenie akcie po označení.

Obr. 6.4: Príklad označenia objektu v scéne s jeho zvýraznením a zobrazením názvu.

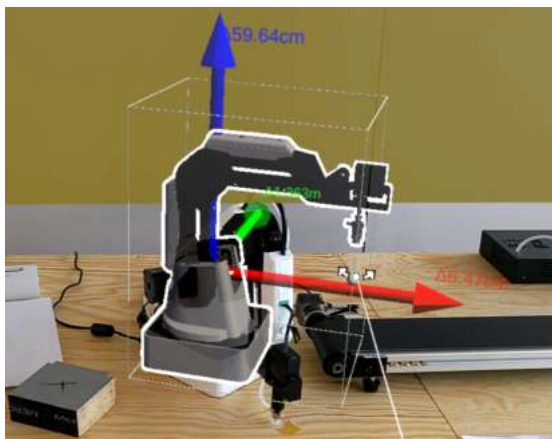
Označenie objektu na ktorý má byť funkcia aplikovaná, sa vykoná ukázaním na objekt a následným gestom (Obrázok 6.4). Pre jednoduchšiu interakciu s menšími objektami ako sú akcie či akčné body som sa rozhodla využiť obalovacie kvádre, ktoré sú jednoduchšie zaregistrovateľné. Oproti tomu zase s robotom, ktorý má nepravidelné tvary je možné pri výbere interagovať len v jeho oblasti. Je tomu tak aj preto, že by nebolo možné manipulovať s objektami, ktoré by boli umiestnené v obsahu jeho obalovacieho kvádra.

Pri implementácii som sa zamerala len na niektoré typy režimov. V tabletovej verzii je možné objekty a akcie rôzne parametrizovať (nastavenie efektorov robota, rýchlosť apod.) alebo premenovať. Vzhľadom na komplexnosť úlohy som sa rozhodla v tejto fáze implementovať len základne funkcie. Okrem vyššie spomenutých bola implementovaná aj možnosť vytvorenia duplicity, ktorá však z hľadiska vyhodnotenia riešenia nemala žiaden vplyv. Manažment prepínania akcií má na starosti skript *SelectorManager*, ktorý po výbere režimu priradí *UnityAction* objektu metódu ktorá ošetruje vybraný režim. Následne keď sa vyberie objekt s ktorým má režim pracovať, vyvolá sa priradená metóda.

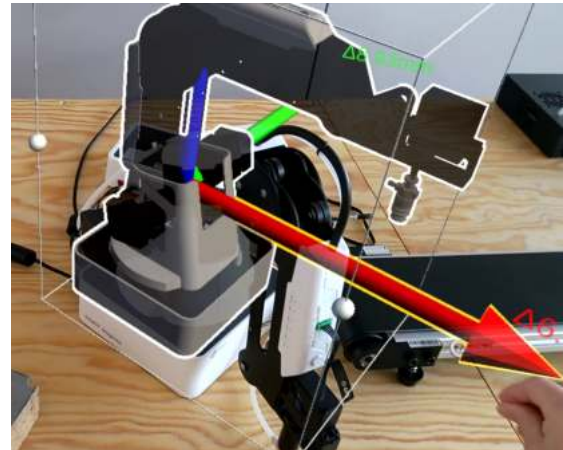
Transformácia

K manipulácií objektov (premiestnenie, zväčšenie, otočenie) boli využité komponenty projektu MRTK. Tie obsahujú viacero kombinácií prednastavení akým spôsobom je možné s objektom manipulovať. Na hranách obalovacej kocky sa nachádzajú body pre otáčanie objektu a na rohu body, pre zmenu veľkosti objektu (Obrázok 6.5). Vývojár si môže určiť obmedzenia na jednotlivé manipulačné prvky.

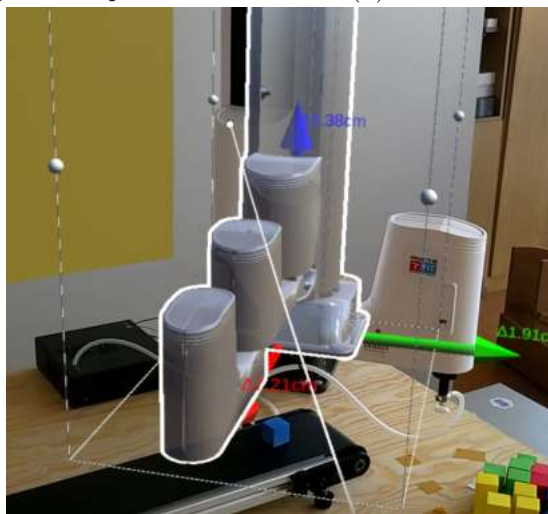
O manažovanie akcií spojené s manipuláciou sa stará *TransformManager*. K tomu bol vytvorený *GameObject TransformPrefab*, ktorý má priradené práve komponenty súvisiace s manipuláciou. Pri výbere objektu pre manipuláciu sa pôvodný model skryje a v *TransformPrefab* sa vytvorí jeho kópia. Vzhľadom k tomu, že každý objekt má iné vlastnosti pre manipuláciu (robot sa môže otáčať len podľa z-osi a nemôže sa zväčšovať, akčné body a akcie sa môžu len premiestňovať, kolízne objekty sú flexibilné) sa po výbere nastaví obmedzenia v *TransformPrefab* podľa toho o aký objekt sa jedná. Pre nastavenie pozície a veľkosti jeho obalovacieho kvádra som využila veľkosti obalovacích kvádrov, ktoré sa nachádzajú aj pri originálnom objekte. V prípade robota bolo potrebné najprv tento obalovací kváder vytvoriť. Keďže robot sa skladá z viacerých menších mriežok, bolo potrebné ich zapuzdrenie z



(a) Otáčanie robota pomocou pointera.



(b) Posúvanie robota po ose x držaním.



(c) Posúvanie robota pomocou pointera

Obr. 6.5: Možnosti manipulácie s 3D objektami cez obalovaciu kocku (a), (c) a držaním objektu (b).

čoho som následne využila veľkosť ohraničenia a stredný bod pre vytvorenie obalovacieho kvádra.

Posúvanie s objektom týmto spôsobom môže byť niekedy až príliš senzitívne, preto som sa rozhodla doplniť možnosť posúvania objektu podľa určitej osi. K tomu som využila model súradnicovej osi, umiestnený v strede objektu a pre každú os pridala obmedzenia posunu podľa toho o akú os ide. Napríklad, šípka predstavujúca osu x sa môže pohybovať iba po osi x. Užívateľ si tak môže spresniť pozíciu, na ktorú chce objekt umiestniť.

Aj v prípade manipulácie s objektom ako celok, ale aj v prípade posúvania podľa osi, môže užívateľ manipulovať s objektom z diaľky alebo z blízka (viac v sekcii 3.2).

Pridanie akcie

Režim pridávania akcií je spojenie dvoch funkcií z tabletovej verzii. Pridanie akčného bodu a pridanie akcie k objektu. Režim pozostáva z troch krokov - Kde, Kto a Čo. O správu akcií spojených režimom sa stará *AddActionPoint*.

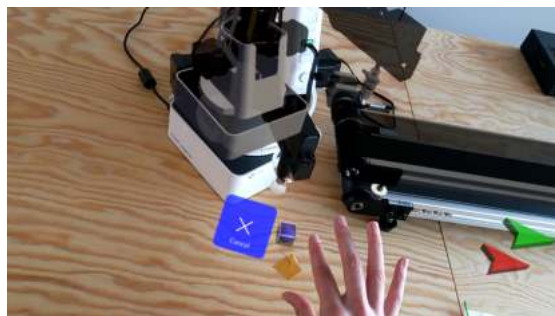
Krok **Kde** znamená že užívateľ vyberie miesto kde sa má akcia vykonať. V podstate ide o vytvorenie akčného bodu. Niektoré akcie nepotrebujú mať definované miesto vykonania (napr. spustenie) a preto nezáleží kde bude akčný bod umiestnený.

Keďže je možné akčný bod vytvoriť buď ako potomka nejakého objektu alebo ako globálny bod, je potrebné zachytiť bod zamerania pozorovacieho oka (pointer). K tomu bol využitý interface MRTK projektu *IMixedRealityPointerHandler*. Z *IMixedRealityPointerHandler* bola využitá metóda na zachytenie udalosti kliknutia (stlačenie prstov). Tá vo svojom parametre uchováva aj informáciu o tom aký objekt bol pointerom zachytený pri kliknutí. Pri spustení aplikácie v editore Unity bolo kliknutie na objekty v scéne považované ako kliknutie do prázdna a teda *null*. To som využila ako podmienku pre vytvorenie globálneho akčného bodu. Avšak, pri nasadení aplikácie do okuliarov je ako objekt, považovaný aj reálny objekt do určitej vzdialenosti. A to vďaka tomu že v nastaveniach je zapnuté priestorové mapovanie. Preto bolo potrebné nie len registrovať kedy je objekt *null* ale aj kedy ide o reálny objekt.

Na vizualizovanie toho, že užívateľ vytvára akčný bod, som implementovala na prst ruky užívateľa model akčného bodu. To bolo možné vďaka funkcionalite sledovania rúk v okuliaroch a využitiu metód MRTK na prácu z ich parametrami (klíby a pod.). Táto metóda bola využitá aj na umiestnenie akčného bodu na miesto „tapnutia“. V prípade aj pri tape označená akcia, nevytvorí sa nový akčný bod ale je vybraný ten, ktorý je priradený k vybranej akcii.



(a) Používateľ stlačí prsty na mieste, kde sa má akčný bod vytvoriť.



(b) Vytvorenie akčného bodu na mieste stlačenia.



(c) Výber objektu, ktorý má akciu vykonať.



(d) Zobrazenie akcií, ktoré je možné s objektom vykonať

Obr. 6.6: Vizualizovanie jednotlivých krokov pri pridávaní akcií.

Po vytvorení (alebo výbere) akčného bodu si užívateľ vyberá **Kto** vykoná na mieste akciu. Teda vyberá si objekt (robot, kolízny objekt). Systém sa implicitne prepne do režimu



(a) Výber výstupnej akcie pre pripojenie. (b) Výber vstupnej akcie pre pripojenie. (c) Pripojenie akcie *pick* a *place*.

Obr. 6.7: Vizualizácia vytvárania prepojenia *pick* a *place*.

vytvorenia akcie, ktorý je súčasťou hlavného režimu a je možné ho spustiť iba po vykonaní predchádzajúceho kroku (Obrázok 6.6).

Keď užívateľ vyberie kliknutím objekt, zobrazí sa na mieste akčného bodu zoznam akcií, ktoré je možné objektu priradiť. Akcie konkrétneho objektu sa získavajú zo servera pomocou RPC⁶ *GetActions*. Užívateľ si tak môže z akcií vybrať **Čo** má objekt vykonať. Každá akcia je parametrizovaná avšak pre uľahčenie implementácie som sa rozhodla použiť predvolené nastavenia parametrov akcií. Pre rýchlejšiu interakciu s akciami som k objektu, ktorý je rodičom akcií, pridala komponent *Billboard* pre zabezpečenia otáčania tlačidiel predstavujúce jednotlivé akcie podľa pohľadu užívateľa.

Po vytvorení akcie (alebo zrušenia tlačidlo Cancel) sa systém opäť prepne do hlavného režimu pridávania akcie.

Pridanie prepojenia

Podobne ako pri akciách, aj pri pridávaní prepojenia má systém jeden hlavný režim a jeden vnútorný. Užívateľ po výbere režimu pridať prepojenie najprv vyberie akciu, ktorá predstavuje výstup odkiaľ pokračuje ďalšia akcia. Následne sa vytvorí prepojenie medzi akciou a používateľovým ukazovák. Vďaka tomu vie, že k prepojeniu bol priradený výstup a teraz nasleduje vstup. Systém sa interne prepne do režimu pridania vstupného bodu a dočasne sa uloží informácia a výstupe. Po kliknutí na vstupnú akciu sa koniec prepojenia na ruke užívateľa presunie na vybranú akciu a vytvorí sa prepojenie medzi týmito dvoma akciami (Obrázok 6.7). Systém sa opäť prepne do hlavného režimu pridania prepojenia. Ak užívateľ klikne niekde „vedľa“ prepojenie sa preruší a tiež sa vráti do hlavného režimu. Problém s indikovaním reálnych objektov v okuliaroch bol vyriešený rovnako ako pri akciách. Na vytvorenie prepojenia som využila *LineRenderer* a k tomu komponenty vytvorené aj pre reprezentovanie prepojenia v tablete. Tieto komponenty bolo potrebné využiť hlavne kvôli tomu, aby bolo možné pri manipulácii s akciami možné manipulovať aj s prepojeniami.

Odstránenie objektu

V prípade odstránenia objektu bolo potrebné vyriešiť situáciu, kedy užívateľ chce odstrániť prepojenie. *LineRenderer* nie je 3D objekt, a preto nie je možné prepojenie označiť. Vytvorenie krivky ako 3D model je síce možné, ale je to komplexnejší problém, ktorý som sa rozhodla vzhľadom k implementáciám iných, dôležitejších častí vynechať. Preto som pre odstránenie prepojenia využila akciu z ktorej vychádza. Ak užívateľ klikne pri odstránení na akciu, ktorá má existujúci výstup, zobrazí sa mu ponuka dvoch tlačidiel označujúcich či

⁶Remote procedure call – vzdialený volanie procedúr



Obr. 6.8: Vizualizácia zobrazenia menu s možnosťou odstránenia akcie alebo prepojenia.

chce odstrániť iba prepojenie alebo aj celú akciu (Obrázok 6.8). Prepojenie je možné odstrániť aj pri samotnom pridávaní prepojenia a to tak, že ak užívateľ chce priradiť výstup prepojenia k akcii z ktorej už prepojenie vychádza, to prepojenie zanikne.

V prípade ostatných objektov sa užívateľovi zobrazí informačné okno či chce naozaj označený objekt odstrániť.

6.4 Vizualizácia zoznamov

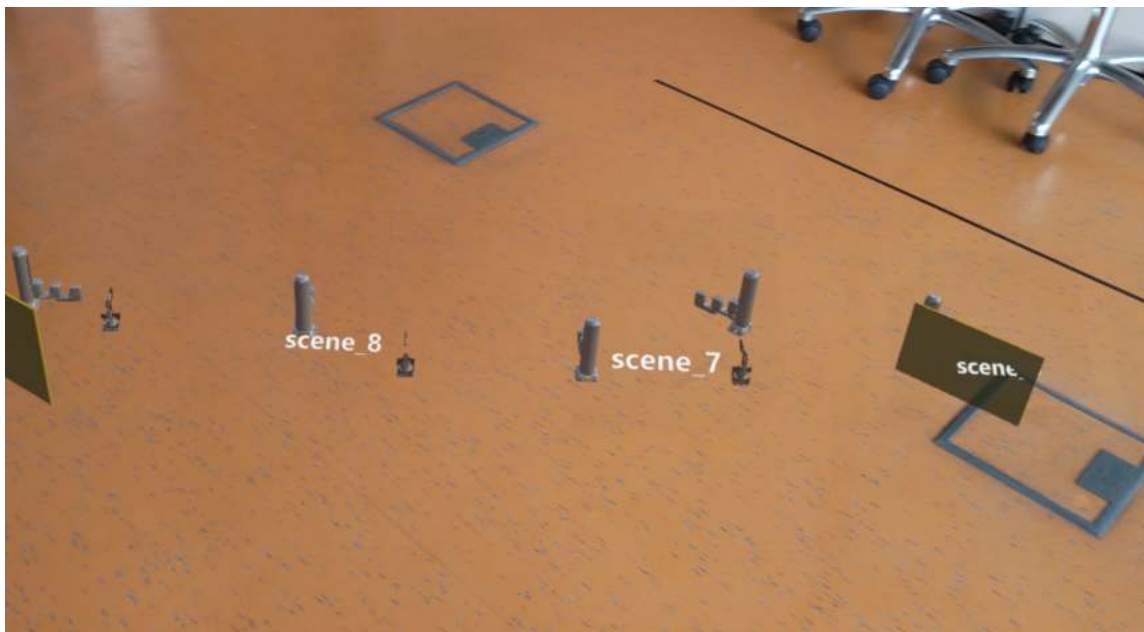
Okrem interakcie s objektami som implementovala aj funkciu pridávania objektu či zobrazenie scén. Ako bolo opísané v sekcii 5.3 pre vizualizáciu objektov som navrhla zobrazenie položiek zoznamov pomocou 3D modelov. Pri oboch prípadoch zobrazenia som ako obalovací objekt využila prefab vytvorený v projekte MRTK v tvare transparentnej kocky, ku ktorej sme priradili interakciu pre výber položky.

Pridanie objektu

Zoznam modelov robotov sa načíta iba raz po pripojení sa k serveru. Zoznam kolíznych objektov je vytvorený staticky v projekte, keďže ide v podstate len o základné 3D objekty, ktoré Unity ponúka (kocka, guľa, valec).

Objekt je umiestnený do stredu obalovacej kocky a jeho veľkosť prispôbená k jej veľkosti. Pri vytvorení nového objektu je potrebné pri požiadavke na server poslať aj parametre vytváraného objektu. Tie sme, ako pri akciách, nastavili na predvolené hodnoty.

Po výbere objektu sa vytvorí jeho inštancia pred užívateľom.



Obr. 6.9: Vizualizácia zobrazenia zoznamu vytvorených scén.

Výber scény a projektu

Pre implementáciu návrhu zobrazenia zoznamu scén a projektov bolo potrebné na servery vytvoriť nové RPC⁷ *GetScene* a *GetProject*, ktoré vrátili informácie o scéne/projekte podľa identifikátora. K týmto informáciám bolo možné prísť iba pri ich otvorení a teda nebolo možné vykresliť objekty všetkých scén a projektov naraz.

Keďže pri zobrazení scény ako položky zoznamu ide vlastne o klasické zobrazenie popísané v sekcii 6.2 ale bez interakcie s objektami, vytvorila som kópie prefabov bez komponentov, ktoré definovali správanie sa týchto objektov. Následne som opäť využila princíp zapuzdrenia ako pri robotovi a podľa toho nastavila veľkosť a stred obalovacej kocky. Scénu som nakoniec zmenšila do prijateľnej veľkosti zobrazenia zoznamu.

Pre skrolovanie kolekcie 3D objektov je v MRTK vytvorený komponent *ScrollingObjectCollection*, ktorý poskytuje základné nastavenia pre typ rolovania. V tomto prípade som sa rozhodla použiť horizontálne rolovanie s jedným riadkom (Obrázok 6.9).

6.5 Prepínanie režimov

Samotné manažovanie režimov, ktoré slúžia na prepínanie akcií pri interakcii s objektami má na starosti *SelectorManager*. Okrem toho je však pri ukončení niektorých z nich vynulovať nastavenia. K týmto nastaveniam patrí napríklad uzamknutie objektu, aby v rovnakom čase nemohli s objektom manipulovať viacerí užívatelia. K tomu som vytvorila skript, ktorý ošetruje pri prepnutí na inú funkciu akcie spojené s ukončením predchádzajúcej funkcie. Napríklad pri *Transform* je to uloženie poslednej zmeny objektu a jeho odomknutie. Ak nie je potrebné čakať na odpoveď zo servera sú akcie spojené s nasledujúcou funkciou vykonané hneď. Inak, sú tieto akcie zaradené do zoznamu čakajúcich akcií na zavolanie po odpovedi zo servera. Príkladom je samotné odomknutie objektu, kedy v prípade ak by

⁷Remote procedure call – vzdialený volanie procedúr

objekt nebol správne odomknutý mohlo by nastať k chybe. Ďalším vhodným príkladom je otvorenie druhej scény alebo projektu z inej scény. Kedy je potrebné najprv aktuálne otvorenú scénu uložiť a zatvoriť. A až potom ako príde informácia o uzatvorení scény je možné otvoriť druhú, zvolenú scénu. Oproti tabletovej verzii je to uľahčenie pre užívateľa kedy nepotrebuje sám najprv scénu zatvoriť a až následne si vybrať inú.

Kapitola 7

Experimenty

Hlavným cieľom práce je zistiť použiteľnosť náhlavného displeja Microsoft HoloLens 2 pre rozšírenú realitu pri programovaní robotického pracoviska. K tomu bol vytvorený experiment pozostávajúci s jednoduchej úlohy. V tejto kapitole je popísaný priebeh a vyhodnotenie experimentu. Detailný popis experimentu je opísaný v sekcii 7.1. V sekcii 7.2 je opísaný priebeh experimentu. Vyhodnotenie výsledkov experimentu je v sekcii 7.3. V závere kapitoly, v sekcii 7.4, je spomenutých niekoľko nápadov a postrehov do budúcnosti od používateľov a mojich vlastných.

7.1 Popis experimentu

Testovanie bolo zamerané na použiteľnosť náhlavného zariadenia Microsoft HoloLens 2 pre rozšírenú realitu pri programovaní robotického pracoviska. Pre porovnanie bolo využité rozhranie AREditor pre tablet, ktoré je navrhnuté na rovnakom základe a preto by funkčnosť nemala vplývať na konečné výsledky experimentu. Rovnako je tak pre účastníka experimentu (ďalej používateľ) jednoduchšie zapamätať si jeden typ aplikácie ako dva. Používatelia teda v rôznom poradí otestovali navrhnuté riešenie v Microsoft HoloLens 2 a rozhranie v tabletovej verzii. Pre tabletové riešenie bolo vybrané zariadenie Samsung Galaxy Tab Active Pro 10.

Experiment prebiehal na Fakulte informačných technológií v Brne, ktorá poskytla priestory, v ktorých sa nachádzajú fyzické modely robotov.

Pre testovanie boli navrhnuté dve jednoduché úlohy, ktoré na seba nadväzovali a komplexnejší dotazník, ktorý sa skladal z viacerých častí.

Prvá úloha – vytvorenie robotického pracoviska (scény)

V prvej úlohe mal používateľ vytvoriť jednoduchú scénu. Scéna pozostávala z dvoch robotov (Dobot Magician, Dobot M1), dopravníkového pásu a kolíznej kocky, ktorá mala predstavovať zadnú stenu. Používateľ mal do prázdnej scény pridať tieto objekty a umiestniť ich na pozíciu reálnych objektov. (Obrázok 7.1). Úloha bola prevažne zameraná na manipuláciu s objektami. Pracoval s funkciami transformačného režimu, pridanie robota a pridanie objektu v nasledujúcich krokoch (Microsoft HoloLens 2):

1. Vyber z menu možnosť pridať robota, dopravníkový pás alebo objekt.
2. Zo zoznamu vyber robota alebo objekt, ktorý ešte nie je v scéne.



Obr. 7.1: Robotické pracovisko v ktorom účastníci experimentu vykonávali zadanú úlohu.

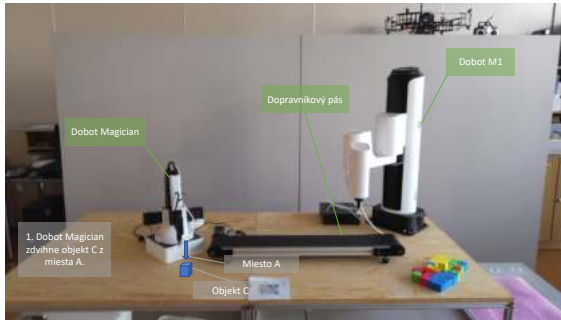
3. Vyber z menu režim *Transform*.
4. Vyber v scéne objekt, s ktorým chceš manipulovať.
5. Presuň objekt na určené miesto alebo otoč.
6. Opakuj od kroku 2 alebo 4, kým všetky objekty v scéne nie sú správne umiestnené.

V tablete sa kroky 2 a 3 vykonali opačne. Používateľ najprv vyberie objekt a až potom z menu možnosť *Transform*. Je to tak preto, že interakcia s objektami v tabletovej verzii je implementovaná tak, že najskôr je potrebné vybrať objekt a až potom funkciu, ktorú má vykonať.

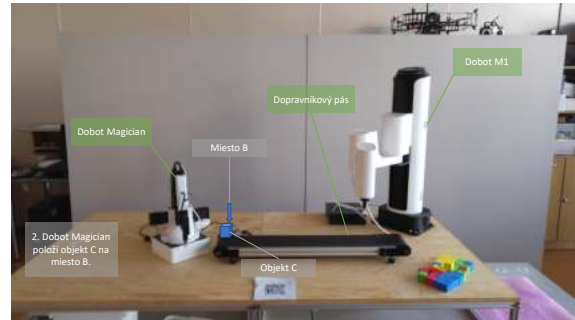
Druhá úloha – vytvorenie úlohy pre pracovisko

V druhej úlohe mal používateľ k jeho vytvorenej scéne vytvoriť projekt a teda akcie, ktoré majú objekty v scéne vykonať. Používateľ mal v tejto časti experimentu pracovať s akciami a prepojeniami. Ako demonštratívny príklad bola vybraná úloha *pick and place* (zdvihni a polož), kde roboty spolu s dopravníkovým pásom navzájom spolupracovali pri premiestnení objektu C (Obrázok 7.2). Úloha pozostávala z týchto piatich akcií:

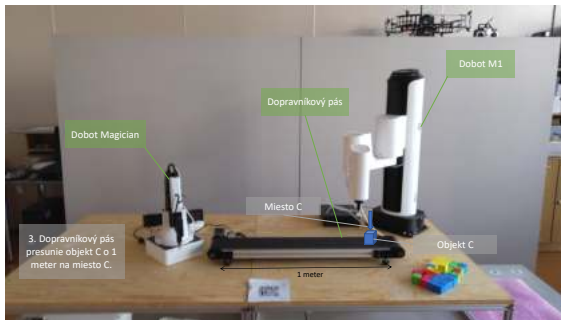
1. Dobot Magician zdvihne objekt C z miesta A.
2. Dobot Magician položí objekt C na miesto B.
3. Dopravníkový pás presunie objekt C o 1 meter na miesto C.
4. Dobot M1 zdvihne objekt C z miesta C.
5. Dobot M1 položí objekt C na miesto D.



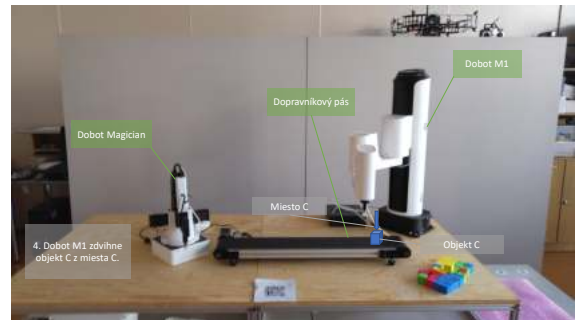
(a) Dobot Magician zdvihne objekt C z miesta A.



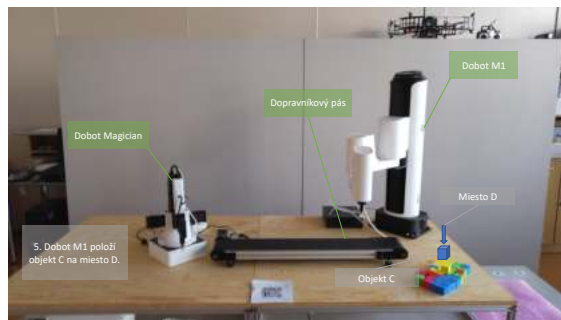
(b) Dobot Magician položí objekt C na miesto B.



(c) Dopravníkový pás presunie objekt C o 1 meter na miesto C.

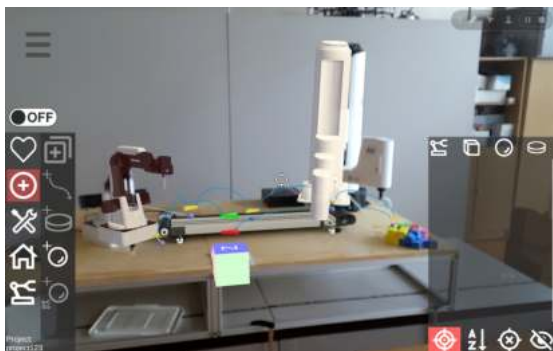


(d) Dobot M1 zdvihne objekt C z miesta C.



(e) Dobot M1 položí objekt C na miesto D.

Obr. 7.2: Jednotlivé kroky úlohy pre vytvorenie programu.



(a) Vykonaná druhá úloha v tablete.



(b) Vykonaná druhá úloha v Microsoft HoloLens 2.

Obr. 7.3: Vizualizácia scény po vykonaní druhej úlohy v tablete (a) a v Microsoft HoloLens 2 (b) .

Používateľ pri vytváraní scenára (Obrázok 7.3) postupoval v nasledujúcich krokoch (Microsoft HoloLens 2):

1. Vytvor projekt s vytvorenej scény.
2. Vyber z menu režim pridať akciu.
3. Polož guľôčku (akčný bod) na miesto, kde sa ma akcia vykonať.
4. Vyber robota alebo dopravníkový pás, podľa toho, kto ma akciu vykonať.
5. Z ponuky vyber akciu, podľa toho, ktorú akciu práve pridávaš.
6. Opakuj od bodu 3 kým nemáš vytvorené všetky akcie.
7. Vyber z menu režim pridať prepojenie.
8. Vyber zo scény akciu (vrátane START), z ktorej má prepojenie vychádzať.
9. Vyber zo scény akciu (vrátane END) podľa toho, ktorá akcia má pokračovať (kam má prepojenie vchádzať).
10. Opakuj od bodu 8 kým nie sú prepojené všetky akcie.
11. Ak je potrebné upraviť pozíciu niektorých akcií, prepni sa do režimu *Transform*.

V tablete okrem výmeny krokov kde používateľ najskôr vyberie objekt a až potom funkciu, ktorú chce vykonať, musí pri tvorbe akcií najprv vybrať akčný bod a následne z menu možnosť pridať akciu.

Niektoré kroky je možné aj vymeniť. Napríklad, používateľ nemusí najprv vytvoriť všetky akcie a až potom pridať prepojenia ale, pre potreby experimentu bolo lepšie použiť používateľovi vysvetliť jeden konkrétny príklad aby nebol zahltený viacerými informáciami naraz.

Skupina	Účastník	Pohlavie	Vek	Vzdelanie
Skupina A	A	Ž	20	SŠ (s maturitou)
	B	M	22	SŠ (s maturitou)
	C	M	21	SŠ (s maturitou)
Skupina B	D	M	21	SŠ (s maturitou)
	E	Ž	21	SŠ (s maturitou)
	F	M	24	VŠ - bakalársky stupeň

Tabuľka 7.1: Demografické údaje účastníkov experimentu. V **skupine A** používatelia najprv vyskúšali rozhranie v tablete a následne v Microsoft HoloLens 2. V **skupine B** používatelia najprv vyskúšali rozhranie v Microsoft HoloLens 2 a následne v tablete.

Dotazník

Súčasťou experimentu sú aj dotazníky, vďaka ktorým je možné výsledky vyhodnotiť. V tejto práci bolo využitých viacero typov dotazníkov.

Pre vyhodnotenie a porovnanie používateľskej skúsenosti s rozhraniami bol použitý dotazník User Experience Questionnaire (UEQ¹), ktorý obsahuje 26 párov protikladov, hodnotených na likertovej škále s výberom možností 1 až 7. Meria klasické aspekty použiteľnosti (*efektívnosť, prehľadnosť, spoľahlivosť*) aj aspekty používateľskej skúsenosti (*originalita, stimulácia*) [6].

Druhým využitým dotazníkom bol index zaťaženia úloh NASA² (NASA TLX). NASA TLX je nástroj na meranie a vykonávanie hodnotenia subjektívnej psychickej záťaže (MWL). Umožňuje určiť MWL účastníka počas vykonávania úlohy. Hodnotí výkon v šiestich dimenziách, aby určil celkové hodnotenie pracovného zaťaženia [5].

K dotazníku boli pridané aj vlastné otázky. Súčasťou bol formulár, ktorý obsahoval demografické otázky, NASA TLX dotazník a vlastné otázky týkajúce sa porovnania rozhraní.

7.2 Pribeh experimentu

Experimentu sa zúčastnili 6 dobrovoľníci vo veku 20 - 24 rokov. Účastníci boli rozdelení do dvoch skupín, skupina A a skupina B. Skupina A ako prvá vyskúšala najprv rozhranie v tablete a skupina B v okuliaroch. Iba dvaja účastníci experimentu mali nejakú skúsenosť s rozšírenou realitou (Tabuľka 7.1).

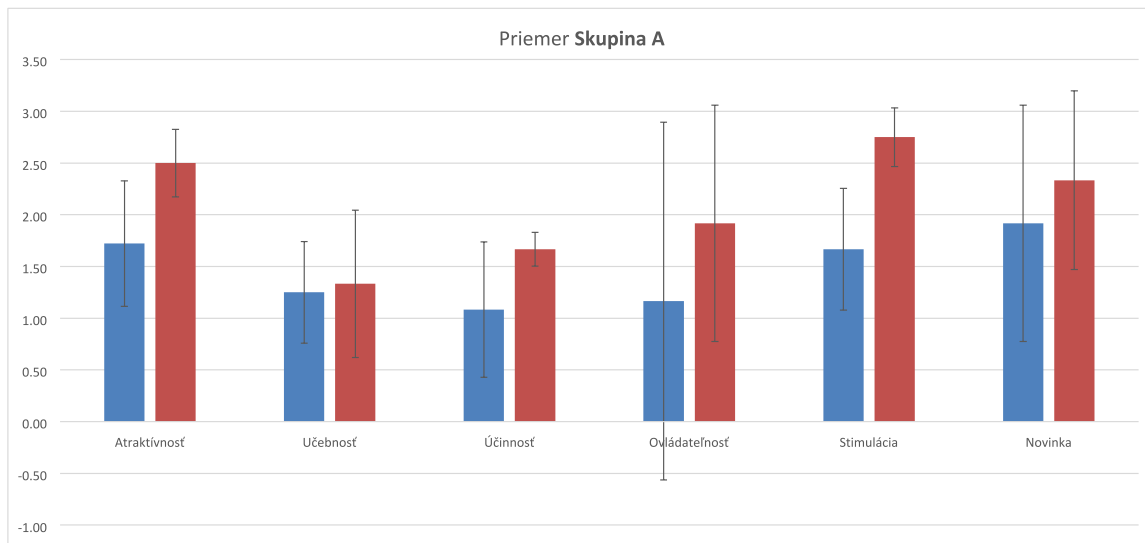
Účastníkom experimentu bolo na začiatku vysvetlené čomu sa venuje táto diplomová práca a základne kroky priebehu experimentu. Používateľ bol v prvom kroku požiadaný o vyplnenie demografických údajov. Následne, podľa toho či používateľ patril do skupiny A alebo skupiny B mu bol vysvetlený koncept riešenia v zariadení pomocou ukážky v ktorej som vytvárala podobné úlohy ako mal používateľ potom vytvoriť sám.

Keďže pri okuliaroch oproti tabletu, používateľ nemohol priamo vidieť aké kroky si vykonávané, bola obrazovka s Microsoft HoloLens 2 premietaná do počítača odkiaľ mohol sledovať rovnaký obraz ako ja.

V ďalšom kroku bolo zariadenie podané používateľovi a mal vykonať úlohy podľa inštrukcií, ktoré mu boli vysvetlené predom. Používatelia pracovali sami a iba v prípade ak to bolo nevyhnutné bola používateľovi poskytnutá pomoc.

¹<https://www.ueq-online.org/>

²<https://digital.ahrq.gov/health-it-tools-and-resources/evaluation-resources/workflow-assessment-health-it-toolkit/all-workflow-tools/nasa-task-load-index>



Obr. 7.4: Graf zobrazuje na ose y priemerné namerané hodnoty v jednotlivých aspektoch použiteľnosti pre **skupinu A**. Priemerné hodnoty pre tablet (modrý stĺpec) sú horšie v porovnaní s hodnotami nameranými pre Microsoft HoloLens 2 (červený stĺpec).

Zároveň, keďže väčšina účastníkov nemala skúsenosť s okuliarmi Microsoft HoloLens 2, bolo potrebné pred testovaním tohto rozhrania spustiť používateľovi aplikáciu priamo dostupnú v zariadení, ktorá ich previedla základným používaním gest a manipuláciami 3D objektami.

Po vyskúšaní rozhrania v prvom zariadení, bol účastník požiadaný vyplniť dotazník, ktorý pozostával z otázok UEQ a NASA TLX dotazník.

Následne mu bol vysvetlený koncept druhého zariadenia spolu s ukážkou používania. Používateľ potom opäť vykonal tú istú úlohu a vyplnil rovnaké dotazníky ako pri prvom zariadení s tým že tentokrát hodnotil druhé zariadenie.

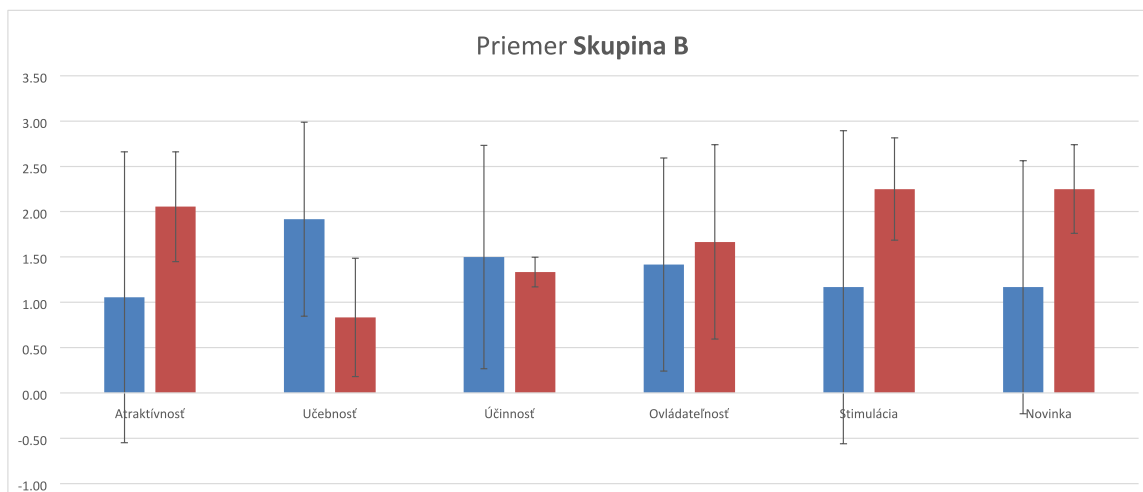
Posledná časť dotazníka sa týkala ohodnotenia ako celku kde účastník odpovedal na otázky porovnávajúc tieto dva zariadenia.

7.3 Výsledky experimentu

Vzhľadom k tomu že vyhodnotenie sa skladá z niekoľkých častí, bola táto sekcia rozdelená do troch pod-sekcií: *Použiteľnosť – UEQ*, *Záťaž – NASA*, *Celkové zhodnotenie*.

Použiteľnosť

Prvá časť vyhodnotenia experimentu sa týka miery použiteľnosti a používateľskej skúsenosti. Na grafe 7.4 sú vyobrazené výsledky, rozdelené podľa meraného aspektu, Skupiny A. Z grafu vyplýva, že okrem učelivosti majú všetky aspekty výraznejšie lepšie priemerné hodnotenie pri používaní náhlavného displeja Microsoft HoloLens 2. Oproti tomu, výsledky skupiny B, vyobrazené na grafe 7.5 ukazujú, že v prípade ak účastníci experimentu vyskúšali ako prvé okuliare, učelivosť tabletovej verzii im prišla výrazne lepšia ako v okuliaroch. Rovnako tak sa aj zmenšil rozdiel medzi účinnosťou a ovládateľnosťou. Naopak, v prípade novosti bolo priemerné hodnotenie ešte o čosi lepšie pre okuliare ako v skupine A.



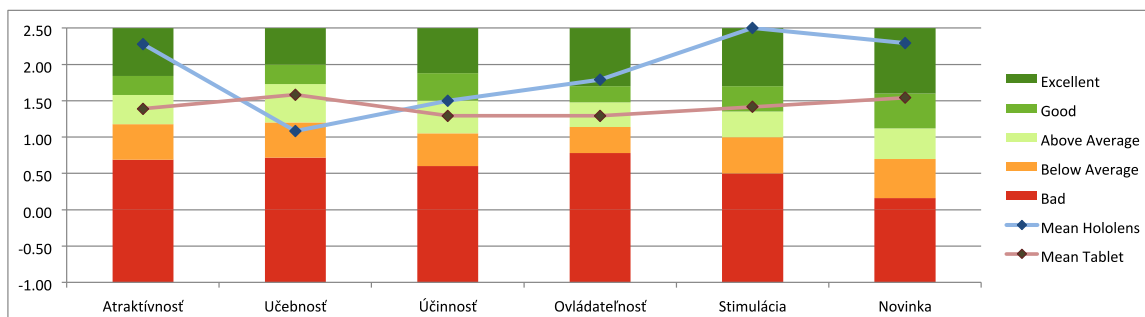
Obr. 7.5: Graf zobrazuje na ose y priemerné namerané hodnoty v jednotlivých aspektoch použiteľnosti pre **skupinu B**. Priemerné hodnoty pre tablet (modrý stĺpec) sú pre učelivosť a účinnosť lepšie v porovnaní s hodnotami nameranými pre Microsoft HoloLens 2 (červený stĺpec). V ostatných prípadoch používatelia hodnotili lepšie rozhranie pre Microsoft HoloLens 2.

Okrem základného vyhodnotenia a následne porovnania medzi skupinami, bol UEQ dotazník použitý aj pre zobrazenie všetkých nameraných hodnôt pre tablet aj okuliare vo vzťahu k existujúcim hodnotám, získaným zo súboru referenčných údajov. Súbor obsahuje dáta od 21175 osôb a 468 štúdií týkajúcich sa rôznych produktov. Vďaka porovnaniu produktu s údajmi v benchmarku je možné určiť jeho relatívnu kvalitu v porovnaní s inými produktami.

Graf 7.6 zobrazuje priemerné hodnoty namerané pre riešenie v Microsoft HoloLens 2 (modrá čiara) a pre riešenie v tablete (červená čiara) v porovnaní s inými produktami. Je vidieť že takmer vo všetkých porovnávacích aspektoch dosahuje riešenie v náhlavnom displeji vysoko nadpriemerné výsledky s porovnaním ostatných produktov. Aspekt učelivosti však ako jediný má horší priemer ako riešenie v tablete a dosahuje aj podpriemerné výsledky v porovnaní s inými riešeniami.

Z týchto výsledkov možno vyhodnotiť, že navrhnuté riešenie pre náhlavný displej Microsoft HoloLens 2 je hodnotené lepšie z pohľadu používateľskej skúsenosti. Používatelia boli viac motivovaní a užívali si používanie tohto riešenia. To môže byť spôsobené aj tým, že nemali predchádzajúcu skúsenosť s týmto zariadením, zatiaľ čo mobilné zariadenia ako je tablet používajú každý deň. Oproti tomu však účinnosť riešenia výraznejšie nepresahovala účinnosť riešenia v tablete a dokonca učelivosť o kus zaostávala. Dôvodom môže byť jednak opäť aspekt novosti, ale aj to že navrhnuté riešenie nebolo prioritne implementované pre tento aspekt a teda riešenie neobsahovalo veľa navigačných prvkov a bolo možné rýchlo vedieť o tom akú funkciu momentálne využíva (Tabuľka 7.2).

Pre vyhodnotenie, či sa namerané výsledky význame líšia bol urobený T-Test s $\alpha = 0.05$. Namerané boli hodnoty pre skupinu A, skupinu B a celkové hodnotenie bez rozdielu (Tabuľka 7.3). Z tabuľky vyplýva, že takmer medzi všetkými nameranými hodnotami nie je významný rozdiel. Výsledky sa významne líšia pri Stimulácii a to celkovo a v prípade kedy účastníci experimentu používali náhlavné zariadenie ako druhé (skupina A). Okrem toho sa v týchto dvoch častiach líšia aj pri Atraktivnosti. Jeden z dôvodov môže byť aj fakt,



Obr. 7.6: Priemerné hodnoty namerané pre riešenie v Microsoft HoloLens 2 (modrá čiara) a pre riešenie v tablete (bordová čiara) v porovnaní s inými produktami.

že experiment bol vykonaný na malej vzorke a pre presnejšie výsledky by bolo potrebné riešenie otestovať na vyššej vzorke.

	Hodnota	Benchmark	Interpretácia
Atraktivnosť	2.28	Excellent	Patrí medzi 10% najlepších výsledkov
Učebnosť	1.08	Below Average	50% riešení je lepších, 25% horších
Účinnosť	1.50	Above Average	25% riešení je lepších, 50% horších
Ovládateľnosť	1.79	Excellent	Patrí medzi 10% najlepších výsledkov
Stimulácia	2.50	Excellent	Patrí medzi 10% najlepších výsledkov
Novinka	2.29	Excellent	Patrí medzi 10% najlepších výsledkov

Tabuľka 7.2: Namerané hodnoty benchmarku pre Microsoft HoloLens 2 v tabuľke. Až 4 zo 6 meraných aspektov patria do 10% najlepších hodnotení. Učebnosť však vykazuje podpriemerné výsledky.

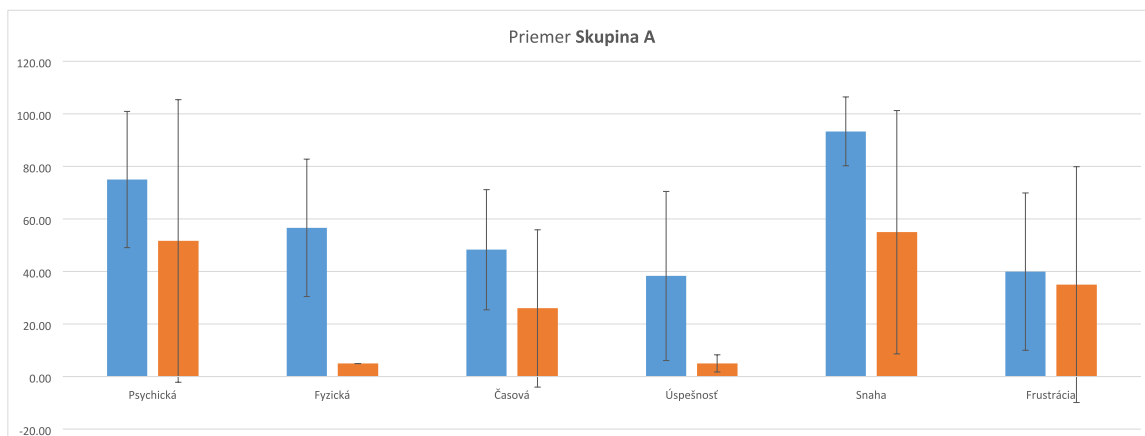
	Skupina A	Skupina B	Všetky
Atraktivnosť	0.034	0.203	0.017
Učebnosť	0.874	0.281	0.336
Účinnosť	0.296	0.830	0.620
Ovládateľnosť	0.188	0.771	0.259
Stimulácia	0.039	0.323	0.038
Novinka	0.199	0.329	0.133

Tabuľka 7.3: P-value pre párový T-test vo všetkých aspektov použiteľnosti pre Skupinu A, Skupinu B a všetky hodnoty spolu. Zvýraznené sú tie hodnoty ktorých $p < 0.05$ a teda sú významne odlišné.

Záťaž

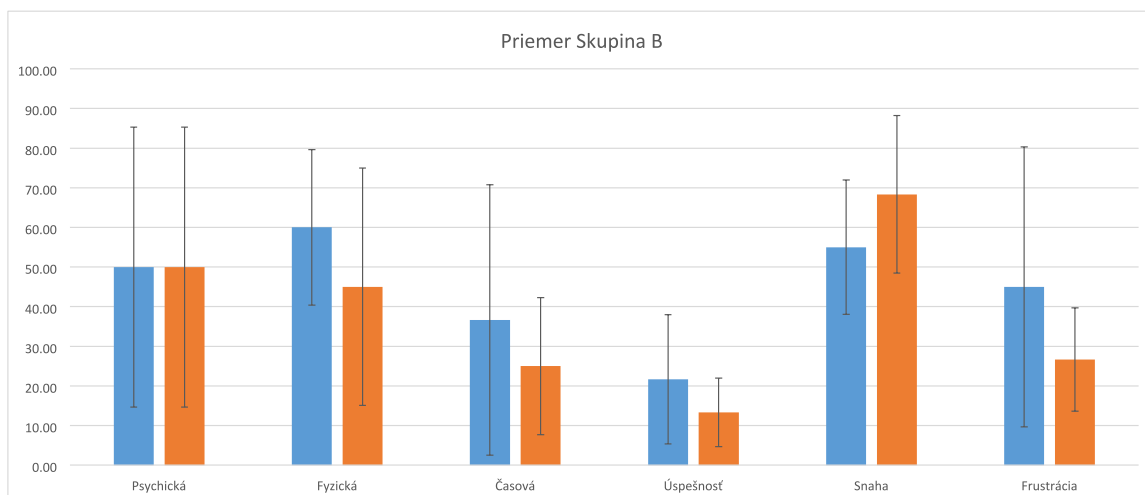
Druhá časť vyhodnotenia sa týka indexu zaťaženia úloh. Pri dotazníku NASA TLX boli využité iba surové dáta bez váhovania. Podobne ako pri vyhodnotení používateľskej časti boli dáta rozdelené podľa skupiny A a skupiny B.

V skupine A pociťovali účastníci experimentu výrazne menšiu mieru fyzického zaťaženia pre Microsoft HoloLens 2 ako pri tablete. Rovnako tak aj pri úspešnosti, mali účastníci



Obr. 7.7: Graf zobrazuje na ose y priemerné namerané hodnoty v jednotlivých aspektoch zataženia pre **skupinu A**. Priemerné hodnoty pre tablet (modrý stĺpec) sú vyššie ako priemerné hodnoty pre Microsoft HoloLens 2. To znamená že tablet má vyššie zataženie pre každý.

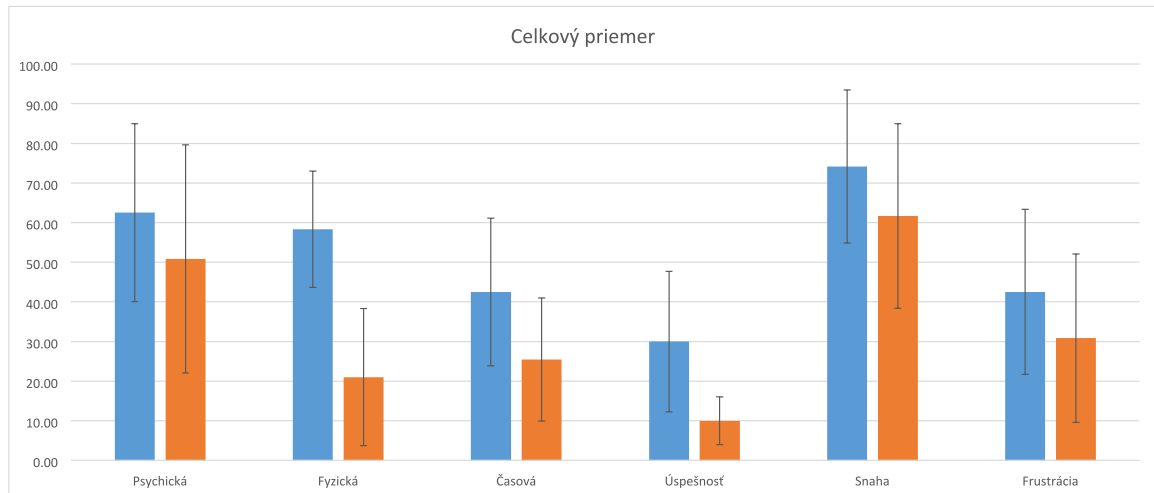
pocit, že sa im darilo lepšie splniť úlohu. V ostatných typoch záťaže boli výsledky pomerne rovnaké. Oproti tomu, v skupine B bola miera zataženia pre všetky typy pomerne rovnaká. Ba dokonca, v prípade vynaloženia snahy pre splnenie úloh sa javí že miera zataženia bola vyššia pri náhľadnom zariadení ako pri tablete.



Obr. 7.8: Graf zobrazuje na ose y priemerné namerané hodnoty v jednotlivých aspektoch zataženia pre **skupinu B**. Priemerné hodnoty pre tablet (modrý stĺpec) sú vyššie ako priemerné hodnoty pre Microsoft HoloLens 2. Vyššiu fyzickú záťaž pocitovali používatelia pri tablete ale zase vyššiu snahu pre vykonanie úlohy museli vynaložiť pri Microsoft HoloLens 2

Z celkového zhodnotenia vyplýva, že fyzické zataženie je významne menšie pre používateľa pri používaní navrhnutého riešenia pre Microsoft HoloLens 2. Tak ako pri používateľskej časti aj tu bol urobený T-Test s $\alpha = 0.05$ a ukázalo sa že rozdiel je naozaj významný. Avšak, pri úspešnosti aj keď sa javí, že používatelia mali pocit že úlohu sa im podarilo splniť lepšie ako pri tablete T-Test neukázal významný rozdiel. V ostatných prípadoch je

síce priemerné zataženie pri každom type menšie, rozdiel nie je významný. Rozdiel vnímať aj pri snahe, kde v prípade skupiny B mali používatelia pocit že musia vynaložiť výrazne viac úsilia pri splnení úlohy. To môže rovnako súvisieť s tým ako pri učenlivosti.



Obr. 7.9: Graf zobrazuje na ose y priemerné namerané hodnoty v jednotlivých aspektoch zataženia pre **všetky** namerané hodnoty bez rozdiel s ktorým zariadením začínali. Priemerné hodnoty pre tablety (modrý stĺpec) sú vyššie ako priemerné hodnoty pre Microsoft HoloLens 2. Celkovo teda používatelia pociťovali vyššie zataženie pri rozhraní v tablete. Najvyšší rozdiel je vidieť pri fyzickom zatažení.

Okrem výsledkov hodnotenia podľa jednotlivých typov je však dôležitý aj celkový index zataženia. Ukazuje sa že je významne odlišný v prípade skupiny A ale aj celkového zhodnotenia (Tabuľka 7.4). Ak sa na hodnotenia použiteľnosti a zataženia vezme ako celok, možno zhodnotiť, že v prípade keď už mali používatelia s aplikáciou nejaké skúsenosti, bolo pre nich celkovo lepšie pracovať s okuliarmi ako v prípade ak najprv pracovali s okuliarmi a potom s tabletom.

	Skupina A	Skupina B	Všetky
Psychická	0.474	1.000	0.430
Fyzická	0.061	0.035	0.004
Časová	0.556	0.225	0.235
Úspešnosť	0.192	0.199	0.087
Snaha	0.330	0.015	0.512
Frustrácia	0.580	0.473	0.312
Celkové	0.003	0.062	0.001

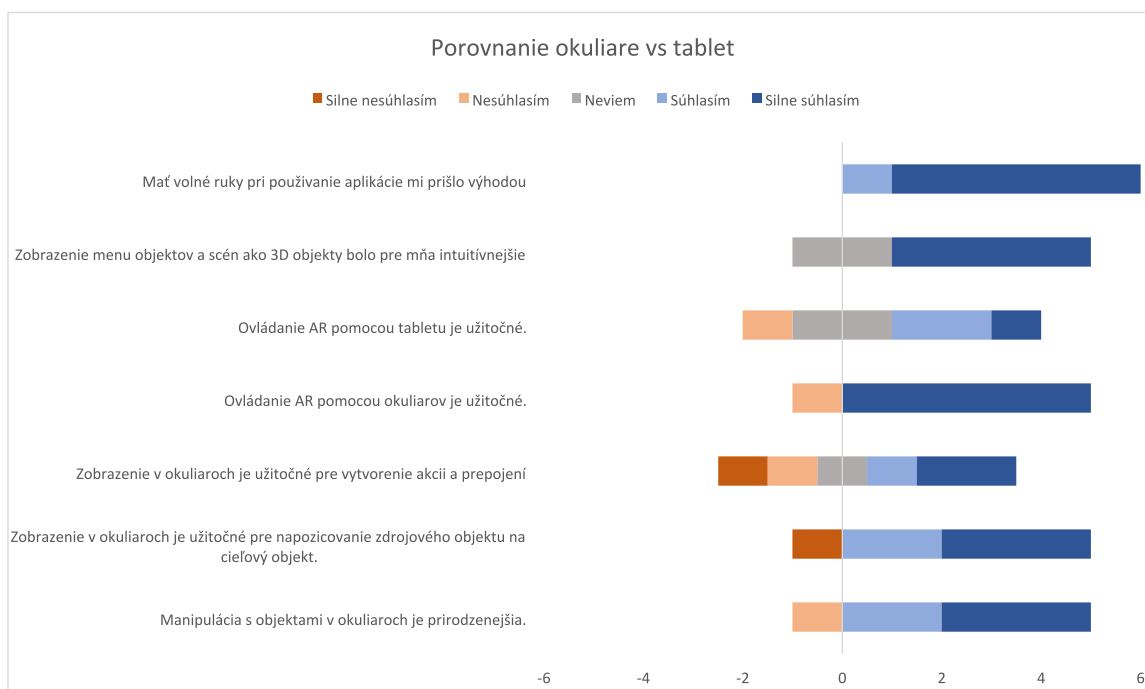
Tabuľka 7.4: P-value pre párový T-test vo všetkých aspektoch zataženia pre Skupinu A, Skupinu B a všetky hodnoty spolu. Zvýraznené sú tie hodnoty ktorých $p < 0.05$ a teda sú významne odlišné. Ako je vidieť z tabuľky, celkové zataženie sa pre rozhranie v tablete a v Microsoft HoloLens 2 výrazne líšia.

Celkové zhodnotenie

Okrem vyhodnotenia použiteľnosti a zataženia bol vytvorený a použitý vlastný dotazník, ktorý sa skladal zo 7 tvrdení a účastník experimentu sa rozhodoval na stupnici 1 až 5 či s tvrdením súhlasí alebo nie. Výsledky hodnotenia sú vyobrazené na grafe 7.10.

Tak ako sa už ukázalo pri záťažovom teste, že z hľadiska fyzickej záťaže sú okuliare porovnateľne lepšie, aj tu pri otázke „Mať voľné ruky pri používaní aplikácie mi prišlo výhodou“ jednoznačne súhlasili. Pri otázkach či je ovládanie AR pomocou okuliarov/tabletu užitočné, účastníci viac súhlasili s tvrdením že „Ovládanie AR pomocou okuliarov je užitočné“. Rovnako tak vyjadrili silný súhlas s tvrdením „Zobrazenie menu objektov a scén ako 3D objekty bolo pre mňa intuitívnejšie“, čo potvrdilo predpoklad toho, že v prípade ak používateľ vidí aký objekt vyberá, vie vybraný objekt porovnať s fyzickým objektom resp. si vie lepšie predstaviť to čo sa pod položkou zoznamu nachádza.

Prirodzenejšie a užitočnejšie im prišla aj manipulácia s objektami z čoho možno usúdiť že ak používateľ pohybuje s objektom podobným spôsobom ako keď pohybuje s fyzickým objektom príde mu to lepšie. O niečo menší súhlas však používateľa vyjadrili pri tvrdení „Zobrazenie v okuliaroch je užitočné pre vytvorenie akcií a prepojení“. Jednou z možných príčin môže byť to že akcie a akčné body bolo ťažšie zaregistrovať kvôli ich veľkosti, ale aj kvôli tomu že sa prelínali s fyzickými alebo inými objektami scény kde nebol na nich dosah.



Obr. 7.10: Výsledné hodnoty vlastného dotazníka v ktorom používatelia vyjadrili mieru súhlasu s jednotlivými tvrdeniami na ľavej strane. Mínusové hodnoty predstavujú mieru nesúhlasu a plusové mieru súhlasu.

7.4 Nápady do budúcnosti

Aj vďaka účastníkom experimentu a ich postrehov sa podarilo nazbierať hneď niekoľko typov na zlepšenie používateľského rozhrania. Sekcia je preto rozdelená na viacero častí

kde som zosumarizovala niekoľko postrehov či už od samotných používateľov ale aj mojich vlastných.

Priestorové mapovanie, prelínanie

V konfiguračných skriptoch je možné jednoducho zapnúť alebo vypnúť priestorové mapovanie. Toto nastavenie som ponechala od začiatku implementácie zapnuté keďže sa javilo ako výhodou. Hlavne pri testovaní sa však ukázalo že v prípade ak je 3D objekt umiestnený pod povrchom fyzického objektu je v podstate nedosiahnuteľný. Problém to spôsobovalo už aj pri umiestnení robota na miesto reálneho robota. Tento problém by bolo možné jednoducho vyriešiť vypnutím priestorového mapovania. Avšak, okrem toho, prelínanie objektov je problém všeobecne aj pri 3D objektoch. Či už pri súradnicových osiach, kedy je možné os zachytiť iba mimo objektu ale hlavne pri umiestnení akcií, ktoré častokrát musia byť umiestnené tesne pri robotovi. Preto by bolo zrejme vhodnejšie si naštudovať problém prelínania a možnosti ako to riešiť.

Navigačné okno, hlasové zadávanie príkazov

Jeden z najväčších problémov pri testovaní bolo, že používatelia nevedeli jednoducho zistiť v akom stave sa aplikácia nachádza. Jedna z prvých najrýchlejších úprav by bolo zvýraznenie tlačidla, ktoré je práve kliknuté. Toto zvýraznenie sa už v aplikácii aj nachádza ale je takmer neviditeľné. Ďalším možným vylepšením by mohlo byť informačné okno, ktoré by ukazovalo používateľovi v akom režime sa práve nachádza, resp. ktorý objekt má práve vybraný. MRTK ponúka aj možnosť ovládania tlačidiel pomocou hlasových príkazov. Táto funkcia nie je vôbec náročná doplniť ale zároveň môže používateľovi pomôcť rýchlejšie interagovať s prostredím.

Manipulácia s robotom

Prvotné návrhy boli implementovať túto funkcionality už v tejto časti riešenia. To však vzhľadom ku komplexnosti problému nebolo časovo realizovateľné. Táto funkcia by však mohla výrazne pomôcť používateľovi pri určovaní umiestnenia akcií, keďže by už priamo podľa modelu vedel určiť či je možné z daného miesta akciu vykonať. Implementácia funkcie by však zrejme vyžadovala zmenu, resp. prídanie novej RPC³ na serveri, keďže už teraz je možné manipulovať s robotom v tabletovej verzii pomocou komunikácie so serverom ale iba pomocou jednej osi.

Ukotvenie k začiatku súradnicovej osi podľa z

Jedným z návrhov používateľov bolo ukotvenie objektov podľa osi z. To by znamenalo že by pri manipulácií s objektami nebolo možné objekty umiestniť pod nulový bod osi z. Objekty by tak bolo jednoduchšie umiestniť na povrch pracovného stola bez toho aby visel vo vzduchu alebo sa prelínal s jeho plochou.

Preddefinovaný režim transformácie

Ďalším, podľa môjho názoru dobrým postrehom bolo určenie preddefinovaného režimu. Táto možnosť by sa týkala nového pridaného objektu. Keď používateľ pridá nový objekt do scény

³Remote procedure call – vzdialený volanie procedúr

automaticky sa nastaví režim transformácie. To by znamenalo že už nemusí opäť otvoriť menu, explicitne vybrať režim transformácie a následne kliknúť na nový objekt ale rovno môže s objektom manipulovať. Vzhľadom k tomu že nový objekt je po pridaní umiestnený rovno pred používateľa, takže manipulácia s ním je vysoko pravdepodobná, by táto možno výrazne uľahčila používateľovi jeho prácu.

Neimplementované funkcionality z tabletu

Niektoré z týchto funkcionalít boli popísane osobitne. Avšak, ´funkcií, ktoré by mohli zlepšiť fungovanie navrhnutého používateľského rozhrania je viac. Napríklad základné premenovanie objektov alebo scén, parametrizované akcie (rýchlosť, dĺžka), pridávanie akcií priamo k efektorom robota, simulácia scény a mnoho iných.

Kapitola 8

Záver

Hlavným cieľom tejto práce bolo zistiť použiteľnosť náhlavného displeja Microsoft HoloLens 2 pre rozšírenú realitu pri programovaní robotického pracoviska. Navrhnuté a implementované riešenie je rozšírením existujúceho riešenia používateľského rozhrania AEditor pripojené k ARServeru. V práci bola implementovaná značná časť funkcionalít, ktoré je možné s ARServer vykonať. Implementované riešenie umožňuje používateľovi zobrazit 3D objekty scény, ktoré sú umiestnené podľa kalibračného bodu do priestoru a následne s nimi interagovať.

K interakcii s objektami bolo implementované jednoduché ručné menu, fungujúce na princípe režimov. To znamená, že používateľ môže interagovať s objektami podľa toho v akom režime je a nie je potrebné vyberať funkciu pri každom objekte zvlášť.

Medzi implementované režimy patrí Transform. Ide o jednu zo základných funkcionalít využívaných pri interakcii. Používateľ môže s objektami manipulovať z blízka ale aj z diaľky pomocou pointeru. K upresneniu pozície bola implementovaná aj súradnicová os, vďaka ktorej je možné manipulovať s každou osou zvlášť.

K ďalším základným, implementovaným režimom patrí pridávanie akcií a akčných bodov. Navrhnuté riešenie ponúka oproti tabletovej verzii niekoľko zmien. Používateľ pridáva k akčnému bodu priamo akciu, ktorú má vykonať. Objekt, ku ktorému má byť pridaná akcia si používateľ vyberá kliknutím na jeho model v scéne a zoznam jeho akcií sa zobrazí na mieste, kde má byť akcia umiestnená. Aby mohol používateľ jednoducho umiestniť akčný bod na miesto kde má byť akcia vykonaná, bola implementovaná funkcia, ktorá zachytáva pozíciu ruky a po stlačení prstov umiestni objekt na miesto ich stlačenia. K prepojeniu akcií boli implementované aj prepojenia, aby používateľ mohol pridávať do scény nové objekty a upravovať existujúce scény, bol implementovaný zoznam objektov a scén. Položky zoznamov sú vizualizované pomocou 3D objektov. Používateľ tak môže ľahšie identifikovať, ktorú položku chce vybrať.

Pre vyhodnotenie riešenia bol vytvorený experiment, ktorého sa zúčastnilo 6 dobrovoľníkov. Účastníci experimentu v jednoduchej úlohe vyskúšali navrhnuté riešenie a tabletovú verziu. Hlavnými sledovanými aspektami boli použiteľnosť a zaťaženie. Z experimentov vyplynulo, že navrhnuté riešenie pri celkovom zaťažení malo výrazne nižšie hodnoty (34,58 bodov) na rozdiel od tabletu (53,75 bodov). Najvyšší rozdiel bol vo fyzickej záťaži, čo potvrdilo predpoklad, že držanie tabletu spôsobuje diskomfort. V použiteľnosti a špeciálne v používateľskej skúsenosti bolo taktiež výrazne lepšie pri navrhnutom riešení. Používatelia si viac užívali prácu s okuliarmi ako s tabletom. Oproti tomu však navrhnuté riešenie dosiahlo pri porovnaní s inými produktami podpriemerné výsledky pri učenívosti. To môže

byť spôsobené hlavne tým, že riešenie neponúka žiadne navigačné prvky a teda je ťažšie sa orientovať v akom stave sa používateľ nachádza.

Celkovo teda možno zhodnotiť, že použitie náhlavného displeja pri programovaní robotického pracoviska je užitočné a prináša niekoľko výhod oproti tabletu. Nedostatky a postrehy zistené pri testovaní môžu byť dobrým základom pre ďalšie pokračovanie práce.

Literatúra

- [1] *BIM POINT A co je vlastně ten BIM?*
[https://www.bim-point.com/blog/a-co-je-vlastne-ten-bim?gclid=Cj0KCQjwsdiTBhD5ARIsAIPw8CK1n-jF5ItL_PmDba6tSQcWR9JH0DL54Ldpz7gGqSj1w2A6hNIm11YaAoCaEALw_wcB]. Accessed: 2022-05-07.
- [2] *FILE FORMAT what is a GLTF file?* [<https://docs.fileformat.com/3d/gltf/>]. Accessed: 2022-05-07.
- [3] *Microsoft Mixed Reality Introducing instinctual interactions*
[<https://docs.microsoft.com/sk-sk/windows/mixed-reality/design/interaction-fundamentals>]. Accessed: 2021-12-20.
- [4] *Microsoft Mixed Reality IUniversal Windows Platform*
[<https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide>]. Accessed: 2021-12-20.
- [5] *NASA Task Load Index* [<https://digital.ahrq.gov/health-it-tools-and-resources/evaluation-resources/workflow-assessment-health-it-toolkit/all-workflow-tools/nasa-task-load-index>]. Accessed: 2021-4-20.
- [6] *UEQ User Experience Questionnaire* [<https://www.ueq-online.org/>]. Accessed: 2022-4-20.
- [7] AZUMA, R. T. A survey of augmented reality. *Presence: teleoperators & virtual environments*. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info 1997, zv. 6, č. 4, s. 355–385.
- [8] BAHRI, H., KRČMARIK, D., MOEZZI, R. a KOČÍ, J. Efficient use of mixed reality for bim system using microsoft HoloLens. *IFAC-PapersOnLine*. Elsevier. 2019, zv. 52, č. 27, s. 235–239.
- [9] BILLINGHURST, M., CLARK, A. a LEE, G. A survey of augmented reality. 2015.
- [10] BIMBER, O. a RASKAR, R. *Spatial augmented reality: merging real and virtual worlds*. CRC press, 2005.
- [11] BURDEA, G. a COIFFET, P. Virtual Reality Technology. *Presence*. December 2003, zv. 12, s. 663–664. DOI: 10.1162/105474603322955950.
- [12] CABONI, F. a HAGBERG, J. Augmented reality in retailing: a review of features, applications and value. *International Journal of Retail & Distribution Management*. Emerald Publishing Limited. 2019.

- [13] CALIGIANA, P., LIVERANI, A., CERUTI, A., SANTI, G. M., DONNICI, G. et al. An Interactive Real-Time Cutting Technique for 3D Models in Mixed Reality. *Technologies*. Multidisciplinary Digital Publishing Institute. 2020, zv. 8, č. 2, s. 23.
- [14] CHIEN, J.-C., TSAI, Y.-R., WU, C.-T. a LEE, J.-D. HoloLens-based AR system with a robust point set registration algorithm. *Sensors*. Multidisciplinary Digital Publishing Institute. 2019, zv. 19, č. 16, s. 3555.
- [15] CONDINO, S., TURINI, G., PARCHI, P. D., VIGLIALORO, R. M., PIOLANTI, N. et al. How to build a patient-specific hybrid simulator for orthopaedic open surgery: benefits and limits of mixed-reality using the Microsoft HoloLens. *Journal of healthcare engineering*. Hindawi. 2018, zv. 2018.
- [16] HANNA, M. G., AHMED, I., NINE, J., PRAJAPATI, S. a PANTANOWITZ, L. Augmented reality technology using Microsoft HoloLens in anatomic pathology. *Archives of pathology & laboratory medicine*. the College of American Pathologists. 2018, zv. 142, č. 5, s. 638–644.
- [17] HIETANEN, A., PIETERS, R., LANZ, M., LATOKARTANO, J. a KÄMÄRÄINEN, J.-K. AR-based interaction for human-robot collaborative manufacturing. *Robotics and Computer-Integrated Manufacturing*. Elsevier. 2020, zv. 63, s. 101891.
- [18] HUANG, J. *End-to-End Programming Tools for Mobile Manipulator Robots*. Dizertačná práca.
- [19] KAPINUS, M., BERAN, V., MATERNA, Z. a BAMBUŠEK, D. Spatially Situated End-User Robot Programming in Augmented Reality. In: IEEE. *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 2019, s. 1–8.
- [20] MILGRAM, P. a KISHINO, F. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*. The Institute of Electronics, Information and Communication Engineers. 1994, zv. 77, č. 12, s. 1321–1329.
- [21] MORALES MOJICA, C. M., NAVKAR, N. V., TSEKOS, N. V., TSAGKARIS, D., WEBB, A. et al. Holographic Interface for three-dimensional Visualization of MRI on HoloLens: A Prototype Platform for MRI Guided Neurosurgeries. In: *2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE)*. 2017, s. 21–27. DOI: 10.1109/BIBE.2017.00-84.
- [22] MORO, C., PHELPS, C., REDMOND, P. a STROMBERGA, Z. HoloLens and mobile augmented reality in medical and health science education: A randomised controlled trial. *British Journal of Educational Technology*. 2021, zv. 52, č. 2, s. 680–694. DOI: <https://doi.org/10.1111/bjet.13049>. Dostupné z: <https://bera-journals.onlinelibrary.wiley.com/doi/abs/10.1111/bjet.13049>.
- [23] MOURTZIS, D., SIATRAS, V. a ZOGOPOULOS, V. Augmented reality visualization of production scheduling and monitoring. *Procedia CIRP*. Elsevier. 2020, zv. 88, s. 151–156.
- [24] OSTANIN, M., MIKHEL, S., EVLAMPIEV, A., SKVORTSOVA, V. a KLIMCHIK, A. Human-robot interaction for robotic manipulator programming in Mixed Reality. In:

- IEEE. *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, s. 2805–2811.
- [25] PARK, S., BOKIJONOV, S. a CHOI, Y. Review of microsoft hololens applications over the past five years. *Applied Sciences*. Multidisciplinary Digital Publishing Institute. 2021, zv. 11, č. 16, s. 7259.
- [26] REKIMOTO, J. a NAGAO, K. The world through the computer: Computer augmented interaction with real world environments. In: *Proceedings of the 8th annual ACM symposium on User interface and software technology*. 1995, s. 29–36.
- [27] UNGUREANU, D., BOGO, F., GALLIANI, S., SAMA, P., DUAN, X. et al. HoloLens 2 Research Mode as a Tool for Computer Vision Research. *CoRR*. 2020, abs/2008.11239. Dostupné z: <https://arxiv.org/abs/2008.11239>.
- [28] VASSALLO, R., RANKIN, A., CHEN, E. C. S. a PETERS, T. M. Hologram stability evaluation for Microsoft HoloLens. In: KUPINSKI, M. A. a NISHIKAWA, R. M., ed. *Medical Imaging 2017: Image Perception, Observer Performance, and Technology Assessment*. SPIE, 2017, sv. 10136, s. 295 – 300. DOI: 10.1117/12.2255831. Dostupné z: <https://doi.org/10.1117/12.2255831>.
- [29] VORRABER, W., GASSER, J., WEBB, H., NEUBACHER, D. a URL, P. Assessing augmented reality in production: Remote-assisted maintenance with HoloLens. *Procedia CIRP*. Elsevier. 2020, zv. 88, s. 139–144.
- [30] WU, M., DAI, S.-L. a YANG, C. Mixed reality enhanced user interactive path planning for omnidirectional mobile robot. *Applied Sciences*. Multidisciplinary Digital Publishing Institute. 2020, zv. 10, č. 3, s. 1135.
- [31] YIGITBAS, E., JOVANOVIKJ, I. a ENGELS, G. Simplifying robot programming using augmented reality and end-user development. In: Springer. *IFIP Conference on Human-Computer Interaction*. 2021, s. 631–651.
- [32] ZHANG, L., CHEN, S., DONG, H. a EL SADDIK, A. Visualizing toronto city data with hololens: Using augmented reality for a city model. *IEEE Consumer Electronics Magazine*. IEEE. 2018, zv. 7, č. 3, s. 73–80.

Príloha A

Obsah priloženého pamäťového média

/	
_ xhiad100.pdf/	Text práce
_ tex/	L ^A T _E X zdrojové kódy
_ src/	Zdrojový kód aplikácie
_ video.mp4	Video aplikácie
_ README.txt	Textový súbor s inštrukciami pre inštaláciu a spustenie aplikácie

Príloha B

**Dotazník User Experience
Questionnaire a NASA Task Load
Index**

Teraz prosím ohodnotte daný produkt označením jedného krúžku v každom riadku.

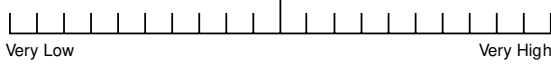

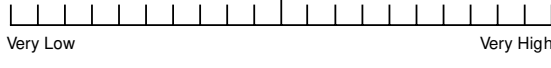

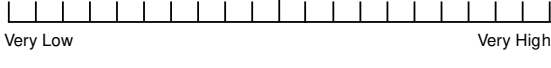

	1	2	3	4	5	6	7		
obťažujúci	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pútavý	1
nepochopiteľný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pochopiteľný	2
nápaditý	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tuctový	3
intuitívny	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	neintuitívny	4
hodnotný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menejcenný	5
nudný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	vzrušujúci	6
nezaujímavý	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	zaujímavý	7
nepredvídateľný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	predvídateľný	8
rýchly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pomalý	9
moderný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tradičný	10
obmedzujúci	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	podporujúci	11
dobrý	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	zlý	12
zložitý	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	jednoduchý	13
odpuďujúci	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	potešujúci	14
bežný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	špičkový	15
neprijemný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	príjemný	16
spoľahlivý	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nespoľahlivý	17
motivujúci	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivujúci	18
spĺňajúci očakávania	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nespĺňajúci očakávania	19
neefektívny	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	efektívny	20
jasný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mätúci	21
nepraktický	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	praktický	22
prehľadný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	neprehľadný	23
príťažlivý	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nepríťažlivý	24
sympatický	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nesympatický	25
konzervatívny	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inovatívny	26

Obr. B.1: User Experience Questionnaire.

Figure 8.6

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
Mental Demand How mentally demanding was the task?		
		
Physical Demand How physically demanding was the task?		
		
Temporal Demand How hurried or rushed was the pace of the task?		
		
Performance How successful were you in accomplishing what you were asked to do?		
		
Effort How hard did you have to work to accomplish your level of performance?		
		
Frustration How insecure, discouraged, irritated, stressed, and annoyed were you?		
		

Obr. B.2: NASA Task Load Index.