



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

**BRÝLE NEBO TABLET? POUŽITELNOST ZAŘÍZENÍ V
KONTEXTU PROGRAMOVÁNÍ ROBOTICKÝCH PRA-
COVIŠŤ**

HEAD-MOUNTED DISPLAY OR TABLET? DEVICE USABILITY IN THE CONTEXT OF PROGRAM-
MING ROBOTIC WORKPLACES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JOSEF KNÍŽE

VEDOUcí PRÁCE

SUPERVISOR

Ing. DANIEL BAMBUŠEK

BRNO 2024

Zadání diplomové práce



153716

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Student: **Kníže Josef, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Vývoj aplikací
Název: **Brýle nebo tablet? Použitelnost zařízení v kontextu programování robotických pracovišť**
Kategorie: Uživatelská rozhraní
Akademický rok: 2023/24

Zadání:

1. Prostudujte koncept rozšířené reality a její využití v robotice. Seznamte se s brýlemi Microsoft HoloLens 2, fakultní experimentální platformou ARCOR2, jejími možnostmi a aplikačním rozhraním.
2. Analyzujte stávající verze uživatelských rozhraní pro programování robotického pracoviště pomocí brýlí HoloLens a pomocí tabletu. Proveďte uživatelský experiment srovnávající tato dvě rozdílná rozhraní se zaměřením na uživatelskou použitelnost a zkušenost.
3. Na základě výsledků experimentu navrhnete sadu zlepšení pro rozhraní v brýlích HoloLens s cílem vylepšit uživatelskou použitelnost a zkušenost.
4. Navrženou sadu implementujte.
5. Proveďte závěrečnou iteraci uživatelských experimentů zaměřených na srovnání zlepšeného rozhraní v brýlích HoloLens s rozhraním v tabletu. Vyhodnoťte vlastnosti výsledného řešení.
6. Vytvořte 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, prof. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 17.5.2024
Datum schválení: 9.11.2023

Abstrakt

Tato práce se zaměřuje na srovnání použitelnosti brýlí Microsoft HoloLens 2 a tabletu v kontextu programování robotického pracoviště pomocí rozšířené reality. V rámci práce jsou analyzována existující rozhraní pro tato zařízení a byl navržen a proveden uživatelský experiment pro jejich porovnání, ve kterém je ve dvou iteracích vyzkoušelo a ohodnotilo celkem 18 účastníků. Na základě zpětné vazby z první iterace testování bylo navrženo a implementováno mnoho změn v rozhraní pro HoloLens 2. Mezi hlavní změny patří převod celé aplikace na novější framework pro smíšenou realitu, programování pomocí interakce se scénou, přepracování umístění robotů, rozšíření o konfiguraci parametrů a umístování akce pomocí robota. Upravená verze rozhraní byla otestována v druhé iteraci experimentů a ukázalo se, že při využití HoloLens 2 jsou uživatelé o 32,3 % rychlejší a hodnotí použitelnost i uživatelskou zkušenost lépe než při použití tabletu. Překvapivě mezi zařízeními nebyl naměřen žádný rozdíl v mentální ani fyzické zátěži uživatele.

Abstract

This thesis focuses on usability comparison of head-mounted display Microsoft HoloLens 2 and tablet in context of programming robotic workplace using augmented reality. The existing interfaces for these devices are analyzed, and a user experiment was designed and conducted to compare them, involving a total of 18 participants who tested and evaluated both devices in two iterations. Based on feedback from the first iteration, many changes were proposed and implemented in the HoloLens 2 interface. Key changes included porting the entire application to newest framework for mixed reality, programming through scene interaction, reworking robot placements, and adding parameter configuration and action placement through the robot. The new version of the interface was tested in the second iteration of experiments, revealing that users are 32,3 % faster when using HoloLens 2 and they rate usability and user experience better compared to tablet. Surprisingly, there was no significant difference measured in user mental or physical load between the devices.

Klíčová slova

Rozšířená realita, Microsoft HoloLens 2, AREditor, Vizualní programování, Robotické pracoviště, Unity

Keywords

Augmented reality, Microsoft HoloLens 2, AREditor, Visual programming, Robotic workplace, Unity

Citace

KNÍŽE, Josef. *Brýle nebo tablet? Použitelnost zařízení v kontextu programování robotických pracovišť*. Brno, 2024. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Daniel Bambušek

Brýle nebo tablet? Použitelnost zařízení v kontextu programování robotických pracovišť

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Daniela Bambuška. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Josef Kníže
15. května 2024

Poděkování

Rád bych poděkoval svému vedoucímu panu Ing. Danielu Bambuškoví, za vedení této práce a cenné rady. Zároveň bych rád poděkoval všem účastníkům uživatelského experimentu.

Obsah

1	Úvod	2
2	Rozšířená realita pro interakci člověka s robotem	3
2.1	Definice	3
2.2	Smíšená realita (Mixed Reality)	4
2.3	Typy AR zařízení	5
2.4	Využití AR v robotice	6
2.5	Microsoft HoloLens 2	7
3	Současné řešení	10
3.1	Reprezentace programu	10
3.2	Rozhraní pro tablet	11
3.3	Rozhraní pro HoloLens 2	13
4	Uživatelský experiment	18
4.1	Existující studie	18
4.2	Popis experimentu	18
4.3	Výsledky 1. iterace uživatelských testů	23
4.4	Odhalené nedostatky	26
5	Návrh změn v uživatelském rozhraní	32
5.1	Změny v existujícím řešení	32
5.2	Nově implementované funkce	35
5.3	Obecné změny	36
6	Implementace	37
6.1	Převod na MRTK3	37
6.2	Interakce s objekty scény	39
6.3	Zobrazení seznamů	43
7	Výsledky finálního experimentu	45
7.1	Srovnání tabletu a HoloLens 2	45
7.2	Efekt implementovaných změn	52
7.3	Návrhy pro navazující práce	53
8	Závěr	56
	Literatura	57

Kapitola 1

Úvod

S rozvojem technologií a zlepšení cenové dostupnosti kolaborativních robotů neustále narůstá automatizace i v menších firmách, ve kterých jsou vyráběny menší počty výrobků, které zároveň mohou být přizpůsobeny podle požadavků zákazníka. Kolaborativní roboti jsou flexibilní zařízení, kteří mohou vykonávat širokou škálu úloh. Ti ale musí být naprogramováni, což může být finančně náročné, obzvláště pokud se úlohy často mění. Z tohoto důvodu je vhodné vytvářet rozhraní, pomocí kterých mohou úlohy pro roboty vytvářet i méně kvalifikováni zaměstnanci.

Z těchto důvodů byl v rámci výzkumné skupiny Robo@FIT¹ vyvinut ARCOR2². Jedná se o systém pro jednoduché programování robotů koncovým uživatelem. Tento systém je možné ovládat pomocí aplikace na tablet AREditor³, která umožňuje uživatelům vizuálně programovat úlohy pro roboty přímo na robotickém pracovišti. Podobná aplikace byla implementována i pro brýle Microsoft HoloLens 2 v rámci diplomové práce Simony Hiadlovské [6].

Vyskytují se tedy otázky. Které z těchto dvou zařízení je vhodnější pro programování robotického pracoviště? Vyplatí větší investice, kterou představuje nákup brýlí Microsoft HoloLens 2? Má smysl dále vyvíjet rozhraní pro tuto platformu? Brýle poskytují uživatelům snadnější vnímání hloubky, a měly by tedy usnadnit vnímání i tvorbu programů pro roboty. Zároveň umožňují interakci se scénou pomocí detekce rukou, která by mohla být pro uživatele přívětivější. Uživatel má při užívání volné ruce a může interagovat s fyzickými roboty. Možnou nevýhodou HoloLens 2 může být fakt, že uživatelé s touto technologií obvykle nejsou obeznámeni a bylo by nutné, aby si na tuto technologii zvykli.

Cílem této práce je navázat na práci Simony Hiadlovské [6], vyřešit případné nedostatky v uživatelském rozhraní pro HoloLens 2 a porovnat výslednou verzi s rozhraním pro tablet. Součástí bude také rozebrat silné a slabé stránky obou zařízení a odpovědět na předešlé otázky.

V kapitole 2 jsou představeny základy rozšířené reality a její využití pro interakci s roboty, včetně detailního rozboru zařízení Microsoft HoloLens 2. Současné řešení uživatelských rozhraní pro obě zařízení jsou probrány v kapitole 3. V kapitole 4 je popsán experiment navržený k porovnání zařízení a rozhraní. Kapitola 5 se zabývá návrhem změn, které cílí na opravení nedostatků odhalených při první iteraci experimentů. Implementace změn je popsána v kapitole 6. V kapitole 7 jsou vyhodnoceny naměřené hodnoty a porovnání tabletu a HoloLens 2. Zhodnocení výsledků této práce je v kapitole 8.

¹<https://www.fit.vut.cz/research/group/robo/.cs>

²<https://github.com/robofit/arcor2>

³https://github.com/robofit/arcor2_areditor

Kapitola 2

Rozšířená realita pro interakci člověka s robotem

Tato kapitola rozebírá základy rozšířené reality (Augmented Reality – AR). Nejdříve bude popsána definice a potřebné části pro vytvoření AR systému (sekce 2.1). V sekci 2.2 je popsána smíšená realita a její vztah k rozšířené realitě. Dále jsou v sekci 2.3 popsány zařízení využívané pro rozšířenou realitu a zobrazovací techniky, které využívají. Existující využití AR pro programování robotů jsou popsány v sekci 2.4. Nakonec je v sekci 2.5 detailněji popsáno zařízení Microsoft HoloLens 2, které bude využito v implementační části.

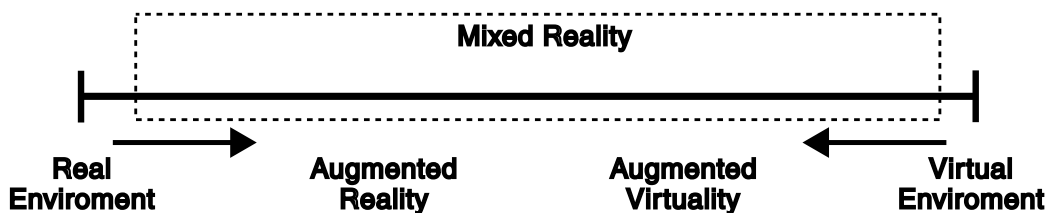
2.1 Definice

Nejrozšířenější definice rozšířené reality byla formulována v roce 1997 v práci Ronalda T. Azumy [1]. Tato definice obsahuje tři charakteristické prvky, které systém musí splňovat, aby mohl být považován za rozšířenou realitu [23]. Mezi tyto prvky patří:

- **Kombinace reálného a virtuálního světa** – Rozšířená realita integruje virtuální prvky do reálného světa. To znamená, že uživatelé vnímají a interagují s fyzickým prostředím, které je obohaceno o digitální informace či objekty.
- **Interakce v reálném čase** – Rozšířená realita umožňuje uživatelům interagovat s virtuálními prvky v reálném čase. Toto minimálně zahrnuje sledování a reagování na pohyb zařízení.
- **3D lokalizace objektů** – Systém rozšířené reality musí být schopen přesně umístit virtuální objekty v reálném prostoru, pro dosažení maximálního realismu a integrace digitálních prvků s okolním světem a to i při pohybu zobrazovacího zařízení.

Pro splnění těchto požadavků musí systém obsahovat následující 3 komponenty:

- **Sledování polohy a rotace (Tracking)** – Pro správné zobrazení virtuálních objektů, je nutné, aby zařízení v reálném čase znalo svou polohu a rotaci v reálném světě. Pro sledování polohy zařízení je možné využít GPS a akcelerometr, kde GPS slouží spíše k venkovnímu využití kvůli její nepřesnosti. K sledování rotace je využíván gyroskop, který poskytuje informace o změnách úhlů natočení zařízení. Akcelerometr a gyroskop jsou často zakomponovány do tzv. *inertial measurement unit* (IMU). Tyto technologie obvykle nejsou dost přesné a je nutné je kombinovat spolu s technikami z počítačového



Obrázek 2.1: Milgramův koncept Reality-Virtuality kontinua.

vidění, jelikož AR zařízení jsou obvykle vybavena kamerou nebo dokonce i hloubkovými senzory [23].

Zajímavý způsob sledování polohy je pomocí využití magnetických trackerů. Tyto zařízení jsou umístěny do prostoru a generují magnetické pole. Podle polarizace a orientace měřeného magnetického pole je zařízení schopné se orientovat v prostoru. Jelikož magnetické pole je náchylné na rušení a nestabilitu je možné ho nahradit trackery, které využívají lasery [2].

- **Umístění virtuálních objektů** (Registration) – Tento proces se zabývá správným umístěním virtuálních objektů do rozšířené projekce reálného světa na základě sledované polohy a rotace zařízení a případně dalších technik, jako například sledování markerů, při kterém jsou v reálném světě sledovány vizuální značky, jako QR kód, nebo jiné obrazce, které systému poskytnou referenční bod pro umístění virtuálních objektů. Proces, při kterém systém detekuje takovouto značku, díky které může vypočítat své umístění v reálném světě se nazývá kalibrace a je nepostradatelná pro aplikace rozšířené reality, které vyžadují přesné a stabilní umístění virtuálních objektů v reálném prostoru [23].
- **Vizualizace** (Visualization) – Vizualizace je proces, jehož cílem je uživateli zobrazit virtuální objekty tak, aby došlo k jejich splynutí s reálným světem. Z velké části se jedná o vykreslování využívané v počítačové grafice (projekce, aplikace textur, stínování atd.), ale vyskytují se zde i speciální problémy, které postihují pouze rozšířenou realitu, jako třeba překrytí s reálnými objekty a synchronizace zdrojů světla v reálném světě s virtuálními [23, 8].

2.2 Smíšená realita (Mixed Reality)

Smíšená realita je pojem, který ve své práci [16] definoval Paul Milgram, který popisuje všechny kombinace reálného a virtuálního prostředí. Spolu se smíšenou realitou představil koncept *Reality-Virtuality* kontinuum (obrázek 2.1), pomocí kterého popisuje příbuznost rozšířené a virtuální reality. V tomto kontinuu je postupný přechod od skutečného prostředí (vlevo) po plně virtuální prostředí (vpravo). Milgram popisuje skutečné prostředí (*Real environment*), jako pohled, který je složen pouze ze všech reálných objektů, které by bylo možné vidět osobně, skrz okno či video. Zatímco virtuální prostředí (*Virtual environment*) popisuje jako prostředí tvořené pouze virtuálními objekty, jedná se tedy o typ počítačem vykreslené simulace, zobrazena buď na monitoru nebo imerzivně. Obohacením skutečného prostředí o digitální obsah získáváme rozšířenou realitu (AR) a v opačném případě, tedy obohacením virtuálního prostředí o skutečné objekty získáváme rozšířenou virtuality (AV).

Milgram se ve své práci spíše zabýval jak je prostředí tvořeno a ne to jak je vnímáno, proto se jeho popis rozšířené reality částečně neslučuje s dříve zmíněnou definicí Azumy, jelikož v jeho popisu chybí interakce v reálném čase.

V dnešní době jsou pojmy AR, MR, VR a další zastřešeny pod pojmem *Extended reality* (XR). Tento pojem se využívá především ve vývoji, jelikož vývoj pro různé reality často řeší stejné problémy a je tedy možné správně pojmenovat sdílené nástroje pro různé reality.

2.3 Typy AR zařízení

V dnešní době se téměř všechna AR zařízení zaměřují především na zrak a většinou se tedy jedná o typ displeje či projekci. V některých případech je vizuální část kombinována s prostorovým zvukem.

Náhlavní displej (Head-Mounted Displays – HMD)

Koncept HMD pro AR je znám od počátku výzkumu rozšířené reality, ale v poslední dekádě zaznamenaly značný technologický růst, který pravděpodobně navazuje na úspěch VR headsetu Oculus Rift a následně první komerčně úspěšnější implementaci AR HMD *HoloLens 1*, která byla uvedena na trh v roce 2016.

Krátce po jejich vydaní následovaly další implementace jako *Magic Leap One*¹ (2018), *HoloLens 2*² (2019), *Meta Quest*³ (2019), *Meta Quest 2*⁴ (2020), *Magic Leap 2*⁵ (2022) a *Meta Quest 3*⁶ (2023) a nejnovější *Apple Vision Pro*⁷ (2024). Tyto displeje mohou být implementovány následujícími způsoby:

- **Video see-through (VST)** – Tento způsob vykreslení AR je založen na kameře, která většinou bývá umístěna na zadní straně displeje. Tato kamera zachycuje reálný svět do kterého jsou přidány virtuální prvky, pomocí principů počítačové grafiky a zpracování videa. Tento systém může být velmi nepříjemný pro uživatele při zvětšené odezvě videa, nebo výpadku zařízení.
- **Optical see-through (OST)** – Tento způsob využívá optického systému, který může využívat částečně průhledné a reflektivní materiály nebo optické hranoly k spojení reálného pohledu s promítanou grafikou. Výhodou těchto displejů je, že systém nemusí nahrávat a vykreslovat skutečný svět. Zároveň při selhání zařízení uživatel stále vidí, což může být důležité při práci v průmyslových podmínkách. Problémy, se kterými se tyto displeje setkávají jsou průhlednost virtuálních objektů a pokrytí uživatelského zorného pole.

Mezi největší výhodu těchto displejů patří stereoskopická projekce (každému oku je obraz zobrazen nezávisle), díky které je možné uživateli navodit dojem hloubky, jelikož lidé přirozeně vnímají hloubku na základě rozdílů dvou obrazů, které vidí očima. Další výhodou je, že při využívání tohoto displeje má uživatel volné ruce a může interagovat s reálným

¹<https://www.magicleap.com/devices-ml1>

²<https://www.microsoft.com/cs-cz/hololens>

³<https://www.oculus.com/quest/refurbished/>

⁴<https://www.meta.com/quest/products/quest-2/>

⁵<https://www.magicleap.com/magic-leap-2>

⁶<https://www.meta.com/quest/quest-3/>

⁷<https://www.apple.com/apple-vision-pro/>

světem. Díky tomuto je možné náhlavní displeje využít, například k trénování montážního procesu [20].

Mezi problémy postihující tento typ zařízení patří především cenová nedostupnost. Například pořizovací cena HoloLens 2 činí 3500 dolarů. Kromě toho mívají tato zařízení omezenou výdrž baterie, která se obvykle pohybuje v rozmezí od 2 do 3 hodin aktivního používání. Zajímavý trend, který je možné sledovat, je postupné snižování výdrže baterie s vývojem nových náhlavních displejů, kde nejnovější Apple Vision Pro bude mít výdrž baterie pouze 2 hodiny. Jedná se pravděpodobně o balancování výkonu a váhy zařízení.

Ruční displeje (Handheld Display)

Ruční displej je nejrozšířenějším typem zařízení, na kterém je využívána rozšířená realita. Jedná se o běžně dostupné chytré telefony a tablety, které pomocí displeje a kamery na zadní straně implementují *Video see-through* AR. Hlavní výhodou těchto zařízení je dostupnost a fakt že většina lidí už umí tato zařízení ovládat. Nevýhodou v kontextu AR je omezené vnímání hloubky virtuálních objektů, které je způsobeno především tím že tablet disponuje pouze jedním dvourozměrným displejem. Další nevýhodou je nutnost displej držet, což uživateli znemožňuje interagovat s fyzickým prostředím.

Promítaná rozšířená realita

Promítaná AR je zajímavou větví tohoto oboru, jedná se promítání obrazu na reálné objekty, čímž je možné změnit jejich vzhled (povrch, texturu a stínování) a doplnit informace, ale není možné měnit jejich tvar či polohu.

2.4 Využití AR v robotice

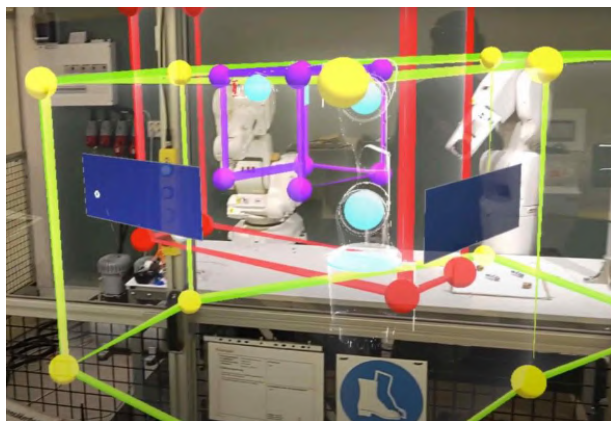
AR se v robotice využívá, jelikož je vhodné obohatit či rozšířit fyzické robotické pracoviště o dodatečné informace. Zároveň je možné, pomocí uživatelova vstupu přes AR, manipulovat s pracovištěm za účelem zefektivnit interakci s robotickým pracovištěm. Obecně se dělí na dvě kategorie:

1. programování a řízení robotů či pracoviště,
2. vizualizace informací pro porozumění programům či pracovišti.

V obou kategoriích bývá přínosem větší efektivita oproti práci na počítači a zároveň je díky AR možné zvýšit bezpečnost [26].

V práci S. Nikolova [17] byla popsána aplikace na HoloLens 2, kterou vytvořil ve spolupráci s firmou ABB Robotics, sloužící k vizualizaci a úpravě bezpečných zón robotů (obrázek 2.2a). Vizualizace přímo v robotickém pracovišti by měla usnadnit nastavení pracoviště a urychlit proces zavedení nových robotů. Bezpečné zóny zahrnují různé omezení, od celkového zakázání vstupu robotického rameno po pouhé omezení rychlosti rotace kloubů.

Další práce využívající HoloLens 2 od Chengxi Li a spol. [10], ve které popisují jejich systém, který se snaží zvýšit bezpečnost robotického pracoviště pomocí vzájemného vnímání prezence robota i člověka. Robot vnímá přítomnost pracovníka pomocí HoloLens 2, díky kterému může robot znát pozice uživatelových rukou a hlavy, ze kterých tvoří „digitální dvojče“ uživatele. Tomuto dvojčeti se následně robot snaží vyhýbat. Samotné vyhýbání bylo implementováno pomocí neuronových sítí.



(a) Ukázka definice různých bezpečných zón kolem robota [17].



(b) Ukázka naplánované trasy robota [18].

Obrázek 2.2: Ukázka existujících řešení využívající AR k interakci s robotickým pracovištěm.

HoloLens 2 ve své práci použili taktéž M. Ostanin a spol. [18], kteří navrhli a implementovali systém pro programování založený na ROS⁸ a MoveIt frameworku, který je schopný operovat i v plné virtualizaci a umožňuje plánovat trasu robotů pomocí detekce rukou (obrázek 2.2b).

2.5 Microsoft HoloLens 2

Microsoft HoloLens 2 jsou brýle pro rozšířenou realitu vyvinuté společností Microsoft. Na trh byly vydány 7. listopadu 2019 a pořizovací cena je 3500 dolarů. Jedná se o samostatně běžící počítač s přenosnou baterií a HMD využívající OST displej (viz sekce 2.3). HoloLens 2 běží na operačním systému Windows Holographic OS, který je založen na Windows 10.

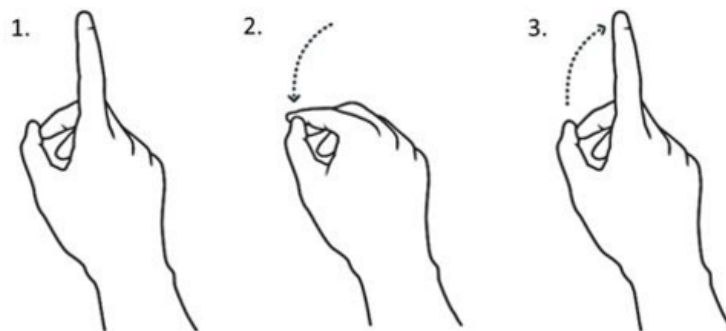
Hardware details

Pro správnou tvorbu hologramů a sledování polohy musí HoloLens 2 disponovat značným množstvím hardwaru. Nejdůležitějším prvkem je pravděpodobně druhá generace holografické jednotky (Holographic processing unit – HPU), což je na míru vytvořená procesní jednotka, která se stará o zpracování vstupů sensorů, jako jsou informace o pohybu, sledování očí a hloubkové vnímání prostoru. Cílem této jednotky je poskytnout vše potřebné pro běh AR aplikace a snímání vstupů od uživatele, s co nejmenším nárokem na energii. Značná část algoritmů na HPU využívá neuronové sítě (Deep Neural Network – DNN), především díky skvělým výsledkům, kterých DNN dosahují v oboru počítačového vidění.

Pro samotný běh aplikace je využit procesor Qualcomm Snapdragon 850, který je umístěn na zadní straně zařízení. Procesor využívá pouze pasivní chlazení a při běhu náročné aplikace může vydávat značné množství tepla a dokonce se i přehřát.

Pro vnímání prostoru a sledování reálných objektů jsou HoloLens 2 vybaveny RGB kamerou, hloubkovou kamerou, čtyřmi světelnými kamery a IMU. Pro sledování uživatele pohledu jsou využity 2 infračervené kamery. Zvuk je nahráván mikrofónovým polem

⁸Robot Operating System



Obrázek 2.3: Uživatelské gesto airtap, které se obvykle používá k výběru objektu. Převzato z https://www.researchgate.net/figure/Execution-of-the-air-tap-hand-gesture-with-the-Microsoft-HoloLens-20_fig1_344400783

s 5 kanály. Pro imerzivní zážitek zařízení také disponuje prostorovým zvukem. Zařízení je napájeno lithiovou baterií s výdrží 2 až 3 hodin aktivního používání [12].

Možnosti ovládání

HoloLens 2 poskytuje 3 hlavní možnosti interakce. Tyto možnosti zahrnují ovládání pomocí rukou a gest, ovládání přirozeným jazykem a ovládání pohledem.

Ovládání pomocí rukou a gest

Díky detekování plně artikulovaných rukou je možné s hologramy interagovat velmi přirozeným způsobem jako uchopení objektu či zmáčknutí tlačítka. Pro práci se vzdálenými objekty je možné využít ruční paprsek (hand ray), který vychází z dlaně ruky. Pro spuštění interakce s objektem pomocí paprsku je potřeba vykonat gesto *airtap* (spojení palce a ukazováčku viz 2.3). Kromě gesta *airtap* se nedoporučuje implementovat další gesta, především proto, aby se uživatel nemusel učit několik gest, ale zároveň, aby interakce v rámci všech aplikací byla unifikována. HoloLens 2 umožňuje i integraci s ovladači, které sice zvýší přesnost, ale značně omezí uživatele, jelikož nemá volné ruce [13].

Ovládání hlasem

V případě, že uživatel nemůže využít ruce, může využít hlasového ovládání, pomocí kterého by měl být schopen vykonat stejné operace jako při ovládání rukou. Systém poskytuje předdefinované povely jako „select“, „Move this“ a „Bigger/Smaller“ ale je možné vytvářet i vlastní povely [15].

Ovládání pohledem

Ovládání pohledem slouží spíše jako doplnění předešlých dvou způsobů interakce, kde pohledem je možné vybrat cíl a gestem či hlasovým příkazem spustit interakci. HoloLens 2 umožňuje i výběr objektu delším pohledem, ale pro plnou interakci s hologramy to není dostatečné [14].

Tvorba aplikací pro HoloLens 2

Jelikož brýle pro rozšířenou realitu jsou značně mladá platforma, stále se prozkoumává jak správně tvořit uživatelská rozhraní. Hlavní důraz bývá kladen na barvy, kde světlé barvy mohou být velmi nepříjemné na oči a proto se doporučuje používat bílou barvu pouze na text v uživatelském rozhraní [11]. Dalším velmi důležitým aspektem je zpětná vazba. Nejčastěji se využívá zvuk pro indikaci úspěšné interakce, ale je vhodné i zobrazovat zpětnou vazbu při přiblížení k objektu, například změnou stínování. Je taky nutné přemýšlet nad pozorností uživatele, není totiž vhodné, aby uživatel musel přesouvat pozornost od scény k UI, když UI může být přímo ve scéně.

Mixed Reality Toolkit 3 (MRTK3)⁹ je sada nástrojů a knihoven pro Unity¹⁰, které umožňují snadný a rychlý vývoj aplikací pro HoloLens 2, ale i pro jiné platformy. Rychlý vývoj je možný především díky simulaci uživatelského vstupu přímo v Unity.

Oproti předešlé verzi byla značná část MRTK přestavěna tak, aby více využívala už existující systémy jako OpenXR¹¹ a Unity XR Interaction Toolkit¹², díky čemu je možné využít předešlé zkušenosti z vývoje XR aplikací. Stejně jako u XR části aplikace i tvorba UI byla unifikována s tvorbou UI v unity pomocí Canvas UI¹³. Zároveň se vývojáři zaměřili na oddělení vzhledu a funkcionality, což umožňuje snadnou výměnu uživatelských prvků. Dalším důležitým cílem, byly optimalizace, aby i složitější aplikace mohly běžet plynule s 60 snímky za sekundu. Kromě technických vylepšení byla další výhodou dlouhodobá podpora, která však byla odvolána, kvůli přerušení vývoje HoloLens zařízení ze strany Microsoftu. Jediný vývoj, který lze očekávat je od komunitních příspěvků do veřejného repozitáře¹⁴, který ovšem bude pravděpodobně pomalejší.

⁹<https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk3-overview/>

¹⁰<https://unity.com/>

¹¹<https://www.khronos.org/openxr/>

¹²<https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0/manual/index.html>

¹³<https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UICanvas.html>

¹⁴<https://github.com/MixedRealityToolkit/MixedRealityToolkit-Unity>

Kapitola 3

Současné řešení

Jak již bylo předem zmíněno, tato práce se zabývá porovnáním použitelnosti tabletu a HoloLens 2 pro programování robotického pracoviště. V této kapitole budou popsány existující rozhraní pro tato zařízení. Jelikož tato práce spíše navazuje na řešení Simony Hladlovské [6] a v rámci práce budou prováděny změny pouze v HoloLens 2 rozhraní, bude toto rozhraní popsáno detailněji.

3.1 Reprezentace programu

Jelikož obě popisované řešení, slouží jako uživatelské rozhraní pro stejný server, které pomocí AR umožňují vizuálně programovat robotické pracoviště, bude zde popsána sdílená část těchto řešení, která se bude především zabývat reprezentací programu.

Jelikož je program tvořen v 3D prostoru, všechny jeho součásti jsou 3D objekty, které uživatel umísťuje do prostoru. Objekty jsou zároveň automaticky umísťovány do hierarchického systému tak, aby při pohybu s rodičovským objektem, byly posunuty i jeho potomci, díky čemuž je snazší dělat úpravy v úloze. Programování pracoviště je rozděleno na tvorbu scény a projektu.

Kalibrace aplikace

Pro správné umístění 3D objektů a možnosti uložení projektu je nutné aby virtuální objekty byly ukotvené k reálnému světu, pomocí kalibračního bodu.

V rozhraní pro tablet je při kalibraci vytvořena fotografie, která je odeslána na server, který v ní pomocí OpenCV hledá AR marker. Při jeho detekci je vypočítána jeho pozice vůči kameře, která je odeslána zpět klientovi. V rámci uživatelského rozhraní tabletu je kalibrace potvrzena zobrazením kalibrační kostky, podle které je zároveň možné sledovat, jak správně je zařízení kalibrováno

V rozhraní pro HoloLens 2 je detekován QR kód přímo v zařízení pomocí knihovny QRTracking¹. Detekce je potvrzena zobrazením virtuálního QR kódu ve scéně.

Scéna

Scéna je první část programu, kterou musí uživatel vytvořit. Slouží jako reprezentace fyzického robotického pracoviště, v rámci které jsou umístěni virtuální roboti a kolizní objekty

¹<https://github.com/yl-msft/QRTracking>

(základní geometrické tvary jako čtverec, koule), které vyznačují oblasti, do kterých se roboti nesmí pohnout, aby při spuštění programu nedošlo ke škodě.

Projekt

Projekt je úloha vytvořená pro robotické pracoviště. Je tedy propojena s jednou ze scén uložených na serveru. Úloha je sekvence, která má připomínat 3D vývojový diagram, který se skládá z akcí. Akce jsou šipky, které jsou propojeny do sekvence, díky orientaci šipky je možné poznat směr toku sekvence. Aby robot věděl kde danou akci vykonat, je akce umístována na akční bod, který reprezentuje bod v prostoru. Akční bod je reprezentován kuličkou. Akční bod zároveň může obsahovat orientaci, což je směr, ze kterého k němu budou roboti přistupovat. Akční body je možné manuálně vkládat do hierarchické struktury tak, aby závislosti posunu dávaly smysl. Zvláštním typem akcí jsou *start* (zelená šipka) a *stop* (červená šipka), které reprezentují začátek a konec programu. Tyto akce nepatří k akčnímu bodu a mají pouze výstup nebo vstup.

3.2 Rozhraní pro tablet

Rozhraní AREditor pro tablet bylo vyvinuto výzkumnou skupinou *Robo@FIT*², jako součást systému ARCOR2 [9] (Augmented Reality Collaborative Robot). Cílem tohoto systému bylo umožnit koncovému uživateli programování a interakci s robotem přímo v prostoru robotického pracoviště, což by usnadnilo programování robotů a ušetřilo uživatelům cesty a střídání pozornosti mezi počítačem a robotickým pracovištěm. Zároveň tato aplikace umožňuje vizualizaci naprogramované úlohy a sloučení více robotů pod jedno rozhraní. Nejdůležitější části uživatelského rozhraní jsou dvě menu. Levé menu umožňuje uživateli vybrat interakci s prostředím, ať už se jedná o posun, přidání či vymazání. Zatímco pravé menu slouží k výběru objektu v hierarchické struktuře programu.

Pravé menu

Jak již bylo řečeno, pravé menu (obrázek 3.1b) slouží k výběru objektu v prostoru (obrázek 3.3). Výběr objektu byl jeden z větších problémů, které museli autoři aplikace řešit a proto poskytli uživateli více možností.

Výchozí možností je výběr pomocí míření na objekt terčem, umístěným uprostřed obrazovky. Po namíření na daný objekt jsou v levém menu ponechány pouze ty objekty, na které uživatel míří s tím, že ten, na který míří přímo by případně byl využit jako cíl při vybrání interakce s objektem.

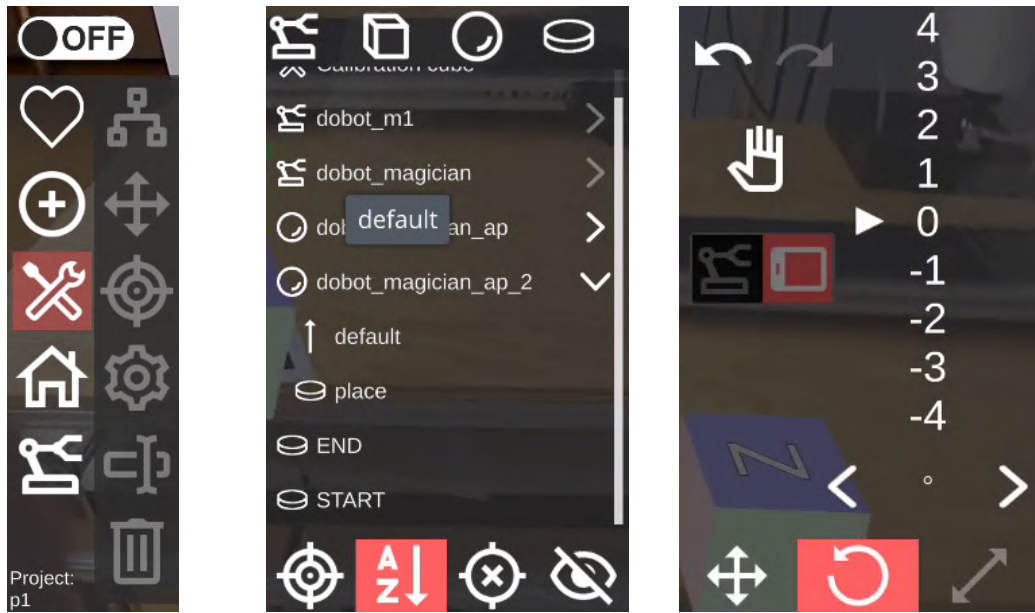
V případě velké hustoty objektů je možné v hierarchickém seznamu vyhledávat abecedně podle názvů objektů.

Zároveň je pravé menu po vybrání interakce nahrazeno a slouží jako uživatelův vstup dané interakce či nastavení parametrů ať už robota nebo akce (obrázek 3.1c).

Levé menu

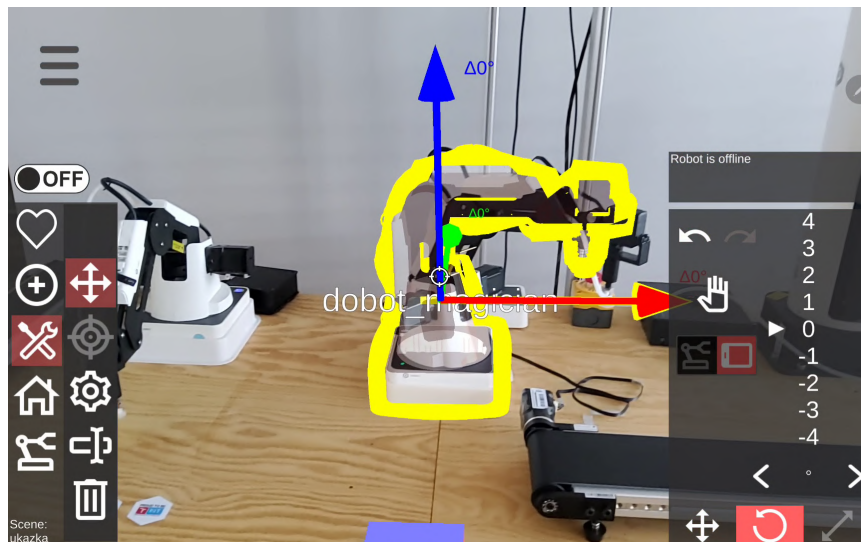
Levé menu slouží k výběru interakce s objekty, zároveň je zde tlačítko domů, které slouží k ukládání či přepínání scén a projektů. Mezi hlavní interakce, které může uživatel se scénou provádět patří: transformace, přidání objektů a přehrání programu.

²<https://www.fit.vut.cz/research/group/robo/.cs>

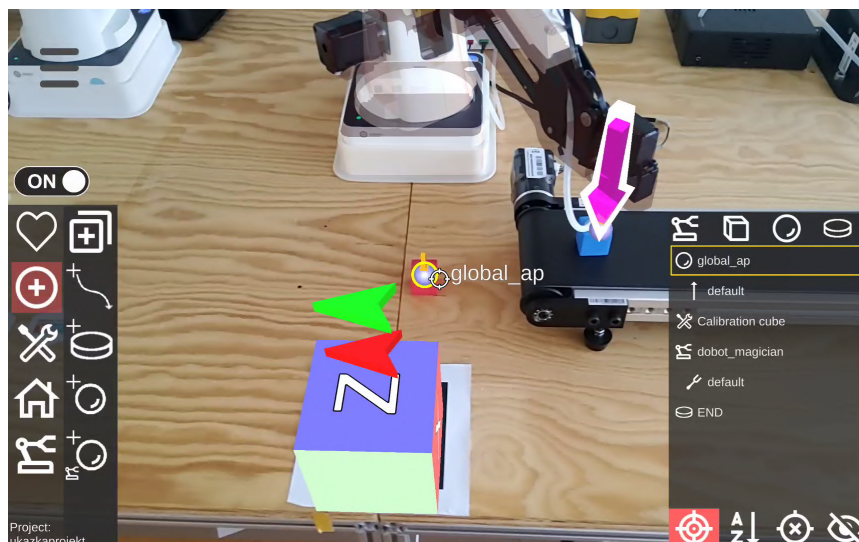


(a) Levé menu obsahující možné interakce s objekty ve scéně. (b) Pravé menu obsahující obecně seřazený seznam prvky 3D scény. (c) Pravé menu přepnuté do režimu transform. Posun je možný pomocí posunem číselnou řadou či uchopením pomocí tlačítka s ikonou ruky.

Obrázek 3.1: Ukázka pravého i levého menu v uživatelském rozhraní tabletu.



Obrázek 3.2: Transformace robota ve scéně. Robota je možné posouvat i otáčet podél barevných os.



Obrázek 3.3: Pohled na akční bod v rozhraní pro tablet. Pro přidání akce na tento bod je nutné ho vybrat mířením či kliknutím s pravém menu.

Transformace

Při výběru transformace se kolem označeného objektu vykreslí osy (obrázek 3.2), podél kterých je možné objekt posouvat či rotovat posunem číselné řady v pravém menu (obrázek 3.1c). Pro posun přes velké vzdálenosti je možné objekt uchopit tlačítkem s ikonou ruky a posouvat ho pohybem tabletu.

Přidání objektu

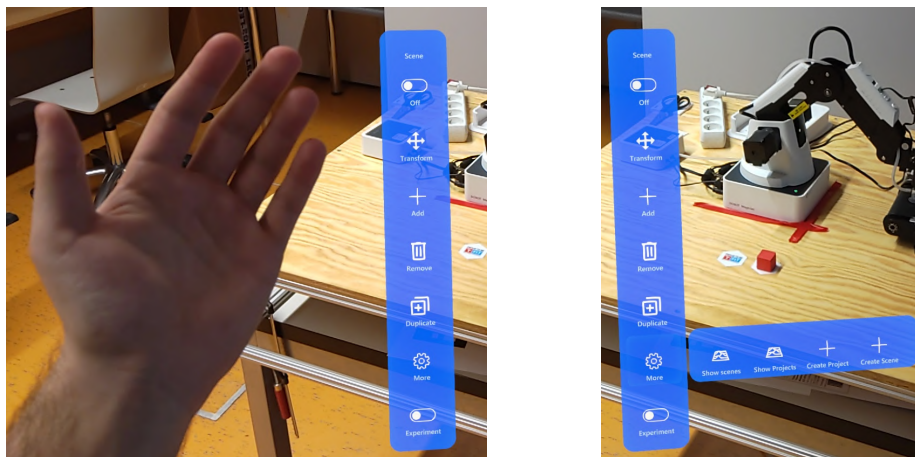
Objekty které může uživatel přidat jsou závislé na tom, zda je v editaci scény či projektu. Pokud existuje více možností přidaných objektů, může si uživatel vybrat ze seznamu v pravém menu. Při přidávání akčního bodu, může uživatel předem označit objekt, který bude hierarchickým rodičem akčního bodu. Akční bod může být přidán do prostoru před uživatelem, nebo pomocí aktuální pozice robota. Přidávání bodu pomocí robota zaručuje přesné umístění. Pro přidání akce je nutné vybrat akční bod na který bude akce umístěna (obrázek 3.3).

Přehrání programu

V tomto režimu může uživatel spustit celý program nebo přehrát pouze označenou akci. Pro spuštění této akce je nutné aby scéna byla v online režimu, při kterém si aplikace od serveru zjišťuje aktuální stav robotů.

3.3 Rozhraní pro HoloLens 2

Řešení S. Hiadlovské [6] implementuje základní funkcionalitu tabletového rozhraní AREditor a umožňuje základní programování robotů přes HoloLens 2. Ze základní funkcionality v řešení chybí přidávání scén, což znamená že před začátkem práce je nutné ji vytvořit přes rozhraní tabletu.



(a) Ruční menu, při pohledu na dlaň. (b) Ruční menu s otevřeným bočním menu.

Obrázek 3.4: Ruční menu, které se zobrazí při pohledu na dlaň ruky. V menu je přidáno tlačítko pro experiment, jelikož obrázky byly pořizovány až po experimentech.

V současném stavu je uživatelské rozhraní implementováno pomocí ručního menu, které se zobrazí při pohledu na dlaň (obrázek 3.4). Menu je velmi inspirováno levým menu v rozhraní tabletu, zatímco funkcionalitu pravého menu nebylo nutné implementovat, jelikož výběr objektů je implementován pomocí funkce airtap a nastavení parametrů není v tomto řešení implementováno.

Transformace

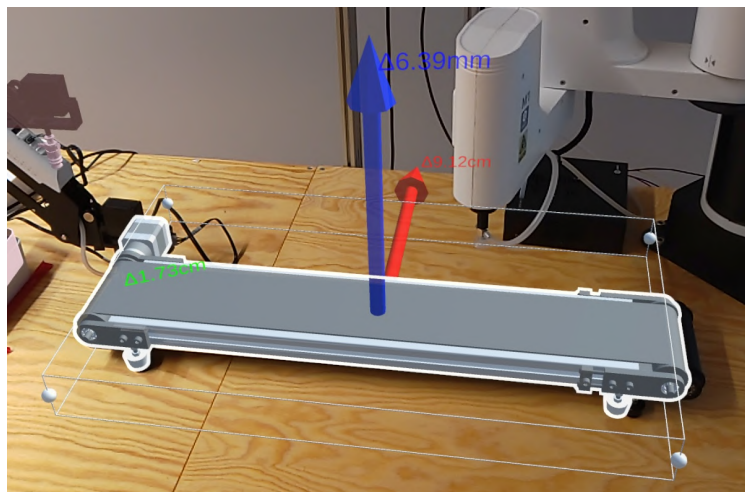
V tomto režimu je při použití funkce airtap objekt vybrán k transformaci. Takto vybraný objekt je ohraničen editační klecí, která umožní objekt uchopit a přesunout. Zároveň jsou na hranách klecí body, kterými lze při uchopení objekt rotovat. Důvod, který autorka udává pro tento systém je ten, že kdyby byla mřížka neustále okolo všech robotů, nebylo by možné uchopit objekt v blízkém prostoru robotů, jelikož by byly uvnitř jejich editační klece.

Spolu s klecí jsou zároveň zobrazeny osy, které je možné uchopit a manipulovat s objektem pouze podél vybrané osy. Důvod pro tento způsob pohybu, který autorka udává je, že při volném pohybu může být odezva příliš intenzivní, což ztěžuje přesné zarovnání. Editační klec i osy je možné vidět na obrázku 3.5.

Přidání objektu

Tato část menu poskytuje následující funkce:

- **Přidání robota a kolizního objektu** jsou implementovány jako samostatné akce, které jsou ukončeny po jednotlivých přidáních. To však způsobuje, že uživatelské rozhraní zůstává bez aktivního režimu, což může být pro uživatele neintuitivní, jak sama uznala i autorka původní práce. Pro výběr přidaného objektu byl implementován seznam, ve kterém může uživatel vidět model přidávaného objektu (obrázek 3.6).
- **Přidání akce** je opět implementováno jako režim tedy po úspěšném přidání je možné pokračovat v přidávání. Přidání akce se skládá z několika kroků. Prvně je umístěn akční bod pomocí funkce airtap, kde bod se vytvoří na místě ukazováčku po tom co



Obrázek 3.5: Transformační klec kolem robota spolu s osy. Robota je možné rotovat uchopením kuliček na hranách klece.

je airtap dokončen (prsty se již nedotýkají), zároveň si program zjistí, jestli airtap v moment puštění mířil na objekt scény a pokud ano, je přidáváný akční bod synem vybraného objektu v hierarchickém uspořádání scény. Místo umístění nového akčního bodu je možné přidávat akci na již existující tím, že uživatel vybere akci na již existujícím bodě. Po přidání akčního bodu, uživatel může gestem airtap vybrat robota, který má na akčním bodě vykonat akci, vybráním robota se naplní menu zobrazené u akčního bodu a uživatel si může vybrat akci.

- **Přidání spoje** je opět implementováno jako režim, ve kterém uživatel jednoduše vybere počátek spoje a konec spoje. Po výběru počátku je spoj veden na uživatelův ukazováček, pro indikaci stavu procesu přidávání spoje.

Duplikace

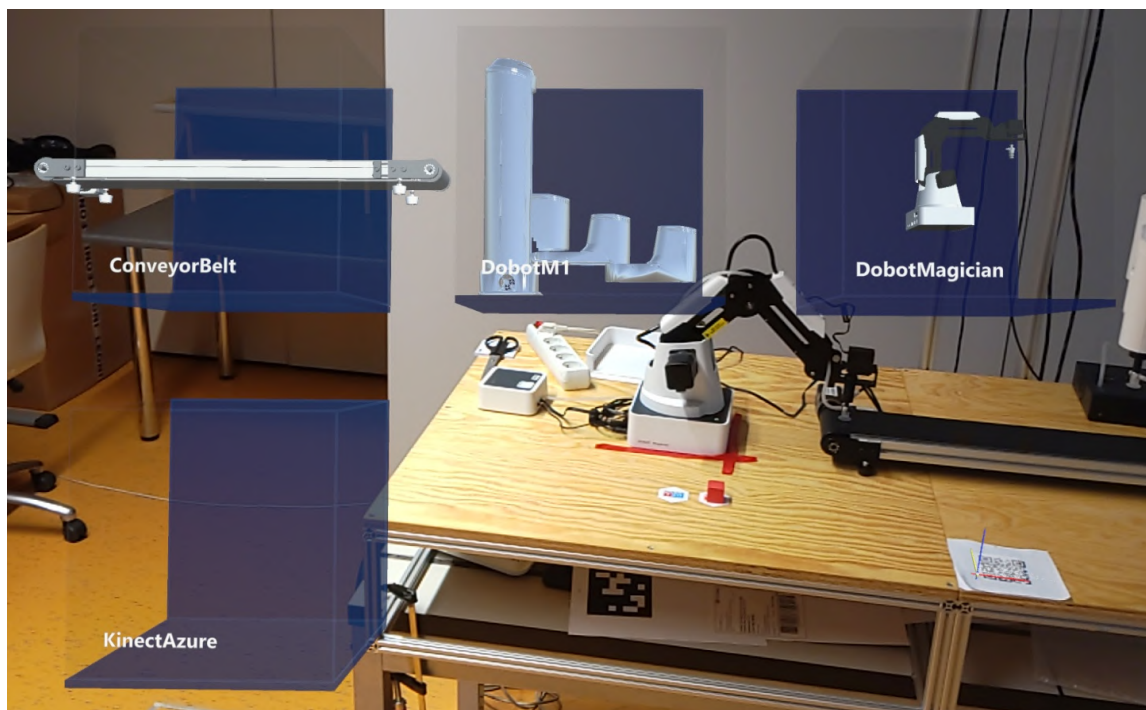
Duplikace je další režim interakce, při kterém je vybraný objekt zduplikován. Zduplikovaný objekt má stejnou polohu a tedy není vidět žádná změna ani potvrzení o duplikaci objektu.

Odstranění

Poslední režim interakce je odstranění. V tomto režimu je při výběru objektu zobrazeno menu, které žádá uživatele o potvrzení, což zabraňuje mylnému vymazání objektu. Pro odstranění spoje je potřeba označit akci, ze které spoj vychází, jelikož označení tenkého spoje by bylo obtížné.

Další funkcionalita

Poslední část menu je položka „More“ v menu, skrz které je možné prohlížet a otevírat scény či projekty, které jsou vizualizovány v seznamu, podobně jako při přidávání robotů. Zároveň je zde tlačítko pro vytvoření projektu. Scény jsou vizualizovány jako 3D kostky obsahující miniaturu celé scény



Obrázek 3.6: Menu pro výběr robota.

Chybějící funkcionality

Jak již bylo zmíněno autorka implementovala pouze základní část funkcí, které jsou implementovány v rozhraní tabletu. Mezi chybějící funkcionality patří:

- manipulace robota,
- úprava orientace akčního bodu,
- umístění bodu podle robota,
- vykonání programu,
- nastavení parametrů,
- přejmenování prvků scény.

Technická část implementace

Autorka původní práce rozšířila původní projekt AREditoru o novou scénu „HOLOGUI“, ze které je sestavena aplikace. V této scéně je vše potřebné pro běh aplikace na platformě HoloLens 2. Zároveň do projektu přidala balíčky potřebné k běhu aplikace na HoloLens 2. Takto tvořená aplikace směřovala na znovuvyužití skriptů z tabletu, jelikož velká část funkcionality se prolíná s rozhraním tabletu (komunikace se serverem, načítání modelů, správa scény). Bohužel se kvůli značné provázanosti těchto tříd s tabletovým rozhraním, rozhodla vytvořit kopie skriptů s písmenem “H” v názvu, aby nemusela refaktorovat kód, což zapříčinilo že tento kód nebyl udržován a změny provedené v tabletové verzi bylo nutné projít a implementovat v skriptech pro HoloLens 2 rozhraní.

Hlavní komponenty systému

Při programování v Unity se pro hlavní řízení systému používají manažeři, neboli třídy, které řídí tok programu, spravují objekty ve scéně, interakci s uživatelem, komunikaci se serverem a podobně. V rámci řešení autorka implementovala následující hlavní manažery:

- *GameManager*,
- *WebSocketManager*,
- *SceneManager*,
- *ProjectManager*,
- *ActionManager*,
- *SelectorManager*,
- *HandMenuManager*,
- *CalibrationManager*.

Mezi sekundární manažery, o jejichž řízení se stará *SelectorManger* patří *TranformManger*, *DeleteActionManager*, *AddActionPoint*, *ConnectionManager* a *AddActionPoint*

Velká část chování byla převzata z původního řešení a upravena tak, aby chování fungovalo s HoloLens 2 prvky. Zdrojové soubory obsahují značné množství zakomentovaného či mrtvého kódu.

Kapitola 4

Uživatelský experiment

Pro porovnání rozhraní a odhalení případných nedostatků je potřeba provést uživatelský experiment, který bude proveden na obou rozhraních. První iterace uživatelských testů bude sloužit k odhalení nedostatků v současné implementaci rozhraní pro HoloLens 2. Druhá iterace bude provedena po implementaci navržených změn a vylepšení. Bude tedy nejen možné pozorovat rozdíly mezi rozhraním tabletu a HoloLens 2, ale i mezi původní a vylepšenou verzí rozhraní pro HoloLens 2.

V této kapitole je popsán prováděný experiment, odhalené nedostatky v uživatelském rozhraní pro HoloLens 2 a výsledky první iterace testování. Výsledky druhé iterace a finální srovnání tabletu a HoloLens 2 jsou popsány v kapitole 7.

4.1 Existující studie

Práce, které porovnávají rozdíly mezi tabletem a HoloLens 1, již existují. Příkladem je například práce Riedlingera a spol. [22], ve které porovnávají tato zařízení při kolaborativním plánování interiéru a správě budovy. Existují také studie, které porovnávají tato zařízení v kontextu programování robotického pracoviště [5], ve kterých uživatelé plánují trajektorii robota pomocí bodů. Testovaný systém zároveň umožňuje vizualizovat pohyb a pozice robota v jednotlivých bodech trajektorie.

Výsledky těchto studií obvykle poukazují na větší efektivitu a lepší hodnocení uživatelské zkušenosti při využití HoloLens 1. Výhodou tabletu je, že jej uživatelé již umí ovládat a připadá jim přirozenější. Zároveň je tablet při spolupráci snazší na předání a umožňuje nahlédnutí dalšího uživatele na displej.

4.2 Popis experimentu

Uživatelský experiment byl cílen na porovnání rozhraní pro tablet a HoloLens 2 a z tohoto důvodu musel každý účastník vyzkoušet obě zařízení. Jelikož jsou si rozhraní velmi podobná, dalo by se očekávat, že při testování druhého rozhraní by účastníci dosáhli lepších výsledků. Z toho důvodu byli účastníci rozděleni na 2 skupiny, které určovaly, které rozhraní bylo testováno první (*counterbalanced testing*), kde skupina A začínala s HoloLens 2 a skupina B začínala s tabletem. Pro testování tabletu byl využit tablet Samsung Galaxy Tab S6, který byl vypůjčen od fakulty.

Experimenty byly se souhlasem účastníků nahrávány pro případnou analýzu problémů, na které uživatelé narazili. Nahrávání na HoloLens 2 bylo obtížné, jelikož brýle neměly do-

statek volného místa v úložišti pro takto dlouhý záznam. Zároveň by nebylo možné sledovat pohled uživatele přes *Live stream*¹, jelikož se tyto funkce vzájemně vylučují. Z těchto důvodů bylo nutné nahrávat live stream v Microsoft HoloLens aplikaci² na počítači přes *OBS Studio*³.

Úlohy experimentu

V rámci experimentu museli uživatelé vykonat 4 úlohy, které byly částečně inspirovány experimentem z původní práce S. Hiadlovské [6]. Původní experiment totiž vhodně pokryl celý proces programování robotického pracoviště a zároveň v něm uživatelé vyzkoušeli většinu funkcí rozhraní. Důvodem pro rozdělení experimentu na menší části bylo měření metrik při interakci s jednotlivými částmi rozhraní.

Cílem změn v programované úloze bylo zmenšit časovou náročnost a proto byl zmenšen počet akcí v programu, jelikož se jedná o opakování stejné interakce s rozhraním. Zároveň bylo cílem zvýšit hustotu objektů v jednom místě (*object clutter*), aby bylo možné sledovat, ve kterém rozhraní je uživateli příjemnější pracovat s takovou scénou. Poslední změnou bylo po uživateli požadovat umístění více akcí na jeden akční bod, jelikož se jedná o běžnou interakci s rozhraním.

1. Úloha – Přidání robotů

V tomto úkolu je po účastníkovi žádáno, aby přidal všechny roboty, které jsou na fyzickém pracovišti. Jedná se tedy o Dobot Magician, Dobot M1 a dopravníkového pásu. V této úloze uživatel roboty neumísťuje i přes to, že přirozený případ užití je přidat robota a ihned ho umístit. Tyto funkce jsou v úlohách oddělené, aby bylo možné sledovat efektivitu uživatelů při jejich používání.

2. Úloha – Umístění robotů

Po přidání virtuálních robotů je účastník požádán o jejich umístění. V této úloze bude měřen čas a přesnost zarovnání robotů.

3. Úloha – Přidání akcí

V této úloze je účastník požádán o přidání akcí tak, aby robot Dobot Magician zdvihl objekt z pásu položil ho na jednu z nálepek na stole, zdvihl další objekt z pásu a položil ho a druhou nálepkou na stole. Jak již bylo zmíněno výše, tato úloha směřovala na testování schopnosti uživatele pracovat se scénou, ve které je velké množství objektů na jednom místě, a zároveň na otestování funkce přidání více akcí na jeden akční bod. Scénu po vykonání úlohy je možné vidět na obrázku 4.1a.

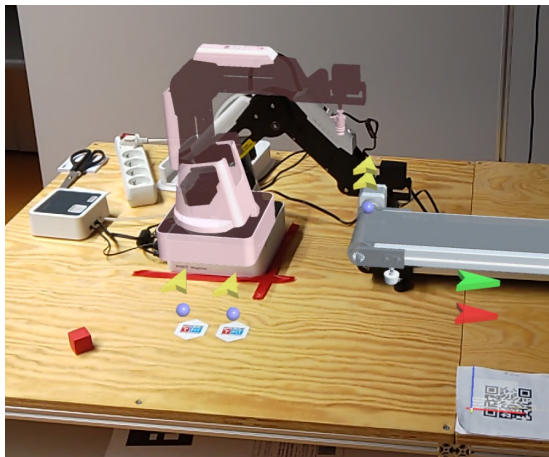
4. Úloha – Propojení akcí do sekvence

Finální úlohou experimentu je propojení akcí do sekvence. V této úloze je porovnávána především schopnost uživatele vybírat objekt ve scéně. Podobně jako ve 3. úloze je testována uživatelská schopnost výběru ve scéně s mnoha objekty. Scénu po vykonání úlohy je možné vidět na obrázku 4.1b.

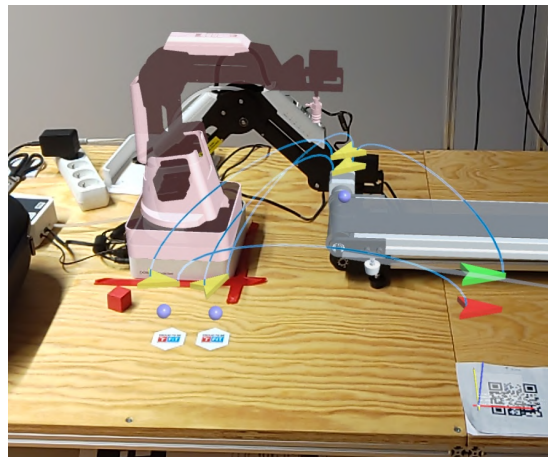
¹<https://learn.microsoft.com/en-us/hololens/holographic-photos-and-videos>

²<https://apps.microsoft.com/detail/9NBLGGH4QWNX?hl=en-US&gl=US>

³<https://obsproject.com/cs>



(a) Pohled na dokončenou 3. úlohu



(b) Pohled na dokončenou 4. úlohu

Obrázek 4.1: Pohledy na dokončené 3. a 4. úlohy.

Měřené metriky

V rámci experimentu bylo měřeno několik objektivních metrik. Všechny měřené metrik jsou zařízením měřené numerické hodnoty, které je možné snadno porovnat. Původně mezi metrikami byl i počet dotazů a chybné akce, ale tyto metriky by bylo velmi obtížné správně měřit. Spolu s objektivními metrikami bylo měřeno i subjektivní hodnocení rozhraní uživatelem pomocí dotazníků, které jsou popsány v sekci 4.2.

Pro přesné měření metrik byla obě rozhraní rozšířena o funkcionalitu, která měří metriky přímo v zařízení při průběhu experimentu. Toto rozšíření umožní v rozhraní spustit a ukončit experiment. V době trvání experimentu jsou v zařízení měřeny tyto údaje:

- **Čas** – Základní metrika, která nejvíce vypovídá o efektivitě uživatelského rozhraní. Jelikož účastníci experimentu viděli systém poprvé, bude tato metrika vypovídat i o tom, jak je systém intuitivní. Ideální by bylo, kdyby uživatelé mohli experiment vykonávat opakovaně, čímž by bylo možné sledovat efektivitu a intuitivnost rozhraní separátně, ale to by experiment značně prodloužilo a bylo by náročné najít dobrovolníky, kteří by byli ochotni obětovat tolik volného času.
- **Vzdálenost pohybu kamery** – Hlavní myšlenka této metriky spočívá v porovnání fyzické náročnosti při používání tabletu a HoloLens2. Jelikož by měření nachozené vzdálenosti bylo velmi náročné a nebylo by možné jí měřit ze samotné aplikace, bude měřena pouze vzdálenost pohybu kamery v aplikaci.
- **Přesnost zarovnání robotů** – V rámci experimentů bude porovnána schopnost uživatele přesně umístit roboty ve scéně. Toto bude možné díky změření fyzického robotického pracoviště a následného výpočtu vzdálenosti robotů od středu QR kódu. Takto bylo možné získat očekávané umístění virtuálních robotů ve scéně, ve které je QR kód počátkem scény. Pro získání přesnosti stačí porovnat takto získanou pozici s uživatelem umístěným virtuálním robotem. Tato metrika by měla sledovat uživatelské vnímání hloubky a schopnost umístění robota, zároveň sleduje schopnost kalibrace a sledování pohybu daného zařízení. Bohužel pravděpodobně není možné oddělit chybu sledování pozice zařízení od chyby uživatele při umísťování robotů, nicméně

tato metrika stále poskytne možnost porovnat zařízení z pohledu umístění objektů ve scéně, která má být propojena s fyzickým světem.

Hypotézy

V této části bude popsáno několik předpokladů, které budou testovány v rámci experimentů.

Hypotéza 1: HoloLens 2 umožní rychlejší vykonání úloh.

HoloLens 2 poskytují přirozenější způsob interakce, kterou je obvykle možné provádět rychleji, než na tabletu. Lze tedy očekávat že při využití HoloLens 2 budou dosahovat uživatelé obecně lepších časů.

Výjimkou z této hypotézy je první úloha, jelikož se menu pro výběr robata v HoloLens 2 rozhraní chová velmi uživatelsky nepřívětivě tím, že se pohybuje podle uživatelovy hlavy a jednotlivé bloky se otáčejí k uživateli. V této úloze je možné očekávat značně horší čas při využití HoloLens 2.

Hypotéza 2: HoloLens 2 umožní přesnější anotaci pracoviště

Díky stereoskopické projekci obrazu by HoloLens 2 měly poskytovat značně lepší vnímání hloubky. V kombinaci se stabilnější kalibrací scény je možné předpokládat, že při zarovnávání robotů dosáhnou uživatelé s tímto zařízením lepšího zarovnání robotů.

Hypotéza 3: HoloLens 2 poskytne rychlejší způsob výběru.

Čtvrtá úloha se především zabývá opakovaným vybíráním objektů. Výběr objektů pomocí paprsku v HoloLens 2 působí přirozeněji a efektivněji a je možné předpokládat že tuto úlohu budou uživatelé zvládat rychleji. Ovšem s narůstajícím počtem objektů ve scéně může být míření paprskem náročnější a méně efektivní, což by mohlo vést k tomu, že hypotéza nebude potvrzena.

Hypotéza 4: Práce s tabletem vyžaduje více pohybu po pracovišti.

Je možné očekávat, že při práci s tabletem, bude naměřená nachozená vzdálenost mnohem vyšší, jelikož kvůli omezenému vnímání hloubky musí uživatel objekty pozorovat z více stran, aby získal přehled o jejich umístění. Takovéto získávání informace o hloubce se nazývá „motion parallax“ [4].

Hypotéza 5: Práce s tabletem bude dosahovat vyšší fyzické náročnosti

Více pohybu kolem pracoviště může být pro uživatele náročnější, což bude možné sledovat i v subjektivním hodnocení fyzické náročnosti.

Dotazníky

Pro měření uživatelova subjektivního vnímání byly využity 3 standardizované dotazníky. Všechny dotazníky byly zahrnuty v jednom online dotazníku, který byl vytvořen v *Google Forms*⁴. Tento dotazník mi byl poskytnut vedoucím práce a obsahoval následující 3 stan-

⁴<https://www.google.com/forms/about/>

dardizované dotazníky a sadu demografických otázek (věk, dosažené vzdělání, zkušenost s programováním, atd.)

System Usability Scale (SUS) je dotazník, který vývojáři používají pro rychlé ohodnocení použitelnosti systémů, aplikací nebo webových stránek. Dotazník se skládá z deseti otázek, z nichž polovina je pozitivně formulovaná a druhá polovina negativně. Uživatelé odpovídají na otázky na škále od 1 do 5, kde 1 znamená „silně nesouhlasím“ a 5 znamená „silně souhlasím“. Následně se získané hodnoty zpracovávají a výsledkem je číslo (SUS skóre), které udává míru celkové použitelnosti systému [3].

Pro výpočet SUS skóre stačí sečíst hodnoty odpovědí podle rovnice (4.1), kde každá z otázek může do sumy přispět hodnotou 0 až 4. Výsledné SUS skóre je hodnota v rozmezí 0 až 100. V rámci 500 studií, které využívali SUS, vyšlo průměrné skóre 68 a tato hodnota se začala používat jako hranice průměrné použitelnosti [7].

$$SUS_{score} = 2,5 * \left(\sum_{i \in \{1,3,5,7,9\}} (x_i - 1) + \sum_{i \in \{2,4,6,8,10\}} (5 - x_i) \right) \quad (4.1)$$

Od vytvoření SUS, byla validita výsledků této metriky zpochybňována, jelikož se jedná o velmi jednoduchý dotazník. Studie však potvrdily korelaci mezi úspěšností plnění úkolů v systému a SUS skóre. Přesto by mělo být SUS skóre využíváno spíše k ordinálnímu porovnání systémů [19].

NASA Task Load Index (NASA-TLX) je dotazník, který se zaměřuje na měření subjektivního vnímání zátěže uživatelem. V dotazníku je uživatel požádán o ohodnocení několika faktorů zátěže na stupnici od 1 do 20. Mezi tyto faktory patří psychická náročnost, fyzická náročnost, časová náročnost, úspěšnost, potřebná snaha a úroveň stresu. Tyto hodnoty lze následně zprůměrovat k získání finální hodnoty, nazývané Task Load Index (TLX). Zároveň je při výpočtu možné jednotlivé faktory násobit váhami. Tyto váhy se nejčastěji opět získávají od uživatelů tím, že z 15 párů kategorií zátěže uživatel vybírá tu, která měla vyšší podíl na jeho zátěži [21].

User Experience Questionnaire (UEQ) se zaměřuje měření uživatelské zkušenosti, při využívání systému. Dotazník obsahuje 26 otázek, které měří 6 aspektů aplikace, mezi které patří:

- **Atraktivita** (Attractiveness) – Popisuje celkový dojem z produktu, tedy nejen vizuální stránku.
- **Prozíravost** (Perspicuity) – Také by se dala označit jako intuitivnost či srozumitelnost. Popisuje jak snadné je se seznámit se systémem a jeho ovládáním.
- **Efektivita** (Efficiency) – Popisuje schopnost uživatele plnit úkoly v systému rychle a bez zbytečného usílí.
- **Spolehlivost** (Dependability) – Popisuje, jak moc si je uživatel jistý při ovládání systému.
- **Stimulace** (Stimulation) – Popisuje schopnost systému udržet uživatelovu pozornost a vzbuzovat v něm zájem.
- **Inovace** (Novelty) – Posuzuje, nakolik je systém vnímán jako inovativní a originální.

Tyto aspekty je možné sloučit do 3 skupin. Atraktivita je samostatná skupina. Druhá skupina jsou **pragmatické aspekty** do které patří prozíravost, efektivnost a spolehlivost.

Poslední skupina se nazývá **hedonická** (zabývá se emocemi a subjektivním pohodlím uživatele) a patří do ní stimulace a originalita [24].

Stejně jako u předešlých dotazníků se vyhodnocení UEQ provádí porovnáním s ostatními aplikacemi. Za tímto účelem byl vytvořen benchmark, v rámci kterého bylo sesbíráno velké množství UEQ hodnocení různých aplikací, se kterými je možné vlastní hodnoty porovnat. Obvyklou slabinou těchto benchmarků je, že nerozlišují mezi typy aplikací a tedy porovnání běžných aplikací a webových stránek s aplikací pro HoloLens 2 nemusí být vypovídající [25].

Průběh experimentu

Před samotným experimentem byl účastníkovi představen cíl výzkumu *Robo@Fit* a motivace pro programování koncovým uživatelem. Následně byl účastník obeznámen s rozhraním a byly mu představeny všechny potřebné funkce. Účastník si během představování rozhraní funkce rovnou vyzkoušel. V této části experimentu nebyl měřen čas a účastník měl možnost poskytnout zpětnou vazbu na uživatelské rozhraní a případně klást dotazy. Zároveň bylo možné nechat účastníky, aby se s některými funkcemi seznámili sami a sledovat, jak je rozhraní či postup funkce intuitivní.

Jelikož většina účastníků využívala HoloLens 2 poprvé, bylo nutné, aby si před zkoušením rozhraní prošli tutoriál *HoloLens Tips*⁵, který seznámí uživatele se základy manipulování s hologramy v HoloLens 2 a především seznámí uživatele s funkcí airtap.

Po tomto úvodu byly účastníkům postupně představeny úlohy popsané v sekci 4.2, které následně vykonali. Při každé úloze bylo nutné spustit experiment na daném zařízení, v rámci kterého se měří metriky zmíněné v sekci 4.2. Při testování rozhraní tabletu bylo možné účastníkovi zasáhnout do rozhraní a spustit experiment za něj, ale při testování rozhraní HoloLens 2 bylo nutné, aby účastník spouštěl a vypínal experiment sám. Z toho důvodu je možné, že měřený čas může zahrnovat chybu vytvořenou opožděným vypnutím experimentu. Nemělo by se však jednat o chybu větší jak pět sekund.

Po dokončení všech úloh na jednom z rozhraní byli účastníci požádáni o vyplnění dotazníků popsaných v sekci 4.2, které budou sloužit k porovnání rozhraní.

4.3 Výsledky 1. iterace uživatelských testů

V první iteraci bylo uživatelské testování zaměřeno především na odhalení nedostatků v uživatelském rozhraní pro HoloLens 2. Z tohoto důvodu byla rozhraní testována pouze čtyřmi účastníky (tabulka 4.1). Jelikož se výsledky mezi skupinami významně nelišily, byly hodnoty mezi skupinami pouze zprůměrovány. Zároveň je při takto malém počtu účastníků možné, že data nebudou dostatečně reprezentovat skutečnost. Současně výsledky při testování HoloLens 2 mohou být mírně horší, jelikož účastníci narazili na technické problémy či chyby v rozhraní, které jim bránily v konání úlohy.

Objektivní data

Při měření objektivních dat se potvrdila většina předpokladů, ale v některých případech byly naměřeny překvapivé hodnoty a ukázalo se, že při využití HoloLens 2 uživatelé nejsou výrazně rychlejší než s tabletem. Co se týče časů potřebných na vykonání úloh, které je možné vidět v grafu na obrázku 4.3, první úloha podle očekávání trvala značně delší dobu při využití HoloLens 2. Druhá úloha překvapivě trvala téměř stejně dlouho na obou zařízeních,

⁵<https://www.microsoft.com/en-us/p/hololens-tips/9pd4cxkklc47>

Účastník	Věk	Pohlaví	Vzdělání	Skupina
A	23	muž	VŠ – bakalářské	A
B	23	muž	VŠ – bakalářské	B
C	22	žena	SŠ	A
D	24	muž	VŠ – bakalářské	B

Tabulka 4.1: Demografické údaje účastníků 1. experimentu.

Zařízení	Chyba pozice	Chyba rotace
HoloLens 2	2,40 cm	1°42'
Tablet	6,97 cm	0°35'

Tabulka 4.2: Průměrné hodnoty chyb při anotaci robotického pracoviště. Chyba byla měřena jako vzdálenost od předem naměřené pozice fyzického robota.

ale uživatelé dosáhli menší chyby zarovnání při využití HoloLens 2 (tabulka 4.2). Zajímavé je, že uživatelé byli schopni přesněji natočit virtuální roboty v rozhraní tabletu. Důvodem může být, že rozhraní pro HoloLens 2 postrádá možnost přesné rotace. Třetí úloha dopadla podle očekávání a HoloLens 2 byly rychlejší. Ve čtvrté úloze uživatelé překvapivě dosáhli velmi podobných časů, které jsou ale spíše způsobeny problémy s detekcí gesta airtap a chybě v umístování akcí nad akčními body.

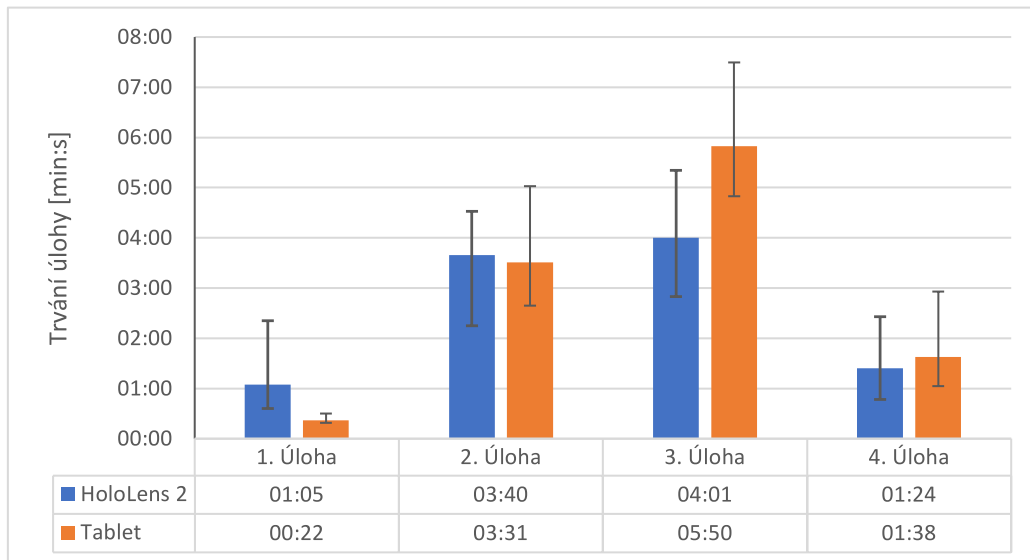
Co se týče naměřených vzdáleností, ukázalo se, že někteří účastníci cítili potřebu se po robotickém pracovišti značně pohybovat i při využití HoloLens 2, ale někteří uživatelé zvládli splnit úlohu se značně menším pohybem. Toto je možné vidět na grafu 4.3, kde sloupce s hodnotami pro HoloLens 2 mají větší rozdíly mezi extrémy.

Subjektivní hodnocení účastníků

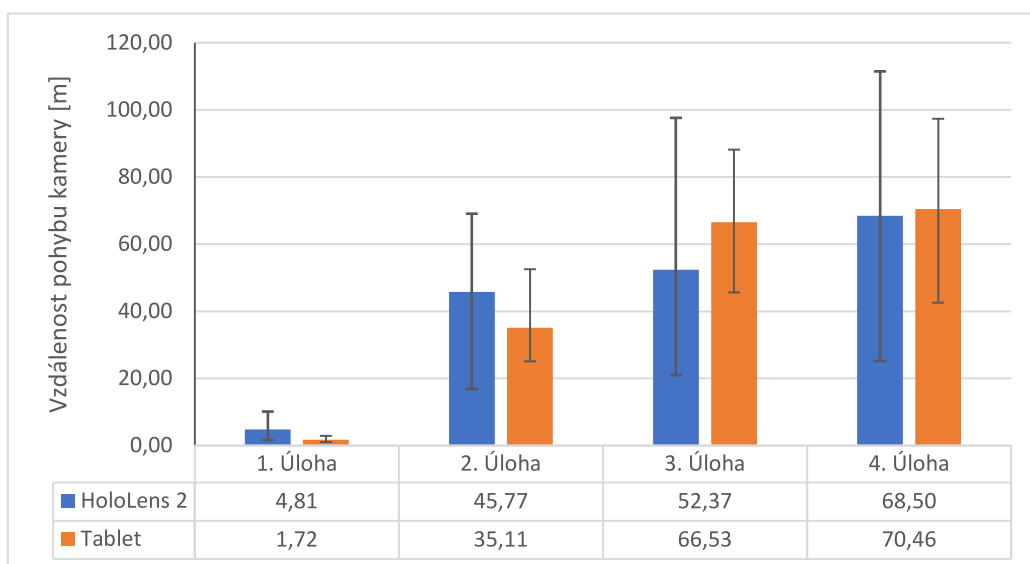
Subjektivní hodnocení bylo od účastníků sbíráno pomocí SUS, UEQ a NASA-TLX dotazníků popsanych v sekci 4.2. Účastníci experimentu byli v dotaznících velmi kritičtí, ale většinou hodnotili rozhraní podobně a je možné sledovat mírné rozdíly mezi tabletem a HoloLens 2.

Na výsledcích SUS skóre (tabulka 4.3) je možné vidět, že hodnocení účastníků má velký rozsah, ale většinou jeden účastník hodnotil obě rozhraní podobně s tím, že HoloLens 2 bylo hodnoceno lépe. Výjimkou je účastník D, který při konání úloh narazil na spoustu chyb v aplikaci a zároveň se mu nedařilo provádět airtap gesto tak, aby ho HoloLens 2 stabilně zachytávaly. Kvůli chybě v aplikaci dokonce musel jednu úlohu opakovat a kvůli delšímu trvání experimentu se HoloLens 2 přehřály, což mohlo být frustrující. V rozhraní tabletu uživatelé taky narazili na chyby v aplikaci, jako jsou zámky pro práci ve více lidech, které účastníkům bezdůvodně bránily vykonávat úlohu, ale tyto problémy nebyly příliš závažné. Z těchto důvodů může být SUS skóre nižší, a tedy ani jedno rozhraní nedosáhlo na průměrné hodnocení 68.

Náročnost hodnotili účastníci u obou rozhraní podobně (obrázek 4.4). Zajímavé je, že u HoloLens 2 uváděli mnohem větší časovou náročnost a menší úspěšnost i přesto, že jim vykonání úloh trvalo podobnou nebo i kratší dobu (obrázek 4.2). Pravděpodobně je toto hodnocení způsobeno tím, že se účastníci při využívání HoloLens 2 zasekli na jedné z operací a připadali si neefektivní a neúspěšní. Fyzická náročnost byla podle očekávání větší při využívání tabletu, ale rozdíl nebyl vysoký. Pravděpodobně je důvodem to, že účastníci i při



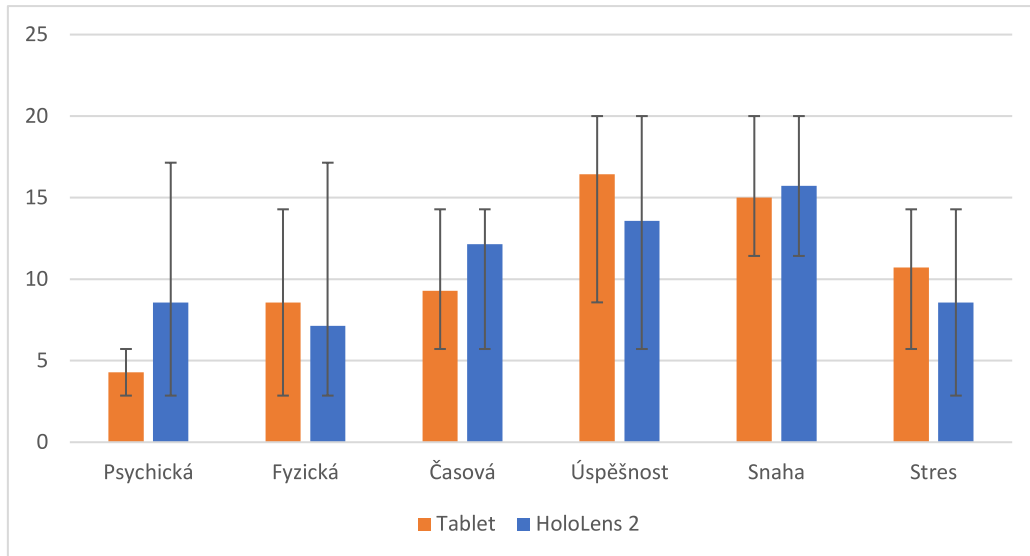
Obrázek 4.2: Graf zobrazující průměrné časy vykonání jednotlivých úloh. Pod grafem je tabulka s přesnými hodnotami průměrů. Sloupce obsahují indikátor extrémů a tedy i rozsah naměřených hodnot.



Obrázek 4.3: Graf zobrazující průměrné vzdálenosti pohybu kamery při jednotlivých úlohách. Pod grafem jsou přesné hodnoty průměrů. Sloupce obsahují indikátor extrémů a tedy i rozsah naměřených hodnot.

Účastník	SUS – Tablet	SUS – HoloLens 2
A	87,5	90
B	47,5	52,5
C	30	42,5
D	90	45
Průměr	63,75	55,625

Tabulka 4.3: Tabulka obsahující jednotlivá SUS skóre pro obě zařízení od jednotlivých účastníků.



Obrázek 4.4: Graf zobrazující průměrné hodnoty jednotlivých zátěží měřených v rámci NASA-TLX dotazníku. Úspěšnost byla invertována kvůli překladu, a tedy je opakem zátěže.

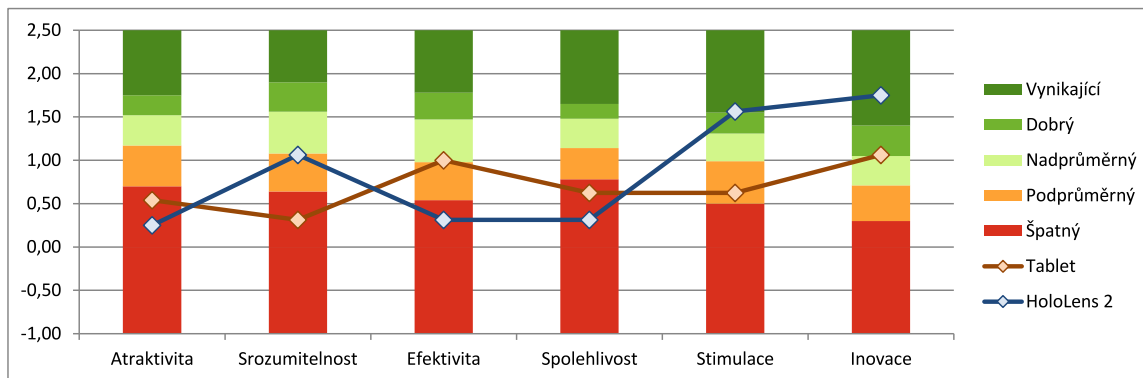
využívání HoloLens 2 nachodili značnou vzdálenost pro získání přehledu o scéně, jak je možné vidět z grafu na obrázku 4.3.

Z výsledků UEQ benchmarku (obrázek 4.5) je možné vidět, že subjektivní hodnocení účastníků bylo dost kritické a ve většině kategorií jsou rozhraní pod průměrným hodnocením. Zajímavé je, že v prvních čtyřech kategoriích jsou výsledky opačné k výsledkům v práci Simony Hiadlovské [6]. Pravděpodobně se jedná o chybu způsobenou malým počtem účastníků.

Je možné, že srozumitelnost rozhraní pro HoloLens 2 je větší, protože obsahuje méně prvků a kromě přidávání akcí jsou všechny funkce velmi intuitivní. Nízká spolehlivost a efektivita jsou pravděpodobně způsobeny chybami v rozhraní pro HoloLens 2, které uživatele frustrovaly a připadali si neefektivní. Nízká efektivita může také být spojena s tím, že většina uživatelů měla problém s detekcí gesta airtap. Vysoká stimulace a inovace byla u HoloLens 2 očekávána, jelikož se jedná o novou technologii a účastníci si jí velmi oblíbili.

4.4 Odhalené nedostatky

V rámci experimentů bylo nalezeno značné množství nedostatků v uživatelském rozhraní pro HoloLens 2. V této části budou kategorizovány a stručně popsány podle funkce rozhraní.



Obrázek 4.5: UEQ benchmark porovnávající průměrné hodnocení tabletu a HoloLens 2 v jednotlivých kategoriích, spolu s dalšími produkty, podle kterých byly vytvořeny oblasti hodnocení.

V průběhu experimentů byla získána i zpětná vazba k rozhraní tabletu a bylo odhaleno několik nedostatků. Nedostatků v rozhraní pro tablet bylo značně méně a většina problémů pramenila z vybírání a označování objektů v pravém menu. Tyto poznatky byly pouze zaznamenány a předány vedoucímu práce.

Změna scény/projektu

Po vybrání možnosti změny scény či projektu se uživatel vykreslí seznam tvořený bloky, které obsahují vizualizaci scény nebo projektu. Problém s tímto seznamem, na který narazila většina účastníků, je ten, že se seznam vykreslí před uživatelem. Jelikož účastník vyvolává tento seznam z ručního menu, na které se většinou kouká se sklopenou a mírně nakloněnou hlavou, je menu umístěno velmi nízko a pod úhlem. Zároveň se seznam v takovémto případě vykreslilo ve stole, což znamenalo, že uživatel hůře viděl obsah scén či projektů. Zároveň se delší názvy scén či projektů mohou překrývat se sousedními názvy.

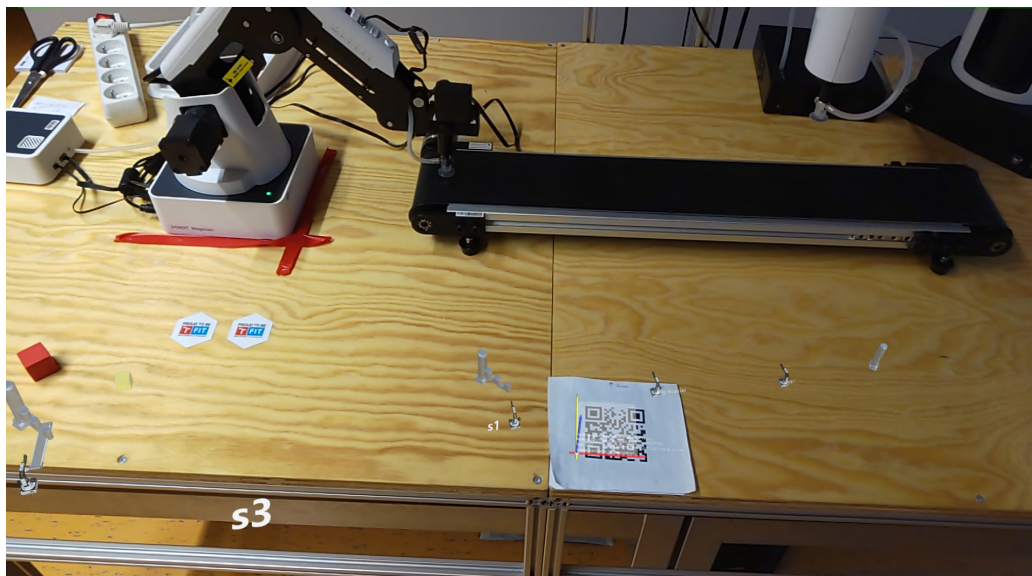
Přidání objektů

Podobně jako při výběru scén je i při přidávání robotů uživateli zobrazeno menu skládající se z bloků obsahujících modely objektů, které lze přidat. Na rozdíl od výběru scén je toto menu vykresleno před uživatelem a posouvá se společně s pohybem jeho hlavy. Současně jsou jednotlivé bloky natáčeny směrem k uživateli. Toto však většině uživatelů připadalo nepříjemné, jelikož pro ně bylo přirozené otočit hlavu směrem na robota, kterého chtěli vybrat, což způsobilo posunutí celého menu. Zároveň není vždy možné vidět celé menu, jelikož je příliš velké a umístěné blízko uživateli, což způsobí, že jeho kraje nejsou vykreslovány kvůli omezenému pokrytí zorného pole brýlí HoloLens 2.

Transformace

V režimu transformace účastníci našli několik problémů. Většina z nich pramenila z uchopitelných os, které slouží k přesné editaci. Mezi tyto problémy patří:

1. **Chybné uchopení objektu místo osy** – Pouze dva z účastníků mylně uchopili celý objekt místo osy, ale z pohledu uživatelského rozhraní se jedná o závažný nedostatek, jelikož při takovémto uchopení uživatel ztratí předešlé zarovnání podél os.

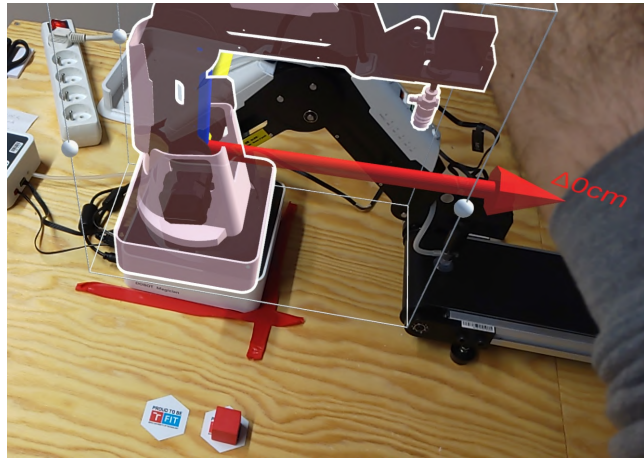
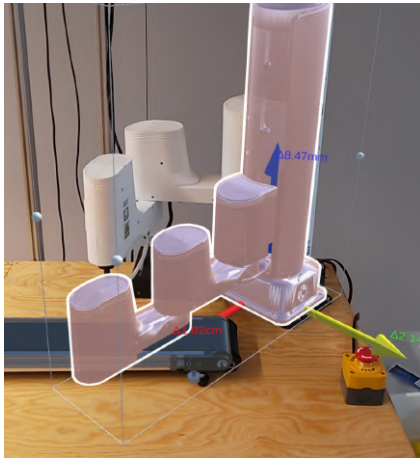


Obrázek 4.6: Seznam pro výběr scény s vizualizací scény v tlačítkách. Seznam je špatně viditelný jelikož je ve stole.

2. **Nepřístupná osa** – Jelikož jsou osy vykreslovány ze středu robota, může se stát, že celá osa bude uvnitř editační kostry robota (obrázek 4.7a) za předpokladu, že je robot dostatečně velký. Takto vykreslenou osu není možné uchopit rukou ani gestem airtap, a není tedy možné využít její funkcionalitu a zároveň uživatel může při pokusu o uchopení takovéto osy uchopit celého robota, čímž přijde o současné zarovnání.
3. **Pozice osy** – Jak již bylo zmíněno výše, osy jsou vykreslovány ze středu robota a míří v kladném směru v souřadnicovém systému, což znamená, že pro posun podél osy Y (nahoru/dolů), musí uživatel uchopit osu nad robotem. Bohužel v tomto případě užítí chce uživatel většinou zarovnávat spodek robota tak, aby seděl na stole, jenže při pohledu na spodek robota HoloLens 2 nemohou nadále vidět ruku, kterou byla osa uchopena, a uživatelův posun není registrován (obrázek 4.7b).
4. **Dvou-kroková editace** – Několikrát se při experimentu stalo, že se účastník pokusil robota uchopit a rovnou posouvat. Nejedná se o velký nedostatek, ale z chování účastníku bylo znát, že je pro ně přirozenější přímá editace bez předešlého výběru objektu.
5. **Transformace akčního bodu** – Pro vyvolání transformace akčního bodu musí uživatel nejdříve vybrat akci nad akčním bodem, a až poté může uchopit editační klec kolem akčního bodu. Většina účastníků se snažila vyvolat transformaci bodu výběrem akčního bodu i přes předešlé vysvětlení této funkce. Zároveň jim dělalo problém namířit paprskem na akční bod, jelikož má malý hitbox a zároveň jsou nad ním akce, které brání ve výběru bodu.

Přidání akce

V této funkci bylo taktéž objeveno několik nedostatků, a jelikož se jedná o nejčastěji opakovanou akci v běžném případě použití, jsou tyto nedostatky nejzávažnější. Mezi tyto nedostatky patří:

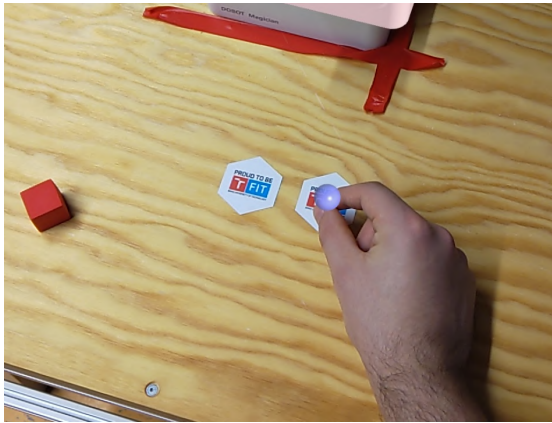


(a) Osy, umístěné uprostřed editační klece robota, se kterými není možné interagovat.

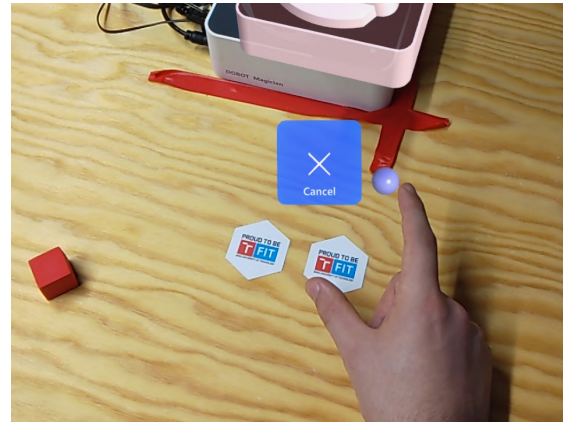
(b) Pohled uživatele snažící se zarovnat robota se stolem. Osa ani ruka která jí uchopila nejsou v pohledu a posun se tedy neregistruje.

Obrázek 4.7: Pohledy na scénáře ve kterých jsou demonstrovány problémy s editačními osami.

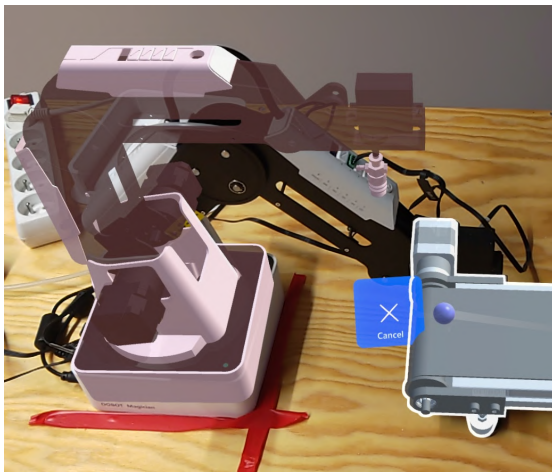
1. **Pozice akčního bodu** – Akční bod je přidán na špičce ukazováčku po uvolnění funkce airtap (rozevření prstů). Všem uživatelům se stalo, že spojili prsty v bodě, kde chtěli bod umístit, a při jejich rozevření byl bod umístěn o několik centimetrů vedle (obrázek 4.8). Po upozornění na implementaci této funkce se někteří účastníci snažili rozevřít prsty tak, aby ukazováček zůstal na stejném místě, což pro ně byl velmi nepřírodní pohyb prstů.
2. **Pořadí operací** – Po přidání akčního bodu se zobrazí prázdné menu obsahující pouze možnost cancel (obrázek 4.8c). Při tomto kroku se musela většina účastníků ptát na další postup (i přes předešlé vysvětlení). Zároveň se většina účastníků shodla na tom, že by bylo přirozenější nejdříve vybrat akci a až následně ji umíšťovat.
3. **Přidání akce na existující bod** – Pro přidání akce na již existující akční bod musí uživatel přidat akci při míření na existující akci, která je nad cílovým akčním bodem. Toto uživatelům nepřipadalo intuitivní a většině se stalo, že přidali akci při míření na cílový akční bod, čímž vytvořili nový akční bod, který byl hierarchickým potomkem cílového akčního bodu.
4. **Umístění menu pro výběr akce** – Jak již bylo zmíněno výše, menu pro výběr akce se vykresluje vedle přidávaného akčního bodu. Toto způsobilo, že se dvěma účastníkům vykreslilo v robotovi a částečně ve stole (obrázek 4.8d), což znemožnilo možnost vybrání několika položek (položky ve stole bylo možné vybrat pomocí gesta airtap, ale položky v robotovi ne). Zároveň jeden účastník podotkl, že po vybrání robota si nevšiml změny v menu, jelikož byl pohledem zaměřen na vybraného robota, což je dobrý poznatek, jelikož v současném řešení musí uživatel dvakrát převést pozornost mezi akčním bodem a robotem.
5. **Omezené menu** – Při přidávání akce je na ukazováčku vykreslován akční bod, který znemožňuje kliknutí na jakékoliv tlačítko v ručním menu (obrázek 4.9). Aby uživatel opustil tento režim, musí v ručním menu zmáčknout tlačítko pomocí funkce airtap.



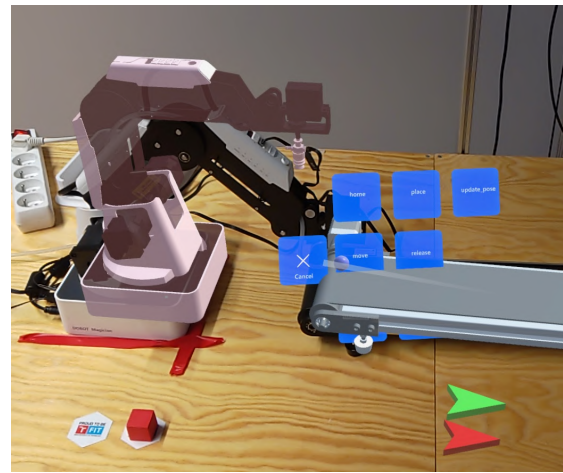
(a) První krok přidávání akce, uživatel spojil prsty v bodě, kde chce přidat bod.



(b) Druhý krok přidávání akce, uživatel rozpojil prsty, ale bod se přidal na polohu ukazováčku.



(c) Prázdné menu, pro výběr akce. Uživateli není řečeno co má dělat dále.



(d) Menu pro výběr přidávané akce je vykresleno ve stole a pásu, není tedy možné vybrat některé položky.

Obrázek 4.8: Scénáře demonstrující problémy s přidáváním akce na které účastníci narazili.

Přidávání spoje

Jediný nedostatek, na který uživatelé narazili při přidávání spojů, je chybné umístění více akcí nad jedním akčním bodem. Pokud je nad jedním bodem více akcí, akce se značně překrývají, což ztěžuje jejich výběr.

Ruční menu

S ručním menu měla většina uživatelů problém. Jednalo se o řadu problémů od drobností jako nevypovídající ikony po mylné výběry, kvůli velikosti tlačítek. Interakce s ručním menu většinu účastníků frustrovalo z následujících důvodů:

1. **Malé velikosti tlačítek** – Většina uživatelů měla problém s velikostí tlačítek a textu u nich. Kvůli menší velikosti tlačítek se účastníkům stávalo, že mylně zmáčkli vedlejší tlačítko. Současně většině účastníků připadal text v tlačítkách hůře čitelný.



Obrázek 4.9: Kulička na ukazováčku brání zmáčknutí tlačítek v menu.

2. **Zpětná vazba** – Většina uživatelů si stěžovala na nedostatek zpětné vazby. Zvuky tlačítek se totiž nepřehrají, pokud se podnabídka vypne po stisknutí tlačítka. Zároveň si uživatelé stěžovali na velmi nepatrné zvýraznění zmáčknutého tlačítka.

Obecné nedostatky

V této části budou popsány nedostatky, které nespádají nutně pod žádnou funkci rozhraní, ale působily negativně na uživatele. Mezi tyto nedostatky patří:

1. **Nízká snímková frekvence** – Většina uživatelů zmínila že běh aplikace se zdá trhavý, což některým způsobovalo větší únavu očí. Pro dlouhodobé využívání aplikace by bylo vhodné, aby aplikace běžela na stabilních 60 Hz⁶, zatímco momentálně se pohybuje mezi 25 a 30 Hz. Současně může nízká snímková frekvence ovlivňovat posun robota, který není plynulý, a dochází k de-synchronizaci ruky a posouvaného robota.
2. **Spuštění aplikace** – Při spuštění aplikace se uživateli zobrazí nápis „Made with Unity“ na bílém pozadí, které je při vykreslení v HoloLens 2 velmi intenzivní a může pálit do očí.
3. **Problikávající modely** – Po připojení na server jsou načítány modely robotů. Při tomto načítání je robot vždy na krátký moment umístěn před uživatele, což způsobuje nepříjemné problikávání velmi intenzivní bílé barvy.

⁶<https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/app-quality-criteria-overview>

Kapitola 5

Návrh změn v uživatelském rozhraní

V této kapitole je popsán návrh změn v uživatelském rozhraní pro HoloLens 2, které by měly zlepšit uživatelskou zkušenost a usnadnit programování robotického pracoviště. Tyto změny jsou založeny na základě poznatků z uživatelského experimentu, popsanych v předešlé kapitole. Kromě změn budou popsána i navržená rozšíření, která budou implementovat funkcionalitu z rozhraní pro tablet, potřebnou k snazšímu programování robotického pracoviště. Při návrhu byl hlavní důraz kladen na často měnící se programy, které je potřeba snadno upravovat tak, aby byla zachována hlavní myšlenka AREditoru.

5.1 Změny v existujícím řešení

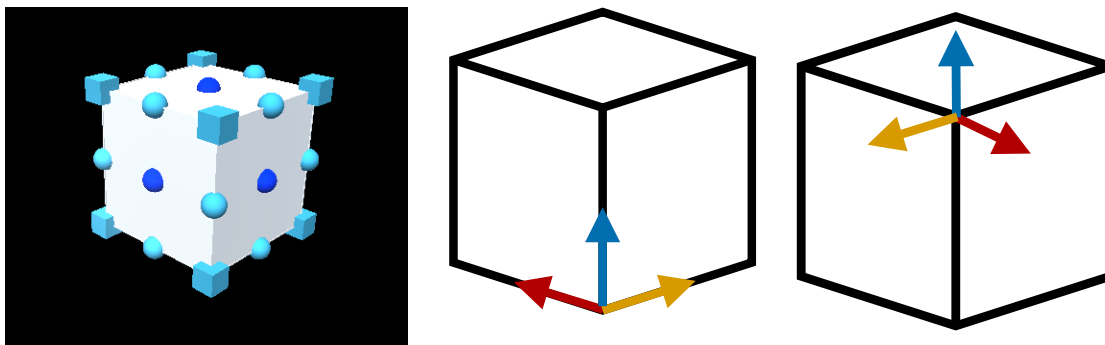
V této části budou popsány přepracované funkce rozhraní. Jedná se pouze o interakce uživatele s aplikací, zatímco ve správě scény a komunikace se serverem nebudou prováděny žádné změny.

Změny v režimu transformace

V režimu transformace je především nutné vyřešit, jak umožnit přesný posun objektů. Pro přesnější transformaci je vhodné omezit pouze na jednu osu, stejně jako je implementován posun podél os v současném řešení. Bohužel jako řešení MRTK poskytuje pouze editaci pomocí bodů na editační kleci (obrázek 5.1a), které trpí stejným problémem jako osy v současném řešení a to, že pro zarovnání robota podél osy y je nutné mít vršek objektu v zorném poli, jelikož spodní bod nelze uchopit kvůli stolu.

Priorita ručního paprsku

Nejjednodušším způsobem by bylo pouze implementovat priority paprsku tak, aby se paprsek nezastavil na editační kleci, ale zkontroloval, zda uživatel nemíří na osy uvnitř klece. Díky tomuto by vždy bylo možné uchopit osy a v případě zarovnání se stolem by stačilo uchopit osu níže. Toto řešení odstraní oba problémy s osami, ale uživatel nebude moci osy uvnitř robotů uchopit rukou.



(a) Editační klec obsahující body (vyznačeny tmavě modrou barvou) pro posun podél jedné osy. Osa podél které je objekt posouván je v souladu se směrem strany na které je uchopený bod.

(b) Editační osy podél editační klece. Takto zobrazené osy mohou potenciálně překrývat rotační body na editační kleci.

(c) Editační osy vycházející z rohu editační klece. Takto umístěné osy je snazší uchopit, ale je možné že budou kolidovat s dalším objektem, například se stolem při zarovnání podél osy y .

Obrázek 5.1: Ukázky možných řešení editace podél jednotlivých os.

Vykreslení editačních os u nejbližšího rohu

Další možnost je vykreslení os na rohu editační klece (obrázky 5.1b a 5.1c). Tento přístup umožňuje uchopení osy rukou a zároveň řeší problém s pozicí os. Určování u kterého rohu mají být osy vykresleny by mohlo být mírně složitější na implementaci. Zároveň by tento způsob nebylo možné zkombinovat se změnou velikosti, která se provádí uchopením bodu v rohu klece, což by neměl být problém, jelikož v aplikaci není změna velikosti robotů povolena.

Jelikož při testování původního rozhraní většina účastníků zarovnávala roboty podle jejich podstavy, je možné omezit umístění os pouze na spodní rohy, což by usnadnilo pozdější implementaci.

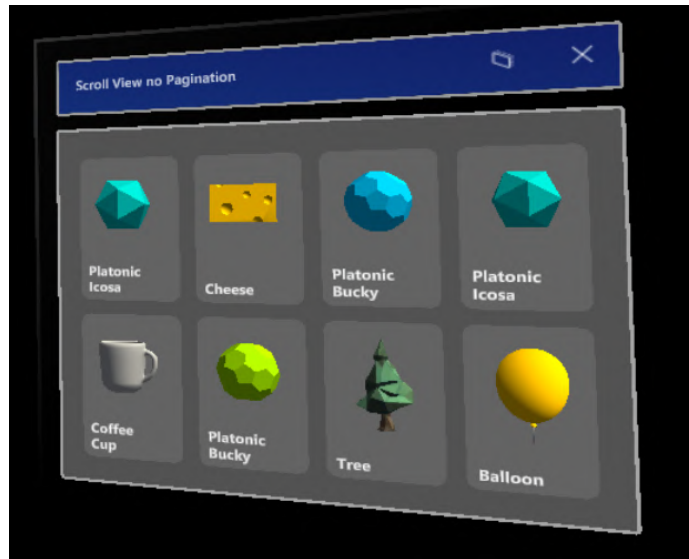
Přímá editace

Spolu s vyřešením přesné editace by bylo vhodné implementovat přímou editaci, která by umožnila objekt uchopit a rovnou posouvat či rotovat. Tento způsob sice bude příjemnější pro uživatele, ale je možné že ho bude obtížné implementovat tak, aby fungoval s manipulací podél os.

Přidávání objektů

Jelikož se většina uživatelů shodla na tom, že vidět modely robotů při výběru přidávaného robota je výhodné, stačí pouze předělat menu tak, aby bylo uživatelsky přívětivější. V případě užití tvorby scény se opakují pouze dvě operace, přidání robota a jeho umístění. Z tohoto důvodu by bylo vhodné, kdyby menu pro přidání robota mohlo zůstat otevřené jako jakýsi „toolbox“ či „paletka“. Podobné řešení (obrázek 5.2) bylo přímo demonstrováno vývojáři MRTK2 v ukázkové scéně pro scrollable komponenty.

Hlavním cílem tedy bude menu zmenšit a změnit jeho umístění. Umístění menu je možné implementovat dvěma způsoby, buď jako okno, které si uživatel může sám umísťovat ve scéně nebo paletku, která se připojí k ručnímu menu.



Obrázek 5.2: Okno sloužící pro výběr přidávaného objektu, implementované v rámci MRTK2 ukázkové scény.

Přidávání akce a akčního bodu

Na základě zpětné vazby by bylo vhodné změnit pořadí akcí pro přidání akce či akčního bodu. Nový postup by se skládal z těchto kroků:

1. výběr robota vykonávající akci,
2. výběr akce,
3. umístění akce.

Zároveň bude změněn moment, kdy je akční bod přidán. Místo momentu uvolnění gesta airtap bude akční bod přidán na začátku gesta, což by mělo usnadnit výběr pozice nového akčního bodu.

V dosavadní implementaci je přidávání akčního bodu spojeno s výběrem rodiče, což nedává smysl. Jelikož ne vždy lze provést airtap gesto na místě, kde uživatel chce umístit akční bod a zároveň mířit na akční objekt, který by měl být jeho rodičem. Zároveň bylo při vytváření akčního bodu velmi snadné omylem mířit na akční objekt, čímž byl zvolen jako rodič, přidávaného bodu. Z těchto důvodů by bylo vhodné oddělit výběr rodiče a umístění bodu a zmenšit objekt pro výběr rodiče. Asi nejjednodušším řešením by bylo při přidávání akčního bodu zobrazit tlačítko u každého akčního objektu, při jehož kliknutí by byl vybrán jako rodič přidávaného bodu. Kliknutí na tlačítko pomocí airtap gesta by bylo ignorováno a akční bod by byl přidán až při dalším airtap gestu.

Interakce na základě zvoleného objektu

Pro omezení interakcí s menu je možné implementovat vyvolání funkce na základě cílového objektu. Toto je možné díky tomu že všechny akce, kromě duplikace a mazání mají vyhrazené cílové objekty. Přidání akce doposud nemělo cílový objekt, ale jak bylo zmíněno v sekci 5.1, v nové verzi bude cílovým objektem robot vykonávající akci.

Programování tedy nebude postaveno na režimech, které je nutné přepínat v menu. Místo toho budou akce zmíněné v tabulce 5.1 vyvolány interakcí s objektem ve scéně. Pro režim mazání a duplikace a případně později implementované akce bude vytvořeno kontextové menu, které bude umístěno ve scéně pod vybraným objektem na straně, která je nejbližší uživateli. Aby bylo možné vyvolat menu bez interakce s objektem, bude možné ho vyvolat delším mířením paprsku na objekt.

Tento způsob programování bude pravděpodobně značně efektivnější, jelikož odstraní značné množství akcí, potřebných k dosažení stejného výsledku, ale je možné, že nebude tak intuitivní. Z tohoto důvodu bude nutné vizualizovat, jakou funkci interakce s objektem vyvolá. Tato vizualizace bude zobrazena při namíření paprskem na objekt.

Akce	Objekt pro vyvolání akce	Stav aplikace
Přidání akce	Robot	Projekt
Přidání spoje	Akce	Projekt
Posun akčního bodu	Akční bod	Projekt
Posun robota	Robot	Scéna

Tabulka 5.1: Popis akcí možných v režimu programování a objektů se kterými je nutné interagovat pro vyvolání akce.

5.2 Nově implementované funkce

Jak již bylo zmíněno v popisu existujících řešení v rozhraní pro HoloLens 2 stále chybí značná část funkcí z rozhraní tabletu. V této části bude popsán návrh integrace několika z těchto funkcí do HoloLens 2 rozhraní.

Možnost „undo“

Během experimentů došlo k několika chybným akcím, především při editaci, kdy uživatel místo editační osy uchopil celého robota a přišel o zarovnání podél předešlých os, nebo HoloLens 2 chybně detekovaly uchopení a uživatel si posunul již zarovnaného robota. Tyto akce mohou být pro uživatele frustrující, a proto by bylo vhodné implementovat možnost vrátit akci zpět – „undo“. Tlačítko pro tuto funkci pravděpodobně bude muset být umístěno v ručním menu, jelikož neexistuje jiné vhodné umístění. Zároveň by bylo vhodné, kdyby tato akce šla vyvolat hlasem, aby uživatel nemusel převést pozornost od scény k menu.

Možnost „undo“ je nutné implementovat především v režimu transform, stejně jako v rozhraní tabletu, ale byla by využitelná i v jiných režimech, jako třeba přidání spoje, kde mylný výběr akce byla také častá chyba, kterou uživatelé dělali.

Editor pro nastavení parametrů

V současném řešení jsou všem parametrům nastaveny výchozí hodnoty, což znemožňuje využívat některé funkce robotického pracoviště. Například nelze nastavit vzdálenost o jakou se má posunout dopravníkový pás nebo nelze nastavit adresa robota, což znemožňuje využití více robotů stejného typu. Z tohoto důvodu bude řešení doplněno o konfigurační editor, který bude možné vyvolat z kontextového menu. Tento editor bude dynamicky generovat svůj obsah podle parametrů konfigurovaného objektu. Pro každý parametr vytvoří editor hodnoty na základě jeho datového typu.

Přidání akčního bodu pomocí robota

V první iteraci uživatelských testů byla většina vytvořených programů nefunkčních, kvůli chybám v umístění robotů a akčních bodů. Z tohoto důvodu by bylo vhodné implementovat funkci přidání akčního bodu na základě aktuální pozice koncového efektoru robota, podobně jako je implementováno v rozhraní pro tablet. Umístění akčního bodu na aktuální pozici efektoru by bylo možné namířením na efektor robota při umísťování akce a následným gestem airtap.

5.3 Obecné změny

Kromě samotných funkcí rozhraní budou v rozhraní prováděny i ostatní změny, které by měly vylepšit celou aplikaci.

Převod na MRTK3

Jedním z hlavních bodů návrhu je převod na novější MRTK, který poskytuje snazší tvorbu UI a lépe stylizované UI komponenty s lepší zpětnou vazbou. Kromě vizuální části je MRTK3 méně náročný na výkon.

Drobné změny

V této sekci budou vyjmenovány drobné změny, které nevyžadují detailní popis. Mezi tyto změny patří:

1. zvětšení vertikálních mezer mezi akcemi,
2. aktualizace umístění akcí při smazání akce,
3. umožnění rekurzivního mazání akčního bodu, akcí a spojů,
4. přidání možnosti posunu všem dialogům v prostoru,
5. umožnění přepnutí do online režimu,
6. odstranění zvýraznění obrysu pomocí QuickOutline¹,
7. omezení blikání bílých objektů při načítání modelů robotů.

¹QuickOutline je knihovna, kterou autorka původní práce využila. Tato knihovna vyžaduje zapnutý anti-aliasing, který je náročný na výkon a snižuje FPS. Knihovna je dostupná na adrese <https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>

Kapitola 6

Implementace

V této kapitole bude popsána implementace navržených změn a rozšíření pro HoloLens 2 rozhraní a řešené problémy, které během implementace nastaly.

Před samotnou implementací bylo potřeba zprovoznit existující řešení, jednalo se především o aplikování změn v komunikaci se serverem, které byly implementovány v tabletovém rozhraní a odstranění konfliktních balíčků v Unity projektu. Zároveň bylo nutné změnit konfiguraci projektu tak, aby fungovaly všechny komponenty. Jednou z takovýchto změn bylo vypnutí načítání modelů v gLTF, využití v MRTK balíčku, který kolidoval s načítáním modelů přes knihovnu TriLib¹. Dále bylo třeba zapnout antialiasing, aby fungovala knihovna QuickOutline². Tato knihovna byla později nahrazena skriptem pro zvýraznění obrysu z MRTK3 balíčku.

6.1 Převod na MRTK3

Před implementováním jakýchkoliv změn jsem projekt převedl z MRTK2 na MRTK3, především kvůli optimalizacím a lepší vizuální stránce, ale i kvůli dlouhodobější podpoře, která jak již bylo zmíněno v sekci 2.5, bude omezena na komunitní příspěvky. Pro převod bylo potřeba nahradit MRTK balíčky za novější a ve scéně nahradit původní MRTK XR Rig za novější. Následně bylo potřeba opravit skripty. Většinou se jednalo o pouhý přepis jmenových prostorů z „Microsoft.MixedReality.Toolkit“ na „MixedReality.Toolkit“, ale objevily se i nekompatibility v názvech tříd i vlastností. Například komponenta pro interakci Interactable byla přejmenována na StatefulInteractable a podobně i její eventy se občas neshodovaly s předchozí verzí. Zároveň bylo nutné předělat všechny UI elementy ve scéně pomocí Canvas UI, odstranit MRTK2 skripty a nahradit je novými varianty, jako jsou StatefulInteractable a ObjectManipulator, v prefabech pro objekty AREditor scény či projektu.

Oprava citlivosti airtap gesta

Jedním z velkých problémů, se kterým jsem se setkal, byla přemrštěná citlivost airtap gesta, kde MRTK3 detekoval airtap gesto i když měl uživatel prsty 3 centimetry od sebe. Jedná se o známou závadu, která již má vytvořený issue³ v MRTK3 GitHub repozitáři.

¹<https://assetstore.unity.com/packages/tools/modeling/trilib-2-model-loading-package-157548>

²<https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>

³<https://github.com/MixedRealityToolkit/MixedRealityToolkit-Unity/issues/82>

Pravděpodobně se jedná se o chybu, která vznikla při migraci z *HandsAggregator*⁴ systému, který měl sloužit jako jednotné rozhraní pro vývojáře, skrz které bylo možné využívat detekci rukou z odlišných zařízení (Microsoft HoloLens 2, Magic Leap 2, Meta Quest 1/2, Oculus Rift), na *ActionBasedController*, který využívá *unity input actions*⁵.

Jelikož jsem nebyl schopný dohledat, zda je možné nastavit citlivost těchto akcí, musel jsem tento problém vyřešit přepsáním třídy *ArticulatedHandsController* tak, aby nebyly ignorovány vstupy z *HandsAggregator* systému. Jednalo se o přepsání několika podmínek, které kontrolují, zda tento update už byly přečteny hodnoty z unity akcí.

Taková úprava kódu balíčku se považuje za špatnou praxi a tyto změny nelze nahrát do repozitáře projektu. Jelikož se spíše jedná o obejítí problému a v budoucnu by bylo lepší využívat *ActionBasedController*, rozhodl jsem se tuto změnu nikam nepropisovat a bude pouze vysvětlena v této práci a README souboru.

Oprava knihovny QRTracking

V původní práci autorka využila knihovnu *QRTracking*⁶, která bohužel nefunguje s MRTK3 a není udržována. Musel jsem tedy upravit skript *SpatialGraphNodeTracker*, aby fungoval s novou verzí. Podobně jako u předešlých skriptů stačilo změnit použité jmenné prostory a přepsat získání reference na hlavní kameru, jelikož ve verzi MRTK3 pre.11⁷ byla odstraněna *CameraCache*. *CameraCache* sloužila jako optimalizovaná náhrada *Camera.main*, jelikož tento způsob získání hlavní kamery byl doposud velmi výkonně náročný⁸. Na rozdíl od změny citlivosti tyto změny by byly vhodné nahrát do repozitáře, a proto jsem vytvořil fork obsahující novou verzi.

Rozšíření komponenty pro interakci

Při implementaci různých částí jsem narazil na chybějící funkcionalitu komponenty pro interakci *StatefulInteractable*. Jednalo se o časově omezené funkce jako krátké kliknutí, dlouhé kliknutí a dlouhé namíření na objekt, které by rozšířily doposud značně omezené možnosti interakce.

Implementace těchto rozšíření je ve skriptu *StatefulInteractableExtensions*. Jelikož již existující třídy nelze rozšířit o event, musel jsem rozšířit třídu o funkce *RegisterOnLongSelect*, *RegisterOnShortClick* a *RegisterOnLongHover*, kterým se předá delegát na akci která se má vykonat. Implementace samotných funkcí je jednoduchá a využívá původních eventů a časovačů. V případě, kdy se má čekat než se provede interakce jsem využil třídu *Timer*, která ovšem spouští delegátní funkci v jiném vlákně, které nemůže interagovat s prvky unity scény a proto jsem musel implementovat *UnityMainThreadExecutor*. Jedná se o jednoduchý skript, který si drží list akcí, které mu ostatní skripty posílají a při zavolání jeho *Update* funkce jsou vykonány na hlavním vlákně.

⁴<https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk3-input/packages/input/hand-tracking>

⁵<https://docs.unity3d.com/Packages/com.unity.inputsystem@1.0/manual/Actions.html>

⁶<https://github.com/yl-msft/QRTracking>

⁷<https://learn.microsoft.com/cs-cz/windows/mixed-reality/mrtk-unity/mrtk3-overview/changelogs/mrtk3-pre-11-changelog>

⁸<https://blog.unity.com/technology/new-performance-improvements-in-unity-2020-2>

6.2 Interakce s objekty scény

Jedním z hlavních bodů návrhu bylo předělání interakce, z tohoto důvodu bylo nutné přepracovat celý *SelectorManager*, jelikož původně využíval režimy vybrané v ručním menu a nepřehledné delegatní funkce, které se vyvolávaly při interakci s objektem. Jediný aspekt, který jsem z původního kódu ponechal byl ten, že každý objekt ve scéně na sobě má *StatefulInteractable* komponentu, která při interakci s daným objektem zavolá metodu *SelectorManageru OnObjectInteraction* a předá referenci na sebe, která je implementací abstraktní třídy *HInteractiveObject*. Na základě typu předaného objektu se rozhodne jaká akce se má vykonat. V případě že se jedná o víceřadovou interakci, jako třeba přidání akčního bodu (viz 5.1), je průběh akce udržován jako enum proměnná *selectorState*, na základě které se může měnit chování při interakci s objekty. Kvůli MRTK3 optimalizacím bylo nutné přidat a nastavit *StatefulInteractable* komponentu z kódu až po načtení modelu, jelikož kvůli optimalizacím se použijí pouze kolizní objekty, přítomné při její inicializaci.

Transformace

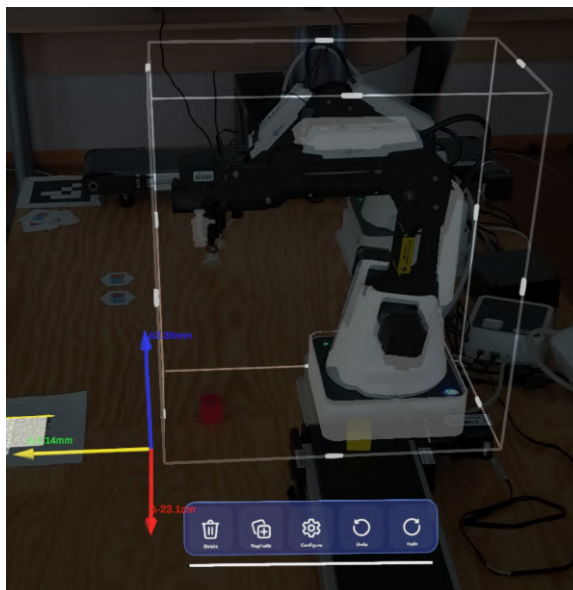
Pro implementaci přímého posunu bylo nutné předělat celý původní systém. Místo skriptu *TransformManager* se o posun starají samotné objekty *AREditor* scény, které na sobě mají připojený skript *ObjectManipulator* starající se o posun objektu. Roboti a kolizních objekty mají navíc připojený skript *BoundsControl*, který se stará o rotaci pomocí madel. O viditelnost kolizní klece a jejich madel se stará skript *BoundsControlAdditionalHandler*, který vyvolá klec po první manipulaci. Původně tento skript zobrazoval klec už při namíření na robota paprskem, ale při testování se ukázalo toto řešení nepraktické, jelikož kolizní krychle robotů byly moc velké a překrývaly se.

Objekty scény musí omezovat editační skripty tak, aby uživatel mohl manipulovat s objekty v souladu s původním *AREditor* řešením. Manipulace je tedy ve výchozím stavu vypnuta a povoluje se na základě počátku a konce interakce, při kterých se kontroluje, zdali je aplikace ve správném režimu. Zároveň je potřeba před každou editací objekt zamknout, pro vyloučení současné editace jednoho objektu více uživateli. Díky této implementaci byla odhalena chyba v zamykacím systému, díky které se aplikace mohla dostat do stavu, ve kterém je objekt zamčený na serveru, ale odemčený na klientovi. Tento stav způsoboval to, že se klient snažil zamknout objekt, ale server žádost odmítal a s objektem tedy nešlo manipulovat. Problém byl nahlášen a jako dočasné řešení jsem navrhl, aby server přijímal tyto žádosti, ale v budoucnosti by bylo vhodné zajistit, aby se stav zámeků v klientské aplikaci vždy shodoval se serverem.

O zarovnání editačních os se stará skript *TransformGizmoHandler*, který umísťuje osy na nejbližší spodní roh k uživateli (obrázek 6.1). Jelikož osy je možné posouvat pouze v jednom směru, je potřeba je individuálně rotovat tak, aby vždy mířily ve směru jejich osy od robota. Toto neplatí pro osu *y*, která vždy míří vzhůru, aby jí bylo možné uchopit i blízkou interakcí.

Kontextové menu

Kromě samotné interakce jsem doplnil *SelectorManager* o funkci vybraní objektu, která se zavolá nejen při interakci, ale i při delším míření paprsku na daný objekt. U vybraného objektu se zobrazí kontextové menu (obrázek 6.1) popsané v sekci 5.1. Důvodem pro takovéto vyvolání byla potřeba vyvolat menu bez interakce s objektem. Menu je umístěno na nejbližším bodě na kolizním objektu ke kameře. Následně je ještě posunuto níže, aby



Obrázek 6.1: Obrázek obsahující editační klec kolem robota. Pod robotem je kontextové menu, obsahující funkce Undo a Redo. U nejbližšího spodního rohu jsou umístěné editační osy.

uživateli nepřekáželo. Pro podobná umístování byl vytvořen skript *UI3DHelper*, který je schopný několika způsoby zarovnat herní objekty k objektu s komponentou *InteractiveObject*. Samotné menu je ještě posouvateľné pomocí bílého madla v případě, že by uživateli překáželo.

Funkce „undo/redo“

Funkci „undo/redo“ obstarává *UndoManager*, který drží list záznamů provedených akcí. Tyto záznamy jsou implementací abstraktní třídy *UndoRecord* a udržují všechny informace potřebné k provedení opačné akce, ale i k znovu provedení akce. Jelikož se mi nepodařilo napojit tento systém na hlasové příkazy, musel jsem tlačítka těchto funkcí umístit do ručního menu, ale při prvním testování se toto řešení ukázalo jako velmi nepraktické a musel jsem tyto tlačítka umístit taktéž do kontextového menu (obrázek 6.1). Tlačítka v kontextovém menu ovšem mohou vyvolat pouze „undo/redo“ záznamy, které souvisí s daným objektem. Prozatím jsem implementoval tuto funkci pouze pro transformaci objektů scény, ale systém je snadno rozšiřitelný.

Přidání akčního bodu a akce

Přidávání akčního bodu a akce jsem podle původního řešení ponechal spojené, ale na základě zpětné vazby jsem změnil postup, který je možné vidět na obrázku 6.2. V novém postupu je prvně vybrán robot, který má akci vykonat. Toto je implementováno pomocí interakce popsáné výše. Následně se otevře menu umístěné rovnou u robota pomocí *UI3DHelperu*. Po vybrání akce se *SelectorManager* přepne do režimu umístování.

Podobně jako v původním řešení je umístování bodu demonstrováno zobrazením kuličky na ukazováčku a akce je umístěna spojením prstů, k tomuto slouží skript *HandInteractionHandler*, který je přepracovanou verzí skriptu *AddActionPointHandler*. Tento skript si

pomocí systému *HandsAggregatorSubsystem* získává pozici ukazováčku, na kterém umísťuje vizualizaci akčního bodu. Zároveň průběžně sleduje míru vykonání gesta airtap, při kterém odesílá do *SelectorManageru*, zprávu o provedení gesta a případném cílovém objektu. Jelikož *HandsAggregatorSubsystem* nevrací vybraný objekt, musel jsem ho získat pomocí reference na *MRTKRayInteractor* dané ruky.

Jelikož je akci možné vybrat pomocí paprsku a airtap gesta, bylo potřeba čekat na následné ukončení tohoto gesta, aby nedocházelo k okamžitému umístění akčního bodu. Po umístění akce pomocí musí systém také počkat na uvolnění gesta airtap, aby nedocházelo k okamžité interakci s nově umístěným akčním bodem či cílovým objektem. Tuto funkcionalitu taktéž obstarává *HandInteractionHandler*.

Jak již bylo zmíněno, při přidávání akce může cílový objekt ovlivnit to, jaká funkce je vykonána při provedení airtap gesta. Mezi možné cílové objekty patří:

- **Akce či akční bod** – Stejně jako v původním řešení výběr existující akce přidá akci na její akční bod. Stejná funkce byla přiřazena i výběru akčního bodu, jelikož nepovažují výběr akčního bodu jako rodiče za důležitou funkci.
- **„Make parent“ tlačítko** – Při umísťování akčního bodu se u akčních objektů objeví tlačítko (obrázek 6.2c), které při stisku vybere objekt jako rodiče nově přidávaného akčního bodu. Kliknutí na toto tlačítko pomocí airtap gesta je tedy ignorováno a místo přidání akčního bodu si pouze *SelectorManager* uloží odkaz na vybraný objekt.
- **Koncový efektor robota** – V případě, že je scéna v online režimu, jsou aktuální pozice robotů zasílány do aplikace, která je následně vizualizuje. Na objekt koncového efektoru robota jsem přidal kolizní objekt a *StatefulInteractable* komponentu (obrázek 6.2c), která při výběru vyvolá přidání akce a akčního bodu pomocí robota. K tomuto přidání jsem využil již existující metodu serveru implementovanou v rozhraní pro tablet. Takto umístěný akční bod má zaručeně správnou pozici a umožní vytvářet spustitelné programy.
- **Žádný cíl** – V případě že uživatel nevybere žádný cílový objekt, je nový akční bod s vybranou akcí umístěn na místě ukazováčku při spojení prstů.

Konfigurace parametrů

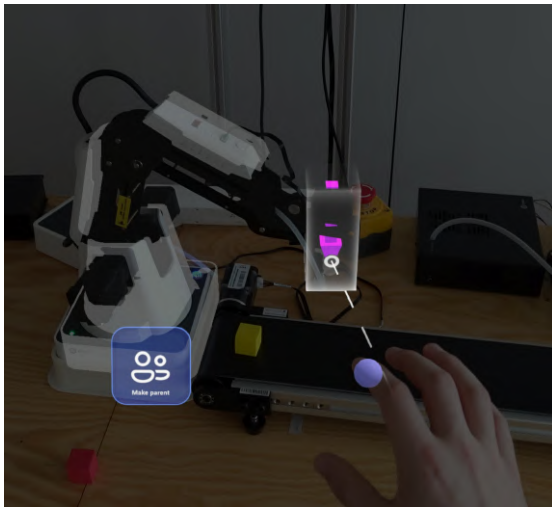
Konfigurace parametrů je možné vyvolat pomocí kontextového menu robota či akce. Při vybrání této možnosti se zobrazí okno do kterého jsou generovány editory jednotlivých parametrů (obrázek 6.3a). Za správu tohoto okna zodpovídá skript *ParameterConfigurationManager*, který nejen generuje jednotlivé editory, ale stará se i o uložení či zrušení změn. Pro každý parametr je vybrán prefab editoru na základě datového typu. Zatím jsem implementoval pouze typy bool, string, int, float a enum, ale konfigurace je snadno rozšiřitelná. Každý editor parametru má přidělený skript, který implementuje abstraktní třídu *ParameterEditorBase*, která slouží jako nutné rozhraní pro *ParameterConfigurationManager*. Samotná implementace této třídy se zabývá pouze tím jak řídit UI prvky daného editoru, jako třeba ComboBox (obrázek 6.3b), který jsem musel implementovat pro enum editor. Pro rozšíření konfigurace o další typ tedy stačí vytvořit prefab se skriptem implementující *ParameterEditorBase* a přidat ho do logiky generující editory.



(a) Vyvolání přidání akce pomocí interakce s robotem. Při namíření na robota je na paprsku zobrazena nápověda



(b) Menu pro výběr vykonané akce, umístěné podle kolizního objektu robota. Menu je možné posunout pomocí bílého madla.

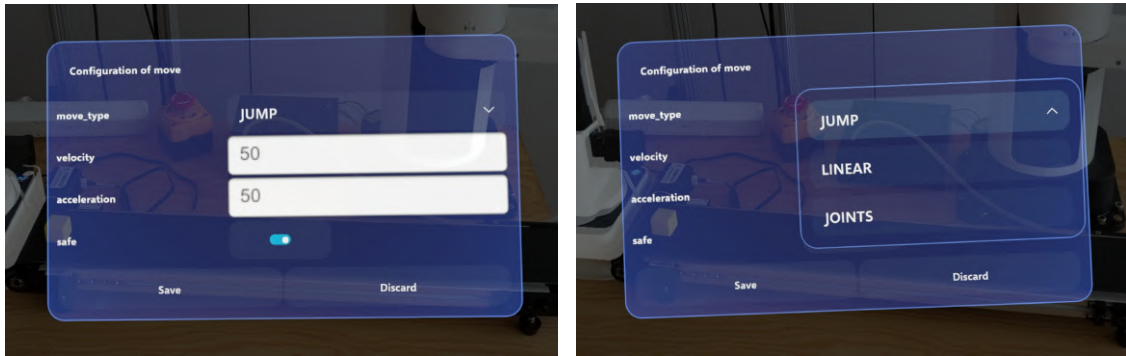


(c) Výběr koncového efektoru robota při přidávání akce, pro umístění akce pomocí robota. U robota je možné vidět „make parent“ tlačítko.



(d) Akce umístěná na základě pozice robota.

Obrázek 6.2: Obrázky demonstrující proces přidání akce pomocí robota.



(a) Dialog konfigurace pro akci move. Dialog obsahuje vygenerované editory pro jednotlivé parametry. (b) Dialog konfigurace obsahující combobox pro typ string enum.

Obrázek 6.3: Ukázky dialogu pro konfiguraci parametrů.

6.3 Zobrazení seznamů

V rámci převodu aplikace na MRTK3 jsem musel předělat všechny UI prvky, včetně menu pro výběr přidávaného akčního objektu a výběru scény či projektu. V sekci 5.1 je popsán návrh na změnu menu pro přidávání robota, jehož implementace bude v této sekci popsána. Menu pro výběr scény či projektu jsem musel celé vytvořit od základu, ale ponechal jsem ho téměř stejně jako v původním řešení.

Menu pro přidání akčního objektu

Menu pro přidání akčního objektu jsem podle návrhu spojil s ručním menu (obrázek 6.4). Většina kódu pro vytvoření modelu, který je vložen do tlačítka, zůstala stejná, bylo však nutné u modelů změnit shader na *Mixed Reality Toolkit/Standard*, jelikož při zmenšení modelu s původním shaderem, docházelo k chybě stínování a modely byly vykreslovány černé. Zároveň bylo potřeba opravit umístování a změnu velikosti modelů tak, aby se vešly do tlačítka. Jelikož modely načtené ze serveru neměly jednotný formát umístění středu, bylo potřeba vypočítat meze každého modelu, na základě kterých byl objekt zmenšen a umístěn. Zároveň bylo potřeba opravit nastavení výchozích parametrů přidávaných robotů, aby bylo možné přepnout scénu do online režimu.

Výběr scény či projektu

Styl reprezentace scény či projektu jsem ponechal jako kostku, obsahující zmenšené modely prvků scény či projektu. Stejně jako u předešlé sekce bylo potřeba opravit výpočty velikostí a zarovnání do kostek. Jelikož MRTK3 neposkytuje posuvné menu pro 3D objekty, musel jsem implementovat skript napodobující chování původního menu. O posun se stará skript *HorizontalObjectScrollMenu*, který využívá *ObjectManipulator* na každé kostce tak, aby posun kostky posouval celé menu. Zároveň bylo nutné rozlišovat mezi krátkým výběrem pro otevření scény a dlouhým pro posun v menu. Menu je vždy umístěné na stole, aby nedocházelo k problémům popsaných v sekci 4.4.



(a) Menu pro výběr robota.



(b) Menu pro výběr kolizního objektu

Obrázek 6.4: Ukázky menu pro výběr akčního objektu, které je nově připojeno k ručnímu menu.

Kapitola 7

Výsledky finálního experimentu

V této kapitole budou popsány výsledky uživatelského testování, spolu s možnými důvody pro naměřené hodnoty. Zároveň zde bude popsán vliv mnou implementovaných změn na výsledky experimentů. Nakonec budou navrženy další způsoby jak porovnávat tato rozhraní a možné implementační rozšíření rozhraní pro HoloLens 2.

Uživatelé prováděli stejný experiment jako v první iteraci, který je popsán v kapitole 4. Jediný rozdíl mezi iteracemi jsou mnou implementované změny, samotní účastníci a mírná změna v rozložení robotického pracoviště, která by neměla ovlivnit výsledky.

7.1 Srovnání tabletu a HoloLens 2

V druhé iteraci experimentů bylo otestováno 14 účastníků. Jednalo se především o studenty vysokých škol ve věku 20–25 let. Většina se skládala ze studentů BUT FIT, ale několik účastníků bylo i z jiných fakult, včetně fakult s netechnickým zaměřením. Několik účastníků mělo již předešlé zkušenosti s HoloLens 2, jelikož s nimi sami pracují v rámci jejich diplomových prací. Tito účastníci obvykle, ale ne vždy, dosahovali lepších výsledků, jelikož neměli takový problém s ovládním HoloLens 2 vstupů. Díky tomuto je možné, alespoň částečně vyloučit neseznámenost s HoloLens 2 jako faktor v naměřených hodnotách. Demografické údaje účastníků je možné vidět v tabulce 7.1.

Podobně jako v první iteraci se data mezi skupinami příliš nelišila a výsledky budou prezentovány jako průměry z obou skupin. Díky většímu počtu účastníků bylo možné statisticky testovat hypotézy popsané v sekci 4.2 pomocí párového t-testu. Přehled hypotéz a jejich potvrzení je možné vidět v tabulce 7.2.

Objektivní metriky

Naměřená objektivní data z experimentů, potvrzují většinu hypotéz. Naměřené hodnoty při užití HoloLens 2 mají větší rozdíl mezi extrémy, což je pravděpodobně způsobeno tím, že někteří uživatelé měli problém s ovládním HoloLens 2. Největší problém měli uživatelé s prováděním airtap gesta tak, aby jej bylo možné stabilně detekovat. Ve všech úlohách uživatelé dosáhli kratších časů při využití HoloLens 2. Porovnání naměřených časů je možné vidět na obrázku 7.1.

Jak je možné vidět v tabulce 7.3 první hypotéza byla potvrzena s p hodnotou 0,003. Při porovnání jednotlivých úloh byl ale nalezen statisticky významný rozdíl pouze u čtvrté úlohy.

Účastník	Věk	Pohlaví	Vzdělání	Skupina
A	22	muž	bakalářské	A (skupina začínající s HoloLens 2)
B	23	muž	bakalářské	
C	23	žena	středoškolské	
D	24	muž	bakalářské	
E	23	muž	bakalářské	
F	23	muž	bakalářské	
G	23	muž	bakalářské	
H	23	muž	bakalářské	B (skupina začínající s tabletem)
I	20	muž	středoškolské	
J	23	muž	bakalářské	
K	23	muž	středoškolské	
L	24	muž	bakalářské	
M	24	muž	bakalářské	
N	24	muž	bakalářské	

Tabulka 7.1: Demografické údaje účastníků 2. experimentu.

#	Hypotéza	P hodnota	Potvrzena
1	HoloLens 2 umožní rychlejší vykonání úloh	0,003	Ano
2	HoloLens 2 umožní přesnější anotaci pracoviště	0,046	Ano
3	HoloLens 2 poskytne rychlejší způsob výběru	0,009	Ano
4	Práce s tabletem vyžaduje více pohybu po pracovišti	0,812	Ne
5	Práce s tabletem bude dosahovat vyšší fyzické náročnosti	0,532	Ne

Tabulka 7.2: Tabulka obsahující přehled hypotéz, p hodnotu využitou při jejich vyhodnocení a potvrzení hypotézy. Pro potvrzení hypotézy bylo vyžadována 95% pravděpodobnost odlišnosti.

Metrika	1. Úloha	2. Úloha	3. Úloha	4. Úloha	Všechny úlohy
Čas	0,749	0,155	0,145	0,009	0,003
Vzdálenost	0,052	0,212	0,431	0,410	0,812

Tabulka 7.3: Tabulka p hodnot při porovnání časů a nachezené vzdálenosti párovým t-testem. Mezi hodnotami nebyl nalezen statisticky významný rozdíl.

Typ chyby	P hodnota
Pozice robota	0,052
Rotace robota	0,191
Pozice a rotace robota	0,112

Tabulka 7.4: Tabulka p hodnot při porovnání chyb umístění robotů párovým t-testem. Tučně jsou vyznačeny hodnoty menší 0,05, které indikují statisticky významný rozdíl mezi zařízeními.

V první úloze byl u HoloLens 2 naměřen značně kratší čas, což je pravděpodobně způsobeno menším počtem položek při přidávání robota, nebo rychleji přístupnějším vypnutím experimentu, které je rovnou v ručním menu.

V druhé úloze nedosáhli uživatelé o moc lepší časové výsledky než při použití tabletu, ale přesnost zarovnání byla značně lepší (obrázek 7.3a). Co se týče rotace, při využití tabletu dosáhli uživatelé lepších výsledků (obrázek 7.3b). Důvodem pro větší chybu rotace robotů je pravděpodobně fakt, že HoloLens 2 rozhraní postrádá možnost přesné rotace. Jedná se ovšem o malé chyby do 2 stupňů, které by neměly výrazně ovlivnit práci s pracovištěm.

Při pomnutí rotace robotů, tak naměřená data potvrzují i druhou hypotézu s p hodnotou 0,046 a je možné říci, že HoloLens 2 umožňují přesnější zarovnání robotů. Výsledky t-testů porovnávající anotaci pracoviště je možné vidět v tabulce 7.4.

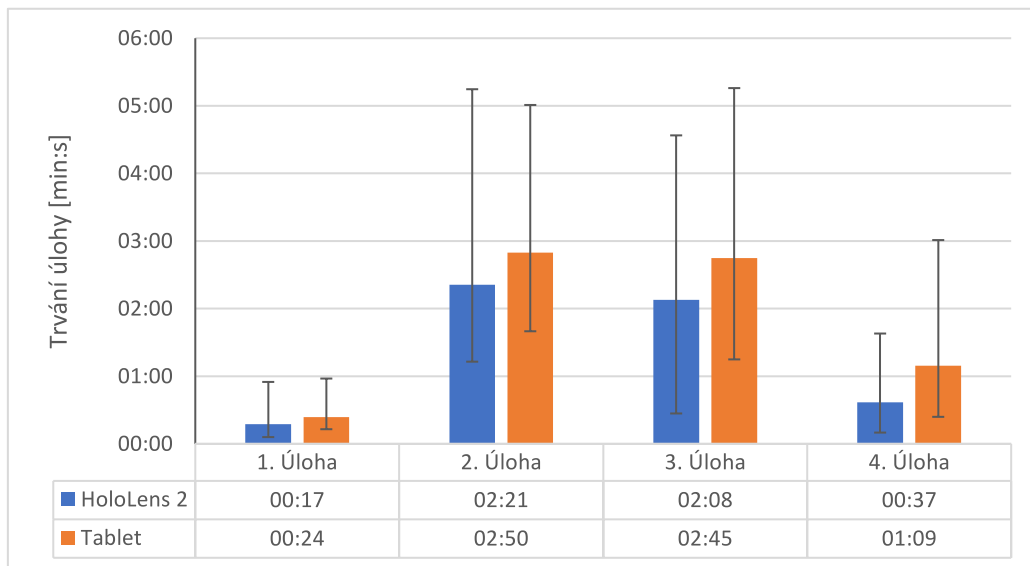
Při přidávání akcí ve třetí úloze byly uživatelé s HoloLens 2 opět rychlejší a většina účastníků ocenila, že při využití HoloLens 2 mají volné ruce pro manipulaci s robotem (obrázek 7.7). Důvodem pro možná menší rozdíl mezi zařízeními, může být umístění menu pro výběr akcí a drobné implementační chyby v HoloLens 2 rozhraní, které byly pro pozdější experimenty opraveny.

Při spojování akcí ve čtvrté úloze byli uživatelé průměrně o 46,3 % rychlejší při použití HoloLens 2 a většina účastníků považovala výběr pomocí paprsku za lepší (obrázek 7.7), ale při dalším dotazování potvrdili, že kdyby scéna obsahovala více objektů, které by se překrývaly, bylo by míření paprskem obtížnější. Úloha tedy nakonec neprověřila schopnost výběru ve scéně s velkou hustotou objektů. Při porovnání časů ve třetí úloze vyšla p hodnota 0,009. Třetí hypotéza tedy byla potvrzena.

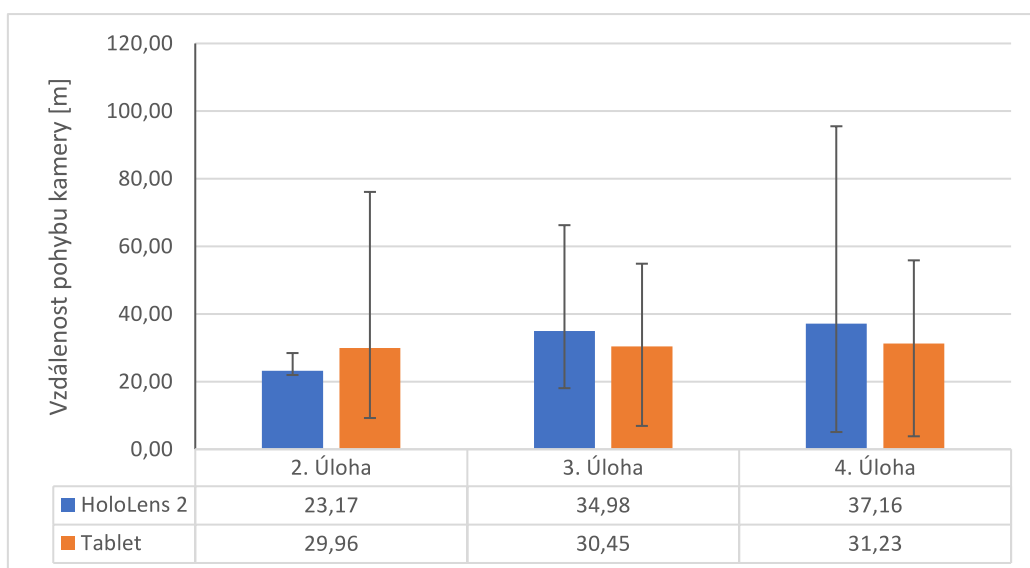
Co se týče naměřených vzdáleností, ukázalo se, že uživatelé s HoloLens 2 se značně pohybovali po pracovišti, aby si usnadnili míření paprskem, a tedy ve všech úlohách, kromě druhé, nachodili větší vzdálenosti. Při zarovnání robotů v druhé úloze se podle očekávání pohybovali uživatelé více s tabletem, ale rozdíl není tak velký, jak by se dalo očekávat. Sami účastníci tvrdili, že pro opravdu přesné zarovnání by se potřebovali dívat na robota z více úhlů i s HoloLens 2, a proto téměř žádný účastník silně nesouhlasil s tím, že by s HoloLens 2 zvládli zarovnání bez pohybu kolem pracoviště (obrázek 7.7). V tabulce 7.3 je možné vidět porovnání nachozených vzdáleností pomocí t-testu. Mezi zařízeními nebyl naměřen statistický rozdíl v jednotlivých úlohách ani celkovém porovnání. Čtvrtá hypotéza tedy není potvrzena.

Subjektivní hodnocení

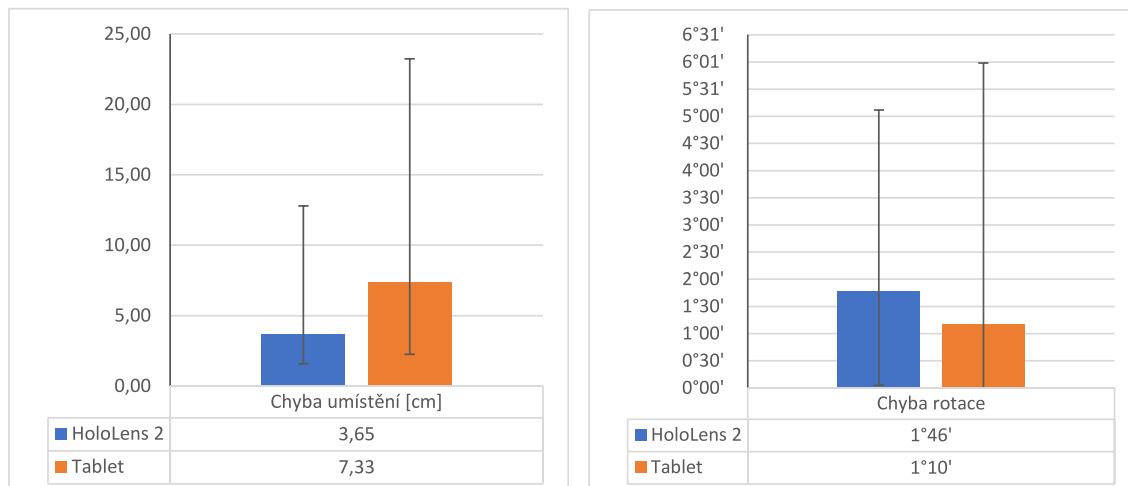
Podobně jako v první iteraci testů, subjektivní hodnocení bylo od účastníků sbíráno pomocí dotazníků popsanych v sekci 4.2. Původní dotazník byl rozšířen o několik vlastních otázek, které příměji porovnávají či hodnotí použitá rozhraní. Na rozdíl od první iterace



Obrázek 7.1: Graf zobrazující průměrné časy vykonání jednotlivých úloh. Sloupce obsahují indikátor extrémů.



Obrázek 7.2: Graf zobrazující průměrné vzdálenosti pohybu kamery při jednotlivých úlohách. Sloupce obsahují indikátor extrémů.



(a) Průměrná chyba umístění robotů podle zařízení.

(b) Průměrná chyba rotace robotů podle zařízení.

Obrázek 7.3: Průměrné chyby umístění podle zařízení, naměřené při druhé úloze.

testů, uživatelé hodnotili obě zařízení velmi mírně, ale podobně jako v zimním semestru nejdůležitější údaj je vždy rozdíl mezi zařízeními.

System Usability Scale

SUS dotazník byl pravděpodobně nejpřísněji vyplňovaný dotazník, ve kterém většina uživatelů hodnotila HoloLens 2 lépe (obrázek 7.4), ale obě zařízení dosáhla průměrného až nadprůměrného hodnocení, kde HoloLens 2 rozhraní dosáhlo SUS skóre **77,14** a rozhraní pro tablet dosáhlo skóre **68,75** což je hodnota indikující průměrné řešení, podle SUS benchmarku. HoloLens 2 rozhraní průměrně získalo o 8,39 SUS skóre více než tablet a většina uživatelů preferovala toto rozhraní i přesto, že někteří měli potíže se naučit s ním pracovat.

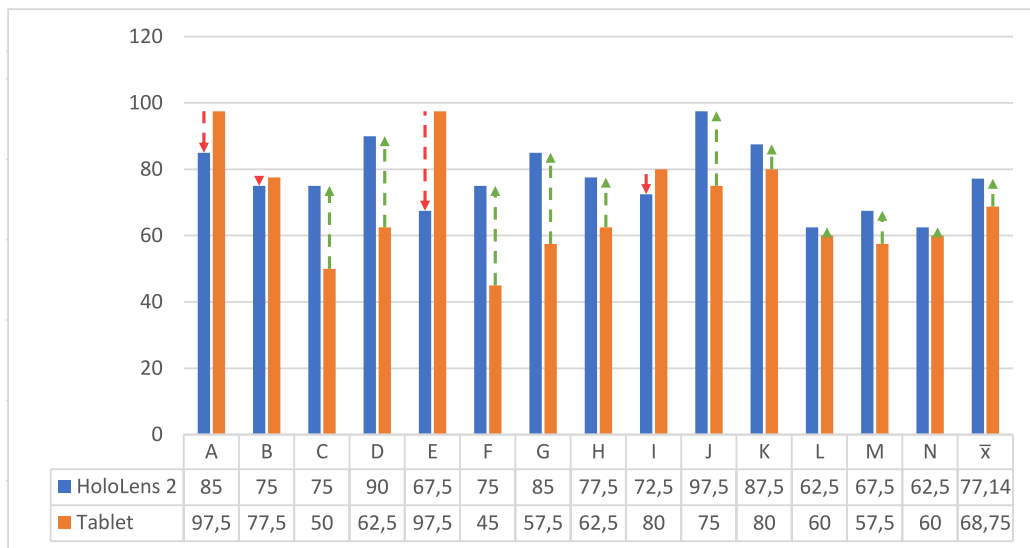
NASA Task Load Index

Náročnost hodnotili účastníci pro obě zařízení velmi podobně (obrázek 7.5), s tím, že většinou hodnotili tablet mírně náročnější na využití (rozdíl 6,4%), ale ani jedno rozhraní nepřípadalo uživatelům příliš náročné na použití. P hodnota při porovnání fyzické náročnosti byla 0,532 a pátá hypotéza tedy nebyla potvrzena.

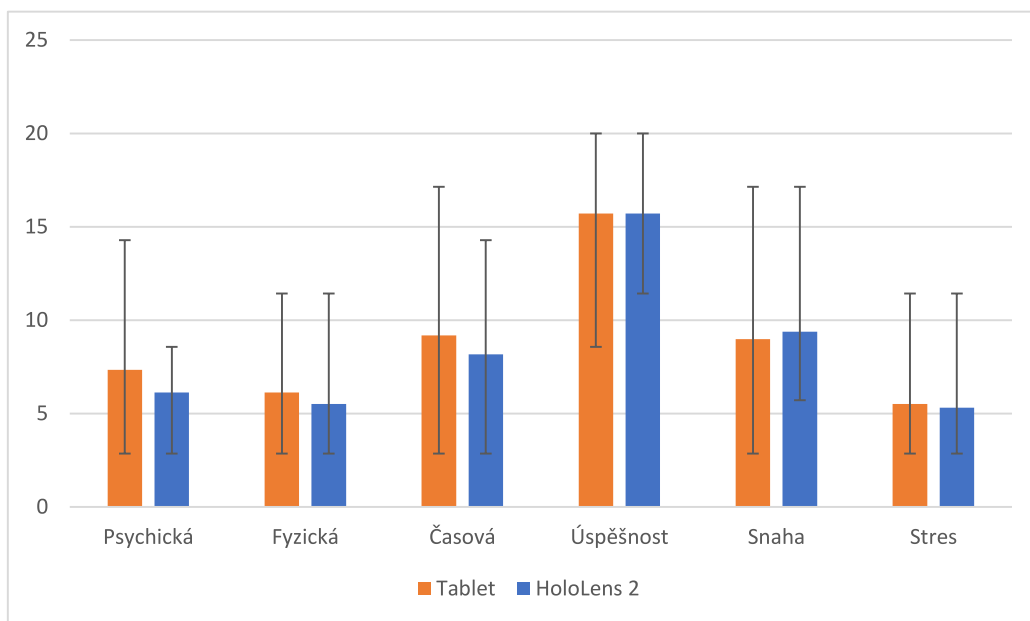
Větší náročnost u tabletu vybírali uživatelé především psychickou, fyzickou a časovou. Tyto data se shodují i s jejich slovním hodnocením, při kterém se většina účastníků shodla na tom, že rozhraní pro tablet je komplikovanější a hůře se v něm orientují. Zároveň si většina účastníků stěžovala na nutnost si několikrát kleknout či dřepnout při zarovnávání robotů, aby získali pojem o jejich umístění v ose y. Na časovou náročnost si nikdo nestěžoval, ale subjektivní hodnocení se shoduje s naměřenými časy, které jsou popsány v předešlé části 7.1.

User Experience Questionnaire

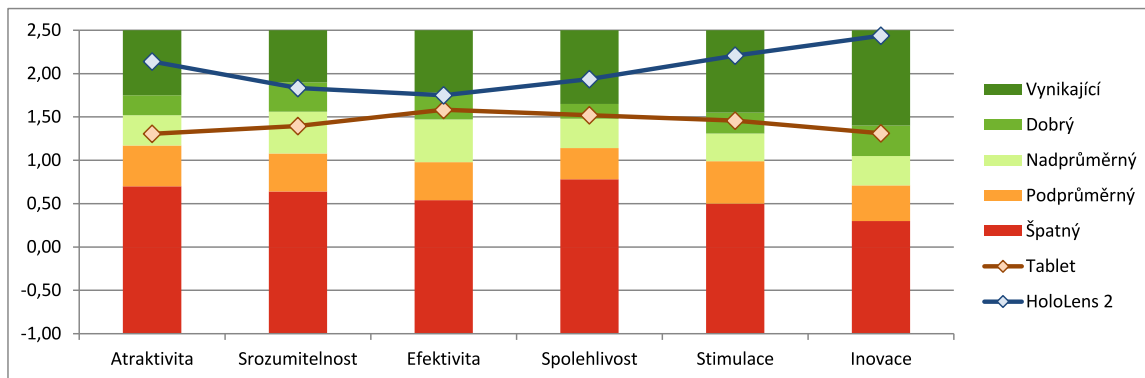
Při hodnocení uživatelské zkušenosti pomocí UEQ dotazníku byli účastníci, na rozdíl od první iterace, velmi mírní a obě rozhraní hodnotili kladně (obrázek 7.6). Ve všech kategoriích hodnotili uživatelé HoloLens 2 rozhraní pozitivněji, což je docela překvapivé, obzvláště



Obrázek 7.4: Graf zobrazující SUS hodnocení od jednotlivých účastníků a celkový průměr pro obě rozhraní. Šipky na sloupcích indikují rozdíly mezi rozhraními, kde zelená indikuje lepší hodnocení HoloLens 2.



Obrázek 7.5: Graf zobrazující průměrné hodnoty jednotlivých zátěží měřených v rámci NASA-TLX dotazníku. Úspěšnost byla invertována kvůli překladu a tedy je opak zátěže.



Obrázek 7.6: UEQ benchmark porovnávající průměrné hodnocení tabletu a HoloLens 2 v jednotlivých kategoriích, spolu s dalšími produkty, podle kterých byly vytvořeny oblasti hodnocení.

u spolehlivosti a efektivity, jelikož při konání úloh měli často problém s registrací gest, což je frustrovalo a zpomalovalo. Lepší hodnocení srozumitelnosti dává smysl, jelikož HoloLens 2 rozhraní obsahuje méně UI prvků a funkcí a zároveň jsou některé funkce srozumitelnější, jako třeba výběr prvků scény pomocí paprsku. Lepší hodnocení atraktivita, stimulace a inovace pravděpodobně pramení z toho, že HoloLens 2 byla pro většinu uživatelů nová technologie a MRTK3 design v samotné aplikaci jim také líbil více než rozhraní tabletu.

Vlastní dotazník

Co se týče výsledků vlastního dotazníku, jehož hodnoty je možné vidět na obrázku 7.7, většina odpovědí se klaní ve prospěch HoloLens 2, ale uživatelé vážali silně souhlasit s většinou výroků.

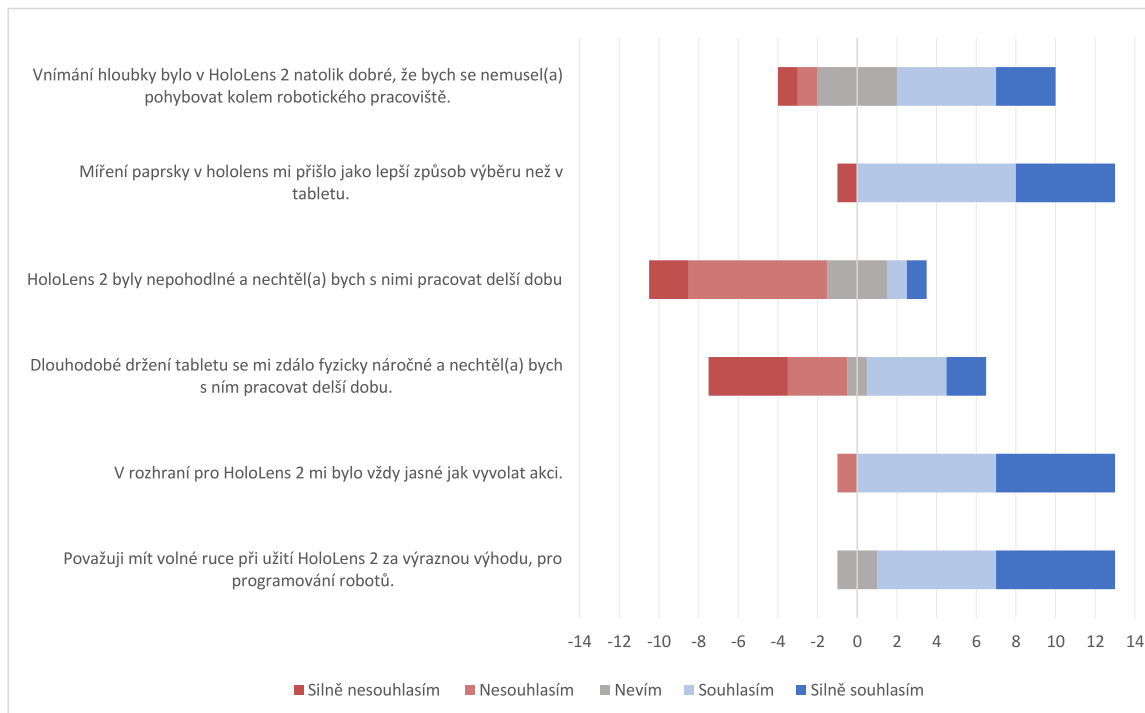
Vnímání hloubky uživatelé považovali za velmi dobré a podle jejich ústního hodnocení i jako výhodu, ale pro opravdu přesné zarovnání robotů či jiných objektů měli potřebu se pohybovat kolem pracoviště.

Míření paprskem přišlo všem, až na jednoho uživatele, jako přirozenější způsob výběru, který ovšem nebyl plně otestován a při následném dotázání většina účastníků uznala, že míření paprskem může být problematické, pokud by objekty byly v zákrytu. Důvodem pro takto kladné hodnocení je také fakt, že většina uživatelů měla problém s menu pro výběr v rozhraní pro tablet. Menu se dynamicky mění podle míření do scény, ale při pokusu o vybrání objektu či akce si uživatelé často pohnuli s tabletem a obsah menu se změnil.

Při přímém dotázání na zátěž většina účastníků neměla se zařízeními problém, ale někteří měli potíže s delším držením tabletu a báli se, že ho upustí. Jiní si zase stěžovali na únavu očí z delšího použití HoloLens 2.

Většina uživatelů neměla problém s vyvoláváním akcí pomocí interakce s objekty ve scéně, ale někteří se museli chvíli zamýšlet, jak vyvolat některou akci.

Dotazování běžných uživatelů na to, zda je mít volné ruce výhodou při programování robotů, pravděpodobně nebylo vhodné, jelikož si většina nedokázala představit nic jiného než manipulaci s robotem Dobot Magician, kterou zvládali i při držení tabletu. Z tohoto důvodu si značná část uživatelů nebyla jistá souhlasem s tímto tvrzením.



Obrázek 7.7: Výsledky vlastního dotazník, ve kterém účastníci vyjadřovali souhlas či nesouhlas s uvedenými výroky.

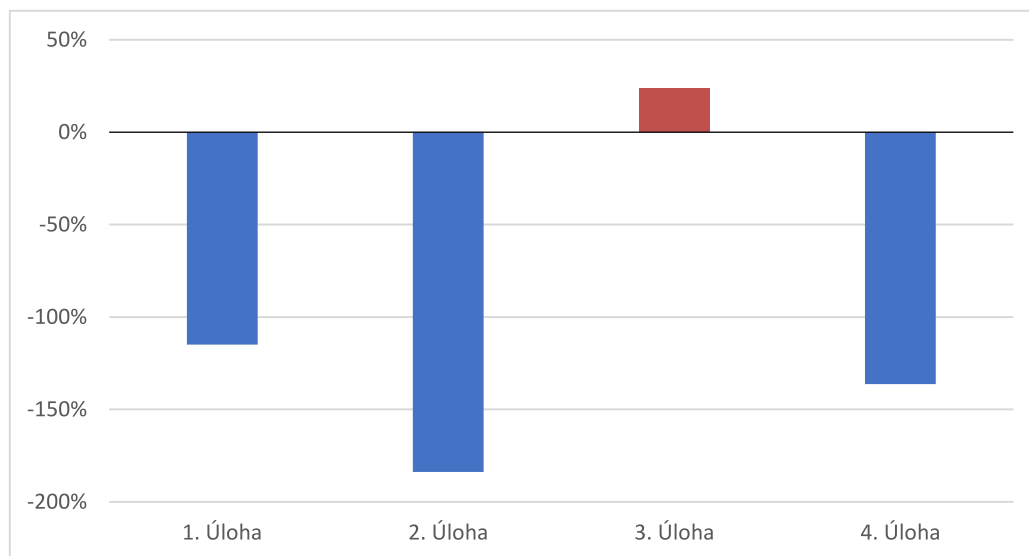
7.2 Efekt implementovaných změn

V této části budou zhodnoceny mnou implementované změny, které cílily na zlepšení uživatelské zkušenosti a efektivity. Jelikož v první iteraci testů testoval rozhraní malý počet účastníků, je možné že změny v naměřených hodnotách nemusí být vypovídající. Zároveň zde budou popsány pouze změny v rozdílech mezi zařízeními, jelikož v první iteraci hodnotili účastníci obě zařízení o dost kritičtěji. Rozdíly budou uváděny v procentuálním nárůstu oproti první iteraci. Procentuální nárůst bude vypočítán podle rovnice 7.1, ve které název zařízení reprezentuje průměrnou hodnotu pro dané zařízení naměřenou v iteraci, kterou určuje číslo indexu.

$$\frac{(HoloLens_2 - Tablet_2) - (HoloLens_1 - Tablet_1)}{|HoloLens_1 - Tablet_1|} \times 100\% \quad (7.1)$$

Naměřené časy se ve většině úloh snížily, především díky tomu, že rozhraní HoloLens 2 běželo s více FPS, ale i díky tomu, že v druhé iteraci byli někteří uživatelé již zblběhlí ve využívání HoloLens 2. První úloha byla díky změnám značně rychlejší a časy se vyrovnaly rozhraní tabletu. Rozdíly ve druhé a čtvrté úloze se zlepšily až příliš i přes to, že mé změny by je neměly moc ovlivňovat. Důvodem je pravděpodobně malý vzorek v první iteraci a v případě druhé úlohy i menší snaha o přesné zarovnání. Rozdíl ve třetí úloze se zhoršil, jelikož uživatelé začali využívat umístění akčního bodu pomocí robota i v rozhraní tabletu, díky čemuž byly o dost rychlejší.

Rozdíly v hodnocení náročnosti se také změnilo k lepšímu ve většině kategorií. Zhoršení hodnocení fyzické náročnosti je jako v předešlém odstavci způsobeno užíváním přidání akčního bodu pomocí robota v rozhraní tabletu, což uživatelům šetřilo fyzicky náročnější za-



Obrázek 7.8: Graf zobrazující procentuální nárůst či pokles rozdílů mezi naměřenými časy pro jednotlivá rozhraní. Negativní hodnoty (modré sloupce) reprezentují změnu rozdílů, která indikuje snížení potřebného času pro vykonání úlohy při užití nové verze rozhraní pro HoloLens 2, zatímco červené indikují zhoršení.

rovnávání akčních bodů. Zhoršení stresové náročnosti nedokáži vysvětlit, ale na obrázku 7.5 je možné vidět, že stres byl u obou zařízení hodnocen velmi nízko a proto bych nepovažoval tuto změnu za zásadní.

V hodnocení uživatelské zkušenosti zaznamenala nová verze rozhraní pro HoloLens 2 nejvíce pozitivních změn, které je možné vidět na obrázku 7.10. Od první iterace došlo k výraznému zlepšení hodnocení atraktivity, efektivity a spolehlivosti. Srozumitelnost se mírně snížila, což možná podle předpokladu zapříčinilo předělání vyvolání akcí, ale nejedná se o zásadní změnu v hodnocení.

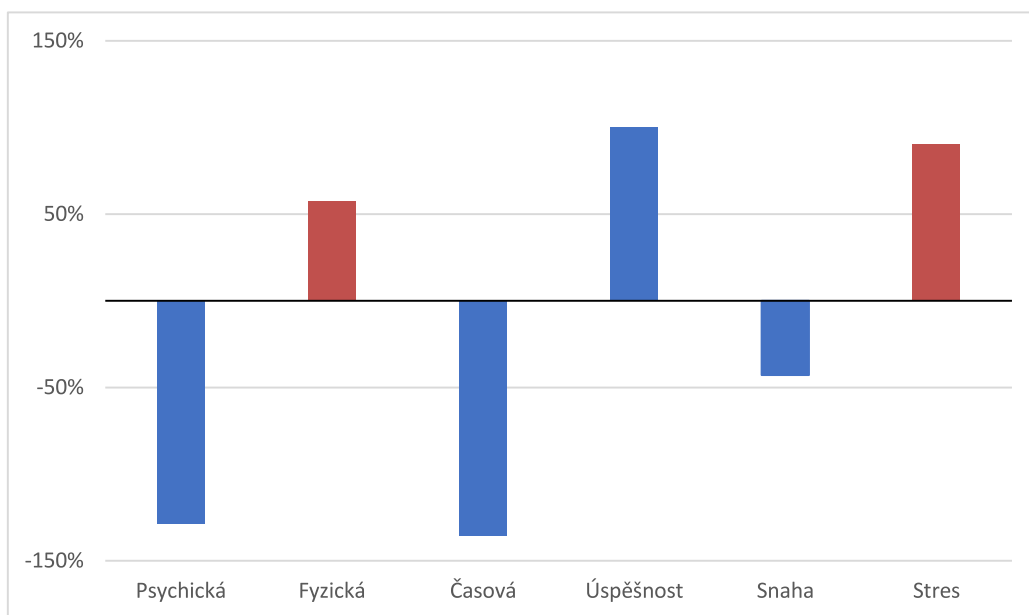
7.3 Návrhy pro navazující práce

Na tuto práci je možné navázat dvěma způsoby. Implementačně rozšířit rozhraní pro HoloLens 2, ať už předěláním existujících funkcí či přidáním nových. Druhý způsob jak navázat na tuto práci, je pokračovat v porovnávání tabletu a HoloLens 2 či jiného HDM.

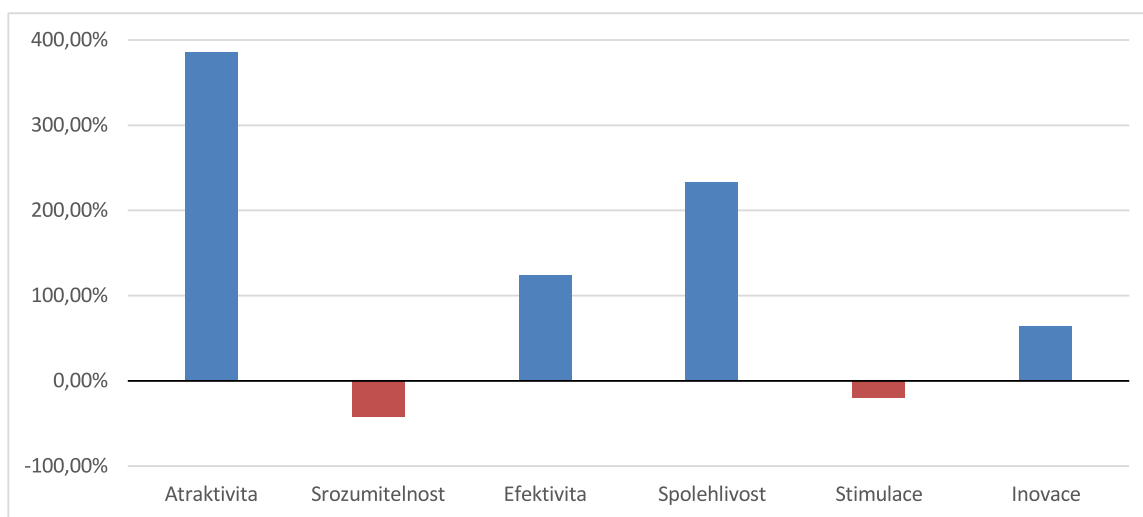
Implementační rozšíření

Ve výsledném projektu stále není implementována plná funkcionalita rozhraní pro tablet. Pokud by někdo navazoval na tuto práci určitě by bylo vhodné implementovat následující položky:

- **Manipulace robota** – Ne s každým robotem je možné snadno manipulovat rukou, jako s robotem využitým v mém experimentu a dokud nebude chyba kalibrace a přesnost HoloLens 2 v jednotkách milimetrů, bude nutné přidávat akce přes koncový efektor robota. Ze stejných důvodů tablet obsahuje řízení robota přes samotné rozhraní, které by bylo vhodné implementovat i pro HoloLens 2 rozhraní. Tuto funkci by bylo možné vyvolat interakcí s koncovým efektozem robota.



Obrázek 7.9: Graf zobrazující procentuální narůst či pokles rozdílů mezi hodnoceními zátěže (NASA-TLX) pro jednotlivá rozhraní. Negativní hodnoty (modré sloupce) reprezentují změnu rozdílů, která indikuje snížení vnímané zátěže při užití nové verze rozhraní pro HoloLens 2. Červené naopak indikují zvýšení. Úspěšnost je opět invertována kvůli překladu.



Obrázek 7.10: Graf zobrazující procentuální narůst či pokles rozdílů mezi hodnoceními uživatelské zkušenosti (UEQ) pro jednotlivá rozhraní. Pozitivní hodnoty (modré sloupce) reprezentují změnu rozdílů, která indikuje zvýšení hodnocení při užití nové verze rozhraní pro HoloLens 2. Červené naopak indikují snížení. Úspěšnost je opět invertována kvůli překladu.

- **Spuštění úlohy** – Jedná se o další funkci z rozhraní tabletu, kterou by bylo vhodné implementovat. Spuštění s počátkem v akci by bylo možné vložit do kontextového menu akce, zatímco spuštění celého programu by pravděpodobně bylo nutné vložit do ručního menu.
- **Zlepšení přehlednosti programu** – Už při docela jednoduché úloze v experimentu si někteří uživatelé stěžovali na přehlednost programu a navrhovali zvýraznění spojů při namíření na akci, což by usnadnilo orientaci v programu. Zároveň by bylo vhodné, kdyby mohl uživatel skrýt některé prvky scény, aby byla přehlednější, a také aby se usnadnil výběr objektů v zákrytu.
- **Notifikační systém** – Rozhraní pro HoloLens 2 stále postrádá notifikační systém, který by uživateli hlásil chyby. Během návrhu jsem sám nad tímto rozšířením přemýšlel, ale došel jsem k názoru, že implementace zobrazování a skrývání zprávy tak, aby uživateli nepřekážela, ale byla dobře viditelná, by byla náročná.

Porovnání tabletu a HoloLens 2

Obě rozhraní jsem v rámci práce porovnával dost povrchově, jelikož jsem se snažil pokrýt celý proces programování robotického pracoviště a zároveň jsem se soustředil více na implementaci, než jsem původně plánoval, jelikož původní řešení vyžadovalo značné množství změn. Nicméně by určitě bylo vhodné porovnávat zařízení i na detailnější úrovni či sbírat více dat, aby z nich bylo možné vyvádět statisticky podložené závěry. Při dalším porovnání bych se zaměřil na tyto body:

- **Výběru objektů** – Jeden z aspektů, který jsem částečně selhal otestovat, byl výběr objektů ve scéně s velkou hustotou objektů, které se překrývají. U tohoto tématu je určitě možné implementovat různé způsoby jak v HoloLens 2 zjednodušit výběr, například pomocí skrývání objektů.
- **Přeprogramování úlohy** – Hlavní myšlenkou AREditoru je snadná editace často měnící se úlohy. Z tohoto důvodu by určitě bylo vhodné testovat tento proces, který by se skládal z pochopení úlohy a její úpravy. Při návrhu jsem se snažil, aby takovýto proces byl v nové verzi jednoduchý, ale úlohy v mém experimentu testovaly vždy jenom jednu funkci rozhraní, a tedy možná úplně neotestovaly tento aspekt rozhraní.
- **Lepší měření pohybu a fyzické náročnosti** – V rámci experimentů jsem kromě subjektivního vnímání fyzické náročnosti měřil i nachozenou vzdálenost po pracovišti. Bohužel jsem kvůli omezenému času a prostředkům musel využít pouze vzdálenost pohybu kamery, která slouží jako aproximace nachozené vzdálenosti. Tato vzdálenost ovšem pravděpodobně není vhodná pro porovnání tabletu a HoloLens 2, jelikož při použití tabletu je kamera v ruce uživatele, zatímco při použití HoloLens 2 je kamera na hlavě. Zároveň tato metrika nebere v potaz rozdíl mezi drobnými pohyby se zařízením a skutečným pohybem kolem pracoviště. Pro skutečné měření pohybu po pracovišti a fyzické náročnosti by bylo možné využít sledování pohybu lidské pózy z videa.

Kapitola 8

Závěr

Cílem této práce bylo porovnat rozhraní pro programování robotů pomocí rozšířené reality implementované pro tablet a Microsoft HoloLens 2. Za tímto účelem byla rozhraní pro toto zařízení otestována v uživatelském experimentu, na základě jehož výsledků byly navrženy a implementovány změny v rozhraní pro HoloLens 2, které cílily na zlepšení uživatelské zkušenosti a efektivity. Vylepšená verze byla otestována a rozhraní byla porovnána v kontextu pro programování robotického pracoviště.

Nejprve byla obě rozhraní otestována uživatelským experimentem, při kterém HoloLens 2 získalo mírně horší hodnocení než rozhraní tabletu. Horší hodnocení pravděpodobně plynulo z chyb v tomto rozhraní.

Na základě zpětné vazby z experimentů bylo navrženo a implementováno značné množství změn. Mezi hlavní změny patří převod na MRTK3, přepracování umístování robotů, přidání akce pomocí robota a omezení nutných přístupů do menu díky programování na základě interakce s objekty ve scéně.

Po implementaci změn byla nová verze opět porovnána s rozhraním tabletu. Hodnocení HoloLens 2 rozhraní se značně zlepšilo oproti hodnocení tabletu. Ukázalo se, že HoloLens 2 jsou značně efektivnější a uživatelé při jejich využití potřebovali průměrně o 32,3 % méně času na vykonání úloh i přes to, že se většina z nich s tímto zařízením teprve seznamovala. Překvapivě byl mezi zařízeními naměřen rozdíl pouze 6,4 % v zátěži uživatelů ve prospěch HoloLens 2 rozhraní. Co se týče uživatelské zkušenosti HoloLens 2 rozhraní, bylo hodnoceno průměrně o 45 % lépe než u tabletu ve všech kategoriích. Podobně jako u uživatelské zkušenosti při hodnocení použitelnosti dosáhlo rozhraní pro HoloLens 2 SUS skóre 77,14, což je o 8,39 více než rozhraní pro tablet a zároveň je to nadprůměrné hodnocení podle SUS benchmarku.

V rámci vyhodnocení byly rozebrány i rozdíly ve výsledcích experimentů po implementaci navržených změn, které byly ve většině případů lepší, ale v první iteraci testů nebyl otestován dostatek účastníků a tyto výsledky tedy nemusejí vypovídat o skutečnosti.

Celkem obě zařízení vyzkoušelo 18 dobrovolníků a většina výsledných hodnot poukazuje na to, že HoloLens 2 je obecně lepší zařízení pro programování robotů. Nicméně stále má mnoho nedostatků, kterými jsou vysoká cena, omezené pokrytí zorného pole, nestabilní detekce rukou a nutnost, aby se s nimi uživatelé naučili zacházet.

Literatura

- [1] AZUMA, R. T. A survey of augmented reality. *Presence: teleoperators & virtual environments*. MIT Press One Rogers Street, Cambridge, MA 02142-1209. 1997, sv. 6, č. 4, s. 355–385. DOI: 10.1162/pres.1997.6.4.355. ISSN 1054-7460.
- [2] BILLINGHURST, M., CLARK, A. a LEE, G. A Survey of Augmented Reality. *Interaction*. 2014, sv. 8, 2-3, s. 103–125. DOI: 10.1561/1100000049. ISSN 1551-3955.
- [3] BROOKE, J. SUS – a quick and dirty usability scale. In: *Usability Evaluation in Industry*. Leden 1996, s. 189–194. ISBN 9780748404605.
- [4] FAUBERT, J. Motion parallax, stereoscopy, and the perception of depth: Practical and theoretical issues. In: SPIE. *Three-Dimensional Video and Display: Devices and Systems: A Critical Review*. 2001, sv. 10298, č. 1029809, s. 168–191. DOI: 10.1117/12.419794.
- [5] GADRE, S. Y., ROSEN, E., CHIEN, G., PHILLIPS, E., TELLEX, S. et al. End-User Robot Programming Using Mixed Reality. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, s. 2707–2713. DOI: 10.1109/ICRA.2019.8793988.
- [6] HIADLOVSKÁ, S. *Programování robotického pracoviště pomocí Microsoft HoloLens 2*. Brno, CZ, 2022. [cit. 2023-05-12]. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií.
- [7] JEFF SAURO, P. *Measuring usability with the system usability scale (SUS)* [online]. Feb 2011 [cit. 2023-05-12]. Dostupné z: <https://measuringu.com/sus/>.
- [8] KALKOFEN, D., MENDEZ, E. a SCHMALSTIEG, D. Comprehensible Visualization for Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*. 2009, sv. 15, č. 2, s. 193–204. DOI: 10.1109/TVCG.2008.96.
- [9] KAPINUS, M., MATERNA, Z., BAMBUŠEK, D., BERAN, V. a SMRŽ, P. *ARCOR2: Framework for Collaborative End-User Management of Industrial Robotic Workplaces using Augmented Reality*. 2023.
- [10] LI, C., ZHENG, P., YIN, Y., PANG, Y. M. a HUO, S. An AR-assisted Deep Reinforcement Learning-based approach towards mutual-cognitive safe human-robot interaction. *Robotics and Computer-Integrated Manufacturing*. 2023, sv. 80, s. 102471. DOI: 10.1016/j.rcim.2022.102471. ISSN 0736-5845. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0736584522001533>.

- [11] MAREK, J. *7 tips to consider when designing for HoloLens* [online]. [cit. 2023-05-12]. Dostupné z: <https://medium.com/inloopx/7-tips-to-consider-when-designing-for-hololens-67d7b09d3e86>.
- [12] MICROSOFT. *About HoloLens 2* [online]. [cit. 2023-05-12]. Dostupné z: <https://learn.microsoft.com/en-us/hololens/hololens2-hardware>.
- [13] MICROSOFT. *Direct manipulation with hands* [online]. [cit. 2023-13-12]. Dostupné z: <https://learn.microsoft.com/en-us/windows/mixed-reality/design/direct-manipulation>.
- [14] MICROSOFT. *Gaze and commit* [online]. [cit. 2023-13-12]. Dostupné z: <https://learn.microsoft.com/en-us/windows/mixed-reality/design/gaze-and-commit>.
- [15] MICROSOFT. *Use your voice to operate HoloLens* [online]. [cit. 2023-13-12]. Dostupné z: <https://learn.microsoft.com/en-us/hololens/hololens-cortana>.
- [16] MILGRAM, P., TAKEMURA, H., UTSUMI, A. a KISHINO, F. Augmented reality: a class of displays on the reality-virtuality continuum. In: International Society for Optics and Photonics. *Telemicroscopy and Telepresence Technologies*. SPIE, 1995, sv. 2351, s. 282 – 292. DOI: 10.1117/12.197321.
- [17] NIKOLOV, S. *Visualization and Validation of Safety Parameters for Industrial Robots Using the HoloLens 2*. 2020. Diplomová práce. Linköping University.
- [18] OSPANIN, M., ZAITSEV, S., SABIROVA, A. a KLIMCHIK, A. Interactive Industrial Robot Programming based on Mixed Reality and Full Hand Tracking. *IFAC-PapersOnLine*. Elsevier. 2022, sv. 55, č. 10, s. 2791–2796. DOI: 10.1016/j.ifacol.2022.10.153. ISSN 2405-8963.
- [19] PERES, S. C., PHAM, T. a PHILLIPS, R. Validation of the system usability scale (SUS): SUS in the wild. In: SAGE Publications Sage CA: Los Angeles, CA. *Proceedings of the human factors and ergonomics society annual meeting*. 2013, sv. 57, č. 1, s. 192–196. DOI: 10.1177/1541931213571043.
- [20] RADKOWSKI, R. a INGEBRAND, J. HoloLens for Assembly Assistance - A Focus Group Report. In: LACKEY, S. a CHEN, J., ed. *Virtual, Augmented and Mixed Reality*. Cham: Springer International Publishing, 2017, s. 274–282. ISBN 978-3-319-57987-0.
- [21] RESEARCH, A. for healthcare a QUALITY. *NASA Task Load Index* [online]. [cit. 2023-05-12]. Dostupné z: <https://digital.ahrq.gov/health-it-tools-and-resources/evaluation-resources/workflow-assessment-health-it-toolkit/all-workflow-tools/nasa-task-load-index>.
- [22] RIEDLINGER, U., OPPERMAN, L. a PRINZ, W. Tango vs. HoloLens: A comparison of collaborative indoor AR visualisations using hand-held and hands-free devices. *Multimodal Technologies and Interaction*. MDPI. 2019, sv. 3, č. 2, s. 23.
- [23] SCHMALSTIEG, D. a HÖLLERER, T. *Augmented Reality - Principles and Practice*. United States: Addison-Wesley Professional, červen 2016. ISBN 9780133153200.

- [24] SCHREPP, M., HINDERKS, A. a THOMASCHEWSKI, J. Applying the User Experience Questionnaire (UEQ) in Different Evaluation Scenarios. In: MARCUS, A., ed. *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience*. Cham: Springer International Publishing, 2014, s. 383–392. ISBN 978-3-319-07668-3.
- [25] SCHREPP, M., HINDERKS, A. a THOMASCHEWSKI, J. Construction of a Benchmark for the User Experience Questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence*. Červen 2017, sv. 4, s. 40–44. DOI: 10.9781/ijimai.2017.445.
- [26] SUZUKI, R., KARIM, A., XIA, T., HEDAYATI, H. a MARQUARDT, N. Augmented reality and robotics: A survey and taxonomy for ar-enhanced human-robot interaction and robotic interfaces. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, s. 1–33.

Obsah přiloženého média

/	
thesis.pdf	Text diplomové práce
tex/	L ^A T _E X zdrojové kódy
video.mov	Video shrnující práci
src/	Zdrojové kódy
build/	Sestavená aplikace
experimentsData	Data z experimentů a jejich vyhodnocení
ObjectiveMeasures.xlsx	Objektivní data
SubjectiveEvaluation.xlsx	Subjektivního hodnocení
UEQ.xlsx	Vyhodnocení UEQ