



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

APLIKACE PRO MICROSOFT HOLOLENS S VYUŽITÍM MOBILNÍHO TELEFONU

APPLICATION FOR MICROSOFT HOLOLENS WITH USE OF MOBILE PHONE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MATEJ STANO

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. DANIEL BAMBUŠEK

BRNO 2019

Zadání bakalářské práce



21677

Student: **Stano Matej**
Program: Informační technologie
Název: **Aplikace pro Microsoft HoloLens s využitím mobilního telefonu**
Application for Microsoft HoloLens with Use of Mobile Phone
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte koncept rozšířené reality a její využití v herním průmyslu. Seznamte se s brýlemi Microsoft HoloLens.
2. Propojte brýle s mobilním telefonem, vyberte vhodné metody a nástroje a navrhňte demonstrační aplikaci, která umožní ovládat virtuální letoun prostřednictvím mobilu (např. pomocí gyroskopu).
3. Navrženou aplikaci implementujte.
4. Proveďte experimenty a vyhodnoťte vlastnosti výsledného řešení.
5. Vytvořte plakát nebo video prezentující klíčové vlastnosti výsledného řešení.

Literatura:

- Dieter Schmalstieg, Tobias Hollerer. *Augmented Reality: Principles and Practice*. Addison-Wesley, 2016. ISBN: 978-0321883575.
- Dále dle pokynu vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a částečně bod 3.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Bambušek Daniel, Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 1. listopadu 2018

Abstrakt

Táto práca sa zameriava na využitie okuliarov Microsoft HoloLens pre rozšírenú realitu a ich prepojenie a komunikáciu s mobilným telefónom za účelom ovládania hologramu. V telefóne sú využité pohybové a polohové senzory ako akcelerometer alebo gyroskop, vďaka ktorým zariadenie slúži ako ovladač produkujúci potrebné dáta. Na základe vyprodukovaných údajov dochádza k pohybu a rotácii objektu vytvoreného okuliarmi. Komunikácia prebieha prostredníctvom internetového spojenia. Implementačná časť sa skladá z dvoch osobitných aplikácií pre jednotlivé zariadenia. Aplikácia pre telefón naväzuje spojenie a dáta získané zo senzorov zasiela okuliarom. Aplikácie pre HoloLens dáta spracúva a následne vytvára a obsluhuje jednotlivé hologramy. Výsledkom je jednoduchá hra pre rozšírenú realitu.

Abstract

This thesis focuses on usage of Microsoft HoloLens head-mounted display for augmented reality and its connection and communication with mobile phone in order to control the hologram. Phone, which uses its motion and position sensors, such as accelerometer or gyroscope, to produce necessary data, represents a wireless controller. Based on this information hologram created by HoloLens is moved or rotated. Communication between devices is maintained through an internet connection. Two separate applications were implemented, each for one of the devices. Mobile app that sends data from sensors to the glasses. HoloLens app that processes received data and controls holograms. The result is simple augmented reality game.

Klíčové slová

Microsoft HoloLens, Rozšírená realita, Unity, Mobilná aplikácia, Gyroskopické ovládanie, Spatial mapping

Keywords

Microsoft HoloLens, Augmented reality, Unity, Mobile app, Gyroscopic control, Spatial mapping

Citácia

STANO, Matej. *Aplikace pro Microsoft HoloLens s využitím mobilního telefonu*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Daniel Bambušek

Aplikace pro Microsoft HoloLens s využitím mobilního telefonu

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Daniela Bambuška. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....

Matej Stano
13. mája 2019

Poďakovanie

Rád by som poďakoval vedúcemu mojej bakalárskej práce, Ing. Danielovi Bambuškov, za jeho užitočné rady a vyhradený čas. Taktiež ďakujem všetkým účastníkom, ktorý sa podieľali na výslednom testovaní tejto práce.

Obsah

1	Úvod	3
2	Rozšírená realita	4
2.1	Definícia rozšírenej reality	4
2.2	Hlavné komponenty rozšírenej reality	4
2.3	Virtuálna vs. rozšírená realita a kontinuum zmiešanej reality	6
2.4	Zobrazovacie techniky a technológie	7
2.5	Typy rozhraní pre rozšírenú realitu	9
3	Microsoft HoloLens	12
3.1	Hardvérové detaily	12
3.2	Rozhranie	13
3.3	Spatial Mapping	14
3.4	Možnosti využitia	15
4	Analýza a návrh riešenia	17
4.1	Existujúce riešenia	17
4.2	Aplikácia pre HoloLens	18
4.3	Mobilná aplikácia	20
4.4	Komunikácia zariadení	22
5	Implementácia	24
5.1	Rozbor andriod aplikácie	24
5.2	Komunikácia a pohyb hologramu	26
5.3	Generovanie checkpointov	28
5.4	Navigácia	29
5.5	Celková funkcionálnosť	30
6	Testovanie	33
6.1	Popis a priebeh testov	33
6.2	Výsledky	34
6.3	Možné vylepšenia a zmeny	36
7	Záver	38
	Literatúra	39
A	Obsah priloženého DVD	41

Kapitola 1

Úvod

Na tému rozšírenej reality (z anglického augmented reality, skratka AR) je v súčasnosti možné naraziť čoraz častejšie. Vďaka za to môže aj rýchlemu pokroku v oblasti elektrotechnológií. Ako z jej názvu vyplýva, rozšírená realita dokáže rozširovať reálny svet o virtuálne prvky. Pomocou nej je možné skĺbiť reálny svet s tým virtuálnym a tým poskytnúť užívateľovi jedinečný zážitok. Na zobrazenie virtuálneho obsahu v reálnom prostredí je však potrebné zariadenie, ktoré to dokáže. Takým zariadením sú aj Microsoft HoloLens.

HoloLens, ako každé iné zariadenie, obsahujú rozhranie, pomocou ktorého s nimi môže užívateľ interagovať a ovládať ich. V tomto prípade ide o gestá a hlasové povely. Cieľom tejto práce je predstaviť iný spôsob ako možno ovládať tento headset a to konkrétne s využitím mobilného telefónu. Nie nadarmo sa o súčasných mobiloch hovorí ako o inteligentných zariadeniach, ktoré dokážu spracovávať a produkovať veľké množstvo informácií vo veľmi malých časových intervaloch. Za také informácie možno považovať aj údaje o rotácii a pohybe telefónu v priestore. Vďaka nim sa mobil v tejto práci mení na určitý typ bezdrôtového ovládača a následne sa využíva pre interakciu s HoloLens.

Ako demonštratívnu ukážku, kde takéto ovládanie dáva zmysel, bola vytvorená hra, v ktorej má užívateľ za úlohu ovládať hologram helikoptéry a zbierať ním virtuálne objekty rozmiestnené po miestnosti.

V nasledujúcej kapitole je popísané čo je to rozšírená realita. Uvedená je jej definícia a rozdieli oproti virtuálnej realite. Kapitola 3 sa zaoberá Microsoft HoloLens. Sú špecifikované hardvérové detaily a bližšie je rozobrané aj ich rozhranie. Priblížená je aj funkcionálnosť snímať okolité prostredie a vytvárať model reálneho sveta, ktorá má veľký význam vo výslednej aplikácii. Kapitola 4 tvorí analýza a návrh aplikácií, ktoré budú v tejto práci vytvárané. Implementáciou sa zaoberá kapitola 5, popisujúca najmä zaujímavejšie časti. Poslednú kapitolu tvorí testovanie a spomenuté sú aj možné vylepšenia výsledného produktu.

Kapitola 2

Rozšírená realita

Táto kapitola sa zaoberá konceptom rozšírenej reality, jej definíciou (2.1) a rozdielom oproti virtuálnej realite (2.3). Ďalej sú rozobraté základné komponenty rozšírenej reality (2.2) a techniky a technológie používané pri zobrazovaní rozšírenej reality (2.4). Posledná časť sa zaoberá typmi rozhrania využívanými pri rozšírenej realite (2.5).

2.1 Definícia rozšírenej reality

Azuma [4] definuje AR ako variáciu virtuálnej reality (VR), kde rozdiel od VR je užívateľovi umožnené vidieť reálny svet okolo neho spolu s virtuálnymi objektami do neho vloženými. AR realitu len dopĺňa, namiesto toho aby ju úplne nahrádzala. V ideálnych podmienkach by sa mohlo zdať, že virtuálne a reálne objekty koexistujú v rovnakom priestore.

Aby AR nebola limitovaná len pre konkrétne technológie, Azuma pokračuje svoju definíciu špecifikovaním rozšírenej reality ako systému, ktorý spĺňa tri charakteristiky:

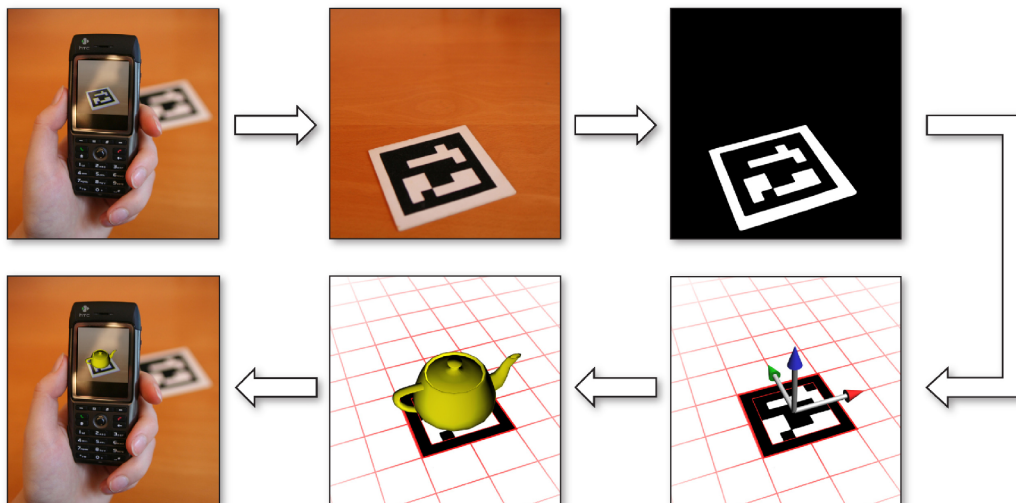
- Kombinuje reálne a virtuálne objekty v reálnom prostredí
- Je interaktívny a beží v reálnom čase
- Všetko je usporiadané a zobrazené v troch dimenziách

Táto definícia umožňuje uvažovať aj iné zariadenia ako len HMD (z anglického head-mounted display) a rovnako tak nelimituje AR len na vizuálne médiá [4].

Takto definovaná AR pre svoj beh vyžaduje priestorovú registráciu a kontrolu v reálnom čase, ktoré zabezpečujú, že zodpovedajúce virtuálne a reálne objekty budú správne usporiadané. To znamená, že užívateľ má kontrolu nad tým kde bude smerovať svoj pohľad a počítačovo generované „rozšírenia“ na displeji zostanú registrované k príslušným objektom v prostredí [21, Kapitola 1].

2.2 Hlavné komponenty rozšírenej reality

Aby bolo možné hovoriť o úplnom AR systéme, sú potrebné aspoň tri hlavné komponenty: tracking, registrácia a vizualizácia. Štvrtý komponent – priestorový model, slúžiaci ako databáza, uchováva informácie o reálnom a virtuálnom svete. Model skutočného sveta je využívaný pri trackingu, ktorý na základe neho určí užívateľovu pozíciu v reálnom svete. Model virtuálneho sveta pozostáva z obsahu, ktorý sa využíva pre virtuálne „rozšírenia“. Obidve



Obr. 2.1: Pipelina marker-based trackingu: Kamera zariadenia zaobstaráva video (*vľavo hore*), ktoré je potom posunuté trackovacímu systému (*stred hore*). Ten následne vykonáva detekciu hrán pomocou tresholdingu (*vpravo hore*), nasledovaného vyhľadávaním štvoruholníkov. Keď je nájdený vzor, určí sa transformácia z roviny kamery do lokálneho súradnicového systému značky (*vpravo dole*), na ktorú sa následne umiestni virtuálny objekt (*stred dole*). Výsledok je zobrazený na displeji zariadenia (*vľavo dole*) [23].

časti priestorového modelu sa musia nachádzať v rovnakom súradnicovom systéme [21, Kapitola 3].

Tracking v kontexte AR predstavuje pojem popisujúci dynamické snímanie a meranie vykonávané systémami pre rozšírenú realitu. Aby bolo možné zobrazit virtuálne objekty naviazané na objekty v skutočnom svete, musíme byť schopný snímať prostredie a sledovať užívateľov pohyb: tri súradnice (x , y , z) pre pozíciu a tri uhly (pozdĺžny sklon, priečny náklon, výchylka/smer) pre orientáciu. Keďže AR operuje v reálnom čase, jednotlivé údaje musia byť aktualizované nepretržite. To akú formu majú výsledné dáta a ako ich aplikácia využíva závisí od použitých techník a technológií. Za zmienku stoja najmä prístupy založené na využití kamier, stereo kamier alebo hĺbkových kamier. Tie získavajú informácie o polohe zariadenia snímaním jeho okolia. Tento typ rozpoznávania nie je jednoduchý, preto sa ako pomôcka využívajú značky umiestnené na cieľné objekty alebo plochy, ktoré majú za úlohu uľahčiť ich detekciu [11] [21, Kapitola 3].

Ide o takzvaný „Marker-based tracking“ (obr. 2.1). Pri tomto prístupe sa ako prvá vykonáva detekcia hrán. Tá pozostáva z tresholdingu a vyhľadávania štvoruholníkov. Nájdené oblasti, ktoré sú buď príliš veľké alebo príliš malé sú automaticky zahodené. Následne sa vnútro zostávajúcich štvoruholníkov normalizuje pomocou perspektívnej transformácie. Výsledné obrazy sa porovnávajú s množinou známych značiek. Keď dôjde k zhode (detekcii vzoru), určí sa prvá, „hrubá“, poloha zariadenia pomocou hrán značiek. Následne sa poloha a rotácia zariadenia doladí iteratívne s využitím maticových výpočtov. Výsledná matica definuje transformáciu z roviny kamery do lokálneho súradnicového systému v strede značky. Vďaka tomu je potom možné zobrazit virtuálne objekty priamo na povrchu markerku [23].

Prístupy, ktoré značky nevyužívajú, vyžadujú lepšiu kvalitu videa a sú značne náročnejšie na výpočet.

Registrácia je proces mapovania virtuálnych objektov na objekty z reálneho sveta. Pri trackingu dostávame niekoľko súradnicových systémov, ktoré musia byť navzájom zosúladené tak, aby bolo zaručené správne prekrytie virtuálnych objektov s tými reálnymi. Úlohou registrácie je previesť tieto údaje, do súradnicového systému výslednej aplikácie.

Aby mohla registrácia prebiehať správne, je potrebné nakalibrovat jednotlivé komponenty zariadenia. Pri kalibrácii dochádza k porovnávaniu nameraných dát dvoch rôznych zariadení – referenčného a toho, ktoré má byť nakalibrované. Cieľom je nájsť také parametre, pri ktorých bude zariadenie produkovať údaje v potrebnej škále. V závislosti od systému, môže byť zariadenie nakalibrované len raz (zvyčajne behom výroby), vždy pred zahájením operácie alebo súčasne s trackingom.

Kalibráciu sledovacieho systému vyžaduje najmä statická registrácia – keď sa kamera alebo užívateľ nehýbe, ktorá má za úlohu stanoviť spoločný súradnicový systém pre virtuálne a reálne objekty. Opačný prípad je dynamická registrácia, prebiehajúca pri pohybe kamery a užívateľa [21, Kapitola 3].

Vizualizácia určuje výslednú podobu zobrazenia. Aby užívateľ nadobudol pocit realistikosti, je potrebné zaručiť vizuálnu súdržnosť prostredia, čo znamená, že virtuálne objekty musia byť kombinované takým spôsobom aby zapadli do reálneho prostredia a v ideálnom prípade boli nerozoznateľné od reálneho sveta. Pre dosiahnutie tohoto cieľa je nevyhnutné dodržať niekoľko pravidiel počítačovej grafiky: relatívna veľkosť – vzdialenejšie objekty sú menšie; relatívna výška – vzdialenejšie objekty majú svoj spodok položený vyššie; perspektíva – súbežné čiary konvergujú ako sa vzdalujú od pozorovateľa; detaily povrchu – na bližších objektoch možno pozorovať detailnejšiu textúru; vplyv atmosféry – vzdialenejšie objekty sa javia viac rozmazané a namodralé; oklúzia – objekty bližšie k pozorovateľovi zakryjú vzdialenejšie objekty; tieňovanie – objekty sú osvetlené vzhľadom na pozíciu a orientáciu svetelného zdroja; tieňovanie – objekty vrhajú tieň na iné objekty. Niektoré časti, ako napríklad veľkosť, výška alebo detaily povrchu, sú priamo produkované s virtuálnou kamerou, ktorá je geometricky registrovaná k tej reálnej. Zvyšné položky, najmä oklúzia, tieňovanie a tieňovanie, vyžadujú špeciálny prístup a ich spracovanie je značne zložitejšie [21, Kapitola 6].

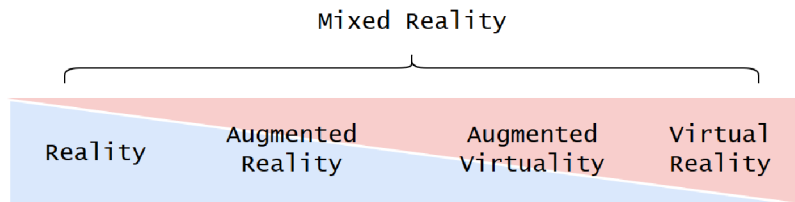
Dôležitým faktorom vizualizácie v súvislosti s rozšírenou realitou je skutočnosť, že je potrebné brať v úvahu aj reálny svet. Preto White a Feiner [24], pre opis výpočtov s kontextom, ktorý je definovaný ako časť fyzickej scény, zaviedli výraz „situated visualization“. Kontextom v tomto prípade nemusí byť len jeden objekt ale čokoľvek od izolovaného bodu v prostredí až po rozsiahle oblasti. Pojem „situated visualization“ teda zahŕňa techniky, ktoré sú pri vizualizačných výpočtoch schopné počítať aj s reálnym svetom.

2.3 Virtuálna vs. rozšírená realita a kontinuum zmiešanej reality

Typický pohľad na virtuálnu realitu je, že ide o vytvorené prostredie v ktorom je užívateľ úplne ponorený a s ktorým môže interagovať. Tento svet je kompletne počítačovo generovaný a môže ale nemusí napodobňovať vlastnosti reálneho prostredia [19].

Pri kombinácii objektov z reálneho sveta s tými virtuálnymi, na rôznych úrovniach, dostávame niečo medzi virtuálnou realitou a realitou. Tento priestor sa nazýva zmiešaná realita (z anglického mixed reality, skratka MR). Mnoho ľudí uprednostňuje pojem „zmiešaná realita“ pred „rozšírenou realitou“ pre jeho väčšiu obecnosť [21, Kapitola 1].

Tento spôsob chápania rozšírenej a zmiešanej reality pochádza od Milgrama a Kishino [19], ktorí navrhli kontinuum popisujúce prechod od reality k virtuálnej realite. Milgram MR opisuje ako kombinovanie reálneho a virtuálneho sveta niekde pozdĺž kontinua, čím dochádza k spájaniu čisto reálneho prostredia s tým virtuálnym (obr. 2.2).

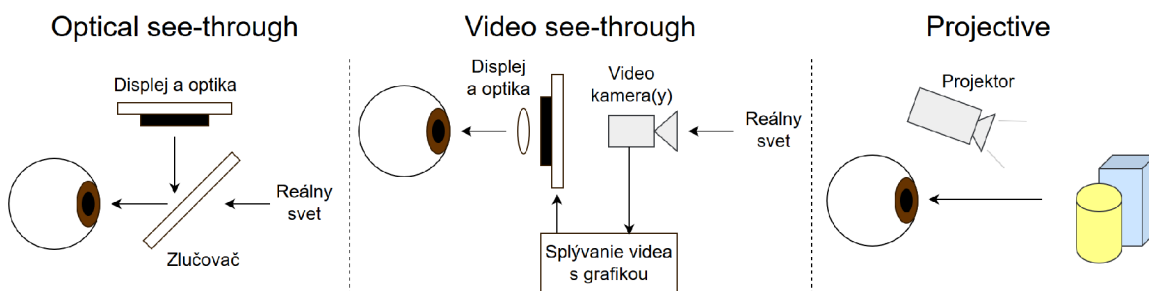


Obr. 2.2: Kontinuum zmiešanej reality zachytávajúce všetky možné kombinácie reálneho a virtuálneho sveta.

Na základe toho možno rozšírenú realitu chápať ako systém, ktorý obsahuje prevažne objekty reálneho sveta a teda je bližšie k realite. Ako príklad možno uviesť hru Pokémon GO¹ pre smartfóny, ktorá do prostredia nasnímaného kamerou pridáva virtuálne objekty. Opačný koncept predstavuje takzvaná „augmented virtuality“, ktorá pozostáva prevažne z virtuálneho prostredia doplneného o malý počet objektov z reálneho sveta. Ako príklad si môžeme predstaviť hru kde sa tvár postavy v reálnom čase nahrádza videom tváre hráča. Všetko okrem tváre je virtuálne [21, Kapitola 1].

2.4 Zobrazovacie techniky a technológie

V zásade existujú tri spôsoby (obr. 2.3) ako vizuálne prezentovať rozšírenú realitu. Najbližšie k virtuálnej realite je metóda nazývaná „video see-through“, pri ktorej užívateľ nevidí skutočný svet priamo, ale len jeho digitalizovanú verziu nasnímanú kamerou. Virtuálne „rozšírenia“ sú následne pridané ako ďalšia vrstva. Opačný prístup predstavuje „optical see-through“, ktorý neovplyvňuje vnímanie reálneho sveta a rozšírenú realitu zobrazuje pomocou priehľadných zrkadiel a šošoviek. Poslednou metódou je premietanie rozšírenej reality priamo na reálne objekty.



Obr. 2.3: Zobrazovacie techniky pre rozšírenú realitu.

Medzi hlavné výhody **video see-through**, v porovnaní so zvyšnými dvoma prístupmi, patrí predovšetkým jednoduchšia implementácia. Keďže je realita digitalizovaná, je menej obtiažne spracovať alebo odstrániť niektoré jej objekty, ako napríklad značky spomínané

¹<https://www.pokemongo.com/en-us/>



Obr. 2.4: *Vľavo*: Optical see-through head-mounted display. *Vpravo*: Video see-through hand-held display.³

v sekcii 2.2. Rovnako tak je jednoduchšie zosúladiť jas a kontrast virtuálnych objektov s reálnym prostredím. Na druhej strane digitalizácia spôsobuje predovšetkým nízke rozlíšenie reality, obmedzené zorné pole a možnú dezorientáciu pozorovateľa v dôsledku rozličnej pozície kamery snímajúcej prostredie a očí.

Metóda **optical see-through** pre zobrazovanie využíva optické prvky, ktoré svetlo čiastočne prepúšťajú aj čiastočne zrkadlia. Za taký prvok možno považovať polopriepustné zrkadlo, ktoré prepustí dostatočné množstvo svetla z reálneho sveta a zároveň dovolí projektoru, umiestneného nad zrkadlom, naň naniesť informácie z virtuálneho sveta. Rozlíšenie reality tým pádom zostáva nezmenené. Nevýhodou však môže byť redukcia jasů virtuálnych aj reálnych objektov, čo má za následok, že sú zariadenia využívajúce tento spôsob zobrazenia menej vhodné pre vonkajšie použitie.

Projekcia hologramov na reálne objekty, úzko súvisí s takzvanou priestorovou rozšírenou realitou (spatial AR). Tento prístup, zobrazuje virtuálne informácie priamo na objekty skutočného sveta. Ako projekčná plocha môže poslúžiť čokoľvek od rovných stien až po komplexné modely. Projekcia nemôže zmeniť tvar objektu, ale môže mu pridať textúru, tieň, detaily a dokonca aj dynamické chovanie [11].

V závislosti od toho v akej vzdialenosti od pozorovateľa sa zobrazovacie zariadenie nachádza, rozlišujeme tri typy displeja: head-attached, hand-held a spatial displej.

Head-attached displej

Ako z pojmu vyplýva, zobrazovacie zariadenie je nasadené na hlave užívateľa. Z toho dôvodu je dôraz kladený najmä na čo najmenšiu váhu zariadenia, no zároveň je potrebné zachovať výkon a veľkosť zorného poľa. V závislosti od toho aká technológia je použitá pre generovanie obrazu rozlišujeme: *sietnicový displej* (retinal display), ktorý využíva nízkoenergetické lasery pre projekciu obrazu priamo na sietnicu oka, *head-mounted projektor*, ktorý nanáša obraz na povrch reálneho prostredia pomocou miniatúrnych projektorov a LCD panelov s podsvietením, alebo *head-mounted displej* (obr. 2.4), využívajúci miniatúrne displeje na-

³Zdroj: <https://www.forbes.com/sites/bernardmarr/2018/07/30/9-powerful-real-world-applications-of-augmented-reality-ar-today/>, <https://www.roadtovr.com/meta-2-development-kit-hands-on-could-do-for-augmented-reality-what-oculus-rift-dk1-did-for-virtual-reality/>



Obr. 2.5: Zariadenie Spatial MRX predstavujúce typ *Spatial Optical See-Through* displeja.⁵

chádzajúce sa pred očami pozorovateľa. Medzi head-mounted, konkrétne optical see-through zariadenia patria aj okuliare Microsoft HoloLens, použité v tejto práci a bližšie rozobraté v kapitole 3.

Hand-held displej

Za typické príklady *hand-held displejov* (obr. 2.4) možno považovať tablety a mobilné telefóny – zariadenia, obsahujúce kameru a displej, vďaka čomu sú ideálne pre použitie s konceptom video see-through. Sledovanie pozície zariadenia je vo väčšine prípadov vykonávané pomocou kamery. Pre väčšiu presnosť, spolu s kamerou, však môžu byť využité aj iné senzory, ako napríklad gyroskop alebo akcelerometer.

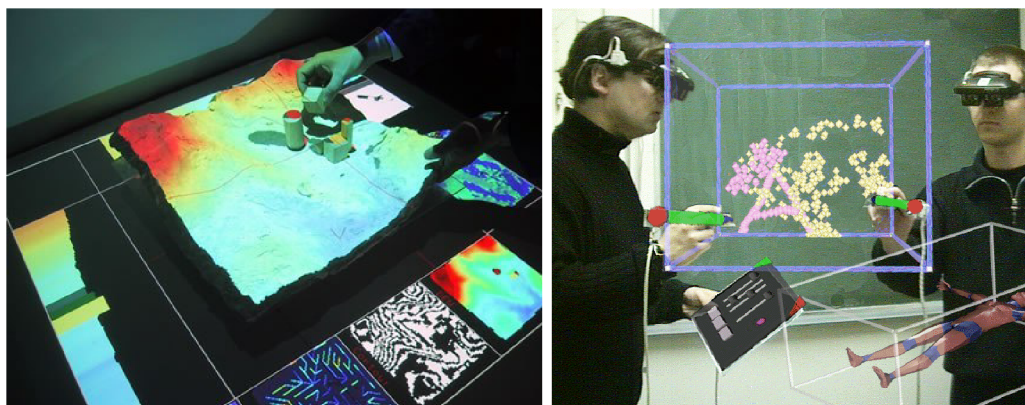
Spatial displej

Narozdiel od vyššie spomínaných displejov, priestorové displeje oddeľujú väčšinu technológií od užívateľa a integrujú ich do prostredia. V závislosti od použitej metódy (video see-through, optical see-through, projekcia) rozlišujeme: *Screen-Based Video See-Through displays* (bežné monitory), *Spatial Optical See-Through displays* (zobrazujú hologramy do reálneho prostredia využívajúc techniky ako optické zlučovanie lúčov so zakrivenými zrkadlami, priehľadné plátna a podobne; obr. 2.5) a *Projection-Based Spatial displays* (pomocou projekcie plynule priemietajú obrazy priamo na povrch fyzických objektov) [5].

2.5 Typy rozhraní pre rozšírenú realitu

Jedným z najdôležitejších aspektov rozšírenej reality je vytvoriť vhodné techniky pre intuitívne interakcie medzi užívateľom a virtuálnym obsahom aplikácie [7]. Na základe podporovaných interakcií a použitých vstupných a výstupných zariadení rozlišujeme štyri typy rozhrania pre rozšírenú realitu: *tangible interface* (hmatateľné rozhranie), *collaborative interface* (kolaboratívne rozhranie), *hybrid interface* (zmiešané rozhranie), *multimodal interface* (multimodálne rozhranie).

⁵Zdroj: <https://vrroom.buzz/vr-news/games/spatial-brings-ar-gaming-tabletop>



Obr. 2.6: *Vľavo*: SandScape – hmatateľné rozhranie pre navrhovanie a pochopenie krajiny prostredníctvom simulácií využívajúcich piesok. *Vpravo*: Príklad kolaboratívneho rozhrania – dvaja užívatelia Studierstube pracujúci spoločne na viacerých aplikáciách.⁷

Tangible AR Interface

Tangible interface (obr. 2.6) je typ rozhrania, ktoré podporuje priamu interakciu s reálnym prostredím využívaním fyzických objektov a nástrojov. Ako príklad možno uviesť stolové „hmatateľné“ rozhranie TaPuMa, využívajúce objekty, ktoré užívateľ nosí so sebou, na interakciu s digitálnou mapou premietanou na povrch stolu. Pomocou týchto objektov je následne možné zobrazit relevantné lokácie alebo informácie na mape. Výhodou takejto aplikácie je, že používaním objektov ako kľúčových slov, sa eliminujú problémy spojené s jazykovou bariérou. Na druhej strane môže mať rovnaký objekt pre viacero ľudí rozdielny význam a preto systém nemusí vždy poskytnúť želaný výsledok. To predstavuje hlavný problém užívateľských rozhraní – znázornenie ako má užívateľ využívať reálne objekty pri interakcii so systémom. Za jedno riešenie možno považovať zobrazovanie virtuálnych rád, napovedajúcich ako sa má s objektom pracovať, priamo na reálnom objekte.

Collaborative AR Interface

Kolaboratívne rozhrania nachádzajú svoje uplatnenie v rámci aktivít vykonávaných vzdialene – v prípade telekonferencií je AR schopná bezproblémovo integrovať niekoľko zariadení s rozličnými lokáciami (remote sharing), ale aj lokálne – 3D rozhrania sú často využívané na zlepšenie spolupráce v rámci spoločného pracoviska (co-located sharing). Ako príklad možno uviesť Studierstube (obr. 2.6), ktoré využíva rozšírenú realitu na premostenie jednotlivých dimenzií užívateľského rozhrania: počet užívateľov, kontext, 3D-okná, platformy, operačné systémy, atď.

Hybrid AR Interface

Hybridné rozhrania kombinujú výber rôznych, no navzájom sa dopĺňujúcich rozhraní. Vďaka svojej flexibilitě sú ideálne pre neplánované, každodenné interakcie, pri ktorých nie je známe v predstihu aký typ displeja alebo zariadenia bude využitý. Takto prispôsobivý systém by mal byť schopný automaticky reagovať na meniacu sa sadu vstupných a výstupných zaria-

⁷Zdroj: <http://www.interactivearchitecture.org/sandscape.html>, https://www.researchgate.net/publication/3872631_Bridging_multiple_user_interface_dimensions_with_augmented_reality

dení a umožňovať tak užívateľovi konfigurovať tok dát medzi týmito zariadeniami a objektami zobrazenými na displeji priamo počas behu.

Multimodal AR Interface

Za multimodálne rozhrania možno považovať také, ktoré sú schopné reagovať na viacero rozdielnych foriem užívateľovho vstupu. Reč je o interakciách, kedy sú informácie premietané na rôzne povrchy a fyzické objekty na základe gést, pohybu rúk, tela, hlavy alebo interakcií so samotnými objektami. Ako príklad možno uviesť typ rozhrania, ktoré na interakciu s prostredím využíva užívateľov pohľad a žmurkanie. Hlavnou výhodou týchto rozhraní je, že na základe kontextu v akom je zariadenie použité, je užívateľ schopný si vybrať formu interakcie, ktorú využije [7].

Multimodálne interakcie podporujú aj Microsoft HoloLens – s hologramami je možné interagovať pomocou pohľadu, gést alebo hlasových príkazov. (viac v časti 3.2).

Kapitola 3

Microsoft HoloLens

Microsoft HoloLens predstavuje prvý bezdrôtový, holografický počítač, ktorý bol uvedený na trh. Je to headset pre zmiešanú realitu, vyvinutý spoločnosťou Microsoft a bežiaci na systéme Windows 10. Zariadenie je schopné snímať okolité prostredie (*spatial mapping*, viac v sekcii 3.3) a následne do neho umiestňovať hologramy. Interakcia s užívateľom môže prebiehať tromi spôsobmi – pohľadom, gestami alebo hlasom.

Prvá generácia zariadenia (obr. 3.1 – vľavo) bola na trh uvedená 30. Marca 2016 v Kanade a Spojených štátoch a neskôr v ďalších krajinách naprieč Európou a Áziou. Headset je dostupný v dvoch verziách – verzii pre vývojárov a drahšej komerčnej verzii [1]. 24. Februára 2019 Microsoft oznámil druhú generáciu zariadenia (obr. 3.1 – vpravo), nazvanú Microsoft HoloLens 2 a vylepšujúcu rôzne aspekty od váhy až po veľkosť zobrazovacej plochy [20].



Obr. 3.1: Prvá (vľavo) a druhá (vpravo) generácia Microsoft HoloLens.¹

Kapitola rozoberá Microsoft HoloLens prvej generácie, použité v tejto práci. Sekcia 3.1 pokrýva hardvérové špecifikácie zariadenie. V sekcii 3.2 sú rozobraté jednotlivé možnosti interakcie. Spatial mapping je popísaný v sekcii 3.3 a možné spôsoby využitia v sekcii 3.4.

3.1 Hardvérové detaily

Optická časť zariadenie pozostáva z priehľadných holografických šošoviek, na ktorých je rozšírená realita zobrazovaná v HD rozlíšení a s pomerom strán 16:9. Headset ďalej ob-

¹Zdroj: <https://www.theverge.com/2016/8/2/12358554/microsoft-hololens-augmented-reality-opens-developer-sales>, <https://xrgo.io/en/microsoft-hololens-2/>



Obr. 3.2: Časti HoloLens headsetu: Optika (vľavo), senzory (stred), základná doska (vpravo).³

sahuje rôzne typy senzorov, vrátane štyroch kamier pre snímanie prostredia a okolitého svetla, jednu hĺbkovú kameru, jednu 2 Mpx foto/video kameru, štyri mikrofóny a jednu IMU (Inertial measurement unit) zahŕňajúcu akcelerometer, gyroskop a magnetometer. Jednotlivé časti možno vidieť na obrázku 3.2.

O výpočty sa stará 32-bitový Intel procesor s podporou TPM 2.0, Microsoft Holographic Processing Unit (HPU 1.0) a 2GB pamäte RAM. Pre aplikácie je prichystané 64GB úložisko. Výdrž batérie sa pohybuje okolo 2-3 hodín pri používaní a do dvoch týždňov v standby móde. Zariadenie je plne použiteľné pri nabíjaní a prehrievaniu zabraňuje pasívne chladenie.

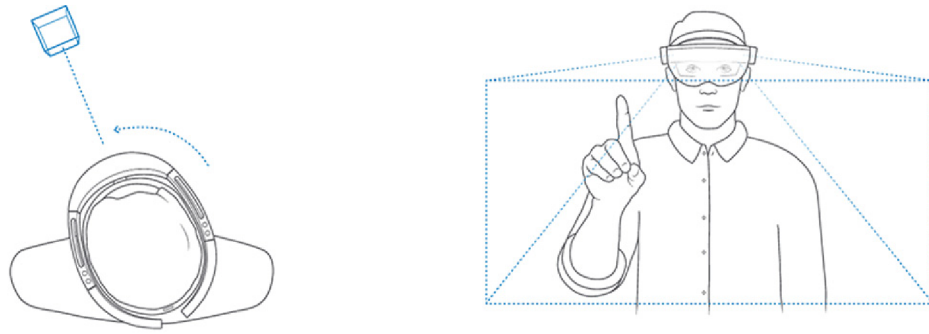
Bezdrôtové spojenie je zaručené prostredníctvom Wi-Fi (IEEE 802.11ac) a Bluetooth 4.1 LE (Low Energy), doplnené o micro USB 2.0 a 3.5mm jack. Zariadenie je taktiež schopné prehrávať zvuk prostredníctvom zabudovaných reproduktorov [14].

3.2 Rozhranie

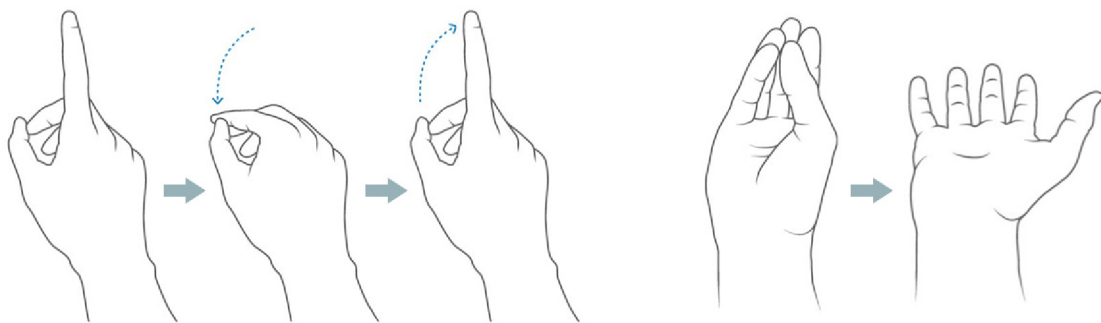
Pre komunikáciu so zariadením môže užívateľ využiť jeden z troch typov interakcie – pohľad (*gaze*), gestá alebo hlas. Všetky interakcie s virtuálnym objektom závisia od užívateľovej schopnosti zamerať sa na tento objekt. Toto je všeobecne docielené pomocou pohľadu (obr. 3.3 – vľavo). Následne je užívateľ schopný s týmto objektom interagovať pomocou gest alebo hlasových pokynov. Microsoft HoloLens v súčasnosti podporujú dve základné gestá – Air tap a Bloom (obr. 3.4). Air tap je gesto ťuknutia s rukou vo vzpriamenej polohe. Ide o ekvivalent kliknutia myšou. V prípade že užívateľ pri vykonávaní gesta zostane v polohe ťuknutia, môže air tap poslúžiť aj pre premiestňovanie hologramov, prechádzanie („skrolovanie“) stránok alebo priblíženie. Bloom predstavuje takzvané „domovské“ gesto a je rezervované pre vyvolanie špeciálnej systémovej akcie, ktorá slúži na návrat do štartovacieho menu. Gestá môžu byť vykonávané ľavou aj pravou rukou, je však potrebné aby sa nachádzali v priestore v ktorom ich kamery, určené na ich snímanie, dokážu zachytiť (väčšinou ide o oblasť medzi ramenami, od nosa po pás; viz. obr. 3.3) [13].

Posledným podporovaným typom interakcie sú zvukové povely. S ich využitím môže užívateľ dosiahnuť podobných výsledkov ako pomocou gest. Windows Mixed Reality využíva takzvaný „See it, say it“ hlasový model, ktorý zaručuje, že označenia na tlačidlách sú rovnaké ako priradené hlasové príkazy. Vďaka tomu dokáže užívateľ ľahko určiť čo povedať aby zariadenie vykonalo želané úkony. Príkazy rezervované systémom, a ktoré by

³Zdroj: <https://docs.microsoft.com/sk-sk/windows/mixed-reality/hololens-hardware-details>



Obr. 3.3: *Vľavo*: Využitie pohľadu na zvýraznenie objektov pre ich následný výber pomocou gést alebo hlasových príkazov. *Vpravo*: Priestor v ktorom je zariadenie schopné snímať géstá.



Obr. 3.4: Dve základné géstá pre ovládanie Microsoft HoloLens – Air Tap (*vľavo*) a Bloom (*vpravo*).⁵

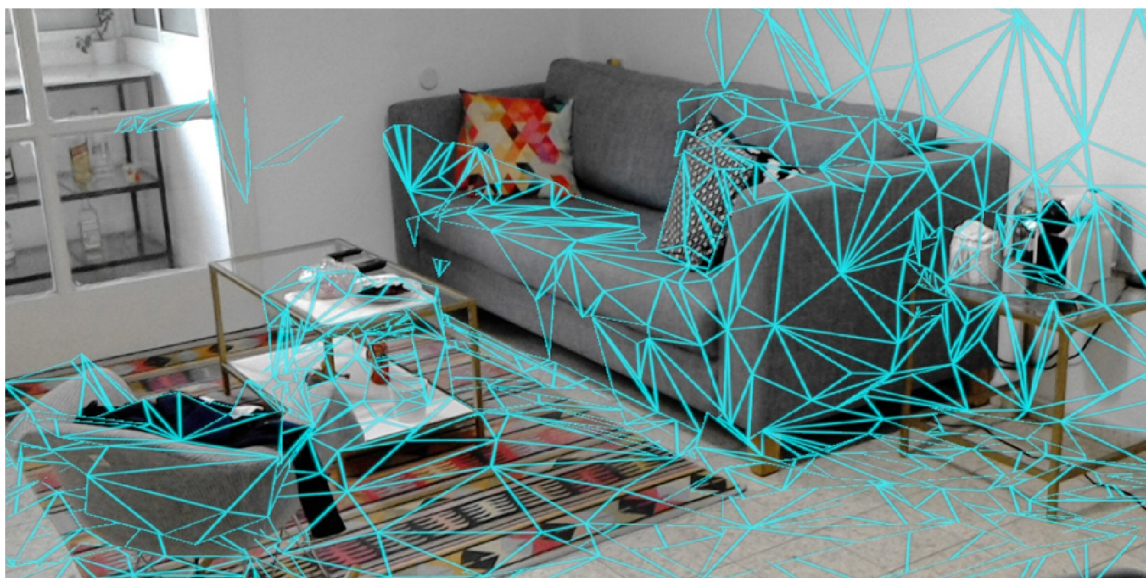
užívateľ nemal implementovať do svojich aplikácií, sú „Select“, ktorý je ekvivalentom pre air tap a „Hey Cortana“, ktorý slúži pre komunikáciu s virtuálnym asistentom vytvoreným Microsoftom [17].

3.3 Spatial Mapping

Vďaka senzorum, ktorými HoloLens disponuje je zariadenie schopné neustále sledovať a mapovať okolité prostredie. Táto funkcionálna, poskytujúca detailnú reprezentáciu povrchov v prostredí okolo zariadenia, sa nazýva *spatial mapping*. Vďaka nej sú hologramy schopné kolidovať s objektami reálneho sveta, čo len umocňuje ilúziu, že sa virtuálne a reálne objekty nachádzajú v spoločnom priestore [15].

Kamera určená na skenovanie sníma okolie v 70 stupňovom kuželi a vo vzdialenosti od 0,8m (aby sa zabránilo skenovaniu užívateľových rúk) do 3,1m. To ako sa užívateľ hýbe prostredím/otáča teda značne ovplyvňuje výslednú kvalitu priestorového modelu. Za ďalšie model-ovplyvňujúce aspekty možno považovať svetelné podmienky a materiály povrchu – rôzne druhy povrchov odrážajú rôzne množstvo svetla (niektoré materiály sú príliš tmavé na to aby boli naskenované, prípadne veľmi lesklé objekty môže byť zariadenie schopné namapovať len z určitých uhlov) [16].

⁵Zdroj: <https://support.microsoft.com/cs-cz/help/12644/hololens-use-gestures>



Obr. 3.5: Spatial mapping mesh pokrývajúci izbu.⁶

Za typické scenáre využitia spatial mappingu možno považovať umiestňovanie hologramov na reálne objekty, prekrývanie hologramov objektami skutočného sveta, simulovanie fyziky, navigovanie hologramov cez reálne prostredie alebo vizualizácia priestorového modelu – vo väčšine prípadov je reprezentovaný ako trojuholníková sieť (triangle mesh, obr. 3.5) [15].

3.4 Možnosti využitia

Zariadenie ako Microsoft HoloLens má potenciál byť využité v rôznych odvedviach. Jeho hlavná náplň – vizualizácia hologramov a informácií, môže byť použitá napríklad pri diaľkovej inštrukcii, kedy by zariadenie pri domácich opravách zobrazovalo postupný návod priamo od experta [10]. Matthew Hanna a spol. tento koncept vyskúšali v praxi, pri vykonávaní pitvy, počas ktorej boli účastníci diaľkovo inštruovaní odborníkom pomocou diagramov, anotácií a hlasových inštrukcií. „Telepatológia“ prebiehala počas hrubého vyšetrenia a v čase intraoperačnej konzultácie, čo umožňovalo operujúcim vzdialene komunikovať s patológom a zároveň virtuálne znázorňovať miesta záujmu na exemplári [9].

*Trimble*⁷ pracuje na integrácii zariadenia do práce architektov a stavebných inžinierov, ktorý by mohli HoloLens využívať pri projektovaní ale aj v neskorších fázach, kedy by sa hologrami integrovali do už rozostavaných budov a slúžili tak ako pomôcka pri výstavbe (obr. 3.6 – hore) [6].

Spolupráca Microsoftu s NASA vyústila k vytvoreniu softwaru nazývaného OnSight, ktorý umožňuje vedcom a inžinierom na základe dát získaných z Curiosity Mars Rover preskúmať Mars prostredníctvom Microsoft HoloLens (obr. 3.6 – vpravo) [18].

HoloLens má svoje využitie aj v automobilovom priemysle, kde automobilové spoločnosti, ako napríklad Volvo, umožňujú zákazníkom nahliadnuť na modifikácie prostredníctvom 3D modelu auta. Ford na druhej strane integruje HoloLens do procesu vývoja –

⁶Zdroj: <https://forums.hololens.com/discussion/6340/miniature-map>

⁷<https://mixedreality.trimble.com/>



Obr. 3.6: Spôsoby využitia Microsoft HoloLens v reálnom prostredí. *Hore*: SketchUp Viewer od spoločnosti Trimble vytvára nové spôsoby vizualizácie, dizajnu a spolupráce na projektoch. *Vlavo*: Využitie na študijné účely. *Vpravo*: OnSight využívajúci dáta z roveru na vytvorenie 3-D simulácie prostredia Marsu.¹²

dizajnéri sú schopní vidieť jednotlivé zmeny prostredníctvom hologramov na fyzickom vozidle.

V rámci medicíny boli Microsoft HoloLens zavedené v nemocnici St Mary v Londýne. Technológia mapuje informácie z CT prístroja na telo pacienta a zobrazuje tak presnú lokáciu krvných ciev. Vďaka tomu je operujúci schopný jednoduchšie určiť lokáciu v ktorej treba rezať [6].

Zariadenie od Microsoftu možno nájsť aj v hernom priemysle (hry ako *RoboRaid*⁸ alebo *Fragments*⁹), alebo využívané vo výučbe – Case Western Reserve University v Clevelande¹⁰ pomocou HoloLens umožňuje študentom zobraziť 3D hologramy ľudského tela a prechádzať jednotlivými vrstvami anatomických štruktúr, od svalov až po obehové systémy a štruktúru kostí (obr. 3.6 – vlavo) [6] [10].

⁸<https://www.microsoft.com/en-us/p/roboraid/9nblggh5fv3j?activetab=pivot:overviewtab>

⁹<http://www.asobostudio.com/games/fragments>

¹⁰<http://case.edu/hololens/>

¹²Zdroj: <https://news.microsoft.com/transform/design-revolution-microsoft-hololens-mixed-reality-changing-architects-world/>, https://casemed.case.edu/cwrumed360/stories/?news_id=481&news_category=1, <https://www.jpl.nasa.gov/news/news.php?feature=4451>

Kapitola 4

Analýza a návrh riešenia

Pred samotnou implementáciou je potrebné si ujasniť ako má vyzerat' výsledná aplikácia. Základom je znalosť požiadavkov a ich rozbor, aby bolo jasné aké operácie má byť schopná aplikácia vykonávať a aké kritéria má spĺňať.

Cieľom tejto práce je využiť pohybové senzory mobilu na ovládanie hologramu zobrazeného v HoloLens. Telefón poslúži ako bezdrátový ovladač, na základe ktorého rotácie sa bude hologram pohybovať. Z toho vyplýva, že je potrebné vytvoriť dve samostatné aplikácie, ktoré budú schopné navzájom komunikovať. Jednu pre mobilné zariadenie, druhú pre samotné HoloLens.

V tejto kapitole sú najprv spomenuté existujúce riešenia, podobné tomu čo je spracovávané v tejto práci – 4.1. Následne sú v častiach 4.2 a 4.3 rozobraté návrhy jednotlivých aplikácií. V časti 4.4 sa ďalej rozoberá spôsob komunikácie medzi zariadeniami.

4.1 Existujúce riešenia

Gyroskop sa spolu s akcelerometrom využíva v mobilných telefónoch už dlhšiu dobu. Či už pre obyčajnú zmenu orientácie obrazovky, ako pomoc pri vytváraní 360 stupňovej fotografie, alebo v rôznych mobilných hrách. Pojem rozšírená realita sa v kontexte mobilných telefónov takisto spomína čoraz častejšie (aplikácie ako *Pokémon GO*, *WallaMe*¹ a mnohé ďalšie). Avšak využitie polohových a pohybových sensorov mobilu pre ovládanie hologramov v Microsoft HoloLens už tak časté nie je. Oproti zariadeniam pre virtuálnu realitu (*Oculus Rift*, *HTC Vive* a ďalšie) HoloLens neposkytujú až tak rozsiahlu kolekciu hier a svoje využitie nachádzajú najmä v iných sférach (viz. 3.4).

Ak sa sústredíme len na prepojenie mobilného telefónu s HoloLens, môžeme hovoriť o aplikácii *HoloStream*² pre iOS zariadenia, vďaka ktorej je možné streamovať pohľad z HoloLens priamo do iPhone alebo iPadu. V rámci opačne orientovanej komunikácie by sa dalo povedať, že je väčšina dohľadateľných aplikácií skôr experimentálneho typu. V [12] sa podarilo pomocou ARKitu³ premeniť iPhone na pohybový ovladač a využiť ho pri ovládaní hologramu zbrane. Alebo v [8] pomocou telefónu vytvárať a následne rotovať objekty zobrazené v HoloLens.

¹<http://walla.me/>

²<https://www.holostream.co.uk/>

³<https://developer.apple.com/arkit/>

4.2 Aplikácia pre HoloLens

Aplikácia pre HoloLens tvorí základnú časť celej práce a z väčšej časti určuje ako bude výsledok vyzeráť. Preto je pri návrhu potrebné si uvedomiť čo má byť jej výstupom. Zadanie udáva, že pomocou mobilu má byť užívateľ schopný ovládať nejaký lietajúci objekt. To o aký objekt pôjde značne ovplyvní ako sa budú dáta z mobilu interpretovať a ako sa bude objekt vo výsledku ovládať a pohybovať.

Ovládaný objekt

Do úvahy pripadajú tri typy objektu: helikoptéra, lietadlo alebo vesmírna loď. Hlavnou nevýhodou lietadla je, že je neustále v pohybe. Užívateľ by ho tak musel sústavne sledovať, aby nedošlo ku kolízii, čo by mohlo miestami pôsobiť frustrujúco. Problém je aj v zložitejšom ovládaní. Vzhľadom na to, že nie je možné zmeniť smer pohybu hneď o 180 stupňov, je manévrovanie v prostredí značne komplikovanejšie.

Helikoptéra a vesmírna loď sa od seba svojim pohybom veľmi nelíšia. Obidve majú dobrú manévrovateľnosť, keďže sa môžu hýbať do ktoréhokoľvek smeru a sú schopné nehybne sa vznášať v prostredí. Rozdielom je len ich vzhľad. Rozhodol som preto použiť helikoptéru, ktorá by mohla viacej zapadnúť do reálneho sveta okolo užívateľa.

Cieľ a funkčnosť

Okrem toho aký objekt bude užívateľ ovládať, je potrebné určiť aj čo s týmto objektom bude robiť, a teda čo bude náplňou aplikácie. Keďže sa jedná o rozšírenú realitu, užívateľ má možnosť pohybovať sa po prostredí spolu s helikoptérou, akoby to bol reálny model. Aby náplňou nebolo len bezcieľne navigovanie hologramu po miestnosti, aplikácia bude do prostredia generovať aj náhodne umiestnené objekty (checkpointy), ktoré bude mať užívateľ za úlohu zbierať v určitom časovom intervale. Pre lepšiu orientáciu možno zobrazovať viac checkpointov naraz, pričom aktívny a zvýraznený (iná farba ako ostatné) bude vždy len jeden. Takto bude mať užívateľ predstavu kde po dosiahnutí práve aktívneho checkpointu možno očakávať ďalší.

Okrem ovládania helikoptéry je potrebné vyriešiť aj celkové ovládanie a možnosti aplikácie a to akú funkcionalitu bude poskytovať. S cieľom zaručiť, že bude výsledné používanie čo najpohodlnejšie, by aplikácia mala podporovať nasledujúce úkony:

- *pozastavenie hry* – zastavenie pohybu helikoptéry a nereagovanie na dáta posielané mobilom
- *spustenie novej hry* – zrušenie aktuálnych checkpointov a vygenerovanie nových, vynulovanie skóre
- *reštartovanie polohy helikoptéry* – umožnenie užívateľovi premiestniť helikoptéru naspäť do jeho blízkosti, najmä v prípade, že bola zničená, ale aj ak sa príliš vzdialila od užívateľa
- *reštartovanie aktívneho checkpointu* – umožnenie užívateľovi premiestniť checkpoint, najmä ak sa vygeneroval na zle dostupnom mieste.

V HoloLens sa aplikácie zväčša ovládajú pomocou gést alebo hlasových príkazov. Keďže v tejto práci mobil funguje ako ovladač, ktorý má užívateľ neustále v rukách, bude výhodnejšie, ak sa na ovládanie aplikácie nevyužije rozhranie HoloLens (gestá, hlasové príkazy),



Obr. 4.1: Návrh rozhrania aplikácie. *Vľavo*: Hologramy helikoptéry a checkpointov – zelený je práve aktívny. V ľavom dolnom rohu sa nachádza ukazovateľ skóre, v pravom zostávajúci čas. *Vpravo*: Návrh navigácie, ktorá sa zobrazí v prípade, že sú objekty „offscreen“. Šípky smerujúce k sledovanému objektu sa zobrazujú pozdĺž hrán displeja. Sledovaný je len aktívny checkpoint a helikoptéra. Informačný text sa zobrazí len ak dôjde k zničeniu helikoptéry alebo odpojeniu telefónu.

ale to, ktoré poskytne mobilná aplikácia (viac v časti 4.3). To aké úkony, nie len v rámci pohybu helikoptéry, bude aplikácia vykonávať tak bude záležať od údajov, ktoré budú prichádzať z mobilného telefónu.

Užívateľské rozhranie

V rámci rozhrania tejto aplikácie je kľúčové určiť ako bude zaobchádzať s objektami ktoré sú „offscreen“ a akým spôsobom bude predávať prípadné informácie užívateľovi. Pri HoloLens je nutné počítať so značne malým zorným polom ($30^\circ \times 17.5^\circ$ ⁴). Môže sa tak veľmi ľahko stať, že objekty zmiznú užívateľovi z dohľadu a ich hľadanie by v niektorých momentoch mohlo byť pomerne náročné a zdĺhavé. Preto je potrebné vytvoriť navigáciu, aspoň v prípade helikoptéry a aktívneho checkpointu, a uľahčiť tak ich lokalizáciu, v prípade, že zmiznú zo zorného poľa zariadenia. Pre tento účel poslúžia jednoduché šípky, smerujúce k objektu, ktoré sa budú zobrazovať po krajoch zobrazovacej plochy kedykoľvek dôjde k strate hologramu. S ich pomocou by malo byť lokalizovanie hľadaného objektu otázkou pár sekúnd.

Keďže, ako bolo spomínané, zobrazovacia plocha zariadenia nie je najväčšia, je podstatné určiť aj ako a aké informácie sa budú zobrazovať. Okrem navigačných šípok sa v ľavom dolnom rohu displeja bude zobrazovať aj skóre, teda počet checkpointov, ktoré sa užívateľovi podarilo pozbierať, a v pravom rohu zostávajúci čas. Rovnako tak by mala aplikácia užívateľa informovať o tom či existuje spojenie medzi mobilom a HoloLens, prípadne či došlo ku kolízii a zničeniu helikoptéry. Tieto texty možno zobrazovať v strede displeja, keďže v oboch prípadoch nebude možné ovládať helikoptéru a tak nebudú prekážať vo výhľade.

⁴<https://uploadvr.com/hololens-2-field-of-view/>

Integrácia okolitého prostredia

Vkladanie virtuálnych objektov do reálneho sveta je základ rozšírenej reality. S prostredím okolo užívateľa je nutné počítať aj pri navrhovaní aplikácie. Aby sa prehĺbil dojem, že sa virtuálne a skutočné objekty nachádzajú v jednom priestore, musia byť spolu schopné interagovať. Reč je o kolíziach helikoptéry a checkpointov s prostredím. Na to poslúži *spatial mapping* (3.3), vďaka ktorému bude aplikácia schopná vytvoriť model sveta, ktorý bol nasnímaný a následne s ním pracovať. Helikoptéra tak nebude prelietavať cez steny, ale naopak bude schopná reagovať na reálne objekty, s ktorými príde do kontaktu – toto poslúži k tomu, že helikoptéra vybuchne ak narazí do objektov reálneho sveta. Model prostredia poslúži aj pri vytváraní checkpointov, tie sa budú generovať na náhodných miestach, no vďaka *spatial mappingu* bude možné kontrolovať aby sa nezobrazovali v stenách, v iných objektoch alebo mimo nasnímanej plochy.

4.3 Mobilná aplikácia

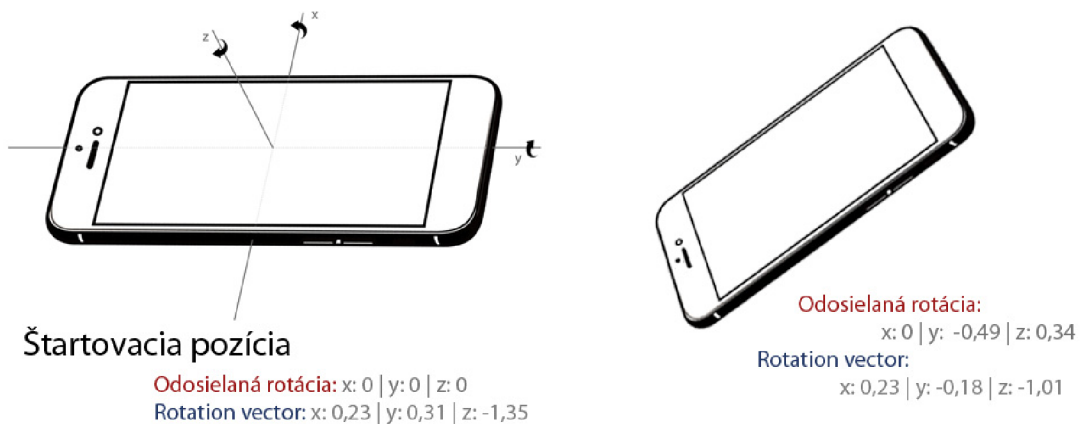
Aby bolo možné ovládať hologram v HoloLens, musí byť mobil schopný poskytovať potrebné údaje, na základe ktorých bude pohyb vykonávaný. Reč je o jeho pohybe a rotácii v priestore. Aplikácia musí vedieť tieto informácie získať, spracovať a následne ich poslať okuliarom. Ako už bolo spomínané v časti 4.2, okrem ovládania hologramu sa bude telefón starať aj o ovládanie samotnej hry (pauza, reštart a ďalšie). Výsledná aplikácia by tak mala pozostávať z dvoch častí – veľmi jednoduchej úvodnej obrazovky, prostredníctvom ktorej bude možné pripojiť telefón k HoloLens, prípadne vykonať nastavenia v rámci mobilnej aplikácie a obrazovky rozhrania s ovládaním (obr. 4.3).

Ovládanie pomocou senzorov

Súčasný mobilné telefóny sú vybavené rôznymi senzormi, ktoré dokážu sledovať polohu, pohyb alebo rozličné enviromentálne podmienky. Tieto senzory sú schopné poskytovať surové dáta s veľkou presnosťou. Aplikácia v tejto práci bude vyvíjaná pre zariadenia bežiacie na systéme Android a využívať bude len senzory na sledovanie pohybu. Tie možno rozdeliť na dva typy – *hardware-based* (sú to fyzické komponenty zabudované priamo do zariadenia, ktoré získavajú svoje dáta priamo meraním špecifických enviromentálnych vlastností, ako napríklad akcelerácia, intenzita geomagnetického poľa alebo zmena uhlov) a *software-based* (nie sú to fyzické súčiastky, dáta získavajú z jedného alebo viacerých *hardware-based* senzorov a niekedy sú nazývané aj virtuálne alebo syntetické senzory).

Na zariadeniach so systémom Android je možné nájsť dvoch predstaviteľov *hardware-based* pohybových senzorov – akcelerometer, ktorý v m/s^2 meria akceleračnú silu, pôsobiacu na zariadenie vo všetkých troch fyzických osiach (x, y, z), vrátane gravitačnej sily. A gyroskop, ktorý poskytuje údaj o rýchlosti rotácie v rad/s , rovnako v rámci týchto troch osí [3].

Aj keď by na ovládanie hologramu stačili len dáta z jedného alebo druhého senzora, je praktickejšie využiť ich kombináciu. To zaručí väčšiu presnosť a čiastočnú elimináciu okamžitých hodnotových výchylok. Virtuálne senzory, ktoré kombinujú dáta z akcelerometra a gyroskopu sú *Rotation vector* a *Game rotation vector*. Obidva poskytujú informáciu o rotácii zariadenia v priestore. Hlavným rozdielom medzi týmito senzormi je, že *Rotation vector* využíva aj dáta z magnetometra a preto udáva orientáciu v takzvanom „East-North-Up“ súradnicovom systéme (X smeruje k východu, Y k severu a Z smerom k nebu) [2].



Obr. 4.2: Obrazné porovnanie údajov získaných z *Rotation vector* senzora s tými, ktoré sa odosielajú HoloLens. Pri nadviazaní spojenia sa uloží aktuálna rotácia telefónu ako štartovacia (vľavo). Tá so potom odčítava od aktuálnych súradníc a výsledok je posielaný HoloLens. To zaručí, že ovládanie nebude naviazané na konkrétne rotácie telefónu.

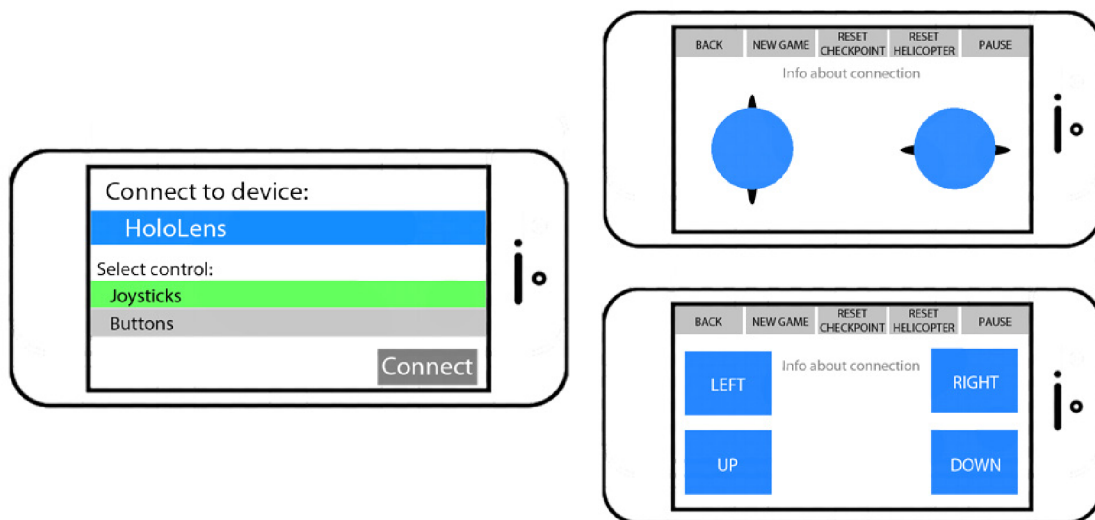
Aby ovládanie nebolo závislé od konkrétnej pozície telefónu a aby sa uľahčili výpočty v HoloLens aplikácii, je lepšie merať veľkosť rotácie telefónu v závislosti od nejakej „štartovacej“ pozície (obr. 4.2) – v určitom okamihu, napríklad pri pripojení mobilu k HoloLens, dôjde k uloženiu aktuálnej pozície mobilu. Následne sa bude pohyb a rotácia udávať relatívne k tejto pozícii. V takomto prípade nezáleží, ktorý zo senzorov (*game rotation vector* alebo *rotation vector*) bude použitý.

Taktiež je potrebné mať na pamäti, že pri rozšírenej realite chce mať užívateľ možnosť ľubovoľne sa prechádzať po prostredí. Hologram sa teda nemôže otáčať v závislosti od toho ako je otočený užívateľ, čo znamená, že je nutné vynechať rotácie okolo osi z a sledovať len otáčanie okolo osí x a y (obr. 4.2). Rotácia okolo osí x poslúži pre náklon a pohyb doprava a doľava, rotácia okolo osi y je pre náklon a pohyb dopredu a dozadu. Pre otáčanie helikoptéry a pohyb hore a dole sú potrebné tlačidlá.

Ovládanie pomocou tlačidiel

Dáta zo senzorov nezvládnu pokryť všetky druhy pohybov. Pokyny pre stúpanie a klesanie alebo otáčanie doľava a doprava je potrebné získavať inak. Po vzore herných ovladačov je najlepšie využiť samostatné tlačidlá pre každý z týchto pohybov. Prípadne to možno doplniť o dva joysticky (hore a dole, doľava a doprava) a umožniť užívateľovi vybrať typ ovládania, ktorý mu viac sedí. Tieto prvky, či už tlačidlá alebo joysticky, by mali zaberat väčšinu obrazovky, aby boli dostatočne veľké a užívateľ nemal problém s ich používaním aj bez toho aby sa díval na obrazovku.

Okrem ovládania helikoptéry treba myslieť aj na ovládanie samotnej hry. Keďže telefón slúži ako ovladač, ktorý má užívateľ neustále v ruke, je možné aplikáciu v HoloLens ovládať pomocou neho. Nesmú preto chýbať tlačidlá pokrývajúce funkcionality spomínanú v časti 4.2 – pozastavenie súčasnej alebo spustenie novej hry, zresetovanie pozície helikoptéry,



Obr. 4.3: Návrh rozloženia jednotlivých prvkov aplikácie. *Vľavo*: Úvodná obrazovka slúžiaca pre pripojenie k HoloLens a výber typu ovládania. *Vpravo*: Obrazovka s ovládaním pomocou joystickov (hore) alebo tlačidiel (dole), spolu s tlačidlami pre ovládanie hry.

ak bola zničená, prípadne sa dostala príliš ďaleko, alebo premiestnenie aktívneho checkpointu. Zoznam môže ešte doplniť tlačidlo späť, ktoré preruší komunikáciu a vráti mobilnú aplikáciu na úvodnú obrazovku. Tu bude možné nastaviť typ ovládania a zadať zariadenie ku ktorému sa má mobil pripojiť. Všetky tieto prvky budú tvoriť výslednú aplikáciu. Návrhy jednotlivých častí sú zobrazené na obrázku 4.3.

4.4 Komunikácia zariadení

Veľmi dôležitou časťou návrhu je určiť spôsob akým budú aplikácie spolu komunikovať. Podľa špecifikácií v časti 3.1, HoloLens dokážu bezdrátovo nadviazať spojenie s iným zariadením dvoma spôsobmi: pomocou Bluetooth alebo WiFi. Aj keď má Bluetooth výhodu, že sa zariadenia spájajú priamo a tak ich je možné spárovať kdekoľvek, tak som sa nakoniec rozhodol pre vytváranie spojenia cez WiFi, pomocou internetových soкетов. Hlavnými dôvodmi bolo, že v dobe riešenia tejto časti som nevedel dohľadať dostatočne dobrú dokumentáciu pokrývajúcu túto tému. Naopak informácie, ktoré sa mi podarilo nájsť zväčša riešili problematiku pripojenia prostredníctvom Bluetooth.

Pri internetovom spojení je potrebné určiť, ktoré technológie budú využité a to najmä v kontexte komunikácie na transportnej vrstve. Na výber je User Datagram Protocol (UDP) alebo Transmission Control Protocol (TCP). Keďže pri UDP komunikácii nedochádza k opätovnému zasielaniu stratených paketov, je jej veľkou výhodou rýchlosť. Zariadenia však budú pripojené v rovnakej sieti, takže možno predpokladať, že by rozdiely v rýchlosti medzi protokolmi neboli moc postrehnutelné. Aj keď občasná strata informácie v tomto prípade taktiež nie je až tak veľký problém, pretože pohybové senzory v mobile sú pomerne citlivé a produkujú dáta vo veľmi malých časových intervaloch, v rámci spojenia by bolo dobré vedieť či je zariadenie stále pripojené alebo už došlo k jeho odpojeniu. Túto informáciu pri

UDP, vzhľadom na to ako funguje, nie je možné priamo určiť a preto som sa rozhodol, že lepšou voľbou bude použitie TCP.

Komunikácia bude riešená architektúrou klient – server, kde aplikácia v HoloLens bude predstavovať server čakajúci na spojenie na určitom porte, zatiaľ čo mobilná aplikácia bude klient naväzujúci spojenie na danom porte.

Kapitola 5

Implementácia

Kapitola sa zaoberá implementáciou obidvoch aplikácií navrhnutých v predchádzajúcej časti dokumentu. Implementácia mobilnej aplikácie je rozobraná v časti 5.1. Sekcia 5.2 rieši vytvorenie serveru v HoloLens a využívanie informácií zaslaných mobilom pri pohybe helikoptéry. Vytváranie a umiestňovanie checkpointov v miestnosti pokrýva časť 5.3, navigáciu k hologramom a zobrazovanie informácií na displeji časť 5.4. V časti 5.5 je popísané riešenie efektov a celkovej podoby a možností aplikácie.

HoloLens aplikácia je implementovaná v Unity¹, čo je „cross-platform“ engine, ktorý sa využíva najmä pri tvorbe 2D a 3D hier. Využitie Unity je veľmi výhodné, pretože má v sebe zabudovanú podporu UWP (Universal Windows Platform) aplikácií, používaných v HoloLens. Vývoj pomocou Unity podporuje aj samotný Microsoft, ktorý uvoľnil *MixedRealityToolkit-Unity*² – zbierku skriptov a komponentov, ktoré majú zjednodušiť a urýchliť vývoj aplikácií pre Microsoft HoloLens, Windows Mixed Reality headsety a OpenVR. *Spatial mapping* v tejto práci je implementovaný za pomoci modulov z tohoto balíčka, presnejšie z jeho orezanej verzie.

Pre vývoj mobilnej aplikácie som zvolil prostredie Android Studio³, využívané pri tvorbe aplikácií na platformu Android. Android Studio poskytuje okrem iného aj vlastný emulátor, vďaka ktorému je možné aplikácie skúšať a debugovať priamo na počítači.

5.1 Rozbor android aplikácie

Pri implementácii možno android aplikáciu rozdeliť na tri časti. Časť *Manifests*, ktorá obsahuje súbor *AndroidManifest.xml*, popisujúci dôležité informácie o aplikácii pre Android build tools, Android operačný systém a Google Play. *Java*, ktorá obsahuje zdrojové kódy aplikácie oddelené menom balíku (package). A časť *res* obsahujúcu položky ako XML layouty, bitmapové obrázky a UI refazce využívané v aplikácii.

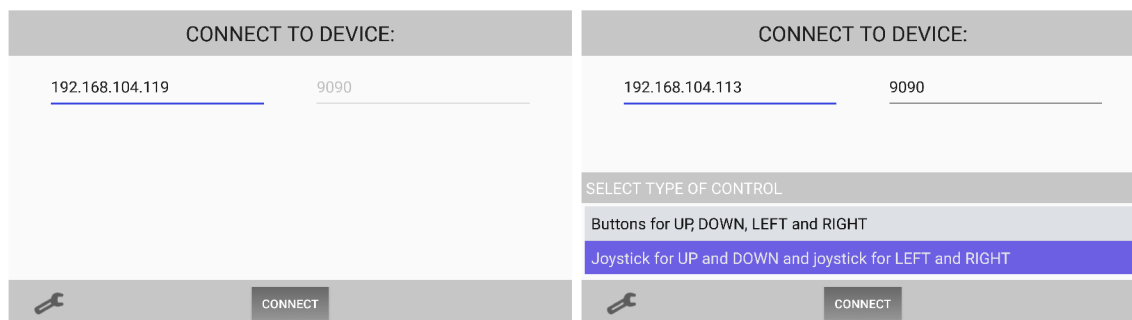
Aktivity

Základom android aplikácií sú aktivity. Ide o triedy, obsluhujúce to čo vidí užívateľ na obrazovke. Aplikácia v tejto práci pozostáva z troch aktivít, medzi ktorými užívateľ prepína na základe toho ako chce aplikáciu využívať. Prvá aktivita, s ktorou príde do kontaktu po zapnutí aplikácie je *MainActivity* (obr. 5.1). Tu je možné zadať IP adresu HoloLens,

¹<https://unity.com/>

²<https://github.com/Microsoft/MixedRealityToolkit-Unity>

³<https://developer.android.com/studio>



Obr. 5.1: Výsledná podoba úvodnej obrazovky mobilnej aplikácie slúžiacej pre pripojenie k HoloLens. Vpravo aj so zobrazenými možnosťami nastavenia ovládania.

prípadne zmeniť port, na ktorom sa bude mobil pripájať. Predvolený port je 9090, ktorý sa nachádza mimo rozsahu registrovaných portov a ktorý využíva aj aplikácia v HoloLens. Možnosť voľby je tu najmä pre pripojenie a testovanie priamo v Unity. V rámci hlavnej aktivity si užívateľ môže vybrať aký typ ovládania preferuje – 4 tlačidlá pre hore, dole, doľava a doprava alebo 2 joysticky pokrývajúce rovnaké pohyby. Každé z týchto dvoch ovládaní má vlastnú aktivitu – *ControlActivity* a *ControlActivity2* (obr. 5.2), ktorú je možné spustiť len z hlavnej obrazovky.

Rozloženie prvkov a výsledný výzor aktivít je definovaný pomocou XML súborov v priečinku *layout* spadajúceho do časti *res. activity_main.xml* definuje podobu hlavnej aktivity a *activity_interface(1-2).xml* je pre príslušné aktivity s ovládaním. Po zadaní cieľovej adresy do príslušného textového poľa, je možné pripojiť sa k želanému zariadeniu. Funkcia *connectToDevice()* predá Intent⁴ (spustí) aktivite s ovládaním, ktorá bola užívateľom vybraná. Ovládanie je možné vybrať z *list_view*, ktorý sa zobrazuje/schováva po stlačení tlačidla s obrázkom kľúča. IP adresa spolu s portom sa posúva novo spustenej aktivite a až z nej dôjde k vytvoreniu spojenia.

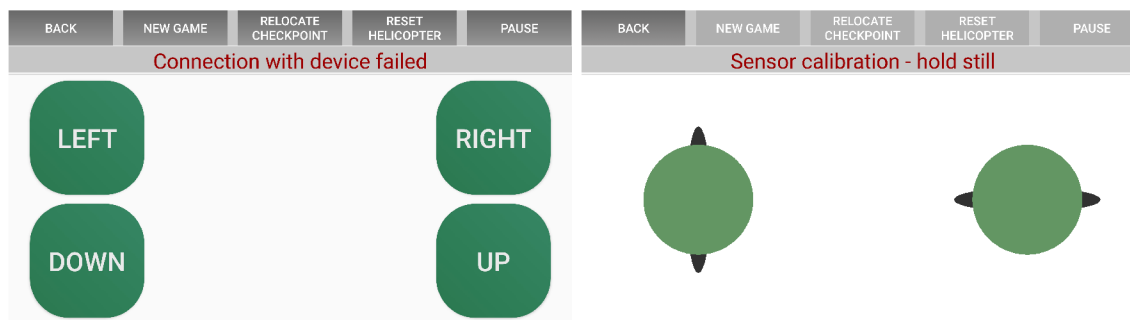
Ovládanie a komunikácia

O spojenie s HoloLens sa stará trieda *Connection*. K jej inicializácii dochádza v ovládacích aktivitách, pri ich štarte v metóde *onCreate()*. Po vytvorení spojenia pomocou sieťových soketov je možné posielat dáta HoloLens pomocou funkcie *SendData()*. Posielanie musí byť vykonávané na inom ako GUI vlákne. To je zaručené pomocou java rozhrania *Runnable*. Každé zavolanie *SendData()* je vykonané na novom vlákne, ktoré sa ukončí akonáhle je údaj poslaný. Na jeho ukončenie hlavné vlákno čaká, aby pri rušení komunikácie nedochádzalo k prípadom, že bude soket uzatvorený skôr ako sa odošle reťazec *<SC>*, informujúci HoloLens o ukočení spojenia. Pokiaľ sa spojenie nepodarí vytvoriť, vďaka tomu, že je použitý protokol TCP, je možné zachytiť výnimky a o neúspechu informovať užívateľa.

K volaniu funkcie *SendData()* dochádza vždy pri zmene stavu tlačidla alebo joysticku, alebo pri výskyte sensorového eventu zachyteného funkciou *onSensorChanged()*. Tá vyfiltruje len eventy spojené so sensorom *GAME_ROTATION_VECTOR*, ktorý sleduje rotáciu mobilu v priestore. Dáta zo senzora sú spracované prostredníctvom maticových výpočtov a následne posielané ako *string* v tvare:

rotácia okolo x | okolo y | okolo z | Tlačidlo dole | Hore | Vľavo | Vpravo

⁴<https://developer.android.com/training/basics/firstapp/starting-activity>



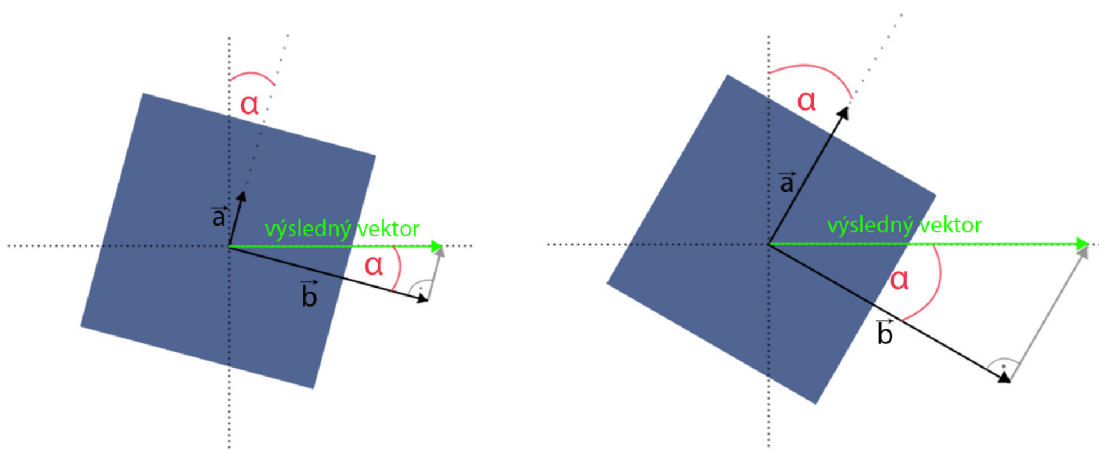
Obr. 5.2: Výsledná podoba ovládacích prvkov mobilnej aplikácie. *Vľavo*: Ovládanie pomocou tlačidiel. *Vpravo*: Ovládanie pomocou joystickov, zachytené pri kalibrácii senzorov – tlačidlá v hornej lište sú blokované.

Keďže využívanie senzorov je pomerne náročné na energiu, dochádza k ich registrácii až v momente keď je daná aktivita aktívna. Tým, že sa senzory zapínajú, je prvá várka dát značne nepresná. Preto sa dáta začnú posielat až po určitom čase (prvých 100 senzorových eventov sa ignoruje) a na displeji svieti text informujúci o ich kalibrácii (obr. 5.2 – vpravo). Po ustálení sa uloží aktuálna rotácia ako štartovacia a dáta sa začnú posielat smerom k HoloLens. Tlačidlá pre novú hru, reštart, atď., posielajú dáta kedykoľvek sú stlačené. Zasielaný refazec má tvar kľúčového slova ohraničeného $< >$.

Pre ovládacie tlačidlá bolo potrebné vytvoriť vlastnú triedu (*CustomButton*), aby bolo možné uchovávať ich stav a navrhnuť im vlastný dizajn. Joysticky sú instanciované takisto z vlastnej triedy (*JoystickView*), ktorá dedí z triedy *SurfaceView*. Tá poskytuje povrch, na ktorý je možné joystick nakresliť – kruhovú hlavu a elipsovú základňu. K prekresľovaniu hlavy dochádza vždy pri dotyku prstu s displejom alebo zmene polohy dotyku – event handler *onTouch()* volá funkciu *drawJoystick()* s atribútmi *newX* a *newY*, predstavujúcimi nový stred. Ten sa berie ako miesto kde k dotyku došlo, alebo ako maximálna možná vzdialenosť od základne. Vďaka tomu podporuje joystick narozdiel od tlačidiel až tri stavy (tlačidlá len dva – stlačené, nestlačené). Stavy joysticku, presnejšie jeho hlavy sú – nevychýlené, vychýlené a vychýlené veľmi. Pri ovládaní pomocou joystickov sa tak môže helikoptéra otáčať a hýbať pomalšie alebo rýchlejšie na základe toho, v ktorej pozícii joystick je.

5.2 Komunikácia a pohyb hologramu

Implementácia HoloLens aplikácie začína s vytvorením TCP servera, schopného nadviazať spojenie s mobilom. Základom programu v Unity sú objekty umiestnené v hierarchii scény a skripty pripnuté k týmto objektom. Skript *TCPServer*, zaobstarávajúci komunikáciu s telefónom, sa nachádza na prázdnom objekte *GameControl*. Pre vytvorenie spojenia v HoloLens boli využité sokety z knižnice *Windows.Networking.Sockets*, ktoré však v Unity nie sú podporované. Preto je potrebné daný kód podmieniť príkazom *#if !UNITY_EDITOR*. Aby bolo možné, aplikáciu spúšťať a testovať aj priamo v Unity, keďže nahrávanie do HoloLens je pomerne zdĺhavý proces, je v skripte vložený aj kód podmienený príkazom *#if UNITY_EDITOR*, implementujúci server pomocou soketov z knižnice *System.Net*. Na základe toho kde je aplikácia spustená následne dochádza k interpretácii jednej alebo druhej časti. Dôvodom využitia *Windows.Networking.Sockets* bolo, že sú schopné jednoduchšie



Obr. 5.3: Závislosť veľkosti výsledného vektoru pohybu (rýchlosti) od uhlu rotácie objektu. Vektor je výsledkom vektorového súčtu vektorov \vec{a} a \vec{b} . Veľkosť vektoru \vec{b} smerujúceho vpred (v súradnicovom systéme objektu) je priamo úmerná uhlu α . Veľkosť vektoru \vec{a} smerujúceho hore (v súradnicovom systéme objektu) je počítaná ako $|\vec{b}| * \tan \alpha$. Tento výpočet zaručuje, že výsledný vektor je vodorovný – helikoptéra sa pri náklone hýbe horizontálne, v rámci súradnicového systému sveta.

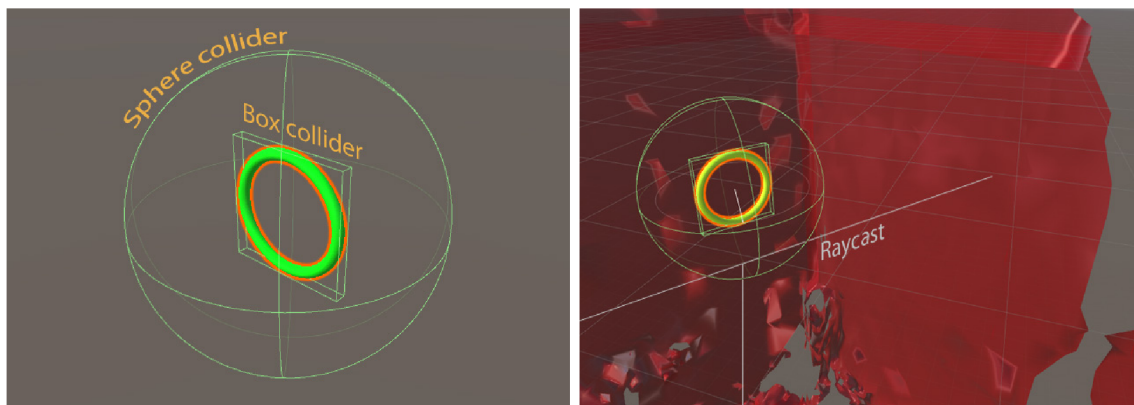
detekovať zrušenie soketu na opačnom konci. Aplikácia tak s pomocou *Exception* dokáže reagovať aj na prípad, keď je mobilná aplikácia ukončená „natvrdo“ bez zaslania $\langle SC \rangle$.

Po prijatí spojenia sa v cykle spracúvajú dáta zaslané z telefónu. V prípade, že prijatý reťazec predstavuje niektoré z kľúčových slov, dôjde k vykonaniu odpovedajúcich akcií (viac v čast 5.5). Pokiaľ ide o informácie o rotácii, dáta sú poslané objektu *Helicopter*, presnejšie jeho skriptu *MoveControl*, kde sú následne interpretované.

Údaje, ktoré poskytol server si *MoveControl* uloží a pracuje s nimi do momentu, kým neprídu novšie. Rotácie sa počítajú a vykonávajú každý záber v rámci *Update()* a to nastavovaním *transform.rotation* objektu. Dáta z telefónu sa objektu nepriradzujú priamo ale pomocou funkcie *Quaternion.Slerp()*, ktorá interpoluje hodnoty medzi aktuálnou rotáciou helikoptéry a rotáciou zadanou mobilom, čím sa výsledný pohyb zjemňuje a stáva viac plynulejší. Ako bolo spomínané v časti 4.3, pre rotovanie sú využité len údaje o rotácii telefónu okolo osí x a y . Unity pre pohyb objektu vpred využíva os z , čo znamená, že rotácia okolo osi x bude nakláňať helikoptéru dopredu a dozadu a rotácia okolo osi z do strán. Pred priradením súradníc je potrebné si ešte uvedomiť, že Unity používa left-handed súradnicový systém, kdežto mobil right-handed. Preto vo výsledku rotácie mobilu okolo osi y slúžia pre náklon okolo osi x a rotácie okolo osi x slúžia pre náklon okolo osi z . Údaje poskytované telefónom sú pred použitím orezané na náklon najviac 70 stupňov do všetkých smerov, aby otáčanie vyzeralo čo najreálnejšie a nestalo sa, že bude helikoptéra letieť dole hlavou.

Pohyb je vykonávaný s pomocou fyziky, ktorú poskytuje engine Unity, a to vďaka komponentu *Rigidbody*⁵, priradenému k objektu. To akým smerom sa bude objekt hýbať závisí od jeho náklonu. V rámci *FixedUpdate()* sa kontroluje uhol rotácie helikoptéry okolo osí x (pre pohyb dopredu a dozadu) a z (pre pohyb doľava a doprava). Helikoptéra sa následne udáva do pohybu aplikovaním síl – *Rigidbody.AddForce()*. Tie však pôsobia v súradnicovom

⁵<https://docs.unity3d.com/ScriptReference/Rigidbody.html>



Obr. 5.4: Vľavo: *Sphere* a *box collider* checkpointu slúžiace pre detekciu kolízií s helikoptérou a priestorovým modelom. Väčší polomer sphere collideru zaručuje, že sa checkpointy umiestňujú až v určitej vzdialenosti od stien a objektov. Vpravo: *Raycast* doprava, doľava a nadol, ktorým checkpoint kontroluje, či sa zobrazil vo vnútri snímanej miestnosti (všetky tri raycasty musia zasiahnuť povrch priestorového modelu). Začiatok raycastu sa nachádza mimo priestoru, ktorý pokrýva sphere collider, aby naň raycast nereagoval.

systéme objektu, ktorý je pôsobením rotácii naklonený pod určitým uhlom. Aby sa objekt hýbal vodorovne v rámci súradnicového systému sveta, je potrebné prepočítavať akou silou majú pôsobiť jednotlivé vektory na steny objektu tak, aby mal výsledný vektor správny smer a veľkosť (obr. 5.3). Sila pôsobiaca na bočnú stranu objektu, je priamo úmerná veľkosti uhlu náklonu. Od nej sa s pomocou tangensu tohoto uhlu dopočíta sila pôsobiaca smerom hore. Výsledný vektor, ktorý reprezentuje smer a rýchlosť pohybu helikoptéry je výsledkom vektorového súčtu týchto dvoch vektorov. Otáčanie vľavo alebo vpravo okolo osi y a stúpanie alebo klesanie helikoptéry nastáva pri stlačení príslušných tlačidiel na mobile (v zasielanom reťazci reprezentované ako 0 – nestlačené, 1 – stlačené, 2 – joystick vychýlený viac) a vykonávané je pomocou *Rigidbody.AddTorque()* respektíve *Rigidbody.AddForce()*.

5.3 Generovanie checkpointov

Pri generovaní checkpointov je potrebné zaistiť, že sa nebudú vytvárať v objektoch reálneho sveta alebo mimo užívateľovho dosahu. Informácie o prostredí získava a spracúva objekt *SpatialMapping* prebratý z orezanej verzie⁶ *MixedRealityToolkit-Unity*. Pomocou neho si aplikácia vytvára model prostredia, s ktorým je následne možné pracovať.

Správu checkpointov má na starosti skript *CheckpointGenerator*, umiestnený na objekte *GameControl*. Skript sa v prvom rade stará o inicializáciu checkpointov pomocou funkcie *InitCheckpoints()*. K jej volaniu dochádza len pri prvom pripojení telefónu k HoloLens. V rámci inicializácie sa do scény umiestnia tri objekty v tvare prstenca, predstavujúce checkpointy, ktoré sa budú využívať po zvyšok behu aplikácie. Každý checkpoint obsahuje *box collider*, ktorý približne opisuje jeho tvar a je využívaný na detekciu kolízií s helikoptérou. Pre zachytávanie kolízií s priestorovým modelom je checkpointu priradený prázdny objekt obsahujúci *sphere collider* (obr. 5.4). Vďaka nemu, kedykoľvek sa checkpoint vygeneruje v stene alebo inom reálnom objekte, dôjde k vyvolaniu eventu kolízie, ktorý v skripte *SphereColliderScript* zachytí metóda *OnTriggerStay()*. Tá následne *CheckpointGenerator*

⁶<https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-230>

informuje, že je potrebné checkpoint premiestniť. Collidre sú rozdelené medzi dva objekty, z dôvodu, že keby sa obidva nachádzali len na checkpointe nebolo by možné určiť, ktorý collider kolíziu vyvolal a tým pádom zaručiť, že *sphere collider* nebude reagovať na helikoptéru.

K zmene pozície dochádza pomocou funkcie *RelocateObject()*. Aby checkpointy zostávali vo vnútri nasnímanej miestnosti, tak sa z nich po ich premiestnení vyšle *Raycast*⁷ smerom doľava, doprava a dole (obr. 5.4). Ak *Raycast* v čo i len jednom z týchto troch smerov nezasiahne objekt priestorového modelu, znamená to, že bol vygenerovaný mimo miestnosť a je potrebné znovu zmeniť jeho polohu. Takto sa objekty presúvajú až dokým nenájdu pozíciu, v ktorej nedochádza k vyššie zmieneným konfliktom. Maximálny počet premiestnení na základe neúspešného raycastu je nastavený na sto, aby nedochádzalo k zacykleniu aplikácie. Ak ani na stý pokus miesto nevyhovuje kritériam, objekt sa už ďalej nepremiestňuje. Pri dobre nasnímanej miestnosti dochádza k maximálne piatim až šiestim neúspešným pokusom.

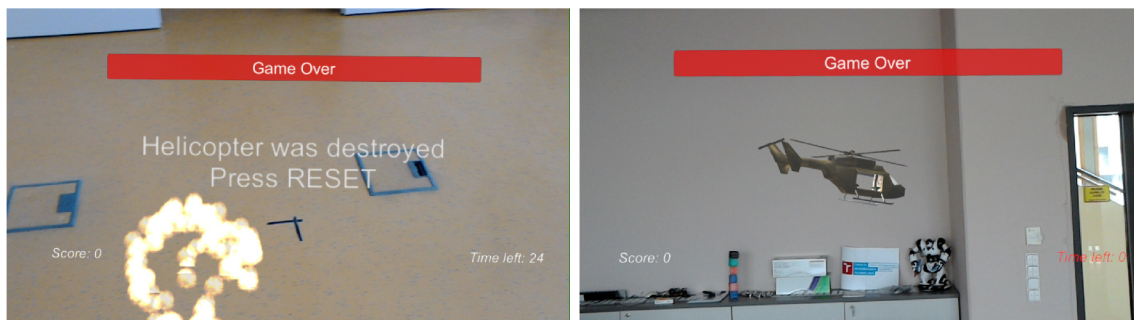
Pri určovaní rozsahov, v ktorých sa majú súradnice pre checkpointy generovať sa naskytovali dva spôsoby. Počítať ich v závislosti od toho, kde sa užívateľ nachádza – umiestňovať checkpointy len do určitej vzdialenosti od kamery. Alebo určiť rozsahy podľa hraníc nasnímaného priestorového modelu – umiestňovať checkpointy kdekoľvek do miestnosti. Model však v aplikácii nie je reprezentovaný ako jeden objekt ale zoznam objektov, ktoré vytvoril *SpatialMappingManager* pri snímaní a modelovaní okolitého prostredia. Preto je pri využití tohoto spôsobu nevyhnutné v určitom momente (pri pripojení mobilu k HoloLens) zastaviť *SpatialMappingObserver* a iterovať cez tieto objekty, za účelom nájsť hranice výsledného priestorového *meshu*. Tým sa eliminuje určitá dynamickosť prostredia a možnosť napríklad navigovať helikoptéru z jednej miestnosti do druhej. Preto, aj keď bol implementovaný aj tento spôsob⁸, sa vo výslednej aplikácii určujú hranice, v ktorých sa súradnice generujú, na základe polohy kamery (5 metrov do všetkých štyroch smerov a 2 metre hore alebo dole). K ich aktualizácií dochádza, s každým zavolaním *Update()*. Okolie je tak možné snímať počas celého behu aplikácie.

5.4 Navigácia

Navigáciu tvoria šípky zobrazované po okrajoch displeja a smerujúce k objektom, ktoré sa nachádzajú mimo zorného poľa HoloLens (obr. 5.6). O zobrazovanie, otáčanie a umiestňovanie šípok na obrazovke sa stará skript *TargetIndicator*, pripnutý k objektom, ktoré majú byť sledované – helikoptéra a aktívny checkpoint. Pri štarte *TargetIndicator* v rámci Canvasu vytvorí objekt šípky, s ktorým bude počas svojho života pracovať. Pozícia a rotácia šípky sa upravuje každý snímok v rámci *Update()*, volaním *UpdateTargetIconPosition()*. Funkcia najskôr pomocou *Camera.WorldToViewportPoint()* transformuje pozíciu sledovaného objektu (*transform.position*) z priestoru sveta do priestoru, ktorý je normalizovaný a relatívny ku kamere. Ak sa objekt nachádza za kamerou (vektor má záporné *z*) tak sa súradnice *x* a *y* preklopia a následne vo funkcii *Vector3Maximize* nastaví na 1 alebo posunú smerom k limitom. Toto je potrebné k tomu aby sa šípky zobrazovali po krajoch displeja, nie v strede, v prípadoch, keď sa objekt nachádza priamo za užívateľom. Posledná vykonaná transformá-

⁷<https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>

⁸Zdrojové kódy obsahujú funkciu *SetCoordinatesRange()*, ktorá iteruje cez objekty reprezentujúce reálny svet (*SurfaceObjects*) a na základe, ich hraníc (*bound.extends*), vypočíta v akom rozsahu možno súradnice generovať. Jej volanie, spolu s pozastavením *SpatialMappingObserver*, sa však v skripte *TCPServer* nevyužíva.



Obr. 5.5: *Vľavo*: Explózia helikoptéry spolu s informačným textom. *Vpravo*: Koniec hry spôsobený vypršaním času



Obr. 5.6: Výsledná podoba aplikácie. Zobrazovanie helikoptéry a checkpointov. Vpravo aj s navigáciou k checkpointu, ktorý je „offscreen“.

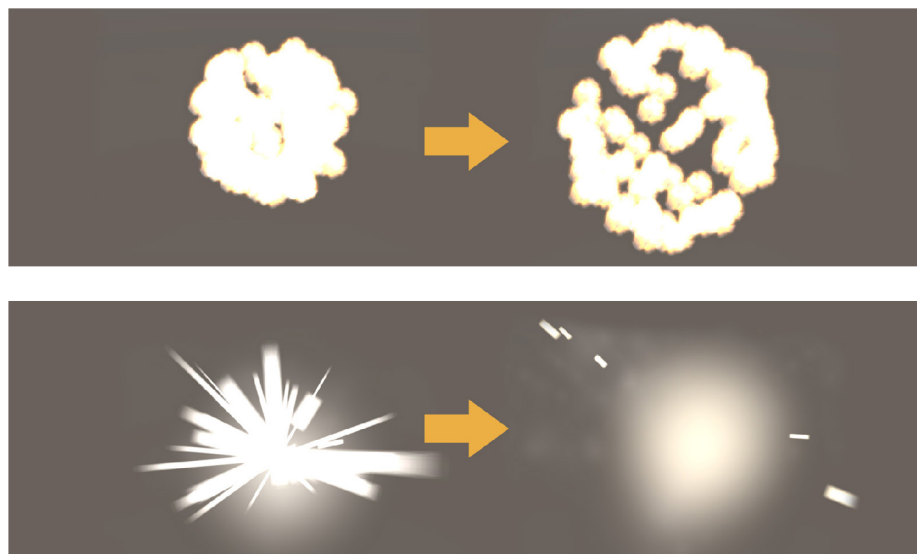
cia je do súradnicového systému obrazovky – pomocou *Camera.ViewportToScreenPoint()*. Výsledok sa oreže tak aby súradnice zodpovedali rozmerom displeja. Ak sa sledovaný objekt nachádza v zornom poli HoloLens šípka sa zneviditeľní nastavením *localScale* na 0. V opačnom prípade je mierka normálna. Navigácia sa schováva aj pri konci hry alebo keď dôjde k zničeniu helikoptéry. V týchto prípadoch sa volá funkcia *HideArrow()*.

Okrem navigačných šípok sa počas hry zobrazujú aj informácie o aktuálnom skóre a čase, ktorý užívateľovi ostáva do konca hry. Tie sú zobrazované po celý čas a nachádzajú sa v dolných rohoch displeja. V strede displeja sa zobrazujú informačné texty a to v prípadoch keď mobil nie je pripojený k HoloLens, došlo k zničeniu helikoptéry alebo bola hra pozastavená (obr. 5.5). O zobrazovanie týchto textov sa stará *TCPServer*, nastavovaním textového objektu *InfoTextUI*.

Všetky elementy užívateľského rozhrania v aplikácii spadajú pod *Canvas*. Ten nerenдерuje hlavná kamera ale *IU Camera* a zaručuje tak, že sa UI elementy zobrazujú pred všetkými ostatnými objektami v scéne.

5.5 Celková funkcionálnosť

Výsledná aplikácia má podobu hry, v ktorej je užívateľovou úlohou nazbierať čo najväčší počet checkpointov pred vypršaním časomier. Počiatočný čas je nastavený na 90 sekúnd s tým, že každý prevzatý checkpoint pridá užívateľovi 10 sekúnd. V momente vypršania časomier v skripte *CheckpointGenerator* funkcia *GameOver()* schová všetky checkpointy a zobrazí informačný text. K volaniu *GameOver()* dochádza aj pri kolízii helikoptéry s pro-



Obr. 5.7: Efekty použité pri vizualizácii explózie vzniknutej pri náraze helikoptéry. *Hore*: Efekt ohnivých častíc vytvorený pomocou *Particle System*, viditeľný v neskoršej fáze explózie. *Dole*: Svetelný efekt a dym, vytvorený objektom zo *Standard Assets*, viditeľný na začiatku explózie.

stredím. Tá sa detekuje pomocou *OnCollisionEnter()* v skripte *MoveControl()*. Zavolaním *objectCrashed()* sa helikoptéra zastaví a deaktivuje (*gameObject.SetActive(false)*), takže ju nie je možné vidieť a nereaguje na žiadne objekty ani na informácie posielané telefónom. K zničeniu helikoptéry nedochádza vždy ale až pri náraze určitou rýchlosťou.

Ak je helikoptéra zničená tak dôjde k explózií. Tá sa skladá z dvoch objektov, ktoré sú vytvorené a umiestnené na miesto kde k zrážke došlo. Objekt *Explosion* je implementovaný ako jednoduchý *Particle System* so skriptom *DestroyMe()*, ktorý ho po určitom čase zničí aby nezaberal priestor. Druhým objektom je *ExplosionMobile* prevzatý zo *Standard Assets*⁹, dopĺňajúci výsledný vzhľad explózie hlavne v jej začiatkoch (obr. 5.7).

O riadenie behu aplikácie sa v zásade stará *TCPServer*, ktorý na základe kľúčových slov prijatých z telefónu vykonáva potrebné akcie:

- *<SC>* – mobil ruší spojenie. Uzavrie sa soket, zobrazí sa text informujúci o prerušení spojenia, zastaví sa odpočítavanie času a pohyb helikoptéry zavolaním funkcie *StopMotion()* v skripte *MoveControl*. Zastavenie nie je okamžité ale postupné, keďže dôjde len k vynulovaniu premenných na základe, ktorých sa rotácia a pohyb helikoptéry počítajú.
- *<NewGame>* – príkaz k spusteniu novej hry. Zresetuje sa časovač a zavolá sa funkcia *NewTrack()* v skripte *CheckpointGenerator*. Tá vygeneruje nové pozície pre checkpointy. V prípade, že nastal koniec hry tak ich pred tým ešte aktivuje.
- *<Pause>* – pozastavenie hry. Helikoptéra sa zastaví pomocou *StopMotion()*, čas sa neodpočítava a aplikácia reaguje len na príkazy *<SC>* a *<Resume>*.
- *<Resume>* – spustenie časovača a odstránenie textu „Pause“.

⁹<https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-32351>

- *<Reset>* – zresetovanie pozície helikoptéry. Helikoptéra sa umiestni meter pred užívateľa a okrem zavolania *StopMotion()* sa aj priamo vynuluje *Rigidbody.velocity*, aby sa helikoptéra v momente zobrazenia nehýbala. V prípade, že bola zničená sa objekt najskôr aktivuje.
- *<RelocateCheckpoint>* – premiestnenie aktívneho checkpointu volaním funkcie *RelocateCheckpoint()*

Pre lepší zážitok sú v aplikácii využité aj zvukové efekty. Zvukovú zložku obsahuje helikoptéra¹⁰ aj checkpointy¹¹. Zvukový efekt checkpointu sa prehráva vo funkcii *OnTriggerEnter()*, ktorá je vyvolaná v momente, keď *box collider* detekuje kolíziu s helikoptérou. Zvuk sa prehráva aj pri explózií helikoptéry¹² a konci hry¹³. Model helikoptéry bol získaný z Unity Asset Store¹⁴ a rotácie jeho vrtúľ vykonáva jednoduchý skript pomocou *transform.Rotate()*.

¹⁰Zdroj zvuku: <https://www.youtube.com/watch?v=TAKEkdPb6Ag>

¹¹Zdroj zvuku: https://www.youtube.com/watch?v=KqgP0wGBa_Q

¹²Zdroj zvuku: https://www.youtube.com/watch?v=Ruwoj_qzCLA

¹³Zdroj zvuku: <https://www.youtube.com/watch?v=J93fc--VsaI&t=61s>

¹⁴<https://assetstore.unity.com/packages/3d/vehicles/air/personnel-transport-helicopter-13448>

Kapitola 6

Testovanie

Kapitola sa zaoberá testovaním výsledného produktu tejto práce. Testovanie bolo vykonávané s cieľom získať spätnú väzbu, ohodnotiť funkčnosť a nájsť časti, v ktorých by aplikácia mohla byť vylepšená. V časti 6.1 sú popísané testy a celkový priebeh testovania, výsledky sú zobrazené v 6.2. V 6.3 sú navrhnuté možné rozšírenia a zmeny.

6.1 Popis a priebeh testov

Funkčnosť aplikácie bola kontrolovaná a testovaná v priebehu vývoja, zakaždým keď bola pridaná nová funkčnosť. V konečnom dôsledku tak došlo k oprave rôznych sémantických a logických nedostatkov už počas vývoja a vďaka tomu možno povedať, že výsledná aplikácia spĺňa požiadavky predstavené v návrhu. V rámci záverečného testovania sa dôraz kladie skôr na celkovú použiteľnosť výslednej aplikácie a teda či funkčnosť, ktorá bola navrhnutá, bola navrhnutá dobre. Vo výsledku by tak testovanie malo odhaliť nedostatky v rámci zložitosti (aké zložité je aplikáciu pochopiť a následne ovládať – vzťahuje sa najmä na mobilnú aplikáciu a ovládanie helikoptéry) a užitočnosti práce (či výsledná funkčnosť pokrýva potreby užívateľa).

Na testovaní sa podieľalo osem účastníkov vo vekovom rozmedzí 21 až 23 rokov. Jeden z účastníkov mal v minulosti možnosť pracovať s Microsoft HoloLens, zvyšných sedem sa pred testovaním so zariadením nikdy nestretlo. Priebeh testovania bol nasledovný:

1. Najprv prišiel krátky úvod do problematiky – čo sú HoloLens, čo sa bude testovať, ako sa to bude testovať a čo je cieľom testovania.
2. Účastníci si nasadili HoloLens s už zapnutou aplikáciou a oboznámili sa s jej prostredím. Následne sa oboznámili aj s mobilnou aplikáciou a bolo im vysvetlené ako ovládanie funguje a čo je cieľom hry.
3. Testujúci strávili 10 až 15 minút hraním. Pričom tento čas bol rozdelený medzi ovládanie helikoptéry pomocou tlačidiel a pomocou joystickov.
4. Po dohraní došlo k vyplneniu dotazníkov a krátkej debata rozoberajúcej stránky aplikácie, ktoré sa užívateľom páčili a naopak nepáči, prípadne či im vo výsledku niečo chýbalo.

Pre vyhodnotenie celkovej kvality implementovanej aplikácie bol využitý User Experience Questionnaire (UEQ) [22]. UEQ meria okrem klasických aspektov použiteľnosti (efek-

tívnosť, prehľadnosť a spoľahlivosť) aj aspekty užívateľovho zážitku (originalita, stimulácia). Vďaka nemu je tak možné vyhodnotiť komplexný dojem užívateľa a jeho skúsenosti s aplikáciou. Okrem UEQ účastníci vyplnili aj dotazník vytvorený mnou, zameriavajúci sa na reakcie helikoptéry a jej ovládanie pomocou pohybov telefónu a taktiež na ovládanie pomocou tlačidiel a joystickov.

6.2 Výsledky

V tabuľke 6.1 je spracovaný vlastný dotazník spolu s odpoveďami od všetkých účastníkov testovania (označený písmenami A až H). Dotazník sa sústreďoval najmä na ovládanie helikoptéry pomocou senzorov a rozhrania na displeji. Účastníci pri odpovediach na otázky vyberali z dvoch opačných vlastností (podobne ako pri UEQ, viac nižšie). Pričom medzi možnosťami vyberali na stupnici od 1 (negatívne) po 6 (pozitívne).

Otázka	A	B	C	D	E	F	G	H	Priemer
Ovládanie mi prišlo: Intuitívne (6) – Neintuitívne (1)	6	5	5	4	5	5	4	5	4,875
Ovládať helikoptéru bolo: Veľmi jednoduché (6) – Veľmi zložité (1)	4	3	4	3	5	6	5	4	4,250
Ovládanie pomocou nakláňania telefónu mi prišlo: Zábavné (6) – Frustrujúce (1)	6	4	3	5	5	5	2	5	4,375
Helikoptéra robila to čo som od nej očakával(a): Stále (6) – Nikdy (1)	5	6	6	4	4	4	3	5	4,625
Lepšie sa mi hologram ovládal pomocou: Joystickov (2) – Tlačidiel (1)	1	2	1	1	2	2	2	1	1,500

Tabuľka 6.1: Tabuľka otázok z vytvoreného dotazníka a odpovede na ne od všetkých účastníkov testovania (A – H). Odpoveď sa vyberá z dvoch vlastností na stupnici od 1 (negatívne) po 6 (pozitívne), alebo 1 a 2 v prípade poslednej otázky.

Podľa otázok 1 a 4 možno usúdiť, že ovládanie pomocou mobilu užívatelia považovali za vcelku intuitívne a helikoptéra a jej pohyby väčšinou odpovedali tomu čo by užívatelia od nej v daný moment očakávali. Z odpovedí na otázky 2 a 3 vyplýva, že niektorým užívateľom prišlo ovládanie helikoptéry obtiažnejšie a miestami frustrujúce. To potvrdili aj pri záverečnej debate. Účastníci B, C a D sa však vyjadrili, že pri dlhšom hraní a používaní by si na ovládanie pravdepodobne zvykli a považovali by ho za menej obtiažne.

Piata otázka sa zameriavala na to, ktoré ovládanie účastníci preferovali. Tu sa testujúci rozdelili presne na polovicu, kde štyria z ôsmich účastníkov vybrali ovládanie pomocou joystickov a štyria pomocou tlačidiel. V rámci návrhu a implementácie bolo spomínané, že joystick má tri polohy a helikoptéra sa vďaka tomu môže pohybovať dvomi rôznymi rýchlosťami. Počas testovania sa skúmala aj táto vlastnosť, keď sa pri úplne vychýlenom joysticku helikoptéra otáčala rýchlejšie ako pri stlačení tlačidla. To malo za následok, že ovládanie pomocou joystickov pôsobilo citlivejšie. Testujúcemu B takáto citlivosť vyhovovala viacej, kdežto testujúcim D, F a H nie. Testujúci D sa vyjadril, že mu ovládanie pomocou joystickov prišlo lepšie, avšak vadila mu príliš veľká citlivosť a preto v dotazníku zvolil tlačidlá. Testujúci E by preferoval rozloženie tlačidiel ovládajúcich rovnaký typ pohybu v stĺpci nie v riadku.

V prípade užívateľského rozhrania v HoloLens sa všetci testujúci vyjadrili, že im prišlo zrozumiteľné a čitateľné a že lokalizovanie hologramov pomocou navigačných šípok bolo rýchle a veľmi pohodlné.

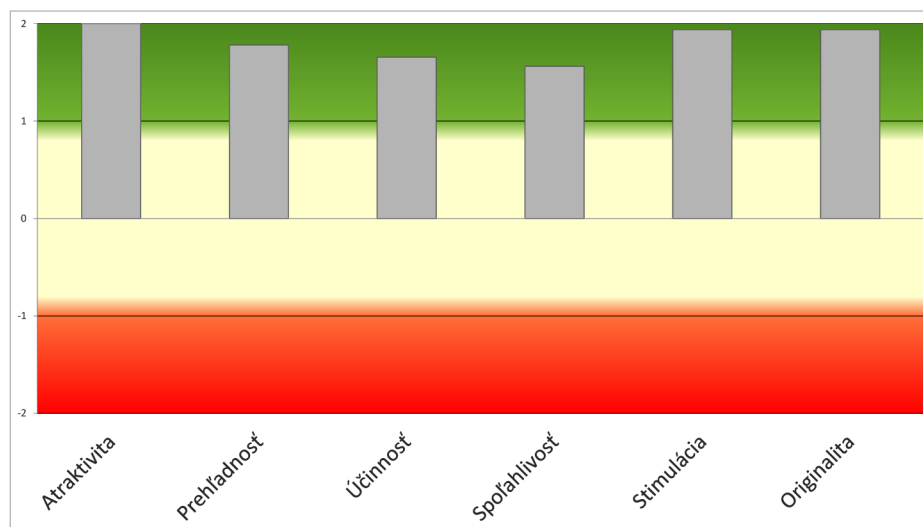
UEQ

Okrem vlastného dotazníku vyplnili účastníci testovania aj dotazník UEQ. Ten tvorí 26 dvojíc opačných vlastností, popisujúcich aplikáciu. Testujúci mali možnosť vybrať zo siedmich možností, pre každú dvojicu, ktoré predstavujú odstupňovanie medzi jednotlivými vlastnosťami. Príklad:

atraktívny ○ ○ ○ ○ ○ ○ ○ neatraktívny

Na základe odpovedí sa vytvoril graf (obr. 6.1) hodnotiaci aplikáciu v šiestich kategóriach:

- *Atraktivita* – celkový dojem z produktu. Páči sa alebo nepáči užívateľovi?
- *Prehľadnosť* – aké ľahké je zoznámiť sa s výrobkom a aké ľahké je naučiť sa ho používať?
- *Účinnosť* – dokážu užívatelia vykonávať a riešiť svoje úlohy bez zbytočného úsilia?
- *Spôľahlivosť* – má užívateľ pocit, že riadi interakciu?
- *Stimulácia* – je vzrušujúce a motivujúce produkt používať?
- *Originalita* – je produkt inovatívny a kreatívny? Zaujíma produkt užívateľov?

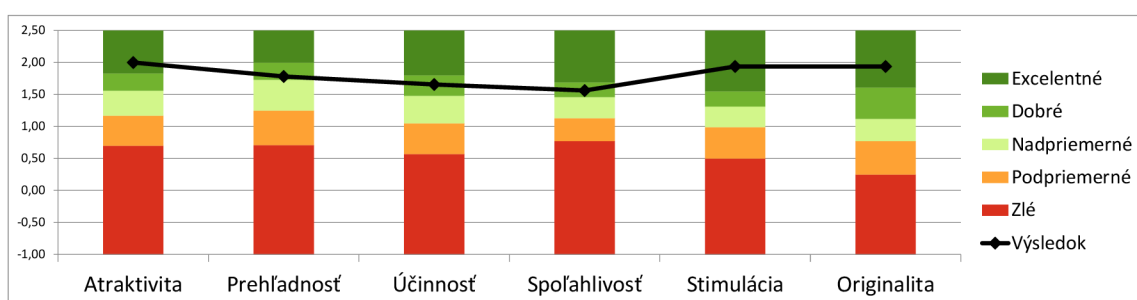


Obr. 6.1: Graf výsledkov testovania pomocou UEQ. Produkt je ohodnotený v šiestich kategóriach. Hodnoty medzi -0,8 a 0,8 sú považované za neutrálne, hodnoty väčšie ako 0,8 za pozitívne a hodnoty pod -0,8 za negatívne.

Z grafu možno vyčítať, že produkt má vo všetkých kategóriach pomerne pozitívne hodnotenia (hodnotenia nad 0,8). Najnižšie boli ohodnotené vlastnosti spojené so spoľahlivosťou a účinnosťou. To možno pripisovať tomu, že niektoré otázky dotazníku hodnotiace tieto

kategórie, si mohli testujúci spojiť s ovládaním helikoptéry, ktoré väčšina užívateľov považovala za zložitejšie. Keďže bolo testovanie vykonávané len na ôsmich účastníkoch a mohlo sa stať, že niektoré otázky nepochopili, je potrebné počítať aj s odchýlkou, tá má rozmedzie od 0,18 do 0,49 jednotiek oboma smermi. Pre spresnenie výsledkov a zmenšenie odchýlky by bol potrebný väčší počet testujúcich.

Okrem štandardného grafu UEQ poskytuje aj graf zobrazujúci *benchmark* (obr. 6.2) – namerané hodnoty sú zobrazené vo vzťahu k existujúcim hodnotám, získaným zo súboru referenčných údajov. Tento súbor obsahuje dáta od 18 483 osôb a 401 štúdií týkajúcich sa rôznych produktov (biznis software, webové stránky, webové obchody, sociálne siete). Porovnanie výsledkov produktu s údajmi v benchmarku umožňuje určiť relatívnu kvalitu produktu v porovnaní s inými produktami [22]. Z tabuľky 6.2, vytvorenej na základe benchmarku, možno vyčítať, že výsledný produkt, podľa vykonaných testov, patrí medzi 25% najlepších produktov, ktoré boli testované pomocou UEQ.



Obr. 6.2: Benchmark zobrazujúci namerané hodnoty produktu v porovnaní s hodnotami existujúcich produktov, ktoré boli testované pomocou UEQ.

Vlastnosť	Hodnota	Benchmark	Interpretácia
Atraktivita	2,00	Excelentné	Patrí medzi 10% najlepších výsledkov
Prehľadnosť	1,78	Dobré	10% výsledkov je lepších, 75% horších
Účinnosť	1,66	Dobré	10% výsledkov je lepších, 75% horších
Spoľahlivosť	1,56	Dobré	10% výsledkov je lepších, 75% horších
Stimulácia	1,94	Excelentné	Patrí medzi 10% najlepších výsledkov
Originalita	1,94	Excelentné	Patrí medzi 10% najlepších výsledkov

Tabuľka 6.2: Číselné zobrazenie hodnôt benchmarku a interpretácia výsledkov.

6.3 Možné vylepšenia a zmeny

Ako prvé z možných vylepšení do úvahy pripadá umožniť užívateľovi ovládať aj niečo iné ako helikoptéru. Ako bolo spomínané v návrhu, napríklad hologram lietadla, no pribudnúť by mohli aj menej zvyčajné objekty ako vzducholoď alebo lietajúci balón. Ovládanie týchto objektov by bolo značne odlišné a pre každý objekt unikátne čo by poskytovalo aplikácii nové možnosti.

Ďalšie z možných vylepšení je pozmeniť koncept hry, keď by užívateľ miesto zbierania checkpointov, napríklad ničil iné lietajúce objekty, ktoré by sa v prostredí náhodne generovali a následne by prenasledovali a pokúšali sa zničiť užívateľov hologram. Prípadne by sa

v prostredí mohli vytvárať rôzne objekty, reprezentujúce náklad, ktorý by musel užívateľ vyzdvihnúť a doručiť na iné miesto v miestnosti. Možných úprav výslednej podoby je veľa. Všetky tieto funkcionality by sa v aplikácii mohli nachádzať súčasne a ich výber by bol sprostredkovaný pomocou mobilu, kde by pribudlo menu s rozšírenými nastaveniami. Tu by si užívateľ mohol vybrať zo všetkých možností.

Keďže sa v rámci testovania objavili výhrady voči ovládaniu, mohli by určité vylepšenia a zmeny nastať aj v tomto smere. Napríklad implementovať nakláňanie hologramu takisto s využitím fyziky v Unity (čo by malo za následok iné reakcie pri nakláňaní mobilu) a umožňovať užívateľovi vyberať z týchto druhov ovládania ten, ktorý mu vyhovuje najviac. Za pokus by možno stálo aj porovnať ovládanie s využitím mobilu oproti ovládaniu herným ovladačom.

Kapitola 7

Záver

Cieľom tejto bakalárskej práce bolo realizovať ovládanie hologramu v Microsoft HoloLens s využitím pohybových senzorov mobilného telefónu. Za týmto účelom boli navrhnuté a implementované dve samostatné aplikácie – jedna pre mobil, druhá pre HoloLens. Začiatok práce je venovaný teoretickému úvodu do problematiky. Rozobratá je ako rozšírená realita, tak aj samotné HoloLens. Následne je popísaný návrh, implementácia a testovanie výsledného produktu.

Mobilná aplikácia bola vyvíjaná pre zariadenia so systémom Android. Mobil pre sledovanie svojej rotácie a následne ovládanie hologramu využíva virtuálny *game rotation vector* senzor, kombinujúci dáta z gyroskopu a akcelerometra. Pre pokrytie všetkých pohybov boli v aplikácii vytvorené rozhrania s tlačidlami alebo joystyckmi, ktoré dopĺňajú informácie o rotácii telefónu poskytované HoloLens. Mobil tak slúži ako bezdrôtový ovladač, nadväzujúci spojenie s headsetom prostredníctvom WiFi.

Aplikácia v HoloLens bola vyvíjaná v Unity, v ktorom je aj spustiteľná. Ako ovládaný objekt bola zvolená helikoptéra. Rovnako tak bol navrhnutý a implementovaný systém, na základe ktorého sa údaje z telefónu v HoloLens interpretujú a využívajú pre pohyby hologramu. V aplikácii boli využité aj funkcionality headsetu a to najmä *spatial mapping*. Vďaka nemu je možné vytvárať kontrolné body vo vnútri snímanej miestnosti a reagovať na kolízie s prostredím. Výsledkom je hra, v ktorej je užívateľovou úlohou zbierať vytvárané checkpointy v zadanom časovom intervale. Celú aplikáciu dopĺňajú aj zvukové efekty, prehlbujúce tak výsledný zážitok.

Výsledok bol testovaný ôsmimi užívateľmi a jeho kvalita vyhodnotená pomocou UEQ. Okrem všeobecného testovania produktu sa testovalo najmä ovládanie helikoptéry. V rámci testovaní sa zistilo, že ovládanie prišlo niektorým užívateľom zložitejšie a zaslúžilo by si tak menšie doladenia. Okrem zmien boli na konci testovania navrhnuté aj rôzne vylepšenia.

Literatúra

- [1] Affi-Sabet, K.; Walker, D.: *Microsoft HoloLens release date, rumours, specs & pricing: HoloLens rolled out for businesses with Dynamics 365 update*. [Online; navštívené 5.4.2019].
URL <https://www.itpro.co.uk/mobile/24780/microsoft-hololens-release-date-rumours-specs-pricing-microsoft-is-creating-ai-chips>
- [2] Android: Sensor types. [Online; navštívené 22.4.2019].
URL https://source.android.com/devices/sensors/sensor-types#game_rotation_vector
- [3] Android: Sensors Overview. [Online; navštívené 22.4.2019].
URL https://developer.android.com/guide/topics/sensors/sensors_overview
- [4] Azuma, R. T.: A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, ročník 6, č. 4, August 1997: s. 355–385, ISSN 1054-7460.
- [5] Bimber, O.; Raskar, R.: *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A K Peters, August 2005, ISBN 978-1568812304.
- [6] Carey, S.: How Microsoft HoloLens is being used in the real world. Máj 2018, [Online; navštívené 10.4.2019].
URL <https://www.computerworlduk.com/galleries/applications/how-microsoft-hololens-is-being-used-in-the-real-world-3648484/>
- [7] Carmigniani, J.; Furth, B.: Augmented Reality: An Overview. In *Handbook of Augmented Reality*, editácia B. Furth, Springer Publishing Company, Júl 2011, ISBN 978-1-4614-0063-9, s. 3–46.
- [8] Feltham, J.: Watch A Smartphone Turn Into A Controller For HoloLens. Marec 2017, [Online; navštívené 21.4.2019].
URL <https://uploadvr.com/watch-smartphone-turn-6dof-controller-hololens/>
- [9] Hanna, M.; Ahmed, I.; Nine, J.; aj.: Augmented Reality Technology Using Microsoft HoloLens in Anatomic Pathology. *Archives of Pathology & Laboratory Medicine*, ročník 142, Január 2018, ISSN 0003-9985.
- [10] Karthika, S.; Praveena, P.; GokilaMani, M.: HOLOLENS. *International Journal of Computer Science and Mobile Computing*, ročník 6, č. 2, Február 2017: s. 41–50, ISSN 2320-088X.

- [11] Krevelen, R. V.; Poelman, R.: A Survey of Augmented Reality Technologies, Applications and Limitations. *International Journal of Virtual Reality*, Jún 2010, ISSN 1081-1451.
- [12] Lang, B.: Devs Use ARKit to Make an iPhone Motion Controller for HoloLens. September 2017, [Online; navštívené 21.4.2019].
URL <https://www.roadtovr.com/devs-use-arkit-make-iphone-motion-controller-hololens/>
- [13] Microsoft: Gestures. [Online; navštívené 6.4.2019].
URL <https://docs.microsoft.com/sk-sk/windows/mixed-reality/gestures>
- [14] Microsoft: HoloLens (1st gen) hardware details. [Online; navštívené 5.4.2019].
URL <https://docs.microsoft.com/sk-sk/windows/mixed-reality/hololens-hardware-details>
- [15] Microsoft: Spatial mapping. [Online; navštívené 10.4.2019].
URL <https://docs.microsoft.com/en-us/windows/mixed-reality/spatial-mapping>
- [16] Microsoft: Spatial mapping design. [Online; navštívené 10.4.2019].
URL <https://docs.microsoft.com/en-us/windows/mixed-reality/spatial-mapping-design>
- [17] Microsoft: Voice Input. [Online; navštívené 6.4.2019].
URL <https://docs.microsoft.com/en-us/windows/mixed-reality/voice-input>
- [18] Microsoft: NASA: Microsoft HoloLens is helping us find the best sites for bases on Mars. Júl 2017, [Online; navštívené 10.4.2019].
URL <https://news.microsoft.com/en-gb/2017/07/10/nasa-microsoft-hololens-is-helping-us-find-the-best-sites-for-bases-on-mars/>
- [19] Milgram, P.; Kishino, F.: A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, ročník E77-D, č. 12, December 1994: s. 1321–1329, ISSN 0916-8532.
- [20] Peckham, J.: Hands on: Microsoft HoloLens 2 review. [Online; navštívené 5.4.2019].
URL <https://www.techradar.com/reviews/microsoft-hololens-2>
- [21] Schmalstieg, D.; Höllerer, T.: *Augmented Reality: Principles and Practice*. Addison-Wesley, 2016, ISBN 0-321-88357-8.
- [22] Schrepp, D. M.: User Experience Questionnaire Handbook. Február 2019, [Online; navštívené 9.5.2019].
URL <https://www.ueq-online.org/>
- [23] Wagner, D.; Schmalstieg, D.: ARToolkitPlus for Pose Tracking on Mobile Devices. In *Proceedings of 12th Computer Vision Winter Workshop*, Január 2007, ISBN 978-3-902465-60-3, s. 139–146.
- [24] White, S.; Feiner, S.: SiteLens: Situated Visualization Techniques for Urban Site Visits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, ACM, 2009, ISBN 978-1-60558-246-7, s. 1117–1120.

Príloha A

Obsah priloženého DVD

Na priloženom DVD sa nachádza:

- `xstano04.pdf` – Technická správa práce
- `tex` – Zdrojové súbory \LaTeX u
- `src` – Zdrojové súbory aplikácií
 - `HoloLensApp` – zdrojové súbory HoloLens aplikácie v Unity
 - `MobileApp` – zdrojové súbory mobilnej aplikácie pre AndroidStudio, spustiteľnú verziu aplikácie (APK) možno nájsť v `app/build/outputs/apk/debug`
- `video.mp4` – Demonštračné video práce
- `UEQ_Data_Analysis_Tool.xlsx` – Excel súbor s výpočtami testovacieho dotazníku
- `README.txt` – Text s inštrukciami ako nainštalovať a spustiť aplikácie

Príloha B

Dotazník

Toto je dotazník hodnotiaci ovládanie aplikácie. Prosím vyberte odpoveď ktorá najlepšie vyjadruje váš pocit o zadanej otázke. Mieru svojho súhlasu s odpoveďou vyjadríte tým, že označíte krúžok, ktorý je najbližšie vašemu názoru.

Ovládanie mi prišlo:		
Intuitívne	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Neintuitívne
Ovládať helikoptéru bolo:		
Veľmi jednoduché	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Veľmi zložité
Ovládanie pomocou nakláňania telefónu mi prišlo:		
Zábavné	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Frustrujúce
Helikoptéra robila to čo som od nej očakával(a):		
Stále	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Nikdy
Lepšie sa mi hologram ovládal pomocou:		
Joystickov	<input type="radio"/> <input type="radio"/>	Tlačidiel

Obr. B.1: Vlastný dotazník zameriavajúci sa na hodnotenie ovládania.