BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

CONTROLLING VIRTUAL AVATAR IN MICROSOFT HOLOLENS WITH USE OF REAL-WORLD ELEMENTS

OVLÁDÁNÍ VIRTUÁLNÍHO AVATARA V MICROSOFT HOLOLENS S VYUŽITÍM PRVKŮ REÁLNÉHO SVĚTA

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **KLAUDIA FAJTOVÁ**
AUTOR PRÁCE

SUPERVISOR **Ing. DANIEL BAMBUŠEK**
VEDOUCÍ PRÁCE

BRNO 2020

Brno University of Technology
Faculty of Information Technology

Department of Computer Graphics and Multimedia (DCGM) Academic year 2019/2020

Bachelor's Thesis Specification

Student: **Fajtová Klaudia**
Programme: Information Technology
Title: **Controlling Virtual Avatar in Microsoft HoloLens with Use of Real-World Elements**
Category: User Interfaces

Assignment:

1. Study the concept of augmented reality and its use in the entertainment industry. Get familiar with Microsoft HoloLens and game controller (Xbox, PS).
2. Connect the game controller to HoloLens, select appropriate methods and tools, and design a demonstration application that will allow to control a virtual character and will use real-world objects and elements.
3. Implement the application.
4. Conduct experiments and evaluate your solution.
5. Create a poster or video presenting key features of the solution.

Recommended literature:

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

Requirements for the first semester:

- Items 1, 2 and partially item 3.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Bambušek Daniel, Ing.**
Head of Department: Černocký Jan, doc. Dr. Ing.
Beginning of work: November 1, 2019
Submission deadline: May 28, 2020
Approval date: April 9, 2020

Abstract

Aim of this paper is to study the problem of surface scanning and mapping in augmented reality using the Microsoft HoloLens headset and to introduce the solution on how to dynamically generate virtual content that would be able to respond to dynamic environment changes. The goal is to collect the data about the real environment in order to gain the knowledge of its modifications, to process it and understand. To accomplish this goal, HoloLens native components for spatial awareness were used. The final application that demonstrates this functionality contains a virtual avatar that moves around the environment in order to collect randomly generated targets. The character is able to react on dynamic environment changes and is controlled by the gamepad, connected to the headset via Bluetooth.

Abstrakt

Cieľom tejto práce je štúdium skenovania a mapovania prostredia v súvislosti s rozšírenou realitou, s využitím zariadenia Microsoft HoloLens a následne vytvoriť riešenie ako dynamicky generovať virtuálny obsah, ktorý by bol schopný reagovať na dynamické zmeny prostredia. Cieľom je zber dát z reálneho prostredia pre získanie informácií o jeho zmenách, ich spracovanie a následné využitie. Za účelom dosiahnutia týchto cieľov boli použité komponenty Microsoft HoloLens zodpovedné za priestorové vnímanie. Výsledná aplikácia obsahuje virtuálneho avatara, ktorý sa pohybuje po priestore a zbiera náhodne vygenerované ciele. Avatar je schopný reagovať na zmeny prostredia a je ovládaný pomocou hráčskeho gamepadu, ktorý je s headsetom spojený pomocou technológie Bluetooth.

Keywords

Microsoft HoloLens, augmented reality, virtual reality, mixed reality, AR, VR, MR, Spatial Mapping, Spatial Understanding, head-mounted display, Unity

Klíčová slova

Microsoft HoloLens, rozšírená realita, virtuálna realita, zmiešaná realita, AR, VR, MR, Spatial Mapping, Spatial Understanding, okuliare pre zmiešanú realitu, Unity

Reference

FAJTOVÁ, Klaudia. *Controlling Virtual Avatar in Microsoft HoloLens with Use of Real-World Elements.* Brno, 2020. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Daniel Bambušek

Rozšířený abstrakt

Dnešná moderná doba nám ponúka možnosť zanechať tradičné 2D technológie a vstúpiť do sveta 3D priestoru. Jednou z možností ako interpretovať túto ponuku používateľovi je pomocou rozšírenej reality. Zmyslom rozšírenej reality je obohatiť typický reálny svet o virtuálne, počítačom vytvorené objekty a tak ponúknuť nový pohľad na užívateľove okolie. Pomocou tejto technológie sme schopní nahradiť doteraz používané monitory a klávesnice novým zaujímavým priestorom, ktorý ešte viac rozvinie ľudskú predstavivosť. Stále zväčšujúci sa záujem o rošírenú realitu napomohol k vytvoreniu viacerých zariadení, ktoré podporujú túto technológiu.

Firma Microsoft Corporation v roku 2016 predstavila unikátne zariadenie nazývané Microsoft HoloLens. Sú to okuliare pre zmiešanú realitu, ktoré užívateľovi ponúkajú jedinečný zážitok. Užívateľ je schopný hologramy nie len vidieť ale s nimi aj pohybovať a rozmiestňovať ich podľa svojich predstáv. Interakcia s virtuálnymi objektami prebieha pomocou špeciálnych jednoduchých gest alebo hlasových príkazov.

Vďaka dnešným mobilným zariadeniam sa dnes s rozšírenou realitou môže stretnúť ktokoľvek a teda sa rozšírená realita stala žiadúcou súčasťou našich životov. Aby bolo možné vytvoriť aplikáciu využivajúcu túto technológiu, cieľové zariadenie, pre ktoré je táto aplikácia naprogramovaná musí byť schopné priestorového vnímania. To je nevyhnutné aby aplikácia podporújúca rozšírenú realitu bola spustiteľná v akomkoľvek, vopred nepoznanom priestore. Tento proces skenovanie stále nie je dokonalý a ponúka veľa priestoru pre jeho zlepšenie.

Cieľom tejto práce je štúdium skenovania a mapovania prostredia v súvislosti s rozšírenou realitou na zariadení Microsoft HoloLens. Informácie získané na základe tohoto štúdia môžu byť následne použité pre statickú ako aj dynamickú prácu so skenovaným prostredím. Výsledky štúdia sú demonštrované na výslednej aplikácií v podobe dynamicky generovaných objektov, ktoré sú schopné reagovať na zmeny reálneho prostredia. Medzi tieto objekty patrí virtuálny avatar a virtuálne ciele, dynamicky rozmiestnené po rozlohe skenovaného prostredia. Užívateľ aplikácie ovláda avatara s cieľom pozbierať všetky rozmiestnené ciele. Aby sa tento zámer uskutočnil, užívateľ dynamicky vytvára avatarovi cestu k cieľom a avatar je schopný reagovať na takto uskutočnené zmeny v reálnom prostredí. Aplikácia aj avatar sú ovládané pomocou Xboxového herného ovládača, ktorý je s okuliarmi Microsoft HoloLens spojený pomocou technológie Bluetooth.

Teoretická časť práce ponúka čitateľovi bližšie priblíženie pojmu rozšírená realita ako aj jej základné rozdiely oproti virtuálnej realite. Opísané sú taktiež hlavné typy zariadení, ktoré rozšírenú realitu poskytujú. Keďže aplikácia je vytvorená pre okuliare Microsoft HoloLens, toto zariadenie je taktiež podrobnejšie opísané.

V praktickej časti tejto práce je navrhnutý vzhľad, ako aj funkcionalita aplikácie. Návrh analyzuje existujúce riešenia a ich nedostatky a prináša vlastné návrhy ako tieto nedostatky vyriešiť aby sa neobjavili vo finálnej implementovanej aplikácií. Aplikácia je implementovaná pomocou herného enginu Unity, ktorý využíva jazyk C# pre objektové skriptovanie. Počas implementácie boli použité niektoré existujúce balíčky a zdrojové súbory za účeľom urýchlenia implementácie a možnosti sústredenia sa na zadanú problematiku. Všetky použité zdroje sú riadne odcitované.

Priestorové vnímanie aplikácie je zabezpečené pomocou dvoch priestorových komponentov, ktoré sú súčasťou využitých okuliarov a ktorých úlohou je vytvorenie priestorovej siete. Komponenta s názvom Spatial Understanding je využitá za cieľom získania statických údajov o skenovanom prostredí. Tieto údaje sú následne využité pre rozmiestnenie virtuálnych objektov na konkrétne miesta v reálnom prostredí. Komponenta na základe ukončeného

skenovania ponúka zoznam miest, na ktoré je možné uložiť virtuálny objekt. Tieto pozície sú vopred podmienené na základe požiadavkov aplikácie. Druhou komponentou je Spatial Mapping, ktorej úlohou je opakovaná aktualizácia priestorovej siete za účeľom detekcie zmien vytvorených v reálnom prostredí. Kombináciou týchto komponent aplikácia dosahuje požadovanej funkcionality ale predstavuje nový problém. Komponenta Spatial Understanding nie je schopná dynamickej aktualizácie svojej vygenerovanej priestorovej siete a teda po presunutí reálneho objektu nie je schopná detekovať túto zmenu. Týmto sa v aplikácií vytvára duplicitný obraz jedného reálneho objektu na dvoch rôznych miestach. Problém je riešený úpravou zdrojového kódu komponenty Spatial Understanding aby bola schopná pravidelne aktualizovať svoju priestorovú sieť.

Finálnú aplikáciu testovalo šesť ľudí. Cieľom experimentov bolo zistiť ako moc je aplikácia intuitívna a ako rýchlo sa užívateľ adaptuje po jej prvom spustení. Účastníci experimentov aplikáciu hodnotili na základe dotazníku o užívateľovej skúsenosti s aplikáciou (UEQ). Výsledky dotazníku preukázali nadštandardnú originalitu aplikácie a jej vysokú atraktívnosť. Účinnosť aplikácie bola hodnotená nižšou hodnotou, čo sa pripisuje k stálym nedostatkom vo vytváraní priestorovej siete okolia. V porovnaní s inými 452 štúdiami, finálna aplikácia je hodnotená nadpriemerne vo všetkých skúmaných oblastiach. Účastníci experimentu taktiež poskytli spätnú väzbu, ktorá je v tejto práci spísaná a môže byť považovaná ako možné budúce vylepšenie aplikácie.

Controlling Virtual Avatar in Microsoft HoloLens with Use of Real-World Elements

Declaration

Hereby I declare that this bachelor's thesis was prepared as an original author's work under the supervision of Ing. Daniel Bambušek. All the relevant information sources, which were used during preparation of this thesis, are properly cited and included in the list of references.

.....
Klaudia Fajtová
May 28, 2020

Acknowledgements

I would like to give my thanks to the supervisor of my thesis, Ing. Daniel Bambušek for his guidance and motivating words. My thanks also belong to the Department of Computer Graphics and Multimedia for lending me all equipment needed for developing the final application. I can't forget to thank all participants who took part in the experiments and all people that motivated me during the development.

Contents

Chapter 1

Introduction

Today's modern technology allows us to leave the classical 2D environment behind and present its content in 3D. One of the possibilities to interpret such approach to the user is augmented reality. That is responsible for enhancing the real world with the computer generated images. Using this technology, we are able to substitute the monitors, keyboards and computer mice with the new, interesting environment and use our imagination even further. The approach of augmented reality helped with creation of multiple devices providing this technology.

The company Microsoft Corporation came with the unique device called Microsoft HoloLens. This mixed reality headset allows user not only to see the holograms but to interact with them as well. This is accomplished by special gestures or user's voice.

Aim of this paper is to study the problem of surface scanning and mapping and introduce the solution on how to dynamically generate virtual content which would be able to respond to dynamic environment changes. The goal is to collect the data about the real environment in order to gain the knowledge of its modifications, to process it and understand.

As the result of this thesis, the final project demonstrates the entertaining application within which the user controls the virtual character. In order to accomplish the goal, the character needs to collect all dynamically generated virtual targets with the use only of the real-world object. As the user relocates the objects, the environment changes. The virtual character must be able to respond to this changes and interact with them.

The interaction possibilities provided by the augmented reality headsets are not the typical input of the entertaining product. The use of gestures or voice in a noisy environment could not be as entertaining. This problem is resolved by the use of the Microsoft Xbox controller as the application's main input. The communication between the headset and the gamepad is established via Bluetooth.

The final application represents the entertaining side of the augmented reality and points out the future of entertainment industry using augmented reality head-mounted display. In the future, when these headsets are smaller and available for everybody as today's smartphones, this could be the way this industry would grow.

The following chapter 2 describes the augmented reality, its components and display technologies as the chapter 3 introduces mentioned headset Microsoft HoloLens more closely. Chapter 4 contains information about the analysis and the design of the application, following with the interesting implementation details in chapter 5. The processed test results are presented in chapter 6. The conclusion of this work content can be found in chapter 7.

Chapter 2

Augmented reality

The aim of this chapter is to illustrate the concept of the augmented reality (AR) for the reader to gain the picture of the subject. Following sections provide definition of augmented reality (2.1) and its difference against the virtual reality (2.2). To fully understand how the augmented reality achieves mixing virtual objects with the real world, main components of AR are described in its own section 2.4. The goal of the last segment of this chapter is to introduce the reader to several user interfaces available for AR (section 2.3).

2.1 Definition

The concept of augmented reality is presented under many definitions across the literature but has only one meaning in information technologies. It combines the real world with a computer-based animations to provide enhanced, or augmented, view of the reality [3]. Technically, it overlays digital information on an image of the real object viewed through a device [6].

According to the book of augmented reality [7], the most widely accepted definition was proposed by Azuma in 1997. He established that the AR must fulfill three following characteristics:

- Combine real and virtual
- Interactive in real time
- Register in 3D

This expanding technology is employed mostly in entertainment industry, medical training, design & modeling, tourism, classroom education or field service [17]. The aim of the AR is to enhance the real world with not only virtual objects but also virtual sounds, tactile perceptions or other sensory stimuli.

2.2 Mixed reality continuum

Depending on the amount of virtuality in the real world, the augmented reality is only one of the many realities in the field. In 1994, Paul Milgram [7] introduced concept of mixed reality continuum as following: ”merging of real and virtual worlds somewhere along the ’virtuality continuum’ which connects completely real environments to completely virtual

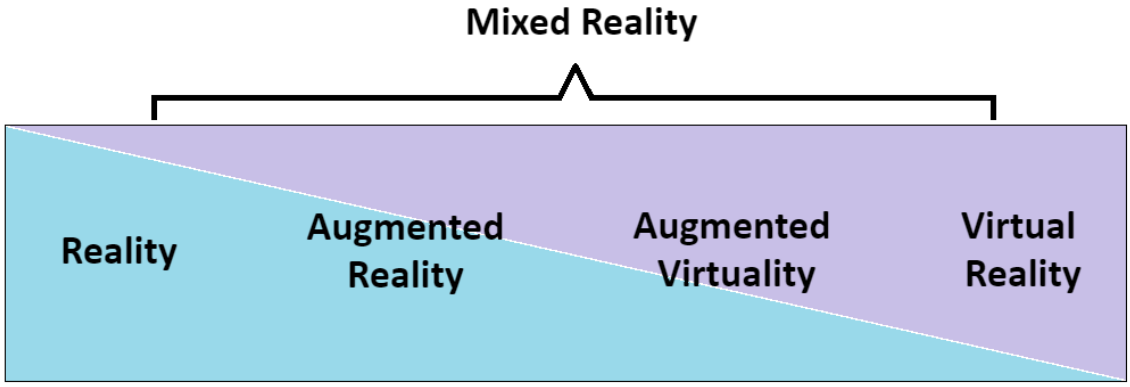

Figure 2.1: Mixed reality continuum. Each part contains different amount of virtual reality. One in the far left represents pure reality without any additional virtual effects. On the other side is placed pure virtual reality where the user has no contact with the real world.

ones". The mixed reality continuum groups all these worlds together. It is represented by continuous scale and defined by four terms as could be seen in Figure 2.1.

At the far left end of this continuum, the real environment occurs. This term represents the real world without any virtual objects used as we see it through eyes, the most familiar environment for human being.

The other side, on the other hand, is pure virtuality. Milgram defined virtual reality as a technology with the quality to fully immerse a user in a completely computer-generated environment using a head-mounted display (HMD) or headset [7]. Person with the headset on cannot see or hear any real-world signals and he becomes the part of his virtual reality. To manipulate objects or move around the environment, the user needs to use adapted haptic controllers. Every object in this end of continuum is virtual so there are no restrictions as to what a person can do or experience.

Between these opposites, phrases augmented reality and augmented virtuality are located [16]. In the context of the quantity of virtual objects in the real world, augmented reality uses virtual objects only to enhance the real environment. As the perfect example stands the well-known game of the 21. century – Pokemon GO[1]. This mobile application inserts virtual objects into the real world, leaving it to stand as the main environment.

In the case when the virtual world predominates over the reality, the phrase augmented virtuality is used. For instance, it can be represented by an RPG game in virtual environment using real faces of the players. The major part of what user experiences is virtual, only integrated with the elements from the real world.

2.3 Display technologies

Before resolving the problem of registration and visualisation of virtual objects onto the real surface, this section will provide introduction of different types of devices responsible for the process. As the sight is human's most widely used sense for perceiving the surrounding

[1] https://pokemongolive.com/en/

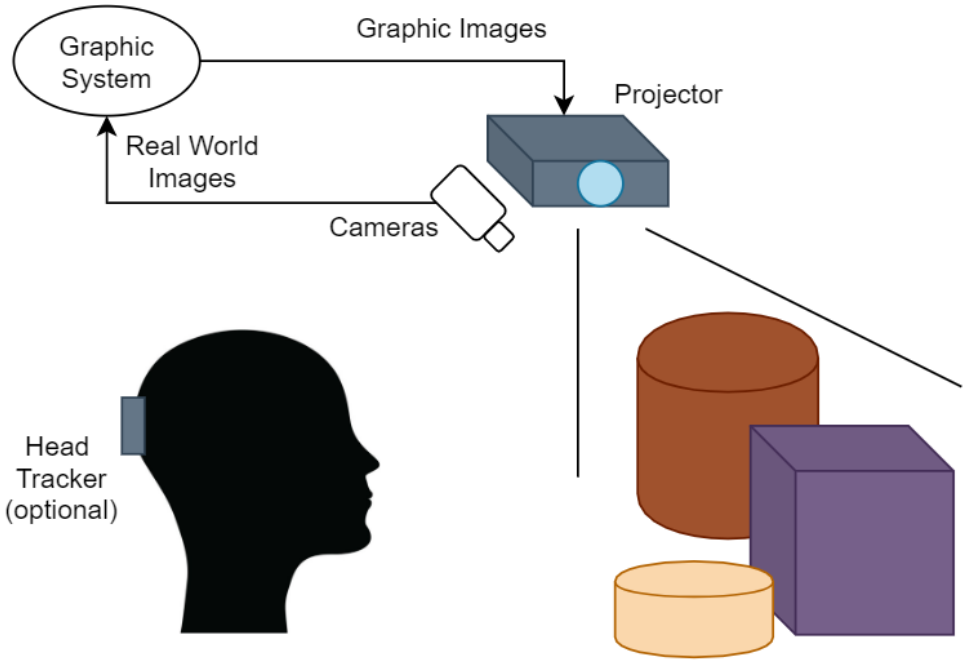

Figure 2.2: Projected Display conceptual diagram

environment, the focus will be set on AR display technologies. I will divide the devices according to user's experience.

Projected Displays

Visualising holograms using spatial projector is oftentimes called *spatial augmented reality* [4]. The virtual objects are projected directly onto the real surface. User needs no other support device to experience the AR. Holograms are visible for naked-eye. Therefore, there are no restrictions on how many users can see them. The holograms are not able to react to user's signals without any additional device collecting user's input such as motion detector or contact surface [1]. However there is space to improve quality of the hologram. Projector may highlight the hologram's details. For instance texture, surface details, shading or dynamic behaviour if in case of animated virtual object. There is possibility to use more than one projector to increase the quality. This type of technology is frequently used for educational or design-creating purposes. The conceptual diagram of this type of display can be found in Figure 2.2.

Handheld Displays

Nowadays, augmented reality can experience anyone using smartphones or tablets. These handheld displays are equipped with all the technology needed to create the abstract objects and visualise them by the help of the actual display. As this is the most accessible type of the AR, majority of augmented reality aimed applications come under field of entertainment [10]. Typical example of the application is already introduced Pokemon GO or today's favorite Minecraft Earth[2] where people together enhance the real world with their

[2] https://www.minecraft.net/en-us/about-earth

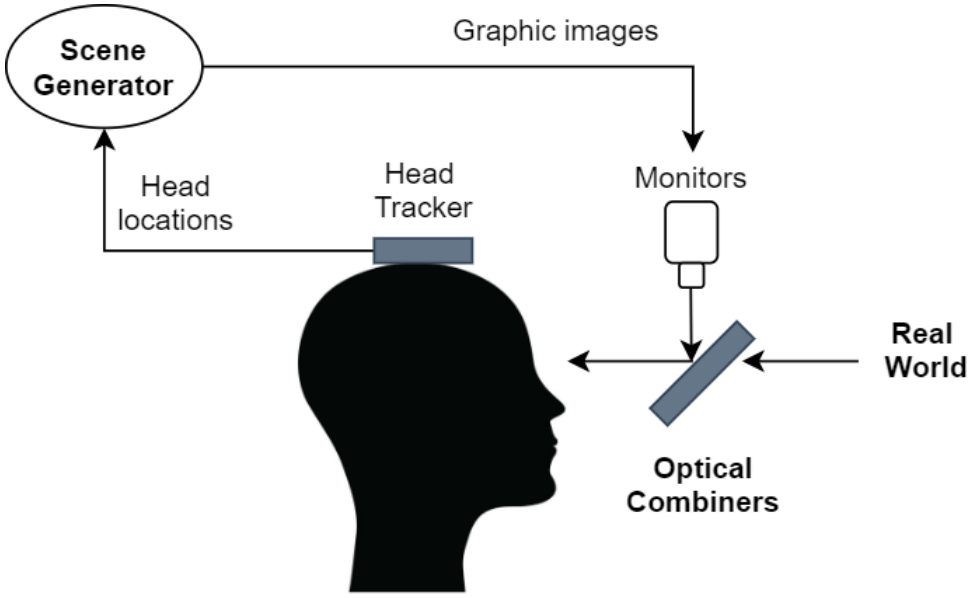

Figure 2.3: Optical see-through HMD conceptual diagram

own creations. To mention an application with another purpose than entertain, Amikasa[3] would help the user to style his room with virtual added furniture or wall paintings.

Head-mounted Displays

These display types have a main problem to solve and that is the weight of the device. As all weight is concentrated on the human's neck, device must be as light as possible while meeting all the technical requirements. When this condition is fulfilled, HMD offers an unique experience. Several devices allow user to fully interact with the virtual objects. Beginning with holding or scaling the hologram, up to moving it around the room. The whole viewed real world can promote to an widely enhanced environment. To imagine the head-mounted type of display, Meta 2 or Microsoft HoloLens propose to as an example.

Based on augmentation method, head-mounted displays differ to two following categories [2]:

The device providing **optical see-through display** must contain special component, known as optical combiner. This element takes the virtual object and links it to the view of the real world. Such device offers the full view of the real environment as the combiner is partially transparent, however the virtual part is only visible through the combiner. That is accomplished by making combiner partially reflective as well. That way the holograms mirror to human eye. Scheme of such device can be seen in Figure 2.3. To advance the quality of the future experience, device producers have to figure out how to redesign the display to cover as much human field of view as possible as now the combiners cover only small part.

Video see-through HMD replaces the vision of the user with one or more video cameras. Instead of the optical combiner as it was in optical see-through display, combining the real world with virtual objects is supplied by the component called video compositor.

[3] http://www.amikasa.com/

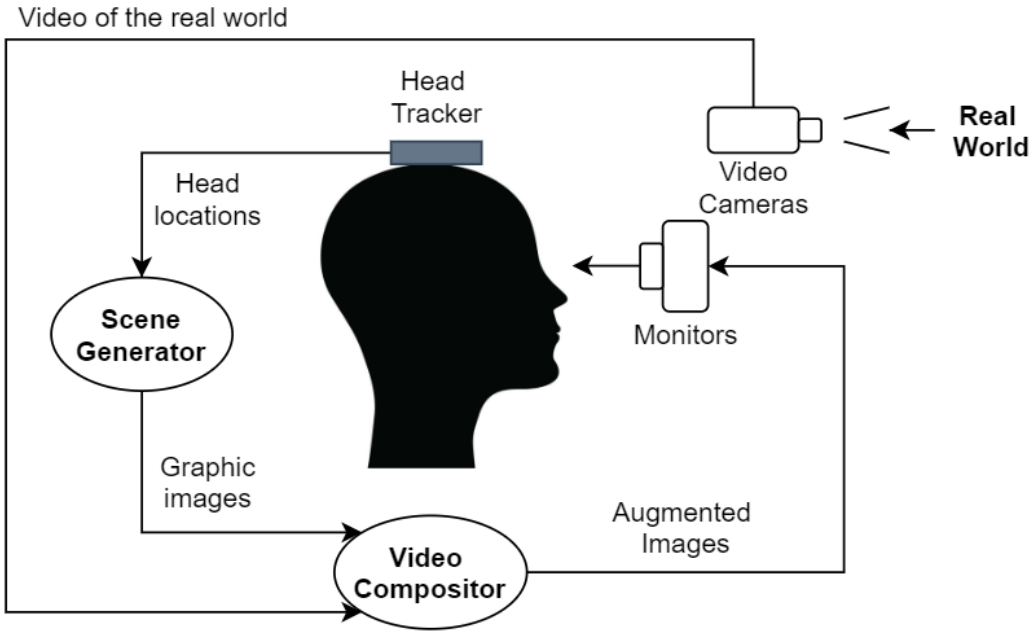

Figure 2.4: Video see-through HMD conceptual diagram

That links the video of user's view with the holograms provided by the scene generator. This device is similar to closed-view display, though user is able to see the environment thanks to the camera. In contrast to optical see-through HMD, virtual objects are all seen in the full field of view. When the device is turned off, the user can not see anything. The resolution of the display also presents the disadvantage as it degrades not only the holograms but the user's view of the real world as well. The process of creating the video by combining the images extends the calculation time by nanoseconds. The conceptual diagram of the video see-through HMD is visible in Figure 2.4.

2.4 Components of the AR

Augmented reality would have no use if the existence of enhanced virtual objects would not be believable. Assigning virtuality to the real world accurately is the typical goal of the AR. To achieve this goal, three overlapping methods are used (Figure 2.5) – *calibration*, *tracking* and *registration* [7].

Real environment is mapped with different coordinate systems as the today's technology. To synchronize these two systems, term called ***calibration*** is used. This process compares measurements of two different devices and is carried out only at discrete times. Some devices needs to be calibrated just once in a lifetime, other every time before commencing an operation. All depending on the measurement system. Given an optical see-trough display, process of calibration refers to displaying calibration pattern to the user, whom is asked afterwards to align a physical object, usually the forefinger of the user, with this calibration pattern. This operation is inseparable part for this type of display as the camera is always moved between places. Without this process, there would be no environment registration possible. After proper accomplished calibration, the device is ready to present object over-

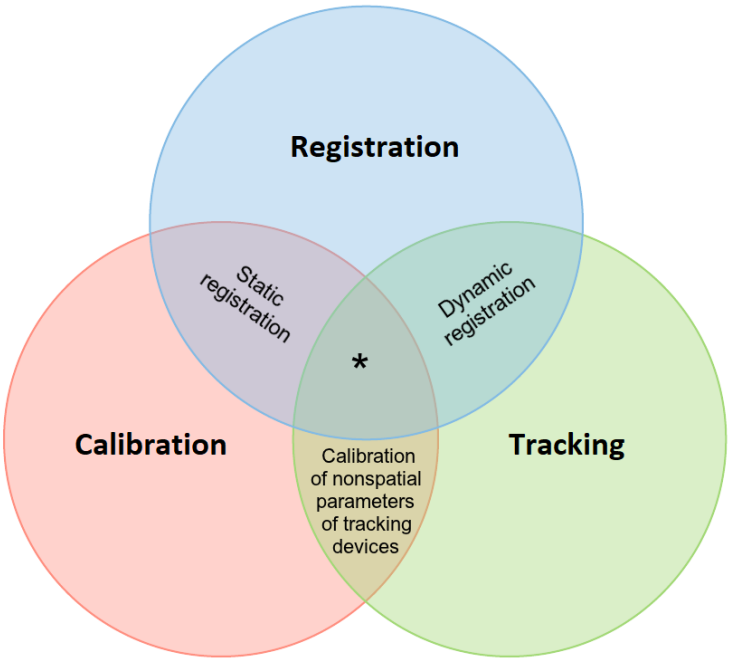

Figure 2.5: Three important overlapping concepts of AR system. The ' * ' stands for calibration of *spatial* parameters of tracking devices

lays on its see-through display.

Performing these measurements only once is not sufficient enough to recognize the environment satisfactorily. AR device needs to be aware of its location toward the real world all the time. The system is expected to react on any environment change as merrily as it is possible. Term ***tracking*** describes dynamic sensing and measuring of AR systems. Since AR operates in real time, tracking is responsible for updating pose measurements continuously. Pose measurements refer to not only the position but as well the orientation of the user. Typically the state of his head or limbs, relative to the real entities [19].
Various techniques of tracking are used, according to scanned location. We divide the location types as an outdoor and indoor location, considering the scale likewise. The best practice is to combine Global Positioning System, better known as GPS, for outdoor scanning with gyroscope for rotation tracking and accelerometers for translational motions. To accomplish a better indoor position accuracy, depth cameras or stereo cameras are used. These technologies use patterns placed on the target's surface, typically corners and edges of the object.

After all important coordinates are collected and handled, the process of ***registration*** begins. This term refers to the alignment of coordinates systems between virtual and real objects. For better understanding, it means to map virtual objects onto the tracked objects of the real world. Using the results of the calibration, AR device establishes the common coordinate system between virtual and real object. This is called *static registration*. When the camera or user is moving within the environment, aforementioned technique tracking is used to obtain *dynamic registration*.

Visualization

Spatial mapping is not the only thing for user to be able to experience augmented reality. The most important part is to visualize the objects for the user. Visualization determines how virtual information should be shown [8]. To achieve realistic feeling, non-real objects have to be depicted as real-world objects so cues of computer graphics must be respected.

One of the cues that are more complicated is the photometric cue, which main focus is to simulate the transport of light between real-world objects and the virtual objects. The most common representation of the photometric cue is displaying the shadows. To real-like shadow visualization, standard shadow simulation methods are applied. *Shadow mapping* or *shadow volumes* are typical example of these techniques.

Another worth-mention cue is the occlusion – how to render virtual scenes behind the real objects. To accomplish this feature, real scene geometry knowledge is important.

To finally satisfy user's experience, the device must resolve what part of the virtual object needs to be settled. White and Feiner entitled the process of computing the position of the virtual object onto the real surface as *situated visualization*. This process sets its usage in uncovering hidden scenes. Underground cables, gas pipes or just room behind the wall are all today's usage examples.

Chapter 3

Microsoft HoloLens

To experience augmented reality at its fullness, special device is needed. This thesis works with the headset called Microsoft HoloLens. It is the product of Microsoft Corporation company. Under development name Project Bamboo, this mixed reality smart-glasses are able to fully operate on its own. That means both computing and object visualisation are the responsibility of the headset alone. All information provided in this chapter are about the HoloLens of the first generation as the application was developed on this device. Both generations are illustrated in Figure 3.1.

Firstly, the chapter provides hardware details of the device (section 3.1), then analysis its spatial components (section 3.2) used for being aware of the environment. Following section introduces most commonly used inputs (3.3) for the device so it can be controlled by user. Section 3.4 illustrates headset's key advantages and disadvantages and finally the chapter provides general information about the second generation of the described device (section 3.5) and compares it to its future version.

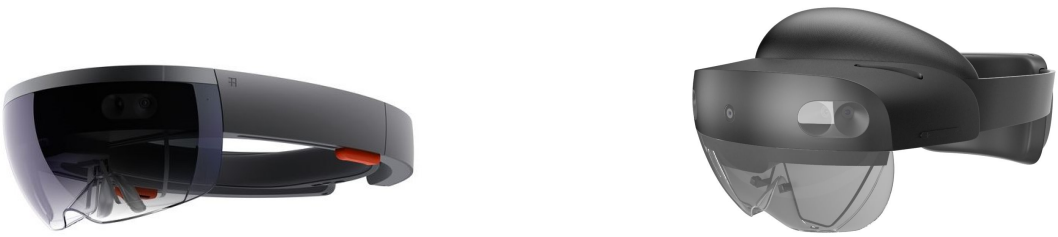

Figure 3.1: Microsoft HoloLens, first (left) and second (right) generation[1]

3.1 Hardware

Rendering the augmented reality scene is not as unit demanding as it could be at first glance [12]. Microsoft HoloLens comfortable operates on 2 GB RAM with the help of Microsoft Holographic Processing unit (HPU) of 1GB. This unit is responsible for handling

[1] https://www.dbpixelhouse.com/microsoft-hololens
[1] https://4experience.co/hololens-2-vs-hololens-1-whats-new/

the data provided by sensors and mapping the holograms into the real world. The smartglasses feature with Intel 32-bit CPU. To sample the environment, the device contents depth cameras with a 120°x120° angle of view and a photographic cameras of 2.4 megapixel. As any other smart device of today's world, HoloLens includes essential unit as accelerometer, gyroscope and magnetometer. Without these, the knowledge of the user direction and position would not be possible. The other product components are WiFi, Bluetooth, audio jack, speakers and head-mounted widescreen used to communicate with the user or other devices. The headset runs on Windows 10 operation system, therefore can be used as a common computer. To create a HoloLens aimed application, the developer is advised to use special open source development kit, known as HoloToolkit.

3.2 Spatial components

Without scanning, the knowledge about the real world would not be possible. The result of this procedure is represented by triangle mesh rendered over the real environment [11]. ***Spatial mapping*** [13] collects all the information about the surroundings and creates the mesh. This functionality detects objects away from the user in the range from 0.8 metres to 3.1 metres. This is to prevent scanning user's hands as objects. Created mesh is always updating in a time interval of three seconds and can not be saved and used other than in present time.

Spatial understanding [9] is, on the other hand, designed to be used by an application. It is applied so the headset recognizes mesh sections such as walls, floor or conditioned surfaces. The scan needs to be finished before its usage. Once the scan is done, the generated mesh is saved and can be used. There is no possibility to update the saved mesh. With this result the application is able to place virtual objects on favoured places. For instance, user is can virtually hang a picture on the wall, place a furniture on the floor or set a virtual character on the couch to create an illusion of sitting hologram. As there is no ability to update the mesh, the application can not fully update the position of moved objects. That means that even though the object is moved and registered in a different location, its original mesh stays at the same place. Mesh generated after spatial scanning is visible in Figure 3.2.

3.3 Interaction options

Any device without a possibility to interact would be pointless. As the headset is in a way the computer itself, it needs the user input to present the requested output. With none mouse or keyboard given, AR devices need to come up with the original way to connect user and the device. When speaking about Microsoft HoloLens, the glasses provide three main options for interaction.

Gestures

The first and most natural way is to operate the HoloLens with gestures. The human has need to touch the object in order to believe in its existence. Two gestures were designed for this headset. The first, called bloom for its similarity for blooming flower, has only one responsibility. To open or close the main menu. To select the application, the gesture called Tap is used. It is the movement that shortly connects the hand's forefinger with the first

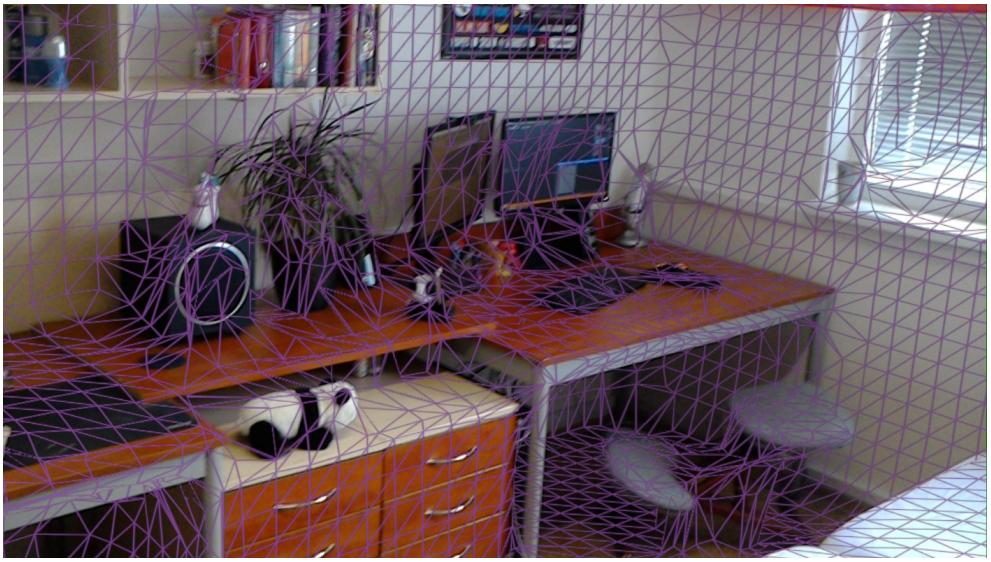

Figure 3.2: Generated mesh after spatial scanning

finger. When the user lets the fingers connected, after moving the hand to side, he is able to scroll, move or resize the application. Illustration of both described gestures is drawn nearer under Figure 3.3.

Voice

The headset features with build-in microphone so it can accept voice input [14]. Simple instructions are implemented such as Select or Close. The voice is also used when searching or writing. Instead of tapping all the characters one by one, user is able to dictate whole phrase. Windows 10 introduced Cortana assistant which you can manage also by voice. There is a possibility to adjust personal commands to not reserved words. For instance to create open-close menu shortcut by saying the word “menu„.

Gaze

Without being aware of interacting, the gaze is the most important input. It is represented by a small cursor adjusting to surface. No preceding input options would be possible without gaze. To select the application, user needs to look at its icon and then proceed with the gesture or voice command. The cursor indicates where the user is looking, however it only follows the head movement, not the eye movement.

Even though these are the main input options, the headset is able to be managed another way. Connecting the third-party device via Bluetooth or WiFi enables to switch these commands. The application created under this work uses Microsoft Xbox Controller connected via Bluetooth as the input option.

3.4 Primary strengths and limitations

Head-mounted headset Microsoft HoloLens represents one of the best today’s AR devices [15]. The feature that makes them to stand out the most is theirs precised spa-

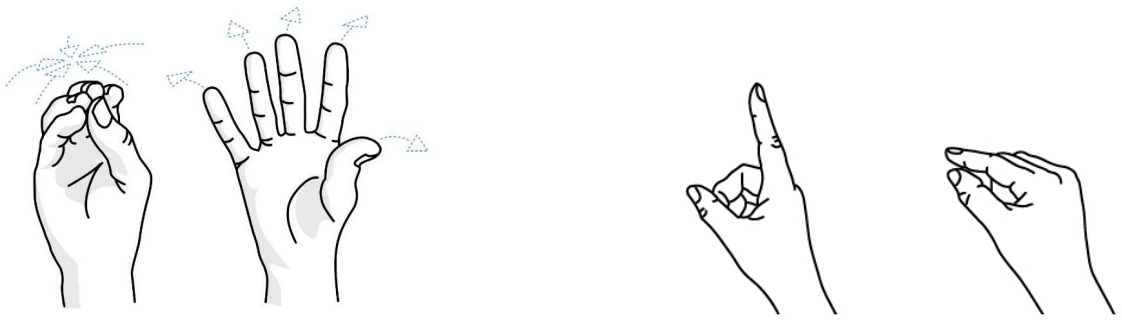

Figure 3.3: Two main input gestures. The second Tap option represents hold position to scroll, move or resize the object[2]

tial scanning. In combination with outstanding position tracking, the device is capable of placing virtual objects with nearly centimeter scale precision [5]. The AR user experience completes the real looking interaction between holograms and real objects. These holograms are able to react on the environment modification almost immediately thanks to always updating spatial mesh as explained in the section above. The user is allowed to interact with the holograms about as if they were part of the real world. This experience perfects developed feature called spatial sounds that provides the effect of distant audio. Utilising *head related transfer function*, the device generates different wave lengths depending on the direction and distance of the user to the hologram. The change of the wave creates the impression of the sound appearing from the side where the hologram is placed.

Apart from the strengths, the user experience can be downgraded by the headset's uncomfortableness. The device with all its components weights more than 550 grams and even though the focus stays primary on the head, the considerable amount of weight lays on the user's nose. Another cause of discomfort is the display alone. After hours of constant use the effect can evoke nausea. The last user experience disadvantage would be the limited field of view of only 30°x17.5°. Figure 3.4 shows the visible scene for better imagination. Apart from limitation that aggravate the perception of the holograms, the headset suffers from short lasting battery life. The device is able to operate only for maximum of three hours or last approximately two weeks on the standby mode. Fortunately, the headset is fully operable while charging.

3.5 Microsoft HoloLens 2

Since the Microsoft HoloLens of the first generation release in 2016, the constant developing modern world of technology has changed past recognition. As expected, the company collected all the feedback and three years later, in 2019, released the second generation of its mixed reality headset [20]. HoloLens 2 comes with many new features apart the hardware upgrade, such as the design of the headset that is now available to wear over the optical glasses. This and the expanded field of view (43°x29°) lead to more comfortable user experience as can be seen in Figure 3.5. The person is now able to interact with holograms in bigger scope. The device is amended to scan all ten fingers therefore apart from the new way how to push the button, user can use both hands to naturally move or resize the objects or to play the virtual keyboard. More precision and accuracy is embedded

Figure 3.4: The limited field of view[3]

into process of registration and visualisation. The headset is developed to scan eye lenses. That means the user does not need to move his head to navigate the cursor but the lens movement suffice.

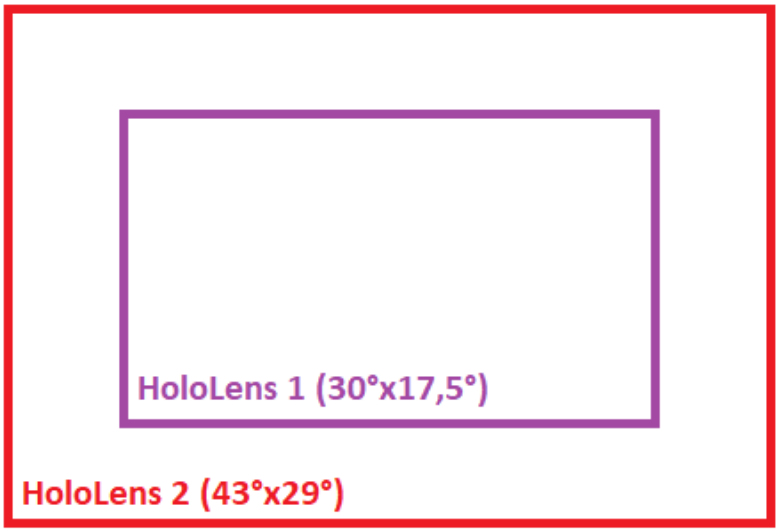

Figure 3.5: The field of view comparison between HoloLens both generations

Chapter 4

Analysis and design

When developing an application, the programmer has to know all the requirements and restrictions. My requirement is to demonstrate the usage of the real environment for a virtual character collecting virtual objects. Both worlds has to cooperate with each other. The character will be controlled via gamepad.

The final decision is to develop an entertaining application within which the virtual character collects dynamically generated targets. To reach the targets, character needs to climb on real objects. The user therefore has to relocate physical entities to create a path for the character to reach the targets. This character will be controlled via Xbox controller connected to headset via Bluetooth.

This chapter firstly provides existing solutions and their limitations in section 4.1. Secondly, it analyses main design problems in section 4.2 and in section 4.3 the chapter introduces decided design solution and the reasons for them to be implemented. Application backstory is described in section 4.4.

4.1 Existing solutions

When it comes to games on Microsoft HoloLens, the user is often found as the main character of the game. This way the application centers its attention on the user. In my case the user just controls the virtual character, therefore the application has to collect data about the third person primarily. The example of the existing solution is Young Conker[1] (Figure 4.1). The character in the appearance of the fox collects golden coins that are generated all over the room with additional virtual objects to help the character. The fox in this game is controlled with one gesture (tap) or voice commands. To make the character move, the user has to move his head all the time as the fox follows user's gaze. For instance in order to collect all generated coins, the user needs to provide many small but quick movements with his neck. For the character, it is possible to jump only at special generated places that invoke the jump animation. Another example is shooting the enemy that can also be performed only at such places. The only activity that Young Conker can perform all the time is walking around the room. From my personal experience, as I was trying this game, I didn't get as much entertainment experience from controlling the character using this type of navigation, as I usually get from games using a gamepad. Additionally, the level contains many small virtual objects so in order to see as much as it is possible, the user needs to stay farther away which makes smaller holograms not completely

[1] https://conker.fandom.com/wiki/Young_Conker

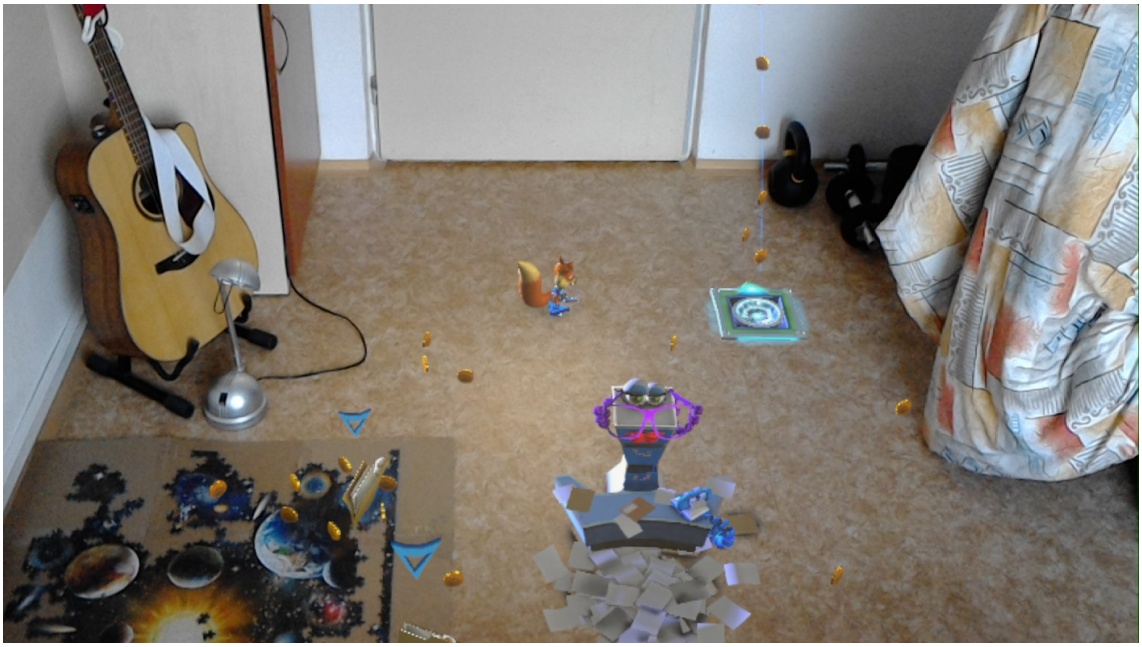

Figure 4.1: Snapshot of the game Young Conker, one of described existing solution. The game generates virtual objects all over the environment. Conker, or the fox, collects virtual coins and additional objects based on application storyline. The fox is controlled by user's gaze and by tap gesture. Young Conker can perform jump activity by moving to special generated location, such as the blue-silver square visible in the right side of the illustration. The game goal is to collect or objects connected to storyline and complete the missions. Collected coins only reflect user's game score.

recognizable. Most holograms are generated on the floor, close to each other. That and the fact that user navigates the fox by moving his head makes the experience a bit exhausting and can't be played for a longer amount of time.

Microsoft HoloLens offers the possibility to connect Xbox controller via Bluetooth, however, this serves only as the replacement of the classical input. The widest usage of the gamepad is while playing games inside the application window in 2D environment when the experience is similar to one behind the computer monitor. Xbox app serves as an example. At the moment, there is no 3D application where the controller is required. The games are managed with the user's gaze and gestures. Also, it is not possible to use gamepad as the alternative input for 3D augmented games. To make the application possible to take an input from the gamepad, the application source code needs to be adapted.

Most applications work with the pre-scanned room and then enhance the game environment with the virtual objects. Asking the user to firstly scan the room and then wait until the holograms are generated are the most common steps taken by such applications. However, after this chain, applications don't observe the room afterwards and don't register the changes made to real environment. Even described example application, Young Conker, scans the environment only when the application is launched and additional changes made to environment are not detected. In my case, the user needs to move the real objects to win the game therefore I need to scan the room all the time to unsure all environment modifications are detected.

4.2 Problem analysis

Before actually settling on particular design decision, developer should ask several questions about his future application and try to consider different design possibilities. The purpose of these questions is to meet with unconsidered problems and deal with their consequences. During my session of creating an application design I encountered with multiple questions worth asking. These problems are categorized in following subsections and correspondingly analysed.

Pre-scan of the surroundings

Previous chapter introduced two main spatial concepts that are responsible for mapping the surroundings. I need to decide which of these components would be the best option for my application. The first spatial component, spatial mapping, provides frequently updated mesh. This feature offers the application to be aware of the constantly modified environment. I could use it to observe changes made by user to help the virtual character reach the target. On the other hand, utilization of spatial understanding is advantageous in a way to save the scanned mesh for its processing. Such data could be put to use in order to find specified locations to arrange the targets around the room. However, once the generated mesh is saved, it won't update anymore and the modifications made by user to environment wouldn't be noticed. To satisfy given assignment, the only possible option is to use both spatial components as only combined together they provide all features needed for the application.

Unfortunately, simple combination of these two components doesn't meet the expected performance. Even though the targets are successfully deployed and modifications to the environment are correctly registered, there is one problem that appears. When user moves an object and relocates it, spatial mapping registers this change and generates freshly updated mesh around object's new location. However, spatial understanding is not active anymore and therefore the mesh generated around object placed on its original location is still there and won't be deleted or updated. This scene therefore contains two independent meshes of the same object but at different locations.

Level area

The advantage of the AR entertainment is the infinite number of levels. The opportunity to dynamically create game level in every room in which the application is launched offers many opportunities. However, it brings several problems as well. For instance, in order to scan the room using spatial understanding, the scan needs to be finished eventually. That means to decide whether the scanned location is extensive enough to suffice the application requirements.

Once the total area is scanned, process of generating level can begin. Details about the virtual objects has to be settled. The question of how many objects are going to be generated for given scanned area should be answered. The application needs to know where to generate holograms and under what circumstances.

Level orientation

Working with randomly generated objects includes the fact that holograms are almost always spawned on different locations. User doesn't know where to look for spawned objects,

therefore he would always need to search for them at the beginning of the game. Due to headset's small field of view, generated object leaves this field even while user is performing slightly wider movement. Always searching for holograms to play the game with totally consumes the entertaining experience. Furthermore, my application requires casually spawned targets as game content which purpose is to be collected. These objects will be small and would not move, therefore without any additional help it wouldn't be possible to find them easily which would lead to inefficiency and indisposition to play the game.

User Interface

User needs to know in what application phase he is situated. Along with this information, user also needs to understand how is he supposed to act. For instance, the application needs to scan the surroundings first. This action requires for the user to move around the room. This fact has to be interpreted to user in some way. Besides requested proceedings, the user should have satisfactory overview about his game progress. He needs to know how many actions he needs to perform in order to complete the level. In other words, game user interface should inform about a game score. Finally, another important state that it should be informed about is the end of the game so the user understands that no additional action is required. the example could be completed with another user actions, such as score registration, game difficulty or any other options that are commonly available inside game menu.

4.3 Application design

After inspecting all key problems, the application design can be shaped. In result, the developer has to settle on specific solution. To create entertaining application that would also satisfy given requirements, various possible designs needs to be considered. After analysing such alternatives, the final decision requires reasonable justification. This section contains solutions to provided problems from the section 4.2 above.

Pre-scan of the surroundings

The application should prevent from happening the situation when the scene has two generated meshes of the same object but at different locations. The best way to avert such situation is to disable the mesh generated by spatial understanding after its necessary usage it's done. However, once the spatial scan is finished and the mesh is generated and saved, it can't be deactivated, nor deleted. After spending some time to research this issue, I found the possible solution. It contains dividing the game into scenes. Each scene could contain different spatial components as the spatial understanding feature is only essential at the beginning of the application. After spatial awareness is completed and saved, this component is not needed anymore. Second scene that wouldn't contain spatial understanding could work only with spatial mapping and its capability to frequently update modifications made to environment. This way the problem of having two separated records of object locations could be solved.

Level area

The process of scanning the room in order to understand the surrounding and its components has to meet its finish. Generally, there should be no restriction on how large the

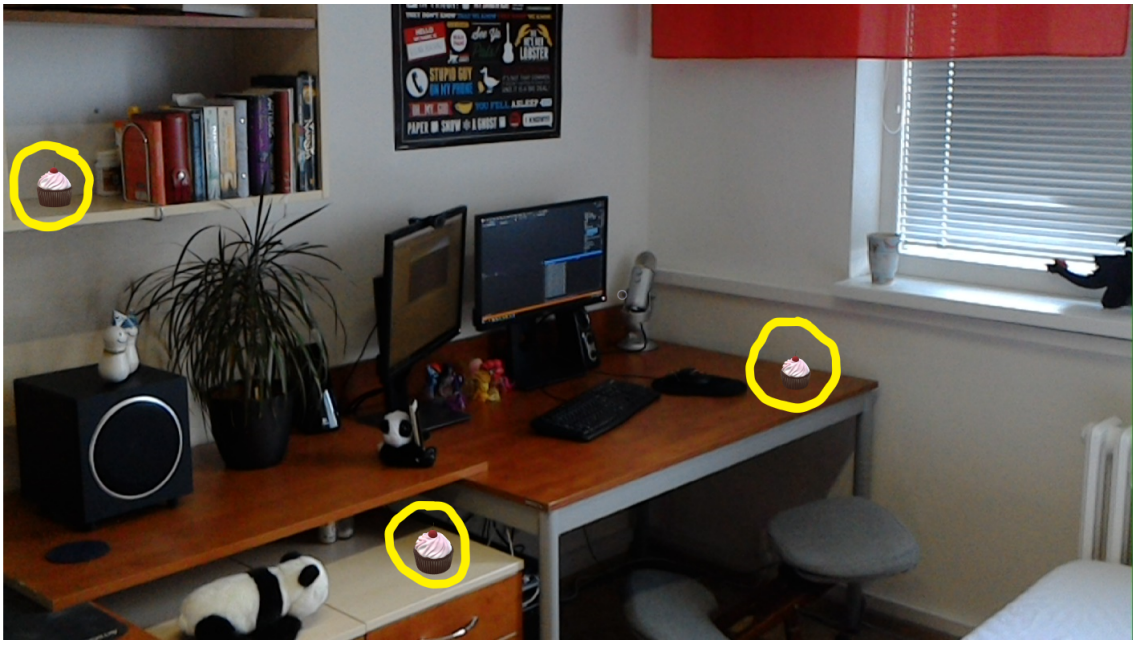

Figure 4.2: Possible scene with randomly generated virtual cupcakes as targets

room should be and the user could decide in what area he wants to perform. Therefore the application needs to accept user's input. When this information is processed by the application, holograms are ready to be spawned. As the controlled character is main hologram, it should be spawned first. Character's position will take place on the surface in front of the user so it is the first hologram user meets.

The aim of the application is to use the real objects in order to create path for the character to collect the targets. That means the randomly generated targets have to be spawned in the location that is not that simple to reach. They need to be placed high enough so the character needs to at least jump up on the surface and at the same time low enough so the character can comfortably move around the surface without crouching. Targets should also spawn in different locations so they won't be close next to each other and game won't be over after one reached location. In result, generated scene could look as displayed in Figure 4.2. Randomly generated targets are represented by cupcakes.

There will be maximum of five spawned targets at a time so the user won't be overwhelmed with application score expectations. The area offers more locations than the maximum number of spawned targets. The application could suggest to recreate the level after all the targets are collected. This way the application offers more than one level inside the single room.

Directional indicator

The application would contain several virtual objects placed all over the room. The location of all these holograms is randomly generated. To save the user from repeatedly searching for the holograms, any directional indicator would be useful. Each hologram should have its own indicator. Two variants of the cursor placement comes to mind. The indicators could all copy the inner border of the user's field of view, each showing the direction to its

Figure 4.3: Two possible cursor designs. Border (left) and center circle (right) directional indicator. Picture differs display part where holograms are visible from non-hologram-visible part

lead. The second possible cursor appearance could be in the middle of headset's field of view. I rejected the first idea due to wide cursor movements even while brief turn of the user. Another reason against this option occurs when realising that as the indicators copy the inner border of the display the headset provides, they simultaneously alert the user of how small the field of view actually is. The idea of both types of directional indicator is presented in Figure 4.3.

Even though each hologram needs to be found, the user should recognize which cursor to follow in order to find desired object. The indicator of the character's location could be distinguished by color, size or the placement in the user's view. To make the application more enjoyable, all the indicators could have different color. When considering the placement, the character's indicator could be in the middle of user's field of view while the target's indicators could be shown around the inner borders. This variant however appeals a bit disorganized and may cause confusion. Additionally, it includes disadvantages described above. I settled on the option to differ the indicators in size in a way that the only bigger cursor would lead to the character and the other smaller indicators would lead to targets.

Navigation

To keep user informed about application states or game score, developer needs to decide how these information would be advised. Most widely used options are to provide such information via voice or navigation text. Voice usage requires user to take attention since the game start or needs to be supported with the text so the user can reread the instructions when needed. Due to simplicity, I decided to use the text option only in form of appropriately short instructions in order not to overwhelm the user with a load of text. This decision is illustrated in Figure 4.4, showing centered directional indicator in addition. These short instructions will be present from the beginning of the game, through its main phase up until the situation when the user collects all the targets. During the phase when user collects the holograms, the instruction text would show briefing about game score. There is a possibility to show this information only when the score state changes, however, to ensure that user is aware of his score permanently, the text will be constantly visible.

After all of the targets are collected and the game seems to be over, two potential options are feasible. The game could be over completely or the user can meet with the possibility to play the game again with newly generated targets which would take place

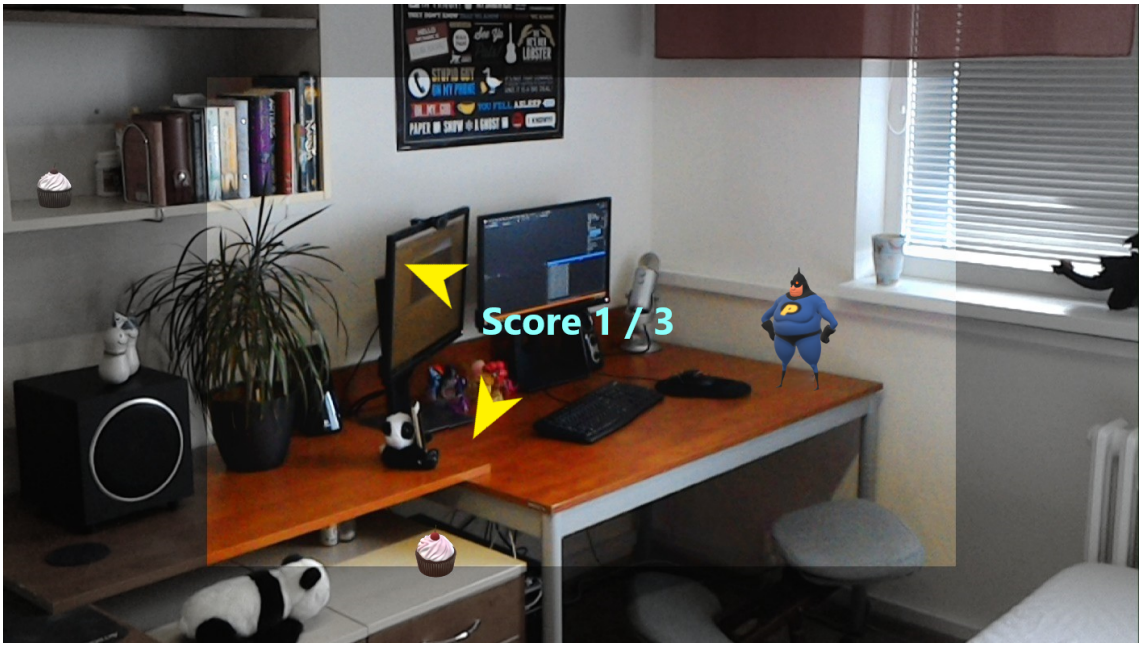

Figure 4.4: Navigation text showing actual score status with additional circle cursor

on different locations than the attempt before. As the environment scanning mostly offers more locations that were already used, this game sequel could be put to use.

When considering the variant when the level is recreated after the end of the game, the application needs an input to decide whether to perform this action or not. This decision could be collected through an in-game menu or via allocated gamepad button. I decided to take user input from the gamepad only so the user won't be confused when to apply which type of input. The application takes no other input therefore developing in-game menu for this one time use only is not necessary. Final decision is to take the input from the gamepad via allocated button.

Command control

In order to make any game utilizable, the application has to be able to process user input. In my application, input serves especially to control virtual character and afterwards to make different gameplay decisions. Requirements created for my application contain the information that the virtual character should be controlled by any kind of gamepad. As an applicable possibility, two widely known gamepads present as the option – PlayStation gaming controller or Xbox gaming controller. Headset Microsoft HoloLens offers the possibility to connect such controller via Bluetooth so my university provided me with wireless Xbox controller. This device is applied as main and only input option for the final application.

Even though HoloLens are able to process input provided by gamepad, such input can only be used to control 2D inner applications. Other 3D immersive entertaining applications can't process this kind of input. To make the application gamepad friendly, source code of such application needs to implement an understanding of the specific input.

4.4 Application backstory

To collect targets without any additional explanation could seem slightly useless. The application of which main purpose is to entertain the user should ensure that the user would be drawn into storyline. As many such applications, even my game disposes with its own backstory that could be provided to user at the beginning of the game.

> Imagine the whole world under lock-down. There is a pandemic going outside of your window. This situation takes longer than you thought and you run out of ideas how to entertain yourself. You spend all your time sitting on the couch, watching series and feeding yourself with month-worth of groceries. This is not the way. You need to stop this madness. And there comes Ethan, the saviour. He is willing to find all your stash and eat it for you so you won't gain any weight. Help him to navigate around your room to collect all the cupcakes. Be aware that after each dessert Ethan increases in weight and is not able to jump as high as before. Good luck!

Chapter 5

Implementation

Developing an immersive application in real environment offers the opportunity to dynamically generate different levels anywhere the application is launched. This process however brings the responsibility for the application to adapt to any surroundings. As the user is able to move around without any restrictions in the environment that is not predefined, one of many problems that need to be solved is placing virtual objects.

This chapter provides solution to this problem as well as the structure of the final application (section 5.1) and its states and introduces the assets used in application development (section 5.2). Next part of the chapter resolves the problem of placing virtual objects onto real surface (section 5.3) and explains the process of detecting modifications made to environment in section 5.4. Later in the text, the problem of understanding controller input is described in section 5.5, as well as is the navigation text (section 5.6) used to inform the user about application states. Finally, basic gameplay is introduced in section 5.7.

The final application was developed using cross-platform game engine Unity[1] which contains several useful development tools and enables importing custom assets. One of the included assets represents the Mixed Reality Toolkit[2] by Microsoft which main focus it to simplify development of HoloLens applications. Using this toolkit, developed application can be deployed directly to the headset or to simulated environment called HoloLens Emulator. Source code of the final application is written using programming language C# and can be deployed within Microsoft Visual Studio development environment.

5.1 Application structure

The application consists of six main components. Each of them works separately and has its own responsibility. Game object *Text* incases navigation text that informs the user about application states, score of the user or actions expected from the user. *Input Manager* processes all the input accepted from the Xbox controller. Such input is responsible for both character's movement and user's decisions on application states. *Cupcake Manager* is an object used to initialize target objects. After initialization part is over, this object is afterwards used to access all cupcakes used in the game so the other objects can work with them properly. To assure that user is always aware about holograms location, game object *Directional Indicator* depictures cursor for each hologram in the game. Game object *Collision Manager* detects all existing collisions and alerts if any collision occurs between

[1] https://unity.com/

[2] https://github.com/Microsoft/MixedRealityToolkit-Unity

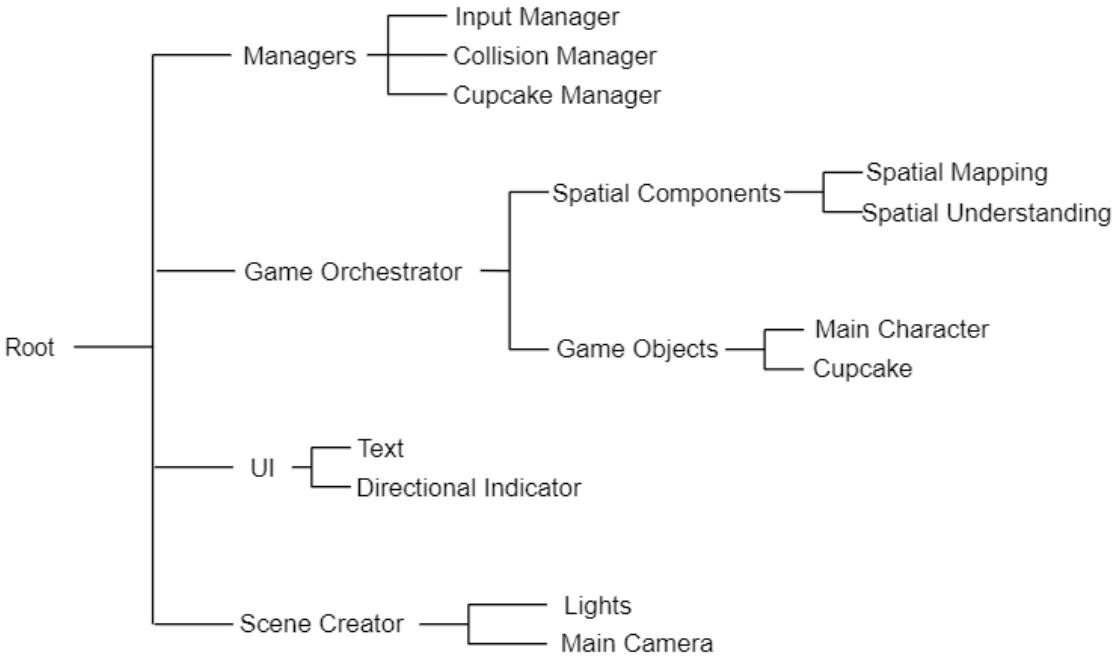

Figure 5.1: Application scheme based on game objects in application hierarchy

main character and target object so the application is able to react on this state change. Finally, the application also contains *Scene Creator* which stores HoloLens components of main camera and scene lights so it is possible for the user to see generated objects. All these objects are utilized by the key game object *Game Orchestrator*. This object switches game states based on flags generated by all these described game objects. The responsibility of this key object is also to create the main scene after the environment pre-scan is done. Scheme pictured in Figure 5.1 aggregates all of application game objects based on their placement in application hierarchy.

Application states

The application automatically starts the phase of *Scanning* the surroundings right after it is launched. The scan is done using spatial understanding component. Continuously generated mesh is, in this application state, visible to the user so he understands what part of the room is already scanned. During this phase, the navigation text centered in the middle of the user's field of view displays commands used to control the character and game tips.

After required area of the room is scanned and processed, the application switches to its second state, *Generating Objects*. This phase works with data obtained from scanning. After adding conditions designed in subsection 4.3, game object Game Orchestrator spawns firstly the main character and then maximum of five targets all over the room. Simultaneously, one directional indicator is assigned to each projected hologram.

When all the holograms are placed on their location, the application switches to *Game* phase. Application persists in this state until all the targets are collected. For the user it is now possible to control the character via gamepad. All targets are conditioned to take place in locations which are not easily accessible for the character. To reach the target,

user needs to use additional real objects to create a path for the character. Using spatial mapping component, the application registers the changes made to environment and is able to react. Every time the target is collected, the application removes the target object and its directional indicator from the game and adds a point to user's score. This phase ends after all generated targets are collected.

The last application state is *Respawn*. To reach this state, the user needs to collect all generated targets. Navigation text then displays the option to continue playing the game with newly generated targets with the location different from the previous targets. If the user decides to choose this option, the application switches to the state two, which is Generating objects. User is able to finish this loop after all locations found by spatial understanding are reached. The game is then beaten.

5.2 Used assets

This project works with several assets. Target objects are presented as cupcakes. For these objects, free asset *Cupcake*[3] was imported.

Used main character object is represented by *ThirdPersonCharacter* provided by Unity along with other Standard Assets[4] characters. Its prefab contains script to control the character using classical computer components. Two key game objects are illustrated in Figure 5.2. To make it possible to control character using Xbox controller, custom script was made. Free package *Xbox Controller Input for HoloLens*[5] from Unity Asset Store provides plugins that enables custom application to read the input from Xbox One S Wireless controller.

Part of the application design in subsection 4.3 is component *Directional Indicator* which main responsibility is to simplify constant searching for holograms. Imported asset is based on existing solution[6] and customized.

Another important component is *Mixed Reality Toolkit*[7] provided by Microsoft. This asset provides key components for HoloLens application. Assets used in this project are e.g. *Main Camera, Lights, Spatial Understanding* and *Spatial Mapping*. Besides these game objects, toolkit also offers materials or the ability of drawing virtual objects.

5.3 Placing virtual objects into real environment

Virtual objects can be generated after the application is finished with the Scanning phase. Game object Game Orchestrator firstly hides all visible objects, i.e. main character and all targets. After the scan is finished and the phase Generating Objects begins, these objects are placed and made visible. Custom function *GameOrchestrator.ShowObjects()* firstly finds location for the main character and then processes other data to place the targets. At the beginning when the application is launched, starting position of the user is saved

[3] https://assetstore.unity.com/packages/3d/characters/cupcake-69486

[4] https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2017-3-32351

[5] https://assetstore.unity.com/packages/tools/input-management/xbox-controller-input-for-hololens-70068

[6] https://github.com/kaorun55/HoloLens-Samples/blob/master/Unity/PlaneFindingDemo/Assets/HoloToolkit/Utilities/Scripts/DirectionIndicator.cs

[7] https://github.com/Microsoft/MixedRealityToolkit-Unity

Figure 5.2: Used assets for main character and targets. Left picture shows main character controlled by user via gamepad. Its main purpose is to collect all targets. On the right is illustrated the target in appearance of the cupcake.

and amended. After the modification, this position is one and a half meters forward on the surface. This is where the main character takes its place once it is spawned.

Data obtained through scanning are processed using Spatial Understanding game object. This object instance creates list of possible locations where the objects can be spawned. These locations are detected based on given conditions about location extent or height. Each position in the list is represented by 3D vector. To make sure that such positions will not be easily reached by the main character, the list is conditioned to retrieve only requested locations. Condition used in the application says that the location should be half a meter upon the floor, it should take place in the maximum height of two meters and the surface should have at least zero point five meters of facing clearance so the main character can step into the area. To prevent targets to be spawned close to each other, each target has its own radius set to 0.4 of meter so other targets can not be spawned inside this range. Snapshots from the application that illustrate generated object are visible in Figure 5.3.

As the application offers the possibility to recreate the level after all targets are collected, while generating targets, all used positions are continuously saved so they can not be used repeatedly. The game is beaten when there are no unused positions left and all targets are collected.

5.4 Mapping modifications made to environment

To collect objects that are too high for the character to reach, user needs to improvise and create a path from nearby real objects. Design described in 4.3 suggests to use different spatial components to accomplish different results. My application automatically starts scanning the surroundings using Spatial Understanding component. Data obtained through this type of scanning provides specific locations for the targets to be spawned. When the

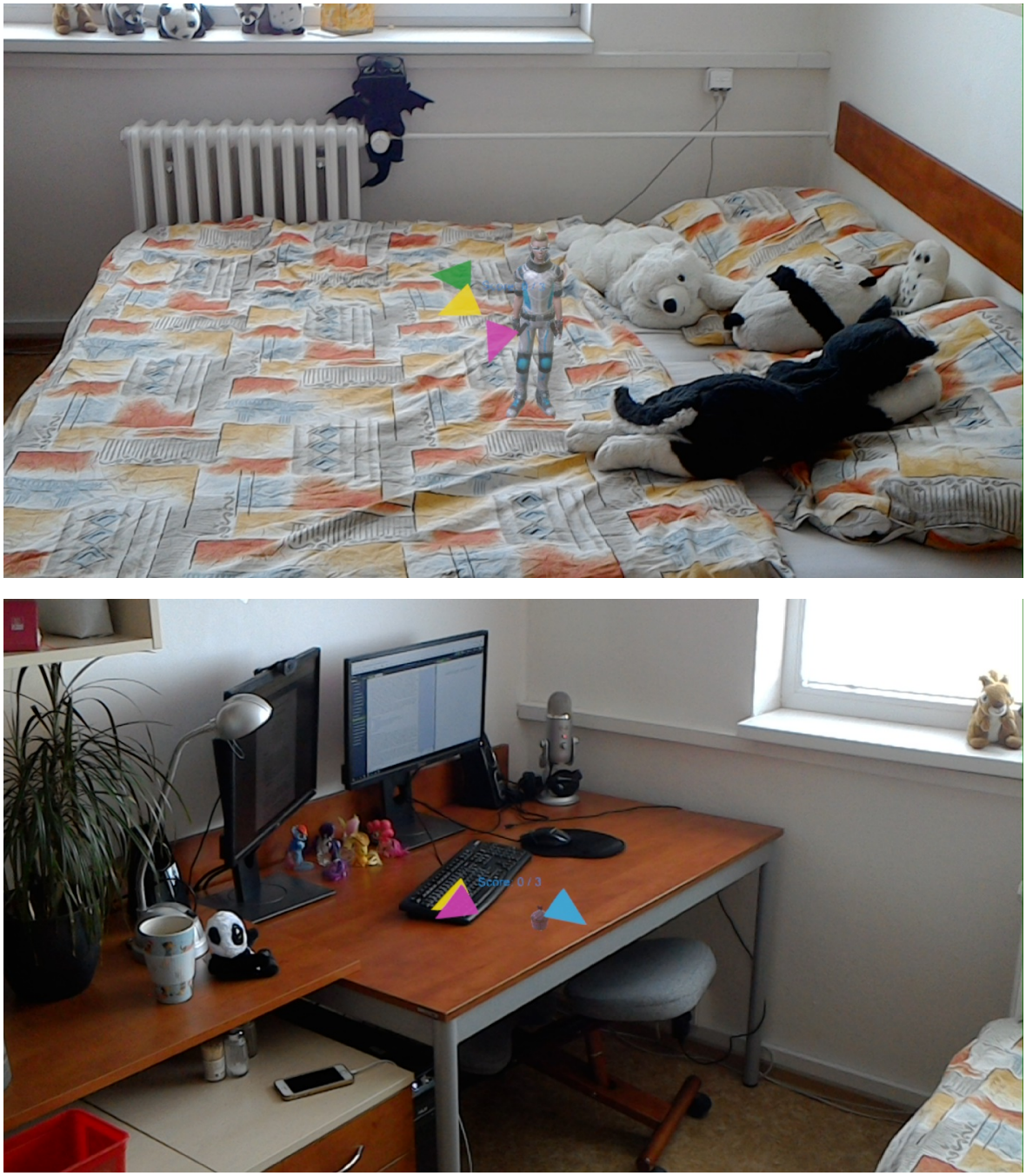

Figure 5.3: Objects generated after primary scan. First picture shows main character spawned in front of the user's starting position. Directional indicators point at three spawned targets. Second illustration displays generated cupcake (in the middle of the picture) on the table as a surface. The cupcake is located in the middle of the table. Blue indicator points to the main character's location. Other two cursors lead the way to another two spawned targets.

application is situated in Game phase, Spatial Mapping component is used to provide frequently updated mesh so all modifications made to the environment are detected.

Problem that occurs when combining these two components is that after Spatial Understanding is finished with scanning, it can not be updated or deleted. Environment changes are therefore detected by Spatial Mapping and its mesh is updated but mesh generated by Spatial Understanding stays the same without any modifications. In the eyes of the character, the original moved object did not disappear and an object of the same shape appeared in another location. I tried to use described solution to divide the game in two scenes. The first one was responsible for primary scanning and generating and placing virtual objects. Second one was the game itself and only contained Spatial Mapping component. This way, Spatial Understanding component should be destroyed while switching scenes as this is the behaviour of the Unity engine. However, this design did not work out as the game object was not destroyed so the generated mesh was still present.

Finally, I was able to solve this problem by amending the source code of Spatial Understanding instance which is part of the Mixed Reality Toolkit. As this object can't be destroyed, I had to change its behaviour from static to dynamic. Its mesh is updated only while scanning until it is finished. As I need to finish this process in order to obtain the data, I have to assure that once the process of scanning is finished and collected data are processed, the component switches to its scanning phase again. This could be established by launching the scan again. However, game object itself is not able to begin the process of scanning once it was finished. I changed the source code so the Spatial Understanding scanning process could be launched again, once the main character and targets are all placed. This time however, this process is prevented from being finished to make sure its data would never be processed. This way, Spatial Understanding mesh is always up to date. In practice, I use Spatial Understanding to scan the environment to obtain data about locations where the targets could be spawned. After all objects are generated, Spatial Understanding component is launched again so its mesh is always up to date. This time, Spatial Understanding scan won't be finished until the application is closed. Meanwhile, Spatial Mapping mesh is constantly updating as well and is used to detect modifications made to environment.

Even though Spatial Understanding now generates mesh that is constantly updated, this mesh is only rough estimation of the environment and does not contain such details as the mesh generated by Spatial Mapping. This is the reason why Spatial Mapping mesh can't be fully replaced and why the application even after additional changes still uses two spatial components. This usage does not affect the application performance in any way as the application processes data only after the Spatial Understanding scan is finished which does not happen. Situation when the application detects changes is illustrated in Figure 5.4.

This design also provides the solution for another problem. In the original behaviour of the Spatial Understanding component (without my code changes), when the scan is finished, the scanned area is bordered by virtual walls. Once the user moves past these walls, the headset loses its spatial awareness and becomes lost. This pauses the application until the device restores its spatial awareness again. This situation leads to forgetting original positions of all generated objects. Once the headset locates its position in the real world again, it newly spawns the objects if anything. Using Spatial Understanding in a way when it constantly scans the surrounding eliminates this problem completely.

5.5 Application input

The application is controlled via Xbox controller which is connected to the headset via Bluetooth. For adjusting the application so it understands the input from the controller,

Figure 5.4: Spawned cupcake in the first picture is too high for the character to reach. User needs to use real object to create a path. Second picture shows that the application detected environment change and mapped the chair. Character now reaches the target.

asset called *Xbox Controller Input for HoloLens* is used. The application does not process core interaction options, such as gaze, gestures or voice commands. The gamepad is the only input. As it is common in third person games, main character's movement is controlled moving left thumbstick, it jumps by pressing A button and is able to move while crouching when the button B is pressed. There is also a possibility to show generated spatial meshes so

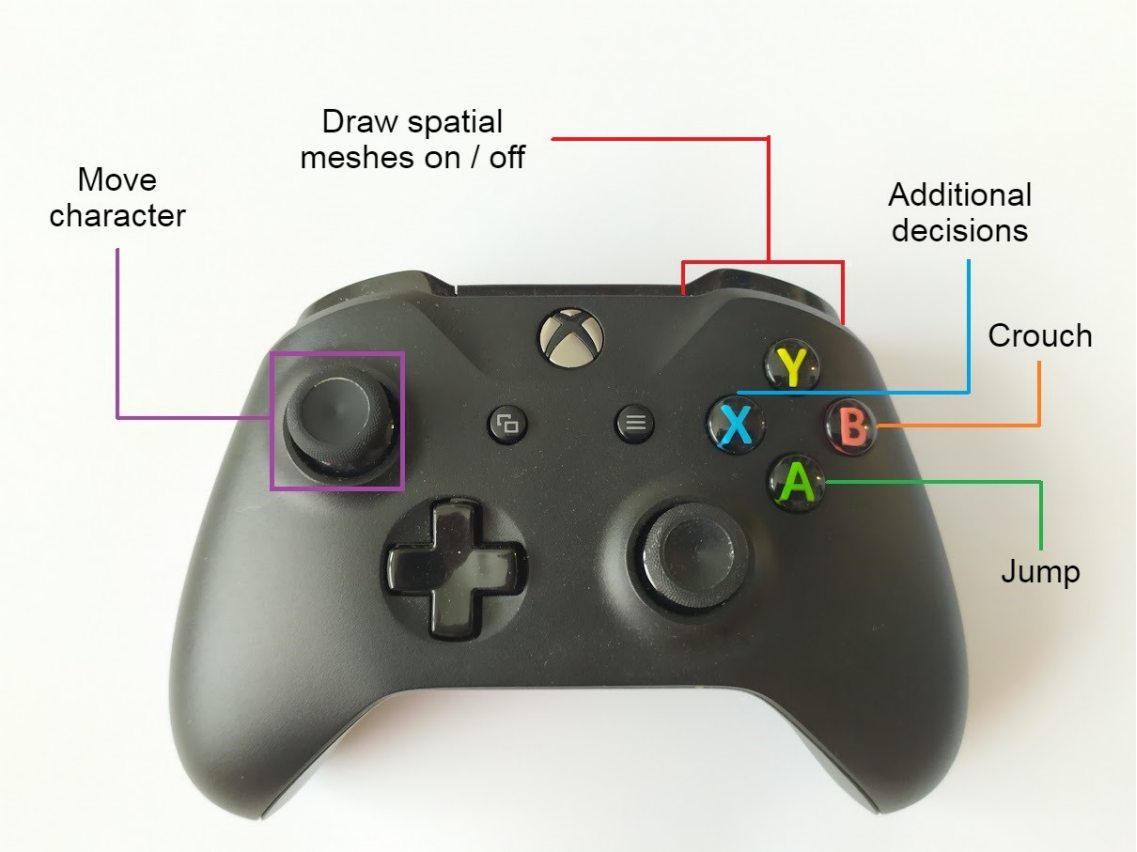

Figure 5.5: Application input options

the user can better navigate in limited places or see how the transferred object is detected. This option is switched on and off by pressing RB button. At the beginning of the game, spatial meshes are visible so the user can see what area of the room is already scanned. After all objects are generated, spatial meshes become invisible and can be displayed using this button. The application does not offer any menu, however if user is asked to decide future steps taken by the application, the user is always told what button to press. For instance when user wants to stop initial scanning or respawn the targets after they are collected. All application input options are illustrated in Figure 5.5.

If the application is launched without any Xbox controller connected, this situation is handled so the primary scan is executed, all objects are generated but there is no way to control the character so the game is not playable. The controller can be safely connected or disconnected during the application runtime.

5.6 Navigation text

Text displayed during the application is implemented using *3D Text* Unity game object. Navigation text should be always visible to user. In HoloLens applications, the user represents the main camera so in order to prevent the text being fixed in one place I used additional Unity components. One of them is *TagAlong* which makes the text to follow user's gaze. Metrics are also different inside HoloLens as are in classic 2D applications. The text has to be farther away with bigger font size and needs implemented speed so it

can follow the user fast enough. Another useful component used is *Billboard* so the rotation of the text is not fixed and the text can always face the user.

When the application is launched, the navigation text asks the user to take a walk around the room to obtain proper scan of surroundings. While this activity, the text shows tips and tricks about how to control the character, finish the phase of scanning and how to play the game. Later, when the application is situated in Game phase, the text constantly informs about game score. Afterwards, the text invites the player to make a decision whether he wants to recreate the level and play again.

5.7 Gameplay

The goal is to collect all the cupcakes. As it is in real life, there is a penalty for eating cupcakes. The main character, Ethan, gains weight each time he collides with the cupcake. This makes him heavier and he is not able to jump as high as before. Script *ThirdPersonCharacter* provided by Unity within Standard Assets provides the variable called *m_JumpPower* which reflects the power of the main character jump. This value is used to calculate the position after character has jumped. Its default value is set to 6. After main character collects the cupcake, this value is decreased by 0.4. Decision to proceed with this constant was made during developing and testing the application. In result, after two or three cupcakes collected, the main character is not able to jump high enough to reach higher surfaces. The user is now urged to create additional path.

In addition after each collected cupcake, main character becomes slightly larger. This is achieved by amending character's skeleton. Each collected cupcake increases the scale of character's body. The scale is represented by 3D vector. Only coordinates x and z are increased as the character is supposed to grow only in weight and depth, not in height. Constant of increasing these coordinates is set only to 0.03. This is due to the decision to make it possible for the user to recreate the level. If the constant would be higher, main character after collecting more than around 15 cupcakes would be too large to fit in small locations where the cupcakes could be spawned.

This feature creates the opportunity for the user to contemplate about game strategy to collect cupcakes from the highest places to the lowest. Once all cupcakes are collected and user decides to recreate the level with new locations, main character Ethan relaxes. This means that in newly generated level he is able to jump high again as he is rested but he still remains thick.

Chapter 6

Experiments

Aim of this chapter is to illustrate the results of experiments made on the final application. The goal was to obtain feedback about the application functionality, its intuitiveness and collect possible suggestions for future work. Chapter is divided in four sections where the section 6.1 describes the process of how the experiments were accomplished. Section 6.2 analyses the behaviour of the participants during the application phases and provides their comments. Section number 6.2 provides the results of the experiments based on the User Experience Questionnaire and the final section 6.4 introduces the possible suggestions for the future work on the application.

6.1 Experiment description

Six people participated in experiments, all of them in age from 22 to 25 years old. None of them had previous experience with Microsoft HoloLens headset. Demographic data about the participants can be found in table 6.1. Aim of the experiment was to observe participants reactions to an unknown application and collect information whether the design of the application is user friendly and the concept could be easily learned. No game-specific instructions were told to the participants, in order to simulate a scenario, when a player downloads and plays an unknown game for the first time. They only obtained gamepad, already connected to the headset. The process was documented using video recording for additional investigation.

All participants were introduced to Microsoft HoloLens so they would be able to launch the application on their own. Without any additional introduction they started to play. My attention was focused on how many times I would have to navigate the participant so he would be able to complete the level.

Participant	Age	Sex	Previous exp. with HoloLens	Gamer
A	25	M	no	yes
B	24	M	no	yes
C	22	F	no	no
D	22	F	no	yes
E	25	M	no	no
F	23	M	no	yes

Table 6.1: General demographic data about the participants

6.2 Experiment observation

Scanning phase

During the phase of *Scanning*, each participant understood that he is asked to walk around the room in order to scan the environment. However, the action required to finish the scan, pressing the allocated gamepad button, wasn't clear to all the participants. Participant C asked whether he is already in the game or the application is still scanning. However, I did not answer as two previous participants didn't trouble with this action. After additional minute, participant C read the instruction provided by application navigation text and understood what is the required action. In following interview I was given feedback about what was the problem. For people with no experience with the augmented reality headset, the most interesting part of the scanning phase was to observe how spatial meshes align to the real world surface. Watching this process, participant C totally ignored navigation text so he did not understand how to finish the scan.

Game phase

When spatial meshes disappeared and directional indicators came to the scene, all participants were aware that the *Game* phase begun. Observing the scene, watching the cursor to change positions as the players move their heads, all of them understood the meaning of the directional indicator. All participants, except participants B and F, firstly found the location of all the target cupcakes and after that they started to play. I was told that once they saw what objects are in the scene, the concept of the game was obvious – to collect all the cupcakes. Having previous experience with gamepad, simple character navigation was an intuitive process. None of them seem to have problem moving the character around the room and to make the character jump. Three of the six participants would prefer different character navigation as the character is not able to change the movement direction while jumping. However, they didn't show any additional struggle with navigating the character. All participants welcomed the feature to switch on the drawing of spatial meshes and they found it very helpful.

Four out of the six participants did not understand the information about how the character slowly looses its ability to jump after each cupcake. All of them claimed that they either didn't understand or didn't see this information provided during the scanning phase. Due to this problem, all participants collected more easily reached cupcakes first and then struggled collecting the rest. Four of the participants intuitively tried to move the real object in order to create path for the character. Another participant, player E, understood this possibility after he saw constantly updated meshes in practice. Player F needed my help to understand this possibility.

All players completed the level successfully. Needed time for solving the level wasn't measured as the experiments were taken in different environments in which different number of targets were spawned. However, only one player chose to continue playing the game and decided to respawn the targets. This is due to difficulty of obtaining the last target with the lowest jump possibility of the character. This experience made participants exhausted and unwilling to play the game again.

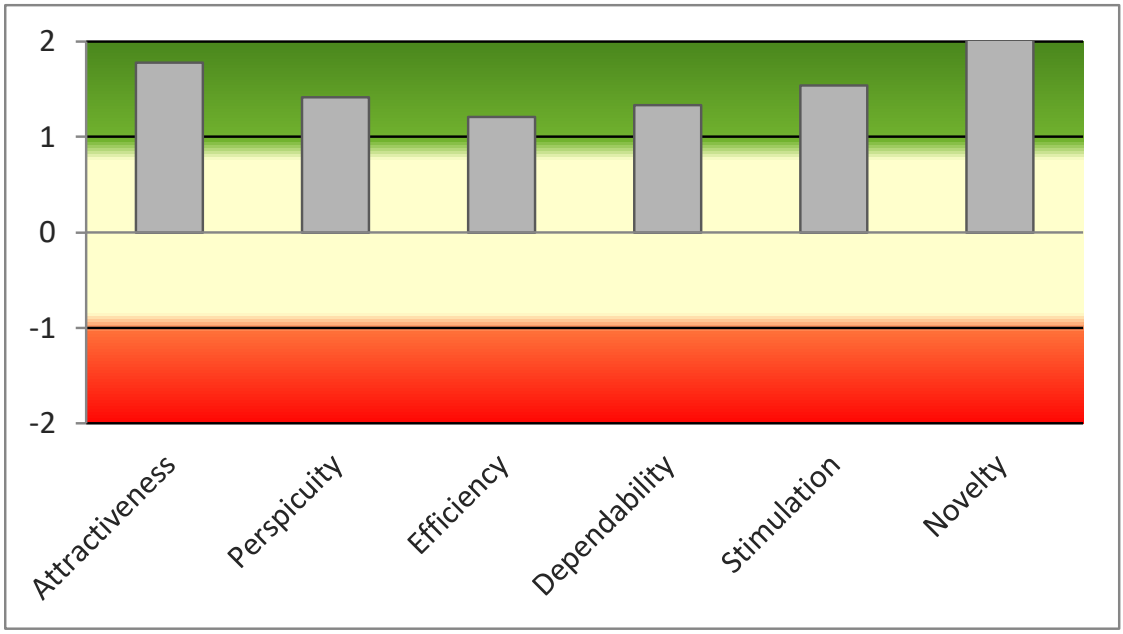

Figure 6.1: Chart representing application rating based on User Experience Questionnaire

6.3 Experiment results

All participants were afterwards asked to fill the questionnaire. I decided to rate my application using User Experience Questionnaire (UEQ) [18]. UEQ measures six key characteristics of the studied application:

- *Attractiveness* – Overall impression of the product. Do users like or dislike it?
- *Perspicuity* – Is it easy to get familiar with the product and to learn how to use it?
- *Efficiency* – Can users solve their tasks without unnecessary effort?
- *Dependability* – Does the user feel in control of the interaction?
- *Stimulation* – Is it exciting and motivating to use the product? Is it fun to use?
- *Novelty* – Is the design of the product creative? Does it catch the interest of users?

UEQ provides 26 pairs of opposite attributes that describe the application. Participants had the opportunity to rate the attribute in seven stages. Example:

attractive ○ ○ ○ ○ ○ ○ ○ unattractive

The results are visible in table 6.2. Based on the answers, the evaluation chart was created. Previously filled seven stages were interpreted as a number between +3 (extremely good) and -3 (horribly bad). The range of the scale on the chart is between +2 and -2 as it is extremely unlikely to observe values above or below 2. Therefore, even a quite good value of +1.5 would on a scale range of +3 to -3 appear not as positive as it really is.

The graph is divided to three stages, where the neutral zone is represented by the area between +0.8 to -0.8. As the graph (Figure 6.1) displays, all results can be counted as

	A	B	C	D	E	F
Attractiveness	1.17	2.50	1.50	1.83	2.00	1.67
Perspicuity	2.50	1.50	1.75	1.00	1.50	0.25
Efficiency	2.00	0.50	0.50	1.50	1.50	1.25
Dependability	2.00	1.25	0.25	1.50	1.25	1.75
Stimulation	1.00	2.00	1.25	2.00	1.50	1.50
Novelty	1.25	2.75	2.00	2.50	2.50	2.25

Table 6.2: Application rating from the participants. Player C rated the dependability of the application due to her experience with angle-wise generated meshes. Player F found application perspicuity really complex as he did not understand that drawing spatial meshes can be switched on and off.

Scale	Mean	Benchmark	Interpretation
Attractiveness	1.78	Good	10% of results better, 75% of results worse
Perspicuity	1.42	Above Average	25% of results better, 50% of results worse
Efficiency	1.21	Above Average	25% of results better, 50% of results worse
Dependability	1.33	Above Average	25% of results better, 50% of results worse
Stimulation	1.54	Good	10% of results better, 75% of results worse
Novelty	2.21	Excellent	In the range of the 10% best results

Table 6.3: Numeric representation of benchmark results

positive as they all appear above the value +0.8. As it is visible from the chart, the lowest rate obtained the *efficiency* attribute. This could be due to lack of understanding of the character loosing his ability to jump after each collected cupcake. Once the character is not able to jump that high, the process of collecting remaining targets takes longer, especially if they are placed on the highly positioned locations.

Second lowest rate belongs to the attribute *dependability*. This result is connected to the application ability of updating meshes. This process takes a few seconds and the headset is not able to detect shiny or too dark materials. Another problem that occurred during the experiments was that the headset generated mesh angle-wise which resulted in character falling from the surface so the user had to start his climb all over again.

The application obtained best result in the *novelty* characteristics as it is developed on attractive modern technology, Microsoft HoloLens. Second best rating belongs to the attribute *attractiveness* which proves that the participants consider the application interesting and pleasant.

Benchmark

Another chart created based on the UEQ results shows the application rate in relation to other existing applications measured by UEQ. Benchmark data set contains data from 20 190 participants from 452 studies. The application appear to be above average in every characteristic, with the rating of good in stimulation attribute and attractiveness and excellent in novelty. Benchmark results are visible in Table 6.3 and the Figure 6.2.

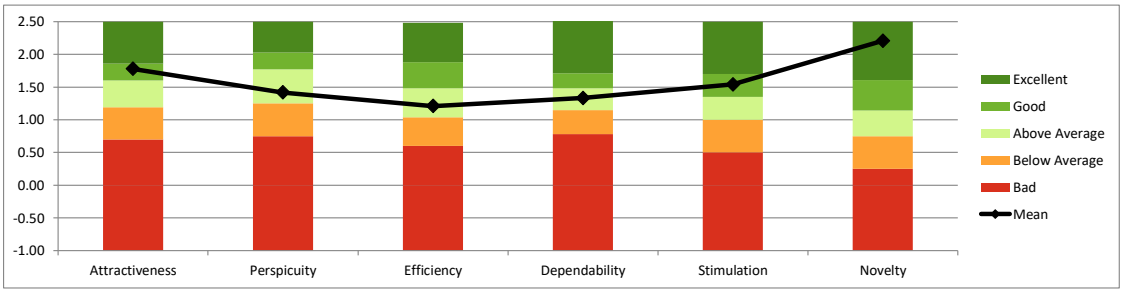

Figure 6.2: Chart representing application rating in benchmark

6.4 Suggestions for Future Work

This work was aimed to achieve required application functionality with the purpose of entertaining. As this was accomplished, goal of the future work could be adding more game enhancements. So far, there are no effects, visual nor acoustic except character moving. The character could be provided with the quality running and jumping sounds and the acoustic effect while collecting cupcake, for instance eating sound or the sound of the victory as the score increases.

Future work on the user interface is to recreate the information text about game score to more game-like design. There could be pictures of collected cupcakes instead of the plain number. User interface could also contain side bar that would inform the user about the jumping ability.

Game itself could gain more entertaining touch using different types of food that needs to be collected. This types could change after each level or there could be more types in one level. Another possibility for future work is to add gravity for the targets so instead of character climbing the path upwards, the shelf on which the target is spawned could be uninstalled from the room so the target would fall on the floor and could be easily collected.

The level itself could be also enhanced with additional game objects that would affect the main character. There could be resting stations that would provide relax for the character so he can regain his ability to jump. On the other hand, the scene could contain objects in appearance of the hole which would have to be avoided by the character so it won't fall into it.

Chapter 7

Conclusion

The goal of this thesis was to realize the cooperation of the virtual objects with the real world objects using mixed reality headset Microsoft HoloLens. This function was demonstrated on the application within which the user is able to control virtual avatar with use of the gamepad, connected to the headset via Bluetooth. Main character is supposed to collect randomly generated targets with use of the path, dynamically created from the real objects. The beginning of the thesis is allocated to theoretically introduce the reader to the augmented reality and the headset Microsoft HoloLens. After that, the thesis provides analysis of the application design, following with interesting implementation details and the process of testing and experiments.

The final application was developed using the Unity game engine. The core functionality depends on the spatial meshes generated during the application. Both spatial components provided by Microsoft HoloLens were used. Component Spatial Understanding scans the room in the first state of the application and is responsible for producing data about the scanned environment. Based on these data, main character and the targets are generated into the environment. Second used component is Spatial Mapping which provides constantly updated mesh thanks to which the application is able to detect dynamic modifications made to environment. This work presents the solution on how to properly create the cooperation between these two spatial components without any additional side effects.

The result of the developed application was tested by six participants and rated using the User Experience Questionnaire. The aim of the experiments was to obtain feedback about the application functionality and its intuitiveness. The participants were invited to following interview in which they were asked about their experience with the application and its user interface. The experiments showed that the application is rated as attractive, motivating and modern. Based on the results, the application is rated as above average, compared to 452 other studies. During the process of testing, several suggestions for future work were collected and are included in the text.

Bibliography

[1] André Stork, R. d. A., Oliver Bimber: *Projection-based Augmented Reality in Engineering Applications*. [Online; Accessed 5/8/2020].
Retrieved from:
https://www.researchgate.net/publication/328685112_Projection-based_Augmented_Reality_in_Engineering_Applications

[2] Azuma, R. T.: Predictive Tracking for Augmented Reality. Technical report. Department of Computer Science. 1995.

[3] Azuma, R. T.: *A Survey of Augmented Reality. In Presence: Teleoperators and Virtual Environments*. vol. 6, no. 4. 1997: pp. 355–385. ISSN 1054-7460.

[4] Bimber, O.: *Spatial Augmented Reality*. Bauhaus-University, Weimar. 2004. ISBN 1-56881-230-2.

[5] Brown, L.: *Microsoft HoloLens Review- a glimpse of holographic future*. [Online; Accessed 5/22/2020].
Retrieved from: https://filmora.wondershare.com/virtual-reality/microsoft-hololens-vr-headset-review.html

[6] Dictionary, M.-W.: *Augmented reality*. [Online; Accessed 5/8/2020].
Retrieved from:
https://www.merriam-webster.com/dictionary/augmented%20reality

[7] Dieter SCHMALSTIEG, T. H.: *Augmented Reality*. Mark L. Taub. 2016. ISBN 9780321883575.

[8] Dieter SCHMALSTIEG, T. H.: *Augmented Reality: Principles and Practice*. Addison-Wesley. 2016. ISBN 03-218-8357-8.

[9] Evertt, J.: *Case study - Expanding the spatial mapping capabilities of HoloLens*. [Online; Accessed 5/22/2020].
Retrieved from: https://docs.microsoft.com/en-us/windows/mixed-reality/case-study-expanding-the-spatial-mapping-capabilities-of-hololens

[10] Jansen, M.: *The best augmented reality apps for Android and iOS*. [Online; Accessed 5/8/2020].
Retrieved from:
https://www.digitaltrends.com/mobile/best-augmented-reality-apps/

[11] Mathew, N.: *Why is Occlusion in Augmented Reality So Hard?* [Online; Accessed 5/22/2020].

Retrieved from: https://hackernoon.com/why-is-occlusion-in-augmented-reality-so-hard-7bc8041607f9

[12] Microsoft: *Microsoft HoloLens*. [Online; Accessed 5/22/2020].
Retrieved from: https://www.microsoft.com/en-us/hololens/hardware

[13] Microsoft: *Spatial Mapping*. [Online; Accessed 5/22/2020].
Retrieved from:
https://docs.microsoft.com/en-us/windows/mixed-reality/spatial-mapping

[14] Microsoft: *Voice input*. [Online; Accessed 5/22/2020].
Retrieved from:
https://docs.microsoft.com/en-us/windows/mixed-reality/voice-input

[15] Newsroom, P.: *Best AR Devices for 2020*. [Online; Accessed 5/22/2020].
Retrieved from: https://pale.blue/2020/02/03/best-ar-devices-for-2020/

[16] Nova, M.: *4R's or get your head around Virtuality Continuum*. [Online; Accessed 5/8/2020].
Retrieved from: https://medium.com/@Maria_Nova/4rs-or-get-your-head-around-virtuality-continuum-625e256ddd1d

[17] PAINE, J.: *10 Real Use Cases for Augmented Reality*. [Online; Accessed 5/8/2020].
Retrieved from: https://www.inc.com/james-paine/10-real-use-cases-for-augmented-reality.html

[18] Schrepp, D. M.: *User Experience Questionnaire Handbook*. [Online; Accessed 5/22/2020].
Retrieved from: https://www.ueq-online.org

[19] Sean White, S. F.: *SiteLens: Situated Visualization Techniques for Urban Site Visits . In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. vol. 9. 2009: page 1117–1120.

[20] UploadVR: *HoloLens 2 AR Headset: On Stage Live Demonstration*. [Online; Accessed 5/22/2020].
Retrieved from: https://www.youtube.com/watch?v=uIHPPtPBgHk

Appendix A

Contents of the CD

```
/
├── xfajto00.pdf ..... Text of the thesis
├── tex/ ..... LaTeX source files
├── src/ ..... Source code of the application
├── video.mp4 ..... Video of the thesis
└── README.txt ... Text file with instructions of how to deploy and run the application
```

Appendix B

User Experience Questionnaire

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26