Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies

Bc. Jozef Blažíček

User representation and interaction in virtual reality focusing on presence

Master's thesis II

Supervisor: Ing. Peter Drahoš, PhD.

December 2017

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies

Bc. Jozef Blažíček

User representation and interaction in virtual reality focusing on presence

Master's thesis II

Study program: Information Systems
Field of Study: 9.2.4 Information Systems
Place: Bratislava
Supervisor: Ing. Peter Drahoš, PhD.

December 2017

Zadanie diplomovej práce

Meno študenta: **Bc. Jozef Blažíček**

Študijný program: Informačné systémy

Študijný odbor: Informačné systémy

Názov práce: **Reprezentácia používateľa a interakcia vo virtuálnej realite s dôrazom na pocit vnorenia**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

Všeobecný cieľ:

Vypracovaním diplomovej práce preukážte, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

Špecifický cieľ:

Vytvorte riešenie zodpovedajúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riaďte sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti, alebo o priebežné výsledky Vášho riešenia, alebo o iné závažné skutočnosti, dospejete spoločne s Vaším vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnite zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

Miesto vypracovania: Ústav počítačového inžinierstva a aplikovanej informatiky, FIIT STU Bratislava

Vedúci práce: **Ing. Peter Drahoš, PhD.**

Termíny odovzdania:

Podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

Predmety odovzdania:

V každom predmete dokument podľa pokynov na www.fiit.stuba.sk v časti: home > Informácie o > štúdiu > organizácia štúdia > diplomový projekt.

V Bratislave dňa 13. 2. 2017

SLOVENSKÁ TECHNICKÁ UNIVERZITA
V BRATISLAVE
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4
1

prof. Ing. Pavol Návrat, PhD.
riaditeľ Ústavu informatiky, informačných systémov a softvérového inžinierstva

SLOVENSKÁ TECHNICKÁ
UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY
A INFORMAČNÝCH TECHNOLÓGIÍ

Návrh zadania diplomovej práce

Finálna verzia do diplomovej práce [1]

Študent:

Meno, priezvisko, tituly:	Jozef Blažíček, Bc.
Študijný program:	Informačné systémy
Kontakt:	j.blazicek17@gmail.com

Výskumník:

Meno, priezvisko, tituly:	Peter Drahoš, Ing. PhD.

Projekt:

Názov:	Reprezentácia používateľa a interakcia vo virtuálnej realite s dôrazom na pocit vnorenia
Názov v angličtine:	User representation and interaction in virtual reality focusing on immersion and presence
Miesto vypracovania:	Ústav počítačového inžinierstva a aplikovanej informatiky, FIIT STU, Bratislava
Oblasť problematiky:	Počítačová grafika, Virtuálna realita

Text návrhu zadania [2]

Aplikácie a zariadenia využívajúce virtuálnu realitu zažívajú v posledných rokoch veľký technologický rozmach a popularitu. Taktiež sa stávajú čoraz dostupnejšími. Napriek rýchlemu rozvoju ostávajú v oblasti mnohé problémy a otázky. Vo virtuálnej realite často nie je možné využiť prostriedky a prístupy používané pri 2D zobrazovacích zariadeniach. Ako príklad možno uviesť klávesnicu a myš, ktoré používateľ nevidí a nemôže ich využiť na interakciu s aplikáciou.

Analyzujte techniky reprezentácie používateľa vo virtuálnom prostredí s ohľadom na zvýšenie pocitu vnorenia (angl. immersion). Zamerajte sa na metódy, ktoré umožňujú manipulovať virtuálnym prostredím bez nutnosti držania, či nosenia dodatočného vybavenia. Porovnajte rôzne spôsoby dynamickej reprezentácie používateľa vo virtuálnej realite a interakcie s virtuálnym prostredím použitím bezkontaktných zariadení.

Navrhnite dynamickú reprezentáciu používateľa vo virtuálnej realite a virtuálne prostredie. Reprezentáciu a prostredie použite v návrhu interaktívnej aplikácie. Na zachytenie pohybu končatín a interakciu s virtuálnym prostredím použite zariadenie LeapMotion Orion. Navrhnuté riešenie overte sadou vyhodnocovacích testov, ktoré sa zamerajú na vyhodnotenie použiteľnosti/ naučiteľnosti. Aplikujte testy na skupine používateľov. Zaznamenajte, analyzujte dáta z ich práce vo virtuálnej realite. Vyhodnoťte výsledky pomocou známych metrík.

---

[1] Vytlačiť obojstranne na jeden list papiera

[2] 150-200 slov (1200-1700 znakov), ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

Anotácia

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMACNÝCH TECHNOLÓGIÍˇ

Študijný program: Informačné systémy

Autor: Bc. Jozef Blažíček

Diplomová práca: Reprezentácia používateľa a interakcia vo virtuálnej realite s dôrazom na pocit vnorenia

Vedúci práce: Ing. Peter Drahoš, PhD.

December 2017

Zariadenia pre virtuálnu realitu sa stávajú čoraz dostupnejšími pre širokú verejnosť. S virtuálnou realitou vystupujú dva dôležité koncepty – koncept vnorenia (angl. immersion) a koncept prítomnosti (angl. presence). Prvý koncept hovorí o tom, aké množstvo zmyslových vstupov poskytuje médium, zatiaľ čo druhý koncept referuje k subjektívnej reakcii používateľa na virtuálne prostredie – pocit „bytia tam“. Keďže sa virtuálna realita stala široko populárnou len pred niekoľkými rokmi, neexistuje veľké množstvo vypracovaných štúdii ako maximalizovať oba koncepty.

Vytváranie obsahu pre virtuálnu realitu predstavuje množstvo výziev. Od samotnej reprezentácie používateľa, cez prostredie až po interakciu s virtuálnymi objektami a prostredím. Podstatný rozdiel v tvorbe obsahu pre virtuálnu realitu a klasické displeje je, že vo virtuálnej realite hrajú hlavnú úlohu faktory ako je hĺbka, zvuk a fakt že virtuálne prostredie vo virtuálnej realite zaberá všetok priestor v okolí používateľa, nie len to čo je na displeji / monitore.

Táto práca sa zameriava na reprezentáciu používateľa vo virtuálnej realite a jeho interakciu s virtuálnym prostredím, objektami a inými používateľmi s cieľom maximalizovať pocit prítomnosti.

Annotation

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: Information Systems

Author: Bc. Jozef Blažíček

Master's thesis: User representation and interaction in virtual reality focusing on presence

Supervisor: Ing. Peter Drahoš, PhD.

December 2017

Accessing devices for virtual reality becomes much and much easier and more affordable for large audience. In context of virtual reality, there are two important concepts – concept of immersion and concept of presence. Immersion is amount of sensory input provided by medium. Presence refers to psychological experience of "being there". Since virtual reality has become popular just recently, not much research has been done so far on how to achieve high levels of presence and immersion in virtual reality.

Many challenges occur when creating content for virtual reality. Starting with user representation, through environment to interaction. In virtual reality, some of key factors are depth, sound or 360-degree environment. In virtual reality, virtual environment contains all space around user, not only what is currently displayed. This factors can't be found with classical displays.

In this thesis we focus on maximizing level of presence in virtual reality by discovering user representations and interaction techniques in virtual reality.

Content

Glossary of Abbreviations iii
1. Introduction 1
2. Virtual Reality 3
2.1. Stereoscopic projection 4
2.2. History of Virtual Reality 5
3. Immersion and Presence 7
3.1. Immersion 7
3.2. Presence 7
4. Human - Computer Interaction 11
4.1. Design for Virtual Reality 11
4.1.1. User Representation 12
4.1.2. VR UI 13
4.1.3. VR Environment 15
4.1.4. Input and interaction 15
4.2. Touch feedback 16
4.3. Testing 17
4.3.1. Measuring Presence 17
5. Technologies 19
5.1. Hardware 19
5.1.1. Head- Mounted Displays 19
5.1.2. Microsoft Kinect v2 19
5.1.3. Leap Motion Orion 19
5.2. Software 20
5.2.1. Virtual Reality Modeling Language 20
5.2.2. Web-Based Graphics Library 20
5.2.3. Web VR 21
5.2.4. Game Engines 22
6. Specification 25
6.1. User Representation 25

6.2. Scenarios ... 25
6.3. Environment ... 26
6.4. Additional requirements ... 26
7. Design ... 27
7.1. User Representation ... 27
7.2. Scenarios ... 27
7.3. Environment ... 28
8. Solution ... 31
8.1. Setup ... 31
8.2. User Representation ... 31
8.3. Scenarios ... 33
9. Conclusion ... 35
10. Resume ... 37
10.1. Vnorenie a prítomnosť ... 37
10.1.1. Vnorenie ... 37
10.1.2. Prítomnosť ... 37
10.2. Interakcia človeka s počítačom ... 38
10.2.1. Dizajn pre virtuálnu realitu ... 38
10.2.2. Testovanie ... 38
10.3. Technológie ... 39
10.4. Špecifikácia ... 39
10.5. Návrh ... 40
10.6. Riešenie ... 40
10.7. Záver ... 41
References ... 43
Appendix A: Plan ... A - 1
A.1. Evaluation of 3rd semester ... A - 1
A.2. Plan for 4th semester ... A - 1
Appendix B: Source ... B - 1
B.1. CD ... B - 1

Glossary of Abbreviations

AR - Augmented Reality

CAVE - Computer Assisted Virtual Environment

FOV - Field of View

GPU - Graphics Processing Unit

HCI - Human-Computer Interaction

HMD - Head-Mounted Display

HW - Hardware

IFA - International radio exhibition in Berlin (from German: Internationale Funkausstellung).

MS - Microsoft

NUI - Natural User Interface

OC3 - Oculus Connect 3

pdd - Pixels per degree

SW - Software

SDK - Software development kit

UE4 - Unreal Engine 4

UI - User Interface

VE - Virtual Environment

VR - Virtual Reality

VRML - Virtual Reality Modeling Language

WebGL - Web-Based Graphics Library

1. Introduction

More and more users are accessing VR, because it provides unique types of experiences. VR also provides many exceptional challenges for both developers and companies. VR significantly differs from other platforms, since factors like depth or 360° view must be considered when creating content for it.

In context of virtual reality, there are two important concepts – concept of immersion and concept of presence. Immersion is amount of sensory input provided by medium. Presence refers to psychological experience of “being there”.

Some of the factors that presence and immersion depend on are user representation and interaction. Choosing it right will increase users experience and their interest of application, on the other hand wrong decisions will make users leave the application and never come back.

The purpose of this thesis is to explore user representations and interactions with objects, environment and other user in VR focusing level of presence.

This thesis is organized as follows:

1st section of this thesis focuses on Analysis (chapters two through five):

- Second chapter provides introduction to VR.
- Third chapter discuses concepts of Immersion and presence.
- Fourth chapter discuses HCI in VR. First part focuses on challenges and recommendations on designing for VR. Second part analyses importance of touch feedback. Last part describes testing techniques with focus on measuring of presence.
- Fifth chapter gives overview of available technologies. Chapter is divided into two parts: HW and SW.

2nd section (chapter six and seven) provides specification and design. Sixth chapter is divided into three parts. Each part focuses on specification and proposed solution of sub-problem. Solution design is divided into following three parts: user representation, scenarios / use-cases and VE.

3rd section (chapter eight) describes implemented solution in detail.

2. Virtual Reality

Webster’s Dictionary (1989) defines “virtual” as “being in essence or effect, but not in fact” and “reality” as “the state of quality of being real. Something that exists independently of ideas concerning it. Something that constitutes a real or actual thing as distinguished from something that is merely apparent” [1].

First definition has been used in computer science in context of virtual memory. When application need more memory (RAM) than there is available, hard disk is used to store the data a.k.a. virtual memory [1].

By simplifying second definition, we can say that reality is something that exists and can be experienced [1].

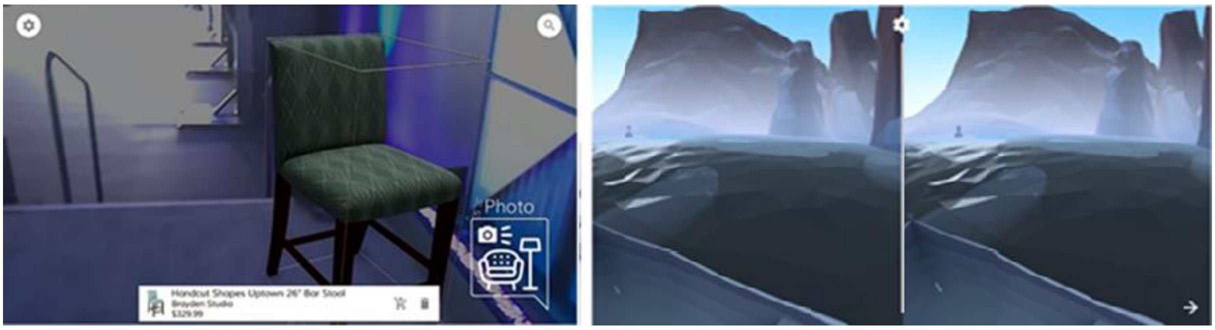

Figure 1: Examples of AR: Google Project Tango [2] (Left)
and VR: Google Cardboard Demos (Right)

Augmented reality (AR) combines input from real environment with computer generated objects (see Figure 1). VR in comparison to AR, excludes any input from real environment. Since both AR and VR use many similar concepts, many findings in AR can be applied in VR and vice versa. The main difference between VR and AR is in what environment they are using.

“A synthetic environment is an interactive space that consists of computer generated 2D or 3D images, sounds, and other content related to that space” [3]. There are three classes of synthetic environment [3]:

- Virtual environment (VE) - all sensory inputs that user experiences are generated by computer [3].
- Teleoperator environment - user controls entity that moves in real environment [3].
- Augmented environment - sensory inputs from bot real world and computer generated are received by user [3].

In order to have true VE, all of the real word inputs must be eliminated. Therefore, user receives only synthetic inputs. Achieving true VE is impossible even with current technology[3].

Nowadays devices for VR are becoming more and more accessible and affordable for large audience. Even modern cellphones are powerful enough to provide a good VR experience. Samsung's Gear VR, Google's Cardboard or Daydream VR [2] are some of the examples.

VR provides unique experience and new forms of entertainment. With VR, it's also possible to visit new or unavailable areas and places like Moon. Getting new abilities and even superpowers is also possible in VR. It enables us to do things that we couldn't do in real life [2-Design for Daydream].

VR has wide range of applications beginning from games (e.g. Sony's PlayStation VR), interactive education [4-Education and VR: Changing the way we learn], military [5] or astronaut trainings and so on.

Research provided by Google on how users use VR shows that they are mostly sitting while in VR and spend about 30 to 45 minutes in VR in one session. It can even take longer. Time that users spend in VR strongly depends on quality of VR content [2-Design for Daydream].

Designing UI for VR is challenging; it differs from design for a usual display that we have been using for years because of depth. Most of classical techniques that focuses on designing 2D content is not applicable in VR.

VR is also a technological challenge. Keeping frame rate at least 90fps on a high pixel density displays is difficult even for modern GPUs. Also, in VR, everything must be rendered twice from different angles. Once for each eye.

Two types of devices are used for VR [4-Opening Keynote]:

- large scaled displays room (often called CAVE)
- Head mounted display (HMD)

CAVE is a room where VE is displayed on walls and floor.

HMD is a headset, often used for VR or AR. In most cases, it's combination of input (containing head tracking) and output (display) devices. Contains a display of small size and high resolution.

HMD's are currently most popular types of devices for VR. Depending on device on which VR application is being processed, we can distinguish two types of HMD's [4-Opening Keynote]:

- Mobile VR – Cellphone processes VR application and displays it on its screen. Phone is inserted into VR headset.
- Computer VR – calculations are performed at computer`s GPU and sent to HMD.

2.1. Stereoscopic projection

The main challenge with VR is Stereoscopic projection. In HMD's each frame has to be rendered twice – once per eye. Both images are rendered from two slightly different angles. Classical VR rendering path is straight forward in terms of understanding but highly difficult for GPU processing (see 5.2.4.Game Engines).

2.2. History of Virtual Reality

1965 - Ivan E. Shutherland published his essay “The Ultimate Display”. In this essay, he describes the idea of “the ultimate display” as a room in which display can form any matter [6].

1968 - Sutherland build first HMD. Construction had to be mounted into the ceiling. Construction was so large that user had experienced fear of possible break down while using it. [6].

1993 - Sega presented prototype of VR HMD. This prototype has never been released.

1994 - First specification of VRML

1995 - Nintendo released VR HMD called Virtual boy.

1997 - ISO standardized VRML [7].

2012, June - Brendan Iribe and Palmer Luckey founded Oculus VR.

2012, August - Oculus started crowdfunding with Rift DK1 at Kickstarter.

2014, June - Goggle presented the concept of cheap Card Board-VR experience.

2014, July - Oculus released Rift DK2

2015, November - Release of Samsung Gear VR. It was developed in collaboration with Oculus.

2016, March - Release of Oculus Rift Consumer version

2016, April - Release of HTC VIVE.

2016, May - Google presented Daydream platform at Google i/o 2016 [2-Keynote].

2016, October - Release of Sony PlayStation VR.

2016, October - At OC3 CEO of Facebook Mark Zuckerberg presented idea of VR conference. Later co-founder of Oculus Nate Mitchel presented framework for creating web VR content [4-Opening Keynote].

2017, August – Microsoft with partners ASUS, Acer, HP and Lenovo presented Windows Mixed Reality at IFA 2017 in Berlin.

3. Immersion and Presence

Two main concepts that are being discussed with VR are immersion and presence. Both concepts have multiple definitions. Relationship between these concepts appears to be confusing, which is why they are often being misused interchangeably [6,8]. Nevertheless, they represent distinct concepts [5].

User experience in VE is a combination of endogenous and exogenous factors. Exogenous factors are divided into task and physical environment. Physical environment is composed of sensory factors which provides state of immersion and task environment is combination of cognitive factors that provides state of involvement. Endogenous factors are user's internal capabilities. User's working and long-term memory are included and interaction between them to find similar experiences to current task and physical environment. These previous experiences can affect current user's decision and experience. Optimal balance between endogenous and exogenous factors emerges sense of presence [3].

Immersion is amount of sensory input that is provided by technology [5,8], while presence refers to user's psychological experience of "being there" [6].

3.1. Immersion

Immersion is objective and measurable; it is possible to compare technologies and say which one has higher degree of immersion [5].

Immersion has multiple of components. Some of them are [5]:

- display resolution,
- display size,
- field of regard - total size of area in angles that user can see,
- FOW - size of area in angles that user can see simultaneously,
- framerate,
- head-based rendering – rendering of scenery based on head-tracking technology (rendering part of scenery on which is user looking at),
- realism of light,
- refresh rate,
- stereoscopy – technology of rendering for each of eyes, to provide depth sense,

In some cases, factors like interaction technology or realism of VE can be discussed. But it is impossible for non-realistic environment to count the degree of "realism".

3.2. Presence

Presence is a user response to exposing VE. It's psychological state and it's subjective. Various users can experience presence differently with the same VE [5].

Involvement is another factor that is often used when discussing presence. Involvement in VE is the result of focusing energy and attention towards meaningful sets of stimuli [9].

Processing information by humans is preformed both consciously and unconsciously. In consciousness we're not doing only perceptual processing - perceiving world around us. But also, we aim to do conceptual (mental) processing - reflecting on what we have previously experienced, facts, events, … Depending on conscious processing we can define absence as psychological focus on conceptual (mental) processing and presence as direct perceptual processing [9,10]. There's an open question – whether there is a threshold of allocation of attention resources in VE which has to be reached before feeling present in VE and if so what that threshold is [9].

Since we cannot eliminate all inputs from real world while in VR, state of presence is shared between real and virtual world [10]. Presence in VE depends on user's ability of shifting attention from real world to VE [9]. Therefore, we can look at "Breaks in presence" as shifts of presence from VE to the real world. "Breaks in presence" can also occur when user's attention moves towards absence [10].

Presence can be divided into three components[10]:

- Focus of attention - between presence and absence.
- Locus of attention - between virtual and physical world.
- Sensus of attention - whether user is conscious or relatively unconscious while interacting with environment.

We cannot experience presence in any environment (real or virtual) while being lost in thoughts, daydreams or fantasies [10].

We can recognize two types of presence [11]:

- Spatial presence,
- Social presence

Social presence is defined as a degree of awareness of other virtual humans. Introducing virtual humans can increase level of social immersion. Virtual humans have been used as trainers for user to teach him how to perform tasks [11]. They have also been used in exposure therapy, specifically on anxiety of public speaking. In VR, patient was exposed to virtual audience while giving speech [5,11].

Degree of social presence in game of 3D volleyball was studied on multiple types of 3D screens. Oculus Rift DK2 and CAVE reached highest levels of both spatial and social presence [11].

Other way to increase social presence is by applying concept of multiuser VR, where users interact not only with virtual humans, but with each other [4-Opening Keynote].

Steuber et. al. performed research in multi user VR. Participants performed task of rescuing in pairs. Scenarios differ on whether participants see each other's avatar and weather haptic feedback (real stretcher) was available. Also, collisions with virtual wall had been monitored. Results of the experiments have shown that humans can compensate the absence of haptic feedback if level of immersion is high [8].

For achieving presence context of data is more important than their representation and user expectations of VE than realism of VE. Photo of environment has higher level of presence than its text description. Photo contains more information than text representation. [12].

Nevertheless, it is possible for user to experience negative effect of VR. Some examples are motion sickness, disorientation, difficulties with vision or headache. These negative effects are called simulation sickness or cyber sickness [13].

4. Human - Computer Interaction

HCI is intersection of computer science and social science. HCI focuses on user-centred design. One part of HCI is usability [14].

There are many definitions of usability. We will use definition by Eric Reiss from [15]: "Usability deals with an individual's ability to accomplish specific tasks or achieve broader goals while 'using' whatever it is you are investigating, improving or designing – including services that don't' even involve a 'thing'". It means that usability depends on current situation and needs. For example, car that won't start has bad usability in case of transportation but still can be good shelter in rain [15].

There have been developed many methodologies, best-practices, advices, etc. in fields of HCI and usability for classical 2D displays. Since VR has become popular just recently, there's lack of content on how to do it in VR. But some of classical approaches are universal and can be applied for VR as well.

Usability is traditionally associated with 5 usability attributes [16]:

1. Learnability – user should be able to easy learn how to use system,
2. Efficiency – user should be able to effectively use the system,
3. Memorability – user should be able to use system after period of time without using it without need of learning it all again.
4. Errors – system should have low error-rate and enable user to fast recover from errors.
5. Satisfaction – system should satisfy users.

10 heuristic rules of usability that have been formulated by Molich and Nielsen in 1990 can be applied for VR as well [16]:

1. Simple and natural dialogue,
2. Speak the 'user's' language,
3. Minimalize the user effort,
4. Consistency,
5. Feedback,
6. Clearly marked exits,
7. Shortcuts,
8. Good error message,
9. Prevent Errors,
10. Help and documentation.

4.1. Design for Virtual Reality

There is not much content for HCI in VR. However, companies like Google or Oculus have their own research in this field and provide guidelines for developers [2,4]. In these guidelines, they present wide range of UI, interaction techniques and other advices on developing and designing for VR.

VR has unique design challenge. Users behavior and expectations are very different than on the other platforms [2- VR Design Process].

The key difference between creating content for classical devices and VR is taking depth into account. Objects that are too close (less than half meter from user's head) can make user feel uncomfortable but when something is too far (more than twenty meters from user), difference in distance is not important [2-Design for Daydream, 17-Design for virtual reality, 18].

User can move his head by 80 degrees but range of 30 degrees is comfortable for both left and right side without body movement and 15 degrees is comfortable for down movements. This numbers give boundaries of what user can comfortably see. [2-Design for Daydream, 18].

It's important to validate the design in VR as soon as possible. Things that looks good on a display can have dramatically different effect in VR [2-Design for Daydream].

Basic skills that are required with designing for VR includes [2- VR Design Process]:

- 3D Design
- Prototyping
- Interaction Design

4.1.1. User Representation

Figure 2: Oculus Avatar Studio [4-Opening Keynote]

User representation highly depends of available tracking technology. It's better to display only parts of user body that we are able to track with high fidelity. Parts of user body that are behaving unexpectedly will have highly negative effect on user`s presence. It's always better not to display parts of user body in VR than display it incorrectly. Brain is more likely to accept absence of body parts than their unexpected behavior [19-Design for VR: Environments and Interactions].

Oculus, with their avatars, displays only neck, head and hands (see Figure 2.) [4-Opening Keynote].

Depth sensing cameras like Kinect can be used for creating 3D human avatar of user. There are two possible ways to create human body avatar [20]:

- Fitting pre-defined human body avatar to size of user,
- Using raw Kinect data and applying smoothing algorithm.

Researchers at Erlangen presented their system of creating face of avatar using Kinect by fitting generic face model to Kinect's depth data and applying texture from Kinect's RGB camera [20].

Since there are known skeletal points tracked by Kinect (Kinect v1 tracks 20 skeletal points, Kinect v2 tracks up to 25 skeletal points), it's possible to pre-defined animations of body movement for avatar [20].

4.1.2. VR UI

Most of the currently available content is based on 2D. Since VR is 3D based, many of this content is not possible to apply in VR.

Po-Wei Lee et. al. created concept of cross-dimensional interface for VR called TranSection. This concept uses HMD and hand-tracking device (like LM). The main idea lies in making use of a virtual device like virtual computer for displaying 2D content inside of VR. Hand-tracking device is used for interaction with VR environment and objects. For interaction with 2D content virtual keyboard is used [21].

Menus and Icons

In most of the games, screen can be divided into two main areas. The center of the screen where actual game VE (player, other layers, environment, …) and main content is placed. The center is surrounded by icons (we will call this information area) such as health, current score or lap, current weapon, content of bag and some kinds of menu icons. While this is good for classical gaming, in VR this will not work that well. In real life, it is unnatural for object to move (hovering) as we move. Also, when user is sitting and scenery is moving, can cause in motion sickness and decreasing level of presence. [19-Design for VR: Environments and Interactions].

According to research presented by Google, users use VR mostly while sitting. Forcing users to stand can decrease their interest in application [2-Design for Daydream].

One of solutions to hovering objects can be Leap's Hovercast VR Menu. Where menu items are attached to user`s hand and displayed if needed [19-Design for VR: Environments and Interactions].

Other solution can be taken from real life. Every day we are in a situation when we are moving to another place but still we have some controlling devices and icons. Example of such situation is driving a car. While driving, driver is sitting, and environment is continuously changing but the interior of car is constant and provides helpful information (current speed). Application of this solution is possible by using cockpit- like environment (Figure 3. shows example of this type of environment) [19-Design for VR: Environments and Interactions].

Figure 3: Cockpit-like environment from Vox Machine

Colors and text

Since objects in VR can be seen from different angles, an issue of choosing colors and textures for objects and environment arises. From one angle, object can be seen but from another, it can be hard to differentiate objects [2-Design for Daydream].

With using contrast colors rises an issue of aliasing. Aliasing produces jagged edges or stair-up effect on lines or edges that should be smooth. In VR, this has negative effect on degree of immersion (real object doesn't alias) [2-Design for Daydream].

Currently most devices require text to be at least 14px height, in order for it to be readable. Comfortably readable text is about 20px height. This numbers can vary depending on font type being used. Since most of game engines are working in terms of meters and not pixels, following equation must be considered:

$$h = d \ \ tan\left(\frac{px}{ppd}\right)$$

Where h is height of text, d is distance from user, px is height of the text in pixels and ppd of device. Ppd refers to how many pixels corresponds to one degree of FOV near center of FOV [2-Design for Daydream].

4.1.3. VR Environment

VE in VR is one of the most important parts of VR application. VE in VR contains most of the space and it is a thing that is always remembered for a long time by the user. VE in VR is not a wallpaper [2-Design for Daydream].

When designing VR environments, it’s important to think about different types of users and things that scare them like heights or small spaces (claustrophobia). This can make them feel uncomfortable and make them leave the application [17-Design for virtual reality]. Other examples of things that can make users feel uncomfortable are mysterious dark hallways, partially open doors, … [2-Design for Daydream].

VR environment and VR UI should be synchronized, therefore there should not be any conflicts between them. [2-Design for Daydream].

Keeping detailed environment, with high polycount and transparency at high framerate is challenging for current GPUs. Parts of environment that are closer to user should be more detailed than ones that are far. For distances over 20 meters some wallpaper could be used [2-Design for Daydream].

Trying to maximize GPU performance at all time can increase chance of exposing user to frame drop down.

Issue which can occur with VE and UI is that it is barely possible to guarantee enough space for UI to be displayed in VR. This can happen if user is standing by the wall and UI will blend into that wall. Preventing this situation is possible if we take user to different VE while showing UI or if we attach UI with an object. Such object can be a book in a fantasy game or a cellphone in sci-fi game [22].

4.1.4. Input and interaction

Currently available interaction techniques for VR include:

1. Gaze selection /tracking (see Figure 4.) is available with any VR HMD. It tracks center of FOV. Mouse could be a good analogy to gaze tracking. Moving mouse by hand, selecting items. Moving mouse to select different items. Looking on it from physiological side – moving hand to select items. In gaze selection, head is part of body that we move to select items [22].
   Pros: available with all headsets
   Cons: selection some items can be harder due to their location and forces users to move head more often.
2. Integrated input is an additional input device built into VR headset. Examples could be Gear VR touchpad and other buttons or Cardboard magnet. Some HMDs may or may not integrate this additional input technique [19-Design for VR: Input, Planning and testing].
   Pros: Static location on headset, users know their location.
   Cons: not universal solution, depending on headset, unavailable for some headsets and each headset has different controls. Forces user to touch the headset.

3. Controllers have been used for gaming for a long time. Since many users are familiar with them, they found their way to the VR. Customer version of Oculus Rift has been originally shipped with Xbox One controller.
   Pros: widely used solution for many platforms – Xbox, PlayStation, …
   Cons: less familiar user can have problems with locations of buttons, which forces user to leave VR to find the correct button.
4. Gestures – input devices that are capable of user tracking. Whether it is full body tracking or just some parts capable of being recognized. Examples of such devices are Microsoft's Kinect or LM [19-Design for VR: Input, Planning and testing].
   Hand gesture recognition and motion tracking systems provide natural user interface. [21].
   Pros: NUI
   Cons: requires additional device or larger space
5. Combined – controllers like HTC Vive Controller or Oculus Touch Controller combines controller input with hand-tracking techniques.
   Pros: precise, offers hardware buttons and vibration feedback feedback
   Cons: platform dependent.

Figure 4: Gaze tracking in Google's Cardboard VR

4.2. Touch feedback

On the one side Gestures provides NUI, but on the other side, user in VR loses touch feedback. To understand how important touch feedback is, we can look at two known cases of patients that lose most of touch sense. By analyzing these cases we can say that somesthesis information are important for accurate and fast interaction. In VR this can be partially compensated by high amount of other sensory inputs [23].

4.3. Testing

Testing can be performed in parallel to development. Development doesn't need to stop while testing. Just between 4 and 5 testers can find about 80% of usability problems [14].

Usability test can be divided into three parts:

1. Planning test
2. Testing
3. Evaluating results

Testing starts with planning. Selecting goals and tasks. Every possible task that user can perform with system cannot be tested. It's good to focus on tasks that can have potential problems, users will do with system or other important criteria for system [14].

During test, everything should be monitored and captured. At least two cameras – one focusing on screen, another on tester. Audio record from testing is important as well [14].

While performing test, nobody from the development team can be present [14].

Users are more likely to give useful feedback on demos that are at early stages of development than on the ones which are at 90-percent completion [2-VR Design Process].

4.3.1. Measuring Presence

Since immersion is an objective measure and can be evaluated by comparison of two or more VEs / VR setups, there's no need for user or other types of testing.

In terms of measuring presence there are many applicable approaches. We can divide them into three main groups:

1. Subjective – questionnaires or interviews with subjects of testing, … [9,14]
2. Behavioral – task performance, observation, …
3. Physiological – hearth rate, …

Most of presence questionnaires are specific for concrete setup and VE. Only few of them could be categorized as generalizable (some of such questionnaires are [9] or [24]) [3].

It's important to note that in case of subjective testing situations when someone's subjective judgement is in direct conflict with their physiological responses can occur. Example of such situation can be when user says that environment is unrealistic but still avoids moving virtual objects [10].

Witmer et al. views presence as function of individual differences (user's individual abilities, experiences, …) and characteristics of VE.

Witmer's questionnaire from 1995 was based on four presence factors [9]:

1. Control factors
2. Sensory Factors
3. Distraction factors
4. Realism factors

5. Technologies

Nowadays, innovative technologies are developed in rapid speed. VR is a great example of it. At time of starting this thesis, there were available only two computer-based headsets: from Oculus and HTC, less than year later with Microsoft Mixed Reality platform came new Headsets from companies like Acer, Asus, Lenovo or HP.

5.1. Hardware

5.1.1. Head- Mounted Displays

HMD is a display device. Construction of the device must guarantee that display is always in front of user's eyes regardless of user's head movement. Some of the examples for such devices are Oculus Rift or Samsung Gear VR.

From hardware perspective we could divide currently available computer-based HMD into two categories:

- Oculus & HTC – both supports outside-in tracking, offers OLED screen with resolution of 2160 x 1200 px (for both eyes) with refresh rate at 90Hz and FOV of 110 degrees.
- Microsoft Mixed Reality – uses inside-out tracking, LCD (or OLED in HMD made by Samsung) screen with resolution 2 880 x 1 440 px with refresh rate at 90Hz and FOV varies between 95 and 110 degrees depending on concrete HMD.

5.1.2. Microsoft Kinect v2

Microsoft's Kinect is device capable of recognizing and tracking people's motions [25]. Kinect allows multiple types of tracking [25-Kinect for Windows Programming Guide]:

- Lean tracing
- Body tracking (25 skeleton point (see figure 5.) up to six people [25-Features]).
- Face tracking

Microsoft with Kinect SDK provides supporting libraries for C# and C++ [25].

5.1.3. Leap Motion Orion

Leap Motion Orion (LM) is hand-tracking device. It is capable of tracking user's fingers. The origin of axes is in the middle of the device. It is able to track [26]:

- Hands
- Arms
- Fingers

LM provides wider range of language support compared to Microsoft: C#, C++, Objective-C, Java, Python and JavaScript [26].

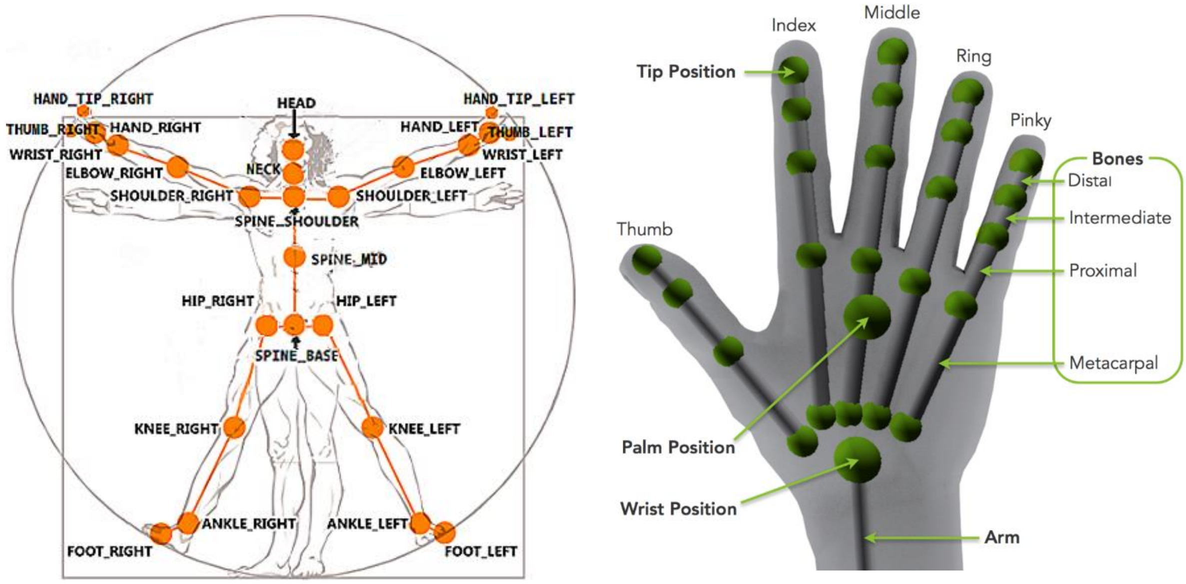

Figure 5: Kinect skeletal points (left) and LM hand (right)

Kinect provides full body tracking while LM focuses only on hands. Kinect data offers possibility of interaction with whole body not just one part. On the other side, LM offers more precise data for hands than Kinect does. This is important for better hand interaction in VR – taking objects, ...

5.2. Software

5.2.1. Virtual Reality Modeling Language

VRML is text language, specified in 1994. It allows bringing 3D content into browser and sharing it via internet. VRML allows building 3D objects from 3D shapes, lightning sources, fog animations and sound effects [7].

VRML browser allows interaction with scene and objects [7].

It reads text file and creates scenery. For actual driving are used graphics toolkit like OpenGL or Microsoft DirectX [7].

VRML data can be stored in multiple files [7].

Objects are composed into scene graph. Scene graph is a hierarchy arranged in tree where leaf can be shape, light, sound, … Unlike in regular tree, in VRML each node within hierarchy can have more than one ancestors, which allows objects to be used multiple times in scenery [7].

Nowadays VRML is hardly used but some web browsers still support it. Currently WebGL is often used.

5.2.2. Web-Based Graphics Library

WebGL is developed by Khronos Group. Same group that also govern OpenGL. WebGL brings 3D content into webpage. WebGL is [27]:

- JavaScript API,
- based on OpenGL ES 2.0 (WebGL 1.0, WebGL 2.0 is based on OpenGL ES 3.0),
- free to use,
- cross platform.

Firstly, it is needed to specify where the canvas will be placed in 2D webpage or where pen starts. WebGL then uses 3D coordinate system. The rest of the work is similar to creating OpenGL ES content [27].

For creating Shaders the same version of OpenGL shading language is used as OpenGL ES (GLSL ES). WebGL requires defining shaders [27].

Three.js is library for WebGL contains most commonly used objects in 3D graphics. As well, it contains many useful utilities. Furthermore, among properties of three.js are [27]:

- is fast,
- includes math,
- objective-oriented,
- use of HTML5 2D canvas,
- hides 3D rendering details.

WebGL is currently supported by all popular web browsers – Google Chrome, Mozilla Firefox, Microsoft Internet Explorer and Edge, Safari, Opera, … [2- Enhancing Applications and Websites with Embeddable VR Views]

5.2.3. Web VR

Web VR combines VR and web technologies. It allows VR developers to reach audience outside of VR because content is available either for both VR and non-VR users. Web VR in web browser mostly uses WebGL.

Web VR also allows cooperation between VR and non-VR users - on PC via web browser (see figure 6.) [4-WebVR Create Portable VR Experiences on the Web].

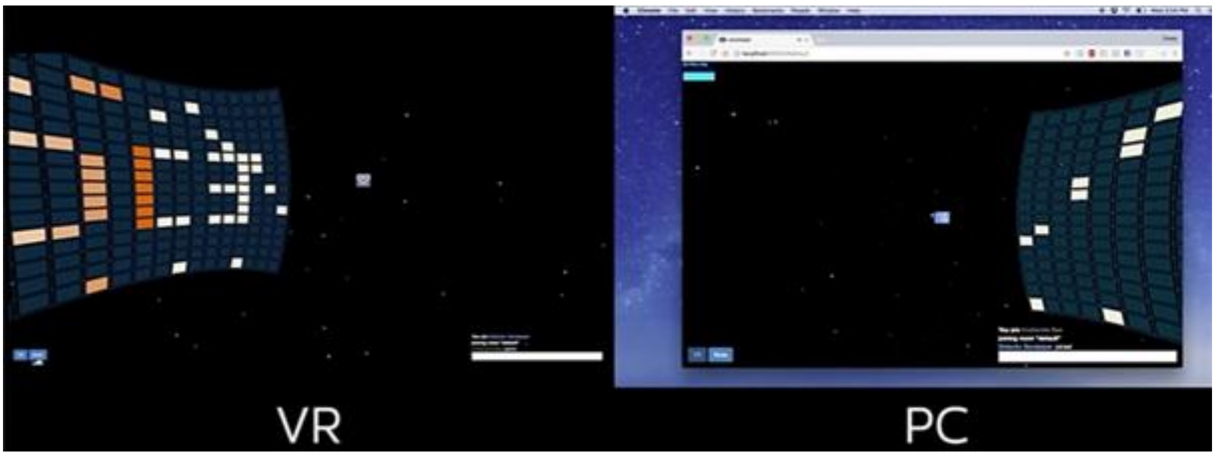

Figure 6: Collaboration in Web VR between VR (left) and non-VR (right) users - Oculoops demo [4-WebVR Create Portable VR Experiences on the Web]

Google's Web VR demo called Chrome Experience is available at g.co/chromevr. This demo contains examples of Web VR game, 360 Video, … accessible via web browser.

5.2.4. Game Engines

As name suggests, primary target of Game Engines is to provide supporting tools for game-development. With VR, they are not only used for creating game content but for any kind of VR content.

UE4 allows developers to edit VE in VR via VR Editor. This feature is meant to help developers with beginning phases of development and quickly validate initial design. Furthermore, it is used when finalizing the application [4-What's New in Unreal VR].

Some of UE4 VR features include [4-What's New in Unreal VR]:

- Touch support – abstracted motion controller – creating interaction once for both Oculus and HTC touch controllers.
- Real-time GPU profiling – allows identifying GPU costs.
- UI Support – 3D UI widgets (see figure 7.).
- Instant Stereo Rendering - classical rendering: draw for left eye, draw for right eye. Instant Stereo Rendering enables driving objects for both eyes with single draw call.

Figure 7: VR widgets in UE4 [4-What's New in Unreal VR]

UE4 offers two ways of creating content (see Figure 8):

- Code written in C++
- Blueprint editor

Blueprints Visual Scripting in UE4 is a scripting system. It is graph-like based. Blueprint nodes represent function calls. Edges from right sides of node represent output of node while on the left side are inputs. Edges connects only right side of one node with left side of another node. There are two main edges:

- Control flow – these edges defines order of function calls
- Data flow

For both ways of creating content UE4 offers many base-classes: for vehicle movement, character, animation, … Code written in C++ can be executed in Blueprint, and not the other way.

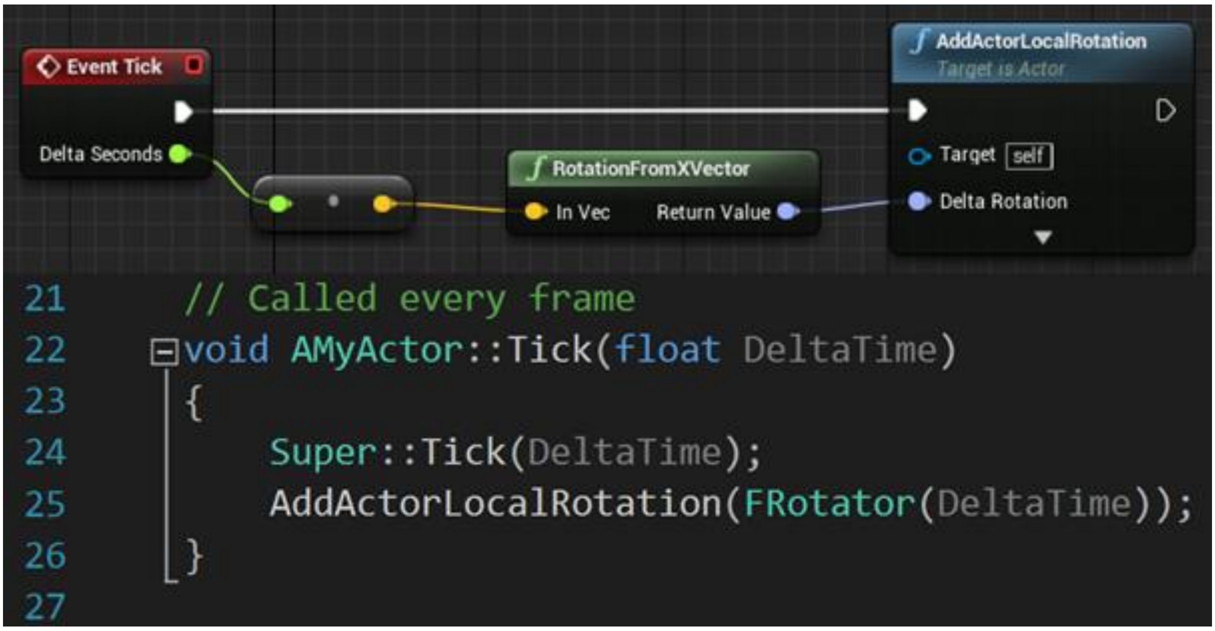

Figure 8: UE4 Blueprint (up) and same functionality in C++ (down)

Both UE4 and Unity provide useful features for developers. The main difference between them is supported programming language. UE4 works with C++ code while Unity uses C#.

6. Specification

The goal of this thesis is to develop dynamic user representation and interaction in VR with the focus put on level of presence. We divided this goal into three main parts:

- User representation
- Scenarios /use-cases
- VE

According to main goal and knowledge we gained from the analysis of the domain, we summarized requirements for each part of our goal.

6.1. User Representation

Firstly, we need to get user into VE – create user representation in VE. User needs to be able to interact with VE that surrounds him/her and perform tasks. This representation has to be created using contactless technologies (e.g Kinect or LM).

To achieve higher presence, user representation needs to be reliable.

6.2. Scenarios

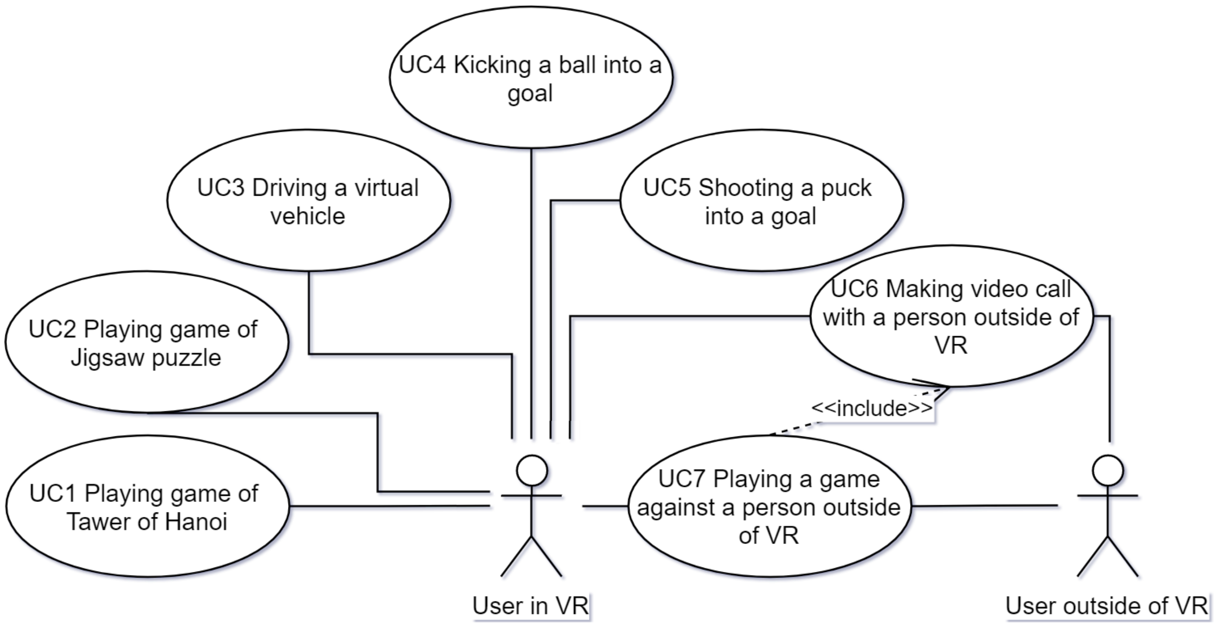

Figure 9: UML use-case diagram

Since presence is subjective, for its evaluation we need to be able to perform presence testing on our scenarios / use-cases. Specifically, presence questionnaire. Therefore, for our purposes we have chosen following seven use-cases (see Figure 9):

- UC1 Playing game of Tower of Hanoi,
- UC2 Playing game of Jigsaw puzzle,

- UC3 Driving a virtual vehicle,
- UC4 Shooting puck into goal,
- UC5 Kicking ball into goal,
- UC6 Video calling with a person outside of VR,
- UC7 Playing simple game with a person outside of VR.

First two use-cases (UC1 and UC2) focus on exposing user longer-term activity in VR. Since solving puzzle games takes time.

Next three use-cases (UC3, UC4 and UC5) focus on full body representation of user. Last two of them (UC4 and UC5) additionally focus on using other parts of body than hands. Activities like kicking ball, accelerating or stopping the vehicle requires usage of legs.

Also with first five use cases we can measure precision.

Last two use-cases (UC6 and UC7) focus on interaction and/or collaboration between users in VR and outside of VR.

With all use-cases we will also measure accuracy and response time for actions.

6.3. Environment

Environment requirements highly depend on choice of use-cases. Environment for each use-case (or multiple use-cases) needs to be in sync with user’s expectations for that task. Environment also needs to be physically based.

6.4. Additional requirements

Additionally, user needs to be able to switch between use-cases without leaving VR. Since user will not have any controller available, solution needs to include intuitive, simple gestures. For example, gesture that brings controls to switch location on settings of game.

7. Design

7.1. User Representation

For user representation we will use data provided from Kinect v2 and LM sensors.

UE4 offers multiple user avatars bodies. Bones (parts) of these bodies have hierarchical structure, which means axis orientation depends on rotation and location of parent body part.

Kinect for Windows SDK offers multiple information about tracked body, from which data about joints locations and orientations seems to be most adequate for user representation. Both UE4 and Kinect offer similarly constructed bones hierarchies.

Kinect offers only limited information about user hand; location of thumb, hand and if user`s hand is opened, closed or makes lasso sign (middle and index fingers opened, others closed). Therefore, using LM for better precision in hand interaction is more adequate.

For graphical user representation we have chosen two models:

- Joint Skeleton – tracked points (spheres) connected by lines (cylinders),
- Body model – animate human avatar provided by UE4 according to sensor data.

7.2. Scenarios

According to rules of Tower of Hanoi we will not allow user to put larger disk on smaller is such attempt, larger disk will spawn at original position. Also, all disks that aren't on top will be locked on their positions until disks above them are moved away. Around both centers of disks and sticks we will put collision components that will check attempts on allowed or banned moves. For each stick we will remember which disks are currently placed there (in an array).

For Jigsaw puzzle we need to create models for at least 4 puzzle pieces: one edge, two corners and one center. Depending on final location of concrete piece we will modify rotation of model and location of texture coordinates. At connecting points, we put collision boxes to check if pieces match.

For both puzzle games we put borders around play area, so when moving object leaves borders, moving object will be automatically spawn at last position. This will prevent cases when user misses the piece and is unable to find it later, which would possibly result in unsolvable puzzle.

To take some object we need to touch that object at least at two opposite areas and then apply some pressure. For moving parts in both puzzle games, we put collision areas around them. While two opposite collision areas detect two different parts of user avatar (e.g.: two fingers), this moving part will be attached to them.

Football and hockey have similarities: both have goals, object that players are trying to place in goal (ball vs. puck), rectangular game area, … These similar parts can be extracted to the parent class and in subclasses for concrete game we will set properties for them. Other similar parts and functionality includes: object spawning, goal detection or detection if object leaves the play area.

UE4 provides vehicle templates, which contain implementation of physics, movement, … We will use this as base for our vehicle. In our case, user will control vehicle with hands (steering wheel and gear) and legs (gas, clutch and break), therefore we will implement our own controlling system and map it to inputs of template. In reality, holding steering wheel with two fingers is not enough. For moving steering wheel, we will require at least thumb and other three fingers to be involved. In case that user holds the steering wheel, rotation of wheel will be average of both hands as long as difference is less than specified threshold. For cases when user would try to move hands in opposite directions. In this case, no action will take place and wheel will return to its default position. Since user will sit inside the vehicle, model needs to have highly detailed interior.

Videocall will be performed via virtual computer, laptop or tablet pc. Image received by other side (non-VR user), will be the image from additional camera placed on device. Acceptance, refusal or ending of call will be performed via pressing 2D buttons on ("touch") screen of virtual device.

7.3. Environment

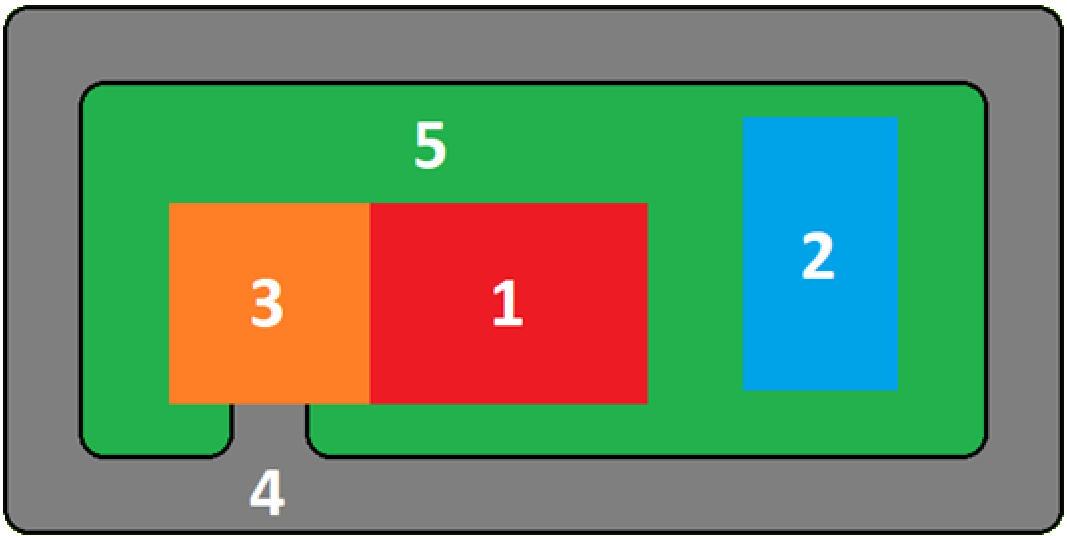

Figure 10: Environment Footprint: 1-Main Room, 2-Hockey/Football ground, 3-Garage, 4-Road, 5-Garden

Kinect v2 can perform good quality tracking of body parts only if user is facing the sensor or slightly turns to side. Thus, we need to focus user attention in VE on specific direction. Therefore, we put more objects in front of user and just a few of them behind.

For achieving high degree of presence, VE needs to be classified by user's brain as possible. For our environment, we take inspiration from real world:

Playing board games, making videocall or some online collaborative work is mostly indoor activity. First part of VE (see Figure 10 - 1) will be some VR room (for UC1, UC2, UC6, UC7).

Hockey and football are games played in large inside /outside stadiums. In smaller scale football can be played on open area with defined goals or hockey is sometimes played on frozen lakes. Second part of our VE will be garden (see Figure 10 - 5) with football /hockey area (see Figure 10 - 2). Based on use-case we can either switch between them or divide the whole area into two parts. Each will represent one game and depending on scenario we make user face the chosen half.

For driving vehicle, we need some road. Last part of VE will be road and garage. For starting position, user will be placed in car inside of a garage (see Figure 10 - 3). Garage and VR room will be part of the house (see Figure 10 – 1 & 3). Road (see Figure 10 - 4), on which user will be driving car will create outside border around house and garden.

For user, to be able to switch between parts of environment (locations), we could use two approaches:

1. Attach buttons for choosing next location to user’s hand (like uses LM in their demo).
2. Put buttons on virtual map of VE whether it’s virtual tablet PC or paper.

In both cases we could additionally extend selection to use-case specific settings like saving, loading or restarting game (e.g. in UC1 or UC2).

We could make an animation of user after he/she choses other location, but this approach could cause motion sickness since in VR user will move bur physically will stay at the same place. Other option is to spawn / teleport user to new location.

8. Solution

We implemented our solution using UE4 and MS Visual Studio 2017. In implementation we used both C++ and UE4 Blueprints. In order to access data from Kinect v2 and LM sensor, we used Kinect for windows C++ SDK (not UE4 plugin) and Leap SDK.

8.1. Setup

Used setup contains four main parts:

- Computer (Lenovo laptop),
- Oculus Rift DK2 HMD,
- MS Kinect v2 Sensor and
- LM Sensor mounted on HMD.

For this setup and project to work properly is necessary to make sure that following conditions are met:

1. LM controller is mounted on HMD in way, that light indicator is facing downwards.
2. The distance between Kinect and user is more than minimal distance and less than 4.5 meters.
3. Before starting application (or computer), HMD is facing Kinect sensor.

Since we know that vertical FOW for Kinect is 60° [25] we can create formula to calculate minimal distance from the sensor:

In most cases, Kinect sensor is not placed at height that is equal to half of user's height. Therefore, we need to calculate distances for highest and lowest points of user`s body. Minimal distance from Kinect is then maximal value from both distances. It can be expressed mathematically in following formula:

$$distance(h) = \frac{h * \cos 30°}{\sin 30°} = h * \cot 30°$$

$$minimal\_distance(h_k, h_u) = \max\left(distance(h_u - h_k), distance(h_k) \right)$$

Where h_k is height at which is Kinect placed and h_u is height of user.

To be more precise with calculation of minimal distance, we should also include in our calculations horizontal FOW (which for Kinect v2 is approximately 70°) and maximal width of user. Distances could be calculated in a similar way.

8.2. User Representation

For user representation we used data from both Kinect and LM sensors. We implemented two models of user representation (see Figure 11):

1. Joint skeleton – created using spheres and cylinders

2. Body model – using models of body parts.

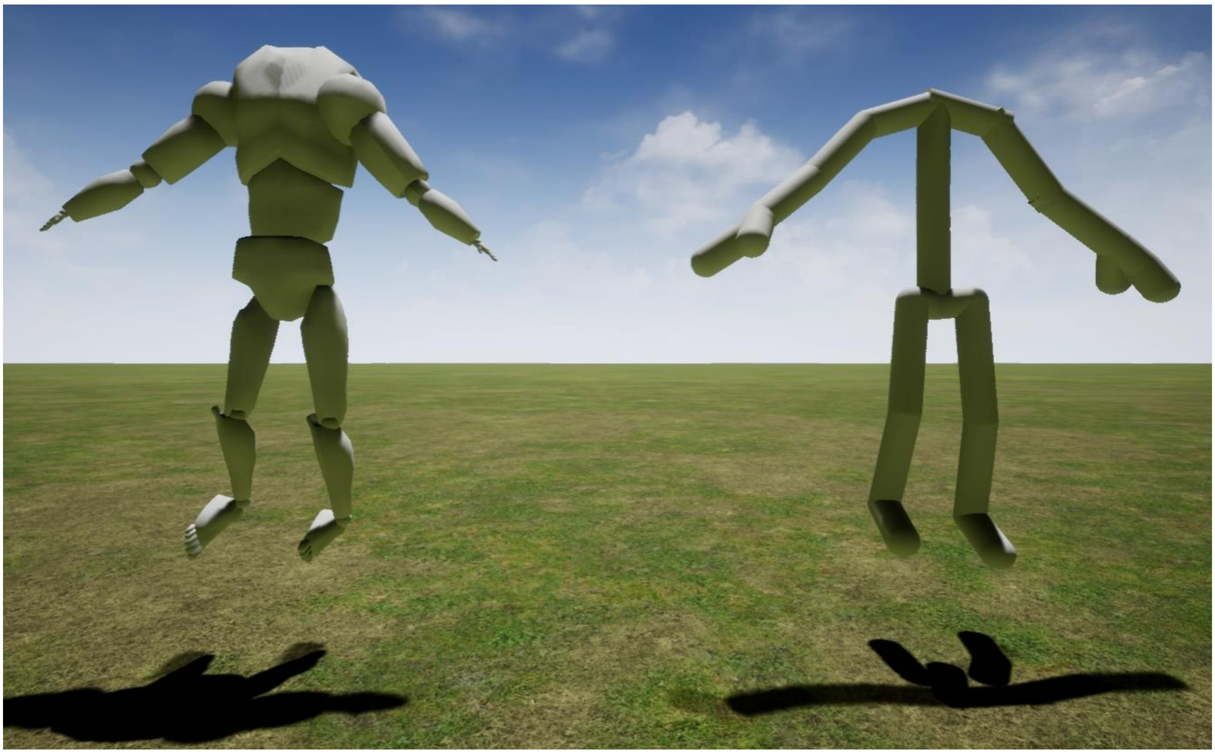

Figure 11: User representations, joint skeleton (right) and body model (left)

Since all three of needed technologies use different orientation of axis, first step was to convert them to UE4 space.

In Kinect case we're doing simple mapping:

$$Location_{UE4}(x, y, z) = Location_{Kinect}(-z, x, y) * 100$$

Multiplication by 100 is needed because Kinect location data are in meters but UE4 uses centimeters.

Since LM is physically mounted on HMD and its orientation depends on what user is currently looking at, application of camera rotation was necessary. For both hands we created additional hierarchy. Each hand component is attached to hand root component (additionally hand root component is attached to body root component). Location and rotation of hand components is set relatively. Than hand root is rotated according to camera rotation and moved to hand global position. This results in additional step to be performed only once per hand. Final location of hand is calculated according to both LM and Kinect data.

Using location of points give us one more useful information – distance between these points. Since we know height of body part model and real distance between its ends, we can calculate scale of each of body parts. This makes both user representations scalable depending on height of user.

In joint skeleton representation we placed spheres on locations retrieved from sensors (Kinect joint location). Then for each of pre-defined pairs of joints we calculated distance between them and scaled cylinder accordingly along z-axis. Then we placed this cylinder in the center between them and calculated location that cylinder connects both spheres.

For most of body model parts we did similar approach: we placed body part on according location from sensor data and calculated and set rotation between both endpoints of body part. For z-scale we used ration between distance of endpoints and height of body part, for x and y scale we used average z-ration.

Data update request to both Kinect and LM sensor is done for every frame. If new data are available, we update current user representation. Connection to devices and first data processing is handled in separate classes for each device - Kinect and LM.

Switching between user representations at runtime is possible by pressing SHIFT+Q keys.

8.3. Scenarios

As a model for our vehicle for UC3 we used BMW X5 (see Figure 12) from: https://free3d.com/3d-model/bmw-x5-1542.html.

Figure 12: Vehicle model of BMW X5 used for UC3

9. Conclusion

All currently available solutions require controller. We believe that using technologies which provide NUI can increase user’s level of presence since user is familiar with this kind of interaction for whole life. Question that rises is whether current devices for NUI are good enough to produce state of presence.

For our testing we created user representation based on data from Kinect and LM sensors. Limitations of this representation are given by limitations of both technologies – distance, precision, refresh rate, ...

To validate effect of this technology on presence we chose seven use-cases. Use-cases on following factors: focuses on long-term usage of technology, precision of technology, interaction with different body- parts, performance, collaboration between two users (outside and inside of VR) and their effect on presence.

As base for our presence questionnaire we will use questionnaire developed by Witmer et al. extended by our own questions.

10. Resume

Virtuálna realita sa stáva čoraz populárnejšou a rozšírenejšou. Počet používateľov a zariadení sa neustále zvyšuje [2-Keynote, 4-opening keynote.

V mnohých oblastiach je to jedinečné médium. Poskytuje unikátny typ zážitku a zábavy.

Medzi aplikácie virtuálnej reality patria rôzne druhy tréningov (armádny [5]), interaktívna výučba [4-Education and VR: Changing the way we learn] či zábavný priemysel (Sony PlayStation VR).

V súčasnosti najviac rozšírenými zariadeniami pre virtuálnu realitu sú náhlavné displeje (angl. head-mounted display - HMD).

Virtuálna realita rovnako predstavuje množstvo výziev. Od hardvérových – udržiavať za každých okolností počet snímok za sekundu na vysokej úrovni, po softvér – dizajn aplikácii, interakcie, reprezentácia používateľa, ...

Nakoľko virtuálna realita nabrala na popularite len nedávno, metodológie na tvorbu obsahu pre virtuálnu realitu buď vôbec neexistujú alebo sú vo veľmi skorých štádiách. Väčšina výskumu doteraz bola vykonávaná zväčša len na akademickej pôde v malom rozsahu. Aj keď spoločnosti ako Google [2,17], Oculus [4,22] či Microsoft [25] poskytujú návody na tvorbu obsahu vo virtuálnej realite, tieto návody nie sú zďaleka na takej úrovni ako pre klasické displeje.

S virtuálnou realitou vystupujú dva dôležité koncepty – koncept vnorenia (angl. immersion) a koncept prítomnosti (angl. presence) [5,6].

10.1. Vnorenie a prítomnosť

10.1.1. Vnorenie

Koncept vnorenia hovorí objektívne o tom, aké množstvo zmyslových vstupov poskytuje médium. Je ale možné porovnať rôzne média a povedať, ktoré poskytuje väčšiu úroveň vnorenia. Úroveň vnorenia ovplyvňujú hardvérové aj softvérové vlastnosti [5]:

- Rozlíšenie displeja,
- Veľkosť displeja,
- Stereoskopia,
- Zorné pole (FOV)
- Počet snímkov za sekundu (angl. framerate)
- ...

10.1.2. Prítomnosť

Koncept prítomnosti referuje k subjektívnej reakcii používateľa na virtuálne prostredie - pocit „bytia tam“ [6]. Na rozdiel od vnorenia je prítomnosť subjektívna a rôzni používatelia môžu mať rozdielnu úroveň prítomnosti v tom istom virtuálnom prostredí [5].

10.2. Interakcia človeka s počítačom

Interakcia človeka s počítačom (angl. human-computer interaction - HCI) je disciplína, ktorá spája informatiku a spoločenské vedy. Zameriava sa na dizajn zameraný na používateľa. Jednou z častí je použiteľnosť (angl. usability).

Použiteľnosť sa zameriava na to, ako jednoducho a rýchlo dokážu používatelia systému dosiahnuť svoje virtuálne ciele [14].

10.2.1. Dizajn pre virtuálnu realitu

Veci ako správanie používateľa či jeho očakávania sú vo virtuálnej realite odlišné od klasických prístupov [2- VR Design Process].

Základný rozdiel v tvorbe dizajnu pre virtuálnu realitu oproti klasickým displejom je v tom, že hlavnú úlohu vo virtuálnej realite tvorí hĺbka (tretí rozmer). Pri tvorbe dizajnu je dôležité brať do úvahy fakt, že používateľ môže vidieť objekty z rôznych uhlov a vzdialeností. [2-Design for Daydream, 17-Design for virtual reality, 18].

Reprezentácia používateľa vo virtuálnej realite záleží od dostupných sledovacích technológii. Je lepšie zobrazovať iba časti tela, ktoré sme schopní sledovať s vysokou presnosťou (pozri Figure 2.) [19-Design for VR: Environments and Interactions].

Virtuálne prostredie nie je pozadie. Zaberá väčšinu zorného poľa používateľa. Viac detailné prostredie by malo byť v blízkosti používateľa. Pri objektoch , ktoré sú vzdialené viac ako 20m, nevie používateľ značne rozoznať rozdiely vo vzdialenosti. [2 Design for Daydream].

Interakcia vo virtuálnej realite závisí od dostupných vstupných zariadení.

Sledovanie pohľadu (angl. gaze tracking) je jediným spôsobom interakcie, ktorý je natívne prítomný vo všetkých náhlavných displejoch pre virtuálnu realitu (pozri Figure 5.). Analógiou k tejto technológii môže byť myš – presunom myši po podložke sa pohybuje kurzor na obrazovke. Otáčaním a kývaním hlavy sa pohybuje kurzor vo virtuálnej realite.

Ďalším spôsobom interakcie môže byť rozpoznávanie pohybu končatín (angl. Gestures). Pre tento spôsob interakcie je potrebné zariadenie schopné sledovania a rozpoznávania pohybov končatín. Príkladom takýchto zariadení môže byť Kinect od Microsoftu alebo LM.

10.2.2. Testovanie

Testovanie môže byť vykonávané súčasne s vývojom. 80% chýb v použiteľnosti sa spravidla odhalí pri testovaní 4 až 5 osobami [14].

Vyhodnotenie vnorenia je možné vykonať troma spôsobmi:

- Subjektívny – dotazník, rozhovor, ...
- Behaviorálny – sledovanie, ...
- Fyziologický – meranie tepu, ...

Väčšina dotazníkov je robená na mieru konkrétneho scenára a testovania. Pri subjektívnom testovaní môže nastať stav keď subjektívny názor používateľa je v priamom rozpore z jeho fyziologickými odozvami – napr. ak tvrdí že prostredie je nerealistické ale vyhýba sa letiacim virtuálnym objektom [10].

10.3. Technológie

Game engine je softvér poskytujúci paletu podporných nástrojov pre tvorbu počítačovej grafiky a hier pre rôzne platformy. V súčasnosti sa využívajú pri tvorbe obsahu pre virtuálnu realitu (nie len hier).

Unreal Engine 4 okrem iného poskytuje editor na tvorbu obsahu pre virtuálnu realitu vo virtuálnej realite. Tento inštrument je primárne určený pre počiatočné a koncové fázy projektu. Na začiatku je dobré čo najrýchlejšie validovať dizajn na ktorom sa pracuje vo virtuálnej realite a v koncovej fáze doladiť detaily. Hlavná časť vývoja – programovanie nie je vhodné tvoriť vo virtuálnej realite [4-What's New in Unreal VR]:

Leap Motion (LM) je zariadenie schopné sledovania a rozpoznávania pohybu rúk [26].

MS Kinect v2 je schopný rozpoznať a sledovať používateľov (angl. body tracking). Sleduje 25 bodov a až 6 používateľov naraz. Sledovanie tváre (angl. face tracking) sa uskutočnuje vo vysokom rozlíšení [25].

10.4. Špecifikácia

Naším cieľom vytvorenie dynamickej reprezentácie používateľa, a interakcie s použitím zariadení čo sprostredkujú prirodzené používateľské rozhranie so zameraním na pocit prítomnosti vo virtuálnom prostredí. Tento cieľ sme rozdelili na tri hlavné časti: Reprezentácia používateľa, scenáre/prípady použitia a Virtuálne prostredie.

Reprezentácia používateľa musí byť vytvorená s použitím bezkontaktných zariadení a byť dôveryhodná. Používateľ musí byť schopný interakcie s virtuálnymi objektmi.

Zvolili sme sedem interaktívnych scenárov (pozri Figure 9):

- UC1 Hranie Hanojských veží,
- UC2 Skladanie puzzle,
- UC3 Riadenie virtuálneho automobilu
- UC4 Kopanie lopty na bránu
- UC5 Triafanie puku do brány
- UC6 Video hovor s osobou mimo virtuálnej reality
- UC7 Kolaborácia / hranie hry s osobou mimo virtuálnej reality

Prvé dva (UC2, UC2) scenáre sa zameriavajú dlhodobé používanie virtuálnej reality (riešenie puzzle trvá nejaký čas). Nasledujúce tri (UC3, UC4, UC5) sa orientujú na reprezentáciu používateľa. Riadenie automobilu na interakciu rukami a nohami, kopanie lopty iba na interakciu nohami a strieľanie puku iba na interakciu rukami. Posledné dva (UC6, UC7) sa orientujú na kolaboráciu v a mimo virtuálnej reality. Vo všetkých prípadoch sledujeme aj presnosť a odozvu použitej metódy (NUI).

Prostredie pre daný scenár musí byť v súlade s očakávaniami používateľa na plnenie danej úlohy.

Medzi ďalšie požiadavky patrí možnosť používateľa meniť scenáre bez nutnosti opustenie virtuálnej reality. Nakoľko používateľ nebude mať dostupný ovládač, je potrebné aby riešenie implementovalo gestá na ovládanie / zmenu nastavení.

10.5. Návrh

UE4 poskytuje niekoľko humanoidných modelov. Pre reprezentáciu používateľa použijeme jeden z týchto modelov a na základe dát z Kinectu a LM nastavíme jednotlivé časti modelu.

Pre interakciu s predmetmi je potrebné aby používateľ držal daný predmet aspoň dvoma časťami ruky na protiľahlých stranách. Okolo každého pohyblivého objektu umiestnime kolízne komponenty, ak dva protiľahlé komponenty zdetegujú používateľa, objektu bude „pripojený“ k používateľovi. V prípade volantu automobilu sa budú vyžadovať aspoň 4 prsty jednej ruky aby sa ho dotýkali.

Virtuálny video hovor sa bude konať prostredníctvom virtuálneho počítača. Začatie, ukončenie hovoru bude používateľ vykonávať prostredníctvom 2D tlačidiel na virtuálnom počítači (dotyková obrazovka).

Pre virtuálne prostredie sme zvolili virtuálny dom so záhradou, garážou a izbou a cestu okolo domu so záhradou (pozri Figure 10). Hranie stolových hier (hanojské veže a puzzle) a online kolaborácia sú vnútorné aktivity – budú sa nachádzať v spoločnej virtuálnej izbe. Kopanie na bránu alebo strieľanie puku do brány sú vonkajšie aktivity – umiestnené v záhrade. Automobil vyžaduje cestu po ktorej ho bude používateľ riadiť a ako začiatočný bod sme zvolili garáž.

10.6. Riešenie

Na reprezentáciu používateľa (pozri Figure 11) sme použili SDK poskytované Microsoftom a Leap-om. Pred aktualizovaním aktuálnej snímky pošleme požiadavku na aktualizáciu polohy do oboch zariadení. Ak nové údaje sú dostupné, aktualizujeme reprezentáciu používateľa na základe nových dát. Nakoľko všetky tri technológie: UE4, Kinect a LM používajú iný súradnicový systém, priestorové dáta najskôr konvertujeme do UE4 priestoru. Časti modelu sme namapovali na body, ktoré sleduje Kinect alebo LM. Na základe konvertovanej polohy týchto bodov nastavíme polohu danej časti teľa. Z vopred stanovených dvojíc (dvojice bodov, ktoré spájajú konce danej časti modelu napr. koleno a začiatok chodidla) vypočítame a nastavíme natočenie. V prípade ruky sme vytvorili hierarchiu kde časti ruky sú deťmi centrálneho komponentu ruky. Poloha častí ruky sa nastaví lokálne vzhľadom na centrálny komponent. Poloha centrálneho komponentu ruky sa určí na základe dát z oboch senzorov. Korekcia natočenia HMD sa aplikuje iba na centrálny komponent.

Ako model automobilu sme použili model BMW X5 (pozri Figure 12).

10.7. Záver

Všetky súčasne dostupné riešenia pre virtuálnu realitu vyžadujú nejaký ovládač. Veríme že použitie bezkontaktnej technológie alebo technológii poskytujúcich prirodzené používateľské rozhranie môže poskytnúť používateľovi jedinečný zážitok z virtuálnej reality.

Na reprezentáciu používateľa sme použili dáta z Kinectu a LM.

Na otestovanie reprezentácie používateľa použijeme sedem scenárov. Scenáre majú za účel otestovať niekoľko faktorov vzhľadom na pocit „bytia tam“. Faktory, ktoré scenáre sledujú sú: dlhodobé používanie technológie, presnosť interakcie s jednotlivými časťami, interakcia medzi používateľmi.

Ako základ pre náš dotazník sme použili dotazník, ktorí vytvorili Witmer a spol.

References

[1] W. R. Sherman a A. B. CrAig, Understanding Virtual Reality: Interface, Application, and Design, Elsevier, 2002.

[2] Google, „Google I/O 2016," Mountain View, California, USA, 18.-20. May 2016.

[3] C. Dustin, Exploring Additional Factors of Presence (Dissertation Thesis), Department of Industrial Engineering and Management Systems in the College of Engineering and Computer Science at the University of Central Florida, Orlando, Florida, 2009.

[4] Oculus, "Oculus Connect 3," San Jose, California, USA, 5.-7 October.2016.

[5] D. A. Bowman and R. P. McMahan, "Virtual Reality: How much Immersion Is Enough?," *Computer,* no. 40.7, 2007.

[6] Jemes J. Cummings, Jeremy N. Bailenson, "How Immersive Is Enough? A Meta-Analysis of the Effect of Immersive Technlogy on User Presence," *Media Psychology,* no. 19.2, pp. 272-309, 2015.

[7] D. R. Nadeau, „Building virtual worlds with VRML," rev. *IEEE Computer Graphics and Applications*, 1999, pp. 18-29.

[8] S. Streuber a A. Chatziastros, „Human Interaction in Multi-User Virtual Reality," *Proceedings of the International Conference on Humans and Computers,* pp. 1-7, 2007.

[9] Witmer, Bob G., and Michael J. Singer, „Measuring presence in virtual environments: A presence questionnaire.," *Presence: Teleoperators and virtual environments 7.3,* pp. 225-240, 1998.

[10] Waterworth, Eva L., and John A. Waterworth., „Focus, locus, and sensus: The three dimensions of virtual experience.," *CyberPsychology & Behavior 4.2,* pp. 203-213, 2001.

[11] D. Thalmann, J. Lee and N. M. Thalmann, "An evaluation of spatial presence, social presence, and interactions with various 3D displays," in *Proceedings of the 29th International Conference on Computer Animation and Social Agents*, ACM, 2016, pp. 197-204.

[12] D. Nunez, „How is presence in non-immersive, non-realistic virtual environments possible?," rev. *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa.*, ACM, 2004, pp. 83-86.

[13] G. Llorach, A. Evans a J. Blat, „Simulator Sickness and Presence using HMDs: comparing use of a game controller and a position estimation system," rev.

Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology, ACM, 2014, pp. 137-140.

[14] J. S. Dumas a J. Redish, A practical guide to usability testing, Intellect books, 1999.

[15] E. Reiss, Usable Usability Simple Steps for Making Stuff Better, Indianapolis, Indiana? USA: WILEY, 2012.

[16] J. Nielsen, Usability Engineering, Elsevier, 1994.

[17] Google, „Google I/O 2015," Moscone Center, San Francisco, California, USA, 28.-29. May2015.

[18] M. Alger, September 2015. [Online]. Available: https://drive.google.com/file/d/0B19l7cJ7tVJyRkpUM0hVYmxJQ0k/view.

[19] L. Erickson, „Channel 9," [Online]. Available: https://channel9.msdn.com/Niners/misslivirose.

[20] Kairat Aitpayev, Jaafar Gaber, „Creation of 3D Human Avatar using Kinect," *Asian Transactions on Fundamentals of Electronics, Communication & Multimedia,* pp. 12-24, 2012.

[21] P.-W. Lee, W. Han-Yu, J.-W. Lin, A. Valstar a Y.-C. Tung, „TranSection: Hand-Based Interaction for Playing a Game within a Virtual Reality Game," rev. *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, ACM, Ed., ACM, 2015, pp. 73-76.

[22] Oculus VR, LLC., „Developer Center | Oculus," 2017. [Online]. Available: http://static.oculus.com/documentation/pdfs/mobilesdk/latest/mobilesdk.pdf.

[23] G. Robles-De-La-Torre, „The importance of the sense of touch in virtual and real environments," rev. *Ieee Multimedia*, IEEE, 2006, pp. 24-30.

[24] Witmer, Bob G., Christian J. Jerome, and Michael J. Singer., „The factor structure of the presence questionnaire.," *Presence: Teleoperators and Virtual Environments 14.3,* pp. 298-312, 2005.

[25] Microsoft, „Microsoft Developer Network," [Online]. Available: https://msdn.microsoft.com/en-us/library/dn799271.aspx.

[26] Leap Motion, „Leap Motion Developer," [Online]. Available: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html.

[27] T. Parisi, WebGL: up and running, O'Reilly Media, Inc., 2012.

Appendix A: Plan

A.1. Evaluation of 3rd semester

For third semester we chose three main goals:

- Partial user representation
- Interaction
- VR VE

Since interaction depends on use-case, we merged this task with implementation of given use-case.

Since devices give us all data, we decided complete user representation will be more efficient. This phase also takes longer because of issues caused by Kinect SDK and incompatibility with UE4, we were unable to use joint orientation from Kinect for representation, therefore we needed to use other approach.

User representation has higher priority, therefore we created it lastly.

Also during implementation process we created some new 3D objects.

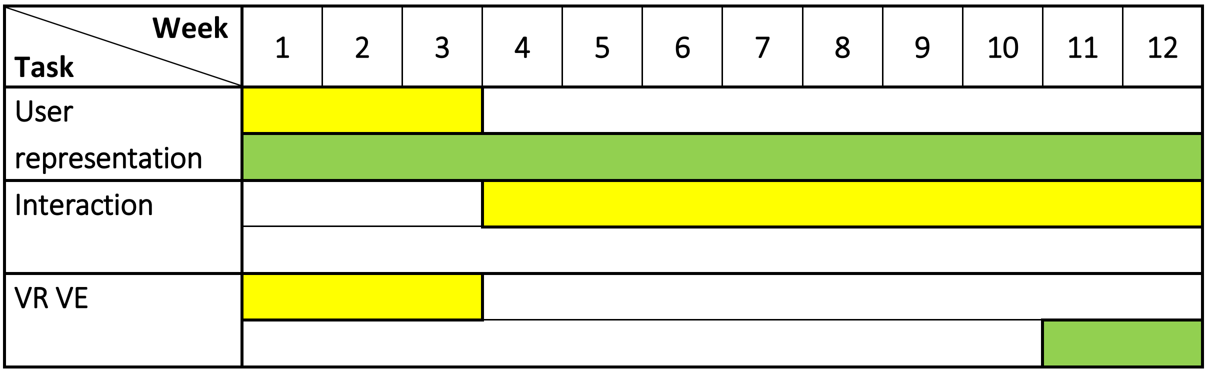

Table 1: plan (yellow) and evaluation (green) for 3rd semester

A.2. Plan for 4th semester

From design for last semester there are use-cases that need to be implemented and performing presence testing. Therefore, we have three main tasks:

- Implementing use-cases,
- Performing presence testing and
- Evaluating results of testing

Since UE4 contains many models, we are unable to say how many models we will be able to use and how many of them we will need to create by ourselves. Because of these circumstances we cannot plan this task.

Task \ Week	1	2	3	4	5	6	7	8	9	10	11	12
Use-cases												
Presence Testing												
Evaluation												

Table 2: Plan for 4th semester

Appendix B: Source

UE4 project with sources, models and electronic version of this document can be find on: **https://github.com/jozef17/MastersThesis-VR-Presence**

B.1. CD

JozefBlazicek.docx	electronic version of this document
JozefBlazicek.pdf	electronic version of this document