Embodiment of a Raindrop: How sense of embodiment is affected in Virtual Reality

Project Report

MTA191032

Aalborg University
IT and Design

AALBORG UNIVERSITY
STUDENT REPORT

IT and Design
Aalborg University
http://www.aau.dk

Title:
Embodiment of a Raindrop: How sense of embodiment is affected in Virtual Reality

Theme:
Master's Thesis

Project Period:
Spring Semester 2019

Project Group:
MTA191032

Participant(s):
Peter Weilgaard Brasen 20143290
Mathias Christoffersen 20146013

Supervisor(s):
Martin Kraus

Copies: 1

Page Numbers: 57

Date of Completion:
May 27, 2019

Abstract:

This report investigates the sense of embodiment users have in a Virtual Reality experience, and how the features haptic feedback, interaction, and attachments to the virtual environment may have an impact. This is done using a commercialised Virtual Reality product, HTC Vive.
To investigate this, a VR experience has been developed with four versions. The control version, called interactive, have all the mentioned features implemented, while the other versions have one feature removed each. Thus, the other versions are called non-haptic, non-interactive, and detached.
In total, the experiment was conducted with 17 participants. The experiment has been done with a within-subject design approach, having all participants test all four version, but in different order, after having tried a tutorial of the VR experience. Questionnaires and recorded tracking data of the participant's movement was used to evaluate the experiment.
It was found that interaction and haptic feedback increase the user's sense of embodiment, while attachments to the environment did not.

Danish Summary

Denne rapport undersøger følelsen af krops besiddelse (sense of embodiment) i brugere af en Virtual Reality oplevelse, og hvordan karakteristisk egenskaber som haptisk feedback, interaktion, og tilknytning til virtuelle omgivelser, har en virkning. Dette er opnået ved brug af kommercielle VR produkter som HTC Vive.

En VR oplevelse er blevet udviklet. Denne oplevelse går ud på at brugeren oplever et scenarie af en regndråbe, fra regndråben danner sig i en regnsky, til den falder og rammer jorden.

Fire versioner blev udviklet med henblik på at undersøge de karakteristiske egenskaber. Kontrol versionen som blev navngivet "interactive" har alle de førnævnte egenskaber implementeret, imens de andre versioner havde hver især frataget en af de nævnte egenskaber. Derfor er de andre version blevet navngivet "non-haptic", "non-interactive" og "detached".

Eksperimentet blev udført med 17 testpersoner, og testen gjorde brug af "within-subject" designet. Alle testpersoner startede med en tutorial for at mindske hvor meget de lærte indimellem hver test. Derefter prøvede testpersonerne de fire versioner i forskellige rækkefølger. Rækkefølgerne var unikke for hver testperson.

Dataen til at evaluere eksperimentet blev indsamlet via spørgeskemaer. Efter hver test skulle testpersonerne svare på spørgeskemaet som bestod af Likert skalaer. På denne måde kunne svarene sammenlignes ved hjælp af Wilcoxon Signed Rank test.

Mere data blev samlet af testpersonernes hoved- og håndbevægelser for hver test de udførte. Disse bevægelser kan beskrives som ændring af position og rotation, og ud fra det kunne der samles gennemsnitlig bevægelser vist som meter per sekund og grader per sekund for hoved og hænder. For at analysere dette data blev Welch t-test brugt.

Det blev vist at H1: "Interaktion i en VR oplevelse øger brugerens krops besiddelse." er accepteret. Dette er på baggrund af at versionen "non-interactive" viste nedsat kropsejerskab og dataen gav udtryk for at brugeren også havde nedsat lokationsevne og handlekraft, hvilket betyder at i versionen "interactive" havde brugeren en øgelse af krops besiddelse.

H2: "Tilknytning til objekter af virtuelle omgivelser i en VR oplevelse øger brugerens krops besiddelse." kunne ikke bevises. Grunden til dette er spekuleret til at være designet af interaktionen i "detached" version af VR oplevelsen. Det er svært at udpege hvad i designet der gør forskellen, men dette giver anledning til en videre undersøgelse af interaktionstypen.

H3: "Haptisk feedback i en VR oplevelse øger brugerens krops besiddelse."

blev accepteret. Versionen "non-haptic" viste nedsættelse i kropsejerskab og handlekraft.

Som konklusion, dette projekt viser at egenskaber som interaktion og haptisk feedback øger brugerens krops besiddelse. Yderligere undersøgelser indenfor dette emne kan gavne udvikling af VR oplevelser som sigter efter forhøjet krops besiddelse.

Contents

Preface

This report is written by Peter Weilgaard Brasen and Mathias Christoffersen. We both received our bachelor's degree in Medialogy at Aalborg University. This is our thesis for our master's degree in Medialogy with a specialisation in games.

We have previously worked with Virtual Reality in several projects. This is due to our interests for the rapidly expanding usage of Virtual Reality, and the great potential it has with the extensive continuation of development we see today.

We give thanks to Martin Kraus for supervising this project, and to all the participants of the experiment.

Aalborg University, May 27, 2019

Peter Weilgaard Brasen
<pbrase14@student.aau.dk>

Mathias Christoffersen
<matchr14@student.aau.dk>

Chapter 1

Introduction

VR (Virtual Reality) gives a unique opportunity to immerse the user into a virtual world, where they can experience a different point of view as another person, animal, or entity. Creating this immersion, or understanding, of being someone or something else, can be a challenge. Thus, having the user become an inanimate object could presumably be more challenging as there may be less for the user to relate to.

Grundfos, the pump manufacturing company, considers SDG6 of UN's Sustainable Development Goals to be very important, due to its focus on access to clean water and sanitation. To support their efforts surrounding clean water, a VR experience could be developed to show the user what happens to the water by letting the user become a drop of water.

For this project, it is relevant for the user to feel the sense of embodiment of being a raindrop, as it may improve the experience. Thus, it is set as a goal to investigate possibilities of increasing the sense of embodiment the user has of being a raindrop in VR using technologies available to commercial VR.

Chapter 2

Background Research

This chapter investigates research of sense of embodiment, and articles that are found to be useful for developing a VR experience.

2.1 Presence and Embodiment

This section investigates research done on presence and embodiment. Subjects that play a factor for presence and the subcomponents that make up embodiment is investigated.

2.1.1 Presence

In VR and other computer-transmitted environments presence has been shown to refer to a *"Psychological state in which even though part or all of the individual's current experience is generated by and/or filtered through human-made technology, part of all of the individual's perception fails to accurately acknowledge the role of the technology in the experience"* (Riva, 2019)[20, p. 159].

There are multiple different forms of presence some of them being Social presence, Co-presence, Self-presence, Hyper presence and Telepresence (which today is considered as just 'presence')[22, p. 434-449]. The most interesting form of presence in accordance to this experiment is the form of self-presence.

2.1.2 Self-Presence

Self-presence occurs when a user feels that there is no distinction between themselves and their representative avatar. It happens often in virtual environments where the user is represented through an avatar or in first person shooter games.

Self-presence can be further categorized as proto, core and extended self-presence. Proto self-presence can be referred to as the body-level self-presence and concerns itself with to what extent the self-representation is coherent to the body schema.

Core self-presence, also known as emotion-level self-presence is the extent to which interaction between objects and self can cause emotions and emotional response. Extended self-presence is also referred to as the identity-level self-presence and it focuses on how the self-representation can be related to personal identity[15, p.322-336].

2.1.3 Display Parameters

The minimum frame-rate for presence is around 15Hz[1, p. 3-16]. A study done by Meehan & Brooks dives into the Physiological measures of presence in stressful virtual environments[16, p. 645-652], the study monitors the change in heart rate as a correlation to presence and had the participants look over a virtual precipice, the study furthermore suggests that the frame-rate should be at a bare minimum of 15Hz.

2.1.4 Visual Realism

A paper by Maria V. Sanchez and Mel Slater[21, p. 7-11] suggests that there has to this date been no support that the visual realism is an important factor to presence. Although the display of dynamic shadows seems to have an improving effect on behavioural presence.

2.1.5 Sound

Sound has a high value in relation to presence in virtual environments, a study done by Hendrix and Barfield[9, p. 290-301] suggests that there is a significant change in higher presence when the virtual environment has spatialized audio compared to non-spatialized or no audio.

A study by Brenda Laurel[11] claims that spatialized audio is one of the core components that defines VR.

2.1.6 Vibrotactile Feedback

A meta-analysis done by Prewett et al., that concerns itself with 45 different case studies of vibrotactile and visual cues for task improvement, suggests that the use of vibrotactile cues have positive results on performance. This is more distinct when the vibrotactile cues repeat rather than replace other cues such as visual, an important note is that the variation of those studies are quite large and so the requirement for careful consideration when developing vibrotactile cues is advised[18, p. 123-132].

Vibrotactile feedback has also been suggested to increase embodiment in amputees by a study of D'Alonzo et al. The study forms itself after the rubber hand illusion with vibrotactile feedback as stimuli and shows promising results of the possibility to equip commercialized prosthetic products with cheap vibrators to create the illusion of embodiment[5, p. 450-457].

2.1.7 Virtual Body Representation

A study done by Slater & Usoh discovers the importance of virtual body representation, in their study they find a relationship of higher presence connected to a more realistic virtual body. In this case the virtual bodies were a complete figure with arms and legs compared to a body only consisting of 3D Arrow cursors. The participants in the test would have their right arm tracked so that the body in the virtual environment would follow their motions, however the left arm would not be tracked and therefore would act as dead weight. The discovery of having such an arm led the participants to believe that there were something wrong with them[23, p. 221-234].

A follow up study was done where the participants left arm would mirror the right arm, this study revealed that some participants would move their real left arm in accordance to the virtual left arm[24, p. 130-144].

2.1.8 Body Engagement

One of the current problems when dealing with a HMD (Head-mounted display) and VR setup is the cables connected to the main computer. These cables might limit the range of which the participant can move around.

There are other ways to compensate for the lack of free-movement, one of the most common ways are to have the participant press a button on the controller to activate movement. Slater et al. found that there seemed to be a higher sense of presence in the participants when they could move around by just “walking in place” rather than pressing a button[25, p. 201-219].

2.1.9 Measuring Presence

There are several ways that presence can be measured, some more convenient than others.

The usual approach to measuring presence is through questionnaires, typically the user will do a task in a virtual environment and then afterwards answer questions with ordinal scales that ranges from “no presence” to “full presence”[12, p. 282-297].

It is important to note that the questionnaire based assessment of presence can be unstable due to prior information[6, p. 1-13]. Another way to measure presence is through behavioural observation. If the participant for example responds physically to something related to the virtual environment, for example like swaying or ducking to avoid flying objects appearing in the virtual environment.

Further specialised behavioural observation can be done in the form of physiological measurement, factors such as galvanic skin response can be used[16, p. 645-652].

2.1.10 Embodiment

Embodiment can be described as the phenomenon of being aware of your own body, it is often suggested that embodiment proves as an invaluable prerequisite for other types of sensation such as self-presence and immersion[13, p. 978-998].

One of the most iconic experiments concerning embodiment is the rubber hand illusion. The experiment has the user view stimulation of a rubber hand while experiencing stimulation on their own as if the rubber hand is part of the user's body[4, p. 229-240].

Another interesting aspect of embodiment is the sense of embodiment, it can be divided into three subcomponents; Sense of agency, sense of self-location and sense of body ownership.

2.1.11 Sense of Agency

Sense of agency refers to the feeling of global motor control and is usually present in active movement.

The sense of agency is increased when the predicted and actual consequences of active movement matches. Mismatch between visual feedback of an action and actual movement have a negative impact on the sense of agency.

2.1.12 Sense of Self-location

Sense of self-location is the spatial experience of being inside a body, the relationship between body and environment.

Self-location can be influenced by tactile input, visuospatial perspective and vestibular signals.

2.1.13 Sense of Body Ownership

Sense of body ownership is induced by both bottom-up and top-down influences, this means that the sensory information received by our brain and the cognitive process of presuming that an artificial body can be one's own are combined into what can be addressed as the sense of body ownership.

The level of sense of body ownership can be affected by the degree of morphological similarities in the body or limbs of the avatar that the user is given.

2.1.14 Measurement of Sense of Embodiment

The approach to measuring degree of sense of embodiment is very like that of the approach to measure presence.

The individual contribution of each subcomponent is widely unclear because of the complexity of the sense of embodiment, there is a lack of experimental evidence when addressing relationships between the subcomponents both positive and negative.

Slater et al. has worked on focus points of measurement when dealing with the three subcomponents of sense of embodiment[10, p. 373-387].

For self-location it is important to look at the physiological response, such as skin conductance, in the view of a threat. Questionnaire items such as *"I experienced that I was located at some distance behind the visual image of myself, almost as if I was looking at someone else"*[10, p. 381 Table 2] and estimation for body position.

Sense of agency is more focussed on control of the avatar limbs, questions such as *"It seemed like I was in control of the hands"* and *"I felt I was controlling the hands"*.

Sense of body ownership aims for the feeling of ownership with questions such as *"I felt as if the virtual body was mine"* and *"I felt as if the limb was my limb"*.

Physiological responses such as heart rate acceleration and deceleration to threat and changes to temperature of the user[10, p. 381 Table 2].

2.2 Rain

This section focuses on rain and one method used to render rain in computer graphics.

2.2.1 Raindrops

Raindrops are described within the hydrologic cycle as precipitation, water particles falling from the atmosphere to the ground.

In clouds, precipitation occurs due to water vapour condensing to droplets. This is due to the vapour rising in altitude and decreasing in temperature, and with enough growth the droplets will start to fall, resulting in rain[7, p. 51].

2.2.2 Rendering

Rendering photorealistic fluids are often complex and taxing on performance. Slomp et al. authored an article proposing a method for rendering raindrops in the number of millions at real-time possible with high performance[26]. Their method can be considered to give nearly the same result as ray-traced raindrops, even though it uses an environment map.

The method uses a pre-processing stage and a run-time stage. The pre-processing stage ray-traces a mask that contains reflection and refraction vectors, and Fresnel coefficients for each pixel of the raindrop, and the real-time stage then renders the raindrops as billboards. While the method gives good results, it does have its limitations.

2.3 State of the Art

This section is concerned with what is state of the art within the subject of VR experiences.

2.3.1 The Stanford Ocean Acidification Experience

The Stanford Ocean Acidification Experience[29] (SOAE for short) is a project made by researchers from Stanford. The experience functions as a wakeup call to reduce the CO2 emission, it takes the user by the hand and show them what the consequences of increasing CO2 can be.

The experience starts in a big city filled with cars (figure 2.1), here the experience follows a CO2 molecule to the ocean where the user gets to witness the creation of ocean acidification. Afterwards the user is taken to the bottom of a reef where a search for snails take place, before and after increased ocean acidification, the conclusion is that the acidification will negatively impact a lot of animal life.

Figure 2.1: Stuck in a traffic jam in the city, the car in front is exhausting many CO2 particles[30]

SOAE utilises the strength of virtual reality for this experience through informational storytelling, interaction and embodiment.

Through the whole experience the user is guided by a female voice that explains what the user must do, the experience requires the participant to interact with certain elements through it i.e. car exhaust, snails and CO2 molecules. To increase embodiment the SOAE make use of haptic feedback, the haptic feedback can for example be found when interacting with hydrothermal vents on the bottom of the reef at the end of the experience (figure 2.2).

Figure 2.2: The user is able to feel the stream from the hydrothermal vents through haptic feedback[30]

Chapter 3

Problem Formulation

In this chapter the problem formulation is presented, and research questions and hypotheses are made.

This project is about investigating sense of embodiment in VR experiences and how it can be affected. Thus, the problem formulation is stated as:

"How can the sense of embodiment be affected in a VR experience?"

3.1 Research Questions

These research questions attempt to narrow down the problem formulation to certain aspects that can be investigated.

1. How does sense of embodiment change between a VR experience using interaction and one that does not use interaction?
2. How does attaching the user to the virtual environment affect the sense of embodiment in an interactable VR experience?
3. How does haptic feedback affect the sense of embodiment in an interactable VR experience?

3.2 Hypotheses

From the research questions hypotheses were made, specifically focusing on an increase of sense of embodiment due to a limited change in the VR experience.

- **H1:** Interaction in a VR experience increases the user's sense of embodiment.
- **H2:** Attachment to objects of the virtual environment in a VR experience increases the user's sense of embodiment.

- **H3:** Haptic feedback in a VR experience increases the user's sense of embodiment.

H1 anticipates that having interaction present in an experience will increase the sense of embodiment.

H2 expects that with the user having an attachment to objects of the virtual environment increases the user's sense of embodiment. This attachment can be described as a design making the user appear as being part of or like the virtual environment.

H3 predicts that having the implementation of haptic feedback will increase the sense of embodiment.

Chapter 4

Implementation

This chapter contains the design and implementation of the VR experience. This includes the game engine and development of several components.

4.1 Design

To develop a VR experience where the user can embody a drop of water, it was determined that the experience should be enclosed to a scenario of a raindrop.

The scenario was chosen to begin in the clouds where small water particles merge into raindrops, which eventually make them fall to the ground. This is the two phases of the experience, collect phase and fall phase.

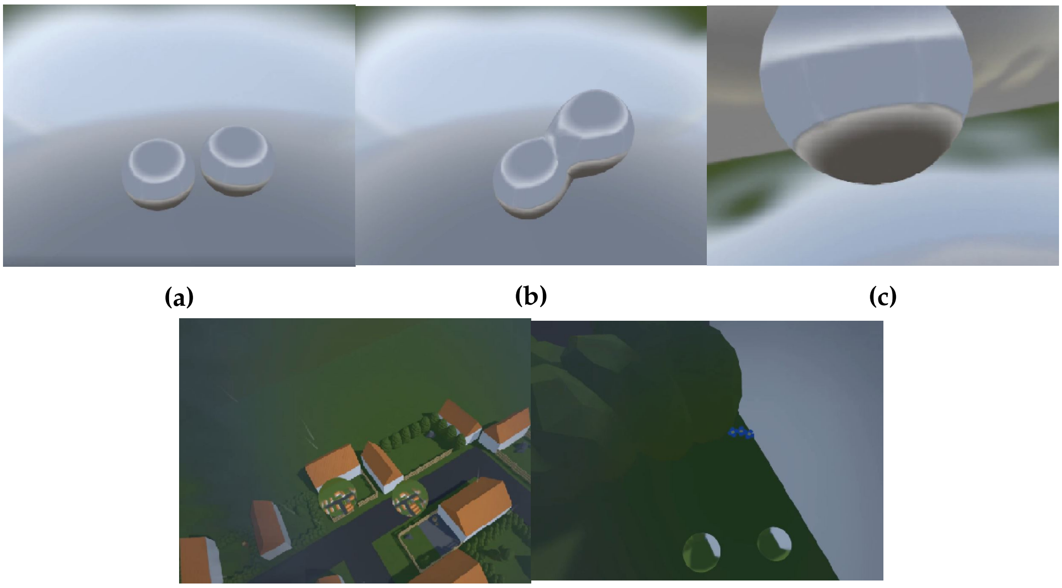
When the user equips themselves with the HMD and controllers, the experience is in a start state, and will stay so until the user gives the command to start (by pressing down the trigger on either controller) to begin the experience.

The experience begins with the collect phase, taking place in a cloud environment, in which the hands of the user will appear as water particles (figure 4.1a). With these, the user can collect small water particles ascending from underneath by merging them together. During the phase, ambient audio of a light breeze is playing, and when collecting the small water particles, the user receives auditory and haptic feedback (figure 4.1b).

After the user has collected the required amount of water particles, the user's water particles will expand and cover the user's head as a transition to the next phase (figure 4.1c).

During the falling phase, the user will find themself inside a raindrop falling from the sky, slowly towards the ground populated by a small neighbourhood (figure 4.1d). During this phase, the user can stretch out their arms and change the direction they are falling in. While falling, the ambient audio changes from a light breeze to a more violent wind.

Finally, when the user reaches a surface, auditory feedback is played to convey the collision, ambient audio changes to rain whether, and the vision of the user fades to black signifying the experience is over (figure 4.1e).



(a) (b) (c) (d) (e)

Figure 4.1: Sequences of the VR experience.

4.1.1 Versions

For the evaluation of the project, multiple versions of the VR experience were created by modifying the main version named the **interactive** version, and alternative versions; **non-interactive**, **non-haptic**, and **detached**.

The interactive version (figure 4.2) can be described as having all features implemented such as audio, haptic feedback when merging rain drops and interactive tasks such as merging the water particles together and steering the raindrop while falling. Furthermore, the hands of the participant would appear as small water particles.

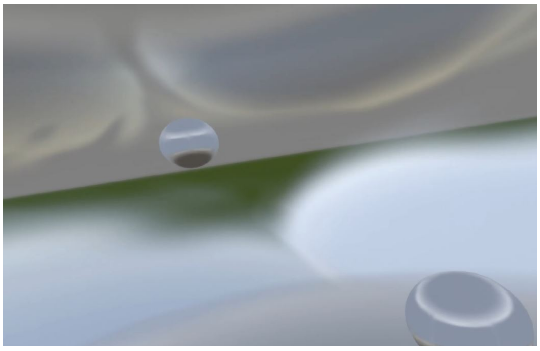


Figure 4.2: Interactive version.

These other versions differ in these ways:

- **Non-interactive** (figure 4.3), much like the **interactive** version, had all the features except the ability to interact with the environment such as merging rain drops and steering the falling raindrop. Those events are scripted and controlled by animations. Additionally, the user's hands are not visible during the experience.

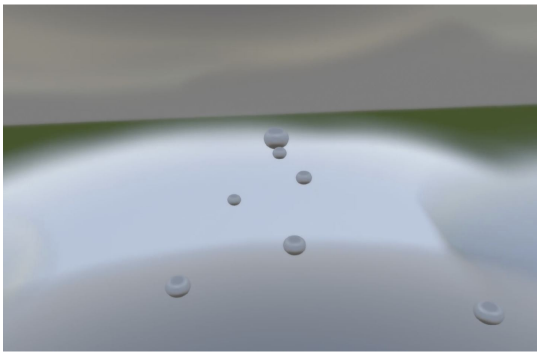


Figure 4.3: Non-interactive version.

- **Non-haptic** makes haptic feedback absent in the event of merging water particles.

- **Detached** changes the interaction during the collect phase. This version does not change the user's hands into water particles but leaves them visualised as controllers (figure 4.4).

  The interaction is changed to the user having to grab water particles with their hands and merge them instead of merging them with water particles on the user's hands.

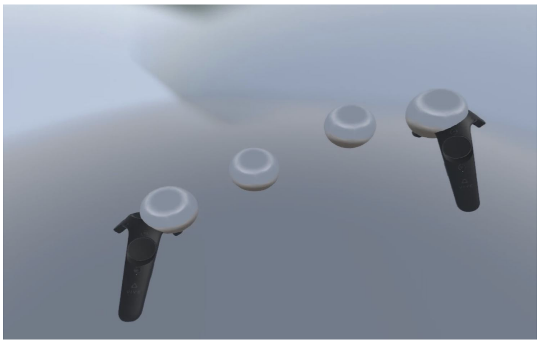


Figure 4.4: Detached version.

A tutorial version was also developed for the user to learn the functionality of the experience. This version is excluding the haptic feedback, and shows more text instructing the user in how to complete the experience.

4.2 Game Engine

The game engine used for the implementation of this project is *Unity*[28] (version 2018.3.9f).

Valve's *SteamVR* plugin[27] for *Unity* is used for the implementation of VR.

4.3 Shaders

Three shaders were developed for the experience, one for the small water drops, one for the large water bubble, and one for clouds. All three shaders were developed using Unity's surface shader approach.

4.3.1 Waterdrops

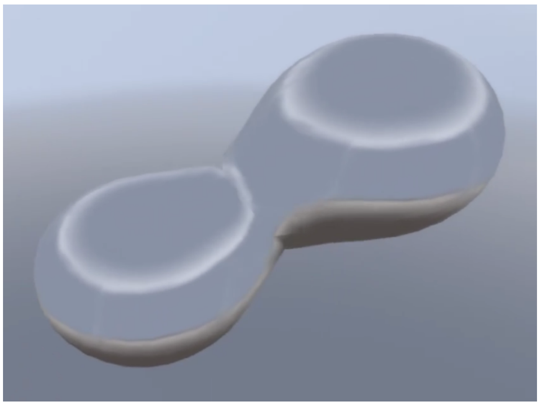


Figure 4.5: Waterdrops merging.

Inspired by the raindrop rendering method by Slomp et al.[26], a simple refraction shader for water drops was developed with additional vertex deformation (figure 4.5).

The deformation occurs when two water drops collide with each other. When deforming the water drop a set of variables are needed, size of the water drop in question, and size and position of the other water drop.

Due to water drops being spherical, the size is seen as radius. The deformation is calculated by finding the position of a mirrored vertex, with respect to the difference in size.

Interpolation is then used with the distance between the water drops as a parameter, making it so between two colliding water drops, more vertices will meet in the middle between them the closer they get.

Calculating new normals are done using the same method of interpolating between the existing normal and a mirrored normal (listing 4.1).

```
        void vert (inout appdata_full v){
            if (_Simple == 1) return;

            float d = dot(v.normal, normalize(_NearSpherePos));
            float3 dir = normalize(_NearSpherePos);
            if (length(_NearSpherePos) == 0) dir = float3(0,0,0);
            float r1 = length(v.vertex.xyz);
            float r2 = (_NearSphereRadius/_SphereRadius) * r1;

            float maxDistance = (r1 * (1 + _MergeDistance)) + (r2 * (1 +
    _MergeDistance));
            float distanceFactor = max(0, maxDistance - length(_NearSpherePos))/
    maxDistance; // range 0-1, higher is closer

            float3 reflectedNormal = reflect(normalize(v.vertex.xyz), normalize(-
    _NearSpherePos));
```

```
            float3 reflectedVertex = (reflectedNormal * r2) + _NearSpherePos;
            if (length(_NearSpherePos) != 0) {
                float3 newPos = lerp(v.vertex.xyz, reflectedVertex, 0.55f *
    expSmooth(0, 1-d, distanceFactor) * step(0, d));
                if (length(newPos) > r1) v.vertex.xyz = newPos;
            }

            if (length(_NearSpherePos) != 0){
                float3 newNormal = lerp(v.normal, reflect(v.normal, normalize(-
    _NearSpherePos)), 0.5f * expSmooth(0, 1-d, distanceFactor) * step(0, d));
                v.normal = newNormal;
            }

        }
```

Listing 4.1: Vertex function of the water drop shader.

Lastly, a refraction direction can be calculated from the view direction and normals. which is then used to sample a reflection cube (listing 4.2).

```
        void frag(Input input, SurfaceOutputStandard o, inout fixed4 color)
        {
            if (_Simple == 1) return;

            float refractiveIndex = 1.5;
            refractiveIndex = _Refraction;
            bool refractMirror = (_Refract > 0.5 ? true : false);
            float3 divergeDir = refract(-input.viewDir, -input.worldNormal, 1.0 /
    refractiveIndex);
            float3 refractedDir = refract(input.viewDir, input.worldNormal, 1.0 /
    refractiveIndex);
            half4 refCube = UNITY_SAMPLE_TEXCUBE(unity_SpecCube0, refractMirror ?
    refractedDir : divergeDir);
            color = refCube + refCube * _LightColor0;
        }
```

Listing 4.2: Frag function of the water drop.

4.3.2 Big Water Bubble

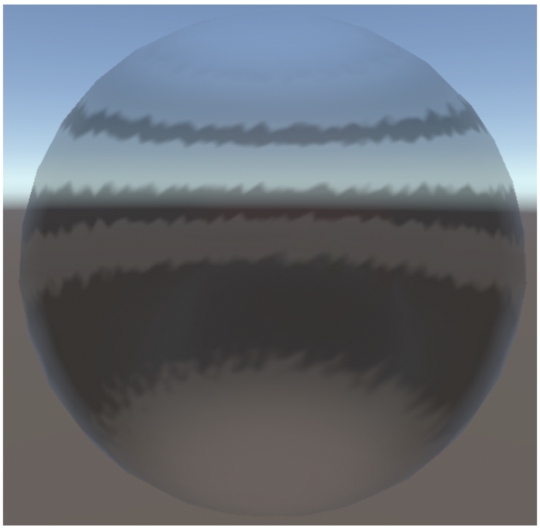


Figure 4.6: Big water bubble shader implementation

Another shader used is for the big water bubble surrounding the user while they are falling. This shader utilises vertex deformation, noise, and animation (figure 4.6).

The vertex deformation is calculated using a dot product of the falling direction of the user and normals (listing 4.3).

The vertices are then moved by a vector calculated by the dot product or 0 (whichever is highest).

,

```
void vert (inout appdata_base v){
    if (_Simple == 1) return;

    float velD = dot(v.normal, _MorphDirection);
    float3 velDeform = _MorphDirection * max(0, velD);
    v.vertex.xyz += velDeform * 0.5;
}
```

Listing 4.3: Vertex function of the big water bubble.

To add more visual movement to the shader, noise combined with animation is implemented. The noise is calculated using a Perlin noise function from a *Shader Graph* node[8]. The noise is animated by using the time variable and adding a noise offset, as seen in listing 4.4.

```
void surf (Input IN, inout SurfaceOutputStandard o)
{
    float3 noiseVector = IN.worldPos;
    float3 dir;
```

```
            if (any(_MorphDirection)) dir = normalize(_MorphDirection);
            else dir = float3(0,−1,0);
            float d = dot(dir, IN.worldNormal);
            float offsetNoise = Unity_GradientNoise_float(IN.worldPos.xz + IN.worldPos.xy + IN.worldPos.yz, 5);
            float offset = _Time.y ∗ 0.1 + offsetNoise ∗ .05;

            float2 noisePos = float2(noiseVector.y + offset, noiseVector.y + offset);
            float2 noisePos2 = float2(length(dir + IN.worldNormal) + offset, length(dir + IN.worldNormal) + offset);
            float intensity = 3;
            float noise = Unity_GradientNoise_float(noisePos2, intensity);

            fixed4 c = _Color ∗ smoothstep(0, 1, length(noise));
            float a = smoothstep(.4, .6, length(noise));

            if (_Simple == 1){
                o.Albedo = _Color.rgb;
                o.Alpha = _Color.a;
            }
            else {
                o.Albedo = abs(noise) ∗ _Color;
                o.Alpha = saturate(lerp(_Color.a, 0.75, a) ∗ (1 − smoothstep(.7,1, abs(d))));
            }
        }
```

Listing 4.4: Surface function of the big water bubble.

4.3.3 Clouds

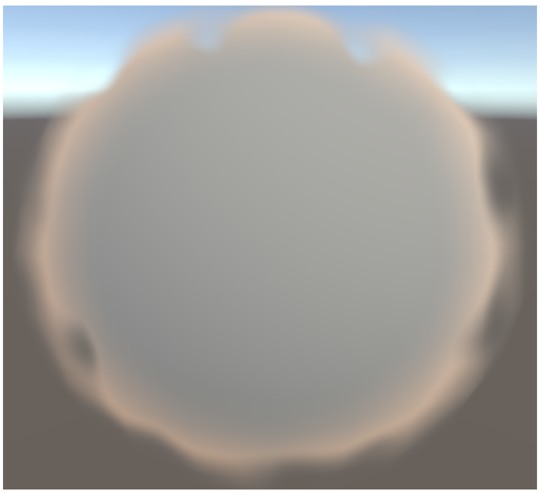


Figure 4.7: Cloud shader implementation

A simple cloud shader is used for the environment (figure 4.7). This shader uses vertex deformation based on the same gradient noise function in the big water bubble shader, and animates it using the time variable (listing 4.5).

```
        void vert (inout appdata_full v){
            float3 perlinValue;
            float offset = _Time.y * _TimeScale;
            float4 worldPos = mul(unity_ObjectToWorld, v.vertex);
            perlinValue.x = Unity_GradientNoise_float(worldPos.xy + float2(offset, 0), _PerlinIntensity) - 0.5;
            perlinValue.y = Unity_GradientNoise_float(worldPos.xz + float2(0, offset), _PerlinIntensity) - 0.5;
            perlinValue.z = Unity_GradientNoise_float(worldPos.yz + float2(offset, 0), _PerlinIntensity) - 0.5;
            v.vertex.xyz += perlinValue * _PerlinVertexScale * clamp(0, 1, dot(perlinValue, v.vertex.xyz));
            v.normal = normalize(v.vertex.xyz);
            v.color.rgb = perlinValue;
        }
```

Listing 4.5: Vertex function of the cloud.

From the dot product of the view direction and normals rim lights are added, and additionally mixing noise and the dot product for the alpha creates a fade effect on the edges of the cloud (listing 4.6).

```
        void surf (Input IN, inout SurfaceOutputStandard o)
        {
            float d = dot(IN.worldNormal, normalize(IN.viewDir));
            float3 perlinValue;
            float offset = _Time.y * _TimeScale * 5;
            float3 noiseVector = IN.worldPos;
            float newScale = _PerlinAlpha;
            perlinValue.x = Unity_GradientNoise_float(noiseVector.xy + float2(offset, 0), newScale);
            perlinValue.y = Unity_GradientNoise_float(noiseVector.xz + float2(0, offset), newScale);
            perlinValue.z = Unity_GradientNoise_float(noiseVector.yz + float2(0, 0), newScale);
            perlinValue = perlinValue * 2 - float3(1,1,1);

            float fresnel = smoothstep(_FresnelMin, _FresnelMax, d);
            float noise = smoothstep(0, 1, length(perlinValue)) * smoothstep(0.1, 1, d);
            fixed4 c = _Color;
            o.Albedo = c.rgb;
            o.Alpha = smoothstep(0, .5, saturate(fresnel + noise));
            o.Emission = smoothstep(.25, 1, 1 - d) * _RimColor * o.Alpha;
        }
```

Listing 4.6: Surface function of the cloud.

4.4 Virtual World

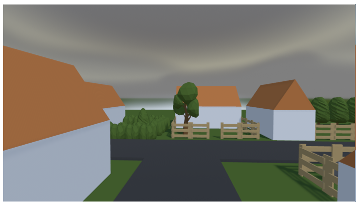
4.4.1 Environment

The environment is several objects representing a small neighbourhood (figure 4.8). These objects consists of primitive cubes, prisms made with *Probuilder*[19], and various props from the asset pack *"Low Poly Pack - Environment Lite"*[14].

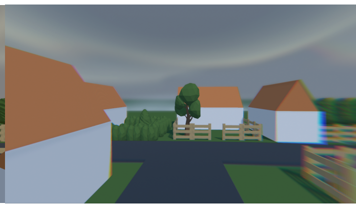
Figure 4.8: Environment in the VR experience.

The material for these objects uses *Unity*'s mobile diffuse shader, and a single texture included with the asset pack.

4.4.2 Post-processing



(a)



(b)

(c)

Figure 4.9: Post FX profiles. From left, normal profile, FX profile, water drop profile.

For post-processing, *Post Processing Stack v2*, or *PostFX v2*[17] was used. During the experience the user will see three different profiles used by *PostFX v2*.

The first profile is global, meaning it is the default profile when no other profile is prioritised (figure 4.9a). This profile only adds depth of field effect making far objects more blurred.

The other two profiles are for when the user have their head inside small water drops (figure 4.9c) and the larger bubble (figure 4.9b), using depth of field, vignette and chromatic aberration for both, and colour grading additionally for the large bubble.

4.4.3 3D Skybox

To add animated clouds to the environment a 3D skybox was implemented. This was done by creating an additional camera in the unity scene with a lower depth, and cull everything but objects intended for the 3D skybox. The main camera will thus display the second camera as an animated skybox.

4.4.4 Rain Effect

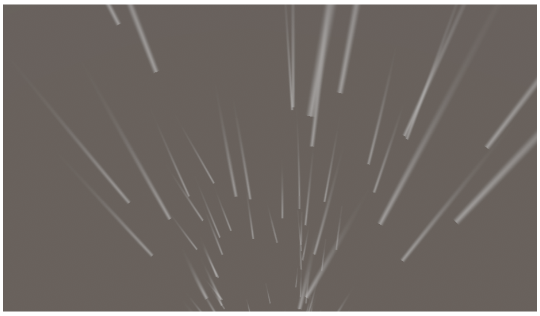


Figure 4.10: Rain effect in the VR experience.

To simulate other water drops, or raindrops, beside the user during the falling section of the experience, Unity's particle system was used together with the default particle sprite.

To simulate rain, the particle system does not render the particle, but instead a trail (figure 4.10). Additionally, the particles receive a small change in velocity over time in order to make the particles appear more different to each other.

Chapter 5

Evaluation

This chapter describes the methods used for evaluating the project, procedure of the experiment, and the results found.

5.1 Method Selection

This section describes the methods used, including the design of the experiment and how data is collected.

5.1.1 Methods

The evaluation used a within-subject design for the different versions of the VR experience, this meant that all the participants played through the four different versions and a tutorial.

The order of the versions were scrambled so that no participant would get the same order, the strengths of choosing within-subject design were that it offers a *"substantial boost in statistical power"* and that it is *"more naturally aligned with most theoretical mindsets"*[2, p. 2].

5.1.2 Data Collection

The data collection consisted of two parts, questionnaires and tracking data.

The questionnaires consisted of 17 questions and could be divided into four sub parts; Demographic, Self-location, Agency and Body ownership, the latter three used a 7-point Likert scale for all their questions, where 1 is "Strongly disagree and 7 is *"Strongly agree"*.

The demographic section aimed to assess the age, gender, previous VR experience and likeliness of motion sickness of the participant.

The self-location included questions such as *"I felt that I had two bodies"*, *"I could no longer feel my body, it was almost as if it had disappeared"* and *"I experienced that i* *was located at some distance behind visual image of myself, almost as if I was looking at someone else"*.

The agency section contained questions like; *"It seemed like I was in control of the water bubble"*, *"It felt like I was able to steer the water bubble while falling"* and *"I was able to merge the water drops with my hands"*.

Examples of the questions in the body ownership section would be; *"I felt as if the virtual body was my body"* and *"I could feel the drops merging with my virtual body"*.

The tracking data recorded during the experiment was done by the application itself. For each frame the application recorded changes in the position and rotation of the user's head and hands. Recorded changes in position are recorded in meters, and changes in rotation recorded in degrees.

Data was recorded for each frame; thus, a timestamp was included for every sample, allowing for finding means from the data. Additionally, a timestamp for when the VR experience transitions from the collect phase to the fall phase was recorded, adding the possibility to create subsets of each phase.

5.2 Test Procedure and Setup

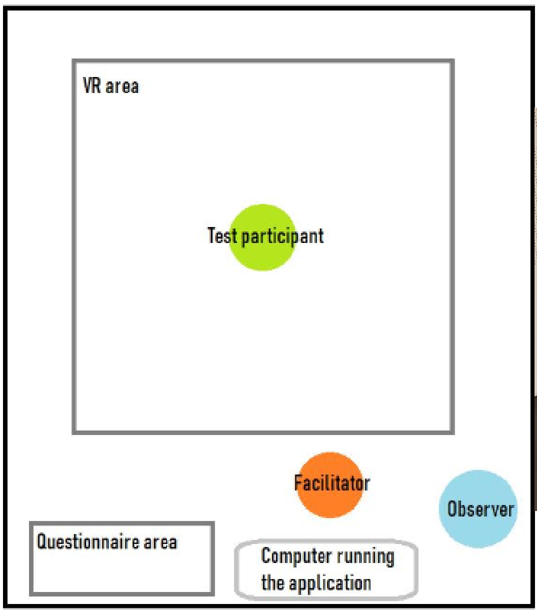
The tests were conducted in the AVALAB at Create, Rendsburggade 14 in Aalborg[3].

Avalab offers three VR setups that includes a computer, HMD and controllers. The area furthermore consisted of two couches and a couple of small tables.

The setup was divided into two areas of operation, as seen on figure 5.1a, the VR area that was mostly empty floor that allowed for VR without having to worry about colliding with walls, tables, chairs etc.

The questionnaire area was equipped with a couch, a table and a laptop. The facilitator instructed the participants about the procedure and equip them with the HMD, headphones and controllers.

The observer took notes if something unforeseen happened such as bugs and comments from the participants.



(a)

(b)

Figure 5.1: Overview of the testing area, including the VR area, questionnaire area, the computer running the application and the facilitator and observer.

The Participants started the test by agreeing to a consent form and answer the introductory questionnaire which assessed demographic, previous VR experience and tendencies of motion sickness both in and out of VR.

Following the questionnaire, the participant was equipped with a wireless VR headset, headphones and controllers, then they were informed about the following procedure.

The participants started out with a tutorial that showcased the VR experience, and prevent noise emerging from learning, after the tutorial was completed the first test version followed.

There were four different test versions that the participant played through and after each test they were required to fill out a questionnaire that aimed to assess the level of embodiment that the participant had through that specific version, the tests took approximately 15 to 20 minutes per participant.

The order in which the four versions appeared was scrambled so each participant had a unique order of versions.

5.3 Participants

The experiment was conducted with 17 participants which ranged from 21 to 28 in age, 3 females and 14 males. 15 had previous experience with VR and 2 had never tried VR before.

All the participants were living in Aalborg.

The participants were not informed about the research area of the test and were only given sparse information such as estimation of time required for completion and how many versions there were.

5.4 Equipment

Below is a list of equipment used for the testing.

1. VR-ready computer with GeForce GTX 1080 graphics card
2. Vive Wireless Adapter and HTC Vive HMD
3. HTC Vive Controllers
4. Lenovo Laptop Model Z50-70
5. Asus Cerberus V2 Headset

5.5 Results

This section tests the data collected in order to find the results of the experiment.

5.5.1 Questionnaire

Testing the responses to the questionnaire, the interactive version was used as a control group when comparing with the Wilcoxon Signed Rank test with an alpha of 0.05.

Table 5.1: Questionnaire comparison: Mean values (positive for interactive version) and p values of Wilcoxon Signed Rank tests for comparing questions. Significant p values are set in **bold**.

Non-haptic	Non-interactive	Detached
Q1: I experienced that I was located at some distance behind a visual image of myself, almost as if I was looking at someone else.		
$M = -0.005; p = 0.12$	$M = -0.76; p = 0.24$	$M = 0.25; p = 1$
Q2: I felt that I had two bodies.		
$M = -0.05; p = 0.14$	$M = -1.14; p = 0.86$	$M = -0.14; p = 1$
Q3: I felt as if my head and body was at different locations, almost as if I had been 'decapitated'.		
$M = 0.2; p = 0.52$	$M = -0.14; p = 0.06$	$M = 0.44; p = 0.19$
Q4: I could no longer feel my body, it was almost as if it had disappeared.		
$M = -0.5; p = 0.96$	$M = -0.24; p = 0.55$	$M = -0.07; p = 1$
Q5: It seemed like I was in control of the water bubble.		
$M = -0.2; p = \mathbf{0.0004}$	$M = 4.27; p = 0.5$	$M = -0.32; p = 1$
Q6: It felt like I was able to steer the water bubble while falling.		
$M = 0.1; p = \mathbf{0.0004}$	$M = 4.36; p = 1$	$M = 0.03; p = 0.61$
Q7: I was able to merge the water drops with my hands.		
$M = 0.06; p = \mathbf{0.0008}$	$M = 3.98; p = 0.86$	$M = -0.53; p = 0.32$
Q8: I felt as if the virtual body was my body.		
$M = 0.17; p = \mathbf{0.0009}$	$M = 2.33; p = 0.46$	$M = -0.25; p = 0.35$
Q9: I could feel the water drops merging with my virtual body.		
$M = 0.52; p = \mathbf{0.02}$	$M = 1.19; p = 0.7$	$M = -0.4; p = \mathbf{0.04}$
Q10: I felt as if the water drops were my hands.		
$M = 0.56; p = \mathbf{0.0005}$	$M = 3.73; p = \mathbf{0.01}$	$M = 2.06; p = 0.16$
Q11: How much did you feel that you were inside a raindrop?		
$M = 0.01; p = 0.62$	$M = 0.09; p = 0.26$	$M = -0.16; p = 0.28$
Q12: How much did you feel that you were the raindrop yourself?		
$M = 0.12; p = \mathbf{0.004}$	$M = 1.46; p = 0.41$	$M = -0.46; p = 0.07$

Table 5.1 shows the mean (M) and p-values of every question from the questionnaire, with each comparison in an individual column. The mean shown is the mean score of the interactive version subtracted by the mean score of the version being compared with. Thus, a positive mean determines that the interactive version resulting in a higher mean score than the other version, and negative is the opposite results. However, due to variance (which is not shown), the magnitude of

the means does not correlate to a low p-value.

From the test results for comparing interactive with non-haptic, questions 5 to 10 and 12 shows a significant difference with questions 5 having a higher score in the non-haptic version, and questions 6 to 10 and 12 higher in the interactive version.

In the next column for comparing non-interactive, question 10 shows a significant difference, scoring higher in the interactive version.

In the last column for comparing detached, question 9 shows a significant difference with detached scoring higher.

5.5.2 Tracking Data

The recorded tracking data test, like the questionnaire test, sets the interactive version as the control group. The Welch T-test is used to compare the data with an alpha of 0.05. For this data, tests have been performed on the complete sets of recorded data, as well as subsets only including the collect phase.

Table 5.2: Head/hands movement/rotation in full experiment: Mean values (positive for interactive version) and p values of Welch T-tests for comparing data. Significant p values are set in **bold**.

Full experiment	Non-haptic	Detached	Non-interactive
Head movement	$M = 0.07; p = 0.67$	$M = 1.01; p = \mathbf{0.0002}$	$M = -1.04; p = 0.85$
Hand movement	$M = 0.03; p = 0.84$	$M = 1.05; p = \mathbf{0.0002}$	$M = -1.99; p = \mathbf{0.0002}$
Head rotation	$M = -0.66; p = 0.85$	$M = -3.73; p = 0.37$	$M = -2.72; p = 0.37$
Hand rotation	$M = -2.02; p = 0.59$	$M = 2.33; p = 0.94$	$M = 10.57; p = 0.13$

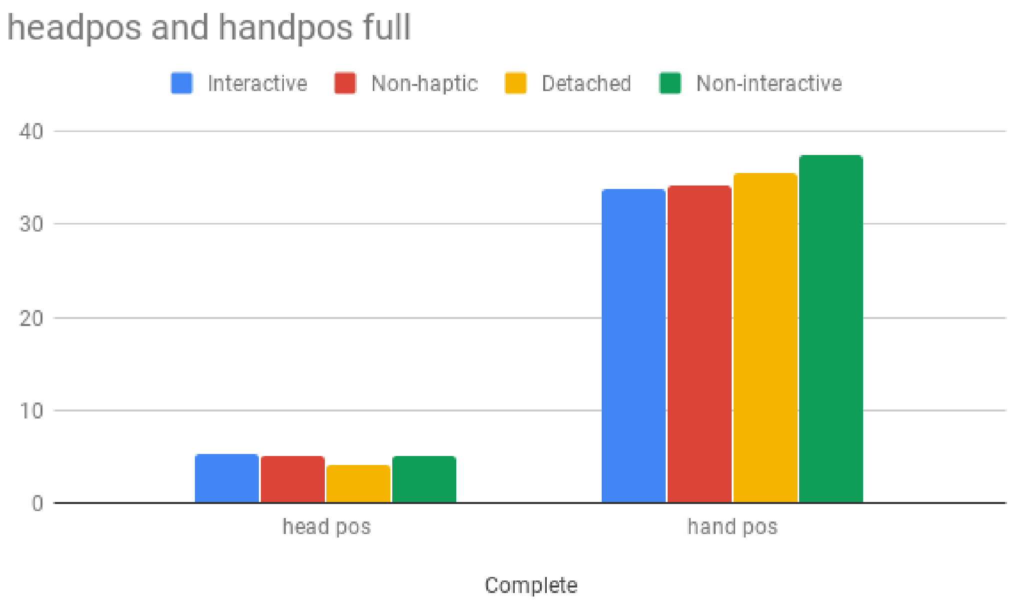


Figure 5.2: Changes in position of the full experiment.

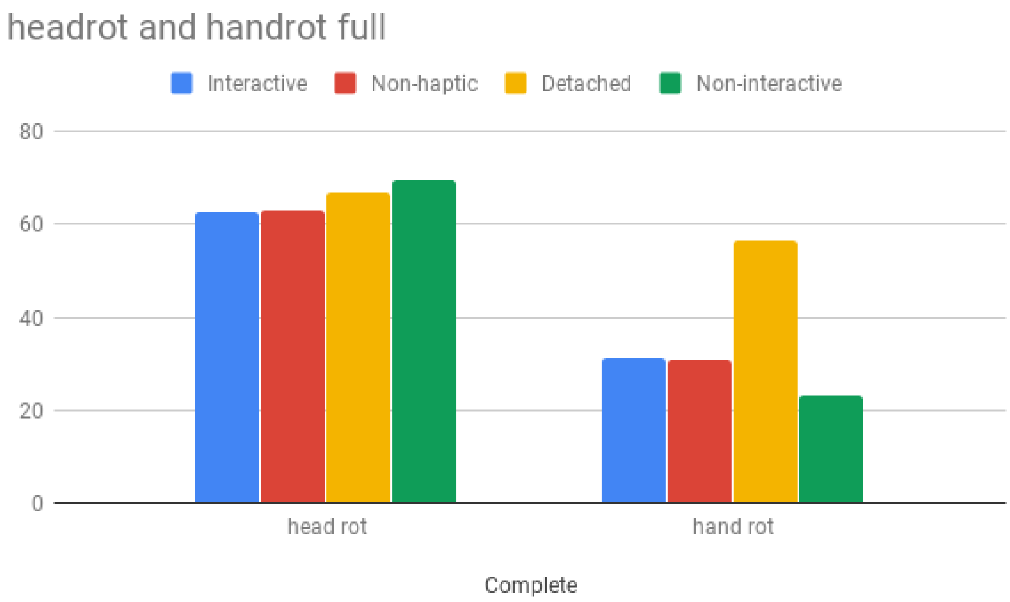


Figure 5.3: Changes in rotation of the full experiment.

In table 5.2, changes in position throughout the whole experience, results in a significant difference for the users' head and hands in the detached version, and hands in the non-interactive version. This shows decreased movement of the users' head and hands in the detached version, and increased movement in the non-interactive version of the users' hands.

For the rotation throughout the whole experience, no significant differences could be found.

Table 5.3: Head/hands movement/rotation in collect phase: Mean values (positive for interactive version) and p values of Welch T-tests for comparing data. Significant p values are set in **bold**.

Collect phase	Non-haptic	Detached	Non-interactive
Head movement	$M = -0.02; p = 0.15$	$M = -0.02; p = \mathbf{0.02}$	$M = 0.06; p = 0.3$
Hand movement	$M = -0.03; p = 0.17$	$M = -0.04; p = \mathbf{0.04}$	$M = 0.11; p = 0.28$
Head rotation	$M = -7.91; p = \mathbf{0.03}$	$M = -10.36; p = \mathbf{0.002}$	$M = 9; p = 0.07$
Hand rotation	$M = -2.76; p = 0.53$	$M = -2.4; p = 0.53$	$M = 5.35; p = 0.98$

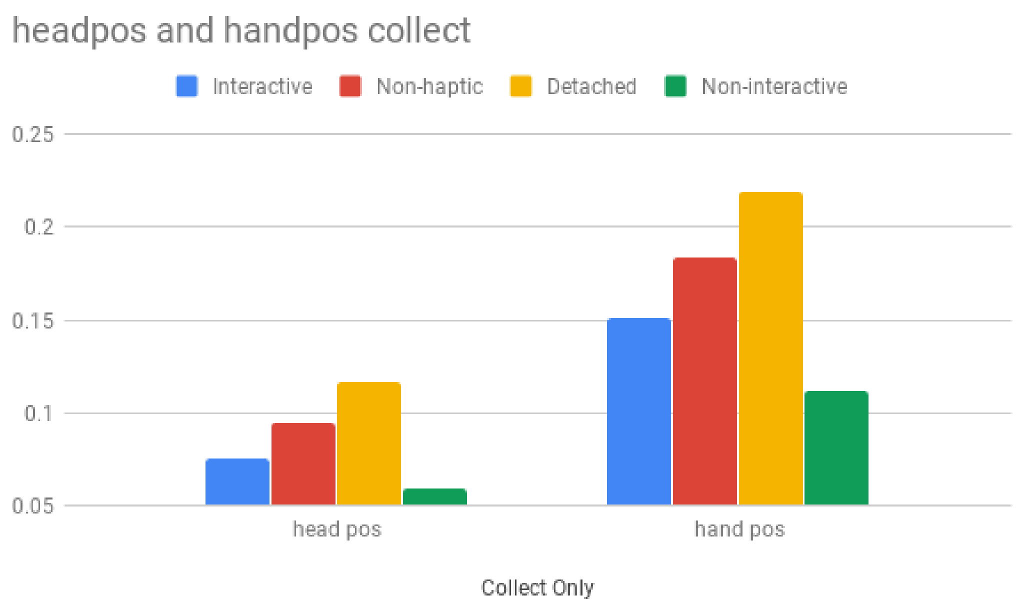


Figure 5.4: Changes in position of the collect phase of the experiment.

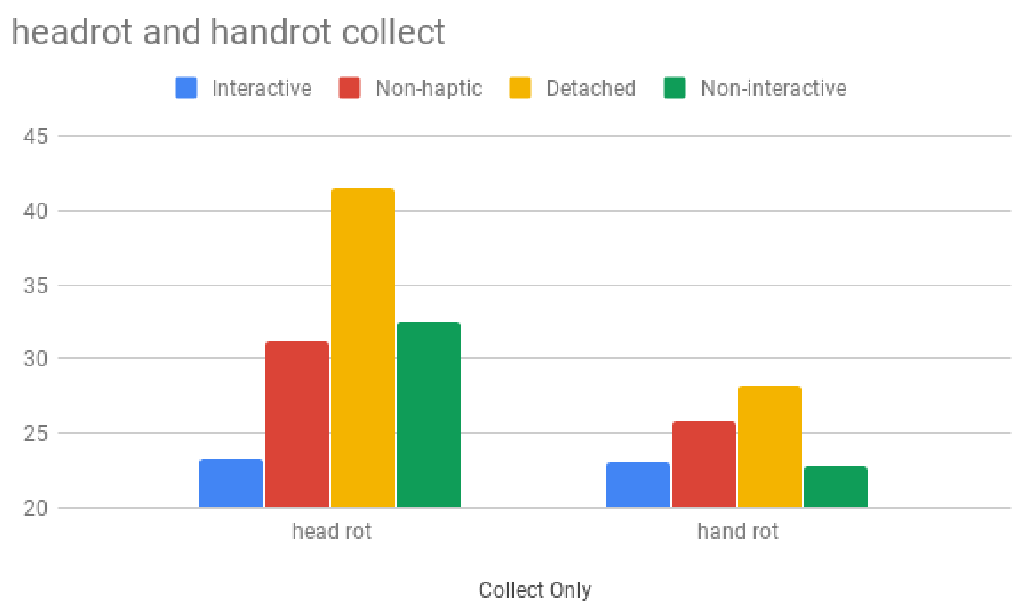


Figure 5.5: Changes in rotation of the collect phase of the experiment.

Testing the subset of data from the collect phase, changes in position again results in a significant difference for the detached version's head and hands, and hands in the non-interactive version, as can be seen in table 5.3.

Different from the previous test of the whole dataset, this shows that an increase in the user's head and hand movement in the detached version, and a decrease of the hand movement in the non-interactive version. In the subset of rotations, a significant difference can be found in the head rotations of the non-haptic and detached version. This shows an increase in the users' head rotations during both versions.

Table 5.4: Time: Mean values (positive for interactive version) and p values of Welch T-tests for comparing data. Significant p values are set in **bold**.

Time	Non-haptic	Detached
Total time	$M = 0.07; p = 0.67$	$M = 1.01; p = \mathbf{0.0002}$
Collect time	$M = 0.03; p = 0.84$	$M = 1.05; p = \mathbf{0.0002}$

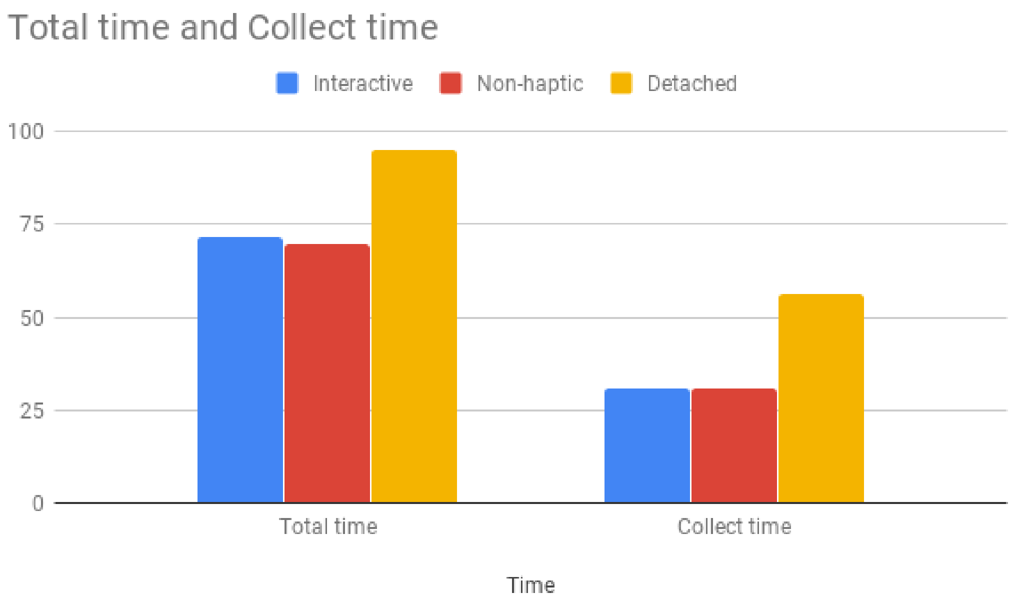


Figure 5.6: Total time taken during the experiment and time taken to finish the collect phase.

For testing the total time taken and time taken in the collect phase, the non-interactive version have been excluded as the version is constant here.

The test results in significant differences for the detached version, for both the total time taken and the time taken in the collect phase. Both results show an increase for the detached version, as seen in table 5.4.

Chapter 6

Discussion

This chapter is the discussion of the project. Following are discussions surrounding the evaluation, experiment, and the results, followed by weaknesses encountered and how the experiment could be improved.

6.1 Discussion of the Results

The following section discusses the results from the experiment in relation to the hypotheses.

H1: Interaction in a VR experience increases the user's sense of embodiment.

For the first hypothesis regarding the sense of embodiment in an interactive and non-interactive experience.

While questions related to self-location did not give significant results, question 3 resulted in a p-value of 0.06, though not low enough to consider significant, it is close and should be noted. A trend can be seen of the non-interactive version being rated higher, which can be described as a decrease in self-location due to the formulation of the questions.

Questions relating to agency did not show any significant results, but the means show a trend of the non-interactive version being rated lower by means of 4.27, 4.36, and 3.98. This could suggest the users are having a lower agency in the non-interactive version, than the interactive version.

The question in the questionnaire *"I felt as if the water drops were my hands"* was rated lower in the non-interactive version with a difference in means of 3.7. This question relates to body ownership, and a higher rating for these questions means increased body ownership.

While the other questions related to body ownership does not have significant results, they do show a tendency of users rating the non-interactive version lower.

The results of testing the users' physical movement shows that during the whole experiment, the significant results show that the users moved their hands more in the non-interactive version. Head movement with a high probability of being random, also shows more movement.

The rotation of the users' head also sees an increase, though not significant. However, while rotations of the hands were not significant, the mean suggests that users rotated their hands less.

When narrowing the dataset down to only test the collect phase, the results changes to show no significance. The means suggests that during the collect phase of the non-interactive version, the movement and rotation of head and hands decreased.

As the time the experience takes cannot be affected by the user in the non-interactive version, it cannot yield any results.

Overall, the results of the non-interactive version suggest a decrease in body ownership and is leaning towards a decrease in self-location and agency. And while the users' show more movement and rotation, the data is not significant, thus does not show any clear changes in core self-presence.

H2: Attachment to objects of the virtual environment in a VR experience increases the user's sense of embodiment.

The second hypothesis concerns itself with seeing an increase in the sense of embodiment when the user has an attachment to objects of the virtual environment.

For the questions related to self-location, no significant results are found. The means for each question leans in the direction of both versions, thus there is no bias to be suggested.

Questions relating to agency also has no significant results, and the means leans in both directions, giving nothing to suggest.

One question relating to body ownership shows a significant result. The question with a significant result is question 9: *"I could feel the water drops merging with my virtual body."*, which was rated to be lower in the detached version, with a difference in means of -0.4.

Three other questions of body ownership were rated lower as well, and one was rated higher, which may suggest that the detached version sees a decrease in body ownership.

Of the whole experiment, head and hand movement gave significant results, with a decrease in the detached version. While not significant, the mean of head rotations shows an increase, and the hand rotations show a decrease.

For the collect phase of the experiment, the movements of the head and hands remains significant, however, they show an increase in the detached version instead of a decrease. The head rotation becomes significant with an increase in the detached version. While not significant, the hand rotation sees a decrease.

The time taken for the detached version give significant results, showing an increase in the total time taken and the time taken for the collect phase.

The results show a possible increase in body ownership and may suggest an increase in movement and time taken. One reason why the increase in movement and time taken exists, could be due to the difference in the design of the interaction between the interactive version and the detached version. Thus, it is unclear if this suggest an increase in core self-presence.

H3: Haptic feedback in a VR experience increases the user's sense of embodiment.

The third hypothesis concerns itself with whether the implementation of haptic feedback increases sense of embodiment.

Questions related to self-location does not give any significant results. However, three of four of the questions lean towards the non-haptic version having a higher rating. Thus, may be due to a possible trend of decreased self-location.

All questions relating to agency gives significant results. Question 5: *"It seemed like I was in control of the water bubble."* was rated higher in the non-haptic version. Question 6: *"It felt like I was able to steer the water bubble while falling."* and question 7: *"I was able to merge the water drops with my hands"* was rated higher in the interactive version.

Due to there not being any difference between the two versions in the fall phase, questions 5 and 6 may not have much relevance in this comparison, leaving question 7 to suggest a decrease in agency in the non-haptic version.

For questions relating to body ownership, four of five gives significant results. Question 8: *"I felt as if the virtual body was my body"*, question 9: *"I could feel the water drops merging with my virtual body."*, questions 10: *"I felt as if the water drops were my hands."*, and question 12: *"How much did you feel that you were the raindrop yourself?"* were all significant and rated lower in the non-haptic version.

The remaining question that did not give significant results was also shown to be rated lower in the non-haptic version on average. This suggests the non-haptic version has a decrease in body ownership.

For the whole experiment, head and hands movement and rotation did not yield significant result. The means may suggest a possible decrease in movement and increase in rotation. For the collect phase, head rotation shows a significant results, which is an increase, while hand rotation, and head and hand movement means suggests an increase as well.

Time taken during the experiment does not have a significant difference, and the means are very similar.

The non-haptic version suggests decreases in agency and body ownership. Due to the significant increase in head rotation, and non-significant increase in hand rotation, and head and hand movement, is could suggest an increase in core self-presence.

6.2 Quality of the Solution

This section focuses on the quality of the VR experience and discuss how aspects of the VR experience could influence the experiment.

The fall phase did not contain any difference between the interactive, non-haptic, and detached versions. Only in the non-interactive version was it different. This may have made the fall phase less relevant for the experiment, and thus have changed the participants experience of the collect phase.

The interaction part of the collect phase in the detached version may have had more of an impact just by being different. Participants have been observed to play around with water droplets more, which may explain why or play as a factor in participants taking longer with the detached version.

6.3 Analysis of the Evaluation Method

This section is an analysis of the method used for evaluating the project, highlighting weaknesses and presents possible improvements.

To ready the VR experience for the experiment, internal testing was conducted in order to find and fix issues. These issues varied from minor to severe, and the priority was only on issues that would cause an effect to the experiment. Minor issues could be viewed as irrelevant if they did not influence the experiment.

After internal testing a pilot test was conducted using an outside participant. The pilot test was conducted the same as the experiment in order to find shortcomings and gain feedback in the structure of the experiment and the VR experience. This feedback was then taken into considerations and changes relevant to the experiment took place.

The tutorial would function as the first version of the game that the participants would encounter, its purpose was to ensure that the participants had an idea about the objectives of the tests to come. The tutorial itself was much like the Interactive version with instructions in the form of extended text. Without the tutorial the project could risk people spending time learning how to operate the experience instead of testing.

The questionnaire contains 12 questions with 7-point Likert scales that can be divided into three subcategories. The participants would have to answer the questionnaire after each version which results in 48 questions per participant. It is difficult to determine how many questions are enough, too many questions could potentially make the participant lose interest and give invalid data, too few could lack statistical significance.

During the experiment, the participants was required to answer the questionnaire. This meant they would have to remove all the VR equipment in order to do so, and then put it all back on when they were ready to continue to the next test. This could be at the very least viewed as an annoyance and possible disrupt the immersion of the participants.

6.3.1 Validity and Reliability

The experiment that was carried out had 17 participants. This is an acceptable sample size, however, with a larger sample size the results may have been more reliable, thus may have been slightly different.

Due to the experiment being a within-subject design, the results may have been affected by the order of the versions they tested. Therefore, it could yield more valid results if the experiment was changed to a between-subject design.

A mistake was made with the recording of the movements of the user. The tracked changes in position was tracked in world-space, thus it did not consider the fall, adding additional change to the position. This did not affect the tracked changes in rotation.

6.4 Wider Context

This section presents solutions to some of the problems found in the current experiment and additional possible developments for this project.

The fall phase could get more implementations relating to the version of the experience, so it could be a separate interaction phase that could be tested on its own, like the collect phase. This could for example be haptic feedback, there would be an absence of haptic feedback in the non-haptic version as a difference in the fall phase, making it more relevant.

It is not clear whether the data of the users' changes in position and rotation gave valid results due to the circumstance of different interaction design in the interactive and detached versions.

More data could have been gathered from the participants by asking them to rank each version. However, this could have proven to be difficult to do as the participants cannot be expected to remember each version.

As mentioned, during the experiment participants would be required to remove VR equipment to answer the questionnaire in between each test. This could be avoided by adding a questionnaire in the VR environment, allowing the participant to answer the questionnaire without leaving VR. This may have a positive impact on the participants retaining their experience. However, it is also possible that in some cases of long tests, it would otherwise be a positive to let the participants have a break.

To address the difference in the interaction within the interactive version and the detached version, and how this difference could unintentionally have an effect, the detached version could be redesigned. Instead of having the participant grab water droplets, the detached version could be almost identical with the interactive version, main difference being the hands of the participant. In the interactive version, the participant's hands appear as water droplets, thus in the detached version, the only difference could be made to be the hands appearing as controllers or human hands.

Other aspects that can be tested, such as auditory feedback and visual effects. As there are many more aspects to a VR experience than interaction, haptic feedback, and the user's attachment to the virtual environment, more versions could be tested and potentially show more results.

Auditory feedback could be a version, where the difference from the control version would be the lack of auditory feedback from merging water droplets. This could also include ambient audio either in the same version or as a separate version.

Visuals is also an aspect that can be tested. Visuals are a wide subject, thus there may be the need to isolate visual effects or categorise them, in order to retain control of the experiment.

Chapter 7

Conclusion

This chapter summarises the project and concludes what was found based on results regarding the problem formulation and hypotheses.

The problem formulation of this project is "How can the sense of embodiment be affected in a VR experience". To investigate this, hypotheses were created, and a VR experience was developed in order to test these hypotheses.

The VR experience had multiple versions implemented. First a standard version called the interactive version, which had all aspects of the experience implemented. Other versions of the experience had aspects absent, such as haptic feedback, interaction, and a version with another type of interaction where the user finds themself less attached to the environment.

An experiment was conducted to find if the absence of these aspects would increase the sense of embodiment within the user, by investigating subcomponents of sense of embodiment, self-location, agency, and body ownership. Additionally, through analysis of the results, conclusions could be made regarding the hypotheses.

H1: Interaction in a VR experience increases the user's sense of embodiment.

It was found that participants in the version without interaction had a decrease in body ownership and other results that could suggest there may also be a decrease in self-location and agency. Thus, H1 is accepted.

H2: Attachment to objects of the virtual environment in a VR experience increases the user's sense of embodiment.

The detached version of the VR experience found that the participants had an increase in body ownership. Thus, the null hypotheses of H2: "Attachment to objects of the virtual environment in a VR experience does not increase the user's sense of embodiment." cannot be rejected.

To speculate on this, this may have been due to the designed interaction of the detached version, which could have led the participants to an increased sense of embodiment. It is difficult to be exact in why this may be the case, whether it is the user being able to see the controllers or grab and move raindrops, but this is a reason to reconsider the concept of the detached version. Regarding future projects on the subject, this form of interaction could be investigated in more detail in order to explain the effects it has on the user's sense of embodiment.

H3: Haptic feedback in a VR experience increases the user's sense of embodiment.

In the version with no haptic feedback the participants are shown to have a decrease in both agency and body ownership. Thus, it is concluded that H3 is accepted.

To conclude this project, some aspects in a VR experience such as interaction and haptic feedback does show an increase in sense of embodiment in the user. Further investigation into aspects such as these and more, may help the development of VR experiences intending a high sense of embodiment for their users.

Bibliography

[1] Woodrow Barfield and Claudia Hendrix. “The effect of update rate on the sense of presence within virtual environments”. In: *Virtual Reality* 1.1 (June 1995), pp. 3–15. ISSN: 1359-4338, 1434-9957. DOI: 10.1007/BF02009709. URL: http://link.springer.com/10.1007/BF02009709 (visited on 05/17/2019).

[2] Gary Charness, Uri Gneezy, and Michael A. Kuhn. “Experimental methods: Between-subject and within-subject design”. In: *Journal of Economic Behavior & Organization* 81.1 (Jan. 2012), pp. 1–8. ISSN: 01672681. DOI: 10.1016/j.jebo.2011.08.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167268111002289 (visited on 05/17/2019).

[3] *Computer Graphics Group » Resources*. URL: https://graphics.create.aau.dk/resources-2/ (visited on 05/17/2019).

[4] Marcello Costantini and Patrick Haggard. “The rubber hand illusion: Sensitivity and reference frame for body ownership”. In: *Consciousness and Cognition* 16.2 (June 2007), pp. 229–240. ISSN: 10538100. DOI: 10.1016/j.concog.2007.01.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S1053810007000037 (visited on 05/17/2019).

[5] Marco D’Alonzo, Francesco Clemente, and Christian Cipriani. “Vibrotactile Stimulation Promotes Embodiment of an Alien Hand in Amputees With Phantom Sensations”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 23.3 (May 2015), pp. 450–457. ISSN: 1534-4320, 1558-0210. DOI: 10.1109/TNSRE.2014.2337952. URL: http://ieeexplore.ieee.org/document/6856175/ (visited on 05/17/2019).

[6] Jonathan Freeman et al. “Effects of Sensory Information and Prior Experience on Direct Subjective Ratings of Presence”. In: *Presence: Teleoperators and Virtual Environments* 8.1 (Feb. 1999), pp. 1–13. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/105474699566017. URL: http://www.mitpressjournals.org/doi/10.1162/105474699566017 (visited on 05/17/2019).

[7] Joel R. Gat. *Isotope Hydrology: A Study of the Water Cycle*. Singapore, SINGAPORE: Imperial College Press, 1998. ISBN: 978-1-84816-474-1. URL: http:

//ebookcentral.proquest.com/lib/aalborguniv-ebooks/detail.action?docID=731265 (visited on 05/17/2019).

[8] *Gradient Noise Node | Package Manager UI website*. URL: https://docs.unity3d.com/Packages/com.unity.shadergraph@6.5/manual/Gradient-Noise-Node.html (visited on 05/27/2019).

[9] Claudia Hendrix and Woodrow Barfield. "The Sense of Presence within Auditory Virtual Environments". In: *Presence: Teleoperators and Virtual Environments* 5.3 (Jan. 1996), pp. 290–301. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/pres.1996.5.3.290. URL: http://www.mitpressjournals.org/doi/10.1162/pres.1996.5.3.290 (visited on 05/17/2019).

[10] Konstantina Kilteni, Raphaela Groten, and Mel Slater. "The Sense of Embodiment in Virtual Reality". In: *Presence: Teleoperators and Virtual Environments* 21.4 (Nov. 2012), pp. 373–387. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/PRES_a_00124. URL: http://www.mitpressjournals.org/doi/10.1162/PRES_a_00124 (visited on 05/17/2019).

[11] Brenda Laurel. "What Is Virtual Reality?" In: (2016). DOI: 10.13140/rg.2.1.4415.0643. URL: http://rgdoi.net/10.13140/RG.2.1.4415.0643 (visited on 05/17/2019).

[12] Jane Lessiter et al. "A Cross-Media Presence Questionnaire: The ITC-Sense of Presence Inventory". In: *Presence: Teleoperators and Virtual Environments* 10.3 (June 2001), pp. 282–297. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/105474601300343612. URL: http://www.mitpressjournals.org/doi/10.1162/105474601300343612 (visited on 05/17/2019).

[13] Matthew R. Longo et al. "What is embodiment? A psychometric approach". In: *Cognition* 107.3 (June 2008), pp. 978–998. ISSN: 00100277. DOI: 10.1016/j.cognition.2007.12.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0010027708000061 (visited on 05/17/2019).

[14] *Low Poly Pack - Environment Lite - Asset Store*. URL: https://assetstore.unity.com/packages/3d/props/exterior/low-poly-pack-environment-lite-102039 (visited on 05/21/2019).

[15] Rocci Luppicini, ed. *Handbook of Research on Technoself: Identity in a Technological Society*. IGI Global, 2013. ISBN: 978-1-4666-2211-1 978-1-4666-2212-8. DOI: 10.4018/978-1-4666-2211-1. URL: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2211-1 (visited on 05/17/2019).

[16] Michael Meehan et al. "Physiological measures of presence in stressful virtual environments". In: *ACM Transactions on Graphics* 21.3 (July 1, 2002). ISSN: 07300301. DOI: 10.1145/566654.566630. URL: http://portal.acm.org/citation.cfm?doid=566654.566630 (visited on 05/17/2019).

[17] *PostFX v2 – Amazing visuals, upgraded – Unity Blog*. Unity Technologies Blog. URL: https://blogs.unity3d.com/2018/06/11/postfx-v2-amazing-visuals-upgraded/ (visited on 05/21/2019).

[18] M. S. Prewett et al. "A Meta-Analysis of Vibrotactile and Visual Information Displays for Improving Task Performance". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.1 (Jan. 2012), pp. 123–132. ISSN: 1094-6977, 1558-2442. DOI: 10.1109/TSMCC.2010.2103057. URL: http://ieeexplore.ieee.org/document/5710431/ (visited on 05/17/2019).

[19] *ProBuilder*. URL: https://unity3d.com/unity/features/worldbuilding/probuilder (visited on 05/21/2019).

[20] Giuseppe Riva. "Is presence a technology issue? Some insights from cognitive sciences". In: *Virtual Reality* 13.3 (Sept. 2009), pp. 159–169. ISSN: 1359-4338, 1434-9957. DOI: 10.1007/s10055-009-0121-6. URL: http://link.springer.com/10.1007/s10055-009-0121-6 (visited on 05/17/2019).

[21] Maria V Sanchez-Vives and Mel Slater. "From Presence Towards Consciousness". In: (), p. 34.

[22] Ulrike Schultze. "Embodiment and presence in virtual worlds: a review". In: *Journal of Information Technology* 25.4 (Dec. 2010), pp. 434–449. ISSN: 0268-3962, 1466-4437. DOI: 10.1057/jit.2010.25. URL: http://link.springer.com/10.1057/jit.2010.25 (visited on 05/17/2019).

[23] Mel Slater and Martin Usoh. "Representations Systems, Perceptual Position, and Presence in Immersive Virtual Environments". In: *Presence: Teleoperators and Virtual Environments* 2.3 (Jan. 1993), pp. 221–233. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/pres.1993.2.3.221. URL: http://www.mitpressjournals.org/doi/10.1162/pres.1993.2.3.221 (visited on 05/17/2019).

[24] Mel Slater, Martin Usoh, and Anthony Steed. "Depth of Presence in Virtual Environments". In: *Presence: Teleoperators and Virtual Environments* 3.2 (Jan. 1994), pp. 130–144. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/pres.1994.3.2.130. URL: http://www.mitpressjournals.org/doi/10.1162/pres.1994.3.2.130 (visited on 05/17/2019).

[25] Mel Slater, Martin Usoh, and Anthony Steed. "Taking steps: the influence of a walking technique on presence in virtual reality". In: *ACM Transactions on Computer-Human Interaction* 2.3 (Sept. 1, 1995), pp. 201–219. ISSN: 10730516. DOI: 10.1145/210079.210084. URL: http://portal.acm.org/citation.cfm?doid=210079.210084 (visited on 05/17/2019).

[26] Marcos Slomp et al. "Photorealistic real-time rendering of spherical raindrops with hierarchical reflective and refractive maps". In: *Computer Animation and Virtual Worlds* 22.4 (2011), pp. 393–404. ISSN: 1546-427X. DOI: 10.1002/cav.421. URL: https://www.onlinelibrary.wiley.com/doi/abs/10.1002/cav.421 (visited on 05/17/2019).

[27] *SteamVR Plugin - Asset Store*. URL: https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647 (visited on 05/17/2019).

[28] Unity Technologies. *Unity*. Unity. URL: https://unity.com/frontpage (visited on 05/17/2019).

[29] *The Stanford Ocean Acidification Experience*. VHIL. URL: https://vhil.stanford.edu/soae/ (visited on 05/17/2019).

[30] *The Stanford Ocean Acidification Experience på Steam*. URL: https://store.steampowered.com/app/409020/The_Stanford_Ocean_Acidification_Experience/ (visited on 05/17/2019).

Appendix A

Shader Code

This appendix contains the full shader code of the following shaders implemented in the project:

- Water droplet shader (WaterRefractionMerge.shader).
- Big water bubble shader (BigWaterBubbleShader.shader).
- Cloud shader (NewCloudShader.shader).

```
Shader "Custom/WaterRefractionMerge"
{
    Properties
    {
        _Color ("Color", Color) = (1,1,1,1)
        _Refraction("Refraction Value", Float) = 1.5

        [HideInInspector]_SphereRadius("Object's radius", Float) = 0
        _MergeDistance("Object's radius merge distance", Range(0,1)) = 0.5
        [HideInInspector]_NearSpherePos("Nearest sphere's position", Vector) =
    (0,0,0,0)
        [HideInInspector]_NearSphereRadius("Nearest sphere's radius", Float) = 0

        [Toggle]_Refract("Refract Active", Range(0,1)) = 1
        [Toggle]_Simple("Simple Mode", Range(0,1)) = 0
    }
    SubShader
    {
        Tags { "RenderType"="Opaque" }
        LOD 200
        Cull Back

        CGPROGRAM

        #pragma surface surf Standard vertex:vert finalcolor:frag

        #pragma target 3.0

        struct Input
```

```
        {
            float3 viewDir;
            float3 worldNormal; INTERNAL_DATA
        };

        fixed4 _Color;
        float _Refraction;
        float _SphereRadius;
        float _MergeDistance;
        float3 _NearSpherePos;
        float _NearSphereRadius;
        int _Refract;
        int _Simple;

        float inverseSmooth (float a, float b, float x){
            float t = saturate((x - a)/(b-a));
            return t + (t - (t * t * (3.0 - 2.0 * t)));
        }

        float expSmooth(float a, float b, float x){
            float t = saturate((x - a)/(b-a));
            return pow(t,2);
        }

        void vert (inout appdata_full v){
            if (_Simple == 1) return;

            float d = dot(v.normal, normalize(_NearSpherePos));
            float3 dir = normalize(_NearSpherePos);
            if (length(_NearSpherePos) == 0) dir = float3(0,0,0);
            float r1 = length(v.vertex.xyz);
            float r2 = (_NearSphereRadius/_SphereRadius) * r1;

            float maxDistance = (r1 * (1 + _MergeDistance)) + (r2 * (1 + _MergeDistance));
            float distanceFactor = max(0, maxDistance - length(_NearSpherePos))/maxDistance; // range 0-1, higher is closer

            float3 reflectedNormal = reflect(normalize(v.vertex.xyz), normalize(-_NearSpherePos));
            float3 reflectedVertex = (reflectedNormal * r2) + _NearSpherePos;
            if (length(_NearSpherePos) != 0) {
                float3 newPos = lerp(v.vertex.xyz, reflectedVertex, 0.55f * expSmooth(0, 1-d, distanceFactor) * step(0, d));
                if (length(newPos) > r1) v.vertex.xyz = newPos;
            }

            if (length(_NearSpherePos) != 0){
                float3 newNormal = lerp(v.normal, reflect(v.normal, normalize(-_NearSpherePos)), 0.5f * expSmooth(0, 1-d, distanceFactor) * step(0, d));
                v.normal = newNormal;
            }

        }

        void frag(Input input, SurfaceOutputStandard o, inout fixed4 color)
        {
            if (_Simple == 1) return;

```

```
            float refractiveIndex = 1.5;
            refractiveIndex = _Refraction;
            bool refractMirror = (_Refract > 0.5 ? true : false);
            float3 divergeDir = refract(-input.viewDir, -input.worldNormal, 1.0 /
    refractiveIndex);
            float3 refractedDir = refract(input.viewDir, input.worldNormal, 1.0 /
    refractiveIndex);
            half4 refCube = UNITY_SAMPLE_TEXCUBE(unity_SpecCube0, refractMirror ?
    refractedDir : divergeDir);
            color = refCube + refCube * _LightColor0;
        }

        void surf (Input IN, inout SurfaceOutputStandard o)
        {
            fixed4 c = _Color;
            float dissimilarity = smoothstep(0, 1, 1 - abs(dot(IN.viewDir, IN.
    worldNormal)));
            o.Albedo = c.rgb;
            o.Alpha = c.a;
        }
        ENDCG
    }
    FallBack "Diffuse"
}
```

code/WaterRefractionMerge.shader

```
Shader "Custom/BigWaterBubbleShader"
{
    Properties
    {
        _Color ("Color", Color) = (1,1,1,1)
        _MorphDirection ("Morph Direction", Vector) = (0,0,0,0)
        [Toggle]_Simple ("Simple Mode", Range(0,1)) = 0
    }
    SubShader
    {
        Tags { "RenderType"="Transparent" "Queue"="Transparent" }
        LOD 200
        Cull front

        CGPROGRAM

        #pragma surface surf Standard vertex:vert alpha:blend

        #pragma target 3.0

        struct Input
        {
            float2 uv_MainTex;
            float3 viewDir;
            float3 worldNormal; INTERNAL_DATA
            float3 worldPos;
        };

        fixed4 _Color;
        float3 _MorphDirection;
        int _Simple;

```

```
        float2 unity_gradientNoise_dir(float2 p)
        {
            p = p % 289;
            float x = (34 * p.x + 1) * p.x % 289 + p.y;
            x = (34 * x + 1) * x % 289;
            x = frac(x / 41) * 2 - 1;
            return normalize(float2(x - floor(x + 0.5), abs(x) - 0.5));
        }

        float unity_gradientNoise(float2 p)
        {
            float2 ip = floor(p);
            float2 fp = frac(p);
            float d00 = dot(unity_gradientNoise_dir(ip), fp);
            float d01 = dot(unity_gradientNoise_dir(ip + float2(0, 1)), fp - float2(0, 1));
            float d10 = dot(unity_gradientNoise_dir(ip + float2(1, 0)), fp - float2(1, 0));
            float d11 = dot(unity_gradientNoise_dir(ip + float2(1, 1)), fp - float2(1, 1));
            fp = fp * fp * fp * (fp * (fp * 6 - 15) + 10);
            return lerp(lerp(d00, d01, fp.y), lerp(d10, d11, fp.y), fp.x);
        }

        float Unity_GradientNoise_float(float2 UV, float Scale)
        {
            return unity_gradientNoise(UV * Scale) + 0.5;
        }

        void vert (inout appdata_base v){
            if (_Simple == 1) return;

            float velD = dot(v.normal, _MorphDirection);
            float3 velDeform = _MorphDirection * max(0, velD);
            v.vertex.xyz += velDeform * 0.5;
        }

        void surf (Input IN, inout SurfaceOutputStandard o)
        {
            float3 noiseVector = IN.worldPos;
            float3 dir;
            if (any(_MorphDirection)) dir = normalize(_MorphDirection);
            else dir = float3(0,-1,0);
            float d = dot(dir, IN.worldNormal);
            float offsetNoise = Unity_GradientNoise_float(IN.worldPos.xz + IN.worldPos.xy + IN.worldPos.yz, 5);
            float offset = _Time.y * 0.1 + offsetNoise * .05;

            float2 noisePos = float2(noiseVector.y + offset, noiseVector.y + offset);
            float2 noisePos2 = float2(length(dir + IN.worldNormal) + offset, length(dir + IN.worldNormal) + offset);
            float intensity = 3;
            float noise = Unity_GradientNoise_float(noisePos2, intensity);

            fixed4 c = _Color * smoothstep(0, 1, length(noise));
            float a = smoothstep(.4, .6, length(noise));

            if (_Simple == 1){
```

```
                o.Albedo = _Color.rgb;
                o.Alpha = _Color.a;
            }
            else {
                o.Albedo = abs(noise) * _Color;
                o.Alpha = saturate(lerp(_Color.a, 0.75, a) * (1 - smoothstep(.7,1,
    abs(d))));
            }
        }
        ENDCG
    }
    FallBack "Diffuse"
}
```

code/BigWaterBubbleShader.shader

```
Shader "Custom/NewCloudShader"
{
    Properties
    {
        _Color ("Color", Color) = (1,1,1,1)
        _RimColor ("Rim Color", Color) = (1,1,1,1)
        _PerlinIntensity("Perlin Intensity", float) = 1.0
        _PerlinVertexScale("Perlin Vertex Scale", float) = 1.0
        _PerlinAlpha("Perlin Alpha Intensity", float) = 1.0
        _TimeScale ("Time Scale", float) = 1.0
        _FresnelMin ("Fresnel Min", Range(0,1)) = 0.1
        _FresnelMax ("Fresnel Max", Range(0,1)) = 0.5
    }
    SubShader
    {
        Tags { "RenderType"="Transparent" "Queue"="Transparent" "IgnoreProjector"="
    False" }
        LOD 200
        Cull back
        ZWrite off

        CGPROGRAM
        #pragma surface surf Standard vertex:vert alpha:blend

        #pragma target 3.0

        struct Input
        {
            float3 viewDir;
            float3 worldNormal;
            float4 color : COLOR;
            float3 worldPos;
        };

        fixed4 _Color;
        fixed4 _RimColor;
        float _PerlinIntensity;
        float _PerlinVertexScale;
        float _PerlinAlpha;
        float _TimeScale;
        float _FresnelMin;
        float _FresnelMax;

```

```
        float2 unity_gradientNoise_dir(float2 p)
        {
            p = p % 289;
            float x = (34 * p.x + 1) * p.x % 289 + p.y;
            x = (34 * x + 1) * x % 289;
            x = frac(x / 41) * 2 - 1;
            return normalize(float2(x - floor(x + 0.5), abs(x) - 0.5));
        }

        float unity_gradientNoise(float2 p)
        {
            float2 ip = floor(p);
            float2 fp = frac(p);
            float d00 = dot(unity_gradientNoise_dir(ip), fp);
            float d01 = dot(unity_gradientNoise_dir(ip + float2(0, 1)), fp - float2
    (0, 1));
            float d10 = dot(unity_gradientNoise_dir(ip + float2(1, 0)), fp - float2
    (1, 0));
            float d11 = dot(unity_gradientNoise_dir(ip + float2(1, 1)), fp - float2
    (1, 1));
            fp = fp * fp * fp * (fp * (fp * 6 - 15) + 10);
            return lerp(lerp(d00, d01, fp.y), lerp(d10, d11, fp.y), fp.x);
        }

        float Unity_GradientNoise_float(float2 UV, float Scale)
        {
            return unity_gradientNoise(UV * Scale) + 0.5;
        }


        void vert (inout appdata_full v){
            float3 perlinValue;
            float offset = _Time.y * _TimeScale;
            float4 worldPos = mul(unity_ObjectToWorld, v.vertex);
            perlinValue.x = Unity_GradientNoise_float(worldPos.xy + float2(offset,
    0), _PerlinIntensity) - 0.5;
            perlinValue.y = Unity_GradientNoise_float(worldPos.xz + float2(0,
    offset), _PerlinIntensity) - 0.5;
            perlinValue.z = Unity_GradientNoise_float(worldPos.yz + float2(offset,
    0), _PerlinIntensity) - 0.5;
            v.vertex.xyz += perlinValue * _PerlinVertexScale * clamp(0, 1, dot(
    perlinValue, v.vertex.xyz));
            v.normal = normalize(v.vertex.xyz);
            v.color.rgb = perlinValue;
        }

        void surf (Input IN, inout SurfaceOutputStandard o)
        {
            float d = dot(IN.worldNormal, normalize(IN.viewDir));
            float3 perlinValue;
            float offset = _Time.y * _TimeScale * 5;
            float3 noiseVector = IN.worldPos;
            float newScale = _PerlinAlpha;
            perlinValue.x = Unity_GradientNoise_float(noiseVector.xy + float2(
    offset, 0), newScale);
            perlinValue.y = Unity_GradientNoise_float(noiseVector.xz + float2(0,
    offset), newScale);
            perlinValue.z = Unity_GradientNoise_float(noiseVector.yz + float2(0, 0)
    , newScale);
```

```
            perlinValue = perlinValue * 2 - float3(1,1,1);

            float fresnel = smoothstep(_FresnelMin, _FresnelMax, d);
            float noise = smoothstep(0, 1, length(perlinValue)) * smoothstep(0.1,
    1, d);
            fixed4 c = _Color;
            o.Albedo = c.rgb;
            o.Alpha = smoothstep(0, .5, saturate(fresnel + noise));
            o.Emission = smoothstep(.25, 1, 1 - d) * _RimColor * o.Alpha;
        }
        ENDCG
    }
    FallBack "Transparent/VertexList"
}
```

code/NewCloudShader.shader

Appendix B

Data

This appendix contains the data collected from the experiment.

B.1 Questionnaire Data

Questionnaire demographic					
	Mean	**Variance**			
Age	25.16666667	2.617647059			
	Male	**Female**			
Gender	15	3			
	Yes	**No**			
Have you ever experienced motion sickness? When driving, sailing etc.	15	3			
Have you had previous experience with Virtual Reality(VR)?	16	2			
Have you experienced motion sickness in Virtual Reality?	10	6			
	Never	**Occasionally**	**Sometimes**	**Often**	**Always**
How often do you experience motion sickness in Virtual Reality?	0	8	2	0	0

Table B.1: Questionnaire demographic data.

Following are the questions asked after each test:

Q1 I experienced that I was located at some distance behind a visual image of myself, almost as if I was looking at someone else

Q2 I felt that I had two bodies

Q3 I felt as if my head and body was at different locations, almost as if I had been 'decapitated'

Q4 I could no longer feel my body, it was almost as if it had disappeared

Q5 It seemed like I was in control of the water bubble

Q6 It felt like I was able to steer the water bubble while falling

Q7 I was able to merge the water drops with my hands

Q8 I felt as if the virtual body was my body

Q9 I could feel the water drops merging with my virtual body

Q10 I felt as if the water drops were my hands

Q11 How much did you feel that you were inside a raindrop?

Q12 How much did you feel that you were the raindrop yourself?

Questions	**Interaction**	**Non-haptic**	**Non-interactive**	**Detached**
Q1	1.411765	1.416667	2.166667	1.166667
Q2	1.529412	1.583333	2.666667	1.666667
Q3	1.941176	1.75	2.083333	1.5
Q4	2.764706	3.25	3	2.833333
Q5	5.764706	5.916667	1.5	6.083333
Q6	5.941176	5.833333	1.583333	5.916667
Q7	6.058824	6	2.083333	6.583333
Q8	5	4.833333	2.666667	5.25
Q9	4.352941	3.833333	3.166667	4.75
Q10	5.647059	5.083333	1.916667	3.583333
Q11	4.176471	4.166667	4.083333	4.333333
Q12	4.705882	4.583333	3.25	5.166667

Table B.2: Questionnaire test version data.

B.2 Tracking Data

Means	Interactive	Non-haptic	Detached	Non-interactive
Full Test				
Head Position	5.1927	5.1230	4.1136	5.1542
Head Rotation	62.5305	63.1936	66.9248	69.6438
Hand Position	4.3175	4.2850	3.2400	5.2255
Hand Rotation	58.9473	60.9689	58.6389	48.0650
Collect Phase				
Head Position	0.0754	0.0942	0.1165	0.0591
Head Rotation	23.2411	31.1527	41.5086	32.5068
Hand Position	0.1509	0.1835	0.2192	0.1121
Hand Rotation	22.9975	25.7570	28.1598	22.8152
Time				
Total Time	71.8464	69.8038	95.1260	46.0184
Collect Time	31.0968	30.7882	56.4410	22.9956

Table B.3: Table displaying the means of the collected data. Positions show the mean of changes in meters every second, rotations show the mean of changes in degrees every second, and time show seconds that has passed.