

UNIVERSITY OF TARTU  
Institute of Computer Science  
Software Engineering Curriculum

Taavi Gilden

# Paralysed by Surprise: Testing the Active Inference Theory with VR

Master's Thesis (30 ECTS)

Supervisor: Jaan Aru, PhD

Supervisor: Madis Vasser, MSc

Tartu 2017

## **Paralysed by Surprise: Testing the Active Inference Theory with VR**

### **Abstract:**

In recent years, there have been new theories to change the current understanding of how the brain perceives and processes stimuli. The long-standing belief has been that the brain passively waits for sensory stimulation and the received signals are processed in a hierarchical manner to control behaviour. Instead, a theory called active inference states that the system is always active and rather than waiting for the sensory stimulation, it tries to predict it beforehand. According to this theory, the crucial input to the hierarchical system is based on perceived deviations from the predicted state also known as prediction errors. This means that an action is a way of minimizing the prediction error by trying to fit the actual sensory input to the predicted one instead of it just being a response to an input. According to this theory movement is not triggered by motor commands, but instead by trying to minimize the prediction error in the intended movement. This means that in order to move it is required to suppress the sensory input. In other words, it is necessary to withdraw attention from the actual sensory consequences of the movement to allow for movement to happen. If such withdrawal cannot happen the movement should not take place. The goal of this project was to create software to test this peculiar prediction of the theory. In particular, we tried to interrupt the withdrawal of attention during movement in an experiment using the latest virtual reality technology. In the produced environment people were presented with a task to raise their arm as a reaction to stimuli they received from their surroundings. By manipulating the saliency of the visual environment we tried to affect the reaction time of the movement.

Our preliminary findings, however, do not provide clear enough evidence to support the theory of active inference. It could be explained by the small amount of data collected, which might not be sufficient enough to draw conclusions from. Additionally, the saliency of the shapes might have too little impact on disturbing the process of sensory attenuation. Many different factors might be at play here, but the experiment is up for improvements and allows for the theory to be studied further.

**Keywords:** active inference, virtual reality, sensory attenuation

**CERCS:** P176 Artificial intelligence

## **Üllatusest tardunud: ennustava töötluse teooria testimine virtuaalreaalsuse abil**

### **Lühikokkuvõte:**

Viimaste aastate jooksul on tekkinud palju uusi teooriaid, et muuta praegust arusaama, kuidas aju tajub ja töötleb erinevaid stiimuleid. Senine tõekspidamine väidab, et aju ootab passiivselt sensoorseid stiimuleid, mida seejärel töödeldakse hierarhiliselt käitumise kontrollimiseks. Senist uskumust püüab asendada ennustava töötluse teooria, mis väidab, et aju kui süsteem on alati aktiivne ning püüab pidevalt ennustada sensoorseid stiimuleid, selle asemel, et neid passiivselt oodata. Ennustava töötluse teooria kohaselt saab hierarhiline süsteem sisendiks tajutavaid kõrvalekaldeid ennustatavast olekust ehk ennustusvigu. Selle teooria kohaselt on liigutuse näol tegemist ennustusvea vähendamise, muutes päris sensoorset sisendit liigutuse abil sellisel, et ta vastaks ennustatud sensoorsele sisendile. Antud teooria kohaselt ei tekita liigutust mitte motoorne käsklus, vaid soov vähendada ennustusviga kavatsetavas liigutuses. Sellest tulenevalt võib öelda, et iseenda liigutamiseks tuleb sensoorset sisendit alla suruda. Teisisõnu sell-eks, et liigutus saaks aset leida, on vajalik vähendada tähelepanu liigutuse poolt tekitatud tajukogemustele. Kui tajukogemusi pole võimalik alla suruda, siis ei tohiks liigutus aset leida. Antud töö eesmärgiks on testida just seda sama ennustava töötluse teooria raames püstitatud oletust. Täpsemalt üritasime takistada taju allasurumist liigutuse vältel, kasutades uusimat virtuaalreaalsusetehnoloogiat. Valminud katsekeskkonnas esitasime katsealustele ülesandeks reageerida keskkonnast tulevatele stiimulitele käe tõstmisega. Proovisime mõjutada stiimulile reageerimiseks kuluvat reaktsiooniaega, muutes ümbritseva keskkonna visuaalset tähelepanutõmbavust.

Eksperimendi käigus kogutud esialgsed andmed ei toeta selgesõnaliselt ennustava töötluse teooriat. Saadud tulemuste hägusust võib selgitada vähene katsealuste hulk ning seetõttu ka väiksemahuline andmekogum, mis ei ole piisav kindlate järelduste tegemiseks. Lisaks on võimalik, et keskkonna tähelepanutõmbavus oli liialt madal ning seetõttu ei õnnestunud sellega segada taju allasurumist. Tulemusi võisid mõjutada paljud faktorid, kuid katsekeskkonna täiendamine avab kindlasti palju võimalusi antud teooria põhjalikumaks uurimiseks.

**Võtmesõnad:** ennustav töötlus, virtuaalreaalsus, allasurutud taju

**CERCS:** P176 Tehisintellekt

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Theoretical Background</b>	<b>10</b>
2.1	Perceptual Inference . . . . .	10
2.2	Perceptual Hierarchy . . . . .	11
2.3	Active Inference . . . . .	12
2.4	Association to Autism . . . . .	15
2.5	Previous Experiments . . . . .	15
2.5.1	Tickle Body Swap . . . . .	15
2.5.2	Force Matching . . . . .	17
2.5.3	Invisible Hand . . . . .	17
<b>3</b>	<b>Background of Technology</b>	<b>19</b>
3.1	Virtual Reality Technology . . . . .	19
3.2	HTC Vive . . . . .	19
3.3	Unreal Engine 4 . . . . .	20
<b>4</b>	<b>Overview of the Experiment</b>	<b>21</b>
4.1	General Idea . . . . .	21
4.2	Detailed Explanation . . . . .	21
<b>5</b>	<b>Implementation</b>	<b>23</b>
5.1	Overview . . . . .	23
5.2	Physical Setup . . . . .	23
5.3	Virtual Environment . . . . .	24
5.4	Experiment Variables . . . . .	25
5.5	Spawning . . . . .	25
5.6	Shapes . . . . .	28
5.6.1	Scaling . . . . .	28
5.6.2	Rotation . . . . .	29
5.6.3	Colour . . . . .	29
5.7	Data Logging . . . . .	29
5.8	Overall Experiment Flow . . . . .	31

5.8.1	Game Instance . . . . .	31
5.8.2	Experimenter’s Messages . . . . .	31
5.9	Experiment Stages . . . . .	33
5.9.1	Tutorial Stage - Stage 1 . . . . .	33
5.9.2	Tutorial Stage - Stage 2 . . . . .	34
5.9.3	Tutorial Stage - Stage 3 . . . . .	34
5.9.4	Tutorial Stage - Stage 4 . . . . .	36
5.9.5	Tutorial Stage - Stage 5 . . . . .	37
5.9.6	Tutorial Stage - Stage 6 . . . . .	37
5.9.7	Real Experiment - Stage 7 . . . . .	38
5.10	Trial Flow . . . . .	38
5.11	Trial Pause . . . . .	39
5.12	Experiment End . . . . .	39
5.13	Error Checks . . . . .	39
5.13.1	Early Start . . . . .	40
5.13.2	Wrong Start . . . . .	40
5.13.3	Incorrect Hand . . . . .	40
5.13.4	Reaction Time Boundaries . . . . .	41
5.14	Reaction Time . . . . .	41
5.15	Performance . . . . .	41
5.15.1	Identifying Issues . . . . .	41
5.15.2	Improvements . . . . .	42
<b>6</b>	<b>Results</b>	<b>44</b>
6.1	Experiment Settings . . . . .	44
6.2	Subjects . . . . .	45
6.3	Instructions to the Subject . . . . .	45
6.4	Data Analysis . . . . .	46
6.4.1	Data Preprocessing . . . . .	46
6.4.2	Statistical Analysis . . . . .	46
6.4.3	Reaction Time Analysis . . . . .	47
<b>7</b>	<b>Discussion</b>	<b>49</b>
7.1	Discussion of Results . . . . .	49

7.2 Future Work . . . . .	51
<b>8 Conclusions</b>	<b>52</b>
<b>References</b>	<b>54</b>
<b>Appendix</b>	<b>55</b>
I. Code . . . . .	55
II. Licence . . . . .	56

# 1 Introduction

Every person discovers at some point in their childhood that it is impossible to tickle oneself, although others can do it to you with ease. This seemingly simple phenomenon of perceiving two identical stimuli differently has been covered in multiple studies, for review see [Cla15, BAP<sup>+</sup>13]. It is suggested that self-generated action results in suppression of the corresponding sensory input [Hoh13, Cla15, BAP<sup>+</sup>13]. To take the example of tickling: the self-generated action executed in order to tickle oneself suppresses the tickle sensory input coming from the tickle sensation thus self-inflicted tickle is felt as less ticklish than externally generated tickle. This process is often called “sensory attenuation”. Although after numerous studies on this effect a definite cause for such attenuation of sensory information is still debated [BAP<sup>+</sup>13].

The prevailing classic understanding of perception and action states that brain waits for sensory information and actions are triggered as a response to the input [Cla15]. This view renders the brain as passive and stimulus-driven, where perception is merely bottom-up feature detection, meaning that as more information becomes available to the brain, the image of the world is built up from simple features to more complex [Cla15].

However, many studies about neural activity in the brain during the resting state show a continuous activity even when there is not any stimulation produced by a specific task [Cla15]. This leads us to believe that the current passive model might not entirely explain perception and action. The passive model also fails to explain why sensory information is attenuated during self-generated movement.

The phenomenon of sensory attenuation could, however, be explained by a more radical theory called active inference [Cla15]. According to this theory, the brain is constantly active and always trying to predict the sensory information based on a predictive model about the world [Cla15, Hoh13]. Active inference uses prediction errors (sensed deviations from the predicted state) as input in order to increase the mutual information between the mind and the world [Cla15, Hoh13]. In contrast to the passive model of the brain, active inference posits that action is an efficient way of changing the world in such way that the sensory data would fit the predicted state [Cla15, Hoh13]. However, there is a constant conflict between changing the sensory data to fit the prediction

and changing the prediction to fit the sensory data, in other terms active and perceptual inference (see the next chapter for a longer explanation) [BAP<sup>+</sup>13, Hoh13]. In short, this means that the precision of sensory data needs to be reduced in order for a movement to happen [BAP<sup>+</sup>13, Hoh13]. Therefore, sensory input is attenuated to allow for self-generated movement [BAP<sup>+</sup>13, Hoh13]. Previous studies have indeed demonstrated that attention is withdrawn from the sensory consequences of the movement [Hoh13, Cla15, BAP<sup>+</sup>13, LVUA17]. In the present study, we investigate the counterpart of this theoretical idea: in the case where attention cannot be withdrawn from the sensory input, action should not take place [Hoh13].

This phenomenon of sensory attenuation has been noted in the aforementioned inability to tickle ourselves as well as the force matching illusion in which the subjects had to match a reference force by either using their own hand or using a robotic hand [Hoh13, DHS14]. Sensory attenuation and the balance between perceptual and active inference could also be linked to some developmental disorders such as autism and schizophrenia [Hoh13, PLH17]. However previous work has not thoroughly investigated what would happen if attention is difficult or even impossible to be withdrawn from the sensory input.

We conducted a virtual reality (VR) experiment where the subjects are presented with a task while having saliently moving shapes around them. The shapes are more salient in one of the selected fields of view sectors and the subject is asked to perform the task with either the left or right hand. The reaction time for completing the task is measured and a comparison is made between the salient and the less salient case in order to determine whether the reaction time was affected by the saliency of the sensory input.

In the first chapter of the thesis, we give an overview of the theoretical background of underlying theories and previous work. The second chapter gives an overview of the technologies used in the conducted experiment in terms of hardware and software. The next chapter introduces the general idea of the experiment and explains how it was first envisioned. It is followed by the implementation chapter which gives an in-depth overview of how the experiment works and the logic behind how it was developed. In order to review the entire developed project see Appendix I. The fifth chapter analyses the collected results which are then discussed in the sixth chapter, where we also dis-

cuss some suggestions on how to improve the experiment further. Finally, everything is concluded in the last chapter.

## 2 Theoretical Background

### 2.1 Perceptual Inference

The active inference theory is best described starting from the theory of perceptual inference. A good way to illustrate the process of perceptual inference in the brain would be to imagine yourself in a house with no windows or any access to the outside world whatsoever [Hoh13]. Inside this house, you hear a knocking sound and you need to figure out what might it be. In order to best determine the source for the sound, you start listing different causes. It could be a woodpecker, alien spaceship landing on the roof of the house, kids playing outside or a neighbour doing construction work to their house. There could be an endless list of possible hypotheses in order to explain the sound. Which one could actually be it? In order to determine the source of the sound, we need to determine how probable each of these hypotheses are. We would consider how the hypothesis would fit the location and also the specifics of the knocking sound. For example, the hypothesis “woodpecker” is quite probable, since there are trees next to the house, but the sound was not quite right. However, the probability of an alien invasion is nearly non-existent. Could it then be the neighbour? The same process also happens inside the brain, which does not receive images or words from the outside world. Instead, the retina of the eye or ear membrane transform the signals from the outside world into electrical impulses which are sent to the brain and the brain has to translate these signals the same way the person in that house did [Hoh13].

This translation process is thought to be based on the Bayes’ rule by which the brain generates hypotheses in order to explain different perceptual stimuli. This phenomenon is also called unconscious Bayesian perceptual inference [Hoh13]. In this process, the brain assigns a probability for each hypothesis based on the likelihood or probability of the hypothesis and the prior probability of the hypothesis. The likelihood is the probability that the stimulus would occur given that the hypothesis was to be true. The prior probability is an independent probability that the hypothesis would be true based on a subjective estimate. The hypotheses are fit to the incoming sensory stimulus by having a prior ordering and then weighing the highest one with its probability. Additionally, with the help of perceptual hierarchy of top-down and bottom-up message passing it is possible for perceptual inference to get closer to the best hypothesis (see Figure 1) [Hoh13].

There is an additional problem that one sensory input can have many different meanings, thus it is impossible to select the correct hypothesis relying on the sensory information alone. Therefore, there must be some kind of supervision to the whole process, otherwise, there would be no difference between right and wrong inference. In order to constrain the Bayesian process prediction error minimization is used [Hoh13]. The prediction error, in essence, is the prediction subtracted from the actual sensory information. It could be said that prediction error is just the sensory input that has not been explained yet [Clark15]. The main idea here is to match the hypothesis to the incoming sensory data, in other words, to minimize the error between them. To minimize the prediction error between the hypothesis and the sensory stimuli it is either required to revise the hypothesis or change the sensory input. This results in an iterative process which lasts until the hypothesis cannot be further improved upon. The main focus of this process is to fit the prediction of the sensory information to the actual sensory input by taking the top-down predictions and comparing them to the actual sensory information [Hoh13].

## **2.2 Perceptual Hierarchy**

The iterative process of perceptual inference follows the brain's perceptual hierarchy. Throughout the hierarchy, different levels are connected and the best prediction for each level provides the input for the next level in the form of prediction error [Hoh13]. This process of minimizing prediction errors using top-down predictions which are generated as a response for precision weighted prediction errors is illustrated in Figure 1.

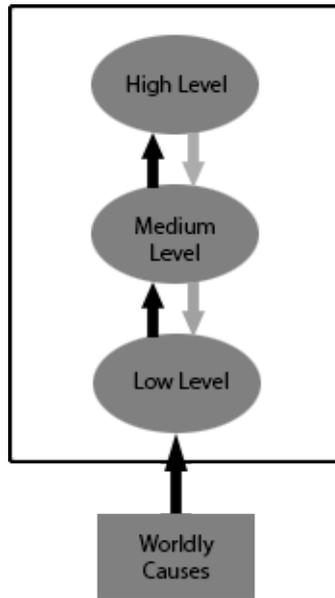


Figure 1. Simplified message flow between the levels of brain’s perceptual hierarchy [Hoh13].

In the figure, black arrows represent the prediction errors from the world and grey arrows show the predictions that have been revised in terms of minimizing the prediction error. Each level decides how strongly the prediction error should influence the changes in the hypothesis at higher levels. This decision is done based on the gain on prediction errors (i.e. their “precision”). This modulates the amount of revisionary work done on each level in the perceptual hierarchy: in the case of a large gain, much of the labour is expected to be done at the higher levels, whereas in the case of a small gain it is mostly done at the current level or lower down. Large gain also translates to the brain learning the global sense in comparison to smaller gain which is focused more individual characteristics and localized learning patterns. Top-down and bottom-up balancing dynamic relies on this logic [Hoh13].

### 2.3 Active Inference

The presented framework of perceptual inference which uses prediction error minimization allows one to also explain self-generated actions. Perceiving and acting are considered as two different methods of achieving the same thing as well as using the same

underlying form of neural computations [Hoh13, Cla15]. Perception tries to fit the hypothesis to the model of the world in order to increase the mutual information between the mind and the world. Another way to increase mutual information would be to fit the model to the hypothesis by changing one's position in the world [Hoh13, Cla15]. For example, there are two different ways to assess whether you have a bicycle or a postcard of a bicycle in front of you. One way to determine the answer would be to try and improve the sensory prediction based on prior beliefs. Another much more efficient way would be to move around and see that it is indeed a real bicycle and not a picture of a bicycle after all [Hoh13]. This method is called active inference where the brain can use action in order to minimize prediction error. When in perceptual inference hypotheses were ordered based on their prior probability, then according to active inference hypotheses are ranked based on their posterior probability [Hoh13]. This all still follows the Bayes' rules and the perceptual hierarchy in a sense that upwards connections are used to transfer prediction errors whereas downwards connections are used for predictions [Hoh13, Cla15]. In perceptual inference, the system minimized prediction error by selecting one hypothesis and predicted how the sensory information should be if the hypothesis were true. In contrast, active inference uses actions to test the hypothesis in order to minimize prediction error [Hoh13]. It involves trying to predict the sensory information that would ensue given the action would take place [Cla15]. Action, therefore, is not a response to an input, rather it is an efficient way of minimizing prediction error by changing the world in order to fit the model to the prediction [Hoh13, Cla15].

There is a clear distinction between active and perceptual inference in the way they handle sensory input and fit to the models. There is a constant competition between shifting from perceptual to active inference in a sense that the brain could fit the input to the world by just adjusting the hypothesis rather than changing the world by taking action. This would prevent the action from being initiated altogether. A mechanism to deal with that involves comparing the expected precision of predicted sensory input to the actual sensory input with a smaller gain. If the predicted input is expected to be more precise active inference is used [Hoh13]. This constant tension between the two ways of perceptual and active inference could be best illustrated by Figure 2.

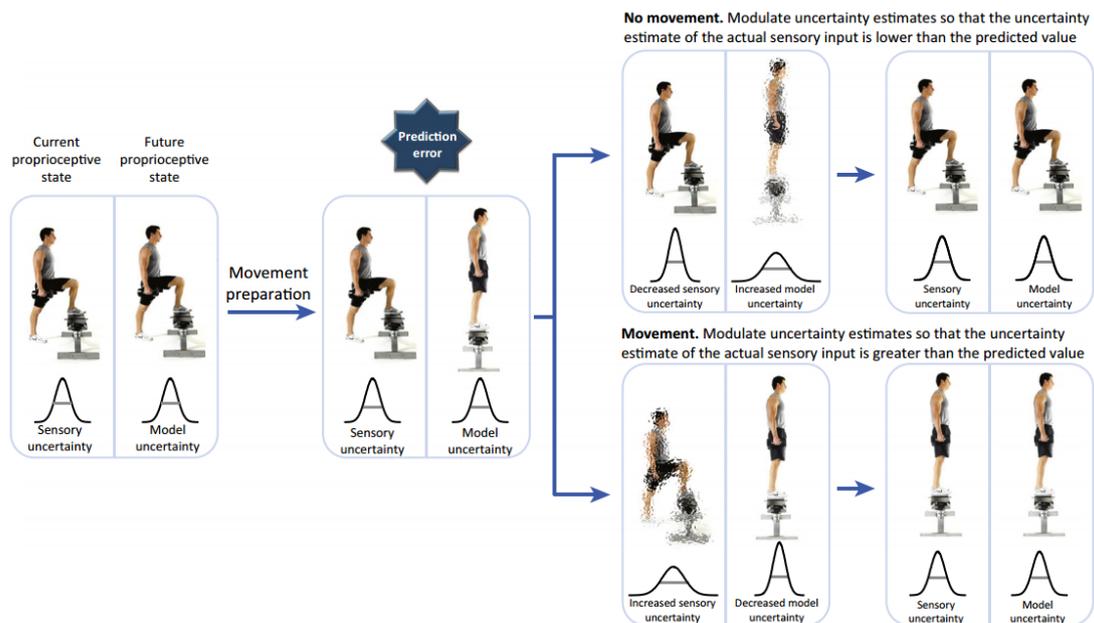


Figure 2. Self-generated movement initiation within active inference [PZK16]. The left image of each panel shows the currently performed action and the right image shows the predicted action. The graphs below each image represent the precision or uncertainty of actual (left) and predicted (right) sensory input [PZK16].

The figure shows the preparation process of self-generated movement using active inference. According to the theory of active inference, before we move we create a prediction about the sensory consequences of the movement and by comparing that prediction to the actual sensory input we get a prediction error. There are two ways to minimize the error: perceptive and active inference. Perceptive inference involves the subject staying still and trying to change the prediction so it would fit the actual sensory input. Active inference, on the other hand, would try to change the actual sensory input to match the prediction by moving. The selection between the two is based on controlling the precision (uncertainty) in these options. In order to move the uncertainty in the actual sensory input is increased so it would exceed the uncertainty of the predicted state. When the predicted sensory input is more precise than the actual sensory input then active inference is used and movement is initiated [PZK16]. Therefore, self-generated movement attenuates actual sensory input [Hoh13].

## **2.4 Association to Autism**

It has been claimed that the theory of active inference could be used to explain some developmental disorders such as autism [Hoh13, PLH17]. Autism is a neurodevelopmental disorder described by impeded development in social interaction as well as language abnormalities and repetitive behaviour [Hoh13, PLH17] It is suggested that due to the reduction of sensory attenuation sampling of the world is prolonged and active investigation of the world is suppressed, in other words, the brain gets stuck in perceptual inference [PLH17].

A known feature in patients with autism is a superior visual search which means that people with autism can locate a target in an array of distractors faster than people without the disorder [PLH17]. Additionally, there are implications that movement initiation such as self-generated arm movement is disrupted in autism. When relating that to active inference it would suggest that sensory attenuation necessary for self-generated movement is impaired in autism. Furthermore, preference for routine and repetitive behaviours imply towards limited sampling of the environment. Repetitive movements are used to continually sample the sensory input and by doing that the precision of the predictions is improved. In essence, this makes the sensory information more predictable and the environment less volatile. It could be that in order to reduce uncertainty in a less volatile environment repetitive sampling may prove to be a better strategy [PLH17]. It could be seen that autism may be described by the balance between perceptual and active inference.

## **2.5 Previous Experiments**

### **2.5.1 Tickle Body Swap**

One of the experiments conducted in order to test the theory that self-generated movement results in sensory attenuation was to demonstrate that we are unable to tickle ourselves. The idea was to show the subjects different perspectives and ask them to rate the ticklishness of externally and self-generated tickle, in other words, show whether the perspective has an influence on the tickle rating [DHS14].

Participants were presented with a wooden rod with a piece of foam attached to both ends, which was used to produce the tickle. Either the experimenter or the subject wore a camera with displayed the hands and the tickle device, and provided a perspective. By using different perspectives, it allowed for an illusion of body swap. Subjects were divided into two groups, differentiated by the perspective. Different setups of the test consisted of replacing the experimenter's arm which was the subject to the tickle with a baseball bat or removing the arm altogether. Removing or replacing the arm allowed to incorporate the rubber hand illusion into the experiment. Additionally, synchronous and asynchronous movements were used. The experiment setup is demonstrated by Figure 3 [DHS14].

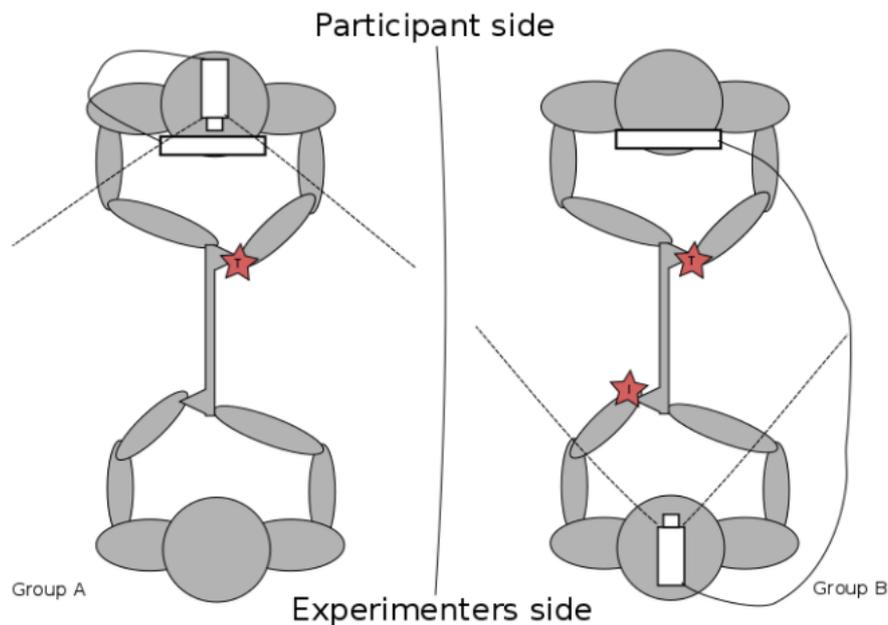


Figure 3. Experiment Setup Seen from Above [DHS14].

Triangular shape at the end of the stick is the foam that is used to produce the tickle. The stick was held from both ends by the experimenter and the subject. There is a camera and goggles mounted to either the participants or the experimenter's head. The stars represent the tickle points, either real or illusory [DHS14].

Another thing that differentiated between groups was the touch type which was either synchronous or asynchronous. In the case of the synchronous touch the foam was seen to be moving away from the body and it was felt that it moved away and in the latter case when the foam moved away from the body it was sensed to be moving towards the body. This illusion was accomplished by manipulating the subject's perspective with the help of the camera and goggles [DHS14].

The ticklishness levels were measured with every run. The results showed that tickles generated by external movement were generally stronger than with self-generated movement. This result remained the same in both cases when the participant saw the scene from experimenter's point of view or his own. Additionally, the tickle effect was reduced by changing the context of the limb by using the bat or invisible hand, but the effect did not completely disappear. These results follow the sensory attenuation theory in a sense that self-generated movement reduces the precision the sensory input [DHS14].

### **2.5.2 Force Matching**

The phenomenon of sensory attenuation is not only present in the tickle action. A good example of sensory attenuation is also demonstrated by the force-matching task [BAP<sup>+</sup>13]. The experiment is constructed in a way that subjects try to match a given reference force. The subjects could try to match the reference force by either using their own hand and pressing directly on themselves or by using a robot to express the reference pressure [BAP<sup>+</sup>13].

The results showed that when the force was self-generated then the force produced was stronger, thus matching the previous statement about sensory attenuation. In other words, the force produced by self-generated movement is perceived as less intense compared to the true force applied [BAP<sup>+</sup>13].

### **2.5.3 Invisible Hand**

Another great example to support the theory of sensory attenuation during self-generated movements would be the invisible hand experiment [LVUA17]. The experiment uses VR technology, similarly to the current project, along with Leap Motion hand track-

ing system. Three separate experiments were conducted in order to gather significant data which could support the theory of active inference. During all the experiments the subject's hand is made invisible to the subject, but is still tracked using Leap Motion technology. This was done to determine whether the brain would suppress the sensory input caused by the movement of the hand in the area where the hand would have been. Throughout the experiment, subjects saw horizontally oscillating spheres which were covering the whole field of view. The participants were asked to move their hand in a smooth motion from the starting position on top of a table to their eye level and a target to which the subjects had to react to was triggered by the described movement. In the first two experiments targets differentiated from the rest of the spheres by moving vertically and in the third experiment the targets changed colour. The subjects had to react to the targets by clicking the mouse as soon as they noticed them [LVUA17].

Comparison of reaction times was conducted between the targets which spawned behind the arm (invisible arm) and those that did not. After data analysis, all three experiments showed overall slower reaction times when the target was in the area of the moving hand supporting the theory of active inference in a sense that self-generated movement attenuates sensory input [LVUA17].

### **3 Background of Technology**

In order to carry out the experiment, several different technologies are used. Participants all need to have an identical environment without any disturbing factors that might affect the results. For that VR technology is used, namely HTC Vive [Viv]. The environment is developed using the Unreal Engine 4 [UE4d] game engine.

#### **3.1 Virtual Reality Technology**

Virtual reality (VR) allows us to generate an immersive three-dimensional environment and interact and move around in it in person. The technology itself has been around for a long time, around 70 years, although at first, it was nothing like the way we know it today. The participants sat in a rotating chair to face the screen which displayed stereoscopic images that gave an impression of depth. Later came head mounted displays, although these were too heavy and had to be held up by a suspension device. Virtual environment has found its usages in many fields mainly military and aviation as combat training and flight simulation, but also in medicine [VR1].

The technology has become more widespread in the last three to five years and also found many more usages. Mainly in the entertainment field, such as theatrical applications, museum exhibitions, and also a proportionate factor in gaming. Additionally, there has been a huge development in the technology itself. The headsets are smaller, more mobile and comfortable and provide users with a high-resolution image which offers a realistic experience. The movement is tracked by infrared sensors and hand controllers are used to interact with the environment as well as use many implemented tools. The increasing accessibility and versatility of the VR technology as well as enabling many restrictions present set by the conventional tools of experimental psychology has made it a popular choice to carry out scientific experiments [VR1].

#### **3.2 HTC Vive**

HTC Vive is the latest VR headset developed by HTC and Valve. It utilizes all the latest technologies starting from the infrared sensors to allows the user to move around the room in a 4.6 by 4.6-meter space and transfer that movement to the virtual environment. The head mounted display uses two OLED displays with each of them having a reso-

lution of 1080x1200 pixels. The refresh rate for the Vive is 90 Hz. Additionally, HTC Vive uses wireless handheld controllers which are also motion tracked and enable the user to interact with the system and use the buttons to give different input [Pra, DS].

### **3.3 Unreal Engine 4**

Unreal Engine 4 is a game engine developed by Epic Games and released in 2012. It includes a complete suite of tools needed to develop different 2D and 3D scenes. All the features, including future updates, were made completely free for anyone to use in 2015. Unreal Engine 4 was chosen because of its strong community and good documentation. The game engine is very capable in handling VR technology, by rendering different scenes at very high framerates. The engine uses C++ as its development language and is a perfect tool to develop the test environment used in the experiment [UE4f, Sir].

## **4 Overview of the Experiment**

### **4.1 General Idea**

In order to test the effects of salient visual distractors on the speed of self-generated movement, a VR experiment had to be developed. The idea was to build a framework, which could be used by non-technical people to carry out the experiment. The vision for the given experiment is to have an environment which could be used to present the subject with a relevant task. It would include a stimulus and a task which has to be conducted by the subject as a reaction to the stimulus. The stimulus object is to be surrounded by objects with different levels of saliency, from which the subject has to move their virtual hand through. The subject should first be allowed to accustom to the task and the controls by having practice trials after which, when the subject is ready, the real test would begin. The experimenter can control the flow of the experiment accordingly taking into account the feedback from the subject. The experimenter should have feedback on subject's errors and data collected after each trial. Additionally, the collected data should be written automatically to a log file.

### **4.2 Detailed Explanation**

The subject is first presented with an empty environment in which the VR head mounted display could be calibrated so that the image is clear. In the next phase of the experiment, the subject is asked to put their hands straight down by their side and the program sets this as the initial location of both hands. Continuing from there, the subject is asked to hold their hands forward at shoulder height and the program spawns a button between the subject's hands on the same height level. The subject can now try pressing the spawned button. Next, it is explained to the subject to respond with a different hand according to the button colour. Blue corresponding to the left hand and red to the right hand. The subject is presented with small animated guidance arrows, which show the rough trajectory by which the hand should move and also the correct side. The subject can now try alternating hands according to the stimulus colour. Next, the subject can try pressing the button with salient shapes around the stimulus button. The shapes should spawn around the button and they should be separated into 4 equal sections. In each of the trials, two of the diagonally opposite sections should have increased quantity and saliency of shapes in it. After the tutorial phases, the experiment proceeds to the final

phase. Now as well as in the last phase, the subject does not have any guiding arrows. In the final phase, all of the subject's actions are logged and data is collected. There are a total of 360 trials with small pauses after every 60 trials. Throughout the different phases, the experimenter can see various instructions to utter to the subject in order to guide them through the experiment. The experimenter screen also contains feedback from subject's actions.

In this framework, it should be easy to modify initial parameters about the subject such as age, gender and subjects number, as well as parameters about the experiment itself, for instance, the number of salient shapes in a sector or the time it takes for stimulus to appear.

## **5 Implementation**

### **5.1 Overview**

This project is using Unreal Engine version 4.14.3 and it is based on the VR blueprint template, including the starter content, both offered by the engine itself. The C++ code of the project is developed relying on the Unreal Coding Standard [UE4a].

The purpose of the template is to set up the basic level and player pawn. The template level comes with a prebuilt room which includes some example intractable content such as cubes you can pick up using the motion controllers. The level also has an ambient light with a sky sphere set up. The template pawn is created with basic input action and axis mappings in order to move and control the character pawn with motion controller buttons and movement. Additionally, it has hand meshes replacing the motion controllers in-game.

The starter content consists of predefined assets which could be used within the respective level and its design. The starter content specifically includes architecture elements such as walls and pillars, sample audio and some commonly used materials, textures and shapes. The use of templates and starter content is a great way to start prototyping an idea.

### **5.2 Physical Setup**

In order to conduct the experiment, it is required to have a sufficiently large space (minimum 2 by 2 meters) to set up the VR setup. The space needs to be covered by the two base stations in order to track subject's hand and head movement. The subject should be equipped with the head mounted display and two handheld motion controllers. All of the aforementioned equipment should be included with the HTC Vive product [Viv]. The subject should be sitting on a chair positioned in the middle of the sensor-tracked space. The experimenter should monitor the experiment from the desktop computer connected to the VR setup.

### 5.3 Virtual Environment

The virtual environment presented to the subject is comprised of an adequately sized room that is surrounded by walls on every side, has a ceiling and a floor. None of the surfaces are transparent. The floor has a walnut wood texture and the walls and the ceiling are black as could be seen from Figure 4.

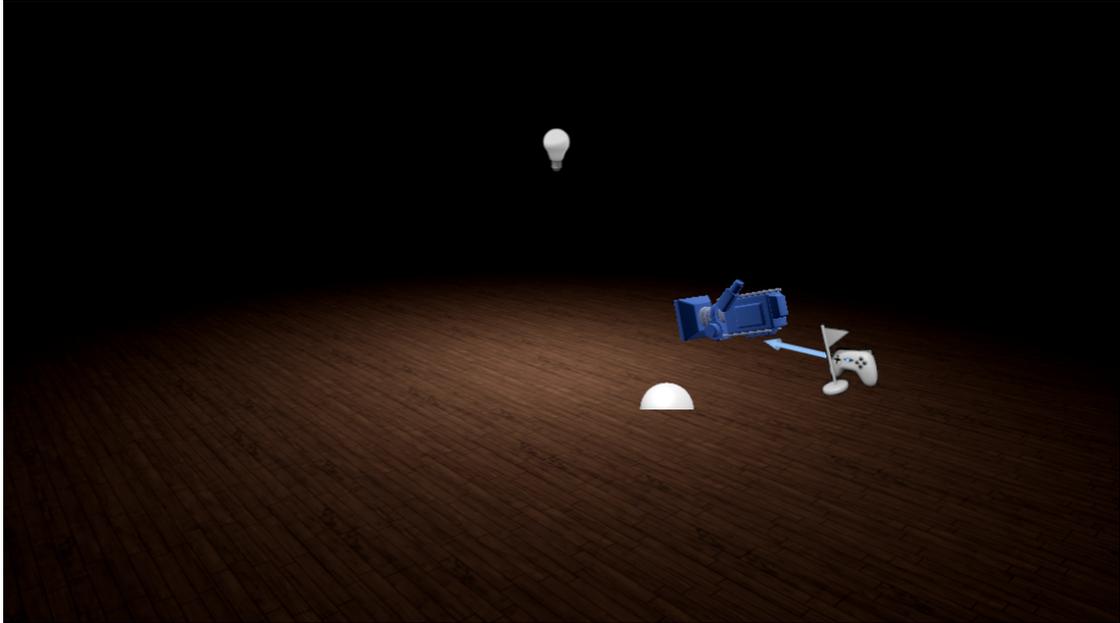


Figure 4. Illustrative image of the virtual environment.

The room is surrounded by a lightmass importance volume which specifies the area in the experiment world in which the lighting must be more detailed in order to decrease the lighting build times. The room is also surrounded by a post processing volume. This allows us to fine-tune the visual post-processing parameters. Inside the room, there is a point light, which emits light to the location to which the subject and the rest of the objects in the experiment are spawned.

## 5.4 Experiment Variables

The experiment framework allows for the experimenter to modify many initial parameters about the subject and the experiment itself. Most of these are located in the spawn component and are made visible for editing in blueprints. The spawn component keeps information about the log file name, subject's identification number, subject's gender, subject's age, the spawning method and current trial number as well as current salient sector index. The log file name specifies the name of the external file into which the experiment data is logged. The name should be set without any file extensions since it is automatically added in the logging method. Spawning method is defined as an enumeration and could be changed between "All sectors" or "One sector". These will be explained in more detail below. All the aforementioned variables are public and readable from a blueprint. The subject's data with log file name and the spawning method are editable by the experimenter before the start of the experiment.

It is also allowed for the experimenter to specify the reference to the class of which objects are spawned, as well as the number of objects in each sector. Additionally, since the objects are spawned in a spherical shape (hemisphere) it is also allowed to change the spawn radius. All the aforementioned variables have predefined default values and can be later modified in the pressable button blueprint class as the experimenter sees fit.

## 5.5 Spawning

The spawning of the shapes is done by the spawn component which is inherited by the pressable button class. The component is responsible for everything that has to do with spawning the salient shapes around the button. The shapes are spawned around its parent object, in the current case the button, in a spherical manner. In order to do that we first get the world location of the component's owner. Next, a random point is generated so that it would be on a surface of an imaginary hemisphere with a predefined radius as could be seen from Figure 5.

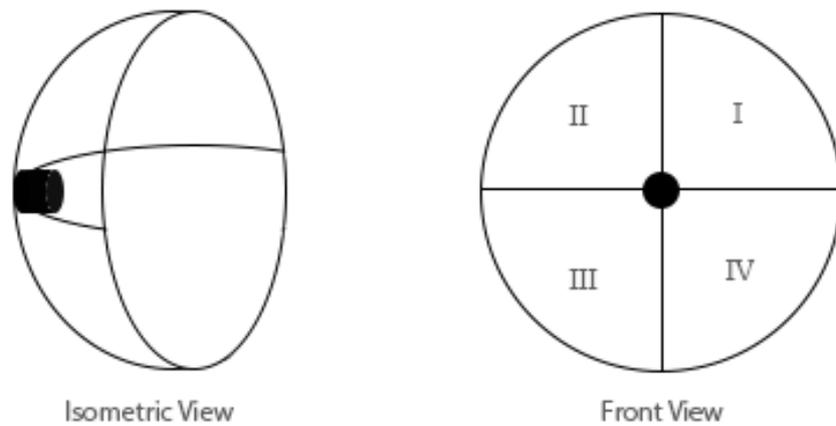


Figure 5. Illustration of the button's location and the hemisphere on which the shapes are spawned as well as the enumeration of the sectors.

The generated three-dimensional location vector is then modified according to which sector it belongs to. The sphere radius is added to the Y-coordinate in order to push the spawned half-sphere backwards for its side to be on the same plane as the parent object. Each of the four sectors is spawned according to its own rules and the shape location vector is modified accordingly. Each shape is spawned with a random rotation.

Spawning component distinguishes shapes as regular shapes and extra shapes. Regular shapes are visible to the subject at all times and do not get changed by switching the salient side around. Extra shapes are defined as the shapes which visibility is dynamically changed. Their in-game visibility is toggled on and off according to the current salient side. That in mind, when spawning a shape to a sector it is noted that whenever a shape is spawned as an extra shape it is hidden in-game and its reference is saved to a map containing sector indices and their corresponding shape references. This map could be later accessed from a blueprint in order to easily modify the shapes' movement parameters to make them more salient compared to the regular shapes as illustrated in Figure 6.

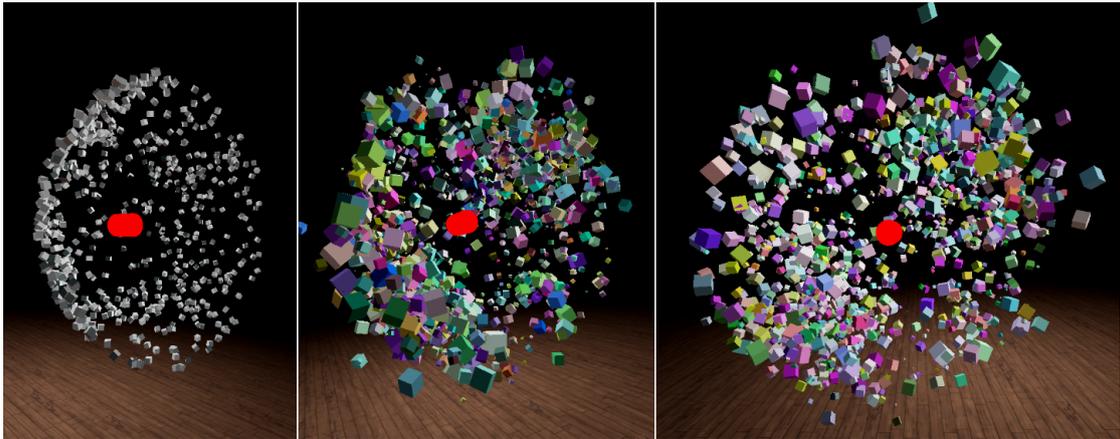


Figure 6. Illustrative image showing the spawned shapes first without any movement or colour added to them and then the shapes with movement and distinguishable sectors from an isometric view and front view.

Spawn component handles two different methods of spawning the shapes to the world and changing the salient sectors around the button. The experimenter can easily switch between the two by selecting the corresponding option in the pre-experiment setup.

The first method which was developed as an initial idea on how the spawning should work is called “All Sectors”. Each time the subject presses the button, all of the shapes around the button are destroyed and respawned according to the experiment and spawning algorithms. It handles the salient sectors by spawning additional shapes in them and the amount is defined by a predefined coefficient. This method works, but was discarded half way through the development process since it was very costly to do in terms of computation and rendering and produced performance issues. The only upside to this method is the ability to use different meshes for the shapes, meaning they all do not have to be cubes and can be cones or spheres or any other random static mesh for each separate trial.

The second method, which is also the default for the experiment is called “One Sector”. In this method, the shapes are spawned around the button only once. The regular shapes, as well as the extra shapes, are all spawned initially at once, but the extra ones are hidden to the subject at first. They are shown by toggling the hidden in-game flag

and scaling the shape size. When the subject presses the button the current salient sector shapes are hidden from the subject by getting the references of all the shapes in the sectors and setting their scaling phase enumeration variable to “Despawning”. From there on the shape class handles the despawning of the shape. The shapes are made visible again also according to the experiment flow and by doing the exact opposite of despawning. That means we set the scaling phase enumeration variable to “Active”, toggle the hidden in-game flag and let the shape class handle the rest. The downside to this method is the restriction on using only one shape mesh for the shapes. It is restricted in order to preserve the smooth transition between salient sectors. However, the major upside to this method is that after each button press the shapes are not respawned, rather their visibility is toggled, which makes it much more efficient in term of computation load and thus resulting in a smoother experience. The “One Sector” spawning method is strongly recommended to be used for carrying out all of the experiments.

According to the predefined values, each of the four sectors has a total of 150 shapes which are constantly visible to the subject. The extra shapes are spawned with a multiplier of 2, resulting in 300 extra shapes per sector. Considering there are two extra sectors visible to the subject at a time results in 1200 shapes which are presented to the subject around the button each trial. The radius of the half-sphere the buttons are spawned in is 50 cm. The base value of shapes in a sector, as well as the radius, can be changed by the experimenter. Changing the base value also affects the number of shapes spawned as the extra salient shapes.

## **5.6 Shapes**

### **5.6.1 Scaling**

All the spawned shapes are based on the shape class. Each shape has a maximum and minimum scale values as well as a pre-set scaling speed. The class is responsible for scaling the shapes between their maximum and minimum scale values along with dealing with despawning and spawning.

Shape scaling phase is stored in an enumeration with three possible values: active, despawning and despawned. The scaling phase is checked each game tick. When it is active the shape pulsates between its set scaling limits at the scaling speed. Scaling

speed is the value which is either added to or subtracted from the shape's current scale each tick, making it change size and appear to be pulsating. In order to make the shapes pulsate faster, the scaling speed must be increased. When the scaling phase is set to despawning the shape is gradually scaled to zero. When the shape reaches scale value of zero its scaling phase variable is set to despawned, which means the shape does not pulsate anymore.

### **5.6.2 Rotation**

Shape blueprint inherits from a rotating movement component which is used to give an actor continuous rotating movement at a specific rate. The rotation can also be offset by a set amount so that the shape rotates around a pivot point. For each spawned shape the input parameters for the rotating movement component are randomised from a set range.

### **5.6.3 Colour**

Each shape also changes colour at a certain interval. Colour changing is defined in the material using a vector noise material expression. The noise position is set to be dependent on time to ensure continuous colour change. A random parameter is added to the time value at the start in order to make the vector noise expression start at a different position for each shape. This results in all the shapes being a different colour.

## **5.7 Data Logging**

Data about user's actions is logged to an external .csv file throughout the experiment with semicolon as the delimiter. The filename into which the data is logged must be set by the experimenter before the experiment starts. Logging of data only happens in the last phase of the experiment after every single trial. Additionally, at the start of the last phase, the log file header is written to the file. A trial is defined as a full cycle from moving hands from the initial location, pressing the button and returning the hands to the initial location. Throughout the trial, the logging message is stored into a map at various stages and it is logged as an entire row at the end of a trial. The data points logged each trial can be seen from Table 1.

Table 1. Data points logged about the subject after each trial.

<b>Index</b>	<b>Name</b>	<b>Description</b>
0	Date	Timestamp of the trial start
1	Subject's Number	Subject's identification number
2	Gender	Subject's gender
3	Age	Subject's age
4	Trial Number	Trial number about which the data is logged
5	Spawn Method	Method used for spawning the shapes
6	Salient Side	Numeric value of 1 or 2 representing the side on which the salient sector spawned, where 1 corresponds to left and 2 corresponds to right side
7	Button Colour	Numeric value for stimulus colour, where 0 corresponds to blue and 1 corresponds red
8	Hand Used	Hand used for pressing the button
9	Wrong Start Detected	Boolean value showing whether subject moved the wrong hand out of the initial location and returned it before pressing the button with it
10	Wrong Hand Used	Boolean value showing whether the subject used the wrong hand to react to the stimulus
11	Reaction Time Too Fast	Boolean value showing whether the subject's reaction time was under a predefined value representing too fast of a reaction time
12	Reaction Time Too Slow	Boolean value showing whether the subject's reaction time was over a predefined value representing too slow of a reaction time
13	Shape Spawn Delay	Length of the delay in seconds between the time when the subject initialised their hand location and the salient sector shapes start appearing
14	Stimulus Delay	Length of the delay in seconds between the time when the salient sector shapes start appearing and the stimulus appearing for the user
15	Stimulus Start Time	Time in seconds from the start of the experiment to the point when the stimulus appeared

16	Button Press Time	Time in seconds from the start of the experiment when the subject successfully reacted to the stimulus by pressing the button
17	Reaction Time	Time difference between stimulus appearing and button pressed (seconds)

“NaN” is written to the columns with indices 7-17 in the case of a failed trial.

## 5.8 Overall Experiment Flow

The overall experiment flow and logic are defined by the Unreal Engine blueprints. It is done in this way in order for the flow to remain easily modifiable later on.

### 5.8.1 Game Instance

The Unreal Engine general game flow follows the order in which after pressing the play button in the editor we create and initialise the game instance. This is followed by Unreal Engine world class begin play method and the game mode, after which the actors are spawned and the game starts [UE4b]. The key point being that the game instance is initialized before the actors are spawned.

Therefore, in the experiment, the game instance blueprint is used to store some of the global variables accessed by actor blueprints. Most notable ones stored are experiment stage, trial number and the left and right hand initial location booleans. Experiment stage handles distinguishing between different tutorial trials and the real experiment trial. Trial number stores the current trial number and the left and right hand initial location booleans show whether the hands are returned to the initial location.

### 5.8.2 Experimenter’s Messages

Throughout the experiment, the experimenter will receive messages, showcased in Table 2, from the output console.

Table 2. Experimenter's messages in Estonian and English.

Stage	Language	Message
1	Estonian	Kalibreerida katsealuse pildi teravus. Palun hoia käsi sirgelt enda kõrval. Fikseeritakse käte algasend.
	English	Calibrate subject's image quality. Please hold your hands straight down by your side. The initial location of your hands is being set.
2	Estonian	Palun siruta mõlemad käed sirgelt õlast ette ja hoia neid seal, kuni käte vahele ilmub nupp.
	English	Please raise both hands straight ahead at shoulder height and hold them there until a button appears between the hands.
3	Estonian	Tegemist on mootorikakatsega, uurime inimeste liigutuste täpsust. Palun hoia käsi enda kõrval ning püüa võimalikult kiirelt reageerida nupu värvi muutustele! Püüa kogu liigutuse vältel hoida käsi sirgelt ja pilk nupu peal.
	English	This is an experiment used to study movement accuracy. Please hold your hands straight by your side and try to react to the colour change of the button as quickly as possible. Try to keep your hands straight and your eyes focused on the button throughout the motion.
4	Estonian	Eri värvi nupule peaksid reageerima erineva käega. Sinine värv on vasak käsi, punane värv parem käsi.
	English	Different hands should be used to react according to the button's colour. Left hand for blue and right hand for red.
5	Estonian	Päris katses ei näe sa abistavaid noolekesi käe juures, vaid erinevaid objekte ümber nupu. Püüa pulti liigutada nii, et see oleks kontaktis objektidega.
	English	In the real experiment, you do not have the guiding arrows by your hands. Instead, you have a variety of objects around the button.
6	Estonian	Et katse oleks sulle visuaalselt huvitavam, on nupu ümbruses ka erinevad liikuvad objektid.
	English	There are different moving objects around the button In order for the experiment to be visually more appealing

7	Estonian	Nüüd algab päris katse. Iga katsekord algab sellest, et hoiad käsi enda keha kõrva ja vaatad nupu poole. Kogu aeg on nupu ümber visuaalsed objektid ning ühel hetkel muutub nupu värv. Olles tuvastanud nupu värvi, vajuta palun õige käe puldiga nuppu selliselt, nagu eelnevalt harjutasime. Vajutamise järel pane käed uuesti keha kõrvale. Kokku toimub 360 katset, see võtab umbes 20 minutit. Iga 60 katse järel on pisike paus.
	English	The real experiment starts now. At the start of each trial, you should keep your hands by your side and look straight at the button. At all times, there are moving objects around the button and at one point the colour of the button changes. Once you have identified the button's colour you press the button using the correct hand as practiced before. After pressing the button, you return your hand back to the initial location straight down by your side. There are a total of 360 trials which takes around 20 minutes. Every 60 trials is followed by a small pause.

These messages help the experimenter guide the subject through each stage of the experiment and also are a reference point for the experimenter in order to know the current stage. The messages displayed to the experimenter at the start of each trial.

## 5.9 Experiment Stages

There are 7 different stages to the experiment. Each stage starts with a message that is displayed to the experimenter and is followed by different logic defined in various blueprints. Stages 1-6 are tutorial stages whereas the 7th stage marks the beginning of the real experiment. The experimenter can proceed to the next stage by pressing the spacebar.

### 5.9.1 Tutorial Stage - Stage 1

Stage one of the experiment is automatically initialized when the experimenter starts the experiment from the Unreal editor. This stage purely for image calibration. The subject can look around and get used to the headset and the motion controllers. There is nothing spawned to the environment yet, the subject can look around and see whether

the quality is adequate. Finally, the subject is asked to keep their hands straight down by their side, before proceeding to the next stage.

### **5.9.2 Tutorial Stage - Stage 2**

At the beginning of the second stage, the initial hand location is set, provided that the subject's hands are in the initial position straight down by their side. This is done by spawning initial hand location blueprint actors to the location of both motion controllers. The spawned actor includes a box collision object which is used to determine whether the hand is in the defined location or not. The spawned actors can be visually seen by the subject in order to give a visual reference point for hand initial location. At the end of the second stage, the subject is asked to raise both arms straight ahead at shoulder height.

### **5.9.3 Tutorial Stage - Stage 3**

At the start of stage three, a button is spawned between subject's hands, as could also be seen from Figure 7.



Figure 7. Stage 3 of the experiment, showing the button and box collision object representing initial hand location.

Hands' location is equivalent to the location of the corresponding motion controller. Button location calculated as the exact middle point of the two motion controllers' locations as is shown in Equation 1.

$$ButtonLocation = \frac{(LeftHandLocation + RightHandLocation)}{2} \quad (1)$$

After the button is spawned to the environment, the subject can try pressing it by touching the button by hand. The triggered overlap event which fires when the motion controller pawn overlaps with the button initiates a small vibration in the corresponding motion controller as a feedback to the subject that the action was successful. In this stage, the button has a generic bright white colour to indicate that the subject can press the button and the button is in fact active.

#### 5.9.4 Tutorial Stage - Stage 4

The fourth stage introduces matching the stimulus colour to the correct hand. The blue button should be pressed with the left hand and red button with the right hand. Button's colour is randomly selected between blue and red each trial as could be seen from Figure 8. Guiding arrows are spawned in order to aid the subject.

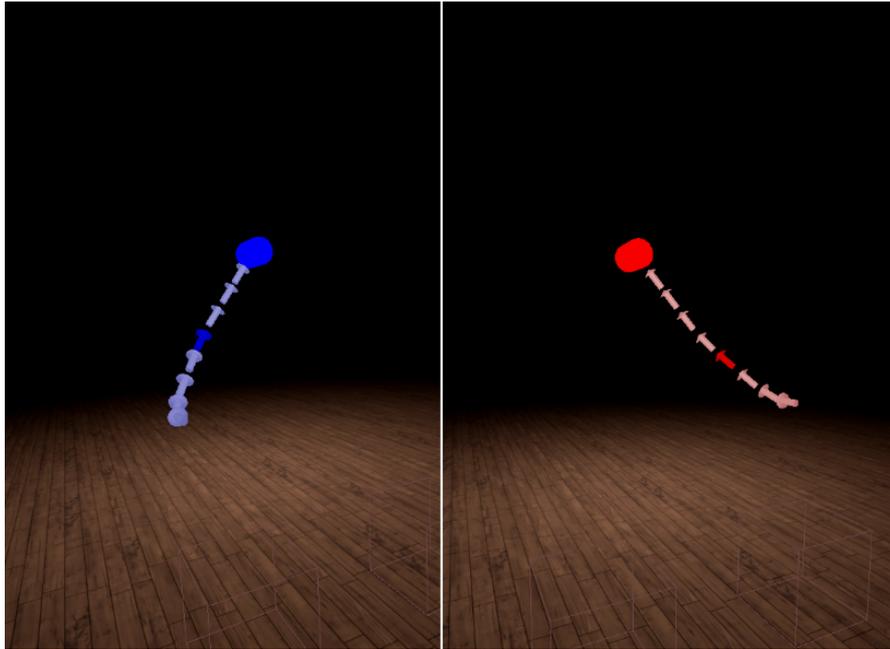


Figure 8. Showing the guiding arrows and different coloured buttons.

The arrows have a short two-second-long animation during which each separate link in the arrow blinks for a small amount of time. The arrows are spawned into the sockets on the button actor, which are predefined locations on a model to which objects can be attached to. Arrow's colour, as well as the animation delay, can be easily modified through spawn arrow function defined in the button blueprint. The guiding arrows should help the subject to understand which hand to use and that the arrow's trajectory should be roughly followed when reacting to the stimulus by pressing the button.

### 5.9.5 Tutorial Stage - Stage 5

Stage five is an intermediate stage. In this stage, the subject can continue practising pressing the button with the guiding arrows. The subject is informed by the experimenter that in the real experiment there are no guiding arrows. Instead, there are different moving object around the button and the hand should move through the shapes.

### 5.9.6 Tutorial Stage - Stage 6

This stage introduces the salient shapes to the subject. This stage best resembles the real experiment. The subject can no longer see the aiding arrows to help them choose the correct hand. In addition, the shapes spawned by the spawn component pulsate in size and change colour randomly. Also, two of the diagonally opposite sectors around the button are more salient as illustrated in Figure 9.

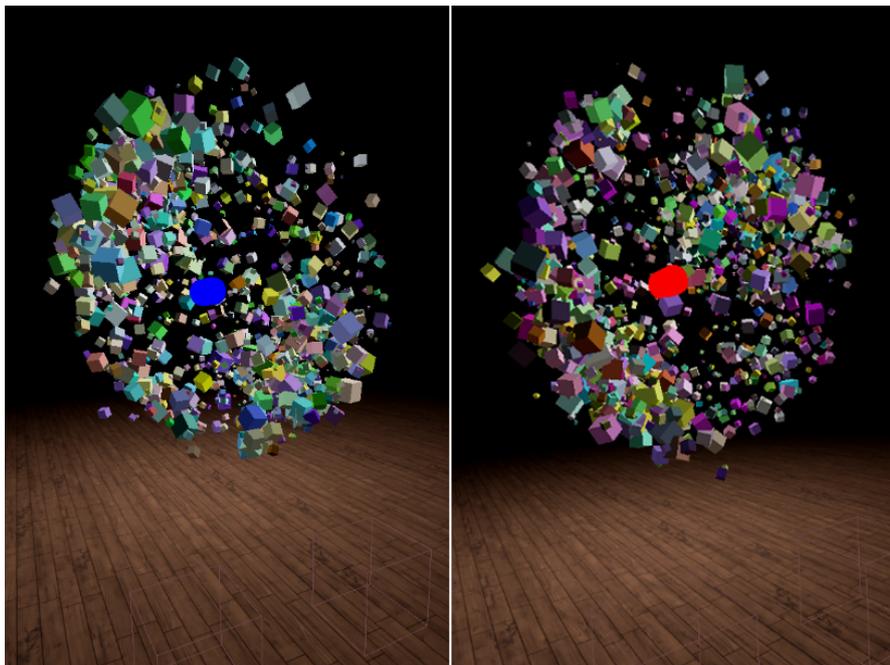


Figure 9. Showing two possibilities for salient sides as well as illustrating the stage 6 of the experiment.

The subject can get used to the shapes being there during the task. This is the last tutorial stage and the experimenter should make sure the subject understood the task

and is ready for the real trials.

### **5.9.7 Real Experiment - Stage 7**

The subject is informed once more about the task and told that the real stage of the experiment starts. There are no guiding arrows in this stage, all the erroneous movements are logged and also the experimenter is informed by a console message in case any incorrect movements happen. The cases checked for are the following:

- Started hand movement with wrong hand and did not press the button;
- Pressed the button with the wrong hand;
- Reaction time was too slow;
- Reaction time was too fast.

Visually the real experiment is identical to the last tutorial stage. In this stage, all of the information mentioned in Section 5.7 is logged about the subject. The subject must do the predefined amount of trials, the default values being 360 trials and after each 60 trials, there will be a small pause.

## **5.10 Trial Flow**

Each of the trials has the following logic:

- Trial starts by the subject having their hands in the initial position;
- Check is done to make sure whether the trials should be on pause;
  - If true, 10-second delay;
- Check is done to make sure whether the experiment should end;
  - If true, experiment is stopped;
- Extra salient shapes start spawning to either the right or the left diagonally opposing sectors;
- Check is done to make sure subject's hands are still in initial location;

- If false, considered as failed trial and trial is reset;
- 600 - 800 ms delay;
- Stimulus colour is selected and button lights up;
- Subject reacts to the stimulus;
- Trial is reset.

The length of the two delays can be modified by the experimenter as seen fit.

### **5.11 Trial Pause**

After each 60 trials, the experiment goes on pause temporarily for 10 seconds. The pause check function checks that when dividing the current trial number by 60, the remainder is zero, meaning that 60 trials have passed since the last pause. At the start of the pause, the subject is notified in-game by text stating that the trial is paused. When there are 3 seconds remaining, a countdown commences for the subject and for the experimenter to inform them that the experiment is about to continue.

### **5.12 Experiment End**

The experiment ends after 360 trials have passed. The end check is done similarly to the pause check. The trial number is checked to be equal to 360 and if it is the experiment ends. The subject, as well as the experimenter, are informed the same way they were when the trial is paused.

### **5.13 Error Checks**

The following checks are done in order to validate the experiment output and give additional data about the number of times the subject initiated an incorrect movement. A notification is logged to the console output for the experimenter, whenever one of these errors occur.

### **5.13.1 Early Start**

The early start check covers the case when the subject moves one or both of their hands away from the initial location at the time when the extra salient shapes are visible, but the stimulus is not. This is done to preserve consistency throughout the trials so that we could be sure that the subject would always start reacting to the stimulus from the initial hand location and all of the reaction times measured would be comparable. In case the subject starts moving their hands before the stimulus appears, the trial is reset without increasing its number and a corresponding message is also logged to the output file. By not increasing the trial number when a failed trial occurs we ensure that each subject does 360 valid trials. Early start error check is the only check considered as a failed trial and results in "NaN" written to the log file.

### **5.13.2 Wrong Start**

This check determines the cases when the subject started to react to the stimulus with the wrong hand and returned the hand to the initial position halfway through the motion without pressing the button. This check is done in the initial hand location blueprint by using global variables initialised in the game instance. These boolean variables are true by default and are set to false by pressing the button. When the hand returns to the initial location, the variables are checked. If they are false then this means that the hand completed the task by pressing the button and there is no wrong start. The wrong start is logged in case the hand returns to the initial location and the variables are still true. Meaning neither of the hands pressed the button to set the variables in game instance false.

### **5.13.3 Incorrect Hand**

The incorrect hand check is defined in the button blueprint. This check takes the reference from the button overlap event and casts it to a motion controller. From that, we can obtain which hand the controller corresponds to. Next, we can compare that information to the button colour and log the appropriate output, whether the button was pressed with the correct hand or not.

#### 5.13.4 Reaction Time Boundaries

Reaction time boundaries are predefined variables which show us whether a trial has a valid reaction time. The reaction time is considered to be too slow if it exceeds 3000ms and too fast if it does not exceed 500 ms. These values can be easily modified by the experimenter to suit their needs.

#### 5.14 Reaction Time

The reaction time is the amount of time in seconds it takes for the subject to react to the stimulus in a single trial. It is started exactly at the time point when the button changes colour and a local variable is set in the button blueprint with the time since the experiment world was brought up for play. The end time is set right when the subject presses the button and the time from the start of the overall experiment is logged. The reaction time for the trial is considered to be the difference between the start time and the end time.

#### 5.15 Performance

Spawning and rendering the 1200 shapes around the button, unfortunately, resulted in some performance issues. The desired frame time for a VR environment that runs on HTC Vive should be roughly 11 ms, which is around 90 frames per second. However, the current frame rate for the experiment is around 60 frames per second.

##### 5.15.1 Identifying Issues

In order to determine whether the performance is bound by CPU or GPU an engine built in feature called *stat unit* was used. The command brings up an on-screen overlay showing frame, game, draw and GPU times in milliseconds, as illustrated in Figure 10.



```
Frame: 11.17 ms
Game: 2.21 ms
Draw: 11.07 ms
GPU: 11.19 ms
```

Figure 10. *stat unit* command overlay.

The frame time is the actual time it takes for each frame to be processed. Game time corresponds to the CPU game thread time, draw time is the CPU render thread time and GPU is the time it takes for the graphics card to process its data each frame. In order to know which is the determining factor we need to compare the latter three ones to the frame time and see which is closest to the frame time. The one closest is the bounding factor.

The next step involves CPU or GPU profiling depending on which one is the bounding factor. For GPU profiling one of the easiest tools to use is GPU visualizer. It allows us to see the GPU cost of different passes to an extent of single draw calls. CPU profiling can be done by running the commands *stat startfile* to start the profile and *stat stopfile* in order to end it. The profile should be recorded during the extent of the experiment for at least ten seconds. The profile can then be opened using Unreal Frontend Session Browser [UE4e] in the editor from which the frame time element can be inspected for tasks which take the most time. Shortcomings can be highlighted using both these profiling techniques and thereafter eliminated.

### 5.15.2 Improvements

One of the major performance issues during development was changing the colour of each shape. The idea at first in order to change each individual shape's colour was to change the RGB value of an individual shape by a small random value each game tick. This would have resulted in a smooth colour change, but was too CPU intensive and had to be scratched. Instead, the colour changing was put inside the actual material, which proved out to be much more efficient.

Small improvements were made to the post-processing and rendering settings in order to improve the frame times. The experiment environment is surrounded by a post-processing volume which allows us to edit some of these parameters. In the case of a VR environment, it is suggested to modify them in order to follow best practices. The following effects were disabled:

- Bloom;
- Lens flare;

- Motion glare;
- Motion blur
- Screen space reflection;
- Ambient occlusion;
- Light propagation.

Bloom effect is a real world light phenomenon in which bright light extends over the borders of darker surfaces. Lens flare effect in which when looking at bright objects the light coming from them scatters. Motion blur is an effect in which the displayed image starts streaking due to rapid movement. Screen space reflection effect affects the way how objects appear on other surfaces, for example, reflections. Ambient occlusion effect changes the exposure of different objects in the scene to ambient lighting, in essence, it makes corners and crevices darker to make them look more realistic. Light propagation affects how light is bounced from various surfaces and for better performance it is also disabled [UE4c].

Also, some rendering option were modified. By default, Unreal Engine uses a deferred renderer for versatility and additional rendering features. However, for some VR cases, it might be better to use forward rendering which offers faster rendering passes. Additionally, it also allows us to choose from better anti-aliasing options such as multisample anti-aliasing (MSAA) to produce better quality environment. After trying out multiple different setting using forward rendering it did not produce better frame times and the settings were reverted back to the defaults. Anti-aliasing was turned off altogether in favour of performance.

The GPU profiling tool also identified some issues with rendering lighting and shadows for the shapes. This was fixed by replacing ambient light source with a point light. The point light was also set to be static in order to reduce rendering times since a static light does not cast shadows on dynamically moving objects, in our case the shapes.

After all these improvements, the best average frame rate achieved was 60 frames per second. This fluctuates strongly whenever a new salient sector is made visible and then stabilises back at around 60.

## **6 Results**

### **6.1 Experiment Settings**

The variables affecting shape rotation and scaling, as well as the number of shapes and time delays were set up according to visual assessment beforehand to result in the best possible outcome for the theory.

The passive shapes were each assigned a rotation rate from a range of 20-200 degrees per second. Additionally, the passive shapes were given a rotation pivot offset randomly from the range of 0-4 cm. The minimum and maximum scaling limits for the passive shapes were selected from the range of 0-0.01 and 0.02-0.06 respectively. The scaling speed was selected randomly from the range of 0.001-0.0025 per game tick.

The salient shapes were distinguished from the passive shapes by increasing the rotation rate range to 70-500 degrees per second, rotation pivot offset range to 3-5 cm, maximum scaling limit range to 0.02-0.07 and lastly the scaling speed range to 0.0025-0.005 per game tick.

There were 150 passive shapes in each of the four sectors and 300 salient shapes resulting in a total of 600 passive shapes and 1200 salient shapes of which only 600 were visible at a time considering the experiment logic. The shapes are spawned on the surface of a hemispherical shape with a radius of 50 cm.

The delay range for the salient shapes to become visible once the subject has returned their hands to the initial location is 700-900 ms and the delay range for the stimuli to appear after the salient shape spawn delay has finished was 600-800 ms.

There were a total of 360 trials for each subject and the subjects had a small 10 s pause after every 60 trials.

## **6.2 Subjects**

11 subjects took part of the VR experiment in which 8 of the subjects were male and 3 were female. Subjects age ranged from 19-30 with a mean of 23.1. A total of 3960 trials in which reaction time was measured were gathered. The average reaction time across all subjects was 826 ms with a standard deviation of 190 ms.

## **6.3 Instructions to the Subject**

At first, the subjects were introduced to the VR device and the motion controllers. They were shown how to properly use and hold the controllers and the headset was put on. The subjects were then seated on the chair. Next, the experiment was started and stage one of the experiment began. The subjects were properly guided through the different stages of the experiment according to the experiment flow described in Section 5.9.

In the first stage, it was made sure that the subject could see everything clearly and the image quality was adequate. Then the subject was asked to look to the correct direction and keep their hands straight down by their side. The experiment was proceeded to the next stage.

In stage two the subject was asked to hold their arms forward at shoulder height after which the experiment was proceeded.

In the third stage, the subject was explained to keep their eyes on the button at all times as to press the button with a straight arm as well as react to the button colour change as fast as possible. After the subject has practiced enough the experiment was proceeded. Next, the subject was introduced to reacting with different hands to different colours, there were guiding arrows and they could practice the movement.

In the next stage, the subject was explained that there would be different objects around the button and to move the hand so that it would be in contact with the shapes. The experiment was proceeded and the shapes spawned around the button.

In this stage, the guiding arrows were removed and the subject was told that this would be the last trial phase and this phase resembles the real experiment the closest. Before

proceeding to the real experiment the subject was explained that each trial would begin by keeping your hand by your side and looking at the button, and after determining the colour you should react with the correct hand as practiced before. After the hand motion, the hands must return to the initial location. There will be 360 trials and after each 60 trials, there will be a small pause. The experimenter proceeds the experiment once more and the real experiment begins.

During the real experiment the experimenter, could help the subject if they encountered any difficulties or had any other issues. The experimenter monitored the experiment from the output log where information about different errors occurred.

## **6.4 Data Analysis**

### **6.4.1 Data Preprocessing**

The data preprocessing and analysis were done by using the R programming language (version 3.2.3) [R]. The data preprocessing began by removing the rows from the data frame corresponding to the failed trials. However, the way the experiment is designed it forces the subjects to repeat the failed trials, leaving us still with 3960 data points as expected from 11 subjects. Additionally, the cases with invalid start with the wrong hand and pressing the button with the wrong hand were removed, which was roughly 5% of the data points. Finally, cases, where the reaction time was too fast or too slow, were removed. Limiting values for these conditions were 500 ms and 3000 ms respectively. However, this process only removed 3 trials from the 3960. The data was not trimmed any further. After the preprocessing, there are 3771 data points which could be analysed.

### **6.4.2 Statistical Analysis**

For statistical testing, repeated measures ANOVA was used. The model used for the statistical test, checked whether reaction time changes depended on the salient side and hand used. Since repeated measures design was used we also try to account for the between-subject variation by defining an error term into the same model.

### 6.4.3 Reaction Time Analysis

Based on the 11 subjects, reaction time analysis was done to determine whether there could be a difference between the mean reaction times of salient and non-salient movement conditions. The results are shown in Figure 11.

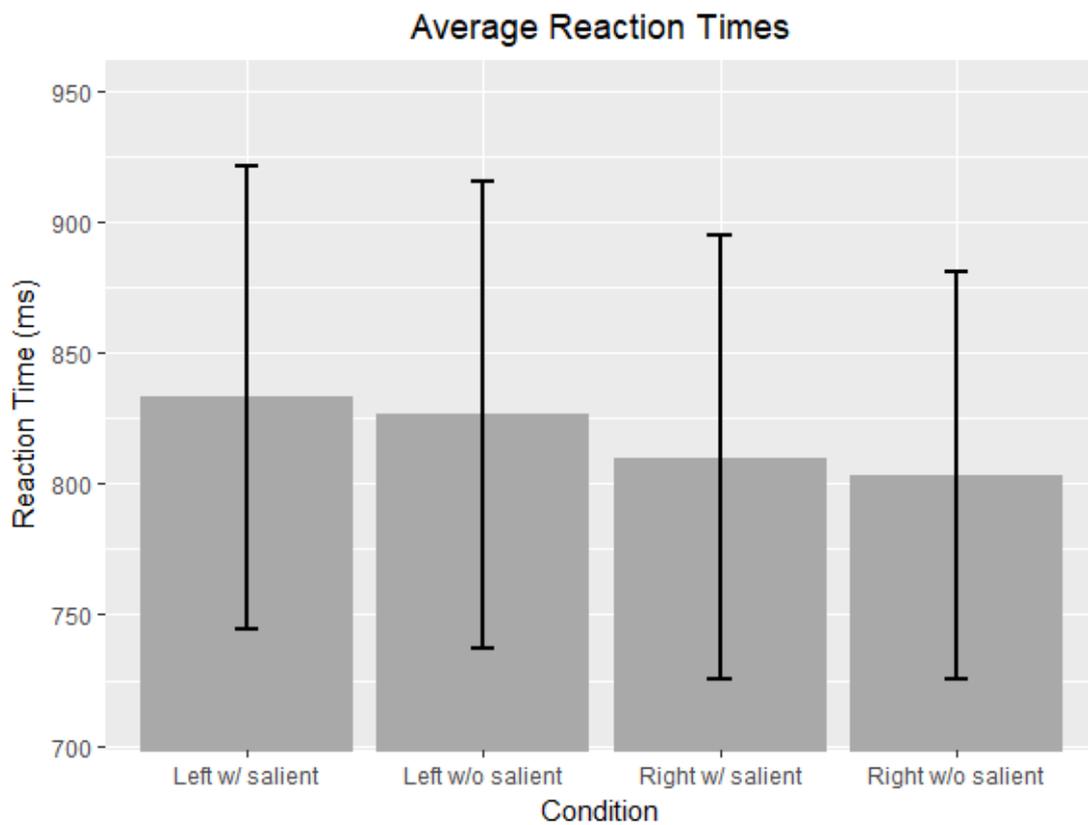


Figure 11. Reaction time means of control and experimental conditions for both hands.

The mean reaction time for the case when the left hand was used and the salient shapes were also on the left was 832.9 ms with a standard deviation of 176.4 ms. The mean reaction time for the left hand without the salient sectors was 826.4 ms with a standard deviation of 178.1 ms. For the right hand, mean reaction time with salient shapes was 810.0 ms with a standard deviation of 169.4 ms and without salient sectors, the mean reaction time was 803.0 ms with a standard deviation of 155.4 ms.

The results of the statistical analysis show that the null hypothesis should not be rejected with an  $f$  value of 0.013 and a  $p$ -value of 0.909, indicating the variance between mean reaction times of movement with salient shapes and movement without salient shapes was not significantly different.

Although there is a slight difference between the salient and non-salient side in terms of the salient side resulting in slower reaction time it is not enough to draw any conclusions from. We could say that the experimental condition is equal to the control condition, supporting the statistical analysis which suggested to accept the null hypothesis.

## 7 Discussion

### 7.1 Discussion of Results

It is essential to have a deeper knowledge of how the brain perceives the outside world, processes stimuli and how everything is linked together to generate movement, since understanding the human brain can lead to improvements in many fields such as artificial intelligence systems or medicine. As illustrated by everyday experiences and general theories about the brain [Cla15, Hoh13, BAP<sup>+</sup>13] the sensory information produced by self-generated movement is attenuated. Understanding that process might help us have a better understanding of developmental problems such as autism and schizophrenia.

In this study, we used the latest VR technology to control the subject's visual scene and through that attempt to increase the saliency of the surroundings in order to manipulate the speed of self-generated movements. In the conducted experiment the saliency was differentiated in both left and right field of view and the subject was asked to react to a stimulus with the correct hand, which was based on the stimulus colour. The reaction time of the self-generated movement was measured. Based on the underlying theory movement produced on the side with more salient distractors should have a greater reaction time.

However, the current results do not decisively support the theory of active inference in a sense that difficulty in attenuating sensory information results in slower movement. The difference between the control conditions and the experimental conditions were marginal thus not allowing us to draw any conclusions from. There could be many reasons for why the results were only so slightly different.

Firstly, the number of subjects was very small. Since the performed action is quite time-consuming and resulting in large deviations in reaction time it might be wise to have a larger sample of subjects in order to get a more accurate reaction time mean.

Another very important reason could be that the extra shapes were not salient enough. In other words, the extra spawned shapes did not induce the disturbance in sensory attenuation and thus did not make it more difficult to initialize movement. So, we did not

see the change in the reaction times between different conditions. It was also mentioned by some subjects that they did not notice a major difference between the passive and the salient shapes. This could be solved by increasing the shapes rotation rate, rotation offset and scaling maximum limit. Additionally, increasing the overall amount of extra shapes could provide us with the desired results. The saliency of the shapes could also be increased by making their colour changing more vigorous.

Third, the results might have been affected by the predictability of the salient sector. The subject might have learned the pattern of constantly changing salient sectors and therefore the anticipated impact to the reaction time from the saliently moving visual distractors was insignificant. A solution for that would be to make the salient sectors appear less predictably. For example, the salient sectors could only appear on a third of the overall trials which would make them less predictable and therefore more greatly affect the reaction time.

It could however also be that the passive shapes that were constantly visible to the subjects were themselves already too salient and the spawning of extra shapes did not increase the already elevated levels of saliency enough to have an effect. This could be solved by decreasing the rotation rate, rotation offset and maximum scaling limit of the passive shapes.

Lastly, the desired results might not have been present because the time delays before spawning the extra shapes and before the stimulus appeared were suboptimal. It could be that the subjects had too long or too short of a time to see the extra salient shapes. The subjects could have gotten used to the extra shapes, meaning that the delay was too long or they saw them too briefly before the stimulus and the saliency was not accounted for.

Sensory attenuation is the foundation on which the theories of perceptual and active inference rely on: in order to generate movement precision of sensory information or perception must be lowered. It is supported by the everyday phenomenon of inability to tickle ourselves [DHS14] as well as many other studies [Cla15, LVUA17, BAP<sup>+</sup>13], which would lead us to believe that the experiment should be reconfigured and further data should be collected in the hope for more decisive results.

## 7.2 Future Work

The virtual environment of the experiment could be improved further to perhaps provide us with better results or just improve the overall quality.

Firstly, one of the biggest issues was performance. The amount shapes with their rotating movement and constant scaling produced a noticeably large frame time. In order to solve the issue, the whole shape logic would need to be redeveloped using instanced static meshes. Instanced static meshes are much more efficient since all the instances are drawn at once whereas with separate static meshes components the draw command is called for every element and it becomes very computationally intensive.

Another minor quality to the experiment could be to rewrite the shape coordinate spawning algorithm so that it would consider the subject's rotation in the environment. Currently, the shapes are spawned in a predefined way, which is why we have the subjects facing a certain way in the beginning of the experiment for the button to spawn with a correct rotation. If the algorithm that generates coordinates for the shapes on a hemisphere were to be rewritten to take into account the subject's rotation then the subject could face any direction at the beginning of the experiment and shapes would be spawned correctly.

Lastly, one of the more minor improvements could be adding additional information to the log files. Some experiment parameters could not be retraced using only the log information. For example, the number of shapes and their rotation and scaling variables would provide valuable information about assessing the saliency of the environment. Currently, this information could only be retrieved directly from the experiment settings but not from the log file.

## **8 Conclusions**

The goal of this thesis was to gain more insight to the theory of active inference by trying to understand the phenomenon of sensory attenuation in self-generated movements. We attempted to show that the difficulty in attenuating sensory input would result in delayed movement. In order to verify the hypothesis, a VR experiment was conducted where both fields of view were distinguished from each other by saliency. This was done by displaying a vast amount of salient visual distractors to the subject. The subjects had to react to a stimulus with either left or right arm movement depending on the stimulus. All of the relevant data produced by the subject's movement was logged, in addition, the erroneous actions were detected. From the experiment we observed that the movement on the side with more salient visual distractors did not produce a significant difference in reaction time when compared to reaction times from the side with no extra distractors. Based on the collected data and analysis no strong conclusions could be drawn, which lead us to believe that the experiment should be reconfigured and more data should be collected for more decisive results.

## References

- [BAP<sup>+</sup>13] Harriet Brown, Rick A. Adams, Isabel Parees, Mark Edwards, and Karl Friston. Active inference, sensory attenuation and illusions. pages 411–427. *Cognitive Processing*, 2013.
- [Cla15] Andy Clark. Embodied prediction. *Open MIND*, 2015.
- [DHS14] George Van Doorn, Jakob Hohwy, and Mark A. Symmons. Can you tickle yourself if you swap bodies with someone else? *Consciousness and Cognition*, 2014.
- [DS] Dante D’Orazio and Vlad Savov. Valve’s vr headset is called the vive and it’s made by htc. <http://www.theverge.com/2015/3/1/8127445/htc-vive-valve-vr-headset>. Accessed: 2017-05-18.
- [Hoh13] Jacob Hohwy. The predictive mind. pages 1–97. Oxford University Press, 2013.
- [LVUA17] Kristjan-Julius Laak, Madis Vasser, Oliver Jared Uibopuu, and Jaan Aru. Attention is withdrawn from the area of the visual field where the own hand is currently moving. *Neuroscience of Consciousness*, 2017.
- [PLH17] Colin J. Palmer, Rebecca P. Lawson, and Jacob Howhy. Bayesian approaches to autism: towards volatility, action, and behaviour. American Psychological Association, 2017.
- [Pra] Lily Prasuethsut. Htc vive: Everything you need to know about the steamvr headset. <https://www.wareable.com/vr/htc-vive-vr-headset-release-date-price-specs-7929>. Accessed: 2017-05-18.
- [PZK16] Clare Palmer, Laura Zapparoli, and James M. Kilner. A new framework to explain sensorimotor beta oscillations. *Trends in Cognitive Sciences*, May 2016.
- [R] The r project for statistical computing. <https://www.r-project.org/>. Accessed: 2017-05-18.

- [Sir] Jordan Sirani. Unreal engine 4 is free for everyone. <http://www.ign.com/articles/2015/03/02/unreal-engine-4-is-free-for-everyone>. Accessed: 2017-05-18.
- [UE4a] Coding standard. <https://docs.unrealengine.com/latest/INT/Programming/Development/CodingStandard/>. Accessed: 2017-05-18.
- [UE4b] Game flow overview. <https://docs.unrealengine.com/latest/INT/Gameplay/Framework/GameFlow/>. Accessed: 2017-05-18.
- [UE4c] Post process effects. <https://docs.unrealengine.com/latest/INT/Engine/Rendering/PostProcessEffects/>. Accessed: 2017-05-18.
- [UE4d] Unreal engine 4. <https://www.unrealengine.com/>. Accessed: 2017-05-18.
- [UE4e] Unreal frontend. <https://docs.unrealengine.com/latest/INT/Engine/Deployment/UnrealFrontend/>. Accessed: 2017-05-18.
- [UE4f] What is unreal engine 4. <https://www.unrealengine.com/what-is-unreal-engine-4>. Accessed: 2017-05-18.
- [Viv] Htc vive. <https://www.vive.com/eu/product/>. Accessed: 2017-05-18.
- [VR1] How is virtual reality used? <http://www.vrs.org.uk/virtual-reality/how-is-it-used.html>. Accessed: 2017-05-18.

# Appendix

## I. Code

In order to more thoroughly review the developed Unreal Engine 4 project as well as the script written for data analysis, see the Bitbucket repository at <https://bitbucket.org/jackstick/myprojectthesis>.

## II. Licence

### Non-exclusive licence to reproduce thesis and make thesis public

I, **Taavi Gilden**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
  - 1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
  - 1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, as of 18.05.2018 until expiry of the term of validity of the copyright,

of my thesis

**Paralysed by surprise: Testing the active inference theory with VR**

supervised by Jaan Aru and Madis Vasser

2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 18.05.2017