Exploring different implementation methods of introducing a forward-thrusting motion into a pointing task

David Mockovsky
Department of Architecture, Design and Media Technology
Aalborg University
Aalborg, Denmark
dmocko18@student.aau.dk

Oscar Bill Zhou
Department of Architecture, Design and Media Technology
Aalborg University
Aalborg, Denmark
ozhou18@student.aau.dk

Mikkel Sang Mee Baunsgaard
Department of Architecture, Design and Media Technology
Aalborg University
Aalborg, Denmark
mbauns18@student.aau.dk

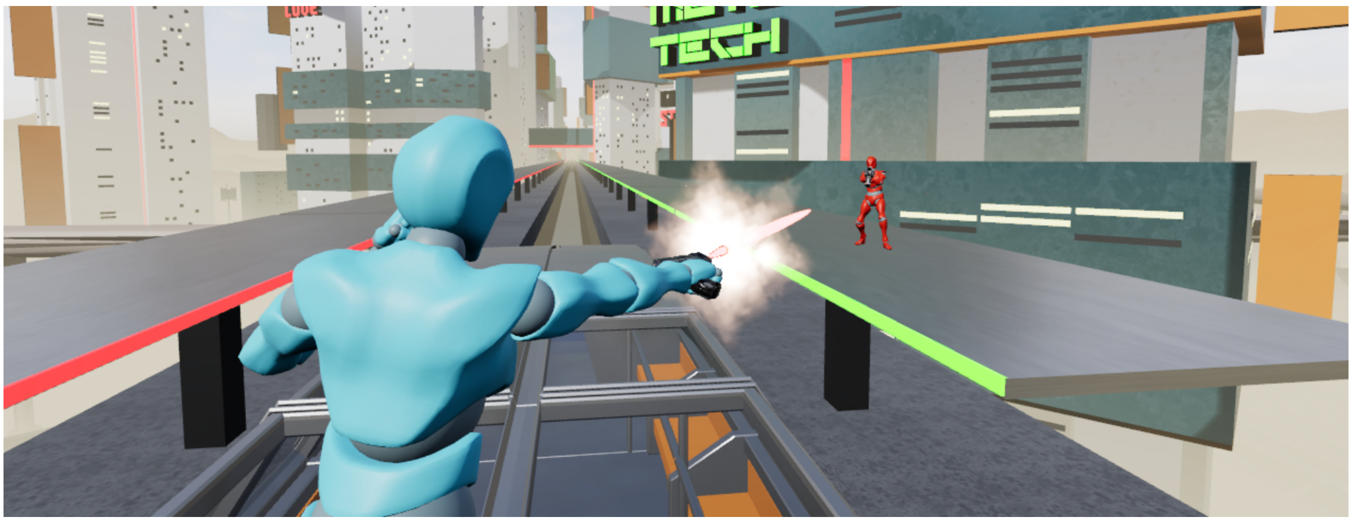

Figure 1. Screenshot from the VR game prototype, with a placeholder figure for the player (blue).

Abstract

Games can be decomposed into different core-game play tasks, which are all composed of particular action levels. These action levels describe the task in terms of lower-level blocks. One such task is pointing, where a review of existing games showed that pointing tasks usually include a button press as an activation step. This project investigated the use of VR to replace this button press with a particular movement instead, which we called thrusting, thus creating a new type of pointing. Using this in correlation with serious games, we created a VR video-game that facilitates exertion in users, intended either for training, or rehabilitation. A within-subject study (n=24) was carried out, investigating three variations of the thrusting implementation. The results show that an ideal interaction of thrusting would use a combination of the wrist direction and hand movement to determine pointing direction. Most prevalent limitations found were human locomotion and incorrect pre-processing of the controller position values. Future research should investigate the relationship between the wrist direction and movement further, to potentially either enforce particular motions to the user in the context of rehabilitation or to create an exercise game.

Keywords: Virtual Reality, Pointing, Core-Gameplay, Serious games, Thrusting Motion, Human Locomotion, Pointing Task, Exertion.

ACM Reference Format:
David Mockovsky, Oscar Bill Zhou, and Mikkel Sang Mee Baunsgaard. 2018. Exploring different implementation methods of introducing a forward-thrusting motion into a pointing task. In . ACM, New York, NY, USA, 17 pages. https://doi.org/XXXXXXX.XXXXXXX

AAU, 2023, MED10

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

In the world of games, every game regardless of it's genre, platform or complexity has core gameplay mechanics.
In the scope of core gameplay in video games, within this paper we focus on the task of pointing.
Video games have a variety of pointing tasks, where traditional shooter games require the user to use a mouse as an input device and shoot at targets. Recently, with the increasing popularity of Virtual Reality (VR) platform, these games have explored the 3 dimensional (3D) motion controller instead of the 2 dimensional (2D) mouse, which changes the actions a player would need to perform.

Furthermore, video games can have different purposes, with the most common one being entertainment. However, this is not always the case, and there is a substantial focus on games that are centered around learning or various real-world problems, which are referred to as ***serious games*** [13].
Laamarti et al. [13] reviewed existing literature on the topic of serious games.They defined *serious games* as applications consisting of three components; entertainment, experience and multimedia. With this in mind they created a taxonomy, to classify serious games into different criteria, which they reviewed within different domains such as education or cultural heritage [13].
With this in mind, we aim to create a serious game that focuses on physical exertion within a virtual environment, while also providing entertainment.

Therefore, we are exploring the aforementioned pointing task and looking at its potential within the realm of serious games.
To accomplish this, we need to deconstruct what pointing is mechanically and look at which variables we can modify in order to turn it into a game that not only exists for entertainment purposes, but also makes the user benefit from exercise.
Here, we have to look at the realm of human computer interaction (HCI) to understand the task holistically.

2 Background Research

Within the field of cognition, there exists an approach created to investigate the analysis and design of human work, referred to as the Systemic-Structural Theory of Activity (SSTA) [4]. SSTA iterates on the notion of activity-theory, which describes cognition as a conjuncture between a process and a structured system of actions, to create taxonomies that can describe human work processes as analytical units. These units describe a particular activity as a whole [4].

Hougaard et al. [1] investigated the use of this within the context of video games, as they described the different units of analysis that constitutes an activity from SSTA standpoint, very briefly described as follows:

An **operation** is the conscious act of doing something, carried out by an unconscious sequence, e.g. pressing a button is the composition of pressing, holding and releasing it [1]. An **action**, which can be a motor or mental action, such as pressing a button.A **task** which is a sequence of one or more actions. An **activity** which is one or more tasks, such as cooking food. And finally, a **goal** which directs the activities.

The same paper [1] reviewed these tasks in relation to Debus [6], relating the task concept from SSTA to game mechanics and goals in order to investigate core gameplay tasks. This, according to a paper by Refai et al. [19], which investigates core tasks and different assistance techniques that can be used within them, is the *"basic motor and perceptual skills used for interacting with game mechanics"* [19]. These include 10 unique types, which Hougaard et al., grouped based on their task type, being motor or mental, and what research has been conducted within it.

The review illustrated that different games can implement a particular task in different ways, meaning that no clear overview of the dimensions of a particular task exists [1]. To demonstrate this, they investigated the core task of pointing, more specifically, a target-to-target task, which refers to hitting one target after another. This was done to exemplify properties that can describe it. With these properties in mind, we will continue to explore the use of pointing tasks and their action levels, within the field of video games. To do this, we carried out a state-of-the-art review, which will be explained below.

3 State of the art review

The state of the art review is two-fold, with the first section focusing on papers related to pointing and the research done within the field. The second section explains video games and their use of pointing.

3.1 Papers

There are several papers within the pointing task domain, that suggest how to investigate the performance of a pointing task.
McArthur et al. [16] investigated pointing in 3D space with 6 degrees of freedom (DOF) using a Wiimote controller. They used a within-subject (n=12) experimental design to test four different input methods, three target widths and two target distances. They evaluated the performance based on Fitt's law, focusing on the throughput - bits per second, of the device. Additionally, they collected accuracy data and participant feedback using a questionnaire (52 items in total, 13 per input method). From this, they reported physical fatigue, effort and force required being linked to poor performance.

Additionally, they found that participants have different approaches to pointing, achieving higher accuracy when using a device which allows for pointing with a wrist, compared to a device which only allows for pointing using an elbow.

Zhai et al. [22] also investigated 6 DOF input devices, focusing on usage of different muscle groups by manipulating an object in a virtual 3D space in a within-subject (n=12) experimental design. The motivation for this paper stems from the differences between representation of body parts in the human motor and sensory cortex. For this, they used two input devices, the first being a glove, which used the rotation and translation of one's wrist, elbow and shoulder to point; and the other device being a "FingerBall", held using one's fingers and using them as an additional muscle group to wrist, elbow and shoulder. The results of their experiment shows a significant advantage of the FingerBall in completion times, supporting their hypothesis of using fine-motor muscle groups as advantageous. However, they discuss that a part of this result was due to their design of the glove, which required a button press (or a clutch of the hand) as an activation, that had a negative impact on its results. However, even when not accounting for the activation time, the glove was still outperformed by the FingerBall.
They also argue for future designs of pointing devices to incorporate multiple muscle groups, especially fingers, as input whenever possible. With these results in mind, we aim to design an interaction that uses multiple muscle groups simultaneously, while keeping the actions required for an interaction as few as possible. However, the design of our implementation requires the user to grasp the controller with the entire hand, omitting fingers despite the study's suggestions.

Conversely, Balakrishnan and MacKenzie [2] investigated controlling of devices with fingers, wrist and forearm based on the muscle's bandwidth (i.e. information processing capability). Just as Zhai et al. [22], their motivation is based on the physiology of the human brain and its motor system's disproportionate sizes dedicated to different body parts. To measure the performance they focused on Fitt's law, investigating movement time and error rate of the task in addition to the bandwidth. Interestingly, unlike previous research mentioned in their paper, they did not find a significant difference in bandwidth between the wrist and the forearm. Here, they believe that the bandwidth is dependent on the specific task, where a left-right motion will have different results to a flexion-extension. Further, they found that whether finger(s) outperform other parts of the upper limb is also task-dependent, and can therefore have lower performance than wrist or arm. However, they found that the unison of index finger and thumb do outperform every other limb segments. Overall, the best design for a pointing device is one that takes advantage of each limb and muscle group's best performance, all working together, based on the specific task.

Based on the studies on the subject of usage of different muscle groups, we strive to design an interaction which uses multiple muscle groups simultaneously, but also isolating them within a task and comparing them against each other.

Further, Mayer et al. [15] investigated mid air pointing in two conditions, the real world (RW) and in VR. They conducted two studies using a within-subject design, where participants (n=20) had seven rigid body markers attached to their arm used for pointing. The first study was concerned with pointing in the two aforementioned conditions. From this, they created pointing models, which were then tested in the second study, where the participants would test the models in VR compared to RW, with or without a cursor and with or without correction. The results show that there is a difference between how participants point in RW compared to VR. The participants were more accurate when using a cursor, while their models can improve accuracy even when not using a cursor.

Another study from Schwind et al. [20] investigated the impact of a virtual representation (avatar) on the accuracy in a within-subject design (n=24), and showed that when people point, they rely on their finger tips and not on their forearm and index finger orientation.

A common problem when using various pointing devices are quick micro movements, referred to as *"jitter"*, which introduce a lot of noise into pointing. To mitigate *"jitter"* and therefore improve accuracy, we looked into pointing assistance techniques. Here, Refai et al. [19] and Kocur et al. [12] investigate this issue and present possible solutions. From this, we chose to focus on a target expansion technique, which was chosen since it does not change the position of a pointer, nor does it affect the direction of a pointing path, which are important factors within our implementation. With this target expansion technique, the targets have a certain area around them which when hit, still counts as a successful target hit. The goal for this is to lower the overall difficulty of the game, increase performance, perceived competence and enjoyment of the game. While the area around the target is quite large and the user can easily notice that even a miss of the visible target is registered as a hit within that area, it should not be a negative factor on the user's experience according to [7].

3.2 Games

Further, we carried out a non-structured search of video-games that include or focus on pointing tasks. This yielded 19 unique games, half being in the **non-VR** domain, and the other being in the **VR** domain (for a list of all games, see Worksheet subsection (2.3)). From these, we identified new pointing tasks, to investigate the pointing task family [1] further:

- **Target-to-target**: pointing at one target after another [1]
- **Tracking-target**: pointing continuously at a target over a certain amount of time.
- **Search-point**: scan an environment to find particular entities.
- **Dragging-to-target**: movement of the input device in a certain direction. This is derived from games such as Chivalry Medieval: Warfare [21] where the user controls the position of their weapon with their mouse.

Among these, we chose to focus on Target-to-target, which was the most present pointing task. It includes examples from various different genres, such as the rhythm game Osu [17], where the user has to hit nodes matching a beat, or the arcade game Whack-a-mole [5], where the user has to to hit the moles as they appear from their holes.

Upon investigating the action levels of these types of the pointing tasks family, it became evident that they all share an ***initiation*** step, a crucial phase to successfully carry-out of a particular interaction. Here, we categorize the variations of this step into the following.

- **Activation**: physically pressing a button.
- **Time-Activation**: activation in cadence (the beat, time, or measure of rhythmical motion or activity).
- **Wait-point**: pointing in cadence.

Activation was the most common in this review, and can be considered a prevalent version of *initiation*, e.g. in shooter games, where the user has to point and activate to fire a gun at a target. Another variation of this is the *time-activation*, which is seen within rhythm games, where the user has to match their activation in accordance to a tempo (usually a music piece) such as the VR shooting game Pistol Whip [10]. Lastly, *wait-point*, in contrast to the aforementioned two, does not use any activation, but instead uses movement. This initiation is derived from the VR game Beat Saber [8], where the user has to flick their controller in accordance to the tiles approaching them, an example of this can be seen in Figure 2. This is a game known for its engaging way of making the users exercise, and in the context of action levels, an interesting way of creating an interaction that does not require a physical button press. However, though related to pointing, it does not carry the same intent as other games in this review, since the game is more concerned with movement in a direction at a certain time, than pointing itself. On the other hand, the task does share similarities to the **Dragging-to-target** pointing tasks of games like For Honor.

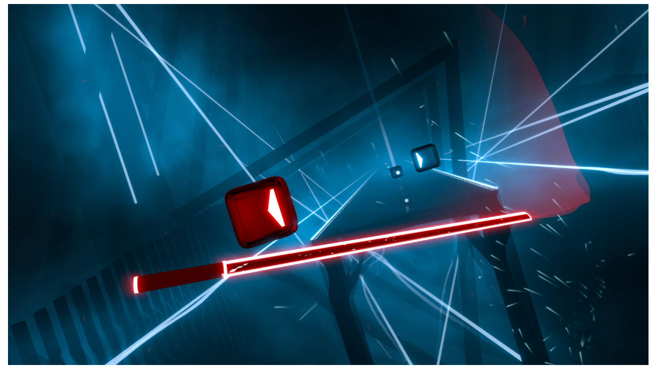

Figure 2. Screen capture of the game Beat Saber [11].

Continuing on from this notion, the use of non-activation was scarce in this state of the art review, and the ones using it was not concerned with pointing, but instead motion in a direction instead. Thus, we wish to explore an interaction in VR, focusing on the action level of pointing, while including a non-activation initiation using movement.

Lastly, it is worth mentioning that motion controller movement as an initiation has been seen before, most notably on the Wii console. While we wish to use the same notion of movement in our interaction, we also wish to offer a higher level of exertion by requiring a certain force of movement, which is something that Wii has a known flaw at. Specifically, in Wii tennis, certain exploits of the implementation could be taken advantage of, where a user could generate a high amount of virtual force by "flicking" the controller with their wrist instead of performing the intended swing motion emulating a tennis racket.

4 Interaction Design of Thrusting

4.1 Thrusting in general

Based on the research and state of the art review, and with the newly created classifications in mind, we set out to design a new way of pointing in VR, which facilitates exertion. The intention was for a potential use for either exercise games or rehabilitation purposes, following the definition from Laamarti et al. [13]. To accomplish this goal, we are including velocity as a variable for pointing, with the intention of turning a basic 3D pointing motion into pointing that also requires a forward-moving motion. We refer to this as a thrusting motion i.e, similar to a punching motion. Specifically, we define thrusting as a motion which starts with a contracted arm, with the hand close to one's shoulder, followed by a rapid outstretch of the arm, with the hand following a linear motion, creating a pointing direction.
This will change the action level of a simple move-point into a move-point-thrust. We acknowledge that there are several combinations and variations of an action level for a thrusting action, however, for this study we will explore

what we deem as the most simple action to perform based on the SSTA action level scheme [1][4] and the findings of Zhai et al. [22].

Converting the human mechanical motion of thrusting into VR is an important design decision as it directly influences the test design and results. Therefore, we will spend some extra time in the following subsections discussing our thought processes and expected outcome, as well as implications of our method of implementation.

4.2 Design of a thrusting motion

When designing thrusting, we first have to define when thrusting takes place in order to implement it. For this, we initially decided on a combination of criteria which had to be fulfilled:

1. **Velocity** of the hand has to reach a certain value.
2. **Acceleration** of the hand has to reach a certain value.
3. **Distance** between the subject (user) and their hand has to reach a certain length.

The intention is that when all three conditions are met, thrusting would have taken place i.e. thrusting would be triggered. However, the values of the conditions are dependable on the user in question and their performance e.g. varied arm lengths or strength.

The initial idea was to set the thresholds of *velocity* and *acceleration* so they would fit the participants performance using a virtual "boxing bag". The participant would punch in VR, where we would save their data and set the threshold. This would then be set at some value below their average during their performance. However, this could result in the participants having vastly different thresholds and therefore having difficulty performing the thrust. This also means that the participants could accidentally set the difficulty of thrusting too high or too low.
Therefore, we decided to put the thresholds to a static value across all participants to give everyone the same testing conditions and to make the data directly comparable. The thresholds were determined based on self-testing with a trial-and-error approach, with the goal of making it possible to reach the threshold while avoiding accidental triggers.

The last condition of *distance*, being from the starting position of the thrust to its final position, was something we could not choose a universally applicable value for.
To determine the distance variable, we needed to have a measurement of the user's arm length. Mayer et al. [15] did this by physically measuring the participants, using 14 measurements, as for their purposes more values than just an arm length were needed. We deemed this not being necessary for our calculations, which resulted in creating a calibration level to determine the participant's arm length. While this might be less accurate than the measurements used by Mayer et al.[15], it is sufficient for our purposes, while saving substantial amount of time during the testing and automating the whole process.
For the caibration, the user's arm length would be used based on their input, and a fraction of the length would be set as a threshold. This threshold would serve as the distance condition that has to be surpassed in order to achieve a successful thrust.

Finally, the most important part to design about the interaction was calculating the intended trajectory (pointing direction) of the thrust. Since this is a completely new pointing interaction, we designed three different methods of calculating the final pointing trajectory, based on:

1. **Wrist direction** - a straight line trace from the final position of the motion controller (at the end of a thrust) based on its rotation.
2. **Combined direction** - split between the wrist rotation and the shoulder direction.
3. **Shoulder direction** - Shoulder-to-hand position trajectory only.

These methods would be tested against each other in terms of accuracy, precision, exercise potential, agency and general preference.

This way, the interaction methods are designed to only afford one specific action, which when performed correctly, would be considered a thrust.

Additionally, the thrusting has a small cool-down which prevents the user from shooting. This was done due to several reasons. Firstly, the implementation point, to make the thrusting register in the first place and differentiate between each thrust, while not making the program register thrusting multiple times simultaneously. Further, to prevent the user from shooting rapidly or by mistake and more importantly to force the user to take a bit of time to perform each thrust correctly. This also ties into giving the user a "penalty" for thrusting incorrectly or too frequently, similarly to a paper by Banovic et al. [3].

4.3 Thrusting calibration

Here, we will describe the aforementioned calibration scene in more detail, in the scope of implementation. In the scene, the user would have to set their shoulder position in the 3D space, as well as stretch out their arm in four directions. From this, the average of the four distances (from the shoulder to their hand, in the four directions) would be used to estimate their arm length. This value is multiplied by 0.8, meaning the user would have to move their hand 80% of their full arm length to fulfill the distance condition. This is illustrated in Figure 3.

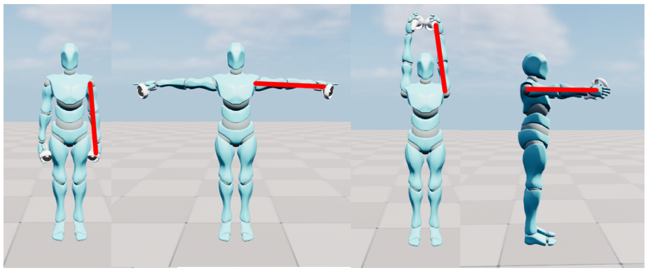

Figure 3. Illustration of the four poses the user would have to make for calibrating their arm length. Red line symbolizing the length we get from shoulder position to controller position.

Further, setting the shoulder position would place a collider [1] on their shoulder, that would encompass the aforementioned 80% of their arm length. To perform the thrust, the hand position would start at the shoulder and within a certain distance of the center of the collider, then leave the collider completely with high enough velocity and acceleration, as shown in Figure 4. To perform another thrust, the hand has to re-enter the collider again within the center distance.
With this system we force the user to perform very specific, "full" motions when thrusting, i.e. going from a fully contracted arm to a fully stretched arm.

5 Game design

With the specific definition and requirements of thrusting in mind, we designed a VR game in Unreal Engine 5 [9] that would include this new interaction within a video game context.

5.1 Game mechanics

In its core, the game is about the player shooting targets by pointing the controller at them and thrusting their hand forward towards the them to inflict damage, therefore eliminating the target. The player's dominant hand takes the appearance of a pistol, which would shoot projectiles and thus provide feedback on the current pointing direction. We decided to not include an avatar, instead only showing a pistol at the position of the user's hand/controller, preventing the possible loss of agency and influence on pointing [20]. Additionally, this avoids the possible implementation issues that come with attaching a skeletal rig to the player, which is problematic due to how Oculus Quest and its controller

[1]An invisible in-game shape that can trigger an action based on various conditions

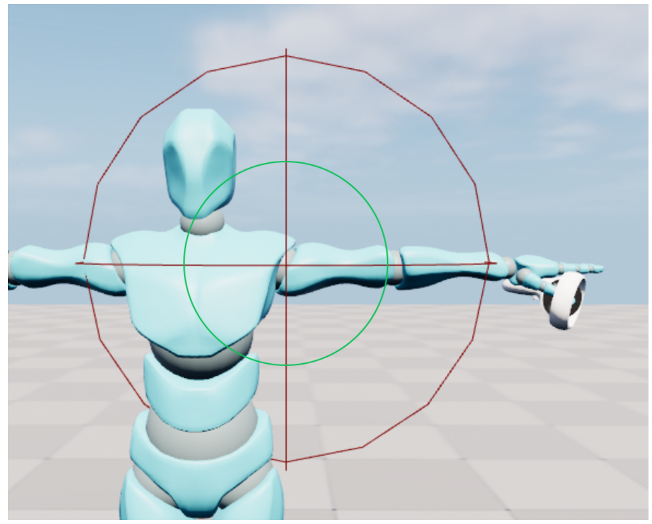

Figure 4. Illustration of the desired shoulder collider position (red) and the inner distance from center (green) represented as a sphere for visual aid.

calculate their position within the virtual world. The thrusting motion has a set of conditions that have to be reached, as described in Subsection 4.2, which will cause the pistol to fire a projectile, therefore no button press activation is required.
The player is being moved through the game-world, with the targets appearing in front of the player, either on the right or left side of the player's path, spawning in increasing frequency. The goal is to eliminate all the targets in the game.

5.2 World design

The visual design of the game world puts the player on top of a moving train, therefore motivating their movement through the world. To make the environment more visually interesting and give the player a sense of movement and own speed, the train moves through a futuristic high-rise city situated in a desert, as seen in Figure 5. The targets are mannequins with a pistol shooting at the player, situated on top of a long train station structure within the city. This is done to make the game visually akin to a more traditional VR game experience.

The game also features a number of visual and sound effects, which primarily serve as a feedback for different in-game events. Both the player and the targets shoot projectiles at each other, which resemble laser gun projectiles commonly used in the sci-fi genre. This means they travel at speeds slow enough for the player to visually keep track of, providing feedback of their shooting direction. A gunshot sound-effect is played with each successful player's shot, and a "laser gun" sound is played for the shooting enemies to differentiate the two. The sound effect and the projectile is

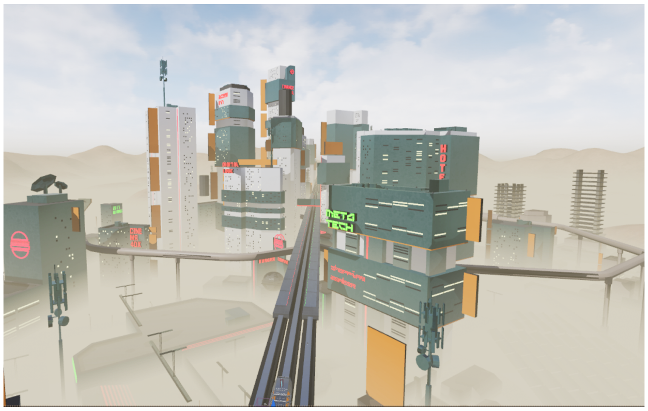

Figure 5. Screenshot of the in-game environment.

also an instant feedback of a registered thrust.

A successful hit of the target by the player is shown with a burst of sparks at the hit location, accompanied by a sound effect. Additionally, the target itself starts a ragdoll physics simulation upon being hit, implying successful elimination. Both can be seen in Figure 6. These feedback visuals are also supposed to be rewarding for the player, giving a sense of satisfaction and entertainment value.

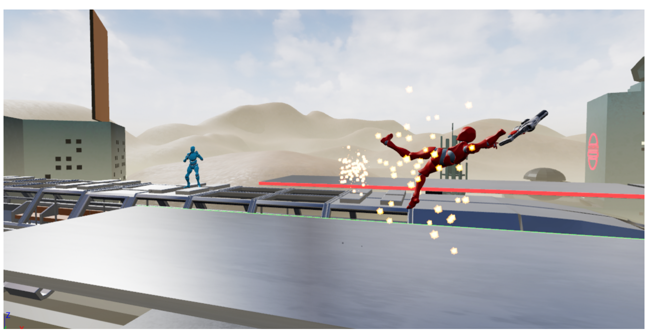

Figure 6. Visual feedback of a hit on a target. Note that the blue figure serves only as a placeholder for the player and is not present during gameplay.

When a target spawns, it is accompanied by an effect similar to a lighting, to highlight where within the world they have spawned, also using a sound effect.
Further, the train which the player is standing on produces an edited sound of a metro train for ambiance.
And finally, the game level includes a sound track that adds to the futuristic and dynamic atmosphere of the game.

5.3 Target-to-target properties

The design choices were made based on the target-to-target taxonomy explained in Section 2. Here, we will go through each property of target-to-target pointing [1] and present arguments for each choice.

First of all, we expanded on the taxonomy with a property called ***Quantity of activation***. This means how many times does an action need to be performed to activate a desired event within the program. The amount of activations is not a set value, rather a dynamically calculated modifier based on user input, e.g. the damage is based on an acceleration of the weapon at the time of the shot, making every hit have a different damage and the user could be rewarded when performing a fast hit with high acceleration. For example, while some events might only require a *single* activation, such as flipping a light switch, others require multiple *repeated* activations, such as a switch for dimming the light, or double-clicking to open a folder. Additionally, the repeated activation can be based on a *threshold*.

Within our game/solution we aim to use *single quantity of activation*, eliminating the target with a single successful hit, since we are more interested in the accuracy of pointing.
For ***Target quantity*** we chose 15 targets to appear within the session. This was chosen to keep the testing time within 2 minutes per condition while getting sufficient data from each participant.
Within ***Target appearance quantity*** we chose both *Single target* and *Multi-target*, which dynamically changes throughout the game. While in the beginning, for the first five enemies, there can be only a single target present, for the second five enemies there can be two targets (if the player doesn't eliminate the first one quickly enough), and for the last five enemies there could be up to all five of them active at the same time.
For the ***Target location placement*** property we chose a variation of *Predetermined location*, where the targets appear at a specific locations, but in random order. This was done to decrease predictability while having the same conditions between participants.
The ***Target lifetime*** was set to be *limited*, with the targets disappearing when they pass the player. The limit is set to further incentivize the player to perform the thrusting motion faster and more often, increasing exertion.
Further, we decided not to include any ***Distractors***, as their function is to increase mental task difficulty and perform search, which is not the goal of this program.
Finally, we implemented the ***Target movement*** to be *Moving* towards the player (from the player's relative perspective), to add a dynamic element into the gameplay and justify the limited target lifetime.

6 Methodology

In this section we will describe the changes which the design of the thrusting and the game went through over the iterations.

6.1 Preliminary study

A preliminary study (further referred to as pre-study) was conducted in order to test the three implementation methods described in Subsection 4.2, as well as the design of the experiment. The design was within-subject and involved 5 participants from Medialogy at Aalborg University, all of which were right-handed.
This version of the game included a calibration sequence, as described in Subsection 4.3, to calculate the starting position for thrusting as well as the distance from this position to a successful thrust, for each player.
Each interaction method was assigned a condition, namely wrist direction being condition 1, combined direction being condition 2 and shoulder direction being condition 3.
Further, the game included a color-changing indicator on the tip of the pistol, which would indicate when the player puts the controller in the calibrated starting position by having green color, and gradually turning more into red as the controller moved further from this starting position.

6.1.1 Pre-study procedure.

During the experiment, the participants filled out a consent form followed by an Simulator Sickness Questionnaire (SSQ) [14] questionnaire. After this, the participants were asked for their dominant hand, and they were briefed about a calibration level and given the VR headset and controllers.
When calibration was successful, they were given some time to familiarize themselves with the shooting interaction until they felt comfortable with it. Then, they were tasked to shoot a specific target to initiate the game level described in Subsection 5.2. Within the game level itself, the task was to attempt to shoot all the enemies which appeared in increasing frequency.
Upon completion of the game level, the participants were given a modified Agency and Ownership questionnaire (9 questions, 7-point Likert scale), see Worksheet Section (2.3). Since the calibration data was saved, the second and third condition were carried out only for the game level, also followed by an Agency and Ownership questionnaire.
Additionally, a logging script recorded data during the gameplay such as frames per second (FPS), acceleration, successful hits and critical hits (i.e. directly hitting the target mesh).
Further, a second SSQ questionnaire was administered to check whether their physical discomfort levels have changed due to the VR experience.
Finally, a semi-structured interview was conducted, which investigated whether the participants noticed the differences between each condition, whether they have any preference between them (if it applied) and whether they paid attention to a color indicator on the in-game pistol. They were also given the opportunity to give any other feedback.

6.1.2 Pre-study results.

The testing uncovered some flaws with our implementation which needed to be addressed, resulting in changes of the fundamental logic of the thrusting system. The system was mostly successful in forcing the user to perform the specific thrusting behavior. However, in some cases the users would perform the interaction in ways the system did not afford, meaning it would not register the motion as thrusting, in turn leading to frustration and a generally poor rating of the system.

The most notable issue was due to the spherical collider used. The participants where asked to place the collider either at their shoulder or more in front of their chest center around the sternum, which heavily affected their accuracy and the system's predictability. To elaborate, when the collider was placed too low, the resulting pointing trajectory for condition 2 and 3 would point upwards, and if the collider was placed too high, the trajectory would in general point downwards. The reason for this was that the starting point for calculation of the trajectory was the center of the collider.

Overall, the participants had a hard time performing the thrusting motion in the desired manner, resulting in failed attempts, especially as the game progressed to the more intense parts. Additionally, the participants had trouble controlling the horizontal and more-so the vertical trajectory of the projectiles.

6.1.3 Changes based on the pre-study findings.

Based on this, we made the following changes to the implementation of the thrusting and design of the experiment:

- The technical implementation of how shooting is calculated was completely reworked to be more robust while not being constrained to a motion based on specific coordinates (which were determined during the calibration level). This was done due to the original implementation's physical constraints, unreliability and unpredictable shooting directions.
- The Agency and Ownership Questionnaire has been changed to include less items in order to reduce the strain on the participants and the testing time per participant.
- The instructions were changed to include a clear description and demonstration of the interaction to inform the participants on the differences and proper use of each condition.
- This in turn altered the focus of the interview, which no longer included a question about noticing changes between the conditions, as the participants were told the differences between the conditions during the experiment briefing.

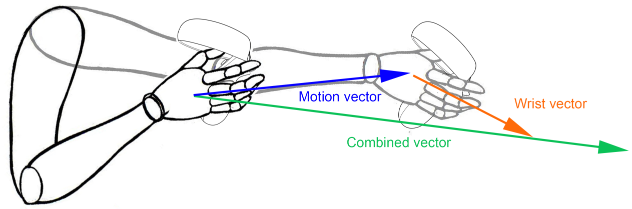

Figure 7. Illustration of the calculation of the motion vector, where the vectors shown would then be added up to a final motion vector.

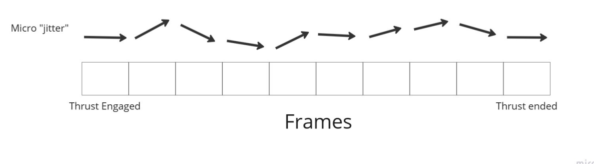

Figure 8. Illustration of the calculation of the motion vector, where the vectors shown would then be added up to a final motion vector.

- The calibration sequence was replaced with a training level, which would be included before each condition instead of only once at the start.
- The color indicator on the tip of the pistol was no longer needed and we removed this functionality.

6.1.4 Second iteration of thrusting. With these results in mind, we designed a new implementation method, which eliminates the collider altogether, therefore minimizes human error during calibration. It also removes the limitation of the system where it only afforded a specific thrusting technique.
The new system relies on calculating the pointing direction based on the vector of the wrist rotation and the vector of starting-to-final hand position, as seen in Figure 7, instead of the previous shoulder direction. This time it is created based on displacement in space instead of exiting of a collider, now being referred to as *motion direction* instead of shoulder, as the shoulder no longer plays a role in the calculation, affording a new range of motion.

Here, a threshold was set of when the hand has been moved a certain distance in space over an amount of frames (displacement) then the thrust would be triggered. We would create multiple vectors over 10 frames and use vector addition to generate the final motion vector, as seen in Figure 8. In addition, we would combine the secondary condition of the acceleration threshold, kept from the first iteration, and when both conditions are met the thrust would be triggered.

Additionally, vector addition is used to generate a new vector which has a path between the wrist rotation vector and the motion vector called direction vector. This way the new direction vector could have weights for the two vectors it was based on, allowing control of the ratio of the final trajectory.
This way, we would keep our original three interaction methods described in Subsection 4.2, only calculated differently, where they are now introduced as the following:

1. **Wrist direction** - when thrusting is registered, the final direction is calculated only based on the rotation of the wrist/motion controller (same as with the previous iteration).
2. **Combined direction** - where the weight would be set to consider both vectors equally, namely 0.5 wrist direction vector and 0.5 motion direction vector.
3. **Motion direction** - will only use the initial motion vector generated and not take the wrist rotation vector into consideration (no longer shoulder direction).

All three interactions are using the following formula: $(1 - k) * v1 + k * v2$ where k is the weight being changed, changing the conditions, and $v1$ and $v2$ are the wrist and motion direction vectors. For example, to rely only on the wrist, the weight would be set to 0.

It is worth mentioning that the new system also affords different ratios between the motion vector and rotation vector, e.g. 80% wrist direction and 20% motion direction. However, for consistency we kept it similar to the first iteration.

This new method affords unique thrusting techniques the participants would be able to use instead of constraining them to a set motion. This way we will be able to observe how precisely the participants perform the thrusting motion and their possibly different techniques. Additionally, we will observe whether they use different techniques between the conditions.

6.2 The study

With the new thrusting implementation and altered test design, we conducted a larger-scale study to yield more results. The study design was set as within-subject, with 3 test conditions:

- **Condition 1**: Wrist direction only
- **Condition 2**: Combined direction - a split of 0.5 and 0.5 between wrist and motion.
- **Condition 3**: Motion direction only.

The order of the conditions was changed for each participant in a way so that each possible combination of the

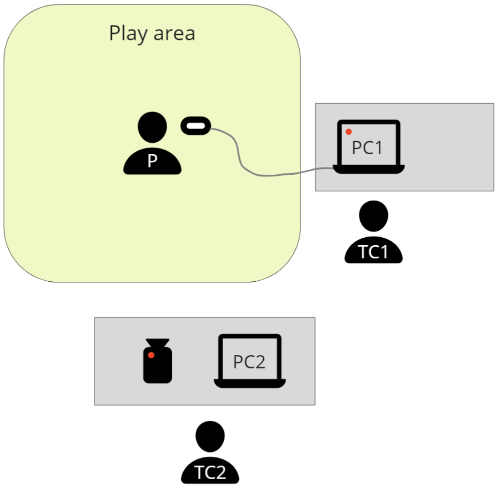

Figure 9. Testing setup, with a participant (P) being in the Play are during the VR part, or using a laptop (PC2) to fill out the questionnaires. One test conductor (TC1) was present to operate the laptop (PC1) running the VR game, while the other test conductor (TC2) provided instructions for the participant and conducted the interview.

conditions was used the same amount of times, where after all six possible combinations, the rotation would start over and we would gather an amount of participants based on this concept. To see each possible combination, see Worksheet Section (2.3).

6.2.1 Participants. The study included 24 participants (23 right-handed, 1 left-handed) recruited at the CREATE campus of Aalborg University. All of the participants were from the Department of Architecture, Design and Media Technology.

6.2.2 Apparatus. The hardware used for this test was a Lenovo Legion 5 laptop, with an AMD Ryzen 5 CPU and Nvidia GeForce 3070 graphics card. The VR headset used was the Oculus Quest 2 with the original motion controllers. A separate laptop was used to fill out the SSQ and Agency and Ownership Questionnaires. A camera was positioned to record the participants during the test in VR. The layout of the test setup can be seen in Figure 9.

A logging script was used to log the FPS, accuracy, acceleration, displacement and X, Y, Z positions of the controllers. Additionally, a separate log was created that kept track whether the participants pressed any buttons.

6.2.3 Procedure. First, the participants were asked to fill out a consent form, SSQ [14] and short written explanation of the experiment. They were asked about their dominant hand and then given verbal instructions together with an example demonstration of how to perform the thrusting in order to shoot. The instructions can be seen in the Worksheet Subsection (2.2.1). Further, they were informed about the current testing condition and which movement is taken into consideration. Additionally, they were told about their perceived movement in the game level, as well as the fact that they do not need to press any buttons on the controllers.
The test in VR started with a Practice level, where the participants got to shoot at targets with instant feedback (red dots where the projectile landed) at different distances to get familiar with the current condition until they were comfortable with the interaction.
This was followed by the game level, in which they had to shoot 15 enemies appearing in an increasing frequency after every 5th enemy (8s, 4s, 2s).
When the game level was completed, they were given two Agency and Ownership questionnaires (4 questions, 7-point Likert scale), one for the Practice level and one for the Game level.
The Practice level followed by the Game level and two questionnaires were repeated 3 times in total. At the end, a second SSQ was filled out and a short interview conducted. The interview focused on preference between the three conditions as well as their reasoning. Further, during the interview the participants were asked which condition they think they performed the best in.

7 Results

This section will present the quantitative and qualitative results from the experiment, in the aforementioned order. This data was analyzed and visualized using R-studio [18].

The results from the SSQ questionnaire showed that no participant experienced discomfort which would exclude their data from the analysis.

7.1 Frame rate

The frame rate across all participants averaged 64 frames per second (FPS), with a median of 67 (IQR 74 – 56). The FPS for each participant, can be seen in Figure 10.

7.2 Agency and Ownership questionnaire

The mean results of the questionnaire will be presented per question, for both the practice (P) and the game (G) scenario, the results can be seen in Figure 11.

The agency of each condition contains the following values:

- 1. P Agency: 4.40 (95% CI 3.69 – 5.11),
- 1. G Agency: 4.38 (95% CI 3.68 – 5.07)
- 2. P Agency: 4.30 (95% CI 3.74 – 4.86)
- 2. G Agency: 4.38 (95% CI 3.82 – 4.93)
- 3. P Agency: 2.75 (95% CI 2.20 – 3.30)
- 3. G Agency: 2.67 (95% CI 2.01 – 3.32)

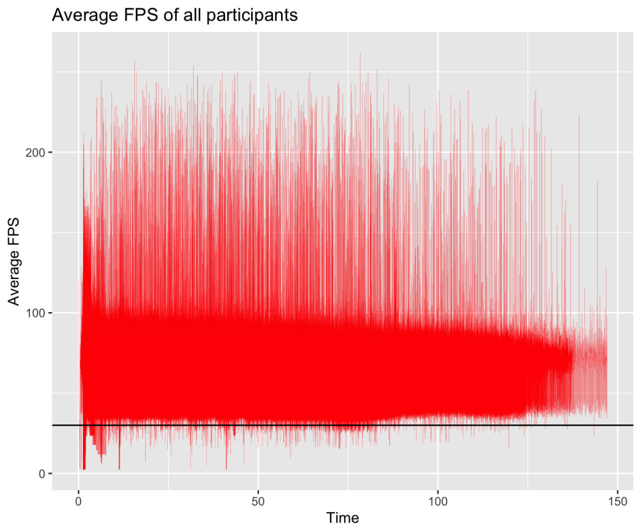

Figure 10. The average FPS for all participants. The black line represent a 30 FPS boundary.

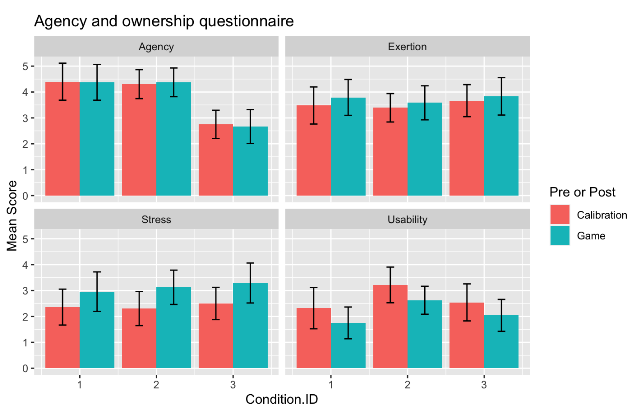

Figure 11. The questionnaire mean scores for each condition, with confidence intervals.

Further, the exertion of each condition contains these values:

- 1. P Exertion: 3.48 (95% CI 2.76 – 4.20)
- 1. G Exertion: 3.79 (95% CI 3.10 – 4.48)
- 2. P Exertion: 3.39 (95% CI 2.84 – 3.94)
- 2. G Exertion: 3.58 (95% CI 2.93 – 4.24)
- 3. P Exertion: 3.67 (95% CI 3.05 – 4.29)
- 3. G Exertion: 3.83 (95% CI 3.11 – 4.56)

The stress of each condition contains the following values:

- 1. P Stress: 2.36 (95% CI 1.67 – 3.05)
- 1. G Stress: 2.96 (95% CI 2.20 – 3.72)
- 2. P Stress: 2.30 (95% CI 1.65 – 2.96)
- 2. G Stress: 3.13 (95% CI 2.46 – 3.79)
- 3. P Stress: 2.50 (95% CI 1.88 – 3.12)
- 3. G Stress: 3.29 (95% CI 2.52 – 4.06)

And finally, the usability of each condition contains the values:

- 1. P Usability: 2.32 (95% CI 1.52 – 3.12)
- 1. G Usability: 1.75 (95% CI 1.14 – 2.36)
- 2. P Usability: 3.22 (95% CI 2.53 – 3.91)
- 2. G Usability: 2.63 (95% CI 2.08 – 3.17)
- 3. P Usability: 2.54 (95% CI 1.83 – 3.26)
- 3. G Usability: 2.04 (95% CI 1.43 – 2.66)

7.3 Accuracy

The accuracy can be seen in Figure 12, which shows the total amount of shots, their hits and the amount they critically hit. The data from the graph can be seen below:

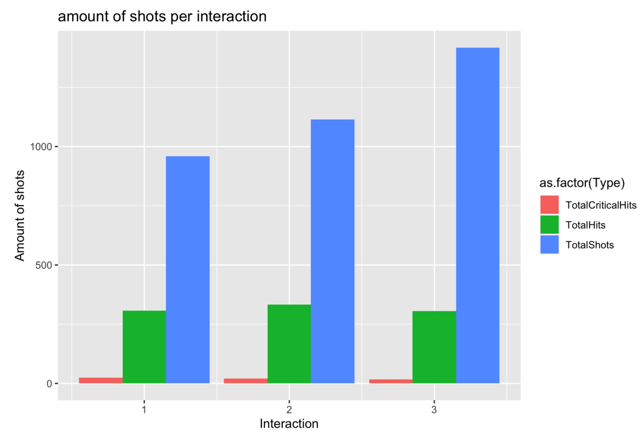

Figure 12. The amount of shots, the amount of hits and the amount of critical hits.

- Condition 1: Shots = 959, Total Hits = 308 (32% accuracy) and Crititcal Hits = 24 (7% accuracy)
- Condition 2: Shots = 1115, Total Hits = 333 (29% accuracy) and Crititcal Hits = 21 (6% accuracy)
- Condition 3: Shots = 1418, Total Hits = 305 (21% accuracy) and Crititcal Hits = 16 (5% accuracy)

A Shapiro-Wilk test of the shots did not show normality ($W = 0.932, p < 0.001$), the Kruskal-Wallis test showed significance ($Chi\ Square = 24.594,\ df = 2,\ p < 4.565e-06$) and a pairwise Dunn test only showed significance between condition 1 & 3 ($p < 2.250214e-06$) and condition 2 & 3 ($p < 1.713968e-02$)

A Shapiro-Wilk test of the hits did not show normality ($W = 0.762,\ p < 0.001$), the Kruskal-Wallis test showed no significance ($Chi\ Square = 5.0872,\ df = 2,\ p < 0.07858$).

A Shapiro-Wilk test of the Critical hits did not show normality ($W = 0.802, p < 0.001$), the Kruskal-Wallis test showed no significance ($Chi\ Square = 1.6635,\ df = 2,\ p < 0.4353$).

7.4 Distance to target

The distances for each condition towards the target when hit can be seen in Figure 13. The IQR from the graph can be seen below:

- Condition 1: median of 1467 (IQR $2027 - 1011$) with a mean of 1521
- Condition 2: median of 1695 (IQR $2152 - 1124$) with a mean of 1625
- Condition 3: median of 1090 (IQR $1601 - 755$) with a mean of 1231

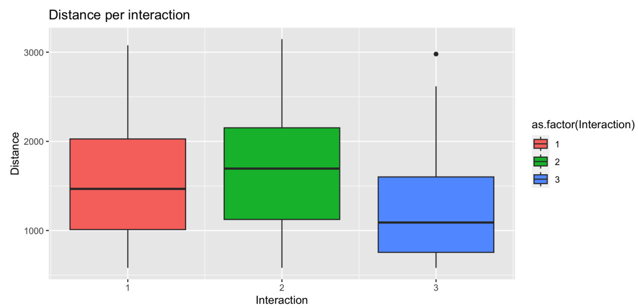

Figure 13. The distances per condition.

A Shapiro-Wilk test of the distances did show normality ($W = 0.968,\ p < 0.0667$), a one way ANOVA test showed significance ($F\ Value = 15.87,\ p < 2.13e - 06$) and Tukey Honest Significant Differences test showed significance between condition 1 and 3 ($p < 0.001$), and 2 and 3 ($p < 0.001$).

7.5 Acceleration

The acceleration can be seen in Figure 14. The IQR from the graph can be seen below:

- Condition 1: median of 22.8 (IQR $30.8 - 17.7$) with a mean of 27.6
- Condition 2: median of 23.6 (IQR $34.1 - 18.2$) with a mean of 29.2
- Condition 3: median of 23.2 (IQR $31.2 - 18.1$) with a mean of 28.2

A Shapiro-Wilk test of the distances did not show normality ($W = 0.871,\ p < 0.001$), the Kruskal-Wallis test showed no significance ($Chi\ Square = 0.16572,\ df = 2,\ p < 0.9205$).

7.6 Displacement

The displacement can be seen in Figure 15. The IQR from the graph can be seen below:

- Condition 1: median of 6.38 (IQR $10.6 - 4.57$) with a mean of 8.11
- Condition 2: median of 6.33 (IQR $9.98 - 4.68$) with a mean of 8.09
- Condition 3: median of 7.06 (IQR $10.9 - 4.71$) with a mean of 8.33

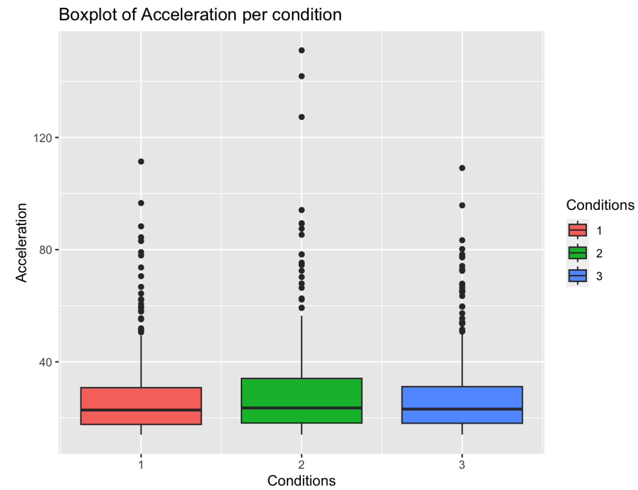

Figure 14. The accelerations per condition.

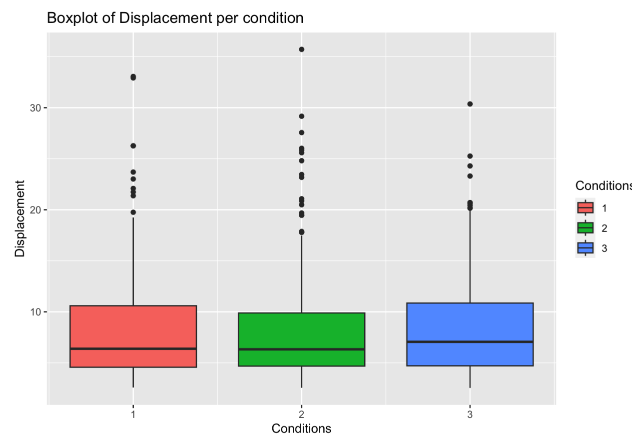

Figure 15. The displacement per condition.

A Shapiro-Wilk test of the distances did not show normality ($W = 0.897, p < 0.001$), the Kruskal-Wallis test showed no significance ($Chi\ Square = 2.2671,\ df = 2,\ p < 0.3219$).

7.7 Movement

The movement covered in each condition can be seen in Figure 16, whose values can be seen below:

- Condition 1: 13.89300
- Condition 2: 10.58366
- Condition 3: 14.10222

An Anderson-Darling test of the movement did not show normality ($A = 94440,\ p < 0.001$), the Kruskal-Wallis test showed significance ($Chi\ Square = 2421.5,\ df = 2,\ p < 2.2e-16$) and a Dunn's Kruskal-Wallis Multiple Comparisons showed significance between all groups ($[1 - 2]\ p < 0$, $[1 - 3]\ p < 6.18e - 06$, $[2 - 3]\ p < 0$).

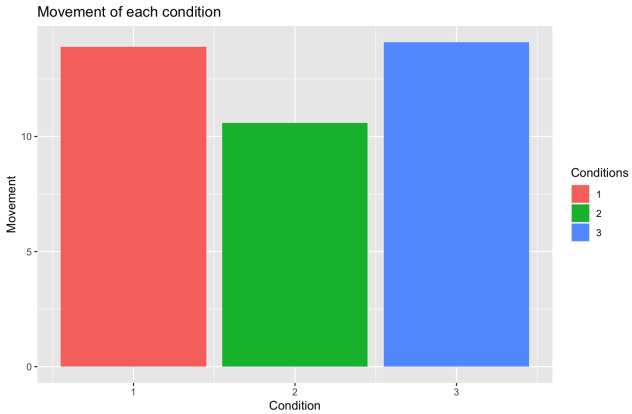

Figure 16. The movement per condition.

7.8 Correlations

From the quantitative data gathered, some patterns have emerged. These patterns and the strength of them are represented as a correlation matrix on Figure 17 below.

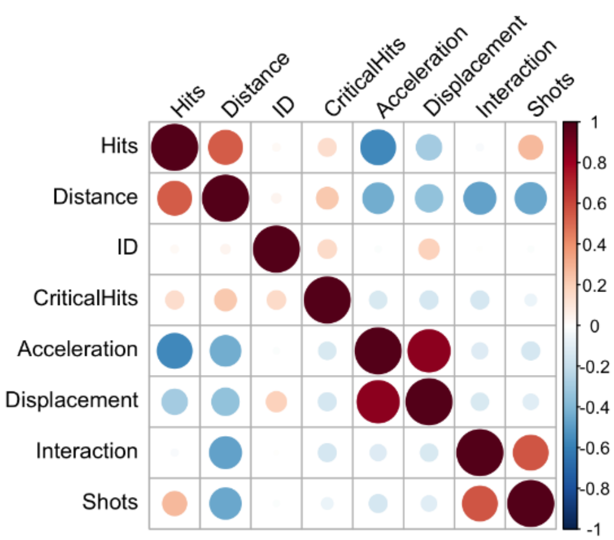

Figure 17. Correlation Matrix, where 1 (dark red) is representing an increase in the variable and -1 (dark blue) representing a decrease.

In Figure 17 the colored circles are shown as to how the variables on the Y-axis are related to the variables on the X-axis e.g, as there is an increase in **Hits**, there is also an increase in **Distance** and a decrease of **Acceleration**. However, a high correlation to the value itself is also being displayed (dark red color going diagonally through the matrix).

7.9 Interview Results

Further, we analyzed the results gathered from the interviews. Most people preferred condition 2 (14 participants), second most was condition 1 (10 participants) and the least preferred was condition 3 (3 participants). The perceived performance was rated with condition 1 as best (11 participants), then condition 2 (8 participants) and lastly condition 3 (5 participants).

Additionally, from the interviews, the participants would often comment on having difficulties controlling the vertical trajectory of their shot when using condition 3, with comments as with Participant 23:

> "...I had a lot of trouble with directing it up or downwards."

However, it is worth noting that there seems to be a tendency for the participants to move their arm slightly upwards when trying to thrust forwards.

8 Discussion

The results from Subsection 7.3, proved to show significance in terms of amount of shots between wrist direction & motion direction and combined direction & motion direction. Here, the motion having an additional 459 shots compared to wrist, but interestingly they had a similar amount of hits. This meant that their accuracies differed vastly as well, scoring an accuracy of 32% and 21% respectively.

The distances from Subsection 7.4 proved to be significant between wrist direction & motion direction and between combined direction & motion direction. Comparing the former, wrist had a higher distance to the targets than motion, which indicates that it was easier to use, since participants shot targets at larger distances with a higher accuracy. Comparing the latter as well, showed that combined contained the largest distances, which interestingly compared to its accuracy, though not significant, scored 29%.

Continuing from accuracy, the movements from Subsection 7.7 proved to be significant between all interactions. Participants moved the most within wrist & motion, and the least in combined.

Upon inspecting this, in relation to the aforementioned results, wrist and motion seems to be the most physically demanding ones, with the former being the most accurate of the two. Motion seems to be the hardest to use, due to distances being the smallest and amount of shots being the largest, indicating that participants could not hit targets reliably. However, as varied as these two interactions are, it is interesting that the amount of movement is roughly the same. Hence, wrist can be considered as a "better" version compared to combined, from the lens of accuracy and exertion. Combined on the other hand, had the largest distances to the targets, had the second highest accuracy, and the lowest amount of movement. This indicates that combined was the easiest to use, since participants moved the least and while still being relatively accurate.

Comparing this to the participants' answers in the Agency Questionnaire, see Subsection 7.2, certain patterns emerge. Wrist and combined scored similarly in terms of agency, with motion being the worst, which aligns with the notion of accuracy mentioned earlier. Interestingly, in terms of exertion, all interactions resided within the same boundary. This contradicts the difference in movements, indicating that participants did not perceive exertion to be different even though it was. In terms of usability, combined was the best and wrist the worst. Inspecting this in relation to accuracy results, there seems to be a difference between performance and preference. Though wrist was the one participants performed the best in, it is still the one they preferred less. This indicates that the combined direction, which calculates the trajectory based on a combination of wrist and motion, seems to be the overall best, considering the preference scores and relatively high performance. However, wrist and motion can be suitable candidates for programs that need a certain level of exertion, since participants did not notice a difference within the exertion in the Agency Questionnaire. Nonetheless, there is a trade-off between the two, since one scored a higher agency than the other and vice versa with usability, which should be considered, but if accuracy is the focus, wrist seems the best of the two.

With this in mind, we will now explain some limitations and improvements of the program, and further iterate on the participants' preferences based on the interviews.

8.1 Pre-processing

Generally, all of the interactions were hard to control due to the sensitive nature of how the trajectory was calculated. Since the count of frames used was relatively low, the slightest movements had an influence on the direction (jitter) in which the participant would shoot, which sometimes lead to unexpected behavior.

One solution to this would be to average the movement out further, using a Dynamic Weighted Moving Average Vector, which calculates a new vector given two other vectors (motion direction and wrist direction) and a distance (how far can the two vectors be apart), which it adjusts based on two weights. This allows to define the behavior given small noise in the sample and vice versa for large noise. This could make the program less sensitive within the game's update loop. Therefore, small movements could be accounted for as passive user behavior and not as a sensible inclusion to the thrusting direction calculation.

Another possible solution to this was outlined by Zhang et al. [23], who investigated jitter in pointing with eye-tracking. They introduced three solutions to this, being Force field, Speed reduction and Warping to target center, with Speed reduction showing the best results as a solution for this problem.

Addressing this issue would be the main focus of our future work to improve the current program.

8.2 Assistance Techniques

Target expansion was included to make it easier for the participant to hit the target, by including a large hit-box. However, though it eased the difficulty, some participants found it difficult to assess themselves in terms of how accurate they were. This could give them a false sense of feedback, since they are less concerned with how they are, and more about whether they hit or not.

Similarly, the inclusion of a cursor or a laser pointer projected onto the enemy could improve their accuracy, similar to the findings of Mayer et al. [15]. It is a promising avenue to investigate, since this could be used with the external assistance technique mentioned from Refai et al. [19] called target locking. Using these in conjunction, one could make a system that makes aiming less sensitive, so the the program would adjust their aim based on proximity to the enemy from the cursor. Thereby making the participants hit more frequently.

8.3 Wrist angle threshold

The system itself would restrict the hand rotation during condition 1 (wrist direction), where the thrust would not be triggered if the rotation was too great. This was implemented as it was deemed that triggers in more extreme positions would only confuse the participants as to why the shooting trajectory would be "too far" off target.

On the other hand, allowing for extreme angles would give the participants feedback to what they are doing wrong as the trajectory becomes more "sideways" than intended, as well as making each thrust easier to trigger, but more inaccurate.

8.4 Upwards locomotion during thrusting

According to our findings, only using the motion direction to calculate trajectory was the worst performing and the least favorable interaction among the participants.

There appeared to be a tendency for the participants to fire the projectile more upwards than they accounted for as described in Subsection 7.9.

The cause of this unintentional upwards trajectory seems to be two-fold. One, Participant 7 said that it is difficult to keep track of the position of one's arm/hand when wearing a VR headset, hindering the hand-eye coordination.

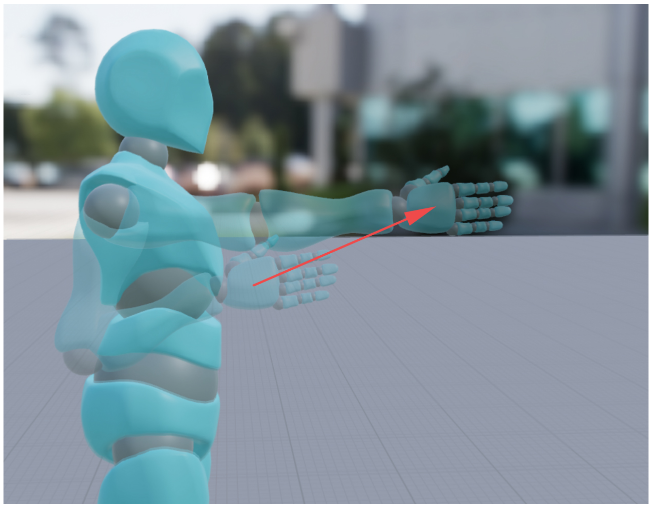

Figure 18. Hand moves up when arm is being stretched out.

> *"It is very hard to have that connection from your viewpoint to how exactly you're moving your arm. To make the thrust, it's very hard to connect, those two angles."*

Two, a possible explanation could be inherent to human locomotion, where the hand moves upwards when stretching one's arm out, as seen in Figure 18. If that is the case, it is clear that this upwards motion goes completely unnoticed by the participants.

To compensate for this, the 10 frames we use to calculate the motion direction might not have been enough information to include a potential vertical correction by the end of the thrust, therefore, only capturing the initial upwards angle of an overall motion curve.

Alternatively, it could be that with the wrist direction, this missing information would not have an impact on the accuracy as the wrist rotation works independently from motion in terms of trajectory. Therefore, the rotation would compensate for the vertical error and the issue might not be influencing the wrist direction condition.

Thirdly, as found by Zhai et al. [22] and stated in Section 2, when designing a pointing device, one should consider multiple muscle groups, and use their best performances in conjunction together. Though our system is not a physical device, it can still be relevant to consider, which would explain why combined direction performed as it did.

8.5 Weight ratio of vector calculation

It is to be said, that the wrist direction and the combined direction were the best overall systems according to our findings, while having mostly similar results.

This could mean that the optimal setting which could yield the best results could be a split of weights that is more favorable towards the wrist vector (wrist direction) than the motion vector (motion direction) i.e., the optimal combined direction could be in the likes of 0.8 wrist vector weight and 0.2 motion vector weight, instead of the current 0.5 split between them.

Using a coefficient within this wrist-favored ratio in combination with the suggested improvements above, the system could offer a generally better user experience with a more reliable triggers and trajectory.

8.6 Thrust trigger issues

All systems however, had issues with reliably registering intended thrusts, as multiple participants had difficulties thrusting, and we believe this to be the cause of a generally low usability score.

To improve this, individualized thresholds based on calibration could make the system more reliable to use for all users. Similarly, the system could provide a "settings interface" which could afford the user to adjust the values to fit their preference.

On the other hand, this was decided against with this implementation of the system, as described in Subsection 4.2.

It is also worth noting that there is a possibility of an undiscovered flaw in the implementation or the game-world which could be causing this issue.

8.7 Alternative interaction approaches

The arm movements the participants made during the experiment were mainly similar. This was thanks to the briefing, as we would demonstrate to them what we meant by thrusting, by showing them a punching motion.

The participants were instructed to perform the thrusting motion as instructed, as well as to try out different ways of thrusting until they found a method they would feel comfortable with during the practice scene.

The default response, however, was to mimic the punching motion we showed the during the briefing, repeat the action at all three practice target distances, then proceed to the game session.

Some participants, would instead experiment with various methods, some even changing their approach depending on which condition they were in e.g., Participant 15 would use a conventional punching motion when they were using the wrist direction only. However, when they would be using the motion direction only, they would change their method, where they would lower their elbow next to their hip, with the hand facing up. Then they would thrust their arm forward where their elbow and hand would stay on the same vertical level throughout the thrusting motion. This can be seen in Figure 19.

This approach would reflect on their performance results as this participant would have one of the best accuracy's through all participants.

Figure 19. Participant 15 at the beginning of the thrusting motion (left), versus at the end of the motion (right).

When the participant was asked during the interview of how they came up with this method, they responded:

> *"I think once I started doing that, it felt kind of natural and that was just like more fun to do. Yeah. Yeah. So, I kind of just like tried different things. Cause you told me it didn't really matter what you, what you used and, you know, so I just tried that and that felt like the way to go."*

Multiple participants were observed to hold their hand high and rotate it 90 degrees to the side, facing their palm downwards during the combined direction, seemingly improving their performance.
Participant 18 approached the experiment analytically, focusing on how the system responded and how to use that feedback to modify their next action. This was the optimal approach, since their accuracy and distance to target was relatively high compared to the other participants. An quote from the interview can be seen below:

> *"[...] there was a moment in time where I was looking at the gun, while I was [...] busy pointing and shooting, so I had a sense of where the bullet came from, [...] in the direction that it goes whenever I direct my arm. [...] So I just used those physical cues to direct my next shots..."*

This indicates that they used the feedback of the previous shot to adjust their next shot continuously throughout the experiment.

8.8 Potential use cases

Overall, there is a potential for the system to be used for exercise games in a similar manner to games like *Beat Saber* and *Pistol Whip* with a leisurely approach, where the user would also happen to exercise as a secondary goal to the goal of entertainment.
Another potential the system has is in the realm of serious games. This could be used with either implementations of the system, based on the goal with an interaction akin to the first iteration focusing on a specific restricted movement (using a pivot point along with the hand e.g. the shoulder), or the second iteration which affords more free range of motion.
With restricted movement you can force the participant to thrust or stretch their limbs in specific manners and adjust the thresholds which could potentially help with rehabilitation, where the system also affords the thresholds for thrusting being dynamically increased as the user finds certain tasks easier with time.

9 Future research

From these results we can thus outline the following future research directions, which could be worth investigating.

9.1 Target Assistance

Though participants were able to hit the targets, their overall accuracy was still quite low. One thing to look into could be the notion of target assistance e.g. target locking, investigating how much aid is required or preferred, and whether target expansion was the correct approach.

9.2 80/20

The experiment consisted of 3 conditions, however, these conditions were not necessarily a ratio with highest possible results. One could investigate different ratios, e.g. 80/20 split as mentioned in Section 8, in relation to accuracy, exertion or usability.

9.3 Locomotion and human behavior

From the experiment there was a tendency of an upwards movement, when participants intended to shoot straight ahead. Thus indicating that humans do not necessarily thrust straight, however, this can vary depending on the person. One could instigate these interactions, with a heavy emphasis on human behavior, to understand the differences in the human interaction.

9.4 Action scheme

For the purposes of our study we used an action scheme which did not include activation.However, incorporating thrusting in a pointing core gameplay mechanic does not have to exclude activation. Many schemes could be explored, such as: **point-thrust-activation** or **activation-point-thrust**. These other schemes could change the behavior of how a user would approach the pointing task in terms of locomotion, which could change the preference and performance.

9.5 Restrictive vs. free movement

Further, one could focus their investigation on the restrictive movement, thereby enforcing a particular motion, which could be of use e.g. in rehabilitation. Though this study merely touched upon this notion, it could be interesting to further instigate.

10 Conclusion

The focus of this paper was based on Systemic-Structural Theory of Activity applied within core gameplay tasks of video games. Here we presented an expansion to the target-to-target task properties with Quantity of activations. Further, based on a state of the art review, we introduced an initiation step categorization for interactions.
Further, we investigated including a forward-thrusting motion into a pointing task, which facilitates exertion through the VR medium. We carried out an experiment, comparing three variations of calculating a pointing trajectory, each using the motion and wrist vector to determine the trajectory, but using different ratios. The results show that participants prefer the combination of both the wrist and motion at equal amounts. Though, it was most preferred, there are still limitations to this interaction, such as overall lower exertion results, unresponsiveness, or too much noise in the pointing. Wrist direction on the other hand, was the most accurate, however, due to the implementation, it is hard to tell whether it would have gotten the same result if there was not an angular restriction.

Our findings suggest that there is a potential for the thrusting motion to be used within serious games, however, variables and calculations will have to be further tweaked in order to a have fully robust system that consistently achieves a desired goal.

References

[1] Anonymous Author(s). in press. Pointing, Steering, Timing: Origin and Utility of Core Tasks in Gameplay. (in press).
[2] Ravin Balakrishnan and I Scott MacKenzie. 1997. Performance differences in the fingers, wrist, and forearm in computer input control. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. 303–310.
[3] Nikola Banovic, Tovi Grossman, and George Fitzmaurice. 2013. The Effect of Time-Based Cost of Error in Target-Directed Pointing Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 1373–1382. https://doi.org/10.1145/2470654.2466181
[4] Gregory Z Bedny and Steven Robert Harris. 2005. The systemic-structural theory of activity: Applications to the study of human work. *Mind, culture, and Activity* 12, 2 (2005), 128–147.
[5] CryEarth. 2013. Whack-a-Mole. [Microsoft Store].
[6] Michael S Debus. 2019. *Unifying game ontology: a faceted classification of game elements*. IT-Universitetet i København.
[7] Ansgar E. Depping, Regan L. Mandryk, Chengzhao Li, Carl Gutwin, and Rodrigo Vicencio-Moreira. 2016. How Disclosing Skill Assistance Affects Play Experience in a Multiplayer First-Person Shooter Game. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3462–3472. https://doi.org/10.1145/2858036.2858156
[8] Beat Games. 2018. Beat Saber. https://store.steampowered.com/app/620980/Beat_Saber/.
[9] Epic Games. 2022. Unreal Engine 5.1 Official Webpage. https://docs.unrealengine.com/5.1/en-US/unreal-engine-5.1-release-notes/
[10] Cloud head games. 2019. Pistol Whip. https://store.steampowered.com/app/1079800/Pistol_Whip/.
[11] For Honor. 2017. Ubisoft Montreal. https://store.steampowered.com/app/304390/FOR_HONOR/.
[12] Martin Kocur, Martin Johannes Dechant, Michael Lankes, Christian Wolff, and Regan Mandryk. 2020. Eye Caramba: Gaze-Based Assistance for Virtual Reality Aiming and Throwing Tasks in Games. In *ACM Symposium on Eye Tracking Research and Applications* (Stuttgart, Germany) *(ETRA '20 Short Papers)*. Association for Computing Machinery, New York, NY, USA, Article 63, 6 pages. https://doi.org/10.1145/3379156.3391841
[13] Fedwa Laamarti, Mohamad Eid, and Abdulmotaleb El Saddik. 2014. An Overview of Serious Games. 2014, Article 11 (jan 2014). https://doi.org/10.1155/2014/358152
[14] Norman E Lane and Robert S Kennedy. 1988. *A New Method for Quantifying Simulator Sickness: Development and Application of the Simulator Sickness Questionnaire (SSQ); Technical Report.* Essex Corporation.
[15] Sven Mayer, Valentin Schwind, Robin Schweigert, and Niels Henze. 2018. The Effect of Offset Correction and Cursor on Mid-Air Pointing in Real and Virtual Environments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3174227
[16] Victoria McArthur, Steven J. Castellucci, and I. Scott MacKenzie. 2009. An Empirical Comparison of "Wiimote" Gun Attachments for Pointing Tasks. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Pittsburgh, PA, USA) *(EICS '09)*. Association for Computing Machinery, New York, NY, USA, 203–208. https://doi.org/10.1145/1570433.1570471
[17] Dean peppy Herbert. 2007. Osu. https://osu.ppy.sh/home.
[18] R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/
[19] Jawad Jandali Refai, Scott Bateman, and Michael W. Fleming. 2020. External Assistance Techniques That Target Core Game Tasks for Balancing Game Difficulty. *Frontiers in Computer Science* 2 (2020). https://doi.org/10.3389/fcomp.2020.00017
[20] Valentin Schwind, Sven Mayer, Alexandre Comeau-Vermeersch, Robin Schweigert, and Niels Henze. 2018. Up to the Finger Tip: The Effect of Avatars on Mid-Air Pointing Accuracy in Virtual Reality. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play* (Melbourne, VIC, Australia) *(CHI PLAY '18)*. Association for Computing Machinery, New York, NY, USA, 477–488. https://doi.org/10.1145/3242671.3242675
[21] Torn Banner Studies. 2012. Chivalry Medieval Warfare. https://store.steampowered.com/app/219640/Chivalry_Medieval_Warfare/.
[22] Shumin Zhai, Paul Milgram, and William Buxton. 1996. The influence of muscle groups on performance of multiple degree-of-freedom input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 308–315.
[23] Xinyong Zhang, Xiangshi Ren, and Hongbin Zha. 2008. Improving Eye Cursor's Stability for Eye Pointing Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 525–534. https://doi.org/10.1145/1357054.1357139