Thesis no: MECS-2014-01

Passive gaze-contingent techniques relation to system latency

By Robin Thunström

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Engineering: Game and Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author:
Robin Thunström
E-mail: roth09@student.bth.se, robin@thunstroem.com

External advisor:
Fredrik Lindh
Software Engineer at Tobii Technology
E-mail: Fredrik.Lindh@tobii.com

University advisor:
Dr Veronica Sundstedt
Department of Creative Technologies
E-mail: veronica.sundstedt@bth.se

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

ABSTRACT

Context. Interactive 3D computer graphics requires a lot of computational resources to render a high quality frame. Typically the process of rendering a frame assumes a naïve approach that the whole frame can be perceived by the user in uniform detail. This is often not true, within 2° horizontal eccentricity from point of gaze is where one can primarily perceive details. Adjusting the quality of a frame based on the visual acuity can increase rendering performance by a factor of five to six at the resolution 1920x1080 without sacrificing perceived quality (Guenter et al., 2012a). Doing so without the user being aware of the manipulation requires a highly sophisticated system with low system latency able to update the display fast enough.

Objectives. The current study aims to answer what system latency is required to support passive gaze-contingent techniques that requires close to real-time gaze data.

Methods. A unique experiment design was developed exposing test subjects to different system latencies by varying eye-tracker and monitor frequency.

Results. The outcome from the current study with 20 participants indicates a configuration with the estimated worst case system latency of 60ms is capable of hiding manipulation for 55% of the participants. Lowering the worst case system latency to 42ms and 95% of the participants reported that they could not detect any change.

Conclusions. The study concludes that the configuration with estimated worst case system latency of 42ms is able to support passive gaze-contingent techniques.

Keywords: Passive gaze-contingent, system latency, eye tracking

Table of content

PREFACE

This thesis was written by Robin Thunström, a master student at Blekinge Institute of Technology in collaboration with Tobii Technology. The majority of the thesis was written at the Tobii's head office in Stockholm Sweden during a period of 20 weeks full-time work started in January and ended in June 2014.

Tobii suggested researching if foveated rendering could work on low cost eye-trackers. This suggestion served as the starting platform for the thesis.

1 INTRODUCTION

The introduction chapter starts with describing the context, the problem and why it is of interest to solve the problem in Section 1.1. Followed by the definition of the aim, the objectives and the research question in Section 1.2. The key contributions are stated Section 1.3. The study's delimitations are stated in Section 1.4 and the chapter concludes with a description of the remaining chapters in Section 1.5.

1.1 Context and problem description

An *interactive real-time computer graphics*[1] (CG) application starts to be interactive at six images per second (6Hz) and truly becomes an interactive experience at 15Hz and above (Akenine-Möller, Haines, & Hoffman, 2011). A rendered image is referred to as a *frame*. The task of creating a *high quality* frame at interactive speed is a formidable task (Andersson, 2012). Quality is the key factor determining the rate at which the frames are rendered. Achieving higher quality to create a more immersive experience demands more resources being spent per frame i.e. creating a photo realistic frame requires more resources than rendering only primitive shapes. However, increasing the frame quality can go unnoticed by a user, as humans can primarily distinguish details in our *foveal vision*. Foveal vision only covers a 2° *horizontal eccentricity* from the point of gaze (Duchowski, 2007; Holmqvist et al., 2011). Consequentially, a large number of the finite computational resources per frame are spent on details that can not be distinguished, as human vision does not distinguish details uniformly.

Utilizing this significant limitation in visual acuity, Guenter et al. (2012a) showed that a reduction based on visual acuity was possible while still maintaining the perceived frame quality. The reduction of resources resulted in a speed-up factor of five to six using common CG technology at the resolution 1920x1080, and the estimated speed-up increases further as the field of view increases. The current study aims to disclose the lower bound for supporting *passive gaze-contingent*[2] techniques, such as *foveated rendering*[3]. If the lower bound for supporting passive gaze-contingent techniques were to be found, hardware could be optimized to reduce production costs. Ultimately, this could make the technology more accessible. The lower bound is defined by the required system latency and the system latency refers to the time it takes from saccade until a reaction of that saccade action is displayed.

If CG applications were to have gaze data from the user they could focus the computational power available where the details make the most difference. As demonstrated by Guenter et al. (2012a); their *foveated rendering* implementation gave a factor of five to six in rendering speedup, with the user rating the quality as comparable to non-foveated CG. This was achieved by primarily focusing on reducing the number of pixels in the periphery while maintaining a maximum resolution at the point of gaze. In addition to reducing the number of pixels in the periphery several other rendering topics in current CG technology could benefit from knowledge about the users point of gaze.

- Adjustments to *level of detail* (LOD), objects further away from the point of gaze objects could be rendered using fewer details. This can work similar to what Lopez, Molla, & Sundstedt (2010) did, replacing the assumption of that

[1] Interactive real-time computer graphics refers to the process of creating a computer generated frame based of 3D models.
[2] Passive gaze-contingent refers to the process of changing displayed stimuli based of a user's point of gaze. The term will be further explained in Section 2.2.
[3] Foveated rendering will be further explained in Section 2.3.

the user is paying attention elsewhere by providing the application with the user's actual point of gaze. Fewer details would decrease memory footprint and computation needed.

- The quality of post processing effects are often based on a fixed number of samples from neighboring pixels to determine the resulting pixel color, e.g. *motion blur* and *depth-of-field* (Sousa, Kasyan, & Schulz, 2012). Varying the number of samples from neighboring pixels in relation to distance of a user's point of gaze would efficiently reduce computation.

- Commonly, tessellation is adapted based on distance from the CG camera and the object. Basing the tessellation factor on the point of gaze would make the tessellation process more efficient, as distance from the camera to the object does not necessarily relate to a user's focus. This technique was adapted by Guenter et al. (2012a).

- Effects that only serve to give a visual hint can have a reduced quality if not focused on by the user. E.g. particles' purpose in CG is often to give a visual hint and if the user can not distinguish the particles then they can be removed or greatly reduced in quality and quantity.

Passive gaze-contingent techniques often focus on increasing the efficiency on already existing technology. When the computation goes down more computation power can be spent elsewhere or not used at all, where the later can reduce power consumption. On mobile devices increased efficiency would enable longer lasting batteries and reducing heat exhaustion. Current trends in hardware sales indicate a growing number of users that move towards mobility rather than stationary computers (Andersson, 2012). Technologies focusing on reducing computational consumption would open up for low performance systems to perform intensive graphics previously not possible.

Ultimately, the future of CG lies in ray-tracing (Keller et al., 2013). Real-time ray-tracing on current CG technology that can compete with rasterization is currently not feasible due to memory and computation constraints of current GPU's (Andersson, 2012). By using knowledge about the user's point of gaze when deciding amount of computation per pixel, ray-tracing techniques could become more resource effective.

1.2 Aim, objectives and research question

The study aims to determine what threshold system latency there is to support passive gaze-contingent techniques through varying hardware frequencies (varying frequencies will increase respectively decrease system latency).

Objectives

1. Investigate how the system latency can be detected when varying the eye-tracker frequency and the monitor frequency.

2. Select, or develop, a method capable of determining if the system latency is fast enough for the user to not notice any modification.

3. Execute selected method and analyze the results to disclose possible system latency requirements to support passive gaze-contingent techniques.

Research question
What threshold values, in milliseconds, are there for system latency to support passive gaze-contingent techniques?

1.3 Key contributions

- A study that for first time vary frequency for both eye-tracker and monitor to increase, respectively decrease system latency.

- The study includes an experiment design in which participants are asked to detect stimuli modification as system latency is varied.

- Results indicate that a configuration with estimated system latency of worst case of 60ms is capable of hiding the change for 55% of the participants. Lowering the worst case system latency to 42ms resulted in that 95% of the participants could not detect any change.

1.4 Research delimitation

The task to determine what system latency is required to support passive gaze-contingent techniques is limited to two *independent* and one *dependent* variable. The independent variables are the eye-tracker frequency and the monitor frequency which affects the dependent variable system latency. This does not take into account the eye-tracker accuracy and precision. By only changing two variables it is assumed that the time it takes for each individual component to do its task is not fluctuating but rather remains stable. This assumption may be unrealistic due to the dynamic nature of the asynchronous components cooperating, i.e. the time it takes to render a frame will most likely change from frame to frame. Testing more variables was deemed to be too complex.

1.5 Structure

The study explains the related work and explanation of key concepts in Chapter 2. The scientific method used is described in Chapter 3. Include in the method section is description of two prototypes and one experiment, how they relate can be seen in Figure 1. Results from the two prototypes and the experiment are presented in Chapter 4. The discussions of the results are in Chapter 5. Conclusions and suggested future work are made in Chapter 6.

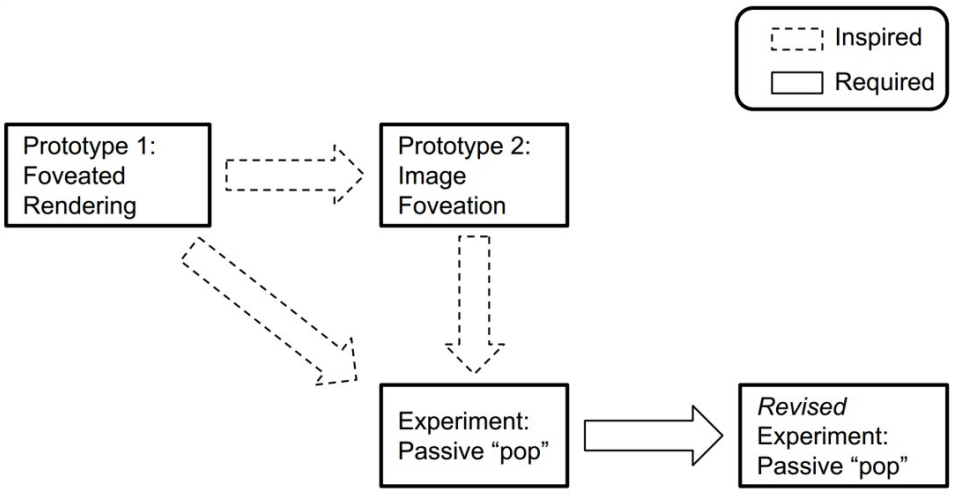

Figure 1: Presents how the prototypes relate to experiment performed in the current study.

2 BACKGROUND

The background chapter provides an explanation of key concepts related to the problem described in Section 1.2. First an introduction to eye-tracking is described in Section 2.1, followed by an explanation of what defines passive gaze-contingent techniques in Section 2.2. Foveated rendering is a passive gaze-contingent technique and is explained in Section 2.3. Foveated rendering is highly dependent on system latency to be able to maintain its illusion and system latency is described in Section 2.4. The background chapter concludes with describing additional related work in Section 2.5.

2.1 Eye tracking

Eye tracking is a method that tracks the movement of the eyes. There are multiple techniques for tracking the eye i.e. optical based, physical connected and electric potential measurements. The most popular technique of eye tracking eyes today are optical eye tracking (Holmqvist et al., 2011). Optical eye-trackers do not typically require the participant to be attached to the eye-tracker; instead they sample a picture of the eye. Included in the picture is a reflection created by the eye-tracker. This reflection is typically made by an *infrared light* (IR-light) to aid in determining the location of gaze. Infrared light's wavelength lies below the visual lights wavelengths so the light is not perceived by the user.

A subcategory to the optical eye-trackers is remote eye-trackers. They allow the user to move freely without any apparatus to restrict head-movement. Remote eye-trackers are only able to track the eyes within the cameras frustum. To get a more in-depth description of the in workings of eye tracking see Duchowski (2007) and Holmqvist et al. (2011).

2.2 Passive gaze-contingent

Once an eye-tracker is connected to a system an application can be fed with information about the user's point of gaze and react accordingly. *Gaze-contingent* is the general term for techniques allowing an application to change stimuli dependent of the user's point of gaze. There are two terms of gaze-contingent, *interactive* (sometimes referred to as *active*) and *passive* (Holmqvist et al., 2011). Interactive gaze-contingent techniques allow the user to actively control an interface with the gaze, i.e. scrolling a page up/down when activated by the user. In contrast to active does passive gaze-contingent techniques not require any active control from the user; instead the stimuli changes, based on the point of gaze, continuously. Passive gaze-contingent techniques are usually the most demanding field in eye tracking with aspect to the eye-tracker frequency as the techniques requires a low response time (Andersson, Nyström, & Holmqvist, 2010). Example of passive gaze-contingent techniques can for example be masking out all words in a sentence except the word at point of gaze or in CG, to simulate depth-of-field (Mantiuk, Bazyluk, & Tomaszewska, 2011).

2.3 Foveated rendering

Foveated rendering, also known as *model based foveation*, aims to reduce CG quality in the periphery based of the visual acuity (Duchowski, 2007). Human vision can primarily distinguish details inside the foveal vision, allowing reduction in quality to go unnoticed. Commonly CG applications assume the naïve assumption that the user can perceive the whole screen in uniform detail. Rendering a frame requires a lot of resources from the *graphical processing unit* (GPU) as it handles the majority of the

processes required to render a frame. This makes the GPU the primary bottleneck for applications dependent on CG and interactivity (Conger, 2010).

To address the GPU performance bottleneck in CG applications a common practice is to cull objects outside the camera's frustum or decrease the level-of-detail, hence reducing the computation loaded onto the GPU (Gregory, 2009). Extending this method to remove quality based on visual acuity can also result in a significant performance boost. The idea is however not new, Levoy & Whitaker (1990) demonstrated how they could focus graphical processing power on the point of gaze to create a rendering application capable of achieving a speedup of a factor 4.6. Their problem was that rendering time was 13.0 seconds per frame, far from being interactive CG. The method described by Guenter et al. (2012a) focused on minimizing the number of pixels processed. Their method reached a performance boost of a factor five to six executed in a resolution of 1920x1080. This was done using common CG technology and with users rating the foveated CG comparable to non-foveated CG.

Foveated rendering shows great promises to solve a lot of current and future computational problems experienced in interactive CG applications. As explained by Johan Andersson[4] during AMD Developer Summit 2013 when asked how a 4K display (3840x2160) can be supplied with frames at 60Hz when rendering CG. However, foveated rendering techniques require a highly sophisticated eye-tracker with a high precision, stable track-ability and low system latency. Even with that, the illusion is easily broken. As described in the study by Guenter et al. (2012b) the participant could blink to disrupt the illusion. Foveated rendering does also require highly sophisticated filters to remove *anti-aliasing* (AA) associated with rendering lower resolution in the periphery. In the article by Guenter et al. (2012a) they used three different types of filters to coop with the lower resolution. Filters used were *multisample anti-aliasing*, *temporal reprojection* and *whole frame jitter sampling*.

2.4 System latency and "pop"

Gaze-contingent techniques rely, more or less, on the system latency. This is especially true for foveated rendering, as participants should preferably not notice the change. When a system is not fast enough to keep up with saccades, the user can then consciously notice the system lag. The lag effect is referred to as "pop". "Pop" can be defined as a brief period of time after a saccade until the system have produced and displayed a corresponding reaction. The term "pop" was coined by Guenter et al. (2012a) and was experienced when they used a Tobii X50 eye-tracker sampling at 50Hz with a latency of 35ms together with a LCD monitor at 60Hz. They later changed eye-tracker to a Tobii TX300, sampling at 300Hz and a stated <10ms latency with a LG W2363D 120Hz LCD. Once they updated their setup the "pop" disappeared, suggesting there is a threshold to when the immersion breaks for foveated rendering. The update of monitor, and not only eye-tracker is important to remark as the monitor and eye-tracker both affect a system's latency. The latency introduced by a monitor is especially important to remark if the technique called *vertical synchronization* (V-sync) is enabled. V-sync synchronizes the rate at which frames are rendered with the refresh rate of the monitor. However, this introduces additional latencies (Carmack, 2013; Wilson, 2009). Disabling v-sync can introduce a visual artefact known as tearing. Tearing occurs when two different frames are visible at the same time. See Figure 2 to get an overview of the typical latencies in an eye-tracking environment.

[4] http://www.dailymotion.com/video/x1ehtw4_johan-andersson-foveated-rendering_tech, retrieved 2014-06-10

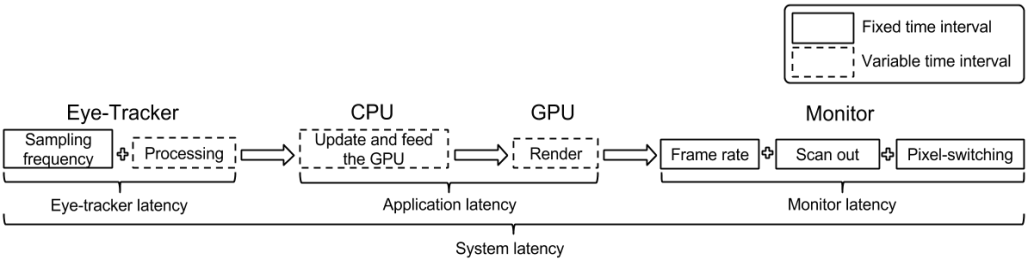

Figure 2: Shows a simplified overview picture of latencies related to gaze-contingent techniques. Individual components may look different depending on the given setup, e.g., a monitor can have a dedicated post processing step for each image it receives adding additional latencies. To get more information on latencies see Wilson (2009) and Carmack (2013).

Previous gaze-contingent applications have, unintentional or intentional, been using fast eye-trackers and fast monitors to decrease system latency. A shorter list of example equipment used is listed below:

- Eye-tracker EyeLink 1000 sampling in 1000Hz together with a 75Hz LCD monitor (Rayner, Castelhano, & Yang, 2009). The monitor is by the manufacture referred to as a fast LCD monitor.

- Eye-tracker EyeLink II 500Hz binocular eye-tracker together with a 150Hz CRT monitor (Häikiö, Bertram, Hyönä, & Niemi, 2008).

- EyeLink 1000 sampling at 2000Hz together with a Iiyama HM204DT 100Hz CRT monitor (Mauderer, Conte, Nacenta, & Vishwanath, 2014).

2.5 Additional related work

In the study by Loschky & Wolverton (2007), they concluded that updating stimuli as late as 60ms after the eye had finished the saccade did not affect the detectability of image blur and/or motion transients. The foveation technique used was *image based foveation*. Image based foveation works similar to foveated rendering described in Section 2.3, except that precomputed images are used. To conclude what is fast enough they used an experiment design that varied the artificial system latency of 20-, 40-, 60- and 80-ms and let users detect blur and/or motion transients. In an earlier article it is suggested that stimuli updates should be completed within 45ms to not increase fixation durations (Loschky & McConkie, 2000). The artificial system latencies tested was of 5-, 15- and 45ms.

3 METHOD

The method chapter describes the processes of selecting a scientific method to answer the research question described in Section 1.2. The Section 3.1 describes the process selecting the scientific method. The Section 3.2 describes the setup used. As described in Section 2.4 is the system latency important to know in order to answer the research question, therefore is the current setup's latency stated in Section 3.3. The different applications developed used the same equation to calculate visual acuity and the equation is described in Section 3.4. Where following the Sections 3.5, 3.6 and 3.7 describe the two prototypes and the experiment developed.

3.1 Selection of method

To provide an answer to the research question in Section 1.2 different research methods are possible. One could for example create a survey collecting quantitative data and ask participants what they would perceive as fast enough to support passive gaze-contingent techniques. A survey method in this context would require participants to be well informed of the problem and the related terminology before answering the survey. Including a detailed description of the problem could inform participants if they are not familiar with the topic, but the survey would run the risk of participants skipping the information and replying without full knowledge about the problem. Instead of creating a survey one can interview experts in the area and collect qualitative data. Collecting qualitative data from interviewing experts can threaten to reduce the results general applicability (Ekström & Larsson, 2010).

Alternatively, an experiment has the benefit that participants can answer questions related to the experiment they are exposed to and results can be general applicable, given enough participants. However, a problem with experiments is the need of isolating the dependent and independent variables so that the results are not affected by non-controlled variables. Ideally, in the current study's context; the experiment would guarantee only conscious detections of "pop". If the system is fast enough the users would not detect any modification. Selection of experiment as the research method is strengthened by, to the author's knowledge, previous research in the field of passive gaze-contingent has exposed test subjects to different types of stimuli.

The process of establishing an experiment design that could answer the research question in Section 1.2 resulted in two prototypes and one experiment, as shown in Figure 1. They will be explained in chronological order and their results will be presented in Section 4. The two prototypes and the experiment used the same setup, they were affected by the same system latency and they used the same calculations to determine the visual acuity.

3.2 Setup

A perfectly controlled experiment environment will not be affected by immediate systems. Unfortunately, the experiment environment for an eye-tracking environment is a complex environment consisting of individual asynchronous devices. The Experiment environment can affect the results and therefore the validity and the reliability of the results (Andersson et al., 2010). The selected individual devices were chosen to minimize their unintentional effect on the results.

The computer used in the experiment hosted an Intel Core i5-4670K (at 4.2GHz) CPU, 16GB DDR3 ram, a Geforce GTX 660 GPU and as storage was a Kingston 120GB SSDNow V300. The monitor used was an Asus VG248QE LCD monitor capable of updating at 144Hz in resolution 1920x1080. Typically a display has internal buffering

of images and image processing. E.g. TV-monitors often have a “gaming mode” that removes as much image buffering and image processing possible, in favor for lower latency traded against lower image quality (Morrison, 2012). The Asus monitor was selected because of the manufacturer’s claims of having minimal latencies related to image buffering, pixel switching and image processing[5]. The supported frequencies for the monitor are 60-, 85-, 100-, 120- and 144Hz.

The eye-tracker used was the Tobii TX300, the same model as used by Guenter et al. (2012). The eye-tracker is capable of sampling up to 300Hz. In addition to the 300Hz did the eye-tracker support sampling at 250Hz, 120Hz and 60Hz. To support additional system latencies a custom firmware was applied that made it possible to add additional frequencies. The sampling frequencies added were 30-, 45-, 85-, 100- and 144-Hz. The setup used can be seen in Figure 3.

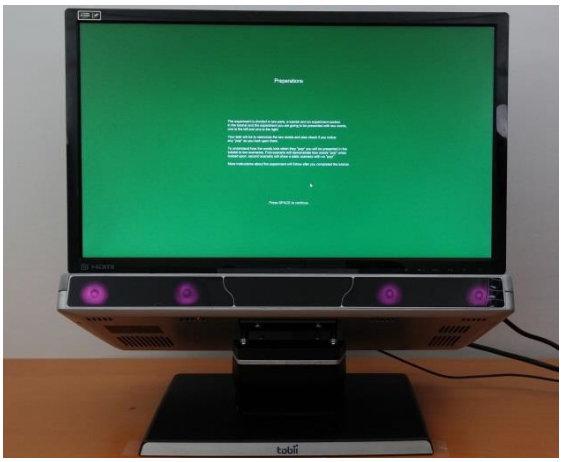

Figure 3: Shows the Asus VG248QE (in black) attached on top of the Tobii TX300 (in grey). The monitor was attached to the eye-tracker through a static stand to keep it fixated. The four purple lights visible in the figure are the IR-lights used to create the reflection in the eye.

The framework used to create the two prototypes and the experiment was *Unity Pro*[6] version 4.3. The author had previous experience developing rapid prototypes using Unity and as the final experiment was unclear prototyping in Unity was deem viable. The gaze data was provided through a custom version of *Tobii EyeX for Windows*. It is a bundle of software’s with the purpose of integrating Tobii eye-trackers with Windows. The custom version was required as the bundle originally could not communicate with the TX300 used in the current study. Streaming gaze data to Unity was made through the Tobii Unity *software development kit* (SDK) version 0.1746. The Tobii Unity SDK was also altered to include information about both eye’s distance from the screen, as it was not originally supplied by the SDK.

3.3 System latency

As explained in Section 2.4, are passive gaze-contingent techniques sensitive to system latency. When asynchronous devices cooperate latency will vary, creating a worst- and best-case scenario. Investigations of system latency follows the principles described by Guenter et al. (2012a).

[5] http://www.asus.com/Monitors_Projectors/VG248QE/, retrieved 2014-06-10
[6] http://unity3d.com/, retrieved 2014-06-10

TX300's manual states a latency of <10ms. This includes mid-point exposure of the eye, to when a sample is available via the API, including all communication latency. In the manual it is noted that there is 1.0-3.3ms processing time for each image after exposure. Adding the 3.3ms exposure time with the processing time gives a best case eye-tracker latency of 4.3ms in and in worst case 6.6ms, excluding the communication latency. Calculations by Guenter et al. (2012a) concluded a worst case eye-tracking latency of 10ms and best case of 7ms, excluding communication latency. The study assumes the latency introduced by the eye-tracker to be *processing latency* (*eP*) and *sampling frequency* (*eF*) as stated in the manual.

The *scan out time* (*mS*) for the monitor is in relation with the monitor's current frequency. The best case scenario is assumed to be 2ms, as stated in Guenter et al. (2012a), and worst case scan out time to be the monitors current update frequency, i.e. the scan out latency for a monitor at 60Hz will be 16ms in worst case.

Common practice of measuring the display's pixel switching latency involves filming the display with a high speed camera (Carmack, 2013; Guenter et al., 2012a; Wilson, 2009). In the current study the camera iDS UI-3370CP sampling at 1000Hz in monochrome was used. Recording in 1000Hz allowed 1ms measurements accuracy. *Pixel-switching timing* (*mP*) for the monitor was determined through the recorded video to be constant 5ms.

The measurements of the total system latency were performed using a mirror and a camera, with the mirror covering half of the cameras field of view. The method to use a mirror to capture both the screen and the eye in the same video was described in Holmqvist et al., (2011). A figure of the physical configuration including the camera, mirror, display and a sample from the recorded video can be seen in Figure 4. The measured system latency can from the recording can be seen in Figure 5.

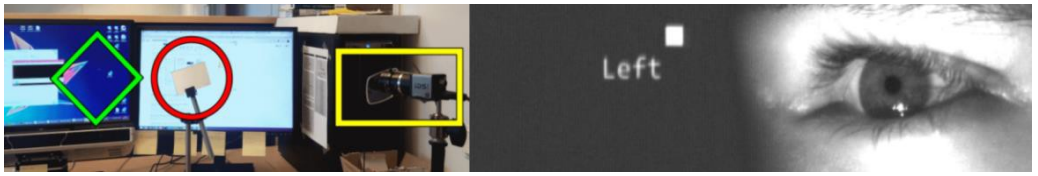

Figure 4: The image to the left presents the physical setup used when recording system latency. The camera (marked with a rectangle) using an IR-filter is aimed towards the monitor (region filmed is marked with a rotated square) and in the middle (marked with a circle) is the mirror. Image on the right the user's right eye in the same video. White square just above the "Left" is the application's prediction of the point of gaze. The bright reflection in the lower part of the eye is caused by a LED-lamp used to increase brightness.

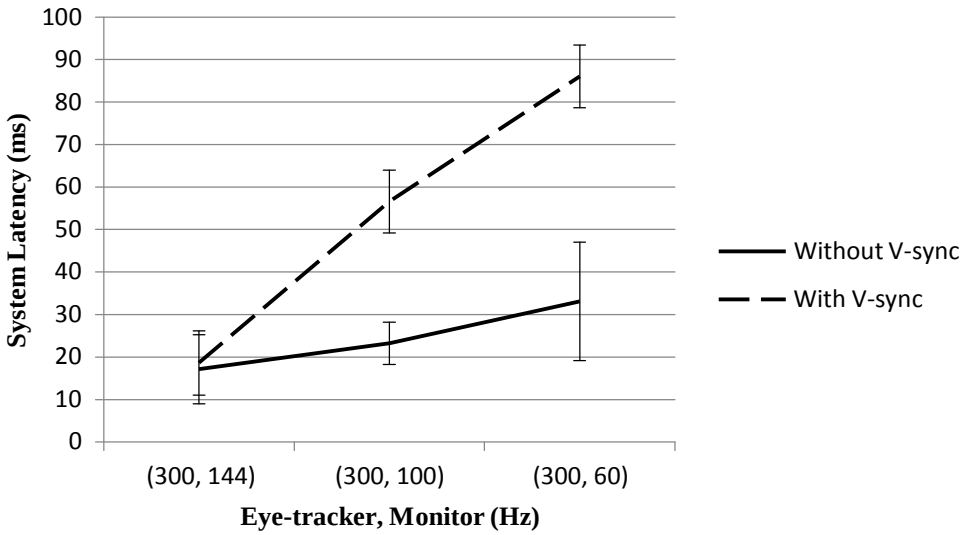

Figure 5: Shows the measured latencies for three distinct configurations. The error bars are calculated using 2 standard deviations. The system latency does not include the 5ms extra for the pixels to switch.

Having disclosed the latency for individual the eye-tracker and the monitor, it is possible to calculate an estimation of best-, worst and average latencies. Worst- and best-case system latencies were estimated using the same equation used by Guenter et al. (2012a), see Equation 1. Estimated system latencies for the configurations used in both the two prototypes and the experiment can be found in Appendix B.

(1)

3.4 Eccentricity layers

The two prototypes and the experiment used the same equation to calculate the visual acuity. The equation is from the eccentricity calculations provided by Guenter et al. (2012a). The values used to calculate the eccentricity are as follows $m = 0.220°$, $V^* =$ 60cm, $W^* = 56$cm, $D^* = 1920$, $\alpha^* = 9/16$ and an angular display sharpness of $w^* =$ 0.0557°. As suggested in the original article, a brute force method[7] was used to minimize the number of resulting pixels given by Equation 2. See the original article for more information about the equation. The results gave an inner eccentricity layer of 4.1780° and outer eccentricity layer of 9.3470°. Note that the values stated are angular diameter and not radius.

{ (2)

[7] Matlab's function *fmincon* was used to minimize Equation 2, http://www.mathworks.se/help/optim/ug/fmincon.html

3.5 Prototype 1: Foveated rendering

The first prototype was implemented according to the foveated rendering details given in Guenter et al. (2012a). Replicating their method seemed as a promising start as it was concluded that a majority of their subjects rated their foveation as comparable as or better than non-foveated rendering. Premises were that if the system could keep up with saccades and updating stimuli fast enough, subjects would not be able to notice the foveation. Altering frequencies on both eye-tracker and monitor would disclose possible system latencies thresholds for when participants stopped detecting "pop".

The prototype did not include any pilot study or saved any data from a user's session. It only included a small virtual environment where the user controlled a spherical avatar using a keyboard and a mouse from a third person perspective. To see a screenshot from prototype see Figure 6.

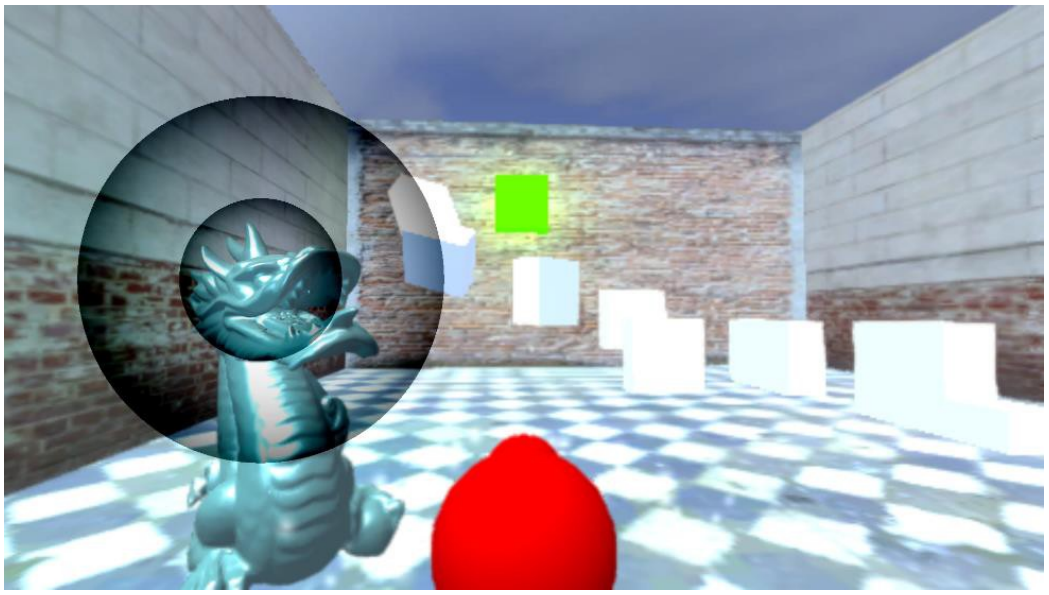

Figure 6: Shows foveated rendering implemented in Unity rendering in 1920x1080. The inner circle (focusing on the head of the dragon) corresponds to the inner eccentricity layer; second circle corresponds to the outer eccentricity layer. The character controlled by the user is the red spherical character and the dragon model to the left is the Stanford Dragon. The implementation used *fast approximation anti-aliasing* (FXAA) and included no temporal reprojection or whole frame jitter sampling reason for this is elaborated in Section 4.1.

3.6 Prototype 2: Image foveation

The second prototype used the same foveation principal as in the first prototype. However, instead of CG graphics it used static images. In contrast to the first prototype did the second prototype not have any AA problems typically associated with CG (Akenine-Möller et al., 2011). The second prototype included four different images. The first image (Tigers) had limited parts of the image in focus, the second image (Refinery) contained vertical and horizontal straight lines, the third image (Wall) contained medium to larger details and the fourth image (Houseboat) contained smaller details. All images are present in Figure 7.

These images were included in eight scenes; each scene showed the same image twice for 7s each, separated with a brief period of 0.5s full screen black. After the period of black the rendering method was changed from either foveated or non-foveated. Chance decided if the first or second image was rendered using foveated rendering or non-foveated. After each imaged had been shown, the user was asked which image had the best quality (1) first image, (2) second image or (3) equal quality. This was repeated

until all four images had been shown. After all four images had been shown the region of foveation was doubled to 8.356° respectively 18.694° and the same four images were shown in the same order again. Premises being to investigate stimuli without any anti-aliasing errors and if doubling foveation size had any major effect on detecting "pop". Configuration of the system was during the pilot study fixed for the eye-tracker to 300Hz and monitor to 144Hz with V-sync enabled.

Each image was precomputed to three different sizes using bilinear interpolation, the three different resolutions (1920x1080, 960x540 and 480x270) correspond to the pixel density for each eccentricity layer calculated in Section 3.4. To create the foveation did the inner eccentricity layer sample from the 1920x1080 version of the image, the outer eccentricity layer sampled from the 960x540 version of the image and the remaining pixels sampled from the 480x270 version of the image. These were then blended together into one frame, in similarity to what was done in the first prototype.

Figure 7: Shows the four different images used in the static foveated image experiment. Each image had precomputed sizes of 1920x1080, 960x540 and 480x270 to represent different level of details. The order of which the images was presented in the experiment was (1) upper left, (2) upper right, (3) bottom left and (4) bottom right.

3.7 The experiment

The experiment was built around altering characters inside a word. The inspiration came from the moving window-paradigm (Häikiö et al., 2008). The moving window-paradigm works much like the foveation in Prototype 1 and 2. Extending out from the point of gaze, is a region defining the window. Outside the window are the stimuli altered and inside the window are the stimuli displayed unaltered. In the current study; the size of the window is the same as the inner eccentricity layer described in Section 3.4. In contrast to Prototype 1 and 2 that interpolates between different layers is no interpolation performed in the experiment. Either the unaltered word is displayed or the altered word is displayed.

Häikiö et al. (2008) showed how adults can perceive 9 characters to the right from point of gaze, measured at 60cm away from the screen with each character is spaced 0.5°. This confirms that text is primarily distinguishable inside the foveal vision[8]. If a

[8] Limitations of reading characters outside the foveal vision can be experienced by having business card at arm lengths. While focusing point of gaze straight forward and slowly moving the business card into point of gaze.

user cannot detect alterations of characters other than inside the foveal vision it should not be possible for the participant to see any "pop". Given that the system changes fast enough and that the form and the contrast from periphery matches what is perceived at the point of gaze. Respectively if the system latency is too high participants can detect the "pop" as the system changes too late. Words used in the experiment can be seen in Table 1. All of the 10 words used, described an animal of five characters. Five characters corresponds to half of the 9 characters, rounded up, to what Häikiö et al. (2008) concluded to be perceivable by adults. The altered versions of the words were subjectively selected to maintain the same form in the periphery as the unaltered word.

Unaltered word	Normal word
Shark	Skenk
Whale	Wkeie
Sheep	Skccp
Horse	Huece
Snake	Saehe
Skunk	Shnuk
Zebra	Zadna
Bison	Blzcn
Panda	Perba
Mouse	Muoce

Table 1: Lists the 10 different words used. The words are using Consolas, a monospaced font. Using the monospaced font Consolas allowed individual characters inside the word to be replaced without changing a word's width. Note that each altered word keeps the first and last character.

The frequencies tested were based on the 60ms update threshold stated in Section 2.5. By including latencies ranging from below to above the 60ms threshold; results are expected to include possible configuration where participants experience "pop" and no-"pop". An individual configuration's system latency was calculated using Equation 1 (from Section 3.3). Configurations that had system latency around 60ms in worst case was the following, monitor at 60-, 85-, 100- and 120-Hz, and eye-tracker at 30-, 45-, 60-, 85-, 100- and 120-Hz. These configurations gave 24 unique system configurations; these were all added as 24 individual experiment scenes.

The procedure of the experiment was to first introduce the term "pop" to the participant of the experiment. As "pop" is not well established term; two tutorial scenes were added to show through example how "pop" and no-"pop" looked. First, two words are shown, one to the left and one to the right. Once the user gazed at the left and the right word it changed from "Pop" to "Left" respectively "Right". Configuration of the system during tutorial was (120, 60)[9] with V-sync enabled, which gave an estimated system latency of around 80ms according to Figure 5. In the following tutorial scene the user were presented with two words that did not change at all, a scene not changing anything will be referred to as a reference scene. To verify that participants correctly understood what "pop" was, 8 reference scenes were added to the list of 24 scenes.

After completing the tutorial, the participant was instructed to prepare for the experiment. The participant received information that they were to be presented with two words and that they should try to detect any "pop". It was also stated that the participant should remain at 60cm away from the screen throughout the experiment.

[9] A configuration of (120, 60) is a shorthand that means the eye-tracker is sampling in 120Hz and monitor is set to 60Hz.

Instructions also stated that each scene was only going to be visible for four seconds and succeeding each experiment scene was a scene stating two questions, (1) two words did you just saw, (2) if you experienced any “pop” (yes or no). Quoted instructions from the experiment can be found in Appendix A.

Once comfortable the participant proceeded to start the first experiment scene. The selection of two words displayed for each experiment scene was randomly selected from the set of 10 words in Table 1. First two experiment scenes were always the same for all participants. One being a reference test and the other being a configuration of (30, 60), the order at which they were presented was randomly selected. These two tests did not record any data and was added to give the participant time to learn the layout for the experiment. For the remaining 32 scenes were the order randomized to remove the ability to anticipate or guess configuration. To motivate the participant to maintain focus on the words an detection of “pop” was a personal score added. If the participant could mark the correct two words from the previous experiment scene they gained one score, maximum score being 34. A screenshot from an experiment scene can be seen in Figure 8.

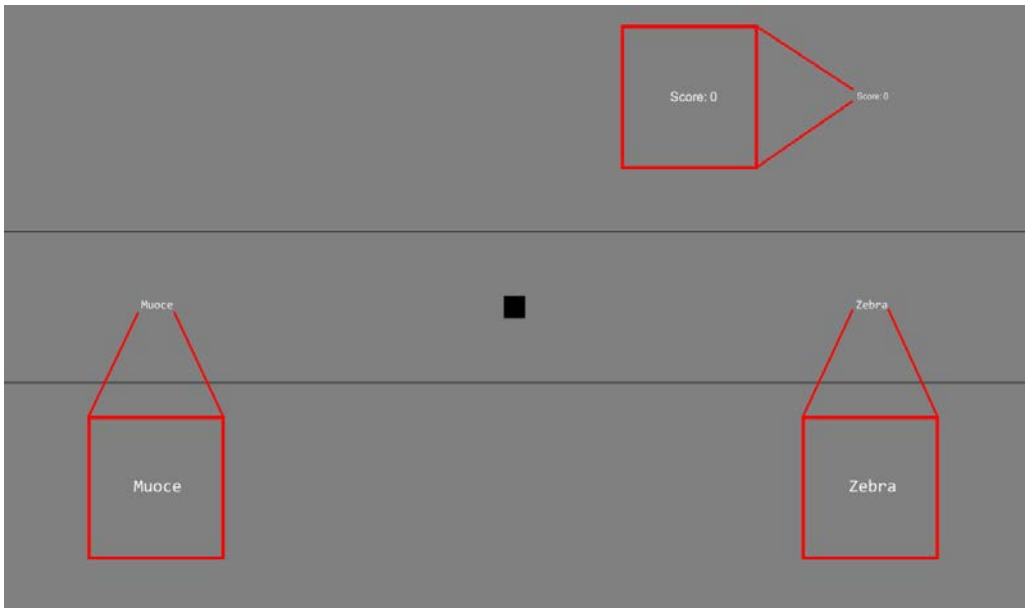

Figure 8: A screenshot from the experiment showing an experiment scene. Where users are presented with two words (marked in with boxes are areas increased in size for demonstrational purpose only). Only when the point of gaze is located within the inner eccentricity layer (4.1780°) does the word show its correct word. The *right* word is showing the correct word “Zebra” (as point of gaze is within the right words region) and *left* word is showing the altered word “Muoce”. The current score was always shown in the upper right corner. The font sized used was 16 pixels; a word spanned roughly half the inner eccentricity size. To orientate the user, two horizontal lines and square at center of the screen was added.

The experiment was conducted in a room without any windows and with lamps that did not emit any IR-light. As the eye-tracker uses IR-lights to create reflections in the eyes; light coming from other sources may disrupt the eye-tracker (Holmqvist et al., 2011). Level of illuminance was measured at two points, at the eyes and where the eye-trackers cameras were located. The measurements were made using a light-meter set to 2000lux. The measured values ranged from 280lux too 360lux.

4 RESULTS

The results chapter presents the results from the two prototypes and the experiment. First, the results from Prototype 1 are presented in Section 4.1 followed by Prototype 2 in Section 4.2. The results from the experiment are presented in Section 4.3.

4.1 Prototype 1: Foveated rendering

After attempts to replicate the foveated rendering method described by Guenter et al. (2012a) was deemed not possible in Unity. This was primarily as Unity's MSAA filter did not work when rendering the cameras viewport to texture. As of today Unity does not currently allow access to the render buffers required to create a custom MSAA filter. It was therefore not possible to create an exact implementation of the foveated rendering technique as described by Guenter et al. (2012a). In an attempt to reduce some of the AA problems, FXAA was used instead. This limitation made the author to dismiss the temporal reprojection and the whole frame jitter sampling as the implementation would not be comparable to the original.

4.2 Prototype 2: Image foveation

As the second prototype only used static imagery there was no need for AA filters. The results from the application was direct comparable with the foveation described by Guenter et al. (2012a). A pilot test was conducted on six subjects. Out of the 48 scenes, only 13 scenes were rated as equal in quality. Of the test that was not rated equal in quality had 100% of participants marked the non-foveated image as highest quality. Of these 13 scenes that were marked as equal in quality, did 8 scenes use doubled foveation size. To see how many times each individual image was marked as equal in quality see Figure 9.

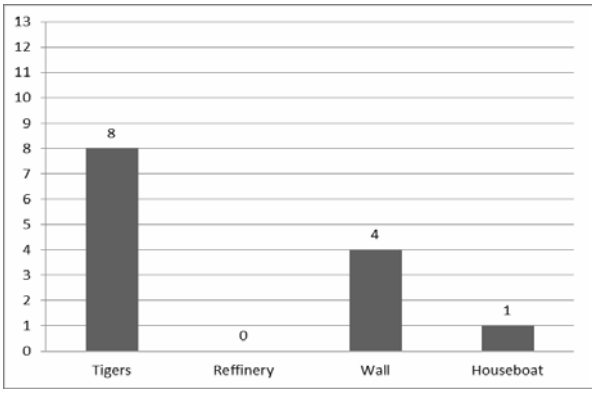

Figure 9: Shows the number of times each image was marked as equal in quality compared to non-foveated image.

4.3 The experiment

The experiment was conducted twice due to implementation issues. After the first run, closer investigations of the recorded data indicated larger execution spikes, especially for the gaze-data. The timestamps spikes were caused by not finishing loading the environment variables before starting the next scene. This effect made it possible for participants to fixate on any of the words and see it "pop" as gaze-data arrived. Additionally, closer investigation of the inner eccentricity region around each word showed that the implementation was incorrect. Instead of a circular region it was in the shape of an ellipse. These two issues made the results from the first run invalid, requiring a re-run of the experiment.

The second run of the experiment included solutions to the issues of timestamp spikes and the region around the word. The presented results come from the experiments second run. The measurements of the timestamps for both the application execution and gaze-timestamps can be found in Appendix B. To ensure the application had successfully finished loading before next scene started a square was included at the center of the screen. The square, once fixated, shifted color from white to black and an option to start was presented to the participant.

A total of 21 participants took part in the second run of the experiment. One of the participants was excluded from the results due to the tracker not being able to track the participant's eyes. Subsequently the final number of participants ended on 20, 16 male and four females. Of these participants the average age was 30.6 with a standard deviation of 5.8. Out of the 20 participants, six had glasses or contact lenses where the remaining had 20/20 or better vision.

Data from the participants' replies to the question when they experienced "pop" can be seen in Figure 10. The exact percentage values from Figure 10 can be seen in Table 2.

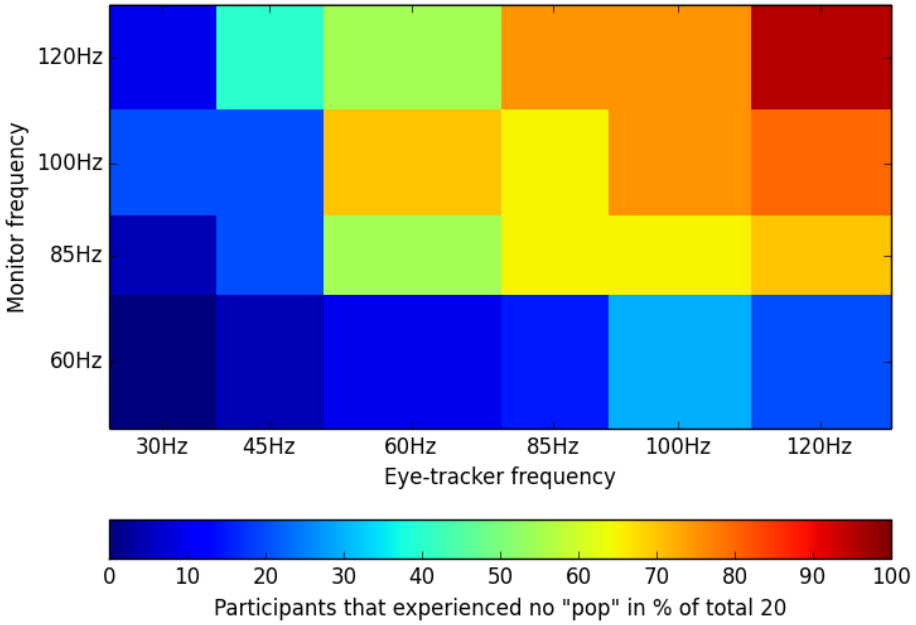

Figure 10: Shows all participants replies from the question if they experienced any "pop" during a given configuration.

The statistical analysis was made using *2 proportion z-test* with $\alpha = 5\%$. Stating the *null hypothesis* ( ) that there is no significant difference between the proportions and the *alternative hypothesis* ( ) that there is a significant difference between the two proportions. If *z* is less than the critical value -1.96 or greater than 1.96 reject the .

The configuration (60, 85) is the first configuration with over 50% participants stating that they cannot see any "pop". Comparing configuration (60, 85) to its lower neighboring configurations, (45, 85) and (60, 85), shows there are a significant difference respectively . While comparing (60, 85) to it higher configurations (85, 85) and (100, 60) there is no significant difference and .

The configuration (120, 120) had the lowest system latency tested and also had the highest rate (95%) of participants experiencing no-“pop”. Comparing this configuration with its neighboring data points (100, 120) and (120, 100) shows and there are no significant differences between the configurations.

Monitor Frequency	30Hz	45Hz	60Hz	85Hz	100Hz	120Hz
120Hz	10 %	40 %	55 %	75 %	75 %	95 %
100Hz	20 %	20 %	70 %	65 %	75 %	80 %
85Hz	5 %	20 %	55 %	65 %	65 %	70 %
60Hz	0 %	5 %	10 %	15 %	30 %	20 %

Eye-Tracker Frequency

Table 2: Shows the percentage of all 20 participants that answered, that they did not experience any “pop” for a given monitor and eye-tracker configuration.

From the data collected during the reference scenes, participants replied to 2 out of 160 reference tests that they experienced “pop”. These two replies were falsely marked “pop” as no words changed during these two scenes.

Out of the 20 participants 11 of them had been participating in the first run of the experiment. They participated again due to limitations in available participants. Their results compared to participants with no previous experience can be seen in Figure 11.

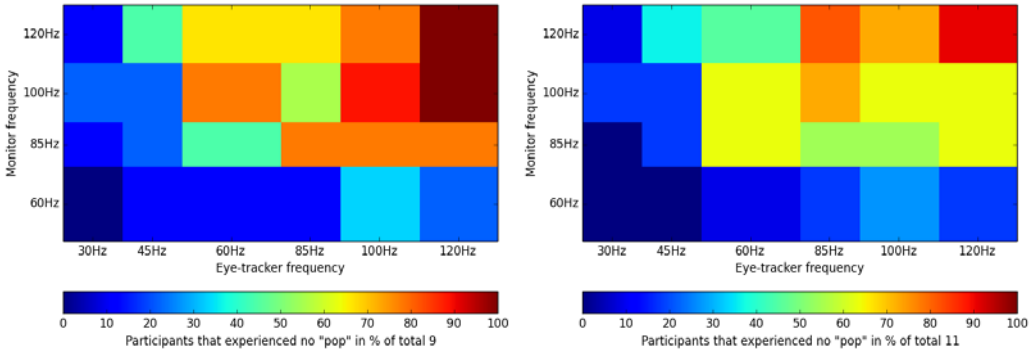

Figure 11: The left image shows the result from the participants that had only run the experiment once. To the right, the results from the participants that had run the test once before are presented.

Results from the two groups of participants using the same 2 proportion z-test as described earlier show that there is no significant difference between the two groups at four key points. These four key points define the corners of the rectangle at which participants replied no-“pop”. They are (60, 85), (60, 120), (120, 85) and (120, 120). Significant test gave , , and .

All participants in addition to replying if they detected a “pop” marked which two words they saw. Maximum score was 34. Out of the 20 participants the average score was 33.45 with a standard deviation of 0.94. The total score from the group of users that had no previous experience of the experiment was 33.56 with a standard deviation of 0.73 and the group with experience had a score of 33.36 with a standard deviation of 1.12.

5 DISCUSSION

The discussion chapter includes a reflection around the results from Section 4 for the two prototypes and the experiment. The chapter follows the same structure from the Method and Results chapters by discussing each individual prototype and experiment in chronological order. The chapter concludes with a discussion of possible threats to the validity of the experiment in Section 5.4.

5.1 Prototype 1: Foveated rendering

Over 50 participants tested Prototype 1 under a period of 2 months. A majority of participants stated the prototype to be working well enough to hide foveation at the point of gaze. But in the periphery vibrating pixels and AA-issues, i.e., jagged edges, made the participants aware of the lowered quality in the periphery. These artifacts caused different responses, from barely noticing them to causing distinct noticeable effects. At the point of gaze the participants exclusively stated that it was not possible to detect the low quality rendering. This was with the system configured to (300, 144) and with V-sync disabled. This indicates that a configuration of (300, 144) has low enough system latency to hide the manipulation at point of gaze.

Hiding the foveation in the periphery did not work as well as expected, due to not being able to implement MSAA. Additionally, in contrast to Guenter et al., (2012a) that used procedural textures, the prototype included textures on the walls and floors to add more visual fidelity. Objects with textures appeared to cause a higher detection of foveation than objects without textures, i.e. the Stanford Dragon model had no textures (see Figure 6). A reason for this could be that rendering with fewer pixels reduces the quality of a texture severally, causing a blurred perception of the texture.

5.2 Prototype 2: Image foveation

In Prototype 2's pilot study, 73% of all scenes shown were marked as a difference in quality. This high percentage was not expected. Rather a number closer to 10% as indicated in the study complemented to Guenter et al. (2012b) where seven out of 76 test subjects stated "periphery looked blurry". That periphery looked blurry was the main expression by the users in the pilot study. Having participants from two independent studies explicitly stating that the periphery looks blurry, indicates that the method of decreasing pixels in the periphery cause visual artifacts that are detectable. Alternating between foveated and non-foveated caused participants to state a distinct loss of quality in the periphery, especially noticeable for the image referred to as "Refinery". This is noticed in the results shown in Figure 9. The images used were selected on a subjective basis and includes more visual fidelity than in Prototype 1. During all the tests was the system configured to (300, 144) with V-sync enabled.

5.3 The experiment

The experiment showed how participants could not detect any alteration at all if the system latency was low enough. The participant's inability to detect any "pop" when explicitly informed to do so, suggest that the results are reliable. In addition with only having two reference scenes out of total 160 being falsely flagged as "pop" indicates that participants knew how "pop" looked like. The experiment results may still include variance due to blinking or different timings for the system. If a participant blinked during any of the four seconds the experiment scene was present it could have triggered "pop", even if the system was fast enough. As with the timings of asynchronous devices their timings in aspect to each other will vary between more or less in sync. This makes it possible for a scene to have good timings for a participant but not another. That may explain why some of the configurations with lower system

latency scored fewer participants stating no-“pop”, see configuration (60,120) against configuration (60, 100) in Figure 10.

In the experiment participants were only asked if any detection of “pop” was made. Not taking in to account that a “pop” may only have been visible once or if the participant believed they saw something. As noticed by the author during the experiment was that a participant could hesitate and doubt if they actually had seen a “pop” or not. This tendency of hesitation was never experienced during any of the reference tests. Suggesting that close to threshold values there may be situations where participants cannot consciously detect the change.

Of the 24 different system configurations tested did none of them reach 100% participants stating no-“pop”. A could be that the experiment design is not working for some people. By not working it is referred to the participant being able to detect the change through periphery vision. A study on human vision of players with a lot of experience with first person shooter games (FPS) concluded that some of these players vision are better than for an normal person (Li, Polat, Makous, & Bavelier, 2009). The participant that detected the “pop” at lowest system latency did after the experiment state to be an experienced FPS player. However the results from that person still showed some configurations where no-“pop” was detected.

After the experiment was finished all participants had the opportunity to ask any question they had and provide feedback. A reoccurring statement was that if the participant experienced any “pop” during a scene it often happened exclusively on the very first saccade, from the square to the left or the right word. A speculation to the reason for this would be that the angular velocity of the eye causes the eye-tracker to miss the saccade. However this behavior should also apply to longer saccades and not only the very first saccade from the center. The angular velocity of the eye is gradually increasing from roughly 100°/seconds to reach a max velocity of 900°/seconds for saccades between 10-20° (Bahill, Clark, & Stark, 1975). Stated by Loschky & Wolverton (2007) in their study was that “pop” was often discovered when longer saccades were made. Or it could be that the alteration of words can be detected when the starting point of the saccade is closer to the word.

In the experiment the participant’s attention was targeted towards memorizing the two words and detecting “pop”. If the participant were to have a different task the detection rate would probably go down as focus would then be on performing another task causing inattentional blindness (Lopez et al., 2010). As first explored by Yarbus (1967) a user’s task affects his or hers search pattern making it possible to anticipate a user’s behavior. E.g. in a first-person-shooter game it was shown that over 80% of all fixations was centered around the crosshair (Kenny, Koesling, & Delaney, 2005). This is an important remark as an application utilizing passive gaze-contingent techniques would probably direct the user attention towards something else other than detecting “pop”.

5.4 Threats to validity

Out of the 20 participants in the experiment were 16 male, four female. The author has not found support in the literature of there being any difference between male and female vision that suggest that a particular sex would experience the experiment differently. To be a participant of the experiment the participant had to be over 18 years old and know about eye-tracking. The requirement of knowledge about eye-tracking was just to ensure that the participant did understand that they could not cover their eyes or similar during the experiment. This was also instructed before the experiment started.

To ensure that participants remained at 60cm away from the screen throughout the experiment was it possible to check current distance from the screen during the question scene.

As the experiment used text as stimuli in contrast to Prototype 1, Prototype 2, Guenter et al. (2012a) and Loschky & Wolverton (2007) all used graphics as stimuli, its validity may be questioned as white text on grey background is not the same as graphical stimuli. In defense it should be noted that contrast and form of the stimuli was still manipulated in the experiment. Given that the stimuli change in contrast and form in periphery to point of gaze is subtle and happens fast enough it can be argued that stimuli is independent from graphical and non-graphical. However what can be considered subtle is not yet well defined.

6 CONCLUSION AND FUTURE WORK

The conclusion and future work chapter contains the conclusion of the two prototypes and the experiment in Section 6.1. The conclusion section investigates if the research question from Section 1.2 has been answered by the current study. The suggested future work is described in Section 6.2.

6.1 Conclusion

Looking at the results from all the participants from the experiment it was shown that the configuration (60, 85) is a significant threshold where over 55% of the participants flagged no-"pop". As the system latency was decreased the number of participants not detecting any "pop" increased. For the lowest measured system latency, 95% of all participants flagged no-"pop". Estimated system latency for the configuration (60, 85) is in the best case 36ms and in the worst case 60ms. Where for the configuration (120, 120) had the best case system latency of 25ms and the worst case system latency of 42ms. These values are estimated system latencies calculated from Equation 1. To see estimated system latencies for more configurations see Appendix B.

To support passive gaze-contingent techniques in 95% of the time a system with a worst case system latency of 42ms is required. While a system with a worst case system latency of 60ms may work for a majority of users. This conclusion answers the research question stated in Section 1.2.

6.2 Future work

Results from the pilot study in Prototype 2 indicates that real-time foveation in CG is not yet solved, as participants could easily detect a foveated image versus a non-foveated. Future work may investigate the creation of filters to maintain form and contrast so it can be executed in real-time to perform a result something similar to what Loschky & Wolverton (2007) pre-calculated.

The experiment conducted used words instead of graphical stimuli. It is argued that the results could still apply to graphical stimuli given that the change from periphery to point of gaze is subtle in both the contrast and the form. Future work may investigate, using a similar experiment as described in Section 3.7, if there is any difference to when the "pop" is experienced using graphical stimuli instead of words.

As explained by Guenter et al. (2012a) changing the size of the eccentricity layers will decrease/increase the savings of foveated rendering. The current equation, see Equation 2 in Section 3.4, to calculate eccentricity layers size does not take into account the system latency. Modifying the equation to take into account the system latency would decrease savings but in return make foveated rendering possible on systems with lower system latency than concluded in the current study.

The current setup used had a monitor with V-sync. Investigating how alternative techniques to V-sync, i.e. G-sync[10] and Adaptive-sync[11] can affect the system latency and ultimately a user's detectability of "pop" would be of interest.

[10] http://www.geforce.com/hardware/technology/g-sync
[11] http://www.vesa.org/news/vesa-adds-adaptive-sync-to-popular-displayport-video-standard/

7 ACKNOWLEDGEMENTS

I would like to acknowledge and show my gratitude to Dr Veronica Sundstedt and Fredrik Lindh for their guidance as supervisors throughout the thesis, much obliged. The critical and thoughtful advices by Mårten Skogö and John Elvsjö have been of vital help to progress and explore different areas. I show my humble gratitude to Jonas Avemo, Richard Hainzl, Tobias Lindgren, Kenneth Häggmark, Anders Vennström, Roland Waltersson, Jenny Melander, Anders Olsson, Johan Bouvin, Mattias Kuldkepp, Ralf Biedert, Dalibor Jovanovic and Erik Holmgren for providing help in their specific field and helping out whenever needed. In addition to the previous mentioned people I would thank the staff at Tobii who despite my somewhat obvious questions has always replied friendly and helpful, thank you all!

Honorable mention to John Snyder and Brian Guenter that have openly and fast answered any question about their “Foveated 3D Graphics article”.

Last but not least I would like to thank a special person who has been with me throughout the thesis both in ups and downs, thank you Johanna Nilsson for the support you have shown me!

REFERENCES

Akenine-Möller, T., Haines, E., & Hoffman, N. (2011). *Real-Time Rendering* (Third Edit.). A K Peters, Ltd.

Andersson, J. (2012). 5 Major Challanges In Real-Time Rendering. In *SIGGRAPH*. Retrieved from http://dice.se/wp-content/uploads/Bps12_5MajorChallenges_Andersson.pdf

Andersson, R., Nyström, M., & Holmqvist, K. (2010). Sampling frequency and eye-tracking measures: how speed affects durations, latencies, and more. *Journal of Eye Movement Research*, *3*(3), 1–12.

Bahill, A., Clark, M., & Stark, L. (1975). The main sequence, a tool for studying human eye movements. *Mathematical Biosciences*, *204*, 191–204.

Carmack, J. (2013). Latency Mitigation Strategies. *February 22*. Retrieved April 02, 2014, from http://www.altdevblogaday.com/2013/02/22/latency-mitigation-strategies/

Conger, D. (2010). High-Performance Games: Addressing Performance Bottlenecks with DirectX*, GPUs, and CPUs. *September*. Retrieved May 29, 2014, from https://software.intel.com/sites/default/files/m/d/4/1/d/8/High-performance-Games.pdf

Duchowski, A. T. (2007). *Eye Tracking Methodology: Theory and Practice* (Second Edi.). Springer.

Ekström, M., & Larsson, L. (2010). *Metoder i kommunikationsvetenskap* (2.1 ed., pp. 13–22). Student Litteratur.

Gregory, J. (2009). *Game engine architecture*. CRC Press.

Guenter, B., Finch, M., Drucker, S., Tan, D., Snyder, J., & Microsoft Research. (2012a). Foveated 3D Graphics. In *ACM Transactions on Graphics* (Vol. 31, pp. 1–10). Guenter, B., Finch, M., Drucker, S., Tan, D., Snyder, J., & Microsoft Research. (2012b). *Supplement to Foveated 3D Graphics: User Study Details* (Vol. 31).

Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Van De Weijer, J. (2011). *Eye Tracking*. Oxford University Press.

Häikiö, T., Bertram, R., Hyönä, J., & Niemi, P. (2008). Development of the letter identity span in reading: Evidence from the eye movement moving window paradigm. *Journal of Experimental Child Psychology*, *102*(2), 167–181.

Keller, A., Karras, T., Wald, I., Aila, T., Laine, S., Jacco, B., … McCombe, J. (2013). Ray Tracing is the Future and Ever Will Be... In *SIGGRAPH*.

Kenny, A., Koesling, H., & Delaney, D. (2005). A preliminary investigation into eye gaze data in a first person shooter game. In *In Proceedings of European Conference on Modelling and Simulation* (pp. 733 – 740).

Levoy, M., & Whitaker, R. (1990). Gaze-Directed Volume Rendering. *ACM*, 217–223.

Li, R., Polat, U., Makous, W., & Bavelier, D. (2009). Enhancing the contrast sensitivity function through action video game training. *Nature Neuroscience*, *12*(5), 549–551.

Lopez, F., Molla, R., & Sundstedt, V. (2010). Exploring Peripheral LOD Change Detections during Interactive Gaming Tasks. *ACM*, 73–80.

Loschky, L. C., & McConkie, G. W. (2000). User Performance With Gaze Contingent Multiresolutional Displays. In *Proceedings of the 2000 symposium on eye tracking research & applications* (pp. 97–103). ACM.

Loschky, L. C., & Wolverton, G. S. (2007). How Late Can You Update Gaze-Contingent Multiresolutional Displays Without Detection? *ACM Transactions on Multimedia Computing, Communications, and Applications*, *3*(4), 1–10.

Mantiuk, R., Bazyluk, B., & Tomaszewska, A. (2011). Gaze-Dependent Depth-of-Field Effect Rendering in Virtual Environments. In *Serious Games Development and Applications* (pp. 1–12).

Mauderer, M., Conte, S., Nacenta, M. A., & Vishwanath, D. (2014). Depth Perception with Gaze-contingent Depth of Field. In *Conference on Human Factors in Computing Systems - Proceedings* (pp. 217 – 226). ACM.

Morrison, G. (2012). What is “Game mode”? *December 10*. Retrieved May 16, 2014, from http://www.cnet.com/news/what-is-game-mode/

Rayner, K., Castelhano, M. S., & Yang, J. (2009). Eye Movements and the Perceptual Span in Older and Younger Readers. *Psychology and Aging*, *24*(3), 755–760.

Sousa, T., Kasyan, N., & Schulz, N. (2012). CryENGINE 3: Three Years of Work in Review. In *GPU Pro 3* (pp. 133–168). CRC Press.

Wilson, D. (2009). AnandTech | Exploring Input Lag Inside and Out. *July 16*. Retrieved January 31, 2014, from http://www.anandtech.com/show/2803/5

Yarbus, A. L. (1967). Eye Movements and Vision. *Vision Science: Photons to Phenomenology*.

APPENDIX A

Additional screenshots from the experiment

Included below are additional screenshots from the application. They follow the order to which they were presented to a participant in the experiment.

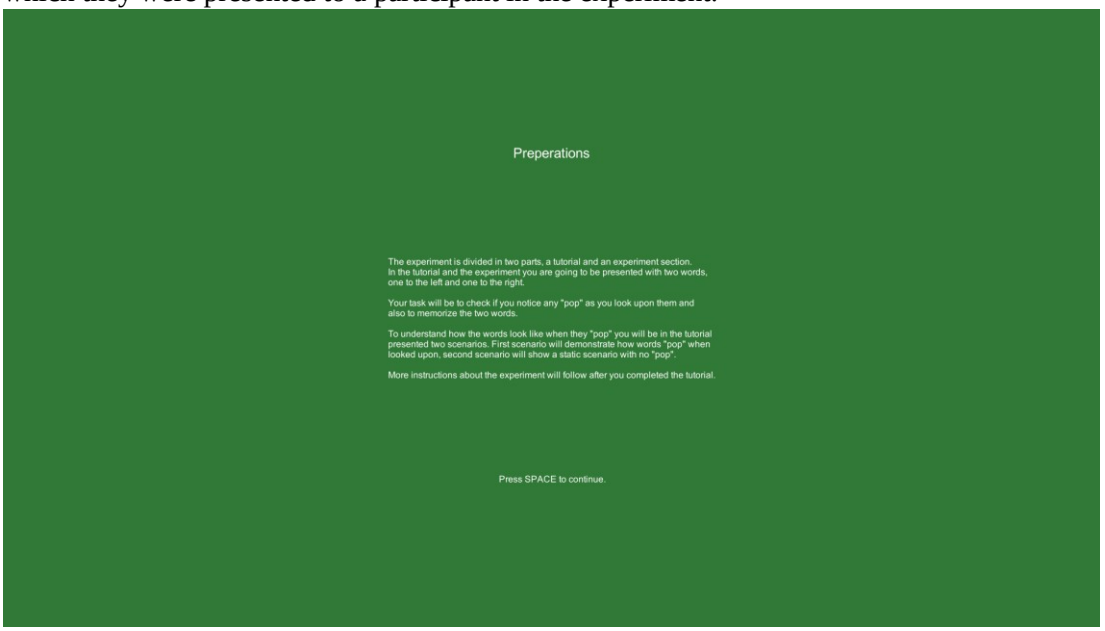

Figure 12: Shows the first scene of the experiment. It describes the process of the experiment. The quoted instructions can be seen in.

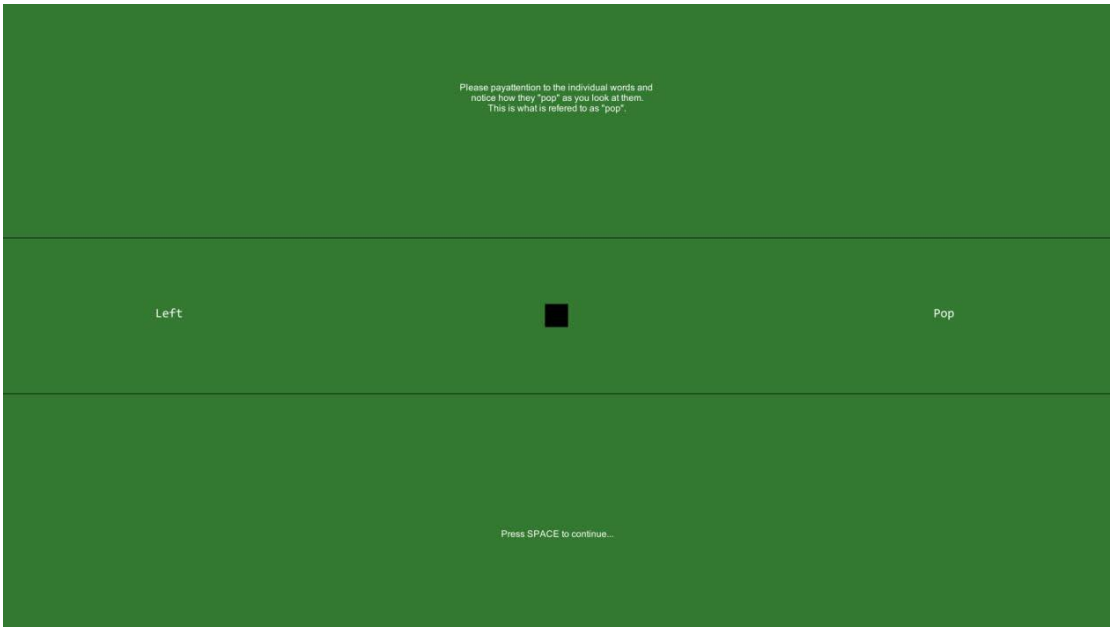

Figure 13: Shows the first tutorial scene demonstrating “pop”. The gaze is this screenshot focused on the word “Left”, where the right word “Pop” will change to “Right” when gazed at. System latency for the tutorial was roughly 80ms.

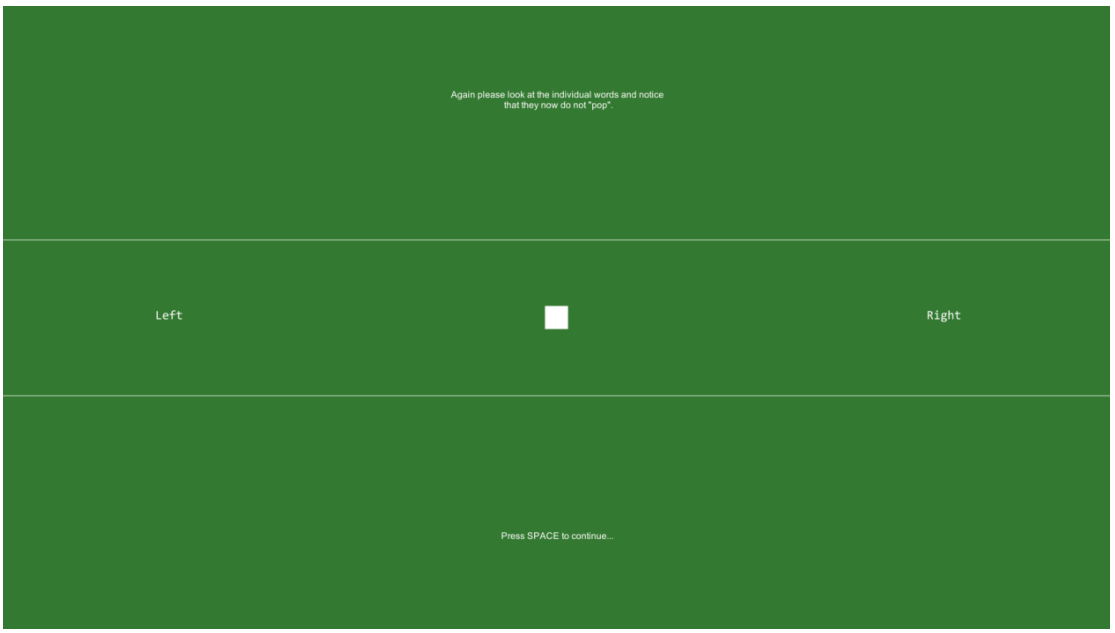

Figure 14: Shows the second tutorial scene. Here the words do not react on gaze and remain static to represent no-"pop". The change from black to white on the square and horizontal lines was to indicate a change of scene.

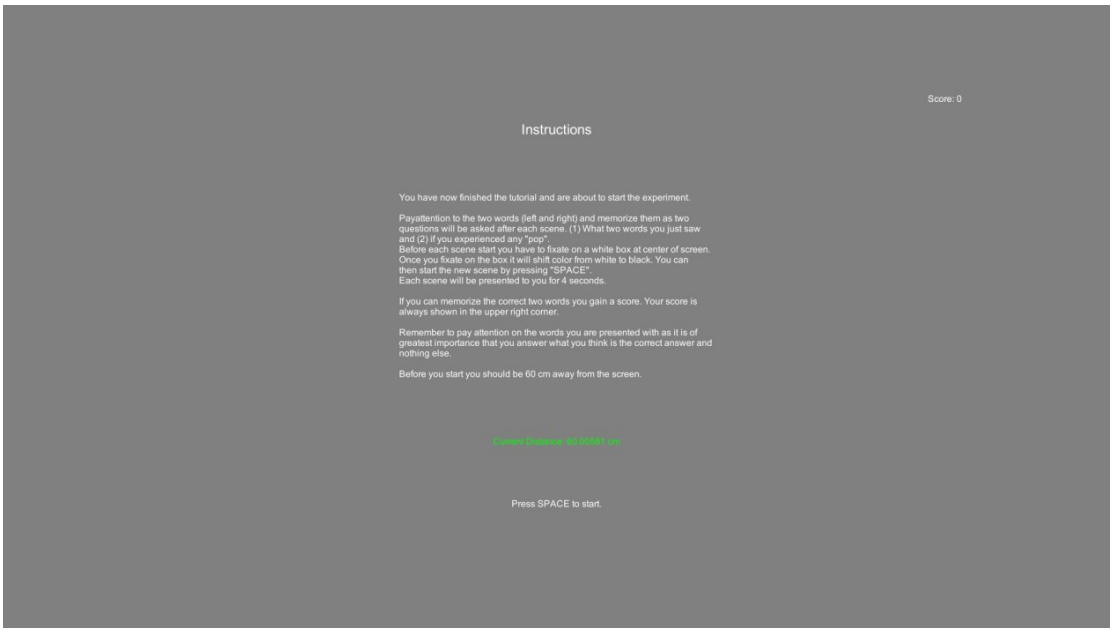

Figure 15: Shows the instructions given after completing the tutorial scene. Notice the text in green that shows the eyes' current position from monitor.

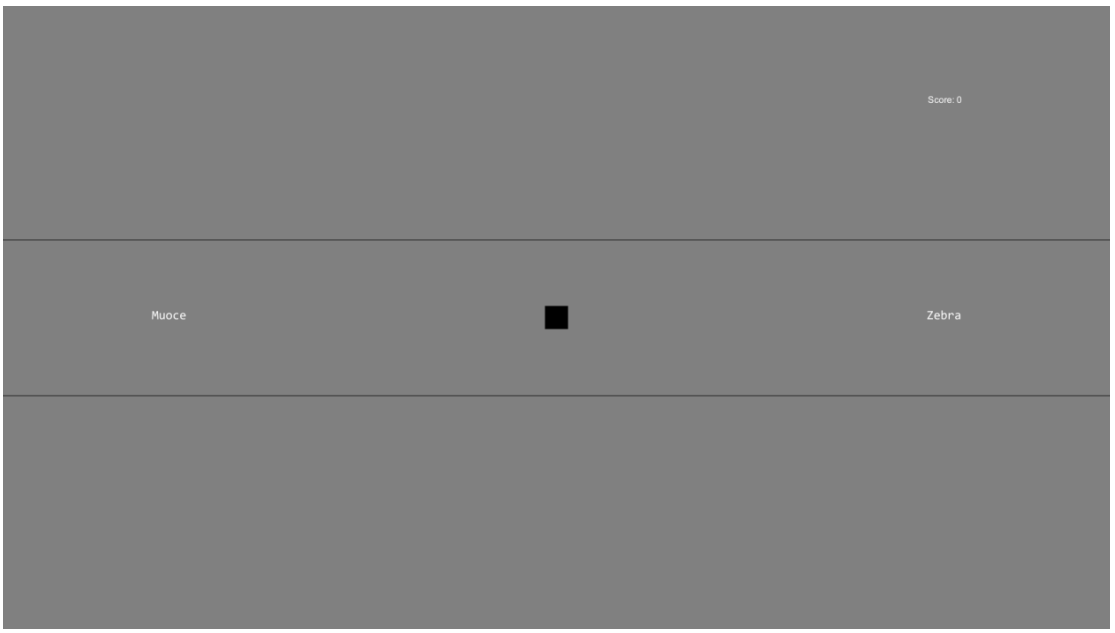

Figure 16: Shows one of the 34 experiment scenes a participant was exposed to during experiment. Here the gaze is focusing on the word to the right “Zebra”.

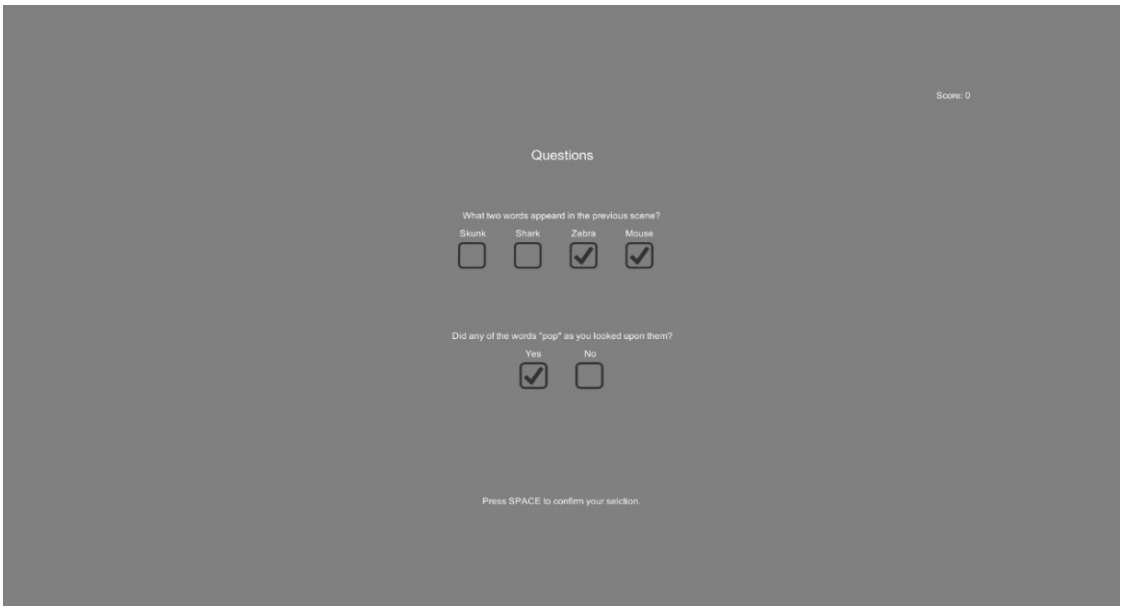

Figure 17: Shows a screenshot from the question scene presented to the user after an experiment scene had been shown.

Quoted instructions from the experiment

Included here are the exact instructions supplied to the test subjects. They follow the order to which they were presented to a participant in the experiment.

Preparation instructions

"*The experiment is divided in two parts, a tutorial and an experiment section. In the tutorial and the experiment you are going to be presented with two words, one to the left and one to the right.*

Your task will be to check if you notice any "pop" as you look upon them and also to memorize the two words.

To understand how the words look like when they "pop" you will be in the tutorial presented two scenarios. First scenario will demonstrate how words "pop" when looked upon, second scenario will show a static scenario with no "pop".

More instructions about the experiment will follow after you completed the tutorial."

Tutorial instructions

Tutorial demonstrating "pop"

"*Please payattention to the individual words and notice how they "pop" as you look at them. This is what is refered to as "pop".*"

Tutorial demonstrating no-"pop"

"Again please look at the individual words and notice that they now do not "pop"."

Instructions just prior to the start of the experiment

"You have now finished the tutorial and are about to start the experiment.
Payattention to the two words (left and right) and memorize them as two questions will be asked after each scene. (1) What two words you just saw and (2) if you experienced any "pop".

Before each scene start you have to fixate on a white box at center of screen. Once you fixate on the box it will shift color from white to black. You can then start the new scene by pressing "SPACE".

Each scene will be presented to you for 4 seconds.
If you can memorize the correct two words you gain a score. Your score is always shown in the upper right corner.

Remember to pay attention on the words you are presented with as it is of greatest importance that you answer what you think is the correct answer and nothing else.
Before you start you should be 60 cm away from the screen."

APPENDIX B

Additional measurements from the experiment

The measured data was collected from the 20 experiment runs and summarized into two tables. In Table 3 are the measurements from the experiments execution timestamps. They were measured using Unity's total time relative to previous frame.

Monitor	Tracker	Samples	Average (ms)	SD (ms)
60Hz	30Hz	4820	16.5664267054	0.300733082949
60Hz	45Hz	4820	16.5669964232	0.342923413999
60Hz	60Hz	4820	16.5666349378	0.384864924068
60Hz	85Hz	4820	16.5666538237	0.362044034161
60Hz	100Hz	4820	16.5668164398	0.375139798417
60Hz	120Hz	4820	16.5671032012	0.369667344624
85Hz	30Hz	6840	11.6647037424	0.386256262907
85Hz	45Hz	6840	11.6646643348	0.379377384624
85Hz	60Hz	6840	11.6646129284	0.3700521043
85Hz	85Hz	6840	11.6646712208	0.392897432055
85Hz	100Hz	6840	11.6644937515	0.372124591167
85Hz	120Hz	6840	11.6650534576	0.374905583665
100Hz	30Hz	8080	9.90044090817	0.282067019292
100Hz	45Hz	8080	9.90006656374	0.259763150453
100Hz	60Hz	8080	9.89976333849	0.260795509737
100Hz	85Hz	8079	9.90021313232	0.250126752735
100Hz	100Hz	8080	9.89976933069	0.238525147491
100Hz	120Hz	8080	9.89994303651	0.241491573626
120Hz	30Hz	9700	8.23329878247	0.173905849016
120Hz	45Hz	9700	8.23333159361	0.206584595011
120Hz	60Hz	9700	8.23327469381	0.239892816339
120Hz	85Hz	9700	8.23356305887	0.244886299083
120Hz	100Hz	9700	8.23330164093	0.237790413229
120Hz	120Hz	9700	8.23334067825	0.248394467525

Table 3: Shows detailed information of the raw application-timestamps. They are calculated by taking elapsed time minus the previous frames elapsed time.

In Table 4 are the measurements from the recorded gaze-data listed. Each gaze data that arrived contained a timestamp set by the engine on the computer. Measurements of precision were made by taking current gaze data's timestamp and subtract it with previous gaze data's timestamp.

Monitor	Tracker	Samples	Average (ms)	SD (ms)
60Hz	30Hz	2380	33.5253828289	4.62944281077
60Hz	45Hz	3570	22.3561418122	3.3212014748
60Hz	60Hz	4631	17.2259768472	3.00729958539
60Hz	85Hz	4801	16.636670644	5.81880476564
60Hz	100Hz	4811	16.5998226111	5.00449456375
60Hz	120Hz	4805	16.6195541144	3.83508512794
85Hz	30Hz	2391	33.3850919627	2.15613778427
85Hz	45Hz	3587	22.2440401646	1.17350333903
85Hz	60Hz	4749	16.7980045552	3.58406878341
85Hz	85Hz	6547	12.1854588322	2.66744701417
85Hz	100Hz	6683	11.9394851997	4.94024528702
85Hz	120Hz	6773	11.7818705346	4.14110515822
100Hz	30Hz	2401	33.3293316709	0.0290925450328
100Hz	45Hz	3587	22.2935455807	3.01314279675
100Hz	60Hz	4779	16.7345034305	1.36304018674
100Hz	85Hz	6628	12.0028249295	2.28635614823
100Hz	100Hz	7723	10.360189487	1.86602138197
100Hz	120Hz	7904	10.1170530744	3.60072890584
120Hz	30Hz	2387	33.4828592212	3.79496112671
120Hz	45Hz	3584	22.2818183899	2.30377447778
120Hz	60Hz	4771	16.7532685983	2.33126579105
120Hz	85Hz	6704	11.9182028031	3.3699880014
120Hz	100Hz	7824	10.2123590325	2.64482346117
120Hz	120Hz	9199	8.6828736161	2.5067140731

Table 4: Shows detailed information of the raw gaze-timestamps. They are calculated by taking elapsed time minus the previous frames elapsed time.

Fel! Hittar inte referenskalla. is a cheat sheet with precomputed estimated values for configurations mentioned in the current study. Estimations are made using Equation 1 (Section 3.3).

(Eye-Tracker, Monitor) (Hz,Hz)	Best case(ms)	Worst case(ms)	Average case(ms)
(30, 60)	58	92	75
(30, 85)	53	77	65
(30, 100)	51	72	61
(30, 120)	50	67	58
(30, 144)	48	62	55
(45, 60)	47	81	64
(45, 85)	42	66	54
(45, 100)	40	61	50
(45, 120)	39	56	47
(45, 144)	37	51	44
(60, 60)	41	75	58
(60, 85)	36	60	48
(60, 100)	35	55	45
(60, 120)	33	50	41
(60, 144)	32	46	39
(85, 60)	36	70	53
(85, 85)	32	55	43
(85, 100)	30	50	40
(85, 120)	28	45	37
(85, 144)	27	41	34
(100, 60)	35	68	51
(100, 85)	30	54	42
(100, 100)	28	48	38
(100, 120)	26	43	35
(100, 144)	25	39	32
(120, 60)	33	67	50
(120, 85)	28	52	40
(120, 100)	26	47	36
(120, 120)	25	42	33
(120, 144)	23	37	30
(144, 60)	32	65	48
(144, 85)	27	51	39
(144, 100)	25	45	35
(144, 120)	23	40	32
(144, 144)	22	36	29
(250, 60)	29	62	45
(250, 85)	24	48	36
(250, 100)	22	42	32
(250, 120)	20	37	29
(250, 144)	19	33	26
(300, 60)	28	62	45
(300, 85)	23	47	35
(300, 100)	21	42	31
(300, 120)	20	37	28
(300, 144)	18	32	25

Table 5 Shows the estimated system latencies for a given configuration used in the current study. It does not take in to account V-sync.