

# Design and Development of Game-based Oculomotor Training using Eye Tracking

Håvard Homme Pettersen

Master's thesis in Software Engineering at

Department of Computing, Mathematics and Physics, Western Norway University of Applied Sciences

Department of Informatics, University of Bergen

June 2018



Western Norway  
University of  
Applied Sciences



University of Bergen

## Abstract

Many people have oculomotor issues. This means that they have a problem with how their eyes behave, or more importantly how they behave together. These issues are present both in children and adults and could lead to the patient being insecure in crowds, disoriented or having trouble reading.

This thesis will focus on creating a computer application for oculomotor training. It will cover most of the important primitive eye movements, and in some cases combine them with other cognitive abilities.

The application consists of six games, together covering the most important fields in a table provided by vision teacher Tina Ihler. This table contains the most important parts in oculomotor training and includes examples of how the different issues can be trained. The games cover about  $\frac{3}{4}$  of the challenges presented here, because not all the areas were possible to train without real depth perception.

Most of the games use eye tracking as the main input method. This makes the user able to control the game behavior using their vision, either by gaze-based or gaze-added interaction. This application uses both, and for the gaze-added interaction both the spacebar and the mouse functions as a sufficient additional input method.

The application has been tested by multiple vision teachers who filled out a pre-defined form. They all considered the application to be a potential supplement to the existing training methods, but not as something that could completely replace it. An in-depth evaluation showed that the main reason it cannot replace existing methods is the lack of depth perception and other cognitive abilities in connection with the eyes. On the other hand, it showed that eye tracking is a very good tool for training some oculomotor issues.

## Acknowledgements

Firstly, I would like to thank Gunvor Birkeland Wilhelmsen for providing the idea for this thesis, and for great insight into the field of oculomotor training. I would also like to thank Heidi Jåstad for being helpful by providing expert opinions and information during the process.

Special thanks to vision teacher Tina Ihler for providing very good and useful information about how to train the different oculomotor abilities.

I would like to thank my supervisors Carsten G. Helgesen, Ilona Heldal and Atle B. Geitung for good support throughout the process. Your feedback has been invaluable.

Lastly, I would like to thank Mads Gjerstad Eide and Ruben Watanabe for very good help and good discussions. The introduction to the field of eye tracking would have been much more difficult without your support.

## Glossary

Binocular	Using both eyes
COGPACK	Application for training of cognitive abilities created in the 80's by Klaus Marker
Eye-hand coordination	The interaction between the eyes and the hands. Hands are making movements according to information from the eyes.
Eye Tracking	The act of tracking the eye's position at any given time.
Fixation	When the eyes are completely still, keeping focus on any given object
Gaze point	The point on a screen where the eyes are looking
GUI	Graphical user interface. The link between the system and the user.
High acuity vision	The central vision. Norwegian: Skarpsyn
Oculomotor	Eye movement
Peripheral vision	The part of the vision that does not have high acuity. Norwegian: Sidesyn
Saccades	Rapid eye movement from one place to another
Serious games	Games used in rehabilitation or training
Smooth pursuit	Eyes are following a constantly moving object
Visual neglect	Reduced attention of the room or your body on one side. May be present after brain injury

## List of figures

Figure 1 Visualization of a saccade. Source:

<https://www.ncbi.nlm.nih.gov/books/NBK10991/figure/A1358/?report=objectonly> ..... 4

Figure 2 Visualization of a smooth pursuit. Source:

<https://www.ncbi.nlm.nih.gov/books/NBK10991/figure/A1359/?report=objectonly> ..... 5

Figure 3 Training in Tanzania. Photos: Mats Gjerstad Eide..... 9

Figure 4 Apparatus used in eye tracking. Source: Wikimedia

([https://en.wikipedia.org/wiki/Eye\\_tracking#/media/File:Yarbus\\_eye\\_tracker.jpg](https://en.wikipedia.org/wiki/Eye_tracking#/media/File:Yarbus_eye_tracker.jpg))..... 13

Figure 5 VT3Mini from EyeTech. Source:

<https://www.eyetechds.com/uploads/4/7/3/0/47308795/1456176188.png> ..... 14

Figure 6 Tobii EyeX. Source: <http://tobiigaming.com/product/tobii-eyex/>..... 15

Figure 7 Screen based eye tracking. Source: [tobiipro.com/learn-and-support/learn/eye-tracking-essentials/how-do-tobii-eye-trackers-work/](http://tobiipro.com/learn-and-support/learn/eye-tracking-essentials/how-do-tobii-eye-trackers-work/)..... 16

Figure 8 Heatmap from Tobii Pro Glasses. Source: <https://www.tobiipro.com/siteassets/tobii-pro/brochures/tobii-pro-glasses-2-brochure.pdf?v=5> ..... 17

Figure 9 Assassin's Creed Origins gameplay with eye tracking. Source:

[https://www.youtube.com/watch?time\\_continue=115&v=ZSoDSiI0mZw](https://www.youtube.com/watch?time_continue=115&v=ZSoDSiI0mZw) ..... 19

Figure 10 Eye tracking in Farming simulator. Source: <https://tobiigaming.com/games/farming-simulator17/> ..... 20

Figure 11 COGPACK by Marker Software..... 22

Figure 12 Technologies replaced by WPF. Source: <https://msdn.microsoft.com/en-us/library/aa663364.aspx> ..... 27

Figure 13 Sample WPF application. Source: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/getting-started/walkthrough-my-first-wpf-desktop-application>..... 28

Figure 14 Empty Unity project ..... 29

Figure 15 Cocos Creator interface. Source: <http://www.cocos2d-x.org/creator>..... 30

Figure 16 Tobii Calibration application..... 35

Figure 17 Game 1 - Compare Words..... 37

Figure 18 Game 2 – Maze..... 38

Figure 19 Game 3 - Save the rocket..... 39

Figure 20 Game 4 - Find the outstanding object.....	40
Figure 21 Game 5 - Follow the ball .....	41
Figure 22 Game 6 - Which one is missing (1st screen) .....	42
Figure 23 Game 6 - Which one is missing (2nd screen).....	43
Figure 24 UML diagram of the application .....	46
Figure 25 Eye Tracker integration .....	47
Figure 26 Eye anatomy. Source: <a href="http://www.moorfields.nhs.uk/sites/default/files/styles/content_page_full_width_500x280/public/AnatomyOfTheEye500x280.jpg">http://www.moorfields.nhs.uk/sites/default/files/styles/content_page_full_width_500x280/public/AnatomyOfTheEye500x280.jpg</a> .....	72
Figure 27 Muscles in action for different eye movements Source: <a href="https://www.ncbi.nlm.nih.gov/books/NBK10793/figure/A1355/?report=objectonly">https://www.ncbi.nlm.nih.gov/books/NBK10793/figure/A1355/?report=objectonly</a> .....	73
Figure 28 Rods and cones density. Source: <a href="http://hyperphysics.phy-astr.gsu.edu/hbase/vision/rodcone.html">http://hyperphysics.phy-astr.gsu.edu/hbase/vision/rodcone.html</a> .....	74

## Table of Content

1	Introduction .....	1
1.1	Background .....	1
1.2	Current Situation .....	2
1.3	Goal .....	2
1.4	Research Questions .....	2
1.5	Limitations .....	3
2	Background.....	4
2.1	Movement.....	4
2.2	Visual Attention .....	6
2.3	Problems.....	6
2.3.1	Causes .....	6
2.3.2	Consequences.....	7
2.4	Lack of Peripheral Vision (Homonym Hemianopsia).....	7
2.5	The Training Process.....	8
2.6	Eye Tracking .....	11
2.6.1	History.....	11
2.6.2	EyeTech .....	14
2.6.3	Tobii .....	14
2.6.4	Technology used in Tobii Trackers .....	15
2.6.5	Gaze Interaction .....	18
2.6.6	Eye Tracking in Modern Games .....	18
2.7	Serious Games.....	21
2.8	Previous work.....	22
2.8.1	COGPACK .....	22
2.8.2	VisionBuilder .....	23
3	Problem analysis.....	24
3.1	What the Training Should Cover .....	24
3.2	Screen Representation .....	25
3.2.1	Saccades .....	25
3.2.2	Visual Attention .....	25

3.2.3	Smooth Pursuit.....	25
3.2.4	Visual Memory .....	25
3.2.5	Eye-hand Coordination .....	26
3.3	Development Choices .....	26
3.4	Choosing Framework .....	26
3.4.1	Windows Presentation Foundation .....	26
3.4.2	Unity .....	28
3.4.3	Cocos2D.....	30
3.4.4	Discussion.....	31
3.5	API.....	32
3.5.1	Tobii Unity API .....	32
3.5.2	Gaze API.....	32
3.5.3	Logging.....	33
4	Solution.....	34
4.1	General .....	34
4.2	Calibration.....	34
4.3	Game 1 – Compare Words.....	36
4.4	Game 2 - Maze .....	37
4.5	Game 3 – Save the Rocket .....	38
4.6	Game 4 – Find the Outstanding Object.....	40
4.7	Game 5 – Follow the Ball .....	41
4.8	Game 6 – Which One is Missing?.....	42
4.9	Fitting the Games into the Exercise Table .....	44
4.10	Architecture.....	45
5	Implementation.....	48
5.1	Initial .....	48
5.2	Game Objects .....	48
5.3	Gaze API .....	49
5.4	Checking for Hits .....	50
5.5	Keeping Track of Time .....	51
5.6	Scene Management .....	51
6	Evaluation.....	53



6.1	Testing Method .....	53
6.2	Results .....	55
6.3	Interpretation of the Answers .....	60
6.4	In-depth Evaluation of the Games.....	60
7	Conclusion .....	64
7.1	Question 1 .....	64
7.2	Question 2 .....	64
7.3	Question 3 .....	65
7.4	Question 4 .....	66
7.5	Final.....	66
8	Further Work .....	67
9	References .....	68
Appendix A: Eyes.....		72
A I	Anatomy .....	72
A II	Extraocular Muscles .....	73
A III	Rods and cones .....	74

# 1 Introduction

## 1.1 Background

Many people struggle with oculomotor issues. This means that their eye movement do not work in a satisfying manner, or that the high acuity vision is not properly trained to compensate for the lack of a peripheral vision. Many of these people are wrongly diagnosed and are living with a problem that could easily be solved (Vareberg, 2012). Examples of diagnoses that is often thrown around for children that struggle with reading or concentrating are ADHD, dyslexia and learning difficulties. Many of these children have oculomotor issues instead, which could be trained to gain a better quality of life. Wrong diagnoses given by doctors may lead to the wrong treatment, which will result in high usage of resources without the problem getting better.

In 2017, Mads Gjerstad Eide and Ruben Watanabe at Western Norway University of Applied Sciences delivered their master's thesis about detection of oculomotor issues in children, where they worked together with Gunvor Birkeland Wilhelmsen from the Faculty of Education, Arts and Sports. The solution that evolved from this thesis proved to be very accurate and became a very useful tool for discovery of such issues. One way this application is used today is to detect initial oculomotor issues, and track improvement in between training sessions. (Eide & Watanabe, 2017)

In addition to detection of these problems, Gunvor Wilhelmsen suggested that the existing computer applications for training of oculomotor issues were outdated and lacking important features, so she decided that there was a need for a newer training application to complement the manual training. That was the initial problem that led to this thesis.

A good and modern solution with better interaction methods may be beneficial both during the training sessions, and for people to use at home. This could potentially lead to a lot more people getting the possibility to train their oculomotor issues, if the solution is made easy and available enough.

## 1.2 Current Situation

There exist some digital solutions, like one called COGPACK that is used by Gunvor and her coworkers today. COGPACK was made in 1986 and is now becoming dated mostly because of primitive input methods. The problem with the traditional input methods for training of oculomotor issues is the excessive need for mouse and keyboard usage and lack of information about the user's vision during the training. Because of this, there is a need for a better computer application to complement the physical training, and maybe replace some of the methods used today.

Gunvor Wilhelmsen has stated that it would not be possible to replace all the exercises performed today, because computer screens lack depth of vision and proper use of peripheral vision. Because of this, the current situation is that only the computer applications should be replaced to complement the other training methods. It would be interesting to look at different methods to make the game experience better than in current solutions.

## 1.3 Goal

The goal of this thesis is to demonstrate how eye tracking can be used to in applications to complement or replace existing training methods for oculomotor issues. The prototype will feature different types of pedagogic games that the user will have to control using his or her eyes.

## 1.4 Research Questions

1. How can computer games complement existing manual methods for oculomotor training?
2. Which benefits or drawbacks do computer applications have compared to other training methods?
3. How suitable is eye tracking technology in training of oculomotor issues?
4. Is there a need for other input methods in addition to eye tracking?

## 1.5 Limitations

Since the project span is less than a year, there might be some challenges connected to the proof of effect. According to vision teachers, it takes around 3 weeks of every day training to see measurable effect. Because the priorities of this project have been on development and connection to existing knowledge, such training would not take place. Therefore, the potential usefulness of the application will be measured in other ways, like from the educator's point of view.

All the information about what effect the application should have and how it should behave needs to come from the experts in the visual education field. It will not be possible to gain enough understanding about the eyes and oculomotor issues to make an application based on that knowledge in the short time span of this thesis. Instead, a couple of vision teachers will follow the process and provide sufficient information needed to develop an application.

There is currently one eye tracker available for the project, Tobii EyeX, which will be used for this project. Other eye trackers will be mentioned but not considered.

## 2 Background

This chapter will be about eye movements, visual attention and problems that may occur in people who struggle with oculomotor issues. For background information about eye anatomy, eye muscles and color vision, see Appendix A.

### 2.1 Movement

The human eye is constantly moving, even though it appears stable. These constant movements are called saccades, and is the movement performed while switching focus between objects or when creating a complete image of any object. The saccades may vary in size according to what activity is performed. For example, while reading, the saccades are much smaller than when moving the focus around in a big room and vice versa.

Saccades are usually done between fixations and lasts until the target object is reached. During a saccade the brain is not able to register movement of the target object, so if the object moves before the eye reaches it the eye must perform a second saccade to correct the error. (Purves et al., 2001)

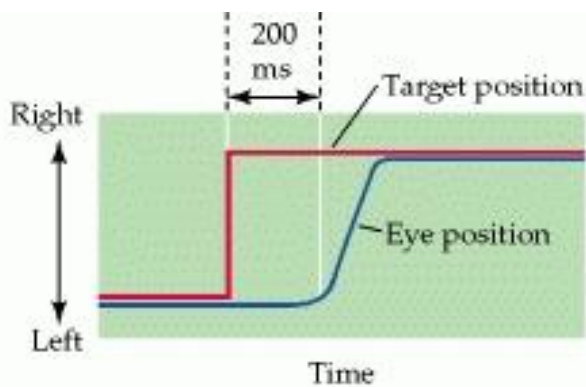


Figure 1 Visualization of a saccade. Source: <https://www.ncbi.nlm.nih.gov/books/NBK10991/figure/A1358/?report=objectonly>

Small saccades are often referred to as micro saccades and is performed while reading or when studying a specific object. These are usually not noticeable, but is performed all the time, even during fixations.

Fixations are the brief period while the eye are not moving. It occurs continuously while the eyes are pointing toward a single point in space, and it usually lasts for about 200ms. During the fixation, the brain is calculating which direction the next saccade should lead. If the target object is already reached, the eyes are constantly performing micro saccades to keep focus and create an overview of the object. (Rucci et al., 2018)

There are two more types of movements of the human eye. The first one is called smooth pursuit movement and is the movement the eyes do when following a smoothly moving object through the visual field. This type of movement is voluntary and not outside the person's control like most saccades. This means that the user can at any point choose not to follow the passing object.

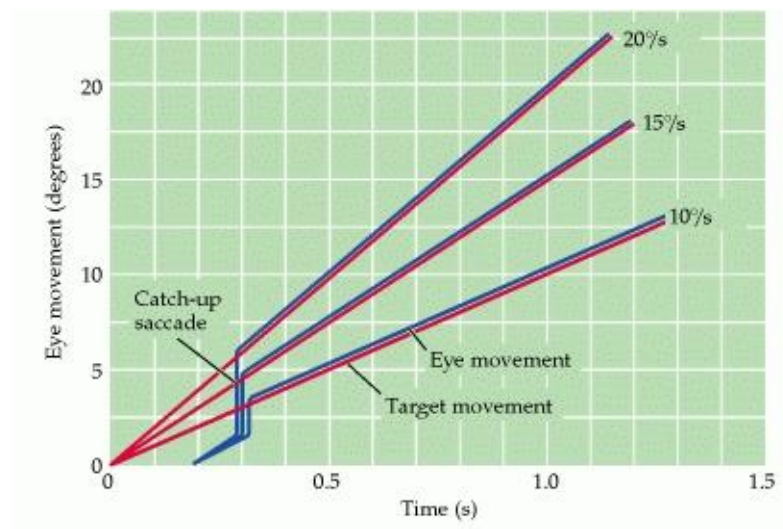


Figure 2 Visualization of a smooth pursuit. Source: <https://www.ncbi.nlm.nih.gov/books/NBK10991/figure/A1359/?report=objectonly>

The last type of eye movement is called Vestibulo-ocular movements. This is a type of movement that makes the brain compensate the vision for head movements and makes the eyes able to keep a certain object in focus even with major head movements. This movement is not directly relevant to this thesis but is mentioned to give a complete image of the most common eye movements. (Rucci et al., 2018)

## 2.2 Visual Attention

The brain does not always process what the gaze is pointing at. This means that you may look at something without giving it your attention, and therefore not remembering what you saw (Frisby and Stone, 2010). Because of this, any tracking will lack sufficient information about the cognitive abilities the user has in connection with their eyes which is an important measure when working with oculomotor issues. To conquer this, the cognitive abilities must be measured in other ways not directly related to the eye gaze like different kinds of input methods.

The question of visual attention vs visual perception is a field that is tied together with the study of consciousness, and it is not possible to accurately measure at the time of writing this thesis (Frisby and Stone, 2010).

## 2.3 Problems

The research paper «Friske øyne – svekket syn» by Gunvor Birkeland Wilhelmsen and Vibeke Dons Vankel (2014) looks at causes and issues connected to oculomotor issues. This sub-chapter will list the causes for both children, adults and elderly people, and look at the consequences the condition may have.

### 2.3.1 Causes

The causes of oculomotor issues often depend on what age the user is and if the condition has been present since birth. The number of possible interfering conditions increase with age.

Following is a list of the possible reasons for oculomotor issues:

- Children: most likely an innate abnormality or a tumor.
- Youths: MS, infection, tumor or a metabolic disease
- Adults: Tumor, cerebrovascular disease, infection, Myasthenia gravis, diabetes or a rheumatic disease
- Elders: Parkinson's disease, dementia, tumor or Myasthenia gravis

Oculomotor issues may also evolve from certain lifestyle factors. One of these are lack of stimulation in children, for instance from overuse of computers or tablets. (Wilhelmsen, personal communication, 2018).

### 2.3.2 Consequences

Living with oculomotor issues may give many problems that decrease quality of life. These problems include, but are not limited to:

- Disorientation
- Insecurity in cities or crowds
- Discomfort while looking at moving things
- Trouble climbing stairs
- Headache
- Reading problems

## 2.4 Lack of Peripheral Vision (Homonym Hemianopsia)

Some diseases limit the person's peripheral vision, which influences how the eyes move. With lack of proper peripheral vision, there is sometimes a lack of visual stimulus in these parts of the eye which leads to the person not seeing things outside the gaze. When walking in traffic or other situations where such attention is needed, these people struggle with navigating and avoiding obstacles. Some people only has vision on one side of the visual field, and they often need to turn their head in a specific direction to perceive everyday objects.

This kind of visual impairment can be trained by using compensational saccades to move the high-acuity vision around in the scene. This means that the person must be trained to continuously move the eyes around to perceive the environment. According to a study performed by Ivanov, Mackeben, Vollmer, Martus, Nguyen and Trauzettel-Klosinski (2016), patients with tunnel vision have good effect of this kind of training.

Compensational saccades would be a focus in some parts of the prototype. This will be noted in chapter 5.



## 2.5 The Training Process

Gunvor Wilhelmsen and several more vision teachers from Western Norway University of Applied Sciences are periodically running training programs to help people who struggle with oculomotor issues. During these programs, every person gets a personalized training program that suits their specific needs. The programs are sometimes run intensively in a short period, and sometimes last for a longer period. The vision teachers have previously been to Tanzania to run intensive programs for children and has trained some local educators on the matter. (Wilhelmsen, personal communication, 2018)

An important aspect of the training is creativity, where the educator needs to come up with ideas for exercises that fit the person's specific needs. This includes:

- Throwing a ball back and fourth
- Books like Where's Waldo
- Drawing through a maze using a pencil
- Finding matching items in a pile

Some of the persons may have a disability which prevents them from following the "traditional" training program, so these have a need for a personalized set of exercises. Some of the exercises involve books, others involve paper figures. Some of the exercises requires the participant to sit still, while others encourage full body movement. Many of the programs also asks much of the educator, in terms of movement and close attention.

In between the training sessions, some of the persons are tested for improvements by the application made by Mads Gjerstad Eide and Ruben Watanabe (2017), and the data is saved for later review and research.



Figure 3 Training in Tanzania. Photos: Mats Gjerstad Eide

Some vision teachers have stated that they are using computer application in the daily training. This involves COGPACK and other smaller games like iPad applications. In the long-lasting training programs, computer applications have commonly been used as a complementary solution to the physical methods. (Wilhelmsen, personal communication, 2018)

Because of the need for creativity when working with people who struggle with oculomotor issues, the vision teachers do not follow specific guidelines when doing the training. The closest one could get to this type of guideline is a table made by vision teacher Tina Ihler. It gives a pinpoint on what needs to be done to train the different parts of the visual abilities. This can be seen in table 1.

The exercises seen in this table are all broad reaching. This means that one exercise usually trains more than just one specific movement. For example, “Searching/scanning” seen in table 1, may train both fixations, endurance, saccades and mini saccades according to how the exercise is performed. For this reason, one challenge has multiple terms connected to it, and all the terms have all the exercises connected to them

<b>Challenge</b>	<b>Terms</b>	<b>Exercises</b>
<b>Field of view</b>	Saccades Visual attention Regression	Horizontal movements Vertical movements Diagonal movements Circular movements
<b>Visual acuity</b>	Fixations Endurance Saccades Mini saccades	Searching/scanning Find objects in a crowd Horizontal movements Vertical movements Diagonal movements Circular movements Smooth pursuit Find pairs/similarities Labyrinths point to point
<b>Stereopsis</b>	Accommodations Convergence Double vision	Movements and flashes at a distance – different depths Objects that varies in size
<b>Eye-hand coordination</b>		Use the keyboard based on events on the screen, and vice versa.

*Table 1 Training of different oculomotor challenges. Source: Private conversation with Tina Ihler*

## 2.6 Eye Tracking

In the book *Eye Tracking in User Experience Design*, Bergstrom, J., Schall, A., (2014) provides a good description of what eye tracking is:

*“Eye tracking is a methodology that helps researchers understand visual attention. With eye tracking, we can detect where users look at a point in time, how long they look at something, and the path their eyes follow. Eye tracking has been applied to numerous fields including human factors, cognitive psychology, marketing, and the broad field of human–computer interaction.”*

Eye tracking was historically an advanced and high-end concept that only researchers were familiar with. In recent years however, it has reached the masses and been adopted for other use cases like gaming. When the eye tracking is used to control the computer rather than understanding the visual attention, the technology naturally becomes interesting to more people than the researchers.

To be able to use eye tracking, there is need for some proper equipment. Today there exists powerful and easy to use consumer targeted eye trackers at a low price. This cheap equipment is accurate enough to provide information about all types of eye movements if properly logged and processed.

### 2.6.1 History

In the beginning of the 20<sup>th</sup> century the main way to study eye movements was through direct visual observation. This was sometimes done using a mirror, where the initiator stood behind the participant to avoid distraction. The challenge with this approach was that the initiator was only able to observe large eye movements, which led to inaccurate and less useful results. This emerged into equipment that could provide a magnified image of the eye, and thus making the experiments more accurate.

In 1914 the first academic paper about mechanical eye tracking was released. This method consisted of a lever and an arm which was connected to the subject's head. One end of the lever was touching the subject's eyes gently while the other end was drawing on a piece of paper. The method was by no doubt very intrusive to the user.

In the second mechanical eye tracker the lever was replaced by an air balloon that touched the eyes gently. Movement of the eyes then changed the air pressure inside the balloon and was recorded.

The third edition consisted of two cups that looked like contact lenses. These lenses were placed on the eyes, and a lever or thread was connected to these cups. The eye movements were then transferred to a recording system. (Yarbus, 1965)

In a paper released by Alfred L. Yarbus (1965), he describes some more sophisticated eye tracking than the methods mention above. One of the methods he used was a mechanical apparatus to track eye movements using “caps” which is a type of suction devices. This method provided the possibility to track the eyes without any pain, but it was still very intrusive to the user. There were 8 different types of caps, and the usage depended on the goal of the experiment.

The apparatus consisted of a chin rest and some cameras. A beam of light was sent from a source to the caps attached to the user’s eyes. The reflection of this light determined the eye position at any given moment. During the experiments, it was important that the user’s head was completely still, or else there would be insufficient data. This was the main function of the chin rest. (Yarbus, 1965)

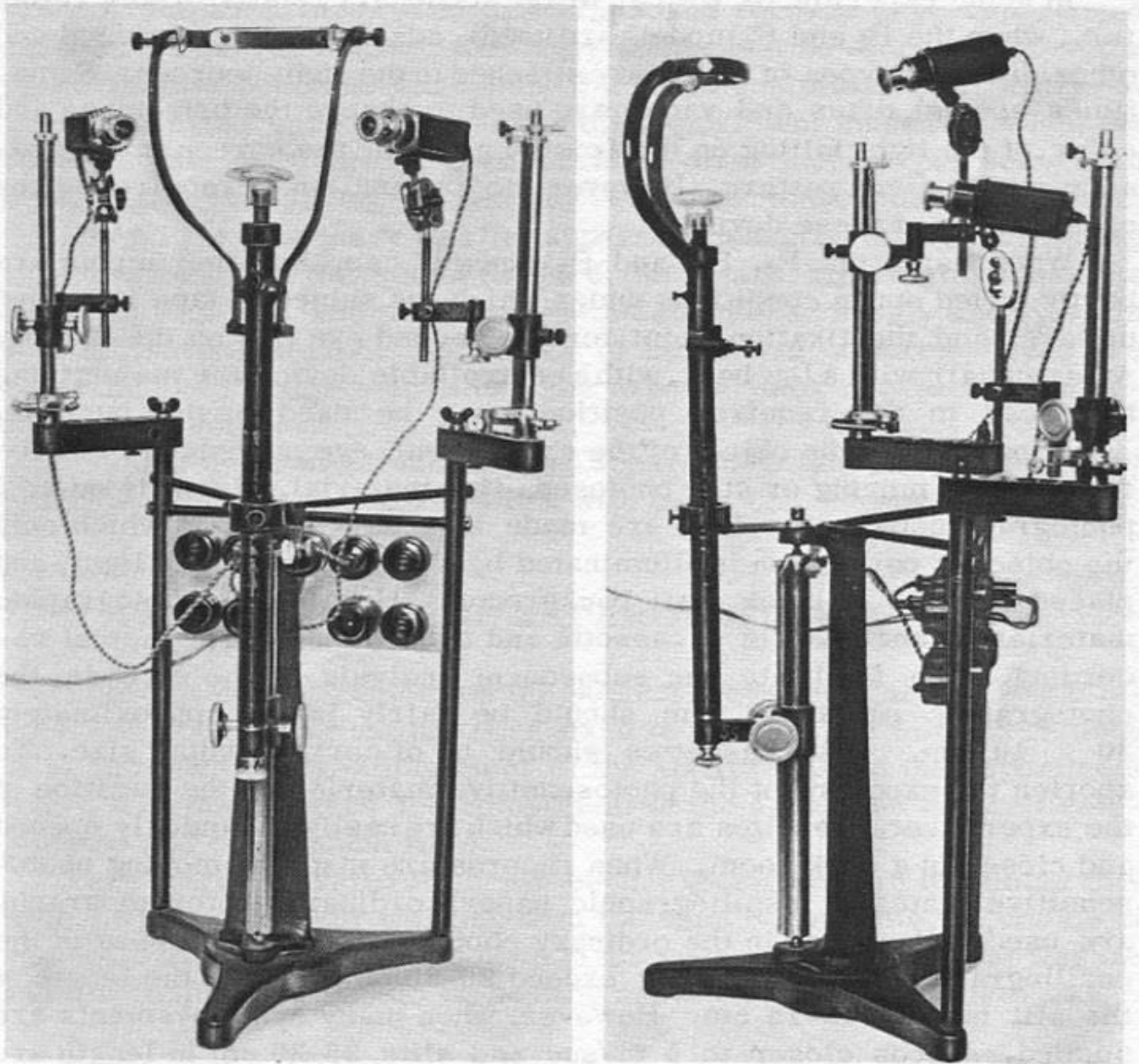


Figure 4 Apparatus used in eye tracking. Source: Wikimedia  
([https://en.wikipedia.org/wiki/Eye\\_tracking#/media/File:Yabus\\_eye\\_tracker.jpg](https://en.wikipedia.org/wiki/Eye_tracking#/media/File:Yabus_eye_tracker.jpg))

## 2.6.2 EyeTech

EyeTech is a technology company based in Arizona. They provide eye tracking hardware and software for eye trackers used in both assistive manners and research purposes. (EyeTech, 2018,

a) EyeTech provides 3 eye trackers for different use cases:

- TM5Mini: Serves as an addition to a speech device to use without the need of moving the hands. Used by people who struggle with ALS, Cerebral Palsy, Muscular Dystrophy and more. (EyeTech, 2018, b)
- VT3Mini: Made for research in areas like science, museums, malls, airports etc. This eye tracker can be mounted onto a computer screen to capture gaze position, blink rates and more. (EyeTech, 2018, c)
- VT3XL: Eye tracker made for large displays, primarily used for research. (EyeTech, 2018, d)



Figure 5 VT3Mini from EyeTech. Source: <https://www.eyetechds.com/uploads/4/7/3/0/47308795/1456176188.png>

## 2.6.3 Tobii

Because of availability of Tobii Eye trackers at Western Norway University College, this is the brand that will be used for this thesis.

Tobii is a Swedish company founded in 2001. At the time of writing the company has around 800 employees and offices in 9 countries. Tobii's main area of business is development and sale of eye tracking equipment to be used in gaming, science, health, design and more (Tobii, 2015, a). They describe themselves as the number one in eye tracking based on market share and portfolio. The Tobii Group consists of three units:

- Tobii Dynavox: Assistive technology for communication
- Tobii Pro: Eye tracking solutions to study human behavior
- Tobii Tech: Eye tracking for consumer technology

Tobii provides several pieces of hardware for consumers at a low price. Their most popular eye trackers are Tobii Eye Tracker 4C and Tobii EyeX for the respective price \$149 and \$99. Examples of other products currently available are laptops with integrated eye tracking from Acer, Dell and MSI. (Tobii, 2016)



Figure 6 Tobii EyeX. Source: <http://tobiigaming.com/product/tobii-eyex/>

One of Tobii's goals is to provide tools for research on treating ocular disease, autism, ADHD, Parkinson's and more. (Tobii, 2015, b) This means that they are willing to partner up with researchers who want to dive deeper into the field. To do this, Tobii provides partnership deals where the researcher gets access to more high-end equipment and get an extended license which allows for logging of the sessions.

#### 2.6.4 Technology used in Tobii Trackers

When using a non-intrusive Tobii eye tracker connected to the computer, it takes advantage of a concept called pupil center reflection (PCCR). This involves a near-infrared light that is sent to the eyes and a camera that captures the reflection of this light from the cornea and pupil. The vector formed by the reflection from the cornea and the pupil is then used to calculate the direction of the user's gaze on the screen.

When looking at the computer screen, the eye tracker continuously captures images of your eyes, sending them through image processing algorithms to identify reflection patterns. Then the information is sent through what Tobii calls a "sophisticated 3D eye model algorithm". This information can be received from the driver software as x and y coordinates. (Tobii, 2015, c)



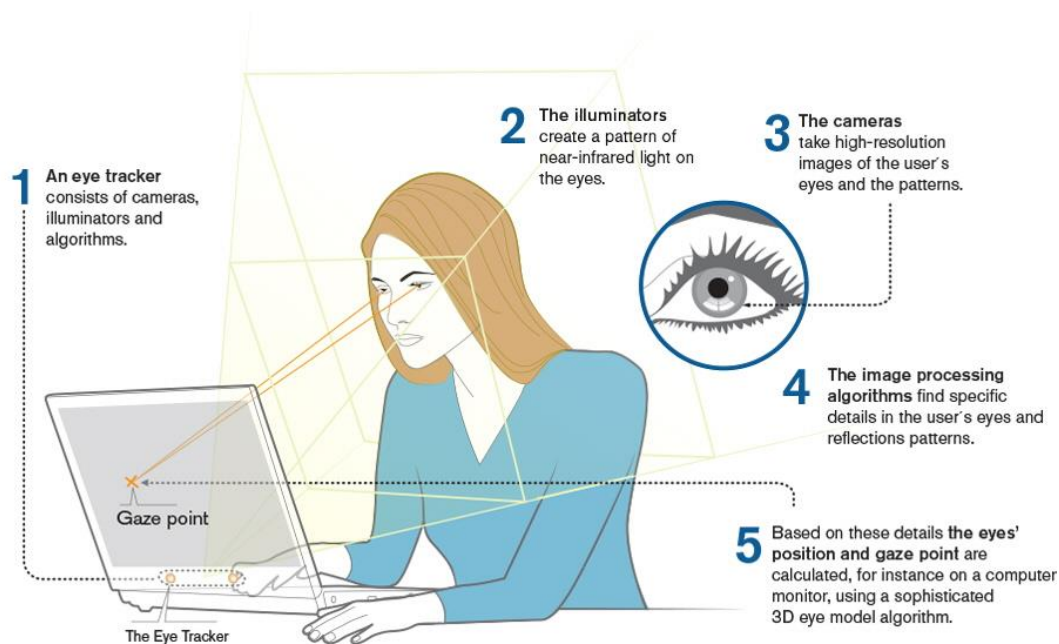


Figure 7 Screen based eye tracking. Source: [tobii.com/learn-and-support/learn/eye-tracking-essentials/how-do-tobii-eye-trackers-work/](https://tobii.com/learn-and-support/learn/eye-tracking-essentials/how-do-tobii-eye-trackers-work/)

Figure 7 describes a concept called screen-based eye tracking. With this concept the person's eye position translates to a specific point on the screen, called the gaze position. This gaze position is all that is needed to use the eye tracker for both research and gaming purposes.

Another eye tracking method Tobii has been developing is the use of glasses to track daily activities without having to be in front of a computer. These glasses are called Tobii Pro Glasses and was developed for research purposes. Tobii Pro Glasses can capture the user's gaze together with an image of the surrounding world, so researchers are able to understand how the visual attention works in different environments. This information is often translated into heat maps, which indicates how long the user was looking at the different areas.



Figure 8 Heatmap from Tobii Pro Glasses. Source: <https://www.tobii.com/siteassets/tobii-pro/brochures/tobii-pro-glasses-2-brochure.pdf?v=5>

The Tobii Pro Glasses consists of 4 eye cameras together with gyroscope and accelerometer. There is one 1080p scene camera that operates on 25fps, while the eye cameras are 50 or 100 Hz. The technology behind is fundamentally the same as in the screen-based trackers. There are cameras capturing images of the eyes, and algorithms that analyses these images to find the gaze direction. (Tobii, 2018)

## 2.6.5 Gaze Interaction

There are two types of computer interaction using the gaze:

- Gaze-based interface: GUI is solely controlled by the gaze
- Gaze-added interface: GUI is controlled by both the gaze and some other input devices

In a gaze-based interface, the interaction must involve the eyes, so pointing must be done through behavior of the eyes. One type of gaze-based interaction is blinking, where a brief closing of the eyes indicates a “click”. Another way is through brief fixations on a certain object.

In gaze-added interface, there is another external input method in addition to the eye input. The most common input method when using gaze-added interface is through a keyboard button or a mouse click. This could be as simple as if the space button indicates a “click” on the area the gaze is pointing at. (Lund, 2015)

## 2.6.6 Eye Tracking in Modern Games

In the previous years, Tobii eye tracking has been implemented into several games from publishers like Ubisoft and Square Enix. The eye tracking has been implemented for multiple use cases to better the game experience, both for 1<sup>st</sup> person and 3<sup>rd</sup> person games.

### Assassin’s Creed Origins

The last game of the Assassin’s Creed series from Ubisoft features Tobii integration out of the box. Assassin’s Creed Origins is a game that features a character walking around a city in Egypt around 50 B.C. This character is controlled by the player and is viewed as 1<sup>st</sup> and 3<sup>rd</sup> person depending on the situation. During the game, the player is given multiple tasks to complete to progress the storyline.

The eye tracking is used for the following features:

- Aim at gaze
- Enemy Tagging
- Interaction at gaze
- Dynamic light
- Sun effects

The eye tracking makes the user able to fire arrows or bullets at enemies by pointing the gaze in the desired direction which makes the gameplay easier for some people. Another feature is that when the user looks at the sun, the environment gets darker, like it would through human eyes.

The screenshot seen in Figure 9 is from a YouTube video showing how eye tracking is utilized in a specific situation where an enemy is approached and should be taken out. The user just looks at the target subject and clicks the aiming button. Then the aim points at the gaze position ready to shoot.

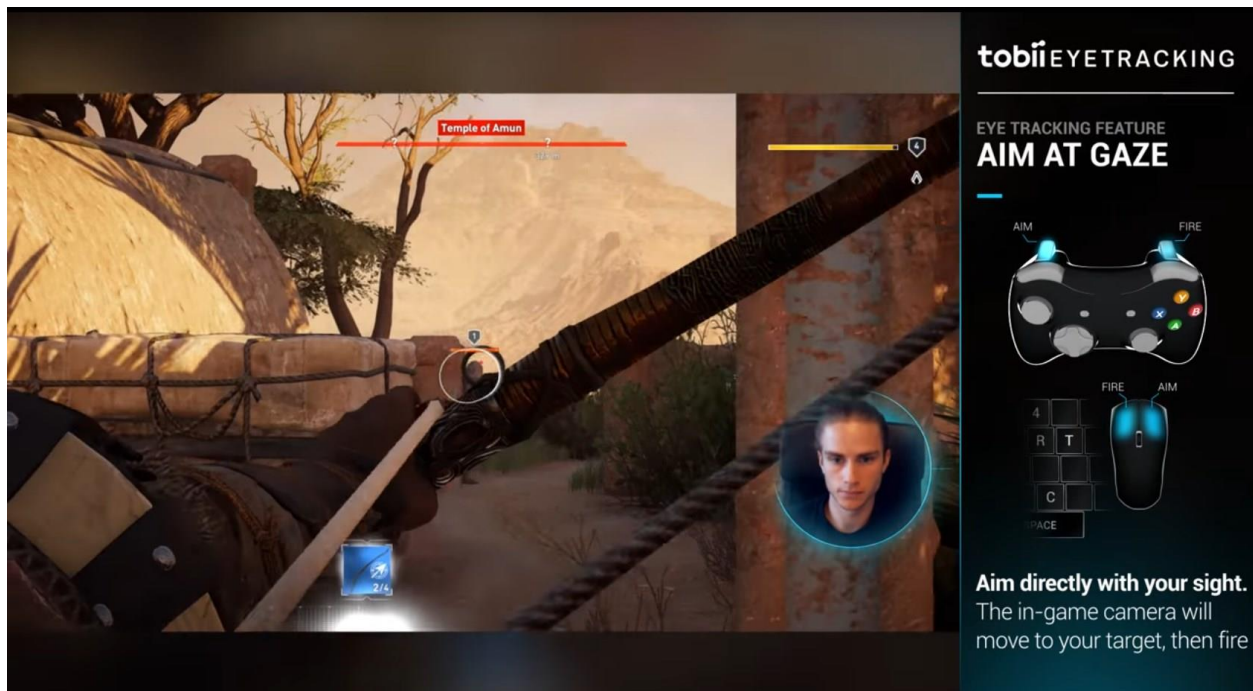


Figure 9 Assassin's Creed Origins gameplay with eye tracking. Source: [https://www.youtube.com/watch?time\\_continue=115&v=ZSoDSiI0mZw](https://www.youtube.com/watch?time_continue=115&v=ZSoDSiI0mZw)

Eye tracking in games are implemented to make the game more realistic, and to make the interactions more natural. It is also in some cases made to make the game easier, eliminating excessive need for eye-hand coordination. (Tobii, 2017, b)

## Farming Simulator 2017

Farming Simulator is a different type of game. The player controls a farmer who needs to do all the tasks that a farm requires. This includes sowing and harvesting at specific times. Farming Simulator is a simulation game, which means that there is no set goal to complete the game.

In farming simulator, the Tobii tracker can be used to get extended view, just like moving the eyes in real life. One example that is brought up is that when riding a tractor, the user may move his/her eyes around to move the screen from side to side. Of the eyes are moved to the left, the player can look to the left of the character and vice versa. (Tobii, 2017, a)



Figure 10 Eye tracking in Farming simulator. Source: <https://tobiigaming.com/games/farming-simulator17/>

For this game eye tracking won't give any advantages in performance, since performance is not measured. But it may add another dimension of realism to the already realistic simulation game.

The issue with eye tracking in Farming simulator is that the eye movements makes the character's head move, which does not feel too realistic. To reach a higher sense of realism there should be a higher degree visual field, which cannot be achieved on computer screens.

Both games mentioned here are 3D games where the main viewpoint is through the character's eyes. This makes eye tracking contribute more to realism than in 3<sup>rd</sup> person or 2D games. In 2D games, eye tracking could be used in a more useful manner but would not necessarily contribute to the level of realism.

## 2.7 Serious Games

According to Tarja Susi, Mikael Johannesson and Per Backlund (2007), there are as many definitions of serious games as there are actors in the market. One common definition is “digital games that is used for other purposes than only entertainment”. Examples of similar concepts are E-learning, edutainment, game-based learning and digital game-based learning. These terms are used in a broad description of different ways to use games as something educational or enhancing.

Serious games could be described as games that involve pedagogy, but where the entertainment part comes first, and the learning or rehabilitation happens without the user being fully aware of it (Susi et al., 2007). But it does not need to be without the user being aware of the training, for instance if a soldier is using a computer game for training before going into battle.

In their paper, Susi, Johannesson and Backlund (2007) provides a table with overview of serious games vs normal games. This table is interesting because it clearly shows a different expectation between the two types of games.

	<b>Serious games</b>	<b>Entertainment games</b>
<b>Task vs. rich experience</b>	Problem solving in focus	Rich experiences preferred
<b>Focus</b>	Important elements of learning,	To have fun
<b>Simulations</b>	Assumptions necessary for workable simulations	Simplified simulation processes
<b>Communication</b>	Should reflect natural (i.e., non-perfect) communication	Communication is often perfect

Table 2 Serious games vs entertainment games. Source: <http://www.diva-portal.org/smash/get/diva2:2416/FULLTEXT01.pdf>

The concepts in table 2 are not completely agreed upon, because people draw the line between serious and entertainment at different places.

There exist numerous sub-fields inside the field of serious games. These include, but are not limited to military games, government games, educational games, corporate games and healthcare games. The focus in this thesis will be healthcare games, which includes

rehabilitation, pain distraction, physical fitness and more. This is also the area that will probably grow the most in the coming years.

## 2.8 Previous work

There exist many eye training methods, and some of them are computer based. Many physicians still rely on manual methods, but there are some effective digital solutions in the market as well. Due to availability, this thesis will focus on a couple of good but dated solutions.

### 2.8.1 COGPACK

One existing solution relevant for this thesis is called COGPACK and features several exercises to train different parts of the oculomotor abilities and other cognitive abilities. COGPACK is a Windows Application made in 1986 by Klaus Marker that provides 64 different tasks for training of visual attention, saccades, eye-hand coordination, visual memory and many other cognitive functions (Marker, 2018). These tasks are considered motivating and fun, because most of them are presented as games where you can improve your score at every run. There is also feedback like sounds or images to have you keep going.

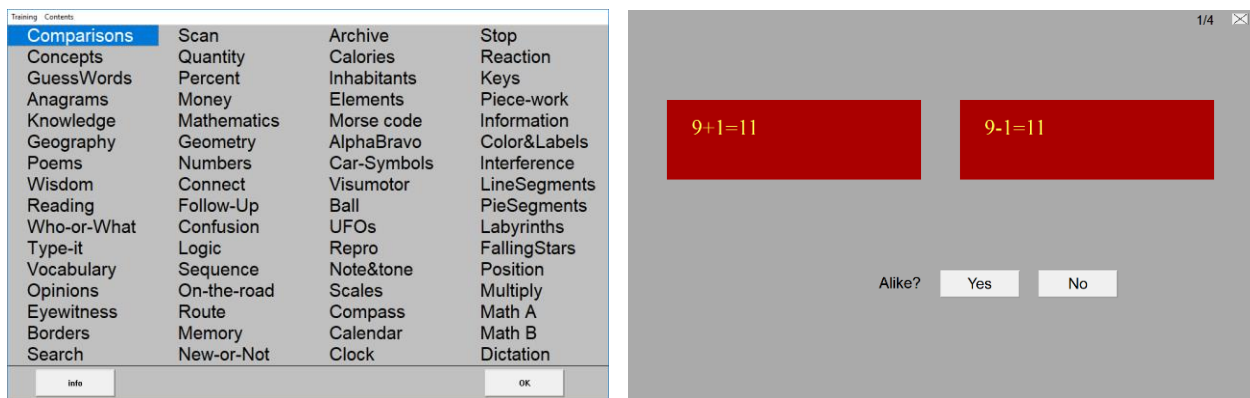


Figure 11 COGPACK by Marker Software

COGPACK is used by Gunvor Wilhelmsen and her team in the training processes. The usual approach is to closely monitor the user while he/she is using the program on a laptop or desktop computer. One issue with this is that it is hard to see exactly which object the user is focusing on, and the progress is therefore harder to measure.

COGPACK is sufficient for many use cases, but one problem with the solution is that it is becoming dated. Some of the exercises require non-relevant interaction with the computer, and these are making the application feel less intuitive than how it could have been. One example of this is input through the mouse, which sometimes require eye-hand-coordination that is not directly related to the task. COGPACK will be a source of inspiration for the solution made in this thesis, but most of the input methods will be replaced with eye tracking.

### 2.8.2 VisionBuilder

Another game used by vision teachers is one called VisionBuilder made by the Norwegian company Haraldseth Software located in Hamar. With this application, the user can train saccades, recognition, word reading, binocular reading and much more. The binocular reading is possible because of included red/blue glasses that the company ships to you. According to The Optometric Extension Program Foundation (2018), VisionBuilder is available through qualified optometrists and is a central part of an optometric vision therapy program.

VisionBuilder is not free of charge and must be purchased from their website for use.



## 3 Problem analysis

This chapter will contain analysis of the background information. The focus will be on connecting theories about oculomotor training to possible technical representations.

### 3.1 What the Training Should Cover

Initially, Gunvor Wilhelmsen provided some guidelines for the project. These will be used as pinpoints in the development process.

1. The application must cover training in all the following areas:
  - a. Saccades (left-right, up-down and diagonal)
  - b. Visual attention
  - c. Smooth pursuit
  - d. Visual memory
  - e. Eye-hand coordination
2. The experience must be motivating as well as educational
3. The contrasts and colors must be sufficient for people with reduced vision
4. The solution must be easy to use, both for educator and participant
5. The equipment must be easily portable

Tina Ihler who provided the table in chapter 2.2, also provided a list of important general considerations to take when making a computer application for training:

- Contrasts
- Colorblindness
- Possibility to control object speed
- Background color
- Brightness
- Screen resolution

## 3.2 Screen Representation

It is important that the theories from chapter 3.1 is converted into appliance on a computer screen. This sub-chapter will have a look at how the different visual abilities can be trained.

### 3.2.1 Saccades

Since saccades are performed while moving the gaze from one object to another, training these does not need to be very advanced. It could be enough training to force the user to look for, and at, different objects appearing on the screen. The general idea here is to place objects with a certain distance between them, to adjust how big the saccades are supposed to be. For micro saccades, a reading exercise could be sufficient. It is also important to not only have horizontal saccades. Vertical and diagonal saccades are also important (Wilhelmsen, personal communication, 2018).

### 3.2.2 Visual Attention

Visual attention is another part that has many possible training methods. One specific focus field inside the field of visual attention, is visual attention in the peripheral vision. This could be represented as objects appearing in the edges of the screen, so the user must use the peripheral vision to detect the objects. Another way of training visual attention could be to present the user for a specific screen for a defined amount of time, and then asking questions about what he/she saw.

### 3.2.3 Smooth Pursuit

Since smooth pursuit is the simple act of following a moving object, this is a good way of training it. The training could consist of an object moving at constant speed across the screen, for the user to follow with the eyes.

### 3.2.4 Visual Memory

Visual memory is very much tied to visual attention, so it can be trained in much of the same ways. The difference with games for visual memory is that the remembering part must be crucial for the game.

### 3.2.5 Eye-hand Coordination

To train eye-hand coordination, there is a need for an object on the screen which the user should point at, either using the mouse or a touch screen. The touch screen would be more realistic because of the user pointing at the object with his/her hands instead of pointing at it through an unnatural input method. Since a touch screen is not available for this project, using a mouse is the only method that can be used. To train eye-hand coordination the user should be forced to move the mouse at certain objects, preferably with a time limit.

## 3.3 Development Choices

To cover all the areas mentioned in 3.2, the games must provide the ability to make moving graphics and a defined number of objects. The objects must be able to interact with each other and must be able to be controlled using eye tracking. It must also be possible for the user to choose settings like object speed, colors or contrasts.

The game must be able to use different input methods like mouse and keyboard, preferably combined with the eye input.

## 3.4 Choosing Framework

In a former eye tracking thesis on Western Norway University of Applied Sciences by Eide & Watanabe (2017), the framework Windows Presentation Foundation (WPF) was used. This framework was reportedly complicated for making moving graphics, so alternatives were considered for this thesis. The application should be able to run on a Windows desktop computer, so the framework needs to suit this requirement. It must also have the possibility to create different kinds of 2D graphics in a not too complicated manner.

### 3.4.1 Windows Presentation Foundation

Windows presentation foundation was made by Microsoft in 2006 as a framework for making graphic user interfaces. It is built upon the .NET framework and is therefore available across most of the recent versions of Windows.

WPF features the following:

- Graphical interface
- On-screen documents
- Images
- Video and audio
- 2D graphics
- 3D graphics

Before the introduction of WPF, several different technologies were required to get all these features as a native windows application. WPF functions as a unifier for these.

	Windows Forms	PDF	Windows Forms/GDI+	Windows Media Player	Direct3D	WPF
<b>Graphical interface, e.g., forms and controls</b>	X					X
<b>On-screen documents</b>	X					X
<b>Fixed-format documents</b>		X				X
<b>Images</b>			X			X
<b>Video and audio</b>				X		X
<b>Two-dimensional graphics</b>			X			X
<b>Three-dimensional graphics</b>					X	X

Figure 12 Technologies replaced by WPF. Source: <https://msdn.microsoft.com/en-us/library/aa663364.aspx>

Using lots of different technologies to do simple things may contribute to compatibility issues thus making the development much more time consuming. This is at the core of what WPF aims to fix.

One thing that makes WPF powerful is the ability to have a backend developer and a designer work together through a markup called XAML. This means that the backend developer can make functioning display objects, and then the designer can make the front-end appearance afterwards.

The framework utilizes the C# language, which is a well-established and powerful high-level language. (Chappell, 2006)

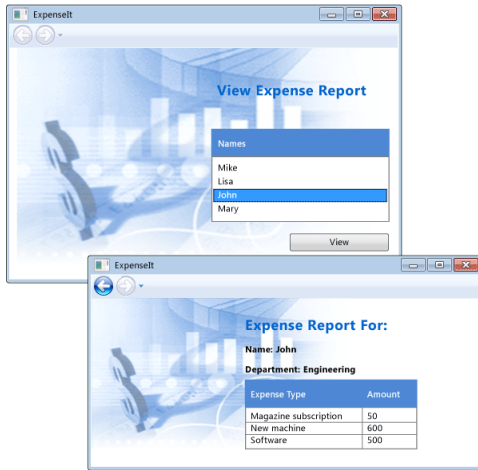


Figure 13 Sample WPF application. Source: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/getting-started/walkthrough-my-first-wpf-desktop-application>

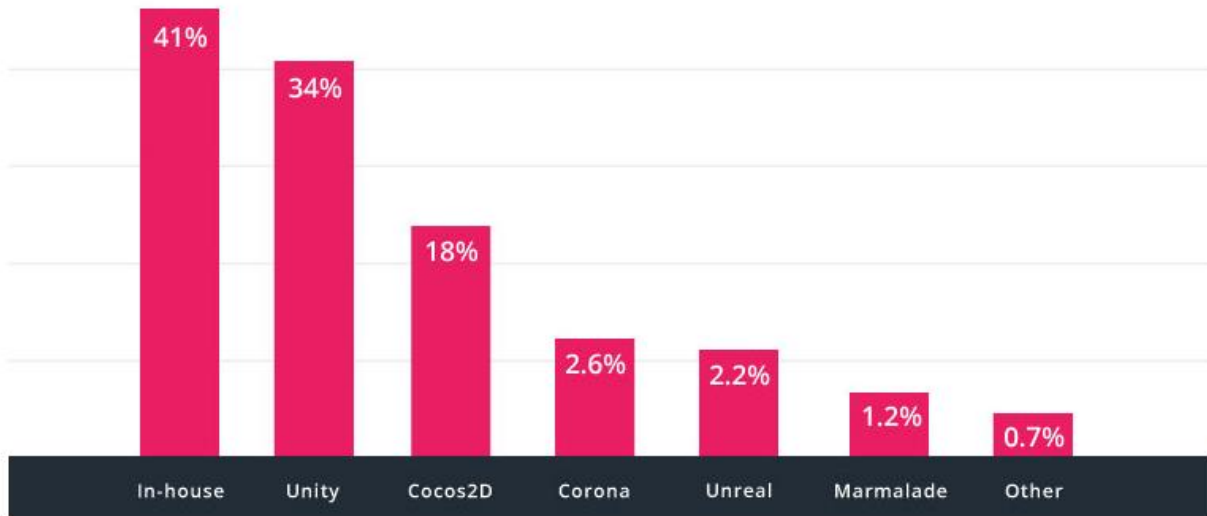
### 3.4.2 Unity

Unity is a closed-source framework made by Unity Technologies mainly used to make 2D and 3D games. It features a Graphical User Interface for simple drag-and-drop of elements which can be accompanied by scripts in C# or UnityScript. Unity also features out-of-the-box physics, which makes it easy to make 2D or 3D objects act more naturally in its environment (Unity, 2017).

Unity applications can be deployed to several platforms like iOS, Android, PlayStation 4, Gear VR and a lot more. This means that an application can be build cross platform without having to rely on multiple framework and technologies.

Unity is widely adopted, and over 770 million people are monthly playing games developed with the framework. This includes desktop, mobile and VR games (Unity, 2016). The wide adoption leads to strong communities where finding answers to development problems is an easy task.

According to SourceDNA, 34% of the top 1000 mobile games in the world were made with Unity, making it the most used game development framework. The only thing that outruns Unity according to the diagram below is in-house development frameworks that companies design specifically for their projects.



Source: SourceDNA, Q1 2016

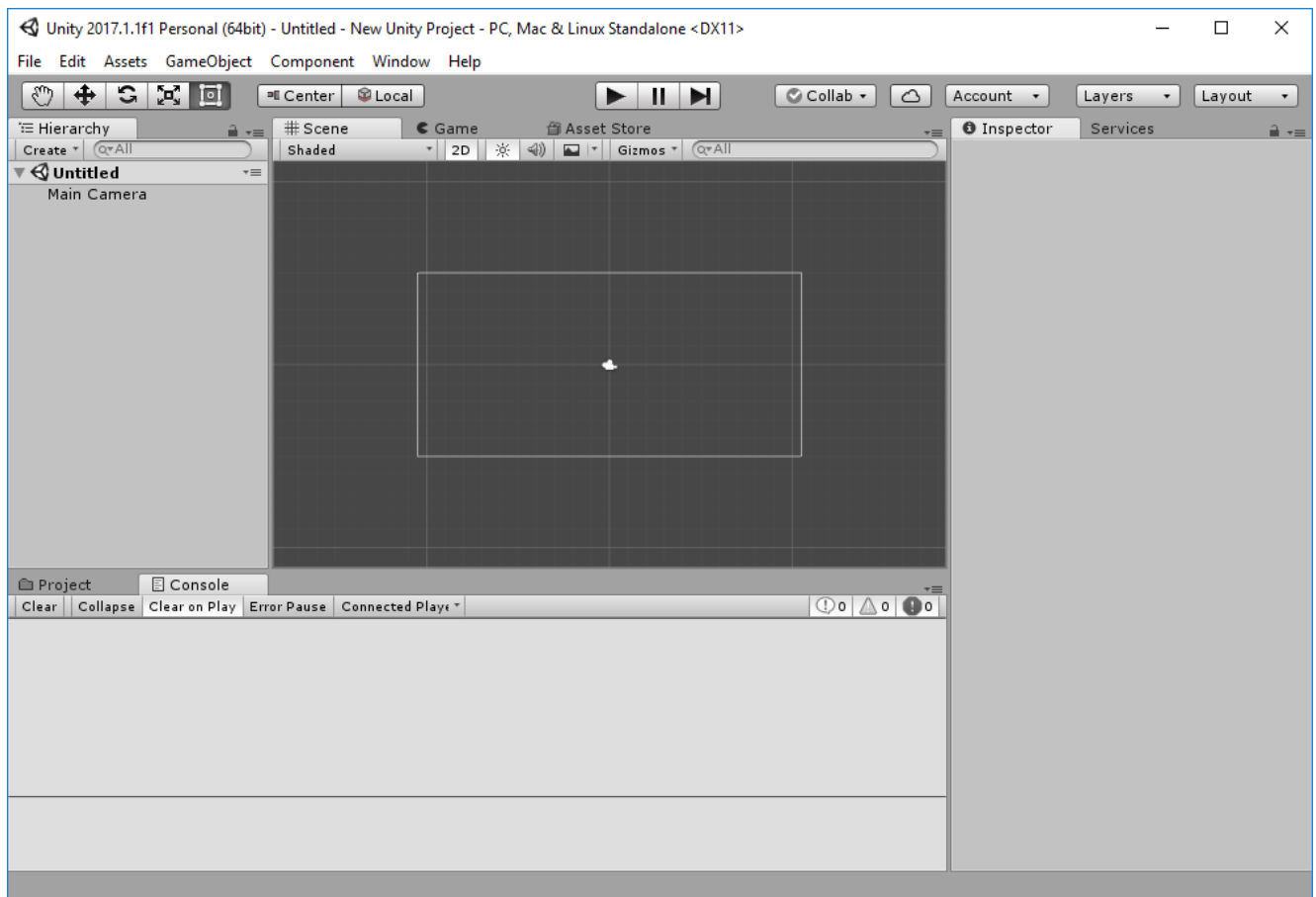


Figure 14 Empty Unity project

### 3.4.3 Cocos2D

Another framework that is relevant in 2D game development is Cocos2D. It is an open source multi-platform framework that uses C++, JavaScript and Lua. Cocos2D provides most of the necessary game engine functionality like rendering, 2D/3D graphics, collision detection, physics, animations and more. This is much like the Unity engine. (Cocos2D, 2018, a)

Cocos is open source, which means that the possibilities are broader than the closed source ones, if one bothers to dive into the code and make the necessary changes. This also means that the development process may be more advanced and lower level than the alternatives.

A big difference between Unity and Cocos2D is that the latter is not directly tied to a development application, so Cocos2D application may be developed in any preferred text editor. To get more functionalities of a graphics development application, Cocos features Cocos Creator which looks very similar to the Unity interface. This creator features both a design-oriented workflow and a develop-oriented workflow, making it easy even for those without much programming experience. (Cocos2D, 2018, b)



Figure 15 Cocos Creator interface. Source: <http://www.cocos2d-x.org/creator>

### 3.4.4 Discussion

During the initial project planning, choosing the right framework was crucial. The framework will function as the main core for the project and shapes the general workflow. The WPF framework was considered at an early stage, but after some research this framework was not found to be the best choice. The main reason for this is that WPF is more of a GUI engine than a game engine, which makes it lack many important features that a game engine requires. WPF lacks physics and is much more low level, which means it requires much more work to make the necessary animations.

Cocos2D was also considered for this project. The problem with Cocos2D was the lack of documentation about Tobii integration and the fact that it had a less graphical workflow making the development process slower. Tobii do provide a low-level C++ SDK, but this was considered more advanced than necessary for this project.

In the course MOD253 Advanced Computer Graphics at HVL, the Unity Framework was used for development of different graphics applications. This framework showed itself very useful for a variety of graphical tasks and had some good workflows available. Unity was considered appropriate also for this project and was chosen to be used for the prototype development. The main reason Unity was chosen is a combination of it being rich on features and providing an easy integration with the Tobii trackers. This makes the development of the prototype fast, with a lower chance of getting stuck along the way than what would be the case with more low-level approaches. For this project, there was only a need for the 2D part of Unity, eliminating the complex polygon meshes required in 3D.

Unity applications typically compile Windows applications to .exe-files with required files packed in a folder, which makes it easy to distribute for testing on different Windows desktop devices. This is considered good enough since the application is only going to be run on Windows desktops and laptops.

Since the project features an eye tracker from Tobii, the framework must be supported by the drivers of this tracker. Tobii provides some of their API's in native C#, so this should be doable in Unity.



## 3.5 API

### 3.5.1 Tobii Unity API

The first API that was tried was the Tobii Unity API. This API has a satisfying integration with Unity and can for instance provide information about which object the user was focusing on, or just a gaze point with x and y coordinates. Additionally, the API can determine head pose (if using an eye tracker with this functionality) and if the user is present.

The Unity API may be directly added to the environment and can be run from the Unity engine. This makes the integration less time consuming than other available methods.

The problem with the Unity API for this project, is the lack of single eye tracking. Tobii has made this API only able to track both eyes simultaneously, which should be sufficient in most gaming situations. And since Unity is a game engine, this excluding of single eye tracking may make the development easier for most game developers.

### 3.5.2 Gaze API

Tobii also provides lower level APIs for their trackers. Since Unity uses C#, a low-level API in the same language should be possible to use. A more fitting API would then be the Gaze API which features implementations in both C++ and C#.

The Gaze API has much more features than the Unity API. The most important part is the ability to check which eye is active, and which is not active. To do this, the API contains what they call “Tracking Status”, which has the following variables:

- BothEyesTracked: true when data from both eyes are captured by the eye tracker
- OnlyLeftEyeTracked: true when only the left eye is captured
- OnlyRightEyeTracked: true when only the right eye is captured
- OneEyeTrackedProbablyLeft: true when the program thinks the left eye is the only one being tracked, but is not completely sure
- OneEyeTrackedProbablyRight: true when the program thinks the right eye is the only one being tracked, but is not completely sure

This gives the ability to make behavior in the games based on which eyes the user is currently using.

Also, from the API, the user may get information about the mean eyes' x and y coordinates, the left eye's x and y coordinates or the right eye's x and y coordinates. For this reason, the Gaze API is the proper choice for this project.

One issue with using a native plugin that is not directly supported in the framework is that it needs to run in a separate thread. This means that every time the eye tracker is to be used, there is a need to create a new thread and assign it to the API. It is also important to close the thread afterwards, or the application will crash.

Another issue that was encountered while using the Gaze API in Unity was that the coordinates for the eye tracker does not necessarily match the coordinates of the game screen. This means that a screen calibration is necessary when changing the size of the game window. This could be done in a satisfying way by using the window bounds in Unity where both the x coordinates and the y coordinates are multiplied with the corresponding screen size.

### 3.5.3 Logging

The initial plan was that the game plays should be logged for later review. Unfortunately, because of a limited Tobii license, this will not be possible for this thesis.

## 4 Solution

This chapter will contain information about a solution to the problem analyzed in the previous chapter. It will be presented as a general solution, followed by information details about every aspect of the application.

### 4.1 General

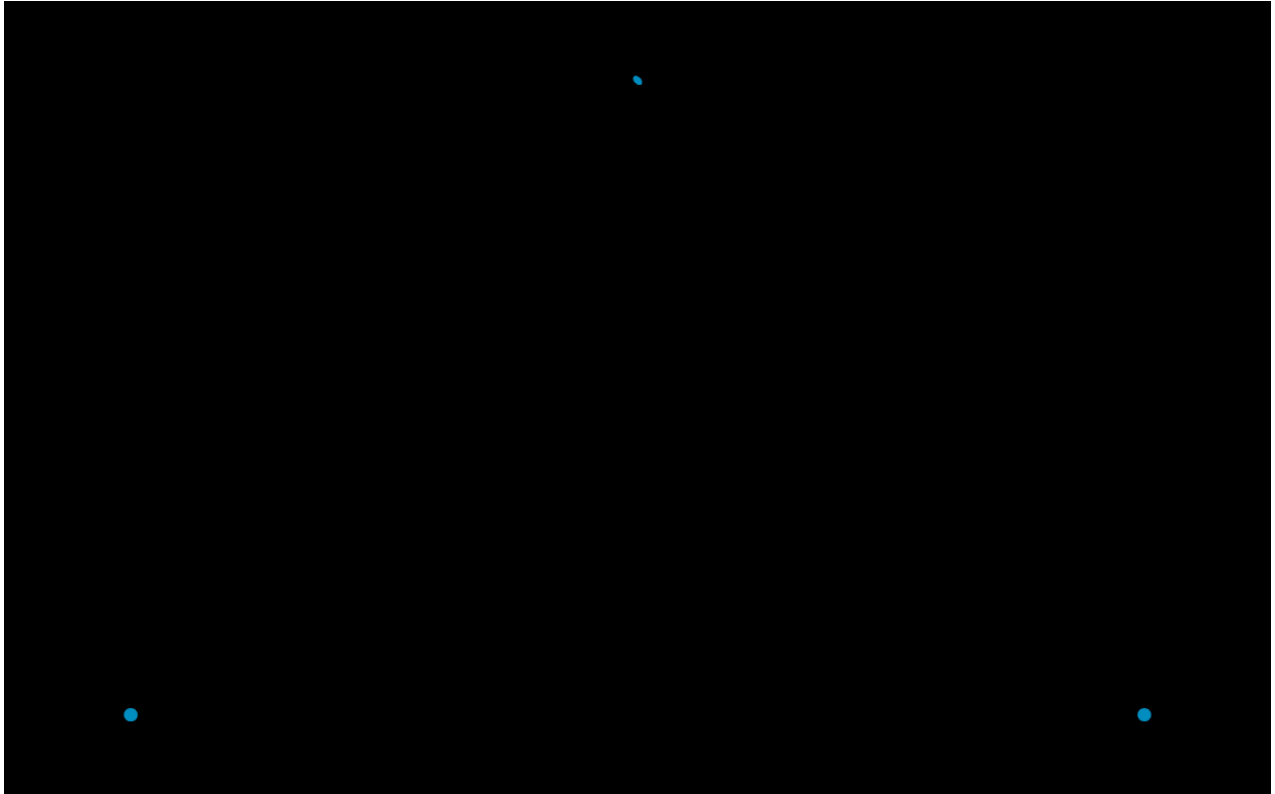
To make a complete training application, there is a need for several smaller applications aimed at training different eye movements. The application must cover training for saccades (left-right, up-down and diagonal), attention in the peripheral vision, smooth pursuit, visual memory and a little bit of eye-hand coordination. Each game may cover more than one of these, and in most cases, this is preferred.

The application consists of 6 games, each having a start and an end. Some of the games runs on time, but others run until completed or canceled. Points will be counted but not stored in this prototype. Some of the games will provide the opportunity to change the difficulty, so the user can reach harder levels when the easiest levels are becoming too easy. In a complete solution, this should be present in all the games.

### 4.2 Calibration

Before every usage, the eye tracker needs to be calibrated to fit the user's head and eyes. This calibration can be implemented into the application through the Gaze API, but that is not considered necessary for this prototype. Instead the calibration is just run from the Windows application which functions as a general cross-application calibration. The drawback with this is that the calibration data cannot be received from the application, but that would only be a problem when high precision is required.

The application is designed to call the calibration application on a button click. This could be done any time whenever user wants. The rule of thumb is that this should be done every time there is a change of user.



*Figure 16 Tobii Calibration application*

To perform a calibration, the application shows some points which the user should look at. When the user looks at the point for 2 seconds, the point disappears, and calibration is done for that specific area. When all the points have been eliminated the calibration is finished. The user is then sent back into the game menu.

During a calibration, the eye tracker captures images of the user's eyes to find specific properties such as reflection properties, placement of the fovea etc. This data is connected to an anatomical 3D eye model. It then makes the most suitable settings for the pair of eyes present and saves it in the current profile. This is to get as exact gaze position possible. (Tobii, 2015, d)

### 4.3 Game 1 – Compare Words

In the first game the participant should look at two alike words and find if these words are identical or different. Only one or two letters should be changed in one of the words to keep the differences at a minimum. When the user has reached a conclusion, he/she must click on the respective button to register their answer. If the answer was right, one point is given. If it was wrong, no points are given. The background is white, and the text is black to keep a sufficient color contrast, so it is universally available. A screenshot of the game can be seen in Figure 17.

To find the differences the user must perform saccades between the words and smaller saccades while studying the single words. This will train both reading, big saccades and the brain's ability to identify the content of words. When the conclusion is reached the game requires a bit of eye-hand coordination to register the answer, but this is only an indirect eye-hand coordination. For a more direct usage of eye-hand, a touch screen should have been used for answer registering. The game is played on time, and the player will have X seconds to go through as many words as he/she can.

Before the game starts, the user can set the difficulty level, which will decide on the length of the presented words. The difficulty can be set from 1 to 4, and the length of the words will be  $1 + \text{difficulty}$ . This means that the words' length is between 2 and 5 characters

This game is similar to one of the games featured in COGPACK which was introduced earlier in the thesis. Since there is no eye tracking in use here, the game is mostly just needed to make a complete training application and does not provide any notable innovation to the area of oculomotor training.

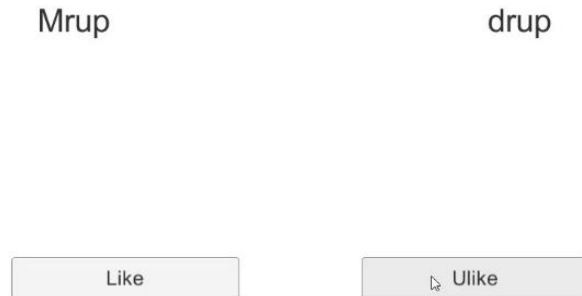


Figure 17 Game 1 - Compare Words

## 4.4 Game 2 - Maze

The second game has a start point and an endpoint colored respectively green and red. The goal is for the user to enter the green point, then navigate to the red point and back without hitting the walls. Every time the user enters the green point after he/she has been visiting the red point, one game point is given. If the user fails to avoid the walls, all points are lost. In this game the player consists of two circles, one for each eye (black right, blue left), and both eyes count when checking for collisions. If an educator is present, he/she can spot the difference in gaze position between the right and the left eye while the user is playing.

This game trains the ability to follow a path with the eyes and keep the attention on the road while avoiding obstacles. Figure 18 shows a screenshot of the game.



*Figure 18 Game 2 – Maze*

## 4.5 Game 3 – Save the Rocket

This game has a more complete graphic user interface (see Figure 19). In the center of the screen there is a rocket which the user should protect. Meteors will emerge from the corners of the screen, which the user must eliminate before they are hitting the rocket. Each meteor is destroyed instantly upon gaze focus. If the user manages to prevent the rocket from getting hit until time runs out, the game is completed. If any meteor hits the rocket, the user has failed. Every time a meteor is hit, a small explosion sound is played to indicate that the meteor was eliminated. The supervisor can set the difficulty for each round, in this case how often meteors occur and how fast they travel towards the rocket.

This game is a suitable training of visual attention and continuous performance of saccades. Since the meteors are coming from random places, the user must move his/her gaze around the screen to detect the meteors. This may be a good training for people who need to compensate the lack of peripheral vision by using their high acuity vision, like the ones with visual neglect.

Before the game starts, the user is presented with a settings screen where he or she can set the frequency of meteor spawning and the speed the meteors have towards the middle of the screen.

The speed can be set from 1 to 10, which is multiplied with delta time in the implementation. This makes the easiest level easy enough and the hard level hard enough. The frequency may also be set from 1 to 10, where frequency of 1 means 10 seconds between spawns and frequency of 10 means 1 second between spawns. Every stage of the frequency setting is 1 second.

The gaze is represented as a bubble on the screen shots.

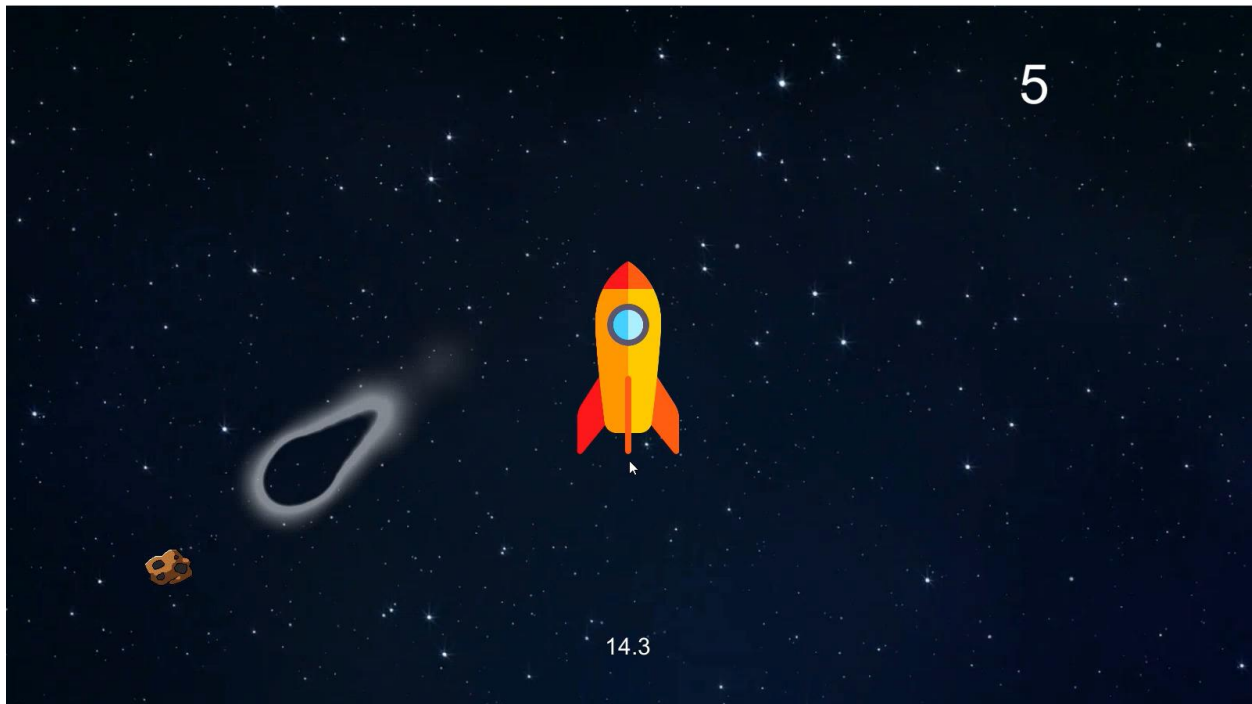


Figure 19 Game 3 - Save the rocket



## 4.6 Game 4 – Find the Outstanding Object

The fourth game is of slower pace. This game consists of several objects on the screen, with one object different from the others (see Figure 20). The goal is to find the different object and keep the gaze pointed at it for a couple of seconds. The application then detects that the user found the object of interest and the level is passed. This is much like the game Where's Waldo (Good Reads, 2018) just with a special input method.

In this game it is possible to change the level of difficulty by changing the amount of distraction objects are present. It is also possible to choose between men/woman and dog/cats. The game counts how many second the user took to find the object of interest, so no time limit is set. This is because the game is not supposed to be stressful, but rather something that will take a bit of time.

Like Game 3, this game also trains visual attention. The difference is that this is a much less stressful environment and gives the user more time to think and process all the present objects. The user must continuously perform saccades to move between objects and micro saccades when analyzing a specific one. In addition, the user must utilize the peripheral vision to find interesting areas to look in.



Figure 20 Game 4 - Find the outstanding object

## 4.7 Game 5 – Follow the Ball

The fifth game is about following an object that moves across the screen (see Figure 21). When the user's gaze is pointed at the object, the points are counting, and when the gaze is elsewhere the points does not count. Every X seconds a distraction appears a random place on the screen, and the user should avoid this and keep looking at the moving object. The points are an indication of how much of the time the user managed to look at the ball, and how he/she managed to avoid the distractions. This game runs until a defined number of points is gained.

This game trains the smooth pursuit movement and the ability to keep focus on the important object even though distractions appear in the visual field.

227



Figure 21 Game 5 - Follow the ball

## 4.8 Game 6 – Which One is Missing?

The last game is a memory game where the user gets to look at some objects and must pick the object that was not present. The user gets to look at X-1 objects for Y seconds and is then presented with X objects. Then he/she must find out which object was not present in the previous screen and point the gaze towards this object. When the gaze is at the right spot, the user must press Space to “pick up” the object and then move the eyes to a square present on the screen. (see Figure 22 and Figure 23) When the object is inside the square, the space button must be pressed again to release the object. If this object was the correct one, the level is completed, and a message and sound gives the user suitable feedback.

This game trains visual memory, visual attention and saccades. When the user is looking at the x objects, the eyes are constantly performing saccades, only this time the visual input must be remembered for later. Afterwards the user must perform saccades to find the right object and put it into the box.

Scene 1:

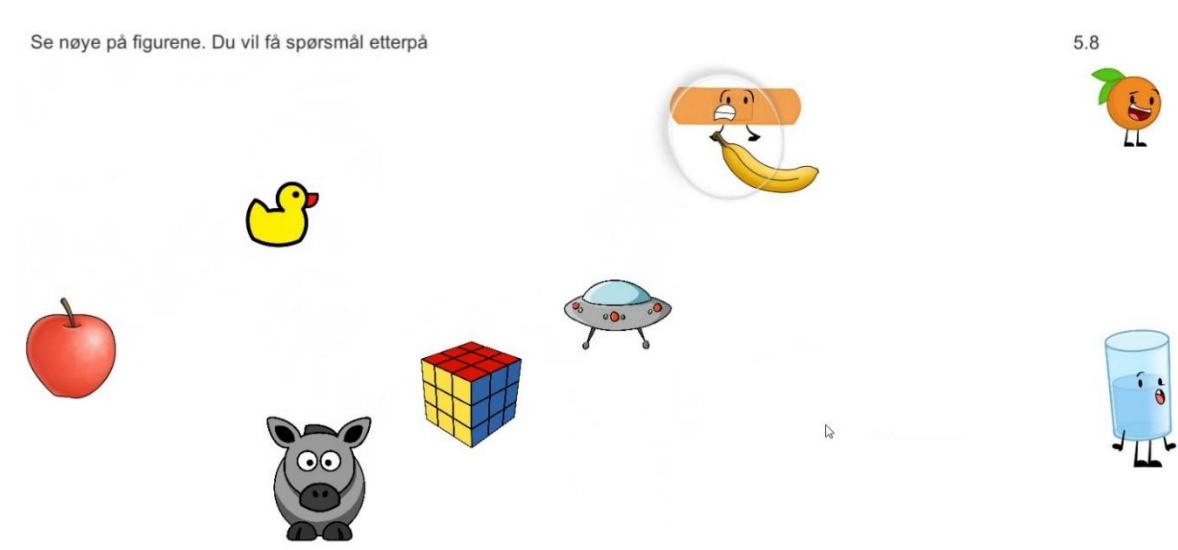


Figure 22 Game 6 - Which one is missing (1st screen)

## Scene 2:

Hvilken figur manglet på forrige bilde? Se på figuren, trykk <space> og flytt synet opp til boksen. Trykk <space> igjen for å slippe.

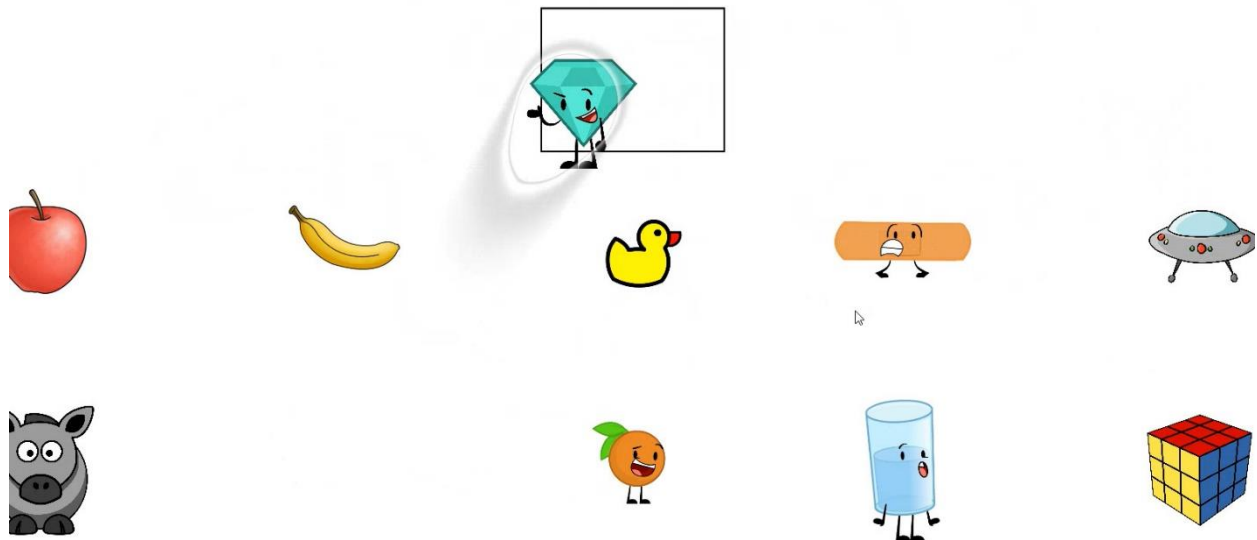


Figure 23 Game 6 - Which one is missing (2nd screen)

Some of these games are using gaze-based interaction and some are using gaze-added interaction. Most of the games do not have any action button, because the action is solely based on gaze focus. The only game where this was an important choice was game 6, where objects are being picked up using the vision. Because there are multiple objects that can be picked up, the user must indicate that a certain object should be chosen. One way of doing this is through blinking, but this is not considered intuitive for people who already struggle with oculomotor issues. Therefore, the spacebar is used to pick up and release items.

If all these games are practiced consistently, the training will cover most of the important aspects discussed in chapter 4.1. This means that if the application is used over a given period, the user could see some improvements in oculomotor abilities and could experience less tiredness than before starting with the program.

## 4.9 Fitting the Games into the Exercise Table

Table 4 contains information provided by Tina Ihler connected with the games for a better overview of what the different games cover.

Game number	Game name
1	Compare words
2	Maze
3	Save the rocket
4	Find the outstanding object
5	Follow the ball
6	Which one is missing?

Table 3 Overview of the games in connection with their number

Challenge	Terms	Exercises	Game number
<b>Field of view</b>	Saccades	Horizontal movements	1, 2, 3, 4, 6
	Visual attention	Vertical movements	2, 3, 4, 6
	Regression	Diagonal movements	2, 3, 4, 6
		Circular movements	3, 4
<b>Visual acuity</b>	Fixations	Searching/scanning	4, 6
	Endurance	Find objects in a crowd	4
	Saccades	Horizontal movements	1, 2, 3, 4, 6
	Mini saccades	Vertical movements	2, 3, 4, 6
		Diagonal movements	2, 3, 4, 6
		Circular movements	3, 4
		Smooth pursuit	5
		Find pairs/similarities	1
	Labyrinths point to point	2	
<b>Stereopsis</b>	Accommodations	Movements and flashes at a distance – different depths	None
	Convergence		
	Double vision	Objects that varies in size	

<b>Eye-hand coordination</b>		Use the mouse or keyboard based on events on the screen, and vice versa.	1 (partly)
------------------------------	--	--	------------

*Table 4 Training of different oculomotor challenges. Source: Private conversation with Tina Ihler*

Since “Field of view” and “Visual acuity” are very broad fields, different games need to complement each other to reach through all aspects. Stereopsis was not a priority in this game, because it is hard to simulate depth vision on a computer screen.

## 4.10 Architecture

Since the application consists of several small games, there is essentially one application for each game connected through scene management in Unity. This is a concept that allows the application to load scenes based on conditions and may be used between all the scenes. Every scene has an object called a “controller”, which acts as the general game controlling object. This controller is responsible for spawning and deleting objects, moving objects around the screen and controlling the general game logic. The only thing the controller cannot do for this project is detect object collisions, so that needs to happen inside the colliding objects.

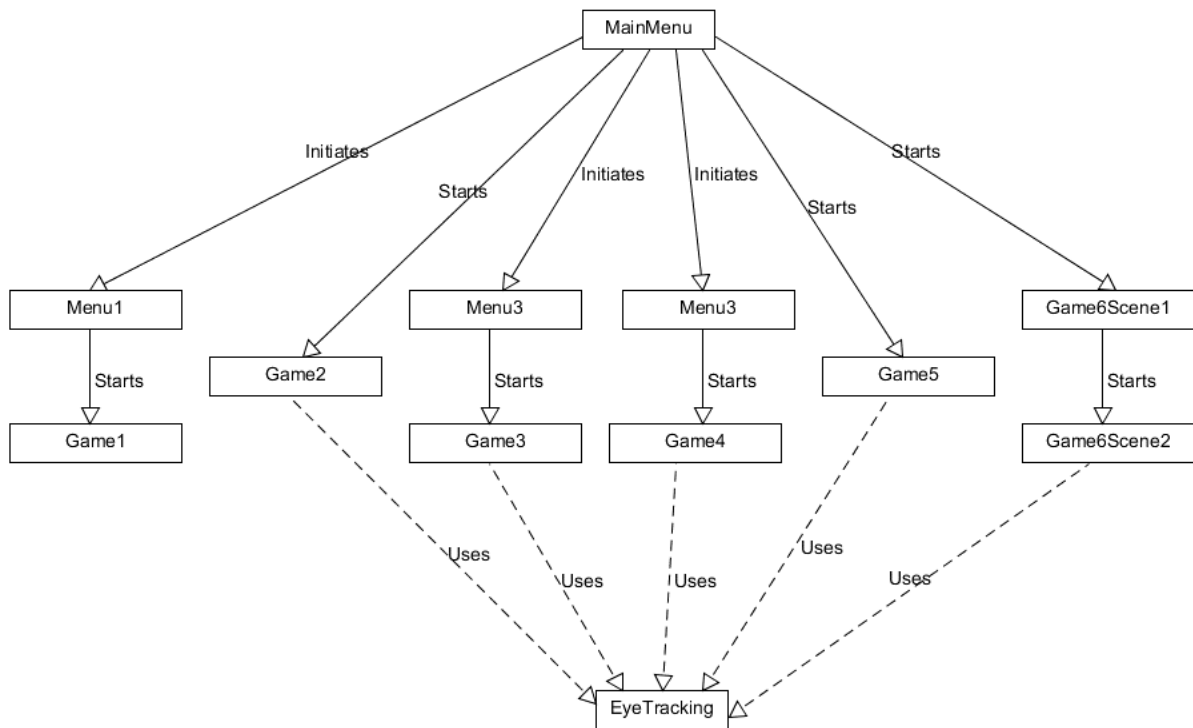


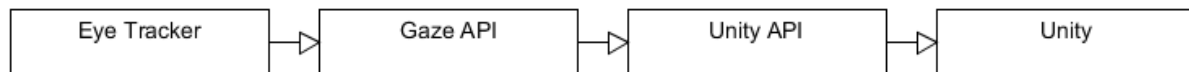
Figure 24 UML diagram of the application

All the menu objects shown in Figure 24 are made with generally the same code. The only difference is that the scenes have a different number of buttons and sliders, so that must be set for each of the objects. This means that making a new game menu is very easy. They all contain objects with a specified movement property, which can be personalized to fit the game. This makes the application very expandable using already existing code.

Every scene in Unity has their own script file, and every script file has a method Start() and a method Update(). The Start() method defines what will happen when the game object is loaded the first time, so spawning of objects and initialization of helper scripts is suitable actions for this method. The Update() method is called on every new frame, which typically occurs 60 times per second. This is a suitable place to move objects to certain coordinates, check for overlap, and control the behavior of the eye tracker. The Update() method can also be used to count points on certain conditions, and can give the application the ability to track time passed.

To integrate Unity with the eye tracker, there is a need for a helper class described as “Gaze API” shown in Figure 25. This helper class is the C# version of the Tobii Gaze API which

integrates with the EyeX tracker. Since this class is running in a separate thread, it is important that the service is connected and disconnected at proper times, to avoid crashing the application. For instance, if the eye tracker service is disconnected and the method `GetMeanEyesX()` are called, the application will throw a null pointer which will result in a crash. For this reason, the tracker's `connect()` method is called on `Start()` and the `disconnect()` method is called from the `OnDestroy()`-method which Unity calls when a game object is destroyed.



*Figure 25 Eye Tracker integration*

Most of the eye tracker logic happens in the `Update()` function of each game controller. This is mostly checking if the coordinates of the eye gaze are inside the coordinates of an object or setting an object's coordinates to where the eye gaze currently is. The helper class feeds the application with a stream of coordinates for where the user's eyes are positioned at every frame, which allows for a very high precision. This precision may make the objects seem a little choppy but evening out the motion through interpolation or similar method are not considered necessary.

In some of the games, there is a need to check for collision between objects to trigger certain events. This check needs to be done in the involved objects through a method called `OnCollisionEnter2D()`, which is called every time the object collides with another object. The method contains information about the collision, like the name of the colliding object. This information can then be received by the game controller to do actions based on the collision event. That includes checking if the colliding objects are the objects of interest, and for example give or take game points.



## 5 Implementation

This chapter will cover some of the details about the implementation of the application. This includes some general concepts and some code snippets.

### 5.1 Initial

The first thing that needed to be done for the implementation was to set up the Unity framework and create a new project. Since Unity is the only necessary framework for this prototype, the setup process was not very time consuming. The Gaze API was downloaded from the Tobii website and imported into the project. To make the application code hierarchy properly structured, the best solution was to make a new folder for each game, named “GameX”, and a menu folder called “MainMenu”. All the sprites were put in a folder called “Graphics”, initially some standard sprites like a white background for all the games.

### 5.2 Game Objects

All the scenes consist of pre-defined objects to appear. These are standard game objects using sprites for visual appearance. Some of the game objects have a pre-defined start position through the Unity engine, while others are created at runtime by the scripts. To create a game object from script, one must first create a pre-defined object in Unity, called a prefab. Then this must be attached to the script for creation. One way of doing this is through the Instantiate()-method featured in the MonoBehaviour class:

```
GameObject obj = Instantiate(asteroid, getPosition(), Quaternion.identity) as GameObject;
```

This method instantiates the asteroid objects in the rocket game as a GameObject. A reference to this object is saved to the “obj” object for further handling. Through this reference, it is possible to change the position of the object, destroy it, among other things.

## 5.3 Gaze API

The Gaze API is the source of the eye tracking data. This API gets the data directly from the tracker and can provide gaze data as x and y coordinates. The class contains the following relevant functions:

- Connect(): Connects the eye tracker
- getMeanEyesX(): Gets the x-coordinates of the mean gaze produced by both eyes
- getMeanEyesY(): Gets the y-coordinates of the mean gaze produced by both eyes
- getLeftEyeX(): Gets the x-coordinates of the left eye gaze
- getLeftEyeY(): Gets the y-coordinates of the left eye gaze
- getRightEyeX(): Gets the x-coordinates of the right eye gaze
- getRightEyeY(): Gets the y-coordinates of the right eye gaze
- Disconnect(): Disconnects the eye tracker

For most of the games, the mean eyes-method are sufficient to provide good enough functionality. In game 2 however, there is a need for single eye tracking to make observation easier for the supervisor.

To import this information into the application, the following code is run:

```
eyePos = (new Vector3(eyeTracking.getMeanEyesX() * Screen.width,  
eyeTracking.getMeanEyesY() * Screen.height, 1));  
eyePosWorld = cam.ScreenToWorldPoint(eyePos);
```

This code gets the x and y coordinates of the mean eye position and creates a vector containing these values. To scale according to the Unity window, the x coordinates must be multiplied with the screen width while the y coordinates must be multiplied by the screen height. This vector is then converted into world point to fit the game window.

The data from the eye tracker can be used to either set the position of any object or check if the gaze area collides with any objects of interest.

## 5.4 Checking for Hits

In Tobii's Unity API, there is pre-defined methods that lets you check if the eye gaze is hitting an object and get information about this hit. But since this prototype utilizes the Gaze API, the collision checking must be done manually. To do this, a method called "IsInside()" has to be created. This method takes in two coordinates, and check if these coordinates are close to each other.

```
public bool IsInside(float x, float y)
{
    return Math.Abs(x - y) < 0.75;
}
```

The method uses the `Math.Abs()` which finds the absolute value of the number. This means that if the number is negative, it will become the same value just positive. The numbers are then checked for differences, and if the difference is less than `x`, the object is considered "inside" the other object. This must be done for both `x` and `y` coordinates, so the call to this method is like this:

```
if (IsInside(eyePosWorld.x, obj.transform.position.x) && IsInside(eyePosWorld.y,
obj.transform.position.y))
{
    doSomething();
}
```

If this condition is met, the application knows that the eyes are looking at the specific object, and the application may act accordingly.

## 5.5 Keeping Track of Time

In most of the games, time plays a central role. This is so the game should not run indefinitely and gives the user able to progress to harder levels. The time is tracked through a double variable which is updated inside the Update() method. If time should count upwards, the following code is run on the Update() method:

```
timeLeft += Time.deltaTime;
```

The deltaTime property of Time returns the time it took to complete the last frame. This means that the time variable always will know how much time has passed if this is calculated for every frame. Then the calculation will be:

Seconds passed = seconds per frame \* number of frames

The standard number of frames per second will be around 60, but this method works even without the user knowing the technical details.

For each frame, the time variable is written to a Text game object to make a visual timer. The variable is first rounded up to 1 decimal using the Math.Round() method.

## 5.6 Scene Management

Unity applications are built around scenes, that contains objects in a separated environment. The scenes are managed through the integrated SceneManager which exists in the UnityEngine class. In this application, there must be one scene for every game, and one scene for every menu.

The first scene is the main menu. This is where all the other scenes should be started from. Since there are only buttons to start the scenes, no information should be sent form the main menu to the games. To open a scene from the main menu, the following code is run:

```
SceneManager.LoadScene(X);
```

Where X is the number of the scene in the build. The main menu buttons point to the individual game's settings that should be set before the game starts.

In the game menus, some data is set and should be sent to the specific game that is about to start. To send information between scenes, one can use the class `PlayerPrefs`, which sets application-wide settings for the current game session. To do this, the following code is run:

```
PlayerPrefs.SetFloat("variable name", value);
```

This method adds the value to a variable name, and may be retrieved by other scenes by running the following code:

```
value = PlayerPrefs.GetFloat("variable name");
```

This example is using float, but the same thing can be done using integers, strings and so on. This makes an easy system for sending variables between scenes or to set player preferences in a single scene for later retrieval.

## 6 Evaluation

As mentioned in the sub-chapter about limitations in chapter 1, measurement of effect is not possible to get during this thesis. This is because it would take several weeks in an intensive program to be able to see changes, which was not possible during writing. Instead, there was performed testing with surveys (User Experience Questionnaire) on 5 vision teachers, and an in-depth evaluation by one of the project supervisors and vision teacher Gunvor Wilhelmsen together with vision teacher Heidi Jåstad.

### 6.1 Testing Method

The 5 vision teachers who tested the application and filled out a form, did so during their stay in Tanzania. They all tested games 1-5, with technical assistance from Mads Gjerstad Eide. The participants were supposed to answer general questions in the beginning, before evaluating every game by themselves. The general questions were:

- Your gender
- Age
- Experience with computer applications
- Your experience with training eye-problems
- What is your first impression of the games?
  - To use by yourself
  - To work (use for training)
  - The functionality
  - The design

These questions were giving an overview of who the participant is, their experience, and the first impression after trying the games. It is important to find out their immediate thoughts about where they think such application could be useful.

After these initial questions, the participants were asked about the impression of each single game. This was formulated as a list where the participants could write a single line about each game.

The third part of the survey asked the participant to rate different aspects of the game from 1 to 7 based on their experience. It was in total 25 aspects to rate, and the participant was only told to put a cross in the circle that represented their answer. The aspects to rate were the following:

	1	2	3	4	5	6	7	
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable
Dull	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Creative
difficult to learn	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy to learn
Inferior	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Valuable
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	exciting
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	interesting
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	predictable
fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow
inventive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	supportive
bad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	good
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasing
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	leading edge
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasant
Not secure	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	secure
demotivating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	motivating
does not meet expectations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	meets expectations
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	efficient
confusing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	clear
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	practical
cluttered	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	organized
unattractive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	attractive
unfriendly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	friendly

In the last part of the survey, the user was asked to give simple “yes” or “no”-answers to 7 questions about the system in general. The questions were designed to give straight answers to the usefulness of the service for easy interpretation.

- Is this technology easy to learn for instructors?
- Would you use it to complete other training?
- Do you think it would replace other training?
- Would you need IT help to use such technology in training?
- Easy to learn for learners (students)?
- What do you think about the size of the screen?
- The size/color/form of the representations?

## 6.2 Results

Below are the results from the survey completed by the 5 vision teachers. In this presentation, each participant has a number 1-5, and is first presented with age and experience.

<b>Number</b>	<b>Gender</b>	<b>Age</b>	<b>Experience with computer applications</b>
1	Female	46	Using computer applications every day, 1-2 times a week for training
2	Female	--	Using computer applications every day
3	Female	35	Does not use computer applications daily
4	Female	44	Using computer applications every day, but never for training
5	Female	55	Using computer applications 1-2 times a week, both for training and other

*Table 5 Survey participants*

The table below contains all the answers to all the questions in the survey. In the left column there is a question and each participant have their own column.



	<b>Person 1</b>	<b>Person 2</b>	<b>Person 3</b>	<b>Person 4</b>	<b>Person 5</b>
Your experience with training eye problems?	Daily	..	Only through regular studies	Regular	In studies and for training + at home
What is your first impression of the games? To use by yourself?	Is not self-instructing	..	The games are good, but needs instruction	Misses instructions during tasks	Yes, it is useful, but needs instructions
What is your first impression of the games? For work?	As a part of a training program, it could be good. But not alone.	OK	Needs some adjustments before it can be used for work	Can be, some trouble with low contrasts	Yes, absolutely. It would be useful if it had instructions at the beginning
What is your first impression of the games? The functionality?	OK	..	Instructions about what to do and how it works	Under construction	It is ok, and easy to use
What is your first impression of the games? The design?	Easy, but needs instruction and the ability to choose levels	..	OK	Needs better contrasts	Good for kids, and easy to understand
Explicitly your impression with Game 1	OK	OK	OK	You must be a reader	Good
Explicitly your impression with Game 2	Not good, the score must be removed because it takes too much attention	Difficult - no instruction	Good	This task needs improvement. Changing number takes all attention	Good
Explicitly your impression with Game 3	OK	OK, enjoyable	Good	Good, but needs instruction	Good

Explicitly your impression with Game 4	Too hard, needs easier levels	OK, but difficult. Woman was half once	Too many colors, too many elements. Maybe make it possible to change difficulty	Good, but needs more contrasts	Good
Explicitly your impression with Game 5	Good exercise for smooth pursuit movement. Maybe make different levels of difficulty with different speed and size/color of the ball. Unsure about the usefulness of the appearing person. Is it for concentration or the vision?	OK, but not fun	Good	Good, this can be used as it is	Good

Table 6 Answers from test persons

**Short evaluation (yes or no)**

	Person 1	Person 2	Person 3	Person 4	Person 5
Is this technology easy to learn for instructors	Yes	No	After being instructed	Yes	Yes
Would you use it to complete other (not with technology) training?	Yes	Yes	Yes	After being a bit improved	Yes
Do you think it would replace other training?	Would only complement	Perhaps	No, some parts of the training must be done	No	Yes

			without a computer		
Would you need IT help to use such at technology to training?	No	No	Maybe for the setup	No	No
Easy to learn for learners (students)?	Yes	Yes, with instructions	Needs more instructions	Yes	Yes
What do you think about the size of the screen?	Can vary between big and small screen	For some it is ok	OK	Too small	Good
The size/color/form of the representations?	OK	Would be good to be able to choose sizes	Sometimes too many elements	Too low contrasts and the calibration dots in the beginning should be bigger	Good

Table 6 Shorter answers from test persons

Please assess the product by ticking one circle per line

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable	5.25
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable	3.75
Dull	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Creative	4
difficult to learn	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy to learn	4.5
Inferior	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Valuable	5,34
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	exciting	5.5
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	interesting	5.25
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	predictable	4.75
fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	3.5
inventive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	4
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	supportive	5.5
bad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	good	5.75
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy	4.25
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasing	5.25
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	leading edge	4
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasant	6.75
Not secure	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	secure	6
demotivating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	motivating	5.5
does not meet expectations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	meets expectations	5.25
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	efficient	5.75
confusing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	clear	4
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	practical	5.75
cluttered	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	organized	5.75
unattractive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	attractive	5.75
unfriendly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	friendly	5.75

This table represents the mean of all the answers. The number on the right is the exact mean, while the ticks are based on rounded numbers.

## 6.3 Interpretation of the Answers

Generally, the responses from the 5 vision teachers were positive to the solution being used in the training. At the time of testing, only 5 games were available, and the games did not have any instructions. Instructions were added to some of the games after this evaluation, in addition to some settings to change difficulty.

Since each game was only tested once, with the current prototype at the time, these answers do not necessarily reflect how the final prototype would have been evaluated. Fortunately, it gave some pinpoints about the usefulness of the program and confirmed the previous statement about the games being complementary and not a replacement for existing techniques. All the participants answered positively on the question about the application being used for work purposes, which is the main aim for this project.

One of the mentioned issues is the lack of contrasts in some of the games. This has an easy fix, where the educator could get the ability choose which colors should appear in the background and foreground.

## 6.4 In-depth Evaluation of the Games

A 2-hour in-depth evaluation by vision teachers was performed on 27.04.2018. This consisted of a demonstration of all the games, followed by a Q&A about the application. No specific questions were prepared beforehand, but the following themes should be touched:

- How do you experience the games? Is this something you would have used in your training programs?
- How do the games compare to the existing solutions like COGPACK? Can any of these be replaced by this application?
- How positive are you about using eye tracking in these games?
- Do the solution miss anything in particular?

### **Game 1 (word differences)**

The first reaction was that this game has a similar game in COGPACK. Very good exercise for left-right saccades and fast reading. Some suggestions for further development was to make the

game run for x words instead of x seconds, and count seconds instead. This would be a better way to track progress. It should also be possible to choose background and position of the words to cover up-down and diagonal saccades. If a touch screen was used here, it could have been a good training of eye-hand coordination as well.

### **Game 2 (maze)**

This game is too hard and should be made easier by making the walls shorter with a longer distance. Apart from that, the game is a good way of training the ability to navigate the eyes through a road avoiding obstacles. One feature that was discussed in this game was initially a bug, when one eye is closed, the user can use the other eye to push the inactive ball. This could have been a fun and challenging game. To make the game easier, the goals could have been lines instead of boxes. All these things should have been made as settings in further work.

In this game, it is good that the vision teacher can see the position of the user's eyes at all times. This may contribute to the detection of problems during the training.

### **Game 3 (save the rocket)**

This game is very similar to another COGPACK game where the user should eliminate incoming shots in a box. The new thing here is that this game feature eye tracking and has better graphics. One way of improving this game could be to give different amount of point per exploding meteor according to how close to the rocket the meteor is. This would encourage more "searching" and less waiting in the middle. The game could also feature different types of elements instead of only the rocket and the meteors and the contrasts could vary. Apart from that, the game is very good for training saccades and is forcing the user to search around the screen with the vision. Also very good for visual attention.

It could have been interesting to try this game on a big touch screen, where the user had to touch the meteors to eliminate them. This would include eye-hand coordination as well. The point system could be changed to give different amounts of points depending on how fast the user eliminates the meteor.

#### **Game 4 (find the outstanding object)**

This game is good as it is and could be a replacement for different types of books used in the training today. It is very straight forward and fits people in all ages. The game trains saccades between objects as intended, both left-right, up-down and diagonal. For improvement, it would be nice to be able to choose where in the screen the objects should appear.

#### **Game 5 (follow the ball)**

This game is ok as it is but should have more settings. One way is for instance to remove the distraction for the easy levels, and maybe add more distractions for the harder levels. One could also vary the size of the ball according to the level of difficulty. The ball should also have a more defined movement. The progress could instead of the current solution be measured by number of points achieved during limited time. Apart from this, the game is a good training for smooth pursuit movements.

#### **Game 6 (which one is missing?)**

This game is a bit too hard. It should give the user the ability to choose the difficulty and maybe have increased difficulty for each round. Since the human mind cannot remember that many objects, the number must be decreased. In this game, the space bar is used to pick up items. Maybe blinking to pick up could make another layer of difficulty? The game is a good training for visual memory and saccades and could be used in the training with these changes applied.

#### **General**

This application could be used in today's training if the mentioned changes had been applied. It would not replace the existing physical training but could be used as a complementary training.

The reason that a computer application cannot replace existing training methods is because of lack of body movement and depth perception. Since the vision is going to be used in the real world, it must also be trained in the real world with body movements connected to the vision. For specific issues like trouble with saccades, a computer application may be used, but this would not provide the whole experience.

The game could be distributed to people at home, where they could do some training without a vision teacher. The equipment is cheap and available enough that it could be a great tool for

people not involved with a vision teacher. The game could also help people with different types of paralysis that cannot move their arms. They would have been able to train the vision because of the eye tracking.

Eye tracking in these games feels natural and the interaction methods are sufficient. A good thing about eye tracking is that it forces the user to look at the object with the high acuity vision and not just aiming in the peripheral vision. Although eye tracking is good, it is important to also utilize the mouse and keyboard for some of the games because of the training of eye-hand coordination.

To be a complete training application that should replace every other computer program, it must have more games than it currently has. COGPACK has 64 games, and it is obvious that 6 games cannot do the same job. But with enough settings, this game may come close.

A suggestion about another game that could be added is an exercise where an object appears on the screen and disappears a few milliseconds later. Afterwards, the user may be asked about which object appeared. This would be a great way to train issues such as visual neglect.

The application would have been more complete if it provided logging of the session. In this way, the vision teachers could have replayed the session afterwards and looked for specific errors in the eye usage.



## 7 Conclusion

In this chapter, answers to the initial research questions from chapter 1 will be presented. The answers are based on the evaluation described in chapter 6.

### 7.1 Question 1

- How can computer games complement existing methods for oculomotor training?

Computer games could be very good tools for motivation in rehabilitation. According to the test subjects from chapter 6, every one of them thought that the games could complement existing training methods. The most important part while making such games are that they should be easy to use, both for the educator and the participant. The most common feedback from the test subjects was connected to ease of use, and this sets a strong pinpoint to the fact that this is important.

Another important aspect to consider while making rehabilitation games is the need for a wide selection of settings, for the vision teacher to personalize the game for the user. A very good indication of this is that some of the test subjects thought that a game was just fine as it is, while others reported problems like too small elements, too little contrasts or too many colors. The ability for the educator to control this would provide the ability to personalize the experience so it has a broader fit.

### 7.2 Question 2

- What benefits or drawbacks do computer applications have compared to traditional training methods?

Every one of the test subjects stated that computer applications cannot completely replace the existing manual methods, and the depth evaluation discussed a few reasons for this. The main reasons are the lack of body movement connected to the eyes, and lack of depth perception. These areas are very important to include while doing oculomotor training.

The depth evaluation also revealed some benefits of computer applications compared to traditional methods. One of these are that the user is not dependent on a vision teacher being present, and the application can therefore have a much broader reach. Another direct benefit is that amputees or people with paralysis also can perform the training. This is not possible in applications like COGPACK because of the excessive need for mouse or keyboard input.

### 7.3 Question 3

- How suitable is eye tracking technology in training of oculomotor issues?

According to the depth analysis, eye tracking in these games feels natural, and forces the user to move the eyes around instead of just moving the mouse and hitting with luck. Since the eyes are the primary input method, there is no way the user can complete the levels without moving the eyes to the specified places. This makes eye tracking a very good fit for training of oculomotor issues.

Another thing that makes eye tracking suitable is the ability to log the sessions. This has not been done during this thesis, but it would be possible to do with the proper licenses. This would give the vision teacher the ability to give homework and check how the user performed afterwards. It could also reveal smaller issues that cannot be perceived by the naked eye.

On the other hand, eye tracking requires special equipment, and for some people \$99 may be a bit much to pay if the motivation is not already present. Most of the games in this thesis would not work at all if eye tracking is not present, so this may be a negative requirement.

The price and barrier to acquire the equipment does not exceed the benefits and possibilities, so the conclusion is that eye tracking is very suitable in training of oculomotor issues.

## 7.4 Question 4

- Is there a need for other input methods in addition to eye tracking?

The answer to this question depends on how broad the application should reach. If the only focus was field of view and visual acuity, using eye tracking alone would be sufficient. But since eye-hand coordination also is a part of the training, some of the games should include the use of mouse or keyboard, or more preferably a touch screen. In this application, this was game 1, where the user must click on buttons to register the answer. To be able to move the mouse to the right button, some eye-hand coordination is required.

In game 6, the user picks up the item that has gaze focus by pressing the space bar. This is a case of gaze-added interface and feels very intuitive for the user. In such an application one could either interact with the application by blinking or through a button click. According to the depth evaluation, this could have been dependent on level of difficulty, where harder levels required blinking. This means that the game could have been made without the space bar as input method, but it would have been much harder.

The conclusion is therefore that the application could have been made with eye tracking as the only input method, but it would then have a smaller reach in terms of what the user can train. Input from mouse or keyboard is considered sufficient as additional methods.

## 7.5 Final

According to the feedback from all the vision teachers, this application could be useful in training of oculomotor issues. It still needs some work before it can be used in the vision teachers' daily training, but the methods have shown great potential.

## 8 Further Work

The application made in this thesis already has a wide reach, but it still lacks some features before it is considered a full digital oculomotor training application. It lacks settings in some of the games, many personalization possibilities and logging which all are considered important for such an application. Another important part is to look at usage of different eye trackers than the EyeX for even more possibilities, or a VR headset for more realistic scenes.

For the part about settings, the issue here has mainly been limited time, so it would certainly be possible to add settings to all the games. Exactly what settings should be present and how deep the settings should go, needs a bit more research to be answered correctly.

For the application to be as useful as possible for vision teachers, it needs a logging possibility. This means that the metadata from the session should be saved for later review by vision teachers. This could also maybe have been connected to the application made by Mads Gjerstad Eide and Ruben Watanabe (2017). Unfortunately for this to be possible, there is a need for an extended Tobii license that is considered expensive. It would therefore be interesting to look at other providers than Tobii to see if they would be as good a fit.

The only eye tracker that was tried for this project is the EyeX tracker from Tobii. It would be interesting to try features from more advanced trackers like Tobii 4C. One of these features are head tracking, which would make the application able to notify the user when their head is not as still as it should be during the training.

The last thing to mention about the technical part of the existing games is that the games generally needs more content to be a complete training application. Some of the potential additional games have already been mentioned in the depth evaluation in chapter 6.

The games need some more testing to prove any direct effect. Therefore, an important part of the further work is to get the application tested over time by people who struggle with oculomotor issues. According to what has been told earlier, less than a month should be enough to see improvement.

## 9 References

Albert, D.M. and Gamm, D.M. 2016. Blind Spot. *Encyclopedia Britannica* [ONLINE]. Available at: <https://www.britannica.com/science/blind-spot> [Accessed 10 April 2018]

Bergstrom, J., Schall, A., 2014. *Eye Tracking in User Experience Design*. 1st ed. Morgan Kaufmann.

Cocos2D. 2018. *Basic Cocos2d-x Concepts*. [ONLINE] Available at: [http://www.cocos2d-x.org/docs/cocos2d-x/en/basic\\_concepts/](http://www.cocos2d-x.org/docs/cocos2d-x/en/basic_concepts/). [Accessed 13 March 2018].

Cocos2D. 2018. *About Cocos Creator* [ONLINE] Available at: <http://www.cocos2d-x.org/docs/creator/manual/en/getting-started/introduction.html>. [Accessed 13 March 2018].

Carl Rod Nave. 2000. *The Rods and Cones of the Human Eye*. [ONLINE] Available at: <http://hyperphysics.phy-astr.gsu.edu/hbase/vision/rodcone.html#c2>. [Accessed 10 February 2018].

Chappell, D. 2006. *Introducing Windows Presentation Foundation*. [ONLINE] Available at: <https://msdn.microsoft.com/en-us/library/aa663364.aspx>. [Accessed 6 February 2018].

EyeTech. 2018. a. *About EyeTech - Eye Tracking Technology for OEMs*. [ONLINE] Available at: <https://www.eyetechds.com/abouteyetech.html>. [Accessed 27 May 2018].

EyeTech. 2018. b. *TM5Mini - Eye Tracking Technology for OEMs*. [ONLINE] Available at: <https://www.eyetechds.com/tm5mini.html>. [Accessed 27 May 2018].

EyeTech. 2018. c. *VT3Mini - Eye Tracking Technology for OEMs*. [ONLINE] Available at: <https://www.eyetechds.com/vt3mini.html>. [Accessed 27 May 2018].

EyeTech. 2018. d. *VT3XL - Eye Tracking Technology for OEMs*. [ONLINE] Available at: <https://www.eyetechds.com/vt3xl.html>. [Accessed 27 May 2018].

Frisby, J.P. and Stone, J.V. 2010. *Seeing – A Computational Approach to Biological Vision*. Second edition. London: The MIT Press.

Good Reads. 2017. *Where's Waldo? By Martin Handford*. [ONLINE] Available at: [https://www.goodreads.com/book/show/344189.Where\\_s\\_Waldo](https://www.goodreads.com/book/show/344189.Where_s_Waldo). [Accessed 18 May 2018].

Ivanov, I. V., Mackeben, M., Vollmer, A., Martus, P., Nguyen, N. X., & Trauzettel-Klosinski, S. 2016. Eye Movement Training and Suggested Gaze Strategies in Tunnel Vision - A Randomized and Controlled Pilot Study. *PLoS ONE*. Available at:

<http://doi.org/10.1371/journal.pone.0157825> [Accessed 05 April 2018]

Ker Than. 2016. *How the human eye works*. [ONLINE] Available at:

<https://www.livescience.com/3919-human-eye-works.html>. [Accessed 23 October 2017].

Klaus Marker. 2016. *Marker Software - COGPACK*. [ONLINE] Available at:

<http://www.markersoftware.com/USA/frames.htm>. [Accessed 18 January 2018].

Lund, K, 2015. *Øyesporing som pekeenhet i plattformspill*. Master. Bergen: Bergen University College & University of Bergen.

National Eye Institute. 2014. *Facts About Retinitis Pigmentosa*. [ONLINE] Available at:

[https://www.nei.nih.gov/health/pigmentosa/pigmentosa\\_facts](https://www.nei.nih.gov/health/pigmentosa/pigmentosa_facts). [Accessed 10 February 2018].

Purves D., Augustine GJ., Fitzpatrick D., William, HC., Lamantia, A., Mcnamara, JO., Williams, SM., 2001. *Neuroscience*. 2nd ed. Sunderland (MA): Sinauer Associates.

Rochester Institute of Technology. 2018. *Rods & Cones*. [ONLINE] Available at:

[https://www.cis.rit.edu/people/faculty/montag/vandplite/pages/chap\\_9/ch9p1.html](https://www.cis.rit.edu/people/faculty/montag/vandplite/pages/chap_9/ch9p1.html). [Accessed 10 February 2018].

Rucci, M., McGraw, PV., Krauzlis, RJ., 2016. Fixational eye movements and perception. *Vision Research*, [Online]. 118, 1-4. Available at:

<https://www.sciencedirect.com/science/article/pii/S0042698915003648?via%3Dihub> [Accessed 9 February 2018].

Synsenteret Drammen. 2017. Synstrening. [ONLINE] Available

at: <http://syndrammen.no/home/test/synstrening/>. [Accessed 3 March 2017].

Susi, T., Johannesson, M., Backlund, P., 2007. *Serious Games - an Overview*. University of Skövde, Sweden: School of Humanities and Informatics.

The Optometric Extension Program Foundation. 2018. *VisionBuilder Home Version 3*. [ONLINE] Available at: <https://www.oepf.org/product/visionbuilder-home-version-0>. [Accessed 18 May 2018].

Tobii. 2015. a. *Tobii is the world leader in eye tracking*. [ONLINE] Available at: <https://www.tobii.com/group/about/>. [Accessed 21 November 2017].

Tobii. 2015. b. *Eye Tracking in Clinical Research*. [ONLINE] Available at: <https://www.tobii.com/fields-of-use/clinical-research/>. [Accessed 17 April 2018].

Tobii. 2015. c. *How do Tobii Eye Trackers work?* [ONLINE] Available at: <https://www.tobii.com/learn-and-support/learn/eye-tracking-essentials/how-do-tobii-eye-trackers-work/>. [Accessed 6 February 2018].

Tobii. 2015. d. *What happens during the eye tracker calibration?* [ONLINE] Available at: <https://www.tobii.com/learn-and-support/learn/eye-tracking-essentials/what-happens-during-the-eye-tracker-calibration/>. [Accessed 18 May 2018].

Tobii. 2016. *Tobii Gaming | Eye Trackers for PC Games in Desktop, Laptops & Monitors*. [ONLINE] Available at: <https://tobiigaming.com/products/>. [Accessed 10 October 2017].

Tobii. 2017. a. *How to Play Farming Simulator 17 with Tobii Eye Tracking*. [ONLINE] Available at: <https://tobiigaming.com/games/farming-simulator17/>. [Accessed 12 March 2018].

Tobii. 2017. b. *Tobii Eye Tracking | Assassin's Creed Origins*. [ONLINE] Available at: <https://tobiigaming.com/games/assassins-creed-origins/>. [Accessed 12 March 2018].

Tobii. 2018. *The world's most innovative wearable eye tracking system for real-world research*. [ONLINE] Available at: <https://www.tobii.com/siteassets/tobii-pro/brochures/tobii-pro-glasses-2-brochure.pdf?v=5>. [Accessed 6 February 2018].

Unity. 2017. *Unity - Fast Facts*. [ONLINE] Available at: <https://unity3d.com/public-relations>. [Accessed 25 October 2017].

Unity. 2016. *Games by the numbers report (2016 Q2)*. [ONLINE] Available at: <http://response.unity3d.com/games-by-the-numbers-q2-2016-report>. [Accessed 25 October 2017].

Vareberg, DO. 2012. Atferdsproblemer kan skyldes synet. [ONLINE] Available at: <http://forskning.no/forebyggende-helse-barn-og-ungdom-kjonn-og-samfunn-pedagogiske-fag-skole-og-utdanning/2012/10>. [Accessed 3 March 2017].

Wilhelmsen, GB., Wankel, V., 2014. *Friske øyne - svekket syn*. 1st ed. Bergen/Moss: Høgskolen i Bergen.

Watanabe, R., Eide, MG. 2017. *Detecting oculomotor problems in children using eye tracking*. Master. Bergen: Western Norway University of Applied Sciences & University of Bergen.

Yarbus, A., 1965. *Eye Movements and Vision*. 1st ed. Moscow: Nauka Press.



## Appendix A: Eyes

### A I Anatomy

The human eye is an organ used to project images from the outside world into the brain. This is done by focusing light through the pupil and lens to the retina, which sends information about the environment to the brain. In the front of the eye is the iris, which has an opening called the pupil. The pupil is always changing in size to adjust the amount of light entering the eye. If there is much light exposure, the pupil is getting smaller and vice versa.

Behind the pupil there is an organ called the lens, which is surrounded by some muscles. These muscles keep the lens in place and has an important role regarding your vision. When the muscles relax, the lens flatten, making the eye able to focus on objects that are far away. When the muscles are contracting, the closer objects come into focus. (Ker Than, 2016)

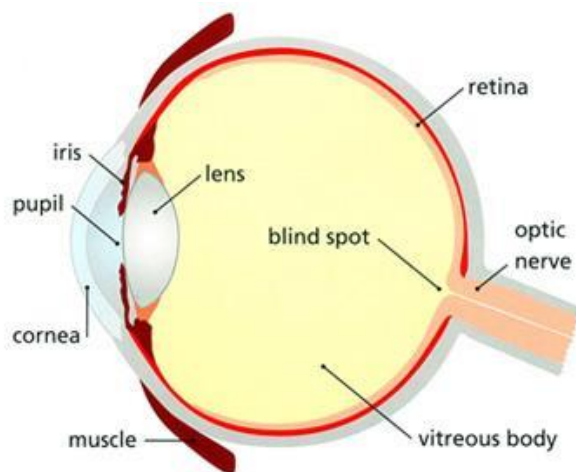


Figure 26 Eye anatomy. Source: [http://www.moorfields.nhs.uk/sites/default/files/styles/content\\_page\\_full\\_width\\_500x280/public/AnatomyOfTheEye500x280.jpg](http://www.moorfields.nhs.uk/sites/default/files/styles/content_page_full_width_500x280/public/AnatomyOfTheEye500x280.jpg)

When light hits the eye, it is sent to light-sensitive receptors in the back of the eyeball. These receptors are either active, inactive or somewhere in between. If a receptor is active, it sends information about light at the exact spot. If it is only partly active, it sends information about a shade of grey to the brain. This information creates what could be called an “inner screen” which creates the complete vision based on this incoming light. (Frisby et al., 2010, page 2).

The retina has a blind spot in which there are no photoreceptors. This means that there is no vision at this specific area. The blind spot is present in the human vision, but when both eyes are open this is not visible. If one eye is closed, the blind spot is not visible because of visual overlap, but if one eye is closed, the blind spot is present. The reason this blind spot is hard to perceive, is that the brain automatically fills in the missing piece. (Albert et al., 2016)

## A II Extraocular Muscles

There are three pairs of extraocular muscles that control the eye movements. These are responsible for moving the gaze to the desired object after getting information from the brain.

The left-right movement is performed by the lateral and medial rectus muscles, where the lateral drags the eye away from the nose and the medial drags towards the nose. The up-down movement is performed by the superior and inferior rectus muscles together with the superior oblique and inferior oblique. When looking straight forward, both the rectus and the oblique muscles contribute to the up-down movement. When the eye is pointed outwards (abduction), the oblique muscles are the primary movers and vice versa. (Purves D., Augustine GJ. et al, 2001)

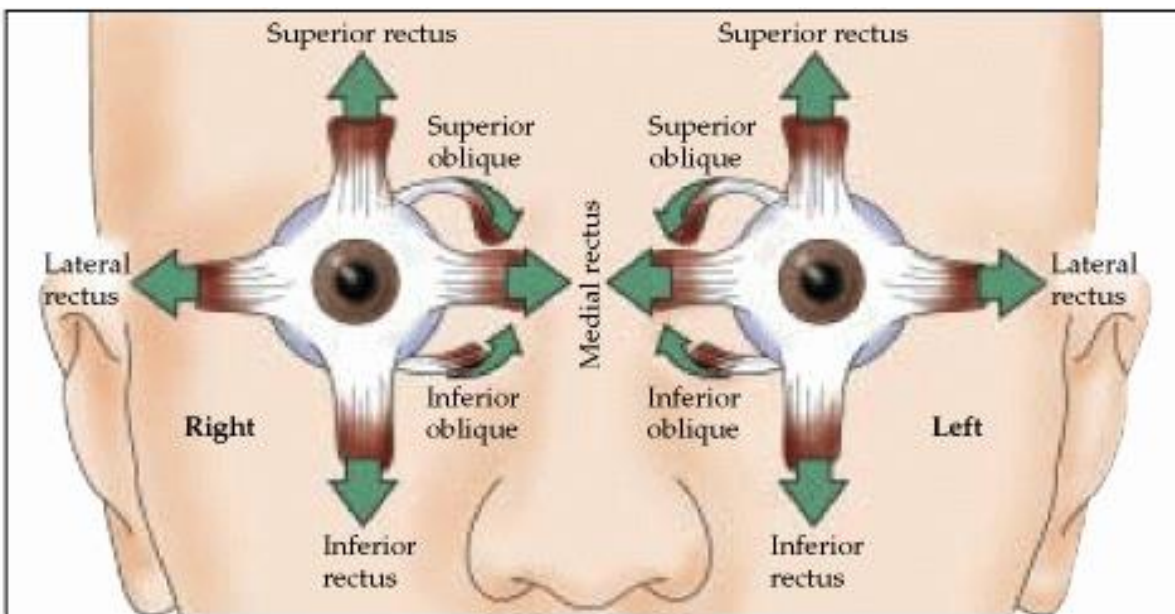


Figure 27 Muscles in action for different eye movements Source: <https://www.ncbi.nlm.nih.gov/books/NBK10793/figure/A1355/?report=objectonly>

### A III Rods and cones

Rods and cones are called photoreceptors and are fundamental in the process of converting incoming light into signals that can be processed by the brain (National Eye Institute, 2014). Rods are responsible for vision at low light levels and does not have any sense of color. Cones are active at a higher light level and provides color vision and higher visual acuity. This means that our peripheral vision is dominated by rods while the high acuity field are dominated by cones. Figure 28 shows rods and cones density at different visual angles. (Rochester Institute of Technology)

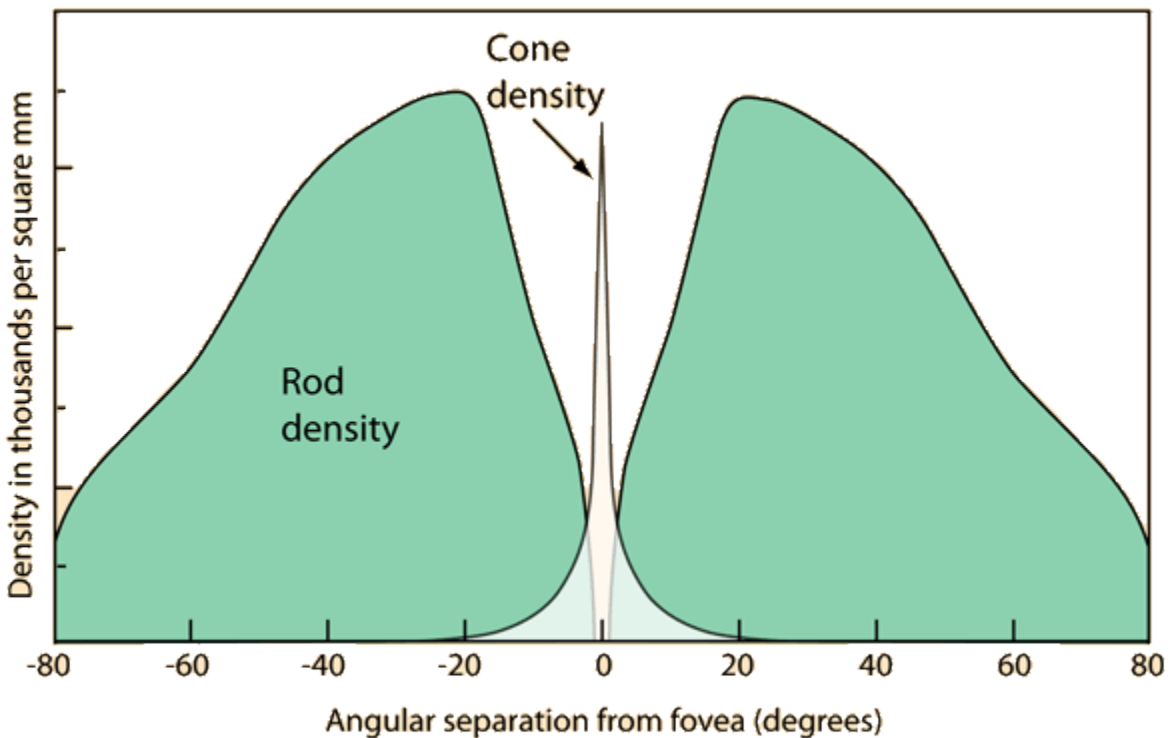


Figure 28 Rods and cones density. Source: <http://hyperphysics.phy-astr.gsu.edu/hbase/vision/rodcone.html>

There exist 6 to 7 million cones on the fovea which can be divided into 64% red cones, 32% green cones and 2% blue cones. The red and green cones are in the center of the fovea, while the blue cones have a higher sensitivity but are located outside the fovea (Nave, 2000). The cones adapt much faster to changing light levels than the rods. The rods are much more sensitive to

both light and motion than the cones, but the adaption process is much slower. Therefore, our low light vision needs around 30 minutes to become satisfactory.