

Maze Gaze - A Gaze-Based Multiplayer Game

Kevin Müller
Saarland University
Saarbrücken, Germany
s9kvmuel@stud.uni-saarland.de

Marco Siweris
Saarland University
Saarbrücken, Germany
s8masiwe@stud.uni-saarland.de

Lena Hornberger
Saarland University
Saarbrücken, Germany
s8lehorn@stud.uni-saarland.de

ABSTRACT

This paper is the result of the Seminar "Multi-User Gaze-Based Interaction" at the Saarland University. We designed and implemented an interactive application which uses gaze input. The result of this development is a gaze based multiplayer game called "Maze Gaze" in the category "Multi-User Content Adaption". In this paper we present our project idea, the requirements we made, some design and software decisions and show that a game with gaze input can make a lot of fun despite working with a eyetracker can cause a lot of issues.

INTRODUCTION

1/4 page intro (Kevin)

Motivation

Gaze can be a very intuitive and fast way of interacting with a system. However, there are still many unsolved problems related to areas such as tracking accuracy, calibration methods or unintended interactions. Because of that, currently eye-tracking is still very uncommon for end users although applications of it are countless. Researchers are working hard on solving those issues and making eye-tracking more usable in different areas of life, both for personal and professional use.

There is another important fact to consider: People who are partially or fully paralyzed have a very limited set of possibilities to interact with their environment. Because it is a hand-free interaction, gaze can be a very beneficial for people with physical impairments, giving them many possibilities to interact with other people and perform tasks that otherwise would be very hard or even impossible.

Goal

Our goal in this project is to develop a game that can be played only using the eyes. Evenmore, we want this game to be fun to play and fair against conventional input methods like mouse or keyboard. This way, everyone, including physically impaired people, can have an immersive and competitive gaming experience against other players. By doing so, we hope to

gain valuable insights and make a contribution in this field of research.

Outline

This paper consists of three main parts. The first one will lay out the foundation for the project. After examining related work from several perspectives, we will define requirements and make concrete design decisions based on the ideas we got from related papers. In the second part, we will look at the actual implementation of the prototype and explain both hardware, software and how we made it play together. The final part closes the software design cycle by evaluating the game from a performance and user-experience perspective. Based on the results and the lessons we learned along the way, we will draw conclusions and propose future work.

RELATED WORK

Gaze Input Performance

[4] 1/2 page (Kevin)

Mouse Against Gaze Input

[1] 1/2 page (Kevin)

Immersive Game Control Using Gaze

The paper "Gaze-Controlled Gaming: Immersive and Difficult but not Cognitively Overloading" was published in 2014 from Kreytz et al. in Warsaw, Poland and present at the UbiCom in September 2014 in Seattle, USA [2].

The paper is about controlling games through eye-movements. In the experiment the participants should guide a character through a maze with their eye movements. It was design within-subjects and had two fixed factors. The first was the game-control type which can be differ in 3 types: gaze-controlled with cues, gaze-controlled without cues and keyboard-controlled. The second factor was the maze complexity(easy and hard). Each participants played all three versions of the game two times, once with the easy and once with the hard maze.

The results important for our project: the completion time was faster with cues, but there were more saccades and there were more saccades with the complex maze, but the participants were gazing on the paths of the maze about 60% of the time. This resulted the main question: moving the character or scanning paths? For our project this means, that we are switching between two modes: gaze-controlled gaming an visual field scanning. This means we only move the character when the

gaze is within a given radius. Moreover we do the game without cues that show possible directions, because this reduces the game experience.

Pursuit Calibration

The paper "Pursuit Calibration: Making Gaze Calibration Less Tedious and More Flexible" was published in 2013 from Pfeuffer et al. and presented at the UIST in October 2013 in St. Andrews, United Kingdom [3].

As we know, every application which works with gaze input needs a user calibration before you can interact with it. But a "normal" calibration maybe is difficult and tedious. It requires five or more calibration points. In this paper they present another method to calibrate: "Pursuit Calibration" at which the users are following moving targets to calibrate.

In the experiment they test how the calibration should be: They test different speeds of the moving target and the time the target moves. First the target moves with a constant speed and second with an accelerating moving target which slows down at the corner of the screen and accelerates at the straight side of the screen. The results were: The target does not have the travel across the entire screen, because the final accuracy is already reached around 66% of its whole path and durations that last longer than 10 seconds are better.

This resulted in how the calibration should be: it should introduce users to the application and be very intuitive (e.g. in a stargazing application, following shooting stars). The duration of the calibration should be between 10 to 20 seconds and the size of the target should be, in our case, the width of the path, because the size of the target has an impact on the accuracy of the calibration.

In our project the pursuit calibration was a may have we did not implement.

TOTAL: 1.5 Pages

REQUIREMENTS AND DESIGN

Gaze Interaction Design Decisions

1/2 page (Marco) (Farben, Content Adaption (private/public shared areas), radius)

Agile Development Approach

The main advantage of taking an iterative approach in the development of such projects is, that one can get working prototypes really quickly and adapt the goals and requirements based on the evaluation of those prototypes. Also, by doing vertical prototypes, i.e. ones that only implement a single unit of functionality, technical limitations can be uncovered early in the development.

Initially, in our small team of three, we defined high-level milestones with deadlines that would eventually lead to the completion of the project within the given time frame. Each week, we met to discuss and assign smaller action items called "issues" to each team member using GitLab. Our initial focus was on developing two prototypes that would ensure our idea was possible. We implemented a horizontal prototype with a limited set of features, very basic graphics and only mouse and keyboard controls. By testing this prototype with some

users, we could quickly find out that our game is indeed fun to play and the competitive nature of the game does work. Because this has not been done before and there was no reference implementation online, we did a vertical prototype covering the eye-tracking functionality to ensure we could receive the gaze data from multiple eye trackers. After a while, this worked and we could focus on putting both prototypes together into a working gaze-controlled game and increasing its technical and visual fidelity.

Formal Requirements

Must-Haves

- You could play the game on a conventional Laptop display or TV set
- It could play 1-4 players at a time
- **Player can enter a running game???**
- The match field is a maze consisting of walls and paths
- Maze is random generated. The Maze does not change in playtime
- Each Player starts in one corner of the Maze
- At the beginning of the game is the hole maze dark
- The play figure of a player is a colored light cone
- Each player has a different color
- Light cone follows the gaze of the player
- Light cone tries to follow the gaze in a constant speed and tries to minimize the distance to the gaze
- If the distance of the gaze and the light cone is too big, the gaze can no longer be control the light cone.
- If the light cone can not minimize the distance to the gaze because of walls, he stops
- A small part around the light cone is lit
- The paths on which the light cone had previously been left also remain bright, but gradually darken
- illuminated area is only for the player visible how reached the are
- If a player's gaze moves near a path discovered by another opponent and not by himself, it is completely darkened
- The light cone of the player is always lit
- Lightened areas a player himself has discovered will not be darkened for this player, even if they were discovered by another player
- At the beginning of a round, a target is placed at a random position in the maze
- It gave different good power ups (enlightenment, show tart, endurance)

May-Haves

- Player can choose different levels at the beginning of a round, thereby increasing the size of the maze
- New players could join a game, without pausing
- More different bad power ups (dim, slowing, darkness)
- If a player does not look at the field for more than 10 seconds, he leaves the game and his light cone disappears from the field

TOTAL: 1.25 Pages

IMPLEMENTATION

1/4 page intro

Hardware

1/4 page (Kevin)

Software

2 pages

Menu

At the beginning of the game you can choose between many game configurations. You have the option to start the game, to finish the application, to get some help information (e.g. link to GitLab) and to call settings, where you can connect to new clients, disconnect connected clients and calibrate clients. On starting the game you can set the game configurations. You can choose the number of players, the number of rounds you want to play and the difficulty of the game (simple, medium or hard) which affects the size of the labyrinth and the paths. If the game lasts longer than one round the next round starts automatically after a short break. The game also supports in-game interruptions with a break button. In this interruption mode you can continue the game and also go back to the main menu.

Maze Generator

The Maze is generated with a depth-first-search algorithm. First we generate a $n * m$ large matrix and in every step we have a wall from x_0 to x_n and from y_0 to y_m . In the next step we generate a datatype Cell with for walls per Cell (north, south, east, west). Now we will choose a random start cell, and destroy a random wall which has not yet been destroyed from this cell and progress with depth-first-search in every cell of the matrix. You can solve this problem with a running time of $\mathcal{O}(n^2)$.

Power-Ups

We distinguish the power-ups in good and bad power ups. The good power-ups are enlightenment, endurance and show target. Enlightenment is a light explosion, if a player collects this target, a big range of $nMazeSize/7 * mMazeSize/7$ will lit. Endurance is a power-up which brings about light path is lit for a longer time. The power-up show target triggers a light flash on the target for one second. The bad power-ups are dim, slowing and darkness. Dim is the opposite of endurance, so the light path is lit for a shorter time, but only of the opponents. The power-ups slowing produce that the opponents get slower and the slower speed is fixed, and the player can

not get his old faster speed. Darkness extinguishes the light of the opponents.

Smart Player Controls

Kevin

Game Fairness

Kevin

Pupil Integration

Kevin

Calibration

Kevin

Game Lobbys

Kevin

Graphical Interface

Marco —

TOTAL: 2.5 pages

EVALUATION

Performance

1/4 page (Lena)

User-Experience

1/4 page (Lena)

TOTAL: 0.5 page

CONCLUSION

Résumé

1/4 page (Lena)

Lessons Learned

1/4 page (Kevin)

Future Work

In general, there are many possibilities to extend the game MazeGaze. Instead of controlling the menu by mouse you could implement a gaze-based control menu. So for example you have to look in different areas to choose different options in the menu. You also could create a collaborative team mode so that two players are in a team and other players are in an opponent team and the teams have to play against each other. For example that means, if a player of the one team collects a bad power-up it just will be triggered on the members of the other team. Another idea would be to change the maze in play time. Like the game "Das Verrückte Labyrinth" you could destroy walls and build up other walls, but still it is a fair maze, so every can reach each point of the maze. **Calibration**

Improve Gaze Precision Queue fehlen noch hier

1/2 page (Marco)

TOTAL: 0.75 page

TOTAL: 8 PAGES

REFERENCES

1. Michael Dorr, Laura Pomarjanschki, and Erhardt Barth. 2009. Gaze beats mouse: A case study on a

- gaze-controlled breakout. *PsychNology Journal* 7, 2 (2009).
2. Krzysztof Krejtz, Cezary Biele, Dominik Chrzastowski, Agata Kopacz, Anna Niedzielska, Piotr Toczyski, and Andrew Duchowski. 2014. Gaze-controlled gaming: immersive and difficult but not cognitively overloading. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 1123–1129.
 3. Ken Pfeuffer, Melodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2013. Pursuit calibration: Making gaze calibration less tedious and more flexible. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 261–270.
 4. Javier San Agustin, Julio C Mateo, John Paulin Hansen, and Arantxa Villanueva. 2009. Evaluation of the potential of gaze input for game interaction. *PsychNology Journal* 7, 2 (2009), 213–236.