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
1/2 page (Kevin)

Goal
1/4 page (Kevin)

Outline
1/4 page (Kevin)

—

TOTAL: 1.25 Pages

RELATED WORK

Gaze Input Performance
1/2 page (Kevin)

Mouse Against Gaze Input
1/2 page (Kevin)

Immersive Game Controll 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 Seatlle, USA.
The paper is about controlling games through eye-movements. In the experiment the participants should guide a character trough 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 reduce 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 present at the UIST in October 2013 in St. Andrews, United Kingdom.
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 a 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 lasts longer than 10 seconds are better.
This resulted 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.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

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

—

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

1/4 page (Marco)

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 coloured light cone
- Each player has a different colour
- 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 an 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 a 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 four walls per Cell (north, south, east, west). Now we will chose a random start cell, and destroy a random wall which has not yet been destroyed from this cell and progress with death-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 collect this target, a big range of $nMazeSize/7 * mMazeSize/7$ will lit. Endurance is a power-up which bring about light path is lit for a longer time. The power-up show target trigger a light flash on the target for one seconds. The bad power-ups are

dim, slowing and darkness. Dim is the opposite of endurance, so the the light path is lit for a shorter time, but only of the opponents. The power ups slowing produce that the opponents getting slower and the slower speed is fix, 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

Résume
1/4 page (Lena)

Lessons Learned
1/4 page (Kevin)

Future Work
In general, there are many possibilities to extend the game MazeGaze. In stead of control the menu by mouse we could implement a gaze bases control menu. So for example the user have to look in different areas to choose different options in the menu. For instance the person has to look in the upper left corner to choose the game option to play in a small maze. Another approach would be, the user has to control a light cone in certain corner to choose something. This is also a good way of training before the the game starts.
We also could create a collaborative team mode so that two players are in a team and other to players are in a opponent team. The teams have to play against each other. That means, if a player of the one team collect a bad power-up it just will be triggered on the members of the other team and not for the player of the own team.
Another idea would be to change the maze in play time. Like the game "*Labyrinth*" from Ravensburger in 1986. A algorithm destroy walls and build up other walls, but still it is a fair maze, so every player can reach each point of the maze. This ensures that the users can not only rely on their memory, but also a luck factor plays a role.
Another point is the continuation and improvement of the calibration. We could implement a In-application-calibration, so that means if we press the calibration button, the calibration will start on the desktop where the game is running and not on the desktop where the eye tracker is connected. This ensures a better and more precise calibration. Also we could implement a in-application pursuit calibration to get a more precise calibration and better results and a low frustration of the user.
To reduce the frustration of the user, we could improve the gaze precision. The gaze precision is currently good enough to have a good gaming experience without major frustrations but not perfect yet. The improvement is possible by adding algorithms to the actual gaze position calculated to the desired position.
Last but not least we could implement a queue to start waiting players in a fair, right sequence. If, for example, 5 people want to play, we have a problem, because the game is design for four users. So the fifth user have wait and getting in a virtual queue, when one of the four players leave to game, the waiting person will spawn in the next round. If we have two or more waiting users, they are arranged in a queue, with the principle, first come first serve.
TOTAL: 1.0 page

TOTAL: 8 PAGES