Product planning

Thijs Boumans
Patrick Kramer
Alexander Overvoorde
Tim van Rossum

1 Game concept

The goal of the game is to solve a puzzle by moving and rotating laser beams and mirrors in such a way that a predefined target is hit. The game can be played by one or more local players and one or more remote players.

There are cards present for the local players that represent mirror bases. These must be placed on the table, which will be the locations for the mirrors. The local players will be able to see the mirrors they place through the use of AR technology. Each of the local players will only be given a few of the mirror bases needed to solve the puzzle, and as such solving the puzzle requires cooperation from all local players.

The remote players can also see the placed mirrors, and can rotate them to influence the path of the laser beam(s). The remote players is also the only one who can see the obstacles that are predefined for that particular puzzle. Only by cooperation between local players (who can only move the mirror bases) and remote players (who can only rotate them) it becomes possible to hit the target and as such solve the puzzle.

The game provides various different types of mirrors with different properties, allowing for more complex puzzles. One example of such a mirror is a colored mirror, and then require the target is hit with the right (combination of) colors. Another way to make puzzles more complex is requiring that the players combine beams together to create more powerful beams.

2 Approach

This section covers development details that aren't directly related to the gameplay itself, such as software engineering practices and guidelines about client meetings.

2.1 Technical details

Although AR glasses have been provided to us, their field-of-view is very limited and is not suitable for most of our concepts. We experienced ourselves that the limited fov causes a lot of problems. Instead, we're going to try to use an Oculus Rift and cameras to create our own high fov augmented reality glasses.

The Unity game engine is going to be used to render the in-game objects like mirrors and laser beams over the camera image.

TODO: Find suitable AR library and write about it here

2.2 Software engineering

Using the Unity game engine means that we'll use a loosely coupled component-based architecture from the start. Unity has built-in unit and integration testing systems, which we'll make extensive use of in development. We'll also maintain UML diagrams of the architecture to keep an overview of how everything works, and to plan integration of new features.

We'll make use of the Scrum software development methodology to plan new features and to help stick to the schedule. As is expected, there will be a playable demo at the end of each weekly sprint. As soon as the game has all the basic ingredients to be fun to play, we'll find users to play test on a regular basis and collect feedback from them to improve the gameplay.

2.3 Guidelines

Here is a list of rules that help prevent problems during development.

NOTE: These are just ideas right now

- Meet with the client and coach every week to show a working demo
- Add tests as soon as new methods are added to verify that they work
- Have integration tests for common scenarios to reduce the need for user testing

3 Planning

The first two weeks represent the research phase. In this phase we will find a suitable augmented reality (AR) library for Unity and prepare an Oculus Rift for AR use with cameras. We'll also design an architecture for the game that covers the marker detection, networking and gameplay mechanics. All of this information will be described in the research report handed in on May 1st.

Main development commences after this phase and is organized in two week sprints. The table below describes the goals per sprint, which will serve as a helpful reference to stay on schedule during development.

Weeks	Deadline	Goal
4.1 + 4.2	May 1st	Research report described above
4.3 + 4.4	May 15th	???
4.5 + 4.6	May 29th	???
4.7 + 4.8	June 12th	???
4.9 + 4.10	June 26th	???