Development Document

**Initial Ideas**

The initial thoughts for the design of the game stemmed from the interpretation of the theme of “waves”. There were various ideas on how the theme could be interpreted. Among them, the concept of light waves seemed the most interesting.

The behaviour of light waves, in a physical sense, is guided by a set of rules. The path of a light beam can be figured out by applying logic. In addition, it is quite easy to visualise. The logical aspect to the behaviour of light prompted the idea to design a logic puzzle game.

The initial idea was to have a series of logical puzzles that the player would need to solve using a beam of light. While planning, an observation was made about the behaviour of light in a 2D perspective. This led to the idea of having varying perspective in the game. It felt original and interesting, so it would be experimented with.

**Development Log**

28/01/2020

The first concept for the game was completed. The puzzles would be separated into levels, each of which would have a detector. By shooting a laser into the detector, the player would pass the level. Each level would have obstacles that prevent the player from shooting directly into the goal. Each obstacle would have reflective and refractive properties, affecting how the laser interacts with them. There would also be a form or diffraction in the game.

29/01/2020

The structure of the project was planned. The programming section of the game would be sorted using namespaces. Scripts using the same namespace would be put into the same folder. The folders would be named the same name as the namespace they hold. The aim of using namespaces was to reduce the number of dependencies between scripts and to restrict program flow. This would make it easier to debug the program, especially once it grew in scale.

In addition, a flow-chart was developed, showing the set of dependencies. For example, the namespace containing low-level components such as pausing was called the “core” namespace. This would be depended upon by namespaces such as “UI”. This is because UI elements like the pause menu would react to the game being paused.

While a relatively solid plan was laid out for the structure of the game from a technical standpoint, it was difficult to plan. The reason is that it was difficult to predict exactly how all the components of the game would interact with each other. It would help to have a solid concept of the game before planning it’s internal architecture.

08/02/2020

The core components of the game were programmed first. At this stage, the player could move and look around. They were also able to shoot the laser from their gun. Some bugs were found, but were not fixed, as they didn’t cause problems with testing the game. Reflective and refractive obstacles were implemented, and basic forms of reflection and refraction were programmed. These were not fully functional but allowed for some testing.

It quickly became apparent that having multiple types of reflective and refractive obstacles would be too complex. It was believed that it would confuse the player too much and that it would be hard for the player to develop an intuitive understanding of the mechanics. Therefore, it would be better to keep them simple. It was decided that reflective obstacles would be guaranteed to reflect, and refractive obstacles would all refract by the same amount.

It is important to note here that by creating a prototype quickly allowed early testing. This, in turn, allowed some important decisions to be made early in the development of the game – potentially saving large amounts of time.

It was also observed that creating a game that mimics reality is not necessarily a good idea. In this case, the laser did not need to travel realistically. Rather, it needed to follow a set of mechanics that felt intuitive to the player, making them feel in control.

20/02/2020

Levels and level progression were implemented. It was decided that all levels should be in the same scene in Unity, to reduce load times of each level.

All scripts thus far were put in their designated namespaces and dependencies implemented stuck to restrictions set by the flow chart. This proved to be rather difficult. There were many occasions where it would be easy to set up a link between scripts, but this would not maintain the planned structure. In these cases, it was required to engineer workarounds that remained within the set restrictions.

However, the usefulness of this structure had already become apparent. The encapsulation of portions of the program meant that these portions would function independently. This not only made tracking down bugs easier, but also helped program organised. This could become especially useful in large projects with multiple people working on them.

Finally, the mechanics of the laser beam had seen further improvements. At this stage, the beam was working well enough to be tested in various scenarios, but it was still buggy. Altogether, it took almost two weeks to get the laser to a working state. There were many problems that emerged when coding reflection and refraction, costing a large amount of time. This is something that needs to be considered, especially since that prior to fixing the problem, it was almost impossible to do any playtesting. While this sort of problem can cause a large setback in a project, it is difficult to predict and hard to plan a solution for.

6/03/2020

Some basic levels were built in order to test the game. Two problems became apparent. The first was that the two 2D perspectives were difficult to read by the player. The laser would often travel behind objects, which wouldn’t be apparent in 2D, confusing the player greatly.

While it was intentional to make the player solve puzzles using the two perspectives, it was not imagined that such complexity would be introduced this way. This is something that should have been thought about in more detail.

The second problem that surfaced was that small errors in the angle at which the laser was shot at would get larger the more reflections/refractions it participated in. In other words, small mistakes resulted in failure. The player would have to be very precise with where they shot in order to hit the detector. This was not at all intended and would need attention in the future. The challenge that the player should have been facing were the logical puzzles and not precision based mechanics. Diffraction was highly dependent on precision and was removed as a result.

It was decided that the rest of the game would be worked on while deciding how to solve the present problems.

19/03/2020

At this stage, the scripting of the game had been mostly finished. This allowed to focus mostly on the design aspect. Some further attempts were made to create puzzles that do not suffer from the mentioned problems (2D views difficult to understand and precision dependent solutions). However, such puzzles proved very difficult to design.

This was a difficult stage of development. While progress had been made, it was mostly in the development of less important features. The problem was that progress could not have been made with levels - the bulk of the game - until something was changed.

29/03/2020

After large amounts of consideration, the decision was made to change some of the game mechanics. The first change was the removal of the 2D perspectives. These were later replaced with a moveable camera. The player would be able to snap the camera to 1 of 8 different positions. The second change was to replace 1st person view with a 3rd person character. This character would shoot in the direction of the mouse and would only not be able to look up or down. The game effectively became 2 dimensional.

The puzzles were re-designed to fit a more 2D layout and some testing was done. The effects of the changes were drastic. The game begun to feel much more understandable and there was a lot of positive feedback from play testers. The removal of one axis from the game meant that the player had far fewer degrees of freedom when aiming, reducing the amount of precision required to solve puzzles. Removal of side views meant that perspective-based puzzles wouldn’t exist anymore. Designing puzzles became simpler and the new puzzles felt far less confusing.

Here it became apparent that some features may seem interesting and fun during planning, but do not hold up in practice. It seems that it is often better to remove or replace these features with something that works better for the game.

08/04/2020

A set of 12 levels were developed for the game. Their development was quite difficult. Alternate solutions were common and the levels hadn’t even gone through a lot of playtesting. Creating a challenging and fun level took a large amount of trial and error. In some cases, levels were remade several times. This took longer than expected.

Part of the struggle with level development came from the fact that the player was given a lot of freedom. It was considered to replace the player character with a turret that can be rotated in increments of 45 or 90 degrees. This would help fix the problem, but was not implemented because it was thought that giving the player the freedom to move and aim was more interesting.

20/04/2020

During this phase of development the game was mostly being polished. Smaller bugs that were ignored earlier were fixed, sound was added, graphics were improved and more options were added to the menus. This included things like giving the player the ability to adjust screen resolution, volume levels etc. While these additions were small, they made the game feel much more complete.

In addition, a lot of playtesting was done. Overall, the feedback was relatively positive, but a lot of problems were raised. Firstly, several alternate solutions had been found to the puzzles. These were repeatedly fixed and tested again. Secondly, several “quality of life” changes were made. These included: adding a beam trajectory, narration for tutorial levels and a menu showing the controls. Playtesting helped to flesh out the game significantly.

**Conclusions**

There are several things that can be learnt from the development. The first is that planning is very important. Creating and organising the structure of the game leads to faster development and leaves less room for improvisation. As seen in this project, some features do not integrate well and have to be changed/removed. Careful planning can reduce the chance of this happening.

However, it is impossible to have a perfect plan and not encounter any problems throughout development. Therefore, the game should be brought to a playable state as soon as possible. Having a prototype allows you to playtest the game and evaluate which features work well and which features do not. The earlier you play test, the faster you can get your game to a more solid state. This is especially important for the core of the game.

Finally, it is important to be able to know when to remove features of the game. As seen in the development of this project, sacrificing an element of the game can lead to a better final product.