

Product Planning

Team Krokot
28-04-2016

Irene van der Blij	ivanderblij	4385691
Jochem Heijltjes	jheijltjes,	1534041
Mayke Kloppenburg	mlkloppenburg	4383265
Alan van Rossum	alanvanrossum	4293932
Harvey van Velton	hvanveltom	4350073

Table of Contents

- 1 Introduction
- 2 MoSCoW
 - 2.1 Must haves
 - 2.2 Should haves
 - 2.3 Could haves
 - 2.4 Won't haves
- 3 Roadmap
- 4 Definition of Done
- Appendix A: User stories
- Appendix B: Glossary

1 Introduction

In recent years, escape rooms have been getting very popular. In an escape room a team is locked up in a room. The team can escape by cooperating and solving puzzles. But what if you want to take it further?

This game is played by three players; one player has an Oculus Rift (virtual reality glasses), and the other two play with smartphones. The three players are a team of CIA agents. They are investigating the disappearance of their fellow agent. The two smartphone players are in the CIA headquarters. The Oculus player is in the woods following a lead, when he or she is knocked unconscious. When the Oculus player wakes, he/she is locked up in a dark, scary room. Deadly gas slowly starts filling the room. The gas will fill the room completely in 15 minutes; a timer starts. Luckily, the person locked in the room still has the earpiece and is able to communicate with the two players in the headquarters. The oculus player has to search the room for clues and puzzles; he can look and move around in the room and interact with objects. The two agents in the headquarters will support and help solve the puzzles. When a puzzle is solved, information (e.g. a key, a secret room, or a new puzzle) is released which will help the Oculus player escape the room.

The experience this game will give is unlike a real escape room. To make things difficult for the Oculus user, and to give a real immersed experience, anything can happen. From loss of gravity to hallucinations, everything is possible in the virtual world...

The structure of this document is as follows. Chapter 2 the features are prioritised with the MoSCoW method. Chapter 3 contains a roadmap of the development of the product. Chapter 4 is the definition of done; it is discussed when the product is considered finished. Appendix A describes user stories and appendix B is the glossary.

2 MoSCoW

In this section the possible features of our product are separated into four categories.

The categories are defined as follows:

Must have

These are the essential features of the game, that guarantee that it works correctly. Without these, the game is unplayable.

Should have

These are the features that make this game worth playing. The game is playable without these, but it will not be too enjoyable.

Could have

These are extra features that will only be implemented if there is enough time.

Won't have

These are features that will not be implemented in our product, but will be taken into consideration for an extending project in the future.

2.1 Must have

- One player must be able to look around the virtual world by turning his head while wearing the Oculus Rift.
- The Rift User must be able to interact with the virtual world through moving the joystick and pressing buttons of a wireless controller.
- Two players must be able to perform actions on their smartphones that directly or indirectly affect the virtual reality.
- There has to be a link between the actions of the Rift User and the information represented on the smartphones.
- Through solving puzzles the players must be able to 'beat' the game and emerge victorious.

2.2 Should have

- A scoring system that takes into account the amount of puzzles solved correctly, time taken and bonuses found.
- Hints must be available in case players get stuck.
- Puzzles that immensely affect the Rift User's perception of reality, such as blurry vision or inverted gravity.
- A soundtrack that plays during the game for both the Rift User and the smartphone users.
- Players should be able to see the remaining time

2.3 Could have

- Specific sound effects that are played when certain actions occur in the virtual reality.
- A scoreboard that tracks the highscores of multiple teams, across game sessions.
- Additional settings, like a crashing airplane or a defect submarine.

2.4 Won't have

- Compatibility for multiple Oculus Rifts
- Compatibility for three or more smartphone users
- Support for microphones to allow players to communicate online rather than locally
- Support for iOS and Windows phone

3 Roadmap

Week 1

- Coming up with game concepts
- Organizing the team

Week 2

- Selecting and pitching the best game concept
- Process the feedback of the client

Week 3

- Final design document
- Start with implementing the core of the system

Week 4

- Submit first peer evaluation
- Start with creating minigames

Week 5

- First playable spikes

Week 6

- Midterm plenary presentation

Week 7

- Second peer evaluation
- Beta version of the game

Week 8

- Solve issues and bugs in the beta version

Week 9

- Script game trailer

Week 10

- Third peer evaluation
- Release
- Game trailer
- Final presentation

4 Definition of Done

The definition of done describes for three different aspects of the project what it means when it is considered to be done by the developers. When the developers claim a certain aspect is done it can be assured the aspects will hold up to the standards described in this section. The three aspects in question are features, sprints and the final product as a whole.

First, the definition of done for when a feature implemented in the escape room is labeled “done”. When saying a feature is “done” it is meant that the newly implemented feature will pass certain requirements. These requirements are a passage of unit tests. These unit tests are designed specifically to ensure that the new feature doesn’t contain bugs that will hinder the system. Besides unit tests the new feature will also have passed integration testing. Unlike unit testing which tests the feature itself the integration tests will ensure that adding the new feature to the existing system won’t cause any errors or loss of functionality. Additionally to testing the new feature will also need proper Javadoc documentation. This documentation is so the developers can quickly determine what every part of the new feature does. The final requirement of a feature is the ensurance of code quality. To ensure the code quality the feature will be tested using static analysis tools namely checkstyle, pmd and findbugs. If a feature succeeds these four goals of unit tests, integration tests, code quality assurance and Javadoc documentation than it is considered done and the feature can be safely merged with the existing system.

Second, the definition for when a sprint is considered done. The requirements for a sprint to be considered done are as follows. A sprint is deemed done when the application as a whole has been tested, this includes both unit tests and integration tests. These tests are designed for both the new features which have been added during the sprint as well as the system itself which may have been updated during the sprint. The tests are designed to ensure no bugs/errors are within the system. Besides testing the system the application in question will have to be able to run. This means that the game in question can be demoed to the stakeholders who will than give their feedback. An additional requirement of the sprint is that the documentation of the system is updated. This includes new documentation for any new features as well as updated documentation for when the system is altered. Besides testing and documentation the system after a sprint will need code quality ensurance. The code quality will be tested using the static analysis tools pmd, findbugs and checkstyle. The final two requirements for when a sprint is deemed done is a sprint reflection and a sprint backlog for the next sprint..In the sprint reflection the team will reflect back on what planned tasks were and weren’t accomplished as well as reflect back on any problems that occurred during the sprint. This new acquired knowledge will be used to create the next sprint backlog.If a sprint has achieved all these goals it is considered done.

Finally, the definition for when the escape room game is considered done. The requirements for the product to be considered done are as follows. Just like with the sprint and feature definition of done the game is considered done if the system as a whole has been tested for

a certain percentage. This percentage will be 75% and will be judged by the cobertura line coverage tool. Besides testing the game needs proper javadoc documentation for every method and class of the application. Additionally the application needs a certain level of code quality ensurance. Errors in the code Quality will be detected using pmd, checkstyle and findbugs and will be corrected. Besides quality ensurance the game also needs to have certain features implemented. These features include all the features described in section 3.1 as well as a certain percentage of the should haves described in section 3.2. If the game includes all these features as well as achieving 75% test coverage, is fully documented in Javadoc and doesn't contain any bad coding habits as detected by checkstyle, findbugs and pmd. It is officially considered done.

Appendix A: User stories

As an Oculus Rift player
I be able to look around
So I can see all of the room

As an Oculus Rift player
I want to be able to interact with the object in the room
So I can find hidden clues and puzzles

As an Oculus Rift player
I want to be able to move around
So I can explore the room

As a smartphone player
I want to be able connect my phone to the game
So I can join the game

As a smartphone player
I want to be able to see the puzzles
So I can solve them and help the Oculus Rift player

As a smartphone player
I want to be able to interact with my phone
So I can solve the puzzles

As a player
I want to be able to solve all the puzzles
So I can win the game

As a Oculus Rift player
I want to be able to see the time left
So I know how fast to be

As an Oculus Rift player
After receiving a hint from the smartphone player
I want to be able to use the information to progress in the game.

Appendix B: Glossary

Checkstyle - A static code analysis tool that checks whether Java source code complies with the Java coding rules.

CIA - Central Intelligence Agency, the foreign Intelligence Service of the United States Government.

Cobertura line coverage - A measurement to describe the degree to which source code is tested.

Findbugs - Static code analysis tool for Java bytecode.

Integration testing - A phase in software testing in which separate software modules are merged into one and tested as whole.

iOS - Mobile operating system by Apple Inc.

Junit testing - Code testing using the JUnit testing framework (Java).

Minigame - A small game within the main game. The outcome of a minigame often influences the course of the bigger game.

Oculus Rift - A pair of Virtual Reality goggles with a stereoscopic field of view, designed for gamers. The device measures the movements of its wearer.

Pmd - Static code analysis tool for programming mistakes.

Smartphone - A phone with many features running an advanced operating system, similar to a desktop computer.

Virtual Reality - An artificial world, generated by a computer.

Windows Phone - A phone running the Microsoft Windows Mobile operating system.