Product Vision

Computer Games Contextproject 2015-2016
Course TI2806, Delft University of Technology

Group PixelPerfect

April 28, 2016

Supervisor
Dr. ir. Rafael Bidarra

Teaching Assistants
Sander van den Oever
Jurgen van Schagen

Team

David Alderliesten	4368703	*J.W.D.Alderliesten@student.tudelft.nl*
Floris Doolaard	4362748	*F.P.Doolaard@student.tudelft.nl*
Dmitry Malarev	4345274	*D.R.Malarev@student.tudelft.nl*
Jesse Tilro	4368142	*J.Tilro@student.tudelft.nl*
Wouter Zirkzee	4398858	*W.J.Zirkzee@student.tudelft.nl*

Table Of Contents

1. Introduction

PixelPerfect was assembled during the fourth quarter of the academic year of 2015-2016 for the Context Project course of the Computer Science bachelor program at the Delft University of Technology. The team consists of five members and was assigned to design, develop, and release a virtual reality game for the Oculus Rift virtual reality platform.

The game had to be playable within a “party situation,” which was outlined to be a situation in which multiple players in the same room were actively interacting with the game at all times. The choice was made to develop a game that would be unique to the Oculus Rift wearer and still provide an interactive and interdependent experience for those players without an Oculus Rift. A requirement given was that all players had to be actively participating in the game, and it was not acceptable to have players simply observe the Rift player or partake in a passive state, such as making a level and watching the Oculus Rift player play through it

The decision was made to develop a space freighter game that was inspired by the game “Keep Talking and Nobody Explodes.” In that game, one virtual reality headset must communicate with one or more other players in order to defuse a bomb based on challenges that must be relayed and communicated about by players who do not wear a virtual reality headset. By successfully completing challenges, the bomb can be diffused and nobody would blow up/lose the game.

The space freighter scenario for our game is also a cooperative experience, in which players without the virtual reality headset must “fix” problems and collect useful resources in a ship in order to prevent it from malfunctioning, killing the crew on board and losing the freight. The player wearing the Oculus Rift must then navigate the ship and assist the players through their tasks, which require intensive communication and are all designed to be interdependent on the player roles.

This Product Vision document contains an outline of the product, the target audience, an outline of the product features (split using the MoSCoW (Haughey, 2000) method), what makes this product unique and the given budget to accomplish this.

2. Target Audience

As with any product on the market, a game has to have a designated target audience that it will sell to as well as a clear definition of the main goal of the game. In the following sections we will discuss what the player has to do to win the game and the target audience of the game.

2.1 Objective

The objective of this game is to make sure that a spaceship remains operational during a journey to a remote planet and to deliver the ship's freight safely with the help of your crew. This is done by performing tasks to maintain the ship under the guidance and supervision of the captain. The captain uses the Oculus Rift to observe the surroundings, to provide assistance to the rest of the crew, and to guide the ship. The other players, who are divided into specific roles (such as engineer, gunner, or navigator), have to perform various mini games in order to avoid damage to the ship. If too many of these tasks are failed, the ship and crew are lost and the game is lost. If the ship arrives at the targeted planetary destination, points are awarded based on the team's performance, and a victory can be attained.

2.2 Target Audience

The target audience of the game is any player with an interest in a science-fiction theme and fans of the simulation game genre. An example of such a situation could be a party, a gathering of people within a home setting, or a convention. The player total must be five. More players allows for role switching, less character makes the game difficult to play.

The demographic must have access to an Oculus Rift virtual reality device, and all other players must have access to a compatible mobile device in order to play the mobile client for the game. The Oculus Rift must have the main game client installed, and all phones must have the respective applications installed before use.

3. Customer Needs

In order for a game to succeed, it needs to be fun and interesting to the people playing it. Although this seems simple, it is difficult to define what one believes is “fun.” How can one categorize fun?

According to the Theory of Flow (Moneta & Csíkszentmihályi, 1996), a game must have a balance between the skills possessed by the user and the challenges found within a game. This means that a game should be easy for beginners, but also scale in difficulty as the player gains knowledge and skill. This balance is required in order to keep the player entertained in the game. The reasoning behind this balance is that if the game is too difficult it may cause a feeling of anxiety or discomfort to the player. If the game is too easy the player might start feeling bored at the lack of challenge and therefore stop playing.

A game should also have a control system that is immediate and easy to grasp (Dillon, 2010, p. 27-31). This means that the players should not feel discouraged while playing the game because the instructions are too difficult to understand or the influence on the game is not directly apparent.

Another concept that is important to keep a player from being dissatisfied is immersion. Lamaree (2002) defines immersion as “a state in which the player’s mind forgets that it is being subjected to entertainment and instead accepts what it perceives as reality”. In order for a game to be good, it has to give the player a sense of reality and immersion, almost an illusion that the virtual world is the real one.

Cooperation is a key factor to take into consideration in a multiplayer game. A concept that is used to improve cooperation is “Distributed Cognition”. Distributed Cognition is an approach to analyse "a collection of individuals and artefacts and their relations to each other in a particular work practice" (Rogers and Ellis, 1994), in other words, the resources of an individual are shared with other people to increase the amount of resources available or to complete a task that is difficult to do alone.This means that, in a cooperative game, players should be able to use the assistance of their teammates to complete an objective easier.

As per the above analysis, the main needs that make a cooperative game fun and interesting are:

- A balance between difficulty and the player’s skill.
- A control and influencing system that is easy to understand.
- The ability to keep a player immersed in the game.
- The use of distributed cognition to improve co-operation in the game.

4. Requirements

In the previous section, we listed a few concepts that make up for a fun and interesting game and keep a player from getting bored/losing his interest.In this section, we will describe how these concepts are going to be fulfilled in our design.

4.1 Functional Requirements

In this section, the features required will be described using the MoSCoW (Haughey, 2000) method. The features are categorised into four groups:

- **Must haves:** features that are essential for the game to work as intended.
- **Should haves:** features that are wanted, but do not break or fundamentally alter the game if they are missing.
- **Could haves:** features that can be implemented given enough time for development.
- **Would haves:** features that are most likely not going to be implemented, but can be made during a continuation of the project or if implementation forgoes better than expected.

The next section will describe the individual features in the game per category of the MoSCoW method.

4.1.1 Must Haves

These features are essential for the creation of the game:

1. The player wearing the VR headset must be able to start the game as a captain.
2. The captain must be able to look around in the virtual spaceship overseeing both the inside and outside view.
3. The captain must be able to choose a location/planet for the ship to travel to before initiating the game.
4. The amount of time (i.e ten minutes) the game will take must be represented by the main timer.
5. The maximum round timer must be initiated at the start of the game.
6. The game must keep track of a damage value, representing the extent to which the spaceship is damaged, in the cockpit.
7. The game must keep track of the damage value, representing the extent to which the spaceship is damaged, to all non-VR players.
8. If the ship's total damage value reaches a threshold (i.e 100 hit points), the game must be terminated and the game is lost.
9. The game must schedule events that pose a threat to the ship and can inflict damage to the ship upon event failure.
10. Each active event must be listed in a log in the HUD visible to the captain.

11. Events must not be scheduled at a fixed interval between each other.
12. The pace at which events occur must be balanced so that no player will be idle for more than 10 seconds.
13. The events must have a fixed lifetime and completion time limit and a damage value that is applied to the ship's health if failed.
14. When the lifetime of an event expires, the ship's total damage value is increased with the event's damage value.
15. The crew players must be able to interact with an interface to respond to events, in the form of pressing buttons on a touch screen.
16. If the crew players manage to perform the appropriate interactions through the presented interface in response to an event, the event must be discarded without having any effect on the total damage value, in other words, not decreasing the health of the ship.
17. If the ship has reached the destination (i.e. the main timer has expired), the game must be won.
18. A victory screen must be displayed when the game has been won..
19. The victory screen must transition the players to the main menu.
20. The Oculus Rift client and all the clients for the non-Oculus Rift players (i.e. the crew) must be interconnected and synchronized with each other.
21. The Oculus Rift player must be able to create the game. The other players must be able to connect to the game after it is created.
22. Any player must be allowed to quit the game at any moment.

4.1.2 Should Haves

These features are not essential for the game to work, but improve the game:

1. As the game progresses, the frequency of events occurring should increase.
2. Different types of events should have different time limits and damage values.
3. Events and damage should be visually indicated at the location where they take place in the ship (inside the virtual ship perceived by the captain).
4. On successfully completing an event, the crew should be awarded with credits (a virtual commodity).
5. On a successful mission, the crew should be awarded credits/points.
6. The captain should be able to choose between multiple navigational paths to reach a selected destination/planet.
7. Each path should have a different effect on the challenges encountered by the players.
8. Events should be parameterized, i.e. the captain is presented some random (e.g. numeric or visual) information related to the event that must be taken into account by the crew players.
9. The crew players should be able to perform actions in the form of mini games rather than just button presses.
10. The player should be able to collect items during idle time which help them in events, such as oil or wiring.

4.1.3 Could Have

These features are seen as additions to the game, and are not essential but should be created if time allows:

1. As the game progresses, more challenging types of events can be introduced.
2. The ship moves linearly, with a fixed pace, over a predefined track through space.
3. The captain can move the ship over a two-dimensional plane orthogonal to the track in order to evade obstacles (e.g. asteroid/debris).
4. The frequency and types of events occurring can be correlated with the captain's steering performance.
5. The crew players are presented a map of the spaceship, allowing them to indicate where they will perform an action.
6. The actions to be performed by the crew can be location-based, i.e. the players must go to the right location in order to handle the event in addition to performing the right action.

4.1.4 Would have

These features are seen as additions that would be interesting changes to the game, but given the allotted time are not likely to be implemented:

1. The captain can move the spaceship in a full 3D environment, meaning the captain can freely alter the route and avoid collisions with asteroids or other space debris. The ship is not forced to move forward unless the captain instructs to do so.
2. Visual feedback can be displayed on the ship for events. The damage is context based, so plasma damage would cause burn damage, and electrical failures would cause a ship without warning lights.
3. The general difficulty (i.e. frequency and types of events, lifetime and damage values, total time, damage threshold) of the game can be different for each possible route through space.
4. Mobile players can move between locations of the ship.
5. Players can call for help from other players if a piece of their puzzle is required from others.

4.2 Non-functional requirements

These requirements concern constraints on the development process and the quality attributes of the final product.

1. The game must be programmed in the Java programming language.
2. The game must run on the jMonkeyEngine library.
3. The game must be able to interface with an Oculus Rift DK2 hardware device through OpenVR.
4. Git must be used as source control software.
5. The software project must be hosted publicly on GitHub.
6. A continuous integration server must be used.
7. The components of the system must be tested using jUnit automated tests.
8. Code coverage of tests must be monitored through Cobertura.
9. The codebase must be statically analyzed using CheckStyle, PMD and Findbugs.
10. Implementation of features must be done in a test-driven manner.
11. The software must be developed according to the pull-based development model (Gousios, Pinzger & Deursen, 2014).

5. Product Uniqueness

The product under development utilizes non-unique features to develop a game that, together, make a unique combination for virtual reality gaming. These features are:

- Verbal communication between players as a requirement to succeed.
- Oculus Rift and mobile phones used in tandem as gaming gear.
- A sci-fi, space theme.
- A simple graphical style.

For our research on product uniqueness, we statistically analyzed the volume of games offered on the Steam store, a video game distribution platform that has an active user base of over 12 million users. If we look at the games that use Oculus Rift, we can see that only two can be found that play local co-op (Image 1) and none that also allow cross-platform multiplayer (Image 2). We chose to analyze the Steam store over consoles because Virtual Reality technology is not an available feature for those platforms at the current moment due to hardware limitations. The product entertains a small group of players that are using the Oculus Rift and want to play with the device together. It has been stated (Anderson, 2015, p. 3-6) that in the United States alone, about 68% of adults in 2015 owned a smartphone, while for people between 18-29, the percentage rises to 86%, which makes it a lot easier to play together if one person owns an Oculus Rift. The widespread availability of smartphones allows a full party of players to enjoy the Oculus Rift at once, instead of having to observe the player with the Oculus Rift and wait their turn.

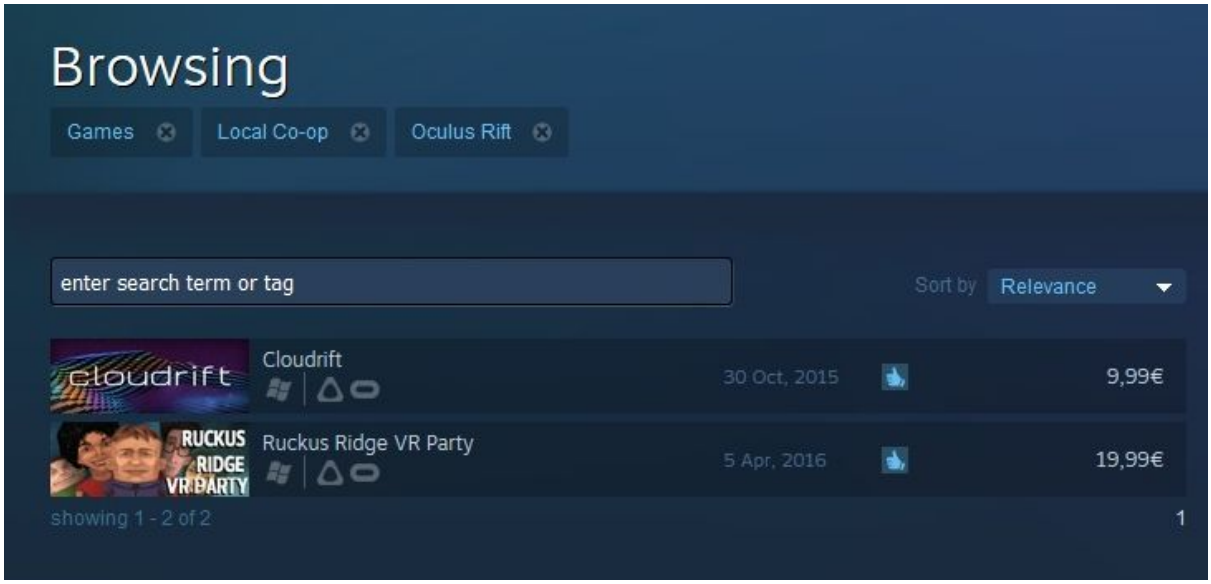

Image 1(Valve, 03 May 2016)

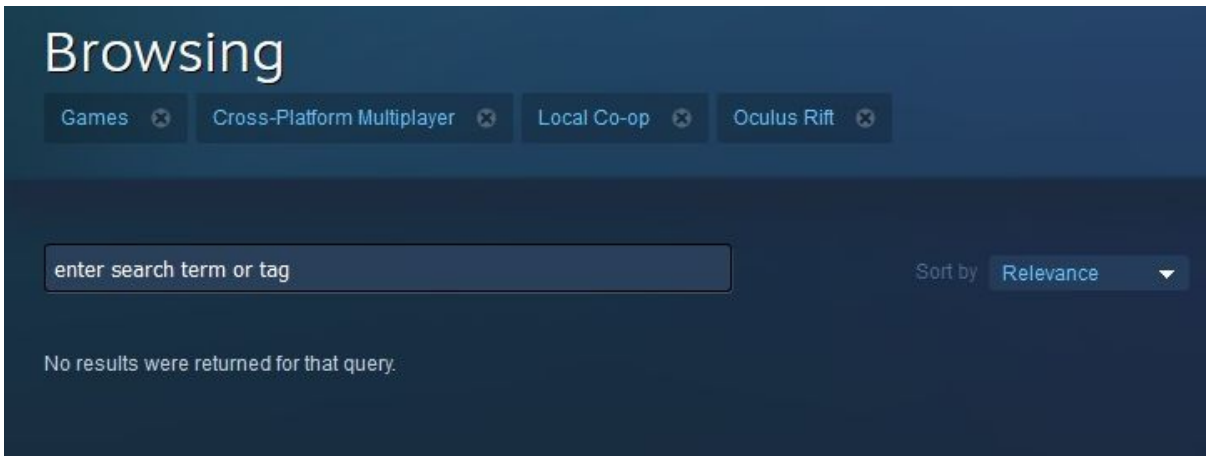

Image 2 (Valve, 03 May 2016)

Another unique concept in the product is the cooperation on a spaceship. The aspect of flying a spaceship and group micromanaging a crew is rare among games at the current time, and a game such as this does not exist on virtual reality (Image 1) at this time. The product distinguishes itself from the other games by placing such a large focus on teamwork and interoperability instead of solo flying.

6. Target Timeframe and Budget

The timeframe given for the creation of the game from the first lines of code to finished product consists of eight sprints of one week each. During each sprint, we set new tasks, ranging from bugfixing to new implementations and we divide them among the five members.from the previous sprint. After the eight sprints have been completed, one additional week will exist for the team to test the product as a whole, as well as present a set of advertisement-based products and presentations of the final product to the client and other interested stakeholders.

The project will have no budget. The equipment available to us is provided either by the client (the Oculus Rift headset), open source software that is freely available, or software that the group has access to through educational licences given by the university.

Glossary

Oculus Rift: A virtual reality headset.

Steam - a game distribution platform developed by Valve, Inc in 2003.

Cross-platform multiplayer - a multiplayer game that is played using two or more different devices (e.g. a PC and mobile phone) that are connected to each other.

HUD - Abbreviation for heads-up display, user interface.

CheckStyle - A tool aimed at enforcing code style agreed upon by the project team.

PMD - A source code analyzing tool, which focuses on finding unused variables, unnecessary object creation, and many other of such checks.

OpenVR - an API and runtime that allows access to VR hardware.

API - Application Program Interface.

Cobertura - Tool that calculates and reports test coverage for Java programs.

FindBugs - A static analysis tool created for Java source code.

jUnit - A unit testing framework for Java.

Github - Website that provides online project hosting using Git.

References

Keep Talking and Nobody Explodes. http://www.keeptalkinggame.com/

Haughey, D. (2000). *MoSCoW Method*. Retrieved October, 28, 2012.

Moneta, G.B & Csikszentmihalyi, M. (1996). *Journal of personality*. Wiley Online Library

Dillon, R. (2010). *On the Way to Fun.* CRC Press.

Lamaree, F. D. (2002). *Game Design Perspective*. Hingham, MA, United States. Charles River Media.

Gousios, G., Pinzger, M., & Deursen, A. V. (2014, May). *An exploratory study of the pull-based software development model*. In *Proceedings of the 36th International Conference on Software Engineering* (pp. 345-355). ACM.

Valve (2016). Retrieved from: http://store.steampowered.com/search/?term=#sort_by=_ASC&category1=998&category3=24&vrsupport=102&page=1

Valve (2016). Retrieved from: http://store.steampowered.com/search/?term=#sort_by=_ASC&category1=998&category3=24%2C27&vrsupport=102&page=1

M. Anderson (2015, October). "*Technology Device Ownership: 2015.*" Pew Research Center.