# Product Vision (Draft)

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

**Group *PixelPerfect***

April 28, 2016

**Supervisor**
Dr. ir. Rafael Bidarra

**Teaching Assistents**
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 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 not in the 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 observer the Rift player or partake in a passive state.

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 wearer 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 could be diffused and nobody would blow up/lose the game.

The space freighter scenario is also a cooperative experience, in which players without the virtual reality headset must “fix” problems in the ship in order to prevent it from malfunctioning and killing the crew on board. The Oculus Rift wearer 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 demographic, an outline of the product features (split using the MoSCoW method), a roadmap for the development of the game, the definition of done for sprints and product features, and user stories for features that must exist in the game.

# 2. Target Audience

## 2.1 Objective

The objective of this game is to make sure that a spaceship remains operational during a journey 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, who uses the Oculus Rift to observe the surroundings, provide assistance to the rest of the crew, and guide the ship. The other players, who are divided in specific roles (such as engineer, gunner, or navigator), have to perform these tasks to avoid consequences that result in damages dealt to the ship or the cargo. 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 destination, points are awarded and a victory can be attained.

## 2.2 Target Demographic

The target demographic of the game is any player with no physical disabilities that impact gameplay (such as problems perceiving visual content or epilepsy) in a context that allows for simple communication without boundaries. An example of such a situation could be a party or a convention. The player total must be five players. 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 mobile device in order to play the game. The Oculus Rift must have the client installed, and all phones must have the respective applications installed.

# 3. Customer Needs

The product to be developed concerns a computer game that fits within a given context. This context prescribes that, as a main purpose, the game must allow a group of colocated people to be collectibely entertained. This means the experience should yield some sense of amusement or fulfillment for each of the involved participants. Examples of contexts the game can be made suitable are a party or a large public space.

In addition there are some constraints on the how the participants are organized and by which means they play the game, which are:

1. One of the players lives immersed in a virtual world that is perceived by this player through an Oculus Rift device.
2. The players in the company of the forementioned player are able to influence the virtual world, and thus are able to play along.
3. All players are to be actively engaged in the gameplay, sharing an equally fun experience.
4. All players are located in the same physical space.
5. The solution exclusively requires utilization of low-budget technology.

A unique concept for computer game was devised conforming to these rather vague needs. For details on this concept we refer the reader to the Game Design document. Both the functional and non-functional requirements for the implementation of this specific solution are listed and prioritized in the next section.

# 4. Requirements

## 4.1 Functional Requirements

In this section, there will be posted the wanted features described using the MoSCoW method. The features are categorised in 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.
- **Would haves:** features that are most likely not going to be implemented, but can be made during a continuation of the project.

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 the game will take must be represented by the main timer.
5. The maximum round timer must be set 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.
10. Each active event must be listed in a log visible to the captain.
11. The events must occur at random time intervals.
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/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.
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.
23. The game must terminate the game and return to the main menu when a player quits the game.

### 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 occuring should increase.
2. Different types of events should have different time limits and damage values.
3. Events and damages should be visually indicated at the location where they take place in the ship (inside the virtual ship perceived by the captain).
4. On succesfully responding to 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 seen by the players.
8. Events should be parameterized, i.e. the captain is presented some random (e.g. numeric) 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 microgames 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, new (more challenging) types of events canbe 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 obstables (e.g. asteroid / debris).
4. The frequency and types of events occuring 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 becan 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 Won't 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 becan be different for each possible route through space.
4. Players, excluding the Oculus Rift player, 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 (technical) 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 java library.
3. The game must be able to interface with an Oculus Rift DK2 hardware device.
4. Git must be used as source control software.
5. The software project must be hosted publicly GitHub.
6. A Continuous Integration server must be used.
7. The software is developed according to a pull-based development model.

# 5. Product Uniqueness

The product under development utilizes a combination of non-unique features to develop a game that is, at this point in time, unique for the Oculus Rift and virtual reality in general. 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

Most of the games currently on the Oculus Rift are single-player games. Our product entertains a small group of players that are using the Oculus Rift and want to play with the device together. Most people these days have a mobile phone which makes it a lot easier to play together if one person has the Oculus Rift. The widespread availability of mobile phones 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.

Another unique concept in the product is the cooperation on a space ship. Many games that feature spaceships in virtual reality are focussed on dogfighting with them or flying them on your own.  The aspect of flying a spaceship and micromanaging a crew is unique and at the current time does not exist on virtual reality.  The product distinguishes itself 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, in which a  sprint has new tasks that will be allocated and divided among the five members based on the progress and the results from the previous sprint. After the 8 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 presentation of the final product to the client and other interested stakeholders.

The project will have no budget, meaning that no additional material, with the exception of the hooks and engine components that are provided by the client at the start of the project, will be available for the team to use. The equipment available is provided by the client (the Oculus Rift headset), open source software that is freely available, and software that the group has access to through educational liscences, such as *Unreal* and *Cryengine* assets.