

Product Planning

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	<i>J.W.D.Alderliesten@student.tudelft.nl</i>
Floris Doolaard	4362748	<i>F.P.Doolaard@student.tudelft.nl</i>
Dmitry Malarev	4345274	<i>D.R.Malarev@student.tudelft.nl</i>
Jesse Tilro	4368142	<i>J.Tilro@student.tudelft.nl</i>
Wouter Zirkzee	4398858	<i>W.J.Zirkzee@student.tudelft.nl</i>

Table of Contents

- [1. Introduction](#)
- [2. Product](#)
 - [2.1 High-level product backlog](#)
 - [2.2 Roadmap](#)
- [3. Product backlog](#)
 - [3.1 User stories of features](#)
 - [3.2 User stories of defects](#)
 - [3.3 User stories of know-how acquisition](#)
 - [3.4 Initial release plan](#)
- [4. Definition of Done](#)
 - [4.1 Backlog Items](#)
 - [4.2 Sprint](#)
 - [4.3 Release](#)
- [5. Glossary](#)

1. Introduction

Pixelperfect was assembled and created 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 requested by the university to develop and release a virtual reality game for the Oculus Rift platform.

A few needs were specified for the product. The game has to be entertaining and interactive for all players involved. One player has to be using an Oculus Rift, while up to four other players have to be able to influence the game through their own compatible Android devices. The gameplay must be enjoyable to the players and must keep all players actively involved, meaning that the players on the phones must also be actively communicating with the Oculus Rift player.

This document contains the long-term planning for PixelPerfect's game project development. It includes the product, product backlog and the definition of done that will be used for this project. The first chapter gives a high level overview of the product and its development process. The backlog gives a high-level outline of the sprint plans for each week for the eight weeks of development, in which user stories and desired features act as guiding beacons for their respective implementation. The definition of done defines when a piece of the software solution is considered to be done. The definition of done also defines when a feature may be added to the release branches.

2. Product

2.1 High-level product backlog

The developed game is one where a person using an Oculus Rift is seated in a virtual cockpit while they control a spaceship along with up to four others on their mobile devices. While the ship travels to another planet, specific problems may arise that require the attention of the pilot and their crew. The pilot needs to give tasks to other players, who will pose as the crew. The crew, who are playing on an Android device, will need to complete mini games to solve the problems. With strong communication and teamwork, together they can complete the mission of surviving the journey.

A game server, an Oculus Rift and multiple Android devices are required to allow the game to be played. The game server will connect the Oculus Rift and the Android devices together to allow for a multiplayer experience. The player wearing the Oculus Rift must be able to observe areas within the ship, which can be observed by moving the head around to move the game camera. The Oculus Rift player is also able to move the ship over the x and y-axis. The players that are not using the Oculus Rift must be able to process assignments and perform actions at the captain's discretion in order to solve problems and assist with the movement of the spacecraft.

2.2 Roadmap

Week 1

- Set up a game project using the JMonkeyEngine

Week 2 to 4

- Design the first level
- Develop first 3 minigames
- Events will occur at random time intervals

Week 5 to 6

- Score system
- Choosable path for captain
- Increase variety of minigames
- Multiple levels
- Implement visual representation of damage dealt to the ship

Week 7 to 8

- Random variables in minigames
- Pickup items
- Improve visuals

3. Product backlog

The user stories give us scenarios of users using the system. This will clarify the use of features and possible failures. The usual format is given by: "As a <role>, I want <action> so that <reaction>".

3.1 User stories of features

1. As an Oculus Rift player, I want to be able to look around in the cockpit of the spaceship using the Oculus Rift so that I can read the quests that are given.
2. As a mobile phone player, I want to be able to use my own phone so that I can play the game.
3. As a player, I want to be in the proximity of the other players so that I can communicate.
4. As a player, I want to be able to verbally communicate with the other players in the game so that I can complete the assignments given.
5. As an Oculus Rift player, I want to be able to control the main menu with my controller so that I can perform actions in the game
6. As a player, I want to be able to spectate the Oculus Rift player in the main menu so that I can follow what is currently being selected in the menu.

3.2 User stories of defects

1. As a player, I want to be disconnected from the game session when the connection cannot be verified.
2. As a mobile player, I want to be disconnected and have my session terminated when the Oculus Rift player ends the session or disconnects.
3. As the Oculus Rift player, I want to be notified when one of the mobile players disconnects from the session.

3.3 User stories of know-how acquisition

1. As a programmer, I need to know how to use the jMonkeyEngine.
2. As an artist, I need to know how to model meshes.
3. As an artist, I need to know how to create and map textures and export models using Blender.
4. As a technical artist, I need to know how to import and integrate models into the game.
5. As a programmer, I need to know how to use the OpenVR library, which provides the interface for the Oculus Rift DK2 hardware device.
6. As a programmer, I need to know how to use the jMonkeyEngine.
7. As an audio engineer, I need to know how to use audio software to create and alter sounds and music in the game.

3.4 Initial release plan

The initial plan consists of SCRUM-based intermediate version releases at the end of each sprint. Between these milestone releases we also have three major version releases: First Playable Version, Beta and Release, which will contain all built-up features from past sprints

Release	Milestone	Due Date
Sprint 1	Set up game in JMonkeyEngine Drafts of product vision and planning Draft architecture design	April 29, 2016
Sprint 2	Product Vision Product Plan	May 6, 2016
Sprint 3	Event back-end Event scheduling Basic Game View	May 13, 2016
Sprint 4	First level 3 Minigames Events occur at random time intervals	May 20, 2016
First Playable Version		May 20, 2016
Sprint 5	Multiple levels Steering of the spaceship	May 27, 2016
Sprint 6	Point system Increase variety of minigames	June 3, 2016
Beta		June 3, 2016
Sprint 7	Item pickups Improved visual representation	June 10, 2016
Sprint 8	Parameterization of minigames As you advance new challenges	June 17, 2016
Release		June 22, 2016

4. Definition of Done

The definition of done defines the requirements for when a user story, sprint, or release is complete. The agreed upon requirements for one of these items to be defined as done are split into appropriate categories below.

4.1 Backlog Items

An implementation of a user story can be considered as done when all the features of that user story have been implemented into the software product. In order for an implementation to be accepted it must hold to:

- At least 80% automatic test coverage where possible.
- Have a suite of automatic tests and have a set of manual tests that can be performed and verified.
- Contain no “PMD,” “Findbugs,” and “Checkstyle” violations.
- Pass Travis continuous integration.
- A user story must be reviewed by at least two members of the programming team in order to be considered for the main branch of the software product.

4.2 Sprint

A sprint is considered done when:

- All elements from the current sprint backlog have been either implemented or added to the sprint backlog of the next week in the case of a lack of time or problems.
- The release branch has been merged into the main software product
- The sprint reflection and the finalization and new sprint backlog have been approved.

4.3 Release

A release can be considered to be done when:

- All members of the team have reviewed the code submitted for release
- They have approved all the changes
- All tests pass
- No obvious bugs must exist in the product or code base for a release.

5. Glossary

Android - An operating system aimed at mobile devices, such as mobile phone.

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

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

Oculus Rift - A virtual reality headset.

PixelPerfect - The name of the project group developing the game.

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

SCRUM - A software development model that uses teams that focus together on a task in short cycles.

Version Release - A name given to a piece of software that has reached a milestone.