Product Planning (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
- 2. Product
    - 2.1 High-level product backlog
    - 2.2 Roadmap
- 3. Backlog
    - 3.1 User stories of features
    - 3.2 User stories of know-how acquisition
    - 3.3 Initial release plan
- 4. Definition of Done
    - 4.1 User Stories and Features
    - 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 Technical University of Delft. The team consists of five members and was requested to create, develop, and release a virtual reality game for the Oculus Rift platform.

This document contains the long-term planning of the PixelPerfect game project's development. The product roadmap gives an overview of the development process. The backlog gives a high-level outline of the sprint plans for each week of the ten 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 as done, and when it may be added to the release branches. A glossary gives an overview of important terms that must be understood to understand the project.

2. Product

2.1 High-level product backlog

As a player wearing the Oculus Rift I want to be able to use the Oculus Rift to look around in the cockpit of the aircraft and see where the aircraft is going and use my controls to move the spaceship.

As a player without an Oculus Rift I want to be able to use my own Android device to process assignments from the captain to assist with the ongoing maintenance of the spaceship by solving mini games.

2.2 Roadmap

Sprint #1:

- Write Drafts for Product Planning & Vision documents
- Pitch Concept
- Set up a game project using the JMonkeyEngine

Sprint #2:

- Write Final versions of Product Planning & Vision documents
- Write Final version of the Game Design document

Sprint #3:

- Write and submit first peer evaluation

Sprint #4:

- Design the first level
- Indications of health/damages/time left
- Develop first 3(?) mini games
- Events will occur at random time intervals

Sprint #5:

- Develop remaining mini games
- As the game progresses the frequency of events occurring should increase
- Option to choose multiple locations/planets for the ship to travel to

Sprint #6:

- Allow the ship to be move over a two-dimensional plane
- Different events should have different time limits and damage values
- Implement point system for successful missions

Sprint #7:

- Allow the players to pick up items during idle time
- Each path should provide different effects on the challenges provided
- Implement visual representation of damage dealt to the ship

Sprint #8:

- Implement parametrization of events
- As the game progresses, new (more challenging) types of events canbe introduced.

3. 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 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 to play the game.
3. As a player in general, I want to be able to verbally talk to the other players in the game so that I can complete the quests given.
4. As an Oculus Rift player, I want to be able to control the main menu with my controller.
5. As a player in general, I want to be able to follow 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 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, create and map textures and export models using programs such as Blender.
3. As a technical artist, I need to know how to import and integrate models into the game.
4. As a programmer, I need to know how to use the library providing the interface to the Oculus Rift DK2 hardware device.
5. As an audio engineer, I need to know how to use audio software to create and alter sounds and music in the game.

3.3 Initial release plan

The initial plan consists of SCRUM[1] intermediate version releases at the end of each sprint and three major version releases:

Release	**Due Date**
Sprint 1	April 29, 2016
Sprint 2	May 6, 2016
Sprint 3	May 13, 2016
Sprint 4	May 20, 2016
First Playable Version	*May 20, 2016*
Sprint 5	May 27, 2016
Sprint 6	June 3, 2016
Beta	*June 3, 2016*
Sprint 7	June 10, 2016
Sprint 8	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 User Stories and Features

An implementation of the 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 have at least 80% automatic test coverage where possible, and have a suite of automatic tests and have a set of manual tests that can be performed and verified. Upon request, a walkthrough demo must be able to be given by the creator to the team members or other relevant stakeholders, and this walkthrough demo must be completed without major issues or errors. 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. Once the backlog has been cleared, all automatic tests must pass with full certainty, and the automatic test coverage must be at around 80% for the portions of the product that can be tested and have been created by the development team. Once the release branch has been merged into the main software product, the sprint can be completed upon approval of the sprint reflection and the finalization and approval of the new sprint backlog.

4.3 Release

A release can be considered to be done when all members of the team have reviewed the code submitted for release, and when they have approved all the changes. The customer and any relevant stakeholders must also approve of the new release based on a demo given by the team. All tests must pass, and no clear or obvious bugs must exist in the product for a release.

5. Glossary

[1] http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf