Product planning, Carried Away

Names:	**Netids:**	**StudentIds:**
Luka Miljak	lmiljak	4368916
Marcel Kuipers	mekuipers	4389042
Damian Voorhout	dvoorhout	4388550
Nils Hullegien	nhullegien	4389069
Remi Flinterman	rflinterman	4362950

Table of Contents

1. Introduction

As a development team, we got the assignment to create a game with the Oculus Rift and other devices, which can be played by multiple players. The idea we came up with and are going to implement, is the following:

One player, wearing the Oculus Rift, is the commander sitting on top of a platform carried by four other players, who are using their phones to play. Together they have to stay on the path, avoid obstacles and defend against enemies.

The problem being dealt with right now, is that the product needs to be finished within a certain amount of time. Ten weeks to be exact. This would be the most ideal for the stakeholders. Time management is very important in a project. Tasks have to be divided, so that the game is being developed in the most efficient way. Scrum is very important for this, cause then there's always a functional product at the end of every sprint, and our priorities can always be adjusted if necessary. This document is made to let the stakeholders know our plan to achieve this goal.

Over the course the document, the following is addressed:

- A description of our product
- A roadmap, which gives a general idea about what has to be done each week
- The product backlog, created with user stories.
- The definition of done, which describes when a feature, a sprint and a release are considered to be done.

At the end of this document, there will be a glossary in which words will be described that are specific for our project that can be difficult to understand.

2. Product description

2.1 High-level product backlog

This game must be played with five players in total, one player wearing an Oculus Rift and four players having a smartphone that uses Android as operating system.
From now on, the Oculus Rift player will be called the commander and the android players will be called the carriers.
For the game, the pc will only be used to connect all players to the game and let the player with the Oculus Rift see the game, so the keyboard will not be used for controlling players.

When the game is started, the commander will see the environment, while the carriers will not see anything from the game. The commander will be on top of a platform, that is carried by the carriers. This platform will be moving forward at all time while in game.
The commander will, based on his perspective, give instructions to the carriers to move the platform in certain directions, or dodge obstacles that appear on the path the platform is moving. The carriers execute these instructions from the commander via their smartphone,

moving the platform can be done by tilting the smartphone and jumping or ducking can be done by moving the smartphone up or down, respectively.
Each carrier has a specific amount of health points, or hp, which will be visible on the application. Once a carrier has no more hp, the player will be knocked-out. This carrier can revive itself by consecutively tapping the smartphone. Once the carrier is revived, he/she can play on as normal.
When two carriers are knocked out at the same time, the platform will fall over and the game will be over. The platform can also fall over when the carriers tilt their smartphone too much. After a while, a checkpoint will be reached in game. The players can return to this platform, so that they don't have to start over from the beginning. After each checkpoint, the game will get more difficult.

2.2 Roadmap

In section an overview is given of how all **major** releases are scheduled. Some sprints are left out as there are no major plans for it as of yet.

Sprint 2
In this sprint the final game idea will be picked and thought out in depth. The game will be presented in a pitch presentation. After receiving feedback on the idea a draft version of the product planning and product vision documents are made, outlining how the following weeks are to be filled. An initial document describing the structure of the game is also delivered.
The following is released:

- A pitch presentation describing the game idea that was picked.
- A draft version of the product planning.
- A draft version of the product vision.
- An initial version of the product architecture document.

Sprint 3
In this sprint the game design, planning and vision are finalized. Final hardware and software choices are made and essential functionalities are implemented. A working Maven build is needed, to allow proper usage of tooling. A basic Android app will be made. Basic networking will be implemented.
The following is released:

- Final product vision document.
- Final product planning document.
- Final product design document.
- A basic Android app that builds with Maven.
- Networking functionality.

Sprint 4
Development now has really started. The first applications that make use of the Oculus Rift will be made this week. Core elements of the game will be implemented and components are connected.
The following is released:

- First jMonkey application that uses the Oculus Rift

Sprint 5
All components of the final game now have an initial version and are connected. The refinement of individual elements starts and a first playable version of the game is released.
The following is released:

- First playable version of the game.

Sprint 6
Players will be able to start a session over the internet. Focus is put on creating responsive controls and establishing basic gameplay. A presentation will be given about the game in its state at that moment.
The following is released:

- A presentation about the game.

Sprint 7
In this sprint a game version will be created that follows the game flow described in the design document. Level design, art design and sound design have are focused on. At the end of the week a beta version of the game is released.
The following is released:

- A beta version of the game.

Sprint 9
The game design is improved and will look close to its final form. The first version of the end report is created, as well as a script for the trailer of the game.
The following is released:

- Draft version of the final report.
- Script for the game trailer.

Sprint 10
In this sprint the game is released. The final report will be made, the final presentation will be given and a demo will be provided. This week finalizing the project is the only thing that is focussed on.
The following is released:

- Final Report.
- Final version of the game.
- Final presentation.

3. Product backlog

3.1 User stories of features

User Story	Estimated time (hours)	Priority (A-E)
As the *commander*, I want to be able to perceive the environment through an Oculus Rift device, so that I can immerse myself more in the game.	15.0	A
As the Oculus Rift player of this game, I want to have a simple UI for this game on my computer, so that I can actually start playing it.	10.0	A
As the user that is controlling the computer, I want to quickly be able to start the server from the UI, so that the Android users are able to connect to my game.	10.0	A
As an Android user wanting to play this game, I want to easily be able to join a server from a UI, so that everyone is able to connect quickly to the game and we can start playing as soon as possible.	.10.0	A
As a *carrier*, I want to carry the *commander* who is standing on top of an (unstable) platform that me together with my three friends can try keep balanced using the gyroscope from our Android device, so that I feel like I'm doing something challenging and by that experiencing fun.	30.0	A
As a *commander*, I want to fall over and get the game to end if the *carriers* fail to properly balance the platform I'm on, so that there is an actual catch to the game and something challenging.	10.0	A
As a *carrier*, I want us to be able to move forward on the map, so that we can increase our score and get the feeling of progress.	5.0	A
As *commander*, I want obstacles (for example a fallen tree) to appear in the map, so that I can warn the *carriers* (who can't see the obstacles) to avoid them and thereby experience a bigger challenge and by that experiencing more fun.	25.0	A
As a player of this game, I want the platform to become much more unstable once we failed to dodge an obstacle, so that there is an actual penalty for hitting an obstacle and it gives us a reason to actually dodge them.	20.0	A
As a *carrier*, I want to be able to jump and duck in the game using the gyroscope from my Android device, so that there are more ways to avoid certain obstacles and also allowed for a bigger variation in obstacles.	15.0	B

As a player of this game, I want a score bar shown that increases the further we progress in the map, so that we can use that number to compare this run to previous run and get a feeling of progress.	10.0	B
As a player of this game, I want our scores to be saved at the end of each game, so that I can take a look back at them and see how we progressed as a group.	5.0	B
As a player of this game, I want the environment to be infinitely long, so that me and my friends can continue to improve our score.	30.0	B
As a player of this game, I want the number of obstacles to increase as we progress further in the game, so that we don't get bored because the game is too easy and does not feel challenging.	10.0	B
As a player of this game, I want checkpoints after reaching certain scores and being able to go back to a checkpoint when we fail, so that we don't have to sit through the beginning of the game which is the easy part.	15.0	C
As a player of this game I want enemies spawning on the sides of the map that try to attack the *carriers*, so that the game feels even more challenging	10.0	C
As a *carrier*, I want to have health points that go down when an enemy manages to hit me and, so that I won't go down with just one hit.	5.0	C
As a *carrier*, I want to go down when my health points reach zero, so that I'm unable to balance the platform further and the platform becomes much more unstable, making the *commander* fall.	5.0	C
As a *carrier*, I want a sort of circle on my Android device, which when pressed at a certain angle of the origin of the circle, will make my character attempt to hit a nearby enemy (which I can not see, but the *commander* has to tell me the location of)	15.0	C
As a *carrier*, instead of immediately going down, I want to be put in a "falling state" first, which I can escape by shaking my Android device rapidly, so that I at least have some chance of redeeming myself.	10.0	D

3.2 User stories of know-how acquisition

User Story	Task	Estimated time (hours)
As a developer I want to learn how the JMonkey SDK works so I can work productively on the project	Learn how the JMonkey SDK works	10.0 * 5
As a developer I want to learn how the Oculus Rift works and how to integrate the functionality into the project so the Oculus works well in Carried Away	Learn how the Oculus Works and how to integrate its functionality into the game	10.0
As a developer I want to learn how to integrate android functionality into our project using the JMonkey Engine so the carriers functionality can be implemented	Learn how to integrate android support into the project	10.0
As a developer I want to learn how to make, implement and use models and animations via blender into a JMonkey project so the game world can be made in short time frame and still look good	Learn how to make models and animations and how to implement them into the project	20.0
As a developer I want to learn how to implement sound and music into the project to increase the immersion of the game	Learn how to implement sound and music into the project	10.0

4. Definition of Done

In our definition of done, we will define under which conditions we consider some part of our project “done”. We will do this for every feature, sprint and for the final release.

We consider a feature to be done when:

- The code and tests for the feature all compile and do not contain any errors.
- It is fully implemented and fully tested.
    - The testing is done through both unit testing and end-to-end testing.
    - All tests for that feature have to pass.
    - The test coverage of the feature is at least 80%, or a sufficient explanation of why that feature can’t be tested using unit / end-to-end tests.
- Both the code and the tests are fully documented with javadoc and comments.

We consider a sprint to be done when:

- The features defined in the previous backlog are all completed and tested.
- Unit tests are passed. That way we know for sure that there are no bugs in the code when the features are all merged.
- Integration with other features of a sprint doesn’t give any problems and still passes all tests.

We consider the releases to be done when:

- The code doesn't contain any errors.
- All code is completely documented with javadoc and comments.
- The sprint backlog is completely finished
- All of the tests pass, both unit and user tests (tested on at least 10 individuals who have no stake in the project in any way). Travis does not fail.
- The continuous integration does not give any errors.
- We reach a total test coverage of at least 80%.

5. Glossary

- **unit testing** is a software testing method by which individual units of source code, sets of one or more computer program modules are tested to determine whether they are fit for use.
- **Usability testing** or user testing is a technique used in user centered interaction design to evaluate a product by testing it on users.

6. Referentials

1. Kenneth, S. Rubin, "Essential Scrum: A Practical Guide to the Most Popular Agile Process"
2. Waters, K., http://www.allaboutagile.com/prioritization-using-moscow/