Product Planning
DuoDrive

Pim van den Bogaerdt, pvandenbogaerd, 4215516
Ramin Erfani, rsafarpourerfa, 4205502
Robert Luijendijk, rluijendijk, 4161467
Mourad el Maouchi, melmaouchi, 4204379
Kevin van Nes, kjmvannes, 4020871

May 15, 2014

TI2805 Contextproject, 2013/2014, TU Delft
Group 5, Computer Games
Version 2

Abstract

The product we are making is called DuoDrive, which is a racing game where people have to work together to win. This document will go over the planning that we have made for this product. The planning consists of a high-level product backlog, a roadmap, a list of features sorted according to the MoSCoW method, user stories and a Definition of Done.

1 Introduction

The product that we are making is a game. This is not a casual game, but a game where interaction with another player is vital for winning the game. However, winning is not the main goal that we want our players to achieve. We are making this game, because we want people to engage in a fun and social activity, while they are waiting somewhere for someone or something.
The concept of the game is as follows:
Essentially, our game is a race game, but the twist is that two people will control the car, instead of the usual one person per car. One person will be in control of steering and the other person will be in control of the throttle. The game will be played on two different screens. The person that is in control of steering has limited view and therefore he can not completely see where he is going. What this player can see, is how fast the car should be going. This information needs to be passed over to the player who controls the throttle, because the throttling player will not know how fast (or slow) he has to throttle. The throttling player can see where the steering player has to steer to and needs to inform the steering player about this. The lack of information on both sides forces the two players to cooperate and communicate if they want to reach the finish first (or at all).
To get this idea from scratch to a final product, we will need a proper planning. Firstly, we will talk about the product itself. Secondly, the product backlog will be described. Finally, we will talk about the definition of done. A glossary can be found at the end of the document.

2 Product

2.1 High-level product backlog

Our game basically consists of two parts: The *steering player* and the *throttling player*. The steering player has a limited view of the track and will therefore have a hard time steering. In order to steer adequately, this player has to cooperate with the throttling player, who has a complete overview of the track. However, this player can only accelerate or decelerate. As both players control the same car, they will have to be connected to a network.

Translating the above to a high-level product backlog gives us the following items:

- Create different interfaces for both players
- Create control events for both players
- Connect both players through a network.

A backlog containing all features that can be expected, along with the planning of their implementation, is listed in the section below.

2.2 Roadmap

This section of the report contains the features that we have designed. We have divided these features according to the *MoSCoW* model to give a brief overview of the features and their priorities. The features that must and should be implemented are planned into sprints. The sprint into which they are planned is stated between brackets.

2.2.1 Must-haves

The following features are essential and will be implemented:

- Race track and car (Sprint 1)
- Network server to achieve online player interaction (Sprint 2)
- Different controls for different roles (*steering player* and *throttling player*), including different User Interfaces (Sprint 4)
- Road sprites (including deceiving ones) (Sprint 1)
- Steering player must have minimal vision of the track (Sprint 1)
- Acceleration player is able to see an overview of the whole track (Sprint 3)
- Playable on devices that run on an Android operating system (Sprint 5)

2.2.2 Should-haves

The following features are not essential to achieve a minimal viable product, but will definitely improve the gameplay:

- Tag teams can match up (Sprint 6)
- Random tracks to remove learning factor (Sprint 5)
- Steering done using a *gyroscope* (Sprint 7)

2.2.3 Could-haves

The following features will only be implemented if time allows us to do so:

- *Ghosts* of other tag teams to see what position you are in.
- Position tracking during the game.
- Several car sprites
- Track times according to random track length.
- Drop objects on track to harass other players.

2.2.4 Won't-haves

These features will not be implemented as time will probably not allow us:

- A second mode: ranked mode
- Cross-platform possibilities

We will give a brief overview of the intended planning in the section below.

3 Product Backlog

In this section, a number of user stories will be described in regard to the following four points: features, defects, technical improvements and know-how acquisition. There will also be a section in which we will go over the initial release plan.

3.1 User stories of features

As a player,
Who controls the steer,
I can steer the car

As a player,
Who controls the throttle,
I can determine the speed of the car

As a player,
Who controls the steering,
I can see when the speed of the car needs adjustment

As a player,
Who controls the throttle,
I can see when the car needs to steer and where to

As a player,
Who wants to start a game,
If there is no game available,
I can start a new game

As a player,
Who wants to start a game,
And there is a game available,
I can join that game

3.2 User stories of know-how acquisition

As a player who controls the throttle,
If I see a corner coming up,
I know I have to tell my teammate to steer,
Or else we will make mistakes,
Leading to losing the game

As a player who controls the steering,
If I see that we need to slow down/speed up,
I know that I need to tell my teammate to adjust the speed,
Or else we will make mistakes,
Leading to losing the game

3.3 Initial release plan

For our project we will have some milestones.

- network implementation finished
- gyroscope control
- different controls for each player

We will see these features as milestones for our project in this specific order. The Minimal Release Features(MRF) consist of:

- The release must have a track with a car, that has limited vision
- It must also have network capabilities
- The car must have separate controls

If these features are not achieved, we can not release the game.

4 Definition of Done

The last section of this report will explain in detail when the final product can be considered as such. In particular, this section will handle the Definition of Done (DoD) of a feature, sprint and release.

4.1 Backlog Items

Within the backlog, an item is considered as done, if all the checklist points below are checked and considered to be true:

- Code complete and approved by lead programmer
- Code satisfies coding standards
- Unit tests written and executed
- Integration system test passed
- Documented

4.2 Sprint

A sprint is considered done, when all the checklist points below are checked and considered to be true:

- All sprint items considered done
- Application is tested globally, all unit tests pass
- Integration system test passed
- User tests passed

4.3 Releases

A release is considered done, when all the checklist points below are checked and considered to be true:

- The product should pass all unit tests
- Integration system test passed
- Interface looks as the product owner demanded.
- (End-)user tests passed
- Code documented and satisfies coding standards.

5 Glossary

Ghosts

You see another player's shape (ghost) in a transparent color. You can not collide with the ghost. It can be used i.e. to show how well your opponents are doing.

Gyroscope

Device used for measuring or maintaining orientation, often built into smartphones.

MoSCoW

Technique to reach a common understanding on the importance of each requirement. M = Must, S = Should, C = Could, W = Won't.

Steering Player

The player that can only steer left or right. This player can not accelerate or decelerate.

Throttling Player

The player that controls the throttle and can only accelerate or decelerate. This player can not steer.