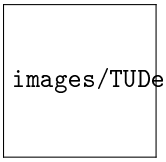
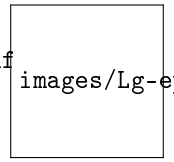
Context Project Group 8





Rob van Bekkum	4210816
Thijs Brands	4247132
Soheil Jahanshahi	4127617
Aidan Mauricio	4195175
Joost van Oorschot	4220471

Product Planning

Contents

1 Introduction

Creating a product is not as simple as having an idea of what to build and then starting to build the product. This holds especially true for an inherently complex product such as a software product. During the short time that Software Engineering has been around, software engineers have struggled to find a fitting way to develop software and to plan a software project.

During the last few years, a new software engineering mantra arose: agile. Agile teaches us iterative, short-cycled development, which alleviates risk and makes for a more maintainable product. During the context project we will be using the agile method Scrum.

Scrum development consists of a number of short development cycles called sprints. The result of each sprint is a working product. During the context project we will use Scrum to structure our development process. This document specifies our planning for the project, and it lists for each sprint what we want to finish at the end of that sprint. We also define a number of user stories that specify what use cases we want to implement in the game. Additionally, we will give a definition of when we consider work done.

2 Product

2.1 High-level Product Backlog

The following lists contains all high level features that we will implement:

- Drivable taxi with collisions
- Intuitive touch controls for the car
- Tiled, extensible world map
- A navigator screen
- Power-ups for the navigator
- 8-player multiplayer
- Teams of two (driver-navigator pair)
- Passengers to pick up and deliver
- Stealing passengers by ramming another car

2.2 Road map

This section specifies the road map of the development of Taxi Trouble.

Design phase

- Brainstorming on game concepts.
- Choosing three best game concepts.
- Picking one concept out of the three to develop.

Sprint 1

In this sprint we create the first working prototype.

- Drivable car with collisions.
- World map.
- On-screen buttons for controlling the car.

Sprint 2
During this sprint we focus on refactoring and research, as well as implementing the navigator screen.

- Implementing the navigator screen.
- Refactoring all the code.
- Researching multiplayer theory and implementation details.

Sprint 3
During this sprint we implement the basics of the multiplayer aspect of the game.

- Matchmaking system with a lobby and invites.
- Team creation (pairing up navigators with drivers).
- First start on realtime multiplayer (syncing car locations).

Sprint 4
During this sprint we start implementing the gameplay.

- Implementing basic gameplay (picking up passengers, dropping them off).
- Refining the multiplayer (reducing lag).

Sprint 5
During this sprint we implement advanced gameplay mechanics.

- Implementing advanded gameplay (car collisions and passenger stealing)
- Further refining of the multiplayer.

Sprint 6
During this sprint we fine tune and polish the game.

- Fine tuning of game balance.
- Improvement of game graphics.
- Fixing any problems that still linger.
- If possible: implement additional features.

Sprint 7
During this sprint we do user tests and we finalize the game.

- User testing.
- Finalizing end product.

3 User stories

As a user
Given that I have installed the app
And I have started the app
Then I can create a new game lobby

As a user
Given that I have created a new game lobby
Then other users can join my game lobby

As a user

Given that I am in a game lobby
And there is at least one other user in the lobby
Then I can team up with that user

As a driver
Given that the game has started
Then I can drive around the car

As a navigator
Given that the game has started
Then I can navigate around the map

As a driver
Given that I collide with a building
Then the car will bounce back

As a driver
Given that I stop next to a passenger
Then the passenger will enter my car

As a driver
Given that I drop off the passenger at the correct location
Then I will receive a point

As a driver
Given that I do not have a passenger
And I crash my car into a car that does have a passenger
Then I steal the passenger of the other car

As a user
Given that I am in the team with the most points at the end of a game
Then I win that game

As a navigator
Given that there is a power-up on the map
Then I can use that power-up to influence the game.

4 Definition of done

The final point that this document will reflect on is the definition that we have given to when certain aspects of development are 'done'. The aspects of development that we will focus one in this section are when a feature, a sprint, and a release are done.

A feature is considered done when the following requirements have been met. Firstly, the feature has been implemented as specified in the user story and it is implemented correctly, as verified by unit tests. Secondly, the implementation of the feature is well documented by comments and JavaDoc. If these first requirements have been met, the feature can be integrated and if the continuous integration server returns a positive result, the feature will be merged. Once all previously mentioned requirements have been met, and the feature is merged, the feature is considered done.

A sprint is considered done, when all features of the sprint backlog are done, as specified in the previous

paragraph. This means that all unit tests and the continuous integration test should pass. Additionally we will play the game to make sure that no hidden bugs are in the code, and that all user stories of the sprint backlog work as specified.

A release, or rather the release of the end product, is considered done when all of the following requirements have been met. All features and sprints must be done as specified in the previous paragraphs. Also, user tests must have been performed to make sure that the user is satisfied with the product, and that the product meets the requirements of the user. Additional requirements are a positive result from the test performed by SIG, and gameplay that is considered fun by the user.

5 Glossary

app
: The app what is installed on the users device and it contains the main menu, the game lobby and the actual game.

game
: The game is the actual game play, e.g. driving around in a car and navigating.

lobby
: A multiplayer matchmaking lobby. Essentially a room that fills up with players that will start a game together.

driver
: A sub-role of the user role. The driver is in charge of driving the car.

navigator
: A sub-role of the suer role. The navigator's job is to set out a route an to provide useful navigation information to the driver.

SIG
: The Software Improvement Group. The company that will test if our code conforms to Software Engineering standards.