Context Project
Computer Games Group 8

Rob van Bekkum	rvanbekkum	4210816
Thijs Brands	tlmbrands	4247132
Soheil Jahanshahi	sjahanshahi	4127617
Aidan Mauricio	amauricio	4195175
Joost van Oorschot	jjevanoorschot	4220471

Date: 24th June 2014

Final Report

1 Introduction

At the beginning of the project we were given a problem to design a solution for. The problem was that in many circumstances, a whole bunch of people have to stay together in the same place for a considerable amount of time. Often, however, these people do not even know each other, but they still stay together at the same location. A game would make some of these situations bearable, e.g. when there is waiting time involved. After playing a game in such a situation it can also give the user a sense of e.g. achievement when a user gets a new high score or solves a difficult puzzle.
For our project we were assigned to design a game to entertain such large groups of people and fulfill our end-user's need. The requirements from the end-user were that the game must be original, i.e. the game concepts must be unique. The game must also be creative, fun to play and also let people interact with each other through e.g. verbal communication with each other.
As a solution to this context problem we developed a game called Taxi Trouble and it will be introduced in the next chapter.

This document is about the developed, implemeted and validated software product. In Chapter 2 it will provide an overview of developed and implemented software product. Chapter 3 will describe the developed functionalities. An HCI module that was realized for the user interaction with the developed solution is provided in Chapter 4. In Chapter 5 an evaluation of the functional modules and the product in its entirety, including the failure analysis will be provided. And at the end of the document in Chapter 6 an outlook will be provided regarding the possible improvements in the future and the strategy to achieve these improvements.

2 Overview

What we would like to include in this section is:

1. Brief description of the game (not elaborate as in the product vision)
2. Give an overview of all of the functionalities that are included and implemented in the final version of the game:
   (a) **Driver and navigator screens of the game.** Tell something about the roles and different views corresponding to those roles in the game.

(b) **Teams of drivers and navigators**: Description of in which way teams are formed, i.e. each team consists of a driver and navigator.

(c) **Taxis**: Controllable by a driver and collidable with other taxis.

(d) **Passengers**: pickup, drop-off and stealing

(e) **Power-ups**: can be picked up by the driver and activated by navigator, there are different types

(f) **Game Head Up Display**: displays team, score and time left in the game

(g) **Sound effects**: different sound effects for different events

(h) **Menu Screen**: when the game starts, you can either start the game or check out the leaderboard

(i) **Multiplayer implementation**: invitations via Google account, player can set the preferred number of other players, team score is submitted on the leaderboard when the game ends.

3 Functionalities

In this section the important functionalities and key features that are developed for game prototype will be explained, each functionaly is important part of the game and make user to experience an intresting and entertaining game, these are as follows:

Visual Aspect

Visual styles relates to how game appears to the player. It focuses on the appearance of the main game world and components used to enhance the game for the player. They are explained in this section:

- Driver's screen has myopic 2d view of the map where it makes driver's sight limited such that it is impossible to win the game without cooperation.There are touchscreen controls that is used by driver to control the taxi's acceleration and direction. Driver can see on this screen several entities that are in the game to interact with as well as non-interaction objects.

- Navigator's Screen has the 2d zoom-able screen in top view and Navigator gets the whole map overview. There is a power-up button where is controlled by Navigator and can be activated at the right moment by him/her.

- Game Head Up Display
  It represents following items on the screen:

  - displays team : is a textual representation of team ID.
  - score : achieved score of each team
  - time left in the game : count down timer indication time left for taxi to drop passenger at his/her destination.

- Menu Screen
  This is the main screen whenever you launch the game on your android device, features in this screen is listed here:

  - Play button: when play button is pressed the player is guided to the lobby room, from there player either can find random people or invite their friends from Google+. From there user will chose number of participants team to play with.
  - Leader-board: player can view the top scorers of all participants in the game.

Game Model

Game model is the main component of the game, These components are designed and developed in order to influence the game state while events is triggered by user and they are as follows:

- Teams
  - Teams is made of two participants where each will takes a role in the game; Either as an driver or Navigator. After all the required teams(from 2-4 teams) are created the game begins.
- Taxi
  - Taxi is a solid object , it has wheels that controls steering and acceleration which is controlled by driver.
  - When two taxis collide with each other, collision is detected and natural reflection of the taxi body upon collision is performed.
- Passenger
  - Passenger is an entity where are spawned in random locations in the map.
  - There are different looks of a passenger(girl, boy, man)
  - each passenger is spawned in random location ready to be picked up by taxi.
  - passengers can be stolen by other taxi while on their way to destination.
  - Destination is assigned to each passenger which is dropped off by taxi in the game.
  - when passenger is picked up by taxi, drop off timer is started and in order to achieve score taxi need to drop passenger off before drop off timer ends.
- Power-ups
  - There are three types of power-ups(invincibility, speed boost, increase drop off timer)
  - Invincibility power up is made when taxi wants to protect the passenger from getting stolen.
  - Speed boost when activated will increase the acceleration of taxi from 10 second
  - Increase drop off powerup will extra 10 second to the time that needs to be dropped off by passenger

Sound

SoundFX is an integral part of the any game. Good sound effect is create an impact on user gaming experience. They are designed to absorb the player in to the virtual game world, making the game more entertaining and satisfying and get a good overall feel of the game when events occuring at the time of game play. Sound effect is activated at the time when events such as taxi collision, passenger pickup/drop off and stealing of the passenger happens.

Multiplayer

One of the primary requirement design of multi-player for this game is to maintain a consistent and develop a shared sense of the virtual space among some numbers of player. The architecture that is chosen is peer to peer(p2p) architectures where necessary messages will be broad-casted by each player.

- Multiplayer
  - Multiplayer is formed via invitation of other participants
  - player can set the preferred number of other players to play with the boundary is from 2 to 4 teams
  - team score is submitted on the leader-board when the game ends.
  - all teams are concurrently updated on the events that are happening inside the game environment.

4 Interaction Design

After seven weeks of development we conducted a user test among visitors of the TU Delft Science Center. This section will describe the Interaction Design aspects of the usability evaluation that we conducted there. Firstly, we will discuss our evaluation methods and what part of the system we tested. Secondly, we will give an overview of how the testing was done, by discussing the setting and location of the user test, a description of the users that tested the game, and the methods that we used during the test. Lastly, we will give a summary of our findings.

For the usability evaluation, we have chosen to use the empirical 'experiment' practice. An experiment was best suited for our user test, because we wanted to observe users interact with the game in order to discover flaws in the game, and to identify gameplay elements that were not considered fun. A big aspect of Taxi Trouble is communication, which resulted in users practicing Think Aloud without being asked by us. This was very useful for our evaluation of the user test, as it gave us a lot of information about how the users perceived the game.

The usability evaluation was done right before the release of the beta release of Taxi Trouble, so that we could incorporate our findings into the beta version. This means that we evaluated the usability of the alpha version of Taxi Trouble. The alpha version was missing a lot of features compared to the final version, but it was stable enough to conduct a user test with.

For the setting of the user test we chose the TU Delft Science Center. The Science Center is a good location for testing because it receives a lot of visitors that fit into the user demographic of Taxi Trouble, and there are sufficient facilities for conducting a user test. We were appointed a large, open room, with two racing chairs in the center of the room. The racing chairs made the user test more fun for the younger users, and fit the theme of Taxi Trouble well.

The users that tested Taxi Trouble ranged in age from 8 to 24. We took great care during the user tests with the younger users. The parents of the younger users were present during the user tests at all times and we made sure to mention that the users could stop the test at any time.

The user test was performed in groups of two users. At the start of the test we asked the users to sit next to each-other and to pick a role in the game (either navigator or driver). Then, we let the users play the game for about five minutes. During this time we logged what the users said to each-other and we did not interact with the users. After playing the game, we interviewed the users about the game, asking open questions about the art style of the game, the controllability of the game, and about the gameplay.

We did a formative evaluation of the test resuslts and we will give a summary of our findings in this section. Overall, the users were very pleased with our game. Even though all users stated that they found the controllability of the game sufficient, we noticed that most users struggled to understand what the buttons to control the car meant, and they had to take some time to figure that out by trial-and-error. This resulted in us creating buttons that mimicked the look of an actual gas pedal and brake.

During the evaluation it became apparent that users did not immediately understand the navigator view. To make the navigator screen more understandable, we created a HUD that shows to which team the navigator belongs.

Another thing that we found out, is that when a taxi is carrying a passenger, it is hard for the users to distinguish the front of the taxi from the back. In response to this we created new sprites for the taxi that clearly show the difference between the front and back of the car.

The results of the usability evaluation helped us identify a number of flaws, which allowed us to fix them. Additionally, we were able to pinpoint which gameplay features the users liked and which features they disliked.

5 Evaluation of Functional Modules

What we would like to include in this section is:

1. Table with overall evaluation results from user tests.

2. Evaluation of individual functional modules (with more specific feedback we got from the users):
    (a) **Driver controls**
    (b) **Navigator controls**
    (c) **Picking up and dropping of passengers**
    (d) **Stealing passengers**
    (e) **Power-ups**
    (f) **Game Head Up Display**
    (g) **Sound effects**
    (h) **Menu Screen**
    (i) **Multiplayer implementation**

6 Outlook

After delivering a product that satisfies our expectations and our end user's needs there is always room for improvement. Taxi Trouble is playable and lots of fun to play, but it still contains a few bugs. Also there are tons of features that didn't make the first release of the game which could still be implemented in the future. Implementing these will greatly increase the quality and the experience of the user who plays the game.

Bugs

A bug is a flaw in the software. The two bugs that are present in Taxi Trouble are namely:

1. When the Host of the game locks his phone or minimizes the game all messages stop getting through meaning that the features of the game won't work until the host resumes the game again.
2. If you want to restart the game to play again you need to completely kill the app or the game will not start correctly anymore.

Solving these bugs will greatly increase the quality and the playability of the game. A few strategies to solve these bugs are:

1. Implementing a pause function. When someone lowers or locks his phone the game should pause. Pausing is not always the answer here. If someone locks his phone or lowers the game they might have had to leave, so implementing a leave function is also viable. But if the host leaves the game then a function should be implemented to switch hosts otherwise no messages will be getting through and the game stops working.
2. A viable strategy for this is implementing a restart function that returns you to the start menu of the game or to the lobby.

Features

Implementing a new feature's difficulty depends on the structure of your software and how much the developer understands this structure. That being said a new developer that can understand the structure of Taxi Trouble will have no problem implementing a few of the features mentioned below. A few ideas of features that we had are:

1. Choosing who you want to be in team with. Right now the game has an auto pick feature which randomly assigns people in a team.

2. Adding more collidable objects e.g. traffic, walking pedestrians, cones, etc. Adding more collidables to the map will increase the challenge of the game and the focus a player has to the game.

3. Cops. Adding this feature will make the game more immersive and realistic.

4. Health for the taxi. Colliding with objects will decrease the health of the taxi. When the taxi's health is decreased there can be a lot of side-effects e.g. the taxi's speed gets decreased, turning radius is increased, etc.

5. More powerups. A few new powerups that can be implemented are e.g. increasing the health of the taxi, slowing down other taxis, calling cops on other taxis, etc.

Adding these new feature will increase the experience a user has with the game and will ultimately make it more fun and challenging to play. A few ways to implement these features are:

1. Implement a choose a team function which puts you in a lobby where a user can switch between teams.

2. Collidables can be fixed on the map or their spawnpoints can be defined on the map. After that you need to define a collision detector for them. If you want to add traffic one must add some kind of agent that can control the traffic so that they don't collide in any objects and stay on the streets. For walking pedestrians one can extend the passenger class and add a walking animition for this. The same counts here for pedestrians as for traffic. The pedestrians may not collide with other objects and they have to stay off the streets.

3. A viable strategy to implement walking pedestrians is extending the Passenger class and adding a new collision detection to it and adding a walking animition to the passenger. They must also have some kind of intelligence to not walk into other objects or onto the middle of the streets.

4. Just like implementing traffic, one must define an agent to control the cops so that they can chase after the taxi.

5. Everytime a taxi collides with an objects a function must be called to decrease the taxi's health.

6. Powerups were implemented using the strategy pattern, so implementing these is just applying the Powerup-Behaviour to the new powerup and adding its new features.

As you can see, implementing some of these features can be pretty easy to do, but there are also a few where it is pretty difficult to implement, e.g. agents for traffic and cops. And a lot of these features depend on one another which is why we did not have enough time to add these to our first realese.