horizontal line

Grand Traffic Auto

Functional Specifications

**14th April 2019**

# Project Team

|  |  |
| --- | --- |
| **Name** | **Student Number** |
| Robert Pennefather | 21511164 |
| Daniel Gonzalez | 22024722 |
| Martin Porebski | 21498791 |
| Jacqueline Soon | 21719676 |

# 

# 

# 1.0 OVERVIEW

## 1.1. Introduction

Due to the increasing complexity of transportation, people now have many different options to choose from during everyday commutes. With the advancement of GPS and navigation technology, software applications can now inform people on the fastest and cheapest ways to get from point A to point B. To understand how people make navigation decisions in complex situations, the client wishes to analyze data in an experimental economics scheme. In past research, decision-based data is collected using paper based surveys. These traditional methods often require participants to provide decision-based answers about situations without having experienced the scenario firsthand. Decision-based data can be collected more accurately if participants are answering questions in response to real-life simulated events.

The client has requested to develop a game which simulates a realistic transportation scenario. The game will gather information regarding the specific choice a passenger makes in complex transportation environments. Ideally this project will provide a digitised and improved way to run surveys, in replacement for paper based ones.

Our functional requirements will focus on the use case of developing the architecture, components and mechanics for the autonomous vehicle simulator. This simulation will be gamified in order to be used in a psychology experiment lab, where volunteers can play it. Their responses of using an autonomous vehicle, including controlled nuisances that affect the time, cost and ambience of the travel experience, will be recorded in a survey that the client will be able to analyse later.

## 1.2 Project Vision

The client has requested that our team develop a game which can be used to perform experiments with/on players. The client has requested that the final product have several different experimental use cases. The three types of experiments covered are:

1. **Autonomous Vehicles Simulation** - Create a 3D World to simulate the real world experience of using an autonomous vehicle for the players. This simulation will then be used to perform a survey to record the players’ reactions and thoughts on the experience.
2. **Prisoner’s Dilemma** - 2D/3D game world with different test scenarios. Testing game theory and Braess’s paradox using a multiplayer game where everyone has to move from the same points A to B. The Game records how users change their route knowing they are competing against each other. Different segments of roads change their travel time based on how many people use it.
3. **Human Controlled Vehicles and Imperfect Information** - 3D game world, testing imperfect information. Players drive their vehicle from point A to B in a single player world. Testing whether or not people use their own knowledge of the road network or trust a simulated Google Maps.

Due to the limitations of time and resources for the project, the team is going to initially focus on developing an architecture to support the three games but specifically develop use case one (Autonomous Vehicles Simulation). If time permits the team will then explore use case one in further detail and attempt to deliver a working MVP.

**Experiment 1 Details - Autonomous Vehicle/Ride Sharing Simulation**

The client wants to build a game that simulates an autonomous vehicle or ride sharing experience. The game will feature services similar to those provided by the transportation company Uber. Players will be prompted to make a choice out of the following options:

* **Uber:** on demand transport.
* **Uber Pool:** share the ride and save money.
* **Uber Express Pool:** walk a certain distance, share ride and then save even more money.

Players will receive a starting monetary allowance, and the cost of each decision or ride will be deducted from their allowance. There will be 12 choice scenarios, all of which offer the same three ride options but with varying prices. In each game, players will play through 6 of the 12 scenarios. If a player selects Uber Express Pool, the game is not required to simulate walking, instead players will be asked to physically walk around outside of the game.

# 2.0 GOALS

The project team aims to achieve the following goals, in order of importance:

1. **Meet the clients acceptance criteria:** The client can confidently use the game platform to run successful research experiments:
   1. Game can be setup and run with no bugs or crash issues
   2. Game gathers accurate experimental data with useful context that aligns with the clients research goals.
   3. The client is able to process the data results in a meaningful way.
2. **Software architecture & documentation supports future development:** documentation should be detailed enough to support the understanding of future developers. Architecture design should allow for future developers to add out of scope features.
3. **Follows excellent software development practices**: team intends to develop skills in agile project management, use appropriate testing methods and frameworks, use continuous integration and version control to improve development.
4. **Delivery of the final product by the expected deadline:** complete project work on time in accordance with the project plan, such that the final product will be delivered as expected by the project deadline.

# 3.0 ACCEPTANCE CRITERIA

As discussed with the client, the final product should meet the following criteria:

* The functional requirements up to and including Prototype Three are completed as described in section 4.
* Performance - the game should support at least 50 players (subject to server capacity).
* Stable system - no lagging or performance issues including bugs and crashes.
* Quality - the in-game animation and graphics looks and feels realistic.
* Output data - all required data is stored as described in section 4.2.1 Sessions. The format of the output is easy to analyse.
* Input data - all required input variables are loaded into sessions as described in section 4.2.1 Sessions. The format of the input is easy to access and modify.
* Interface - user-friendly GUI, the game should be intuitive and require no training to play.
* Architecture - supportive of future development as considered in section 6.0 Out Of Scope.
* Documentation - in-code comments and summaries are provided from which overall documentation can be auto-generated.

# 4.0 FUNCTIONAL REQUIREMENTS

To facilitate this iterative progression of requirements the team has split the functional requirements of the project into three separate *prototypes*.

**Prototype One** will be to familiarise the team with the languages, frameworks and workflow we will be using for the rest of the project.   
**Prototype Two** will develop the architecture of the game, the team will create the entire flow of the game from start to finish without the major components of the actual game. This will include things like the login screen, the lobby creation and config file, and a database to record survey information and analytical information from in the game.   
**Prototype Three** will develop the simulated experience of using an autonomous vehicle, this will mean adding the logic for the self driving car routings as well as the option of picking up new passengers.

## 

## 4.1 Prototype One - Basic Game Framework

The aim of the first prototype is to set up the overall architecture of the system. It will focus on the basic mechanics of the game in which the player can drive around in a “toyworld”, which is a simple 3D world that will consist of sample (placeholder) buildings and streets.

### Session

* Supports single player sessions to setup the building blocks of the overall architecture, this will be expanded to a multiplayer system in further prototypes.
* Sessions are offline (local machine) since the focus is on the interactions of the physical assets (models).

### Point of View

* Player experiences first person point of view. The camera will align with the players “eyes” and “follows” players movement as they drive. If the player is in the car, the camera will remain stationary relative to the car, however it will follow the movements of the vehicle.
* Limited camera rotation of 180 degrees to simulate the line of vision in a vehicle. The user can pan horizontally in game to change camera rotation.

### 3D World

* 3D landscape is a toyworld that includes simple buildings, roads, traffic lights and other cars. Development of a basic toyworld serves as a a test ground for the game’s physical interactions and collisions.
* Invisible barriers should exist at the sides of roads to prevent players from driving off-road and crashing into buildings.

### Physics

* Assign rigidbodys to assets: buildings, cars, invisible barriers. This will allow to apply physics to the models in the world, such as, for example gravity or collision detection.
* Force-based driving - a physics-based system to control the vehicles through steering, accelerating and braking. This will allow for natural interaction with the rest of the models.
* Collision detection - detect when appropriate assets (e.g. cars on roads) are in contact and fire an appropriate event:
  + Using an invisible box around vehicles to detect if they are going to collide and stop them before the visible assets collide.
* Car wheel animation - the rotation of the wheels has to be animated according to the speed of the vehicle
* Steering wheel animation - rotate it in relation to the direction of the car or the player controls.

## 4.2 Prototype Two - Architecture

The inclusion of sessions is the main objective of prototype two. Users will be able to Log in; Admin will use a config file to modify different variables and create the game sessions, while Players will join active sessions made by Admin. The results of the session will be recorded for the client.

### Sessions

* Login screen
  + Admin login to sessions screen.
  + User login to a session (game) - users need to be able to join active sessions with the session code that they will receive from an admin.
* Session screen visible only to the admin.
  + Lists current active sessions and for each game it display details about number of players in it as well the session code.
  + Option to create new session.
    - The admin will need to create a config file (JSON file with settings options).
  + Option to close an existing session.
* Ability to import the session config file. This will read the selected by the user file and load the following options to the program as session variables:
  + Vehicle - different kind of AV will have distinct characteristics that affect their interaction with the world and the player (velocity, number of seats available, type of vehicle).
  + Array of route configurations. Each route will be presented as a choice to the user and needs contain the following information:
    - Distance of the route
    - Predicted time to the destination
    - Actual time to the destination
    - Total cost of the trip
    - Money saved - optional field that will be displayed to the users in an AV only as an option to pick up other users and save some money (but lose some time)
    - Quality of information - the information presented to the user can be:
      * 100% accurate
      * Distorted
      * Random and inaccurate
  + Change values that affect the function for the point system, the formula for the point system will include a set of values which can be weighted and scaled based on the configuration file values. The values include:
    - Base point value
    - Time
    - Money
    - Routes completed

### Data Gathering & Analytics

* Store analytics in JSON - this type of file does a relationship between the results, which will be easier to query.
* Survey at the start to record basic player information, stored in database
  + Gender
  + Age/ Age Group
  + Occupation
  + Optional Income Groups
  + Culture/Background, Ethnic Groups
  + Education
* Add ability for user to make very simple decisions in game and record in database
* Survey at the end of the simulation about game, information stored in database. Placeholder survey questions until game developed
  + Attitude
  + How likely to use autonomous vehicles
  + What they think of autonomous vehicles
  + Comments

## 4.3 Prototype Three - Autonomous Vehicle Game Logic

Focus on a realistic world environment where other vehicles create congestion and affect the routing of the players’ vehicle and give a score system based on the added variety of choices the player can make. In this prototype the player will also have to wait for the vehicle to arrive, adding waiting and response times to the survey and the scoring system.

### 3D World

* Add physical assets: civilians, autonomous vehicle
* Add civilians to populate autonomous vehicle when they’re picked up, based on capacity of vehicle
* Add moving vehicle bots to the roads

### Logic

* Create autonomous cars, logic for driving around world
* Logic for optimising routes for autonomous cars
* Introduction scene where player is outside car waiting for transport.
* Decision making options for the user to select to pick up more passengers, reducing fare and increasing travel time
* Addition of point system, based on some combination of money spent, time taken and objectives reached

### Data Gathering & Analytics

* Record game play logic and store in database
  + Response time for replying to pickups
  + How many people are picked up
  + Record pickup time/money change and user response
  + Final point score for user
* Add survey between each route and record answers in database
* Finalise survey questions

# 5.0 NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements gathered are key to consider when implementing all of the prototypes described in section 4.0.

## 5.1 Flexible System

The overall architecture of the system has to be flexible due to the experimental and iterative nature of this project. As requested by the client, it has to enable the rapid prototyping methodology which will make it easier to add any related requirements in the future.

Furthermore, this project is the initial phase of a larger system. The client aims to gradually develop a larger, complex system that is beyond the scope of the assigned resources and timeframe. However, the overall framework has to be designed with this in mind. Hence, the framework has to be flexible and able to accommodate the features that will be desired in the future. This is described in more detail in section 6.

## 5.2 Documentation

A comprehensive, technical documentation has to be provided upon the delivery of the project. It has to be sufficient enough for any competent developers to be able to continue the development of the system in the future. Hence, the documentation will have to be:

* Clear, consistent and well-structured.
* Provide a technical overview.
* Provide any necessary details such as functions’ inputs, outputs and side effects.

## 5.3 Deployment & Maintenance

The game will be required to run on multiple client machines. Hence, it is crucial to develop and implement a maintenance strategy. The game and any updates to it will have to be deployed automatically as it is unrealistic to manually install the software on hundreds of client machines.

## 5.4 Performance

The system will have to be able to handle at least 50 players simultaneously, given that the provided machines are powerful enough. The game has to perform well with no visible connection or frame lags. This is important as it will help players immerse themselves into the game and will result in a better simulation of “real-life”.

The system has to be stable. A suitable testing strategy has to be implemented to determine that no run-time crashes will occur. It also has to ensure that system components and functions perform as expected.

## 5.5 Game Design

The game is expected to simulate real-life experience. Hence, the driving has to feel intuitive and follow “real-life” rules and physics.

The game has to be as close to reality as possible, car movement and driving controls should be portray a driving experience close to one in real life.

## 5.6 Inputs & Control

User inputs are expected to be a keyboard and mouse.

* Driving & steering - arrow or WASD keys.
* Point of view control - mouse movements with a limited view.
* Option selection - mouse left click.

## 5.7 Interface

User interface has to be simple and intuitive to navigate. It has to have a consistent design.

In-game graphics (models, textures etc.) should be as close to reality as possible. However, within the scope of this project it is more relevant to achieve a working, consistent design.

## 

## 5.8 Users

The system has to account for two types of users with different privileges: administrators and players.

The administrators need to be able to login to their admin account and then create, monitor and close sessions. Each session with the same or varying configurations file depending on the administrator.

The players need to be able to login, answer a basic pre-game survey, join one of the active open sessions, play the game by making decisions about their route, and answer post-game survey questions based on their experiences in the game.

## 5.9 Language, Globalisation & Localisation

At this point the client only plans to use the system to perform experiments in a controlled environment in psychology labs. These sessions will solely be run by the client and the participants will be required to speak English. Changes to the game’s steering wheel position, and which side of the road the car drives on have been omitted due to the fact the experiment will be only run in Australia.

As the system will only be used in this controlled environment there are no foreseeable issues related to language, globalisation or localisation.

# 6.0 OUT OF SCOPE

The client aims to continue the development of this project after the team has completed working on it. With this in mind, the entire system has to be considered including features that has been agreed to be outside of the scope of this project. Hence, it will allow on designing a suitable and flexible architecture that can be expanded as according to clients wishes.

Furthermore, if the requirements outlined in section 4 are completed to a satisfactory level before the delivery date, the below list can be considered as bonus features. Hence, the requirements out of the scope were prioritised with a $100 analysis. This is because it clearly identifies which “bonus” features are the most important and how much value would be gained from implementing them.

|  |  |
| --- | --- |
| * MiniMap.   + 2D Google Maps style top down map of the world.   + Use this minimap to display route of autonomous vehicle to destination and additionally display any additional pickups the player may choose to make. Update the minimap accordingly such that new route takes into account all passengers travel destinations.   + Due to the testing required to create efficient, intelligent rerouting calculations, this requirements falls in the out of scope section. | $30 |
| * Multiplayer framework.   + The building blocks for a multiplayer game would have already been developed as mentioned in section 4. However, due the large complexity associated with development of a server, client and reliable communication between them, this requirement falls outside of the scope of the project. | $20 |
| * NPCs   + Automated players or vehicles that will simulate the real-life experience as well as create traffic and congestion.   + The framework will be designed in a way to allow for an easy integration of NPCs. However, due to additional overhead associated with designing of AI and testing, this requirement is too large. | $15 |
| * Allowing players to exit out of vehicles and walk.   + Walking would result in more combinations of options when deciding on the best travel route. It would also contribute to the “realistic” feel of the world. However, it is not essential to the architecture of the project as it is a modular task. Animating a realistic walk would require advanced models. Furthermore, the physical engine would have to be expanded to facilitate the new feature. Hence, due to the time required to deliver it, walking falls out of the scope. | $7 |
| * Adjust session configurations mid-session.   + It would allow to measure a wider spectrum of how the environment impacts the decisions that the players make.   + However, it requires the completion of the multiplayer framework first. | $7 |
| * Measuring congestion of traffic   + It only makes sense to measure the congestion if the multiplayer and NPCs are implemented. | $5 |
| * Use case 3 - Human Controlled Vehicles and Imperfect Information   + In this scenario players would have an option of driving through the city.   + This would require a further development of     - user controls     - more advanced physics engine     - simulating traffic     - modeling and coding traffic lights, lamp posts, daytime and nighttime settings     - some of the above requirements including NPCs and multiplayer   + However, framework will be designed in a way that will allow on easy integration of this use case. | $5 |
| * Allow admin to manually create roadblocks and/or construction work to create artificial delays as well as distracting events. | $4 |
| * Ability for players to chat with other players and/or bots in game   + This is only possible after the multiplayer framework has been implemented. | $2 |
| * More diverse assets   + This might require a help of a graphic designer to deliver a consistent-looking world. | $2 |
| * Maps.   + Generate new ways for administrator to create maps     - Procedurally randomly generated     - Map generation from Google map images     - 3D world map editor   + As one standard premade map will serve to work with the experiment, and the complexity of this requirement is extremely large this requirement falls outside the scope of the project. | $2 |
| * Use case 2 - Prisoner’s Dilemma   + This was described as a 2D decision-based mini-game. Hence it falls out of scope of this project as the focus is on delivering a 3D framework. | $1 |

# 7.0 USE CASES

## 7.1 Admin

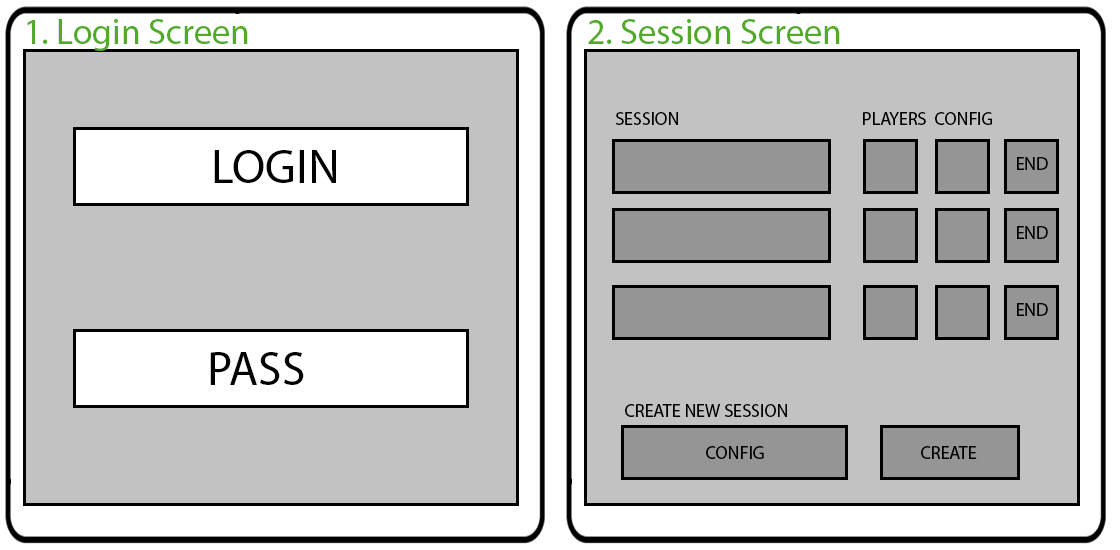
|  |
| --- |
| **Use Case Name:** Running a Session  **Summary:** Admin creates a new session with prespecified configuration, run the sessions and closes it when complete  **Actor:** Administrator, Player  **Precondition:** Configuration file created for session  **Description:**   1. Administrator opens website 2. System prompts for login details 3. Administrator logs in with credentials 4. System displays session screen 5. Administrator selects create session 6. System prompts for configuration file 7. Administrator uploads configuration file 8. Session is created, system provides session code 9. Administrator can provide session code for Players to access session 10. Administrator returns to session screen 11. System updates to show number of Players in session 12. Administrator closes session once all Players have finished in session 13. Administrator closes website   **Alternatives:**   * Administrator logins with incorrect login details, the system will throw an error describing incorrect login details * Administrator uploads incorrect file type/config file specifications, the system will throw an error describing problem and return to session screen * Administrator closes session when Players are still using it, remaining Players will be kicked from session * Administrator close website without closing session, session will continue to run and will be accessible for Players * Administrator creates another session, sessions will run simultaneously with two different joining codes * No Players join the session, no information about Player’s experiences are collected   **Postcondition:**  Database contains information about session created and each Player’s experience in the game including survey information and responses to game world |

## 7.2 Player

|  |
| --- |
| **Use Case Name:** Player plays the game  **Summary:** Player will enter game session, fill in pre-game form survey and play the game  **Actor:** Player, Administrator  **Precondition:** Administrator has created and is running at least one game session  **Description:**   1. Player opens website 2. System prompts for login details 3. Player fills in name and surname and presses submit 4. System displays pre-game survey 5. Player fills in form and presses submit 6. System allocates the Player to a current open session and displays sessions information 7. Player starts playing the game 8. System will prompt the Player with buttons to make route decisions throughout the game. Player can choose to change driving route or keep the same route with money and time incentives offered to the Player 9. Player selects a decision button 10. System applies the Player’s decision to the game world 11. System may prompt Player to make additional decisions (as specified in the session’s configuration file), if this is the case return to step 8. Otherwise the game will finish the route and the player will see the game end screen 12. System displays post-game survey 13. Player completes and submits final survey 14. Player closes website   **Alternatives:**   * Player incorrectly fills in user details form blank and submits, error is displayed asking player to fill in all fields correctly * Player incorrectly fills out one of the surveys and submits, error is displayed asking to correctly fill in incorrect section of survey and submit again * Player does not press any decision buttons, after a predefined time the decision will vanish and the game will continue taking the original traffic route with no changes * Player exits session early, no information about Player after point of exit will be stored * Administrator ends session whilst user is playing, error message will be displayed on players screen.   **Postcondition:**  Database contains information about Player’s experience in the game including survey information and responses to game world |

# 8.0 STORYBOARDS

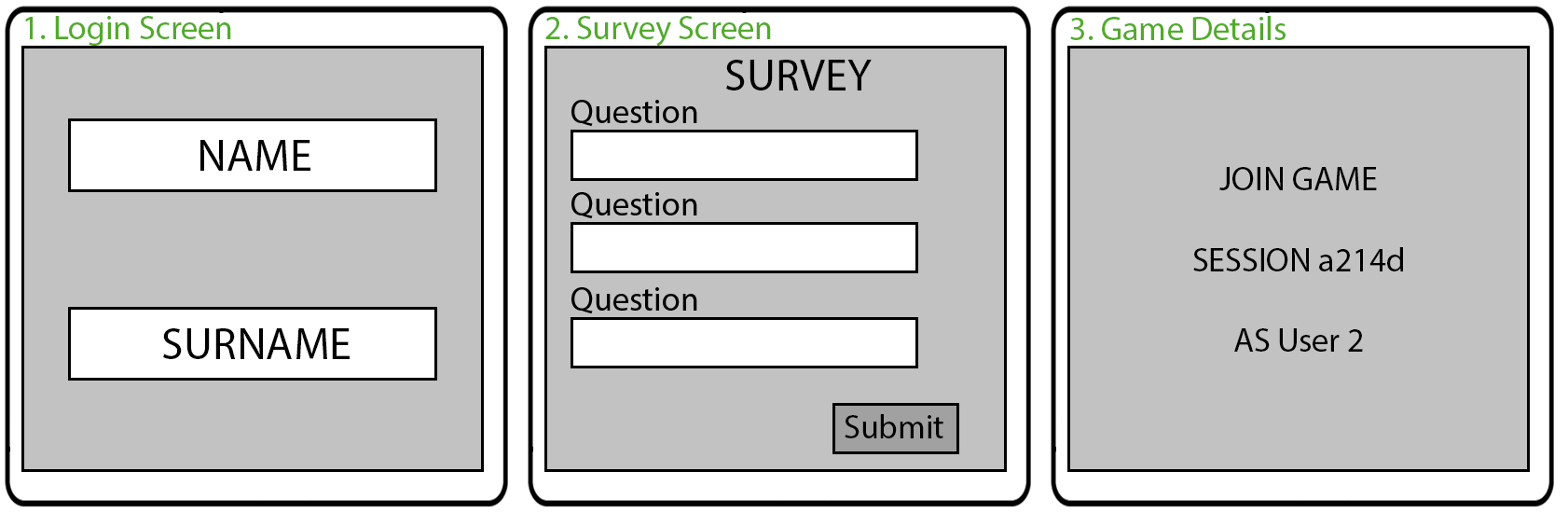
## 8.1 Admin



1. To create a new session, the admins have to access the website. They will be presented with a login screen.
2. Upon entering the correct details they will be redirected to a control panel which will contain a list of sessions. Each listing has the name of the session, how many players are in it, link to the config file used and an end button which will terminate the session. There will also be an option to create a new session by submitting a config file and submitting the create button.

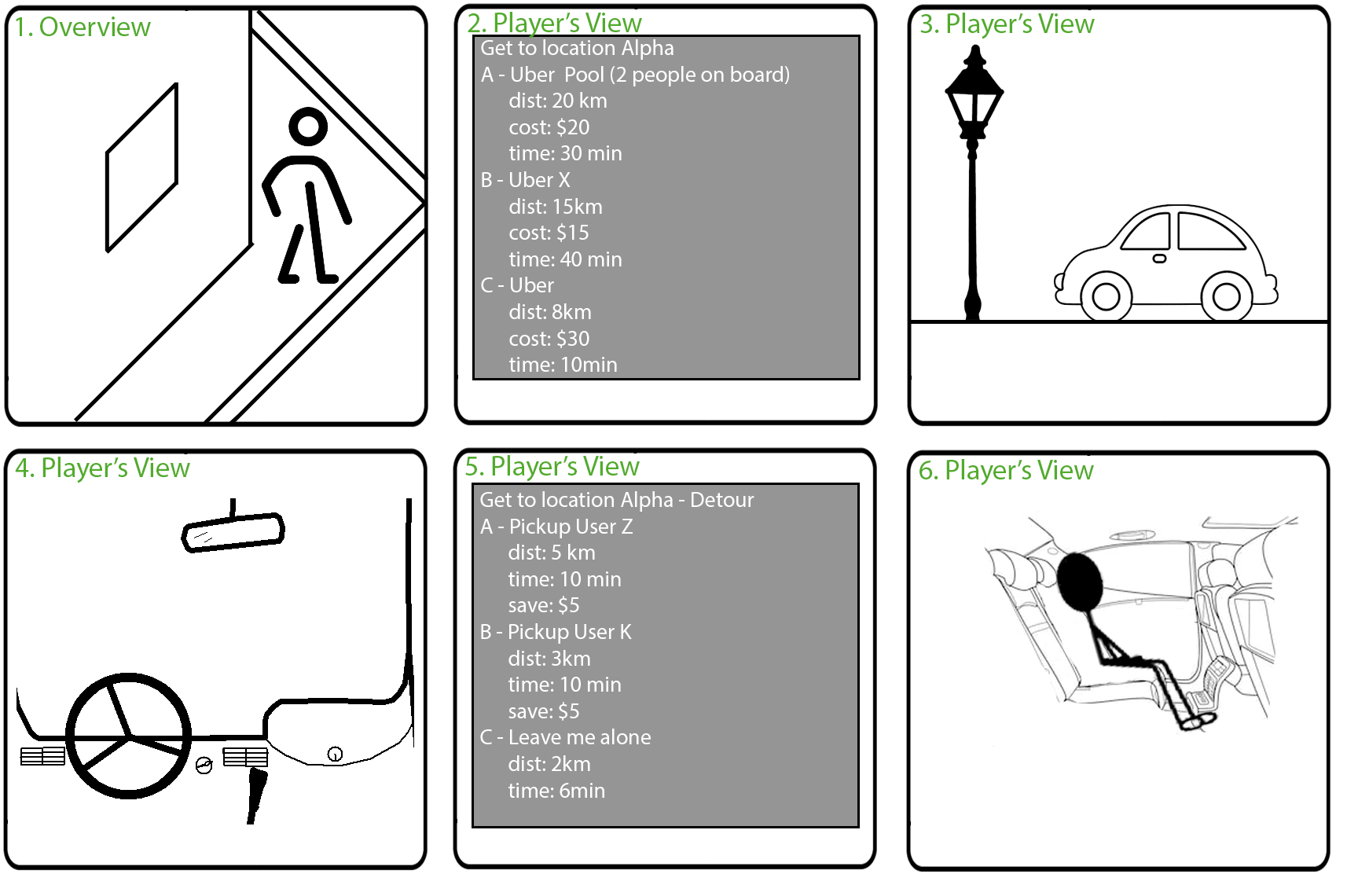
## 

## 8.2 Player



1. Before entering the game, the players have to access the website. They will be presented with a form which will require their name and surname.
2. They will then be required to complete a survey.
3. Upon completing the form the players will be presented with session details with which they can access a game session.

## 8.3 Playthrough



1. Upon entering the game, the players will start in an arbitrary location in the world.
2. The players will be tasked with getting to a different location in the world. They will be presented with a list of options. Those options and their details will be predefined in the config file as stated in section 4.2. The players will be able to pick one selection, based on their best judgment.
3. They will then wait until their method of transportation arrives.
4. Upon entering their method of transportation the players will see the interior of the vehicle. They will be able to see the outside world pass-by as the vehicle drives to their destination.
5. The players might be prompted with updates about their journey and will be forced to make more decisions. The details of the updates will be specified in the config file as stated in section 4.2.
6. They players will be able to see other passengers in their vehicles (if there are any).

# 9. GLOSSARY

AVs - Autonomous vehicles

NPC - Non-player characters.

AI - Players controlled by an algorithm that dictates their behaviour.

* User: person who is interacts with the game
* Player: person who is playing the game to provide information about their experience
* Administrator/Admin: person who controls and observes the game for experimental purposes