METROPOLIA UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Information Technologies

Sofia Ojala
Minh Nguyen
Elizaveta Vasiukova
Anna Hyttinen

REQUIREMENT SPECIFICATION DOCUMENT

Moomin’s Travelling Game

Game project TXL22S1-A. Group number 1.

December 2022

CONTENTS

1. INTRODUCTION

The purpose of this requirement specification document is to reflect what can be done with the game from the user's perspective. The document explains the core of the game and provides a description of the game process and its current state. The main goal of the document is to create a guide for users, teachers, developers, designers, potential investors, marketing and development teams and other parties interested in the project improvement on how the game was designed, provide explanation of the rules of the game and describe the target audience.

This document describes in detail the game's current state, vision, functional and quality requirements.

2. CURRENT STATE

2.1 First version of the Moomin's travelling game.

During the first studying trimester our group has developed the first version of Moomin's Travelling game. The game was designed using Python and could have been implemented only from localhost. The game implies that the gamer is Moomin character and requires the player to travel between Finnish airports to his girlfriend Snorkmaiden.

The purpose of the game is to fly to Snorkmaiden in the shortest way. The player cannot fly to the final destination straight away but must make 3 transits in 3 different airports.

In this part, we can provide more information about the technical part of the game. The game was designed for 2 players. It begins when the players see their current airport and destination airport. Both of the locations are fixed: the start is in Helsinki and ends in Ivalo. The first player who starts the game will be registered as player1 in the database. Second will be player2. Players must choose where to make transits.

In total there must be three transits and the choice is among the main airports in Finland. We have reduced the number of airports to 20 and have created a database for it with coordinates.

After the choice of transits is done, player1 will arrive at the final location and his travelling distance will be saved to the database. The distance travelled will be calculated using coordinates. After the first round of the game is over for player1, player2 starts the game. Game will run in the same way. In the end of the game results of the players are compared and the one who had the shortest distance WIN! GG! :)

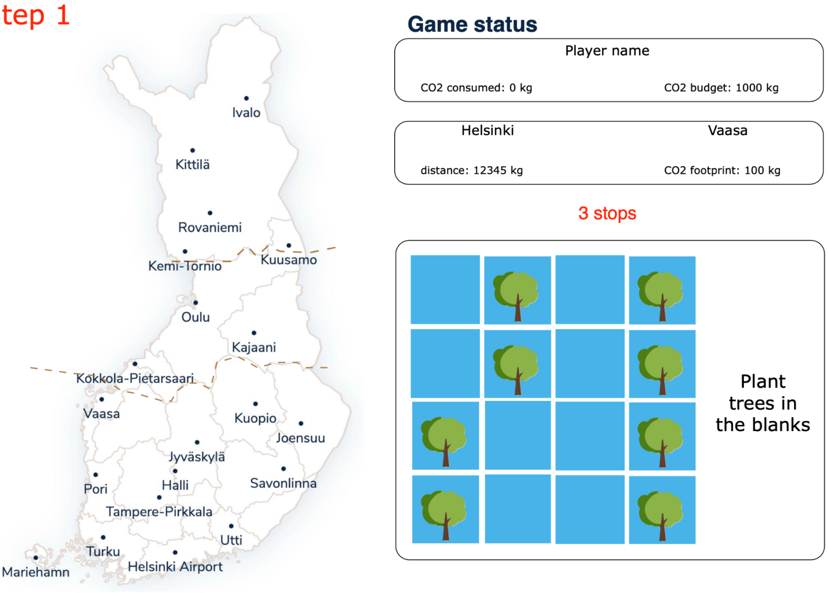
2.2 Second version of the game.

Second version of the game is called Moomin's Travelling Version 2.0. In this second period the game will be updated with a mini game and will be implemented on the website using backend, Java Script, HTML and CSS. Currently the team is developing a mini game inside of the main game in order to fulfil the requirement of considering CO2 consumption inside the game.

A mini game will be a memory game. It is quite a creative game where the user has to use memory skills in order to remember the card's position. In quite a short time a player will have to put cards in the same order as she/he saw it before they will be disappeared. If players order cards in the correct way then the amount of CO2 will be reduced by 1000kg.

Below you can see a model design of the future game. You can see the map with Finnish airports, life time weather forecast, your current amount of CO2 consumed, memory game with cards and also there will be music from Moomin Valley playing.

Step 1



Game status

Player name

CO2 consumed: 0 kg CO2 budget: 1000 kg

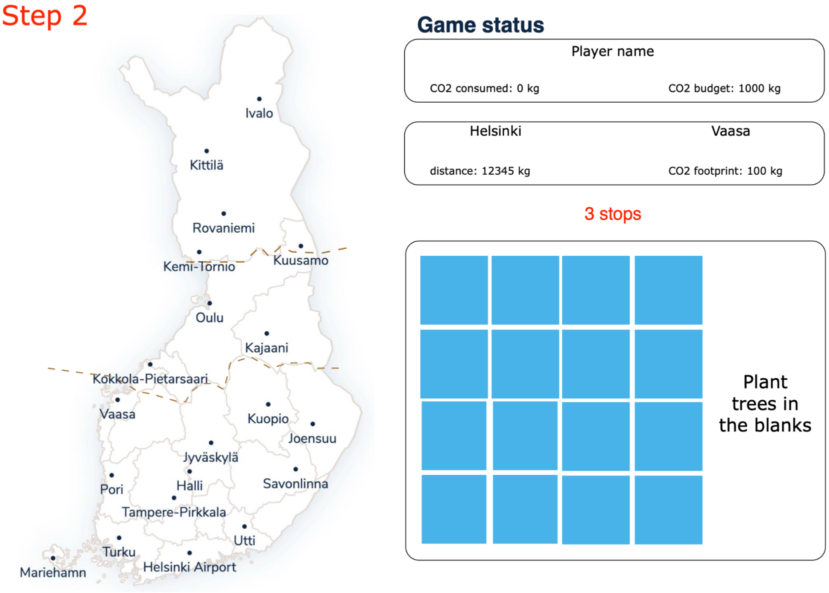
Helsinki Vaasa

distance: 12345 kg CO2 footprint: 100 kg

3 stops

Plant trees in the blanks

Step 2



Game status

Player name

CO2 consumed: 0 kg CO2 budget: 1000 kg

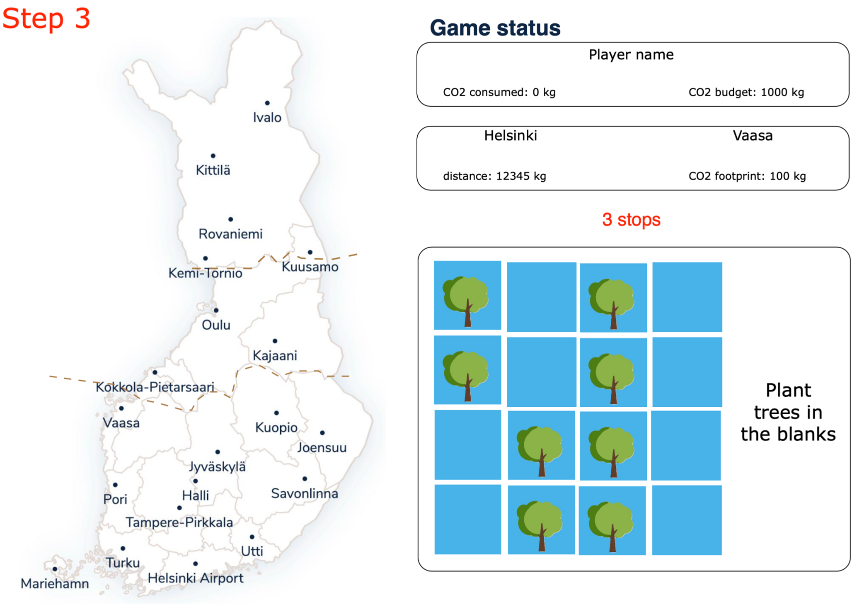
Helsinki Vaasa

distance: 12345 kg CO2 footprint: 100 kg

3 stops

Plant trees in the blanks

Step 3



Game status

Player name

CO2 consumed: 0 kg CO2 budget: 1000 kg

Helsinki Vaasa

distance: 12345 kg CO2 footprint: 100 kg

3 stops

Plant trees in the blanks

2.3 Time frame and group roles.

Currently we have about 5 weeks to implement a project. We will be working on our task that we have divided. Then we will combine our files using GitHub and implement and test the game.

Each team member has chosen the part of the programming to do. Besides, we have created visual elements for the game, such as winning, losing pictures and playing cards. At the final stage of the project we will work all together on the presentation slides.

3. VISION

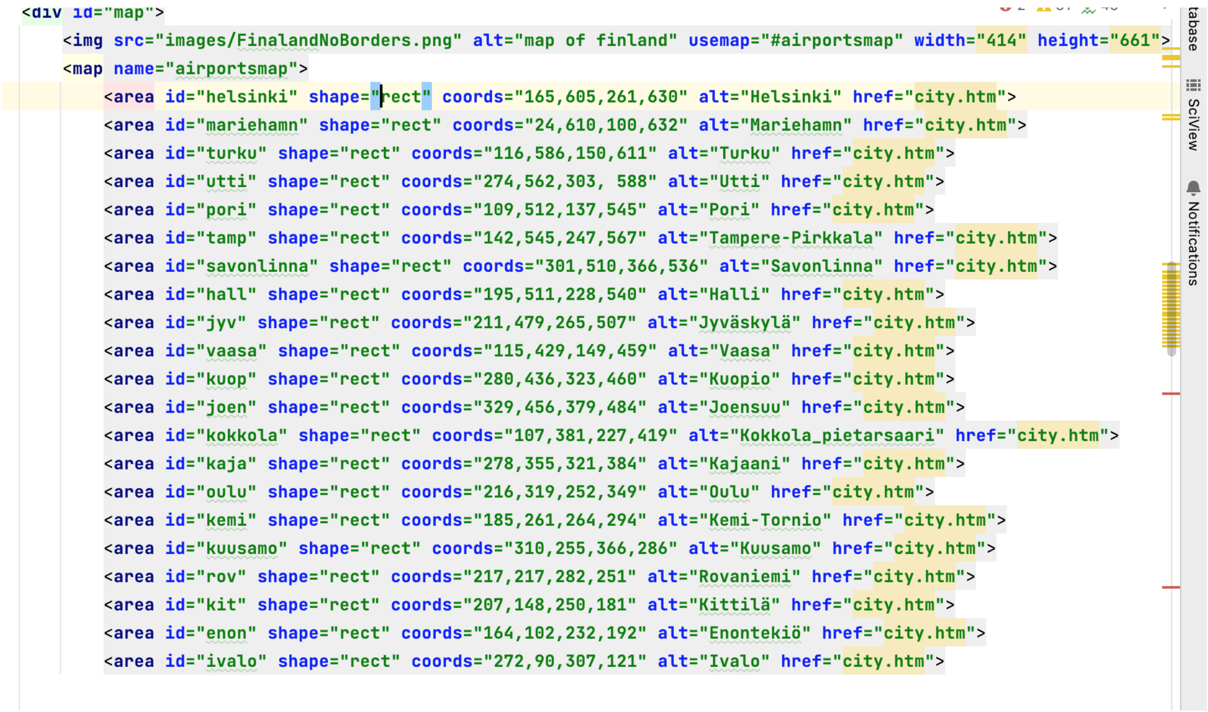
Moomin characters are very popular in Finland. The very first Moomin book called “The Moomins and the Great Flood” was written in 1945. Nowadays you can find a wide selection of Moomin products all over Finland. There is food with Moomin characters, cups, clothes, cartoons and more and more. Creating a game with Moomin characters will let people plunge into the Moomin’s world and help Moomins find each other by playing our game.

The vision is to create an interactive game with Moomin characters for all ages with many different challenging mini games and puzzles.

The game Moomin’s Travelling Version 2.0 is a multiple-player game. In the storyline, the players are helping Moomin get from Helsinki to Ivalo to his girlfriend Snorkmaiden. There are no direct flights, so he has to choose three transits. Since he wants to get to Snorkmaiden as soon as possible, he wants to choose the shortest possible route.

The technical idea of the game is to find the shortest distance from one fixed point (Helsinki) to the final fixed point (Ivalo) using three stops. Another challenge of the game is reducing CO2 consumption by playing a memory mini game at every stop.

By using coordinates we will create a code for calculating the distances. Map is a picture with Finnish airports that is presented on the website. Using a map in HTML we will add coordinates of the points (cities) on the map. First coordinates of points on the map are found in the editor GIMP and then they will be added to the HTML and then we will create clickable links on the map. Have a look at the code we have done below:



```
<div id="map">
    <img src="images/FinalandNoBorders.png" alt="map of finland" usemap="#airportsmap" width="414" height="661">
    <map name="airportsmap">
        <area id="helsinki" shape="rect" coords="165,605,261,630" alt="Helsinki" href="city.htm">
        <area id="mariehamn" shape="rect" coords="24,610,100,632" alt="Mariehamn" href="city.htm">
        <area id="turku" shape="rect" coords="116,586,150,611" alt="Turku" href="city.htm">
        <area id="utti" shape="rect" coords="274,562,303, 588" alt="Utti" href="city.htm">
        <area id="pori" shape="rect" coords="109,512,137,545" alt="Pori" href="city.htm">
        <area id="tamp" shape="rect" coords="142,545,247,567" alt="Tampere-Pirkkala" href="city.htm">
        <area id="savonlinna" shape="rect" coords="301,510,366,536" alt="Savonlinna" href="city.htm">
        <area id="hall" shape="rect" coords="195,511,228,540" alt="Halli" href="city.htm">
        <area id="jyv" shape="rect" coords="211,479,265,507" alt="Jyväskylä" href="city.htm">
        <area id="vaasa" shape="rect" coords="115,429,149,459" alt="Vaasa" href="city.htm">
        <area id="kuop" shape="rect" coords="280,436,323,460" alt="Kuopio" href="city.htm">
        <area id="joen" shape="rect" coords="329,456,379,484" alt="Joensuu" href="city.htm">
        <area id="kokkola" shape="rect" coords="107,381,227,419" alt="Kokkola_pietarsaari" href="city.htm">
        <area id="kaja" shape="rect" coords="278,355,321,384" alt="Kajaani" href="city.htm">
        <area id="oulu" shape="rect" coords="216,319,252,349" alt="Oulu" href="city.htm">
        <area id="kemi" shape="rect" coords="185,261,264,294" alt="Kemi-Tornio" href="city.htm">
        <area id="kuusamo" shape="rect" coords="310,255,366,286" alt="Kuusamo" href="city.htm">
        <area id="rov" shape="rect" coords="217,217,282,251" alt="Rovaniemi" href="city.htm">
        <area id="kit" shape="rect" coords="207,148,250,181" alt="Kittilä" href="city.htm">
        <area id="enon" shape="rect" coords="164,102,232,192" alt="Enontekiö" href="city.htm">
        <area id="ivalo" shape="rect" coords="272,90,307,121" alt="Ivalo" href="city.htm">
```

The distances are connected to the map picture presented on the webpage and to the Java Script.

Once both players have chosen the three airports, the lengths of both routes are calculated using the coordinates of the airports. The results are then compared, and the winner is the one who chose the shorter route and has the smallest CO2 consumed.

4. FUNCTIONAL REQUIREMENTS - THE GAMER STORY.

As the game page opens in the browser, the gamer is asked to type and submit a name.

Most of the upper part of the game screen includes the navigation for the web page, where the gamer can click ‘about’ to view the background story with a picture.

Back to the game screen, the gamer can see a music player, switch the music on, adjust the volume and switch it off any time.

On the left side the gamer can see a map of Finland with destinations to choose from. The right side of the screen holds information about current location with current weather data, distance flown so far and the carbon footprint information.

As the gamer chooses a city by clicking the map and the location. Weather data, distance and carbon footprint information gets updated as the gamer chooses a location.

The player can try to follow the story and fly from the starting point of Helsinki with three stops on the way, all the way to the ending point of Ivalo.

On the way to Ivalo, the gamer faces a minigame challenge, where the carbon footprint can be reduced by planting trees in a memory game fashion. Success in this minigame reduces the carbon footprint 1000 units.

5. QUALITY REQUIREMENTS.

The gameplay works in a browser. The web page carries a responsive feel to the player.

The background story is available through a navigation bar.

The user interface can be toyed around and tested by the gamer.

A theme music loop is available for the gamer: It has easy access during the gameplay. Play, stop and volume controls are available.

Besides a clickable map of Finland with a selection of destinations, there is a minigame that pop up with memory game-like logic in it, allowing the gamer to make a difference in the game world.

Game gives a thought to the environmental aspect of carbon emissions. The carbon footprint gets calculated and refreshed during every flight and after every successful minigame.

The game utilises a real time weather service API, which is being called after every click on a map and some weather data is shown about the chosen destination.