

# Flight Simulator Game Project

**Group: 3**

Team members:

*Hoang Phan Duc Hanh*

*Arijana Määttä*

*Adrian Garcia*

The purpose of this document is to give an overview of the Flight Simulator Game project and describe the implementation of the program. Target group of the game is players at the age of 12 onwards.

The report consists of general ideas and the purpose of the game, functional requirements and quality requirements.

### **General idea and the purpose of the game**

The aim of the single-player Flight Simulator Game project is to produce an entertaining game where the player gets to use an airplane to fly from a selected airport to Rovaniemi and use as little CO<sub>2</sub> consumption as possible. It is a text baseline and interactive game, allowing users to use the keyboard to engage throughout the game. Based on players' inputs the program will generate different outcomes.

The program begins with a short story. The purpose of the storyline is to raise the interest of the player in the game. The story tells that people who either do or do not believe in Santa Claus once a year take a plane to fly to Rovaniemi to either pull Santa's beard or hug him. To meet Santa Claus, players need to fly through 5 stops and keep the CO<sub>2</sub> consumption during the trip as low as possible.

In the beginning of the game, to fly to the first destination, the player has consumed CO<sub>2</sub> of 5000 units, per default. In order to win the game, he needs to pass through 5 stops within a budget of 10 000 CO<sub>2</sub> units. Once the player's CO<sub>2</sub> score is equal or over 10 000 units, the game is over.

In every destination, the player needs to answer a question related to environment and sustainability. Depending on his answer, the weather in the game will change and affect CO<sub>2</sub> score. If the answer is correct, his CO<sub>2</sub> consumption will be deducted and vice versa.

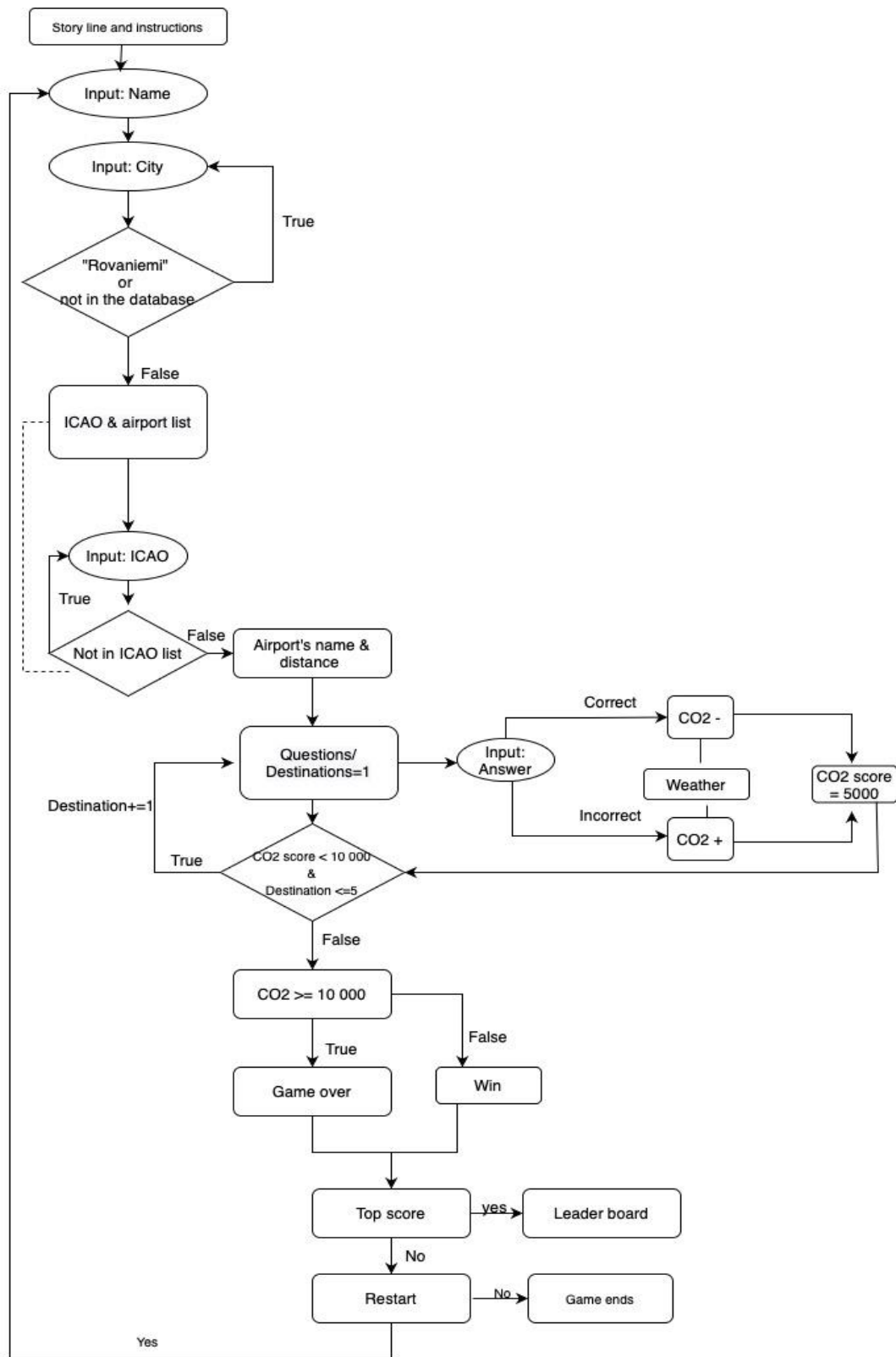


Figure 1: Flight Simulator Game Diagram

## Functional requirements

Figure 1 represents the game's diagram. The game starts by telling a player a storyline and instruction on how to play the game. Game's mission and goal is also clarified explicitly before the player starts the game. To raise users' experience and interactive engagement, the player is asked to answer a few questions before his flight. After that, the player starts the game by firstly choosing a city where he would like to depart from. The function is set to ensure that the input city either is available in the database to be called or is not "Rovaniemi " since the final destination is Rovaniemi. As long as the player chooses the city which meets these requirements, all airports in the chosen city and ICAO codes are listed. The program then asks the player to select one among those listed airports to start his journey. While loop function is set to make sure the player inputs ICAO which is in the provided list. Otherwise, he is asked to re-enter his choice. As long as the ICAO is input correctly, the program announces the distance between the chosen airport to the final destination, Rovaniemi Airport. To help the player persist in his mission and goal, the program reminds him again that the CO2 he has consumed in order to reach the first destination is 5000 units.

In order to win the game, the player needs to pass through 5 destinations and keep CO2 consumption under 10 000 units. The while loop condition is created to ensure the user plays the game under those conditions. In every destination, a random question is called from a list of questions. After the question is called, it is removed from the list to avoid duplicated questions in the following destination. Depending on the answer (either correctly or not), the player can get rewarded or punished. The reward is a deduction of the player's CO2 consumption which is randomly chosen based on the weather condition connected to a different score. In contrast, when the player answers the question incorrectly, his CO2 consumption is increased by random scores which are connected with weather conditions in the database. As long as CO2 consumption score is lower than 10 000 units within a range of 5 destinations, the player can continue to the next spot. If those conditions are not met, the player gets out of the loop. After that,

“if-else” condition is called to check if the player won or lost the game by checking if CO2 consumption is smaller than 10 000 units.

After finishing the game, players' names, their final scores, and ICAO codes of the departure airports are pushed in the database. The player can decide if he would like to check the top 5 best scores of the game. Finally, he can also choose to restart the game or not.

### **Quality requirements**

One of the quality requirements applied in the program to improve users' experience is the effect text. Not only does it give the player some spare moments to comprehend instructions or messages, but also it stops the game for a few seconds before moving further steps, allowing users to build anticipation.

The second effect in terms of quality requirements is color text. It emphasizes on important messages and helps users have a good understanding of what they need to focus on. For instance, if the player's answer is incorrect, the text is in red color. Any message requiring players to interact with would appear in green color. Blue color applies for correct answers and positive messages such as “You won”.

Additionally, the effect of underlining certain text helps players to focus on the game's important messages. ASCII art increases the game experience and makes the game more entertaining.

Ultimately, engagement is also an essential part in the quality of a game. Adding score rank encourages the user's desire to keep playing and improving the previous scores.

Including messages such as “Take-off 3...2...1...” also help users visualize the game better.

For more references and demonstration, please check:

<https://github.com/hannahhoang2704/GameProject>