Boom! stop the nysse -pelin dokumentaatio

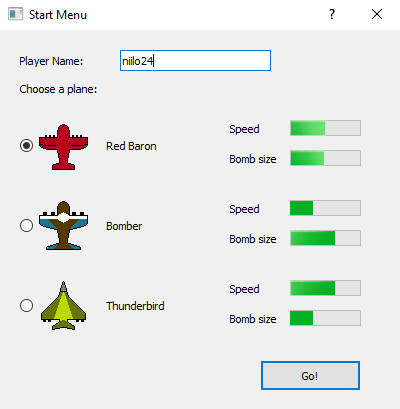
This document contains instructions for playing the game and description of the student side program logic. Also, the workload of both team members is listed.

# Game instructions

The goal of the player is to destroy as many nysses as they can. The player controls a plane capable of dropping bombs. In addition to the nysses, there are also clouds moving across the game area that the player must avoid. Colliding with a cloud damages the plane and decreases score by 5, and after 4 hits the plane is destroyed. After collision, there is a short cooldown phase (different for each plane) during which the plane will not take damage from collisions. The game goes on until the game time runs out or until the plane is destroyed.

## Starting the game

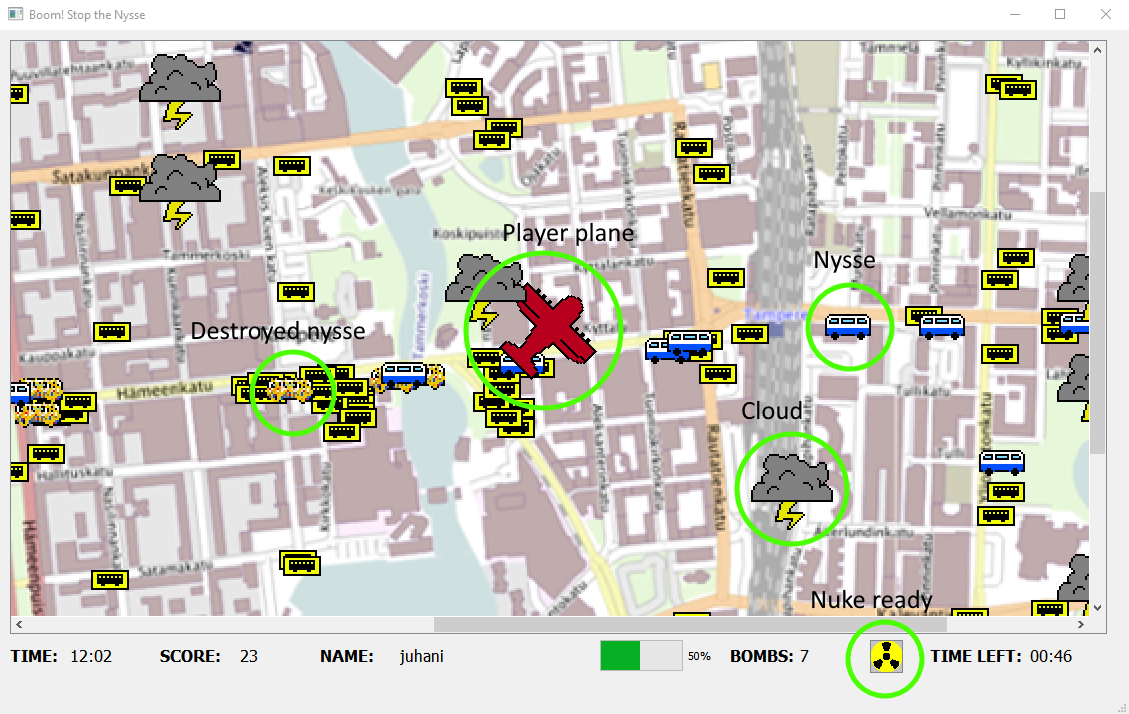
At the start of the game, a start menu opens up where player must enter their name and choose a plane. There are three different planes available, each of which have different capabilities.



*Start menu*

## Playing the game

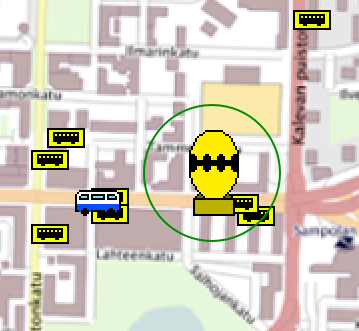
The game round begins, when player clicks the ”Go!”-button. The direction of the plane is controlled with WASD keys and bombs are dropped using SPACEBAR. Player gains one (1) point for every nysse destroyed and one (1) point for every passenger inside the destroyed nysses.



*Game view*

## Special weapon

During the game, the player may find a nuke that spawns within the game area in a random location. The player may collect the nuke, after which they can drop the nuke using R key. The nuke is much more powerful than a regular bomb and thus has area of effect. They player may only have one nuke collected at a time. New nuke spawns after the previous one has been collected and dropped. If the player has a nuke ready to be dropped, it is indicated with a symbol in the lower right corner of the MainWindow.



*Nuke ready to be collected*

## End of the game

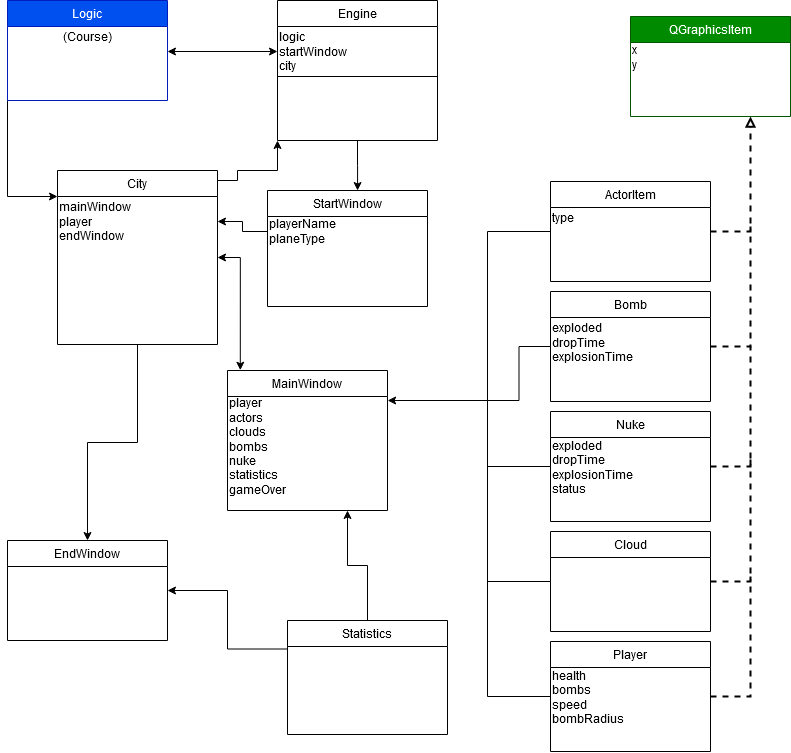
The game ends, when the game time runs out or when the plane is destroyed after hitting clouds too many times. After the game ends, game round statistics are displayed in the end dialog, along with all time TOP5 scores of the game.

# Game logic & workload distribution

The course side template of the project contains the logic for moving the nysses and passengers, as well as some interfaces. During the coding of the game, the most important interface was the CourseSide::ICity, from which the City class is inherited. The City class moves the actors in the city by receiving function calls from the CourseSide::Logic object. The City also calls the actor functions in the MainWindow, which controls the actor graphics. Other than the City class, the student side classes were inherited from Qt Classes.

## Class diagram

The following diagram contains the most important student side classes of the program and the essential data stored in the classes. The interaction between classes is described with arrows.



During the game round, the MainWindow controls most of the game actions. MainWindow stores all the actors in the game and controls their movement and drawing grahics.

## Additional features

Below is a list of additional features added to the game, as described in project info (<https://plus.tuni.fi/tie-0240x/fall-2020/modules_00/00_tyoohje/>)

* Even screen updates
* Scrollable map
* Even movement of the player figure
* Passenger amounts
* Following the game state
* Top5-list (top5 is better than top10)
* Updates to the playable figure
* Own feature: health.

## Workload distribution

Below, the workload of both group members is roughly listed. However, it should be noted that many classes contain changes made by both members.

Väinö Kahala

* Graphics

Sound

* Movement of actors
* Player plane & controls
* Bombs
* Nuke
* Clouds
* MainWindow

Elias Halkola

* Start dialog
* End dialog
* Statistics
* Statistics unit tests
* Saving the data (hiscores)