Name: Thong Nguyen & Minh Do

Group: ABC

PROJECT DOCUMENTATION

1.Introduction

Tampere is on verge of downfall due to the alien invasion. The invading forces are swarming the sky of our beloved city and rain down destruction upon it. To avoid the inevitable dire fate, governors of Tampere decided to evacuate its citizen to several bunkers built underneath the city by bus. However, the military is not present in time to escort the buses due to busy fighting the vanguard alien force in the frontline. Luckily, a group of students from Tampere University bring in their ultimate weapon for the rescue: a gunship, which was their school's project prototype. The gunship is the last hope of mankind between victory and destruction.

2. Game control and manual

2.1. Game menu and GUI

The game start with a start window, which is a main menu. The players have options to start the game, open score leaderboard, open user manual or exit the game. In addition, the players can mute the background sound by checking the mute checkbox.

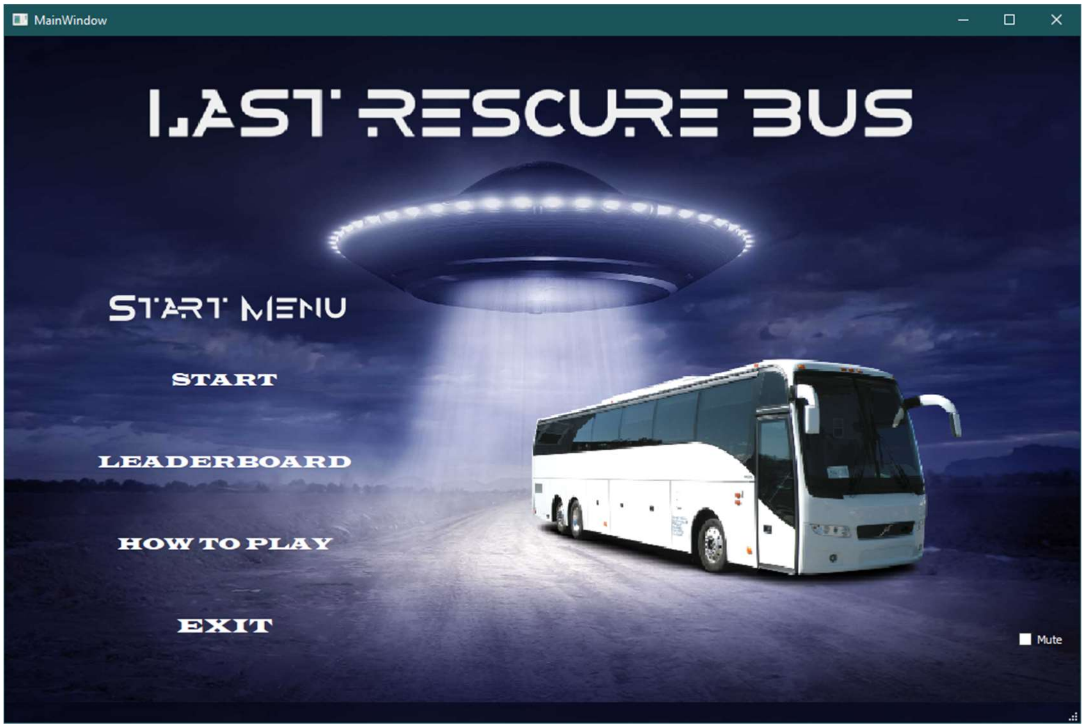


Figure 1 : Main menu of the game

When the players click start, an initialize dialog will appear to ask the for the player's name and ID. If either line edit is empty, a warning will be thrown asking the player to input the required information. A checkbox is used to initialize the second player if checked. The players can click OK to proceed the game or Cancel to close the Dialog.

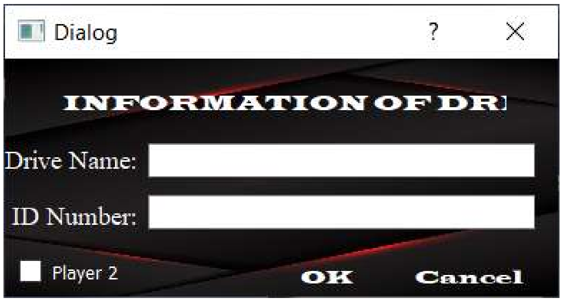
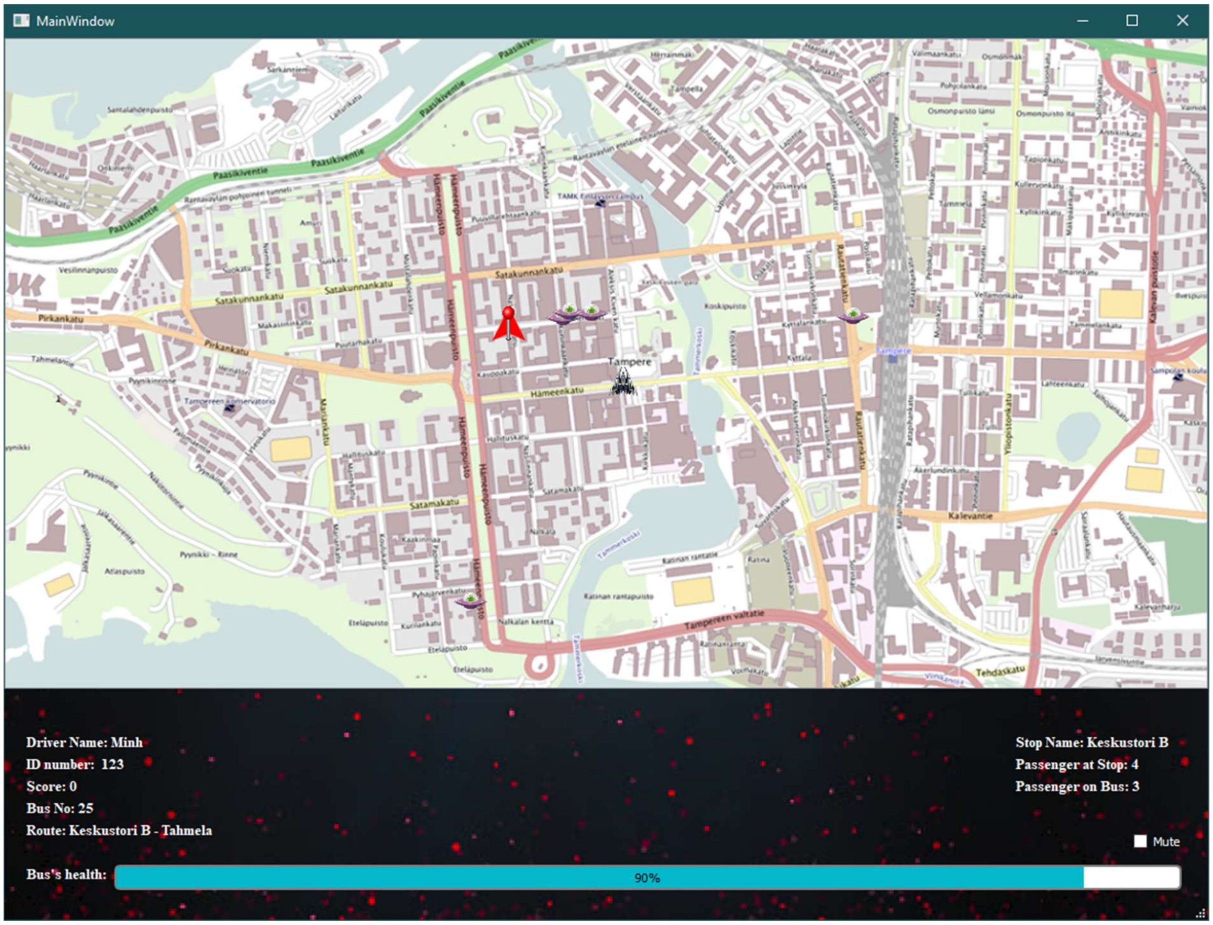


Figure 2: Initialize Dialog

After the player click OK, the main gameplay window will be invoked.



Driver Name: Minh
ID number: 123
Score: 0
Bus No: 25
Route: Keskustori B - Tahmela
Bus's health: 90%

Stop Name: Keskustori B
Passenger at Stop: 4
Passenger on Bus: 3

☐ Mute

Figure 3: Gameplay Window.

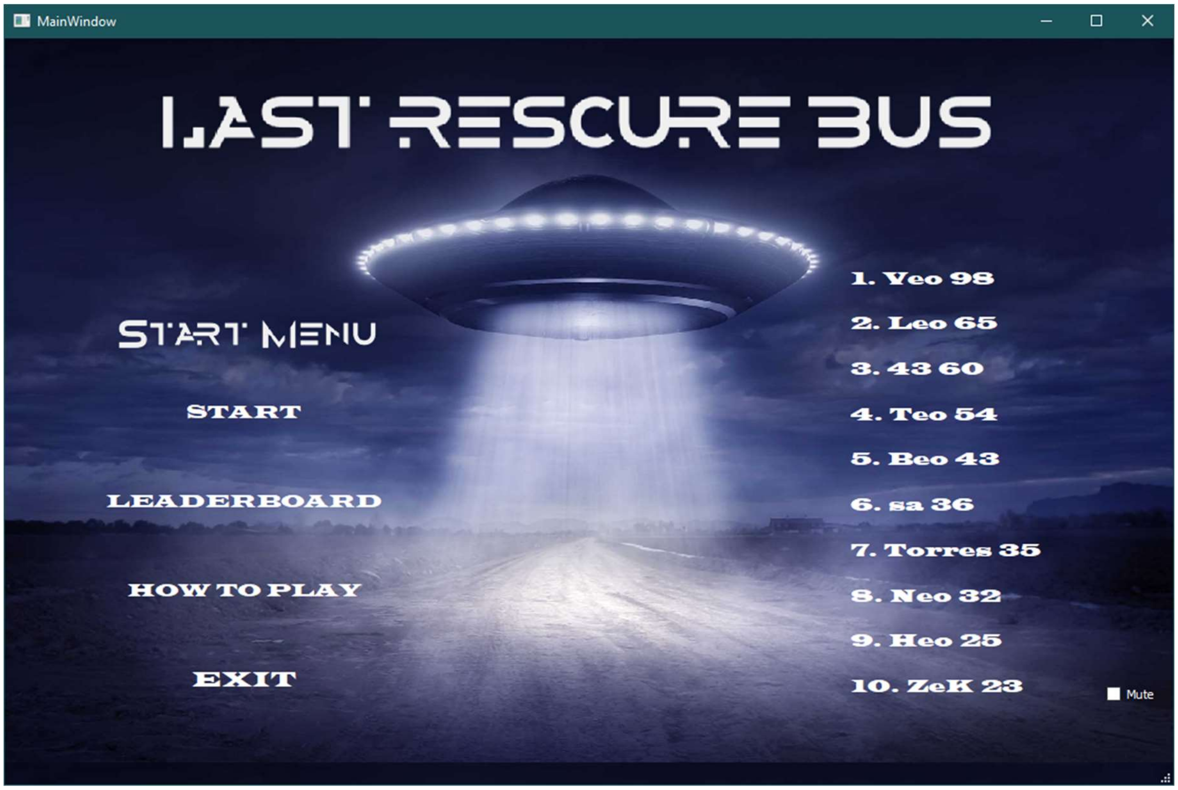
The initial window appears with the city map of Tampere, starting position of the bus, its initial pin of the bus stop and the player's gunship. At the bottom of the window, to the left corner, player's ID and name, Score, Information of the bus are displayed. On the other hand, to the right corner, stop name, number of passengers are displayed. Finally, at the very bottom is the bus's health. If the bus's health is reduced to 0, the game is over. The background sound can be muted using mute checkbox.

2.2. Gameplay Control

- The gunship can be moved by using arrow key on the keyboard.
- The gunship fire rocket to the alien by using space key on the keyboard.

3.Game rule

The object the players control is a gunship so it can move freely on the map. The bus using to evacuate the citizen randomly choose the next bus stop destination on the map based on the info obtained from the offline external file. The aliens spawn randomly on the map and tracking the position of the bus and pursuit it. Every time the alien hit the bus; the bus's health is decreased by 10. When the bus's health reaches 0, the game is over, the current score of the players is saved and the players have option to restart the game. The alien can be defeated by being hit from the player's gunship bullet. When the alien is destroyed, score is awarded to the players. The player who kept the bus survives longest and defeated the largest number of aliens will be placed first place in the leaderboard and considered the winner of the game.



1. Yeo 98
2. Leo 65
3. 43 60
4. Teo 54
5. Beo 43
6. sa 36
7. Torres 35
8. Neo 32
9. Heo 25
10. ZeK 23

Figure 4: Score Leaderboard.

4. Class Diagram

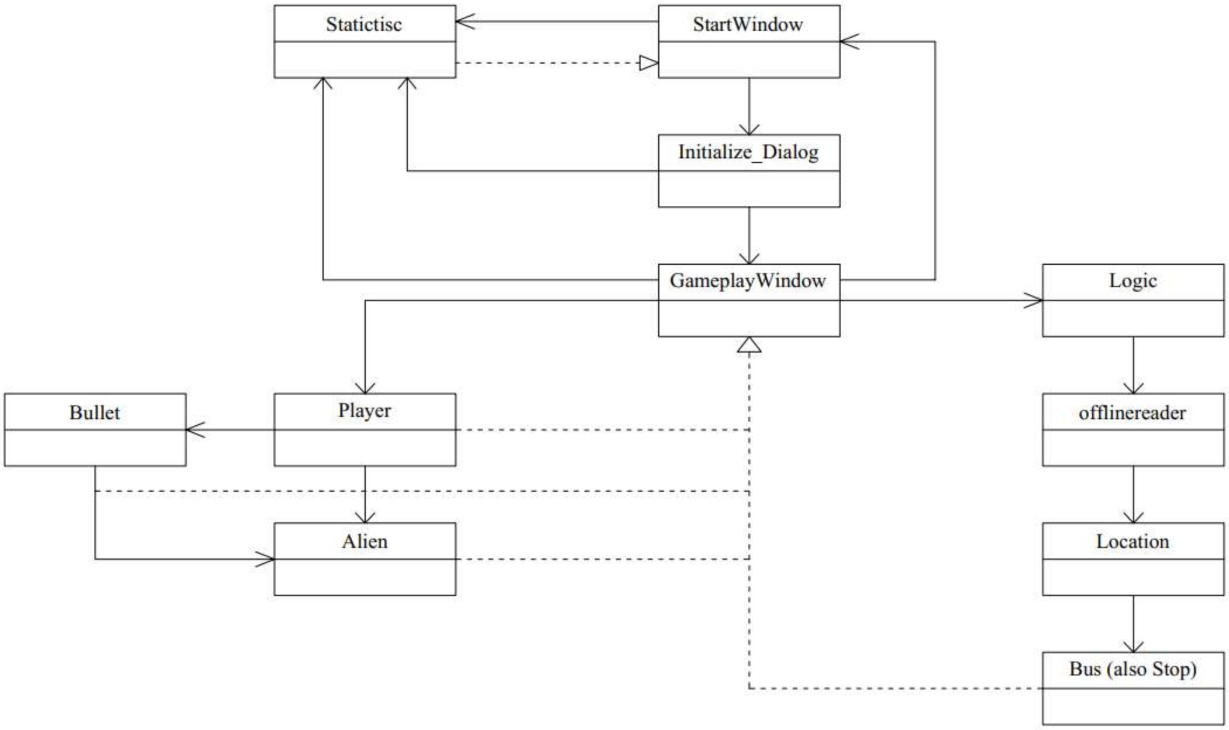


Figure 5: Class Diagram of the project

C++ File	UI File/Header File	Class and relation with other classes	Description
gameplaywindow.cpp	Gameplaywindow.ui Gameplaywindow.hh	GameplayWindow This class's object can be called from Initialize_Dialog, putting score and player's name to Statistic's object, adding Bus, Player and Alien's Objects to Tampere city map, using Logic's method to spawn aliens, guiding bus to stops, start the timer.	The class provides methods that set size of the window, set the picture from external to the background, setting the player's information on the window, moving bus object to next stop, moving the pin of the stop to the next stop playing the background's sound, update the player's score, update the bus's health if it got

			hit, checkbox for player who does not want background sound, reading data from the file to assign the bus's route
initialize_dialog.cpp	initialize_dialog.ui initialize_dialog.hh	Initialize_Dialog This class opens GameplayWindow ‘s object if OK is clicked, save the player’s input to Statistics class’s object.	The class provides methods that receive data from the user input to line edits, OK push button to start the game, Cancel push button to close the Dialog, checkbox for initialize 2nd player, start the gameplay window.
startwindow.cpp	startwindow.ui startwindow.hh	StartWindow This class opens Initialize_Dialog’s object if button clicked, update its label to print class Statistics’s leader map if button is clicked	The class provides methods that using push button to start the initialize_dialog, score leaderboard, quit the game and checkbox to mute/unmute the background sound.
alien.cpp	alien.hh	Alien This class has method to track Bus’s object’s coordinate on the map and emit signal to Player class’s object to count number of hit the bus took	The class provides signal emitted when the alien hit the bus, the alien is out of the map, methods of movement of the alien object, getting the bus's location to follow/pursuit
player.cpp	player.hh	Player This class has method to control where the Bullet class’s object	The class provides methods using keyboard to control the airship, calculate score, get number of aliens in the window

		spawn and its moving direction, increase number of aliens, send bus location signal to Alien object.	get the bus's coordinate, count number of hits, spawning aliens to the maximum number.
logic.cpp	logic.hh	Logic	The class provides methods reading data from json file, find route to destination for the bus, timer calculation, generate number of aliens
bullet.cpp	bullet.hh	Bullet Remove Alien’s object if the Bullet hit the Alien, and emit signal to Gameplay_window class	The class provides signal emitted when the number of bullet is more than 5, when the bullet hit the alien, movement of the bullet object, getting the bus's location to follow/pursuit
bus.cpp	bus.hh	Bus	The class provides methods of amount of health the bus is lost when get hit, receiving, releasing passengers its current coordinate on the map
score.cpp	score.hh	Score	The class provides methods increase or decrease the player's score and return current score.
statistics.cpp	statistics.hh	Statistics	The class provides methods reading player's info database, print out player's score board, obtain player's info from input and save it to the dabase

			sorting player's score

5. Work Division:

Table of works

Thong Nguyen	**Minh Do**
Design front-end and back-end	Statistics class
User manual	Unit test
Draw Class diagram	Write documentation

6. Missing features and bus:

- **Mode 2-players:** In the initialize dialog, we also add a check box for user to choose mode to play. However, this feature still has not finished yet.
- **Pop-up information of bus stops:** Initially, we intend to pop a small message box to show the information of the stop like:

  + Stop name.

  + Passenger wait to pick up.

  + Passenger want to leave bus.

- **Final way to win the game:** because map only can show the location of bus stops that locate in the center of Tampere, so we cannot move the bus the final destination and implemented method to win. We intend to pop up an animation window to inform user that game is finished and win (also get some extra point).
- **Bug:** Bus cannot move correctly in road due the wrong scale size of the map and our adjustment seem not work correctly.

7. Additional features:

- **Top10-list:** This feature is implemented in statistics class, method *save_Database()* will save data to a database locate in the application path (debug folder or release folder). method *read_Database(QString fileName)* will read data and update to UI.
- **Updates to the playable figure:** This feature is implemented in *move()* of class Bullet. When alien and bullet hit each other, both will disappear.
- **Updates to the playable movement:** player aircraft and bullet's movement is updated in their method call *move()* in Player and Bullet classes. By setting a origin point in the center of their image, the figure will rotate a specific degree which depend on the direction that user choose.
- **Information of stop and passenger:** these information is located at the bottom right of the gameplaywindow UI and being updated in *MovingBus()* method in gameplaywindow.

When bus reach to next location on map after 15s in game, these information will be update immediately.