

Project documentation

1. Overview

Starting the software opens a menu for three different options; play, options and quit. In options another menu is opened where user can choose the difficulty level, which also changes the background. Quit quits the game and closes window. When game is started the player spawns, and after brief delay enemies are spawned. Player can move with w to gain speed, use a and d to control the pitch of the plane and shoot bullets with spacebar.

Enemies track the location of the player and shoot at the location. Enemy planes also try to catch up to the player. Ground units have a set angle in which they can shoot, in order to reduce friendly fire among them.

Player has finite number of lives, hp and ammunition. When hp of the player goes to zero, player is spawned at initial location and number of lives is reduced.

Winning condition is to destroy all enemies by shooting bullets at their model and reducing their hp to zero. Losing condition is if all lives are lost. Both losing and winning screen prompt unique screens in which player is advised to press enter to get back to start menu giving a possibility to start over.

2. Software structure

Main loop of the game is done inside gameLoop. Box2D bodies are defined, created, and connected to SFML sprites inside Bullet, Plane and groundUnit classes, which are derived from a purely virtual Entity class. All created groundUnit objects are stored and controlled inside an enemyGround object, bullets inside Projectiles and Planes either inside either Player or enemyAir. The b2World is created inside World class.

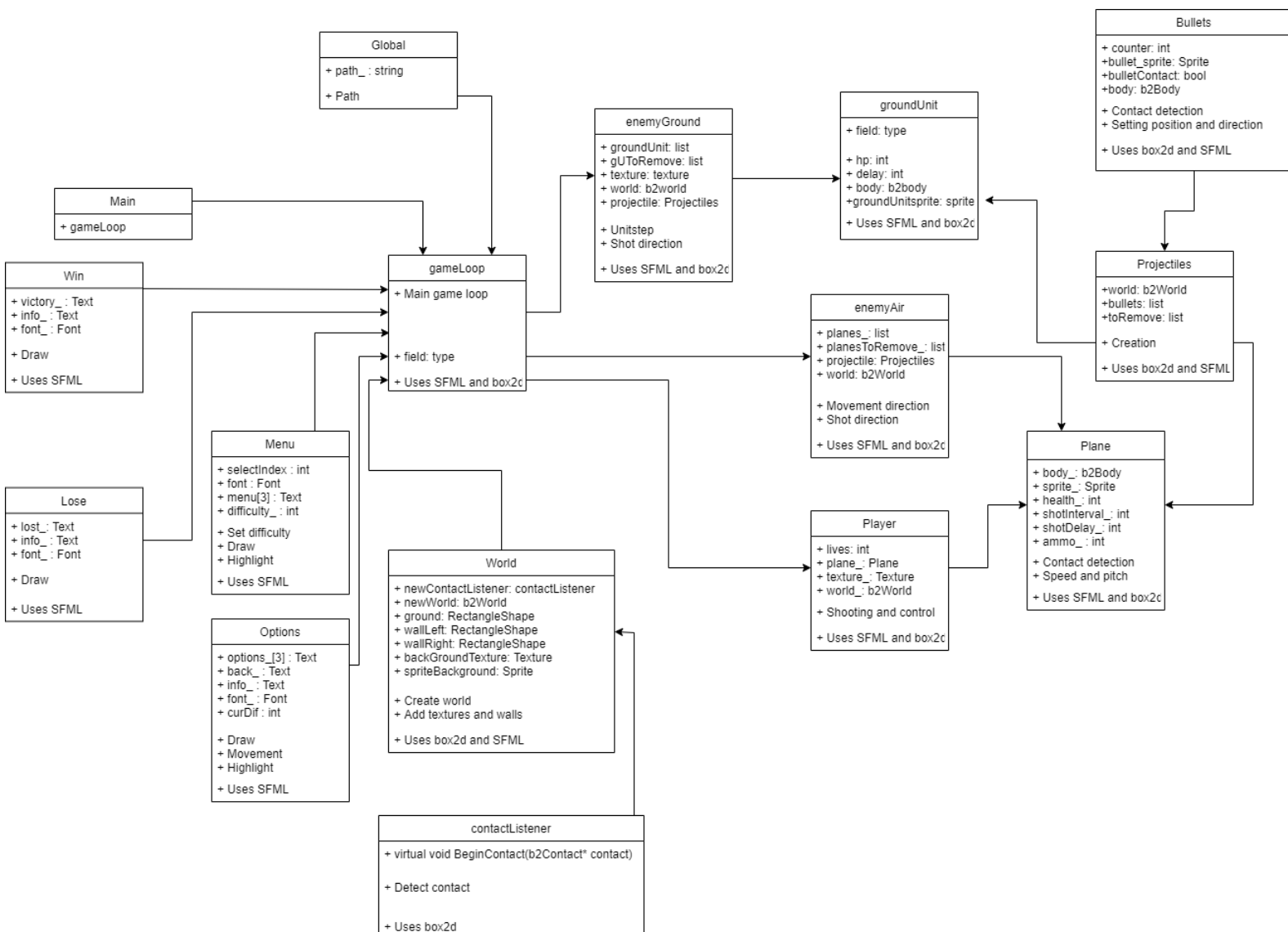


Figure 1. Diagram of the software structure

3. Instructions

Prerequisites for compiling the program are sfml and cmake. Box2D files are included. Tested on 2.4.2 version of sfml on Debian Linux. For WLS, xserver display is needed to display the game, for example xming. With xserver following command must be executed:

```
export DISPLAY=:0
```

to install cmake and sfml:

```
sudo apt install cmake && sudo apt install libsFML-dev
```

To compile and run the program, move to air-combat-2019-1/src and execute following commands

```
mkdir build
```

```
cd build
```

```
cmake ..  
make  
./fighter
```

In the main menu of the game, play starts the game, difficulty can be chosen from the settings menu and quit closes the game. The player plane is controlled with W, A, D and space. W accelerates the plane, A and D control the pitch and space shoots. Main menu can be returned to with ESC. Goal of the game is to survive and kill all enemies without losing all lives.

4. Testing

Program was tested manually over the course of the program. Visual cues, error messages and prints were used to detect issues and solve them. Valgrind was used to detect and correct memory leaks. Some leaks were detected from bullet class, but could not be resolved before deadline, which cause some memory leaks. During development the group determined that beforementioned testing methods were superior for us in comparison to making dedicated tests.

5. Work log

Development of project was left on the back burner for the first two weeks of the project due to other priorities of the group. Only thing made was inclusion of box2d files by Patrick. During the third week we got to know the SFML and box2d libraries and got them working. The base world of the game was made, and plane sprite started moving according to user inputs with initial flying physics. Time spent was approximately 8 hours. The fourth week was spent on building the projectiles for player and enemies, and for spawning enemy planes with AI for them. The time spent was around 12h.

The project kicked into overdrive mode during the last two weeks. The fifth week consisted on constructing classes for menu and making collision detection work the way we wanted. Modifications were also made to almost all classes made before as the know-how of the group grew. The plane flight pattern was modified to feel better. Time spent was nearly 20 hours.

During the last week we changed gameloop from main to its own specific class, modifying it heavily. Enemies were added to ground and their shooting AI was modified few times. The menu got its difficulty settings which also made the game background change. The plane was made to not take damage if landing was slow enough. Time was spent on fixing some of the memory leaks which were plenty. Time spent was close to 40 hours.

Much of the project was produced in a group setting and so most parts were heavily influenced by various members of the team. Some of the main responsibilities are listed below.

Patrick Lahti was responsible for planes and their control and physics as well as the collision information extraction from box2d. Henri-Matias Tuomaala was responsible for the ui, menus, end screens and their control and logic. Tommi Paasonen was responsible for the

ground units and projectiles. Jouni Kleemola provided finishing touches and various smaller additions to the later stages of the project.