ELEC-A7151 Object oriented programming with C++
Air Combat 3
Aarni Kolehmainen, Henrik Granö, Joona Viheriä, Waleed Iftikhar

Project documentation

Overview

Modern air combat - team match is multiplayer airplane game for two players. It was inspired by the classic game triplane turmoil, which was released in 1996.

Gameplay

In the game two users battle each other by trying to shoot the other player. Player 1 moves the plane with the arrow keys and and shoots with the M-keym. Player 2 moves with A, S, W, D and shoots with the tab key. The physics of the game are very complex and have been made to ensure smooth gaming experience for the users. The planes can take 4 hits before being destroyed. Air-to-air collisions and collisions with the ground are also lethal. There was a plan to provide option for a single player mode against an ai foe, but due to time limit we unfortunately were never able to finish this.

Quick user guide

Upon launching the game, the players are greeted by the main menu where they can choose their desired plane, and launch the game. The menu items can be navigated by using the mouse and by clicking. By choosing “Start game”, the game presents the controls before starting the game itself. When either player loses, the game returns to the main menu where the a new game can be started.

Planes

There are different planes with different appearances available. However, only one of these is used currently. For future reference, players can choose which plane they want to play as, and this choice will determine some plane stats such as speed and maneuverability.

Game worlds

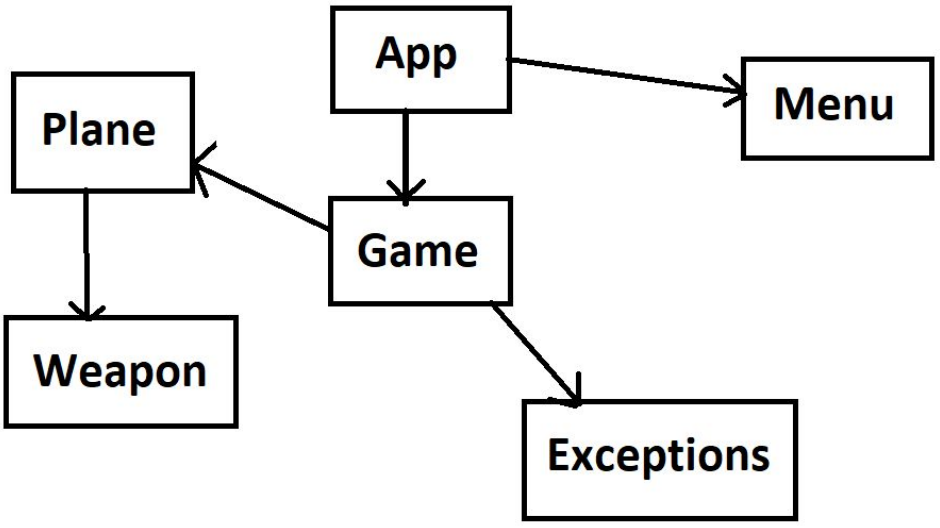
There are four different battle environments available: Fields, City, High altitude and Mountains. The maps have different collideable terrains defined by their respective physics bitmap. These are easy to create using a suitable image editor of the user’s choice, for instance, MS Paint. The

scale of the physics bitmap is 10x less than the environment (therefore, if the environment image is 1280x720, the physics bitmap will be 128x72), and it has to be saved as a 24-bit bitmap. Flyable areas should be marked with completely white pixels, while collideable pixels can be any other colour. To use user-defined maps, they have to replace old maps in the /src/maps/-directory.

User interface

User interface is designed to facilitate user to use both keyboard and mouse to select options and move back and forth between the windows. User can directly play default mode. User can also select different battle grounds and planes to have great gaming experience.

Software structure



App runs the program and handles game and menu. Menu handles input about users will to play the game or not. Game initializes map and planes and game object is located as a member of app. Plane has information about bullets and box2d body for plane itself. Plane has functions for moving and shooting which are called in apps gameloop.

Work log

Week 1 5.-11.11

Planning
Figuring out the libraries
Programming environment setup

Week 2 12.-18.11

Figuring out the libraries
Programming environment setup
Basic class setup of App and Game

Week 3 19.-25.11

Working main menu and map loading for terrain and plane coordinates
Begin work on main game loop
Create Plane and Weapon classes

Week 4 26.11.-2.12

Gameloop runs and displays map
Start work on box2d components
Plane controls and rudimentary ground collision

Week 5 3.-9.12

Much better main menu
Add terrain physics from map
Modify flight characteristics

Week 6 10.-14.12

Finalize flight characteristics and shooting
Add another player and winning condition
Finalize graphics and sound

Week	Aarni	Henkka	Joona	Waleed
1		4	4	2
2		4	4	3
3		12	8	6

4		12	10	30
5		12	10	15
6		22	20	38

Testing

We did extensive user testing several times during the projects lifecycle. We always tested the recently developed features extremely thoroughly and discussed if something needed to be changed or our plan needed amending. We found this process very suitable for the needs of our project.

Building & makefile

A makefile is provided. However, while it compiles on Linux, it does not run well. Full functionality is guaranteed when compiling in CodeBlocks without using the makefile. Required libraries are Box2D and SFML.

Classes

App	**main_menu** **app_main** **game_loo** **render** **background** **count_bullets** **position_handler** **handle_input** **moveaction** **check_collisions** **show_instructions** **print_hp** **print_bar**
Plane	**get_type** **get_body** **move_plane**

	shoot **get_bullets** **get_counters** **change_hp** **get_hp**
Game	**Init** **isrunning** **update_physics** **get_player** **get_world** **objectify_map** **get_ground** **get_planecount**
Menu	**draw** **MoveUp** **MoveDown** **BoxSetu** **MenuSetup** **GetPressedItem** **MouseEvent** **BoxHighlight** **BoxClear** **get_Number_Of_Items**
FileException	**what**
Weapon	**get_type**