Project Plan

project structure:

```
air-combat/
├── bin/
├── data/
├── doc/
├── Makefile
├── obj/
├── plan/
├── src/
└── test/
```

Bin folder: contains all executable code.

Data folder: leaderboard, logs, images, audio, game settings, game levels

Documentation: We use **Doxygen** to generate documentation and graphs (of class relationship) from source automatically. Everyone should write documents, but there should be one or two proofreader (who read all documents))

Makefile: Using **CMake** to generate Makefile. Tam Nguyen can do **CMake**

Source folder: Contains all implementations and their header files. Does not contain test code.

Test folder: contain all test code using Google Test

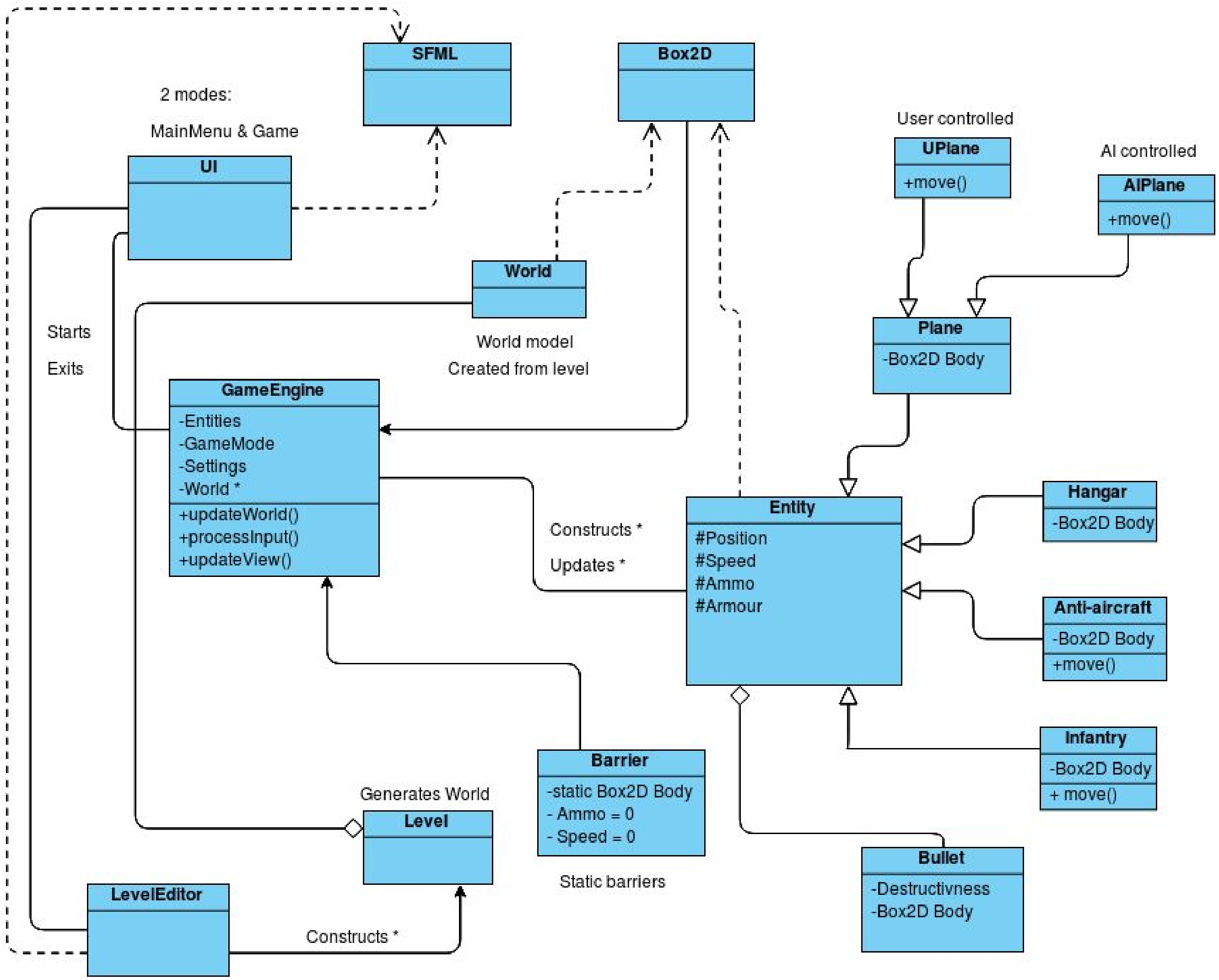
Tasks:

1. Game Menu (**Lauri Westerholm**):
   Start Single Player, Start Multiplayer, Start Online Game, Settings (e.g. controls, resolutions, difficulties) and Leaderboard
2. Define entity, infantry, anti-aircraft, plane and hangar (**Tam Nguyen ja Lauri Blomberg**). See Class relationships-figure below.
3. AI-level : higher AI means shoot faster, predict player’s plane better, and be able to decide for computer to shoot a bomb which is greater threat than bullets (**Oskari Järvi**).
4. Missions: free fight, escort, rescue, and destroy specific targets (**Oskari Järvi**).
5. Levels (**Oskari Järvi**).
6. Level editor (**Lauri Westerholm**)
7. Multiplayer on same keyboard (**Lauri Blomberg**)
8. Online Game: Two human player support on a fixed map (**Tam Nguyen**).
9. Code review : investigate and update code for secure, fast and clean code (**Tam Nguyen**).
10. Documentation review : proofreader. (**Lauri Blomberg**)
11. Game Engine (**Tam Nguyen**, **Oskari Järvi**)

Technologies:

We use following technologies in game development: Box2D, SFML, CMake, Makefile, Doxygen, g++, C++17, Git, Google Test, spdlog.

Class relationships (just a sketch):


SFML
Box2D
2 modes:
MainMenu & Game
UI
World
World model
Created from level
User controlled
UPlane
+move()
AI controlled
AIPlane
+move()
Plane
-Box2D Body
Starts
Exits
GameEngine
-Entities
-GameMode
-Settings
-World *
+updateWorld()
+processInput()
+updateView()
Constructs *
Updates *
Entity
#Position
#Speed
#Ammo
#Armour
Hangar
-Box2D Body
Anti-aircraft
-Box2D Body
+move()
Infantry
-Box2D Body
+ move()
Barrier
-static Box2D Body
- Ammo = 0
- Speed = 0
Static barriers
Generates World
Level
LevelEditor
Constructs *
Bullet
-Destructivness
-Box2D Body