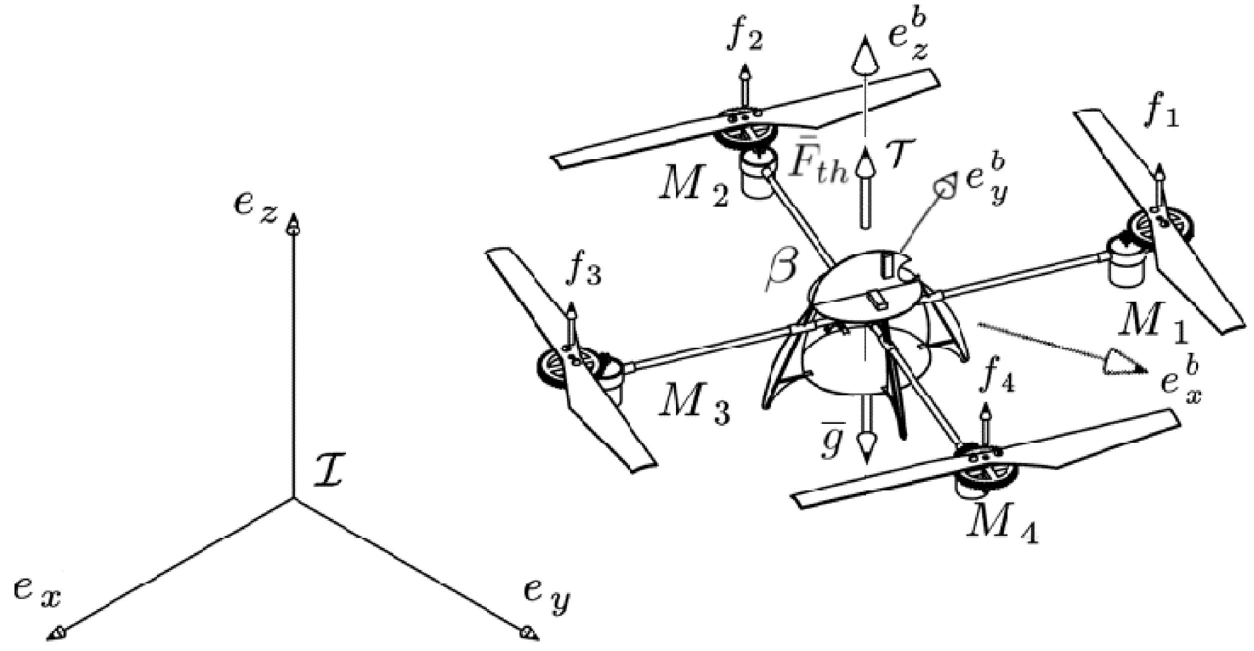
I. Major functional components

1. 3D graphic scene design and loading, model movement, and background boxing - Zhanfan Yu
2. Aircraft dynamic and control model design - Chenhao Yang
3. Autonomous obstacle avoidance planning strategy design - Shaobo Wang
4. User interface control logic and display design - Yipeng Lin
5. First-person POV and third-person POV switching logic, sound effects - Wei Wu



II. State of UAV:

1. Position:x,y,z
2. Rotation: Φ(roll, y), 𝚿(yaw, z), Θ(pitch, x)

III. Control logic:

- Arrow up: move forward(increase Y, pitch down)
- Arrow down: move backward(decrease Y, pitch up)
- Arrow right: move right (increase X, roll right)
- Arrow left: move left (decrease X, roll left)
- W: move upwards (increase Z)
- S: move downwards (decrease Z)
- A: rotate anti-clockwise about Z
- D: rotate clockwise about Z
- Q: switch between first and third POV

IV. Structure of the code:

main.cpp

Building blocks;
Initialize scenes;
Contains keyboard controls;

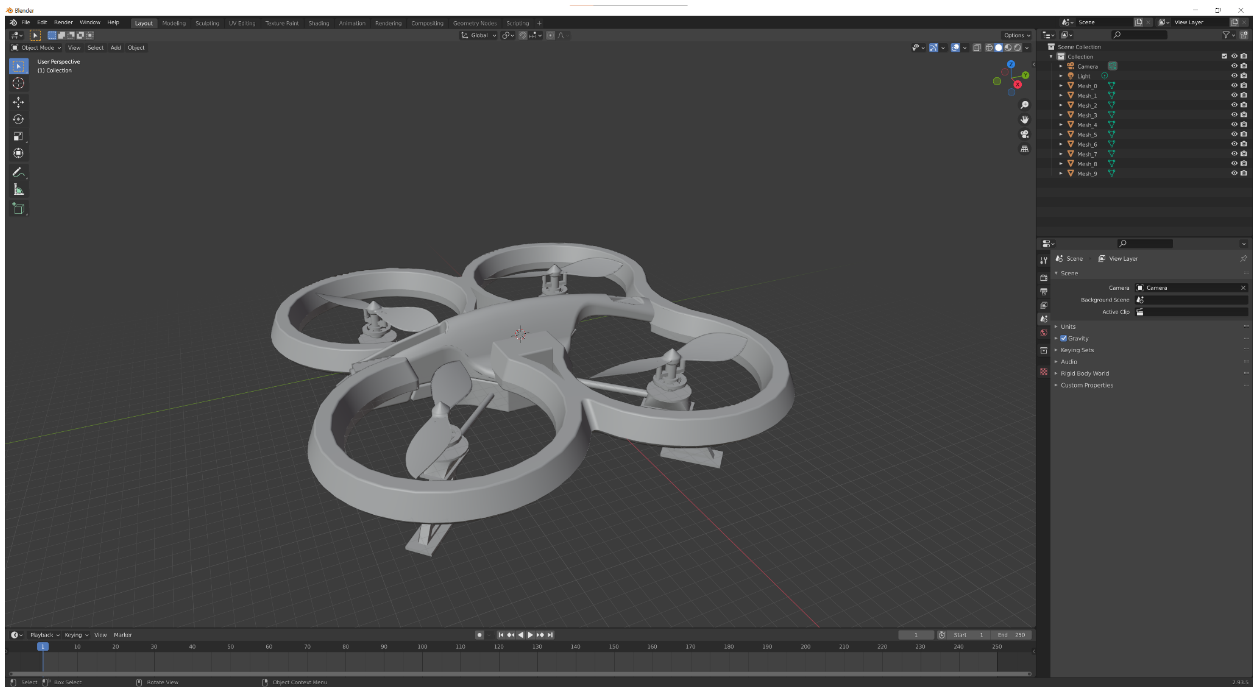
Flightcontrol.cpp
POV control.cpp
Class for calculating dynamics and controlling UAV

For mesh and model organization
mesh.h
model.h
Mesh models of buildings and UAVs
Shader.h
For render the model

For controlling camera position and angle
camera.h

V. The UAV model

https://sketchfab.com/tags/quadcopter



(Will be used as a starting point for the project)

[Autonomous obstacle avoidance planning strategy design]

VI. Obstacle detection:

i) obstacles position information

ii) aircraft position information
iii) collision tolerance for possible impact: computational effectiveness
iv) forward searching with predefined map

VII. Obstacle avoidance:

Two possible ways: dynamic method and kinematic method.
Dynamic method: treat obstacle avoidance as plastic/elastic impact between two spheres/balls with each object's safety range as radius.
Kinematic method: treat obstacle avoidance as positional constraint between similar spheres pair.

VIII. Planning strategy:

If no obstacle is detected, no planning strategy applied for aircraft's motion.
Else if an obstacle is detected, use "Coulomb's law" as an implicit model of motion law. The planning problem can be transformed into an optimization problem of finding the path with lowest cost. Or can be coupled with dynamic problems to avoid defective dynamics inside the collision tolerance. We can also implement a potential energy filed which calculates the potential energy on each point of the 2D map, and program the drone to avoid the highest potential energy coordinates on the 2D map

IX. Switching POVs:

The user can use the Q key to switch between first person and third person point of view of the aircraft. First person means the user will not see the aircraft, but will be similar to sitting inside the aircraft. Third person means the camera will be following the aircraft at a constant distance, and the camera's angle will change according to the pitch angle of the aircraft.

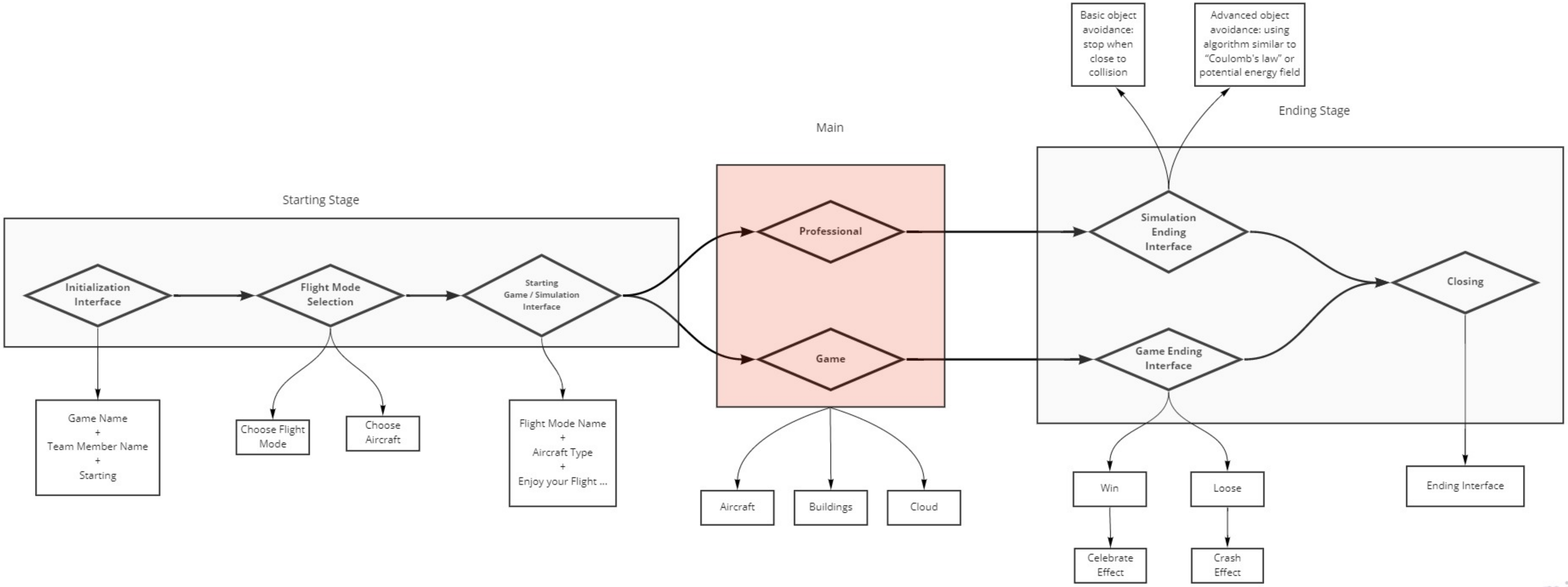
X. Sound effect:

The program will incorporate a copyright free recording of UAV operating. Because the aircraft will have varying speeds during user operation (speed up and slow down the velocity), the recording should also have speeding up and slowing down features to resemble the change of aircraft velocity.

XI. Appendix

User Interface Structure and full UI design can be found in the following appendix:

Structure



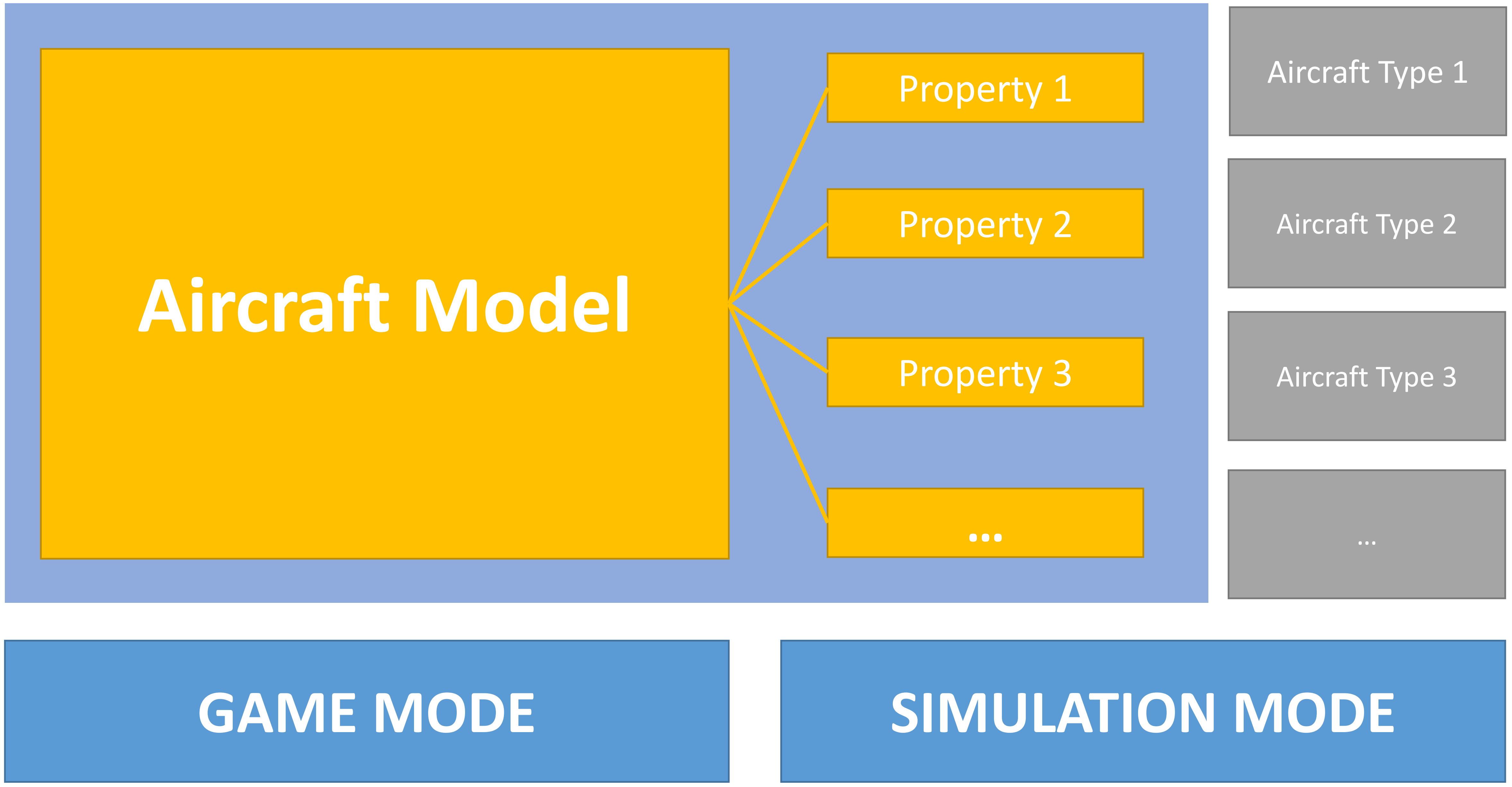
Starting Stage
Initialization Interface
Flight Mode Selection
Starting Game / Simulation Interface
Game Name + Team Member Name + Starting
Choose Flight Mode
Choose Aircraft
Flight Mode Name + Aircraft Type + Enjoy your Flight ...
Main
Professional
Game
Aircraft
Buildings
Cloud
Basic object avoidance: stop when close to collision
Advanced object avoidance: using algorithm similar to "Coulomb's law" or potential energy field
Ending Stage
Simulation Ending Interface
Game Ending Interface
Closing
Win
Loose
Celebrate Effect
Crash Effect
Ending Interface

Game Name

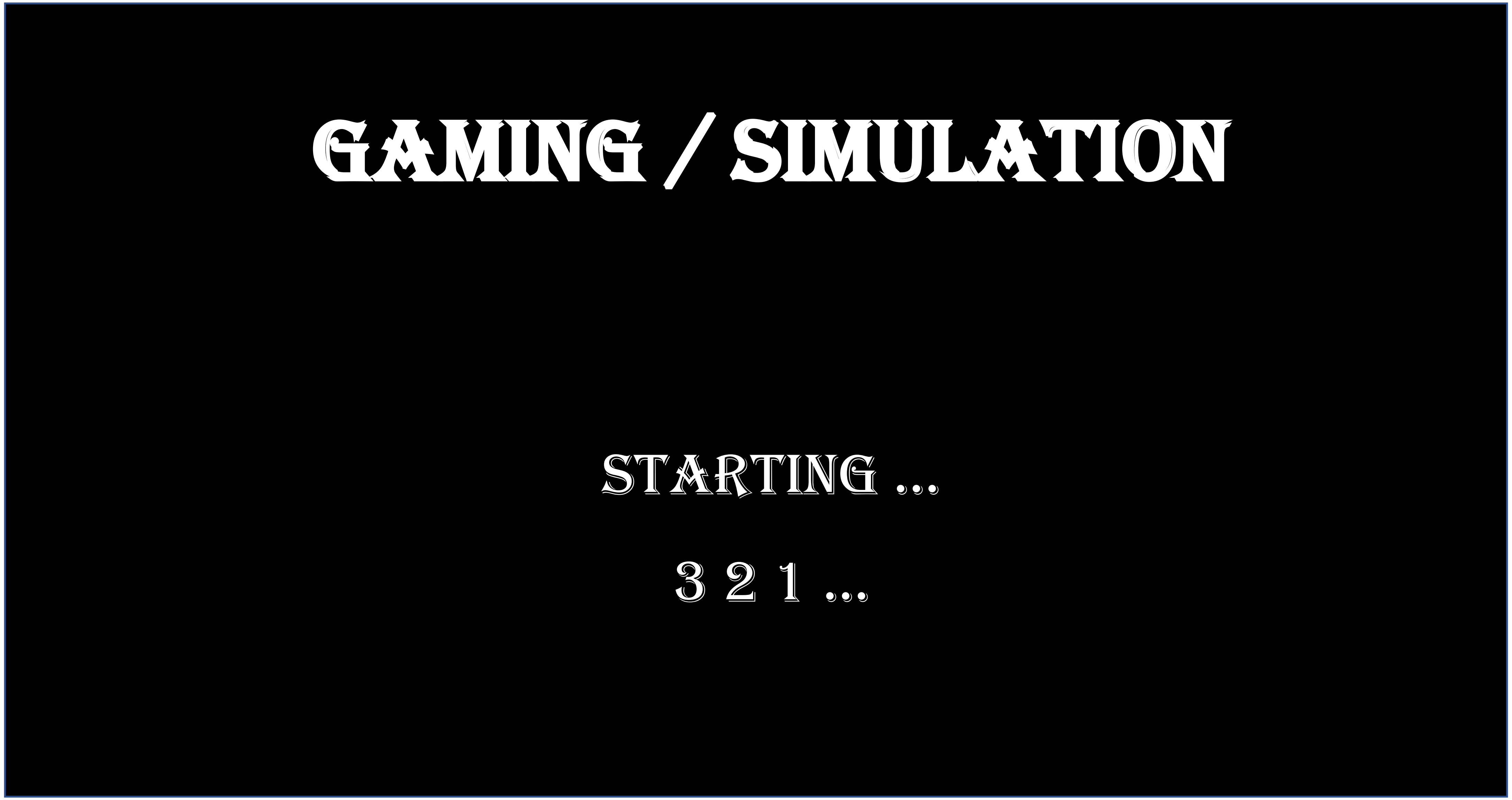
Loading...

Engineer Republic Made

I. Flight Mode Selection Interface



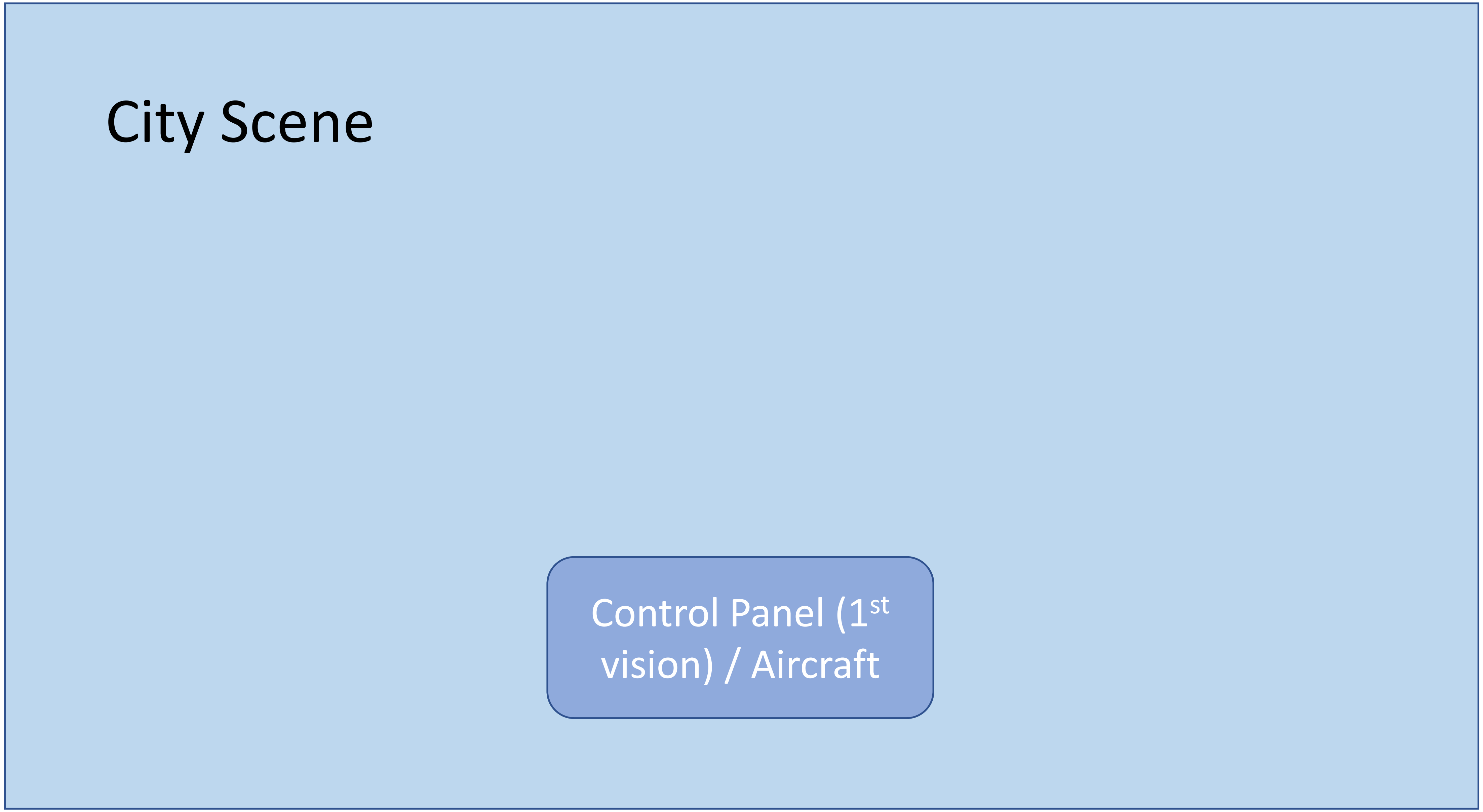
II. Starting Game / Simulation Interface


GAMING / SIMULATION

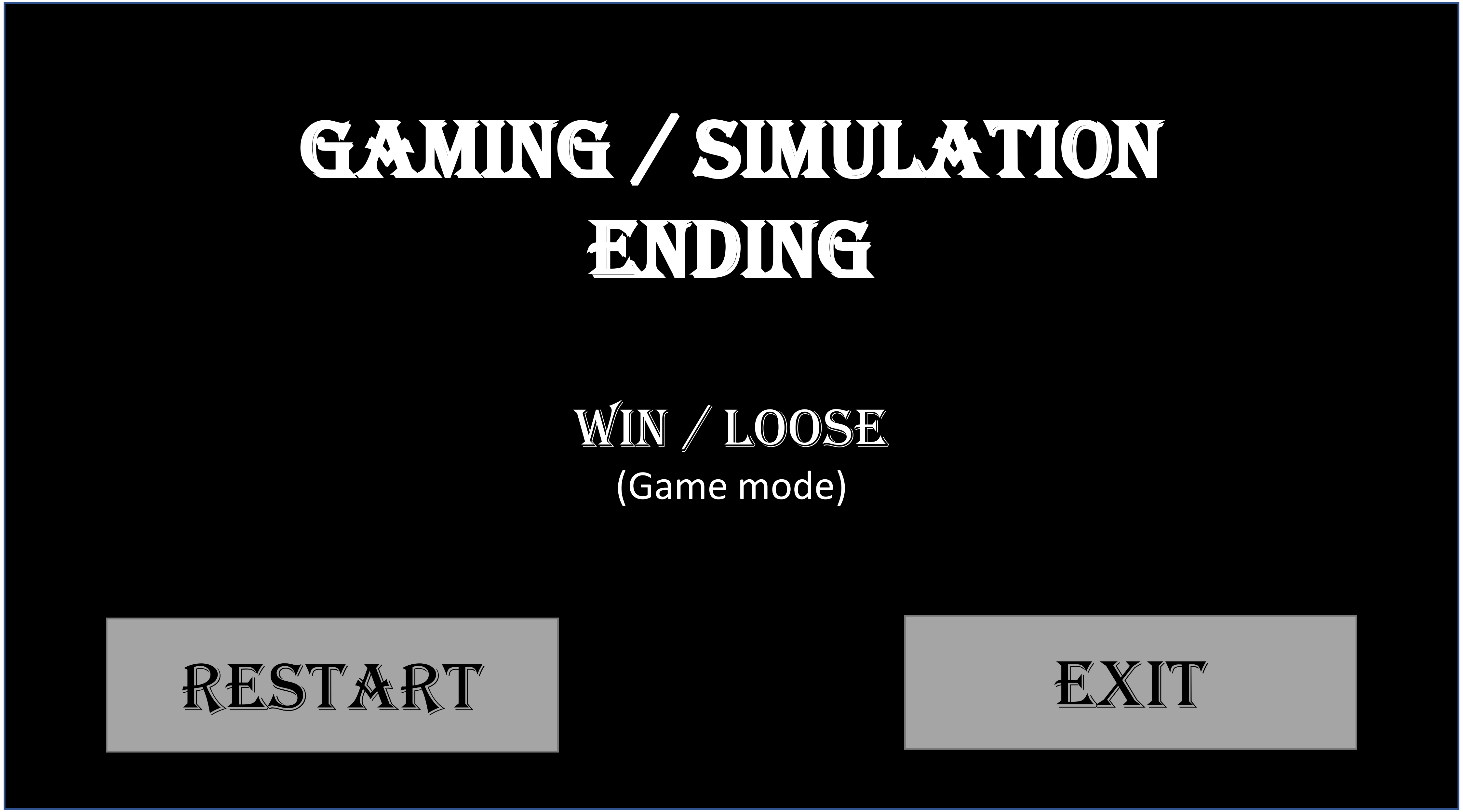
STARTING ...

3 2 1 ...

III. Main Part Interface



IV. Ending Interface



GAMING / SIMULATION
ENDING

WIN / LOOSE
(Game mode)

Closing Interface

City Scene Picture in the game

Thank you for playing