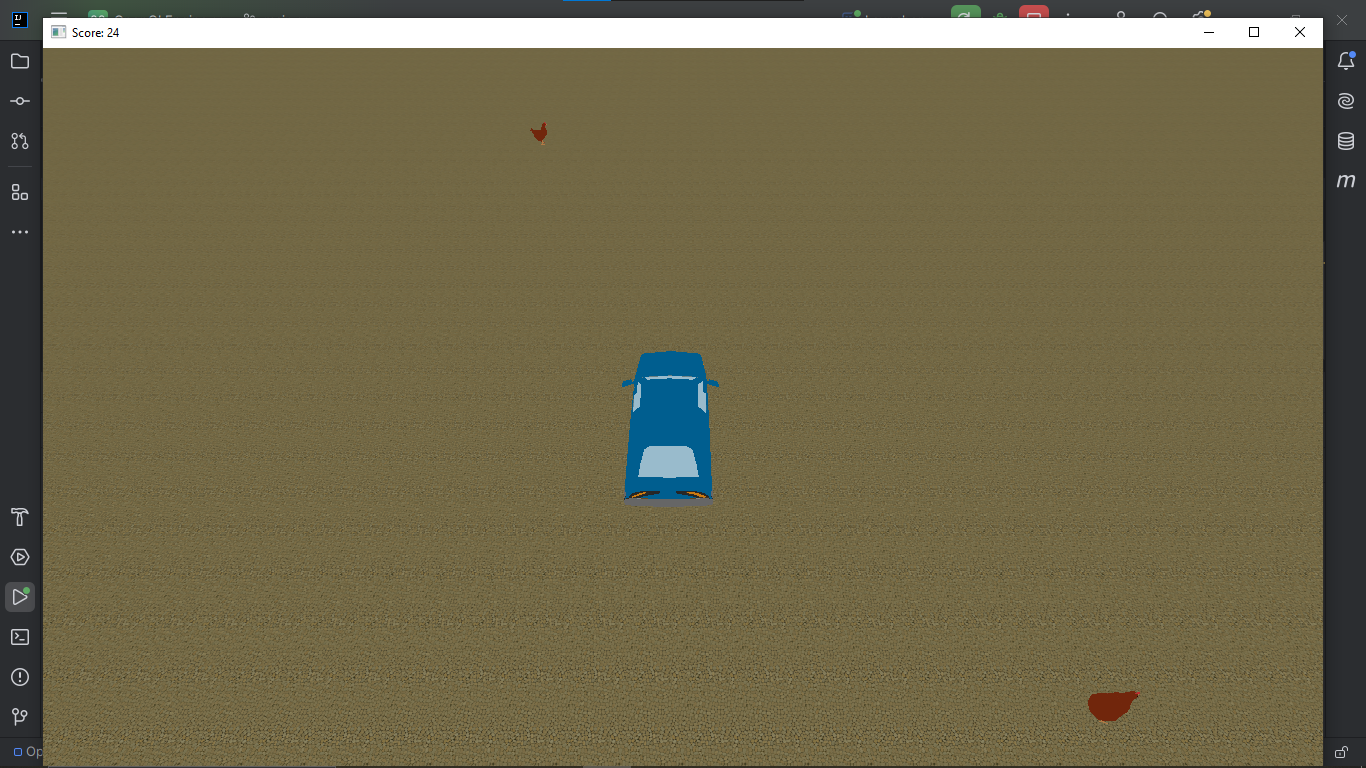
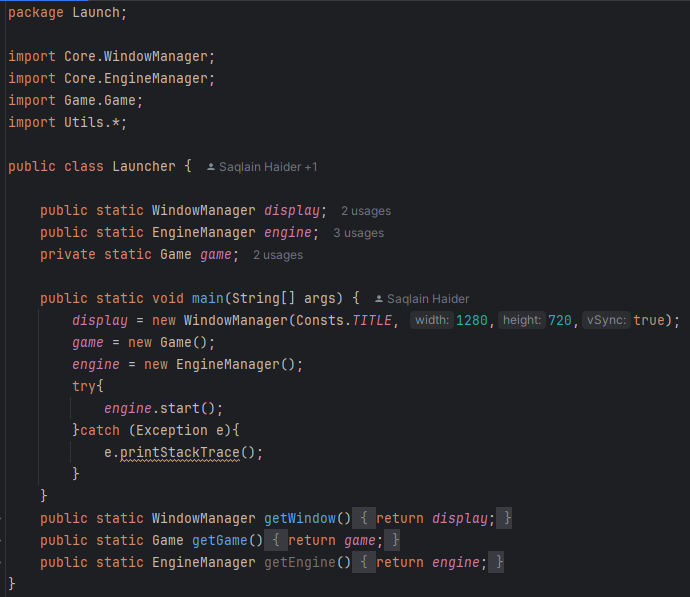
**Saqlain Haider (SP24-BSE-109)**

**OpenGL Engine**

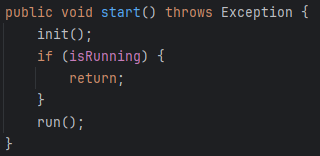
**Overview:**



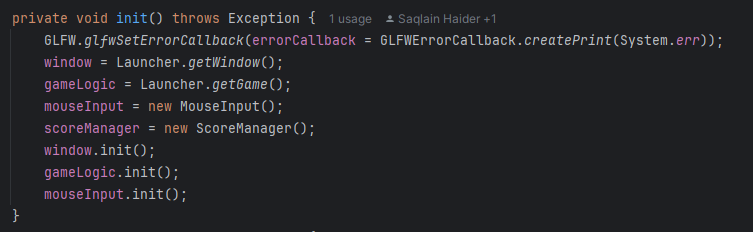
1. The engine is capable of rendering 3D graphics and Directional Light.
2. A simple game is made inside the engine.
3. Ability to detect key events, mouse events etc.
4. Collision detection between valid entities.



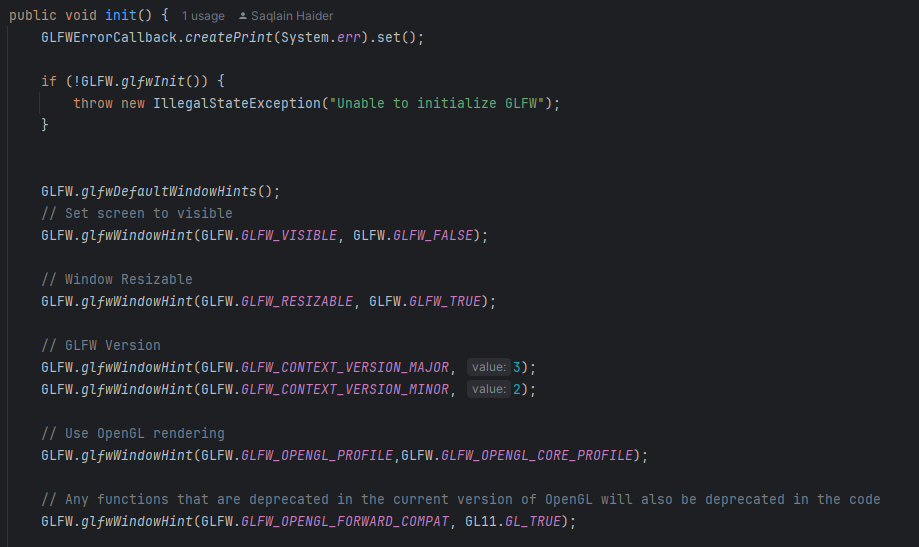
The launcher class creates the display and the instance of the game but the EngineManager class is responsible for displaying the window and the game. The start method is called inside a try catch block as it throws an exception in the EngineManager Class.

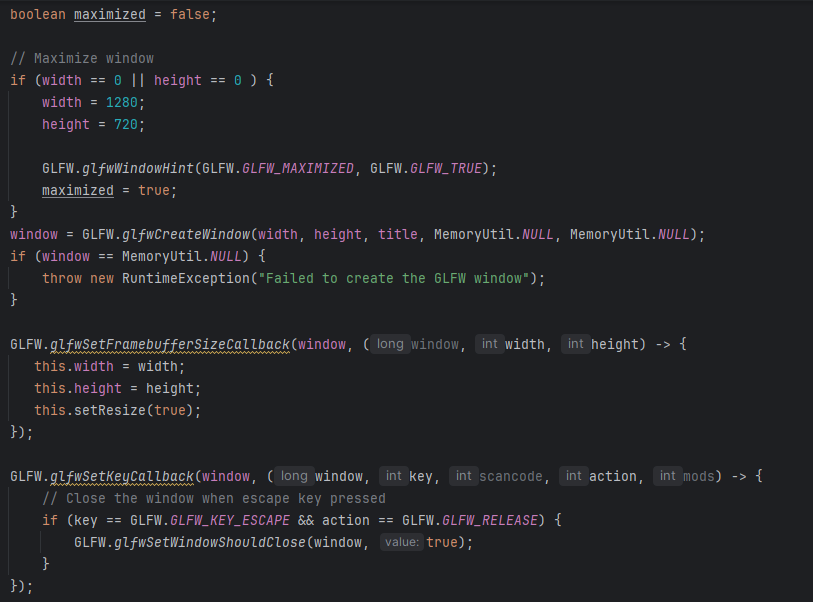


The start method calls the init method to initialize the engine and later runs the engine is not already running



The methods now calls the init method in window (to show the window) the game logic to run the game and mouseInput to enable mouse input.

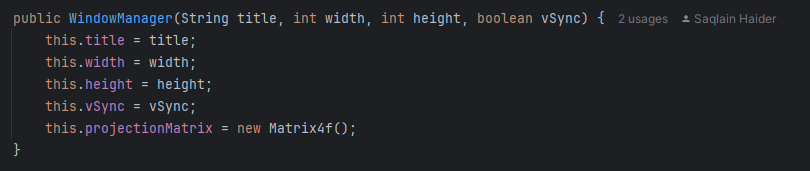




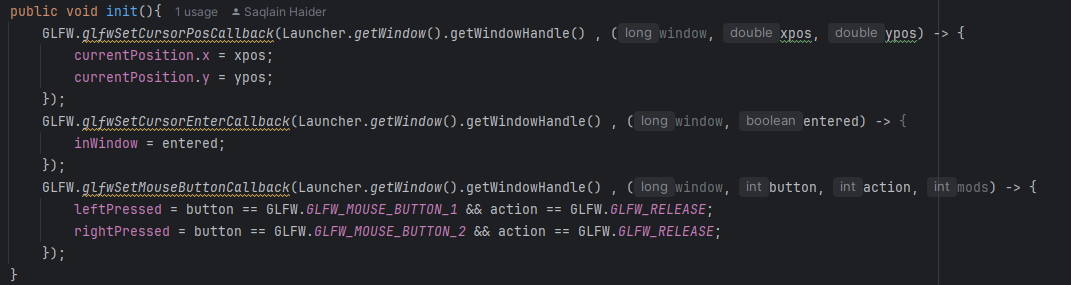
A screen shot of a computer program

Description automatically generated

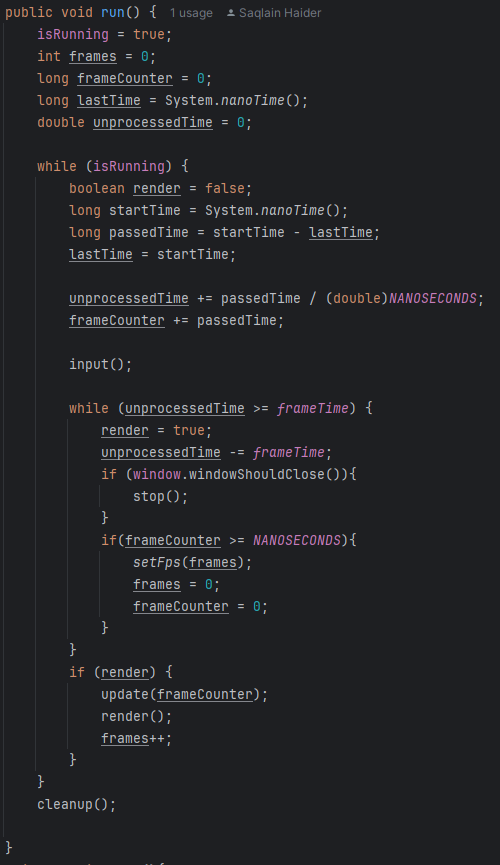
The init method in WindowManager is responsible for displaying the window basic constructor to set the height, width etc.



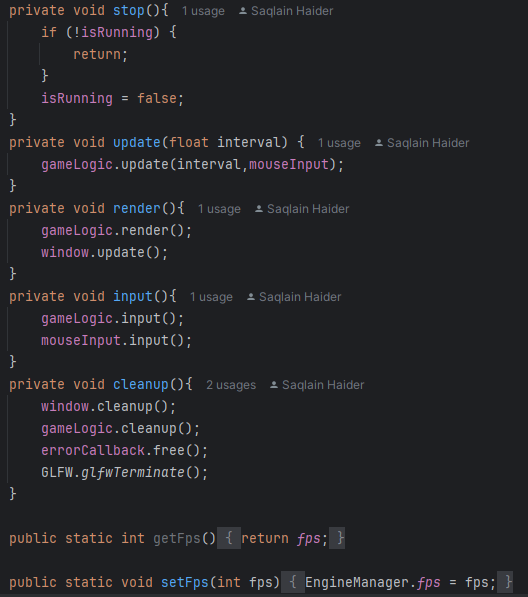
The init method now sets the window to visible, resizable, sets the OpenGL versions the upper bound and the lower bound etc. If the height and width is 0,0 the screen is maximized. Later creates the screen and throws and exception if unable to create screen. Calls events when a key is pressed, and the window is resized. If screen not maximized it centers the screen using vidmode.



The mouseInput init() the events exits which are called when ever the mouse position changes on screen, enters the screen, left or right button is pressed.



The run function sets isRunning to true and handles the main gameloop. Input() is called every iteration of the of the while loop. A basic timer judges the time elapsed between each frame. And only runs the next frame when the time passed between frames is greater or equal to the time each frame should run after (frame time). Frame time is calculated by 1s / FPS (how many frames we desire to show in 1 second) eg. If frames are 60 so 1/60 = 0.01667 is the delay between each frame this is the time the loop waits to call the next iteration of update() and render(). Stop function is called when escape key is pressed (checked from window class). Calls cleanup when breaks out of the loop i.e. engine is closed and not running engine runs first but closes in the end.



Update(), render(), input(), cleanup() calls the update/ input/ cleanup/ render in gameLogic, mouseInput and window

Renderer:

The gameInput() class