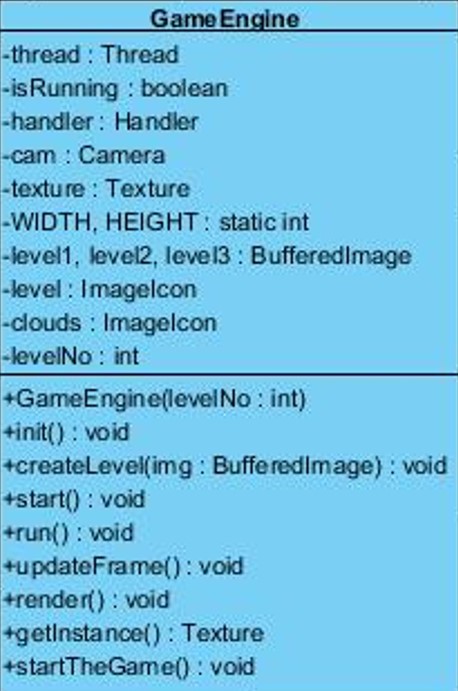
### 3.6 Window Subsystem

#### GameEngine class



This class is the base class for all the entity and control objects, it is responsible for initializing all the power-ups and enemy objects as well as setting up the background and the obstacles for each stage. It takes input from the user through the InputManager, and passes that information to the Player for movement.

*Constructor*

**public GameEngine(int levelNo):** This constructor is responsible for initializing the level.

*Attributes*

**private boolean running:** this attribute holds the boolean value to see if the game is running or not. This attribute will be used to stop the game engine for creating a new Thread for the game in the game loop.

**private Thread thread:** This attribute holds the thread object that is going to start during the game play.

**private Handler handler:** This attribute holds an Handler object.

**private Camera cam;**

**static Texture texture;** This attribute initialize an static Texture object.

**public static int WIDTH, HEIGHT: This attribute**

**private BufferedImage level1, level2, level3:** This attributes holds the background images of their levels.

**private ImageIcon level:**

**private ImageIcon clouds:** This attribute holds the images of clouds.

**private int levelNo:** This attribute holds the value for the level that is going to be chosen by the player before game play.

*Methods*

**public void init():** This method isfor initializing all the entity and control objects of the game and for setting up display for the game play according to the level.

**public synchronized void start():** This method is responsible for starting the threads.

**public void run():** This method calls updateframe and render methods to update the frame when the player is running.

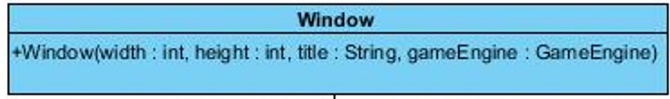
**public void render():** This method will perform the rendering of the animations through the BufferedImage so that we increase the performance of the game.

**private void updateFrame():** This method is for updating the frame.

**public static Texture getInstance():** This method returns the texture object.

**public void startTheGame():** This method initiates the game play and load the display. It is executed after the player has pressed the “play game” button.

#### Window class



*Constructor*

**public Window (int width, int height, String title, GameEngine gameEngine):** This constructor sets the size, frames and images in a window. Its sets a window with the parameters width and height with a title of the parameter title and starts the gameEngine.