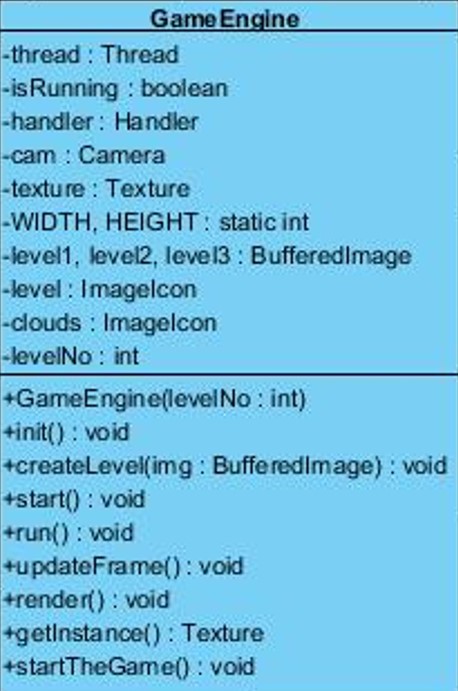
### 3.6 Window Subsystem

#### 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 Thread thread :** this attribute holds the thread object for the game.

**private boolean isRunning :** this attribute holds the boolean value of player if he is running or not.

**private Handler handler:** this attribute holds the handler object for event handling.

**private Camera cam:** this attribute holds the camera object of the game

**static Texture texture :** this attribute holds the Texture object.

**public static int WIDTH, HEIGHT:** these attributes holds the width and height of the image of player.

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

**private ImageIcon level:** holds the image for the Level of the game

**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.it loads the images for each of the levels, then initializes texture, handler and camera objects. Then displays the image according to the level selected by the player. Then sends the handler object to the inputManager instance and adds that to the keyListener.

**public void createLevel( BufferedImage img ):** this methods reads a picture pixel by pixel and adds an object to the frame in place of the pixel. The color of the pixel represents what kind of object it is, for example: player, block, enemy or powerUp.

**public synchronized void start():** This method is responsible for creating a new thread and starting the thread, if there isnt one running already.

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

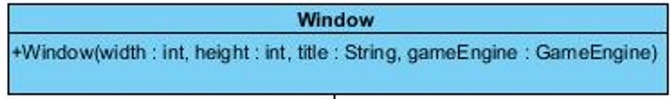
**public void render():** This method will perform the rendering of the animations through the BufferedImage so that we increase the performance of the game.this method draws the graphics onto the screen by calling drawImage method.

**private void updateFrame():** This method is for updating the frame by calling the hnadler’s updateFrame method.

**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. This method creates a new window for the gameplay.

#### 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.