# Project Overview

# This is a desktop-based clone of the popular Flappy Bird game, developed using Java and Swing. The player controls a bird navigating through pairs of pipes without collision. The game implements basic physics, scoring logic, and sprite rendering.

# Tools and Technologies Used

* **Java SE (JDK 8 or above)**
* **Java Swing** (for GUI)
* **AWT** (Graphics and Events)
* **NetBeans / IntelliJ / VS Code** (Any IDE of your choice)
* **Image Resources** (PNG format for sprites)

# Features

#  Player controls bird with Spacebar

#  Gravity & velocity physics for realistic motion

#  Dynamic pipe generation with random gaps

#  Score tracking

#  Game Over and restart on spacebar press

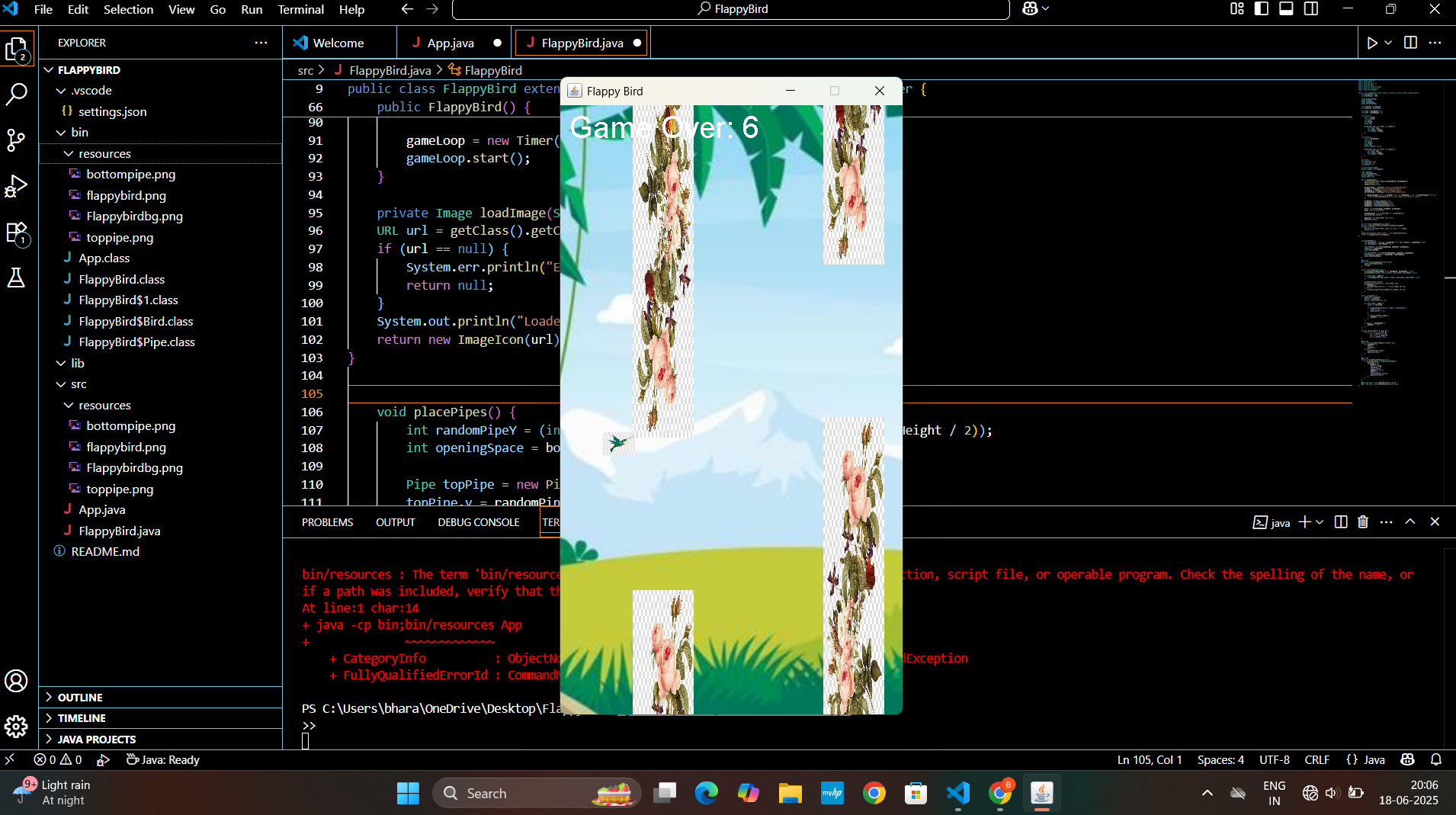
#  Smooth 60 FPS rendering using Timer

# How to Run the Project

1. Open VS Code and create new project named FlappyBird
2. Create folder: src (Inside src, create: App.java, FlappyBird.java, resources(Add your image files here: flappybird.png, toppipe.png, bottompipe.png, Flappybirdbg.png))

flappybird.png

bottompipe.png Flappybirdbg.png toppipe.png

 FlappyBird/

├── src/

│ ├── App.java

│ ├── FlappyBird.java

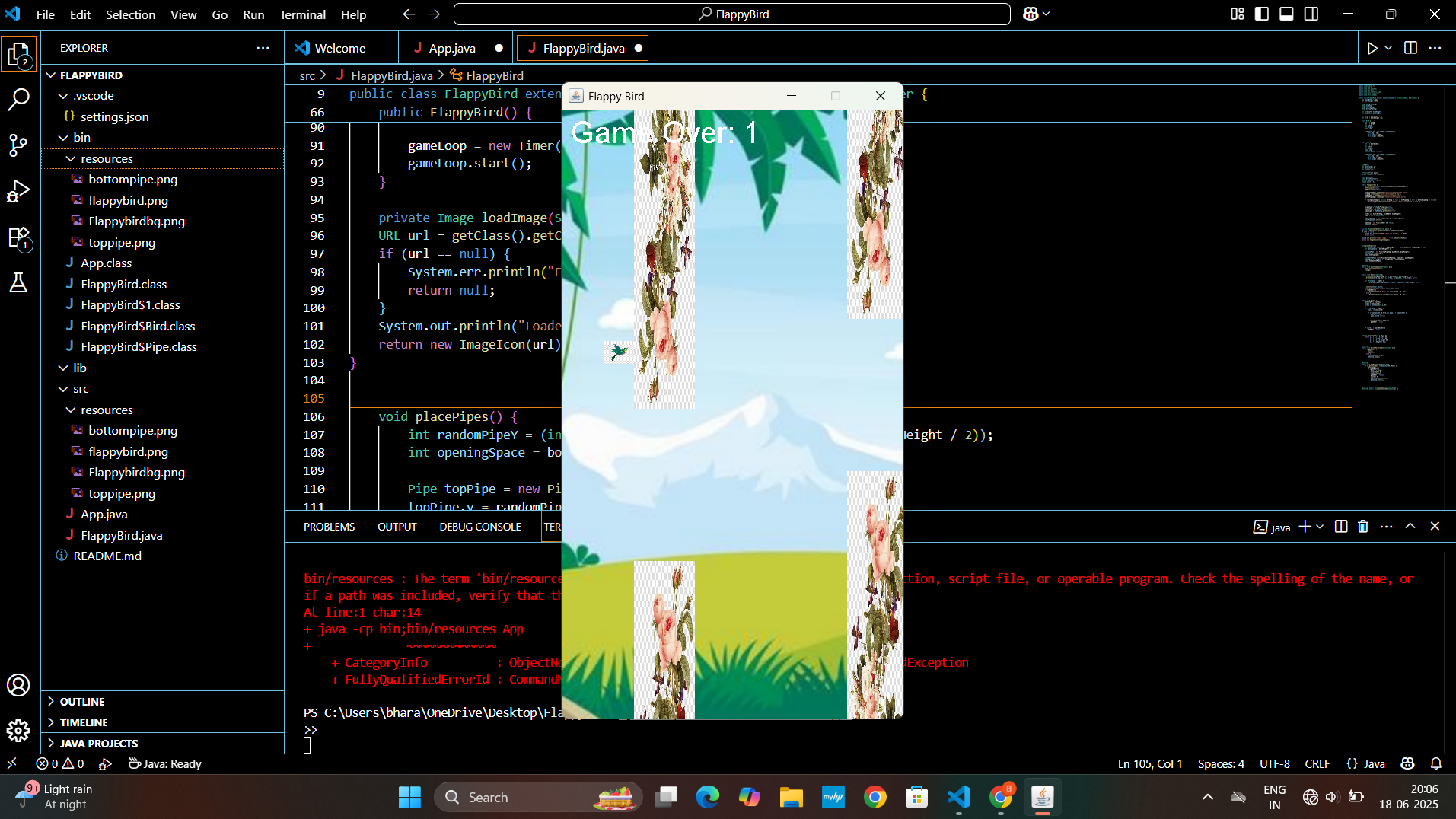
│ └── resources/

│ ├── flappybird.png

│ ├── toppipe.png

│ ├── bottompipe.png

│ └── Flappybirdbg.png

1. Copy the given code in App.java and FlappyBird.java
2. Compile your code: javac -d bin -sourcepath src src/App.java src/FlappyBird.java
3. Run by java -cp bin;src App
4. 

**Possibleerrors**locating the image

Exception in thread "main" java.lang.NullPointerException: Cannot invoke "java.net.URL.toExternalForm()" because "location" is null

at java.desktop/javax.swing.ImageIcon.<init>(ImageIcon.java:232)

at FlappyBird.<init>(FlappyBird.java:77)

at App.main(App.java:15)

solution : run statement changed to java -cp "bin;bin/resources" App

# Code Explanation

#  FlappyBird.java: Main game logic (bird movement, collisions, rendering)

#  App.java: Initializes the game window using JFrame.

#  Bird & Pipe (Inner Classes): Handle individual object properties.

#  paintComponent() and draw() handle all graphics rendering.

#  placePipes() generates top & bottom pipes with a random gap.

#  move() updates bird physics and pipe movement.

#  collision() checks for overlap between bird and pipes.