🐤 Flappy Bird – Java Swing Game

A modernized and team-developed version of Flappy Bird, built in Java using Swing. The game features multiple difficulty levels, real-time scoring, sound effects, and a clean design structure based on object-oriented principles.

# 🎯 Features

* ✅ Responsive gameplay with real-time physics and keyboard controls
* ✅ Difficulty levels: Easy and Hard (affects pipe speed and gap)
* ✅ +10 score per pipe passed, using accurate sensor-based detection
* ✅ Session-based high score tracking
* ✅ Audio feedback:
* • jump.wav when the bird flaps
* • hit.wav when the bird collides
* ✅ Stylized graphics with tiled pipe rendering and retro-styled UI
* ✅ Welcome screen, pause panel, and level selector panel
* ✅ Clean Code Principles followed:
* • Single Responsibility Principle
* • Favor composition over inheritance
* • Modular and maintainable code structure

# 🛠️ Technologies Used

* Java (JDK 8 or later)
* Swing for GUI and rendering
* Java Sound API for audio playback
* Object-Oriented Design principles

# 🚀 How to Run

1. Clone the Repository: git clone https://github.com/your-team-repo/flappy-bird.git
2. Compile the Project: javac \*.java
3. Run the Game: java FlappyBird
4. Controls:
5. - Press UP arrow to flap
6. - Press SPACE to start/unpause

# 📂 Project Structure

* Bird.java # Bird physics and rendering
* FlappyBird.java # Game controller and main logic
* GamePanel.java # Rendering and UI management
* jump.wav, hit.wav # Sound files
* pipe.png # Pipe body sprite
* pipe\_part.png # Pipe head/cap sprite
* menu\_bg.png # Background for level select
* background.png # Main game background
* README.md # This file

# 👨‍💻 Team & Contributions

Each team member contributed to code, design, and documentation through Git commits and code reviews. All logic was built collaboratively under shared ownership, adhering to version control best practices.

# 📋 License

This project is for educational and academic use only.

# ✅ TODO (Post-Beta)

* Persistent high score (using file serialization)
* Add leaderboard panel
* Add power-ups or advanced game modes
* Mute toggle for sound