BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**MÔN HỌC: Lập trình Java**

**ĐỀ TÀI: Trò chơi Flappy Bird**

**SINH VIÊN: Nguyễn Hữu Trọng – 65133958 – 65.CNTT-CLC**

**GIẢNG VIÊN: Phạm Văn Nam**

**HỌC KỲ II (2024 - 2025)**

Nha Trang, ngày 21 tháng 4 năm 2025

[1 Giới thiệu Project 3](#_Toc196137339)

[1.1 Mục tiêu Project 3](#_Toc196137340)

[1.2 Phạm vi và giới hạn của Project 3](#_Toc196137341)

[1.3 Công nghệ sử dụng trong Project 3](#_Toc196137342)

[2 Phân tích yêu cầu 3](#_Toc196137343)

[2.1 Yêu cầu chức năng 3](#_Toc196137344)

[2.1.1 Đăng nhập người chơi 3](#_Toc196137345)

[2.1.2 Chơi game Flappy Bird 3](#_Toc196137346)

[2.1.3 Tăng độ khó theo điểm 4](#_Toc196137347)

[2.1.4 Hiển thị Game Over 4](#_Toc196137348)

[2.1.5 Lưu trữ dữ liệu 4](#_Toc196137349)

[2.2 Yêu cầu phi chức năng 4](#_Toc196137350)

[2.3 Use case diagram và Mô tả 4](#_Toc196137351)

[2.4 Sequence Diagram cho các chức năng chính 5](#_Toc196137352)

[2.4.1 Sequence: Đăng nhập 5](#_Toc196137353)

[2.4.2 Sequence: Chơi game 5](#_Toc196137354)

[2.4.3 Sequence: Ghi điểm 5](#_Toc196137355)

[3 Thiết kế 5](#_Toc196137356)

[3.1 Kiến trúc tổng thể 5](#_Toc196137357)

[3.2 Class Diagram 6](#_Toc196137358)

[3.3 Database Schema 6](#_Toc196137359)

[3.4 Giao diện người dùng 7](#_Toc196137360)

[3.4.1 LoginRegisterFrame.java 7](#_Toc196137361)

[3.4.2 FlappyBird.java 8](#_Toc196137362)

[4 Cài đặt 9](#_Toc196137363)

[4.1 Cấu trúc Source Code 9](#_Toc196137364)

[4.2 Các thuật toán/ kỹ thuật quan trọng 9](#_Toc196137365)

[4.3 Code mẫu cho các chức năng chính 10](#_Toc196137366)

[4.3.1 Tính điểm và tăng độ khó 10](#_Toc196137367)

[4.3.2 Va chạm 10](#_Toc196137368)

[4.3.3 Lưu điểm 11](#_Toc196137369)

[4.4 Hướng dẫn build và chạy chương trình 11](#_Toc196137370)

[4.4.1 Cài đặt môi trường 11](#_Toc196137371)

[4.4.2 Biên dịch 11](#_Toc196137372)

[4.4.3 Chạy chương trình 11](#_Toc196137373)

[5 Kiểm thử 11](#_Toc196137374)

[5.1 Kế hoạch kiểm thử 11](#_Toc196137375)

[5.2 Kết quả kiểm thử 12](#_Toc196137376)

[5.3 Demo các chức năng 12](#_Toc196137377)

[6 Kết luận 16](#_Toc196137378)

[6.1 Kết quả đạt được 16](#_Toc196137379)

[6.2 Hạn chế và hướng phát triển 16](#_Toc196137380)

[6.2.1 Hạn chế 16](#_Toc196137381)

[6.2.2 Hướng phát triển 16](#_Toc196137382)

[6.3 Bài học kinh nghiệm 17](#_Toc196137383)

[7 Tài liệu kham khảo 17](#_Toc196137384)

# Giới thiệu Project

## Mục tiêu Project

* Xây dựng một trò chơi Flappy Bird bằng ngôn ngữ Java, mô phỏng theo cách chơi đơn giản nhưng gây nghiện của bản gốc.
* Rèn luyện kỹ năng lập trình hướng đối tượng, xử lý sự kiện, đồ họa, và quản lý dữ liệu người chơi.

## Phạm vi và giới hạn của Project

* Phạm vi:
* Phát triển giao diện chơi game Flappy Bird với các chức năng cơ bản: điều khiển chim, tạo chướng ngại vật, tính điểm và hiển thị điểm.
* Thêm chức năng đăng nhập người chơi, hiển thị điểm cao nhất theo từng người chơi.
* Tạo giao diện Game Over hiển thị điểm hiện tại, điểm cao và thông báo nếu đạt kỷ lục mới.
* Giới hạn:
* Game chỉ hỗ trợ chế độ chơi đơn, không có chế độ nhiều người chơi cùng lúc.
* Dữ liệu người chơi được lưu cục bộ (bằng file), chưa kết nối đến cơ sở dữ liệu từ xa.
* Giao diện đơn giản.

## Công nghệ sử dụng trong Project

* Ngôn ngữ lập trình: Java.
* Thư viện sử dụng: Java AWT và Swing để xử lý đồ họa 2D và giao diện.
* Môi trường phát triển: IntelliJ IDEA / Eclipse.
* Quản lý dữ liệu: File I/O (đọc ghi file để lưu điểm cao).
* Mô hình lập trình: Hướng đối tượng (OOP).

# Phân tích yêu cầu

## Yêu cầu chức năng

### Đăng nhập người chơi

* Cho phép người dùng nhập tên trước khi chơi.
* Nếu là người chơi mới thì tạo hồ sơ mới.

### Chơi game Flappy Bird

* Người chơi nhấn phím Space để điều khiển chim bay.
* Chim phải tránh va chạm với cột và mặt đất.
* Tính điểm theo số cột vượt qua.

### Tăng độ khó theo điểm

* Khi điểm tăng:
* Các cột sẽ di chuyển lên xuống nhanh dần.
* Khoảng cách trên dưới giảm dần.
* Tốc độ di chuyển tăng.

### Hiển thị Game Over

* Hiện điểm hiện tại, điểm cao nhất.
* Thông báo nếu đạt kỷ luật mới.

### Lưu trữ dữ liệu

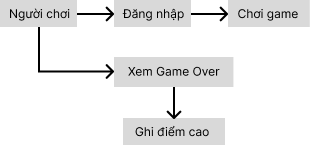
* Ghi lại điểm cao nhất cho từng người chơi bằng file.

## Yêu cầu phi chức năng

* Hiệu năng: Game chạy mượt ở 60 FPS trên máy cấu hình phổ thông.
* Tính dễ dùng: Giao diện đơn giản, dễ thao tác bằng bàn phím.
* Bảo trì: Mã nguồn tách lớp rõ ràng theo hướng đối tượng, dễ mở rộng.
* Lưu trữ: Dữ liệu người chơi được lưu an toàn bằng tệp cục bộ.

## Use case diagram và Mô tả

* Use Case Diagram: Người dùng 🡪 (1) Đăng nhập 🡪 (2) Chơi game 🡪 (3) Xem Game Over 🡪 (4) Ghi điểm



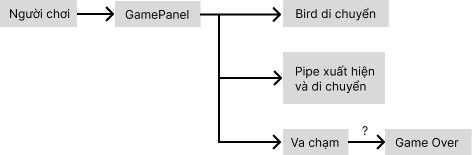
* Mô tả Use Case chính:
* Đăng nhập: Người chơi nhập tên 🡪 Kiểm tra tên có tồn tại không 🡪 Nếu chưa có tạo mới.
* Chơi game: Người chơi điều khiển chim bằng phím 🡪 Vượt cột 🡪 Tăng điểm 🡪 Nếu va chạm thì kết thúc.
* Game Over: Hiển thị điểm và so sánh với điểm cao nhất 🡪 Nếu cao hơn thì ghi đè.

## Sequence Diagram cho các chức năng chính

### Sequence: Đăng nhập



### Sequence: Chơi game



### Sequence: Ghi điểm

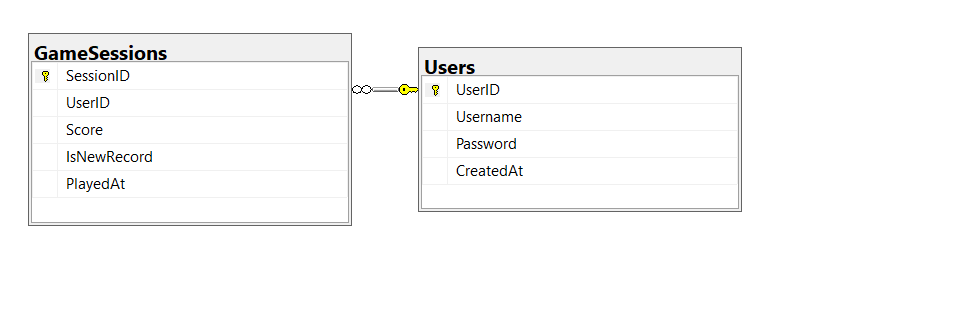


# Thiết kế

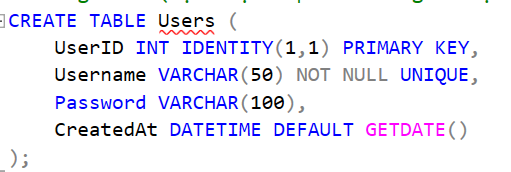
## Kiến trúc tổng thể

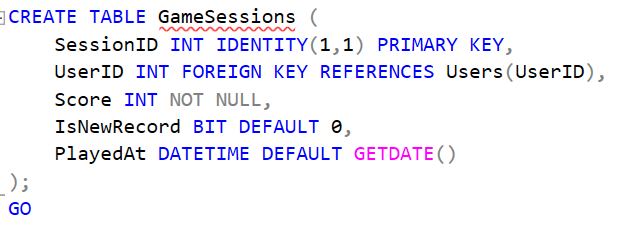
* Ứng dụng được thiết kế theo kiến trúc hướng đối tượng (OOP), chia thành các lớp riêng biệt theo chức năng:
* Hiển thị và điều khiển game: FlappyBird.java, AnimatedBackground.java
* Giao diện người dùng: LoginRegisterFrame.java
* Xử lý logic: AuthManager.java, SoundPlayer.java
* Điểm khởi chạy: App.java

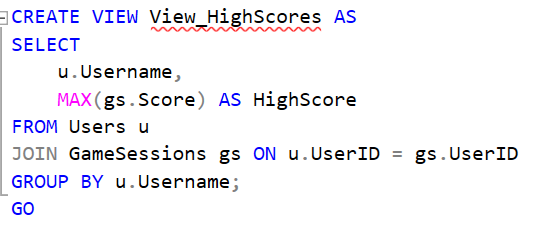
## Class Diagram



## Database Schema



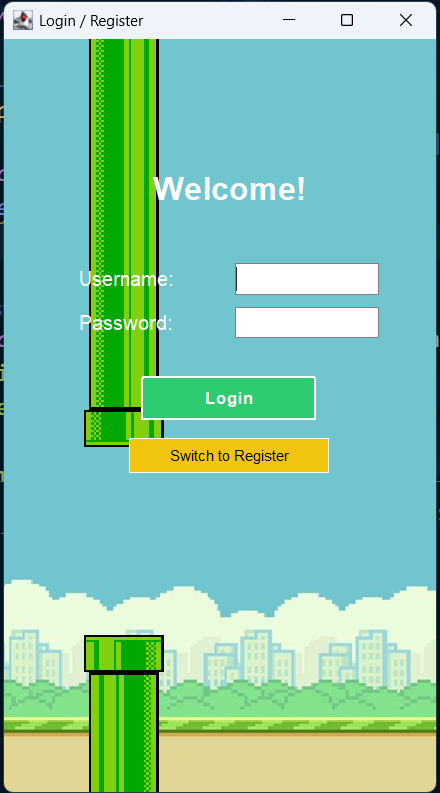




## Giao diện người dùng

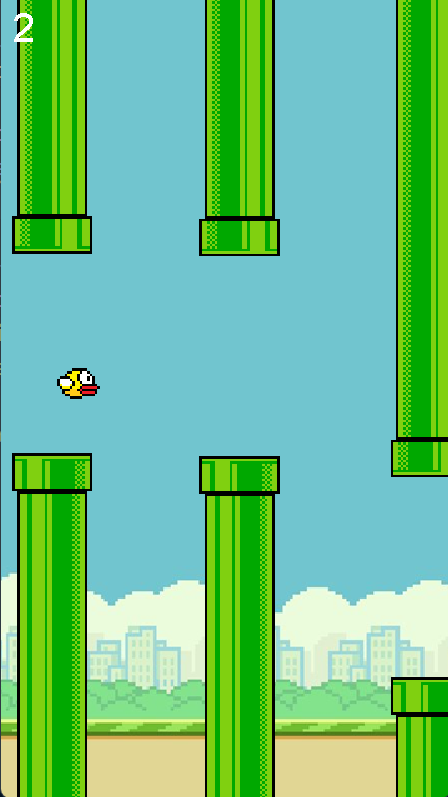
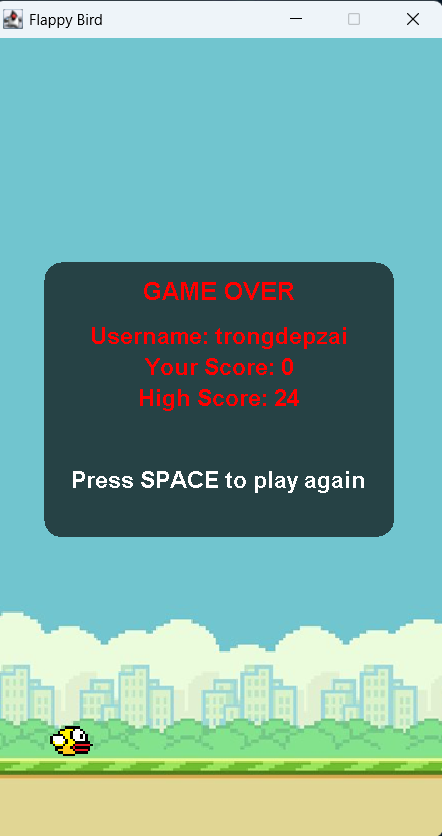
### LoginRegisterFrame.java

* Hiển thị màn hình đăng nhập hoặc đăng kí người chơi.
* Nhập username và password.
* Giao diện đơn giản bằng JFrame, JTextField, JButton.
* Lớp điều khiển chính (Main): Khởi tạo giao diện và quản lý chuyển đổi giữa các màn hình.



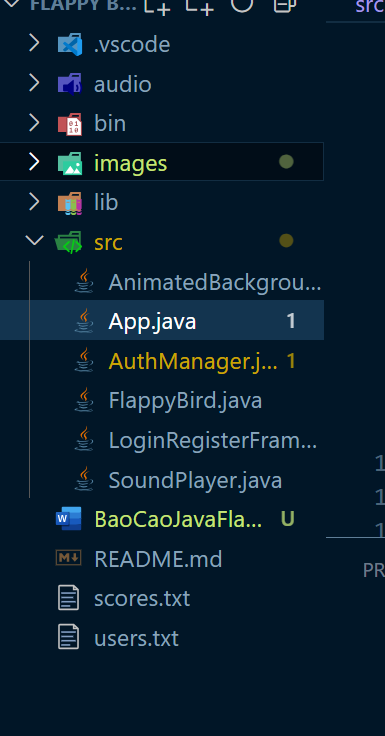
### FlappyBird.java

* Giao diện chơi game: Chim bay, cột di chuyển.
* Hiển thị điểm hiện tại.
* Hiện Game Over và điểm cao.



# Cài đặt

## Cấu trúc Source Code

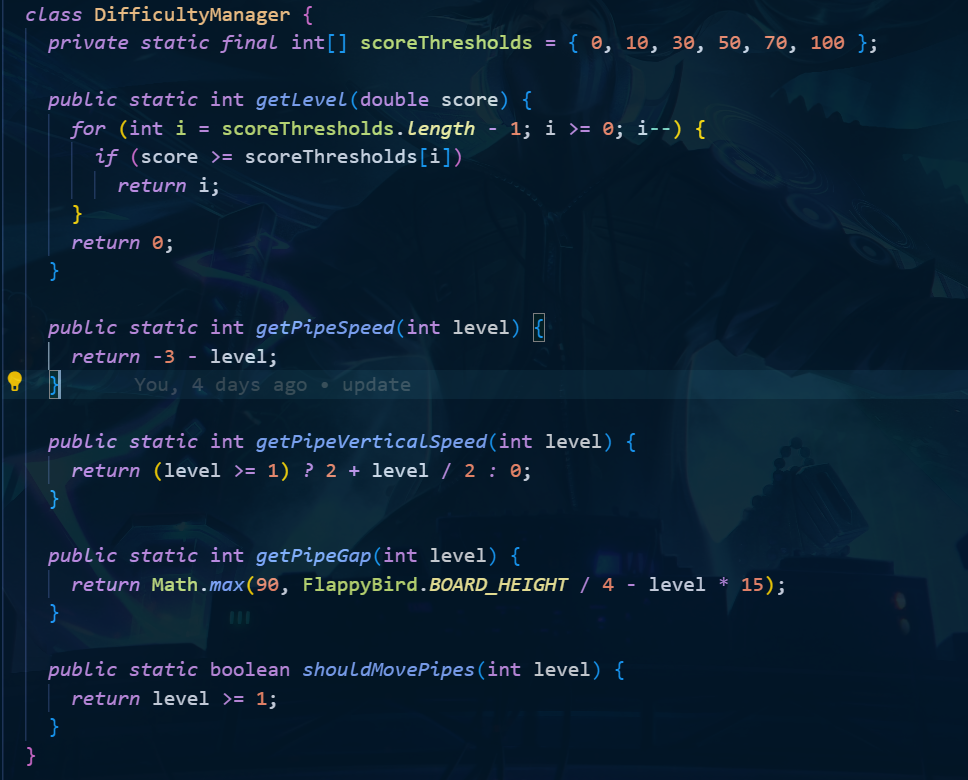


## Các thuật toán/ kỹ thuật quan trọng

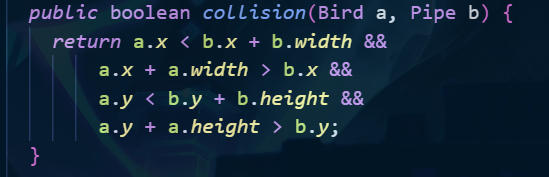
* Dựa theo điểm số, tốc độ và độ khó tăng dần (qua DiffcultyManager).
* Va chạm: Sử dụng bounding box giữa chim và ống.
* Giao diện Swing: Vẽ hình nền, chim, ống và chữ bằng Graphics.
* Lưu điểm cao: Dùng AuthManager.saveHighScore() để ghi điểm theo tên người chơi.
* Âm thanh: Dùng SoundPlayer.play() để phát hiệu ứng.

## Code mẫu cho các chức năng chính

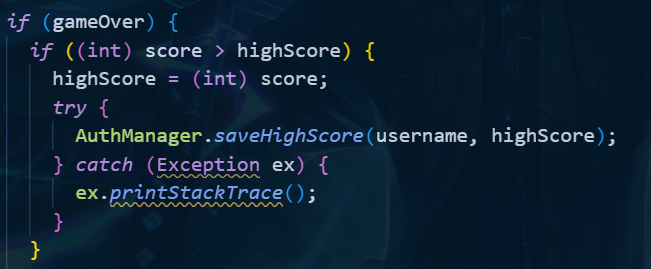
### Tính điểm và tăng độ khó



### Va chạm



### Lưu điểm



## Hướng dẫn build và chạy chương trình

### Cài đặt môi trường

* JDK 17 hoặc cao hơn.
* IDE: IntelliJ IDEA hoặc VS Code.
* Thư mục images/ và audio/ phải được đặt trong cùng thư mục với .java

### Biên dịch

* javac \*.java

### Chạy chương trình

* java FlappyBird

# Kiểm thử

## Kế hoạch kiểm thử

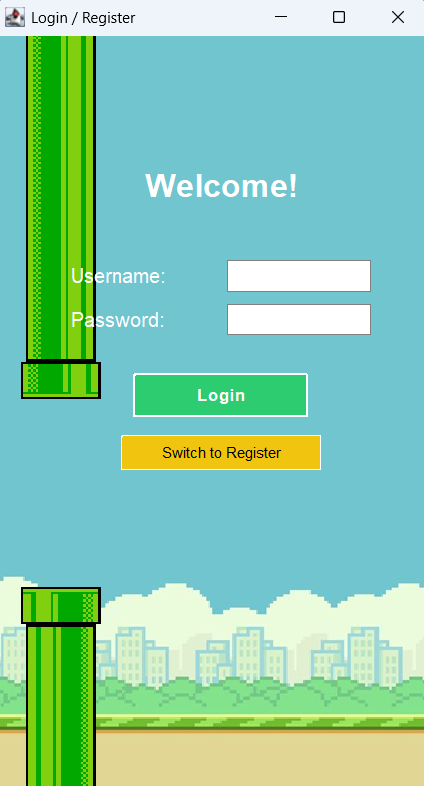
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Chức năng kiểm thử | Mô tả | Dữ liệu đầu vào | Kết quả mong đợi |
| 1 | Đăng nhập / Đăng ký | Nhập tên người chơi và mật khẩu. | “trongdepzai”  “123456” | Thông báo đăng nhập/đăng kí thành công/không thành công |
| 2 | Bắt đầu game | Nhấn phím space | Không có | Game bắt đầu, ống di chuyển, chim rơi dần. |
| 3 | Va chạm ống | Chim chạm ống | Không có | Game Over xuất hiện |
| 4 | Va chạm nền đất | Chim chạm nền đất | Không có | Game Over xuất hiện |
| 5 | Tính điểm | Chim vượt qua ống | Không có | Điểm tăng đúng |
| 6 | Hiển thị điểm | Điểm trên góc trái màn hình | Không có | Điểm tăng đúng theo hiện tại |
| 7 | Game Over | Hiện màn hình Game Over | Không có | Hiển thị username, điểm hiện tại, điểm cao nhất |
| 8 | Lưu điểm theo user | Chơi lại nhiều lần với cùng username | Nhiều lần chơi | Điểm cao nhất được cập nhật đúng |
| 9 | Phá kỉ luật | Điểm hiện tại cao hơn điểm cao nhất của user | Không có | Hiển thị username, kỉ lục mới, điểm cao nhất |
| 10 | Tăng độ khó | Điểm đạt 10, 30, 50, 70, 100 | Không có | Tốc độ tăng, khoảng cách ống thay đổi, ống di chuyển. |
| 11 | Nhạc nền & Hiệu ứng | Khi chơi và va chạm | Không có | Nhạc nền phát khi đăng nhập/đăng kí, khi chơi, âm thanh khi bay, khi tăng điểm, khi va chạm, khi thua, khi bắt đầu game, khi bắt đầu lại game. |

## Kết quả kiểm thử

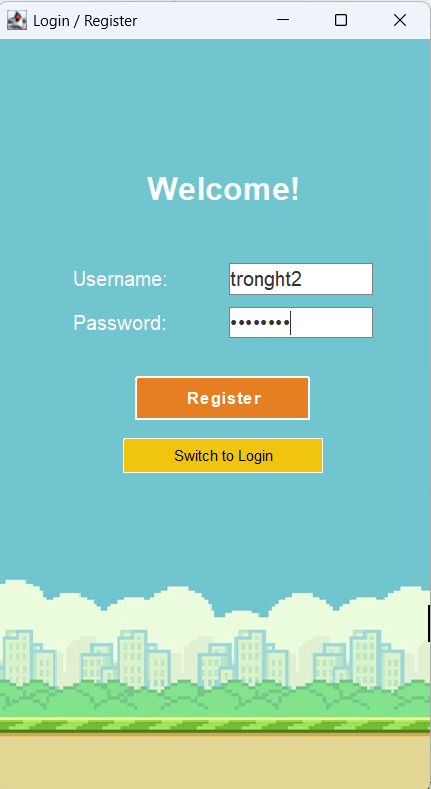
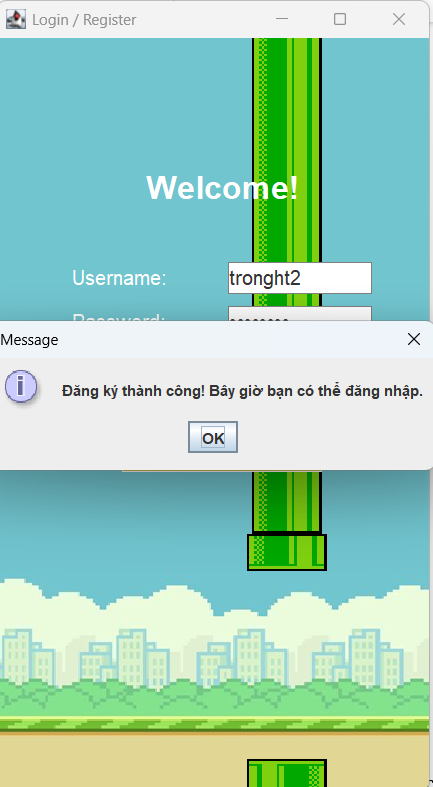
* Tất cả các chức năng đều hoạt động đúng như kế hoạch kiểm thử.
* Không phát hiện lỗi nghiêm trọng.
* Độ khó thay đổi rõ rệt theo điểm số.
* Giao diện mượt, không giật lag.
* Nhạc nền và âm thanh hoạt động tốt.

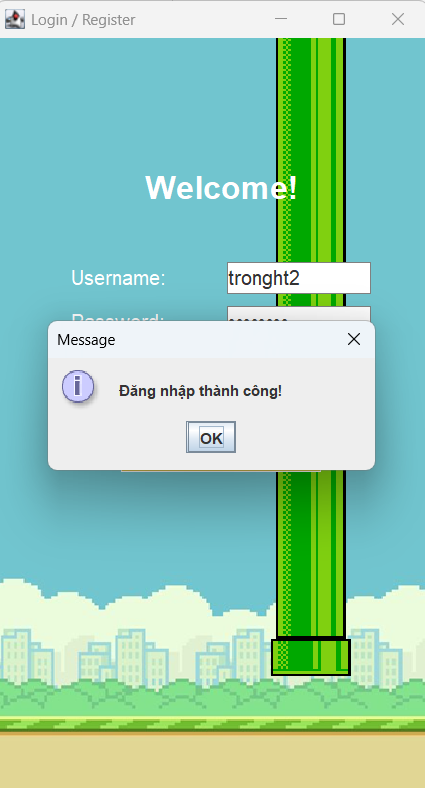
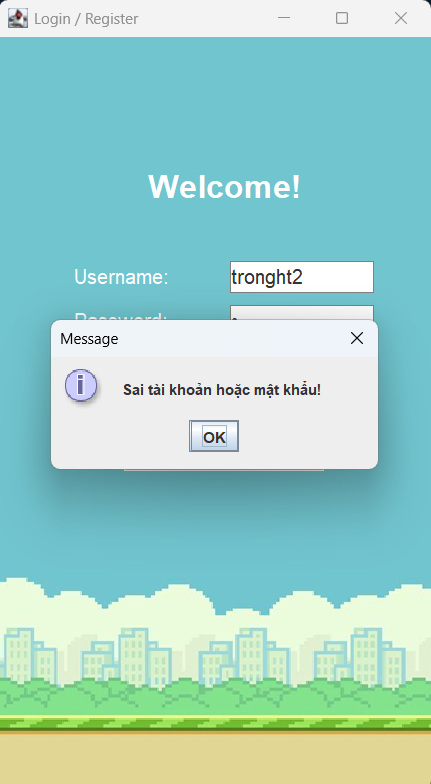
## Demo các chức năng

B1: Giao diện khi khởi chạy chương trình

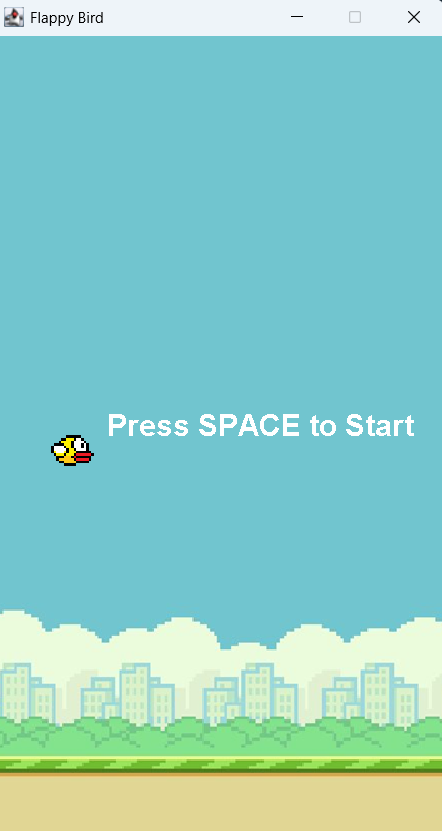


B2: Đăng nhập tài khoản nếu chưa có thì đăng kí, đăng kí thành công sẽ hiện thông báo. Và đăng nhập thành công/không thành công cũng hiện thông báo.

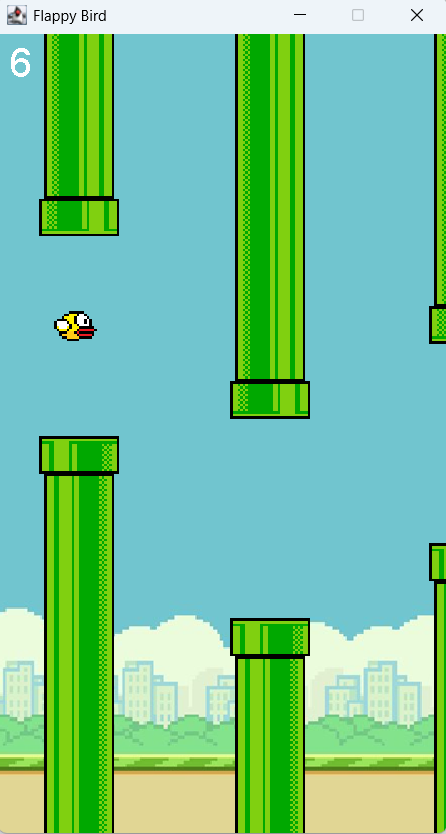
 

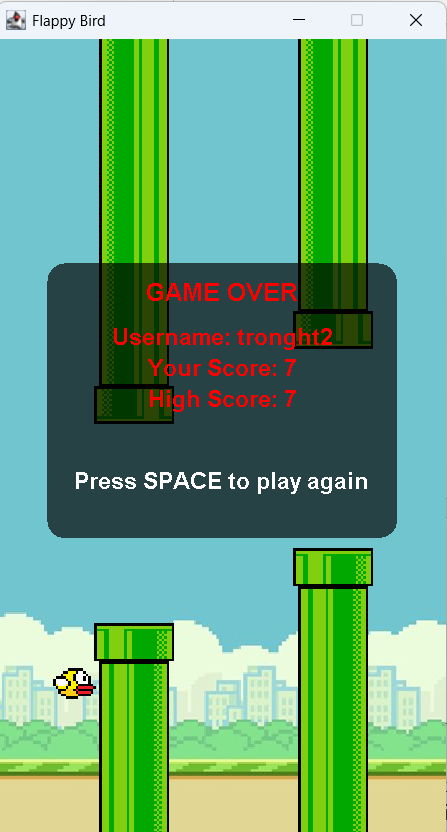
B3: Màn hình game khi bắt đầu, nhấn space để chạy game



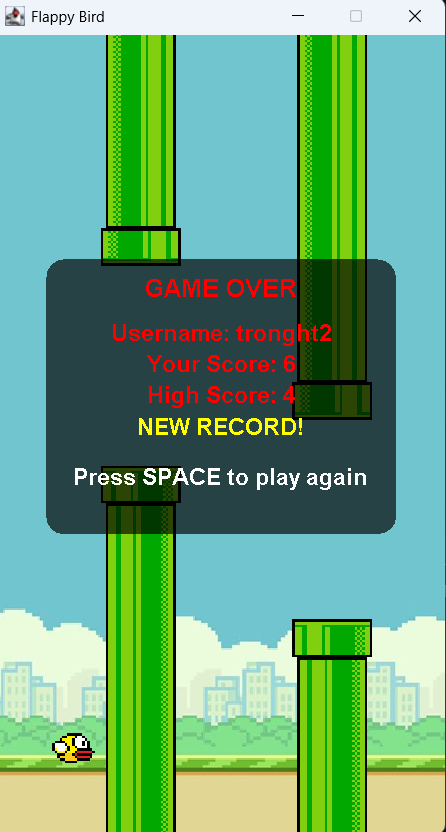
B4: Người chơi sẽ chơi game, mỗi lần qua 1 cột sẽ cộng 1 điểm, điểm được hiển thị bên góc trái trên màn hình



B5: Nếu thua(khi chạm cột hoặc chạm đất) sẽ hiện thông báo game over, game over sẽ hiện tên người chơi, điểm cao nhất và điểm hiện tại.



B6: Khi phá kỉ luật thì trong game over sẽ hiện thêm “NEW RECORD!”



B7: Người dùng ấn space để chơi lại, và tiếp tục phá kỉ luật.

# Kết luận

## Kết quả đạt được

* Hoàn thiện game Flappy Bird bằng ngôn ngữ Java với đầy đủ chức năng cơ bản: bay qua ống, tính điểm, Game Over.
* Thêm chức năng đăng nhập bằng username, lưu điểm cao cho từng người chơi.
* Tích hợp nhạc nền và hiệu ứng âm thanh, tăng trải nghiệm người chơi.
* Giao diện mượt mà, dễ sử dụng, trải nghiệm ổn định.

## Hạn chế và hướng phát triển

### Hạn chế

* Giao diện đơn giản, chưa có hiệu ứng hình ảnh phong phú.
* Chưa có chức năng bảng xếp hạng điểm cao giữa các người chơi.
* Dữ liệu người dùng lưu cục bộ, chưa kết nối cơ sỡ dữ liệu online.

### Hướng phát triển

* Thiết kế thêm bảng xếp hạng top người chơi.
* Cải tiến đồ họa, thêm nhân vật (chim), hiệu ứng chuyển động.
* Đưa game lên nền tảng web hoặc mobile, cho phép chơi online.

## Bài học kinh nghiệm

* Nắm vững cách tổ chức dự án Java theo mô hình hướng đối tượng.
* Biết cách xử lý luồng game, sự kiện bàn phím và âm thanh trong Java.
* Hiểu rõ hơn về cách thiết kế UI bằng Swing.
* Rèn luyện kỹ năng quản lý thời gian, gỡ lỗi và làm việc theo quy trình.
* Biết cách tạo sản phẩm có tính tương tác và lưu trữ dữ liệu người dùng.

# Tài liệu kham khảo

Tác giả: Kenny Yip Coding (<https://github.com/ImKennyYip/flappy-bird-java>)