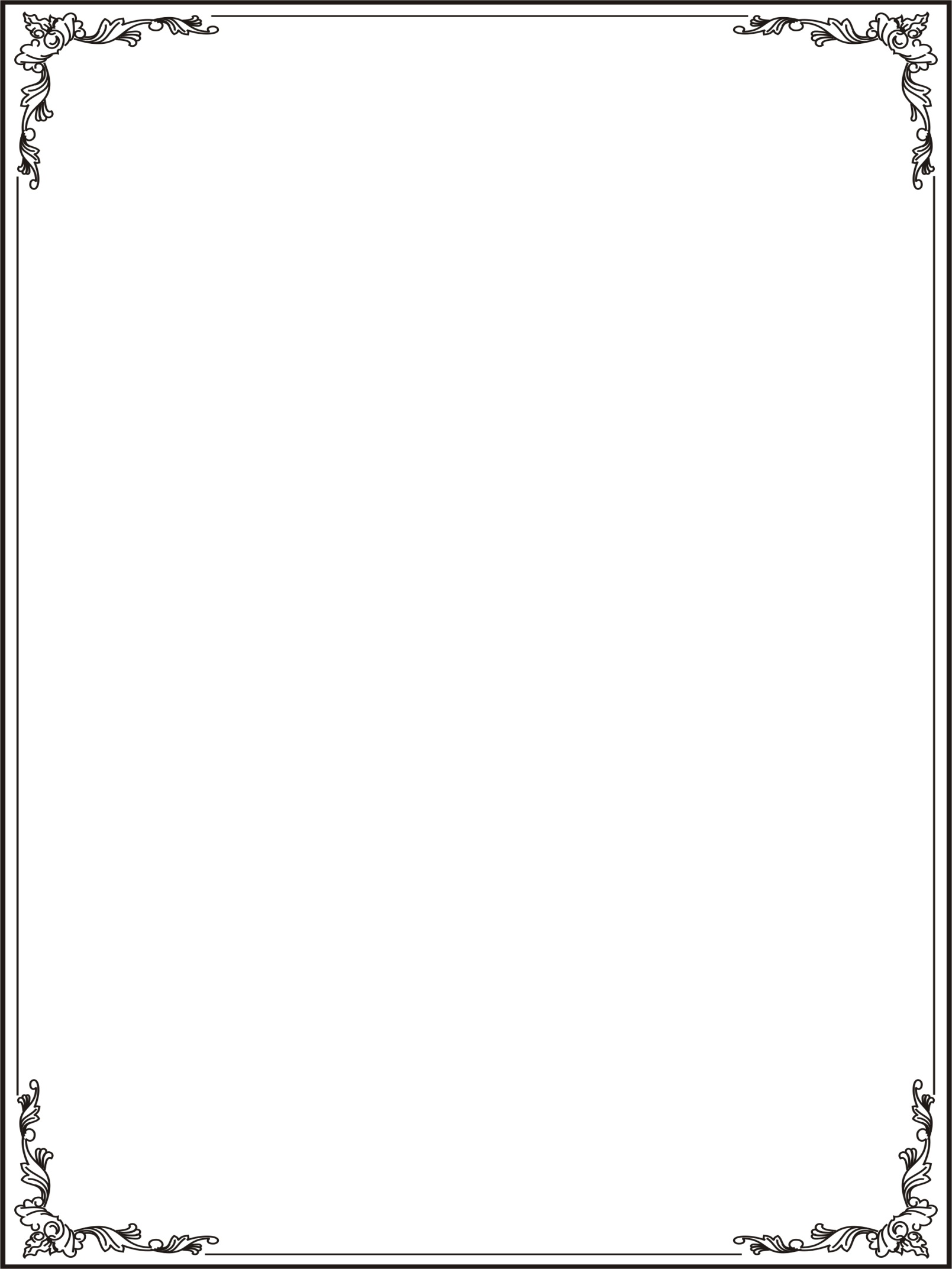
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
VÀ TRUYỀN THÔNG  
BỘ MÔN AN TOÀN HỆ THỐNG THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**Báo cáo môn Lập trình Java**

**Đề tài:** Lập trình game FlappyBird

**Giảng viên:** Trịnh Minh Đức

Nguyễn Văn Quang

**Sinh Viên:** Vũ Thanh Bình

**Lớp:** An toàn thông tin K13A

Thái nguyên, 10/2016

MỤC LỤC

[**TÓM TẮT ĐỀ TÀI** 2](#_Toc465809888)

[**1.** **Tổng quan** 2](#_Toc465809889)

[**2.** **Phạm vi nghiên cứu và phạm vi ứng dụng** 2](#_Toc465809890)

[**3.** **Chức năng công việc đã làm được** 2](#_Toc465809891)

[**I.** **CƠ SỞ LÝ THUYẾT** 3](#_Toc465809892)

[**1.** **Giới thiệu về ngôn ngữ Java** 3](#_Toc465809893)

[**2.** **Một số tính chất của ngôn ngữ Java** 3](#_Toc465809894)

[**II.** **QUÁ TRÌNH XÂY DỰNG GAME** 3](#_Toc465809895)

[**1.** **Thiết kế game** 3](#_Toc465809896)

[**a.** **Giao diện game** 3](#_Toc465809897)

[**b.** **Âm thanh game** 3](#_Toc465809898)

[**2.** **Thực hiện** 4](#_Toc465809899)

[**a.** **Các lớp tự tạo** 4](#_Toc465809900)

[**b.** **Các đối tượng** 13](#_Toc465809901)

[**c.** **Lớp chứa hàm main** 20](#_Toc465809902)

[**III.** **KẾT QUẢ** 24](#_Toc465809903)

[**1.** **Hiểu hơn về ngôn ngữ Java** 24](#_Toc465809904)

[**2.** **Tạo được game FlappyBird hoàn chỉnh** 25](#_Toc465809905)

[**IV.** **HẠN CHẾ THIẾU SÓT VÀ HƯỚNG PHÁT TRIỂN** 25](#_Toc465809906)

[**1.** **Hạn chế** 25](#_Toc465809907)

[**2.** **Hướng phát triển** 25](#_Toc465809908)

[**V.** **TÀI LIỆU THAM KHẢO** 25](#_Toc465809909)

# 

# **TÓM TẮT ĐỀ TÀI**

1. **Tổng quan**

Java là ngôn ngữ lập trình dạng lập trình hướng đối tượng (OOP). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy. Bằng cách này, Java thường chạy chậm hơn những ngôn ngữ lập trình thông dịch khác như C++, Python, PHP, C#.....

Để hiểu rõ hơn về Java em đã tìm hiểu và xây dựng một chương trình đơn giản là “**Game FlappyBird**”. Trò chơi được trình bày theo phong cách side-scroller (phong cách game với các đối tượng được nhìn thấy ở mặt bên (side-view) và di chuyển từ cạnh trái sang cạnh phải của màn hình), trong đó người chơi điều khiển một chú chim, cố gắng vượt qua các hàng ống màu xanh lá cây mà không chạm vào chúng để dành được điểm cao nhất có thể và không có giới hạn về số điểm.

1. **Phạm vi nghiên cứu và phạm vi ứng dụng**

* Phạm vi nghiên cứu: tự nghiên cứu
* Phạm vi ứng dụng: cá nhân

1. **Chức năng công việc đã làm được**

|  |  |
| --- | --- |
| STT | Chức năng, công việc đã làm được |
| 1 | Tạo giao diện thân thiện dễ sử dụng |
| 2 | Tính điểm cho người chơi |
| 3 | Tạo được âm thanh cho trò chơi |
| 4 | Tạo chuyển động cho con chim và ống khói |

1. **CƠ SỞ LÝ THUYẾT**
2. **Giới thiệu về ngôn ngữ Java**

Năm 1990, Sun MicroSystems thực hiện dự án Green nhằm phát triển phần mềm trong các thiết bị dân dụng. James Gosling, chuyên gia lập trình đã tạo ra một ngôn ngữ lập trình mới có tên là Oak. Ngôn ngữ này có cú pháp gần giống như C++ nhưng bỏ qua các tính năng nguy hiểm của C++ như truy cập trực tiếp tài nguyên hệ thống, con trỏ, định nghĩa chồng các tác tử…

Khi ngôn ngữ Oak trưởng thành, WWW cũng đang vào thời kỳ phát triển mạnh mẽ, Sun cho rằng đây là một ngôn ngữ thích hợp cho Internet. Năm 1995, Oak đổi tên thành Java và sau đó đến 1996  Java đã được xem như một chuẩn công nghiệp cho Internet.

1. **Một số tính chất của ngôn ngữ Java**

* Đơn giản
* Hướng đối tượng
* Độc lập phần cứng và hệ điều hành
* Mạnh mẽ
* Bảo mật
* Phân tán
* Đa luồng
* Linh động

1. **QUÁ TRÌNH XÂY DỰNG GAME**
2. **Thiết kế game**

Chương trình có giao diện dễ nhìn, chạy ổn định. Đảm bảo một số tính năng như: tiếng động, tính điểm người chơi, chuyển động của con chim, chuyển động của ống khói, chuyển động của nền và va chạm với ống khói hay nền.

1. **Giao diện game**

Game được thiết kế là một nền có các cặp ống khói ở trên và dưới xếp đều nhau, có một chú chim ở giữa để di chuyển qua các cặp ống khói

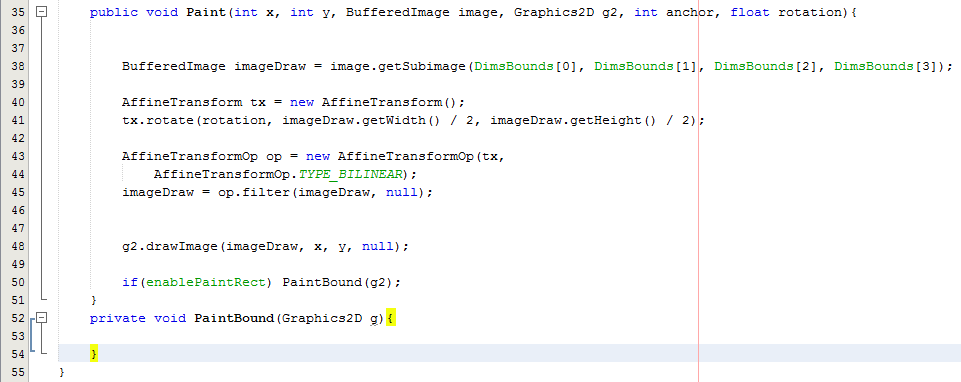
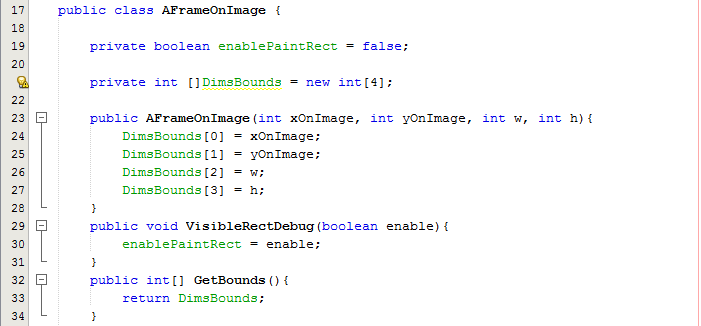
1. **Âm thanh game**

Trong game sử dụng file nhạc .wav để phát âm thanh trong game khi điều khiểu chú chim, va chạm và ăn điểm.

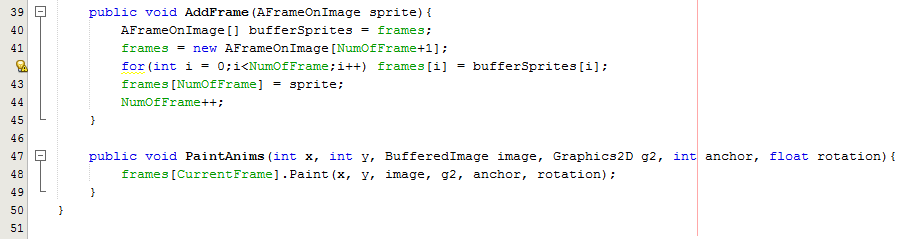
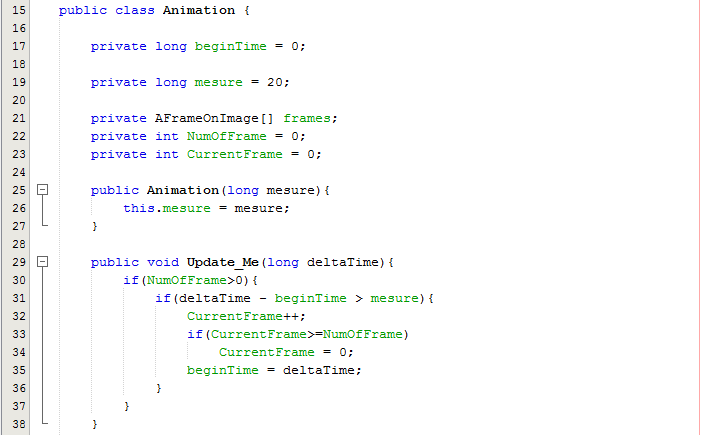
1. **Thực hiện**
2. **Các lớp tự tạo**

Tiến hành tạo ra 7 class để phục vụ cho quá trình làm game.

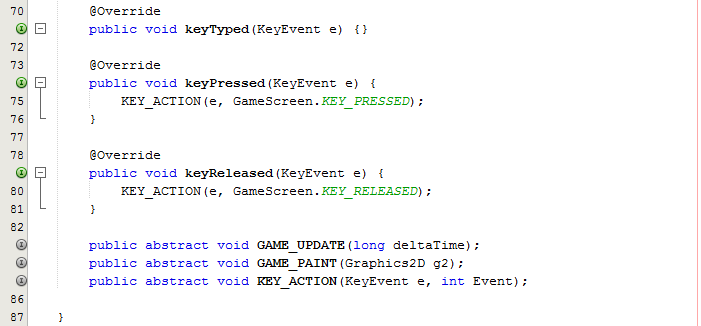
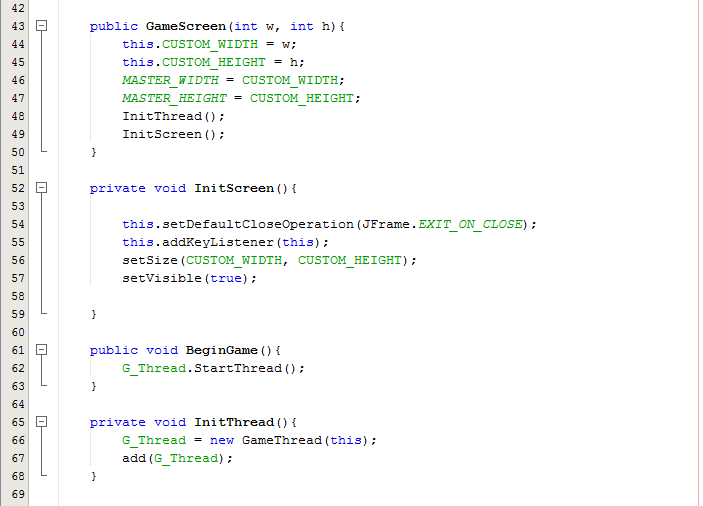
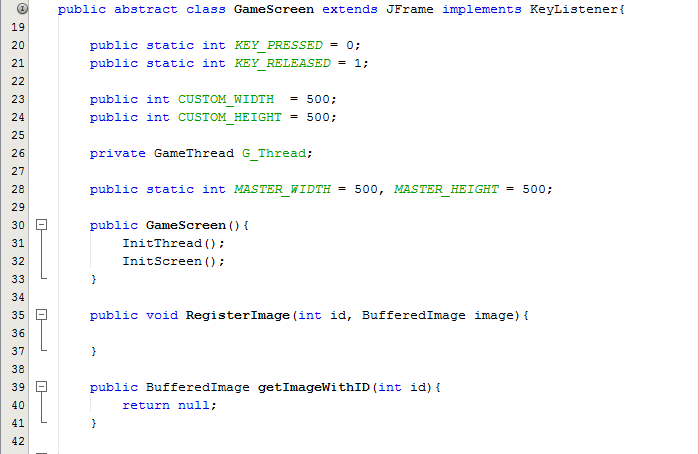
* **Class AFrameOnImage.java** : xử lý từng Frame trong một bức ảnh
* Code :



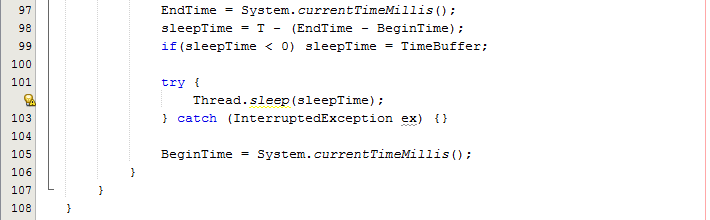
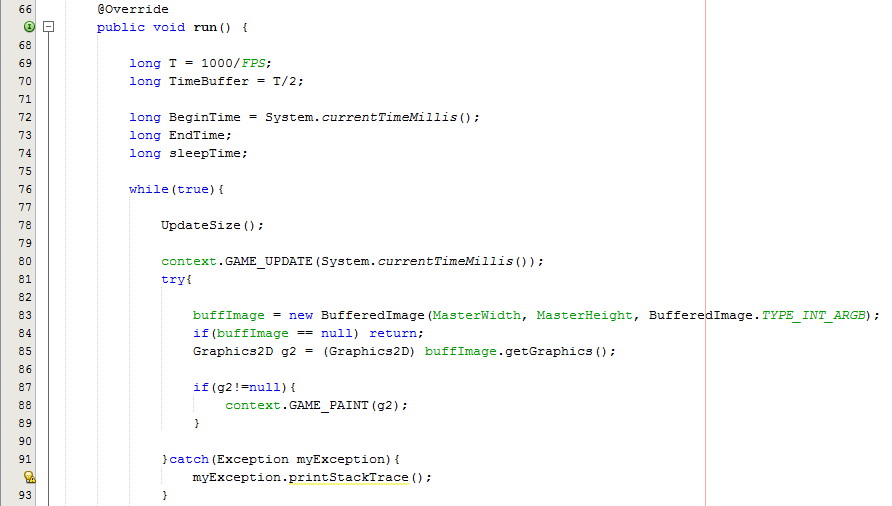
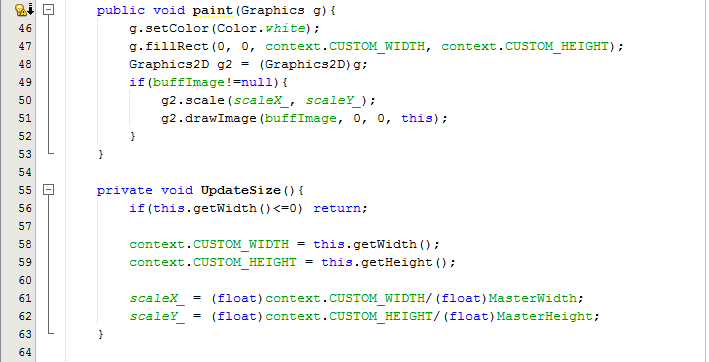
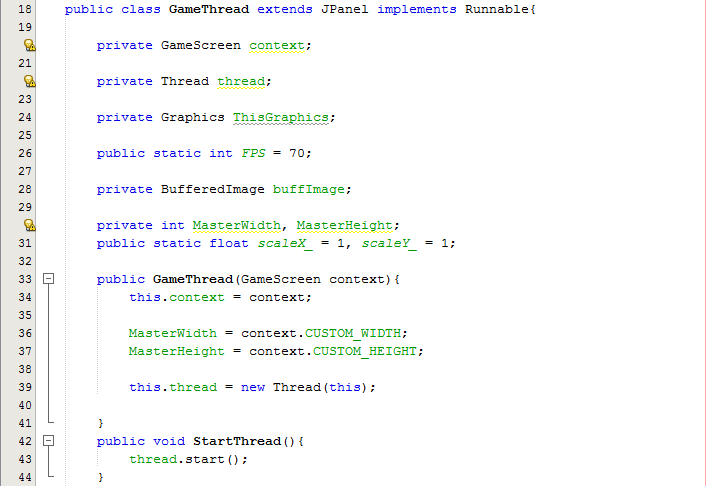
* **Class Animation.java** : xử lý,thao tác chuỗi các Frame
* Code :



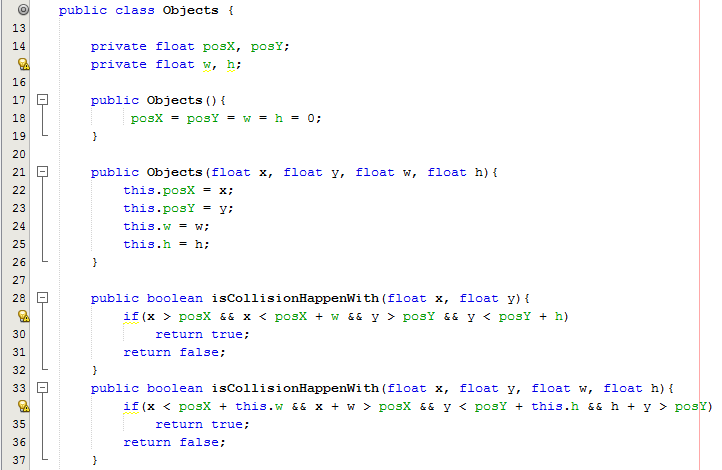
* **Class GameScreen.java** : kế thừa từ Jframe nên có thể bật lên như một cửa sổ
* Code :



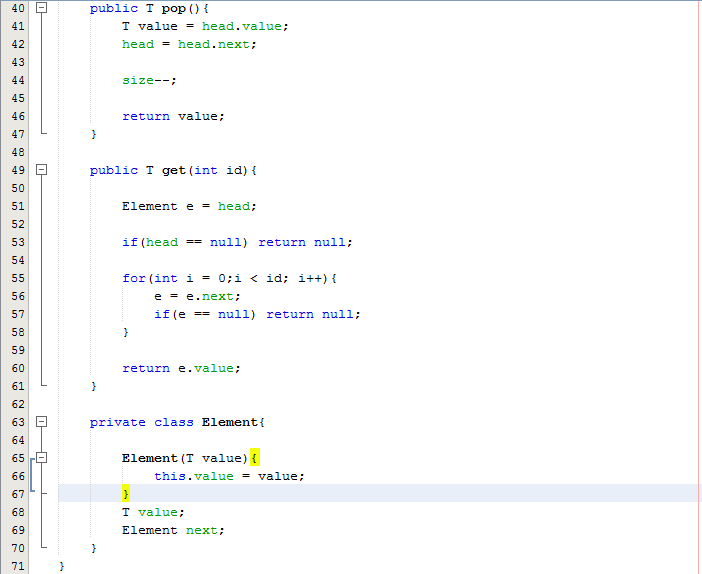
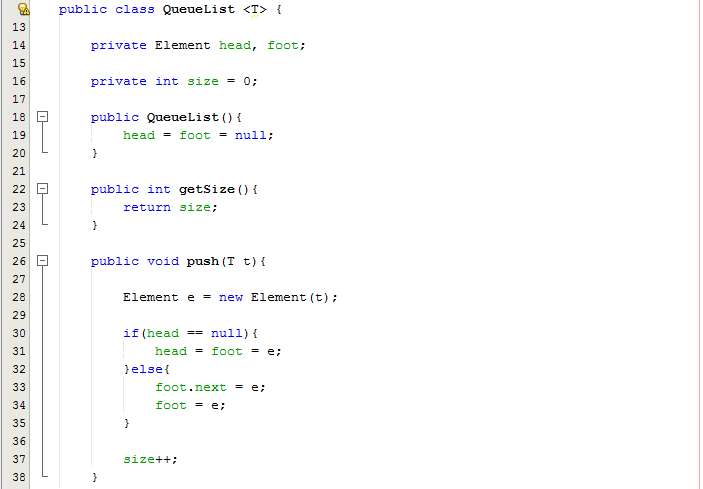
* **Class GameThread.java** : kế thừa từ Jpanel
* Code :



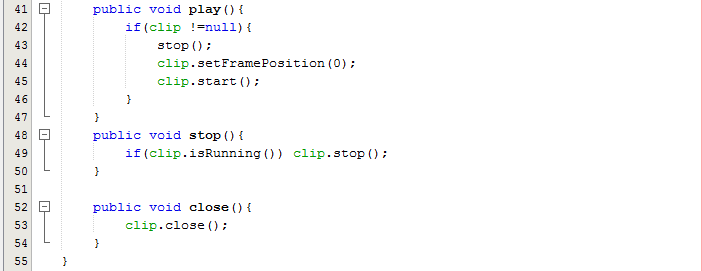
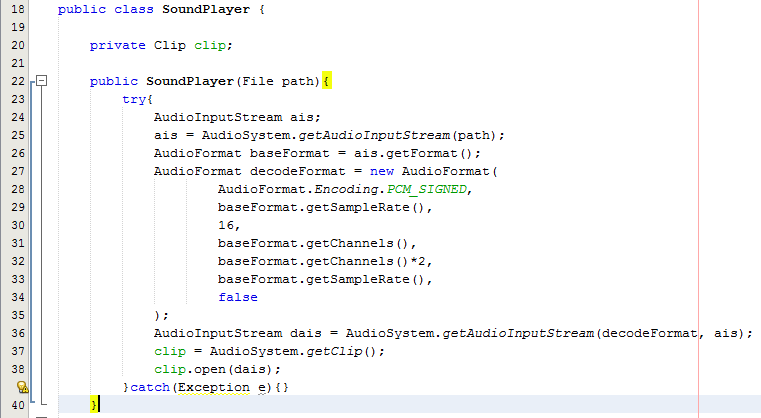
* **Class Objects.java** : chứa tham số về độ rộng và dài, tọa độ của đối tượng.
* Code :



* **Class QueueList.java** : hàng đợi có thể chứa bất kỳ đối tượng nào
* Code :

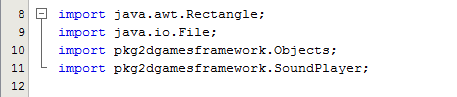


* **Class SoundPlayer.java** : xử lý âm thanh
* Code :



1. **Các đối tượng**

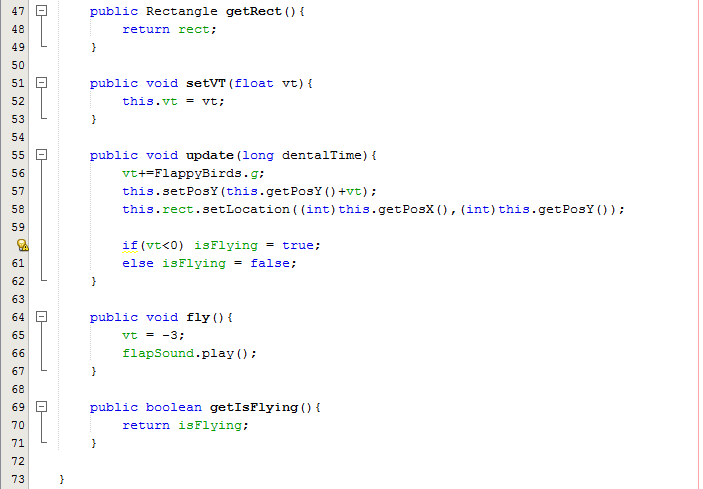
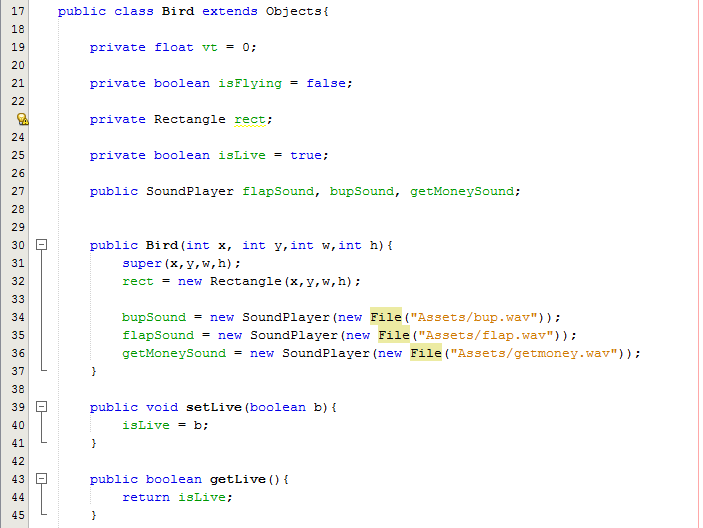
* **Bird.java** : đối tượng con chim
* Sử dụng thư viện:



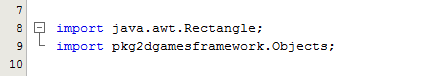
* Kế thừa từ lớp Objects
* Thân chương trình:
* Constructor : Bird(int x, int y,int w,int h)
* Hàm cập nhật : update()
* Hàm trả về sự va chạm : getRect()
* Phương thức cập nhật cho thuộc tính sự sống của con chim: setLive(boolean b) và getLive()
* Hàm đặt lại tốc độ cho con chim : setVT(float vt)
* Hàm tạo hiệu ứng bay cho con chim: fly(), getIsFlying()
* Thêm âm thanh vào file Assets : "Assets/bup.wav" , "Assets/flap.wav" , "Assets/getmoney.wav"



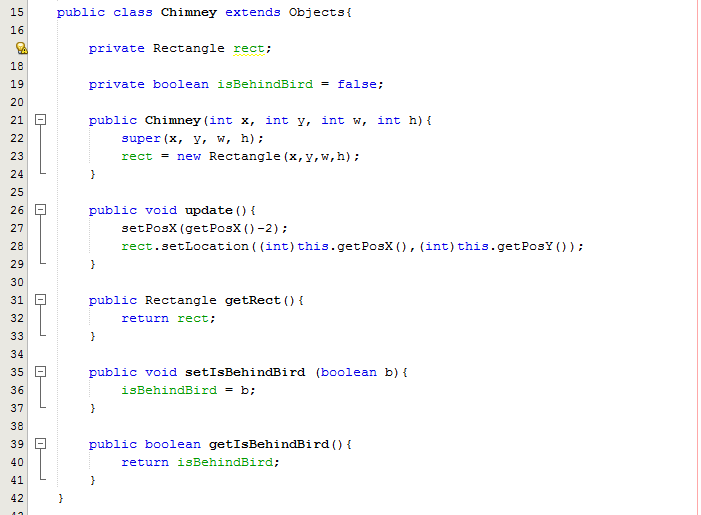
* Code :



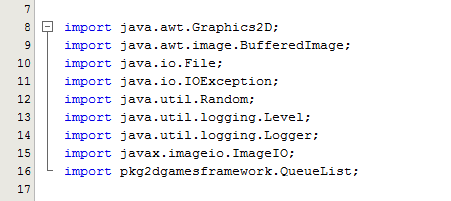
* **Chimney.java** : đối tượng một ống khói
* Sử dụng thư viện :



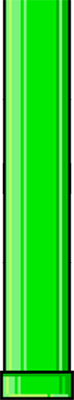
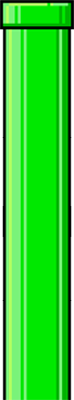
* Kế thừa từ lớp Objects
* Thân chương trình:
* Constructor : Chimney(int x, int y, int w, int h)
* Hàm trả về sự va chạm : getRect()
* Hàm xác định con chim đằng sau ống khói : setIsBehindBird(boolean b) và getIsBehindBird()
* Code :



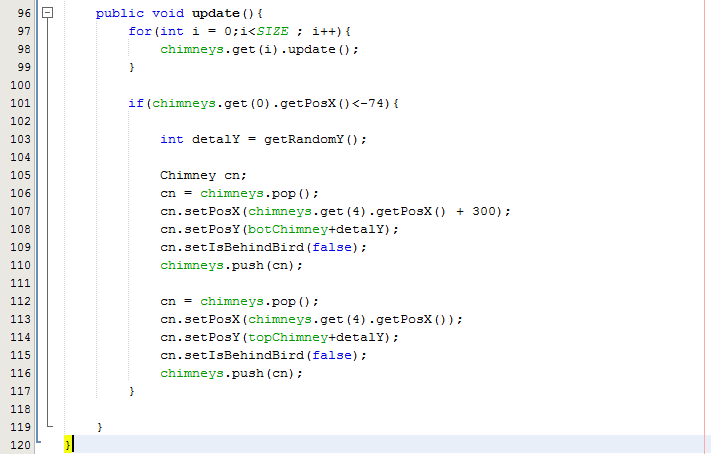
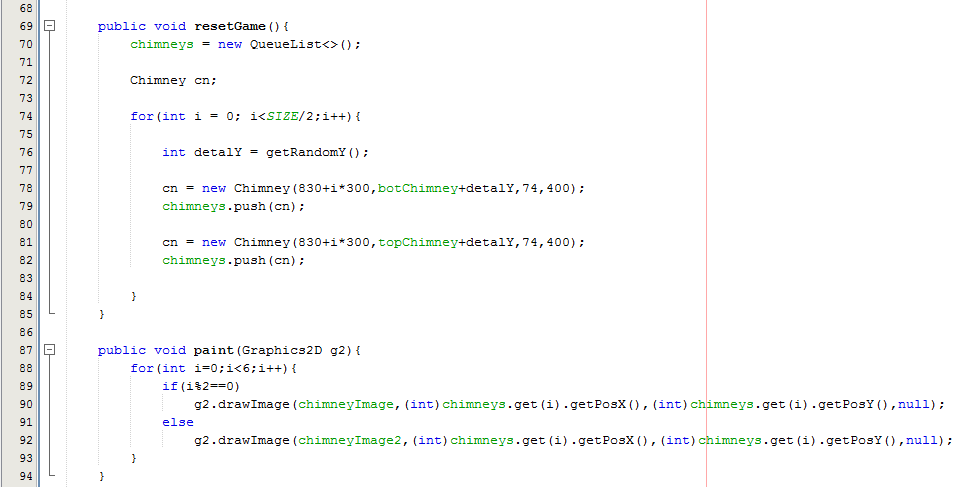
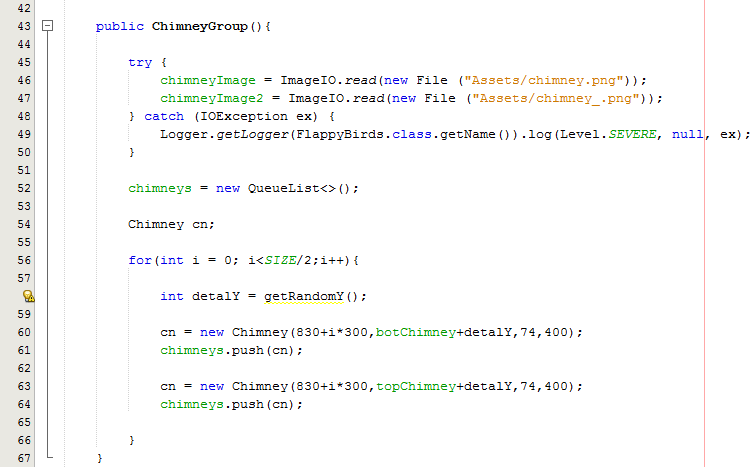
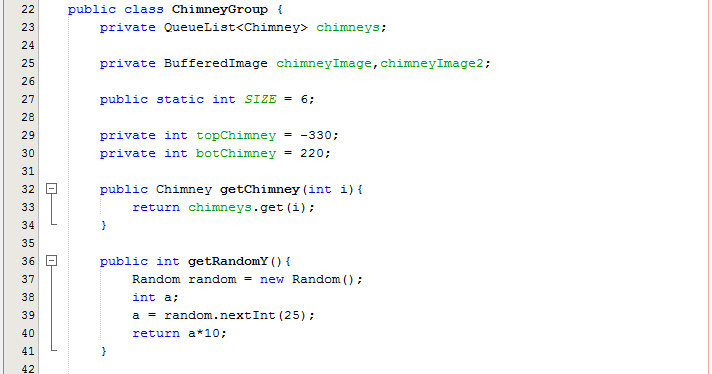
* **ChimneyGroup.java** : đối tượng 6 ống khói
* Sử dụng thư viện :



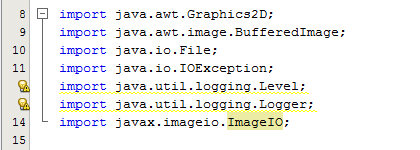
* Thân chương trình :
* Constructor : getChimney(int i)
* Tạo chuyển động lên xuống ngẫu nhiên cho ống khói : getRandomY()
* Tạo chuyển động cho ống khói : update()
* Hàm khởi tạo lại các ống khói : resetGame()
* Hàm vẽ các ống khói : paint(Graphics2D g2)
* Thêm ảnh ống khói vào file Assets : "Assets/chimney.png" và "Assets/chimney\_.png"



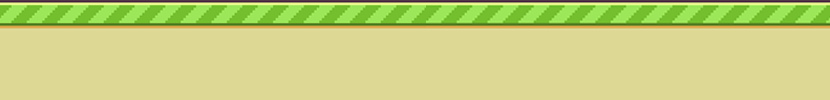
* Code :



* **Ground.java** : đối tượng nền trò chơi
* Sử dụng thư viện :



* Thân chương trình :
* Constructor : Ground()
* Hàm hiển thị ảnh nền : Paint()
* Hàm cật nhật chuyển động ống khói : Update()
* Hàm lấy chiều cao của ảnh nền : getYGround()
* Thêm ảnh nền vào file Assets : "Assets/ground.png"



* Code :



1. **Lớp chứa hàm main**

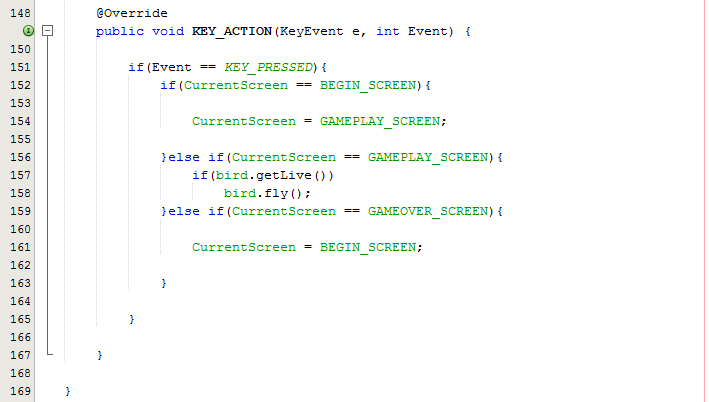
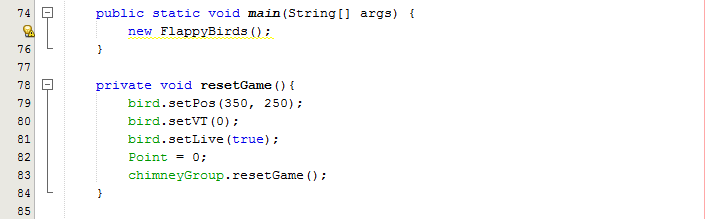
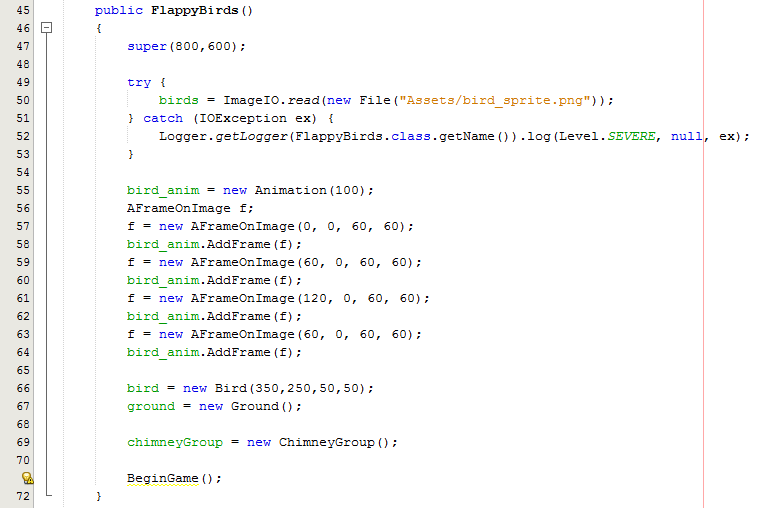
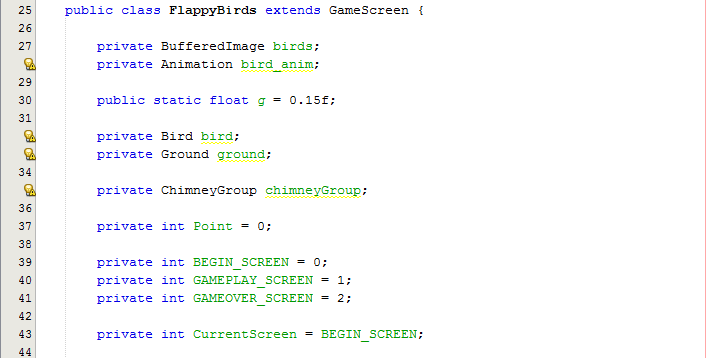
* **FlappyBirds.java** :
* Sử dụng thư viện :



* Kế thừa từ lớp GameScreen
* Thân chương trình :
* Constructor cửa sổ game
* Khởi tạo đối tượng FlappyBird trong hàm Main
* Phương thức cập nhật sự kiện: Game\_Update
* Phương thức hiển thị nội dung lên màn hình game : Game\_Paint
* Phương thức Key\_Action: xử lý chương trình khi sử dụng phím
* Phương thức resetGame() : đưa về màn hình bắt đầu và các thông số trở về ban đầu khi thua
* Đặt màu cho phông nền với mã màu: "#FF69B4"
* Thêm ảnh con chim vào file Assets: "Assets/bird\_sprite.png"

E:\Ky V\Java\Practice\FlappyBirds\Assets\bird_sprite.png

* Code :



1. **KẾT QUẢ**
2. **Hiểu hơn về ngôn ngữ Java**

* Biết thêm nhiều các hàm hỗ trợ mới
* Sử dụng thành thạo hơn ngôn ngữ Java

1. **Tạo được game FlappyBird hoàn chỉnh**

* Tạo được giao diện thân thiện dễ sử dụng
* Tạo được chuyển động của con chim, nền và ống khói
* Tính điểm cho người chơi
* Tạo âm thanh khi chơi

1. **HẠN CHẾ THIẾU SÓT VÀ HƯỚNG PHÁT TRIỂN**
2. **Hạn chế**

* Chưa đưa lên online
* Chưa tạo được bảng xếp hạng điểm
* Chưa có phần hướng dẫn và trợ giúp
* Chưa có phần tạm dừng,tiếp tục và chơi lại game

1. **Hướng phát triển**

* Thiết kế đồ họa game đẹp hơn
* Làm chuyển động của game mượt và nhanh hơn
* Đưa lên lên online hoặc trong mạng LAN

1. **TÀI LIỆU THAM KHẢO**

https://www.youtube.com/channel/UCsLcawci5OJgxB94DZU6qRQ/featured