

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**

---



**BÁO CÁO BÀI TẬP NHÓM**  
**HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**  
**Đề tài: Game Flappy Bat**

**Nhóm học phần: INT1332 nhóm 07**

**Nhóm bài tập: 03**

**Giảng viên hướng dẫn: Nguyễn Mạnh Sơn**

**Thành viên nhóm:**

<b>Nguyễn Đức Linh (Leader)</b>	<b>B20DCAT109</b>
<b>Vũ Ngọc Khánh</b>	<b>B20DCAT105</b>
<b>Nguyễn Đăng Hạnh</b>	<b>B20DCAT053</b>
<b>Lê Xuân Lộc</b>	<b>B20DCAT113</b>
<b>Vương Trung Kiên</b>	<b>B20DCAT101</b>

# MỤC LỤC

LỜI MỞ ĐẦU .....	3
I/ GIỚI THIỆU BÀI TOÁN .....	4
II/ KHẢO SÁT .....	4
1. Khảo sát các tựa game tương tự. ....	4
2. Các tính năng phát triển trong trò chơi. ....	5
3. Phương pháp – ngôn ngữ - công cụ .....	5
III/ XÁC ĐỊNH CÁC MODULE CHÍNH .....	6
1. View .....	6
2. Model .....	7
3. Control.....	13
IV/ GIỚI THIỆU HÌNH ẢNH KẾT QUẢ.....	13
V/ DANH SÁCH TÀI LIỆU THAM KHẢO .....	17
<b>KẾT LUẬN</b> .....	18

## LỜI MỞ ĐẦU

Java là ngôn ngữ lập trình do công ty Sun Microsystems phát hành lần đầu vào năm 1995. Với khởi đầu khiêm tốn, Java đã vươn lên trở thành một trong những ngôn ngữ lập trình quan trọng trong thời đại công nghệ số với số lượng lớn người dùng. Tại môn học lập trình hướng đối tượng với ngôn ngữ Java, chúng em đã được tiếp cận với những kiến thức bổ ích về hướng đối tượng và các kiến thức liên quan. Để áp dụng kiến thức đã học vào trong thực tế, nhóm 3 quyết định triển khai đề tài trò chơi Flappy Bird. Với nội dung vui nhộn và hình thức chơi đơn giản, Flappy Bird là một tựa game đã làm mưa làm gió trong thời gian vừa qua và trở thành một trong những tựa game đạt được rất nhiều kỷ lục.

Áp dụng lối chơi tương tự Flappy Bird với một số cải tiến để trò chơi hấp dẫn, kịch tính và thu hút hơn, nhóm 3 đã cho ra đời Flappy Bat. Mong rằng Flappy Bat được đón nhận từ thầy giáo và các bạn.

## I/ GIỚI THIỆU BÀI TOÁN

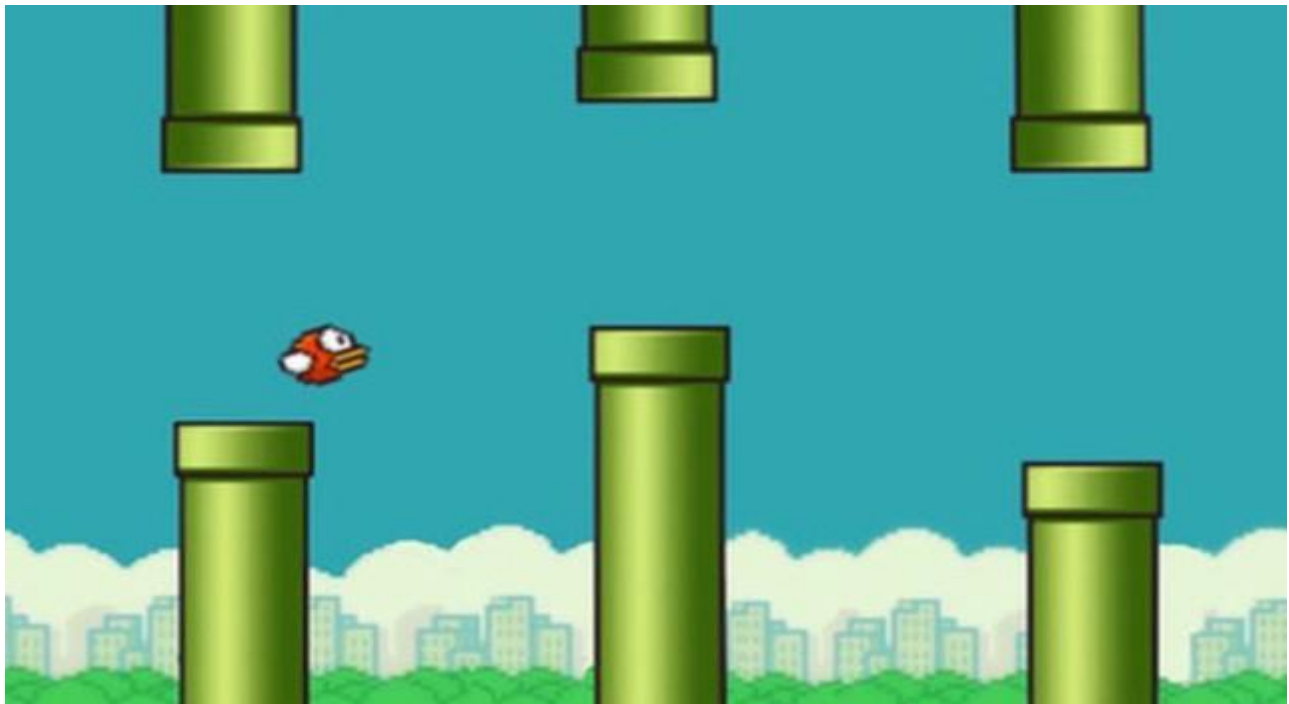
- Dựa trên nguyên mẫu của một tựa game đình đám một thời Flappy Bird, Flappy Bat với những cải tiến mới giúp người chơi có được những trải nghiệm thú vị và giải trí sau những giờ làm việc, học tập căng thẳng.

## II/ KHẢO SÁT

### 1. Khảo sát các tựa game tương tự.

Các thể loại game Flappy Bird hiện nay trên thị trường có rất nhiều phiên bản khác nhau. Tuy nhiên, hầu hết chúng đều giống nhau về mục tiêu của trò chơi đó là điều khiển một chú chim bay qua những cái ống. Nếu chú chim chạm vào chướng ngại vật thì trò chơi sẽ kết thúc. Mỗi khi chú chim vượt qua một cặp ống thì người chơi nhận được một điểm. Nó sử dụng đồ họa tương tự như Super Mario với một hệ thống điều khiển vô cùng đơn giản.

Yếu tố chính tạo nên thành công của trò chơi chính là điều luật khắc khe. Điều luật của Flappy bird không cho người chơi có nhiều mạng, thay vào đó chỉ một va chạm vào chướng ngại vật thì màn hình “Game Over” hiện ra. Rất khó để có thể đạt được điểm cao ở trò chơi này.



*Hình 1: minh họa game Flappy Bird*

## 2. Các tính năng phát triển trong trò chơi.

- Giao diện bắt đầu game: Giao diện bắt đầu game hiển thị các nút bấm Start, Help, Exit hỗ trợ người dùng điều phối game.
    - + Start: Bắt đầu một game mới.
    - + Help: Hiển thị thông tin về cách chơi.
    - + Exit: Thoát game.
  - Tạo các cột chướng ngại vật với hình dạng và khoảng trống khác nhau.
  - Random khoảng cách giữa các cột chướng ngại vật.
  - Hiệu ứng dơi chuyển động và vỗ cánh (Khi bấm Space cho dơi nhảy -> Dơi vỗ cánh)
  - Hiệu ứng dịch chuyển background (map) tạo độ sâu cho game.
  - Cộng điểm mỗi khi vượt qua một cột chướng ngại vật.
  - Độ khó của game tăng dần tỉ lệ thuận với tốc độ chuyển động và khoảng cách giữa các cột.
  - Hiệu ứng phóng to khi bấm nút Up và tăng thêm điểm sau mỗi lần vượt qua chướng ngại vật.
  - Chức năng tạm dừng (Pause) và tiếp tục (Resume) chơi game
  - Hiệu ứng âm thanh cho game
    - + Âm thanh nền khi chơi game
    - + Âm thanh khi bấm các nút chọn
    - + Âm thanh mỗi lần vượt qua chướng ngại vật
    - + Âm thanh khi dơi chết
  - End game hiện tổng điểm và gợi ý chơi lại hoặc thoát game
- ## 3. Phương pháp – ngôn ngữ - công cụ
- Sử dụng ngôn ngữ Java để viết chương trình.
  - Mô hình MVC.
  - Lớp Java Swing
  - Lớp Graphics
    - + Lớp JFrame
    - + Lớp JPanel
    - + Lớp JLabel

- Nhập xuất File
- Class Timer
- Lớp ImageIO
- Random
- Các lớp xử lý âm thanh

### III/ XÁC ĐỊNH CÁC MODULE CHÍNH

#### 1. View

- Class View để tạo cửa sổ bắt đầu game (Bao gồm MenuHelp, MenuMain, MenuManager)
- + Thuộc tính
  - public static final int W\_FRAME = 1500; // chiều ngang khung cửa sổ mở đầu game
  - public static final int H\_FRAME = 800; // chiều dọc khung cửa sổ mở đầu game
  - private MenuMain menuMain;
  - private EndGame endgame;
  - private MenuManager menuManager;
  - private FlappyBird flappyBird;
- + Phương thức
  - Phương thức khởi tạo: Public View()
- Class FlappyBird để tạo cửa sổ game và vẽ nền cho game (Sử dụng class Graphics để vẽ Background, vẽ dơi và tô màu cột)
- + Thuộc tính
  - public SoundPlayer fapSound, fallSound, getPointSound, themesong, zoomsong, clicksong; // các loại âm thanh sử dụng trong game
  - public static int WIDTH = 1500, HEIGHT = 800;  
// chiều ngang, dọc khung cửa sổ game
  - public static Rectangle bird, map1, map2, map3, map4; // dơi và các map trong game
  - public static ArrayList<Rectangle> columns; // mảng lưu trữ các cột chướng ngại vật
  - public Random random;

- `public static int score, tick, yMotion; // điểm số`
- `public static boolean start, end; // biến đánh dấu bắt đầu và kết thúc game`
- `public Background background;`
- `public static Modeller model;`
- `public static Controller control;`
- `public static int vk_space; // đánh dấu dơi vỗ cánh`
- `public MenuManager menuManager;`
- `public View view;`
- `public static Button pause, pause1; // nút tạm dừng game`
- `public static int pause_int = 4;`

+ Phương thức

- Phương thức khởi tạo `public FlappyBird()`
- `static void repaint(Graphics g) // vẽ lại giao diện game sau khoảng thời gian timer`
- `private static void paintColumn(Graphics g, Rectangle column) // tô màu cột`
- `private static void paintGround(Graphics g) // tô màu đất`
- `private static void paintMap(Graphics g) // tô màu nền`

- Class EndGame hiển thị cửa sổ kết thúc game (Bao gồm điểm và gợi ý chơi lại)

+ Thuộc tính

- `private JDialog dialog; // khởi tạo hộp thoại endgame`
- `private JLabel topic, guide, mark; // tiêu đề hộp thoại, điều hướng, điểm số`
- `private JButton yes, no; // nút yes, no`
- `private FlappyBird flappyBird;`
- `private Controller controller;`
- `private MenuManager menuManager;`

+ Phương thức

- `public EndGame(int a, FlappyBird flappyBird, Controller controller) //phương thức khởi tạo`
- `public void showDialog(int a) // hiển thị hộp thoại`

## 2. Model

- Class Modeller

+ Thuộc tính

- private FlappyBird flappyBird;
- public int speed; // tốc độ chuyển động của cột

+ Phương thức

- public Modeller(FlappyBird flappyBird) // phương thức khởi tạo
- public void jump() // Xử lý chuyển động nhảy lên của dơi: sau mỗi lần kích chuột dơi nhảy lên 10 đơn vị theo trục Oy và dơi sẽ vỗ cánh (Hình 2). Thiết lập về vị trí ban đầu của dơi sau mỗi lần bắt đầu lại (Hình 3)



Hình 2: Hoạt ảnh dơi vỗ cánh

```
public void jump() {
    flappyBird.fapSound.play();
    if (flappyBird.end) { //ket thuc va bat dau lai

        flappyBird.bird = new Rectangle(flappyBird.WIDTH / 3, flappyBird.HEIGHT / 2 - 150, width: 50, height: 50); //dat vi tri bird ban dau

        flappyBird.columns.clear();
        flappyBird.yMotion = 0;
        flappyBird.score = 0;
        flappyBird.vk_space = 0;

        flappyBird.model.addColumn( start: true);
        flappyBird.model.addColumn( start: true);
        flappyBird.model.addColumn( start: true);
        flappyBird.model.addColumn( start: true);

        flappyBird.end = false;
    }

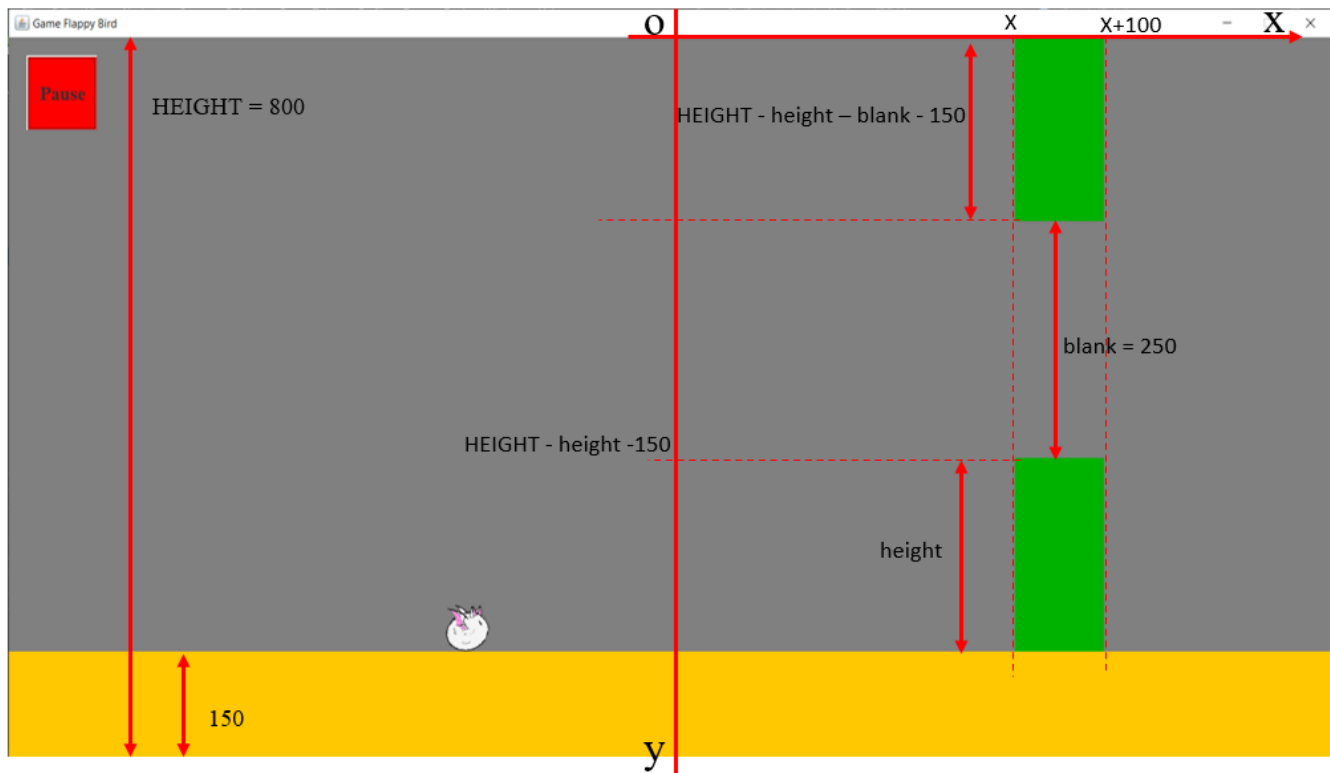
    if (!flappyBird.start) { //bat dau choi
        flappyBird.start = true;
        flappyBird.themesong.play();
    } else if (!flappyBird.end) { // chua ket thuc
        if (flappyBird.yMotion > 0) {
            flappyBird.yMotion = 0;
        }

        flappyBird.yMotion -= 10;
    }
}
```

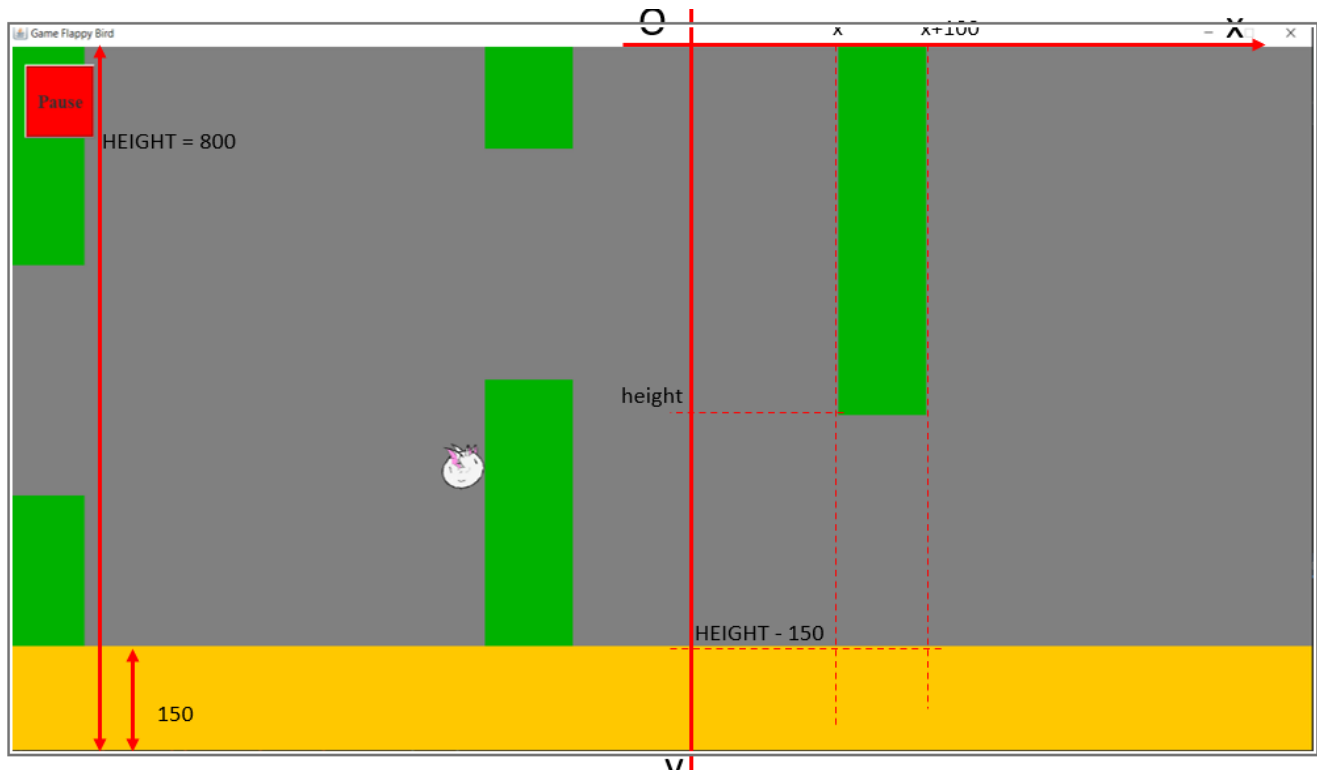
Hình 3: Minh họa xử lý chuyển động nhảy lên của dơi



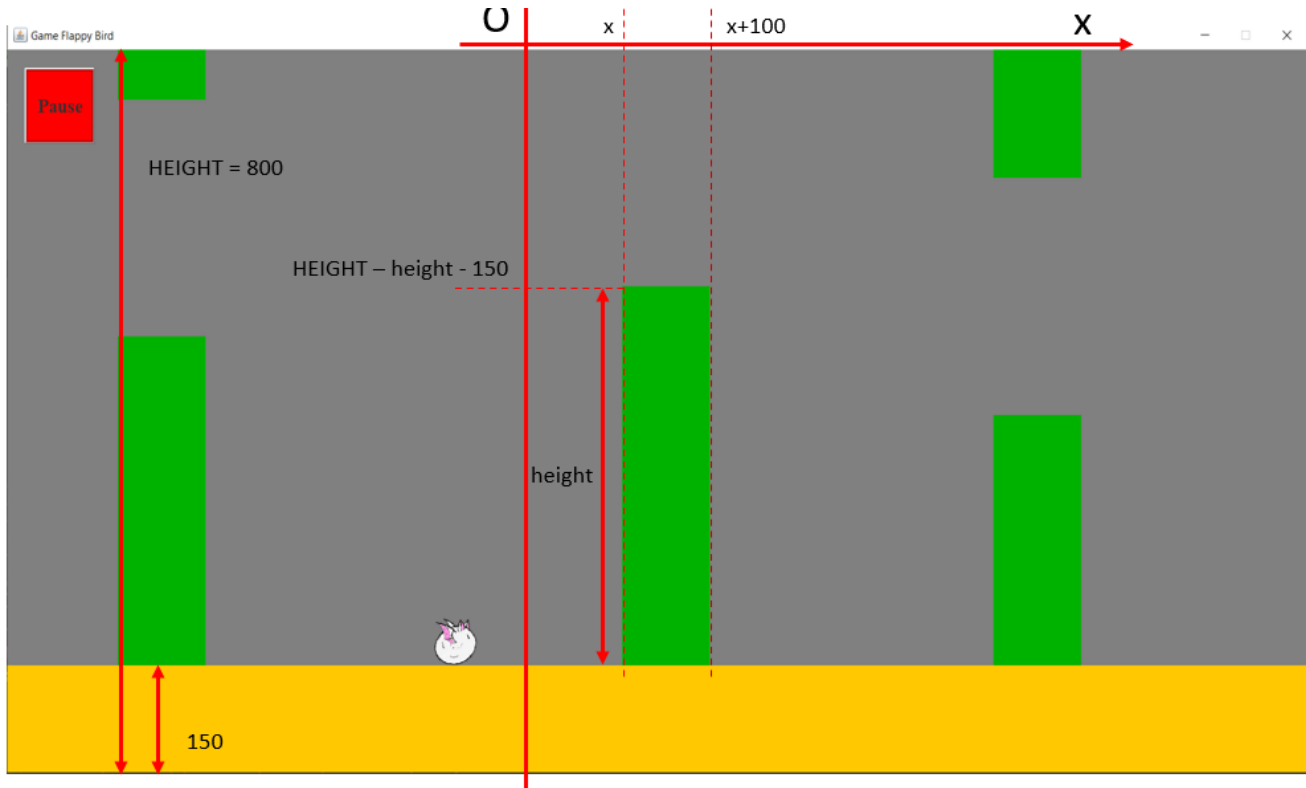
- `public void addColumn(boolean start) // Sinh 5 loại cột chướng ngại vật bằng cách random khoảng cách giữa hai cột (Hình 4 -> 8)`



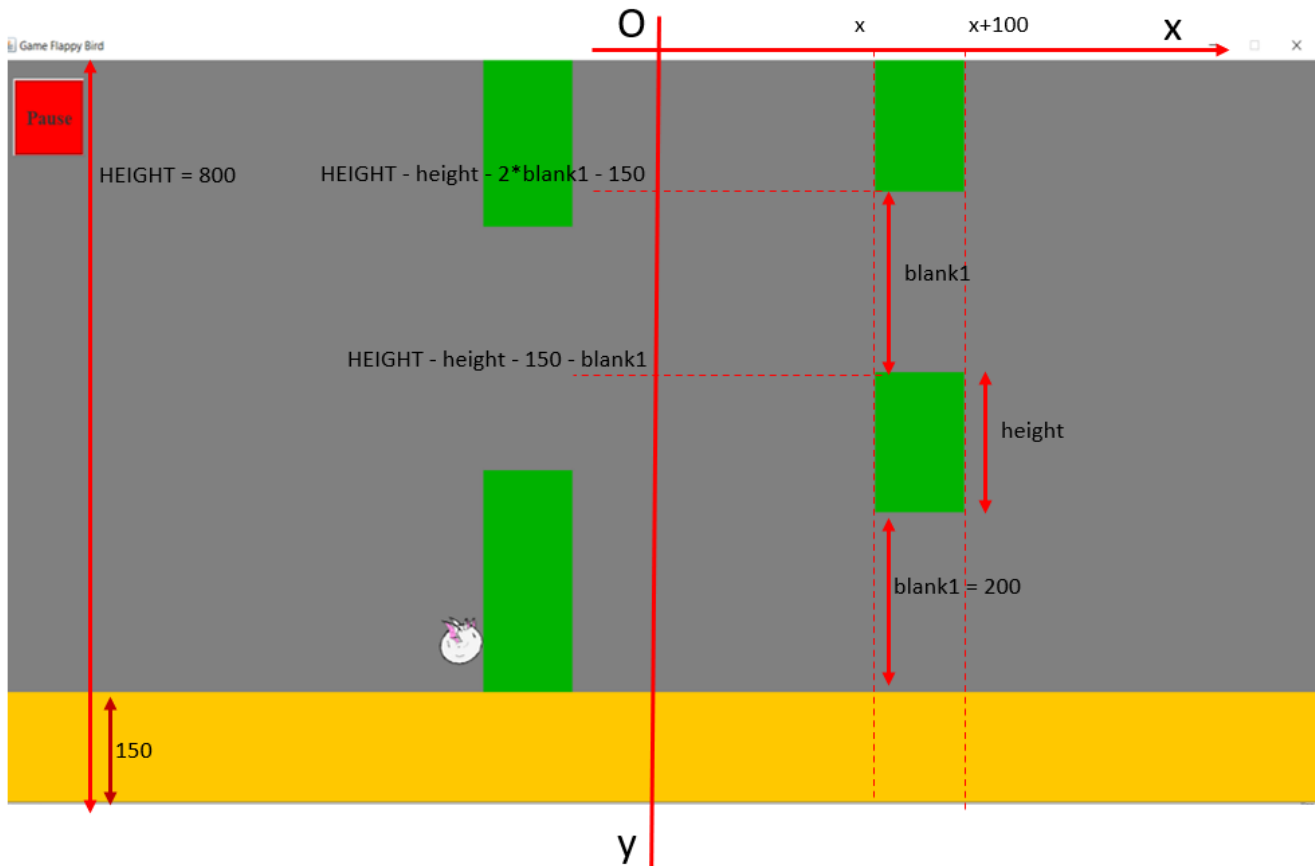
Hình 4



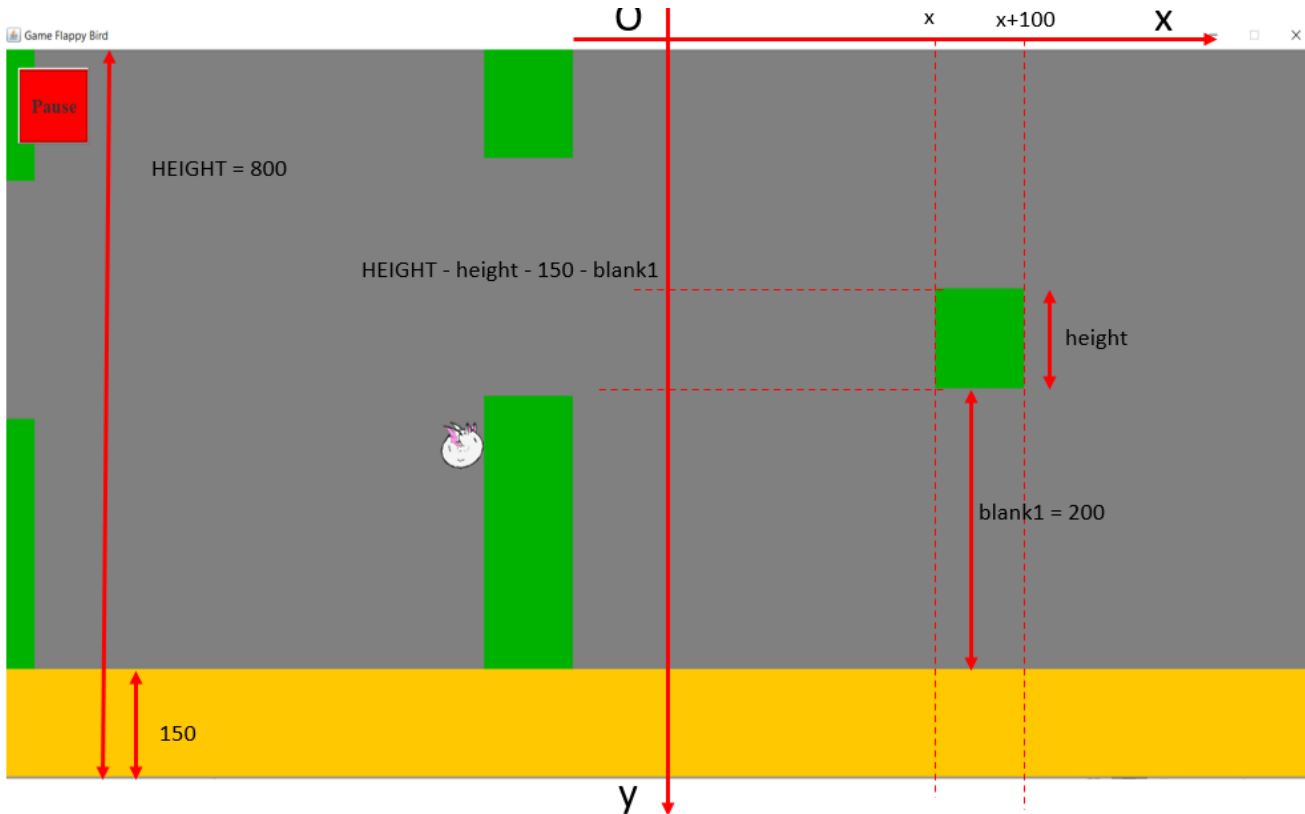
Hình 5



Hình 6



Hình 7



Hình 8

- Trong đó biến *height* được sinh ngẫu nhiên bằng cách random
- `public void birdFly() // Tăng tốc độ di chuyển của cột sau mỗi 3 điểm (Hình 9)`

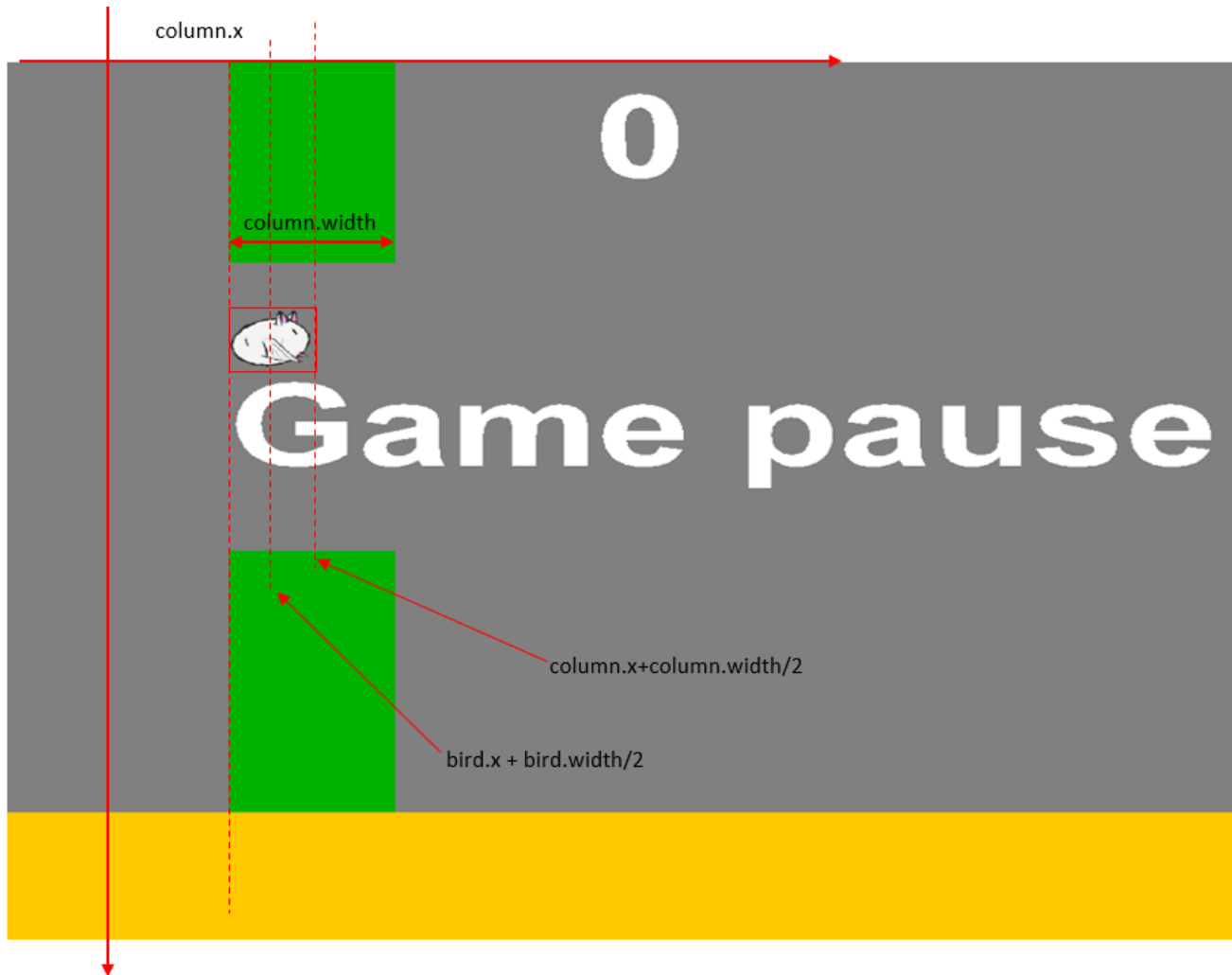
```
speed = 10; //tốc độ
speed += (flappyBird.score / 3); //tăng tốc độ sau add 3 score
flappyBird.tick++;
```

Hình 9: hình minh họa tăng tốc độ di chuyển cột

- Mỗi lần dơi đi qua cột điểm tăng thêm 1 điểm (Hình 10 + 10.1)

```
for (Rectangle column : flappyBird.columns) {
    if (column.y == 0 && flappyBird.bird.x + flappyBird.bird.width / 2 >= column.x + column.width / 2 &&
        flappyBird.bird.x + flappyBird.bird.width / 2 < column.x + column.width / 2 + speed) {
        flappyBird.score++; //tăng điểm
        flappyBird.getPointSound.play();
    }
}
```

Hình 10: hình minh họa tăng điểm sau mỗi lần vượt qua cột



Hình 11

- Sau mỗi lần  $\text{column.x} + \text{column.width} < 0$  thì xóa cột hiện tại và add thêm cột mới vào ArrayList. (Hình 11)

```
//add and remove column
for (int i = 0; i < flappyBird.columns.size(); i++) {
    Rectangle column = flappyBird.columns.get(index: i);
    if (column.x + column.width < 0) { //di het man hinh
        flappyBird.columns.remove(0: column); //remove column
        if (column.y == 0) { //cot tren
            flappyBird.model.addColumn(start: false); //add column
        }
    }
}
```

*Hình 12: hình minh họa các xóa cột và thêm cột mới*

- Kiểm tra, nếu xảy ra va chạm thì game kết thúc và hiện bảng điểm.

### 3. Control

#### - Class Controller

##### + Thuộc tính

- private FlappyBird flappyBird;
- public Timer timer;
- private EndGame a;

##### + Phương thức

- public Controller(FlappyBird flappyBird) // khởi tạo (Sử dụng class Timer để cập nhật giao diện game sau 20ms)
- public void actionPerformed(ActionEvent e)
- public void mouseClicked(MouseEvent e) // bắt sự kiện kích chuột
- public void keyReleased(KeyEvent e) // bắt sự kiện nhấn phím

##### + Ngoài ra còn có công dụng các phím

- Click chuột / Phím Space: Dơi nhảy lên và xuất hiện hiệu ứng vỗ cánh
- Phím Up: Phóng to dơi để giảm tốc độ chuyển động của các cột + Dơi nhảy lên
- Nhấn phím Down: Tạm dừng game

Ngoài ra: Tích hợp âm thanh (vào game, dơi ăn điểm, kết thúc game) + hình ảnh

## **IV/ GIỚI THIỆU HÌNH ẢNH KẾT QUẢ**

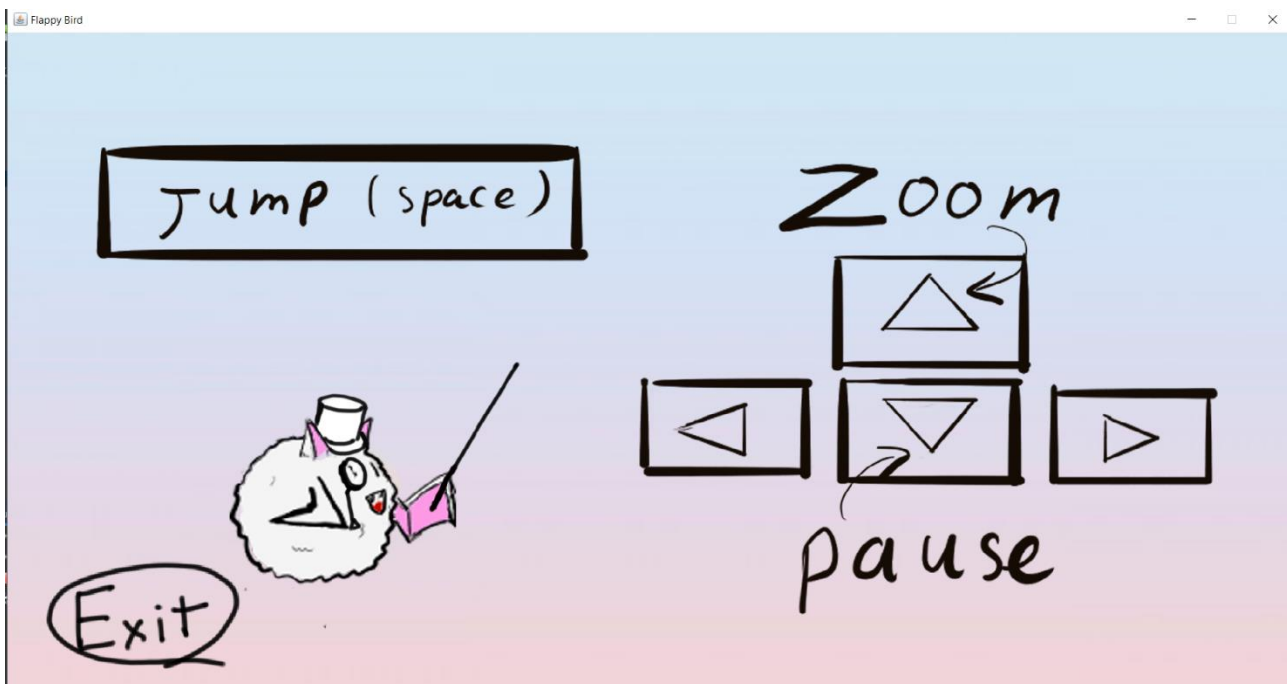
Một số hình ảnh kết quả:

### 1. Giao diện bắt đầu game



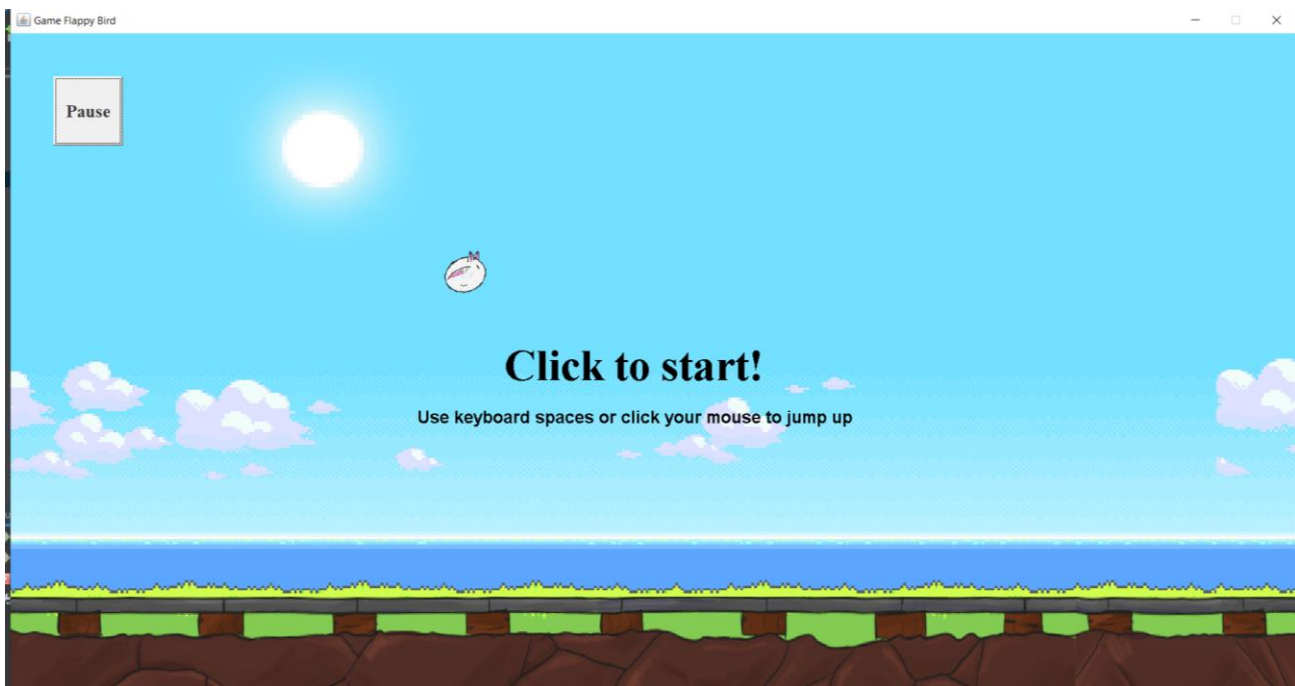
Hình 13: minh họa giao diện bắt đầu game

## 2. Màn hình hướng dẫn chơi game

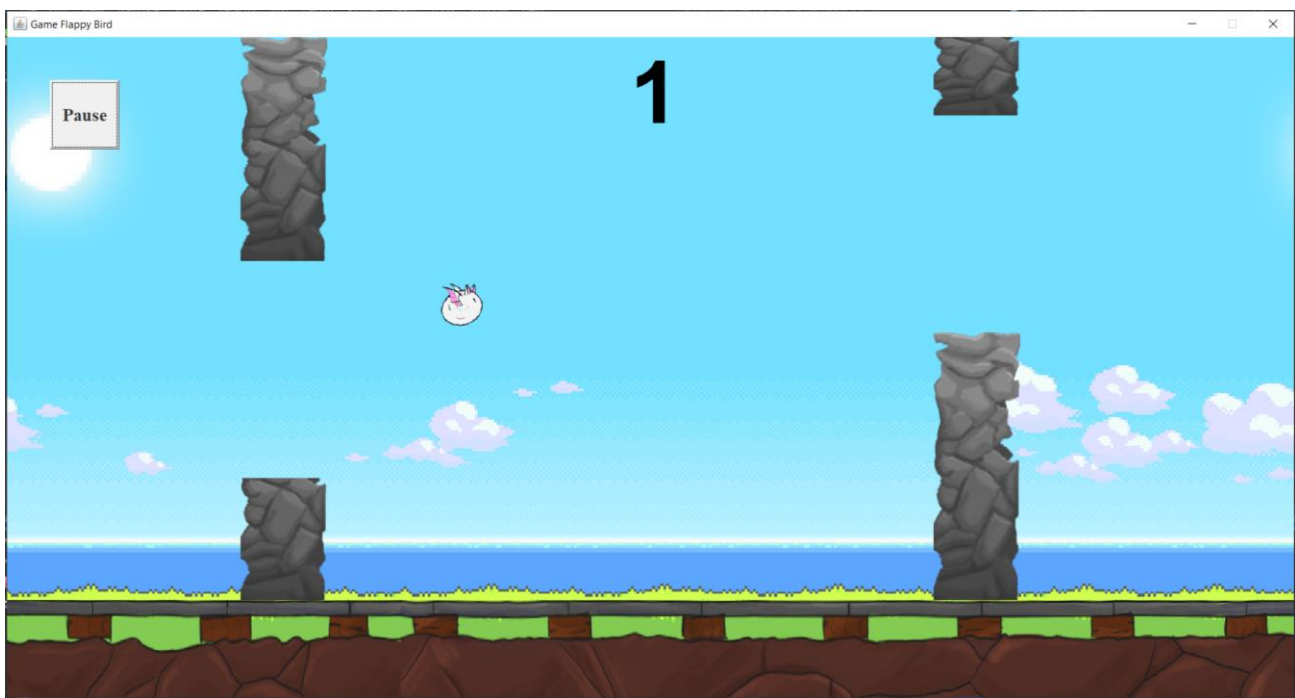


Hình 14: minh họa màn hình hướng dẫn chơi game

## 3. Giao diện game chính

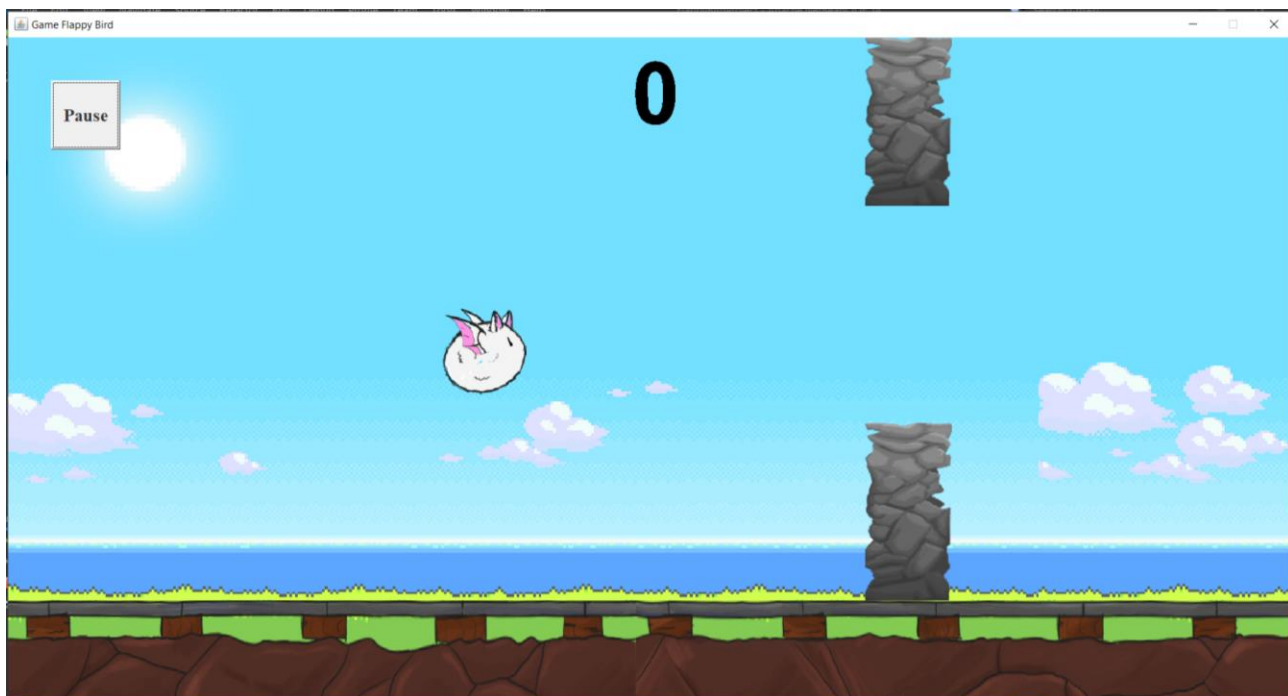


Hình 15.



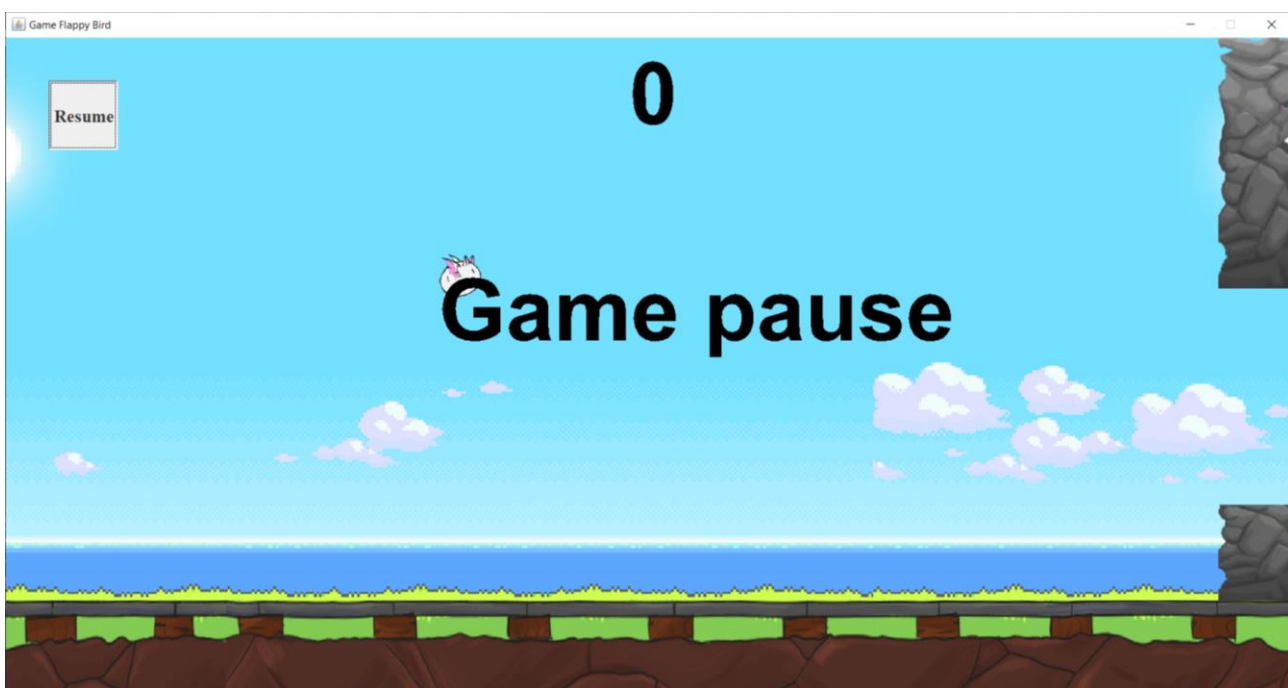
Hình 16.

Hình ảnh minh họa dơi phóng to (hình 17):



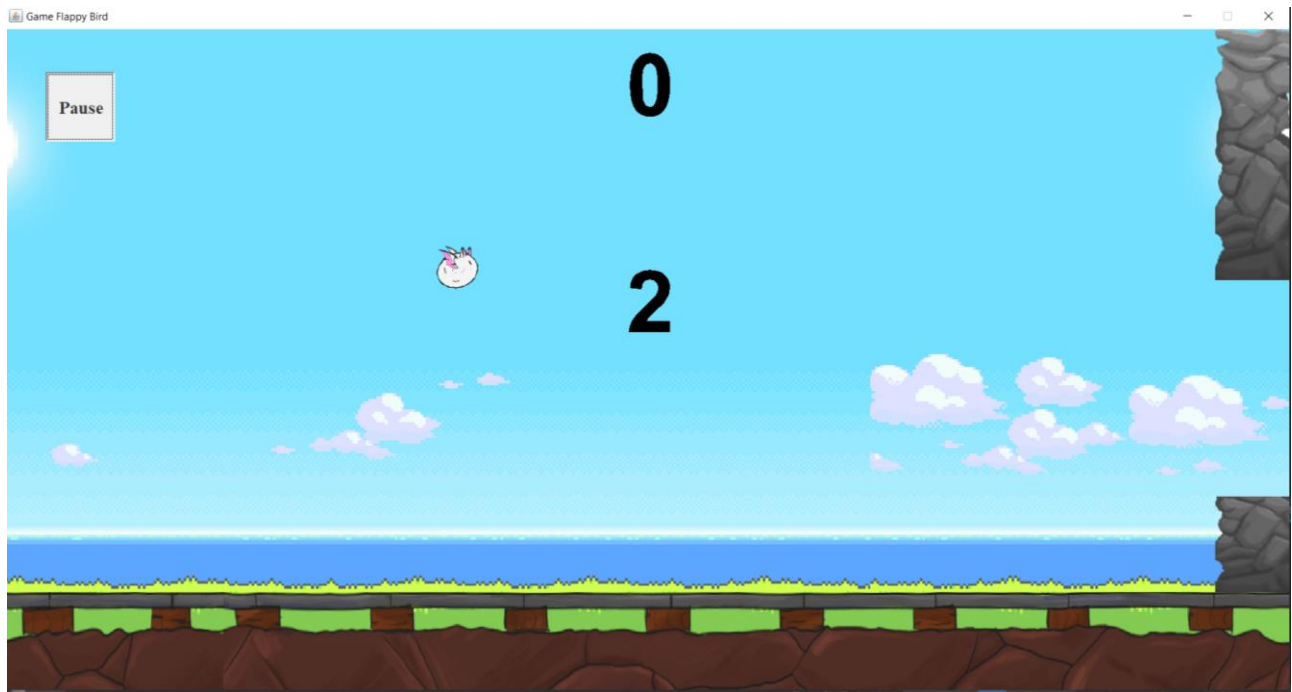
Hình 17.

Minh họa chức năng tạm dừng (hình 18, hình 19).



Hình 18.





Hình 19.

## V/ DANH SÁCH TÀI LIỆU THAM KHẢO

- <https://codereview.stackexchange.com/questions/196375/java-flappy-birds-clone>
- <https://www.instructables.com/Java-Game-Programming-Tutorial-Flappy-Bird-Redux/#:~:text=1%20Concept%20Building%20%2F%20Graphic%20Design.%20This%20is,Obstacle%20Classes.%20...%206%20The%20Graphics%20Class.%20>
- <https://www.youtube.com/watch?v=I1qTZaUcFX0>
- <https://flappybird.io/>

## KẾT LUẬN

Báo cáo bao gồm giới thiệu bài toán, khảo sát, xác định module chính, phương pháp – ngôn ngữ - công cụ, giới thiệu hình ảnh kết quả và danh sách tài liệu tham khảo.

Do mới làm quen với ngôn ngữ Java và các kỹ năng xử lý kỹ thuật trong lập trình hướng đối tượng nên trong đề tài còn có một số hạn chế. Nhóm 3 rất mong nhận được nhận xét của thầy về đề tài để nhóm có thể khắc phục những hạn chế này trong các đề tài tương lai.