

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

KHOA CÔNG NGHỆ THÔNG TIN

-----o0o-----



Bài tập lớn môn học

Công nghệ Java

Giảng viên hướng dẫn	: TS. Vũ Huấn.
Sinh viên thực hiện	: Hoàng Hữu Chiến. Hoàng Văn Đại.
Nhóm	: 23.
Lớp	:CNTT5 –K62.

HÀ NỘI - 2023

LỜI NÓI ĐẦU

Ngày nay, cùng với sự phát triển của đất nước, hòa vào đó Công nghệ thông tin là một ngành đã có những bước phát triển mạnh mẽ không ngừng và nó đã trở thành chiếc chìa khóa dẫn đến thành công cho nhiều cá nhân, tổ chức. Với những ứng dụng của mình, công nghệ thông tin đã góp phần mang lại nhiều lợi ích mà không ai có thể phủ nhận được. Đặc biệt trong lĩnh vực quản lý kinh tế, công nghệ thông tin đã góp phần tạo ra sự thay đổi nhanh chóng cho bộ mặt xã hội.

Trong quá trình thực hiện đề tài, nhóm đã gặp không ít khó khăn và trở ngại, tuy nhiên, với sự trợ giúp của bạn bè, cùng với đó là sự hướng dẫn tận tình của TS. Vũ Huân, cuối cùng nhóm chúng em cũng đã cố gắng hoàn thành đề tài này. Chúng em xin gửi lời cảm ơn chân thành đến thầy về sự hướng dẫn và hỗ trợ trong quá trình thực hiện làm game “Snake game”. Nhờ có sự hướng dẫn tận tình, thầy đã giúp nhóm có được cái nhìn tổng quan về quá trình phát triển game, từ việc thiết kế game, cài đặt và lập trình cho đến các bước kiểm thử và hoàn thiện game. Những kiến thức và kinh nghiệm chúng em học được từ thầy giúp em nắm vững kiến thức về lập trình game.

Do kiến thức thực tế và hiểu biết còn hạn chế nên trong quá trình hoàn thành báo cáo không thể tránh khỏi sai sót, nhóm chúng em rất mong sự góp ý đóng góp của thầy và các bạn để nhóm có thể hoàn thiện bài tập lớn được tốt hơn.

Nhóm xin chân thành cảm ơn!

I. Mô tả game.

1. Giới thiệu tổng quan.

- Game con rắn (Snake Game) là một trò chơi điện tử kinh điển được phát triển từ những năm 1970 và 1980. Trong game, người chơi điều khiển một con rắn ăn thức ăn (có thể là quả táo hoặc chuối), và cứ mỗi lần ăn được thức ăn, con rắn sẽ dài thêm một đốt. Nhiệm vụ của người chơi là giúp con rắn ăn càng nhiều thức ăn càng tốt để đạt điểm số cao nhất.
- Game con rắn là một trò chơi đơn giản, dễ chơi và phổ biến, thường được cài đặt trên các thiết bị di động, trang web hay các nền tảng game. Đồ họa và cách chơi của game con rắn có thể thay đổi tùy theo từng phiên bản, nhưng luôn giữ được đặc trưng cơ bản của trò chơi: điều khiển con rắn di chuyển trên màn hình, tránh chạm vào tường hoặc đuôi của chính nó, và ăn thật nhiều thức ăn.
- Game con rắn là một trong những trò chơi đơn giản, gây nghiện và rất thú vị. Trò chơi không chỉ giúp người chơi giải trí mà còn rèn luyện kỹ năng tập trung, phản xạ nhanh và khả năng quyết định trong thời gian ngắn.

2. Mục tiêu chính.

- Điều khiển một con rắn để ăn thức ăn và trở nên dài hơn. Với mỗi món ăn con rắn ăn được, chiều dài của con rắn sẽ tăng thêm một đơn vị. Tuy nhiên, khi con rắn va chạm vào tường hoặc chính thân mình, trò chơi sẽ kết thúc và người chơi sẽ phải bắt đầu lại từ đầu. Mục tiêu của người chơi là ăn được nhiều thức ăn nhất có thể để đạt được số điểm cao nhất và vượt qua các thử thách khác nhau trong trò chơi.
- Ngoài mục tiêu chính trên, thì trò chơi Snake Game cũng có thể giúp người chơi giải trí và thư giãn.

3. Các tính năng của game.

Trò chơi Chrome Snake Game có một số tính năng đơn giản như:

- Điểm số : Mỗi khi con rắn ăn được mồi thì điểm sẽ được cộng thêm 1. Mục tiêu của người chơi là chơi sao cho con rắn ăn được nhiều điểm nhất có thể và không để bị chết.
- Level : Khi độ dài của con rắn là tăng thêm 5 thì con rắn sẽ tăng level. Ví dụ: con rắn level 1 là từ 1-4 khi ăn mồi để tăng lên 5 thì level sẽ tăng lên. Khi level tăng lên thì tốc độ của con rắn cũng tăng lên.
- Chướng ngại vật : Tăng thêm 1 level ta cũng sẽ thêm 1 chướng ngại vật ngẫu nhiên. Khi ta chạm vào chướng ngại vật chúng ta sẽ “Game Over”.
- Tạm dừng : Người chơi có thể tạm dừng trò chơi bằng cách nhấn phím Space. Khi trò chơi bị tạm dừng, con rắn sẽ dừng lại và trò chơi sẽ tạm thời dừng.
- Game over : Khi đầu con rắn chạm vào thân con rắn thì game sẽ kết thúc và hiện lên tab để chúng ta lưu tên và điểm số.

- Lưu điểm số: Khi chết màn hình sẽ hiện ra tab lưu tên bạn nhập tên bạn vào nó sẽ lưu lên bên phải tên của bạn số level của bạn và số điểm mà bạn đã đạt được.

II. Các công nghệ sử dụng.

Các công nghệ đã sử dụng để xây dựng nên game Snake Game:

1. Ngôn ngữ lập trình Java

- Java là một ngôn ngữ lập trình mạnh mẽ và đa nền tảng. Nó được sử dụng rộng rãi trong phát triển trong việc phát triển ứng dụng máy tính và trò chơi trên nhiều nền tảng khác nhau. Nếu muốn phát triển một trò chơi tương tự Chrome Offline Dinosaur, chúng ta có thể sử dụng một số công nghệ như JavaFX, Java Swing, ...

2. Công cụ phát triển

- Ta có thể sử dụng công cụ Eclipse để xây dựng nên game Chrome Offline Dinosaur - Eclipse là một môi trường phát triển tích hợp (IDE) cho Java. Nó cung cấp các tính năng để phát triển ứng dụng desktop và trò chơi trên nền tảng Java.

3. Các thư viện, frameworks sử dụng

- Java Swing: là một bộ công cụ đồ họa được tích hợp sẵn trong Java Development Kit (JDK) để phát triển ứng dụng desktop trên nền tảng Java. Java Swing cung cấp các thành phần đồ họa như nút bấm, textbox, danh sách và các thành phần đồ họa khác để phát triển giao diện người dùng cho ứng dụng.

III. Chức năng của hàm

1. Tạo con rắn

```
public ConRan() {  
    x = new int[100];  
    y = new int[100];  
  
    x[0] = 5;  
    y[0] = 4;  
  
    x[1] = 5;  
    y[1] = 3;  
  
    x[2] = 5;  
    y[2] = 2;  
}
```

2. Reset trò chơi

```
public void resetGame() {  
    x = new int[100];  
    y = new int[100];  
  
    x[0] = 5;  
    y[0] = 4;  
  
    x[1] = 5;  
    y[1] = 3;  
  
    x[2] = 5;  
    y[2] = 2;  
  
    doDai = 3;  
    vector = ConRan.XUONG;  
}
```

3. Tạo môi

```
public Point layToaDoMoi() {  
    Random r = new Random();  
    int x;  
    int y;  
    do {  
        x = r.nextInt(19);  
        y = r.nextInt(19);  
    }while(tdCoNamTrongRan(x,y));  
  
    return new Point(x,y);  
}
```

4. Tạo vật cản

```
public Point layToaDoVC() {  
    Random r = new Random();  
    int x;  
    int y;  
    do {  
        x = r.nextInt(19);  
        y = r.nextInt(19);  
    }while(tdCoNamTrongRan(x,y));  
  
    return new Point(x,y);  
}
```

5. Tăng level, tốc độ trò chơi

```
if(doDai == maxLen) {
    GameScreen.CurrentLevel++;
    //
    GameScreen.bg1[layToaDoMoi().x][layToaDoMoi().y] = 3 ;
    //
    maxLen += 5;
    speed = getCurrentSpeed();
    vector = ConRan.XUONG;
}

public int getCurrentSpeed() {
    int speed = 200;
    for(int i=0; i<GameScreen.CurrentLevel; i++)
        speed*=0.8;
    return speed;
}
```

6. Xử lý va chạm

```
for(int i=1; i<doDai; i++) {
    if(x[0] == x[i] && y[0] == y[i]) {
        GameScreen.isPlaying = false;
        GameScreen.isGameOver = true;

        String name = JOptionPane.showInputDialog("Moi ban nhap ten: ");
        RanSanMoi.user.add(new User(name, String.valueOf(GameScreen.CurrentLevel), String.valueOf(GameScreen.diem)));

        GameScreen.diem = 0;
        GameScreen.CurrentLevel = 1;
        speed = 200;
        maxLen = 5;
    }
}

if(GameScreen.bg1[x[0]][y[0]] == 3) {
    GameScreen.isPlaying = false;
    GameScreen.isGameOver = true;

    String name = JOptionPane.showInputDialog("Moi ban nhap ten: ");
    RanSanMoi.user.add(new User(name, String.valueOf(GameScreen.CurrentLevel), String.valueOf(GameScreen.diem)));

    GameScreen.diem = 0;
    GameScreen.CurrentLevel = 1;
    speed = 200;
    maxLen = 5;
}
```

7. Xử lý các sự kiện nhấn phím

```

private class xuLy implements KeyListener{
    @Override
    public void keyTyped(KeyEvent e) {}
    @Override
    public void keyPressed(KeyEvent e) {
        if(e.getKeyCode() == KeyEvent.VK_SPACE) {
            GameScreen.isPlaying=!GameScreen.isPlaying;
            if(GameScreen.isGameOver) {
                GameScreen.isGameOver = false;
                game.ran.resetGame();
            }
        }
        if(e.getKeyCode() == KeyEvent.VK_UP) {
            game.ran.setVector(ConRan.LEN);
        }
        if(e.getKeyCode() == KeyEvent.VK_DOWN) {
            game.ran.setVector(ConRan.XUONG);
        }
        if(e.getKeyCode() == KeyEvent.VK_LEFT) {
            game.ran.setVector(ConRan.TRAI);
        }
        if(e.getKeyCode() == KeyEvent.VK_RIGHT) {
            game.ran.setVector(ConRan.PHAI);
        }
    }
    @Override
    public void keyReleased(KeyEvent e) {}
}

```

8. Ghi dữ liệu ra file

```

public static void UpdateData() {
    FileWriter fw;
    BufferedWriter bw = null;
    try {
        fw = new FileWriter("data/data.txt");
        bw = new BufferedWriter(fw);

        for(User u: user) {
            bw.write(u.getName() + " " + u.getLevel() + " " + u.getDiem());
            bw.newLine();
        }
    } catch (IOException e) {}
    finally {
        try {
            bw.close();
        } catch (IOException e) {}
    }
}

```

9. Đọc dữ liệu từ file

```

public static void ReadData() {
    FileReader fr;
    BufferedReader br = null;
    try {
        fr = new FileReader("data/data.txt");
        br = new BufferedReader(fr);

        String line = null;
        while((line = br.readLine()) != null){
            String[] str = line.split(" ");
            user.add(new User(str[0], str[1], str[2]));
        }
    } catch (IOException e) {}
    finally {
        try {
            br.close();
        } catch (IOException e) {}
    }
}

```

10. Vẽ các giao diện

```

public void paint(Graphics g) {
    paintBg(g);
    ran.veRan(g);
    veKhung(g);

    if(!isPlaying) {
        g.setColor(Color.white);
        g.setFont(g.getFont().deriveFont(22.0f));
        g.drawString("PRESS SPACE TO PLAY GAME!", 40, 200);
    }
    if(isGameOver) {
        g.setColor(Color.white);
        g.setFont(g.getFont().deriveFont(25.0f));
        g.drawString("GAME OVER!", 100, 250);
    }
    g.setColor(Color.white);
    g.setFont(g.getFont().deriveFont(28.0f));
    g.drawString("LEVEL: "+CurrentLevel, 450, 50);

    g.setFont(g.getFont().deriveFont(20.0f));
    g.drawString("Diem: "+diem, 450, 100);

    for(int i=0; i<RanSanMoi.user.size(); i++) {
        g.drawString(RanSanMoi.user.get(i).toString(), 450, i*30+150 );
    }
}

```



```

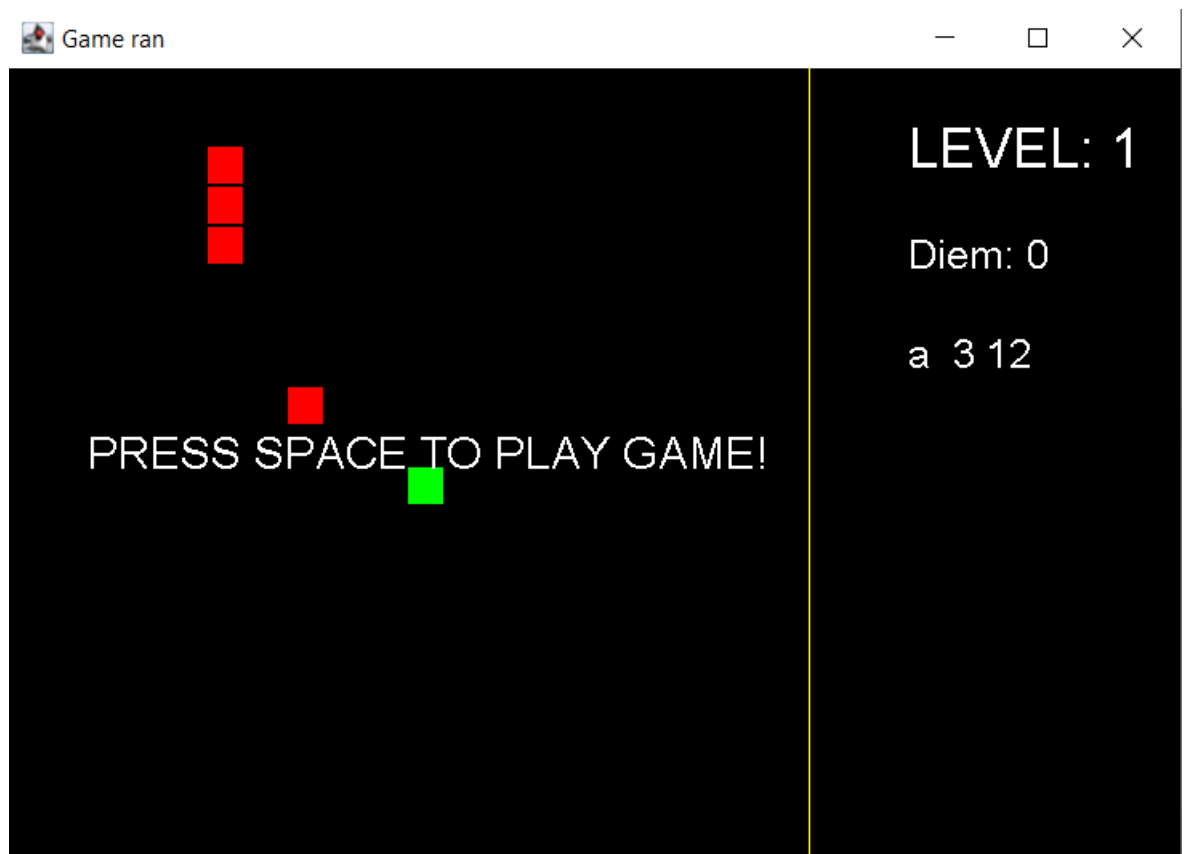
public void paintBg(Graphics g) {

    g.setColor(Color.black);
    g.fillRect(0, 0, WIDTH+200, HEIGHT);
    for(int i=0; i<20; i++)
        for(int j=0; j<20; j++) {
            g.fillRect(i*20, j*20, 18, 18);
            if(bg[i][j] == 2) {
                g.setColor(Color.red);
                g.fillRect(i*20, j*20, 18, 18);
                g.setColor(Color.black);
            }
            //
            if(bg1[i][j] == 3) {
                g.setColor(Color.green);
                g.fillRect(i*20, j*20, 18, 18);
                g.setColor(Color.black);
            }
            //
        }
    }
}

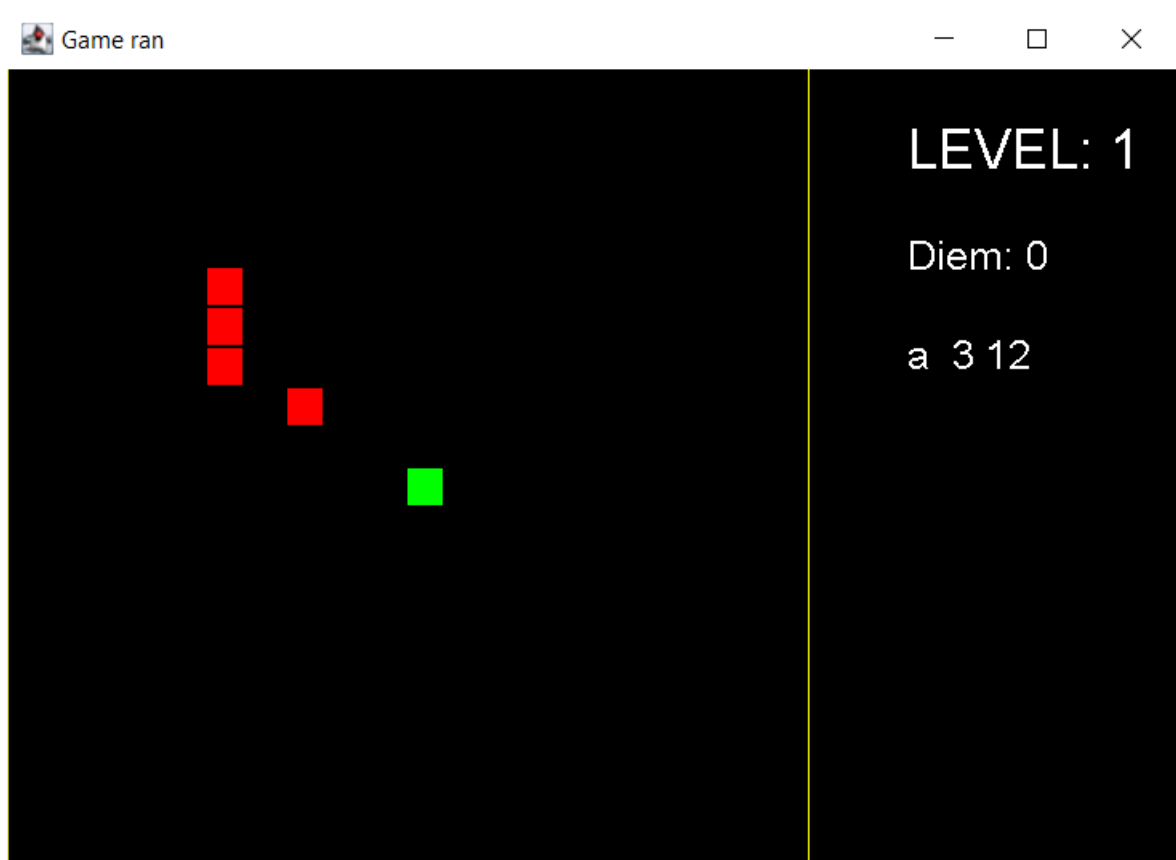
```

IV. Giao diện trò chơi

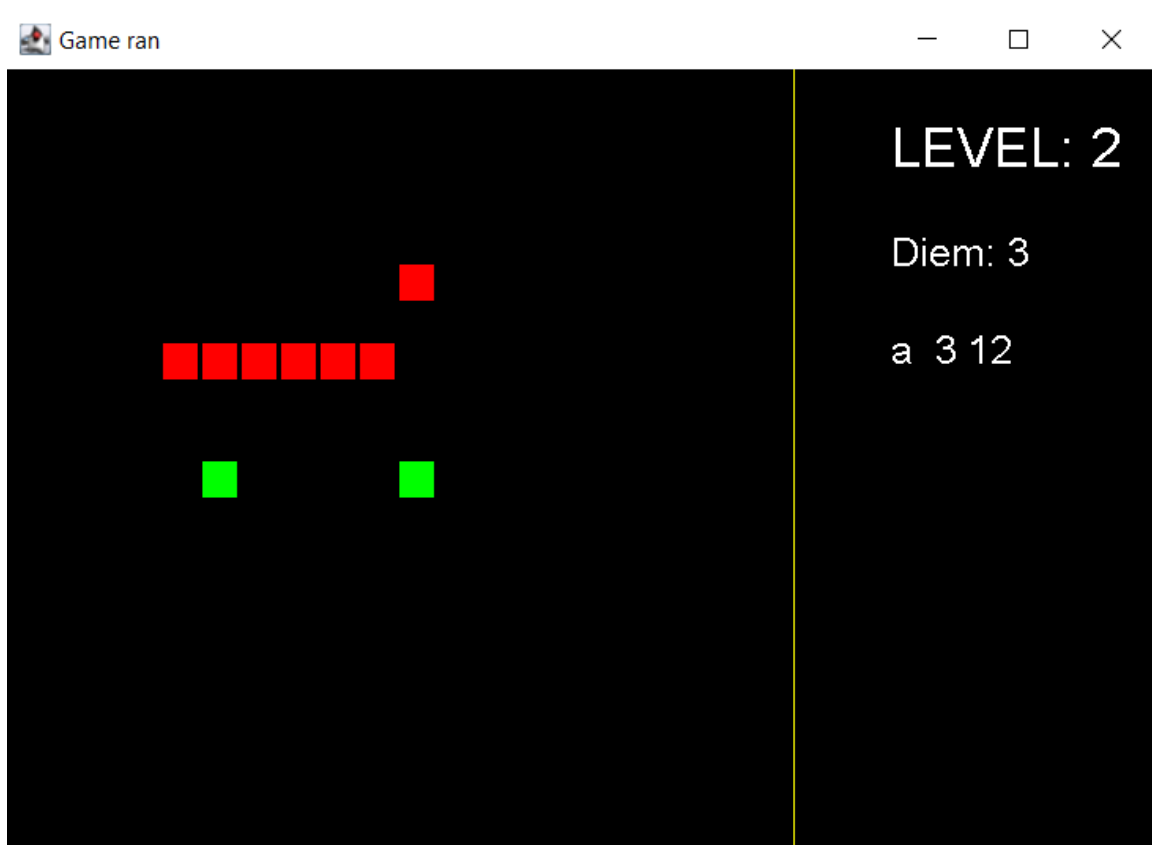
1. Giao diện bắt đầu



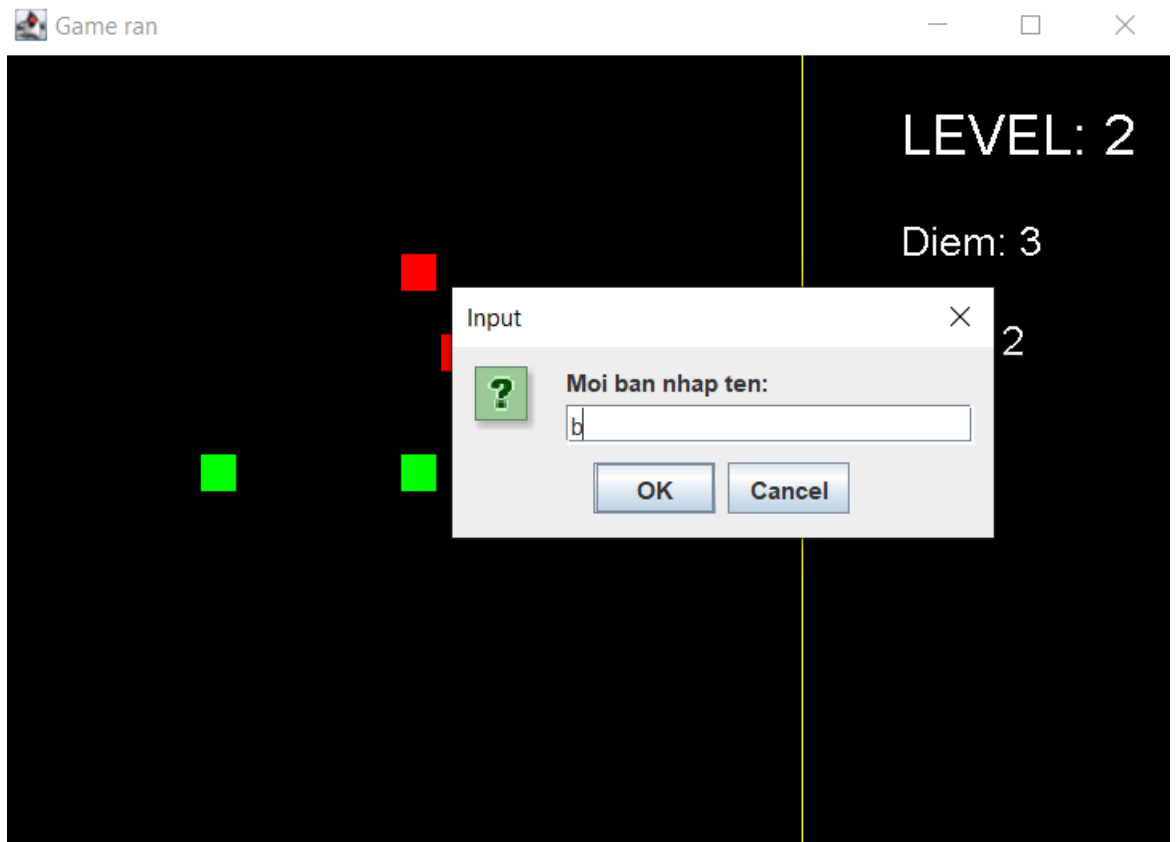
2. Giao diện khi chơi



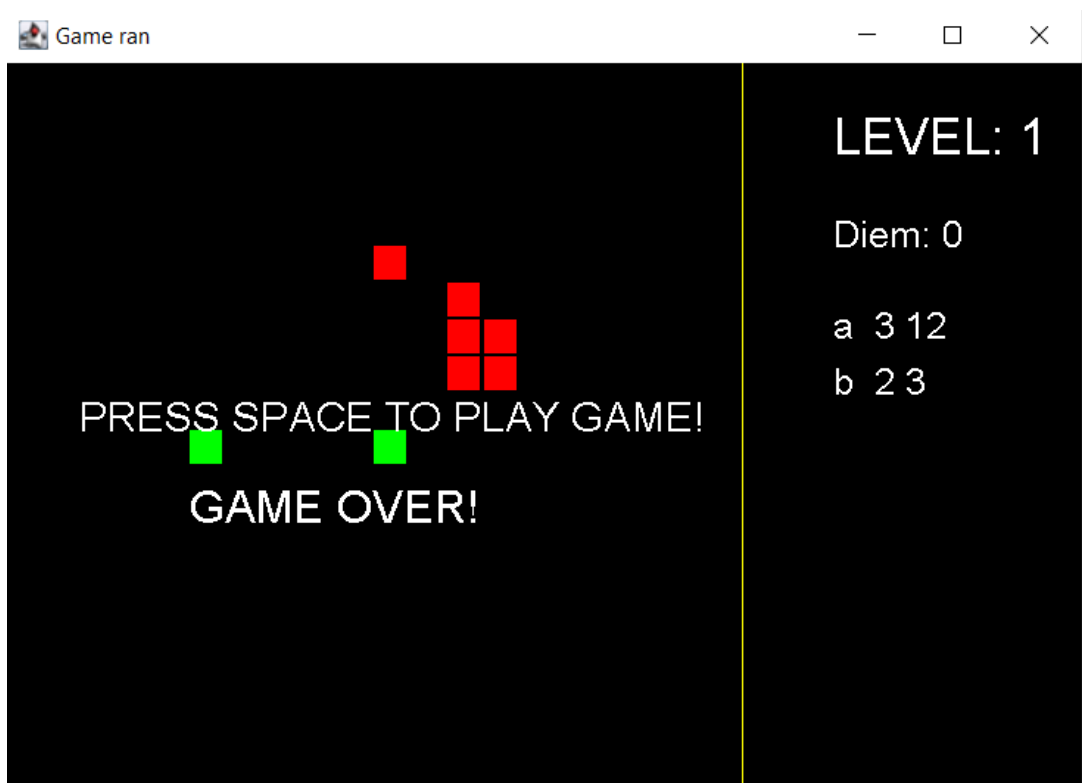
3. Giao diện khi tăng level



4. Giao diện ghi tên



5. Giao diện kết thúc



V. Kết luận.

1. Đánh giá về game

- Thành phần game đầy đủ, giao diện đơn giản.
- Các chức năng của game đều ổn định.

2. Kiến thức rút ra từ game.

- Hiểu rõ hơn về các Collection và IO.
- Bắt lỗi
- Dùng cơ sở dữ liệu để lưu trữ thông tin dữ liệu của game, dùng các câu truy vấn để tương tác với cơ sở dữ liệu trong công nghệ sử dụng.
- Sử dụng các phương thức chung bằng cách kế thừa từ một lớp.

VI. Các tài liệu tham khảo

- <https://www.youtube.com/@thachphamtech>
- <https://openplanning.net/>