Shape, square

Description automatically generated

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**UNIVERSITY OF TRANSPOST AND COMMUNICATIONS**

Logo

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

**MÔN CÔNG NGHỆ JAVA**

***ĐỀ TÀI: GAME 2048***

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Vũ Huấn** |
| **Sinh viên thực hiện:** | **Nguyễn Văn Khá** |
| **Mã sinh viên:** | **211203215** |
| **Lớp:** | **CNTT6 – K62 (N08)** |

**Hà Nội, ngày 3 tháng 5 Năm 2023**

**MỤC LỤC**

[**LỜI NÓI ĐẦU** 3](#_Toc133962927)

[**I.** **Giới thiệu** 4](#_Toc133962928)

[**1.** **Lịch sử ngôn ngữ Java** 4](#_Toc133962929)

[**2.** **Lịch sử game 2048** 4](#_Toc133962930)

[**3.** **Giao diện game** 4](#_Toc133962931)

[**4.** **Quy luật chơi** 6](#_Toc133962932)

[**II.** **Các công nghệ hướng đối tượng được sử dụng trong ứng dụng** 6](#_Toc133962933)

[**1.** **Kế thừa** 6](#_Toc133962934)

[**2.** **Bắt sự kiện trong Java AWT** 7](#_Toc133962935)

[**3.** **Giao diện tương tác với người dùng** 9](#_Toc133962936)

[**a)** **Gói Java AWT** 9](#_Toc133962937)

[**b)** **Gói Java Swing** 10](#_Toc133962938)

[**III.** **Thiết kế và chức năng** 12](#_Toc133962939)

[**1.** **Các lớp có trong game 2048** 12](#_Toc133962940)

[**2.** **Mô tả các lớp có trong game 2048** 12](#_Toc133962941)

[**a)** **Class JDBCUtil** 12](#_Toc133962942)

[**b)** **Class Main** 13](#_Toc133962943)

[**c)** **Class Data** 13](#_Toc133962944)

[**d)** **Class Game2048** 13](#_Toc133962945)

[**e)** **Class Music** 14](#_Toc133962946)

[**f)** **Class QuanLyData** 14](#_Toc133962947)

[**g)** **Class QuanLyManHinh** 17](#_Toc133962948)

[**h)** **Class VeManHinh** 18](#_Toc133962949)

[**IV.** **Kết luận** 20](#_Toc133962950)

[**V.** **Tài liệu tham khảo** 21](#_Toc133962951)

# **LỜI NÓI ĐẦU**

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn của em trong việc hoàn thành bài tập lớn này. Trong bài tập này, em đã phát triển một trò chơi 2048 bằng ngôn ngữ lập trình Java.

Trò chơi 2048 là một trò chơi trí tuệ đơn giản nhưng rất gây nghiện, nơi người chơi phải di chuyển các ô vuông và tạo ra một ô vuông có giá trị là 2048. Trong bài tập này, em đã sử dụng các kiến thức về lập trình hướng đối tượng, thuật toán và cấu trúc dữ liệu để phát triển trò chơi.

Báo cáo này sẽ trình bày vế quá trình phát triển trò chơi 2048 bằng Java, bao gồm các bước từ thiết kế đến triển khai và kiểm thử, hạn chế. Em cũng sẽ trình bày về các tính năng cơ bản của trò chơi, cách thức hoạt động, các thuật toán và cấu trúc dữ liệu được sử dụng để xử lý trò chơi, và các tính năng bổ sung khác.

Em hy vọng rằng báo cáo này sẽ cung cấp một cái nhìn tổng quan về quá trình phát triển trò chơi 2048 bằng Java và là một nguồn tham khảo hữu ích cho những ai quan tâm đến lĩnh vực lập trình game.

1. **Giới thiệu**
2. **Lịch sử ngôn ngữ Java**

Java là một công nghệ xây dựng các ứng dụng phần mềm có vị trí rất lớn trong những năm cuối thế kỷ XX đầu thế kỷ XXI. Nó được coi là công nghệ mang tính cách mạng và khả thi nhất trong việc tạo ra các ứng dụng có khả năng chạy thống nhất trên nhiều nền tảng mà chỉ cần biên dịch một lần.

Lần đầu tiên xuất hiện là vào năm 1995 như là một ngôn ngữ dùng trong nội bộ tập đoàn Sun Microsystems để xây dựng ứng dụng điều kiển các bộ xử lý bên trong máy điện thoại cầm tay, lò vi sóng, các thiết bị điện tử dân dụng khác. Không chỉ là một ngôn ngữ, Java còn là một nền tảng phát triển và triển khai ứng dụng trong đó máy ảo Java, bộ thông dịch có vai trò trung tâm.

Sun, công ty đã phát minh ra ngôn ngữ Java, chính thức phát hành bản Java Development Kit 1.0 vào năm 1996 hoàn toàn miễn phí để các nhà phát triển có thể tải về, học Java, xây dựng các ứng dụng Java và triển khai chúng trên các hệ điều hành có hỗ trợ Java. Khi nhắc đến Java người ta không còn chỉ nhắc đến Java như là một ngôn ngữ mà nhắc đến Java như là một công nghệ hay một nên tảng phát triển.

Cũng như nhiều trường Đại học khác, Đại học Giao Thông Vận Tải từ lâu đã đưa Java vào chương trình giảng dạy cho sinh viên. Bài tập lớn này nhằm mục đích giúp sinh viên tìm hiểu và bước đầu làm quen với lập trình Java, lập trình những ứng dụng Java đơn giản có mục đích và tĩnh hữu dụng trong thực tế.

1. **Lịch sử game 2048**

Game 2048 là một thể loại game tư duy trí tuệ. Cũng giống như game Xếp hình, game 2048 cũng là game block puzzle nhưng giao diện và luật chơi của nó hoàn toàn khác với game Xếp hình. Game 2048 được phát triển bởi một lập trình viên người Ý 19 tuổi tên là Gabriele Cirulli. Game đầu tiên được viết ra bằng JavaScript và CSS và phát hành vào ngày 09-03-2014. Game sau khi được đăng tải thì nhận được hơn 4 triệu lượt chơi trong gần 1 tuần. Sau đó, game 2048 đã được viết ra thêm trên nhiều nền tảng như Android, iOS, Windows,… và được biến tướng theo nhiều kiểu chơi và quy luật chơi khác nhau. Trong báo cáo của em về game 2048, em xin giới thiệu về game 2048 thuần do em viết bằng ngôn ngữ Java.

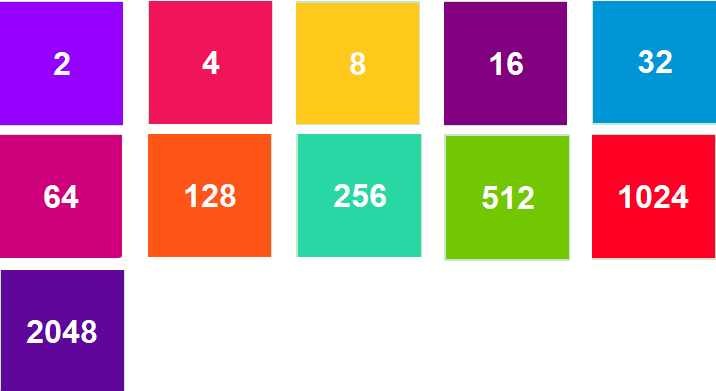
1. **Giao diện game**

A picture containing chart

Description automatically generated

*Màn hình chơi game*

Như hình ảnh phía trên ta thấy giao diện của game cực kì đơn giản, nó là một lưới các ô vuông ma trận 4x4. Các khối ô trong game có giá trị theo thứ tự là 2, 4, 6, 8, 16, 32, 64, 128, 256, 512, 1024, 2048.



*Các khối ô trong game*

1. **Quy luật chơi**

*2048* chơi trên một lưới vuông 4×4. Mỗi lần di chuyển là một lượt, người chơi sử dụng các phím mũi tên và các khối vuông sẽ trượt theo một trong bốn hướng tương ứng (lên, xuống, trái, phải). Mỗi lượt có một khối có giá trị 2 hoặc 4 sẽ xuất hiện ngẫu nhiên ở một ô trống trên lưới. Các khối vuông trượt theo hướng chỉ định cho đến khi chạm đến biên của lưới hoặc chạm vào khối vuông khác. Nếu hai khối vuông có cùng giá trị chạm vào nhau, chúng sẽ kết hợp lại thành một khối vuông có giá trị bằng tổng giá trị hai khối vuông đó (giá trị gấp đôi). Khối vuông kết quả không thể kết hợp với khối vuông khác một lần nữa trong một lượt di chuyển. Để dễ nhận biết thì các khối vuông giá trị khác nhau sẽ có màu sắc khác nhau.

Bảng điểm ở góc bên trái cho biết điểm của người chơi. Ban đầu điểm bằng 0. Khi hai khối vuông kết hợp thì người chơi sẽ tăng điểm là giá trị khối vuông mới. Bảng điểm bên góc phải là kỷ lục của những lượt chơi trước, ấn vào sẽ hiện những 3 người có điểm số cao nhất (Sẽ hiện “No Rank” nếu người chơi chưa từng chơi và chưa có kỷ lục). Ở giữa có nút “New Game” để chơi lại game từ đầu (Điểm người chơi sẽ về 0).

Khi người chơi tạo được ô vuông có giá trị 2048 thì thắng cuộc. Lúc này người chơi có thể chơi tiếp đến ô có giá trị cao hơn. Khi không còn nước đi hợp lệ thì trò chơi kết thúc.

1. **Các công nghệ hướng đối tượng được sử dụng trong ứng dụng**
2. **Kế thừa**

Tính kế thừa trong Java là một kỹ thuật mà trong đó một đối tượng thu được tất cả thuộc tính và hành vi của đối tượng cha. Ý tưởng đằng sau tính kế thừa trong Java là ta có thể tạo các lớp mới mà được xây dựng dựa trên các lớp đang tồn tại. Khi ta kế thừa từ một lớp đang tồn tại, ta có thể tái sử dụng các phương thức và các trường của lớp cha, và ta cũng có thể bổ sung thêm các phương thức và các trường khác. Tính kế thừa biểu diễn mối quan hệ IS-A, còn được gọi là mối quan hệ cha-con.

Khi chúng ta nói về tính kế thừa, từ khóa thường xuyên nhất được sử dụng là **extends** và **implements.** Những từ khóa này có thể định nghĩa một kiểu là loại IS-A của loại khác. Sử dụng những từ khóa, chúng ta có thể tạo một đối tượng sử dụng thuộc tính của đối tượng khác. Chúng ta sử dụng từ khóa extends của lớp con để có thể kế thừa thuộc tính của lớp cha trừ các thuộc tính private của lớp cha.

Cú pháp của Tính kế thừa tring Java:

Logo, company name

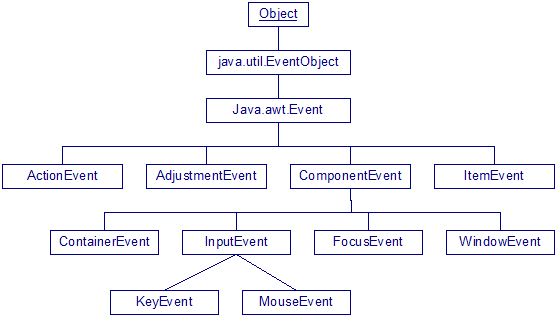
Description automatically generated

1. **Bắt sự kiện trong Java AWT**

Các hệ thống GUI xử lý các tương tác người dùng với sự trợ giúp của mô hình hướng sự kiện (event-driven).

Sự kiện trong Java:

* Sự kiện có thể được định nghĩa như là một kiểu tín hiệu tới chương trình thông báo một việc gì đó đã xảy ra.
* Sự kiện được sinh ra bởi các hành động của người dùng, ví dụ như di chuyển, click chuột, hay nhấn phím....



*Cây phân cấp các lớp của gói Event*

Việc xử lý những sự kiện này phụ thuộc vào ứng dụng. Abstract Windowing Toolkit (AWT) xử lý một vài sự kiện. Môi trường mà các ứng dụng này được thi hành ví dụ như trình duyệt cũng có thể xử lý các sự kiện khác. Người lập trình cũng cần phải xử lý những sự kiện nhất định và cần phải viết hàm xử lý các sự kiện đó.

Ứng dụng cần đăng ký một hàm xử lý sự kiện với một đối tượng. Hàm xử lý sự kiện này sẽ được gọi bất cứ khi nào sự kiện tương ứng phát sinh. JDK1.2 làm việc theo mô hình xử lý sự kiện này.

Trong quy trình này, ứng dụng cho phép bạn đăng ký các phương thức (handler), hay gọi là listener với các đối tượng. Những handler này tự động được gọi khi một sự kiện thích hợp phát sinh.

Một Event Listener lắng nghe một sự kiện nào đó mà một đối tượng đã thiết lập. Mỗi event listener cung cấp các phương thức xử lý những sự kiện này. Lớp thi hành listener cần phải định nghĩa những phương thức này. Để sử dụng mô hình này, bạn làm theo các bước sau:

* Cài đặt giao diện listener thích hợp. Cấu trúc như sau:

**public class MyApp extends Frame implements ActionListener**

* Xác định tất cả các thành phần tạo ra sự kiện. Các thành phần có thể là các button, label, menu item, hay window.

Ví dụ, để đăng ký một thành phần với listener, ta có thể sử dụng:

**exitbtn.addActionListener(This);**

Xác định tất cả các sự kiện được xử lý. Các sự kiện có thể là một ‘ActionEvent’ nếu một button được click hay một ‘mouseEvent’ nếu như chuột được kéo đi.

Thi hành các phương thức của listener và viết hàm xử lý sự kiện tương ứng với các phương thức.

|  |  |
| --- | --- |
| Lớp sự kiện | Mô tả |
| ActionEvent | Phát sinh khi một button được nhấn, một item trong danh sách chọn lựa được nhấn đúp (double-click) hay một menu được chọn. |
| TextEvent | Phát sinh khi giá trị trong thành phần textfield hay textarea bị thay đổi. |
| MouseEvent | Phát sinh khi chuột di chuyển, được click, được kéo hay thả ra. |
| KeyEvent | Phát sinh khi bàn phím ấn, nhả. |

*Các sự kiện khác nhau được sử dụng trong ứng dụng và mô tả về chúng*

Các giao diện cần được cài đặt để xử lý một trong số những sự kiện này là:

* + ActionListener
  + TextListener
  + MouseListener
  + MouseMotionListener
  + KeyListener

Các giao diện định nghĩa một số phương thức để xử lý mỗi sự kiện. Những phương thức này sẽ được nạp chồng trong lớp mà cài đặt những giao diện này.

1. **Giao diện tương tác với người dùng**

Các ứng dụng phần mềm hiện nay rất thân thiện vì được trình bày nhiều màn hình giao diện đồ họa đẹp mắt. Các ngôn ngữ lập trình hiện nay cung cấp các đối tượng đồ họa, chúng có thể được điều khiển bởi người lập trình, hay bởi người sử dụng. Một trong số những kết quả quan trọng nhất chính là các ngôn ngữ hiện nay được dựa trên Giao diện người dùng đồ họa (Graphical User Interface - GUI).

GUI cung cấp chức năng nhập liệu theo cách thân thiện với người dùng. GUI đa dạng từ ứng dụng đến ứng dụng và có thể chứa nhiều điều khiển như hộp văn bản, nhãn, hộp danh sách hay các điều khiển khác. Các ngôn ngữ lập trình khác nhau cung cấp nhiều cách khác nhau để tạo GUI. Các ngôn ngữ như VB hay VC++ có thể cung cấp chức năng kéo và thả trong khi đó phần mềm giống như C++ yêu cầu người lập trình phải viết toàn bộ mã để xây dựng GUI.

Một phần tử (element) GUI được thiết lập bằng cách sử dụng thủ tục sau:

* Tạo đối tượng
* Xác định sự xuất hiện ban đầu của đối tượng
* Chỉ ra nó nằm ở đâu
* Thêm phần tử vào giao diện trên màn hình

Một thành phần (Component) GUI là một đối tượng trực quan. Người dùng tương tác với đối tượng này thông qua con trỏ chuột hay bàn phím. Các thành phần như là button, label v.v… có thể được nhìn thấy trên màn hình. Bất kỳ cái gì chung cho tất cả các thành phần GUI đều được tìm thấy trong lớp Component. Để tạo các đối tượng GUI chúng ta cần nhập gói **java.awt** hoặc gói **javax.swing**.

Giao diện tương tác với người dùng của ứng dụng game 2048 được xây dựng dựa trên các gói **java.awt** và **javax.swing**. Sau đây là các lớp được sử dụng trong ứng dụng từ 2 gói trên.

1. **Gói Java AWT**

* **Lớp Color**
* Màu được tạo từ 3 màu: đỏ, xanh lá cây hoặc xanh da trời. Mỗi màu được biểu diễn bởi một giá trị kiểu byte, từ 0 (tối nhất) đến 255 (sáng nhất) -> *RGB model*.

Color c = new Color(r, g, b);

* r, g, và b chỉ tới 1 màu đỏ, xanh lá cây, xanh da trời bởi các components của nó.

Ví dụ: Color c = new Color(228, 100, 255);

* 13 màu chuẩn **(black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow)** được định nghĩa như hằng số trong java.awt.Color.
* Từ JDK 1.4: BLACK, BLUE, CYAN, DARK\_GRAY, GRAY, GREEN, LIGHT\_GRAY, MAGENTA, ORANGE, PINK, RED, WHITE, and YELLOW.
* Thiết lập màu nền (background) và nổi (foreground) : **setBackground(Color c)**

**setForeground(Color c)**

* **Lớp Graphics (Trong bài làm của em không sử dụng)**
* Lớp trừu tượng Graphics cung cấp giao tiếp đồ hoạ độc lập với thiết bị để hiện thị hình minh hoạ và ảnh trên màn hình của các nền tảng khác nhau. Bất kể thành phần nào (button, label ... ) được hiển thị, một đối tượng Graphic sẽ được tạo trên nền tảng gốc (native). Có thể lấy về đối tượng này thông qua phương thức getGraphics(). Ví dụ, đề lấy về graphic của một đối tượng nhãn jlblBanner
* Graphics graphics = jlblBanner.getGraphics();
* Sau đó, có thể sử dụng các phương thức của lớp Graphics để vẽ lên nhãn này.
* Phương thức paintComponent thuộc lớp JComponent :

**public void paintComponent(Graphics g)**

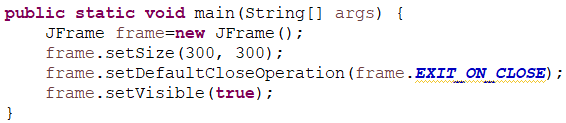
* Phương thức này được gọi mỗi khi hiển thị hoặc hiện thị lại.

1. **Gói Java Swing**

****

*Cấu trúc cây của gói Swing*

* **Frame:**
* Cửa sổ không nằm bên trong cửa sổ khác; thành phần cơ bản để chứa các GUI khác.
* Lớp JFrame được dùng đề tạo cửa sổ.
* Với các chương trình GUI Swing -> sử dụng JFrame -> tạo cửa sổ.



Thêm một button vào frame

**frame.getContentPane().add(new JButton("Button1"));**

* **Layout managers**
* Trình quản lý layout (sự bố trí) của Java cung cấp một mức độ trừu tượng hoá cho phép định hình giao diện lên tất cả các hệ thống cửa sổ.
* Các thành phần UI được đặt trong containe. Mỗi container có một trình quản lý layout để sắp xếp các thành phần này.
* Sử dụng phương thức setLayout(LayoutManager) của container.
* **Panels:**
* Là Container con để nhóm các thành phần UI.
* Recomend: đặt component, panel -> panel -> frameIl.
* Để thêm thành phần -> JFrame: thêm vào thành nội dung của JFrame. Để thêm thành phần vào panel -> thêm trực tiếp vào panel bằng phương thức add.
* **Tạo một Jpanel:**
* Sử dụng new JPanel() : layout mặc định là FlowLayout
* Sử dụng new JPanel(LayoutManager).
* Sử dụng phương thức add(Component)

**JPanel panel = new JPanel();**

**panel.add(new JButton("Button2"));**

* **Image Icon**
* Java sử dụng các lớp javax.swing.ImageIcon để biểu diễn cho một biểu tượng. Một biểu tượng là một hình ảnh kích thước cố định.
* Hình ảnh được thường được lưu trong tập tin hình ảnh. Bạn có thể sử dụng ImageIcon mới (tên tập tin) để xây dựng một biểu tượng hình ảnh.
* Ví dụ, câu lệnh sau đây tạo ra một Icon từ một file ảnh us.gif trong thư mục hình ảnh theo đường dẫn lớp hiện tại:

**Icon ImageIcon = new ImageIcon ("Assets/BG.jpg");**

1. **Thiết kế và chức năng**
2. **Các lớp có trong game 2048**

* Lớp có chức năng điều khiển game: QuanLyManHinh.java, QuanLyData.java
* Lớp có chức năng hỗ trợ giao diện, âm thanh: VeManHinh.java, Music.java
* Ngoài ra còn một số lớp khác như:

Graphical user interface, text, application

Description automatically generated

1. **Mô tả các lớp có trong game 2048**
2. **Class JDBCUtil**

* Mô tả: Đây là class kết nối game với cơ sở dữ liệu SQL.
* Nội dung: Kết nối với MySQL Workbean để lưu trữ điểm của người chơi.

Graphical user interface, text, application

Description automatically generated

1. **Class Main**

* Mô tả: Đây là nơi chạy game, nó quản lý cùng lúc 2 luồng, 1 luồng khởi tạo game và một luồng âm thanh chạy cùng lúc với luồng game.
* Nội dung: Sử dụng đa luồng (Multi-Thread) để chạy cùng lúc game và âm thanh.

Text

Description automatically generated

1. **Class Data**

* Mô tả: Đây là một Singleton Pattern để lưu trữ dữ liệu duy nhất một instance (data)
* Nội dung: Lưu dữ liệu các ô hiện có trên frame, ban đầu có một mảng 4x4 đều có giá trị là 0.

Text

Description automatically generated

1. **Class Game2048**

* Mô tả: Đây là class gọi hàm tạo frame, khỏi động luồng game.
* Nội dung: Luồng chạy game qua hàm run()

Text, letter

Description automatically generated

1. **Class Music**

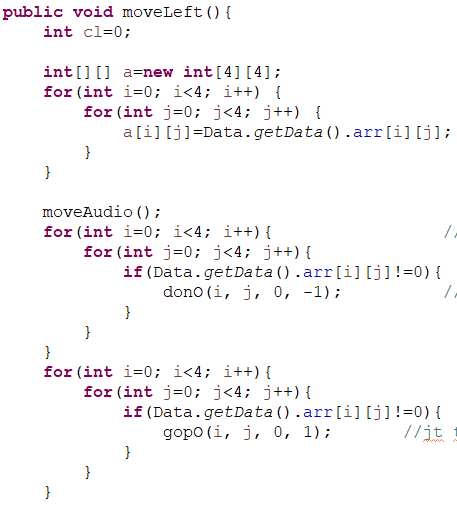
* Mô tả: Đây là class chạy file âm thanh
* Nội dung: Luồng âm thanh được gọi chạy cùng luồng game

Graphical user interface, text, application

Description automatically generated

1. **Class QuanLyData**

* Mô tả: Đây là một trong những class quan trọng nhất, quản lý dữ liệu các ô số, ô điểm mỗi khi di chuyển.
* Nội dung: Sử dụng các hàm random, dữ liệu các ô số khi ta di chuyển thay đổi
* Hàm di chuyển trái:



Text

Description automatically generated

* Tương tự cho các hàm di chuyển phải, lên, xuống

Graphical user interface, text

Description automatically generated

* Hàm dồn các ô lại một phía khi di chuyển

Text

Description automatically generated

* Hàm gộp 2 ô cùng giá trị

Text

Description automatically generated

* Hàm kiểm tra game đã kết thúc hay chưa

Text

Description automatically generated

* Hàm tìm giá trị lớn nhất khi game kết thúc

Text, letter

Description automatically generated

1. **Class QuanLyManHinh**

* Mô tả: Hiển thị frame lên màn hình
* Nội dung: Là nơi tiếp nhận các hành động từ bàn phím

Text, letter

Description automatically generated

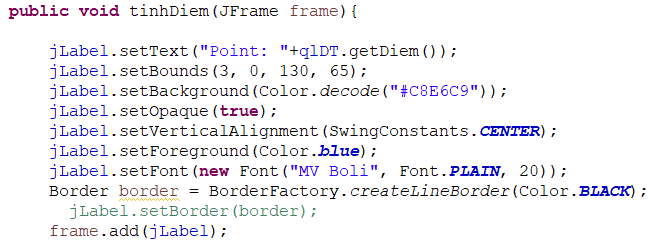
* Lấy sự kiện ta ấn nút bàn phím

A picture containing text

Description automatically generated

1. **Class VeManHinh**

* Mô tả: Nơi hiển thị các thay đổi trên màn hình frame
* Nội dung: Hiển thị sự thay đổi của các ô số, các ô điểm
* Hàm tính điểm



* Hàm vẽ các ô số

Text, letter

Description automatically generated

* Hàm xét màu cho các ô số

Table

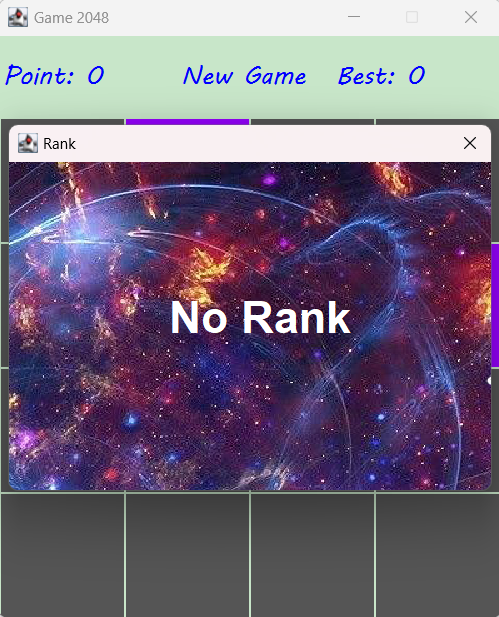
Description automatically generated with low confidence

* Hàm cập nhật màn hình sau mỗi sự kiện ấn nút **(Đây là hàm được gọi nhiều nhất và vô cùng quan trọng)**

Text

Description automatically generated

* Hiển thị ra màn hình khi chưa có rank



1. **Kết luận**

Hiện nay, lập trình game đang là một lĩnh vực mới thu hút nhiều lập trình viên. Việc xây dựng các ứng dụng vui chơi giải trí là rất cần thiết cho sự phát triển công nghệ. Trong phạm vi đề tài, em chỉ trình bày những phần công nghệ cơ bản có liên quan đến ứng dụng như lập trình giao diện người dùng với Swing và AWT và bắt sự kiện với AWT.

Nhìn chung game tạo ra đáp ứng nhu cầu giải trí của người chơi. Giao diện thân thiện, đơn giản, dễ sử dụng và thao tác. Tuy nhiên vẫn còn một số lỗi như chưa nghĩ đến việc ô số 4096 trở lên có màu gì, giao diện game chưa được đẹp lắm,…

Tuy vậy, do những hạn chế về trình độ, thời gian cũng như thiết bị nên em không khỏi những vướng mắc, sai sót trong quá trình tìm hiểu và nghiên cứu. Em rất mong nhận được những đánh giá và chỉnh sửa từ các thầy.

Em xin chân thành cảm ơn!

1. **Tài liệu tham khảo**
2. Website: [Tại đây](https://tuicocach.com/lap-trinh-game-2048-tren-android-studio-ma-nguon-game-2048/)
3. Youtube: [Tại đây](https://www.youtube.com/watch?v=5GUP9bngvu8&list=PLO5kKXGLQ2OoEtDYjMCiog1NLIVIObIuO)