🙠🙟🕮🙝🙢

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

~~~~~~\*~~~~~~



**BÁO CÁO HỌC PHẦN**

*ĐỀ TÀI :* QUÁ TRÌNH HÌNH THÀNH VÀ PHÁT TRIỂN CỦA TƯ TƯỞNG HỒ CHÍ MINH

(1890-1920)

|  |  |  |
| --- | --- | --- |
| *Sinh viên thực hiện* | : | DƯƠNG XUÂN BÁCH |
| *Lớp* | : | CNTT-6 |
| *Mã sinh viên* | : | 211210614 |
| *Giảng viên* | : | Vũ Huấn |

HÀ NỘI – 2023

|  |
| --- |
| **MỤC LỤC**  [LỜI NÓI ĐẦU 3](#_Toc134025528)  [I. TỔNG QUAN ĐỀ TÀI 4](#_Toc134025529)  [1. Giới thiệu chung về công nghệ sử dụng 4](#_Toc134025530)  [2. Giới thiệu đề tài 4](#_Toc134025531)  [3. Đặt vấn đề 4](#_Toc134025532)  [II. THIẾT KẾ CHƯƠNG TRÌNH 5](#_Toc134025533)  [III. XÂY DỰNG CHƯƠNG TRÌNH 6](#_Toc134025534)  [1. Xây dựng 6](#_Toc134025535)  [2. Kêt quả 14](#_Toc134025536)  [3. Thử nghiệm 15](#_Toc134025537)  [IV. KẾT LUẬN VÀ ĐÁNH GIÁ 19](#_Toc134025538)  [1. Kết quả đạt được 19](#_Toc134025539)  [2. Hạn chế 19](#_Toc134025540)  [3. Hướng phát triển 19](#_Toc134025541)  [V. TÀI LIỆU THAM KHẢO VÀ PHỤ LỤC 20](#_Toc134025542)  [1. Tài liệu tham khảo 20](#_Toc134025543)  [2. Phụ lục 20](#_Toc134025544) |

# LỜI NÓI ĐẦU

Game máy tính cổ điển bắt đầu phát triển vào thập niên 1960 với các trò chơi đơn giản như Tic Tac Toe và những trò chơi đơn giản khác. Trong những năm 1970, game Pong đã trở thành một hiện tượng và đã mở ra một thời đại mới cho ngành công nghiệp game. Vào những năm 1980, các game console như Atari và Nintendo đã trở thành hành tinh của trò chơi điện tử. Sau đó, vào những năm 1990, PC trở thành nền tảng chính cho các game máy tính, và các trò chơi như Doom và Sim City đã trở thành các trò chơi kinh điển của thời đại đó. Và hiện nay, ngành công nghiệp game đã phát triển mạnh mẽ với nhiều thể loại trò chơi đa dạng và sự bùng nổ của game di động, đưa các trò chơi vào một tầm cao mới.

Trong báo cáo này, em sẽ trình bày về quá trình lập trình game Minesweeper - một trò chơi logic phổ biến đã xuất hiện từ thập niên 1990. Minesweeper là một trò chơi đơn giản, trong đó người chơi phải tìm ra tất cả các ô không có mìn trong một bảng ô vuông bằng cách mở ô một cách cẩn thận, tránh mìn và dùng những số hiển thị trên bảng để dự đoán vị trí của mìn.

Công nghệ nền tảng mà em sử dụng là Java. Lập trình game bằng Java đòi hỏi kiến thức về lập trình hướng đối tượng và các thư viện đồ họa như AWT, Swing để tạo ra giao diện đồ họa. Sử dụng Java để phát triển game cũng có lợi thế về tính di động và đa nền tảng, vì Java được hỗ trợ trên nhiều hệ điều hành khác nhau.

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu chung về công nghệ sử dụng

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

## Giới thiệu đề tài

Dò mìn hay gỡ bom ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh" \o "Tiếng Anh): Minesweeper) là một trò chơi [giải đố](https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i_%C4%91%E1%BB%91" \o "Giải đố) trên [máy tính](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh" \o "Máy tính) dành cho [một người chơi](https://vi.wikipedia.org/wiki/Tr%C3%B2_ch%C6%A1i_%C4%91i%E1%BB%87n_t%E1%BB%AD_m%E1%BB%99t_ng%C6%B0%E1%BB%9Di_ch%C6%A1i" \o "Trò chơi điện tử một người chơi). Trò chơi bao gồm một "bãi mìn" là những ô vuông có thể chứa "mìn", và người chơi cần phải dựa vào những con số thể hiện số mìn xung quanh để mở hết tất cả những ô vuông trống mà không kích nổ quả mìn nào. Trò chơi được xây dựng như một chương trình giải trí cài đặt trên hệ điều hành [Microsoft Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows).

Hiện nay, trò chơi Dò mìn rất phổ biến. Ngoài Windows, trò chơi còn có trong [Linux](https://vi.wikipedia.org/wiki/Linux) và gnomine. Nhờ quy luật đơn giản, nhiều biến thể của trò chơi cũng được phát hành như Dò mìn 3D, Minesweeper X, và Crossmines.

## Đặt vấn đề

Trong Dò mìn, người chơi phải mở được tất cả các ô không có mìn trên một bảng ô vuông, đồng thời không được kích nổ bất cứ một quả mìn nào. Trò chơi được xếp hạng bằng thời gian hoàn thành, vì vậy việc hoàn thành trò chơi trong thời gian sớm nhất cũng là một mục tiêu quan trọng đối với người chơi đã thành thạo.

# THIẾT KẾ CHƯƠNG TRÌNH

**Quá trình thiết kế chương trình:**

**Bước 1: Xác định yêu cầu**

Bước đầu tiên là xác định các yêu cầu chức năng và phi chức năng của chương trình. Điều này bao gồm sự tìm hiểu về mục đích của chương trình, các tính năng cần thiết, dữ liệu đầu vào và đầu ra cũng như các ràng buộc về kỹ thuật và chức năng.

**Bước 2: Phác thảo chương trình**

Sau bước xác định các yêu cầu của chương trình là quá trình lập kế hoạch thiết kế. Kế hoạch này bao gồm việc phân tích và thiết kế kiến trúc chương trình, chỉ định chức năng, tạo sơ dữ liệu.

**Bước 3: Thiết kế giao diện**

Thiết kế giao diện bao gồm tạo bố cục, chọn màu, định vị các nút và bố cục giao diện.

**Bước 4: Lập trình**

Sử dụng Java để có thể lập trình các đối tượng, tương tác, chức năng, cũng như điều chỉnh theo bố cục đã có.

**Bước 5: Sửa lỗi**

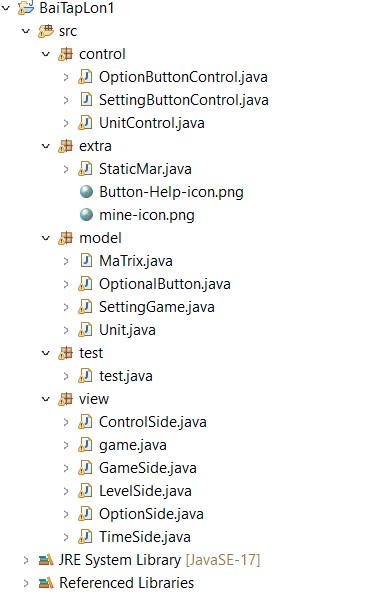
Sau khi viết xong mã, bước tiếp theo là kiểm tra chương trình để đảm bảo nó hoạt động đúng và không có lỗi.

# XÂY DỰNG CHƯƠNG TRÌNH

## Xây dựng

* 1. Tổng quan cấu trúc chương trình

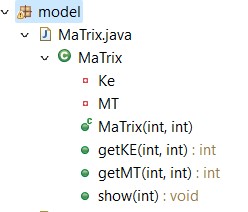
Chương trình được xây dựng dựa trên mô hình thiết kế MVC.



* 1. Chi tiết

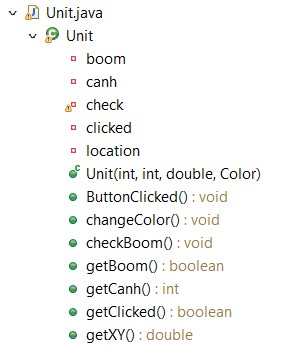
b.1. Model

b.1.1. MaTrix



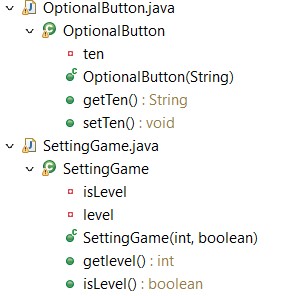
b.1.2. Unit

Lớp Unit kế thừa lớp JButton



b.1.3. SettingGame & OptionButton

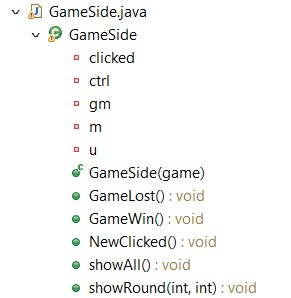
Lớp SettintGame và lớp OptionButton kế thừa lớp JButton



b.2. View

b.2.1. GameSide

Lớp GameSide kế thừa lớp JPanel



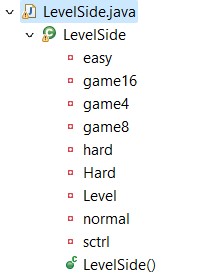
b.2.2. ControlSide

Lớp ControlSide kế thừa lớp JPanel



b.2.2.1. LevelSide

Lớp LevelSide kế thừa lớp JPanel



b.2.2.2. TimeSide

Lớp TimeSide kế thừa lớp JPanel và interface Runnable



b.2.2.3. OptionSide

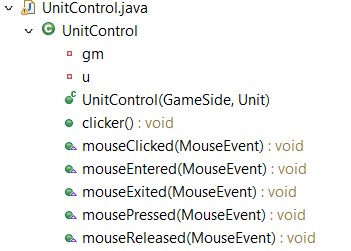
Lớp LevelSide kế thừa lớp JPanel



b.3. Control

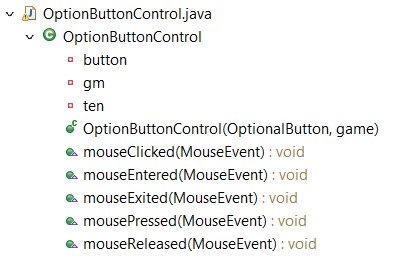
b.3.1. UnitControl

Lớp UnitControl kế thừa interface MouseListener



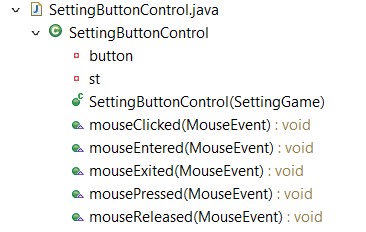
b.3.2. OptionButtonControl

Lớp OptionButtonControl kế thừa interface MouseListener



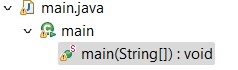
b.3.3. SettingButtonControl

Lớp SettingButtonControl kế thừa interface MouseListener



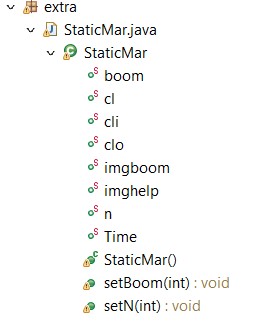
b.4. Test

b.4.1. Main



b.5. Extra

Chứa các hình ảnh sử dụng và các giá trị chung



* 1. Tương tác và Chức năng

c.1. Game

Giao diện của Game là một Ma trận các nút, người chơi có nhiệm vụ mở tất cả các ô ở những an toàn và né các ô chứa bom.

public GameSide(game gm) {

super();

this.setLayout(new GridLayout(StaticMar.n, StaticMar.n));

this.gm = gm;

this.m = new MaTrix(StaticMar.n, StaticMar.boom);

this.u = new Unit[StaticMar.n + 2][StaticMar.n + 2];

this.ctrl = new UnitControl[StaticMar.n + 2][StaticMar.n + 2];

for (int i = 1; i <= StaticMar.n; i++) {

for (int j = 1; j <= StaticMar.n; j++) {

double x = i + (double) j / 100;

u[i][j]=newUnit(this.m.getMT(i, j), this.m.getKE(i, j), x, StaticMar.cl);

ctrl[i][j] = new UnitControl(this, u[i][j]);

u[i][j].addMouseListener(ctrl[i][j]);

this.add(u[i][j]);

}

}

}

c.2. Độ khó

Game hiện tại có 3 kích thước của Ma trận, ứng với mỗi kích thước sẽ có 3 độ khó tương ứng là dễ, vừa và khó. Số bom trong mỗi màn chơi sẽ phụ thuộc vào kích thước của ma trận và độ khó của màn.

public void mouseClicked(MouseEvent e) {

if (this.button.isLevel() == true) st.setN(this.button.getlevel());

if (this.button.isLevel() == false) st.setBoom(this.button.getlevel());

}

c.3. Các nút chức năng

c.3.1. Button New

Khi nhấp chuột vào nút “New”, toàn bộ các thông số liên quan sẽ được thiết lập lại , phần giao diện trò chơi của ứng dụng sẽ được lấp lại.

public void newGame() {

this.remove(game);

this.remove(box);

box = new JPanel();

this.game = new GameSide(this);

this.option = new OptionSide(this);

option.cancelledTask();

box.add(emptySide, BorderLayout.WEST);

box.add(option, BorderLayout.CENTER);

this.add(BorderLayout.CENTER, game);

this.add(BorderLayout.EAST, box);

this.init();

}

c.3.2. Button Key

Khi nhấp chuột vào nút “Key”, toàn bộ các nút trên phần trò chơi sẽ được mở ra.

public void showAll() {

for (int i = 1; i <= StaticMar.n; i++) {

for (int j = 1; j <= StaticMar.n; j++) {

u[i][j].checkBoom();

}

}

}

c.3.3. Button Record

Khi nhấp chuột vào nút “Record”, một frame chứa thông tin những người chơi đã hoàn thành trò chơi sẽ hiện ra.

public void showRecord() {

try {

FileReader fr = new FileReader("D:\\java-2022- 12\\BaiTapLon1\\src\\test\\save.txt");

BufferedReader br = new BufferedReader(fr);

JFrame f= new JFrame();

f.setSize(400,300);

f.setLocationRelativeTo(null);

JLabel jp = new JLabel();

f.add(jp);

Scanner sc = new Scanner(br);

String td;

String out = "";

while(sc.hasNextLine()) {

out += sc.nextLine()+ " "+"\r\n";

}

jp.setText(out);

sc.close();

f.setVisible(true);

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

## Kêt quả



## Thử nghiệm

* 1. Sử dụng nút Key



* 1. Sau khi thua cuộc

Sau khí nhấp phải ô có bom sẽ xuất hiện thông báo kết thúc game, người chơi có thể lựa chọn chơi lại hoặc không.

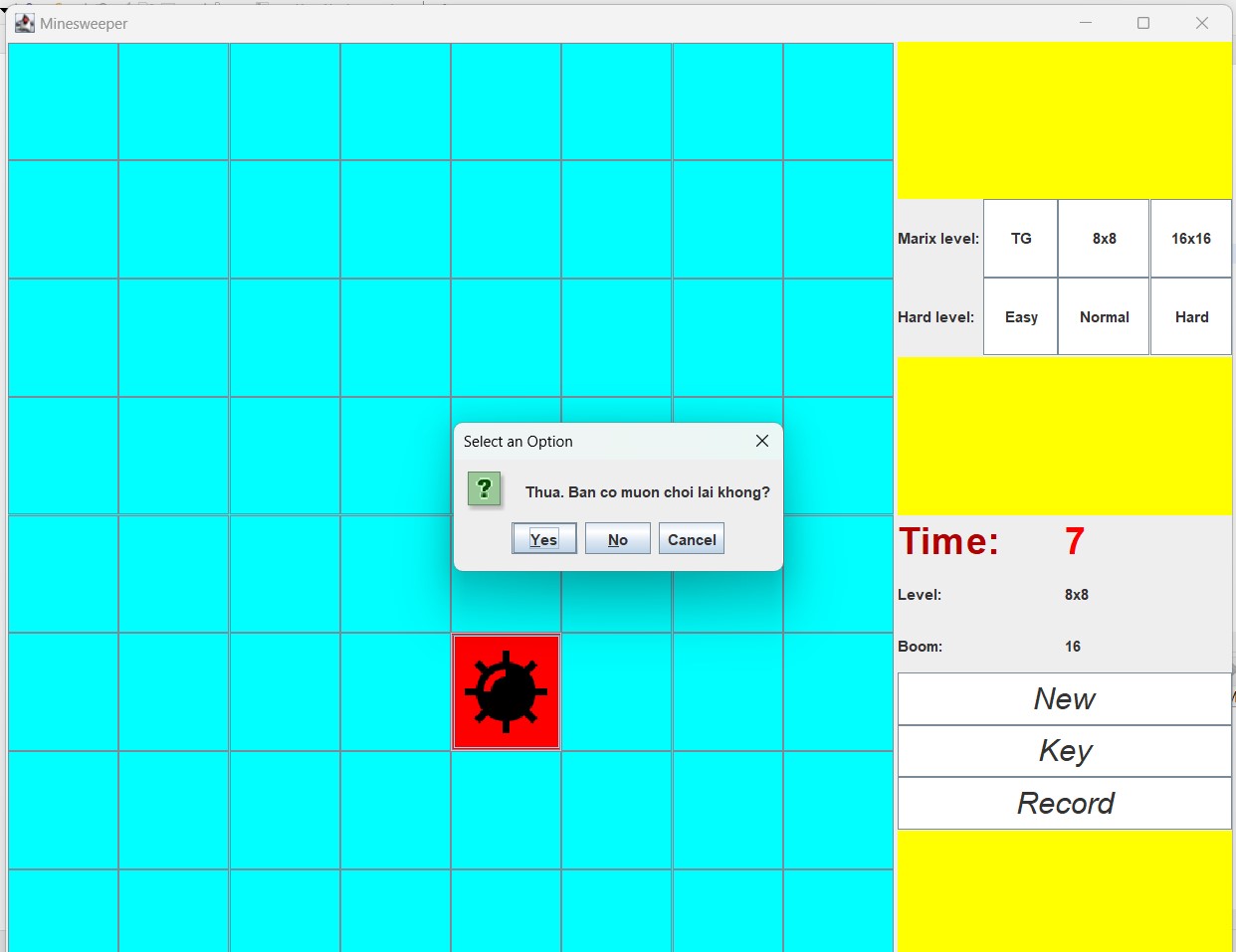
public void GameLost() {

int a = JOptionPane.showConfirmDialog(this, "Thua. Ban co muon choi lai khong?");

if (a == JOptionPane.YES\_OPTION)

this.newGame();

}



* 1. Sau khi thắng cuộc

Người chơi có thể điền tên của mình vào ô để lưu kết quả chiến thắng màn chơi.

public void GameWin(){

String s = JOptionPane.showInputDialog("Ban da thang, ban co muon luu ten hay khong?");

FileWriter pw = null;

try {

try {

pw=newFileWriter("D:\\java-2022-12\\BaiTapLon1\\src\\test\\save.txt",true);

} catch (IOException e1) {

e1.printStackTrace();

}

BufferedWriter bw = new BufferedWriter(pw);

try {

bw.write(s+" "+StaticMar.n+" "+ StaticMar.boom+" "+StaticMar.Time);

bw.newLine();

} catch (IOException e) {

e.printStackTrace();

}finally {

bw.close();

pw.close();

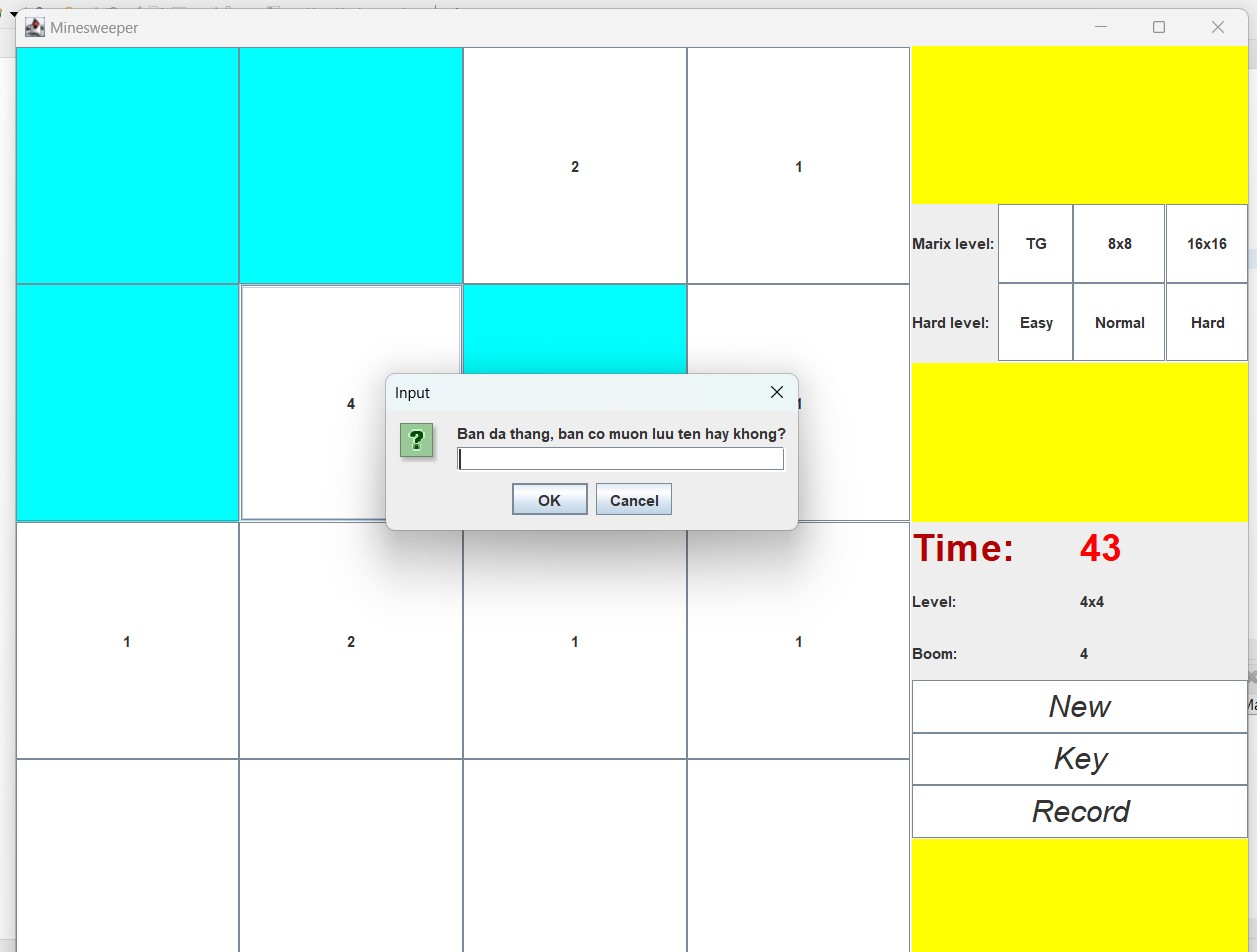
}

} catch (Exception e) {

e.printStackTrace();

}

}



# KẾT LUẬN VÀ ĐÁNH GIÁ

## Kết quả đạt được

* Game đã có được các chức năng cơ bản giống như tựa Game gốc.
* Giao diện dễ sử dụng, dễ tiếp cận.

## Hạn chế

* Game chưa tối ưu được một vài chức năng.
* Độ khó của game chưa được tối ưu, chức năng tự động dò mìn của game chưa hoàn thiện.
* Giao diện trò chơi còn đơn giản.

## Hướng phát triển

* Tối ưu thao tác trò chơi.
* Trau chuốt giao diện than thiện với người dung.
* Hoàn thiện chức năng dò mìn.
* Thiết kế đa dạng màn chơi, lối chơi, tránh sự nhàm chán.

# V. TÀI LIỆU THAM KHẢO VÀ PHỤ LỤC

## Tài liệu tham khảo

<https://titv.vn/courses-page/lap-trinh-java-java-core/>

<https://minesweeper.one/>

Slide bài giảng, tài liệu giảng dạy

## Phụ lục

Link chương trình: https://github.com/ImCypr/BTL