**LỜI CẢM ƠN**

Cùng với sự giúp đỡ hướng dẫn tận tình của thầy Đặng Như Phú em đã phát triển nên game sin tồn IronMan. Giúp người chơi thư giãn , giải trí sau những giờ học tập căng thẳng game còn là công cụ rèn luyện kỹ năng phản xạ, tính tập trung, khả năng trính toán suy nghĩ để có thể vượt qua các thử thách của game.

Do thời gian còn hạn hẹp cũng như chưa có Kinh nghiệm làm một game lớn ,nền tảng Framework mới xuất hiện trong những năm gần đây , chương trình thành phầm của em có thể còn tồn tại lỗi không mong muốn. Rất mong nhận được góp ý của cô để chúng em hoàn thiện hơn và rút kinh nghiệm cho những đồ án lớn hơn trong tương lai.

Một lần nữa, em xin chân thành cảm ơn thầy và chúc thầy nhiều sức khỏe.

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Thủ Dầu Một, Ngày tháng năm 2019*

G**iáo viên hướng dẫn**

**LỜI NÓI ĐẦU**

Trong thời đại công nghệ thông tin như hiện nay, sản phẩm công nghệ ngày càng chịu sự đánh giá khắt khe hơn từ phía những người dùng, đặc biệt là về sản phẩm Game được nhận rất nhiều sự đánh giá từ phía các Game thủ, hay chỉ là những người chơi bình thường. Ngành công nghiệp Game hiện nay có thể nói là bùng nổ, với tốc độ phát triển đến chóng mặt, rất nhiều những Game hay và hấp dẫn đã được ra đời trong thời gian qua. Phía sau những Game phát triển và nổi tiếng như vậy đều có những ngôn ngữ lập trình như c#, java …. Trong đó java là một ngôn ngữ hỗ trợ, giúp người phát triển viết Game một cách nhanh chóng và đơn giản..

Từ xu hướng phát triển và những bất cập trên, đồ án này sẽ khảo sát và nghiên cứu về ngôn ngữ Java– một ngôn ngữ rất phổ biến và không kém mạnh mẽ hiện nay nhằm thực nghiệm việc phát triển một trò chơi (Demo) game nhập vai megaman. Chuẩn bị kiến thức và kỹ năng cho định hướng nghề nghiệp (phát triển Game) sau này của chúng em, góp phần vào sự phát triển của ngành công nghiệp Game nước nhà. Các chương đầu trong tài liệu sẽ trình bày lần lượt các khái niệm chung về java.

Các chương tiếp theo sẽ giới thiệu về Java nâng cao bao gồm các đặc điểm, tính năng, công cụ và thành phần trong java. Sau những nội dung về thiết kế, tài liệu sẽ trình bày về việc lập trình trên Eclipes khi giới thiệu về ngôn ngữ lập trình cùng các lớp, hàm trong thư viện dựng sẵn của Java thông qua các ví dụ thực tế khi phát triển một game nhập vai 2D. Cuối cùng là giới thiệu về bối cảnh, cốt truyện, tài liệu hướng dẫn sử dụng của trò chơi Demo cùng chương tổng kết về các khó khăn và những thành quả trong suốt quá trình phát triển trò chơi Demo trên Eclipes bằng ngôn ngữ Java

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc58237711)

[LỜI NÓI ĐẦU 4](#_Toc58237712)

[MỤC LỤC HÌNH ẢNH 7](#_Toc58237713)

[CHƯƠNG I: KHẢO SÁT VÀ THU THẬP YÊU CẦU 8](#_Toc58237714)

[1. Thông tin cá nhân: 8](#_Toc58237715)

[2. Các công nghệ sử dụng trong đề tài: 8](#_Toc58237716)

[3. Khảo sát hiện trạng: 8](#_Toc58237717)

[a) Mô tả Game: 8](#_Toc58237718)

[b) Mô tả yêu cầu 8](#_Toc58237719)

[c) Các chức năng của game: 9](#_Toc58237720)

[CHƯƠNG 2: TÌM HIỂU CÁC CÔNG NGHỆ 10](#_Toc58237721)

[1. Java cơ bản: 10](#_Toc58237722)

[a) Giới thiệu về ngôn ngữ java 10](#_Toc58237723)

[b) Tổng quan về Java 10](#_Toc58237724)

[c) Đặc điểm của ngôn ngữ lập trình java 10](#_Toc58237725)

[d) Phân tích chi tiết chương trình java đầu tiên 11](#_Toc58237726)

[2. Java nâng cao: 12](#_Toc58237727)

[a) Khái niệm hướng đối tượng opp trong java 12](#_Toc58237728)

[b) Lập trình hướng đối tượng 12](#_Toc58237729)

[3. Java swing : 14](#_Toc58237730)

[a) Giới thiệu về Java Swing 14](#_Toc58237731)

[b) Phân cấp các lớp Java Swing 14](#_Toc58237732)

[c) Các phương thức thường dùng của lớp Component 15](#_Toc58237733)

[CHƯƠNG 3: XÂY DỰNG GAME 16](#_Toc58237734)

[1. Các phần mềm , ngôn ngữ sử dụng trong game 16](#_Toc58237735)

[2. Dữ liệu trong game 16](#_Toc58237736)

[3. Thiết kế giao diện Game Nhập Vai 18](#_Toc58237737)

[4. Tạo nhân vật trong Game Nhập Vai 20](#_Toc58237738)

[5. Tạo các trải nghiệm cho Game Nhập Vai 21](#_Toc58237739)

[6. Hướng dẫn cách chơi game 25](#_Toc58237740)

[KẾT LUẬN 27](#_Toc58237741)

[1. Đánh giá kết quả 27](#_Toc58237742)

[1.1. Kết quả đạt được: 27](#_Toc58237743)

[1.2. Hạn chế của đề tài: 27](#_Toc58237744)

[2. Hướng phát triển của đề tài 27](#_Toc58237745)

[TÀI LIỆU KHAM KHẢO 29](#_Toc58237746)

# MỤC LỤC HÌNH ẢNH

[Hình ảnh 0‑1: hình ảnh đồ họa 2D của map được đổi thành mảng 2 chiều 16](#_Toc58237521)

[Hình ảnh 0‑2: hình ảnh các thao tác của nhân vật và các viên đạn 17](#_Toc58237522)

[Hình ảnh 0‑3 : các file dữ liệu của game được lưu trong data trong project 17](#_Toc58237523)

[Hình ảnh 0‑4 : hình ảnh các thao tác của boss trong game 18](#_Toc58237524)

[Hình ảnh 0‑5 : Giao diện chờ để vào game 19](#_Toc58237525)

[Hình ảnh 0‑6 : giao diện game khi chưa có đồ họa 19](#_Toc58237526)

[Hình ảnh 0‑7: hình ảnh giao diện game khi đã có đồ họa 20](#_Toc58237527)

[Hình ảnh 0‑8 : Thêm được nhân vật vào game 20](#_Toc58237528)

[Hình ảnh 0‑9 : khi vào game người chơi sẽ được đọc cốt truyện 21](#_Toc58237529)

[Hình ảnh 0‑10: Tạo được quái và quái có thể bắn nhưng chưa gây sát thương 21](#_Toc58237530)

[Hình ảnh 0‑11: Tạo được mạng và máu cho nhân vật và quái bắn gây sát thương 22](#_Toc58237531)

[Hình ảnh 0‑12 : quái có thể di chuyển 22](#_Toc58237532)

[Hình ảnh 0‑13: Tạo được màn cuối cùng đánh nhau với boss 23](#_Toc58237533)

[Hình ảnh 0‑14 : sát thương của boss rất lớn và di chuyển phức tạp 24](#_Toc58237534)

[Hình ảnh 0‑15 : khi bạn bị hết mạng 24](#_Toc58237535)

[Hình ảnh 0‑16 : khi bạn đánh đươc hết các con quái và boss 25](#_Toc58237536)

[Hình ảnh 6‑1 : máu và mạng của nhân vật 26](#_Toc58237537)

[Hình ảnh 0‑1: đoạn code tối ưu hóa game 27](#_Toc58237538)

# CHƯƠNG I: KHẢO SÁT VÀ THU THẬP YÊU CẦU

* 1. **Thông tin cá nhân:**
* Họ tên sinh viên: Nguyễn Phi Hùng
* Lớp: D18PM01
* MSSV: 1824801030033
  1. **Các công nghệ sử dụng trong đề tài:**
* Phần mềm code Eclipse
* Ngôn Ngữ Java cơ bản
* Ngôn Ngữ Java nâng cao
* Phần mềm Paint
* Phần mềm notepad
  1. **Khảo sát hiện trạng:**

**Mô tả Game:**

* Tạo ra game nhập vai , người chơi có thể điểu khiển nhân vật quay trái quay phải , chạy , bay , nhảy ,
* Tăng độ khó cho game bằng cách tạo ra các con quái có thể bắn ra những viên đạn và những viên đạn này bay trúng con nhân vật của mình thì con nhân vật sẽ bị mất máu , nếu số mạng về 0 thì dừng game và hiện ra dòng dữ Game Over và đưa người chơi về giao diện chờ

**Mô tả yêu cầu**

* Thiết kế giao diện :
* Thiết kế giao diện khi vào game : giao diện phải thân thiện với người chơi
* Thiết kế giao diện game : giao diện game phải sinh động , hấp dẫn gây sự thu hút đối với người chơi
* Tạo nhân vật:
* Tạo nhân vật 2D : tạo các thao tác di chuyển như : quay trái quay phải , bắn , bay , ngồi , chạy
* Tạo các trải nghiệm cho game :
* Thêm các con quái có thể bắn đạn chúng nhân vật
* Chia màn cho game

**Các chức năng của game:**

* Chức năng của con nhân vật :
* Bay bằng nút space
* Quay trái khi người chơi bấm
* Quay phải khi người chơi bấm
* Ngồi khi người chơi bấm
* Bắn khi người chơi bấm **A**
* Chức năng của các con quái trong game
* Quái sẽ được mặc định di chuyển qua các tọa độ đã được lập trình sẵn và sẽ tự động bắn ra các viên đạn

# CHƯƠNG 2: TÌM HIỂU CÁC CÔNG NGHỆ

## Java cơ bản:

1. **Giới thiệu về ngôn ngữ java**

Java là ngôn ngữ lập trình máy tính có tính chất hướng đối tượng, dựa trên các lớp, thường được sử dụng cho các hệ thống có tính độc lập cao. Nó được sử dụng để hướng tới các lập trình viên viết ứng dụng "write one, run everywhere" (viết một lần, chạy mọi nơi, nghĩa là đoạn code Java sau khi được biên dịch có thể chạy được trên tất cả các nền tảng hỗ trợ Java mà không cần phải được biên dịch lại. Các ứng dụng Java sau khi đã được biên dịch thành bytecode có thể chạy trên bất kỳ máy ảo Java nào (Java virtual machine)

Cho đến năm 2018, Java là một trong những ngôn ngữ được dùng phổ biến nhất trên thế giới, đặc biệt cho các úng dựng web client- server. Theo thống kê trên thế giới có khoảng 9 triệu lập trình viên Java

1. **Tổng quan về Java**

* Bạn có thể hiểu rằng, đặc điểm đơn giản của Java là do:
* Thứ nhất, cú pháp của nó dựa trên C++ (vì thế nó khá dễ dàng cho các sinh viên sau khi đã học C++ tại năm đầu đại học).
* Thứ hai, gõ bỏ nhiều đặc điểm gây bối rối và hiếm khi được sử dụng chẳng hạn như các con trỏ tường minh, nạp chồng toán tử, …
* Và cuối cùng, bạn không cần xóa các đối tượng mà không được tham chiếu, bởi vì hãy để những thứ đó cho Bộ dọn rác tự động (Garbage Collection) trong Java.

1. **Đặc điểm của ngôn ngữ lập trình java**

* Tương tự như C++ , hướng đối tượng hoàn toàn
* Độc lập phần cứng và hệ điều hành
* Ngôn ngữ thông dịch
* Cơ chế thu gom rác tự động
* Đa luồng
* Tính an toàn và bảo mật

1. **Phân tích chi tiết chương trình java đầu tiên**

* Dưới đây, là ý nghĩa của class, public, static, void, main, String[], System.out.println().
* Từ khóa **class** được sử dụng để khai báo một lớp trong Java.
* Từ khóa **public** là một Access Modifier mà biểu diễn tính nhìn thấy, nghĩa rằng nó là nhìn nhất với tất cả.
* **static** là một từ khóa, mà nếu chúng ta khai báo bất cứ phương thức nào là static thì nó còn được gọi là phương thức tĩnh hoặc phương thức static. Lợi thế chủ yếu của phương thức static là không cần thiết tạo đối tượng để triệu hồi phương thức static. Phương thức main được thực thi bởi JVM, vì thế bạn không cần thiết tạo một đối tượng để gọi phương thức main. Việc này giúp tiết kiệm bộ nhớ.
* **void** là kiểu trả về của phương thức, nghĩa là phương thức không trả về bất cứ giá trị nào.
* **main** tượng trưng cho khởi động chương trình.
* **String[] args** được sử dụng cho tham số dòng lệnh. Bạn sẽ tìm hiểu về chúng sau.
* **System.out.println()** được sử dụng như là lệnh in. Chương sau, bạn sẽ thấy cách làm việc nội tại của lệnh System.out.println này.

Để viết một chương trình đơn giản, bạn mở notepad bởi **Start menu** -> **All Programs** -> **Accessories** -> **notepad** và viết chương trình đơn giản Hello World trên.

Viết và lưu chương trình dưới dạng Simple.java. Để biên dịch và chạy chương trình này, bạn cần mở dòng nhắc lệnh command prompt bởi **Start menu** -> **All Programs** -> **Accessories** -> **command prompt**.

* Để biên dịch, gõ: javac Simple.java
* Để thực thi, gõ: java Simple

1. **Java nâng cao:**
2. **Khái niệm hướng đối tượng opp trong java**

* **[Khái niệm hướng đối tượng OOP trong Java]**Chương này, chúng ta sẽ tìm hiểu các khái niệm cơ bản về Hướng đối tượng (OOP). Lập trình hướng đối tượng bao gồm nhiều khái niệm như tính kế thừa, gắn kết dữ liệu (Data Binding), tính đa hình, …
* **Simula** được xem như là ngôn ngữ lập trình hướng đối tượng đầu tiên. Là một hệ lập trình mà ở đó mọi thứ được biểu diễn dưới dạng một đối tượng, và được biết đến như là ngôn ngữ hướng đối tượng thực sự. Tuy nhiên, **Smalltalk** mới được xem như là ngôn ngữ lập trình hướng đối tượng thực sự đầu tiên.

1. **Lập trình hướng đối tượng**

* **Object** (đối tượng) nghĩa là một thực thể trong thế giới thực, chẳng hạn như bàn, quả bóng, con bò, … Lập trình hướng đối tượng là một phương pháp để thiết kế một chương trình bởi sử dụng các lớp và các đối tượng. Nó làm đơn giản hóa việc duy trì và phát triển phần mềm bằng việc cung cấp một số khái niệm:
* **Đối tượng**: Một thực thể có trạng thái và hành vi. Ví dụ như xe đạp, bàn, ghế, … Nó có thể mang tính vật lý hoặc logic.
* **Lớp**: Một tập hợp các đối tượng. Nó là một thực thể logic.
* **Tính kế thừa**: Khi một đối tượng đạt được các thuộc tính và các hành vi của đối tượng cha, thì đó là tính kế thừa. Điều này làm tăng tính tái sử dụng cho code. Nó được sử dụng để đạt được tính đa hình tại runtime.
* **Tính đa hình**: Khi một tác vụ được thực hiện theo nhiều cách khác nhau được gọi là tính đa hình. Ví dụ: như vẽ hình chữ nhật hoặc hình tam giác, … Trong Java, chúng ta sử dụng nạp chồng phương thức (method overloading) và ghi đè phương thức (method overriding) để có tính đa hình. Một ví dụ khác: con mèo kêu meooo, còn chú chó thì sủa goooo.
* **Tính trừu tượng**: Đó là ẩn các chi tiết nội tại và hiển thị tính năng. Ví dụ, với cuộc gọi điện thoại, chúng ta không biết tiến trình xử lý nội tại là như thế nào. Trong Java, chúng là sử dụng lớp abstract và abstract interface để có tính trừu tượng.
* **Tính bao đóng**: Đó là gắn kết code và dữ liệu cùng với nhau vào trong một đơn vị unit đơn. Ví dụ: có thể bạn đã biết đến viên thuốc con nhộng (hay đơn giản hơn là gói bột giặt), các viên thuốc (hạt bột giặt) khác nhau được đóng gói.
* Một lớp Java là một ví dụ về tính bao đóng. Java Bean là lớp bao đóng thực sự, vì tất cả thành viên dữ liệu là private.
* Lợi thế của OOP khi so với ngôn ngữ lập trình hướng thủ tục (procedure-oriented)
* OOP giúp việc thiết kế, phát triển và bảo trì dễ dàng hơn trong khi với lập trình hướng thủ tục thì việc quản lý code là khá khó khăn nếu lượng code tăng lên. Điều này làm tăng hiệu quả có quá trình phát triển phần mềm.
* OOP cung cấp Data Hiding (ẩn dữ liệu) trong khi đó trong hướng thủ tục một dữ liệu toàn cục có thể được truy cập từ bất cứ đâu.
* OOP cung cấp cho bạn khả năng để mô phỏng các sự kiện trong thế giới thực một cách hiệu quả hơn. Chúng ta có thể cung cấp giải pháp cho các vấn đề trong thế giới thực nếu chúng ta sử dụng Lập trình hướng đối tượng.

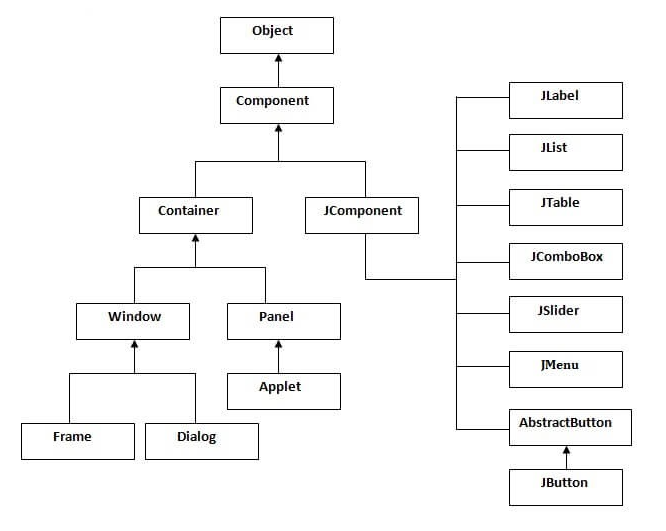
1. **Java swing :**
2. **Giới thiệu về Java Swing**

Java Swing là một phần của Java Foundation Classes (JFC) được sử dụng để tạo các ứng dụng window-based. Nó được xây dựng trên API AWT (Abstract Windowing Toolkit) và được viết hoàn toàn bằng Java.

Không giống như AWT, Java Swing cung cấp các thành phần không phụ thuộc vào nền tảng và nhẹ hơn.

Gói javax.swing cung cấp các lớp cho java swing API như JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser, v.v.

1. **Phân cấp các lớp Java Swing**



1. **Các phương thức thường dùng của lớp Component**

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| public void add(Component c) | thêm một thành phần vào thành phần khác. |
| public void setSize(int width, int height) | thiết lập kích thước của thành phần. |
| public void setLayout(LayoutManager m) | thiết lập trình quản lý bố cục (layout) cho thành phần. |
| public void setVisible(boolean b) | thiết lập khả năng hiển thị của thành phần. Nó theo mặc định là false (ẩn) |

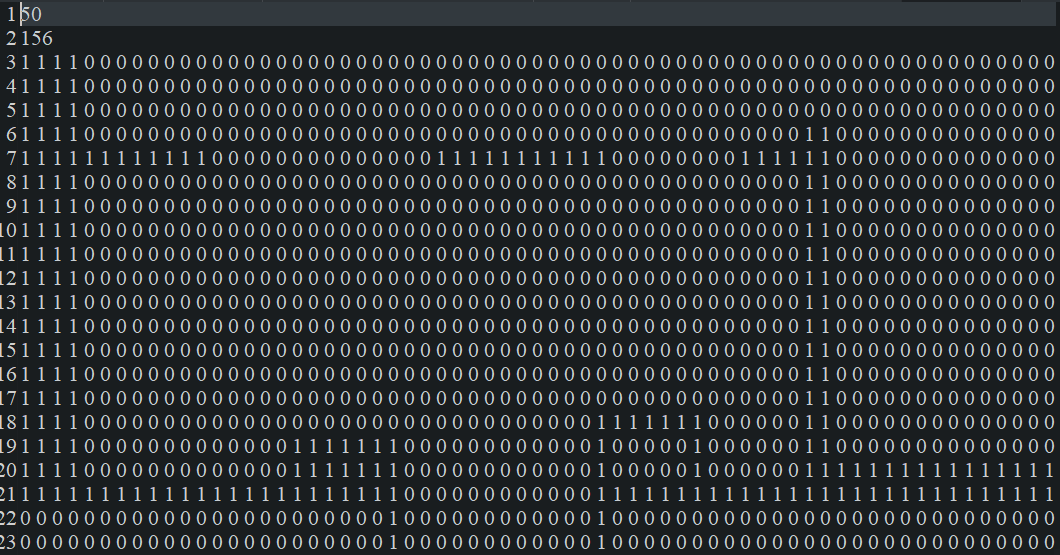
# CHƯƠNG 3: XÂY DỰNG GAME

1. **Các phần mềm , ngôn ngữ sử dụng trong game**

* Phần mềm code Eclipse
* Sử dụng thư viện java swing
* Ngôn ngữ java

1. **Dữ liệu trong game**

* Dữ liệu map trong game được chuyển thành mảng 2 chiểu để đưa vào game



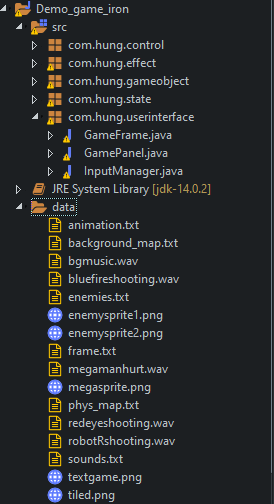
Hình ảnh 0‑1: hình ảnh đồ họa 2D của map được đổi thành mảng 2 chiều

* Hình ảnh các thao tác của nhân vật
  + Nhân vật chạy
  + Nhân vật nháy mắt
  + Nhân vật bị thương
  + Trái tim thể hiện mạng của nhân vật
  + Viên đạn của nhân vật bắn ra và của quái bắn ra
  + Nhân vật mới hồi sinh
  + Nhân vật ngồi
  + Nhân vật bắn
  + Avatar của cốt truyện



Hình ảnh 0‑2: hình ảnh các thao tác của nhân vật và các viên đạn

* Tất cả dữ liệu của game đều được đưa vào folder data và để trong java project



Hình ảnh 0‑3 : các file dữ liệu của game được lưu trong data trong project

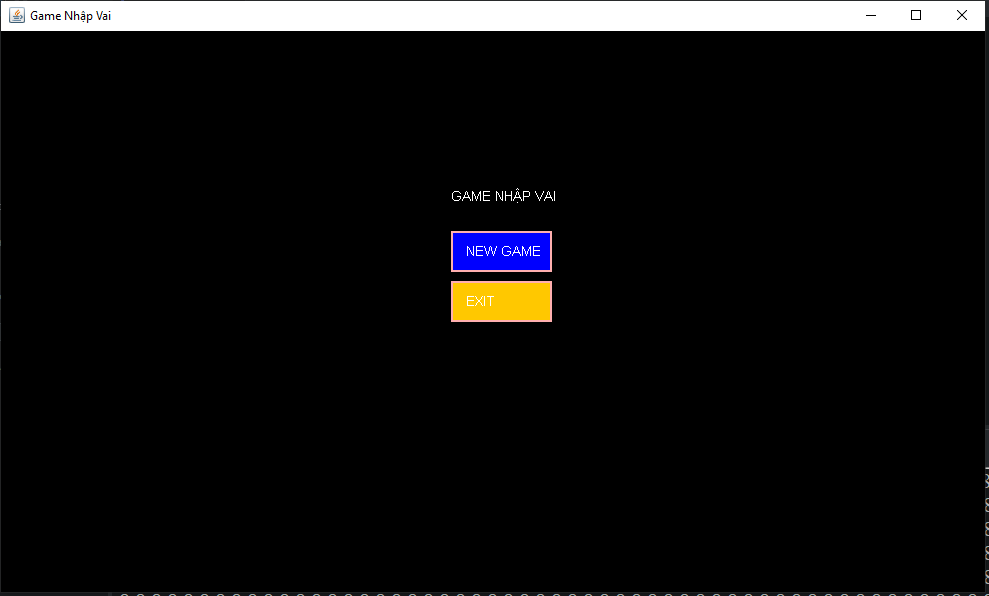
* Hình ảnh các thao tác của con boss và 1 số con quái nhỏ



Hình ảnh 0‑4 : hình ảnh các thao tác của boss trong game

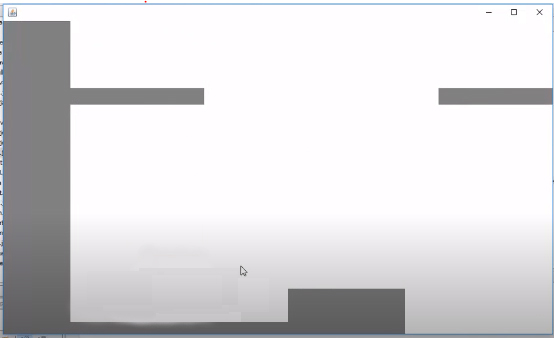
1. **Thiết kế giao diện Game Nhập Vai**

* Giao diện chờ
  + Người chơi bấm phím up để lên , down để xuống . Di chuyển lên xuống để chọn chức năng :
    - New Game
    - Thoát
  + Người chơi bấm phím Enter để chọn chức năng

****

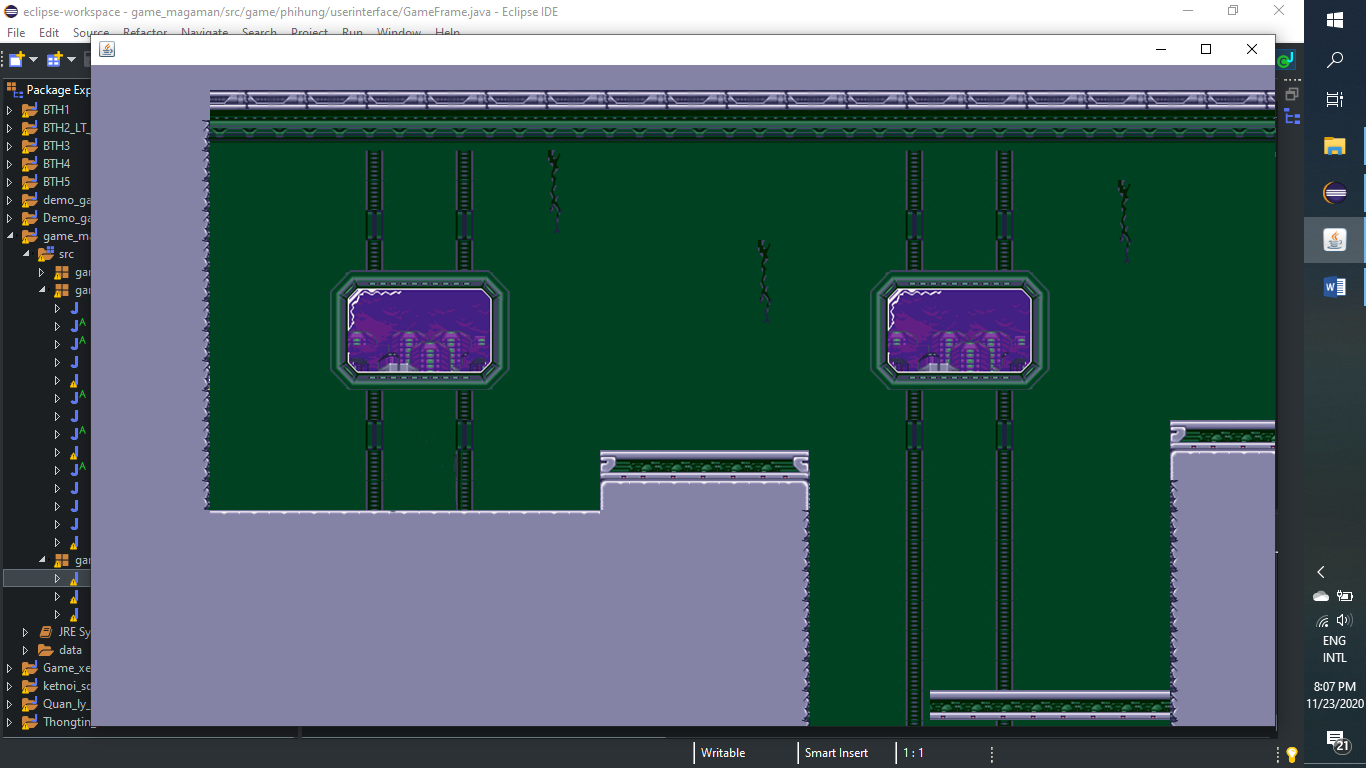
Hình ảnh 0‑5 : Giao diện chờ để vào game

* Thiết kế giao diện game là để tạo các bức tường và các chỗ đứng cho nhân vật
* Tạo map cho nhân vật có thể di chuyển được nhiều nơi hơn

****

Hình ảnh 0‑6 : giao diện game khi chưa có đồ họa

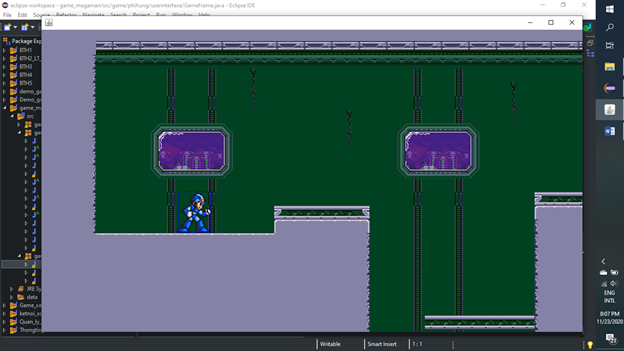
* Giao diện khi đã được thêm đồ họa vào . Thêm đồ họa vào để cho game sinh động hơn



Hình ảnh 0‑7: hình ảnh giao diện game khi đã có đồ họa

1. **Tạo nhân vật trong Game Nhập Vai**

* Tạo ra nhân vật 2D : nhân vật có thể di chuyển do người chơi điều khiển
* Ngoài ra nhân vật khi đứng yên thì sẽ có trạng thái nhắm mở mắt

****

Hình ảnh 0‑8 : Thêm được nhân vật vào game

1. **Tạo các trải nghiệm cho Game Nhập Vai**

* Khi mới vào game thì sẽ hiện lên cốt truyện cho người chơi đọc và cũng là hướng dẫn người chơi biết được mục tiêu phải đánh bại

****

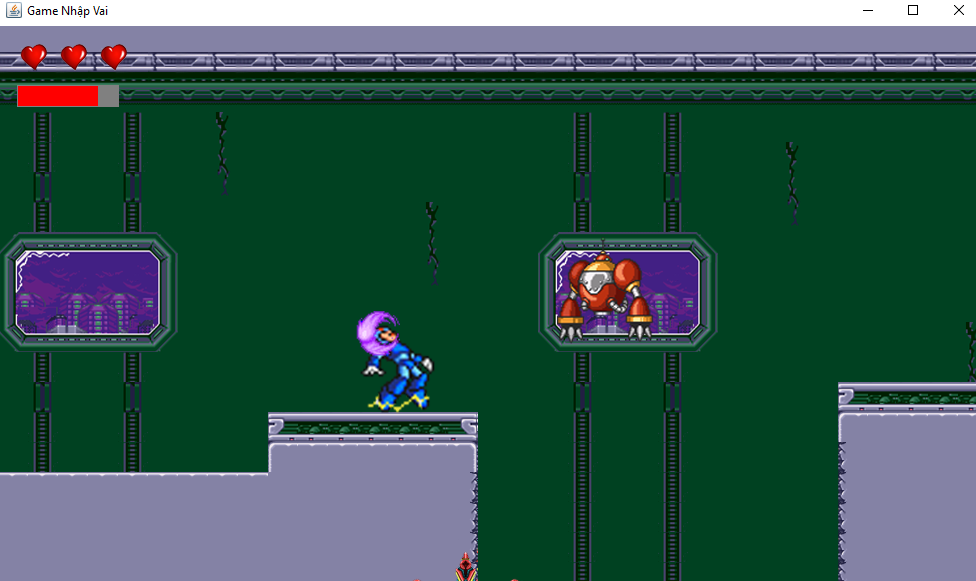
Hình ảnh 0‑9 : khi vào game người chơi sẽ được đọc cốt truyện

* Tạo thêm các con quái trong game tăng sự trải nghiệm cho người chơi
* Làm cho quái tự động di chuyển và bắn ra đạn để tạo độ khó cho game

****

Hình ảnh 0‑10: Tạo được quái và quái có thể bắn nhưng chưa gây sát thương

* Tạo cho nhân vật thêm mạng và máu để tăng độ hấp dẫn cho game và cho người chơi biết mình còn bao nhiêu máu và còn còn bai nhiêu mạng

****

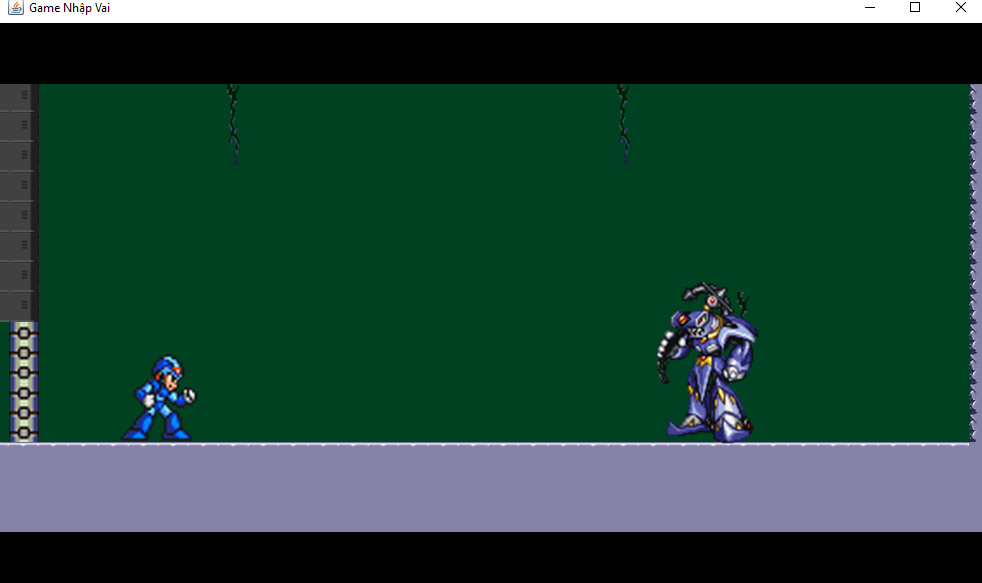
Hình ảnh 0‑11: Tạo được mạng và máu cho nhân vật và quái bắn gây sát thương

* khi người chơi để nhân vật của mình chạm vào quái thì nhân vật sẽ bị mất máu

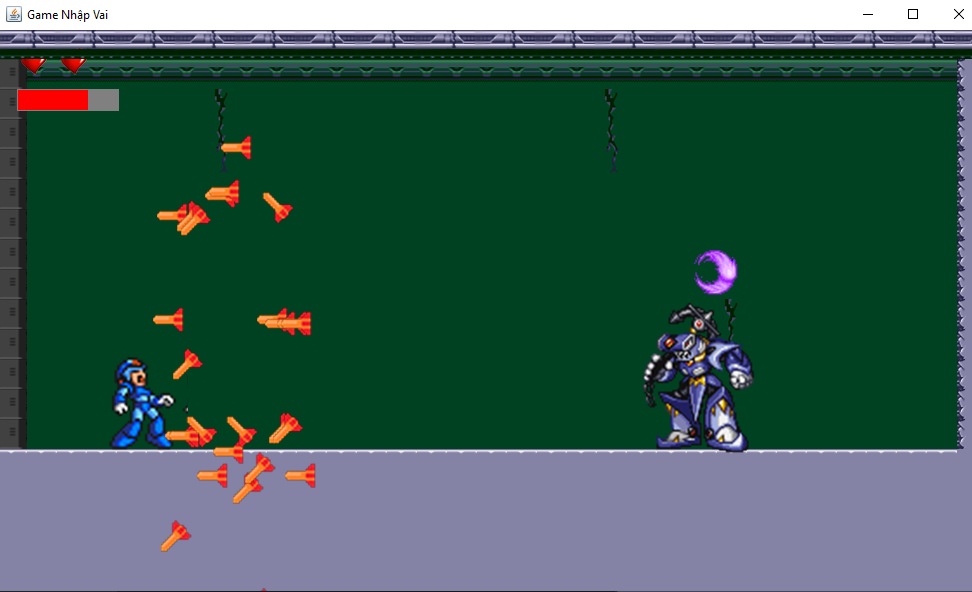
****

Hình ảnh 0‑12 : quái có thể di chuyển

* Tạo màn cho game sau khi đánh hết quái nhỏ thì người chơi sẽ đi qua cánh cửa và phải chiến đấu với quái boss
* Quái boss sẽ mạnh hơn rất nhiều so với các con quái khác và mức độ gây sát thương sẽ cao hơn.
* Mức độ khó khi đánh nhau với boss sẽ khó hơn rất nhiều so với đánh những con quái nhỏ

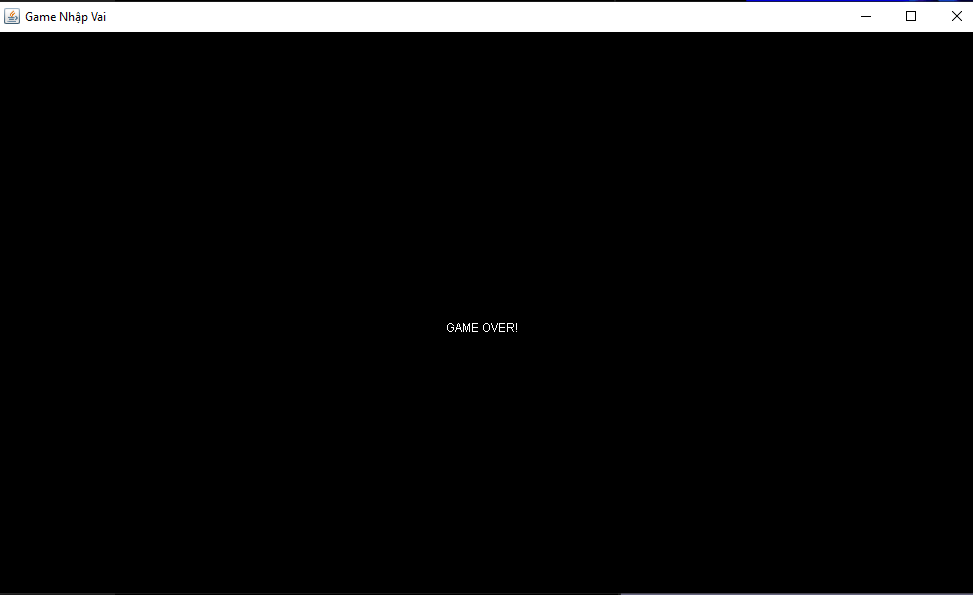
****

Hình ảnh 0‑13: Tạo được màn cuối cùng đánh nhau với boss

****

Hình ảnh 0‑14 : sát thương của boss rất lớn và di chuyển phức tạp

* Nếu chơi để nhân vật của mình bị chết thì game sẽ dừng lại và hiện lên từ Game Over như dưới
* Sau đó thì đưa người chơi trở về giao diện chờ

****

Hình ảnh 0‑15 : khi bạn bị hết mạng

* Nếu người chơi vượt qua hết các ải thì hệ thống sẽ hiện lên Game Win
* Và đưa người chơi trở về giao diện chờ

****

Hình ảnh 0‑16 : khi bạn đánh đươc hết các con quái và boss

# Hướng dẫn cách chơi game

* Bước 1: người chơi mở game lên
* Bước 2: khi vào giao diện chờ người chơi chọn chức năng : new game hoặc exit bằng nút lên xuống trên bàn phím sau đó nhấn enter để thực hiện chức năng
* Bước 3 : khi vào game người chơi sẽ được đọc cốt truyện của game và người chơi bấm enter để bỏ qua
* Bước 4 : người chơi điểu khiển nhân vật bằng các phím
  + Phím xuống : nhân vật sẽ ngồi xuống
  + Phím qua trái : nhân vật sẽ đi qua trái
  + Phím qua phải : nhân vật sẽ đi qua bên phải
  + Phím space : nhân vật sẽ nhảy lên
  + Phím A : nhân vật sẽ bắn ra các viên đạn
* Người chơi sẽ có tối đa là 3 mạng và mỗi mạng sẽ có 1 thanh máu nếu người chơi để hết 1 thanh máu nghĩa là sẽ mất 1 mạng và cứ như thế nếu mất hết ba mạng thì người chơi sẽ thua và phải chơi game lại từ đầu

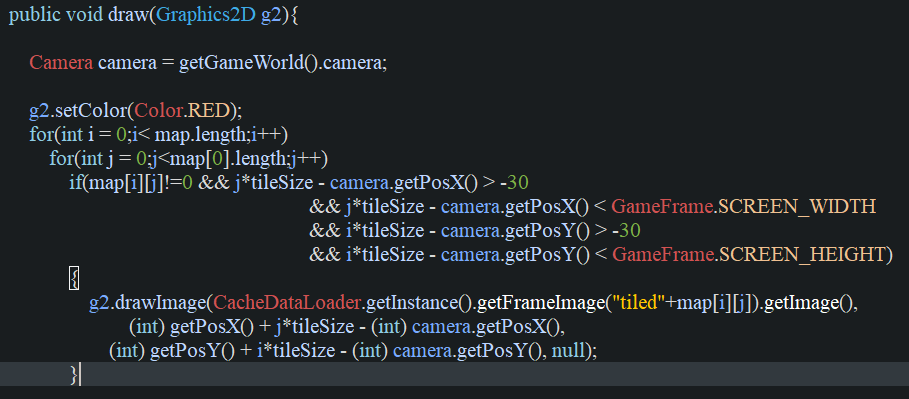


Hình ảnh 6‑1 : máu và mạng của nhân vật

# KẾT LUẬN

1. **Đánh giá kết quả** 
   1. **Kết quả đạt được:**

* Đã tạo được hoàn chỉnh một game nhập vai với đồ họa 2D
* Đã tối ưu hóa được được code làm cho game chạy mượt mà hơn



Hình ảnh 0‑1: đoạn code tối ưu hóa game

* 1. **Hạn chế của đề tài:**
* Vẫn chưa thêm được nhiều Map cho người chơi trải nghiệm
* Game chưa được mượt mà
* Giao diện chờ còn sơ xài chưa bắt mắt người dùng

1. **Hướng phát triển của đề tài**

* Thêm các chức năng trong giao diện chờ
  + - Tăng chỉnh âm lượng
    - Cài đặt phím
    - Hỗ trợ người chơi
* Tạo thêm nhiều Map hơn cho người chơi trải nghiệm
* Tạo thêm nhiều quái và quái thông minh hơn để tăng độ hấp dẫn và đô khó cho game

# TÀI LIỆU KHAM KHẢO

1. <https://viettuts.vn/java-swing>
2. <http://hiepsiit.com/detail/java/javacore>
3. <https://giasutinhoc.vn/libgdx/lap-trinh-game-voi-java-bai-2/>
4. Phạm Văn Trung, Phạm Văn Tho, Bùi Công Thành, Phạm Thị Minh Lượng (2018). Lập trình Java cơ bản. Nhà xuất bản Xây dựng.
5. [Lassoff, Mark (2017). *Java programming for beginners: learn the fundamentals of programming with Java.*PACKT Publishing.](http://viewer.igroupnet.com/viewer/service/iglib3/default/PACKT0000676/default/default/desktop/vertical/0/1/viewer.html?1effea9cffa4f434c2aad30e2e08491f)
6. [*http://elib.tdmu.edu.vn/tailieuvn/doc/bai-thuc-hanh-lap-trinh-java-1-bai-2-kieu-toan-tu-lenh-if-va-switch-2154441.html*](http://elib.tdmu.edu.vn/tailieuvn/doc/bai-thuc-hanh-lap-trinh-java-1-bai-2-kieu-toan-tu-lenh-if-va-switch-2154441.html)