

**TRƯỜNG ĐẠI HỌC THÔNG TIN LIÊN LẠC
KHOA CÔNG NGHỆ THÔNG TIN**



BÀI TẬP LỚN

**XÂY DỰNG GAME JAVA DRAGON TALE TUTORIAL
MÔN : LẬP TRÌNH TRÒ CHƠI VÀ MÔ PHỎNG**

Giảng viên hướng dẫn : Nguyễn Việt Hùng
Sinh viên thực hiện : Trần Ngọc Nam (NT)
Lê Cao Diễm Quyên
Nguyễn Trung Hiếu

Khóa : 2014-2018
Lớp : Công Nghệ Phần Mềm

Khánh hòa, Tháng 01 Năm 2018

**TRƯỜNG ĐẠI HỌC THÔNG TIN LIÊN LẠC
KHOA CÔNG NGHỆ THÔNG TIN**



BÀI TẬP LỚN

**XÂY DỰNG GAME JAVA DRAGON TALE TUTORIAL
MÔN : LẬP TRÌNH TRÒ CHƠI VÀ MÔ PHỎNG**

Giảng viên hướng dẫn : Nguyễn Việt Hùng

Sinh viên thực hiện :

- 1. Trần Ngọc Nam (NT)**
- 2. Lê Cao Diễm Quyên**
- 3. Nguyễn Trung Hiếu**

Khóa : 2014-2018

Lớp : Công Nghệ Phần Mềm

Khánh hòa, Tháng 01 Năm 2018

[illegible]

Mục Lục

LỜI CẢM ƠN	1
LỜI MỞ ĐẦU	2
1. Lý do chọn đề tài.....	2
2. Mục tiêu của đề tài.....	2
3. Phạm vi nghiên cứu	2
4. Công nghệ sử dụng	2
5. Thông tin thành viên trong nhóm	3
CHƯƠNG 1 : TỔNG QUAN.....	4
1.1. Giới thiệu về ngôn ngữ Java	4
1.2. Tổng quan về lập trình game.....	6
CHƯƠNG 2 : LẬP TRÌNH VỀ GAME	8
2.1. Khái quát về game Dragon Tale Tutorial	8
2.2. Kịch bản game	8
2.3. Luật chơi.....	8
2.4. Hình ảnh nhân vật thiết kế	9
2.4.1. Hình ảnh nhân vật dragon và đạn	9
2.4.2. Hình ảnh chương ngại vật	9
2.4.3. Hình ảnh nhân vật bị trúng đạn	10
2.3.4. Hình ảnh về thiết kế đường đi	10
2.3.5. Hình ảnh background.....	10
CHƯƠNG 3 : KẾT QUẢ.....	13
3.1. Giao diện của game	13
3.1.1. Giao diện bắt đầu game	13
3.1.2. Giao diện game level 1	13
3.1.3. Giao diện game level 2	14
3.1.4. Giao diện game level 3	14
3.1.5. Giao diện game level 4	15
3.1.6. Giao diện game level 5	15
3.1.7. Giao diện game khi chết	16

3.2. Một số code chính của chương trình	16
KẾT LUẬN	19
TÀI LIỆU THAM KHẢO	20

DANH MỤC HÌNH ẢNH

Hình 1 : Nhân vật và đạn của game

Hình 2 : Chương ngại vật của game

Hình 3: Chương ngại vật khi trúng đạn

Hình 4 : Một số chi tiết khung cảnh

Hình 5 : Hình nền của game

Hình 6 : Giao diện màn hình game bắt đầu

Hình 7 : Giao diện màn hình game level 1

Hình 8 : Giao diện màn hình game level 2

Hình 9 : Giao diện màn hình game level 3

Hình 10 : Giao diện màn hình game level 4

Hình 11 : Giao diện màn hình game level 5

Hình 12 : Giao diện màn hình game khi bị chết

LỜI CẢM ƠN

Trước tiên nhóm em xin gửi lời cảm ơn chân thành sâu sắc tới thầy cô giáo trong trường Đại Học Thông Tin Liên Lạc nói chung và thầy cô giáo trong khoa Công Nghệ Thông Tin nói riêng đã tận tình giảng dạy, truyền đạt cho em những kiến thức, kinh nghiệm quý báu trong suốt thời gian qua.

Đặc biệt nhóm em xin gửi lời cảm ơn đến thầy Nguyễn Việt Hùng đã tận tình giúp đỡ, trực tiếp chỉ bảo, hướng dẫn nhóm em trong suốt quá trình làm đề tài. Trong thời gian học tập nghiên cứu với thầy, nhóm em không ngừng tiếp thu thêm những kiến thức bổ ích.

LỜI MỞ ĐẦU

1. Lý do chọn đề tài

Công nghệ thông tin ngày càng có vai trò quan trọng trong cuộc sống hằng ngày của chúng ta. Việc ứng dụng công nghệ thông tin vào các lĩnh vực trong đời sống giúp công việc được tiến hành nhanh chóng và hiệu quả hơn. Có rất nhiều công việc mới phát triển song song với sự phát triển của công nghệ thông tin, một trong số đó là phát triển game.

Chúng em chọn đề tài “Lập trình game 2D Dragon Tale Tutorial trên nền tảng Java” nhằm hiểu sâu hơn về ngôn ngữ lập trình java, từ đó viết một ứng dụng cụ thể thực nghiệm làm cơ sở củng cố kiến thức và định hướng, kế hoạch xây dựng những ứng dụng game cụ thể, phát triển theo hướng dịch vụ tương lai.

2. Mục tiêu của đề tài

Mục tiêu của đề tài nhằm nghiên cứu ngôn ngữ lập trình Java và xây dựng game đơn giản để hiểu rõ ngôn ngữ Java.

3. Phạm vi nghiên cứu

Nghiên cứu môi trường phát triển, ngôn ngữ Java trong lập trình hướng đối tượng. Tìm hiểu phương thức và cách làm game đơn giản bằng ngôn ngữ Java.

4. Công nghệ sử dụng

Trong xu thế phát triển công nghệ thông tin ngày càng lớn mạnh như hiện nay, đặc biệt là ngành công nghệ phần mềm, ngày càng đòi hỏi trình độ cao trong kỹ thuật lập trình. Chính vì vậy mà phương pháp lập trình hướng thủ tục cổ điển trước đây không đáp ứng được nhu cầu đặt ra của thời đại, một phương pháp lập trình mới được xây dựng theo nguyên lý Alan-Kay đã được

ra đời nhằm đáp ứng những nhu cầu cấp thiết đó : “ Phương pháp lập trình hướng đối tượng ”.

Đề tài này được thiết kế theo phương pháp lập trình hướng đối tượng bằng ngôn ngữ Java, do SunMicroSystem đưa ra vào năm 1991. Chính vì vậy mà nó giải quyết được những vướng mắc gặp phải khi thiết kế theo phương pháp lập trình thủ tục thuần túy.

- Mã chương trình rõ ràng, dễ đọc, dễ hiểu và cô đọng
- Chương trình được tổ chức thành những Class lắp ghép lại với nhau thành một khối thống nhất
- Mỗi Class gồm có nhiều Method đảm nhận các vai trò khác nhau trong chương trình.
- Chương trình có tính mềm dẻo cao
- Có khả năng tái sử dụng tài nguyên

5. Thông tin thành viên trong nhóm

ST T	Mã sinh viên	Họ tên	Email
1	14DC050	Trần Ngọc Nam	ngocnam071094@gmail.com
2	14DC161	Lê Cao Diễm Quyên	diemquyen2596dhcn1b@gmail.com
3	14DC123	Nguyễn Trung Hiếu	nguyentrunghieutcu@gmail.com

CHƯƠNG 1 : TỔNG QUAN

1.1. Giới thiệu về ngôn ngữ Java

Ngôn ngữ lập trình Java ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995 (Java 1.0 [J2SE]). Tính đến thời điểm này (tháng 2/2015) phiên bản mới nhất của Java Standard Edition (JSE) là 8. Với ưu thế về đa nền tảng (multi platform) Java càng lúc càng được ứng dụng rộng rãi trên nhiều thiết bị từ máy tính đến mobile và nhiều thiết bị phần cứng khác...

Java là ngôn ngữ lập trình hướng đối tượng nên nó cũng có 4 đặc điểm chung của các ngôn ngữ hướng đối tượng.

- **Tính trừu tượng (Abstraction):** là tiến trình xác định và nhóm các thuộc tính, các hành động liên quan đến một thực thể đặc thù, xét trong mối tương quan với ứng dụng đang phát triển.
- **Tính đa hình (Polymorphism):** cho phép một phương thức có các tác động khác nhau trên nhiều loại đối tượng khác nhau. Với tính đa hình, nếu cùng một phương thức ứng dụng cho các đối tượng thuộc các lớp khác nhau thì nó đưa đến những kết quả khác nhau. Bản chất của sự việc chính là phương thức này bao gồm cùng một số lượng các tham số.
- **Tính kế thừa (Inheritance):** Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.
- **Tính đóng gói (Encapsulation):** là tiến trình che giấu việc thực thi những chi tiết của một đối tượng đối với người sử dụng đối tượng ấy.

Bên cạnh đó Java còn có một số đặc tính khác:

- Độc lập nền (Write Once, Run Anywhere): Không giống như nhiều ngôn ngữ lập trình khác như C và C ++, khi Java được biên dịch, nó không được biên dịch sang mã máy cụ thể, mà thay vào đó là mã byte code chạy trên máy ảo Java (JVM). Điều này đồng nghĩa với việc bất cứ thiết bị nào có cài đặt JVM sẽ có thể thực thi được các chương trình Java.
- Đơn giản: [học Java](#) thật sự dễ hơn nhiều so với C/C++, nếu bạn đã quen với các ngôn ngữ lập trình hướng đối tượng thì việc học Java sẽ dễ dàng hơn. Java trở nên đơn giản hơn so với C/C++ do đã loại bỏ tính đa kế thừa và phép toán con trỏ từ C/C++.
- Bảo mật: Java hỗ trợ bảo mật rất tốt bởi các thuật toán mã hóa như mã hóa một chiều (one way hashing) hoặc mã hóa công cộng (public key)...
- Đa luồng: Với tính năng đa luồng Java có thể viết chương trình có thể thực thi nhiều task cùng một lúc. Tính năng này thường được sử dụng rất nhiều trong lập trình game.
- Hiệu suất cao nhờ vào trình thu gom rác (garbage collection), giải phóng bộ nhớ đối với các đối tượng không được dùng đến.
- Linh hoạt: Java được xem là linh hoạt hơn C/C ++ vì nó được thiết kế để thích ứng với nhiều môi trường phát triển.

1.2. Tổng quan về lập trình game

Tổng quan về lập trình game sẽ trình bày đến bạn đọc những kiến thức cần biết trước khi bắt đầu tìm hiểu kỹ thuật lập trình game.

Các bạn nên biết, khi phát triển một game nào đó thông thường có 2 cách tiếp cận sau: Một là bắt tay vào viết code ngay, hai là tạo tài liệu thiết kế bao gồm ý tưởng về kịch bản (cốt truyện), đồ hoạ, âm thanh, ...

Và trong một đội (team) phát triển game, tối thiểu phải có những vị trí như :

- 1/ Bộ phận viết kịch bản (cốt truyện)
- 2/ Bộ phận thiết kế đồ hoạ (thiết kế nhân vật, thiết kế bản đồ game, thiết kế vật phẩm,...)
- 3/ Bộ phận thiết kế âm thanh (nhạc nền, âm thanh tương tác, nhạc chuyển cảnh, ...)
- 4/ Bộ phận lập trình (Cài đặt code)
- 5/ Bộ phận kiểm thử (tester)

Quy trình phát triển game

- **Giai đoạn tiền sản phẩm/ý tưởng** bao gồm cốt truyện game, thể loại game (hành động, đối kháng, đánh trận,...), phong cách đồ hoạ và âm thanh.
- **Giai đoạn đặc tả lập trình** thể hiện giao diện game và sự liên kết giữa chúng
 - Màn hình chính, màn hình thiết lập, màn hình chơi game, màn hình hiển thị điểm số,...
 - Trên mỗi màn hình có nhiều thành phần: nút nhấn, hình ảnh,...
 - Màn hình phải cho phép người chơi tương tác: chọn New Game,...

- **Giai đoạn thiết kế** bao gồm Đồ họa (nhân vật chính, quân địch,...), Âm thanh, Cấp độ chơi
- Giai đoạn lập trình là giai đoạn dành cho các lập trình viên là những người phải sử dụng ngôn ngữ lập trình thích hợp để cài đặt những xử lý như:
 - Quản lý màn hình: đóng (closing), dừng (pausing), chơi lại (resuming) trong game
 - Xử lý đầu vào: xử lý các sự kiện như chạm màn hình, xoay màn hình,...
 - Xử lý đọc file: file âm thanh, file hình ảnh, file txt,...
 - Hiển thị đồ họa và phát âm thanh
 - Lựa chọn và sử dụng engine phù hợp (nếu có)

CHƯƠNG 2 : LẬP TRÌNH VỀ GAME

2.1. Khái quát về game Dragon Tale Tutorial

Ứng dụng Game dragon tale tutorial được xây dựng nhằm mục đích làm quen với ngôn ngữ java, các câu lệnh cơ bản và một số công nghệ hướng đối tượng tiên tiến.

Ứng dụng còn có mục đích chính nhằm giải trí nhẹ nhàng cho những ai đang căng thẳng.

2.2. Kịch bản game

Người chơi game nhấn nút bắt đầu để vào game và sử dụng các nút như:

- ❖ ← : qua trái
- ❖ → : qua phải
- ❖ W : nhảy
- ❖ E : bay
- ❖ R : chém
- ❖ F : bắn

Khi chơi sử dụng các phím để vượt qua các vật cản nhảy qua và khi gặp một số chướng ngại vật như ốc sên, nhện, boss thì sử dụng các tính năng như nhảy lên đầu, chém hoặc bắn để vượt qua về đích qua màn mới.

Khi bắn một số ốc sên, nhện hoặc boss sẽ có các vật phẩm như mạng, tăng số đạn. Số đạn được giới hạn khi bắn hết sẽ được tăng lên lại theo thời gian.

2.3. Luật chơi

Phải vượt qua các chướng ngại vật như các bậc cao và bắn hết chướng ngại vật như ốc sên, nhện, boss mới được qua màn còn không bắn hết thì không được qua màn.

Nếu nhảy bị rớt xuống nước, chạm vào vật cản như ốc sên, nhện, boss sẽ bị chết và sẽ xuất hiện lại vị trí đó khi đang còn mạng. Còn không còn mạng thì quay lại level đó.

2.4. Hình ảnh nhân vật thiết kế

2.4.1. Hình ảnh nhân vật dragon và đạn



Hình 1 : Nhân vật và đạn của game

2.4.2. Hình ảnh chương ngại vật



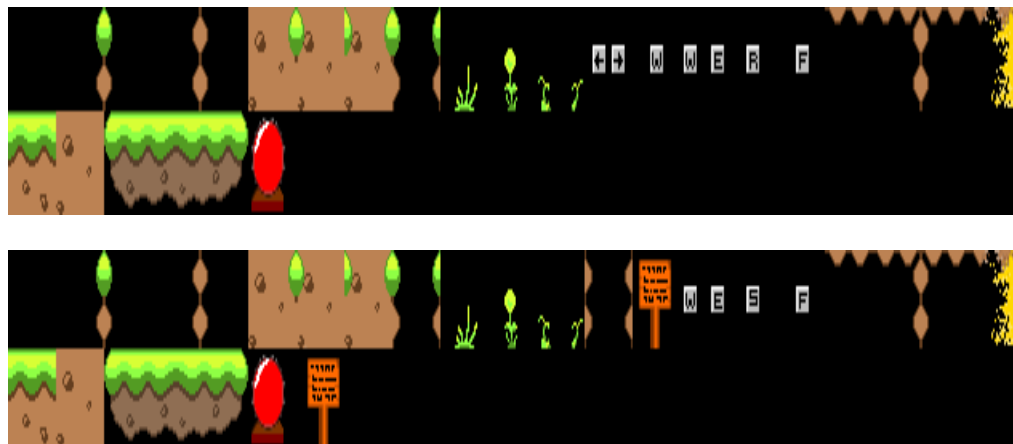
Hình 2 : Chương ngại vật của game

2.4.3. Hình ảnh nhân vật bị trúng đạn



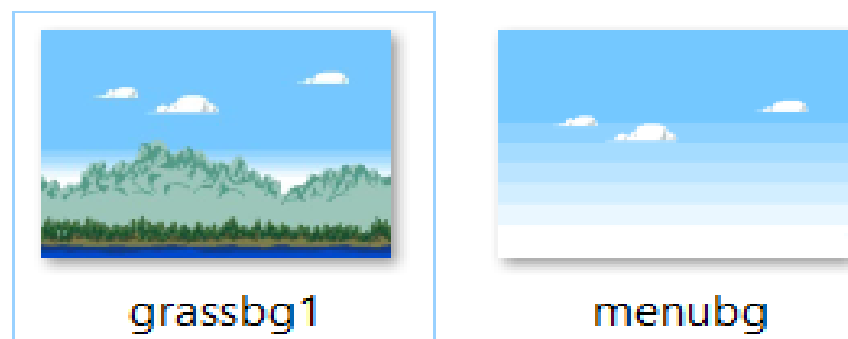
Hình 3: Chương ngại vật khi trúng đạn

2.3.4. Hình ảnh về thiết kế đường đi



Hình 4 : Một số chi tiết khung cảnh

2.3.5. Hình ảnh background



Hình 5 : Hình nền của game

2.5. Phân công công việc và kế hoạch làm game

Ngày	Công việc	Phân công
25/12/2017 đến 31/12/2017	Tìm hiểu một số thư viện, công cụ: Eclipse, netBeans	Nam, Quyên, Hiếu
	Kịch bản game	Nam, Quyên, Hiếu
	Tạo github lưu trữ code	Nam
	Thiết kế giao diện background	Quyên
	Thiết kế maps của 5 level	Quyên
	Thiết kế giao diện thể hiện số mạng, số đạn, level	Nam
	Thiết kế các nút bắt đầu, trợ giúp, thoát	Hiếu
	Test các chức năng xây dựng được	Quyên, Hiếu

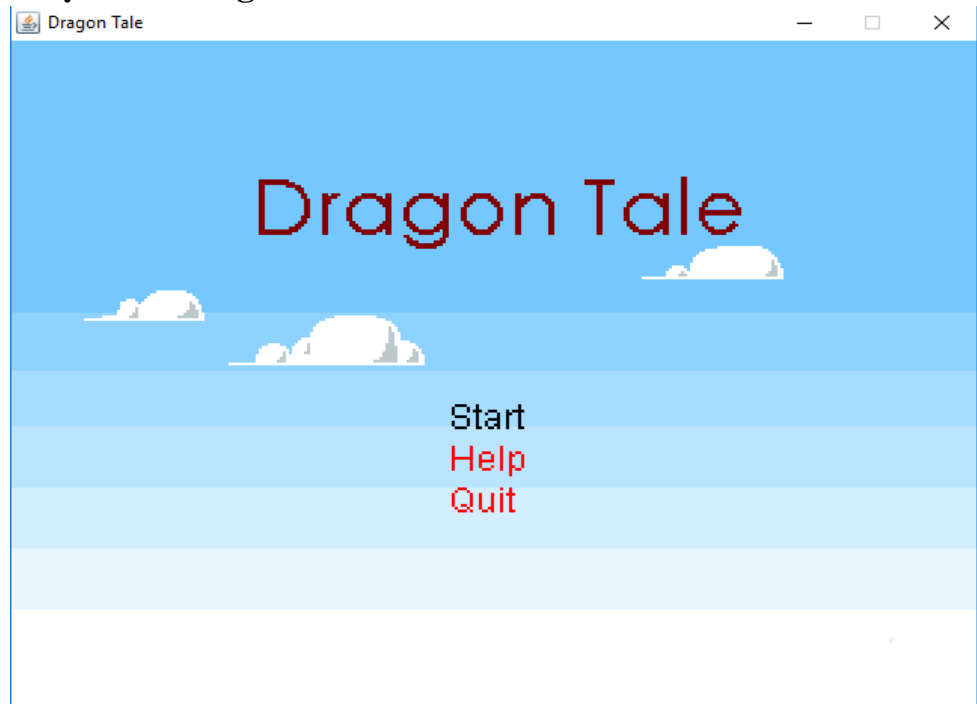
Ngày	Công việc	Phân công
1/1/2018 đến 7/1/2018	Thiết kế nhân vật: rồng, ốc sên, nhện, boss	Hiếu
	Thiết kế các chướng ngại vật	Nam
	Tạo luật chơi cho game	Nam
	Viết funtion điều khiển	Nam
	Thiết kế các nút gợi ý: ← : qua trái W : nhảy → : qua phải E : bay R : chém F : bắn	Hiếu
	Test các chức năng đã xây dựng được	Quyên

Ngày	Công việc	Phân công
8/1/2018 đến 14/1/2018	Chèn âm thanh	Hiếu
	Viết testcase	Quyên
	Fix lỗi	Nam
	Viết báo cáo	Nam
	Slide báo cáo	Hiếu

CHƯƠNG 3 : KẾT QUẢ

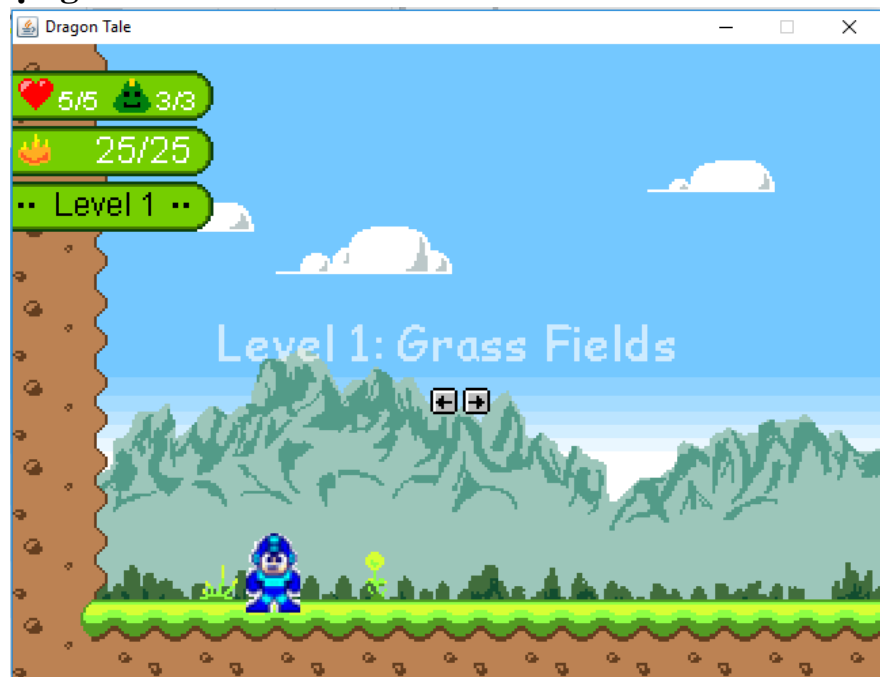
3.1. Giao diện của game

3.1.1. Giao diện bắt đầu game



Hình 6 : Giao diện màn hình game bắt đầu

3.1.2. Giao diện game level 1



Hình 7 : Giao diện màn hình game level 1

3.1.3. Giao diện game level 2



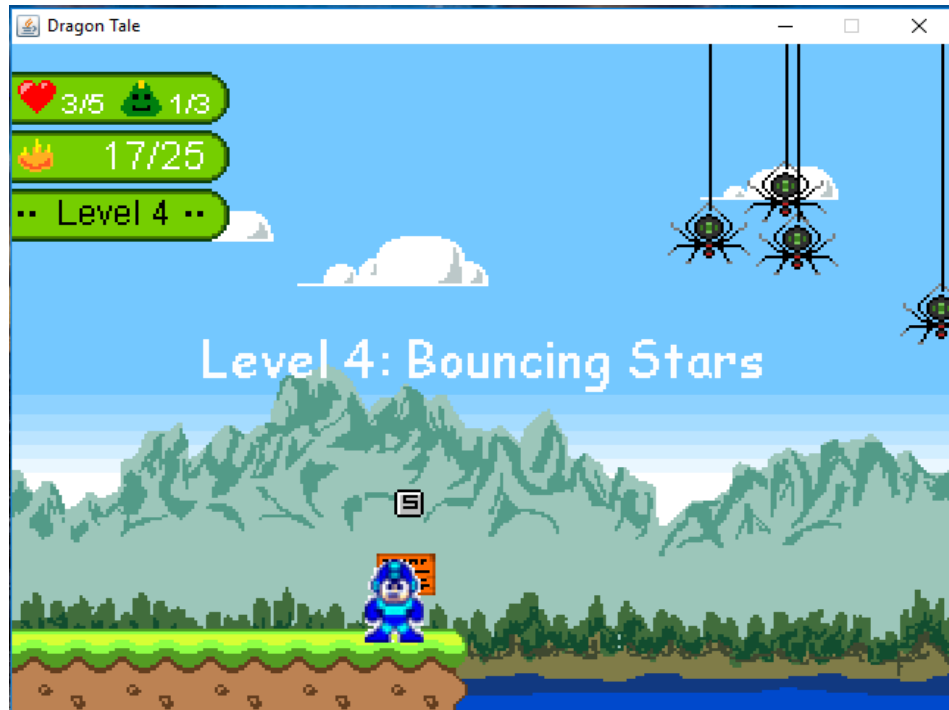
Hình 8 : Giao diện màn hình game level 2

3.1.4. Giao diện game level 3



Hình 9 : Giao diện màn hình game level 3

3.1.5. Giao diện game level 4



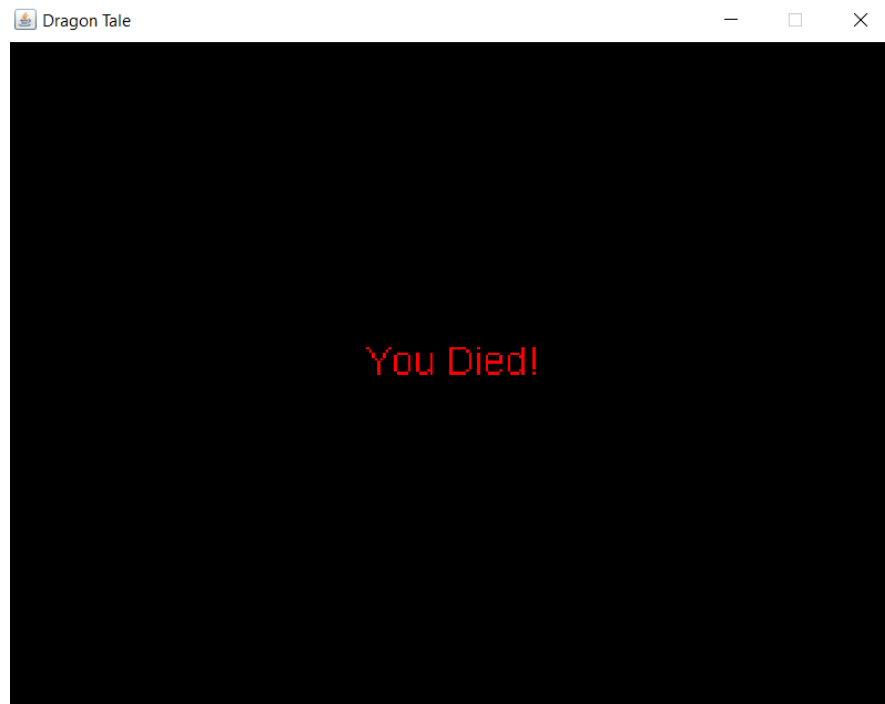
Hình 10 : Giao diện màn hình game level 4

3.1.6. Giao diện game level 5



Hình 11 : Giao diện màn hình game level 5

3.1.7. Giao diện game khi chết



Hình 12 : Giao diện màn hình game khi bị chết

3.2. Một số code chính của chương trình

3.2.1. Xây dựng kích thước của map

```
public class GamePanel extends JPanel implements Runnable, KeyListener{
    public static final int WIDTH = 320, HEIGHT = 240, SCALE = 2;

    private Thread thread;
    private boolean running;
    private int FPS = 60;
    private long targetTime = 1000/FPS;
    private BufferedImage image;
    private Graphics2D g;
    private GameStateManager gsm;
    public GamePanel()
    {
        super();
        setPreferredSize(new Dimension(WIDTH*SCALE, HEIGHT*SCALE));
        setFocusable(true);
        requestFocus();
    }
}
```

3.2.2. Thiết lập phím khi chơi

```
public void keyPressed(int k) {
    if(k == KeyEvent.VK_LEFT) player.setLeft(true);
    if(k == KeyEvent.VK_RIGHT) player.setRight(true);
    if(k == KeyEvent.VK_A) player.setLeft(true);
    if(k == KeyEvent.VK_D) player.setRight(true);
    if(k == KeyEvent.VK_UP) player.setJumping(true);
    if(k == KeyEvent.VK_DOWN) player.setDown(true);
    if(k == KeyEvent.VK_W) player.setJumping(true);
    if(k == KeyEvent.VK_E) player.setGliding(true);
    if(k == KeyEvent.VK_R) player.setScratching();
    if(k == KeyEvent.VK_F) player.setFiring();
    if(k == KeyEvent.VK_M && !player.getMute())
    {
        player.setMute(true);bgMusic.stop();}
    else if(k == KeyEvent.VK_M && player.getMute())
    {
        player.setMute(false);bgMusic.loop();}
    }
    public void keyReleased(int k) {
        if(k == KeyEvent.VK_LEFT) player.setLeft(false);
        if(k == KeyEvent.VK_RIGHT) player.setRight(false);
        if(k == KeyEvent.VK_UP) player.setJumping(false);
        if(k == KeyEvent.VK_DOWN) player.setDown(false);|
        if(k == KeyEvent.VK_W) player.setJumping(false);
        if(k == KeyEvent.VK_E) player.setGliding(false);
        if(k == KeyEvent.VK_A) player.setLeft(false);
        if(k == KeyEvent.VK_D) player.setRight(false);
    }
}
```

3.2.3. Tốc độ boss cấp 1

```
public Boss1(TileMap tm) {  
    super(tm); rank = 2;  
    moveSpeed = 1; maxSpeed = 2;  
    fallSpeed = 0.2; maxFallSpeed = 10.0;  
    width = 30; height = 30;  
    cwidth = 20; cheight = 20;  
    health = maxHealth = 70; damage = 1;  
    bounce = true; // load sprites  
    try {  
        BufferedImage spritesheet = ImageIO.read(  
            getClass().getResourceAsStream(  
                "/Sprites/Enemies/sluggger.gif"  
            ));  
        sprites = new BufferedImage[3];  
        for(int i = 0; i < sprites.length; i++) {  
            sprites[i] = spritesheet.getSubimage(  
                i * width,  
                height,  
                width,  
                height);}}  
    catch(Exception e) {  
        e.printStackTrace();  
    }  
    animation = new Animation();  
    animation.setFrames(sprites);  
    animation.setDelay(300);  
    right = true;  
    facingRight = true;  
}
```


KẾT LUẬN

4.1. Ưu và nhược điểm

❖ Ưu điểm

- Tạo ra một trò chơi giúp giải trí cao
- Không gây tốn nhiều thời gian tiền bạc

❖ Nhược điểm

- Chưa tạo ra được giao diện hoàn chỉnh bắt mắt người sử dụng
- Cách thức chơi còn chưa lôi cuốn người sử dụng

4.2. Kết luận

Sau khi hoàn thành bài tập này, nhóm đã rút ra được một số kinh nghiệm như sau:

- Hiểu rõ nhiều hơn về ngôn ngữ C#, java
- Biết tạo ra trò chơi giải trí
- Làm việc nhóm và phân công công việc một cách hiệu quả
- Bổ sung được những kiến thức còn thiếu sót khi học về lập trình

TÀI LIỆU THAM KHẢO

1. Java 2D Game Programming Platformer Tutorial - Part 1 - The Game State Manager,
Link:<https://www.youtube.com/watch?v=9dzhgsVaiSo&list=PLX0xZJGWHYp2wPT9QLu7uAnDDZeUP9p77&index=1>, truy cập lần cuối : 25/01/2018
2. Java 2D Game Programming Platformer Tutorial - Part 2 - The Game State Manager
Link:<https://www.youtube.com/watch?v=qJpdRFvSj1A&index=2&list=PLX0xZJGWHYp2wPT9QLu7uAnDDZeUP9p77>, truy cập lần cuối : 25/01/2018
3. Java 2D Game Programming Platformer Tutorial - Part 3 - The Game State Manager
Link:<https://www.youtube.com/watch?v=z9IIRG1ILOU&list=PLX0xZJGWHYp2wPT9QLu7uAnDDZeUP9p77&index=3>, truy cập lần cuối : 25/01/2018
4. Java 2D Game Programming Platformer Tutorial - Part 4 - The Game State Manager
Link:https://www.youtube.com/watch?v=zUOkoyY_Ylo&list=PLX0xZJGWHYp2wPT9QLu7uAnDDZeUP9p77&index=4, truy cập lần cuối : 25/01/2018
5. Java 2D Game Programming Platformer Tutorial - Part 5 - The Game State Manager
Link:<https://www.youtube.com/watch?v=R2DL8nGE9jg&index=5&list=PLX0xZJGWHYp2wPT9QLu7uAnDDZeUP9p77>, truy cập lần cuối : 25/01/2018
6. Java 2D Game Programming Platformer Tutorial - Part 6 - The Game State Manager
Link:https://www.youtube.com/watch?v=GofKgFpNY_g&list=PLX0xZJGWHYp2wPT9QLu7uAnDDZeUP9p77&index=6, truy cập lần cuối : 25/01/2018

7. Java 2D Game Programming Platformer Tutorial - Part 7 - The Game State Manager

Link:<https://www.youtube.com/watch?v=TJ77bQZrX7c&list=PLX0xZJGWHYp2wPT9QLu7uAnDDZeUP9p77&index=7>, truy cập lần cuối : 25/01/2018

8. Java 2D Game Programming Platformer Tutorial - Part 8 - The Game State Manager

Link:<https://www.youtube.com/watch?v=ar0hTsb9sxM&list=PLX0xZJGWHYp2wPT9QLu7uAnDDZeUP9p77&index=8>, truy cập lần cuối : 25/01/2018