

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



**MÔN : LẬP TRÌNH GAME VÀ MÔ PHỎNG  
BÁO CÁO : TỔNG KẾT ĐỒ ÁN MÔN HỌC**

**SẢN PHẨM GAME  
PACMAN**

**Nhóm 4:** 1.Nguyễn Đăng Khoa  
2.Bùi Nguyễn Hồng Phúc  
3.Trịnh Đình Phúc

**GVHD :** Nguyễn Việt Hùng

Nha Trang , Ngày 24 tháng 01 năm 2018

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



**MÔN : LẬP TRÌNH GAME VÀ MÔ PHỎNG  
BÁO CÁO : TỔNG KẾT ĐỒ ÁN MÔN HỌC**

**SẢN PHẨM GAME  
PACMAN**

**Nhóm 4:** 1.Nguyễn Đăng Khoa  
2.Bùi Nguyễn Hồng Phúc  
3.Trịnh Đình Phúc

**GVHD :** Nguyễn Việt Hùng

Nha Trang , Ngày 24 tháng 01 năm 2018

[illegible]

## **Lời nói đầu**

Thời kỳ công nghệ thông tin ngày càng phát triển , các thiết bị công nghệ cũng theo đó mà ngày một đổi mới. Chính vì vậy mà nhu cầu giải trí trên các thiết bị này cũng không ngừng đổi mới, các tựa game với các nền tảng đơn giản tới các đồ họa 3D, các công nghệ multimedia ngày càng phát triển. Trong khi các dòng game thiên về đồ họa cũng như áp dụng những công nghệ mới ngày một phát triển thì các dòng sản phẩm game đơn giản cũng ngày một đổi mới. Các nền tảng game đơn giản được phát triển dựa trên các môi trường của Java, Unity 3D cũng khá được ưa thích với người dùng và sau khi nghiên cứu cũng như được hướng dẫn qua cách làm sản phẩm game của 2 môi trường trên nhóm em quyết định chọn sản phẩm “Game Pacman” trên nền tảng Java để hệ thống kiến thức môn học cũng như phục vụ cho đồ án kết thúc môn học.

Nhóm chúng em xin chân thành cảm ơn thầy Nguyễn Việt Hùng, giáo viên hướng dẫn đã giúp nhóm chúng em hoàn thành bài báo cáo cũng như quá trình làm sản phẩm. Tuy nhiên do kinh nghiệm làm bài về sản phẩm game chưa nhiều nên còn tồn tại nhiều thiếu sót mong thầy bổ sung để bài báo cáo cũng như sản phẩm được hoàn thiện hơn .

## Mục Lục

<b>Lời nói đầu .....</b>	<b>4</b>
<b>Chương 1 TỔNG QUAN .....</b>	<b>6</b>
<b>1.1 Java là gì ?.....</b>	<b>6</b>
<b>1.2 Tại sao lại sử dụng Java để lập trình .....</b>	<b>6</b>
<b>1.3 Thông tin thành viên .....</b>	<b>7</b>
<b>Chương 2 SẢN PHẨM GAME PACMAN .....</b>	<b>8</b>
<b>2.1 Phân công nhiệm vụ và test chức năng :.....</b>	<b>8</b>
<b>2.2 Kịch bản Game PacMan: .....</b>	<b>8</b>
<b>2.3 Xây dựng ảnh các nhân vật và Map : .....</b>	<b>9</b>
<b>2.4 Các thiết kế xử lý với Code Java.....</b>	<b>10</b>
<b>2.5 Các giao diện và hướng dẫn sản phẩm .....</b>	<b>15</b>
<b>Chương 3 TỔNG KẾT .....</b>	<b>17</b>
<b>3.1 Ưu điểm và hạn chế khi hiện thực .....</b>	<b>17</b>
<b>3.2 Các kinh nghiệm học được .....</b>	<b>17</b>

## 1.1 Java là gì ?

Java là một ngôn ngữ lập trình, được phát triển bởi **Sun Microsystem** vào năm 1995, là ngôn ngữ kế thừa trực tiếp từ C/C++ và là một ngôn ngữ lập trình hướng đối tượng. Vì sao ngôn ngữ này lại được đặt tên là Java? Java là tên một hòn đảo ở Indonesia - hòn đảo nổi tiếng với loại coffee Peet và cũng là loại nước uống phổ biến của các kỹ sư Sun. Ban đầu Ngôn ngữ này được đặt tên là "*Oak*" (có nghĩa là "*Cây sồi*" - 1991), nhưng các luật sư của Sun xác định rằng tên đó đã được đăng ký nhãn hiệu nên các nhà phát triển đã phải thay thế bằng một tên mới - và cũng vì lý do trên mà cái tên Java đã ra đời và trở thành tên gọi chính thức của Ngôn ngữ này - Ngôn ngữ Lập trình Java. [1]

Ngày nay Java được sử dụng với các mục đích sau:

- Phát triển ứng dụng cho các thiết bị điện tử thông minh, các ứng dụng cho doanh nghiệp với quy mô lớn.
- Tạo các trang web có nội dung động (*web applet*), nâng cao chức năng của server.
- Phát triển nhiều loại ứng dụng khác nhau: Cơ sở dữ liệu, mạng, Internet, viễn thông, giải trí,...

## 1.2 Tại sao lại sử dụng Java để lập trình

Tiêu chí hàng đầu của Ngôn ngữ Lập trình Java là "**Write Once, Run Anywhere**" (*Viết một lần, chạy mọi nơi*), nghĩa là Java cho phép chúng ta viết code một lần và thực thi được trên các hệ điều hành khác nhau. Ví dụ, chúng ta viết code trên Hệ điều hành Windows và nó có thể thực thi được trên các Hệ điều hành Linux và Mac OS... [2]

Với đặc điểm nổi bật đó, Java có những đặc điểm cơ bản như sau:

- Đơn giản và quen thuộc: Vì Java kế thừa trực tiếp từ C/C++ nên nó có những đặc điểm của ngôn ngữ này, Java đơn giản vì mặc dù dựa trên cơ sở

C++ nhưng Sun đã cẩn thận lược bỏ các tính năng khó nhất của của C++ để làm cho ngôn ngữ này dễ sử dụng hơn.

- Hướng đối tượng và quen thuộc.
- Mạnh mẽ (*thể hiện ở cơ chế tự động thu gom rác - Garbage Collection*) và an toàn.
- Kiến trúc trung lập, độc lập nền tảng và có tính khả chuyển (*Portability*).
- Hiệu suất cao.
- Máy ảo (biên dịch và thông dịch).
- Phân tán.
- Đa nhiệm: Ngôn ngữ Java cho phép xây dựng trình ứng dụng, trong đó nhiều quá trình có thể xảy ra đồng thời. Tính đa nhiệm cho phép các nhà lập trình có thể biên soạn phần mềm đáp ứng tốt hơn, tương tác tốt hơn và thực hiện theo thời gian thực.
- ...

### 1.3 Thông tin thành viên

STT	Họ và tên	Mã Sinh Viên	Email	Số điện thoại
1	Nguyễn Đăng Khoa	14DC041	khoanguyen8947@gmail.com	0978899101
2	Bùi Nguyễn Hồng Phúc	14DC041	hongggphuccc700@gmail.com	0936216700
3	Trịnh Đình Phúc	14DC062	phuccoi996@gmail.com	01216585084

**Bảng 1. Thông tin thành viên nhóm 4**

## Chương 2

## SẢN PHẨM GAME PACMAN

### 2.1 Phân công nhiệm vụ và test chức năng :

Thành viên	Công việc	Kết quả	Test	Trạng thái
Khoa	- Nghiên cứu Java - Hướng đối tượng trên Java - Thiết kế các đối tượng Pacman, hạt gạo, kẹo bắt tử - Xử lý các sự kiện cho nhân vật Pacman	- Giao diện trên Java - Các Object tương ứng - Nhân vật pacman hành động theo luật.	Hồng Phúc	OK
Hồng Phúc	- Lập kịch bản game - Bố cục của mê cung - Đưa ra tập luật của trò chơi - Test các luật chơi - Tài liệu tổng hợp báo cáo	- Nhân vật, kịch bản hành động của các nhân vật. - Luật chơi - Báo cáo và Slide thuyết trình - TestCase	Đình Phúc	OK
Đình Phúc	- Nghiên cứu game Java - Xử lý các tọa độ hình thành mê cung theo bố cục - Xử lý các sự kiện cho nhân vật Ghost - Trí tuệ nhân tạo cho trò chơi	- Mê cung - Nhân vật ghost hành động theo luật. - Trí tuệ : thuật toán Dijkstra tìm đường đi ngắn nhất để tiêu diệt nếu thấy Pacman	Khoa	OK

**Bảng 2. Phân công nhiệm vụ của từng thành viên trong nhóm**

### 2.2 Kịch bản Game PacMan:

Nhân vật Pacman tìm đường đi để có thể ăn hết tất cả các hạt gạo trong mê cung với rất nhiều quái vật sẽ chặn giữa các lối đi. Đồng hành với Pacman trong mê cung là bốn ghost xấu xa đang canh giữ những hạt gạo trong mê cung và chúng sẽ tìm đường ngắn nhất để bắt được Pacman. Một vật phẩm hạt gạo to sẽ hỗ trợ Pacman có thể chống lại Ghost trong 10 giây.

Luật chơi: Người chơi điều khiển Pacman bằng các nút : Lên, Xuống, Trái, Phải. Nếu Pacman đụng trúng quái vật sẽ bị trừ mạng. Hết mạng thì Pacman sẽ thua

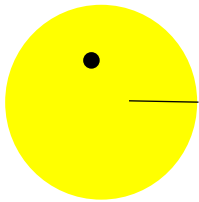


cuộc. Pacman nhận trạng thái bất khả xâm phạm trong 10 giây nếu ăn được hạt gạo to.

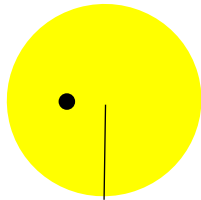
Trí tuệ nhân tạo: Ghost sẽ tìm để tiêu diệt Pacman, nếu Pacman bị phát hiện.

### 2.3 Xây dựng ảnh các nhân vật và Map :

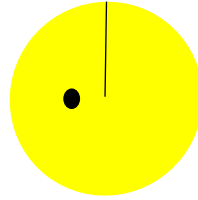
- Xây dựng các nhân vật trong Game



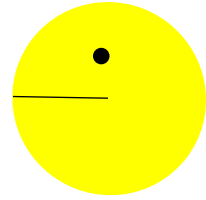
PacMan closeright



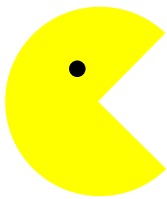
PacMan closedown



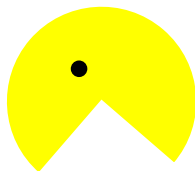
PacMan closeUP



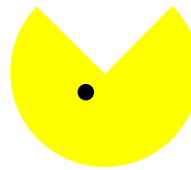
PacMan closeleft



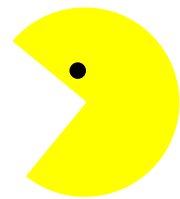
PacMan openright



PacMan opendown



PacMan openup



PacMan openleft

- Ghost



Ghost normal 1



Ghost normal 2



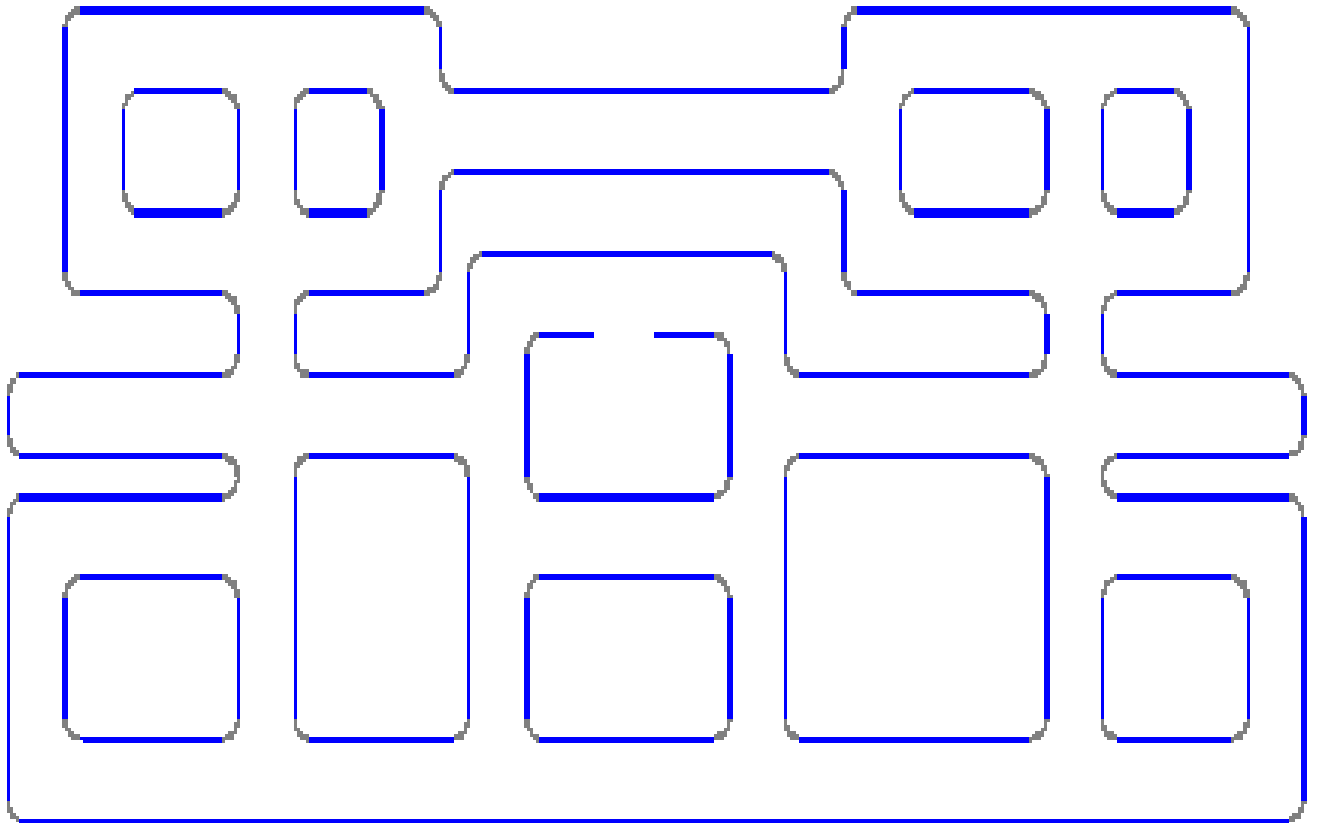
Ghost Scared 1



Ghost Scared 2

**Hình 1. Xây dựng hình ảnh các nhân vật trong Game PacMan**

- Bản đồ trong Game



**Hình 2. Xây dựng bản đồ trong Game PacMan .**

## **2.4 Các thiết kế xử lý với Code Java**

- Thiết kế xử lý PacMan

```
//xây dựng nhân vật PacMan

public Pacman(int initialRow, int initialColumn, Maze startMaze, int lives)
{
    pacmanRow = initialRow;
    pacmanCol = initialColumn;
    maze = startMaze;
    livesLeft = lives;
    cells = maze.getCells();
}

// Chuyển hình ảnh tọa độ PacMan như đang di chuyển

public void movePacman(int x, int y)
{
    pacmanRow = pacmanRow+ x;
    pacmanCol = pacmanCol+ y;
}
```

```

// lấy hướng đi của pacman để gọi ảnh phù hợp với animated
while(isRunning)
{
    if (direction == 'u')
    {
        if(isCellNavigable(pacmanCol-1,pacmanRow))
            movePacman(0,-1);
    }
    else
        if (direction == 'd')
        {
            if(isCellNavigable(pacmanCol+1,pacmanRow))
                movePacman(0,+1);
        }
        else
            if (direction == 'l')
            {
                if(isCellNavigable(pacmanCol,pacmanRow-1))
                    movePacman(-1,0);
            }
            else
                if (direction == 'r')
                {
                    if(isCellNavigable(pacmanCol,pacmanRow+1))
                        movePacman(+1,0);
                }
        }
}

public boolean isCellNavigable(int column, int row)
{
    if(cells[column][row].getType()=='o' || cells[column][row].getType()=='d'
    || cells[column][row].getType()=='p') return true;
    return false;
}

```

**//Xây dựng nhân vật Ghost**

```
public Ghost(int initialRow, int initialCol, Maze startMaze, int Number, int
timer, int difc)
{
    difficulty = difc;
    try {
        alertURL = alertFile.toURL();
    } catch (MalformedURLException ex) {
        Logger.getLogger(Ghost.class.getName()).log(Level.SEVERE, null, ex);
    }
    alertPlayer = Applet.newAudioClip(alertURL);
    escapeTimer = timer;
    ghostRow = initialRow;
    ghostCol = initialCol;
    maze = startMaze;
    cells = maze.getCells();
    ghostNumber = Number;
    if (maze.firstPlay == true) {
        createGraph();
        System.out.println("GraphSize:" + graphSize());
    }
}
```

**// Lấy vị trí của Ghost để tính toán đường đi tới Pacman**

```
public Node getGhostNode()
{
    int nodeNum = 0;
    for (int i = 0; i
        < nodes.length; i++) {
        if (getRow() == nodes[i].getX() && getCol() == nodes[i].getY()) {
            nodeNum = i;
        }
    }
    return nodes[nodeNum];
}
```

```

// Lấy vị trí của PacMan để tính toán đường đi tới Pacman
public Node getPacmanNode()
{
    int nodeNum = 0;
    for (int i = 0; i
        < nodes.length; i++) {
        if (maze.getPacmanX() == nodes[i].getX() && maze.getPacmanY() ==
nodes[i].getY()) {
            nodeNum = i;
        }
    }
    return nodes[nodeNum];
}

// Tìm đường đi ngắn nhất tới PacMan từ vị trí của Ghost thấy Pacman
public void createGraph()
{
    int counter = 0;
    shortestPath = new UndirectedGraph(graphSize());
    nodes = new Node[graphSize()];
    for (int i = 0; i < maze.tileHeight; i++) {
        for (int j = 0; j < maze.tileWidth; j++) {
            nodes[counter] = new Node("Node:" + counter);
            nodes[counter].SetX(cells[i][j].x);
            nodes[counter].SetY(cells[i][j].y);
            nodes[counter].setId(counter);
            counter++;
        }
    }
    for (int o = 0; o < graphSize(); o++) {
        shortestPath.addNode(nodes[o]);
    }
}

```

**// Xây dựng hàng đợi với các vị trí mà Pacman di chuyển**

```
PriorityQueue<Node> nodeQueue = new PriorityQueue<Node>(6, comp);

nodeQueue.add(src);

//Fills the queue with nodes
while (!temp.equals(dst) && !nodeQueue.isEmpty()) {
    temp = nodeQueue.poll();
    if (!temp.isVisited()) {
        temp.setVisited(true);
        it = temp.getNeighbors();
        while (it.hasNext()) {
            neigh = it.next();

            if (pathWeights[neigh.getId()] > pathWeights[temp.getId()]
+ getEdgeWeight(temp, neigh)) {
                pathWeights[neigh.getId()] = pathWeights[temp.getId()]
+ getEdgeWeight(temp, neigh);
                predecessors[neigh.getId()] = temp;
            }
            nodeQueue.add(neigh);
        }
    }
}
```

**//Xây dựng vector cho đường đi ngắn nhất**

```
Vector<Node> pathVec = new Vector<Node>();

Node step = dst;
while (step != null) {
    pathVec.add(step);
    step = predecessors[step.getId()];
}

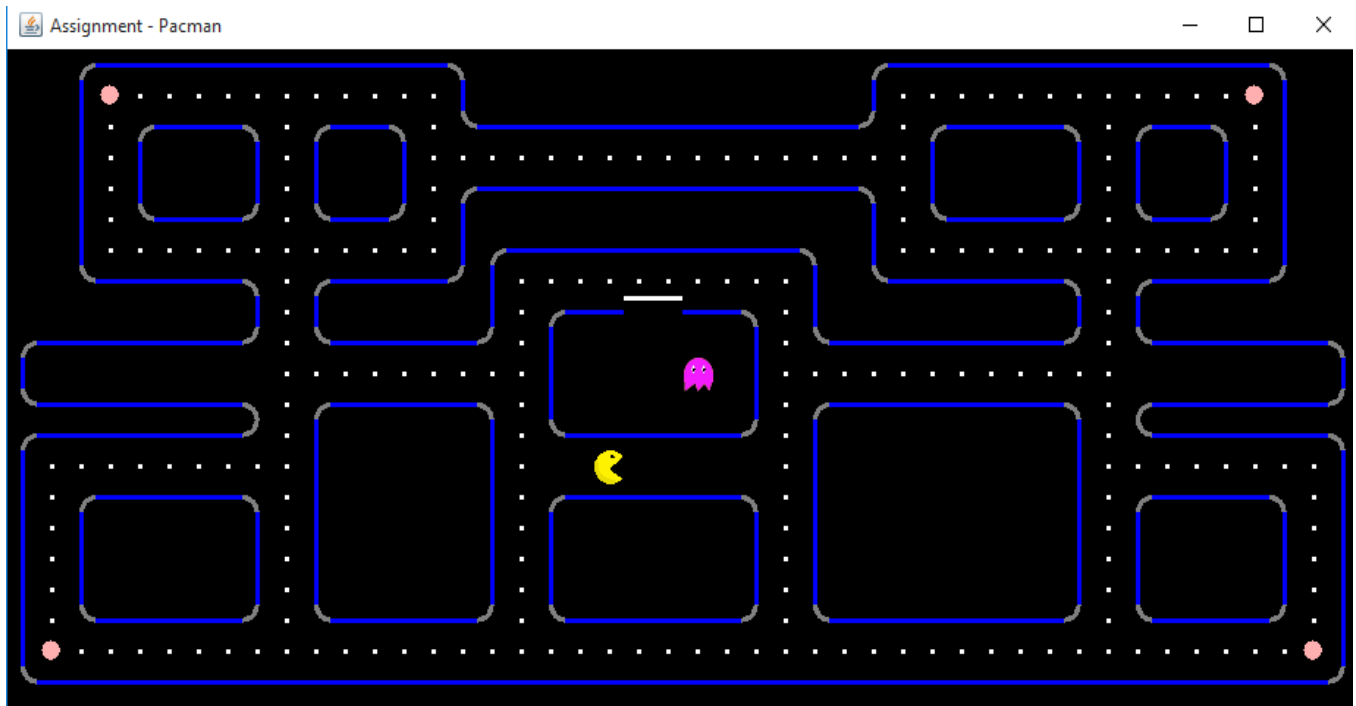
if (!(pathVec.get(pathVec.size() - 1)).equals(src)) {
    System.out.println("Cannot find shortest path");
}
```

## 2.5 Các giao diện và hướng dẫn sản phẩm

-Download gói game và giải nén > khởi động exe và thường thức

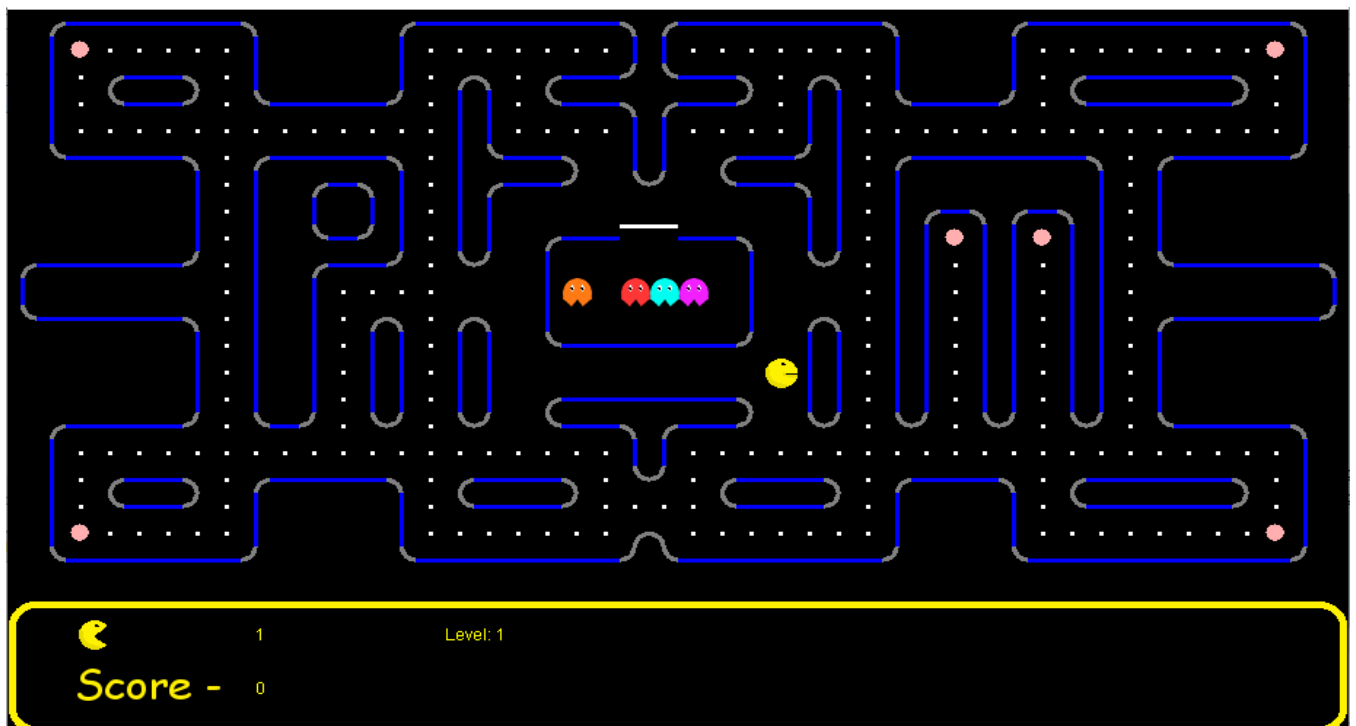
[https://drive.google.com/open?id=1sFvYl8xI\\_juicXpnB67s3HfTxBoV0Hat](https://drive.google.com/open?id=1sFvYl8xI_juicXpnB67s3HfTxBoV0Hat)

-Map 1 : Thực hiện ↑ → ↓ ← để điều khiển PacMan để ăn hết những hạt gạo



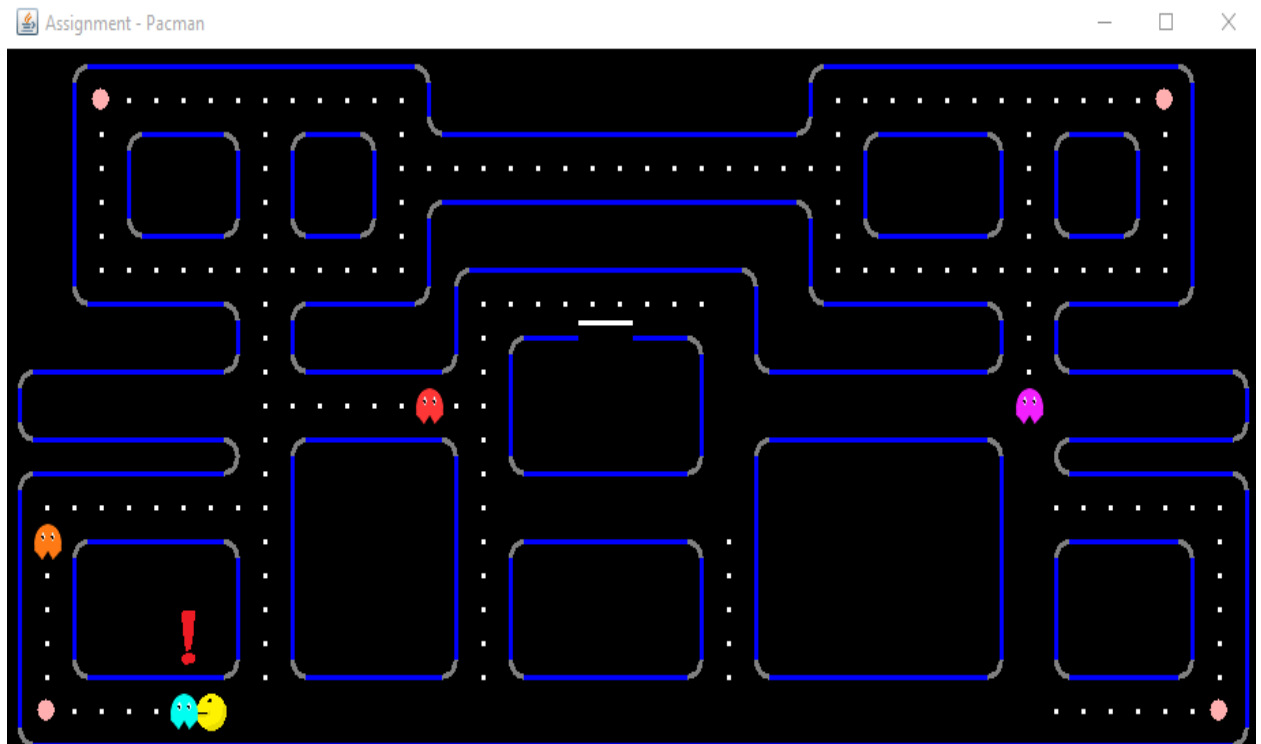
**Hình 3. Giao diện Map 1**

-Map 2: Thực hiện ↑ → ↓ ← để điều khiển PacMan để ăn hết những hạt gạo



**Hình 4. Giao diện Map 2**

-Ghost phát hiện và đuổi theo Pacman : Cố gắng tránh ghost hoặc ăn những hạt gạo lớn để có thể tiêu diệt được bọn Ghost xấu xí



Hình 4. Ghost phát hiện và rượt theo PacMan

-HighScore khi thua cuộc

<u>High Scores</u>	
1	8600
2	8260
3	8090
4	7690
5	4440

Hình 5. Bảng High Scores



### 3.1 Ưu điểm và hạn chế khi hiện thực

#### -Ưu điểm :

- +Thực hiện gần như đầy đủ các chức năng theo yêu cầu
- +Test các testcase ổn
- +Thành viên trong nhóm tích cực làm việc nhóm mặc dù có nhiều khó khăn
- +Thực hiện xong trước kỳ hạn nộp sản phẩm.

#### -Hạn chế :

- + Do thời gian làm bài khá ngắn nên còn nhiều thiếu sót.
- + Do áp dụng kiến thức mới về thuật toán Dijkstra nên còn hạn chế về các sự kiện bất lời.

### 3.2 Các kinh nghiệm học được

- Kinh nghiệm sử dụng công cụ làm việc nhóm kho Github, cách tổ chức nhánh Feature cho từng chức năng.
- Hoàn thiện kỹ năng viết Testcase
- Lập trình được Game trên nền tảng Java
- Cách các sự kiện cũng như debug lỗi trên Java

### Tài liệu tham khảo

[1] Tham khảo “Java là gì ? “ tại [http://vietjack.com/java/java\\_la\\_gi.jsp](http://vietjack.com/java/java_la_gi.jsp) có truy cập lần cuối là 20/01/2018

[2] Tham khảo “ Tại sao phải học java “ tại <http://Techmaster.vn/posts/33794/java-la-gi-tai-sao-ban-nen-hoc-lap-trinh-java> có truy cập lần cuối là 20/01/2018

[\*] Tham khảo các video làm game tại <https://www.youtube.com/watch?v=lgir2R7G9ws> có truy cập lần cuối là 15/01/2018