MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Skills Based Mini Project Report

On

PAC-MAN GAME

Submitted By:

MOHIT PURI

0901CA211031

Mentor:

Dr. Anshu Chaturvedi

(Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

Jan – June 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Mohit Puri** (0901CA211031) has submitted the project report titled **PAC-MAN GAME** under the mentorship of **Dr. Anshu Chaturvedi** (Professor), as the skills based mini project in 1st year of Master of Computer Application in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

[illegible]

Dr. Anshu Chaturvedi
(Professor)
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the fulfilment of partial requirement of the award of the skills based mini project in 1 year of Master of Computer Application in Computer Science and Engineering at **Madhav Institute of Technology & Science,** Gwalior is an authenticated and original record of my work under the mentorship of **Dr. Anshu Chaturvedi** (Professor), MITS GWALIOR.

I declare that I have not submitted the matter embodied in this report anywhere else.

[illegible]

Mohit Puri
0901CA211031
2021-2023 Year,
Master of Computer Application,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary project. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty coordinator. I am grateful to the guidance of **Dr. Anshu Chaturvedi**, Professor, Computer Science and Engineering, for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

[illegible]

Mohit Puri
0901CA211031
2021-2023 Year,
Master of Computer Application,
Computer Science and Engineering

ABSTRACT

This project discusses about the popular Pacman game using java. I built a simple Pacman implementation with the maze, a Pac-man and Pac-dots for the Pac-man to eat. For this, I created a total of eight Pac-man images, one of which is displayed depending on direction with alternating open/closed mouth. The Pac-man moves around legal positions randomly and eats the Pac-dots. I added several ghosts. With checking to see if the game has ended (ghost catches Pacman or Pacman eats all of the dots) this second increment was completed. Although the most noticeable feature of the game is the graphics representation, the implementation involves multi-threading using synchronized methods and algorithms from Data Structures.

CONTENTS

1: INTRODUCTION

Pac-man is a game in which Pac-man is an agent that eats the Pac-dots.Ghosts are agents that attack the Pac-man.The whole game is played in maze environment.

In this project, I've designed agents for the classic version of Pac-man. Along the way, I've also implemented minimax algorithm.

Pac-Man is an action maze chase video game; the player controls the eponymous character through an enclosed maze. The objective of the game is to eat all of the dots placed in the maze while avoiding four colored ghosts or spirits that pursue him. When Pac-Man eats all of the dots, the player advances to the next level. If Pac-Man makes contact with a ghost, he will lose a life; the game ends .But if pacman eats up all the dots in the maze the player is the winner.

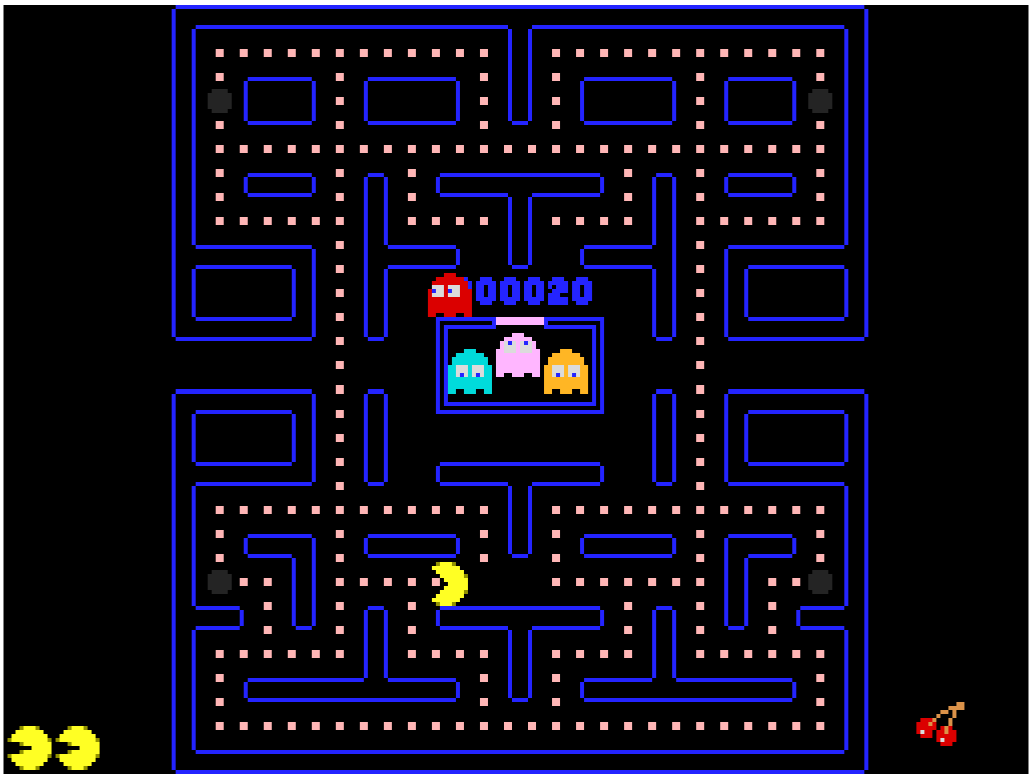

Figure 1: Classic Pac-man environment

2: ALGORITHM

It is a recursive Backtracking algorithm used in game theory. In MiniMax two players are called Maximiser and Minimiser. The Maximiser tries to get highest score possible while minimiser tries to get lowest score possible. It is widely used in two player games such as chess,tic-tac-toe,pacman etc..In this game we use MiniMax algorithm to maximise score of Pac-man while computer(acts as other player) tries to minimise the score by moving ghosts.

2.1: MINMAX ALGORITHM :

```
   i.  function minimax(node, depth, maximizingPlayer)
  ii.  if depth = 0 or node is a terminal node
 iii.  return the utility of the node
  iv.  if maximizingPlayer
   v.  bestValue := ??
  vi.  for each child of node
 vii.  v := minimax(child, depth ? 1, FALSE)
viii.  bestValue := max(bestValue, v)
  ix.  return bestValue
   x.    else (* minimizing player *)
  xi.    bestValue := +?
 xii.    for each child of node
xiii.    v := minimax(child, depth ? 1, TRUE)
 xiv.    bestValue := min(bestValue, v)
  xv.    return bestValue
```

3: DESIGN

3.1 Software Requirements Specification

This section describes the intended purpose, requirements and nature of a software developed. Software requirements specification (SRS) is a description of a software system to be developed, its defined after business requirements specification. The SRS lays out functional and nonfunctional requirements and may include a set of use cases that describe user interactions that the software must provide.

3.1.1 Purpose

The purpose of this document is to build an application which is used to provide stationary and printed materials easily

3.1.2 Requirements to Develop / Rest Web Services Application on PC

Hardware Requirements

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- 2 GB RAM minimum, 8 GB RAM recommended
- 1 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE)
- 1280 x 800 minimum screen resolution
- JDK 8

Software Requirements

- NETBEANS (IDE)
- JAVA (programming language)

4: IMPLEMENTATION

GAMES CODE:

```
Package PacMan ;

import java.awt.Canvas;

import java.awt.Graphics;

import java.awt.image.BufferStrategy;

import javax.swing.JOptionPane;

import PacMan.display.Display;

import PacMan.gfx.Assets;

import PacMan.gfx.GameCamera;

import PacMan.input.KeyManager;

import PacMan.input.MouseManager;

import PacMan.states.GameState;

import PacMan.states.MenuState;

import PacMan.states.State;

public class Game implements Runnable{

private Display display;
private Thread thread;
private int width ,height;
public String title;
private boolean running = false;
private BufferStrategy bs;
private Graphics g;
//worldNumber and Score
private int worldNumber = 1;

private int score;

//States

public State gameState;
public State menuState;
```

```
//Input

private KeyManager keyManager;
MouseManager mouseManager;

//Camera

GameCamera gameCamera;

//Handler

private Handler handler;
public Game(String title , int width , int height)
{

this.width = width;
this.height = height;
this.title = title;
keyManager = new KeyManager();
mouseManager = new MouseManager();
}

private void initComponents()
{

display = new Display(title , width , height);
display.getFrame().addKeyListener(keyManager);
display.getFrame().addMouseListener(mouseManager);
5display.getFrame().addMouseMotionListener(mouseManager);
display.getCanvas().addMouseListener(mouseManager);
display.getCanvas().addMouseMotionListener(mouseManager);

Assets.init();

handler = new Handler(this);

gameCamera = new GameCamera( handler , 0 , 0 );
gameState = new GameState(handler , worldNumber);
menuState = new MenuState(handler);
State.setCurrentState(menuState);

}
public Canvas getCanvas(){

return display.getCanvas();
}

private void tick()
{

keyManager.tick();
if(State.getCurrentState() != null)
```

```
State.getCurrentState().tick();
}
private void render()
{

bs = display.getCanvas().getBufferStrategy();
if(bs == null)
{

display.getCanvas().createBufferStrategy(3);
return;
}
6g = bs.getDrawGraphics();

//Clear Screen

g.clearRect(0,0, width, height);

//Draw

if(State.getCurrentState() != null)
State.getCurrentState().render(g);

//End Draw

bs.show();
g.dispose();
}

public void run()
{

initComponents();
int fps = 60; //frame per second
double timePerTick = 1000000000/fps;
double delta = 0;
long now ;
long lastTime = System.nanoTime();
while(running)
{

now = System.nanoTime();
delta += (now-lastTime)/timePerTick;
lastTime = now;
if(delta >= 1){

tick();
7render();
delta--;
}

}
```

```
stop();
}

public Graphics getG() {
return g;
}

public KeyManager getKeyManager()
{

return keyManager;
}

public MouseManager getMouseManager() {
return mouseManager;
}

public GameCamera getGameCamera() {
return gameCamera;
}

public int getWidth() {
return width;
}

8public int getHeight() {
return height;
}

public int getWorldNumber(){
return worldNumber;
}

public void setWorldNumber(int worldNumber){
this.worldNumber = worldNumber;
}

public int getScore() {
return score;
}

public void setScore(int score) {
this.score = this.score + score;
}

public synchronized void start()
{

if(running)
return;
running = true;
thread = new Thread(this);
thread.start();
```

```
}

public synchronized void stop()
9{

if(!running)
return;
running = false;

try {

thread.join();

} catch (InterruptedException ex) {

JOptionPane.showMessageDialog(null,
ex,"ERROR",JOptionPane.ERROR_MESSAGE);

}

}

}
```

}

HANDLER CODE:

```
package PacMan;

import PacMan.gfx.GameCamera;

import PacMan.input.KeyManager;

import PacMan.input.MouseManager;

import PacMan.worlds.World;

public class Handler {

private Game game;
private World world;
public Handler(Game game)
{

this.game = game;
}

public KeyManager getKeyManager()
{

return game.getKeyManager();
}

public MouseManager getMouseManager(){

return game.getMouseManager();
}

public int getWidth()
{

return game.getWidth();
}

11public int getHeight()
{

return game.getHeight();
}

public GameCamera getGameCamera()
{

return game.getGameCamera();
}
public Game getGame() {

return game;

}

public void setGame(Game game) {

this.game = game;

}

public World getWorld() {

return world;

}

public void setWorld(World world) {

this.world = world;

}

}
```

LAUNCHER CODE :

```
package PacMan;

import PacMan.display.Display;

import PacMan.states.GameState;

import PacMan.states.State;

import java.io.File;

import java.io.FileInputStream;

import javafx.embed.swing.JFXPanel;

import javafx.scene.media.Media;

import javafx.scene.media.MediaPlayer;

import sun.audio.AudioData;

import sun.audio.AudioPlayer;

import sun.audio.AudioStream;

import sun.audio.ContinuousAudioDataStream;

public class Launcher
{

public static void main(String[] args)
{

Game game = new Game("PacMan !!",1300,1000);

game.start();

}

}
```

5: RESULTS

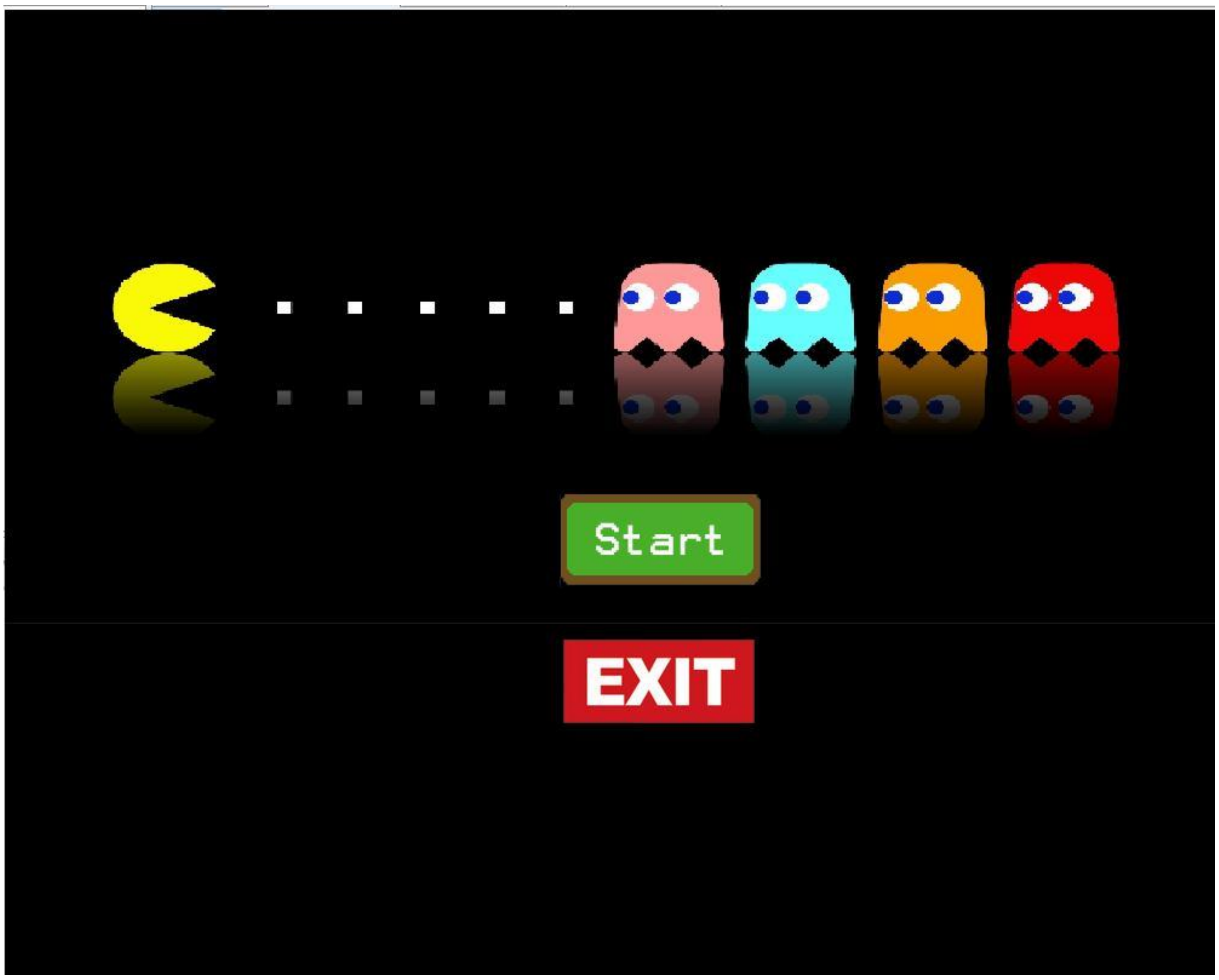

Figure 2 : Game Menu

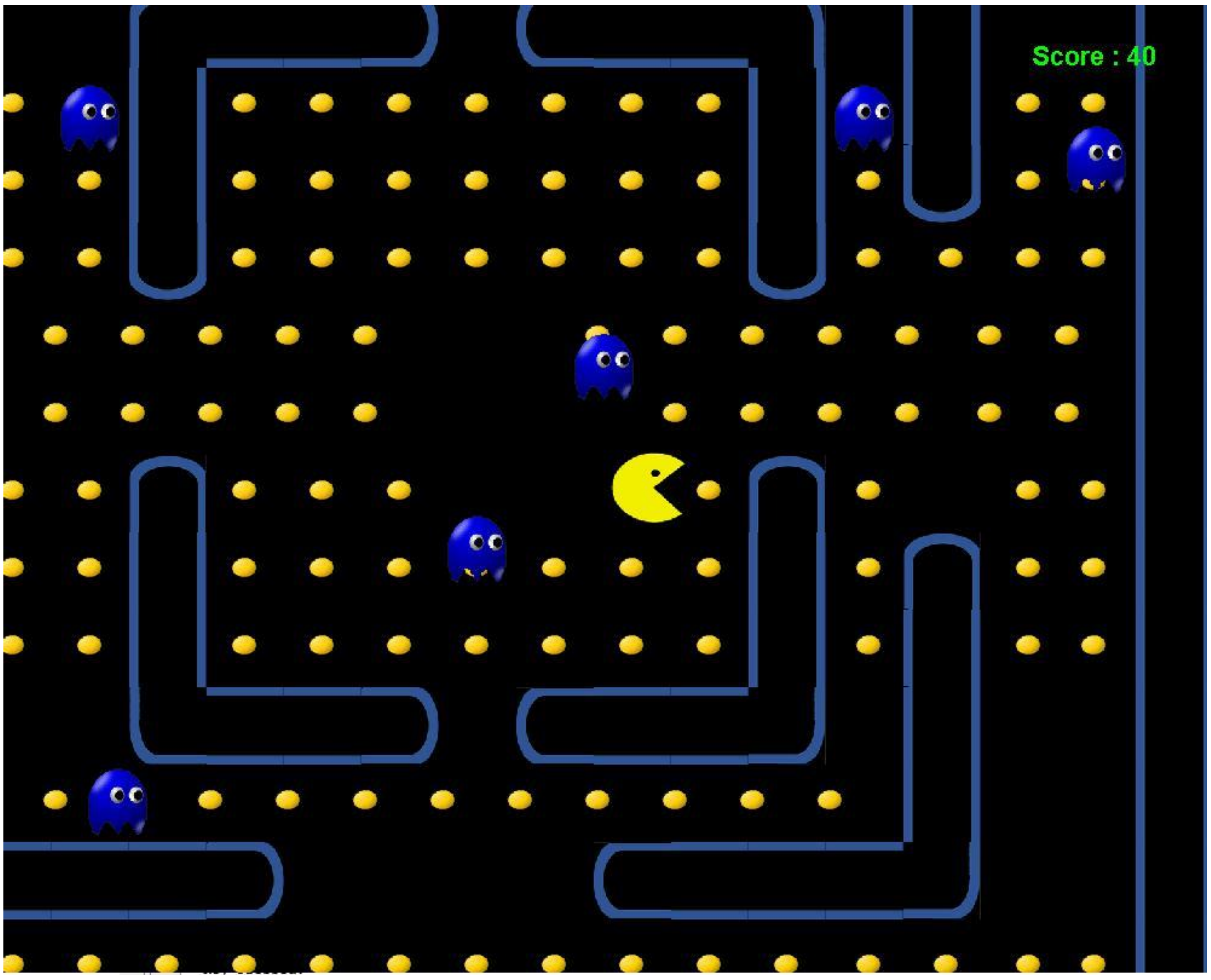

Figure 3 : Game Processing

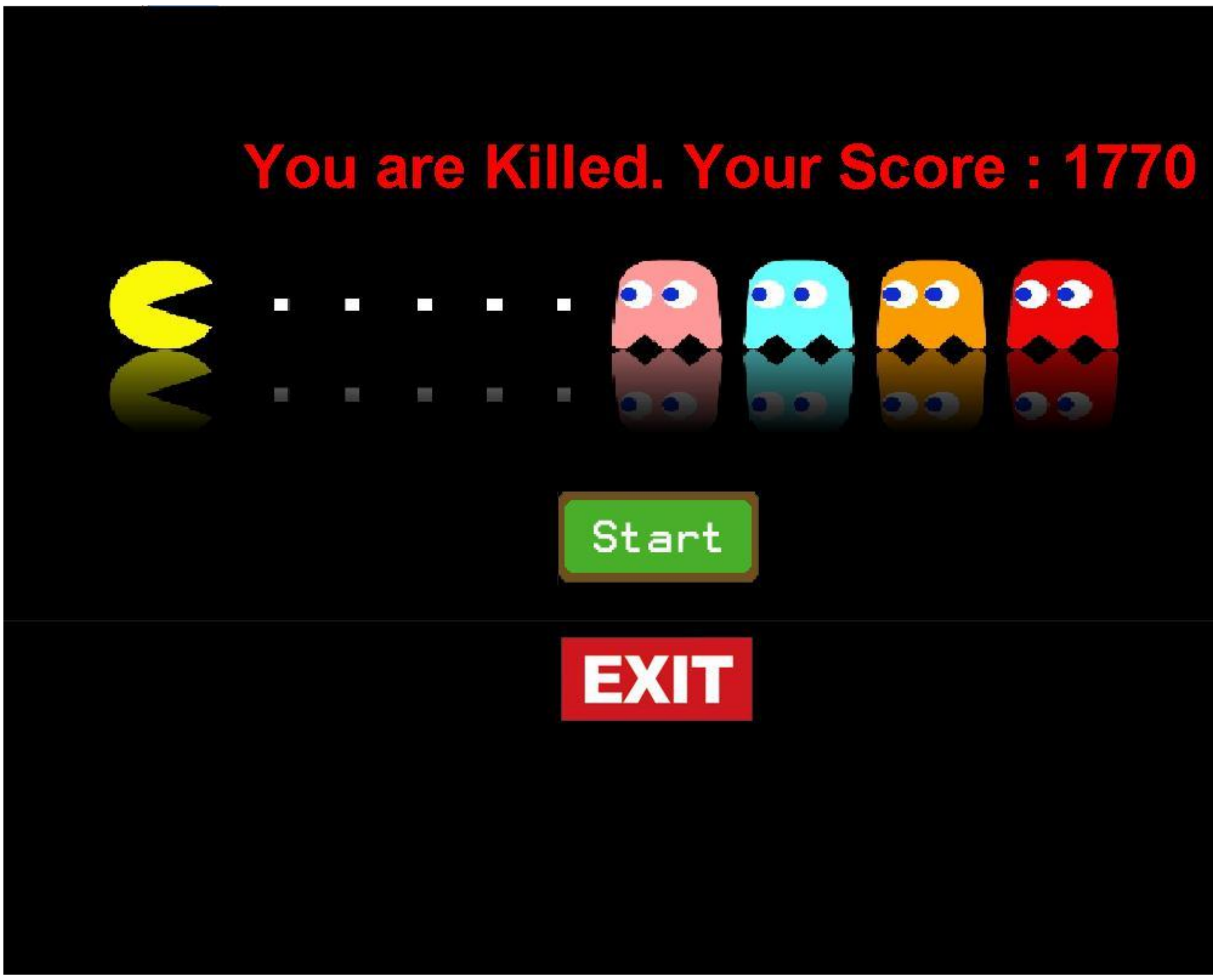

Figure 4 : Game End

6: CONCLUSION

The player who succeeds in eating more dots by avoiding ghosts scores much points. The code I wrote is for desktop application. This can be extended and can be used in mobile applications so that it will be very flexible for the user to play the game. The project which I undertaken has helped me gain a better perspective on various aspects related to my course of study as well as particular knowledge of particular web-based applications. I became familiar with software analysis, designing Sprite Sheets,
Implementation, testing and maintenance considered with my project.

7: WEBLINKS

1 https://stackoverflow.com
2 https://www.tutorialpoint.com
3 https://www.wikipedia.com
4 https://www.slideshare.com
5 https://www.w3schools.com