**REPORT FOR SNAKE GAME AI**

As a

project

work for Course

**ARTIFICIAL INTELLIGENCE (INT404)**

---------------------------------------------------------------------------------------------------------------------------

Name : ABHIKUSH K SINGH

Registration Number : 11802749

Name

:

SHIVAM CHAUHAN

Re

gistration Number : 1180

**2772**

Program : CSE B.Tech

Semester : FORTH

School : School of Computer

Science and

Engineering

Name of the Unive

rsity : Lovely Professional

University

Date of

submission :

31

th

MARCH

2020

---------------------------------------------------------------------------------------------------------------------------

Lovely Professional University

Jalandhar, Punjab, India.



**SNAKE GAME AI**

**31th MARCH 2020**

# ABSTRACT:-

This project aims to bring the fun and simplicity of snake game with some new features. It will

include computer controlled intelligent opponents whose aim will be to challenge the human

players. It will also have the multiplayer feature that will allow more than one players to play

the game over a network.

This project explores a new dimension in the traditional snake game to make it more interesting and challenging.

## *ACKNOWLEDGEMENT:-*

I would like to thank my mentor - Prof. Sagar Pande for his advice and inputs on this project. Many thanks to my friends and seniors as well, who spent countless hours to listen and provide feedbacks.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Table of Contents** | | |  | |
|  | **1. ABSTRACT** | **2** | |
| 1. **INTRODUCTION**     1. **CONTEXT**    2. **MOTIVATION**    3. **IDEA** | **4** | |
| 1. **TEAM MEMBERS WITH ROLES**     1. **TEAM LEADER**    2. **MEMBERS**    3. **CONTRIBUTIONS** | **5** | |
| 1. **LIBRARIES**     1. **DIFFERENT TYPES**    2. **WHY THEY ARE USED** | **6** | |
| 1. **Objectives**   **5.1**. **Methodology**  **5.2. Programming Environment** | **7** |  |
| 1. **Screenshots** |  |
| 1. **ALGORITHMS**    1. **Shortest Path BFS**    2. **Shortest Path DFS**    3. **Longest path**    4. **Hamilton** | **11** |
| 1. **How to make snake game**   **8.1 Flow chart** | **17** |
| **9.Refrences** | **22** |
|  | | |

# INTRODUCTION:-

1.1 Context

This project has been done as part of my course for the CSE at Lovely Professional University . Supervised by Sagar Pande, I have three months to fulfill the requirements in order to succeed the module.

1.2 Motivations

Being extremely interested in everything having a relation with the Artificial Intelligence , the group project was a great occasion to give us the time to learn and confirm our interest for this field. The fact that we can make estimations, predictions and give the ability for machines to learn by themselves is both powerful and limitless in term of application possibilities. That’s why I decided to conduct my project around the Artificial Intelligence.

1.3 Idea:-

As a first experience, we wanted to make my project as much didactic as possible by approaching every different steps of the Artificial Intelligence process and trying to understand them deeply. Known as ” toy problem” the problems that are not immediate scientific interest but useful to illustrate and practice, we chose to take Snake games as approach. The goal was to a game which work or play automatically by using its own intelligence.

# TEAM MEMBERS:-

TEAM LEADER:-

Abhikush Singh:-

Contributions:-

1. Coding(joined)
2. Multivariable Regressing
3. GUI
4. Artificial Intelligence (joined)

Shivam chauhan:-

Contributions:-

1. Coding(joined)
2. Datasets
3. Linear regression
4. Reports
5. Artificial Intelligence (joined)

# LIBRARIES:-

## Numpy:-

NumPy is a general-purpose array-processing package. It provides a highperformance multidimensional array object, and tools for working with these arrays.It is the fundamental package for scientific computing with Python.

## Pandas:-

***Pandas*** is the most popular python library that is used for data analysis. We will provide highly optimized performance with back-end source code with the use of Pandas.

## Pygame:-

***Pygame*** is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language

**Objectives:-**

This game aims to change the way people think of traditional snake game. It will offer the experience of commercial games to the player retaining the simplicity of traditional snake game.

The major objectives of this project are:

Create a snake game that will have all the functionality of traditional snake games.

\* Introduce artificial intelligence functionality in the game that will allow artificial intelligent to play a game simultaneously.

\* Introduce computer controlled intelligent opponent (unique feature of this game) to make the game more challenging and interesting. The movement and action of these intelligent opponents will be controlled by computer whose aim will be to eat the food.

**Methodology:-**

The complete Snake game application is divided into four major components:

\* Snake Client Application

\* Snake Game Server

\* Intelligent Autonomous Opponent Snakes

\* Snake Game Server Manager (GSM)

**Programming Environment:-**

**\* Netbeans 5.5 IDE**

All the developers of Snake game used Netbeans IDE for the development of this

project.

**\*Inkscape 0.45 and Gimp 2.2**

These graphics development tools were extensively used for the development of user interface components . The illustrations presented in this report have also been prepared using these open source tools.

**\* Gnuplot**

The data obtained during profiling of two path finding algorithms viz.

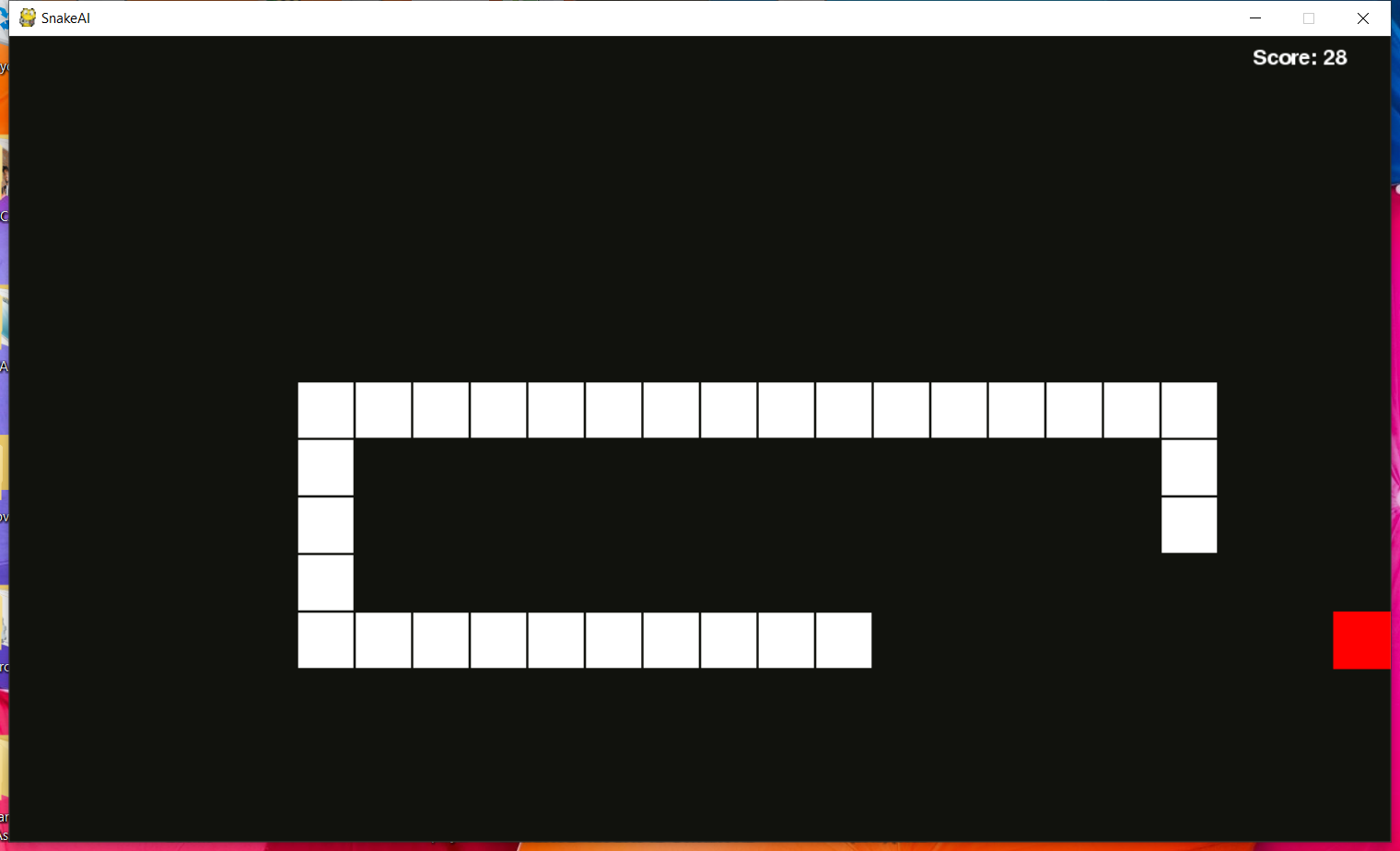
Blackmamba and Viper was plotted using gnuplot.

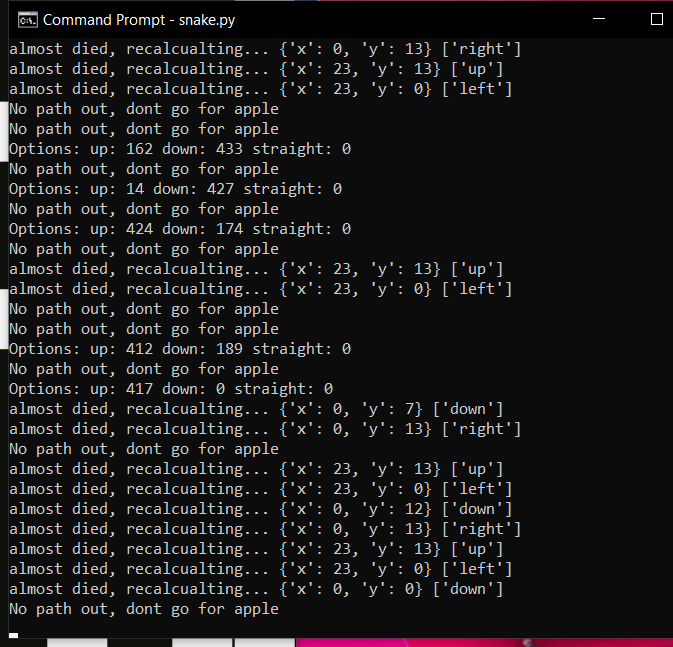
**\*OpenOffice Writer 2.2**

All the project documents and this report were prepared using OpenOffice Writer.

# SCREENSHOTS:-

**Game:-**



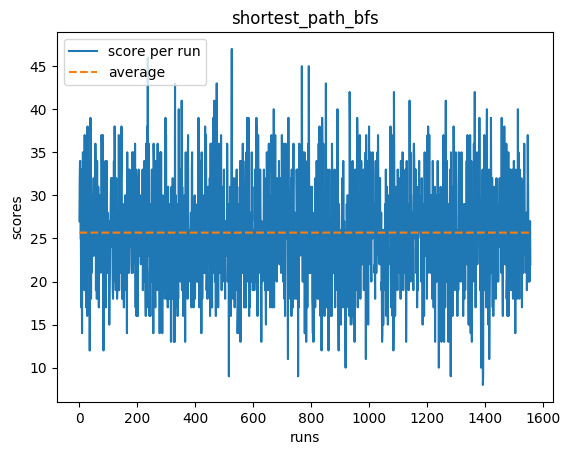


**ALGORITHMS**

## ****Shortest Path BFS - Breadth First Search****

BFS ([Breadth First Search](https://en.wikipedia.org/wiki/Breadth-first_search)) is a graph traversal algorithm and it isn’t the shortest path algorithm per se (if you are more curious you can check and implement [Dijkstra](https://en.wikipedia.org/wiki/Dijkstra's_algorithm) or [A\*](https://en.wikipedia.org/wiki/A*_search_algorithm)), but in our case, it can get a job done.

The algorithm starts at the root node (head of the snake) and explores all neighbor nodes (positions on the grid) at the present depth, before going deeper. It terminates when it finds the fruit.

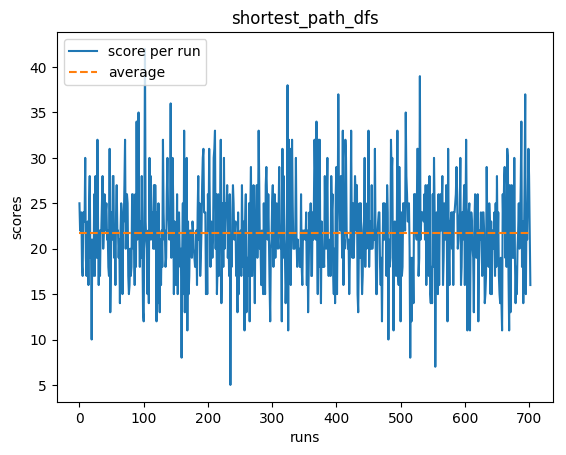


Shortest Path BFS performs optimally until snake’s length interrupts its shortest path to the fruit.

CODE:- python slitherin.py --shortest\_path\_bfs

## ****Shortest Path DFS - Depth First Search****

If you take a look at the DFS ([Depth First Search](https://en.wikipedia.org/wiki/Depth-first_search)) implementation and compare it with BFS you’ll notice that it’s exactly the same except for one difference - data structure that holds neighbor nodes to explore. BFS uses a queue which is a FIFO list type (First In First Out) and DFS uses a stack which is LIFO (Last in First Out). This subtle difference drastically changes the behavior of the algorithm. While BFS starts with exploring present-depth nodes, DFS starts with the highest-depth first. It makes DFS a good way of checking if a graph contains a cycle or not (FYI: Snake’s graph is cyclic). Without termination when detecting a cycle our DFS algorithm would go forever still finding a deeper neighbor.

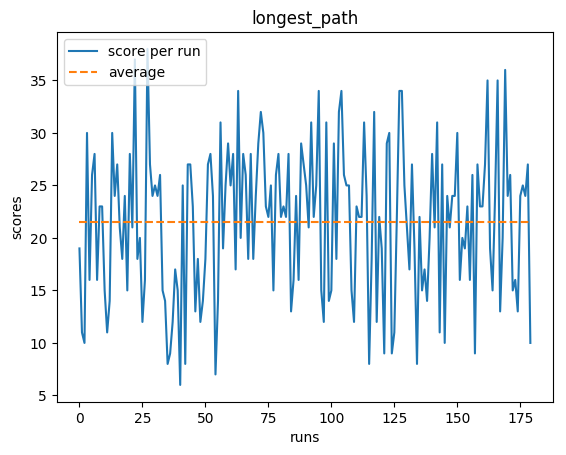


We’ve already explored two shortest path solvers and while they yield decent results, they are nowhere close to solving the game. Their major drawback is that they don’t take into consideration the snake’s body which gets longer after every eaten fruit.After a couple of games played, I have found that as it doesn’t matter how fast are we going to eat the fruit, we may choose to take the longest path to reach it instead of the shortest one, just to make sure that we do it safely.

***CODE:-***python slitherin.py --shortest\_path\_dfs

## ****Longest Path:-****

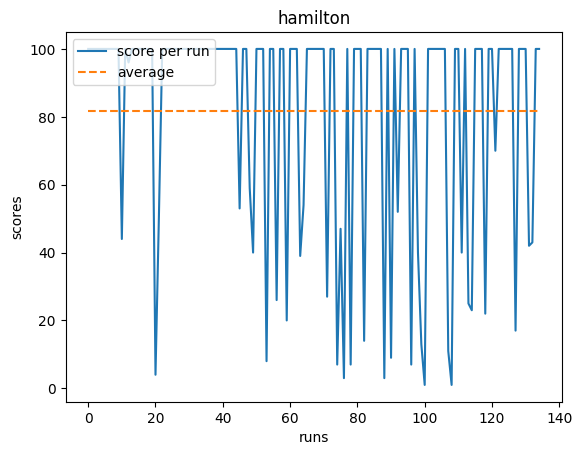
Longest Path’s results are comparable and definitely not significantly better than the ones from the Shortest Path algorithms.We didn’t see any significant improvement, because while following the longest path to the fruit, our snake may leave its body on the path and accidentally crash into it after eating the fruit.



## CODE:-python slitherin.py --longest\_path

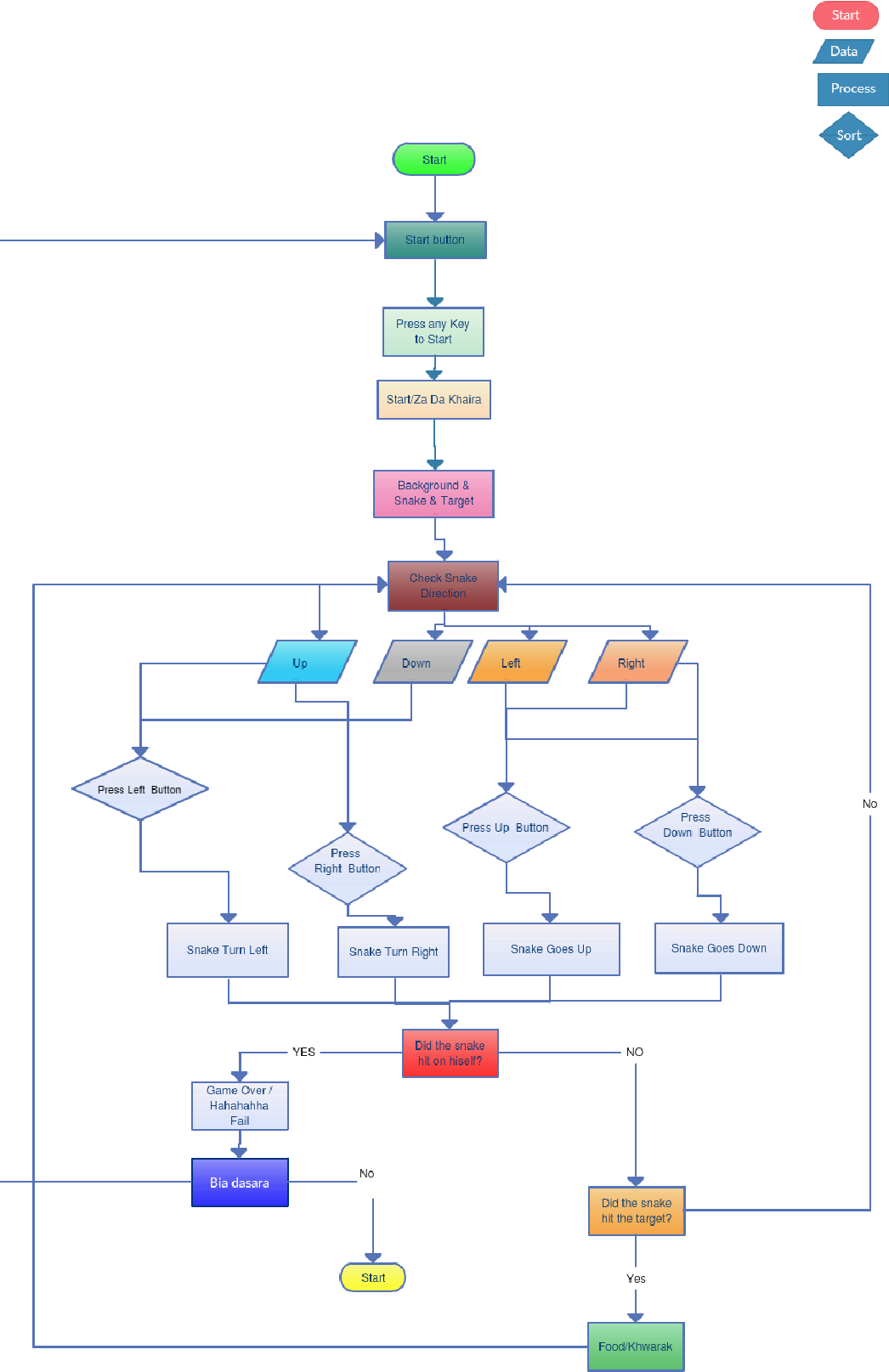
## ****Hamilton:-****

Hamilton solver gets a perfect score of width\*height (100 in above example) thus solving a game in the majority of the cases. Creating a Hamiltonian cycle is not always possible as it depends on the initial snake position and its direction.



CODE***:-*** python slitherin.py --hamilton

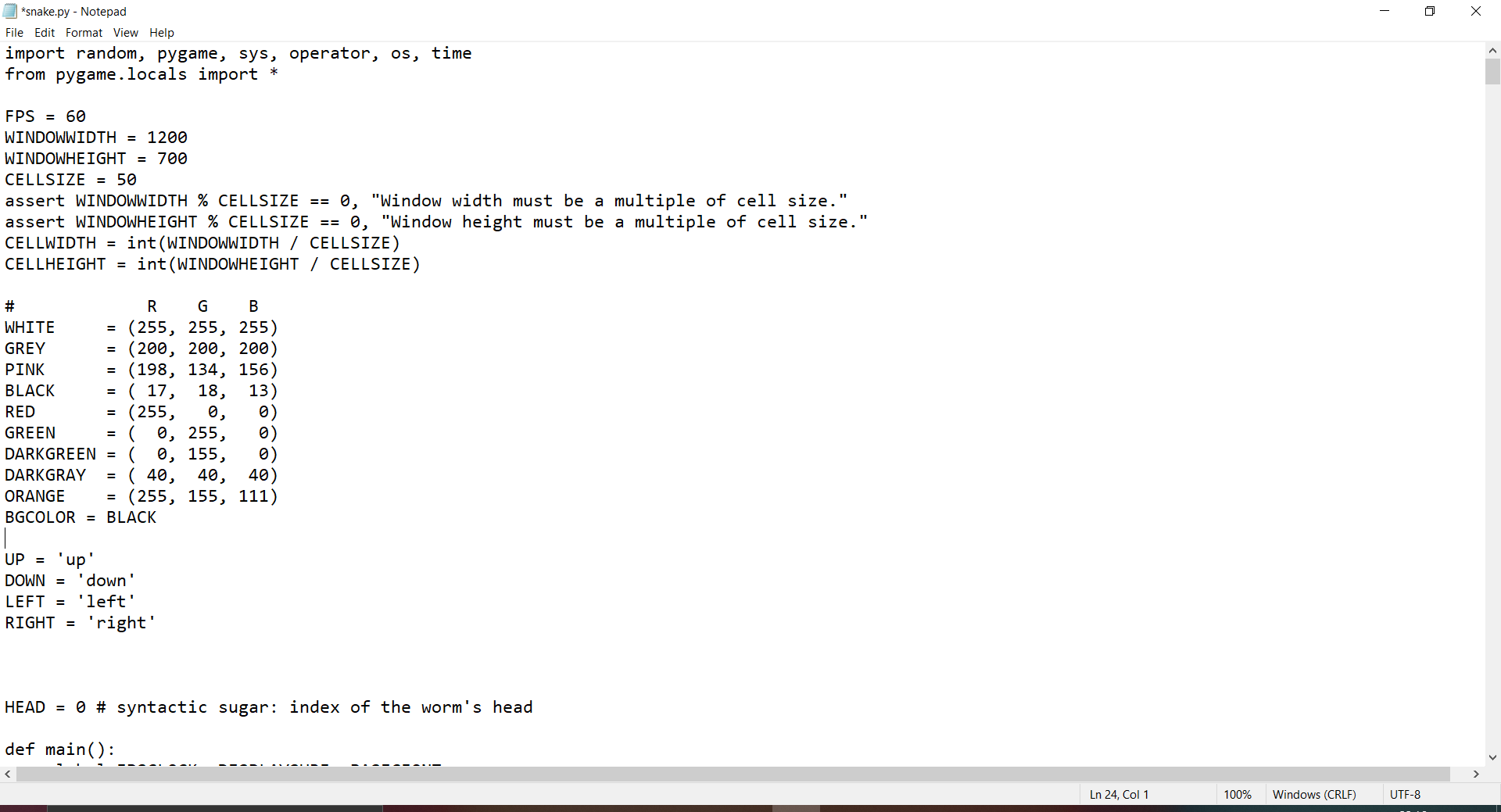
**FLOW CHART:-**



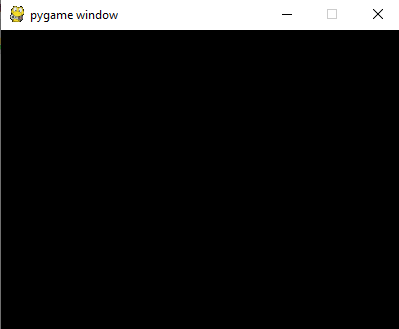
***HOW TO MAKE SNAKE GAME:-***

## ****Create the Screen:-****

To create the screen using Pygame, you will need to make use of the display.set\_mode() [function](https://www.edureka.co/blog/python-functions). Also, you will have to make use of the init()  and the quit() methods to initialize and uninitialize everything at the start and the end of the code. The update() method is used to update any changes made to the screen. There is another method i.e flip() that works similarly to the update() function. The difference is that the update() method updates only the changes that are made (however, if no parameters are passed, updates the complete screen) but the flip() method redoes the complete screen again.



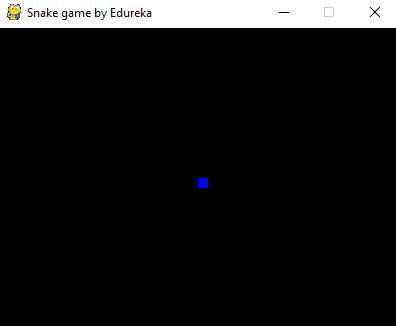
**OUTPUT:-**



## ****Create the Snake:-****

To create the snake, I will first initialize a few color variables in order to color the snake, food, screen, etc. The color scheme used in Pygame is RGB i.e “Red Green Blue”. In case you set all these to 0’s, the color will be black and all 255’s will be white. So our snake will actually be a rectangle. To draw rectangles in Pygame, you can make use of a function called draw.rect() which will help yo draw the rectangle with the desired color and size.

**OUTPUT:-**



## Conclusions:-

It is our team’s hope that this document will be of huge help with understanding of our little project as we have used a different approach which has proved beneficialfor us and easy for us to understand the Artificial intelligent. The computer

controlled intelligent opponents have been successfully tested in the game is a unique feature of Snake game.

We learned several project management techniques used by professionals to develop large scale project. The experience of working in team and integration of modules developed independently, with just requirement specifications, is a very important achievement for my group.

# REFRENCES:-

To conduct this project the following tools have been used :

* Jupyter notebook and spyder
* Pandas (Library) : <http://pandas.pydata.org/> ● Numpy (Library) : <http://www.numpy.org/>

## *Coursera:-*

We have used this side for our basis knowledge gain of the methods that will be used in the project

[***https://www.coursera.org/learn/machine-learning-with-python***](https://www.coursera.org/learn/machine-learning-with-python)

**EDUREKA:-**

[**https://www.edureka.co/blog/snake-game-with-pygame/**](https://www.edureka.co/blog/snake-game-with-pygame/)