**VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM 590014**

****

Python Project Report on

**“PACMAN GAME”**

By

SHARAN RUDRESH (1BM16CS092)

PRANAV AGARWAL(1BM16CS069)

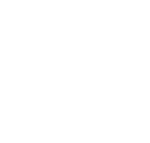
Under the Guidance of

**Mr. Vikranth B.M.**

Assistant Professor, Department of CSE

BMS College of Engineering

Python Project carried out at



Department of Computer Science and Engineering

BMS College of Engineering

(Autonomous college under VTU)

P.O. Box No.: 1908, Bull Temple Road, Bangalore-560 019

2018-2019

**BMS COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

***CERTIFICATE***

This is to certify that the python project development titled “PACMAN GAME” has been carried out by SHARAN RUDRESH(1BM16CS092), PRANAV AGARWAL(1BM16CS069) during academic year 2018-2019.

**Signature of Examiners**

Signature of the guide

**Mr.Vikranth B.M**

Assistant Professor

Department of Computer Science and Engineering

BMS College of Engineering, Bangalore

**BMS COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

***DECLARATION***

We, SHARAN RUDRESH(1BM16CS092), PRANAV AGARWAL(1BM16CS069) students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this Python PyGame project development work entitled "Pacman Game" has been carried out by us under the guidance of Mr.Vikranth B.M, Assistant Professor, Department of CSE, BMS College of Engineering, Bangalore during the academic semester Aug-Dec 2018.

We also declare that to the best of our knowledge and belief, the development reported here is not from any other work/report by any other students.

Signature

SHARAN RUDRESH(1BM16CS092)

PRANAV AGARWAL(1BM16CS069)

**ABSTRACT**

We have tried to implement some of the algorithms learnt by us in our course in a fun and interesting way. This way we expect the students to learn the algorithms and apply them in fun applications. Pacman is an arcade game developed by Namco and first released in Japan as Puck Man in May 1980. It is also one of the highest-grossing video games of all time. According to the Davie-Brown Index, Pac-Man has the highest brand awareness of any video game character among American consumers, recognized by 94 percent of them.[20] Pac-Man is one of the longest-running video game franchises from the golden age of video arcade games. The Google Doodle version of Pac-Man was estimated to have been played by over 1 billion people worldwide in 2010.

We have used Pythons package pygame to make this game. With this project we hope to recreate the magic and nostalgia of the original game.

**TABLE OF CONTENTS**

1. Introduction

1.1 A\* Algorithm

1. Agents
2. Software Used
3. Screenshots
4. Conclusion
5. Future Enhancements

**1.INTRODUCTION**

The player navigates Pac-Man through a maze containing various dots, known as Pac-Dots, and four multi-colored ghosts: Blinky, Pinky, Inky and Clyde. The goal of the game is to accumulate points by eating all the Pac-Dots in the maze, completing that 'stage' of the game and starting the next stage and maze of Pac-dots. Between some stages, one of three intermission animations plays.[24] The four ghosts roam the maze hunting for Pac-Man. If any one of the ghosts touches him, a life is lost; when all lives have been lost, the game is over.

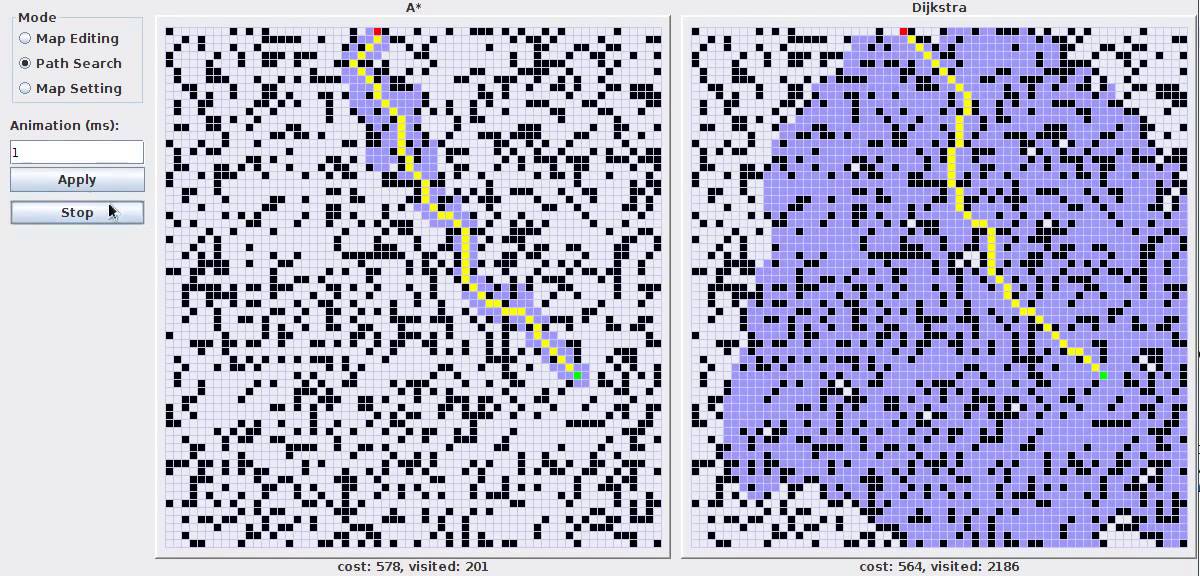
* 1. **A\* Algorithm**

A\* is an informed search algorithm, or a best-first search, meaning that it is formulated in terms of weighted graphs: starting from a specific starting node of a graph, it aims to find a path to the given goal node having the smallest cost (least distance travelled, shortest time, etc.). It does this by maintaining a tree of paths originating at the start node and extending those paths one edge at a time until its termination criterion is satisfied.

At each iteration of its main loop, A\* needs to determine which of its paths to extend. It does so based on the cost of the path and an estimate of the cost required to extend the path all the way to the goal. Specifically, A\* selects the path that minimizes

f ( n ) = g ( n ) + h ( n )

where n is the next node on the path, g(n) is the cost of the path from the start node to n, and h(n) is a heuristic function that estimates the cost of the cheapest path from n to the goal. A\* terminates when the path it chooses to extend is a path from start to goal or if there are no paths eligible to be extended. The heuristic function is problem-specific. If the heuristic function is admissible, meaning that it never overestimates the actual cost to get to the goal, A\* is guaranteed to return a least-cost path from start to goal.



**2. AGENTS**

**2.1 Grid**

The grid is a maze for the player to navigate around. There are walls and open paths in the grids. The player and the ghosts can move only on open paths. We have handled all collision scenarios of the player and the ghost with the walls of the maze.

**2.2 Pacman**

Pacman is the player. The pacman figure is controlled by the users input using the arrow keys. The user cannot collide with the gridwalls. The users main objective is to collect as many coins as he can and be away from the ghosts. If the ghosts touch the pacman a life is lost. Every player is given three lives to start with.

**2.3 Coins**

Coins are placed all along the open paths on the maze. As the pacman touches the coins the coins disapper. Also the score of the pacman is incremented. The ghosts cannot collect coins. So when the ghosts touch the coins they pass through it.

**2.4 Ghost**

Ghosts are the creatures that follow the pacman. The enemies in Pac-Man are known variously as "ghosts", "goblins", and "monsters". Despite the seemingly random nature of the enemies, their movements are strictly deterministic. Each ghost has their own distinct personality in order to keep the game from becoming impossibly difficult or boring to play. The red enemy chases Pac-Man, and the pink and blue enemies try to position themselves in front of Pac-Man's mouth. orange enemy's behavior is random.

**Blinky**

Shadow, better known by his nickname, Blinky, is a red ghost who, in the original arcade game, follows behind Pac-Man. He is considered the leader of the ghosts. Blinky Uses the A\* algorithm by use pacman’s location as the final destination.

**Pinky**

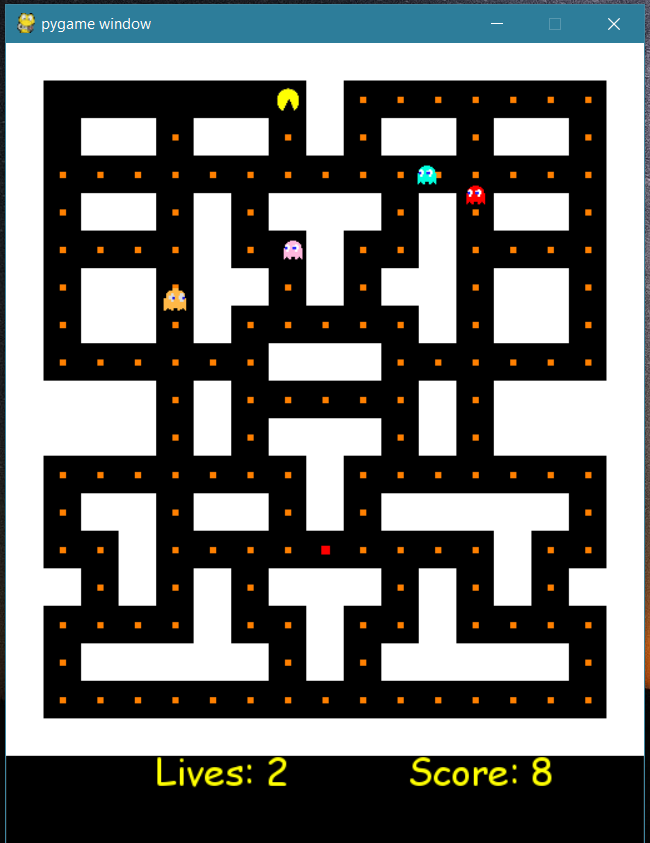
Speedy, known better as Pinky, is a pink ghost, who is in the original arcade game, positions in front of Pac-Man. In the Pac-Man game the ghost uses 4 steps ahead of pacman to the A\* algorithm as the final destination.

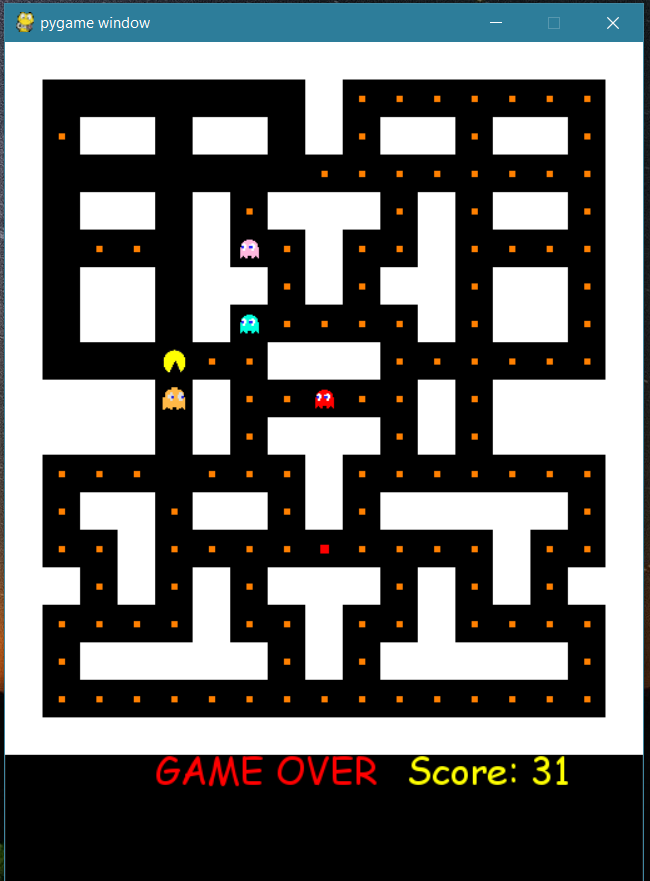
**Inky**

Bashful, or, Inky is a baby blue ghost who, in the original arcade game, has a fickle mood. He can be unpredictable. Sometimes he chases Pac-Man aggressively like Blinky; other times he jumps ahead of Pac-Man as Pinky would. He uses the position of Blinky and pacman to determine where pacman would move next and uses for the A\* algorithm.

**Clyde**

Clyde, whose original name is listed as Pokey, is an orange ghost who, in the original arcade game, acts stupid. He will chase after Pac-Man in Blinky's manner. He has a random behavior which is unpredictable. His main objective is to scare pacman.

**4. SCREENSHOTS**



**5.CONCLUSION**

Python is actually quite capable at running games. It will likely even surprise you how much is possible in under 30 milliseconds. Still, it is not hard to reach the ceiling once your game begins to get more complex. Any game running in realtime will be making full use of the computer.

Over the past several years there has been an interesting trend in game development, the move towards higher level languages. Usually a game is split into two major parts. The game engine, which must be as fast as possible, and the game logic, which makes the engine actually do something. It wasn't long ago when the engine of a game was written in assembly, with portions written in C. Nowadays, C has moved to the game engine, while often the game itself is written in higher level scripting languages. there are very clear limitations. The best way to manage hardware accelerated graphics is not always the way to get fastest results from software rendering. Hardware support is not available on all platforms. When a game gets more complex, it often must commit to one or the other. SDL has some other design limitations, things like full screen scrolling graphics can quickly bring your game down to unplayable speeds. While SDL is not suitable for all types of games, remember companies like Loki have used SDL to run a wide variety of retail quality titles.

Pygame is fairly low-level when it comes to writing games. You'll quickly find yourself needing to wrap common functions into your own game environment. The great thing about this is there is nothing inside pygame to get in your way. Your program is in full control of everything. The side effect of that is you will find yourself borrowing a lot of code to get a more advanced framework put together. You'll need a better understanding of what you are doing.

**6. FUTURE ENHANCEMENTS**

We plan to advance the game by including multiple levels. We also plan to Automate Pacman by using machine learning modules. We can also add multiple modes like two player, or online gaming.