

By-

Prashanthi sudha Kosgi

PROJECT REPORT

Enhanced PAC-MAN

CSC7014

Practice of Computer Programming

Prof |Nguyen Thai

**b. Executive Summary:**

Key capabilities of the project

1. Main screen with instructions to the player.
2. Game with music.
3. PAC-MAN move.
4. Coins and power boosters.
5. Gain points.
6. Movement of ghosts in maze.
7. Collision of ghosts with PAC-MAN.
8. Display score, time left and lives remaining.
9. Display results.
10. Restart the game.
11. Quit the game.

**c. Introduction:**

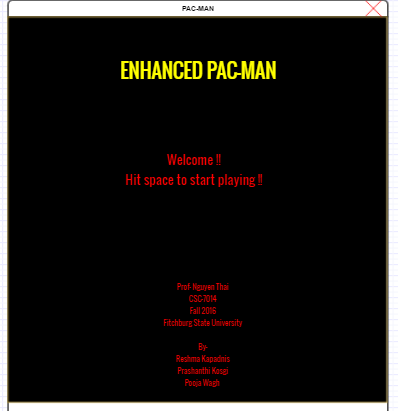
This project is a simple Pac-man game. Where user play the game using arrow keys on keyboard. In the process of the game, PAC-MAN will move inside the maze boundaries’ and eats the coins and gains the points. PAC-MAN will gain extra points for power booster’s and looses life with ghost collision. User has to play the game in a given time span. On the last page, the score is to the user’s gain points and quit and restart options are for quitting and restarting the game again.

**d. Body:**

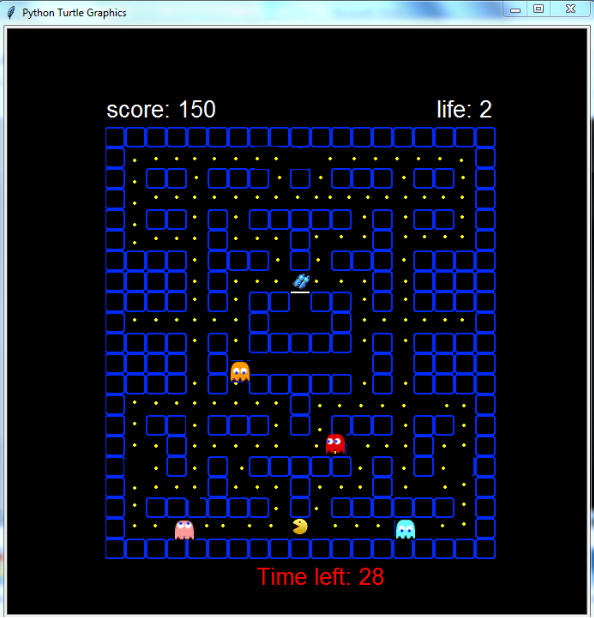
**Project overview:**

Enhanced Pac-man is a game.

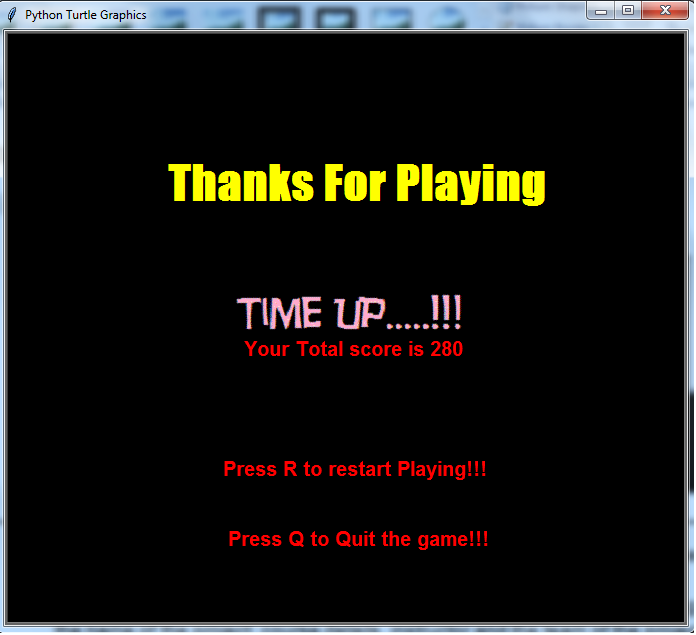
1. The game starts with the main screen launch with sound and instruction “Hit space to start playing”, if the player hits the space bar then the player lands on next game screen else player remains on the main screen.



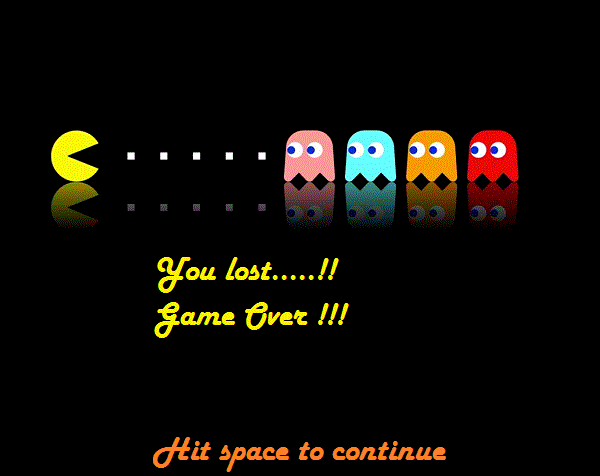
1. Once player hits the space bar, player lands on the main screen.



1. PAC-MAN can move in all four directions (up, down, left, right). PAC-MAN moves will be controlled by the player using keyboard arrows (up, down, left, right keys).
2. PAC-MAN will only move inside the maze boundaries. PAC-MAN moves in the maze according to user input directions and eats the coins in the maze along with the boosting powers. PAC-MAN has three lives.
3. If PAC-MAN collides with ghosts or if PAC-MAN does not eat the complete coins in given span of time then PAC-MAN looses’ lives one at a time. Remaining number of lives will display on the screen.
4. Four ghosts (are in red, pink, orange, blue colored) move inside the maze. Each enemy move in its own path. If PAC-MAN collides with ghosts, PAC-MAN loses’ his life
5. “Boosting Power” and “Ghost Power” are the two special powers present in the maze along with the coins. If PAC-MAN eats the “Boosting Power” then the score increases by 100 points, PAC-MAN gains extra time and speed of the PAC-MAN increases.
6. If PAC-MAN eats the “Ghost Power” then the score drops to “0”, PAC-MAN freezes for a few seconds and the speed of PAC-MAN decreases.
7. “Cherry Power” increases the time for the PAC-MAN.
8. “Apple Power” gains extra points.
9. “Strawberry Power” provides extra power for PAC-MAN and clock provides extra time to play the game but it reduces 50 points from your score.
10. The PAC-MAN score and time remaining displays on the screen.
11. If the PAC-MAN looses’ life then the remaining lives are visible to the player on the screen.
12. The speed of PAC-MAN increases simultaneously with time.
13. If the player does not complete the game in the time, then “Time up screen” visible to the player. This page even consists of instructions for the player to restart the game or to quit the game.



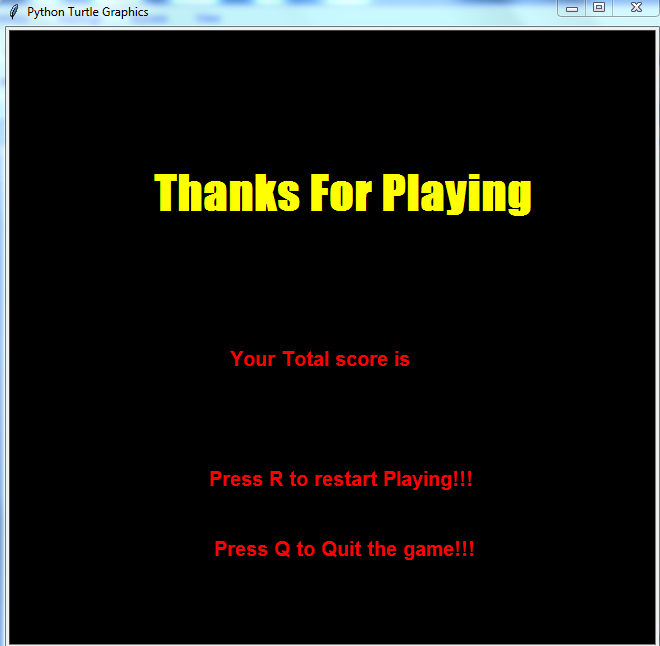
1. If the player loses’ the game then “You lost screen” is displayed to the player with the instruction to hit the space to continue. Once the player hits the space bar it directs the player to next screen that is score screen.



1. “You win” screen display when the player wins the game (finish all the coins within time limit) and instructs the user to hit the space bar to continue. Once the player hits the space bar it directs the player to the score screen.



16.  If the player hits the space bar it displays a new page with the score and instructions to restart and quit the game.



      17. Player lands on the main screen on hitting the restart key “R” and exits from the game on hitting the quit key “Q” (no caps lock key on).

**e. Project requirements:**

**1. Functional Requirement:**

1a) Main screen.

1b) Play the game.

1c) Move the PAC-MAN.

1d) Display the score and result

1e) Exit the game.

1f) Restart or quit the game.

**2. External Interface Requirement:**

**Keyboard:**

Project keyboard is the only one external interface we used. The main purpose of a keyboard is an input device used for Pac-man movements in the maze. Basically, we are using only the four keys on the keyboard. The left arrow is to move the Pac-man towards the left side, the right arrow to move the Pac-man towards the right side, the up arrow is to move the Pac-man upwards in the maze and the down arrow is to move the Pac-man downwards in the maze. Also we are using keys like “R” and “Q” for restarting and quitting the game.

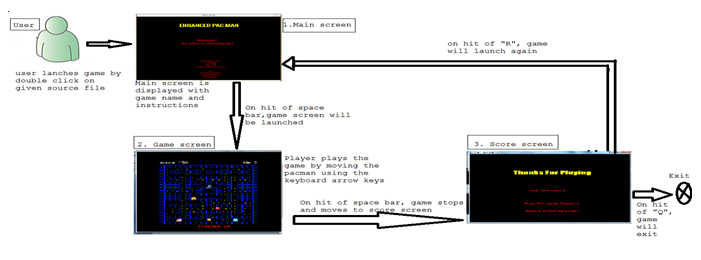
**3. Performance Requirement:**

**3a) Static Requirements:** One User can play Pac-man game at a time.

**3b) Dynamic Requirements:**

We will update the moves on monitor (quickly) like when user tries to move the Pac-man left, right, forward, backward. So that user finds that the packman is moving according to him/her. Booster power and ghost power will also appear in the maze with its specific image and after eating this Pac-man will get different types of energy

**f. Project design and analysis:**

**Use Case Definitions:**

1. Double click of the source file will be launching the game with a main screen on pc/laptop. “Hit space to start playing” this instruction is given on the main screen with music along with the name of the project, course details, instructor and the team of the project.
2. If the player hits space bar, than player lands on next game screen else, player remains on the main screen. In this game screen, PAC-MAN, ghosts, maze, ghost power, ghost power, score, lives, time left are visible to the player to start the game. The power balls are also visible in the maze.
3. Player starts the game from this screen. Maze is designed with a pathway for the PAC-MAN to move in the maze. PAC-MAN can move in all four directions (up, down, left, right).  Player moves PAC-MAN using keyboard arrows (up, down, left, right keys).
4. PACMAN will only move inside the maze boundaries. The ghosts will move in a specified path all over the maze. The score display on screen and instructions to restart and quit the game on hitting the space bar.
5. player will be directed to main screen on hitting the restart key ”R”,  If the player hits the quit key “Q”, the player exits from the game.

**UML class diagrams and add descriptions:**

**Pacman**

maze: Maze

pacman: Pacman

shape: Shape

score: int

life: int

playtime: int

boostPower: int

speed: int

coinlist: int





+nextpage()

+make\_turtle()

+isCollision()

+speedUp()

+ghostmove()

+turnleft()

+turnright()

+turnup()

+turndown()

+timeup()

+quitgame()

+restartgame()

+clearscreen()

Displays next screen.

Creates object.

Collision between pacman and ghosts.

Increases Pacman moving speed.

Movements of ghosts.

Pacman moves left direction

Pacman moves right direction

Pacman moves up ward direction.

Pacman moves down ward direction.

Displays when given time is completed.

Quit’s the game.

Restart’s the game.

Clears the screen.

Maze in which the game is played.

Player in the game.

Shapes of objects.

Score of the player.

Number of lives of pacman.

Time provided to play the game.

The boost power gained by pacman.

Speed of pacman

No of coins in the maze.

**Software development:** The language and tools used to develop this project are

Language: Python 3.5.2 (Anaconda3 4.1.1).

Tools: Spyder 2.3.9.

**Project Operation:**

**Instructions on how to use the project:**

Prerequisite:

1. Anaconda image with python version 3.5.

Instructions

1. Download the “Source code” zip file and extract all the files including images.
2. Double click on the “source code”
3. System will launch the game.
4. Press the “space bar” to start the game.
5. Play the game using keyboard arrow keys.
6. Once the game is finished score screen is displayed with instructions to restart and quit the game.
7. Press “R” to restart the game.
8. Press “Q” to quit the game.

**Features of the project:**

**Movement of Pacman:** PAC-MAN will only move inside the maze boundaries according to user input directions. PACMAN moves in all four directions. Player controls the Pac-man’s moves using keyboard arrows (up, down, left, right keys).

**Movement of ghost‘s:** There are four ghosts in the game, they are red, pink, orange and blue. All the four ghosts move inside the maze in a specified path.

**Collision:** If PAC-MAN collides with ghosts, then the Pac-man freezes for a span of time and looses life.

**Powers:** If PAC-MAN eats/collide with below listed powers they will show some effects like increase/decrease in score and time count.

“Boosting Power”: score increases by 100 points, Pac-man gains extra time and speed of the Pac-man increases.

“Ghost Power”: score drops to “0”, PAC-MAN freezes for a span of time and the speed of Pac-man decreases.

“Cherry Power”: increases the time for the Pac-man.

“Apple Power”: gains extra points.

“Strawberry Power”: provides extra power for PAC-MAN

“Clock”: provides extra time to play the game but it reduces 50 points from your score.

**Time left:** Displays the time left on the screen to finish the game.

**Score:** Calculates the score of the player and displays on the screen with instructions to restart and quit the game.

**Results**: If the player loses’ the game then “You lost screen” will display to the player with the instruction to hit the space to continue. Once the player hits the space bar it directs the player to next screen that is score screen.

“You win” screen will display, when the player wins the game and instructs the user to hit the space bar to continue. Once the player hits the space bar it directs the player to the score screen.

 If the player does not complete the game in the time, then “Time up screen” will display to the player. This page even consists of instructions for the player to restart the game or to quit the game.

Restart game: If the player hits the restart key “R”, then the player will be directed to the main screen

Quit game: If the player hits the quit key “Q”, the player exits from the game.

**Key issues and challenges of project development:**

**a. Challenges:**

1. Designing the path for ghosts and Pac-man.

2. Designing the program for eating coins.

3. Handling the collision of Pac-man and ghost’s.

**b. key issues and Challenges :**

Building the maze and moving the Pac-man within the maze boundaries.

Completing project design, coding, documentation and testing within the project time line considering other classes.

c**. Bugs not able to fix:** Perfect movement ofPAC-MAN in maze.

**d. Recommendations:**

The maze layout was complex, it could have been easy for movement of pacman and ghost, if layout was designed simple with minimum blocks.

**e. Conclusion:**

The project has been completed and tested as of on 12-05-2016. It was a great experience learning a new language “Python” and developing a gaming application and also we have learned the process of project implementation like requirements, design, documentation, coding, project presentation, testing, etc.

**f. References :**

* Class notes.
* Professor Thai, Nguyen guidance. (CSC7014 the Practice Computer Program)
* [***https://en.wikipedia.org/wiki/Pac-Man***](https://en.wikipedia.org/wiki/Pac-Man)
* Resources from the internet.
* Online literatures.