CG Mini-Project Report.

Topic - **PACMAN**

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

MEMBERS:  
 YASH SARANG  
 OM GAYDHANE  
 MANAV PAHILWANI  
 SURABHI TAMBE

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

Table of contents:

1. Abstract

2. Introduction

3. Requirement analysis

4. Implementation

5. Result

6. Conclusion

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

Abstract

The following report consists of a detailed description of our CG Mini Project. We have created a classic arcade game - ‘Pacman’ in which we have implemented Computer Graphics taught in our current Semester as well as basic gaming algorithms.

To make it a bit user-friendly and less complex, we used a high-level coding language - Python over a low leveled language like C.

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

Introduction

We started by importing the turtle package in python which would help us in implementing CG in our code. Following by turtle, we imported the choice module from random and the vector and floor modules from freegames library.

The start of the code represents the initial state of the game/game objects. Then we have multiple user-defined functions 1. square( ) - which fills the tile with the x and y coordinates.

2. offset( ) - which returns the next index in which the

pacman / ghosts will be moving.

3. valid( ) - which returns True if the next tile is valid

for movement.

4. world() - used for Drawing the world, the tiles, and the score

points.

5. move( ) - used For controlling the movement of all the pacman and ghosts.

6. change() - change the direction vector of pacman if its valid.

Then we use the setup( ) function from turtle to set up the GUI window for our game.

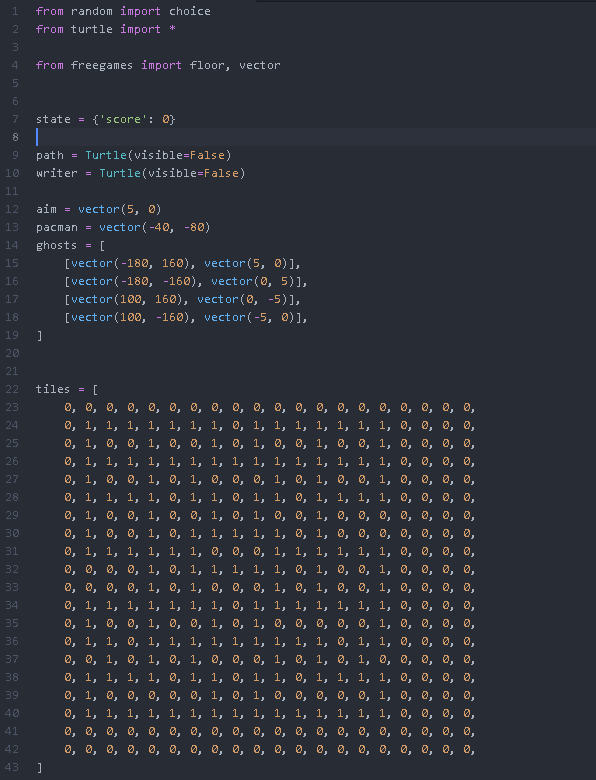
The write( ) function is used to display the score in which state['score'] is used to input the current score of the user.

And at last, the listen( ) function is used to read the inputs of the user and the code acts according to the input.

The game ends when the Pacman gets hit by a ghost, as soon as the Pacman hits a ghost the move( ) function stops, and hence, the game is over.

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

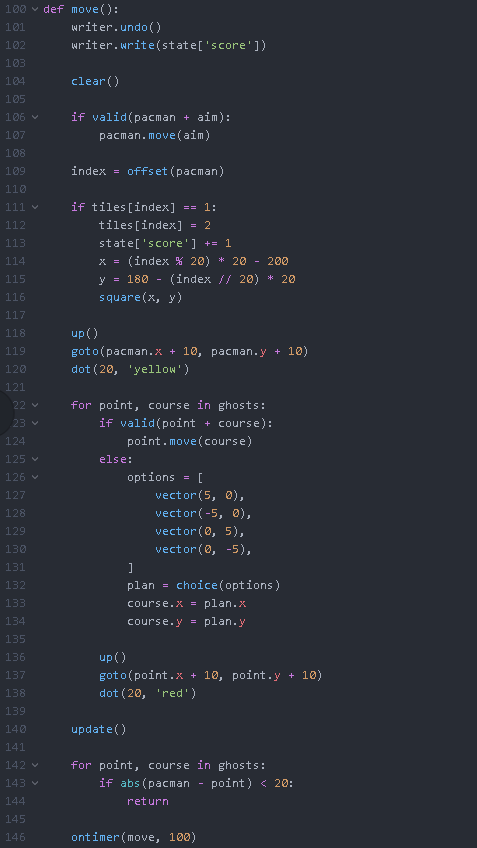
Implementation (Code Snippets)

Initialization   


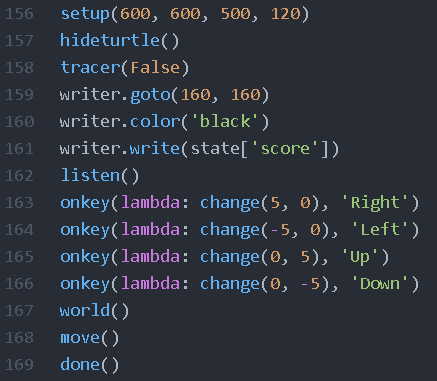
World creating function



Movement function



Setting up the GUI window and Recieving user input



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

Requirement analysis (s/w and h/w)

HARDWARE-

Computing Device (PC or Laptops)

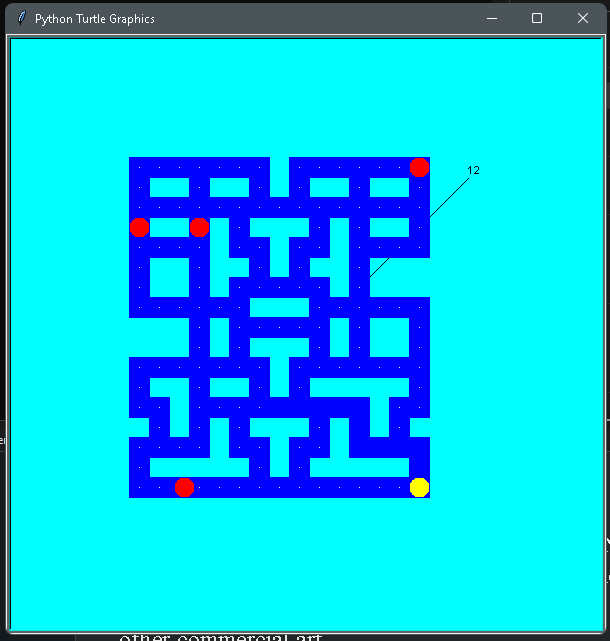
SOFTWARE-

Python 2.5 or newer

(imported with pygames and free games packages)

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

Snapshots



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

Conclusion

As we have implemented CG in our PACMAN game, we can use CG in various day-to-day programs from movie making, video game development, scientific modeling, designing for catalogs, and other commercial art.

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