GAME AI CS 5150
HOMEWORK 2
Sushruth Bhandary

CODING QUESTION EXPLANATION AND INSTRUCTIONS ON RUNNING THE CODE

- The main code is in a file named ‘pacman.py’ which is a basic pacman game.
- All the coding requirements in part A, B and C are in this file
- For part A, I’ve used multiple PyGame functions in the program
- For part B, you can run the file using the command below which will run a basic paceman implementation where the **pacman can move with the arrow keys** and the monster is stationary

  command : **python3 pacman.py**
- For part C, I have used command line arguments to give the monster some steering behaviors.The monster will either seek, chase or flee depending on the command line arguments
  1. Seeking: the monster will try and reach the paceman. The pacman can move while the monster is seeking it and it will change its trajectory accordingly. Given below is the command

     command: > **python3 pacman.py --behavior seek**
  2. Chasing : It uses the seek function but it doesn’t seek the present position of the pacman. It seeks its future position

     command: > **python3 pacman.py --behavior chase**
  3. Fleeing : It runs away from the pacman when the pacman is in a radius of 200 pixels. It stops fleeing when its farther than 200 pixels

     command: > **python3 pacman.py --behavior flee**

SOLUTION PART A

The Snake AI game failed to compile due to a few bugs and hence, as mentioned by the professor on blackboard, I shall be explaining my Pacman game code instead.The code for the game can be found in the file pacman.py

I have used the pygame library for building the GUI and have also used the argparse library to pass command line arguments to the program. Given below is an explanation of the basic functions that I used.

1. **pacman(x,y,rotation_angle)**
   - This function takes the x and y coordinates and and places the pacman object at that position on the game window using the blit function
   - It also takes another argument called rotation_angle and makes th e pacman face in that direction of the movement using the pygame.transform.rotate() function
2. **monster(x,y,is_eaten)**
   - This function places the monster object at the x and y coordinates passed to it.
   - If the is_eaten argument is false, it doesn't display the monster
3. **printMsg(text_string,x,y)**
   - This function prints a text block on the game window.
   - I've used the text block to show the distance between the pacman and the monster at any given point which I shall be using for the flee function
4. **seek(x,y,mx,my,mvel)**
   - The seek function is used to steer the monster in the direction of the player.
   - It takes the coordinates of the monster and pacman and the angular velocity of the monster
   - It first calculates the angle between the monster and the pacman using basic trigonometry. Using the angle found, it calculates and returns the x and y velocities required to make the monster move in the direction of the target

5. **flee(x,y,mx,my,mvel,flee_radius)**
   - Flee takes in the same arguments as seek but takes one additional argument called flee_radius.
   - So the monster starts running away from the player if the player is within the flee_radius. It keeps running until it's out of the flee radius and then it stops
   - The logic behind it is exactly the same as seek but the directions will be opposite

6. **calcDistance(x1,y1,x2,y2)**
   - It take the coordinates of the monster and the pacman and returns the distance between them
7. **Main**
   - The main function initializes a few variables like the positions and velocities for the characters
   - It consists of a while loop and doesn't exit the loops until the the game window is closed or 'q' is pressed
   - Within the main while loop is another for loop which polls for keyboard inputs (player input). Depending on the keys pressed, it either moves the pacman or quits the game
   - Once the player input is taken, it's time for the computer to make the move. Depending on the command line argument passed for the behaviors, it either calls the seek, chase or the flee function. The chase function is basically a seek function but instead of the target coordinates, we pass the future target coordinates.

SOLUTION PART C

Q. Whats the difference between seeking and chasing ?

A. Chasing and seeking are similar to some extent but are still different in the sense that seek tries to steer the object towards the target and chase tries to steer the object towards the future position of the target. So chase is basically seek, but for the future position of the target. From the code point of view :

```
seeking :  seek(target_x, target_y)
chasing :  seek(target_future_x, target_future_y)
```