“THE GAME” Project Report

It is a top-down shooter where you get more points for keeping the enemies alive and lose points for wasting bullets

Motivation:

I love video Games and always wanted to make one, This is my first try at making a working real time one

The functioning of each class in the order of their conception:

1. Cell:

   It has your basic x, y coordinates a surface and 4 walls on the edges of the rectangular cell
   The show function blits a wall if the wall is active for the cell
   The main function for this is “next”, this if possible will randomly select a cell in the 4 sides and change the walls between the two cells to inactive, If all the four sides are already visited then it will revert back to the previous cell
   This will continue till the selected cell comes back to the starting cell
   I also randomly remove all walls from a cell to have round about passages and improve the mobility
   It is basically depth first search Algorithm

2. Player:

This is a player class which will need the maze list it has a surface which can be moved by using the arrow keys and WASD keys the update function prevents the player to cross a wall or leave the game board
The rotate function rotates the surface toward the mouse at its current position
It also has a shoot function which will return a bullet object

3. Camera:
   Its takes the height and the width of the camera and it has a position x , y variables
   We make a surface of height, width dimensions and offset all the sprites drawn on the screen
   The update function keeps track of the player and moves the camera coordinates according to it, not allowing the camera to move out of the play area

4. Bullet:
   A bullet is fired whenever the player clicks the Mouse but be aware that the bullet costs 10 points
   The bullet is pretty simple as at the time of its initialization it will store the direction of the mouse and then in each update move the sprite by that
   The update function keeps track of the cell through which the bullet is travelling and if there is a wall on the edge and the

bullet collides with it then the bullet is kill()

5. Enemy:

It is a class similar to player which is not allowed to move past walls
It has 3 different types of AI:

One which I call mouseA, it will move left if possible otherwise it will move forward, if that is also not possible then it will move right

Another is mouseB, which is mirror of mouseA

Then there is the Drone, in this a depth First search algorithm is used with the player as the starting position and the enemy as the end position It will use the last 2 entries in the queue to find the next position of the enemy
(Due to me adding multiple paths between any two points the path may not always be the most optimized path)
The enemies also have the feature that if the enemy is in the same cell as the player, then their speed will increase and they will start to follow the player till the player can again escape to another cell out of their reach, I call this the pounce mechanic

To make the enemy move on a slower speed I have used a (I think) a pretty neat way, whenever it decides a direction to move, It will select a cell, and will not decide a new direction till it reaches a certain part of this cell (which depends on the offset of the enemy)
After that it will again calls its ai and decide on a new direction

6. Game:

It is the master Class which will combine all the others
The Main feature are the enemies and bullets sprite group
It has a maze function which will create a list of cells to be used by all others
It has a get event function which receives the pygame events and also keep track of 2 user defined comments SPAWN and SCORE, they spawn a new enemy in the farthest corner of the player and increase the score by number of enemies on the board respectively
Then we have show function which blits all the surfaces on the screen (after applying the offset due to the camera)
In the update function it will update all the sprites to there new location and also keep track of any sprite collisions

It also has a menu function which just displays the name of the game (which is also “the game”)

7. There is a settings file which is used to change the parameters of the game like screen size, map size and also define all the global variables

I also made a getClick() function which is used to wait till a click is made on the screen and also return the pos of the mouse click(It is inspired from a function in simplecpp and is very useful when Debugging)

THE MAIN PROBLEMS:

The major problems I had while making the project were:
Trying to download and make sublime work

The rotation of the Sprites were permanently distorting the images which was difficult to resolve

The drone AI started acting up due to me adding multiple paths that is why I changed the target and the starting point

The implementation of cameras was also very difficult

RESOURCES:

Making games with python and pygame by Al Swiegart

Articles on realPython

The kids can code youtube Channel

CONCLUSION:

I had a lot of fun while making this project and I learned lot

I wanted to improve the graphics of the game but I am very poor in the design area and was only able to make only decent player and enemy sprites

I Hope you like the game and if possible point out any bugs I might have missed

THANK YOU