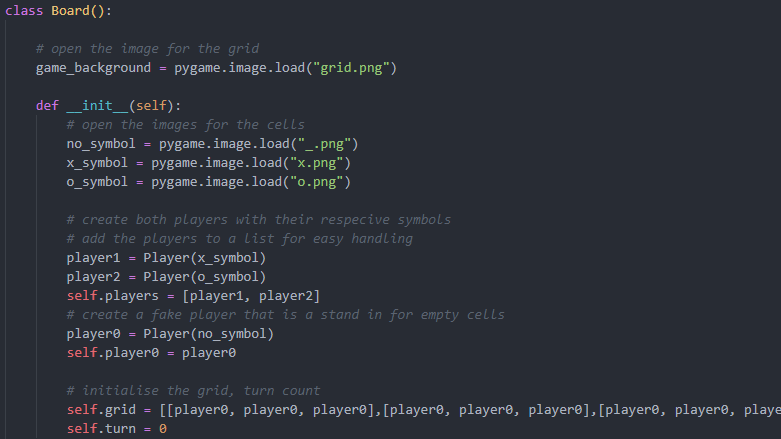
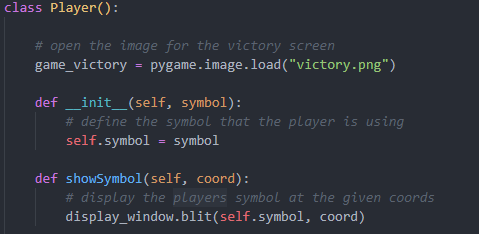
Video Demonstration:

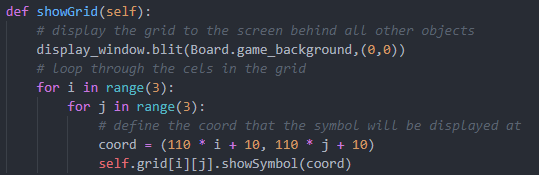
<https://www.youtube.com/watch?v=P6BE6eUhzfI&feature=youtu.be>



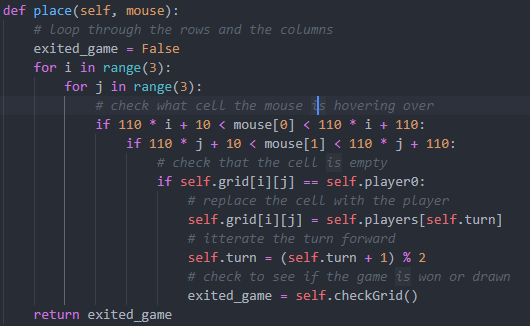
The board class defines the logic behind the game. It is also the access point to the player objects. It has the class variable of game\_background which is used so that any board can use that image and it does not have to be reopened and stored for each board. The players are created in the boards constructor by first loading the symbols that they will use, then defining them and finally adding them to a list. Their is a turn count which switches between 0 and 1 so it can be used to access either the first or second item in the player list aka player 1 and player 2. The grid is a nested list where each nested list is a column. The player 0 is created so stand in for an empty cell. This is because the displaying of the cells is done though and for loop and each object in each cell runs its own display function.



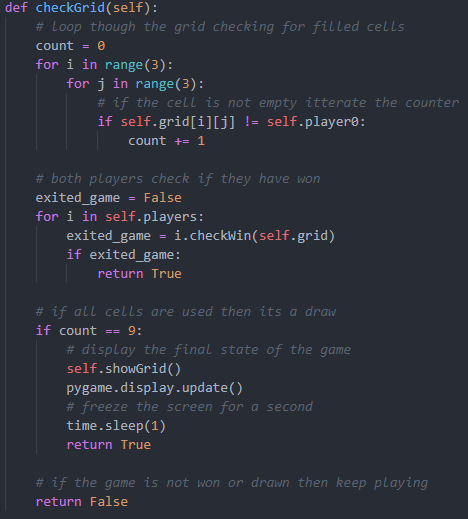
The player class defines the symbols that each player has and runs the code required to display and check for victories. The constructor only needs the symbol variable. The game\_victory image is a class variable for the same reason as the class variable used in the board class.



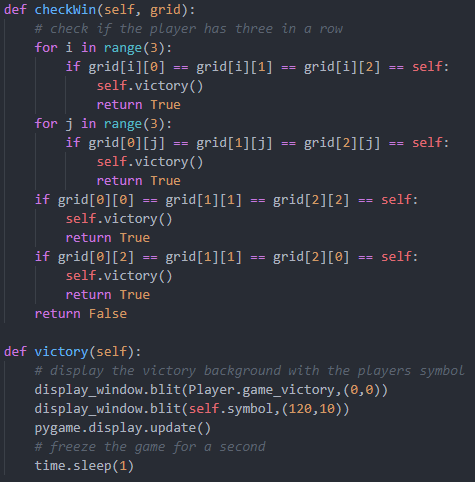
The symbols are displayed by iterating through the grid and running the show symbol on each object. The co-ordinate represents the top left corner of each cell as despite the symbol being smaller than a cell there is a white border around each of the images.



If the players press down on the mouse it will run this code. It checks where the mouse is by iterating though the cells. If it finds the mouse in one of these cells it will check that the cell is empty. If the cell is empty it will replace the player 0 object with the player object of who’s turn it is. The turn counter is then flipped by adding 1 and modulating by 2 to bring it back to a 1 or 0. This changes the players turn while still allowing easy access of the player objects from the players list.



There are two possible reasons to stop the game. The first is a draw. This is detected by iterating though the cells and checking if they have been replaced by a player object. If every cell has it will update the display and freeze for a second so that the players can see the final arrangement of the grid. The other possibility is a win which is handled within the player classes so that they an run their own victory screen. The draw check is performed after the victory check so that you can still win a game on the final move.



The first if statement checks each column for 3 cells of the players kind. The next checks the rows and the final two check the diagonals. If one of these conditions is found to be true it means that the player has won. This runs the victory method which will display a victory screen with the players symbol overlaid in the top middle cell. The game is frozen for a second so the screen can be seen. The return value is true which will pass back though to the main game loop breaking the while loop and bring the players back to the home screen.