Explaining the code in sections

# Section 1

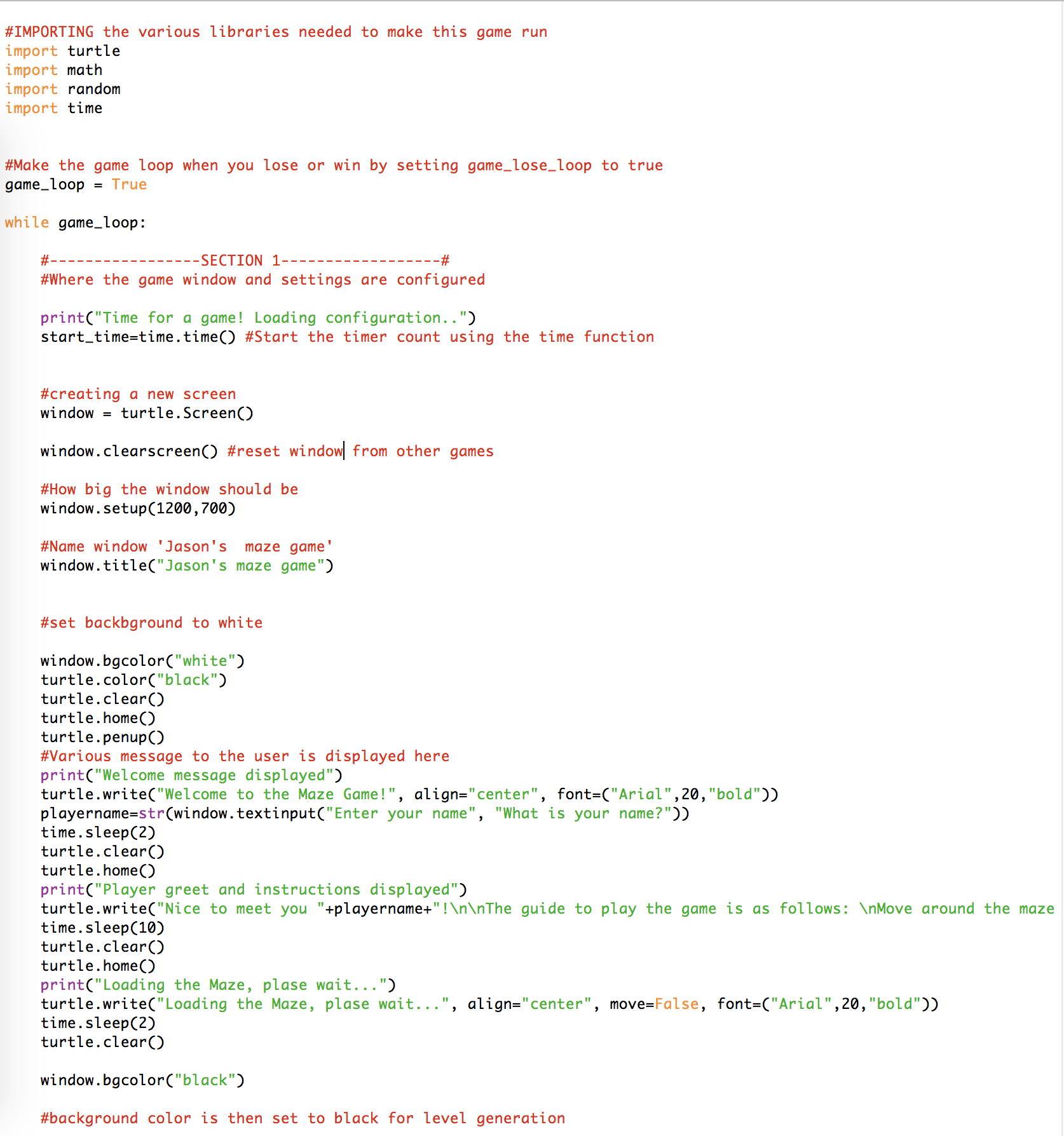
Importing the various libraries needed for the code to function.

Turtle – To draw objects on screen

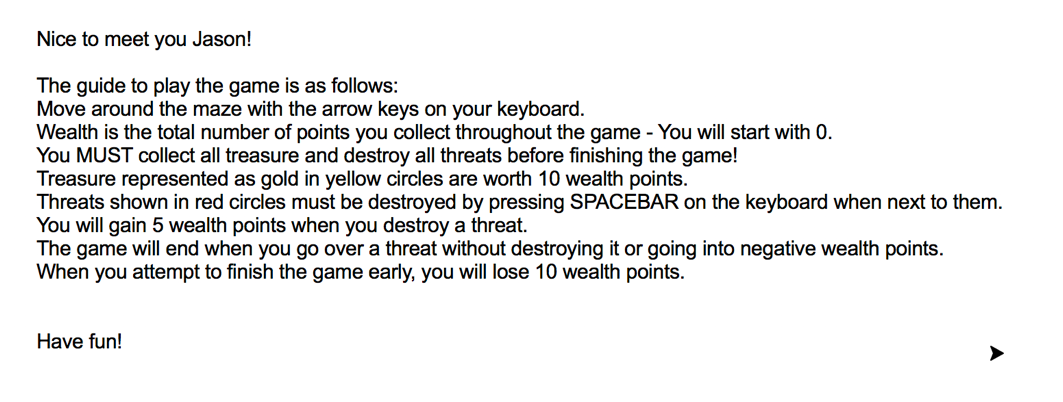
Math – To calculate distances between a player and object

Random – To randomly choose a level

Time – To calculate how long a user has spent in the game



Setting a game loop by default to true means the game will always keep playing unless set to False.

ddd

Once these messages are displayed, all writing is cleared and the background color of the window goes to black.

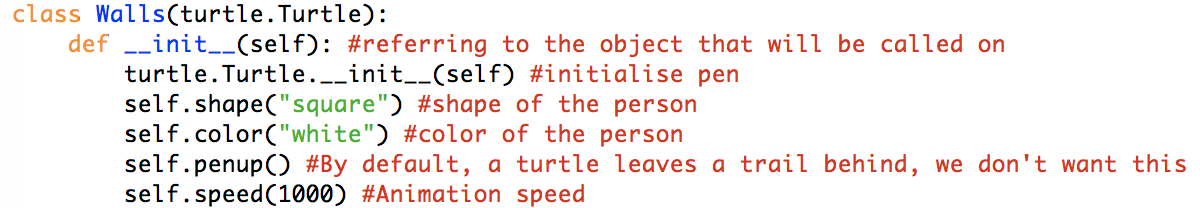
This is where the various messages are displayed to the user. The welcome message and other messages are displayed in Arial font size 20.

This is the first major section of the code. The window settings are defined and titled as “Jason’s maze game”. It is then given a white background for various messages to be displayed to the user. The user is also asked for their name.

# Section 2 – Defining different classes to be used in the game

## The wall class code

The wall class will be drawn by a turtle. It is defined to be a square shape and have a colour that is white. When it is being added to the screen, the pen will be up so that a trail will not be left behind.



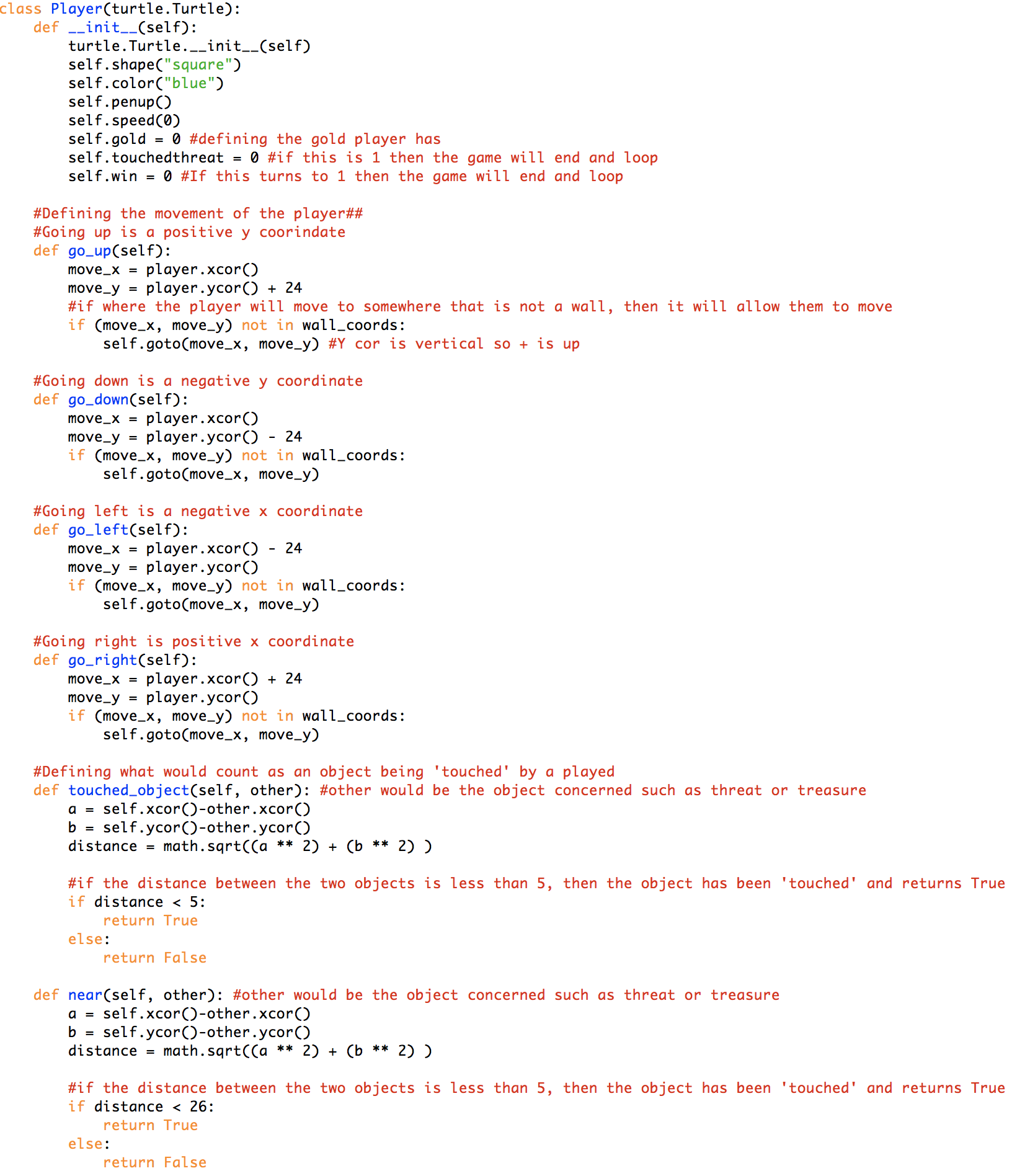
## The wall class output

The output of the wall class is simply seen below



## The player class code

The player object is defined as being a blue square. It also has the attributes gold, touchedthreat and win.



For both touched\_object and near routines they have similar objectives. The maths behind it is that a distance can be calculated when comparing the current player coordinates to the object in question, such as a threat or treasure.

An object is defined as ‘touched\_object’ by the code when the player’s distance is less than 5 coordinates away.

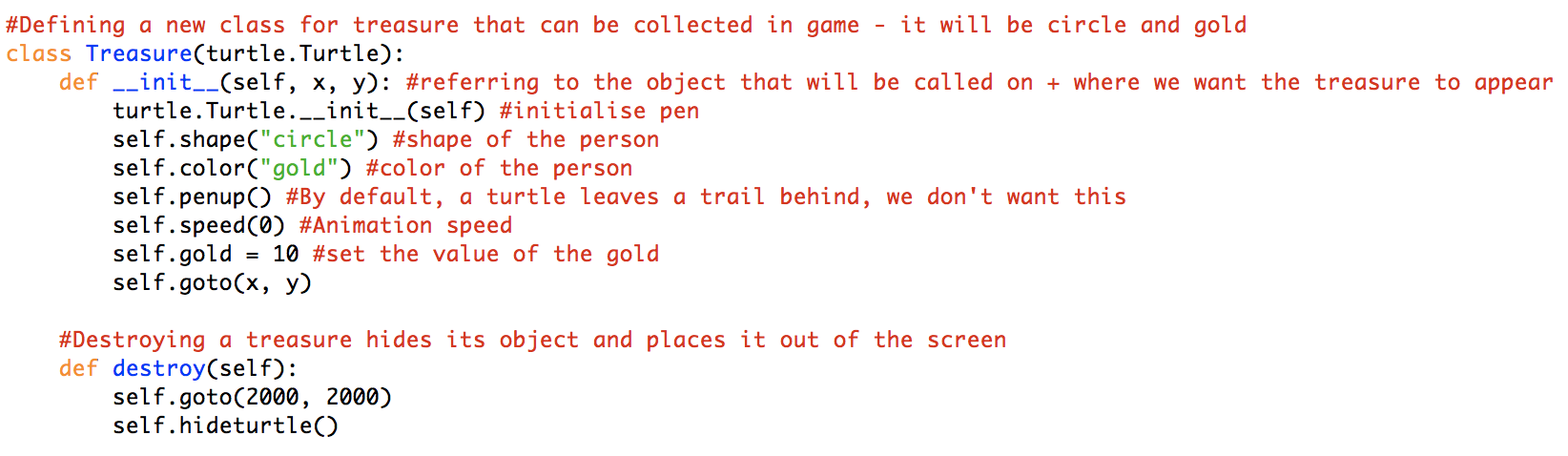
An object is defined as ‘near’ when the player’s distance is less than 26 coordinates away.

Each go\_x routine has either a x or y coordinate change to the player. These routines will be called upon when the player presses any of the arrow keys which will be seen later.

Outputs of the player will be seen later.

## The treasure class code

The treasure class code defines the object and characteristics that a treasure item on screen would have.



The destroy routine within the treasure class is when the treasure would be removed from the players view

The treasure object is defined as being a gold circle.

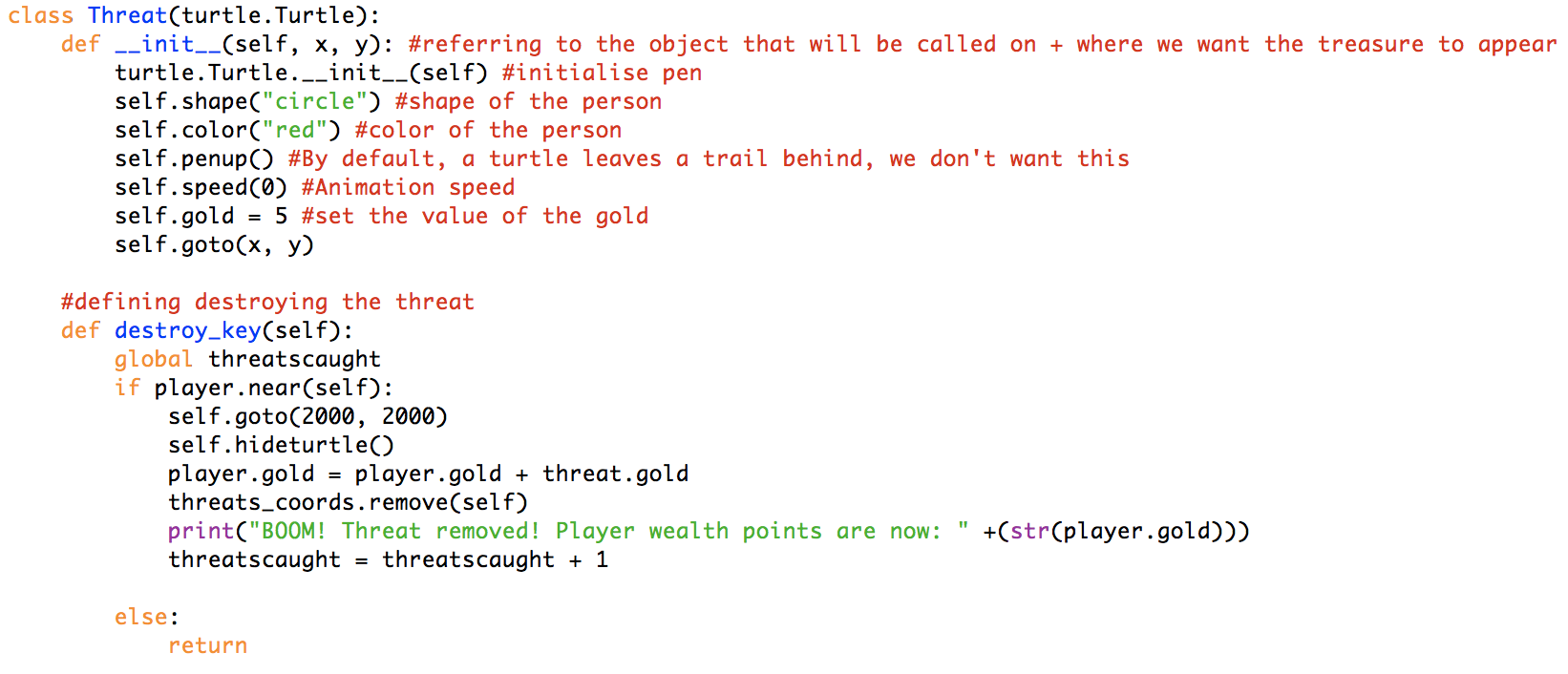
### The treasure class output

The output of when the treasure class is called is seen below.



## The threat class code

The threat class code defines the object and characteristics that a threat item on screen would have.



The destroy routine within the threat class is when the player is near the threat, presses the spacebar and therefore triggers this routine to occur

The treasure object is defined as being a red circle.

### The threat class output

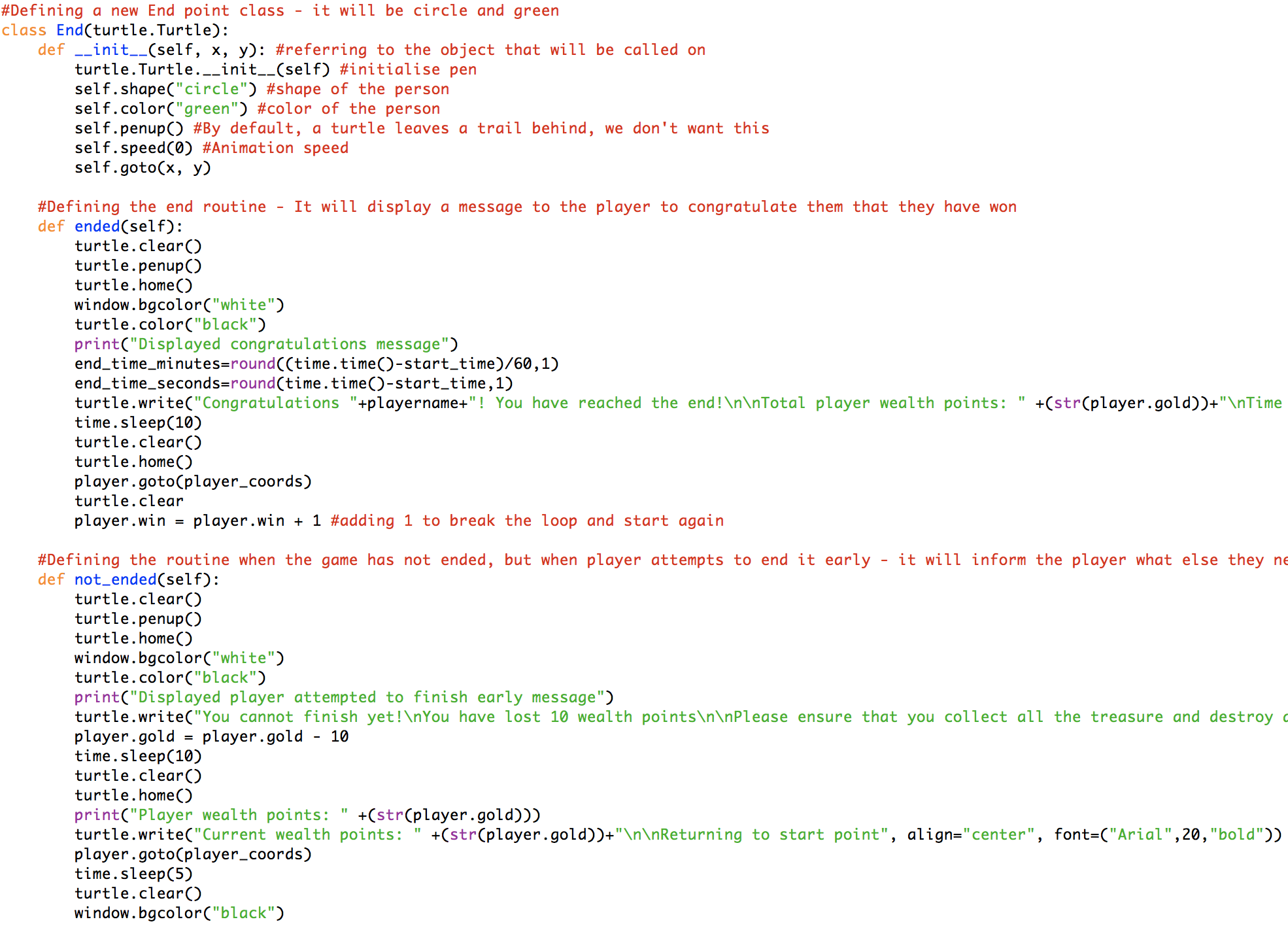
The output of when the threat class is called is seen below.



## The end class code

The end class code defines the object that would be see by the player. The end class also defines the routine that runs when the game ends and when the player attempts to end the game early.

The treasure object is defined as being a green circle.



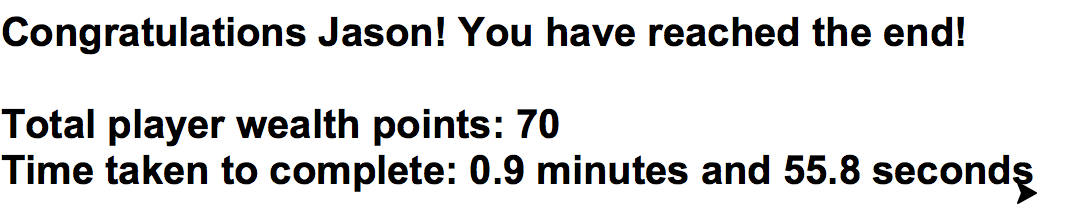
When the player tries to end the game early, the screen will turn white and let them know that they can’t finish. (to be below)

The player.win status is also updated to 1, which allows the game to end.

When the game ends, the screen will be cleared and a congratulations message will be displayed (to be shown below)

### The end class outputs

#### Ending a game completely

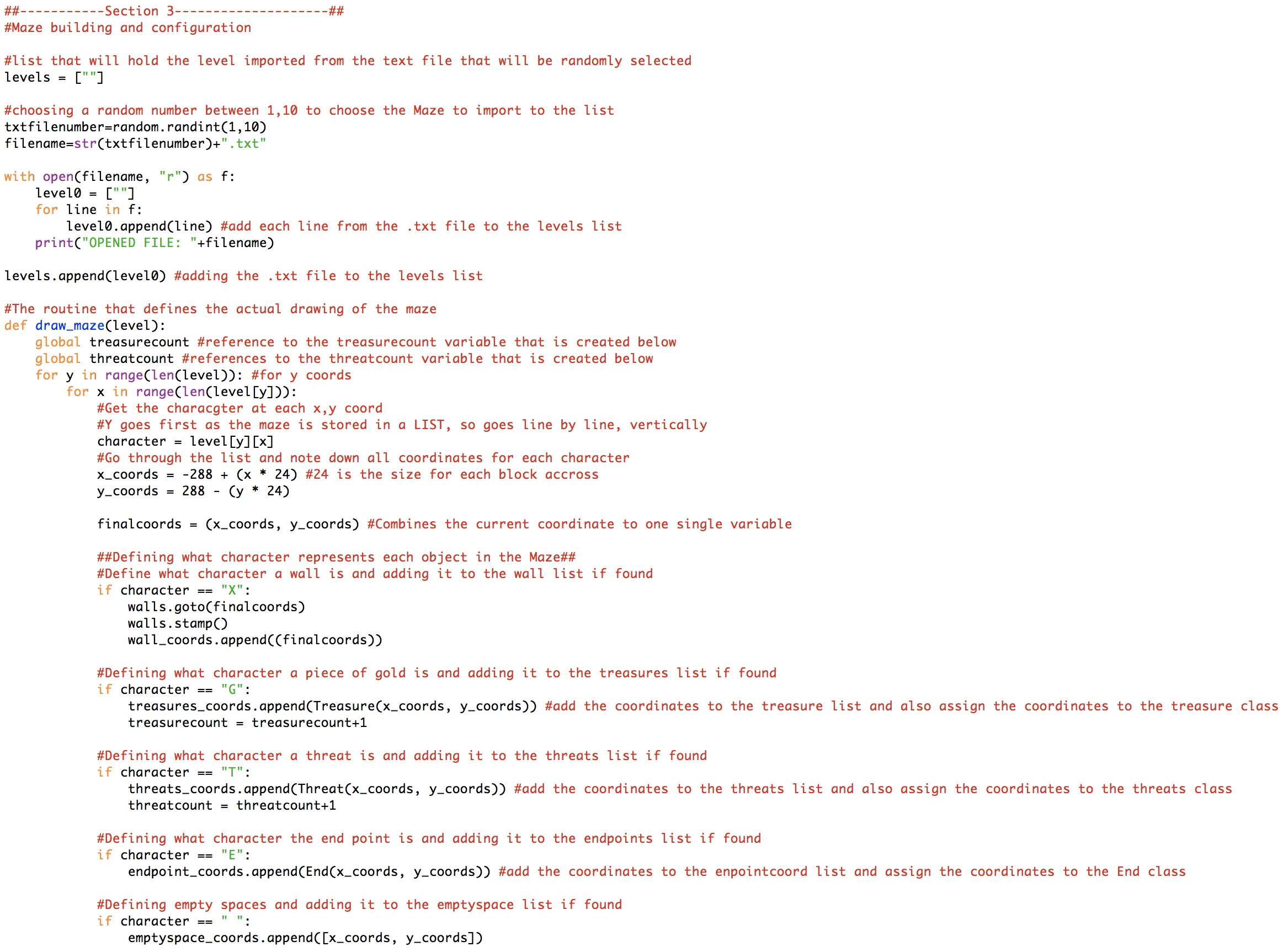


#### Ending a game early



# Section 3 – Maze building and configuration

Within this section is where the maze level and definition of objects comes to place. Each level is configured within their own text file and is composed of 25 characters horizontally and vertically. The code then randomly chooses a number between 1 and 10 (which is currently how many levels there are) and it is loaded. Also, each individual character within the text file are defined here.



This section defines what character within the textfile belongs to what class. The comments explain what each character is

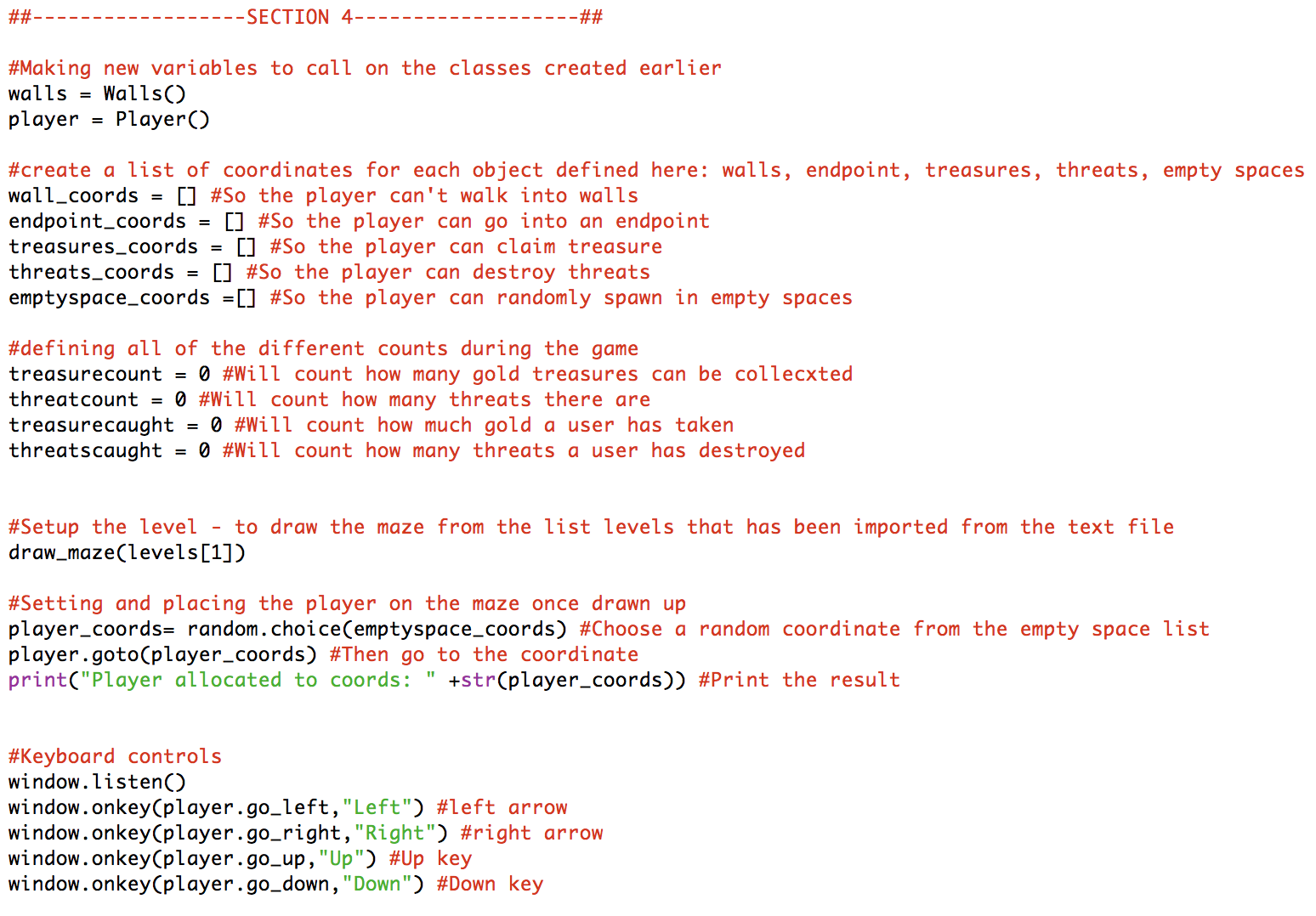
Defining the coordinates in the level. Each coordinate ranges from x -288 – 288 and y -288 - 288

For every line within the text file, add this to the level0 list.

The treasure object is defined as being a green circle.

# Section 4 – Defining lists and trackers

Defining a variable for the walls class and player class so that they can be called on



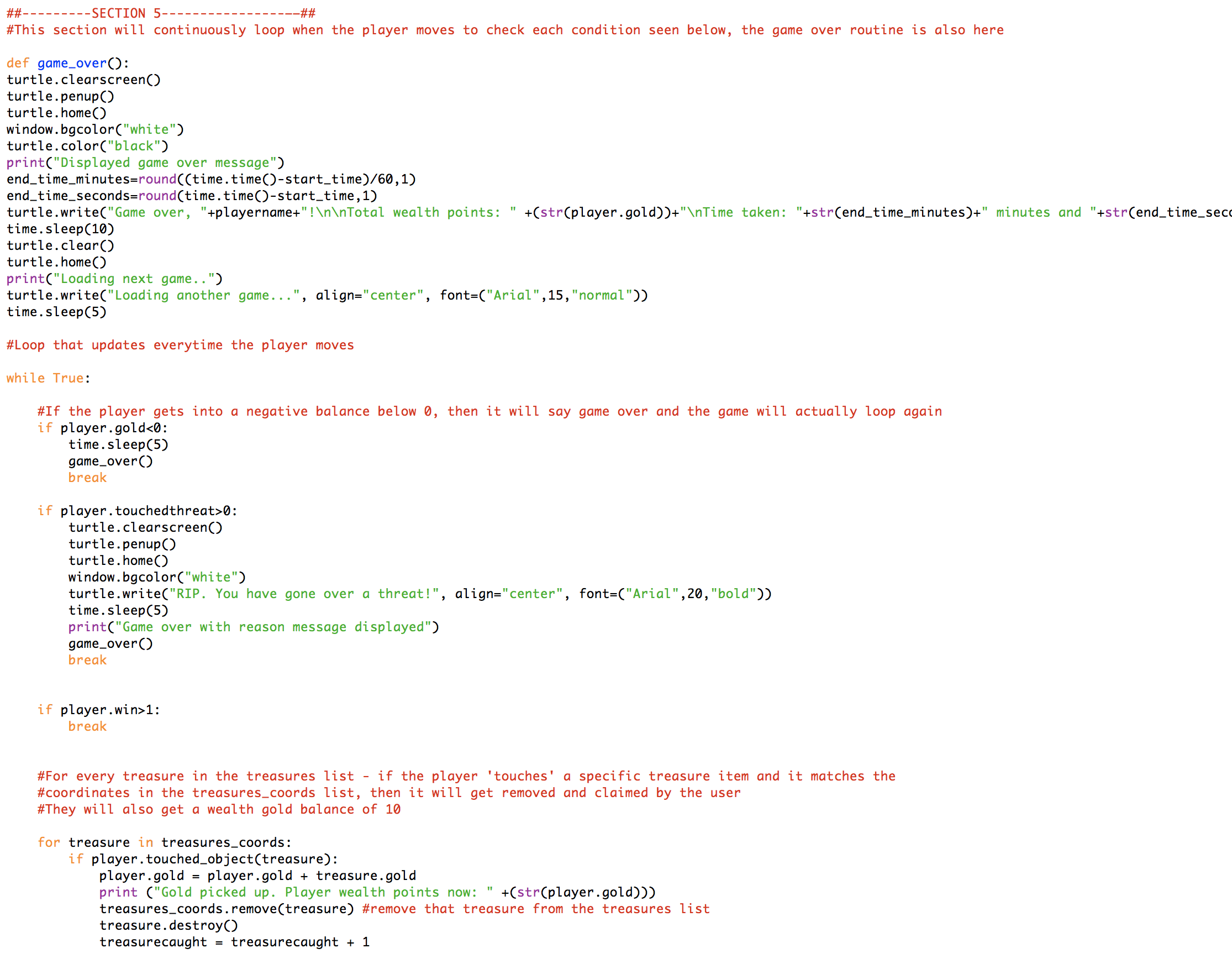
Setting the keyboard controls for the player and calling the routine when the key is pressed

Set the player to a random coordinate within the maze that has been recorded as empty

This section defines all of the different variables that will hold key information about the maze.

# Section 5 (part 1) – Checking conditions to win / lose a game

Section 5 contains a continuous loop which checks against certain conditions before deciding to end or continue a game.

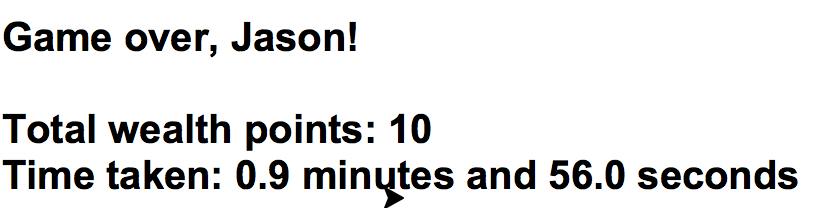


If the player gold goes below 0 to negative or if a player touches a threat the game will end and go to the game over routine. Upon touching a threat, an additional message telling the user that they have touched a threat will be displayed. If the player has won a game, then the code will break out of the loop and start a new game

The game over routine clears the screen and displays a game over message to the user. The output will be seen below

## Outputs from section 5 (part 1)

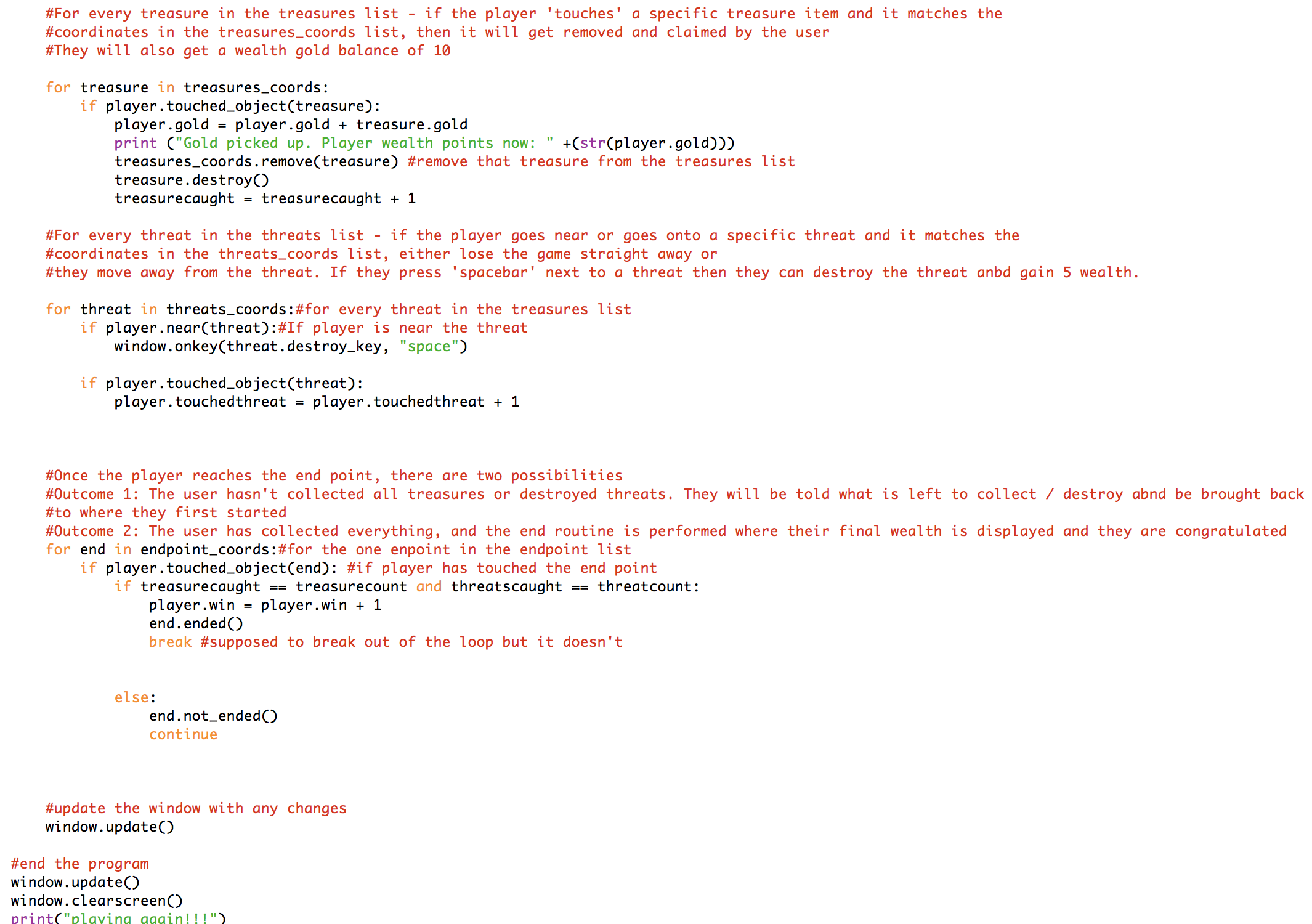
### Game over message



### Going over a threat



# Section 5 (part 2) – Checking conditions to win / lose a game



At the end, the window is updated to reflect changes and when the loop is broken a message saying playing again is displayed in the background python window.

For a player going to an endpoint there are 2 outcomes as seen in the comments. If the player meets the first condition where everything has been collected, the player.win tracker variable will update to 1 and will end the congratulations message. If not then the attempted to finish early messxage will be displayed.

This section checks conditions for both treasures and threats. If a treasure is picked up it runs the treasure.destroy routine which removes it away from the screen.

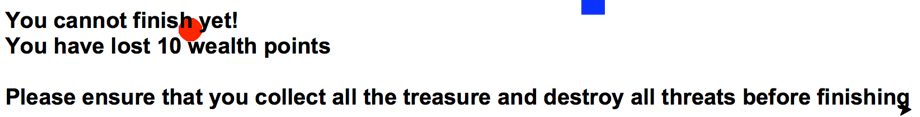
For a threat, if the player is near and presses spacebar, then the threat.destroy routine is run which removes the threat from the screen. If the player touches the screen, then it updates the .touchedthreat tracker to 1 and the game will end.

## Outputs from section 5 (part 1)

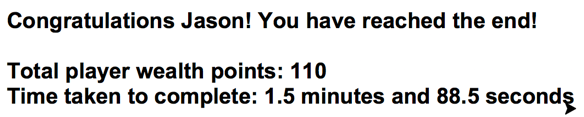
#### Collecting treasure



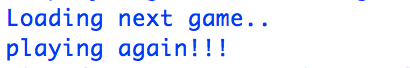
#### Attempting to finish early



#### Finishing the game completely



#### Playing another game



# Classes used in the python game

Walls = Defined the characteristics of a wall in the game

Player = Defined the characteristics of a player in the game

Treasure = Defined the characteristics of a treasure in the game

Threat = Defined the characteristics of a threat in the game

End = Defined the end point in the game and what to do when the game was ended early

# Routines used in the python game

Player.go\_up = Move the player up

Player.go\_down = Move the player down

Player.go\_left = Move the player left

Player.go\_right = Move the player right

Player.touched\_object = Outputting TRUE or FALSE if a player has touched an object

Player.near = Outputting TRUE or FALSE if a player is within distance of another object

Player.gold = Contains how much wealth points a player has

Player.touchedthreat = If this turns to 1, the player game is ended instantly

Player.win = If this turns to 1m the player game is ended and congratulations message shown

Threat.destroy\_key = When the player presses SPACEBAR near a threat, the threat will be removed and 10 wealth points added to the player

End.ended = Defining the routine for the end message to the player and breaking the loop that checks if conditions have been met to end a game

End.not\_ended = Displays the attempted to finish early message to player and reduces player wealth points by 10

Draw\_maze = Defining the routine that draws the maze and what characters defined what objects were where

# Variables used in the python game

Window = Defines the turtle window in the game

Levels = The list that contains the level that is loaded into the game

Txtfilenumber = Will hold a random number between 1 – 10 for the text file to be opened

Filename = combines the txtfilenumber to .txt to make it into a valid file

Level0 = Defines a list that will be added to the levels list when loading the game

Wall\_coords = The list that contains all coordinates of walls in the game

Endpoint\_coords = The list that contains the end point within the game

Treasures\_coords = The list that contains the coordinates of all treasure in the game

Emptyspace\_coords = The list that contains all coordinates of empty space in the game

Treasurecount = Contains how many gold treasures there are

Threatcount = Contains how many threats there are

Treasurecaught = Contains how many items of gold a player has caught

Threatscaught = Contains how many threats a player has destroyed

Player\_coords = Contains the randomly selected coordinates from the emptyspace\_coords list

# Improvements that can be made to the code

1. **Reducing the number of lines within the code**

Currently the total number of lines within the code is 434. There are many lines that could be removed or at least cut down. This could be done by removing some lines which may be duplicate and doing the same thing or placing similar lines together.

1. **Increasing efficiency in the code**

Efficiency within the code could be increased by considering adding repeated lines of code to a routine like most of the code. For example, when the timer was ended, it was duplicated when a game ended or when a person had died. This could be added to a routine next time.

1. **Allowing a player to continue a game or not**

At the end of every game, the player has no control of the starting the game again. It would be nice to allow the player to have the choice to be able to start a new game or not.

1. **Adding messages during the game on screen**

Currently messages such as key events when a player picks up a gold item are displayed within the background window of the game. It would be nice to be able to display this somewhere within the game while the player is running so that they aware of their wealth points and if they die, the reason why.

1. **Adding moving threats**

Threats don’t currently move, but if they did it would add more challenging conditions for the player to make the game more interesting

1. **Adding more rooms and not allowing the player to view the whole level**

Currently the game runs and when loaded it shows the player the whole maze. In the future, it would be good to make a version of the game where the player is put into a room, where they cannot move away from unless they choose a certain path. When the player moves, more of the maze and pathways will be revealed to them.