Contents

[Introduction: (~100 Words) 2](#_Toc80952957)

[Function One, *mainloop()*: 2](#_Toc80952958)

[Flow Diagram: 2](#_Toc80952959)

[Pseudo Code: 2](#_Toc80952960)

[Analysis: 2](#_Toc80952961)

[Function Two, *cursor function*: 2](#_Toc80952962)

[Flow Diagram: 2](#_Toc80952963)

[Pseudo Code: 2](#_Toc80952964)

[Analysis: 2](#_Toc80952965)

[Function Three, *has\_won*: (~150 Words) 2](#_Toc80952966)

[Flow Diagram: 2](#_Toc80952967)

[Pseudo Code: 2](#_Toc80952968)

[Analysis: 2](#_Toc80952969)

[Summary : (~150 Words) 3](#_Toc80952970)

[Appendix : 4](#_Toc80952971)

[Entire Game Source Code : 4](#_Toc80952972)

# Introduction:

This reports purpose is to demonstrate a basic understanding of the Python-3 Language in a limited form by analysing three functions found within a section of code created in Python3, specifically a game. This is expressed through the use of flow diagrams, pseudo code of the game, and a breakdown of it’s functionality. These three functions/processes that have been selected include the game’s mainloop() which is an infinite loop used to run the application and process events, the game’s cursor location system which determines where on the application the mouse is, and finally the ‘has-won’ & ‘has-tied’ statements which determines whether or not a player has won or lost.

# Function One, mainloop():

## Diagram Description automatically generatedFlow Diagram:

## Analysis:

This code starts and confirms that Pygame is running, the term ‘mainloop’ derives from the fact that it is constantly looping in order to confirm that the game is running, which in turn keeps the game running. Since this constant loop is happening, it’s possible to update events (meaning checking if something has happened) through the loop. Because of an if/else statement it is constantly checking if ‘mousebtn\_down’ has been pressed. If ‘mousebtn\_down’ is clicked, then it will update the ‘click’ event and then continue with the loop, otherwise it will continue as normal. Similarly, there is the ‘exit’ event. If ‘mousebtn\_esc’ is clicked, then it will close Pygame and the application that is running it which will end the loop.

## Pseudo Code: Text Description automatically generated

# Function Two, cursor function:

## Flow Diagram:

Diagram

Description automatically generated

## Analysis:

This function find’s the mouse’s location to check if it’s a valid location to place an X or an O. If the mouse is not in a square, it will continue to check the cursors location until it is. When the cursor is finally in a square, it will move on to the next if/else statement which determines if the ‘mousebtn\_left‘ has been clicked. If not, it will loop like the prior if/else statement. If the mouse button has been clicked, it will decide whether or not it is X’s turn, as X is the default. If it is X’s turn then it will place an X in the square and continue to the next turn. Otherwise, it will place an O. This will loop continuously as long as the application is open.

## Pseudo Code:

## Text Description automatically generated

# 

# Function Three, has\_won: (~150 Words)

## Flow Diagram:

Diagram

Description automatically generated

## Analysis:

This function determines who won the match, if it is both

## Pseudo Code:

Text

Description automatically generated

# Summary : (~150 Words)

# Appendix :

## Entire Game Source Code :