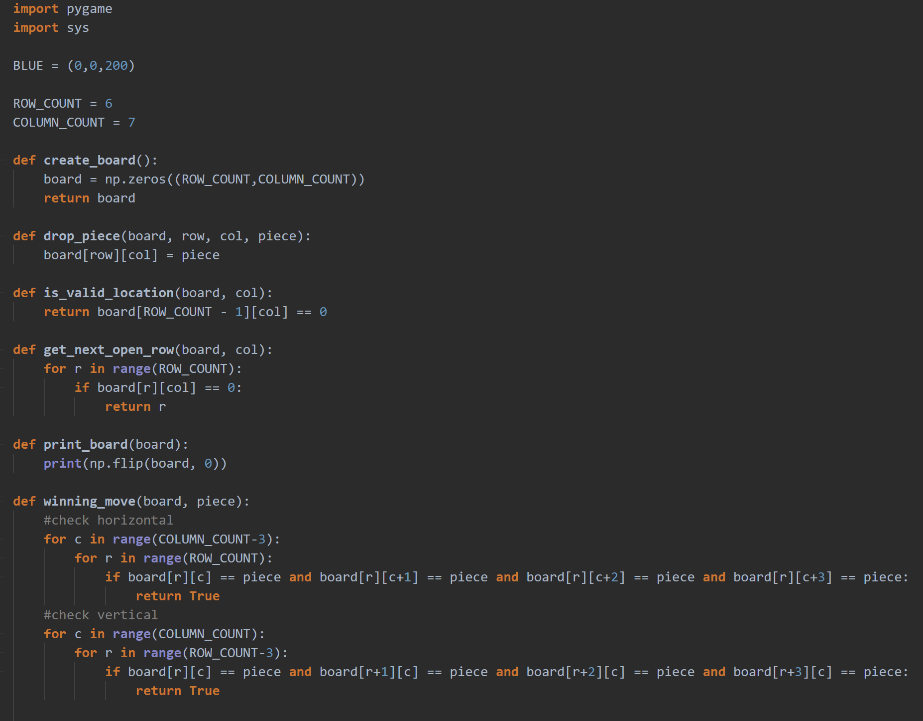
Mr. Lai

Software Design and Development

Student Number: 3168 3025

GUI development   
Project Logbook

Matt Starr



## Machine generated alternative text: Connect 4 GUI.py Connect 4 Terminal interface.py X Assignment 2 Matt-Starr-SDD-Assignment-2 •P Connect 4 Terminal in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 import numpy as np ROW COUNT = 6 COLUMN COUNT 7 def def def def def def create _ board( ) : np . zeros ( ( ROW_COUNT , COLUMN_COUNT) ) board return board drop_piece(board, row, col, piece) : board[row] [col] - piece is_valid_location( board , return board [ROW_COUNT - get_next_open row( board, col): 1] [col] col): for r in range(ROW COUNT) : if board[r][col] - e: return r print _ board (board) : print(np.flip(board, 0)) winning_move(board, piece) : #check horizontal for c in range(COLUMN COUNT-3)•. Logbook Entry 1 - Tuesday, 23 July

Today I developed a plan for the features of my connect 4 game and began construction on a test program that uses terminal as a user interface. I have decided to construct a terminal version of the connect 4 game as I am more familiar with the coding environment used to construct this version of the game. I can utilise this familiarity to construct the algorithms required to make the game, which can later be repurposed for the new environment.

I have also decided to create the GUI version of the game using pygame, which is a python extension that allows the control of a graphical interface. Once I have completed the terminal-based version of the game I will adapt it into something that works within the pygame extension.

Logbook Entry 2 - Wednesday, 24 July

Terminal controlled version of Connect 4 game