David Skerritt

https://github.com/DohnJoe-ost/Focus2

Assignment 2

For initialization of the game I used the code provided in the labs (board initialize) and I created a function to initialize the players. The initialize_players function contained an array of type players as an argument – the type players is a struct that contains info such as name, colour chosen etc. The function initialize_players asks two users to enter their names and also to select a colour – I put in place a condition that detects if a user has selected a colour that another user has chosen, and repeats until the user selects a unique colour.

I set up a function that initially outputs and prints one piece on each valid square of the board – 18 pieces for each player. At the start of the project I used a stack for the pieces however I could not get it to implement properly with the mechanics of the game so I decided to use an array to store the number of pieces of each player.
For the logic of the game I created a function called make_move – this function has a 2D array of type square and an array of type player as arguments. This function is set up to control the flow of the game by a while loop that repeats until the number of pieces a player captures is greater than 18 – once a player has all opponents pieces, the game is over and the winning player is announced on the terminal. Inside the while loop are the steps to ask a player what piece of their own they would like to move and to where they would like to place it onto the other players piece. I set up the board to have numbers on top of the columns and beside each row so users can easily specify where they would like to move a piece to. I set up a condition that if a user selects an invalid square or if they select a square with a piece that is not their own, they will be informed that they have selected a wrong square.
When the user selects the squares they would like to move from and to, the columns and rows are passed to a function (numb_pieces) that decrements the number of pieces on the initial square and increments the number of pieces on the square they choose to capture. The function numb_pieces also reprints the board to the terminal however it prints the board with the number of each piece on each square instead of what colour is on each square – this way users can keep track of how many pieces are left and determine how to play the game on each move.
As stated above, when a user has collected more than 18 pieces of an opponent then they are deemed to have won the game.

Conclusion:

My project does have some problems in the logic of the game – there is room to improve.
Choosing to use an array to store pieces instead of a stack was a better choice in my case as it allowed some sort of functionality to the game – however I will experiment with the stack to fully understand how to implement it and a linked list, out of interest.
Given the current covid circumstances and the disruption caused to the original plan to the assignment, I believe I would have been able to perform better on this project had circumstances been normal.