

Focus Game Project Summary

Student Number: 19366741

GitHub Repository: <https://github.com/EdmondAgbaje/FocusGame>

To create this game to function efficiently in C, I created a couple of different functions that dealt with different elements of the game. I had functions to deal with initialising the players, printing the board game, moving the pieces on the board, stacking pieces and dealing with general game logic such as each players turn and allowing players to add reserved pieces back into the game if they wish to.

To initialise the players, I created a function where I initialised different elements associated with each player such as their name, colour, number of captured pieces and number of kept pieces. This was done using the structs created. Player one is set to red while Two is green and they get to choose their name for the game at the beginning.

For the board, I created a function that deals with invalid pieces on the board and also sets red and green into their positions on the board. It sets the positions that can't be moved to on the board also.

To move pieces I used the positions to move the pieces. The player chooses the co-ordinates of the position they would like to pick and the one they would like to move to and I check the difference between the selected co-ordinates and the number of moves allowed at that position before making the move. I created a separate function that dealt with stacking the pieces keeping the newly added piece on top and increasing the number of pieces in the stack. I had to create conditions to cut the stack if the number of pieces in a stack is greater than 5. These excess pieces then become kept pieces or captured pieces depending on the players turn.

For my function that dealt with the player turns, I had to deal with all the scenarios that may arise at each players turn: if a player doesn't have any pieces on the board but has a kept piece, if the player has a kept piece and would like to use it, if the player has a kept piece and would like to continue the game as normal without using the kept piece and if the player doesn't have any kept pieces and must continue the game as normal. After these scenarios are played, the turn switches to the next player.

I had to also create functions that added kept pieces back into the game if a player would wish to do so. The logic I used behind these functions is very similar to the logic for the movement and stacking of normal pieces on the board.

Finally I had to create a function that would deal with conditions for when a winner has been found. This is when a player has no more top pieces on the board and has no more kept pieces remaining so the other player loses and a winner is found.

I also implemented functionalities that prints the rules of the game before the game begins so that players fully understand the rules beforehand.