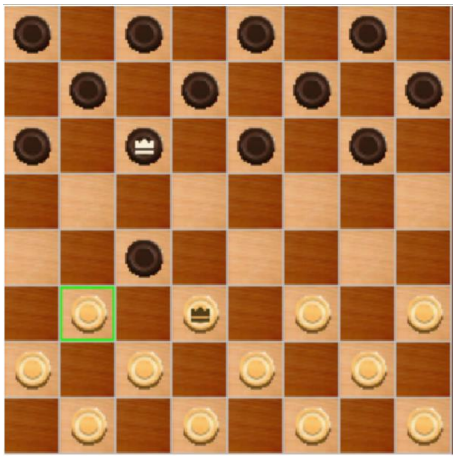
Checkers

Checkers is a two-player game that one can win by occupying all the other pieces of the opposing player. Initially pieces can only move in diagonal direction. Player 1's pieces can only move one up and one right and one up, one left. Player 2's pieces can only move down and right and down and left. These pieces can take opposition pieces when they are directly diagonal. Pieces can turn into king pieces when they reach the opposite side of the board. King pieces can move in any direction. Pieces can also perform double jumps where they take multiple pieces at a time.

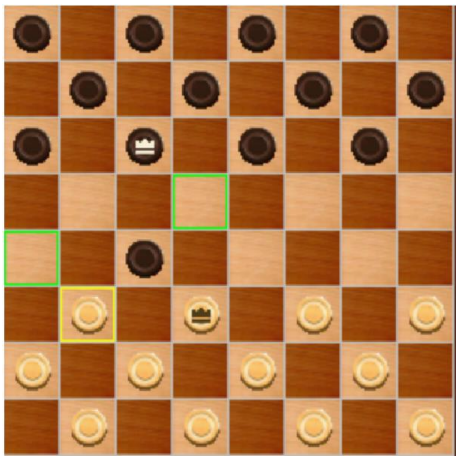
Timeline of the Game

The user can start playing the game by clicking the space bar to move on from the welcome screen.

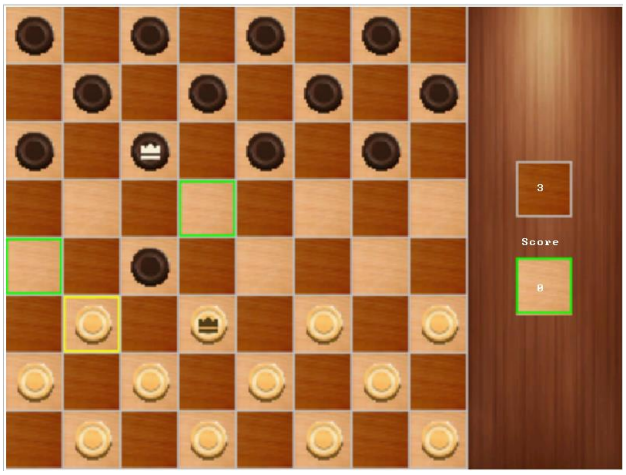

Once the game has started, the user can use the arrow keys to move around and select the pieces that they want to move.



Once a piece is selected, the user is prompted with valid moves.



The user can use arrow keys to select the move to make. The view on the side of the board tells the user whose turn it is and how many points each player has.




Once a player gathers 12 points, the game will end and announce the winning player.

Attribution Table

Shlok and Meet worked on first setting up the the initial display of the board and the pieces by creating sprites.

Meet Analyzed the DE1-Soc board manual to learn how to configure PS/2 interrupts as well as how to setup interrupts in C. Ensured to setup the stack pointers for interrupt mode in the main function. Setup the interrupt service routine (ISR) for the PS/2 by reading the port and calling the corresponding functions to run the desired output for interrupts from the arrow keys for movement and interrupts from the enter key for selection.

Shlok worked on the game logic, this included:

- Determining all the valid moves for a selected piece. For a normal piece this means checking the blocks directly diagonal to it in its direction. If the block contains 0 (no piece) it is a valid move. If a block contains an opponent and then a 0 (no piece) it is a jump and is a valid move.
- Determining if the piece performs a jump or double jump. (Recursive) This was a recursive function that checked if the piece could perform multiple jumps.
- On the move being performed, we perform any of the takes/kills. (Recursive) This function took the starting and ending points and went through every possible combination of movements till the cursor reached the ending position, when the ending position was reached, it would go back and take/kill all the opposition pieces that came in the way.
- Determining if a piece is turned into a king piece. When a piece reaches the opposite border of the board it turns into a king piece.
- Implementing the movement of the king piece, and finding all valid movements, jumps of king pieces. The king can move in all 4 directions, so we had to add conditions to previous functions to take that into account.
- Implementing the score functionality to determine if the game is over and which user won. Incremented the kill count for each user after a take/kill is performed, the one that reaches 12 first is the winner.

At this point the game was basically done, after this, Shlok continued to work on fixing bugs and edge cases.

Meet worked on improving UI by changing the background board with a board with wooden finish and switching the piece sprites to those that matched the board.

Meet worked on implementing the starting page, where he implemented polling to start the game only when the user presses the space bar.

Meet implemented character buffer to show a live count of the score and implemented a status bar to show us which players turn it currently was.

Meet then connected the logic of the game ending/ game winning to the UI, where he displayed an ending screen with the player number.