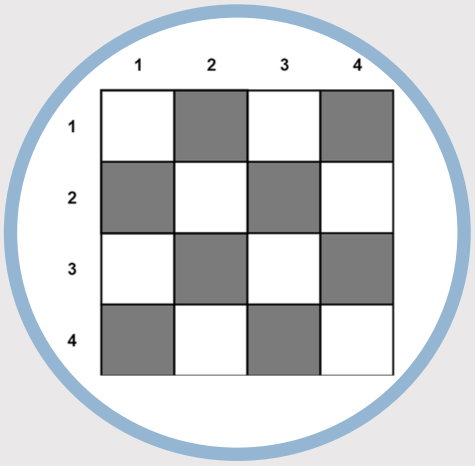

ALT CHESS

DESCRIPTION

Alt Chess is a 2-player game which is an alternate take on the game of chess. It consists of a 4X4 chessboard with 2 pawns, 1 knight and 1 king with the priority of pawn being the lowest and king being the highest.
The objective is to kill as many opponent pieces as we can and get points according to the priority of the killed piece.

COMPONENTS USED

1. Structures
2. Strings
3. If-else statements
4. Nested loops
5. Data Structures
    - Arrays
    - Linked Lists
    - Stacks (Implemented with linked lists)
6. gotoxy coordinate system
7. Custom type definitions
8. Combination of labels and goto

GROUP MEMBERS

Hriday Pradhan
Kavya Lilhare
Khushi Chawla
Mehak Agrawal
Shraddha Arora

FUNCTIONS USED

- **void main ()**
- **#define gotoxy(x,y) printf ("\033[%d;%dH", (y), (x))**
  Using gotoxy (x, y) which sets the cursor on entered x and y coordinates
- **void printStr (char c [], int times)**
  This function takes a string and a number and prints that string that number of times.
- **void printStr2 (char c1[], int times1, char c2[], int times2)**
  Overloading earlier function for different use cases
- **void printStr3 (char c1[], int t1, char c2[], int t2, char c3[], int t3)**
  Overloading earlier function for different use cases
- **void instructions ()**
  Instructions for proceeding with the game.
- **void scorecounter (int newrpos, int newcpos)**
  Records points of both teams.
- **void scoreboard ()**
  Displays points of both teams.
- **void pushInMoveStack (moveStack *stack, int row, int column)**
  Stores all possible moves for current piece to check if move entered by the user is a valid move
- **void pushInLeftStack (leftStack *stack, char toAdd[])**
  New Record of all the remaining pieces for a player
- **void emptyMoveStack (moveStack *stack)**
  Function to empty a stack of moves
- **int remainingCtr (leftStack *stack)**
  Counter variable for checking remaining pieces on board
- **void delFromLeftStack (leftStack *stack, char toDel[])**
  Deleting killed pieces from left stack
- **int pieceExists (leftStack *stack, char toFind[])**
  Function to check if a piece is still alive
- **void updateLeftStack (leftStack *stack, char before[], char after[])**
  Function to change a value in a stack
- **int checkIfValidMove (moveStack *stack, int row, int col)**
  Function to check if an entered move is valid
- **void createboard (leftStack *black, leftStack *white)**
  Creating a new board and assigning starting positions to the pieces
- **void displayboard ()**
  Displays current state of the board for a game

- **void displayPiecesP1 ()**
  Displays available pieces for Player 1 (White)

- **void displayPiecesP2 ()**
  Displays available pieces for Player 2 (Black)

- **int checkedgepiece (int a)**
  Function to check if a piece is at one of the edges of the board

- **int findenemy (int r, int c, int piececolor)**
  Function to check if an enemy is present at an entered position

- **void fullScreenMessage ()**
  Message to enter full screen mode is displayed

- **int showvalidmovesforuser (moveStack *stack, piece a , int val)**
  Shows valid moves for all pieces

- **void movepiece (int a, int newrpos, int newcpos, leftStack *stack)**
  Function to move a piece to a given position

- **void promote (piece *a, leftStack *stack)**
  Pawn promotes to rook once it reaches opposite end of the board

- **void display_name()**
  Function to enter the names of both players

- **void declareWinner (leftStack *black, leftStack *white)**
  Function for declaring winner

- **void display_textbox ()**
  Prints the box in which piece number is entered

- **int skipTurn (moveStack *stack1, leftStack *stack2, int piececolor)**
  When no moves are available for the player then their turn is skipped

- **void experience**
  Experience of all group members

. . . .