***FINAL PROJECT***

***DOTS AND BOXES***

***By:***

***Abdel Rahman Amgad Hassan , ID : 22010871***

***Abdel Rahman Mohamed Ahmed Nasr , ID : 22010887***

***Game Description:***

**We have implemented the application that allows you to play the game “Dots and Boxes” , which is played by filling in the lines between two different dots (or the borders of a cell/box) , with the player filling in the final border of a box being awarded the box and therefore a point , it does not matter who has the most lines in the box , only the final move matters , which forces you to corner your opponent into filling in the 3rd border and you to avoid filling any 3rd borders in order not to grant your opponent any points , the game ends when no more cells are empty (even if one of the users has already passed the winning point i.e : has attained more than half the cells in score points) .**

***Game Features:***

**Ou game provides a variety of features to add a lot of flexibility and options to the players :**

1. **The Main Menu:**

**It contains select options for the players to determine how they wish to play :**

1. **[2X2] which is the grid suitable for beginners which can be played with a friend , or against computer AI**
2. **[5X5] which is the grid suitable for experts which can also be played with a friend , or against a computer AI**
3. **[Leaderboard] which is the feature to display the top 10 scores rank-holders and their respective impressive scores ranked from top to bottom**
4. **[Load] which is the feature to load 1 of 10 save files (given that these files are already present and not empty) so that the user may pick up the game from where they left off last time they played , with the same game state features all as was (Names , Scores , Time played , Cells remaining , The grid itself and etc..)**
5. **[Exit] which of course exits the game till we meet you next time you return to our game 😊**
6. **In game Menu :**

**Lets say you picked any of the 4 game modes (2x2 or 5x5 , 2 or 1 players) you are greeted first with the option to enter the names for the players , which then would be used to store your scores , for a chance to enter the leaderboard 😉**

**Then you are introduced to the GRID , which is a number of boxes (4 or 25) depending on your choice of game mode and a small in game menu with a variety of options :**

1. **[Continue] the option to continue playing the game which when pressed well prompt you to enter a string of 3 characters [Row,Col,Border] which then would fill this border (if free and valid) and color it according to your player number (Red for 1 , Blue for 2)**
2. **[Undo] the option to delete the last move played and all the changes to game state structure caused by it (don’t use to cheat)**
3. **[Redo] the option to reinstate the last moved deleted by the undo option**
4. **[Save] the option to save your current game to 1 of 10 files to continue them when you are back (so you can finally pause the game and answer your mother calling for you 😊)**
5. **[Exit] of course the option to exit the game**
6. **Non menu features :**
7. **Leaderboard , the winner’s name and score are saved in a file which is rearranged whenever a user wins a new game to re-order and determine whether he gets to be on the leaderboard or not and remove the tails of the leaderboard if needed**

***Game Design:***

**The Design for the game is simple yet pleasing and efficient displaying the main menu as a sort of bullets placement in different lines and the grid as a simple design of ‘+’ signs as dots assorted vertically and horizontally with a space between each two dots waiting to be filled by any of the two players**

***Assumptions:***

***Data Structures:***

**Multiple structures and arrays were used in the implementation of the program to facilitate and optimize the programming procedure , for examples:**

* **[GameState] Data structure : which is the main data structure for the game containing all of the game data like (grid size , time , turns , player names , player scores , number of cells filled and the Cell data structure)**
* **[History] array: a 1-D array of data type gameState containing all instances of gameState structure to implement the function of undo and redo easily**
* **[Cell] Data structure : which is the data structure of each singular cell and containing the state of each of its borders (up , right , bottom and left borders)**
* **[Cells] array : which a 2D array (matrix) of cells data structure containing NxN cells (where N=2 or N=5)**
* **[LeaderboardEntry] Data structure : a simple data structure containing the winners name and scores**
* **[Leaderboard] array : a 1-D array of size 10 which is used to load up the top 10 leaderboard entries using the loadleaderboard() function and displayed using the displayleaderboard() function**

***Important functions and modules:***

* **saveGameState: saves current gameState**
* **loadGameState : loads gameState from a certain file**
* **loadLeaderboard : loads top 10 entries of leaderboard to the leaderboard array**
* **saveWinner : saves the winner and adds him to the leaderboard and rearranging it from top to bottom according to the new entries**
* **saveLeaderboard : saves leaderboard after updating it from the sameWinner() function**
* **displayLeaderboard : prints the leaderboard array which only contains the top 10 entries in the leaderboard file**
* **undo : deletes last move played and all of its effects (goes one step back in the history array)**
* **redo : reinstates the deleted move**
* **scanNames : takes the name of the two players if the names are not loaded from a loaded file**
* **printMenuAndGetCommand : prints main menu and waits for the command**
* **createArr : dynamic allocation for cells 2-D array**
* **initializeGameState : initializes the gameState (if not loaded from a save file)**
* **printBoard : prints the grid for the game including using color for the borders played by each of the two players ( Red for player 1 , Blue for player 2 )**
* **checkCellFull : checks if a cell has been filled by a player and adds the point to him and increases the number of cells filled by one**
* **checkWinner : when the number of cells filled is equal to the number of total cells , a quick check to see who has more cells and rules out a winner (or a draw in the case of 2x2 grid where a draw is possible).**
* **checkValidity : function checks validity of the “play or move” made by a player to see if it checks out or not (if not , it prompts the player to reenter another valid move till a valid move is entered)**
* **currentGameTurn : the function which carries out the move asked by the player (if the player chooses the “continue” option and continues playing and changes the cells structure 2-D array and changes the neighboring cell to it to so that both touching borders match up)**
* **printData : prints names , scores of each player , unfilled cells and time elapsed since game started**
* **updateHistory : the function that updates the history array whenever a new move is made to enable the ability of undoing and redoing moves**
* **debugHistory : a function used to debug errors in the implementation of history array**
* **gameLoop : the most important functions in the game that first starts by checking for game type (loaded or new) and therefore starts a game , keeps looping , checking , calling other various functions to ensure a game flows correctly and efficiently till the game is over and checks for a winner calling almost all other functions**
* **main : starts game and calls other functions**

***Header Files:***

* **colors.h : it is used to implement all colors used in the game (red , cyan , blue , magenta , yellow , green and white)**
* **datastructure.h : it is used to define the datastructures like gameState , cell and LeaderBoardEntry**
* **functions.h : contains almost all functions used in the implementation of the game**

***Flow Charts and Pseudo Codes:***

***User Manual:***

***Game Rules:***

* **You are required to fill as much boxes as you can before all boxes are filled.**
* **Only the last player to fill a box matters , even if player 1 fills all three borders , if player 2 fills the 4th border he gets the point**
* **You can’t fill a line/border that has already been filled either by you or another player**
* **You can’t skip your turn**
* **The player who fills the final border in the box is – as mentioned before – awarded the point and granted an extra move after he fills the box which is stackable as long as he is filling boxes which each move he plays**
* **Filling a box in a complete shape (an enclosed shape) causes a chain filling all the boxes in the enclosure**

***Main Menu:***

**Enter a number from 1 to 5 to choose any of the main menu options**

**1-> 2x2**

**2->5x5**

**3->Leaderboard**

**4->Load**

**5->Exit Game**

***In Game Menu:***

**First , enter the names of both players for displaying in-game and recording in the leaderboard if possible**

***THEN*:**

**1-> Continue (play move)**

**2-> Undo**

**3-> Redo**

**4-> Save**

**5-> Exit Game**

***Sample Runs:***

**Winning :**