\*Features:

1. Player can choose to play beginner, expert or custom modes where each one differs from the other in the number of squares.

2. Player can choose between playing with another player (two players) or with the computer.

3. The computer is programmed to close a box if available, then to play in a box with 3 sides left, 4 sides left then 2 sides left on that order.

4. Player can return to the previous menu any time.

5. in-game menu contains undo, redo, save and exit as well as exit to main menu without saving.

6. time updates every second.

7. which turn it is, total moves left, turns played by each player and each player’s score are all displayed along with the time elapsed since beginning of game under the grid.

8. numbers are put on the grid in a way that allows the user to choose where to play with only two numbers separated by a comma.

9. player can undo all moves till beginning of game and then redo all of them if they wish to do so.

10. there are 3 different saved games at any given time and the user can choose to overwrite any of them if they chose to save the game.

11. player can load any of the 3 games from main menu and continue playing where he left off.

12. from the main menu, the player can choose to view the top 10 scores along with their usernames.

13. if a player achieves a high score, they are prompted to input their username in order to put in the top 10.

14. usernames are case-insensitive; meaning that if “aHmaD” is in the top 10 with a score of 8 while the least score in the top 10 is 3 for example, a new username “Ahmad” won’t make it to the top 10 unless his score is above 8. Note that the user name is taken from user after game is finished, that means in the previous example, if a player scored 5 he will be prompted to enter his username, however, he will not be put in the top 10 if he inputs “Ahmad” because he is considered the same user “aHmaD” with a higher score of 8.

15. top 10 are displayed after a player achieves high scores and enters their name.

16. there is a “how to play” section where the player can see how to choose the desired column/row. However, this section assumes the player is familiar with the rules of the game. If not, a user manual is present in this document for how to play any game of dots and boxes.

17. the game is designed not to crash given any input. This is supposed to work for any number of inputs even if in a single line.

18. player can alternate between undo and redo in a streak as many times as he want but all the redo is deleted once he plays a new move.

\*Assumptions:

1. The game is designed to look like the photo given in the .pdf provided by you.

2. It is assumed that the user will see the list of top 10 so as to not to input a username as a present one; since they will both be assumed to be the same player.

3. the player name is only taken if a high score is achieved.

4. it is assumed that only one list of rankings is required. i.e. not one for each of the grids.

5. we also assumed the user will not change any of the .txt files

\*User Manual:

The game starts with an empty grid of dots. Usually two players take turns adding a single horizontal or vertical line between two not joined adjacent dots. A player who completes the fourth side of a 1×1 box earns one point and takes another turn. The game ends when no more lines can be placed. The winner is the player with the most points.

In our game the player chooses the column where they wish to play then the row separated by a comma. For more information on how to choose the column and row please refer to the “How to play” option from the main menu.

\*Description of files:

Note that only the important functions are stated below.

-grid.h:

Contains functions related to the grid like creating the grid, printing it, drawing a line in the grid and assigning a box in the grid to a certain player.

-gmaePlay.h:

This contains the structure Player as well as multiple other functions related to the game play like movesLeft which returns the number of moves left essential to know when to end game, checkbox which checks if the line drawn resulted in a box being closed and therefore needs to be assigned to the player using assignBox from grid.h, printBar which prints the bar containing info. Below the grid. It also contains the most important function for playing: play; which uses all of the other functions to loop until no moves left. There is also the function getInput which makes sure the user inputs a valid input.

-rankings.h:

It contains function needed for the Top 10 section. readTop10 reads the rankings from a file. checkHigh score checks if the new score is a high score. updateTop10 then puts the new name in the list.

-undo.h:

It contains some functions that are the inverse of some above like deleteLine, assignBoxAgain(with a blank space), checkBoxAgain to see if the undo will decrease the score of a playeras well as some other functions like addUndo, addRedo, undoPlay and redoPlay.

-computer.h:

Contains functions that enables the computer to play not only a valid play but also a good one. compSearchNum is used to search for a specific number of sides left in all the boxes then compSearchRc searches which side to play in that given box. The function compChoose is the one implementing both previous functions.

-save.h:

It has the functions essential for the process of saving and loadimg. The functions save the main grid, array of boxes, all the other data like the moves played by each player, moves left and the scores.

It also contains all the functions necessary to load all the saved info.