Programming report

App description:

Dots and Boxes is a game for two players starting with an empty grid of dots, two players take turns adding a single horizontal or vertical line between two unjoined adjacent dots. The player who completes the fourth side of a 1x1 box (or groups of one or more adjacent boxes) earns one point (s) and takes another turn

In this application we have designed a version of this game with ability to try different types of grids and other gamming features like ( how to play, save, load, undo, redo….etc). It was implemented using C programming language.

Data structures:

We have used different kinds of data structure

Arrays: like “grid” which is the main array in the app that saves the dots and lines in the game mode

And “boxes” which correspond every box in the grid and how many sides are available to draw lines

Structures: like “player” we have used this structure to save all players’ information like turn, score, and moves

Pseudo code:

This program is a version of dots and boxes game

Print game interface, get the user input and loop until it is a valid input

Case based on user input

Case 1 then start the game

Get user input and loop until it is valid

Case based on user input

Case 1 beginner mode set size to 2

Get user input

IF it is 1 then its human VS computer

Start “play” function and loop until there are no moves left

IF it is 2 then it is human VS human

Start “play” function and loop until there are no moves left

Case 2 expert mode set size to 5

Get user input

IF it is 1 then its human VS computer

Start “play” function and loop until there are no moves left

IF it is 2 then it is human VS human

Start “play” function and loop until there are no moves left

Case 3 custom mode get size from user

Get user input

IF it is 1 then its human VS computer

Start “play” function and loop until there are no moves left

IF it is 2 then it is human VS human

Start “play” function and loop until there are no moves left

Case 4 back to previous menu

Case 2 get file number from user

IF it is 1 then load saved game1

Load grid size , moves left, and players’ scores from the chosen file

Start “play” function and loop until there are no moves left

IF it is 2 then load saved game2

Load grid size , moves left, and players’ scores from the chosen file

Start “play” function and loop until there are no moves left

IF it is 3 then load saved game3

Load grid size , moves left, and players’ scores from the chosen file

Start “play” function and loop until there are no moves left

Case 3 print top 10

Case 4 start “how to” function

Case 5 exit game

Function play

Pass in: char array “grid”, size, computer state, loaded state and loaded moves

IF loaded is 1 or 2 or 3

then load specific data from text file using “loadBoxes”,”loadDatat” functions and start the game with them

Else

Start the game with default data

While moves left dose not equal zero

Print the grid with the drawn lines

Print information bar

IF it is computer mode and computer turn

Then let computer choose where to put the line

Else

Get user input while updating the time every second

IF user input is ‘e’ or ‘E’

Then exit to main menu

Else IF user input is ‘u’ or ‘U’

Then undo the last move

Else IF user input is ‘s’ or ‘S’

Then ask user for file number then save

While the chosen line is not valid

Print “invalid”

Get user input while updating the time every second

IF user input is ‘e’ or ‘E’

Then exit to main menu

Else IF user input is ‘u’ or ‘U’

Then undo the last move

Else IF user input is ‘s’ or ‘S’

Then ask user for file number then save

IF it is player 1 turn

Draw line

IF the line completes a box then increase player 1 score

Else

Draw line

If the line completes a box then increase player 2 score

Print grid

Print information bar

IF computer score > player score

Print “computer has won the game”

Else IF player’s 1 score dose not equal player’s 2 score

Print who has won the game

IF it is new high score

Get user name and update top 10 list

Else

Print “it’s a tie!”