By: Nikita Borisov, Justin Khanhkham, Augustus Lindell and Dan Prumanov

Dot and Box Game User Manual

This user manual will guide you through the process of compiling, running, and playing the Dot and Box game implemented in MIPS assembly language using the following four files: board.asm, Dots_AI.asm, resultsPrint.asm, and userInput.asm.

Prerequisites

To run this MIPS assembly code, you will need a MIPS simulator such as MARS (MIPS Assembler and Runtime Simulator). You can download MARS from the following link: http://courses.missouristate.edu/KenVollmar/MARS/

Loading the code

Open the MARS application.

Click on File > Open, then navigate to the directory containing the Dot and Box MIPS assembly code files: board.asm, Dots_AI.asm, resultsPrint.asm, and userInput.asm. Open all four files in MARS.

Compiling the code

Click Settings and check “Assemble all files in directory”, this ensures that all your files will be assembled together

Click the "Assemble the current file and clear breakpoints" button (represented by a wrench and hammer icon) or press F3 while resultsPrint.asm is selected in the Mars Edit window

Running the program

Click the "Run the current MIPS program" button (represented by a play icon) or press F5.

The game will start, and the game board will be displayed on the console in ASCII form.

Playing the game

The game consists of a grid of dots, and players take turns connecting adjacent dots with horizontal or vertical lines. When a player completes a square (i.e., a line is placed on all four sides), they capture the square and earn a point. The game ends when all squares have been captured, and the player with the most points wins.

Follow the on-screen instructions to make your moves. You will be prompted to enter the coordinates for two adjacent dots you want to connect with a line. The coordinates should be entered in the following format:

`“a1-b1”`

- `a1` represents the label of the first dot.
- `b1` represents the label of the second dot.

For example, to connect two adjacent dots labeled 'e1' and 'e2', you would enter:

`“e1-e2”`

Enter your move in the specified format, and the game should accept it. The game will process your move, update the board, and display it in the console. Play alternates between you and the AI opponent until the game concludes. The results will be printed on the console, displaying the winner and the final score.

Have fun!