**Gomoku**

User Manual

By: Nate Le, Abis Naqvi, Andrei Favorito, Danny Lee

CS 2340-001

Computer Architecture

Contents

1. How to Run the Gomoku Program 3
2. Running the Program in MIPS 3
3. Running the Program in the Command Prompt / Terminal 7
4. How to Use the Gomoku Program 10
5. User Input 10
   1. Illegal Moves 10
6. Aim of the Game 13
7. Ending the Game 16
   1. Ending the program in the Command Prompt / Terminal 16
   2. Ending the program in MARS 16
8. How to Run the Gomoku Program
9. Running the Program in MIPS

The .asm file contains the entire assembly program needed to run the Gomoku game. The .asm should be placed in a local directory that can be accessed by MIPS. Once MIPS is open, the .asm file can be opened via a couple of ways:

1. > Click on “File” in the top left of the MIPS window

> Click on “Open”

> Locate the directory that contains the .asm file

> Click on the .asm file which will open and be loaded onto the MIPS text editor.

Graphical user interface, application

Description automatically generated

“File” and “Open” shown in the MARS 4.5 window on MacOS.

1. > Click on the icon of a blue folder with a document in it near the top left of the MIPS window

> Locate the directory that contains the .asm file

> Click on the .asm file which will open and be loaded onto the MIPS text editor.

Graphical user interface

Description automatically generated

Shows the “blue folder with document” icon in the MARS 4.5 window on MacOS.

Once the contents of the .asm file is loaded onto the MIPS text editor, the program can now be assembled and ran. The program can be executed via a couple of ways:

1. > Click on the “Run” in the top left of the MIPS window

> Click on “Assemble” (or use the keyboard shortcut for it) to assemble the program.

> Click on “Run” in the top left corner of the MIPS window

> Click on “Go” (or use the keyboard shortcut for it) to run the program.

Graphical user interface

Description automatically generated

“Run” and “Assemble” shown in the Mars 4.5 window on MacOS. “Go” is also shown right below “Assemble.”

1. > Click on the icon of a screwdriver and wrench at the top of the MIPS window to assemble the program.

> Click on the icon of a green play button at the top of the MIPS window near the icon of the assemble icon to run the program.

Graphical user interface, text, application

Description automatically generated

Shows the “Run” button/icon in the MARS 4.5 window on MacOS.

1. Running the Program in the Command Prompt / Terminal…

The .asm file that contains the Gomoku program can be run on the command prompt / terminal of the respective operating system (OS). This can be accomplished by following the following procedure:

> Place the MARS .jar file into the directory that contains the .asm file.

> Open the Command Prompt (Windows) / Terminal (MacOS)

> We can use the following commands in both the Command Prompt / Terminal:

* cd - Change directory
  + This command changes the directory that is being currently worked with.
* ls - List
  + This command lists the contents of the directory.
* pwd - Print working directory
  + This command prints the path towards the directory that is being worked with / the directory that is currently being worked with.
* clear - Clear
  + Clears what is written in the Command Prompt / Terminal
  + \* However, this command does not quit a program running in the CP / Terminal
* \* USEFUL keyboard shortcut that quits a running program for both Windows & MacOS:
  + CTRL + C

> Using the commands above to finally get to the directory that contains the .asm file (see image “1B\_1.” [shown below] for what is expected in the Terminal of MacOS), we can finally open the .asm file using the following command (in the quotation marks, but do NOT include those marks in the command):

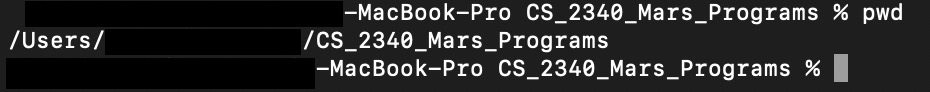
“java -jar Mars4\_5.jar [.asm file WITH extension]”

\* Type the name of the .asm file with the “.asm” extension following it.

\*\* It is better to have no spaces for the name of files/directories to ensure proper flow/path and access of those files/directories (see image “1B\_3.” [shown below]).

> The program should now be seen and working in the Command Prompt / Terminal (see image “1B\_2.” [shown below] for what is expected in the Terminal of MacOS [board should be shown]).

Image “1B\_1.”



(Note: The name of the user in this image has been redacted.)

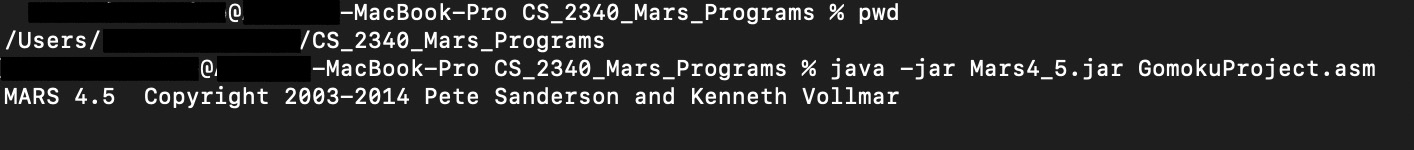
Image “1B\_2.”

Text

Description automatically generated

(Note: The name of the user in this image has been redacted.)

Image “1B\_3.”



(Note: The name of the user in this image has been redacted.)

(\* Important Note: The name of the file directory that contains the .asm file “CS

2340\_Mars\_Programs has NO spaces. The same goes for the name of the .asm file “GomokuProject.asm”)

2. How to Use the Gomoku Program

From here, we will now use the Gomoku Program once it has started running. If the program is not running yet, view step 1., “How to Run the Gomoku Program.” To play the game, the program will first inquire the player’s move.

1. User Input

The prompt “move?” is displayed which prompts user input - the player’s move. The program prompts the user for a letter and a positive integer which will both be read at the same/one time. This input will be processed by the program to display the player’s move on the board ‘P’ against the computer’s move denoted as ‘C’.

The user’s input must be an uppercase alphabetical letter (from A - T inclusive) which denotes the column of the board followed by an integer from (1 - 19 inclusive) which denotes the row of the board.

Other inputs will result in errors / improper execution of the program.

* 1. Illegal Moves

Although the user may enter a “correct” input that may not end / digress from the regular flow/execution of the program, a user’s input may be an “illegal move.”

An illegal move is made when the user enters an input / makes a move that occupies an already-occupied position (either by the player or the computer) on the board.

An example of this would be the user entering “A6” while a ‘C’ already occupies position “A6.” This also happens when the user enters let’s say “A6” again when the player has already occupied that “A6” position with a ‘P’.

If an illegal move is made by the user, the message, “Illegal Move”, is displayed.

Once the player makes an illegal move, the program does not stop execution but instead prompts for the player’s move again. This will happen repeatedly until the player makes a valid move.

The following images show the example above of the illegal moves made by the Player when they make a move that is already occupied by the Player:

Background pattern

Description automatically generated

Program has started execution and user input “A6”.

Background pattern

Description automatically generated

“A6” is displayed. The Computer plays “B18”. The Player then inputs “A8”.

Background pattern

Description automatically generated

“A8” is displayed. The Computer plays “H18”. The Player then inputs “A6” AGAIN. The message, “Illegal Move”, is displayed immediately after the player input. The program prompts for the player’s move again.

1. Aim of the Game

The aim/purpose of Gomoku is for either the Player or the Computer to get 5 of their pieces (‘P’ or ‘C’) in a row whether that’d be orthogonally (vertically or horizontally) or diagonally. The first to get 5 of their pieces in a row / form a line of 5 of the same pieces wins the game.

An example of the player winning would be the following set of the player’s moves: H8, H9, H10, H11, H12.

This is shown in the images below:

A picture containing table

Description automatically generated

Player inputs move “H8”.

Table

Description automatically generated with low confidence

“H8” is displayed. Player then inputs move “H9”.

Table

Description automatically generated with medium confidence

“H9” is displayed. Player then inputs move “H10”.

Table

Description automatically generated

“H10” is displayed. Player then inputs move “H11”.

Table

Description automatically generated

“H11” is displayed. Player then inputs move “H12”.

A picture containing table

Description automatically generated

“H12” is displayed. The Player wins by getting 5 of its ‘P’ pieces in a row / vertical line which also ends the game.

1. Ending the Game
2. Ending the Program in the Command Prompt / Terminal

The Gomoku game/program ends when the player or the computer first gets 5 of their pieces in a row (orthogonally or diagonally). The game/program can also end the running program when the Player/user hits “CTRL + C” in the CP / terminal.

* 1. Ending the Program in MIPS

The program also ends when the player or the computer first gets 5 of their pieces in a row (orthogonally or diagonally). The program ends when the user also hits the “Stop” button in the MIPS window.