ANH TRAN – pat200000

**DOTS AND BOXES PROJECT REPORT**

* Component of the program:
  + The program consists of 4 modules: userInput – ask the user for the coordinates, interface – update the board with the input and print out the board, errorChecking – check the program with numerous logic function after receiving the input, and computerMovement – programming the movement of the computer.
  + The program would then take in our input and run the algorithms to find the coordinates and connect the dots together and print it out, then the computer would take its turn.
  + A sound is generated when a box is completed whether it’s the player’s or the computer’s
* Problems Faced:
  + Throughout the project, we encountered problem to connect our modules in a seamless way as we were overlapping registers and it wouldn’t give the interface the correct coordinates to store into the board and print out. We figured out by making our function in the modules .globl so that our main module would be able to call on the functions or even global variables.
  + Also, we were experiencing problems in the errorChecking as it weren’t checking the input correctly, so the program kept going into infinite loops. We tried to fix and align all our input and registers so that the data and the address were correct and fix the logic of the error functions.
  + For myself and my module – interface, I was having issues with how to connect the input with the board and store it correctly and keep updating it while also check if we need to update the board or just print out the current board. I managed to used load byte to retrieve the input from the userInput module and save it in a variable so that I can call and store in the board while iterating through the 2d-array.
  + In addition, we ran into a simple problem where we didn’t know if we should make the logic of finding out whether the connection would be vertical or horizontal to be placed in the input module or the interface as putting it in the module would overcomplicate the 2-d loop. So we put it in the input module and it would pass 3 arguments for the interface: the row, the column, and the type of connection.
* Lessons Learned:
  + By doing this project, me and teammates have learn more about the structure of the memory address of MIPS and how we can manipulate the data in it to print out the content we needed.
  + In addition to learning about the various instructions that helped us achieve the logic to do the calculations for the program, we also improve our critical thinking since assembly is more logical than high-level language.
  + We also learned how to utilized the stack to store the return address of the subroutines
  + Using the system calls to produce sounds.
* Implementation:
  + The program works as when you input two coordinates to connect, the program would then calculate the index of the actual connection that would be on the board, as well as whether it will be a horizontal or a vertical connection. After that it will call the error checking module to validate the input and other things and then will send the inputs to the interface to be printed out.
  + For the interface, I just used a simple 2-d iteration for the board and print out each byte in address. But I also made additional functions that would be called upon by the main createBoard function to split up the process and make it more readable and reusable.
  + For the box completion sound, we created multiple labels for the pitch, the duration of sound, the instrument, and the volume. Then we load them each onto registers $t0 to $t3, respectively, and using syscall service 31 for MIDI out to produce the sound signifying the box being made by either the player or computer.
* Peer Evaluation:
  + All of us pretty much knew what we wanted to do for the program so we divided up the program into 4 and we randomly chose which part each of us would work on, and we have been collaborating to connect the modules as well as fixing each other’s problems in the modules.
  + Zen Yamaguchi was responsible for the computer movement module
  + Antony Batres was responsible for the user input module
  + I was responsible for the interface module
  + Nikhil Maraboyina was responsible for the error checking module
  + Nikhil’s part was pretty large so after we finished our part we also helped him with debugging his part and then all of collaborated online to connect all the modules together.