Carlos Vazquez

SE 2340.004

Project Report

1. Description of the program,

* A multiplication board game that matches a multiplication expression and a value on two matching cards. If the player gets all eight matches from the 16 cards, they beat the game.

1. The challenges that you and your partner had and how did the team overcome them,

* Trying to find a balance between how we could divide the work was a challenge initially, especially since my partner was not interested in adding descriptive comments to his code initially. However, I as we worked together to make better progress he understood to make more helpful comments and labels even with the talks through the implementations of his code.
* I struggled to make the meeting times with my partner in the beginning because of conflicting times with coursework from other classes, but we worked to meet twice a week and keep contact through text if more progress was needed.
* When trying to implement eachothers files together, they wouldn’t work until we worked together on eachother’s files so that they would.

1. What you have learned by doing the project,

* Coding can be difficult to divide if two people work on the same file or section of a file.
* Having a set coding structure between partners can create ease of reading for those reading eachother’s implementations of files or sections of code.
* Code comments make it easier for a team to interpret code and implement their work alongside; thus, all teammates should enforce their use of them.
* Significant changes to how the project functions should be mutually agreed upon and given a proper reason to be implemented.

1. A discussion of algorithms and techniques used in the program, e.g. how to display the board, how to check and remember a move from the user? How does the program work?

* The board is displayed through a system of .asciiz arrays that are iterated through loops each row printing the board and printing the cards that are flipped, keeping them flipped if the match is correct.
* We used a temporary print to print the user’s decision if it was wrong and if the decision was correct match, the chosen cards would be flagged and kept flipped.
* The program works on 4 .asm files, main.asm, locationCheck(Rand).asm, board.asm, and DataRand.asm.
  + main.asm prompts the user for card inputs, makes the card checks with subroutine calls to other files, and exits the program when the game is completed
  + locationCheck(Rand).asm checks the location of a card and returns the integer value of the card from the randomized positions.
  + board.asm prints the board temporarily for incorrect decisions, changes the board permanently for correct decisions, and prints the current board while waiting for the next decision.
  + DataRand.asm randomizes a .word array with a set of indices 0 - 15 to return to main and offset the cards to randomize the board. The array that is returned to main is given as a argument to the other subroutine calls to give the randomization to the .asciiz and .word representation of the cards,

1. **Contributions of your partner (peer evaluation),**

* Sawyer finalized the version of the board.asm and main.asm file
* Sawyer created the locationCheck(Rand).asm
* When my randomization file did not work, he restructured the DataRand.asm so my implementation could work properly with the other files.

1. Any suggestions you may have (optional)

* I would have liked more help with beginning or finding out how to divide and conquer certain project parts, such as the card randomization/shuffling and how to do the timer. If not, maybe some in-class MARS MIPS coding could help guide our learning of the program rather than relying primarily on independent learning.