Project Report

Created By: Noel Emmanuel and Donovan Okwonna

A description of the program

- · This program is a matching game in which there is a board of 4x4 cards. Each card has either a value or a corresponding expression that relates to the value of another card. The user selects a card which will then be revealed. The user then selects another card which is also revealed. If the two cards have a matching pair of value and expression, they will stay revealed, otherwise they will be hidden again. The user then selects two cards again. This process is repeated until the user has found all the matching pairs. Once all the cards are matched the game is finished.

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

- While developing this program, there were many challenges that I and my partner managed to overcome.
- The first challenge was simply finding a way to break the program down into parts. When first beginning this project, we found it difficult to start because we did not know which parts of the program should be in separate modules. Eventually, with help from a video from the TA, we managed to find a way to break the program down into several parts that were much more manageable.
- The second challenge that we faced was the creation of the board and the randomization of the cards on the board. Again, this challenge was overcome by breaking the issue down into parts. First, we focused on the creation of the board. This was relatively easy as we just had to put values into an array. Lastly, was the randomization of the board. This part was harder, but we were able to code a random number generator and used that to populate a randomized board. (Alter how you see fit)
- Another challenge that we faced was creating the display for the board. When we created the display, it was giving an error for misaligned data. This was at first difficult to solve because we were trying to debug it through writing debug code statements in the code. After a while, we used the actual debugger to step through the program and find the solution to the issue. In the end we added a end_display function that fixed this issue
- A big difficulty that we faced was the comparison check and how exactly that would be done. Since one of the cards would be a formula (a x b) and the other a number(c), it was difficult to figure out how to properly represent that in code. Eventually, we came to the conclusion that instead of doing the multiplication math operations, we could simply link two cards together so if a user clicks both, it is not doing the operations to check but rather seeing if they are already linked.

- Additionally, separating all the functions into different files and finding ways to efficiently make them talk to each other was a big ongoing challenge throughout the whole project. Many times, we had to completely scrap what we had and restart due to us making things more complicated than they had to be. From that point on we decided to try to make things as simple as possible and only create new functions when absolutely needed as compared to whenever we want.
- Lastly, the biggest challenge that we faced was integrating the split-up parts together. This took the most time out of the entire project as once we tried to put it all together, it completely broke. We had to spend a lot of time refactoring our code to fit each other's as well as naming conventions and register usage. The reason this took the most time was because by the end of it, we had to almost rewrite the whole code again, only this time with a blueprint to do so.

What have you learned by doing the project

- Developing this project with MIPS assembly taught me much about not only coding and developing a program in MIPS but also how to develop a program with collaboration. When it comes to coding in MIPS assembly I learned how to use and integrate multiple modules along with being able to break down code into multiple parts. Another important thing I learned with the MIPS assembly language was the use of arrays and how they played an important role in the development of this project. In terms of collaboration, I learned how to work as a team in order to split responsibility in the creation of a program. Much of our program was split into smaller pieces and we had to plan and design before even getting started on the code. It was also challenging to overcome combining the code that we wrote individually but through a joint effort we managed to make it work. Overall, this experience of working in a group along with the knowledge of how to build a project will undoubtedly be used to help me in my future endeavors.

Algorithms and Techniques

- How the board was displayed
    - The display module uses nested iteration to traverse through the array of cards. It checks each individual card to see if it is flipped based on the flipped status. The flipped status can either be 0 (card is not flipped) or 1 (card is flipped). If the flipped status is 0 then a question mark is displayed to hide the value or expression and if the flipped status is 1 then the value or expression is shown. This module also implements modular design and function reusability through the way that the borders of the board are displayed.
- How to check user input
    - In order to check valid moves from the user we used branching to make sure that the cards selected by the user were inbounds (between 1-16). There is also a

section within the code that checks whether the user has already matched a card or if the card has already been selected. We use different functions and branching statements to check whether the card that the user entered was valid. We also used message statements to tell the user whether what they inputted was valid or not.

- Data management
    - The arrays in our program were used to both store the values for the board along with storing the status of whether a card was flipped. We also implemented the use of pointer arithmetic which helped to access the value of the cards from the arrays. Overall, these arrays helped manage and track the state of the game across all the different files.
- How does the program work
    - Overall, in the creation of the program we ended with four different modules. The display module was used to display the board and show the flipped values. The user input module kept track of whether the user input was valid and which items the user successfully or unsuccessfully matched. The logic module which held the core functionality of our project. And lastly, the main module of this program handles the game loop, checks for whether a match was found or not, and whether the game has ended.

Contributions of your partner

- On this project my partner and I worked very hard on this project. We both contributed heavily to the creation and building of this program. I was charged with creating the display of the board and he was charged with the backend game logic. Although we had issues when developing this program, me and my partner were able to overcome many of them. My partner also excelled when it came to integrating the modules of the overall program. He ensured there was smooth functionality and debugged the program to optimize performance. Overall, my partner heavily contributed to the design and development of this game.

Program Vido Clip Links:

- Youtube Link:
    - https://youtu.be/aVs25XGQ0vs
- Screencastify Link (Backup)
    - https://app.screencastify.com/v3/watch/DxWf0s58jr5KQuHsgQ2X