**Team Project Report**

**Program Description**

Our team project was a MIPS instruction program written in Mars. The program that we made is a Connect 4 Game, the goal being to connect 4 of your tokens in a row either diagonally, vertically, or horizontally before your opponent does. The game has two players, the user, and the opponent. The game draws the board in a grid with underscores \_ and asks the user to input a number between 1-7 to select which column they would like their token to be inserted. The board is redrawn, but with a X in the column the user selected, replacing the bottommost \_ in that column. Afterwards the opponent, which is the computer, randomly selects on the grid and the board is redrawn again like before, but with a O instead of a X. This process repeats itself until one of three conditions are met, the first being the user having four X’s connected in a row, diagonally, horizontally, or vertically which results in a user win and displays a victory message. The second condition is similar; however, it is when the computer has four O’s connected in a row, diagonally, horizontally, or vertically, and displays a message declaring the computer has won. The third and final condition is met when all the slots on the board are taken with X’s or O’s but neither the user or computer has 4 connected in a row, which results in a draw and an accompanying message along with it. There is also a way to exit the game, by typing 8.

**Challenges and Solutions**

There were many problems that we faced during this project, those problems all being of two categories. One: the code and problems with the program itself, two: the individuals and anything to do with collaboration attempts. As for the first category to resolve any of these problems we would have to come together to figure out solutions. In the beginning it was hard to get everyone to take time out to come together and think about the issues as we thought that the project would eventually just be done. However, when we finally started to get into the project details and the code itself, everyone eventually showed initiative and participated greatly. This problem resolved itself as the due date came closer.

As for the first category, problems with code and implementing solutions became much easier as everyone began to participate and input great and unique ideas. A major problem we faced initially was on how to print the ascii board itself, we could not figure it out but then realized that to print characters you need to use ‘11’ instead of ‘4’ in register ‘v0’. This helped a lot as this is one of the major requirements of the project. Another problem we faced was the win conditions, to figure out how to check for wins, especially in a diagonal manner took a lot of thinking and discussing before we could even attempt a solution. When we did attempt, we were able to get it to work immediately because of how well thought out the solution was.

Another issue was that if we brought all our code together how we would integrate each other’s code so that it works fluidly. The solution we came up with was that each person would work on their code sequentially, so that it all went together without anyone having to change anything significant.

**What I Have Learned**

I have learned two major things, as mentioned previously. The first category would be how to work with a team when it comes to coding. Initially I wanted to take everything and do it myself as that’s how I’ve always done projects, however I held myself back because I thought maybe there was a reason the professor assigned us a project with people. This made me learn to trust others code, and honestly it helped a lot as there was a lot of burden that was taken off my shoulders, which then allowed me to come up with better ideas for the code that I was responsible for writing.

For the code, I learned a lot about how to use arrays, and loops. I also learned a lot about how mips works.

**Algorithms and Techniques**

We used many different algorithms and techniques while doing this project. As stated, before in the last section, we used different arrays for each of the columns to store values. We also had a system to check each array and the rows that was used to check for win conditions. Another technique we implemented was a while loop to create the board. There is an outer loop that can be used to quit the game at the end of each round by inputting the number 8. The game printed by using 7 arrays as each column. The way it would print is by printing the first index of each column and then a new line was printed, then the next index was printed of each column until we reached the full board. The game would also ask the user for an input 1-7 and we used if/else statements so if the input was 0 or lower, it would display an error message, and also if the input was 7 or greater (except 8) than it would also display a error message. The program also checks if the column is full after a user input, also displaying an error message. For computer input, the computer randomly selects a number between 1-7 like the user, and if that column was full, it would loop and pick again until it successfully took its turn. The if else statements also would check to see, starting from the bottom up to add at the bottom of the columns first and then go up from there if there were other values in the columns. We had win conditions using the following: the horizontal, vertical, diagonal and draw. We had two ideas, one way would be to see the last inserted place and do all checks from there or to check everywhere that was possible for the wins. We chose the second method, as this was something a little easier to wrap our heads around. We had the horizontal check that went through the first indexes of each array, like the print function, the vertical check went through each array and then checked the next one, as each array was a column. The diagonal had to check each position individually. The draw only checked the first index of each array to see if the board was full.

**Peer Evaluation**

Ibrahim Barney(ME): I was the person to make the initial ascii board and the general foundation of the program; the print function, the user input, and the computer’s random number generator and input into the print function. I also helped get everyone started and assigned the job roles. Assisted with anyone that needed additional help as I wrote the initial code.

Samuel Say: Samuel assisted with the draw condition and helped me with the pseudo code for the entire program. He was essential to get things started and get things where they needed to go. He also assisted when we had major problems with the code in crunch time moments. He also helped make the program more efficient, proofreading the code after a major update was done.

Donovan Rodgers: Donvan had incredible critical thinking skills and was the key to solving one of our toughest problems, being the win conditions as well as when we didn’t know how to make the print an ascii board he came up with the solution, which I then implemented with the help of Alaan and Samuel. He did most of the second half of the program and was a key part of the solutions to many problems, showing a great knowledge and talent of mips and coding ideas in general.

Alaan Sharif (Myself): Alaan was our designated spokesperson, he also helped greatly with pseudo code for the program and made sure the project was coming along and tied together. He also helped make the program more efficient, proofreading the code after a major update was done. He made the reports, the user manual, and did the demonstration video.