**CS 3340 Project Report**

by Sudarshana Jagadeeshi

**a) a description of the program,**

The program is a game of Connect 4 in which the user plays (using the console) against a computer which has no winning algorithm. The board was stored using six arrays of integers, with each array representing a row. When the game starts, the user will be given a blank board - the user always gets the first move. The user must enter the column number in which they would like to drop the coin. The coin then drops in the column if there is room and a ‘ding’ sound is played. If the column number entered is invalid, the program will prompt the user again. After the user enters the turn, the program will check whether the user won. It does this by checking for vertical victory, then horizontal victory, then diagonal victory. After that, the computer will choose a column (using the random number generator) and its coin will be placed. Then the program will check whether the user won. This sequence continues until someone gets 4 in a row or the board is full. The user’s coins are represented as “1” and the computer’s pieces are represented as “2”. If the user wins, a jingle is played and a congratulations message is displayed.

**BONUS FEATURES:**

* There is a ‘ding’ sound after every valid move to let the user know that their move was valid and processed. If the move is invalid, the ‘ding’ does not play. It lets the user instantly know whether their move was acceptable without having to read the console.
* There is a victory jingle that plays only if the user wins the game. It is in the tune of Twinkle Twinkle Little Star and is demonstrated in our video recording of the game.

**b) the challenges that you and your team had and how did you or the team overcome them**

One of our challenges was on how to implement the diagonal connect. To solve this, we had to draw out the board and enumerate all the possible combinations. We also had to initially settle on an approach. We could have used one 2D array for the gameboard. However, we preferred simplicity over elegance, so we settled on just brute force checking. Though our code was long, we were able to complete our checking algorithm much faster.

**c) what you have learned by doing the project**

I have learned how to modularize and implement large assembly programs with several subroutines. I also learned how to comment my code and keep it organized on a larger scale. At the start of the project, I still had some ambiguity on very simple instructions(i.e. lw, li, la), but now all those instructions are very clear to me.

**d) a discussion of algorithms and techniques used in the program**

We used a massive loop to execute the main part of the game. The order goes like this:

1. Prompt user for their move
2. Place coin on board
3. Check whether the user won
4. Generate a random number for the computer’s move
5. Place coin on board
6. Check whether the computer won
7. Check for tie (42 pieces placed and no victor)
8. Repeat

We separated the checking of who won into three parts. The first part loops through the board horizontally, starting from the bottom row, to see if there is a sequence of 4 pieces in a row. The second part loops through the board vertically, starting from the top row, to see if there is a sequence of 4 pieces in a column. The third part checks diagonally. First, it checks all diagonals going from the top left to the bottom right. Then it checks all diagonals going from the top right to the bottom left. If at any point a victor is detected, the program branches to the victory/defeat message and plays the jingle.

When the user gives a column number to insert a coin in, we do not immediately place it. The program first checks if the column number is valid (1-6). If not, it reprompts. Furthermore, whenever the user or computer gives a column number, the program checks to see if the column is already full. If it is full, it prompts the user for a new column number (or gets a new random number from the computer).

**e) contributions of each team member (peer evaluation)**

My partner was very helpful. He wrote a good portion of the code to check for a connect 4 (horizontal/vertical/diagonal), though I did help him with this as well. I coded the board, sounds, prompts, etc. He was also very fast to communicate and got started on the program well in advance.