**Project Report**

**Description:**

The program is a game of Reversi which is written in MIPS language. The game features an 8 by 8 ASCII formatted board which is labeled A1-H8 and black and white pieces represented as ‘B’ and ‘W’ respectively. The player has the option to choose which color he or she wants to play as. Implemented in the program is AI that favors moves that gives it the best opportunities by choosing the option that returns the most amount of captured pieces. The program ends when there is no other possible moves the player could make either when the board is filled entirely or the player has no other options to place his or her pieces. The program then counts the total pieces of each player and displays the winner.

**Challenges:**

We faced a few challenges when creating the program, one of them being the process of formatting the board with ASCII characters. It was an arduous process with lots of trial and error to space everything correctly and place pieces. Another challenge we faced was scheduling our meetings. We had a bit of trouble finding available times that worked for all of us at once. We solved this by working on chunks of the project in split groups for whoever was available that day and we just combined that work with whatever we already have.

**What I learned:**

This project has strengthened my ability to code in MIPS and reinforce concepts about MIPS that I learned throughout the semester. I also learned that it is important to communicate ideas clearly when trying to collaborate in a program. At times, I find myself having trouble going through someone else’s piece of code and understanding what they are trying to do.

**Algorithms and Techniques:**

The board is an 8x8 array displayed in ASCII letters. Initially, there is 4 spaces in the centered that is occupied by B and W pieces such as it is in the actual game of Reversi. After the player makes a move, the AI goes through all the possible moves it could make. If available, the AI will choose the move that gives it the corner space. If such move is not available, it will choose the move that gives it the highest number of captured pieces. However, if there are no moves it could make, the player will be displayed as the winner and the program ends. Conversely, if the player has no other options for moves, the AI will be displayed as winner. In case of invalid moves, whether the player chooses an incorrect space or gives an unacceptable input, an error message will be displayed and the player will be prompted to choose a different space.

**Peer Evaluations:**

Matthew Brazzel : Leader of the group, came up with main ideas and concepts of the project

Jack Walsh : Assisted in the code, organized the project

Michael Wilson : Assisted in the code, organized the project

Toby Chow (Me) : Assisted in the code, did comments for program