Ryan Smith

COMP1050

4/24/20

Final Report – Tic Tac Toe

Tic tac toe has been a personal programming goal for me for several years and this project gave me the perfect opportunity to accomplish this goal. Creating a basic, player versus player game would not have been too challenging and I wanted a problem to work through. The problem I chose was a basic computer that a human player could play against. This computer would consist of three difficulty levels (easy, normal, and hard) in order to give it some variety and to give the player a choice. The algorithm used to decide where the computer should put its marker is a basic points system that sets the points of each spot on the game board according to the spots value. A good spot gets more points, while a bad spot gets less. To do this, the Computer class of the program checks each spot on the game board. Each spot has zero points by default and gets five added to show that it is open. Then five more are added if the spot is in the corner, and 10 are added to the middle spot if the player went first. After all the points are calculated, the Computer class uses another method to choose a spot based on the spot's points and the current difficulty level. The easy difficulty level simply chooses a random spot on the board unless a spot has over 100 points, in which case it chooses that spot. The normal level chooses spots with over 100 points first, then over 50 if there are none over 100, then a random spot if there are none over 50. The hardest level chooses the spot with the most points, usually resulting in a tie.

The rest of the Tic Tac Toe program is controlled by two classes: the Game class and the Main class. Game controls all the functions of the game. It has several classes that provide different functions related to the game board, such as checking for victory from either player or adding a player's marker to the board. It also controls the score, which is incremented each time a player wins. The Main class handles any interaction between the player and the game via the scanner and controls the game loop, or the flow of the game. The game loop consists of a several parts and only ends if a player wins or if there is a tie. If the player goes first, it starts by displaying the game board and getting the player's move. The Game method is then used to check for victory. If a victory or tie is detected, the current game will end, and the player will be asked if they want to play again. If no victory has been found, then the program will use the Computer class to decide the computers next move and check for victory. If there is no victory again, the loop will restart by printing the game board. The process is similar if the computer goes first, although the computers move will be calculated before showing the game board for the first time.

Although the game is done entirely through the console, the controls and display are meant to be friendly and intuitive. The game starts by welcoming the player and prompting them to choose a game mode. The player uses numbers to make choices, which are 1 (player versus computer) or 2 (player versus player) in this case. If the player chooses option 2 the game starts, but option 1 leads to choosing the difficulty of the computer. After that, the game will start, with either the player or computer going first at random. The game board is displayed with numbers 1 through 9 filling the spaces, representing the number to enter to place a mark in each spot. If at any point the player enters an invalid input, such as a letter or a number that does not represent anything, they will be asked to enter a valid choice and the program will continue. Once the

game ends, the player will be asked if they want to play again. The score will be displayed after each game and will keep track until the program is ended.