Connect Four Structure and Textual Description

San Marcos 3

*Jacob Hopkins, Andrew Kelly, Anne Leach, Daniel Lopez*

The project is a Java recreation of the classic tabletop game Connect4. The project is structured into three main packages following the “model, view, controller” architecture, such that the View classes will read user input and call Controller functions which may send the information to the Model package for interpretation. The game logic itself is held in the Model package, which includes rules and the AI for single player games. The window structures are held in the View package, the classes of which are responsible for the graphical user interface. Input from the user interface is sent to the Controller class, in the Controller package, where it will either interpret the information and update the view or interact with the model.

The core game logic is held in the “Game” class in the model package. This class will receive “on click” events from the interface, such as the player clicking on a column on the board. The class will first check if it is the player’s turn or if the game is over, then call a “addDisc” function which will ensure the column is not full. The piece is then created with the appropriate color and animated into position, then have its place recorded into a grid. Finally, the function will then call a win condition check function which will determine if the previous placement resulted in a game over condition, which results in the game ending.

For single player games an AI script was made which determines optimal placement by means of a series of comparisons. The AI has access to the grid, and evaluates each column individually, storing each score in an array. The AI then returns the highest scoring column. The scoring algorithm will first check for a win condition. A placement resulting in its own victory will yield the highest score, while a player win condition is given the second highest. Next the algorithm will check if the column is adjacent to two or fewer pieces of the same color. This is given the next highest scores, as such placement will be either offensive or defensive depending on the piece color. Lastly, if placing a piece would set up the player for a win, it is given the lowest score, unless such a placement would result in the AI winning. Before evaluating each score, any full columns are given the score of zero. The highest scoring column is returned.