Final Project Proposal: Connect-3

CS-230

Prof. Andrew Mehri

Proposal by William Varner

**Introduction**

Connect-4 is a classic board game that uses checker pieces on a vertical grid to arrange a tic-tac-toe type of victory by matching four pieces in a row. For the classes final project, we have decided to create a variation of Connect-4 due to how popular and simple it is as well as how easy it is to expand on it. The two main design changes we are making are (1) make scoring a point possible by connecting three pieces in a row and (2) add items that make the game more interesting. We believe that this provides enough of a challenge for the class as well as having it be something interesting and fun to make.

**Game Design**

To mix up the game to make it our own, we have decided to make the grid slightly bigger. The traditional game is usually seven columns wide with each column being able to hold 6 pieces. We will be using vectors to hold each column, therefore allowing freedom among the size of the grid, allowing us to change the size in development to conform to the game length and difficulty we desire. This also allows for more possibilities to connect three pieces and score points.

The game is supposed to end either when a point threshold is met or the board fills. This allows multiple win conditions which provides more depth to the game without adding complexity, keeping the game’s simple nature intact. This is especially true when we allow users to use points to get items. There are currently two items in mind. The first is a “boulder” that clears an entire column. The second is a “swapper” that swaps two pieces. These are simple enough to understand but will be somewhat of a challenge to implement.

**Technical Design**

The game is set up in a two-dimensional grid system, therefore a vector will be the data structure used to hold and display the state of the grid. Each column will be a vector with a piece dropping being simulated by pushing into the vector. This also allows the size of the board to be modified during development due to vectors being made in heap.

Checking to see if a player has scored a point will take place at the end of every turn. This is done by calling a function that checks surrounding coordinates in the grid. If a matching color is found, it then calls the individual function to check for another piece of the same color. If the area checked contains that color, the player scores a point. The checking function will call three others to check for horizontal, vertical, and diagonal matches. They each will take the grid location as an argument and will check the other grid locations in respect to the location passed.

When it comes to the items, the boulder will have to clear out an entire vector. This must be done using an iterator so as not to ruin the grid by deleting the entire vector. The swapper will simply receive two coordinates and switch the color of each piece. The point checking function will then be called twice to make sure that both pieces at their coordinates can gain a point.

In terms of development, the order must be considered. I believe getting the base game up and running without points or items is the first step. Afterwards, adding in the point system and the items can be divided into the group for different people to work on. The point system will have to come before the items because the items need to be bought with points.