Artificial Intelligence - Five in a Row

Chia-Hua (Kai) Peng

Purdue University

**Summary**

The Five in a Row Game is a computer board game designed in C++ and Qt. Player has to connect five pieces of the same color in a row horizontally, vertically, or diagonally to win the game. Player may use the mouse or the arrow keys to control the cursor. User may select two player mode to play with a friend, or challenge the computer that mimics human strategy about how to win the game.

**Algorithm**

There are two parts of the artificial intelligence algorithm – to set up a priority mechanism, and to determine at which coordinate the computer should place a piece at.

This algorithm utilizes two priority queues to set up the priority mechanism. The purpose of the first one is to help the computer prevent the player from making connections, and the purpose of the second is to help the computer making connections for itself. Those priority queues store the possible coordinates that player/ computer may place the pieces at. The coordinate that may result five pieces in a row has the highest priority, and the coordinate that may results less pieces in a row has the lower priority.

As a player place a piece at an empty coordinate, the critical coordinate, it may cause connection of pieces horizontally, vertically or diagonally. Notice that only the row, the column, and the two diagonal lines running through that critical coordinate will have additional connections. For each entry in the row, the column, and the two diagonals, run through the neighboring entries that are within 5-suqre-unit away from the entry in horizontal, vertical, and diagonal, check whether a connection is made if player place another piece at that entry, and insert the entry coordinate with the maximum length of the connection in the first priority queue. If the entry coordinate already exists in the priority queue and the priority is higher than the one in the queue, update the priority of the entry coordinate in the queue. This priority queue stores the coordinates the computer need to place in order to prevent the player from making connections. Similarly, when the computer place a piece at an empty coordinate, the critical coordinate, follow the same algorithm to store the possible entries the computer can make connections for itself in the second priority queue. The first priority queue is called the defensive queue and the second is called the offensive queue, naming after the purpose of the data structure.

Next, the computer will compare the front elements of the two priority queues, the two coordinates with the highest priority in each of the two queues. Pop the front element of the queue that has a higher priority and have the computer place the piece at that coordinate to make a defensive or offensive move. If the coordinate of the front element is already occupied by a piece, pop the front element and restart the comparison process. For the cases in which the front elements have the same priority, shift to offensive mode when the priority is high, otherwise keep the computer playing in defensive mode. Thus, the computer will attempt to win the game when there is a chance to connect 4 or 5 pieces in a row, and also prevent the user getting in a situation to make multiple 3 pieces in a row.