Assignment #2 – COMP 4106 W 17

**DOMINATION – GAME IMPLEMENTATION**

*Overview*

This artificial intelligence problem focuses on the game Domination. The game is played on a 34-space board, and involves players taking turns trying to make stacks of pieces and capture the opponent’s pieces. A player wins when his or her opponent cannot make any more moves; this means that he or she has no exposed pieces/stacks, and no pieces in his or her reserve of captured pieces that he or she owns.

Players make one move during their turn and may move a stack of pieces vertically or horizontally up to the size of the stack. Pieces that land on a space with a stack already on it are added to the top of the pile on that space. A stack may be split, with the player taking only the top half, however the max move distance is reduced to the size of the moving half of the stack. When a stack reaches a height greater than 5, the excess is removed from the bottom and added to the active player’s reserve. During a player’s turn, he or she may put a piece from his or her reserve anywhere on the board instead of moving a stack as normal.

*Search Space*

The number of moves a player can make at any point changes with the state of the game. Near the beginning of the game, with the default sized board and regular setup, the active player can make anywhere from 60 to 80 moves, depending on the distribution of pieces. These moves have very little score distribution and are very similar, so the vast majority of them are not ignored. As stacks are created, the number of moves decreases, as pieces not in the dominant position in the stack are dormant and do not add to the board complexity. The number of moves increases, however when pieces are captured, as moving a piece from the reserves to the board incurs another set of moves equal to the number of spaces on the board. As the game progresses, pieces that are captured by their respective owner’s opponents are effectively removed from the game, meaning that the games complexity trends to 0 as the time goes to infinity. Near the end of the game, the players are often only presented with 20 to 30 choices with very extreme score results, so the search space reduces drastically.

*Heuristics*

1. **Stack Count** – This heuristic uses the number of stacks the current player has versus the number of stacks the opponent has. This is equivalent to the number of exposed spaces with a corresponding piece, and is equivalent to the number of board-based moves the player has access to.  
     
   This heuristic performed poorly as it did not account for the reserve pieces. It also frequently led to an infinite game as it focused on protecting stacks and was therefore defensive in nature.
2. **Hoard** - This heuristic heavily weighted the number of pieces the current player had in reserve (to use at a later time).  
     
   This performed much better, as it more often terminated the game. It had an effect of “snowballing” a winning player, as the reserve move becomes more powerful and versatile as the game progress. The player that was able to secure pieces early both protected his or her own and invested in the later game.
3. **Attack** – This heuristic heavily weighted the capturing of pieces from the opposing player. The higher the number of captured pieces, the more likely the game is to end soon.  
     
   This heuristic performed very well as it both increased the speed of the game and exposed a certain weakness of the game. More often than not, the player going second would win with the heuristic. A hypothesis to the occurrence is that the inherent disadvantage of going first comes in the form of making a stack first. An opponent who wishes to capture pieces of the other player is highly benefitted by the opponent making stacks.
4. **Stack Size** – This heuristic uses the total size of the player’s stacks as a score.   
     
   This heuristic led to immense advantages for one player. In the tests run, this heuristic culminated with a staggering victory for one player. It is hypothesized that, as in reserve size, a high stack size represents a significant investment for a player (it allows them increased mobility and versatility while splitting).