**CMP682 ARTIFICIAL INTELLIGENCE TERM PROJECT REPORT**

**Rules of Game**

Aim of the game is to capture cells. The player who is captured more cells than other wins the game.

Figure . A screenshot of Capture Cell game

Player can capture cells which are neigbours.

When player clicks cell neighbour cells will be captured. (Not diagonal neigbours.)

**Beginning of Game**

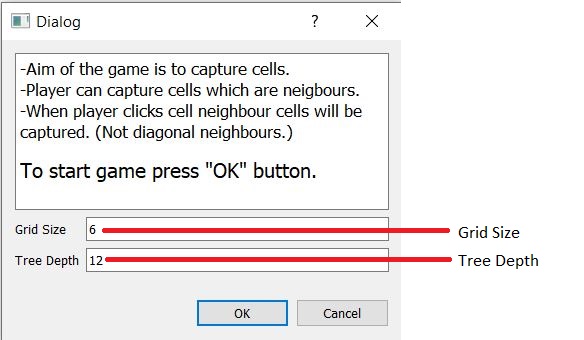
Player can arrange depth of minimax tree and dimensions of scenario on this screen.

Figure . A screenshot of beginning of Capture Cell game.

Player should be careful about that if they enter higher number to these lines, most probably game will be slower since the computation work will be increase too much. The computation work can be calculated as follows;

Computation Work = O(b^m) where b is number of possible moves and m is number of maximum depth (alpha-beta pruning is ignored). Obviously, pruning will reduce computation work but possible number of moves is too much and it increases exponential also number of possible moves increases till mid-game. Another important parameter is grid size. Grid size increases number of possible moves since grid size default value is a small number.

Let’s observe how number of possible moves affect computation work;

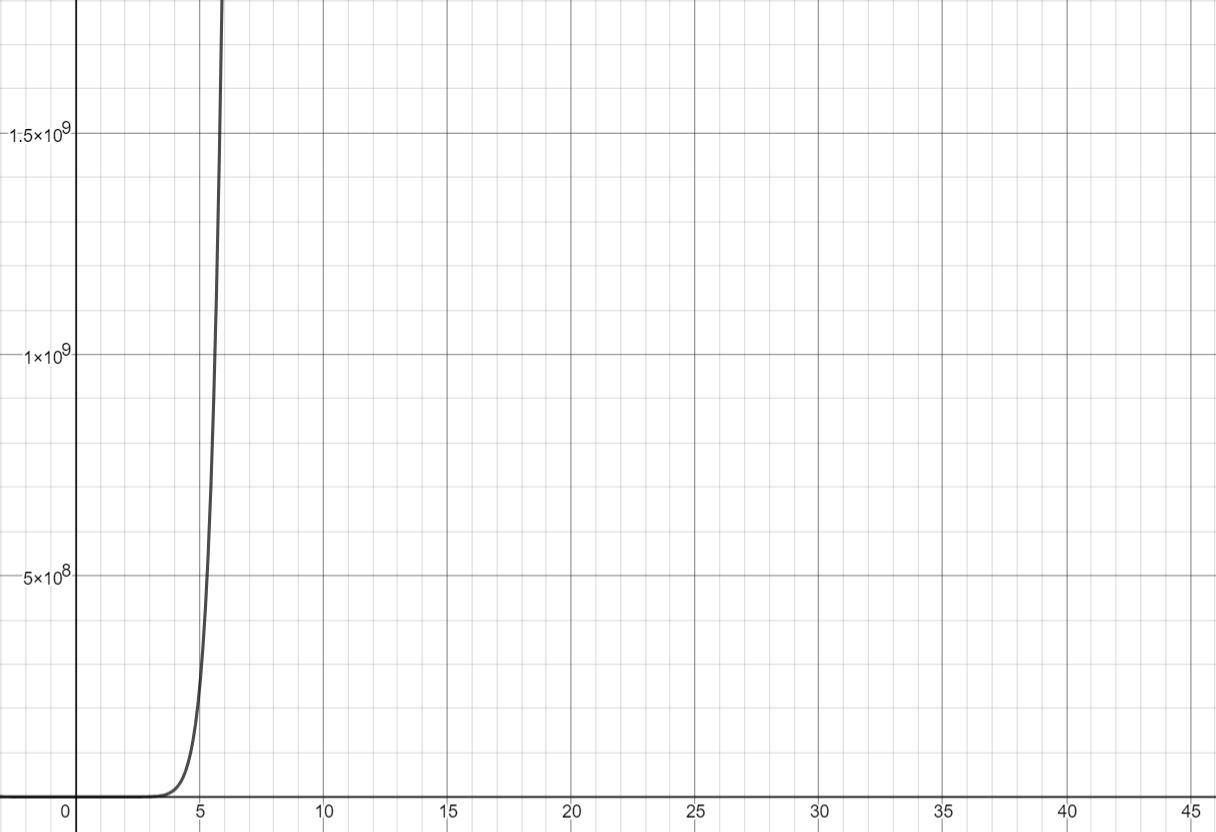


Figure . The effect of number of possible moves on complexity.

X-axis is the number of possible moves. As seen from graph number of possible moves increases computation work too much. (y-axis)

Let’s observe how tree depth affect computation work;

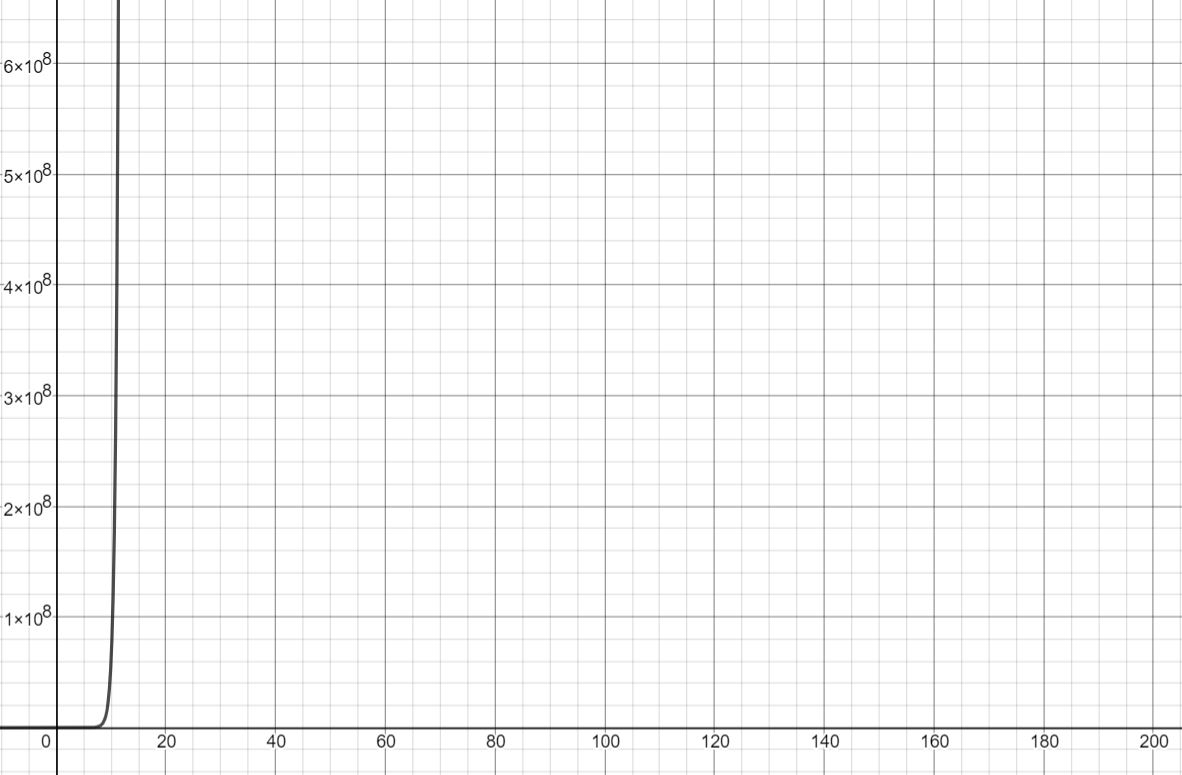


Figure . The affect of tree depth on computation work.

On this graph, number of possible moves accepted as 6. This is the graph of y = 6^x where x is tree depth. As seen on graph, tree depth increases computation work too much.