YACQ

Yet Another Conquest: An AI Based Board Warfare Game

Submitted By:

Arpit Kumar 11CS10007

Siddharth Rakesh 11CS30036

Sachin Kumar 11CS30043

Manav Sethi 11CS30044

Under the guidance of:

Prof. Pabitra Mitra

10/19/2014

YACQ

This game, 'Yet Another Conquest', has been designed as an example for illustrating the mechanism of the Minimax and Alpha-Beta Pruning algorithms, while providing an interactive environment to the users for playing against the bots employing these artificial intelligence techniques. Five pre-defined board configurations, namely Keren, Narvik, Sevastopol, Smolensk and Westerplatte, are used for comparing and contrasting the performance of both the algorithms.

Rules of the game:

- The game board is a 6x6 grid representing a city.
- Each square has a fixed point value between 1 and 99.
- There are two players, "blue" and "green". Each player takes turns: blue moves first, then green, then blue, etc.
- The object of the game is to be the player in the end with the largest total value of squares in their possession. That is, one wants to capture the squares worth the most points.
- The game ends when all the squares are occupied by all players since no more moves are left.
- Movement is always vertical and horizontal but never diagonal.
- Pieces can be conquered in the vertical and horizontal direction, but never the diagonal direction.
- The values of the squares can be changed for each game, but remain constant within a game.

In each turn, a player can make one of two moves:

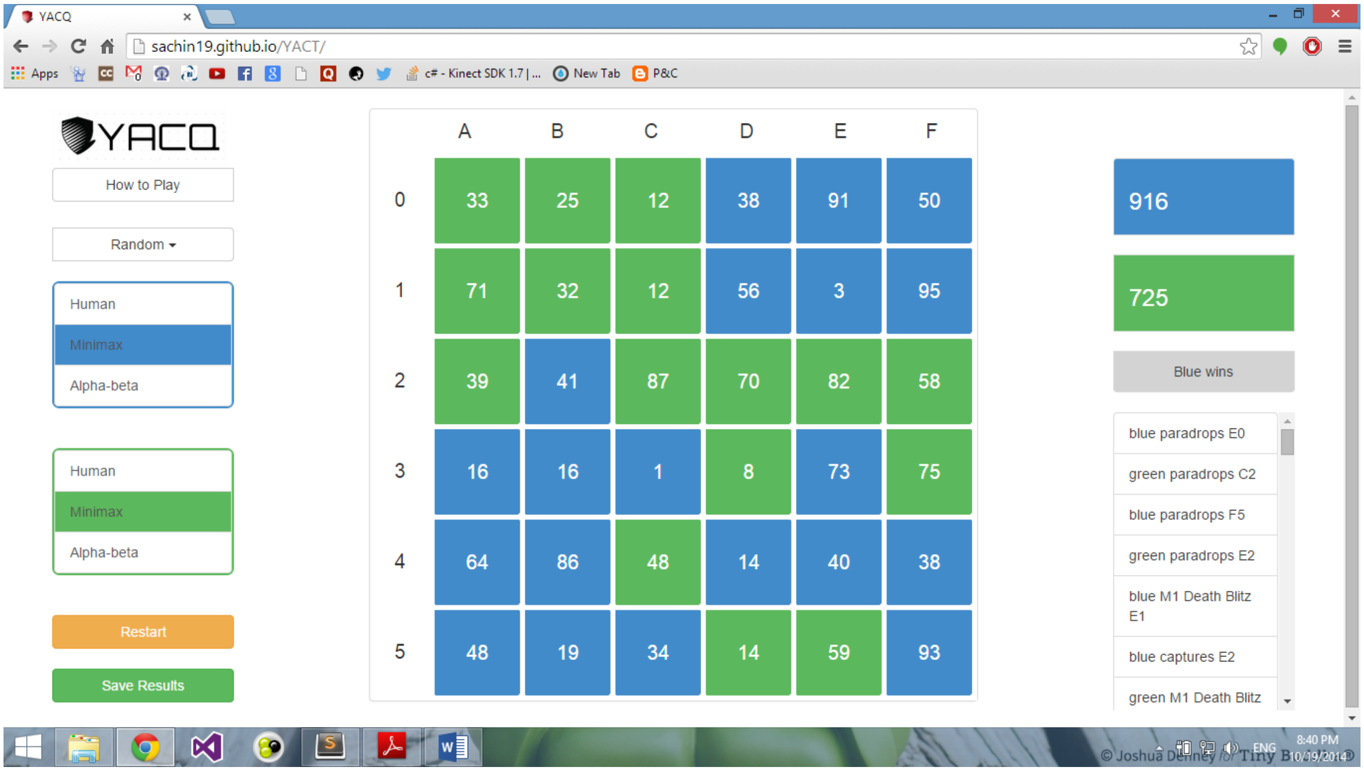
Commando Para Drop: You can take any open space on the board with a Para Drop. This will create a new piece on the board. This move can be made as many times as one wants to during the game, but only once per turn. A Commando Para Drop cannot conquer any pieces. It simply allows one to arbitrarily place a piece on any unoccupied square on the board. Once you have done a Para Drop, your turn is complete.

M1 Death Blitz: From any space you occupy on the board, you can take the one next to it (up, down, left, right, but not diagonally) if it is unoccupied. The space you originally held is still occupied. Thus, you get to create a new piece in the blitzed square. Any enemy touching the square you have taken is conquered and that square is turned to your side (you turn its piece to your side). An M1 Death Blitz can be done even if it will not conquer another piece. Once you have made this move, your turn is over.

Technologies involved:

Javascript, Bootstrap, HTML, CSS

Design



Screen shot for Yet Another Conquest in Google Chrome browser

Performance measure:

The performance measure used for judging the algorithms is whether they win against the opponent or not.

Heuristic:

The Heuristic used in the game for predicting the moves has been chosen to be the difference of the scores of both the players.

Based on this Heuristic, the artificial agent can adopt either of the two techniques for move prediction as described below:

Minimax Algorithm

Minimax is a decision rule used in decision theory, game theory, statistics and philosophy for *mini*mizing the possible loss for a worst case (*max*imum loss) scenario. Originally formulated for two-player zero-sum game theory, covering both the cases where players take alternate moves and those where they make simultaneous moves, it has also been extended to more complex games and to general decision making in the presence of uncertainty.

In the theory of simultaneous games, a minimax strategy is a mixed strategy which is part of the solution to a zero-sum game. In zero-sum games, the minimax solution is the same as the Nash equilibrium.

Minimax theorem:

The minimax theorem states:

For every two-person, zero-sum game with finitely many strategies, there exists a value V and a mixed strategy for each player, such that

(a) Given player 2's strategy, the best payoff possible for player 1 is V, and

(b) Given player 1's strategy, the best payoff possible for player 2 is −V.

In our game, Minimax algorithm evaluates to a depth of 3 levels or plies for predicting the required move, based on the heuristic mention above. Although a powerful technique to employ, an improvement, known as the Alpha Beta pruning techniques, renders the Minimax algorithm even more potent. The Alpha Beta pruning technique is described below.

Alpha-Beta Pruning

Alpha Beta pruning is a search algorithm that seeks to decrease the number of nodes that are evaluated by the minimax algorithm in its search tree. It is an adversarial search algorithm used commonly for machine playing of two-player games. It stops completely evaluating a move when at least one possibility has been found that proves the move to be worse than a previously examined move. Such moves need not be evaluated further. When applied to a standard minimax tree, it returns the same move as minimax would, but prunes away branches that cannot possibly influence the final decision, thereby increasing the capacity to evaluate further levels or plies.

The Alpha-Beta pruning algorithm in our game evaluates up to four plies, thereby giving a significant advantage over the Minimax algorithm, as is seen by the comparison of Matchups of the Minimax algorithm with the Alpha – Beta pruning algorithm.

Results

The detailed results have been submitted along with this project report; the main observations from the results are stated below:

Board Type	Matchup	Player	Total score of each player	Total Number of game nodes Expanded	Average number of nodes expanded per move	Average amount of time to make a move
Keren	Minimax vs Minimax	Blue	15	217758	12097.66	26.89
		Green	21	194754	10819.66	24.56
	Alpha Beta vs Alpha Beta	Blue	18	32977	1832.06	6.00
		Green	18	27948	1552.67	4.56
	Minimax vs Alpha Beta	Blue	14	217758	12097.67	22.28
		Green	22	30461	1692.28	8.83
	Alpha Beta vs Minimax	Blue	18	32977	1832.06	4.61
		Green	18	194754	10819.66	19.89
Narvik	Minimax vs Minimax	Blue	704	217758	12097.66	20.89
		Green	1096	194754	10819.66	18.78
	Alpha Beta vs Alpha Beta	Blue	900	68996	3833.11	8.50
		Green	900	77861	4325.61	9.89
	Minimax vs Alpha Beta	Blue	899	217758	12097.67	18.00
		Green	901	75391	4188.39	9.00
	Alpha Beta vs Minimax	Blue	1097	67451	3747.28	8.94
		Green	703	194754	10819.66	16.11
Sevastopol	Minimax vs Minimax	Blue	226	217758	12097.66	26.56
		Green	152	194754	10819.66	17.22
	Alpha Beta vs Alpha Beta	Blue	189	3774565	209698.06	188.83
		Green	189	3406105	189228.06	178.17
	Minimax vs Alpha Beta	Blue	148	217758	12097.67	20.28
		Green	230	3057887	169882.61	159.39
	Alpha Beta vs Minimax	Blue	219	4015347	223074.83	212.39
		Green	159	194754	10819.67	12.44
Smolensk	Minimax vs Minimax	Blue	535	217758	12097.66	18.00
		Green	1118	194754	10819.66	17.06
	Alpha Beta vs Alpha Beta	Blue	890	277568	15429.45	20.50
		Green	763	367624	20423.56	31.39
	Minimax vs Alpha Beta	Blue	645	217758	12097.67	18.33
		Green	1008	819921	45551.17	50.83
	Alpha Beta vs Minimax	Blue	983	233924	12995.78	27.89
		Green	670	194754	10819.66	18.33
Westerplatte	Minimax vs Minimax	Blue	31	217758	12097.66	21.89
		Green	41	194754	10819.66	16.28
	Alpha Beta vs Alpha Beta	Blue	33	294325	16351.39	27.22
		Green	39	176658	9814.33	18.94
	Minimax vs Alpha Beta	Blue	33	217758	12097.66	19.22
		Green	39	255978	14221	26.83
	Alpha Beta vs Minimax	Blue	34	328222	18234.56	26.00
		Green	38	194754	10819.66	16.89

Observations

As can be seen from the results, Alpha beta performs better than the simple Minimax algorithm almost every time.

The results for the number of nodes expanded and time taken for moves vary according to the board types, maintaining a trend within a game board.

Conclusion

Through the results, it has been established that the Alpha – Beta pruning algorithm provides significantly better performance than the simple Minimax algorithm.