



CS 440 Programming Assignment 4: Game Playing

[Boston University](#)

Instructor: [Prof. Margrit Betke](#)

Departments of [Computer Science](#) and [Mathematics](#)

Wei Wei

Teammates: Tianqi Xu, [Yehui Huang](#), Xiao Zhou

4/20/2016

Problem Definition

In this assignment, we implemented a strategy (alpha-beta pruning / Minimax) to play this game intelligently (as opposed to randomly or via brute-force search), which is a static evaluator with [] adversial algorithm for Atropos game. We packaged our scripts according to the requirements so that its executable can participate in an automatic tournament among the class.

Method and Implementation

1. function availmove(): Find all the available moves and put it in a 2d array

This function will return the 2-d int array with number 1 for circles that are available for the moves with the same index as (height,leftdistance), and with number 0 for circles that are not available for move. However, if the last Play is null(the script goes first), then it will return an empty array.

2. function bestmove(): Find the best move using minmax & alpha-beta pruning algorithm.

This function will go through all the colors with moves which will not lose the game and evaluate all possible moves by calling MinMax. If game is definitely going to lose, the bestmove will return the losing move.

3. function eval(): Evaluate the AtroposState for a certain move

This function will evaluate a certain move in the board.

Our evaluation method prefers a place where more of its neighbours were filled, therefore, the more of the options/circles were filled, the higher score the move will be given. Besides, different colors will have a bonus score while same colors will have a negative bonus score.

4. function minMax(): It returns the highest/lowest score based on the player.

This function implements the Minimax method with alpha-beta pruning. Therefore, it will stop searching along the tree if certain conditions were met. (e.g. alpha [less than] beta when it's our script's turn; vice versa when considering opponents' turn)

Experiments

Experiment number:

Variables/Inputs:

#1: Integer

Indicates the size of the game board

Required

range: ≥ 3 , recommended ≥ 6

#2: String

Indicates the script that performs player 1's algorithm

Optional, if not specified, default player will be chosen

range: standard inputs. (e.g. "java zhouxiaoPlayer")

#3: String

Indicates the script that performs player 2's algorithm

Optional, if not specified, default player will be choose

range: standard inputs. (e.g. "java zhouxiaoPlayer")

The results of experiments varies; however, our scripts win in most times

Results

In all experiments, where the variables were correctly entered and were within the specified range, our program runs without break.

For a tournament between random player and our script with 10 games, our script wins 9 games (90%)

For a tournament between ourself and our script with 10 games, our script wins 8 games (80%)

Results Table

Test runs	Inputs	Outputs/Results
Test 1 for Our Script vs Default Script	java AtroposGame 6 "java zhouxiaoPlayer"	

```
The game had the following moves:  
[Circle colored 3 at: (3, 4, 1).  
, Circle colored 1 at: (2, 4, 2).  
, Circle colored 1 at: (1, 4, 3).  
, Circle colored 1 at: (1, 3, 4).  
, Circle colored 2 at: (1, 2, 5).  
, Circle colored 1 at: (2, 1, 5).  
, Circle colored 3 at: (3, 1, 4).  
, Circle colored 2 at: (3, 2, 3).  
, Circle colored 3 at: (2, 3, 3).  
, Circle colored 3 at: (3, 3, 2).  
, Circle colored 2 at: (4, 3, 1).  
, Circle colored 2 at: (4, 2, 2).  
, Circle colored 3 at: (5, 1, 2).  
, Circle colored 3 at: (6, 1, 1).  
, Circle colored 3 at: (5, 2, 1).  
, Circle colored 3 at: (4, 1, 3).  
, Circle colored 3 at: (2, 5, 1).  
, Circle colored 1 at: (1, 5, 2).  
, Circle colored 1 at: (1, 6, 1).  
]  
Script has won in 19 moves!  
Final board:  
[ 3   2 ]  
[ 1   3   3 ]  
[ 3   3   3   2 ]  
[ 1   3   2   2   3 ]  
[ 3   3   2   3   3   2 ]  
[ 1   1   0   3   1   3   3 ]  
[ 3   0   2   1   1   1   1   2 ]  
[ 1   2   1   2   1   2   1 ]  
Last Play: (1, 1, 6, 1)
```

Test 2 for Our Script vs Default Script

```
java AtroposGame 7 "java zhouxiaoPlayer"
```

The game had the following moves:
[Circle colored 2 at: (2, 6, 1).
, Circle colored 1 at: (1, 6, 2).
, Circle colored 2 at: (1, 7, 1).
, Circle colored 1 at: (3, 3, 3).
, Circle colored 2 at: (4, 3, 2).
, Circle colored 2 at: (4, 2, 3).
, Circle colored 3 at: (4, 1, 4).
, Circle colored 1 at: (3, 1, 5).
, Circle colored 1 at: (2, 1, 6).
, Circle colored 1 at: (1, 2, 6).
, Circle colored 2 at: (1, 3, 5).
, Circle colored 1 at: (2, 2, 5).
, Circle colored 2 at: (2, 3, 4).
, Circle colored 1 at: (1, 4, 4).
, Circle colored 2 at: (2, 4, 3).
, Circle colored 2 at: (3, 4, 2).
, Circle colored 1 at: (2, 5, 2).
, Circle colored 2 at: (3, 5, 1).
, Circle colored 3 at: (4, 4, 1).
, Circle colored 3 at: (5, 3, 1).
, Circle colored 2 at: (5, 2, 2).
, Circle colored 3 at: (6, 2, 1).
, Circle colored 3 at: (7, 1, 1).
, Circle colored 3 at: (6, 1, 2).
, Circle colored 3 at: (5, 1, 3).
, Circle colored 2 at: (1, 5, 3).
, Circle colored 1 at: (1, 1, 7).
, Circle colored 2 at: (3, 2, 4).
]
Default has won in 28 moves!
Final board:

[1 3]
[3 3 2]
[1 3 3 3]
[3 3 2 3 2]
[1 3 2 2 3 3]
[3 1 2 1 2 2 2]
[1 1 1 2 2 1 2 3]
[3 1 1 2 1 2 1 2 2]
[1 2 1 2 1 2 1 2]

Last Play: (2, 3, 2, 4)

Test 3 for Our Script vs Default Script

```
java AtroposGame 7 "java zhouxiaoPlayer"
```

Default will go first.

The game had the following moves:
[Circle colored 3 at: (2, 2, 5).
, Circle colored 2 at: (1, 3, 5).
, Circle colored 2 at: (1, 2, 6).
, Circle colored 1 at: (1, 1, 7).
, Circle colored 1 at: (2, 1, 6).
]

Script has won in 5 moves!

Final board:

```
[1   3]  
[3   0   2]  
[1   0   0   3]  
[3   0   0   0   2]  
[1   0   0   0   0   3]  
[3   0   0   0   0   0   2]  
[1   1   3   0   0   0   0   3]  
[3   1   2   2   0   0   0   0   2]  
[1   2   1   2   1   2   1   2]
```

Last Play: (1, 2, 1, 6)

Test 4 for Our Script vs Default Script

```
java AtroposGame 8 "java zhouxiaoPlayer"
```

The game had the following moves:
[Circle colored 2 at: (1, 7, 2).
, Circle colored 1 at: (2, 6, 2).
, Circle colored 3 at: (3, 6, 1).
, Circle colored 1 at: (3, 5, 2).
, Circle colored 3 at: (4, 4, 2).
, Circle colored 3 at: (4, 5, 1).
, Circle colored 3 at: (5, 4, 1).
, Circle colored 2 at: (6, 3, 1).
, Circle colored 2 at: (6, 2, 2).
, Circle colored 2 at: (5, 3, 2).
, Circle colored 2 at: (4, 3, 3).
, Circle colored 1 at: (3, 3, 4).
, Circle colored 1 at: (2, 4, 4).
, Circle colored 1 at: (1, 4, 5).
, Circle colored 1 at: (2, 3, 5).
, Circle colored 1 at: (3, 2, 5).
, Circle colored 3 at: (2, 2, 6).
, Circle colored 1 at: (3, 1, 6).
, Circle colored 1 at: (4, 1, 5).
, Circle colored 1 at: (4, 2, 4).
, Circle colored 3 at: (5, 1, 4).
, Circle colored 1 at: (6, 1, 3).
, Circle colored 1 at: (7, 1, 2).
, Circle colored 2 at: (7, 2, 1).
, Circle colored 1 at: (8, 1, 1).
]

Script has won in 25 moves!

Final board:

[3 2]
[1 1 3]
[3 1 2 2]
[1 1 2 2 3]
[3 3 0 2 3 2]
[1 1 1 2 3 3 3]
[3 1 1 1 0 1 3 2]
[1 0 3 1 1 0 1 0 3]
[3 0 0 0 1 0 0 2 0 2]
[1 2 1 2 1 2 1 2 1]

Last Play: (1, 8, 1, 1)

Discussion

1. We have found from our experiments with our players that it is not always the case that a deeper look ahead will make the program perform better. When playing a board size of 6 with Default Player, we found our program has the highest winning rate with a look ahead depth of 6.
 2. Alpha-beta pruning for minimax algorithm can decrease the number of circles that are evaluated by the minimax algorithm in the search tree in most time. As an adversarial search, this algorithm is suitable for the Atropos game.
-

Conclusions

This assignment challenges our understanding of alpha-beta pruning technique for minimax algorithm and Java coding skills. After finished this assignment, we have learned a lot and may try implement better evaluation methods in the future.

Credits and Bibliography

References:

Notes from professor Betke's class

Skeleton code by professor [Kyle Burke](#)

Reference codes from GitHub [timmockPlayer.java](#) and [HowellEvaluator.java](#)

Discussed with teammates

Contributions:

Wei Wei : Implemented the Minmax with AB Pruning and eval based on the reference into the scriptPlayer

Yehui Huang: Implement the bestmove, test of the bugs and optimized the program

Tianqi Xu : Report & other function

Xiao Zhou : Report & other function

[Back](#)