

Assignment 3

Minimax and Alpha-beta pruning



Minimax code

```
109 class MinimaxAgent(MultiAgentSearchAgent):
110     n_ghosts = 0
111
112     def getAction(self, gameState):
113
114         self.n_ghosts = gameState.getNumAgents() - 1
115
116         def minimax(state, depth, is_maximizer):
117             # Because these guys apparently like fake depth, self.depth is not the amount of layers
118             # But the amount of times we need to do n layers
119             if(depth == self.depth*(self.n_ghosts+1) or state.isWin() or state.isLose()):
120                 return [self.evaluationFunction(state), None]
121
122             if(is_maximizer):
123                 #print("pacman can do", state.getLegalActions(0))
124                 curr_val = -9999999
125                 best_move = None
126                 # Maximizing layer, so we know we're pacman
127                 agent_num = 0
128                 for possible_move in state.getLegalActions(0):
129                     move_val = minimax(state.generateSuccessor(0, possible_move), depth+1, False)[0]
130
131                     if(move_val > curr_val):
132                         curr_val = move_val
133                         best_move = possible_move
134
135                 return [curr_val, best_move]
136             # Minimizing layer, ghost
137             else:
138                 curr_val = 9999999
139                 best_move = None
140                 # Figure out which ghost we are
141                 agent_num = (self.depth - (self.depth - depth)) % (self.n_ghosts+1)
142                 legal_moves = state.getLegalActions(agent_num)
143                 for possible_move in legal_moves:
144                     if(agent_num == self.n_ghosts):
145                         # Next layer is maximizing (pacman)
146                         move_val = minimax(state.generateSuccessor(agent_num, possible_move), depth+1, True)[0]
147                         if(move_val < curr_val):
148                             curr_val = move_val
149                             best_move = possible_move
150
151                     else:
152                         # Next layer is still minimizing (another ghost)
153                         move_val = minimax(state.generateSuccessor(agent_num, possible_move), depth+1, False)[0]
154                         if(move_val < curr_val):
155                             curr_val = move_val
156                             best_move = possible_move
157
158                 return [curr_val, best_move]
159
160         best_score, best_move = minimax(gameState, 0, True)
161         return best_move
```

Minimax output

```

PS C:\Users\Jorgen\Desktop\Datateknologi\TDT4136\A3> python autograder.py -q q2 --no-graphics
autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
import imp
Starting on 10-14 at 8:55:15

Question q2
=====

*** PASS: test_cases\q2\0-eval-function-lose-states-1.test
*** PASS: test_cases\q2\0-eval-function-lose-states-2.test
*** PASS: test_cases\q2\0-eval-function-win-states-1.test
*** PASS: test_cases\q2\0-eval-function-win-states-2.test
*** PASS: test_cases\q2\0-lecture-6-tree.test
*** PASS: test_cases\q2\0-small-tree.test
*** PASS: test_cases\q2\1-1-minmax.test
*** PASS: test_cases\q2\1-2-minmax.test
*** PASS: test_cases\q2\1-3-minmax.test
*** PASS: test_cases\q2\1-4-minmax.test
*** PASS: test_cases\q2\1-5-minmax.test
*** PASS: test_cases\q2\1-6-minmax.test
*** PASS: test_cases\q2\1-7-minmax.test
*** PASS: test_cases\q2\1-8-minmax.test
*** PASS: test_cases\q2\2-1a-vary-depth.test
*** PASS: test_cases\q2\2-1b-vary-depth.test
*** PASS: test_cases\q2\2-2a-vary-depth.test
*** PASS: test_cases\q2\2-2b-vary-depth.test
*** PASS: test_cases\q2\2-3a-vary-depth.test
*** PASS: test_cases\q2\2-3b-vary-depth.test
*** PASS: test_cases\q2\2-4a-vary-depth.test
*** PASS: test_cases\q2\2-4b-vary-depth.test
*** PASS: test_cases\q2\2-one-ghost-3level.test
*** PASS: test_cases\q2\3-one-ghost-4level.test
*** PASS: test_cases\q2\4-two-ghosts-3level.test
*** PASS: test_cases\q2\5-two-ghosts-4level.test
*** PASS: test_cases\q2\6-tied-root.test
*** PASS: test_cases\q2\7-1a-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-1b-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-1c-check-depth-one-ghost.test
*** PASS: test_cases\q2\7-2a-check-depth-two-ghosts.test
*** PASS: test_cases\q2\7-2b-check-depth-two-ghosts.test
*** PASS: test_cases\q2\7-2c-check-depth-two-ghosts.test
*** Running MinimaxAgent on smallClassic 1 time(s).
Pacman died! Score: 84
Average Score: 84.0
Scores:      84.0
Win Rate:    0/1 (0.00)
Record:      Loss
*** Finished running MinimaxAgent on smallClassic after 0 seconds.
*** PASS: test_cases\q2\8-pacman-game.test

### Question q2: 5/5 ###

Finished at 8:55:16

Provisional grades
=====
Question q2: 5/5
-----
Total: 5/5

Your grades are NOT yet registered. To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.

```

With alpha-beta pruning code

```

164 class AlphaBetaAgent(MultiAgentSearchAgent):
165     n_ghosts = 0
166
167     def getAction(self, gameState):
168         self.n_ghosts = gameState.getNumAgents() - 1
169
170     def minimax_ab(state, depth, is_maximizer, alpha, beta):
171         if(depth == self.depth*(self.n_ghosts+1) or state.isWin() or state.isLose()):
172             return [scoreEvaluationFunction(state), None]
173
174         if(is_maximizer):
175             #print("pacman can do", state.getLegalActions(0))
176             curr_val = -9999999
177             best_move = None
178             # Maximizing layer, so we know we're pacman
179             agent_num = 0
180             for possible_move in state.getLegalActions(0):
181                 move_val = minimax_ab(state.generateSuccessor(0, possible_move), depth+1, False, alpha, beta)[0]
182
183                 if(move_val > curr_val):
184                     curr_val = move_val
185                     best_move = possible_move
186
187                 alpha = max(alpha, move_val)
188                 if(beta < alpha):
189                     break
190
191             return [curr_val, best_move]
192         # Minimizing layer, ghost
193         else:
194             curr_val = 9999999
195             best_move = None
196             # Figure out which ghost we are
197             agent_num = (self.depth-(self.depth-depth)) % (self.n_ghosts+1)
198             legal_moves = state.getLegalActions(agent_num)
199             for possible_move in legal_moves:
200                 if(agent_num == self.n_ghosts):
201                     # Next layer is maximizing (pacman)
202                     move_val = minimax_ab(state.generateSuccessor(agent_num, possible_move), depth+1, True, alpha, beta)[0]
203                     if(move_val < curr_val):
204                         curr_val = move_val
205                         best_move = possible_move
206
207                     beta = min(beta, move_val)
208                     if(beta < alpha):
209                         break
210                 else:
211                     # Next layer is still minimizing (another ghost)
212                     move_val = minimax_ab(state.generateSuccessor(agent_num, possible_move), depth+1, False, alpha, beta)[0]
213                     if(move_val < curr_val):
214                         curr_val = move_val
215                         best_move = possible_move
216
217                     beta = min(beta, move_val)
218                     if(beta < alpha):
219                         break
220
221             return [curr_val, best_move]
222
223     best_score, best_move = minimax_ab(gameState, 0, True, -9999999, 9999999)
224     return best_move
225

```

Alpha-beta pruning output

```

PS C:\Users\Jorgen\Desktop\Datateknologi\TDT4136\A3> python autograder.py -q q3 --no-graphics
autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  import imp
Starting on 10-14 at 8:55:26

Question q3
=====

*** PASS: test_cases\q3\0-eval-function-lose-states-1.test
*** PASS: test_cases\q3\0-eval-function-lose-states-2.test
*** PASS: test_cases\q3\0-eval-function-win-states-1.test
*** PASS: test_cases\q3\0-eval-function-win-states-2.test
*** PASS: test_cases\q3\0-lecture-6-tree.test
*** PASS: test_cases\q3\0-small-tree.test
*** PASS: test_cases\q3\1-1-minmax.test
*** PASS: test_cases\q3\1-2-minmax.test
*** PASS: test_cases\q3\1-3-minmax.test
*** PASS: test_cases\q3\1-4-minmax.test
*** PASS: test_cases\q3\1-5-minmax.test
*** PASS: test_cases\q3\1-6-minmax.test
*** PASS: test_cases\q3\1-7-minmax.test
*** PASS: test_cases\q3\1-8-minmax.test
*** PASS: test_cases\q3\2-1a-vary-depth.test
*** PASS: test_cases\q3\2-1b-vary-depth.test
*** PASS: test_cases\q3\2-2a-vary-depth.test
*** PASS: test_cases\q3\2-2b-vary-depth.test
*** PASS: test_cases\q3\2-3a-vary-depth.test
*** PASS: test_cases\q3\2-3b-vary-depth.test
*** PASS: test_cases\q3\2-4a-vary-depth.test
*** PASS: test_cases\q3\2-4b-vary-depth.test
*** PASS: test_cases\q3\2-one-ghost-3level.test
*** PASS: test_cases\q3\3-one-ghost-4level.test
*** PASS: test_cases\q3\4-two-ghosts-3level.test
*** PASS: test_cases\q3\5-two-ghosts-4level.test
*** PASS: test_cases\q3\6-tied-root.test
*** PASS: test_cases\q3\7-1a-check-depth-one-ghost.test
*** PASS: test_cases\q3\7-1b-check-depth-one-ghost.test
*** PASS: test_cases\q3\7-1c-check-depth-one-ghost.test
*** PASS: test_cases\q3\7-2a-check-depth-two-ghosts.test
*** PASS: test_cases\q3\7-2b-check-depth-two-ghosts.test
*** PASS: test_cases\q3\7-2c-check-depth-two-ghosts.test
*** Running AlphaBetaAgent on smallClassic 1 time(s).
Pacman died! Score: 84
Average Score: 84.0
Scores:      84.0
Win Rate:    0/1 (0.00)
Record:      Loss
*** Finished running AlphaBetaAgent on smallClassic after 0 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases\q3\8-pacman-game.test

### Question q3: 5/5 ###

Finished at 8:55:27

Provisional grades
=====
Question q3: 5/5
-----
Total: 5/5

Your grades are NOT yet registered. To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.

```