CMPE 540

ARTIFICIAL INTELLIGENCE

PROJECT #3

GENERAL GAME PLAYER

Emin Şenay
2002103907

Atilla Soner Balkır
2002101414

Introduction

This project is about designing a General Game Player, which is able to communicate with the Stanford’s Game server and play various games using a general game playing algorithm.

The game server sends the description of the game to the clients in the beginning of each game. The amount of time the client will have initially and in between the turns will be sent as STARTCLOCK and PLAYCLOCK variables. The player tries to come up with a reasonable move before PLAYCLOCK ends. Furthermore, it also understands the game rules at the beginning of the game. STARTCLOCK is the amount of time that can be used for this purpose.

After the client chooses and sends a legal action, the game server sends the client the chosen move back. Then, the client compares it with its intended move. If a legal action cannot be provided on time, a random move is played by the game server.

The client program also updates the current state of the game according to the moves that come back from the game server accordingly.

Structure

The game is implemented in the Java programming language. The Java program given in Dresden's website provides a basis for the communication parts of the client. Apart from this client, the JavaProver package which is present at the Stanford’s Game Server is used to parse the server responses that include the game description as well as the moves of the client and the opponent. The Java prover class also performs the necessary operations given below:

- Given a state of the game and a legal move, calculates the next state.
- Given a state of the game, checks whether the current state is a terminal state or not.
- Given a state of the game, checks whether the current state is a goal state or not.
- Given a state of the game, calculates the goal value, (i.e. utility value) of that state which is necessary for any game playing algorithm.

Implementation Details

The Java classes provided by Dresden University are used to communicate with the game server. The main class used for this purpose is GamePlayer.java which includes the following methods inside:

- commandPlay(java.lang.String msg) this method is called once for each move. Inside this method, the client runs the game playing algorithms and tries to come up with a reasonable move, considering the current state of the game and available legal moves. Since the amount of time that the client is able to ***think*** is limited by PLAYCLOCK, it must return a reasonable move before the time ends.

- commandStart(java.lang.String msg) this method is called when a new match begins. When a new match begins, the client parses the game description and then initializes the number of players, the roles of each player. Then it analyzes the rules of the game and returns the server before the time STARTCLOCK ends.

- commandStop(java.lang.String msg) this method is called if the match is over. It simply outputs a message to the screen informing the user that the game is over.

- main(java.lang.String[] args) starts the game player and waits for messages from the game master Command line options: [port]

Apart from these methods that are used to communicate with the game server, this class also includes the following methods, which are implemented in order to play the game, (i.e. analyze the search space and come up with a reasonable move)

- **MiniMaxDecision(int numSteps, boolean singlePlayer, Atom playerName, Atom opponent, int depth):** this is the main method used to calculate a reasonable move using minimax algorithm for both single as well as multi player games. The parameters taken as input are as follows:

  numSteps: number of steps required to calculate the move.
  singleplayer: boolean variable that becomes true or false.
  playerName: the name of the player represented by the client.
  opponent: the name of the opponent.
  depth: maximum depth that the algorithm is allowed to go. This part is explained in the algorithm section.

- **maxValue(GameState state, int numSteps, boolean singlePlayer, Atom playerName, Atom opponentName, int depth, int currDepth):** this method is used to calculate a max value, which is required for the minimax algorithm. The parameters taken as input are as follows:

  state: current state of the game.
  numSteps: number of steps required to calculate the move.
  singleplayer: boolean variable that becomes true or false.
  playerName: the name of the player represented by the client.
  opponent: the name of the opponent.
  depth: maximum depth that the algorithm is allowed to go. This part is explained in the algorithm section.
  currDepth: current depth of the algorithm. This is used to make sure that the current depth does not exceed the maximum depth allowed.

- **minValue(GameState state, int numSteps, boolean singlePlayer, Atom playerName, Atom opponentName, int depth, int currDepth):** this method is used to calculate a mi value, which is required for the minimax algorithm. The parameters taken as input are as follows:

  state: current state of the game.
  numSteps: number of steps required to calculate the move.
  singleplayer: boolean variable that becomes true or false.
  playerName: the name of the player represented by the client.
  opponent: the name of the opponent.
  depth: maximum depth that the algorithm is allowed to go. This part is explained in the algorithm section.
  currDepth: current depth of the algorithm. This is used to make sure that the current depth does not exceed the maximum depth allowed.

- **getGoalValue(Atom atm):** This method is used to get the utility of the current state of the game. If the current state is a terminal state and the client is in a winning position, this method returns a maximum positive value. If the current state is a terminal state and the client is in a losing position, this method returns the minimum negative value. The parameters taken as input are as follows:

  atm: the name of the player represented by the client.

Game Playing Algorithm

The game playing algorithm used by the client is ***MiniMax*** search together with ***Iterative Deepening***. In the early versions of the project, we did not use iterative deepening and minimax was the only algorithm used. However, since some games have extremely huge state spaces, using minimax alone usually resulted in incomplete searches that forced the client to make a random move.

The minimax algorithm used in the client is a bit different than the usual minimax algorithm which is designed for multiplayer games. Since the general game player should also able to play the single player games as well, we have made some modifications in the algorithm. Normally, in a multiplayer game, the minValue method returns the minimum value (the opponent's best move that results in the minimum gain of the player) of each node, and the maxValue method returns the maximum one that results in the maximum gain of the player.

However, in a single player game, at each step the player makes a new move. Hence, the minimax algorithm should run in order to maximize the utility of the player *at each step.* In order to achieve this property in the General Game Player client, we modified the minimax algorithm in such a way that, if the game is a single player game, maxValue method calls the maxValue method instead of the minValue method. In other words, the minimax algorithm has been modified to run in *maximax* mode.

Furthermore, the minimax algorithm usually runs in order to maximize the utility of the player but often ignores to prevent the opponent from reaching a winning state. To implement a General Game Player client that is programmed to both maximizing its utility as well as minimizing the opponent's utility we have implemented a goal function which returns negative values if the opponent is in an advantageous state and positive values if the player is in an advantageous state.

From the description of the project and the General Game Player given at the Stanford's we site, we know that the client has a limited amount of time to calculate a reasonable move. However, minimax performs a full search on the state space and it leads to a high time complexity. Because of this, the client is usually not able to finish the search and gives a random or unreasonable move as output. To prevent this, we have added the ***Iterative Deepening Algorithm*** to the minimax algorithm and modified the minimax code in such a way that instead of exploring all the states, it searches the game states up to a certain depth. The Iterative Deepening Algorithm starts to run the minimax algorithm for depth 1 and increases the depth one by one at each step. If at any step the time is over or all states are searched, the algorithm terminates returning the best move it has found.

Input Output of the Program

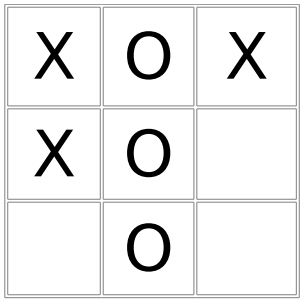
Game 1 – TicTacToe

Client: O Player

NanoHTTPD is listening on port 4000
POST / HTTP/1.0
1:53:57 AM EET
Command: (START MATCH.3378412393 OPLAYER ((ROLE XPLAYER) (ROLE OPLAYER) (INIT (CELL 1 1 B)) (INIT (CELL 1 2 B)) (INIT (CELL 1 3 B)) (INIT (CELL 2 1 B)) (INIT (CELL 2 2 B)) (INIT (CELL 2 3 B)) (INIT (CELL 3 1 B)) (INIT (CELL 3 2 B)) (INIT (CELL 3 3 B)) (INIT (CONTROL XPLAYER)) (<= (NEXT (CELL ?M ?N X)) (DOES XPLAYER (MARK ?M ?N)) (TRUE (CELL ?M ?N B))) (<= (NEXT (CELL ?M ?N O)) (DOES OPLAYER (MARK ?M ?N)) (TRUE (CELL ?M ?N B))) (<= (NEXT (CELL ?M ?N ?W)) (TRUE (CELL ?M ?N ?W)) (DISTINCT ?W B)) (<= (NEXT (CELL ?M ?N B)) (DOES ?W (MARK ?J ?K)) (TRUE (CELL ?M ?N B)) (OR (DISTINCT ?M ?J) (DISTINCT ?N ?K))) (<= (NEXT (CONTROL XPLAYER)) (TRUE (CONTROL OPLAYER))) (<= (NEXT (CONTROL OPLAYER)) (TRUE (CONTROL XPLAYER))) (<= (ROW ?M ?X) (TRUE (CELL ?M 1 ?X)) (TRUE (CELL ?M 2 ?X)) (TRUE (CELL ?M 3 ?X))) (<= (COLUMN ?N ?X) (TRUE (CELL 1 ?N ?X)) (TRUE (CELL 2 ?N ?X)) (TRUE (CELL 3 ?N ?X))) (<= (DIAGONAL ?X) (TRUE (CELL 1 1 ?X)) (TRUE (CELL 2 2 ?X)) (TRUE (CELL 3 3 ?X))) (<= (DIAGONAL ?X) (TRUE (CELL 1 3 ?X)) (TRUE (CELL 2 2 ?X)) (TRUE (CELL 3 1 ?X))) (<= (LINE ?X) (ROW ?M ?X)) (<= (LINE ?X) (COLUMN ?M ?X)) (<= (LINE ?X) (DIAGONAL ?X)) (<= OPEN (TRUE (CELL ?M ?N B))) (<= (LEGAL ?W (MARK ?X ?Y)) (TRUE (CELL ?X ?Y B)) (TRUE (CONTROL ?W))) (<= (LEGAL XPLAYER NOOP) (TRUE (CONTROL OPLAYER))) (<= (LEGAL OPLAYER NOOP) (TRUE (CONTROL XPLAYER))) (<= (GOAL XPLAYER 100) (LINE X)) (<= (GOAL XPLAYER 50) (NOT (LINE X)) (NOT (LINE O)) (NOT OPEN)) (<= (GOAL XPLAYER 0) (LINE O)) (<= (GOAL OPLAYER 100) (LINE O)) (<= (GOAL OPLAYER 50) (NOT (LINE X)) (NOT (LINE O)) (NOT OPEN)) (<= (GOAL OPLAYER 0) (LINE X)) (<= TERMINAL (LINE X)) (<= TERMINAL (LINE O)) (<= TERMINAL (NOT OPEN))) 5 10)
********** Game Initialization of match MATCH.3378412393 **********
Player XPLAYER
Legal moves: (DOES XPLAYER (MARK 1 1)) (DOES XPLAYER (MARK 1 2)) (DOES XPLAYER (MARK 1 3)) (DOES XPLAYER (MARK 2 1)) (DOES XPLAYER (MARK 2 2)) (DOES XPLAYER (MARK 2 3)) (DOES XPLAYER (MARK 3 1)) (DOES XPLAYER (MARK 3 2)) (DOES XPLAYER (MARK 3 3))
No goal values found.
Player OPLAYER
Legal moves: (DOES OPLAYER NOOP)
No goal values found.
My role is OPLAYER
Game is in non-terminal state.
********** End of Inıtialization of match MATCH.3378412393 **********
1:53:57 AM EET
Response:READY
POST / HTTP/1.0
1:54:13 AM EET
Command: (PLAY MATCH.3378412393 NIL)
********** next turn **********
my legal moves are: (DOES OPLAYER NOOP)
My move is: NOOP
********** end of turn **********
1:54:13 AM EET
Response:NOOP
POST / HTTP/1.0
1:54:29 AM EET
Command: (PLAY MATCH.3378412393 ((MARK 1 1) NOOP))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 1 1 X)) (TRUE (CELL 1 2 B)) (TRUE (CELL 1 3 B)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 1 1 B)) (TRUE (CELL 1 2 B)) (TRUE (CELL 1 3 B)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES OPLAYER (MARK 1 2)) (DOES OPLAYER (MARK 1 3)) (DOES OPLAYER (MARK 2 1)) (DOES OPLAYER (MARK 2 2)) (DOES OPLAYER (MARK 2 3)) (DOES OPLAYER (MARK 3 1)) (DOES OPLAYER (MARK 3 2)) (DOES OPLAYER (MARK 3 3))
end of depth 3
end of depth 4
end of depth 5
My move is: (MARK 1 2)
********** end of turn **********
1:54:39 AM EET
Response:(MARK 1 2)
POST / HTTP/1.0
1:54:45 AM EET

Command: (PLAY MATCH.3378412393 (NOOP (MARK 1 2)))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 1 3 B)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 1 1 X)) (TRUE (CELL 1 2 B)) (TRUE (CELL 1 3 B)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES OPLAYER NOOP)
My move is: NOOP
********** end of turn **********
1:54:45 AM EET
Response:NOOP
POST / HTTP/1.0
1:55:01 AM EET
Command: (PLAY MATCH.3378412393 ((MARK 1 3) NOOP))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 1 3 B)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES OPLAYER (MARK 2 1)) (DOES OPLAYER (MARK 2 2)) (DOES OPLAYER (MARK 2 3)) (DOES OPLAYER (MARK 3 1)) (DOES OPLAYER (MARK 3 2)) (DOES OPLAYER (MARK 3 3))
end of depth 3
end of depth 4
end of depth 5
end of depth 6
My move is: (MARK 2 2)
********** end of turn **********
1:55:08 AM EET
Response:(MARK 2 2)
POST / HTTP/1.0
1:55:17 AM EET
Command: (PLAY MATCH.3378412393 (NOOP (MARK 2 2)))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 2 2 O)) (TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES OPLAYER NOOP)
My move is: NOOP
********** end of turn **********
1:55:17 AM EET
Response:NOOP
POST / HTTP/1.0
1:55:33 AM EET
Command: (PLAY MATCH.3378412393 ((MARK 2 1) NOOP))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 2 1 X)) (TRUE (CELL 2 2 O)) (TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 2 2 O)) (TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES OPLAYER (MARK 2 3)) (DOES OPLAYER (MARK 3 1)) (DOES OPLAYER (MARK 3 2)) (DOES OPLAYER (MARK 3 3))
end of depth 3
end of depth 4
end of depth 5
end of depth 6
end of depth 7
end of depth 8
end of depth 9
end of depth 10
end of depth 11
end of depth 12
end of depth 13
end of depth 14
end of depth 15
end of depth 16

end of depth 17
end of depth 18
end of depth 19
end of depth 20
end of depth 21
end of depth 22
end of depth 23
end of depth 24
end of depth 25
end of depth 26
end of depth 27
end of depth 28
end of depth 29
end of depth 30
end of depth 31
end of depth 32
end of depth 33
end of depth 34
end of depth 35
end of depth 36
My move is: (MARK 3 2)
********** end of turn **********
1:55:38 AM EET
Response:(MARK 3 2)
POST / HTTP/1.0
1:55:48 AM EET
Command: (STOP MATCH.3378412393 (NOOP (MARK 3 2)))
********** end of game, exiting... **********



Game 2 – TicTacToe

Client: X Player

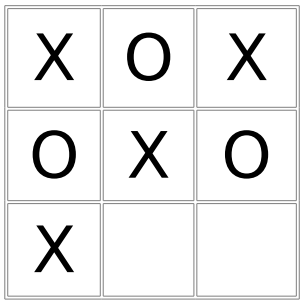
NanoHTTPD is listening on port 4000
POST / HTTP/1.0
2:00:43 AM EET
Command: (START MATCH.3378412793 XPLAYER ((ROLE XPLAYER) (ROLE OPLAYER) (INIT (CELL 1 1 B)) (INIT (CELL 1 2 B)) (INIT (CELL 1 3 B)) (INIT (CELL 2 1 B)) (INIT (CELL 2 2 B)) (INIT (CELL 2 3 B)) (INIT (CELL 3 1 B)) (INIT (CELL 3 2 B)) (INIT (CELL 3 3 B)) (INIT (CONTROL XPLAYER)) (<= (NEXT (CELL ?M ?N X)) (DOES XPLAYER (MARK ?M ?N)) (TRUE (CELL ?M ?N B))) (<= (NEXT (CELL ?M ?N O)) (DOES OPLAYER (MARK ?M ?N)) (TRUE (CELL ?M ?N B))) (<= (NEXT (CELL ?M ?N ?W)) (TRUE (CELL ?M ?N ?W)) (DISTINCT ?W B)) (<= (NEXT (CELL ?M ?N B)) (DOES ?W (MARK ?J ?K)) (TRUE (CELL ?M ?N B)) (OR (DISTINCT ?M ?J) (DISTINCT ?N ?K))) (<= (NEXT (CONTROL XPLAYER)) (TRUE (CONTROL OPLAYER))) (<= (NEXT (CONTROL OPLAYER)) (TRUE (CONTROL XPLAYER))) (<= (ROW ?M ?X) (TRUE (CELL ?M 1 ?X)) (TRUE (CELL ?M 2 ?X)) (TRUE (CELL ?M 3 ?X))) (<= (COLUMN ?N ?X) (TRUE (CELL 1 ?N ?X)) (TRUE (CELL 2 ?N ?X)) (TRUE (CELL 3 ?N ?X))) (<= (DIAGONAL ?X) (TRUE (CELL 1 1 ?X)) (TRUE (CELL 2 2 ?X)) (TRUE (CELL 3 3 ?X))) (<= (DIAGONAL ?X) (TRUE (CELL 1 3 ?X)) (TRUE (CELL 2 2 ?X)) (TRUE (CELL 3 1 ?X))) (<= (LINE ?X) (ROW ?M ?X)) (<= (LINE ?X) (COLUMN ?M ?X)) (<= (LINE ?X) (DIAGONAL ?X)) (<= OPEN (TRUE (CELL ?M ?N B))) (<= (LEGAL ?W (MARK ?X ?Y)) (TRUE (CELL ?X ?Y B)) (TRUE (CONTROL ?W))) (<= (LEGAL XPLAYER NOOP) (TRUE (CONTROL OPLAYER))) (<= (LEGAL OPLAYER NOOP) (TRUE (CONTROL XPLAYER))) (<= (GOAL XPLAYER 100) (LINE X)) (<= (GOAL XPLAYER 50) (NOT (LINE X)) (NOT (LINE O)) (NOT OPEN)) (<= (GOAL XPLAYER 0) (LINE O)) (<= (GOAL OPLAYER 100) (LINE O)) (<= (GOAL OPLAYER 50) (NOT (LINE X)) (NOT (LINE O)) (NOT OPEN)) (<= (GOAL OPLAYER 0) (LINE X)) (<= TERMINAL (LINE X)) (<= TERMINAL (LINE O)) (<= TERMINAL (NOT OPEN))) 5 10)
********** Game Initialization of match MATCH.3378412793 **********
Player XPLAYER
Legal moves: (DOES XPLAYER (MARK 1 1)) (DOES XPLAYER (MARK 1 2)) (DOES XPLAYER (MARK 1 3)) (DOES

XPLAYER (MARK 2 1)) (DOES XPLAYER (MARK 2 2)) (DOES XPLAYER (MARK 2 3)) (DOES XPLAYER (MARK 3 1)) (DOES XPLAYER (MARK 3 2)) (DOES XPLAYER (MARK 3 3))
No goal values found.
Player OPLAYER
Legal moves: (DOES OPLAYER NOOP)
No goal values found.
My role is XPLAYER
Game is in non-terminal state.
********** End of Inıtialization of match MATCH.3378412793 **********
2:00:43 AM EET
Response:READY
POST / HTTP/1.0
2:01:00 AM EET
Command: (PLAY MATCH.3378412793 NIL)
********** next turn **********
my legal moves are: (DOES XPLAYER (MARK 1 1)) (DOES XPLAYER (MARK 1 2)) (DOES XPLAYER (MARK 1 3)) (DOES XPLAYER (MARK 2 1)) (DOES XPLAYER (MARK 2 2)) (DOES XPLAYER (MARK 2 3)) (DOES XPLAYER (MARK 3 1)) (DOES XPLAYER (MARK 3 2)) (DOES XPLAYER (MARK 3 3))
end of depth 3
end of depth 4
end of depth 5
My move is: (MARK 1 1)
********** end of turn **********
2:01:10 AM EET
Response:(MARK 1 1)
POST / HTTP/1.0
2:01:17 AM EET
Command: (PLAY MATCH.3378412793 ((MARK 1 1) NOOP))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 1 1 X)) (TRUE (CELL 1 2 B)) (TRUE (CELL 1 3 B)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 1 1 B)) (TRUE (CELL 1 2 B)) (TRUE (CELL 1 3 B)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES XPLAYER NOOP)
My move is: NOOP
********** end of turn **********
2:01:17 AM EET
Response:NOOP
POST / HTTP/1.0
2:01:34 AM EET
Command: (PLAY MATCH.3378412793 (NOOP (MARK 1 2)))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 1 3 B)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 1 1 X)) (TRUE (CELL 1 2 B)) (TRUE (CELL 1 3 B)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES XPLAYER (MARK 1 3)) (DOES XPLAYER (MARK 2 1)) (DOES XPLAYER (MARK 2 2)) (DOES XPLAYER (MARK 2 3)) (DOES XPLAYER (MARK 3 1)) (DOES XPLAYER (MARK 3 2)) (DOES XPLAYER (MARK 3 3))
end of depth 3
end of depth 4
end of depth 5
My move is: (MARK 1 3)
********** end of turn **********
2:01:40 AM EET
Response:(MARK 1 3)
POST / HTTP/1.0
2:01:51 AM EET
Command: (PLAY MATCH.3378412793 ((MARK 1 3) NOOP))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 1 3 B)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES XPLAYER NOOP)
My move is: NOOP
********** end of turn **********
2:01:51 AM EET
Response:NOOP
POST / HTTP/1.0

2:02:08 AM EET
Command: (PLAY MATCH.3378412793 (NOOP (MARK 2 1)))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 2 1 O)) (TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 1 B)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES XPLAYER (MARK 2 2)) (DOES XPLAYER (MARK 2 3)) (DOES XPLAYER (MARK 3 1)) (DOES XPLAYER (MARK 3 2)) (DOES XPLAYER (MARK 3 3))
end of depth 3
end of depth 4
end of depth 5
end of depth 6
end of depth 7
end of depth 8
end of depth 9
end of depth 10
My move is: (MARK 2 2)
********** end of turn **********
2:02:14 AM EET
Response:(MARK 2 2)
POST / HTTP/1.0
2:02:25 AM EET
Command: (PLAY MATCH.3378412793 ((MARK 2 2) NOOP))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 2 2 X)) (TRUE (CELL 2 1 O)) (TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 2 1 O)) (TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 2 B)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES XPLAYER NOOP)
My move is: NOOP
********** end of turn **********
2:02:25 AM EET
Response:NOOP
POST / HTTP/1.0
2:02:42 AM EET
Command: (PLAY MATCH.3378412793 (NOOP (MARK 2 3)))
********** next turn **********
Next state: < GameState: {CONTROL=(TRUE (CONTROL XPLAYER)) , CELL=(TRUE (CELL 2 3 O)) (TRUE (CELL 2 2 X)) (TRUE (CELL 2 1 O)) (TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
GGP's old state after Simulate:
< GameState: {CONTROL=(TRUE (CONTROL OPLAYER)) , CELL=(TRUE (CELL 2 2 X)) (TRUE (CELL 2 1 O)) (TRUE (CELL 1 3 X)) (TRUE (CELL 1 2 O)) (TRUE (CELL 1 1 X)) (TRUE (CELL 2 3 B)) (TRUE (CELL 3 1 B)) (TRUE (CELL 3 2 B)) (TRUE (CELL 3 3 B)) } >
my legal moves are: (DOES XPLAYER (MARK 3 1)) (DOES XPLAYER (MARK 3 2)) (DOES XPLAYER (MARK 3 3))
end of depth 3
end of depth 4
end of depth 5
end of depth 6
end of depth 7
end of depth 8
end of depth 9
end of depth 10
end of depth 11
end of depth 12
end of depth 13
end of depth 14
end of depth 15
end of depth 16
end of depth 17
end of depth 18
end of depth 19
end of depth 20
end of depth 21
end of depth 22
end of depth 23
end of depth 24
end of depth 25
end of depth 26
end of depth 27
end of depth 28

end of depth 29
end of depth 30
end of depth 31
end of depth 32
end of depth 33
end of depth 34
end of depth 35
end of depth 36
end of depth 37
end of depth 38
end of depth 39
end of depth 40
end of depth 41
end of depth 42
end of depth 43
end of depth 44
end of depth 45
end of depth 46
end of depth 47
end of depth 48
end of depth 49
end of depth 50
end of depth 51
end of depth 52
end of depth 53
end of depth 54
end of depth 55
end of depth 56
end of depth 57
end of depth 58
end of depth 59
end of depth 60
end of depth 61
end of depth 62
end of depth 63
end of depth 64
end of depth 65
end of depth 66
end of depth 67
end of depth 68
end of depth 69
end of depth 70
end of depth 71
end of depth 72
end of depth 73
end of depth 74
end of depth 75
end of depth 76
end of depth 77
end of depth 78
end of depth 79
end of depth 80
end of depth 81
end of depth 82
end of depth 83
end of depth 84
end of depth 85
end of depth 86
end of depth 87
end of depth 88
end of depth 89
end of depth 90
end of depth 91
end of depth 92
end of depth 93
end of depth 94
end of depth 95
end of depth 96
end of depth 97
end of depth 98
end of depth 99
end of depth 100
end of depth 101
end of depth 102
end of depth 103
end of depth 104
end of depth 105
end of depth 106

end of depth 107
end of depth 108
end of depth 109
end of depth 110
end of depth 111
end of depth 112
end of depth 113
end of depth 114
end of depth 115
end of depth 116
end of depth 117
end of depth 118
end of depth 119
end of depth 120
end of depth 121
end of depth 122
end of depth 123
end of depth 124
end of depth 125
end of depth 126
end of depth 127
end of depth 128
end of depth 129
end of depth 130
end of depth 131
end of depth 132
end of depth 133
end of depth 134
end of depth 135
end of depth 136
end of depth 137
end of depth 138
end of depth 139
end of depth 140
end of depth 141
end of depth 142
end of depth 143
end of depth 144
end of depth 145
end of depth 146
end of depth 147
end of depth 148
end of depth 149
end of depth 150
end of depth 151
end of depth 152
end of depth 153
end of depth 154
end of depth 155
end of depth 156
end of depth 157
end of depth 158
end of depth 159
end of depth 160
end of depth 161
end of depth 162
end of depth 163
end of depth 164
end of depth 165
end of depth 166
end of depth 167
end of depth 168
end of depth 169
My move is: (MARK 3 1)
********** end of turn **********
2:02:47 AM EET
Response:(MARK 3 1)
POST / HTTP/1.0
2:02:59 AM EET
Command: (STOP MATCH.3378412793 ((MARK 3 1) NOOP))
********** end of game, exiting... **********



Game 3 – Maze

Client: Robot

NanoHTTPD is listening on port 4000
POST / HTTP/1.0
2:08:29 AM EET
Command: (START MATCH.3378413291 ROBOT ((ROLE ROBOT) (INIT (CELL A)) (INIT (GOLD C)) (INIT (STEP 1)) (<= (NEXT (CELL ?Y)) (DOES ROBOT MOVE) (TRUE (CELL ?X)) (ADJACENT ?X ?Y)) (<= (NEXT (CELL ?X)) (DOES ROBOT GRAB) (TRUE (CELL ?X))) (<= (NEXT (CELL ?X)) (DOES ROBOT DROP) (TRUE (CELL ?X))) (<= (NEXT (GOLD ?X)) (DOES ROBOT MOVE) (TRUE (GOLD ?X))) (<= (NEXT (GOLD I)) (DOES ROBOT GRAB) (TRUE (CELL ?X)) (TRUE (GOLD ?X))) (<= (NEXT (GOLD I)) (DOES ROBOT GRAB) (TRUE (GOLD I))) (<= (NEXT (GOLD ?Y)) (DOES ROBOT GRAB) (TRUE (CELL ?X)) (TRUE (GOLD ?Y)) (DISTINCT ?X ?Y)) (<= (NEXT (GOLD ?X)) (DOES ROBOT DROP) (TRUE (CELL ?X)) (TRUE (GOLD I))) (<= (NEXT (GOLD ?X)) (DOES ROBOT DROP) (TRUE (GOLD ?X)) (DISTINCT ?X I)) (<= (NEXT (STEP ?Y)) (TRUE (STEP ?X)) (SUCC ?X ?Y)) (ADJACENT A B) (ADJACENT B C) (ADJACENT C D) (ADJACENT D A) (SUCC 1 2) (SUCC 2 3) (SUCC 3 4) (SUCC 4 5) (SUCC 5 6) (SUCC 6 7) (SUCC 7 8) (SUCC 8 9) (SUCC 9 10) (<= (LEGAL ROBOT MOVE)) (<= (LEGAL ROBOT GRAB) (TRUE (CELL ?X)) (TRUE (GOLD ?X))) (<= (LEGAL ROBOT DROP) (TRUE (GOLD I))) (<= (GOAL ROBOT 100) (TRUE (GOLD A))) (<= (GOAL ROBOT 0) (TRUE (GOLD ?X)) (DISTINCT ?X A)) (<= TERMINAL (TRUE (STEP 10))) (<= TERMINAL (TRUE (GOLD A)))) 5 10)
********** Game Initialization of match MATCH.3378413291 **********
Player ROBOT
Legal moves: (DOES ROBOT MOVE)
Goal values: 0
My role is ROBOT
Game is in non-terminal state.
********** End of Inıtialization of match MATCH.3378413291 **********
2:08:29 AM EET
Response:READY
POST / HTTP/1.0
2:08:44 AM EET
Command: (PLAY MATCH.3378413291 NIL)
********** next turn **********
my legal moves are: (DOES ROBOT MOVE)
end of depth 3
end of depth 4
end of depth 5
end of depth 6
end of depth 7
end of depth 8
end of depth 9
end of depth 10
end of depth 11
end of depth 12
end of depth 13
end of depth 14
end of depth 15
end of depth 16
end of depth 17
end of depth 18
end of depth 19
end of depth 20
end of depth 21
end of depth 22

end of depth 23
end of depth 24
end of depth 25
end of depth 26
end of depth 27
end of depth 28
end of depth 29
end of depth 30
end of depth 31
end of depth 32
end of depth 33
end of depth 34
end of depth 35
end of depth 36
end of depth 37
end of depth 38
end of depth 39
end of depth 40
end of depth 41
end of depth 42
end of depth 43
end of depth 44
end of depth 45
end of depth 46
end of depth 47
end of depth 48
end of depth 49
end of depth 50
end of depth 51
end of depth 52
end of depth 53
end of depth 54
end of depth 55
end of depth 56
end of depth 57
end of depth 58
end of depth 59
end of depth 60
end of depth 61
end of depth 62
end of depth 63
end of depth 64
end of depth 65
end of depth 66
end of depth 67
end of depth 68
end of depth 69
end of depth 70
My move is: MOVE
********** end of turn **********
2:08:49 AM EET
Response:MOVE
POST / HTTP/1.0
2:08:50 AM EET
Command: (PLAY MATCH.3378413291 (MOVE))
********** next turn **********
Next state: < GameState: {CELL=(TRUE (CELL B)) , GOLD=(TRUE (GOLD C)) , STEP=(TRUE (STEP 2)) } >
GGP's old state after Simulate:
< GameState: {CELL=(TRUE (CELL A)) , GOLD=(TRUE (GOLD C)) , STEP=(TRUE (STEP 1)) } >
my legal moves are: (DOES ROBOT MOVE)
end of depth 3
end of depth 4
end of depth 5
end of depth 6
end of depth 7
end of depth 8
end of depth 9
end of depth 10
end of depth 11
end of depth 12
end of depth 13
end of depth 14
end of depth 15
end of depth 16
end of depth 17
end of depth 18
end of depth 19
end of depth 20

end of depth 21
end of depth 22
end of depth 23
end of depth 24
end of depth 25
end of depth 26
end of depth 27
end of depth 28
end of depth 29
end of depth 30
end of depth 31
end of depth 32
end of depth 33
end of depth 34
end of depth 35
end of depth 36
end of depth 37
end of depth 38
end of depth 39
end of depth 40
end of depth 41
end of depth 42
end of depth 43
end of depth 44
end of depth 45
end of depth 46
end of depth 47
end of depth 48
end of depth 49
end of depth 50
end of depth 51
end of depth 52
end of depth 53
end of depth 54
end of depth 55
end of depth 56
end of depth 57
end of depth 58
end of depth 59
end of depth 60
end of depth 61
end of depth 62
end of depth 63
end of depth 64
end of depth 65
end of depth 66
end of depth 67
end of depth 68
end of depth 69
end of depth 70
end of depth 71
end of depth 72
end of depth 73
end of depth 74
My move is: MOVE
********** end of turn **********
2:08:55 AM EET
Response:MOVE
POST / HTTP/1.0
2:08:55 AM EET
Command: (PLAY MATCH.3378413291 (MOVE))
********** next turn **********
Next state: < GameState: {CELL=(TRUE (CELL C)) , GOLD=(TRUE (GOLD C)) , STEP=(TRUE (STEP 3)) } >
GGP's old state after Simulate:
< GameState: {CELL=(TRUE (CELL B)) , GOLD=(TRUE (GOLD C)) , STEP=(TRUE (STEP 2)) } >
my legal moves are: (DOES ROBOT MOVE) (DOES ROBOT GRAB)
end of depth 3
end of depth 4
end of depth 5
end of depth 6
end of depth 7
end of depth 8
end of depth 9
end of depth 10
end of depth 11
end of depth 12
end of depth 13
end of depth 14

end of depth 15
end of depth 16
end of depth 17
end of depth 18
end of depth 19
end of depth 20
end of depth 21
end of depth 22
end of depth 23
end of depth 24
end of depth 25
end of depth 26
end of depth 27
end of depth 28
end of depth 29
end of depth 30
end of depth 31
end of depth 32
end of depth 33
end of depth 34
end of depth 35
end of depth 36
end of depth 37
end of depth 38
end of depth 39
end of depth 40
end of depth 41
end of depth 42
end of depth 43
end of depth 44
end of depth 45
end of depth 46
end of depth 47
end of depth 48
end of depth 49
end of depth 50
end of depth 51
end of depth 52
end of depth 53
end of depth 54
end of depth 55
end of depth 56
end of depth 57
end of depth 58
end of depth 59
end of depth 60
end of depth 61
end of depth 62
end of depth 63
end of depth 64
end of depth 65
end of depth 66
end of depth 67
end of depth 68
end of depth 69
end of depth 70
end of depth 71
end of depth 72
end of depth 73
end of depth 74
end of depth 75
end of depth 76
My move is: GRAB
********** end of turn **********
2:09:00 AM EET
Response:GRAB
POST / HTTP/1.0
2:09:01 AM EET
Command: (PLAY MATCH.3378413291 (GRAB))
********** next turn **********
Next state: < GameState: {CELL=(TRUE (CELL C)) , GOLD=(TRUE (GOLD I)) , STEP=(TRUE (STEP 4)) } >
GGP's old state after Simulate:
< GameState: {CELL=(TRUE (CELL C)) , GOLD=(TRUE (GOLD C)) , STEP=(TRUE (STEP 3)) } >
my legal moves are: (DOES ROBOT MOVE) (DOES ROBOT DROP)
end of depth 3
end of depth 4
end of depth 5
end of depth 6

end of depth 7
end of depth 8
end of depth 9
end of depth 10
end of depth 11
end of depth 12
end of depth 13
end of depth 14
end of depth 15
end of depth 16
end of depth 17
end of depth 18
end of depth 19
end of depth 20
end of depth 21
end of depth 22
end of depth 23
end of depth 24
end of depth 25
end of depth 26
end of depth 27
end of depth 28
end of depth 29
end of depth 30
end of depth 31
end of depth 32
end of depth 33
end of depth 34
end of depth 35
end of depth 36
end of depth 37
end of depth 38
end of depth 39
end of depth 40
end of depth 41
end of depth 42
end of depth 43
end of depth 44
end of depth 45
end of depth 46
end of depth 47
end of depth 48
end of depth 49
end of depth 50
end of depth 51
end of depth 52
end of depth 53
end of depth 54
end of depth 55
end of depth 56
end of depth 57
end of depth 58
end of depth 59
end of depth 60
end of depth 61
end of depth 62
end of depth 63
end of depth 64
end of depth 65
end of depth 66
end of depth 67
end of depth 68
end of depth 69
end of depth 70
end of depth 71
end of depth 72
end of depth 73
end of depth 74
end of depth 75
end of depth 76
end of depth 77
end of depth 78
end of depth 79
end of depth 80
end of depth 81
end of depth 82
end of depth 83
end of depth 84

end of depth 85
end of depth 86
end of depth 87
end of depth 88
My move is: MOVE
********** end of turn **********
2:09:06 AM EET
Response:MOVE
POST / HTTP/1.0
2:09:07 AM EET
Command: (PLAY MATCH.3378413291 (MOVE))
********** next turn **********
Next state: < GameState: {CELL=(TRUE (CELL D)) , GOLD=(TRUE (GOLD I)) , STEP=(TRUE (STEP 5)) } >
GGP's old state after Simulate:
< GameState: {CELL=(TRUE (CELL C)) , GOLD=(TRUE (GOLD I)) , STEP=(TRUE (STEP 4)) } >
my legal moves are: (DOES ROBOT MOVE) (DOES ROBOT DROP)
end of depth 3
end of depth 4
end of depth 5
end of depth 6
end of depth 7
end of depth 8
end of depth 9
end of depth 10
end of depth 11
end of depth 12
end of depth 13
end of depth 14
end of depth 15
end of depth 16
end of depth 17
end of depth 18
end of depth 19
end of depth 20
end of depth 21
end of depth 22
end of depth 23
end of depth 24
end of depth 25
end of depth 26
end of depth 27
end of depth 28
end of depth 29
end of depth 30
end of depth 31
end of depth 32
end of depth 33
end of depth 34
end of depth 35
end of depth 36
end of depth 37
end of depth 38
end of depth 39
end of depth 40
end of depth 41
end of depth 42
end of depth 43
end of depth 44
end of depth 45
end of depth 46
end of depth 47
end of depth 48
end of depth 49
end of depth 50
end of depth 51
end of depth 52
end of depth 53
end of depth 54
end of depth 55
end of depth 56
end of depth 57
end of depth 58
end of depth 59
end of depth 60
end of depth 61
end of depth 62
end of depth 63
end of depth 64

end of depth 65
end of depth 66
end of depth 67
end of depth 68
end of depth 69
end of depth 70
end of depth 71
end of depth 72
end of depth 73
end of depth 74
end of depth 75
end of depth 76
end of depth 77
end of depth 78
end of depth 79
end of depth 80
end of depth 81
end of depth 82
end of depth 83
end of depth 84
end of depth 85
end of depth 86
end of depth 87
end of depth 88
end of depth 89
end of depth 90
end of depth 91
end of depth 92
end of depth 93
end of depth 94
end of depth 95
end of depth 96
end of depth 97
end of depth 98
end of depth 99
end of depth 100
end of depth 101
end of depth 102
end of depth 103
end of depth 104
end of depth 105
end of depth 106
end of depth 107
end of depth 108
end of depth 109
end of depth 110
end of depth 111
end of depth 112
end of depth 113
end of depth 114
end of depth 115
end of depth 116
end of depth 117
end of depth 118
end of depth 119
end of depth 120
end of depth 121
end of depth 122
end of depth 123
end of depth 124
end of depth 125
end of depth 126
end of depth 127
end of depth 128
end of depth 129
end of depth 130
end of depth 131
end of depth 132
end of depth 133
end of depth 134
end of depth 135
end of depth 136
end of depth 137
end of depth 138
end of depth 139
end of depth 140
end of depth 141
end of depth 142

end of depth 143
end of depth 144
end of depth 145
end of depth 146
end of depth 147
end of depth 148
end of depth 149
end of depth 150
end of depth 151
end of depth 152
end of depth 153
end of depth 154
end of depth 155
end of depth 156
end of depth 157
end of depth 158
end of depth 159
end of depth 160
end of depth 161
end of depth 162
end of depth 163
end of depth 164
end of depth 165
end of depth 166
end of depth 167
My move is: MOVE
********** end of turn **********
2:09:12 AM EET
Response:MOVE
POST / HTTP/1.0
2:09:12 AM EET
Command: (PLAY MATCH.3378413291 (MOVE))
********** next turn **********
Next state: < GameState: {CELL=(TRUE (CELL A)) , GOLD=(TRUE (GOLD I)) , STEP=(TRUE (STEP 6)) } >
GGP's old state after Simulate:
< GameState: {CELL=(TRUE (CELL D)) , GOLD=(TRUE (GOLD I)) , STEP=(TRUE (STEP 5)) } >
my legal moves are: (DOES ROBOT MOVE) (DOES ROBOT DROP)
end of depth 3
end of depth 4
end of depth 5
end of depth 6
end of depth 7
end of depth 8
end of depth 9
end of depth 10
end of depth 11
end of depth 12
end of depth 13
end of depth 14
end of depth 15
end of depth 16
end of depth 17
end of depth 18
end of depth 19
end of depth 20
end of depth 21
end of depth 22
end of depth 23
end of depth 24
end of depth 25
end of depth 26
end of depth 27
end of depth 28
end of depth 29
end of depth 30
end of depth 31
end of depth 32
end of depth 33
end of depth 34
end of depth 35
end of depth 36
end of depth 37
end of depth 38
end of depth 39
end of depth 40
end of depth 41
end of depth 42
end of depth 43

end of depth 44
end of depth 45
end of depth 46
end of depth 47
end of depth 48
end of depth 49
end of depth 50
end of depth 51
end of depth 52
end of depth 53
end of depth 54
end of depth 55
end of depth 56
end of depth 57
end of depth 58
end of depth 59
end of depth 60
end of depth 61
end of depth 62
end of depth 63
end of depth 64
end of depth 65
end of depth 66
end of depth 67
end of depth 68
end of depth 69
end of depth 70
end of depth 71
end of depth 72
end of depth 73
end of depth 74
end of depth 75
end of depth 76
end of depth 77
end of depth 78
end of depth 79
end of depth 80
end of depth 81
end of depth 82
end of depth 83
end of depth 84
end of depth 85
end of depth 86
end of depth 87
end of depth 88
end of depth 89
end of depth 90
end of depth 91
end of depth 92
end of depth 93
end of depth 94
end of depth 95
end of depth 96
end of depth 97
end of depth 98
end of depth 99
end of depth 100
end of depth 101
end of depth 102
end of depth 103
end of depth 104
end of depth 105
end of depth 106
end of depth 107
end of depth 108
end of depth 109
end of depth 110
end of depth 111
end of depth 112
end of depth 113
end of depth 114
end of depth 115
end of depth 116
end of depth 117
end of depth 118
end of depth 119
end of depth 120
end of depth 121

end of depth 122
end of depth 123
end of depth 124
end of depth 125
end of depth 126
end of depth 127
end of depth 128
end of depth 129
end of depth 130
end of depth 131
end of depth 132
end of depth 133
end of depth 134
end of depth 135
end of depth 136
end of depth 137
end of depth 138
end of depth 139
end of depth 140
end of depth 141
end of depth 142
end of depth 143
end of depth 144
end of depth 145
end of depth 146
end of depth 147
end of depth 148
end of depth 149
end of depth 150
end of depth 151
end of depth 152
end of depth 153
end of depth 154
end of depth 155
end of depth 156
end of depth 157
end of depth 158
end of depth 159
end of depth 160
end of depth 161
end of depth 162
end of depth 163
end of depth 164
end of depth 165
end of depth 166
end of depth 167
end of depth 168
end of depth 169
end of depth 170
end of depth 171
end of depth 172
end of depth 173
end of depth 174
end of depth 175
end of depth 176
end of depth 177
end of depth 178
end of depth 179
end of depth 180
end of depth 181
end of depth 182
end of depth 183
end of depth 184
end of depth 185
end of depth 186
end of depth 187
end of depth 188
end of depth 189
end of depth 190
end of depth 191
end of depth 192
end of depth 193
end of depth 194
end of depth 195
end of depth 196
end of depth 197
end of depth 198
end of depth 199

end of depth 200
end of depth 201
end of depth 202
end of depth 203
end of depth 204
end of depth 205
end of depth 206
end of depth 207
end of depth 208
end of depth 209
end of depth 210
end of depth 211
end of depth 212
end of depth 213
end of depth 214
end of depth 215
end of depth 216
end of depth 217
end of depth 218
end of depth 219
end of depth 220
end of depth 221
end of depth 222
end of depth 223
end of depth 224
end of depth 225
end of depth 226
end of depth 227
end of depth 228
end of depth 229
end of depth 230
end of depth 231
end of depth 232
end of depth 233
end of depth 234
end of depth 235
end of depth 236
end of depth 237
end of depth 238
end of depth 239
end of depth 240
end of depth 241
end of depth 242
end of depth 243
end of depth 244
end of depth 245
end of depth 246
end of depth 247
end of depth 248
end of depth 249
end of depth 250
end of depth 251
end of depth 252
end of depth 253
end of depth 254
end of depth 255
end of depth 256
end of depth 257
end of depth 258
end of depth 259
end of depth 260
end of depth 261
end of depth 262
end of depth 263
end of depth 264
end of depth 265
end of depth 266
end of depth 267
end of depth 268
end of depth 269
end of depth 270
end of depth 271
end of depth 272
end of depth 273
end of depth 274
end of depth 275
end of depth 276
end of depth 277

end of depth 278
end of depth 279
end of depth 280
end of depth 281
end of depth 282
end of depth 283
end of depth 284
end of depth 285
end of depth 286
end of depth 287
end of depth 288
end of depth 289
end of depth 290
end of depth 291
end of depth 292
end of depth 293
end of depth 294
end of depth 295
end of depth 296
end of depth 297
end of depth 298
end of depth 299
end of depth 300
end of depth 301
end of depth 302
end of depth 303
end of depth 304
end of depth 305
end of depth 306
end of depth 307
end of depth 308
end of depth 309
end of depth 310
end of depth 311
end of depth 312
end of depth 313
end of depth 314
end of depth 315
end of depth 316
end of depth 317
end of depth 318
end of depth 319
end of depth 320
end of depth 321
end of depth 322
end of depth 323
end of depth 324
end of depth 325
end of depth 326
end of depth 327
end of depth 328
end of depth 329
end of depth 330
end of depth 331
end of depth 332
end of depth 333
end of depth 334
end of depth 335
end of depth 336
end of depth 337
end of depth 338
end of depth 339
end of depth 340
end of depth 341
end of depth 342
end of depth 343
end of depth 344
end of depth 345
end of depth 346
end of depth 347
end of depth 348
end of depth 349
end of depth 350
end of depth 351
end of depth 352
end of depth 353
end of depth 354
end of depth 355

end of depth 356
My move is: DROP
********** end of turn **********
2:09:17 AM EET
Response:DROP
POST / HTTP/1.0
2:09:18 AM EET
Command: (STOP MATCH.3378413291 (DROP))
********** end of game, exiting... **********