AIGS Documentation

Last revision: 8 October 2008

Table of Contents

AIGS

Overview

AIGS (Artificial Intelligence Game Server) is a simple server that facilitates communication between AI clients playing a variety of games.

The AIGS is written in Java 1.6 and should run correctly without any additional packages. To run the server, use Java to run the JAR file containing the server.

Protocol to start/join a game

To open communications with an AIGS server, send a message of one of the following forms to the server (a comma separated list of 3 elements):
AIGS,<Username>,<GameType>

AIGS is the string AIGS that identifies to the server that you are in fact trying to communicate with it.
<Username> is a unique ID identifying each player.
<GameType> is a unique string identifying each specific game format. For example, the GameType of Hex is the string "Hex".

In each case, if you have now joined a game, the server will respond with the following message:
AIGS,JoinSuccess,<GameID>,<PlayerNumber>

<GameID> is a unique identifier for the game you just joined (the timestamp of when the server created the game).
<PlayerNumber> is 1 for first player, 2 for second player.

If it was not possible to join a game, the server will respond with the following message:
AIGS,JoinFail,<Reason>

Once you have received a JoinSuccess message, use the below protocol to play the game.

Protocol to play a game

Once you have sent the first message to start/join a game, there is only one message for the rest of the game:
AIGS,<Move>

<Move> is a move string unique to the game that you are playing. See below for the example move format for Hex.

If the move is successful, the player that just played will receive the following message:
AIGS,MoveSuccess,<Move>

And the other player (whose turn it is now), will receive the following message (with your move):
AIGS,YourTurn,<Move>

If a move is not successful, the player that tried to play will receive the following message:
AIGS,MoveFail,<Reason>

If either player is taking a relatively long time to play, the game make remind them that it is their turn by sending the following message: (The interval of reminders is not guaranteed, nor are the reminders themselves.)
AIGS,YourTurn

There are several universal moves that will be supported by all game types:
Resign - Quit the current game, the other player is given credit for winning.
Update - Request an update on the current game state. The server will return the following message:

AIGS,GameState,<State>,<PlayerNumber>

Player - Request your player number. The server will return the following message:

AIGS,Player,<PlayerNumber>

Opponent - Request your opponent's username. The server will return the following message:

AIGS,Opponent,<Username>

<State> is an encoding of the game as a string. See below for the example game state for Hex.
<PlayerNumber> is 1 for first player, 2 for second player.

Finally, use the following message to taunt your opponent:
AIGS,Taunt,<Message>

Your opponent will receive the following:
AIGS,Taunt,<Message>

Hex

Board layout for Hex

The board is constructed of a regular layout of hexagonal pieces in a NxN board (where N is by default 11).

Rows and columns are numbered from 0 to N-1 with the top left being (0, 0) and the bottom right being (N-1, N-1). See below the numbering (row, col) for a 4x4 board:

```
0,0   0,1   0,2   0,3
   1,0   1,1   1,2   1,3
      2,0   2,1   2,2   2,3
         3,0   3,1   3,2   3,3
```

On each turn, the players place one of their pieces on a currently empty board. The only exception is on the second player's first turn. To make the game even, the second player may replace the first player's piece with one of their own.

Win conditions for Hex

To win a game of Hex, Player 1 must connect the top and bottom of the board with a line of adjacent pieces. Similiarly, Player 2 must connect the left and right edges of the board to win. The four corners are considered edges for both players.

As proven by John Nash (of A Beautiful Mind fame), a game of Hex cannot end in a tie. The only way to prevent the other player from winning is to win yourself.

Move format for Hex

The move format for hex is simple:
<Row>x<Col>

<Row> and <Col> are both integers in the range 0 to N-1 (inclusive). This specifies the board location where the player would like to place a piece. If the hex is empty, the piece will be placed. If not (with the exception of the second player's first turn, see above), the player cannot play there and will receive and error message (see above).

Game state format for Hex

The string format for a Hex game consists of reading across each row of the game board from left to right, returning a 1 for Player 1's pieces, a 2 for Player 2's pieces, and a dash (-) for empty hexes.

For example, given the following game state (on a 4x4 board):

```
-  -  2  -
 -  -  1  -
   -  2  1  -
      -  -  -  -
```

The following <State> will be returned:

AIGS,GameState,--2---1--21-----