

# Server/Client Architecture

---

To facilitate the development of AI agents for playing Angry Birds we have separated the game interaction functionality from the agent logic and reasoning functionality through a client/server architecture. The server communicates with the Angry Birds game and maintains a history of shots played by each agent. It exposes a simple communication protocol for the client to execute shots and perform auxiliary functions such as reload a level. This document describes that protocol. The same protocol will be used for the Angry Birds AI Competition (IJCAI 2013)

## Protocols

---

A typical client message contains a message header that shows the ID of the message (MID) and a message body (the body can be empty). All the messages (bytes) are encoded in network order (big endian) form.

### Configuration Messages

A configuration message is sent by the client when it first connects with the server. The message contains the team ID (a four- bytes Integer). On receiving the message, the server will check the database to verify the ID. The client will be rejected if the ID is invalid.

In development version, the server will accept all integers.

### In-Game Action Messages:

In the game, an agent is allowed to make shots and zoom out.

#### Shot in Cartesian coordinates (cshot)

The body of the message contains 6 parameters with each of the parameter consumes 4 bytes. The parameters are:

1. focus\_x : the x coordinate of the focus point
2. focus\_y: the y coordinate of the focus point
3. dx: the x coordinate of the release point minus focus\_x
4. dy: the y coordinate of the release point minus focus\_y
5. t1: the release time
6. t2: the gap between the release time and the tap time.

If t1 is set to 0, the server will execute the shot immediately.

### The Message Format

MID	focus_x	focus_y	dx	dy	t1	t2
-----	---------	---------	----	----	----	----

Return 1 or 0

1: the shot has been made

0: the shot has been rejected

### Shot in Polar coordinates (pshot)

This message is almost the same as the chost message except it uses  $r$  and  $\theta$  instead of  $d1$  and  $d2$ .

$r$ : the radial coordinate

$\theta$ : the angular coordinate by degree from -90 to 90

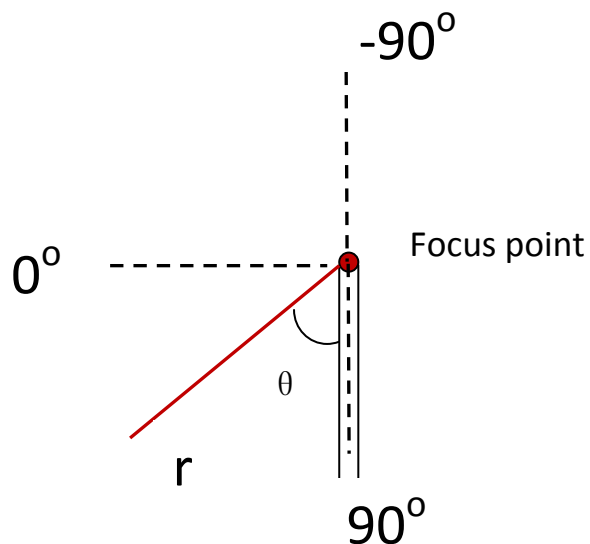
### The Message Format

MID	focus_x	focus_y	$r$	$\theta$	t1	t2
-----	---------	---------	-----	----------	----	----

Return 1 or 0

1: the shot has been made

0: the shot has been rejected



### Shooting sequence

Shooting sequence contains a set of shots that will be made sequentially.

The Message Format (the shot below refers to either a cshot or pshot message):

MID	num of shots	shot	....	shot
-----	--------------	------	------	------

Return:

An array with each slot indicates a good/bad shot. The bad shots are those shots that have been rejected.

For example, the server received 5 shots, and the third one was rejected due to some reason, then the server will return

[1][1][0][1][1]

## FullyZoom

The server will fully zoom out on receiving this message.

The Message Format

MID
-----

Return 1 or 0

1: The server has fully zoomed out in the current game.

0: The server cannot zoom out

## Query Messages

In the competition, an agent is allowed to query the current screen, game state , level and global scores.

## DoScreenShot

The Message Format

MID
-----

Return:

Width	Height	Image Bytes
-------	--------	-------------

The agent will be returned with a bytes array. The first and second four bytes tells the image width and height respectively. The remaining bytes are raw bytes (as RGB triples) of the image.

## Get the state

The Message Format

MID
-----

Return:

Ordinal of the state
----------------------

The server will return One byte indicates the ordinal of the state

[0]: UNKNOWN [1] : MAIN\_MENU [2]: EPISODE\_MENU [3]: LEVEL\_SELECTION [4]: LOADING  
[5]: PLAYING [6]: WON [7]: LOST

## Get the current level

The Message Format

MID
-----

Return:

Level
-------

## Get Best Scores

The Message Format

MID
-----

Return:

Level 1 score	Level 2 score	.....	Level 21 Score
---------------	---------------	-------	----------------

The server will return a fixed length bytes array with every four slots indicates a best score of the corresponding level

## Level Selection Messages

An agent is allowed to select a level by using the following two messages

### Load a Level

The Message Format

MID	Level
-----	-------

The level is indicated by one byte. The value can be set from 0 to the maximum level number (will be released before the competition). To load the current level, please set the value to 0.

Return:

Return 1 or 0

1: the level has been loaded

0: The server cannot load the level

## Restart a Level

The Message Format

MID
-----

Return 1 or 0

1: the level has been restarted

0: The server cannot restart the level

## Summary of the Protocols

MID	Request	Format (byte[])	Return	Format (byte[])
1-10	Configuration Messages			
1	Configure(Team ID)	[1][ID] ID: 4 bytes	OK/ERR	[1]/[0]
11-30	Query Messages			
11	Do screenshot	[11]	Width, height, plus image bytes	[width][height][image bytes] Width, height: 4 bytes
12	Get the state	[12]	One byte indicates the ordinal of the state	[0]: UNKNOWN [1]: MAIN_MENU [2]: EPISODE_MENU [3]: LEVEL_SELECTION [4]: LOADING [5]: PLAYING [6]: WON [7]: LOST
13	Get Best Scores	[13]	A fixed length bytes array with every four slots indicates a best score	[score_level1]....[score_level21]
14	Get the current level	[14]	One byte indicates the current level	[1-21]
31-50	In-Game Action Messages			
31	Shoot using Cartesian coordinates	[31][fx][fy][dx][dy][t1][t2] Each parameter consumes 4 bytes	OK/ERR	[1]/[0]
32	Shoot using Polar coordinates	[32][fx][fy][theta][r][t1][t2] Each parameter consumes 4 bytes	OK/ERR	[1]/[0]
33	Sequence of Shot	[33][shots length][MID][Params][MID][Params] Maximum sequence length: 16 shots	An array with each slot indicates good/bad shot. The bad shots are those shots that are rejected by the server	For example, the server received 5 shots, and the third one was not executed due to some reason, then the server will return [1][1][0][1][1]

34	Fully Zoom	[34]	OK/ERR	[1]/[0]
51-60	Level Selection Messages			
51	Load a Level	[51][Level] Level : 1 byte	OK/ERR	[1]/[0]
52	Restart a level	[52]	OK/ERR	[1]/[0]