

---

# **pyagar Documentation**

*Release 0.0.8*

**Roberto Abdelkader Martínez Pérez**

July 22, 2015



<b>1</b>	<b>User documentation</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Command Line Interface . . . . .	3
1.3	Controls . . . . .	4
<b>2</b>	<b>Developer documentation</b>	<b>5</b>
2.1	pyagar.client . . . . .	5
2.2	pyagar.cmdline . . . . .	6
2.3	pyagar.controller . . . . .	6
2.4	pyagar.messages . . . . .	7
2.5	pyagar.utils . . . . .	10
2.6	pyagar.visual . . . . .	10
<b>3</b>	<b>0.0.8</b>	<b>13</b>
<b>4</b>	<b>0.0.3</b>	<b>15</b>
<b>5</b>	<b>Other Implementations</b>	<b>17</b>
<b>6</b>	<b>Indices and tables</b>	<b>19</b>
	<b>Python Module Index</b>	<b>21</b>



pyagar is a python client for the popular online game at <http://agar.io>.

**Warning:** This software is not the official client and is not associated in any manner with the original website.

This package allows you to play the game, see the gameplay as an spectator, record and replay games and also develop your own bot.

Please, [checkout the documentation](#) for further instructions.

Enjoy!



---

## User documentation

---

### 1.1 Installation

#### 1.1.1 Dependencies

You'll need some software in order to run pyagar.

- Python  $\geq$  3.4
- libsd12
- libsd12-gfx
- libsd12-ttf

#### 1.1.2 Linux

##### Ubuntu 15.04

```
$ sudo apt-get install python3 python3-pip libsd12-2.0-0 libsd12-gfx-1.0-0 libsd12-ttf-2.0-0
$ sudo pip3 install pyagar
```

#### 1.1.3 Windows

Grab the last Windows installer from <https://github.com/nilp0inter/pyagar/releases/latest> and run it.

### 1.2 Command Line Interface

```
usage: pyagar [-h] [--disable-hw] [-n NICK] [-d] [-r REGION] [-s SAVE]
              [--version] [--create-party | --join-party JOIN_PARTY]
              {list-regions,play,spectate,bot,replay} ...
```

#### Options:

- disable-hw=False** disable hardware acceleration
- n=pyagar, --nick=pyagar** player cell's nickname
- d, --debug** enable debug mode; use multiple times to increase the debug level

- r=EU-London, --region=EU-London** the region you want to connect to
- s, --save** save the gameplay in a file; you can replay it later using the “replay” command
- version** show program’s version number and exit
- create-party=False** create a new party and print the shared token
- join-party** join an already created party

**Sub-commands:**

**list-regions** print a table with the list of available regions

usage: pyagar list-regions [-h]

**play** start a new game

usage: pyagar play [-h]

**spectate** connect to the server in spectator mode

usage: pyagar spectate [-h]

**bot** like “play” mode but the cell is controlled by a “Controller” class

usage: pyagar bot [-h] (--list-types | --type TYPE | --from-file FROM\_FILE)

**Options:**

- list-types=False** print a table with the available “Controllers”
- type** type of controller to use
- from-file** use a controller from a python file

**replay** play back a recorded gameplay

usage: pyagar replay [-h] gameplay\_file

**Positional arguments:**

- gameplay\_file** full path to the record file

## 1.3 Controls

Action	Peripheral	Detail
<b>Eject</b>	Keyboard	W
<b>Exit</b>	Keyboard	ESC
<b>Fullscreen</b>	Keyboard	F
<b>Move</b>	Mouse	
<b>Split</b>	Keyboard	Space
<b>Start</b>	Mouse	Left
<b>Zoom</b>	Mouse	Wheel



---

## Developer documentation

---

### 2.1 pyagar.client

This module contains the Client class.

**class** `pyagar.client.Client` (*nick, region='EU-London', party=False*)

The client.

Manages the connection, receives the data from the server and sends back any command requested by the player.

**connect** ()

Connects to the server.

**create\_party** ()

Create a new party.

**eject** ()

Sends the `mass eject` command.

**classmethod get\_regions** ()

Request the list of regions.

**get\_server** ()

Requests a new server and token.

**move** (*x, y*)

Sends the `movement` command.

**read** ()

Read, decode and queue data packets from the server.

**spawn** ()

Sends the `spawn` command.

**spectate** ()

Initiates the spectator mode.

**split** ()

Sends the `split cell` command.

`pyagar.client.build_request` (*set\_header*)

Build a handshake request to send to the server.

Return the *key* which must be passed to `check_response` () .

## 2.2 pyagar.cmdline

Provides the cmdline facility.

`pyagar.cmdline.pyagar` (*argv=None*)  
pyagar cli interface.

`pyagar.cmdline.pyagar_parser` ()  
Generates the argument parser.

## 2.3 pyagar.controller

Some very simple bots.

**class** `pyagar.control.Center` (*client*)  
Go to the center.

**class** `pyagar.control.Closer` (*client*)  
Go to the closer “non-virus” cell, no matter the type.

**class** `pyagar.control.Controller` (*client*)  
All bots should inherit from this class.

**do\_move** ()  
Make a movement.

**edible**  
You can eat cells 10% smaller than you.

**get\_movement** ()  
The method that subclasses must implement.

**get\_name** ()  
Returns the name of this bot.

**opponents**  
Return list of other cells.

**player**  
Returns the player main cell. None if not exists.

**predators**  
Cells that can eat me.

**run** ()  
The main loop of the bot.

**viruses**  
Returns a list of visible viruses.

**class** `pyagar.control.EatWhenNoPredators` (*client*)  
Only eats when all visible cells are smaller than itself.

**class** `pyagar.control.Escape` (*client*)  
Escape from bigger opponents.

**compound\_escape\_vector** (*vectors, cells*)  
Returns the sum of all vectors.

**static escape\_vector** (*player, cell*)  
Movement to escape from a cell.

**class** `pyagar.control.Greedy` (*client*)  
Only wants to eat.

**class** `pyagar.control.Movement` (*x, y*)

**x**  
Alias for field number 0

**y**  
Alias for field number 1

`pyagar.log`

Contains the logger.

## 2.4 `pyagar.messages`

Protocol implementation.

**class** `pyagar.messages.BaseMSG` (*buf, offset=0*)  
All messages inherits from this class.

Contains utility methods for unpack the data.

**get** (*ctype*)  
Unpack the given *ctype* and update the offset.

**getFloat32** ()  
Unpack an `FLOAT32`.

**getFloat64** ()  
Unpack an `FLOAT64`.

**getInt16** ()  
Unpack an `INT16`.

**getInt32** ()  
Unpack an `INT32`.

**getInt64** ()  
Unpack an `INT64`.

**getInt8** ()  
Unpack an `INT8`.

**getUint16** ()  
Unpack an `UINT16`.

**getUint32** ()  
Unpack an `UINT32`.

**getUint64** ()  
Unpack an `UINT64`.

**getUint8** ()  
Unpack an `UINT8`.

**parse** ()  
This method must be implemented in the message subclass.

**string** (*ctype='H'*)  
Unpack a string.

**class** pyagar.messages.**Camera** (*x, y, zoom*)

**x**  
Alias for field number 0

**y**  
Alias for field number 1

**zoom**  
Alias for field number 2

**class** pyagar.messages.**CameraPosition** (*buf, offset=0*)  
Change in the camera position and/or the zoom.

Only received in Spectate mode.

**parse** ()  
Unpacks the data.

**class** pyagar.messages.**Cell** (*id, x, y, size, color, is\_virus, name*)

**color**  
Alias for field number 4

**id**  
Alias for field number 0

**is\_virus**  
Alias for field number 5

**name**  
Alias for field number 6

**size**  
Alias for field number 3

**x**  
Alias for field number 1

**y**  
Alias for field number 2

**class** pyagar.messages.**Dissapear** (*id*)

**id**  
Alias for field number 0

**class** pyagar.messages.**Eat** (*eater, eatee*)

**eatee**  
Alias for field number 1

**eater**  
Alias for field number 0

**class** pyagar.messages.**Leaderboard** (*buf, offset=0*)  
The Leaderboard.

The top bigger cells in descending order.

This message is only received in FFA and Experimental mode.

**parse ()**  
Unpacks the data.

**class** `pyagar.messages.MSG (buf, offset=0)`  
All messages.

This class identify the specific message type and calls the proper parser.

**parse ()**  
Unpacks the message identifier and instantiate the parser.

**class** `pyagar.messages.MSGType`  
This enum contains the identifier of each message along with the name of the class which parses it.

**cls**  
Returns the parser class of this enum.

**class** `pyagar.messages.Player (id, name)`

**id**  
Alias for field number 0

**name**  
Alias for field number 1

**class** `pyagar.messages.PlayerCell (buf, offset=0)`  
ID of the player.

**parse ()**  
Unpacks the data.

**class** `pyagar.messages.PlayerID (id)`

**id**  
Alias for field number 0

**class** `pyagar.messages.ResetSomething (buf, offset=0)`  
This message is present in the code but never seen.

**parse ()**  
Unpacks the data.

**class** `pyagar.messages.Screen (x1, y1, x2, y2)`

**x1**  
Alias for field number 0

**x2**  
Alias for field number 2

**y1**  
Alias for field number 1

**y2**  
Alias for field number 3

**class** `pyagar.messages.ScreenAndCamera (buf, offset=0)`  
Screen and Camera position, all in one.

This message is the first message in the stream.

```
parse ()  
    Unpacks the data.
```

```
class pyagar.messages.SetQARA (buf, offset=0)  
    This message is present in the code but never seen.
```

```
parse ()  
    Unpacks the data.
```

```
class pyagar.messages.Status (buf, offset=0)  
    The status of the stage.
```

Who eats who, what is visible, what disappears...

```
parse ()  
    Unpacks the data.
```

```
class pyagar.messages.TeamsScore (buf, offset=0)  
    The TeamScore.
```

The percent of mass of each team.

This message is only received in the Team mode.

```
parse ()  
    Unpacks the data.
```

## 2.5 pyagar.utils

Non categorized stuff.

```
class pyagar.utils.GameReplay (filename)  
    Replay the messages saved with GameplaySaver.
```

```
class pyagar.utils.GameplaySaver (filename)  
    Store the gameplay messages in a file with a timestamp.
```

```
class pyagar.utils.Output  
    Prints every message received.
```

```
run ()  
    Logs all everything.
```

```
pyagar.utils.hub (src, *dsts)  
    Broadcasts msgs from src.messages to all dsts.messages.
```

```
pyagar.utils.print_regions (regions)  
    Prints a pretty table with the region data.
```

## 2.6 pyagar.visual

Provides the default visualizer.

```
class pyagar.visual.Visualizer (client, view_only=False, hardware=True)  
    SDL based visualizer.
```

**refresh ()**

Draw the current status of the game in window.

The overall process is:

- 1.The server send information about the board size and status.
  - 1.1. We keep the information about the board in gamescreen.
- 2.We draw the game in the texture stage. This texture can be smaller than gamescreen.
- 3.The rectangle camera (in game coordinates) is copied from stage to window.

**static remap (o\_val, o\_min, o\_max, n\_min, n\_max)**

Map a value from one range to another.

**stage = None**

The texture we draw to.

**tr\_game2stage\_coords (x, y)**

Translate from game cords to stage coordinates.

**tr\_game2stage\_size (size)**

Translate a size (in pixels) from game to stage.

**tr\_win2game\_coords (x, y)**

Translate from window coords to game coordinates.

**window = None**

The window where we show the game.

**pyagar.visual.asrt (code)**

If there is an error on a SDL call raise an exception with the error description.





- Party mode implementation (creation and joining).
- Better mouse control.
- Game recording and replaying.
- Documentation.
- Leaderboard.
- Windows installer.



- Documenting pyagar module.



---

## Other Implementations

---

- <https://github.com/Gjum/pyAgar.io>
- <https://github.com/Raeon/pygar>



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`





**p**

pyagar.client, 5  
pyagar.cmdline, 5  
pyagar.control, 6  
pyagar.log, 7  
pyagar.messages, 7  
pyagar.utils, 10  
pyagar.visual, 10



**A**

asrt() (in module pyagar.visual), 11

**B**

BaseMSG (class in pyagar.messages), 7

build\_request() (in module pyagar.client), 5

**C**

Camera (class in pyagar.messages), 8

CameraPosition (class in pyagar.messages), 8

Cell (class in pyagar.messages), 8

Center (class in pyagar.control), 6

Client (class in pyagar.client), 5

Closer (class in pyagar.control), 6

cls (pyagar.messages.MSGType attribute), 9

color (pyagar.messages.Cell attribute), 8

compound\_escape\_vector() (pyagar.control.Escape method), 6

connect() (pyagar.client.Client method), 5

Controller (class in pyagar.control), 6

create\_party() (pyagar.client.Client method), 5

**D**

Dissapear (class in pyagar.messages), 8

do\_move() (pyagar.control.Controller method), 6

**E**

Eat (class in pyagar.messages), 8

eatee (pyagar.messages.Eat attribute), 8

eater (pyagar.messages.Eat attribute), 8

EatWhenNoPredators (class in pyagar.control), 6

edible (pyagar.control.Controller attribute), 6

eject() (pyagar.client.Client method), 5

Escape (class in pyagar.control), 6

escape\_vector() (pyagar.control.Escape static method), 6

**G**

GameplaySaver (class in pyagar.utils), 10

GameReplay (class in pyagar.utils), 10

get() (pyagar.messages.BaseMSG method), 7

get\_movement() (pyagar.control.Controller method), 6

get\_name() (pyagar.control.Controller method), 6

get\_regions() (pyagar.client.Client class method), 5

get\_server() (pyagar.client.Client method), 5

getFloat32() (pyagar.messages.BaseMSG method), 7

getFloat64() (pyagar.messages.BaseMSG method), 7

getInt16() (pyagar.messages.BaseMSG method), 7

getInt32() (pyagar.messages.BaseMSG method), 7

getInt64() (pyagar.messages.BaseMSG method), 7

getInt8() (pyagar.messages.BaseMSG method), 7

getUInt16() (pyagar.messages.BaseMSG method), 7

getUInt32() (pyagar.messages.BaseMSG method), 7

getUInt64() (pyagar.messages.BaseMSG method), 7

getUInt8() (pyagar.messages.BaseMSG method), 7

Greedy (class in pyagar.control), 7

**H**

hub() (in module pyagar.utils), 10

**I**

id (pyagar.messages.Cell attribute), 8

id (pyagar.messages.Dissapear attribute), 8

id (pyagar.messages.Player attribute), 9

id (pyagar.messages.PlayerID attribute), 9

is\_virus (pyagar.messages.Cell attribute), 8

**L**

Leaderboard (class in pyagar.messages), 8

**M**

move() (pyagar.client.Client method), 5

Movement (class in pyagar.control), 7

MSG (class in pyagar.messages), 9

MSGType (class in pyagar.messages), 9

**N**

name (pyagar.messages.Cell attribute), 8

name (pyagar.messages.Player attribute), 9

**O**

opponents (pyagar.control.Controller attribute), 6

Output (class in pyagar.utils), 10

## P

parse() (pyagar.messages.BaseMSG method), 7  
parse() (pyagar.messages.CameraPosition method), 8  
parse() (pyagar.messages.Leaderboard method), 9  
parse() (pyagar.messages.MSG method), 9  
parse() (pyagar.messages.PlayerCell method), 9  
parse() (pyagar.messages.ResetSomething method), 9  
parse() (pyagar.messages.ScreenAndCamera method), 10  
parse() (pyagar.messages.SetQARA method), 10  
parse() (pyagar.messages.Status method), 10  
parse() (pyagar.messages.TeamsScore method), 10  
Player (class in pyagar.messages), 9  
player (pyagar.control.Controller attribute), 6  
PlayerCell (class in pyagar.messages), 9  
PlayerID (class in pyagar.messages), 9  
predators (pyagar.control.Controller attribute), 6  
print\_regions() (in module pyagar.utils), 10  
pyagar() (in module pyagar.cmdline), 6  
pyagar.client (module), 5  
pyagar.cmdline (module), 5  
pyagar.control (module), 6  
pyagar.log (module), 7  
pyagar.messages (module), 7  
pyagar.utils (module), 10  
pyagar.visual (module), 10  
pyagar\_parser() (in module pyagar.cmdline), 6

## R

read() (pyagar.client.Client method), 5  
refresh() (pyagar.visual.Visualizer method), 10  
remap() (pyagar.visual.Visualizer static method), 11  
ResetSomething (class in pyagar.messages), 9  
run() (pyagar.control.Controller method), 6  
run() (pyagar.utils.Output method), 10

## S

Screen (class in pyagar.messages), 9  
ScreenAndCamera (class in pyagar.messages), 9  
SetQARA (class in pyagar.messages), 10  
size (pyagar.messages.Cell attribute), 8  
spawn() (pyagar.client.Client method), 5  
spectate() (pyagar.client.Client method), 5  
split() (pyagar.client.Client method), 5  
stage (pyagar.visual.Visualizer attribute), 11  
Status (class in pyagar.messages), 10  
string() (pyagar.messages.BaseMSG method), 7

## T

TeamsScore (class in pyagar.messages), 10  
tr\_game2stage\_coords() (pyagar.visual.Visualizer method), 11

tr\_game2stage\_size() (pyagar.visual.Visualizer method), 11  
tr\_win2game\_coords() (pyagar.visual.Visualizer method), 11

## V

viruses (pyagar.control.Controller attribute), 6  
Visualizer (class in pyagar.visual), 10

## W

window (pyagar.visual.Visualizer attribute), 11

## X

x (pyagar.control.Movement attribute), 7  
x (pyagar.messages.Camera attribute), 8  
x (pyagar.messages.Cell attribute), 8  
x1 (pyagar.messages.Screen attribute), 9  
x2 (pyagar.messages.Screen attribute), 9

## Y

y (pyagar.control.Movement attribute), 7  
y (pyagar.messages.Camera attribute), 8  
y (pyagar.messages.Cell attribute), 8  
y1 (pyagar.messages.Screen attribute), 9  
y2 (pyagar.messages.Screen attribute), 9

## Z

zoom (pyagar.messages.Camera attribute), 8