

# Distributed Pac-Man

Generated by Doxygen 1.6.3

Sun Apr 25 15:27:47 2010



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Package List . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Namespace Documentation</b>	<b>5</b>
3.1	Package board . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.2	Package client . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.3	Package game . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.3.2	Function Documentation . . . . .	8
3.3.2.1	intervalExecute . . . . .	8
3.4	Package matchmaker . . . . .	9
3.4.1	Detailed Description . . . . .	9
3.5	Package server . . . . .	10
3.5.1	Detailed Description . . . . .	10
3.6	board.board Class Reference . . . . .	11
3.6.1	Detailed Description . . . . .	11
3.6.2	Member Function Documentation . . . . .	11
3.6.2.1	canMove . . . . .	11
3.6.2.2	ghostStart . . . . .	12
3.6.2.3	pacmanStart . . . . .	12
3.7	client.client Class Reference . . . . .	13
3.7.1	Detailed Description . . . . .	13
3.8	game.game Class Reference . . . . .	14
3.8.1	Detailed Description . . . . .	14

3.8.2	Member Function Documentation . . . . .	14
3.8.2.1	__init__ . . . . .	14
3.9	matchmaker.matchmaker Class Reference . . . . .	15
3.9.1	Detailed Description . . . . .	15
3.10	game.state Class Reference . . . . .	16
3.10.1	Detailed Description . . . . .	16
3.10.2	Member Function Documentation . . . . .	16
3.10.2.1	__init__ . . . . .	16
3.10.2.2	changeType . . . . .	16
3.10.2.3	getState . . . . .	16
3.10.2.4	setState . . . . .	17

# Chapter 1

# Namespace Index

## 1.1 Package List

Here are the packages with brief descriptions (if available):

<a href="#">board</a> (Defines the board ) . . . . .	5
<a href="#">client</a> (Insert description ) . . . . .	6
<a href="#">game</a> (Controls the game state and handles the operation of the game ) . . . . .	7
<a href="#">matchmaker</a> (Controls the interface with the matchmaking server ) . . . . .	9
<a href="#">server</a> (Defines the server ) . . . . .	10



# Chapter 2

## Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">board.board</a> (A class to define the board ) . . . . .	11
<a href="#">client.client</a> (Insert description ) . . . . .	13
<a href="#">game.game</a> (A class to control the game ) . . . . .	14
<a href="#">matchmaker.matchmaker</a> (A class to help control the interface with the matchmaking server ) . .	15
<a href="#">game.state</a> (Defines the player's state ) . . . . .	16





## Chapter 3

# Namespace Documentation

### 3.1 Package board

Defines the board.

#### Classes

- class `board`

*A class to define the board.*

#### Variables

- dictionary `bops` = {'FLOOR':0, 'WALL':1, 'DOT':2, 'SUPER\_DOT':3}

*Defines options for tiles on the board.*

- dictionary `dirs` = {'LEFT':0, 'RIGHT':1, 'UP':2, 'DOWN':3}

*Defines the directions for movement.*

#### 3.1.1 Detailed Description

Defines the board.

#### Author

Myles Megyesi

## 3.2 Package client

insert description

### Classes

- class `client`  
*insert description*

### Functions

- def `run`  
*run from command line*

### Variables

- tuple `NAMESERVER` = `socket.gethostbyname(socket.gethostname())`
- int `NSPORT` = 5555
- tuple `file_lock` = `threading.Lock()`
- int `TIMEOUT` = 90

#### 3.2.1 Detailed Description

insert description

#### Author

Paul Schorfheide

## 3.3 Package game

Controls the game state and handles the operation of the game.

### Classes

- class `state`  
*Defines the player's state.*
- class `game`  
*A class to control the game.*

### Functions

- def `intervalExecute`  
*Executes a function repeatedly at the given interval.*

### Variables

- dictionary `dirs` = {'LEFT':0, 'RIGHT':1, 'UP':2, 'DOWN':3}  
*Defines the directions for movement.*
- dictionary `sops` = {'PACMAN':0, 'GHOST':1}  
*Defines the player types.*
- tuple `board` = `board.board()`  
*The global playing board.*
- tuple `mlock` = `threading.RLock()`  
*A semaphore used to when pushing or popping the messages queue.*
- float `update_interval` = 1.0  
*The interval at which to run the the game loop.*

#### 3.3.1 Detailed Description

Controls the game state and handles the operation of the game.

#### Author

Myles Megyesi

### 3.3.2 Function Documentation

#### 3.3.2.1 `def game.intervalExecute ( interval, func, args, argd)`

Executes a function repeatedly at the given interval.

##### Parameters

*interval* executes `func(*args, **argd)` each interval

##### Returns

a callable object to enable you terminate the timer

## 3.4 Package matchmaker

Controls the interface with the matchmaking server.

### Classes

- class `matchmaker`

*A class to help control the interface with the matchmaking server.*

### Functions

- def `formatPlayers`
- def `parseAddr`

#### 3.4.1 Detailed Description

Controls the interface with the matchmaking server.

#### Author

Paul Schorfheide

## 3.5 Package server

Defines the server.

### Functions

- def **listenForRequests**
- def **joinGame**
- def **addPlayer**
- def **makeTimer**
- def **parseRequest**
- def **changeLeader**
- def **clearGame**
- def **logAndSend**
- def **log**
- def **parseAddr**

### Variables

- list **games** = [ ]
- int **LISTEN\_PORT** = 5555
- string **LOGFILE\_NAME** = 'server.log'
- int **TIMEOUT** = 10
- int **logfile** = 0
- tuple **port** = int(sys.argv[1])
- dictionary **timers** = { }
- tuple **t** = threading.Thread(target=listenForRequests())

### 3.5.1 Detailed Description

Defines the server.

#### Author

Paul Schorfheide

# Class Documentation

## 3.6 board.board Class Reference

A class to define the board.

### Public Member Functions

- def `__init__`  
*Constructor.*
- def `canMove`  
*Given a direction and your current position, returns whether you a making a valid move.*
- def `pacmanStart`  
*Defines the starting position for a PACMAN player.*
- def `ghostStart`  
*Defines the starting position for a Ghost player.*

### Public Attributes

- `board`

### 3.6.1 Detailed Description

A class to define the board.

### 3.6.2 Member Function Documentation

#### 3.6.2.1 def board.board.canMove ( self, dir, x, y)

Given a direction and your current position, returns whether you a making a valid move.

#### Parameters

- dir* The direction to move
- (x,y)* A tuple of your x and y coordinates

**Returns**

A boolean indicating a valid move

**3.6.2.2 def board.board.ghostStart ( *self* )**

Defines the starting position for a Ghost player.

**Returns**

A tuple coordinates and position of the Ghost start

**3.6.2.3 def board.board.pacmanStart ( *self* )**

Defines the starting position for a PACMAN player.

**Returns**

a tuple coordinates and position of the PACMAN start

The documentation for this class was generated from the following file:

- src/board.py



## 3.7 client.client Class Reference

insert description

### Public Member Functions

- def **findGame**
- def **disconnect**
- def **getSelf**
- def **getLeader**
- def **getPlayers**
- def **send**
- def **sendToAll**
- def **\_\_init\_\_**
- def **log**

### 3.7.1 Detailed Description

insert description

The documentation for this class was generated from the following file:

- src/client.py

## 3.8 game.game Class Reference

A class to control the game.

### Public Member Functions

- def [disconnect](#)  
*Disconnects the player from the socket, used upon exit of game.*
- def [printStats2](#)
- def [printStats](#)
- def [draw](#)  
*Draws the board on the screen, with the players.*
- def [\\_\\_init\\_\\_](#)  
*Constructor.*
- def [update](#)  
*The game loop.*

### 3.8.1 Detailed Description

A class to control the game.

### 3.8.2 Member Function Documentation

#### 3.8.2.1 def game.game.\_\_init\_\_ ( self, server\_ip, server\_port = 5555, wait\_time = None, isSafe = True, printStates = True)

Constructor.

#### Parameters

- server\_ip** The IP address of the server
- server\_port** The port of the server
- wait\_time** The time to run the game. If not set, the game will run indefinitely
- isSafe** Boolean to toggle the timeout threads on and off
- printStates** Boolean to toggle the drawing on and off

The documentation for this class was generated from the following file:

- src/game.py

## 3.9 matchmaker.matchmaker Class Reference

A class to help control the interface with the matchmaking server.

### Public Member Functions

- def **getLeader**
- def **getAddress**
- def **changeLeader**
- def **getPlayers**
- def **removePlayer**
- def **findGame**
- def **disconnect**
- def **send**
- def **\_\_init\_\_**

### 3.9.1 Detailed Description

A class to help control the interface with the matchmaking server.

The documentation for this class was generated from the following file:

- src/matchmaker.py

## 3.10 game.state Class Reference

Defines the player's state.

### Public Member Functions

- def `changeType`  
*Changes the type of player.*
- def `getState`  
*State getter.*
- def `setState`  
*State setter.*
- def `move`
- def `__init__`  
*Constructor.*

### 3.10.1 Detailed Description

Defines the player's state.

### 3.10.2 Member Function Documentation

#### 3.10.2.1 `def game.state.__init__ ( self, type)`

Constructor.

##### Parameters

*type* The type to make the player

#### 3.10.2.2 `def game.state.changeType ( self, type)`

Changes the type of player.

Used during leader election

##### Parameters

*type* The player type to change to

#### 3.10.2.3 `def game.state.getState ( self)`

State getter.

##### Returns

The state of the player

**3.10.2.4** `def game.state.setState ( self, x, y, dir, type )`

State setter.

**Parameters**

- x* The X coordinate
- y* The Y coordinate
- dir* the direction to face
- type* The type of the player

The documentation for this class was generated from the following file:

- `src/game.py`