

Project: Jarl  
Design document  
Group Number: 18

Magnus Andermo  
Mattias Frånberg  
Carl Johan Gustavsson  
Pontus Stenetorp

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>System Overview</b>	<b>2</b>
<b>3</b>	<b>Design Considerations</b>	<b>2</b>
<b>4</b>	<b>Graphical user interface</b>	<b>2</b>
<b>5</b>	<b>Design details</b>	<b>2</b>

- 1 Introduction**
- 2 System Overview**
- 3 Design Considerations**
- 4 Graphical user interface**
- 5 Design details**

# Jarl Reference Manual

Generated by Doxygen 1.5.4

Mon Feb 18 15:02:36 2008



# Contents

<b>1 Jarl Hierarchical Index</b>	<b>1</b>
1.1 Jarl Class Hierarchy . . . . .	1
<b>2 Jarl Class Index</b>	<b>3</b>
2.1 Jarl Class List . . . . .	3
<b>3 Jarl Class Documentation</b>	<b>5</b>
3.1 game.client.units.Archer Class Reference . . . . .	5
3.2 game.client.buildings.Building Class Reference . . . . .	7
3.3 util.CAxisAlignedRect Class Reference . . . . .	10
3.4 game.server.Client Class Reference . . . . .	11
3.5 game.server.ClientController Class Reference . . . . .	12
3.6 game.server.CommandFilter Class Reference . . . . .	13
3.7 util.CPosition Class Reference . . . . .	14
3.8 util.CVec2 Class Reference . . . . .	16
3.9 lobby.server.DatabaseConnection Class Reference . . . . .	17
3.10 game.client.buildings.DefensiveBuilding Class Reference . . . . .	18
3.11 game.network.FrameEndCommand Class Reference . . . . .	20
3.12 game.client.Game Class Reference . . . . .	21
3.13 game.network.GameClientNetwork Class Reference . . . . .	22
3.14 game.client.GameObject Interface Reference . . . . .	24
3.15 game.server.GameServer Class Reference . . . . .	26
3.16 lobby.server.GameServerHandler Class Reference . . . . .	27
3.17 game.server.GameServerNetwork Class Reference . . . . .	28
3.18 game.client.Input Class Reference . . . . .	29
3.19 lobby.server.Lobby Class Reference . . . . .	30
3.20 lobby.client.LobbyGUI Class Reference . . . . .	31
3.21 lobby.server.LobbyServerNetwork Class Reference . . . . .	32
3.22 game.network.NetworkCommand Class Reference . . . . .	33

---

3.23	game.network.NetworkGameCommand Class Reference . . . . .	34
3.24	game.network.NetworkInteractionCommand Class Reference . . . . .	36
3.25	util.Position Class Reference . . . . .	37
3.26	game.client.buildings.ProductionBuilding Class Reference . . . . .	39
3.27	game.server.Proxy Class Reference . . . . .	41
3.28	game.client.units.Swordsman Class Reference . . . . .	42
3.29	game.client.units.Unit Class Reference . . . . .	44
3.30	util.Vector2 Class Reference . . . . .	47
3.31	game.client.World Class Reference . . . . .	49

# Chapter 1

## Jarl Hierarchical Index

### 1.1 Jarl Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

util.CAxisAlignedRect . . . . .	10
game.server.Client . . . . .	11
game.server.ClientController . . . . .	12
game.server.CommandFilter . . . . .	13
util.CPosition . . . . .	14
util.Position . . . . .	37
util.CVec2 . . . . .	16
util.Vec2 . . . . .	47
lobby.server.DatabaseConnection . . . . .	17
game.client.Game . . . . .	21
game.network.GameClientNetwork . . . . .	22
game.client.GameObject . . . . .	24
game.client.buildings.Building . . . . .	7
game.client.buildings.DefensiveBuilding . . . . .	18
game.client.buildings.ProductionBuilding . . . . .	39
game.client.units.Unit . . . . .	44
game.client.units.Archer . . . . .	5
game.client.units.Swordsman . . . . .	42
game.server.GameServer . . . . .	26
lobby.server.GameServerHandler . . . . .	27
game.server.GameServerNetwork . . . . .	28
game.client.Input . . . . .	29
lobby.server.Lobby . . . . .	30
lobby.client.LobbyGUI . . . . .	31
lobby.server.LobbyServerNetwork . . . . .	32
game.network.NetworkCommand . . . . .	33
game.network.FrameEndCommand . . . . .	20
game.network.NetworkGameCommand . . . . .	34
game.network.NetworkInteractionCommand . . . . .	36
game.server.Proxy . . . . .	41
game.client.World . . . . .	49





# Chapter 2

## Jarl Class Index

### 2.1 Jarl Class List

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

<b>game.client.units.Archer</b> . . . . .	5
<b>game.client.buildings.Building</b> . . . . .	7
<b>util.CAxisAlignedRect</b> . . . . .	10
<b>game.server.Client</b> . . . . .	11
<b>game.server.ClientController</b> . . . . .	12
<b>game.server.CommandFilter</b> . . . . .	13
<b>util.CPosition</b> . . . . .	14
<b>util.CVec2</b> . . . . .	16
<b>lobby.server.DatabaseConnection</b> . . . . .	17
<b>game.client.buildings.DefensiveBuilding</b> . . . . .	18
<b>game.network.FrameEndCommand</b> . . . . .	20
<b>game.client.Game</b> . . . . .	21
<b>game.network.GameClientNetwork</b> . . . . .	22
<b>game.client.GameObject</b> . . . . .	24
<b>game.server.GameServer</b> . . . . .	26
<b>lobby.server.GameServerHandler</b> . . . . .	27
<b>game.server.GameServerNetwork</b> . . . . .	28
<b>game.client.Input</b> . . . . .	29
<b>lobby.server.Lobby</b> . . . . .	30
<b>lobby.client.LobbyGUI</b> . . . . .	31
<b>lobby.server.LobbyServerNetwork</b> . . . . .	32
<b>game.network.NetworkCommand</b> . . . . .	33
<b>game.network.NetworkGameCommand</b> . . . . .	34
<b>game.network.NetworkInteractionCommand</b> . . . . .	36
<b>util.Position</b> . . . . .	37
<b>game.client.buildings.ProductionBuilding</b> . . . . .	39
<b>game.server.Proxy</b> . . . . .	41
<b>game.client.units.Swordsman</b> . . . . .	42
<b>game.client.units.Unit</b> . . . . .	44
<b>util.Vec2</b> . . . . .	47
<b>game.client.World</b> . . . . .	49

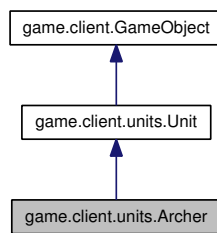


# Chapter 3

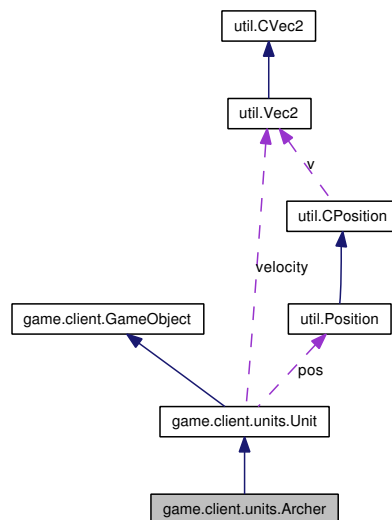
## Jarl Class Documentation

### 3.1 game.client.units.Archer Class Reference

Inheritance diagram for game.client.units.Archer:



Collaboration diagram for game.client.units.Archer:



## Public Member Functions

- float **getShootingRange** ()
- void **damage** (int amount)
- void **destroy** ()
- boolean **isDead** ()
- boolean **shallBeRemoved** ()
- void **think** ()

### 3.1.1 Detailed Description

Is a ranged unit that shoots arrows.

### 3.1.2 Member Function Documentation

#### 3.1.2.1 float `game.client.units.Archer.getShootingRange` ()

Returns how far the archer can shoot.

#### 3.1.2.2 void `game.client.units.Archer.damage` (int *amount*)

It shall implement a function that removes hit points from the object.

Reimplemented from `game.client.units.Unit` (p. 45).

#### 3.1.2.3 void `game.client.units.Archer.destroy` ()

Shall remove the object from the game world.

Reimplemented from `game.client.units.Unit` (p. 46).

#### 3.1.2.4 boolean `game.client.units.Archer.isDead` ()

Returns true if the object is considered to be dead.

Reimplemented from `game.client.units.Unit` (p. 46).

#### 3.1.2.5 boolean `game.client.units.Archer.shallBeRemoved` ()

Returns true if the object should be removed from the game.

Reimplemented from `game.client.units.Unit` (p. 46).

#### 3.1.2.6 void `game.client.units.Archer.think` ()

Shall perform the actions of a object that should be executed every frame.

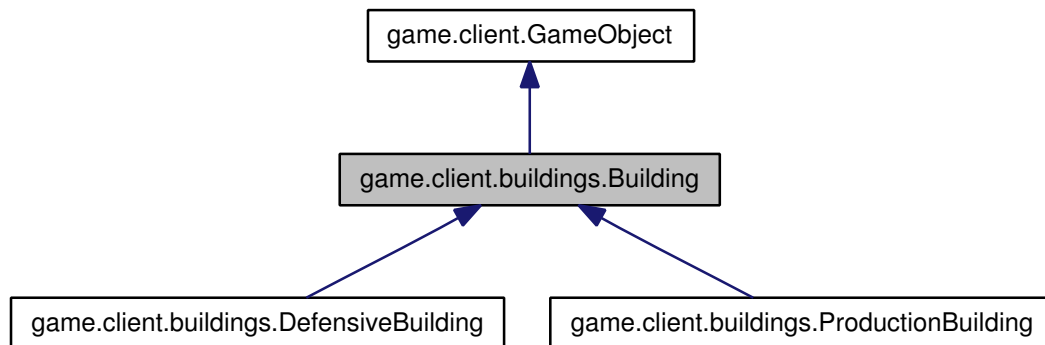
Reimplemented from `game.client.units.Unit` (p. 46).

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

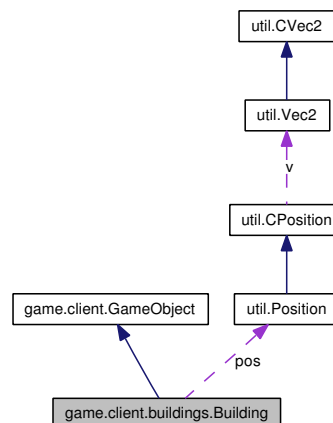
- `game/client/units/Archer.java`

## 3.2 game.client.buildings.Building Class Reference

Inheritance diagram for game.client.buildings.Building:



Collaboration diagram for game.client.buildings.Building:



### Public Member Functions

- final **CPosition** **getPos** ()
- final void **setPos** (**CPosition** pos)
- **CAxisAlignedRect** **getBoundingBox** ()
- void **damage** (int amount)
- void **destroy** ()
- boolean **isDead** ()
- boolean **shallBeRemoved** ()
- void **think** ()

### 3.2.1 Detailed Description

An abstract class describing behaviour of abuilding.

## 3.2.2 Member Function Documentation

### 3.2.2.1 final CPosition game.client.buildings.Building.getPos ()

Returns the position of the unit.

Implements **game.client.GameObject** (p. 25).

### 3.2.2.2 final void game.client.buildings.Building.setPos (CPosition pos)

Sets the position of the unit in the game world.

Implements **game.client.GameObject** (p. 25).

### 3.2.2.3 CAxisAlignedRect game.client.buildings.Building.getBoundingBox ()

The building is within the returned bounding box.

Implements **game.client.GameObject** (p. 25).

### 3.2.2.4 void game.client.buildings.Building.damage (int amount)

It shall implement a function that removes hit points from the object.

Implements **game.client.GameObject** (p. 25).

Reimplemented in **game.client.buildings.DefensiveBuilding** (p. 19).

### 3.2.2.5 void game.client.buildings.Building.destroy ()

Shall remove the object from the game world.

Implements **game.client.GameObject** (p. 24).

Reimplemented in **game.client.buildings.DefensiveBuilding** (p. 19), and **game.client.buildings.ProductionBuilding** (p. 39).

### 3.2.2.6 boolean game.client.buildings.Building.isDead ()

Returns true if the object is considered to be dead.

Implements **game.client.GameObject** (p. 25).

Reimplemented in **game.client.buildings.DefensiveBuilding** (p. 19).

### 3.2.2.7 boolean game.client.buildings.Building.shallBeRemoved ()

Returns true if the object should be removed from the game.

Implements **game.client.GameObject** (p. 24).

Reimplemented in **game.client.buildings.DefensiveBuilding** (p. 19).

### 3.2.2.8 `void game.client.buildings.Building.think ()`

Shall perform the actions of a object that should be executed every frame.

Implements `game.client.GameObject` (p. 25).

Reimplemented in `game.client.buildings.DefensiveBuilding` (p. 19), and `game.client.buildings.ProductionBuilding` (p. 40).

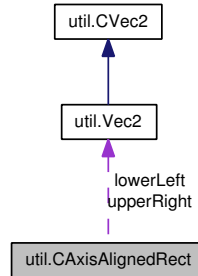
The documentation for this class was generated from the following file:

- `game/client/buildings/Building.java`



### 3.3 util.CAxisAlignedRect Class Reference

Collaboration diagram for util.CAxisAlignedRect:



#### Public Member Functions

- **CAxisAlignedRect** (**Vec2** *v*, float *width*, float *height*)
- boolean **isPointIn** (**Vec2** *v*)

#### 3.3.1 Detailed Description

A bounding box for a game object.

#### 3.3.2 Constructor & Destructor Documentation

##### 3.3.2.1 util.CAxisAlignedRect.CAxisAlignedRect (**Vec2** *v*, float *width*, float *height*)

###### Parameters:

- v* The point of the lower left corner of the bounding box.
- width* The width of the bounding box.
- height* The height of the bounding box.

#### 3.3.3 Member Function Documentation

##### 3.3.3.1 boolean util.CAxisAlignedRect.isPointIn (**Vec2** *v*)

Returns true if a point is inside the bounding box.

###### Parameters:

- v* Any point in the world.

###### Returns:

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

- util/CAxisAlignedRect.java

## 3.4 game.server.Client Class Reference

### Public Member Functions

- int `getID ()`

#### 3.4.1 Detailed Description

Holds data related to a **Client** (p. 11).

#### 3.4.2 Member Function Documentation

##### 3.4.2.1 int game.server.Client.getID ()

Returns the ID of a client.

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

- game/server/Client.java

## 3.5 game.server.ClientController Class Reference

### Public Member Functions

- void `processCommand` (`NetworkInteractionCommand cmd`)

### 3.5.1 Detailed Description

Handles client management.

**Author:**

ninjin

### 3.5.2 Member Function Documentation

#### 3.5.2.1 void game.server.ClientController.processCommand (`NetworkInteractionCommand cmd`)

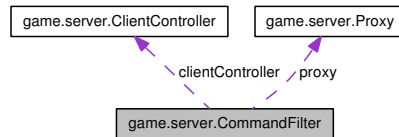
Processes network interaction commands.

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

- game/server/ClientController.java

## 3.6 game.server.CommandFilter Class Reference

Collaboration diagram for game.server.CommandFilter:



### Public Member Functions

- void **filter** (NetworkCommand cmd)

#### 3.6.1 Detailed Description

Holds the client controller and proxy and is also responsible for distributing network commands between them.

#### 3.6.2 Member Function Documentation

##### 3.6.2.1 void game.server.CommandFilter.filter (NetworkCommand cmd)

Distributes game commands and network interaction commands to the client controller and proxy respectively.

##### Parameters:

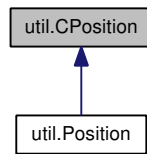
*cmd*

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

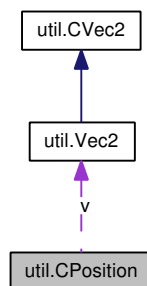
- game/server/CommandFilter.java

## 3.7 util.CPosition Class Reference

Inheritance diagram for util.CPosition:



Collaboration diagram for util.CPosition:



### Public Member Functions

- final float **x** ()
- final float **y** ()
- final **CVec2** **combatPos** ()
- final int **track** ()
- **Position** **copy** ()

### Package Attributes

- **Vec2** **v**
- int **track**

#### 3.7.1 Detailed Description

A constant position in the game.

#### 3.7.2 Member Function Documentation

##### 3.7.2.1 final float util.CPosition.x ()

Returns the x coordinate.

**3.7.2.2 final float util.CPosition.y ()**

Returns the y coordinate.

**3.7.2.3 final CVec2 util.CPosition.combatPos ()**

**Position** (p. 37) in the combat plane.

**3.7.2.4 final int util.CPosition.track ()**

Returns the game track of the object.

**3.7.2.5 Position util.CPosition.copy ()**

Returns a copy of the position.

**3.7.3 Member Data Documentation****3.7.3.1 Vec2 util.CPosition.v [package]**

**Position** (p. 37) in the 2d plane.

**3.7.3.2 int util.CPosition.track [package]**

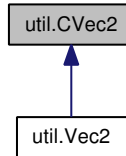
Game track.

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

- util/CPosition.java

## 3.8 util.CVec2 Class Reference

Inheritance diagram for util.CVec2:



### Public Member Functions

- final float **x** ()
- final float **y** ()
- **Vec2** **copy** ()

#### 3.8.1 Detailed Description

A constant vector with two floats. Usually used to represent a position in the a combat position plane. A combat position plane is a plane that a whole trace lies in.

### 3.8.2 Member Function Documentation

#### 3.8.2.1 final float util.CVec2.x ()

Returns the x cordinate.

#### 3.8.2.2 final float util.CVec2.y ()

Returns the y cordinate.

#### 3.8.2.3 Vec2 util.CVec2.copy ()

Returns a copy of the class.

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

- util/CVec2.java

## 3.9 lobby.server.DatabaseConnection Class Reference

### Public Member Functions

- void **insert** (String query)
- String **select** (String query)

### 3.9.1 Detailed Description

Handles the **Lobby** (p. 30) connection to the database.

### 3.9.2 Member Function Documentation

#### 3.9.2.1 void lobby.server.DatabaseConnection.insert (String *query*)

Inserts the given query into the database.

**Parameters:**

*query*

#### 3.9.2.2 String lobby.server.DatabaseConnection.select (String *query*)

Returns the result of the given database query.

**Parameters:**

*query*

**Returns:**

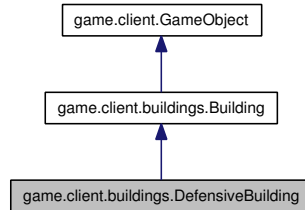
The documentation for this class was generated from the following file:

- lobby/server/DatabaseConnection.java

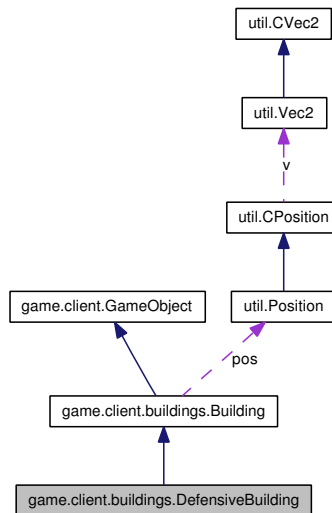


### 3.10 game.client.buildings.DefensiveBuilding Class Reference

Inheritance diagram for game.client.buildings.DefensiveBuilding:



Collaboration diagram for game.client.buildings.DefensiveBuilding:



#### Public Member Functions

- float **getShootingRange** ()
- void **damage** (int amount)
- void **destroy** ()
- boolean **isDead** ()
- boolean **shallBeRemoved** ()
- void **think** ()

#### 3.10.1 Detailed Description

The super class for the production buildings. A production building produces moving units.

#### 3.10.2 Member Function Documentation

##### 3.10.2.1 float game.client.buildings.DefensiveBuilding.getShootingRange ()

Returns the distance the defensive building can shoot.

**3.10.2.2** `void game.client.buildings.DefensiveBuilding.damage (int amount)`

It shall implement a function that removes hit points from the object.

Reimplemented from `game.client.buildings.Building` (p. 8).

**3.10.2.3** `void game.client.buildings.DefensiveBuilding.destroy ()`

Shall remove the object from the game world.

Reimplemented from `game.client.buildings.Building` (p. 8).

**3.10.2.4** `boolean game.client.buildings.DefensiveBuilding.isDead ()`

Returns true if the object is considered to be dead.

Reimplemented from `game.client.buildings.Building` (p. 8).

**3.10.2.5** `boolean game.client.buildings.DefensiveBuilding.shallBeRemoved ()`

Returns true if the object should be removed from the game.

Reimplemented from `game.client.buildings.Building` (p. 8).

**3.10.2.6** `void game.client.buildings.DefensiveBuilding.think ()`

Shall perform the actions of a object that should be executed every frame.

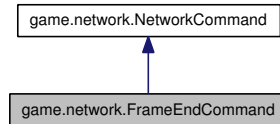
Reimplemented from `game.client.buildings.Building` (p. 9).

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

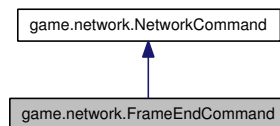
- `game/client/buildings/DefensiveBuilding.java`

## 3.11 game.network.FrameEndCommand Class Reference

Inheritance diagram for game.network.FrameEndCommand:



Collaboration diagram for game.network.FrameEndCommand:



### Public Member Functions

- **FrameEndCommand** (int frameID)
- int **getFrameID** ()

#### 3.11.1 Detailed Description

Represents the end of a frame.

#### 3.11.2 Constructor & Destructor Documentation

##### 3.11.2.1 game.network.FrameEndCommand.FrameEndCommand (int *frameID*)

See also:

[NetworkCommand](#) (p. 33)

#### 3.11.3 Member Function Documentation

##### 3.11.3.1 int game.network.FrameEndCommand.getFrameID ()

See also:

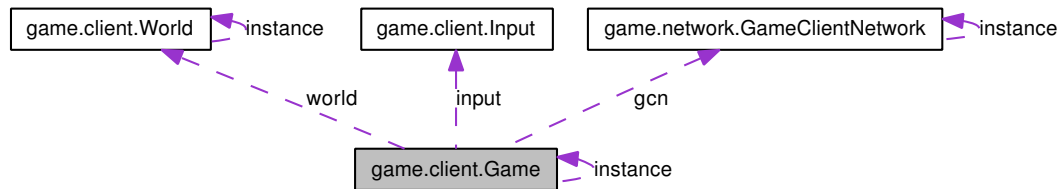
[NetworkCommand.getFrameId\(\)](#)

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

- [game/network/FrameEndCommand.java](#)

## 3.12 game.client.Game Class Reference

Collaboration diagram for game.client.Game:



### Static Public Member Functions

- static final **Game** ins ()

#### 3.12.1 Detailed Description

Main class of the client.

#### 3.12.2 Member Function Documentation

##### 3.12.2.1 static final **Game** game.client.Game.ins () [static]

Creates a singleton instance of game.

##### Returns:

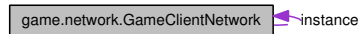
- the static reference to **Game** (p. 21).

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

- game/client/Game.java

### 3.13 game.network.GameClientNetwork Class Reference

Collaboration diagram for game.network.GameClientNetwork:



#### Public Member Functions

- abstract void **sendGameCommand** (**NetworkGameCommand** c)
- abstract void **sendFrameEndCommand** ()
- abstract Collection< **NetworkGameCommand** > **popCommandsForCurrentFrame** ()

#### Static Public Member Functions

- static **GameClientNetwork** **ins** ()
- static void **setInstance** (**GameClientNetwork** instance)

#### 3.13.1 Detailed Description

The client network class. When we test **ins ()** (p. 22) will probably return a instance of a class that will emulate a GameServerNetwork.

##### Author:

mandermo

#### 3.13.2 Member Function Documentation

##### 3.13.2.1 static GameClientNetwork game.network.GameClientNetwork.ins () [static]

Return the "singleton instance"

##### Returns:

##### 3.13.2.2 static void game.network.GameClientNetwork.setInstance (GameClientNetwork instance) [static]

Sets the "singleton" instance.

##### Parameters:

*instance*

**3.13.2.3 abstract void game.network.GameClientNetwork.sendGameCommand  
(NetworkGameCommand c) [pure virtual]**

Sends a game command that will be replicated to all clients and scheduled to run at the same frame on all clients.

**Parameters:**

*c* The command to be sent.

**3.13.2.4 abstract void game.network.GameClientNetwork.sendFrameEndCommand () [pure  
virtual]**

Called to signal all commands for current frame is sent.

**3.13.2.5 abstract Collection<NetworkGameCommand>  
game.network.GameClientNetwork.popCommandsForCurrentFrame ()  
[pure virtual]**

Called to get all commands that should be executed on this frame. Will pop those commands from the network.

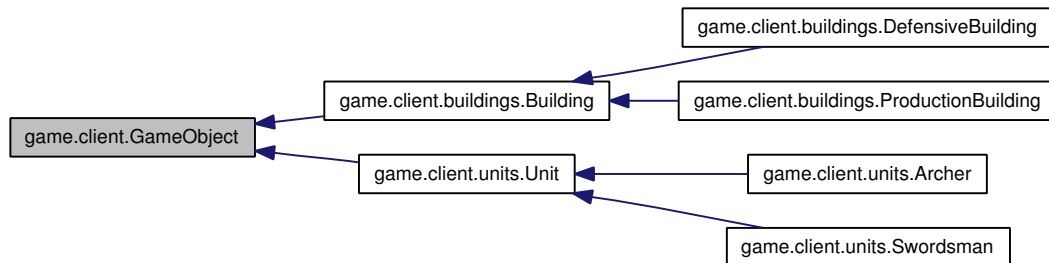
**Returns:**

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

- game/network/GameClientNetwork.java

## 3.14 game.client.GameObject Interface Reference

Inheritance diagram for game.client.GameObject:



### Public Member Functions

- boolean **shallBeRemoved** ()
- void **destroy** ()
- boolean **isDead** ()
- void **think** ()
- void **damage** (int amount)
- **CPosition** **getPos** ()
- void **setPos** (**CPosition** pos)
- **CAxisAlignedRect** **getBoundingBox** ()

### 3.14.1 Detailed Description

An interface that every object that can perform some kind of logic shall implement.

### 3.14.2 Member Function Documentation

#### 3.14.2.1 boolean game.client.GameObject.shallBeRemoved ()

Returns true if the object should be removed from the game.

Implemented in **game.client.buildings.Building** (p. 8), **game.client.buildings.DefensiveBuilding** (p. 19), **game.client.units.Archer** (p. 6), **game.client.units.Swordsman** (p. 43), and **game.client.units.Unit** (p. 46).

#### 3.14.2.2 void game.client.GameObject.destroy ()

Shall remove the object from the game world.

Implemented in **game.client.buildings.Building** (p. 8), **game.client.buildings.DefensiveBuilding** (p. 19), **game.client.buildings.ProductionBuilding** (p. 39), **game.client.units.Archer** (p. 6), **game.client.units.Swordsman** (p. 43), and **game.client.units.Unit** (p. 46).

### 3.14.2.3 `boolean game.client.GameObject.isDead ()`

Returns true if the object is considered to be dead.

Implemented in `game.client.buildings.Building` (p. 8), `game.client.buildings.DefensiveBuilding` (p. 19), `game.client.units.Archer` (p. 6), `game.client.units.Swordsman` (p. 43), and `game.client.units.Unit` (p. 46).

### 3.14.2.4 `void game.client.GameObject.think ()`

Shall perform the actions of a object that should be executed every frame.

Implemented in `game.client.buildings.Building` (p. 9), `game.client.buildings.DefensiveBuilding` (p. 19), `game.client.buildings.ProductionBuilding` (p. 40), `game.client.units.Archer` (p. 6), `game.client.units.Swordsman` (p. 43), and `game.client.units.Unit` (p. 46).

### 3.14.2.5 `void game.client.GameObject.damage (int amount)`

It shall implement a function that removes hit points from the object.

Implemented in `game.client.buildings.Building` (p. 8), `game.client.buildings.DefensiveBuilding` (p. 19), `game.client.units.Archer` (p. 6), `game.client.units.Swordsman` (p. 42), and `game.client.units.Unit` (p. 45).

### 3.14.2.6 `CPosition game.client.GameObject.getPos ()`

Returns the position of the unit in the lower left corner..

Implemented in `game.client.buildings.Building` (p. 8), and `game.client.units.Unit` (p. 45).

### 3.14.2.7 `void game.client.GameObject.setPos (CPosition pos)`

Sets the position of the unit in the game world.

Implemented in `game.client.buildings.Building` (p. 8), and `game.client.units.Unit` (p. 45).

### 3.14.2.8 `CAxisAlignedRect game.client.GameObject.getBoundingBox ()`

Returns the bounding box of a game object inside the world.

Implemented in `game.client.buildings.Building` (p. 8), and `game.client.units.Unit` (p. 45).

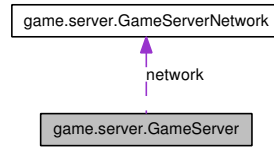
The documentation for this interface was generated from the following file:

- `game/client/GameObject.java`



## 3.15 game.server.GameServer Class Reference

Collaboration diagram for game.server.GameServer:



### Public Member Functions

- void `run ()`

#### 3.15.1 Detailed Description

Responsible for setting up and handling the game server.

#### 3.15.2 Member Function Documentation

##### 3.15.2.1 void game.server.GameServer.run ()

Listens on the given port and verifies that the given client id is correct.

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

- game/server/GameServer.java

## 3.16 lobby.server.GameServerHandler Class Reference

### Public Member Functions

- void **startGameServer** (ArrayList< **Client** > clients, int port)
- void **run** ()

#### 3.16.1 Detailed Description

Handles the collection of GameServers that are currently running or that have finished their games and needs to be reaped.

#### 3.16.2 Member Function Documentation

##### 3.16.2.1 void lobby.server.GameServerHandler.startGameServer (ArrayList< Client > *clients*, int *port*)

Starts a new game server for the given clients.

##### Parameters:

- clients*
- port*

##### 3.16.2.2 void lobby.server.GameServerHandler.run ()

Iterates over the game servers and checks their status.

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

- lobby/server/GameServerHandler.java

## 3.17 game.server.GameServerNetwork Class Reference

### Public Member Functions

- void **send** (**NetworkCommand** cmd)

### Package Attributes

- ArrayList< Socket > **clients**

### 3.17.1 Detailed Description

Handles network connections with the clients.

### 3.17.2 Member Function Documentation

#### 3.17.2.1 void game.server.GameServerNetwork.send (**NetworkCommand** cmd)

Sends a given command.

#### Parameters:

*cmd*

### 3.17.3 Member Data Documentation

#### 3.17.3.1 ArrayList<Socket> game.server.GameServerNetwork.clients [package]

A list of sockets to the clients.

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

- game/server/GameServerNetwork.java

## 3.18 game.client.Input Class Reference

### Public Member Functions

- boolean **isLeftMouseButtonClicked** ()
- boolean **isRightMouseButtonClicked** ()
- ArrayList **keysPressed** ()

#### 3.18.1 Detailed Description

Is a wrapper class to JInput devices.

#### 3.18.2 Member Function Documentation

##### 3.18.2.1 boolean game.client.Input.isLeftMouseButtonClicked ()

Resets event data. The input gotten after All input gotten after the reset is Returns true if the left mouse button is clicked this frame.

##### 3.18.2.2 boolean game.client.Input.isRightMouseButtonClicked ()

Returns true if the right mouse button is clicked this frame.

##### 3.18.2.3 ArrayList game.client.Input.keysPressed ()

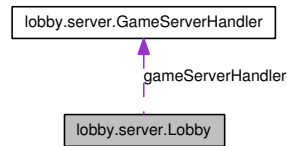
Returns a list of keys pressed.

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

- game/client/Input.java

## 3.19 lobby.server.Lobby Class Reference

Collaboration diagram for lobby.server.Lobby:



### Public Member Functions

- void **run** ()

#### 3.19.1 Detailed Description

Responsible for managing the game lobby.

#### 3.19.2 Member Function Documentation

##### 3.19.2.1 void lobby.server.Lobby.run ()

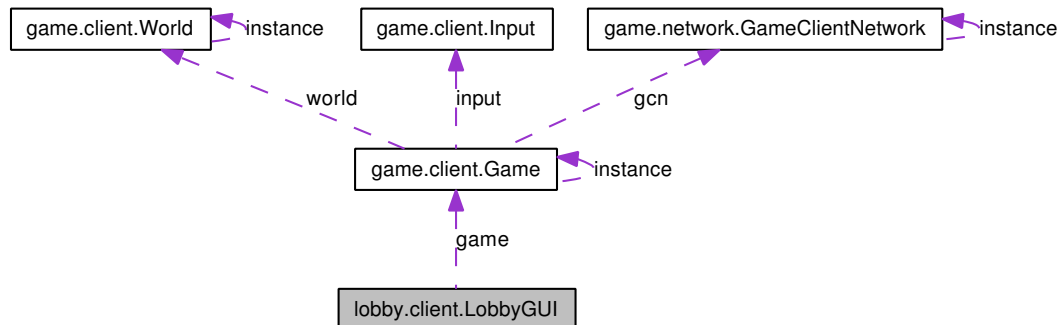
Continuously pushes and processes commands from the clients. The commands contain the current data for the lobby.

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

- lobby/server/Lobby.java

## 3.20 lobby.client.LobbyGUI Class Reference

Collaboration diagram for lobby.client.LobbyGUI:



### Public Member Functions

- LobbyGUI ()

#### 3.20.1 Detailed Description

Handles the **LobbyGUI** (p. 31) using Swing. Also sets up the game object.

#### 3.20.2 Constructor & Destructor Documentation

##### 3.20.2.1 lobby.client.LobbyGUI.LobbyGUI ()

Sets up the full-fledged GUI using Swing.

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

- lobby/client/LobbyGUI.java

## 3.21 lobby.server.LobbyServerNetwork Class Reference

### Public Member Functions

- void **broadcast** (**NetworkCommand** *cmd*)

### Package Attributes

- int **port**
- ArrayList< Socket > **clients**

### 3.21.1 Detailed Description

Responsible for the lobby network connection.

### 3.21.2 Member Function Documentation

#### 3.21.2.1 void lobby.server.LobbyServerNetwork.broadcast (**NetworkCommand** *cmd*)

Pushes the given command to all connected clients.

#### Parameters:

*cmd*

### 3.21.3 Member Data Documentation

#### 3.21.3.1 int lobby.server.LobbyServerNetwork.port [package]

Connection port number.

#### 3.21.3.2 ArrayList<Socket> lobby.server.LobbyServerNetwork.clients [package]

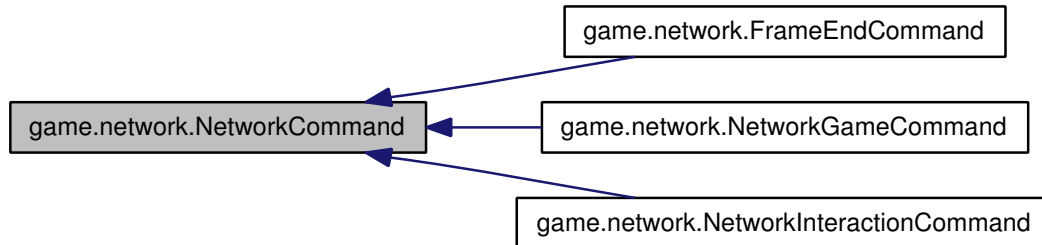
A list of connected clients.

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

- lobby/server/LobbyServerNetwork.java

## 3.22 game.network.NetworkCommand Class Reference

Inheritance diagram for game.network.NetworkCommand:



### 3.22.1 Detailed Description

An abstract description of commands sent of the network.

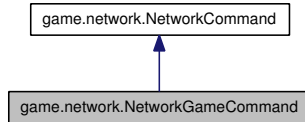
The documentation for this class was generated from the following file:

- `game/network/NetworkCommand.java`

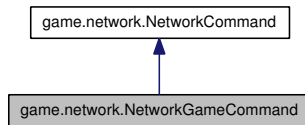


## 3.23 game.network.NetworkGameCommand Class Reference

Inheritance diagram for game.network.NetworkGameCommand:



Collaboration diagram for game.network.NetworkGameCommand:



### Public Member Functions

- final int **getSentFrame** ()
- final void **setSentFrame** (int sentFrame)
- final int **getExecuteFrame** ()
- final void **setExecuteFrame** (int x)
- abstract void **execute** ()

#### 3.23.1 Detailed Description

A command representing actions during the game play.

#### 3.23.2 Member Function Documentation

##### 3.23.2.1 final int game.network.NetworkGameCommand.getSentFrame ()

Returns in which frame the command was sent.

##### 3.23.2.2 final void game.network.NetworkGameCommand.setSentFrame (int *sentFrame*)

Sets the sent frame of the command.

##### 3.23.2.3 final int game.network.NetworkGameCommand.getExecuteFrame ()

Returns the frame in which the command should be executed.

##### 3.23.2.4 final void game.network.NetworkGameCommand.setExecuteFrame (int *x*)

Sets the frame in which the command should be executed.

**3.23.2.5 abstract void game.network.NetworkGameCommand.execute ()** [pure virtual]

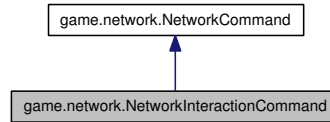
Shall implement the execution of a network command.

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

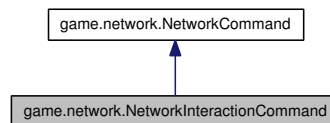
- game/network/NetworkGameCommand.java

## 3.24 game.network.NetworkInteractionCommand Class Reference

Inheritance diagram for game.network.NetworkInteractionCommand:



Collaboration diagram for game.network.NetworkInteractionCommand:



### 3.24.1 Detailed Description

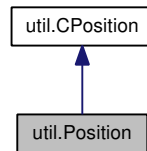
Used to transfer information regarding network activity. Such as time-out and leaving.

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

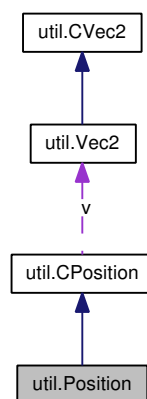
- game/network/NetworkInteractionCommand.java

## 3.25 util.Position Class Reference

Inheritance diagram for util.Position:



Collaboration diagram for util.Position:



### Public Member Functions

- **Position** (**CVec2** *v*, int *track*)
- **Position** (float *x*, float *y*, int *track*)
- final void **x** (float *xx*)
- final void **y** (float *yy*)
- final void **track** (int *t*)
- final void **combatPos** (**CVec2** *v*)

### 3.25.1 Detailed Description

The non constant version of **CPosition** (p. 14).

### 3.25.2 Constructor & Destructor Documentation

#### 3.25.2.1 util.Position.Position (**CVec2** *v*, int *track*)

**Parameters:**

- v* - **Position** (p. 37) in the 2d plane.
- track* - The track of the game object.

### 3.25.2.2 `util.Position.Position` (float *x*, float *y*, int *track*)

#### Parameters:

*x* - x cordinate of the object.

*y* - y cordinate of the object.

*track* - track of the object.

## 3.25.3 Member Function Documentation

### 3.25.3.1 `final void util.Position.x` (float *xx*)

Returns the x cordinate.

### 3.25.3.2 `final void util.Position.y` (float *yy*)

Returns the y cordinate.

### 3.25.3.3 `final void util.Position.track` (int *t*)

Returns the track.

### 3.25.3.4 `final void util.Position.combatPos` (CVec2 *v*)

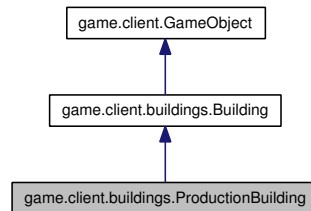
Returns the combat position.

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

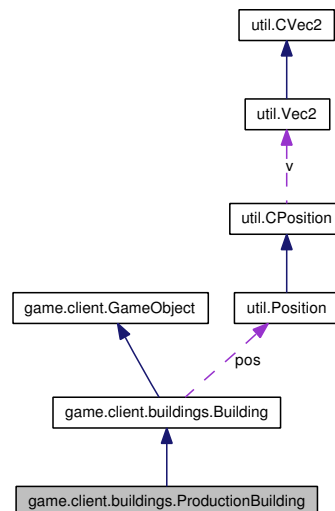
- `util/Position.java`

## 3.26 game.client.buildings.ProductionBuilding Class Reference

Inheritance diagram for game.client.buildings.ProductionBuilding:



Collaboration diagram for game.client.buildings.ProductionBuilding:



### Public Member Functions

- void **destroy** ()
- void **think** ()

#### 3.26.1 Detailed Description

The class for the production buildings. A production building produces moving units. The game uses the Prototype Design Pattern to create new buildings.

#### 3.26.2 Member Function Documentation

##### 3.26.2.1 void game.client.buildings.ProductionBuilding.destroy ()

See also:

[game.client.buildings.Building.destroy\(\)](#) (p. 8)

Reimplemented from **game.client.buildings.Building** (p. 8).

### 3.26.2.2 void **game.client.buildings.ProductionBuilding.think** ()

Spawn a unit depending on a counter state.

**See also:**

**game.client.buildings.Building.think()** (p. 9)

Reimplemented from **game.client.buildings.Building** (p. 9).

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

- `game/client/buildings/ProductionBuilding.java`

## 3.27 game.server.Proxy Class Reference

### Public Member Functions

- void **broadcast** (*NetworkGameCommand cmd*)

### 3.27.1 Detailed Description

The **Proxy** (p. 41) is responsible for tagging and broadcasting *NetworkGameCommands* to all connected clients.

### 3.27.2 Member Function Documentation

#### 3.27.2.1 void game.server.Proxy.broadcast (*NetworkGameCommand cmd*)

Tags and broadcasts the given *NetworkGameCommand* to all connected clients.

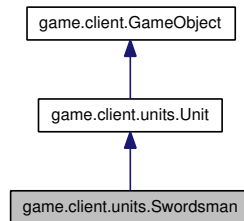
The documentation for this class was generated from the following file:

- game/server/Proxy.java

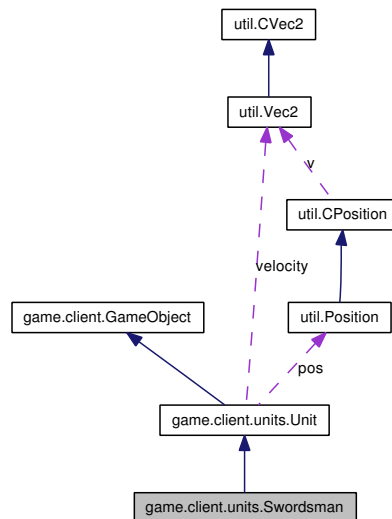


## 3.28 game.client.units.Swordsman Class Reference

Inheritance diagram for game.client.units.Swordsman:



Collaboration diagram for game.client.units.Swordsman:



### Public Member Functions

- void **damage** (int amount)
- void **destroy** ()
- boolean **isDead** ()
- boolean **shallBeRemoved** ()
- void **think** ()

### 3.28.1 Detailed Description

Is a melee unit that attacks units within range.

### 3.28.2 Member Function Documentation

#### 3.28.2.1 void game.client.units.Swordsman.damage (int amount)

It shall implement a function that removes hit points from the object.

Reimplemented from `game.client.units.Unit` (p. 45).

#### 3.28.2.2 `void game.client.units.Swordsman.destroy ()`

Shall remove the object from the game world.

Reimplemented from `game.client.units.Unit` (p. 46).

#### 3.28.2.3 `boolean game.client.units.Swordsman.isDead ()`

Returns true if the object is considered to be dead.

Reimplemented from `game.client.units.Unit` (p. 46).

#### 3.28.2.4 `boolean game.client.units.Swordsman.shallBeRemoved ()`

Returns true if the object should be removed from the game.

Reimplemented from `game.client.units.Unit` (p. 46).

#### 3.28.2.5 `void game.client.units.Swordsman.think ()`

Shall perform the actions of a object that should be executed every frame.

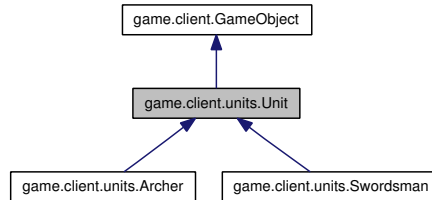
Reimplemented from `game.client.units.Unit` (p. 46).

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

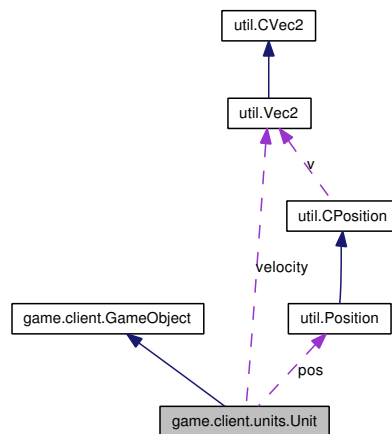
- `game/client/units/Swordsman.java`

## 3.29 game.client.units.Unit Class Reference

Inheritance diagram for game.client.units.Unit:



Collaboration diagram for game.client.units.Unit:



### Public Types

- enum **Responsibility**

### Public Member Functions

- final **Responsibility** **getResponsibility** ()
- final void **setResponsibility** (**Responsibility** responsibility)
- final **CPosition** **getPos** ()
- final void **setPos** (**CPosition** pos)
- **CAxisAlignedRect** **getBoundingBox** ()
- void **damage** (int amount)
- void **destroy** ()
- boolean **isDead** ()
- boolean **shallBeRemoved** ()
- void **think** ()

#### 3.29.1 Detailed Description

The base class for units.

## 3.29.2 Member Enumeration Documentation

### 3.29.2.1 enum game::client::units::Unit::Responsibility

A unit can be responsible for either attacking or defending.

**Author:**

mandermo

## 3.29.3 Member Function Documentation

### 3.29.3.1 final Responsibility game.client.units.Unit.getResponsibility ()

See also:

**Responsibility** (p. 45)

### 3.29.3.2 final void game.client.units.Unit.setResponsibility (Responsibility *responsibility*)

**Parameters:**

*responsibility* The responsibility to set.

### 3.29.3.3 final CPosition game.client.units.Unit.getPos ()

Returns the position of the unit.

Implements **game.client.GameObject** (p. 25).

### 3.29.3.4 final void game.client.units.Unit.setPos (CPosition *pos*)

Sets the position of the unit in the game world.

Implements **game.client.GameObject** (p. 25).

### 3.29.3.5 CAxisAlignedRect game.client.units.Unit.getBoundingBox ()

The unit is within the returned bounding box.

Implements **game.client.GameObject** (p. 25).

### 3.29.3.6 void game.client.units.Unit.damage (int *amount*)

It shall implement a function that removes hit points from the object.

Implements **game.client.GameObject** (p. 25).

Reimplemented in **game.client.units.Archer** (p. 6), and **game.client.units.Swordsman** (p. 42).

**3.29.3.7 void game.client.units.Unit.destroy ()**

Shall remove the object from the game world.

Implements **game.client.GameObject** (p. 24).

Reimplemented in **game.client.units.Archer** (p. 6), and **game.client.units.Swordsman** (p. 43).

**3.29.3.8 boolean game.client.units.Unit.isDead ()**

Returns true if the object is considered to be dead.

Implements **game.client.GameObject** (p. 25).

Reimplemented in **game.client.units.Archer** (p. 6), and **game.client.units.Swordsman** (p. 43).

**3.29.3.9 boolean game.client.units.Unit.shallBeRemoved ()**

Returns true if the object should be removed from the game.

Implements **game.client.GameObject** (p. 24).

Reimplemented in **game.client.units.Archer** (p. 6), and **game.client.units.Swordsman** (p. 43).

**3.29.3.10 void game.client.units.Unit.think ()**

Shall perform the actions of a object that should be executed every frame.

Implements **game.client.GameObject** (p. 25).

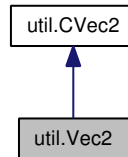
Reimplemented in **game.client.units.Archer** (p. 6), and **game.client.units.Swordsman** (p. 43).

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

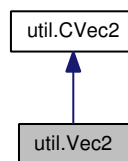
- game/client/units/Unit.java

## 3.30 util.Vec2 Class Reference

Inheritance diagram for util.Vec2:



Collaboration diagram for util.Vec2:



### Public Member Functions

- **Vec2** (float xx, float yy)
- final void **x** (float xx)
- final void **y** (float yy)

#### 3.30.1 Detailed Description

A non constant version of **CVec2** (p. 16).

#### 3.30.2 Constructor & Destructor Documentation

##### 3.30.2.1 util.Vec2.Vec2 (float xx, float yy)

Creates a **Vec2** (p. 47) instance.

##### Parameters:

- xx** - The x coordinate.
- yy** - The y coordinate.

#### 3.30.3 Member Function Documentation

##### 3.30.3.1 final void util.Vec2.x (float xx)

Sets the x coordinate.

##### Parameters:

- xx** - The x coordinate.

### 3.30.3.2 final void util.Vec2.y (float yy)

Sets the y cordinate.

**Parameters:**

`yy` - The y cordinate.

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

- util/Vec2.java

## 3.31 game.client.World Class Reference

Collaboration diagram for game.client.World:



### Public Member Functions

- `Iterator< TraceInfo > trace (Vec2 combatPos, int track, TraceInfo.Dir dir, float dist, Set< TraceInfo.Attr > attr)`

#### 3.31.1 Detailed Description

Contains all game objects and can be used to trace units and buildings.

#### 3.31.2 Member Function Documentation

##### 3.31.2.1 `Iterator<TraceInfo> game.client.World.trace (Vec2 combatPos, int track, TraceInfo.Dir dir, float dist, Set< TraceInfo.Attr > attr)`

Returns a iterator that can be traversed to incrementally get units or buildings in front of `combatPos`.

##### Parameters:

- combatPos* The position that the trace starts from.
- track* The track to trace on or all tracks if `track==-1`.
- dir* The direction to trace in.
- dist* The max distance to trace.
- attr* Attributes for the trace.

##### Returns:

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

- `game/client/World.java`



# Index

- broadcast
  - game::server::Proxy, 41
  - lobby::server::LobbyServerNetwork, 32
- CAxisAlignedRect
  - util::CAxisAlignedRect, 10
- clients
  - game::server::GameServerNetwork, 28
  - lobby::server::LobbyServerNetwork, 32
- combatPos
  - util::CPosition, 15
  - util::Position, 38
- copy
  - util::CPosition, 15
  - util::CVec2, 16
- damage
  - game::client::buildings::Building, 8
  - game::client::buildings::DefensiveBuilding, 18
  - game::client::GameObject, 25
  - game::client::units::Archer, 6
  - game::client::units::Swordsman, 42
  - game::client::units::Unit, 45
- destroy
  - game::client::buildings::Building, 8
  - game::client::buildings::DefensiveBuilding, 19
  - game::client::buildings::ProductionBuilding, 39
  - game::client::GameObject, 24
  - game::client::units::Archer, 6
  - game::client::units::Swordsman, 43
  - game::client::units::Unit, 45
- execute
  - game::network::NetworkGameCommand, 34
- filter
  - game::server::CommandFilter, 13
- FrameEndCommand
  - game::network::FrameEndCommand, 20
- game::client::buildings::Building, 7
  - damage, 8
  - destroy, 8
  - getBoundingBox, 8
  - getPos, 8
  - isDead, 8
  - setPos, 8
  - shallBeRemoved, 8
  - think, 8
- game::client::buildings::DefensiveBuilding, 18
  - damage, 18
  - destroy, 19
  - getShootingRange, 18
  - isDead, 19
  - shallBeRemoved, 19
  - think, 19
- game::client::buildings::ProductionBuilding, 39
  - destroy, 39
  - think, 40
- game::client::Game, 21
  - ins, 21
- game::client::GameObject, 24
  - damage, 25
  - destroy, 24
  - getBoundingBox, 25
  - getPos, 25
  - isDead, 24
  - setPos, 25
  - shallBeRemoved, 24
  - think, 25
- game::client::Input, 29
  - isLeftMouseButtonClicked, 29
  - isRightMouseButtonClicked, 29
  - keysPressed, 29
- game::client::units::Archer, 5
  - damage, 6
  - destroy, 6
  - getShootingRange, 6
  - isDead, 6
  - shallBeRemoved, 6
  - think, 6
- game::client::units::Swordsman, 42
  - damage, 42
  - destroy, 43
  - isDead, 43
  - shallBeRemoved, 43
  - think, 43
- game::client::units::Unit, 44
  - damage, 45
  - destroy, 45

- getBoundingBox, 45
- getPos, 45
- getResponsibility, 45
- isDead, 46
- Responsibility, 45
- setPos, 45
- setResponsibility, 45
- shallBeRemoved, 46
- think, 46
- game::client::World, 49
  - trace, 49
- game::network::FrameEndCommand, 20
  - FrameEndCommand, 20
  - getFrameID, 20
- game::network::GameClientNetwork, 22
  - ins, 22
  - popCommandsForCurrentFrame, 23
  - sendFrameEndCommand, 23
  - sendGameCommand, 22
  - setInstance, 22
- game::network::NetworkCommand, 33
- game::network::NetworkGameCommand, 34
  - execute, 34
  - getExecuteFrame, 34
  - getSentFrame, 34
  - setExecuteFrame, 34
  - setSentFrame, 34
- game::network::NetworkInteractionCommand, 36
- game::server::Client, 11
  - getID, 11
- game::server::ClientController, 12
  - processCommand, 12
- game::server::CommandFilter, 13
  - filter, 13
- game::server::GameServer, 26
  - run, 26
- game::server::GameServerNetwork, 28
  - clients, 28
  - send, 28
- game::server::Proxy, 41
  - broadcast, 41
- getBoundingBox
  - game::client::buildings::Building, 8
  - game::client::GameObject, 25
  - game::client::units::Unit, 45
- getExecuteFrame
  - game::network::NetworkGameCommand, 34
- getFrameID
  - game::network::FrameEndCommand, 20
- getID
  - game::server::Client, 11
- getPos
  - game::client::buildings::Building, 8
  - game::client::GameObject, 25
  - game::client::units::Unit, 45
- getResponsibility
  - game::client::units::Unit, 45
- getSentFrame
  - game::network::NetworkGameCommand, 34
- getShootingRange
  - game::client::buildings::DefensiveBuilding, 18
  - game::client::units::Archer, 6
- ins
  - game::client::Game, 21
  - game::network::GameClientNetwork, 22
- insert
  - lobby::server::DatabaseConnection, 17
- isDead
  - game::client::buildings::Building, 8
  - game::client::buildings::DefensiveBuilding, 19
  - game::client::GameObject, 24
  - game::client::units::Archer, 6
  - game::client::units::Swordsman, 43
  - game::client::units::Unit, 46
- isLeftMouseButtonClicked
  - game::client::Input, 29
- isPointIn
  - util::CAxisAlignedRect, 10
- isRightMouseButtonClicked
  - game::client::Input, 29
- keysPressed
  - game::client::Input, 29
- lobby::client::LobbyGUI, 31
  - LobbyGUI, 31
- lobby::server::DatabaseConnection, 17
  - insert, 17
  - select, 17
- lobby::server::GameServerHandler, 27
  - run, 27
  - startGameServer, 27
- lobby::server::Lobby, 30
  - run, 30
- lobby::server::LobbyServerNetwork, 32
  - broadcast, 32
  - clients, 32
  - port, 32
- LobbyGUI
  - lobby::client::LobbyGUI, 31
- popCommandsForCurrentFrame
  - game::network::GameClientNetwork, 23
- port
  - lobby::server::LobbyServerNetwork, 32
- Position
  - util::Position, 37

- processCommand
  - game::server::ClientController, 12
- Responsibility
  - game::client::units::Unit, 45
- run
  - game::server::GameServer, 26
  - lobby::server::GameServerHandler, 27
  - lobby::server::Lobby, 30
- select
  - lobby::server::DatabaseConnection, 17
- send
  - game::server::GameServerNetwork, 28
- sendFrameEndCommand
  - game::network::GameClientNetwork, 23
- sendGameCommand
  - game::network::GameClientNetwork, 22
- setExecuteFrame
  - game::network::NetworkGameCommand, 34
- setInstance
  - game::network::GameClientNetwork, 22
- setPos
  - game::client::buildings::Building, 8
  - game::client::GameObject, 25
  - game::client::units::Unit, 45
- setResponsibility
  - game::client::units::Unit, 45
- setSentFrame
  - game::network::NetworkGameCommand, 34
- shallBeRemoved
  - game::client::buildings::Building, 8
  - game::client::buildings::DefensiveBuilding, 19
  - game::client::GameObject, 24
  - game::client::units::Archer, 6
  - game::client::units::Swordsman, 43
  - game::client::units::Unit, 46
- startGameServer
  - lobby::server::GameServerHandler, 27
- think
  - game::client::buildings::Building, 8
  - game::client::buildings::DefensiveBuilding, 19
  - game::client::buildings::ProductionBuilding, 40
  - game::client::GameObject, 25
  - game::client::units::Archer, 6
  - game::client::units::Swordsman, 43
  - game::client::units::Unit, 46
- trace
  - game::client::World, 49
- track
  - util::CPosition, 15
  - util::Position, 38
- util::CAxisAlignedRect, 10
  - CAxisAlignedRect, 10
  - isPointIn, 10
- util::CPosition, 14
  - combatPos, 15
  - copy, 15
  - track, 15
  - v, 15
  - x, 14
  - y, 14
- util::CVec2, 16
  - copy, 16
  - x, 16
  - y, 16
- util::Position, 37
  - combatPos, 38
  - Position, 37
  - track, 38
  - x, 38
  - y, 38
- util::Vec2, 47
  - Vec2, 47
  - x, 47
  - y, 47
- v
  - util::CPosition, 15
- Vec2
  - util::Vec2, 47
- x
  - util::CPosition, 14
  - util::CVec2, 16
  - util::Position, 38
  - util::Vec2, 47
- y
  - util::CPosition, 14
  - util::CVec2, 16
  - util::Position, 38
  - util::Vec2, 47