My Project

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

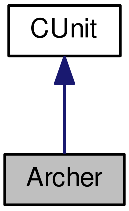
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

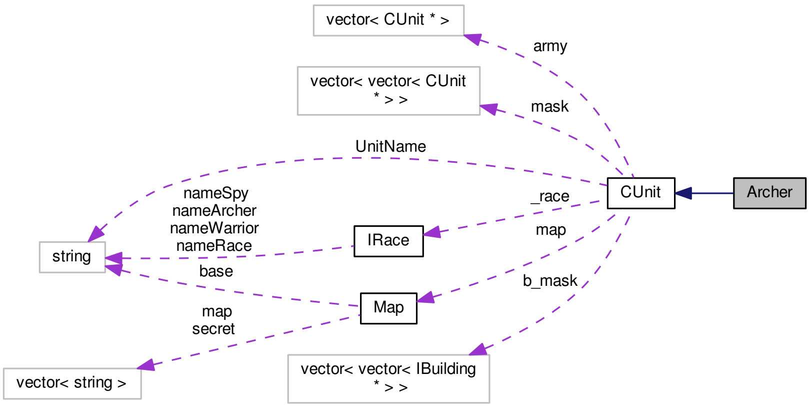
Class Documentation

3.1 Archer Class Reference

Inheritance diagram for Archer:



Collaboration diagram for Archer:



Public Member Functions

- **Archer** (IRace ∗race, Map ∗m, bool ie)
- void Move (int &mp) override
- void Damage (int &move, int cx, int cy) override

Additional Inherited Members

3.1.1 Member Function Documentation

3.1.1.1 void Archer::Damage (int & *move,* int *x,* int *y*) [inline], [override], [virtual]

hit the enemy

Parameters

move	way to enemy on (x,y)
x	
y	

Implements CUnit.

3.1.1.2 void Archer::Move (int & *mp*) [inline], [override], [virtual]

your unit's moving

Parameters

mp	movepoints

Implements CUnit.

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

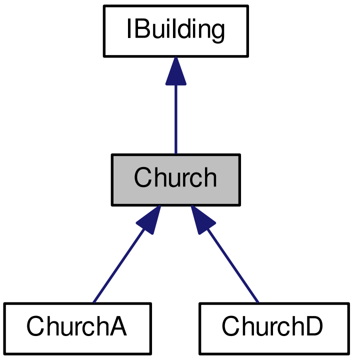
- Units.h

3.2 Church Class Reference

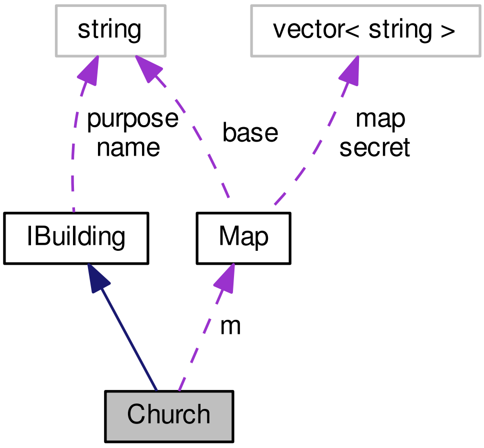
buffs attack or defence

```
#include <Buildings.h>
```

Inheritance diagram for Church:



Collaboration diagram for Church:



Public Member Functions

- **Church** (int x, int y, Map ∗m, int p)

Protected Attributes

- int **x**
- int **y**
- Map ∗ **m**

3.2.1 Detailed Description

buffs attack or defence

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

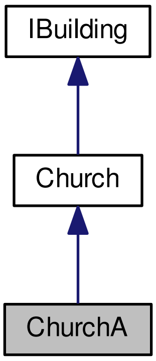
- Buildings.h

3.3 ChurchA Class Reference

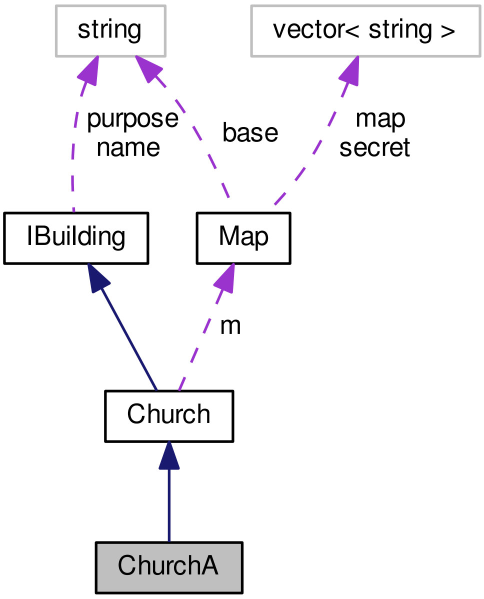
buffs attack

```
#include <Buildings.h>
```

Inheritance diagram for ChurchA:



Collaboration diagram for ChurchA:



Public Member Functions

- **ChurchA** (int x, int y, Map ∗m, int p)

Additional Inherited Members

3.3.1 Detailed Description

buffs attack

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

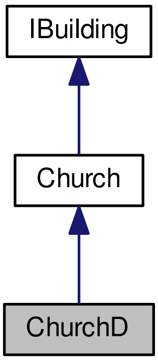
- Buildings.h

3.4 ChurchD Class Reference

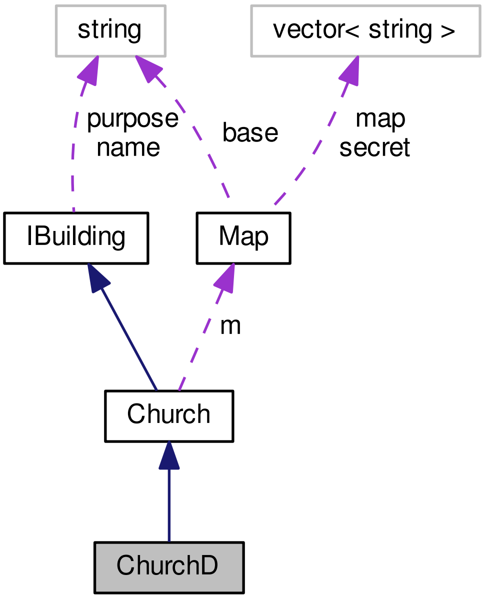
buffs defence

```
#include <Buildings.h>
```

Inheritance diagram for ChurchD:



Collaboration diagram for ChurchD:



Public Member Functions

- **ChurchD** (int x, int y, Map ∗m, int p)

Additional Inherited Members

3.4.1 Detailed Description

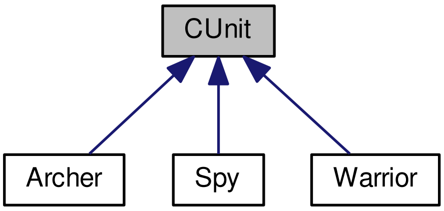
buffs defence

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

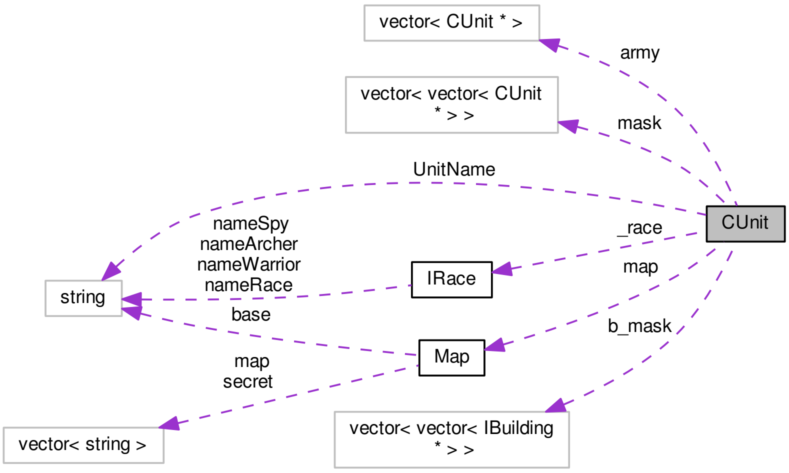
- Buildings.h

3.5 CUnit Class Reference

Inheritance diagram for CUnit:



Collaboration diagram for CUnit:



Public Member Functions

- **CUnit** (IRace ∗r, Map ∗map)
- virtual void Move (int &mp)=0
- virtual void Damage (int &move, int x, int y)=0
- void TakeDamage (int damage)
- void getInfo () const
- string getName () const
- void takeOrders ()
- void setPlace (int x, int y)
- char getSymbol () const
- int getX () const
- int getY () const
- void healing ()
- int getHealth () const
- int getMaxHealth () const
- void setArmy (vector< CUnit ∗ > ∗a)
- void setMask (vector< vector< CUnit ∗ >> ∗m)
- void setBMask (vector< vector< IBuilding ∗ >> ∗bm)
- int getAWL () const
- int getDWL () const

Protected Member Functions

- void Death ()
- void churchEffect (int tx, int ty)
- void findChurche (int x, int y)
- void look ()
- void unlook ()

Protected Attributes

- IRace ∗ **_race**
- vector< CUnit ∗ > ∗ **army**
- vector< vector< CUnit ∗ > > ∗ **mask**
- vector< vector< IBuilding ∗ > > ∗ **b_mask**
- string **UnitName**
- int **move**
- int **defence**
- int **attack**
- int **attackDistance** = 1
- int **lookDistance**
- int **health**
- int **maxhealth**
- int **attackWeaponLevel** = 0
- int **defenceWeaponLevel** = 0
- int **x**
- int **y**
- char **symbol**
- Map ∗ **map**
- bool **isEnemy**
- int **id**
- int **powerWeaponLevel** = 5

3.5.1 Member Function Documentation

3.5.1.1 void CUnit::churchEffect (int *tx*, int *ty*) [inline], [protected]

check (tx,ty) church's effect's

Parameters

tx	
ty	

3.5.1.2 virtual void CUnit::Damage (int & *move*, int *x*, int *y*) [pure virtual]

hit the enemy

Parameters

move	way to enemy on (x,y)
x	
y	

Implemented in Spy, Archer, and Warrior.

3.5.1.3 void CUnit::Death ( ) [inline], [protected]

Unit's death: deleting from maps

3.5.1.4 void CUnit::findChurche (int *x*, int *y*) [inline], [protected]

check is there a churche

Parameters

x	
y	

3.5.1.5 int CUnit::getAWL ( ) const [inline]

Returns

weapon's attack level

3.5.1.6 int CUnit::getDWL ( ) const [inline]

Returns

weapon's defence level

3.5.1.7 int CUnit::getHealth ( ) const [inline]

Returns

health of unit

3.5.1.8 void CUnit::getInfo ( ) const [inline]

say about the current unit

3.5.1.9 int CUnit::getMaxHealth ( ) const [inline]

Returns

maxhealth of unit

3.5.1.10 string CUnit::getName ( ) const [inline]

Returns

unit's name

3.5.1.11 char CUnit::getSymbol ( ) const [inline]

get unit's mark on text map

Returns

symbol

3.5.1.12 int CUnit::getX ( ) const [inline]

get x-coordinate

Returns

x

3.5.1.13 int CUnit::getY ( ) const [inline]

get y-coordinate

Returns

y

3.5.1.14 void CUnit::healing ( ) [inline]

heal unit if it possible

3.5.1.15 void CUnit::look ( ) [inline], [protected]

change '∗' to '.' in unit's area of visibility

3.5.1.16 virtual void CUnit::Move (int & *mp*) [pure virtual]

your unit's moving

Parameters

mp	movepoints

Implemented in Spy, Archer, and Warrior.

3.5.1.17 void CUnit::setArmy (vector< CUnit ∗ > ∗ *a*) [inline]

set unit's army

Parameters

a	

3.5.1.18 void CUnit::setBMask (vector< vector< IBuilding ∗ >> ∗ *bm*) [inline]

set map with buildings

Parameters

bm	

3.5.1.19 void CUnit::setMask (vector< vector< CUnit * >> * *m*) [inline]

set map with units

Parameters

m	

3.5.1.20 void CUnit::setPlace (int *x*, int *y*) [inline]

set place for unit

Parameters

x	
y	

3.5.1.21 void CUnit::TakeDamage (int *damage*) [inline]

count the damage to unit's health if it bigger than health, unit will die

Parameters

damage	damage that your unit get from hit

3.5.1.22 void CUnit::takeOrders ( ) [inline]

unit recieve orders from you

- move
- fight
- nothing

3.5.1.23 void CUnit::unlook ( ) [inline], [protected]

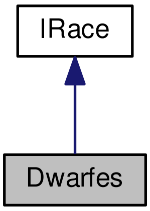
change '.' to '*' in unit's area of visibility when he moves

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

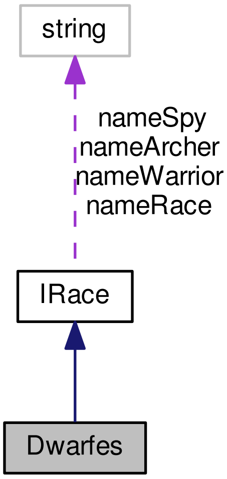
- Units.h

3.6 Dwarfes Class Reference

Inheritance diagram for Dwarfes:



Collaboration diagram for Dwarfes:



Public Member Functions

- void **superAction** () override

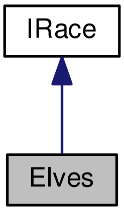
Additional Inherited Members

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

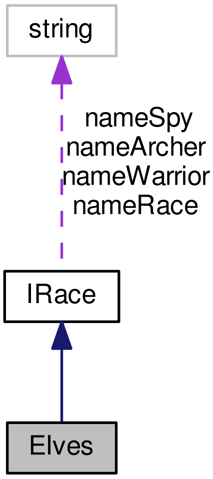
- Race.h

3.7 Elves Class Reference

Inheritance diagram for Elves:



Collaboration diagram for Elves:



Public Member Functions

- void **superAction** () override

Additional Inherited Members

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

- Race.h

3.8 Factory Class Reference

Unit's factory.

```
#include <Units.h>
```

Static Public Member Functions

- static CUnit ∗ **create** (string type, IRace ∗race, Map ∗m, bool ie)

3.8.1 Detailed Description

Unit's factory.

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

- Units.h

3.9 GameMap Class Reference

Singleton class. Makes gamemap.

```
#include <World.h>
```

Public Member Functions

- **GameMap** (const GameMap &)=delete
- GameMap & **operator=** (const GameMap &)=delete

Static Public Member Functions

- static Map ∗ **instance** ()

3.9.1 Detailed Description

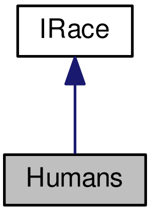
Singleton class. Makes gamemap.

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

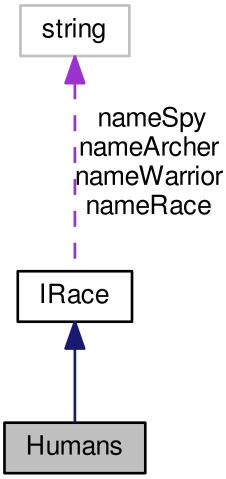
- World.h

3.10 Humans Class Reference

Inheritance diagram for Humans:



Collaboration diagram for Humans:



Public Member Functions

- void **superAction** () override

Additional Inherited Members

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

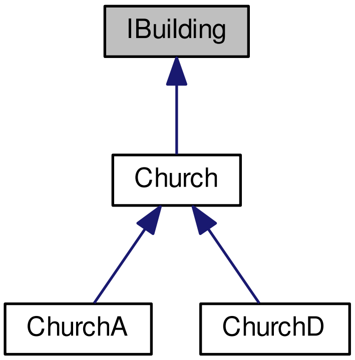
- Race.h

3.11 IBuilding Class Reference

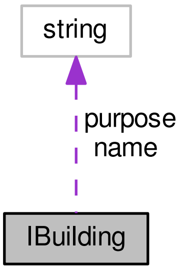
building's interface

```
#include <Buildings.h>
```

Inheritance diagram for IBuilding:



Collaboration diagram for IBuilding:



Public Member Functions

- string getPurpose ()
- int getPower ()

Protected Attributes

- int **high**
- int **wall**
- string **name**
- string **purpose**
- int **power**

3.11.1 Detailed Description

building's interface

3.11.2 Member Function Documentation

3.11.2.1 int IBuilding::getPower ( ) `[inline]`

Returns

purpose(bonus) that unit can take staying near the building

3.11.2.2 string IBuilding::getPurpose ( ) `[inline]`

Returns

goal of building, defence or attack

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

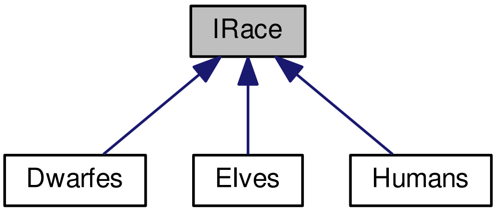
- Buildings.h

3.12 IRace Class Reference

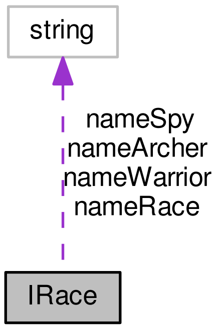
Interface of races.

`#include <Race.h>`

Inheritance diagram for IRace:



Collaboration diagram for IRace:



Public Member Functions

- string **getNameRace** () const
- string **getNameArcher** () const
- string **getNameWarrior** () const
- string **getNameSpy** () const
- int **getBonusMove** () const
- int **getBonusAttack** () const
- int **getBonusDefence** () const
- int **getBonusArcherDistance** () const
- int **getBonusLookDistance** () const
- int **getBonusHealth** () const
- virtual void **superAction** ()=0

Protected Attributes

- string **nameRace**
- string **nameArcher**
- string **nameWarrior**
- string **nameSpy**
- int **bonusMove**
- int **bonusAttack**
- int **bonusDefence**
- int **bonusArcherDistance**
- int **bonusLookDistance**
- int **bonusHealth**

3.12.1 Detailed Description

Interface of races.

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

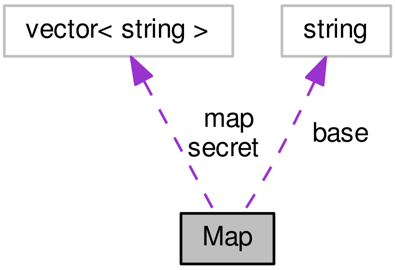
- Race.h

3.13 Map Class Reference

Map text image.

```
#include <Map.h>
```

Collaboration diagram for Map:



Public Member Functions

- Map ()
- void showMap () const
- int getSize () const

Public Attributes

- string **base**
- vector< string > **map**
- vector< string > **secret**

3.13.1 Detailed Description

Map text image.

Map is showing like number of symbols.

- $*$ - unknown place
- **.** - place in your area of visibility
- **A - Z** - your units
- **a - z** - enemy units
- **+** - church

3.13.2 Constructor & Destructor Documentation

3.13.2.1 Map::Map ( ) [inline]

initialize map with '*'

3.13.3 Member Function Documentation

3.13.3.1 int Map::getSize ( ) const [inline]

Returns

map's size

3.13.3.2 void Map::showMap ( ) const [inline]

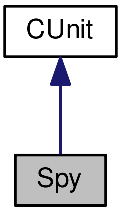
show the map

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

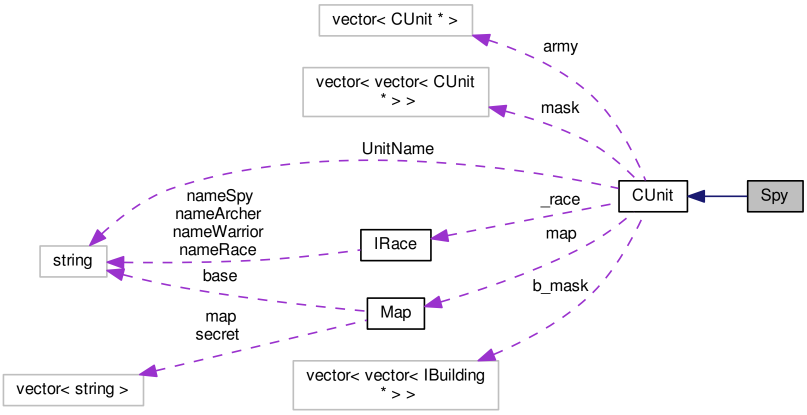
- Map.h

3.14 Spy Class Reference

Inheritance diagram for Spy:



Collaboration diagram for Spy:



Public Member Functions

- **Spy** (IRace ∗race, Map ∗m, bool ie)
- void Move (int &mp) override
- void Damage (int &move, int cx, int cy) override

Additional Inherited Members

3.14.1 Member Function Documentation

3.14.1.1 void Spy::Damage (int & *move,* int *x,* int *y*) [inline], [override], [virtual]

hit the enemy

Parameters

move	way to enemy on (x,y)
x	
y	

Implements CUnit.

3.14.1.2 void Spy::Move (int & *mp*) [inline], [override], [virtual]

your unit's moving

Parameters

mp	movepoints

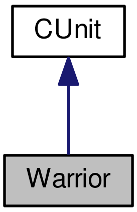
Implements CUnit.

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

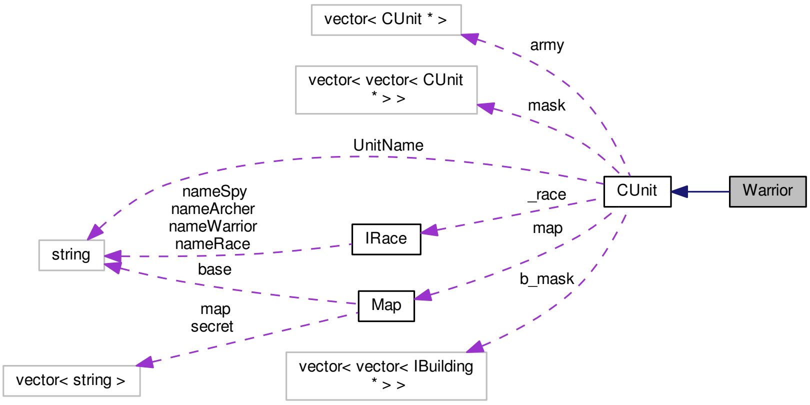
- Units.h

3.15 Warrior Class Reference

Inheritance diagram for Warrior:



Collaboration diagram for Warrior:



Public Member Functions

- **Warrior** (IRace ∗race, Map ∗m, bool ie)
- void Move (int &mp) override
- void Damage (int &move, int cx, int cy) override

Additional Inherited Members

3.15.1 Member Function Documentation

3.15.1.1 void Warrior::Damage (int & *move,* int *x,* int *y*) `[inline], [override], [virtual]`

hit the enemy

Parameters

move	way to enemy on (x,y)
x	
y	

Implements CUnit.

3.15.1.2 void Warrior::Move (int & *mp*) `[inline], [override], [virtual]`

your unit's moving

Parameters

mp	movepoints

Implements CUnit.

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

- Units.h

3.16 World Class Reference

Makes the gameworld. Initializes maps with objects.

```
#include <World.h>
```

Public Member Functions

- World ()
- CUnit ∗ getMyUnit (int n) const
- void healing () const
- bool war ()

3.16.1 Detailed Description

Makes the gameworld. Initializes maps with objects.

3.16.2 Constructor & Destructor Documentation

3.16.2.1 World::World ( ) [inline]

Initializing

3.16.3 Member Function Documentation

3.16.3.1 CUnit∗ World::getMyUnit (int *n*) const [inline]

Parameters

n	unit's number

Returns

unit

3.16.3.2 void World::healing ( ) const [inline]

Heal alive units

3.16.3.3 bool World::war ( ) [inline]

One step of the game. Every alive unit takes orders. If there nobody alive, game ends.

Returns

is the game ended

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

- World.h

3.17 Writer Class Reference

this class is for speaking with player

`#include <Units.h>`

Static Public Member Functions

- static void wChoise (int movepoints)
- static void wTakeCoordinates ()
- static void wWrongInfo ()
- static void wDeath (string &UnitName)
- static void **wDamage** (string &UnitName, int damage)
- static void wHealing (string &UnitName)
- static void wMoveInfoWarriorArcher ()
- static void wMistake ()
- static void wMoveInfoSpy ()
- static void wLine ()

3.17.1 Detailed Description

this class is for speaking with player

3.17.2 Member Function Documentation

3.17.2.1 static void Writer::wChoise (int *movepoints*) [inline], [static]

ask you to order something

Parameters

movepoints	current unit movepoints

3.17.2.2 static void Writer::wDeath (string & *UnitName*) [inline], [static]

say that this unit has dead

Parameters

UnitName	unit's name

3.17.2.3 static void Writer::wHealing (string & *UnitName*) [inline], [static]

say that your unit is healing

Parameters

UnitName	

3.17.2.4 static void Writer::wLine ( ) [inline], [static]

print 2 blank lines

3.17.2.5 static void Writer::wMistake ( ) [inline], [static]

say about an error

3.17.2.6 static void Writer::wMoveInfoSpy ( ) [inline], [static]

ask you to order the way

3.17.2.7 static void Writer::wMoveInfoWarriorArcher ( ) [inline], [static]

ask you to order the way

3.17.2.8 static void Writer::wTakeCoordinates ( ) [inline], [static]

ask you coordinates

3.17.2.9 static void Writer::wWrongInfo ( ) [inline], [static]

say about error

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

- Units.h

Index