

BigLittleWargame

Generated by Doxygen 1.8.10

Thu Nov 5 2015 15:43:54

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	Bomber Class Reference	5
3.1.1	Constructor & Destructor Documentation	5
3.1.1.1	Bomber(textureID, const textureHolder &textures, sf::Vector2f, sf::Color)	5
3.2	Building Class Reference	5
3.2.1	Constructor & Destructor Documentation	6
3.2.1.1	Building(textureID, const textureHolder &textures, sf::Vector2f, sf::Color)	6
3.2.2	Member Function Documentation	6
3.2.2.1	checkAction(CommandQueue &command)	6
3.3	Button Class Reference	6
3.3.1	Constructor & Destructor Documentation	7
3.3.1.1	Button(textureID, const textureHolder &textures, sf::Vector2f)	7
3.3.2	Member Function Documentation	7
3.3.2.1	getClicked(sf::Vector2f)	7
3.4	Command Struct Reference	7
3.5	CommandQueue Class Reference	7
3.5.1	Member Function Documentation	8
3.5.1.1	isEmpty() const	8
3.5.1.2	pop()	8
3.5.1.3	push(const Command &command)	8
3.6	DragonButton Class Reference	8
3.6.1	Constructor & Destructor Documentation	8
3.6.1.1	DragonButton(textureID, const textureHolder &textures, sf::Vector2f pos)	8
3.6.2	Member Function Documentation	9
3.6.2.1	bAction(const textureHolder &textures, sf::Vector2f pos, sf::Color color) override	9
3.7	endOfFile Class Reference	9

3.8	EndTurnButton Class Reference	9
3.8.1	Constructor & Destructor Documentation	10
3.8.1.1	EndTurnButton(textureID, const textureHolder &textures, sf::Vector2f)	10
3.8.2	Member Function Documentation	10
3.8.2.1	handleClick(CommandQueue &queue) override	10
3.9	Factory Class Reference	10
3.9.1	Constructor & Destructor Documentation	11
3.9.1.1	Factory(textureID, const textureHolder &textures, sf::Vector2f, sf::Color)	11
3.9.2	Member Function Documentation	12
3.9.2.1	checkAction(CommandQueue &command) override	12
3.10	Game Class Reference	12
3.10.1	Member Function Documentation	12
3.10.1.1	getExit()	12
3.11	GameObject Class Reference	12
3.11.1	Constructor & Destructor Documentation	13
3.11.1.1	GameObject(textureID, const textureHolder &textures, sf::Vector2f pos)	13
3.11.2	Member Function Documentation	13
3.11.2.1	draw(sf::RenderWindow &window)	13
3.11.2.2	getPosition()	14
3.11.2.3	getTilePosition()	14
3.11.2.4	handleMouse(sf::Vector2f pos)	14
3.11.2.5	setPosition(sf::Vector2f pos)	14
3.11.2.6	setTextRect(sf::IntRect)	14
3.11.2.7	update(sf::Time)	14
3.12	Headquarters Class Reference	15
3.12.1	Constructor & Destructor Documentation	15
3.12.1.1	Headquarters(textureID, const textureHolder &textures, sf::Vector2f, sf::Color)	15
3.12.2	Member Function Documentation	15
3.12.2.1	checkAction(CommandQueue &command) override	15
3.13	HolyGrail Class Reference	16
3.13.1	Constructor & Destructor Documentation	16
3.13.1.1	HolyGrail(textureID, const textureHolder &textures, sf::Vector2f)	16
3.13.2	Member Function Documentation	16
3.13.2.1	Reckoning(CommandQueue &queue, sf::Color)	16
3.14	InputHandler Class Reference	16
3.14.1	Constructor & Destructor Documentation	17
3.14.1.1	InputHandler(sf::RenderWindow &window)	17
3.14.2	Member Function Documentation	18
3.14.2.1	processInput(CommandQueue &queue)	18
3.15	MenuButton Class Reference	18

3.15.1	Constructor & Destructor Documentation	18
3.15.1.1	MenuButton(textureID, const textureHolder &textures, sf::Vector2f pos=sf::Vector2f(0, 0))	18
3.15.2	Member Function Documentation	19
3.15.2.1	handleMouse(sf::Vector2f pos, sf::RenderWindow &>window, std::vector< std::unique_ptr< MenuButton >> &container, Music &music)	19
3.15.2.2	update(sf::Time) override	19
3.16	Music Class Reference	19
3.16.1	Member Function Documentation	20
3.16.1.1	getPaused()	20
3.16.1.2	play(musicID id)	20
3.16.1.3	setVolume(float volume)	20
3.17	Player Class Reference	20
3.17.1	Constructor & Destructor Documentation	21
3.17.1.1	Player(sf::Color, bool)	21
3.17.2	Member Function Documentation	21
3.17.2.1	diffUnitAttacks(int attacks)	21
3.17.2.2	diffUnitWalks(int steps)	21
3.17.2.3	getActive()	21
3.17.2.4	getMoney()	22
3.17.2.5	getPlayer()	22
3.17.2.6	getPoints()	22
3.17.2.7	getUnitAttacks()	22
3.17.2.8	getUnitWalks()	22
3.17.2.9	setActive(bool)	22
3.17.2.10	setMoney(int mon)	22
3.17.2.11	setUnitAttacks(int attacks)	23
3.17.2.12	subtractPoints(CommandQueue &queue)	23
3.18	PlayerButton Class Reference	23
3.18.1	Constructor & Destructor Documentation	23
3.18.1.1	PlayerButton(textureID, const textureHolder &textures, sf::Vector2f)	23
3.18.2	Member Function Documentation	24
3.18.2.1	handleClick(CommandQueue &queue)	24
3.19	PlayerObject Class Reference	24
3.19.1	Constructor & Destructor Documentation	25
3.19.1.1	PlayerObject(textureID, const textureHolder &textures, sf::Vector2f, sf::Color)	25
3.19.2	Member Function Documentation	25
3.19.2.1	checkClicked(sf::Vector2f pos)	25
3.19.2.2	damage(int points)	25
3.19.2.3	getHP()	25

3.19.2.4	getSelected()	25
3.19.2.5	makeSelected(sf::Vector2f pos)	26
3.19.2.6	setOldSelected(bool)	27
3.19.2.7	setSelected(bool)	27
3.20	ReadText Class Reference	27
3.20.1	Member Function Documentation	27
3.20.1.1	read(std::ifstream &input)	27
3.21	Recruit Class Reference	28
3.21.1	Constructor & Destructor Documentation	28
3.21.1.1	Recruit(textureID, const textureHolder &textures, sf::Vector2f, sf::Color)	28
3.22	RecruitButton Class Reference	28
3.22.1	Constructor & Destructor Documentation	29
3.22.1.1	RecruitButton(textureID, const textureHolder &textures, sf::Vector2f pos)	29
3.22.2	Member Function Documentation	29
3.22.2.1	bAction(const textureHolder &textures, sf::Vector2f pos, sf::Color color) override	29
3.23	Resource Class Reference	29
3.23.1	Constructor & Destructor Documentation	30
3.23.1.1	Resource(textureID, const textureHolder &textures, sf::Vector2f)	30
3.23.2	Member Function Documentation	30
3.23.2.1	checkClicked(sf::Vector2f pos)	30
3.23.2.2	getMoney()	30
3.23.2.3	getResourceMoney()	31
3.23.2.4	getSelected()	31
3.23.2.5	makeSelected(sf::Vector2f pos)	31
3.23.2.6	setOldSelected(bool)	31
3.23.2.7	setSelected(bool)	31
3.24	ResourceHolder< Resource, Identifier > Class Template Reference	31
3.25	Scout Class Reference	32
3.25.1	Constructor & Destructor Documentation	32
3.25.1.1	Scout(textureID, const textureHolder &textuvers, sf::Vector2f, sf::Color)	32
3.26	ScoutButton Class Reference	32
3.26.1	Constructor & Destructor Documentation	33
3.26.1.1	ScoutButton(textureID, const textureHolder &textures, sf::Vector2f pos)	33
3.26.2	Member Function Documentation	33
3.26.2.1	bAction(const textureHolder &textures, sf::Vector2f pos, sf::Color color) override	33
3.27	Selectable Class Reference	33
3.28	Soldier Class Reference	34
3.28.1	Constructor & Destructor Documentation	34
3.28.1.1	Soldier(textureID id, const textureHolder &textures, sf::Vector2f pos, sf::Color color)	34
3.29	SoldierButton Class Reference	34

3.29.1	Constructor & Destructor Documentation	35
3.29.1.1	SoldierButton(textureID, const textureHolder &textures, sf::Vector2f pos)	35
3.29.2	Member Function Documentation	35
3.29.2.1	bAction(const textureHolder &textures, sf::Vector2f pos, sf::Color color) override	35
3.30	Sound Class Reference	35
3.30.1	Member Function Documentation	36
3.30.1.1	play(soundID)	36
3.31	Terrain Class Reference	36
3.31.1	Constructor & Destructor Documentation	36
3.31.1.1	Terrain(textureID, const textureHolder &textures, sf::Vector2f)	36
3.31.2	Member Function Documentation	37
3.31.2.1	changeColor(sf::Color col)	37
3.31.2.2	checkClicked(sf::Vector2f pos)	37
3.31.2.3	getColor()	37
3.31.2.4	getFree()	37
3.31.2.5	setFree(bool f)	37
3.32	Unit Class Reference	38
3.32.1	Constructor & Destructor Documentation	38
3.32.1.1	Unit(textureID, const textureHolder &textures, sf::Vector2f, sf::Color)	38
3.32.2	Member Function Documentation	39
3.32.2.1	attack()	39
3.32.2.2	getAttackrange()	39
3.32.2.3	getDP()	39
3.32.2.4	getName()	39
3.32.2.5	getWalklimit()	39
3.32.2.6	setPosition(sf::Vector2f pos) override	39
3.32.2.7	walk(sf::Vector2f pos)	40
3.33	UnitButton Class Reference	40
3.33.1	Constructor & Destructor Documentation	40
3.33.1.1	UnitButton(textureID, const textureHolder &textures, sf::Vector2f pos)	40
3.33.2	Member Function Documentation	41
3.33.2.1	bAction(const textureHolder &textures, sf::Vector2f pos, sf::Color color)	41
3.33.2.2	getCostMoney()	41
3.34	unknownCharacter Class Reference	41
3.35	WonButton Class Reference	41
3.35.1	Constructor & Destructor Documentation	42
3.35.1.1	WonButton(textureID, const textureHolder &textures, sf::Vector2f)	42
3.35.2	Member Function Documentation	42
3.35.2.1	handleClick(CommandQueue &queue) override	42

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Command	7
CommandQueue	7
exception	
endOfFile	9
unknownCharacter	41
Game	12
GameObject	12
Button	6
MenuButton	18
PlayerButton	23
EndTurnButton	9
WonButton	41
UnitButton	40
DragonButton	8
RecruitButton	28
ScoutButton	32
SoldierButton	34
HolyGrail	16
PlayerObject	24
Building	5
Factory	10
Headquarters	15
Unit	38
Bomber	5
Recruit	28
Scout	32
Soldier	34
Soldier	34
Resource	29
Selectable	33
Terrain	36
InputHandler	16
NonCopyable	
Music	19
Sound	35
Player	20
ReadText	27

ResourceHolder< Resource, Identifier >	31
ResourceHolder< sf::SoundBuffer, soundID >	31
ResourceHolder< sf::Texture, textureID >	31

Chapter 2

Class Index

2.1 Class List

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

Bomber	5
Building	5
Button	6
Command	7
CommandQueue	7
DragonButton	8
endOfFile	9
EndTurnButton	9
Factory	10
Game	12
GameObject	12
Headquarters	15
HolyGrail	16
InputHandler	16
MenuButton	18
Music	19
Player	20
PlayerButton	23
PlayerObject	24
ReadText	27
Recruit	28
RecruitButton	28
Resource	29
ResourceHolder< Resource, Identifier >	31
Scout	32
ScoutButton	32
Selectable	33
Soldier	34
SoldierButton	34
Sound	35
Terrain	36
Unit	38
UnitButton	40
unknownCharacter	41
WonButton	41

Chapter 3

Class Documentation

3.1 Bomber Class Reference

Inheritance diagram for Bomber:

Public Member Functions

- **Bomber** (textureID, const **textureHolder** &textures, sf::Vector2f, sf::Color)
Constructor.

Additional Inherited Members

3.1.1 Constructor & Destructor Documentation

3.1.1.1 Bomber::Bomber (textureID *id*, const **textureHolder** & *textures*, sf::Vector2f *pos*, sf::Color *color*)

Constructor.

Parameters

<i>id</i>	The identifier of the Bomber (p. 5).
<i>textures</i>	The textures of the Bomber (p. 5).
<i>pos</i>	The position of the Bomber (p. 5) on the screen.
<i>color</i>	The color of the Bomber (p. 5) (tells which player is the owner).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/SubUnits.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Bomber.cpp
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/SubUnits.cpp

3.2 Building Class Reference

Inheritance diagram for Building:

Public Member Functions

- virtual void **checkAction** (**CommandQueue** &command)
Check action.

Protected Member Functions

- **Building** (textureID, const **textureHolder** &textures, sf::Vector2f pos, sf::Color)
Constructor.

Additional Inherited Members

3.2.1 Constructor & Destructor Documentation

3.2.1.1 **Building::Building** (textureID *id*, const **textureHolder** & *textures*, sf::Vector2f *pos*, sf::Color *color*)
[protected]

Constructor.

Parameters

<i>id</i>	The identifier of the Building (p. 5).
<i>textures</i>	The textures of the Building (p. 5).
<i>pos</i>	The position of the Building (p. 5) on the field.
<i>color</i>	The color of the Building (p. 5), tells which player is the owner of it.

3.2.2 Member Function Documentation

3.2.2.1 void **Building::checkAction** (**CommandQueue** & *queue*) [virtual]

Check action.

Parameters

<i>in, out</i>	<i>queue</i>	The commandqueue which contains all the commands.
----------------	--------------	---

Reimplemented in **Factory** (p. 12), and **Headquarters** (p. 15).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Building.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Building.cpp

3.3 Button Class Reference

Inheritance diagram for Button:

Public Member Functions

- **Button** (textureID, const **textureHolder** &textures, sf::Vector2f)
Constructor.
- bool **getClicked** (sf::Vector2f)
Gets a clicked.

Additional Inherited Members

3.3.1 Constructor & Destructor Documentation

3.3.1.1 Button::Button (textureID *id*, const textureHolder & *textures*, sf::Vector2f *pos*)

Constructor.

Parameters

<i>id</i>	The identifier of the Button (p. 6).
<i>textures</i>	The textures of the Button (p. 6).
<i>pos</i>	The position of the Button (p. 6) on the screen.

3.3.2 Member Function Documentation

3.3.2.1 bool Button::getClicked (sf::Vector2f *pos*)

Gets a clicked.

Parameters

<i>pos</i>	The position where is clicked on the screen.
------------	--

Returns

true if **Button** (p. 6) is clicked, false if not.

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Button.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Button.cpp

3.4 Command Struct Reference

Public Attributes

- commandID **id**
- sf::Vector2f **pos**

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Command.h

3.5 CommandQueue Class Reference

Public Member Functions

- void **push** (const **Command** &command)
Pushes an object onto this stack.
- **Command pop** ()
Removes and returns the top-of-stack object.
- bool **isEmpty** () const
Query if this object is empty.

3.5.1 Member Function Documentation

3.5.1.1 `bool CommandQueue::isEmpty () const`

Query if this object is empty.

Returns

true if empty, false if not.

3.5.1.2 `Command CommandQueue::pop ()`

Removes and returns the top-of-stack object.

Returns

The previous top-of-stack object.

3.5.1.3 `void CommandQueue::push (const Command & command)`

Pushes an object onto this stack.

Parameters

<i>command</i>	The command to push.
----------------	----------------------

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/CommandQueue.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/CommandQueue.↵
cpp

3.6 DragonButton Class Reference

Inheritance diagram for DragonButton:

Public Member Functions

- **DragonButton** (textureID, const **textureHolder** &textures, sf::Vector2f pos)
Constructor.
- **Unit * bAction** (const **textureHolder** &textures, sf::Vector2f pos, sf::Color color) override
Actions.

Additional Inherited Members

3.6.1 Constructor & Destructor Documentation

3.6.1.1 `DragonButton::DragonButton (textureID id, const textureHolder & textures, sf::Vector2f pos)`

Constructor.

Parameters

<i>id</i>	The identifier of the DragonButton (p. 8).
<i>textures</i>	The textures of the DragonButton (p. 8).
<i>pos</i>	The position of the DragonButton (p. 8) on the screen.

3.6.2 Member Function Documentation

3.6.2.1 **Unit * DragonButton::bAction (const textureHolder & textures, sf::Vector2f pos, sf::Color color)**
 [override], [virtual]

Actions.

Parameters

<i>textures</i>	The textures.
<i>pos</i>	The position.
<i>color</i>	The color.

Returns

null if it fails, else an Unit*.

Reimplemented from **UnitButton** (p. 41).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/UnitButton.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/UnitButton.cpp

3.7 endOfFile Class Reference

Inheritance diagram for endOfFile:

Public Member Functions

- const char * **what** () const override

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Exceptions.h

3.8 EndTurnButton Class Reference

Inheritance diagram for EndTurnButton:

Public Member Functions

- **EndTurnButton** (textureID, const **textureHolder** &textures, sf::Vector2f)

Constructor.

- void **handleClick** (**CommandQueue** &queue) override

Handles the mouse click.

Additional Inherited Members

3.8.1 Constructor & Destructor Documentation

3.8.1.1 EndTurnButton::EndTurnButton (textureID id, const textureHolder & textures, sf::Vector2f pos)

Constructor.

Parameters

<i>id</i>	The identifier of the EndTurnButton (p. 9).
<i>textures</i>	The textures of the EndTurnButton (p. 9).
<i>pos</i>	The position of the EndTurnButton (p. 9) on the screen.

3.8.2 Member Function Documentation

3.8.2.1 void EndTurnButton::handleClick (**CommandQueue** & queue) [override],[virtual]

Handles the mouse click.

Parameters

<i>in, out</i>	<i>queue</i>	The queue.
----------------	--------------	------------

Reimplemented from **PlayerButton** (p. 24).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/PlayerButton.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/PlayerButton.cpp

3.9 Factory Class Reference

Inheritance diagram for Factory:

Public Member Functions

- **Factory** (textureID, const **textureHolder** &textures, sf::Vector2f, sf::Color)

Constructor.

- void **checkAction** (**CommandQueue** &command) override

Check action.

Additional Inherited Members

3.9.1 Constructor & Destructor Documentation

3.9.1.1 `Factory::Factory (textureID id, const textureHolder & textures, sf::Vector2f pos, sf::Color color)`

Constructor.

Parameters

<i>id</i>	The identifier of the Factory (p. 10).
<i>textures</i>	The textures of the Factory (p. 10).
<i>pos</i>	The position of the Factory (p. 10) on the field.
<i>color</i>	The color of the Factory (p. 10), tells which player is the owner of it.

3.9.2 Member Function Documentation

3.9.2.1 void Factory::checkAction (CommandQueue & queue) [override],[virtual]

Check action.

Parameters

<i>in, out</i>	<i>queue</i>	The commandqueue which contains all the commands.
----------------	--------------	---

Reimplemented from **Building** (p. 6).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Building.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Building.cpp

3.10 Game Class Reference

Public Member Functions

- **Game** ()
Default constructor.
- void **run** ()
Runs this object.
- bool **getExit** ()
Gets the exit.

3.10.1 Member Function Documentation

3.10.1.1 bool Game::getExit ()

Gets the exit.

Returns

true if exit event is true, false if not.

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Game.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Game.cpp

3.11 GameObject Class Reference

Inheritance diagram for GameObject:

Public Member Functions

- void **draw** (sf::RenderWindow &window)
Draws the object sprite on the given window.
- sf::Vector2f **getPosition** ()
*Gets the position of the **GameObject** (p. 12).*
- sf::Vector2f **getTilePosition** ()
*Gets tile position of the **GameObject** (p. 12).*
- virtual void **update** (sf::Time)
Updates the sprite so an animation is possible.
- virtual void **setPosition** (sf::Vector2f pos)
*Sets the position of the **GameObject** (p. 12) on the screen.*
- virtual void **handleMouse** (sf::Vector2f pos)
Handles the mouse events at a given position of the mouse.

Protected Member Functions

- **GameObject** (textureID, const **textureHolder** &textures, sf::Vector2f pos)
Constructor.
- void **setTextRect** (sf::IntRect)
Sets text rectangle for the sprite.

Protected Attributes

- textureID **id**
- sf::Sprite **sprite**
- std::size_t **numFrames**
- std::size_t **currentFrame** = 0
- sf::Vector2i **frameSize** = sf::Vector2i(TILESIZE, TILESIZE)
- sf::Time **duration** = sf::seconds(1)
- sf::Time **elapsedTime** = sf::Time::Zero
- bool **repeat**

3.11.1 Constructor & Destructor Documentation**3.11.1.1 `GameObject::GameObject (textureID id, const textureHolder & textures, sf::Vector2f pos)` [protected]**

Constructor.

Parameters

<i>id</i>	The identifier of the GameObject (p. 12).
<i>textures</i>	The textures of the GameObject (p. 12).
<i>pos</i>	The position of the GameObject (p. 12) on the screen.

3.11.2 Member Function Documentation**3.11.2.1 `void GameObject::draw (sf::RenderWindow & window)`**

Draws the object sprite on the given window.

Parameters

<i>in, out</i>	<i>window</i>	The window of the game.
----------------	---------------	-------------------------

3.11.2.2 `sf::Vector2f` `GameObject::getPosition ()`

Gets the position of the **GameObject** (p. 12).

Returns

The position on the screen.

3.11.2.3 `sf::Vector2f` `GameObject::getTilePosition ()`

Gets tile position of the **GameObject** (p. 12).

Returns

The tile position.

3.11.2.4 `void` `GameObject::handleMouse (sf::Vector2f pos)` [virtual]

Handles the mouse events at a given position of the mouse.

Parameters

<i>pos</i>	The position of the mouse.
------------	----------------------------

3.11.2.5 `void` `GameObject::setPosition (sf::Vector2f pos)` [virtual]

Sets the position of the **GameObject** (p. 12) on the screen.

Parameters

<i>pos</i>	The position on the screen.
------------	-----------------------------

Reimplemented in **Unit** (p. 39).

3.11.2.6 `void` `GameObject::setTextRect (sf::IntRect rect)` [protected]

Sets text rectangle for the sprite.

Parameters

<i>rect</i>	The rectangle of the sprite.
-------------	------------------------------

3.11.2.7 `void` `GameObject::update (sf::Time dt)` [virtual]

Updates the sprite so an animation is possible.

Parameters

<i>dt</i>	The delta time.
-----------	-----------------

Reimplemented in **MenuButton** (p. 19).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/GameObject.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/GameObject.cpp

3.12 Headquarters Class Reference

Inheritance diagram for Headquarters:

Public Member Functions

- **Headquarters** (textureID, const **textureHolder** &textures, sf::Vector2f, sf::Color)
Constructor.
- void **checkAction** (**CommandQueue** &command) override
Check action.

Additional Inherited Members

3.12.1 Constructor & Destructor Documentation

3.12.1.1 Headquarters::Headquarters (textureID *id*, const textureHolder & *textures*, sf::Vector2f *pos*, sf::Color *color*)

Constructor.

Parameters

<i>id</i>	The identifier of the HQ.
<i>textures</i>	The textures for the HQ.
<i>pos</i>	The position of the HQ on the field.
<i>color</i>	The color of the HQ, tells which player is the owner of it.

3.12.2 Member Function Documentation

3.12.2.1 void Headquarters::checkAction (**CommandQueue** & *queue*) [override],[virtual]

Check action.

Parameters

<i>in, out</i>	<i>queue</i>	The commandqueue which contains all the commands.
----------------	--------------	---

Reimplemented from **Building** (p. 6).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Building.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Building.cpp

3.13 HolyGrail Class Reference

Inheritance diagram for HolyGrail:

Public Member Functions

- **HolyGrail** (textureID, const **textureHolder** &textures, sf::Vector2f)
Constructor.
- void **Reckoning** (**CommandQueue** &queue, sf::Color)
Deals damage to the other team if units of a team are nearby.

Additional Inherited Members

3.13.1 Constructor & Destructor Documentation

3.13.1.1 HolyGrail::HolyGrail (textureID *id*, const textureHolder & *textures*, sf::Vector2f *pos*)

Constructor.

Parameters

<i>id</i>	The identifier of the Holy Grail.
<i>textures</i>	The textures of the Holy Grail.
<i>pos</i>	The position of the Holy Grail on the screen.

3.13.2 Member Function Documentation

3.13.2.1 void HolyGrail::Reckoning (CommandQueue & *queue*, sf::Color *color*)

Deals damage to the other team if units of a team are nearby.

Parameters

<i>in, out</i>	<i>queue</i>	The queue.
	<i>color</i>	The color of the team that's nearby.

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/HolyGrail.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/HolyGrail.cpp

3.14 InputHandler Class Reference

Public Member Functions

- **InputHandler** (sf::RenderWindow &window)
Constructor.
- void **processInput** (**CommandQueue** &queue)
Process the input described by queue.

3.14.1 Constructor & Destructor Documentation

3.14.1.1 InputHandler::InputHandler (sf::RenderWindow & *window*)

Constructor.

Parameters

<code>in, out</code>	<code>window</code>	The window of the game.
----------------------	---------------------	-------------------------

3.14.2 Member Function Documentation

3.14.2.1 void InputHandler::processInput (CommandQueue & queue)

Process the input described by queue.

Parameters

<code>in, out</code>	<code>queue</code>	The queue.
----------------------	--------------------	------------

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/InputHandler.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/InputHandler.cpp

3.15 MenuButton Class Reference

Inheritance diagram for MenuButton:

Public Member Functions

- **MenuButton** (textureID, const **textureHolder** &textures, sf::Vector2f pos=sf::Vector2f(0, 0))
Constructor.
- void **update** (sf::Time) override
Makes animation of the sprites possible.
- int **handleMouse** (sf::Vector2f pos, sf::RenderWindow &window, std::vector< std::unique_ptr< **MenuButton** >> &container, **Music** &music)
Handles the mouse events.

Public Attributes

- bool **LoadedInScreen** = true

Additional Inherited Members

3.15.1 Constructor & Destructor Documentation

3.15.1.1 MenuButton::MenuButton (textureID id, const textureHolder & textures, sf::Vector2f pos = sf::Vector2f(0, 0))

Constructor.

Parameters

<i>id</i>	The identifier of the MenuButton (p. 18).
<i>textures</i>	The textures of the MenuButton (p. 18).
<i>pos</i>	The position of the MenuButton (p. 18) on the screen.

3.15.2 Member Function Documentation

3.15.2.1 `int MenuButton::handleMouse (sf::Vector2f pos, sf::RenderWindow & window, std::vector< std::unique_ptr< MenuButton >> & container, Music & music)`

Handles the mouse events.

Parameters

	<i>pos</i>	The position of the mouse.
<i>in, out</i>	<i>window</i>	The window of the game.
<i>in, out</i>	<i>container</i>	The container of the mousebuttons.
<i>in, out</i>	<i>music</i>	The music.

Returns

An int.

3.15.2.2 `void MenuButton::update (sf::Time dt) [override],[virtual]`

Makes animation of the sprites possible.

Parameters

<i>dt</i>	The delta time.
-----------	-----------------

Reimplemented from **GameObject** (p. 14).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/MenuButton.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/MenuButton.cpp

3.16 Music Class Reference

Inheritance diagram for Music:

Public Member Functions

- **Music** ()
Default constructor.
- void **play** (musicID id)
Plays the music with the given identifier.
- void **stop** ()
Stops the music from playing.
- void **setVolume** (float volume)
Sets a volume to a certain level.
- bool **getPaused** ()

Gets wheither or not the music is paused.

- void **togglePlaying** ()

Toggle between playing and pausing the music.

3.16.1 Member Function Documentation

3.16.1.1 bool Music::getPaused ()

Gets wheither or not the music is paused.

Returns

true if music is paused, false if not.

3.16.1.2 void Music::play (musicID id)

Plays the music with the given identifier.

Exceptions

<i>std::runtime_error</i>	Thrown when a runtime error condition occurs.
---------------------------	---

Parameters

<i>id</i>	The identifier of the song.
-----------	-----------------------------

3.16.1.3 void Music::setVolume (float vol)

Sets a volume to a certain level.

Parameters

<i>vol</i>	The volume of the music.
------------	--------------------------

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Music.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Music.cpp

3.17 Player Class Reference

Public Member Functions

- **Player** (sf::Color, bool)
Constructor.
- sf::Color **getPlayer** ()
Gets the color of the player.
- bool **getActive** ()
Gets wheiter or not the player is active.
- void **setActive** (bool)
Sets the activity of the player.
- int **getPoints** ()
Gets the points of the player.
- void **subtractPoints** (CommandQueue &queue)

Subtract points.

- int **getMoney** ()
Gets the money of the player.
- void **setMoney** (int mon)
Sets the money value of the player.
- int **getUnitAttacks** ()
Gets the number unit attacks left.
- void **setUnitAttacks** (int attacks)
Sets the number unit attacks for this turn.
- void **diffUnitAttacks** (int attacks)
Difference the number of unit attacks.
- int **getUnitWalks** ()
Gets the number of units that can still walk this turn.
- void **resetUnitWalks** ()
Resets the number of units that can walk this turn.
- void **diffUnitWalks** (int steps)
Difference the number of units that can walk this turn.

3.17.1 Constructor & Destructor Documentation

3.17.1.1 Player::Player (sf::Color side, bool set)

Constructor.

Parameters

<i>side</i>	The color of the player (which side he is on).
<i>set</i>	true if active, false if not.

3.17.2 Member Function Documentation

3.17.2.1 void Player::diffUnitAttacks (int attacks)

Difference the number of unit attacks.

Parameters

<i>attacks</i>	The number of unit attacks for this turn.
----------------	---

3.17.2.2 void Player::diffUnitWalks (int steps)

Difference the number of units that can walk this turn.

Parameters

<i>steps</i>	The number of units already walked this turn.
--------------	---

3.17.2.3 bool Player::getActive ()

Gets wheiter or not the player is active.

Returns

true if active, false if not.

3.17.2.4 int Player::getMoney ()

Gets the money of the player.

Returns

The money of the player.

3.17.2.5 sf::Color Player::getPlayer ()

Gets the color of the player.

Returns

The color of the player (color is like an id tag for the player).

3.17.2.6 int Player::getPoints ()

Gets the points of the player.

Returns

The points of the player.

3.17.2.7 int Player::getUnitAttacks ()

Gets the number unit attacks left.

Returns

The unit attacks left for this turn.

3.17.2.8 int Player::getUnitWalks ()

Gets the number of units that can still walk this turn.

Returns

The number of units that can still walk this turn.

3.17.2.9 void Player::setActive (bool *set*)

Sets the activity of the player.

Parameters

<i>set</i>	true if active, false if not.
------------	-------------------------------

3.17.2.10 void Player::setMoney (int *mon*)

Sets the money value of the player.

Parameters

<i>mon</i>	The money.
------------	------------

3.17.2.11 void Player::setUnitAttacks (int *attacks*)

Sets the number unit attacks for this turn.

Parameters

<i>attacks</i>	The number of unit attacks for this turn.
----------------	---

3.17.2.12 void Player::subtractPoints (**CommandQueue** & *queue*)

Subtract points.

Parameters

<i>in, out</i>	<i>queue</i>	The queue.
----------------	--------------	------------

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Player.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Player.cpp

3.18 PlayerButton Class Reference

Inheritance diagram for PlayerButton:

Public Member Functions

- **PlayerButton** (textureID, const **textureHolder** &textures, sf::Vector2f)
Constructor.
- virtual void **handleClick** (**CommandQueue** &queue)
Handles mouse clicks.

Additional Inherited Members

3.18.1 Constructor & Destructor Documentation

3.18.1.1 PlayerButton::PlayerButton (textureID *id*, const textureHolder & *textures*, sf::Vector2f *pos*)

Constructor.

Parameters

<i>id</i>	The identifier of the PlayerButton (p. 23).
<i>textures</i>	The textures of the PlayerButton (p. 23).

<i>pos</i>	The position of the PlayerButton (p. 23) on the screen.
------------	--

3.18.2 Member Function Documentation

3.18.2.1 void PlayerButton::handleClick (**CommandQueue** & *queue*) [virtual]

Handles mouse clicks.

Parameters

<i>in, out</i>	<i>queue</i>	The queue.
----------------	--------------	------------

Reimplemented in **WonButton** (p. 42), and **EndTurnButton** (p. 10).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/PlayerButton.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/PlayerButton.cpp

3.19 PlayerObject Class Reference

Inheritance diagram for PlayerObject:

Public Member Functions

- void **setSelected** (bool)
*Sets the **PlayerObject** (p. 24) to being selected.*
- void **setOldSelected** (bool)
*Sets wheither or not the **PlayerObject** (p. 24) used to be selected.*
- bool **getSelected** ()
*Gets wheither or not the **PlayerObject** (p. 24) is selected.*
- int **getHP** ()
Gets the hitpoints.
- bool **makeSelected** (sf::Vector2f pos)
*Makes the **PlayerObject** (p. 24) selected or unselected.*
- bool **checkClicked** (sf::Vector2f pos)
*Check if the **PlayerObject** (p. 24) is being clicked on.*
- bool **damage** (int points)
*Damages the **PlayerObject** (p. 24).*

Protected Member Functions

- **PlayerObject** (textureID, const **textureHolder** &textures, sf::Vector2f, sf::Color)
Constructor.

Protected Attributes

- sf::Color **side**
- bool **selected** = false
- bool **oldSelected** = false
- int **hitpoints** = 100

3.19.1 Constructor & Destructor Documentation

3.19.1.1 `PlayerObject::PlayerObject (textureID id, const textureHolder & textures, sf::Vector2f pos, sf::Color color)` [protected]

Constructor.

Parameters

<i>id</i>	The identifier of the PlayerObject (p. 24).
<i>textures</i>	The textures of the PlayerObject (p. 24).
<i>pos</i>	The position of the PlayerObject (p. 24) on the screen.
<i>color</i>	The color of the PlayerObject (p. 24) (tells which player owns this object, red/blue).

3.19.2 Member Function Documentation

3.19.2.1 `bool PlayerObject::checkClicked (sf::Vector2f pos)`

Check if the **PlayerObject** (p. 24) is being clicked on.

Parameters

<i>pos</i>	The position of the mouse.
------------	----------------------------

Returns

true if clicked on, false if not.

3.19.2.2 `bool PlayerObject::damage (int points)`

Damages the **PlayerObject** (p. 24).

Parameters

<i>points</i>	The damage points.
---------------	--------------------

Returns

true if **PlayerObject** (p. 24) ran out of hitpoints, false if not.

3.19.2.3 `int PlayerObject::getHP ()`

Gets the hitpoints.

Returns

The hitpoints.

3.19.2.4 `bool PlayerObject::getSelected ()`

Gets wheither or not the **PlayerObject** (p. 24) is selected.

Returns

true if selected, false if not.

3.19.2.5 `bool PlayerObject::makeSelected (sf::Vector2f pos)`

Makes the **PlayerObject** (p. 24) selected or unselected.

Parameters

<i>pos</i>	The position of the mouse.
------------	----------------------------

Returns

true if selected, false if not.

3.19.2.6 void PlayerObject::setOldSelected (bool *b*)

Sets wheither or not the **PlayerObject** (p. 24) used to be selected.

Parameters

<i>b</i>	true if selected, false if not.
----------	---------------------------------

3.19.2.7 void PlayerObject::setSelected (bool *b*)

Sets the **PlayerObject** (p. 24) to being selected.

Parameters

<i>b</i>	true if selected, false if not.
----------	---------------------------------

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/PlayerObject.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/PlayerObject.cpp
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Selectable.cpp

3.20 ReadText Class Reference

Public Member Functions

- char **read** (std::ifstream &input)
Reads the given input (a text file with the level information).

3.20.1 Member Function Documentation

3.20.1.1 char ReadText::read (std::ifstream & *input*)

Reads the given input (a text file with the level information).

Exceptions

<i>endOfFile</i> (p. 9)	Thrown when an end of file error condition occurs.
<i>unknownCharacter</i> (p. 41)	Thrown when an unknown character error condition occurs.

Parameters

<i>in, out</i>	<i>input</i>	The input to read.
----------------	--------------	--------------------

Returns

A char.

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/ReadText.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/ReadText.cpp

3.21 Recruit Class Reference

Inheritance diagram for Recruit:

Public Member Functions

- **Recruit** (textureID, const **textureHolder** &textures, sf::Vector2f, sf::Color)
Constructor.

Additional Inherited Members

3.21.1 Constructor & Destructor Documentation

3.21.1.1 **Recruit::Recruit** (textureID *id*, const **textureHolder** & *textures*, sf::Vector2f *pos*, sf::Color *color*)

Constructor.

Parameters

<i>id</i>	The identifier of the Recruit (p. 28).
<i>textures</i>	The textures of the Recruit (p. 28).
<i>pos</i>	The position of the Recruit (p. 28) on the screen.
<i>color</i>	The color of the Recruit (p. 28) (tells which player is the owner).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/SubUnits.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/SubUnits.cpp

3.22 RecruitButton Class Reference

Inheritance diagram for RecruitButton:

Public Member Functions

- **RecruitButton** (textureID, const **textureHolder** &textures, sf::Vector2f pos)
Constructor.
- **Unit * bAction** (const **textureHolder** &textures, sf::Vector2f pos, sf::Color color) override
Actions.

Additional Inherited Members

3.22.1 Constructor & Destructor Documentation

3.22.1.1 `RecruitButton::RecruitButton (textureID id, const textureHolder & textures, sf::Vector2f pos)`

Constructor.

Parameters

<i>id</i>	The identifier of the RecruitButton (p. 28).
<i>textures</i>	The textures of the RecruitButton (p. 28).
<i>pos</i>	The position of the RecruitButton (p. 28) on the screen.

3.22.2 Member Function Documentation

3.22.2.1 `Unit * RecruitButton::bAction (const textureHolder & textures, sf::Vector2f pos, sf::Color color)` [`override`], [`virtual`]

Actions.

Parameters

<i>textures</i>	The textures.
<i>pos</i>	The position.
<i>color</i>	The color.

Returns

null if it fails, else an Unit*.

Reimplemented from **UnitButton** (p. 41).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/UnitButton.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/UnitButton.cpp

3.23 Resource Class Reference

Inheritance diagram for Resource:

Public Member Functions

- **Resource** (textureID, const textureHolder & textures, sf::Vector2f)
Constructor.
- void **setSelected** (bool)
Sets the resource to being selected or unselected.
- void **setOldSelected** (bool)
Sets if the resource used to be selected or not.
- bool **getSelected** ()
Gets wheiter or not the resource is selected.
- int **getResourceMoney** ()

Gets the money left in the resource.

- bool **makeSelected** (sf::Vector2f pos)

Makes the resource selected or unselected (depends on the mouseposition).

- bool **checkClicked** (sf::Vector2f pos)

Check wheiter or not the resource is clicked on.

- int **getMoney** ()

Gets money from the resource.

Protected Attributes

- bool **selected** = false
- bool **oldSelected** = false
- int **resourceMoney** = 800

Additional Inherited Members

3.23.1 Constructor & Destructor Documentation

3.23.1.1 Resource::Resource (textureID id, const textureHolder & textures, sf::Vector2f pos)

Constructor.

Parameters

<i>id</i>	The identifier of the Resource (p. 29).
<i>textures</i>	The textures of the Resource (p. 29).
<i>pos</i>	The position of the Resource (p. 29) on the screen.

3.23.2 Member Function Documentation

3.23.2.1 bool Resource::checkClicked (sf::Vector2f pos)

Check wheiter or not the resource is clicked on.

Parameters

<i>pos</i>	The position of the mouse.
------------	----------------------------

Returns

true if clicked on, false if not.

3.23.2.2 int Resource::getMoney ()

Gets money from the resource.

Returns

The money gotton from the resource.

3.23.2.3 int Resource::getResourceMoney ()

Gets the money left in the resource.

Returns

The money left in the resource.

3.23.2.4 bool Resource::getSelected ()

Gets wheiter or not the resource is selected.

Returns

true if selected, false if not.

3.23.2.5 bool Resource::makeSelected (sf::Vector2f pos)

Makes the resource selected or unselected (depends on the mouseposition).

Parameters

<i>pos</i>	The position of the mouse.
------------	----------------------------

Returns

true if selected, false if not.

3.23.2.6 void Resource::setOldSelected (bool b)

Sets if the resource used to be selected or not.

Parameters

<i>b</i>	true if selected, false if not.
----------	---------------------------------

3.23.2.7 void Resource::setSelected (bool b)

Sets the resource to being selected or unselected.

Parameters

<i>b</i>	true if selected, false if not.
----------	---------------------------------

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Resource.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Resource.cpp

3.24 ResourceHolder< Resource, Identifier > Class Template Reference

Public Member Functions

- void **load** (Identifier id, const std::string &filename)

- `template<typename Parameter >`
void **load** (Identifier id, const std::string &filename, Parameter para)
- **Resource** & **get** (Identifier id)
- const **Resource** & **get** (Identifier id) const

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/ResourceHolder.h

3.25 Scout Class Reference

Inheritance diagram for Scout:

Public Member Functions

- **Scout** (textureID, const **textureHolder** &textuers, sf::Vector2f, sf::Color)
Constructor.

Additional Inherited Members

3.25.1 Constructor & Destructor Documentation

3.25.1.1 Scout::Scout (textureID id, const textureHolder & textures, sf::Vector2f pos, sf::Color color)

Constructor.

Parameters

<i>id</i>	The identifier of the Scout (p. 32).
<i>textures</i>	The textures of the Scout (p. 32).
<i>pos</i>	The position of the Scout (p. 32) on the screen.
<i>color</i>	The color of the Scout (p. 32) (tells which player is the owner).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/SubUnits.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/SubUnits.cpp

3.26 ScoutButton Class Reference

Inheritance diagram for ScoutButton:

Public Member Functions

- **ScoutButton** (textureID, const **textureHolder** &textures, sf::Vector2f pos)
Constructor.
- **Unit * bAction** (const **textureHolder** &textures, sf::Vector2f pos, sf::Color color) override
Actions.

Additional Inherited Members

3.26.1 Constructor & Destructor Documentation

3.26.1.1 `ScoutButton::ScoutButton (textureID id, const textureHolder & textures, sf::Vector2f pos)`

Constructor.

Parameters

<i>id</i>	The identifier of the ScoutButton (p. 32).
<i>textures</i>	The textures of the ScoutButton (p. 32).
<i>pos</i>	The position of the ScoutButton (p. 32) on the screen.

3.26.2 Member Function Documentation

3.26.2.1 `Unit * ScoutButton::bAction (const textureHolder & textures, sf::Vector2f pos, sf::Color color)` [override], [virtual]

Actions.

Parameters

<i>textures</i>	The textures.
<i>pos</i>	The position.
<i>color</i>	The color.

Returns

null if it fails, else an Unit*.

Reimplemented from **UnitButton** (p. 41).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/UnitButton.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/UnitButton.cpp

3.27 Selectable Class Reference

Inheritance diagram for Selectable:

Public Member Functions

- void **setSelected** (bool)
- void **setOldSelected** (bool)
- bool **getSelected** ()
- bool **makeSelected** (sf::Vector2f pos)
- bool **checkClicked** (sf::Vector2f pos)

Protected Member Functions

- **Selectable** (textureID, const **textureHolder** &textures, sf::Vector2f)

Protected Attributes

- bool **selected** = false
- bool **oldSelected** = false

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Selectable.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Selectable.cpp

3.28 Soldier Class Reference

Inheritance diagram for Soldier:

Public Member Functions

- **Soldier** (textureID id, const **textureHolder** &textures, sf::Vector2f pos, sf::Color color)
Constructor.
- **Soldier** (textureID, const **textureHolder** &textures, sf::Vector2f, sf::Color)

Additional Inherited Members

3.28.1 Constructor & Destructor Documentation

3.28.1.1 Soldier::Soldier (textureID id, const textureHolder & textures, sf::Vector2f pos, sf::Color color)

Constructor.

Parameters

<i>id</i>	The identifier of the Soldier (p. 34).
<i>textures</i>	The textures of the Soldier (p. 34).
<i>pos</i>	The position of the Soldier (p. 34) on the screen.
<i>color</i>	The color of the Soldier (p. 34) (tells which player is the owner).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Soldier.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/SubUnits.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/SubUnits.cpp

3.29 SoldierButton Class Reference

Inheritance diagram for SoldierButton:

Public Member Functions

- **SoldierButton** (textureID, const **textureHolder** &textures, sf::Vector2f pos)
Constructor.
- **Unit * bAction** (const **textureHolder** &textures, sf::Vector2f pos, sf::Color color) override
Actions.

Additional Inherited Members

3.29.1 Constructor & Destructor Documentation

3.29.1.1 SoldierButton::SoldierButton (textureID id, const textureHolder & textures, sf::Vector2f pos)

Constructor.

Parameters

<i>id</i>	The identifier of the SoldierButton (p. 34).
<i>textures</i>	The textures of the SoldierButton (p. 34).
<i>pos</i>	The position of the SoldierButton (p. 34) on the screen.

3.29.2 Member Function Documentation

3.29.2.1 Unit * SoldierButton::bAction (const textureHolder & textures, sf::Vector2f pos, sf::Color color) [override], [virtual]

Actions.

Parameters

<i>textures</i>	The textures.
<i>pos</i>	The position.
<i>color</i>	The color.

Returns

null if it fails, else an Unit*.

Reimplemented from **UnitButton** (p. 41).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/UnitButton.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/UnitButton.cpp

3.30 Sound Class Reference

Inheritance diagram for Sound:

Public Member Functions

- **Sound** ()
Default constructor.

- void **play** (soundID)
Plays the sound with the given identifier.
- void **removeStoppedSound** ()
Removes the stopped sound.

3.30.1 Member Function Documentation

3.30.1.1 void Sound::play (soundID id)

Plays the sound with the given identifier.

Parameters

<i>id</i>	The identifier of the song.
-----------	-----------------------------

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Sound.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Sound.cpp

3.31 Terrain Class Reference

Inheritance diagram for Terrain:

Public Member Functions

- **Terrain** (textureID, const **textureHolder** &textures, sf::Vector2f)
Constructor.
- void **changeColor** (sf::Color col)
Change the color of the terrain tile.
- sf::Color **getColor** ()
Gets the color of the terrain.
- bool **checkClicked** (sf::Vector2f pos)
Check wheither or not a terrain tile is being clicked on.
- bool **getFree** ()
Gets wheither or not the terrain tile is free (has nothing on it).
- void **setFree** (bool f)
Sets the terrain tile free.

Additional Inherited Members

3.31.1 Constructor & Destructor Documentation

3.31.1.1 Terrain::Terrain (textureID id, const textureHolder & textures, sf::Vector2f pos)

Constructor.

Parameters

<i>id</i>	The identifier of the Terrain (p. 36).
<i>textures</i>	The textures of the Terrain (p. 36).
<i>pos</i>	The position of the Terrain (p. 36) on the screen.

3.31.2 Member Function Documentation

3.31.2.1 void Terrain::changeColor (sf::Color *col*)

Change the color of the terrain tile.

Parameters

<i>col</i>	The color of the terrain.
------------	---------------------------

3.31.2.2 bool Terrain::checkClicked (sf::Vector2f *pos*)

Check wheither or not a terrain tile is being clicked on.

Parameters

<i>pos</i>	The position of the mouse.
------------	----------------------------

Returns

true if clicked on, false if not.

3.31.2.3 sf::Color Terrain::getColor ()

Gets the color of the terrain.

Returns

The color of the terrain.

3.31.2.4 bool Terrain::getFree ()

Gets wheither or not the terrain tile is free (has nothing on it).

Returns

true if free, false if not.

3.31.2.5 void Terrain::setFree (bool *f*)

Sets the terrain tile free.

Parameters

<i>f</i>	true if free, false if not.
----------	-----------------------------

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Terrain.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Terrain.cpp

3.32 Unit Class Reference

Inheritance diagram for Unit:

Public Member Functions

- void **setPosition** (sf::Vector2f pos) override
Sets the position of the unit in the screen.
- void **resetTurn** ()
Resets the turn.
- int **getDP** ()
Gets the attack points(damage points).
- int **getWalklimit** ()
Gets the walklimit of the unit.
- std::string **getName** ()
Gets the name of the unit.
- int **getAttackrange** ()
Gets the attackrange of the unit.
- void **walk** (sf::Vector2f pos)
Walks to the given position.
- int **attack** ()
Simulates an attack by setting the new attackrange to zero after the attack end returning the attack points.
- void **resource** ()
Attackrange set to zero to simulate the resource being used.

Protected Member Functions

- **Unit** (textureID, const **textureHolder** &textures, sf::Vector2f, sf::Color)
Constructor.

Protected Attributes

- int **walklimit** = 2
- int **attackrange** = 3
- int **turnWalklimit** = walklimit
- int **turnAttackrange** = attackrange
- int **attackpoints** = 20
- std::string **unitName**

3.32.1 Constructor & Destructor Documentation

3.32.1.1 Unit::Unit (textureID id, const textureHolder & textures, sf::Vector2f pos, sf::Color color) [protected]

Constructor.

Parameters

<i>id</i>	The identifier of the Unit (p. 38).
<i>textures</i>	The textures of the Unit (p. 38).
<i>pos</i>	The position of the Unit (p. 38) in the screen.
<i>color</i>	The color of the Unit (p. 38) (blue/ red, depends on the active player).

3.32.2 Member Function Documentation

3.32.2.1 `int Unit::attack ()`

Simulates an attack by setting the new attackrange to zero after the attack end returning the attack points.

Returns

the attack points.

3.32.2.2 `int Unit::getAttackrange ()`

Gets the attackrange of the unit.

Returns

The attackrange of the unit (how many tiles his attack can reach).

3.32.2.3 `int Unit::getDP ()`

Gets the attack points(damage points).

Returns

The attack points.

3.32.2.4 `std::string Unit::getName ()`

Gets the name of the unit.

Returns

The name of the unit.

3.32.2.5 `int Unit::getWalklimit ()`

Gets the walklimit of the unit.

Returns

The walklimit of the unit (how many tiles he can move).

3.32.2.6 `void Unit::setPosition (sf::Vector2f pos) [override],[virtual]`

Sets the position of the unit in the screen.

Parameters

<i>pos</i>	The position of the unit.
------------	---------------------------

Reimplemented from **GameObject** (p. 14).

3.32.2.7 void Unit::walk (sf::Vector2f pos)

Walks to the given position.

Parameters

<i>pos</i>	The position where to walk to.
------------	--------------------------------

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Unit.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Unit.cpp

3.33 UIButton Class Reference

Inheritance diagram for UIButton:

Public Member Functions

- virtual **Unit** * **bAction** (const **textureHolder** &textures, sf::Vector2f pos, sf::Color color)
Actions.
- int **getCostMoney** ()
Gets how much money the unit costs.

Protected Member Functions

- **UIButton** (textureID, const **textureHolder** &textures, sf::Vector2f pos)
Constructor.

Protected Attributes

- int **cost** = 100

3.33.1 Constructor & Destructor Documentation

3.33.1.1 UIButton::UIButton (textureID id, const textureHolder & textures, sf::Vector2f pos) [protected]

Constructor.

Parameters

<i>id</i>	The identifier of the UIButton (p. 40).
-----------	--

<i>textures</i>	The textures of the UnitButton (p. 40).
<i>pos</i>	The position of the UnitButton (p. 40) on the screen.

3.33.2 Member Function Documentation

3.33.2.1 **Unit** * **UnitButton::bAction** (const textureHolder & *textures*, sf::Vector2f *pos*, sf::Color *color*) [virtual]

Actions.

Parameters

<i>textures</i>	The textures.
<i>pos</i>	The position.
<i>color</i>	The color.

Returns

null if it fails, else an Unit*.

Reimplemented in **ScoutButton** (p. 33), **SoldierButton** (p. 35), **DragonButton** (p. 9), and **RecruitButton** (p. 29).

3.33.2.2 int **UnitButton::getCostMoney** ()

Gets how much money the unit costs.

Returns

The amount of money the unit costs.

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/UnitButton.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/UnitButton.cpp

3.34 unknownCharacter Class Reference

Inheritance diagram for unknownCharacter:

Public Member Functions

- **unknownCharacter** (const std::string &name)
- const char * **what** () const override

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/Exceptions.h

3.35 WonButton Class Reference

Inheritance diagram for WonButton:

Public Member Functions

- **WonButton** (textureID, const **textureHolder** &textures, sf::Vector2f)
Constructor.
- void **handleClick** (**CommandQueue** &queue) override
Handles the mouse click.

Additional Inherited Members

3.35.1 Constructor & Destructor Documentation

3.35.1.1 WonButton::WonButton (textureID id, const textureHolder & textures, sf::Vector2f pos)

Constructor.

Parameters

<i>id</i>	The identifier of the WonButton (p. 41).
<i>textures</i>	The textures of the WonButton (p. 41).
<i>pos</i>	The position of the WonButton (p. 41) on the screen.

3.35.2 Member Function Documentation

3.35.2.1 void WonButton::handleClick (**CommandQueue** & queue) [override],[virtual]

Handles the mouse click.

Parameters

<i>in, out</i>	<i>queue</i>	The queue.
----------------	--------------	------------

Reimplemented from **PlayerButton** (p. 24).

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

- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/PlayerButton.h
- C:/Users/Remco/Documents/GitHub/Themaopdracht-5/Themaopdracht-5/gameproject/PlayerButton.cpp

Index

attack
Unit, 39

bAction
DragonButton, 9
RecruitButton, 29
ScoutButton, 33
SoldierButton, 35
UnitButton, 41

Bomber, 5
Bomber, 5

Building, 5
Building, 6
checkAction, 6

Button, 6
Button, 7
getClicked, 7

changeColor
Terrain, 37

checkAction
Building, 6
Factory, 12
Headquarters, 15

checkClicked
PlayerObject, 25
Resource, 30
Terrain, 37

Command, 7
CommandQueue, 7
isEmpty, 8
pop, 8
push, 8

damage
PlayerObject, 25

diffUnitAttacks
Player, 21

diffUnitWalks
Player, 21

DragonButton, 8
bAction, 9
DragonButton, 8

draw
GameObject, 13

endOfFile, 9
EndTurnButton, 9
EndTurnButton, 10
handleClick, 10

Factory, 10
checkAction, 12
Factory, 11

Game, 12
getExit, 12
GameObject, 12
draw, 13
GameObject, 13
getPosition, 14
getTilePosition, 14
handleMouse, 14
setPosition, 14
setTextRect, 14
update, 14

getActive
Player, 21
getAttackrange
Unit, 39
getClicked
Button, 7
getColor
Terrain, 37
getCostMoney
UnitButton, 41
getDP
Unit, 39
getExit
Game, 12
getFree
Terrain, 37
getHP
PlayerObject, 25
getMoney
Player, 21
Resource, 30
getName
Unit, 39
getPaused
Music, 20
getPlayer
Player, 22
getPoints
Player, 22
getPosition
GameObject, 14
getResourceMoney
Resource, 30
getSelected
PlayerObject, 25

- Resource, 31
- getTilePosition
 - GameObject, 14
- getUnitAttacks
 - Player, 22
- getUnitWalks
 - Player, 22
- getWalklimit
 - Unit, 39
- handleClick
 - EndTurnButton, 10
 - PlayerButton, 24
 - WonButton, 42
- handleMouse
 - GameObject, 14
 - MenuButton, 19
- Headquarters, 15
 - checkAction, 15
 - Headquarters, 15
- HolyGrail, 16
 - HolyGrail, 16
 - Reckoning, 16
- InputHandler, 16
 - InputHandler, 17
 - processInput, 18
- isEmpty
 - CommandQueue, 8
- makeSelected
 - PlayerObject, 25
 - Resource, 31
- MenuButton, 18
 - handleMouse, 19
 - MenuButton, 18
 - update, 19
- Music, 19
 - getPaused, 20
 - play, 20
 - setVolume, 20
- play
 - Music, 20
 - Sound, 36
- Player, 20
 - diffUnitAttacks, 21
 - diffUnitWalks, 21
 - getActive, 21
 - getMoney, 21
 - getPlayer, 22
 - getPoints, 22
 - getUnitAttacks, 22
 - getUnitWalks, 22
 - Player, 21
 - setActive, 22
 - setMoney, 22
 - setUnitAttacks, 23
 - subtractPoints, 23
- PlayerButton, 23
 - handleClick, 24
 - PlayerButton, 23
- PlayerObject, 24
 - checkClicked, 25
 - damage, 25
 - getHP, 25
 - getSelected, 25
 - makeSelected, 25
 - PlayerObject, 25
 - setOldSelected, 27
 - setSelected, 27
- pop
 - CommandQueue, 8
- processInput
 - InputHandler, 18
- push
 - CommandQueue, 8
- read
 - ReadText, 27
- ReadText, 27
 - read, 27
- Reckoning
 - HolyGrail, 16
- Recruit, 28
 - Recruit, 28
- RecruitButton, 28
 - bAction, 29
 - RecruitButton, 29
- Resource, 29
 - checkClicked, 30
 - getMoney, 30
 - getResourceMoney, 30
 - getSelected, 31
 - makeSelected, 31
 - Resource, 30
 - setOldSelected, 31
 - setSelected, 31
- ResourceHolder< Resource, Identifier >, 31
- Scout, 32
 - Scout, 32
- ScoutButton, 32
 - bAction, 33
 - ScoutButton, 33
- Selectable, 33
 - setActive
 - Player, 22
 - setFree
 - Terrain, 37
 - setMoney
 - Player, 22
 - setOldSelected
 - PlayerObject, 27
 - Resource, 31
 - setPosition
 - GameObject, 14
 - Unit, 39

- setSelected
 - PlayerObject, 27
 - Resource, 31
- setTextRect
 - GameObject, 14
- setUnitAttacks
 - Player, 23
- setVolume
 - Music, 20
- Soldier, 34
 - Soldier, 34
- SoldierButton, 34
 - bAction, 35
 - SoldierButton, 35
- Sound, 35
 - play, 36
- subtractPoints
 - Player, 23
- Terrain, 36
 - changeColor, 37
 - checkClicked, 37
 - getColor, 37
 - getFree, 37
 - setFree, 37
 - Terrain, 36
- Unit, 38
 - attack, 39
 - getAttackrange, 39
 - getDP, 39
 - getName, 39
 - getWalklimit, 39
 - setPosition, 39
 - Unit, 38
 - walk, 40
- UnitButton, 40
 - bAction, 41
 - getCostMoney, 41
 - UnitButton, 40
- unknownCharacter, 41
- update
 - GameObject, 14
 - MenuButton, 19
- walk
 - Unit, 40
- WonButton, 41
 - handleClick, 42
 - WonButton, 42