Zappy [GUI]

Generated by Doxygen 1.9.1

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 gui Namespace Reference

Data Structures

- class Camera

  Camera class.
- class Button

  Button class.
- class PanelBar

  PanelBar class.
- class Parallax

  Parallax class.
- class Window

  Window class.
- class Data

  Data class.
- class ItemDrawer

  ItemDrawer class.
- class Map

  Map Class.
- class Tile
- class Handler

  Handler class.
- class Network

  Network class.
- class SceneManager

  SceneManager class.
- class Team

  Team class.

Chapter 6

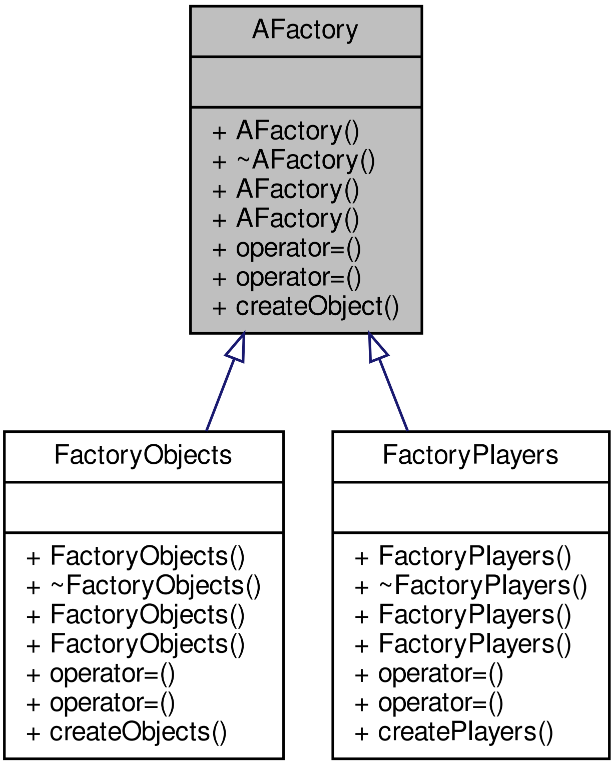
Data Structure Documentation

6.1 AFactory Class Reference

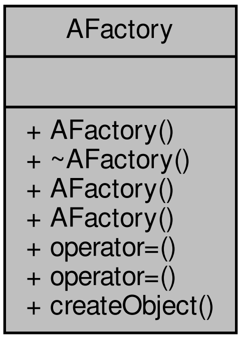
AFactory class.

```
#include <AFactory.hpp>
```

Inheritance diagram for AFactory:



Collaboration diagram for AFactory:



Public Member Functions

- AFactory ()=default

  Construct a new AFactory object.
- ∼AFactory () noexcept=default

  Destroy the AFactory object.
- AFactory (const AFactory &other)=delete

  Construct a new AFactory object by copy = delete.
- AFactory (AFactory &&other)=default

  Construct a new AFactory object by move = default.
- AFactory & operator= (const AFactory &other)=delete

  Assign a AFactory object by copy = delete.
- AFactory & operator= (AFactory &&other)=default

  Assign a AFactory object by move = default.
- template<typename T , typename ... Args>
  std::unique_ptr< T > createObject (Args &&...args)

  Create a Object.

6.1.1 Detailed Description

AFactory class.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 AFactory() [1/3]

```
AFactory::AFactory ( )  [default]
```

Construct a new AFactory object.

6.1.2.2 ∼AFactory()

```
AFactory::∼AFactory ( )  [default], [noexcept]
```

Destroy the AFactory object.

6.1.2.3 AFactory() [2/3]

```
AFactory::AFactory (
                const AFactory & other )  [delete]
```

Construct a new AFactory object by copy = delete.

Parameters

other	

6.1.2.4 AFactory() [3/3]

```
AFactory::AFactory (
                AFactory && other )  [default]
```

Construct a new AFactory object by move = default.

Parameters

other	

6.1.3 Member Function Documentation

6.1.3.1 createObject()

```
template<typename T , typename ... Args>
std::unique_ptr<T> AFactory::createObject (
             Args &&... args ) [inline]
```

Create a Object.

Template Parameters

T	
Args	

Parameters

args	

Returns

std::unique_ptr<T>

6.1.3.2 operator=() [1/2]

```
AFactory& AFactory::operator= (
             AFactory && other ) [default]
```

Assign a AFactory object by move = default.

Parameters

other	

Returns

AFactory&

6.1.3.3 operator=() [2/2]

```
AFactory& AFactory::operator= (
             const AFactory & other ) [delete]
```

Assign a AFactory object by copy = delete.

Parameters

other	

Returns

AFactory&

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

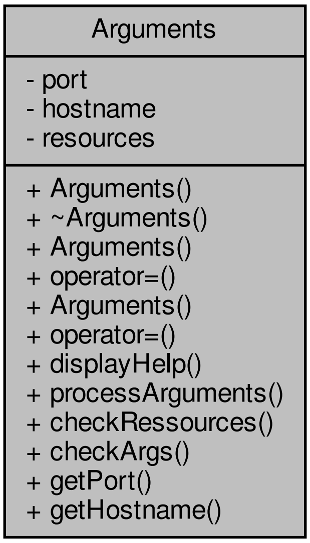
- Factory/AFactory.hpp

6.2 Arguments Class Reference

Class for the arguments.

```
#include <Arguments.hpp>
```

Collaboration diagram for Arguments:



Public Member Functions

- Arguments ()

  Construct a new Arguments object.
- ∼Arguments ()=default

  Destroy the Arguments object.
- Arguments (const Arguments &)=delete

  Construct a new Arguments object by copy deleted.
- Arguments & operator= (const Arguments &)=delete

  Assign a Arguments object by copy deleted.
- Arguments (Arguments &&)=default

  Construct a new Arguments object by move.
- Arguments & operator= (Arguments &&)=default

  Assign a Arguments object by move.
- void displayHelp ()

  Display the help.
- void processArguments (const int argc, char ∗const argv[ ])

  Process the arguments.
- void checkRessources ()

  Check if the resources exist.
- int checkArgs (const int argc, char ∗const argv[ ])

  Check the arguments.
- unsigned int getPort () const

  Get the port.
- const char ∗ getHostname () const

  Get the hostname.

Private Attributes

- unsigned int port
- const char ∗ hostname
- std::vector< std::string > resources

6.2.1 Detailed Description

Class for the arguments.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Arguments() [1/3]

```
Arguments::Arguments ( )
```

Construct a new Arguments object.

6.2.2.2 ∼Arguments()

```
Arguments::∼Arguments ( )  [default]
```

Destroy the Arguments object.

6.2.2.3 Arguments() [2/3]

```
Arguments::Arguments (
             const Arguments &  )  [delete]
```

Construct a new Arguments object by copy deleted.

6.2.2.4 Arguments() [3/3]

```
Arguments::Arguments (
             Arguments &&  )  [default]
```

Construct a new Arguments object by move.

6.2.3 Member Function Documentation

6.2.3.1 checkArgs()

```
int Arguments::checkArgs (
             const int argc,
             char *const argv[] )
```

Check the arguments.

Parameters

argc	The number of arguments
argv	The arguments

Returns

0 if no error, 84 otherwise

6.2.3.2 checkRessources()

```
void Arguments::checkRessources ( )
```

Check if the resources exist.

6.2.3.3 displayHelp()

```
void Arguments::displayHelp ( )
```

Display the help.

6.2.3.4 getHostname()

```
const char * Arguments::getHostname ( ) const
```

Get the hostname.

Returns

The hostname

6.2.3.5 getPort()

```
unsigned int Arguments::getPort ( ) const
```

Get the port.

Returns

The port

6.2.3.6 operator=() [1/2]

```
Arguments& Arguments::operator= (
             Arguments && ) [default]
```

Assign a Arguments object by move.

Returns

Arguments&

6.2.3.7 operator=() [2/2]

```
Arguments& Arguments::operator= (
             const Arguments & )  [delete]
```

Assign a Arguments object by copy deleted.

Returns

Arguments&

6.2.3.8 processArguments()

```
void Arguments::processArguments (
             const int argc,
             char *const argv[] )
```

Process the arguments.

Parameters

argc	The number of arguments
argv	The arguments

6.2.4 Field Documentation

6.2.4.1 hostname

```
const char * Arguments::hostname  [private]
```

6.2.4.2 port

```
unsigned int Arguments::port  [private]
```

6.2.4.3 resources

```
std::vector< std::string > Arguments::resources  [private]
```

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

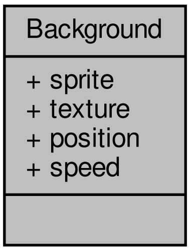
- Arguments/Arguments.hpp
- Arguments/Arguments.cpp

6.3 Background Struct Reference

Background struct.

```
#include <Parallax.hpp>
```

Collaboration diagram for Background:



Data Fields

- sf::Sprite sprite
- sf::Texture texture
- sf::Vector2f position
- float speed

6.3.1 Detailed Description

Background struct.

6.3.2 Field Documentation

6.3.2.1 position

```
sf::Vector2f Background::position
```

6.3.2.2 speed

```
float Background::speed
```

6.3.2.3 sprite

```
sf::Sprite Background::sprite
```

6.3.2.4 texture

```
sf::Texture Background::texture
```

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

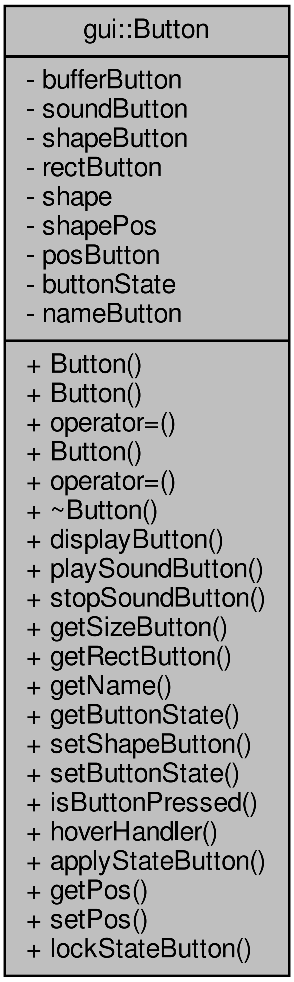
- Components/Parallax/Parallax.hpp

6.4 gui::Button Class Reference

Button class.

```
#include <Button.hpp>
```

Collaboration diagram for gui::Button:



Public Member Functions

- Button (const std::string &name, sf::IntRect rectButton, sf::Vector2f position)

  Construct a new Button object.
- Button (const Button &other)=delete

  Construct a new Button object by copy.
- Button & operator= (const Button &other)=delete

  Assign a button object by copy.
- Button (Button &&other)=default

  Construct a new Button object by move.
- Button & operator= (Button &&other)=default

  Assign a button object by move.
- ∼Button ()=default

  Destroy the Button object.

- void displayButton (Window &window, sf::Sprite &sprite)
- void playSoundButton ()
- void stopSoundButton ()
- const sf::Vector2f & getSizeButton ()
- const sf::Vector2f getRectButton ()
- const std::string & getName ()
- const ButtonState & getButtonState ()
- void setShapeButton (sf::Vector2f shape)
- void setButtonState (ButtonState state)
- bool isButtonPressed (sf::Vector2f mousePos)
- void hoverHandler (sf::Vector2f mousePos)
- void applyStateButton (sf::Sprite &sprite)
- sf::Vector2f & getPos ()
- void setPos (sf::Vector2f pos)

  Set the Pos of the button object.
- void lockStateButton ()

  Lock the state of the button.

Private Attributes

- sf::SoundBuffer bufferButton
- sf::Sound soundButton
- sf::RectangleShape shapeButton
- sf::IntRect rectButton
- sf::Vector2f shape
- sf::Vector2f shapePos
- sf::Vector2f posButton
- ButtonState buttonState
- std::string nameButton

6.4.1 Detailed Description

Button class.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Button() [1/3]

```
gui::Button::Button (
            const std::string & name,
            sf::IntRect rectButton,
            sf::Vector2f position )
```

Construct a new Button object.

Parameters

name	
rectButton	
position	

Exceptions

Errors::ErrorTexture	

6.4.2.2 Button() [2/3]

```
gui::Button::Button (
            const Button & other )  [delete]
```

Construct a new Button object by copy.

Parameters

other	

6.4.2.3 Button() [3/3]

```
gui::Button::Button (
            Button && other )  [default]
```

Construct a new Button object by move.

Parameters

other	

6.4.2.4 ∼Button()

```
gui::Button::∼Button ( )  [default]
```

Destroy the Button object.

6.4.3 Member Function Documentation

6.4.3.1 applyStateButton()

```
void gui::Button::applyStateButton (
             sf::Sprite & sprite )
```

6.4.3.2 displayButton()

```
void gui::Button::displayButton (
             Window & window,
             sf::Sprite & sprite )
```

6.4.3.3 getButtonState()

```
const ButtonState & gui::Button::getButtonState ( )
```

6.4.3.4 getName()

```
const std::string & gui::Button::getName ( )
```

6.4.3.5 getPos()

```
sf::Vector2f & gui::Button::getPos ( )
```

6.4.3.6 getRectButton()

```
const sf::Vector2f gui::Button::getRectButton ( )
```

6.4.3.7 getSizeButton()

```
const sf::Vector2f & gui::Button::getSizeButton ( )
```

6.4.3.8 hoverHandler()

```
void gui::Button::hoverHandler (
             sf::Vector2f mousePos )
```

6.4.3.9 isButtonPressed()

```
bool gui::Button::isButtonPressed (
             sf::Vector2f mousePos )
```

6.4.3.10 lockStateButton()

```
void gui::Button::lockStateButton ( )
```

Lock the state of the button.

6.4.3.11 operator=() [1/2]

```
Button& gui::Button::operator= (
             Button && other )  [default]
```

Assign a button object by move.

Parameters

other	

Returns

Button&

6.4.3.12 operator=() [2/2]

```
Button& gui::Button::operator= (
             const Button & other )  [delete]
```

Assign a button object by copy.

Parameters

other	

Returns

Button&

6.4.3.13 playSoundButton()

```
void gui::Button::playSoundButton ( )
```

6.4.3.14 setButtonState()

```
void gui::Button::setButtonState (
             ButtonState state )
```

6.4.3.15 setPos()

```
void gui::Button::setPos (
             sf::Vector2f pos )
```

Set the Pos of the button object.

Parameters

pos	

6.4.3.16 setShapeButton()

```
void gui::Button::setShapeButton (
             sf::Vector2f shape )
```

6.4.3.17 stopSoundButton()

```
void gui::Button::stopSoundButton ( )
```

6.4.4 Field Documentation

6.4.4.1 bufferButton

```
sf::SoundBuffer gui::Button::bufferButton  [private]
```

6.4.4.2 buttonState

```
ButtonState gui::Button::buttonState  [private]
```

6.4.4.3 nameButton

```
std::string gui::Button::nameButton  [private]
```

6.4.4.4 posButton

```
sf::Vector2f gui::Button::posButton  [private]
```

6.4.4.5 rectButton

```
sf::IntRect gui::Button::rectButton  [private]
```

6.4.4.6 shape

```
sf::Vector2f gui::Button::shape  [private]
```

6.4.4.7 shapeButton

```
sf::RectangleShape gui::Button::shapeButton  [private]
```

6.4.4.8 shapePos

```
sf::Vector2f gui::Button::shapePos  [private]
```

6.4.4.9 soundButton

```
sf::Sound gui::Button::soundButton  [private]
```

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

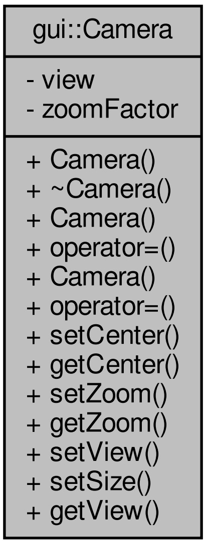
- Components/Buttons/Button.hpp
- Components/Buttons/Button.cpp

6.5 gui::Camera Class Reference

Camera class.

```
#include <Camera.hpp>
```

Collaboration diagram for gui::Camera:



Public Member Functions

- Camera (unsigned int width, unsigned int height)

    Construct a new Camera object.
- ∼Camera () noexcept=default

    Destroy the Camera object.
- Camera (const Camera &cpy)=delete

    Construct a new Camera object by copy.
- Camera & operator= (const Camera &src)=delete

    Assign a Camera object by copy.
- Camera (Camera &&cpy)=default

    Construct a new Camera object by move.
- Camera & operator= (Camera &&src)=default

    Assign a Camera object by move.
- void setCenter (sf::Vector2f pos)

    Set the Center of the camera Object.
- sf::Vector2f getCenter ()

    Get the Center of the camera Object.
- void setZoom (float zoom)

    Set the Zoom of the camera Object.
- const float & getZoom ()

    Get the Zoom object.
- void setView (sf::RenderWindow &window)

    Set the View of the camera Object.
- void setSize (sf::Vector2f size)

    Set the Size of the camera Object.
- sf::View & getView ()

    Get the View of the camera Object.

Private Attributes

- sf::View view
- float zoomFactor

6.5.1 Detailed Description

Camera class.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 Camera() [1/3]

```
gui::Camera::Camera (
            unsigned int width,
            unsigned int height )
```

Construct a new Camera object.

Parameters

width	
height	

6.5.2.2 ∼Camera()

```
gui::Camera::∼Camera ( )  [default], [noexcept]
```

Destroy the Camera object.

6.5.2.3 Camera() [2/3]

```
gui::Camera::Camera (
            const Camera & cpy )  [delete]
```

Construct a new Camera object by copy.

Parameters

cpy	

6.5.2.4 Camera() [3/3]

```
gui::Camera::Camera (
            Camera && cpy )  [default]
```

Construct a new Camera object by move.

Parameters

cpy	

6.5.3 Member Function Documentation

6.5.3.1 getCenter()

```
sf::Vector2f gui::Camera::getCenter ( )
```

Get the Center of the camera Object.

Returns

sf::Vector2f

6.5.3.2 getView()

```
sf::View & gui::Camera::getView ( )
```

Get the View of the camera Object.

Returns

sf::View&

6.5.3.3 getZoom()

```
const float & gui::Camera::getZoom ( )
```

Get the Zoom object.

Returns

float

6.5.3.4 operator=() [1/2]

```
Camera& gui::Camera::operator= (
             Camera && src )  [default]
```

Assign a Camera object by move.

Parameters

src	

Returns

Camera&

6.5.3.5 operator=() [2/2]

```
Camera& gui::Camera::operator= (
             const Camera & src )  [delete]
```

Assign a Camera object by copy.

Parameters

src	

Returns

Camera&

6.5.3.6 setCenter()

```
void gui::Camera::setCenter (
             sf::Vector2f pos )
```

Set the Center of the camera Object.

Parameters

pos	

6.5.3.7 setSize()

```
void gui::Camera::setSize (
             sf::Vector2f size )
```

Set the Size of the camera Object.

Parameters

size	

6.5.3.8 setView()

```
void gui::Camera::setView (
             sf::RenderWindow & window )
```

Set the View of the camera Object.

Parameters

window	

6.5.3.9 setZoom()

```
void gui::Camera::setZoom (
             float zoom )
```

Set the Zoom of the camera Object.

Parameters

zoom	

6.5.4 Field Documentation

6.5.4.1 view

```
sf::View gui::Camera::view  [private]
```

6.5.4.2 zoomFactor

```
float gui::Camera::zoomFactor  [private]
```

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

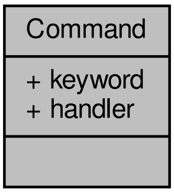
- Camera/Camera.hpp
- Camera/Camera.cpp

6.6 Command Struct Reference

Command struct.

```
#include <Handler.hpp>
```

Collaboration diagram for Command:



Data Fields

- std::string keyword
- std::function< void(std::istringstream &, gui::Data &)> handler

6.6.1 Detailed Description

Command struct.

6.6.2 Field Documentation

6.6.2.1 handler

```
std::function<void(std::istringstream&, gui::Data &)> Command::handler
```

6.6.2.2 keyword

```
std::string Command::keyword
```

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

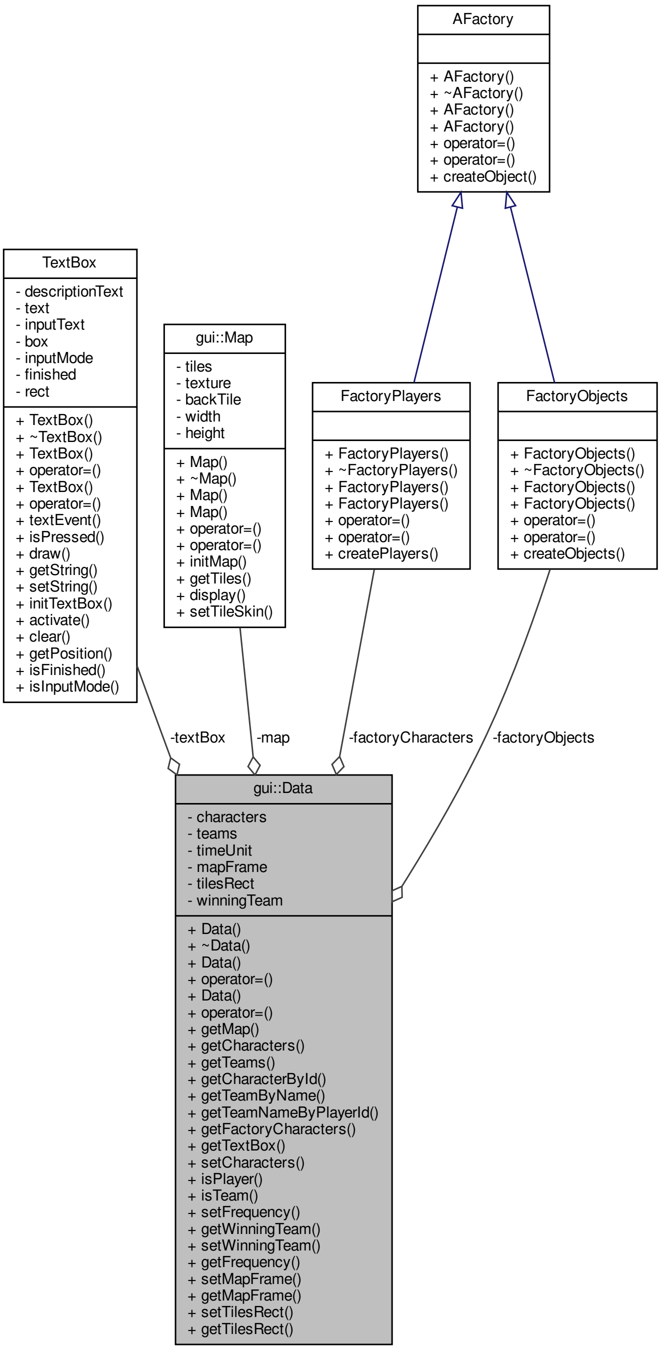
- Network/Handlers/Handler.hpp

6.7 gui::Data Class Reference

Data class.

```
#include <Data.hpp>
```

Collaboration diagram for gui::Data:



Public Member Functions

- Data ()=default

  Construct a new Data object.
- ∼Data () noexcept=default

  Destroy the Data object.
- Data (Data &&)=default

  Construct a new Data object by move.
- Data & operator= (Data &&)=default

  Assign a Data object by move.
- Data (const Data &)=delete

  Construct a new Data object by copy.
- Data & operator= (const Data &)=delete

  Assign a Data object by copy.
- Map & getMap ()

  Get the Map object.
- std::vector< std::unique_ptr< ICharacters > > & getCharacters ()

  Get the Characters.
- std::vector< gui::Team > & getTeams ()

  Get the Teams.
- ICharacters & getCharacterById (int id)

  Get the Character By Id.
- gui::Team & getTeamByName (const std::string &name)

  Get the Team By Name.
- const std::string & getTeamNameByPlayerId (int id)

  Get the Team Name By Player Id.
- FactoryPlayers & getFactoryCharacters ()

  Get the Factory Characters.
- TextBox & getTextBox ()

  Get the Text Box.
- void setCharacters (std::vector< std::unique_ptr< ICharacters >> &characters)

  Set the Characters.
- bool isPlayer (int id)

  Check if the player is in the game.
- bool isTeam (const std::string &name)

  Check if the team is in the game.
- void setFrequency (int frequency)

  Set the Frequency.
- const std::string & getWinningTeam ()

  Get the Winning Team.
- void setWinningTeam (const std::string &team)

  Set the Winning Team.
- const int & getFrequency ()

  Get the Frequency.
- void setMapFrame (int frame)

  Set the Map Frame.
- const int & getMapFrame ()

  Get the Map Frame.
- void setTilesRect (int rect)

  Set the Tiles Rect.
- const int & getTilesRect ()

  Get the Tiles Rect.

Private Attributes

- gui::Map map
- std::vector< std::unique_ptr< ICharacters > > characters
- FactoryObjects factoryObjects
- FactoryPlayers factoryCharacters
- std::vector< gui::Team > teams
- int timeUnit
- int mapFrame {0}
- int tilesRect {0}
- std::string winningTeam {""}
- TextBox textBox

6.7.1 Detailed Description

Data class.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 Data() [1/3]

```
gui::Data::Data ( )  [default]
```

Construct a new Data object.

6.7.2.2 ∼Data()

```
gui::Data::∼Data ( )  [default], [noexcept]
```

Destroy the Data object.

6.7.2.3 Data() [2/3]

```
gui::Data::Data (
               Data &&  )  [default]
```

Construct a new Data object by move.

6.7.2.4 Data() [3/3]

```
gui::Data::Data (
             const Data &  )  [delete]
```

Construct a new Data object by copy.

6.7.3 Member Function Documentation

6.7.3.1 getCharacterById()

```
ICharacters & gui::Data::getCharacterById (
             int id )
```

Get the Character By Id.

Parameters

id	

Returns

ICharacters&

Exceptions

Errors::ErrorCharacter	

6.7.3.2 getCharacters()

```
std::vector< std::unique_ptr< ICharacters > > & gui::Data::getCharacters ( )
```

Get the Characters.

Returns

std::vector<std::unique_ptr<ICharacters>>&

6.7.3.3 getFactoryCharacters()

```
FactoryPlayers & gui::Data::getFactoryCharacters ( )
```

Get the Factory Characters.

Returns

FactoryPlayers&

6.7.3.4 getFrequency()

```
const int & gui::Data::getFrequency ( )
```

Get the Frequency.

Returns

int

6.7.3.5 getMap()

```
gui::Map & gui::Data::getMap ( )
```

Get the Map object.

Returns

Map&

6.7.3.6 getMapFrame()

```
const int & gui::Data::getMapFrame ( )
```

Get the Map Frame.

Returns

int

6.7.3.7 getTeamByName()

```
gui::Team & gui::Data::getTeamByName (
             const std::string & name )
```

Get the Team By Name.

Parameters

name	

Returns

gui::Team&

Exceptions

Errors::ErrorTeam	

6.7.3.8 getTeamNameByPlayerId()

```
const std::string & gui::Data::getTeamNameByPlayerId (
            int id )
```

Get the Team Name By Player Id.

Parameters

id	

Returns

const std::string&

Exceptions

Errors::ErrorTeam	

6.7.3.9 getTeams()

```
std::vector< gui::Team > & gui::Data::getTeams ( )
```

Get the Teams.

Returns

std::vector<gui::Team>&

6.7.3.10 getTextBox()

```
TextBox & gui::Data::getTextBox ( )
```

Get the Text Box.

Returns

TextBox&

6.7.3.11 getTilesRect()

```
const int & gui::Data::getTilesRect ( )
```

Get the Tiles Rect.

Returns

int

6.7.3.12 getWinningTeam()

```
const std::string & gui::Data::getWinningTeam ( )
```

Get the Winning Team.

Returns

std::string&

6.7.3.13 isPlayer()

```
bool gui::Data::isPlayer (
             int id )
```

Check if the player is in the game.

Parameters

id	

Returns

true

false

6.7.3.14 isTeam()

```
bool gui::Data::isTeam (
            const std::string & name )
```

Check if the team is in the game.

Parameters

name	

Returns

true

false

6.7.3.15 operator=() [1/2]

```
Data& gui::Data::operator= (
            const Data & ) [delete]
```

Assign a Data object by copy.

Returns

Data&

6.7.3.16 operator=() [2/2]

```
Data& gui::Data::operator= (
            Data && ) [default]
```

Assign a Data object by move.

Returns

Data&

6.7.3.17 setCharacters()

```
void gui::Data::setCharacters (
            std::vector< std::unique_ptr< ICharacters >> & characters )
```

Set the Characters.

Parameters

characters	

6.7.3.18 setFrequency()

```
void gui::Data::setFrequency (
             int frequency )
```

Set the Frequency.

Parameters

frequency	

6.7.3.19 setMapFrame()

```
void gui::Data::setMapFrame (
             int frame )
```

Set the Map Frame.

Parameters

frame	

6.7.3.20 setTilesRect()

```
void gui::Data::setTilesRect (
             int rect )
```

Set the Tiles Rect.

Parameters

rect	

6.7.3.21 setWinningTeam()

```
void gui::Data::setWinningTeam (
             const std::string & team )
```

Set the Winning Team.

Parameters

team	

6.7.4 Field Documentation

6.7.4.1 characters

```
std::vector<std::unique_ptr<ICharacters> > gui::Data::characters  [private]
```

6.7.4.2 factoryCharacters

```
FactoryPlayers gui::Data::factoryCharacters  [private]
```

6.7.4.3 factoryObjects

```
FactoryObjects gui::Data::factoryObjects  [private]
```

6.7.4.4 map

```
gui::Map gui::Data::map  [private]
```

6.7.4.5 mapFrame

```
int gui::Data::mapFrame {0}  [private]
```

6.7.4.6 teams

```
std::vector< gui::Team > gui::Data::teams  [private]
```

6.7.4.7 textBox

```
TextBox gui::Data::textBox  [private]
```

6.7.4.8 tilesRect

```
gui::Data::tilesRect {0}  [private]
```

6.7.4.9 timeUnit

```
int gui::Data::timeUnit  [private]
```

6.7.4.10 winningTeam

```
std::string gui::Data::winningTeam {""}  [private]
```

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

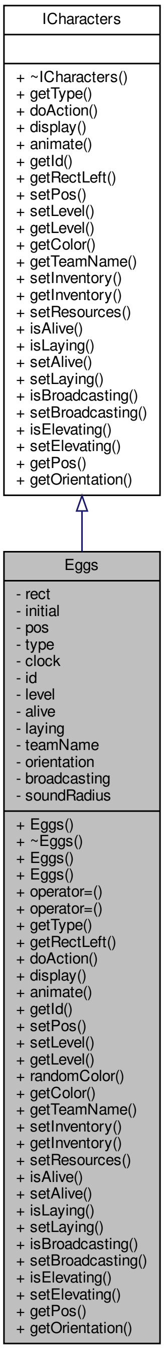
- Database/Data.hpp
- Database/Data.cpp

6.8 Eggs Class Reference

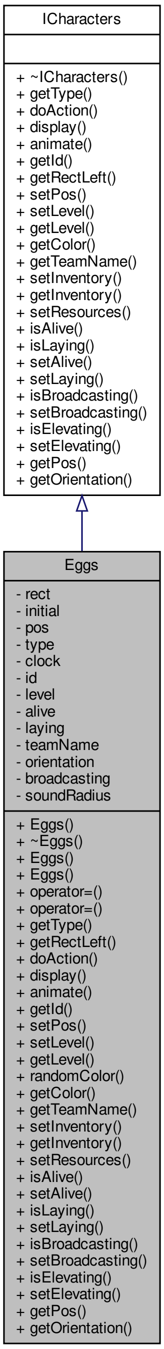
Eggs class.

```
#include <Eggs.hpp>
```

Inheritance diagram for Eggs:



Collaboration diagram for Eggs:



Public Member Functions

- Eggs (sf::Vector2f pos, int id, std::string teamName, gui::Data &data, Orientation orientation)

  Construct a new Eggs object.
- ∼Eggs () noexcept override=default

  Destroy the Eggs object.
- Eggs (const Eggs &other)=delete

Construct a new Eggs object by copy deleted.

- Eggs (Eggs &&other)=default

  Construct a new Eggs object by move.
- Eggs & operator= (const Eggs &other)=delete

  Assign a Eggs object by copy deleted.
- Eggs & operator= (Eggs &&other)=default

  Assign a Eggs object by move.
- CharacterType getType () const override

  Get the Type of the ICharacters.
- const int & getRectLeft () override

  Get the Rect Left of the Eggs.
- void doAction () override

  Do specific action for Eggs.
- void display (sf::RenderWindow &window, sf::Sprite &sprite, gui::ItemDrawer &drawer, int timeUnit, SoundBox &soundBox) override

  Display the Eggs.
- void animate (sf::Sprite &sprite) override

  Animate the Eggs.
- const int & getId () override

  Get the Id of the Eggs.
- void setPos (sf::Vector2f pos, Orientation orientation) override

  Set the Pos of the Eggs.
- void setLevel (int level) override

  Set the Level of the Eggs.
- const int & getLevel () override

  Get the Level of the Eggs.
- TypeEggs randomColor (gui::Data &data)

  Choose a random color for the Eggs.
- const std::string & getColor () override

  Get the Color of the Eggs.
- const std::string & getTeamName () override

  Get the Team Name of the Eggs.
- void setInventory (const std::array< int, 7 > &res) override

  Set the Inventory of the Eggs.
- const std::array< int, 7 > & getInventory () override

  Get the Inventory of the Eggs.
- void setResources (const int &res, const int &nb) override

  Set the Resources of the Eggs.
- const bool & isAlive () override

  Check if the Eggs is alive.
- void setAlive (bool alive) override

  Set the Alive boolean of the Eggs.
- const bool & isLaying () override

  Check if the Eggs is laying.
- void setLaying (bool laying) override

  Set the Laying boolean of the Eggs.
- const bool & isBroadcasting () override

  Check if the Eggs is broadcasting.
- void setBroadcasting (bool broadcastingParams) override

  Set the Broadcasting boolean of the Eggs.
- const int & isElevating () override

Check if the Eggs is elevating.

- void setElevating (int elevating) override

  Set the Elevating process of the Eggs.
- const sf::Vector2f & getPos () override

  Get the Pos of the Eggs.
- const Orientation & getOrientation () override

  Get the Orientation of the Eggs.

Private Attributes

- sf::IntRect rect
- sf::IntRect initial
- sf::Vector2f pos
- TypeEggs type
- sf::Clock clock
- int id
- int level
- bool alive
- bool laying
- std::string teamName
- Orientation orientation
- bool broadcasting
- int soundRadius

6.8.1 Detailed Description

Eggs class.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 Eggs() [1/3]

```
Eggs::Eggs (
            sf::Vector2f pos,
            int id,
            std::string teamName,
            gui::Data & data,
            Orientation orientation )
```

Construct a new Eggs object.

Parameters

pos	
id	
teamName	
data	
orientation	

6.8.2.2 ∼Eggs()

```
Eggs::∼Eggs ( )  [override], [default], [noexcept]
```

Destroy the Eggs object.

6.8.2.3 Eggs() [2/3]

```
Eggs::Eggs (
             const Eggs & other )  [delete]
```

Construct a new Eggs object by copy deleted.

Parameters

other	

6.8.2.4 Eggs() [3/3]

```
Eggs::Eggs (
             Eggs && other )  [default]
```

Construct a new Eggs object by move.

Parameters

other	

6.8.3 Member Function Documentation

6.8.3.1 animate()

```
void Eggs::animate (
             sf::Sprite & sprite )  [override], [virtual]
```

Animate the Eggs.

Parameters

sprite	

Implements ICharacters.

6.8.3.2 display()

```
void Eggs::display (
             sf::RenderWindow & window,
             sf::Sprite & sprite,
             gui::ItemDrawer & drawer,
             int timeUnit,
             SoundBox & soundBox )  [override], [virtual]
```

Display the Eggs.

Parameters

window	
sprite	
drawer	
timeUnit	
soundbox	

Implements ICharacters.

6.8.3.3 doAction()

```
void Eggs::doAction ( )  [override], [virtual]
```

Do specific action for Eggs.

Implements ICharacters.

6.8.3.4 getColor()

```
const std::string & Eggs::getColor ( )  [override], [virtual]
```

Get the Color of the Eggs.

Returns

const std::string&

Exceptions

Errors::ErrorColor	

Implements ICharacters.

6.8.3.5 getId()

```
const int & Eggs::getId ( )  [override], [virtual]
```

Get the Id of the Eggs.

Returns

const int&

Implements ICharacters.

6.8.3.6 getInventory()

```
const std::array< int, 7 > & Eggs::getInventory ( )  [override], [virtual]
```

Get the Inventory of the Eggs.

Returns

std::array<int, 7>&

Exceptions

Errors::ErrorInventory	

Implements ICharacters.

6.8.3.7 getLevel()

```
const int & Eggs::getLevel ( )  [override], [virtual]
```

Get the Level of the Eggs.

Returns

const int&

Implements ICharacters.

6.8.3.8 getOrientation()

```
const Orientation & Eggs::getOrientation ( ) [override], [virtual]
```

Get the Orientation of the Eggs.

Returns

const Orientation&

Implements ICharacters.

6.8.3.9 getPos()

```
const sf::Vector2f & Eggs::getPos ( ) [override], [virtual]
```

Get the Pos of the Eggs.

Returns

sf::Vector2f

Implements ICharacters.

6.8.3.10 getRectLeft()

```
const int & Eggs::getRectLeft ( ) [override], [virtual]
```

Get the Rect Left of the Eggs.

Returns

const int&

Implements ICharacters.

6.8.3.11 getTeamName()

```
const std::string & Eggs::getTeamName ( )  [override], [virtual]
```

Get the Team Name of the Eggs.

Returns

const std::string&

Implements ICharacters.

6.8.3.12 getType()

```
CharacterType Eggs::getType ( ) const  [override], [virtual]
```

Get the Type of the ICharacters.

Returns

CharacterType

Implements ICharacters.

6.8.3.13 isAlive()

```
const bool & Eggs::isAlive ( )  [override], [virtual]
```

Check if the Eggs is alive.

Returns

true

false

Implements ICharacters.

6.8.3.14 isBroadcasting()

```
const bool & Eggs::isBroadcasting ( )  [override], [virtual]
```

Check if the Eggs is broadcasting.

Returns

true

false

Implements ICharacters.

6.8.3.15 isElevating()

```
const int & Eggs::isElevating ( )  [override], [virtual]
```

Check if the Eggs is elevating.

Returns

int

Exceptions

Errors::ErrorElevate	

Implements ICharacters.

6.8.3.16 isLaying()

```
const bool & Eggs::isLaying ( )  [override], [virtual]
```

Check if the Eggs is laying.

Returns

true

false

Implements ICharacters.

6.8.3.17 operator=() [1/2]

```
Eggs& Eggs::operator= (
            const Eggs & other )  [delete]
```

Assign a Eggs object by copy deleted.

Parameters

other	

Returns

Eggs&

6.8.3.18 operator=() [2/2]

```
Eggs& Eggs::operator= (
             Eggs && other )  [default]
```

Assign a Eggs object by move.

Parameters

other	

Returns

Eggs&

6.8.3.19 randomColor()

```
TypeEggs Eggs::randomColor (
             gui::Data & data )
```

Choose a random color for the Eggs.

Parameters

data	

Returns

TypeEggs

6.8.3.20 setAlive()

```
void Eggs::setAlive (
             bool alive )  [override], [virtual]
```

Set the Alive boolean of the Eggs.

Parameters

alive	

Implements ICharacters.

6.8.3.21 setBroadcasting()

```
void Eggs::setBroadcasting (
             bool broadcastingParams )  [override], [virtual]
```

Set the Broadcasting boolean of the Eggs.

Parameters

broadcasting	

Implements ICharacters.

6.8.3.22 setElevating()

```
void Eggs::setElevating (
             int elevating )  [override], [virtual]
```

Set the Elevating process of the Eggs.

Parameters

elevating	

Exceptions

Errors::ErrorElevate	

Implements ICharacters.

6.8.3.23 setInventory()

```
void Eggs::setInventory (
             const std::array< int, 7 > & res )  [override], [virtual]
```

Set the Inventory of the Eggs.

Parameters

res	

Implements ICharacters.

6.8.3.24 setLaying()

```
void Eggs::setLaying (
            bool laying )  [override], [virtual]
```

Set the Laying boolean of the Eggs.

Parameters

laying	

Implements ICharacters.

6.8.3.25 setLevel()

```
void Eggs::setLevel (
            int level )  [override], [virtual]
```

Set the Level of the Eggs.

Parameters

level	

Implements ICharacters.

6.8.3.26 setPos()

```
void Eggs::setPos (
            sf::Vector2f pos,
            Orientation orientation )  [override], [virtual]
```

Set the Pos of the Eggs.

Parameters

pos	
orientation	

Implements ICharacters.

6.8.3.27 setResources()

```
void Eggs::setResources (
              const int & res,
              const int & nb )  [override], [virtual]
```

Set the Resources of the Eggs.

Parameters

res	
nb	

Implements ICharacters.

6.8.4 Field Documentation

6.8.4.1 alive

```
bool Eggs::alive  [private]
```

6.8.4.2 broadcasting

```
bool Eggs::broadcasting  [private]
```

6.8.4.3 clock

```
sf::Clock Eggs::clock  [private]
```

6.8.4.4 id

```
int Eggs::id  [private]
```

6.8.4.5 initial

```
sf::IntRect Eggs::initial  [private]
```

6.8.4.6 laying

```
bool Eggs::laying  [private]
```

6.8.4.7 level

```
int Eggs::level  [private]
```

6.8.4.8 orientation

```
Orientation Eggs::orientation  [private]
```

6.8.4.9 pos

```
sf::Vector2f Eggs::pos  [private]
```

6.8.4.10 rect

```
sf::IntRect Eggs::rect  [private]
```

6.8.4.11 soundRadius

```
int Eggs::soundRadius  [private]
```

6.8.4.12 teamName

```
std::string Eggs::teamName  [private]
```

6.8.4.13 type

```
TypeEggs Eggs::type  [private]
```

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

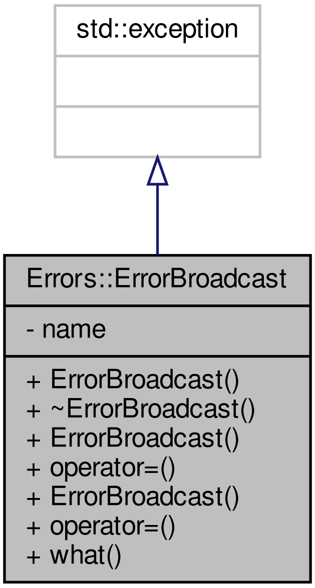
- Characters/Eggs/Eggs.hpp
- Characters/Eggs/Eggs.cpp

6.9 Errors::ErrorBroadcast Class Reference

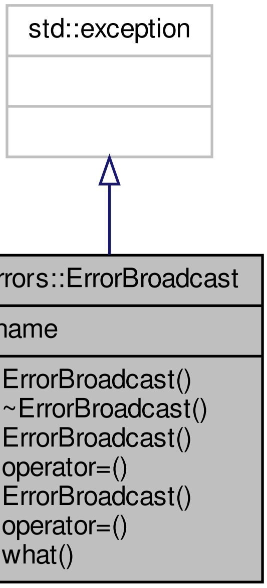
A class for broadcast Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorBroadcast:



Collaboration diagram for Errors::ErrorBroadcast:



Public Member Functions

- ErrorBroadcast (std::string m_what)

  Constructor a new Error Broadcast object.
- ∼ErrorBroadcast () noexcept override=default

  Destructor a new Error Broadcast object.
- ErrorBroadcast (const ErrorBroadcast &other)=delete

  Construct a new Error Broadcast object by copy.
- ErrorBroadcast & operator= (const ErrorBroadcast &other)=delete

  Assign a ErrorBroadcast object by copy.
- ErrorBroadcast (ErrorBroadcast &&other)=default

  Construct a new ErrorBroadcast object by move.
- ErrorBroadcast & operator= (ErrorBroadcast &&other)=default

  Assign a ErrorBroadcast object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.9.1 Detailed Description

A class for broadcast Error.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 ErrorBroadcast() [1/3]

```
Errors::ErrorBroadcast::ErrorBroadcast (
            std::string m_what )  [explicit]
```

Constructor a new Error Broadcast object.

Parameters

m_what	The error message

6.9.2.2 ∼ErrorBroadcast()

```
Errors::ErrorBroadcast::∼ErrorBroadcast ( )  [override], [default], [noexcept]
```

Destructor a new Error Broadcast object.

6.9.2.3 ErrorBroadcast() [2/3]

```
Errors::ErrorBroadcast::ErrorBroadcast (
            const ErrorBroadcast & other )  [delete]
```

Construct a new Error Broadcast object by copy.

Parameters

other	

6.9.2.4 ErrorBroadcast() [3/3]

```
Errors::ErrorBroadcast::ErrorBroadcast (
            ErrorBroadcast && other )  [default]
```

Construct a new ErrorBroadcast object by move.

Parameters

other	

6.9.3 Member Function Documentation

6.9.3.1 operator=() [1/2]

```
ErrorBroadcast& Errors::ErrorBroadcast::operator= (
            const ErrorBroadcast & other )  [delete]
```

Assign a ErrorBroadcast object by copy.

Parameters

other	

Returns

ErrorBroadcast&

6.9.3.2 operator=() [2/2]

```
ErrorBroadcast& Errors::ErrorBroadcast::operator= (
            ErrorBroadcast && other )  [default]
```

Assign a ErrorBroadcast object by move.

Parameters

other	

Returns

ErrorBroadcast&

6.9.3.3 what()

```
const char * Errors::ErrorBroadcast::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.9.4 Field Documentation

6.9.4.1 name

```
std::string Errors::ErrorBroadcast::name  [private]
```

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

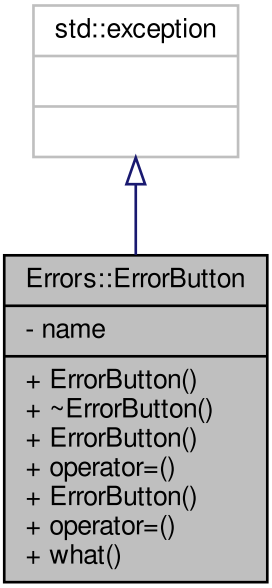
- Errors/Errors.hpp
- Errors/Errors.cpp

6.10 Errors::ErrorButton Class Reference

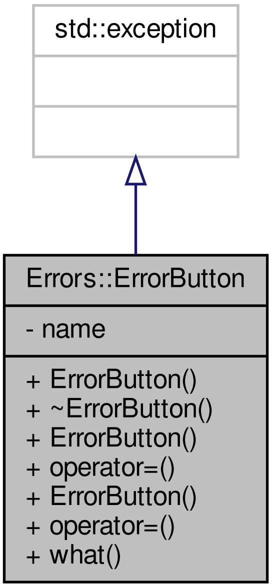
A class for Button Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorButton:



Collaboration diagram for Errors::ErrorButton:



Public Member Functions

- ErrorButton (std::string m_what)

    Constructor a new Error Button object.
- ∼ErrorButton () noexcept override=default

    Destructor a new Error Button object.
- ErrorButton (const ErrorButton &other)=delete

    Construct a new Error Button object by copy.
- ErrorButton & operator= (const ErrorButton &other)=delete

    Assign a ErrorButton object by copy.
- ErrorButton (ErrorButton &&other)=default

    Construct a new ErrorButton object by move.
- ErrorButton & operator= (ErrorButton &&other)=default

    Assign a ErrorButton object by move.
- const char ∗ what () const noexcept override

    Get the What object.

Private Attributes

- std::string name

6.10.1 Detailed Description

A class for Button Error.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 ErrorButton() [1/3]

```
Errors::ErrorButton::ErrorButton (
            std::string m_what )  [explicit]
```

Constructor a new Error Button object.

Parameters

m_what	The error message

6.10.2.2 ∼ErrorButton()

```
Errors::ErrorButton::∼ErrorButton ( )  [override], [default], [noexcept]
```

Destructor a new Error Button object.

6.10.2.3 ErrorButton() [2/3]

```
Errors::ErrorButton::ErrorButton (
            const ErrorButton & other )  [delete]
```

Construct a new Error Button object by copy.

Parameters

other	

6.10.2.4 ErrorButton() [3/3]

```
Errors::ErrorButton::ErrorButton (
            ErrorButton && other )  [default]
```

Construct a new ErrorButton object by move.

Parameters

other	

6.10.3 Member Function Documentation

6.10.3.1 operator=() [1/2]

```
ErrorButton& Errors::ErrorButton::operator= (
            const ErrorButton & other )  [delete]
```

Assign a ErrorButton object by copy.

Parameters

other	

Returns

ErrorButton&

6.10.3.2 operator=() [2/2]

```
ErrorButton& Errors::ErrorButton::operator= (
            ErrorButton && other )  [default]
```

Assign a ErrorButton object by move.

Parameters

other	

Returns

ErrorButton&

6.10.3.3 what()

```
const char * Errors::ErrorButton::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.10.4 Field Documentation

6.10.4.1 name

```
std::string Errors::ErrorButton::name  [private]
```

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

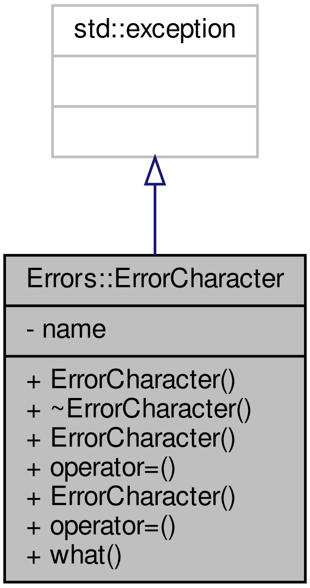
- Errors/Errors.hpp
- Errors/Errors.cpp

6.11 Errors::ErrorCharacter Class Reference

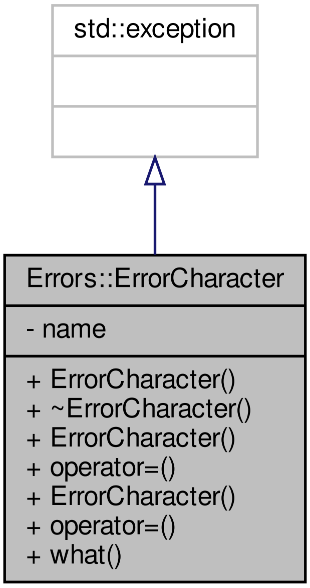
A class for Character Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorCharacter:



Collaboration diagram for Errors::ErrorCharacter:



Public Member Functions

- ErrorCharacter (std::string m_what)

  Constructor a new Error Character object.
- ∼ErrorCharacter () noexcept override=default

  Destructor a new Error Character object.
- ErrorCharacter (const ErrorCharacter &other)=delete

  Construct a new Error Character object by copy.
- ErrorCharacter & operator= (const ErrorCharacter &other)=delete

  Assign a ErrorCharacter object by copy.
- ErrorCharacter (ErrorCharacter &&other)=default

  Construct a new ErrorCharacter object by move.
- ErrorCharacter & operator= (ErrorCharacter &&other)=default

  Assign a ErrorCharacter object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.11.1 Detailed Description

A class for Character Error.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 ErrorCharacter() [1/3]

```
Errors::ErrorCharacter::ErrorCharacter (
            std::string m_what )  [explicit]
```

Constructor a new Error Character object.

Parameters

m_what	The error message

6.11.2.2 ∼ErrorCharacter()

```
Errors::ErrorCharacter::∼ErrorCharacter ( )  [override], [default], [noexcept]
```

Destructor a new Error Character object.

6.11.2.3 ErrorCharacter() [2/3]

```
Errors::ErrorCharacter::ErrorCharacter (
            const ErrorCharacter & other )  [delete]
```

Construct a new Error Character object by copy.

Parameters

other	

6.11.2.4 ErrorCharacter() [3/3]

```
Errors::ErrorCharacter::ErrorCharacter (
            ErrorCharacter && other )  [default]
```

Construct a new ErrorCharacter object by move.

Parameters

other	

6.11.3 Member Function Documentation

6.11.3.1 operator=() [1/2]

```
ErrorCharacter& Errors::ErrorCharacter::operator= (
            const ErrorCharacter & other )  [delete]
```

Assign a ErrorCharacter object by copy.

Parameters

other	

Returns

ErrorCharacter&

6.11.3.2 operator=() [2/2]

```
ErrorCharacter& Errors::ErrorCharacter::operator= (
            ErrorCharacter && other )  [default]
```

Assign a ErrorCharacter object by move.

Parameters

other	

Returns

ErrorCharacter&

6.11.3.3 what()

```
const char * Errors::ErrorCharacter::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.11.4 Field Documentation

6.11.4.1 name

```
std::string Errors::ErrorCharacter::name  [private]
```

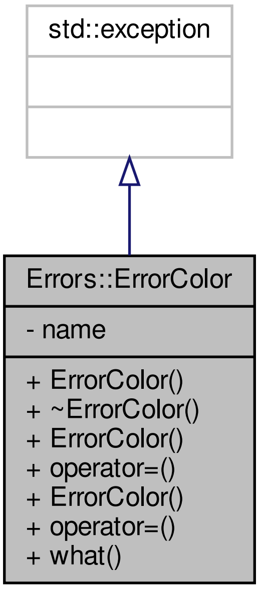
The documentation for this class was generated from the following files:

- Errors/Errors.hpp
- Errors/Errors.cpp

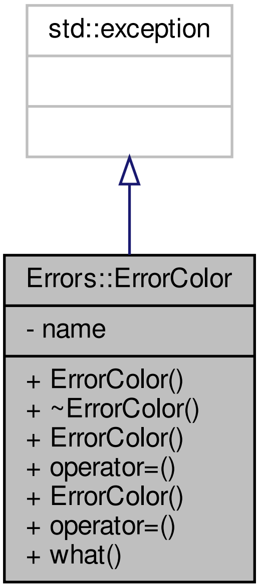
6.12 Errors::ErrorColor Class Reference

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorColor:



Collaboration diagram for Errors::ErrorColor:



Public Member Functions

- ErrorColor (std::string m_what)

  Constructor a new Error Color object.
- ∼ErrorColor () noexcept override=default

  Destructor a new Error Color object.
- ErrorColor (const ErrorColor &other)=delete

  Construct a new Error Color object by copy.
- ErrorColor & operator= (const ErrorColor &other)=delete

  Assign a ErrorColor object by copy.
- ErrorColor (ErrorColor &&other)=default

  Construct a new ErrorColor object by move.
- ErrorColor & operator= (ErrorColor &&other)=default

  Assign a ErrorColor object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.12.1 Constructor & Destructor Documentation

6.12.1.1 ErrorColor() [1/3]

```
Errors::ErrorColor::ErrorColor (
            std::string m_what )  [explicit]
```

Constructor a new Error Color object.

Parameters

m_what	The error message

6.12.1.2 ∼ErrorColor()

```
Errors::ErrorColor::∼ErrorColor ( )  [override], [default], [noexcept]
```

Destructor a new Error Color object.

6.12.1.3 ErrorColor() [2/3]

```
Errors::ErrorColor::ErrorColor (
            const ErrorColor & other )  [delete]
```

Construct a new Error Color object by copy.

Parameters

other	

6.12.1.4 ErrorColor() [3/3]

```
Errors::ErrorColor::ErrorColor (
            ErrorColor && other )  [default]
```

Construct a new ErrorColor object by move.

Parameters

other	

6.12.2 Member Function Documentation

6.12.2.1 operator=() [1/2]

```
ErrorColor& Errors::ErrorColor::operator= (
             const ErrorColor & other )  [delete]
```

Assign a ErrorColor object by copy.

Parameters

other	

Returns

ErrorColor&

6.12.2.2 operator=() [2/2]

```
ErrorColor& Errors::ErrorColor::operator= (
             ErrorColor && other )  [default]
```

Assign a ErrorColor object by move.

Parameters

other	

Returns

ErrorColor&

6.12.2.3 what()

```
const char * Errors::ErrorColor::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.12.3 Field Documentation

6.12.3.1 name

```
std::string Errors::ErrorColor::name  [private]
```

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

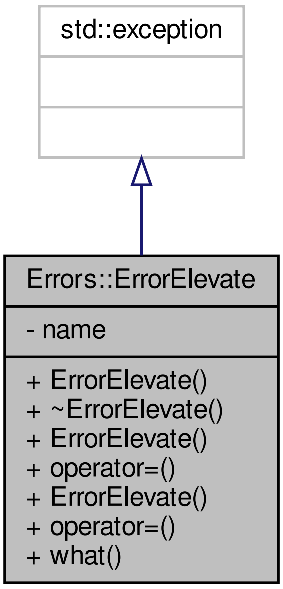
- Errors/Errors.hpp
- Errors/Errors.cpp

6.13 Errors::ErrorElevate Class Reference

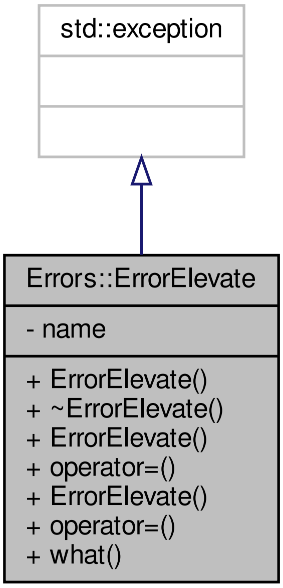
A class for Elevate Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorElevate:



Collaboration diagram for Errors::ErrorElevate:



Public Member Functions

- ErrorElevate (std::string m_what)

  Constructor a new Error Elevate object.
- ∼ErrorElevate () noexcept override=default

  Destructor a new Error Elevate object.
- ErrorElevate (const ErrorElevate &other)=delete

  Construct a new Error Elevate object by copy.
- ErrorElevate & operator= (const ErrorElevate &other)=delete

  Assign a ErrorElevate object by copy.
- ErrorElevate (ErrorElevate &&other)=default

  Construct a new ErrorElevate object by move.
- ErrorElevate & operator= (ErrorElevate &&other)=default

  Assign a ErrorElevate object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.13.1 Detailed Description

A class for Elevate Error.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 ErrorElevate() [1/3]

```
Errors::ErrorElevate::ErrorElevate (
             std::string m_what )  [explicit]
```

Constructor a new Error Elevate object.

Parameters

m_what	The error message

6.13.2.2 ∼ErrorElevate()

```
Errors::ErrorElevate::∼ErrorElevate ( )  [override], [default], [noexcept]
```

Destructor a new Error Elevate object.

6.13.2.3 ErrorElevate() [2/3]

```
Errors::ErrorElevate::ErrorElevate (
             const ErrorElevate & other )  [delete]
```

Construct a new Error Elevate object by copy.

Parameters

other	

6.13.2.4 ErrorElevate() [3/3]

```
Errors::ErrorElevate::ErrorElevate (
             ErrorElevate && other )  [default]
```

Construct a new ErrorElevate object by move.

Parameters

other	

6.13.3 Member Function Documentation

6.13.3.1 operator=() [1/2]

```
ErrorElevate& Errors::ErrorElevate::operator= (
            const ErrorElevate & other )  [delete]
```

Assign a ErrorElevate object by copy.

Parameters

other	

Returns

ErrorElevate&

6.13.3.2 operator=() [2/2]

```
ErrorElevate& Errors::ErrorElevate::operator= (
            ErrorElevate && other )  [default]
```

Assign a ErrorElevate object by move.

Parameters

other	

Returns

ErrorElevate&

6.13.3.3 what()

```
const char * Errors::ErrorElevate::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.13.4 Field Documentation

6.13.4.1 name

```
std::string Errors::ErrorElevate::name  [private]
```

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

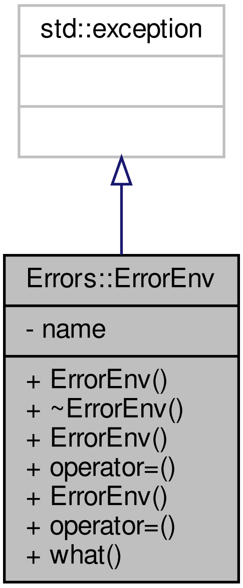
- Errors/Errors.hpp
- Errors/Errors.cpp

6.14 Errors::ErrorEnv Class Reference

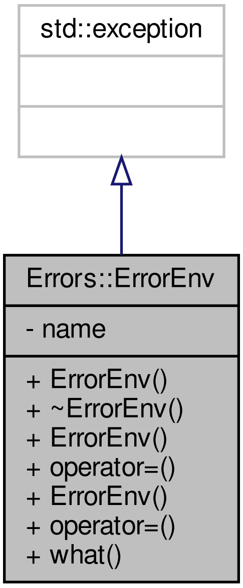
A class for env Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorEnv:



Collaboration diagram for Errors::ErrorEnv:



Public Member Functions

- ErrorEnv (std::string m_what)

  Constructor a new Error Env object.
- ∼ErrorEnv () noexcept override=default

  Destructor a new Error Env object.
- ErrorEnv (const ErrorEnv &other)=delete

  Construct a new Error Env object by copy.
- ErrorEnv & operator= (const ErrorEnv &other)=delete

  Assign a ErrorEnv object by copy.
- ErrorEnv (ErrorEnv &&other)=default

  Construct a new ErrorEnv object by move.
- ErrorEnv & operator= (ErrorEnv &&other)=default

  Assign a ErrorEnv object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.14.1 Detailed Description

A class for env Error.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 ErrorEnv() [1/3]

```
Errors::ErrorEnv::ErrorEnv (
            std::string m_what )  [explicit]
```

Constructor a new Error Env object.

Parameters

m_what	The error message

6.14.2.2 ∼ErrorEnv()

```
Errors::ErrorEnv::∼ErrorEnv ( )  [override], [default], [noexcept]
```

Destructor a new Error Env object.

6.14.2.3 ErrorEnv() [2/3]

```
Errors::ErrorEnv::ErrorEnv (
            const ErrorEnv & other )  [delete]
```

Construct a new Error Env object by copy.

Parameters

other	

6.14.2.4 ErrorEnv() [3/3]

```
Errors::ErrorEnv::ErrorEnv (
            ErrorEnv && other )  [default]
```

Construct a new ErrorEnv object by move.

Parameters

other	

6.14.3 Member Function Documentation

6.14.3.1 operator=() [1/2]

```
ErrorEnv& Errors::ErrorEnv::operator= (
            const ErrorEnv & other )  [delete]
```

Assign a ErrorEnv object by copy.

Parameters

other	

Returns

ErrorEnv&

6.14.3.2 operator=() [2/2]

```
ErrorEnv& Errors::ErrorEnv::operator= (
            ErrorEnv && other )  [default]
```

Assign a ErrorEnv object by move.

Parameters

other	

Returns

ErrorEnv&

6.14.3.3 what()

```
const char * Errors::ErrorEnv::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.14.4 Field Documentation

6.14.4.1 name

```
std::string Errors::ErrorEnv::name [private]
```

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

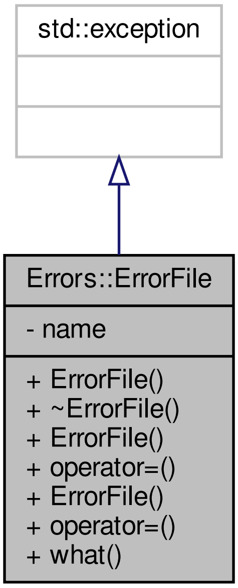
- Errors/Errors.hpp
- Errors/Errors.cpp

6.15 Errors::ErrorFile Class Reference

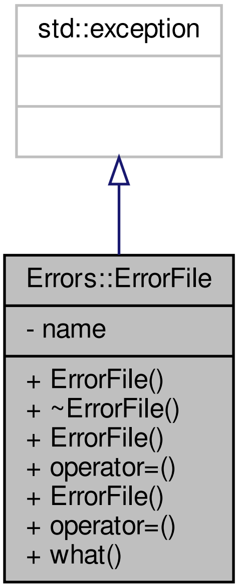
A class for file Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorFile:



Collaboration diagram for Errors::ErrorFile:



Public Member Functions

- ErrorFile (std::string m_what)

  Constructor a new Error File object.
- ∼ErrorFile () noexcept override=default

  Destructor a new Error File object.
- ErrorFile (const ErrorFile &other)=delete

  Construct a new Error File object by copy.
- ErrorFile & operator= (const ErrorFile &other)=delete

  Assign a ErrorFile object by copy.
- ErrorFile (ErrorFile &&other)=default

  Construct a new ErrorFile object by move.
- ErrorFile & operator= (ErrorFile &&other)=default

  Assign a ErrorFile object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.15.1 Detailed Description

A class for file Error.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 ErrorFile() [1/3]

```
Errors::ErrorFile::ErrorFile (
            std::string m_what )  [explicit]
```

Constructor a new Error File object.

Parameters

m_what	The error message

6.15.2.2 ∼ErrorFile()

```
Errors::ErrorFile::∼ErrorFile ( )  [override], [default], [noexcept]
```

Destructor a new Error File object.

6.15.2.3 ErrorFile() [2/3]

```
Errors::ErrorFile::ErrorFile (
            const ErrorFile & other )  [delete]
```

Construct a new Error File object by copy.

Parameters

other	

6.15.2.4 ErrorFile() [3/3]

```
Errors::ErrorFile::ErrorFile (
            ErrorFile && other )  [default]
```

Construct a new ErrorFile object by move.

Parameters

other	

6.15.3 Member Function Documentation

6.15.3.1 operator=() [1/2]

```
ErrorFile& Errors::ErrorFile::operator= (
             const ErrorFile & other )  [delete]
```

Assign a ErrorFile object by copy.

Parameters

other	

Returns

ErrorFile&

6.15.3.2 operator=() [2/2]

```
ErrorFile& Errors::ErrorFile::operator= (
             ErrorFile && other )  [default]
```

Assign a ErrorFile object by move.

Parameters

other	

Returns

ErrorFile&

6.15.3.3 what()

```
const char * Errors::ErrorFile::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.15.4 Field Documentation

6.15.4.1 name

```
std::string Errors::ErrorFile::name  [private]
```

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

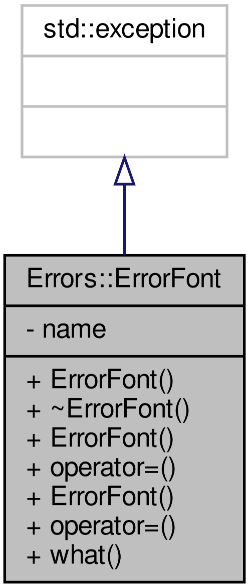
- Errors/Errors.hpp
- Errors/Errors.cpp

6.16 Errors::ErrorFont Class Reference

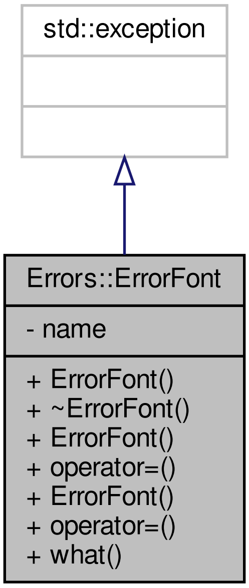
A class for font Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorFont:



Collaboration diagram for Errors::ErrorFont:



Public Member Functions

- ErrorFont (std::string m_what)

  Constructor a new Error Font object.
- ∼ErrorFont () noexcept override=default

  Destructor a new Error Font object.
- ErrorFont (const ErrorFont &other)=delete

  Construct a new Error Font object by copy.
- ErrorFont & operator= (const ErrorFont &other)=delete

  Assign a ErrorFont object by copy.
- ErrorFont (ErrorFont &&other)=default

  Construct a new ErrorFont object by move.
- ErrorFont & operator= (ErrorFont &&other)=default

  Assign a ErrorFont object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.16.1 Detailed Description

A class for font Error.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 ErrorFont() [1/3]

```
Errors::ErrorFont::ErrorFont (
            std::string m_what )  [explicit]
```

Constructor a new Error Font object.

Parameters

m_what	The error message

6.16.2.2 ∼ErrorFont()

```
Errors::ErrorFont::∼ErrorFont ( )  [override], [default], [noexcept]
```

Destructor a new Error Font object.

6.16.2.3 ErrorFont() [2/3]

```
Errors::ErrorFont::ErrorFont (
            const ErrorFont & other )  [delete]
```

Construct a new Error Font object by copy.

Parameters

other	

6.16.2.4 ErrorFont() [3/3]

```
Errors::ErrorFont::ErrorFont (
            ErrorFont && other )  [default]
```

Construct a new ErrorFont object by move.

Parameters

other	

6.16.3 Member Function Documentation

6.16.3.1 operator=() [1/2]

```
ErrorFont& Errors::ErrorFont::operator= (
             const ErrorFont & other )  [delete]
```

Assign a ErrorFont object by copy.

Parameters

other	

Returns

ErrorFont&

6.16.3.2 operator=() [2/2]

```
ErrorFont& Errors::ErrorFont::operator= (
             ErrorFont && other )  [default]
```

Assign a ErrorFont object by move.

Parameters

other	

Returns

ErrorFont&

6.16.3.3 what()

```
const char * Errors::ErrorFont::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.16.4 Field Documentation

6.16.4.1 name

```
std::string Errors::ErrorFont::name  [private]
```

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

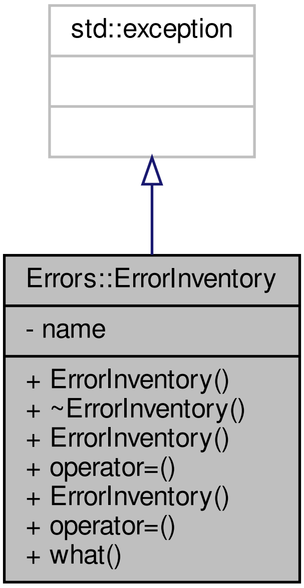
- Errors/Errors.hpp
- Errors/Errors.cpp

6.17 Errors::ErrorInventory Class Reference

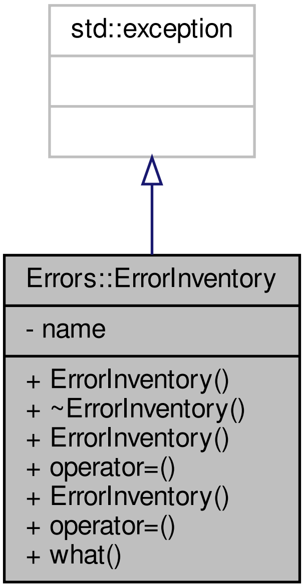
A class for Inventory Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorInventory:



Collaboration diagram for Errors::ErrorInventory:



Public Member Functions

- ErrorInventory (std::string m_what)

    Constructor a new Error Inventory object.
- ∼ErrorInventory () noexcept override=default

    Destructor a new Error Inventory object.
- ErrorInventory (const ErrorInventory &other)=delete

    Construct a new Error Inventory object by copy.
- ErrorInventory & operator= (const ErrorInventory &other)=delete

    Assign a ErrorInventory object by copy.
- ErrorInventory (ErrorInventory &&other)=default

    Construct a new ErrorInventory object by move.
- ErrorInventory & operator= (ErrorInventory &&other)=default

    Assign a ErrorInventory object by move.
- const char ∗ what () const noexcept override

    Get the What object.

Private Attributes

- std::string name

6.17.1 Detailed Description

A class for Inventory Error.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 ErrorInventory() [1/3]

```
Errors::ErrorInventory::ErrorInventory (
            std::string m_what )  [explicit]
```

Constructor a new Error Inventory object.

Parameters

m_what	The error message

6.17.2.2 ∼ErrorInventory()

```
Errors::ErrorInventory::∼ErrorInventory ( )  [override], [default], [noexcept]
```

Destructor a new Error Inventory object.

6.17.2.3 ErrorInventory() [2/3]

```
Errors::ErrorInventory::ErrorInventory (
            const ErrorInventory & other )  [delete]
```

Construct a new Error Inventory object by copy.

Parameters

other	

6.17.2.4 ErrorInventory() [3/3]

```
Errors::ErrorInventory::ErrorInventory (
            ErrorInventory && other )  [default]
```

Construct a new ErrorInventory object by move.

Parameters

other	

6.17.3 Member Function Documentation

6.17.3.1 operator=() [1/2]

```
ErrorInventory& Errors::ErrorInventory::operator= (
            const ErrorInventory & other ) [delete]
```

Assign a ErrorInventory object by copy.

Parameters

other	

Returns

ErrorInventory&

6.17.3.2 operator=() [2/2]

```
ErrorInventory& Errors::ErrorInventory::operator= (
            ErrorInventory && other ) [default]
```

Assign a ErrorInventory object by move.

Parameters

other	

Returns

ErrorInventory&

6.17.3.3 what()

```
const char * Errors::ErrorInventory::what ( ) const [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.17.4 Field Documentation

6.17.4.1 name

```
std::string Errors::ErrorInventory::name  [private]
```

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

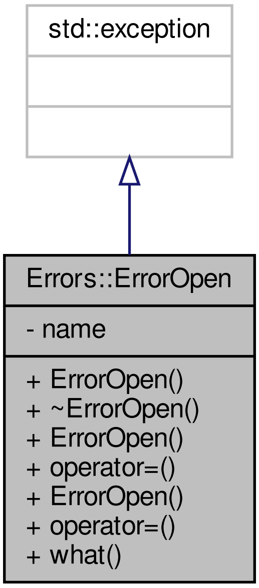
- Errors/Errors.hpp
- Errors/Errors.cpp

6.18 Errors::ErrorOpen Class Reference

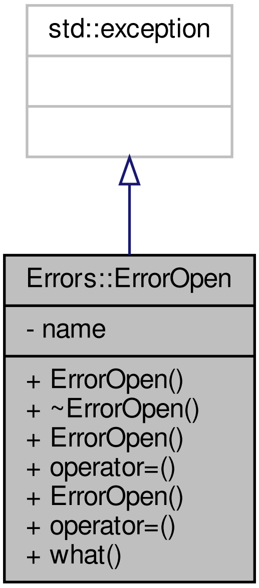
A class for open Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorOpen:



Collaboration diagram for Errors::ErrorOpen:



Public Member Functions

- ErrorOpen (std::string m_what)

  Constructor a new Error Open object.
- ∼ErrorOpen () noexcept override=default

  Destructor a new Error Open object.
- ErrorOpen (const ErrorOpen &other)=delete

  Construct a new Error Open object by copy.
- ErrorOpen & operator= (const ErrorOpen &other)=delete

  Assign a ErrorOpen object by copy.
- ErrorOpen (ErrorOpen &&other)=default

  Construct a new ErrorOpen object by move.
- ErrorOpen & operator= (ErrorOpen &&other)=default

  Assign a ErrorOpen object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.18.1 Detailed Description

A class for open Error.

6.18.2 Constructor & Destructor Documentation

6.18.2.1 ErrorOpen() [1/3]

```
Errors::ErrorOpen::ErrorOpen (
            std::string m_what )  [explicit]
```

Constructor a new Error Open object.

Parameters

m_what	The error message

6.18.2.2 ∼ErrorOpen()

```
Errors::ErrorOpen::∼ErrorOpen ( )  [override], [default], [noexcept]
```

Destructor a new Error Open object.

6.18.2.3 ErrorOpen() [2/3]

```
Errors::ErrorOpen::ErrorOpen (
            const ErrorOpen & other )  [delete]
```

Construct a new Error Open object by copy.

Parameters

other	

6.18.2.4 ErrorOpen() [3/3]

```
Errors::ErrorOpen::ErrorOpen (
            ErrorOpen && other )  [default]
```

Construct a new ErrorOpen object by move.

Parameters

other	

6.18.3 Member Function Documentation

6.18.3.1 operator=() [1/2]

```
ErrorOpen& Errors::ErrorOpen::operator= (
             const ErrorOpen & other )  [delete]
```

Assign a ErrorOpen object by copy.

Parameters

other	

Returns

ErrorOpen&

6.18.3.2 operator=() [2/2]

```
ErrorOpen& Errors::ErrorOpen::operator= (
             ErrorOpen && other )  [default]
```

Assign a ErrorOpen object by move.

Parameters

other	

Returns

ErrorOpen&

6.18.3.3 what()

```
const char * Errors::ErrorOpen::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.18.4 Field Documentation

6.18.4.1 name

```
std::string Errors::ErrorOpen::name [private]
```

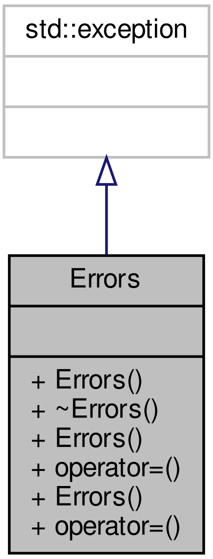
The documentation for this class was generated from the following files:

- Errors/Errors.hpp
- Errors/Errors.cpp

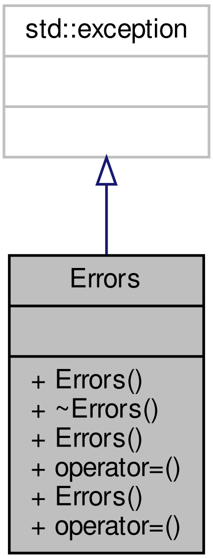
6.19 Errors Class Reference

```
#include <Errors.hpp>
```

Inheritance diagram for Errors:



Collaboration diagram for Errors:



Data Structures

- class ErrorBroadcast

  A class for broadcast Error.
- class ErrorButton

  A class for Button Error.
- class ErrorCharacter

  A class for Character Error.
- class ErrorColor
- class ErrorElevate

  A class for Elevate Error.
- class ErrorEnv

  A class for env Error.
- class ErrorFile

  A class for file Error.
- class ErrorFont

  A class for font Error.
- class ErrorInventory

  A class for Inventory Error.
- class ErrorOpen

  A class for open Error.
- class ErrorSocket

  A class for socket Error.
- class ErrorTeam

  A class for Team Error.
- class ErrorTexture

A class for Texture Error.

- class ErrorUsage

    A class for Usage Error.

Public Member Functions

- Errors ()=default

    Default Constructor a new Error T object.

- ∼Errors ()=default

    Default Destructor a new Error T object.

- Errors (const Errors &other)=delete

    Construct a new Errors object by copy.

- Errors & operator= (const Errors &other)=delete

    Assign a Errors object by copy.

- Errors (Errors &&other)=default

    Construct a new Errors object by move.

- Errors & operator= (Errors &&other)=default

    Assign a Errors object by move.

6.19.1 Constructor & Destructor Documentation

6.19.1.1 Errors() [1/3]

```
Errors::Errors ( )  [default]
```

Default Constructor a new Error T object.

6.19.1.2 ∼Errors()

```
Errors::∼Errors ( )  [default]
```

Default Destructor a new Error T object.

6.19.1.3 Errors() [2/3]

```
Errors::Errors (
            const Errors & other )  [delete]
```

Construct a new Errors object by copy.

Parameters

other	

6.19.1.4 Errors() [3/3]

```
Errors::Errors (
            Errors && other )  [default]
```

Construct a new Errors object by move.

Parameters

other	

6.19.2 Member Function Documentation

6.19.2.1 operator=() [1/2]

```
Errors& Errors::operator= (
            const Errors & other )  [delete]
```

Assign a Errors object by copy.

6.19.2.2 operator=() [2/2]

```
Errors& Errors::operator= (
            Errors && other )  [default]
```

Assign a Errors object by move.

Parameters

other	

Returns

Errors&

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

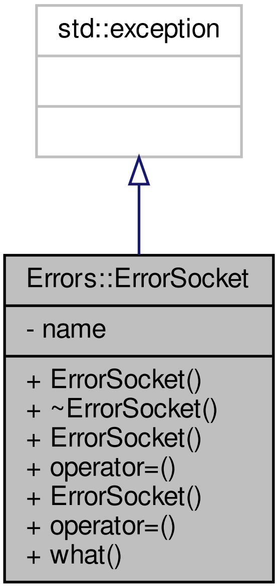
- Errors/Errors.hpp

6.20 Errors::ErrorSocket Class Reference

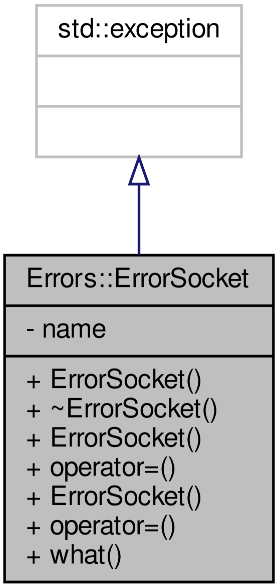
A class for socket Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorSocket:



Collaboration diagram for Errors::ErrorSocket:



Public Member Functions

- ErrorSocket (std::string m_what)

  Constructor a new Error Socket object.
- ∼ErrorSocket () noexcept override=default

  Destructor a new Error Socket object.
- ErrorSocket (const ErrorSocket &other)=delete

  Construct a new Error Socket object by copy.
- ErrorSocket & operator= (const ErrorSocket &other)=delete

  Assign a ErrorSocket object by copy.
- ErrorSocket (ErrorSocket &&other)=default

  Construct a new ErrorSocket object by move.
- ErrorSocket & operator= (ErrorSocket &&other)=default

  Assign a ErrorSocket object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.20.1 Detailed Description

A class for socket Error.

6.20.2 Constructor & Destructor Documentation

6.20.2.1 ErrorSocket() [1/3]

```
Errors::ErrorSocket::ErrorSocket (
            std::string m_what )  [explicit]
```

Constructor a new Error Socket object.

Parameters

m_what	The error message

6.20.2.2 ∼ErrorSocket()

```
Errors::ErrorSocket::∼ErrorSocket ( )  [override], [default], [noexcept]
```

Destructor a new Error Socket object.

6.20.2.3 ErrorSocket() [2/3]

```
Errors::ErrorSocket::ErrorSocket (
            const ErrorSocket & other )  [delete]
```

Construct a new Error Socket object by copy.

Parameters

other	

6.20.2.4 ErrorSocket() [3/3]

```
Errors::ErrorSocket::ErrorSocket (
            ErrorSocket && other )  [default]
```

Construct a new ErrorSocket object by move.

Parameters

other	

6.20.3 Member Function Documentation

6.20.3.1 operator=() [1/2]

```
ErrorSocket& Errors::ErrorSocket::operator= (
             const ErrorSocket & other )  [delete]
```

Assign a ErrorSocket object by copy.

Parameters

other	

Returns

ErrorSocket&

6.20.3.2 operator=() [2/2]

```
ErrorSocket& Errors::ErrorSocket::operator= (
             ErrorSocket && other )  [default]
```

Assign a ErrorSocket object by move.

Parameters

other	

Returns

ErrorSocket&

6.20.3.3 what()

```
const char * Errors::ErrorSocket::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.20.4 Field Documentation

6.20.4.1 name

```
std::string Errors::ErrorSocket::name  [private]
```

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

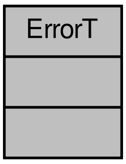
- Errors/Errors.hpp
- Errors/Errors.cpp

6.21 ErrorT Class Reference

A class to handle errors. This class provides a way to throw errors.

```
#include <Errors.hpp>
```

Collaboration diagram for ErrorT:



6.21.1 Detailed Description

A class to handle errors. This class provides a way to throw errors.

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

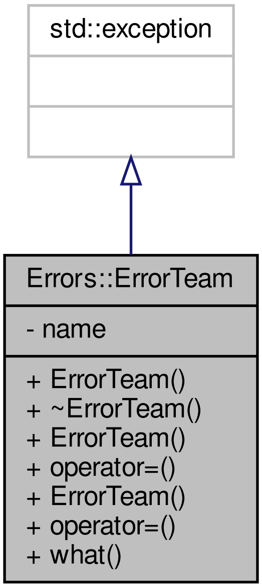
- Errors/Errors.hpp

6.22 Errors::ErrorTeam Class Reference

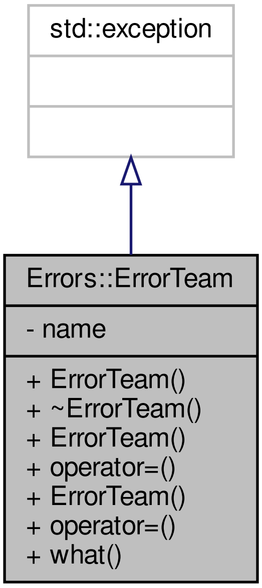
A class for Team Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorTeam:



Collaboration diagram for Errors::ErrorTeam:



Public Member Functions

- ErrorTeam (std::string m_what)

    Constructor a new Error Team object.
- ∼ErrorTeam () noexcept override=default

    Destructor a new Error Team object.
- ErrorTeam (const ErrorTeam &other)=delete

    Construct a new Error Team object by copy.
- ErrorTeam & operator= (const ErrorTeam &other)=delete

    Assign a ErrorTeam object by copy.
- ErrorTeam (ErrorTeam &&other)=default

    Construct a new ErrorTeam object by move.
- ErrorTeam & operator= (ErrorTeam &&other)=default

    Assign a ErrorTeam object by move.
- const char ∗ what () const noexcept override

    Get the What object.

Private Attributes

- std::string name

6.22.1 Detailed Description

A class for Team Error.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 ErrorTeam() [1/3]

```
Errors::ErrorTeam::ErrorTeam (
            std::string m_what )  [explicit]
```

Constructor a new Error Team object.

Parameters

m_what	The error message

6.22.2.2 ∼ErrorTeam()

```
Errors::ErrorTeam::∼ErrorTeam ( )  [override], [default], [noexcept]
```

Destructor a new Error Team object.

6.22.2.3 ErrorTeam() [2/3]

```
Errors::ErrorTeam::ErrorTeam (
            const ErrorTeam & other )  [delete]
```

Construct a new Error Team object by copy.

Parameters

other	

6.22.2.4 ErrorTeam() [3/3]

```
Errors::ErrorTeam::ErrorTeam (
            ErrorTeam && other )  [default]
```

Construct a new ErrorTeam object by move.

Parameters

other	

6.22.3 Member Function Documentation

6.22.3.1 operator=() [1/2]

```
ErrorTeam& Errors::ErrorTeam::operator= (
             const ErrorTeam & other )  [delete]
```

Assign a ErrorTeam object by copy.

Parameters

other	

Returns

ErrorTeam&

6.22.3.2 operator=() [2/2]

```
ErrorTeam& Errors::ErrorTeam::operator= (
             ErrorTeam && other )  [default]
```

Assign a ErrorTeam object by move.

Parameters

other	

Returns

ErrorTeam&

6.22.3.3 what()

```
const char * Errors::ErrorTeam::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.22.4 Field Documentation

6.22.4.1 name

```
std::string Errors::ErrorTeam::name  [private]
```

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

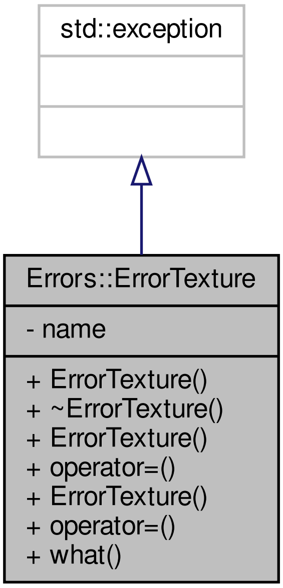
- Errors/Errors.hpp
- Errors/Errors.cpp

6.23 Errors::ErrorTexture Class Reference

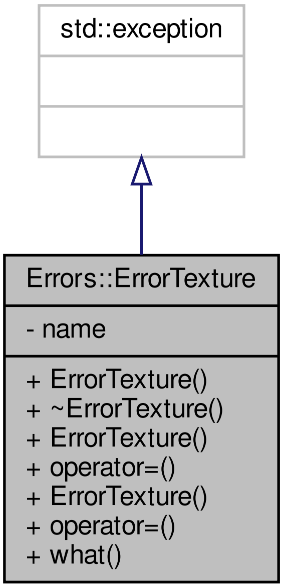
A class for Texture Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorTexture:



Collaboration diagram for Errors::ErrorTexture:



Public Member Functions

- ErrorTexture (std::string m_what)

  Constructor a new Error Texture object.
- ∼ErrorTexture () noexcept override=default

  Destructor a new Error Texture object.
- ErrorTexture (const ErrorTexture &other)=delete

  Construct a new Error Texture object by copy.
- ErrorTexture & operator= (const ErrorTexture &other)=delete

  Assign a ErrorTexture object by copy.
- ErrorTexture (ErrorTexture &&other)=default

  Construct a new ErrorTexture object by move.
- ErrorTexture & operator= (ErrorTexture &&other)=default

  Assign a ErrorTexture object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.23.1 Detailed Description

A class for Texture Error.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 ErrorTexture() [1/3]

```
Errors::ErrorTexture::ErrorTexture (
            std::string m_what )  [explicit]
```

Constructor a new Error Texture object.

Parameters

m_what	The error message

6.23.2.2 ∼ErrorTexture()

```
Errors::ErrorTexture::~ErrorTexture ( )  [override], [default], [noexcept]
```

Destructor a new Error Texture object.

6.23.2.3 ErrorTexture() [2/3]

```
Errors::ErrorTexture::ErrorTexture (
            const ErrorTexture & other )  [delete]
```

Construct a new Error Texture object by copy.

Parameters

other	

6.23.2.4 ErrorTexture() [3/3]

```
Errors::ErrorTexture::ErrorTexture (
            ErrorTexture && other )  [default]
```

Construct a new ErrorTexture object by move.

Parameters

other	

6.23.3 Member Function Documentation

6.23.3.1 operator=() [1/2]

```
ErrorTexture& Errors::ErrorTexture::operator= (
             const ErrorTexture & other )  [delete]
```

Assign a ErrorTexture object by copy.

Parameters

other	

Returns

ErrorTexture&

6.23.3.2 operator=() [2/2]

```
ErrorTexture& Errors::ErrorTexture::operator= (
             ErrorTexture && other )  [default]
```

Assign a ErrorTexture object by move.

Parameters

other	

Returns

ErrorTexture&

6.23.3.3 what()

```
const char * Errors::ErrorTexture::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.23.4 Field Documentation

6.23.4.1 name

```
std::string Errors::ErrorTexture::name  [private]
```

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

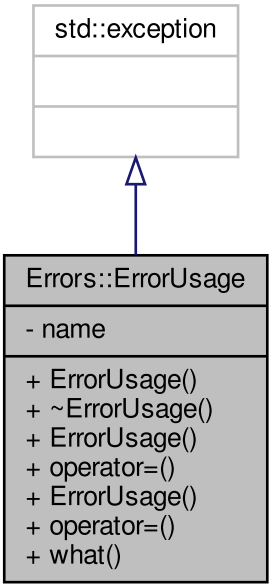
- Errors/Errors.hpp
- Errors/Errors.cpp

6.24 Errors::ErrorUsage Class Reference

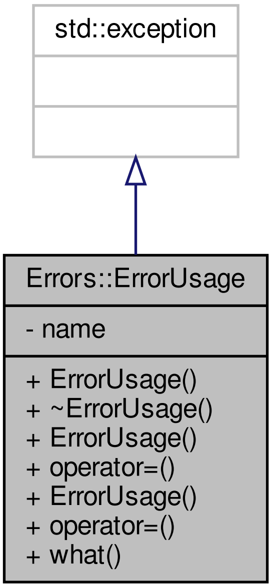
A class for Usage Error.

```
#include <Errors.hpp>
```

Inheritance diagram for Errors::ErrorUsage:



Collaboration diagram for Errors::ErrorUsage:



Public Member Functions

- ErrorUsage (std::string m_what)

  Constructor a new Error Usage object.
- ∼ErrorUsage () noexcept override=default

  Destructor a new Error Usage object.
- ErrorUsage (const ErrorUsage &other)=delete

  Construct a new Error Usage object by copy.
- ErrorUsage & operator= (const ErrorUsage &other)=delete

  Assign a ErrorUsage object by copy.
- ErrorUsage (ErrorUsage &&other)=default

  Construct a new ErrorUsage object by move.
- ErrorUsage & operator= (ErrorUsage &&other)=default

  Assign a ErrorUsage object by move.
- const char ∗ what () const noexcept override

  Get the What object.

Private Attributes

- std::string name

6.24.1 Detailed Description

A class for Usage Error.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 ErrorUsage() [1/3]

```
Errors::ErrorUsage::ErrorUsage (
             std::string m_what )  [explicit]
```

Constructor a new Error Usage object.

Parameters

m_what	The error message

6.24.2.2 ∼ErrorUsage()

```
Errors::ErrorUsage::∼ErrorUsage ( )  [override], [default], [noexcept]
```

Destructor a new Error Usage object.

6.24.2.3 ErrorUsage() [2/3]

```
Errors::ErrorUsage::ErrorUsage (
             const ErrorUsage & other )  [delete]
```

Construct a new Error Usage object by copy.

Parameters

other	

6.24.2.4 ErrorUsage() [3/3]

```
Errors::ErrorUsage::ErrorUsage (
             ErrorUsage && other )  [default]
```

Construct a new ErrorUsage object by move.

Parameters

other	

6.24.3 Member Function Documentation

6.24.3.1 operator=() [1/2]

```
ErrorUsage& Errors::ErrorUsage::operator= (
            const ErrorUsage & other )  [delete]
```

Assign a ErrorUsage object by copy.

Parameters

other	

Returns

ErrorUsage&

6.24.3.2 operator=() [2/2]

```
ErrorUsage& Errors::ErrorUsage::operator= (
            ErrorUsage && other )  [default]
```

Assign a ErrorUsage object by move.

Parameters

other	

Returns

ErrorUsage&

6.24.3.3 what()

```
const char * Errors::ErrorUsage::what ( ) const  [override], [noexcept]
```

Get the What object.

Returns

const char∗ The error message

6.24.4 Field Documentation

6.24.4.1 name

```
std::string Errors::ErrorUsage::name  [private]
```

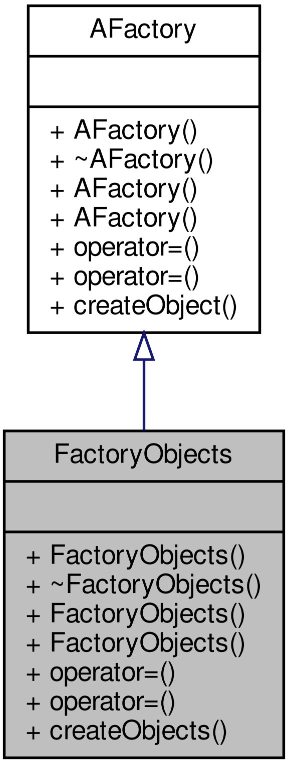
The documentation for this class was generated from the following files:

- Errors/Errors.hpp
- Errors/Errors.cpp

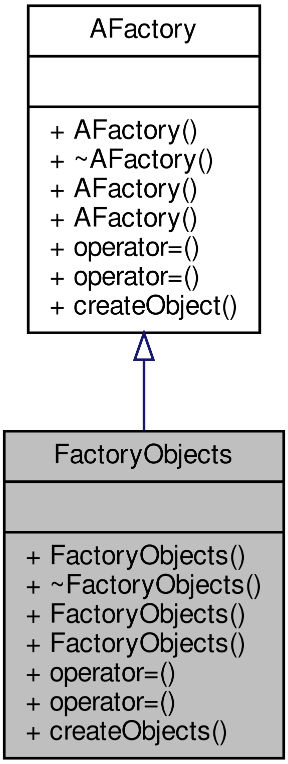
6.25 FactoryObjects Class Reference

```
#include <FactoryObjects.hpp>
```

Inheritance diagram for FactoryObjects:



Collaboration diagram for FactoryObjects:



Public Member Functions

- FactoryObjects ()=default
- ∼FactoryObjects () noexcept=default
- FactoryObjects (const FactoryObjects &other)=delete
- FactoryObjects (FactoryObjects &&other)=default
- FactoryObjects & operator= (const FactoryObjects &other)=delete
- FactoryObjects & operator= (FactoryObjects &&other)=default
- template<typename T , typename... Args>
  std::unique_ptr< T > createObjects (Args &&... args)

6.25.1 Constructor & Destructor Documentation

6.25.1.1 FactoryObjects() [1/3]

```
FactoryObjects::FactoryObjects ( )  [default]
```

6.25.1.2 ∼FactoryObjects()

```
FactoryObjects::∼FactoryObjects ( )  [default], [noexcept]
```

6.25.1.3 FactoryObjects() [2/3]

```
FactoryObjects::FactoryObjects (
             const FactoryObjects & other )  [delete]
```

6.25.1.4 FactoryObjects() [3/3]

```
FactoryObjects::FactoryObjects (
             FactoryObjects && other )  [default]
```

6.25.2 Member Function Documentation

6.25.2.1 createObjects()

```
template<typename T , typename...  Args>
std::unique_ptr<T> FactoryObjects::createObjects (
             Args &&...  args )  [inline]
```

6.25.2.2 operator=() [1/2]

```
FactoryObjects& FactoryObjects::operator= (
             const FactoryObjects & other )  [delete]
```

6.25.2.3 operator=() [2/2]

```
FactoryObjects& FactoryObjects::operator= (
             FactoryObjects && other )  [default]
```

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

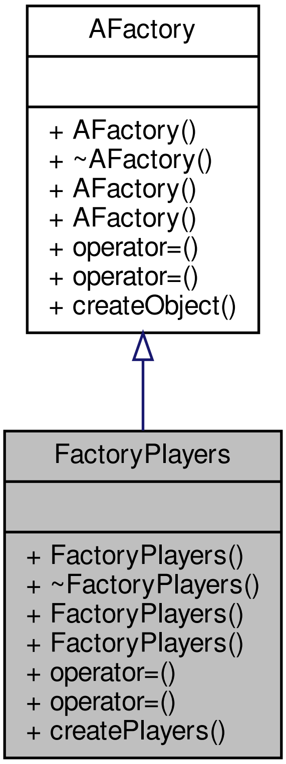
- Factory/FactoryObjects/FactoryObjects.hpp

6.26 FactoryPlayers Class Reference

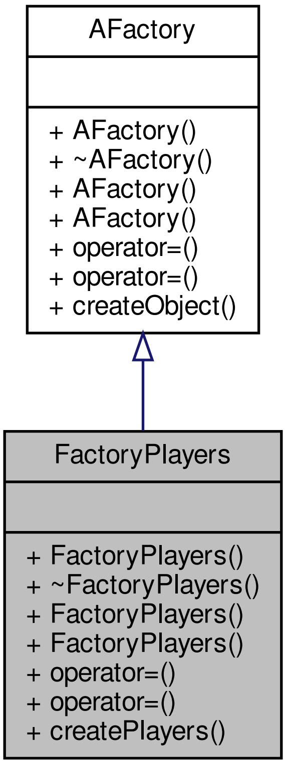
FactoryPlayers class.

```
#include <FactoryPlayers.hpp>
```

Inheritance diagram for FactoryPlayers:



Collaboration diagram for FactoryPlayers:



Public Member Functions

- FactoryPlayers ()=default

  Construct a new Factory Players object.
- ∼FactoryPlayers () noexcept=default

  Destroy the Factory Players object.
- FactoryPlayers (const FactoryPlayers &other)=delete

  Construct a new Factory Players object by copy = delete.
- FactoryPlayers (FactoryPlayers &&other)=default

  Construct a new Factory Players object by move.
- FactoryPlayers & operator= (const FactoryPlayers &other)=delete

  Assign a Factory Players object by copy = delete.
- FactoryPlayers & operator= (FactoryPlayers &&other)=default

  Assign a Factory Players object by move.
- template<typename T , typename... Args>
  std::unique_ptr< T > createPlayers (Args &&... args)

  Create a Characters object.

6.26.1 Detailed Description

FactoryPlayers class.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 FactoryPlayers() [1/3]

```
FactoryPlayers::FactoryPlayers ( )  [default]
```

Construct a new Factory Players object.

6.26.2.2 ∼FactoryPlayers()

```
FactoryPlayers::∼FactoryPlayers ( )  [default], [noexcept]
```

Destroy the Factory Players object.

6.26.2.3 FactoryPlayers() [2/3]

```
FactoryPlayers::FactoryPlayers (
              const FactoryPlayers & other )  [delete]
```

Construct a new Factory Players object by copy = delete.

Parameters

other	

6.26.2.4 FactoryPlayers() [3/3]

```
FactoryPlayers::FactoryPlayers (
              FactoryPlayers && other )  [default]
```

Construct a new Factory Players object by move.

Parameters

other	

6.26.3 Member Function Documentation

6.26.3.1 createPlayers()

```
template<typename T , typename...  Args>
std::unique_ptr<T> FactoryPlayers::createPlayers (
            Args &&...  args )  [inline]
```

Create a Characters object.

Template Parameters

T	
Args	

Parameters

args	

Returns

std::unique_ptr<T>

6.26.3.2 operator=() [1/2]

```
FactoryPlayers& FactoryPlayers::operator= (
            const FactoryPlayers & other )  [delete]
```

Assign a Factory Players object by copy = delete.

Parameters

other	

Returns

FactoryPlayers&

6.26.3.3 operator=() [2/2]

```
FactoryPlayers& FactoryPlayers::operator= (
            FactoryPlayers && other )  [default]
```

Assign a Factory Players object by move.

Parameters

other	

Returns

FactoryPlayers&

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

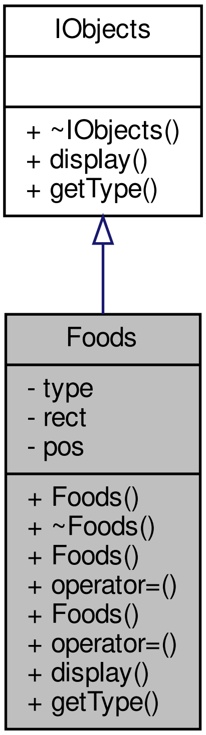
- Factory/FactoryPlayers/FactoryPlayers.hpp

6.27 Foods Class Reference

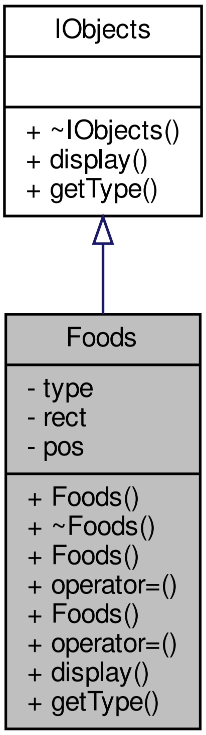
Foods class.

`#include <Foods.hpp>`

Inheritance diagram for Foods:



Collaboration diagram for Foods:



Public Member Functions

- Foods (sf::Vector2f pos, TypeObject typeObject)

    Construct a new Foods object.
- ∼Foods () noexcept override=default

    Destroy the Foods object.
- Foods (const Foods &stone)=default

    Construct a new Foods object by copy.
- Foods & operator= (const Foods &stone)=default

    Assign a Foods object by copy.
- Foods (Foods &&stone)=default

    Construct a new Foods object by move.
- Foods & operator= (Foods &&stone)=default

    Assign a Foods object by move.
- void display (sf::RenderWindow &window, sf::Sprite &sprite, float size) override

    Display the object.
- const TypeObject & getType () override

    Get The type of the object.

Private Attributes

- TypeObject type

  Get The Sprite of the object.
- sf::IntRect rect
- sf::Vector2f pos

6.27.1 Detailed Description

Foods class.

6.27.2 Constructor & Destructor Documentation

6.27.2.1 Foods() [1/3]

```
Foods::Foods (
                sf::Vector2f pos,
                TypeObject typeObject )
```

Construct a new Foods object.

Parameters

pos	
typeObject	

6.27.2.2 ∼Foods()

```
Foods::∼Foods ( )  [override], [default], [noexcept]
```

Destroy the Foods object.

6.27.2.3 Foods() [2/3]

```
Foods::Foods (
                const Foods & stone )  [default]
```

Construct a new Foods object by copy.

Parameters

stone	

6.27.2.4 Foods() [3/3]

```
Foods::Foods (
             Foods && stone )  [default]
```

Construct a new Foods object by move.

Parameters

stone	

6.27.3 Member Function Documentation

6.27.3.1 display()

```
void Foods::display (
             sf::RenderWindow & window,
             sf::Sprite & sprite,
             float size )  [override], [virtual]
```

Display the object.

Parameters

window	The window where the object will be displayed
sprite	The sprite of the object
size	The size of the object

Implements IObjects.

6.27.3.2 getType()

```
const TypeObject & Foods::getType ( )  [override], [virtual]
```

Get The type of the object.

Returns

The type of the object

Implements IObjects.

6.27.3.3 operator=() [1/2]

```
Foods& Foods::operator= (
            const Foods & stone )   [default]
```

Assign a Foods object by copy.

Parameters

stone	

Returns

Foods&

6.27.3.4 operator=() [2/2]

```
Foods& Foods::operator= (
            Foods && stone )   [default]
```

Assign a Foods object by move.

Parameters

stone	

Returns

Foods&

6.27.4 Field Documentation

6.27.4.1 pos

```
sf::Vector2f Foods::pos  [private]
```

6.27.4.2 rect

```
sf::IntRect Foods::rect [private]
```

6.27.4.3 type

```
TypeObject Foods::type [private]
```

Get The Sprite of the object.

Returns

The Sprite of the object

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

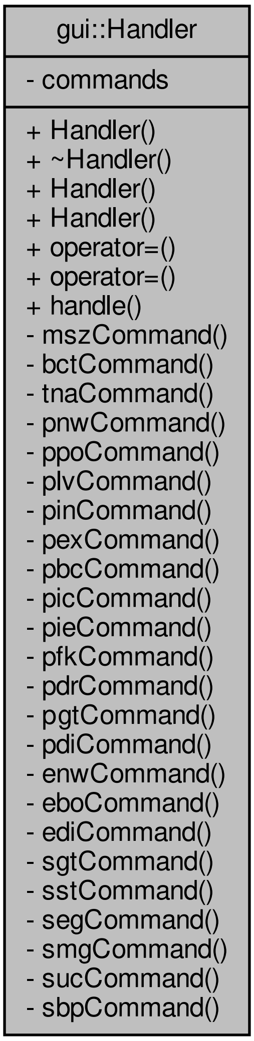
- Objects/Foods/Foods.hpp
- Objects/Foods/Foods.cpp

6.28 gui::Handler Class Reference

Handler class.

```
#include <Handler.hpp>
```

Collaboration diagram for gui::Handler:



Public Member Functions

- Handler ()=default

  Construct a new Handler object.
- ∼Handler ()=default

  Destroy the Handler object.
- Handler (const Handler &)=delete

  Construct a new Handler object by copy.
- Handler (Handler &&)=default

  Construct a new Handler object by move.
- Handler & operator= (const Handler &)=delete

  Assign a Handler object by copy.

- Handler & operator= (Handler &&)=default

  Assign a Handler object by move.
- void handle (const std::string &keyword, gui::Data &game, std::istringstream &iss)

  handle the command

Static Private Member Functions

- static void mszCommand (std::istringstream &iss, gui::Data &game)

  msz command
- static void bctCommand (std::istringstream &iss, gui::Data &game)

  bct command
- static void tnaCommand (std::istringstream &iss, gui::Data &game)

  tna command
- static void pnwCommand (std::istringstream &iss, gui::Data &game)

  pnw command
- static void ppoCommand (std::istringstream &iss, gui::Data &game)

  ppo command
- static void plvCommand (std::istringstream &iss, gui::Data &game)

  plv command
- static void pinCommand (std::istringstream &iss, gui::Data &game)

  pin command
- static void pexCommand (std::istringstream &iss, gui::Data &game)

  pex command
- static void pbcCommand (std::istringstream &iss, gui::Data &game)

  pbc command
- static void picCommand (std::istringstream &iss, gui::Data &game)

  pic command
- static void pieCommand (std::istringstream &iss, gui::Data &game)

  pie command
- static void pfkCommand (std::istringstream &iss, gui::Data &game)

  pfk command
- static void pdrCommand (std::istringstream &iss, gui::Data &game)

  pdr command
- static void pgtCommand (std::istringstream &iss, gui::Data &game)

  pgt command
- static void pdiCommand (std::istringstream &iss, gui::Data &game)

  pdi command
- static void enwCommand (std::istringstream &iss, gui::Data &game)

  enw command
- static void eboCommand (std::istringstream &iss, gui::Data &game)

  ebo command
- static void ediCommand (std::istringstream &iss, gui::Data &game)

  edi command
- static void sgtCommand (std::istringstream &iss, gui::Data &game)

  sgt command
- static void sstCommand (std::istringstream &iss, gui::Data &game)

  sst command
- static void segCommand (std::istringstream &iss, gui::Data &game)

  seg command
- static void smgCommand (std::istringstream &iss, gui::Data &game)

  smg command
- static void sucCommand (std::istringstream &iss, gui::Data &game)
  suc command
- static void sbpCommand (std::istringstream &iss, gui::Data &game)
  sbp command

Private Attributes

- std::vector< Command > commands

6.28.1 Detailed Description

Handler class.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 Handler() [1/3]

```
gui::Handler::Handler ( )  [default]
```

Construct a new Handler object.

6.28.2.2 ∼Handler()

```
gui::Handler::∼Handler ( )  [default]
```

Destroy the Handler object.

6.28.2.3 Handler() [2/3]

```
gui::Handler::Handler (
            const Handler &  )  [delete]
```

Construct a new Handler object by copy.

6.28.2.4 Handler() [3/3]

```
gui::Handler::Handler (
            Handler && )  [default]
```

Construct a new Handler object by move.

6.28.3 Member Function Documentation

6.28.3.1 bctCommand()

```
void gui::Handler::bctCommand (
            std::istringstream & iss,
            gui::Data & game )  [static], [private]
```

bct command

Parameters

iss	
game	

6.28.3.2 eboCommand()

```
void gui::Handler::eboCommand (
            std::istringstream & iss,
            gui::Data & game )  [static], [private]
```

ebo command

Parameters

iss	
game	

6.28.3.3 ediCommand()

```
void gui::Handler::ediCommand (
            std::istringstream & iss,
            gui::Data & game )  [static], [private]
```

edi command

Parameters

iss	
game	

6.28.3.4 enwCommand()

```
void gui::Handler::enwCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

enw command

Parameters

iss	
game	

6.28.3.5 handle()

```
void gui::Handler::handle (
             const std::string & keyword,
             gui::Data & game,
             std::istringstream & iss )
```

handle the command

Parameters

keyword	
game	
iss	

6.28.3.6 mszCommand()

```
void gui::Handler::mszCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

msz command

Parameters

iss	
game	

6.28.3.7 operator=() [1/2]

```
Handler& gui::Handler::operator= (
             const Handler & )  [delete]
```

Assign a Handler object by copy.

Returns

Handler&

6.28.3.8 operator=() [2/2]

```
Handler& gui::Handler::operator= (
             Handler && )  [default]
```

Assign a Handler object by move.

Returns

Handler&

6.28.3.9 pbcCommand()

```
void gui::Handler::pbcCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

pbc command

Parameters

iss	
game	

6.28.3.10 pdiCommand()

```
void gui::Handler::pdiCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

pdi command

Parameters

iss	
game	

6.28.3.11 pdrCommand()

```
void gui::Handler::pdrCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

pdr command

Parameters

iss	
game	

6.28.3.12 pexCommand()

```
void gui::Handler::pexCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

pex command

Parameters

iss	
game	

6.28.3.13 pfkCommand()

```
void gui::Handler::pfkCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

pfk command

Parameters

iss	
game	

6.28.3.14 pgtCommand()

```
void gui::Handler::pgtCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

pgt command

Parameters

iss	
game	

6.28.3.15 picCommand()

```
void gui::Handler::picCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

pic command

Parameters

iss	
game	

6.28.3.16 pieCommand()

```
void gui::Handler::pieCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

pie command

Parameters

iss	
game	

6.28.3.17 pinCommand()

```
void gui::Handler::pinCommand (
              std::istringstream & iss,
              gui::Data & game )  [static], [private]
```

pin command

Parameters

iss	
game	

6.28.3.18 plvCommand()

```
void gui::Handler::plvCommand (
              std::istringstream & iss,
              gui::Data & game )  [static], [private]
```

plv command

Parameters

iss	
game	

6.28.3.19 pnwCommand()

```
void gui::Handler::pnwCommand (
              std::istringstream & iss,
              gui::Data & game )  [static], [private]
```

pnw command

Parameters

iss	
game	

6.28.3.20 ppoCommand()

```
void gui::Handler::ppoCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

ppo command

Parameters

iss	
game	

6.28.3.21 sbpCommand()

```
void gui::Handler::sbpCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

sbp command

Parameters

iss	
game	

6.28.3.22 segCommand()

```
void gui::Handler::segCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

seg command

Parameters

iss	
game	

6.28.3.23 sgtCommand()

```
void gui::Handler::sgtCommand (
            std::istringstream & iss,
            gui::Data & game )  [static], [private]
```

sgt command

Parameters

iss	
game	

6.28.3.24 smgCommand()

```
void gui::Handler::smgCommand (
            std::istringstream & iss,
            gui::Data & game )  [static], [private]
```

smg command

Parameters

iss	
game	

6.28.3.25 sstCommand()

```
void gui::Handler::sstCommand (
            std::istringstream & iss,
            gui::Data & game )  [static], [private]
```

sst command

Parameters

iss	
game	

6.28.3.26 sucCommand()

```
void gui::Handler::sucCommand (
            std::istringstream & iss,
            gui::Data & game )  [static], [private]
```

suc command

Parameters

iss	
game	

6.28.3.27 tnaCommand()

```
void gui::Handler::tnaCommand (
             std::istringstream & iss,
             gui::Data & game )  [static], [private]
```

tna command

Parameters

iss	
game	

6.28.4 Field Documentation

6.28.4.1 commands

```
std::vector< Command > gui::Handler::commands  [private]
```

Initial value:

```
= {
                    {"msz", mszCommand},
                    {"bct", bctCommand},
                    {"tna", tnaCommand},
                    {"pnw", pnwCommand},
                    {"ppo", ppoCommand},
                    {"plv", plvCommand},
                    {"pin", pinCommand},
                    {"pex", pexCommand},
                    {"pbc", pbcCommand},
                    {"pic", picCommand},
                    {"pie", pieCommand},
                    {"pfk", pfkCommand},
                    {"pdr", pdrCommand},
                    {"pgt", pgtCommand},
                    {"pdi", pdiCommand},
                    {"enw", enwCommand},
                    {"ebo", eboCommand},
                    {"edi", ediCommand},
                    {"sgt", sgtCommand},
                    {"sst", sstCommand},
                    {"seg", segCommand},
                    {"smg", smgCommand},
                    {"suc", sucCommand},
                    {"sbp", sbpCommand},
            }
```

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

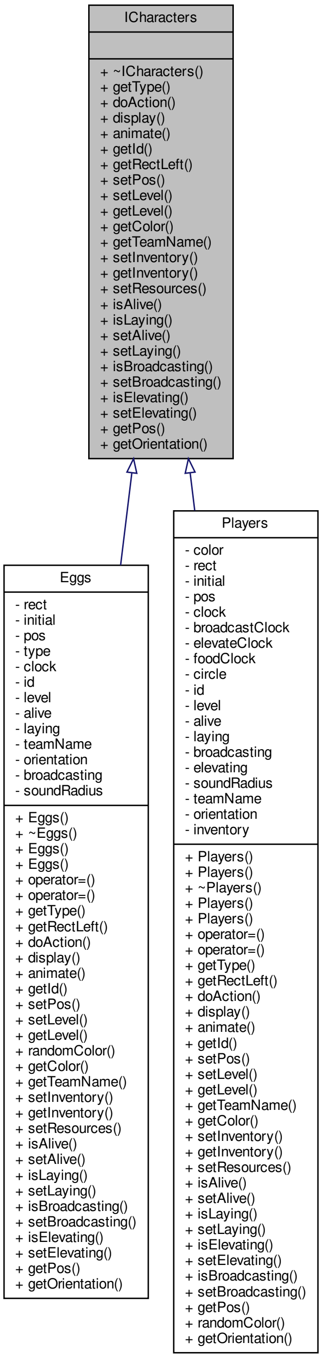
- Network/Handlers/Handler.hpp
- Network/Handlers/Commands/bct.cpp
- Network/Handlers/Commands/ebo.cpp
- Network/Handlers/Commands/edi.cpp
- Network/Handlers/Commands/enw.cpp
- Network/Handlers/Commands/msz.cpp
- Network/Handlers/Commands/pbc.cpp
- Network/Handlers/Commands/pdi.cpp
- Network/Handlers/Commands/pdr.cpp
- Network/Handlers/Commands/pex.cpp
- Network/Handlers/Commands/pfk.cpp
- Network/Handlers/Commands/pgt.cpp
- Network/Handlers/Commands/pic.cpp
- Network/Handlers/Commands/pie.cpp
- Network/Handlers/Commands/pin.cpp
- Network/Handlers/Commands/plv.cpp
- Network/Handlers/Commands/pnw.cpp
- Network/Handlers/Commands/ppo.cpp
- Network/Handlers/Commands/sbp.cpp
- Network/Handlers/Commands/seg.cpp
- Network/Handlers/Commands/sgt.cpp
- Network/Handlers/Commands/smg.cpp
- Network/Handlers/Commands/sst.cpp
- Network/Handlers/Commands/suc.cpp
- Network/Handlers/Commands/tna.cpp
- Network/Handlers/Handler.cpp

6.29 ICharacters Class Reference

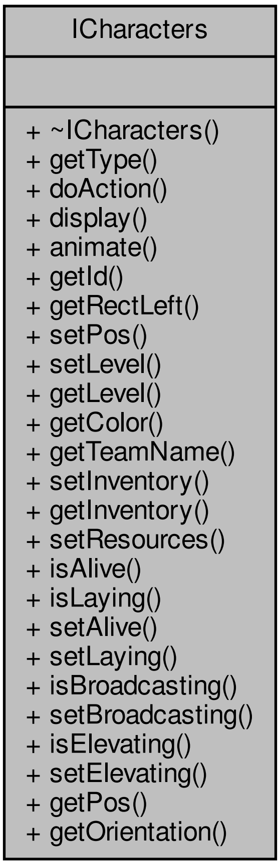
Interface for the characters.

```
#include <ICharacters.hpp>
```

Inheritance diagram for ICharacters:



Collaboration diagram for ICharacters:



Public Member Functions

- virtual ~ICharacters () noexcept=default

  Destroy of the ICharacters object.
- virtual CharacterType getType () const =0

  Get the Type object.
- virtual void doAction ()=0

  Do specific action for the character.
- virtual void display (sf::RenderWindow &window, sf::Sprite &sprite, gui::ItemDrawer &drawer, int timeUnit, SoundBox &soundBox)=0

  Display the character.
- virtual void animate (sf::Sprite &sprite)=0

  Animate the character.
- virtual const int & getId ()=0

  Get the Id of the character.
- virtual const int & getRectLeft ()=0

  Get the Rect Left of the character.

- virtual void setPos (sf::Vector2f pos, Orientation orientation)=0

  Set the Pos of the character.
- virtual void setLevel (int level)=0

  Set the Level of the character.
- virtual const int & getLevel ()=0

  Get the Level of the character.
- virtual const std::string & getColor ()=0

  Get the Color of the character.
- virtual const std::string & getTeamName ()=0

  Get the Team Name of the character.
- virtual void setInventory (const std::array< int, 7 > &res)=0

  Set the Inventory of the character.
- virtual const std::array< int, 7 > & getInventory ()=0

  Get the Inventory of the character.
- virtual void setResources (const int &res, const int &nb)=0

  Set the Resources of the character.
- virtual const bool & isAlive ()=0

  Check if the character is alive.
- virtual const bool & isLaying ()=0

  Check if the character is laying.
- virtual void setAlive (bool alive)=0

  Set the Alive boolean of the character.
- virtual void setLaying (bool laying)=0

  Set the Laying boolean of the character.
- virtual const bool & isBroadcasting ()=0

  Check if the character is broadcasting.
- virtual void setBroadcasting (bool broadcasting)=0

  Set the Broadcasting boolean of the character.
- virtual const int & isElevating ()=0

  Check if the character is elevating.
- virtual void setElevating (int elevating)=0

  Set the Elevating process of the character.
- virtual const sf::Vector2f & getPos ()=0

  Get the Pos of the character.
- virtual const Orientation & getOrientation ()=0

  Get the Orientation of the character.

6.29.1 Detailed Description

Interface for the characters.

6.29.2 Constructor & Destructor Documentation

6.29.2.1 ∼ICharacters()

```
virtual ICharacters::∼ICharacters ( )  [virtual], [default], [noexcept]
```

Destroy of the ICharacters object.

6.29.3 Member Function Documentation

6.29.3.1 animate()

```
virtual void ICharacters::animate (
             sf::Sprite & sprite )  [pure virtual]
```

Animate the character.

Parameters

sprite	

Implemented in Players, and Eggs.

6.29.3.2 display()

```
virtual void ICharacters::display (
             sf::RenderWindow & window,
             sf::Sprite & sprite,
             gui::ItemDrawer & drawer,
             int timeUnit,
             SoundBox & soundBox )  [pure virtual]
```

Display the character.

Parameters

window	
sprite	
drawer	
timeUnit	
soundbox	

Implemented in Players, and Eggs.

6.29.3.3 doAction()

```
virtual void ICharacters::doAction ( )  [pure virtual]
```

Do specific action for the character.

Implemented in Players, and Eggs.

6.29.3.4 getColor()

```
virtual const std::string& ICharacters::getColor ( )  [pure virtual]
```

Get the Color of the character.

Returns

const std::string&

Implemented in Players, and Eggs.

6.29.3.5 getId()

```
virtual const int& ICharacters::getId ( )  [pure virtual]
```

Get the Id of the character.

Returns

const int&

Implemented in Players, and Eggs.

6.29.3.6 getInventory()

```
virtual const std::array<int, 7>& ICharacters::getInventory ( )  [pure virtual]
```

Get the Inventory of the character.

Returns

std::array<int, 7>&

Implemented in Players, and Eggs.

6.29.3.7 getLevel()

```
virtual const int& ICharacters::getLevel ( )  [pure virtual]
```

Get the Level of the character.

Returns

const int&

Implemented in Players, and Eggs.

6.29.3.8 getOrientation()

```
virtual const Orientation& ICharacters::getOrientation ( )  [pure virtual]
```

Get the Orientation of the character.

Returns

const Orientation&

Implemented in Players, and Eggs.

6.29.3.9 getPos()

```
virtual const sf::Vector2f& ICharacters::getPos ( )  [pure virtual]
```

Get the Pos of the character.

Returns

sf::Vector2f

Implemented in Players, and Eggs.

6.29.3.10 getRectLeft()

```
virtual const int& ICharacters::getRectLeft ( )  [pure virtual]
```

Get the Rect Left of the character.

Returns

const int&

Implemented in Players, and Eggs.

6.29.3.11 getTeamName()

```
virtual const std::string& ICharacters::getTeamName ( )  [pure virtual]
```

Get the Team Name of the character.

Returns

const std::string&

Implemented in Players, and Eggs.

6.29.3.12 getType()

```
virtual CharacterType ICharacters::getType ( ) const  [pure virtual]
```

Get the Type object.

Returns

CharacterType

Implemented in Players, and Eggs.

6.29.3.13 isAlive()

```
virtual const bool& ICharacters::isAlive ( )  [pure virtual]
```

Check if the character is alive.

Returns

true

false

Implemented in Players, and Eggs.

6.29.3.14 isBroadcasting()

```
virtual const bool& ICharacters::isBroadcasting ( )  [pure virtual]
```

Check if the character is broadcasting.

Returns

true

false

Implemented in Players, and Eggs.

6.29.3.15 isElevating()

```
virtual const int& ICharacters::isElevating ( ) [pure virtual]
```

Check if the character is elevating.

Returns

int

Implemented in Players, and Eggs.

6.29.3.16 isLaying()

```
virtual const bool& ICharacters::isLaying ( ) [pure virtual]
```

Check if the character is laying.

Returns

true

false

Implemented in Players, and Eggs.

6.29.3.17 setAlive()

```
virtual void ICharacters::setAlive (
             bool alive ) [pure virtual]
```

Set the Alive boolean of the character.

Parameters

alive	

Implemented in Players, and Eggs.

6.29.3.18 setBroadcasting()

```
virtual void ICharacters::setBroadcasting (
             bool broadcasting ) [pure virtual]
```

Set the Broadcasting boolean of the character.

Parameters

broadcasting	

Implemented in Eggs, and Players.

6.29.3.19 setElevating()

```
virtual void ICharacters::setElevating (
            int elevating )  [pure virtual]
```

Set the Elevating process of the character.

Parameters

elevating	

Implemented in Players, and Eggs.

6.29.3.20 setInventory()

```
virtual void ICharacters::setInventory (
            const std::array< int, 7 > & res )  [pure virtual]
```

Set the Inventory of the character.

Parameters

res	

Implemented in Players, and Eggs.

6.29.3.21 setLaying()

```
virtual void ICharacters::setLaying (
            bool laying )  [pure virtual]
```

Set the Laying boolean of the character.

Parameters

laying	

Implemented in Players, and Eggs.

6.29.3.22 setLevel()

```
virtual void ICharacters::setLevel (
             int level )  [pure virtual]
```

Set the Level of the character.

Parameters

level	

Implemented in Players, and Eggs.

6.29.3.23 setPos()

```
virtual void ICharacters::setPos (
             sf::Vector2f pos,
             Orientation orientation )  [pure virtual]
```

Set the Pos of the character.

Parameters

pos	
orientation	

Implemented in Players, and Eggs.

6.29.3.24 setResources()

```
virtual void ICharacters::setResources (
             const int & res,
             const int & nb )  [pure virtual]
```

Set the Resources of the character.

Parameters

res	
nb	

Implemented in Players, and Eggs.

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

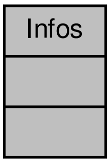
- Characters/ICharacters.hpp

6.30 Infos Struct Reference

Infos struct.

```
#include <Tile.hpp>
```

Collaboration diagram for Infos:



6.30.1 Detailed Description

Infos struct.

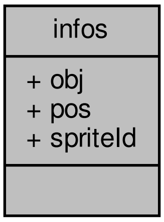
The documentation for this struct was generated from the following file:

- Map/Tile.hpp

6.31 infos Struct Reference

```
#include <Tile.hpp>
```

Collaboration diagram for infos:



Data Fields

- int obj
- sf::Vector2f pos
- int spriteId

6.31.1 Field Documentation

6.31.1.1 obj

```
int infos::obj
```

6.31.1.2 pos

```
sf::Vector2f infos::pos
```

6.31.1.3 spriteId

```
int infos::spriteId
```

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

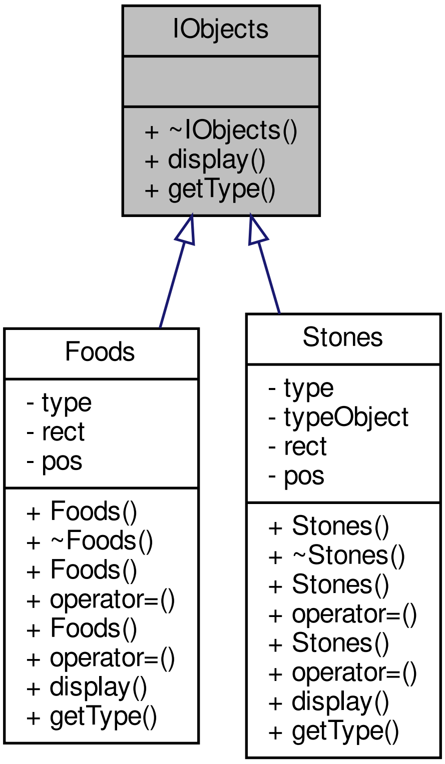
- Map/Tile.hpp

6.32 IObjects Class Reference

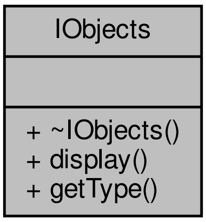
IObjects class.

```
#include <IObjects.hpp>
```

Inheritance diagram for IObjects:



Collaboration diagram for IObjects:



Public Member Functions

- virtual ~IObjects () noexcept=default

  Destroy the IObjects object.

- virtual void display (sf::RenderWindow &window, sf::Sprite &sprite, float size)=0

    Display the object.
- virtual const TypeObject & getType ()=0

    Get The type of the object.

6.32.1 Detailed Description

IObjects class.

6.32.2 Constructor & Destructor Documentation

6.32.2.1 ~IObjects()

```
virtual IObjects::~IObjects ( )  [virtual], [default], [noexcept]
```

Destroy the IObjects object.

6.32.3 Member Function Documentation

6.32.3.1 display()

```
virtual void IObjects::display (
             sf::RenderWindow & window,
             sf::Sprite & sprite,
             float size )  [pure virtual]
```

Display the object.

Parameters

window	The window where the object will be displayed
sprite	The sprite of the object
size	The size of the object

Implemented in Stones, and Foods.

6.32.3.2 getType()

```
virtual const TypeObject& IObjects::getType ( )  [pure virtual]
```

Get The type of the object.

Returns

The type of the object

Implemented in Stones, and Foods.

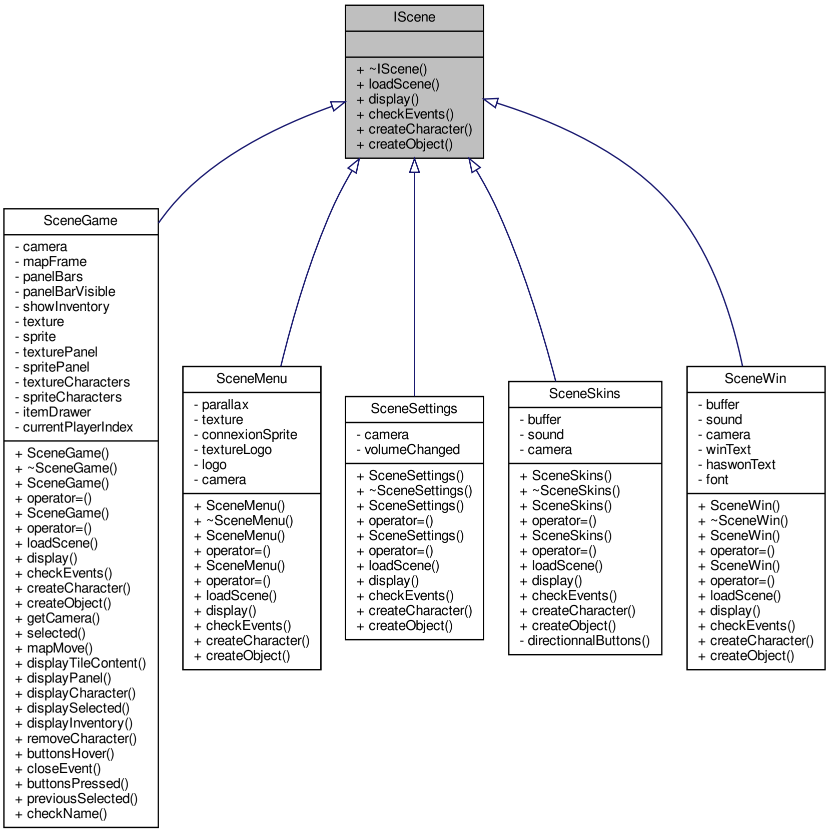
The documentation for this class was generated from the following file:

- Objects/IObjects.hpp

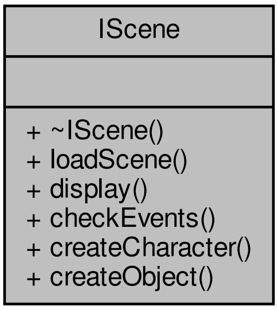
6.33 IScene Class Reference

```
#include <IScene.hpp>
```

Inheritance diagram for IScene:



Collaboration diagram for IScene:



Public Member Functions

- virtual ∼IScene () noexcept=default
- virtual void loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data)=0
- virtual void display (gui::SceneManager &manager, gui::Window &window, gui::Data &data)=0
- virtual void checkEvents (gui::SceneManager &manager, gui::Window &window, gui::Data &data)=0
- virtual void createCharacter ()=0
- virtual void createObject (gui::Data &data)=0

6.33.1 Constructor & Destructor Documentation

6.33.1.1 ∼IScene()

```
virtual IScene::∼IScene ( )  [virtual], [default], [noexcept]
```

6.33.2 Member Function Documentation

6.33.2.1 checkEvents()

```
virtual void IScene::checkEvents (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )  [pure virtual]
```

Implemented in SceneWin, SceneSkins, SceneSettings, SceneMenu, and SceneGame.

6.33.2.2 createCharacter()

```
virtual void IScene::createCharacter ( )  [pure virtual]
```

Implemented in SceneWin, SceneSkins, SceneSettings, SceneMenu, and SceneGame.

6.33.2.3 createObject()

```
virtual void IScene::createObject (
             gui::Data & data )  [pure virtual]
```

Implemented in SceneGame.

6.33.2.4 display()

```
virtual void IScene::display (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )  [pure virtual]
```

Implemented in SceneWin, SceneSkins, SceneSettings, SceneMenu, and SceneGame.

6.33.2.5 loadScene()

```
virtual void IScene::loadScene (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )  [pure virtual]
```

Implemented in SceneWin, SceneSkins, SceneSettings, SceneMenu, and SceneGame.

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

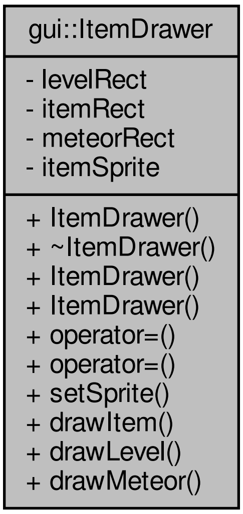
- Scene/IScene.hpp

6.34 gui::ItemDrawer Class Reference

ItemDrawer class.

```
#include <ItemDrawer.hpp>
```

Collaboration diagram for gui::ItemDrawer:



Public Member Functions

- ItemDrawer ()=default

  Construct a new Item Drawer object.
- ∼ItemDrawer ()=default

  Destroy the Item Drawer object.
- ItemDrawer (const ItemDrawer &other)=delete

  Construct a new Item Drawer object by copy = deleted.
- ItemDrawer (ItemDrawer &&other)=default

  Construct a new Item Drawer object by move = default.
- ItemDrawer & operator= (const ItemDrawer &other)=delete

  Assign a Item Drawer object by copy = deleted.
- ItemDrawer & operator= (ItemDrawer &&other)=default

  Assign a Item Drawer object by move = default.
- void setSprite (sf::Sprite sprite)

  Set the Sprite.
- void drawItem (sf::RenderWindow &window, int id, int x, int y, float scale)

  draw the item
- void drawLevel (sf::RenderWindow &window, int id, int x, int y, float scale)

  draw the level
- void drawMeteor (sf::RenderWindow &window, int id, int x, int y, float scale)

  draw the Meteor

Private Attributes

- std::vector< sf::IntRect > levelRect
- std::vector< sf::IntRect > itemRect
- std::vector< sf::IntRect > meteorRect
- sf::Sprite itemSprite

6.34.1 Detailed Description

ItemDrawer class.

6.34.2 Constructor & Destructor Documentation

6.34.2.1 ItemDrawer() [1/3]

```
gui::ItemDrawer::ItemDrawer ( )  [default]
```

Construct a new Item Drawer object.

6.34.2.2 ∼ItemDrawer()

```
gui::ItemDrawer::∼ItemDrawer ( )  [default]
```

Destroy the Item Drawer object.

6.34.2.3 ItemDrawer() [2/3]

```
gui::ItemDrawer::ItemDrawer (
              const ItemDrawer & other )  [delete]
```

Construct a new Item Drawer object by copy = deleted.

Parameters

other	

6.34.2.4 ItemDrawer() [3/3]

```
gui::ItemDrawer::ItemDrawer (
             ItemDrawer && other )  [default]
```

Construct a new Item Drawer object by move = default.

Parameters

other	

Returns

ItemDrawer&

6.34.3 Member Function Documentation

6.34.3.1 drawItem()

```
void gui::ItemDrawer::drawItem (
             sf::RenderWindow & window,
             int id,
             int x,
             int y,
             float scale )
```

draw the item

Parameters

window	
id	
x	
y	
scale	

6.34.3.2 drawLevel()

```
void gui::ItemDrawer::drawLevel (
             sf::RenderWindow & window,
             int id,
             int x,
             int y,
             float scale )
```

draw the level

Parameters

window	
id	
x	
y	
scale	

6.34.3.3 drawMeteor()

```
void gui::ItemDrawer::drawMeteor (
             sf::RenderWindow & window,
             int id,
             int x,
             int y,
             float scale )
```

draw the Meteor

Parameters

window	
id	
x	
y	
scale	

6.34.3.4 operator=() [1/2]

```
ItemDrawer& gui::ItemDrawer::operator= (
             const ItemDrawer & other )  [delete]
```

Assign a Item Drawer object by copy = deleted.

Parameters

other	

Returns

ItemDrawer&

6.34.3.5 operator=() [2/2]

```
ItemDrawer& gui::ItemDrawer::operator= (
             ItemDrawer && other )  [default]
```

Assign a Item Drawer object by move = default.

Parameters

other	

Returns

ItemDrawer&

6.34.3.6 setSprite()

```
void gui::ItemDrawer::setSprite (
             sf::Sprite sprite )
```

Set the Sprite.

Parameters

sprite	

6.34.4 Field Documentation

6.34.4.1 itemRect

```
std::vector< sf::IntRect > gui::ItemDrawer::itemRect  [private]
```

Initial value:

```
= {
              sf::IntRect(800, 1560, 75, 95),
              sf::IntRect(795, 1665, 75, 80),
              sf::IntRect(960, 1670, 80, 70),
              sf::IntRect(1190, 1670, 75, 80),
              sf::IntRect(1040, 1675, 75, 80),
              sf::IntRect(1115, 1675, 65, 80),
              sf::IntRect(885, 1660, 65, 80),
              sf::IntRect(800, 1560, 75, 95),
              sf::IntRect(895, 1560, 75, 95),
              sf::IntRect(995, 1560, 75, 95),
              sf::IntRect(1095, 1560, 75, 95),
              sf::IntRect(1200, 1560, 75, 95),
              sf::IntRect(1300, 1560, 75, 95),
              sf::IntRect(1400, 1560, 75, 95),
              sf::IntRect(1500, 1560, 75, 95),
              sf::IntRect(1610, 1560, 75, 95),
              sf::IntRect(1720, 1560, 75, 95),
              sf::IntRect(1815, 1560, 75, 95),
              sf::IntRect(1920, 1560, 75, 95),
              sf::IntRect(2025, 1560, 75, 95),
              sf::IntRect(2120, 1560, 85, 95),
              sf::IntRect(2230, 1560, 75, 95),
              sf::IntRect(2330, 1560, 75, 95),
              sf::IntRect(2432, 1560, 85, 95)
          }
```

6.34.4.2 itemSprite

```
sf::Sprite gui::ItemDrawer::itemSprite [private]
```

6.34.4.3 levelRect

```
std::vector< sf::IntRect > gui::ItemDrawer::levelRect [private]
```

Initial value:

```
= {
          sf::IntRect(2215, 50, 170, 130),
          sf::IntRect(2410, 50, 170, 130),
          sf::IntRect(2610, 50, 170, 130),
          sf::IntRect(2310, 190, 170, 130),
          sf::IntRect(2520, 190, 170, 130),
          sf::IntRect(2215, 320, 170, 130),
          sf::IntRect(2410, 320, 170, 130),
          sf::IntRect(2610, 320, 170, 130)
        }
```

6.34.4.4 meteorRect

```
std::vector< sf::IntRect > gui::ItemDrawer::meteorRect [private]
```

Initial value:

```
= {
            sf::IntRect(2775, 1265, 345, 255),
            sf::IntRect(3150, 1330, 360, 225),
            sf::IntRect(2710, 1575, 100, 95),
            sf::IntRect(2815, 1575, 115, 85),
            sf::IntRect(2950, 1570, 90, 110)
        }
```

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

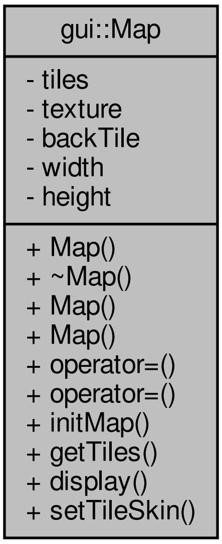
- ItemDrawer/ItemDrawer.hpp
- ItemDrawer/ItemDrawer.cpp

6.35 gui::Map Class Reference

Map Class.

`#include <Map.hpp>`

Collaboration diagram for gui::Map:



Public Member Functions

- Map ()=default

    Construct a new Map object.
- ∼Map ()=default

    Destroy the Map object.
- Map (const Map &other)=delete

    Construct a new Map object by copy.
- Map (Map &&other)=default

    Construct a new Map object.
- Map & operator= (const Map &other)=delete

    Assign a Map object by copy.
- Map & operator= (Map &&other)=default

    Assign a Map object by move.
- void initMap (unsigned int width, unsigned int height)

    Init the map.
- std::vector< std::vector< Tile > > & getTiles ()

    Get the tiles.
- void display (sf::RenderWindow &window, gui::ItemDrawer &itemDrawer)

    Display the map.
- void setTileSkin (int tileSkin)

    Set the Tile Skin.

Private Attributes

- std::vector< std::vector< Tile > > tiles
- sf::Texture texture
- sf::Sprite backTile
- unsigned int width {0}
- unsigned int height {0}

6.35.1 Detailed Description

Map Class.

6.35.2 Constructor & Destructor Documentation

6.35.2.1 Map() [1/3]

```
gui::Map::Map ( )  [default]
```

Construct a new Map object.

6.35.2.2 ∼Map()

```
gui::Map::∼Map ( )  [default]
```

Destroy the Map object.

6.35.2.3 Map() [2/3]

```
gui::Map::Map (
             const Map & other )  [delete]
```

Construct a new Map object by copy.

Parameters

other	

6.35.2.4 Map() [3/3]

```
gui::Map::Map (
            Map && other )  [default]
```

Construct a new Map object.

Parameters

other	

6.35.3 Member Function Documentation

6.35.3.1 display()

```
void gui::Map::display (
            sf::RenderWindow & window,
            gui::ItemDrawer & itemDrawer )
```

Display the map.

Parameters

window	The window to display the map
itemDrawer	The itemDrawer to display the map

6.35.3.2 getTiles()

```
std::vector< std::vector< Tile > > & gui::Map::getTiles ( )
```

Get the tiles.

Returns

const std::vector<std::vector<Tile>> &

6.35.3.3 initMap()

```
void gui::Map::initMap (
            unsigned int width,
            unsigned int height )
```

Init the map.

Parameters

width	
height	

Exceptions

Errors::ErrorTexture	

6.35.3.4 operator=() [1/2]

```
Map& gui::Map::operator= (
             const Map & other )  [delete]
```

Assign a Map object by copy.

Parameters

other	

Returns

Map&

6.35.3.5 operator=() [2/2]

```
Map& gui::Map::operator= (
             Map && other )  [default]
```

Assign a Map object by move.

Parameters

other	

Returns

Map&

6.35.3.6 setTileSkin()

```
void gui::Map::setTileSkin (
             int tileSkin )
```

Set the Tile Skin.

Parameters

tileSkin	

6.35.4 Field Documentation

6.35.4.1 backTile

```
sf::Sprite gui::Map::backTile  [private]
```

6.35.4.2 height

```
unsigned int gui::Map::height {0}  [private]
```

6.35.4.3 texture

```
sf::Texture gui::Map::texture  [private]
```

6.35.4.4 tiles

```
std::vector<std::vector<Tile> > gui::Map::tiles  [private]
```

6.35.4.5 width

```
unsigned int gui::Map::width {0}  [private]
```

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

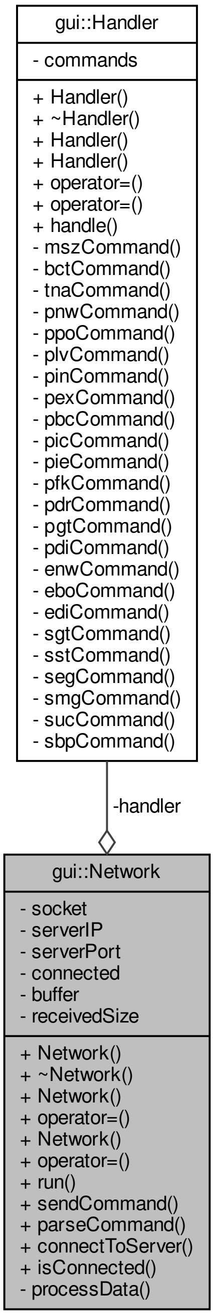
- Map/Map.hpp
- Map/Map.cpp

6.36 gui::Network Class Reference

Network class.

```
#include <Network.hpp>
```

Collaboration diagram for gui::Network:



Public Member Functions

- Network (unsigned int port, const char ∗path)

    Construct a new Network object.
- ∼Network ()

    Destroy the Network object.
- Network (Network &&)=default

    Construct a new Network object by move.
- Network & operator= (Network &&)=default

    Assign a Network object by move.
- Network (const Network &)=delete

    Construct a new Network object.
- Network & operator= (const Network &)=delete

    Assign a Network object by copy.
- void run (gui::Data &game)

    run the network
- int sendCommand (const std::string &command)

    send a command to the server
- void parseCommand (const std::string &command, gui::Data &game)

    parse the command
- void connectToServer ()

    connect to the server
- bool isConnected ()

    Check if socket is disconnect.

Private Member Functions

- void processData (const char ∗data, std::size_t size)

    process the data

Private Attributes

- sf::TcpSocket socket
- sf::IpAddress serverIP
- unsigned int serverPort
- bool connected = false
- char buffer [maxBufferSize] {0}
- std::size_t receivedSize {0}
- gui::Handler handler

6.36.1 Detailed Description

Network class.

6.36.2 Constructor & Destructor Documentation

6.36.2.1 Network() [1/3]

```
gui::Network::Network (
            unsigned int port,
            const char * path )
```

Construct a new Network object.

Parameters

port	
path	

6.36.2.2 ∼Network()

```
gui::Network::∼Network ( )
```

Destroy the Network object.

6.36.2.3 Network() [2/3]

```
gui::Network::Network (
            Network && ) [default]
```

Construct a new Network object by move.

6.36.2.4 Network() [3/3]

```
gui::Network::Network (
            const Network & ) [delete]
```

Construct a new Network object.

6.36.3 Member Function Documentation

6.36.3.1 connectToServer()

```
void gui::Network::connectToServer ( )
```

connect to the server

Exceptions

Errors::ErrorNetwork	

6.36.3.2 isConnected()

```
bool gui::Network::isConnected ( )
```

Check if socket is disconnect.

6.36.3.3 operator=() [1/2]

```
Network& gui::Network::operator= (
             const Network & )  [delete]
```

Assign a Network object by copy.

Returns

Network&

6.36.3.4 operator=() [2/2]

```
Network& gui::Network::operator= (
             Network && )  [default]
```

Assign a Network object by move.

Returns

Network&

6.36.3.5 parseCommand()

```
void gui::Network::parseCommand (
             const std::string & command,
             gui::Data & game )
```

parse the command

Parameters

command	
game	

6.36.3.6 processData()

```
void gui::Network::processData (
             const char * data,
             std::size_t size )  [private]
```

process the data

Parameters

data	
size	

6.36.3.7 run()

```
void gui::Network::run (
             gui::Data & game )
```

run the network

Parameters

game	

6.36.3.8 sendCommand()

```
int gui::Network::sendCommand (
             const std::string & command )
```

send a command to the server

Parameters

command	

Returns

int

6.36.4 Field Documentation

6.36.4.1 buffer

```
char gui::Network::buffer {0}  [private]
```

6.36.4.2 connected

```
bool gui::Network::connected = false  [private]
```

6.36.4.3 handler

```
gui::Handler gui::Network::handler  [private]
```

6.36.4.4 receivedSize

```
std::size_t gui::Network::receivedSize {0}  [private]
```

6.36.4.5 serverIP

```
sf::IpAddress gui::Network::serverIP  [private]
```

6.36.4.6 serverPort

```
unsigned int gui::Network::serverPort  [private]
```

6.36.4.7 socket

```
sf::TcpSocket gui::Network::socket  [private]
```

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

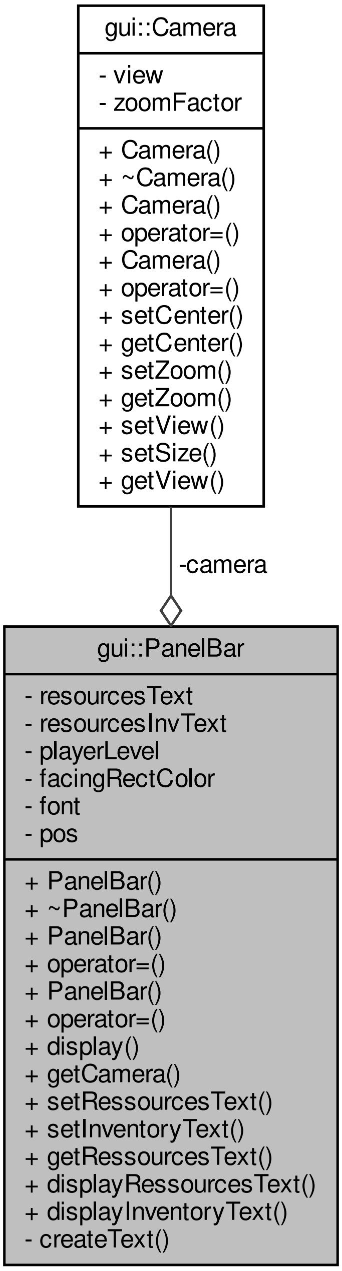
- Network/Network.hpp
- Network/Network.cpp

6.37 gui::PanelBar Class Reference

PanelBar class.

```
#include <PanelBar.hpp>
```

Collaboration diagram for gui::PanelBar:



Public Member Functions

- PanelBar (sf::Vector2f pos)

  Construct a new Panel Bar object.
- ∼PanelBar () noexcept=default

  Destroy the Panel Bar object.
- PanelBar (const PanelBar &panelBar)=delete

  Construct a new Panel Bar object by copy.
- PanelBar & operator= (const PanelBar &panelBar)=delete

  Assign a Panel Bar object by copy.
- PanelBar (PanelBar &&panelBar)=default

  Construct a new Panel Bar object by move.
- PanelBar & operator= (PanelBar &&panelBar)=default

  Assign a Panel Bar object by move.
- void display (sf::RenderWindow &window, sf::Sprite &sprite)
- gui::Camera & getCamera ()
- void setRessourcesText (std::vector< infos > &resources)

  Set the Ressources Text.
- void setInventoryText (ICharacters &character)

  Set the Inventory Text.
- const std::vector< sf::Text > & getRessourcesText () const

  Get the Ressources Text object.
- void displayRessourcesText (gui::Window &window, gui::ItemDrawer &drawer)

  Display the Ressources Text.
- void displayInventoryText (gui::Window &window, gui::ItemDrawer &drawer, sf::Sprite &sprite)

  Display the Inventory Text.

Private Member Functions

- sf::Text createText (const std::string &str, const sf::Vector2f &pos)

  Create a Text object.

Private Attributes

- std::vector< sf::Text > resourcesText
- std::vector< sf::Text > resourcesInvText
- int playerLevel
- int facingRectColor
- sf::Font font
- gui::Camera camera {1920, 1080}
- sf::Vector2f pos

6.37.1 Detailed Description

PanelBar class.

6.37.2 Constructor & Destructor Documentation

6.37.2.1 PanelBar() [1/3]

```
gui::PanelBar::PanelBar (
            sf::Vector2f pos )
```

Construct a new Panel Bar object.

Parameters

pos	

Exceptions

Errors::ErrorTexture	

6.37.2.2 ∼PanelBar()

```
gui::PanelBar::∼PanelBar ( )  [default], [noexcept]
```

Destroy the Panel Bar object.

6.37.2.3 PanelBar() [2/3]

```
gui::PanelBar::PanelBar (
            const PanelBar & panelBar )  [delete]
```

Construct a new Panel Bar object by copy.

Parameters

panelBar	

6.37.2.4 PanelBar() [3/3]

```
gui::PanelBar::PanelBar (
            PanelBar && panelBar )  [default]
```

Construct a new Panel Bar object by move.

Parameters

panelBar	

6.37.3 Member Function Documentation

6.37.3.1 createText()

```
sf::Text gui::PanelBar::createText (
            const std::string & str,
            const sf::Vector2f & pos )  [private]
```

Create a Text object.

Parameters

str	
pos	

Returns

sf::Text

6.37.3.2 display()

```
void gui::PanelBar::display (
            sf::RenderWindow & window,
            sf::Sprite & sprite )
```

6.37.3.3 displayInventoryText()

```
void gui::PanelBar::displayInventoryText (
            gui::Window & window,
            gui::ItemDrawer & drawer,
            sf::Sprite & sprite )
```

Display the Inventory Text.

Parameters

window	
drawer	
sprite	

6.37.3.4 displayRessourcesText()

```
void gui::PanelBar::displayRessourcesText (
            gui::Window & window,
            gui::ItemDrawer & drawer )
```

Display the Ressources Text.

Parameters

window	
drawer	

6.37.3.5 getCamera()

```
gui::Camera & gui::PanelBar::getCamera ( )
```

6.37.3.6 getRessourcesText()

```
const std::vector< sf::Text > & gui::PanelBar::getRessourcesText ( ) const
```

Get the Ressources Text object.

Returns

const std::vector<sf::Text>&

6.37.3.7 operator=() `[1/2]`

```
PanelBar& gui::PanelBar::operator= (
             const PanelBar & panelBar )  [delete]
```

Assign a Panel Bar object by copy.

Parameters

panelBar	

Returns

PanelBar&

6.37.3.8 operator=() `[2/2]`

```
PanelBar& gui::PanelBar::operator= (
             PanelBar && panelBar )  [default]
```

Assign a Panel Bar object by move.

Parameters

panelBar	

Returns

PanelBar&

6.37.3.9 setInventoryText()

```
void gui::PanelBar::setInventoryText (
            ICharacters & character )
```

Set the Inventory Text.

Parameters

character	

6.37.3.10 setRessourcesText()

```
void gui::PanelBar::setRessourcesText (
            std::vector< infos > & resources )
```

Set the Ressources Text.

Parameters

resources	

6.37.4 Field Documentation

6.37.4.1 camera

```
gui::Camera gui::PanelBar::camera {1920, 1080}  [private]
```

6.37.4.2 facingRectColor

```
int gui::PanelBar::facingRectColor  [private]
```

6.37.4.3 font

```
sf::Font gui::PanelBar::font  [private]
```

6.37.4.4 playerLevel

```
int gui::PanelBar::playerLevel  [private]
```

6.37.4.5 pos

```
sf::Vector2f gui::PanelBar::pos  [private]
```

6.37.4.6 resourcesInvText

```
std::vector< sf::Text > gui::PanelBar::resourcesInvText  [private]
```

6.37.4.7 resourcesText

```
std::vector< sf::Text > gui::PanelBar::resourcesText  [private]
```

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

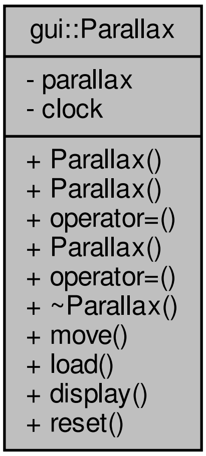
- Components/PanelBar/PanelBar.hpp
- Components/PanelBar/PanelBar.cpp

6.38 gui::Parallax Class Reference

Parallax class.

```
#include <Parallax.hpp>
```

Collaboration diagram for gui::Parallax:



Public Member Functions

- Parallax ()

  Construct a new Parallax object.
- Parallax (const Parallax &other)=delete

  Construct a new Parallax object by copy.
- Parallax & operator= (const Parallax &other)=delete

  Assign a Parallax object by copy.
- Parallax (Parallax &&other)=default

  Construct a new Parallax object by move.
- Parallax & operator= (Parallax &&other)=default

  Assign a Parallax object by move.
- ∼Parallax ()=default

  Destroy the Parallax object.
- void move ()

  Move each layer of the parallax by a certain speed.
- void load ()

  Load the parallax by setting the position of each layer.
- void display (gui::Window &window)

  Display the parallax.
- void reset ()

  Reset the parallax to make it loop.

Private Attributes

- std::vector< std::unique_ptr< Background > > parallax
- sf::Clock clock

6.38.1 Detailed Description

Parallax class.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 Parallax() [1/3]

```
gui::Parallax::Parallax ( )
```

Construct a new Parallax object.

Exceptions

Errors::ErrorTexture	

6.38.2.2 Parallax() [2/3]

```
gui::Parallax::Parallax (
            const Parallax & other )  [delete]
```

Construct a new Parallax object by copy.

Parameters

other	

6.38.2.3 Parallax() [3/3]

```
gui::Parallax::Parallax (
            Parallax && other )  [default]
```

Construct a new Parallax object by move.

Parameters

other	

6.38.2.4 ∼Parallax()

```
gui::Parallax::∼Parallax ( )  [default]
```

Destroy the Parallax object.

6.38.3 Member Function Documentation

6.38.3.1 display()

```
void gui::Parallax::display (
             gui::Window & window )
```

Display the parallax.

Parameters

window	The window where the parallax will be displayed

6.38.3.2 load()

```
void gui::Parallax::load ( )
```

Load the parallax by setting the position of each layer.

6.38.3.3 move()

```
void gui::Parallax::move ( )
```

Move each layer of the parallax by a certain speed.

6.38.3.4 operator=() [1/2]

```
Parallax& gui::Parallax::operator= (
             const Parallax & other )  [delete]
```

Assign a Parallax object by copy.

Parameters

other	

Returns

Parallax&

6.38.3.5 operator=() [2/2]

```
Parallax& gui::Parallax::operator= (
            Parallax && other )  [default]
```

Assign a Parallax object by move.

Parameters

other	

Returns

Parallax&

6.38.3.6 reset()

```
void gui::Parallax::reset ( )
```

Reset the parallax to make it loop.

6.38.4 Field Documentation

6.38.4.1 clock

```
sf::Clock gui::Parallax::clock  [private]
```

6.38.4.2 parallax

```
std::vector<std::unique_ptr<Background> > gui::Parallax::parallax  [private]
```

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

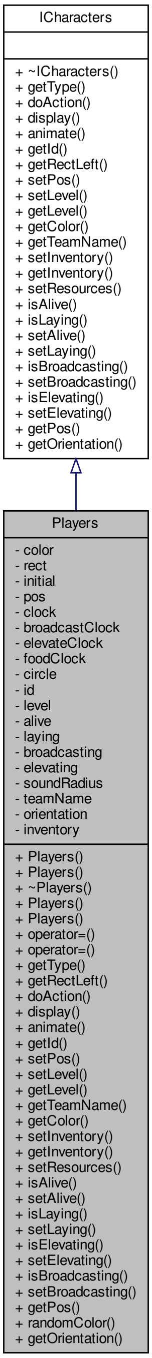
- Components/Parallax/Parallax.hpp
- Components/Parallax/Parallax.cpp

6.39 Players Class Reference

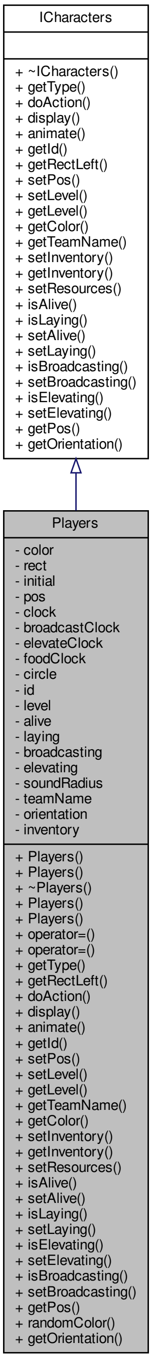
Players class.

```
#include <Players.hpp>
```

Inheritance diagram for Players:



Collaboration diagram for Players:



Public Member Functions

- Players (sf::IntRect rect, sf::Vector2f pos)

  Construct a new Players object.
- Players (sf::Vector2f pos, int id, int level, const std::string &teamName, Orientation orientation, gui::Data &data, bool broadcastingParams)

  Construct a new Players object.

- ∼Players () noexcept override=default

  Destroy the Players object.
- Players (const Players &other)=delete

  Construct a new Players object by copy deleted.
- Players (Players &&other)=default

  Construct a new Players object by move.
- Players & operator= (const Players &other)=delete

  Assign a Players object by copy deleted.
- Players & operator= (Players &&other)=default

  Assign a Players object by move.
- CharacterType getType () const override

  Get the Type of the Characters.
- const int & getRectLeft () override

  Get the Rect Left of the Player.
- void doAction () override

  Do the action of the Player.
- void display (sf::RenderWindow &window, sf::Sprite &sprite, gui::ItemDrawer &drawer, int timeUnit, SoundBox &soundbox) override

  Display the Player.
- void animate (sf::Sprite &sprite) override

  Animate the Player.
- const int & getId () override

  Get the Id of the Player.
- void setPos (sf::Vector2f pos, Orientation orientation) override

  Set the Pos of the Player.
- void setLevel (int level) override

  Set the Level of the Player.
- const int & getLevel () override

  Get the Level of the Player.
- const std::string & getTeamName () override

  Get the Team Name of the Player.
- const std::string & getColor () override

  Get the Color of the Player.
- void setInventory (const std::array< int, 7 > &res) override

  Set the Inventory of the Player.
- const std::array< int, 7 > & getInventory () override

  Get the Inventory of the Player.
- void setResources (const int &res, const int &nb) override

  Set the Resources of the Player.
- const bool & isAlive () override

  Check if the Player is Alive.
- void setAlive (bool alive) override

  Set the Alive boolean of the Player.
- const bool & isLaying () override

  Check if the Player is Laying.
- void setLaying (bool laying) override

  Set the Laying boolean of the Player.
- const int & isElevating () override

  Check if the Player is Elevating.
- void setElevating (int elevating) override

  Set the Elevating process of the Player.

- const bool & isBroadcasting () override

  Check if the Player is Broadcasting.
- void setBroadcasting (bool broadcasting) override

  Set the Broadcasting boolean of the Player.
- const sf::Vector2f & getPos () override

  Get the Pos of the Player.
- std::string randomColor (gui::Data &data)

  Get a random color.
- const Orientation & getOrientation () override

  Get the Orientation of the Player.

Private Attributes

- std::string color
- sf::IntRect rect
- sf::IntRect initial
- sf::Vector2f pos
- sf::Clock clock
- sf::Clock broadcastClock
- sf::Clock elevateClock
- sf::Clock foodClock
- sf::CircleShape circle
- int id
- int level
- bool alive
- bool laying
- bool broadcasting
- int elevating
- int soundRadius
- std::string teamName
- Orientation orientation
- std::array< int, 7 > inventory {0, 0, 0, 0, 0, 0, 0}

6.39.1 Detailed Description

Players class.

6.39.2 Constructor & Destructor Documentation

6.39.2.1 Players() [1/4]

```
Players::Players (
            sf::IntRect rect,
            sf::Vector2f pos )
```

Construct a new Players object.

Parameters

rect	
pos	

6.39.2.2 Players() [2/4]

```
Players::Players (
             sf::Vector2f pos,
             int id,
             int level,
             const std::string & teamName,
             Orientation orientation,
             gui::Data & data,
             bool broadcastingParams )
```

Construct a new Players object.

Parameters

pos	
id	
level	
teamName	
orientation	
data	
broadcastingParams	

6.39.2.3 ∼Players()

```
Players::∼Players ( )  [override], [default], [noexcept]
```

Destroy the Players object.

6.39.2.4 Players() [3/4]

```
Players::Players (
             const Players & other )  [delete]
```

Construct a new Players object by copy deleted.

Parameters

other	

6.39.2.5 Players() [4/4]

```
Players::Players (
            Players && other )  [default]
```

Construct a new Players object by move.

Parameters

other	

6.39.3 Member Function Documentation

6.39.3.1 animate()

```
void Players::animate (
            sf::Sprite & sprite )  [override], [virtual]
```

Animate the Player.

Parameters

sprite	

Implements ICharacters.

6.39.3.2 display()

```
void Players::display (
            sf::RenderWindow & window,
            sf::Sprite & sprite,
            gui::ItemDrawer & drawer,
            int timeUnit,
            SoundBox & soundbox )  [override], [virtual]
```

Display the Player.

Parameters

window	
sprite	
drawer	
timeUnit	
soundbox	

Implements ICharacters.

6.39.3.3 doAction()

```
void Players::doAction ( )  [override], [virtual]
```

Do the action of the Player.

Implements ICharacters.

6.39.3.4 getColor()

```
const std::string & Players::getColor ( )  [override], [virtual]
```

Get the Color of the Player.

Returns

const std::string&

Implements ICharacters.

6.39.3.5 getId()

```
const int & Players::getId ( )  [override], [virtual]
```

Get the Id of the Player.

Returns

const int&

Implements ICharacters.

6.39.3.6 getInventory()

```
const std::array< int, 7 > & Players::getInventory ( )  [override], [virtual]
```

Get the Inventory of the Player.

Returns

std::array<int, 7>&

Implements ICharacters.

6.39.3.7 getLevel()

```
const int & Players::getLevel ( )  [override], [virtual]
```

Get the Level of the Player.

Returns

const int&

Implements ICharacters.

6.39.3.8 getOrientation()

```
const Orientation & Players::getOrientation ( )  [override], [virtual]
```

Get the Orientation of the Player.

Returns

const Orientation&

Implements ICharacters.

6.39.3.9 getPos()

```
const sf::Vector2f & Players::getPos ( )  [override], [virtual]
```

Get the Pos of the Player.

Returns

sf::Vector2f

Implements ICharacters.

6.39.3.10 getRectLeft()

```
const int & Players::getRectLeft ( )  [override], [virtual]
```

Get the Rect Left of the Player.

Returns

const int&

Implements ICharacters.

6.39.3.11 getTeamName()

```
const std::string & Players::getTeamName ( )  [override], [virtual]
```

Get the Team Name of the Player.

Returns

const std::string&

Implements ICharacters.

6.39.3.12 getType()

```
CharacterType Players::getType ( ) const  [override], [virtual]
```

Get the Type of the Characters.

Returns

CharacterType

Implements ICharacters.

6.39.3.13 isAlive()

```
const bool & Players::isAlive ( )  [override], [virtual]
```

Check if the Player is Alive.

Returns

true

false

Implements ICharacters.

6.39.3.14 isBroadcasting()

```
const bool & Players::isBroadcasting ( )  [override], [virtual]
```

Check if the Player is Broadcasting.

Returns

true

false

Implements ICharacters.

6.39.3.15 isElevating()

```
const int & Players::isElevating ( )  [override], [virtual]
```

Check if the Player is Elevating.

Returns

int

Implements ICharacters.

6.39.3.16 isLaying()

```
const bool & Players::isLaying ( )  [override], [virtual]
```

Check if the Player is Laying.

Returns

true

false

Implements ICharacters.

6.39.3.17 operator=() [1/2]

```
Players& Players::operator= (
             const Players & other )  [delete]
```

Assign a Players object by copy deleted.

Parameters

other	

Returns

Players&

6.39.3.18 operator=() [2/2]

```
Players& Players::operator= (
            Players && other )  [default]
```

Assign a Players object by move.

Parameters

other	

Returns

Players&

6.39.3.19 randomColor()

```
std::string Players::randomColor (
            gui::Data & data )
```

Get a random color.

Parameters

data	

Returns

std::string

6.39.3.20 setAlive()

```
void Players::setAlive (
            bool alive )  [override], [virtual]
```

Set the Alive boolean of the Player.

Parameters

alive	

Implements ICharacters.

6.39.3.21 setBroadcasting()

```
void Players::setBroadcasting (
             bool broadcasting )  [override], [virtual]
```

Set the Broadcasting boolean of the Player.

Parameters

broadcasting	

Implements ICharacters.

6.39.3.22 setElevating()

```
void Players::setElevating (
             int elevating )  [override], [virtual]
```

Set the Elevating process of the Player.

Parameters

elevating	

Implements ICharacters.

6.39.3.23 setInventory()

```
void Players::setInventory (
             const std::array< int, 7 > & res )  [override], [virtual]
```

Set the Inventory of the Player.

Parameters

res	

Implements ICharacters.

6.39.3.24 setLaying()

```
void Players::setLaying (
            bool laying )  [override], [virtual]
```

Set the Laying boolean of the Player.

Parameters

laying	

Implements ICharacters.

6.39.3.25 setLevel()

```
void Players::setLevel (
            int level )  [override], [virtual]
```

Set the Level of the Player.

Parameters

level	

Implements ICharacters.

6.39.3.26 setPos()

```
void Players::setPos (
            sf::Vector2f pos,
            Orientation orientation )  [override], [virtual]
```

Set the Pos of the Player.

Parameters

pos	
orientation	

Implements ICharacters.

6.39.3.27 setResources()

```
void Players::setResources (
             const int & res,
             const int & nb )  [override], [virtual]
```

Set the Resources of the Player.

Parameters

res	
nb	

Implements ICharacters.

6.39.4 Field Documentation

6.39.4.1 alive

```
bool Players::alive  [private]
```

6.39.4.2 broadcastClock

```
sf::Clock Players::broadcastClock  [private]
```

6.39.4.3 broadcasting

```
bool Players::broadcasting  [private]
```

6.39.4.4 circle

```
sf::CircleShape Players::circle  [private]
```

6.39.4.5 clock

```
sf::Clock Players::clock  [private]
```

6.39.4.6 color

```
std::string Players::color  [private]
```

6.39.4.7 elevateClock

```
sf::Clock Players::elevateClock  [private]
```

6.39.4.8 elevating

```
int Players::elevating  [private]
```

6.39.4.9 foodClock

```
sf::Clock Players::foodClock  [private]
```

6.39.4.10 id

```
int Players::id  [private]
```

6.39.4.11 initial

```
sf::IntRect Players::initial  [private]
```

6.39.4.12 inventory

```
std::array< int, 7 > Players::inventory {0, 0, 0, 0, 0, 0, 0}  [private]
```

6.39.4.13 laying

```
bool Players::laying  [private]
```

6.39.4.14 level

```
int Players::level  [private]
```

6.39.4.15 orientation

```
Orientation Players::orientation  [private]
```

6.39.4.16 pos

```
sf::Vector2f Players::pos  [private]
```

6.39.4.17 rect

```
sf::IntRect Players::rect  [private]
```

6.39.4.18 soundRadius

```
int Players::soundRadius  [private]
```

6.39.4.19 teamName

```
std::string Players::teamName  [private]
```

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

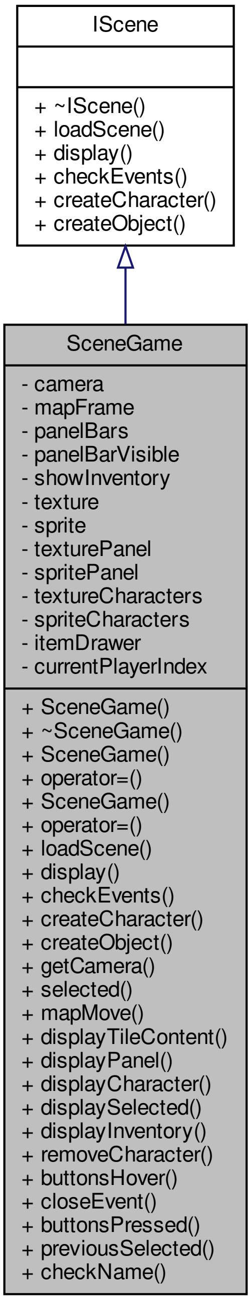
- Characters/Players/Players.hpp
- Characters/Players/Players.cpp

6.40 SceneGame Class Reference

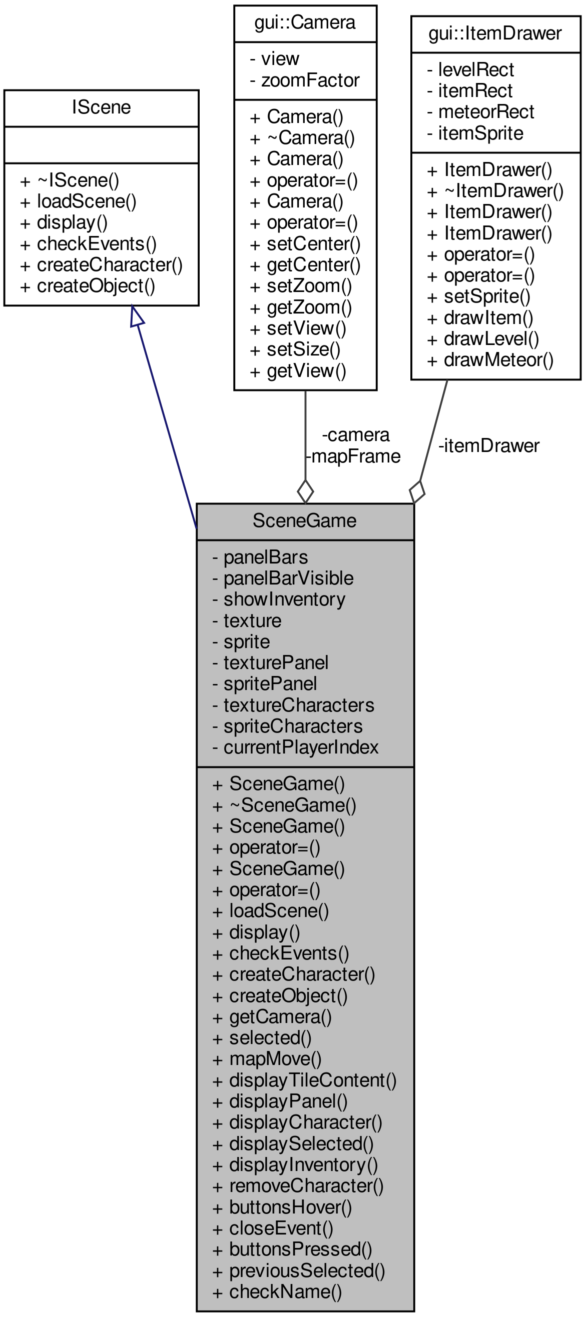
Game scene class.

```
#include <SceneGame.hpp>
```

Inheritance diagram for SceneGame:



Collaboration diagram for SceneGame:



Public Member Functions

- SceneGame ()=default

  Construct a new Scene Game object.
- ∼SceneGame () noexcept override=default

  Destroy the Scene Game object.
- SceneGame (const SceneGame &other)=delete

Construct a new Scene Game object by copy.
- SceneGame & operator= (const SceneGame &other)=delete

  Assign a Scene Game object by copy.
- SceneGame (SceneGame &&other)=default

  Construct a new Scene Game object by move.
- SceneGame & operator= (SceneGame &&other)=default

  Assign a Scene Game object by move.
- void loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Load the scene.
- void display (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Display the scene.
- void checkEvents (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Check the events of the scene.
- void createCharacter () override

  Create the character of the scene.
- void createObject (gui::Data &data) override

  Create the objects of the scene.
- const gui::Camera & getCamera ()

  Get the camera of the scene.
- void selected (std::vector< std::vector< gui::Tile >> &tiles, sf::Vector2f pos)

  Set the outline of the selected tile.
- void mapMove (gui::Window &window, gui::Data &data)

  Move the map.
- void displayTileContent (std::vector< std::vector< gui::Tile >> &map, gui::Window &window, gui::ItemDrawer &itemDrawer, gui::Data &data)

  Display tile of the Map.
- void displayPanel (gui::SceneManager &manager, gui::Window &window, gui::Data &data)

  Display the panel bar.
- void displayCharacter (gui::Window &window, gui::Data &data, gui::SceneManager &manager)

  Display the Character.
- void displaySelected (gui::Window &window, gui::Data &data, gui::ItemDrawer &itemDrawer, gui::Tile &tile)

  Display the selected tile.
- void displayInventory (gui::Data &data, gui::Tile &tile)

  Display the inventory.
- void removeCharacter (gui::SceneManager &manager, gui::Data &data)

  Remove the character.
- void buttonsHover (gui::SceneManager &manager, gui::Window &window)

  Hover state and the buttons.
- void closeEvent (gui::Window &window, gui::Data &data)

  Handle close event.
- void buttonsPressed (gui::SceneManager &manager, gui::Window &window, gui::Data &data)

  Handle buttons pressed.
- void previousSelected (std::vector< std::vector< gui::Tile >> &map)

  Handle buttons released.
- void checkName (const std::unique_ptr< gui::Button > &button, gui::Data &data)

  Check if the name is valid.

Private Attributes

- gui::Camera camera {0, 0}
- gui::Camera mapFrame {0, 0}
- std::vector< gui::PanelBar > panelBars
- bool panelBarVisible {false}
- bool showInventory {false}
- sf::Texture texture
- sf::Sprite sprite
- sf::Texture texturePanel
- sf::Sprite spritePanel
- sf::Texture textureCharacters
- sf::Sprite spriteCharacters
- gui::ItemDrawer itemDrawer
- int currentPlayerIndex

6.40.1 Detailed Description

Game scene class.

6.40.2 Constructor & Destructor Documentation

6.40.2.1 SceneGame() [1/3]

```
SceneGame::SceneGame ( )  [default]
```

Construct a new Scene Game object.

6.40.2.2 ∼SceneGame()

```
SceneGame::∼SceneGame ( )  [override], [default], [noexcept]
```

Destroy the Scene Game object.

6.40.2.3 SceneGame() [2/3]

```
SceneGame::SceneGame (
                const SceneGame & other )  [delete]
```

Construct a new Scene Game object by copy.

Parameters

other	

6.40.2.4 SceneGame() [3/3]

```
SceneGame::SceneGame (
            SceneGame && other )  [default]
```

Construct a new Scene Game object by move.

Parameters

other	

6.40.3 Member Function Documentation

6.40.3.1 buttonsHover()

```
void SceneGame::buttonsHover (
            gui::SceneManager & manager,
            gui::Window & window )
```

Hover state and the buttons.

Parameters

manager	
window	

6.40.3.2 buttonsPressed()

```
void SceneGame::buttonsPressed (
            gui::SceneManager & manager,
            gui::Window & window,
            gui::Data & data )
```

Handle buttons pressed.

Parameters

manager	
window	
data	

6.40.3.3 checkEvents()

```
void SceneGame::checkEvents (
            gui::SceneManager & manager,
            gui::Window & window,
            gui::Data & data )  [override], [virtual]
```

Check the events of the scene.

Parameters

window	The window where the events will be checked
data	The data of the game
manager	The scene manager

Implements IScene.

6.40.3.4 checkName()

```
void SceneGame::checkName (
            const std::unique_ptr< gui::Button > & button,
            gui::Data & data )
```

Check if the name is valid.

Parameters

button	
data	

6.40.3.5 closeEvent()

```
void SceneGame::closeEvent (
            gui::Window & window,
            gui::Data & data )
```

Handle close event.

Parameters

window	
data	

6.40.3.6 createCharacter()

```
void SceneGame::createCharacter ( )  [inline], [override], [virtual]
```

Create the character of the scene.

Implements IScene.

6.40.3.7 createObject()

```
void SceneGame::createObject (
             gui::Data & data )  [override], [virtual]
```

Create the objects of the scene.

Parameters

data	The data of the game

Exceptions

Errors::ErrorTexture	

Implements IScene.

6.40.3.8 display()

```
void SceneGame::display (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )  [override], [virtual]
```

Display the scene.

Parameters

window	The window where the scene will be displayed
data	The data of the game
manager	The scene manager

Implements IScene.

6.40.3.9 displayCharacter()

```
void SceneGame::displayCharacter (
            gui::Window & window,
            gui::Data & data,
            gui::SceneManager & manager )
```

Display the Character.

Parameters

window	
data	
manager	

6.40.3.10 displayInventory()

```
void SceneGame::displayInventory (
            gui::Data & data,
            gui::Tile & tile )
```

Display the inventory.

Parameters

data	
tile	

6.40.3.11 displayPanel()

```
void SceneGame::displayPanel (
            gui::SceneManager & manager,
            gui::Window & window,
            gui::Data & data )
```

Display the panel bar.

Parameters

manager	
window	
data	

6.40.3.12 displaySelected()

```
void SceneGame::displaySelected (
            gui::Window & window,
            gui::Data & data,
            gui::ItemDrawer & itemDrawer,
            gui::Tile & tile )
```

Display the selected tile.

Parameters

window	
data	
itemDrawer	
tile	

6.40.3.13 displayTileContent()

```
void SceneGame::displayTileContent (
            std::vector< std::vector< gui::Tile >> & map,
            gui::Window & window,
            gui::ItemDrawer & itemDrawer,
            gui::Data & data )
```

Display tile of the Map.

Parameters

map	
window	
itemDrawer	
data	

6.40.3.14 getCamera()

```
const gui::Camera & SceneGame::getCamera ( )
```

Get the camera of the scene.

Returns

The camera of the scene

6.40.3.15 loadScene()

```
void SceneGame::loadScene (
            gui::SceneManager & manager,
            gui::Window & window,
            gui::Data & data )  [override], [virtual]
```

Load the scene.

Parameters

window	The window where the scene will be loaded
data	The data of the game
manager	The scene manager

Implements IScene.

6.40.3.16 mapMove()

```
void SceneGame::mapMove (
            gui::Window & window,
            gui::Data & data )
```

Move the map.

Parameters

window	The window where the map will be moved
data	The data of the game

6.40.3.17 operator=() [1/2]

```
SceneGame& SceneGame::operator= (
            const SceneGame & other )  [delete]
```

Assign a Scene Game object by copy.

Parameters

other	

Returns

SceneGame&

6.40.3.18 operator=() [2/2]

```
SceneGame& SceneGame::operator= (
            SceneGame && other )  [default]
```

Assign a Scene Game object by move.

Parameters

other	

Returns

SceneGame&

6.40.3.19 previousSelected()

```
void SceneGame::previousSelected (
            std::vector< std::vector< gui::Tile >> & map )
```

Handle buttons released.

Parameters

map	

6.40.3.20 removeCharacter()

```
void SceneGame::removeCharacter (
            gui::SceneManager & manager,
            gui::Data & data )
```

Remove the character.

Parameters

manager	
data	

6.40.3.21 selected()

```
void SceneGame::selected (
            std::vector< std::vector< gui::Tile >> & tiles,
            sf::Vector2f pos )
```

Set the outline of the selected tile.

Parameters

tiles	The tiles of the map
pos	The position of the mouse

6.40.4 Field Documentation

6.40.4.1 camera

```
gui::Camera SceneGame::camera {0, 0}  [private]
```

6.40.4.2 currentPlayerIndex

```
int SceneGame::currentPlayerIndex  [private]
```

6.40.4.3 itemDrawer

```
gui::ItemDrawer SceneGame::itemDrawer  [private]
```

6.40.4.4 mapFrame

```
gui::Camera SceneGame::mapFrame {0, 0}  [private]
```

6.40.4.5 panelBars

```
std:vector< gui::PanelBar > SceneGame::panelBars  [private]
```

6.40.4.6 panelBarVisible

```
bool SceneGame::panelBarVisible {false}  [private]
```

6.40.4.7 showInventory

```
bool SceneGame::showInventory {false}  [private]
```

6.40.4.8 sprite

```
sf::Sprite SceneGame::sprite  [private]
```

6.40.4.9 spriteCharacters

```
sf::Sprite SceneGame::spriteCharacters  [private]
```

6.40.4.10 spritePanel

```
sf::Sprite SceneGame::spritePanel  [private]
```

6.40.4.11 texture

```
sf::Texture SceneGame::texture  [private]
```

6.40.4.12 textureCharacters

```
sf::Texture SceneGame::textureCharacters  [private]
```

6.40.4.13 texturePanel

```
sf::Texture SceneGame::texturePanel  [private]
```

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

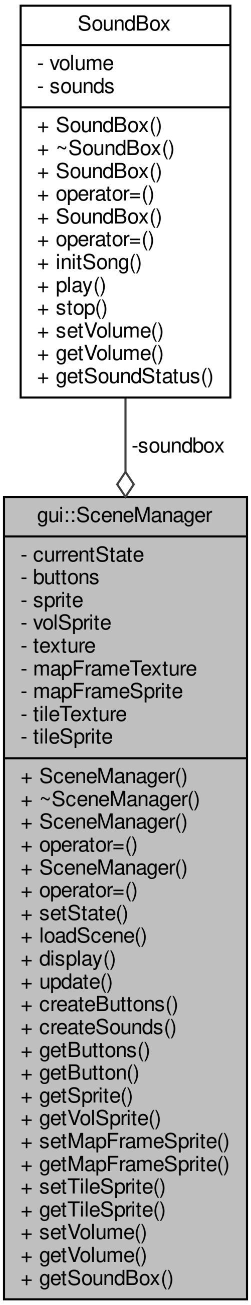
- Scene/Game/SceneGame.hpp
- Scene/Game/Display/Display.cpp
- Scene/Game/Display/DisplayTile.cpp
- Scene/Game/Event/Event.cpp
- Scene/Game/Event/MapMove.cpp
- Scene/Game/Event/Selected.cpp
- Scene/Game/SceneGame.cpp

6.41 gui::SceneManager Class Reference

SceneManager class.

```
#include <SceneManager.hpp>
```

Collaboration diagram for gui::SceneManager:



Public Member Functions

- SceneManager ()

  Construct a new Scene Manager object.
- ∼SceneManager () noexcept=default

  Destroy the Scene Manager object.
- SceneManager (const SceneManager &other)=delete

Construct a new Scene Manager object by copy.

- SceneManager & operator= (const SceneManager &other)=delete

  Assign a Scene Manager object by copy.
- SceneManager (SceneManager &&other)=default

  Construct a new Scene Manager object by move.
- SceneManager & operator= (SceneManager &&other)=default

  Assign a Scene Manager object by move.
- void setState (std::unique_ptr< IScene > state)

  Set the state of the scene.
- void loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data)
- void display (gui::Window &window, gui::Data &data)
- void update (gui::Window &window, gui::Data &data)

  Check the events of the scene.
- void createButtons ()

  Create a Buttons object.
- void createSounds ()

  Create a Sounds object.
- std::vector< std::unique_ptr< gui::Button > > & getButtons ()

  Get the Buttons object.
- gui::Button & getButton (std::string name)

  Get the Button.
- sf::Sprite & getSprite ()

  Get the Sprite.
- sf::Sprite & getVolSprite ()

  Get the Vol Sprite.
- void setMapFrameSprite (int rect)

  Set the Map Frame Sprite.
- sf::Sprite & getMapFrameSprite ()

  Get the Map Frame Sprite.
- void setTileSprite (int rect)

  Set the Tile Sprite.
- const sf::Sprite & getTileSprite ()

  Get the Tile Sprite.
- void setVolume (float volume)

  Set the Volume.
- int getVolume ()

  Get the Volume.
- SoundBox & getSoundBox ()

  Get the Sound Box.

Private Attributes

- std::unique_ptr< IScene > currentState
- std::vector< std::unique_ptr< gui::Button > > buttons
- sf::Sprite sprite
- sf::Sprite volSprite
- sf::Texture texture
- sf::Texture mapFrameTexture
- sf::Sprite mapFrameSprite
- sf::Texture tileTexture
- sf::Sprite tileSprite
- SoundBox soundbox

6.41.1 Detailed Description

SceneManager class.

6.41.2 Constructor & Destructor Documentation

6.41.2.1 SceneManager() [1/3]

```
gui::SceneManager::SceneManager ( )
```

Construct a new Scene Manager object.

Exceptions

Errors::ErrorTexture	

6.41.2.2 ∼SceneManager()

```
gui::SceneManager::∼SceneManager ( )  [default], [noexcept]
```

Destroy the Scene Manager object.

6.41.2.3 SceneManager() [2/3]

```
gui::SceneManager::SceneManager (
              const SceneManager & other )  [delete]
```

Construct a new Scene Manager object by copy.

Parameters

other	

6.41.2.4 SceneManager() [3/3]

```
gui::SceneManager::SceneManager (
              SceneManager && other )  [default]
```

Construct a new Scene Manager object by move.

Parameters

other	

6.41.3 Member Function Documentation

6.41.3.1 createButtons()

```
void gui::SceneManager::createButtons ( )
```

Create a Buttons object.

6.41.3.2 createSounds()

```
void gui::SceneManager::createSounds ( )
```

Create a Sounds object.

6.41.3.3 display()

```
void gui::SceneManager::display (
            gui::Window & window,
            gui::Data & data )
```

6.41.3.4 getButton()

```
gui::Button & gui::SceneManager::getButton (
            std::string name )
```

Get the Button.

Parameters

name	

Returns

gui::Button&

6.41.3.5 getButtons()

```
std::vector< std::unique_ptr< gui::Button > > & gui::SceneManager::getButtons ( )
```

Get the Buttons object.

Returns

std::vector<std::unique_ptr<gui::Button>>&

6.41.3.6 getMapFrameSprite()

```
sf::Sprite & gui::SceneManager::getMapFrameSprite ( )
```

Get the Map Frame Sprite.

Returns

sf::Sprite&

6.41.3.7 getSoundBox()

```
SoundBox & gui::SceneManager::getSoundBox ( )
```

Get the Sound Box.

Returns

SoundBox&

6.41.3.8 getSprite()

```
sf::Sprite & gui::SceneManager::getSprite ( )
```

Get the Sprite.

Returns

sf::Sprite&

6.41.3.9 getTileSprite()

```
const sf::Sprite & gui::SceneManager::getTileSprite ( )
```

Get the Tile Sprite.

Returns

sf::Sprite&

6.41.3.10 getVolSprite()

```
sf::Sprite & gui::SceneManager::getVolSprite ( )
```

Get the Vol Sprite.

Returns

sf::Sprite&

6.41.3.11 getVolume()

```
int gui::SceneManager::getVolume ( )
```

Get the Volume.

Returns

const int&

6.41.3.12 loadScene()

```
void gui::SceneManager::loadScene (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )
```

6.41.3.13 operator=() [1/2]

```
SceneManager& gui::SceneManager::operator= (
             const SceneManager & other )  [delete]
```

Assign a Scene Manager object by copy.

Parameters

other	

Returns

SceneManager&

6.41.3.14 operator=() [2/2]

```
SceneManager& gui::SceneManager::operator= (
             SceneManager && other )  [default]
```

Assign a Scene Manager object by move.

Parameters

other	

Returns

SceneManager&

6.41.3.15 setMapFrameSprite()

```
void gui::SceneManager::setMapFrameSprite (
             int rect )
```

Set the Map Frame Sprite.

Parameters

rect	

6.41.3.16 setState()

```
void gui::SceneManager::setState (
             std::unique_ptr< IScene > state )
```

Set the state of the scene.

Parameters

state	The state of the scene

6.41.3.17 setTileSprite()

```
void gui::SceneManager::setTileSprite (
             int rect )
```

Set the Tile Sprite.

Parameters

rect	

6.41.3.18 setVolume()

```
void gui::SceneManager::setVolume (
             float volume )
```

Set the Volume.

Parameters

volume	

6.41.3.19 update()

```
void gui::SceneManager::update (
             gui::Window & window,
             gui::Data & data )
```

Check the events of the scene.

Parameters

window	The window where the events will be checked
data	The data of the game

6.41.4 Field Documentation

6.41.4.1 buttons

```
std::vector<std::unique_ptr<gui::Button> > gui::SceneManager::buttons  [private]
```

6.41.4.2 currentState

```
std::unique_ptr< IScene > gui::SceneManager::currentState  [private]
```

6.41.4.3 mapFrameSprite

```
sf::Sprite gui::SceneManager::mapFrameSprite  [private]
```

6.41.4.4 mapFrameTexture

```
sf::Texture gui::SceneManager::mapFrameTexture  [private]
```

6.41.4.5 soundbox

```
SoundBox gui::SceneManager::soundbox  [private]
```

6.41.4.6 sprite

```
sf::Sprite gui::SceneManager::sprite  [private]
```

6.41.4.7 texture

```
sf::Texture gui::SceneManager::texture  [private]
```

6.41.4.8 tileSprite

```
sf::Sprite gui::SceneManager::tileSprite  [private]
```

6.41.4.9 tileTexture

```
sf::Texture gui::SceneManager::tileTexture  [private]
```

6.41.4.10 volSprite

```
sf::Sprite gui::SceneManager::volSprite  [private]
```

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

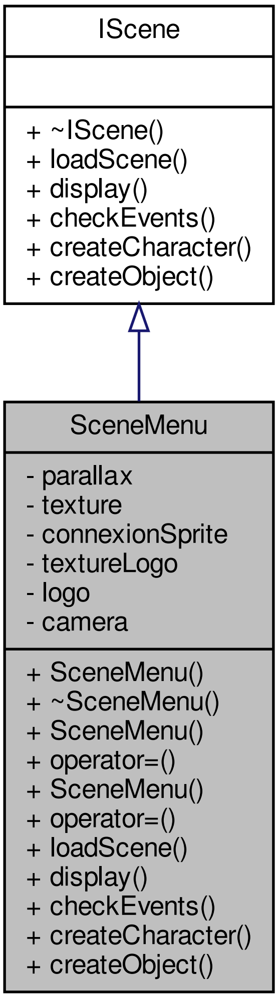
- Scene/Manager/SceneManager.hpp
- Scene/Manager/SceneManager.cpp

6.42 SceneMenu Class Reference

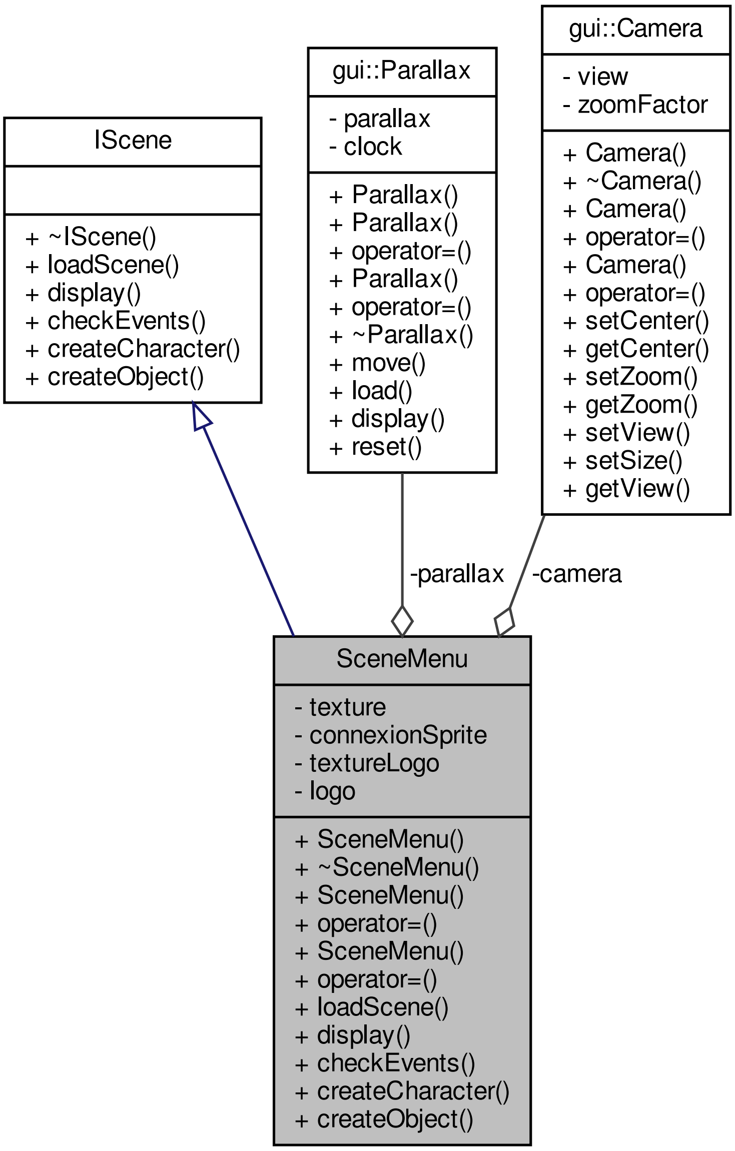
SceneMenu class.

```
#include <SceneMenu.hpp>
```

Inheritance diagram for SceneMenu:



Collaboration diagram for SceneMenu:



Public Member Functions

- SceneMenu ()=default

  Construct a new Scene Menu object.
- ∼SceneMenu () noexcept override=default

  Destroy the Scene Menu object.
- SceneMenu (const SceneMenu &other)=delete

  Construct a new Scene Menu object by copy.
- SceneMenu & operator= (const SceneMenu &other)=delete

  Assign a Scene Menu object by copy.

- SceneMenu (SceneMenu &&other)=default

  Construct a new Scene Menu object by move.
- SceneMenu & operator= (SceneMenu &&other)=default

  Assign a Scene Menu object by move.
- void loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Load the scene.
- void display (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Display the scene.
- void checkEvents (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Check the events of the scene.
- void createCharacter () override

  Create the character of the scene.
- void createObject (__attribute__((unused)) gui::Data &data) override

  Create the objects of the scene.

Private Attributes

- gui::Parallax parallax
- sf::Texture texture
- sf::Sprite connexionSprite
- sf::Texture textureLogo
- sf::Sprite logo
- gui::Camera camera {0, 0}

6.42.1 Detailed Description

SceneMenu class.

6.42.2 Constructor & Destructor Documentation

6.42.2.1 SceneMenu() [1/3]

```
SceneMenu::SceneMenu ( )  [default]
```

Construct a new Scene Menu object.

6.42.2.2 ∼SceneMenu()

```
SceneMenu::∼SceneMenu ( )  [override], [default], [noexcept]
```

Destroy the Scene Menu object.

6.42.2.3 SceneMenu() [2/3]

```
SceneMenu::SceneMenu (
            const SceneMenu & other )  [delete]
```

Construct a new Scene Menu object by copy.

Parameters

other	

6.42.2.4 SceneMenu() [3/3]

```
SceneMenu::SceneMenu (
            SceneMenu && other )  [default]
```

Construct a new Scene Menu object by move.

Parameters

other	

6.42.3 Member Function Documentation

6.42.3.1 checkEvents()

```
void SceneMenu::checkEvents (
            gui::SceneManager & manager,
            gui::Window & window,
            gui::Data & data )  [override], [virtual]
```

Check the events of the scene.

Parameters

window	The window where the events will be checked
data	The data of the game
manager	The scene manager

Implements IScene.

6.42.3.2 createCharacter()

```
void SceneMenu::createCharacter ( )  [inline], [override], [virtual]
```

Create the character of the scene.

Implements IScene.

6.42.3.3 createObject()

```
void SceneMenu::createObject (
             __attribute__((unused)) gui::Data & data )  [inline], [override]
```

Create the objects of the scene.

Parameters

data	The data of the game

6.42.3.4 display()

```
void SceneMenu::display (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )  [override], [virtual]
```

Display the scene.

Parameters

window	The window where the scene will be displayed
data	The data of the game
manager	The scene manager

Implements IScene.

6.42.3.5 loadScene()

```
void SceneMenu::loadScene (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )  [override], [virtual]
```

Load the scene.

Parameters

window	The window where the scene will be loaded
data	The data of the game
manager	The scene manager

Exceptions

Error::ErrorTexture	

Implements IScene.

6.42.3.6 operator=() [1/2]

```
SceneMenu& SceneMenu::operator= (
             const SceneMenu & other )  [delete]
```

Assign a Scene Menu object by copy.

Parameters

other	

Returns

SceneMenu&

6.42.3.7 operator=() [2/2]

```
SceneMenu& SceneMenu::operator= (
             SceneMenu && other )  [default]
```

Assign a Scene Menu object by move.

Parameters

other	

Returns

SceneMenu&

6.42.4 Field Documentation

6.42.4.1 camera

```
gui::Camera SceneMenu::camera {0, 0}  [private]
```

6.42.4.2 connexionSprite

```
sf::Sprite SceneMenu::connexionSprite  [private]
```

6.42.4.3 logo

```
sf::Sprite SceneMenu::logo  [private]
```

6.42.4.4 parallax

```
gui::Parallax SceneMenu::parallax  [private]
```

6.42.4.5 texture

```
sf::Texture SceneMenu::texture  [private]
```

6.42.4.6 textureLogo

```
sf::Texture SceneMenu::textureLogo  [private]
```

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

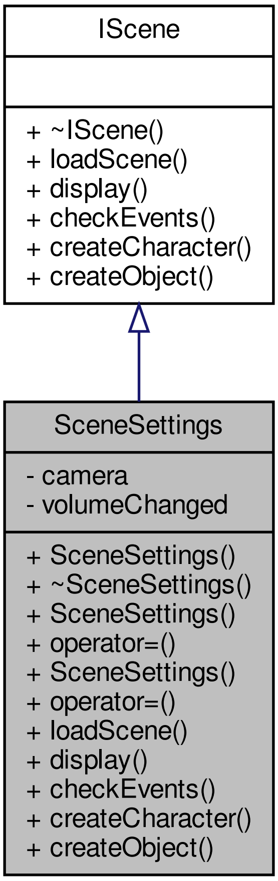
- Scene/Menu/SceneMenu.hpp
- Scene/Menu/SceneMenu.cpp

6.43 SceneSettings Class Reference

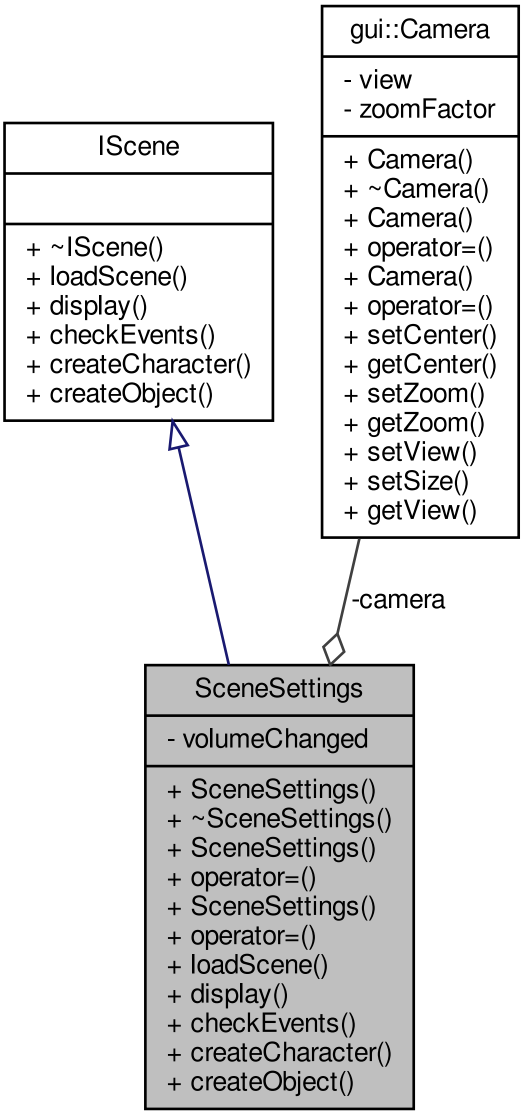
Settings scene class.

```
#include <SceneSettings.hpp>
```

Inheritance diagram for SceneSettings:



Collaboration diagram for SceneSettings:



Public Member Functions

- SceneSettings ()=default

  Construct a new Scene Settings object.
- ∼SceneSettings () noexcept override=default

  Destroy the Scene Settings object.
- SceneSettings (const SceneSettings &other)=delete

  Construct a new Scene Settings object by copy.
- SceneSettings & operator= (const SceneSettings &other)=delete

  Assign a Scene Settings object by copy.

- SceneSettings (SceneSettings &&other)=default

  Construct a new Scene Settings object by move.
- SceneSettings & operator= (SceneSettings &&other)=default

  Assign a Scene Settings object by move.
- void loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Load the scene.
- void display (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Display the scene.
- void checkEvents (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Check the events of the scene.
- void createCharacter () override

  Create the character of the scene.
- void createObject (__attribute__((unused)) gui::Data &data) override

  Create the objects of the scene.

Private Attributes

- gui::Camera camera {0, 0}
- bool volumeChanged {false}

6.43.1 Detailed Description

Settings scene class.

6.43.2 Constructor & Destructor Documentation

6.43.2.1 SceneSettings() [1/3]

```
SceneSettings::SceneSettings ( )  [default]
```

Construct a new Scene Settings object.

6.43.2.2 ∼SceneSettings()

```
SceneSettings::∼SceneSettings ( )  [override], [default], [noexcept]
```

Destroy the Scene Settings object.

6.43.2.3 SceneSettings() [2/3]

```
SceneSettings::SceneSettings (
             const SceneSettings & other )  [delete]
```

Construct a new Scene Settings object by copy.

Parameters

other	

6.43.2.4 SceneSettings() [3/3]

```
SceneSettings::SceneSettings (
             SceneSettings && other )  [default]
```

Construct a new Scene Settings object by move.

Parameters

other	

6.43.3 Member Function Documentation

6.43.3.1 checkEvents()

```
void SceneSettings::checkEvents (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )  [override], [virtual]
```

Check the events of the scene.

Parameters

window	The window where the events will be checked
data	The data of the game
manager	The scene manager

Implements IScene.

6.43.3.2 createCharacter()

```
void SceneSettings::createCharacter ( )  [inline], [override], [virtual]
```

Create the character of the scene.

Implements IScene.

6.43.3.3 createObject()

```
void SceneSettings::createObject (
            __attribute__((unused)) gui::Data & data )  [inline], [override]
```

Create the objects of the scene.

Parameters

data	The data of the game

6.43.3.4 display()

```
void SceneSettings::display (
            gui::SceneManager & manager,
            gui::Window & window,
            gui::Data & data )  [override], [virtual]
```

Display the scene.

Parameters

window	The window where the scene will be displayed
data	The data of the game
manager	The scene manager

Implements IScene.

6.43.3.5 loadScene()

```
void SceneSettings::loadScene (
            gui::SceneManager & manager,
            gui::Window & window,
            gui::Data & data )  [override], [virtual]
```

Load the scene.

Parameters

window	The window where the scene will be loaded
data	The data of the game
manager	The scene manager

Implements IScene.

6.43.3.6 operator=() [1/2]

```
SceneSettings& SceneSettings::operator= (
            const SceneSettings & other )  [delete]
```

Assign a Scene Settings object by copy.

Parameters

other	

Returns

SceneSettings&

6.43.3.7 operator=() [2/2]

```
SceneSettings& SceneSettings::operator= (
            SceneSettings && other )  [default]
```

Assign a Scene Settings object by move.

Parameters

other	

Returns

SceneSettings&

6.43.4 Field Documentation

6.43.4.1 camera

```
gui::Camera SceneSettings::camera {0, 0}  [private]
```

6.43.4.2 volumeChanged

```
bool SceneSettings::volumeChanged {false}  [private]
```

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

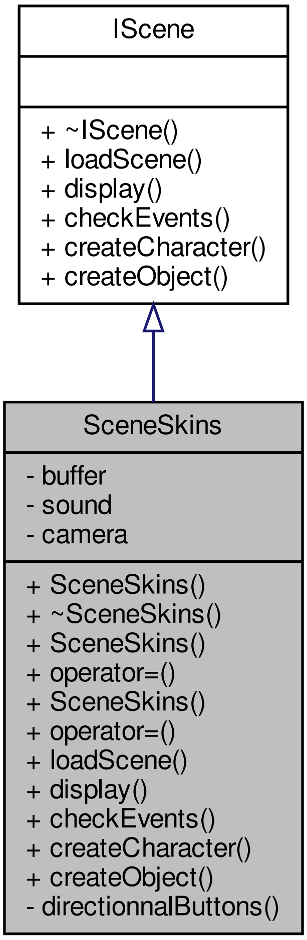
- Scene/Settings/SceneSettings.hpp
- Scene/Settings/SceneSettings.cpp

6.44 SceneSkins Class Reference

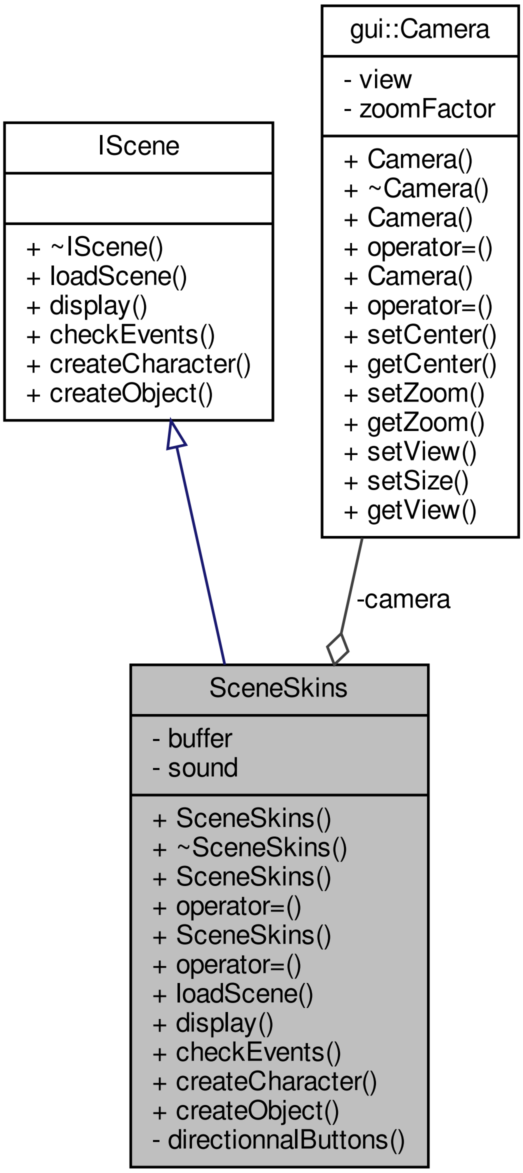
Skins scene class.

```
#include <SceneSkins.hpp>
```

Inheritance diagram for SceneSkins:



Collaboration diagram for SceneSkins:



Public Member Functions

- SceneSkins ()=default

  Construct a new Scene Skins object.
- ∼SceneSkins () noexcept override=default

  Destroy the Scene Skins object.
- SceneSkins (const SceneSkins &other)=delete

  Construct a new Scene Skins.
- SceneSkins & operator= (const SceneSkins &other)=delete

Assign a Scene Skins object by copy.

- SceneSkins (SceneSkins &&other)=default

  Construct a new Scene Skins object by move.
- SceneSkins & operator= (SceneSkins &&other)=default

  Assign a Scene Skins object by move.
- void loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Load the scene.
- void display (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Display the scene.
- void checkEvents (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Check the events of the scene.
- void createCharacter () override

  Create the character of the scene.
- void createObject (__attribute__((unused)) gui::Data &data) override

  Create the objects of the scene.

Private Member Functions

- void directionnalButtons (gui::SceneManager &manager, gui::Button &button, gui::Data &dat, gui::Window &window)

  Set Directionnal Buttons object.

Private Attributes

- sf::SoundBuffer buffer
- sf::Sound sound
- gui::Camera camera {0, 0}

6.44.1 Detailed Description

Skins scene class.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 SceneSkins() [1/3]

```
SceneSkins::SceneSkins ( )  [default]
```

Construct a new Scene Skins object.

6.44.2.2 ∼SceneSkins()

```
SceneSkins::∼SceneSkins ( )  [override], [default], [noexcept]
```

Destroy the Scene Skins object.

6.44.2.3 SceneSkins() [2/3]

```
SceneSkins::SceneSkins (
            const SceneSkins & other )  [delete]
```

Construct a new Scene Skins.

Parameters

other	

6.44.2.4 SceneSkins() [3/3]

```
SceneSkins::SceneSkins (
            SceneSkins && other )  [default]
```

Construct a new Scene Skins object by move.

Parameters

other	

6.44.3 Member Function Documentation

6.44.3.1 checkEvents()

```
void SceneSkins::checkEvents (
            gui::SceneManager & manager,
            gui::Window & window,
            gui::Data & data )  [override], [virtual]
```

Check the events of the scene.

Parameters

window	The window where the events will be checked
data	The data of the game
manager	The scene manager

Implements IScene.

6.44.3.2 createCharacter()

```
void SceneSkins::createCharacter ( )  [inline], [override], [virtual]
```

Create the character of the scene.

Implements IScene.

6.44.3.3 createObject()

```
void SceneSkins::createObject (
             __attribute__((unused)) gui::Data & data )  [inline], [override]
```

Create the objects of the scene.

Parameters

data	The data of the game

6.44.3.4 directionnalButtons()

```
void SceneSkins::directionnalButtons (
             gui::SceneManager & manager,
             gui::Button & button,
             gui::Data & dat,
             gui::Window & window )  [private]
```

Set Directionnal Buttons object.

Parameters

manager	
button	
data	

6.44.3.5 display()

```
void SceneSkins::display (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )  [override], [virtual]
```

Display the scene.

Parameters

window	The window where the scene will be displayed
data	The data of the game
manager	The scene manager

Implements IScene.

6.44.3.6 loadScene()

```
void SceneSkins::loadScene (
            gui::SceneManager & manager,
            gui::Window & window,
            gui::Data & data )  [override], [virtual]
```

Load the scene.

Parameters

window	The window where the scene will be loaded
data	The data of the game
manager	The scene manager

Implements IScene.

6.44.3.7 operator=() [1/2]

```
SceneSkins& SceneSkins::operator= (
            const SceneSkins & other )  [delete]
```

Assign a Scene Skins object by copy.

Parameters

other	

Returns

SceneSkins&

6.44.3.8 operator=() [2/2]

```
SceneSkins& SceneSkins::operator= (
             SceneSkins && other )  [default]
```

Assign a Scene Skins object by move.

Parameters

other	

Returns

SceneSkins&

6.44.4 Field Documentation

6.44.4.1 buffer

```
sf::SoundBuffer SceneSkins::buffer  [private]
```

6.44.4.2 camera

```
gui::Camera SceneSkins::camera {0, 0}  [private]
```

6.44.4.3 sound

```
sf::Sound SceneSkins::sound  [private]
```

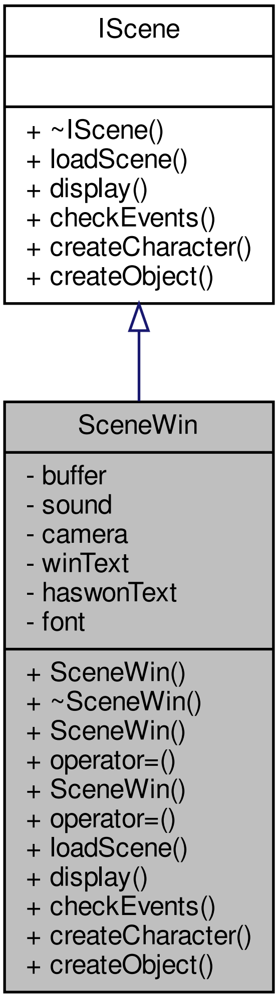
The documentation for this class was generated from the following files:

- Scene/Skins/SceneSkins.hpp
- Scene/Skins/SceneSkins.cpp

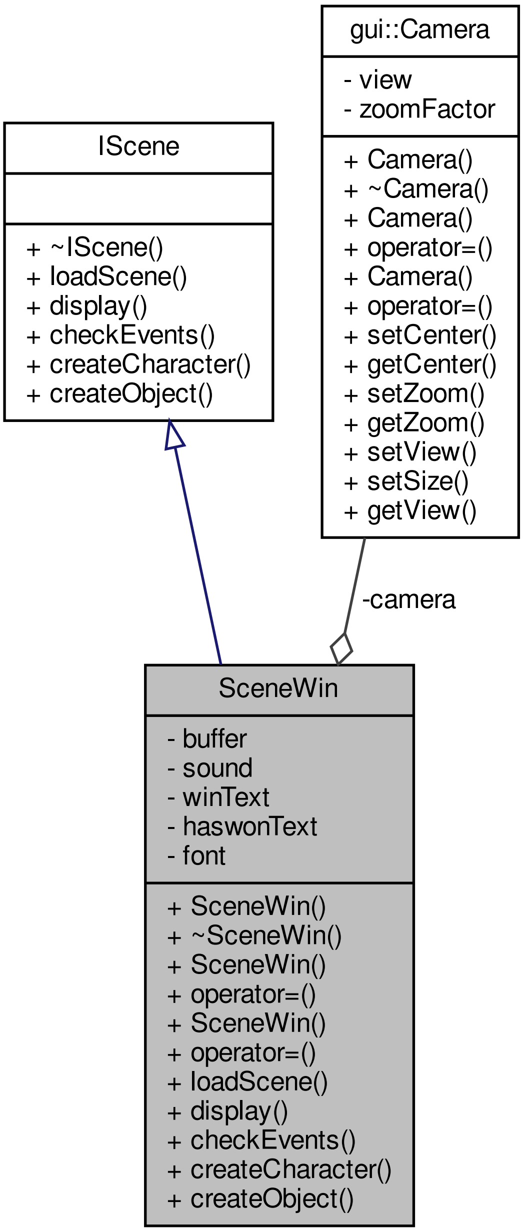
6.45 SceneWin Class Reference

```
#include <SceneWin.hpp>
```

Inheritance diagram for SceneWin:



Collaboration diagram for SceneWin:



Public Member Functions

- SceneWin ()=default

  Construct a new Scene Win object.
- ∼SceneWin () noexcept override=default

  Destroy the Scene Win object.
- SceneWin (const SceneWin &other)=delete

Construct a new Scene Win object by copy.

- SceneWin & operator= (const SceneWin &other)=delete

  Assign a Scene Win object by copy.
- SceneWin (SceneWin &&other)=default

  Construct a new Scene Win object by move.
- SceneWin & operator= (SceneWin &&other)=default

  Assign a Scene Win object by move.
- void loadScene (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Load the scene.
- void display (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Display the scene.
- void checkEvents (gui::SceneManager &manager, gui::Window &window, gui::Data &data) override

  Check the events of the scene.
- void createCharacter () override

  Create the character of the scene.
- void createObject (__attribute__((unused)) gui::Data &data) override

  Create the objects of the scene.

Private Attributes

- sf::SoundBuffer buffer
- sf::Sound sound
- gui::Camera camera {0, 0}
- sf::Text winText
- sf::Text haswonText
- sf::Font font

6.45.1 Constructor & Destructor Documentation

6.45.1.1 SceneWin() [1/3]

```
SceneWin::SceneWin ( )  [default]
```

Construct a new Scene Win object.

6.45.1.2 ∼SceneWin()

```
SceneWin::∼SceneWin ( )  [override], [default], [noexcept]
```

Destroy the Scene Win object.

6.45.1.3 SceneWin() [2/3]

```
SceneWin::SceneWin (
            const SceneWin & other )  [delete]
```

Construct a new Scene Win object by copy.

Parameters

other	

6.45.1.4 SceneWin() [3/3]

```
SceneWin::SceneWin (
            SceneWin && other )  [default]
```

Construct a new Scene Win object by move.

Parameters

other	

6.45.2 Member Function Documentation

6.45.2.1 checkEvents()

```
void SceneWin::checkEvents (
            gui::SceneManager & manager,
            gui::Window & window,
            gui::Data & data )  [override], [virtual]
```

Check the events of the scene.

Parameters

window	The window where the events will be checked
data	The data of the game
manager	The scene manager

Implements IScene.

6.45.2.2 createCharacter()

```
void SceneWin::createCharacter ( )  [inline], [override], [virtual]
```

Create the character of the scene.

Implements IScene.

6.45.2.3 createObject()

```
void SceneWin::createObject (
             __attribute__((unused)) gui::Data & data )  [inline], [override]
```

Create the objects of the scene.

6.45.2.4 display()

```
void SceneWin::display (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )  [override], [virtual]
```

Display the scene.

Parameters

window	The window where the scene will be displayed
data	The data of the game
manager	The scene manager

Implements IScene.

6.45.2.5 loadScene()

```
void SceneWin::loadScene (
             gui::SceneManager & manager,
             gui::Window & window,
             gui::Data & data )  [override], [virtual]
```

Load the scene.

Parameters

window	The window where the scene will be loaded
data	The data of the game
manager	The scene manager

Exceptions

Errors::ErrorFont	

Implements IScene.

6.45.2.6 operator=() [1/2]

```
SceneWin& SceneWin::operator= (
             const SceneWin & other )  [delete]
```

Assign a Scene Win object by copy.

Parameters

other	

Returns

SceneWin&

6.45.2.7 operator=() [2/2]

```
SceneWin& SceneWin::operator= (
             SceneWin && other )  [default]
```

Assign a Scene Win object by move.

Parameters

other	

Returns

SceneWin&

6.45.3 Field Documentation

6.45.3.1 buffer

```
sf::SoundBuffer SceneWin::buffer  [private]
```

6.45.3.2 camera

```
gui::Camera SceneWin::camera {0, 0}  [private]
```

6.45.3.3 font

```
sf::Font SceneWin::font [private]
```

6.45.3.4 haswonText

```
sf::Text SceneWin::haswonText [private]
```

6.45.3.5 sound

```
sf::Sound SceneWin::sound [private]
```

6.45.3.6 winText

```
sf::Text SceneWin::winText [private]
```

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

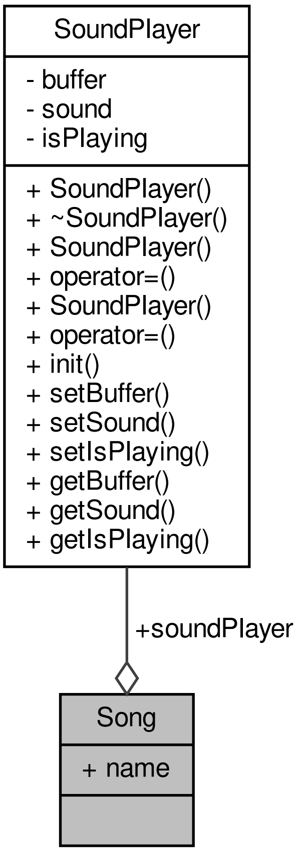
- Scene/Win/SceneWin.hpp
- Scene/Win/SceneWin.cpp

6.46 Song Struct Reference

Song struct.

```
#include <SoundBox.hpp>
```

Collaboration diagram for Song:



Data Fields

- std::string name
- SoundPlayer soundPlayer

6.46.1 Detailed Description

Song struct.

6.46.2 Field Documentation

6.46.2.1 name

```
std::string Song::name
```

6.46.2.2 soundPlayer

```
SoundPlayer Song::soundPlayer
```

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

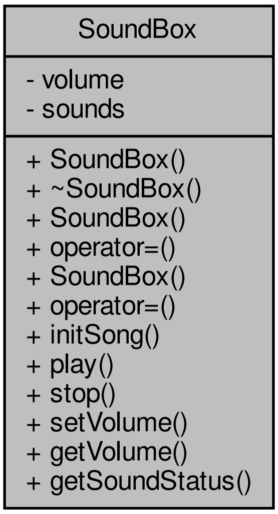
- Components/SoundBox/SoundBox.hpp

6.47 SoundBox Class Reference

SoundBox class.

```
#include <SoundBox.hpp>
```

Collaboration diagram for SoundBox:



Public Member Functions

- SoundBox ()=default

  Construct a new SoundBox object.
- ∼SoundBox () noexcept=default

  Destroy the SoundBox object.
- SoundBox (const SoundBox &other)=delete

  Construct a new SoundBox object by copy.
- SoundBox & operator= (const SoundBox &other)=delete

  Assign a SoundBox object by copy.
- SoundBox (SoundBox &&other)=default

  Construct a new SoundBox object by move.
- SoundBox & operator= (SoundBox &&other)=default

  Assign a SoundBox object by move.
- void initSong (const std::string &songName, const std::string &filePath, bool looping)

  Init a song.
- void play (const std::string &songName)

  Play a song.
- void stop (const std::string &songName)

  Stop a song.
- void setVolume (float volume)

  Set the Volume object.
- int getVolume ()

  Get the Volume object.
- sf::SoundSource::Status getSoundStatus (const std::string &songName)

Private Attributes

- float volume {30}
- std::vector< Song > sounds

6.47.1 Detailed Description

SoundBox class.

6.47.2 Constructor & Destructor Documentation

6.47.2.1 SoundBox() `[1/3]`

```
SoundBox::SoundBox ( )  [default]
```

Construct a new SoundBox object.

6.47.2.2 ∼SoundBox()

```
SoundBox::∼SoundBox ( )  [default], [noexcept]
```

Destroy the SoundBox object.

6.47.2.3 SoundBox() [2/3]

```
SoundBox::SoundBox (
              const SoundBox & other )  [delete]
```

Construct a new SoundBox object by copy.

Parameters

other	

6.47.2.4 SoundBox() [3/3]

```
SoundBox::SoundBox (
              SoundBox && other )  [default]
```

Construct a new SoundBox object by move.

Parameters

other	

6.47.3 Member Function Documentation

6.47.3.1 getSoundStatus()

```
sf::SoundSource::Status SoundBox::getSoundStatus (
              const std::string & songName )
```

6.47.3.2 getVolume()

```
int SoundBox::getVolume ( )
```

Get the Volume object.

Returns

int

6.47.3.3 initSong()

```
void SoundBox::initSong (
            const std::string & songName,
            const std::string & filePath,
            bool looping )
```

Init a song.

Parameters

songName	
filePath	

6.47.3.4 operator=() [1/2]

```
SoundBox& SoundBox::operator= (
            const SoundBox & other )  [delete]
```

Assign a SoundBox object by copy.

Parameters

other	

Returns

SoundBox&

6.47.3.5 operator=() [2/2]

```
SoundBox& SoundBox::operator= (
            SoundBox && other )  [default]
```

Assign a SoundBox object by move.

Assign a SoundBox object by move.

Parameters

other	

Returns

SoundBox&

6.47.3.6 play()

```
void SoundBox::play (
             const std::string & songName )
```

Play a song.

Parameters

songName	

6.47.3.7 setVolume()

```
void SoundBox::setVolume (
             float volume )
```

Set the Volume object.

Parameters

songName	
volume	

6.47.3.8 stop()

```
void SoundBox::stop (
             const std::string & songName )
```

Stop a song.

Parameters

songName	

6.47.4 Field Documentation

6.47.4.1 sounds

```
std::vector<Song> SoundBox::sounds  [private]
```

6.47.4.2 volume

```
float SoundBox::volume {30}  [private]
```

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

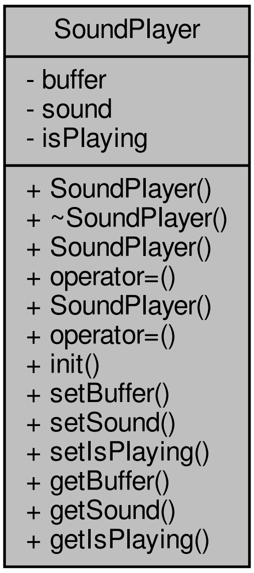
- Components/SoundBox/SoundBox.hpp
- Components/SoundBox/SoundBox.cpp

6.48 SoundPlayer Class Reference

SoundPlayer class.

```
#include <SoundPlayers.hpp>
```

Collaboration diagram for SoundPlayer:



Public Member Functions

- SoundPlayer ()=default

  Construct a new Sound Player object.
- ∼SoundPlayer () noexcept=default

  Destroy the Sound Player object.
- SoundPlayer (const SoundPlayer &other)=delete

  Construct a new Sound Player object by copy.
- SoundPlayer & operator= (const SoundPlayer &other)=delete

  Assign a Sound Player object by copy.
- SoundPlayer (SoundPlayer &&other)=default

  Construct a new Sound Player object by move.
- SoundPlayer & operator= (SoundPlayer &&other)=default

  Assign a Sound Player object by move.
- void init (const std::string &filePath)

  Play the sound.
- void setBuffer (const sf::SoundBuffer &buffer)
- void setSound (const sf::Sound &sound)
- void setIsPlaying (bool isPlaying)
- const sf::SoundBuffer & getBuffer () const
- sf::Sound & getSound ()
- const bool & getIsPlaying () const

Private Attributes

- sf::SoundBuffer buffer
- sf::Sound sound
- bool isPlaying

6.48.1 Detailed Description

SoundPlayer class.

6.48.2 Constructor & Destructor Documentation

6.48.2.1 SoundPlayer() [1/3]

```
SoundPlayer::SoundPlayer ( )  [default]
```

Construct a new Sound Player object.

6.48.2.2 ∼SoundPlayer()

```
SoundPlayer::∼SoundPlayer ( )  [default], [noexcept]
```

Destroy the Sound Player object.

6.48.2.3 SoundPlayer() [2/3]

```
SoundPlayer::SoundPlayer (
            const SoundPlayer & other )  [delete]
```

Construct a new Sound Player object by copy.

Parameters

other	

6.48.2.4 SoundPlayer() [3/3]

```
SoundPlayer::SoundPlayer (
            SoundPlayer && other )  [default]
```

Construct a new Sound Player object by move.

Parameters

other	

6.48.3 Member Function Documentation

6.48.3.1 getBuffer()

```
const sf::SoundBuffer & SoundPlayer::getBuffer ( ) const
```

6.48.3.2 getIsPlaying()

```
const bool & SoundPlayer::getIsPlaying ( ) const
```

6.48.3.3 getSound()

```
sf::Sound & SoundPlayer::getSound ( )
```

6.48.3.4 init()

```
void SoundPlayer::init (
             const std::string & filePath )
```

Play the sound.

Parameters

filePath	

6.48.3.5 operator=() [1/2]

```
SoundPlayer& SoundPlayer::operator= (
             const SoundPlayer & other )  [delete]
```

Assign a Sound Player object by copy.

Parameters

other	

6.48.3.6 operator=() [2/2]

```
SoundPlayer& SoundPlayer::operator= (
             SoundPlayer && other )  [default]
```

Assign a Sound Player object by move.

Parameters

other	

6.48.3.7 setBuffer()

```
void SoundPlayer::setBuffer (
            const sf::SoundBuffer & buffer )
```

6.48.3.8 setIsPlaying()

```
void SoundPlayer::setIsPlaying (
            bool isPlaying )
```

6.48.3.9 setSound()

```
void SoundPlayer::setSound (
            const sf::Sound & sound )
```

6.48.4 Field Documentation

6.48.4.1 buffer

```
sf::SoundBuffer SoundPlayer::buffer  [private]
```

6.48.4.2 isPlaying

```
bool SoundPlayer::isPlaying  [private]
```

6.48.4.3 sound

```
sf::Sound SoundPlayer::sound  [private]
```

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

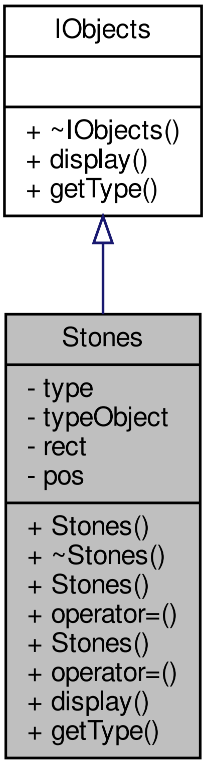
- Components/SoundBox/SoundPlayers.hpp
- Components/SoundBox/SoundPlayers.cpp

6.49 Stones Class Reference

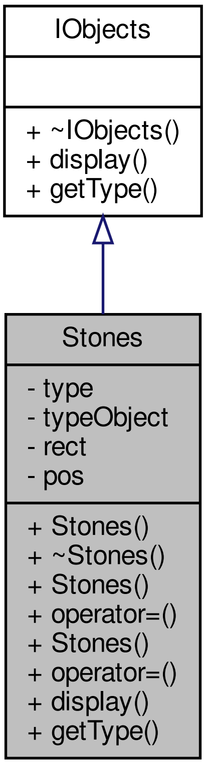
Stones class.

```
#include <Stones.hpp>
```

Inheritance diagram for Stones:



Collaboration diagram for Stones:



Public Member Functions

- Stones (TypeStone type, sf::Vector2f pos, TypeObject typeObject)

  Construct a new Stones object.
- ∼Stones () noexcept override=default

  Destroy the Stones object.
- Stones (const Stones &stone)=default

  Construct a new Stones object by copy.
- Stones & operator= (const Stones &stone)=default

  Assign a Stones object by copy.
- Stones (Stones &&stone)=default

  Construct a new Stones object by move.
- Stones & operator= (Stones &&stone)=default

  Assign a Stones object by move.
- void display (sf::RenderWindow &window, sf::Sprite &sprite, float size) override

  Display the object.
- const TypeObject & getType () override

  Get The type of the object.

Private Attributes

- TypeStone type

    Get The Sprite of the object.
- TypeObject typeObject
- sf::IntRect rect
- sf::Vector2f pos

6.49.1 Detailed Description

Stones class.

6.49.2 Constructor & Destructor Documentation

6.49.2.1 Stones() [1/3]

```
Stones::Stones (
            TypeStone type,
            sf::Vector2f pos,
            TypeObject typeObject )
```

Construct a new Stones object.

Parameters

type	
pos	
typeObject	

6.49.2.2 ~Stones()

```
Stones::~Stones ( )  [override], [default], [noexcept]
```

Destroy the Stones object.

6.49.2.3 Stones() [2/3]

```
Stones::Stones (
            const Stones & stone )  [default]
```

Construct a new Stones object by copy.

Parameters

stone	

6.49.2.4 Stones() [3/3]

```
Stones::Stones (
             Stones && stone )  [default]
```

Construct a new Stones object by move.

Parameters

stone	

6.49.3 Member Function Documentation

6.49.3.1 display()

```
void Stones::display (
             sf::RenderWindow & window,
             sf::Sprite & sprite,
             float size )  [override], [virtual]
```

Display the object.

Parameters

window	The window where the object will be displayed

Implements IObjects.

6.49.3.2 getType()

```
const TypeObject & Stones::getType ( )  [override], [virtual]
```

Get The type of the object.

Returns

The type of the object

Implements IObjects.

6.49.3.3 operator=() [1/2]

```
Stones& Stones::operator= (
             const Stones & stone )  [default]
```

Assign a Stones object by copy.

Parameters

stone	

Returns

Stones&

6.49.3.4 operator=() [2/2]

```
Stones& Stones::operator= (
             Stones && stone )  [default]
```

Assign a Stones object by move.

Parameters

stone	

Returns

Stones&

6.49.4 Field Documentation

6.49.4.1 pos

```
sf::Vector2f Stones::pos  [private]
```

6.49.4.2 rect

```
sf::IntRect Stones::rect  [private]
```

6.49.4.3 type

```
TypeStone Stones::type  [private]
```

Get The Sprite of the object.

Returns

The Sprite of the object

6.49.4.4 typeObject

```
TypeObject Stones::typeObject  [private]
```

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

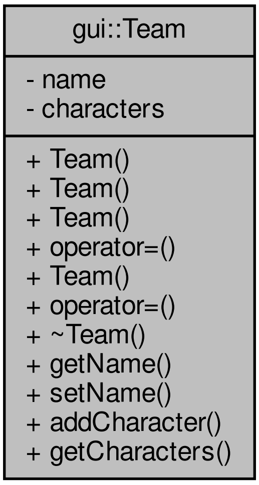
- Objects/Stones/Stones.hpp
- Objects/Stones/Stones.cpp

6.50 gui::Team Class Reference

Team class.

```
#include <Team.hpp>
```

Collaboration diagram for gui::Team:



Public Member Functions

- Team ()=default

    Construct a new Team object.
- Team (const std::string &name)

    Construct a new Team object.
- Team (const Team &other)=delete

    Construct a new Team object by copy.
- Team & operator= (const Team &other)=delete

    Assign a Team object by copy.
- Team (Team &&other)=default

    Construct a new Team object by move.
- Team & operator= (Team &&other)=default

    Assign a Team object by move.
- ∼Team () noexcept=default

    Destroy the Team object.
- const std::string & getName () const

    Get the Name.
- void setName (const std::string &name)

    Set the Name object.
- void addCharacter (const int &id)

    Add a character to the team.
- const std::vector< int > & getCharacters ()

    Get the Characters.

Private Attributes

- std::string name {""}
- std::vector< int > characters

6.50.1 Detailed Description

Team class.

6.50.2 Constructor & Destructor Documentation

6.50.2.1 Team() [1/4]

```
gui::Team::Team ( )  [default]
```

Construct a new Team object.

6.50.2.2 Team() [2/4]

```
gui::Team::Team (
             const std::string & name )  [inline], [explicit]
```

Construct a new Team object.

Parameters

name	

6.50.2.3 Team() [3/4]

```
gui::Team::Team (
             const Team & other )  [delete]
```

Construct a new Team object by copy.

Parameters

other	

6.50.2.4 Team() [4/4]

```
gui::Team::Team (
             Team && other )  [default]
```

Construct a new Team object by move.

Parameters

other	

6.50.2.5 ∼Team()

```
gui::Team::∼Team ( )  [default], [noexcept]
```

Destroy the Team object.

6.50.3 Member Function Documentation

6.50.3.1 addCharacter()

```
void gui::Team::addCharacter (
             const int & id )
```

Add a character to the team.

Parameters

id	

6.50.3.2 getCharacters()

```
const std::vector< int > & gui::Team::getCharacters ( )
```

Get the Characters.

Returns

const std::vector<int>&

6.50.3.3 getName()

```
const std::string & gui::Team::getName ( ) const
```

Get the Name.

Returns

const std::string&

6.50.3.4 operator=() [1/2]

```
Team& gui::Team::operator= (
              const Team & other )  [delete]
```

Assign a Team object by copy.

Parameters

other	

Returns

Team&

6.50.3.5 operator=() [2/2]

```
Team& gui::Team::operator= (
             Team && other )  [default]
```

Assign a Team object by move.

Parameters

other	

Returns

Team&

6.50.3.6 setName()

```
void gui::Team::setName (
             const std::string & name )
```

Set the Name object.

Parameters

name	

6.50.4 Field Documentation

6.50.4.1 characters

```
std::vector< int > gui::Team::characters  [private]
```

6.50.4.2 name

```
std::string gui::Team::name {""}  [private]
```

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

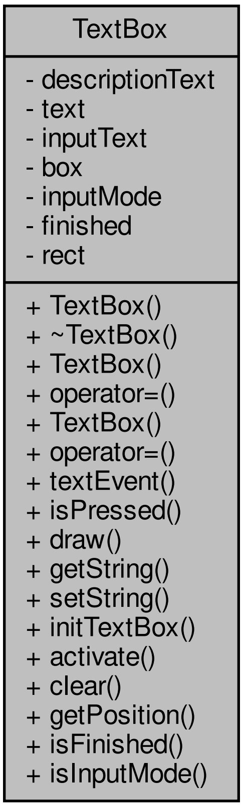
- Team/Team.hpp
- Team/Team.cpp

6.51 TextBox Class Reference

TextBox class.

```
#include <TextBox.hpp>
```

Collaboration diagram for TextBox:



Public Member Functions

- TextBox ()=default

  Construct a new Text Box object.
- ∼TextBox () noexcept=default

  Destroy the Text Box object.
- TextBox (const TextBox &other)=delete

  Construct a new Text Box object by copy.
- TextBox & operator= (const TextBox &other)=delete

  Assign a Text Box object by copy.
- TextBox (TextBox &&other)=default

  Construct a new Text Box object by move.
- TextBox & operator= (TextBox &&other)=default

Assign a Text Box object by move.

- int textEvent (sf::Event event)

  Handle the text event.
- bool isPressed (sf::Vector2f mousePos)

  Check if the mouse is pressed.
- void draw (sf::RenderWindow &window, sf::Sprite &sprite)

  Draw the text box.
- const sf::String & getString ()

  Get the String.
- void setString (const sf::String &str)

  Set the String.
- void initTextBox (const sf::Font &font, unsigned int characterSize, sf::Vector2f size, sf::Vector2f position)

  Init the text box.
- void activate (bool mode)

  Activate the text box.
- void clear ()

  Clear the text box.
- sf::Vector2f getPosition ()

  Get the Position.
- const bool & isFinished ()

  Check if the text box is finished.
- const bool & isInputMode ()

  Check if the text box is in input mode.

Private Attributes

- sf::Text descriptionText
- sf::Text text
- sf::String inputText
- sf::RectangleShape box
- bool inputMode
- bool finished
- sf::IntRect rect

6.51.1 Detailed Description

TextBox class.

6.51.2 Constructor & Destructor Documentation

6.51.2.1 TextBox() [1/3]

```
TextBox::TextBox ( )  [default]
```

Construct a new Text Box object.

6.51.2.2 ∼TextBox()

```
TextBox::∼TextBox ( )  [default], [noexcept]
```

Destroy the Text Box object.

6.51.2.3 TextBox() [2/3]

```
TextBox::TextBox (
              const TextBox & other )  [delete]
```

Construct a new Text Box object by copy.

Parameters

other	

6.51.2.4 TextBox() [3/3]

```
TextBox::TextBox (
              TextBox && other )  [default]
```

Construct a new Text Box object by move.

Parameters

other	

6.51.3 Member Function Documentation

6.51.3.1 activate()

```
void TextBox::activate (
              bool mode )
```

Activate the text box.

Parameters

mode	

6.51.3.2 clear()

```
void TextBox::clear ( )
```

Clear the text box.

6.51.3.3 draw()

```
void TextBox::draw (
            sf::RenderWindow & window,
            sf::Sprite & sprite )
```

Draw the text box.

Parameters

window	
sprite	

6.51.3.4 getPosition()

```
sf::Vector2f TextBox::getPosition ( )
```

Get the Position.

Returns

sf::Vector2f

6.51.3.5 getString()

```
const sf::String & TextBox::getString ( )
```

Get the String.

Returns

const sf::String&

6.51.3.6 initTextBox()

```
void TextBox::initTextBox (
             const sf::Font & font,
             unsigned int characterSize,
             sf::Vector2f size,
             sf::Vector2f position )
```

Init the text box.

Parameters

font	
characterSize	
size	
position	

6.51.3.7 isFinished()

```
const bool & TextBox::isFinished ( )
```

Check if the text box is finished.

Returns

true

false

6.51.3.8 isInputMode()

```
const bool & TextBox::isInputMode ( )
```

Check if the text box is in input mode.

Returns

true

false

6.51.3.9 isPressed()

```
bool TextBox::isPressed (
             sf::Vector2f mousePos )
```

Check if the mouse is pressed.

Parameters

mousePos	

Returns

true

false

6.51.3.10 operator=() [1/2]

```
TextBox& TextBox::operator= (
            const TextBox & other )  [delete]
```

Assign a Text Box object by copy.

Parameters

other	

Returns

TextBox&

6.51.3.11 operator=() [2/2]

```
TextBox& TextBox::operator= (
            TextBox && other )  [default]
```

Assign a Text Box object by move.

Parameters

other	

Returns

TextBox&

6.51.3.12 setString()

```
void TextBox::setString (
            const sf::String & str )
```

Set the String.

Parameters

str	

6.51.3.13 textEvent()

```
int TextBox::textEvent (
            sf::Event event )
```

Handle the text event.

Parameters

event	

Returns

int

6.51.4 Field Documentation

6.51.4.1 box

```
sf::RectangleShape TextBox::box  [private]
```

6.51.4.2 descriptionText

```
sf::Text TextBox::descriptionText  [private]
```

6.51.4.3 finished

```
bool TextBox::finished  [private]
```

6.51.4.4 inputMode

```
bool TextBox::inputMode  [private]
```

6.51.4.5 inputText

```
sf::String TextBox::inputText  [private]
```

6.51.4.6 rect

```
sf::IntRect TextBox::rect  [private]
```

6.51.4.7 text

```
sf::Text TextBox::text  [private]
```

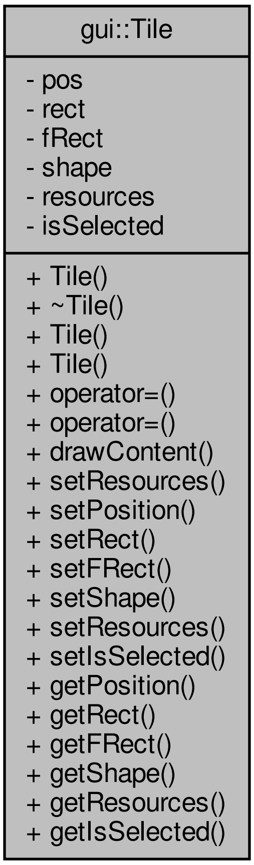
The documentation for this class was generated from the following files:

- Components/TextBox/TextBox.hpp
- Components/TextBox/TextBox.cpp

6.52 gui::Tile Class Reference

```
#include <Tile.hpp>
```

Collaboration diagram for gui::Tile:



Public Member Functions

- Tile ()=default

  Construct a new Tile object.
- ∼Tile ()=default

  Destroy the Tile object.
- Tile (const Tile &other)=delete

  Construct a new Tile object by copy.
- Tile (Tile &&other)=default

  Construct a new Tile object by move.
- Tile & operator= (const Tile &other)=delete

  Assign a Tile object by copy.
- Tile & operator= (Tile &&other)=default

  Assign a Tile object by move.
- void drawContent (gui::ItemDrawer &itemDrawer, sf::RenderWindow &window)

  Set the Resources.
- void setResources (const int &res, const int &nb)

Set the Resources.
- void setPosition (const sf::Vector2f &pos)

  Set the Position.
- void setRect (const sf::IntRect &rect)

  Set the Rect.
- void setFRect (const sf::FloatRect &fRect)

  Set the Float Rect.
- void setShape (const sf::RectangleShape &shape)

  Set the Shape.
- void setResources (const std::vector< infos > &resources)

  Set the Resources.
- void setIsSelected (const bool &isSelected)

  Set boolean if tiles is selected.
- const sf::Vector2f & getPosition ()

  Get the Position.
- const sf::IntRect & getRect ()

  Get the Rect.
- const sf::FloatRect & getFRect ()

  Get the Float Rect.
- sf::RectangleShape & getShape ()

  Get the Shape.
- std::vector< infos > & getResources ()

  Get the Resources.
- const bool & getIsSelected ()

  Get the boolean selected.

Private Attributes

- sf::Vector2f pos
- sf::IntRect rect
- sf::FloatRect fRect
- sf::RectangleShape shape
- std::vector< infos > resources
- bool isSelected

6.52.1 Constructor & Destructor Documentation

6.52.1.1 Tile() [1/3]

```
gui::Tile::Tile ( )  [default]
```

Construct a new Tile object.

6.52.1.2 ∼Tile()

```
gui::Tile::∼Tile ( )  [default]
```

Destroy the Tile object.

6.52.1.3 Tile() [2/3]

```
gui::Tile::Tile (
            const Tile & other )  [delete]
```

Construct a new Tile object by copy.

Parameters

other	

6.52.1.4 Tile() [3/3]

```
gui::Tile::Tile (
            Tile && other )  [default]
```

Construct a new Tile object by move.

Parameters

other	

6.52.2 Member Function Documentation

6.52.2.1 drawContent()

```
void Tile::drawContent (
            gui::ItemDrawer & itemDrawer,
            sf::RenderWindow & window )
```

Set the Resources.

Parameters

itemDrawer	
window	

6.52.2.2 getFRect()

```
const sf::FloatRect & Tile::getFRect ( )
```

Get the Float Rect.

Returns

const sf::FloatRect&

6.52.2.3 getIsSelected()

```
const bool & Tile::getIsSelected ( )
```

Get the boolean selected.

Returns

true

false

6.52.2.4 getPosition()

```
const sf::Vector2f & Tile::getPosition ( )
```

Get the Position.

Returns

const sf::Vector2f&

6.52.2.5 getRect()

```
const sf::IntRect & Tile::getRect ( )
```

Get the Rect.

Returns

const sf::IntRect&

6.52.2.6 getResources()

```
std::vector< infos > & Tile::getResources ( )
```

Get the Resources.

Returns

std::vector<infos>&

6.52.2.7 getShape()

```
sf::RectangleShape & Tile::getShape ( )
```

Get the Shape.

Returns

sf::RectangleShape&

6.52.2.8 operator=() [1/2]

```
Tile& gui::Tile::operator= (
            const Tile & other )  [delete]
```

Assign a Tile object by copy.

Parameters

itemDrawer	
window	

6.52.2.9 operator=() [2/2]

```
Tile& gui::Tile::operator= (
            Tile && other )  [default]
```

Assign a Tile object by move.

Parameters

itemDrawer	
window	

6.52.2.10 setFRect()

```
void Tile::setFRect (
             const sf::FloatRect & fRect )
```

Set the Float Rect.

Parameters

fRect	

6.52.2.11 setIsSelected()

```
void Tile::setIsSelected (
             const bool & isSelected )
```

Set boolean if tiles is selected.

Parameters

isSelected	

6.52.2.12 setPosition()

```
void Tile::setPosition (
             const sf::Vector2f & pos )
```

Set the Position.

Parameters

pos	

6.52.2.13 setRect()

```
void Tile::setRect (
             const sf::IntRect & rect )
```

Set the Rect.

Parameters

rect	

6.52.2.14 setResources() [1/2]

```
void Tile::setResources (
            const int & res,
            const int & nb )
```

Set the Resources.

Parameters

res	
nb	

6.52.2.15 setResources() [2/2]

```
void Tile::setResources (
            const std::vector< infos > & resources )
```

Set the Resources.

Parameters

resources	

6.52.2.16 setShape()

```
void Tile::setShape (
            const sf::RectangleShape & shape )
```

Set the Shape.

Parameters

shape	

6.52.3 Field Documentation

6.52.3.1 fRect

```
sf::FloatRect Tile::fRect  [private]
```

6.52.3.2 isSelected

```
bool Tile::isSelected  [private]
```

6.52.3.3 pos

```
sf::Vector2f Tile::pos  [private]
```

6.52.3.4 rect

```
sf::IntRect Tile::rect  [private]
```

6.52.3.5 resources

```
std::vector< infos > Tile::resources  [private]
```

6.52.3.6 shape

```
sf::RectangleShape Tile::shape  [private]
```

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

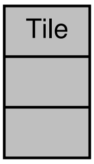
- Map/Tile.hpp
- Map/Tile.cpp

6.53 Tile Class Reference

Tile class.

```
#include <Tile.hpp>
```

Collaboration diagram for Tile:



6.53.1 Detailed Description

Tile class.

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

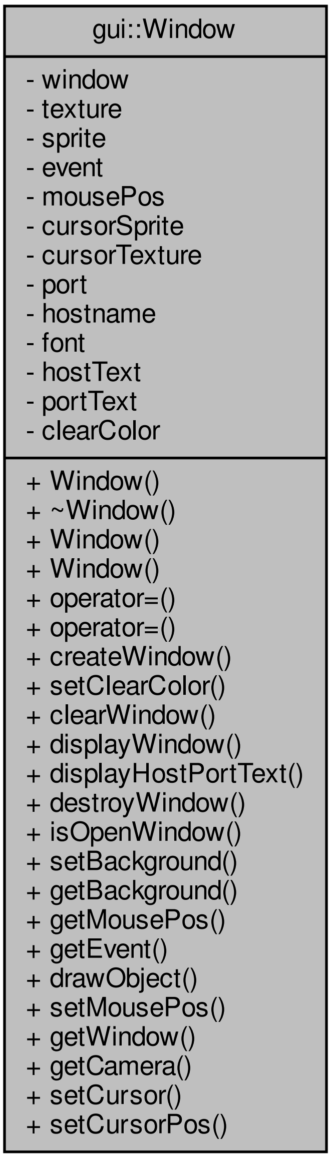
- Map/Tile.hpp

6.54 gui::Window Class Reference

Window class.

```
#include <Window.hpp>
```

Collaboration diagram for gui::Window:



Public Member Functions

- Window ()=default

  Default constructor.
- ∼Window () noexcept=default

  Destroy the Window object.
- Window (const Window &w)=delete

  Construct a new Window object by copy = delete.
- Window (Window &&w)=default

Construct a new Window object by move = delete.

- Window & operator= (const Window &w)=delete

  Overload equal operator by copy = delete.
- Window & operator= (Window &&w)=default

  Assign a Window object by move = default.
- void createWindow (const std::string &title, int width, int height, int port, const std::string &hostname)

  Creates a window in SFML.
- void setClearColor (sf::Color color)

  Set the clear color in SFML.
- void clearWindow ()

  Clear the window in SFML.
- void displayWindow ()

  Display the window in SFML.
- void displayHostPortText ()

  Display the text in SFML.
- void destroyWindow ()

  Destroy the window in SFML.
- bool isOpenWindow ()

  Check if the window is open in SFML.
- void setBackground (const std::string &path)

  Set the background in SFML.
- sf::Sprite & getBackground ()

  Get the background in SFML.
- const sf::Vector2f & getMousePos () const

  Get the mouse position in SFML.
- sf::Event & getEvent ()

  Get the event in SFML.
- template<typename T >
  void drawObject (T &drawable)

  Draw a drawable object in SFML.
- void setMousePos (sf::Vector2f mousePos)

  Set the mouse position in SFML.
- sf::RenderWindow & getWindow ()

  Get the window in SFML.
- Camera & getCamera ()

  Get the camera in SFML.
- void setCursor (float size)

  Set the cursor in SFML.
- void setCursorPos (sf::Vector2f pos)

  Set the Cursor Pos.

Private Attributes

- sf::RenderWindow window
- sf::Texture texture
- sf::Sprite sprite
- sf::Event event
- sf::Vector2f mousePos
- sf::Sprite cursorSprite
- sf::Texture cursorTexture
- int port

- std::string hostname
- sf::Font font
- sf::Text hostText
- sf::Text portText
- sf::Color clearColor

6.54.1 Detailed Description

Window class.

6.54.2 Constructor & Destructor Documentation

6.54.2.1 Window() [1/3]

```
gui::Window::Window ( )  [default]
```

Default constructor.

6.54.2.2 ∼Window()

```
gui::Window::∼Window ( )  [default], [noexcept]
```

Destroy the Window object.

6.54.2.3 Window() [2/3]

```
gui::Window::Window (
             const Window & w )  [delete]
```

Construct a new Window object by copy = delete.

6.54.2.4 Window() [3/3]

```
gui::Window::Window (
             Window && w )  [default]
```

Construct a new Window object by move = delete.

Parameters

w	

6.54.3 Member Function Documentation

6.54.3.1 clearWindow()

```
void gui::Window::clearWindow ( )
```

Clear the window in SFML.

6.54.3.2 createWindow()

```
void gui::Window::createWindow (
             const std::string & title,
             int width,
             int height,
             int port,
             const std::string & hostname )
```

Creates a window in SFML.

Parameters

title	The title of the window.
width	The width of the window.
height	The height of the window.
port	The port of the server.
hostname	The hostname of the server.

Exceptions

Errors::ErrorTexture	
Errors::ErrorFont	

6.54.3.3 destroyWindow()

```
void gui::Window::destroyWindow ( )
```

Destroy the window in SFML.

6.54.3.4 displayHostPortText()

```
void gui::Window::displayHostPortText ( )
```

Display the text in SFML.

6.54.3.5 displayWindow()

```
void gui::Window::displayWindow ( )
```

Display the window in SFML.

6.54.3.6 drawObject()

```
template<typename T >
void gui::Window::drawObject (
             T & drawable )  [inline]
```

Draw a drawable object in SFML.

Parameters

drawable	The drawable object to draw.

6.54.3.7 getBackground()

```
sf::Sprite & gui::Window::getBackground ( )
```

Get the background in SFML.

Returns

sf::Sprite The background.

6.54.3.8 getCamera()

```
Camera& gui::Window::getCamera ( )
```

Get the camera in SFML.

Returns

Camera The camera.

6.54.3.9 getEvent()

```
sf::Event & gui::Window::getEvent ( )
```

Get the event in SFML.

Returns

sf::Event The event.

6.54.3.10 getMousePos()

```
const sf::Vector2f & gui::Window::getMousePos ( ) const
```

Get the mouse position in SFML.

Returns

sf::Vector2f The mouse position.

6.54.3.11 getWindow()

```
sf::RenderWindow & gui::Window::getWindow ( )
```

Get the window in SFML.

Returns

sf::RenderWindow The window.

6.54.3.12 isOpenWindow()

```
bool gui::Window::isOpenWindow ( )
```

Check if the window is open in SFML.

Returns

True if the window is open, False otherwise.

6.54.3.13 operator=() [1/2]

```
Window& gui::Window::operator= (
             const Window & w )  [delete]
```

Overload equal operator by copy = delete.

Parameters

w	

Returns

Window&

6.54.3.14 operator=() [2/2]

```
Window& gui::Window::operator= (
             Window && w )  [default]
```

Assign a Window object by move = default.

Parameters

w	

Returns

Window&

6.54.3.15 setBackground()

```
void gui::Window::setBackground (
             const std::string & path )
```

Set the background in SFML.

Parameters

path	The path of the background.

Exceptions

Errors::ErrorTexture	

6.54.3.16 setClearColor()

```
void gui::Window::setClearColor (
             sf::Color color )
```

Set the clear color in SFML.

Parameters

color	The color to set.

6.54.3.17 setCursor()

```
void gui::Window::setCursor (
             float size )
```

Set the cursor in SFML.

Parameters

size	The size of the cursor.

6.54.3.18 setCursorPos()

```
void gui::Window::setCursorPos (
             sf::Vector2f pos )
```

Set the Cursor Pos.

Parameters

pos	

6.54.3.19 setMousePos()

```
void gui::Window::setMousePos (
             sf::Vector2f mousePos )
```

Set the mouse position in SFML.

Returns

sf::Vector2f The mouse position.

6.54.4 Field Documentation

6.54.4.1 clearColor

```
sf::Color gui::Window::clearColor  [private]
```

The clear color in SFML

6.54.4.2 cursorSprite

```
sf::Sprite gui::Window::cursorSprite  [private]
```

6.54.4.3 cursorTexture

```
sf::Texture gui::Window::cursorTexture  [private]
```

6.54.4.4 event

```
sf::Event gui::Window::event  [private]
```

6.54.4.5 font

```
sf::Font gui::Window::font  [private]
```

6.54.4.6 hostname

```
std::string gui::Window::hostname  [private]
```

6.54.4.7 hostText

```
sf::Text gui::Window::hostText  [private]
```

6.54.4.8 mousePos

```
sf::Vector2f gui::Window::mousePos  [private]
```

6.54.4.9 port

```
int gui::Window::port  [private]
```

6.54.4.10 portText

```
sf::Text gui::Window::portText  [private]
```

6.54.4.11 sprite

```
sf::Sprite gui::Window::sprite  [private]
```

6.54.4.12 texture

```
sf::Texture gui::Window::texture  [private]
```

6.54.4.13 window

```
sf::RenderWindow gui::Window::window  [private]
```

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

- Components/Window/Window.hpp
- Components/Window/Window.cpp

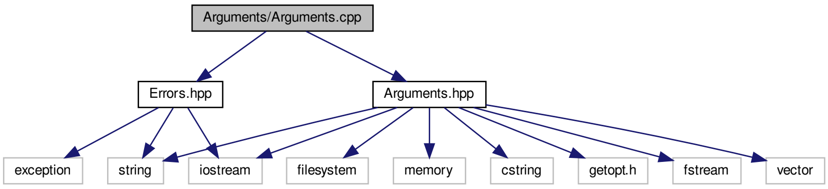
Chapter 7

File Documentation

7.1 Arguments/Arguments.cpp File Reference

```
#include "Arguments.hpp"
#include "Errors.hpp"
```

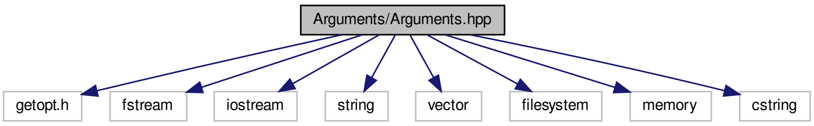
Include dependency graph for Arguments.cpp:



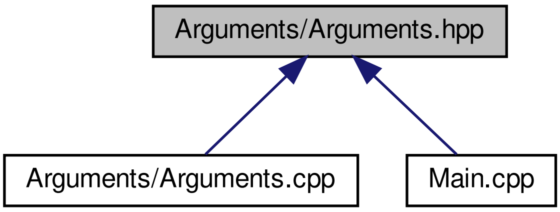
7.2 Arguments/Arguments.hpp File Reference

```
#include <getopt.h>
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
#include <filesystem>
#include <memory>
#include <cstring>
```

Include dependency graph for Arguments.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

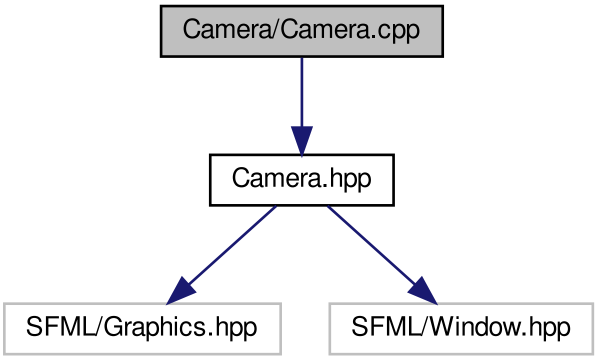
- class Arguments

  Class for the arguments.

7.3 Camera/Camera.cpp File Reference

```
#include "Camera.hpp"
```

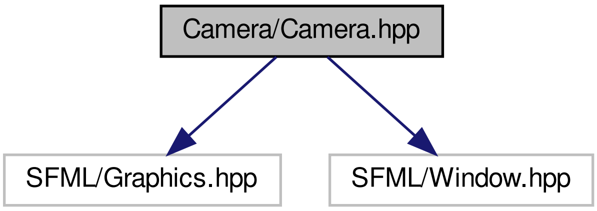
Include dependency graph for Camera.cpp:



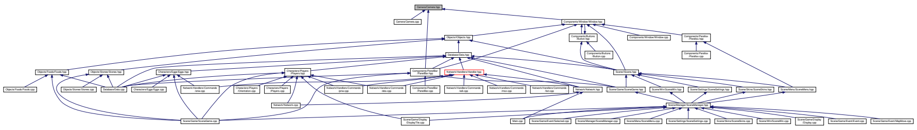
7.4 Camera/Camera.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
```

Include dependency graph for Camera.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class gui::Camera

    Camera class.

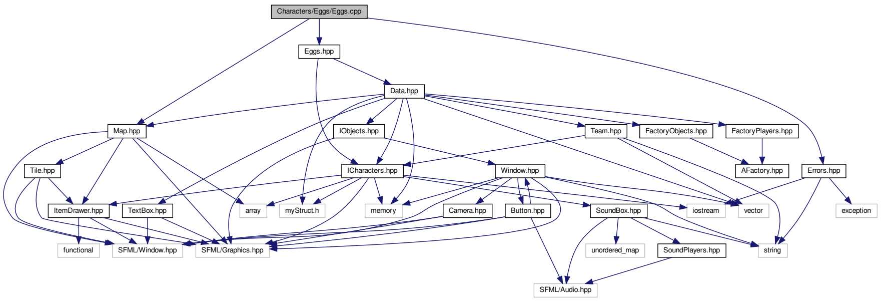
Namespaces

- gui

7.5 Characters/Eggs/Eggs.cpp File Reference

```
#include "Eggs.hpp"
#include "Map.hpp"
#include "Errors.hpp"
```

Include dependency graph for Eggs.cpp:



Functions

- static sf::IntRect findRectEggs (TypeEggs typeParams)

Variables

- static std::vector< sf::IntRect > rectEggs

7.5.1 Function Documentation

7.5.1.1 findRectEggs()

```
static sf::IntRect findRectEggs (
            TypeEggs typeParams )  [static]
```

7.5.2 Variable Documentation

7.5.2.1 rectEggs

```
std::vector<sf::IntRect> rectEggs  [static]
```

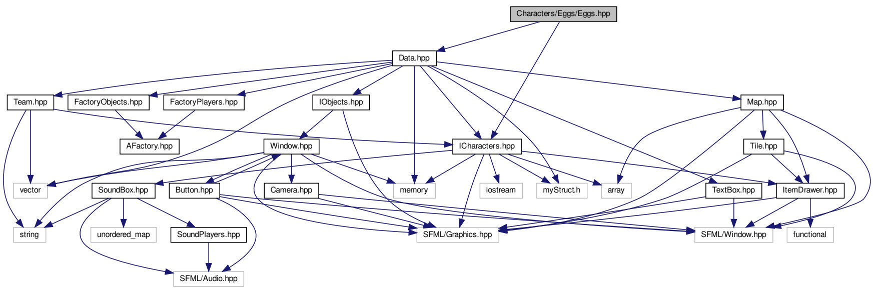
Initial value:

```
= {
    sf::IntRect(95, 6599, 105, 83),
    sf::IntRect(95, 6682, 105, 83),
    sf::IntRect(95, 6765, 105, 83),
    sf::IntRect(95, 6848, 105, 83),
    sf::IntRect(95, 6931, 105, 83),
    sf::IntRect(95, 7014, 105, 83),
    sf::IntRect(95, 7097, 105, 83)
}
```

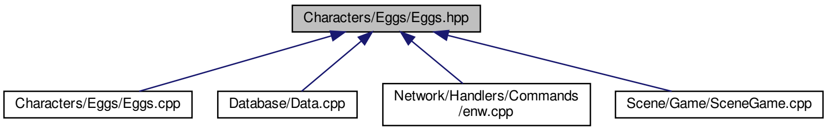
7.6 Characters/Eggs/Eggs.hpp File Reference

```
#include "ICharacters.hpp"
#include "Data.hpp"
```

Include dependency graph for Eggs.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Eggs

    Eggs class.

Enumerations

- enum class TypeEggs {
  POURPRE , BLUE , YELLOW , ICE ,
  PURPLE , GREEN , WHITE }

    TypesEggs enum class.

7.6.1 Enumeration Type Documentation

7.6.1.1 TypeEggs

```
enum TypeEggs  [strong]
```

TypesEggs enum class.

Enumerator

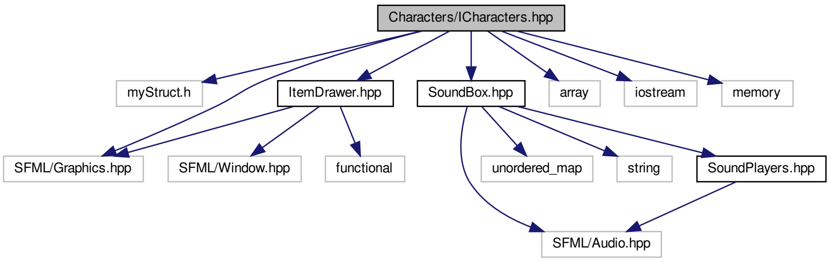
POURPRE	
BLUE	
YELLOW	
ICE	
PURPLE	
GREEN	
WHITE	

7.7 Characters/ICharacters.hpp File Reference

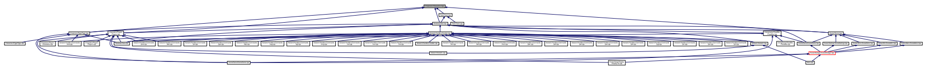
```
#include "myStruct.h"
#include "ItemDrawer.hpp"
```

```
#include "SoundBox.hpp"
#include <SFML/Graphics.hpp>
#include <array>
#include <iostream>
#include <memory>
```

Include dependency graph for ICharacters.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class ICharacters

  Interface for the characters.

Enumerations

- enum class CharacterType { Player , Egg , Commentator }

7.7.1 Enumeration Type Documentation

7.7.1.1 CharacterType

```
enum CharacterType  [strong]
```

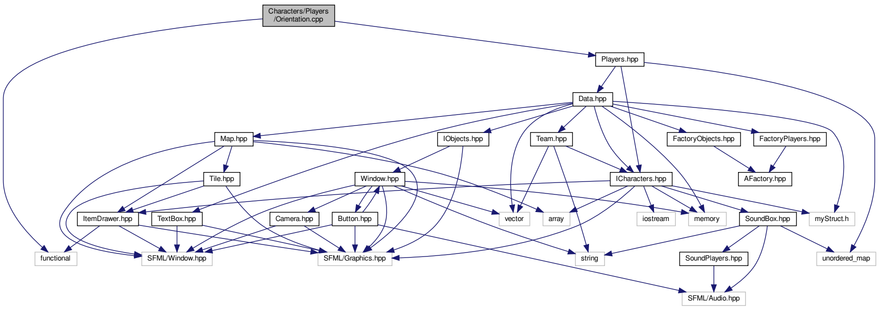
Enumerator

Player	
Egg	
Commentator	

7.8 Characters/Players/Orientation.cpp File Reference

```
#include "Players.hpp"
#include <functional>
```

Include dependency graph for Orientation.cpp:



Functions

- int redRectFunc (Orientation orientation)
- int yellowRectFunc (Orientation orientation)
- int greenRectFunc (Orientation orientation)
- int blueRectFunc (Orientation orientation)
- int darkBlueRectFunc (Orientation orientation)
- int purpleRectFunc (Orientation orientation)
- int getRightRect (const std::string &colorParams, Orientation orientation)
- int getFacingRectByColor (const std::string &colorParams)

7.8.1 Function Documentation

7.8.1.1 blueRectFunc()

```
int blueRectFunc (
             Orientation orientation )
```

7.8.1.2 darkBlueRectFunc()

```
int darkBlueRectFunc (
             Orientation orientation )
```

7.8.1.3 getFacingRectByColor()

```
int getFacingRectByColor (
            const std::string & colorParams )
```

7.8.1.4 getRightRect()

```
int getRightRect (
            const std::string & colorParams,
            Orientation orientation )
```

7.8.1.5 greenRectFunc()

```
int greenRectFunc (
            Orientation orientation )
```

7.8.1.6 purpleRectFunc()

```
int purpleRectFunc (
            Orientation orientation )
```

7.8.1.7 redRectFunc()

```
int redRectFunc (
            Orientation orientation )
```

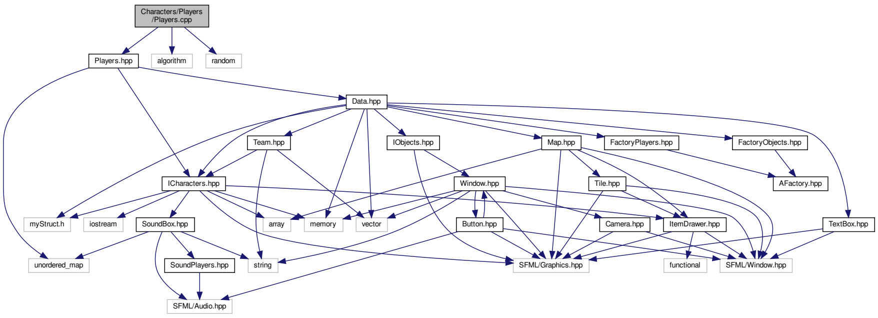
7.8.1.8 yellowRectFunc()

```
int yellowRectFunc (
            Orientation orientation )
```

7.9 Characters/Players/Players.cpp File Reference

```
#include "Players.hpp"
#include <algorithm>
#include <random>
```

Include dependency graph for Players.cpp:



Functions

- int getRightRect (const std::string &color, Orientation orientation)

Variables

- constexpr char PATH_GOLEM [ ] = "./gui/Resources/golem_spritesheet.png"

7.9.1 Function Documentation

7.9.1.1 getRightRect()

```
int getRightRect (
            const std::string & color,
            Orientation orientation )
```

7.9.2 Variable Documentation

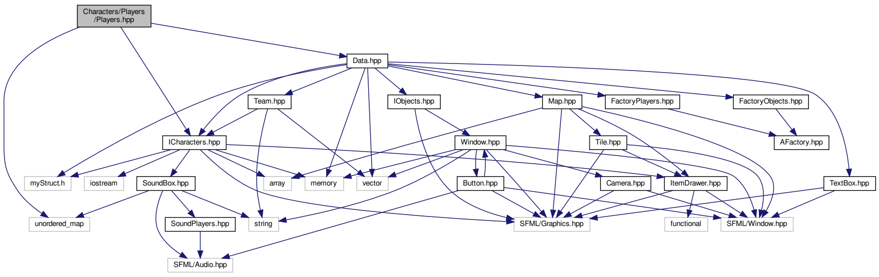
7.9.2.1 PATH_GOLEM

```
constexpr char PATH_GOLEM[ ] = "./gui/Resources/golem_spritesheet.png"  [constexpr]
```

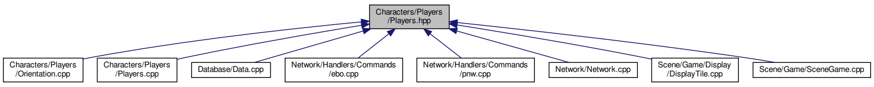
7.10 Characters/Players/Players.hpp File Reference

```
#include "ICharacters.hpp"
#include "Data.hpp"
#include <unordered_map>
```

Include dependency graph for Players.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Players

  Players class.

Variables

- static std::unordered_map< std::string, sf::Color > broadcastColor

7.10.1 Variable Documentation

7.10.1.1 broadcastColor

```
std::unordered_map<std::string, sf::Color> broadcastColor  [static]
```

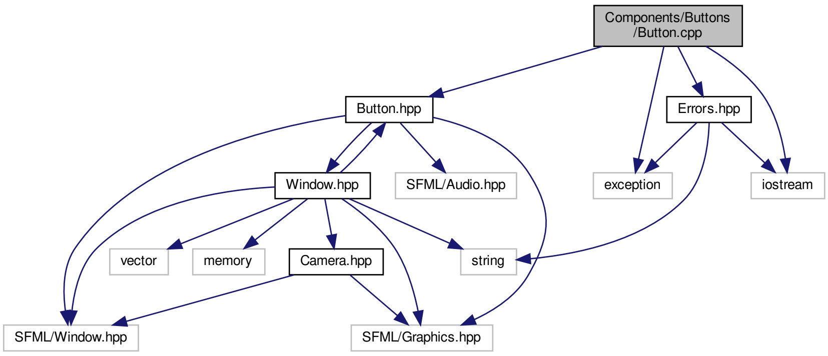
Initial value:

```
= {
    {"red", sf::Color(242, 14, 27, 155)},
    {"yellow", sf::Color(242, 228, 24, 155)},
    {"green", sf::Color(34, 239, 23, 155)},
    {"blue", sf::Color(32, 173, 240, 155)},
    {"darkBlue", sf::Color(17, 21, 238, 155)},
    {"purple", sf::Color(183, 21, 243, 155)},
}
```

7.11 Components/Buttons/Button.cpp File Reference

```
#include "Button.hpp"
#include <exception>
#include <iostream>
#include "Errors.hpp"
```

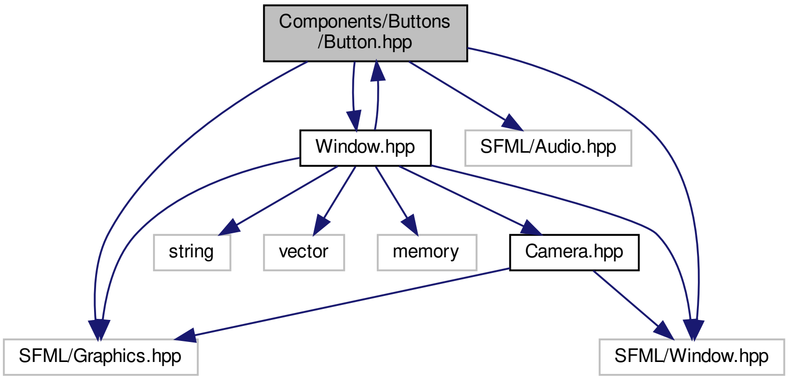
Include dependency graph for Button.cpp:



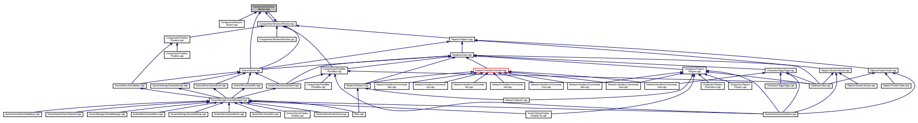
7.12 Components/Buttons/Button.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include "Window.hpp"
```

Include dependency graph for Button.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class gui::Button

  Button class.

Namespaces

- gui

Enumerations

- enum ButtonState { BUTTON_IDLE = 0 , BUTTON_HOVER , BUTTON_CLICKED , BUTTON_LOCKED }

7.12.1 Enumeration Type Documentation

7.12.1.1 ButtonState

```
enum ButtonState
```

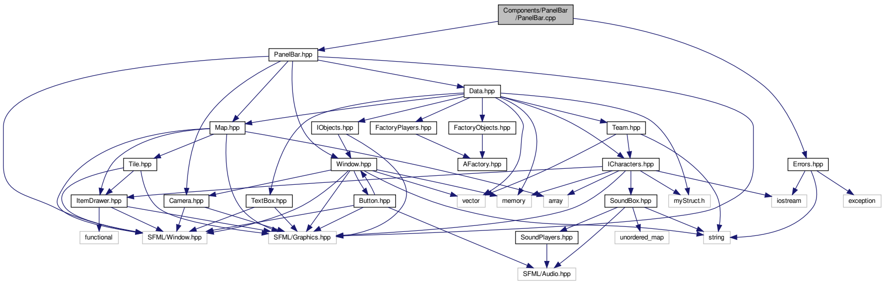
Enumerator

BUTTON_IDLE	
BUTTON_HOVER	
BUTTON_CLICKED	
BUTTON_LOCKED	

7.13 Components/PanelBar/PanelBar.cpp File Reference

```
#include "PanelBar.hpp"
#include "Errors.hpp"
```

Include dependency graph for PanelBar.cpp:



Functions

- int getFacingRectByColor (const std::string &color)

7.13.1 Function Documentation

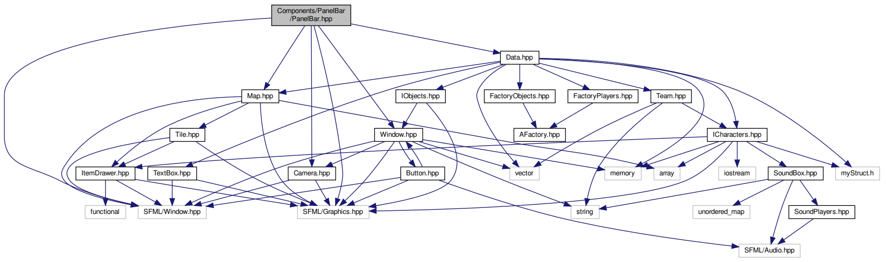
7.13.1.1 getFacingRectByColor()

```
int getFacingRectByColor (
            const std::string & color )
```

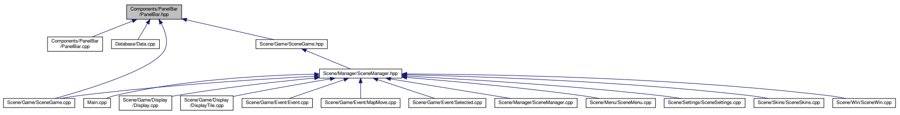
7.14 Components/PanelBar/PanelBar.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include "Camera.hpp"
#include "Map.hpp"
#include "Window.hpp"
#include "Data.hpp"
```

Include dependency graph for PanelBar.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class gui::PanelBar

  PanelBar class.

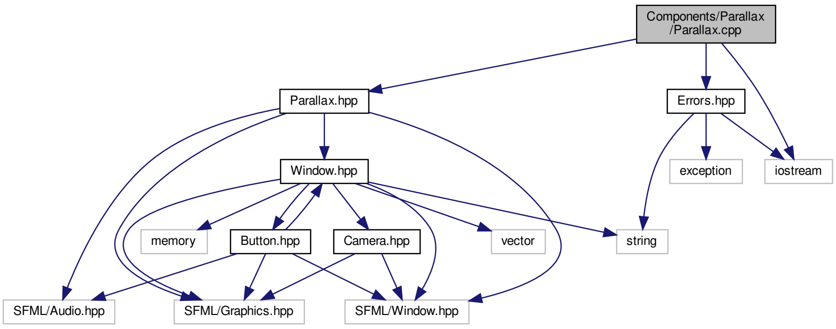
Namespaces

- gui

7.15 Components/Parallax/Parallax.cpp File Reference

```
#include "Parallax.hpp"
#include <iostream>
#include "Errors.hpp"
```

Include dependency graph for Parallax.cpp:

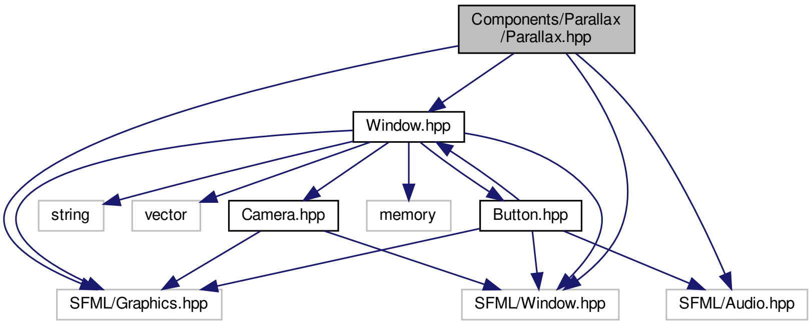


7.16 Components/Parallax/Parallax.hpp File Reference

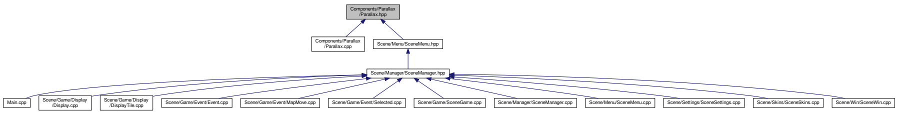
```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
```

```
#include "Window.hpp"
```

Include dependency graph for Parallax.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct Background

  Background struct.
- class gui::Parallax

  Parallax class.

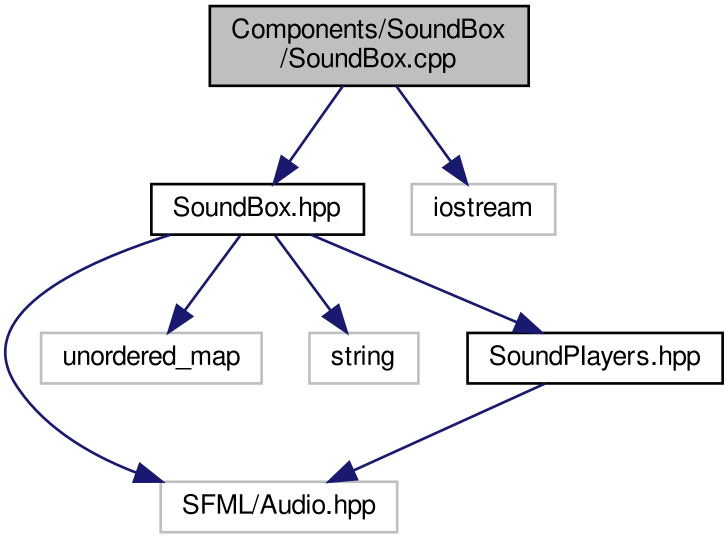
Namespaces

- gui

7.17 Components/SoundBox/SoundBox.cpp File Reference

```
#include "SoundBox.hpp"
#include <iostream>
```

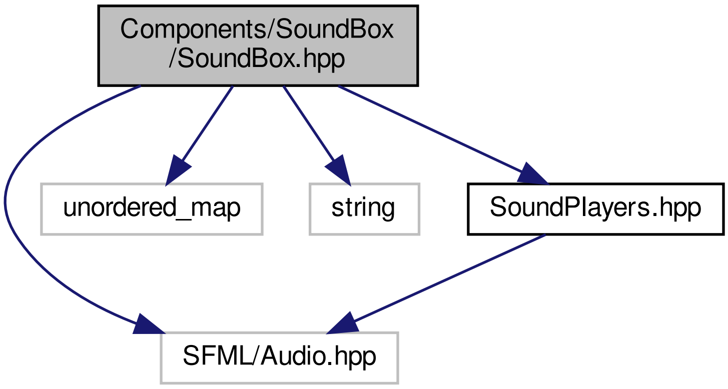
Include dependency graph for SoundBox.cpp:



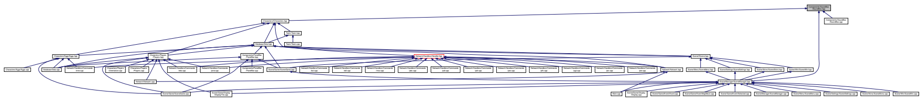
7.18 Components/SoundBox/SoundBox.hpp File Reference

```
#include <SFML/Audio.hpp>
#include <unordered_map>
#include <string>
#include "SoundPlayers.hpp"
```

Include dependency graph for SoundBox.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct Song

    Song struct.

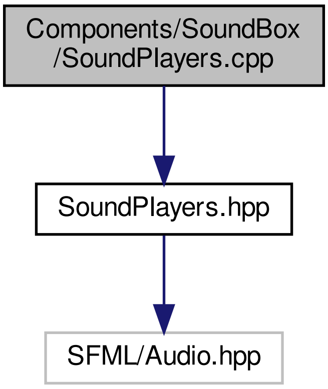
- class SoundBox

    SoundBox class.

7.19 Components/SoundBox/SoundPlayers.cpp File Reference

```
#include "SoundPlayers.hpp"
```

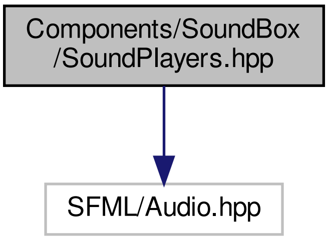
Include dependency graph for SoundPlayers.cpp:



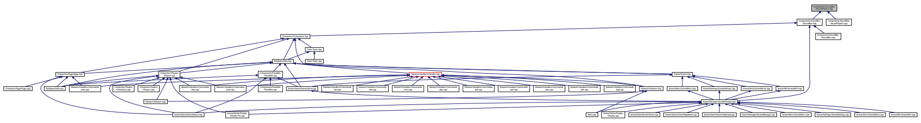
7.20 Components/SoundBox/SoundPlayers.hpp File Reference

```
#include "SFML/Audio.hpp"
```

Include dependency graph for SoundPlayers.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

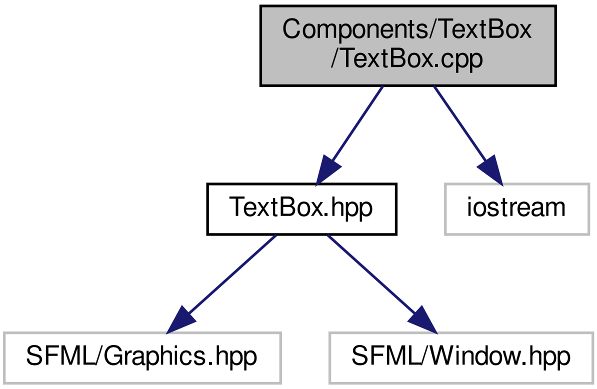
- class SoundPlayer

  SoundPlayer class.

7.21 Components/TextBox/TextBox.cpp File Reference

```
#include "TextBox.hpp"
#include <iostream>
```

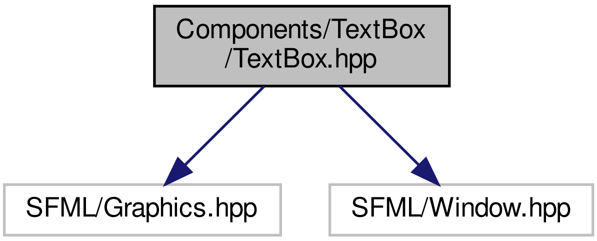
Include dependency graph for TextBox.cpp:



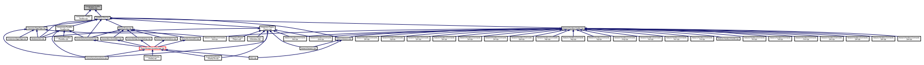
7.22 Components/TextBox/TextBox.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
```

Include dependency graph for TextBox.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

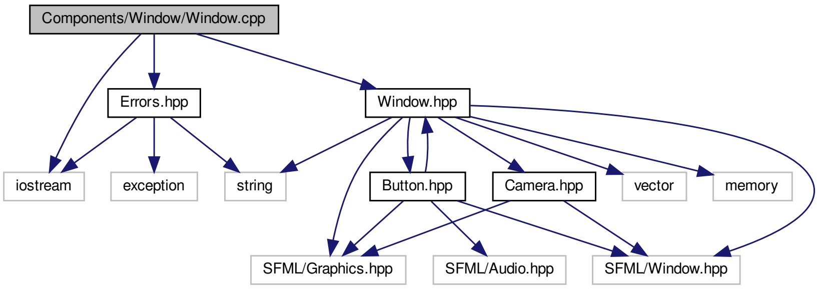
- class TextBox

  TextBox class.

7.23 Components/Window/Window.cpp File Reference

```
#include "Window.hpp"
#include <iostream>
#include "Errors.hpp"
```

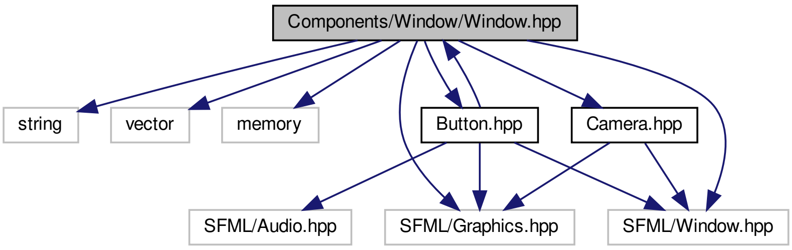
Include dependency graph for Window.cpp:



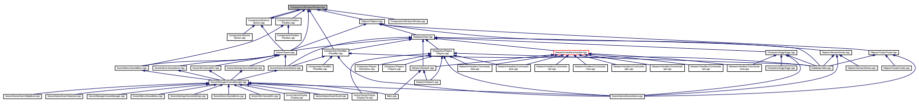
7.24 Components/Window/Window.hpp File Reference

```
#include <string>
#include <vector>
#include <memory>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include "Button.hpp"
#include "Camera.hpp"
```

Include dependency graph for Window.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class gui::Window

    Window class.

Namespaces

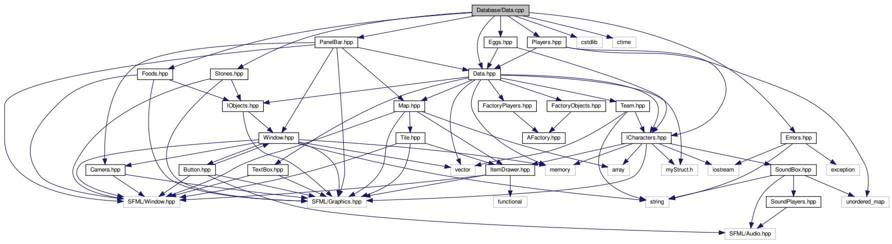
- gui

7.25 Database/Data.cpp File Reference

```
#include "Data.hpp"
#include "Foods.hpp"
#include "Stones.hpp"
#include "PanelBar.hpp"
#include "Players.hpp"
#include "Eggs.hpp"
#include <cstdlib>
```

```
#include <ctime>
#include "Errors.hpp"
```

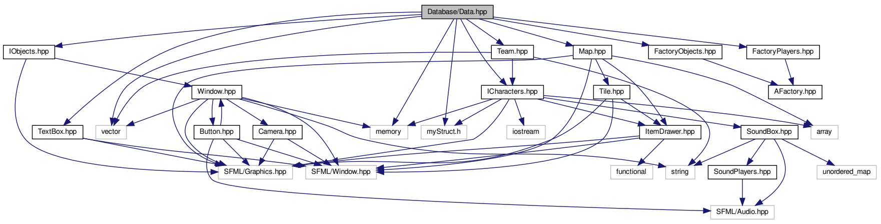
Include dependency graph for Data.cpp:



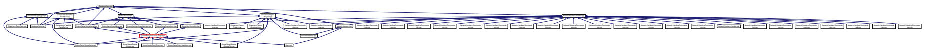
7.26 Database/Data.hpp File Reference

```
#include <vector>
#include <memory>
#include "IObjects.hpp"
#include "ICharacters.hpp"
#include "Map.hpp"
#include "FactoryObjects.hpp"
#include "FactoryPlayers.hpp"
#include "myStruct.h"
#include "Team.hpp"
#include "TextBox.hpp"
```

Include dependency graph for Data.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class gui::Data

  Data class.

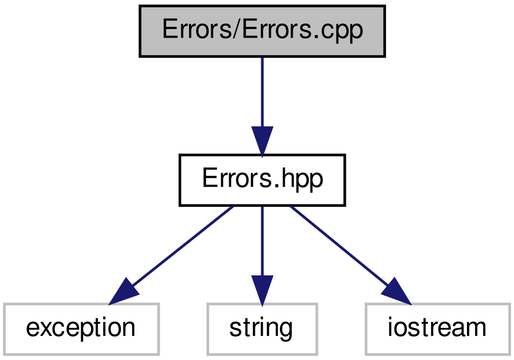
Namespaces

- gui

7.27 Errors/Errors.cpp File Reference

```
#include "Errors.hpp"
```

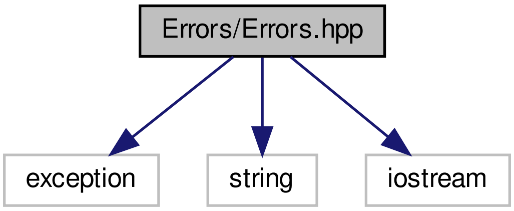
Include dependency graph for Errors.cpp:



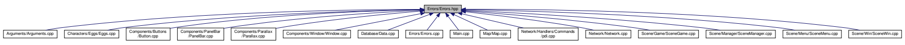
7.28 Errors/Errors.hpp File Reference

```
#include <exception>
#include <string>
#include <iostream>
```

Include dependency graph for Errors.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Errors
- class Errors::ErrorTeam

  A class for Team Error.
- class Errors::ErrorTexture

  A class for Texture Error.
- class Errors::ErrorUsage

  A class for Usage Error.
- class Errors::ErrorFile

  A class for file Error.
- class Errors::ErrorOpen

  A class for open Error.
- class Errors::ErrorCharacter

  A class for Character Error.
- class Errors::ErrorEnv

  A class for env Error.
- class Errors::ErrorFont

  A class for font Error.
- class Errors::ErrorSocket

  A class for socket Error.
- class Errors::ErrorBroadcast

  A class for broadcast Error.
- class Errors::ErrorElevate

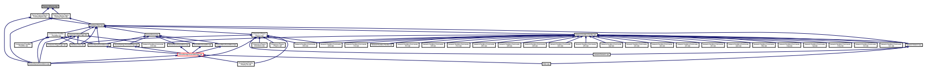
  A class for Elevate Error.
- class Errors::ErrorColor
- class Errors::ErrorInventory

  A class for Inventory Error.
- class Errors::ErrorButton

  A class for Button Error.

7.29 Factory/AFactory.hpp File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

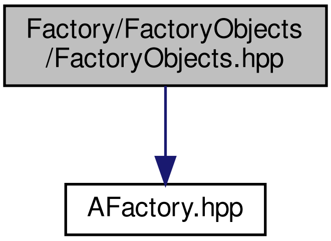
- class AFactory

  AFactory class.

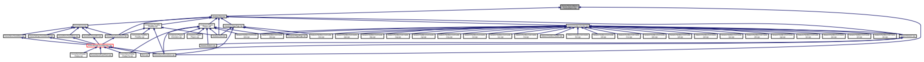
7.30 Factory/FactoryObjects/FactoryObjects.hpp File Reference

```
#include "AFactory.hpp"
```

Include dependency graph for FactoryObjects.hpp:



This graph shows which files directly or indirectly include this file:



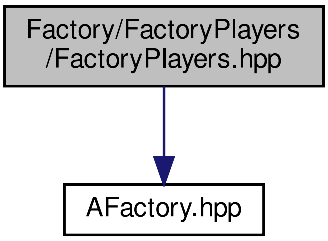
Data Structures

- class FactoryObjects

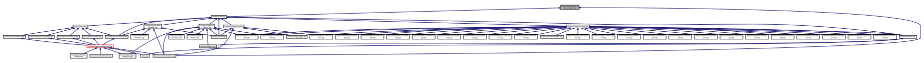
7.31 Factory/FactoryPlayers/FactoryPlayers.hpp File Reference

```
#include "AFactory.hpp"
```

Include dependency graph for FactoryPlayers.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

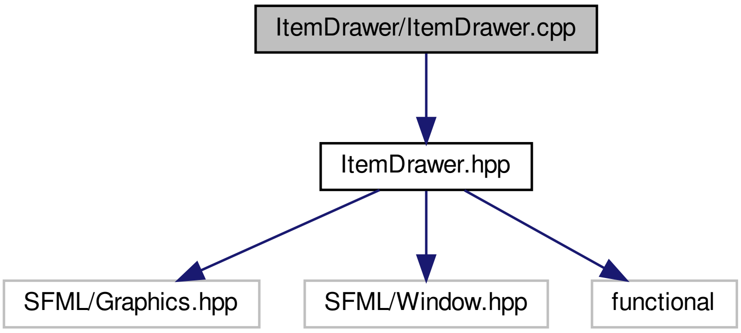
- class FactoryPlayers

  FactoryPlayers class.

7.32 ItemDrawer/ItemDrawer.cpp File Reference

```
#include "ItemDrawer.hpp"
```

Include dependency graph for ItemDrawer.cpp:



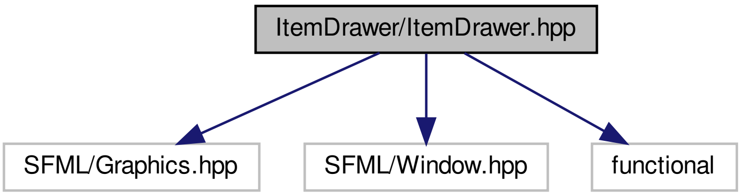
Namespaces

- gui

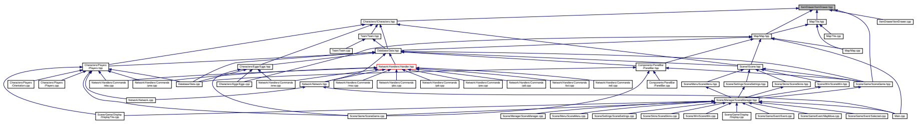
7.33 ItemDrawer/ItemDrawer.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <functional>
```

Include dependency graph for ItemDrawer.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class gui::ItemDrawer

  ItemDrawer class.

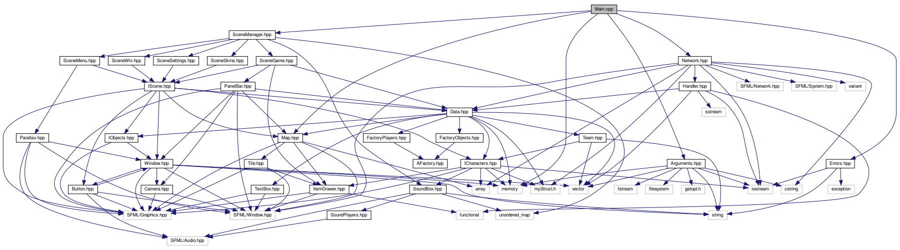
Namespaces

- gui

7.34 Main.cpp File Reference

```
#include "Map.hpp"
#include "SceneManager.hpp"
#include "Arguments.hpp"
#include "Network.hpp"
#include <memory>
#include "Errors.hpp"
```

Include dependency graph for Main.cpp:



Functions

- int launch (gui::Network &client, Arguments &args)
- int main (const int argc, char ∗const argv[ ])

7.34.1 Function Documentation

7.34.1.1 launch()

```
int launch (
            gui::Network & client,
            Arguments & args )
```

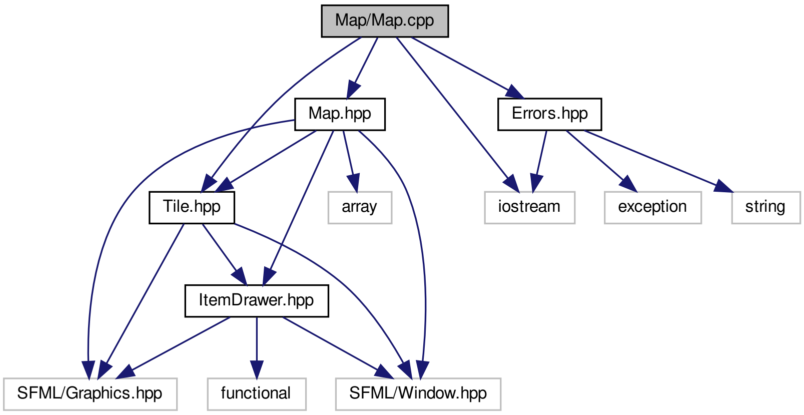
7.34.1.2 main()

```
int main (
            const int argc,
            char *const argv[] )
```

7.35 Map/Map.cpp File Reference

```
#include "Map.hpp"
#include "Tile.hpp"
#include <iostream>
#include "Errors.hpp"
```

Include dependency graph for Map.cpp:



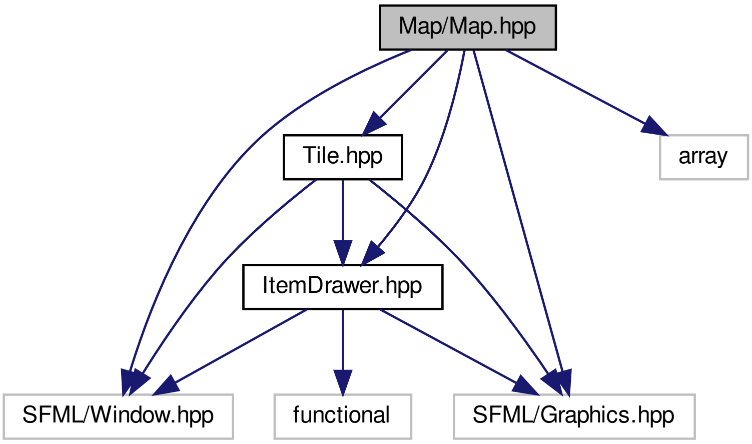
Namespaces

- gui

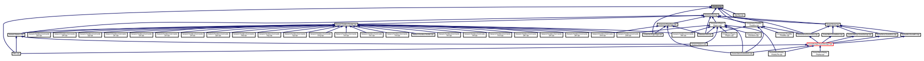
7.36 Map/Map.hpp File Reference

```
#include "ItemDrawer.hpp"
#include "Tile.hpp"
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <array>
```

Include dependency graph for Map.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class gui::Map

  Map Class.

Namespaces

- gui

Variables

- constexpr int RECT_GRASS = 64
- constexpr char PATH_BACK [ ] = "./gui/Resources/tiles.png"

7.36.1 Variable Documentation

7.36.1.1 PATH_BACK

```
constexpr char PATH_BACK[ ] = "./gui/Resources/tiles.png"  [constexpr]
```

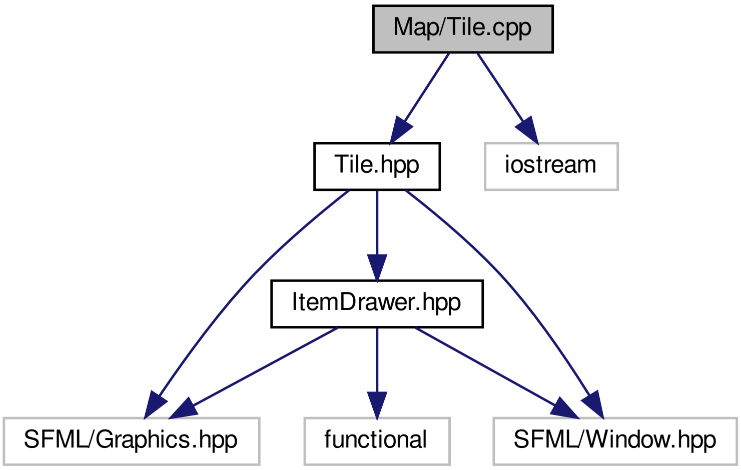
7.36.1.2 RECT_GRASS

```
constexpr int RECT_GRASS = 64  [constexpr]
```

7.37 Map/Tile.cpp File Reference

```
#include "Tile.hpp"
#include <iostream>
```

Include dependency graph for Tile.cpp:



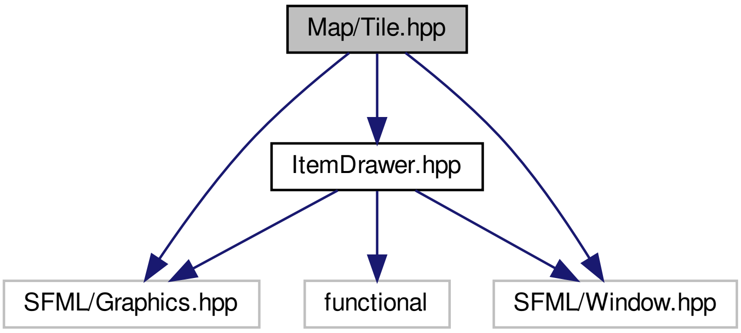
Namespaces

- gui

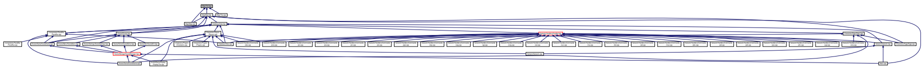
7.38 Map/Tile.hpp File Reference

```
#include "ItemDrawer.hpp"
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
```

Include dependency graph for Tile.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct infos
- class gui::Tile

Namespaces

- gui

Variables

- constexpr float TILE_SIZE = 64

7.38.1 Variable Documentation

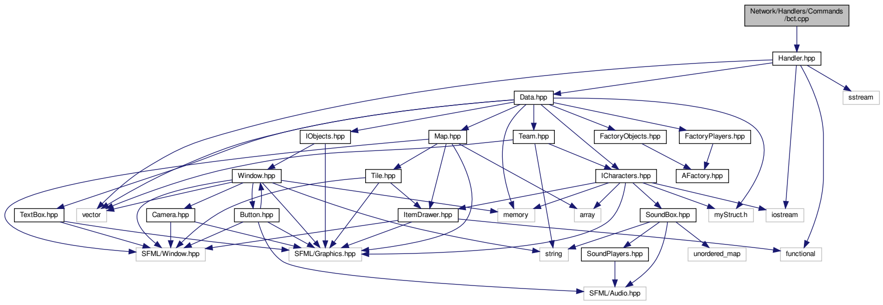
7.38.1.1 TILE_SIZE

```
constexpr float TILE_SIZE = 64  [constexpr]
```

7.39 Network/Handlers/Commands/bct.cpp File Reference

```
#include "Handler.hpp"
```

Include dependency graph for bct.cpp:



Functions

- std::size_t get_char_occurrence (std::string &str, char c)

7.39.1 Function Documentation

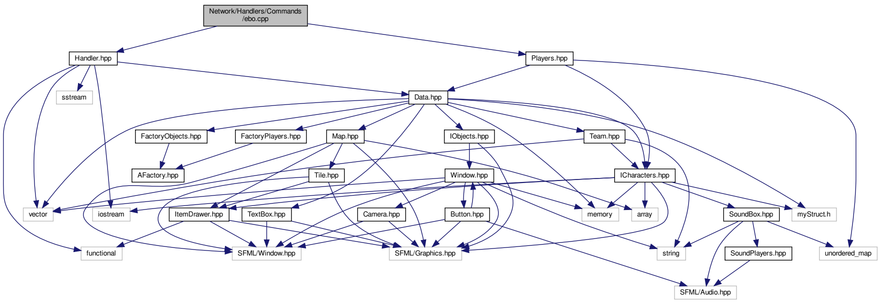
7.39.1.1 get_char_occurrence()

```
std::size_t get_char_occurrence (
            std::string & str,
            char c )
```

7.40 Network/Handlers/Commands/ebo.cpp File Reference

```
#include "Handler.hpp"
#include "Players.hpp"
```

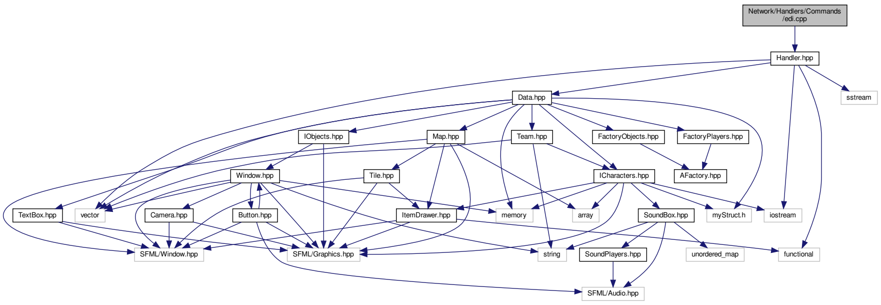
Include dependency graph for ebo.cpp:



7.41 Network/Handlers/Commands/edi.cpp File Reference

```
#include "Handler.hpp"
```

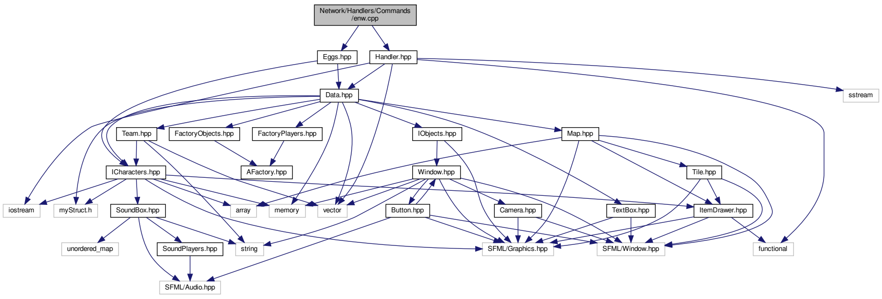
Include dependency graph for edi.cpp:



7.42 Network/Handlers/Commands/enw.cpp File Reference

```
#include "Handler.hpp"
#include "Eggs.hpp"
```

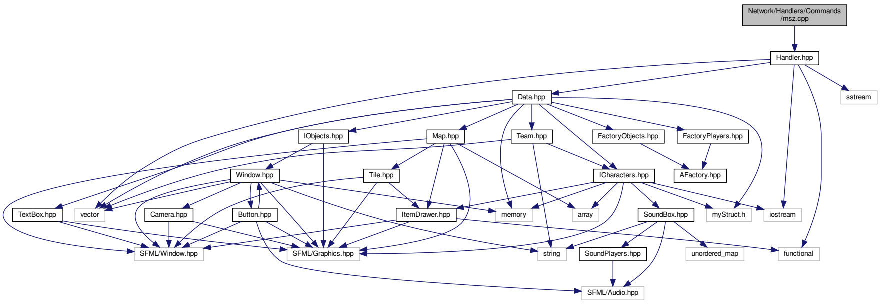
Include dependency graph for enw.cpp:



7.43 Network/Handlers/Commands/msz.cpp File Reference

```
#include "Handler.hpp"
```

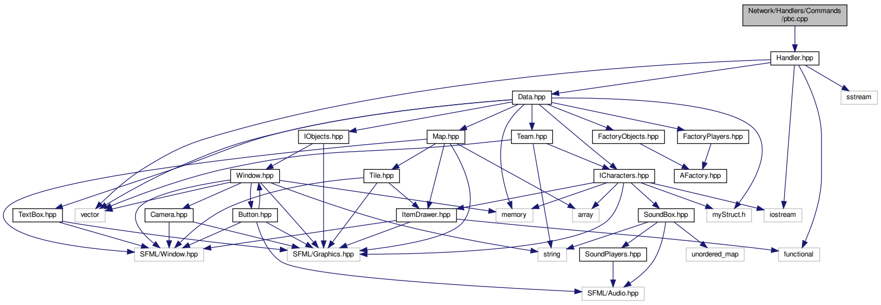
Include dependency graph for msz.cpp:



7.44 Network/Handlers/Commands/pbc.cpp File Reference

```
#include "Handler.hpp"
```

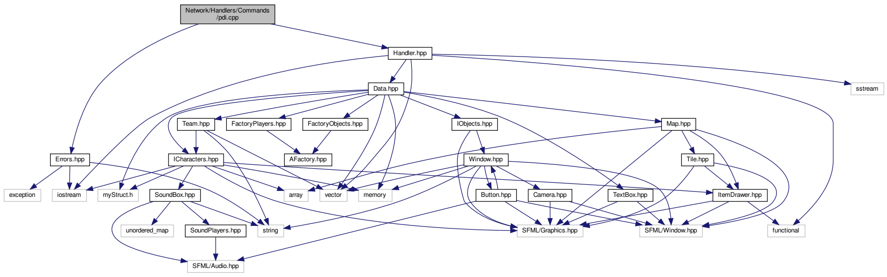
Include dependency graph for pbc.cpp:



7.45 Network/Handlers/Commands/pdi.cpp File Reference

```
#include "Handler.hpp"
#include "Errors.hpp"
```

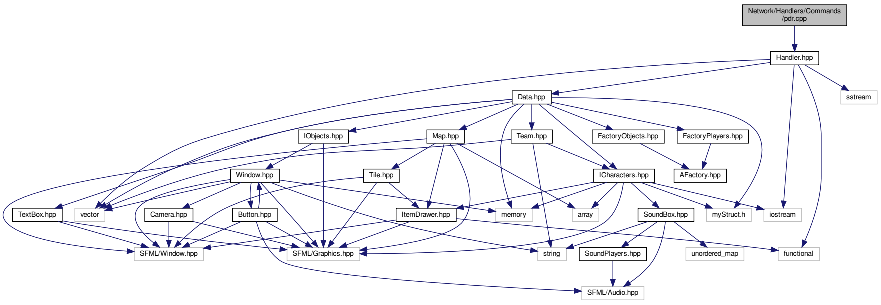
Include dependency graph for pdi.cpp:



7.46 Network/Handlers/Commands/pdr.cpp File Reference

```
#include "Handler.hpp"
```

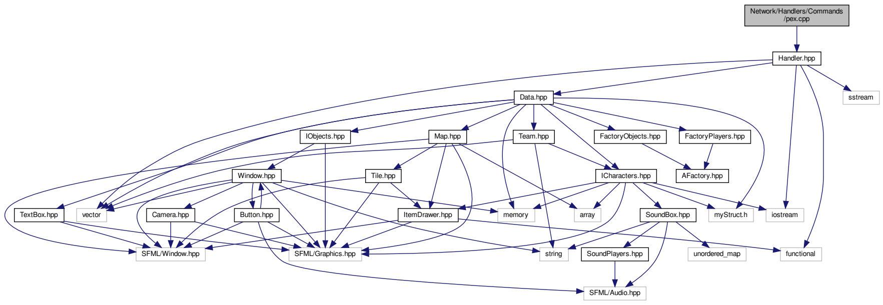
Include dependency graph for pdr.cpp:



7.47 Network/Handlers/Commands/pex.cpp File Reference

```
#include "Handler.hpp"
```

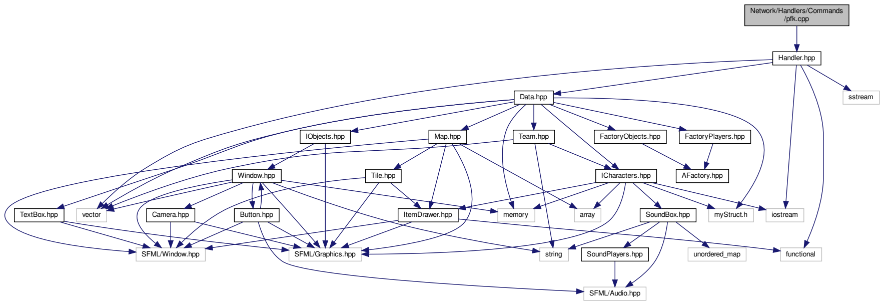
Include dependency graph for pex.cpp:



7.48 Network/Handlers/Commands/pfk.cpp File Reference

```
#include "Handler.hpp"
```

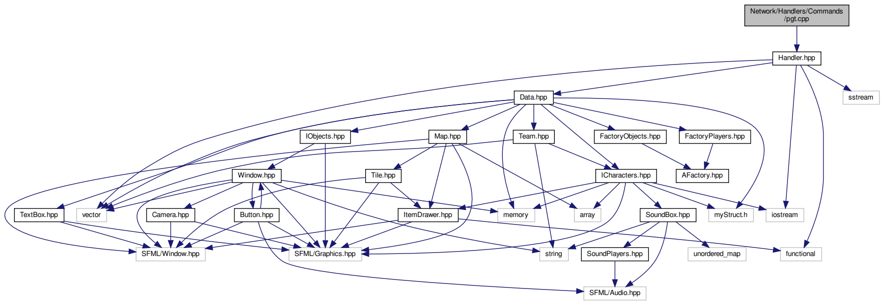
Include dependency graph for pfk.cpp:



7.49 Network/Handlers/Commands/pgt.cpp File Reference

```
#include "Handler.hpp"
```

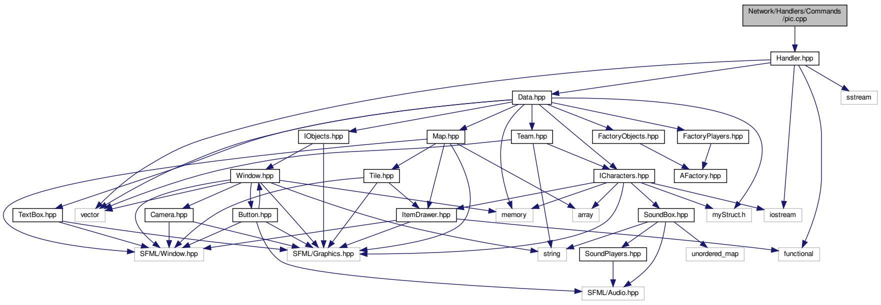
Include dependency graph for pgt.cpp:



7.50 Network/Handlers/Commands/pic.cpp File Reference

```
#include "Handler.hpp"
```

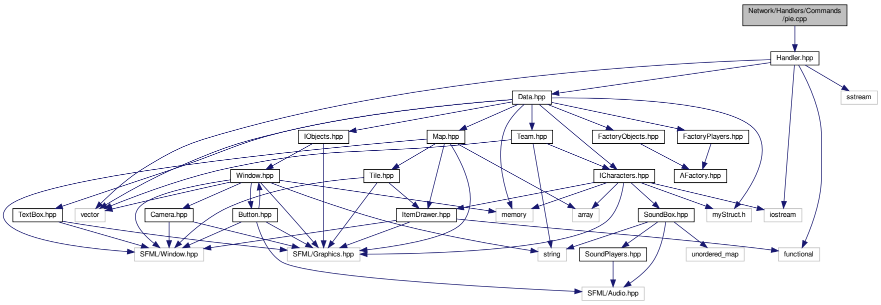
Include dependency graph for pic.cpp:



7.51 Network/Handlers/Commands/pie.cpp File Reference

```
#include "Handler.hpp"
```

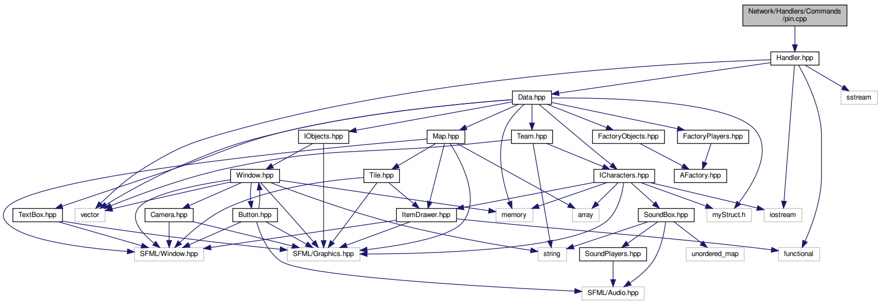
Include dependency graph for pie.cpp:



7.52 Network/Handlers/Commands/pin.cpp File Reference

```
#include "Handler.hpp"
```

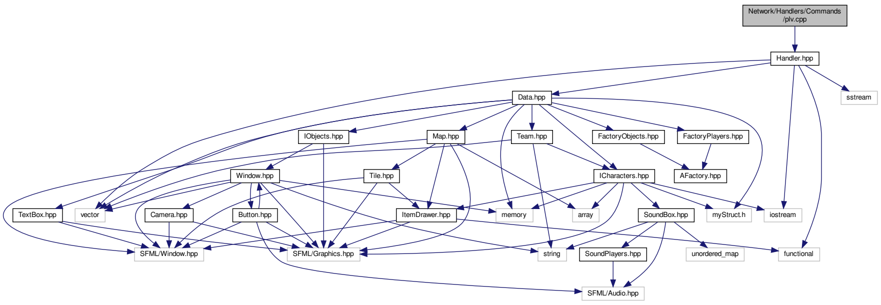
Include dependency graph for pin.cpp:



7.53 Network/Handlers/Commands/plv.cpp File Reference

```
#include "Handler.hpp"
```

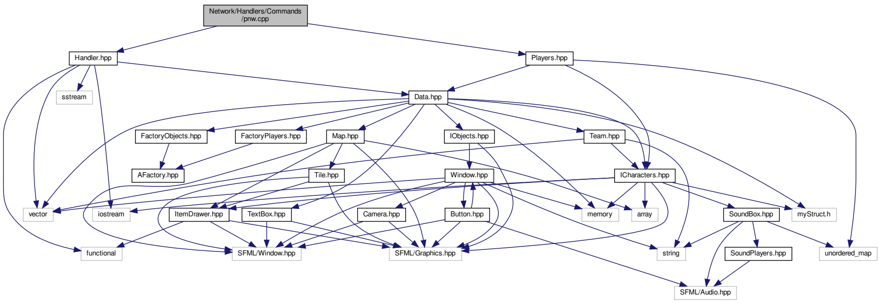
Include dependency graph for plv.cpp:



7.54 Network/Handlers/Commands/pnw.cpp File Reference

```
#include "Handler.hpp"
#include "Players.hpp"
```

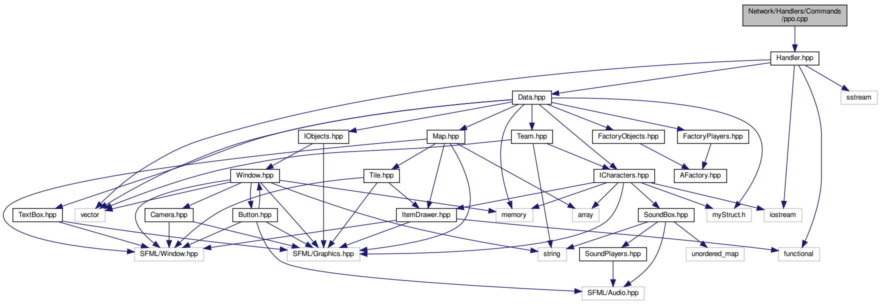
Include dependency graph for pnw.cpp:



7.55 Network/Handlers/Commands/ppo.cpp File Reference

```
#include "Handler.hpp"
```

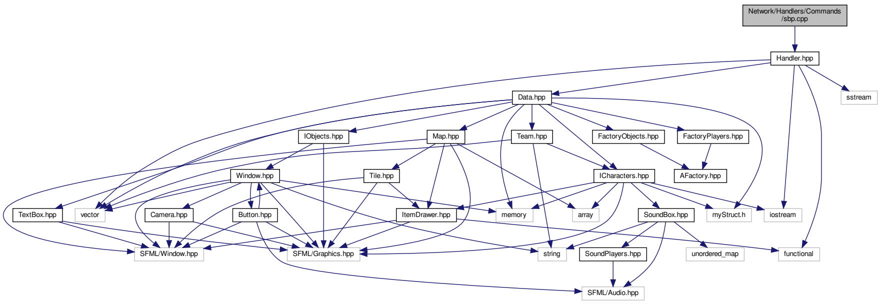
Include dependency graph for ppo.cpp:



7.56 Network/Handlers/Commands/sbp.cpp File Reference

`#include "Handler.hpp"`

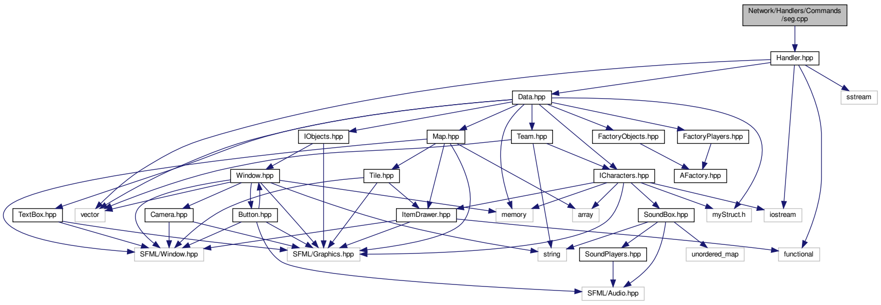
Include dependency graph for sbp.cpp:



7.57 Network/Handlers/Commands/seg.cpp File Reference

`#include "Handler.hpp"`

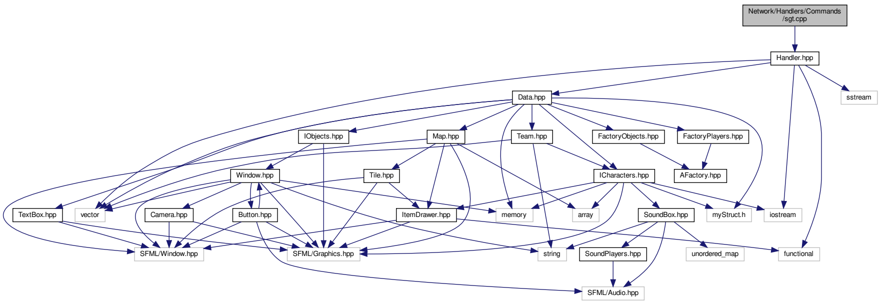
Include dependency graph for seg.cpp:



7.58 Network/Handlers/Commands/sgt.cpp File Reference

`#include "Handler.hpp"`

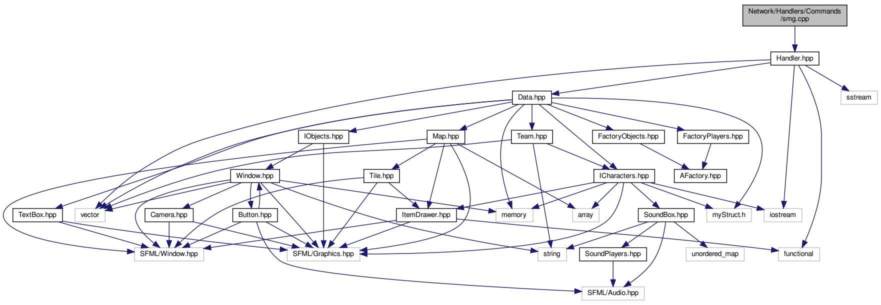
Include dependency graph for sgt.cpp:



7.59 Network/Handlers/Commands/smg.cpp File Reference

```
#include "Handler.hpp"
```

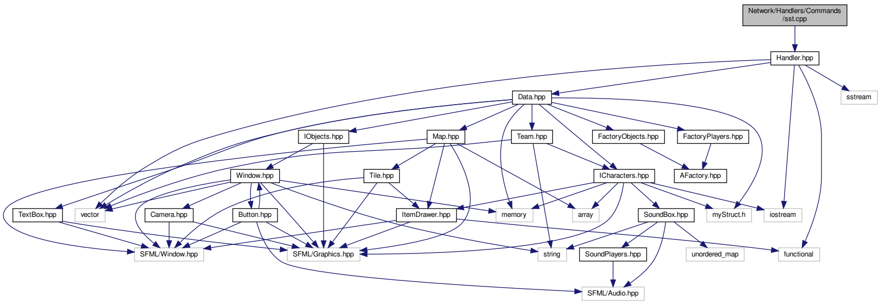
Include dependency graph for smg.cpp:



7.60 Network/Handlers/Commands/sst.cpp File Reference

```
#include "Handler.hpp"
```

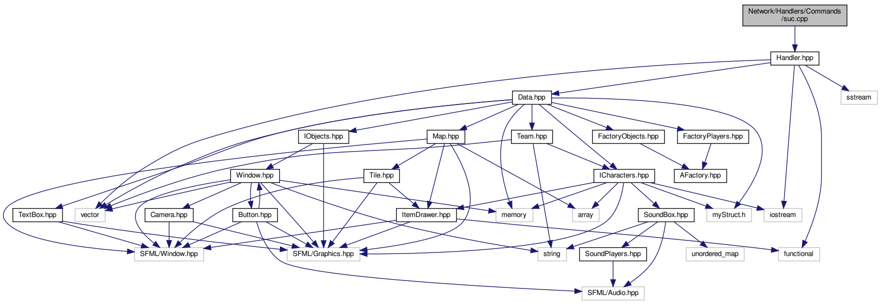
Include dependency graph for sst.cpp:



7.61 Network/Handlers/Commands/suc.cpp File Reference

```
#include "Handler.hpp"
```

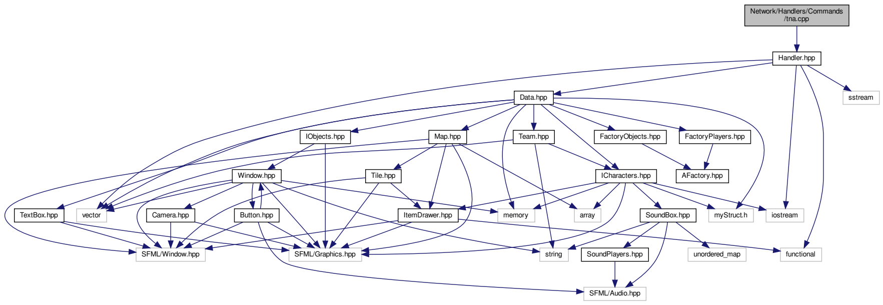
Include dependency graph for suc.cpp:



7.62 Network/Handlers/Commands/tna.cpp File Reference

`#include "Handler.hpp"`

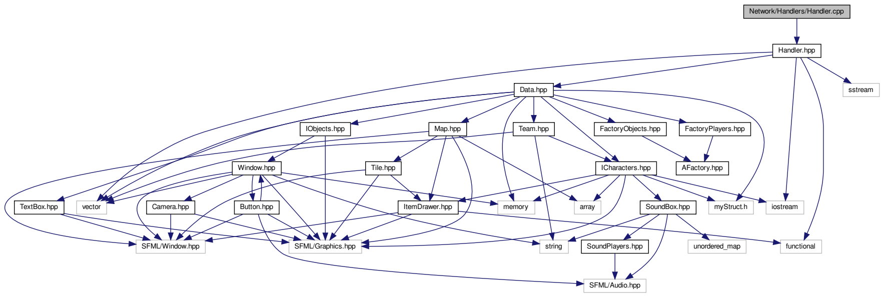
Include dependency graph for tna.cpp:



7.63 Network/Handlers/Handler.cpp File Reference

`#include "Handler.hpp"`

Include dependency graph for Handler.cpp:



Functions

- std::istream & operator>> (std::istream &is, Orientation &orientation)

  operator<< overload for Orientation

7.63.1 Function Documentation

7.63.1.1 operator>>()

```
std::istream& operator>> (
            std::istream & is,
            Orientation & orientation )
```

operator<< overload for Orientation

Parameters

is	
orientation	

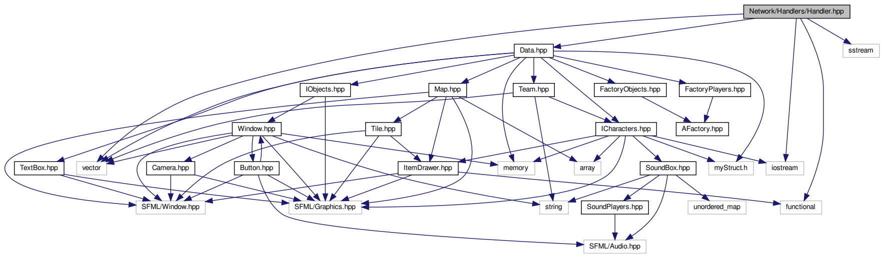
Returns

std::istream&

7.64 Network/Handlers/Handler.hpp File Reference

```
#include <vector>
#include <functional>
#include <sstream>
#include <iostream>
#include "Data.hpp"
```

Include dependency graph for Handler.hpp:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct Command

  Command struct.
- class gui::Handler

  Handler class.

Namespaces

- gui

Functions

- std::istream & operator>> (std::istream &is, Orientation &orientation)

  operator<< overload for Orientation

7.64.1 Function Documentation

7.64.1.1 operator>>()

```
std::istream& operator>> (
            std::istream & is,
            Orientation & orientation )
```

operator<< overload for Orientation

Parameters

is	
orientation	

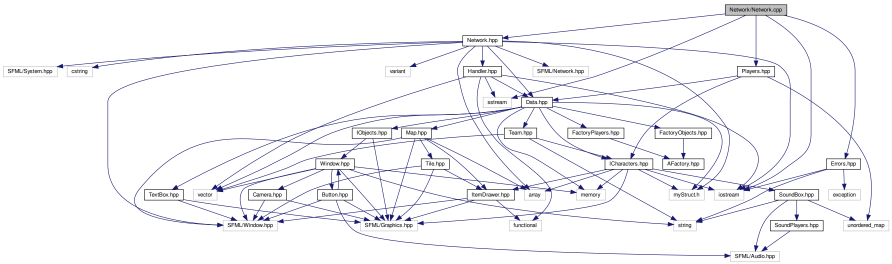
Returns

std::istream&

7.65 Network/Network.cpp File Reference

```
#include "Network.hpp"
#include <iostream>
#include <sstream>
#include "Players.hpp"
#include "Errors.hpp"
```

Include dependency graph for Network.cpp:

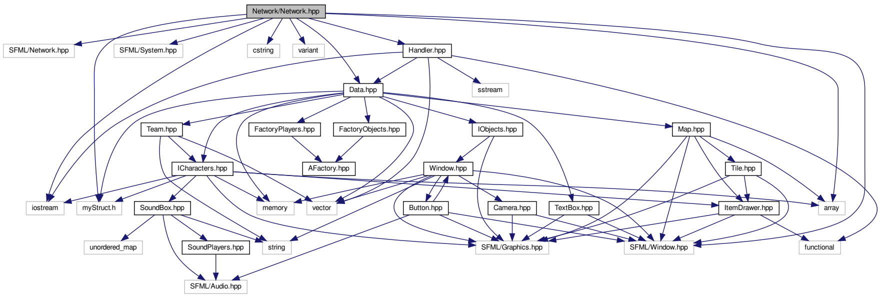


7.66 Network/Network.hpp File Reference

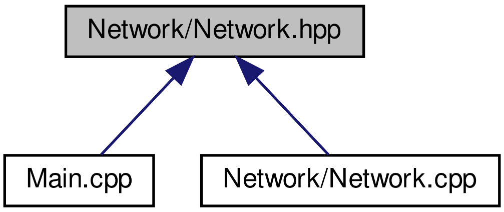
```
#include <SFML/Network.hpp>
#include <SFML/Window.hpp>
#include <SFML/System.hpp>
#include <iostream>
#include <cstring>
```

```
#include <variant>
#include "Data.hpp"
#include <array>
#include "myStruct.h"
#include "Handler.hpp"
```

Include dependency graph for Network.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class gui::Network

  Network class.

Namespaces

- gui

Variables

- constexpr std::size_t maxBufferSize = 4092

  max buffer size @const std::size_t maxBufferSize

7.66.1 Variable Documentation

7.66.1.1 maxBufferSize

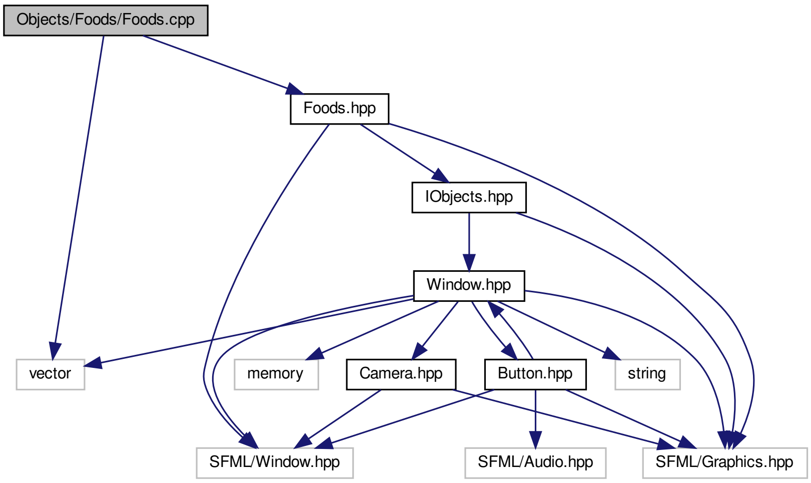
```
constexpr std::size_t maxBufferSize = 4092  [constexpr]
```

max buffer size @const std::size_t maxBufferSize

7.67 Objects/Foods/Foods.cpp File Reference

```
#include "Foods.hpp"
#include <vector>
```

Include dependency graph for Foods.cpp:



Functions

- sf::IntRect findRect ()

Variables

- static std::vector< sf::IntRect > rects

7.67.1 Function Documentation

7.67.1.1 findRect()

```
sf::IntRect findRect ( )
```

7.67.2 Variable Documentation

7.67.2.1 rects

```
std::vector<sf::IntRect> rects  [static]
```

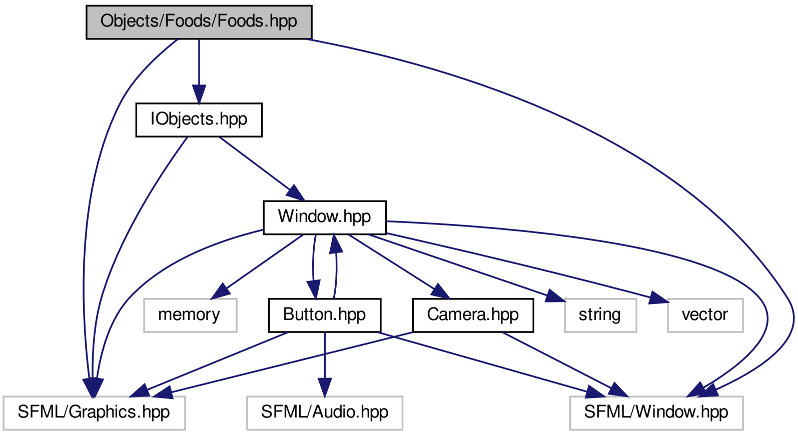
Initial value:

```
= {
    sf::IntRect(800, 1560, 75, 95),
    sf::IntRect(720, 1555, 75, 95),










}
```

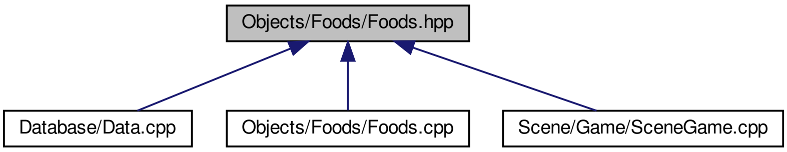
7.68 Objects/Foods/Foods.hpp File Reference

```
#include "IObjects.hpp"
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
```

Include dependency graph for Foods.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

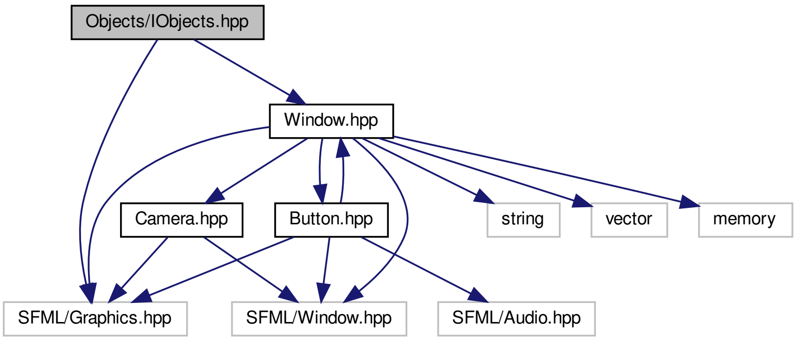
- class Foods

    Foods class.

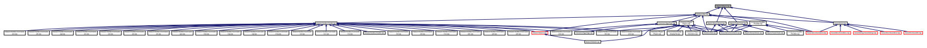
7.69 Objects/IObjects.hpp File Reference

```
#include "Window.hpp"
#include <SFML/Graphics.hpp>
```

Include dependency graph for IObjects.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class IObjects

    IObjects class.

Enumerations

- enum class TypeObject { STONES , FOOD }

Variables

- constexpr char PATH_SPRITE [ ] = "./gui/Resources/spritesheet.png"

7.69.1 Enumeration Type Documentation

7.69.1.1 TypeObject

```
enum TypeObject  [strong]
```

Enumerator

STONES	
FOOD	

7.69.2 Variable Documentation

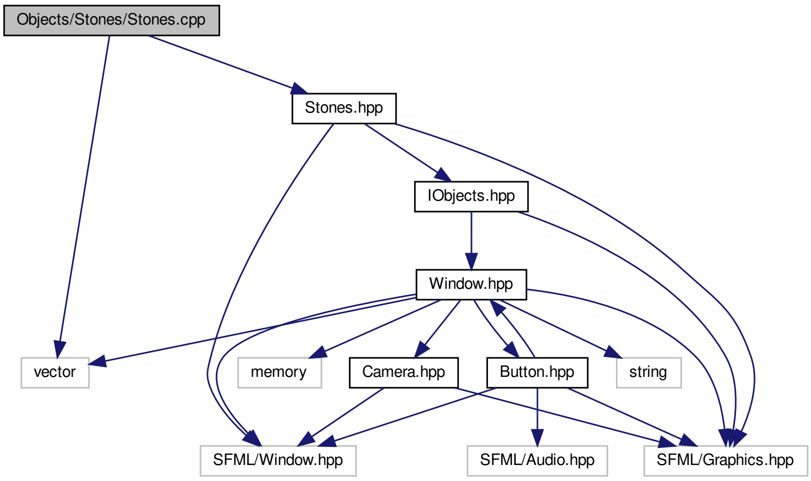
7.69.2.1 PATH_SPRITE

```
constexpr char PATH_SPRITE[ ] = "./gui/Resources/spritesheet.png"  [constexpr]
```

7.70 Objects/Stones/Stones.cpp File Reference

```
#include "Stones.hpp"
#include <vector>
```

Include dependency graph for Stones.cpp:



Functions

- static sf::IntRect findRect (TypeStone type)

Variables

- static std::vector< sf::IntRect > rects

7.70.1 Function Documentation

7.70.1.1 findRect()

```
static sf::IntRect findRect (
            TypeStone type )  [static]
```

7.70.2 Variable Documentation

7.70.2.1 rects

```
std::vector<sf::IntRect> rects  [static]
```

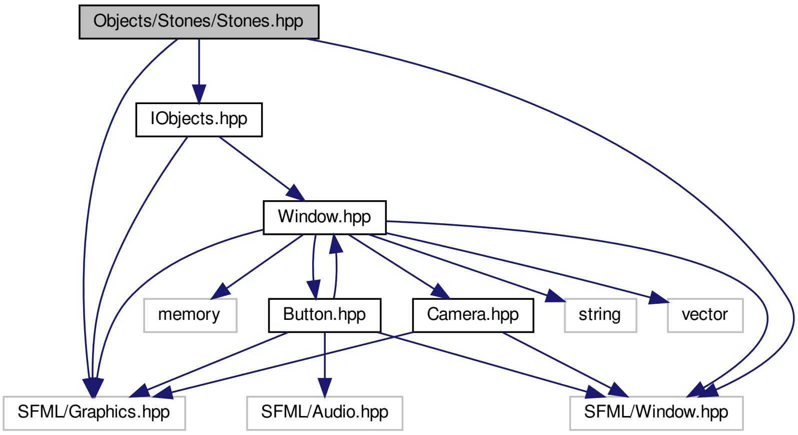
Initial value:

```
= {
    sf::IntRect(794, 1665, 75, 83),
    sf::IntRect(958, 1670, 75, 85),
    sf::IntRect(1200, 1670, 65, 83),
    sf::IntRect(1050, 1675, 77, 70),
    sf::IntRect(1128, 1675, 65, 75),
    sf::IntRect(875, 1660, 80, 89)
}
```

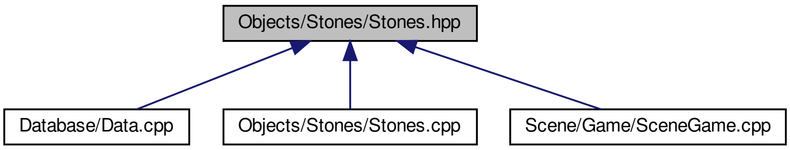
7.71 Objects/Stones/Stones.hpp File Reference

```
#include "IObjects.hpp"
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
```

Include dependency graph for Stones.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Stones

    Stones class.

Enumerations

- enum class TypeStone {
  LINEMATE , DERAUMERE , SIBUR , MENDIANE ,
  PHIRAS , THYSTAME }

7.71.1 Enumeration Type Documentation

7.71.1.1 TypeStone

```
enum TypeStone  [strong]
```

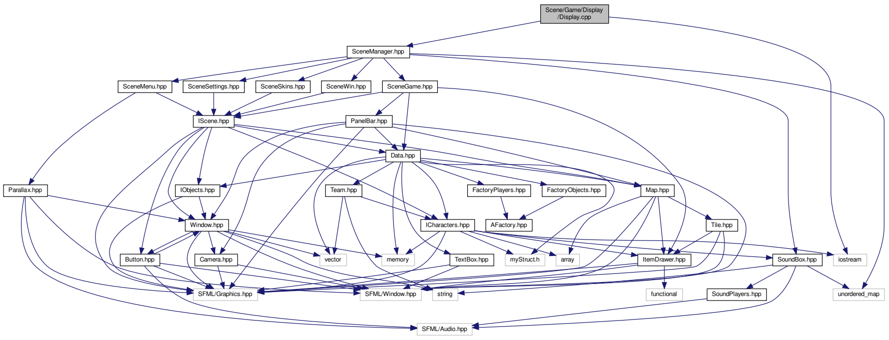
Enumerator

LINEMATE	
DERAUMERE	
SIBUR	
MENDIANE	
PHIRAS	
THYSTAME	

7.72 Scene/Game/Display/Display.cpp File Reference

```
#include "SceneManager.hpp"
#include <iostream>
```

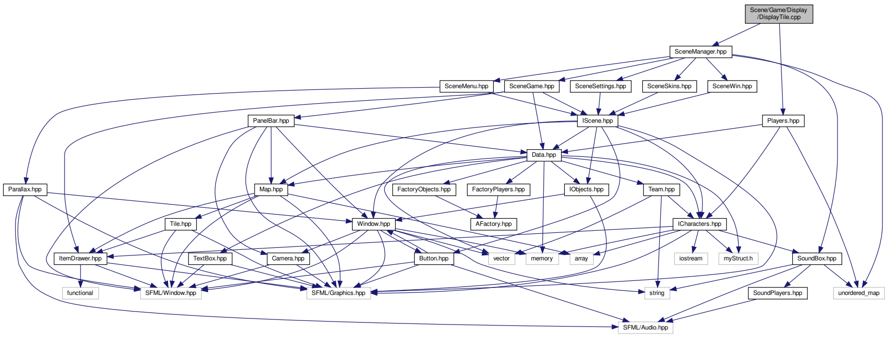
Include dependency graph for Display.cpp:



7.73 Scene/Game/Display/DisplayTile.cpp File Reference

```
#include "SceneManager.hpp"
#include "Players.hpp"
```

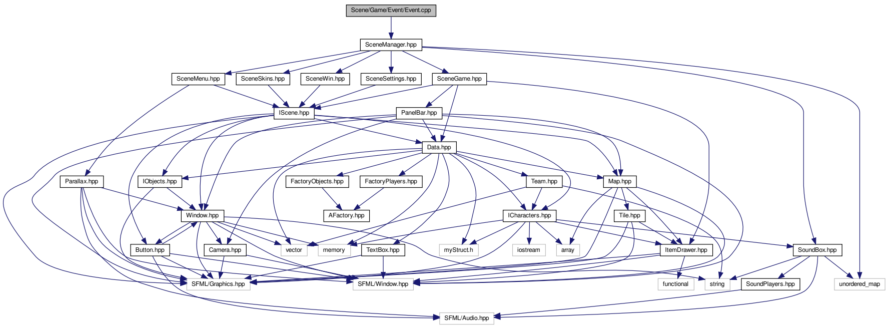
Include dependency graph for DisplayTile.cpp:



7.74 Scene/Game/Event/Event.cpp File Reference

```
#include "SceneManager.hpp"
```

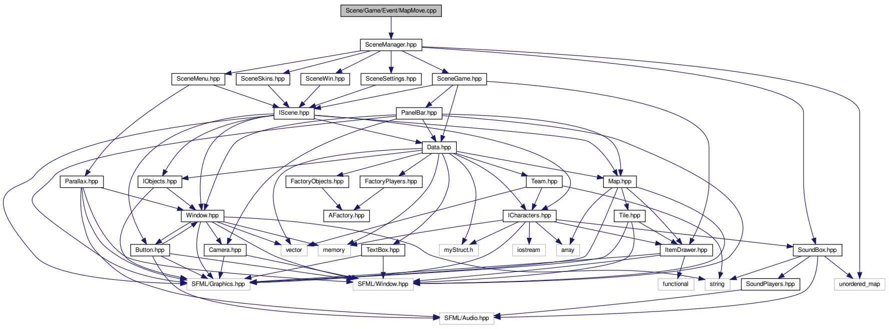
Include dependency graph for Event.cpp:



7.75 Scene/Game/Event/MapMove.cpp File Reference

```
#include "SceneManager.hpp"
```

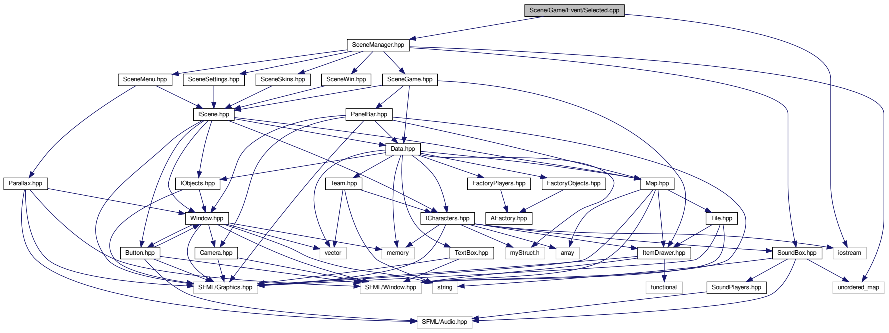
Include dependency graph for MapMove.cpp:



7.76 Scene/Game/Event/Selected.cpp File Reference

```
#include "SceneManager.hpp"
#include <iostream>
```

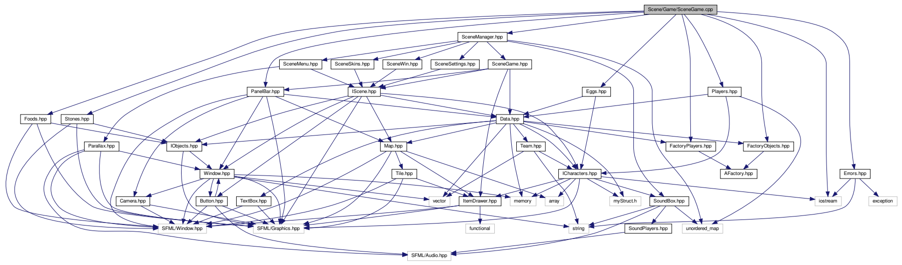
Include dependency graph for Selected.cpp:



7.77 Scene/Game/SceneGame.cpp File Reference

```
#include "SceneManager.hpp"
#include <iostream>
#include "FactoryObjects.hpp"
#include "FactoryPlayers.hpp"
#include "Foods.hpp"
#include "Stones.hpp"
#include "PanelBar.hpp"
#include "Players.hpp"
#include "Eggs.hpp"
#include "Errors.hpp"
```

Include dependency graph for SceneGame.cpp:



Variables

- constexpr char PATH_GOLEM [ ] = "./gui/Resources/golem_spritesheet.png"
- constexpr char PATH_PANEL [ ] = "./gui/Resources/inventory.png"

7.77.1 Variable Documentation

7.77.1.1 PATH_GOLEM

```
constexpr char PATH_GOLEM[ ] = "./gui/Resources/golem_spritesheet.png"  [constexpr]
```

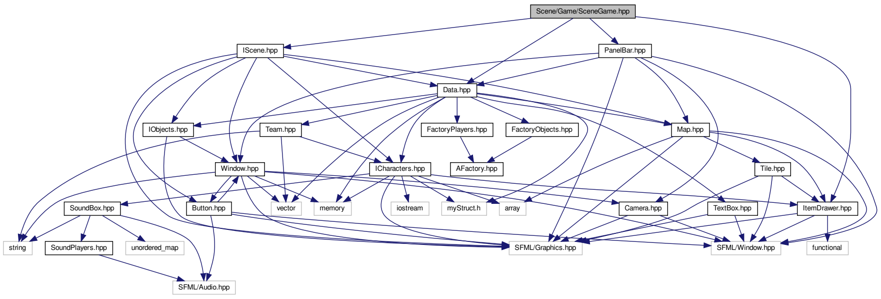
7.77.1.2 PATH_PANEL

```
constexpr char PATH_PANEL[ ] = "./gui/Resources/inventory.png"  [constexpr]
```

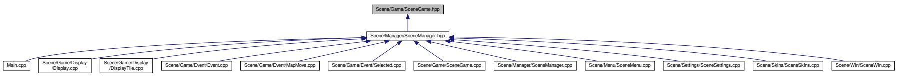
7.78 Scene/Game/SceneGame.hpp File Reference

```
#include "IScene.hpp"
#include "PanelBar.hpp"
#include "Data.hpp"
#include "ItemDrawer.hpp"
```

Include dependency graph for SceneGame.hpp:



This graph shows which files directly or indirectly include this file:


Scene/Game/SceneGame.hpp
Scene/Manager/SceneManager.hpp
Main.cpp
Scene/Game/Display /Display.cpp
Scene/Game/Display /DisplayTile.cpp
Scene/Game/Event/Event.cpp
Scene/Game/Event/MapMove.cpp
Scene/Game/Event/Selected.cpp
Scene/Game/SceneGame.cpp
Scene/Manager/SceneManager.cpp
Scene/Menu/SceneMenu.cpp
Scene/Settings/SceneSettings.cpp
Scene/Skins/SceneSkins.cpp
Scene/Win/SceneWin.cpp

Data Structures

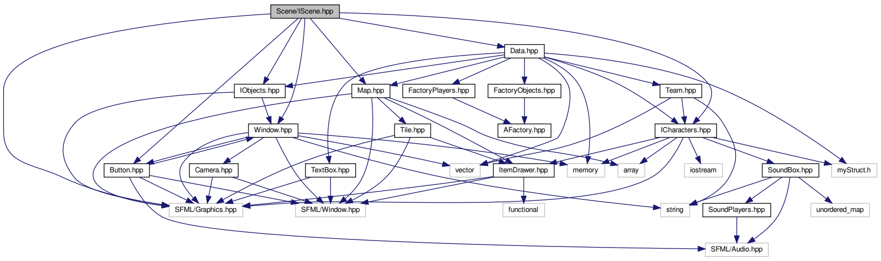
- class SceneGame

  Game scene class.

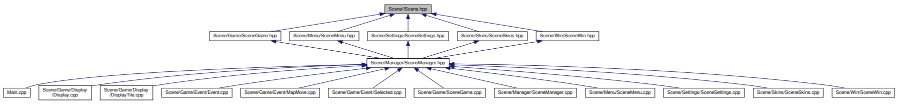
7.79 Scene/IScene.hpp File Reference

```
#include <SFML/Graphics.hpp>
#include "Window.hpp"
#include "Button.hpp"
#include "Map.hpp"
#include "IObjects.hpp"
#include "ICharacters.hpp"
#include "Data.hpp"
```

Include dependency graph for IScene.hpp:



This graph shows which files directly or indirectly include this file:


Scene/IScene.hpp
Scene/Game/SceneGame.hpp
Scene/Menu/SceneMenu.hpp
Scene/Settings/SceneSettings.hpp
Scene/Skins/SceneSkins.hpp
Scene/Win/SceneWin.hpp
Scene/Manager/SceneManager.hpp
Main.cpp
Scene/Game/Display /Display.cpp
Scene/Game/Display /DisplayTile.cpp
Scene/Game/Event/Event.cpp
Scene/Game/Event/MapMove.cpp
Scene/Game/Event/Selected.cpp
Scene/Game/SceneGame.cpp
Scene/Manager/SceneManager.cpp
Scene/Menu/SceneMenu.cpp
Scene/Settings/SceneSettings.cpp
Scene/Skins/SceneSkins.cpp
Scene/Win/SceneWin.cpp

Data Structures

- class IScene

Namespaces

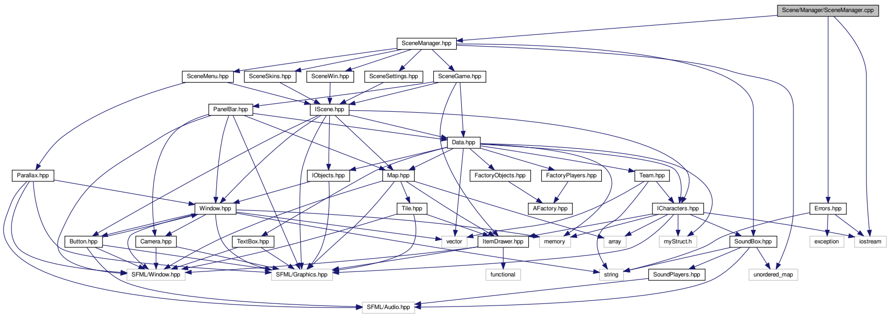
- gui

7.80 Scene/Manager/SceneManager.cpp File Reference

```
#include "SceneManager.hpp"
#include <iostream>
```

```
#include "Errors.hpp"
```

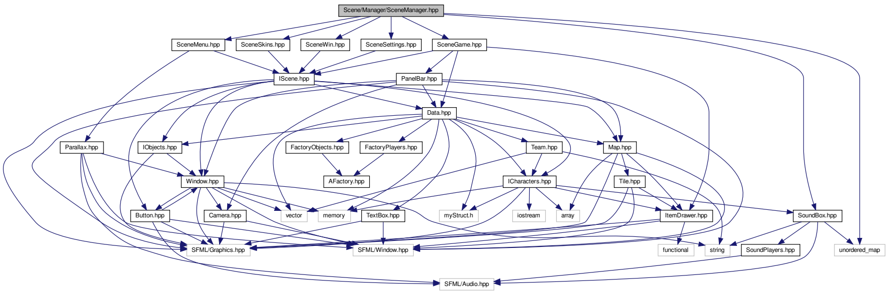
Include dependency graph for SceneManager.cpp:



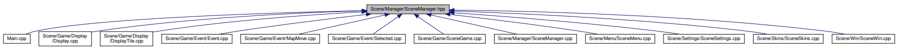
7.81 Scene/Manager/SceneManager.hpp File Reference

```
#include "SceneMenu.hpp"
#include "SceneSettings.hpp"
#include "SceneGame.hpp"
#include "SceneSkins.hpp"
#include "SceneWin.hpp"
#include <unordered_map>
#include "SoundBox.hpp"
```

Include dependency graph for SceneManager.hpp:



This graph shows which files directly or indirectly include this file:


Scene/Manager/SceneManager.hpp
Main.cpp
Scene/Game/Display /Display.cpp
Scene/Game/Display /DisplayTile.cpp
Scene/Game/Event/Event.cpp
Scene/Game/Event/MapMove.cpp
Scene/Game/Event/Selected.cpp
Scene/Game/SceneGame.cpp
Scene/Manager/SceneManager.cpp
Scene/Menu/SceneMenu.cpp
Scene/Settings/SceneSettings.cpp
Scene/Skins/SceneSkins.cpp
Scene/Win/SceneWin.cpp

Data Structures

- class gui::SceneManager

  SceneManager class.

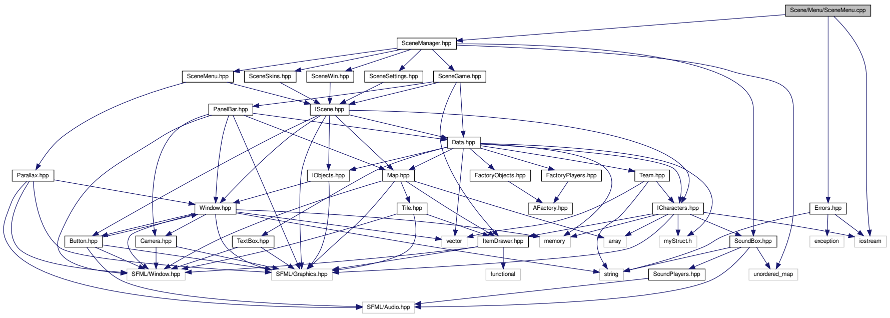
Namespaces

- gui

7.82 Scene/Menu/SceneMenu.cpp File Reference

```
#include "SceneManager.hpp"
#include <iostream>
#include "Errors.hpp"
```

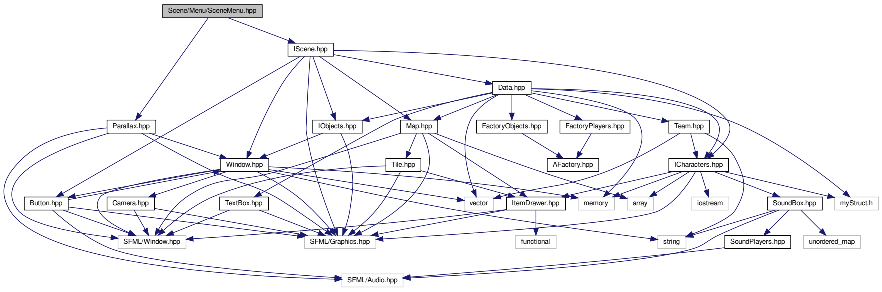
Include dependency graph for SceneMenu.cpp:



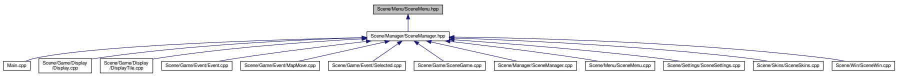
7.83 Scene/Menu/SceneMenu.hpp File Reference

```
#include "IScene.hpp"
#include "Parallax.hpp"
```

Include dependency graph for SceneMenu.hpp:



This graph shows which files directly or indirectly include this file:


Scene/Menu/SceneMenu.hpp
Scene/Manager/SceneManager.hpp
Main.cpp
Scene/Game/Display/Display.cpp
Scene/Game/Display/DisplayTile.cpp
Scene/Game/Event/Event.cpp
Scene/Game/Event/MapMove.cpp
Scene/Game/Event/Selected.cpp
Scene/Game/SceneGame.cpp
Scene/Manager/SceneManager.cpp
Scene/Menu/SceneMenu.cpp
Scene/Settings/SceneSettings.cpp
Scene/Skins/SceneSkins.cpp
Scene/Win/SceneWin.cpp

Data Structures

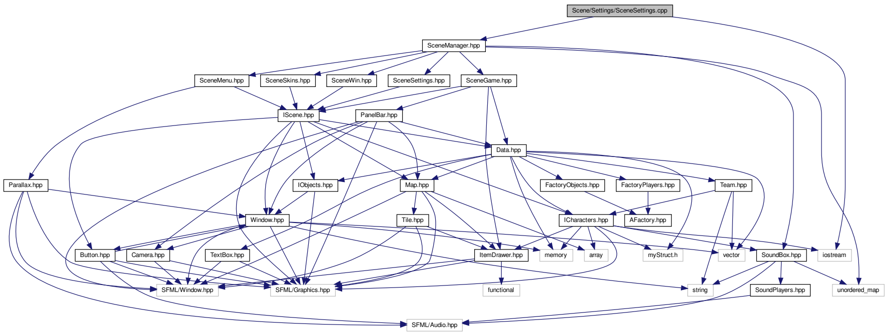
- class SceneMenu

  SceneMenu class.

7.84 Scene/Settings/SceneSettings.cpp File Reference

```
#include <iostream>
#include "SceneManager.hpp"
```

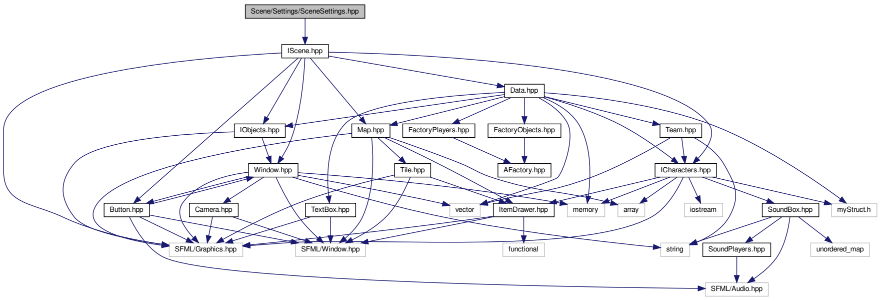
Include dependency graph for SceneSettings.cpp:



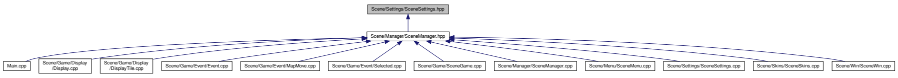
7.85 Scene/Settings/SceneSettings.hpp File Reference

```
#include "IScene.hpp"
```

Include dependency graph for SceneSettings.hpp:



This graph shows which files directly or indirectly include this file:


Scene/Settings/SceneSettings.hpp
Scene/Manager/SceneManager.hpp
Main.cpp
Scene/Game/Display /Display.cpp
Scene/Game/Display /DisplayTile.cpp
Scene/Game/Event/Event.cpp
Scene/Game/Event/MapMove.cpp
Scene/Game/Event/Selected.cpp
Scene/Game/SceneGame.cpp
Scene/Manager/SceneManager.cpp
Scene/Menu/SceneMenu.cpp
Scene/Settings/SceneSettings.cpp
Scene/Skins/SceneSkins.cpp
Scene/Win/SceneWin.cpp

Data Structures

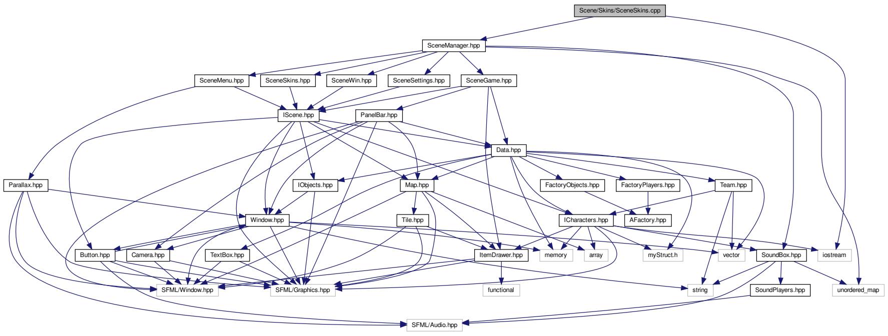
- class SceneSettings

    Settings scene class.

7.86 Scene/Skins/SceneSkins.cpp File Reference

```
#include <iostream>
#include "SceneManager.hpp"
```

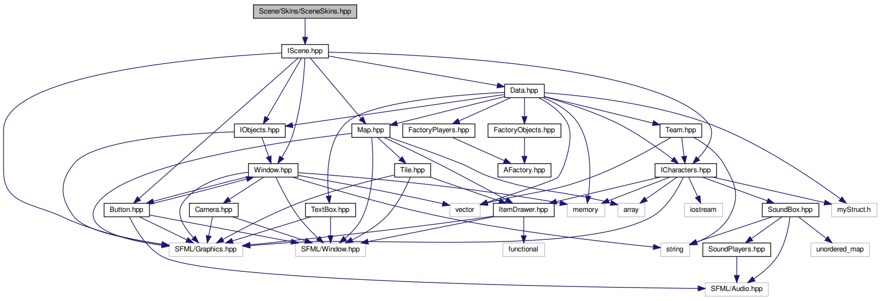
Include dependency graph for SceneSkins.cpp:



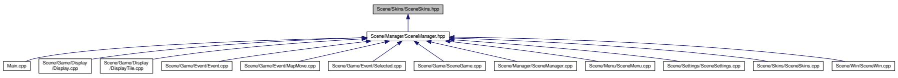
7.87 Scene/Skins/SceneSkins.hpp File Reference

```
#include "IScene.hpp"
```

Include dependency graph for SceneSkins.hpp:



This graph shows which files directly or indirectly include this file:


Scene/Skins/SceneSkins.hpp
Scene/Manager/SceneManager.hpp
Main.cpp
Scene/Game/Display
/Display.cpp
Scene/Game/Display
/DisplayTile.cpp
Scene/Game/Event/Event.cpp
Scene/Game/Event/MapMove.cpp
Scene/Game/Event/Selected.cpp
Scene/Game/SceneGame.cpp
Scene/Manager/SceneManager.cpp
Scene/Menu/SceneMenu.cpp
Scene/Settings/SceneSettings.cpp
Scene/Skins/SceneSkins.cpp
Scene/Win/SceneWin.cpp

Data Structures

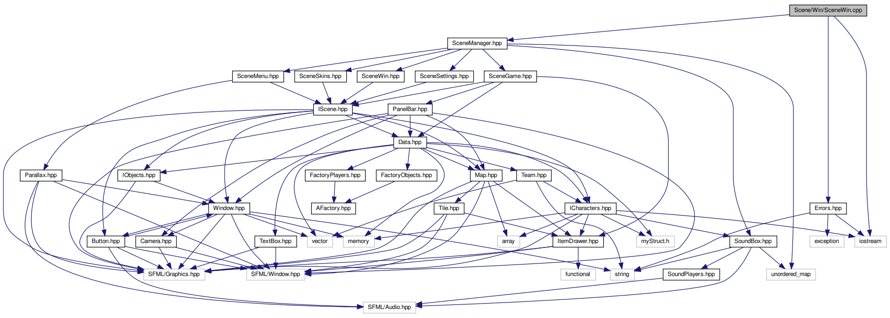
- class SceneSkins

  Skins scene class.

7.88 Scene/Win/SceneWin.cpp File Reference

```
#include <iostream>
#include "SceneManager.hpp"
#include "Errors.hpp"
```

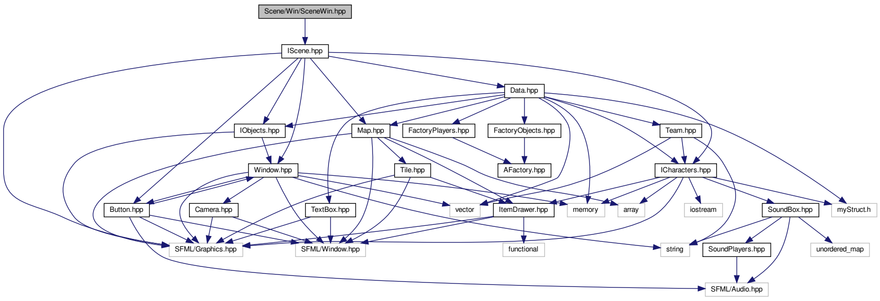
Include dependency graph for SceneWin.cpp:



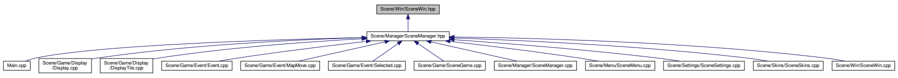
7.89 Scene/Win/SceneWin.hpp File Reference

```
#include "IScene.hpp"
```

Include dependency graph for SceneWin.hpp:



This graph shows which files directly or indirectly include this file:


Scene/Win/SceneWin.hpp
Scene/Manager/SceneManager.hpp
Main.cpp
Scene/Game/Display /Display.cpp
Scene/Game/Display /DisplayTile.cpp
Scene/Game/Event/Event.cpp
Scene/Game/Event/MapMove.cpp
Scene/Game/Event/Selected.cpp
Scene/Game/SceneGame.cpp
Scene/Manager/SceneManager.cpp
Scene/Menu/SceneMenu.cpp
Scene/Settings/SceneSettings.cpp
Scene/Skins/SceneSkins.cpp
Scene/Win/SceneWin.cpp

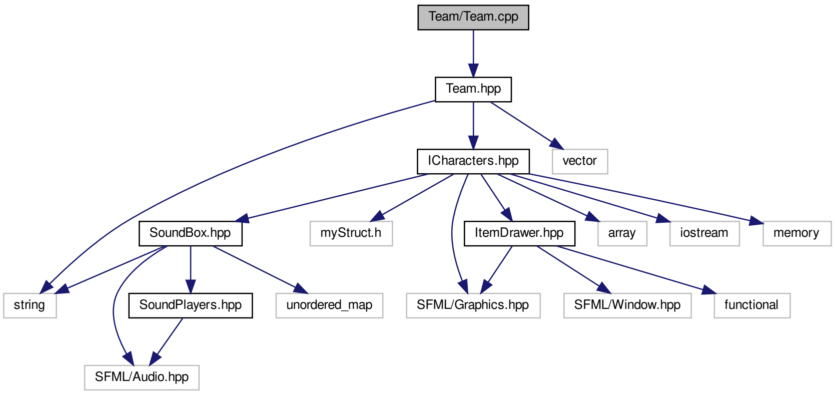
Data Structures

- class SceneWin

7.90 Team/Team.cpp File Reference

```
#include "Team.hpp"
```

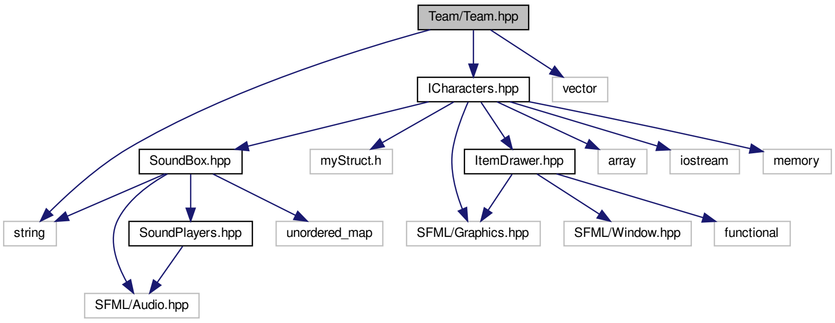
Include dependency graph for Team.cpp:



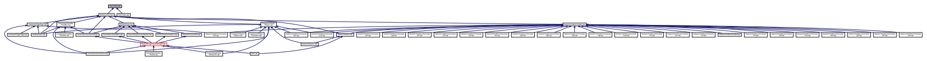
7.91 Team/Team.hpp File Reference

```
#include <string>
#include <vector>
#include "ICharacters.hpp"
```

Include dependency graph for Team.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class gui::Team

    Team class.

Namespaces

- gui

Index