My Project

Generated by Doxygen 1.9.8

Chapter 1

Zappy Project - GUI Client Documentation

1.1 Overview

The Zappy project is a network game where multiple teams compete on a tile map filled with resources. The goal is for a team to have six players reach the highest level. This documentation details the architecture and functionality of the GUI client, which is written in C++ and uses the Raylib library for graphics and GTests for unit testing.

1.2 Table of Contents

1.3 Introduction

The GUI client for the Zappy project connects to the server, displays the game world, and handles user interactions. It is designed to manage the game loop, rendering, and interactions seamlessly. The client is developed in C++ and leverages the Raylib library for graphics rendering and GTests for unit testing.

1.4 Architecture

1.4.1 IException

Handles exceptions in the client, ensuring robust error management.

1.4.2 GUI Components

1.4.2.1 IObject

An interface for all game objects.

1.4.2.2 AObject

Abstract base class for all objects, including food, stones, and eggs. Manages position and quantity.

1.4.2.3 Position

Manages the position of objects, including setters and getters for X and Y coordinates.

1.4.2.4 Game

Manages the main game loop, display, and events, coordinating the entire gameplay experience.

1.4.2.5 Display

Handles the window, tiles, Trantorians, objects, teams, eggs, stones, food, and menus.

1.4.2.6 Events

Manages mouse events, including interactions with tiles and Trantorians.

1.4.2.7 Tile

Represents tiles and includes functions to modify their content and position.

1.4.2.8 Trantorian

Represents Trantorians, managing their creation, 3D models, textures, position, orientation, and inventory.

1.4.2.9 Team

Represents teams, managing their Trantorians, number, and name.

1.4.2.10 Action

Represents actions performed by Trantorians, managing possible actions.

1.4.2.11 World

Represents the world, managing tiles and their content.

1.5 Network

1.5.1 Client

Manages the client connection to the server, handling connection, disconnection, and data transmission.

1.5.2 ASocket

Abstract base class for sockets, defining essential socket operations.

1.5.3 TCPSocket

Implements ASocket for TCP connections, managing connection, data transmission, and socket state.

1.5.4 ProtocolHandler

Receives data and sends it to the CommandFactory for processing, ensuring proper handling of incoming data.

1.5.5 CommandFactory

Creates and manages commands, directing them to the appropriate callbacks.

1.6 Launching the GUI Client

To launch the GUI client, use the following command:

```
./zappy_gui -p port [-h hostname]
```

- `-p port`: Specifies the port number.
- `-h hostname`: (Optional) Specifies the server hostname. Defaults to `127.0.0.1`.

1.7 Authors

- Made by: `@BxptisteM` & `@Klayni`

For more detailed information, refer to the project PDF documentation provided.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 5

Class Documentation

5.1 GUI::Actions Class Reference

Public Member Functions

- **Actions** ()=default

  Construct a new Actions object.
- ~**Actions** ()=default

  Destroy the Actions object.
- bool moveForward ()

  Trantorian move up one tile.
- bool turnRight ()

  Trantorian turn 90 ° right.
- bool turnLeft ()

  Trantorian turn 90 ° left.
- std::vector< std::vector< int > > lookAround ()

  Trantorian look tiles around him.
- std::map< std::string, int > openInventory ()

  Trantorian open inventory.
- bool broadcastText (std::string text)

  Trantorian turn 90 ° left.
- int connectNbr ()

  Number of team unused slots.
- bool forkPlayer ()

  Trantorian fork a player.
- void **playerDead** ()

  Death of a player.
- bool takeObj ()

  Trantorian take an object.
- bool setObj ()

  Trantorian set an object.
- bool startIncantation ()

  Trantorian start incantation.

5.1.1 Member Function Documentation

5.1.1.1 broadcastText()

```
bool GUI::Actions::broadcastText (
                std::string text )
```

Trantorian turn 90 ° left.

Parameters

text	

Returns

true if action succeed

5.1.1.2 connectNbr()

```
int GUI::Actions::connectNbr ( )
```

Number of team unused slots.

Returns

int

5.1.1.3 forkPlayer()

```
bool GUI::Actions::forkPlayer ( )
```

Trantorian fork a player.

Returns

true if action succeed

5.1.1.4 lookAround()

```
std::vector< std::vector< int > > GUI::Actions::lookAround ( )
```

Trantorian look tiles around him.

Returns

vector containing tiles positions

5.1.1.5 moveForward()

```
bool GUI::Actions::moveForward ( )
```

Trantorian move up one tile.

Returns

true if action succeed

5.1.1.6 openInventory()

```
std::map< std::string, int > GUI::Actions::openInventory ( )
```

Trantorian open inventory.

Returns

map containing objects in inventory

5.1.1.7 setObj()

```
bool GUI::Actions::setObj ( )
```

Trantorian set an object.

Returns

true if action succeed

5.1.1.8 startIncantation()

```
bool GUI::Actions::startIncantation ( )
```

Trantorian start incantation.

Returns

true if action succeed

5.1.1.9 takeObj()

```
bool GUI::Actions::takeObj ( )
```

Trantorian take an object.

Returns

true if action succeed

5.1.1.10 turnLeft()

```
bool GUI::Actions::turnLeft ( )
```

Trantorian turn 90 ° left.

Returns

true if action succeed

5.1.1.11 turnRight()

```
bool GUI::Actions::turnRight ( )
```

Trantorian turn 90 ° right.

Returns

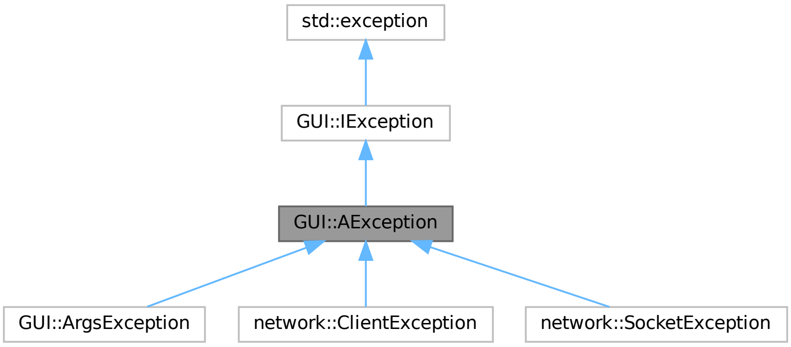
true if action succeed

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

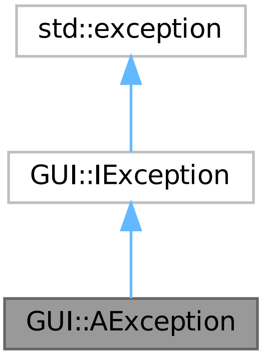
- src/trantorians/Actions.hpp
- src/trantorians/Actions.cpp

5.2 GUI::AException Class Reference

Inheritance diagram for GUI::AException:



Collaboration diagram for GUI::AException:



Public Member Functions

- AException (std::string message, std::string type) noexcept

  Construct a new AException object.
- virtual ∼**AException** () noexcept=default

  Destroy the AException object.
- const char ∗ what () const noexcept final

  Get the message object.
- std::string getType () const noexcept final

  Get the Type object.

Public Member Functions inherited from GUI::IException

- **IException** () noexcept=default

  Construct a new IException object.
- virtual ∼**IException** () noexcept=default

  Destroy the IException object.

Protected Attributes

- std::string **_message**
- std::string **_type**

5.2.1 Constructor & Destructor Documentation

5.2.1.1 AException()

```
AException::AException (
              std::string message,
              std::string type )  [noexcept]
```

Construct a new AException object.

Parameters

message	(error message)

5.2.2 Member Function Documentation

5.2.2.1 getType()

`std::string AException::getType ( ) const [final], [virtual], [noexcept]`

Get the Type object.

Returns

std::string (error type)

Implements GUI::IException.

5.2.2.2 what()

`const char * AException::what ( ) const [final], [virtual], [noexcept]`

Get the message object.

Returns

const char∗ (error message)

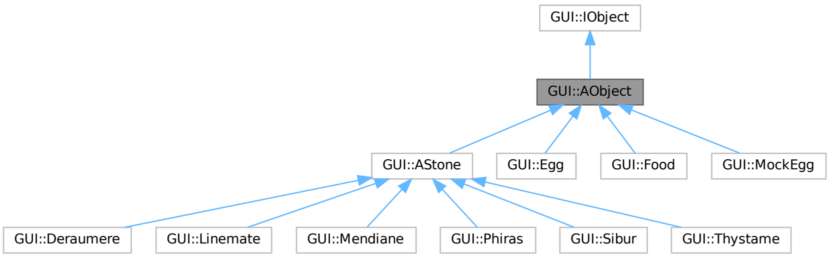
Implements GUI::IException.

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

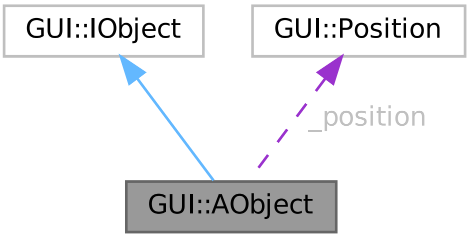
- src/exception/abstract/AException.hpp
- src/exception/abstract/AException.cpp

5.3 GUI::AObject Class Reference

Inheritance diagram for GUI::AObject:



Collaboration diagram for GUI::AObject:



Public Member Functions

- AObject (unsigned int quantity, Position tile)

  Construct a new AObject object.
- virtual ∼**AObject** () noexcept=default

  Destroy the AObject object.
- unsigned int getQuantity () const noexcept final

  Get the Quantity object.
- const Position & getPosition () const noexcept final

  Get the Position of the tile where the object is.
- Position & getPosition () noexcept

  Get the Position of the tile where the object is.
- unsigned int getType () noexcept

  Get the Type of the object.
- virtual void setQuantity (unsigned int quantity) noexcept final

  Set the Quantity of the object.
- virtual std::string getName () const noexcept

  get the name of the object

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

  Construct a new IObject object.
- virtual ∼**IObject** () noexcept=default

  Destroy the IObject object.

Protected Attributes

- Position **_position**
- unsigned int **_quantity**

5.3.1 Constructor & Destructor Documentation

5.3.1.1 AObject()

```
AObject::AObject (
              unsigned int quantity,
              Position tile )
```

Construct a new AObject object.

Parameters

quantity	
tile	

5.3.2 Member Function Documentation

5.3.2.1 getName()

```
std::string AObject::getName ( ) const  [virtual], [noexcept]
```

get the name of the object

Implements GUI::IObject.

Reimplemented in GUI::Food, GUI::Deraumere, GUI::Linemate, GUI::Mendiane, GUI::Phiras, GUI::Sibur, and GUI::Thystame.

5.3.2.2 getPosition() [1/2]

```
const Position & AObject::getPosition ( ) const  [final], [virtual], [noexcept]
```

Get the Position of the tile where the object is.

Returns

const Position& tile where the object is (read-only)

Implements GUI::IObject.

5.3.2.3 getPosition() [2/2]

```
Position & AObject::getPosition ( )  [virtual], [noexcept]
```

Get the Position of the tile where the object is.

Returns

Position& tile where the object is (modifiable)

Implements GUI::IObject.
Reimplemented in GUI::MockEgg, and GUI::Egg.

5.3.2.4 getQuantity()

```
unsigned int AObject::getQuantity ( ) const  [final], [virtual], [noexcept]
```

Get the Quantity object.

Returns

unsigned int (Quantity of the object)

Implements GUI::IObject.

5.3.2.5 getType()

```
unsigned int AObject::getType ( )  [virtual], [noexcept]
```

Get the Type of the object.

Returns

unsigned int (type of the object (follow the protocol))

Implements GUI::IObject.
Reimplemented in GUI::Food, GUI::Deraumere, GUI::Linemate, GUI::Mendiane, GUI::Phiras, GUI::Sibur, and GUI::Thystame.

5.3.2.6 setQuantity()

```
void AObject::setQuantity (
             unsigned int quantity )  [final], [virtual], [noexcept]
```

Set the Quantity of the object.

Parameters

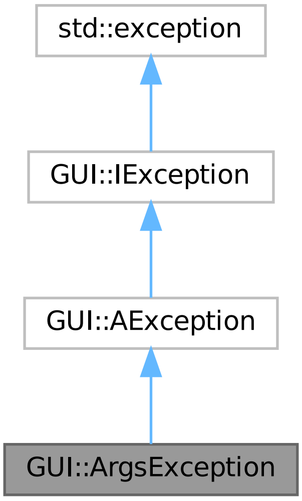
quantity	(Quantity of the object)

Implements GUI::IObject.
The documentation for this class was generated from the following files:

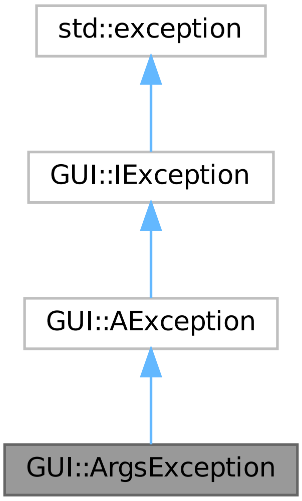
- src/objects/abstracts/AObject.hpp
- src/objects/abstracts/AObject.cpp

5.4 GUI::ArgsException Class Reference

Inheritance diagram for GUI::ArgsException:



Collaboration diagram for GUI::ArgsException:



Public Member Functions

- ArgsException (std::string message)

    Construct a new ArgsException object.

Public Member Functions inherited from GUI::AException

- AException (std::string message, std::string type) noexcept

  Construct a new AException object.
- virtual ∼**AException** () noexcept=default

  Destroy the AException object.
- const char ∗ what () const noexcept final

  Get the message object.
- std::string getType () const noexcept final

  Get the Type object.

Public Member Functions inherited from GUI::IException

- **IException** () noexcept=default

  Construct a new IException object.
- virtual ∼**IException** () noexcept=default

  Destroy the IException object.

Additional Inherited Members

Protected Attributes inherited from GUI::AException

- std::string **_message**
- std::string **_type**

5.4.1 Constructor & Destructor Documentation

5.4.1.1 ArgsException()

```
ArgsException::ArgsException (
                std::string message )
```

Construct a new ArgsException object.

Parameters

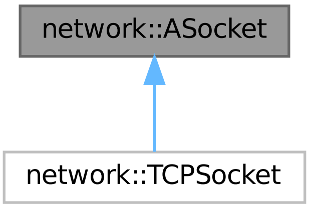
message	(error message)

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

- src/exception/ArgsException.hpp
- src/exception/ArgsException.cpp

5.5 network::ASocket Class Reference

Inheritance diagram for network::ASocket:



Public Member Functions

- **ASocket** ()=default

  Construct a new ASocket object.
- virtual ∼**ASocket** () noexcept=default

  Destroy the ASocket object.
- virtual void connect (const std::string &hostname, unsigned int port)=0

  Connect to the server.
- virtual void close ()=0

  Close the connection to the server.
- virtual void send (const std::string &data)=0

  Send data to the server.
- virtual std::vector< std::string > receive ()=0

  Receive data from the server.
- virtual int getSockfd () const =0

  Get the socket file descriptor.

5.5.1 Member Function Documentation

5.5.1.1 close()

```
virtual void network::ASocket::close ( )  [pure virtual]
```

Close the connection to the server.
Implemented in network::TCPSocket.

5.5.1.2 connect()

```
virtual void network::ASocket::connect (
             const std::string & hostname,
             unsigned int port )  [pure virtual]
```

Connect to the server.

Parameters

hostname	Hostname of the server
port	Port of the server

Implemented in network::TCPSocket.

5.5.1.3 getSockfd()

```
virtual int network::ASocket::getSockfd ( ) const  [pure virtual]
```

Get the socket file descriptor.

Returns

int The socket file descriptor

Implemented in network::TCPSocket.

5.5.1.4 receive()

```
virtual std::vector< std::string > network::ASocket::receive ( )  [pure virtual]
```

Receive data from the server.

Returns

std::vector<std::string> Data received

Implemented in network::TCPSocket.

5.5.1.5 send()

```
virtual void network::ASocket::send (
             const std::string & data )  [pure virtual]
```

Send data to the server.

Parameters

data	Data to send

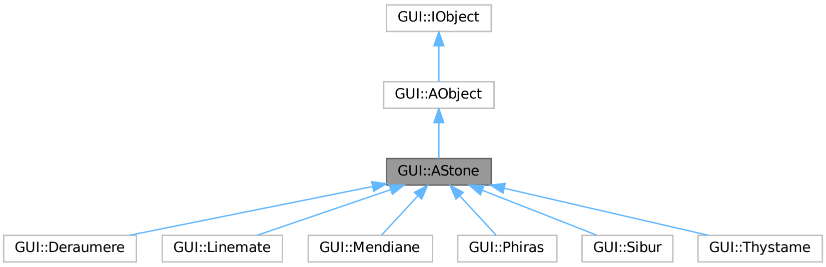
Implemented in network::TCPSocket.

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

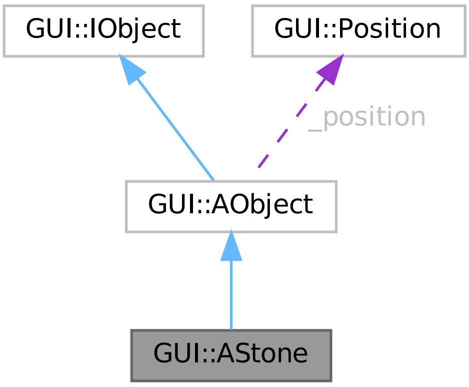
- src/network/socket/ASocket.hpp

5.6 GUI::AStone Class Reference

Inheritance diagram for GUI::AStone:



Collaboration diagram for GUI::AStone:



Public Member Functions

- AStone (unsigned int quantity, Position tile)

  Construct a new AStone object.
- virtual ∼**AStone** () noexcept=default

  Destroy the AStone object.

Public Member Functions inherited from GUI::AObject

- AObject (unsigned int quantity, Position tile)

  Construct a new AObject object.
- virtual ∼**AObject** () noexcept=default

  Destroy the AObject object.
- unsigned int getQuantity () const noexcept final

  Get the Quantity object.
- const Position & getPosition () const noexcept final

  Get the Position of the tile where the object is.
- Position & getPosition () noexcept

  Get the Position of the tile where the object is.
- unsigned int getType () noexcept

  Get the Type of the object.
- virtual void setQuantity (unsigned int quantity) noexcept final

  Set the Quantity of the object.
- virtual std::string getName () const noexcept

  get the name of the object

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

  Construct a new IObject object.
- virtual ∼**IObject** () noexcept=default

  Destroy the IObject object.

Additional Inherited Members

Protected Attributes inherited from GUI::AObject

- Position **_position**
- unsigned int **_quantity**

5.6.1 Constructor & Destructor Documentation

5.6.1.1 AStone()

```
AStone::AStone (
            unsigned int quantity,
            Position tile )
```

Construct a new AStone object.

Parameters

quantity	
tile	

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

- src/objects/abstracts/AStone.hpp
- src/objects/abstracts/AStone.cpp

5.7 network::Client Class Reference

Public Member Functions

- Client (const std::string &hostname, unsigned int port)

  Construct a new Client object.
- void **handleConnection** ()

  Handle the connection to the server.
- void **handleDisconnection** ()

  Handle the disconnection from the server.
- std::vector< std::string > readData ()

  Send data to the server.

Public Attributes

- std::unique_ptr< ASocket > **_socket**

  Socket of the client.

Protected Attributes

- unsigned int **_port**

  Port of the server.
- std::string **_hostname**

  Hostname of the server.

5.7.1 Constructor & Destructor Documentation

5.7.1.1 Client()

```
Client::Client (
            const std::string & hostname,
            unsigned int port )
```

Construct a new Client object.

Construct a new Client object.

Parameters

hostname	Hostname of the server
port	Port of the server

5.7.2 Member Function Documentation

5.7.2.1 readData()

```
std::vector< std::string > Client::readData ( )
```

Send data to the server.

Parameters

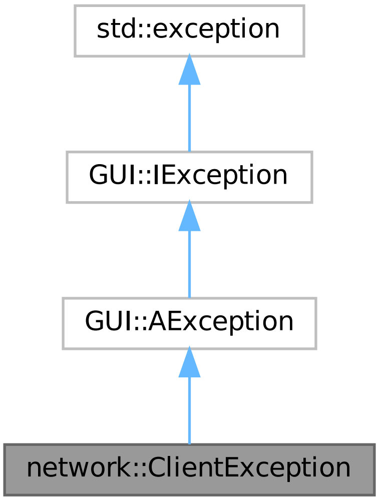
data	Data to send

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

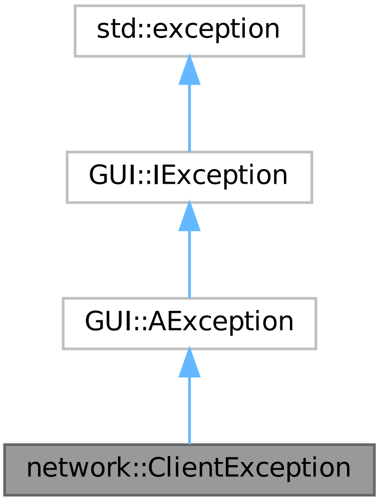
- src/network/Client.hpp
- src/network/Client.cpp

5.8 network::ClientException Class Reference

Inheritance diagram for network::ClientException:



Collaboration diagram for network::ClientException:



Public Member Functions

- ClientException (std::string message)

  Construct a new ClientException object.

Public Member Functions inherited from GUI::AException

- AException (std::string message, std::string type) noexcept

  Construct a new AException object.
- virtual ∼**AException** () noexcept=default

  Destroy the AException object.
- const char ∗ what () const noexcept final

  Get the message object.
- std::string getType () const noexcept final

  Get the Type object.

Public Member Functions inherited from GUI::IException

- **IException** () noexcept=default

  Construct a new IException object.
- virtual ∼**IException** () noexcept=default

  Destroy the IException object.

Additional Inherited Members

Protected Attributes inherited from GUI::AException

- std::string **_message**
- std::string **_type**

5.8.1 Constructor & Destructor Documentation

5.8.1.1 ClientException()

```
ClientException::ClientException (
                std::string message )
```

Construct a new ClientException object.

Parameters

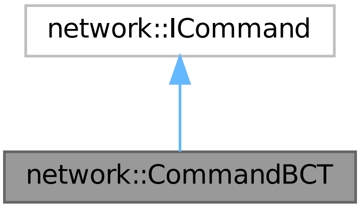
message	(error message)

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

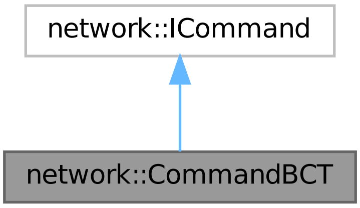
- src/exception/ClientException.hpp
- src/exception/ClientException.cpp

5.9 network::CommandBCT Class Reference

Inheritance diagram for network::CommandBCT:



Collaboration diagram for network::CommandBCT:



Public Member Functions

- **CommandBCT** ()=default

    Construct a new Command BCT object.
- ∼**CommandBCT** ()=default

Destroy the Command BCT object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.9.1 Member Function Documentation

5.9.1.1 clone()

```
std::unique_ptr< ICommand > CommandBCT::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.9.1.2 execute()

```
void CommandBCT::execute (
            std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

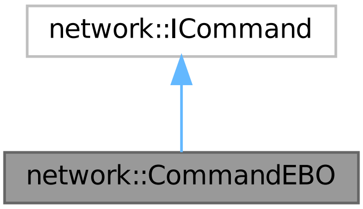
Implements network::ICommand.

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

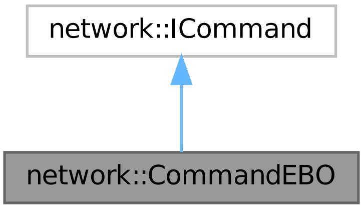
- src/network/commands/bct/CommandBCT.hpp
- src/network/commands/bct/CommandBCT.cpp

5.10 network::CommandEBO Class Reference

Inheritance diagram for network::CommandEBO:



Collaboration diagram for network::CommandEBO:



Public Member Functions

- **CommandEBO** ()=default

  Construct a new Command EBO object.
- ∼**CommandEBO** ()=default

  Destroy the Command EBO object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.10.1 Member Function Documentation

5.10.1.1 clone()

```
std::unique_ptr< ICommand > CommandEBO::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.10.1.2 execute()

```
void CommandEBO::execute (
            std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

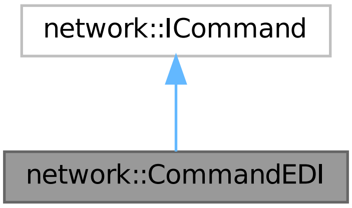
Implements network::ICommand.

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

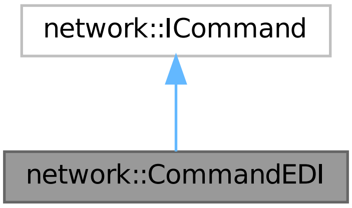
- src/network/commands/ebo/CommandEBO.hpp
- src/network/commands/ebo/CommandEBO.cpp

5.11 network::CommandEDI Class Reference

Inheritance diagram for network::CommandEDI:



Collaboration diagram for network::CommandEDI:



Public Member Functions

- **CommandEDI** ()=default

  Construct a new Command EDI object.
- ∼**CommandEDI** ()=default

  Destroy the Command EDI object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.11.1 Member Function Documentation

5.11.1.1 clone()

```
std::unique_ptr< ICommand > CommandEDI::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.11.1.2 execute()

```
void CommandEDI::execute (
            std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

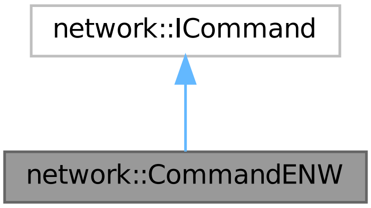
Implements network::ICommand.

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

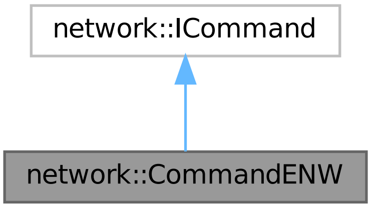
- src/network/commands/edi/CommandEDI.hpp
- src/network/commands/edi/CommandEDI.cpp

5.12 network::CommandENW Class Reference

Inheritance diagram for network::CommandENW:



Collaboration diagram for network::CommandENW:



Public Member Functions

- **CommandENW** ()=default

    Construct a new Command ENW object.
- ∼**CommandENW** ()=default

    Destroy the Command ENW object.

- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.12.1 Member Function Documentation

5.12.1.1 clone()

```
std::unique_ptr< ICommand > CommandENW::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.12.1.2 execute()

```
void CommandENW::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

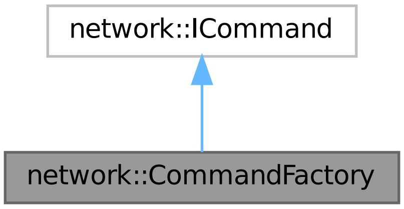
Implements network::ICommand.

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

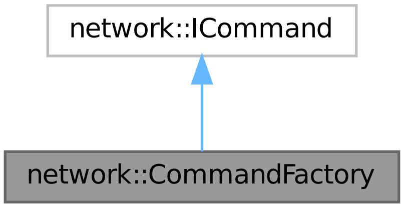
- src/network/commands/enw/CommandENW.hpp
- src/network/commands/enw/CommandENW.cpp

5.13 network::CommandFactory Class Reference

Inheritance diagram for network::CommandFactory:



Collaboration diagram for network::CommandFactory:



Public Member Functions

- **CommandFactory** ()

  Construct a new Command Factory object.
- ∼**CommandFactory** ()=default

  Destroy the Command Factory object.
- std::unique_ptr< ICommand > createCommand (const std::string &commandName)

  Execute the command.
- void setCallback (const std::string &commandName, ICommand::Callback callback)

  Execute the command.
- void execute (std::istringstream &iss) override

  Execute the command.
- std::unique_ptr< ICommand > clone () const override

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.13.1 Member Function Documentation

5.13.1.1 clone()

```
std::unique_ptr< ICommand > CommandFactory::clone ( ) const  [override], [virtual]
```

Clone the command.

Implements network::ICommand.

5.13.1.2 createCommand()

```
std::unique_ptr< ICommand > CommandFactory::createCommand (
             const std::string & commandName )
```

Execute the command.

Parameters

std::string	commandName, name of the command to create

Returns

std::unique_ptr<ICommand> the created command

5.13.1.3 execute()

```
void CommandFactory::execute (
             std::istringstream & iss )  [override], [virtual]
```

Execute the command.

Parameters

std::istringstream	&iss, the command to execute

Implements network::ICommand.

5.13.1.4 setCallback()

```
void CommandFactory::setCallback (
             const std::string & commandName,
             ICommand::Callback callback )
```

Execute the command.

Parameters

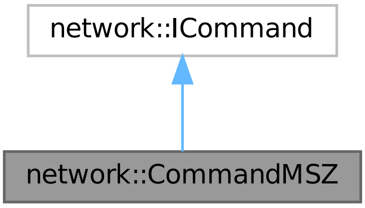
std::string	&commandName, name of the command to create
Callback	callback, the callback to attach to the command

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

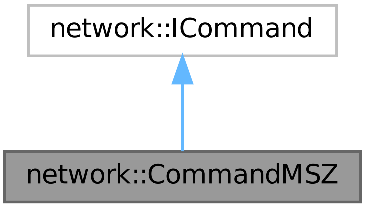
- src/network/commands/factory/CommandFactory.hpp
- src/network/commands/factory/CommandFactory.cpp

5.14 network::CommandMSZ Class Reference

Inheritance diagram for network::CommandMSZ:



Collaboration diagram for network::CommandMSZ:



Public Member Functions

- **CommandMSZ** ()=default

  Construct a new Command PPO object.
- ∼**CommandMSZ** ()=default

  Destroy the Command PPO object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.14.1 Member Function Documentation

5.14.1.1 clone()

```
std::unique_ptr< ICommand > CommandMSZ::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.14.1.2 execute()

```
void CommandMSZ::execute (
                std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

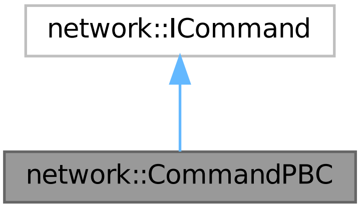
Implements network::ICommand.

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

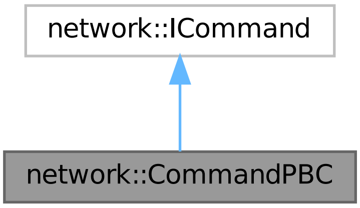
- src/network/commands/msz/CommandMSZ.hpp
- src/network/commands/msz/CommandMSZ.cpp

5.15 network::CommandPBC Class Reference

Inheritance diagram for network::CommandPBC:



Collaboration diagram for network::CommandPBC:



Public Member Functions

- **CommandPBC** ()=default

    Construct a new Command PBC object.
- ∼**CommandPBC** ()=default

    Destroy the Command PBC object.
- void execute (std::istringstream &iss) final

    Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

    Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

    Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

    Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

    Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.15.1 Member Function Documentation

5.15.1.1 clone()

```
std::unique_ptr< ICommand > CommandPBC::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.15.1.2 execute()

```
void CommandPBC::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

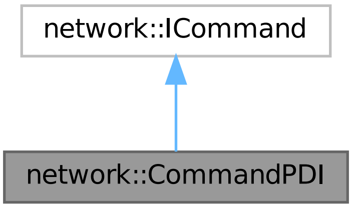
Implements network::ICommand.

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

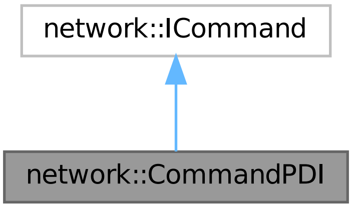
- src/network/commands/pbc/CommandPBC.hpp
- src/network/commands/pbc/CommandPBC.cpp

5.16 network::CommandPDI Class Reference

Inheritance diagram for network::CommandPDI:



Collaboration diagram for network::CommandPDI:



Public Member Functions

- **CommandPDI** ()=default

  Construct a new Command PDI object.
- ∼**CommandPDI** ()=default

  Destroy the Command PDI object.

- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ~**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.16.1 Member Function Documentation

5.16.1.1 clone()

```
std::unique_ptr< ICommand > CommandPDI::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.16.1.2 execute()

```
void CommandPDI::execute (
            std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

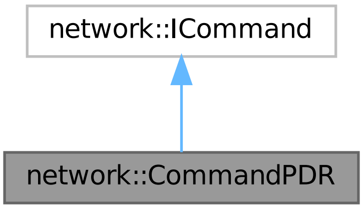
Implements network::ICommand.

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

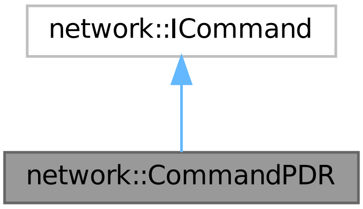
- src/network/commands/pdi/CommandPDI.hpp
- src/network/commands/pdi/CommandPDI.cpp

5.17 network::CommandPDR Class Reference

Inheritance diagram for network::CommandPDR:



Collaboration diagram for network::CommandPDR:



Public Member Functions

- **CommandPDR** ()=default

  Construct a new Command PDR object.
- ∼**CommandPDR** ()=default

  Destroy the Command PDR object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.17.1 Member Function Documentation

5.17.1.1 clone()

```
std::unique_ptr< ICommand > CommandPDR::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.17.1.2 execute()

```
void CommandPDR::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

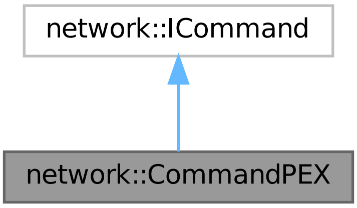
iss	std::istringstream

Implements network::ICommand.
The documentation for this class was generated from the following files:

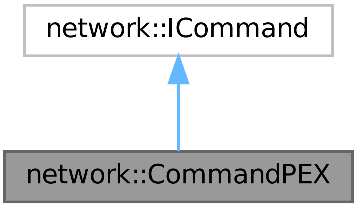
- src/network/commands/pdr/CommandPDR.hpp
- src/network/commands/pdr/CommandPDR.cpp

5.18 network::CommandPEX Class Reference

Inheritance diagram for network::CommandPEX:



Collaboration diagram for network::CommandPEX:



Public Member Functions

- **CommandPEX** ()=default

  Construct a new Command PLV object.
- ∼**CommandPEX** ()=default

  Destroy the Command PLV object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.18.1 Member Function Documentation

5.18.1.1 clone()

```
std::unique_ptr< ICommand > CommandPEX::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.18.1.2 execute()

```
void CommandPEX::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

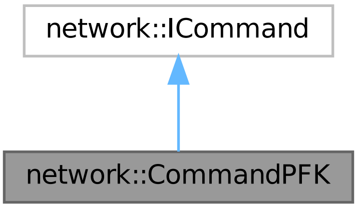
Implements network::ICommand.

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

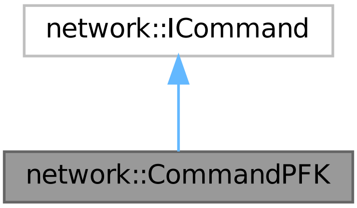
- src/network/commands/pex/CommandPEX.hpp
- src/network/commands/pex/CommandPEX.cpp

5.19 network::CommandPFK Class Reference

Inheritance diagram for network::CommandPFK:



Collaboration diagram for network::CommandPFK:



Public Member Functions

- **CommandPFK** ()=default

  Construct a new Command PFK object.
- ∼**CommandPFK** ()=default

  Destroy the Command PFK object.

- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ~**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.19.1 Member Function Documentation

5.19.1.1 clone()

```
std::unique_ptr< ICommand > CommandPFK::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.19.1.2 execute()

```
void CommandPFK::execute (
            std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

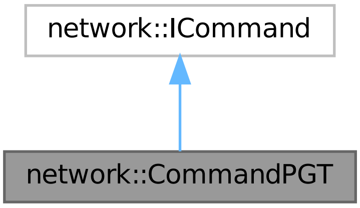
Implements network::ICommand.

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

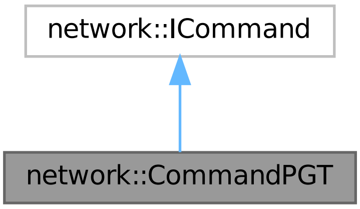
- src/network/commands/pfk/CommandPFK.hpp
- src/network/commands/pfk/CommandPFK.cpp

5.20 network::CommandPGT Class Reference

Inheritance diagram for network::CommandPGT:



Collaboration diagram for network::CommandPGT:



Public Member Functions

- **CommandPGT** ()=default

  Construct a new Command PGT object.
- ∼**CommandPGT** ()=default

  Destroy the Command PGT object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.20.1 Member Function Documentation

5.20.1.1 clone()

```
std::unique_ptr< ICommand > CommandPGT::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.20.1.2 execute()

```
void CommandPGT::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

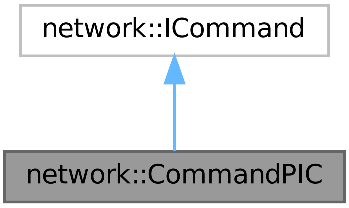
Implements network::ICommand.

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

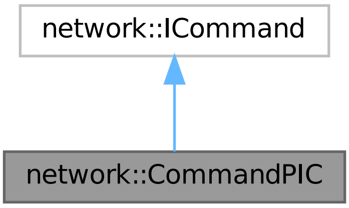
- src/network/commands/pgt/CommandPGT.hpp
- src/network/commands/pgt/CommandPGT.cpp

5.21 network::CommandPIC Class Reference

Inheritance diagram for network::CommandPIC:



Collaboration diagram for network::CommandPIC:



Public Member Functions

- **CommandPIC** ()=default

  Construct a new CommandPIC object.
- ∼**CommandPIC** ()=default

  Destroy the CommandPIC object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.21.1 Member Function Documentation

5.21.1.1 clone()

```
std::unique_ptr< ICommand > CommandPIC::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.21.1.2 execute()

```
void CommandPIC::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

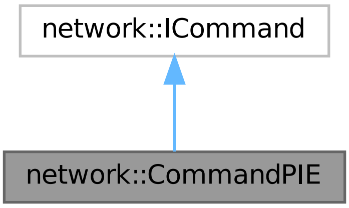
Implements network::ICommand.

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

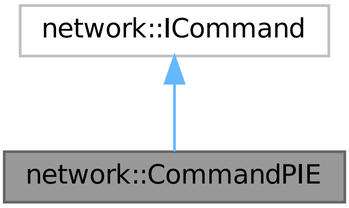
- src/network/commands/pic/CommandPIC.hpp
- src/network/commands/pic/CommandPIC.cpp

5.22 network::CommandPIE Class Reference

Inheritance diagram for network::CommandPIE:



Collaboration diagram for network::CommandPIE:



Public Member Functions

- **CommandPIE** ()=default

  Construct a new CommandPIE object.

- ∼**CommandPIE** ()=default

  Destroy the CommandPIE object.

- void execute (std::istringstream &iss) final

    Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

    Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

    Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

    Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

    Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.22.1 Member Function Documentation

5.22.1.1 clone()

```
std::unique_ptr< ICommand > CommandPIE::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.22.1.2 execute()

```
void CommandPIE::execute (
            std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

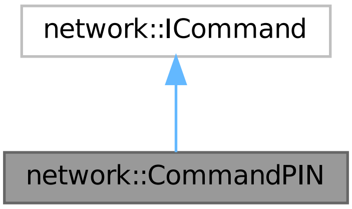
Implements network::ICommand.

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

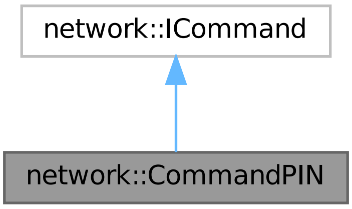
- src/network/commands/pie/CommandPIE.hpp
- src/network/commands/pie/CommandPIE.cpp

5.23 network::CommandPIN Class Reference

Inheritance diagram for network::CommandPIN:



Collaboration diagram for network::CommandPIN:



Public Member Functions

- **CommandPIN** ()=default

  Construct a new Command PIN object.
- ∼**CommandPIN** ()=default

  Destroy the Command PIN object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.23.1 Member Function Documentation

5.23.1.1 clone()

```
std::unique_ptr< ICommand > CommandPIN::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.23.1.2 execute()

```
void CommandPIN::execute (
            std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

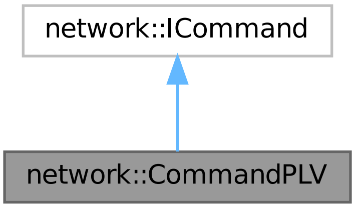
Implements network::ICommand.

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

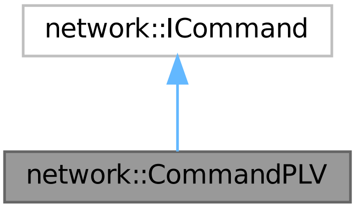
- src/network/commands/pin/CommandPIN.hpp
- src/network/commands/pin/CommandPIN.cpp

5.24 network::CommandPLV Class Reference

Inheritance diagram for network::CommandPLV:



Collaboration diagram for network::CommandPLV:



Public Member Functions

- **CommandPLV** ()=default

  Construct a new Command PLV object.
- ∼**CommandPLV** ()=default

  Destroy the Command PLV object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.24.1 Member Function Documentation

5.24.1.1 clone()

```
std::unique_ptr< ICommand > CommandPLV::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.24.1.2 execute()

```
void CommandPLV::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

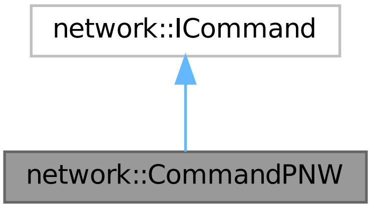
Implements network::ICommand.

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

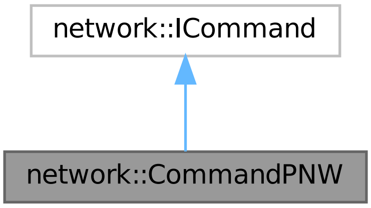
- src/network/commands/plv/CommandPLV.hpp
- src/network/commands/plv/CommandPLV.cpp

5.25 network::CommandPNW Class Reference

Inheritance diagram for network::CommandPNW:



Collaboration diagram for network::CommandPNW:



Public Member Functions

- **CommandPNW** ()=default

  Construct a new Command PNW object.
- ∼**CommandPNW** ()=default

  Destroy the Command PNW object.

- void execute (std::istringstream &iss) final

  Execute the command.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.25.1 Member Function Documentation

5.25.1.1 clone()

```
std::unique_ptr< ICommand > CommandPNW::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.25.1.2 execute()

```
void CommandPNW::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command.

Parameters

std::istringstream	&iss

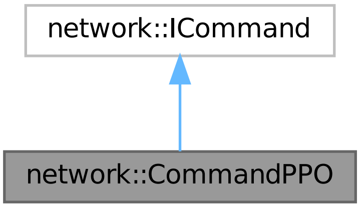
Implements network::ICommand.

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

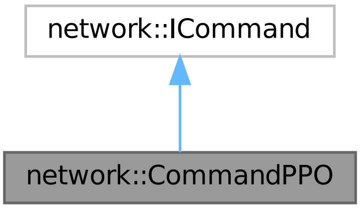
- src/network/commands/pnw/CommandPNW.hpp
- src/network/commands/pnw/CommandPNW.cpp

5.26 network::CommandPPO Class Reference

Inheritance diagram for network::CommandPPO:



Collaboration diagram for network::CommandPPO:



Public Member Functions

- **CommandPPO** ()=default

  Construct a new Command PPO object.
- ∼**CommandPPO** ()=default

  Destroy the Command PPO object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.26.1 Member Function Documentation

5.26.1.1 clone()

```
std::unique_ptr< ICommand > CommandPPO::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.26.1.2 execute()

```
void CommandPPO::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

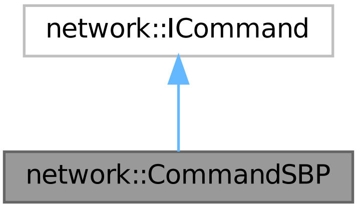
Implements network::ICommand.

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

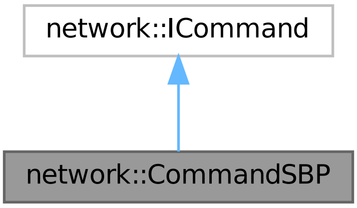
- src/network/commands/ppo/CommandPPO.hpp
- src/network/commands/ppo/CommandPPO.cpp

5.27 network::CommandSBP Class Reference

Inheritance diagram for network::CommandSBP:



Collaboration diagram for network::CommandSBP:



Public Member Functions

- **CommandSBP** ()=default

  Construct a new Command SBP object.
- ∼**CommandSBP** ()=default

  Destroy the Command SBP object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.27.1 Member Function Documentation

5.27.1.1 clone()

```
std::unique_ptr< ICommand > CommandSBP::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.27.1.2 execute()

```
void CommandSBP::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

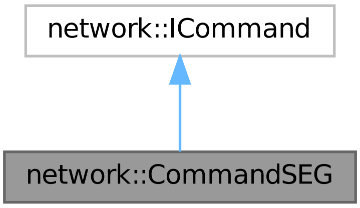
Implements network::ICommand.

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

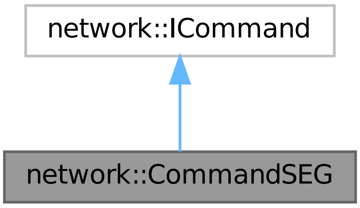
- src/network/commands/sbp/CommandSBP.hpp
- src/network/commands/sbp/CommandSBP.cpp

5.28 network::CommandSEG Class Reference

Inheritance diagram for network::CommandSEG:



Collaboration diagram for network::CommandSEG:



Public Member Functions

- **CommandSEG** ()=default

  Construct a new Command SEG object.
- ∼**CommandSEG** ()=default

  Destroy the Command SEG object.

- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.28.1 Member Function Documentation

5.28.1.1 clone()

```
std::unique_ptr< ICommand > CommandSEG::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.28.1.2 execute()

```
void CommandSEG::execute (
            std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

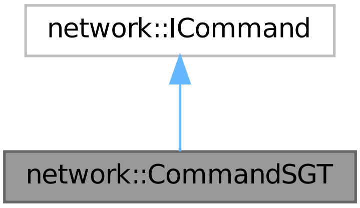
Implements network::ICommand.

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

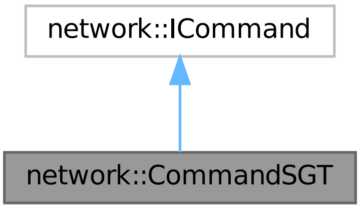
- src/network/commands/seg/CommandSEG.hpp
- src/network/commands/seg/CommandSEG.cpp

5.29 network::CommandSGT Class Reference

Inheritance diagram for network::CommandSGT:



Collaboration diagram for network::CommandSGT:



Public Member Functions

- **CommandSGT** ()=default

  Construct a new CommandSGT object.
- ∼**CommandSGT** ()=default

  Destroy the CommandSGT object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.29.1 Member Function Documentation

5.29.1.1 clone()

```
std::unique_ptr< ICommand > CommandSGT::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.29.1.2 execute()

```
void CommandSGT::execute (
            std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

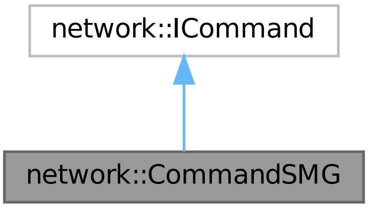
Implements network::ICommand.

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

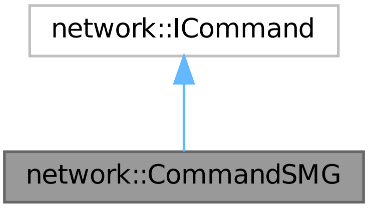
- src/network/commands/sgt/CommandSGT.hpp
- src/network/commands/sgt/CommandSGT.cpp

5.30 network::CommandSMG Class Reference

Inheritance diagram for network::CommandSMG:



Collaboration diagram for network::CommandSMG:



Public Member Functions

- **CommandSMG** ()=default

  Construct a new Command SMG object.
- ∼**CommandSMG** ()=default

  Destroy the Command SMG object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.30.1 Member Function Documentation

5.30.1.1 clone()

```
std::unique_ptr< ICommand > CommandSMG::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.30.1.2 execute()

```
void CommandSMG::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

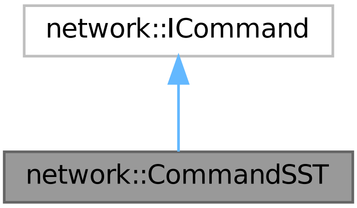
Implements network::ICommand.

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

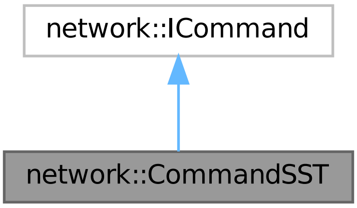
- src/network/commands/smg/CommandSMG.hpp
- src/network/commands/smg/CommandSMG.cpp

5.31 network::CommandSST Class Reference

Inheritance diagram for network::CommandSST:



Collaboration diagram for network::CommandSST:



Public Member Functions

- **CommandSST** ()=default

  Construct a new CommandSST object.
- ∼**CommandSST** ()=default

  Destroy the CommandSST object.

- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ~**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.31.1 Member Function Documentation

5.31.1.1 clone()

```
std::unique_ptr< ICommand > CommandSST::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.31.1.2 execute()

```
void CommandSST::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

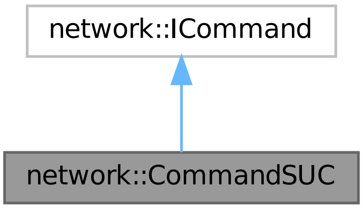
Implements network::ICommand.

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

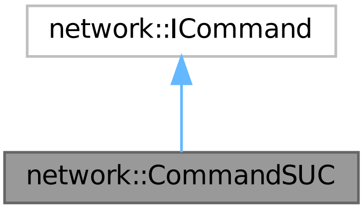
- src/network/commands/sst/CommandSST.hpp
- src/network/commands/sst/CommandSST.cpp

5.32 network::CommandSUC Class Reference

Inheritance diagram for network::CommandSUC:



Collaboration diagram for network::CommandSUC:



Public Member Functions

- **CommandSUC** ()=default

  Construct a new Command SUC object.
- ∼**CommandSUC** ()=default

  Destroy the Command SUC object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.32.1 Member Function Documentation

5.32.1.1 clone()

```
std::unique_ptr< ICommand > CommandSUC::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.32.1.2 execute()

```
void CommandSUC::execute (
             std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

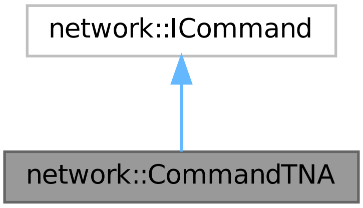
Implements network::ICommand.

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

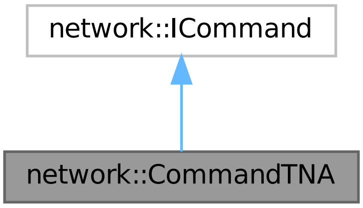
- src/network/commands/suc/CommandSUC.hpp
- src/network/commands/suc/CommandSUC.cpp

5.33 network::CommandTNA Class Reference

Inheritance diagram for network::CommandTNA:



Collaboration diagram for network::CommandTNA:



Public Member Functions

- **CommandTNA** ()=default

  Construct a new Command TNA object.
- ∼**CommandTNA** ()=default

  Destroy the Command TNA object.
- void execute (std::istringstream &iss) final

  Execute the command by callback.
- std::unique_ptr< ICommand > clone () const final

  Clone the command.

Public Member Functions inherited from network::ICommand

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- void **setCallback** (Callback callback)

Additional Inherited Members

Public Types inherited from network::ICommand

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Protected Attributes inherited from network::ICommand

- Callback **_callback**

5.33.1 Member Function Documentation

5.33.1.1 clone()

```
std::unique_ptr< ICommand > CommandTNA::clone ( ) const  [final], [virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implements network::ICommand.

5.33.1.2 execute()

```
void CommandTNA::execute (
                std::istringstream & iss )  [final], [virtual]
```

Execute the command by callback.

Parameters

iss	std::istringstream

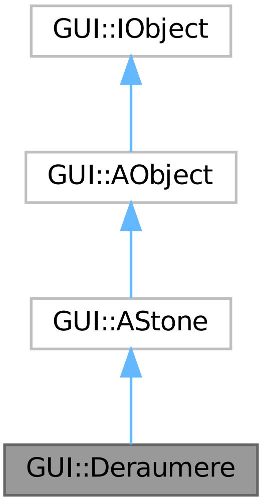
Implements network::ICommand.

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

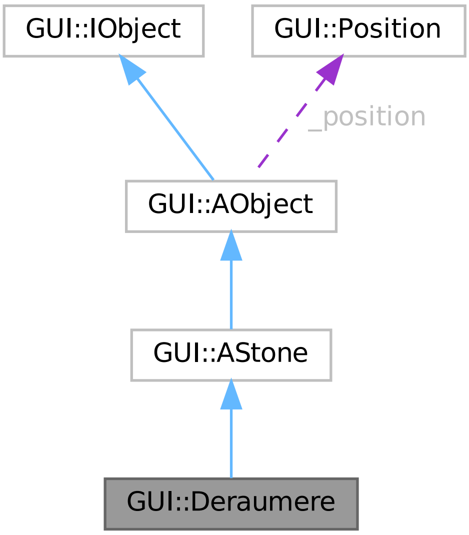
- src/network/commands/tna/CommandTNA.hpp
- src/network/commands/tna/CommandTNA.cpp

5.34 GUI::Deraumere Class Reference

Inheritance diagram for GUI::Deraumere:



Collaboration diagram for GUI::Deraumere:



Public Member Functions

- Deraumere (Position tile=Position(0, 0))

  Construct a new Deraumere object.
- virtual ∼**Deraumere** () noexcept=default

  Destroy the Deraumere object.
- unsigned int getType () noexcept final

  Get the Type of the object.
- std::string getName () const noexcept final

  Get the name of the object.

Public Member Functions inherited from GUI::AStone

- AStone (unsigned int quantity, Position tile)

  Construct a new AStone object.
- virtual ∼**AStone** () noexcept=default

  Destroy the AStone object.

Public Member Functions inherited from GUI::AObject

- AObject (unsigned int quantity, Position tile)

  Construct a new AObject object.
- virtual ∼**AObject** () noexcept=default

  Destroy the AObject object.
- unsigned int getQuantity () const noexcept final

  Get the Quantity object.
- const Position & getPosition () const noexcept final

  Get the Position of the tile where the object is.

- Position & getPosition () noexcept

  Get the Position of the tile where the object is.
- virtual void setQuantity (unsigned int quantity) noexcept final

  Set the Quantity of the object.

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

  Construct a new IObject object.
- virtual ∼**IObject** () noexcept=default

  Destroy the IObject object.

Additional Inherited Members

Protected Attributes inherited from GUI::AObject

- Position **_position**
- unsigned int **_quantity**

5.34.1 Constructor & Destructor Documentation

5.34.1.1 Deraumere()

```
Deraumere::Deraumere (
             Position tile = Position(0, 0) )
```

Construct a new Deraumere object.

Parameters

tile	Tile where the object is

5.34.2 Member Function Documentation

5.34.2.1 getName()

```
std::string Deraumere::getName ( ) const  [final], [virtual], [noexcept]
```

Get the name of the object.

Reimplemented from GUI::AObject.

5.34.2.2 getType()

```
unsigned int Deraumere::getType ( )  [final], [virtual], [noexcept]
```

Get the Type of the object.

Returns

unsigned int (type of the object (follow the protocol))

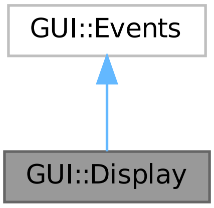
Reimplemented from GUI::AObject.

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

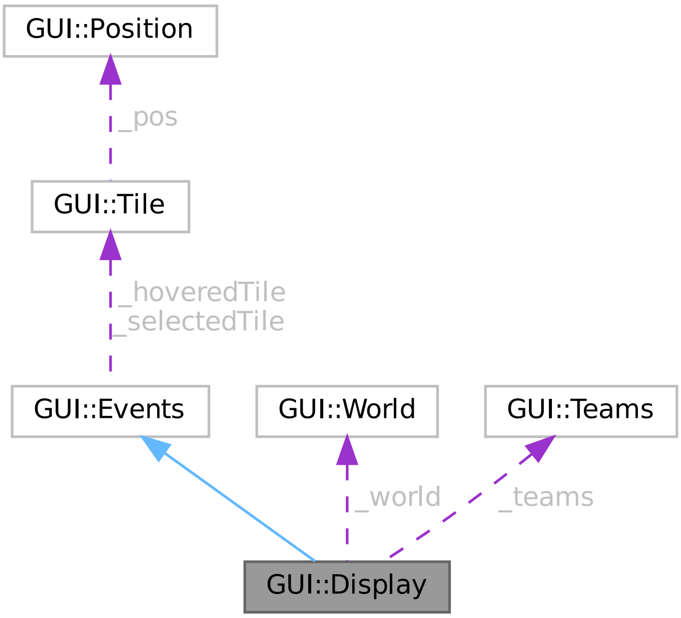
- src/objects/stones/Deraumere.hpp
- src/objects/stones/Deraumere.cpp

5.35 GUI::Display Class Reference

Inheritance diagram for GUI::Display:



Collaboration diagram for GUI::Display:



Public Member Functions

- Display (World &world, Teams &teams)

  Construct a new Display object.
- ∼**Display** ()=default

  Destroy the Display object.
- void **displayElements** ()

  calls all the draw functions
- void DrawTiles (std::vector< std::vector< Tile > > tiles)

Draw the tiles.

- void **DrawClouds** ()

  Draw the clouds.
- void DrawTrantorians (std::list< Teams > teams)

  Draw the trantorians /!\ TO FIX /!.
- void **DrawTileInfo** ()

  Draw the tile info (list of items, position)
- void **DrawTrantorianInfo** ()

  Draw the trantorian info (team, time to live, level, inventory, position, id)
- void DrawObjects (std::list< IObject ∗ > objects)

  Draw the objects (food, linemate, deraumere, sibur, mendiane, phiras, thystame) if there is many obj we just have bigger squares.
- void **DrawEgg** ()

  Draw the egg.
- void **DrawScoreBoard** (Teams &teams)

  Draw the ScoreBoard.
- void **cleanupModels** ()

  Cleanup the models /!\ ADD TRANTORIANS /!.
- void **initClouds** ()

  Initialize the clouds.
- bool windowShouldClose ()

  Check if the window should close.
- void **updateCamera** ()

  Update the camera.
- void **closeWindow** ()

  Close the window.
- std::vector< Model > getClouds () const

  Get the Camera object.
- void **DrawSSTBox** ()

  Draw the SST Box.
- std::string **getNewTimeUnit** ()

  Draw the TextBox.
- void **DisplayHelpMenu** ()

  Display the help menu.
- void **DisplayGameInformations** ()

  Display the game informations on the top right corner.
- void **setTimeUnit** (unsigned int timeUnit)

  setTheTimeUnit
- void **setNewTimeUnit** (std::string newTimeUnit)

  Set the new time unit.
- void addLog (const std::string &log)

  add the log to the history
- void **DrawLogs** ()

  Draw the logs.

Public Member Functions inherited from GUI::Events

- **Events** ()

  Construct a new Events object.
- ∼**Events** ()=default

  Destroy the Events object.
- void detectHoveredTile (Camera _camera, World &_world)

  Detect the hovered tile.
- void detectHoveredTrantorian (Camera _camera, Teams &_teams)

  Detect the selected tile.

Protected Attributes

- Camera **_camera**
- std::vector< Model > **_clouds**
- std::vector< Vector3 > **_cloudPositions**
- World & **_world**
- Teams & **_teams**
- unsigned int **_timeUnit**
- char **_inputText** [256] = ""
- bool **_textBoxActive** = false
- int **_framesCounter** = 0
- int **_ignoreInputFrames** = 0
- std::string **_newTimeUnit**
- bool **_gameInfo** = false
- bool **_drawLogs** = false
- std::vector< std::string > **_logs**

Protected Attributes inherited from GUI::Events

- Tile ∗ **_selectedTile**
- Tile ∗ **_hoveredTile**
- std::unique_ptr< Trantorian > **_selectedTrantorian**
- std::unique_ptr< Trantorian > **_hoveredTrantorian**

5.35.1 Constructor & Destructor Documentation

5.35.1.1 Display()

```
Display::Display (
             World & world,
             Teams & teams )
```

Construct a new Display object.

Parameters

world	World object reference
teams	teams object reference

5.35.2 Member Function Documentation

5.35.2.1 addLog()

```
void Display::addLog (
             const std::string & log )
```

add the log to the history

Parameters

log	the log to add

5.35.2.2 DrawObjects()

```
void Display::DrawObjects (
            std::list< IObject * > objects )
```

Draw the objects (food, linemate, deraumere, sibur, mendiane, phiras, thystame) if there is many obj we just have bigger squares.

Parameters

objects	list of objects

5.35.2.3 DrawTiles()

```
void Display::DrawTiles (
            std::vector< std::vector< Tile > > tiles )
```

Draw the tiles.

Parameters

tiles	2D vector of tiles

5.35.2.4 DrawTrantorians()

```
void Display::DrawTrantorians (
            std::list< Teams > teams )
```

Draw the trantorians /!\ TO FIX /!.

Parameters

teams	list of teams

5.35.2.5 getClouds()

```
std::vector< Model > Display::getClouds ( ) const
```

Get the Camera object.

Returns

Camera object

5.35.2.6 windowShouldClose()

```
bool Display::windowShouldClose ( )
```

Check if the window should close.

Returns

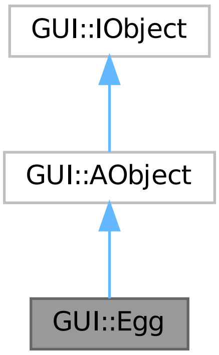
true if the window should close

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

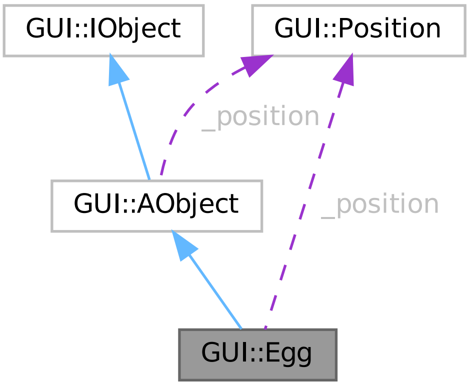
- src/game/display/Display.hpp
- src/game/display/Display.cpp

5.36 GUI::Egg Class Reference

Inheritance diagram for GUI::Egg:



Collaboration diagram for GUI::Egg:



Public Member Functions

- Egg (std::string teamName, std::string ownerID="-1", Position tile=Position(0, 0))

  Construct a new Egg object.
- ∼**Egg** ()

  Destroy the Egg object.
- std::string getTeam () const

  Get the team name.
- std::string getOwnerID () const

  Get the owner ID.
- bool getIsEjected () const

Get the isEjected.

- void setIsEjected (bool isEjected)

  Set the isEjected.
- void setPosition (int x, int y)

  Set egg position.
- Position & getPosition () noexcept

  Get egg position.
- Model getModel ()

  Get egg 3d model.

Public Member Functions inherited from GUI::AObject

- AObject (unsigned int quantity, Position tile)

  Construct a new AObject object.
- virtual ∼**AObject** () noexcept=default

  Destroy the AObject object.
- unsigned int getQuantity () const noexcept final

  Get the Quantity object.
- const Position & getPosition () const noexcept final

  Get the Position of the tile where the object is.
- unsigned int getType () noexcept

  Get the Type of the object.
- virtual void setQuantity (unsigned int quantity) noexcept final

  Set the Quantity of the object.
- virtual std::string getName () const noexcept

  get the name of the object

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

  Construct a new IObject object.
- virtual ∼**IObject** () noexcept=default

  Destroy the IObject object.

Protected Attributes

- std::string **_teamName**
- std::string **_ownerID**
- bool **_isEjected**
- Position **_position**
- Model **_eggModel**
- Texture **_eggTexture**

Protected Attributes inherited from GUI::AObject

- Position **_position**
- unsigned int **_quantity**

5.36.1 Constructor & Destructor Documentation

5.36.1.1 Egg()

```
Egg::Egg (
             std::string teamName,
             std::string ownerID = "-1",
             Position tile = Position(0, 0) )
```

Construct a new Egg object.

Parameters

string	teamName
string	ownerID
Position	Tile where the object is

5.36.2 Member Function Documentation

5.36.2.1 getIsEjected()

```
bool Egg::getIsEjected ( ) const
```

Get the isEjected.

Returns

bool (true if the egg is ejected, false if not)

5.36.2.2 getModel()

```
Model Egg::getModel ( )
```

Get egg 3d model.

Returns

Model type from raylib

5.36.2.3 getOwnerID()

```
std::string Egg::getOwnerID ( ) const
```

Get the owner ID.

Returns

std::string (owner of the egg ID)

5.36.2.4 getPosition()

```
Position & Egg::getPosition ( )  [virtual], [noexcept]
```

Get egg position.

Returns

Position

Reimplemented from GUI::AObject.

5.36.2.5 getTeam()

```
std::string Egg::getTeam ( ) const
```

Get the team name.

Returns

std::string (team name)

5.36.2.6 setIsEjected()

```
void Egg::setIsEjected (
             bool isEjected )
```

Set the isEjected.

Parameters

bool	isEjected

Returns

void (nothing to return)

5.36.2.7 setPosition()

```
void Egg::setPosition (
             int x,
             int y )
```

Set egg position.

Parameters

int	x position and y position

Returns

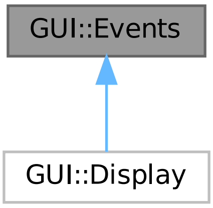
void (nothing to return)

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

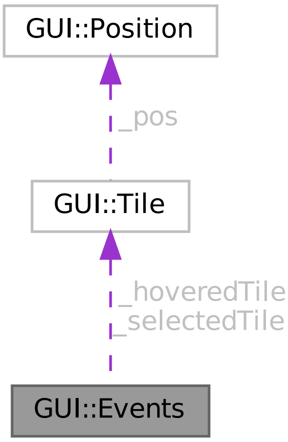
- src/objects/eggs/Egg.hpp
- src/objects/eggs/Egg.cpp

5.37 GUI::Events Class Reference

Inheritance diagram for GUI::Events:



Collaboration diagram for GUI::Events:



Public Member Functions

- **Events** ()

  Construct a new Events object.
- ∼**Events** ()=default

  Destroy the Events object.
- void detectHoveredTile (Camera _camera, World &_world)

  Detect the hovered tile.
- void detectHoveredTrantorian (Camera _camera, Teams &_teams)

  Detect the selected tile.

Protected Attributes

- Tile ∗ **_selectedTile**
- Tile ∗ **_hoveredTile**
- std::unique_ptr< Trantorian > **_selectedTrantorian**
- std::unique_ptr< Trantorian > **_hoveredTrantorian**

5.37.1 Member Function Documentation

5.37.1.1 detectHoveredTile()

```
void Events::detectHoveredTile (
             Camera _camera,
             World & _world )
```

Detect the hovered tile.

Parameters

_camera	camera object
_world	world object

5.37.1.2 detectHoveredTrantorian()

```
void Events::detectHoveredTrantorian (
             Camera _camera,
             Teams & _teams )
```

Detect the selected tile.

Parameters

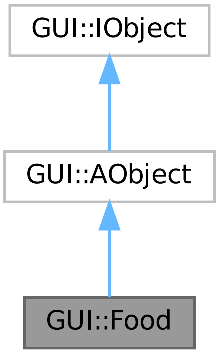
_camera	camera object
_world	world object

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

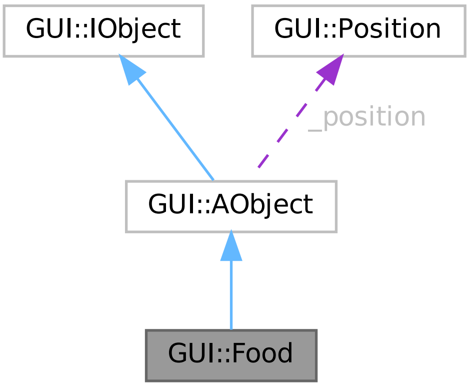
- src/game/events/Events.hpp
- src/game/events/Events.cpp

5.38 GUI::Food Class Reference

Inheritance diagram for GUI::Food:



Collaboration diagram for GUI::Food:



Public Member Functions

- Food (Position tile=Position(0, 0))

  Construct a new Food object.
- virtual ∼**Food** () noexcept=default

  Destroy the Food object.
- unsigned int getType () noexcept final

  Get the Type of the object.
- std::string getName () const noexcept final

  Get the name of the object.

Public Member Functions inherited from GUI::AObject

- AObject (unsigned int quantity, Position tile)

  Construct a new AObject object.
- virtual ∼**AObject** () noexcept=default

  Destroy the AObject object.
- unsigned int getQuantity () const noexcept final

  Get the Quantity object.
- const Position & getPosition () const noexcept final

  Get the Position of the tile where the object is.
- Position & getPosition () noexcept

  Get the Position of the tile where the object is.
- virtual void setQuantity (unsigned int quantity) noexcept final

  Set the Quantity of the object.

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

  Construct a new IObject object.
- virtual ∼**IObject** () noexcept=default

  Destroy the IObject object.

Additional Inherited Members

Protected Attributes inherited from GUI::AObject

- Position **_position**
- unsigned int **_quantity**

5.38.1 Constructor & Destructor Documentation

5.38.1.1 Food()

```
Food::Food (
            Position tile = Position(0, 0) )
```

Construct a new Food object.

Parameters

tile	Tile where the object is

5.38.2 Member Function Documentation

5.38.2.1 getName()

```
std::string Food::getName ( ) const  [final], [virtual], [noexcept]
```

Get the name of the object.
Reimplemented from GUI::AObject.

5.38.2.2 getType()

```
unsigned int Food::getType ( )  [final], [virtual], [noexcept]
```

Get the Type of the object.

Returns

unsigned int (type of the object (follow the protocol))

Reimplemented from GUI::AObject.
The documentation for this class was generated from the following files:

- src/objects/food/Food.hpp
- src/objects/food/Food.cpp

5.39 GUI::Game Class Reference

Public Member Functions

- Game (std::string hostname, unsigned int port)

  Construct a new Game object.
- ∼**Game** ()=default

  Destroy the Game object.
- void **initGame** ()

  Initialize the game elements (Window, Client, World)
- void **runGame** ()

  Run the game loop.
- void **createWorld** (std::vector< std::string > data)

  Create the world object.
- void **initTimeUnit** (std::vector< std::string > data)

  initialize the time unit of the game (ticks)

- void **handleNewTimeUnit** ()

  Check if the GUI has requested a new time unit if true send it to the server.
- void **initializeCallbacks** ()

  Initialize the callbacks of the game.
- void **ensureGameInit** ()

  Ensure the game is initialized by requesting the game info to the server.
- void **ensureGameInformation** ()

  Ensure the info are received from the server.

5.39.1 Constructor & Destructor Documentation

5.39.1.1 Game()

```
Game::Game (
            std::string hostname,
            unsigned int port )
```

Construct a new Game object.

Parameters

hostname	Hostname of the server
port	Port of the server

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

- src/game/Game.hpp
- src/game/Game.cpp

5.40 GUI::HandleArgs Class Reference

Public Member Functions

- **HandleArgs** ()=default

  Construct a new HandleArgs object.
- ∼**HandleArgs** ()=default

  Destroy the HandleArgs object.
- int checkArgs (int nbArgs, char ∗∗args)

  Check if the given args are valid.
- int checkPort (const std::string &port)

  Get the Port object.
- int checkHostname (const std::string &hostname)

  Check if the Hostname given as argument is valid.
- unsigned int getPort () const

  Get the Port object.
- std::string getHostname () const

  Get the Hostname object.
- void printUsage ()

  Print the usage of the program.

Protected Attributes

- unsigned int **_port**
- std::string **_hostname**

5.40.1 Member Function Documentation

5.40.1.1 checkArgs()

```
int HandleArgs::checkArgs (
            int nbArgs,
            char ** args )
```

Check if the given args are valid.

Parameters

nbArgs	(number of arguments)
args	(arguments)

Returns

0 if the args are valid, 84 if not (and throw an exception)

5.40.1.2 checkHostname()

```
int HandleArgs::checkHostname (
            const std::string & hostname )
```

Check if the Hostname given as argument is valid.

Parameters

hostname	(hostname given as argument)

Returns

int, 0 if the hostname is valid, 84 if not

5.40.1.3 checkPort()

```
int HandleArgs::checkPort (
            const std::string & port )
```

Get the Port object.

Parameters

port	(port given as argument)

Returns

int, 0 if the port is valid, 84 if not

5.40.1.4 getHostname()

```
std::string HandleArgs::getHostname ( ) const
```

Get the Hostname object.

Returns

std::string (hostname)

5.40.1.5 getPort()

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

Get the Port object.

Returns

unsigned int (port)

5.40.1.6 printUsage()

```
void HandleArgs::printUsage ( )
```

Print the usage of the program.

Returns

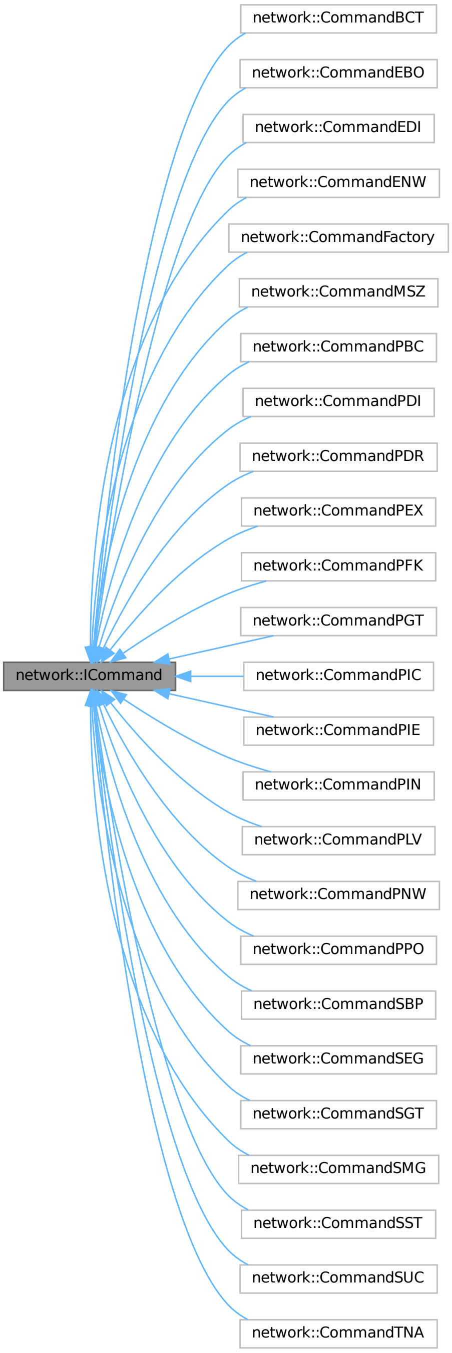
void (nothing to return)

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

- src/handle_args/HandleArgs.hpp
- src/handle_args/HandleArgs.cpp

5.41 network::ICommand Class Reference

Inheritance diagram for network::ICommand:



Public Types

- using Callback = std::function< void(std::istringstream &)>

  Set the callback.

Public Member Functions

- **ICommand** () noexcept=default

  Construct a new ICommand object.
- virtual ∼**ICommand** () noexcept=default

  Destroy the ICommand object.
- virtual void execute (std::istringstream &iss)=0

  Execute the command.
- virtual std::unique_ptr< ICommand > clone () const =0

  Clone the command.
- void **setCallback** (Callback callback)

Protected Attributes

- Callback **_callback**

5.41.1 Member Typedef Documentation

5.41.1.1 Callback

```
using network::ICommand::Callback = std::function<void(std::istringstream&)>
```

Set the callback.

Parameters

callback	std::function<void(std::istringstream&)>

5.41.2 Member Function Documentation

5.41.2.1 clone()

```
virtual std::unique_ptr< ICommand > network::ICommand::clone ( ) const  [pure virtual]
```

Clone the command.

Returns

std::unique_ptr<ICommand>

Implemented in network::CommandBCT, network::CommandEBO, network::CommandEDI, network::CommandENW, network::CommandMSZ, network::CommandPBC, network::CommandPDI, network::CommandPDR, network::CommandPEX, network::CommandPFK, network::CommandPGT, network::CommandPIC, network::CommandPIE, network::CommandPIN, network::CommandPLV, network::CommandPNW, network::CommandPPO, network::CommandSBP, network::CommandSEG, network::CommandSGT, network::CommandSMG, network::CommandSST, network::CommandSUC, network::CommandTNA, and network::CommandFactory.

5.41.2.2 execute()

```
virtual void network::ICommand::execute (
             std::istringstream & iss )  [pure virtual]
```

Execute the command.

Parameters

iss	std::istringstream

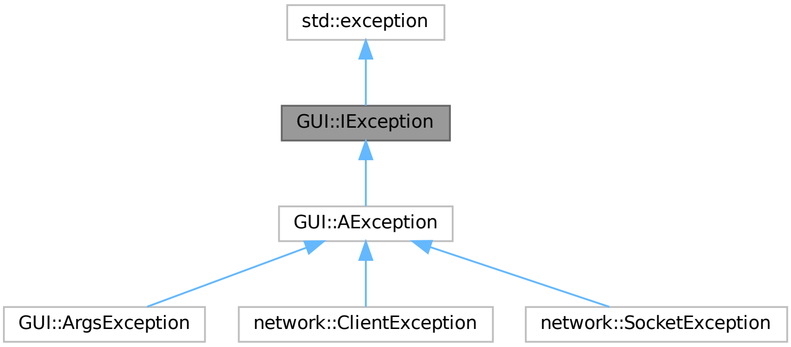
Implemented in network::CommandBCT, network::CommandEBO, network::CommandEDI, network::CommandENW, network::CommandMSZ, network::CommandPBC, network::CommandPDI, network::CommandPDR, network::CommandPEX, network::CommandPFK, network::CommandPGT, network::CommandPIC, network::CommandPIE, network::CommandPIN, network::CommandPLV, network::CommandPNW, network::CommandPPO, network::CommandSBP, network::CommandSEG, network::CommandSGT, network::CommandSMG, network::CommandSST, network::CommandSUC, network::CommandTNA, and network::CommandFactory.

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

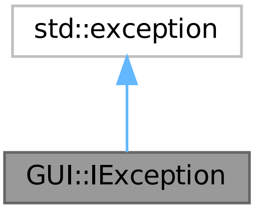
- src/network/commands/interface/ICommand.hpp

5.42 GUI::IException Class Reference

Inheritance diagram for GUI::IException:



Collaboration diagram for GUI::IException:



Public Member Functions

- **IException** () noexcept=default

  Construct a new IException object.
- virtual ∼**IException** () noexcept=default

  Destroy the IException object.
- const char ∗ what () const noexcept override=0

  Get the what object.
- virtual std::string getType () const noexcept=0

  Get the Type object.

5.42.1 Member Function Documentation

5.42.1.1 getType()

```
virtual std::string GUI::IException::getType ( ) const  [pure virtual], [noexcept]
```

Get the Type object.

Returns

std::string (error type)

Implemented in GUI::AException.

5.42.1.2 what()

```
const char * GUI::IException::what ( ) const  [override], [pure virtual], [noexcept]
```

Get the what object.

Returns

const char∗ (error message)

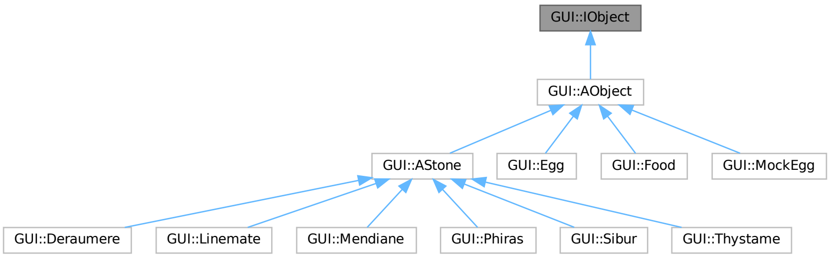
Implemented in GUI::AException.

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

- src/exception/interface/IException.hpp

5.43 GUI::IObject Class Reference

Inheritance diagram for GUI::IObject:



Public Member Functions

- **IObject** () noexcept=default

  Construct a new IObject object.
- virtual ∼**IObject** () noexcept=default

  Destroy the IObject object.
- virtual unsigned int getQuantity () const noexcept=0

  Get the Quantity of the object.
- virtual const Position & getPosition () const noexcept=0

  Get the Position of the tile where the object is.
- virtual Position & getPosition () noexcept=0

  Get the Position of the tile where the object is.
- virtual unsigned int getType () noexcept=0

  Get the Type of the object.
- virtual void setQuantity (unsigned int quantity) noexcept=0

  Set the Quantity of the object.
- virtual std::string getName () const noexcept=0

  get the name of the object

5.43.1 Member Function Documentation

5.43.1.1 getName()

```
virtual std::string GUI::IObject::getName ( ) const  [pure virtual], [noexcept]
```

get the name of the object

Implemented in GUI::AObject, GUI::Food, GUI::Deraumere, GUI::Linemate, GUI::Mendiane, GUI::Phiras, GUI::Sibur, and GUI::Thystame.

5.43.1.2 getPosition() [1/2]

```
virtual const Position & GUI::IObject::getPosition ( ) const  [pure virtual], [noexcept]
```

Get the Position of the tile where the object is.

Returns

const Position& tile where the object is (read-only)

Implemented in GUI::AObject.

5.43.1.3 getPosition() [2/2]

```
virtual Position & GUI::IObject::getPosition ( )  [pure virtual], [noexcept]
```

Get the Position of the tile where the object is.

Returns

Position& tile where the object is (modifiable)

Implemented in GUI::MockEgg, GUI::AObject, and GUI::Egg.

5.43.1.4 getQuantity()

```
virtual unsigned int GUI::IObject::getQuantity ( ) const  [pure virtual], [noexcept]
```

Get the Quantity of the object.

Returns

unsinged int (Quantity of the object)

Implemented in GUI::AObject.

5.43.1.5 getType()

```
virtual unsigned int GUI::IObject::getType ( )  [pure virtual], [noexcept]
```

Get the Type of the object.

Returns

unsigned int (type of the object (follow the protocol))

Implemented in GUI::AObject, GUI::Food, GUI::Deraumere, GUI::Linemate, GUI::Mendiane, GUI::Phiras, GUI::Sibur, and GUI::Thystame.

5.43.1.6 setQuantity()

```
virtual void GUI::IObject::setQuantity (
             unsigned int quantity )  [pure virtual], [noexcept]
```

Set the Quantity of the object.

Parameters

quantity	(Quantity of the object)

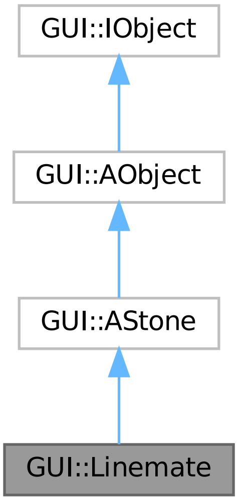
Implemented in GUI::AObject.

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

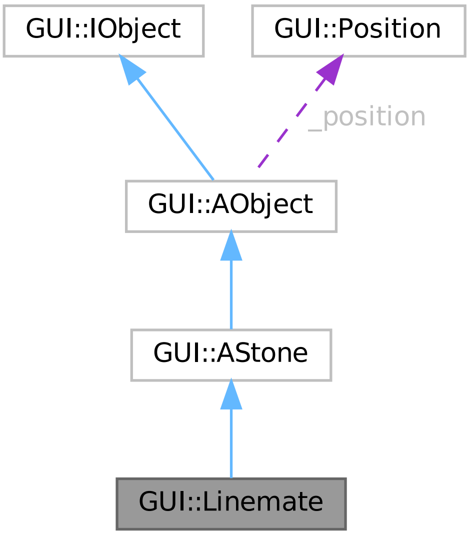
- src/objects/interface/IObject.hpp

5.44 GUI::Linemate Class Reference

Inheritance diagram for GUI::Linemate:



Collaboration diagram for GUI::Linemate:



Public Member Functions

- Linemate (Position tile=Position(0, 0))

    Construct a new Linemate object.
- virtual ∼**Linemate** () noexcept=default

    Destroy the Linemate object.
- unsigned int getType () noexcept final

    Get the Type of the object.
- std::string getName () const noexcept final

    Get the name of the object.

Public Member Functions inherited from GUI::AStone

- AStone (unsigned int quantity, Position tile)

    Construct a new AStone object.
- virtual ∼**AStone** () noexcept=default

    Destroy the AStone object.

Public Member Functions inherited from GUI::AObject

- AObject (unsigned int quantity, Position tile)

    Construct a new AObject object.
- virtual ∼**AObject** () noexcept=default

    Destroy the AObject object.
- unsigned int getQuantity () const noexcept final

    Get the Quantity object.
- const Position & getPosition () const noexcept final

    Get the Position of the tile where the object is.

- Position & getPosition () noexcept

  Get the Position of the tile where the object is.
- virtual void setQuantity (unsigned int quantity) noexcept final

  Set the Quantity of the object.

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

  Construct a new IObject object.
- virtual ~**IObject** () noexcept=default

  Destroy the IObject object.

Additional Inherited Members

Protected Attributes inherited from GUI::AObject

- Position **_position**
- unsigned int **_quantity**

5.44.1 Constructor & Destructor Documentation

5.44.1.1 Linemate()

```
Linemate::Linemate (
             Position tile = Position(0, 0) )
```

Construct a new Linemate object.

Parameters

tile	Tile where the object is

5.44.2 Member Function Documentation

5.44.2.1 getName()

```
std::string Linemate::getName ( ) const  [final], [virtual], [noexcept]
```

Get the name of the object.
Reimplemented from GUI::AObject.

5.44.2.2 getType()

```
unsigned int Linemate::getType ( )  [final], [virtual], [noexcept]
```

Get the Type of the object.

Returns

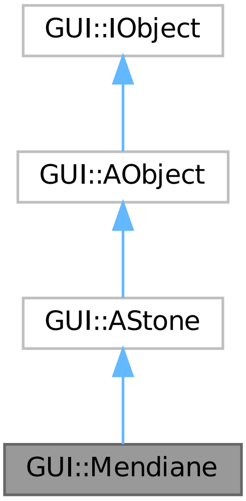
unsigned int (type of the object (follow the protocol))

Reimplemented from GUI::AObject.
The documentation for this class was generated from the following files:

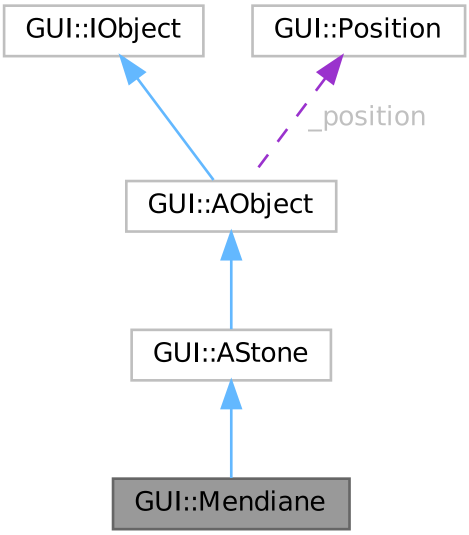
- src/objects/stones/Linemate.hpp
- src/objects/stones/Linemate.cpp

5.45 GUI::Mendiane Class Reference

Inheritance diagram for GUI::Mendiane:



Collaboration diagram for GUI::Mendiane:



Public Member Functions

- Mendiane (Position tile=Position(0, 0))

Construct a new Mendiane object.

- unsigned int getType () noexcept final

  Get the Type of the object.
- std::string getName () const noexcept final

  Get the name of the object.

Public Member Functions inherited from GUI::AStone

- AStone (unsigned int quantity, Position tile)

  Construct a new AStone object.
- virtual ∼**AStone** () noexcept=default

  Destroy the AStone object.

Public Member Functions inherited from GUI::AObject

- AObject (unsigned int quantity, Position tile)

  Construct a new AObject object.
- virtual ∼**AObject** () noexcept=default

  Destroy the AObject object.
- unsigned int getQuantity () const noexcept final

  Get the Quantity object.
- const Position & getPosition () const noexcept final

  Get the Position of the tile where the object is.
- Position & getPosition () noexcept

  Get the Position of the tile where the object is.
- virtual void setQuantity (unsigned int quantity) noexcept final

  Set the Quantity of the object.

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

  Construct a new IObject object.
- virtual ∼**IObject** () noexcept=default

  Destroy the IObject object.

Additional Inherited Members

Protected Attributes inherited from GUI::AObject

- Position **_position**
- unsigned int **_quantity**

5.45.1 Constructor & Destructor Documentation

5.45.1.1 Mendiane()

```
Mendiane::Mendiane (
                Position tile = Position(0, 0) )
```

Construct a new Mendiane object.

Parameters

tile	Tile where the object is

5.45.2 Member Function Documentation

5.45.2.1 getName()

`std::string Mendiane::getName ( ) const [final], [virtual], [noexcept]`

Get the name of the object.
Reimplemented from GUI::AObject.

5.45.2.2 getType()

`unsigned int Mendiane::getType ( ) [final], [virtual], [noexcept]`

Get the Type of the object.

Returns

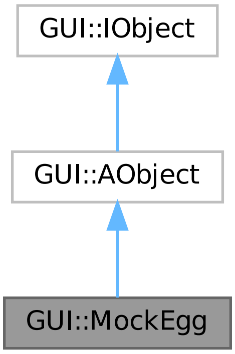
unsigned int (type of the object (follow the protocol))

Reimplemented from GUI::AObject.
The documentation for this class was generated from the following files:

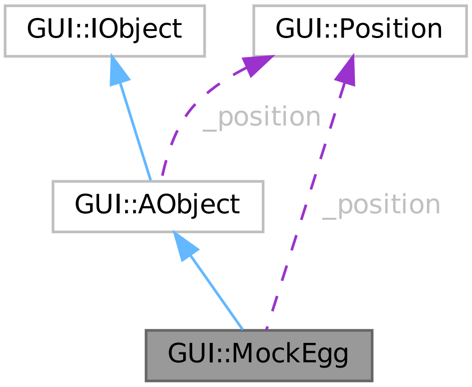
- src/objects/stones/Mendiane.hpp
- src/objects/stones/Mendiane.cpp

5.46 GUI::MockEgg Class Reference

Inheritance diagram for GUI::MockEgg:



Collaboration diagram for GUI::MockEgg:



Public Member Functions

- MockEgg (std::string teamName, std::string ownerID="-1", Position tile=Position(0, 0))

  Construct a new MockEgg object.
- ∼**MockEgg** ()

  Destroy the MockEgg object.
- std::string getTeam () const

  Get the team name.
- std::string getOwnerID () const

  Get the owner ID.
- bool getIsEjected () const

  Get the isEjected.
- void setIsEjected (bool isEjected)

  Set the isEjected.
- void setPosition (int x, int y)

  Set MockEgg position.
- Position & getPosition () noexcept

  Get MockEgg position.

Public Member Functions inherited from GUI::AObject

- AObject (unsigned int quantity, Position tile)

  Construct a new AObject object.
- virtual ∼**AObject** () noexcept=default

  Destroy the AObject object.
- unsigned int getQuantity () const noexcept final

  Get the Quantity object.
- const Position & getPosition () const noexcept final

  Get the Position of the tile where the object is.
- unsigned int getType () noexcept

  Get the Type of the object.
- virtual void setQuantity (unsigned int quantity) noexcept final

Set the Quantity of the object.
- virtual std::string getName () const noexcept

    get the name of the object

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

    Construct a new IObject object.
- virtual ∼**IObject** () noexcept=default

    Destroy the IObject object.

Protected Attributes

- std::string **_teamName**
- std::string **_ownerID**
- bool **_isEjected**
- Position **_position**

Protected Attributes inherited from GUI::AObject

- Position **_position**
- unsigned int **_quantity**

5.46.1 Constructor & Destructor Documentation

5.46.1.1 MockEgg()

```
MockEgg::MockEgg (
            std::string teamName,
            std::string ownerID = "-1",
            Position tile = Position(0, 0) )
```
Construct a new MockEgg object.

Parameters

string	teamName
string	ownerID
Position	Tile where the object is

5.46.2 Member Function Documentation

5.46.2.1 getIsEjected()

```
bool MockEgg::getIsEjected ( ) const
```
Get the isEjected.

Returns

bool (true if the MockEgg is ejected, false if not)

5.46.2.2 getOwnerID()

```
std::string MockEgg::getOwnerID ( ) const
```
Get the owner ID.

Returns

std::string (owner of the MockEgg ID)

5.46.2.3 getPosition()

```
Position & MockEgg::getPosition ( )  [virtual], [noexcept]
```

Get MockEgg position.

Returns

Position

Reimplemented from GUI::AObject.

5.46.2.4 getTeam()

```
std::string MockEgg::getTeam ( ) const
```

Get the team name.

Returns

std::string (team name)

5.46.2.5 setIsEjected()

```
void MockEgg::setIsEjected (
             bool isEjected )
```

Set the isEjected.

Parameters

bool	isEjected

Returns

void (nothing to return)

5.46.2.6 setPosition()

```
void MockEgg::setPosition (
             int x,
             int y )
```

Set MockEgg position.

Parameters

int	x position and y position

Returns

void (nothing to return)

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

- src/mocks/MockEgg.hpp
- src/mocks/MockEgg.cpp

5.47 GUI::MockTeams Class Reference

Public Member Functions

- **MockTeams** (std::string name="")

Create Team object.

- ∼**MockTeams** ()=default

  Destroy Team object.
- void setName (std::string name)

  Set name of the team.
- std::string getName () const

  Get team's name.
- void addMockTrantorians (GUI::MockTrantorians &MockTrantorians)

  Add new MockTrantorians to team.
- std::list< GUI::MockTrantorians > getMockTrantoriansList () const

  Get list of MockTrantorians in the team.
- MockTrantorians getMockTrantoriansById (std::string id)

  Get MockTrantorians by id.

Static Public Member Functions

- static void addMockTeamToMockTeamsList (const GUI::MockTeams &team)

  Add team to teams list.
- static std::list< GUI::MockTeams > & getMockTeamsList ()

  Get list of teams.
- static MockTeams ∗ getTeamByName (const std::string &name)

  Get team by name.

5.47.1 Member Function Documentation

5.47.1.1 addMockTeamToMockTeamsList()

```
void MockTeams::addMockTeamToMockTeamsList (
                const GUI::MockTeams & team )  [static]
```

Add team to teams list.

Parameters

team	GUI::MockTeams object

5.47.1.2 addMockTrantorians()

```
void MockTeams::addMockTrantorians (
                GUI::MockTrantorians & MockTrantorians )
```

Add new MockTrantorians to team.

Parameters

MockTrantorians	GUI::MockTrantorians object

5.47.1.3 getMockTeamsList()

```
std::list< MockTeams > & MockTeams::getMockTeamsList ( )  [static]
```

Get list of teams.

Returns

list of teams objects

5.47.1.4 getMockTrantoriansById()

```
MockTrantorians MockTeams::getMockTrantoriansById (
                std::string id )
```

Get MockTrantorians by id.

Parameters

id	int of MockTrantorians's id

Returns

MockTrantorians object

5.47.1.5 getMockTrantoriansList()

```
std::list< MockTrantorians > MockTeams::getMockTrantoriansList ( ) const
```

Get list of MockTrantorians in the team.

Returns

list of MockTrantorians objects

5.47.1.6 getName()

```
std::string MockTeams::getName ( ) const
```

Get team's name.

Returns

std::string of team's name

5.47.1.7 getTeamByName()

```
MockTeams * MockTeams::getTeamByName (
                const std::string & name )  [static]
```

Get team by name.

Parameters

name	std::string of team's name

Returns

MockTeams object

5.47.1.8 setName()

```
void MockTeams::setName (
                std::string name )
```

Set name of the team.

Parameters

name	std::string of team's name

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

- src/mocks/MockTeams.hpp
- src/mocks/MockTeams.cpp

5.48 GUI::MockTrantorians Class Reference

Public Types

- enum **ResourceType** {
  FOOD , **LINEMATE** , **DERAUMERE** , **SIBUR** ,
  MENDIANE , **PHIRAS** , **THYSTAME** , **RESOURCE_COUNT** }
- enum **Orientation** { **NORTH** = 1 , **EAST** = 2 , **SOUTH** = 3 , **WEST** = 4 }

Public Member Functions

- **MockTrantorians** (std::string id="null", int x=0, int y=0, float orientation=NORTH, int level=1, std::string teamName="")

  Create MockTrantorians object.
- ∼**MockTrantorians** ()=default

  Destroy MockTrantorians object.
- void setPosition (int x, int y)

  Set MockTrantorians position.
- GUI::Position getPosition ()

  Get MockTrantorians postion.
- void setTeam (MockTeams ∗team)

  Set MockTrantorians team.
- MockTeams ∗ getTeam ()

  Get MockTrantorians team.
- void setLifetime (int lifetime)

  Set MockTrantorians's lifetime.
- int getLifetime ()

  Get MockTrantorians lifetime.
- void setIsAlive (bool isAlive)

  Set MockTrantorians alive.
- bool getIsAlive ()

  Get MockTrantorians alive.
- void setAction (bool Action)

  Set MockTrantorians action.
- bool getAction ()

  Get MockTrantorians action.
- void setId (std::string id)

  Set MockTrantorians id.
- std::string getId ()

  Get MockTrantorians id.
- void setLevel (int level)

  Set MockTrantorians level.
- int getLevel ()

  Get MockTrantorians level.
- bool operator== (const MockTrantorians &other) const

  Compare two MockTrantorianss.
- void setOrientation (float orientation)

  Set MockTrantorians orientation.
- float **getOrientation** ()

Get MockTrantorians orientation.

- IObject ∗ createObjectByType (ResourceType type, Position pos)

  Get the object by type.
- void setInventory (std::vector< std::string > inventory)

  Set the MockTrantorians's inventory.
- void clearInventory ()

  Clear the MockTrantorians's inventory.
- void addObject (IObject ∗object)

  Add an object to the tile.
- void removeObject (IObject ∗object)

  Remove an object from the tile.
- std::list< IObject ∗ > getInventory () const

  Get the inventory of the player.

5.48.1 Member Function Documentation

5.48.1.1 addObject()

```
void MockTrantorians::addObject (
                IObject * object )
```

Add an object to the tile.

Parameters

object	(object to add)

5.48.1.2 clearInventory()

```
void MockTrantorians::clearInventory ( )
```

Clear the MockTrantorians's inventory.

Returns

void

5.48.1.3 createObjectByType()

```
IObject * MockTrantorians::createObjectByType (
                ResourceType type,
                Position pos )
```

Get the object by type.

Parameters

type	(type of the object)
pos	(Position of the TILE where the object is supposed to be created)

Returns

IObject∗ (object)

5.48.1.4 getAction()

```
bool MockTrantorians::getAction ( )
```

Get MockTrantorians action.

Returns

true if doing an action, false if not

5.48.1.5 getId()

```
std::string MockTrantorians::getId ( )
```

Get MockTrantorians id.

Returns

int representing player's id

5.48.1.6 getInventory()

```
std::list< IObject * > MockTrantorians::getInventory ( ) const
```

Get the inventory of the player.

Returns

list of IObject*

5.48.1.7 getIsAlive()

```
bool MockTrantorians::getIsAlive ( )
```

Get MockTrantorians alive.

Returns

false if dead, true if alive

5.48.1.8 getLevel()

```
int MockTrantorians::getLevel ( )
```

Get MockTrantorians level.

Returns

int representing player's level

5.48.1.9 getLifetime()

```
int MockTrantorians::getLifetime ( )
```

Get MockTrantorians lifetime.

Returns

int representing life time remaining

5.48.1.10 getPosition()

```
Position MockTrantorians::getPosition ( )
```

Get MockTrantorians postion.

Returns

pair of ints

5.48.1.11 getTeam()

```
MockTeams * MockTrantorians::getTeam ( )
```

Get MockTrantorians team.

Returns

MockTeams∗ pointer to team

5.48.1.12 operator==()

```
bool MockTrantorians::operator== (
            const MockTrantorians & other ) const
```

Compare two MockTrantorianss.

Parameters

other	

Returns

true if equal, false if not

5.48.1.13 removeObject()

```
void MockTrantorians::removeObject (
            IObject * object )
```

Remove an object from the tile.

Parameters

object	(object to remove)

5.48.1.14 setAction()

```
void MockTrantorians::setAction (
            bool Action )
```

Set MockTrantorians action.

Parameters

bool	

5.48.1.15 setId()

```
void MockTrantorians::setId (
            std::string id )
```

Set MockTrantorians id.

Parameters

std::string	

5.48.1.16 setInventory()

```
void MockTrantorians::setInventory (
             std::vector< std::string > inventory )
```

Set the MockTrantorians's inventory.

Parameters

inventory	(vector of strings)

Returns

void

5.48.1.17 setIsAlive()

```
void MockTrantorians::setIsAlive (
             bool isAlive )
```

Set MockTrantorians alive.

Parameters

bool	

5.48.1.18 setLevel()

```
void MockTrantorians::setLevel (
             int level )
```

Set MockTrantorians level.

Parameters

level	

5.48.1.19 setLifetime()

```
void MockTrantorians::setLifetime (
             int lifetime )
```

Set MockTrantorians's lifetime.

Parameters

lifetime	

5.48.1.20 setOrientation()

```
void MockTrantorians::setOrientation (
             float orientation )
```

Set MockTrantorians orientation.

Parameters

orientation	

5.48.1.21 setPosition()

```
void MockTrantorians::setPosition (
              int x,
              int y )
```

Set MockTrantorians position.

Parameters

postion	(int x, int y)

5.48.1.22 setTeam()

```
void MockTrantorians::setTeam (
              MockTeams * team )
```

Set MockTrantorians team.

Parameters

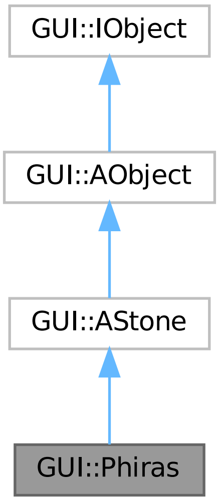
team	

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

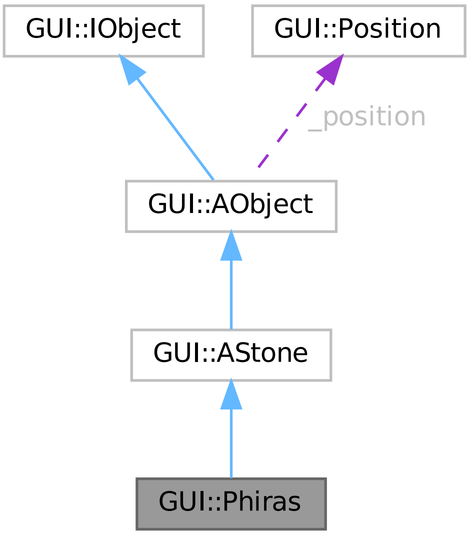
- src/mocks/MockTrantorians.hpp
- src/mocks/MockTrantorians.cpp

5.49 GUI::Phiras Class Reference

Inheritance diagram for GUI::Phiras:



Collaboration diagram for GUI::Phiras:



Public Member Functions

- Phiras (Position tile=Position(0, 0))

  Construct a new Phiras object.
- virtual ∼**Phiras** () noexcept=default

  Destroy the Phiras object.
- unsigned int getType () noexcept final

  Get the Type of the object.
- std::string getName () const noexcept final

  Get the name of the object.

Public Member Functions inherited from GUI::AStone

- AStone (unsigned int quantity, Position tile)

  Construct a new AStone object.
- virtual ∼**AStone** () noexcept=default

  Destroy the AStone object.

Public Member Functions inherited from GUI::AObject

- AObject (unsigned int quantity, Position tile)

  Construct a new AObject object.
- virtual ∼**AObject** () noexcept=default

  Destroy the AObject object.
- unsigned int getQuantity () const noexcept final

  Get the Quantity object.
- const Position & getPosition () const noexcept final

  Get the Position of the tile where the object is.

- Position & getPosition () noexcept

    Get the Position of the tile where the object is.
- virtual void setQuantity (unsigned int quantity) noexcept final

    Set the Quantity of the object.

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

    Construct a new IObject object.
- virtual ∼**IObject** () noexcept=default

    Destroy the IObject object.

Additional Inherited Members

Protected Attributes inherited from GUI::AObject

- Position **_position**
- unsigned int **_quantity**

5.49.1 Constructor & Destructor Documentation

5.49.1.1 Phiras()

```
Phiras::Phiras (
                Position tile = Position(0, 0) )
```

Construct a new Phiras object.

Parameters

tile	Tile where the object is

5.49.2 Member Function Documentation

5.49.2.1 getName()

```
std::string Phiras::getName ( ) const  [final], [virtual], [noexcept]
```

Get the name of the object.
Reimplemented from GUI::AObject.

5.49.2.2 getType()

```
unsigned int Phiras::getType ( )  [final], [virtual], [noexcept]
```

Get the Type of the object.

Returns

unsigned int (type of the object (follow the protocol))

Reimplemented from GUI::AObject.
The documentation for this class was generated from the following files:

- src/objects/stones/Phiras.hpp
- src/objects/stones/Phiras.cpp

5.50 GUI::Position Class Reference

Public Member Functions

- Position (unsigned int x=0, unsigned int y=0)

  Construct a new Position object.
- unsigned int getX () const noexcept

  Get the X position.
- unsigned int getY () const noexcept

  Get the Y position.
- void setX (unsigned int x) noexcept

  Set the X position.
- void setY (unsigned int y) noexcept

  Set the Y position.

Protected Attributes

- unsigned int **_x**
- unsigned int **_y**

5.50.1 Constructor & Destructor Documentation

5.50.1.1 Position()

```
Position::Position (
             unsigned int x = 0,
             unsigned int y = 0 )
```

Construct a new Position object.

Parameters

x	(x position) default is 0
y	(y position) default is 0

5.50.2 Member Function Documentation

5.50.2.1 getX()

```
unsigned int Position::getX ( ) const  [noexcept]
```

Get the X position.

Returns

unsigned int (x position)

5.50.2.2 getY()

```
unsigned int Position::getY ( ) const  [noexcept]
```

Get the Y position.

Returns

unsigned int (y position)

5.50.2.3 setX()

```
void Position::setX (
             unsigned int x )  [noexcept]
```

Set the X position.

Parameters

x	(x position)

5.50.2.4 setY()

```
void Position::setY (
                unsigned int y )  [noexcept]
```

Set the Y position.

Parameters

y	(y position)

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

- src/position/Position.hpp
- src/position/Position.cpp

5.51 network::ProtocolHandler Class Reference

Public Member Functions

- ProtocolHandler (CommandFactory &factory)

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

  Destroy the Protocol Handler object.
- void handleData (const std::vector< std::string > &data)

  Handle the data.

5.51.1 Constructor & Destructor Documentation

5.51.1.1 ProtocolHandler()

```
ProtocolHandler::ProtocolHandler (
                CommandFactory & factory )
```

Construct a new Protocol Handler object.

Parameters

CommandFactory	&factory

5.51.2 Member Function Documentation

5.51.2.1 handleData()

```
void ProtocolHandler::handleData (
                const std::vector< std::string > & data )
```

Handle the data.

Parameters

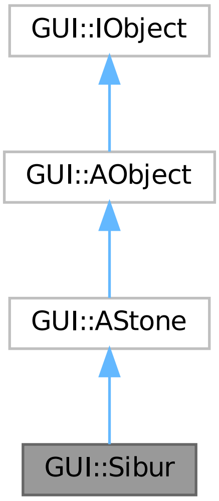
std::vector<std::string>	&data

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

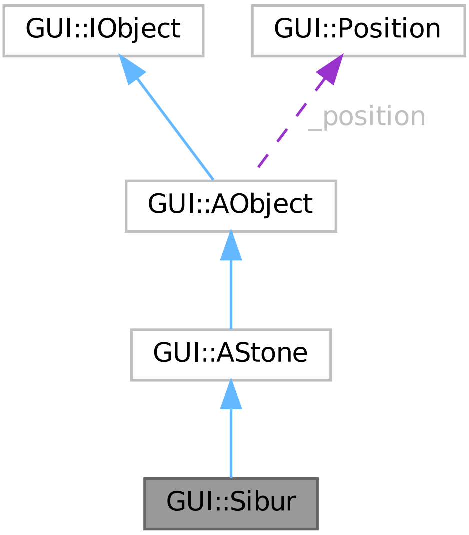
- src/network/protocol_handler/ProtocolHandler.hpp
- src/network/protocol_handler/ProtocolHandler.cpp

5.52 GUI::Sibur Class Reference

Inheritance diagram for GUI::Sibur:



Collaboration diagram for GUI::Sibur:



Public Member Functions

- Sibur (Position tile=Position(0, 0))

  Construct a new Sibur object.
- virtual ∼**Sibur** () noexcept=default

  Destroy the Sibur object.
- unsigned int getType () noexcept final

  Get the Type of the object.
- std::string getName () const noexcept final

  Get the name of the object.

Public Member Functions inherited from GUI::AStone

- AStone (unsigned int quantity, Position tile)

  Construct a new AStone object.
- virtual ∼**AStone** () noexcept=default

  Destroy the AStone object.

Public Member Functions inherited from GUI::AObject

- AObject (unsigned int quantity, Position tile)

  Construct a new AObject object.
- virtual ∼**AObject** () noexcept=default

  Destroy the AObject object.
- unsigned int getQuantity () const noexcept final

  Get the Quantity object.
- const Position & getPosition () const noexcept final

  Get the Position of the tile where the object is.
- Position & getPosition () noexcept

  Get the Position of the tile where the object is.
- virtual void setQuantity (unsigned int quantity) noexcept final

  Set the Quantity of the object.

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

  Construct a new IObject object.
- virtual ∼**IObject** () noexcept=default

  Destroy the IObject object.

Additional Inherited Members

Protected Attributes inherited from GUI::AObject

- Position **_position**
- unsigned int **_quantity**

5.52.1 Constructor & Destructor Documentation

5.52.1.1 Sibur()

```
Sibur::Sibur (
                Position tile = Position(0, 0) )
```

Construct a new Sibur object.

Parameters

tile	Tile where the object is

5.52.2 Member Function Documentation

5.52.2.1 getName()

```
std::string Sibur::getName ( ) const  [final], [virtual], [noexcept]
```

Get the name of the object.
Reimplemented from GUI::AObject.

5.52.2.2 getType()

```
unsigned int Sibur::getType ( )  [final], [virtual], [noexcept]
```

Get the Type of the object.

Returns

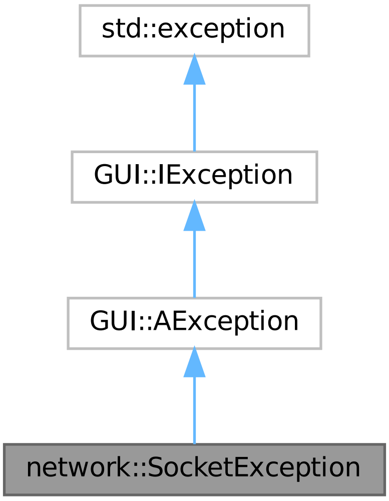
unsigned int (type of the object (follow the protocol))

Reimplemented from GUI::AObject.
The documentation for this class was generated from the following files:

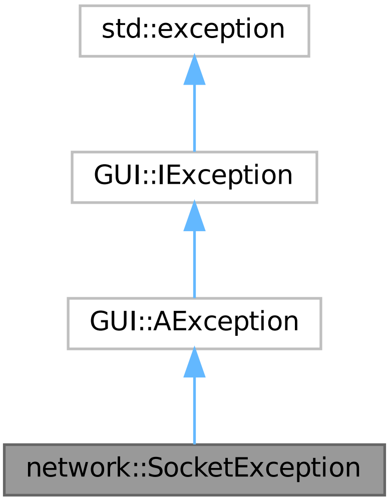
- src/objects/stones/Sibur.hpp
- src/objects/stones/Sibur.cpp

5.53 network::SocketException Class Reference

Inheritance diagram for network::SocketException:



Collaboration diagram for network::SocketException:



Public Member Functions

- SocketException (std::string message)

    Construct a new SocketException object.

Public Member Functions inherited from GUI::AException

- AException (std::string message, std::string type) noexcept

    Construct a new AException object.
- virtual ∼**AException** () noexcept=default

    Destroy the AException object.
- const char ∗ what () const noexcept final

    Get the message object.
- std::string getType () const noexcept final

    Get the Type object.

Public Member Functions inherited from GUI::IException

- **IException** () noexcept=default

    Construct a new IException object.
- virtual ∼**IException** () noexcept=default

    Destroy the IException object.

Additional Inherited Members

Protected Attributes inherited from GUI::AException

- std::string **_message**
- std::string **_type**

5.53.1 Constructor & Destructor Documentation

5.53.1.1 SocketException()

```
SocketException::SocketException (
                std::string message )
```

Construct a new SocketException object.

Parameters

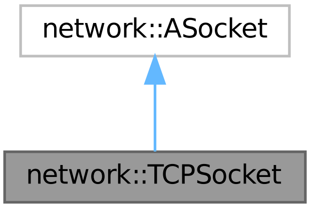
message	(error message)

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

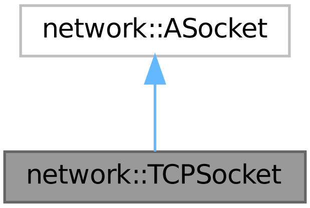
- src/exception/SocketException.hpp
- src/exception/SocketException.cpp

5.54 network::TCPSocket Class Reference

Inheritance diagram for network::TCPSocket:



Collaboration diagram for network::TCPSocket:



Public Member Functions

- **TCPSocket** ()

  Construct a new TCPSocket object and set it to -1.
- ∼**TCPSocket** ()

Destroy the TCPSocket object, close the socket.

- void connect (const std::string &hostname, unsigned int port) final

  Connect to the server.
- void close () final

  Close the connection to the server.
- void send (const std::string &data) final

  Send data to the server.
- std::vector< std::string > receive () final

  Receive data from the server.
- int getSockfd () const

  Get the socket file descriptor.

Public Member Functions inherited from network::ASocket

- **ASocket** ()=default

  Construct a new ASocket object.
- virtual ∼**ASocket** () noexcept=default

  Destroy the ASocket object.

Protected Attributes

- int **_sockfd**

  Socket file descriptor.
- struct sockaddr_in **serv_addr**

  Server address.
- std::string **_partialBuffer**

5.54.1 Member Function Documentation

5.54.1.1 close()

```
void TCPSocket::close ( )  [final], [virtual]
```

Close the connection to the server.

Implements network::ASocket.

5.54.1.2 connect()

```
void TCPSocket::connect (
             const std::string & hostname,
             unsigned int port )  [final], [virtual]
```

Connect to the server.

Parameters

hostname	Hostname of the server
port	Port of the server

Implements network::ASocket.

5.54.1.3 getSockfd()

```
int TCPSocket::getSockfd ( ) const  [virtual]
```

Get the socket file descriptor.

Returns

int The socket file descriptor

Implements network::ASocket.

5.54.1.4 receive()

```
std::vector< std::string > TCPSocket::receive ( )  [final], [virtual]
```

Receive data from the server.

Returns

std::vector<std::string> Data received

Implements network::ASocket.

5.54.1.5 send()

```
void TCPSocket::send (
                const std::string & data )  [final], [virtual]
```

Send data to the server.

Parameters

data	Data to send

Implements network::ASocket.

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

- src/network/socket/TCPSocket.hpp
- src/network/socket/TCPSocket.cpp

5.55 GUI::Teams Class Reference

Public Member Functions

- **Teams** (std::string name="")

  Create Team object.
- ∼**Teams** ()=default

  Destroy Team object.
- void setName (std::string name)

  Set name of the team.
- std::string getName () const

  Get team's name.
- void addTrantorian (Trantorian &trantorian)

  Add new trantorian to team.
- std::list< Trantorian > getTrantorianList () const

  Get list of trantorian in the team.
- Trantorian getTrantorianById (std::string id)

  Get trantorian by id.
- bool **hasTrantorian** (const std::string &id) const
- void addEggToList (Egg newEgg)

  Add a new egg to egg's list.
- std::list< Egg > getEggList () const

  Get list of egg in the team.
- Trantorian ∗ getTrantorianByIdMod (const std::string &id)

  Get trantorian by id.

Static Public Member Functions

- static void addTeamToTeamsList (const GUI::Teams &team)

  Add team to teams list.
- static std::list< GUI::Teams > & getTeamsList ()

  Get list of teams.
- static Teams ∗ getTeamByName (const std::string &name)

  Get team by name.

5.55.1 Member Function Documentation

5.55.1.1 addEggToList()

```
void Teams::addEggToList (
                Egg newEgg )
```

Add a new egg to egg's list.

Parameters

Egg	egg object

Returns

void (nothing to return)

5.55.1.2 addTeamToTeamsList()

```
void Teams::addTeamToTeamsList (
                const GUI::Teams & team )  [static]
```

Add team to teams list.

Parameters

team	GUI::Teams object

5.55.1.3 addTrantorian()

```
void Teams::addTrantorian (
                Trantorian & trantorian )
```

Add new trantorian to team.

Parameters

trantorian	GUI::Trantorian object

5.55.1.4 getEggList()

```
std::list< Egg > Teams::getEggList ( ) const
```

Get list of egg in the team.

Returns

list of egg objects

5.55.1.5 getName()

```
std::string Teams::getName ( ) const
```

Get team's name.

Returns

std::string of team's name

5.55.1.6 getTeamByName()

```
Teams * Teams::getTeamByName (
            const std::string & name ) [static]
```

Get team by name.

Parameters

name	std::string of team's name

Returns

Teams object

5.55.1.7 getTeamsList()

```
std::list< Teams > & Teams::getTeamsList ( ) [static]
```

Get list of teams.

Returns

list of teams objects

5.55.1.8 getTrantorianById()

```
Trantorian Teams::getTrantorianById (
            std::string id )
```

Get trantorian by id.

Parameters

id	int of trantorian's id

Returns

Trantorian object

5.55.1.9 getTrantorianByIdMod()

```
Trantorian * Teams::getTrantorianByIdMod (
            const std::string & id )
```

Get trantorian by id.

Parameters

id	std::string of trantorian's id

Returns

Trantorian object

Note

this function is used to get a modifiable trantorian

5.55.1.10 getTrantorianList()

```
std::list< Trantorian > Teams::getTrantorianList ( ) const
```

Get list of trantorian in the team.

Returns

list of trantorian objects

5.55.1.11 setName()

```
void Teams::setName (
             std::string name )
```

Set name of the team.

Parameters

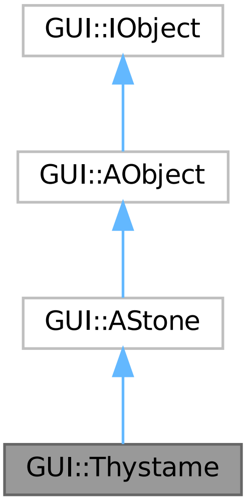
name	std::string of team's name

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

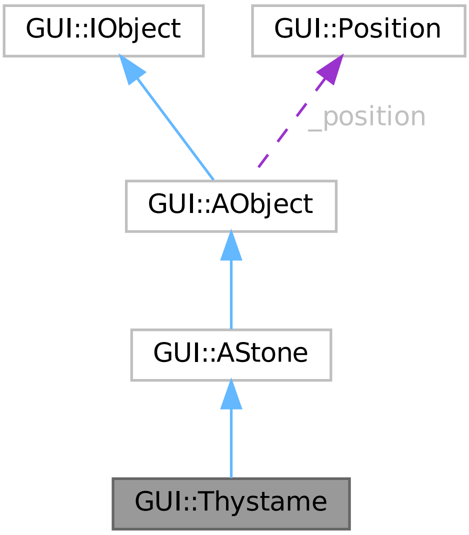
- src/trantorians/Teams.hpp
- src/trantorians/Teams.cpp

5.56 GUI::Thystame Class Reference

Inheritance diagram for GUI::Thystame:



Collaboration diagram for GUI::Thystame:



Public Member Functions

- Thystame (Position tile=Position(0, 0))

Construct a new Thystame object.

- virtual ∼**Thystame** () noexcept=default

  Destroy the Thystame object.

- unsigned int getType () noexcept final

  Get the Type of the object.

- std::string getName () const noexcept final

  Get the name of the object.

Public Member Functions inherited from GUI::AStone

- AStone (unsigned int quantity, Position tile)

  Construct a new AStone object.

- virtual ∼**AStone** () noexcept=default

  Destroy the AStone object.

Public Member Functions inherited from GUI::AObject

- AObject (unsigned int quantity, Position tile)

  Construct a new AObject object.

- virtual ∼**AObject** () noexcept=default

  Destroy the AObject object.

- unsigned int getQuantity () const noexcept final

  Get the Quantity object.

- const Position & getPosition () const noexcept final

  Get the Position of the tile where the object is.

- Position & getPosition () noexcept

  Get the Position of the tile where the object is.

- virtual void setQuantity (unsigned int quantity) noexcept final

  Set the Quantity of the object.

Public Member Functions inherited from GUI::IObject

- **IObject** () noexcept=default

  Construct a new IObject object.

- virtual ∼**IObject** () noexcept=default

  Destroy the IObject object.

Additional Inherited Members

Protected Attributes inherited from GUI::AObject

- Position **_position**
- unsigned int **_quantity**

5.56.1 Constructor & Destructor Documentation

5.56.1.1 Thystame()

```
Thystame::Thystame (
            Position tile = Position(0, 0) )
```

Construct a new Thystame object.

Parameters

tile	Tile where the object is

5.56.2 Member Function Documentation

5.56.2.1 getName()

```
std::string Thystame::getName ( ) const  [final], [virtual], [noexcept]
```

Get the name of the object.
Reimplemented from GUI::AObject.

5.56.2.2 getType()

```
unsigned int Thystame::getType ( )  [final], [virtual], [noexcept]
```

Get the Type of the object.

Returns

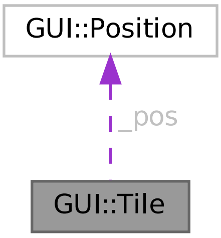
unsigned int (type of the object (follow the protocol))

Reimplemented from GUI::AObject.
The documentation for this class was generated from the following files:

- src/objects/stones/Thystame.hpp
- src/objects/stones/Thystame.cpp

5.57 GUI::Tile Class Reference

Collaboration diagram for GUI::Tile:



Public Types

- enum **ResourceType** {
  FOOD , **LINEMATE** , **DERAUMERE** , **SIBUR** ,
  MENDIANE , **PHIRAS** , **THYSTAME** , **RESOURCE_COUNT** }

Public Member Functions

- Tile (Position pos)

  Construct a new Tile object.
- ∼**Tile** ()=default

  Destroy the Tile object.
- Position getPosition () const

  Get the Position of the tile.
- void addObject (IObject ∗object)

  Add an object to the tile.

- void removeObject (IObject ∗object)

  Remove an object from the tile.
- std::list< IObject ∗ > getObjects () const

  Get the list of objects on the tile.
- void **clearObjects** ()

  Clear all objects on the tile.
- void updateTileContent (const std::vector< std::string > &tileContent)

  Get the object by type.
- IObject ∗ createObjectByType (ResourceType type, Position pos)

  Get the object by type.
- BoundingBox getBounds () const

  get the bounds of the tile

Protected Attributes

- std::list< IObject ∗ > **_objects**
- Position **_pos**

5.57.1 Constructor & Destructor Documentation

5.57.1.1 Tile()

```
Tile::Tile (
            Position pos )
```

Construct a new Tile object.

Parameters

pos	(position of the tile)

5.57.2 Member Function Documentation

5.57.2.1 addObject()

```
void Tile::addObject (
            IObject * object )
```

Add an object to the tile.

Parameters

object	(object to add)

5.57.2.2 createObjectByType()

```
IObject * Tile::createObjectByType (
            ResourceType type,
            Position pos )
```

Get the object by type.

Parameters

type	(type of the object)
pos	(position of the Tile where the object is supposed to be created)

Returns

IObject∗ (object)

5.57.2.3 getBounds()

```
BoundingBox GUI::Tile::getBounds ( ) const
```

get the bounds of the tile

Returns

BoundingBox (bounding box of the tile)

5.57.2.4 getObjects()

```
std::list< IObject * > Tile::getObjects ( ) const
```

Get the list of objects on the tile.

Returns

std::list<IObject ∗> (list of objects)

5.57.2.5 getPosition()

```
Position Tile::getPosition ( ) const
```

Get the Position of the tile.

Returns

Position (position of the tile)

5.57.2.6 removeObject()

```
void Tile::removeObject (
            IObject * object )
```

Remove an object from the tile.

Parameters

object	(object to remove)

5.57.2.7 updateTileContent()

```
void Tile::updateTileContent (
            const std::vector< std::string > & tileContent )
```

Get the object by type.

Parameters

type	(type of the object)

Returns

IObject∗ (object)

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

- src/tiles/Tile.hpp
- src/tiles/Tile.cpp

5.58 GUI::Trantorian Class Reference

Public Types

- enum **ResourceType** {
  FOOD , **LINEMATE** , **DERAUMERE** , **SIBUR** ,
  MENDIANE , **PHIRAS** , **THYSTAME** , **RESOURCE_COUNT** }
- enum **Orientation** { **NORTH** = 1 , **EAST** = 2 , **SOUTH** = 3 , **WEST** = 4 }

Public Member Functions

- **Trantorian** (std::string id="null", int x=0, int y=0, float orientation=NORTH, int level=1)

  Create trantorian object.
- ∼**Trantorian** ()=default

  Destroy trantorian object.
- void setPosition (int x, int y)

  Set trantorian position.
- const GUI::Position & getPosition () const

  Get trantorian postion.
- void setLifetime (int lifetime)

  Set trantorian's lifetime.
- int getLifetime ()

  Get trantorian lifetime.
- void setIsAlive (bool isAlive)

  Set trantorian alive.
- bool getIsAlive ()

  Get trantorian alive.
- void setAction (bool Action)

  Set trantorian action.
- bool getAction ()

  Get trantorian action.
- void setId (std::string id)

  Set trantorian id.
- std::string getId () const

  Get trantorian id.
- void setLevel (int level)

  Set trantorian level.
- int getLevel ()

  Get trantorian level.
- bool operator== (const Trantorian &other) const

  Compare two trantorians.
- void setOrientation (float orientation)

  Set trantorian orientation.
- float **getOrientation** ()

  Get trantorian orientation.
- IObject ∗ createObjectByType (ResourceType type, Position pos)

  Get the object by type.
- void setInventory (std::vector< std::string > inventory)

  Set the trantorian's inventory.
- void clearInventory ()

  Clear the trantorian's inventory.
- void addObject (IObject ∗object)

Add an object to the tile.

- void removeObject (IObject ∗object)

  Remove an object from the tile.
- std::list< IObject ∗ > getInventory () const

  Get the inventory of the player.
- Model **getModel** () const

  Get the Model object.
- Texture2D **getTexture** () const

  Get the Texture object.

5.58.1 Member Function Documentation

5.58.1.1 addObject()

```
void Trantorian::addObject (
            IObject ∗ object )
```

Add an object to the tile.

Parameters

object	(object to add)

5.58.1.2 clearInventory()

```
void Trantorian::clearInventory ( )
```

Clear the trantorian's inventory.

Returns

void

5.58.1.3 createObjectByType()

```
IObject ∗ Trantorian::createObjectByType (
            ResourceType type,
            Position pos )
```

Get the object by type.

Parameters

type	(type of the object)
pos	(Position of the TILE where the object is supposed to be created)

Returns

IObject∗ (object)

5.58.1.4 getAction()

```
bool Trantorian::getAction ( )
```

Get trantorian action.

Returns

true if doing an action, false if not

5.58.1.5 getId()

```
std::string Trantorian::getId ( ) const
```

Get trantorian id.

Returns

int representing player's id

5.58.1.6 getInventory()

```
std::list< IObject * > Trantorian::getInventory ( ) const
```

Get the inventory of the player.

Returns

list of IObject*

5.58.1.7 getIsAlive()

```
bool Trantorian::getIsAlive ( )
```

Get trantorian alive.

Returns

false if dead, true if alive

5.58.1.8 getLevel()

```
int Trantorian::getLevel ( )
```

Get trantorian level.

Returns

int representing player's level

5.58.1.9 getLifetime()

```
int Trantorian::getLifetime ( )
```

Get trantorian lifetime.

Returns

int representing life time remaining

5.58.1.10 getPosition()

```
const GUI::Position & Trantorian::getPosition ( ) const
```

Get trantorian postion.

Returns

pair of ints

5.58.1.11 operator==()

```
bool Trantorian::operator== (
             const Trantorian & other ) const
```

Compare two trantorians.

Parameters

other	

Returns

true if equal, false if not

5.58.1.12 removeObject()

```
void Trantorian::removeObject (
            IObject * object )
```

Remove an object from the tile.

Parameters

object	(object to remove)

5.58.1.13 setAction()

```
void Trantorian::setAction (
            bool Action )
```

Set trantorian action.

Parameters

bool	

5.58.1.14 setId()

```
void Trantorian::setId (
            std::string id )
```

Set trantorian id.

Parameters

std::string	

5.58.1.15 setInventory()

```
void Trantorian::setInventory (
            std::vector< std::string > inventory )
```

Set the trantorian's inventory.

Parameters

inventory	(vector of strings)

Returns

void

5.58.1.16 setIsAlive()

```
void Trantorian::setIsAlive (
            bool isAlive )
```

Set trantorian alive.

Parameters

bool	

5.58.1.17 setLevel()

```
void Trantorian::setLevel (
            int level )
```

Set trantorian level.

Parameters

level	

5.58.1.18 setLifetime()

```
void Trantorian::setLifetime (
            int lifetime )
```

Set trantorian's lifetime.

Parameters

lifetime	

5.58.1.19 setOrientation()

```
void Trantorian::setOrientation (
            float orientation )
```

Set trantorian orientation.

Parameters

orientation	

5.58.1.20 setPosition()

```
void Trantorian::setPosition (
            int x,
            int y )
```

Set trantorian position.

Parameters

postion	(int x, int y)

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

- src/trantorians/Trantorian.hpp
- src/trantorians/Trantorian.cpp

5.59 GUI::World Class Reference

Public Member Functions

- World (unsigned int width=0, unsigned int height=0)

  Construct a new World object.
- ∼**World** ()=default

  Destroy the World object.
- void setWidth (unsigned int width)

  Set the Width of the world.
- void setHeight (unsigned int height)

  Set the Height of the world.
- unsigned int getWidth () const

  Get the Width of the world.
- unsigned int getHeight () const

  Get the Height of the world.
- void addObject (IObject ∗object, Position pos)

  Add an object to the world at a specific position.
- void removeObject (IObject ∗object, Position pos)

  Remove an object from the world.
- std::list< IObject ∗ > getObjects () const

  Get the list of objects in the world.
- std::list< IObject ∗ > getObjectsAt (Position tile) const

  Get the list of objects at a specific tile.
- void setWorldSize (unsigned int width, unsigned int height)

  Set the size of the world and initialize tiles.
- std::vector< std::vector< Tile > > & getTiles ()

  Get the tiles of the world.
- Tile & getTileAt (unsigned int x, unsigned int y)

  Get the tile at the specified position.
- void addTile (Tile tile)

  Add a tile to the world.

Protected Attributes

- unsigned int **_width**
- unsigned int **_height**
- std::vector< std::vector< Tile > > **_tiles**

5.59.1 Constructor & Destructor Documentation

5.59.1.1 World()

```
World::World (
            unsigned int width = 0,
            unsigned int height = 0 )
```

Construct a new World object.

Parameters

width	(width of the world x)
height	(height of the world y)

5.59.2 Member Function Documentation

5.59.2.1 addObject()

```
void World::addObject (
             IObject * object,
             Position pos )
```

Add an object to the world at a specific position.

Parameters

object	(object to add)
pos	(position to add the object)

5.59.2.2 addTile()

```
void World::addTile (
             Tile tile )
```

Add a tile to the world.

Parameters

tile	(tile to add)

5.59.2.3 getHeight()

```
unsigned int World::getHeight ( ) const
```

Get the Height of the world.

Returns

unsigned int (height of the world y)

5.59.2.4 getObjects()

```
std::list< IObject * > World::getObjects ( ) const
```

Get the list of objects in the world.

Returns

std::list<IObject *> (list of objects)

5.59.2.5 getObjectsAt()

```
std::list< IObject * > World::getObjectsAt (
             Position tile ) const
```

Get the list of objects at a specific tile.

Parameters

tile	(Position object that contains uint X and uint Y as protected)

5.59.2.6 getTileAt()

```
Tile & World::getTileAt (
            unsigned int x,
            unsigned int y )
```

Get the tile at the specified position.

Parameters

x	(x coordinate)
y	(y coordinate)

Returns

Tile& (reference to the tile)

5.59.2.7 getTiles()

```
std::vector< std::vector< Tile > > & World::getTiles ( )
```

Get the tiles of the world.

Returns

const std::vector<std::vector<Tile>>& (2D vector of tiles)

5.59.2.8 getWidth()

```
unsigned int World::getWidth ( ) const
```

Get the Width of the world.

Returns

unsigned int (width of the world x)

5.59.2.9 removeObject()

```
void World::removeObject (
            IObject * object,
            Position pos )
```

Remove an object from the world.

Parameters

object	(object to remove)

5.59.2.10 setHeight()

```
void World::setHeight (
            unsigned int height )
```

Set the Height of the world.

Parameters

height	(height of the world y)

5.59.2.11 setWidth()

```
void World::setWidth (
             unsigned int width )
```

Set the Width of the world.

Parameters

width	(width of the world x)

5.59.2.12 setWorldSize()

```
void World::setWorldSize (
             unsigned int width,
             unsigned int height )
```

Set the size of the world and initialize tiles.

Parameters

width	(width of the world x)
height	(height of the world y)

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

- src/world/World.hpp
- src/world/World.cpp

Chapter 6

File Documentation

6.1 AException.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** AException
*/

#ifndef AEXCEPTION_HPP_
    #define AEXCEPTION_HPP_

    #include "exception/interface/IException.hpp"
    #include <string>

namespace GUI {
    class AException : public IException {
    public:
        AException(std::string message, std::string type) noexcept;
        virtual ~AException() noexcept = default;

        [[nodiscard]] const char *what() const noexcept final;
        [[nodiscard]] std::string getType() const noexcept final;

    protected:
        std::string _message;       // error message
        std::string _type;          // error type (ArgsException, ...)
    private:
    };
} // namespace GUI //
#endif /* !AEXCEPTION_HPP_ */
```

6.2 ArgsException.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** ArgsException
*/

#ifndef ARGSEXCEPTION_HPP_
    #define ARGSEXCEPTION_HPP_

    #include "abstract/AException.hpp"

namespace GUI {
    class ArgsException : public AException {
    public:
        ArgsException(std::string message);

    protected:
    };
} // namespace GUI //
#endif /* !ARGSEXCEPTION_HPP_ */
```

6.3 ClientException.hpp

```
/*
```

```
** EPITECH PROJECT, 2024
** Zappy
** File description:
** ClientException
*/

#ifndef CLIENTEXCEPTION_HPP_
    #define CLIENTEXCEPTION_HPP_

    #include "abstract/AException.hpp"

namespace network {
    class ClientException : public GUI::AException {
    public:
        ClientException(std::string message);
    };
} // namespace network //
#endif /* !CLIENTEXCEPTION_HPP_ */
```

6.4 IException.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** IException
*/

#ifndef IEXCEPTION_HPP_
    #define IEXCEPTION_HPP_

    #include <exception>
    #include <string>

namespace GUI {
    class IException : public std::exception {
    public:
        IException() noexcept = default;
        virtual ~IException() noexcept = default;

        [[nodiscard]] const char* what() const noexcept override = 0;
        [[nodiscard]] virtual std::string getType() const noexcept = 0;
    };
} // namespace GUI //
#endif /* !IEXCEPTION_HPP_ */
```

6.5 SocketException.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** SocketException
*/

#ifndef SOCKETEXCEPTION_HPP_
    #define SOCKETEXCEPTION_HPP_

    #include "abstract/AException.hpp"

namespace network {
    class SocketException : public GUI::AException {
    public:
        SocketException(std::string message);

    protected:
    };
} // namespace network //
#endif /* !SOCKETEXCEPTION_HPP_ */
```

6.6 Display.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Display
*/

```

```
#ifndef DISPLAY_HPP_
    #define DISPLAY_HPP_

    #include <sstream>
    #include <raylib.h>
    #include "world/World.hpp"
    #include "trantorians/Teams.hpp"
    #include "trantorians/Trantorian.hpp"
    #include "game/events/Events.hpp"
    #include <cstring>

namespace GUI {
    class Display : public Events {
    public:
        Display(World &world, Teams &teams);
        ~Display() = default;

        void displayElements();

        void DrawTiles(std::vector<std::vector<Tile» tiles);
        void DrawClouds();
        void DrawTrantorians(std::list<Teams> teams);
        void DrawTileInfo();
        void DrawTrantorianInfo();
        void DrawObjects(std::list<IObject*> objects);
        void DrawEgg();
        void DrawScoreBoard(Teams &teams);
        void cleanupModels();
        void initClouds();

        bool windowShouldClose();
        void updateCamera();
        void closeWindow();

        std::vector<Model> getClouds() const;

        void DrawSSTBox();

        std::string getNewTimeUnit();

        void DisplayHelpMenu();

        void DisplayGameInformations();

        void setTimeUnit(unsigned int timeUnit);

        void setNewTimeUnit(std::string newTimeUnit);

        void addLog(const std::string& log);
        void DrawLogs();

    protected:
        Camera _camera;                          // Camera
        std::vector<Model> _clouds;              // Clouds models
        std::vector<Vector3> _cloudPositions;    // Clouds positions
        World &_world;                           // World
        Teams &_teams;                           // Teams
        unsigned int _timeUnit;                  // Time unit

        char _inputText[256] = "";
        bool _textBoxActive = false;
        int _framesCounter = 0;
        int _ignoreInputFrames = 0;
        std::string _newTimeUnit;
        bool _gameInfo = false;
        bool _drawLogs = false;
        std::vector<std::string> _logs;
    };
}

#endif // DISPLAY_HPP_
```

6.7 Events.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Events
*/

#ifndef EVENTS_HPP_
    #define EVENTS_HPP_

    #include "tiles/Tile.hpp"
    #include "world/World.hpp"
    #include "trantorians/Teams.hpp"

    #include <memory>
    #include <raylib.h>
    #include <iostream>
    #include <vector>
    #include <algorithm>

namespace GUI {
    class Events {
    public:
        Events();
        ~Events() = default;

        void detectHoveredTile(Camera _camera, World &_world);
        void detectHoveredTrantorian(Camera _camera, Teams &_teams);

    protected:
        Tile* _selectedTile;                // tile selected by the user
        Tile* _hoveredTile;                 // tile hovered by the user

        std::unique_ptr<Trantorian> _selectedTrantorian;    // trantorian selected by the user
        std::unique_ptr<Trantorian> _hoveredTrantorian;     // trantorian hovered by the user
    };
} // namespace GUI //
#endif /* !EVENTS_HPP_ */
```

6.8 Game.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Game
*/

#ifndef GAME_HPP_
#define GAME_HPP_

    #include <sstream>
    #include <raylib.h>
    #include "world/World.hpp"
    #include "trantorians/Teams.hpp"
    #include "trantorians/Trantorian.hpp"
    #include "network/Client.hpp"
    #include "network/commands/factory/CommandFactory.hpp"
    #include "network/protocol_handler/ProtocolHandler.hpp"
    #include "display/Display.hpp"

namespace GUI {
    class Game {
    public:
        Game(std::string hostname, unsigned int port);
        ~Game() = default;

        void initGame();
        void runGame();

        void createWorld(std::vector<std::string> data);
        void initTimeUnit(std::vector<std::string> data);

        void handleNewTimeUnit();
        void initializeCallbacks();

        void ensureGameInit();

        void ensureGameInformation();
    private:
        World _world;                            // World object
        Teams _teams;                            // Teams object
        network::Client _client;                 // Client object
        network::CommandFactory _commandFactory; // CommandFactory object
        unsigned int _timeUnit;                  // Time unit of the game
        Display _display;                        // Display object
    };
} // namespace GUI //
#endif /* !GAME_HPP_ */

```

6.9 HandleArgs.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** HandleArgs
*/

#ifndef HANDLEARGS_HPP_
    #define HANDLEARGS_HPP_

    #include <string>
    #include "exception/ArgsException.hpp"

namespace GUI {
    class HandleArgs {
    public:
        HandleArgs() = default;
        ~HandleArgs() = default;

        [[nodiscard]] int checkArgs(int nbArgs, char **args);
        [[nodiscard]] int checkPort(const std::string &port);
        [[nodiscard]] int checkHostname(const std::string &hostname);

        [[nodiscard]] unsigned int getPort() const;

        [[nodiscard]] std::string getHostname() const;

        void printUsage();
    protected:
        unsigned int _port;         // port of the server
        std::string _hostname;      // hostname of the server
    };
} // namespace GUI //
#endif /* !HANDLEARGS_HPP_ */
```

6.10 MockEgg.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** MockEgg
*/

#ifndef MockEgg_HPP_
    #define MockEgg_HPP_

    #include <iostream>
    #include "objects/abstracts/AObject.hpp"
    #include "position/Position.hpp"

namespace GUI {
    class MockEgg : public AObject {
    public:
        MockEgg(std::string teamName, std::string ownerID = "-1",
            Position tile = Position(0, 0));
        ~MockEgg();

        [[nodiscard]] std::string getTeam() const;

        [[nodiscard]] std::string getOwnerID() const;

        [[nodiscard]] bool getIsEjected() const;

        void setIsEjected(bool isEjected);

        void setPosition(int x, int y);

        [[nodiscard]] Position& getPosition() noexcept;

    protected:
        std::string _teamName;      // team name
        std::string _ownerID;               // ownerID (player who laid it) default -1
        bool _isEjected;            // is MockEgg ejected
        Position _position;         // position of the MockEgg
    };
} // namespace GUI //
#endif /* !MockEgg_HPP_ */
```

6.11 MockTeams.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** MockTeams
*/

#ifndef MOCKTEAMS_
    #define MOCKTEAMS_

    #include <iostream>
    #include <list>
    #include "MockTrantorians.hpp"

namespace GUI {
    class MockTrantorians; // Forward declaration

    class MockTeams {
        public:
            MockTeams(std::string name = "");
            ~MockTeams() = default;

            void setName(std::string name);

            [[nodiscard]] std::string getName() const;

            void addMockTrantorians(GUI::MockTrantorians& MockTrantorians);

            [[nodiscard]] std::list<GUI::MockTrantorians> getMockTrantoriansList() const;

            static void addMockTeamToMockTeamsList(const GUI::MockTeams& team);

            [[nodiscard]] static std::list<GUI::MockTeams>& getMockTeamsList();

            [[nodiscard]] static MockTeams* getTeamByName(const std::string& name);

            [[nodiscard]] MockTrantorians getMockTrantoriansById(std::string id);

        private:
            std::string _name;
            std::list<GUI::MockTrantorians> _MockTrantorians;

            // Static member to hold all teams
            static std::list<GUI::MockTeams> _Mockteams;
    };
} // namespace GUI

#endif // TEAMS_
```

6.12 MockTrantorians.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** MockTrantorians
*/

#ifndef MOCKTRANTORIAN_
    #define MOCKTRANTORIAN_

    #include <iostream>
    #include <utility>
    #include <vector>
    #include <list>
    #include <raylib.h>
    #include "position/Position.hpp"
    #include "objects/interface/IObject.hpp"
    #include "objects/interface/IObject.hpp"
    #include "objects/food/Food.hpp"
    #include "objects/stones/Linemate.hpp"
    #include "objects/stones/Deraumere.hpp"
    #include "objects/stones/Sibur.hpp"
    #include "objects/stones/Mendiane.hpp"
    #include "objects/stones/Phiras.hpp"
    #include "objects/stones/Thystame.hpp"

namespace GUI {
    class MockTeams; // Forward declaration

    class MockTrantorians {
        public:
          enum ResourceType {
                FOOD,
                LINEMATE,
                DERAUMERE,
                SIBUR,
                MENDIANE,
                PHIRAS,
                THYSTAME,
                RESOURCE_COUNT
            };
            enum Orientation {
                NORTH = 1,
                EAST = 2,
                SOUTH = 3,
                WEST = 4
            };
            MockTrantorians(std::string id = "null", int x = 0, int y = 0, float orientation = NORTH,
                int level = 1, std::string teamName = "");
            ~MockTrantorians() = default;

            void setPosition(int x, int y);
            [[nodiscard]] GUI::Position getPosition();

            void setTeam(MockTeams* team);
            [[nodiscard]] MockTeams* getTeam();

            void setLifetime(int lifetime);
            [[nodiscard]] int getLifetime();

            void setIsAlive(bool isAlive);
            [[nodiscard]] bool getIsAlive();

            void setAction(bool Action);
            [[nodiscard]] bool getAction();

            void setId(std::string id);
            [[nodiscard]] std::string getId();

            void setLevel(int level);
            [[nodiscard]] int getLevel();

            [[nodiscard]]
            bool operator==(const MockTrantorians& other) const;

            void setOrientation(float orientation);

            [[nodiscard]] float getOrientation();

            IObject* createObjectByType(ResourceType type, Position pos);

            void setInventory(std::vector<std::string> inventory);

            void clearInventory();
            void addObject(IObject *object);
            void removeObject(IObject *object);
            [[nodiscard]] std::list<IObject *> getInventory() const;

        private:
            GUI::Position _position;
            MockTeams* _team;
            int _lifetimeRemaining;
            bool _alive;
            bool _action;
            std::string _id;
            int _level;
            float _orientation;
            std::list<IObject *> _inventory;
    };
} // namespace GUI  //
#endif // MockTrantorians_
```

6.13 Client.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Client
*/

#ifndef CLIENT_HPP_
    #define CLIENT_HPP_

    #include <memory>
    #include <iostream>
    #include "network/socket/TCPSocket.hpp"
    #include "exception/ClientException.hpp"

namespace network {
    class Client {
    public:
        Client(const std::string &hostname, unsigned int port);
        ~Client() = default;

        void handleConnection();

        void handleDisconnection();

        std::vector<std::string> readData();

        std::unique_ptr<ASocket> _socket;
    protected:
        unsigned int _port;
        std::string _hostname;
    };
} // namespace network //
#endif /* !CLIENT_HPP_ */
```

6.14 CommandBCT.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandBCT
*/

#ifndef COMMANDBCT_HPP_
    #define COMMANDBCT_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandBCT : public ICommand {
    public:
        CommandBCT() = default;
        ~CommandBCT() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDBCT_HPP_ */
```

6.15 CommandEBO.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandEBO
*/

#ifndef COMMANDEBO_HPP_
    #define COMMANDEBO_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandEBO : public ICommand {
    public:
        CommandEBO() = default;
        ~CommandEBO() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDEBO_HPP_ */
```

6.16 CommandEDI.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandEDI
*/

#ifndef COMMANDEDI_HPP_
    #define COMMANDEDI_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandEDI : public ICommand {
    public:
        CommandEDI() = default;
        ~CommandEDI() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDEDI_HPP_ */
```

6.17 CommandENW.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandENW
*/

#ifndef COMMANDENW_HPP_
    #define COMMANDENW_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandENW : public ICommand {
    public:
        CommandENW() = default;
        ~CommandENW() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDENW_HPP_ */
```

6.18 CommandFactory.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandFactory
*/

#ifndef COMMANDFACTORY_HPP_
    #define COMMANDFACTORY_HPP_

    #include "network/commands/bct/CommandBCT.hpp"
    #include "network/commands/tna/CommandTNA.hpp"
    #include "network/commands/pnw/CommandPNW.hpp"
    #include "network/commands/ppo/CommandPPO.hpp"
    #include "network/commands/plv/CommandPLV.hpp"
    #include "network/commands/msz/CommandMSZ.hpp"
    #include "network/commands/pin/CommandPIN.hpp"
    #include "network/commands/pex/CommandPEX.hpp"
    #include "network/commands/pbc/CommandPBC.hpp"
    #include "network/commands/pfk/CommandPFK.hpp"
    #include "network/commands/pie/CommandPIE.hpp"
    #include "network/commands/pic/CommandPIC.hpp"
    #include "network/commands/sst/CommandSST.hpp"
    #include "network/commands/sgt/CommandSGT.hpp"
    #include "network/commands/pdi/CommandPDI.hpp"
    #include "network/commands/pdr/CommandPDR.hpp"
    #include "network/commands/pgt/CommandPGT.hpp"
    #include "network/commands/enw/CommandENW.hpp"
    #include "network/commands/ebo/CommandEBO.hpp"
    #include "network/commands/edi/CommandEDI.hpp"
    #include "network/commands/seg/CommandSEG.hpp"
    #include "network/commands/smg/CommandSMG.hpp"
    #include "network/commands/suc/CommandSUC.hpp"
    #include "network/commands/sbp/CommandSBP.hpp"

    #include "network/commands/interface/ICommand.hpp"
    #include <unordered_map>
    #include <string>
    #include <memory>

namespace network {
    class CommandFactory : public ICommand {
    public:
        CommandFactory();
        ~CommandFactory() = default;

        std::unique_ptr<ICommand> createCommand(const std::string& commandName);

        void setCallback(const std::string& commandName,
            ICommand::Callback callback);

        void execute(std::istringstream& iss) override;

        std::unique_ptr<ICommand> clone() const override;

    private:
        std::unordered_map<std::string,
            std::unique_ptr<ICommand» _commands;   // Map of commands
        std::unordered_map<std::string,
            ICommand::Callback> _callbacks;         // Map of callbacks
    };
} // namespace network //
#endif /* !COMMANDFACTORY_HPP_ */
```

6.19 ICommand.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** ICommand
*/

#ifndef ICOMMAND_HPP_
    #define ICOMMAND_HPP_

    #include <functional>
    #include <string>
    #include <sstream>
    #include <memory>

namespace network {
    class ICommand {
    public:
        ICommand() noexcept = default;
        virtual ~ICommand() noexcept = default;

        virtual void execute(std::istringstream& iss) = 0;

        virtual std::unique_ptr<ICommand> clone() const = 0;

        using Callback = std::function<void(std::istringstream&)>;
        void setCallback(Callback callback) {
            _callback = std::move(callback);
        }
    protected:
        Callback _callback;     // Callback function
    };
} // namespace network //
#endif /* !ICOMMAND_HPP_ */
```

6.20 CommandMSZ.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandMSZ
*/

#ifndef COMMANDMSZ_HPP_
    #define COMMANDMSZ_HPP_

    #include "network/commands/interface/ICommand.hpp"
```

```

namespace network {
    class CommandMSZ : public ICommand {
    public:
        CommandMSZ() = default;
        ~CommandMSZ() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDMSZ_HPP_ */
```

6.21 CommandPBC.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPBC
*/

#ifndef COMMANDPBC_HPP_
    #define COMMANDPBC_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPBC : public ICommand {
    public:
        CommandPBC() = default;
        ~CommandPBC() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPBC_HPP_ */
```

6.22 CommandPDI.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPDI
*/

#ifndef COMMANDPDI_HPP_
    #define COMMANDPDI_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPDI : public ICommand {
    public:
        CommandPDI() = default;
        ~CommandPDI() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPDI_HPP_ */
```

6.23 CommandPDR.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPDR
*/

#ifndef COMMANDPDR_HPP_
    #define COMMANDPDR_HPP_

    #include "network/commands/interface/ICommand.hpp"

```

```
namespace network {
    class CommandPDR : public ICommand {
    public:
        CommandPDR() = default;
        ~CommandPDR() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPDR_HPP_ */
```

6.24 CommandPEX.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPEX
*/

#ifndef COMMANDPEX_HPP_
    #define COMMANDPEX_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPEX : public ICommand {
    public:
        CommandPEX() = default;
        ~CommandPEX() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
}   // namespace network //
#endif /* !COMMANDPEX_HPP_ */
```

6.25 CommandPFK.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPFK
*/

#ifndef COMMANDPFK_HPP_
    #define COMMANDPFK_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPFK : public ICommand {
    public:
        CommandPFK() = default;
        ~CommandPFK() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPFK_HPP_ */
```

6.26 CommandPGT.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPGT
*/

#ifndef COMMANDPGT_HPP_
    #define COMMANDPGT_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPGT : public ICommand {
    public:
        CommandPGT() = default;
        ~CommandPGT() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPGT_HPP_ */
```

6.27 CommandPIC.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPIC
*/

#ifndef COMMANDPIC_HPP_
    #define COMMANDPIC_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPIC : public ICommand {
    public:
        CommandPIC() = default;
        ~CommandPIC() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPIC_HPP_ */
```

6.28 CommandPIE.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPIE
*/

#ifndef COMMANDPIE_HPP_
    #define COMMANDPIE_HPP_

#include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPIE : public ICommand {
    public:
        CommandPIE() = default;
        ~CommandPIE() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPIE_HPP_ */
```

6.29 CommandPIN.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPIN
*/

#ifndef COMMANDPIN_HPP_
    #define COMMANDPIN_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPIN : public ICommand {
```

```
    public:
        CommandPIN() = default;
        ~CommandPIN() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPIN_HPP_ */
```

6.30 CommandPLV.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPLV
*/

#ifndef COMMANDPLV_HPP_
    #define COMMANDPLV_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPLV : public ICommand {
    public:
        CommandPLV() = default;
        ~CommandPLV() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPLV_HPP_ */
```

6.31 CommandPNW.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPNW
*/

#ifndef COMMANDPNW_HPP_
    #define COMMANDPNW_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPNW : public ICommand {
    public:
        CommandPNW() = default;
        ~CommandPNW() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPNW_HPP_ */
```

6.32 CommandPPO.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandPPO
*/

#ifndef COMMANDPPO_HPP_
    #define COMMANDPPO_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandPPO : public ICommand {
    public:
        CommandPPO() = default;
        ~CommandPPO() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDPPO_HPP_ */
```

6.33 CommandSBP.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandSBP
*/

#ifndef COMMANDSBP_HPP_
    #define COMMANDSBP_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandSBP : public ICommand {
    public:
        CommandSBP() = default;
        ~CommandSBP() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDSBP_HPP_ */
```

6.34 CommandSEG.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandSEG
*/

#ifndef COMMANDSEG_HPP_
    #define COMMANDSEG_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandSEG : public ICommand {
    public:
        CommandSEG() = default;
        ~CommandSEG() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDSEG_HPP_ */
```

6.35 CommandSGT.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandSGT
*/

#ifndef COMMANDSGT_HPP_
    #define COMMANDSGT_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandSGT : public ICommand {
    public:
        CommandSGT() = default;
        ~CommandSGT() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDSGT_HPP_ */
```

6.36 CommandSMG.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandSMG
*/

#ifndef COMMANDSMG_HPP_
    #define COMMANDSMG_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandSMG : public ICommand {
    public:
        CommandSMG() = default;
        ~CommandSMG() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDSMG_HPP_ */
```

6.37 CommandSST.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandSST
*/

#ifndef COMMANDSST_HPP_
    #define COMMANDSST_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandSST : public ICommand {
    public:
        CommandSST() = default;
        ~CommandSST() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
}
#endif /* !COMMANDSST_HPP_ */
```

6.38 CommandSUC.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandSUC
*/

#ifndef COMMANDSUC_HPP_
    #define COMMANDSUC_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandSUC : public ICommand {
    public:
        CommandSUC() = default;
        ~CommandSUC() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDSUC_HPP_ */
```

6.39 CommandTNA.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** CommandTNA
*/

#ifndef COMMANDTNA_HPP_
    #define COMMANDTNA_HPP_

    #include "network/commands/interface/ICommand.hpp"

namespace network {
    class CommandTNA : public ICommand {
    public:
        CommandTNA() = default;
        ~CommandTNA() = default;

        void execute(std::istringstream &iss) final;
        std::unique_ptr<ICommand> clone() const final;
    };
} // namespace network //
#endif /* !COMMANDTNA_HPP_ */
```

6.40 ProtocolHandler.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** ProtocolHandler
*/

#ifndef PROTOCOLHANDLER_HPP_
    #define PROTOCOLHANDLER_HPP_

    #include <vector>
    #include <string>
    #include <sstream>
    #include "network/commands/factory/CommandFactory.hpp"

namespace network {
    class ProtocolHandler {
    public:
        ProtocolHandler(CommandFactory& factory);
        ~ProtocolHandler() = default;

        void handleData(const std::vector<std::string> &data);
    private:
        CommandFactory& _factory;   // Command factory
    };
} // namespace network //
#endif /* !PROTOCOLHANDLER_HPP_ */
```

6.41 ASocket.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** ASocket
*/

#ifndef ASOCKET_HPP_
    #define ASOCKET_HPP_

    #include <iostream>
    #include <vector>

namespace network {
    class ASocket {
    public:
        ASocket() = default;

        virtual ~ASocket() noexcept = default;

        virtual void connect(const std::string &hostname,
            unsigned int port) = 0;

        virtual void close() = 0;

        virtual void send(const std::string &data) = 0;

        [[nodiscard]] virtual std::vector<std::string> receive() = 0;

        [[nodiscard]] virtual int getSockfd() const = 0;
    };
} // namespace network
#endif /* !ASOCKET_HPP_ */
```

6.42 TCPSocket.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** TCPSocket
*/

#ifndef TCPSOCKET_HPP_
    #define TCPSOCKET_HPP_

    #include <cstring>
    #include <iostream>
    #include <sys/socket.h>
    #include <netinet/in.h>
    #include <arpa/inet.h>
    #include <unistd.h>
    #include <vector>
    #include "network/socket/ASocket.hpp"
    #include "exception/SocketException.hpp"

namespace network {
    class TCPSocket : public ASocket {
    public:
        TCPSocket();

        ~TCPSocket();

        void connect(const std::string &hostname, unsigned int port) final;

        void close() final;

        void send(const std::string &data) final;

        [[nodiscard]] std::vector<std::string> receive() final;

        [[nodiscard]] int getSockfd() const;

    protected:
        int _sockfd;
        struct sockaddr_in serv_addr;
        std::string _partialBuffer; // Partial buffer
    };
} // namespace network //
#endif /* !TCPSOCKET_HPP_ */
```

6.43 AObject.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** AObject
*/

#ifndef AOBJECT_HPP_
    #define AOBJECT_HPP_

    #include "objects/interface/IObject.hpp"
    #include <string>

```

```
namespace GUI {
    class AObject : public IObject {
    public:
        AObject(unsigned int quantity, Position tile);
        virtual ~AObject() noexcept = default;

        [[nodiscard]] unsigned int getQuantity() const noexcept final;

        [[nodiscard]] const Position& getPosition() const noexcept final;

        [[nodiscard]] Position& getPosition() noexcept;

        [[nodiscard]] unsigned int getType() noexcept;

        virtual void setQuantity(unsigned int quantity) noexcept final;

        [[nodiscard]] virtual std::string getName() const noexcept;

    protected:
        Position _position;      // tile where the object is (x, y)
        unsigned int _quantity;  // quantity of the object
    };
}  // namespace GUI //
#endif /* !AOBJECT_HPP_ */
```

6.44 AStone.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** AStone
*/

#ifndef ASTONE_HPP_
    #define ASTONE_HPP_

    #include "AObject.hpp"

namespace GUI {
    class AStone : public AObject {
    public:
        AStone(unsigned int quantity, Position tile);
        virtual ~AStone() noexcept = default;
    };
} // namespace GUI
#endif /* !ASTONE_HPP_ */
```

6.45 Egg.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Egg
*/

#ifndef EGG_HPP_
    #define EGG_HPP_

    #include <iostream>
    #include "objects/abstracts/AObject.hpp"
    #include "position/Position.hpp"
    #include <raylib.h>

namespace GUI {
    class Egg : public AObject {
    public:
        Egg(std::string teamName, std::string ownerID = "-1",
            Position tile = Position(0, 0));
        ~Egg();

        [[nodiscard]] std::string getTeam() const;

        [[nodiscard]] std::string getOwnerID() const;

        [[nodiscard]] bool getIsEjected() const;

        void setIsEjected(bool isEjected);

        void setPosition(int x, int y);

        [[nodiscard]] Position& getPosition() noexcept;

        [[nodiscard]] Model getModel();

    protected:
        std::string _teamName;      // team name
        std::string _ownerID;               // ownerID (player who laid it) default -1
        bool _isEjected;            // is egg ejected
        Position _position;         // position of the egg
        Model _eggModel;            // 3d model for the egg
        Texture _eggTexture;        // 3d texture for the egg
    };
} // namespace GUI //
#endif /* !EGG_HPP_ */
```

6.46 Food.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Food
*/

#ifndef FOOD_HPP_
    #define FOOD_HPP_

    #include "objects/abstracts/AObject.hpp"

namespace GUI {
    class Food : public AObject {
    public:
        Food(Position tile = Position(0, 0));
        virtual ~Food() noexcept = default;

        [[nodiscard]] unsigned int getType() noexcept final;
        [[nodiscard]] std::string getName() const noexcept final;
    };
} // namespace GUI //
#endif /* !FOOD_HPP_ */
```

6.47 IObject.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** IObject
*/

#ifndef IOBJECT_HPP_
    #define IOBJECT_HPP_

    #include "position/Position.hpp"
    #include <string>

namespace GUI {
    class IObject {
    public:
        IObject() noexcept = default;
        virtual ~IObject() noexcept = default;

        [[nodiscard]] virtual unsigned int getQuantity() const noexcept = 0;

        [[nodiscard]] virtual const Position& getPosition() const noexcept = 0;

        [[nodiscard]] virtual Position& getPosition() noexcept = 0;


        [[nodiscard]] virtual unsigned int getType() noexcept = 0;

        virtual void setQuantity(unsigned int quantity) noexcept = 0;

        [[nodiscard]] virtual std::string getName() const noexcept = 0;
    };
}  // namespace GUI
#endif /* !IOBJECT_HPP_ */
```

6.48 Deraumere.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Deraumere
*/

#ifndef DERAUMERE_HPP_
    #define DERAUMERE_HPP_

    #include "objects/abstracts/AStone.hpp"

namespace GUI {
    class Deraumere : public AStone {
    public:
        Deraumere(Position tile = Position(0, 0));
        virtual ~Deraumere() noexcept = default;

        [[nodiscard]] unsigned int getType() noexcept final;

        [[nodiscard]] std::string getName() const noexcept final;
    };
} // namespace GUI //
#endif /* !DERAUMERE_HPP_ */
```

6.49 Linemate.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Linemate
*/

#ifndef LINEMATE_HPP_
    #define LINEMATE_HPP_

    #include "objects/abstracts/AStone.hpp"

namespace GUI {
    class Linemate : public AStone{
    public:
        Linemate(Position tile = Position(0, 0));
        virtual ~Linemate() noexcept = default;

        [[nodiscard]] unsigned int getType() noexcept final;

        [[nodiscard]] std::string getName() const noexcept final;
    };
} // namespace GUI //
#endif /* !LINEMATE_HPP_ */
```

6.50 Mendiane.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Mendiane
*/

#ifndef MENDIANE_HPP_
    #define MENDIANE_HPP_

    #include "objects/abstracts/AStone.hpp"

namespace GUI {
    class Mendiane : public AStone {
    public:
        Mendiane(Position tile = Position(0, 0));
        virtual ~Mendiane() noexcept = default;

        [[nodiscard]] unsigned int getType() noexcept final;

        [[nodiscard]] std::string getName() const noexcept final;
    };
} // namespace GUI //
#endif /* !MENDIANE_HPP_ */
```

6.51 Phiras.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Phiras
*/

#ifndef PHIRAS_HPP_
    #define PHIRAS_HPP_

    #include "objects/abstracts/AStone.hpp"

namespace GUI {
    class Phiras : public AStone {
    public:
        Phiras(Position tile = Position(0, 0));
        virtual ~Phiras() noexcept = default;

        [[nodiscard]] unsigned int getType() noexcept final;

        [[nodiscard]] std::string getName() const noexcept final;
    };
} // namespace GUI //
#endif /* !PHIRAS_HPP_ */
```

6.52 Sibur.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Sibur
*/

#ifndef SIBUR_HPP_
    #define SIBUR_HPP_

    #include "objects/abstracts/AStone.hpp"

namespace GUI {
    class Sibur : public AStone {
    public:
        Sibur(Position tile = Position(0, 0));
        virtual ~Sibur() noexcept = default;

        [[nodiscard]] unsigned int getType() noexcept final;

        [[nodiscard]] std::string getName() const noexcept final;
    };
} // namespace GUI //
#endif /* !SIBUR_HPP_ */
```

6.53 Thystame.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Thystame
*/

#ifndef THYSTAME_HPP_
    #define THYSTAME_HPP_

    #include "objects/abstracts/AStone.hpp"

namespace GUI {
    class Thystame : public AStone {
    public:
        Thystame(Position tile = Position(0, 0));
        virtual ~Thystame() noexcept = default;

        [[nodiscard]] unsigned int getType() noexcept final;

        [[nodiscard]] std::string getName() const noexcept final;
    };
} // namespace GUI //
#endif /* !THYSTAME_HPP_ */
```

6.54 Position.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Position
*/

#ifndef POSITION_HPP_
    #define POSITION_HPP_

namespace GUI {
    class Position {
    public:
        Position(unsigned int x = 0, unsigned int y = 0);
        ~Position() = default;

        [[nodiscard]] unsigned int getX() const noexcept;
        [[nodiscard]] unsigned int getY() const noexcept;

        void setX(unsigned int x) noexcept;
        void setY(unsigned int y) noexcept;

    protected:
        unsigned int _x;    //position x
        unsigned int _y;    //position y
    private:
    };
} // namespace GUI //
#endif /* !POSITION_HPP_ */
```

6.55 Tile.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Tile
*/

#ifndef TILE_HPP_
    #define TILE_HPP_

    #include <list>
    #include <string>
    #include <sstream>
    #include <vector>
    #include <iostream>
    #include <algorithm>
    #include <unordered_map>
    #include "position/Position.hpp"
    #include "objects/interface/IObject.hpp"
    #include "objects/food/Food.hpp"
    #include "objects/stones/Linemate.hpp"
    #include "objects/stones/Deraumere.hpp"
    #include "objects/stones/Sibur.hpp"
    #include "objects/stones/Mendiane.hpp"
    #include "objects/stones/Phiras.hpp"
    #include "objects/stones/Thystame.hpp"
    #include <raylib.h>

namespace GUI {
    class Tile {
    public:
        enum ResourceType {
            FOOD,
            LINEMATE,
            DERAUMERE,
            SIBUR,
            MENDIANE,
            PHIRAS,
            THYSTAME,
            RESOURCE_COUNT
        };
        Tile(Position pos);
        ~Tile() = default;
        [[nodiscard]] Position getPosition() const;

        void addObject(IObject *object);

        void removeObject(IObject *object);

        [[nodiscard]] std::list<IObject *> getObjects() const;

        void clearObjects();

        void updateTileContent(const std::vector<std::string>& tileContent);
        [[nodiscard]] IObject* createObjectByType(ResourceType type, Position pos);

        [[nodiscard]] BoundingBox getBounds() const;

    protected:
        std::list<IObject *> _objects;  // list of objects on the tile
        Position _pos;                  // position of the tile
    };
} // namespace GUI
#endif // TILE_HPP_
```

6.56 Actions.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Actions
*/

#ifndef ACTIONS_HPP_
    #define ACTIONS_HPP_

    #include <stdbool.h>
    #include <vector>
    #include <iostream>
    #include <map>

namespace GUI {
    enum action {
        Forward,
        Right,
        Left,
        Look,
        Inventory,
        Broadcast,
        Connect,
        Fork,
        Eject,
        Take,
        Set,
        Incantation
    };

    class Actions {
        public:
            Actions() = default;
            ~Actions() = default;

            [[nodiscard]] bool moveForward();
            [[nodiscard]] bool turnRight();
            [[nodiscard]] bool turnLeft();

            [[nodiscard]] std::vector<std::vector<int» lookAround();
            [[nodiscard]] std::map<std::string, int> openInventory();
            [[nodiscard]] bool broadcastText(std::string text);
            [[nodiscard]] int connectNbr();
            [[nodiscard]] bool forkPlayer();
            void playerDead();

            [[nodiscard]] bool takeObj();
            [[nodiscard]] bool setObj();
            [[nodiscard]] bool startIncantation();
    };
} // namespace GUI

#endif // ACTIONS_HPP_
```

6.57 Teams.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Teams
*/

#ifndef TEAMS_
```

```
    #define TEAMS_

    #include <iostream>
    #include <list>
    #include "Trantorian.hpp"
    #include "objects/eggs/Egg.hpp"

namespace GUI {
    class Teams {
        public:
            Teams(std::string name = "");
            ~Teams() = default;

            void setName(std::string name);

            [[nodiscard]] std::string getName() const;

            void addTrantorian(Trantorian& trantorian);

            [[nodiscard]] std::list<Trantorian> getTrantorianList() const;

            static void addTeamToTeamsList(const GUI::Teams& team);

            [[nodiscard]] static std::list<GUI::Teams>& getTeamsList();

            [[nodiscard]] static Teams* getTeamByName(const std::string& name);

            [[nodiscard]] Trantorian getTrantorianById(std::string id);

            bool hasTrantorian(const std::string& id) const;

            void addEggToList(Egg newEgg);

            [[nodiscard]] std::list<Egg> getEggList() const;

            Trantorian* getTrantorianByIdMod(const std::string& id);
        private:
            std::string _name;
            std::list<Trantorian> _trantorians;
            std::list<Egg> _eggs;
            static std::list<GUI::Teams> _teams;
    };
} // namespace GUI

#endif // TEAMS_
```

6.58 Trantorian.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Trantorian
*/

#ifndef TRANTORIAN_
    #define TRANTORIAN_

    #include <iostream>
    #include <utility>
    #include <vector>
    #include <list>
    #include <raylib.h>
    #include "position/Position.hpp"
    #include "objects/interface/IObject.hpp"
    #include "objects/interface/IObject.hpp"
    #include "objects/food/Food.hpp"
    #include "objects/stones/Linemate.hpp"
    #include "objects/stones/Deraumere.hpp"
    #include "objects/stones/Sibur.hpp"
    #include "objects/stones/Mendiane.hpp"
    #include "objects/stones/Phiras.hpp"
    #include "objects/stones/Thystame.hpp"

namespace GUI {
    class Trantorian {
        public:
          enum ResourceType {
                FOOD,
                LINEMATE,
                DERAUMERE,
                SIBUR,
                MENDIANE,
                PHIRAS,
                THYSTAME,
                RESOURCE_COUNT
            };
            enum Orientation {
                NORTH = 1,
                EAST = 2,
                SOUTH = 3,
                WEST = 4
            };
            Trantorian(std::string id = "null", int x = 0, int y = 0, float orientation = NORTH,
                int level = 1);
            ~Trantorian() = default;

            void setPosition(int x, int y);
            [[nodiscard]] const GUI::Position& getPosition() const;

            void setLifetime(int lifetime);
            [[nodiscard]] int getLifetime();

            void setIsAlive(bool isAlive);
            [[nodiscard]] bool getIsAlive();

            void setAction(bool Action);
            [[nodiscard]] bool getAction();

            void setId(std::string id);
            [[nodiscard]] std::string getId() const;

            void setLevel(int level);
            [[nodiscard]] int getLevel();

            [[nodiscard]]
            bool operator==(const Trantorian& other) const;

            void setOrientation(float orientation);

            [[nodiscard]] float getOrientation();

            IObject* createObjectByType(ResourceType type, Position pos);

            void setInventory(std::vector<std::string> inventory);

            void clearInventory();
            void addObject(IObject *object);
            void removeObject(IObject *object);
            [[nodiscard]] std::list<IObject *> getInventory() const;

            [[nodiscard]] Model getModel() const;
            [[nodiscard]] Texture2D getTexture() const;

        private:
            GUI::Position _position;
            int _lifetimeRemaining;
            bool _alive;
            bool _action;
            std::string _id;
            int _level;
            float _orientation;
            std::list<IObject *> _inventory;
            Model _trantorianModel;
            Texture2D _trantorianTexture;
    };
} // namespace GUI  //
#endif // TRANTORIAN_
```

6.59 World.hpp

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** World
*/

#ifndef WORLD_HPP_
    #define WORLD_HPP_

    #include <vector>
    #include "tiles/Tile.hpp"

namespace GUI {
    class World {
    public:
        World(unsigned int width = 0, unsigned int height = 0);

        ~World() = default;

        void setWidth(unsigned int width);

        void setHeight(unsigned int height);

        [[nodiscard]] unsigned int getWidth() const;

        [[nodiscard]] unsigned int getHeight() const;

        void addObject(IObject *object, Position pos);

        void removeObject(IObject *object, Position pos);

        [[nodiscard]] std::list<IObject *> getObjects() const;

        [[nodiscard]] std::list<IObject *> getObjectsAt(Position tile) const;

        void setWorldSize(unsigned int width, unsigned int height);

        [[nodiscard]] std::vector<std::vector<Tile»& getTiles();

        [[nodiscard]] Tile& getTileAt(unsigned int x, unsigned int y);

        void addTile(Tile tile);

    protected:
        unsigned int _width;  // width of the world
        unsigned int _height; // height of the world
        std::vector<std::vector<Tile» _tiles;  // 2D vector of tiles
    };
} // namespace GUI
#endif /* !WORLD_HPP_ */
```