Arcade architecture

Generated by Doxygen 1.10.0

Chapter 1

Description

The goal of this project is to create a network game where several teams confront each other on a tile map containing resources. The winning team is the first one where at least 6 players reach the maximum elevation.

1.1 Documentation :

1.1.1 Docusorus :

To start the docusarus documentation : cd documentation/my-website npx docusaurus start

1.1.2 Doxygen :

The basic documentation fo the project is generated using the doxygen, to run the doxygen executable, please make sure you installed the pdf-latex librairie. To generate the PDF : ./generateDoc.sh

1.2 Commit norm :

[Gitmoji] : [Element / Module] : [MESSAGE]

Gitmoji = The emoji approriate for the current modification. [Element / Module] = The elemenet you applied the modification. [MESSAGE] = A detail message of what you did.

Gitmojies:

```
Code feature :
    - :sparkles: (): Introduce new features
    - :recycle: (): Refactor / update code
    - :bug: (): Fix a bug
    - :poop: () : Remove Coding style or temporary fix
    - :rotating_light: () : Fix Compiling Warning
    - :fire: (): Remove code or files

Test feature :
    - :white_check_mark: (): Add, update, or pass tests

Architecture :
    - :see_no_evil: (): Add or update .gitignore files
    - :construction_worker: (): Add or update CI build system
    - :building_construction: () : Make Architectural changes
    - :memo: () : Add or update documentation
```

1.2.1 Pull Request

- :tada: (): This Gitmoji must be used for each PR created!
- :lipstick: (): This Gitmoji must be used for each PR merged!
- :rewind: (): This Gitmoji must be used for each revert done!

1.3 Git-Cli :

- Changer message de commit, avant qu'il soit push :
  ```
  git commit --amend -m "New commit message"
  ```
- Changer le message de commit, si il a deja été push :
  ```
  git commit --amend -m "New commit message"
  git push --force
  ```
- Un-add un ficher add par erreur qui est pas encore push:
  ```
  git restore --staged <file>
  ```
- Un-add un fichier qui a été commit :
  ```
  git reset --soft HEAD~1
  git restore --staged fichier-a-retirer.txt
  git commit -m "Nouveau message de commit (sans le fichier)"
  ```

Chapter 2

Zappy Server

A server, created in C, that generates the inhabitants' world.

2.1 Usage

```
USAGE: ./zappy_server -p port -x width -y height -n name1 name2 ... -c clientsNb -f freq --auto-start on|off
      --display-eggs true|false [-v | --verbose]
    port        is the port number
    width       is the width of the world
    height      is the height of the world
    nameX       is the name of the team X
    clientsNb   is the number of authorized clients per team
    freq        is the reciprocal of time unit for execution of actions
    auto-start  does the greeting is send automaticly #(see bonus part)
    display-eggs    eggs are visible and destructible
```

The server is executed in the form of one, single process and one, single thread. It must use select to handle socket multiplexing; the select must unlock only if something happen on a socket or if an event is ready to be executed.

The team name GRAPHIC is reserved for the GUI to authenticate itself as such to the server.

2.2 AI protocol

Each player responds to the following actions and only to these ones, with the following syntax :

Action	Command	Time limit	Response
move up one tile	**Forward**	7/f	ok
turn 90° right	**Right**	7/f	ok
turn 90° left	**Left**	7/f	ok
look around	**Look**	7/f	[tile1, tile2,...]
inventory	**Inventory**	1/f	[linemate n, sibur n, ...]
broadcast text	**Broadcast text**	7/f	ok
number of team unused slots	**Connect_nbr**	-	value
fork a player	**Fork**	42/f	ok
eject players from this tile	**Eject**	7/f	ok/ko
death of a player	-	-	dead
take object	**Take object**	7/f	ok/ko
set object down	**Set object**	7/f	ok/ko

| start incantation | **Incantation** | 300/f | Elevation underway | Current level: k/ko |

> In case of a bad/unknown command, the server must answer “ko”.

The AI client’s connection to the server happens as follows:

```
1. the client opens a socket on the server’s port,

2. the server and the client communicate the following way:
    Server --> WELCOME\n
           <-- TEAM-NAME\n
           --> game informations (see tha above array)
```

X and Y indicate the world’s dimensions.

CLIENT-NUM indicates the number of slots available on the server for the TEAM-NAME team. If this number is greater than or equal to 1, a new client can connect.

> The client can send up to 10 requests in a row without any response from the server. Over 10, the server will drop the incomming commands.

The server executes the client’s requests in the order they were received.

The requests are buffered and a command’s execution time only blocks the player in question.

Trantorians have adopted an international time unit. The time unit is seconds.

An action’s execution time is calculated with the following formula:

action / f

Where f is an integer representing the reciprocal (multiplicative inverse) of time unit.

For instance, if f=1, “forward” takes 7 / 1 = 7 seconds.

By default f=100.

2.3 GUI protocol

SYMBOL	MEANING
X	width or horizontal position
Y	height or vertical position
q0	resource 0 (food) quantity
q1	resource 1 (linemate) quantity
q2	resource 2 (deraumere) quantity
q3	resource 3 (sibur) quantity
q4	resource 4 (mendiane) quantity
q5	resource 5 (phiras) quantity
q6	resource 6 (thystame) quantity
n	player number
O	orientation: 1(N), 2(E), 3(S), 4(W)
L	player or incantation level
e	egg number
T	time unit

SERVER	CLIENT	DETAILS	TO A GUI client	TO ALL GUI client
msz X Y	msz	map size	new GUI client connection or msz command	
bct X Y q0 q1 q2 q3 q4 q5 q6	bct X Y	content of a tile	bct command	
bct X Y q0 q1 q2 q3 q4 q5 q6 ∗ nbr_tiles	mct	content of the map (all the tiles)	new GUI client connection or mct command or map refill	
tna N ∗ nbr_teams	tna	name of all the teams	new GUI client connection	
pnw #n X Y O L N		connection of a new player	new GUI client connection	new AI client connection
ppo #n X Y O	ppo #n	player’s position	ppo command	AI left, right forward action or AI is ejected
plv #n L	plv #n	player’s level	new GUI client connection or plv command	AI sucessfully incantate
pin #n X Y q0 q1 q2 q3 q4 q5 q6	pin #n	player’s inventory	new GUI client connection or pin command	new AI client connection or AI set, take action or AI lost food
pex #n		expulsion		AI eject action
pbc #n M		broadcast		AI broadcast action
pic X Y L #n #n ...		start of an incantation (by the first player)		AI incantation action
pie X Y R		end of an incantation		AI incatation end
pfk #n		egg laying by the player		AI fork action
pdr #n i		resource dropping		AI set action
pgt #n i		resource collecting		AI take action
pdi #n		death of a player		AI client disconnection or AI lost all it's food
enw #e #n X Y		an egg was laid by a player	new GUI client connection	AI fork action end (after 42/f)
ebo #e		player connection for an egg		new AI client connection
edi #e		death of an egg		egg is ejected by an AI
sgt T	sgt	time unit request	new GUI client connection or sgt	sst command
sst T	sst T	time unit modification		
seg N		end of game		an AI team reach the victory conditions
smg M		message from the server		server send a message

SERVER	CLIENT	DETAILS	TO A GUI client	TO ALL GUI client
suc		unknown command		empty or unknown command
sbp		command parameter		invalid command (wrong parameter.s)

The GUI client's connection to the server happens as follows:

```
1. the client opens a socket on the server's port,

2. the server and the client communicate the following way:
    Server --> WELCOME\n
           <-- GRAPHIC\n
           --> game informations (see the above array)
```

2.4 Informations

2.4.1 Incantations

This ritual, which augments physical and mental capacities, must be done according to a particular rite: they must gather the following on the same unit of terrain:

- At least a certain number of each stones
- At least a certain number of players with the same level

The elevation begins as soon as a player initiates the incantation. The player who starts an incantation will receive ko if all the requirements are not satisfied and the incantation will be canceled, the player will receive the ko instantly after the initial server check (not at the end of the incantation duration).

It is not necessary for the players to be on the same team; they only need to be of the same level. Every player with the corresponding level and present at the beginning and at the end of the incantation attain the higher level.

During the incantation, the participants can not make any action until the end of the rite.

At the end of the incantation, the exact quantity of resources needed by the rite are consumed.

Command	Effect

2.5 Bonus

2.5.1 Server commands

The server accepts command in its standard input.

Command	Effect
/clients	list all connected clients
/quit	stop the server
/send_ais msg	send messages to all AI
/send_guis msg	send messages to all GUI
/map	display map informations
/clear	clear the shell
/pause	pause the AI's actions
/start	start the server
/setTile ressource quantity x y	set the given ressource quantity of a tile
/tile x y	get the inventory of a tile
/tp id x y	tp an AI by it's id
/kill id	kill an AI by it's id
/noFood true or false	disable the food management
/broadcast "message" x y	simulate a broadcast
/setLevel id level	set the level of an AI by it's id
/setInventory id ressource quantity	set the given ressource quantity inside an AI inventory by it's id
/setClientsNb nb	set the minimum number of AI per team
/setFreq freq	set the frequency of the server
/noRefill true or false	disable the map refill
/fork team x y	simulate a fork for the given team at the given position
/incantate x y	simulate an incantation of the given level at the given position

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

File Index

5.1 File List

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

Chapter 6

Class Documentation

6.1 CLI Class Reference

Public Member Functions

- **CLI** (int ac, const char ∗const ∗av)
- zappy::structs::Config **parseArguments** (int ac, const char ∗const ∗av) const

Private Member Functions

- bool **hasCorrectNumberOfArguments** (int ac) const
- int **parsePort** (const char ∗portStr) const
- std::string **parseHostname** (const char ∗hostnameStr) const
- void **validateConfig** (bool portFound, bool hostFound) const

Private Attributes

- int **_ac**
- const char ∗const ∗ **_av**

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

- gui/src/CLI/CLI.hpp
- gui/src/CLI/CLI.cpp

6.2 CLI.CLI Class Reference

Public Member Functions

- **__init__** (self)
- **parse_args** (self, args)
- **parse_port** (self, port_str)
- **parse_name** (self, name)
- **parse_machine** (self, machine_str)
- **validate_config** (self, port_found, name_found, machine_found)

Public Attributes

- **port**
- **name**
- **machine**

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

- ai/src/CLI/CLI.py

6.3 Client Class Reference

Public Member Functions

- **Client** (int ac, const char ∗const ∗av)

Private Member Functions

- void **initialize** (int ac, const char ∗const ∗av)

Private Attributes

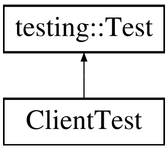
- zappy::structs::Config **_config**
- std::unique_ptr< Communication > **_communication**

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

- gui/src/Client/Client.hpp
- gui/src/Client/Client.cpp

6.4 ClientTest Class Reference

Inheritance diagram for ClientTest:



Protected Member Functions

- void **SetUp** () override
- void **TearDown** () override
- char ∗∗ **createArgv** (const std::vector< std::string > &args)
- void **cleanupArgv** (char ∗∗argv, int argc)

Protected Attributes

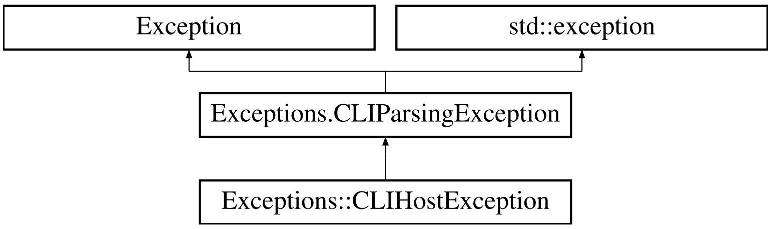
- std::stringstream **buffer**
- std::streambuf ∗ **originalCout**

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

- gui/tests/unit/Client/Client_test.cpp

6.5 Exceptions::CLIHostException Class Reference

Inheritance diagram for Exceptions::CLIHostException:



Public Member Functions

- **CLIHostException** (const std::string &message)

Public Member Functions inherited from Exceptions.CLIParsingException

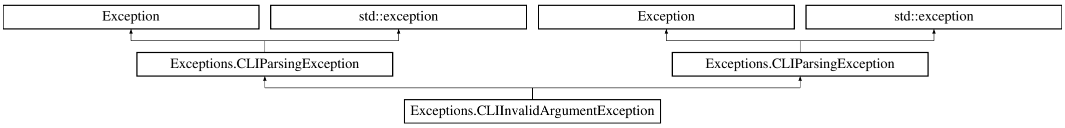
- **__init__** (self, str message)
- **CLIParsingException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

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

- gui/src/Exceptions/Exceptions.hpp

6.6 Exceptions.CLIInvalidArgumentException Class Reference

Inheritance diagram for Exceptions.CLIInvalidArgumentException:


Exception
std::exception
Exception
std::exception
Exceptions.CLIParsingException
Exceptions.CLIParsingException
Exceptions.CLIInvalidArgumentException

Public Member Functions

- __init__ (self, str message)
- **CLIInvalidArgumentException** (const std::string &message)

Public Member Functions inherited from Exceptions.CLIParsingException

- **CLIParsingException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

6.6.1 Constructor & Destructor Documentation

6.6.1.1 __init__()

```
Exceptions.CLIInvalidArgumentException.__init__ (
             self,
            str message )
```

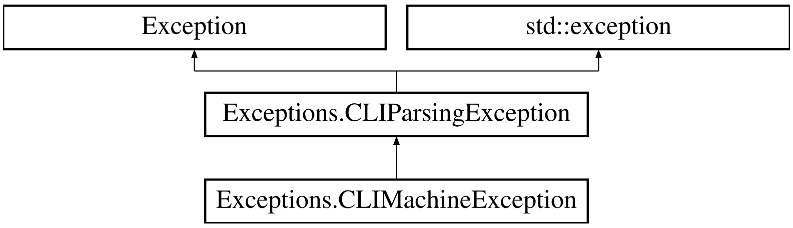
Reimplemented from Exceptions.CLIParsingException.

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

- ai/src/Exceptions/Exceptions.py
- gui/src/Exceptions/Exceptions.hpp

6.7 Exceptions.CLIMachineException Class Reference

Inheritance diagram for Exceptions.CLIMachineException:


Exception
std::exception
Exceptions.CLIParsingException
Exceptions.CLIMachineException

Public Member Functions

- __init__ (self, str message)

Public Member Functions inherited from Exceptions.CLIParsingException

- **CLIParsingException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

6.7.1 Constructor & Destructor Documentation

6.7.1.1 __init__()

```
Exceptions.CLIMachineException.__init__ (
              self,
             str message )
```

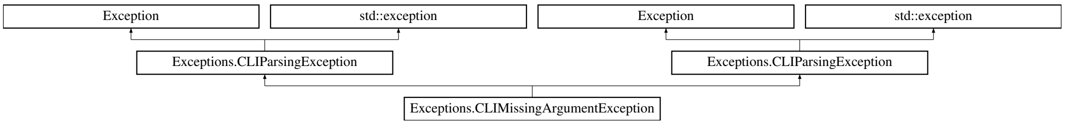
Reimplemented from Exceptions.CLIParsingException.

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

- ai/src/Exceptions/Exceptions.py

6.8 Exceptions.CLIMissingArgumentException Class Reference

Inheritance diagram for Exceptions.CLIMissingArgumentException:


Exception
std::exception
Exception
std::exception
Exceptions.CLIParsingException
Exceptions.CLIParsingException
Exceptions.CLIMissingArgumentException

Public Member Functions

- __init__ (self, str message)
- **CLIMissingArgumentException** (const std::string &message)

Public Member Functions inherited from Exceptions.CLIParsingException

- **CLIParsingException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

6.8.1 Constructor & Destructor Documentation

6.8.1.1 __init__()

```
Exceptions.CLIMissingArgumentException.__init__ (
              self,
             str message )
```

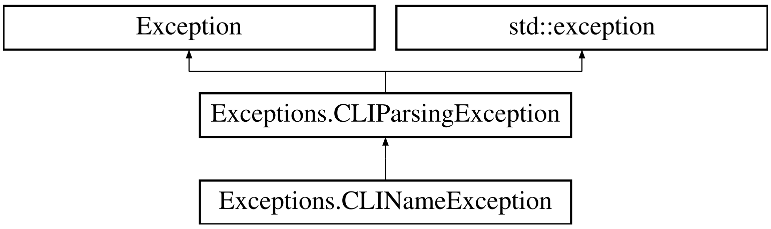
Reimplemented from Exceptions.CLIParsingException.

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

- ai/src/Exceptions/Exceptions.py
- gui/src/Exceptions/Exceptions.hpp

6.9 Exceptions.CLINameException Class Reference

Inheritance diagram for Exceptions.CLINameException:



Public Member Functions

- __init__ (self, str message)

Public Member Functions inherited from Exceptions.CLIParsingException

- **CLIParsingException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

6.9.1 Constructor & Destructor Documentation

6.9.1.1 __init__()

```
Exceptions.CLINameException.__init__ (
                self,
               str message )
```

Reimplemented from Exceptions.CLIParsingException.

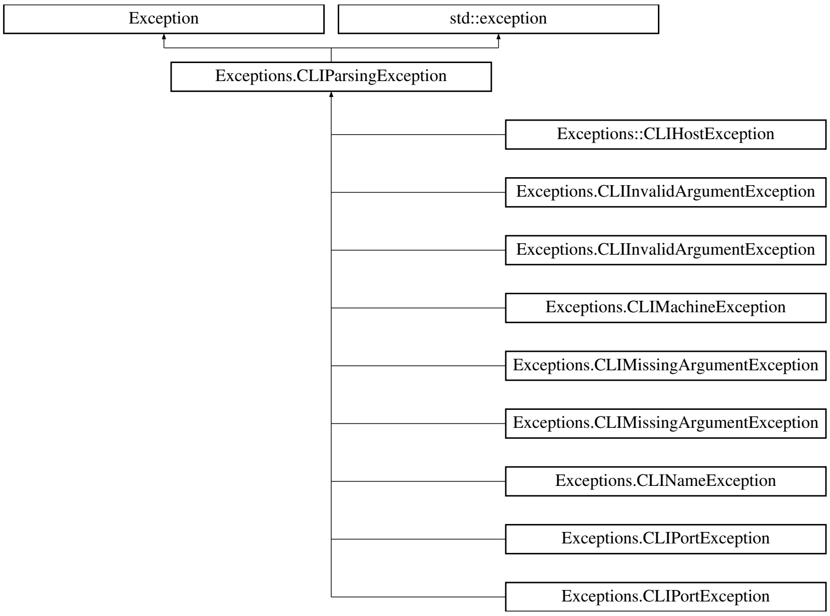
The documentation for this class was generated from the following file:

- ai/src/Exceptions/Exceptions.py

6.10 Exceptions.CLIParsingException Class Reference

EPITECH PROJECT, 2025 B-YEP-400-NAN-4-1-zappy-albane.merian File description: Exceptions.

Inheritance diagram for Exceptions.CLIParsingException:


Exception
std::exception
Exceptions.CLIParsingException
Exceptions::CLIHostException
Exceptions.CLIInvalidArgumentException
Exceptions.CLIInvalidArgumentException
Exceptions.CLIMachineException
Exceptions.CLIMissingArgumentException
Exceptions.CLIMissingArgumentException
Exceptions.CLINameException
Exceptions.CLIPortException
Exceptions.CLIPortException

Public Member Functions

- **__init__** (self, str message)
- **CLIParsingException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

Private Attributes

- std::string **_message**

6.10.1 Detailed Description

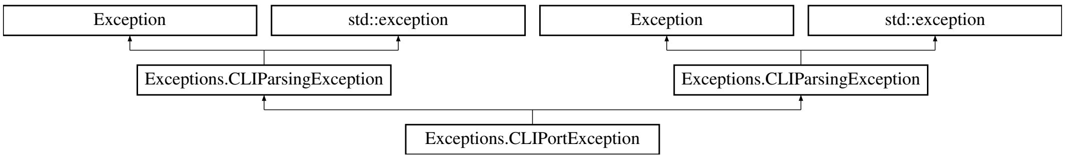
EPITECH PROJECT, 2025 B-YEP-400-NAN-4-1-zappy-albane.merian File description: Exceptions.

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

- ai/src/Exceptions/Exceptions.py
- gui/src/Exceptions/Exceptions.hpp

6.11 Exceptions.CLIPortException Class Reference

Inheritance diagram for Exceptions.CLIPortException:


Exception
std::exception
Exception
std::exception
Exceptions.CLIParsingException
Exceptions.CLIParsingException
Exceptions.CLIPortException

Public Member Functions

- __init__ (self, str message)
- **CLIPortException** (const std::string &message)

Public Member Functions inherited from Exceptions.CLIParsingException

- **CLIParsingException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

6.11.1 Constructor & Destructor Documentation

6.11.1.1 __init__()

```
Exceptions.CLIPortException.__init__ (
                self,
               str message )
```

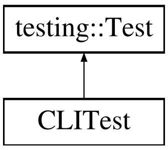
Reimplemented from Exceptions.CLIParsingException.

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

- ai/src/Exceptions/Exceptions.py
- gui/src/Exceptions/Exceptions.hpp

6.12 CLITest Class Reference

Inheritance diagram for CLITest:



Protected Member Functions

- void **SetUp** () override
- void **TearDown** () override
- char ∗∗ **createArgv** (const std::vector< std::string > &args)
- void **cleanupArgv** (char ∗∗argv, int argc)

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

- gui/tests/unit/CLI/CLI_test.cpp

6.13 Utils.Colors Class Reference

Static Public Attributes

- str **BOLD** = "\033[1m"
- str **RED** = "\033[1m\033[31m"
- str **GREEN** = "\033[1m\033[32m"
- str **YELLOW** = "\033[1m\033[33m"
- str **BLUE** = "\033[1m\033[34m"
- str **MAGENTA** = "\033[1m\033[35m"
- str **CYAN** = "\033[1m\033[36m"
- str **WHITE** = "\033[1m\033[37m"
- str **RESET** = "\033[0m"

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

- ai/src/Utils/Utils.py

6.14 command_pf_s Struct Reference

Public Attributes

- char const ∗ **flag**
- bool(∗ **checker**)(const char ∗, const char ∗, params_t ∗)

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

- server/include/zappy.h

6.15 Communication Class Reference

Public Member Functions

- **Communication** (zappy::structs::Config config)
- void **sendMessage** (const std::string &message)
- bool **hasMessages** () const
- std::string **popMessage** ()
- bool **isConnected** () const
- void **disconnect** ()

Private Member Functions

- void **setupConnection** ()
- void **createSocket** ()
- void **connectToServer** ()
- void **setupNonBlocking** ()
- void **startCommunicationThread** ()
- void **communicationLoop** ()
- bool **handlePoll** ()
- void **processWrite** ()
- void **processRead** ()
- void **parseReceivedData** ()

Private Attributes

- zappy::structs::Config **_config**
- std::thread **_thread**
- std::mutex **_mutex**
- std::condition_variable **_cv**
- std::atomic< bool > **_running**
- std::atomic< bool > **_connected**
- std::queue< std::string > **_outgoingMessages**
- std::queue< std::string > **_incomingMessages**
- std::string **_receiveBuffer**
- std::string **_sendBuffer**
- int **_socket**
- struct pollfd **_pollfd**

Static Private Attributes

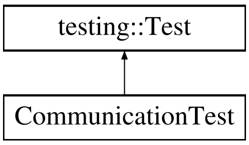
- static const int **BUFFER_SIZE** = 4096
- static const int **POLL_TIMEOUT** = 100
- static const char **MESSAGE_DELIMITER** = '\n'

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

- gui/src/Communication/Communication.hpp
- gui/src/Communication/Communication.cpp

6.16 CommunicationTest Class Reference

Inheritance diagram for CommunicationTest:



Protected Member Functions

- void **SetUp** () override
- void **TearDown** () override
- zappy::structs::Config **createValidConfig** ()

Protected Attributes

- std::unique_ptr< MockServer > **mockServer**

Static Protected Attributes

- static const int **TEST_PORT** = 9876

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

- gui/tests/unit/Communication/Communication_test.cpp

6.17 zappy::structs::Config Struct Reference

Public Attributes

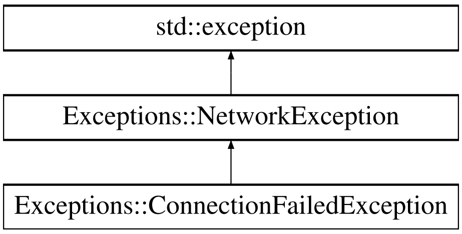
- int **port**
- std::string **hostname**

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

- gui/src/Utils/Constants.hpp

6.18 Exceptions::ConnectionFailedException Class Reference

Inheritance diagram for Exceptions::ConnectionFailedException:


std::exception
Exceptions::NetworkException
Exceptions::ConnectionFailedException

Public Member Functions

- **ConnectionFailedException** (const std::string &message)

Public Member Functions inherited from Exceptions::NetworkException

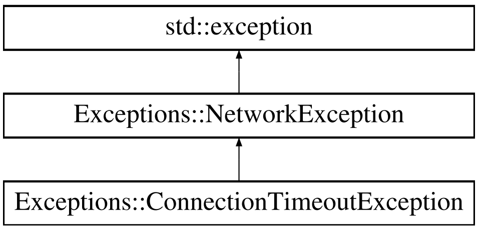
- **NetworkException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

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

- gui/src/Exceptions/Exceptions.hpp

6.19 Exceptions::ConnectionTimeoutException Class Reference

Inheritance diagram for Exceptions::ConnectionTimeoutException:


std::exception
Exceptions::NetworkException
Exceptions::ConnectionTimeoutException

Public Member Functions

- **ConnectionTimeoutException** (const std::string &message)

Public Member Functions inherited from Exceptions::NetworkException

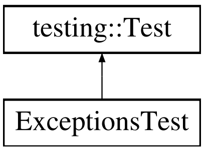
- **NetworkException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

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

- gui/src/Exceptions/Exceptions.hpp

6.20 ExceptionsTest Class Reference

Inheritance diagram for ExceptionsTest:



Protected Member Functions

- void **SetUp** () override
- void **TearDown** () override

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

- gui/tests/unit/Exceptions/Exceptions_test.cpp

6.21 MockServer Class Reference

Public Member Functions

- **MockServer** (int port)
- bool **start** ()
- void **stop** ()
- bool **sendToAllClients** (const std::string &message)
- bool **hasClients** () const

Private Member Functions

- void **acceptLoop** ()

Private Attributes

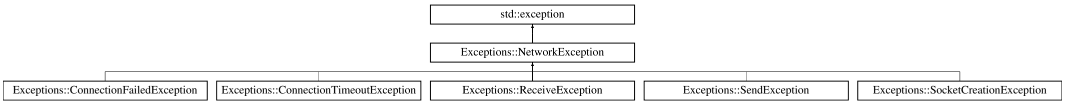
- int **_port**
- bool **_running**
- int **_serverSocket**
- std::thread **_thread**
- std::vector< int > **_clientSockets**

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

- gui/tests/unit/Communication/Communication_test.cpp

6.22 Exceptions::NetworkException Class Reference

Inheritance diagram for Exceptions::NetworkException:


std::exception
Exceptions::NetworkException
Exceptions::ConnectionFailedException
Exceptions::ConnectionTimeoutException
Exceptions::ReceiveException
Exceptions::SendException
Exceptions::SocketCreationException

Public Member Functions

- **NetworkException** (const std::string &message)
- virtual const char $*$ **what** () const noexcept override

Private Attributes

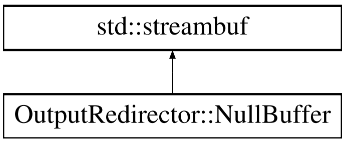
- std::string **_message**

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

- gui/src/Exceptions/Exceptions.hpp

6.23 OutputRedirector::NullBuffer Class Reference

Inheritance diagram for OutputRedirector::NullBuffer:



Protected Member Functions

- int **overflow** (int c) override

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

- gui/tests/unit/main_test.cpp

6.24 OutputRedirector Class Reference

Classes

- class NullBuffer

Private Attributes

- std::streambuf ∗ **originalCout**
- std::streambuf ∗ **originalCerr**
- NullBuffer **nullBuffer**

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

- gui/tests/unit/main_test.cpp

6.25 params_s Struct Reference

Public Attributes

- int **port**
- int **x**
- int **y**
- int **nb_team**
- char ∗∗ **teams**
- int **nb_client**
- int **freq**

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

- server/include/zappy.h

6.26 Parser.Parser Class Reference

Public Member Functions

- **__init__** (self)
- **run** (self)
- **parseConfig** (self)
- **parseJsons** (self)
- **getTests** (self)

Public Attributes

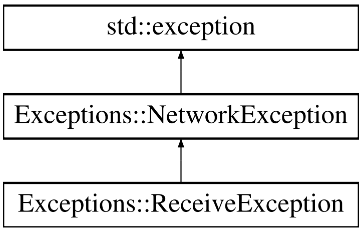
- **tests_folder**
- **tests_files_names**
- **tests_files**
- **output_folder**
- **testsObjects**

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

- tests/functional/Parser.py

6.27 Exceptions::ReceiveException Class Reference

Inheritance diagram for Exceptions::ReceiveException:



Public Member Functions

- **ReceiveException** (const std::string &message)

Public Member Functions inherited from Exceptions::NetworkException

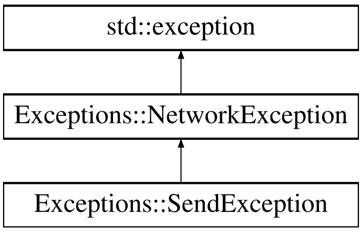
- **NetworkException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

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

- gui/src/Exceptions/Exceptions.hpp

6.28 Exceptions::SendException Class Reference

Inheritance diagram for Exceptions::SendException:



Public Member Functions

- **SendException** (const std::string &message)

Public Member Functions inherited from Exceptions::NetworkException

- **NetworkException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

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

- gui/src/Exceptions/Exceptions.hpp

6.29 server_s Struct Reference

Public Attributes

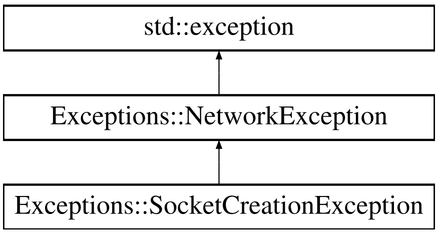
- params_t ∗ **params**

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

- server/include/zappy.h

6.30 Exceptions::SocketCreationException Class Reference

Inheritance diagram for Exceptions::SocketCreationException:


std::exception
Exceptions::NetworkException
Exceptions::SocketCreationException

Public Member Functions

- **SocketCreationException** (const std::string &message)

Public Member Functions inherited from Exceptions::NetworkException

- **NetworkException** (const std::string &message)
- virtual const char ∗ **what** () const noexcept override

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

- gui/src/Exceptions/Exceptions.hpp

6.31 TestCase.TestCase Class Reference

Public Member Functions

- **__init__** (self, name, desc, input, output, value, output_folder)
- **execute** (self)
- **check** (self)
- **displayPassed** (self, index)
- **displayFailed** (self, index)

Public Attributes

- **name**
- **desc**
- **input**
- **output**
- **value**
- **tty_mode**
- **tty_input**
- **succeed_after**
- **succeed_forced**
- **real_output**
- **real_value**
- **raw_output**

Protected Member Functions

- **_execute_normal** (self)
- **_execute_tty** (self)

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

- tests/functional/TestCase.py

6.32 test_cli.TestCLI Class Reference

Public Member Functions

- test_parse_args_valid (self)
- test_parse_args_valid_ip (self)
- test_parse_args_invalid_option (self)
- test_parse_args_missing_value (self)
- test_parse_args_not_enough_args (self)
- test_parse_port_invalid (self)
- test_parse_port_negative (self)
- test_parse_port_too_large (self)
- test_parse_name_empty (self)
- test_parse_name_whitespace (self)
- test_parse_machine_empty (self)
- test_parse_machine_invalid_ip_format (self)
- test_parse_machine_invalid_ip_value (self)
- test_parse_machine_invalid_ip_chars (self)
- test_validate_config_missing_port (self)
- test_validate_config_missing_name (self)
- test_validate_config_missing_machine (self)

6.32.1 Member Function Documentation

6.32.1.1 test_parse_args_invalid_option()

```
test_cli.TestCLI.test_parse_args_invalid_option (
              self )
```

```
Test parsing invalid option
```

6.32.1.2 test_parse_args_missing_value()

```
test_cli.TestCLI.test_parse_args_missing_value (
              self )
```

```
Test parsing missing value for option
```

6.32.1.3 test_parse_args_not_enough_args()

```
test_cli.TestCLI.test_parse_args_not_enough_args (
              self )
```

```
Test parsing not enough arguments
```

6.32.1.4 test_parse_args_valid()

```
test_cli.TestCLI.test_parse_args_valid (
              self )
```

```
Test parsing valid command line arguments
```

6.32.1.5 test_parse_args_valid_ip()

```
test_cli.TestCLI.test_parse_args_valid_ip (
              self )
```

```
Test parsing valid IP address
```

6.32.1.6 test_parse_machine_empty()

```
test_cli.TestCLI.test_parse_machine_empty (
              self )
```

```
Test parsing empty machine name
```

6.32.1.7 test_parse_machine_invalid_ip_chars()

```
test_cli.TestCLI.test_parse_machine_invalid_ip_chars (
              self )
```

```
Test parsing IP with invalid characters
```

6.32.1.8 test_parse_machine_invalid_ip_format()

```
test_cli.TestCLI.test_parse_machine_invalid_ip_format (
              self )
```

```
Test parsing invalid IP format
```

6.32.1.9 test_parse_machine_invalid_ip_value()

```
test_cli.TestCLI.test_parse_machine_invalid_ip_value (
              self )
```

```
Test parsing invalid IP value
```

6.32.1.10 test_parse_name_empty()

```
test_cli.TestCLI.test_parse_name_empty (
              self )
```

```
Test parsing empty team name
```

6.32.1.11 test_parse_name_whitespace()

```
test_cli.TestCLI.test_parse_name_whitespace (
              self )
```

```
Test parsing whitespace team name
```

6.32.1.12 test_parse_port_invalid()

```
test_cli.TestCLI.test_parse_port_invalid (
              self )
```

```
Test parsing invalid port
```

6.32.1.13 test_parse_port_negative()

```
test_cli.TestCLI.test_parse_port_negative (
              self )
```

```
Test parsing negative port
```

6.32.1.14 test_parse_port_too_large()

```
test_cli.TestCLI.test_parse_port_too_large (
              self )
```

```
Test parsing port that is too large
```

6.32.1.15 test_validate_config_missing_machine()

```
test_cli.TestCLI.test_validate_config_missing_machine (
              self )
```

```
Test validating config with missing machine
```

6.32.1.16 test_validate_config_missing_name()

```
test_cli.TestCLI.test_validate_config_missing_name (
              self )
```

```
Test validating config with missing name
```

6.32.1.17 test_validate_config_missing_port()

```
test_cli.TestCLI.test_validate_config_missing_port (
              self )
```

```
Test validating config with missing port
```

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

- ai/tests/unit/CLI/test_cli.py

Chapter 7

File Documentation

7.1 CLI.hpp

```
/*
** EPITECH PROJECT, 2025
** B-YEP-400-NAN-4-1-zappy-albane.merian
** File description:
** CLI
*/

#ifndef CLI_HPP_
#define CLI_HPP_

#include "../Utils/Constants.hpp"

class CLI {
    public:
        CLI(int ac, const char *const *av);
        ~CLI();

        zappy::structs::Config parseArguments(int ac, const char *const *av) const;

    private:
        int _ac;
        const char *const *_av;

        bool hasCorrectNumberOfArguments(int ac) const;
        int parsePort(const char *portStr) const;
        std::string parseHostname(const char *hostnameStr) const;
        void validateConfig(bool portFound, bool hostFound) const;
};

#endif /* !CLI_HPP_ */
```

7.2 Client.hpp

```
/*
** EPITECH PROJECT, 2025
** B-YEP-400-NAN-4-1-zappy-albane.merian
** File description:
** Client
*/

#ifndef CLIENT_HPP_
#define CLIENT_HPP_

#include "../Utils/Constants.hpp"
#include "../Communication/Communication.hpp"
#include <memory>

class Client {
    public:
        Client(int ac, const char *const *av);
        ~Client();

    private:
        zappy::structs::Config _config;
        std::unique_ptr<Communication> _communication;
        void initialize(int ac, const char * const *av);
};

#endif /* !CLIENT_HPP_ */
```

7.3 Communication.hpp

```
/*
** EPITECH PROJECT, 2025
** B-YEP-400-NAN-4-1-zappy-albane.merian
** File description:
** Communication
*/

#ifndef COMMUNICATION_HPP_
#define COMMUNICATION_HPP_

#include <thread>
#include <mutex>
#include <atomic>
#include <condition_variable>
#include <queue>
#include <string>
#include <vector>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <fcntl.h>
#include <poll.h>
#include <netdb.h>
#include "../Utils/Constants.hpp"
#include "../Exceptions/Exceptions.hpp"

class Communication {
    public:
        Communication(zappy::structs::Config config);
        ~Communication();

        void sendMessage(const std::string &message);
        bool hasMessages() const;
        std::string popMessage();
        bool isConnected() const;
        void disconnect();

    private:
        void setupConnection();
        void createSocket();
        void connectToServer();
        void setupNonBlocking();

        void startCommunicationThread();
        void communicationLoop();
        bool handlePoll();
        void processWrite();
        void processRead();

        void parseReceivedData();

        zappy::structs::Config _config;
        std::thread _thread;
        std::mutex _mutex;
        std::condition_variable _cv;
        std::atomic<bool> _running;
        std::atomic<bool> _connected;

        std::queue<std::string> _outgoingMessages;
        std::queue<std::string> _incomingMessages;

        std::string _receiveBuffer;
        std::string _sendBuffer;

        int _socket;
        struct pollfd _pollfd;
        static const int BUFFER_SIZE = 4096;
        static const int POLL_TIMEOUT = 100;
        static const char MESSAGE_DELIMITER = '\n';
};

#endif /* !COMMUNICATION_HPP_ */
```

7.4 Exceptions.hpp

```
/*
** EPITECH PROJECT, 2025
** B-YEP-400-NAN-4-1-zappy-albane.merian
** File description:
** Exceptions
*/

#ifndef EXCEPTIONS_HPP_
#define EXCEPTIONS_HPP_

#include <exception>
#include <string>
#include "../Utils/Constants.hpp"

namespace Exceptions {

    // CLI Exceptions
    class CLIParsingException : public std::exception {
        public:
            explicit CLIParsingException(const std::string &message)
                : _message(std::string(colors::RED) +
                          "CLI Parsing Error: " + message +
                          colors::RESET) {}

            virtual const char *what() const noexcept override {
                return _message.c_str();
            }

        private:
            std::string _message;
    };

    class CLIPortException : public CLIParsingException {
        public:
            explicit CLIPortException(const std::string &message)
                : CLIParsingException(std::string(colors::CYAN) +
                                     "Port Error: " + message +
                                     colors::RESET) {}
    };

    class CLIHostException : public CLIParsingException {
        public:
            explicit CLIHostException(const std::string &message)
                : CLIParsingException(std::string(colors::CYAN) +
                                     "Hostname Error: " + message +
                                     colors::RESET) {}
    };

    class CLIMissingArgumentException : public CLIParsingException {
        public:
            explicit CLIMissingArgumentException(const std::string &message)
                : CLIParsingException(std::string(colors::CYAN) +
                                     "Missing Argument: " + message +
                                     colors::RESET) {}
    };

    class CLIInvalidArgumentException : public CLIParsingException {
        public:
            explicit CLIInvalidArgumentException(const std::string &message)
                : CLIParsingException(std::string(colors::CYAN) +
                                     "Invalid Argument: " + message +
                                     colors::RESET) {}
    };

    class NetworkException : public std::exception {
        public:
            explicit NetworkException(const std::string &message)
                : _message(std::string(colors::RED) +
                          "Network Error: " + message +
                          colors::RESET) {}

            virtual const char *what() const noexcept override {
                return _message.c_str();
            }

        private:
            std::string _message;
    };

    class ConnectionFailedException : public NetworkException {
        public:
            explicit ConnectionFailedException(const std::string &message)
                : NetworkException(std::string(colors::CYAN) +
                                  "Connection Failed: " + message +
                                  colors::RESET) {}
    };

    class SocketCreationException : public NetworkException {
        public:
            explicit SocketCreationException(const std::string &message)
                : NetworkException(std::string(colors::CYAN) +
                                  "Socket Creation Failed: " + message +
                                  colors::RESET) {}
    };

    class ConnectionTimeoutException : public NetworkException {
        public:
            explicit ConnectionTimeoutException(const std::string &message)
                : NetworkException(std::string(colors::CYAN) +
                                  "Connection Timeout: " + message +
                                  colors::RESET) {}
    };

    class SendException : public NetworkException {
        public:
            explicit SendException(const std::string &message)
                : NetworkException(std::string(colors::CYAN) +
                                  "Send Error: " + message +
                                  colors::RESET) {}
    };

    class ReceiveException : public NetworkException {
        public:
            explicit ReceiveException(const std::string &message)
                : NetworkException(std::string(colors::CYAN) +
                                  "Receive Error: " + message +
                                  colors::RESET) {}
    };
}

#endif /* !EXCEPTIONS_HPP_ */
```

7.5 Constants.hpp

```
/*
** EPITECH PROJECT, 2025
** B-YEP-400-NAN-4-1-zappy-albane.merian
** File description:
** Constants
*/

#ifndef CONSTANTS_HPP_
#define CONSTANTS_HPP_

#include <string>

namespace zappy::constants {

    inline const char *USAGE_STRING = "USAGE: ./zappy_gui -p port -h machine\n"
                               "option\t\tdescription\n"
                               "-p port\t\tport number\n"
                               "-h machine\thostname of the server";

    inline const int FAILURE_EXIT_CODE = 84;
    inline const int SUCCESS_EXIT_CODE = 0;
};

namespace colors {

    inline const char *BOLD = "\033[1m";
    inline const char *RED = "\033[1m\033[31m";
    inline const char *GREEN = "\033[1m\033[32m";
    inline const char *YELLOW = "\033[1m\033[33m";
    inline const char *BLUE = "\033[1m\033[34m";
    inline const char *MAGENTA = "\033[1m\033[35m";
    inline const char *CYAN = "\033[1m\033[36m";
    inline const char *WHITE = "\033[1m\033[37m";
    inline const char *RESET = "\033[0m";

};

namespace zappy::structs {

    struct Config {
        int port;
        std::string hostname;
    };
};
```

```

#endif /* !CONSTANTS_HPP_ */
```

7.6 zappy.h

```
/*
** EPITECH PROJECT, 2025
** Zappy
** File description:
** Server :: Zappy header
*/

#include <stdbool.h>

#ifndef ZAPPY_H_
    #define ZAPPY_H_

typedef struct params_s {
    int port;
    int x;
    int y;
    int nb_team;
    char **teams;
    int nb_client;
    int freq;
} params_t;

typedef struct server_s {
    params_t *params;
} server_t;

typedef struct command_pf_s {
    char const *flag;
    bool (*checker)(const char *, const char *, params_t *);
} command_pf_t;

/* errors.c */
int helper(void);
void error_message(const char *message);

/* checkers.c */
bool check_port(char const *flag, char const *value, params_t *params);
bool check_width(char const *flag, char const *value, params_t *params);
bool check_height(char const *flag, char const *value, params_t *params);
bool check_client(char const *flag, char const *value, params_t *params);
bool check_freq(char const *flag, char const *value, params_t *params);

/* params.c */
params_t *check_args(int argc, char **argv);
void *free_params(params_t *params);

/* server.c */
server_t *init_server(int argc, char **argv);
void *free_server(server_t *server);

#endif /* !ZAPPY_H_ */
```