Zappy architecture

Generated by Doxygen 1.12.0

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

Si erreur dans le lancement comme ici :

```
npm error could not determine executable to run
```

executer ça : `npm install --save-dev @docusaurus/types`

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 you need move the my-zappy-doc, folder out of the repositorie because the Unicode emojis used make the generation fails, then execute this :

./generateDoc.sh

After that you can move back the folder in the documentation folder.

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
N	name of the team
R	incantation result
M	message
i	resource number

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
suc		unknown command		empty or unknown command

SERVER	CLIENT	DETAILS	TO A GUI client	TO ALL GUI client
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.

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

Command	Effect
/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 buffer_s Struct Reference

Public Attributes

- char **data** [BUFFER_SIZE]
- int **head**
- int **tail**
- int **full**

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

- server/include/buffer.h

6.2 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

6.3 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** = None
- **name** = None
- str **machine** = ""
- bool **port** = True
- bool **name** = True
- bool **machine** = True

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

- ai/src/CLI/CLI.py

6.4 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

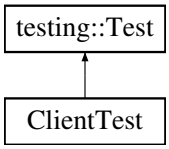
- zappy::structs::Config **_config**
- std::shared_ptr< ICommunication > **_communication**
- std::shared_ptr< GameInfos > **_gameInfos**
- std::unique_ptr< MsgHandler > **_msgHandler**
- std::unique_ptr< GUI > **_gui**

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

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

6.5 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

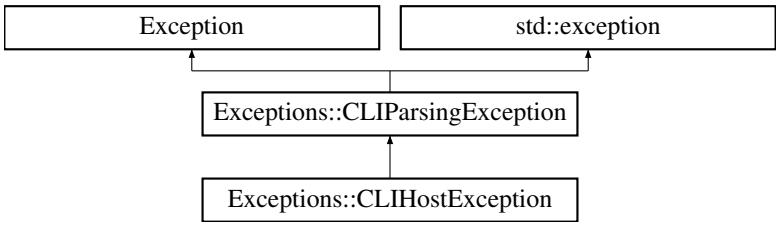
- std::stringstream **buffer**
- std::streambuf ∗ **originalCout**

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

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

6.6 Exceptions::CLIHostException Class Reference

Inheritance diagram for Exceptions::CLIHostException:



Public Member Functions

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

Public Member Functions inherited from Exceptions::CLIParsingException

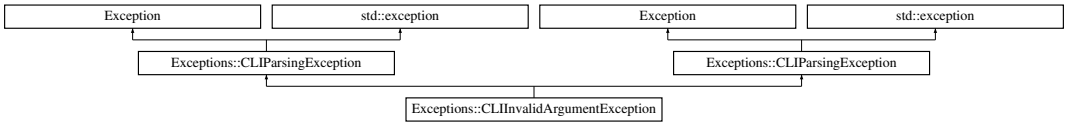
- **__init__** (self, str message)
- **CLIParsingException** (const std::string &message)
- virtual const char * **what** () const noexcept override

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

- gui/src/Exceptions/Exceptions.hpp

6.7 Exceptions::CLIInvalidArgumentException Class Reference

Inheritance diagram for 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

6.7.1 Constructor & Destructor Documentation

6.7.1.1 __init__()

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

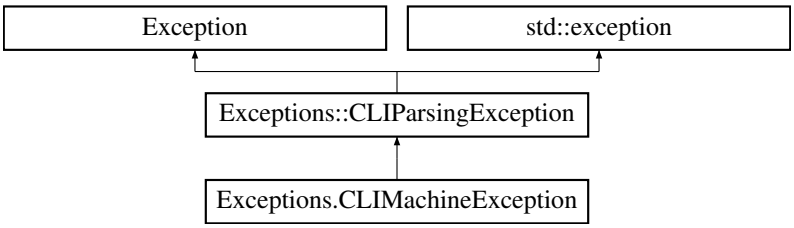
Reimplemented from Exceptions::CLIParsingException.

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

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

6.8 Exceptions.CLIMachineException Class Reference

Inheritance diagram for 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

6.8.1 Constructor & Destructor Documentation

6.8.1.1 __init__()

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

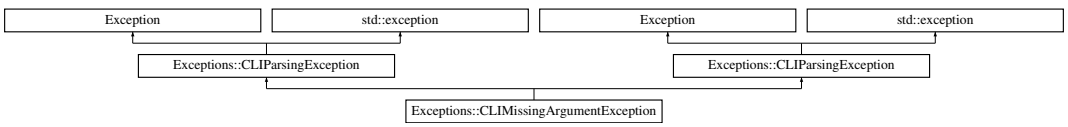
Reimplemented from Exceptions::CLIParsingException.

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

- ai/src/Exceptions/Exceptions.py

6.9 Exceptions::CLIMissingArgumentException Class Reference

Inheritance diagram for 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

6.9.1 Constructor & Destructor Documentation

6.9.1.1 __init__()

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

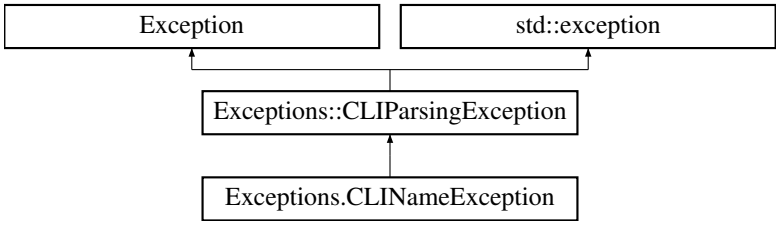
Reimplemented from Exceptions::CLIParsingException.

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

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

6.10 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

6.10.1 Constructor & Destructor Documentation

6.10.1.1 __init__()

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

Reimplemented from Exceptions::CLIParsingException.

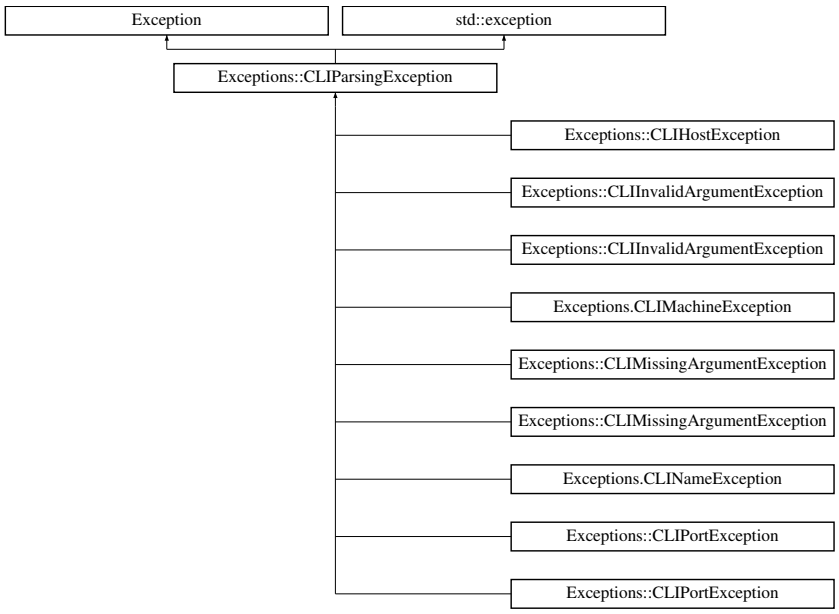
The documentation for this class was generated from the following file:

- ai/src/Exceptions/Exceptions.py

6.11 Exceptions::CLIParsingException Class Reference

EPITECH PROJECT, 2025 zappy File description: Exceptions.

Inheritance diagram for Exceptions::CLIParsingException:



Public Member Functions

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

Private Attributes

- std::string **_message**

6.11.1 Detailed Description

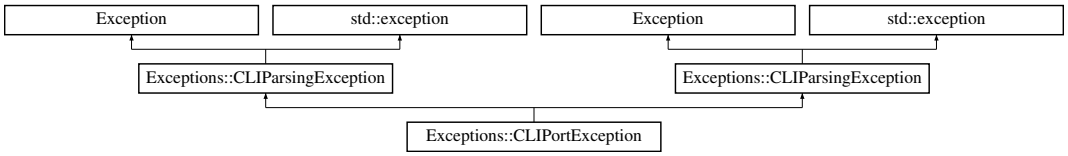
EPITECH PROJECT, 2025 zappy File description: Exceptions.

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

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

6.12 Exceptions::CLIPortException Class Reference

Inheritance diagram for 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

6.12.1 Constructor & Destructor Documentation

6.12.1.1 __init__()

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

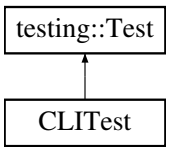
Reimplemented from Exceptions::CLIParsingException.

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

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

6.13 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:

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

6.14 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

6.15 command_pf_s Struct Reference

Public Attributes

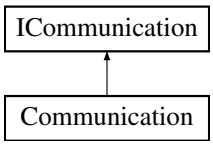
- char const ∗ **flag**
- bool(∗ **checker**)(const char ∗, const char ∗, params_t ∗)

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

- server/include/zappy.h

6.16 Communication Class Reference

Inheritance diagram for Communication:



Public Member Functions

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

Public Member Functions inherited from ICommunication

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'

6.16.1 Member Function Documentation

6.16.1.1 disconnect()

```
void Communication::disconnect ()  [override], [virtual]
```

Implements ICommunication.

6.16.1.2 hasMessages()

```
bool Communication::hasMessages () const  [override], [virtual]
```

Implements ICommunication.

6.16.1.3 isConnected()

```
bool Communication::isConnected () const  [override], [virtual]
```

Implements ICommunication.

6.16.1.4 popMessage()

```
std::string Communication::popMessage ()  [override], [virtual]
```

Implements ICommunication.

6.16.1.5 sendMessage()

```
void Communication::sendMessage (
            const std::string & message)  [override], [virtual]
```

Implements ICommunication.

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

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

6.17 Communication.Communication Class Reference

Public Member Functions

- **__init__** (self, str name, str host, int port)
- **connectToServer** (self)
- **sendForward** (self)
- **sendRight** (self)
- **sendLeft** (self)
- list[dict[str, int]] **getLook** (self)
- dict[str, int] **getInventory** (self)
- **sendBroadcast** (self, str message)
- int **getCetConnectNbr** (self)
- **sendFork** (self)
- **sendEject** (self)
- **sendTakeObject** (self, str object_name)
- **sendSetObject** (self, str object_name)
- **sendIncantation** (self)

Protected Attributes

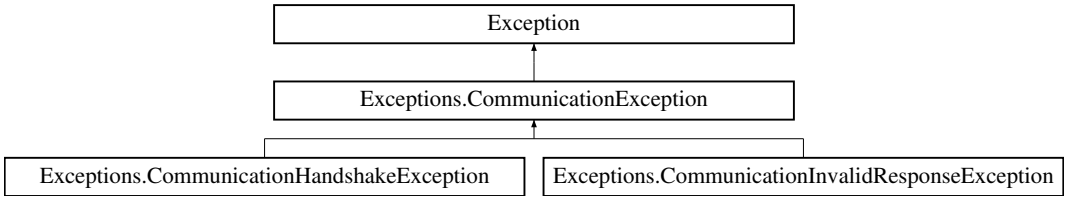
- **_name** = name
- **_host** = host
- **_port** = port
- **_socket** = Socket(host, port)

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

- ai/src/Communication/Communication.py

6.18 Exceptions.CommunicationException Class Reference

Inheritance diagram for Exceptions.CommunicationException:



Public Member Functions

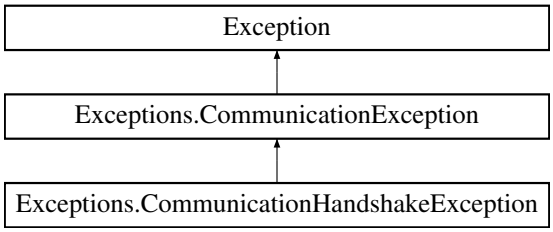
- **__init__** (self, str message)

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

- ai/src/Exceptions/Exceptions.py

6.19 Exceptions.CommunicationHandshakeException Class Reference

Inheritance diagram for Exceptions.CommunicationHandshakeException:



Public Member Functions

- __init__ (self, str message)

Public Member Functions inherited from Exceptions.CommunicationException

6.19.1 Constructor & Destructor Documentation

6.19.1.1 __init__()

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

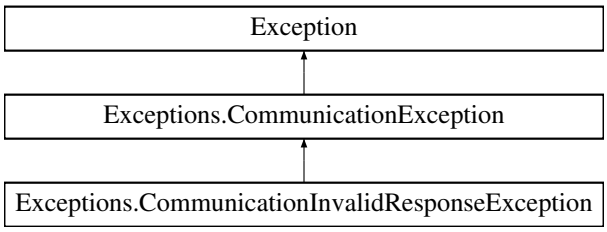
Reimplemented from Exceptions.CommunicationException.

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

- ai/src/Exceptions/Exceptions.py

6.20 Exceptions.CommunicationInvalidResponseException Class Reference

Inheritance diagram for Exceptions.CommunicationInvalidResponseException:



Public Member Functions

- __init__ (self, str message)

Public Member Functions inherited from Exceptions.CommunicationException

6.20.1 Constructor & Destructor Documentation

6.20.1.1 __init__()

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

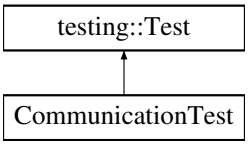
Reimplemented from Exceptions.CommunicationException.

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

- ai/src/Exceptions/Exceptions.py

6.21 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:

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

6.22 zappy::structs::Config Struct Reference

Public Attributes

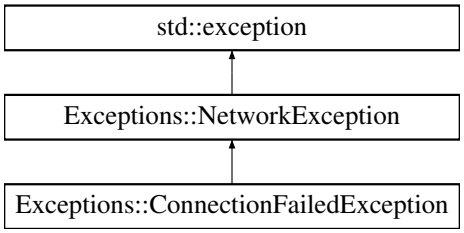
- int **port**
- std::string **hostname**

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

- gui/src/Utils/Constants.hpp

6.23 Exceptions::ConnectionFailedException Class Reference

Inheritance diagram for Exceptions::ConnectionFailedException:



Public Member Functions

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

Public Member Functions inherited from Exceptions::NetworkException

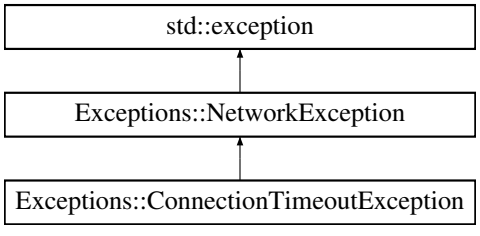
- **NetworkException** (const std::string &message)
- virtual const char * **what** () const noexcept override

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

- gui/src/Exceptions/Exceptions.hpp

6.24 Exceptions::ConnectionTimeoutException Class Reference

Inheritance diagram for 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

6.25 zappy::structs::Egg Struct Reference

Public Member Functions

- **Egg** (int eggNumber=0, int playerNumber=0, int x=0, int y=0, bool hatched=false, const std::string &team↩ Name="")

Public Attributes

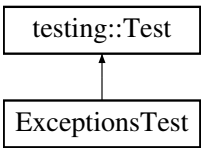
- int **eggNumber**
- int **playerNumber**
- int **x**
- int **y**
- bool **hatched**
- std::string **teamName**

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

- gui/src/Utils/Constants.hpp

6.26 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:

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

6.27 GameInfos Class Reference

Public Member Functions

- void **setMapSize** (int width, int height)
- std::pair< int, int > **getMapSize** () const
- void **setTimeUnit** (int timeUnit)
- int **getTimeUnit** () const
- void **updateTile** (const zappy::structs::Tile tile)
- const std::vector< zappy::structs::Tile > **getTiles** () const
- const zappy::structs::Tile **getTile** (int x, int y) const
- void **updateTeamName** (const std::string &teamName)
- const std::vector< std::string > **getTeamNames** () const
- void **addPlayer** (const zappy::structs::Player player)
- void **updatePlayerPosition** (int playerNumber, int x, int y)
- void **updatePlayerLevel** (int playerNumber, int level)
- void **updatePlayerInventory** (int playerNumber, const zappy::structs::Inventory inventory)
- void **updatePlayerExpulsion** (int playerNumber)
- void **updatePlayerDeath** (int playerNumber)
- void **updatePlayerResourceAction** (int playerNumber, int resourceId, bool isCollecting)
- void **updatePlayerFork** (int playerNumber)
- const std::vector< zappy::structs::Player > **getPlayers** () const
- void **addPlayerBroadcast** (int playerNumber, const std::string &message)
- std::vector< std::pair< int, std::string > > **getPlayersBroadcasting** () const
- void **addIncantation** (const zappy::structs::Incantation incantation)
- void **removeIncantation** (int x, int y, int result)
- void **addEgg** (const zappy::structs::Egg egg)
- void **updateEggHatched** (int eggNumber)
- void **updateEggDeath** (int eggNumber)
- const std::vector< zappy::structs::Egg > **getEggs** () const
- void **setGameOver** (const std::string &winningTeam)
- std::pair< bool, std::string > **isGameOver** () const

Private Attributes

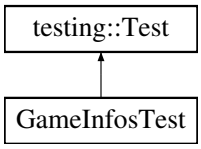
- int **_mapWidth**
- int **_mapHeight**
- int **_timeUnit**
- std::vector< zappy::structs::Tile > **_tiles**
- std::vector< std::string > **_teamNames**
- std::vector< zappy::structs::Player > **_players**
- std::vector< std::pair< int, bool > > **_playersExpulsing**
- std::vector< std::pair< int, std::string > > **_playersBroadcasting**
- std::vector< zappy::structs::Incantation > **_incantations**
- std::vector< zappy::structs::Egg > **_eggs**
- bool **_gameOver**
- std::string **_winningTeam**

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

- gui/src/Game/GameInfos.hpp
- gui/src/Game/GameInfos.cpp

6.28 GameInfosTest Class Reference

Inheritance diagram for GameInfosTest:



Protected Member Functions

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

Protected Attributes

- std::unique_ptr< GameInfos > **gameInfos**

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

- tests/unit/gui/Game/GameInfos_test.cpp

6.29 graph_s Struct Reference

Public Attributes

- int **fd**
- struct pollfd * **pollfd**

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

- server/include/zappy.h

6.30 GUI Class Reference

Public Member Functions

- void **run** ()

Private Member Functions

- void **updateCamera** ()
- void **drawEnvironment** ()

Private Attributes

- RayLib **_raylib**
- bool **_isRunning**

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

- gui/src/Graphic/GUI.hpp
- gui/src/Graphic/GUI.cpp

6.31 Hash.Hash Class Reference

Public Member Functions

- **__init__** (self, str hash_key)
- bytes **simple_xor** (self, bytes data)
- str **hashMessage** (self, str message)
- str **unHashMessage** (self, str hex_message)

Public Attributes

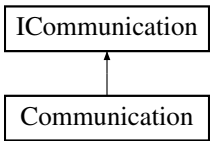
- int **key** = sum((i + 1) ∗ ord(c) for i, c in enumerate(hash_key)) % 256

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

- ai/src/Hash/Hash.py

6.32 ICommunication Class Reference

Inheritance diagram for ICommunication:



Public Member Functions

- virtual void **sendMessage** (const std::string &message)=0
- virtual bool **hasMessages** () const =0
- virtual std::string **popMessage** ()=0
- virtual bool **isConnected** () const =0
- virtual void **disconnect** ()=0

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

- gui/src/Communication/ICommunication.hpp

6.33 zappy::structs::Incantation Struct Reference

Public Member Functions

- **Incantation** (int x=0, int y=0, int level=1, const std::vector< int > &players={})

Public Attributes

- int **x**
- int **y**
- int **level**
- std::vector< int > **players**

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

- gui/src/Utils/Constants.hpp

6.34 zappy::structs::Inventory Struct Reference

Public Member Functions

- **Inventory** (int food=0, int linemate=0, int deraumere=0, int sibur=0, int mendiane=0, int phiras=0, int thystame=0)

Public Attributes

- int **food**
- int **linemate**
- int **deraumere**
- int **sibur**
- int **mendiane**
- int **phiras**
- int **thystame**

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

- gui/src/Utils/Constants.hpp

6.35 inventory_s Struct Reference

Public Attributes

- int **nbLinemate**
- int **nbDeraumere**
- int **nbSibur**
- int **nbMendiane**
- int **nbPhiras**
- int **nbThystame**

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

- server/include/game.h

6.36 lives_s Struct Reference

Public Attributes

- int **freq**
- int **nbFood**
- time_t **startRefresh**
- time_t **endRefresh**

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

- server/include/game.h

6.37 map_s Struct Reference

Public Attributes

- int **width**
- int **heigt**
- team_t * **teams**
- ressources_t * **ressources**

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

- server/include/game.h

6.38 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:

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

6.39 MsgHandler Class Reference

Public Member Functions

- **MsgHandler** (std::shared_ptr< GameInfos > gameInfos, std::shared_ptr< ICommunication > communication)
- void **start** ()
- void **stop** ()

Protected Member Functions

- void **messageLoop** ()
- void **handleMessage** (const std::string &message)
- bool **handleWelcomeMessage** (const std::string &message)
- bool **handleMszMessage** (const std::string &message)
- bool **handleBctMessage** (const std::string &message)
- bool **handleTnaMessage** (const std::string &message)
- bool **handlePnwMessage** (const std::string &message)
- bool **handlePpoMessage** (const std::string &message)
- bool **handlePlvMessage** (const std::string &message)
- bool **handlePinMessage** (const std::string &message)
- bool **handlePexMessage** (const std::string &message)
- bool **handlePbcMessage** (const std::string &message)
- bool **handlePicMessage** (const std::string &message)
- bool **handlePieMessage** (const std::string &message)
- bool **handlePfkMessage** (const std::string &message)
- bool **handlePdrMessage** (const std::string &message)
- bool **handlePgtMessage** (const std::string &message)
- bool **handlePdiMessage** (const std::string &message)
- bool **handleEnwMessage** (const std::string &message)
- bool **handleEboMessage** (const std::string &message)
- bool **handleEdiMessage** (const std::string &message)
- bool **handleSgtMessage** (const std::string &message)
- bool **handleSstMessage** (const std::string &message)
- bool **handleSegMessage** (const std::string &message)
- bool **handleSmgMessage** (const std::string &message)
- bool **handleSucMessage** (const std::string &message)
- bool **handleSbpMessage** (const std::string &message)

Private Attributes

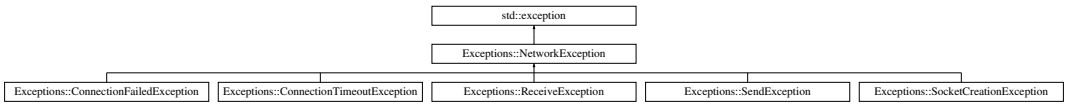
- std::thread **_thread**
- std::atomic< bool > **_running**
- std::mutex **_mutex**
- std::condition_variable **_condition**
- std::shared_ptr< GameInfos > **_gameInfos**
- std::shared_ptr< ICommunication > **_communication**
- std::mutex **_gameInfosMutex**
- std::map< std::string, std::function< bool(const std::string &)> > **_messageHandlers**

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

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

6.40 Exceptions::NetworkException Class Reference

Inheritance diagram for Exceptions::NetworkException:



Public Member Functions

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

Private Attributes

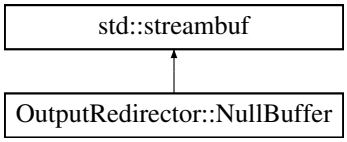
- std::string **_message**

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

- gui/src/Exceptions/Exceptions.hpp

6.41 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:

- tests/unit/gui/main_test.cpp

6.42 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:

- tests/unit/gui/main_test.cpp

6.43 params_s Struct Reference

Public Attributes

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

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

- server/include/zappy.h

6.44 Parser.Parser Class Reference

Public Member Functions

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

Public Attributes

- str **tests_folder** = ""
- list **tests_files_names** = [ ]
- list **tests_files** = [ ]
- str **output_folder** = ""
- list **testsObjects** = [ ]
- str **tests_files_names** = self.tests_folder + f + ".json"

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

- tests/functional/Parser.py

6.45 Player.Player Class Reference

Public Member Functions

- **__init__** (self, name)
- **__str__** (self)
- None **begin_incantation** (self)
- None **lay_an_egg** (self)
- None **loop** (self)

Public Attributes

- str **teamName** = name
- int **level** = 1
- Hash **hash** = Hash(name)
- dict inventory
- bool **alive** = True
- bool **in_incantation** = False
- Hash **alive** = self.hash.hashMessage("J'ai tous les objets pour incantation !")

6.45.1 Member Data Documentation

6.45.1.1 inventory

```
dict Player.Player.inventory
```

Initial value:

```
=  {
            "food": 10, "linemate": 0, "deraumere": 0, "sibur": 0,
            "mendiane": 0, "phiras": 0, "thystame": 0
        }
```

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

- ai/src/Player/Player.py

6.46 zappy::structs::Player Struct Reference

Public Member Functions

- **Player** (int number=0, int x=0, int y=0, int orientation=0, int level=1, const std::string &teamName="", struct Inventory inventory=Inventory())

Public Attributes

- int **number**
- int **x**
- int **y**
- int **orientation**
- int **level**
- std::string **teamName**
- struct Inventory **inventory**

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

- gui/src/Utils/Constants.hpp

6.47 player_s Struct Reference

Public Attributes

- int **id**
- int **level**
- int **posX**
- int **posY**
- bool **isAlive**
- direction_t **direction**
- inventory_t * **inventory**
- lives_t * **lives**
- struct player_s * **next**

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

- server/include/game.h

6.48 RayLib Class Reference

Public Member Functions

- void **initWindow** (int width, int height, const std::string &title)
- void **closeWindow** ()
- bool **windowShouldClose** () const
- void **beginDrawing** ()
- void **endDrawing** ()
- void **clearBackground** (Color color=WHITE)
- bool **isWindowReady** () const
- void **begin3DMode** ()
- void **end3DMode** ()
- void **initCamera** ()
- void **setCameraPosition** (Vector3 position)
- void **setCameraTarget** (Vector3 target)
- void **setCameraUp** (Vector3 up)
- void **setCameraFovy** (float fovy)
- void **setCameraProjection** (int projection)
- void **updateCamera** (int mode=CAMERA_FREE)
- Camera3D **getCamera** () const
- void **drawGrid** (int slices, float spacing)
- void **drawCube** (Vector3 position, float width, float height, float length, Color color)
- void **drawCubeWires** (Vector3 position, float width, float height, float length, Color color)
- void **drawSphere** (Vector3 position, float radius, Color color)
- void **drawCylinder** (Vector3 position, float radiusTop, float radiusBottom, float height, int slices, Color color)
- void **drawPlane** (Vector3 position, Vector2 size, Color color)

Private Attributes

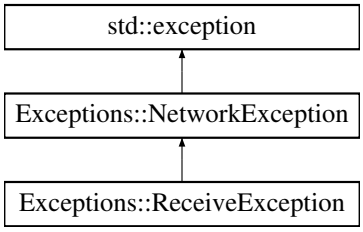
- bool **_isInitialized**
- Camera3D **_camera**

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

- gui/src/Graphic/RayLib/RayLib.hpp
- gui/src/Graphic/RayLib/RayLib.cpp

6.49 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

6.50 ressources_s Struct Reference

Public Attributes

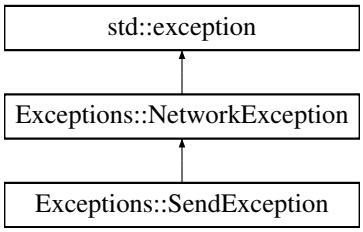
- crystal_t **type**
- int **posX**
- int **posY**
- struct ressources_s * **next**

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

- server/include/game.h

6.51 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

6.52 server_s Struct Reference

Public Attributes

- int **sockfd**
- params_t * **params**
- map_t * **map**
- graph_t * **graph**

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

- server/include/zappy.h

6.53 Socket.Socket Class Reference

Public Member Functions

- **__init__** (self, str host, int port)
- **connect** (self)
- **send** (self, str content)
- str **receive** (self)
- **close** (self)

Protected Attributes

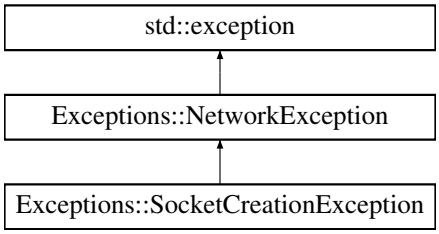
- **_host** = host
- **_port** = port
- tuple **_address** = (host, port)
- **_socket** = None
- str **_buffer** = ""

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

- ai/src/Communication/Socket.py

6.54 Exceptions::SocketCreationException Class Reference

Inheritance diagram for Exceptions::SocketCreationException:



Public Member Functions

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

Public Member Functions inherited from Exceptions::NetworkException

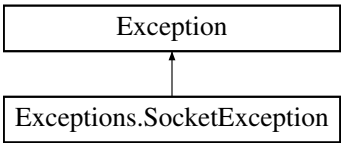
- **NetworkException** (const std::string &message)
- virtual const char * **what** () const noexcept override

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

- gui/src/Exceptions/Exceptions.hpp

6.55 Exceptions.SocketException Class Reference

Inheritance diagram for Exceptions.SocketException:



Public Member Functions

- **__init__** (self, str message)

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

- ai/src/Exceptions/Exceptions.py

6.56 team_s Struct Reference

Public Attributes

- char * **name**
- int **nbPlayers**
- int **nbPlayerAlive**
- player_t * **players**
- struct team_s * **next**

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

- server/include/game.h

6.57 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** = name
- **desc** = desc
- **input** = input
- **output** = output
- **value** = value
- bool **tty_mode** = False
- list **tty_input** = [ ]
- **succeed_after** = None
- bool **succeed_forced** = False
- **real_output** = None
- int **real_value** = None
- **raw_output** = None

Protected Member Functions

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

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

- tests/functional/TestCase.py

6.58 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)

6.58.1 Member Function Documentation

6.58.1.1 test_parse_args_invalid_option()

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

Test parsing invalid option

6.58.1.2 test_parse_args_missing_value()

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

Test parsing missing value for option

6.58.1.3 test_parse_args_not_enough_args()

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

Test parsing not enough arguments

6.58.1.4 test_parse_args_valid()

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

Test parsing valid command line arguments

6.58.1.5 test_parse_args_valid_ip()

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

Test parsing valid IP address

6.58.1.6 test_parse_machine_empty()

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

Test parsing empty machine name

6.58.1.7 test_parse_machine_invalid_ip_chars()

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

Test parsing IP with invalid characters

6.58.1.8 test_parse_machine_invalid_ip_format()

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

Test parsing invalid IP format

6.58.1.9 test_parse_machine_invalid_ip_value()

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

Test parsing invalid IP value

6.58.1.10 test_parse_name_empty()

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

Test parsing empty team name

6.58.1.11 test_parse_name_whitespace()

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

Test parsing whitespace team name

6.58.1.12 test_parse_port_invalid()

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

Test parsing invalid port

6.58.1.13 test_parse_port_negative()

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

Test parsing negative port

6.58.1.14 test_parse_port_too_large()

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

Test parsing port that is too large

6.58.1.15 test_validate_config_missing_machine()

```
test_cli.TestCLI.test_validate_config_missing_machine (
               self)

Test validating config with missing machine
```

6.58.1.16 test_validate_config_missing_name()

```
test_cli.TestCLI.test_validate_config_missing_name (
               self)

Test validating config with missing name
```

6.58.1.17 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:

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

6.59 test_com.TestCommunication Class Reference

Public Member Functions

- test_communication_init (self)
- test_connect_to_server_success (self, mock_socket_class)
- test_connect_to_server_invalid_welcome (self, mock_socket_class)
- test_connect_to_server_invalid_slots (self, mock_socket_class)
- test_connect_to_server_invalid_coordinates (self, mock_socket_class)
- test_connect_to_server_invalid_handshake_values (self, mock_socket_class)
- test_send_forward_success (self, mock_socket_class)
- test_send_forward_invalid_response (self, mock_socket_class)
- test_send_right_success (self, mock_socket_class)
- test_send_left_success (self, mock_socket_class)
- test_get_look_success (self, mock_socket_class)
- test_get_look_invalid_format (self, mock_socket_class)
- test_get_inventory_success (self, mock_socket_class)
- test_get_inventory_empty (self, mock_socket_class)
- test_get_inventory_invalid_item_format (self, mock_socket_class)
- test_send_broadcast_success (self, mock_socket_class)
- test_get_connect_nbr_success (self, mock_socket_class)
- test_get_connect_nbr_invalid_number (self, mock_socket_class)
- test_send_fork_success (self, mock_socket_class)
- test_send_eject_success (self, mock_socket_class)
- test_send_take_object_success (self, mock_socket_class)
- test_send_set_object_success (self, mock_socket_class)
- test_send_incantation_elevation_underway (self, mock_socket_class)
- test_send_incantation_current_level (self, mock_socket_class)
- test_send_incantation_ko_response (self, mock_socket_class)
- test_send_incantation_invalid_level_format (self, mock_socket_class)
- test_send_incantation_unexpected_response (self, mock_socket_class)

6.59.1 Member Function Documentation

6.59.1.1 test_communication_init()

```
test_com.TestCommunication.test_communication_init (
              self)
```

Test communication initialization

6.59.1.2 test_connect_to_server_invalid_coordinates()

```
test_com.TestCommunication.test_connect_to_server_invalid_coordinates (
              self,
              mock_socket_class)
```

Test server connection with invalid coordinates

6.59.1.3 test_connect_to_server_invalid_handshake_values()

```
test_com.TestCommunication.test_connect_to_server_invalid_handshake_values (
              self,
              mock_socket_class)
```

Test server connection with invalid handshake values (zero or negative)

6.59.1.4 test_connect_to_server_invalid_slots()

```
test_com.TestCommunication.test_connect_to_server_invalid_slots (
              self,
              mock_socket_class)
```

Test server connection with invalid slots number

6.59.1.5 test_connect_to_server_invalid_welcome()

```
test_com.TestCommunication.test_connect_to_server_invalid_welcome (
              self,
              mock_socket_class)
```

Test server connection with invalid welcome message

6.59.1.6 test_connect_to_server_success()

```
test_com.TestCommunication.test_connect_to_server_success (
              self,
              mock_socket_class)
```

Test successful server connection and handshake

6.59.1.7 test_get_connect_nbr_invalid_number()

```
test_com.TestCommunication.test_get_connect_nbr_invalid_number (
              self,
              mock_socket_class)
```

Test connect_nbr command with invalid number

6.59.1.8 test_get_connect_nbr_success()

```
test_com.TestCommunication.test_get_connect_nbr_success (
              self,
              mock_socket_class)
```

Test successful connect_nbr command

6.59.1.9 test_get_inventory_empty()

```
test_com.TestCommunication.test_get_inventory_empty (
              self,
              mock_socket_class)
```

Test inventory command with empty inventory

6.59.1.10 test_get_inventory_invalid_item_format()

```
test_com.TestCommunication.test_get_inventory_invalid_item_format (
              self,
              mock_socket_class)
```

Test inventory command with invalid item format

6.59.1.11 test_get_inventory_success()

```
test_com.TestCommunication.test_get_inventory_success (
              self,
              mock_socket_class)
```

Test successful inventory command

6.59.1.12 test_get_look_invalid_format()

```
test_com.TestCommunication.test_get_look_invalid_format (
              self,
              mock_socket_class)
```

Test look command with invalid response format

6.59.1.13 test_get_look_success()

```
test_com.TestCommunication.test_get_look_success (
              self,
              mock_socket_class)
```

Test successful look command

6.59.1.14 test_send_broadcast_success()

```
test_com.TestCommunication.test_send_broadcast_success (
              self,
              mock_socket_class)
```

Test successful broadcast command

6.59.1.15 test_send_eject_success()

```
test_com.TestCommunication.test_send_eject_success (
              self,
              mock_socket_class)
```

Test successful eject command

6.59.1.16 test_send_fork_success()

```
test_com.TestCommunication.test_send_fork_success (
              self,
              mock_socket_class)
```

Test successful fork command

6.59.1.17 test_send_forward_invalid_response()

```
test_com.TestCommunication.test_send_forward_invalid_response (
              self,
              mock_socket_class)
```

Test forward command with invalid response

6.59.1.18 test_send_forward_success()

```
test_com.TestCommunication.test_send_forward_success (
              self,
              mock_socket_class)
```

Test successful forward command

6.59.1.19 test_send_incantation_current_level()

```
test_com.TestCommunication.test_send_incantation_current_level (
              self,
              mock_socket_class)
```

Test incantation command with current level response

6.59.1.20 test_send_incantation_elevation_underway()

```
test_com.TestCommunication.test_send_incantation_elevation_underway (
              self,
              mock_socket_class)
```

Test incantation command with elevation underway response

6.59.1.21 test_send_incantation_invalid_level_format()

```
test_com.TestCommunication.test_send_incantation_invalid_level_format (
              self,
              mock_socket_class)
```

Test incantation command with invalid level format

6.59.1.22 test_send_incantation_ko_response()

```
test_com.TestCommunication.test_send_incantation_ko_response (
              self,
              mock_socket_class)
```

Test incantation command with ko response

6.59.1.23 test_send_incantation_unexpected_response()

```
test_com.TestCommunication.test_send_incantation_unexpected_response (
              self,
              mock_socket_class)
```

Test incantation command with unexpected response

6.59.1.24 test_send_left_success()

```
test_com.TestCommunication.test_send_left_success (
              self,
              mock_socket_class)
```

Test successful left command

6.59.1.25 test_send_right_success()

```
test_com.TestCommunication.test_send_right_success (
              self,
              mock_socket_class)
```

Test successful right command

6.59.1.26 test_send_set_object_success()

```
test_com.TestCommunication.test_send_set_object_success (
              self,
              mock_socket_class)
```

```
Test successful set object command
```

6.59.1.27 test_send_take_object_success()

```
test_com.TestCommunication.test_send_take_object_success (
              self,
              mock_socket_class)
```

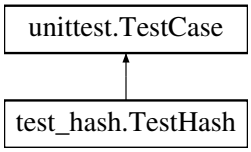
```
Test successful take object command
```

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

- tests/unit/ai/Communication/test_com.py

6.60 test_hash.TestHash Class Reference

Inheritance diagram for test_hash.TestHash:



Public Member Functions

- **setUp** (self)
- **test_hash_initialization** (self)
- **test_simple_xor** (self)
- **test_hash_message** (self)
- **test_unhash_message** (self)
- **test_hash_unhash_roundtrip** (self)
- **test_different_keys_produce_different_hashes** (self)

Public Attributes

- **hash_obj** = Hash("test_key")

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

- tests/unit/ai/Hash/test_hash.py

6.61 test_integration.TestIntegration Class Reference

Public Member Functions

- server_process (self)
- test_socket_real_connection (self, server_process)
- test_communication_real_handshake (self, server_process)
- test_communication_real_commands (self, server_process)
- test_communication_real_object_manipulation (self, server_process)
- test_communication_real_incantation (self, server_process)
- test_multiple_clients (self, server_process)
- test_socket_connection_refused (self)
- test_communication_connection_refused (self)
- test_socket_buffer_with_real_server (self, server_process)
- test_invalid_team_name (self, server_process)

6.61.1 Detailed Description

```
Integration tests with the actual zappy server
```

6.61.2 Member Function Documentation

6.61.2.1 server_process()

```
test_integration.TestIntegration.server_process (
              self)
```

```
Start the zappy server for integration testing
```

6.61.2.2 test_communication_connection_refused()

```
test_integration.TestIntegration.test_communication_connection_refused (
              self)
```

```
Test Communication behavior when server is not running
```

6.61.2.3 test_communication_real_commands()

```
test_integration.TestIntegration.test_communication_real_commands (
              self,
              server_process)
```

```
Test Communication class commands with real server
```

6.61.2.4 test_communication_real_handshake()

```
test_integration.TestIntegration.test_communication_real_handshake (
              self,
              server_process)
```

Test Communication class handshake with real server

6.61.2.5 test_communication_real_incantation()

```
test_integration.TestIntegration.test_communication_real_incantation (
              self,
              server_process)
```

Test incantation command with real server

6.61.2.6 test_communication_real_object_manipulation()

```
test_integration.TestIntegration.test_communication_real_object_manipulation (
              self,
              server_process)
```

Test object manipulation commands with real server

6.61.2.7 test_invalid_team_name()

```
test_integration.TestIntegration.test_invalid_team_name (
              self,
              server_process)
```

Test behavior with invalid team name

6.61.2.8 test_multiple_clients()

```
test_integration.TestIntegration.test_multiple_clients (
              self,
              server_process)
```

Test multiple clients connecting to the same server

6.61.2.9 test_socket_buffer_with_real_server()

```
test_integration.TestIntegration.test_socket_buffer_with_real_server (
              self,
              server_process)
```

Test Socket buffer handling with real server responses

6.61.2.10 test_socket_connection_refused()

```
test_integration.TestIntegration.test_socket_connection_refused (
              self)
```

Test Socket behavior when server is not running

6.61.2.11 test_socket_real_connection()

```
test_integration.TestIntegration.test_socket_real_connection (
              self,
              server_process)
```

Test Socket class with real server connection

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

- tests/unit/ai/Communication/test_integration.py

6.62 test_player.TestPlayer Class Reference

Public Member Functions

- **setup_method** (self)
- **test_player_initialization** (self)
- **test_player_str_representation** (self)
- **test_begin_incantation** (self, mock_print, mock_sleep)
- **test_lay_a_egg** (self, mock_print)

Public Attributes

- **player** = Player("test_team")

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

- tests/unit/ai/Player/test_player.py

6.63 test_socket.TestSocket Class Reference

Public Member Functions

- test_socket_init (self)
- test_socket_connect_success (self, mock_socket)
- test_socket_connect_failure (self, mock_socket)
- test_socket_send_success (self, mock_socket)
- test_socket_send_unicode (self, mock_socket)
- test_socket_receive_single_message (self, mock_socket)
- test_socket_receive_partial_messages (self, mock_socket)
- test_socket_receive_multiple_messages_in_buffer (self, mock_socket)
- test_socket_receive_timeout (self, mock_socket)
- test_socket_receive_connection_closed (self, mock_socket)
- test_socket_receive_unicode (self, mock_socket)
- test_socket_close (self, mock_socket)
- test_socket_buffer_persistence (self, mock_socket)
- test_socket_different_hosts_and_ports (self)

6.63.1 Member Function Documentation

6.63.1.1 test_socket_buffer_persistence()

```
test_socket.TestSocket.test_socket_buffer_persistence (
              self,
              mock_socket)
```

Test that buffer content persists between receive calls

6.63.1.2 test_socket_close()

```
test_socket.TestSocket.test_socket_close (
              self,
              mock_socket)
```

Test socket close

6.63.1.3 test_socket_connect_failure()

```
test_socket.TestSocket.test_socket_connect_failure (
              self,
              mock_socket)
```

Test socket connection failure

6.63.1.4 test_socket_connect_success()

```
test_socket.TestSocket.test_socket_connect_success (
              self,
              mock_socket)
```

Test successful socket connection

6.63.1.5 test_socket_different_hosts_and_ports()

```
test_socket.TestSocket.test_socket_different_hosts_and_ports (
              self)
```

Test socket creation with different hosts and ports

6.63.1.6 test_socket_init()

```
test_socket.TestSocket.test_socket_init (
              self)
```

Test socket initialization

6.63.1.7 test_socket_receive_connection_closed()

```
test_socket.TestSocket.test_socket_receive_connection_closed (
              self,
              mock_socket)
```

Test handling closed connection during receive

6.63.1.8 test_socket_receive_multiple_messages_in_buffer()

```
test_socket.TestSocket.test_socket_receive_multiple_messages_in_buffer (
              self,
              mock_socket)
```

Test receiving multiple messages in one recv call

6.63.1.9 test_socket_receive_partial_messages()

```
test_socket.TestSocket.test_socket_receive_partial_messages (
              self,
              mock_socket)
```

Test receiving message in multiple parts

6.63.1.10 test_socket_receive_single_message()

```
test_socket.TestSocket.test_socket_receive_single_message (
              self,
              mock_socket)
```

Test receiving a single complete message

6.63.1.11 test_socket_receive_timeout()

```
test_socket.TestSocket.test_socket_receive_timeout (
              self,
              mock_socket)
```

Test socket receive timeout

6.63.1.12 test_socket_receive_unicode()

```
test_socket.TestSocket.test_socket_receive_unicode (
              self,
              mock_socket)
```

Test receiving unicode messages

6.63.1.13 test_socket_send_success()

```
test_socket.TestSocket.test_socket_send_success (
              self,
              mock_socket)
```

Test successful message sending

6.63.1.14 test_socket_send_unicode()

```
test_socket.TestSocket.test_socket_send_unicode (
              self,
              mock_socket)
```

Test sending unicode messages

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

- tests/unit/ai/Communication/test_socket.py

6.64 zappy::structs::Tile Struct Reference

Public Member Functions

- **Tile** (int x=0, int y=0, int food=0, int linemate=0, int deraumere=0, int sibur=0, int mendiane=0, int phiras=0, int thystame=0)

Public Attributes

- int **x**
- int **y**
- int **food**
- int **linemate**
- int **deraumere**
- int **sibur**
- int **mendiane**
- int **phiras**
- int **thystame**

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

- gui/src/Utils/Constants.hpp

6.65 tiles_s Struct Reference

Public Attributes

- int **x**
- int **y**

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

- server/include/algo.h

Chapter 7

File Documentation

7.1 CLI.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** CLI
00006 */
00007
00008 #ifndef CLI_HPP_
00009 #define CLI_HPP_
00010
00011 #include "../Utils/Constants.hpp"
00012
00013 class CLI {
00014     public:
00015         CLI(int ac, const char *const *av);
00016         ~CLI();
00017
00018         zappy::structs::Config parseArguments(int ac, const char *const *av) const;
00019
00020     private:
00021         int _ac;
00022         const char *const *_av;
00023
00024         bool hasCorrectNumberOfArguments(int ac) const;
00025         int parsePort(const char *portStr) const;
00026         std::string parseHostname(const char *hostnameStr) const;
00027         void validateConfig(bool portFound, bool hostFound) const;
00028 };
00029
00030 #endif /* !CLI_HPP_ */
```

7.2 Client.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** Client
00006 */
00007
00008 #ifndef CLIENT_HPP_
00009 #define CLIENT_HPP_
00010
00011 #include "../Utils/Constants.hpp"
00012 #include "../Communication/ICommunication.hpp"
00013 #include "../Game/GameInfos.hpp"
00014 #include "../Graphic/GUI.hpp"
00015 #include "MsgHandler.hpp"
00016
00017 #include <memory>
00018
00019 class Client {
00020     public:
00021         Client(int ac, const char *const *av);
```

```
00022         ~Client();
00023
00024     private:
00025         zappy::structs::Config _config;
00026         void initialize(int ac, const char * const *av);
00027
00028         std::shared_ptr<ICommunication> _communication;
00029         std::shared_ptr<GameInfos> _gameInfos;
00030         std::unique_ptr<MsgHandler> _msgHandler;
00031         std::unique_ptr<GUI> _gui;
00032 };
00033
00034 #endif /* !CLIENT_HPP_ */
```

7.3 MsgHandler.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** MsgHandler
00006 */
00007
00008 #ifndef MSGHANDLER_HPP_
00009 #define MSGHANDLER_HPP_
00010
00011 #include "../Game/GameInfos.hpp"
00012 #include "../Communication/ICommunication.hpp"
00013 #include "../Utils/Constants.hpp"
00014
00015 #include <memory>
00016 #include <map>
00017 #include <functional>
00018 #include <thread>
00019 #include <mutex>
00020 #include <atomic>
00021 #include <queue>
00022 #include <condition_variable>
00023 #include <string>
00024
00025 class MsgHandler {
00026     public:
00027         MsgHandler(std::shared_ptr<GameInfos> gameInfos, std::shared_ptr<ICommunication>
      communication);
00028         ~MsgHandler();
00029
00030         void start();
00031         void stop();
00032
00033     protected:
00034         void messageLoop();
00035
00036         void handleMessage(const std::string& message);
00037         bool handleWelcomeMessage(const std::string& message);
00038         bool handleMszMessage(const std::string& message);
00039         bool handleBctMessage(const std::string& message);
00040         bool handleTnaMessage(const std::string& message);
00041         bool handlePnwMessage(const std::string& message);
00042         bool handlePpoMessage(const std::string& message);
00043         bool handlePlvMessage(const std::string& message);
00044         bool handlePinMessage(const std::string& message);
00045         bool handlePexMessage(const std::string& message);
00046         bool handlePbcMessage(const std::string& message);
00047         bool handlePicMessage(const std::string& message);
00048         bool handlePieMessage(const std::string& message);
00049         bool handlePfkMessage(const std::string& message);
00050         bool handlePdrMessage(const std::string& message);
00051         bool handlePgtMessage(const std::string& message);
00052         bool handlePdiMessage(const std::string& message);
00053         bool handleEnwMessage(const std::string& message);
00054         bool handleEboMessage(const std::string& message);
00055         bool handleEdiMessage(const std::string& message);
00056         bool handleSgtMessage(const std::string& message);
00057         bool handleSstMessage(const std::string& message);
00058         bool handleSegMessage(const std::string& message);
00059         bool handleSmgMessage(const std::string& message);
00060         bool handleSucMessage(const std::string& message);
00061         bool handleSbpMessage(const std::string& message);
00062
00063     private:
00064         std::thread _thread;
00065         std::atomic<bool> _running;
00066         std::mutex _mutex;
```

```
00067         std::condition_variable _condition;
00068
00069         std::shared_ptr<GameInfos> _gameInfos;
00070         std::shared_ptr<ICommunication> _communication;
00071         std::mutex _gameInfosMutex;
00072
00073         std::map<std::string, std::function<bool(const std::string&)» _messageHandlers;
00074 };
00075
00076 #endif /* !MSGHANDLER_HPP_ */
```

7.4 Communication.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** Communication
00006 */
00007
00008 #ifndef COMMUNICATION_HPP_
00009 #define COMMUNICATION_HPP_
00010
00011 #include <thread>
00012 #include <mutex>
00013 #include <atomic>
00014 #include <condition_variable>
00015 #include <queue>
00016 #include <string>
00017 #include <vector>
00018 #include <sys/socket.h>
00019 #include <netinet/in.h>
00020 #include <arpa/inet.h>
00021 #include <unistd.h>
00022 #include <fcntl.h>
00023 #include <poll.h>
00024 #include <netdb.h>
00025
00026 #include "../Utils/Constants.hpp"
00027 #include "../Exceptions/Exceptions.hpp"
00028 #include "ICommunication.hpp"
00029
00030 class Communication : public ICommunication {
00031     public:
00032         Communication(zappy::structs::Config config);
00033         ~Communication();
00034
00035         void sendMessage(const std::string &message) override;
00036         bool hasMessages() const override;
00037         std::string popMessage() override;
00038         bool isConnected() const override;
00039         void disconnect() override;
00040
00041     private:
00042         void setupConnection();
00043         void createSocket();
00044         void connectToServer();
00045         void setupNonBlocking();
00046
00047         void startCommunicationThread();
00048         void communicationLoop();
00049         bool handlePoll();
00050         void processWrite();
00051         void processRead();
00052
00053         void parseReceivedData();
00054
00055         zappy::structs::Config _config;
00056         std::thread _thread;
00057         std::mutex _mutex;
00058         std::condition_variable _cv;
00059         std::atomic<bool> _running;
00060         std::atomic<bool> _connected;
00061
00062         std::queue<std::string> _outgoingMessages;
00063         std::queue<std::string> _incomingMessages;
00064
00065         std::string _receiveBuffer;
00066         std::string _sendBuffer;
00067
00068         int _socket;
00069         struct pollfd _pollfd;
00070         static const int BUFFER_SIZE = 4096;
```

```
00071         static const int POLL_TIMEOUT = 100;
00072         static const char MESSAGE_DELIMITER = '\n';
00073 };
00074
00075 #endif /* !COMMUNICATION_HPP_ */
```

7.5 ICommunication.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** ICommunication
00006 */
00007
00008 #ifndef ICOMMUNICATION_HPP_
00009 #define ICOMMUNICATION_HPP_
00010
00011 #include <string>
00012
00013 class ICommunication {
00014     public:
00015         virtual ~ICommunication() = default;
00016
00017         virtual void sendMessage(const std::string &message) = 0;
00018         virtual bool hasMessages() const = 0;
00019         virtual std::string popMessage() = 0;
00020         virtual bool isConnected() const = 0;
00021         virtual void disconnect() = 0;
00022 };
00023
00024 #endif /* !ICOMMUNICATION_HPP_ */
```

7.6 Exceptions.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** Exceptions
00006 */
00007
00008 #ifndef EXCEPTIONS_HPP_
00009 #define EXCEPTIONS_HPP_
00010
00011 #include <exception>
00012 #include <string>
00013 #include "../Utils/Constants.hpp"
00014
00015 namespace Exceptions {
00016
00017     // CLI Exceptions
00018     class CLIParsingException : public std::exception {
00019         public:
00020             explicit CLIParsingException(const std::string &message)
00021                 : _message(std::string(colors::T_RED) +
00022                           "CLI Parsing Error: " + message +
00023                           colors::RESET) {}
00024
00025             virtual const char *what() const noexcept override {
00026                 return _message.c_str();
00027             }
00028
00029         private:
00030             std::string _message;
00031     };
00032
00033     class CLIPortException : public CLIParsingException {
00034         public:
00035             explicit CLIPortException(const std::string &message)
00036                 : CLIParsingException(std::string(colors::T_CYAN) +
00037                                       "Port Error: " + message +
00038                                       colors::RESET) {}
00039     };
00040
00041     class CLIHostException : public CLIParsingException {
00042         public:
00043             explicit CLIHostException(const std::string &message)
00044                 : CLIParsingException(std::string(colors::T_CYAN) +
```

```
00045                                   "Hostname Error: " + message +
00046                                   colors::RESET) {}
00047     };
00048
00049     class CLIMissingArgumentException : public CLIParsingException {
00050         public:
00051             explicit CLIMissingArgumentException(const std::string &message)
00052                 : CLIParsingException(std::string(colors::T_CYAN) +
00053                                   "Missing Argument: " + message +
00054                                   colors::RESET) {}
00055     };
00056
00057     class CLIInvalidArgumentException : public CLIParsingException {
00058         public:
00059             explicit CLIInvalidArgumentException(const std::string &message)
00060                 : CLIParsingException(std::string(colors::T_CYAN) +
00061                                   "Invalid Argument: " + message +
00062                                   colors::RESET) {}
00063     };
00064
00065     class NetworkException : public std::exception {
00066         public:
00067             explicit NetworkException(const std::string &message)
00068                 : _message(std::string(colors::T_RED) +
00069                           "Network Error: " + message +
00070                           colors::RESET) {}
00071
00072             virtual const char *what() const noexcept override {
00073                 return _message.c_str();
00074             }
00075
00076         private:
00077             std::string _message;
00078     };
00079
00080     class ConnectionFailedException : public NetworkException {
00081         public:
00082             explicit ConnectionFailedException(const std::string &message)
00083                 : NetworkException(std::string(colors::T_CYAN) +
00084                                   "Connection Failed: " + message +
00085                                   colors::RESET) {}
00086     };
00087
00088     class SocketCreationException : public NetworkException {
00089         public:
00090             explicit SocketCreationException(const std::string &message)
00091                 : NetworkException(std::string(colors::T_CYAN) +
00092                                   "Socket Creation Failed: " + message +
00093                                   colors::RESET) {}
00094     };
00095
00096     class ConnectionTimeoutException : public NetworkException {
00097         public:
00098             explicit ConnectionTimeoutException(const std::string &message)
00099                 : NetworkException(std::string(colors::T_CYAN) +
00100                                   "Connection Timeout: " + message +
00101                                   colors::RESET) {}
00102     };
00103
00104     class SendException : public NetworkException {
00105         public:
00106             explicit SendException(const std::string &message)
00107                 : NetworkException(std::string(colors::T_CYAN) +
00108                                   "Send Error: " + message +
00109                                   colors::RESET) {}
00110     };
00111
00112     class ReceiveException : public NetworkException {
00113         public:
00114             explicit ReceiveException(const std::string &message)
00115                 : NetworkException(std::string(colors::T_CYAN) +
00116                                   "Receive Error: " + message +
00117                                   colors::RESET) {}
00118     };
00119 }
00120
00121 #endif /* !EXCEPTIONS_HPP_ */
```

7.7 GameInfos.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
```

```
00004 ** File description:
00005 ** GameInfos
00006 */
00007
00008 #ifndef GAMEINFOS_HPP_
00009 #define GAMEINFOS_HPP_
00010
00011 #include <utility>
00012 #include <vector>
00013 #include <memory>
00014
00015 #include "../Utils/Constants.hpp"
00016
00017 class GameInfos {
00018     public:
00019         GameInfos();
00020         ~GameInfos();
00021
00022         void setMapSize(int width, int height);
00023         std::pair<int, int> getMapSize() const;
00024
00025         void setTimeUnit(int timeUnit);
00026         int getTimeUnit() const;
00027
00028         void updateTile(const zappy::structs::Tile tile);
00029         const std::vector<zappy::structs::Tile> getTiles() const;
00030         const zappy::structs::Tile getTile(int x, int y) const;
00031
00032         void updateTeamName(const std::string &teamName);
00033         const std::vector<std::string> getTeamNames() const;
00034
00035         void addPlayer(const zappy::structs::Player player);
00036         void updatePlayerPosition(int playerNumber, int x, int y);
00037         void updatePlayerLevel(int playerNumber, int level);
00038         void updatePlayerInventory(int playerNumber, const zappy::structs::Inventory inventory);
00039         void updatePlayerExpulsion(int playerNumber);
00040         void updatePlayerDeath(int playerNumber);
00041         void updatePlayerResourceAction(int playerNumber, int resourceId, bool isCollecting);
00042         void updatePlayerFork(int playerNumber);
00043         const std::vector<zappy::structs::Player> getPlayers() const;
00044
00045         void addPlayerBroadcast(int playerNumber, const std::string &message);
00046         std::vector<std::pair<int, std::string» getPlayersBroadcasting() const;
00047
00048         void addIncantation(const zappy::structs::Incantation incantation);
00049         void removeIncantation(int x, int y, int result);
00050
00051         void addEgg(const zappy::structs::Egg egg);
00052         void updateEggHatched(int eggNumber);
00053         void updateEggDeath(int eggNumber);
00054         const std::vector<zappy::structs::Egg> getEggs() const;
00055
00056         void setGameOver(const std::string &winningTeam);
00057         std::pair<bool, std::string> isGameOver() const;
00058
00059     private:
00060         int _mapWidth;
00061         int _mapHeight;
00062         int _timeUnit;
00063
00064         std::vector<zappy::structs::Tile> _tiles;
00065         std::vector<std::string> _teamNames;
00066         std::vector<zappy::structs::Player> _players;
00067         std::vector<std::pair<int, bool» _playersExpulsing;
00068         std::vector<std::pair<int, std::string» _playersBroadcasting;
00069         std::vector<zappy::structs::Incantation> _incantations;
00070         std::vector<zappy::structs::Egg> _eggs;
00071
00072         bool _gameOver;
00073         std::string _winningTeam;
00074 };
00075
00076 #endif /* !GAMEINFOS_HPP_ */
```

7.8 GUI.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** GUI
00006 */
00007
```

```
00008 #ifndef GUI_HPP_
00009 #define GUI_HPP_
00010
00011 #include "RayLib/RayLib.hpp"
00012
00013 class GUI {
00014     public:
00015         GUI();
00016         ~GUI();
00017
00018         void run();
00019
00020     private:
00021         void updateCamera();
00022         void drawEnvironment();
00023
00024         RayLib _raylib;
00025         bool _isRunning;
00026 };
00027
00028 #endif /* !GUI_HPP_ */
```

7.9 RayLib.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** RayLib
00006 */
00007
00008 #ifndef RAYLIB_HPP_
00009 #define RAYLIB_HPP_
00010
00011 #include <string>
00012 #include "raylib.h"
00013
00014 class RayLib {
00015     public:
00016         RayLib();
00017         ~RayLib();
00018
00019         // Window management methods
00020         void initWindow(int width, int height, const std::string &title);
00021         void closeWindow();
00022         bool windowShouldClose() const;
00023         void beginDrawing();
00024         void endDrawing();
00025         void clearBackground(Color color = WHITE);
00026         bool isWindowReady() const;
00027
00028         // 3D Environment methods
00029         void begin3DMode();
00030         void end3DMode();
00031
00032         // Camera methods
00033         void initCamera();
00034         void setCameraPosition(Vector3 position);
00035         void setCameraTarget(Vector3 target);
00036         void setCameraUp(Vector3 up);
00037         void setCameraFovy(float fovy);
00038         void setCameraProjection(int projection);
00039         void updateCamera(int mode = CAMERA_FREE);
00040         Camera3D getCamera() const;
00041
00042         // 3D Drawing methods
00043         void drawGrid(int slices, float spacing);
00044         void drawCube(Vector3 position, float width, float height, float length, Color color);
00045         void drawCubeWires(Vector3 position, float width, float height, float length, Color color);
00046         void drawSphere(Vector3 position, float radius, Color color);
00047         void drawCylinder(Vector3 position, float radiusTop, float radiusBottom, float height, int
      slices, Color color);
00048         void drawPlane(Vector3 position, Vector2 size, Color color);
00049
00050     protected:
00051     private:
00052         bool _isInitialized;
00053         Camera3D _camera;
00054 };
00055
00056 #endif /* !RAYLIB_HPP_ */
```

7.10 Constants.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** Constants
00006 */
00007
00008 #ifndef CONSTANTS_HPP_
00009 #define CONSTANTS_HPP_
00010
00011 #include <string>
00012 #include <vector>
00013
00014 namespace zappy::constants {
00015
00016     inline const char *USAGE_STRING = "USAGE: ./zappy_gui -p port -h machine\n"
00017                                 "option\t\tdescription\n"
00018                                 "-p port\t\tport number\n"
00019                                 "-h machine\thostname of the server";
00020
00021     inline const int FAILURE_EXIT_CODE = 84;
00022     inline const int SUCCESS_EXIT_CODE = 0;
00023 };
00024
00025 namespace colors {
00026
00027     inline const char *T_BOLD = "\033[1m";
00028     inline const char *T_RED = "\033[1m\033[31m";
00029     inline const char *T_GREEN = "\033[1m\033[32m";
00030     inline const char *T_YELLOW = "\033[1m\033[33m";
00031     inline const char *T_BLUE = "\033[1m\033[34m";
00032     inline const char *T_MAGENTA = "\033[1m\033[35m";
00033     inline const char *T_CYAN = "\033[1m\033[36m";
00034     inline const char *T_WHITE = "\033[1m\033[37m";
00035     inline const char *RESET = "\033[0m";
00036
00037 };
00038
00039 namespace zappy::structs {
00040
00041     struct Config {
00042         int port;
00043         std::string hostname;
00044     };
00045
00046     struct Tile {
00047         int x;
00048         int y;
00049         int food;
00050         int linemate;
00051         int deraumere;
00052         int sibur;
00053         int mendiane;
00054         int phiras;
00055         int thystame;
00056
00057         Tile(int x = 0, int y = 0, int food = 0, int linemate = 0,
00058              int deraumere = 0, int sibur = 0, int mendiane = 0,
00059              int phiras = 0, int thystame = 0)
00060             : x(x), y(y), food(food), linemate(linemate),
00061               deraumere(deraumere), sibur(sibur),
00062               mendiane(mendiane), phiras(phiras), thystame(thystame) {}
00063     };
00064
00065     struct Inventory {
00066         int food;
00067         int linemate;
00068         int deraumere;
00069         int sibur;
00070         int mendiane;
00071         int phiras;
00072         int thystame;
00073
00074         Inventory(int food = 0, int linemate = 0, int deraumere = 0,
00075                   int sibur = 0, int mendiane = 0, int phiras = 0,
00076                   int thystame = 0)
00077             : food(food), linemate(linemate), deraumere(deraumere),
00078               sibur(sibur), mendiane(mendiane), phiras(phiras),
00079               thystame(thystame) {}
00080     };
00081     struct Player {
00082         int number;
00083         int x;
00084         int y;
00085         int orientation;
```

```
00086         int level;
00087         std::string teamName;
00088         struct Inventory inventory;
00089
00090         Player(int number = 0, int x = 0, int y = 0, int orientation = 0,
00091                int level = 1, const std::string &teamName = "",
00092                struct Inventory inventory = Inventory())
00093             : number(number), x(x), y(y), orientation(orientation),
00094               level(level), teamName(teamName), inventory(inventory) {}
00095     };
00096
00097     struct Incantation {
00098         int x;
00099         int y;
00100         int level;
00101         std::vector<int> players;
00102
00103         Incantation(int x = 0, int y = 0, int level = 1,
00104                     const std::vector<int> &players = {})
00105             : x(x), y(y), level(level), players(players) {}
00106     };
00107
00108     struct Egg {
00109         int eggNumber;
00110         int playerNumber;
00111         int x;
00112         int y;
00113         bool hatched;
00114         std::string teamName;
00115
00116         Egg(int eggNumber = 0, int playerNumber = 0, int x = 0, int y = 0,
00117             bool hatched = false, const std::string &teamName = "")
00118             : eggNumber(eggNumber), playerNumber(playerNumber), x(x), y(y),
00119               hatched(hatched), teamName(teamName) {}
00120     };
00121 };
00122
00123 inline const int WINDOW_WIDTH = 1920;
00124 inline const int WINDOW_HEIGHT = 1080;
00125 inline const std::string WINDOW_TITLE = "Zappy GUI";
00126
00127 #endif /* !CONSTANTS_HPP_ */
```

7.11 algo.h

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** algo
00006 */
00007
00008 #ifndef ALGO_H_
00009     #define ALGO_H_
00010
00011 typedef struct tiles_s {
00012     int x;
00013     int y;
00014 } tiles_t;
00015
00016 /* Algo.c */
00017 tiles_t *shuffle_fisher(int width, int heigth);
00018
00019 #endif /* !ALGO_H_ */
```

7.12 buffer.h

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** buffer
00006 */
00007
00008 #include <stddef.h>
00009
00010 #ifndef BUFFER_H_
00011     #define BUFFER_H_
00012
```

```
00013     #define BUFFER_SIZE 1024
00014
00015
00016 typedef struct buffer_s {
00017     char data[BUFFER_SIZE];
00018     int head;
00019     int tail;
00020     int full;
00021 } buffer_t;
00022
00023 /* buffer.c */
00024 int advance(int idx);
00025 void cb_write(buffer_t *cb, char c);
00026 int cb_getline(buffer_t *cb, char *line, int max_len);
00027
00028 #endif /* !BUFFER_H_ */
```

7.13 game.h

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** B-YEP-400-NAN-4-1-zappy-albane.merian
00004 ** File description:
00005 ** game
00006 */
00007
00008 #include <time.h>
00009
00010 #ifndef GAME_H_
00011     #define GAME_H_
00012
00013 /* Definition of the directions */
00014 typedef enum direction_e {
00015     NORTH,
00016     SOUTH,
00017     EAST,
00018     WEST
00019 } direction_t;
00020
00021 /* definintion od the different element on the map */
00022 typedef enum crystal_e {
00023     FOOD,
00024     LINEMATE,
00025     DERAUMERE,
00026     SIBUR,
00027     MENDIANE,
00028     PHIRAS,
00029     THYSTAME
00030 } crystal_t;
00031
00032 /* Struct that "counts" the current stat of the player */
00033 typedef struct lives_s {
00034     int freq;
00035     int nbFood;
00036     time_t startRefresh;
00037     time_t endRefresh;
00038 } lives_t;
00039
00040 /* Struct defining the invetory of the player */
00041 typedef struct inventory_s {
00042     int nbLinemate;
00043     int nbDeraumere;
00044     int nbSibur;
00045     int nbMendiane;
00046     int nbPhiras;
00047     int nbThystame;
00048 } inventory_t;
00049
00050 /* Player struct */
00051 typedef struct player_s {
00052     int id; /* This is equal to the current FD */
00053     int level;
00054     int posX;
00055     int posY;
00056     bool isAlive;
00057     direction_t direction;
00058     inventory_t *inventory;
00059     lives_t *lives;
00060     struct player_s *next;
00061 } player_t;
00062
00063 /* Team Strcut */
00064 typedef struct team_s {
```

```
00065     char *name;
00066     int nbPlayers;
00067     int nbPlayerAlive;
00068     player_t *players;
00069     struct team_s *next;
00070 } team_t;
00071
00072 /* Ressources, and there pos on the map */
00073 typedef struct ressources_s {
00074     crystal_t type;
00075     int posX;
00076     int posY;
00077     struct ressources_s *next;
00078 } ressources_t;
00079
00080 /* Map struct */
00081 typedef struct map_s {
00082     int width;
00083     int heigt;
00084     team_t *teams;
00085     ressources_t *ressources;
00086 } map_t;
00087
00088 #endif /* !GAME_H_ */
```

7.14 zappy.h

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** Zappy
00004 ** File description:
00005 ** Server :: Zappy header
00006 */
00007
00008 #include <stdbool.h>
00009 #include <poll.h>
00010 #include "game.h"
00011
00012 #ifndef ZAPPY_H_
00013     #define ZAPPY_H_
00014
00015 typedef struct params_s {
00016     int port;
00017     int x;
00018     int y;
00019     int nb_team;
00020     char **teams;
00021     int nb_client;
00022     int freq;
00023     bool is_debug;
00024 } params_t;
00025
00026 typedef struct graph_s {
00027     int fd;
00028     struct pollfd *pollfd;
00029 } graph_t;
00030
00031 typedef struct server_s {
00032     int sockfd;
00033     params_t *params;
00034     map_t *map;
00035     graph_t *graph;
00036 } server_t;
00037
00038 typedef struct command_pf_s {
00039     char const *flag;
00040     bool (*checker)(const char *, const char *, params_t *);
00041 } command_pf_t;
00042
00043 /* messages.c */
00044 int helper(void);
00045 void error_message(const char *message);
00046 void printfd(char const *message, int fd);
00047 char *get_message(int fd, server_t *server);
00048
00049 /* checkers.c */
00050 bool check_port(char const *flag, char const *value, params_t *params);
00051 bool check_width(char const *flag, char const *value, params_t *params);
00052 bool check_height(char const *flag, char const *value, params_t *params);
00053 bool check_client(char const *flag, char const *value, params_t *params);
00054 bool check_freq(char const *flag, char const *value, params_t *params);
00055
00056 /* params.c */
```

```
00057 params_t *check_args(int argc, char **argv);
00058 void *free_params(params_t *params);
00059
00060 /* server.c */
00061 server_t *init_server(int argc, char **argv);
00062 void *free_server(server_t *server);
00063
00064 /* protocol.c */
00065 int start_protocol(server_t *server);
00066
00067 /* client.c */
00068 bool valid_team_name(const char *team_name, params_t *params);
00069 bool graphic(const char *team_name, int fd, server_t *server);
00070
00071 /* init_map.c */
00072 void inti_map(server_t *server);
00073
00074 /* free server  */
00075 void *free_server(server_t *server);
00076 void *free_params(params_t *params);
00077 #endif /* !ZAPPY_H_ */
```