

## Zappy GUI

Generated by Doxygen 1.9.6



<b>1 Zappy - GUI</b>	<b>1</b>
1.1 Project Overview	1
1.1.1 Features	1
1.2 Installation	1
1.2.1 Installation Steps	2
1.2.1.1 Update Your System	2
1.2.1.2 Install Necessary Dependencies	2
1.3 Usage	2
1.4 Development	2
1.4.1 Project Structure	2
1.4.2 Running Tests	2
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 Gui::Errors::AError Class Reference	9
5.1.1 Detailed Description	10
5.1.2 Member Function Documentation	10
5.1.2.1 what()	10
5.2 Gui::AEvent Class Reference	10
5.3 Gui::AGUIUpdater Class Reference	10
5.3.1 Constructor & Destructor Documentation	11
5.3.1.1 AGUIUpdater()	11
5.3.2 Member Function Documentation	11
5.3.2.1 update()	11
5.4 Gui::AHud Class Reference	12
5.4.1 Constructor & Destructor Documentation	13
5.4.1.1 ~AHud()	13
5.4.2 Member Function Documentation	13
5.4.2.1 display()	13
5.4.2.2 getType()	13
5.4.2.3 setPlayer()	13
5.4.2.4 setTile()	14
5.4.2.5 setType()	14
5.5 Gui::ANetwork Class Reference	14
5.5.1 Constructor & Destructor Documentation	16
5.5.1.1 ANetwork()	16

5.5.1.2 ~ANetwork()	16
5.5.2 Member Function Documentation	16
5.5.2.1 connectToServer()	16
5.5.2.2 getBuffer()	17
5.5.2.3 getHostName()	17
5.5.2.4 getPort()	17
5.5.2.5 listenServer()	18
5.5.2.6 sendMessageServer()	18
5.5.2.7 setHostName()	18
5.5.2.8 setPort()	19
5.6 Gui::Decoration Class Reference	19
5.6.1 Constructor & Destructor Documentation	19
5.6.1.1 Decoration()	19
5.6.1.2 ~Decoration()	20
5.6.2 Member Function Documentation	20
5.6.2.1 display()	20
5.6.2.2 getGenerationItem()	20
5.7 Gui::Egg Class Reference	21
5.7.1 Constructor & Destructor Documentation	21
5.7.1.1 Egg()	21
5.7.1.2 ~Egg()	22
5.7.2 Member Function Documentation	22
5.7.2.1 getId()	22
5.7.2.2 getPosition()	22
5.7.2.3 getState()	23
5.7.2.4 getTeam()	23
5.7.2.5 setId()	23
5.7.2.6 setPosition()	23
5.7.2.7 setState()	24
5.7.2.8 setTeam()	24
5.8 Gui::Engine Class Reference	24
5.8.1 Constructor & Destructor Documentation	24
5.8.1.1 Engine()	25
5.8.1.2 ~Engine()	25
5.8.2 Member Function Documentation	25
5.8.2.1 run()	25
5.9 Gui::Errors::Error Class Reference	25
5.9.1 Detailed Description	26
5.10 Gui::Event Class Reference	26
5.10.1 Constructor & Destructor Documentation	27
5.10.1.1 Event()	27
5.10.1.2 ~Event()	28

5.10.2 Member Function Documentation	28
5.10.2.1 listen()	28
5.11 Gui::GameData Class Reference	28
5.11.1 Constructor & Destructor Documentation	30
5.11.1.1 GameData()	30
5.11.1.2 ~GameData()	30
5.11.2 Member Function Documentation	30
5.11.2.1 addPlayerToTeam()	30
5.11.2.2 addServerEgg()	31
5.11.2.3 addTeam() [1/2]	31
5.11.2.4 addTeam() [2/2]	31
5.11.2.5 getEndMessage()	31
5.11.2.6 getIsEndGame()	32
5.11.2.7 getLastError()	32
5.11.2.8 getLastTick()	32
5.11.2.9 getLastTickMctCommand()	32
5.11.2.10 getMap()	33
5.11.2.11 getMapSize()	33
5.11.2.12 getNbBCTCommandReceived()	33
5.11.2.13 getPlayer()	33
5.11.2.14 getServerEggs()	34
5.11.2.15 getServerError()	34
5.11.2.16 getServerTick()	34
5.11.2.17 getTeam()	34
5.11.2.18 getTeamById()	35
5.11.2.19 getTeams()	35
5.11.2.20 getTile()	35
5.11.2.21 getTimeUnitFromServer()	36
5.11.2.22 removeServerEgg()	36
5.11.2.23 restartLastTick()	36
5.11.2.24 restartLastTickMctCommand()	36
5.11.2.25 setAnimsCount()	37
5.11.2.26 setEggModel()	37
5.11.2.27 setEndMessage()	37
5.11.2.28 setIsEndGame()	37
5.11.2.29 setLastError()	38
5.11.2.30 setMap()	38
5.11.2.31 setMapSize()	38
5.11.2.32 setNbBCTCommandReceived()	39
5.11.2.33 setPlayerModel()	39
5.11.2.34 setPlayerModelAnimation()	39
5.11.2.35 setServerError()	39

5.11.2.36 setServerTick()	40
5.11.2.37 setTile()	40
5.11.2.38 setTimeUnitFromServer()	40
5.12 Gui::Errors::GuiGameDataException Class Reference	41
5.12.1 Detailed Description	41
5.12.2 Constructor & Destructor Documentation	41
5.12.2.1 GuiGameDataException()	41
5.13 Gui::GUIUpdater Class Reference	42
5.13.1 Constructor & Destructor Documentation	43
5.13.1.1 GUIUpdater()	43
5.13.2 Member Function Documentation	43
5.13.2.1 update()	43
5.14 Gui::Errors::GuiUpdaterException Class Reference	43
5.14.1 Detailed Description	44
5.14.2 Constructor & Destructor Documentation	44
5.14.2.1 GuiUpdaterException()	44
5.15 Gui::HudEnd Class Reference	45
5.15.1 Constructor & Destructor Documentation	46
5.15.1.1 HudEnd()	46
5.15.1.2 ~HudEnd()	46
5.15.2 Member Function Documentation	46
5.15.2.1 display()	47
5.16 Gui::HudGame Class Reference	47
5.16.1 Constructor & Destructor Documentation	48
5.16.1.1 HudGame()	48
5.16.1.2 ~HudGame()	49
5.16.2 Member Function Documentation	49
5.16.2.1 display()	49
5.17 Gui::HudHelp Class Reference	49
5.17.1 Constructor & Destructor Documentation	51
5.17.1.1 HudHelp()	51
5.17.1.2 ~HudHelp()	51
5.17.2 Member Function Documentation	51
5.17.2.1 display()	51
5.18 Gui::HudPlayer Class Reference	51
5.18.1 Constructor & Destructor Documentation	53
5.18.1.1 HudPlayer()	53
5.18.1.2 ~HudPlayer()	53
5.18.2 Member Function Documentation	53
5.18.2.1 display()	53
5.19 Gui::HudTile Class Reference	54
5.19.1 Constructor & Destructor Documentation	55

5.19.1.1 HudTile()	55
5.19.1.2 ~HudTile()	55
5.19.2 Member Function Documentation	55
5.19.2.1 display()	56
5.19.2.2 displayNbEggs()	56
5.19.2.3 displayNbPlayers()	56
5.20 Gui::Errors::IError Class Reference	56
5.20.1 Member Function Documentation	56
5.20.1.1 what()	57
5.21 Gui::IEvent Class Reference	57
5.21.1 Constructor & Destructor Documentation	57
5.21.1.1 IEvent()	57
5.21.1.2 ~IEvent()	58
5.21.2 Member Function Documentation	58
5.21.2.1 listen()	58
5.21.2.2 setGameData()	58
5.21.2.3 setRender()	58
5.22 Gui::IGUIUpdater Class Reference	59
5.22.1 Member Function Documentation	59
5.22.1.1 update()	59
5.23 Gui::IHud Class Reference	59
5.23.1 Member Enumeration Documentation	60
5.23.1.1 TypeScene	60
5.23.2 Constructor & Destructor Documentation	60
5.23.2.1 ~IHud()	60
5.23.3 Member Function Documentation	60
5.23.3.1 display()	60
5.23.3.2 getType()	61
5.23.3.3 setPlayer()	61
5.23.3.4 setTile()	61
5.23.3.5 setType()	61
5.24 Gui::INetwork Class Reference	62
5.24.1 Constructor & Destructor Documentation	62
5.24.1.1 ~INetwork()	63
5.24.2 Member Function Documentation	63
5.24.2.1 connectToServer()	63
5.24.2.2 getBuffer()	63
5.24.2.3 getHostName()	63
5.24.2.4 getPort()	64
5.24.2.5 listenServer()	64
5.24.2.6 sendMessageServer()	64
5.24.2.7 setHostName()	64

5.24.2.8 setPort()	65
5.25 Gui::Inventory Class Reference	65
5.25.1 Constructor & Destructor Documentation	66
5.25.1.1 Inventory() [1/2]	66
5.25.1.2 Inventory() [2/2]	67
5.25.1.3 ~Inventory()	67
5.25.2 Member Function Documentation	67
5.25.2.1 addResource()	67
5.25.2.2 getDeraumere()	68
5.25.2.3 getFood()	68
5.25.2.4 getLinemate()	68
5.25.2.5 getMendiane()	68
5.25.2.6 getPhiras()	69
5.25.2.7 getRessources()	69
5.25.2.8 getSibur()	69
5.25.2.9 getThystame()	69
5.25.2.10 removeResource()	69
5.25.2.11 setDeraumere()	70
5.25.2.12 setFood()	70
5.25.2.13 setLinemate()	70
5.25.2.14 setMendiane()	71
5.25.2.15 setPhiras()	71
5.25.2.16 setRessources()	71
5.25.2.17 setSibur()	71
5.25.2.18 setThystame()	72
5.26 Gui::IServerParser Class Reference	72
5.26.1 Member Function Documentation	72
5.26.1.1 parse()	72
5.27 Gui::Network Class Reference	73
5.27.1 Constructor & Destructor Documentation	74
5.27.1.1 Network()	74
5.27.1.2 ~Network()	75
5.27.2 Member Function Documentation	75
5.27.2.1 connectToServer()	75
5.27.2.2 listenServer()	75
5.27.2.3 sendMessageServer()	75
5.28 Gui::Errors::NetworkException Class Reference	76
5.28.1 Detailed Description	77
5.28.2 Constructor & Destructor Documentation	77
5.28.2.1 NetworkException()	77
5.29 Gui::ParseCommandLine Class Reference	77
5.29.1 Constructor & Destructor Documentation	77



5.29.1.1 ParseCommandLine()	77
5.29.1.2 ~ParseCommandLine()	78
5.29.2 Member Function Documentation	78
5.29.2.1 getHostName()	78
5.29.2.2 getPort()	78
5.29.2.3 parseFlags()	78
5.30 Gui::Errors::ParseCommandLineException Class Reference	79
5.30.1 Detailed Description	80
5.30.2 Constructor & Destructor Documentation	80
5.30.2.1 ParseCommandLineException()	80
5.31 Gui::Player Class Reference	80
5.31.1 Constructor & Destructor Documentation	81
5.31.1.1 Player()	82
5.31.1.2 ~Player()	82
5.31.2 Member Function Documentation	82
5.31.2.1 getAnimationTimeEllapsed()	82
5.31.2.2 getBroadcast()	82
5.31.2.3 getCenterPosition()	83
5.31.2.4 getCurrentFrame()	83
5.31.2.5 getId()	83
5.31.2.6 getLevel()	83
5.31.2.7 getOrientation()	84
5.31.2.8 getPosition()	84
5.31.2.9 getPosition3D()	84
5.31.2.10 getRotationFromOrientation()	84
5.31.2.11 getState()	85
5.31.2.12 getTeam()	85
5.31.2.13 restartAnimationTimeEllapsed()	85
5.31.2.14 setBroadcast()	85
5.31.2.15 setCurrentFrame()	86
5.31.2.16 setId()	86
5.31.2.17 setLevel()	86
5.31.2.18 setOrientation()	86
5.31.2.19 setPosition()	87
5.31.2.20 setPosition3D()	87
5.31.2.21 setState()	87
5.31.2.22 setTeam()	88
5.31.3 Member Data Documentation	88
5.31.3.1 inventory	88
5.32 Gui::Render Class Reference	88
5.32.1 Constructor & Destructor Documentation	89
5.32.1.1 Render()	90

5.32.1.2 ~Render()	90
5.32.2 Member Function Documentation	90
5.32.2.1 changePlayerPOV()	90
5.32.2.2 changePOVToFirstPerson()	90
5.32.2.3 changePOVToSecondPerson()	91
5.32.2.4 changePOVToThirdPerson()	91
5.32.2.5 draw()	91
5.32.2.6 drawEnd()	91
5.32.2.7 getCamera()	92
5.32.2.8 getCameraPlayerPov()	92
5.32.2.9 getCameraTile()	92
5.32.2.10 getCameraType()	92
5.32.2.11 getHelpMenu()	93
5.32.2.12 getIsDebug()	93
5.32.2.13 getPlayerVision()	93
5.32.2.14 getRenderDistance()	93
5.32.2.15 getTileModel()	94
5.32.2.16 getTimeUnit()	94
5.32.2.17 isCameraInPlayerPov()	94
5.32.2.18 isOpen()	94
5.32.2.19 setCameraPlayerPov()	94
5.32.2.20 setCameraTile()	95
5.32.2.21 setCameraType()	95
5.32.2.22 setHelpMenu()	95
5.32.2.23 setIsDebug()	96
5.32.2.24 setPlayerPov()	96
5.32.2.25 setPlayerVision()	96
5.32.2.26 setRenderDistance()	96
5.32.2.27 setTimeUnit()	98
5.33 Gui::ServerParser Class Reference	98
5.33.1 Constructor & Destructor Documentation	99
5.33.1.1 ServerParser()	99
5.33.1.2 ~ServerParser()	99
5.33.2 Member Function Documentation	99
5.33.2.1 parse()	99
5.34 Gui::Errors::ServerParserException Class Reference	100
5.34.1 Detailed Description	100
5.34.2 Constructor & Destructor Documentation	100
5.34.2.1 ServerParserException()	100
5.35 Gui::Team Class Reference	101
5.35.1 Constructor & Destructor Documentation	102
5.35.1.1 Team()	102

5.35.1.2 ~Team()	102
5.35.2 Member Function Documentation	102
5.35.2.1 addEgg()	103
5.35.2.2 addPlayer()	103
5.35.2.3 getEgg()	103
5.35.2.4 getEggModel()	103
5.35.2.5 getEggs()	104
5.35.2.6 getName()	104
5.35.2.7 getPlayer()	104
5.35.2.8 getPlayerBoundingBoxes()	105
5.35.2.9 getPlayerColor()	105
5.35.2.10 getPlayerModel()	105
5.35.2.11 getPlayerModelAnimation()	105
5.35.2.12 getPlayerModelHitbox()	105
5.35.2.13 getPlayerPositionIn3DSpace()	106
5.35.2.14 getPlayers()	106
5.35.2.15 isPlayerHit()	106
5.35.2.16 removeEgg()	107
5.35.2.17 removePlayer()	107
5.35.2.18 setEggModelPath()	108
5.35.2.19 setName()	108
5.35.2.20 setPlayerModelPath()	108
5.36 Gui::Tile Class Reference	108
5.36.1 Constructor & Destructor Documentation	109
5.36.1.1 Tile() [1/2]	109
5.36.1.2 Tile() [2/2]	109
5.36.1.3 ~Tile()	110
5.36.2 Member Function Documentation	110
5.36.2.1 getPosition()	110
5.36.2.2 getPositionIn3DSpace()	110
5.36.2.3 getTileBoundingBoxes()	110
5.36.2.4 getTileModelHitbox()	111
5.36.2.5 isTileHit()	111
5.36.2.6 setPosition()	112
5.36.3 Member Data Documentation	112
5.36.3.1 inventory	112
5.37 Gui::UserCamera Class Reference	112
5.37.1 Constructor & Destructor Documentation	113
5.37.1.1 UserCamera()	113
5.37.1.2 ~UserCamera()	113
5.37.2 Member Function Documentation	114
5.37.2.1 getCamera()	114

5.37.2.2 getFovy()	114
5.37.2.3 getPlayerId()	114
5.37.2.4 getPosition()	114
5.37.2.5 getTarget()	115
5.37.2.6 getTilePos()	115
5.37.2.7 getType()	115
5.37.2.8 getUp()	115
5.37.2.9 isPlayerPov()	116
5.37.2.10 isPlayerVision()	116
5.37.2.11 setFovy()	116
5.37.2.12 setPlayerId()	116
5.37.2.13 setPlayerVision()	118
5.37.2.14 setPosition()	118
5.37.2.15 setTarget()	118
5.37.2.16 setTilePos()	119
5.37.2.17 setType()	119
5.37.2.18 setUp()	119

## 6 File Documentation 121

6.1 Assets.hpp	121
6.2 Colors.hpp	123
6.3 Config.hpp	123
6.4 Engine.hpp	124
6.5 AError.hpp	124
6.6 Error.hpp	125
6.7 IError.hpp	126
6.8 AEvent.hpp	126
6.9 Event.hpp	126
6.10 IEvent.hpp	128
6.11 Egg.hpp	128
6.12 GameData.hpp	129
6.13 Inventory.hpp	131
6.14 Player.hpp	132
6.15 Team.hpp	133
6.16 Tile.hpp	134
6.17 AGUIUpdater.hpp	134
6.18 GUIUpdater.hpp	135
6.19 IGUIUpdater.hpp	136
6.20 AHud.hpp	137
6.21 HudEnd.hpp	137
6.22 HudGame.hpp	138
6.23 HudHelp.hpp	138

---

6.24 HudPlayer.hpp . . . . .	139
6.25 HudTile.hpp . . . . .	139
6.26 IHud.hpp . . . . .	140
6.27 ANetwork.hpp . . . . .	141
6.28 INetwork.hpp . . . . .	141
6.29 Network.hpp . . . . .	142
6.30 IServerParser.hpp . . . . .	142
6.31 ParseCommandLine.hpp . . . . .	143
6.32 ServerParser.hpp . . . . .	143
6.33 Decoration.hpp . . . . .	144
6.34 Render.hpp . . . . .	145
6.35 UserCamera.hpp . . . . .	147
6.36 Types.hpp . . . . .	148
<b>Index</b>	<b>149</b>



# Chapter 1

## Zappy - GUI

Welcome to the GUI component of the Zappy project! This document will guide you through the setup, development, and contribution processes for the GUI part of the project.

### 1.1 Project Overview

The Zappy project is a multiplayer, real-time strategy game developed as part of the Epitech Secondary year curriculum. The GUI component provides a graphical interface for players to interact with the game, visualize game states, and issue commands.

#### 1.1.1 Features

- Real-time updates of game state
- Interactive map and player controls
- Visualization of game statistics and resources

### 1.2 Installation

To get started with the GUI, follow these steps:

#### 1. Clone the repository:

```
git clone https://github.com/yourusername/zappy.git
cd Zappy/gui
```

#### 1. Clone the repository:

Ensure you have the necessary tools and libraries installed, such as `raylib` and a C++ compiler.

## 1.2.1 Installation Steps

Make sure you have an up-to-date version of Fedora and that you have administrator privileges (sudo).

### 1.2.1.1 Update Your System

Before installing new packages, it is recommended to update your system:

```
sudo dnf update
```

### 1.2.1.2 Install Necessary Dependencies

Raylib requires certain libraries to function properly. Install them with the following command:

```
sudo dnf install alsa-lib-devel mesa-libGL-devel libX11-devel libXrandr-devel libXi-devel libXcursor-devel  
libXinerama-devel libatomic
```

Fedora offers Raylib directly in its repositories. You can install it using dnf:

```
sudo dnf install raylib-devel
```

**To make it easier, you can install everything in one command:**

```
make install-deps
```

#### 1. Build the project

At the root of the Zappy project:

```
make
```

## 1.3 Usage

To run the GUI application:

1. Ensure that the server component of Zappy is running and accessible.
2. Start the GUI application:

```
./zappy_gui -p Server_port -h Server_port
```

1. Connect to the server using the GUI and start interacting with the game.

## 1.4 Development

### 1.4.1 Project Structure

- src/ - Source code for the GUI application
- assets/ - Images, icons, and other graphical assets
- include/ - Header files
- ..tests/gui/tests/ - Unit and integration tests

### 1.4.2 Running Tests

To run the tests at the root of the Zappy project:

```
make tests_run
```



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

Gui::Decoration . . . . .	19
Gui::Egg . . . . .	21
Gui::Engine . . . . .	24
std::exception	
Gui::Errors::IError . . . . .	56
Gui::Errors::AError . . . . .	9
Gui::Errors::Error . . . . .	25
Gui::Errors::GuiGameDataException . . . . .	41
Gui::Errors::GuiUpdaterException . . . . .	43
Gui::Errors::NetworkException . . . . .	76
Gui::Errors::ParseCommandLineException . . . . .	79
Gui::Errors::ServerParserException . . . . .	100
Gui::GameData . . . . .	28
Gui::IEvent . . . . .	57
Gui::AEvent . . . . .	10
Gui::Event . . . . .	26
Gui::IGUIUpdater . . . . .	59
Gui::AGUIUpdater . . . . .	10
Gui::GUIUpdater . . . . .	42
Gui::IHud . . . . .	59
Gui::AHud . . . . .	12
Gui::HudEnd . . . . .	45
Gui::HudGame . . . . .	47
Gui::HudHelp . . . . .	49
Gui::HudPlayer . . . . .	51
Gui::HudTile . . . . .	54
Gui::INetwork . . . . .	62
Gui::ANetwork . . . . .	14
Gui::Network . . . . .	73
Gui::Inventory . . . . .	65
Gui::IServerParser . . . . .	72
Gui::ServerParser . . . . .	98
Gui::ParseCommandLine . . . . .	77

Gui::Player . . . . .	80
Gui::Render . . . . .	88
Gui::Team . . . . .	101
Gui::Tile . . . . .	108
Gui::UserCamera . . . . .	112

## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Gui::Errors::AError</a>	
Base class for custom error types. This class is derived from the <a href="#">IError</a> interface and provides a common base for custom error types. It contains a protected member <code>_message</code> to store the error message	9
<a href="#">Gui::AEvent</a>	10
<a href="#">Gui::AGUIUpdater</a>	10
<a href="#">Gui::AHud</a>	12
<a href="#">Gui::ANetwork</a>	14
<a href="#">Gui::Decoration</a>	19
<a href="#">Gui::Egg</a>	21
<a href="#">Gui::Engine</a>	24
<a href="#">Gui::Errors::Error</a>	
Base class for argument-related errors	25
<a href="#">Gui::Event</a>	26
<a href="#">Gui::GameData</a>	28
<a href="#">Gui::Errors::GuiGameDataException</a>	
Error class for <a href="#">GameData</a> errors	41
<a href="#">Gui::GUIUpdater</a>	42
<a href="#">Gui::Errors::GuiUpdaterException</a>	
Error class for <a href="#">GUIUpdater</a> errors	43
<a href="#">Gui::HudEnd</a>	45
<a href="#">Gui::HudGame</a>	47
<a href="#">Gui::HudHelp</a>	49
<a href="#">Gui::HudPlayer</a>	51
<a href="#">Gui::HudTile</a>	54
<a href="#">Gui::Errors::IError</a>	56
<a href="#">Gui::IEvent</a>	57
<a href="#">Gui::IGUIUpdater</a>	59
<a href="#">Gui::IHud</a>	59
<a href="#">Gui::INetwork</a>	62
<a href="#">Gui::Inventory</a>	65
<a href="#">Gui::IServerParser</a>	72
<a href="#">Gui::Network</a>	73
<a href="#">Gui::Errors::NetworkException</a>	
Error class for network errors	76

<a href="#">Gui::ParseCommandLine</a> . . . . .	77
<a href="#">Gui::Errors::ParseCommandLineException</a>	
Error class for parseCommandLine errors . . . . .	79
<a href="#">Gui::Player</a> . . . . .	80
<a href="#">Gui::Render</a> . . . . .	88
<a href="#">Gui::ServerParser</a> . . . . .	98
<a href="#">Gui::Errors::ServerParserException</a>	
Error class for network errors . . . . .	100
<a href="#">Gui::Team</a> . . . . .	101
<a href="#">Gui::Tile</a> . . . . .	108
<a href="#">Gui::UserCamera</a> . . . . .	112

## Chapter 4

# File Index

### 4.1 File List

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

/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Assets.hpp	121
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Colors.hpp	123
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Config.hpp	123
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Types.hpp	148
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Engine/Engine.hpp	124
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/AError.hpp	124
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp	125
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/IError.hpp	126
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/AEvent.hpp	126
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/Event.hpp	126
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/IEvent.hpp	128
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Egg.hpp	128
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/GameData.hpp	129
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Inventory.hpp	131
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Player.hpp	132
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Team.hpp	133
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Tile.hpp	134
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/AGUIUpdater.hpp	134
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/GUIUpdater.hpp	135
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/IGUIUpdater.hpp	136
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/AHud.hpp	137
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudEnd.hpp	137
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp	138
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudHelp.hpp	138
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudPlayer.hpp	139
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudTile.hpp	139
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/IHud.hpp	140
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/ANetwork.hpp	141
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/INetwork.hpp	141
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/Network.hpp	142
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/IServerParser.hpp	142
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ParseCommandLine.hpp	143
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ServerParser.hpp	143
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Decoration.hpp	144
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Render.hpp	145
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/UserCamera.hpp	147



## Chapter 5

# Class Documentation

### 5.1 Gui::Errors::AError Class Reference

Base class for custom error types. This class is derived from the [IError](#) interface and provides a common base for custom error types. It contains a protected member `_message` to store the error message.

```
#include <AError.hpp>
```

Inheritance diagram for Gui::Errors::AError:

Collaboration diagram for Gui::Errors::AError:

#### Public Member Functions

- `~AError ()` override=default  
*Destructor.*
- `const char * what ()` const noexcept override  
*Returns the error message.*

#### Public Member Functions inherited from [Gui::Errors::IError](#)

- `virtual ~IError ()`=default  
*Destructor for [IError](#).*
- `virtual const char * what ()` const noexcept=0  
*Returns a C-style string describing the error. This function must be implemented by derived classes to provide a description of the error. The returned string should be null-terminated.*

#### Protected Attributes

- `std::string _message`  
*The error message.*

### 5.1.1 Detailed Description

Base class for custom error types. This class is derived from the [IError](#) interface and provides a common base for custom error types. It contains a protected member `_message` to store the error message.

### 5.1.2 Member Function Documentation

#### 5.1.2.1 `what()`

```
const char * Gui::Errors::AError::what ( ) const [override], [virtual], [noexcept]
```

Returns the error message.

#### Returns

A pointer to a constant character string representing the error message.

Implements [Gui::Errors::IError](#).

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

- `/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/AError.hpp`
- `/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Error/Error.cpp`

## 5.2 Gui::AEvent Class Reference

Inheritance diagram for Gui::AEvent:

## 5.3 Gui::AGUIUpdater Class Reference

Inheritance diagram for Gui::AGUIUpdater:

Collaboration diagram for Gui::AGUIUpdater:

### Public Member Functions

- [AGUIUpdater](#) (std::shared\_ptr< [GameData](#) > gameData, std::shared\_ptr< [INetwork](#) > network)  
*Construct a new [AGUIUpdater](#) object.*
- `~AGUIUpdater ()=default`  
*Destroy the [AGUIUpdater](#) object.*
- void [update](#) (const std::string &command, const std::vector< std::string > &data) override=0  
*Update the GUI [GameData](#).*



**Public Member Functions inherited from Gui::IGUIUpdater**

- virtual `~IGUIUpdater()`=default  
*Destroy the IGUIUpdater object.*
- virtual void `update` (const std::string &command, const std::vector< std::string > &data)=0  
*Update the GUI GameData.*

**Protected Attributes**

- std::shared\_ptr< [GameData](#) > `_gameData`  
*The GUI GameData to update.*
- std::shared\_ptr< [INetwork](#) > `_network`  
*The network to send commands to the server.*

**5.3.1 Constructor & Destructor Documentation****5.3.1.1 AGUIUpdater()**

```
Gui::AGUIUpdater::AGUIUpdater (
    std::shared_ptr< GameData > gameData,
    std::shared_ptr< INetwork > network )
```

Construct a new [AGUIUpdater](#) object.

**Parameters**

<i>gameData</i>	The GUI <a href="#">GameData</a> to update.
<i>network</i>	The network to send commands to the server.

**5.3.2 Member Function Documentation****5.3.2.1 update()**

```
void Gui::AGUIUpdater::update (
    const std::string & command,
    const std::vector< std::string > & data ) [override], [pure virtual]
```

Update the GUI [GameData](#).

Implements [Gui::IGUIUpdater](#).

Implemented in [Gui::GUIUpdater](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/AGUIUpdater.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/GUIUpdater/AGUIUpdater.cpp

## 5.4 Gui::AHud Class Reference

Inheritance diagram for Gui::AHud:

Collaboration diagram for Gui::AHud:

### Public Member Functions

- [~AHud](#) ()=default  
*Destroy the AHud object.*
- virtual void [display](#) ()=0  
*Display Hud.*
- void [setPlayer](#) (std::shared\_ptr< [Player](#) > player)  
*Set the Player object.*
- void [setTile](#) (std::shared\_ptr< [Tile](#) > tile)  
*Set the Tile object.*
- [TypeScene](#) [getType](#) () const  
*Get the Type object.*
- void [setType](#) ([TypeScene](#) type)  
*Set the Type object.*

### Public Member Functions inherited from Gui::IHud

- virtual [~IHud](#) ()=default  
*Destroy the IHud object.*
- virtual void [display](#) ()=0  
*Display the Hud.*
- virtual void [setPlayer](#) (std::shared\_ptr< [Player](#) > player)=0  
*Set the Player object.*
- virtual void [setTile](#) (std::shared\_ptr< [Tile](#) > tile)=0  
*Set the Tile object.*
- virtual [TypeScene](#) [getType](#) () const =0  
*Get the Type object.*
- virtual void [setType](#) ([TypeScene](#) type)=0  
*Set the Type object.*

### Protected Attributes

- [TypeScene](#) [\\_typeScene](#)  
*Type of the scene.*
- std::shared\_ptr< [GameData](#) > [\\_gameData](#)  
*GameData class.*
- std::shared\_ptr< [Player](#) > [\\_player](#)  
*Player to display hud.*
- std::shared\_ptr< [Tile](#) > [\\_tile](#)  
*Tile to display hud.*

## Additional Inherited Members

### Public Types inherited from [Gui::IHud](#)

- enum [TypeScene](#) {  
    **GAME** , **POV\_PLAYER** , **END\_GAME** , **TILE** ,  
    **HELP\_TEXT** , **HELP\_MENU** , **END** }

*Hud enum for the different scenes.*

## 5.4.1 Constructor & Destructor Documentation

### 5.4.1.1 ~AHud()

```
Gui::AHud::~~AHud ( ) [default]
```

Destroy the [AHud](#) object.

## 5.4.2 Member Function Documentation

### 5.4.2.1 display()

```
virtual void Gui::AHud::display ( ) [pure virtual]
```

Display Hud.

Implements [Gui::IHud](#).

Implemented in [Gui::HudEnd](#), [Gui::HudGame](#), [Gui::HudHelp](#), [Gui::HudPlayer](#), and [Gui::HudTile](#).

### 5.4.2.2 getType()

```
Gui::AHud::TypeScene Gui::AHud::getType ( ) const [virtual]
```

Get the Type object.

#### Returns

TypeScene - Type of the scene.

Implements [Gui::IHud](#).

### 5.4.2.3 setPlayer()

```
void Gui::AHud::setPlayer (
    std::shared_ptr< Player > player ) [virtual]
```

Set the [Player](#) object.

**Parameters**

<i>player</i>	<a href="#">Player</a> to display infos.
---------------	--

Implements [Gui::IHud](#).

**5.4.2.4 setTile()**

```
void Gui::AHud::setTile (
    std::shared_ptr< Tile > tile ) [virtual]
```

Set the [Tile](#) object.

**Parameters**

<i>tile</i>	<a href="#">Tile</a> to display infos.
-------------	--

Implements [Gui::IHud](#).

**5.4.2.5 setType()**

```
void Gui::AHud::setType (
    TypeScene type ) [virtual]
```

Set the Type object.

**Parameters**

<i>type</i>	Type of the scene.
-------------	--------------------

Implements [Gui::IHud](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/AHud.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Hud/AHud.cpp

**5.5 Gui::ANetwork Class Reference**

Inheritance diagram for Gui::ANetwork:

Collaboration diagram for Gui::ANetwork:

## Public Member Functions

- **ANetwork** (int port, const std::string &hostName)  
*Construct a new ANetwork object.*
- **~ANetwork** ()=default  
*Destroy the ANetwork object.*
- void **setPort** (int port) final  
*Set the port object.*
- void **setHostName** (const std::string &hostName) final  
*Set the host name object.*
- int **getPort** () const final  
*Get the host name object.*
- std::string **getHostName** () const final  
*Get the host name object.*
- virtual void **connectToServer** ()=0  
*Connect to the server.*
- virtual BufferState **listenServer** ()=0  
*Listen the server and return it message.*
- virtual void **sendMessageServer** (const std::string &message)=0  
*Send a message to the Server.*
- std::string **getBuffer** ()  
*Get the Buffer object. Be careful, this method will delete the current buffer.*

## Public Member Functions inherited from Gui::INetwork

- virtual **~INetwork** ()=default  
*Destroy the INetwork object.*
- virtual void **setPort** (int port)=0  
*Set the port object.*
- virtual void **setHostName** (const std::string &hostName)=0  
*Set the host name object.*
- virtual int **getPort** () const =0  
*Get the host name object.*
- virtual std::string **getHostName** () const =0  
*Get the host name object.*
- virtual void **connectToServer** ()=0  
*Connect to the server.*
- virtual BufferState **listenServer** ()=0  
*Listen to the server.*
- virtual void **sendMessageServer** (const std::string &message)=0  
*Send a message to the server.*
- virtual std::string **getBuffer** ()=0  
*Get the Buffer object. Be careful, this method will delete the current buffer.*

## Protected Attributes

- int **\_port**  
*Port of the server.*
- std::string **\_hostName**  
*Host name of the server.*
- std::string **\_buffer**  
*Buffer to receive server message.*

## Additional Inherited Members

Public Types inherited from [Gui::INetwork](#)

- enum **BufferState** { **NONE** , **READY** , **SERVER\_ERROR** }

## 5.5.1 Constructor & Destructor Documentation

### 5.5.1.1 ANetwork()

```
Gui::ANetwork::ANetwork (
    int port,
    const std::string & hostName )
```

Construct a new [ANetwork](#) object.

#### Parameters

<i>port</i>	Port of the server.
<i>hostName</i>	Host of the server.

### 5.5.1.2 ~ANetwork()

```
Gui::ANetwork::~~ANetwork ( ) [default]
```

Destroy the [ANetwork](#) object.

## 5.5.2 Member Function Documentation

### 5.5.2.1 connectToServer()

```
virtual void Gui::ANetwork::connectToServer ( ) [pure virtual]
```

Connect to the server.

#### Exceptions

<i>NetworkException</i>	If the connection failed.
-------------------------	---------------------------

Implements [Gui::INetwork](#).

Implemented in [Gui::Network](#).

#### 5.5.2.2 getBuffer()

```
std::string Gui::ANetwork::getBuffer ( ) [virtual]
```

Get the Buffer object. Be careful, this method will delete the current buffer.

##### Returns

std::string - Buffer message.

Implements [Gui::INetwork](#).

#### 5.5.2.3 getHostName()

```
std::string Gui::ANetwork::getHostName ( ) const [final], [virtual]
```

Get the host name object.

##### Returns

std::string Host of the server.

Implements [Gui::INetwork](#).

#### 5.5.2.4 getPort()

```
int Gui::ANetwork::getPort ( ) const [final], [virtual]
```

Get the host name object.

##### Returns

std::string Host of the server.

Implements [Gui::INetwork](#).

#### 5.5.2.5 listenServer()

```
virtual BufferState Gui::ANetwork::listenServer ( ) [pure virtual]
```

Listen the server and return it message.

##### Returns

BufferState - Buffer state.

Implements [Gui::INetwork](#).

Implemented in [Gui::Network](#).

#### 5.5.2.6 sendMessageServer()

```
virtual void Gui::ANetwork::sendMessageServer (
    const std::string & message ) [pure virtual]
```

Send a message to the Server.

##### Parameters

<i>message</i>	Message to send to the server.
----------------	--------------------------------

Implements [Gui::INetwork](#).

Implemented in [Gui::Network](#).

#### 5.5.2.7 setHostName()

```
void Gui::ANetwork::setHostName (
    const std::string & hostName ) [final], [virtual]
```

Set the host name object.

##### Parameters

<i>hostName</i>	Host of the server.
-----------------	---------------------

Implements [Gui::INetwork](#).



### 5.5.2.8 setPort()

```
void Gui::ANetwork::setPort (
    int port ) [final], [virtual]
```

Set the port object.

#### Parameters

<i>port</i>	Port of the server.
-------------	---------------------

#### Exceptions

<i>NetworkException</i>	If the port is not in range 1 to 65535.
-------------------------	---

Implements [Gui::INetwork](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/ANetwork.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Network/ANetwork.cpp

## 5.6 Gui::Decoration Class Reference

### Public Member Functions

- [Decoration](#) ()  
*Construct a new [Decoration](#) object.*
- [~Decoration](#) ()=default  
*Destroy the [Decoration](#) object.*
- void [display](#) (std::pair< std::size\_t, std::size\_t > mapSize, size\_t renderDistance, std::pair< std::size\_t, std::size\_t > camPos, std::vector< Vector2 > displayPos)  
*Display decorations.*
- Map< bool > [getGenerationItem](#) (std::size\_t ratio)  
*Generate random emplacement for decorations.*

### 5.6.1 Constructor & Destructor Documentation

#### 5.6.1.1 Decoration()

```
Gui::Decoration::Decoration ( )
```

Construct a new [Decoration](#) object.

### 5.6.1.2 ~Decoration()

```
Gui::Decoration::~~Decoration ( ) [default]
```

Destroy the [Decoration](#) object.

## 5.6.2 Member Function Documentation

### 5.6.2.1 display()

```
void Gui::Decoration::display (
    std::pair< std::size_t, std::size_t > mapSize,
    size_t renderDistance,
    std::pair< std::size_t, std::size_t > camPos,
    std::vector< Vector2 > displayPos )
```

Display decorations.

#### Parameters

<i>mapSize</i>	Size of the map.
<i>renderDistance</i>	Distance to render.
<i>camPos</i>	Position of the camera.
<i>displayPos</i>	Positions to know what to display.

### 5.6.2.2 getGenerationItem()

```
Map< bool > Gui::Decoration::getGenerationItem (
    std::size_t ratio )
```

Generate random emplacement for decorations.

#### Parameters

<i>ratio</i>	Ratio fo random emplacement. If ratio = 10 for a tree, there will be one chance at a ten to have a tree on the tile.
--------------	--

#### Returns

Map<bool> - Boolean list to display item.

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Decoration.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Render/Decoration.cpp

## 5.7 Gui::Egg Class Reference

### Public Types

- enum **EggState** { **IDLE** , **DEAD** , **BORN** , **HATCHING** }

### Public Member Functions

- **Egg** (size\_t id, const std::string &team, std::pair< std::size\_t, std::size\_t > position)  
*Construct a new **Egg** object.*
- **~Egg** ()  
*Destroy the **Egg** object.*
- std::size\_t **getId** () const  
*Get the **Id** object.*
- std::string **getTeam** () const  
*Get the **Team** object.*
- std::pair< std::size\_t, std::size\_t > **getPosition** () const  
*Get the **Position** object.*
- void **setId** (std::size\_t id)  
*Set the **id** object.*
- void **setTeam** (const std::string &team)  
*Set the **team** object.*
- void **setPosition** (std::pair< std::size\_t, std::size\_t > position)  
*Set the **position** object.*
- void **setState** (EggState state)  
*Set the **state** object.*
- EggState **getState** () const  
*Get the **state** object.*

### 5.7.1 Constructor & Destructor Documentation

#### 5.7.1.1 Egg()

```
Gui::Egg::Egg (
    size_t id,
    const std::string & team,
    std::pair< std::size_t, std::size_t > position )
```

Construct a new **Egg** object.

#### Parameters

<i>id</i>	Id of the egg.
<i>team</i>	<b>Team</b> name of the egg.
<i>position</i>	Position of the egg.

**Note**

The egg is created when a player lays an egg.  
The constructor starts the egg animation if implemented.

**5.7.1.2 ~Egg()**

```
Gui::Egg::~~Egg ( )
```

Destroy the [Egg](#) object.

**Note**

The destructor starts the egg animation if implemented.

**5.7.2 Member Function Documentation****5.7.2.1 getId()**

```
std::size_t Gui::Egg::getId ( ) const
```

Get the Id object.

**Returns**

std::size\_t Id of the egg.

**5.7.2.2 getPosition()**

```
std::pair< std::size_t, std::size_t > Gui::Egg::getPosition ( ) const
```

Get the Position object.

**Returns**

std::pair<std::size\_t, std::size\_t> Position of the egg.

### 5.7.2.3 getState()

```
Gui::Egg::EggState Gui::Egg::getState ( ) const
```

Get the state object.

#### Returns

EggState State of the egg.

### 5.7.2.4 getTeam()

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

Get the [Team](#) object.

#### Returns

std::string [Team](#) name of the egg.

### 5.7.2.5 setId()

```
void Gui::Egg::setId (
    std::size_t id )
```

Set the id object.

#### Parameters

<i>id</i>	Id of the egg.
-----------	----------------

### 5.7.2.6 setPosition()

```
void Gui::Egg::setPosition (
    std::pair< std::size_t, std::size_t > position )
```

Set the position object.

#### Parameters

<i>position</i>	Position of the egg.
-----------------	----------------------

### 5.7.2.7 setState()

```
void Gui::Egg::setState (
    EggState state )
```

Set the state object.

#### Parameters

<i>state</i>	State of the egg.
--------------	-------------------

### 5.7.2.8 setTeam()

```
void Gui::Egg::setTeam (
    const std::string & team )
```

Set the team object.

#### Parameters

<i>team</i>	<a href="#">Team</a> name of the egg.
-------------	---------------------------------------

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Egg.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/GameDatas/Egg.cpp

## 5.8 Gui::Engine Class Reference

### Public Member Functions

- [Engine](#) (std::shared\_ptr< [INetwork](#) > network)  
*Construct a new [Engine](#) object.*
- [~Engine](#) ()=default  
*Destroy the [Engine](#) object.*
- void [run](#) ()  
*Run the engine loop.*

### 5.8.1 Constructor & Destructor Documentation

### 5.8.1.1 Engine()

```
Gui::Engine::Engine (
    std::shared_ptr< INetwork > network )
```

Construct a new [Engine](#) object.

#### Parameters

<i>network</i>	<a href="#">Network</a> class.
----------------	--------------------------------

### 5.8.1.2 ~Engine()

```
Gui::Engine::~~Engine ( ) [default]
```

Destroy the [Engine](#) object.

## 5.8.2 Member Function Documentation

### 5.8.2.1 run()

```
void Gui::Engine::run ( )
```

Run the engine loop.

#### Note

This method runs in the main thread.

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Engine/Engine.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Engine/Engine.cpp

## 5.9 Gui::Errors::Error Class Reference

Base class for argument-related errors.

```
#include <Error.hpp>
```

Inheritance diagram for Gui::Errors::Error:

Collaboration diagram for Gui::Errors::Error:

## Additional Inherited Members

### Public Member Functions inherited from [Gui::Errors::AError](#)

- [~AError](#) () override=default  
*Destructor.*
- const char \* [what](#) () const noexcept override  
*Returns the error message.*

### Public Member Functions inherited from [Gui::Errors::IError](#)

- virtual [~IError](#) ()=default  
*Destructor for [IError](#).*
- virtual const char \* [what](#) () const noexcept=0  
*Returns a C-style string describing the error. This function must be implemented by derived classes to provide a description of the error. The returned string should be null-terminated.*

### Protected Attributes inherited from [Gui::Errors::AError](#)

- std::string [\\_message](#)  
*The error message.*

## 5.9.1 Detailed Description

Base class for argument-related errors.

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp

## 5.10 Gui::Event Class Reference

Inheritance diagram for [Gui::Event](#):

Collaboration diagram for [Gui::Event](#):

### Public Member Functions

- [Event](#) ()  
*Construct a new [Event](#) object.*
- [~Event](#) ()=default  
*Destroy the [Event](#) object.*
- void [listen](#) ()  
*Listen the user's events.*



**Public Member Functions inherited from Gui::AEvent**

- [AEvent](#) ()  
*Construct a new AEvent object.*
- [~AEvent](#) ()=default  
*Destroy the AEvent object.*
- virtual void [listen](#) ()=0  
*Listen the user's events.*
- void [setRender](#) (std::shared\_ptr< [Render](#) > render)  
*Set the Render object.*
- void [setGameData](#) (std::shared\_ptr< [GameData](#) > gameData)  
*Set the GameData object.*
- std::shared\_ptr< [Render](#) > [getRender](#) ()  
*Get the Render object.*
- std::shared\_ptr< [GameData](#) > [getGameData](#) ()  
*Get the GameData object.*

**Public Member Functions inherited from Gui::IEvent**

- [IEvent](#) ()=default  
*Construct a new IEvent object.*
- virtual [~IEvent](#) ()=default  
*Destroy the IEvent object.*
- virtual void [listen](#) ()=0  
*Listen the user's events.*
- virtual void [setRender](#) (std::shared\_ptr< [Render](#) > render)=0  
*Set the Render object.*
- virtual void [setGameData](#) (std::shared\_ptr< [GameData](#) > gameData)=0  
*Set the GameData object.*

**Additional Inherited Members****Protected Attributes inherited from Gui::AEvent**

- std::shared\_ptr< [Render](#) > [\\_render](#)  
*Render class to draw scene.*
- std::shared\_ptr< [GameData](#) > [\\_gameData](#)  
*GameData class to contain scene.*

**5.10.1 Constructor & Destructor Documentation****5.10.1.1 Event()**

```
Gui::Event::Event ( )
```

Construct a new [Event](#) object.

### 5.10.1.2 ~Event()

```
Gui::Event::~~Event ( ) [default]
```

Destroy the [Event](#) object.

## 5.10.2 Member Function Documentation

### 5.10.2.1 listen()

```
void Gui::Event::listen ( ) [virtual]
```

Listen the user's events.

Implements [Gui::AEvent](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/Event.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Event/Event.cpp

## 5.11 Gui::GameData Class Reference

### Public Types

- enum **TimeUnitState** { **INCREASE** , **DECREASE** , **NONE** }

### Public Member Functions

- [GameData](#) ()  
*Construct a new [GameData](#) object.*
- [~GameData](#) ()=default  
*Destroy the [GameData](#) object.*
- std::vector< [Gui::Team](#) > & [getTeams](#) ()  
*Get the Teams object.*
- [Gui::Team](#) & [getTeam](#) (const std::string &name)  
*Get a [Team](#) object.*
- void [addTeam](#) (const [Gui::Team](#) &team)  
*Add a team to the game.*
- void [addTeam](#) (const std::string &name, Color playerColor)  
*Add a team to the game.*
- void [addPlayerToTeam](#) (const std::string &teamName, const [Gui::Player](#) &player)  
*Add a player to a team.*
- [Gui::Player](#) & [getPlayer](#) (size\_t id)  
*Get a player object.*

- Map< Gui::Tile > & getMap ()  
*Get the Map object.*
- void setMap (const Map< Gui::Tile > &map)  
*Set the Map object.*
- void setMapSize (size\_t x, size\_t y)  
*Set the Map Size object.*
- std::pair< size\_t, size\_t > getMapSize () const  
*Get the Map Size object.*
- Gui::Tile & getTile (size\_t x, size\_t y)  
*Get a Tile object.*
- void setTile (const Gui::Tile &tile)  
*Set the Tile object.*
- void restartLastTick (void)  
*Restart the last tick clock.*
- void setServerTick (std::size\_t tick)  
*Set the Server Tick object.*
- clock\_t getLastTick () const  
*Get the Last Tick object.*
- std::size\_t getServerTick () const  
*Get the Server Tick object.*
- void setIsEndGame (bool isEndGame)  
*Set the IsEnd Game object.*
- bool getIsEndGame () const  
*Get the IsEnd Game object.*
- void setLastError (const std::string &error)  
*Set the Last Error object.*
- std::string getLastError () const  
*Get the Last Error object.*
- Team & getTeamById (std::size\_t id)  
*Get the Team From Player object.*
- TimeUnitState getTimeUnitFromServer () const  
*Get the Time Unit From Server object.*
- void setTimeUnitFromServer (TimeUnitState timeUnitFromServer)  
*Set the Time Unit From Server object.*
- std::vector< Gui::Egg > & getServerEggs ()  
*Get the Server Eggs object.*
- void addServerEgg (const Gui::Egg &egg)  
*Add an egg to the server ones.*
- void removeServerEgg (size\_t id)  
*Remove an egg from the server ones.*
- void setNbBCTCommandReceived (std::size\_t nb)  
*Set the number of bct command received.*
- std::size\_t getNbBCTCommandReceived () const  
*Get the number of bct command received.*
- void restartLastTickMctCommand ()  
*Restart the last tick mct command clock.*
- clock\_t getLastTickMctCommand () const  
*Get the Last Tick mct command object.*
- void setEndMessage (const std::string &endMessage)  
*Set the End Message object.*
- std::string getEndMessage () const

- Get the End Message object.*
- void [setPlayerModel](#) (const Model &playerModel)  
*Set the [Player](#) Model object.*
- void [setEggModel](#) (const Model &eggModel)  
*Set the [Egg](#) Model object.*
- void [setPlayerModelAnimation](#) (ModelAnimation \*playerModelAnimation)  
*Set the [Player](#) Model Animation object.*
- void [setAnimsCount](#) (int animsCount)  
*Get the [Player](#) Model Animation object.*
- void [setServerError](#) (bool isServerError)  
*Get the Anims Count object.*
- bool [getServerError](#) () const  
*Get the Server Error object.*

## 5.11.1 Constructor & Destructor Documentation

### 5.11.1.1 [GameData\(\)](#)

```
Gui::GameData::GameData ( )
```

Construct a new [GameData](#) object.

### 5.11.1.2 [~GameData\(\)](#)

```
Gui::GameData::~~GameData ( ) [default]
```

Destroy the [GameData](#) object.

## 5.11.2 Member Function Documentation

### 5.11.2.1 [addPlayerToTeam\(\)](#)

```
void Gui::GameData::addPlayerToTeam (
    const std::string & teamName,
    const Gui::Player & player )
```

Add a player to a team.

Parameters

<i>teamName</i>	Name of the team.
<i>player</i>	<a href="#">Player</a> to add.

### 5.11.2.2 addServerEgg()

```
void Gui::GameData::addServerEgg (
    const Gui::Egg & egg )
```

Add an egg to the server ones.

#### Parameters

<i>egg</i>	Egg to add.
------------	-------------

### 5.11.2.3 addTeam() [1/2]

```
void Gui::GameData::addTeam (
    const Gui::Team & team )
```

Add a team to the game.

#### Parameters

<i>team</i>	Team to add.
-------------	--------------

### 5.11.2.4 addTeam() [2/2]

```
void Gui::GameData::addTeam (
    const std::string & name,
    Color playerColor )
```

Add a team to the game.

#### Parameters

<i>name</i>	Name of the team.
<i>playerColor</i>	Color of the team.

### 5.11.2.5 getEndMessage()

```
std::string Gui::GameData::getEndMessage ( ) const
```

Get the End Message object.

**Returns**

std::string - End message of the game.

**5.11.2.6 getIsEndGame()**

```
bool Gui::GameData::getIsEndGame ( ) const
```

Get the IsEnd Game object.

**Returns**

true - The game is finished.

false - The game continue.

**5.11.2.7 getLastError()**

```
std::string Gui::GameData::getLastError ( ) const
```

Get the Last Error object.

**Returns**

std::string - Last error message.

**5.11.2.8 getLastTick()**

```
clock_t Gui::GameData::getLastTick ( ) const
```

Get the Last Tick object.

**Returns**

clock\_t - Last Tick.

**5.11.2.9 getLastTickMctCommand()**

```
clock_t Gui::GameData::getLastTickMctCommand ( ) const
```

Get the Last Tick mct command object.

**Returns**

clock\_t - Last Tick Mct command.

#### 5.11.2.10 getMap()

```
Map< Gui::Tile > & Gui::GameData::getMap ( )
```

Get the Map object.

##### Returns

Map<Gui::Tile>& Map of the game.

#### 5.11.2.11 getMapSize()

```
std::pair< size_t, size_t > Gui::GameData::getMapSize ( ) const
```

Get the Map Size object.

##### Returns

std::pair<size\_t, size\_t> Size of the map.

#### 5.11.2.12 getNbBCTCommandReceived()

```
std::size_t Gui::GameData::getNbBCTCommandReceived ( ) const
```

Get the number of bct command received.

##### Returns

std::size\_t - Number of bct command received.

#### 5.11.2.13 getPlayer()

```
Gui::Player & Gui::GameData::getPlayer (
    size_t id )
```

Get a player object.

##### Parameters

<i>id</i>	Id of the player.
-----------	-------------------

#### 5.11.2.14 `getServerEggs()`

```
std::vector< Gui::Egg > & Gui::GameData::getServerEggs ( )
```

Get the Server Eggs object.

##### Returns

std::vector<Gui::Egg>& Eggs from the server.

#### 5.11.2.15 `getServerError()`

```
bool Gui::GameData::getServerError ( ) const
```

Get the Server Error object.

##### Returns

bool - True if the server is in error.

#### 5.11.2.16 `getServerTick()`

```
std::size_t Gui::GameData::getServerTick ( ) const
```

Get the Server Tick object.

##### Returns

std::size\_t - Server Tick.

#### 5.11.2.17 `getTeam()`

```
Gui::Team & Gui::GameData::getTeam (
    const std::string & name )
```

Get a [Team](#) object.

##### Parameters

<i>name</i>	Name of the team.
-------------	-------------------



**Returns**

[Gui::Team](#)& [Team](#) object.

**5.11.2.18 getTeamById()**

```
Gui::Team & Gui::GameData::getTeamById (
    std::size_t id )
```

Get the [Team](#) From [Player](#) object.

**Parameters**

<i>id</i>	Id of the player.
-----------	-------------------

**Returns**

[Gui::Team](#)& [Team](#) of the player.

**5.11.2.19 getTeams()**

```
std::vector< Gui::Team > & Gui::GameData::getTeams ( )
```

Get the Teams object.

**Returns**

std::vector<Gui::Team>& Teams of the game.

**5.11.2.20 getTile()**

```
Gui::Tile & Gui::GameData::getTile (
    size_t x,
    size_t y )
```

Get a [Tile](#) object.

**Parameters**

<i>x</i>	X position of the tile.
<i>y</i>	Y position of the tile.

**Returns**

[Gui::Tile](#)& [Tile](#) object.

**5.11.2.21 getTimeUnitFromServer()**

```
Gui::GameData::TimeUnitState Gui::GameData::getTimeUnitFromServer ( ) const
```

Get the Time Unit From Server object.

**Returns**

true - The time unit has changed.

**5.11.2.22 removeServerEgg()**

```
void Gui::GameData::removeServerEgg (
    size_t id )
```

Remove an egg from the server ones.

**Parameters**

<i>id</i>	Id of the egg.
-----------	----------------

**5.11.2.23 restartLastTick()**

```
void Gui::GameData::restartLastTick (
    void )
```

Restart the last tick clock.

**5.11.2.24 restartLastTickMctCommand()**

```
void Gui::GameData::restartLastTickMctCommand ( )
```

Restart the last tick mct command clock.

#### 5.11.2.25 setAnimsCount()

```
void Gui::GameData::setAnimsCount (
    int animCount )
```

Get the [Player](#) Model Animation object.

##### Returns

ModelAnimation\* - Model to animate players.

#### 5.11.2.26 setEggModel()

```
void Gui::GameData::setEggModel (
    const Model & eggModel )
```

Set the [Egg](#) Model object.

##### Parameters

<i>eggModel</i>	Model asset of the <a href="#">Team</a> .
-----------------	---

#### 5.11.2.27 setEndMessage()

```
void Gui::GameData::setEndMessage (
    const std::string & endMessage )
```

Set the End Message object.

##### Parameters

<i>endMessage</i>	End message of the game.
-------------------	--------------------------

#### 5.11.2.28 setIsEndGame()

```
void Gui::GameData::setIsEndGame (
    bool isEndGame )
```

Set the IsEnd Game object.

## Parameters

<i>isEndGame</i>	EndGame state.
------------------	----------------

**5.11.2.29 setLastError()**

```
void Gui::GameData::setLastError (
    const std::string & error )
```

Set the Last Error object.

## Parameters

<i>error</i>	Error message.
--------------	----------------

**5.11.2.30 setMap()**

```
void Gui::GameData::setMap (
    const Map< Gui::Tile > & map )
```

Set the Map object.

## Parameters

<i>map</i>	Map of the game.
------------	------------------

**5.11.2.31 setMapSize()**

```
void Gui::GameData::setMapSize (
    size_t x,
    size_t y )
```

Set the Map Size object.

## Parameters

<i>x</i>	X size of the map.
<i>y</i>	Y size of the map.

**Note**

This method resizes the map.

**5.11.2.32 setNbBCTCommandReceived()**

```
void Gui::GameData::setNbBCTCommandReceived (
    std::size_t nb )
```

Set the number of bct command received.

**Parameters**

<i>nb</i>	Number of bct command received.
-----------	---------------------------------

**5.11.2.33 setPlayerModel()**

```
void Gui::GameData::setPlayerModel (
    const Model & playerModel )
```

Set the [Player](#) Model object.

**Parameters**

<i>playerModel</i>	Model asset of the <a href="#">Team</a> .
--------------------	---

**5.11.2.34 setPlayerModelAnimation()**

```
void Gui::GameData::setPlayerModelAnimation (
    ModelAnimation * playerModelAnimation )
```

Set the [Player](#) Model Animation object.

**Parameters**

<i>playerModelAnimation</i>	Model to animate players.
-----------------------------	---------------------------

**5.11.2.35 setServerError()**

```
void Gui::GameData::setServerError (
```

```
bool isServerError )
```

Get the Anims Count object.

#### Returns

int - Animation number of players.

#### 5.11.2.36 setServerTick()

```
void Gui::GameData::setServerTick (
    std::size_t tick )
```

Set the Server Tick object.

#### Parameters

<i>tick</i>	Tick of the server.
-------------	---------------------

#### 5.11.2.37 setTile()

```
void Gui::GameData::setTile (
    const Gui::Tile & tile )
```

Set the [Tile](#) object.

#### Parameters

<i>x</i>	X position of the tile.
<i>y</i>	Y position of the tile.
<i>tile</i>	<a href="#">Tile</a> to set.

#### 5.11.2.38 setTimeUnitFromServer()

```
void Gui::GameData::setTimeUnitFromServer (
    TimeUnitState timeUnitFromServer )
```

Set the Time Unit From Server object.

#### Parameters

<i>timeUnitFromServer</i>	Time unit state.
---------------------------	------------------

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/GameData.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/GameDatas/GameData.cpp

## 5.12 Gui::Errors::GuiGameDataException Class Reference

[Error](#) class for [GameData](#) errors.

```
#include <Error.hpp>
```

Inheritance diagram for Gui::Errors::GuiGameDataException:

Collaboration diagram for Gui::Errors::GuiGameDataException:

### Public Member Functions

- [GuiGameDataException](#) (std::string message)  
*Constructor for [GuiGameDataException](#).*

### Public Member Functions inherited from [Gui::Errors::AError](#)

- [~AError](#) () override=default  
*Destructor.*
- const char \* [what](#) () const noexcept override  
*Returns the error message.*

### Public Member Functions inherited from [Gui::Errors::IError](#)

- virtual [~IError](#) ()=default  
*Destructor for [IError](#).*
- virtual const char \* [what](#) () const noexcept=0  
*Returns a C-style string describing the error. This function must be implemented by derived classes to provide a description of the error. The returned string should be null-terminated.*

### Additional Inherited Members

#### Protected Attributes inherited from [Gui::Errors::AError](#)

- std::string [\\_message](#)  
*The error message.*

### 5.12.1 Detailed Description

[Error](#) class for [GameData](#) errors.

### 5.12.2 Constructor & Destructor Documentation

#### 5.12.2.1 GuiGameDataException()

```
Gui::Errors::GuiGameDataException::GuiGameDataException (
    std::string message )
```

Constructor for [GuiGameDataException](#).

## Parameters

<i>message</i>	The error message.
----------------	--------------------

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Error/Error.cpp

## 5.13 Gui::GUIUpdater Class Reference

Inheritance diagram for Gui::GUIUpdater:

Collaboration diagram for Gui::GUIUpdater:

### Public Member Functions

- **GUIUpdater** (std::shared\_ptr< [GameData](#) > gameData, std::shared\_ptr< [INetwork](#) > network)  
*Construct a new [GUIUpdater](#) object.*
- **~GUIUpdater** ()=default  
*Destroy the [GUIUpdater](#) object.*
- void **update** (const std::string &command, const std::vector< std::string > &data)  
*Update the GUI [GameData](#).*

### Public Member Functions inherited from [Gui::AGUIUpdater](#)

- **AGUIUpdater** (std::shared\_ptr< [GameData](#) > gameData, std::shared\_ptr< [INetwork](#) > network)  
*Construct a new [AGUIUpdater](#) object.*
- **~AGUIUpdater** ()=default  
*Destroy the [AGUIUpdater](#) object.*
- void **update** (const std::string &command, const std::vector< std::string > &data) override=0  
*Update the GUI [GameData](#).*

### Public Member Functions inherited from [Gui::IGUIUpdater](#)

- virtual **~IGUIUpdater** ()=default  
*Destroy the [IGUIUpdater](#) object.*
- virtual void **update** (const std::string &command, const std::vector< std::string > &data)=0  
*Update the GUI [GameData](#).*

### Additional Inherited Members

### Protected Attributes inherited from [Gui::AGUIUpdater](#)

- std::shared\_ptr< [GameData](#) > **\_gameData**  
*The GUI [GameData](#) to update.*
- std::shared\_ptr< [INetwork](#) > **\_network**  
*The network to send commands to the server.*



### 5.13.1 Constructor & Destructor Documentation

#### 5.13.1.1 GUIUpdater()

```
Gui::GUIUpdater::GUIUpdater (
    std::shared_ptr< GameData > gameData,
    std::shared_ptr< INetwork > network )
```

Construct a new [GUIUpdater](#) object.

##### Parameters

<i>gameData</i>	The GUI <a href="#">GameData</a> to update.
<i>network</i>	The network to send commands to the server.

### 5.13.2 Member Function Documentation

#### 5.13.2.1 update()

```
void Gui::GUIUpdater::update (
    const std::string & command,
    const std::vector< std::string > & data ) [virtual]
```

Update the GUI [GameData](#).

##### Parameters

<i>command</i>	The command to update the GUI <a href="#">GameData</a> .
<i>data</i>	The data to update the GUI <a href="#">GameData</a> .

Implements [Gui::AGUIUpdater](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/GUIUpdater.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/GUIUpdater/GUIUpdater.cpp

## 5.14 Gui::Errors::GuiUpdaterException Class Reference

[Error](#) class for [GUIUpdater](#) errors.

```
#include <Error.hpp>
```

Inheritance diagram for Gui::Errors::GuiUpdaterException:

Collaboration diagram for Gui::Errors::GuiUpdaterException:

## Public Member Functions

- [GuiUpdaterException](#) (std::string message)  
*Constructor for [GuiUpdaterException](#).*

## Public Member Functions inherited from [Gui::Errors::AError](#)

- `~AError ()` override=default  
*Destructor.*
- `const char * what ()` const noexcept override  
*Returns the error message.*

## Public Member Functions inherited from [Gui::Errors::IError](#)

- `virtual ~IError ()`=default  
*Destructor for [IError](#).*
- `virtual const char * what ()` const noexcept=0  
*Returns a C-style string describing the error. This function must be implemented by derived classes to provide a description of the error. The returned string should be null-terminated.*

## Additional Inherited Members

## Protected Attributes inherited from [Gui::Errors::AError](#)

- `std::string _message`  
*The error message.*

### 5.14.1 Detailed Description

[Error](#) class for [GUIUpdater](#) errors.

### 5.14.2 Constructor & Destructor Documentation

#### 5.14.2.1 GuiUpdaterException()

```
Gui::Errors::GuiUpdaterException::GuiUpdaterException (
    std::string message )
```

Constructor for [GuiUpdaterException](#).

#### Parameters

<i>message</i>	The error message.
----------------	--------------------

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Error/Error.cpp

## 5.15 Gui::HudEnd Class Reference

Inheritance diagram for Gui::HudEnd:

Collaboration diagram for Gui::HudEnd:

### Public Member Functions

- [HudEnd](#) (std::shared\_ptr< [GameData](#) > gameData)  
*Construct a new Hud End object.*
- [~HudEnd](#) ()=default  
*Destroy the Hud End object.*
- void [display](#) ()  
*Display End Hud.*

### Public Member Functions inherited from [Gui::AHud](#)

- [~AHud](#) ()=default  
*Destroy the AHud object.*
- virtual void [display](#) ()=0  
*Display Hud.*
- void [setPlayer](#) (std::shared\_ptr< [Player](#) > player)  
*Set the Player object.*
- void [setTile](#) (std::shared\_ptr< [Tile](#) > tile)  
*Set the Tile object.*
- [TypeScene](#) [getType](#) () const  
*Get the Type object.*
- void [setType](#) ([TypeScene](#) type)  
*Set the Type object.*

### Public Member Functions inherited from [Gui::IHud](#)

- virtual [~IHud](#) ()=default  
*Destroy the IHud object.*
- virtual void [display](#) ()=0  
*Display the Hud.*
- virtual void [setPlayer](#) (std::shared\_ptr< [Player](#) > player)=0  
*Set the Player object.*
- virtual void [setTile](#) (std::shared\_ptr< [Tile](#) > tile)=0  
*Set the Tile object.*
- virtual [TypeScene](#) [getType](#) () const =0  
*Get the Type object.*
- virtual void [setType](#) ([TypeScene](#) type)=0  
*Set the Type object.*

## Additional Inherited Members

### Public Types inherited from [Gui::IHud](#)

- enum [TypeScene](#) {  
**GAME** , **POV\_PLAYER** , **END\_GAME** , **TILE** ,  
**HELP\_TEXT** , **HELP\_MENU** , **END** }  
*Hud enum for the different scenes.*

### Protected Attributes inherited from [Gui::AHud](#)

- [TypeScene](#) \_typeScene  
*Type of the scene.*
- std::shared\_ptr< [GameData](#) > \_gameData  
[GameData](#) class.
- std::shared\_ptr< [Player](#) > \_player  
[Player](#) to display hud.
- std::shared\_ptr< [Tile](#) > \_tile  
[Tile](#) to display hud.

## 5.15.1 Constructor & Destructor Documentation

### 5.15.1.1 HudEnd()

```
Gui::HudEnd::HudEnd (
    std::shared_ptr< GameData > gameData )
```

Construct a new Hud End object.

#### Parameters

<i>gameData</i>	<a href="#">GameData</a> class.
-----------------	---------------------------------

### 5.15.1.2 ~HudEnd()

```
Gui::HudEnd::~~HudEnd ( ) [default]
```

Destroy the Hud End object.

## 5.15.2 Member Function Documentation

### 5.15.2.1 display()

```
void Gui::HudEnd::display ( ) [virtual]
```

Display End Hud.

Implements [Gui::AHud](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudEnd.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Hud/HudEnd.cpp

## 5.16 Gui::HudGame Class Reference

Inheritance diagram for Gui::HudGame:

Collaboration diagram for Gui::HudGame:

### Public Member Functions

- [HudGame](#) (std::shared\_ptr< [GameData](#) > gameData)  
*Construct a new Hud Game object.*
- [~HudGame](#) ()=default  
*Destroy the Hud Game object.*
- void [display](#) ()  
*Display Game Hud.*

### Public Member Functions inherited from [Gui::AHud](#)

- [~AHud](#) ()=default  
*Destroy the AHud object.*
- virtual void [display](#) ()=0  
*Display Hud.*
- void [setPlayer](#) (std::shared\_ptr< [Player](#) > player)  
*Set the Player object.*
- void [setTile](#) (std::shared\_ptr< [Tile](#) > tile)  
*Set the Tile object.*
- [TypeScene](#) [getType](#) () const  
*Get the Type object.*
- void [setType](#) ([TypeScene](#) type)  
*Set the Type object.*

### Public Member Functions inherited from [Gui::IHud](#)

- virtual [~IHud](#) ()=default  
*Destroy the [IHud](#) object.*
- virtual void [display](#) ()=0  
*Display the Hud.*
- virtual void [setPlayer](#) (std::shared\_ptr< [Player](#) > player)=0  
*Set the [Player](#) object.*
- virtual void [setTile](#) (std::shared\_ptr< [Tile](#) > tile)=0  
*Set the [Tile](#) object.*
- virtual [TypeScene](#) [getType](#) () const =0  
*Get the [Type](#) object.*
- virtual void [setType](#) ([TypeScene](#) type)=0  
*Set the [Type](#) object.*

### Additional Inherited Members

#### Public Types inherited from [Gui::IHud](#)

- enum [TypeScene](#) {  
    **GAME** , **POV\_PLAYER** , **END\_GAME** , **TILE** ,  
    **HELP\_TEXT** , **HELP\_MENU** , **END** }  
*Hud enum for the different scenes.*

#### Protected Attributes inherited from [Gui::AHud](#)

- [TypeScene](#) **\_typeScene**  
*Type of the scene.*
- std::shared\_ptr< [GameData](#) > **\_gameData**  
*[GameData](#) class.*
- std::shared\_ptr< [Player](#) > **\_player**  
*[Player](#) to display hud.*
- std::shared\_ptr< [Tile](#) > **\_tile**  
*[Tile](#) to display hud.*

## 5.16.1 Constructor & Destructor Documentation

### 5.16.1.1 HudGame()

```
Gui::HudGame::HudGame (
    std::shared_ptr< GameData > gameData )
```

Construct a new Hud Game object.

## Parameters

<i>gameData</i>	<a href="#">GameData</a> class.
-----------------	---------------------------------

**5.16.1.2 ~HudGame()**

```
Gui::HudGame::~~HudGame ( ) [default]
```

Destroy the Hud Game object.

**5.16.2 Member Function Documentation****5.16.2.1 display()**

```
void Gui::HudGame::display ( ) [virtual]
```

Display Game Hud.

Implements [Gui::AHud](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Hud/HudGame.cpp

**5.17 Gui::HudHelp Class Reference**

Inheritance diagram for Gui::HudHelp:

Collaboration diagram for Gui::HudHelp:

**Public Member Functions**

- [HudHelp](#) (std::shared\_ptr< [GameData](#) > gameData)  
*Construct a new Hud Help object.*
- [~HudHelp](#) ()=default  
*Destroy the Hud Help object.*
- void [display](#) ()  
*Display Help Hud.*

### Public Member Functions inherited from [Gui::AHud](#)

- [~AHud](#) ()=default  
*Destroy the [AHud](#) object.*
- virtual void [display](#) ()=0  
*Display Hud.*
- void [setPlayer](#) (std::shared\_ptr< [Player](#) > player)  
*Set the [Player](#) object.*
- void [setTile](#) (std::shared\_ptr< [Tile](#) > tile)  
*Set the [Tile](#) object.*
- [TypeScene](#) [getType](#) () const  
*Get the [Type](#) object.*
- void [setType](#) ([TypeScene](#) type)  
*Set the [Type](#) object.*

### Public Member Functions inherited from [Gui::IHud](#)

- virtual [~IHud](#) ()=default  
*Destroy the [IHud](#) object.*
- virtual void [display](#) ()=0  
*Display the Hud.*
- virtual void [setPlayer](#) (std::shared\_ptr< [Player](#) > player)=0  
*Set the [Player](#) object.*
- virtual void [setTile](#) (std::shared\_ptr< [Tile](#) > tile)=0  
*Set the [Tile](#) object.*
- virtual [TypeScene](#) [getType](#) () const =0  
*Get the [Type](#) object.*
- virtual void [setType](#) ([TypeScene](#) type)=0  
*Set the [Type](#) object.*

## Additional Inherited Members

### Public Types inherited from [Gui::IHud](#)

- enum [TypeScene](#) {  
    **GAME** , **POV\_PLAYER** , **END\_GAME** , **TILE** ,  
    **HELP\_TEXT** , **HELP\_MENU** , **END** }  
*Hud enum for the different scenes.*

### Protected Attributes inherited from [Gui::AHud](#)

- [TypeScene](#) **\_typeScene**  
*Type of the scene.*
- std::shared\_ptr< [GameData](#) > **\_gameData**  
*[GameData](#) class.*
- std::shared\_ptr< [Player](#) > **\_player**  
*[Player](#) to display hud.*
- std::shared\_ptr< [Tile](#) > **\_tile**  
*[Tile](#) to display hud.*



## 5.17.1 Constructor & Destructor Documentation

### 5.17.1.1 HudHelp()

```
Gui::HudHelp::HudHelp (
    std::shared_ptr< GameData > gameData )
```

Construct a new Hud Help object.

#### Parameters

<i>gameData</i>	<a href="#">GameData</a> class.
-----------------	---------------------------------

### 5.17.1.2 ~HudHelp()

```
Gui::HudHelp::~HudHelp ( ) [default]
```

Destroy the Hud Help object.

## 5.17.2 Member Function Documentation

### 5.17.2.1 display()

```
void Gui::HudHelp::display ( ) [virtual]
```

Display Help Hud.

Implements [Gui::AHud](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudHelp.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Hud/HudHelp.cpp

## 5.18 Gui::HudPlayer Class Reference

Inheritance diagram for Gui::HudPlayer:

Collaboration diagram for Gui::HudPlayer:

## Public Member Functions

- **HudPlayer** (std::shared\_ptr< **GameData** > gameData)  
*Construct a new Hud **Player** object.*
- **~HudPlayer** ()=default  
*Destroy the Hud **Player** object.*
- void **display** ()  
*Display **Player** Hud.*

## Public Member Functions inherited from **Gui::AHud**

- **~AHud** ()=default  
*Destroy the **AHud** object.*
- virtual void **display** ()=0  
*Display Hud.*
- void **setPlayer** (std::shared\_ptr< **Player** > player)  
*Set the **Player** object.*
- void **setTile** (std::shared\_ptr< **Tile** > tile)  
*Set the **Tile** object.*
- **TypeScene** **getType** () const  
*Get the **Type** object.*
- void **setType** (**TypeScene** type)  
*Set the **Type** object.*

## Public Member Functions inherited from **Gui::IHud**

- virtual **~IHud** ()=default  
*Destroy the **IHud** object.*
- virtual void **display** ()=0  
*Display the Hud.*
- virtual void **setPlayer** (std::shared\_ptr< **Player** > player)=0  
*Set the **Player** object.*
- virtual void **setTile** (std::shared\_ptr< **Tile** > tile)=0  
*Set the **Tile** object.*
- virtual **TypeScene** **getType** () const =0  
*Get the **Type** object.*
- virtual void **setType** (**TypeScene** type)=0  
*Set the **Type** object.*

## Additional Inherited Members

### Public Types inherited from **Gui::IHud**

- enum **TypeScene** {  
    **GAME** , **POV\_PLAYER** , **END\_GAME** , **TILE** ,  
    **HELP\_TEXT** , **HELP\_MENU** , **END** }  
*Hud enum for the different scenes.*

**Protected Attributes inherited from Gui::AHud**

- [TypeScene](#) **\_typeScene**  
*Type of the scene.*
- `std::shared_ptr< GameData > _gameData`  
*[GameData](#) class.*
- `std::shared_ptr< Player > _player`  
*[Player](#) to display hud.*
- `std::shared_ptr< Tile > _tile`  
*[Tile](#) to display hud.*

**5.18.1 Constructor & Destructor Documentation****5.18.1.1 HudPlayer()**

```
Gui::HudPlayer::HudPlayer (
    std::shared_ptr< GameData > gameData )
```

Construct a new Hud [Player](#) object.

**Parameters**

<i>gameData</i>	<a href="#">GameData</a> class.
-----------------	---------------------------------

**5.18.1.2 ~HudPlayer()**

```
Gui::HudPlayer::~~HudPlayer ( ) [default]
```

Destroy the Hud [Player](#) object.

**5.18.2 Member Function Documentation****5.18.2.1 display()**

```
void Gui::HudPlayer::display ( ) [virtual]
```

Display [Player](#) Hud.

Implements [Gui::AHud](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudPlayer.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Hud/HudPlayer.cpp

## 5.19 Gui::HudTile Class Reference

Inheritance diagram for Gui::HudTile:

Collaboration diagram for Gui::HudTile:

### Public Member Functions

- [HudTile](#) (std::shared\_ptr< [GameData](#) > gameData)  
*Construct a new Hud [Tile](#) object.*
- [~HudTile](#) ()=default  
*Destroy the Hud [Tile](#) object.*
- void [display](#) ()  
*Display [Tile](#) Hud.*
- void [displayNbPlayers](#) ()  
*Display number of players.*
- void [displayNbEggs](#) ()  
*Display number of eggs.*

### Public Member Functions inherited from [Gui::AHud](#)

- [~AHud](#) ()=default  
*Destroy the [AHud](#) object.*
- virtual void [display](#) ()=0  
*Display Hud.*
- void [setPlayer](#) (std::shared\_ptr< [Player](#) > player)  
*Set the [Player](#) object.*
- void [setTile](#) (std::shared\_ptr< [Tile](#) > tile)  
*Set the [Tile](#) object.*
- [TypeScene](#) [getType](#) () const  
*Get the [Type](#) object.*
- void [setType](#) ([TypeScene](#) type)  
*Set the [Type](#) object.*

### Public Member Functions inherited from [Gui::IHud](#)

- virtual [~IHud](#) ()=default  
*Destroy the [IHud](#) object.*
- virtual void [display](#) ()=0  
*Display the Hud.*
- virtual void [setPlayer](#) (std::shared\_ptr< [Player](#) > player)=0  
*Set the [Player](#) object.*
- virtual void [setTile](#) (std::shared\_ptr< [Tile](#) > tile)=0  
*Set the [Tile](#) object.*
- virtual [TypeScene](#) [getType](#) () const =0  
*Get the [Type](#) object.*
- virtual void [setType](#) ([TypeScene](#) type)=0  
*Set the [Type](#) object.*

## Additional Inherited Members

### Public Types inherited from [Gui::IHud](#)

- enum [TypeScene](#) {  
**GAME** , **POV\_PLAYER** , **END\_GAME** , **TILE** ,  
**HELP\_TEXT** , **HELP\_MENU** , **END** }  
*Hud enum for the different scenes.*

### Protected Attributes inherited from [Gui::AHud](#)

- [TypeScene](#) \_typeScene  
*Type of the scene.*
- std::shared\_ptr< [GameData](#) > \_gameData  
[GameData](#) class.
- std::shared\_ptr< [Player](#) > \_player  
[Player](#) to display hud.
- std::shared\_ptr< [Tile](#) > \_tile  
[Tile](#) to display hud.

## 5.19.1 Constructor & Destructor Documentation

### 5.19.1.1 HudTile()

```
Gui::HudTile::HudTile (
    std::shared_ptr< GameData > gameData )
```

Construct a new Hud [Tile](#) object.

#### Parameters

<i>gameData</i>	<a href="#">GameData</a> class.
-----------------	---------------------------------

### 5.19.1.2 ~HudTile()

```
Gui::HudTile::~~HudTile ( ) [default]
```

Destroy the Hud [Tile](#) object.

## 5.19.2 Member Function Documentation

### 5.19.2.1 display()

```
void Gui::HudTile::display ( ) [virtual]
```

Display [Tile](#) Hud.

Implements [Gui::AHud](#).

### 5.19.2.2 displayNbEggs()

```
void Gui::HudTile::displayNbEggs ( )
```

Display number of eggs.

### 5.19.2.3 displayNbPlayers()

```
void Gui::HudTile::displayNbPlayers ( )
```

Display number of players.

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

- [/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudTile.hpp](#)
- [/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Hud/HudTile.cpp](#)

## 5.20 Gui::Errors::IError Class Reference

Inheritance diagram for [Gui::Errors::IError](#):

Collaboration diagram for [Gui::Errors::IError](#):

### Public Member Functions

- virtual [~IError](#) ()=default  
*Destructor for [IError](#).*
- virtual const char \* [what](#) () const noexcept=0  
*Returns a C-style string describing the error. This function must be implemented by derived classes to provide a description of the error. The returned string should be null-terminated.*

### 5.20.1 Member Function Documentation

### 5.20.1.1 what()

```
virtual const char * Gui::Errors::IError::what ( ) const [pure virtual], [noexcept]
```

Returns a C-style string describing the error. This function must be implemented by derived classes to provide a description of the error. The returned string should be null-terminated.

#### Returns

const char\* A C-style string describing the error.

Implemented in [Gui::Errors::AError](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/IError.hpp

## 5.21 Gui::IEvent Class Reference

Inheritance diagram for Gui::IEvent:

### Public Member Functions

- [IEvent](#) ()=default  
*Construct a new [IEvent](#) object.*
- virtual [~IEvent](#) ()=default  
*Destroy the [IEvent](#) object.*
- virtual void [listen](#) ()=0  
*Listen the user's events.*
- virtual void [setRender](#) (std::shared\_ptr< [Render](#) > render)=0  
*Set the [Render](#) object.*
- virtual void [setGameData](#) (std::shared\_ptr< [GameData](#) > gameData)=0  
*Set the [GameData](#) object.*

### 5.21.1 Constructor & Destructor Documentation

#### 5.21.1.1 IEvent()

```
Gui::IEvent::IEvent ( ) [default]
```

Construct a new [IEvent](#) object.

### 5.21.1.2 ~IEvent()

```
virtual Gui::IEvent::~~IEvent ( ) [virtual], [default]
```

Destroy the [IEvent](#) object.

## 5.21.2 Member Function Documentation

### 5.21.2.1 listen()

```
virtual void Gui::IEvent::listen ( ) [pure virtual]
```

Listen the user's events.

Implemented in [Gui::Event](#), and [Gui::AEvent](#).

### 5.21.2.2 setGameData()

```
virtual void Gui::IEvent::setGameData (
    std::shared_ptr< GameData > gameData ) [pure virtual]
```

Set the [GameData](#) object.

#### Parameters

<i>gameData</i>	<a href="#">GameData</a> class.
-----------------	---------------------------------

Implemented in [Gui::AEvent](#).

### 5.21.2.3 setRender()

```
virtual void Gui::IEvent::setRender (
    std::shared_ptr< Render > render ) [pure virtual]
```

Set the [Render](#) object.

#### Parameters

<i>render</i>	<a href="#">Render</a> class.
---------------	-------------------------------

Implemented in [Gui::AEvent](#).



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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/IEvent.hpp

## 5.22 Gui::IGUIUpdater Class Reference

Inheritance diagram for Gui::IGUIUpdater:

### Public Member Functions

- virtual `~IGUIUpdater()`=default  
*Destroy the [IGUIUpdater](#) object.*
- virtual void `update` (const std::string &command, const std::vector< std::string > &data)=0  
*Update the GUI [GameData](#).*

### 5.22.1 Member Function Documentation

#### 5.22.1.1 update()

```
virtual void Gui::IGUIUpdater::update (
    const std::string & command,
    const std::vector< std::string > & data ) [pure virtual]
```

Update the GUI [GameData](#).

Implemented in [Gui::GUIUpdater](#), and [Gui::AGUIUpdater](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/IGUIUpdater.hpp

## 5.23 Gui::IHud Class Reference

Inheritance diagram for Gui::IHud:

### Public Types

- enum [TypeScene](#) {  
    **GAME** , **POV\_PLAYER** , **END\_GAME** , **TILE** ,  
    **HELP\_TEXT** , **HELP\_MENU** , **END** }  
*Hud enum for the different scenes.*

## Public Member Functions

- virtual `~IHud()`=default  
*Destroy the [IHud](#) object.*
- virtual void `display()`=0  
*Display the Hud.*
- virtual void `setPlayer` (std::shared\_ptr< [Player](#) > player)=0  
*Set the [Player](#) object.*
- virtual void `setTile` (std::shared\_ptr< [Tile](#) > tile)=0  
*Set the [Tile](#) object.*
- virtual `TypeScene getType()` const =0  
*Get the [Type](#) object.*
- virtual void `setType` ([TypeScene](#) type)=0  
*Set the [Type](#) object.*

## 5.23.1 Member Enumeration Documentation

### 5.23.1.1 TypeScene

```
enum Gui::IHud::TypeScene
```

Hud enum for the different scenes.

## 5.23.2 Constructor & Destructor Documentation

### 5.23.2.1 ~IHud()

```
virtual Gui::IHud::~IHud ( ) [virtual], [default]
```

Destroy the [IHud](#) object.

## 5.23.3 Member Function Documentation

### 5.23.3.1 display()

```
virtual void Gui::IHud::display ( ) [pure virtual]
```

Display the Hud.

Implemented in [Gui::HudEnd](#), [Gui::HudGame](#), [Gui::HudHelp](#), [Gui::HudPlayer](#), [Gui::HudTile](#), and [Gui::AHud](#).

### 5.23.3.2 getType()

```
virtual TypeScene Gui::IHud::getType ( ) const [pure virtual]
```

Get the Type object.

#### Returns

TypeScene - Type of the scene.

Implemented in [Gui::AHud](#).

### 5.23.3.3 setPlayer()

```
virtual void Gui::IHud::setPlayer (
    std::shared_ptr< Player > player ) [pure virtual]
```

Set the [Player](#) object.

#### Parameters

<i>player</i>	<a href="#">Player</a> to display infos.
---------------	--

Implemented in [Gui::AHud](#).

### 5.23.3.4 setTile()

```
virtual void Gui::IHud::setTile (
    std::shared_ptr< Tile > tile ) [pure virtual]
```

Set the [Tile](#) object.

#### Parameters

<i>tile</i>	<a href="#">Tile</a> to display infos.
-------------	--

Implemented in [Gui::AHud](#).

### 5.23.3.5 setType()

```
virtual void Gui::IHud::setType (
    TypeScene type ) [pure virtual]
```

Set the Type object.

## Parameters

<i>type</i>	Type of the scene.
-------------	--------------------

Implemented in [Gui::AHud](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/IHud.hpp

## 5.24 Gui::INetwork Class Reference

Inheritance diagram for Gui::INetwork:

### Public Types

- enum **BufferState** { **NONE** , **READY** , **SERVER\_ERROR** }

### Public Member Functions

- virtual [~INetwork](#) ()=default  
*Destroy the [INetwork](#) object.*
- virtual void [setPort](#) (int port)=0  
*Set the port object.*
- virtual void [setHostName](#) (const std::string &hostName)=0  
*Set the host name object.*
- virtual int [getPort](#) () const =0  
*Get the host name object.*
- virtual std::string [getHostName](#) () const =0  
*Get the host name object.*
- virtual void [connectToServer](#) ()=0  
*Connect to the server.*
- virtual BufferState [listenServer](#) ()=0  
*Listen to the server.*
- virtual void [sendMessageServer](#) (const std::string &message)=0  
*Send a message to the server.*
- virtual std::string [getBuffer](#) ()=0  
*Get the Buffer object. Be careful, this method will delete the current buffer.*

#### 5.24.1 Constructor & Destructor Documentation

### 5.24.1.1 ~INetwork()

```
virtual Gui::INetwork::~~INetwork ( ) [virtual], [default]
```

Destroy the [INetwork](#) object.

## 5.24.2 Member Function Documentation

### 5.24.2.1 connectToServer()

```
virtual void Gui::INetwork::connectToServer ( ) [pure virtual]
```

Connect to the server.

#### Exceptions

<i>Error::NetworkError</i>	If the connection failed.
----------------------------	---------------------------

Implemented in [Gui::Network](#), and [Gui::ANetwork](#).

### 5.24.2.2 getBuffer()

```
virtual std::string Gui::INetwork::getBuffer ( ) [pure virtual]
```

Get the Buffer object. Be careful, this method will delete the current buffer.

#### Returns

std::string - Buffer message.

Implemented in [Gui::ANetwork](#).

### 5.24.2.3 getHostName()

```
virtual std::string Gui::INetwork::getHostName ( ) const [pure virtual]
```

Get the host name object.

#### Returns

std::string Host of the server.

Implemented in [Gui::ANetwork](#).

#### 5.24.2.4 getPort()

```
virtual int Gui::INetwork::getPort ( ) const [pure virtual]
```

Get the host name object.

##### Returns

std::string Host of the server.

Implemented in [Gui::ANetwork](#).

#### 5.24.2.5 listenServer()

```
virtual BufferState Gui::INetwork::listenServer ( ) [pure virtual]
```

Listen to the server.

##### Returns

BufferState - Buffer state.

Implemented in [Gui::Network](#), and [Gui::ANetwork](#).

#### 5.24.2.6 sendMessageServer()

```
virtual void Gui::INetwork::sendMessageServer (
    const std::string & message ) [pure virtual]
```

Send a message to the server.

##### Parameters

<i>message</i>	Message to send.
----------------	------------------

Implemented in [Gui::Network](#), and [Gui::ANetwork](#).

#### 5.24.2.7 setHostName()

```
virtual void Gui::INetwork::setHostName (
    const std::string & hostName ) [pure virtual]
```

Set the host name object.

## Parameters

<i>hostName</i>	Host of the server.
-----------------	---------------------

Implemented in [Gui::ANetwork](#).

## 5.24.2.8 setPort()

```
virtual void Gui::INetwork::setPort (
    int port ) [pure virtual]
```

Set the port object.

## Parameters

<i>port</i>	Port of the server.
-------------	---------------------

## Exceptions

<i>NetworkException</i>	If the port is not in range 1 to 65535.
-------------------------	---

Implemented in [Gui::ANetwork](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/INetwork.hpp

## 5.25 Gui::Inventory Class Reference

## Public Types

- using **Ressources** = size\_t[RessourcesNumber]

## Public Member Functions

- [Inventory](#) ()  
*Construct a new [Inventory](#) object.*
- [Inventory](#) (std::size\_t food, std::size\_t linemate, std::size\_t deraumere, std::size\_t sibur, std::size\_t mendiane, std::size\_t phiras, std::size\_t thystame)  
*Construct a new [Inventory](#) object.*
- [~Inventory](#) ()=default  
*Destroy the [Inventory](#) object.*
- void [setFood](#) (std::size\_t food)  
*Set the Food object.*

- void [setLinemate](#) (std::size\_t linemate)  
*Set the Linemate object.*
- void [setDeraumere](#) (std::size\_t deraumere)  
*Set the Deraumere object.*
- void [setSibur](#) (std::size\_t sibur)  
*Set the Sibur object.*
- void [setMendiane](#) (std::size\_t mendiane)  
*Set the Mendiane object.*
- void [setPhiras](#) (std::size\_t phiras)  
*Set the Phiras object.*
- void [setThystame](#) (std::size\_t thystsme)  
*Set the Thystame object.*
- void [setRessources](#) (Ressources ressources)  
*Set the Ressources object.*
- std::size\_t [getFood](#) (void)  
*Get the Food object.*
- std::size\_t [getLinemate](#) (void)  
*Get the Linemate object.*
- std::size\_t [getDeraumere](#) (void)  
*Get the Deraumere object.*
- std::size\_t [getSibur](#) (void)  
*Get the Sibur object.*
- std::size\_t [getMendiane](#) (void)  
*Get the Mendiane object.*
- std::size\_t [getPhiras](#) (void)  
*Get the Phiras object.*
- std::size\_t [getThystame](#) (void)  
*Get the Thystame object.*
- Ressources & [getRessources](#) (void)  
*Get the Ressources object.*
- void [addResource](#) (std::size\_t resource, std::size\_t quantity)  
*Add resources to inventory.*
- void [removeResource](#) (std::size\_t resource, std::size\_t quantity)  
*Remove resources to inventory.*

## 5.25.1 Constructor & Destructor Documentation

### 5.25.1.1 Inventory() [1/2]

```
Gui::Inventory::Inventory ( )
```

Construct a new [Inventory](#) object.



### 5.25.1.2 Inventory() [2/2]

```
Gui::Inventory::Inventory (
    std::size_t food,
    std::size_t linemate,
    std::size_t deraumere,
    std::size_t sibur,
    std::size_t mendiane,
    std::size_t phiras,
    std::size_t thystame )
```

Construct a new [Inventory](#) object.

#### Parameters

<i>food</i>	Food to set.
<i>linemate</i>	Linemate to set.
<i>deraumere</i>	Deraumere to set.
<i>sibur</i>	Sibur to set.
<i>mendiane</i>	Mendiane to set.
<i>phiras</i>	Phiras to set.
<i>thystame</i>	Thystame to set.

### 5.25.1.3 ~Inventory()

```
Gui::Inventory::~Inventory ( ) [default]
```

Destroy the [Inventory](#) object.

## 5.25.2 Member Function Documentation

### 5.25.2.1 addResource()

```
void Gui::Inventory::addResource (
    std::size_t resource,
    std::size_t quantity )
```

Add resources to inventory.

#### Parameters

<i>resource</i>	Index resource's. resource 0 (food) resource 1 (linemate) resource 2 (deraumere) resource 3 (sibur) resource 4 (mendiane) resource 5 (phiras) resource 6 (thystame)
<i>quantity</i>	Quantity to add.

#### 5.25.2.2 getDeraumere()

```
std::size_t Gui::Inventory::getDeraumere (
    void )
```

Get the Deraumere object.

##### Returns

std::size\_t - deraumere

#### 5.25.2.3 getFood()

```
std::size_t Gui::Inventory::getFood (
    void )
```

Get the Food object.

##### Returns

std::size\_t - food

#### 5.25.2.4 getLinemate()

```
std::size_t Gui::Inventory::getLinemate (
    void )
```

Get the Linemate object.

##### Returns

std::size\_t - linemate

#### 5.25.2.5 getMendiane()

```
std::size_t Gui::Inventory::getMendiane (
    void )
```

Get the Mendiane object.

##### Returns

std::size\_t - mendiane

### 5.25.2.6 getPhiras()

```
std::size_t Gui::Inventory::getPhiras (
    void )
```

Get the Phiras object.

#### Returns

std::size\_t - phiras

### 5.25.2.7 getRessources()

```
Gui::Inventory::Ressources & Gui::Inventory::getRessources (
    void )
```

Get the Ressources object.

#### Returns

Ressources - ressources

### 5.25.2.8 getSibur()

```
std::size_t Gui::Inventory::getSibur (
    void )
```

Get the Sibur object.

#### Returns

std::size\_t - sibur

### 5.25.2.9 getThystame()

```
std::size_t Gui::Inventory::getThystame (
    void )
```

Get the Thystame object.

#### Returns

std::size\_t - thystame

### 5.25.2.10 removeResource()

```
void Gui::Inventory::removeResource (
    std::size_t resource,
    std::size_t quantity )
```

Remove resources to inventory.

## Parameters

<i>resource</i>	Index resource's. resource 0 (food) resource 1 (linemate) resource 2 (deraumere) resource 3 (sibur) resource 4 (mendiane) resource 5 (phiras) resource 6 (thystame)
<i>quantity</i>	Quantity to remove.

**5.25.2.11 setDeraumere()**

```
void Gui::Inventory::setDeraumere (
    std::size_t deraumere )
```

Set the Deraumere object.

## Parameters

<i>deraumere</i>	Deraumere to set.
------------------	-------------------

**5.25.2.12 setFood()**

```
void Gui::Inventory::setFood (
    std::size_t food )
```

Set the Food object.

## Parameters

<i>food</i>	Food to set.
-------------	--------------

**5.25.2.13 setLinemate()**

```
void Gui::Inventory::setLinemate (
    std::size_t linemate )
```

Set the Linemate object.

## Parameters

<i>linemate</i>	Linemate to set.
-----------------	------------------

#### 5.25.2.14 setMendiane()

```
void Gui::Inventory::setMendiane (
    std::size_t mendiane )
```

Set the Mendiane object.

##### Parameters

<i>mendiane</i>	Mendiane to set.
-----------------	------------------

#### 5.25.2.15 setPhiras()

```
void Gui::Inventory::setPhiras (
    std::size_t phiras )
```

Set the Phiras object.

##### Parameters

<i>phiras</i>	Phiras to set.
---------------	----------------

#### 5.25.2.16 setRessources()

```
void Gui::Inventory::setRessources (
    Ressources ressources )
```

Set the Ressources object.

##### Parameters

<i>ressources</i>	Ressources to set.
-------------------	--------------------

#### 5.25.2.17 setSibur()

```
void Gui::Inventory::setSibur (
    std::size_t sibur )
```

Set the Sibur object.

## Parameters

<i>sibur</i>	Sibur to set.
--------------	---------------

**5.25.2.18 setThystame()**

```
void Gui::Inventory::setThystame (
    std::size_t thystsame )
```

Set the Thystame object.

## Parameters

<i>thystame</i>	Thystame to set.
-----------------	------------------

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Inventory.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/GameDatas/Inventory.cpp

**5.26 Gui::IServerParser Class Reference**

Inheritance diagram for Gui::IServerParser:

**Public Member Functions**

- virtual **~IServerParser** ()=default  
*Destroy the [IServerParser](#) object.*
- virtual std::vector< std::string > **parse** (const std::string &command)=0  
*Parse the command server.*

**5.26.1 Member Function Documentation****5.26.1.1 parse()**

```
virtual std::vector< std::string > Gui::IServerParser::parse (
    const std::string & command ) [pure virtual]
```

Parse the command server.

## Parameters

<i>command</i>	Command to parse.
----------------	-------------------

## Returns

std::vector<std::string> - arguments parsed.

Implemented in [Gui::ServerParser](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/IServerParser.hpp

## 5.27 Gui::Network Class Reference

Inheritance diagram for Gui::Network:

Collaboration diagram for Gui::Network:

### Public Member Functions

- [Network](#) (int port, const std::string &hostName)  
*Construct a new [Network](#) object.*
- [~Network](#) ()  
*Destroy the [Network](#) object.*
- void [connectToServer](#) ()  
*Connect the Gui network with the server.*
- BufferState [listenServer](#) ()  
*Listen the server and return it message.*
- void [sendMessageServer](#) (const std::string &message)  
*Send a message to the Server.*

### Public Member Functions inherited from [Gui::ANetwork](#)

- [ANetwork](#) (int port, const std::string &hostName)  
*Construct a new [ANetwork](#) object.*
- [~ANetwork](#) ()=default  
*Destroy the [ANetwork](#) object.*
- void [setPort](#) (int port) final  
*Set the port object.*
- void [setHostName](#) (const std::string &hostName) final  
*Set the host name object.*
- int [getPort](#) () const final  
*Get the host name object.*
- std::string [getHostName](#) () const final  
*Get the host name object.*
- virtual void [connectToServer](#) ()=0  
*Connect to the server.*
- virtual BufferState [listenServer](#) ()=0  
*Listen the server and return it message.*
- virtual void [sendMessageServer](#) (const std::string &message)=0  
*Send a message to the Server.*
- std::string [getBuffer](#) ()  
*Get the Buffer object. Be careful, this method will delete the current buffer.*

## Public Member Functions inherited from [Gui::INetwork](#)

- virtual [~INetwork](#) ()=default  
*Destroy the [INetwork](#) object.*
- virtual void [setPort](#) (int port)=0  
*Set the port object.*
- virtual void [setHostName](#) (const std::string &hostName)=0  
*Set the host name object.*
- virtual int [getPort](#) () const =0  
*Get the host name object.*
- virtual std::string [getHostName](#) () const =0  
*Get the host name object.*
- virtual void [connectToServer](#) ()=0  
*Connect to the server.*
- virtual BufferState [listenServer](#) ()=0  
*Listen to the server.*
- virtual void [sendMessageServer](#) (const std::string &message)=0  
*Send a message to the server.*
- virtual std::string [getBuffer](#) ()=0  
*Get the Buffer object. Be careful, this method will delete the current buffer.*

## Additional Inherited Members

### Public Types inherited from [Gui::INetwork](#)

- enum **BufferState** { **NONE** , **READY** , **SERVER\_ERROR** }

### Protected Attributes inherited from [Gui::ANetwork](#)

- int **\_port**  
*Port of the server.*
- std::string **\_hostName**  
*Host name of the server.*
- std::string **\_buffer**  
*Buffer to receive server message.*

## 5.27.1 Constructor & Destructor Documentation

### 5.27.1.1 [Network](#)()

```
Gui::Network::Network (
    int port,
    const std::string & hostName )
```

Construct a new [Network](#) object.



## Parameters

<i>port</i>	Port of the server.
<i>hostName</i>	Host of the server.

**5.27.1.2 ~Network()**

```
Gui::Network::~~Network ( )
```

Destroy the [Network](#) object.

**5.27.2 Member Function Documentation****5.27.2.1 connectToServer()**

```
void Gui::Network::connectToServer ( ) [virtual]
```

Connect the Gui network with the server.

## Exceptions

<i>NetworkException</i>	If the connection failed.
-------------------------	---------------------------

Implements [Gui::ANetwork](#).

**5.27.2.2 listenServer()**

```
Gui::Network::BufferState Gui::Network::listenServer ( ) [virtual]
```

Listen the server and return it message.

## Returns

BufferState - Buffer state.

Implements [Gui::ANetwork](#).

**5.27.2.3 sendMessageServer()**

```
void Gui::Network::sendMessageServer (
    const std::string & message ) [virtual]
```

Send a message to the Server.

## Parameters

<i>message</i>	Message to send to the server.
----------------	--------------------------------

Implements [Gui::ANetwork](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/Network.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Network/Network.cpp

## 5.28 Gui::Errors::NetworkException Class Reference

[Error](#) class for network errors.

```
#include <Error.hpp>
```

Inheritance diagram for Gui::Errors::NetworkException:

Collaboration diagram for Gui::Errors::NetworkException:

### Public Member Functions

- [NetworkException](#) (std::string message)  
*Constructor for [NetworkException](#).*

### Public Member Functions inherited from [Gui::Errors::AError](#)

- [~AError](#) () override=default  
*Destructor.*
- const char \* [what](#) () const noexcept override  
*Returns the error message.*

### Public Member Functions inherited from [Gui::Errors::IError](#)

- virtual [~IError](#) ()=default  
*Destructor for [IError](#).*
- virtual const char \* [what](#) () const noexcept=0  
*Returns a C-style string describing the error. This function must be implemented by derived classes to provide a description of the error. The returned string should be null-terminated.*

### Additional Inherited Members

### Protected Attributes inherited from [Gui::Errors::AError](#)

- std::string [\\_message](#)  
*The error message.*

### 5.28.1 Detailed Description

[Error](#) class for network errors.

### 5.28.2 Constructor & Destructor Documentation

#### 5.28.2.1 NetworkException()

```
Gui::Errors::NetworkException::NetworkException (
    std::string message )
```

Constructor for [NetworkException](#).

Parameters

<i>message</i>	The error message.
----------------	--------------------

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Error/Error.cpp

## 5.29 Gui::ParseCommandLine Class Reference

### Public Member Functions

- [ParseCommandLine](#) (int argc, char \*\*argv)  
*Construct a new Parse Command Line object.*
- [~ParseCommandLine](#) ()=default  
*Destroy the Parse Command Line object.*
- void [parseFlags](#) (int argc, char \*\*argv)  
*Parse flags in command line.*
- int [getPort](#) (void)  
*Get the port object.*
- std::string [getHostName](#) (void)  
*Get the hostName object.*

### 5.29.1 Constructor & Destructor Documentation

#### 5.29.1.1 ParseCommandLine()

```
Gui::ParseCommandLine::ParseCommandLine (
    int argc,
    char ** argv )
```

Construct a new Parse Command Line object.

**Parameters**

<i>argc</i>	Number of arguments in command line.
<i>argv</i>	Array with command line arguments.

**5.29.1.2 ~ParseCommandLine()**

```
Gui::ParseCommandLine::~~ParseCommandLine ( ) [default]
```

Destroy the Parse Command Line object.

**5.29.2 Member Function Documentation****5.29.2.1 getHostName()**

```
std::string Gui::ParseCommandLine::getHostName (
    void )
```

Get the hostName object.

**Returns**

std::string - hostName

**5.29.2.2 getPort()**

```
int Gui::ParseCommandLine::getPort (
    void )
```

Get the port object.

**Returns**

int - port

**5.29.2.3 parseFlags()**

```
void Gui::ParseCommandLine::parseFlags (
    int argc,
    char ** argv )
```

Parse flags in command line.

## Parameters

<i>argc</i>	Number of arguments in command line.
<i>argv</i>	Array with command line arguments.

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ParseCommandLine.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Parsing/ParseCommandLine.cpp

## 5.30 Gui::Errors::ParseCommandLineException Class Reference

[Error](#) class for parseCommandLine errors.

```
#include <Error.hpp>
```

Inheritance diagram for Gui::Errors::ParseCommandLineException:

Collaboration diagram for Gui::Errors::ParseCommandLineException:

### Public Member Functions

- [ParseCommandLineException](#) (std::string message)  
*Constructor for [ParseCommandLineException](#).*

### Public Member Functions inherited from [Gui::Errors::AError](#)

- [~AError](#) () override=default  
*Destructor.*
- const char \* [what](#) () const noexcept override  
*Returns the error message.*

### Public Member Functions inherited from [Gui::Errors::IError](#)

- virtual [~IError](#) ()=default  
*Destructor for [IError](#).*
- virtual const char \* [what](#) () const noexcept=0  
*Returns a C-style string describing the error. This function must be implemented by derived classes to provide a description of the error. The returned string should be null-terminated.*

### Additional Inherited Members

### Protected Attributes inherited from [Gui::Errors::AError](#)

- std::string [\\_message](#)  
*The error message.*

### 5.30.1 Detailed Description

[Error](#) class for parseCommandLine errors.

### 5.30.2 Constructor & Destructor Documentation

#### 5.30.2.1 ParseCommandLineException()

```
Gui::Errors::ParseCommandLineException::ParseCommandLineException (
    std::string message )
```

Constructor for [ParseCommandLineException](#).

Parameters

<i>message</i>	The error message.
----------------	--------------------

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Error/Error.cpp

## 5.31 Gui::Player Class Reference

Collaboration diagram for Gui::Player:

### Public Types

- enum **PlayerState** {  
**IDLE** = 2 , **BORN** = 8 , **BROADCAST** = 12 , **EJECT** = 5 ,  
**BEING\_EJECTED** = 15 , **EJECTED** = 7 , **WALK** = 6 , **INCANTATION** = 0 ,  
**LAY\_EGG** = 7 , **DROP** = 9 , **COLLECT** = 9 , **DEAD** = 1 }

### Public Member Functions

- [Player](#) (std::size\_t id, const std::string &team, std::pair< std::size\_t, std::size\_t > position, std::size\_t orientation, std::size\_t level=1)  
Construct a new [Player](#) object.
- [~Player](#) ()=default  
Destroy the [Player](#) object.
- void [setPosition](#) (std::pair< std::size\_t, std::size\_t > position)  
Set the *Position* object.
- void [setPosition3D](#) (Vector3 position3D)

- Set the *Position3D* object.
  - void [setId](#) (std::size\_t id)
- Set the *Id* object.
  - void [setLevel](#) (std::size\_t level)
- Set the *Level* object.
  - void [setOrientation](#) (std::size\_t orientation)
- Set the *Orientation* object.
  - void [setTeam](#) (const std::string &team)
- Set the *Team* object.
  - std::pair< std::size\_t, std::size\_t > [getPosition](#) (void) const
- Get the *Position* object.
  - Vector3 [getPosition3D](#) (void) const
- Get the *Position3D* object.
  - std::size\_t [getId](#) (void) const
- Get the *Id* object.
  - std::size\_t [getLevel](#) (void) const
- Get the *Level* object.
  - std::size\_t [getOrientation](#) (void) const
- Get the *Orientation* object.
  - std::string [getTeam](#) (void) const
- Get the *Team* object.
  - void [setState](#) (PlayerState state)
- Set the *State* object.
  - PlayerState [getState](#) (void) const
- Get the *State* object.
  - void [setBroadcast](#) (const std::string &broadcast)
- Set the *Broadcast* object.
  - std::string [getBroadcast](#) () const
- Get the *Broadcast* object.
  - float [getRotationFromOrientation](#) () const
- Get the *Vector From Orientation* object.
  - Vector3 [getCenterPosition](#) ()
- Get the *Center Position* object.
  - void [setCurrentFrame](#) (int currentFrame)
- Set the *Current Frame* object.
  - int [getCurrentFrame](#) () const
- Get the *Current Frame* object.
  - void [restartAnimationTimeEllapsed](#) ()
- Restart the timer animation.
  - clock\_t [getAnimationTimeEllapsed](#) () const
- Get the *Animation Time Ellapsed* object.

## Public Attributes

- [Inventory](#) inventory  
*Inventory* of the player.

### 5.31.1 Constructor & Destructor Documentation

### 5.31.1.1 Player()

```
Gui::Player::Player (
    std::size_t id,
    const std::string & team,
    std::pair< std::size_t, std::size_t > position,
    std::size_t orientation,
    std::size_t level = 1 )
```

Construct a new [Player](#) object.

#### Parameters

<i>id</i>	Id of the player.
<i>team</i>	<a href="#">Team</a> name of the player.
<i>position</i>	Position of the <a href="#">Player</a> .

### 5.31.1.2 ~Player()

```
Gui::Player::~~Player ( ) [default]
```

Destroy the [Player](#) object.

## 5.31.2 Member Function Documentation

### 5.31.2.1 getAnimationTimeEllapsed()

```
clock_t Gui::Player::getAnimationTimeEllapsed ( ) const
```

Get the Animation Time Ellapsed object.

#### Returns

clock\_t - Animation time ellapsed.

### 5.31.2.2 getBroadcast()

```
std::string Gui::Player::getBroadcast ( ) const
```

Get the Broadcast object.

#### Returns

std::string - Broadcast message.



### 5.31.2.3 getCenterPosition()

```
Vector3 Gui::Player::getCenterPosition ( )
```

Get the Center Position object.

#### Returns

Vector3 - Center position.

### 5.31.2.4 getCurrentFrame()

```
int Gui::Player::getCurrentFrame ( ) const
```

Get the Current Frame object.

#### Returns

int - Current frame.

### 5.31.2.5 getId()

```
std::size_t Gui::Player::getId (
    void ) const
```

Get the Id object.

#### Returns

std::size\_t - id

### 5.31.2.6 getLevel()

```
std::size_t Gui::Player::getLevel (
    void ) const
```

Get the Level object.

#### Returns

std::size\_t - level

#### 5.31.2.7 getOrientation()

```
std::size_t Gui::Player::getOrientation (
    void ) const
```

Get the Orientation object.

##### Returns

std::size\_t - orientation

#### 5.31.2.8 getPosition()

```
std::pair< std::size_t, std::size_t > Gui::Player::getPosition (
    void ) const
```

Get the Position object.

##### Returns

std::pair<std::size\_t, std::size\_t> - position

#### 5.31.2.9 getPosition3D()

```
Vector3 Gui::Player::getPosition3D (
    void ) const
```

Get the Position3D object.

##### Returns

Vector3 - position3D

#### 5.31.2.10 getRotationFromOrientation()

```
float Gui::Player::getRotationFromOrientation ( ) const
```

Get the Vector From Orientation object.

##### Returns

Vector3 - Vector3 from orientation.

#### 5.31.2.11 getState()

```
Gui::Player::PlayerState Gui::Player::getState (
    void ) const
```

Get the State object.

##### Returns

PlayerState - [Player](#) state.

#### 5.31.2.12 getTeam()

```
std::string Gui::Player::getTeam (
    void ) const
```

Get the [Team](#) object.

##### Returns

std::string - team name

#### 5.31.2.13 restartAnimationTimeEllapsed()

```
void Gui::Player::restartAnimationTimeEllapsed ( )
```

Restart the timer animation.

#### 5.31.2.14 setBroadcast()

```
void Gui::Player::setBroadcast (
    const std::string & broadcast )
```

Set the Broadcast object.

##### Parameters

<i>broadcast</i>	New broadcast message.
------------------	------------------------

### 5.31.2.15 setCurrentFrame()

```
void Gui::Player::setCurrentFrame (
    int currentFrame )
```

Set the Current Frame object.

#### Parameters

<i>currentFrame</i>	Current Frame to set.
---------------------	-----------------------

### 5.31.2.16 setId()

```
void Gui::Player::setId (
    std::size_t id )
```

Set the Id object.

#### Parameters

<i>id</i>	Id of the player.
-----------	-------------------

### 5.31.2.17 setLevel()

```
void Gui::Player::setLevel (
    std::size_t level )
```

Set the Level object.

#### Parameters

<i>level</i>	Level of the player.
--------------	----------------------

### 5.31.2.18 setOrientation()

```
void Gui::Player::setOrientation (
    std::size_t orientation )
```

Set the Orientation object.

## Parameters

<i>orientation</i>	Orientation of the player.
--------------------	----------------------------

**5.31.2.19 setPosition()**

```
void Gui::Player::setPosition (
    std::pair< std::size_t, std::size_t > position )
```

Set the Position object.

## Parameters

<i>position</i>	Position of the player.
-----------------	-------------------------

**5.31.2.20 setPosition3D()**

```
void Gui::Player::setPosition3D (
    Vector3 position3D )
```

Set the Position3D object.

## Parameters

<i>position3D</i>	Position of the player.
-------------------	-------------------------

**5.31.2.21 setState()**

```
void Gui::Player::setState (
    PlayerState state )
```

Set the State object.

## Parameters

<i>state</i>	New player state.
--------------	-------------------

### 5.31.2.22 setTeam()

```
void Gui::Player::setTeam (
    const std::string & team )
```

Set the [Team](#) object.

#### Parameters

<i>team</i>	<a href="#">Team</a> name of the player.
-------------	--

## 5.31.3 Member Data Documentation

### 5.31.3.1 inventory

```
Inventory Gui::Player::inventory
```

[Inventory](#) of the player.

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Player.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/GameDatas/Player.cpp

## 5.32 Gui::Render Class Reference

### Public Member Functions

- [Render](#) (std::shared\_ptr< [GameData](#) > gameData)  
*Construct a new [Render](#) object.*
- [~Render](#) ()  
*Destroy the [Render](#) object.*
- bool [isOpen](#) ()  
*Check if the window is open.*
- void [draw](#) ()  
*Draw the scene.*
- std::shared\_ptr< Camera > [getCamera](#) ()  
*Get the Camera object.*
- void [setIsDebug](#) (bool isDebug)  
*Set the Is Debug object.*
- bool [getIsDebug](#) (void)  
*Get the Is Debug object.*
- void [setCameraType](#) (Gui::UserCamera::CameraType type)  
*Set the Type object.*
- Gui::UserCamera::CameraType [getCameraType](#) () const

- Get the Type object.*

  - void [setCameraPlayerPov](#) (std::size\_t id)

*Set the Camera player pov id.*
- std::size\_t [getCameraPlayerPov](#) () const

*Get the Camera player pov id.*
- void [setCameraTile](#) (std::pair< std::size\_t, std::size\_t > pos)

*Set the Camera [Tile](#) object.*
- std::pair< std::size\_t, std::size\_t > [getCameraTile](#) () const

*Get the Camera [Tile](#) object.*
- Model [getTileModel](#) () const

*Get the [Tile](#) model.*
- void [setRenderDistance](#) (size\_t renderDistance)

*Set the [Render](#) Distance value.*
- size\_t [getRenderDistance](#) () const

*Get the [Render](#) Distance value.*
- bool [isCameraInPlayerPov](#) () const

*Check if the camera is in player pov.*
- void [changePlayerPOV](#) (size\_t playerId)

*Change the player point of view.*
- void [setPlayerPov](#) (size\_t playerId)

*Sets the Pov of the player.*
- void [changePOVToFirstPerson](#) (size\_t id)

*Change the camera to the player.*
- void [changePOVToSecondPerson](#) (size\_t id)

*Change the camera to the player.*
- void [changePOVToThirdPerson](#) (size\_t id)

*Change the camera to the player.*
- size\_t [getTimeUnit](#) () const

*Get the Time Unit value.*
- void [setTimeUnit](#) (size\_t timeUnit)

*Set the Time Unit value.*
- void [setPlayerVision](#) (bool isPlayerVision)

*Set the [Player](#) Vision value.*
- bool [getPlayerVision](#) () const

*Get the [Player](#) Vision value.*
- void [setHelpMenu](#) (bool isHelpMenu)

*Set the Help Menu value.*
- bool [getHelpMenu](#) () const

*Get the Help Menu value.*
- void [drawEnd](#) () const

*Draw the end of the game.*

### 5.32.1 Constructor & Destructor Documentation

### 5.32.1.1 Render()

```
Gui::Render::Render (
    std::shared_ptr< GameData > gameData )
```

Construct a new [Render](#) object.

### 5.32.1.2 ~Render()

```
Gui::Render::~~Render ( )
```

Destroy the [Render](#) object.

## 5.32.2 Member Function Documentation

### 5.32.2.1 changePlayerPOV()

```
void Gui::Render::changePlayerPOV (
    size_t playerId )
```

Change the player point of view.

#### Parameters

<i>playerId</i>	<a href="#">Player</a> id to select.
-----------------	--------------------------------------

#### Note

The player point of view is the first person, second person and third person.

### 5.32.2.2 changePOVToFirstPerson()

```
void Gui::Render::changePOVToFirstPerson (
    size_t id )
```

Change the camera to the player.

#### Parameters

<i>player</i>	<a href="#">Player</a> to select.
---------------	-----------------------------------



### 5.32.2.3 changePOVToSecondPerson()

```
void Gui::Render::changePOVToSecondPerson (
    size_t id )
```

Change the camera to the player.

#### Parameters

<i>player</i>	<a href="#">Player</a> to select.
---------------	-----------------------------------

### 5.32.2.4 changePOVToThirdPerson()

```
void Gui::Render::changePOVToThirdPerson (
    size_t id )
```

Change the camera to the player.

#### Parameters

<i>player</i>	<a href="#">Player</a> to select.
---------------	-----------------------------------

### 5.32.2.5 draw()

```
void Gui::Render::draw ( )
```

Draw the scene.

### 5.32.2.6 drawEnd()

```
void Gui::Render::drawEnd ( ) const
```

Draw the end of the game.

### 5.32.2.7 `getCamera()`

```
std::shared_ptr< Camera > Gui::Render::getCamera ( )
```

Get the Camera object.

#### Returns

`std::shared_ptr<Camera>` - camera

### 5.32.2.8 `getCameraPlayerPov()`

```
std::size_t Gui::Render::getCameraPlayerPov ( ) const
```

Get the Camera player pov id.

#### Returns

`std::size_t` - Id of the player.

### 5.32.2.9 `getCameraTile()`

```
std::pair< std::size_t, std::size_t > Gui::Render::getCameraTile ( ) const
```

Get the Camera [Tile](#) object.

#### Returns

`std::pair<std::size_t, std::size_t>` - [Tile](#) position.

### 5.32.2.10 `getCameraType()`

```
Gui::UserCamera::CameraType Gui::Render::getCameraType ( ) const
```

Get the Type object.

#### Returns

`CameraType` - Camera type.

#### 5.32.2.11 getHelpMenu()

```
bool Gui::Render::getHelpMenu ( ) const
```

Get the Help Menu value.

##### Returns

true - Display the help menu.

false - Do not display the help menu.

#### 5.32.2.12 getIsDebug()

```
bool Gui::Render::getIsDebug (
    void )
```

Get the Is Debug object.

##### Returns

true - diplay debug

false - do not display debug

#### 5.32.2.13 getPlayerVision()

```
bool Gui::Render::getPlayerVision ( ) const
```

Get the [Player](#) Vision value.

##### Returns

true - Display player vision.

false - Do not display player vision.

#### 5.32.2.14 getRenderDistance()

```
size_t Gui::Render::getRenderDistance ( ) const
```

Get the [Render](#) Distance value.

#### 5.32.2.15 `getTileModel()`

```
Model Gui::Render::getTileModel ( ) const
```

Get the [Tile](#) model.

#### 5.32.2.16 `getTimeUnit()`

```
size_t Gui::Render::getTimeUnit ( ) const
```

Get the Time Unit value.

##### Returns

`size_t` - Time unit value.

#### 5.32.2.17 `isCameraInPlayerPov()`

```
bool Gui::Render::isCameraInPlayerPov ( ) const
```

Check if the camera is in player pov.

##### Returns

`true` - Camera is in player pov.

`false` - Camera is not in player pov.

##### Note

The player pov is the first person, second person and third person.

#### 5.32.2.18 `isOpen()`

```
bool Gui::Render::isOpen ( )
```

Check if the window is open.

##### Returns

`true` - the window is open

`false` - the window is closed

#### 5.32.2.19 `setCameraPlayerPov()`

```
void Gui::Render::setCameraPlayerPov (
    std::size_t id )
```

Set the Camera player pov id.

## Parameters

<i>id</i>	Id of the player.
-----------	-------------------

**5.32.2.20 setCameraTile()**

```
void Gui::Render::setCameraTile (
    std::pair< std::size_t, std::size_t > pos )
```

Set the Camera [Tile](#) object.

## Parameters

<i>pos</i>	<a href="#">Tile</a> position.
------------	--------------------------------

**5.32.2.21 setCameraType()**

```
void Gui::Render::setCameraType (
    Gui::UserCamera::CameraType type )
```

Set the Type object.

## Parameters

<i>type</i>	Type to set.
-------------	--------------

**5.32.2.22 setHelpMenu()**

```
void Gui::Render::setHelpMenu (
    bool isHelpMenu )
```

Set the Help Menu value.

## Parameters

<i>isHelpMenu</i>	New help menu value.
-------------------	----------------------

**5.32.2.23 setIsDebug()**

```
void Gui::Render::setIsDebug (
    bool isDebug )
```

Set the Is Debug object.

**Parameters**

<i>isDebug</i>	New Is Debug value to set.
----------------	----------------------------

**5.32.2.24 setPlayerPov()**

```
void Gui::Render::setPlayerPov (
    size_t playerId )
```

Sets the Pov of the player.

**Parameters**

<i>playerId</i>	<a href="#">Player</a> id to select.
-----------------	--------------------------------------

**5.32.2.25 setPlayerVision()**

```
void Gui::Render::setPlayerVision (
    bool isPlayerVision )
```

Set the [Player](#) Vision value.

**Parameters**

<i>isPlayerVision</i>	New player vision value.
-----------------------	--------------------------

**Note**

True to display player vision.

False to not display player vision.

**5.32.2.26 setRenderDistance()**

```
void Gui::Render::setRenderDistance (
    size_t renderDistance )
```

Set the [Render](#) Distance value.

## Parameters

<i>renderDistance</i>	New render distance value.
-----------------------	----------------------------

**5.32.2.27 setTimeUnit()**

```
void Gui::Render::setTimeUnit (
    size_t timeUnit )
```

Set the Time Unit value.

## Parameters

<i>timeUnit</i>	New time unit value.
-----------------	----------------------

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Render.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Render/Render.cpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Render/RenderAnimations.cpp

**5.33 Gui::ServerParser Class Reference**

Inheritance diagram for Gui::ServerParser:

Collaboration diagram for Gui::ServerParser:

**Public Member Functions**

- [ServerParser](#) ()=default  
*Construct a new Server Parser object.*
- [~ServerParser](#) ()=default  
*Destroy the Server Parser object.*
- `std::vector< std::string > parse (const std::string &command)`  
*Parse the command server.*

**Public Member Functions inherited from [Gui::IServerParser](#)**

- virtual [~IServerParser](#) ()=default  
*Destroy the [IServerParser](#) object.*
- virtual `std::vector< std::string > parse (const std::string &command)=0`  
*Parse the command server.*



## 5.33.1 Constructor & Destructor Documentation

### 5.33.1.1 ServerParser()

```
Gui::ServerParser::ServerParser ( ) [default]
```

Construct a new Server Parser object.

### 5.33.1.2 ~ServerParser()

```
Gui::ServerParser::~~ServerParser ( ) [default]
```

Destroy the Server Parser object.

## 5.33.2 Member Function Documentation

### 5.33.2.1 parse()

```
std::vector< std::string > Gui::ServerParser::parse (
    const std::string & command ) [virtual]
```

Parse the command server.

#### Parameters

<i>command</i>	Command to parse.
----------------	-------------------

#### Returns

std::vector<std::string> - arguments parsed

Implements [Gui::IServerParser](#).

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ServerParser.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Parsing/ServerParser.cpp

## 5.34 Gui::Errors::ServerParserException Class Reference

[Error](#) class for network errors.

```
#include <Error.hpp>
```

Inheritance diagram for Gui::Errors::ServerParserException:

Collaboration diagram for Gui::Errors::ServerParserException:

### Public Member Functions

- [ServerParserException](#) (std::string message)

*Constructor for [ServerParserException](#).*

### Public Member Functions inherited from [Gui::Errors::AError](#)

- [~AError](#) () override=default  
*Destructor.*
- const char \* [what](#) () const noexcept override

*Returns the error message.*

### Public Member Functions inherited from [Gui::Errors::IError](#)

- virtual [~IError](#) ()=default  
*Destructor for [IError](#).*
- virtual const char \* [what](#) () const noexcept=0

*Returns a C-style string describing the error. This function must be implemented by derived classes to provide a description of the error. The returned string should be null-terminated.*

### Additional Inherited Members

### Protected Attributes inherited from [Gui::Errors::AError](#)

- std::string [\\_message](#)  
*The error message.*

#### 5.34.1 Detailed Description

[Error](#) class for network errors.

#### 5.34.2 Constructor & Destructor Documentation

##### 5.34.2.1 ServerParserException()

```
Gui::Errors::ServerParserException::ServerParserException (
    std::string message )
```

Constructor for [ServerParserException](#).

## Parameters

<i>message</i>	The error message.
----------------	--------------------

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Error/Error.cpp

## 5.35 Gui::Team Class Reference

### Public Member Functions

- **Team** (const std::string &name, Model playerModel, Model eggModel, ModelAnimation \*modelAnimation, Color playerColor)  
*Construct a new **Team** object.*
- **~Team** ()  
*Destroy the **Team** object.*
- const std::string & **getName** () const  
*Get the Name object.*
- std::vector< **Gui::Player** > & **getPlayers** ()  
*Get the Players object.*
- std::vector< **Gui::Egg** > & **getEggs** ()  
*Get the Eggs object.*
- void **setName** (const std::string &name)  
*Set the Name object.*
- void **addPlayer** (const **Gui::Player** &player)  
*Add a player to the team.*
- void **addEgg** (const **Gui::Egg** &egg)  
*Add an egg to the team.*
- bool **removePlayer** (std::size\_t id)  
*Remove a player from the team.*
- bool **removeEgg** (std::size\_t id)  
*Remove an egg from the team.*
- std::shared\_ptr< **Gui::Player** > **getPlayer** (std::size\_t id)  
*Get the **Player** object.*
- Model **getPlayerModel** () const  
*Get the Model object.*
- ModelAnimation \* **getPlayerModelAnimation** () const  
*Get the **Player** Model Animation object.*
- void **setPlayerModelPath** (const std::string &playerModelPath)  
*Set the Model object.*
- std::shared\_ptr< **Gui::Egg** > **getEgg** (std::size\_t id)  
*Get the **Egg** object.*
- Model **getEggModel** () const  
*Get the **Egg** Model Path object.*
- void **setEggModelPath** (const std::string &eggModelPath)  
*Set the **Egg** Model Path object.*

- `std::vector< BoundingBox > getPlayerBoundingBoxes` (`std::pair< size_t, size_t > pos`, `size_t orientation`, `Vector3 center`)  
*Get the [Player](#) Boundig Boxes object.*
- `Vector3 getPlayerPositionIn3DSpace` (`size_t id`, `Map< Tile > map`)  
*Get the [Player](#) position in 3D space.*
- `std::vector< RayCollision > getPlayerModelHitbox` (`size_t id`, `Camera camera`)  
*Get the [Player](#) Model hitbox.*
- `bool isPlayerHit` (`size_t id`, `Camera camera`)  
*Check if the player is hit.*
- `Color getPlayerColor () const`  
*Get the [Player](#) Color object.*

### 5.35.1 Constructor & Destructor Documentation

#### 5.35.1.1 Team()

```
Gui::Team::Team (
    const std::string & name,
    Model playerModel,
    Model eggModel,
    ModelAnimation * modelAnimation,
    Color playerColor )
```

Construct a new [Team](#) object.

##### Parameters

<i>name</i>	Name of the team.
<i>playerModel</i>	Model asset of the team.
<i>eggSkin</i>	Model asset of the eggs.
<i>playerColor</i>	Color of the players.

#### 5.35.1.2 ~Team()

```
Gui::Team::~~Team ( )
```

Destroy the [Team](#) object.

### 5.35.2 Member Function Documentation

### 5.35.2.1 addEgg()

```
void Gui::Team::addEgg (
    const Gui::Egg & egg )
```

Add an egg to the team.

#### Parameters

<i>egg</i>	Egg to add.
------------	-------------

### 5.35.2.2 addPlayer()

```
void Gui::Team::addPlayer (
    const Gui::Player & player )
```

Add a player to the team.

#### Parameters

<i>player</i>	Player to add.
---------------	----------------

### 5.35.2.3 getEgg()

```
std::shared_ptr< Gui::Egg > Gui::Team::getEgg (
    std::size_t id )
```

Get the Egg object.

#### Parameters

<i>id</i>	Id of the egg.
-----------	----------------

#### Returns

std::shared\_ptr<Gui::Egg> Egg.

### 5.35.2.4 getEggModel()

```
Model Gui::Team::getEggModel ( ) const
```

Get the Egg Model Path object.

**Returns**

const std::string& Path to the eggs Model of the team.

**5.35.2.5 getEggs()**

```
std::vector< Gui::Egg > & Gui::Team::getEggs ( )
```

Get the Eggs object.

**Returns**

std::vector<Gui::Egg>& Eggs of the team.

**5.35.2.6 getName()**

```
const std::string & Gui::Team::getName ( ) const
```

Get the Name object.

**Returns**

const std::string& Name of the team.

**5.35.2.7 getPlayer()**

```
std::shared_ptr< Gui::Player > Gui::Team::getPlayer (
    std::size_t id )
```

Get the [Player](#) object.

**Parameters**

<i>id</i>	Id of the player.
-----------	-------------------

**Returns**

std::shared\_ptr<Gui::Player> [Player](#).

#### 5.35.2.8 getPlayerBoundingBoxes()

```
std::vector< BoundingBox > Gui::Team::getPlayerBoundingBoxes (
    std::pair< size_t, size_t > pos,
    size_t orientation,
    Vector3 center )
```

Get the [Player](#) Boundig Boxes object.

##### Parameters

<i>pos</i>	Position of the player.
<i>orientation</i>	Orientation of the player.

##### Returns

std::vector<BoundingBox> Bounding boxes of the player.

#### 5.35.2.9 getPlayerColor()

```
Color Gui::Team::getPlayerColor ( ) const
```

Get the [Player](#) Color object.

#### 5.35.2.10 getPlayerModel()

```
Model Gui::Team::getPlayerModel ( ) const
```

Get the Model object.

##### Returns

Model - Model asset of the [Team](#).

#### 5.35.2.11 getPlayerModelAnimation()

```
ModelAnimation * Gui::Team::getPlayerModelAnimation ( ) const
```

Get the [Player](#) Model Animation object.

##### Returns

ModelAnimation\* - Players' animations.

#### 5.35.2.12 getPlayerModelHitbox()

```
std::vector< RayCollision > Gui::Team::getPlayerModelHitbox (
    size_t id,
    Camera camera )
```

Get the [Player](#) Model hitbox.

**Parameters**

<i>id</i>	Id of the player.
<i>camera</i>	Camera of the game.

**Returns**

std::vector<RayCollision> Hitbox of the player.

**5.35.2.13 getPlayerPositionIn3DSpace()**

```
Vector3 Gui::Team::getPlayerPositionIn3DSpace (
    size_t id,
    Map< Tile > map )
```

Get the [Player](#) position in 3D space.

**Parameters**

<i>id</i>	Id of the player.
<i>map</i>	Map of the game.

**5.35.2.14 getPlayers()**

```
std::vector< Gui::Player > & Gui::Team::getPlayers ( )
```

Get the Players object.

**Returns**

std::vector<Gui::Player>& Players of the team.

**5.35.2.15 isPlayerHit()**

```
bool Gui::Team::isPlayerHit (
    size_t id,
    Camera camera )
```

Check if the player is hit.



**Parameters**

<i>id</i>	Id of the player.
<i>camera</i>	Camera of the game.

**Returns**

true If the player is hit.

**5.35.2.16 removeEgg()**

```
bool Gui::Team::removeEgg (
    std::size_t id )
```

Remove an egg from the team.

**Parameters**

<i>id</i>	Id of the egg to remove.
-----------	--------------------------

**Returns**

true If the egg has been removed.

false If the egg has not been removed.

**5.35.2.17 removePlayer()**

```
bool Gui::Team::removePlayer (
    std::size_t id )
```

Remove a player from the team.

**Parameters**

<i>id</i>	Id of the player to remove.
-----------	-----------------------------

**Returns**

true If the player has been removed.

false If the player has not been removed.

### 5.35.2.18 setEggModelPath()

```
void Gui::Team::setEggModelPath (
    const std::string & eggModelPath )
```

Set the [Egg](#) Model Path object.

#### Parameters

<i>eggSkinPath</i>	Path to the eggs Model of the team.
--------------------	-------------------------------------

### 5.35.2.19 setName()

```
void Gui::Team::setName (
    const std::string & name )
```

Set the Name object.

#### Parameters

<i>name</i>	Name of the team.
-------------	-------------------

### 5.35.2.20 setPlayerModelPath()

```
void Gui::Team::setPlayerModelPath (
    const std::string & playerModelPath )
```

Set the Model object.

#### Parameters

<i>playerModelPath</i>	Path to the team model asset for players.
------------------------	---

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Team.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/GameDatas/Team.cpp

## 5.36 Gui::Tile Class Reference

Collaboration diagram for Gui::Tile:

## Public Member Functions

- [Tile](#) (std::pair< std::size\_t, std::size\_t > position)  
*Construct a new [Tile](#) object.*
- [Tile](#) (std::pair< std::size\_t, std::size\_t > position, [Inventory](#) inventory)  
*Construct a new [Tile](#) object.*
- [~Tile](#) ()=default  
*Destroy the [Tile](#) object.*
- void [setPosition](#) (std::pair< std::size\_t, std::size\_t > position)  
*Set the [Position](#) object.*
- std::pair< std::size\_t, std::size\_t > [getPosition](#) () const  
*Get the [Position](#) object.*
- Vector3 [getPositionIn3DSpace](#) ()  
*Get the [Position](#) In Space object.*
- std::vector< BoundingBox > [getTileBoundingBoxes](#) ([Tile](#) tile, Model tileModel)  
*Get the [Tile](#) Bounding Boxes object.*
- std::vector< RayCollision > [getTileModelHitbox](#) ([Tile](#) tile, Camera camera, Model tileModel)  
*Get the [Tile](#) Model Hitbox object.*
- bool [isTileHit](#) (Camera camera, Model \_tileModel)  
*Check if the tile is hit.*

## Public Attributes

- [Inventory](#) inventory  
*[Inventory](#) of the tile.*

### 5.36.1 Constructor & Destructor Documentation

#### 5.36.1.1 [Tile](#)() [1/2]

```
Gui::Tile::Tile (
    std::pair< std::size_t, std::size_t > position )
```

Construct a new [Tile](#) object.

##### Parameters

<i>position</i>	Set the position of the tile.
-----------------	-------------------------------

#### 5.36.1.2 [Tile](#)() [2/2]

```
Gui::Tile::Tile (
    std::pair< std::size_t, std::size_t > position,
    Inventory inventory )
```

Construct a new [Tile](#) object.

#### Parameters

<i>position</i>	Set the position of the tile.
<i>inventory</i>	Set the inventory of the tile.

#### 5.36.1.3 ~Tile()

```
Gui::Tile::~~Tile ( ) [default]
```

Destroy the [Tile](#) object.

### 5.36.2 Member Function Documentation

#### 5.36.2.1 getPosition()

```
std::pair< std::size_t, std::size_t > Gui::Tile::getPosition ( ) const
```

Get the Position object.

#### Returns

std::pair<std::size\_t, std::size\_t> - position x y

#### 5.36.2.2 getPositionIn3DSpace()

```
Vector3 Gui::Tile::getPositionIn3DSpace ( )
```

Get the Position In Space object.

#### Returns

Vector3 - Position in space.

#### 5.36.2.3 getTileBoundingBoxes()

```
std::vector< BoundingBox > Gui::Tile::getTileBoundingBoxes (
    Tile tile,
    Model tileModel )
```

Get the [Tile](#) Bounding Boxes object.

## Parameters

<i>tile</i>	Tile to get the bounding boxes.
-------------	---------------------------------

## Returns

std::vector<BoundingBox> - Bounding boxes of the tile.

**5.36.2.4   getTileModelHitbox()**

```
std::vector< RayCollision > Gui::Tile::getTileModelHitbox (
    Tile tile,
    Camera camera,
    Model tileModel )
```

Get the Tile Model Hitbox object.

## Parameters

<i>tile</i>	Tile to get the hitbox.
<i>camera</i>	Camera to get the hitbox.

## Returns

std::vector<RayCollision> - Hitbox of the tile.

**5.36.2.5   isTileHit()**

```
bool Gui::Tile::isTileHit (
    Camera camera,
    Model _tileModel )
```

Check if the tile is hit.

## Parameters

<i>camera</i>	Camera to check if the tile is hit.
<i>_tileModel</i>	Model of the tile.

## Returns

true - The tile is hit.

false - The tile is not hit.

### 5.36.2.6 setPosition()

```
void Gui::Tile::setPosition (
    std::pair< std::size_t, std::size_t > position )
```

Set the Position object.

#### Parameters

<i>position</i>	New position of the tile.
-----------------	---------------------------

## 5.36.3 Member Data Documentation

### 5.36.3.1 inventory

[Inventory](#) Gui::Tile::inventory

[Inventory](#) of the tile.

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Tile.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/GameDatas/Tile.cpp

## 5.37 Gui::UserCamera Class Reference

### Public Types

- enum **CameraType** {  
**FREE** , **FIRST\_PERSON** , **SECOND\_PERSON** , **THIRD\_PERSON** ,  
**FREE\_TILE** }

### Public Member Functions

- [UserCamera](#) ()  
Construct a new User Camera object.
- [~UserCamera](#) ()=default  
Destroy the User Camera object.
- void [setPosition](#) (Vector3 position)  
Set the Position object.
- void [setTarget](#) (Vector3 target)  
Set the Target object.
- void [setUp](#) (Vector3 up)  
Set the Up object.

- void [setFovy](#) (float fovy)  
*Set the Fovy object.*
- Vector3 [getPosition](#) (void) const  
*Get the Position object.*
- Vector3 [getTarget](#) (void) const  
*Get the Target object.*
- Vector3 [getUp](#) (void) const  
*Get the Up object.*
- float [getFovy](#) (void) const  
*Get the Fovy object.*
- std::shared\_ptr< Camera > [getCamera](#) ()  
*Get the Camera object.*
- void [setType](#) (CameraType type)  
*Set the Type object.*
- CameraType [getType](#) () const  
*Get the Type object.*
- void [setPlayerId](#) (size\_t playerId)  
*Set the *Player* Id object.*
- int [getPlayerId](#) () const  
*Get the *Player* Id object.*
- void [setTilePos](#) (std::pair< std::size\_t, std::size\_t > pos)  
*Set the *Tile* Pos object.*
- std::pair< std::size\_t, std::size\_t > [getTilePos](#) () const  
*Get the *Tile* position object.*
- bool [isPlayerPov](#) () const  
*Check if the camera is in player pov.*
- bool [isPlayerVision](#) () const  
*Check if the camera is in player vision.*
- void [setPlayerVision](#) (bool [isPlayerVision](#))  
*Set the *Player* Vision object.*

### 5.37.1 Constructor & Destructor Documentation

#### 5.37.1.1 UserCamera()

```
Gui::UserCamera::UserCamera ( )
```

Construct a new User Camera object.

#### 5.37.1.2 ~UserCamera()

```
Gui::UserCamera::~UserCamera ( ) [default]
```

Destroy the User Camera object.

## 5.37.2 Member Function Documentation

### 5.37.2.1 getCamera()

```
std::shared_ptr< Camera > Gui::UserCamera::getCamera ( )
```

Get the Camera object.

#### Returns

Camera - camera

### 5.37.2.2 getFovy()

```
float Gui::UserCamera::getFovy (
    void ) const
```

Get the Fovy object.

#### Returns

float - fovy

### 5.37.2.3 getPlayerId()

```
int Gui::UserCamera::getPlayerId ( ) const
```

Get the [Player](#) Id object.

#### Returns

int - [Player](#) id.

### 5.37.2.4 getPosition()

```
Vector3 Gui::UserCamera::getPosition (
    void ) const
```

Get the Position object.

#### Returns

Vector3 - position



#### 5.37.2.5 getTarget()

```
Vector3 Gui::UserCamera::getTarget (
    void ) const
```

Get the Target object.

##### Returns

Vector3 - target

#### 5.37.2.6 getTilePos()

```
std::pair< std::size_t, std::size_t > Gui::UserCamera::getTilePos ( ) const
```

Get the [Tile](#) position object.

##### Returns

std::pair<std::size\_t, std::size\_t> - Position of the tile.

#### 5.37.2.7 getType()

```
Gui::UserCamera::CameraType Gui::UserCamera::getType ( ) const
```

Get the Type object.

##### Returns

CameraType - Camera type.

#### 5.37.2.8 getUp()

```
Vector3 Gui::UserCamera::getUp (
    void ) const
```

Get the Up object.

##### Returns

Vector3 - up

### 5.37.2.9 isPlayerPov()

```
bool Gui::UserCamera::isPlayerPov ( ) const
```

Check if the camera is in player pov.

#### Returns

true - Camera is in player pov.  
false - Camera is not in player pov.

#### Note

The player pov is the first person, second person and third person.

### 5.37.2.10 isPlayerVision()

```
bool Gui::UserCamera::isPlayerVision ( ) const
```

Check if the camera is in player vision.

#### Returns

true - Camera is in player vision.  
false - Camera is not in player vision.

### 5.37.2.11 setFovy()

```
void Gui::UserCamera::setFovy (
    float fovy )
```

Set the Fovy object.

#### Parameters

<i>fovy</i>	New camera fovy.
-------------	------------------

### 5.37.2.12 setPlayerId()

```
void Gui::UserCamera::setPlayerId (
    size_t playerId )
```

Set the [Player](#) Id object.

## Parameters

<i>player</i> ↔ <i>Id</i>	Player id to set.
------------------------------	-------------------

**5.37.2.13 setPlayerVision()**

```
void Gui::UserCamera::setPlayerVision (
    bool isPlayerVision )
```

Set the [Player](#) Vision object.

## Parameters

<i>isPlayerVision</i>	Is player vision.
-----------------------	-------------------

**5.37.2.14 setPosition()**

```
void Gui::UserCamera::setPosition (
    Vector3 position )
```

Set the Position object.

## Parameters

<i>position</i>	New camera position.
-----------------	----------------------

**5.37.2.15 setTarget()**

```
void Gui::UserCamera::setTarget (
    Vector3 target )
```

Set the Target object.

## Parameters

<i>target</i>	New camera target.
---------------	--------------------

#### 5.37.2.16 setTilePos()

```
void Gui::UserCamera::setTilePos (
    std::pair< std::size_t, std::size_t > pos )
```

Set the [Tile](#) Pos object.

##### Parameters

<i>pos</i>	Position of the tile.
------------	-----------------------

#### 5.37.2.17 setType()

```
void Gui::UserCamera::setType (
    CameraType type )
```

Set the Type object.

##### Parameters

<i>type</i>	Type to set.
-------------	--------------

#### 5.37.2.18 setUp()

```
void Gui::UserCamera::setUp (
    Vector3 up )
```

Set the Up object.

##### Parameters

<i>up</i>	New camera up vector.
-----------	-----------------------

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/UserCamera.hpp
- /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/src/Render/UserCamera.cpp



## Chapter 6

# File Documentation

### 6.1 Assets.hpp

```
00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** Assets
00006  */
00007
00008  #pragma once
00009
00010  #include "Config.hpp"
00011
00012  #include <vector>
00013  #include <string>
00014
00015  #define PATH_ASSETS "gui/assets/"
00016
00017  #define PATH_RESOURCES "resources/"
00018  #define PATH_PLAYER "player/"
00019  #define PATH_TILE "tile/"
00020  #define PATH_DECORATION "decoration/"
00021  #define PATH_HUD "hud/"
00022  #define PATH_HELP "help/"
00023  #define PATH_END "end/"
00024
00025  #define PNG_CURSOR PATH_ASSETS PATH_HUD "cursor.png"
00026  #define PNG_END_HUD PATH_ASSETS PATH_END "end_hud.png"
00027  #define PNG_HUD PATH_ASSETS PATH_HUD "hud.png"
00028  #define PNG_FOOD PATH_ASSETS PATH_HUD "food.png"
00029  #define PNG_LINEMATE PATH_ASSETS PATH_HUD "linemate.png"
00030  #define PNG_DERAUMERE PATH_ASSETS PATH_HUD "deraumere.png"
00031  #define PNG_MENDIANE PATH_ASSETS PATH_HUD "mendiane.png"
00032  #define PNG_PHIRAS PATH_ASSETS PATH_HUD "phiras.png"
00033  #define PNG_SIBUR PATH_ASSETS PATH_HUD "sibur.png"
00034  #define PNG_THYSTAME PATH_ASSETS PATH_HUD "thystame.png"
00035  #define PNG_PLAYER PATH_ASSETS PATH_HUD "player.png"
00036  #define PNG_TILE PATH_ASSETS PATH_HUD "tile.png"
00037  #define PNG_EGG PATH_ASSETS PATH_HUD "egg.png"
00038  #define FONT_HUD PATH_ASSETS PATH_HUD "SimplyMono-Bold.ttf"
00039
00040  #define PNG_HELP PATH_ASSETS PATH_HELP "help_menu.png"
00041  #define PNG_HELP_KEYS PATH_ASSETS PATH_HELP "keys.png"
00042  #define PNG_HUD PATH_ASSETS PATH_HUD "hud.png"
00043  #define PNG_FOOD PATH_ASSETS PATH_HUD "food.png"
00044  #define PNG_LINEMATE PATH_ASSETS PATH_HUD "linemate.png"
00045  #define PNG_DERAUMERE PATH_ASSETS PATH_HUD "deraumere.png"
00046  #define PNG_MENDIANE PATH_ASSETS PATH_HUD "mendiane.png"
00047  #define PNG_PHIRAS PATH_ASSETS PATH_HUD "phiras.png"
00048  #define PNG_SIBUR PATH_ASSETS PATH_HUD "sibur.png"
00049  #define PNG_THYSTAME PATH_ASSETS PATH_HUD "thystame.png"
00050  #define PNG_PLAYER PATH_ASSETS PATH_HUD "player.png"
00051  #define PNG_TILE PATH_ASSETS PATH_HUD "tile.png"
00052  #define FONT_HUD PATH_ASSETS PATH_HUD "SimplyMono-Bold.ttf"
00053
00054  #define MODEL_TILE PATH_ASSETS PATH_TILE "tile.glb"
00055  #define MODEL_FOOD PATH_ASSETS PATH_RESOURCES "food.glb"
00056  #define MODEL_LINEMATE PATH_ASSETS PATH_RESOURCES "linemate.glb"
00057  #define MODEL_MENDIANE PATH_ASSETS PATH_RESOURCES "mendiane.glb"
00058  #define MODEL_PHIRAS PATH_ASSETS PATH_RESOURCES "phiras.glb"
```

```

00059 #define MODEL_SIBUR                PATH_ASSETS PATH_RESOURCES "sibur.glb"
00060 #define MODEL_THYSTAME                PATH_ASSETS PATH_RESOURCES "thystame.glb"
00061 #define MODEL_DERAUMERE               PATH_ASSETS PATH_RESOURCES "deraumere.glb"
00062 #define MODEL_PLAYER                  PATH_ASSETS PATH_PLAYER "player.glb"
00063 #define MODEL_EGG                     PATH_ASSETS PATH_PLAYER "egg.glb"
00064 #define MODEL_TREE                    PATH_ASSETS PATH_DECORATION "tree.glb"
00065 #define MODEL_LANTERN                 PATH_ASSETS PATH_DECORATION "lantern.glb"
00066
00067 #define SCALE_FOOD                     (Vector3){1, 0.5, 1}
00068 #define SCALE_LINEMATE                 (Vector3){0.1, 0.1, 0.1}
00069 #define SCALE_MENDIANE                 (Vector3){0.1, 0.1, 0.1}
00070 #define SCALE_PHIRAS                   (Vector3){0.001, 0.001, 0.005}
00071 #define SCALE_SIBUR                   (Vector3){0.01, 0.01, 0.01}
00072 #define SCALE_THYSTAME                 (Vector3){2, 2, 2}
00073 #define SCALE_DERAUMERE                (Vector3){0.5, 0.5, 0.5}
00074 #define SCALE_PLAYER                   (Vector3){0.5, 0.5, 0.5}
00075 #define SCALE_EGG                      (Vector3){0.5, 0.5, 0.5}
00076 #define SCALE_TREE                     (Vector3){1, 1, 1}
00077 #define SCALE_LANTERN                  (Vector3){1, 1, 1}
00078
00079 #define ROTATION_ANGLE_FOOD             45
00080 #define ROTATION_ANGLE_LINEMATE        270
00081 #define ROTATION_ANGLE_MENDIANE        0
00082 #define ROTATION_ANGLE_PHIRAS          270
00083 #define ROTATION_ANGLE_SIBUR           270
00084 #define ROTATION_ANGLE_THYSTAME         270
00085 #define ROTATION_ANGLE_DERAUMERE        90
00086 #define ROTATION_ANGLE_PLAYER           0
00087 #define ROTATION_ANGLE_EGG              270
00088 #define ROTATION_ANGLE_TREE             0
00089 #define ROTATION_ANGLE_LANTERN          0
00090
00091 #define ROTATION_AXIS_FOOD              (Vector3){0, 1, 0}
00092 #define ROTATION_AXIS_LINEMATE          (Vector3){1, 0, 0}
00093 #define ROTATION_AXIS_MENDIANE          (Vector3){1, 0, 0}
00094 #define ROTATION_AXIS_PHIRAS            (Vector3){1, 0, 0}
00095 #define ROTATION_AXIS_SIBUR             (Vector3){1, 0, 0}
00096 #define ROTATION_AXIS_THYSTAME          (Vector3){1, 0, 0}
00097 #define ROTATION_AXIS_DERAUMERE          (Vector3){1, 0, 0}
00098 #define ROTATION_AXIS_PLAYER             (Vector3){0, 1, 0}
00099 #define ROTATION_AXIS_EGG                (Vector3){1, 0, 0}
00100 #define ROTATION_AXIS_TREE               (Vector3){1, 0, 0}
00101 #define ROTATION_AXIS_LANTERN            (Vector3){1, 0, 0}
00102
00103 #define POS_FOOD                        (Vector3){0.5, -0.1, 1.5}
00104 #define POS_LINEMATE                    (Vector3){1, -0.3, -0.5}
00105 #define POS_MENDIANE                    (Vector3){2, -0.25, -0.5}
00106 #define POS_PHIRAS                      (Vector3){0.5, -0.3, -1.5}
00107 #define POS_SIBUR                       (Vector3){1.5, -0.3, -1.5}
00108 #define POS_THYSTAME                     (Vector3){1, 0, -2}
00109 #define POS_DERAUMERE                    (Vector3){2, -0.3, -2}
00110 #define POS_PLAYER                       (Vector3){0, -0.25, 0}
00111 #define POS_EGG                          (Vector3){0.5, 0, 0.5}
00112 #define POS_TREE                         (Vector3){2, -0.3, 2}
00113 #define POS_LANTERN                      (Vector3){1, -0.3, 2}
00114 #define POS_Y_DELIMITATION              -0.27f
00115
00116 #define PLAYER_TEXT_SIZE                 40
00117 #define PLAYER_TEXT_SIZE_RATIO           1.5f
00118
00119 #define HELP_BACKGROUND_SCALE             3.5f
00120 #define HELP_BACKGROUND_TEXTURE_SIZE      (Vector2){400 * HELP_BACKGROUND_SCALE, 200 *
HELP_BACKGROUND_SCALE}
00121 #define HELP_BACKGROUND_POSITION          (Vector2){WINDOW_WIDTH / 2 - HELP_BACKGROUND_TEXTURE_SIZE.x /
2, WINDOW_HEIGHT / 2 - HELP_BACKGROUND_TEXTURE_SIZE.y / 2}
00122 #define HELP_TEXT_POSITION_LEFT           (Vector2){HELP_BACKGROUND_POSITION.x + 270,
HELP_BACKGROUND_POSITION.y + 180}
00123 #define HELP_TEXT_POSITION_RIGHT          (Vector2){HELP_TEXT_POSITION_LEFT.x + 610,
HELP_TEXT_POSITION_LEFT.y}
00124 #define HELP_TEXT_SPACING                 30
00125 #define HELP_TITLE_OFFSET                 90
00126
00127 #define END_HUD_TEXT_SIZE                 40
00128 #define END_HUD_ADVICE_TEXT_SIZE          30
00129 #define END_HUD_ADVICE_OFFSET             150
00130
00131 static std::vector<std::string> globalControlsTexts = {
00132     "Move the camera in the space",
00133     "Move the camera up",
00134     "Move the camera down",
00135     "Switch on/off the debug display",
00136     "Reset the HUD from tile view to game view",
00137     "Switch on/off the help HELP",
00138     "Increase the render distance",
00139     "Decrease the render distance",
00140     "Increase the time unit",
00141     "Decrease the time unit",

```



```

00142     "Select the player point of view",
00143     "Select the tile to display its HUD",
00144 };
00145
00146 static std::vector<std::string> playerControlsTexts = {
00147     "Change the player point of view",
00148     "Get off the player point of view",
00149     "Switch to the next player",
00150     "Switch to the previous player",
00151     "Switch on/off the player vision",
00152 };

```

## 6.2 Colors.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** Colors
00006 */
00007
00008 #pragma once
00009
00010 #include "raylib.h"
00011
00012 #include <vector>
00013
00014 #define STR_BLUE "\033[0;34m"
00015 #define STR_GREEN "\033[0;32m"
00016 #define STR_RED "\033[0;31m"
00017 #define STR_YELLOW "\033[0;33m"
00018 #define STR_VIOLET "\033[0;35m"
00019 #define STR_CYAN "\033[0;36m"
00020 #define STR_RESET "\033[0m"
00021
00022 static std::vector<Color> playerColors = {
00023     PINK,
00024     LIGHTGRAY,
00025     GRAY,
00026     DARKGRAY,
00027     YELLOW,
00028     GOLD,
00029     ORANGE,
00030     RED,
00031     MAROON,
00032     GREEN,
00033     LIME,
00034     DARKGREEN,
00035     SKYBLUE,
00036     BLUE,
00037     DARKBLUE,
00038     PURPLE,
00039     VIOLET,
00040     DARKPURPLE,
00041     BEIGE,
00042     BROWN,
00043     DARKBROWN,
00044     WHITE,
00045     BLACK,
00046     MAGENTA,
00047     RAYWHITE
00048 };

```

## 6.3 Config.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** Config
00006 */
00007
00008 #pragma once
00009
00010 #define WINDOW_WIDTH 1920
00011 #define WINDOW_HEIGHT 1080
00012 #define WINDOW_TITLE "Zappy GUI"
00013
00014 #define SERVER_DOWN_MESSAGE "THE GUI MUST BE CLOSED FOR UNEXPECTED REASONS"

```

```

00015
00016 #define SIZE_TILE 4.7
00017
00018 #define PLAYER_HEIGHT 2
00019
00020 #define DEFAULT_RENDER_DISTANCE 15
00021 #define MAX_RENDER_DISTANCE 20
00022 #define MIN_RENDER_DISTANCE 1
00023
00024 #define HIGH_CAMERA_INCREASE 0.1
00025 #define LOW_CAMERA_INCREASE 0.1
00026
00027 #define PLAYER_SECOND_PERSON_FOV 4.0f
00028 #define PLAYER_THIRD_PERSON_FOV 5.0f

```

## 6.4 Engine.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Engine
00006 */
00007
00008 #pragma once
00009
00010 #include "Event/Event.hpp"
00011 #include "Render/Render.hpp"
00012 #include "Network/INetwork.hpp"
00013 #include "GameDatas/GameData.hpp"
00014 #include "Parsing/ServerParser.hpp"
00015 #include "GUIUpdater/GUIUpdater.hpp"
00016
00017 #include <thread>
00018 #include <time.h>
00019
00020 #define TIME_UNIT_MAP_UPDATE 20
00021
00022 namespace Gui {
00023     class Engine;
00024 };
00025
00026 class Gui::Engine {
00027     public:
00028         Engine(std::shared_ptr<INetwork> network);
00029         ~Engine() = default;
00030         void run();
00031     private:
00032         std::unique_ptr<IServerParser> _parser;
00033         std::shared_ptr<INetwork> _network;
00034         std::shared_ptr<Render> _render;
00035         std::unique_ptr<IEvent> _event;
00036         std::shared_ptr<GameData> _gameData;
00037         std::unique_ptr<IGUIUpdater> _guiUpdater;
00038         std::thread _networkThread;
00039
00040         void listenServer();
00041         void sendMessageUpdate();
00042         void updateMap();
00043         void sendUpdateMapMessage();
00044         void threadLoop();
00045 };

```

## 6.5 AError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI

```

```

00004  ** File description:
00005  ** AError
00006  */
00007
00008
00009 #pragma once
00010
00011 #include "Error/IError.hpp"
00012
00013 #include <string>
00014
00015 namespace Gui {
00016     namespace Errors {
00023         class AError;
00024     };
00025 };
00026
00027 class Gui::Errors::AError : public IError {
00028
00029     public:
00030
00034         ~AError() override = default;
00035
00040         const char *what() const noexcept override;
00041
00042     protected:
00043
00044         std::string _message;
00045 };

```

## 6.6 Error.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** Error
00006  */
00007
00008 #pragma once
00009
00010 #include "Error/AError.hpp"
00011
00012 namespace Gui {
00013     namespace Errors {
00014
00019         class Error : public AError {};
00020
00025         class NetworkException : public Error {
00026
00027             public:
00033                 NetworkException(std::string message);
00034         };
00035
00040         class ServerParserException : public Error {
00041
00042             public:
00048                 ServerParserException(std::string message);
00049         };
00050
00055         class ParseCommandLineException : public Error {
00056
00057             public:
00063                 ParseCommandLineException(std::string message);
00064         };
00065
00070         class GuiGameDataException : public Error {
00071
00072             public:
00078                 GuiGameDataException(std::string message);
00079         };
00080
00085         class GuiUpdaterException : public Error {
00086
00087             public:
00093                 GuiUpdaterException(std::string message);
00094         };
00095     };
00096 };

```

## 6.7 IError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** IError
00006 */
00007
00008 #pragma once
00009
00010 #include <exception>
00011
00012 namespace Gui {
00013     namespace Errors {
00014         class IError;
00015     };
00016 };
00017
00018 class Gui::Errors::IError : public std::exception {
00019 public:
00020     virtual ~IError() = default;
00021     virtual const char *what() const noexcept = 0;
00022 };

```

## 6.8 AEvent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** AEvent
00006 */
00007
00008 #pragma once
00009
00010 #include "Event/IEvent.hpp"
00011
00012 namespace Gui {
00013     class AEvent;
00014 }
00015
00016 class Gui::AEvent : public Gui::IEvent {
00017 public:
00018     AEvent();
00019     ~AEvent() = default;
00020     virtual void listen() = 0;
00021     void setRender(std::shared_ptr<Render> render);
00022     void setGameData(std::shared_ptr<GameData> gameData);
00023     std::shared_ptr<Render> getRender();
00024     std::shared_ptr<GameData> getGameData();
00025 protected:
00026     std::shared_ptr<Render> _render;
00027     std::shared_ptr<GameData> _gameData;
00028 };

```

## 6.9 Event.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Event
00006 */
00007

```

```

00008 #pragma once
00009
00010 #include "Config.hpp"
00011 #include "Event/AEvent.hpp"
00012 #include "Render/Render.hpp"
00013
00014 #include <functional>
00015 #include <unordered_map>
00016
00017 namespace Gui {
00018
00023     class Event;
00024 };
00025
00026 class Gui::Event : public Gui::AEvent {
00027
00028     public:
00029
00034         Event();
00035
00040         ~Event() = default;
00041
00046         void listen();
00047
00048     private:
00049
00054         std::unordered_map<KeyboardKey, std::function<void()>> _eventsKeyDown =
00055         {
00056             {KEY_SPACE, [this]() {moveUpCamera();}},
00057             {KEY_LEFT_SHIFT, [this]() {moveDownCamera();}},
00058         };
00059
00060         std::unordered_map<GamepadButton, std::function<void()>> _eventsGamepadButtonDown =
00061         {
00062             {GAMEPAD_BUTTON_RIGHT_TRIGGER_2, [this]() {handleSpaceGamepad();}},
00063             {GAMEPAD_BUTTON_LEFT_TRIGGER_2, [this]() {moveDownCamera();}},
00064         };
00065
00070         std::unordered_map<KeyboardKey, std::function<void()>> _eventsKeyPressed =
00071         {
00072             {KEY_THREE, [this]() {switchDisplayDebug();}},
00073             {KEY_F3, [this]() {switchDisplayDebug();}},
00074             {KEY_SPACE, [this]() {setFreeCam();}},
00075             {KEY_R, [this]() {switchTileHudToGame();}},
00076             {KEY_J, [this]() {increaseRenderDistance();}},
00077             {KEY_K, [this]() {decreaseRenderDistance();}},
00078             {KEY_F5, [this]() {changeActualPlayerPov();}},
00079             {KEY_FOUR, [this]() {changeActualPlayerPov();}},
00080             {KEY_V, [this]() {setPlayerVision();}},
00081             {KEY_KP_ADD, [this]() {increaseTimeUnit();}},
00082             {KEY_KP_SUBTRACT, [this]() {decreaseTimeUnit();}},
00083             {KEY_H, [this]() {displayHelpMenu();}},
00084         };
00085
00086         std::unordered_map<GamepadButton, std::function<void()>> _eventsGamepadButtonPressed =
00087         {
00088             {GAMEPAD_BUTTON_RIGHT_FACE_DOWN, [this]() {handleLeftClick();}},
00089             {GAMEPAD_BUTTON_RIGHT_TRIGGER_1, [this]() {handleLeftClick();}},
00090             {GAMEPAD_BUTTON_LEFT_TRIGGER_1, [this]() {handleRightClick();}},
00091             {GAMEPAD_BUTTON_LEFT_FACE_UP, [this]() {increaseRenderDistance();}},
00092             {GAMEPAD_BUTTON_LEFT_FACE_DOWN, [this]() {decreaseRenderDistance();}},
00093             {GAMEPAD_BUTTON_LEFT_FACE_LEFT, [this]() {decreaseTimeUnit();}},
00094             {GAMEPAD_BUTTON_LEFT_FACE_RIGHT, [this]() {increaseTimeUnit();}},
00095             {GAMEPAD_BUTTON_RIGHT_FACE_LEFT, [this]() {switchDisplayDebug();}},
00096             {GAMEPAD_BUTTON_RIGHT_FACE_RIGHT, [this]() {switchTileHudToGame();}},
00097             {GAMEPAD_BUTTON_RIGHT_FACE_UP, [this]() {changeActualPlayerPov();}},
00098             {GAMEPAD_BUTTON_MIDDLE_RIGHT, [this]() {closeWindowGamepad();}},
00099         };
00100
00105         std::unordered_map<MouseButton, std::function<void()>> _eventsMousePressed =
00106         {
00107             {MOUSE_BUTTON_LEFT, [this]() {handleLeftClick();}},
00108             {MOUSE_BUTTON_RIGHT, [this]() {handleRightClick();}},
00109         };
00110
00115         void moveUpCamera();
00116
00121         void moveDownCamera();
00122
00127         void switchDisplayDebug();
00128
00132         void setFreeCam();
00133
00138         void handleLeftClick();
00139
00143         void handleRightClick();
00144

```

```

00149         void selectPlayer();
00150
00154         void selectTile();
00155
00161         void changePlayer(bool turn);
00162
00167         void changeActualPlayerPov();
00168
00173         void switchTileHudToGame();
00174
00179         void increaseRenderDistance();
00180
00185         void decreaseRenderDistance();
00186
00191         void increaseTimeUnit();
00192
00197         void decreaseTimeUnit();
00198
00203         void displayHelpMenu();
00204
00208         void handleSpaceGamepad();
00209
00213         void closeWindowGamepad();
00214
00219         void setPlayerVision();
00220 };

```

## 6.10 IEvent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** IEvent
00006 */
00007
00008 #pragma once
00009
00010 #include "Render/Render.hpp"
00011 #include "GameDatas/GameData.hpp"
00012 #include <memory>
00013
00014 namespace Gui {
00015
00020     class IEvent;
00021 }
00022
00023 class Gui::IEvent {
00024
00025     public:
00026
00031         IEvent() = default;
00032
00037         virtual ~IEvent() = default;
00038
00043         virtual void listen() = 0;
00044
00050         virtual void setRender(std::shared_ptr<Render> render) = 0;
00051
00057         virtual void setGameData(std::shared_ptr<GameData> gameData) = 0;
00058 };

```

## 6.11 Egg.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** Egg
00006 */
00007
00008 #pragma once
00009
00010 #include "Error/Error.hpp"
00011
00012 #include <string>
00013
00014 namespace Gui {
00015

```

```

00019     class Egg;
00020 };
00021
00022 class Gui::Egg {
00023     public:
00024
00025         enum EggState {
00026             IDLE,
00027             DEAD,
00028             BORN,
00029             HATCHING,
00030         };
00031
00032         Egg(size_t id, const std::string &team, std::pair<std::size_t, std::size_t> position);
00033
00034         ~Egg();
00035
00036         std::size_t getId() const;
00037
00038         std::string getTeam() const;
00039
00040         std::pair<std::size_t, std::size_t> getPosition() const;
00041
00042         void setId(std::size_t id);
00043
00044         void setTeam(const std::string &team);
00045
00046         void setPosition(std::pair<std::size_t, std::size_t> position);
00047
00048         void setState(EggState state);
00049
00050         EggState getState() const;
00051
00052     private:
00053
00054         std::size_t _id;
00055         std::string _team;
00056         std::pair<std::size_t, std::size_t> _position;
00057         EggState _state;
00058 };

```

## 6.12 GameData.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** GameData
00006  */
00007
00008 #pragma once
00009
00010 #include "Types.hpp"
00011 #include "Error/Error.hpp"
00012 #include "GameDatas/Team.hpp"
00013 #include "GameDatas/Tile.hpp"
00014
00015 #define NO_TICK int(-1)
00016
00017 namespace Gui {
00018
00019     class GameData;
00020 };
00021
00022 class Gui::GameData {
00023     public:
00024
00025         enum TimeUnitState {
00026             INCREASE,
00027             DECREASE,
00028             NONE
00029         };
00030
00031         GameData();
00032
00033         ~GameData() = default;
00034
00035         std::vector<Gui::Team> &getTeams();
00036
00037         Gui::Team &getTeam(const std::string &name);
00038 };

```

```

00068     void addTeam(const Gui::Team &team);
00069
00076     void addTeam(const std::string &name, Color playerColor);
00077
00084     void addPlayerToTeam(const std::string &teamName, const Gui::Player &player);
00085
00091     Gui::Player &getPlayer(size_t id);
00092
00098     Map<Gui::Tile> &getMap();
00099
00105     void setMap(const Map<Gui::Tile> &map);
00106
00114     void setMapSize(size_t x, size_t y);
00115
00121     std::pair<size_t, size_t> getMapSize() const;
00122
00130     Gui::Tile &getTile(size_t x, size_t y);
00131
00139     void setTile(const Gui::Tile &tile);
00140
00145     void restartLastTick(void);
00146
00152     void setServerTick(std::size_t tick);
00153
00159     clock_t getLastTick() const;
00160
00166     std::size_t getServerTick() const;
00167
00173     void setIsEndGame(bool isEndGame);
00174
00181     bool getIsEndGame() const;
00182
00188     void setLastError(const std::string &error);
00189
00195     std::string getLastError() const;
00196
00203     Team &getTeamById(std::size_t id);
00204
00210     TimeUnitState getTimeUnitFromServer() const;
00211
00217     void setTimeUnitFromServer(TimeUnitState timeUnitFromServer);
00218
00224     std::vector<Gui::Egg> &getServerEggs();
00225
00231     void addServerEgg(const Gui::Egg &egg);
00232
00238     void removeServerEgg(size_t id);
00239
00245     void setNbBCTCommandReceived(std::size_t nb);
00246
00252     std::size_t getNbBCTCommandReceived() const;
00253
00258     void restartLastTickMctCommand();
00259
00265     clock_t getLastTickMctCommand() const;
00266
00272     void setEndMessage(const std::string &endMessage);
00273
00279     std::string getEndMessage() const;
00280
00286     void setPlayerModel(const Model &playerModel);
00287
00293     void setEggModel(const Model &eggModel);
00294
00300     void setPlayerModelAnimation(ModelAnimation *playerModelAnimation);
00301
00307     void setAnimsCount(int animsCount);
00308
00314     void setServerError(bool isServerError);
00315
00321     bool getServerError() const;
00322
00323     private:
00324
00325         std::vector<Gui::Team>      _teams;
00326         Map<Gui::Tile>              _map;
00327         std::size_t                 _serverTick;
00328         clock_t                     _lastTick;
00329         bool                        _isEndGame;
00330         std::size_t                 _nbBCTCommandReceived;
00331         clock_t                     _lastTickMctCommand;
00332         std::string                  _lastError;
00333         TimeUnitState                _timeUnitFromServer;
00334         std::vector<Gui::Egg>       _serverEggs;
00335         std::string                  _endMessage;
00336         Model                       _playerModel;
00337         Model                       _eggModel;

```



```

00338         ModelAnimation*           _playerModelAnimation;
00339         int                       _animCount;
00340         bool                      _isServerError;
00341     };

```

## 6.13 Inventory.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** Inventory
00006  */
00007
00008  #pragma once
00009
00010  #include <iostream>
00011
00012  #define RessourcesNumber 6
00013
00014  namespace Gui {
00015
00020      class Inventory;
00021  };
00022
00023  class Gui::Inventory {
00024
00025      public:
00026
00027          using Ressources = size_t [RessourcesNumber];
00028
00033          Inventory();
00034
00046          Inventory(std::size_t food, std::size_t linemate, std::size_t deraumere, std::size_t sibur,
std::size_t mendiane, std::size_t phiras, std::size_t thystame);
00047
00052          ~Inventory() = default;
00053
00059          void setFood(std::size_t food);
00060
00066          void setLinemate(std::size_t linemate);
00067
00073          void setDeraumere(std::size_t deraumere);
00074
00080          void setSibur(std::size_t sibur);
00081
00087          void setMendiane(std::size_t mendiane);
00088
00094          void setPhiras(std::size_t phiras);
00095
00101          void setThystame(std::size_t thystsame);
00102
00108          void setRessources(Ressources ressources);
00109
00115          std::size_t getFood(void);
00116
00122          std::size_t getLinemate(void);
00123
00129          std::size_t getDeraumere(void);
00130
00136          std::size_t getSibur(void);
00137
00143          std::size_t getMendiane(void);
00144
00150          std::size_t getPhiras(void);
00151
00157          std::size_t getThystame(void);
00158
00164          Ressources &getRessources(void);
00165
00179          void addResource(std::size_t resource, std::size_t quantity);
00180
00194          void removeResource(std::size_t resource, std::size_t quantity);
00195
00196      private:
00197
00198          std::size_t      _food;
00199          Ressources      _ressources;
00200  };

```

## 6.14 Player.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** Player
00006  */
00007
00008  #pragma once
00009
00010  #include "raylib.h"
00011  #include "GameDatas/Inventory.hpp"
00012
00013  namespace Gui {
00014
00015      class Player;
00016  };
00017
00018  class Gui::Player {
00019  public:
00020
00021      enum PlayerState {
00022          IDLE = 2,
00023          BORN = 8,
00024          BROADCAST = 12,
00025          EJECT = 5,
00026          BEING_EJECTED = 15,
00027          EJECTED = 7,
00028          WALK = 6, // or 10
00029          INCANTATION = 0,
00030          LAY_EGG = 7,
00031          DROP = 9,
00032          COLLECT = 9,
00033          DEAD = 1,
00034      };
00035
00036      Player(std::size_t id, const std::string &team, std::pair<std::size_t, std::size_t> position,
00037             std::size_t orientation, std::size_t level = 1);
00038
00039      ~Player() = default;
00040
00041      void setPosition(std::pair<std::size_t, std::size_t> position);
00042
00043      void setPosition3D(Vector3 position3D);
00044
00045      void setId(std::size_t id);
00046
00047      void setLevel(std::size_t level);
00048
00049      void setOrientation(std::size_t orientation);
00050
00051      void setTeam(const std::string &team);
00052
00053      std::pair<std::size_t, std::size_t> getPosition(void) const;
00054
00055      Vector3 getPosition3D(void) const;
00056
00057      std::size_t getId(void) const;
00058
00059      std::size_t getLevel(void) const;
00060
00061      std::size_t getOrientation(void) const;
00062
00063      std::string getTeam(void) const;
00064
00065      void setState(PlayerState state);
00066
00067      PlayerState getState(void) const;
00068
00069      void setBroadcast(const std::string &broadcast);
00070
00071      std::string getBroadcast() const;
00072
00073      float getRotationFromOrientation() const;
00074
00075      Vector3 getCenterPosition();
00076
00077      void setCurrentFrame(int currentFrame);
00078
00079      int getCurrentFrame() const;
00080
00081      void restartAnimationTimeEllapsed();
00082
00083      clock_t getAnimationTimeEllapsed() const;
00084

```

```

00213         Inventory                                inventory;
00214
00215     private:
00216
00217         std::size_t                                _id;
00218         std::string                                _team;
00219         std::pair<std::size_t, std::size_t>         _position;
00220         Vector3                                     _position3D;
00221         std::size_t                                _orientation;
00222         std::size_t                                _level;
00223         PlayerState                                 _state;
00224         std::string                                 _broadcast;
00225         int                                          _currentFrame;
00226         clock_t                                     _animationTimeEllapsed;
00227     };

```

## 6.15 Team.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** Team
00006  */
00007
00008  #pragma once
00009
00010  #include "raylib.h"
00011  #include "Types.hpp"
00012  #include "GameDatas/Egg.hpp"
00013  #include "GameDatas/Tile.hpp"
00014  #include "GameDatas/Player.hpp"
00015
00016  #include <vector>
00017  #include <memory>
00018
00019  namespace Gui {
00020
00021      class Team;
00022  };
00023
00024  class Gui::Team {
00025  public:
00026
00027      Team(const std::string &name, Model playerModel, Model eggModel, ModelAnimation
00028      *modelAnimation, Color playerColor);
00029
00030      ~Team();
00031
00032      const std::string &getName() const;
00033
00034      std::vector<Gui::Player> &getPlayers();
00035
00036      std::vector<Gui::Egg> &getEggs();
00037
00038      void setName(const std::string &name);
00039
00040      void addPlayer(const Gui::Player &player);
00041
00042      void addEgg(const Gui::Egg &egg);
00043
00044      bool removePlayer(std::size_t id);
00045
00046      bool removeEgg(std::size_t id);
00047
00048      std::shared_ptr<Gui::Player> getPlayer(std::size_t id);
00049
00050      Model getPlayerModel() const;
00051
00052      ModelAnimation *getPlayerModelAnimation() const;
00053
00054      void setPlayerModelPath(const std::string &playerModelPath);
00055
00056      std::shared_ptr<Gui::Egg> getEgg(std::size_t id);
00057
00058      Model getEggModel() const;
00059
00060      void setEggModelPath(const std::string &eggModelPath);
00061
00062      std::vector<BoundingBox> getPlayerBoundingBoxes(std::pair<size_t, size_t> pos, size_t
00063      orientation, Vector3 center);
00064
00065  };

```

```

00174     Vector3 getPlayerPositionIn3DSpace(size_t id, Map<Tile> map);
00175
00183     std::vector<RayCollision> getPlayerModelHitbox(size_t id, Camera camera);
00184
00192     bool isPlayerHit(size_t id, Camera camera);
00193
00198     Color getPlayerColor() const;
00199
00200     private:
00201
00202         ModelAnimation* _modelAnimation;
00203         int _animCount;
00204         std::string _name;
00205         std::vector<Gui::Player> _players;
00206         Model _playerModel;
00207         std::vector<Gui::Egg> _eggs;
00208         Model _eggModel;
00209         Color _playerColor;
00210
00220         BoundingBox rotateBoundingBoxByOrientation(BoundingBox bbox, size_t orientation,
00221 std::pair<size_t, size_t> pos, Vector3 center);
00221 };

```

## 6.16 Tile.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Tile
00006 */
00007
00008 #pragma once
00009
00010 #include "raylib.h"
00011 #include "GameDatas/Inventory.hpp"
00012
00013 #include <vector>
00014
00015 namespace Gui {
00016
00021     class Tile;
00022 };
00023
00024 class Gui::Tile {
00025
00026     public:
00027
00033         Tile(std::pair<std::size_t, std::size_t> position);
00034
00041         Tile(std::pair<std::size_t, std::size_t> position, Inventory inventory);
00042
00047         ~Tile() = default;
00048
00054         void setPosition(std::pair<std::size_t, std::size_t> position);
00055
00061         std::pair<std::size_t, std::size_t> getPosition() const;
00062
00068         Vector3 getPositionIn3DSpace();
00069
00076         std::vector<BoundingBox> getTileBoundingBoxes(Tile tile, Model tileModel);
00077
00085         std::vector<RayCollision> getTileModelHitbox(Tile tile, Camera camera, Model tileModel);
00086
00095         bool isTileHit(Camera camera, Model _tileModel);
00096
00101         Inventory inventory;
00102
00103     private:
00104
00105         std::pair<std::size_t, std::size_t> _position;
00106         Vector3 _positionIn3DSpace;
00107 };

```

## 6.17 AGUIUpdater.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy

```

```

00004  ** File description:
00005  ** AGUIUpdater
00006  */
00007
00008  #pragma once
00009
00010  #include "Network/Network.hpp"
00011  #include "GameDatas/GameData.hpp"
00012  #include "GUIUpdater/IGUIUpdater.hpp"
00013
00014  #include <memory>
00015
00016  namespace Gui {
00017
00021      class AGUIUpdater;
00022  }
00023
00024  class Gui::AGUIUpdater : public Gui::IGUIUpdater {
00025  public:
00026
00033      AGUIUpdater(std::shared_ptr<GameData> gameData, std::shared_ptr<INetwork> network);
00034
00038      ~AGUIUpdater() = default;
00039
00044      void update(const std::string &command, const std::vector<std::string> &data) override = 0;
00045
00046  protected:
00047
00048      std::shared_ptr<GameData>          _gameData;
00049      std::shared_ptr<INetwork>         _network;
00050  };

```

## 6.18 GUIUpdater.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** GUIUpdater
00006  */
00007
00008  #pragma once
00009
00010  #include "GUIUpdater/AGUIUpdater.hpp"
00011
00012  #include <string>
00013  #include <functional>
00014  #include <unordered_map>
00015
00016  namespace Gui {
00017
00023      class GUIUpdater;
00024  }
00025
00026  class Gui::GUIUpdater : public Gui::AGUIUpdater {
00027  public:
00028
00029
00036      GUIUpdater(std::shared_ptr<GameData> gameData, std::shared_ptr<INetwork> network);
00037
00041      ~GUIUpdater() = default;
00042
00049      void update(const std::string &command, const std::vector<std::string> &data);
00050
00051  private:
00052
00053      size_t          _colorIndex;
00054
00055      std::unordered_map<std::string, std::function<void(std::vector<std::string>)>> _updateMap =
00056      {
00057          {"msz", [this](std::vector<std::string> data) {updateMapSize(data);}},
00058          {"bct", [this](std::vector<std::string> data) {updateMapContent(data);}},
00059          {"tna", [this](std::vector<std::string> data) {updateTeamNames(data);}},
00060          {"pnw", [this](std::vector<std::string> data) {updateTeamMember(data);}},
00061          {"ppo", [this](std::vector<std::string> data) {updatePlayerPosition(data);}},
00062          {"plv", [this](std::vector<std::string> data) {updatePlayerLevel(data);}},
00063          {"pin", [this](std::vector<std::string> data) {updatePlayerInventory(data);}},
00064          {"pex", [this](std::vector<std::string> data) {updatePlayerExpulsion(data);}},
00065          {"pbc", [this](std::vector<std::string> data) {updatePlayerBroadcast(data);}},
00066          {"pic", [this](std::vector<std::string> data) {updatePlayerStartIncantation(data);}},
00067          {"pie", [this](std::vector<std::string> data) {updatePlayerEndIncantation(data);}},
00068          {"pfk", [this](std::vector<std::string> data) {updatePlayerEggLaying(data);}},
00069          {"pdr", [this](std::vector<std::string> data) {updatePlayerResourceDropping(data);}},

```

```

00070         {"pgt", [this] (std::vector<std::string> data) {updatePlayerRessourceCollecting(data);}},
00071         {"pdi", [this] (std::vector<std::string> data) {updatePlayerDeath(data);}},
00072         {"enw", [this] (std::vector<std::string> data) {updateEggLaidByPlayer(data);}},
00073         {"ebo", [this] (std::vector<std::string> data) {updatePlayerBorn(data);}},
00074         {"edi", [this] (std::vector<std::string> data) {updateEggDeath(data);}},
00075         {"sgt", [this] (std::vector<std::string> data) {updateTimeUnitRequest(data);}},
00076         {"sst", [this] (std::vector<std::string> data) {updateTimeUnitModification(data);}},
00077         {"seg", [this] (std::vector<std::string> data) {updateEndOfGame(data);}},
00078         {"smg", [this] (std::vector<std::string> data) {updateMessageFromServer(data);}},
00079         {"suc", [this] (std::vector<std::string> data) {updateUnknownMessage(data);}},
00080         {"sbp", [this] (std::vector<std::string> data) {updateCommandParameter(data);}},
00081     }; // The map of commands to update the GUI GameData.
00082
00083     void updateMapSize(const std::vector<std::string> &data);
00084
00085     void updateMapContent(const std::vector<std::string> &data);
00086
00087     void updateTeamNames(const std::vector<std::string> &data);
00088
00089     void updateTeamMember(const std::vector<std::string> &data);
00090
00091     void updatePlayerPosition(const std::vector<std::string> &data);
00092
00093     void updatePlayerLevel(const std::vector<std::string> &data);
00094
00095     void updatePlayerInventory(const std::vector<std::string> &data);
00096
00097     void updatePlayerExpulsion(const std::vector<std::string> &data);
00098
00099     void updatePlayerBroadcast(const std::vector<std::string> &data);
00100
00101     void updatePlayerStartIncantation(const std::vector<std::string> &data);
00102
00103     void updatePlayerEndIncantation(const std::vector<std::string> &data);
00104
00105     void updatePlayerEggLaying(const std::vector<std::string> &data);
00106
00107     void updatePlayerRessourceDropping(const std::vector<std::string> &data);
00108
00109     void updatePlayerRessourceCollecting(const std::vector<std::string> &data);
00110
00111     void updatePlayerDeath(const std::vector<std::string> &data);
00112
00113     void updateEggLaidByPlayer(const std::vector<std::string> &data);
00114
00115     void updatePlayerBorn(const std::vector<std::string> &data);
00116
00117     void updateEggDeath(const std::vector<std::string> &data);
00118
00119     void updateTimeUnitRequest(const std::vector<std::string> &data);
00120
00121     void updateTimeUnitModification(const std::vector<std::string> &data);
00122
00123     void updateEndOfGame(const std::vector<std::string> &data);
00124
00125     void updateMessageFromServer(const std::vector<std::string> &data);
00126
00127     void updateUnknownMessage(const std::vector<std::string> &data);
00128
00129     void updateCommandParameter(const std::vector<std::string> &data);
00130
00131     void increaseColorIndex();
00132 };

```

## 6.19 IGUIUpdater.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** IGUIUpdater
00006  */
00007
00008 #pragma once
00009
00010 #include <string>
00011 #include <vector>
00012
00013 namespace Gui {
00014
00015     class IGUIUpdater;
00016 }
00017

```

```

00022 class Gui::IGUIUpdater {
00023     public:
00027         virtual ~IGUIUpdater() = default;
00028
00033         virtual void update(const std::string &command, const std::vector<std::string> &data) = 0;
00034 };

```

## 6.20 AHud.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** AHud
00006 */
00007
00008 #pragma once
00009
00010 #include "Hud/IHud.hpp"
00011 #include "GameDatas/GameData.hpp"
00012
00013 namespace Gui {
00014
00019     class AHud;
00020 };
00021
00022 class Gui::AHud : public Gui::IHud {
00023     public:
00024
00025         ~AHud() = default;
00030
00036         virtual void display() = 0;
00037
00043         void setPlayer(std::shared_ptr<Player> player);
00044
00050         void setTile(std::shared_ptr<Tile> tile);
00051
00057         TypeScene getType() const;
00058
00064         void setType(TypeScene type);
00065
00066     protected:
00067
00068         TypeScene _typeScene;
00069         std::shared_ptr<GameData> _gameData;
00070         std::shared_ptr<Player> _player;
00071         std::shared_ptr<Tile> _tile;
00072 };

```

## 6.21 HudEnd.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** HudEnd
00006 */
00007
00008 #pragma once
00009
00010 #include "Hud/AHud.hpp"
00011
00012 namespace Gui {
00013
00018     class HudEnd;
00019 };
00020
00021 class Gui::HudEnd : public Gui::AHud {
00022     public:
00023
00024         HudEnd(std::shared_ptr<GameData> gameData);
00030
00036         ~HudEnd() = default;
00037
00042         void display();
00043
00044     private:

```

```

00045
00046         Texture2D    _texture;
00047         Font          _font;
00048     };

```

## 6.22 HudGame.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** HudGame
00006  */
00007
00008  #pragma once
00009
00010  #include "Hud/AHud.hpp"
00011
00012  #define HUD_GAME_POS Vector2{0, 340}
00013  #define HUD_GAME_TEXT_POS Vector2{25, 420}
00014  #define HUD_GAME_TEXT_TITLE_POS Vector2{50, 380}
00015  #define HUD_GAME_TEXT_MARGING 30
00016
00017  namespace Gui {
00018
00023      class HudGame;
00024  };
00025
00026  class Gui::HudGame : public Gui::AHud {
00027
00028      public:
00029
00035          HudGame(std::shared_ptr<GameData> gameData);
00036
00041          ~HudGame() = default;
00042
00047          void display();
00048
00049      private:
00050
00051          Texture2D    _texture;
00052          Font          _font;
00053          Texture2D    _playerTexture;
00054  };

```

## 6.23 HudHelp.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** HudHelp
00006  */
00007
00008  #pragma once
00009
00010  #include "Hud/AHud.hpp"
00011
00012  namespace Gui {
00013
00018      class HudHelp;
00019  };
00020
00021  class Gui::HudHelp : public Gui::AHud {
00022
00023      public:
00024
00030          HudHelp(std::shared_ptr<GameData> gameData);
00031
00036          ~HudHelp() = default;
00037
00042          void display();
00043
00044      private:
00045
00046          Texture2D    _texture;           // Texture for Hud Background.
00047          Texture2D    _textureKeys;      // Texture for Hud keys.
00048          Font          _font;             // Font for Hud's texts.
00049          clock_t       _helpMenuClock;    // Clock to display the help menu.

```



```

00050
00056         void displayHelpMenuControls();
00057     };

```

## 6.24 HudPlayer.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** HudPlayer
00006  */
00007
00008  #pragma once
00009
00010  #include "Hud/AHud.hpp"
00011
00012  #define HUD_PLAYER_POS Vector2{0, 340}
00013  #define HUD_PLAYER_TEXT_POS Vector2{55, 420}
00014  #define HUD_PLAYER_TEXT_TITLE_POS Vector2{80, 380}
00015  #define HUD_PLAYER_TEXT_MARGING 30
00016  #define HUD_PLAYER_ICONS_MARGING -32
00017  #define HUD_PLAYER_TITLE_ICON_MARGING Vector2{45, 35}
00018
00019  namespace Gui {
00020
00025      class HudPlayer;
00026  };
00027
00028  class Gui::HudPlayer : public Gui::AHud {
00029
00030      public:
00031
00037          HudPlayer(std::shared_ptr<GameData> gameData);
00038
00043          ~HudPlayer() = default;
00044
00049          void display();
00050
00051      private:
00052
00053          Texture2D _texture;
00054          Font _font;
00055
00056          Texture2D _food;
00057          Texture2D _linemate;
00058          Texture2D _deraumere;
00059          Texture2D _mendienae;
00060          Texture2D _phiras;
00061          Texture2D _sibur;
00062          Texture2D _thystame;
00063          Texture2D _playerTexture;
00064  };

```

## 6.25 HudTile.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** HudTile
00006  */
00007
00008  #pragma once
00009
00010  #include "Hud/AHud.hpp"
00011
00012  #define HUD_TILE_POS Vector2{0, 340}
00013  #define HUD_TILE_TEXT_POS Vector2{55, 420}
00014  #define HUD_TILE_TEXT_TITLE_POS Vector2{60, 380}
00015  #define HUD_TILE_TEXT_MARGING 30
00016  #define HUD_TILE_ICONS_MARGING -32
00017  #define HUD_TILE_TITLE_ICON_MARGING Vector2{45, 40}
00018
00019  namespace Gui {
00020
00025      class HudTile;
00026  };
00027

```

```

00028 class Gui::HudTile : public Gui::AHud {
00029
00030     public:
00031
00037         HudTile(std::shared_ptr<GameData> gameData);
00038
00043         ~HudTile() = default;
00044
00049         void display();
00050
00055         void displayNbPlayers();
00056
00061         void displayNbEggs();
00062
00063     private:
00064
00065         Texture2D _texture;
00066         Font _font;
00067
00068         Texture2D _food;
00069         Texture2D _linemate;
00070         Texture2D _deraumere;
00071         Texture2D _mendiane;
00072         Texture2D _phiras;
00073         Texture2D _sibur;
00074         Texture2D _thystame;
00075         Texture2D _tileTexture;
00076         Texture2D _playerTexture;
00077         Texture2D _eggTexture;
00078 };

```

## 6.26 IHud.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** IHud
00006 */
00007
00008 #pragma once
00009
00010 #include "GameDatas/Player.hpp"
00011 #include "GameDatas/Tile.hpp"
00012
00013 #include <memory>
00014
00015 namespace Gui {
00016
00021     class IHud;
00022 };
00023
00024 class Gui::IHud {
00025
00026     public:
00027
00032         enum TypeScene {
00033             GAME,
00034             POV_PLAYER,
00035             END_GAME,
00036             TILE,
00037             HELP_TEXT,
00038             HELP_MENU,
00039             END
00040         };
00041
00046         virtual ~IHud() = default;
00047
00052         virtual void display() = 0;
00053
00059         virtual void setPlayer(std::shared_ptr<Player> player) = 0;
00060
00066         virtual void setTile(std::shared_ptr<Tile> tile) = 0;
00067
00073         virtual TypeScene getType() const = 0;
00074
00080         virtual void setType(TypeScene type) = 0;
00081 };

```

## 6.27 ANetwork.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** ANetwork
00006  */
00007
00008  #pragma once
00009
00010  #include "Network/INetwork.hpp"
00011
00012  #define MIN_PORT 1
00013  #define MAX_PORT 65535
00014
00015  namespace Gui {
00016
00021      class ANetwork;
00022  };
00023
00024  class Gui::ANetwork : public Gui::INetwork {
00025  public:
00026
00033      ANetwork(int port, const std::string &hostName);
00034
00039      ~ANetwork() = default;
00040
00047      void setPort(int port) final;
00048
00054      void setHostName(const std::string &hostName) final;
00055
00061      int getPort() const final;
00062
00068      std::string getHostName() const final;
00069
00075      virtual void connectToServer() = 0;
00076
00082      virtual BufferState listenServer() = 0;
00083
00089      virtual void sendMessageServer(const std::string& message) = 0;
00090
00098      std::string getBuffer();
00099
00100  protected:
00101      int _port;
00102      std::string _hostName;
00103      std::string _buffer;
00104  };

```

## 6.28 INetwork.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** INetwork
00006  */
00007
00008  #pragma once
00009
00010  #include "Error/Error.hpp"
00011
00012  #include <string>
00013
00014  namespace Gui {
00015
00020      class INetwork;
00021  };
00022
00023  class Gui::INetwork {
00024  public:
00026
00027      enum BufferState {
00028          NONE,
00029          READY,
00030          SERVER_ERROR
00031      };
00032
00037      virtual ~INetwork() = default;
00038
00045      virtual void setPort(int port) = 0;

```

```

00046
00052     virtual void setHostName(const std::string &hostName) = 0;
00053
00059     virtual int getPort() const = 0;
00060
00066     virtual std::string getHostName() const = 0;
00067
00073     virtual void connectToServer() = 0;
00074
00080     virtual BufferState listenServer() = 0;
00081
00087     virtual void sendMessageServer(const std::string &message) = 0;
00088
00096     virtual std::string getBuffer() = 0;
00097 };

```

## 6.29 Network.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Network
00006 */
00007
00008 #pragma once
00009
00010 #include "Network/ANetwork.hpp"
00011
00012 #include <arpa/inet.h>
00013 #include <sys/socket.h>
00014 #include <netinet/in.h>
00015
00016 namespace Gui {
00017
00022     class Network;
00023 };
00024
00025 class Gui::Network : public Gui::ANetwork {
00026
00027     public:
00028
00035         Network(int port, const std::string &hostName);
00036
00041         ~Network();
00042
00048         void connectToServer();
00049
00055         BufferState listenServer();
00056
00062         void sendMessageServer(const std::string& message);
00063
00064     private:
00065
00070         void selectServer();
00071
00077         BufferState readInfoServer();
00078
00079         int         _serverFd;
00080         fd_set       _writeFd;
00081         fd_set       _readFd;
00082         bool         _isConnected;
00083 };

```

## 6.30 IServerParser.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** IServerParser
00006 */
00007
00008 #pragma once
00009
00010 #include <string>
00011 #include <vector>
00012
00013 namespace Gui {

```

```

00014
00018     class IServerParser;
00019 }
00020
00021 class Gui::IServerParser {
00022
00023     public:
00024
00028         virtual ~IServerParser() = default;
00029
00036         virtual std::vector<std::string> parse(const std::string& command) = 0;
00037 };

```

## 6.31 ParseCommandLine.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** ParseCommandLine
00006 */
00007
00008 #pragma once
00009
00010 #include <string>
00011
00012 #define GUI_USAGE "USAGE:\t./zappy_gui -p port -h machine"
00013
00014 namespace Gui {
00015
00020     class ParseCommandLine;
00021 };
00022
00023 class Gui::ParseCommandLine {
00024
00025     public:
00026
00033         ParseCommandLine(int argc, char **argv);
00034
00039         ~ParseCommandLine() = default;
00040
00047         void parseFlags(int argc, char **argv);
00048
00054         int getPort(void);
00055
00061         std::string getHostName(void);
00062
00063     private:
00064
00065         int _port;
00066         std::string _hostname;
00067 };

```

## 6.32 ServerParser.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Parse server command
00006 */
00007
00008 #pragma once
00009
00010 #include "Error/Error.hpp"
00011 #include "Parsing/IServerParser.hpp"
00012
00013 #include <functional>
00014 #include <unordered_map>
00015
00016 namespace Gui {
00017
00022     class ServerParser;
00023 };
00024
00025 class Gui::ServerParser : public Gui::IServerParser {
00026
00027     public:
00028

```

```

00033         ServerParser() = default;
00034
00039         ~ServerParser() = default;
00040
00047         std::vector<std::string> parse(const std::string& command);
00048
00049     private:
00050
00055         enum ParseType {
00056             INT,
00057             STRING,
00058             MESSAGE,
00059             LIST_INT
00060         };
00061
00066         std::unordered_map<std::string, std::vector<ParseType> _typesCommand =
00067         {
00068             {"msz", std::vector<ParseType>{INT, INT}},
00069             {"bct", std::vector<ParseType>{INT, INT, INT, INT, INT, INT, INT, INT, INT, INT}},
00070             {"tna", std::vector<ParseType>{STRING}},
00071             {"pnw", std::vector<ParseType>{INT, INT, INT, INT, INT, STRING}},
00072             {"ppo", std::vector<ParseType>{INT, INT, INT, INT}},
00073             {"plv", std::vector<ParseType>{INT, INT}},
00074             {"pin", std::vector<ParseType>{INT, INT, INT, INT, INT, INT, INT, INT, INT, INT, INT}},
00075             {"pex", std::vector<ParseType>{INT}},
00076             {"pbc", std::vector<ParseType>{INT, MESSAGE}},
00077             {"pic", std::vector<ParseType>{INT, INT, INT, LIST_INT}},
00078             {"pie", std::vector<ParseType>{INT, INT, INT}},
00079             {"pfk", std::vector<ParseType>{INT}},
00080             {"pdr", std::vector<ParseType>{INT, INT}},
00081             {"pgt", std::vector<ParseType>{INT, INT}},
00082             {"pdi", std::vector<ParseType>{INT}},
00083             {"enw", std::vector<ParseType>{INT, INT, INT, INT}},
00084             {"ebo", std::vector<ParseType>{INT}},
00085             {"edi", std::vector<ParseType>{INT}},
00086             {"sgt", std::vector<ParseType>{INT}},
00087             {"sst", std::vector<ParseType>{INT}},
00088             {"seg", std::vector<ParseType>{STRING}},
00089             {"smg", std::vector<ParseType>{MESSAGE}},
00090             {"suc", std::vector<ParseType>{}},
00091             {"sbp", std::vector<ParseType>{}}
00092         };
00093
00101         std::vector<std::string> parseCommand(const std::string& command, std::vector<ParseType>
types);
00102
00110         std::vector<std::string> parseInt(std::istringstream& stream, std::vector<std::string>
arguments);
00111
00119         std::vector<std::string> parseString(std::istringstream& stream, std::vector<std::string>
arguments);
00120
00129         std::vector<std::string> parseMessage(std::istringstream& stream, std::vector<std::string>
arguments, std::string commandName);
00130
00139         std::vector<std::string> parseListInt(std::istringstream& stream, std::vector<std::string>
arguments, std::string commandName);
00140     };

```

## 6.33 Decoration.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** Decoration
00006  */
00007
00008  #pragma once
00009
00010  #include "raylib.h"
00011  #include "Types.hpp"
00012  #include "Assets.hpp"
00013
00014  #include <vector>
00015  #include <iostream>
00016
00017  namespace Gui {
00018
00024      class Decoration;
00025  }
00026
00027  class Gui::Decoration {

```

```

00028
00029     public:
00030
00035         Decoration();
00036
00041         ~Decoration() = default;
00042
00051         void display(std::pair<std::size_t, std::size_t> mapSize, size_t renderDistance,
std::pair<std::size_t, std::size_t> camPos, std::vector<Vector2> displayPos);
00052
00061         Map<bool> getGenerationItem(std::size_t ratio);
00062
00063     private:
00064
00065         Model _treeModel;
00066         Map<bool> _mapTree;
00067
00068         std::pair<std::size_t, std::size_t> _mapSize;
00069
00077         void displayTree(size_t i, size_t j, Vector3 posTile);
00078
00087         bool isInArrayPlayerVision(std::pair<size_t, size_t> pos, std::vector<Vector2>
_playerVisionPositions);
00088     };

```

## 6.34 Render.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Render
00006 */
00007
00008 #pragma once
00009
00010
00011 #include "raylib.h"
00012 #include "Config.hpp"
00013 #include "Hud/HudGame.hpp"
00014 #include "Hud/HudTile.hpp"
00015 #include "Hud/HudPlayer.hpp"
00016 #include "Hud/HudHelp.hpp"
00017 #include "Hud/HudEnd.hpp"
00018 #include "Render/Decoration.hpp"
00019 #include "Render/UserCamera.hpp"
00020 #include "GameDatas/GameData.hpp"
00021
00022 #include <functional>
00023 #include <unordered_map>
00024
00025 namespace Gui {
00026
00031     class Render;
00032 };
00033
00034 class Gui::Render {
00035
00036     public:
00037
00042         Render(std::shared_ptr<GameData> gameData);
00043
00048         ~Render();
00049
00056         bool isOpen();
00057
00062         void draw();
00063
00069         std::shared_ptr<Camera> getCamera();
00070
00076         void setIsDebug(bool isDebug);
00077
00084         bool getIsDebug(void);
00085
00091         void setCameraType(Gui::UserCamera::CameraType type);
00092
00098         Gui::UserCamera::CameraType getCameraType() const;
00099
00105         void setCameraPlayerPov(std::size_t id);
00106
00112         std::size_t getCameraPlayerPov() const;
00113
00119         void setCameraTile(std::pair<std::size_t, std::size_t> pos);

```

```

00120
00126         std::pair<std::size_t, std::size_t> getCameraTile() const;
00127
00132         Model getTileModel() const;
00133
00139         void setRenderDistance(size_t renderDistance);
00140
00145         size_t getRenderDistance() const;
00146
00154         bool isCameraInPlayerPov() const;
00155
00162         void changePlayerPOV(size_t playerId);
00163
00169         void setPlayerPov(size_t playerId);
00170
00176         void changePOVToFirstPerson(size_t id);
00177
00183         void changePOVToSecondPerson(size_t id);
00184
00190         void changePOVToThirdPerson(size_t id);
00191
00197         size_t getTimeUnit() const;
00198
00204         void setTimeUnit(size_t timeUnit);
00205
00213         void setPlayerVision(bool isPlayerVision);
00214
00221         bool getPlayerVision() const;
00222
00228         void setHelpMenu(bool isHelpMenu);
00229
00236         bool getHelpMenu() const;
00237
00242         void drawEnd() const;
00243
00244     private:
00245
00246         UserCamera                                _camera;
00247         bool                                       _isDebug;
00248         std::shared_ptr<GameData>                 _gameData;
00249         std::shared_ptr<Decoration>               _decoration;
00250         std::vector<std::shared_ptr<Gui::IHud>>    _hudList;
00251         size_t                                    _renderDistance;
00252         bool                                       _isHelpMenu;
00253
00254         Model                                     _tileModel;
00255         Model                                     _foodModel;
00256         Model                                     _linemateModel;
00257         Model                                     _mendianeModel;
00258         Model                                     _phirasModel;
00259         Model                                     _siburModel;
00260         Model                                     _thystameModel;
00261         Model                                     _deraumereModel;
00262         Texture2D                                 _cursorTexture;
00263         std::vector<Vector2>                     _playerVisionPositions;
00264         bool                                       _endHudSet;
00265
00270         void LoadModels();
00271
00276         void displayHUD();
00277
00282         void displayDebug();
00283
00288         void displayPlayers();
00289
00297         void displayPlayerLevel(Player &player, Vector3 position, Team &team);
00298
00305         void displayPlayerBroadcast(Team &team, Player &player);
00306
00311         void displayMap();
00312
00317         void displayTile(Tile tile);
00318
00324         void displayEggs(Tile tile) const;
00325
00331         void displayFood(Tile tile) const;
00332
00338         void displayResources(Tile tile) const;
00339
00345         void displayLinemate(Tile tile) const;
00346
00352         void displayMendiane(Tile tile) const;
00353
00359         void displayPhiras(Tile tile) const;
00360
00366         void displaySibur(Tile tile) const;
00367

```



```

00373     void displayThystame(Tile tile) const;
00374
00380     void displayDeraumere(Tile tile) const;
00381
00388     bool displayAnimations(Team &team, Player &player);
00389
00398     ModelAnimation displayWalkAnimation(Team &team, Player &player, ModelAnimation anim);
00399
00404     void displayCursor();
00405
00411     std::pair<std::size_t, std::size_t> getCameraTile();
00412
00418     std::vector<Vector2> getPositionsInPlayerVision(size_t playerId);
00419
00427     bool isInArrayPlayerVision(std::pair<size_t, size_t> pos);
00428
00436     std::vector<Vector2> getLineOfVision(Vector2 pos, size_t sizeOfHalf, size_t orientation);
00437
00445     std::vector<Vector2> addVisionPosition(std::vector<Vector2> vision, std::vector<Vector2> pos);
00446
00451     void displayHelpMenu(std::shared_ptr<IHud> hud);
00452
00458     void displayHelpMenuControls(Vector2 position);
00459 };

```

## 6.35 UserCamera.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** Camera
00006  */
00007
00008 #pragma once
00009
00010 #include "raylib.h"
00011
00012 #include <memory>
00013
00014 namespace Gui {
00015
00020     class UserCamera;
00021 };
00022
00023 class Gui::UserCamera {
00024
00025     public:
00026
00027         enum CameraType {
00028             FREE,
00029             FIRST_PERSON,
00030             SECOND_PERSON,
00031             THIRD_PERSON,
00032             FREE_TILE
00033         };
00034
00039         UserCamera();
00040
00045         ~UserCamera() = default;
00046
00052         void setPosition(Vector3 position);
00053
00059         void setTarget(Vector3 target);
00060
00066         void setUp(Vector3 up);
00067
00073         void setFovy(float fovy);
00074
00080         Vector3 getPosition(void) const;
00081
00087         Vector3 getTarget(void) const;
00088
00094         Vector3 getUp(void) const;
00095
00101         float getFovy(void) const;
00102
00108         std::shared_ptr<Camera> getCamera();
00109
00115         void setType(CameraType type);
00116
00122         CameraType getType() const;
00123

```

```
00129         void setPlayerId(size_t playerId);
00130
00136         int getPlayerId() const;
00137
00143         void setTilePos(std::pair<std::size_t, std::size_t> pos);
00144
00150         std::pair<std::size_t, std::size_t> getTilePos() const;
00151
00159         bool isPlayerPov() const;
00160
00167         bool isPlayerVision() const;
00168
00174         void setPlayerVision(bool isPlayerVision);
00175
00176     private:
00177
00178         std::shared_ptr<Camera> _camera;
00179         CameraType _type;
00180         int _playerId;
00181         std::pair<std::size_t, std::size_t> _tilePos;
00182         bool _isPlayerVision;
00183     };
```

## 6.36 Types.hpp

```
00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** Types
00006  */
00007
00008  #pragma once
00009
00010  #include <vector>
00011
00017  template<typename T>
00018  using Map = std::vector<std::vector<T>>;
```

# Index

/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Assets.hpp, 139  
121 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/IHud.hpp,  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Colors.hpp, 140  
123 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/ANetwork  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Config.hpp, 141  
123 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/INetwork.I  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Engine/Engine.hpp,  
124 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/Network.h  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/AError.hpp,  
124 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/IServerPar  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp, 142  
125 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ParseCom  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/IError.hpp, 143  
126 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ServerPar  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/AEvent.hpp,  
126 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Decoration  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/Event.hpp, 144  
126 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Render.hpp  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/IEvent.hpp,  
128 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/UserCame  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/AGUIUpdater.hpp,  
134 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Types.hpp,  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/GUIUpdater.hpp,  
135 ~AHud  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/GUIUpdater.hpp,  
136 ~ANetwork  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EGG.hpp, 16  
128 ~Decoration  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EGGData.hpp,  
129 ~Egg  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EGGData.hpp,  
131 ~Engine  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EGGData.hpp,  
132 ~Event  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EGGData.hpp,  
133 ~GameData  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EGGData.hpp, 30  
134 ~HudEnd  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/AHud.hpp, 46  
137 ~HudGame  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp, 49  
137 ~HudHelp  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp, 51  
138 ~HudPlayer  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp, 53  
138 ~HudTile  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp, 55  
139 ~IEvent  
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp, 57

- ~IHud
  - Gui::IHud, 60
- ~INetwork
  - Gui::INetwork, 62
- ~Inventory
  - Gui::Inventory, 67
- ~Network
  - Gui::Network, 75
- ~ParseCommandLine
  - Gui::ParseCommandLine, 78
- ~Player
  - Gui::Player, 82
- ~Render
  - Gui::Render, 90
- ~ServerParser
  - Gui::ServerParser, 99
- ~Team
  - Gui::Team, 102
- ~Tile
  - Gui::Tile, 110
- ~UserCamera
  - Gui::UserCamera, 113
- addEgg
  - Gui::Team, 102
- addPlayer
  - Gui::Team, 103
- addPlayerToTeam
  - Gui::GameData, 30
- addResource
  - Gui::Inventory, 67
- addServerEgg
  - Gui::GameData, 31
- addTeam
  - Gui::GameData, 31
- AGUIUpdater
  - Gui::AGUIUpdater, 11
- ANetwork
  - Gui::ANetwork, 16
- changePlayerPOV
  - Gui::Render, 90
- changePOVToFirstPerson
  - Gui::Render, 90
- changePOVToSecondPerson
  - Gui::Render, 91
- changePOVToThirdPerson
  - Gui::Render, 91
- connectToServer
  - Gui::ANetwork, 16
  - Gui::INetwork, 63
  - Gui::Network, 75
- Decoration
  - Gui::Decoration, 19
- display
  - Gui::AHud, 13
  - Gui::Decoration, 20
  - Gui::HudEnd, 46
  - Gui::HudGame, 49
  - Gui::HudHelp, 51
  - Gui::HudPlayer, 53
  - Gui::HudTile, 55
  - Gui::IHud, 60
- displayNbEggs
  - Gui::HudTile, 56
- displayNbPlayers
  - Gui::HudTile, 56
- draw
  - Gui::Render, 91
- drawEnd
  - Gui::Render, 91
- Egg
  - Gui::Egg, 21
- Engine
  - Gui::Engine, 24
- Event
  - Gui::Event, 27
- GameData
  - Gui::GameData, 30
- getAnimationTimeElapsed
  - Gui::Player, 82
- getBroadcast
  - Gui::Player, 82
- getBuffer
  - Gui::ANetwork, 17
  - Gui::INetwork, 63
- getCamera
  - Gui::Render, 91
  - Gui::UserCamera, 114
- getCameraPlayerPov
  - Gui::Render, 92
- getCameraTile
  - Gui::Render, 92
- getCameraType
  - Gui::Render, 92
- getCenterPosition
  - Gui::Player, 82
- getCurrentFrame
  - Gui::Player, 83
- getDeraumere
  - Gui::Inventory, 68
- getEgg
  - Gui::Team, 103
- getEggModel
  - Gui::Team, 103
- getEggs
  - Gui::Team, 104
- getEndMessage
  - Gui::GameData, 31
- getFood
  - Gui::Inventory, 68
- getFovy
  - Gui::UserCamera, 114
- getGenerationItem
  - Gui::Decoration, 20

getHelpMenu  
    Gui::Render, 92  
getHostName  
    Gui::ANetwork, 17  
    Gui::INetwork, 63  
    Gui::ParseCommandLine, 78  
getId  
    Gui::Egg, 22  
    Gui::Player, 83  
getIsDebug  
    Gui::Render, 93  
getIsEndGame  
    Gui::GameData, 32  
getLastError  
    Gui::GameData, 32  
getLastTick  
    Gui::GameData, 32  
getLastTickMctCommand  
    Gui::GameData, 32  
getLevel  
    Gui::Player, 83  
getLinemate  
    Gui::Inventory, 68  
getMap  
    Gui::GameData, 32  
getMapSize  
    Gui::GameData, 33  
getMendiane  
    Gui::Inventory, 68  
getName  
    Gui::Team, 104  
getNbBCTCommandReceived  
    Gui::GameData, 33  
getOrientation  
    Gui::Player, 83  
getPhiras  
    Gui::Inventory, 68  
getPlayer  
    Gui::GameData, 33  
    Gui::Team, 104  
getPlayerBoundingBoxes  
    Gui::Team, 104  
getPlayerColor  
    Gui::Team, 105  
getPlayerId  
    Gui::UserCamera, 114  
getPlayerModel  
    Gui::Team, 105  
getPlayerModelAnimation  
    Gui::Team, 105  
getPlayerModelHitbox  
    Gui::Team, 105  
getPlayerPositionIn3DSpace  
    Gui::Team, 106  
getPlayers  
    Gui::Team, 106  
getPlayerVision  
    Gui::Render, 93  
getPort  
    Gui::ANetwork, 17  
    Gui::INetwork, 63  
    Gui::ParseCommandLine, 78  
getPosition  
    Gui::Egg, 22  
    Gui::Player, 84  
    Gui::Tile, 110  
    Gui::UserCamera, 114  
getPosition3D  
    Gui::Player, 84  
getPositionIn3DSpace  
    Gui::Tile, 110  
getRenderDistance  
    Gui::Render, 93  
getRessources  
    Gui::Inventory, 69  
getRotationFromOrientation  
    Gui::Player, 84  
getServerEggs  
    Gui::GameData, 33  
getServerError  
    Gui::GameData, 34  
getServerTick  
    Gui::GameData, 34  
getSibur  
    Gui::Inventory, 69  
getState  
    Gui::Egg, 22  
    Gui::Player, 84  
getTarget  
    Gui::UserCamera, 114  
getTeam  
    Gui::Egg, 23  
    Gui::GameData, 34  
    Gui::Player, 85  
getTeamById  
    Gui::GameData, 35  
getTeams  
    Gui::GameData, 35  
getThystame  
    Gui::Inventory, 69  
getTile  
    Gui::GameData, 35  
getTileBoundingBoxes  
    Gui::Tile, 110  
getTileModel  
    Gui::Render, 93  
getTileModelHitbox  
    Gui::Tile, 111  
getTilePos  
    Gui::UserCamera, 115  
getTimeUnit  
    Gui::Render, 94  
getTimeUnitFromServer  
    Gui::GameData, 36  
getType  
    Gui::AHud, 13

- Gui::IHud, 60
- Gui::UserCamera, 115
- getUp
  - Gui::UserCamera, 115
- Gui::AEvent, 10
- Gui::AGUIUpdater, 10
  - AGUIUpdater, 11
  - update, 11
- Gui::AHud, 12
  - ~AHud, 13
  - display, 13
  - getType, 13
  - setPlayer, 13
  - setTile, 14
  - setType, 14
- Gui::ANetwork, 14
  - ~ANetwork, 16
  - ANetwork, 16
  - connectToServer, 16
  - getBuffer, 17
  - getHostName, 17
  - getPort, 17
  - listenServer, 17
  - sendMessageServer, 18
  - setHostName, 18
  - setPort, 18
- Gui::Decoration, 19
  - ~Decoration, 19
  - Decoration, 19
  - display, 20
  - getGenerationItem, 20
- Gui::Egg, 21
  - ~Egg, 22
  - Egg, 21
  - getId, 22
  - getPosition, 22
  - getState, 22
  - getTeam, 23
  - setId, 23
  - setPosition, 23
  - setState, 24
  - setTeam, 24
- Gui::Engine, 24
  - ~Engine, 25
  - Engine, 24
  - run, 25
- Gui::Errors::AError, 9
  - what, 10
- Gui::Errors::Error, 25
- Gui::Errors::GuiGameDataException, 41
  - GuiGameDataException, 41
- Gui::Errors::GuiUpdaterException, 43
  - GuiUpdaterException, 44
- Gui::Errors::IError, 56
  - what, 56
- Gui::Errors::NetworkException, 76
  - NetworkException, 77
- Gui::Errors::ParseCommandLineException, 79
  - ParseCommandLineException, 80
- Gui::Errors::ServerParserException, 100
  - ServerParserException, 100
- Gui::Event, 26
  - ~Event, 27
  - Event, 27
  - listen, 28
- Gui::GameData, 28
  - ~GameData, 30
  - addPlayerToTeam, 30
  - addServerEgg, 31
  - addTeam, 31
  - GameData, 30
  - getEndMessage, 31
  - getIsEndGame, 32
  - getLastError, 32
  - getLastTick, 32
  - getLastTickMctCommand, 32
  - getMap, 32
  - getMapSize, 33
  - getNbBCTCommandReceived, 33
  - getPlayer, 33
  - getServerEggs, 33
  - getServerError, 34
  - getServerTick, 34
  - getTeam, 34
  - getTeamById, 35
  - getTeams, 35
  - getTile, 35
  - getTimeUnitFromServer, 36
  - removeServerEgg, 36
  - restartLastTick, 36
  - restartLastTickMctCommand, 36
  - setAnimsCount, 36
  - setEggModel, 37
  - setEndMessage, 37
  - setIsEndGame, 37
  - setLastError, 38
  - setMap, 38
  - setMapSize, 38
  - setNbBCTCommandReceived, 39
  - setPlayerModel, 39
  - setPlayerModelAnimation, 39
  - setServerError, 39
  - setServerTick, 40
  - setTile, 40
  - setTimeUnitFromServer, 40
- Gui::GUIUpdater, 42
  - GUIUpdater, 43
  - update, 43
- Gui::HudEnd, 45
  - ~HudEnd, 46
  - display, 46
  - HudEnd, 46
- Gui::HudGame, 47
  - ~HudGame, 49
  - display, 49
  - HudGame, 48

- Gui::HudHelp, 49
  - ~HudHelp, 51
  - display, 51
  - HudHelp, 51
- Gui::HudPlayer, 51
  - ~HudPlayer, 53
  - display, 53
  - HudPlayer, 53
- Gui::HudTile, 54
  - ~HudTile, 55
  - display, 55
  - displayNbEggs, 56
  - displayNbPlayers, 56
  - HudTile, 55
- Gui::IEvent, 57
  - ~IEvent, 57
  - IEvent, 57
  - listen, 58
  - setGameData, 58
  - setRender, 58
- Gui::IGUIUpdater, 59
  - update, 59
- Gui::IHud, 59
  - ~IHud, 60
  - display, 60
  - getType, 60
  - setPlayer, 61
  - setTile, 61
  - setType, 61
  - TypeScene, 60
- Gui::INetwork, 62
  - ~INetwork, 62
  - connectToServer, 63
  - getBuffer, 63
  - getHostName, 63
  - getPort, 63
  - listenServer, 64
  - sendMessageServer, 64
  - setHostName, 64
  - setPort, 65
- Gui::Inventory, 65
  - ~Inventory, 67
  - addResource, 67
  - getDeraumere, 68
  - getFood, 68
  - getLinemate, 68
  - getMendiane, 68
  - getPhiras, 68
  - getRessources, 69
  - getSibur, 69
  - getThystame, 69
  - Inventory, 66
  - removeResource, 69
  - setDeraumere, 70
  - setFood, 70
  - setLinemate, 70
  - setMendiane, 70
  - setPhiras, 71
  - setRessources, 71
  - setSibur, 71
  - setThystame, 72
- Gui::IServerParser, 72
  - parse, 72
- Gui::Network, 73
  - ~Network, 75
  - connectToServer, 75
  - listenServer, 75
  - Network, 74
  - sendMessageServer, 75
- Gui::ParseCommandLine, 77
  - ~ParseCommandLine, 78
  - getHostName, 78
  - getPort, 78
  - ParseCommandLine, 77
  - parseFlags, 78
- Gui::Player, 80
  - ~Player, 82
  - getAnimationTimeEllapsed, 82
  - getBroadcast, 82
  - getCenterPosition, 82
  - getCurrentFrame, 83
  - getId, 83
  - getLevel, 83
  - getOrientation, 83
  - getPosition, 84
  - getPosition3D, 84
  - getRotationFromOrientation, 84
  - getState, 84
  - getTeam, 85
  - inventory, 88
  - Player, 81
  - restartAnimationTimeEllapsed, 85
  - setBroadcast, 85
  - setCurrentFrame, 85
  - setId, 86
  - setLevel, 86
  - setOrientation, 86
  - setPosition, 87
  - setPosition3D, 87
  - setState, 87
  - setTeam, 87
- Gui::Render, 88
  - ~Render, 90
  - changePlayerPOV, 90
  - changePOVToFirstPerson, 90
  - changePOVToSecondPerson, 91
  - changePOVToThirdPerson, 91
  - draw, 91
  - drawEnd, 91
  - getCamera, 91
  - getCameraPlayerPov, 92
  - getCameraTile, 92
  - getCameraType, 92
  - getHelpMenu, 92
  - getIsDebug, 93
  - getPlayerVision, 93

- getRenderDistance, 93
- getTileModel, 93
- getTimeUnit, 94
- isCameraInPlayerPov, 94
- isOpen, 94
- Render, 89
- setCameraPlayerPov, 94
- setCameraTile, 95
- setCameraType, 95
- setHelpMenu, 95
- setIsDebug, 95
- setPlayerPov, 96
- setPlayerVision, 96
- setRenderDistance, 96
- setTimeUnit, 98
- Gui::ServerParser, 98
  - ~ServerParser, 99
  - parse, 99
  - ServerParser, 99
- Gui::Team, 101
  - ~Team, 102
  - addEgg, 102
  - addPlayer, 103
  - getEgg, 103
  - getEggModel, 103
  - getEggs, 104
  - getName, 104
  - getPlayer, 104
  - getPlayerBoundingBoxes, 104
  - getPlayerColor, 105
  - getPlayerModel, 105
  - getPlayerModelAnimation, 105
  - getPlayerModelHitbox, 105
  - getPlayerPositionIn3DSpace, 106
  - getPlayers, 106
  - isPlayerHit, 106
  - removeEgg, 107
  - removePlayer, 107
  - setEggModelPath, 107
  - setName, 108
  - setPlayerModelPath, 108
  - Team, 102
- Gui::Tile, 108
  - ~Tile, 110
  - getPosition, 110
  - getPositionIn3DSpace, 110
  - getTileBoundingBoxes, 110
  - getTileModelHitbox, 111
  - inventory, 112
  - isTileHit, 111
  - setPosition, 111
  - Tile, 109
- Gui::UserCamera, 112
  - ~UserCamera, 113
  - getCamera, 114
  - getFovy, 114
  - getPlayerId, 114
  - getPosition, 114
  - getTarget, 114
  - getTilePos, 115
  - getType, 115
  - getUp, 115
  - isPlayerPov, 115
  - isPlayerVision, 116
  - setFovy, 116
  - setPlayerId, 116
  - setPlayerVision, 118
  - setPosition, 118
  - setTarget, 118
  - setTilePos, 118
  - setType, 119
  - setUp, 119
  - UserCamera, 113
- GuiGameDataException
  - Gui::Errors::GuiGameDataException, 41
- GUIUpdater
  - Gui::GUIUpdater, 43
- GuiUpdaterException
  - Gui::Errors::GuiUpdaterException, 44
- HudEnd
  - Gui::HudEnd, 46
- HudGame
  - Gui::HudGame, 48
- HudHelp
  - Gui::HudHelp, 51
- HudPlayer
  - Gui::HudPlayer, 53
- HudTile
  - Gui::HudTile, 55
- IEvent
  - Gui::IEvent, 57
- Inventory
  - Gui::Inventory, 66
- inventory
  - Gui::Player, 88
  - Gui::Tile, 112
- isCameraInPlayerPov
  - Gui::Render, 94
- isOpen
  - Gui::Render, 94
- isPlayerHit
  - Gui::Team, 106
- isPlayerPov
  - Gui::UserCamera, 115
- isPlayerVision
  - Gui::UserCamera, 116
- isTileHit
  - Gui::Tile, 111
- listen
  - Gui::Event, 28
  - Gui::IEvent, 58
- listenServer
  - Gui::ANetwork, 17
  - Gui::INetwork, 64



- Gui::Network, 75
- Network
  - Gui::Network, 74
- NetworkException
  - Gui::Errors::NetworkException, 77
- parse
  - Gui::IServerParser, 72
  - Gui::ServerParser, 99
- ParseCommandLine
  - Gui::ParseCommandLine, 77
- ParseCommandLineException
  - Gui::Errors::ParseCommandLineException, 80
- parseFlags
  - Gui::ParseCommandLine, 78
- Player
  - Gui::Player, 81
- removeEgg
  - Gui::Team, 107
- removePlayer
  - Gui::Team, 107
- removeResource
  - Gui::Inventory, 69
- removeServerEgg
  - Gui::GameData, 36
- Render
  - Gui::Render, 89
- restartAnimationTimeEllapsed
  - Gui::Player, 85
- restartLastTick
  - Gui::GameData, 36
- restartLastTickMctCommand
  - Gui::GameData, 36
- run
  - Gui::Engine, 25
- sendMessageServer
  - Gui::ANetwork, 18
  - Gui::INetwork, 64
  - Gui::Network, 75
- ServerParser
  - Gui::ServerParser, 99
- ServerParserException
  - Gui::Errors::ServerParserException, 100
- setAnimsCount
  - Gui::GameData, 36
- setBroadcast
  - Gui::Player, 85
- setCameraPlayerPov
  - Gui::Render, 94
- setCameraTile
  - Gui::Render, 95
- setCameraType
  - Gui::Render, 95
- setCurrentFrame
  - Gui::Player, 85
- setDeraumere
  - Gui::Inventory, 70
- setEggModel
  - Gui::GameData, 37
- setEggModelPath
  - Gui::Team, 107
- setEndMessage
  - Gui::GameData, 37
- setFood
  - Gui::Inventory, 70
- setFovy
  - Gui::UserCamera, 116
- setGameData
  - Gui::IEvent, 58
- setHelpMenu
  - Gui::Render, 95
- setHostName
  - Gui::ANetwork, 18
  - Gui::INetwork, 64
- setId
  - Gui::Egg, 23
  - Gui::Player, 86
- setIsDebug
  - Gui::Render, 95
- setIsEndGame
  - Gui::GameData, 37
- setLastError
  - Gui::GameData, 38
- setLevel
  - Gui::Player, 86
- setLinemate
  - Gui::Inventory, 70
- setMap
  - Gui::GameData, 38
- setMapSize
  - Gui::GameData, 38
- setMendiane
  - Gui::Inventory, 70
- setName
  - Gui::Team, 108
- setNbBCTCommandReceived
  - Gui::GameData, 39
- setOrientation
  - Gui::Player, 86
- setPhiras
  - Gui::Inventory, 71
- setPlayer
  - Gui::AHud, 13
  - Gui::IHud, 61
- setPlayerId
  - Gui::UserCamera, 116
- setPlayerModel
  - Gui::GameData, 39
- setPlayerModelAnimation
  - Gui::GameData, 39
- setPlayerModelPath
  - Gui::Team, 108
- setPlayerPov
  - Gui::Render, 96

- setPlayerVision
  - Gui::Render, 96
  - Gui::UserCamera, 118
- setPort
  - Gui::ANetwork, 18
  - Gui::INetwork, 65
- setPosition
  - Gui::Egg, 23
  - Gui::Player, 87
  - Gui::Tile, 111
  - Gui::UserCamera, 118
- setPosition3D
  - Gui::Player, 87
- setRender
  - Gui::IEvent, 58
- setRenderDistance
  - Gui::Render, 96
- setRessources
  - Gui::Inventory, 71
- setServerError
  - Gui::GameData, 39
- setServerTick
  - Gui::GameData, 40
- setSibur
  - Gui::Inventory, 71
- setState
  - Gui::Egg, 24
  - Gui::Player, 87
- setTarget
  - Gui::UserCamera, 118
- setTeam
  - Gui::Egg, 24
  - Gui::Player, 87
- setThystame
  - Gui::Inventory, 72
- setTile
  - Gui::AHud, 14
  - Gui::GameData, 40
  - Gui::IHud, 61
- setTilePos
  - Gui::UserCamera, 118
- setTimeUnit
  - Gui::Render, 98
- setTimeUnitFromServer
  - Gui::GameData, 40
- setType
  - Gui::AHud, 14
  - Gui::IHud, 61
  - Gui::UserCamera, 119
- setUp
  - Gui::UserCamera, 119
- Team
  - Gui::Team, 102
- Tile
  - Gui::Tile, 109
- TypeScene
  - Gui::IHud, 60
- update
  - Gui::AGUIUpdater, 11
  - Gui::GUIUpdater, 43
  - Gui::IGUIUpdater, 59
- UserCamera
  - Gui::UserCamera, 113
- what
  - Gui::Errors::AError, 10
  - Gui::Errors::LError, 56