

Zappy GUI

Generated by Doxygen 1.9.6

1 Zappy GUI	1
1.1 Technology Notes	1
1.2 Installing Raylib on Fedora	1
1.2.1 Prerequisites	1
1.2.2 Installation Steps	1
1.2.2.1 1. Update Your System	1
1.2.2.2 2. Install Necessary Dependencies	1
1.2.2.3 3. Verify the Installation	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
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
5.5 Gui::ANetwork Class Reference	14
5.5.1 Constructor & Destructor Documentation	15
5.5.1.1 ANetwork()	15
5.5.1.2 ~ANetwork()	15
5.5.2 Member Function Documentation	16
5.5.2.1 connectToServer()	16
5.5.2.2 getHostName()	16

5.5.2.3	getPort()	16
5.5.2.4	listenServer()	17
5.5.2.5	sendMessageServer()	17
5.5.2.6	setHostName()	17
5.5.2.7	setPort()	18
5.6	Gui::Decoration Class Reference	18
5.6.1	Constructor & Destructor Documentation	18
5.6.1.1	Decoration()	18
5.6.1.2	~Decoration()	19
5.6.2	Member Function Documentation	19
5.6.2.1	display()	19
5.6.2.2	getGenerationItem()	19
5.7	Gui::Egg Class Reference	20
5.7.1	Constructor & Destructor Documentation	20
5.7.1.1	Egg()	20
5.7.1.2	~Egg()	21
5.7.2	Member Function Documentation	21
5.7.2.1	getId()	21
5.7.2.2	getPosition()	21
5.7.2.3	getState()	22
5.7.2.4	getTeam()	22
5.7.2.5	setId()	22
5.7.2.6	setPosition()	22
5.7.2.7	setState()	23
5.7.2.8	setTeam()	23
5.8	Gui::Engine Class Reference	23
5.8.1	Constructor & Destructor Documentation	23
5.8.1.1	Engine()	24
5.8.1.2	~Engine()	24
5.8.2	Member Function Documentation	24
5.8.2.1	run()	24
5.9	Gui::Errors::Error Class Reference	24
5.9.1	Detailed Description	25
5.10	Gui::Event Class Reference	25
5.10.1	Constructor & Destructor Documentation	26
5.10.1.1	Event()	26
5.10.1.2	~Event()	27
5.10.2	Member Function Documentation	27
5.10.2.1	listen()	27
5.11	Gui::GameData Class Reference	27
5.11.1	Constructor & Destructor Documentation	29
5.11.1.1	GameData()	29

5.11.1.2 ~GameData()	29
5.11.2 Member Function Documentation	29
5.11.2.1 addPlayerToTeam()	29
5.11.2.2 addServerEgg()	29
5.11.2.3 addTeam() [1/2]	30
5.11.2.4 addTeam() [2/2]	30
5.11.2.5 getIsEndGame()	30
5.11.2.6 getLastError()	31
5.11.2.7 getLastTick()	31
5.11.2.8 getLastTickMctCommand()	31
5.11.2.9 getMap()	31
5.11.2.10 getMapSize()	32
5.11.2.11 getNbBCTCommandReceived()	32
5.11.2.12 getPlayer()	32
5.11.2.13 getServerEggs()	32
5.11.2.14 getServerTick()	33
5.11.2.15 getTeam()	33
5.11.2.16 getTeamById()	33
5.11.2.17 getTeams()	34
5.11.2.18 getTile()	34
5.11.2.19 getTimeUnitFromServer()	34
5.11.2.20 removeServerEgg()	34
5.11.2.21 restartLastTick()	35
5.11.2.22 restartLastTickMctCommand()	35
5.11.2.23 setIsEndGame()	35
5.11.2.24 setLastError()	35
5.11.2.25 setMap()	36
5.11.2.26 setMapSize()	36
5.11.2.27 setNbBCTCommandReceived()	36
5.11.2.28 setServerTick()	37
5.11.2.29 setTile()	37
5.11.2.30 setTimeUnitFromServer()	37
5.12 Gui::Errors::GuiGameDataException Class Reference	37
5.12.1 Detailed Description	38
5.12.2 Constructor & Destructor Documentation	38
5.12.2.1 GuiGameDataException()	38
5.13 Gui::GUIUpdater Class Reference	39
5.13.1 Constructor & Destructor Documentation	40
5.13.1.1 GUIUpdater()	40
5.13.2 Member Function Documentation	40
5.13.2.1 update()	40
5.14 Gui::Errors::GuiUpdaterException Class Reference	40

5.14.1 Detailed Description	41
5.14.2 Constructor & Destructor Documentation	41
5.14.2.1 GuiUpdaterException()	41
5.15 Gui::HudGame Class Reference	42
5.15.1 Constructor & Destructor Documentation	43
5.15.1.1 HudGame()	43
5.15.1.2 ~HudGame()	43
5.15.2 Member Function Documentation	43
5.15.2.1 display()	44
5.16 Gui::HudPlayer Class Reference	44
5.16.1 Constructor & Destructor Documentation	45
5.16.1.1 HudPlayer()	45
5.16.1.2 ~HudPlayer()	45
5.16.2 Member Function Documentation	45
5.16.2.1 display()	46
5.17 Gui::HudTile Class Reference	46
5.17.1 Constructor & Destructor Documentation	47
5.17.1.1 HudTile()	47
5.17.1.2 ~HudTile()	48
5.17.2 Member Function Documentation	48
5.17.2.1 display()	48
5.17.2.2 displayNbEggs()	48
5.17.2.3 displayNbPlayers()	48
5.18 Gui::Errors::IError Class Reference	48
5.18.1 Member Function Documentation	49
5.18.1.1 what()	49
5.19 Gui::IEvent Class Reference	49
5.19.1 Constructor & Destructor Documentation	50
5.19.1.1 IEvent()	50
5.19.1.2 ~IEvent()	50
5.19.2 Member Function Documentation	50
5.19.2.1 listen()	50
5.19.2.2 setGameData()	50
5.19.2.3 setRender()	51
5.20 Gui::IGUIUpdater Class Reference	51
5.20.1 Member Function Documentation	51
5.20.1.1 update()	51
5.21 Gui::IHud Class Reference	52
5.21.1 Member Enumeration Documentation	52
5.21.1.1 TypeScene	52
5.21.2 Constructor & Destructor Documentation	52
5.21.2.1 ~IHud()	52

5.21.3 Member Function Documentation	52
5.21.3.1 display()	53
5.21.3.2 getType()	53
5.21.3.3 setPlayer()	53
5.21.3.4 setTile()	53
5.22 Gui::INetwork Class Reference	54
5.22.1 Constructor & Destructor Documentation	54
5.22.1.1 ~INetwork()	54
5.22.2 Member Function Documentation	54
5.22.2.1 connectToServer()	54
5.22.2.2 getHostName()	55
5.22.2.3 getPort()	55
5.22.2.4 listenServer()	55
5.22.2.5 sendMessageServer()	55
5.22.2.6 setHostName()	56
5.22.2.7 setPort()	56
5.23 Gui::Inventory Class Reference	56
5.23.1 Constructor & Destructor Documentation	57
5.23.1.1 Inventory() [1/2]	58
5.23.1.2 Inventory() [2/2]	58
5.23.1.3 ~Inventory()	58
5.23.2 Member Function Documentation	58
5.23.2.1 addResource()	58
5.23.2.2 getDeraumere()	59
5.23.2.3 getFood()	59
5.23.2.4 getLinemate()	59
5.23.2.5 getMendiane()	60
5.23.2.6 getPhiras()	60
5.23.2.7 getRessources()	60
5.23.2.8 getSibur()	60
5.23.2.9 getThystame()	61
5.23.2.10 removeResource()	61
5.23.2.11 setDeraumere()	61
5.23.2.12 setFood()	61
5.23.2.13 setLinemate()	62
5.23.2.14 setMendiane()	62
5.23.2.15 setPhiras()	62
5.23.2.16 setRessources()	63
5.23.2.17 setSibur()	63
5.23.2.18 setThystame()	63
5.24 Gui::IServerParser Class Reference	63
5.24.1 Member Function Documentation	64

5.24.1.1 parse()	64
5.25 Gui::Network Class Reference	64
5.25.1 Constructor & Destructor Documentation	66
5.25.1.1 Network()	66
5.25.1.2 ~Network()	66
5.25.2 Member Function Documentation	66
5.25.2.1 connectToServer()	66
5.25.2.2 listenServer()	66
5.25.2.3 sendMessageServer()	67
5.26 Gui::Errors::NetworkException Class Reference	67
5.26.1 Detailed Description	68
5.26.2 Constructor & Destructor Documentation	68
5.26.2.1 NetworkException()	68
5.27 Gui::ParseCommandLine Class Reference	68
5.27.1 Constructor & Destructor Documentation	69
5.27.1.1 ParseCommandLine()	69
5.27.1.2 ~ParseCommandLine()	69
5.27.2 Member Function Documentation	69
5.27.2.1 getHostName()	69
5.27.2.2 getPort()	70
5.27.2.3 parseFlags()	70
5.28 Gui::Errors::ParseCommandLineException Class Reference	70
5.28.1 Detailed Description	71
5.28.2 Constructor & Destructor Documentation	71
5.28.2.1 ParseCommandLineException()	71
5.29 Gui::Player Class Reference	72
5.29.1 Constructor & Destructor Documentation	73
5.29.1.1 Player()	73
5.29.1.2 ~Player()	73
5.29.2 Member Function Documentation	74
5.29.2.1 getAnimationTimeEllapsed()	74
5.29.2.2 getBroadcast()	74
5.29.2.3 getCenterPosition()	74
5.29.2.4 getCurrentFrame()	74
5.29.2.5 getId()	75
5.29.2.6 getLevel()	75
5.29.2.7 getOrientation()	75
5.29.2.8 getPosition()	75
5.29.2.9 getPosition3D()	76
5.29.2.10 getRotationFromOrientation()	76
5.29.2.11 getState()	76
5.29.2.12 getTeam()	76

5.29.2.13 restartAnimationTimeElapsed()	77
5.29.2.14 setBroadcast()	77
5.29.2.15 setCurrentFrame()	78
5.29.2.16 setId()	78
5.29.2.17 setLevel()	78
5.29.2.18 setOrientation()	79
5.29.2.19 setPosition()	79
5.29.2.20 setPosition3D()	79
5.29.2.21 setState()	79
5.29.2.22 setTeam()	80
5.29.3 Member Data Documentation	80
5.29.3.1 inventory	80
5.30 Gui::Render Class Reference	80
5.30.1 Constructor & Destructor Documentation	81
5.30.1.1 Render()	81
5.30.1.2 ~Render()	82
5.30.2 Member Function Documentation	82
5.30.2.1 changePlayerPOV()	82
5.30.2.2 changePOVToFirstPerson()	82
5.30.2.3 changePOVToSecondPerson()	82
5.30.2.4 changePOVToThirdPerson()	83
5.30.2.5 draw()	83
5.30.2.6 getCamera()	83
5.30.2.7 getCameraPlayerPov()	83
5.30.2.8 getCameraTile()	84
5.30.2.9 getCameraType()	84
5.30.2.10 getIsDebug()	84
5.30.2.11 getRenderDistance()	84
5.30.2.12 getTileModel()	85
5.30.2.13 getTimeUnit()	85
5.30.2.14 isCameraInPlayerPov()	85
5.30.2.15 isOpen()	85
5.30.2.16 setCameraPlayerPov()	85
5.30.2.17 setCameraTile()	86
5.30.2.18 setCameraType()	86
5.30.2.19 setIsDebug()	86
5.30.2.20 setPlayerPov()	87
5.30.2.21 setRenderDistance()	87
5.30.2.22 setTimeUnit()	87
5.31 Gui::ServerParser Class Reference	87
5.31.1 Constructor & Destructor Documentation	88
5.31.1.1 ServerParser()	88

5.31.1.2 ~ServerParser()	88
5.31.2 Member Function Documentation	88
5.31.2.1 parse()	88
5.32 Gui::Errors::ServerParserException Class Reference	89
5.32.1 Detailed Description	90
5.32.2 Constructor & Destructor Documentation	90
5.32.2.1 ServerParserException()	90
5.33 Gui::Team Class Reference	90
5.33.1 Constructor & Destructor Documentation	91
5.33.1.1 Team()	91
5.33.1.2 ~Team()	92
5.33.2 Member Function Documentation	92
5.33.2.1 addEgg()	92
5.33.2.2 addPlayer()	92
5.33.2.3 getEgg()	93
5.33.2.4 getEggModel()	93
5.33.2.5 getEggs()	93
5.33.2.6 getName()	93
5.33.2.7 getPlayer()	93
5.33.2.8 getPlayerBoundingBoxes()	94
5.33.2.9 getPlayerColor()	94
5.33.2.10 getPlayerModel()	94
5.33.2.11 getPlayerModelAnimation()	95
5.33.2.12 getPlayerModelHitbox()	95
5.33.2.13 getPlayerPositionIn3DSpace()	95
5.33.2.14 getPlayers()	95
5.33.2.15 isPlayerHit()	96
5.33.2.16 removeEgg()	96
5.33.2.17 removePlayer()	96
5.33.2.18 setEggModelPath()	98
5.33.2.19 setName()	98
5.33.2.20 setPlayerModelPath()	98
5.34 Gui::Tile Class Reference	99
5.34.1 Constructor & Destructor Documentation	99
5.34.1.1 Tile() [1/2]	99
5.34.1.2 Tile() [2/2]	100
5.34.1.3 ~Tile()	100
5.34.2 Member Function Documentation	100
5.34.2.1 getPosition()	100
5.34.2.2 getPositionIn3DSpace()	101
5.34.2.3 getTileBoundingBoxes()	101
5.34.2.4 getTileModelHitbox()	101

5.34.2.5 isTileHit()	102
5.34.2.6 setPosition()	102
5.34.3 Member Data Documentation	102
5.34.3.1 inventory	102
5.35 Gui::UserCamera Class Reference	103
5.35.1 Constructor & Destructor Documentation	103
5.35.1.1 UserCamera()	104
5.35.1.2 ~UserCamera()	104
5.35.2 Member Function Documentation	104
5.35.2.1 getCamera()	104
5.35.2.2 getFovy()	104
5.35.2.3 getPlayerId()	105
5.35.2.4 getPosition()	105
5.35.2.5 getTarget()	105
5.35.2.6 getTilePos()	105
5.35.2.7 getType()	106
5.35.2.8 getUp()	106
5.35.2.9 isPlayerPov()	106
5.35.2.10 setFovy()	106
5.35.2.11 setPlayerId()	107
5.35.2.12 setPosition()	107
5.35.2.13 setTarget()	107
5.35.2.14 setTilePos()	108
5.35.2.15 setType()	108
5.35.2.16 setUp()	108
6 File Documentation	109
6.1 Assets.hpp	109
6.2 Colors.hpp	110
6.3 Config.hpp	111
6.4 Engine.hpp	111
6.5 AError.hpp	112
6.6 Error.hpp	112
6.7 IError.hpp	113
6.8 AEvent.hpp	113
6.9 Event.hpp	114
6.10 IEvent.hpp	115
6.11 Egg.hpp	115
6.12 GameData.hpp	116
6.13 Inventory.hpp	117
6.14 Player.hpp	118
6.15 Team.hpp	119

6.16 Tile.hpp	120
6.17 AGUIUpdater.hpp	121
6.18 GUIUpdater.hpp	121
6.19 IGUIUpdater.hpp	123
6.20 AHud.hpp	123
6.21 HudGame.hpp	124
6.22 HudPlayer.hpp	124
6.23 HudTile.hpp	125
6.24 IHud.hpp	125
6.25 ANetwork.hpp	126
6.26 INetwork.hpp	127
6.27 Network.hpp	127
6.28 IServerParser.hpp	128
6.29 ParseCommandLine.hpp	128
6.30 ServerParser.hpp	129
6.31 Decoration.hpp	130
6.32 Render.hpp	130
6.33 UserCamera.hpp	132
6.34 Types.hpp	133
Index	135

Chapter 1

Zappy GUI

1.1 Technology Notes

The graphic part of Zappy was developed using C++.

The graphic library used in this project is `raylib` developed by Raysan5.

1.2 Installing Raylib on Fedora

1.2.1 Prerequisites

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

1.2.2 Installation Steps

1.2.2.1 1. Update Your System

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

```
sudo dnf update
```

1.2.2.2 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
```

1.2.2.3 3. Verify the Installation

To ensure Raylib is installed correctly, you can compile and run a simple example. Create a file `main.c` with the include **raylib.h**. To run the program, don't forget to use the flag **-lraylib**.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Gui::Decoration	18
Gui::Egg	20
Gui::Engine	23
std::exception	
Gui::Errors::IError	48
Gui::Errors::AError	9
Gui::Errors::Error	24
Gui::Errors::GuiGameDataException	37
Gui::Errors::GuiUpdaterException	40
Gui::Errors::NetworkException	67
Gui::Errors::ParseCommandLineException	70
Gui::Errors::ServerParserException	89
Gui::GameData	27
Gui::IEvent	49
Gui::AEvent	10
Gui::Event	25
Gui::IGUIUpdater	51
Gui::AGUIUpdater	10
Gui::GUIUpdater	39
Gui::IHud	52
Gui::AHud	12
Gui::HudGame	42
Gui::HudPlayer	44
Gui::HudTile	46
Gui::INetwork	54
Gui::ANetwork	14
Gui::Network	64
Gui::Inventory	56
Gui::IServerParser	63
Gui::ServerParser	87
Gui::ParseCommandLine	68
Gui::Player	72
Gui::Render	80
Gui::Team	90
Gui::Tile	99
Gui::UserCamera	103

Chapter 3

Class Index

3.1 Class List

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

Gui::Errors::AError	
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 <code>_message</code> to store the error message	9
Gui::AEvent	10
Gui::AGUIUpdater	10
Gui::AHud	12
Gui::ANetwork	14
Gui::Decoration	18
Gui::Egg	20
Gui::Engine	23
Gui::Errors::Error	
Base class for argument-related errors	24
Gui::Event	25
Gui::GameData	27
Gui::Errors::GuiGameDataException	
Error class for GameData errors	37
Gui::GUIUpdater	39
Gui::Errors::GuiUpdaterException	
Error class for GUIUpdater errors	40
Gui::HudGame	42
Gui::HudPlayer	44
Gui::HudTile	46
Gui::Errors::IError	48
Gui::IEvent	49
Gui::IGUIUpdater	51
Gui::IHud	52
Gui::INetwork	54
Gui::Inventory	56
Gui::IServerParser	63
Gui::Network	64
Gui::Errors::NetworkException	
Error class for network errors	67
Gui::ParseCommandLine	68
Gui::Errors::ParseCommandLineException	
Error class for <code>parseCommandLine</code> errors	70

Gui::Player	72
Gui::Render	80
Gui::ServerParser	87
Gui::Errors::ServerParserException	
Error class for network errors	89
Gui::Team	90
Gui::Tile	99
Gui::UserCamera	103

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	109
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Colors.hpp	110
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Config.hpp	111
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Types.hpp	133
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Engine/Engine.hpp	111
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/AError.hpp	112
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp	112
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/IError.hpp	113
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/AEvent.hpp	113
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/Event.hpp	114
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/IEvent.hpp	115
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Egg.hpp	115
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/GameData.hpp	116
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Inventory.hpp	117
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Player.hpp	118
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Team.hpp	119
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Tile.hpp	120
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/AGUIUpdater.hpp	121
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/GUIUpdater.hpp	121
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/IGUIUpdater.hpp	123
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/AHud.hpp	123
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp	124
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudPlayer.hpp	124
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudTile.hpp	125
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/IHud.hpp	125
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/ANetwork.hpp	126
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/INetwork.hpp	127
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/Network.hpp	127
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/IServerParser.hpp	128
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ParseCommandLine.hpp	128
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ServerParser.hpp	129
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Decoration.hpp	130
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Render.hpp	130
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/UserCamera.hpp	132

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 GameData 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.

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.

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](#) }
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::HudGame](#), [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>	Player 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>	Tile to display infos.
-------------	--

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 const std::string [listenServer](#) ()=0
Listen the server and return it message.
- virtual void [sendMessageServer](#) (const std::string &message)=0
Send a message to the Server.

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 const std::string `listenServer` ()=0
Listen to the server.
- virtual void `sendMessageServer` (const std::string &message)=0
Send a message to the server.

Protected Attributes

- int `_port`
Port of the server.
- std::string `_hostName`
Host name of the server.

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 getHostName()

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

Get the host name object.

Returns

std::string Host of the server.

Implements [Gui::INetwork](#).

5.5.2.3 getPort()

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

Get the host name object.

Returns

std::string Host of the server.

Implements [Gui::INetwork](#).

5.5.2.4 listenServer()

```
virtual const std::string Gui::ANetwork::listenServer ( ) [pure virtual]
```

Listen the server and return it message.

Returns

std::string - Message of the server.

Implements [Gui::INetwork](#).

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

5.5.2.5 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](#).

5.5.2.6 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](#).

5.5.2.7 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)
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 )
```

Display decorations.

Parameters

<i>mapSize</i>	Size of the map.
<i>renderDistance</i>	Distance to render.
<i>camPos</i>	Position of the camera.

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>	Team 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>	Team 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>	Network 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.

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.

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, const std::string &playerModelPath, const std::string &eggModelPath, 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.

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>	Player 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,
    const std::string & playerModelPath,
    const std::string & eggModelPath,
    Color playerColor )
```

Add a team to the game.

Parameters

<i>name</i>	Name of the team.
<i>playerModelPath</i>	Path to the asset of the team for players.
<i>eggModelPath</i>	Path to the asset of the team for eggs.
<i>playerColor</i>	Color of the team.

5.11.2.5 getIsEndGame()

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

Get the IsEnd Game object.

Returns

- true - The game is finished.
- false - The game continue.

5.11.2.6 getLastError()

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

Get the Last Error object.

Returns

std::string - Last error message.

5.11.2.7 getLastTick()

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

Get the Last Tick object.

Returns

clock_t - Last Tick.

5.11.2.8 getLastTickMctCommand()

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

Get the Last Tick mct command object.

Returns

clock_t - Last Tick Mct command.

5.11.2.9 getMap()

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

Get the Map object.

Returns

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

5.11.2.10 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.

5.11.2.11 getNbBCTCommandReceived()

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

Get the number of bct command received.

Returns

std::size_t - Number of bct command received.

5.11.2.12 getPlayer()

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

Get a player object.

Parameters

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

5.11.2.13 getServerEggs()

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

Get the Server Eggs object.

Returns

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

5.11.2.14 getServerTick()

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

Get the Server Tick object.

Returns

std::size_t - Server Tick.

5.11.2.15 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.

5.11.2.16 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.

5.11.2.17 getTeams()

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

Get the Teams object.

Returns

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

5.11.2.18 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.

5.11.2.19 getTimeUnitFromServer()

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

Get the Time Unit From Server object.

Returns

true - The time unit has changed.

5.11.2.20 removeServerEgg()

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

Remove an egg from the server ones.

Parameters

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

5.11.2.21 restartLastTick()

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

Restart the last tick clock.

5.11.2.22 restartLastTickMctCommand()

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

Restart the last tick mct command clock.

5.11.2.23 setIsEndGame()

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

Set the IsEnd Game object.

Parameters

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

5.11.2.24 setLastError()

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

Set the Last Error object.

Parameters

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

5.11.2.25 setMap()

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

Set the Map object.

Parameters

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

5.11.2.26 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.

5.11.2.27 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.
-----------	---------------------------------

5.11.2.28 setServerTick()

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

Set the Server Tick object.

Parameters

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

5.11.2.29 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>	Tile to set.

5.11.2.30 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 GameData 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 GameData .
<i>data</i>	The data to update the GUI GameData .

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::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.

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.

Additional Inherited Members

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

- enum [TypeScene](#) { [GAME](#) , [POV_PLAYER](#) , [END_GAME](#) , [TILE](#) }
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 HudGame()

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

Construct a new Hud Game object.

Parameters

<i>gameData</i>	GameData class.
-----------------	---------------------------------

5.15.1.2 ~HudGame()

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

Destroy the Hud Game object.

5.15.2 Member Function Documentation

5.15.2.1 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

5.16 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.

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.

Additional Inherited Members

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

- enum [TypeScene](#) { [GAME](#) , [POV_PLAYER](#) , [END_GAME](#) , [TILE](#) }
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 HudPlayer()

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

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

Parameters

<i>gameData</i>	GameData class.
-----------------	---------------------------------

5.16.1.2 ~HudPlayer()

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

Destroy the Hud [Player](#) object.

5.16.2 Member Function Documentation

5.16.2.1 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

5.17 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.

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.

Additional Inherited Members**Public Types inherited from Gui::IHud**

- enum [TypeScene](#) { [GAME](#) , [POV_PLAYER](#) , [END_GAME](#) , [TILE](#) }
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 HudTile()**

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

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

Parameters

<i>gameData</i>	GameData class.
-----------------	---------------------------------

5.17.1.2 ~HudTile()

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

Destroy the Hud [Tile](#) object.

5.17.2 Member Function Documentation

5.17.2.1 display()

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

Display [Tile](#) Hud.

Implements [Gui::AHud](#).

5.17.2.2 displayNbEggs()

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

Display number of eggs.

5.17.2.3 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](#)

5.18 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.

5.18.1 Member Function Documentation

5.18.1.1 `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`

5.19 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.

5.19.1 Constructor & Destructor Documentation

5.19.1.1 IEvent()

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

Construct a new [IEvent](#) object.

5.19.1.2 ~IEvent()

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

Destroy the [IEvent](#) object.

5.19.2 Member Function Documentation

5.19.2.1 listen()

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

Listen the user's events.

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

5.19.2.2 setGameData()

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

Set the [GameData](#) object.

Parameters

<i>gameData</i>	GameData class.
-----------------	---------------------------------

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

5.19.2.3 setRender()

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

Set the [Render](#) object.

Parameters

<i>render</i>	Render 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

5.20 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](#).

5.20.1 Member Function Documentation

5.20.1.1 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

5.21 Gui::IHud Class Reference

Inheritance diagram for Gui::IHud:

Public Types

- enum [TypeScene](#) { **GAME** , **POV_PLAYER** , **END_GAME** , **TILE** }
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.

5.21.1 Member Enumeration Documentation

5.21.1.1 TypeScene

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

Hud enum for the different scenes.

5.21.2 Constructor & Destructor Documentation

5.21.2.1 ~IHud()

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

Destroy the [IHud](#) object.

5.21.3 Member Function Documentation

5.21.3.1 display()

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

Display the Hud.

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

5.21.3.2 getType()

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

Get the Type object.

Returns

TypeScene - Type of the scene.

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

5.21.3.3 setPlayer()

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

Set the [Player](#) object.

Parameters

<i>player</i>	Player to display infos.
---------------	--

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

5.21.3.4 setTile()

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

Set the [Tile](#) object.

Parameters

<i>tile</i>	Tile to display infos.
-------------	--

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](#)

5.22 Gui::INetwork Class Reference

Inheritance diagram for Gui::INetwork:

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 const std::string [listenServer](#) ()=0
Listen to the server.
- virtual void [sendMessageServer](#) (const std::string &message)=0
Send a message to the server.

5.22.1 Constructor & Destructor Documentation

5.22.1.1 ~INetwork()

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

Destroy the [INetwork](#) object.

5.22.2 Member Function Documentation

5.22.2.1 connectToServer()

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

Connect to the server.

Exceptions

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

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

5.22.2.2 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](#).

5.22.2.3 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](#).

5.22.2.4 listenServer()

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

Listen to the server.

Returns

std::string Message from the server.

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

5.22.2.5 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](#).

5.22.2.6 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](#).

5.22.2.7 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

5.23 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 thysame)
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.

5.23.1 Constructor & Destructor Documentation

5.23.1.1 Inventory() [1/2]

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

Construct a new [Inventory](#) object.

5.23.1.2 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.

5.23.1.3 ~Inventory()

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

Destroy the [Inventory](#) object.

5.23.2 Member Function Documentation

5.23.2.1 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.

5.23.2.2 getDeraumere()

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

Get the Deraumere object.

Returns

std::size_t - deraumere

5.23.2.3 getFood()

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

Get the Food object.

Returns

std::size_t - food

5.23.2.4 getLinemate()

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

Get the Linemate object.

Returns

std::size_t - linemate

5.23.2.5 getMendiane()

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

Get the Mendiane object.

Returns

std::size_t - mendiane

5.23.2.6 getPhiras()

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

Get the Phiras object.

Returns

std::size_t - phiras

5.23.2.7 getRessources()

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

Get the Ressources object.

Returns

Ressources - ressources

5.23.2.8 getSibur()

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

Get the Sibur object.

Returns

std::size_t - sibur

5.23.2.9 getThystame()

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

Get the Thystame object.

Returns

std::size_t - thystame

5.23.2.10 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.

5.23.2.11 setDeraumere()

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

Set the Deraumere object.

Parameters

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

5.23.2.12 setFood()

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

Set the Food object.

Parameters

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

5.23.2.13 setLinemate()

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

Set the Linemate object.

Parameters

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

5.23.2.14 setMendiane()

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

Set the Mendiane object.

Parameters

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

5.23.2.15 setPhiras()

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

Set the Phiras object.

Parameters

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

5.23.2.16 setRessources()

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

Set the Ressources object.

Parameters

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

5.23.2.17 setSibur()

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

Set the Sibur object.

Parameters

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

5.23.2.18 setThystame()

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

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

5.24 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.

5.24.1 Member Function Documentation

5.24.1.1 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`

5.25 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.
- const std::string [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 const std::string **listenServer** ()=0
Listen the server and return it message.
- virtual void **sendMessageServer** (const std::string &message)=0
Send a message to the Server.

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 const std::string **listenServer** ()=0
Listen to the server.
- virtual void **sendMessageServer** (const std::string &message)=0
Send a message to the server.

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

- int **_port**
Port of the server.
- std::string **_hostName**
Host name of the server.

5.25.1 Constructor & Destructor Documentation

5.25.1.1 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.

5.25.1.2 ~Network()

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

Destroy the [Network](#) object.

5.25.2 Member Function Documentation

5.25.2.1 connectToServer()

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

Connect the Gui network with the server.

Exceptions

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

Implements [Gui::ANetwork](#).

5.25.2.2 listenServer()

```
const std::string Gui::Network::listenServer ( ) [virtual]
```

Listen the server and return it message.

Returns

std::string - Message of the server.

Implements [Gui::ANetwork](#).

5.25.2.3 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

5.26 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.

5.26.1 Detailed Description

[Error](#) class for network errors.

5.26.2 Constructor & Destructor Documentation

5.26.2.1 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`

5.27 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 ()`
- Get the port object.*
 - `std::string getHostName ()`
- Get the hostName object.*

5.27.1 Constructor & Destructor Documentation

5.27.1.1 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.

5.27.1.2 ~ParseCommandLine()

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

Destroy the Parse Command Line object.

5.27.2 Member Function Documentation

5.27.2.1 getHostName()

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

Get the hostName object.

Returns

std::string - hostName

5.27.2.2 `getPort()`

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

Get the port object.

Returns

int - port

5.27.2.3 `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`

5.28 `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.

5.28.1 Detailed Description

[Error](#) class for parseCommandLine errors.

5.28.2 Constructor & Destructor Documentation**5.28.2.1 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`

5.29 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.

5.29.1 Constructor & Destructor Documentation

5.29.1.1 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>	Team name of the player.
<i>position</i>	Position of the Player .

5.29.1.2 ~Player()

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

Destroy the [Player](#) object.

5.29.2 Member Function Documentation

5.29.2.1 getAnimationTimeElapsed()

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

Get the Animation Time Ellapsed object.

Returns

clock_t - Animation time ellapsed.

5.29.2.2 getBroadcast()

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

Get the Broadcast object.

Returns

std::string - Broadcast message.

5.29.2.3 getCenterPosition()

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

Get the Center Position object.

Returns

Vector3 - Center position.

5.29.2.4 getCurrentFrame()

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

Get the Current Frame object.

Returns

int - Current frame.

5.29.2.5 getId()

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

Get the Id object.

Returns

std::size_t - id

5.29.2.6 getLevel()

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

Get the Level object.

Returns

std::size_t - level

5.29.2.7 getOrientation()

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

Get the Orientation object.

Returns

std::size_t - orientation

5.29.2.8 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

5.29.2.9 getPosition3D()

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

Get the Position3D object.

Returns

Vector3 - position3D

5.29.2.10 getRotationFromOrientation()

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

Get the Vector From Orientation object.

Returns

Vector3 - Vector3 from orientation.

5.29.2.11 getState()

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

Get the State object.

Returns

PlayerState - [Player](#) state.

5.29.2.12 getTeam()

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

Get the [Team](#) object.

Returns

std::string - team name

5.29.2.13 restartAnimationTimeEllapsed()

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

Restart the timer animation.

5.29.2.14 setBroadcast()

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

Set the Broadcast object.

Parameters

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

5.29.2.15 setCurrentFrame()

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

Set the Current Frame object.

Parameters

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

5.29.2.16 setId()

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

Set the Id object.

Parameters

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

5.29.2.17 setLevel()

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

Set the Level object.

Parameters

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

5.29.2.18 setOrientation()

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

Set the Orientation object.

Parameters

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

5.29.2.19 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.
-----------------	-------------------------

5.29.2.20 setPosition3D()

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

Set the Position3D object.

Parameters

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

5.29.2.21 setState()

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

Set the State object.

Parameters

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

5.29.2.22 setTeam()

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

Set the [Team](#) object.

Parameters

<i>team</i>	Team name of the player.
-------------	--

5.29.3 Member Data Documentation**5.29.3.1 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

5.30 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.*

5.30.1 Constructor & Destructor Documentation

5.30.1.1 Render()

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

Construct a new [Render](#) object.

5.30.1.2 ~Render()

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

Destroy the [Render](#) object.

5.30.2 Member Function Documentation

5.30.2.1 changePlayerPOV()

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

Change the player point of view.

Parameters

<i>playerId</i>	Player id to select.
-----------------	--------------------------------------

Note

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

5.30.2.2 changePOVToFirstPerson()

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

Change the camera to the player.

Parameters

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

5.30.2.3 changePOVToSecondPerson()

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

Change the camera to the player.

Parameters

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

5.30.2.4 changePOVToThirdPerson()

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

Change the camera to the player.

Parameters

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

5.30.2.5 draw()

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

Draw the scene.

5.30.2.6 getCamera()

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

Get the Camera object.

Returns

std::shared_ptr<Camera> - camera

5.30.2.7 getCameraPlayerPov()

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

Get the Camera player pov id.

Returns

std::size_t - Id of the player.

5.30.2.8 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.

5.30.2.9 getCameraType()

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

Get the Type object.

Returns

CameraType - Camera type.

5.30.2.10 getIsDebug()

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

Get the Is Debug object.

Returns

true - display debug

false - do not display debug

5.30.2.11 getRenderDistance()

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

Get the [Render](#) Distance value.

5.30.2.12 `getTileModel()`

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

Get the [Tile](#) model.

5.30.2.13 `getTimeUnit()`

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

Get the Time Unit value.

Returns

`size_t` - Time unit value.

5.30.2.14 `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.

5.30.2.15 `isOpen()`

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

Check if the window is open.

Returns

`true` - the window is open

`false` - the window is closed

5.30.2.16 `setCameraPlayerPov()`

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

Set the Camera player pov id.

Parameters

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

5.30.2.17 setCameraTile()

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

Set the Camera [Tile](#) object.

Parameters

<i>pos</i>	Tile position.
------------	--------------------------------

5.30.2.18 setCameraType()

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

Set the Type object.

Parameters

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

5.30.2.19 setIsDebug()

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

Set the Is Debug object.

Parameters

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

5.30.2.20 setPlayerPov()

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

Sets the Pov of the player.

Parameters

<i>playerId</i>	Player id to select.
-----------------	----------------------

5.30.2.21 setRenderDistance()

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

Set the [Render](#) Distance value.

Parameters

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

5.30.2.22 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

5.31 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.

5.31.1 Constructor & Destructor Documentation

5.31.1.1 [ServerParser](#)()

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

Construct a new Server Parser object.

5.31.1.2 [~ServerParser](#)()

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

Destroy the Server Parser object.

5.31.2 Member Function Documentation

5.31.2.1 [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

5.32 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.

5.32.1 Detailed Description

[Error](#) class for network errors.

5.32.2 Constructor & Destructor Documentation

5.32.2.1 [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`

5.33 [Gui::Team](#) Class Reference

Public Member Functions

- [Team](#) (const std::string &name, const std::string &playerModelPath, const std::string &eggModelPath, 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.*

5.33.1 Constructor & Destructor Documentation

5.33.1.1 Team()

```
Gui::Team::Team (
    const std::string & name,
    const std::string & playerModelPath,
    const std::string & eggModelPath,
    Color playerColor )
```

Construct a new [Team](#) object.

Parameters

<i>name</i>	Name of the team.
<i>playerModelPath</i>	Path to the team model asset for players.
<i>eggSkinPath</i>	Path to the skin of the team.
<i>playerColor</i>	Color of the players.

5.33.1.2 ~Team()

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

Destroy the [Team](#) object.

5.33.2 Member Function Documentation**5.33.2.1 addEgg()**

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

Add an egg to the team.

Parameters

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

5.33.2.2 addPlayer()

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

Add a player to the team.

Parameters

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

5.33.2.3 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](#).

5.33.2.4 getEggModel()

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

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

Returns

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

5.33.2.5 getEggs()

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

Get the Eggs object.

Returns

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

5.33.2.6 getName()

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

Get the Name object.

Returns

const std::string& Name of the team.

5.33.2.7 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](#).

5.33.2.8 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.

5.33.2.9 getPlayerColor()

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

Get the [Player](#) Color object.

5.33.2.10 getPlayerModel()

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

Get the Model object.

Returns

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

5.33.2.11 getPlayerModelAnimation()

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

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

Returns

ModelAnimation* - Players' animations.

5.33.2.12 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.

5.33.2.13 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.

5.33.2.14 getPlayers()

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

Get the Players object.

Returns

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

5.33.2.15 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.

5.33.2.16 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.

5.33.2.17 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.

5.33.2.18 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.
--------------------	-------------------------------------

5.33.2.19 setName()

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

Set the Name object.

Parameters

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

5.33.2.20 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

5.34 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.

5.34.1 Constructor & Destructor Documentation

5.34.1.1 [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.
-----------------	-------------------------------

5.34.1.2 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.

5.34.1.3 ~Tile()

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

Destroy the [Tile](#) object.

5.34.2 Member Function Documentation**5.34.2.1 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

5.34.2.2 getPositionIn3DSpace()

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

Get the Position In Space object.

Returns

Vector3 - Position in space.

5.34.2.3 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.

5.34.2.4 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.

5.34.2.5 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.

5.34.2.6 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.
-----------------	---------------------------

5.34.3 Member Data Documentation

5.34.3.1 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

5.35 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.
- size_t [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.

5.35.1 Constructor & Destructor Documentation

5.35.1.1 UserCamera()

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

Construct a new User Camera object.

5.35.1.2 ~UserCamera()

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

Destroy the User Camera object.

5.35.2 Member Function Documentation

5.35.2.1 getCamera()

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

Get the Camera object.

Returns

Camera - camera

5.35.2.2 getFovy()

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

Get the Fovy object.

Returns

float - fovy

5.35.2.3 getPlayerId()

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

Get the [Player](#) Id object.

Returns

size_t - [Player](#) id.

5.35.2.4 getPosition()

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

Get the Position object.

Returns

Vector3 - position

5.35.2.5 getTarget()

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

Get the Target object.

Returns

Vector3 - target

5.35.2.6 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.

5.35.2.7 getType()

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

Get the Type object.

Returns

CameraType - Camera type.

5.35.2.8 getUp()

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

Get the Up object.

Returns

Vector3 - up

5.35.2.9 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.

5.35.2.10 setFovy()

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

Set the Fovy object.

Parameters

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

5.35.2.11 setPlayerId()

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

Set the [Player](#) Id object.

Parameters

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

5.35.2.12 setPosition()

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

Set the Position object.

Parameters

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

5.35.2.13 setTarget()

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

Set the Target object.

Parameters

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

5.35.2.14 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.
------------	-----------------------

5.35.2.15 setType()

```
void Gui::UserCamera::setType (
    CameraType type )
```

Set the Type object.

Parameters

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

5.35.2.16 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  #define PATH_ASSETS          "gui/assets/"
00011
00012  #define PATH_RESOURCES      "resources/"
00013  #define PATH_PLAYER        "player/"
00014  #define PATH_TILE          "tile/"
00015  #define PATH_DECORATION    "decoration/"
00016  #define PATH_HUD           "hud/"
00017
00018  #define PNG_CURSOR         PATH_ASSETS PATH_HUD "cursor.png"
00019
00020  #define PNG_HUD            PATH_ASSETS PATH_HUD "hud.png"
00021  #define PNG_FOOD          PATH_ASSETS PATH_HUD "food.png"
00022  #define PNG_LINEMATE      PATH_ASSETS PATH_HUD "linemate.png"
00023  #define PNG_DERAUMERE     PATH_ASSETS PATH_HUD "deraumere.png"
00024  #define PNG_MENDIANE      PATH_ASSETS PATH_HUD "mendiane.png"
00025  #define PNG_PHIRAS        PATH_ASSETS PATH_HUD "phiras.png"
00026  #define PNG_SIBUR         PATH_ASSETS PATH_HUD "sibur.png"
00027  #define PNG_THYSTAME      PATH_ASSETS PATH_HUD "thystame.png"
00028  #define PNG_PLAYER        PATH_ASSETS PATH_HUD "player.png"
00029  #define PNG_TILE          PATH_ASSETS PATH_HUD "tile.png"
00030  #define PNG_EGG           PATH_ASSETS PATH_HUD "egg.png"
00031  #define FONT_HUD          PATH_ASSETS PATH_HUD "SimplyMono-Bold.ttf"
00032
00033  #define MODEL_TILE        PATH_ASSETS PATH_TILE "tile.glb"
00034  #define MODEL_FOOD        PATH_ASSETS PATH_RESOURCES "food.glb"
00035  #define MODEL_LINEMATE    PATH_ASSETS PATH_RESOURCES "linemate.glb"
00036  #define MODEL_MENDIANE    PATH_ASSETS PATH_RESOURCES "mendiane.glb"
00037  #define MODEL_PHIRAS      PATH_ASSETS PATH_RESOURCES "phiras.glb"
00038  #define MODEL_SIBUR       PATH_ASSETS PATH_RESOURCES "sibur.glb"
00039  #define MODEL_THYSTAME    PATH_ASSETS PATH_RESOURCES "thystame.glb"
00040  #define MODEL_DERAUMERE   PATH_ASSETS PATH_RESOURCES "deraumere.glb"
00041  #define MODEL_PLAYER      PATH_ASSETS PATH_PLAYER "player.glb"
00042  #define MODEL_EGG         PATH_ASSETS PATH_PLAYER "egg.glb"
00043  #define MODEL_TREE        PATH_ASSETS PATH_DECORATION "tree.glb"
00044  #define MODEL_LANTERN     PATH_ASSETS PATH_DECORATION "lantern.glb"
00045
00046  #define SCALE_FOOD        (Vector3){1, 0.5, 1}
00047  #define SCALE_LINEMATE    (Vector3){0.1, 0.1, 0.1}
00048  #define SCALE_MENDIANE    (Vector3){0.1, 0.1, 0.1}
00049  #define SCALE_PHIRAS      (Vector3){0.001, 0.001, 0.005}
00050  #define SCALE_SIBUR       (Vector3){0.01, 0.01, 0.01}
00051  #define SCALE_THYSTAME    (Vector3){2, 2, 2}
00052  #define SCALE_DERAUMERE   (Vector3){0.5, 0.5, 0.5}
00053  #define SCALE_PLAYER      (Vector3){0.5, 0.5, 0.5}
00054  #define SCALE_EGG         (Vector3){0.5, 0.5, 0.5}
00055  #define SCALE_TREE        (Vector3){1, 1, 1}
00056  #define SCALE_LANTERN     (Vector3){1, 1, 1}
00057
00058  #define ROTATION_ANGLE_FOOD 45
```

```

00059 #define ROTATION_ANGLE_LINEMATE      270
00060 #define ROTATION_ANGLE_MENDIANE        0
00061 #define ROTATION_ANGLE_PHIRAS         270
00062 #define ROTATION_ANGLE_SIBUR           270
00063 #define ROTATION_ANGLE_THYSTAME        270
00064 #define ROTATION_ANGLE_DERAUMERE       90
00065 #define ROTATION_ANGLE_PLAYER          0
00066 #define ROTATION_ANGLE_EGG             270
00067 #define ROTATION_ANGLE_TREE            0
00068 #define ROTATION_ANGLE_LANTERN         0
00069
00070 #define ROTATION_AXIS_FOOD              (Vector3){0, 1, 0}
00071 #define ROTATION_AXIS_LINEMATE          (Vector3){1, 0, 0}
00072 #define ROTATION_AXIS_MENDIANE          (Vector3){1, 0, 0}
00073 #define ROTATION_AXIS_PHIRAS            (Vector3){1, 0, 0}
00074 #define ROTATION_AXIS_SIBUR             (Vector3){1, 0, 0}
00075 #define ROTATION_AXIS_THYSTAME          (Vector3){1, 0, 0}
00076 #define ROTATION_AXIS_DERAUMERE         (Vector3){1, 0, 0}
00077 #define ROTATION_AXIS_PLAYER            (Vector3){0, 1, 0}
00078 #define ROTATION_AXIS_EGG               (Vector3){1, 0, 0}
00079 #define ROTATION_AXIS_TREE              (Vector3){1, 0, 0}
00080 #define ROTATION_AXIS_LANTERN           (Vector3){1, 0, 0}
00081
00082 #define POS_FOOD                        (Vector3){0.5, -0.1, 1.5}
00083 #define POS_LINEMATE                    (Vector3){1, -0.3, -0.5}
00084 #define POS_MENDIANE                    (Vector3){2, -0.25, -0.5}
00085 #define POS_PHIRAS                      (Vector3){0.5, -0.3, -1.5}
00086 #define POS_SIBUR                       (Vector3){1.5, -0.3, -1.5}
00087 #define POS_THYSTAME                    (Vector3){1, 0, -2}
00088 #define POS_DERAUMERE                   (Vector3){2, -0.3, -2}
00089 #define POS_PLAYER                      (Vector3){0, -0.25, 0}
00090 #define POS_EGG                         (Vector3){0.5, 0, 0.5}
00091 #define POS_TREE                        (Vector3){2, -0.3, 2}
00092 #define POS_LANTERN                     (Vector3){1, -0.3, 2}
00093 #define POS_Y_DELIMITATION              -0.27f
00094
00095 #define PLAYER_TEXT_SIZE                40

```

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 SERVER_DOWN_MESSAGE      "THE GUI MUST BE CLOSED FOR UNEXPECTED REASONS"
00011
00012 #define SIZE_TILE                4.7
00013
00014 #define PLAYER_HEIGHT           2
00015
00016 #define DEFAULT_RENDER_DISTANCE 15
00017 #define MAX_RENDER_DISTANCE     20
00018 #define MIN_RENDER_DISTANCE     1
00019
00020 #define HIGH_CAMERA_INCREASE    0.1
00021 #define LOW_CAMERA_INCREASE     0.1
00022
00023 #define PLAYER_SECOND_PERSON_FOV 4.0f
00024 #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 <time.h>
00018
00019 #define TIME_UNIT_MAP_UPDATE 20
00020
00021 namespace Gui {
00022
00023     class Engine;
00024 };
00025
00026 class Gui::Engine {
00027 public:
00028
00029     Engine(std::shared_ptr<INetwork> network);
00030
00031     ~Engine() = default;
00032
00033     void run();
00034
00035 private:
00036
00037     std::unique_ptr<IServerParser> _parser;
00038     std::shared_ptr<INetwork> _network;
00039     std::shared_ptr<Render> _render;
00040     std::unique_ptr<IEvent> _event;
00041     std::shared_ptr<GameData> _gameData;
00042     std::unique_ptr<IGUIUpdater> _guiUpdater;
00043 };

```

```
00066         void listenServer();
00067
00075         void sendMessageUpdate();
00076
00081         void updateMap();
00082
00087         void sendUpdateMapMessage();
00088     };
```

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     };
00066 }
```

```

00065
00070         class GuiGameDataException : public Error {
00071
00072             public:
00073                 GuiGameDataException(std::string message);
00074         };
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
00020     public:
00021
00022         virtual ~IError() = default;
00023
00024         virtual const char *what() const noexcept = 0;
00025 };

```

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
00018     public:
00019
00020         AEvent();
00021
00022         ~AEvent() = default;
00023
00024         virtual void listen() = 0;
00025
00026         void setRender(std::shared_ptr<Render> render);
00027
00028         void setGameData(std::shared_ptr<GameData> gameData);
00029
00030     protected:
00031
00032         std::shared_ptr<Render> _render;
00033         std::shared_ptr<GameData> _gameData;
00034 };

```

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
00064          std::unordered_map<KeyboardKey, std::function<void()>> _eventsKeyPressed =
00065          {
00066              {KEY_THREE, [this]() {switchDisplayDebug();}},
00067              {KEY_F3, [this]() {switchDisplayDebug();}},
00068              {KEY_SPACE, [this]() {setFreeCam();}},
00069              {KEY_R, [this]() {switchTileHudToGame();}},
00070              {KEY_J, [this]() {increaseRenderDistance();}},
00071              {KEY_K, [this]() {decreaseRenderDistance();}},
00072              {KEY_F5, [this]() {changeActualPlayerPov();}},
00073              {KEY_FOUR, [this]() {changeActualPlayerPov();}},
00074              {KEY_KP_ADD, [this]() {increaseTimeUnit();}},
00075              {KEY_KP_SUBTRACT, [this]() {decreaseTimeUnit();}},
00076          };
00077
00082          std::unordered_map<MouseButton, std::function<void()>> _eventsMousePressed =
00083          {
00084              {MOUSE_BUTTON_LEFT, [this]() {handleLeftClick();}},
00085              {MOUSE_BUTTON_RIGHT, [this]() {handleRightClick();}},
00086          };
00087
00092          void moveUpCamera();
00093
00098          void moveDownCamera();
00099
00104          void switchDisplayDebug();
00105
00109          void setFreeCam();
00110
00115          void handleLeftClick();
00116
00120          void handleRightClick();
00121
00126          void selectPlayer();
00127
00131          void selectTile();
00132
00138          void changePlayer(bool turn);
00139
00144          void changeActualPlayerPov();
00145
00150          void switchTileHudToGame();
00151
00156          void increaseRenderDistance();
00157
00162          void decreaseRenderDistance();
00163

```

```

00168         void increaseTimeUnit();
00169
00174         void decreaseTimeUnit();
00175     };

```

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
00024      public:
00025
00026          enum EggState {
00027              IDLE,
00028              DEAD,
00029              BORN,
00030              HATCHING,
00031          };
00032
00043          Egg(size_t id, const std::string &team, std::pair<std::size_t, std::size_t> position);
00044
00050          ~Egg();
00051
00057          std::size_t getId() const;
00058
00064          std::string getTeam() const;
00065

```

```

00071         std::pair<std::size_t, std::size_t> getPosition() const;
00072
00073         void setId(std::size_t id);
00074
00075         void setTeam(const std::string &team);
00076
00077         void setPosition(std::pair<std::size_t, std::size_t> position);
00078
00079         void setState(EggState state);
00080
00081         EggState getState() const;
00082
00083     private:
00084
00085         std::size_t                _id;
00086         std::string                _team;
00087         std::pair<std::size_t, std::size_t> _position;
00088         EggState                   _state;
00089 };

```

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
00024  public:
00025
00026      enum TimeUnitState {
00027          INCREASE,
00028          DECREASE,
00029          NONE
00030      };
00031
00032      GameData();
00033
00034      ~GameData() = default;
00035
00036      std::vector<Gui::Team> &getTeams();
00037
00038      Gui::Team &getTeam(const std::string &name);
00039
00040      void addTeam(const Gui::Team &team);
00041
00042      void addTeam(const std::string &name, const std::string &playerModelPath, const std::string
&eggModelPath, Color playerColor);
00043
00044      void addPlayerToTeam(const std::string &teamName, const Gui::Player &player);
00045
00046      Gui::Player &getPlayer(size_t id);
00047
00048      Map<Gui::Tile> &getMap();
00049
00050      void setMap(const Map<Gui::Tile> &map);
00051
00052      void setMapSize(size_t x, size_t y);
00053
00054      std::pair<size_t, size_t> getMapSize() const;
00055
00056      Gui::Tile &getTile(size_t x, size_t y);
00057
00058      void setTile(const Gui::Tile &tile);
00059
00060      void restartLastTick(void);

```



```

00148
00154     void setServerTick(std::size_t tick);
00155
00161     clock_t getLastTick() const;
00162
00168     std::size_t getServerTick() const;
00169
00175     void setIsEndGame(bool isEndGame);
00176
00183     bool getIsEndGame() const;
00184
00190     void setLastError(const std::string &error);
00191
00197     std::string getLastError() const;
00198
00205     Team &getTeamById(std::size_t id);
00206
00212     TimeUnitState getTimeUnitFromServer() const;
00213
00219     void setTimeUnitFromServer(TimeUnitState timeUnitFromServer);
00220
00226     std::vector<Gui::Egg> &getServerEggs();
00227
00233     void addServerEgg(const Gui::Egg &egg);
00234
00240     void removeServerEgg(size_t id);
00241
00247     void setNbBCTCommandReceived(std::size_t nb);
00248
00254     std::size_t getNbBCTCommandReceived() const;
00255
00260     void restartLastTickMctCommand();
00261
00267     clock_t getLastTickMctCommand() const;
00268
00269     private:
00270
00271         std::vector<Gui::Team>         _teams;
00272         Map<Gui::Tile>                 _map;
00273         std::size_t                     _serverTick;
00274         clock_t                         _lastTick;
00275         bool                           _isEndGame;
00276         std::size_t                     _nbBCTCommandReceived;
00277         clock_t                         _lastTickMctCommand;
00278         std::string                     _lastError;
00279         TimeUnitState                   _timeUnitFromServer;
00280         std::vector<Gui::Egg>           _serverEggs;
00281 };

```

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,
00047                 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 thytsame);
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
00019     class Player;
00020 };
00021
00022 class Gui::Player {
00023
00024     public:
00025
00026         enum PlayerState {
00027             IDLE = 2,
00028             BORN = 8,
00029             BROADCAST = 12,
00030             EJECT = 5,
00031             BEING_EJECTED = 15,
00032             EJECTED = 7,
00033             WALK = 6, // or 10
00034             INCANTATION = 0,
00035             LAY_EGG = 7,
00036             DROP = 9,
00037             COLLECT = 9,
00038             DEAD = 1,
00039         };
00040
00048         Player(std::size_t id, const std::string &team, std::pair<std::size_t, std::size_t> position,
00049             std::size_t orientation, std::size_t level = 1);
00049
00054         ~Player() = default;

```

```

00055
00061     void setPosition(std::pair<std::size_t, std::size_t> position);
00062
00068     void setPosition3D(Vector3 position3D);
00069
00075     void setId(std::size_t id);
00076
00082     void setLevel(std::size_t level);
00083
00089     void setOrientation(std::size_t orientation);
00090
00096     void setTeam(const std::string &team);
00097
00103     std::pair<std::size_t, std::size_t> getPosition(void) const;
00104
00110     Vector3 getPosition3D(void) const;
00111
00117     std::size_t getId(void) const;
00118
00124     std::size_t getLevel(void) const;
00125
00131     std::size_t getOrientation(void) const;
00132
00138     std::string getTeam(void) const;
00139
00145     void setState(PlayerState state);
00146
00152     PlayerState getState(void) const;
00153
00159     void setBroadcast(const std::string &broadcast);
00160
00166     std::string getBroadcast() const;
00167
00173     float getRotationFromOrientation() const;
00174
00180     Vector3 getCenterPosition();
00181
00187     void setCurrentFrame(int currentFrame);
00188
00194     int getCurrentFrame() const;
00195
00200     void restartAnimationTimeEllapsed();
00201
00207     clock_t getAnimationTimeEllapsed() const;
00208
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

```

```

00025     class Team;
00026 };
00027
00028 class Gui::Team {
00029     public:
00030
00031         Team(const std::string &name, const std::string &playerModelPath, const std::string
00040 &eggModelPath, Color playerColor);
00041
00042         ~Team();
00043
00044         const std::string &getName() const;
00045
00046         std::vector<Gui::Player> &getPlayers();
00047
00048         std::vector<Gui::Egg> &getEggs();
00049
00050         void setName(const std::string &name);
00051
00052         void addPlayer(const Gui::Player &player);
00053
00054         void addEgg(const Gui::Egg &egg);
00055
00056         bool removePlayer(std::size_t id);
00057
00058         bool removeEgg(std::size_t id);
00059
00060         std::shared_ptr<Gui::Player> getPlayer(std::size_t id);
00061
00062         Model getPlayerModel() const;
00063
00064         ModelAnimation *getPlayerModelAnimation() const;
00065
00066         void setPlayerModelPath(const std::string &playerModelPath);
00067
00068         std::shared_ptr<Gui::Egg> getEgg(std::size_t id);
00069
00070         Model getEggModel() const;
00071
00072         void setEggModelPath(const std::string &eggModelPath);
00073
00074         std::vector<BoundingBox> getPlayerBoundingBoxes(std::pair<size_t, size_t> pos, size_t
00075 orientation, Vector3 center);
00076
00077         Vector3 getPlayerPositionIn3DSpace(size_t id, Map<Tile> map);
00078
00079         std::vector<RayCollision> getPlayerModelHitbox(size_t id, Camera camera);
00080
00081         bool isPlayerHit(size_t id, Camera camera);
00082
00083         Color getPlayerColor() const;
00084
00085     private:
00086
00087         ModelAnimation*      _modelAnimation;
00088         int                   _animationsCount;
00089         std::string           _name;
00090         std::vector<Gui::Player> _players;
00091         Model                 _playerModel;
00092         std::vector<Gui::Egg>  _eggs;
00093         Model                 _eggModel;
00094         Color                 _playerColor;
00095
00096         BoundingBox rotateBoundingBoxByOrientation(BoundingBox bbox, size_t orientation,
00097 std::pair<size_t, size_t> pos, Vector3 center);
00098 };

```

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
00028  public:
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          {"pdi", [this](std::vector<std::string> data){updatePlayerEggLaying(data);}},
00069          {"pdr", [this](std::vector<std::string> data){updatePlayerRessourceDropping(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
00088      void updateMapSize(const std::vector<std::string> &data);
00089
00095      void updateMapContent(const std::vector<std::string> &data);
00096
00102      void updateTeamNames(const std::vector<std::string> &data);
00103
00109      void updateTeamMember(const std::vector<std::string> &data);
00110
00116      void updatePlayerPosition(const std::vector<std::string> &data);
00117
00123      void updatePlayerLevel(const std::vector<std::string> &data);
00124
00130      void updatePlayerInventory(const std::vector<std::string> &data);
00131
00137      void updatePlayerExpulsion(const std::vector<std::string> &data);
00138
00144      void updatePlayerBroadcast(const std::vector<std::string> &data);
00145
00151      void updatePlayerStartIncantation(const std::vector<std::string> &data);
00152
00158      void updatePlayerEndIncantation(const std::vector<std::string> &data);
00159
00165      void updatePlayerEggLaying(const std::vector<std::string> &data);
00166
00172      void updatePlayerRessourceDropping(const std::vector<std::string> &data);
00173
00179      void updatePlayerRessourceCollecting(const std::vector<std::string> &data);
00180

```

```

00186         void updatePlayerDeath(const std::vector<std::string> &data);
00187
00193         void updateEggLaidByPlayer(const std::vector<std::string> &data);
00194
00200         void updatePlayerBorn(const std::vector<std::string> &data);
00201
00207         void updateEggDeath(const std::vector<std::string> &data);
00208
00214         void updateTimeUnitRequest(const std::vector<std::string> &data);
00215
00221         void updateTimeUnitModification(const std::vector<std::string> &data);
00222
00228         void updateEndOfGame(const std::vector<std::string> &data);
00229
00235         void updateMessageFromServer(const std::vector<std::string> &data);
00236
00242         void updateUnknownMessage(const std::vector<std::string> &data);
00243
00249         void updateCommandParameter(const std::vector<std::string> &data);
00250
00255         void increaseColorIndex();
00256 };

```

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
00019     class IGUIUpdater;
00020 }
00021
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
00030     ~AHud() = default;
00031
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
00059     protected:
00060
00061         TypeScene _typeScene;
00062         std::shared_ptr<GameData> _gameData;
00063         std::shared_ptr<Player> _player;
00064         std::shared_ptr<Tile> _tile;
00065 };

```

6.21 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 };

```

6.22 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    _mendiane;
00060         Texture2D    _phiras;
00061         Texture2D    _sibur;
00062         Texture2D    _thystame;
00063         Texture2D    _playerTexture;
00064 };

```

6.23 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 };

```

6.24 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          };
00038
00043          virtual ~IHud() = default;
00044
00049          virtual void display() = 0;
00050
00056          virtual void setPlayer(std::shared_ptr<Player> player) = 0;
00057
00063          virtual void setTile(std::shared_ptr<Tile> tile) = 0;
00064
00070          virtual TypeScene getType() const = 0;
00071  };

```

6.25 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 const std::string listenServer() = 0;
00083
00089          virtual void sendMessageServer(const std::string& message) = 0;
00090
00091      protected:

```

```

00092         int             _port;
00093         std::string      _hostName;
00094     };

```

6.26 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
00025      public:
00026
00031          virtual ~INetwork() = default;
00032
00039          virtual void setPort(int port) = 0;
00040
00046          virtual void setHostName(const std::string &hostName) = 0;
00047
00053          virtual int getPort() const = 0;
00054
00060          virtual std::string getHostName() const = 0;
00061
00067          virtual void connectToServer() = 0;
00068
00074          virtual const std::string listenServer() = 0;
00075
00081          virtual void sendMessageServer(const std::string &message) = 0;
00082  };

```

6.27 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          const std::string listenServer();
00056

```

```

00062         void sendMessageServer(const std::string& message);
00063
00064     private:
00065
00070         void selectServer();
00071
00077         const std::string readInfoServer();
00078
00079         int             _serverFd;
00080         fd_set          _writeFd;
00081         fd_set          _readFd;
00082         bool            _isConnected;
00083 };

```

6.28 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 };

```

6.29 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 };

```

6.30 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}},
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}},
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::istream& stream, std::vector<std::string>
arguments);
00111
00119      std::vector<std::string> parseString(std::istream& stream, std::vector<std::string>
arguments);
00120
00129      std::vector<std::string> parseMessage(std::istream& stream, std::vector<std::string>
arguments, std::string commandName);
00130
00139      std::vector<std::string> parseListInt(std::istream& stream, std::vector<std::string>
arguments, std::string commandName);
00140  };

```

6.31 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     class Decoration;
00019 }
00020
00021 class Gui::Decoration {
00022 public:
00023     Decoration();
00024     ~Decoration() = default;
00025
00026     void display(std::pair<std::size_t, std::size_t> mapSize, size_t renderDistance,
00027                 std::pair<std::size_t, std::size_t> camPos);
00028
00029     Map<bool> getGenerationItem(std::size_t ratio);
00030
00031 private:
00032     Model _treeModel;
00033     Map<bool> _mapTree;
00034
00035     std::pair<std::size_t, std::size_t> _mapSize;
00036
00037     void displayTree(size_t i, size_t j, Vector3 posTile);
00038 };

```

6.32 Render.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Render
00006 */
00007
00008 #pragma once
00009
00010 #define WINDOW_WIDTH 1920
00011 #define WINDOW_HEIGHT 1080
00012 #define WINDOW_TITLE "Zappy GUI"
00013
00014 #include "raylib.h"
00015 #include "Hud/HudGame.hpp"
00016 #include "Hud/HudTile.hpp"
00017 #include "Hud/HudPlayer.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     class Render;
00027 };
00028
00029 class Gui::Render {
00030 public:
00031     Render(std::shared_ptr<GameData> gameData);
00032 };

```

```

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
00206 private:
00207
00208     UserCamera                _camera;
00209     bool                      _isDebug;
00210     std::shared_ptr<GameData> _gameData;
00211     std::shared_ptr<Decoration> _decoration;
00212     std::vector<std::shared_ptr<Gui::IHud> _hudList;
00213     size_t                    _renderDistance;
00214
00215     Model _tileModel;
00216     Model _foodModel;
00217     Model _linemateModel;
00218     Model _mendianeModel;
00219     Model _phirasModel;
00220     Model _siburModel;
00221     Model _thystameModel;
00222     Model _deraumereModel;
00223     Texture2D _cursorTexture;
00224
00229     void LoadModels();
00230
00235     void displayHUD();
00236
00241     void displayDebug();
00242
00247     void displayPlayers();
00248
00255     void displayPlayerLevel(Team &team, Player &player);
00256
00263     void displayPlayerBroadcast(Team &team, Player &player);
00264
00269     void displayMap();
00270
00275     void displayTile(Tile tile);
00276
00282     void displayEggs(Tile tile) const;
00283
00289     void displayFood(Tile tile) const;
00290
00296     void displayResources(Tile tile) const;
00297

```

```

00303         void displayLinemate(Tile tile) const;
00304
00310         void displayMendiane(Tile tile) const;
00311
00317         void displayPhiras(Tile tile) const;
00318
00324         void displaySibur(Tile tile) const;
00325
00331         void displayThystame(Tile tile) const;
00332
00338         void displayDeraumere(Tile tile) const;
00339
00346         bool displayAnimations(Team &team, Player &player);
00347
00356         ModelAnimation displayWalkAnimation(Team &team, Player &player, ModelAnimation anim);
00357
00362         void displayCursor();
00363
00369         std::pair<std::size_t, std::size_t> getCameraTile();
00370     };

```

6.33 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          size_t 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
00161     private:
00162
00163         std::shared_ptr<Camera>          _camera;
00164         CameraType                       _type;
00165         size_t                           _playerId;
00166         std::pair<std::size_t, std::size_t> _tilePos;
00167     };
```

6.34 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, 126
109 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/INetwork.h
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Colors.hpp, 127
110 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/Network.h
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Config.hpp, 127
111 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/IServerPar
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Engine/Engine.hpp, 126
111 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ParseCom
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/AError.hpp, 128
112 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ServerPar
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp, 129
112 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Decoration
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/IError.hpp, 130
113 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Render.hpp
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/AEvent.hpp, 130
113 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/UserCame
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/Event.hpp, 132
114 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Types.hpp,
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/IEvent.hpp, 133
115 ~AHud
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdate/AGUIUpdater.hpp, 134
121 ~ANetwork
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdate/ANetworkUpdater.hpp, 135
121 ~Decoration
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdate/IDecorator.hpp, 136
123 ~Egg
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EGG.hpp, 137
115 ~Engine
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EngineData.hpp, 138
116 ~Event
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EventData.hpp, 139
117 ~GameData
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/PlayerData, 29
118 ~HudGame
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/HudGame, 43
119 ~HudPlayer
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/HudPlayer, 45
120 ~HudTile
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/AHudTile, 47
123 ~IEvent
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame, 50
124 ~IHud
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudPlayer, 52
124 ~INetwork
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudNetwork, 54
125 ~Inventory
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/IHudInventory, 58
125 ~Network
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/ANetwork, 66

- ~ParseCommandLine
 - Gui::ParseCommandLine, 69
- ~Player
 - Gui::Player, 73
- ~Render
 - Gui::Render, 81
- ~ServerParser
 - Gui::ServerParser, 88
- ~Team
 - Gui::Team, 92
- ~Tile
 - Gui::Tile, 100
- ~UserCamera
 - Gui::UserCamera, 104
- addEgg
 - Gui::Team, 92
- addPlayer
 - Gui::Team, 92
- addPlayerToTeam
 - Gui::GameData, 29
- addResource
 - Gui::Inventory, 58
- addServerEgg
 - Gui::GameData, 29
- addTeam
 - Gui::GameData, 30
- AGUIUpdater
 - Gui::AGUIUpdater, 11
- ANetwork
 - Gui::ANetwork, 15
- changePlayerPOV
 - Gui::Render, 82
- changePOVToFirstPerson
 - Gui::Render, 82
- changePOVToSecondPerson
 - Gui::Render, 82
- changePOVToThirdPerson
 - Gui::Render, 83
- connectToServer
 - Gui::ANetwork, 16
 - Gui::INetwork, 54
 - Gui::Network, 66
- Decoration
 - Gui::Decoration, 18
- display
 - Gui::AHud, 13
 - Gui::Decoration, 19
 - Gui::HudGame, 43
 - Gui::HudPlayer, 45
 - Gui::HudTile, 48
 - Gui::IHud, 52
- displayNbEggs
 - Gui::HudTile, 48
- displayNbPlayers
 - Gui::HudTile, 48
- draw
 - Gui::Render, 83
- Egg
 - Gui::Egg, 20
- Engine
 - Gui::Engine, 23
- Event
 - Gui::Event, 26
- GameData
 - Gui::GameData, 29
- getAnimationTimeEllapsed
 - Gui::Player, 74
- getBroadcast
 - Gui::Player, 74
- getCamera
 - Gui::Render, 83
 - Gui::UserCamera, 104
- getCameraPlayerPov
 - Gui::Render, 83
- getCameraTile
 - Gui::Render, 83
- getCameraType
 - Gui::Render, 84
- getCenterPosition
 - Gui::Player, 74
- getCurrentFrame
 - Gui::Player, 74
- getDeraumere
 - Gui::Inventory, 59
- getEgg
 - Gui::Team, 92
- getEggModel
 - Gui::Team, 93
- getEggs
 - Gui::Team, 93
- getFood
 - Gui::Inventory, 59
- getFovy
 - Gui::UserCamera, 104
- getGenerationItem
 - Gui::Decoration, 19
- getHostName
 - Gui::ANetwork, 16
 - Gui::INetwork, 55
 - Gui::ParseCommandLine, 69
- getId
 - Gui::Egg, 21
 - Gui::Player, 74
- getIsDebug
 - Gui::Render, 84
- getIsEndGame
 - Gui::GameData, 30
- getLastError
 - Gui::GameData, 30
- getLastTick
 - Gui::GameData, 31
- getLastTickMctCommand
 - Gui::GameData, 31

- getLevel
 - Gui::Player, 75
- getLinemate
 - Gui::Inventory, 59
- getMap
 - Gui::GameData, 31
- getMapSize
 - Gui::GameData, 31
- getMendiane
 - Gui::Inventory, 59
- getName
 - Gui::Team, 93
- getNbBCTCommandReceived
 - Gui::GameData, 32
- getOrientation
 - Gui::Player, 75
- getPhiras
 - Gui::Inventory, 60
- getPlayer
 - Gui::GameData, 32
 - Gui::Team, 93
- getPlayerBoundingBoxes
 - Gui::Team, 94
- getPlayerColor
 - Gui::Team, 94
- getPlayerId
 - Gui::UserCamera, 104
- getPlayerModel
 - Gui::Team, 94
- getPlayerModelAnimation
 - Gui::Team, 94
- getPlayerModelHitbox
 - Gui::Team, 95
- getPlayerPositionIn3DSpace
 - Gui::Team, 95
- getPlayers
 - Gui::Team, 95
- getPort
 - Gui::ANetwork, 16
 - Gui::INetwork, 55
 - Gui::ParseCommandLine, 69
- getPosition
 - Gui::Egg, 21
 - Gui::Player, 75
 - Gui::Tile, 100
 - Gui::UserCamera, 105
- getPosition3D
 - Gui::Player, 75
- getPositionIn3DSpace
 - Gui::Tile, 100
- getRenderDistance
 - Gui::Render, 84
- getRessources
 - Gui::Inventory, 60
- getRotationFromOrientation
 - Gui::Player, 76
- getServerEggs
 - Gui::GameData, 32
- getServerTick
 - Gui::GameData, 32
- getSibur
 - Gui::Inventory, 60
- getState
 - Gui::Egg, 21
 - Gui::Player, 76
- getTarget
 - Gui::UserCamera, 105
- getTeam
 - Gui::Egg, 22
 - Gui::GameData, 33
 - Gui::Player, 76
- getTeamById
 - Gui::GameData, 33
- getTeams
 - Gui::GameData, 33
- getThystame
 - Gui::Inventory, 60
- getTile
 - Gui::GameData, 34
- getTileBoundingBoxes
 - Gui::Tile, 101
- getTileModel
 - Gui::Render, 84
- getTileModelHitbox
 - Gui::Tile, 101
- getTilePos
 - Gui::UserCamera, 105
- getTimeUnit
 - Gui::Render, 85
- getTimeUnitFromServer
 - Gui::GameData, 34
- getType
 - Gui::AHud, 13
 - Gui::IHud, 53
 - Gui::UserCamera, 105
- getUp
 - Gui::UserCamera, 106
- Gui::AEvent, 10
- Gui::AGUIUpdater, 10
 - AGUIUpdater, 11
 - update, 11
- Gui::AHud, 12
 - ~AHud, 13
 - display, 13
 - getType, 13
 - setPlayer, 13
 - setTile, 14
- Gui::ANetwork, 14
 - ~ANetwork, 15
 - ANetwork, 15
 - connectToServer, 16
 - getHostName, 16
 - getPort, 16
 - listenServer, 16
 - sendMessageServer, 17
 - setHostName, 17

- setPort, 17
- Gui::Decoration, 18
 - ~Decoration, 18
 - Decoration, 18
 - display, 19
 - getGenerationItem, 19
- Gui::Egg, 20
 - ~Egg, 21
 - Egg, 20
 - getId, 21
 - getPosition, 21
 - getState, 21
 - getTeam, 22
 - setId, 22
 - setPosition, 22
 - setState, 23
 - setTeam, 23
- Gui::Engine, 23
 - ~Engine, 24
 - Engine, 23
 - run, 24
- Gui::Errors::AError, 9
 - what, 10
- Gui::Errors::Error, 24
- Gui::Errors::GuiGameDataException, 37
 - GuiGameDataException, 38
- Gui::Errors::GuiUpdaterException, 40
 - GuiUpdaterException, 41
- Gui::Errors::IError, 48
 - what, 49
- Gui::Errors::NetworkException, 67
 - NetworkException, 68
- Gui::Errors::ParseCommandLineException, 70
 - ParseCommandLineException, 71
- Gui::Errors::ServerParserException, 89
 - ServerParserException, 90
- Gui::Event, 25
 - ~Event, 26
 - Event, 26
 - listen, 27
- Gui::GameData, 27
 - ~GameData, 29
 - addPlayerToTeam, 29
 - addServerEgg, 29
 - addTeam, 30
 - GameData, 29
 - getIsEndGame, 30
 - getLastError, 30
 - getLastTick, 31
 - getLastTickMctCommand, 31
 - getMap, 31
 - getMapSize, 31
 - getNbBCTCommandReceived, 32
 - getPlayer, 32
 - getServerEggs, 32
 - getServerTick, 32
 - getTeam, 33
 - getTeamById, 33
- getTeams, 33
 - getTile, 34
 - getTimeUnitFromServer, 34
 - removeServerEgg, 34
 - restartLastTick, 35
 - restartLastTickMctCommand, 35
 - setIsEndGame, 35
 - setLastError, 35
 - setMap, 36
 - setMapSize, 36
 - setNbBCTCommandReceived, 36
 - setServerTick, 36
 - setTile, 37
 - setTimeUnitFromServer, 37
- Gui::GUIUpdater, 39
 - GUIUpdater, 40
 - update, 40
- Gui::HudGame, 42
 - ~HudGame, 43
 - display, 43
 - HudGame, 43
- Gui::HudPlayer, 44
 - ~HudPlayer, 45
 - display, 45
 - HudPlayer, 45
- Gui::HudTile, 46
 - ~HudTile, 47
 - display, 48
 - displayNbEggs, 48
 - displayNbPlayers, 48
 - HudTile, 47
- Gui::IEvent, 49
 - ~IEvent, 50
 - IEvent, 50
 - listen, 50
 - setGameData, 50
 - setRender, 50
- Gui::IGUIUpdater, 51
 - update, 51
- Gui::IHud, 52
 - ~IHud, 52
 - display, 52
 - getType, 53
 - setPlayer, 53
 - setTile, 53
 - TypeScene, 52
- Gui::INetwork, 54
 - ~INetwork, 54
 - connectToServer, 54
 - getHostName, 55
 - getPort, 55
 - listenServer, 55
 - sendMessageServer, 55
 - setHostName, 56
 - setPort, 56
- Gui::Inventory, 56
 - ~Inventory, 58
 - addResource, 58

- getDeraumere, 59
- getFood, 59
- getLinemate, 59
- getMendiane, 59
- getPhiras, 60
- getRessources, 60
- getSibur, 60
- getThystame, 60
- Inventory, 57, 58
- removeResource, 61
- setDeraumere, 61
- setFood, 61
- setLinemate, 62
- setMendiane, 62
- setPhiras, 62
- setRessources, 62
- setSibur, 63
- setThystame, 63
- Gui::IServerParser, 63
 - parse, 64
- Gui::Network, 64
 - ~Network, 66
 - connectToServer, 66
 - listenServer, 66
 - Network, 66
 - sendMessageServer, 67
- Gui::ParseCommandLine, 68
 - ~ParseCommandLine, 69
 - getHostName, 69
 - getPort, 69
 - ParseCommandLine, 69
 - parseFlags, 70
- Gui::Player, 72
 - ~Player, 73
 - getAnimationTimeEllapsed, 74
 - getBroadcast, 74
 - getCenterPosition, 74
 - getCurrentFrame, 74
 - getId, 74
 - getLevel, 75
 - getOrientation, 75
 - getPosition, 75
 - getPosition3D, 75
 - getRotationFromOrientation, 76
 - getState, 76
 - getTeam, 76
 - inventory, 80
 - Player, 73
 - restartAnimationTimeEllapsed, 76
 - setBroadcast, 77
 - setCurrentFrame, 78
 - setId, 78
 - setLevel, 78
 - setOrientation, 78
 - setPosition, 79
 - setPosition3D, 79
 - setState, 79
 - setTeam, 80
- Gui::Render, 80
 - ~Render, 81
 - changePlayerPOV, 82
 - changePOVToFirstPerson, 82
 - changePOVToSecondPerson, 82
 - changePOVToThirdPerson, 83
 - draw, 83
 - getCamera, 83
 - getCameraPlayerPov, 83
 - getCameraTile, 83
 - getCameraType, 84
 - getIsDebug, 84
 - getRenderDistance, 84
 - getTileModel, 84
 - getTimeUnit, 85
 - isCameraInPlayerPov, 85
 - isOpen, 85
 - Render, 81
 - setCameraPlayerPov, 85
 - setCameraTile, 86
 - setCameraType, 86
 - setIsDebug, 86
 - setPlayerPov, 86
 - setRenderDistance, 87
 - setTimeUnit, 87
- Gui::ServerParser, 87
 - ~ServerParser, 88
 - parse, 88
 - ServerParser, 88
- Gui::Team, 90
 - ~Team, 92
 - addEgg, 92
 - addPlayer, 92
 - getEgg, 92
 - getEggModel, 93
 - getEggs, 93
 - getName, 93
 - getPlayer, 93
 - getPlayerBoundingBoxes, 94
 - getPlayerColor, 94
 - getPlayerModel, 94
 - getPlayerModelAnimation, 94
 - getPlayerModelHitbox, 95
 - getPlayerPositionIn3DSpace, 95
 - getPlayers, 95
 - isPlayerHit, 96
 - removeEgg, 96
 - removePlayer, 96
 - setEggModelPath, 98
 - setName, 98
 - setPlayerModelPath, 98
 - Team, 91
- Gui::Tile, 99
 - ~Tile, 100
 - getPosition, 100
 - getPositionIn3DSpace, 100
 - getTileBoundingBoxes, 101
 - getTileModelHitbox, 101

- inventory, 102
- isTileHit, 101
- setPosition, 102
- Tile, 99, 100
- Gui::UserCamera, 103
 - ~UserCamera, 104
 - getCamera, 104
 - getFovy, 104
 - getPlayerId, 104
 - getPosition, 105
 - getTarget, 105
 - getTilePos, 105
 - getType, 105
 - getUp, 106
 - isPlayerPov, 106
 - setFovy, 106
 - setPlayerId, 107
 - setPosition, 107
 - setTarget, 107
 - setTilePos, 107
 - setType, 108
 - setUp, 108
 - UserCamera, 103
- GuiGameDataException
 - Gui::Errors::GuiGameDataException, 38
- GUIUpdater
 - Gui::GUIUpdater, 40
- GuiUpdaterException
 - Gui::Errors::GuiUpdaterException, 41
- HudGame
 - Gui::HudGame, 43
- HudPlayer
 - Gui::HudPlayer, 45
- HudTile
 - Gui::HudTile, 47
- IEvent
 - Gui::IEvent, 50
- Inventory
 - Gui::Inventory, 57, 58
- inventory
 - Gui::Player, 80
 - Gui::Tile, 102
- isCameraInPlayerPov
 - Gui::Render, 85
- isOpen
 - Gui::Render, 85
- isPlayerHit
 - Gui::Team, 96
- isPlayerPov
 - Gui::UserCamera, 106
- isTileHit
 - Gui::Tile, 101
- listen
 - Gui::Event, 27
 - Gui::IEvent, 50
- listenServer
 - Gui::ANetwork, 16
 - Gui::INetwork, 55
 - Gui::Network, 66
- Network
 - Gui::Network, 66
- NetworkException
 - Gui::Errors::NetworkException, 68
- parse
 - Gui::IServerParser, 64
 - Gui::ServerParser, 88
- ParseCommandLine
 - Gui::ParseCommandLine, 69
- ParseCommandLineException
 - Gui::Errors::ParseCommandLineException, 71
- parseFlags
 - Gui::ParseCommandLine, 70
- Player
 - Gui::Player, 73
- removeEgg
 - Gui::Team, 96
- removePlayer
 - Gui::Team, 96
- removeResource
 - Gui::Inventory, 61
- removeServerEgg
 - Gui::GameData, 34
- Render
 - Gui::Render, 81
- restartAnimationTimeElapsed
 - Gui::Player, 76
- restartLastTick
 - Gui::GameData, 35
- restartLastTickMctCommand
 - Gui::GameData, 35
- run
 - Gui::Engine, 24
- sendMessageServer
 - Gui::ANetwork, 17
 - Gui::INetwork, 55
 - Gui::Network, 67
- ServerParser
 - Gui::ServerParser, 88
- ServerParserException
 - Gui::Errors::ServerParserException, 90
- setBroadcast
 - Gui::Player, 77
- setCameraPlayerPov
 - Gui::Render, 85
- setCameraTile
 - Gui::Render, 86
- setCameraType
 - Gui::Render, 86
- setCurrentFrame
 - Gui::Player, 78
- setDeraumere

- Gui::Inventory, 61
- setEggModelPath
 - Gui::Team, 98
- setFood
 - Gui::Inventory, 61
- setFovy
 - Gui::UserCamera, 106
- setGameData
 - Gui::IEvent, 50
- setHostName
 - Gui::ANetwork, 17
 - Gui::INetwork, 56
- setId
 - Gui::Egg, 22
 - Gui::Player, 78
- setIsDebug
 - Gui::Render, 86
- setIsEndGame
 - Gui::GameData, 35
- setLastError
 - Gui::GameData, 35
- setLevel
 - Gui::Player, 78
- setLinemate
 - Gui::Inventory, 62
- setMap
 - Gui::GameData, 36
- setMapSize
 - Gui::GameData, 36
- setMendiane
 - Gui::Inventory, 62
- setName
 - Gui::Team, 98
- setNbBCTCommandReceived
 - Gui::GameData, 36
- setOrientation
 - Gui::Player, 78
- setPhiras
 - Gui::Inventory, 62
- setPlayer
 - Gui::AHud, 13
 - Gui::IHud, 53
- setPlayerId
 - Gui::UserCamera, 107
- setPlayerModelPath
 - Gui::Team, 98
- setPlayerPov
 - Gui::Render, 86
- setPort
 - Gui::ANetwork, 17
 - Gui::INetwork, 56
- setPosition
 - Gui::Egg, 22
 - Gui::Player, 79
 - Gui::Tile, 102
 - Gui::UserCamera, 107
- setPosition3D
 - Gui::Player, 79
- setRender
 - Gui::IEvent, 50
- setRenderDistance
 - Gui::Render, 87
- setRessources
 - Gui::Inventory, 62
- setServerTick
 - Gui::GameData, 36
- setSibur
 - Gui::Inventory, 63
- setState
 - Gui::Egg, 23
 - Gui::Player, 79
- setTarget
 - Gui::UserCamera, 107
- setTeam
 - Gui::Egg, 23
 - Gui::Player, 80
- setThystame
 - Gui::Inventory, 63
- setTile
 - Gui::AHud, 14
 - Gui::GameData, 37
 - Gui::IHud, 53
- setTilePos
 - Gui::UserCamera, 107
- setTimeUnit
 - Gui::Render, 87
- setTimeUnitFromServer
 - Gui::GameData, 37
- setType
 - Gui::UserCamera, 108
- setUp
 - Gui::UserCamera, 108
- Team
 - Gui::Team, 91
- Tile
 - Gui::Tile, 99, 100
- TypeScene
 - Gui::IHud, 52
- update
 - Gui::AGUIUpdater, 11
 - Gui::GUIUpdater, 40
 - Gui::IGUIUpdater, 51
- UserCamera
 - Gui::UserCamera, 103
- what
 - Gui::Errors::AError, 10
 - Gui::Errors::IError, 49