

Zappy GUI

Generated by Doxygen 1.9.6

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Gui::Errors::AError Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Function Documentation	8
4.1.2.1 what()	8
4.1.3 Member Data Documentation	8
4.1.3.1 _message	8
4.2 Gui::AHud Class Reference	8
4.3 Gui::Decoration Class Reference	8
4.3.1 Constructor & Destructor Documentation	9
4.3.1.1 Decoration()	9
4.3.1.2 ~Decoration()	9
4.3.2 Member Function Documentation	9
4.3.2.1 display()	9
4.3.2.2 getGenerationItem()	9
4.4 Gui::Egg Class Reference	10
4.4.1 Constructor & Destructor Documentation	10
4.4.1.1 Egg()	10
4.4.1.2 ~Egg()	11
4.4.2 Member Function Documentation	11
4.4.2.1 getId()	11
4.4.2.2 getPosition()	11
4.4.2.3 getState()	12
4.4.2.4 getTeam()	12
4.4.2.5 setId()	12
4.4.2.6 setPosition()	12
4.4.2.7 setState()	13
4.4.2.8 setTeam()	13
4.5 Gui::Engine Class Reference	13
4.5.1 Constructor & Destructor Documentation	13
4.5.1.1 Engine()	14
4.5.1.2 ~Engine()	14
4.5.2 Member Function Documentation	14
4.5.2.1 run()	14

4.6 Gui::Errors::Error Class Reference	14
4.6.1 Detailed Description	15
4.7 Gui::Event Class Reference	15
4.7.1 Constructor & Destructor Documentation	15
4.7.1.1 Event()	16
4.7.1.2 ~Event()	16
4.7.2 Member Function Documentation	16
4.7.2.1 listen()	16
4.7.2.2 setGameData()	16
4.7.2.3 setRender()	16
4.8 Gui::GameData Class Reference	17
4.8.1 Constructor & Destructor Documentation	18
4.8.1.1 GameData()	18
4.8.1.2 ~GameData()	18
4.8.2 Member Function Documentation	18
4.8.2.1 addPlayerToTeam()	18
4.8.2.2 addTeam() [1/2]	19
4.8.2.3 addTeam() [2/2]	19
4.8.2.4 getIsEndGame()	19
4.8.2.5 getLastError()	20
4.8.2.6 getLastTick()	20
4.8.2.7 getMap()	20
4.8.2.8 getMapSize()	20
4.8.2.9 getPlayer()	20
4.8.2.10 getServerTick()	21
4.8.2.11 getTeam()	21
4.8.2.12 getTeamById()	21
4.8.2.13 getTeams()	22
4.8.2.14 getTile()	22
4.8.2.15 getTimeUnitFromServer()	22
4.8.2.16 restartLastTick()	23
4.8.2.17 setIsEndGame()	23
4.8.2.18 setLastError()	23
4.8.2.19 setMap()	23
4.8.2.20 setMapSize()	24
4.8.2.21 setServerTick()	24
4.8.2.22 setTile()	24
4.8.2.23 setTimeUnitFromServer()	25
4.9 Gui::Errors::GuiGameDataException Class Reference	25
4.9.1 Detailed Description	26
4.9.2 Constructor & Destructor Documentation	26
4.9.2.1 GuiGameDataException()	26

4.10 Gui::GUIUpdater Class Reference	26
4.10.1 Constructor & Destructor Documentation	26
4.10.1.1 GUIUpdater()	27
4.10.2 Member Function Documentation	27
4.10.2.1 update()	27
4.11 Gui::Errors::GuiUpdaterException Class Reference	27
4.11.1 Detailed Description	28
4.11.2 Constructor & Destructor Documentation	28
4.11.2.1 GuiUpdaterException()	28
4.12 Gui::HudGame Class Reference	29
4.12.1 Constructor & Destructor Documentation	30
4.12.1.1 HudGame()	30
4.12.1.2 ~HudGame()	30
4.12.2 Member Function Documentation	30
4.12.2.1 display()	31
4.13 Gui::HudPlayer Class Reference	31
4.13.1 Constructor & Destructor Documentation	32
4.13.1.1 HudPlayer()	32
4.13.1.2 ~HudPlayer()	32
4.13.2 Member Function Documentation	32
4.13.2.1 display()	33
4.14 Gui::HudTile Class Reference	33
4.14.1 Constructor & Destructor Documentation	34
4.14.1.1 HudTile()	34
4.14.1.2 ~HudTile()	34
4.14.2 Member Function Documentation	34
4.14.2.1 display()	35
4.15 Gui::Errors::IError Class Reference	35
4.15.1 Member Function Documentation	35
4.15.1.1 what()	35
4.16 Gui::IHud Class Reference	36
4.16.1 Member Enumeration Documentation	36
4.16.1.1 TypeScene	36
4.16.2 Constructor & Destructor Documentation	36
4.16.2.1 ~IHud()	36
4.16.3 Member Function Documentation	36
4.16.3.1 display()	37
4.16.3.2 getType()	37
4.16.3.3 setPlayer()	37
4.16.3.4 setTile()	37
4.17 Gui::Inventory Class Reference	38
4.17.1 Constructor & Destructor Documentation	39

4.17.1.1 Inventory() [1/2]	39
4.17.1.2 Inventory() [2/2]	39
4.17.1.3 ~Inventory()	39
4.17.2 Member Function Documentation	39
4.17.2.1 getDeraumere()	40
4.17.2.2 getFood()	40
4.17.2.3 getLinemate()	40
4.17.2.4 getMendiane()	40
4.17.2.5 getPhiras()	41
4.17.2.6 getRessources()	41
4.17.2.7 getSibur()	41
4.17.2.8 getThystame()	41
4.17.2.9 setDeraumere()	41
4.17.2.10 setFood()	42
4.17.2.11 setLinemate()	42
4.17.2.12 setMendiane()	42
4.17.2.13 setPhiras()	43
4.17.2.14 setRessources()	43
4.17.2.15 setSibur()	43
4.17.2.16 setThystame()	43
4.18 Gui::Network Class Reference	44
4.18.1 Constructor & Destructor Documentation	44
4.18.1.1 Network()	44
4.18.1.2 ~Network()	45
4.18.2 Member Function Documentation	45
4.18.2.1 connectToServer()	45
4.18.2.2 getHostName()	45
4.18.2.3 getPort()	45
4.18.2.4 listenServer()	45
4.18.2.5 sendMessageServer()	45
4.18.2.6 setHostName()	46
4.18.2.7 setPort()	46
4.19 Gui::Errors::NetworkException Class Reference	46
4.19.1 Detailed Description	47
4.19.2 Constructor & Destructor Documentation	47
4.19.2.1 NetworkException()	47
4.20 Gui::ParseCommandLine Class Reference	48
4.20.1 Constructor & Destructor Documentation	48
4.20.1.1 ParseCommandLine()	48
4.20.1.2 ~ParseCommandLine()	48
4.20.2 Member Function Documentation	49
4.20.2.1 getHostName()	49

4.20.2.2	getPort()	49
4.20.2.3	parseFlags()	49
4.21	Gui::Errors::ParseCommandLineException Class Reference	50
4.21.1	Detailed Description	50
4.21.2	Constructor & Destructor Documentation	50
4.21.2.1	ParseCommandLineException()	50
4.22	Gui::Player Class Reference	51
4.22.1	Constructor & Destructor Documentation	52
4.22.1.1	Player()	52
4.22.1.2	~Player()	52
4.22.2	Member Function Documentation	53
4.22.2.1	getBroadcast()	53
4.22.2.2	getCenterPosition()	53
4.22.2.3	getId()	53
4.22.2.4	getLevel()	53
4.22.2.5	getOrientation()	54
4.22.2.6	getPosition()	54
4.22.2.7	getRotationFromOrientation()	54
4.22.2.8	getState()	54
4.22.2.9	getTeam()	55
4.22.2.10	setBroadcast()	55
4.22.2.11	setId()	55
4.22.2.12	setLevel()	55
4.22.2.13	setOrientation()	56
4.22.2.14	setPosition()	56
4.22.2.15	setState()	56
4.22.2.16	setTeam()	57
4.22.3	Member Data Documentation	57
4.22.3.1	inventory	57
4.23	Gui::Render Class Reference	57
4.23.1	Constructor & Destructor Documentation	58
4.23.1.1	Render()	58
4.23.1.2	~Render()	58
4.23.2	Member Function Documentation	58
4.23.2.1	draw()	59
4.23.2.2	getCamera()	59
4.23.2.3	getCameraPlayerPov()	59
4.23.2.4	getCameraTile()	59
4.23.2.5	getCameraType()	60
4.23.2.6	getIsDebug()	60
4.23.2.7	getRenderDistance()	60
4.23.2.8	getTileModel()	60

4.23.2.9 getTimeUnit()	60
4.23.2.10 isCameraInPlayerPov()	61
4.23.2.11 isOpen()	61
4.23.2.12 setCameraPlayerPov()	61
4.23.2.13 setCameraTile()	61
4.23.2.14 setCameraType()	63
4.23.2.15 setIsDebug()	63
4.23.2.16 setRenderDistance()	63
4.23.2.17 setTimeUnit()	64
4.24 Gui::ServerParser Class Reference	64
4.24.1 Member Enumeration Documentation	64
4.24.1.1 ParseType	64
4.24.2 Constructor & Destructor Documentation	64
4.24.2.1 ServerParser()	65
4.24.2.2 ~ServerParser()	65
4.24.3 Member Function Documentation	65
4.24.3.1 parse()	65
4.25 Gui::Errors::ServerParserException Class Reference	65
4.25.1 Detailed Description	66
4.25.2 Constructor & Destructor Documentation	66
4.25.2.1 ServerParserException()	66
4.26 Gui::Team Class Reference	67
4.26.1 Constructor & Destructor Documentation	68
4.26.1.1 Team()	68
4.26.1.2 ~Team()	68
4.26.2 Member Function Documentation	68
4.26.2.1 addEgg()	68
4.26.2.2 addPlayer()	69
4.26.2.3 getEgg()	69
4.26.2.4 getEggModel()	69
4.26.2.5 getEggs()	69
4.26.2.6 getName()	70
4.26.2.7 getPlayer()	70
4.26.2.8 getPlayerBoundingBoxes()	70
4.26.2.9 getPlayerModel()	71
4.26.2.10 getPlayerModelHitbox()	71
4.26.2.11 getPlayerPositionIn3DSpace()	71
4.26.2.12 getPlayers()	71
4.26.2.13 isPlayerHit()	72
4.26.2.14 removeEgg()	72
4.26.2.15 removePlayer()	72
4.26.2.16 setEggModelPath()	74

4.26.2.17 setName()	74
4.26.2.18 setPlayerModelPath()	74
4.27 Gui::Tile Class Reference	75
4.27.1 Constructor & Destructor Documentation	75
4.27.1.1 Tile() [1/2]	75
4.27.1.2 Tile() [2/2]	76
4.27.1.3 ~Tile()	76
4.27.2 Member Function Documentation	76
4.27.2.1 getPosition()	76
4.27.2.2 getPositionIn3DSpace()	77
4.27.2.3 getTileBoundingBoxes()	77
4.27.2.4 getTileModelHitbox()	77
4.27.2.5 isTileHit()	78
4.27.2.6 setPosition()	78
4.27.3 Member Data Documentation	78
4.27.3.1 inventory	78
4.28 Gui::UserCamera Class Reference	79
4.28.1 Constructor & Destructor Documentation	79
4.28.1.1 UserCamera()	80
4.28.1.2 ~UserCamera()	80
4.28.2 Member Function Documentation	80
4.28.2.1 getCamera()	80
4.28.2.2 getFovy()	80
4.28.2.3 getPlayerId()	81
4.28.2.4 getPosition()	81
4.28.2.5 getTarget()	81
4.28.2.6 getTilePos()	81
4.28.2.7 getType()	82
4.28.2.8 getUp()	82
4.28.2.9 isPlayerPov()	82
4.28.2.10 setFovy()	82
4.28.2.11 setPlayerId()	83
4.28.2.12 setPosition()	83
4.28.2.13 setTarget()	83
4.28.2.14 setTilePos()	84
4.28.2.15 setType()	84
4.28.2.16 setUp()	84
5 File Documentation	85
5.1 Assets.hpp	85
5.2 Colors.hpp	86
5.3 Config.hpp	86

5.4 Engine.hpp	86
5.5 AError.hpp	87
5.6 Error.hpp	87
5.7 IError.hpp	88
5.8 Event.hpp	88
5.9 Egg.hpp	90
5.10 GameData.hpp	90
5.11 Inventory.hpp	92
5.12 Player.hpp	92
5.13 Team.hpp	94
5.14 Tile.hpp	95
5.15 GUIUpdater.hpp	95
5.16 AHud.hpp	97
5.17 HudGame.hpp	97
5.18 HudPlayer.hpp	98
5.19 HudTile.hpp	98
5.20 IHud.hpp	98
5.21 Network.hpp	99
5.22 ParseCommandLine.hpp	100
5.23 ServerParser.hpp	100
5.24 Decoration.hpp	101
5.25 Render.hpp	102
5.26 UserCamera.hpp	103
5.27 Types.hpp	104
Index	107

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Gui::Decoration	8
Gui::Egg	10
Gui::Engine	13
Gui::Event	15
std::exception	
Gui::Errors::IError	35
Gui::Errors::AError	7
Gui::Errors::Error	14
Gui::Errors::GuiGameDataException	25
Gui::Errors::GuiUpdaterException	27
Gui::Errors::NetworkException	46
Gui::Errors::ParseCommandLineException	50
Gui::Errors::ServerParserException	65
Gui::GameData	17
Gui::GUIUpdater	26
Gui::IHud	36
Gui::AHud	8
Gui::HudGame	29
Gui::HudPlayer	31
Gui::HudTile	33
Gui::Inventory	38
Gui::Network	44
Gui::ParseCommandLine	48
Gui::Player	51
Gui::Render	57
Gui::ServerParser	64
Gui::Team	67
Gui::Tile	75
Gui::UserCamera	79

Chapter 2

Class Index

2.1 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	7
Gui::AHud	8
Gui::Decoration	8
Gui::Egg	10
Gui::Engine	13
Gui::Errors::Error	
Base class for argument-related errors	14
Gui::Event	15
Gui::GameData	17
Gui::Errors::GuiGameDataException	
Error class for GameData errors	25
Gui::GUIUpdater	26
Gui::Errors::GuiUpdaterException	
Error class for GUIUpdater errors	27
Gui::HudGame	29
Gui::HudPlayer	31
Gui::HudTile	33
Gui::Errors::IError	35
Gui::IHud	36
Gui::Inventory	38
Gui::Network	44
Gui::Errors::NetworkException	
Error class for network errors	46
Gui::ParseCommandLine	48
Gui::Errors::ParseCommandLineException	
Error class for parseCommandLine errors	50
Gui::Player	51
Gui::Render	57
Gui::ServerParser	64
Gui::Errors::ServerParserException	
Error class for network errors	65
Gui::Team	67
Gui::Tile	75
Gui::UserCamera	79

Chapter 3

File Index

3.1 File List

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

/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Assets.hpp	85
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Colors.hpp	86
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Config.hpp	86
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Types.hpp	104
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Engine/Engine.hpp	86
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/AError.hpp	87
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp	87
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/IError.hpp	88
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/Event.hpp	88
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Egg.hpp	90
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/GameData.hpp	90
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Inventory.hpp	92
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Player.hpp	92
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Team.hpp	94
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Tile.hpp	95
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/GUIUpdater.hpp	95
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/AHud.hpp	97
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp	97
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudPlayer.hpp	98
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudTile.hpp	98
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/IHud.hpp	98
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/Network.hpp	99
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ParseCommandLine.hpp	100
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ServerParser.hpp	100
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Decoration.hpp	101
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Render.hpp	102
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/UserCamera.hpp	103

Chapter 4

Class Documentation

4.1 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`

4.1.1 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.

4.1.2 Member Function Documentation

4.1.2.1 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](#).

4.1.3 Member Data Documentation

4.1.3.1 _message

```
std::string Gui::Errors::AError::_message [protected]
```

The error message.

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

4.2 Gui::AHud Class Reference

Inheritance diagram for Gui::AHud:

4.3 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.

4.3.1 Constructor & Destructor Documentation

4.3.1.1 Decoration()

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

Construct a new [Decoration](#) object.

4.3.1.2 ~Decoration()

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

Destroy the [Decoration](#) object.

4.3.2 Member Function Documentation

4.3.2.1 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.

4.3.2.2 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

4.4 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.

4.4.1 Constructor & Destructor Documentation

4.4.1.1 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.

4.4.1.2 ~Egg()

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

Destroy the [Egg](#) object.

Note

The destructor starts the egg animation if implemented.

4.4.2 Member Function Documentation**4.4.2.1 getId()**

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

Get the Id object.

Returns

std::size_t Id of the egg.

4.4.2.2 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.

4.4.2.3 getState()

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

Get the state object.

Returns

EggState State of the egg.

4.4.2.4 getTeam()

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

Get the [Team](#) object.

Returns

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

4.4.2.5 setId()

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

Set the id object.

Parameters

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

4.4.2.6 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.
-----------------	----------------------

4.4.2.7 setState()

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

Set the state object.

Parameters

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

4.4.2.8 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

4.5 Gui::Engine Class Reference

Public Member Functions

- [Engine](#) (std::shared_ptr< [Network](#) > network)
Construct a new [Engine](#) object.
- [~Engine](#) ()=default
Destroy the [Engine](#) object.
- void [run](#) (void)
Run the engine loop.

4.5.1 Constructor & Destructor Documentation

4.5.1.1 Engine()

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

Construct a new [Engine](#) object.

Parameters

<i>network</i>	Network class.
----------------	--------------------------------

4.5.1.2 ~Engine()

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

Destroy the [Engine](#) object.

4.5.2 Member Function Documentation

4.5.2.1 run()

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

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

4.6 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](#)

4.6.1 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

4.7 Gui::Event Class Reference

Public Member Functions

- [Event](#) ()
Construct a new [Event](#) object.
- [~Event](#) ()=default
Destroy the [Event](#) object.
- void [listen](#) ()
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.

4.7.1 Constructor & Destructor Documentation

4.7.1.1 Event()

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

Construct a new [Event](#) object.

4.7.1.2 ~Event()

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

Destroy the [Event](#) object.

4.7.2 Member Function Documentation

4.7.2.1 listen()

```
void Gui::Event::listen ( )
```

Listen the user's events.

4.7.2.2 setGameData()

```
void Gui::Event::setGameData (
    std::shared_ptr< GameData > gameData )
```

Set the [GameData](#) object.

Parameters

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

4.7.2.3 setRender()

```
void Gui::Event::setRender (
    std::shared_ptr< Render > render )
```

Set the [Render](#) object.

Parameters

<i>render</i>	Render class.
---------------	-------------------------------

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

4.8 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)
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.

4.8.1 Constructor & Destructor Documentation

4.8.1.1 [GameData\(\)](#)

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

Construct a new [GameData](#) object.

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

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

Destroy the [GameData](#) object.

4.8.2 Member Function Documentation

4.8.2.1 [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.

4.8.2.2 addTeam() [1/2]

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

Add a team to the game.

Parameters

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

4.8.2.3 addTeam() [2/2]

```
void Gui::GameData::addTeam (
    const std::string & name,
    const std::string & playerModelPath,
    const std::string & eggModelPath )
```

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.

4.8.2.4 getIsEndGame()

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

Get the IsEnd Game object.

Returns

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

4.8.2.5 getLastError()

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

Get the Last Error object.

Returns

std::string - Last error message.

4.8.2.6 getLastTick()

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

Get the Last Tick object.

Returns

clock_t - Last Tick.

4.8.2.7 getMap()

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

Get the Map object.

Returns

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

4.8.2.8 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.

4.8.2.9 getPlayer()

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

Get a player object.

Parameters

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

4.8.2.10 getServerTick()

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

Get the Server Tick object.

Returns

std::size_t - Server Tick.

4.8.2.11 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.

4.8.2.12 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.

4.8.2.13 getTeams()

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

Get the Teams object.

Returns

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

4.8.2.14 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.

4.8.2.15 getTimeUnitFromServer()

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

Get the Time Unit From Server object.

Returns

true - The time unit has changed.

4.8.2.16 restartLastTick()

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

Restart the last tick clock.

4.8.2.17 setIsEndGame()

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

Set the IsEnd Game object.

Parameters

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

4.8.2.18 setLastError()

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

Set the Last Error object.

Parameters

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

4.8.2.19 setMap()

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

Set the Map object.

Parameters

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

4.8.2.20 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.

4.8.2.21 setServerTick()

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

Set the Server Tick object.

Parameters

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

4.8.2.22 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.

4.8.2.23 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

4.9 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`

4.9.1 Detailed Description

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

4.9.2 Constructor & Destructor Documentation

4.9.2.1 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`

4.10 Gui::GUIUpdater Class Reference

Public Member Functions

- [GUIUpdater](#) (`std::shared_ptr< GameData > gameData, std::shared_ptr< Network > 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](#).

4.10.1 Constructor & Destructor Documentation

4.10.1.1 GUIUpdater()

```
Gui::GuiUpdater::GuiUpdater (
    std::shared_ptr< GameData > gameData,
    std::shared_ptr< Network > 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.

4.10.2 Member Function Documentation

4.10.2.1 update()

```
void Gui::GuiUpdater::update (
    const std::string & command,
    const std::vector< std::string > & data )
```

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 .

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

4.11 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](#)

4.11.1 Detailed Description

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

4.11.2 Constructor & Destructor Documentation

4.11.2.1 [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

4.12 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
- std::shared_ptr< [GameData](#) > _gameData
- std::shared_ptr< [Player](#) > _player
- std::shared_ptr< [Tile](#) > _tile

4.12.1 Constructor & Destructor Documentation

4.12.1.1 HudGame()

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

Construct a new Hud Game object.

Parameters

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

4.12.1.2 ~HudGame()

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

Destroy the Hud Game object.

4.12.2 Member Function Documentation

4.12.2.1 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

4.13 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](#)
- std::shared_ptr< [GameData](#) > [_gameData](#)
- std::shared_ptr< [Player](#) > [_player](#)
- std::shared_ptr< [Tile](#) > [_tile](#)

4.13.1 Constructor & Destructor Documentation

4.13.1.1 HudPlayer()

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

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

Parameters

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

4.13.1.2 ~HudPlayer()

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

Destroy the Hud [Player](#) object.

4.13.2 Member Function Documentation

4.13.2.1 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

4.14 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.

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
- std::shared_ptr< [GameData](#) > _gameData
- std::shared_ptr< [Player](#) > _player
- std::shared_ptr< [Tile](#) > _tile

4.14.1 Constructor & Destructor Documentation

4.14.1.1 HudTile()

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

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

Parameters

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

4.14.1.2 ~HudTile()

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

Destroy the Hud [Tile](#) object.

4.14.2 Member Function Documentation

4.14.2.1 display()

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

Display [Tile](#) Hud.

Implements [Gui::AHud](#).

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

4.15 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.

4.15.1 Member Function Documentation

4.15.1.1 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

4.16 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.

4.16.1 Member Enumeration Documentation

4.16.1.1 TypeScene

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

Hud enum for the different scenes.

4.16.2 Constructor & Destructor Documentation

4.16.2.1 ~IHud()

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

Destroy the [IHud](#) object.

4.16.3 Member Function Documentation

4.16.3.1 display()

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

Display the Hud.

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

4.16.3.2 getType()

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

Get the Type object.

Returns

TypeScene - Type of the scene.

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

4.16.3.3 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](#).

4.16.3.4 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

4.17 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 thystsame)
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.

4.17.1 Constructor & Destructor Documentation

4.17.1.1 Inventory() [1/2]

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

Construct a new [Inventory](#) object.

4.17.1.2 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.

4.17.1.3 ~Inventory()

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

Destroy the [Inventory](#) object.

4.17.2 Member Function Documentation

4.17.2.1 getDeraumere()

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

Get the Deraumere object.

Returns

std::size_t - deraumere

4.17.2.2 getFood()

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

Get the Food object.

Returns

std::size_t - food

4.17.2.3 getLinemate()

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

Get the Linemate object.

Returns

std::size_t - linemate

4.17.2.4 getMendiane()

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

Get the Mendiane object.

Returns

std::size_t - mendiane

4.17.2.5 getPhiras()

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

Get the Phiras object.

Returns

std::size_t - phiras

4.17.2.6 getRessources()

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

Get the Ressources object.

Returns

Ressources - ressources

4.17.2.7 getSibur()

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

Get the Sibur object.

Returns

std::size_t - sibur

4.17.2.8 getThystame()

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

Get the Thystame object.

Returns

std::size_t - thystame

4.17.2.9 setDeraumere()

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

Set the Deraumere object.

Parameters

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

4.17.2.10 setFood()

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

Set the Food object.

Parameters

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

4.17.2.11 setLinemate()

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

Set the Linemate object.

Parameters

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

4.17.2.12 setMendiane()

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

Set the Mendiane object.

Parameters

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

4.17.2.13 setPhiras()

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

Set the Phiras object.

Parameters

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

4.17.2.14 setRessources()

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

Set the Ressources object.

Parameters

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

4.17.2.15 setSibur()

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

Set the Sibur object.

Parameters

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

4.17.2.16 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

4.18 Gui::Network Class Reference

Public Member Functions

- [Network](#) (int port, const std::string &hostName)
Construct a new [Network](#) object.
- [~Network](#) ()=default
Destroy the [Network](#) object.
- void [setPort](#) (int port)
Set the port object.
- void [setHostName](#) (const std::string &hostName)
Set the host name object.
- int [getPort](#) () const
Get the port object.
- std::string [getHostName](#) () const
Get the host name 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.

4.18.1 Constructor & Destructor Documentation

4.18.1.1 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.

4.18.1.2 ~Network()

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

Destroy the [Network](#) object.

4.18.2 Member Function Documentation

4.18.2.1 connectToServer()

```
void Gui::Network::connectToServer ( )
```

Connect the Gui network with the server.

4.18.2.2 getHostName()

```
std::string Gui::Network::getHostName ( ) const
```

Get the host name object.

Returns

const std::string - Host name of the server.

4.18.2.3 getPort()

```
int Gui::Network::getPort ( ) const
```

Get the port object.

Returns

const int - Port of the server.

4.18.2.4 listenServer()

```
const std::string Gui::Network::listenServer ( )
```

Listen the server and return it message.

Returns

std::string - Message of the server.

4.18.2.5 sendMessageServer()

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

Send a message to the Server.

Parameters

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

4.18.2.6 setHostName()

```
void Gui::Network::setHostName (
    const std::string & hostName )
```

Set the host name object.

Parameters

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

4.18.2.7 setPort()

```
void Gui::Network::setPort (
    int port )
```

Set the port object.

Parameters

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

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

4.19 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](#)

4.19.1 Detailed Description

[Error](#) class for network errors.

4.19.2 Constructor & Destructor Documentation

4.19.2.1 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

4.20 Gui::ParseCommandLine Class Reference

Public Member Functions

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

4.20.1 Constructor & Destructor Documentation

4.20.1.1 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.

4.20.1.2 ~ParseCommandLine()

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

Destroy the Parse Command Line object.

4.20.2 Member Function Documentation

4.20.2.1 getHostName()

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

Get the hostName object.

Returns

std::string - hostName

4.20.2.2 getPort()

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

Get the port object.

Returns

int - port

4.20.2.3 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

4.21 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](#)

4.21.1 Detailed Description

[Error](#) class for parseCommandLine errors.

4.21.2 Constructor & Destructor Documentation

4.21.2.1 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

4.22 Gui::Player Class Reference

Collaboration diagram for Gui::Player:

Public Types

- enum **PlayerState** {
IDLE , **BORN** , **BROADCAST** , **EJECT** ,
BEING_EJECTED , **WALK** , **INCANTATION** , **FINISHED_INCANTATION** ,
LAY_EGG , **DROP** , **COLLECT** , **DEAD** }

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 **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.
- 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.

Public Attributes

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

4.22.1 Constructor & Destructor Documentation

4.22.1.1 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 .

4.22.1.2 ~Player()

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

Destroy the [Player](#) object.

4.22.2 Member Function Documentation

4.22.2.1 getBroadcast()

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

Get the Broadcast object.

Returns

std::string - Broadcast message.

4.22.2.2 getCenterPosition()

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

Get the Center Position object.

Returns

Vector3 - Center position.

4.22.2.3 getId()

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

Get the Id object.

Returns

std::size_t - id

4.22.2.4 getLevel()

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

Get the Level object.

Returns

std::size_t - level

4.22.2.5 getOrientation()

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

Get the Orientation object.

Returns

std::size_t - orientation

4.22.2.6 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

4.22.2.7 getRotationFromOrientation()

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

Get the Vector From Orientation object.

Returns

Vector3 - Vector3 from orientation.

4.22.2.8 getState()

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

Get the State object.

Returns

PlayerState - [Player](#) state.

4.22.2.9 getTeam()

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

Get the [Team](#) object.

Returns

std::string - team name

4.22.2.10 setBroadcast()

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

Set the Broadcast object.

Parameters

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

4.22.2.11 setId()

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

Set the Id object.

Parameters

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

4.22.2.12 setLevel()

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

Set the Level object.

Parameters

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

4.22.2.13 setOrientation()

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

Set the Orientation object.

Parameters

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

4.22.2.14 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
-----------------	------------------------

4.22.2.15 setState()

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

Set the State object.

Parameters

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

4.22.2.16 setTeam()

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

Set the [Team](#) object.

Parameters

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

4.22.3 Member Data Documentation

4.22.3.1 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

4.23 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.
- size_t [getTimeUnit](#) () const

Get the Time Unit value.
- void [setTimeUnit](#) (size_t timeUnit)

Set the Time Unit value.

4.23.1 Constructor & Destructor Documentation

4.23.1.1 Render()

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

Construct a new [Render](#) object.

4.23.1.2 ~Render()

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

Destroy the [Render](#) object.

4.23.2 Member Function Documentation

4.23.2.1 draw()

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

Draw the scene.

4.23.2.2 getCamera()

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

Get the Camera object.

Returns

std::shared_ptr<Camera> - camera

4.23.2.3 getCameraPlayerPov()

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

Get the Camera player pov id.

Returns

std::size_t - Id of the player.

4.23.2.4 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.

4.23.2.5 getCameraType()

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

Get the Type object.

Returns

CameraType - Camera type.

4.23.2.6 getIsDebug()

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

Get the Is Debug object.

Returns

true - diplay debug

false - do not display debug

4.23.2.7 getRenderDistance()

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

Get the [Render](#) Distance value.

4.23.2.8 getTileModel()

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

Get the [Tile](#) model.

4.23.2.9 getTimeUnit()

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

Get the Time Unit value.

Returns

size_t - Time unit value.

4.23.2.10 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.

4.23.2.11 isOpen()

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

Check if the window is open.

Returns

true - the window is open
false - the window is closed

4.23.2.12 setCameraPlayerPov()

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

Set the Camera player pov id.

Parameters

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

4.23.2.13 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.
------------	----------------

4.23.2.14 setCameraType()

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

Set the Type object.

Parameters

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

4.23.2.15 setIsDebug()

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

Set the Is Debug object.

Parameters

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

4.23.2.16 setRenderDistance()

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

Set the [Render](#) Distance value.

Parameters

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

4.23.2.17 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

4.24 Gui::ServerParser Class Reference

Public Types

- enum [ParseType](#) { INT , STRING , MESSAGE , LIST_INT }
Enum of types to parse.

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.

4.24.1 Member Enumeration Documentation

4.24.1.1 ParseType

```
enum Gui::ServerParser::ParseType
```

Enum of types to parse.

4.24.2 Constructor & Destructor Documentation

4.24.2.1 ServerParser()

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

Construct a new Server Parser object.

4.24.2.2 ~ServerParser()

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

Destroy the Server Parser object.

4.24.3 Member Function Documentation

4.24.3.1 parse()

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

Parse the command server.

Parameters

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

Returns

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

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

4.25 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](#)

4.25.1 Detailed Description

[Error](#) class for network errors.

4.25.2 Constructor & Destructor Documentation

4.25.2.1 [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

4.26 Gui::Team Class Reference

Public Member Functions

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

4.26.1 Constructor & Destructor Documentation

4.26.1.1 Team()

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

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.

4.26.1.2 ~Team()

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

Destroy the [Team](#) object.

4.26.2 Member Function Documentation

4.26.2.1 addEgg()

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

Add an egg to the team.

Parameters

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

4.26.2.2 addPlayer()

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

Add a player to the team.

Parameters

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

4.26.2.3 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](#).

4.26.2.4 getEggModel()

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

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

Returns

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

4.26.2.5 getEggs()

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

Get the Eggs object.

Returns

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

4.26.2.6 getName()

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

Get the Name object.

Returns

const std::string& Name of the team.

4.26.2.7 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](#).

4.26.2.8 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.

4.26.2.9 getPlayerModel()

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

Get the Model object.

Returns

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

4.26.2.10 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.

4.26.2.11 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.

4.26.2.12 getPlayers()

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

Get the Players object.

Returns

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

4.26.2.13 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.

4.26.2.14 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.

4.26.2.15 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.

4.26.2.16 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.
--------------------	-------------------------------------

4.26.2.17 setName()

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

Set the Name object.

Parameters

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

4.26.2.18 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

4.27 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.

4.27.1 Constructor & Destructor Documentation

4.27.1.1 [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.
-----------------	-------------------------------

4.27.1.2 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.

4.27.1.3 ~Tile()

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

Destroy the [Tile](#) object.

4.27.2 Member Function Documentation**4.27.2.1 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

4.27.2.2 getPositionIn3DSpace()

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

Get the Position In Space object.

Returns

Vector3 - Position in space.

4.27.2.3 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.

4.27.2.4 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.

4.27.2.5 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.

4.27.2.6 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.
-----------------	---------------------------

4.27.3 Member Data Documentation

4.27.3.1 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

4.28 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.

4.28.1 Constructor & Destructor Documentation

4.28.1.1 UserCamera()

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

Construct a new User Camera object.

4.28.1.2 ~UserCamera()

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

Destroy the User Camera object.

4.28.2 Member Function Documentation

4.28.2.1 getCamera()

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

Get the Camera object.

Returns

Camera - camera

4.28.2.2 getFovy()

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

Get the Fovy object.

Returns

float - fovy

4.28.2.3 getPlayerId()

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

Get the [Player](#) Id object.

Returns

size_t - [Player](#) id.

4.28.2.4 getPosition()

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

Get the Position object.

Returns

Vector3 - position

4.28.2.5 getTarget()

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

Get the Target object.

Returns

Vector3 - target

4.28.2.6 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.

4.28.2.7 getType()

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

Get the Type object.

Returns

CameraType - Camera type.

4.28.2.8 getUp()

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

Get the Up object.

Returns

Vector3 - up

4.28.2.9 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.

4.28.2.10 setFovy()

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

Set the Fovy object.

Parameters

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

4.28.2.11 setPlayerId()

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

Set the [Player](#) Id object.

Parameters

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

4.28.2.12 setPosition()

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

Set the Position object.

Parameters

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

4.28.2.13 setTarget()

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

Set the Target object.

Parameters

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

4.28.2.14 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.
------------	-----------------------

4.28.2.15 setType()

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

Set the Type object.

Parameters

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

4.28.2.16 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 5

File Documentation

5.1 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  #define PNG_HUD            PATH_ASSETS PATH_HUD "hud.png"
00020  #define FONT_HUD           PATH_ASSETS PATH_HUD "SimplyMono-Bold.ttf"
00021
00022  #define MODEL_TILE         PATH_ASSETS PATH_TILE "tile.glb"
00023  #define MODEL_FOOD         PATH_ASSETS PATH_RESOURCES "food.glb"
00024  #define MODEL_LINEMATE     PATH_ASSETS PATH_RESOURCES "linemate.glb"
00025  #define MODEL_MENDIANE     PATH_ASSETS PATH_RESOURCES "mendiane.glb"
00026  #define MODEL_PHIRAS       PATH_ASSETS PATH_RESOURCES "phiras.glb"
00027  #define MODEL_SIBUR        PATH_ASSETS PATH_RESOURCES "sibur.glb"
00028  #define MODEL_THYSTAME     PATH_ASSETS PATH_RESOURCES "thystame.glb"
00029  #define MODEL_DERAUMERE    PATH_ASSETS PATH_RESOURCES "deraumere.glb"
00030  #define MODEL_PLAYER       PATH_ASSETS PATH_PLAYER "player.glb"
00031  #define MODEL_EGG          PATH_ASSETS PATH_PLAYER "egg.glb"
00032  #define MODEL_TREE         PATH_ASSETS PATH_DECORATION "tree.glb"
00033  #define MODEL_LANTERN      PATH_ASSETS PATH_DECORATION "lantern.glb"
00034
00035  #define SCALE_FOOD          (Vector3){1, 0.5, 1}
00036  #define SCALE_LINEMATE     (Vector3){0.1, 0.1, 0.1}
00037  #define SCALE_MENDIANE     (Vector3){0.1, 0.1, 0.1}
00038  #define SCALE_PHIRAS       (Vector3){0.001, 0.001, 0.005}
00039  #define SCALE_SIBUR        (Vector3){0.01, 0.01, 0.01}
00040  #define SCALE_THYSTAME     (Vector3){2, 2, 2}
00041  #define SCALE_DERAUMERE    (Vector3){0.5, 0.5, 0.5}
00042  #define SCALE_PLAYER       (Vector3){0.5, 0.5, 0.5}
00043  #define SCALE_EGG          (Vector3){0.5, 0.5, 0.5}
00044  #define SCALE_TREE         (Vector3){1, 1, 1}
00045  #define SCALE_LANTERN      (Vector3){1, 1, 1}
00046
00047  #define ROTATION_ANGLE_FOOD 45
00048  #define ROTATION_ANGLE_LINEMATE 270
00049  #define ROTATION_ANGLE_MENDIANE 0
00050  #define ROTATION_ANGLE_PHIRAS 270
00051  #define ROTATION_ANGLE_SIBUR 270
00052  #define ROTATION_ANGLE_THYSTAME 0
00053  #define ROTATION_ANGLE_DERAUMERE 0
00054  #define ROTATION_ANGLE_PLAYER 0
00055  #define ROTATION_ANGLE_EGG 270
00056  #define ROTATION_ANGLE_TREE 0
00057  #define ROTATION_ANGLE_LANTERN 0
00058
```

```

00059 #define ROTATION_AXIS_FOOD          (Vector3){0, 1, 0}
00060 #define ROTATION_AXIS_LINEMATE       (Vector3){1, 0, 0}
00061 #define ROTATION_AXIS_MENDIANE       (Vector3){1, 0, 0}
00062 #define ROTATION_AXIS_PHIRAS         (Vector3){1, 0, 0}
00063 #define ROTATION_AXIS_SIBUR          (Vector3){1, 0, 0}
00064 #define ROTATION_AXIS_THYSTAME       (Vector3){0, 0, 0}
00065 #define ROTATION_AXIS_DERAUMERE      (Vector3){0, 0, 0}
00066 #define ROTATION_AXIS_PLAYER         (Vector3){0, 1, 0}
00067 #define ROTATION_AXIS_EGG            (Vector3){1, 0, 0}
00068 #define ROTATION_AXIS_TREE           (Vector3){1, 0, 0}
00069 #define ROTATION_AXIS_LANTERN        (Vector3){1, 0, 0}
00070
00071 #define POS_FOOD                     (Vector3){0.5, -0.1, 1.5}
00072 #define POS_LINEMATE                 (Vector3){1, -0.3, -0.5}
00073 #define POS_MENDIANE                 (Vector3){2, -0.25, -0.5}
00074 #define POS_PHIRAS                   (Vector3){0.5, -0.3, -1.5}
00075 #define POS_SIBUR                    (Vector3){1.5, -0.3, -1.5}
00076 #define POS_THYSTAME                 (Vector3){1, -0.2, -2}
00077 #define POS_DERAUMERE                (Vector3){2, -0.3, -2}
00078 #define POS_PLAYER                   (Vector3){0, -0.25, 0}
00079 #define POS_EGG                      (Vector3){0.5, 0, 0.5}
00080 #define POS_TREE                     (Vector3){2, -0.3, 2}
00081 #define POS_LANTERN                  (Vector3){1, -0.3, 2}

```

5.2 Colors.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** Colors
00006 */
00007
00008 #pragma once
00009
00010 #define STR_BLUE "\033[0;34m"
00011 #define STR_GREEN "\033[0;32m"
00012 #define STR_RED "\033[0;31m"
00013 #define STR_YELLOW "\033[0;33m"
00014 #define STR_VIOLET "\033[0;35m"
00015 #define STR_CYAN "\033[0;36m"
00016 #define STR_RESET "\033[0m"

```

5.3 Config.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** Config
00006 */
00007
00008 #pragma once
00009
00010 #define SIZE_TILE 4.7
00011
00012 #define PLAYER_HEIGHT 2
00013
00014 #define DEFAULT_RENDER_DISTANCE 5
00015 #define MAX_RENDER_DISTANCE 15
00016 #define MIN_RENDER_DISTANCE 1
00017
00018 #define HIGH_CAMERA_INCREASE 0.1
00019 #define LOW_CAMERA_INCREASE 0.1
00020
00021 #define PLAYER_SECOND_PERSON_FOV 4.0f
00022 #define PLAYER_THIRD_PERSON_FOV 5.0f

```

5.4 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/Network.hpp"
00013 #include "GameData/GameData.hpp"
00014 #include "Parsing/ServerParser.hpp"
00015 #include "GUIUpdater/GUIUpdater.hpp"
00016
00017 #include <time.h>
00018
00019 namespace Gui {
00020
00021     class Engine;
00022 };
00023
00024 class Gui::Engine {
00025     public:
00026
00027         Engine(std::shared_ptr<Network> network);
00028
00029         ~Engine() = default;
00030
00031         void run(void);
00032
00033     private:
00034
00035         ServerParser _parser;           // Parser class for server's command.
00036         std::shared_ptr<Network> _network; // Network class to connect to the server.
00037         std::shared_ptr<Render> _render;  // Render class to draw the scene.
00038         Event _event;                    // Event class to listen the user's inputs.
00039         std::shared_ptr<GameData> _gameData; // GameData class to store the game's data.
00040         GUIUpdater _guiUpdater;          // GUIUpdater class to update the GUI.
00041
00042         void listenServer(void);
00043
00044         void sendMessageUpdate(void);
00045 };

```

5.5 AError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** AError
00006 */
00007
00008 #pragma once
00009
00010 #include "Error/IError.hpp"
00011
00012 #include <string>
00013
00014 namespace Gui {
00015     namespace Errors {
00016         class AError;
00017     };
00018
00019     class Gui::Errors::AError : public IError {
00020     public:
00021
00022         ~AError() override = default;
00023
00024         const char *what() const noexcept override;
00025
00026     protected:
00027
00028         std::string _message;
00029     };

```

5.6 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
00015          class Error : public AError {};
00016
00017          class NetworkException : public Error {
00018          public:
00019              NetworkException(std::string message);
00020          };
00021
00022          class ServerParserException : public Error {
00023          public:
00024              ServerParserException(std::string message);
00025          };
00026
00027          class ParseCommandLineException : public Error {
00028          public:
00029              ParseCommandLineException(std::string message);
00030          };
00031
00032          class GuiGameDataException : public Error {
00033          public:
00034              GuiGameDataException(std::string message);
00035          };
00036
00037          class GuiUpdaterException : public Error {
00038          public:
00039              GuiUpdaterException(std::string message);
00040          };
00041      };
00042  };
00043  };

```

5.7 IError.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** IError
00006  */
00007
00008  #pragma once
00009
00010  #include <exception>
00011
00012  namespace Gui {
00013      namespace Errors {
00014          class IError;
00015      };
00016  };
00017
00018  class Gui::Errors::IError : public std::exception {
00019  public:
00020      virtual ~IError() = default;
00021      virtual const char *what() const noexcept = 0;
00022  };

```

5.8 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 "Render/Render.hpp"
00012
00013  #include <functional>
00014  #include <unordered_map>
00015
00016  namespace Gui {
00017
00022      class Event;
00023  };
00024
00025  class Gui::Event {
00026
00027      public:
00028
00033          Event();
00034
00039          ~Event() = default;
00040
00045          void listen();
00046
00052          void setRender(std::shared_ptr<Render> render);
00053
00059          void setGameData(std::shared_ptr<GameData> gameData);
00060
00061      private:
00062
00063          std::shared_ptr<Render> _render; // Render class to draw scene.
00064          std::shared_ptr<GameData> _gameData; // GameData class to contain scene.
00065
00070          std::unordered_map<KeyboardKey, std::function<void()> _eventsKeyDown =
00071          {
00072              {KEY_SPACE, [this]() {moveUpCamera();}},
00073              {KEY_LEFT_SHIFT, [this]() {moveDownCamera();}},
00074          };
00075
00080          std::unordered_map<KeyboardKey, std::function<void()> _eventsKeyPressed =
00081          {
00082              {KEY_THREE, [this]() {switchDisplayDebug();}},
00083              {KEY_F3, [this]() {switchDisplayDebug();}},
00084              {KEY_SPACE, [this]() {setFreeCam();}},
00085              {KEY_R, [this]() {switchTileHudToGame();}},
00086              {KEY_J, [this]() {increaseRenderDistance();}},
00087              {KEY_K, [this]() {decreaseRenderDistance();}},
00088              {KEY_F5, [this]() {changeActualPlayerPov();}},
00089              {KEY_APOSTROPHE, [this]() {changeActualPlayerPov();}},
00090              {KEY_KP_ADD, [this]() {increaseTimeUnit();}},
00091              {KEY_KP_SUBTRACT, [this]() {decreaseTimeUnit();}},
00092          };
00093
00098          std::unordered_map<MouseButton, std::function<void()> _eventsMousePressed =
00099          {
00100              {MOUSE_BUTTON_LEFT, [this]() {handleLeftClick();}},
00101              {MOUSE_BUTTON_RIGHT, [this]() {handleRightClick();}},
00102          };
00103
00108          void moveUpCamera();
00109
00114          void moveDownCamera();
00115
00120          void switchDisplayDebug();
00121
00125          void setFreeCam();
00126
00131          void handleLeftClick();
00132
00136          void handleRightClick();
00137
00142          void selectPlayer();
00143
00147          void selectTile();
00148
00154          void changePlayer(bool turn);
00155
00162          void changePlayerPOV(size_t playerId);
00163
00169          void setPlayerPov(size_t playerId);
00170
00175          void changeActualPlayerPov();
00176

```

```

00182         void changePOVToFirstPerson(size_t id);
00183
00189         void changePOVToSecondPerson(size_t id);
00190
00196         void changePOVToThirdPerson(size_t id);
00197
00202         void switchTileHudToGame();
00203
00208         void increaseRenderDistance();
00209
00214         void decreaseRenderDistance();
00215
00220         void increaseTimeUnit();
00221
00226         void decreaseTimeUnit();
00227 };

```

5.9 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
00078         void setId(std::size_t id);
00079
00085         void setTeam(const std::string &team);
00086
00092         void setPosition(std::pair<std::size_t, std::size_t> position);
00093
00099         void setState(EggState state);
00100
00106         EggState getState() const;
00107
00108     private:
00109
00110         std::size_t _id; // Id of the egg.
00111         std::string _team; // Team name of the egg.
00112         std::pair<std::size_t, std::size_t> _position; // Position of the egg.
00113         EggState _state; // State of the egg.
00114 };

```

5.10 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     class GameData;
00023 };
00024
00025 class Gui::GameData {
00026 public:
00027     enum TimeUnitState {
00028         INCREASE,
00029         DECREASE,
00030         NONE
00031     };
00032
00033     GameData();
00034     ~GameData() = default;
00035
00040     std::vector<Gui::Team> &getTeams();
00041
00046     Gui::Team &getTeam(const std::string &name);
00047
00053     void addTeam(const Gui::Team &team);
00054
00061     void addTeam(const std::string &name, const std::string &playerModelPath, const std::string
00062 &eggModelPath);
00063
00068     void addPlayerToTeam(const std::string &teamName, const Gui::Player &player);
00069
00078     Gui::Player &getPlayer(size_t id);
00079
00085     Map<Gui::Tile> &getMap();
00086
00092     void setMap(const Map<Gui::Tile> &map);
00093
00099     void setMapSize(size_t x, size_t y);
00100
00106     std::pair<size_t, size_t> getMapSize() const;
00107
00115     Gui::Tile &getTile(size_t x, size_t y);
00116
00122     void setTile(const Gui::Tile &tile);
00123
00131     void restartLastTick(void);
00132
00140     void setServerTick(std::size_t tick);
00141
00146     clock_t getLastTick() const;
00147
00153     std::size_t getServerTick() const;
00154
00160     void setIsEndGame(bool isEndGame);
00161
00167     bool getIsEndGame() const;
00168
00174     void setLastError(const std::string &error);
00175
00182     std::string getLastError() const;
00183
00189     Team &getTeamById(std::size_t id);
00190
00196     TimeUnitState getTimeUnitFromServer() const;
00197
00204     void setTimeUnitFromServer(TimeUnitState timeUnitFromServer);
00205
00211 private:
00212     std::vector<Gui::Team> _teams; // Teams of the game.
00213     Map<Gui::Tile> _map; // Map of the game.
00214     std::size_t _serverTick; // Tick value of the server.
00215     clock_t _lastTick; // Last tick of the GameData (based on the server

```

```

        tick).
00226         bool                _isEndGame;        // Is true if the game is finished.
00227         std::string          _lastError;        // Last error message.
00228         TimeUnitState        _timeUnitFromServer; // True if the time unit has
        changed.
00229     };

```

5.11 Inventory.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** Inventory
00006  */
00007
00008  #pragma once
00009
00010  #include <iostream>
00011
00012  #define RessourcesNumber 6
00013
00014  namespace Gui {
00015
00020      class Inventory;
00021  };
00022
00023  class Gui::Inventory {
00024
00025      public:
00026
00027          using Ressources = size_t [RessourcesNumber];
00028
00033          Inventory();
00034
00046          Inventory(std::size_t food, std::size_t linemate, std::size_t deraumere, std::size_t sibur,
std::size_t mendiane, std::size_t phiras, std::size_t thystame);
00047
00052          ~Inventory() = default;
00053
00059          void setFood(std::size_t food);
00060
00066          void setLinemate(std::size_t linemate);
00067
00073          void setDeraumere(std::size_t deraumere);
00074
00080          void setSibur(std::size_t sibur);
00081
00087          void setMendiane(std::size_t mendiane);
00088
00094          void setPhiras(std::size_t phiras);
00095
00101          void setThystame(std::size_t thystsame);
00102
00108          void setRessources(Ressources ressources);
00109
00115          std::size_t getFood(void);
00116
00122          std::size_t getLinemate(void);
00123
00129          std::size_t getDeraumere(void);
00130
00136          std::size_t getSibur(void);
00137
00143          std::size_t getMendiane(void);
00144
00150          std::size_t getPhiras(void);
00151
00157          std::size_t getThystame(void);
00158
00164          Ressources &getRessources(void);
00165
00166      private:
00167
00168          std::size_t    _food;        // Food.
00169          Ressources     _ressources;  // Ressources.
00170  };

```

5.12 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     class Player;
00015 };
00016
00017 class Gui::Player {
00018     public:
00019         enum PlayerState {
00020             IDLE,
00021             BORN,
00022             BROADCAST,
00023             EJECT,
00024             BEING_EJECTED,
00025             WALK,
00026             INCANTATION,
00027             FINISHED_INCANTATION,
00028             LAY_EGG,
00029             DROP,
00030             COLLECT,
00031             DEAD,
00032         };
00033
00034         Player(std::size_t id, const std::string &team, std::pair<std::size_t, std::size_t> position,
00035             std::size_t orientation, std::size_t level = 1);
00036
00037         ~Player() = default;
00038
00039         void setPosition(std::pair<std::size_t, std::size_t> position);
00040
00041         void setId(std::size_t id);
00042
00043         void setLevel(std::size_t level);
00044
00045         void setOrientation(std::size_t orientation);
00046
00047         void setTeam(const std::string &team);
00048
00049         std::pair<std::size_t, std::size_t> getPosition(void) const;
00050
00051         std::size_t getId(void) const;
00052
00053         std::size_t getLevel(void) const;
00054
00055         std::size_t getOrientation(void) const;
00056
00057         std::string getTeam(void) const;
00058
00059         void setState(PlayerState state);
00060
00061         PlayerState getState(void) const;
00062
00063         void setBroadcast(const std::string &broadcast);
00064
00065         std::string getBroadcast() const;
00066
00067         float getRotationFromOrientation() const;
00068
00069         Vector3 getCenterPosition();
00070
00071         Inventory inventory;
00072
00073     private:
00074         std::size_t _id; // Id of the player.
00075         std::string _team; // Team name.
00076         std::pair<std::size_t, std::size_t> _position; // Position x y.
00077         std::size_t _orientation; // Orientation of the player.
00078         std::size_t _level; // Level between 1 - 8.
00079         PlayerState _state; // Player state.
00080         std::string _broadcast; // Broadcast message.
00081 };

```

5.13 Team.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** Team
00006 */
00007
00008 #pragma once
00009
00010 #include "raylib.h"
00011 #include "Types.hpp"
00012 #include "GameDatas/Egg.hpp"
00013 #include "GameDatas/Tile.hpp"
00014 #include "GameDatas/Player.hpp"
00015
00016 #include <vector>
00017 #include <memory>
00018
00019 namespace Gui {
00020
00021     class Team;
00022 };
00023
00024 class Gui::Team {
00025 public:
00026
00027     Team(const std::string &name, const std::string &playerModelPath, const std::string
&eggModelPath);
00028
00029     ~Team();
00030
00031     const std::string &getName() const;
00032
00033     std::vector<Gui::Player> &getPlayers();
00034
00035     std::vector<Gui::Egg> &getEggs();
00036
00037     void setName(const std::string &name);
00038
00039     void addPlayer(const Gui::Player &player);
00040
00041     void addEgg(const Gui::Egg &egg);
00042
00043     bool removePlayer(std::size_t id);
00044
00045     bool removeEgg(std::size_t id);
00046
00047     std::shared_ptr<Gui::Player> getPlayer(std::size_t id);
00048
00049     Model getPlayerModel() const;
00050
00051     void setPlayerModelPath(const std::string &playerModelPath);
00052
00053     std::shared_ptr<Gui::Egg> getEgg(std::size_t id);
00054
00055     Model getEggModel() const;
00056
00057     void setEggModelPath(const std::string &eggModelPath);
00058
00059     std::vector<BoundingBox> getPlayerBoundingBoxes(std::pair<size_t, size_t> pos, size_t
orientation, Vector3 center);
00060
00061     Vector3 getPlayerPositionIn3DSpace(size_t id, Map<Tile> map);
00062
00063     std::vector<RayCollision> getPlayerModelHitbox(size_t id, Camera camera);
00064
00065     bool isPlayerHit(size_t id, Camera camera);
00066
00067 private:
00068
00069     std::string _name; // Name of the team.
00070     std::vector<Gui::Player> _players; // Players of the team.
00071     Model _playerModel; // Model player asset of the team.
00072     std::vector<Gui::Egg> _eggs; // Eggs of the team.
00073     Model _eggModel; // Eggs Model of the team.
00074
00075     BoundingBox rotateBoundingBoxByOrientation(BoundingBox bbox, size_t orientation,
std::pair<size_t, size_t> pos, Vector3 center);
00076 };

```


5.14 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;           // Position x y.
00106          Vector3 _positionIn3DSpace;           // Position in 3D space.
00107  };

```

5.15 GUIUpdater.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** GUIUpdater
00006  */
00007
00008  #pragma once
00009
00010  #include "Network/Network.hpp"
00011  #include "GameDatas/GameData.hpp"
00012
00013  #include <string>
00014  #include <functional>
00015  #include <unordered_map>
00016
00017  namespace Gui {
00018
00024      class GUIUpdater;
00025  }
00026
00027  class Gui::GUIUpdater {
00028
00029      public:
00030
00037          GUIUpdater(std::shared_ptr<GameData> gameData, std::shared_ptr<Network> network);
00038
00042          ~GUIUpdater() = default;
00043
00050          void update(const std::string &command, const std::vector<std::string> &data);

```

```

00051
00052     private:
00053
00054         std::shared_ptr<GameData> _gameData; // The GUI GameData to update.
00055         std::shared_ptr<Network> _network; // The network to send commands to the server.
00056
00057         std::unordered_map<std::string, std::function<void(std::vector<std::string>)>> _updateMap =
00058         {
00059             {"msz", [this](std::vector<std::string> data){updateMapSize(data);}},
00060             {"bct", [this](std::vector<std::string> data){updateMapContent(data);}},
00061             {"tna", [this](std::vector<std::string> data){updateTeamNames(data);}},
00062             {"pnw", [this](std::vector<std::string> data){updateTeamMember(data);}},
00063             {"ppo", [this](std::vector<std::string> data){updatePlayerPosition(data);}},
00064             {"plv", [this](std::vector<std::string> data){updatePlayerLevel(data);}},
00065             {"pin", [this](std::vector<std::string> data){updatePlayerInventory(data);}},
00066             {"pex", [this](std::vector<std::string> data){updatePlayerExpulsion(data);}},
00067             {"pbc", [this](std::vector<std::string> data){updatePlayerBroadcast(data);}},
00068             {"pic", [this](std::vector<std::string> data){updatePlayerStartIncantation(data);}},
00069             {"pie", [this](std::vector<std::string> data){updatePlayerEndIncantation(data);}},
00070             {"pfk", [this](std::vector<std::string> data){updatePlayerEggLaying(data);}},
00071             {"pdr", [this](std::vector<std::string> data){updatePlayerRessourceDropping(data);}},
00072             {"pgt", [this](std::vector<std::string> data){updatePlayerRessourceCollecting(data);}},
00073             {"pdi", [this](std::vector<std::string> data){updatePlayerDeath(data);}},
00074             {"enw", [this](std::vector<std::string> data){updateEggLaidByPlayer(data);}},
00075             {"ebo", [this](std::vector<std::string> data){updatePlayerBorn(data);}},
00076             {"edi", [this](std::vector<std::string> data){updateEggDeath(data);}},
00077             {"sgt", [this](std::vector<std::string> data){updateTimeUnitRequest(data);}},
00078             {"sst", [this](std::vector<std::string> data){updateTimeUnitModification(data);}},
00079             {"seg", [this](std::vector<std::string> data){updateEndOfGame(data);}},
00080             {"smg", [this](std::vector<std::string> data){updateMessageFromServer(data);}},
00081             {"suc", [this](std::vector<std::string> data){updateUnknownMessage(data);}},
00082             {"sbp", [this](std::vector<std::string> data){updateCommandParameter(data);}},
00083         }; // The map of commands to update the GUI GameData.
00084
00090         void updateMapSize(const std::vector<std::string> &data);
00091
00097         void updateMapContent(const std::vector<std::string> &data);
00098
00104         void updateTeamNames(const std::vector<std::string> &data);
00105
00111         void updateTeamMember(const std::vector<std::string> &data);
00112
00118         void updatePlayerPosition(const std::vector<std::string> &data);
00119
00125         void updatePlayerLevel(const std::vector<std::string> &data);
00126
00132         void updatePlayerInventory(const std::vector<std::string> &data);
00133
00139         void updatePlayerExpulsion(const std::vector<std::string> &data);
00140
00146         void updatePlayerBroadcast(const std::vector<std::string> &data);
00147
00153         void updatePlayerStartIncantation(const std::vector<std::string> &data);
00154
00160         void updatePlayerEndIncantation(const std::vector<std::string> &data);
00161
00167         void updatePlayerEggLaying(const std::vector<std::string> &data);
00168
00174         void updatePlayerRessourceDropping(const std::vector<std::string> &data);
00175
00181         void updatePlayerRessourceCollecting(const std::vector<std::string> &data);
00182
00188         void updatePlayerDeath(const std::vector<std::string> &data);
00189
00195         void updateEggLaidByPlayer(const std::vector<std::string> &data);
00196
00202         void updatePlayerBorn(const std::vector<std::string> &data);
00203
00209         void updateEggDeath(const std::vector<std::string> &data);
00210
00216         void updateTimeUnitRequest(const std::vector<std::string> &data);
00217
00223         void updateTimeUnitModification(const std::vector<std::string> &data);
00224
00230         void updateEndOfGame(const std::vector<std::string> &data);
00231
00237         void updateMessageFromServer(const std::vector<std::string> &data);
00238
00244         void updateUnknownMessage(const std::vector<std::string> &data);
00245
00251         void updateCommandParameter(const std::vector<std::string> &data);
00252     };

```

5.16 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
00015     class AHud;
00016 };
00017
00018 class Gui::AHud : public Gui::IHud {
00019 public:
00020
00021     ~AHud() = default;
00022
00023     virtual void display() = 0;
00024
00025     void setPlayer(std::shared_ptr<Player> player);
00026
00027     void setTile(std::shared_ptr<Tile> tile);
00028
00029     TypeScene getType() const;
00030
00031 protected:
00032
00033     TypeScene _typeScene; // Type of the scene.
00034     std::shared_ptr<GameData> _gameData; // GameData class.
00035     std::shared_ptr<Player> _player; // Player to display hud.
00036     std::shared_ptr<Tile> _tile; // Tile to display hud.
00037 };

```

5.17 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
00019     class HudGame;
00020 };
00021
00022 class Gui::HudGame : public Gui::AHud {
00023 public:
00024
00025     HudGame(std::shared_ptr<GameData> gameData);
00026
00027     ~HudGame() = default;
00028
00029     void display();
00030
00031 private:
00032
00033     Texture2D _texture; // Texture for Hud Background.
00034     Font _font; // Font for Hud's texts.
00035 };

```

5.18 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{25, 420}
00014  #define HUD_PLAYER_TEXT_TITLE_POS Vector2{50, 380}
00015  #define HUD_PLAYER_TEXT_MARGING 30
00016
00017  namespace Gui {
00018
00019      class HudPlayer;
00020  };
00021
00022  class Gui::HudPlayer : public Gui::AHud {
00023  public:
00024
00025      HudPlayer(std::shared_ptr<GameData> gameData);
00026
00027      ~HudPlayer() = default;
00028
00029      void display();
00030
00031  private:
00032
00033      Texture2D _texture;          // Texture for Hud Background.
00034      Font _font;                  // Font for Hud's texts.
00035  };

```

5.19 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{25, 420}
00014  #define HUD_TILE_TEXT_TITLE_POS Vector2{50, 380}
00015  #define HUD_TILE_TEXT_MARGING 30
00016
00017  namespace Gui {
00018
00019      class HudTile;
00020  };
00021
00022  class Gui::HudTile : public Gui::AHud {
00023  public:
00024
00025      HudTile(std::shared_ptr<GameData> gameData);
00026
00027      ~HudTile() = default;
00028
00029      void display();
00030
00031  private:
00032
00033      Texture2D _texture;          // Texture for Hud Background.
00034      Font _font;                  // Font for Hud's texts.
00035  };

```

5.20 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  };

```

5.21 Network.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** Network
00006  */
00007
00008  #pragma once
00009
00010  #include "Error/Error.hpp"
00011
00012  #include <string>
00013  #include <arpa/inet.h>
00014  #include <sys/socket.h>
00015  #include <netinet/in.h>
00016
00017  #define MAX_PORT 65535
00018  #define MIN_PORT 1
00019
00020  namespace Gui {
00021
00026      class Network;
00027  };
00028
00029  class Gui::Network {
00030
00031      public:
00032
00039          Network(int port, const std::string& hostName);
00040
00045          ~Network() = default;
00046
00052          void setPort(int port);
00053
00059          void setHostName(const std::string& hostName);
00060
00066          int getPort() const;
00067
00073          std::string getHostName() const;
00074
00079          void connectToServer();

```

```

00080
00086     const std::string listenServer();
00087
00093     void sendMessageServer(const std::string& message);
00094
00095     private:
00096
00101         void selectServer();
00102
00108         const std::string readInfoServer();
00109
00110         int             _port;           // server port
00111         std::string      _hostName;       // server hostname
00112         int              _serverFd;       // server file descriptor
00113         fd_set            _writeFd;       // file descriptor for write access
00114         fd_set            _readFd;        // file descriptor for read access
00115         bool              _isConnected;   // is true if the gui is connected to the server
00116 };

```

5.22 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;           // port server
00066         std::string      _hostName;       // host name server
00067 };

```

5.23 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
00012 #include <string>
00013 #include <functional>
00014 #include <unordered_map>
00015
00016 namespace Gui {
00017
00022     class ServerParser;

```

```

00023 };
00024
00025 class Gui::ServerParser {
00026     public:
00027
00028         ServerParser() = default;
00033
00034         ~ServerParser() = default;
00039
00040         std::vector<std::string> parse(const std::string& command);
00047
00048         enum ParseType {
00053             INT,
00054             STRING,
00055             MESSAGE,
00056             LIST_INT
00057         };
00058
00059     private:
00060
00061         std::unordered_map<std::string, std::vector<ParseType> > _typesCommand =
00066         {
00067             {"msz", std::vector<ParseType>{INT, INT}},
00068             {"bct", std::vector<ParseType>{INT, INT, INT, INT, INT, INT, INT, INT, INT, INT}},
00069             {"tna", std::vector<ParseType>{STRING}},
00070             {"pnw", std::vector<ParseType>{INT, INT, INT, INT, INT, INT, STRING}},
00071             {"ppo", std::vector<ParseType>{INT, INT, INT, INT}},
00072             {"plv", std::vector<ParseType>{INT, INT}},
00073             {"pin", std::vector<ParseType>{INT, INT, INT, INT, INT, INT, INT, INT, INT, INT, INT}},
00074             {"pex", std::vector<ParseType>{INT}},
00075             {"pbc", std::vector<ParseType>{INT, MESSAGE}},
00076             {"pic", std::vector<ParseType>{INT, INT, INT, LIST_INT}},
00077             {"pie", std::vector<ParseType>{INT, INT, INT}},
00078             {"pfk", std::vector<ParseType>{INT}},
00079             {"pdr", std::vector<ParseType>{INT, INT}},
00080             {"pgt", std::vector<ParseType>{INT, INT}},
00081             {"pdi", std::vector<ParseType>{INT}},
00082             {"enw", std::vector<ParseType>{INT, INT, INT, INT}},
00083             {"ebo", std::vector<ParseType>{INT}},
00084             {"edi", std::vector<ParseType>{INT}},
00085             {"sgt", std::vector<ParseType>{INT}},
00086             {"sst", std::vector<ParseType>{INT}},
00087             {"seg", std::vector<ParseType>{STRING}},
00088             {"smg", std::vector<ParseType>{MESSAGE}},
00089             {"suc", std::vector<ParseType>{}},
00090             {"sbp", std::vector<ParseType>{}}
00091         };
00092
00093         std::vector<std::string> parseCommand(const std::string& command, std::vector<ParseType>
00101         types);
00102
00110         std::vector<std::string> parseInt(std::istringstream& stream, std::vector<std::string>
00110         arguments);
00111
00119         std::vector<std::string> parseString(std::istringstream& stream, std::vector<std::string>
00119         arguments);
00120
00129         std::vector<std::string> parseMessage(std::istringstream& stream, std::vector<std::string>
00129         arguments, std::string commandName);
00130
00139         std::vector<std::string> parseListInt(std::istringstream& stream, std::vector<std::string>
00139         arguments, std::string commandName);
00140 };

```

5.24 Decoration.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** Decoration
00006  */
00007
00008 #pragma once
00009
00010 #include "raylib.h"
00011 #include "Types.hpp"
00012 #include "Assets.hpp"
00013
00014 #include <vector>
00015 #include <iostream>
00016

```

```

00017 namespace Gui {
00018
00024     class Decoration;
00025 }
00026
00027 class Gui::Decoration {
00028
00029     public:
00030
00035         Decoration();
00036
00041         ~Decoration() = default;
00042
00050         void display(std::pair<std::size_t, std::size_t> mapSize, size_t renderDistance,
std::pair<std::size_t, std::size_t> camPos);
00051
00060         Map<bool> getGenerationItem(std::size_t ratio);
00061
00062     private:
00063
00064         Model                _treeModel;           // Tree model asset.
00065         Map<bool>            _mapTree;              // Map to display trees.
00066         Model                _lanternModel;         // Lantern model asset.
00067         Map<bool>            _mapLantern;           // Map to display lanterns.
00068
00069         std::pair<std::size_t, std::size_t> _mapSize; // Size of the map.
00070
00078         void displayTree(size_t i, size_t j, Vector3 posTile);
00079
00087         void displayLantern(size_t i, size_t j, Vector3 posTile);
00088 };

```

5.25 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
00031     class Render;
00032 };
00033
00034 class Gui::Render {
00035
00036     public:
00037
00042         Render(std::shared_ptr<GameData> gameData);
00043
00048         ~Render();
00049
00056         bool isOpen();
00057
00062         void draw();
00063
00069         std::shared_ptr<Camera> getCamera();
00070
00076         void setIsDebug(bool isDebug);
00077
00084         bool getIsDebug(void);
00085
00091         void setCameraType(Gui::UserCamera::CameraType type);
00092

```



```

00098         Gui::UserCamera::CameraType getCameraType() const;
00099
00105         void setCameraPlayerPov(std::size_t id);
00106
00112         std::size_t getCameraPlayerPov() const;
00113
00119         void setCameraTile(std::pair<std::size_t, std::size_t> pos);
00120
00126         std::pair<std::size_t, std::size_t> getCameraTile() const;
00127
00132         Model getTileModel() const;
00133
00139         void setRenderDistance(size_t renderDistance);
00140
00145         size_t getRenderDistance() const;
00146
00154         bool isCameraInPlayerPov() const;
00155
00161         size_t getTimeUnit() const;
00162
00168         void setTimeUnit(size_t timeUnit);
00169
00170     private:
00171
00172         UserCamera _camera; // Camera of the scene.
00173         bool _isDebug; // Display or not the debug
00174         informations.
00175         std::shared_ptr<GameData> _gameData; // GameData class to store the
00176         game's data.
00177         std::shared_ptr<Decoration> _decoration; // Decoration to display;
00178         std::vector<std::shared_ptr<Gui::IHud> _hudList; // List of huds.
00179         size_t _renderDistance; // Distance to render from the
00180         3d position of the camera.
00181         Model _tileModel; // Model to display tiles.
00182         Model _foodModel; // Model to display foods.
00183         Model _linemateModel; // Model to display linemates.
00184         Model _mendianeModel; // Model to display mendianes.
00185         Model _phirasModel; // Model to display phiras.
00186         Model _siburModel; // Model to display siburs.
00187         Model _thystameModel; // Model to display thystames.
00188         Model _deraumereModel; // Model to display
00189         deraumeres.
00190         Texture2D _cursorTexture; // Cursor texture.
00191
00193         void LoadModels(void);
00194
00199         void displayHUD(void);
00200
00205         void displayDebug(void);
00206
00211         void displayPlayers(void);
00212
00217         void displayMap(void);
00218
00223         void displayTile(Tile tile);
00224
00230         void displayEggs(Tile tile) const;
00231
00237         void displayFood(Tile tile) const;
00238
00244         void displayResources(Tile tile) const;
00245
00251         void displayLinemate(Tile tile) const;
00252
00258         void displayMendiane(Tile tile) const;
00259
00265         void displayPhiras(Tile tile) const;
00266
00272         void displaySibur(Tile tile) const;
00273
00279         void displayThystame(Tile tile) const;
00280
00286         void displayDeraumere(Tile tile) const;
00287
00292         void displayCursor();
00293
00299         std::pair<std::size_t, std::size_t> getCameraTile();
00300 };

```

5.26 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     class UserCamera;
00021 };
00022
00023 class Gui::UserCamera {
00024     public:
00025
00026         enum CameraType {
00027             FREE,
00028             FIRST_PERSON,
00029             SECOND_PERSON,
00030             THIRD_PERSON,
00031             FREE_TILE
00032         };
00033
00034         UserCamera();
00039
00040         ~UserCamera() = default;
00045
00046         void setPosition(Vector3 position);
00052
00053         void setTarget(Vector3 target);
00059
00060         void setUp(Vector3 up);
00066
00067         void setFovy(float fovy);
00073
00074         Vector3 getPosition(void) const;
00080
00081         Vector3 getTarget(void) const;
00087
00088         Vector3 getUp(void) const;
00094
00095         float getFovy(void) const;
01001
01002         std::shared_ptr<Camera> getCamera();
01008
01009         void setType(CameraType type);
01015
01016         CameraType getType() const;
01022
01023         void setPlayerId(size_t playerId);
01029
01030         size_t getPlayerId() const;
01036
01037         void setTilePos(std::pair<std::size_t, std::size_t> pos);
01043
01044         std::pair<std::size_t, std::size_t> getTilePos() const;
01050
01051         bool isPlayerPov() const;
01059
01060     private:
01061
01062         std::shared_ptr<Camera> _camera;           // Camera raylib instance.
01063         CameraType _type;                         // Type of camera.
01064         size_t _playerId;                         // Player id.
01065         std::pair<std::size_t, std::size_t> _tilePos; // Tile position.
01066
01067 };

```

5.27 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, 102
85
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Colors.hpp, 103
86
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Config.hpp, 104
86
_message
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Engine/Engine.hpp, 8
86
~Decoration
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/AError.hpp, 9
87
~Egg
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/EngineEgg.hpp, 11
87
~Engine
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Engine.hpp, 14
88
~Event
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/Event.hpp, 16
88
~GameData
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUI/GUIData.hpp, 18
95
~HudGame
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EngineGame, 30
90
~HudPlayer
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EngineGame, 30
90
~HudTile
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EngineGame, 30
92
~IHud
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EngineGame, 30
92
~Inventory
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EngineGame, 30
94
~Network
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/EngineGame, 30
95
~ParseCommandLine
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/AHud.hpp, 48
97
~Player
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp, 52
97
~Render
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp, 52
98
~ServerParser
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp, 52
98
~Team
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/IHudGame.hpp, 68
98
~Tile
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/Network.hpp, 70
99
~UserCamera
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ParseCommandLine.hpp, 80
100
addEgg
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ServerParser.hpp, 80
100
Gui::Team, 68
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Decoration.hpp, 80
101
Gui::Team, 68
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Render.hpp, 80
addPlayerToTeam
Gui::GameData, 18

- addTeam
 - Gui::GameData, [19](#)
- connectToServer
 - Gui::Network, [45](#)
- Decoration
 - Gui::Decoration, [9](#)
- display
 - Gui::Decoration, [9](#)
 - Gui::HudGame, [30](#)
 - Gui::HudPlayer, [32](#)
 - Gui::HudTile, [34](#)
 - Gui::IHud, [36](#)
- draw
 - Gui::Render, [58](#)
- Egg
 - Gui::Egg, [10](#)
- Engine
 - Gui::Engine, [13](#)
- Event
 - Gui::Event, [15](#)
- GameData
 - Gui::GameData, [18](#)
- getBroadcast
 - Gui::Player, [53](#)
- getCamera
 - Gui::Render, [59](#)
 - Gui::UserCamera, [80](#)
- getCameraPlayerPov
 - Gui::Render, [59](#)
- getCameraTile
 - Gui::Render, [59](#)
- getCameraType
 - Gui::Render, [59](#)
- getCenterPosition
 - Gui::Player, [53](#)
- getDeraumere
 - Gui::Inventory, [39](#)
- getEgg
 - Gui::Team, [69](#)
- getEggModel
 - Gui::Team, [69](#)
- getEggs
 - Gui::Team, [69](#)
- getFood
 - Gui::Inventory, [40](#)
- getFovy
 - Gui::UserCamera, [80](#)
- getGenerationItem
 - Gui::Decoration, [9](#)
- getHostName
 - Gui::Network, [45](#)
 - Gui::ParseCommandLine, [49](#)
- getId
 - Gui::Egg, [11](#)
 - Gui::Player, [53](#)
- getIsDebug
 - Gui::Render, [60](#)
- getIsEndGame
 - Gui::GameData, [19](#)
- getLastError
 - Gui::GameData, [19](#)
- getLastTick
 - Gui::GameData, [20](#)
- getLevel
 - Gui::Player, [53](#)
- getLinemate
 - Gui::Inventory, [40](#)
- getMap
 - Gui::GameData, [20](#)
- getMapSize
 - Gui::GameData, [20](#)
- getMendiane
 - Gui::Inventory, [40](#)
- getName
 - Gui::Team, [69](#)
- getOrientation
 - Gui::Player, [53](#)
- getPhiras
 - Gui::Inventory, [40](#)
- getPlayer
 - Gui::GameData, [20](#)
 - Gui::Team, [70](#)
- getPlayerBoundingBoxes
 - Gui::Team, [70](#)
- getPlayerId
 - Gui::UserCamera, [80](#)
- getPlayerModel
 - Gui::Team, [70](#)
- getPlayerModelHitbox
 - Gui::Team, [71](#)
- getPlayerPositionIn3DSpace
 - Gui::Team, [71](#)
- getPlayers
 - Gui::Team, [71](#)
- getPort
 - Gui::Network, [45](#)
 - Gui::ParseCommandLine, [49](#)
- getPosition
 - Gui::Egg, [11](#)
 - Gui::Player, [54](#)
 - Gui::Tile, [76](#)
 - Gui::UserCamera, [81](#)
- getPositionIn3DSpace
 - Gui::Tile, [76](#)
- getRenderDistance
 - Gui::Render, [60](#)
- getRessources
 - Gui::Inventory, [41](#)
- getRotationFromOrientation
 - Gui::Player, [54](#)
- getServerTick
 - Gui::GameData, [21](#)
- getSibur

- Gui::Inventory, 41
- getState
 - Gui::Egg, 11
 - Gui::Player, 54
- getTarget
 - Gui::UserCamera, 81
- getTeam
 - Gui::Egg, 12
 - Gui::GameData, 21
 - Gui::Player, 54
- getTeamById
 - Gui::GameData, 21
- getTeams
 - Gui::GameData, 22
- getThystame
 - Gui::Inventory, 41
- getTile
 - Gui::GameData, 22
- getTileBoundingBoxes
 - Gui::Tile, 77
- getTileModel
 - Gui::Render, 60
- getTileModelHitbox
 - Gui::Tile, 77
- getTilePos
 - Gui::UserCamera, 81
- getTimeUnit
 - Gui::Render, 60
- getTimeUnitFromServer
 - Gui::GameData, 22
- getType
 - Gui::IHud, 37
 - Gui::UserCamera, 81
- getUp
 - Gui::UserCamera, 82
- Gui::AHud, 8
- Gui::Decoration, 8
 - ~Decoration, 9
 - Decoration, 9
 - display, 9
 - getGenerationItem, 9
- Gui::Egg, 10
 - ~Egg, 11
 - Egg, 10
 - getId, 11
 - getPosition, 11
 - getState, 11
 - getTeam, 12
 - setId, 12
 - setPosition, 12
 - setState, 13
 - setTeam, 13
- Gui::Engine, 13
 - ~Engine, 14
 - Engine, 13
 - run, 14
- Gui::Errors::AError, 7
 - _message, 8
 - what, 8
- Gui::Errors::Error, 14
- Gui::Errors::GuiGameDataException, 25
 - GuiGameDataException, 26
- Gui::Errors::GuiUpdaterException, 27
 - GuiUpdaterException, 28
- Gui::Errors::IError, 35
 - what, 35
- Gui::Errors::NetworkException, 46
 - NetworkException, 47
- Gui::Errors::ParseCommandLineException, 50
 - ParseCommandLineException, 50
- Gui::Errors::ServerParserException, 65
 - ServerParserException, 66
- Gui::Event, 15
 - ~Event, 16
 - Event, 15
 - listen, 16
 - setGameData, 16
 - setRender, 16
- Gui::GameData, 17
 - ~GameData, 18
 - addPlayerToTeam, 18
 - addTeam, 19
 - GameData, 18
 - getIsEndGame, 19
 - getLastError, 19
 - getLastTick, 20
 - getMap, 20
 - getMapSize, 20
 - getPlayer, 20
 - getServerTick, 21
 - getTeam, 21
 - getTeamById, 21
 - getTeams, 22
 - getTile, 22
 - getTimeUnitFromServer, 22
 - restartLastTick, 22
 - setIsEndGame, 23
 - setLastError, 23
 - setMap, 23
 - setMapSize, 23
 - setServerTick, 24
 - setTile, 24
 - setTimeUnitFromServer, 24
- Gui::GUIUpdater, 26
 - GUIUpdater, 26
 - update, 27
- Gui::HudGame, 29
 - ~HudGame, 30
 - display, 30
 - HudGame, 30
- Gui::HudPlayer, 31
 - ~HudPlayer, 32
 - display, 32
 - HudPlayer, 32
- Gui::HudTile, 33
 - ~HudTile, 34

- display, 34
- HudTile, 34
- Gui::IHud, 36
 - ~IHud, 36
 - display, 36
 - getType, 37
 - setPlayer, 37
 - setTile, 37
 - TypeScene, 36
- Gui::Inventory, 38
 - ~Inventory, 39
 - getDeraumere, 39
 - getFood, 40
 - getLinemate, 40
 - getMendiane, 40
 - getPhiras, 40
 - getRessources, 41
 - getSibur, 41
 - getThystame, 41
 - Inventory, 39
 - setDeraumere, 41
 - setFood, 42
 - setLinemate, 42
 - setMendiane, 42
 - setPhiras, 42
 - setRessources, 43
 - setSibur, 43
 - setThystame, 43
- Gui::Network, 44
 - ~Network, 45
 - connectToServer, 45
 - getHostName, 45
 - getPort, 45
 - listenServer, 45
 - Network, 44
 - sendMessageServer, 45
 - setHostName, 46
 - setPort, 46
- Gui::ParseCommandLine, 48
 - ~ParseCommandLine, 48
 - getHostName, 49
 - getPort, 49
 - ParseCommandLine, 48
 - parseFlags, 49
- Gui::Player, 51
 - ~Player, 52
 - getBroadcast, 53
 - getCenterPosition, 53
 - getId, 53
 - getLevel, 53
 - getOrientation, 53
 - getPosition, 54
 - getRotationFromOrientation, 54
 - getState, 54
 - getTeam, 54
 - inventory, 57
 - Player, 52
 - setBroadcast, 55
 - setId, 55
 - setLevel, 55
 - setOrientation, 56
 - setPosition, 56
 - setState, 56
 - setTeam, 56
- Gui::Render, 57
 - ~Render, 58
 - draw, 58
 - getCamera, 59
 - getCameraPlayerPov, 59
 - getCameraTile, 59
 - getCameraType, 59
 - getIsDebug, 60
 - getRenderDistance, 60
 - getTileModel, 60
 - getTimeUnit, 60
 - isCameraInPlayerPov, 60
 - isOpen, 61
 - Render, 58
 - setCameraPlayerPov, 61
 - setCameraTile, 61
 - setCameraType, 63
 - setIsDebug, 63
 - setRenderDistance, 63
 - setTimeUnit, 63
- Gui::ServerParser, 64
 - ~ServerParser, 65
 - parse, 65
 - ParseType, 64
 - ServerParser, 64
- Gui::Team, 67
 - ~Team, 68
 - addEgg, 68
 - addPlayer, 68
 - getEgg, 69
 - getEggModel, 69
 - getEggs, 69
 - getName, 69
 - getPlayer, 70
 - getPlayerBoundingBoxes, 70
 - getPlayerModel, 70
 - getPlayerModelHitbox, 71
 - getPlayerPositionIn3DSpace, 71
 - getPlayers, 71
 - isPlayerHit, 72
 - removeEgg, 72
 - removePlayer, 72
 - setEggModelPath, 74
 - setName, 74
 - setPlayerModelPath, 74
 - Team, 68
- Gui::Tile, 75
 - ~Tile, 76
 - getPosition, 76
 - getPositionIn3DSpace, 76
 - getTileBoundingBoxes, 77
 - getTileModelHitbox, 77

- inventory, 78
- isTileHit, 77
- setPosition, 78
- Tile, 75, 76
- Gui::UserCamera, 79
 - ~UserCamera, 80
 - getCamera, 80
 - getFovy, 80
 - getPlayerId, 80
 - getPosition, 81
 - getTarget, 81
 - getTilePos, 81
 - getType, 81
 - getUp, 82
 - isPlayerPov, 82
 - setFovy, 82
 - setPlayerId, 83
 - setPosition, 83
 - setTarget, 83
 - setTilePos, 83
 - setType, 84
 - setUp, 84
 - UserCamera, 79
- GuiGameDataException
 - Gui::Errors::GuiGameDataException, 26
- GUIUpdater
 - Gui::GUIUpdater, 26
- GuiUpdaterException
 - Gui::Errors::GuiUpdaterException, 28
- HudGame
 - Gui::HudGame, 30
- HudPlayer
 - Gui::HudPlayer, 32
- HudTile
 - Gui::HudTile, 34
- Inventory
 - Gui::Inventory, 39
- inventory
 - Gui::Player, 57
 - Gui::Tile, 78
- isCameraInPlayerPov
 - Gui::Render, 60
- isOpen
 - Gui::Render, 61
- isPlayerHit
 - Gui::Team, 72
- isPlayerPov
 - Gui::UserCamera, 82
- isTileHit
 - Gui::Tile, 77
- listen
 - Gui::Event, 16
- listenServer
 - Gui::Network, 45
- Network
 - Gui::Network, 44
 - NetworkException
 - Gui::Errors::NetworkException, 47
 - parse
 - Gui::ServerParser, 65
 - ParseCommandLine
 - Gui::ParseCommandLine, 48
 - ParseCommandLineException
 - Gui::Errors::ParseCommandLineException, 50
 - parseFlags
 - Gui::ParseCommandLine, 49
 - ParseType
 - Gui::ServerParser, 64
 - Player
 - Gui::Player, 52
 - removeEgg
 - Gui::Team, 72
 - removePlayer
 - Gui::Team, 72
 - Render
 - Gui::Render, 58
 - restartLastTick
 - Gui::GameData, 22
 - run
 - Gui::Engine, 14
 - sendMessageServer
 - Gui::Network, 45
 - ServerParser
 - Gui::ServerParser, 64
 - ServerParserException
 - Gui::Errors::ServerParserException, 66
 - setBroadcast
 - Gui::Player, 55
 - setCameraPlayerPov
 - Gui::Render, 61
 - setCameraTile
 - Gui::Render, 61
 - setCameraType
 - Gui::Render, 63
 - setDeraumere
 - Gui::Inventory, 41
 - setEggModelPath
 - Gui::Team, 74
 - setFood
 - Gui::Inventory, 42
 - setFovy
 - Gui::UserCamera, 82
 - setGameData
 - Gui::Event, 16
 - setHostName
 - Gui::Network, 46
 - setId
 - Gui::Egg, 12
 - Gui::Player, 55
 - setIsDebug
 - Gui::Render, 63

- setIsEndGame
 - Gui::GameData, [23](#)
- setLastError
 - Gui::GameData, [23](#)
- setLevel
 - Gui::Player, [55](#)
- setLinemate
 - Gui::Inventory, [42](#)
- setMap
 - Gui::GameData, [23](#)
- setMapSize
 - Gui::GameData, [23](#)
- setMendiane
 - Gui::Inventory, [42](#)
- setName
 - Gui::Team, [74](#)
- setOrientation
 - Gui::Player, [56](#)
- setPhiras
 - Gui::Inventory, [42](#)
- setPlayer
 - Gui::IHud, [37](#)
- setPlayerId
 - Gui::UserCamera, [83](#)
- setPlayerModelPath
 - Gui::Team, [74](#)
- setPort
 - Gui::Network, [46](#)
- setPosition
 - Gui::Egg, [12](#)
 - Gui::Player, [56](#)
 - Gui::Tile, [78](#)
 - Gui::UserCamera, [83](#)
- setRender
 - Gui::Event, [16](#)
- setRenderDistance
 - Gui::Render, [63](#)
- setRessources
 - Gui::Inventory, [43](#)
- setServerTick
 - Gui::GameData, [24](#)
- setSibur
 - Gui::Inventory, [43](#)
- setState
 - Gui::Egg, [13](#)
 - Gui::Player, [56](#)
- setTarget
 - Gui::UserCamera, [83](#)
- setTeam
 - Gui::Egg, [13](#)
 - Gui::Player, [56](#)
- setThystame
 - Gui::Inventory, [43](#)
- setTile
 - Gui::GameData, [24](#)
 - Gui::IHud, [37](#)
- setTilePos
 - Gui::UserCamera, [83](#)
- setTimeUnit
 - Gui::Render, [63](#)
- setTimeUnitFromServer
 - Gui::GameData, [24](#)
- setType
 - Gui::UserCamera, [84](#)
- setUp
 - Gui::UserCamera, [84](#)
- Team
 - Gui::Team, [68](#)
- Tile
 - Gui::Tile, [75](#), [76](#)
- TypeScene
 - Gui::IHud, [36](#)
- update
 - Gui::GUIUpdater, [27](#)
- UserCamera
 - Gui::UserCamera, [79](#)
- what
 - Gui::Errors::AError, [8](#)
 - Gui::Errors::LError, [35](#)