

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	8
4.1.2 Member Function Documentation	8
4.1.2.1 what()	8
4.2 Gui::AEvent Class Reference	8
4.3 Gui::AGUIUpdater Class Reference	8
4.3.1 Constructor & Destructor Documentation	9
4.3.1.1 AGUIUpdater()	9
4.3.2 Member Function Documentation	9
4.3.2.1 update()	9
4.4 Gui::AHud Class Reference	10
4.4.1 Constructor & Destructor Documentation	11
4.4.1.1 ~AHud()	11
4.4.2 Member Function Documentation	11
4.4.2.1 display()	11
4.4.2.2 getType()	11
4.4.2.3 setPlayer()	11
4.4.2.4 setTile()	12
4.5 Gui::ANetwork Class Reference	12
4.5.1 Constructor & Destructor Documentation	13
4.5.1.1 ANetwork()	13
4.5.1.2 ~ANetwork()	13
4.5.2 Member Function Documentation	14
4.5.2.1 connectToServer()	14
4.5.2.2 getHostName()	14
4.5.2.3 getPort()	14
4.5.2.4 listenServer()	15
4.5.2.5 sendMessageServer()	15
4.5.2.6 setHostName()	15
4.5.2.7 setPort()	16
4.6 Gui::Decoration Class Reference	16
4.6.1 Constructor & Destructor Documentation	17
4.6.1.1 Decoration()	17

4.6.1.2 ~Decoration()	17
4.6.2 Member Function Documentation	17
4.6.2.1 display()	17
4.6.2.2 displayTree()	17
4.6.2.3 getGenerationItem()	18
4.7 Gui::Egg Class Reference	18
4.7.1 Constructor & Destructor Documentation	19
4.7.1.1 Egg()	19
4.7.1.2 ~Egg()	19
4.7.2 Member Function Documentation	20
4.7.2.1 getId()	20
4.7.2.2 getPosition()	20
4.7.2.3 getState()	20
4.7.2.4 getTeam()	20
4.7.2.5 setId()	20
4.7.2.6 setPosition()	21
4.7.2.7 setState()	21
4.7.2.8 setTeam()	21
4.8 Gui::Engine Class Reference	22
4.8.1 Constructor & Destructor Documentation	22
4.8.1.1 Engine()	22
4.8.1.2 ~Engine()	23
4.8.2 Member Function Documentation	23
4.8.2.1 listenServer()	23
4.8.2.2 run()	23
4.8.2.3 sendMessageUpdate()	23
4.8.2.4 sendUpdateMapMessage()	24
4.8.2.5 updateMap()	24
4.9 Gui::Errors::Error Class Reference	24
4.9.1 Detailed Description	25
4.10 Gui::Event Class Reference	25
4.10.1 Constructor & Destructor Documentation	27
4.10.1.1 Event()	27
4.10.1.2 ~Event()	27
4.10.2 Member Function Documentation	27
4.10.2.1 changeActualPlayerPov()	27
4.10.2.2 changePlayer()	27
4.10.2.3 decreaseRenderDistance()	28
4.10.2.4 decreaseTimeUnit()	28
4.10.2.5 handleLeftClick()	28
4.10.2.6 increaseRenderDistance()	28
4.10.2.7 increaseTimeUnit()	28

4.10.2.8 listen()	29
4.10.2.9 moveDownCamera()	29
4.10.2.10 moveUpCamera()	29
4.10.2.11 selectPlayer()	29
4.10.2.12 switchDisplayDebug()	29
4.10.2.13 switchTileHudToGame()	29
4.10.3 Member Data Documentation	30
4.10.3.1 _eventsKeyDown	30
4.10.3.2 _eventsKeyPressed	30
4.10.3.3 _eventsMousePressed	30
4.11 Gui::GameData Class Reference	31
4.11.1 Constructor & Destructor Documentation	32
4.11.1.1 GameData()	32
4.11.1.2 ~GameData()	33
4.11.2 Member Function Documentation	33
4.11.2.1 addPlayerToTeam()	33
4.11.2.2 addServerEgg()	33
4.11.2.3 addTeam() [1/2]	33
4.11.2.4 addTeam() [2/2]	34
4.11.2.5 getIsEndGame()	34
4.11.2.6 getLastError()	34
4.11.2.7 getLastTick()	35
4.11.2.8 getLastTickMctCommand()	35
4.11.2.9 getMap()	35
4.11.2.10 getMapSize()	35
4.11.2.11 getNbBCTCommandReceived()	36
4.11.2.12 getPlayer()	36
4.11.2.13 getServerEggs()	37
4.11.2.14 getServerTick()	37
4.11.2.15 getTeam()	37
4.11.2.16 getTeamById()	38
4.11.2.17 getTeams()	38
4.11.2.18 getTile()	38
4.11.2.19 getTimeUnitFromServer()	39
4.11.2.20 removeServerEgg()	39
4.11.2.21 restartLastTick()	39
4.11.2.22 restartLastTickMctCommand()	39
4.11.2.23 setIsEndGame()	39
4.11.2.24 setLastError()	40
4.11.2.25 setMap()	40
4.11.2.26 setMapSize()	40
4.11.2.27 setNbBCTCommandReceived()	41

4.11.2.28 setServerTick()	41
4.11.2.29 setTile()	41
4.11.2.30 setTimeUnitFromServer()	42
4.12 Gui::Errors::GuiGameDataException Class Reference	42
4.12.1 Detailed Description	43
4.12.2 Constructor & Destructor Documentation	43
4.12.2.1 GuiGameDataException()	43
4.13 Gui::GUIUpdater Class Reference	43
4.13.1 Constructor & Destructor Documentation	45
4.13.1.1 GUIUpdater()	45
4.13.2 Member Function Documentation	46
4.13.2.1 increaseColorIndex()	46
4.13.2.2 update()	46
4.13.2.3 updateCommandParameter()	46
4.13.2.4 updateEggDeath()	46
4.13.2.5 updateEggLaidByPlayer()	47
4.13.2.6 updateEndOfGame()	47
4.13.2.7 updateMapContent()	47
4.13.2.8 updateMapSize()	48
4.13.2.9 updateMessageFromServer()	48
4.13.2.10 updatePlayerBorn()	48
4.13.2.11 updatePlayerBroadcast()	48
4.13.2.12 updatePlayerDeath()	49
4.13.2.13 updatePlayerEggLaying()	49
4.13.2.14 updatePlayerEndIncantation()	49
4.13.2.15 updatePlayerExpulsion()	50
4.13.2.16 updatePlayerInventory()	50
4.13.2.17 updatePlayerLevel()	50
4.13.2.18 updatePlayerPosition()	50
4.13.2.19 updatePlayerRessourceCollecting()	51
4.13.2.20 updatePlayerRessourceDropping()	51
4.13.2.21 updatePlayerStartIncantation()	51
4.13.2.22 updateTeamMember()	52
4.13.2.23 updateTeamNames()	52
4.13.2.24 updateTimeUnitModification()	52
4.13.2.25 updateTimeUnitRequest()	52
4.13.2.26 updateUnknownMessage()	53
4.13.3 Member Data Documentation	53
4.13.3.1 _updateMap	53
4.14 Gui::Errors::GuiUpdaterException Class Reference	54
4.14.1 Detailed Description	54
4.14.2 Constructor & Destructor Documentation	54

4.14.2.1 GuiUpdaterException()	54
4.15 Gui::HudGame Class Reference	55
4.15.1 Constructor & Destructor Documentation	56
4.15.1.1 HudGame()	56
4.15.1.2 ~HudGame()	56
4.15.2 Member Function Documentation	56
4.15.2.1 display()	57
4.16 Gui::HudPlayer Class Reference	57
4.16.1 Constructor & Destructor Documentation	58
4.16.1.1 HudPlayer()	58
4.16.1.2 ~HudPlayer()	58
4.16.2 Member Function Documentation	59
4.16.2.1 display()	59
4.17 Gui::HudTile Class Reference	59
4.17.1 Constructor & Destructor Documentation	60
4.17.1.1 HudTile()	60
4.17.1.2 ~HudTile()	61
4.17.2 Member Function Documentation	61
4.17.2.1 display()	61
4.17.2.2 displayNbEggs()	61
4.17.2.3 displayNbPlayers()	61
4.18 Gui::Errors::!Error Class Reference	62
4.18.1 Member Function Documentation	62
4.18.1.1 what()	62
4.19 Gui::IEvent Class Reference	62
4.19.1 Constructor & Destructor Documentation	63
4.19.1.1 IEvent()	63
4.19.1.2 ~IEvent()	63
4.19.2 Member Function Documentation	63
4.19.2.1 listen()	63
4.19.2.2 setGameData()	63
4.19.2.3 setRender()	64
4.20 Gui::IGUIUpdater Class Reference	64
4.20.1 Member Function Documentation	64
4.20.1.1 update()	64
4.21 Gui::IHud Class Reference	65
4.21.1 Member Enumeration Documentation	65
4.21.1.1 TypeScene	65
4.21.2 Constructor & Destructor Documentation	65
4.21.2.1 ~IHud()	65
4.21.3 Member Function Documentation	65
4.21.3.1 display()	66

4.21.3.2	getType()	66
4.21.3.3	setPlayer()	66
4.21.3.4	setTile()	66
4.22	Gui::INetwork Class Reference	67
4.22.1	Constructor & Destructor Documentation	67
4.22.1.1	~INetwork()	67
4.22.2	Member Function Documentation	67
4.22.2.1	connectToServer()	67
4.22.2.2	getHostName()	68
4.22.2.3	getPort()	68
4.22.2.4	listenServer()	68
4.22.2.5	sendMessageServer()	68
4.22.2.6	setHostName()	69
4.22.2.7	setPort()	69
4.23	Gui::Inventory Class Reference	69
4.23.1	Constructor & Destructor Documentation	71
4.23.1.1	Inventory() [1/2]	71
4.23.1.2	Inventory() [2/2]	71
4.23.1.3	~Inventory()	71
4.23.2	Member Function Documentation	71
4.23.2.1	addResource()	72
4.23.2.2	getDeraumere()	72
4.23.2.3	getFood()	72
4.23.2.4	getLinemate()	72
4.23.2.5	getMendiane()	73
4.23.2.6	getPhiras()	73
4.23.2.7	getRessources()	73
4.23.2.8	getSibur()	73
4.23.2.9	getThystame()	74
4.23.2.10	removeResource()	74
4.23.2.11	setDeraumere()	74
4.23.2.12	setFood()	74
4.23.2.13	setLinemate()	75
4.23.2.14	setMendiane()	75
4.23.2.15	setPhiras()	75
4.23.2.16	setRessources()	76
4.23.2.17	setSibur()	76
4.23.2.18	setThystame()	76
4.24	Gui::IServerParser Class Reference	76
4.24.1	Member Function Documentation	77
4.24.1.1	parse()	77
4.25	Gui::Network Class Reference	77

4.25.1 Constructor & Destructor Documentation	79
4.25.1.1 Network()	79
4.25.1.2 ~Network()	79
4.25.2 Member Function Documentation	79
4.25.2.1 connectToServer()	80
4.25.2.2 listenServer()	80
4.25.2.3 readInfoServer()	80
4.25.2.4 selectServer()	80
4.25.2.5 sendMessageServer()	80
4.26 Gui::Errors::NetworkException Class Reference	81
4.26.1 Detailed Description	82
4.26.2 Constructor & Destructor Documentation	82
4.26.2.1 NetworkException()	82
4.27 Gui::ParseCommandLine Class Reference	82
4.27.1 Constructor & Destructor Documentation	83
4.27.1.1 ParseCommandLine()	83
4.27.1.2 ~ParseCommandLine()	83
4.27.2 Member Function Documentation	83
4.27.2.1 getHostName()	83
4.27.2.2 getPort()	84
4.27.2.3 parseFlags()	84
4.28 Gui::Errors::ParseCommandLineException Class Reference	84
4.28.1 Detailed Description	85
4.28.2 Constructor & Destructor Documentation	85
4.28.2.1 ParseCommandLineException()	85
4.29 Gui::Player Class Reference	86
4.29.1 Constructor & Destructor Documentation	87
4.29.1.1 Player()	88
4.29.1.2 ~Player()	88
4.29.2 Member Function Documentation	88
4.29.2.1 getAnimationTimeElapsed()	88
4.29.2.2 getBroadcast()	88
4.29.2.3 getCenterPosition()	89
4.29.2.4 getCurrentFrame()	89
4.29.2.5 getId()	89
4.29.2.6 getLevel()	89
4.29.2.7 getOrientation()	90
4.29.2.8 getPosition()	90
4.29.2.9 getPosition3D()	90
4.29.2.10 getRotationFromOrientation()	90
4.29.2.11 getState()	91
4.29.2.12 getTeam()	91

4.29.2.13 restartAnimationTimeElapsed()	91
4.29.2.14 setBroadcast()	91
4.29.2.15 setCurrentFrame()	92
4.29.2.16 setId()	92
4.29.2.17 setLevel()	92
4.29.2.18 setOrientation()	92
4.29.2.19 setPosition()	93
4.29.2.20 setPosition3D()	93
4.29.2.21 setState()	93
4.29.2.22 setTeam()	94
4.29.3 Member Data Documentation	94
4.29.3.1 inventory	94
4.30 Gui::Render Class Reference	94
4.30.1 Constructor & Destructor Documentation	96
4.30.1.1 Render()	97
4.30.1.2 ~Render()	97
4.30.2 Member Function Documentation	97
4.30.2.1 changePlayerPOV()	97
4.30.2.2 changePOVToFirstPerson()	97
4.30.2.3 changePOVToSecondPerson()	98
4.30.2.4 changePOVToThirdPerson()	98
4.30.2.5 displayAnimations()	98
4.30.2.6 displayCursor()	98
4.30.2.7 displayDebug()	99
4.30.2.8 displayDeraumere()	99
4.30.2.9 displayEggs()	99
4.30.2.10 displayFood()	99
4.30.2.11 displayHUD()	100
4.30.2.12 displayLinemate()	100
4.30.2.13 displayMap()	100
4.30.2.14 displayMendiane()	100
4.30.2.15 displayPhiras()	100
4.30.2.16 displayPlayerBroadcast()	101
4.30.2.17 displayPlayerLevel()	101
4.30.2.18 displayPlayers()	101
4.30.2.19 displayResources()	102
4.30.2.20 displaySibur()	102
4.30.2.21 displayThystame()	102
4.30.2.22 displayTile()	102
4.30.2.23 displayWalkAnimation()	103
4.30.2.24 draw()	103
4.30.2.25 getCamera()	103

4.30.2.26	getCameraPlayerPov()	103
4.30.2.27	getCameraTile() [1/2]	104
4.30.2.28	getCameraTile() [2/2]	104
4.30.2.29	getCameraType()	104
4.30.2.30	getIsDebug()	104
4.30.2.31	getRenderDistance()	105
4.30.2.32	getTileModel()	105
4.30.2.33	getTimeUnit()	105
4.30.2.34	isCameraInPlayerPov()	105
4.30.2.35	isOpen()	106
4.30.2.36	LoadModels()	106
4.30.2.37	setCameraPlayerPov()	106
4.30.2.38	setCameraTile()	106
4.30.2.39	setCameraType()	107
4.30.2.40	setIsDebug()	107
4.30.2.41	setPlayerPov()	107
4.30.2.42	setRenderDistance()	107
4.30.2.43	setTimeUnit()	108
4.31	Gui::ServerParser Class Reference	108
4.31.1	Member Enumeration Documentation	109
4.31.1.1	ParseType	109
4.31.2	Constructor & Destructor Documentation	109
4.31.2.1	ServerParser()	109
4.31.2.2	~ServerParser()	110
4.31.3	Member Function Documentation	110
4.31.3.1	parse()	110
4.31.3.2	parseCommand()	110
4.31.3.3	parseInt()	111
4.31.3.4	parseListInt()	111
4.31.3.5	parseMessage()	111
4.31.3.6	parseString()	112
4.31.4	Member Data Documentation	112
4.31.4.1	_typesCommand	112
4.32	Gui::Errors::ServerParserException Class Reference	113
4.32.1	Detailed Description	114
4.32.2	Constructor & Destructor Documentation	114
4.32.2.1	ServerParserException()	114
4.33	Gui::Team Class Reference	114
4.33.1	Constructor & Destructor Documentation	116
4.33.1.1	Team()	116
4.33.1.2	~Team()	116
4.33.2	Member Function Documentation	117

4.33.2.1 addEgg()	117
4.33.2.2 addPlayer()	117
4.33.2.3 getEgg()	117
4.33.2.4 getEggModel()	118
4.33.2.5 getEggs()	118
4.33.2.6 getName()	118
4.33.2.7 getPlayer()	118
4.33.2.8 getPlayerBoundingBoxes()	119
4.33.2.9 getPlayerColor()	119
4.33.2.10 getPlayerModel()	119
4.33.2.11 getPlayerModelAnimation()	120
4.33.2.12 getPlayerModelHitbox()	120
4.33.2.13 getPlayerPositionIn3DSpace()	120
4.33.2.14 getPlayers()	120
4.33.2.15 isPlayerHit()	121
4.33.2.16 removeEgg()	121
4.33.2.17 removePlayer()	121
4.33.2.18 rotateBoundingBoxByOrientation()	123
4.33.2.19 setEggModelPath()	123
4.33.2.20 setName()	124
4.33.2.21 setPlayerModelPath()	124
4.34 Gui::Tile Class Reference	124
4.34.1 Constructor & Destructor Documentation	125
4.34.1.1 Tile() [1/2]	125
4.34.1.2 Tile() [2/2]	125
4.34.1.3 ~Tile()	126
4.34.2 Member Function Documentation	126
4.34.2.1 getPosition()	126
4.34.2.2 getPositionIn3DSpace()	126
4.34.2.3 getTileBoundingBoxes()	126
4.34.2.4 getTileModelHitbox()	127
4.34.2.5 isTileHit()	127
4.34.2.6 setPosition()	128
4.34.3 Member Data Documentation	128
4.34.3.1 inventory	128
4.35 Gui::UserCamera Class Reference	128
4.35.1 Constructor & Destructor Documentation	129
4.35.1.1 UserCamera()	129
4.35.1.2 ~UserCamera()	130
4.35.2 Member Function Documentation	130
4.35.2.1 getCamera()	130
4.35.2.2 getFovy()	130

4.35.2.3 getPlayerId()	130
4.35.2.4 getPosition()	131
4.35.2.5 getTarget()	131
4.35.2.6 getTilePos()	131
4.35.2.7 getType()	131
4.35.2.8 getUp()	132
4.35.2.9 isPlayerPov()	132
4.35.2.10 setFovy()	132
4.35.2.11 setPlayerId()	132
4.35.2.12 setPosition()	133
4.35.2.13 setTarget()	133
4.35.2.14 setTilePos()	133
4.35.2.15 setType()	134
4.35.2.16 setUp()	134
5 File Documentation	135
5.1 Assets.hpp	135
5.2 Colors.hpp	136
5.3 Config.hpp	137
5.4 Engine.hpp	137
5.5 AError.hpp	138
5.6 Error.hpp	138
5.7 IError.hpp	139
5.8 AEvent.hpp	139
5.9 Event.hpp	140
5.10 IEvent.hpp	141
5.11 Egg.hpp	141
5.12 GameData.hpp	142
5.13 Inventory.hpp	143
5.14 Player.hpp	144
5.15 Team.hpp	145
5.16 Tile.hpp	146
5.17 AGUIUpdater.hpp	147
5.18 GUIUpdater.hpp	147
5.19 IGUIUpdater.hpp	149
5.20 AHud.hpp	149
5.21 HudGame.hpp	150
5.22 HudPlayer.hpp	150
5.23 HudTile.hpp	151
5.24 IHud.hpp	152
5.25 ANetwork.hpp	152
5.26 INetwork.hpp	153

5.27 Network.hpp	153
5.28 IServerParser.hpp	154
5.29 ParseCommandLine.hpp	154
5.30 ServerParser.hpp	155
5.31 Decoration.hpp	156
5.32 Render.hpp	156
5.33 UserCamera.hpp	158
5.34 Types.hpp	159
Index	161

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Gui::Decoration	16
Gui::Egg	18
Gui::Engine	22
std::exception	
Gui::Errors::IError	62
Gui::Errors::AError	7
Gui::Errors::Error	24
Gui::Errors::GuiGameDataException	42
Gui::Errors::GuiUpdaterException	54
Gui::Errors::NetworkException	81
Gui::Errors::ParseCommandLineException	84
Gui::Errors::ServerParserException	113
Gui::GameData	31
Gui::IEvent	62
Gui::AEvent	8
Gui::Event	25
Gui::IGUIUpdater	64
Gui::AGUIUpdater	8
Gui::GUIUpdater	43
Gui::IHud	65
Gui::AHud	10
Gui::HudGame	55
Gui::HudPlayer	57
Gui::HudTile	59
Gui::INetwork	67
Gui::ANetwork	12
Gui::Network	77
Gui::Inventory	69
Gui::IServerParser	76
Gui::ServerParser	108
Gui::ParseCommandLine	82
Gui::Player	86
Gui::Render	94
Gui::Team	114
Gui::Tile	124
Gui::UserCamera	128

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::AEvent	8
Gui::AGUIUpdater	8
Gui::AHud	10
Gui::ANetwork	12
Gui::Decoration	16
Gui::Egg	18
Gui::Engine	22
Gui::Errors::Error	
Base class for argument-related errors	24
Gui::Event	25
Gui::GameData	31
Gui::Errors::GuiGameDataException	
Error class for GameData errors	42
Gui::GUIUpdater	43
Gui::Errors::GuiUpdaterException	
Error class for GUIUpdater errors	54
Gui::HudGame	55
Gui::HudPlayer	57
Gui::HudTile	59
Gui::Errors::IError	62
Gui::IEvent	62
Gui::IGUIUpdater	64
Gui::IHud	65
Gui::INetwork	67
Gui::Inventory	69
Gui::IServerParser	76
Gui::Network	77
Gui::Errors::NetworkException	
Error class for network errors	81
Gui::ParseCommandLine	82
Gui::Errors::ParseCommandLineException	
Error class for <code>parseCommandLine</code> errors	84

Gui::Player	86
Gui::Render	94
Gui::ServerParser	108
Gui::Errors::ServerParserException Error class for network errors	113
Gui::Team	114
Gui::Tile	124
Gui::UserCamera	128

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	135
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Colors.hpp	136
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Config.hpp	137
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Types.hpp	159
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Engine/Engine.hpp	137
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/AError.hpp	138
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp	138
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/IError.hpp	139
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/AEvent.hpp	139
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/Event.hpp	140
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/IEvent.hpp	141
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Egg.hpp	141
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/GameData.hpp	142
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Inventory.hpp	143
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Player.hpp	144
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Team.hpp	145
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameDatas/Tile.hpp	146
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/AGUIUpdater.hpp	147
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/GUIUpdater.hpp	147
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdater/IGUIUpdater.hpp	149
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/AHud.hpp	149
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame.hpp	150
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudPlayer.hpp	150
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudTile.hpp	151
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/IHud.hpp	152
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/ANetwork.hpp	152
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/INetwork.hpp	153
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/Network.hpp	153
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/IServerParser.hpp	154
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ParseCommandLine.hpp	154
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ServerParser.hpp	155
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Decoration.hpp	156
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Render.hpp	156
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/UserCamera.hpp	158

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`
The error 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](#).

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::AEvent Class Reference

Inheritance diagram for Gui::AEvent:

4.3 Gui::AGUIUpdater Class Reference

Inheritance diagram for Gui::AGUIUpdater:

Collaboration diagram for Gui::AGUIUpdater:

Public Member Functions

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

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

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

Protected Attributes

- std::shared_ptr< [GameData](#) > `_gameData`
- std::shared_ptr< [INetwork](#) > `_network`

4.3.1 Constructor & Destructor Documentation**4.3.1.1 AGUIUpdater()**

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

Construct a new [AGUIUpdater](#) object.

Parameters

<i>gameData</i>	The GUI GameData to update.
<i>network</i>	The network to send commands to the server.

4.3.2 Member Function Documentation**4.3.2.1 update()**

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

Update the GUI [GameData](#).

Implements [Gui::IGUIUpdater](#).

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

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

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

4.4 Gui::AHud Class Reference

Inheritance diagram for Gui::AHud:

Collaboration diagram for Gui::AHud:

Public Member Functions

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

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

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

Protected Attributes

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

Additional Inherited Members

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

- enum [TypeScene](#) { [GAME](#) , [POV_PLAYER](#) , [END_GAME](#) , [TILE](#) }
Hud enum for the different scenes.

4.4.1 Constructor & Destructor Documentation

4.4.1.1 ~AHud()

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

Destroy the [AHud](#) object.

4.4.2 Member Function Documentation

4.4.2.1 display()

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

Display Hud.

Implements [Gui::IHud](#).

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

4.4.2.2 getType()

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

Get the Type object.

Returns

TypeScene - Type of the scene.

Implements [Gui::IHud](#).

4.4.2.3 setPlayer()

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

Set the [Player](#) object.

Parameters

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

Implements [Gui::IHud](#).

4.4.2.4 setTile()

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

Set the [Tile](#) object.

Parameters

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

Implements [Gui::IHud](#).

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

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

4.5 Gui::ANetwork Class Reference

Inheritance diagram for Gui::ANetwork:

Collaboration diagram for Gui::ANetwork:

Public Member Functions

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

Public Member Functions inherited from Gui::INetwork

- virtual `~INetwork()`=default
Destroy the INetwork object.
- virtual void `setPort` (int port)=0
Set the port object.
- virtual void `setHostName` (const std::string &hostName)=0
Set the host name object.
- virtual int `getPort` () const =0
Get the host name object.
- virtual std::string `getHostName` () const =0
Get the host name object.
- virtual void `connectToServer` ()=0
Connect to the server.
- virtual const std::string `listenServer` ()=0
Listen to the server.
- virtual void `sendMessageServer` (const std::string &message)=0
Send a message to the server.

Protected Attributes

- int `_port`
- std::string `_hostName`

4.5.1 Constructor & Destructor Documentation

4.5.1.1 ANetwork()

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

Construct a new [ANetwork](#) object.

Parameters

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

4.5.1.2 ~ANetwork()

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

Destroy the [ANetwork](#) object.

4.5.2 Member Function Documentation

4.5.2.1 connectToServer()

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

Connect to the server.

Exceptions

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

Implements [Gui::INetwork](#).

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

4.5.2.2 getHostName()

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

Get the host name object.

Returns

std::string Host of the server.

Implements [Gui::INetwork](#).

4.5.2.3 getPort()

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

Get the host name object.

Returns

std::string Host of the server.

Implements [Gui::INetwork](#).

4.5.2.4 listenServer()

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

Listen the server and return it message.

Returns

std::string - Message of the server.

Implements [Gui::INetwork](#).

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

4.5.2.5 sendMessageServer()

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

Send a message to the Server.

Parameters

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

Implements [Gui::INetwork](#).

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

4.5.2.6 setHostName()

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

Set the host name object.

Parameters

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

Implements [Gui::INetwork](#).

4.5.2.7 setPort()

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

Set the port object.

Parameters

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

Exceptions

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

Implements [Gui::INetwork](#).

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

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

4.6 Gui::Decoration Class Reference

Public Member Functions

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

Private Member Functions

- void [displayTree](#) (size_t i, size_t j, Vector3 posTile)
Display Trees.

Private Attributes

- Model [_treeModel](#)
- Map< bool > [_mapTree](#)
- std::pair< std::size_t, std::size_t > [_mapSize](#)

4.6.1 Constructor & Destructor Documentation

4.6.1.1 Decoration()

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

Construct a new [Decoration](#) object.

4.6.1.2 ~Decoration()

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

Destroy the [Decoration](#) object.

4.6.2 Member Function Documentation

4.6.2.1 display()

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

Display decorations.

Parameters

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

4.6.2.2 displayTree()

```
void Gui::Decoration::displayTree (
    size_t i,
    size_t j,
    Vector3 posTile ) [private]
```

Display Trees.

Parameters

<i>i</i>	Index width in map.
<i>j</i>	Index height in map.
<i>posTile</i>	Position of a tile.

4.6.2.3 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

4.7 Gui::Egg Class Reference

Public Types

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

Public Member Functions

- [Egg](#) (size_t id, const std::string &team, std::pair< std::size_t, std::size_t > position)
Construct a new [Egg](#) object.
- [~Egg](#) ()
Destroy the [Egg](#) object.
- std::size_t [getId](#) () const
*Get the *Id* object.*
- std::string [getTeam](#) () const
*Get the *Team* object.*
- std::pair< std::size_t, std::size_t > [getPosition](#) () const

Get the Position object.

- void [setId](#) (std::size_t id)

Set the id object.

- void [setTeam](#) (const std::string &team)

Set the team object.

- void [setPosition](#) (std::pair< std::size_t, std::size_t > position)

Set the position object.

- void [setState](#) (EggState state)

Set the state object.

- EggState [getState](#) () const

Get the state object.

Private Attributes

- std::size_t **_id**
- std::string **_team**
- std::pair< std::size_t, std::size_t > **_position**
- EggState **_state**

4.7.1 Constructor & Destructor Documentation

4.7.1.1 Egg()

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

Construct a new [Egg](#) object.

Parameters

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

Note

The egg is created when a player lays an egg.

The constructor starts the egg animation if implemented.

4.7.1.2 ~Egg()

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

Destroy the [Egg](#) object.

Note

The destructor starts the egg animation if implemented.

4.7.2 Member Function Documentation

4.7.2.1 getId()

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

Get the Id object.

Returns

std::size_t Id of the egg.

4.7.2.2 getPosition()

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

Get the Position object.

Returns

std::pair<std::size_t, std::size_t> Position of the egg.

4.7.2.3 getState()

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

Get the state object.

Returns

EggState State of the egg.

4.7.2.4 getTeam()

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

Get the [Team](#) object.

Returns

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

4.7.2.5 setId()

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

Set the id object.

Parameters

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

4.7.2.6 setPosition()

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

Set the position object.

Parameters

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

4.7.2.7 setState()

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

Set the state object.

Parameters

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

4.7.2.8 setTeam()

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

Set the team object.

Parameters

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

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

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

4.8 Gui::Engine Class Reference

Public Member Functions

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

Private Member Functions

- void [listenServer](#) ()
Listen the server and update [Engine](#) with its commands.
- void [sendMessageUpdate](#) ()
Send Messages to server at each tick.
- void [updateMap](#) ()
Update the map at each 20 / ticks units.
- void [sendUpdateMapMessage](#) ()
Send update map message.

Private Attributes

- std::unique_ptr< [IServerParser](#) > [_parser](#)
!< Parser class for server's command.
- std::shared_ptr< [INetwork](#) > [_network](#)
[Network](#) class to connect to the server.
- std::shared_ptr< [Render](#) > [_render](#)
[Render](#) class to draw the scene.
- std::unique_ptr< [IEvent](#) > [_event](#)
[Event](#) class to listen the user's inputs.
- std::shared_ptr< [GameData](#) > [_gameData](#)
[GameData](#) class to store the game's data.
- std::unique_ptr< [IGUIUpdater](#) > [_guiUpdater](#)
[GUIUpdater](#) class to update the GUI.

4.8.1 Constructor & Destructor Documentation

4.8.1.1 Engine()

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

Construct a new [Engine](#) object.

Parameters

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

4.8.1.2 ~Engine()

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

Destroy the [Engine](#) object.

4.8.2 Member Function Documentation

4.8.2.1 listenServer()

```
void Gui::Engine::listenServer ( ) [private]
```

Listen the server and update [Engine](#) with its commands.

4.8.2.2 run()

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

Run the engine loop.

4.8.2.3 sendMessageUpdate()

```
void Gui::Engine::sendMessageUpdate ( ) [private]
```

Send Messages to server at each tick.

- The content of tiles.
- The size of the map.
- The names of teams.
- The tick.

4.8.2.4 sendUpdateMapMessage()

```
void Gui::Engine::sendUpdateMapMessage ( ) [private]
```

Send update map message.

4.8.2.5 updateMap()

```
void Gui::Engine::updateMap ( ) [private]
```

Update the map at each 20 / ticks units.

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

4.9 Gui::Errors::Error Class Reference

Base class for argument-related errors.

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

Inheritance diagram for Gui::Errors::Error:

Collaboration diagram for Gui::Errors::Error:

Additional Inherited Members

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

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

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

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

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

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

4.9.1 Detailed Description

Base class for argument-related errors.

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

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

4.10 Gui::Event Class Reference

Inheritance diagram for `Gui::Event`:

Collaboration diagram for `Gui::Event`:

Public Member Functions

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

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

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

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

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

Private Member Functions

- void [moveUpCamera](#) ()
Move up the camera.
- void [moveDownCamera](#) ()
Move down the camera.
- void [switchDisplayDebug](#) ()
Switch on/off the debug display.
- void [setFreeCam](#) ()
Set the free camera.
- void [handleLeftClick](#) ()
Handle the left click.
- void [handleRightClick](#) ()
Handle the right click.
- void [selectPlayer](#) ()
Select the player pov.
- void [selectTile](#) ()
Select the tile.
- void [changePlayer](#) (bool turn)
Change the player.
- void [changeActualPlayerPov](#) ()
Change the actual player point of view.
- void [switchTileHudToGame](#) ()
Change the Hud of [Tile](#) to Game.
- void [increaseRenderDistance](#) ()
Increase the render distance.
- void [decreaseRenderDistance](#) ()
Decrease the render distance.
- void [increaseTimeUnit](#) ()
Increase the time unit.
- void [decreaseTimeUnit](#) ()
Decrease the time unit.

Private Attributes

- `std::unordered_map< KeyboardKey, std::function< void()> > _eventsKeyDown`
Map for events by down key.
- `std::unordered_map< KeyboardKey, std::function< void()> > _eventsKeyPressed`
Map for events by pressing key.
- `std::unordered_map< MouseButton, std::function< void()> > _eventsMousePressed`
Map for events by pressing mouse.

Additional Inherited Members

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

- `std::shared_ptr< Render > _render`
[Render](#) class to draw scene.
- `std::shared_ptr< GameData > _gameData`
[GameData](#) class to contain scene.

4.10.1 Constructor & Destructor Documentation

4.10.1.1 Event()

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

Construct a new [Event](#) object.

4.10.1.2 ~Event()

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

Destroy the [Event](#) object.

4.10.2 Member Function Documentation

4.10.2.1 changeActualPlayerPov()

```
void Gui::Event::changeActualPlayerPov ( ) [private]
```

Change the actual player point of view.

4.10.2.2 changePlayer()

```
void Gui::Event::changePlayer (
    bool turn ) [private]
```

Change the player.

Parameters

<i>turn</i>	Turn to select the player.
-------------	----------------------------

4.10.2.3 decreaseRenderDistance()

```
void Gui::Event::decreaseRenderDistance ( ) [private]
```

Decrease the render distance.

4.10.2.4 decreaseTimeUnit()

```
void Gui::Event::decreaseTimeUnit ( ) [private]
```

Decrease the time unit.

4.10.2.5 handleLeftClick()

```
void Gui::Event::handleLeftClick ( ) [private]
```

Handle the left click.

4.10.2.6 increaseRenderDistance()

```
void Gui::Event::increaseRenderDistance ( ) [private]
```

Increase the render distance.

4.10.2.7 increaseTimeUnit()

```
void Gui::Event::increaseTimeUnit ( ) [private]
```

Increase the time unit.

4.10.2.8 listen()

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

Listen the user's events.

Implements [Gui::AEvent](#).

4.10.2.9 moveDownCamera()

```
void Gui::Event::moveDownCamera ( ) [private]
```

Move down the camera.

4.10.2.10 moveUpCamera()

```
void Gui::Event::moveUpCamera ( ) [private]
```

Move up the camera.

4.10.2.11 selectPlayer()

```
void Gui::Event::selectPlayer ( ) [private]
```

Select the player pov.

4.10.2.12 switchDisplayDebug()

```
void Gui::Event::switchDisplayDebug ( ) [private]
```

Switch on/off the debug display.

4.10.2.13 switchTileHudToGame()

```
void Gui::Event::switchTileHudToGame ( ) [private]
```

Change the Hud of [Tile](#) to Game.

4.10.3 Member Data Documentation

4.10.3.1 `_eventsKeyDown`

```
std::unordered_map<KeyboardKey, std::function<void()> > Gui::Event::_eventsKeyDown [private]
```

Initial value:

```
=
{
    {KEY_SPACE, [this]() {moveUpCamera();}},
    {KEY_LEFT_SHIFT, [this]() {moveDownCamera();}},
}
```

Map for events by down key.

4.10.3.2 `_eventsKeyPressed`

```
std::unordered_map<KeyboardKey, std::function<void()> > Gui::Event::_eventsKeyPressed [private]
```

Initial value:

```
=
{
    {KEY_THREE, [this]() {switchDisplayDebug();}},
    {KEY_F3, [this]() {switchDisplayDebug();}},
    {KEY_SPACE, [this]() {setFreeCam();}},
    {KEY_R, [this]() {switchTileHudToGame();}},
    {KEY_J, [this]() {increaseRenderDistance();}},
    {KEY_K, [this]() {decreaseRenderDistance();}},
    {KEY_F5, [this]() {changeActualPlayerPov();}},
    {KEY_FOUR, [this]() {changeActualPlayerPov();}},
    {KEY_KP_ADD, [this]() {increaseTimeUnit();}},
    {KEY_KP_SUBTRACT, [this]() {decreaseTimeUnit();}},
}
```

Map for events by pressing key.

4.10.3.3 `_eventsMousePressed`

```
std::unordered_map<MouseButton, std::function<void()> > Gui::Event::_eventsMousePressed [private]
```

Initial value:

```
=
{
    {MOUSE_BUTTON_LEFT, [this]() {handleLeftClick();}},
    {MOUSE_BUTTON_RIGHT, [this]() {handleRightClick();}},
}
```

Map for events by pressing mouse.

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

4.11 Gui::GameData Class Reference

Public Types

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

Public Member Functions

- [GameData](#) ()
Construct a new [GameData](#) object.
- [~GameData](#) ()=default
Destroy the [GameData](#) object.
- std::vector< [Gui::Team](#) > & [getTeams](#) ()
Get the [Teams](#) object.
- [Gui::Team](#) & [getTeam](#) (const std::string &name)
Get a [Team](#) object.
- void [addTeam](#) (const [Gui::Team](#) &team)
Add a team to the game.
- void [addTeam](#) (const std::string &name, const std::string &playerModelPath, const std::string &eggModelPath, Color playerColor)
Add a team to the game.
- void [addPlayerToTeam](#) (const std::string &teamName, const [Gui::Player](#) &player)
Add a player to a team.
- [Gui::Player](#) & [getPlayer](#) (size_t id)
Get a player object.
- Map< [Gui::Tile](#) > & [getMap](#) ()
Get the Map object.
- void [setMap](#) (const Map< [Gui::Tile](#) > &map)
Set the Map object.
- void [setMapSize](#) (size_t x, size_t y)
Set the Map Size object.
- std::pair< size_t, size_t > [getMapSize](#) () const
Get the Map Size object.
- [Gui::Tile](#) & [getTile](#) (size_t x, size_t y)
Get a [Tile](#) object.
- void [setTile](#) (const [Gui::Tile](#) &tile)
Set the [Tile](#) object.
- void [restartLastTick](#) (void)
Restart the last tick clock.
- void [setServerTick](#) (std::size_t tick)
Set the Server Tick object.
- clock_t [getLastTick](#) () const
Get the Last Tick object.
- std::size_t [getServerTick](#) () const
Get the Server Tick object.
- void [setIsEndGame](#) (bool isEndGame)
Set the IsEnd Game object.
- bool [getIsEndGame](#) () const
Get the IsEnd Game object.
- void [setLastError](#) (const std::string &error)

- Set the Last Error object.*

 - `std::string getLastError ()` const

Get the Last Error object.
- `Team & getTeamById (std::size_t id)`

Get the Team From Player object.
- `TimeUnitState getTimeUnitFromServer ()` const

Get the Time Unit From Server object.
- `void setTimeUnitFromServer (TimeUnitState timeUnitFromServer)`

Set the Time Unit From Server object.
- `std::vector< Gui::Egg > & getServerEggs ()`

Get the Server Eggs object.
- `void addServerEgg (const Gui::Egg &egg)`

Add an egg to the server ones.
- `void removeServerEgg (std::size_t id)`

Remove an egg from the server ones.
- `void setNbBCTCommandReceived (std::size_t nb)`

Set the number of bct command received.
- `std::size_t getNbBCTCommandReceived ()` const

Get the number of bct command received.
- `void restartLastTickMctCommand ()`

Restart the last tick mct command clock.
- `clock_t getLastTickMctCommand ()` const

Get the Last Tick mct command object.

Private Attributes

- `std::vector< Gui::Team > _teams`
- `Map< Gui::Tile > _map`
- `std::size_t _serverTick`
- `clock_t _lastTick`
- `bool _isEndGame`
- `std::size_t _nbBCTCommandReceived`
- `clock_t _lastTickMctCommand`
- `std::string _lastError`
- `TimeUnitState _timeUnitFromServer`
- `std::vector< Gui::Egg > _serverEggs`

4.11.1 Constructor & Destructor Documentation

4.11.1.1 GameData()

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

Construct a new `GameData` object.

4.11.1.2 ~GameData()

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

Destroy the [GameData](#) object.

4.11.2 Member Function Documentation

4.11.2.1 addPlayerToTeam()

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

Add a player to a team.

Parameters

<i>teamName</i>	Name of the team.
<i>player</i>	Player to add.

4.11.2.2 addServerEgg()

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

Add an egg to the server ones.

Parameters

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

4.11.2.3 addTeam() [1/2]

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

Add a team to the game.

Parameters

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

4.11.2.4 addTeam() [2/2]

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

Add a team to the game.

Parameters

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

4.11.2.5 getIsEndGame()

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

Get the IsEnd Game object.

Returns

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

4.11.2.6 getLastError()

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

Get the Last Error object.

Returns

std::string - Last error message.

4.11.2.7 getLastTick()

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

Get the Last Tick object.

Returns

clock_t - Last Tick.

4.11.2.8 getLastTickMctCommand()

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

Get the Last Tick mct command object.

Returns

clock_t - Last Tick Mct command.

4.11.2.9 getMap()

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

Get the Map object.

Returns

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

4.11.2.10 getMapSize()

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

Get the Map Size object.

Returns

std::pair<size_t, size_t> Size of the map.

4.11.2.11 `getNbBCTCommandReceived()`

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

Get the number of bct command received.

Returns

std::size_t - Number of bct command received.

4.11.2.12 `getPlayer()`

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

Get a player object.

Parameters

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

4.11.2.13 getServerEggs()

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

Get the Server Eggs object.

Returns

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

4.11.2.14 getServerTick()

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

Get the Server Tick object.

Returns

std::size_t - Server Tick.

4.11.2.15 getTeam()

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

Get a [Team](#) object.

Parameters

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

Returns

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

4.11.2.16 getTeamById()

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

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

Parameters

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

Returns

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

4.11.2.17 getTeams()

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

Get the Teams object.

Returns

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

4.11.2.18 getTile()

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

Get a [Tile](#) object.

Parameters

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

Returns

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

4.11.2.19 getTimeUnitFromServer()

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

Get the Time Unit From Server object.

Returns

true - The time unit has changed.

4.11.2.20 removeServerEgg()

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

Remove an egg from the server ones.

Parameters

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

4.11.2.21 restartLastTick()

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

Restart the last tick clock.

4.11.2.22 restartLastTickMctCommand()

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

Restart the last tick mct command clock.

4.11.2.23 setIsEndGame()

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

Set the IsEnd Game object.

Parameters

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

4.11.2.24 setLastError()

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

Set the Last Error object.

Parameters

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

4.11.2.25 setMap()

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

Set the Map object.

Parameters

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

4.11.2.26 setMapSize()

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

Set the Map Size object.

Parameters

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

Note

This method resizes the map.

4.11.2.27 setNbBCTCommandReceived()

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

Set the number of bct command received.

Parameters

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

4.11.2.28 setServerTick()

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

Set the Server Tick object.

Parameters

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

4.11.2.29 setTile()

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

Set the [Tile](#) object.

Parameters

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

4.11.2.30 setTimeUnitFromServer()

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

Set the Time Unit From Server object.

Parameters

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

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

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

4.12 Gui::Errors::GuiGameDataException Class Reference

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

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

Inheritance diagram for Gui::Errors::GuiGameDataException:

Collaboration diagram for Gui::Errors::GuiGameDataException:

Public Member Functions

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

Public Member Functions inherited from Gui::Errors::AError

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

Public Member Functions inherited from Gui::Errors::IError

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

Additional Inherited Members

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

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

4.12.1 Detailed Description

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

4.12.2 Constructor & Destructor Documentation

4.12.2.1 GuiGameDataException()

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

Constructor for [GuiGameDataException](#).

Parameters

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

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

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

4.13 Gui::GUIUpdater Class Reference

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

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

Public Member Functions

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

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

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

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

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

Private Member Functions

- void [updateMapSize](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) map size.
- void [updateMapContent](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) map content.
- void [updateTeamNames](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) team names.
- void [updateTeamMember](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) team member.
- void [updatePlayerPosition](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player position.
- void [updatePlayerLevel](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player level.
- void [updatePlayerInventory](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player inventory.
- void [updatePlayerExpulsion](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player expulsion.
- void [updatePlayerBroadcast](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player broadcast.
- void [updatePlayerStartIncantation](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player start incantation.
- void [updatePlayerEndIncantation](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player end incantation.
- void [updatePlayerEggLaying](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player egg laying.
- void [updatePlayerRessourceDropping](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player ressource dropping.
- void [updatePlayerRessourceCollecting](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player ressource collecting.
- void [updatePlayerDeath](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player death.
- void [updateEggLaidByPlayer](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) egg laid by player.

- void [updatePlayerBorn](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) player born.
- void [updateEggDeath](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) egg death.
- void [updateTimeUnitRequest](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) time unit request.
- void [updateTimeUnitModification](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) time unit modification.
- void [updateEndOfGame](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) end of game.
- void [updateMessageFromServer](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) message from server.
- void [updateUnknownMessage](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) unknow message.
- void [updateCommandParameter](#) (const std::vector< std::string > &data)
Update the GUI [GameData](#) command parameter.
- void [increaseColorIndex](#) ()
Increase the color index.

Private Attributes

- size_t [_colorIndex](#)
- std::unordered_map< std::string, std::function< void(std::vector< std::string >)> > [_updateMap](#)

Additional Inherited Members

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

- std::shared_ptr< [GameData](#) > [_gameData](#)
- std::shared_ptr< [INetwork](#) > [_network](#)

4.13.1 Constructor & Destructor Documentation

4.13.1.1 GUIUpdater()

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

Construct a new [GUIUpdater](#) object.

Parameters

<i>gameData</i>	The GUI GameData to update.
<i>network</i>	The network to send commands to the server.

4.13.2 Member Function Documentation

4.13.2.1 increaseColorIndex()

```
void Gui::GUIUpdater::increaseColorIndex ( ) [private]
```

Increase the color index.

4.13.2.2 update()

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

Update the GUI [GameData](#).

Parameters

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

Implements [Gui::AGUIUpdater](#).

4.13.2.3 updateCommandParameter()

```
void Gui::GUIUpdater::updateCommandParameter (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) command parameter.

Parameters

<i>data</i>	The data to update the GUI GameData command parameter.
-------------	--

4.13.2.4 updateEggDeath()

```
void Gui::GUIUpdater::updateEggDeath (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) egg death.

Parameters

<i>data</i>	The data to update the GUI GameData egg death.
-------------	--

4.13.2.5 updateEggLaidByPlayer()

```
void Gui::GUIUpdater::updateEggLaidByPlayer (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) egg laid by player.

Parameters

<i>data</i>	The data to update the GUI GameData egg laid by player.
-------------	---

4.13.2.6 updateEndOfGame()

```
void Gui::GUIUpdater::updateEndOfGame (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) end of game.

Parameters

<i>data</i>	The data to update the GUI GameData end of game.
-------------	--

4.13.2.7 updateMapContent()

```
void Gui::GUIUpdater::updateMapContent (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) map content.

Parameters

<i>data</i>	The data to update the GUI GameData map content.
-------------	--

4.13.2.8 updateMapSize()

```
void Gui::GUIUpdater::updateMapSize (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) map size.

Parameters

<i>data</i>	The data to update the GUI GameData map size.
-------------	---

4.13.2.9 updateMessageFromServer()

```
void Gui::GUIUpdater::updateMessageFromServer (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) message from server.

Parameters

<i>data</i>	The data to update the GUI GameData message from server.
-------------	--

4.13.2.10 updatePlayerBorn()

```
void Gui::GUIUpdater::updatePlayerBorn (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player born.

Parameters

<i>data</i>	The data to update the GUI GameData player born.
-------------	--

4.13.2.11 updatePlayerBroadcast()

```
void Gui::GUIUpdater::updatePlayerBroadcast (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player broadcast.

Parameters

<i>data</i>	The data to update the GUI GameData player broadcast.
-------------	---

4.13.2.12 updatePlayerDeath()

```
void Gui::GUIUpdater::updatePlayerDeath (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player death.

Parameters

<i>data</i>	The data to update the GUI GameData player death.
-------------	---

4.13.2.13 updatePlayerEggLaying()

```
void Gui::GUIUpdater::updatePlayerEggLaying (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player egg laying.

Parameters

<i>data</i>	The data to update the GUI GameData player egg laying.
-------------	--

4.13.2.14 updatePlayerEndIncantation()

```
void Gui::GUIUpdater::updatePlayerEndIncantation (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player end incantation.

Parameters

<i>data</i>	The data to update the GUI GameData player end incantation.
-------------	---

4.13.2.15 updatePlayerExpulsion()

```
void Gui::GUIUpdater::updatePlayerExpulsion (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player expulsion.

Parameters

<i>data</i>	The data to update the GUI GameData player expulsion.
-------------	---

4.13.2.16 updatePlayerInventory()

```
void Gui::GUIUpdater::updatePlayerInventory (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player inventory.

Parameters

<i>data</i>	The data to update the GUI GameData player inventory.
-------------	---

4.13.2.17 updatePlayerLevel()

```
void Gui::GUIUpdater::updatePlayerLevel (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player level.

Parameters

<i>data</i>	The data to update the GUI GameData player level.
-------------	---

4.13.2.18 updatePlayerPosition()

```
void Gui::GUIUpdater::updatePlayerPosition (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player position.

Parameters

<i>data</i>	The data to update the GUI GameData player position.
-------------	--

4.13.2.19 updatePlayerRessourceCollecting()

```
void Gui::GUIUpdater::updatePlayerRessourceCollecting (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player ressource collecting.

Parameters

<i>data</i>	The data to update the GUI GameData player ressource collecting.
-------------	--

4.13.2.20 updatePlayerRessourceDropping()

```
void Gui::GUIUpdater::updatePlayerRessourceDropping (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player ressource dropping.

Parameters

<i>data</i>	The data to update the GUI GameData player ressource dropping.
-------------	--

4.13.2.21 updatePlayerStartIncantation()

```
void Gui::GUIUpdater::updatePlayerStartIncantation (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) player start incantation.

Parameters

<i>data</i>	The data to update the GUI GameData player start incantation.
-------------	---

4.13.2.22 updateTeamMember()

```
void Gui::GUIUpdater::updateTeamMember (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) team member.

Parameters

<i>data</i>	The data to update the GUI GameData team member.
-------------	--

4.13.2.23 updateTeamNames()

```
void Gui::GUIUpdater::updateTeamNames (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) team names.

Parameters

<i>data</i>	The data to update the GUI GameData team names.
-------------	---

4.13.2.24 updateTimeUnitModification()

```
void Gui::GUIUpdater::updateTimeUnitModification (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) time unit modification.

Parameters

<i>data</i>	The data to update the GUI GameData time unit modification.
-------------	---

4.13.2.25 updateTimeUnitRequest()

```
void Gui::GUIUpdater::updateTimeUnitRequest (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) time unit request.

Parameters

<i>data</i>	The data to update the GUI GameData time unit request.
-------------	--

4.13.2.26 updateUnknownMessage()

```
void Gui::GUIUpdater::updateUnknownMessage (
    const std::vector< std::string > & data ) [private]
```

Update the GUI [GameData](#) unknow message.

Parameters

<i>data</i>	The data to update the GUI GameData unknow message.
-------------	---

4.13.3 Member Data Documentation

4.13.3.1 _updateMap

```
std::unordered_map<std::string, std::function<void(std::vector<std::string>)>> > Gui::GUIUpdater::_updateMap [private]
```

Initial value:

```
=
{
    {"msz", [this](std::vector<std::string> data){updateMapSize(data);}},
    {"bct", [this](std::vector<std::string> data){updateMapContent(data);}},
    {"tna", [this](std::vector<std::string> data){updateTeamNames(data);}},
    {"pnw", [this](std::vector<std::string> data){updateTeamMember(data);}},
    {"ppo", [this](std::vector<std::string> data){updatePlayerPosition(data);}},
    {"plv", [this](std::vector<std::string> data){updatePlayerLevel(data);}},
    {"pin", [this](std::vector<std::string> data){updatePlayerInventory(data);}},
    {"pex", [this](std::vector<std::string> data){updatePlayerExpulsion(data);}},
    {"pbc", [this](std::vector<std::string> data){updatePlayerBroadcast(data);}},
    {"pic", [this](std::vector<std::string> data){updatePlayerStartIncantation(data);}},
    {"pie", [this](std::vector<std::string> data){updatePlayerEndIncantation(data);}},
    {"pfc", [this](std::vector<std::string> data){updatePlayerEggLaying(data);}},
    {"pdr", [this](std::vector<std::string> data){updatePlayerRessourceDropping(data);}},
    {"pgt", [this](std::vector<std::string> data){updatePlayerRessourceCollecting(data);}},
    {"pdi", [this](std::vector<std::string> data){updatePlayerDeath(data);}},
    {"enw", [this](std::vector<std::string> data){updateEggLaidByPlayer(data);}},
    {"ebo", [this](std::vector<std::string> data){updatePlayerBorn(data);}},
    {"edi", [this](std::vector<std::string> data){updateEggDeath(data);}},
    {"sgt", [this](std::vector<std::string> data){updateTimeUnitRequest(data);}},
    {"sst", [this](std::vector<std::string> data){updateTimeUnitModification(data);}},
    {"seg", [this](std::vector<std::string> data){updateEndOfGame(data);}},
    {"smg", [this](std::vector<std::string> data){updateMessageFromServer(data);}},
    {"suc", [this](std::vector<std::string> data){updateUnknownMessage(data);}},
    {"sbp", [this](std::vector<std::string> data){updateCommandParameter(data);}},
}
```

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

4.14 Gui::Errors::GuiUpdaterException Class Reference

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

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

Inheritance diagram for Gui::Errors::GuiUpdaterException:

Collaboration diagram for Gui::Errors::GuiUpdaterException:

Public Member Functions

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

Constructor for [GuiUpdaterException](#).

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

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

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

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

Additional Inherited Members

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

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

4.14.1 Detailed Description

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

4.14.2 Constructor & Destructor Documentation

4.14.2.1 GuiUpdaterException()

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

Constructor for [GuiUpdaterException](#).

Parameters

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

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

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

4.15 Gui::HudGame Class Reference

Inheritance diagram for Gui::HudGame:

Collaboration diagram for Gui::HudGame:

Public Member Functions

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

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

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

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

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

Private Attributes

- Texture2D `_texture`
- Font `_font`
- Texture2D `_playerTexture`

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`

4.15.1 Constructor & Destructor Documentation

4.15.1.1 HudGame()

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

Construct a new Hud Game object.

Parameters

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

4.15.1.2 ~HudGame()

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

Destroy the Hud Game object.

4.15.2 Member Function Documentation

4.15.2.1 display()

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

Display Game Hud.

Implements [Gui::AHud](#).

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

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

4.16 Gui::HudPlayer Class Reference

Inheritance diagram for Gui::HudPlayer:

Collaboration diagram for Gui::HudPlayer:

Public Member Functions

- [HudPlayer](#) (std::shared_ptr< [GameData](#) > gameData)
Construct a new Hud [Player](#) object.
- [~HudPlayer](#) ()=default
Destroy the Hud [Player](#) object.
- void [display](#) ()
Display [Player](#) Hud.

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

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

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

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

Private Attributes

- Texture2D `_texture`
- Font `_font`
- Texture2D `_food`
- Texture2D `_linemate`
- Texture2D `_deraumere`
- Texture2D `_mendiane`
- Texture2D `_phiras`
- Texture2D `_sibur`
- Texture2D `_thystame`
- Texture2D `_playerTexture`

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`

4.16.1 Constructor & Destructor Documentation

4.16.1.1 HudPlayer()

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

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

Parameters

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

4.16.1.2 ~HudPlayer()

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

Destroy the Hud [Player](#) object.

4.16.2 Member Function Documentation

4.16.2.1 display()

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

Display [Player](#) Hud.

Implements [Gui::AHud](#).

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

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

4.17 Gui::HudTile Class Reference

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

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

Public Member Functions

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

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

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

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

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

Private Attributes

- Texture2D [_texture](#)
- Font [_font](#)
- Texture2D [_food](#)
- Texture2D [_linemate](#)
- Texture2D [_deraumere](#)
- Texture2D [_mendiane](#)
- Texture2D [_phiras](#)
- Texture2D [_sibur](#)
- Texture2D [_thystame](#)
- Texture2D [_tileTexture](#)
- Texture2D [_playerTexture](#)
- Texture2D [_eggTexture](#)

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](#)

4.17.1 Constructor & Destructor Documentation

4.17.1.1 HudTile()

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

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

Parameters

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

4.17.1.2 ~HudTile()

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

Destroy the Hud [Tile](#) object.

4.17.2 Member Function Documentation

4.17.2.1 display()

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

Display [Tile](#) Hud.

Implements [Gui::AHud](#).

4.17.2.2 displayNbEggs()

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

Display number of eggs.

4.17.2.3 displayNbPlayers()

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

Display number of players.

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

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

4.18 Gui::Errors::IError Class Reference

Inheritance diagram for Gui::Errors::IError:

Collaboration diagram for Gui::Errors::IError:

Public Member Functions

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

4.18.1 Member Function Documentation

4.18.1.1 what()

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

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

Returns

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

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

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

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

4.19 Gui::IEvent Class Reference

Inheritance diagram for Gui::IEvent:

Public Member Functions

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

4.19.1 Constructor & Destructor Documentation

4.19.1.1 IEvent()

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

Construct a new [IEvent](#) object.

4.19.1.2 ~IEvent()

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

Destroy the [IEvent](#) object.

4.19.2 Member Function Documentation

4.19.2.1 listen()

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

Listen the user's events.

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

4.19.2.2 setGameData()

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

Set the [GameData](#) object.

Parameters

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

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

4.19.2.3 setRender()

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

Set the [Render](#) object.

Parameters

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

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

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

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

4.20 Gui::IGUIUpdater Class Reference

Inheritance diagram for Gui::IGUIUpdater:

Public Member Functions

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

4.20.1 Member Function Documentation

4.20.1.1 update()

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

Update the GUI [GameData](#).

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

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

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

4.21 Gui::IHud Class Reference

Inheritance diagram for Gui::IHud:

Public Types

- enum [TypeScene](#) { [GAME](#) , [POV_PLAYER](#) , [END_GAME](#) , [TILE](#) }
Hud enum for the different scenes.

Public Member Functions

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

4.21.1 Member Enumeration Documentation

4.21.1.1 TypeScene

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

Hud enum for the different scenes.

4.21.2 Constructor & Destructor Documentation

4.21.2.1 ~IHud()

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

Destroy the [IHud](#) object.

4.21.3 Member Function Documentation

4.21.3.1 display()

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

Display the Hud.

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

4.21.3.2 getType()

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

Get the Type object.

Returns

[TypeScene](#) - Type of the scene.

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

4.21.3.3 setPlayer()

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

Set the [Player](#) object.

Parameters

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

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

4.21.3.4 setTile()

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

Set the [Tile](#) object.

Parameters

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

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

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

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

4.22 Gui::INetwork Class Reference

Inheritance diagram for Gui::INetwork:

Public Member Functions

- virtual [~INetwork](#) ()=default
Destroy the [INetwork](#) object.
- virtual void [setPort](#) (int port)=0
Set the port object.
- virtual void [setHostName](#) (const std::string &hostName)=0
Set the host name object.
- virtual int [getPort](#) () const =0
Get the host name object.
- virtual std::string [getHostName](#) () const =0
Get the host name object.
- virtual void [connectToServer](#) ()=0
Connect to the server.
- virtual const std::string [listenServer](#) ()=0
Listen to the server.
- virtual void [sendMessageServer](#) (const std::string &message)=0
Send a message to the server.

4.22.1 Constructor & Destructor Documentation

4.22.1.1 ~INetwork()

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

Destroy the [INetwork](#) object.

4.22.2 Member Function Documentation

4.22.2.1 connectToServer()

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

Connect to the server.

Exceptions

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

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

4.22.2.2 getHostName()

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

Get the host name object.

Returns

std::string Host of the server.

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

4.22.2.3 getPort()

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

Get the host name object.

Returns

std::string Host of the server.

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

4.22.2.4 listenServer()

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

Listen to the server.

Returns

std::string Message from the server.

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

4.22.2.5 sendMessageServer()

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

Send a message to the server.

Parameters

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

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

4.22.2.6 setHostName()

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

Set the host name object.

Parameters

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

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

4.22.2.7 setPort()

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

Set the port object.

Parameters

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

Exceptions

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

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

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

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

4.23 Gui::Inventory Class Reference

Public Types

- using **Ressources** = size_t[RessourcesNumber]

Public Member Functions

- [Inventory](#) ()
Construct a new [Inventory](#) object.
- [Inventory](#) (std::size_t food, std::size_t linemate, std::size_t deraumere, std::size_t sibur, std::size_t mendiane, std::size_t phiras, std::size_t thystame)
Construct a new [Inventory](#) object.
- [~Inventory](#) ()=default
Destroy the [Inventory](#) object.
- void [setFood](#) (std::size_t food)
Set the Food object.
- void [setLinemate](#) (std::size_t linemate)
Set the Linemate object.
- void [setDeraumere](#) (std::size_t deraumere)
Set the Deraumere object.
- void [setSibur](#) (std::size_t sibur)
Set the Sibur object.
- void [setMendiane](#) (std::size_t mendiane)
Set the Mendiane object.
- void [setPhiras](#) (std::size_t phiras)
Set the Phiras object.
- void [setThystame](#) (std::size_t thysame)
Set the Thystame object.
- void [setRessources](#) (Ressources ressources)
Set the Ressources object.
- std::size_t [getFood](#) (void)
Get the Food object.
- std::size_t [getLinemate](#) (void)
Get the Linemate object.
- std::size_t [getDeraumere](#) (void)
Get the Deraumere object.
- std::size_t [getSibur](#) (void)
Get the Sibur object.
- std::size_t [getMendiane](#) (void)
Get the Mendiane object.
- std::size_t [getPhiras](#) (void)
Get the Phiras object.
- std::size_t [getThystame](#) (void)
Get the Thystame object.
- Ressources & [getRessources](#) (void)
Get the Ressources object.
- void [addResource](#) (std::size_t resource, std::size_t quantity)
Add resources to inventory.
- void [removeResource](#) (std::size_t resource, std::size_t quantity)
Remove resources to inventory.

Private Attributes

- std::size_t **_food**
- Ressources **_ressources**

4.23.1 Constructor & Destructor Documentation

4.23.1.1 Inventory() [1/2]

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

Construct a new [Inventory](#) object.

4.23.1.2 Inventory() [2/2]

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

Construct a new [Inventory](#) object.

Parameters

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

4.23.1.3 ~Inventory()

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

Destroy the [Inventory](#) object.

4.23.2 Member Function Documentation

4.23.2.1 addResource()

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

Add resources to inventory.

Parameters

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

4.23.2.2 getDeraumere()

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

Get the Deraumere object.

Returns

std::size_t - deraumere

4.23.2.3 getFood()

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

Get the Food object.

Returns

std::size_t - food

4.23.2.4 getLinemate()

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

Get the Linemate object.

Returns

std::size_t - linemate

4.23.2.5 getMendiane()

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

Get the Mendiane object.

Returns

std::size_t - mendiane

4.23.2.6 getPhiras()

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

Get the Phiras object.

Returns

std::size_t - phiras

4.23.2.7 getRessources()

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

Get the Ressources object.

Returns

Ressources - ressources

4.23.2.8 getSibur()

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

Get the Sibur object.

Returns

std::size_t - sibur

4.23.2.9 getThystame()

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

Get the Thystame object.

Returns

std::size_t - thystame

4.23.2.10 removeResource()

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

Remove resources to inventory.

Parameters

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

4.23.2.11 setDeraumere()

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

Set the Deraumere object.

Parameters

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

4.23.2.12 setFood()

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

Set the Food object.

Parameters

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

4.23.2.13 setLinemate()

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

Set the Linemate object.

Parameters

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

4.23.2.14 setMendiane()

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

Set the Mendiane object.

Parameters

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

4.23.2.15 setPhiras()

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

Set the Phiras object.

Parameters

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

4.23.2.16 setRessources()

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

Set the Ressources object.

Parameters

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

4.23.2.17 setSibur()

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

Set the Sibur object.

Parameters

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

4.23.2.18 setThystame()

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

Set the Thystame object.

Parameters

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

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

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

4.24 Gui::IServerParser Class Reference

Inheritance diagram for Gui::IServerParser:

Public Member Functions

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

4.24.1 Member Function Documentation

4.24.1.1 parse()

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

Parse the command server.

Parameters

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

Returns

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

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

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

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

4.25 Gui::Network Class Reference

Inheritance diagram for Gui::Network:

Collaboration diagram for Gui::Network:

Public Member Functions

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

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

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

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

- virtual [~INetwork](#) ()=default
Destroy the [INetwork](#) object.
- virtual void [setPort](#) (int port)=0
Set the port object.
- virtual void [setHostName](#) (const std::string &hostName)=0
Set the host name object.
- virtual int [getPort](#) () const =0
Get the host name object.
- virtual std::string [getHostName](#) () const =0
Get the host name object.
- virtual void [connectToServer](#) ()=0
Connect to the server.
- virtual const std::string [listenServer](#) ()=0
Listen to the server.
- virtual void [sendMessageServer](#) (const std::string &message)=0
Send a message to the server.

Private Member Functions

- void [selectServer](#) ()
Listen if there is a server event.
- const std::string [readInfoServer](#) ()
Read the server output.

Private Attributes

- `int _serverFd`
server file descriptor
- `fd_set _writeFd`
file descriptor for write access
- `fd_set _readFd`
file descriptor for read access
- `bool _isConnected`
is true if the gui is connected to the server

Additional Inherited Members

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

- `int _port`
- `std::string _hostName`

4.25.1 Constructor & Destructor Documentation

4.25.1.1 `Network()`

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

Construct a new [Network](#) object.

Parameters

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

4.25.1.2 `~Network()`

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

Destroy the [Network](#) object.

4.25.2 Member Function Documentation

4.25.2.1 connectToServer()

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

Connect the Gui network with the server.

Exceptions

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

Implements [Gui::ANetwork](#).

4.25.2.2 listenServer()

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

Listen the server and return it message.

Returns

std::string - Message of the server.

Implements [Gui::ANetwork](#).

4.25.2.3 readInfoServer()

```
const std::string Gui::Network::readInfoServer ( ) [private]
```

Read the server output.

Returns

const std::string - Server message.

4.25.2.4 selectServer()

```
void Gui::Network::selectServer ( ) [private]
```

Listen if there is a server event.

4.25.2.5 sendMessageServer()

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

Send a message to the Server.

Parameters

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

Implements [Gui::ANetwork](#).

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

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

4.26 Gui::Errors::NetworkException Class Reference

[Error](#) class for network errors.

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

Inheritance diagram for Gui::Errors::NetworkException:

Collaboration diagram for Gui::Errors::NetworkException:

Public Member Functions

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

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

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

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

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

Additional Inherited Members

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

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

4.26.1 Detailed Description

[Error](#) class for network errors.

4.26.2 Constructor & Destructor Documentation

4.26.2.1 NetworkException()

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

Constructor for [NetworkException](#).

Parameters

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

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

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

4.27 Gui::ParseCommandLine Class Reference

Public Member Functions

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

Private Attributes

- int **_port**
- std::string **_hostName**

4.27.1 Constructor & Destructor Documentation

4.27.1.1 ParseCommandLine()

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

Construct a new Parse Command Line object.

Parameters

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

4.27.1.2 ~ParseCommandLine()

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

Destroy the Parse Command Line object.

4.27.2 Member Function Documentation

4.27.2.1 getHostName()

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

Get the hostName object.

Returns

std::string - hostName

4.27.2.2 `getPort()`

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

Get the port object.

Returns

int - port

4.27.2.3 `parseFlags()`

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

Parse flags in command line.

Parameters

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

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

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

4.28 `Gui::Errors::ParseCommandLineException` Class Reference

[Error](#) class for `parseCommandLine` errors.

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

Inheritance diagram for `Gui::Errors::ParseCommandLineException`:

Collaboration diagram for `Gui::Errors::ParseCommandLineException`:

Public Member Functions

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

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

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

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

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

Additional Inherited Members**Protected Attributes inherited from [Gui::Errors::AError](#)**

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

4.28.1 Detailed Description

[Error](#) class for parseCommandLine errors.

4.28.2 Constructor & Destructor Documentation**4.28.2.1 ParseCommandLineException()**

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

Constructor for [ParseCommandLineException](#).

Parameters

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

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

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

4.29 Gui::Player Class Reference

Collaboration diagram for Gui::Player:

Public Types

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

Public Member Functions

- **Player** (std::size_t id, const std::string &team, std::pair< std::size_t, std::size_t > position, std::size_t orientation, std::size_t level=1)
Construct a new **Player** object.
- **~Player** ()=default
Destroy the **Player** object.
- void **setPosition** (std::pair< std::size_t, std::size_t > position)
Set the **Position** object.
- void **setPosition3D** (Vector3 position3D)
Set the **Position3D** object.
- void **setId** (std::size_t id)
Set the **Id** object.
- void **setLevel** (std::size_t level)
Set the **Level** object.
- void **setOrientation** (std::size_t orientation)
Set the **Orientation** object.
- void **setTeam** (const std::string &team)
Set the **Team** object.
- std::pair< std::size_t, std::size_t > **getPosition** (void) const
Get the **Position** object.
- Vector3 **getPosition3D** (void) const
Get the **Position3D** object.
- std::size_t **getId** (void) const
Get the **Id** object.
- std::size_t **getLevel** (void) const
Get the **Level** object.
- std::size_t **getOrientation** (void) const
Get the **Orientation** object.
- std::string **getTeam** (void) const
Get the **Team** object.
- void **setState** (PlayerState state)
Set the **State** object.
- PlayerState **getState** (void) const
Get the **State** object.
- void **setBroadcast** (const std::string &broadcast)
Set the **Broadcast** object.
- std::string **getBroadcast** () const

- Get the Broadcast object.*

 - float [getRotationFromOrientation](#) () const

Get the Vector From Orientation object.
- Vector3 [getCenterPosition](#) ()

Get the Center Position object.
- void [setCurrentFrame](#) (int currentFrame)

Set the Current Frame object.
- int [getCurrentFrame](#) () const

Get the Current Frame object.
- void [restartAnimationTimeEllapsed](#) ()

Restart the timer animation.
- clock_t [getAnimationTimeEllapsed](#) () const

Get the Animation Time Ellapsed object.

Public Attributes

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

Private Attributes

- std::size_t **_id**
Id of the player.
- std::string **_team**
Team name.
- std::pair< std::size_t, std::size_t > **_position**
Position x y relative to tiles.
- Vector3 **_position3D**
Position in 3D scene.
- std::size_t **_orientation**
Orientation of the player.
- std::size_t **_level**
Level between 1 - 8.
- PlayerState **_state**
Player state.
- std::string **_broadcast**
Broadcast message.
- int **_currentFrame**
Current frame animation.
- clock_t **_animationTimeEllapsed**
Time ellapsed during animation.

4.29.1 Constructor & Destructor Documentation

4.29.1.1 Player()

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

Construct a new [Player](#) object.

Parameters

<i>id</i>	Id of the player.
<i>team</i>	Team name of the player.
<i>position</i>	Position of the Player .

4.29.1.2 ~Player()

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

Destroy the [Player](#) object.

4.29.2 Member Function Documentation

4.29.2.1 getAnimationTimeEllapsed()

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

Get the Animation Time Ellapsed object.

Returns

clock_t - Animation time ellapsed.

4.29.2.2 getBroadcast()

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

Get the Broadcast object.

Returns

std::string - Broadcast message.

4.29.2.3 getCenterPosition()

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

Get the Center Position object.

Returns

Vector3 - Center position.

4.29.2.4 getCurrentFrame()

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

Get the Current Frame object.

Returns

int - Current frame.

4.29.2.5 getId()

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

Get the Id object.

Returns

std::size_t - id

4.29.2.6 getLevel()

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

Get the Level object.

Returns

std::size_t - level

4.29.2.7 getOrientation()

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

Get the Orientation object.

Returns

std::size_t - orientation

4.29.2.8 getPosition()

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

Get the Position object.

Returns

std::pair<std::size_t, std::size_t> - position

4.29.2.9 getPosition3D()

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

Get the Position3D object.

Returns

Vector3 - position3D

4.29.2.10 getRotationFromOrientation()

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

Get the Vector From Orientation object.

Returns

Vector3 - Vector3 from orientation.

4.29.2.11 getState()

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

Get the State object.

Returns

PlayerState - [Player](#) state.

4.29.2.12 getTeam()

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

Get the [Team](#) object.

Returns

std::string - team name

4.29.2.13 restartAnimationTimeEllapsed()

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

Restart the timer animation.

4.29.2.14 setBroadcast()

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

Set the Broadcast object.

Parameters

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

4.29.2.15 setCurrentFrame()

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

Set the Current Frame object.

Parameters

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

4.29.2.16 setId()

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

Set the Id object.

Parameters

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

4.29.2.17 setLevel()

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

Set the Level object.

Parameters

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

4.29.2.18 setOrientation()

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

Set the Orientation object.

Parameters

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

4.29.2.19 setPosition()

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

Set the Position object.

Parameters

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

4.29.2.20 setPosition3D()

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

Set the Position3D object.

Parameters

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

4.29.2.21 setState()

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

Set the State object.

Parameters

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

4.29.2.22 setTeam()

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

Set the [Team](#) object.

Parameters

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

4.29.3 Member Data Documentation

4.29.3.1 inventory

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

[Inventory](#) of the player.

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

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

4.30 Gui::Render Class Reference

Collaboration diagram for Gui::Render:

Public Member Functions

- [Render](#) (std::shared_ptr< [GameData](#) > gameData)
Construct a new [Render](#) object.
- [~Render](#) ()
Destroy the [Render](#) object.
- bool [isOpen](#) ()
Check if the window is open.
- void [draw](#) ()
Draw the scene.
- std::shared_ptr< Camera > [getCamera](#) ()
Get the Camera object.
- void [setIsDebug](#) (bool isDebug)
Set the Is Debug object.
- bool [getIsDebug](#) (void)
Get the Is Debug object.

- void [setCameraType](#) (Gui::UserCamera::CameraType type)
Set the Type object.
- Gui::UserCamera::CameraType [getCameraType](#) () const
Get the Type object.
- void [setCameraPlayerPov](#) (std::size_t id)
Set the Camera player pov id.
- std::size_t [getCameraPlayerPov](#) () const
Get the Camera player pov id.
- void [setCameraTile](#) (std::pair< std::size_t, std::size_t > pos)
Set the Camera Tile object.
- std::pair< std::size_t, std::size_t > [getCameraTile](#) () const
Get the Camera Tile object.
- Model [getTileModel](#) () const
Get the Tile model.
- void [setRenderDistance](#) (size_t renderDistance)
Set the Render Distance value.
- size_t [getRenderDistance](#) () const
Get the Render Distance value.
- bool [isCameraInPlayerPov](#) () const
Check if the camera is in player pov.
- void [changePlayerPOV](#) (size_t playerId)
Change the player point of view.
- void [setPlayerPov](#) (size_t playerId)
Sets the Pov of the player.
- void [changePOVToFirstPerson](#) (size_t id)
Change the camera to the player.
- void [changePOVToSecondPerson](#) (size_t id)
Change the camera to the player.
- void [changePOVToThirdPerson](#) (size_t id)
Change the camera to the player.
- size_t [getTimeUnit](#) () const
Get the Time Unit value.
- void [setTimeUnit](#) (size_t timeUnit)
Set the Time Unit value.

Private Member Functions

- void [LoadModels](#) ()
Load the models to draw.
- void [displayHUD](#) ()
Display HUD infos related to player's pov.
- void [displayDebug](#) ()
Display the debug interface.
- void [displayPlayers](#) ()
Display players.
- void [displayPlayerLevel](#) (Team &team, Player &player)
Display player level.
- void [displayPlayerBroadcast](#) (Team &team, Player &player)
Display player broadcast.
- void [displayMap](#) ()

- *Display the map.*
• void [displayTile](#) ([Tile](#) tile)
- *Display a [Tile](#).*
• void [displayEggs](#) ([Tile](#) tile) const
- *Display the eggs.*
• void [displayFood](#) ([Tile](#) tile) const
- *Display the food.*
• void [displayResources](#) ([Tile](#) tile) const
- *Display resources.*
• void [displayLinemate](#) ([Tile](#) tile) const
- *Display Linemate.*
• void [displayMendiane](#) ([Tile](#) tile) const
- *Display Mendiane.*
• void [displayPhiras](#) ([Tile](#) tile) const
- *Display Phiras.*
• void [displaySibur](#) ([Tile](#) tile) const
- *Display Sibur.*
• void [displayThystame](#) ([Tile](#) tile) const
- *Display Thystam.*
• void [displayDeraumere](#) ([Tile](#) tile) const
- *Display Deraumere.*
• bool [displayAnimations](#) ([Team](#) &team, [Player](#) &player)
- *Display animations.*
• [ModelAnimation](#) [displayWalkAnimation](#) ([Team](#) &team, [Player](#) &player, [ModelAnimation](#) anim)
- *Display walk animation.*
• void [displayCursor](#) ()
- *Display the cursor.*
• std::pair< std::size_t, std::size_t > [getCameraTile](#) ()
- *Get the closest tile from the camera.*

Private Attributes

- [UserCamera](#) _camera
- bool _isDebug
- std::shared_ptr< [GameData](#) > _gameData
- std::shared_ptr< [Decoration](#) > _decoration
- std::vector< std::shared_ptr< [Gui::IHud](#) > > _hudList
- size_t _renderDistance
- [Model](#) _tileModel
- [Model](#) _foodModel
- [Model](#) _linemateModel
- [Model](#) _mendianeModel
- [Model](#) _phirasModel
- [Model](#) _siburModel
- [Model](#) _thystameModel
- [Model](#) _deraumereModel
- [Texture2D](#) _cursorTexture

4.30.1 Constructor & Destructor Documentation

4.30.1.1 Render()

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

Construct a new [Render](#) object.

4.30.1.2 ~Render()

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

Destroy the [Render](#) object.

4.30.2 Member Function Documentation

4.30.2.1 changePlayerPOV()

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

Change the player point of view.

Parameters

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

Note

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

4.30.2.2 changePOVToFirstPerson()

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

Change the camera to the player.

Parameters

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

4.30.2.3 changePOVToSecondPerson()

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

Change the camera to the player.

Parameters

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

4.30.2.4 changePOVToThirdPerson()

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

Change the camera to the player.

Parameters

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

4.30.2.5 displayAnimations()

```
bool Gui::Render::displayAnimations (
    Team & team,
    Player & player ) [private]
```

Display animations.

Parameters

<i>team</i>	Team for model.
<i>player</i>	Player to draw animation.

4.30.2.6 displayCursor()

```
void Gui::Render::displayCursor ( ) [private]
```


Display the cursor.

4.30.2.7 displayDebug()

```
void Gui::Render::displayDebug ( ) [private]
```

Display the debug interface.

4.30.2.8 displayDeraumere()

```
void Gui::Render::displayDeraumere (
    Tile tile ) const [private]
```

Display Deraumere.

Parameters

<i>tile</i>	Tile with resources.
-------------	----------------------

4.30.2.9 displayEggs()

```
void Gui::Render::displayEggs (
    Tile tile ) const [private]
```

Display the eggs.

Parameters

<i>tile</i>	Tile with eggs.
-------------	-----------------

4.30.2.10 displayFood()

```
void Gui::Render::displayFood (
    Tile tile ) const [private]
```

Display the food.

Parameters

<i>tile</i>	Tile with food.
-------------	-----------------

4.30.2.11 displayHUD()

```
void Gui::Render::displayHUD (
    void ) [private]
```

Display HUD infos related to player's pov.

4.30.2.12 displayLinemate()

```
void Gui::Render::displayLinemate (
    Tile tile ) const [private]
```

Display Linemate.

Parameters

<i>tile</i>	Tile with resources.
-------------	--------------------------------------

4.30.2.13 displayMap()

```
void Gui::Render::displayMap ( ) [private]
```

Display the map.

4.30.2.14 displayMendiane()

```
void Gui::Render::displayMendiane (
    Tile tile ) const [private]
```

Display Mendiane.

Parameters

<i>tile</i>	Tile with resources.
-------------	--------------------------------------

4.30.2.15 displayPhiras()

```
void Gui::Render::displayPhiras (
```

```
Tile tile ) const [private]
```

Display Phiras.

Parameters

<i>tile</i>	Tile with resources.
-------------	----------------------

4.30.2.16 displayPlayerBroadcast()

```
void Gui::Render::displayPlayerBroadcast (
    Gui::Team & team,
    Gui::Player & player ) [private]
```

Display player broadcast.

Parameters

<i>team</i>	Team for the player 3d position.
<i>player</i>	Player to display broadcast.

4.30.2.17 displayPlayerLevel()

```
void Gui::Render::displayPlayerLevel (
    Gui::Team & team,
    Gui::Player & player ) [private]
```

Display player level.

Parameters

<i>team</i>	Team for the player 3d position.
<i>player</i>	Player to display level.

4.30.2.18 displayPlayers()

```
void Gui::Render::displayPlayers ( ) [private]
```

Display players.

4.30.2.19 displayResources()

```
void Gui::Render::displayResources (
    Tile tile ) const [private]
```

Display resources.

Parameters

<i>tile</i>	Tile with resources.
-------------	----------------------

4.30.2.20 displaySibur()

```
void Gui::Render::displaySibur (
    Tile tile ) const [private]
```

Display Sibur.

Parameters

<i>tile</i>	Tile with resources.
-------------	----------------------

4.30.2.21 displayThystame()

```
void Gui::Render::displayThystame (
    Tile tile ) const [private]
```

Display Thystam.

Parameters

<i>tile</i>	Tile with resources.
-------------	----------------------

4.30.2.22 displayTile()

```
void Gui::Render::displayTile (
    Tile tile ) [private]
```

Display a Tile.

4.30.2.23 displayWalkAnimation()

```
ModelAnimation Gui::Render::displayWalkAnimation (
    Team & team,
    Player & player,
    ModelAnimation anim ) [private]
```

Display walk animation.

Parameters

<i>team</i>	Team for model.
<i>player</i>	Player to draw animation.
<i>anim</i>	Animation Model.

Returns

ModelAnimation - Animation.

4.30.2.24 draw()

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

Draw the scene.

4.30.2.25 getCamera()

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

Get the Camera object.

Returns

std::shared_ptr<Camera> - camera

4.30.2.26 getCameraPlayerPov()

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

Get the Camera player pov id.

Returns

std::size_t - Id of the player.

4.30.2.27 getCameraTile() [1/2]

```
std::pair< size_t, size_t > Gui::Render::getCameraTile ( ) [private]
```

Get the closest tile from the camera.

Returns

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

4.30.2.28 getCameraTile() [2/2]

```
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.

4.30.2.29 getCameraType()

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

Get the Type object.

Returns

CameraType - Camera type.

4.30.2.30 getIsDebug()

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

Get the Is Debug object.

Returns

true - diplay debug

false - do not display debug

4.30.2.31 getRenderDistance()

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

Get the [Render](#) Distance value.

4.30.2.32 getTileModel()

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

Get the [Tile](#) model.

4.30.2.33 getTimeUnit()

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

Get the Time Unit value.

Returns

size_t - Time unit value.

4.30.2.34 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.

4.30.2.35 isOpen()

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

Check if the window is open.

Returns

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

4.30.2.36 LoadModels()

```
void Gui::Render::LoadModels (
    void ) [private]
```

Load the models to draw.

4.30.2.37 setCameraPlayerPov()

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

Set the Camera player pov id.

Parameters

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

4.30.2.38 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.
------------	--------------------------------

4.30.2.39 setCameraType()

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

Set the Type object.

Parameters

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

4.30.2.40 setIsDebug()

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

Set the Is Debug object.

Parameters

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

4.30.2.41 setPlayerPov()

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

Sets the Pov of the player.

Parameters

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

4.30.2.42 setRenderDistance()

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

Set the [Render](#) Distance value.

Parameters

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

4.30.2.43 setTimeUnit()

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

Set the Time Unit value.

Parameters

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

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

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

4.31 Gui::ServerParser Class Reference

Inheritance diagram for Gui::ServerParser:

Collaboration diagram for Gui::ServerParser:

Public Member Functions

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

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

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

Private Types

- enum [ParseType](#) { INT , STRING , MESSAGE , LIST_INT }

Enum of types to parse.

Private Member Functions

- std::vector< std::string > [parseCommand](#) (const std::string &command, std::vector< [ParseType](#) > types)
Parse the command with its types.
- std::vector< std::string > [parseInt](#) (std::istream &stream, std::vector< std::string > arguments)
Parse an int in the command stream.
- std::vector< std::string > [parseString](#) (std::istream &stream, std::vector< std::string > arguments)
Parse a string in the command stream.
- std::vector< std::string > [parseMessage](#) (std::istream &stream, std::vector< std::string > arguments, std::string commandName)
Parse a message in the command stream.
- std::vector< std::string > [parseListInt](#) (std::istream &stream, std::vector< std::string > arguments, std::string commandName)
Parse a list of int in the command stream.

Private Attributes

- std::unordered_map< std::string, std::vector< [ParseType](#) > > [_typesCommand](#)
Map of types to parse related to the command.

4.31.1 Member Enumeration Documentation

4.31.1.1 ParseType

```
enum Gui::ServerParser::ParseType [private]
```

Enum of types to parse.

4.31.2 Constructor & Destructor Documentation

4.31.2.1 ServerParser()

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

Construct a new Server Parser object.

4.31.2.2 ~ServerParser()

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

Destroy the Server Parser object.

4.31.3 Member Function Documentation

4.31.3.1 parse()

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

Parse the command server.

Parameters

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

Returns

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

Implements [Gui::IServerParser](#).

4.31.3.2 parseCommand()

```
std::vector< std::string > Gui::ServerParser::parseCommand (
    const std::string & command,
    std::vector< ParseType > types ) [private]
```

Parse the command with its types.

Parameters

<i>command</i>	Command to parse.
<i>types</i>	Types within parse the command.

Returns

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

4.31.3.3 parseInt()

```
std::vector< std::string > Gui::ServerParser::parseInt (
    std::istream & stream,
    std::vector< std::string > arguments ) [private]
```

Parse an int in the command stream.

Parameters

<i>stream</i>	Stream to parse.
<i>arguments</i>	List of arguments parsed.

Returns

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

4.31.3.4 parseListInt()

```
std::vector< std::string > Gui::ServerParser::parseListInt (
    std::istream & stream,
    std::vector< std::string > arguments,
    std::string commandName ) [private]
```

Parse a list of int in the command stream.

Parameters

<i>stream</i>	Stream to parse.
<i>arguments</i>	List of arguments parsed.
<i>commandName</i>	Name of the server command.

Returns

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

4.31.3.5 parseMessage()

```
std::vector< std::string > Gui::ServerParser::parseMessage (
    std::istream & stream,
    std::vector< std::string > arguments,
    std::string commandName ) [private]
```

Parse a message in the command stream.

Parameters

<i>stream</i>	Stream to parse.
<i>arguments</i>	List of arguments parsed.
<i>commandName</i>	Name of the server command.

Returns

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

4.31.3.6 `parseString()`

```
std::vector< std::string > Gui::ServerParser::parseString (
    std::istream & stream,
    std::vector< std::string > arguments ) [private]
```

Parse a string in the command stream.

Parameters

<i>stream</i>	Stream to parse.
<i>arguments</i>	List of arguments parsed.

Returns

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

4.31.4 Member Data Documentation

4.31.4.1 `_typesCommand`

```
std::unordered_map<std::string, std::vector<ParseType> > Gui::ServerParser::_typesCommand
[private]
```

Initial value:

=

```
{
    {"msz", std::vector<ParseType>{INT, INT}},
    {"bct", std::vector<ParseType>{INT, INT, INT, INT, INT, INT, INT, INT, INT}},
    {"tna", std::vector<ParseType>{STRING}},
    {"pnw", std::vector<ParseType>{INT, INT, INT, INT, INT, STRING}},
    {"ppo", std::vector<ParseType>{INT, INT, INT, INT}},
    {"plv", std::vector<ParseType>{INT, INT}},
    {"pin", std::vector<ParseType>{INT, INT, INT, INT, INT, INT, INT, INT, INT, INT}},
    {"pex", std::vector<ParseType>{INT}},
    {"pbc", std::vector<ParseType>{INT, MESSAGE}},
    {"pic", std::vector<ParseType>{INT, INT, INT, LIST_INT}},
    {"pie", std::vector<ParseType>{INT, INT, INT}},
    {"pfk", std::vector<ParseType>{INT}},
}
```

```

    {"pdr", std::vector<ParseType>{INT, INT}},
    {"pgt", std::vector<ParseType>{INT, INT}},
    {"pdi", std::vector<ParseType>{INT}},
    {"enw", std::vector<ParseType>{INT, INT, INT, INT}},
    {"ebo", std::vector<ParseType>{INT}},
    {"edi", std::vector<ParseType>{INT}},
    {"sgt", std::vector<ParseType>{INT}},
    {"sst", std::vector<ParseType>{INT}},
    {"seg", std::vector<ParseType>{STRING}},
    {"smg", std::vector<ParseType>{MESSAGE}},
    {"suc", std::vector<ParseType>{}},
    {"sbp", std::vector<ParseType>{}}
}

```

Map of types to parse related to the command.

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

4.32 Gui::Errors::ServerParserException Class Reference

[Error](#) class for network errors.

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

Inheritance diagram for Gui::Errors::ServerParserException:

Collaboration diagram for Gui::Errors::ServerParserException:

Public Member Functions

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

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

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

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

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

Additional Inherited Members

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

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

4.32.1 Detailed Description

[Error](#) class for network errors.

4.32.2 Constructor & Destructor Documentation

4.32.2.1 [ServerParserException\(\)](#)

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

Constructor for [ServerParserException](#).

Parameters

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

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

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

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

Public Member Functions

- [Team](#) (const std::string &name, const std::string &playerModelPath, const std::string &eggModelPath, Color playerColor)
Construct a new [Team](#) object.
- [~Team](#) ()
Destroy the [Team](#) object.
- const std::string & [getName](#) () const
Get the Name object.
- std::vector< [Gui::Player](#) > & [getPlayers](#) ()
Get the Players object.
- std::vector< [Gui::Egg](#) > & [getEggs](#) ()

- Get the Eggs object.*

 - void [setName](#) (const std::string &name)

Set the Name object.

 - void [addPlayer](#) (const [Gui::Player](#) &player)

Add a player to the team.

 - void [addEgg](#) (const [Gui::Egg](#) &egg)

Add an egg to the team.

 - bool [removePlayer](#) (std::size_t id)

Remove a player from the team.

 - bool [removeEgg](#) (std::size_t id)

Remove an egg from the team.

 - std::shared_ptr< [Gui::Player](#) > [getPlayer](#) (std::size_t id)

Get the [Player](#) object.

 - Model [getPlayerModel](#) () const

Get the Model object.

 - ModelAnimation * [getPlayerModelAnimation](#) () const

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

 - void [setPlayerModelPath](#) (const std::string &playerModelPath)

Set the Model object.

 - std::shared_ptr< [Gui::Egg](#) > [getEgg](#) (std::size_t id)

Get the [Egg](#) object.

 - Model [getEggModel](#) () const

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

 - void [setEggModelPath](#) (const std::string &eggModelPath)

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

 - std::vector< BoundingBox > [getPlayerBoundingBoxes](#) (std::pair< size_t, size_t > pos, size_t orientation, Vector3 center)

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

 - Vector3 [getPlayerPositionIn3DSpace](#) (size_t id, Map< [Tile](#) > map)

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

 - std::vector< RayCollision > [getPlayerModelHitbox](#) (size_t id, Camera camera)

Get the [Player](#) Model hitbox.

 - bool [isPlayerHit](#) (size_t id, Camera camera)

Check if the player is hit.

 - Color [getPlayerColor](#) () const

Get the [Player](#) Color object.

Private Member Functions

- BoundingBox [rotateBoundingBoxByOrientation](#) (BoundingBox bbox, size_t orientation, std::pair< size_t, size_t > pos, Vector3 center)
- Rotate a bounding box by orientation.*

Private Attributes

- `ModelAnimation * _modelAnimation`
Model to animate players.
- `int _animCount`
Animation number of players.
- `std::string _name`
Name of the team.
- `std::vector< Gui::Player > _players`
Players of the team.
- `Model _playerModel`
Model player asset of the team.
- `std::vector< Gui::Egg > _eggs`
Eggs of the team.
- `Model _eggModel`
Eggs Model of the team.
- `Color _playerColor`
Color of the players.

4.33.1 Constructor & Destructor Documentation

4.33.1.1 Team()

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

Construct a new [Team](#) object.

Parameters

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

4.33.1.2 ~Team()

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

Destroy the [Team](#) object.

4.33.2 Member Function Documentation

4.33.2.1 addEgg()

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

Add an egg to the team.

Parameters

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

4.33.2.2 addPlayer()

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

Add a player to the team.

Parameters

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

4.33.2.3 getEgg()

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

Get the Gui::Egg object.

Parameters

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

Returns

std::shared_ptr<Gui::Egg> Gui::Egg.

4.33.2.4 getEggModel()

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

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

Returns

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

4.33.2.5 getEggs()

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

Get the Eggs object.

Returns

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

4.33.2.6 getName()

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

Get the Name object.

Returns

const std::string& Name of the team.

4.33.2.7 getPlayer()

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

Get the [Player](#) object.

Parameters

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

Returns

std::shared_ptr<Gui::Player> [Player](#).

4.33.2.8 getPlayerBoundingBoxes()

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

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

Parameters

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

Returns

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

4.33.2.9 getPlayerColor()

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

Get the [Player](#) Color object.

4.33.2.10 getPlayerModel()

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

Get the Model object.

Returns

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

4.33.2.11 getPlayerModelAnimation()

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

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

Returns

ModelAnimation* - Players' animations.

4.33.2.12 getPlayerModelHitbox()

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

Get the [Player](#) Model hitbox.

Parameters

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

Returns

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

4.33.2.13 getPlayerPositionIn3DSpace()

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

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

Parameters

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

4.33.2.14 getPlayers()

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

Get the Players object.

Returns

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

4.33.2.15 isPlayerHit()

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

Check if the player is hit.

Parameters

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

Returns

true If the player is hit.

4.33.2.16 removeEgg()

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

Remove an egg from the team.

Parameters

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

Returns

true If the egg has been removed.

false If the egg has not been removed.

4.33.2.17 removePlayer()

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

Remove a player from the team.

Parameters

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

Returns

true If the player has been removed.
false If the player has not been removed.

4.33.2.18 rotateBoundingBoxByOrientation()

```
BoundingBox Gui::Team::rotateBoundingBoxByOrientation (
    BoundingBox bbox,
    size_t orientation,
    std::pair< size_t, size_t > pos,
    Vector3 center ) [private]
```

Rotate a bounding box by orientation.

Parameters

<i>bbox</i>	Bounding box to rotate.
<i>orientation</i>	Orientation of the player.
<i>pos</i>	Position of the player.
<i>center</i>	Center of the player.

Returns

BoundingBox Rotated bounding box.

4.33.2.19 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.
--------------------	-------------------------------------

4.33.2.20 setName()

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

Set the Name object.

Parameters

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

4.33.2.21 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

4.34 Gui::Tile Class Reference

Collaboration diagram for Gui::Tile:

Public Member Functions

- [Tile](#) (std::pair< std::size_t, std::size_t > position)
Construct a new [Tile](#) object.
- [Tile](#) (std::pair< std::size_t, std::size_t > position, [Inventory inventory](#))
Construct a new [Tile](#) object.
- [~Tile](#) ()=default
Destroy the [Tile](#) object.
- void [setPosition](#) (std::pair< std::size_t, std::size_t > position)
Set the Position object.
- std::pair< std::size_t, std::size_t > [getPosition](#) () const
Get the Position object.
- Vector3 [getPositionIn3DSpace](#) ()

Get the Position In Space object.

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

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

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

Get the [Tile](#) Model Hitbox object.

- `bool isTileHit (Camera camera, Model _tileModel)`

Check if the tile is hit.

Public Attributes

- [Inventory](#) `inventory`

[Inventory](#) of the tile.

Private Attributes

- `std::pair< std::size_t, std::size_t > _position`

Position x y.

- `Vector3 _positionIn3DSpace`

Position in 3D space.

4.34.1 Constructor & Destructor Documentation

4.34.1.1 [Tile\(\)](#) [1/2]

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

Construct a new [Tile](#) object.

Parameters

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

4.34.1.2 [Tile\(\)](#) [2/2]

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

Construct a new [Tile](#) object.

Parameters

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

4.34.1.3 ~Tile()

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

Destroy the [Tile](#) object.

4.34.2 Member Function Documentation**4.34.2.1 getPosition()**

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

Get the Position object.

Returns

std::pair<std::size_t, std::size_t> - position x y

4.34.2.2 getPositionIn3DSpace()

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

Get the Position In Space object.

Returns

Vector3 - Position in space.

4.34.2.3 getTileBoundingBoxes()

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

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

Parameters

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

Returns

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

4.34.2.4 getTileModelHitbox()

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

Get the Tile Model Hitbox object.

Parameters

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

Returns

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

4.34.2.5 isTileHit()

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

Check if the tile is hit.

Parameters

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

Returns

true - The tile is hit.

false - The tile is not hit.

4.34.2.6 setPosition()

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

Set the Position object.

Parameters

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

4.34.3 Member Data Documentation

4.34.3.1 inventory

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

[Inventory](#) of the tile.

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

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

4.35 Gui::UserCamera Class Reference

Public Types

- enum **CameraType** {
FREE , **FIRST_PERSON** , **SECOND_PERSON** , **THIRD_PERSON** ,
FREE_TILE }

Public Member Functions

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

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

Private Attributes

- std::shared_ptr< Camera > **_camera**
- CameraType **_type**
- size_t **_playerId**
- std::pair< std::size_t, std::size_t > **_tilePos**

4.35.1 Constructor & Destructor Documentation

4.35.1.1 UserCamera()

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

Construct a new User Camera object.

4.35.1.2 ~UserCamera()

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

Destroy the User Camera object.

4.35.2 Member Function Documentation

4.35.2.1 getCamera()

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

Get the Camera object.

Returns

Camera - camera

4.35.2.2 getFovy()

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

Get the Fovy object.

Returns

float - fovy

4.35.2.3 getPlayerId()

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

Get the [Player](#) Id object.

Returns

size_t - [Player](#) id.

4.35.2.4 getPosition()

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

Get the Position object.

Returns

Vector3 - position

4.35.2.5 getTarget()

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

Get the Target object.

Returns

Vector3 - target

4.35.2.6 getTilePos()

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

Get the [Tile](#) position object.

Returns

std::pair<std::size_t, std::size_t> - Position of the tile.

4.35.2.7 getType()

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

Get the Type object.

Returns

CameraType - Camera type.

4.35.2.8 `getUp()`

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

Get the Up object.

Returns

Vector3 - up

4.35.2.9 `isPlayerPov()`

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

Check if the camera is in player pov.

Returns

true - Camera is in player pov.

false - Camera is not in player pov.

Note

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

4.35.2.10 `setFovy()`

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

Set the Fovy object.

Parameters

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

4.35.2.11 `setPlayerId()`

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

Set the [Player](#) Id object.

Parameters

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

4.35.2.12 setPosition()

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

Set the Position object.

Parameters

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

4.35.2.13 setTarget()

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

Set the Target object.

Parameters

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

4.35.2.14 setTilePos()

```
void Gui::UserCamera::setTilePos (
    std::pair< std::size_t, std::size_t > pos )
```

Set the [Tile](#) Pos object.

Parameters

<i>pos</i>	Position of the tile.
------------	-----------------------

4.35.2.15 setType()

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

Set the Type object.

Parameters

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

4.35.2.16 setUp()

```
void Gui::UserCamera::setUp (
    Vector3 up )
```

Set the Up object.

Parameters

<i>up</i>	New camera up vector.
-----------	-----------------------

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

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

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

```

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

```

5.2 Colors.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** Colors
00006 */
00007
00008 #pragma once
00009
00010 #include "raylib.h"
00011
00012 #include <vector>
00013
00014 #define STR_BLUE "\033[0;34m"
00015 #define STR_GREEN "\033[0;32m"
00016 #define STR_RED "\033[0;31m"
00017 #define STR_YELLOW "\033[0;33m"
00018 #define STR_VIOLET "\033[0;35m"
00019 #define STR_CYAN "\033[0;36m"
00020 #define STR_RESET "\033[0m"
00021
00022 static std::vector<Color> playerColors = {
00023     PINK,
00024     LIGHTGRAY,
00025     GRAY,
00026     DARKGRAY,
00027     YELLOW,
00028     GOLD,
00029     ORANGE,
00030     RED,
00031     MAROON,
00032     GREEN,
00033     LIME,
00034     DARKGREEN,
00035     SKYBLUE,
00036     BLUE,
00037     DARKBLUE,
00038     PURPLE,
00039     VIOLET,
00040     DARKPURPLE,
00041     BEIGE,
00042     BROWN,
00043     DARKBROWN,

```

```

00044     WHITE,
00045     BLACK,
00046     MAGENTA,
00047     RAYWHITE
00048 };

```

5.3 Config.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** Config
00006 */
00007
00008 #pragma once
00009
00010 #define SERVER_DOWN_MESSAGE      "THE GUI MUST BE CLOSED FOR UNEXPECTED REASONS"
00011
00012 #define SIZE_TILE                4.7
00013
00014 #define PLAYER_HEIGHT            2
00015
00016 #define DEFAULT_RENDER_DISTANCE 15
00017 #define MAX_RENDER_DISTANCE     20
00018 #define MIN_RENDER_DISTANCE     1
00019
00020 #define HIGH_CAMERA_INCREASE     0.1
00021 #define LOW_CAMERA_INCREASE     0.1
00022
00023 #define PLAYER_SECOND_PERSON_FOV 4.0f
00024 #define PLAYER_THIRD_PERSON_FOV 5.0f

```

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/INetwork.hpp"
00013 #include "GameDatas/GameData.hpp"
00014 #include "Parsing/ServerParser.hpp"
00015 #include "GUIUpdater/GUIUpdater.hpp"
00016
00017 #include <time.h>
00018
00019 #define TIME_UNIT_MAP_UPDATE 20
00020
00021 namespace Gui {
00022
00023     class Engine;
00024 };
00025
00026 class Gui::Engine {
00027 public:
00028
00029     Engine(std::shared_ptr<INetwork> network);
00030
00031     ~Engine() = default;
00032
00033     void run();
00034
00035 private:
00036
00037     std::unique_ptr<IServerParser> _parser;
00038     std::shared_ptr<INetwork> _network;
00039     std::shared_ptr<Render> _render;
00040     std::unique_ptr<IEvent> _event;
00041     std::shared_ptr<GameData> _gameData;
00042     std::unique_ptr<IGUIUpdater> _guiUpdater;
00043
00044 };

```

```

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

```

5.5 AError.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** AError
00006  */
00007
00008
00009 #pragma once
00010
00011 #include "Error/IError.hpp"
00012
00013 #include <string>
00014
00015 namespace Gui {
00016     namespace Errors {
00023         class AError;
00024     };
00025 };
00026
00027 class Gui::Errors::AError : public IError {
00028
00029     public:
00030
00034         ~AError() override = default;
00035
00040         const char *what() const noexcept override;
00041
00042     protected:
00043
00044         std::string _message;
00045 };

```

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
00019         class Error : public AError {};
00020
00025         class NetworkException : public Error {
00026
00027             public:
00033                 NetworkException(std::string message);
00034         };
00035
00040         class ServerParserException : public Error {
00041
00042             public:
00048                 ServerParserException(std::string message);
00049         };
00050
00055         class ParseCommandLineException : public Error {
00056
00057             public:
00063                 ParseCommandLineException(std::string message);
00064         };

```



```

00065
00070         class GuiGameDataException : public Error {
00071
00072             public:
00073                 GuiGameDataException(std::string message);
00074         };
00080
00085         class GuiUpdaterException : public Error {
00086
00087             public:
00093                 GuiUpdaterException(std::string message);
00094         };
00095     };
00096 };

```

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
00020     public:
00021
00022         virtual ~IError() = default;
00023
00024         virtual const char *what() const noexcept = 0;
00025 };

```

5.8 AEvent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** AEvent
00006 */
00007
00008 #pragma once
00009
00010 #include "Event/IEvent.hpp"
00011
00012 namespace Gui {
00013     class AEvent;
00014 }
00015
00016 class Gui::AEvent : public Gui::IEvent {
00017
00018     public:
00019
00020         AEvent();
00021
00022         ~AEvent() = default;
00023
00024         virtual void listen() = 0;
00025
00026         void setRender(std::shared_ptr<Render> render);
00027
00028         void setGameData(std::shared_ptr<GameData> gameData);
00029
00030     protected:
00031
00032         std::shared_ptr<Render> _render;
00033         std::shared_ptr<GameData> _gameData;
00034 };

```

5.9 Event.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** Event
00006  */
00007
00008  #pragma once
00009
00010  #include "Config.hpp"
00011  #include "Event/AEvent.hpp"
00012  #include "Render/Render.hpp"
00013
00014  #include <functional>
00015  #include <unordered_map>
00016
00017  namespace Gui {
00018
00023      class Event;
00024  };
00025
00026  class Gui::Event : public Gui::AEvent {
00027
00028      public:
00029
00034          Event();
00035
00040          ~Event() = default;
00041
00046          void listen();
00047
00048      private:
00049
00054          std::unordered_map<KeyboardKey, std::function<void()>> _eventsKeyDown =
00055          {
00056              {KEY_SPACE, [this]() {moveUpCamera();}},
00057              {KEY_LEFT_SHIFT, [this]() {moveDownCamera();}},
00058          };
00059
00064          std::unordered_map<KeyboardKey, std::function<void()>> _eventsKeyPressed =
00065          {
00066              {KEY_THREE, [this]() {switchDisplayDebug();}},
00067              {KEY_F3, [this]() {switchDisplayDebug();}},
00068              {KEY_SPACE, [this]() {setFreeCam();}},
00069              {KEY_R, [this]() {switchTileHudToGame();}},
00070              {KEY_J, [this]() {increaseRenderDistance();}},
00071              {KEY_K, [this]() {decreaseRenderDistance();}},
00072              {KEY_F5, [this]() {changeActualPlayerPov();}},
00073              {KEY_FOUR, [this]() {changeActualPlayerPov();}},
00074              {KEY_KP_ADD, [this]() {increaseTimeUnit();}},
00075              {KEY_KP_SUBTRACT, [this]() {decreaseTimeUnit();}},
00076          };
00077
00082          std::unordered_map<MouseButton, std::function<void()>> _eventsMousePressed =
00083          {
00084              {MOUSE_BUTTON_LEFT, [this]() {handleLeftClick();}},
00085              {MOUSE_BUTTON_RIGHT, [this]() {handleRightClick();}},
00086          };
00087
00092          void moveUpCamera();
00093
00098          void moveDownCamera();
00099
00104          void switchDisplayDebug();
00105
00109          void setFreeCam();
00110
00115          void handleLeftClick();
00116
00120          void handleRightClick();
00121
00126          void selectPlayer();
00127
00131          void selectTile();
00132
00138          void changePlayer(bool turn);
00139
00144          void changeActualPlayerPov();
00145
00150          void switchTileHudToGame();
00151
00156          void increaseRenderDistance();
00157
00162          void decreaseRenderDistance();
00163

```

```

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

```

5.10 IEvent.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** IEvent
00006  */
00007
00008  #pragma once
00009
00010  #include "Render/Render.hpp"
00011  #include "GameDatas/GameData.hpp"
00012  #include <memory>
00013
00014  namespace Gui {
00015
00020      class IEvent;
00021  }
00022
00023  class Gui::IEvent {
00024
00025      public:
00026
00031          IEvent() = default;
00032
00037          virtual ~IEvent() = default;
00038
00043          virtual void listen() = 0;
00044
00050          virtual void setRender(std::shared_ptr<Render> render) = 0;
00051
00057          virtual void setGameData(std::shared_ptr<GameData> gameData) = 0;
00058  };

```

5.11 Egg.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** Egg
00006  */
00007
00008  #pragma once
00009
00010  #include "Error/Error.hpp"
00011
00012  #include <string>
00013
00014  namespace Gui {
00015
00019      class Egg;
00020  };
00021
00022  class Gui::Egg {
00023
00024      public:
00025
00026          enum EggState {
00027              IDLE,
00028              DEAD,
00029              BORN,
00030              HATCHING,
00031          };
00032
00043          Egg(size_t id, const std::string &team, std::pair<std::size_t, std::size_t> position);
00044
00050          ~Egg();
00051
00057          std::size_t getId() const;
00058
00064          std::string getTeam() const;
00065

```

```

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

```

5.12 GameData.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** GameData
00006  */
00007
00008  #pragma once
00009
00010  #include "Types.hpp"
00011  #include "Error/Error.hpp"
00012  #include "GameDatas/Team.hpp"
00013  #include "GameDatas/Tile.hpp"
00014
00015  #define NO_TICK int(-1)
00016
00017  namespace Gui {
00018
00019      class GameData;
00020  };
00021
00022  class Gui::GameData {
00023  public:
00024
00025      enum TimeUnitState {
00026          INCREASE,
00027          DECREASE,
00028          NONE
00029      };
00030
00031      GameData();
00032
00033      ~GameData() = default;
00034
00035      std::vector<Gui::Team> &getTeams();
00036
00037      Gui::Team &getTeam(const std::string &name);
00038
00039      void addTeam(const Gui::Team &team);
00040
00041      void addTeam(const std::string &name, const std::string &playerModelPath, const std::string
&eggModelPath, Color playerColor);
00042
00043      void addPlayerToTeam(const std::string &teamName, const Gui::Player &player);
00044
00045      Gui::Player &getPlayer(size_t id);
00046
00047      Map<Gui::Tile> &getMap();
00048
00049      void setMap(const Map<Gui::Tile> &map);
00050
00051      void setMapSize(size_t x, size_t y);
00052
00053      std::pair<size_t, size_t> getMapSize() const;
00054
00055      Gui::Tile &getTile(size_t x, size_t y);
00056
00057      void setTile(const Gui::Tile &tile);
00058
00059      void restartLastTick(void);

```

```

00148
00154     void setServerTick(std::size_t tick);
00155
00161     clock_t getLastTick() const;
00162
00168     std::size_t getServerTick() const;
00169
00175     void setIsEndGame(bool isEndGame);
00176
00183     bool getIsEndGame() const;
00184
00190     void setLastError(const std::string &error);
00191
00197     std::string getLastError() const;
00198
00205     Team &getTeamById(std::size_t id);
00206
00212     TimeUnitState getTimeUnitFromServer() const;
00213
00219     void setTimeUnitFromServer(TimeUnitState timeUnitFromServer);
00220
00226     std::vector<Gui::Egg> &getServerEggs();
00227
00233     void addServerEgg(const Gui::Egg &egg);
00234
00240     void removeServerEgg(size_t id);
00241
00247     void setNbBCTCommandReceived(std::size_t nb);
00248
00254     std::size_t getNbBCTCommandReceived() const;
00255
00260     void restartLastTickMctCommand();
00261
00267     clock_t getLastTickMctCommand() const;
00268
00269     private:
00270
00271         std::vector<Gui::Team>      _teams;                // Teams of the game.
00272         Map<Gui::Tile>              _map;                  // Map of the game.
00273         std::size_t                 _serverTick;           // Tick value of the server.
00274         clock_t                     _lastTick;             // Last tick of the GameData (based on the
server tick).
00275         bool                        _isEndGame;            // Is true if the game is finished.
00276         std::size_t                 _nbBCTCommandReceived; // Number of bct command received.
00277         clock_t                     _lastTickMctCommand;  // Last tick when mct command is send.
00278         std::string                 _lastError;           // Last error message.
00279         TimeUnitState               _timeUnitFromServer;  // True if the time unit has changed.
00280         std::vector<Gui::Egg>       _serverEggs;          // Eggs from the server.
00281 };

```

5.13 Inventory.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Inventory
00006 */
00007
00008 #pragma once
00009
00010 #include <iostream>
00011
00012 #define RessourcesNumber 6
00013
00014 namespace Gui {
00015
00020     class Inventory;
00021 };
00022
00023 class Gui::Inventory {
00024
00025     public:
00026
00027         using Ressources = size_t [RessourcesNumber];
00028
00033         Inventory();
00034
00046         Inventory(std::size_t food, std::size_t linemate, std::size_t deraumere, std::size_t sibur,
std::size_t mendiane, std::size_t phiras, std::size_t thystame);
00047
00052         ~Inventory() = default;
00053

```

```

00059     void setFood(std::size_t food);
00060
00066     void setLinemate(std::size_t linemate);
00067
00073     void setDeraumere(std::size_t deraumere);
00074
00080     void setSibur(std::size_t sibur);
00081
00087     void setMendiane(std::size_t mendiane);
00088
00094     void setPhiras(std::size_t phiras);
00095
00101     void setThystame(std::size_t thytsame);
00102
00108     void setRessources(Ressources ressources);
00109
00115     std::size_t getFood(void);
00116
00122     std::size_t getLinemate(void);
00123
00129     std::size_t getDeraumere(void);
00130
00136     std::size_t getSibur(void);
00137
00143     std::size_t getMendiane(void);
00144
00150     std::size_t getPhiras(void);
00151
00157     std::size_t getThystame(void);
00158
00164     Ressources &getRessources(void);
00165
00179     void addResource(std::size_t resource, std::size_t quantity);
00180
00194     void removeResource(std::size_t resource, std::size_t quantity);
00195
00196 private:
00197
00198     std::size_t      _food;      // Food.
00199     Ressources      _ressources; // Ressources.
00200 };

```

5.14 Player.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Player
00006 */
00007
00008 #pragma once
00009
00010 #include "raylib.h"
00011 #include "GameDatas/Inventory.hpp"
00012
00013 namespace Gui {
00014
00019     class Player;
00020 };
00021
00022 class Gui::Player {
00023
00024     public:
00025
00026         enum PlayerState {
00027             IDLE = 2,
00028             BORN = 8,
00029             BROADCAST = 12,
00030             EJECT = 5,
00031             BEING_EJECTED = 15,
00032             EJECTED = 7,
00033             WALK = 6, // or 10
00034             INCANTATION = 0,
00035             LAY_EGG = 7,
00036             DROP = 9,
00037             COLLECT = 9,
00038             DEAD = 1,
00039         };
00040
00048         Player(std::size_t id, const std::string &team, std::pair<std::size_t, std::size_t> position,
00049             std::size_t orientation, std::size_t level = 1);
00049

```

```

00054     ~Player() = default;
00055
00061     void setPosition(std::pair<std::size_t, std::size_t> position);
00062
00068     void setPosition3D(Vector3 position3D);
00069
00075     void setId(std::size_t id);
00076
00082     void setLevel(std::size_t level);
00083
00089     void setOrientation(std::size_t orientation);
00090
00096     void setTeam(const std::string &team);
00097
00103     std::pair<std::size_t, std::size_t> getPosition(void) const;
00104
00110     Vector3 getPosition3D(void) const;
00111
00117     std::size_t getId(void) const;
00118
00124     std::size_t getLevel(void) const;
00125
00131     std::size_t getOrientation(void) const;
00132
00138     std::string getTeam(void) const;
00139
00145     void setState(PlayerState state);
00146
00152     PlayerState getState(void) const;
00153
00159     void setBroadcast(const std::string &broadcast);
00160
00166     std::string getBroadcast() const;
00167
00173     float getRotationFromOrientation() const;
00174
00180     Vector3 getCenterPosition();
00181
00187     void setCurrentFrame(int currentFrame);
00188
00194     int getCurrentFrame() const;
00195
00200     void restartAnimationTimeEllapsed();
00201
00207     clock_t getAnimationTimeEllapsed() const;
00208
00213     Inventory                                inventory;
00214
00215     private:
00216
00217         std::size_t                                _id;
00218         std::string                                _team;
00219         std::pair<std::size_t, std::size_t>        _position;
00220         Vector3                                    _position3D;
00221         std::size_t                                _orientation;
00222         std::size_t                                _level;
00223         PlayerState                                _state;
00224         std::string                                _broadcast;
00225         int                                         _currentFrame;
00226         clock_t                                    _animationTimeEllapsed;
00227 };

```

5.15 Team.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** Team
00006  */
00007
00008  #pragma once
00009
00010  #include "raylib.h"
00011  #include "Types.hpp"
00012  #include "GameDatas/Egg.hpp"
00013  #include "GameDatas/Tile.hpp"
00014  #include "GameDatas/Player.hpp"
00015
00016  #include <vector>
00017  #include <memory>
00018
00019  namespace Gui {

```

```

00020
00025     class Team;
00026 };
00027
00028 class Gui::Team {
00029
00030     public:
00031
00040         Team(const std::string &name, const std::string &playerModelPath, const std::string
&eggModelPath, Color playerColor);
00041
00046         ~Team();
00047
00053         const std::string &getName() const;
00054
00060         std::vector<Gui::Player> &getPlayers();
00061
00067         std::vector<Gui::Egg> &getEggs();
00068
00074         void setName(const std::string &name);
00075
00081         void addPlayer(const Gui::Player &player);
00082
00088         void addEgg(const Gui::Egg &egg);
00089
00097         bool removePlayer(std::size_t id);
00098
00106         bool removeEgg(std::size_t id);
00107
00114         std::shared_ptr<Gui::Player> getPlayer(std::size_t id);
00115
00121         Model getPlayerModel() const;
00122
00128         ModelAnimation *getPlayerModelAnimation() const;
00129
00135         void setPlayerModelPath(const std::string &playerModelPath);
00136
00143         std::shared_ptr<Gui::Egg> getEgg(std::size_t id);
00144
00150         Model getEggModel() const;
00151
00157         void setEggModelPath(const std::string &eggModelPath);
00158
00166         std::vector<BoundingBox> getPlayerBoundingBoxes(std::pair<size_t, size_t> pos, size_t
orientation, Vector3 center);
00167
00174         Vector3 getPlayerPositionIn3DSpace(size_t id, Map<Tile> map);
00175
00183         std::vector<RayCollision> getPlayerModelHitbox(size_t id, Camera camera);
00184
00192         bool isPlayerHit(size_t id, Camera camera);
00193
00198         Color getPlayerColor() const;
00199
00200     private:
00201
00202
00203         ModelAnimation*           _modelAnimation;
00204         int                       _animCount;
00205         std::string                _name;
00206         std::vector<Gui::Player>   _players;
00207         Model                     _playerModel;
00208         std::vector<Gui::Egg>      _eggs;
00209         Model                     _eggModel;
00210         Color                     _playerColor;
00211
00221         BoundingBox rotateBoundingBoxByOrientation(BoundingBox bbox, size_t orientation,
std::pair<size_t, size_t> pos, Vector3 center);
00222 };

```

5.16 Tile.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Tile
00006 */
00007
00008 #pragma once
00009
00010 #include "raylib.h"
00011 #include "GameDatas/Inventory.hpp"

```



```

00012
00013 #include <vector>
00014
00015 namespace Gui {
00016     class Tile;
00022 };
00023
00024 class Gui::Tile {
00025     public:
00027
00033     Tile(std::pair<std::size_t, std::size_t> position);
00034
00041     Tile(std::pair<std::size_t, std::size_t> position, Inventory inventory);
00042
00047     ~Tile() = default;
00048
00054     void setPosition(std::pair<std::size_t, std::size_t> position);
00055
00061     std::pair<std::size_t, std::size_t> getPosition() const;
00062
00068     Vector3 getPositionIn3DSpace();
00069
00076     std::vector<BoundingBox> getTileBoundingBoxes(Tile tile, Model tileModel);
00077
00085     std::vector<RayCollision> getTileModelHitbox(Tile tile, Camera camera, Model tileModel);
00086
00095     bool isTileHit(Camera camera, Model _tileModel);
00096
00101     Inventory inventory;
00102
00103     private:
00104
00105     std::pair<std::size_t, std::size_t> _position;
00106     Vector3 _positionIn3DSpace;
00107 };

```

5.17 AGUIUpdater.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** AGUIUpdater
00006 */
00007
00008 #pragma once
00009
00010 #include "Network/Network.hpp"
00011 #include "GameDatas/GameData.hpp"
00012 #include "GUIUpdater/IGUIUpdater.hpp"
00013
00014 #include <memory>
00015
00016 namespace Gui {
00017     class AGUIUpdater;
00022 }
00023
00024 class Gui::AGUIUpdater : public Gui::IGUIUpdater {
00025     public:
00026
00033     AGUIUpdater(std::shared_ptr<GameData> gameData, std::shared_ptr<INetwork> network);
00034
00038     ~AGUIUpdater() = default;
00039
00044     void update(const std::string &command, const std::vector<std::string> &data) override = 0;
00045
00046     protected:
00047
00048     std::shared_ptr<GameData> _gameData; // The GUI GameData to update.
00049     std::shared_ptr<INetwork> _network; // The network to send commands to the server.
00050 };

```

5.18 GUIUpdater.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024

```

```

00003  ** Zappy
00004  ** File description:
00005  ** GUIUpdater
00006  */
00007
00008  #pragma once
00009
00010  #include "GUIUpdater/AGUIUpdater.hpp"
00011
00012  #include <string>
00013  #include <functional>
00014  #include <unordered_map>
00015
00016  namespace Gui {
00017
00023      class GUIUpdater;
00024  }
00025
00026  class Gui::GUIUpdater : public Gui::AGUIUpdater {
00027
00028  public:
00029
00036      GUIUpdater(std::shared_ptr<GameData> gameData, std::shared_ptr<INetwork> network);
00037
00041      ~GUIUpdater() = default;
00042
00049      void update(const std::string &command, const std::vector<std::string> &data);
00050
00051  private:
00052
00053      size_t _colorIndex; // The index of the color to use for the team.
00054
00055      std::unordered_map<std::string, std::function<void(std::vector<std::string>)>> _updateMap =
00056      {
00057          {"msz", [this](std::vector<std::string> data){updateMapSize(data);}},
00058          {"bct", [this](std::vector<std::string> data){updateMapContent(data);}},
00059          {"tna", [this](std::vector<std::string> data){updateTeamNames(data);}},
00060          {"pnw", [this](std::vector<std::string> data){updateTeamMember(data);}},
00061          {"ppo", [this](std::vector<std::string> data){updatePlayerPosition(data);}},
00062          {"plv", [this](std::vector<std::string> data){updatePlayerLevel(data);}},
00063          {"pin", [this](std::vector<std::string> data){updatePlayerInventory(data);}},
00064          {"pex", [this](std::vector<std::string> data){updatePlayerExpulsion(data);}},
00065          {"pbc", [this](std::vector<std::string> data){updatePlayerBroadcast(data);}},
00066          {"pic", [this](std::vector<std::string> data){updatePlayerStartIncantation(data);}},
00067          {"pie", [this](std::vector<std::string> data){updatePlayerEndIncantation(data);}},
00068          {"pfk", [this](std::vector<std::string> data){updatePlayerEggLaying(data);}},
00069          {"pdr", [this](std::vector<std::string> data){updatePlayerRessourceDropping(data);}},
00070          {"pgt", [this](std::vector<std::string> data){updatePlayerRessourceCollecting(data);}},
00071          {"pdi", [this](std::vector<std::string> data){updatePlayerDeath(data);}},
00072          {"enw", [this](std::vector<std::string> data){updateEggLaidByPlayer(data);}},
00073          {"ebo", [this](std::vector<std::string> data){updatePlayerBorn(data);}},
00074          {"edi", [this](std::vector<std::string> data){updateEggDeath(data);}},
00075          {"sgt", [this](std::vector<std::string> data){updateTimeUnitRequest(data);}},
00076          {"sst", [this](std::vector<std::string> data){updateTimeUnitModification(data);}},
00077          {"seg", [this](std::vector<std::string> data){updateEndOfGame(data);}},
00078          {"smg", [this](std::vector<std::string> data){updateMessageFromServer(data);}},
00079          {"suc", [this](std::vector<std::string> data){updateUnknownMessage(data);}},
00080          {"sbp", [this](std::vector<std::string> data){updateCommandParameter(data);}},
00081      }; // The map of commands to update the GUI GameData.
00082
00088      void updateMapSize(const std::vector<std::string> &data);
00089
00095      void updateMapContent(const std::vector<std::string> &data);
00096
00102      void updateTeamNames(const std::vector<std::string> &data);
00103
00109      void updateTeamMember(const std::vector<std::string> &data);
00110
00116      void updatePlayerPosition(const std::vector<std::string> &data);
00117
00123      void updatePlayerLevel(const std::vector<std::string> &data);
00124
00130      void updatePlayerInventory(const std::vector<std::string> &data);
00131
00137      void updatePlayerExpulsion(const std::vector<std::string> &data);
00138
00144      void updatePlayerBroadcast(const std::vector<std::string> &data);
00145
00151      void updatePlayerStartIncantation(const std::vector<std::string> &data);
00152
00158      void updatePlayerEndIncantation(const std::vector<std::string> &data);
00159
00165      void updatePlayerEggLaying(const std::vector<std::string> &data);
00166
00172      void updatePlayerRessourceDropping(const std::vector<std::string> &data);
00173
00179      void updatePlayerRessourceCollecting(const std::vector<std::string> &data);

```

```

00180
00186     void updatePlayerDeath(const std::vector<std::string> &data);
00187
00193     void updateEggLaidByPlayer(const std::vector<std::string> &data);
00194
00200     void updatePlayerBorn(const std::vector<std::string> &data);
00201
00207     void updateEggDeath(const std::vector<std::string> &data);
00208
00214     void updateTimeUnitRequest(const std::vector<std::string> &data);
00215
00221     void updateTimeUnitModification(const std::vector<std::string> &data);
00222
00228     void updateEndOfGame(const std::vector<std::string> &data);
00229
00235     void updateMessageFromServer(const std::vector<std::string> &data);
00236
00242     void updateUnknownMessage(const std::vector<std::string> &data);
00243
00249     void updateCommandParameter(const std::vector<std::string> &data);
00250
00255     void increaseColorIndex();
00256 };

```

5.19 IGUIUpdater.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** IGUIUpdater
00006 */
00007
00008 #pragma once
00009
00010 #include <string>
00011 #include <vector>
00012
00013 namespace Gui {
00014
00019     class IGUIUpdater;
00020 }
00021
00022 class Gui::IGUIUpdater {
00023 public:
00027     virtual ~IGUIUpdater() = default;
00028
00033     virtual void update(const std::string &command, const std::vector<std::string> &data) = 0;
00034 };

```

5.20 AHud.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** AHud
00006 */
00007
00008 #pragma once
00009
00010 #include "Hud/IHud.hpp"
00011 #include "GameDatas/GameData.hpp"
00012
00013 namespace Gui {
00014
00019     class AHud;
00020 };
00021
00022 class Gui::AHud : public Gui::IHud {
00023 public:
00025
00030     ~AHud() = default;
00031
00036     virtual void display() = 0;
00037
00043     void setPlayer(std::shared_ptr<Player> player);
00044

```

```

00050         void setTile(std::shared_ptr<Tile> tile);
00051
00052         TypeScene getType() const;
00053
00054     protected:
00055
00056         TypeScene _typeScene; // Type of the scene.
00057         std::shared_ptr<GameData> _gameData; // GameData class.
00058         std::shared_ptr<Player> _player; // Player to display hud.
00059         std::shared_ptr<Tile> _tile; // Tile to display hud.
00060 };

```

5.21 HudGame.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** HudGame
00006 */
00007
00008 #pragma once
00009
00010 #include "Hud/AHud.hpp"
00011
00012 #define HUD_GAME_POS Vector2{0, 340}
00013 #define HUD_GAME_TEXT_POS Vector2{25, 420}
00014 #define HUD_GAME_TEXT_TITLE_POS Vector2{50, 380}
00015 #define HUD_GAME_TEXT_MARGING 30
00016
00017 namespace Gui {
00018
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     Texture2D _playerTexture; // Texture for player png.
00036 };

```

5.22 HudPlayer.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** HudPlayer
00006 */
00007
00008 #pragma once
00009
00010 #include "Hud/AHud.hpp"
00011
00012 #define HUD_PLAYER_POS Vector2{0, 340}
00013 #define HUD_PLAYER_TEXT_POS Vector2{55, 420}
00014 #define HUD_PLAYER_TEXT_TITLE_POS Vector2{80, 380}
00015 #define HUD_PLAYER_TEXT_MARGING 30
00016 #define HUD_PLAYER_ICONS_MARGING -32
00017 #define HUD_PLAYER_TITLE_ICON_MARGING Vector2{45, 35}
00018
00019 namespace Gui {
00020
00021     class HudPlayer;
00022 };
00023
00024 class Gui::HudPlayer : public Gui::AHud {
00025

```

```

00030     public:
00031
00037         HudPlayer(std::shared_ptr<GameData> gameData);
00038
00043         ~HudPlayer() = default;
00044
00049         void display();
00050
00051     private:
00052
00053         Texture2D _texture;          // Texture for Hud Background.
00054         Font _font;                  // Font for Hud's texts.
00055
00056         Texture2D _food;              // Texture for food png.
00057         Texture2D _linemate;          // Texture for linemate png.
00058         Texture2D _deraumere;         // Texture for deraumere png.
00059         Texture2D _mendiane;          // Texture for mendiane png.
00060         Texture2D _phiras;            // Texture for phiras png.
00061         Texture2D _sibur;             // Texture for sibur png.
00062         Texture2D _thystame;          // Texture for thystame png.
00063         Texture2D _playerTexture;     // Texture for player png.
00064 };

```

5.23 HudTile.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** HudTile
00006 */
00007
00008 #pragma once
00009
00010 #include "Hud/AHud.hpp"
00011
00012 #define HUD_TILE_POS Vector2{0, 340}
00013 #define HUD_TILE_TEXT_POS Vector2{55, 420}
00014 #define HUD_TILE_TEXT_TITLE_POS Vector2{60, 380}
00015 #define HUD_TILE_TEXT_MARGING 30
00016 #define HUD_TILE_ICONS_MARGING -32
00017 #define HUD_TILE_TITLE_ICON_MARGING Vector2{45, 40}
00018
00019 namespace Gui {
00020
00025     class HudTile;
00026 };
00027
00028 class Gui::HudTile : public Gui::AHud {
00029
00030     public:
00031
00037         HudTile(std::shared_ptr<GameData> gameData);
00038
00043         ~HudTile() = default;
00044
00049         void display();
00050
00055         void displayNbPlayers();
00056
00061         void displayNbEggs();
00062
00063     private:
00064
00065         Texture2D _texture;          // Texture for Hud Background.
00066         Font _font;                  // Font for Hud's texts.
00067
00068         Texture2D _food;              // Texture for food png.
00069         Texture2D _linemate;          // Texture for linemate png.
00070         Texture2D _deraumere;         // Texture for deraumere png.
00071         Texture2D _mendiane;          // Texture for mendiane png.
00072         Texture2D _phiras;            // Texture for phiras png.
00073         Texture2D _sibur;             // Texture for sibur png.
00074         Texture2D _thystame;          // Texture for thystame png.
00075         Texture2D _tileTexture;       // Texture for tile png.
00076         Texture2D _playerTexture;     // Texture for player png.
00077         Texture2D _eggTexture;        // Texture for egg png.
00078 };

```

5.24 IHud.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** IHud
00006  */
00007
00008  #pragma once
00009
00010  #include "GameDatas/Player.hpp"
00011  #include "GameDatas/Tile.hpp"
00012
00013  #include <memory>
00014
00015  namespace Gui {
00016
00021      class IHud;
00022  };
00023
00024  class Gui::IHud {
00025
00026      public:
00027
00032          enum TypeScene {
00033              GAME,
00034              POV_PLAYER,
00035              END_GAME,
00036              TILE
00037          };
00038
00043          virtual ~IHud() = default;
00044
00049          virtual void display() = 0;
00050
00056          virtual void setPlayer(std::shared_ptr<Player> player) = 0;
00057
00063          virtual void setTile(std::shared_ptr<Tile> tile) = 0;
00064
00070          virtual TypeScene getType() const = 0;
00071  };

```

5.25 ANetwork.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** ANetwork
00006  */
00007
00008  #pragma once
00009
00010  #include "Network/INetwork.hpp"
00011
00012  #define MIN_PORT 1
00013  #define MAX_PORT 65535
00014
00015  namespace Gui {
00016
00021      class ANetwork;
00022  };
00023
00024  class Gui::ANetwork : public Gui::INetwork {
00025      public:
00026
00033          ANetwork(int port, const std::string &hostName);
00034
00039          ~ANetwork() = default;
00040
00047          void setPort(int port) final;
00048
00054          void setHostName(const std::string &hostName) final;
00055
00061          int getPort() const final;
00062
00068          std::string getHostName() const final;
00069
00075          virtual void connectToServer() = 0;
00076
00082          virtual const std::string listenServer() = 0;
00083

```

```

00089         virtual void sendMessageServer(const std::string& message) = 0;
00090
00091     protected:
00092         int         _port;           // Port of the server.
00093         std::string  _hostName;      // Host name of the server.
00094 };

```

5.26 INetwork.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy
00004  ** File description:
00005  ** INetwork
00006  */
00007
00008 #pragma once
00009
00010 #include "Error/Error.hpp"
00011
00012 #include <string>
00013
00014 namespace Gui {
00015
00016     class INetwork;
00017 };
00018
00019 class Gui::INetwork {
00020
00021     public:
00022
00023         virtual ~INetwork() = default;
00024
00025         virtual void setPort(int port) = 0;
00026
00027         virtual void setHostName(const std::string &hostName) = 0;
00028
00029         virtual int getPort() const = 0;
00030
00031         virtual std::string getHostName() const = 0;
00032
00033         virtual void connectToServer() = 0;
00034
00035         virtual const std::string listenServer() = 0;
00036
00037         virtual void sendMessageServer(const std::string &message) = 0;
00038 };

```

5.27 Network.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** Network
00006  */
00007
00008 #pragma once
00009
00010 #include "Network/ANetwork.hpp"
00011
00012 #include <arpa/inet.h>
00013 #include <sys/socket.h>
00014 #include <netinet/in.h>
00015
00016 namespace Gui {
00017
00018     class Network;
00019 };
00020
00021 class Gui::Network : public Gui::ANetwork {
00022
00023     public:
00024
00025         Network(int port, const std::string &hostName);
00026
00027         ~Network();
00028
00029         void connectToServer();

```

```

00049
00055     const std::string listenServer();
00056
00062     void sendMessageServer(const std::string& message);
00063
00064     private:
00065
00070     void selectServer();
00071
00077     const std::string readInfoServer();
00078
00079     int             _serverFd;
00080     fd_set           _writeFd;
00081     fd_set           _readFd;
00082     bool             _isConnected;
00083 };

```

5.28 IServerParser.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** IServerParser
00006 */
00007
00008 #pragma once
00009
00010 #include <string>
00011 #include <vector>
00012
00013 namespace Gui {
00014
00018     class IServerParser;
00019 }
00020
00021 class Gui::IServerParser {
00022
00023     public:
00024
00028     virtual ~IServerParser() = default;
00029
00036     virtual std::vector<std::string> parse(const std::string& command) = 0;
00037 };

```

5.29 ParseCommandLine.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** ParseCommandLine
00006 */
00007
00008 #pragma once
00009
00010 #include <string>
00011
00012 #define GUI_USAGE "USAGE:\t./zappy_gui -p port -h machine"
00013
00014 namespace Gui {
00015
00020     class ParseCommandLine;
00021 };
00022
00023 class Gui::ParseCommandLine {
00024
00025     public:
00026
00033     ParseCommandLine(int argc, char **argv);
00034
00039     ~ParseCommandLine() = default;
00040
00047     void parseFlags(int argc, char **argv);
00048
00054     int getPort(void);
00055
00061     std::string getHostName(void);
00062

```



```

00063     private:
00064
00065         int          _port;          // port server
00066         std::string   _hostName;     // host name server
00067 };

```

5.30 ServerParser.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Parse server command
00006 */
00007
00008 #pragma once
00009
00010 #include "Error/Error.hpp"
00011 #include "Parsing/IServerParser.hpp"
00012
00013 #include <functional>
00014 #include <unordered_map>
00015
00016 namespace Gui {
00017
00022     class ServerParser;
00023 };
00024
00025 class Gui::ServerParser : public Gui::IServerParser {
00026
00027     public:
00028
00033         ServerParser() = default;
00034
00039         ~ServerParser() = default;
00040
00047         std::vector<std::string> parse(const std::string& command);
00048
00049     private:
00050
00055         enum ParseType {
00056             INT,
00057             STRING,
00058             MESSAGE,
00059             LIST_INT
00060         };
00061
00066         std::unordered_map<std::string, std::vector<ParseType> _typesCommand =
00067         {
00068             {"msz", std::vector<ParseType>{INT, INT}},
00069             {"bct", std::vector<ParseType>{INT, INT, INT, INT, INT, INT, INT, INT, INT, INT}},
00070             {"tna", std::vector<ParseType>{STRING}},
00071             {"pnw", std::vector<ParseType>{INT, INT, INT, INT, INT, STRING}},
00072             {"ppo", std::vector<ParseType>{INT, INT, INT, INT}},
00073             {"plv", std::vector<ParseType>{INT, INT}},
00074             {"pin", std::vector<ParseType>{INT, INT, INT, INT, INT, INT, INT, INT, INT, INT}},
00075             {"pex", std::vector<ParseType>{INT}},
00076             {"pbc", std::vector<ParseType>{INT, MESSAGE}},
00077             {"pic", std::vector<ParseType>{INT, INT, INT, LIST_INT}},
00078             {"pie", std::vector<ParseType>{INT, INT, INT}},
00079             {"pfr", std::vector<ParseType>{INT}},
00080             {"pdr", std::vector<ParseType>{INT, INT}},
00081             {"pgt", std::vector<ParseType>{INT, INT}},
00082             {"pdi", std::vector<ParseType>{INT}},
00083             {"enw", std::vector<ParseType>{INT, INT, INT, INT}},
00084             {"ebo", std::vector<ParseType>{INT}},
00085             {"edi", std::vector<ParseType>{INT}},
00086             {"sgt", std::vector<ParseType>{INT}},
00087             {"sst", std::vector<ParseType>{INT}},
00088             {"seg", std::vector<ParseType>{STRING}},
00089             {"smg", std::vector<ParseType>{MESSAGE}},
00090             {"suc", std::vector<ParseType>{}},
00091             {"sbp", std::vector<ParseType>{}}
00092         };
00093
00101         std::vector<std::string> parseCommand(const std::string& command, std::vector<ParseType>
00102         types);
00110         std::vector<std::string> parseInt(std::istream& stream, std::vector<std::string>
00111         arguments);
00119         std::vector<std::string> parseString(std::istream& stream, std::vector<std::string>
00120         arguments);

```

```

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

```

5.31 Decoration.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** Decoration
00006 */
00007
00008 #pragma once
00009
00010 #include "raylib.h"
00011 #include "Types.hpp"
00012 #include "Assets.hpp"
00013
00014 #include <vector>
00015 #include <iostream>
00016
00017 namespace Gui {
00018
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
00067         std::pair<std::size_t, std::size_t> _mapSize; // Size of the map.
00068
00076         void displayTree(size_t i, size_t j, Vector3 posTile);
00077 };

```

5.32 Render.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy GUI
00004 ** File description:
00005 ** Render
00006 */
00007
00008 #pragma once
00009
00010 #define WINDOW_WIDTH 1920
00011 #define WINDOW_HEIGHT 1080
00012 #define WINDOW_TITLE "Zappy GUI"
00013
00014 #include "raylib.h"
00015 #include "Hud/HudGame.hpp"
00016 #include "Hud/HudTile.hpp"
00017 #include "Hud/HudPlayer.hpp"
00018 #include "Render/Decoration.hpp"
00019 #include "Render/UserCamera.hpp"
00020 #include "GameDatas/GameData.hpp"
00021
00022 #include <functional>
00023 #include <unordered_map>

```

```

00024
00025 namespace Gui {
00026
00031     class Render;
00032 };
00033
00034 class Gui::Render {
00035
00036     public:
00037
00042         Render(std::shared_ptr<GameData> gameData);
00043
00048         ~Render();
00049
00056         bool isOpen();
00057
00062         void draw();
00063
00069         std::shared_ptr<Camera> getCamera();
00070
00076         void setIsDebug(bool isDebug);
00077
00084         bool getIsDebug(void);
00085
00091         void setCameraType(Gui::UserCamera::CameraType type);
00092
00098         Gui::UserCamera::CameraType getCameraType() const;
00099
00105         void setCameraPlayerPov(std::size_t id);
00106
00112         std::size_t getCameraPlayerPov() const;
00113
00119         void setCameraTile(std::pair<std::size_t, std::size_t> pos);
00120
00126         std::pair<std::size_t, std::size_t> getCameraTile() const;
00127
00132         Model getTileModel() const;
00133
00139         void setRenderDistance(size_t renderDistance);
00140
00145         size_t getRenderDistance() const;
00146
00154         bool isCameraInPlayerPov() const;
00155
00162         void changePlayerPOV(size_t playerId);
00163
00169         void setPlayerPov(size_t playerId);
00170
00176         void changePOVToFirstPerson(size_t id);
00177
00183         void changePOVToSecondPerson(size_t id);
00184
00190         void changePOVToThirdPerson(size_t id);
00191
00197         size_t getTimeUnit() const;
00198
00204         void setTimeUnit(size_t timeUnit);
00205
00206     private:
00207
00208         UserCamera          _camera;          // Camera of the scene.
00209         bool                 _isDebug;         // Display or not the debug
00210         informations.
00210         std::shared_ptr<GameData> _gameData;    // GameData class to store the
00211         game's data.
00211         std::shared_ptr<Decoration> _decoration; // Decoration to display;
00212         std::vector<std::shared_ptr<Gui::IHud> _hudList; // List of huds.
00213         size_t                 _renderDistance; // Distance to render from the
00214         3d position of the camera.
00214
00215         Model                 _tileModel;      // Model to display tiles.
00216         Model                 _foodModel;      // Model to display foods.
00217         Model                 _linemateModel;  // Model to display linemates.
00218         Model                 _mendianeModel;  // Model to display mendianes.
00219         Model                 _phirasModel;    // Model to display phiras.
00220         Model                 _siburModel;     // Model to display siburs.
00221         Model                 _thystameModel;  // Model to display thystames.
00222         Model                 _deraumereModel; // Model to display
00223         deraumeres.
00223         Texture2D             _cursorTexture; // Cursor texture.
00224
00229         void LoadModels();
00230
00235         void displayHUD();
00236
00241         void displayDebug();
00242

```

```

00247     void displayPlayers();
00248
00255     void displayPlayerLevel(Team &team, Player &player);
00256
00263     void displayPlayerBroadcast(Team &team, Player &player);
00264
00269     void displayMap();
00270
00275     void displayTile(Tile tile);
00276
00282     void displayEggs(Tile tile) const;
00283
00289     void displayFood(Tile tile) const;
00290
00296     void displayResources(Tile tile) const;
00297
00303     void displayLinemate(Tile tile) const;
00304
00310     void displayMendiane(Tile tile) const;
00311
00317     void displayPhiras(Tile tile) const;
00318
00324     void displaySibur(Tile tile) const;
00325
00331     void displayThystame(Tile tile) const;
00332
00338     void displayDeraumere(Tile tile) const;
00339
00346     bool displayAnimations(Team &team, Player &player);
00347
00356     ModelAnimation displayWalkAnimation(Team &team, Player &player, ModelAnimation anim);
00357
00362     void displayCursor();
00363
00369     std::pair<std::size_t, std::size_t> getCameraTile();
00370 };

```

5.33 UserCamera.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2024
00003  ** Zappy GUI
00004  ** File description:
00005  ** Camera
00006  */
00007
00008 #pragma once
00009
00010 #include "raylib.h"
00011
00012 #include <memory>
00013
00014 namespace Gui {
00015
00020     class UserCamera;
00021 };
00022
00023 class Gui::UserCamera {
00024
00025     public:
00026
00027         enum CameraType {
00028             FREE,
00029             FIRST_PERSON,
00030             SECOND_PERSON,
00031             THIRD_PERSON,
00032             FREE_TILE
00033         };
00034
00039         UserCamera();
00040
00045         ~UserCamera() = default;
00046
00052         void setPosition(Vector3 position);
00053
00059         void setTarget(Vector3 target);
00060
00066         void setUp(Vector3 up);
00067
00073         void setFovy(float fovy);
00074
00080         Vector3 getPosition(void) const;
00081

```

```

00087     Vector3 getTarget(void) const;
00088
00094     Vector3 getUp(void) const;
00095
00101     float getFovy(void) const;
00102
00108     std::shared_ptr<Camera> getCamera();
00109
00115     void setType(CameraType type);
00116
00122     CameraType getType() const;
00123
00129     void setPlayerId(size_t playerId);
00130
00136     size_t getPlayerId() const;
00137
00143     void setTilePos(std::pair<std::size_t, std::size_t> pos);
00144
00150     std::pair<std::size_t, std::size_t> getTilePos() const;
00151
00159     bool isPlayerPov() const;
00160
00161     private:
00162
00163         std::shared_ptr<Camera> _camera;           // Camera raylib instance.
00164         CameraType _type;                          // Type of camera.
00165         size_t _playerId;                          // Player id.
00166         std::pair<std::size_t, std::size_t> _tilePos; // Tile position.
00167 };

```

5.34 Types.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2024
00003 ** Zappy
00004 ** File description:
00005 ** Types
00006 */
00007
00008 #pragma once
00009
00010 #include <vector>
00011
00017 template<typename T>
00018 using Map = std::vector<std::vector<T>>;

```


Index

/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Assets.hpp, 152
135 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/INetwork.I
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Colors.hpp, 153
136 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/Network.h
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Config.hpp, 153
137 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/IServerPar
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Engine/Engine.hpp,
137 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ParseCom
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/AError.hpp,
138 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Parsing/ServerPar
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/Error.hpp,
138 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Decoration
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Error/IError.hpp,
139 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/Render.hpp
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/AEvent.hpp,
139 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Render/UserCame
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/Event.hpp,
140 /home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Types.hpp,
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Event/IEvent.hpp,
141 _eventsKeyDown
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdate/GuiUpdater.hpp,
147 _eventsKeyPressed
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdate/GuiUpdater.hpp,
147 _eventsMousePressed
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GUIUpdate/GuiUpdater.hpp,
149 _typesCommand
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/ServerParser, 112
141 _updateMap
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/GuiUpdater, 53
142 ~AHud
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/Inventory, 11
143 ~ANetwork
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/Player, 13
144 ~Decoration
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/Decoration, 17
145 ~Egg
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/GameData/Flag, 19
146 ~Engine
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/AHud, 23
149 ~Event
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGame, 27
150 ~GameData
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudGameData, 32
150 ~HudGame
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/HudPlayer, 56
151 ~HudPlayer
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Hud/IHud, 58
152 ~HudTile
/home/tjerome-rocher/Desktop/Tek2/Zappy/gui/include/Network/ClientFile, 61

- ~IEvent
 - Gui::IEvent, [63](#)
- ~IHud
 - Gui::IHud, [65](#)
- ~INetwork
 - Gui::INetwork, [67](#)
- ~Inventory
 - Gui::Inventory, [71](#)
- ~Network
 - Gui::Network, [79](#)
- ~ParseCommandLine
 - Gui::ParseCommandLine, [83](#)
- ~Player
 - Gui::Player, [88](#)
- ~Render
 - Gui::Render, [97](#)
- ~ServerParser
 - Gui::ServerParser, [109](#)
- ~Team
 - Gui::Team, [116](#)
- ~Tile
 - Gui::Tile, [126](#)
- ~UserCamera
 - Gui::UserCamera, [129](#)
- addEgg
 - Gui::Team, [117](#)
- addPlayer
 - Gui::Team, [117](#)
- addPlayerToTeam
 - Gui::GameData, [33](#)
- addResource
 - Gui::Inventory, [71](#)
- addServerEgg
 - Gui::GameData, [33](#)
- addTeam
 - Gui::GameData, [33](#), [34](#)
- AGUIUpdater
 - Gui::AGUIUpdater, [9](#)
- ANetwork
 - Gui::ANetwork, [13](#)
- changeActualPlayerPov
 - Gui::Event, [27](#)
- changePlayer
 - Gui::Event, [27](#)
- changePlayerPOV
 - Gui::Render, [97](#)
- changePOVToFirstPerson
 - Gui::Render, [97](#)
- changePOVToSecondPerson
 - Gui::Render, [98](#)
- changePOVToThirdPerson
 - Gui::Render, [98](#)
- connectToServer
 - Gui::ANetwork, [14](#)
 - Gui::INetwork, [67](#)
 - Gui::Network, [79](#)
- Decoration
 - Gui::Decoration, [17](#)
- decreaseRenderDistance
 - Gui::Event, [28](#)
- decreaseTimeUnit
 - Gui::Event, [28](#)
- display
 - Gui::AHud, [11](#)
 - Gui::Decoration, [17](#)
 - Gui::HudGame, [56](#)
 - Gui::HudPlayer, [59](#)
 - Gui::HudTile, [61](#)
 - Gui::IHud, [65](#)
- displayAnimations
 - Gui::Render, [98](#)
- displayCursor
 - Gui::Render, [98](#)
- displayDebug
 - Gui::Render, [99](#)
- displayDeraumere
 - Gui::Render, [99](#)
- displayEggs
 - Gui::Render, [99](#)
- displayFood
 - Gui::Render, [99](#)
- displayHUD
 - Gui::Render, [100](#)
- displayLinemate
 - Gui::Render, [100](#)
- displayMap
 - Gui::Render, [100](#)
- displayMendiane
 - Gui::Render, [100](#)
- displayNbEggs
 - Gui::HudTile, [61](#)
- displayNbPlayers
 - Gui::HudTile, [61](#)
- displayPhiras
 - Gui::Render, [100](#)
- displayPlayerBroadcast
 - Gui::Render, [101](#)
- displayPlayerLevel
 - Gui::Render, [101](#)
- displayPlayers
 - Gui::Render, [101](#)
- displayResources
 - Gui::Render, [101](#)
- displaySibur
 - Gui::Render, [102](#)
- displayThystame
 - Gui::Render, [102](#)
- displayTile
 - Gui::Render, [102](#)
- displayTree
 - Gui::Decoration, [17](#)
- displayWalkAnimation
 - Gui::Render, [102](#)
- draw

- Gui::Render, 103
- Egg
 - Gui::Egg, 19
- Engine
 - Gui::Engine, 22
- Event
 - Gui::Event, 27
- GameData
 - Gui::GameData, 32
- getAnimationTimeEllapsed
 - Gui::Player, 88
- getBroadcast
 - Gui::Player, 88
- getCamera
 - Gui::Render, 103
 - Gui::UserCamera, 130
- getCameraPlayerPov
 - Gui::Render, 103
- getCameraTile
 - Gui::Render, 103, 104
- getCameraType
 - Gui::Render, 104
- getCenterPosition
 - Gui::Player, 88
- getCurrentFrame
 - Gui::Player, 89
- getDeraumere
 - Gui::Inventory, 72
- getEgg
 - Gui::Team, 117
- getEggModel
 - Gui::Team, 117
- getEggs
 - Gui::Team, 118
- getFood
 - Gui::Inventory, 72
- getFovy
 - Gui::UserCamera, 130
- getGenerationItem
 - Gui::Decoration, 18
- getHostName
 - Gui::ANetwork, 14
 - Gui::INetwork, 68
 - Gui::ParseCommandLine, 83
- getId
 - Gui::Egg, 20
 - Gui::Player, 89
- getIsDebug
 - Gui::Render, 104
- getIsEndGame
 - Gui::GameData, 34
- getLastError
 - Gui::GameData, 34
- getLastTick
 - Gui::GameData, 34
- getLastTickMctCommand
 - Gui::GameData, 35
- getLevel
 - Gui::Player, 89
- getLinemate
 - Gui::Inventory, 72
- getMap
 - Gui::GameData, 35
- getMapSize
 - Gui::GameData, 35
- getMendiane
 - Gui::Inventory, 72
- getName
 - Gui::Team, 118
- getNbBCTCommandReceived
 - Gui::GameData, 35
- getOrientation
 - Gui::Player, 89
- getPhiras
 - Gui::Inventory, 73
- getPlayer
 - Gui::GameData, 36
 - Gui::Team, 118
- getPlayerBoundingBoxes
 - Gui::Team, 119
- getPlayerColor
 - Gui::Team, 119
- getPlayerId
 - Gui::UserCamera, 130
- getPlayerModel
 - Gui::Team, 119
- getPlayerModelAnimation
 - Gui::Team, 119
- getPlayerModelHitbox
 - Gui::Team, 120
- getPlayerPositionIn3DSpace
 - Gui::Team, 120
- getPlayers
 - Gui::Team, 120
- getPort
 - Gui::ANetwork, 14
 - Gui::INetwork, 68
 - Gui::ParseCommandLine, 83
- getPosition
 - Gui::Egg, 20
 - Gui::Player, 90
 - Gui::Tile, 126
 - Gui::UserCamera, 130
- getPosition3D
 - Gui::Player, 90
- getPositionIn3DSpace
 - Gui::Tile, 126
- getRenderDistance
 - Gui::Render, 104
- getRessources
 - Gui::Inventory, 73
- getRotationFromOrientation
 - Gui::Player, 90
- getServerEggs
 - Gui::GameData, 37

- getServerTick
 - Gui::GameData, 37
- getSibur
 - Gui::Inventory, 73
- getState
 - Gui::Egg, 20
 - Gui::Player, 90
- getTarget
 - Gui::UserCamera, 131
- getTeam
 - Gui::Egg, 20
 - Gui::GameData, 37
 - Gui::Player, 91
- getTeamById
 - Gui::GameData, 37
- getTeams
 - Gui::GameData, 38
- getThystame
 - Gui::Inventory, 73
- getTile
 - Gui::GameData, 38
- getTileBoundingBoxes
 - Gui::Tile, 126
- getTileModel
 - Gui::Render, 105
- getTileModelHitbox
 - Gui::Tile, 127
- getTilePos
 - Gui::UserCamera, 131
- getTimeUnit
 - Gui::Render, 105
- getTimeUnitFromServer
 - Gui::GameData, 38
- getType
 - Gui::AHud, 11
 - Gui::IHud, 66
 - Gui::UserCamera, 131
- getUp
 - Gui::UserCamera, 131
- Gui::AEvent, 8
- Gui::AGUIUpdater, 8
 - AGUIUpdater, 9
 - update, 9
- Gui::AHud, 10
 - ~AHud, 11
 - display, 11
 - getType, 11
 - setPlayer, 11
 - setTile, 12
- Gui::ANetwork, 12
 - ~ANetwork, 13
 - ANetwork, 13
 - connectToServer, 14
 - getHostName, 14
 - getPort, 14
 - listenServer, 14
 - sendMessageServer, 15
 - setHostName, 15
 - setPort, 15
- Gui::Decoration, 16
 - ~Decoration, 17
 - Decoration, 17
 - display, 17
 - displayTree, 17
 - getGenerationItem, 18
- Gui::Egg, 18
 - ~Egg, 19
 - Egg, 19
 - getId, 20
 - getPosition, 20
 - getState, 20
 - getTeam, 20
 - setId, 20
 - setPosition, 21
 - setState, 21
 - setTeam, 21
- Gui::Engine, 22
 - ~Engine, 23
 - Engine, 22
 - listenServer, 23
 - run, 23
 - sendMessageUpdate, 23
 - sendUpdateMapMessage, 23
 - updateMap, 24
- Gui::Errors::AError, 7
 - what, 8
- Gui::Errors::Error, 24
- Gui::Errors::GuiGameDataException, 42
 - GuiGameDataException, 43
- Gui::Errors::GuiUpdaterException, 54
 - GuiUpdaterException, 54
- Gui::Errors::IError, 62
 - what, 62
- Gui::Errors::NetworkException, 81
 - NetworkException, 82
- Gui::Errors::ParseCommandLineException, 84
 - ParseCommandLineException, 85
- Gui::Errors::ServerParserException, 113
 - ServerParserException, 114
- Gui::Event, 25
 - _eventsKeyDown, 30
 - _eventsKeyPressed, 30
 - _eventsMousePressed, 30
 - ~Event, 27
 - changeActualPlayerPov, 27
 - changePlayer, 27
 - decreaseRenderDistance, 28
 - decreaseTimeUnit, 28
 - Event, 27
 - handleLeftClick, 28
 - increaseRenderDistance, 28
 - increaseTimeUnit, 28
 - listen, 28
 - moveDownCamera, 29
 - moveUpCamera, 29
 - selectPlayer, 29

- switchDisplayDebug, 29
- switchTileHudToGame, 29
- Gui::GameData, 31
 - ~GameData, 32
 - addPlayerToTeam, 33
 - addServerEgg, 33
 - addTeam, 33, 34
 - GameData, 32
 - getIsEndGame, 34
 - getLastError, 34
 - getLastTick, 34
 - getLastTickMctCommand, 35
 - getMap, 35
 - getMapSize, 35
 - getNbBCTCommandReceived, 35
 - getPlayer, 36
 - getServerEggs, 37
 - getServerTick, 37
 - getTeam, 37
 - getTeamByld, 37
 - getTeams, 38
 - getTile, 38
 - getTimeUnitFromServer, 38
 - removeServerEgg, 39
 - restartLastTick, 39
 - restartLastTickMctCommand, 39
 - setIsEndGame, 39
 - setLastError, 40
 - setMap, 40
 - setMapSize, 40
 - setNbBCTCommandReceived, 41
 - setServerTick, 41
 - setTile, 41
 - setTimeUnitFromServer, 41
- Gui::GUIUpdater, 43
 - _updateMap, 53
 - GUIUpdater, 45
 - increaseColorIndex, 46
 - update, 46
 - updateCommandParameter, 46
 - updateEggDeath, 46
 - updateEggLaidByPlayer, 47
 - updateEndOfGame, 47
 - updateMapContent, 47
 - updateMapSize, 47
 - updateMessageFromServer, 48
 - updatePlayerBorn, 48
 - updatePlayerBroadcast, 48
 - updatePlayerDeath, 49
 - updatePlayerEggLaying, 49
 - updatePlayerEndIncantation, 49
 - updatePlayerExpulsion, 49
 - updatePlayerInventory, 50
 - updatePlayerLevel, 50
 - updatePlayerPosition, 50
 - updatePlayerRessourceCollecting, 51
 - updatePlayerRessourceDropping, 51
 - updatePlayerStartIncantation, 51
 - updateTeamMember, 51
 - updateTeamNames, 52
 - updateTimeUnitModification, 52
 - updateTimeUnitRequest, 52
 - updateUnknownMessage, 53
- Gui::HudGame, 55
 - ~HudGame, 56
 - display, 56
 - HudGame, 56
- Gui::HudPlayer, 57
 - ~HudPlayer, 58
 - display, 59
 - HudPlayer, 58
- Gui::HudTile, 59
 - ~HudTile, 61
 - display, 61
 - displayNbEggs, 61
 - displayNbPlayers, 61
 - HudTile, 60
- Gui::IEvent, 62
 - ~IEvent, 63
 - IEvent, 63
 - listen, 63
 - setGameData, 63
 - setRender, 63
- Gui::IGUIUpdater, 64
 - update, 64
- Gui::IHud, 65
 - ~IHud, 65
 - display, 65
 - getType, 66
 - setPlayer, 66
 - setTile, 66
 - TypeScene, 65
- Gui::INetwork, 67
 - ~INetwork, 67
 - connectToServer, 67
 - getHostName, 68
 - getPort, 68
 - listenServer, 68
 - sendMessageServer, 68
 - setHostName, 69
 - setPort, 69
- Gui::Inventory, 69
 - ~Inventory, 71
 - addResource, 71
 - getDeraumere, 72
 - getFood, 72
 - getLinemate, 72
 - getMendiane, 72
 - getPhiras, 73
 - getRessources, 73
 - getSibur, 73
 - getThystame, 73
 - Inventory, 71
 - removeResource, 74
 - setDeraumere, 74
 - setFood, 74

- setLinemate, 75
- setMendiane, 75
- setPhiras, 75
- setRessources, 75
- setSibur, 76
- setThystame, 76
- Gui::IServerParser, 76
 - parse, 77
- Gui::Network, 77
 - ~Network, 79
 - connectToServer, 79
 - listenServer, 80
 - Network, 79
 - readInfoServer, 80
 - selectServer, 80
 - sendMessageServer, 80
- Gui::ParseCommandLine, 82
 - ~ParseCommandLine, 83
 - getHostName, 83
 - getPort, 83
 - ParseCommandLine, 83
 - parseFlags, 84
- Gui::Player, 86
 - ~Player, 88
 - getAnimationTimeEllapsed, 88
 - getBroadcast, 88
 - getCenterPosition, 88
 - getCurrentFrame, 89
 - getId, 89
 - getLevel, 89
 - getOrientation, 89
 - getPosition, 90
 - getPosition3D, 90
 - getRotationFromOrientation, 90
 - getState, 90
 - getTeam, 91
 - inventory, 94
 - Player, 87
 - restartAnimationTimeEllapsed, 91
 - setBroadcast, 91
 - setCurrentFrame, 91
 - setId, 92
 - setLevel, 92
 - setOrientation, 92
 - setPosition, 93
 - setPosition3D, 93
 - setState, 93
 - setTeam, 93
- Gui::Render, 94
 - ~Render, 97
 - changePlayerPOV, 97
 - changePOVToFirstPerson, 97
 - changePOVToSecondPerson, 98
 - changePOVToThirdPerson, 98
 - displayAnimations, 98
 - displayCursor, 98
 - displayDebug, 99
 - displayDeraumere, 99
 - displayEggs, 99
 - displayFood, 99
 - displayHUD, 100
 - displayLinemate, 100
 - displayMap, 100
 - displayMendiane, 100
 - displayPhiras, 100
 - displayPlayerBroadcast, 101
 - displayPlayerLevel, 101
 - displayPlayers, 101
 - displayResources, 101
 - displaySibur, 102
 - displayThystame, 102
 - displayTile, 102
 - displayWalkAnimation, 102
 - draw, 103
 - getCamera, 103
 - getCameraPlayerPov, 103
 - getCameraTile, 103, 104
 - getCameraType, 104
 - getIsDebug, 104
 - getRenderDistance, 104
 - getTileModel, 105
 - getTimeUnit, 105
 - isCameraInPlayerPov, 105
 - isOpen, 105
 - LoadModels, 106
 - Render, 96
 - setCameraPlayerPov, 106
 - setCameraTile, 106
 - setCameraType, 106
 - setIsDebug, 107
 - setPlayerPov, 107
 - setRenderDistance, 107
 - setTimeUnit, 108
- Gui::ServerParser, 108
 - _typesCommand, 112
 - ~ServerParser, 109
 - parse, 110
 - parseCommand, 110
 - parseInt, 110
 - parseListInt, 111
 - parseMessage, 111
 - parseString, 112
 - ParseType, 109
 - ServerParser, 109
- Gui::Team, 114
 - ~Team, 116
 - addEgg, 117
 - addPlayer, 117
 - getEgg, 117
 - getEggModel, 117
 - getEggs, 118
 - getName, 118
 - getPlayer, 118
 - getPlayerBoundingBoxes, 119
 - getPlayerColor, 119
 - getPlayerModel, 119

- getPlayerModelAnimation, 119
- getPlayerModelHitbox, 120
- getPlayerPositionIn3DSpace, 120
- getPlayers, 120
- isPlayerHit, 121
- removeEgg, 121
- removePlayer, 121
- rotateBoundingBoxByOrientation, 123
- setEggModelPath, 123
- setName, 123
- setPlayerModelPath, 124
- Team, 116
- Gui::Tile, 124
 - ~Tile, 126
 - getPosition, 126
 - getPositionIn3DSpace, 126
 - getTileBoundingBoxes, 126
 - getTileModelHitbox, 127
 - inventory, 128
 - isTileHit, 127
 - setPosition, 127
 - Tile, 125
- Gui::UserCamera, 128
 - ~UserCamera, 129
 - getCamera, 130
 - getFovy, 130
 - getPlayerId, 130
 - getPosition, 130
 - getTarget, 131
 - getTilePos, 131
 - getType, 131
 - getUp, 131
 - isPlayerPov, 132
 - setFovy, 132
 - setPlayerId, 132
 - setPosition, 133
 - setTarget, 133
 - setTilePos, 133
 - setType, 133
 - setUp, 134
 - UserCamera, 129
- GuiGameDataException
 - Gui::Errors::GuiGameDataException, 43
- GUIUpdater
 - Gui::GUIUpdater, 45
- GuiUpdaterException
 - Gui::Errors::GuiUpdaterException, 54
- handleLeftClick
 - Gui::Event, 28
- HudGame
 - Gui::HudGame, 56
- HudPlayer
 - Gui::HudPlayer, 58
- HudTile
 - Gui::HudTile, 60
- IEvent
 - Gui::IEvent, 63
- increaseColorIndex
 - Gui::GUIUpdater, 46
- increaseRenderDistance
 - Gui::Event, 28
- increaseTimeUnit
 - Gui::Event, 28
- Inventory
 - Gui::Inventory, 71
- inventory
 - Gui::Player, 94
 - Gui::Tile, 128
- isCameraInPlayerPov
 - Gui::Render, 105
- isOpen
 - Gui::Render, 105
- isPlayerHit
 - Gui::Team, 121
- isPlayerPov
 - Gui::UserCamera, 132
- isTileHit
 - Gui::Tile, 127
- listen
 - Gui::Event, 28
 - Gui::IEvent, 63
- listenServer
 - Gui::ANetwork, 14
 - Gui::Engine, 23
 - Gui::INetwork, 68
 - Gui::Network, 80
- LoadModels
 - Gui::Render, 106
- moveDownCamera
 - Gui::Event, 29
- moveUpCamera
 - Gui::Event, 29
- Network
 - Gui::Network, 79
- NetworkException
 - Gui::Errors::NetworkException, 82
- parse
 - Gui::IServerParser, 77
 - Gui::ServerParser, 110
- parseCommand
 - Gui::ServerParser, 110
- ParseCommandLine
 - Gui::ParseCommandLine, 83
- ParseCommandLineException
 - Gui::Errors::ParseCommandLineException, 85
- parseFlags
 - Gui::ParseCommandLine, 84
- parseInt
 - Gui::ServerParser, 110
- parseListInt
 - Gui::ServerParser, 111
- parseMessage

- Gui::ServerParser, 111
- parseString
 - Gui::ServerParser, 112
- ParseType
 - Gui::ServerParser, 109
- Player
 - Gui::Player, 87
- readInfoServer
 - Gui::Network, 80
- removeEgg
 - Gui::Team, 121
- removePlayer
 - Gui::Team, 121
- removeResource
 - Gui::Inventory, 74
- removeServerEgg
 - Gui::GameData, 39
- Render
 - Gui::Render, 96
- restartAnimationTimeElapsed
 - Gui::Player, 91
- restartLastTick
 - Gui::GameData, 39
- restartLastTickMctCommand
 - Gui::GameData, 39
- rotateBoundingBoxByOrientation
 - Gui::Team, 123
- run
 - Gui::Engine, 23
- selectPlayer
 - Gui::Event, 29
- selectServer
 - Gui::Network, 80
- sendMessageServer
 - Gui::ANetwork, 15
 - Gui::INetwork, 68
 - Gui::Network, 80
- sendMessageUpdate
 - Gui::Engine, 23
- sendUpdateMapMessage
 - Gui::Engine, 23
- ServerParser
 - Gui::ServerParser, 109
- ServerParserException
 - Gui::Errors::ServerParserException, 114
- setBroadcast
 - Gui::Player, 91
- setCameraPlayerPov
 - Gui::Render, 106
- setCameraTile
 - Gui::Render, 106
- setCameraType
 - Gui::Render, 106
- setCurrentFrame
 - Gui::Player, 91
- setDeraumere
 - Gui::Inventory, 74
- setEggModelPath
 - Gui::Team, 123
- setFood
 - Gui::Inventory, 74
- setFovy
 - Gui::UserCamera, 132
- setGameData
 - Gui::IEvent, 63
- setHostName
 - Gui::ANetwork, 15
 - Gui::INetwork, 69
- setId
 - Gui::Egg, 20
 - Gui::Player, 92
- setIsDebug
 - Gui::Render, 107
- setIsEndGame
 - Gui::GameData, 39
- setLastError
 - Gui::GameData, 40
- setLevel
 - Gui::Player, 92
- setLinemate
 - Gui::Inventory, 75
- setMap
 - Gui::GameData, 40
- setMapSize
 - Gui::GameData, 40
- setMendiane
 - Gui::Inventory, 75
- setName
 - Gui::Team, 123
- setNbBCTCommandReceived
 - Gui::GameData, 41
- setOrientation
 - Gui::Player, 92
- setPhiras
 - Gui::Inventory, 75
- setPlayer
 - Gui::AHud, 11
 - Gui::IHud, 66
- setPlayerId
 - Gui::UserCamera, 132
- setPlayerModelPath
 - Gui::Team, 124
- setPlayerPov
 - Gui::Render, 107
- setPort
 - Gui::ANetwork, 15
 - Gui::INetwork, 69
- setPosition
 - Gui::Egg, 21
 - Gui::Player, 93
 - Gui::Tile, 127
 - Gui::UserCamera, 133
- setPosition3D
 - Gui::Player, 93
- setRender

- Gui::IEvent, 63
- setRenderDistance
 - Gui::Render, 107
- setRessources
 - Gui::Inventory, 75
- setServerTick
 - Gui::GameData, 41
- setSibur
 - Gui::Inventory, 76
- setState
 - Gui::Egg, 21
 - Gui::Player, 93
- setTarget
 - Gui::UserCamera, 133
- setTeam
 - Gui::Egg, 21
 - Gui::Player, 93
- setThystame
 - Gui::Inventory, 76
- setTile
 - Gui::AHud, 12
 - Gui::GameData, 41
 - Gui::IHud, 66
- setTilePos
 - Gui::UserCamera, 133
- setTimeUnit
 - Gui::Render, 108
- setTimeUnitFromServer
 - Gui::GameData, 41
- setType
 - Gui::UserCamera, 133
- setUp
 - Gui::UserCamera, 134
- switchDisplayDebug
 - Gui::Event, 29
- switchTileHudToGame
 - Gui::Event, 29
- Team
 - Gui::Team, 116
- Tile
 - Gui::Tile, 125
- TypeScene
 - Gui::IHud, 65
- update
 - Gui::AGUIUpdater, 9
 - Gui::GUIUpdater, 46
 - Gui::IGUIUpdater, 64
- updateCommandParameter
 - Gui::GUIUpdater, 46
- updateEggDeath
 - Gui::GUIUpdater, 46
- updateEggLaidByPlayer
 - Gui::GUIUpdater, 47
- updateEndOfGame
 - Gui::GUIUpdater, 47
- updateMap
 - Gui::Engine, 24
- updateMapContent
 - Gui::GUIUpdater, 47
- updateMapSize
 - Gui::GUIUpdater, 47
- updateMessageFromServer
 - Gui::GUIUpdater, 48
- updatePlayerBorn
 - Gui::GUIUpdater, 48
- updatePlayerBroadcast
 - Gui::GUIUpdater, 48
- updatePlayerDeath
 - Gui::GUIUpdater, 49
- updatePlayerEggLaying
 - Gui::GUIUpdater, 49
- updatePlayerEndIncantation
 - Gui::GUIUpdater, 49
- updatePlayerExpulsion
 - Gui::GUIUpdater, 49
- updatePlayerInventory
 - Gui::GUIUpdater, 50
- updatePlayerLevel
 - Gui::GUIUpdater, 50
- updatePlayerPosition
 - Gui::GUIUpdater, 50
- updatePlayerRessourceCollecting
 - Gui::GUIUpdater, 51
- updatePlayerRessourceDropping
 - Gui::GUIUpdater, 51
- updatePlayerStartIncantation
 - Gui::GUIUpdater, 51
- updateTeamMember
 - Gui::GUIUpdater, 51
- updateTeamNames
 - Gui::GUIUpdater, 52
- updateTimeUnitModification
 - Gui::GUIUpdater, 52
- updateTimeUnitRequest
 - Gui::GUIUpdater, 52
- updateUnknownMessage
 - Gui::GUIUpdater, 53
- UserCamera
 - Gui::UserCamera, 129
- what
 - Gui::Errors::AError, 8
 - Gui::Errors::IError, 62