

Zappy AI

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 Class Documentation	5
3.1 Action.Action Class Reference	5
3.1.1 Detailed Description	5
3.2 AI.AI Class Reference	6
3.2.1 Detailed Description	6
3.2.2 Constructor & Destructor Documentation	6
3.2.2.1 __init__()	6
3.2.3 Member Function Documentation	7
3.2.3.1 run()	7
3.3 API.API Class Reference	7
3.3.1 Detailed Description	7
3.3.2 Constructor & Destructor Documentation	8
3.3.2.1 __init__()	8
3.3.3 Member Function Documentation	8
3.3.3.1 close()	8
3.3.3.2 connect()	8
3.3.3.3 initConnection()	9
3.3.3.4 receiveData()	9
3.3.3.5 sendData()	9
3.4 APIException.APIException Class Reference	10
3.5 ArgsException.ArgsException Class Reference	10
3.5.1 Detailed Description	10
3.5.2 Constructor & Destructor Documentation	10
3.5.2.1 __init__()	10
3.6 IError.IError Class Reference	10
3.6.1 Detailed Description	11
3.6.2 Constructor & Destructor Documentation	11
3.6.2.1 __init__()	11
3.6.3 Member Function Documentation	11
3.6.3.1 __repr__()	12
3.6.3.2 __str__()	12
3.7 Inventory.Inventory Class Reference	12
3.7.1 Detailed Description	13
3.7.2 Constructor & Destructor Documentation	13
3.7.2.1 __init__()	13
3.7.3 Member Function Documentation	14
3.7.3.1 __eq__()	14

3.7.3.2 __str__()	14
3.7.3.3 addAnObject()	14
3.7.3.4 removeAnObject()	15
3.7.3.5 updateCaseContent()	15
3.7.3.6 updateInventory()	15
3.8 Item.Item Class Reference	15
3.8.1 Detailed Description	16
3.9 Player.Player Class Reference	16
3.9.1 Detailed Description	17
3.9.2 Constructor & Destructor Documentation	18
3.9.2.1 __init__()	18
3.9.3 Member Function Documentation	18
3.9.3.1 __str__()	18
3.9.3.2 broadcast()	19
3.9.3.3 chooseAction()	19
3.9.3.4 cmdInventory()	19
3.9.3.5 connectNbr()	19
3.9.3.6 eject()	19
3.9.3.7 fork()	20
3.9.3.8 handleElevation()	20
3.9.3.9 handleResponse()	20
3.9.3.10 hasSomethingHappened()	20
3.9.3.11 incantation()	21
3.9.3.12 look()	21
3.9.3.13 moveForward()	21
3.9.3.14 none()	21
3.9.3.15 set()	21
3.9.3.16 take()	22
3.9.3.17 turnLeft()	22
3.9.3.18 turnRight()	22
3.9.3.19 updateBroadcastReceived()	22
3.9.3.20 updateEjectionReceived()	23
3.9.3.21 updateInventory()	23
3.9.3.22 updateLevel()	23
3.9.3.23 updateVision()	23
3.10 PlayerException.PlayerDeathException Class Reference	24
3.10.1 Detailed Description	24
3.10.2 Constructor & Destructor Documentation	24
3.10.2.1 __init__()	24
3.11 PlayerException.PlayerException Class Reference	24
3.11.1 Detailed Description	25
3.11.2 Constructor & Destructor Documentation	25

3.11.2.1 <code>__init__()</code>	25
--	----

Index	27
--------------	-----------

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

AI.AI	6
API.API	7
Exception	
IError.IError	10
Inventory.Inventory	12
Player.Player	16
Enum	
Action.Action	5
Item.Item	15
IError	
APIException.APIException	10
ArgsException.ArgsException	10
PlayerException.PlayerException	24
PlayerException.PlayerDeathException	24

Chapter 2

Class Index

2.1 Class List

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

Action.Action	5
AI.AI	6
API.API	7
APIException.APIException	10
ArgsException.ArgsException	10
IError.IError	10
Inventory.Inventory	
EPITECH PROJECT, 2024 Zappy File description: Inventory	12
Item.Item	15
Player.Player	16
PlayerException.PlayerDeathException	24
PlayerException.PlayerException	24

Chapter 3

Class Documentation

3.1 Action.Action Class Reference

Inheritance diagram for Action.Action:

Collaboration diagram for Action.Action:

Static Public Attributes

- str **FORWARD** = "Forward"
- str **RIGHT** = "Right"
- str **LEFT** = "Left"
- str **LOOK** = "Look"
- str **INVENTORY** = "Inventory"
- str **BROADCAST** = "Broadcast"
- str **CONNECT_NBR** = "Connect_nbr"
- str **FORK** = "Fork"
- str **EJECT** = "Eject"
- str **TAKE** = "Take"
- str **SET** = "Set"
- str **INCANTATION** = "Incantation"
- str **NONE** = "None"

3.1.1 Detailed Description

Action class

A class to list the actions the player can do

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/ai/src/Enum/Action.py

3.2 AI.AI Class Reference

Public Member Functions

- def `__init__` (self, host, port, teamName, isLeader=False)
- def `run` (self)

Public Attributes

- `api`
- `player`
- `teamName`

3.2.1 Detailed Description

```
AI class
A class to handle the AI of the Zappy project

Attributes :
    api : API
        the API to communicate with the server
    player : Player
        the player
    teamName : str
        the name of the team

-----

Methods :
    __init__(host : str, port : int, teamName : str)
        Constructor of the AI class
    run()
        Run the AI
```

3.2.2 Constructor & Destructor Documentation

3.2.2.1 `__init__()`

```
def AI.AI.__init__ (
    self,
    host,
    port,
    teamName,
    isLeader = False )

Constructor of the AI class
Assign the API, the player and the team name

Parameters :
    host : str
        the host of the server
    port : int
        the port of the server
    teamName : str
        the name of the team
```

3.2.3 Member Function Documentation

3.2.3.1 run()

```
def AI.AI.run (
    self )
```

Run the AI (the main loop)
Connect to the server, initialize the connection and start the main loop

The main loop is an infinite loop that will select an action for the player and send it to the server and after that, it will receive the response from the server and handle it

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/ai/src/AI.py

3.3 API.API Class Reference

Public Member Functions

- def [__init__](#) (self, str host, int port)
- def [connect](#) (self)
- def [sendData](#) (self, str data, int timeout=None)
- def [receiveData](#) (self, int timeout=None)
- def [initConnection](#) (self, str teamName)
- def [close](#) (self)

3.3.1 Detailed Description

API class
A class to communicate with the server

Attributes :

- host : str
the host of the server
- port : int
the port of the server
- inputs : list
the list of inputs
- outputs : list
the list of outputs
- sock : socket
the socket to communicate with the server

Methods :

- sendData(data : str, timeout : int = None)
send data to the server
- receiveData(timeout : int = None)
receive data from the server
- connect(team_name : str)
connect to the server
- close()
close the connection

3.3.2 Constructor & Destructor Documentation

3.3.2.1 `__init__()`

```
def API.API.__init__ (
    self,
    str host,
    int port )
```

Constructor of the API class

Assign the host and the port of the server
Create the socket to communicate with the server
Connect to the server and add the socket to the inputs and outputs lists

Parameters :

- host : str
the host of the server
- port : int
the port of the server

3.3.3 Member Function Documentation

3.3.3.1 `close()`

```
def API.API.close (
    self )
```

Close the connection with the server

3.3.3.2 `connect()`

```
def API.API.connect (
    self )
```

Connect to the server
Add the socket to the inputs and outputs lists

3.3.3.3 initConnection()

```
def API.API.initConnection (
    self,
    str teamName )
```

Function to do the first exchange with the server

Send the team name to the server

Receive the client number and the map size from the server

Print the client number and the map size

Parameters :

```
    team_name : str
        the name of the team
```

Returns :

```
    client_num : int
        the client number
    x : int
        the x size of the map
    y : int
        the y size of the map
```

3.3.3.4 receiveData()

```
def API.API.receiveData (
    self,
    int timeout = None )
```

Receive data from the server

Parameters :

```
    timeout : int
        the timeout to wait for the server to send data
        (default is None which means no timeout)
```

3.3.3.5 sendData()

```
def API.API.sendData (
    self,
    str data,
    int timeout = None )
```

Send data to the server

Parameters :

```
    data : str
        the data to send
    timeout : int
        the timeout to wait for the server to be ready to receive data
        (default is None which means no timeout)
```

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/ai/src/Network/API.py

3.4 APIException.APIException Class Reference

Inheritance diagram for APIException.APIException:

3.5 ArgsException.ArgsException Class Reference

Inheritance diagram for ArgsException.ArgsException:

Collaboration diagram for ArgsException.ArgsException:

Public Member Functions

- `def __init__ (self, message)`

3.5.1 Detailed Description

ArgsException class

A class to handle exceptions that can occur in the Args
The ArgsException class inherits from the IError class

Attributes :
 message : str
 the message of the exception

3.5.2 Constructor & Destructor Documentation

3.5.2.1 `__init__()`

```
def ArgsException.ArgsException.__init__ (
    self,
    message )
```

Constructor of the ArgsException class

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

- `/home/tjerome-rocher/Desktop/Tek2/Zappy/ai/src/Errors/ArgsException.py`

3.6 IError.IError Class Reference

Inheritance diagram for IError.IError:

Collaboration diagram for IError.IError:

Public Member Functions

- def `__init__`(self, message)
- def `__str__`(self)
- def `__repr__`(self)

Public Attributes

- `message`

3.6.1 Detailed Description

IError class

A class to handle errors that can occur in the project

Attributes :

 message : str
 the message of the error

Methods :

 __str__()
 return the message of the error
 __repr__()
 return the message of the error

3.6.2 Constructor & Destructor Documentation

3.6.2.1 `__init__()`

```
def IError.IError.__init__ (
    self,
    message )
```

Constructor of the IError class

Assign the message of the error

Parameters :

 message : str
 the message of the error

3.6.3 Member Function Documentation

3.6.3.1 `__repr__()`

```
def IError.IError.__repr__ (
    self )
```

Return the message of the error

3.6.3.2 `__str__()`

```
def IError.IError.__str__ (
    self )
```

Return the message of the error

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

- `/home/tjerome-rocher/Desktop/Tek2/Zappy/ai/src/Errors/IError.py`

3.7 Inventory.Inventory Class Reference

EPITECH PROJECT, 2024 Zappy File description: Inventory.

Public Member Functions

- `def __init__` (self, food=10, linemate=0, deraumere=0, sibur=0, mendiane=0, phiras=0, thystame=0, player=0)
- `def __str__` (self)
- `def __eq__` (self, inventory)
- `def updateInventory` (self, str data)
- `def updateCaseContent` (self, list data)
- `def addAnObject` (self, str ressource)
- `def removeAnObject` (self, str ressource)

Public Attributes

- `food`
- `linemate`
- `deraumere`
- `sibur`
- `mendiane`
- `phiras`
- `thystame`
- `player`

3.7.1 Detailed Description

EPITECH PROJECT, 2024 Zappy File description: Inventory.

Inventory class

A class to handle the inventory of the player

Attributes :

```
    food : int
        the number of food
    linemate : int
        the number of linemate
    deraumere : int
        the number of deraumere
    sibur : int
        the number of sibur
    mendiane : int
        the number of mendiane
    phiras : int
        the number of phiras
    thystame : int
        the number of thystame
    player : int
        the number of players
```

Methods :

```
    __init__()
        Constructor of the Inventory class
    __str__()
        Print the inventory
    updateInventory(data)
        Update the inventory with the data from the inventory command
    updateCaseContent(data)
        Update the case content with the data from the vision command
    addAnObject(ressource)
        Add an object to the inventory
    removeAnObject(ressource)
        Remove an object from the inventory
```

3.7.2 Constructor & Destructor Documentation

3.7.2.1 __init__()

```
def Inventory.Inventory.__init__ (
    self,
    food = 10,
    linemate = 0,
    deraumere = 0,
    sibur = 0,
    mendiane = 0,
    phiras = 0,
    thystame = 0,
    player = 0 )
```

Constructor of the Inventory class

3.7.3 Member Function Documentation

3.7.3.1 `__eq__()`

```
def Inventory.Inventory.__eq__ (
    self,
    inventory )
```

Compare two inventories

Parameters :

- `inventory` : Inventory
the inventory to compare with

Returns :

- `bool`
True if the inventories are the same, False otherwise

3.7.3.2 `__str__()`

```
def Inventory.Inventory.__str__ (
    self )
```

Print the inventory

3.7.3.3 `addAnObject()`

```
def Inventory.Inventory.addAnObject (
    self,
    str ressource )
```

Add an object to the inventory

Parameters :

- `ressource` : str
the ressource to add

3.7.3.4 removeAnObject()

```
def Inventory.Inventory.removeAnObject (
    self,
    str ressource )
```

Remove an object from the inventory

Parameters :

- ressource : str
the ressource to remove

3.7.3.5 updateCaseContent()

```
def Inventory.Inventory.updateCaseContent (
    self,
    list data )
```

Update the case content with the data from the vision command

Parameters :

- data : list
the data from the vision command

3.7.3.6 updateInventory()

```
def Inventory.Inventory.updateInventory (
    self,
    str data )
```

Update the inventory with the data from the inventory command

Parameters :

- data : str
the data from the inventory command

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/ai/src/Player/Inventory.py

3.8 Item.Item Class Reference

Inheritance diagram for Item.Item:

Collaboration diagram for Item.Item:

Static Public Attributes

- str **FOOD** = "food"
- str **LINEMATE** = "linemate"
- str **DERAUMERE** = "deraumere"
- str **SIBUR** = "sibur"
- str **MENDIANE** = "mendiane"
- str **PHIRAS** = "phiras"
- str **THYSTAME** = "thystame"

3.8.1 Detailed Description

```
Item class
A class to list the items in the game
```

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/ai/src/Enum/Item.py

3.9 Player.Player Class Reference

Public Member Functions

- def `__init__` (self, bool isLeader)
- def `__str__` (self)
- def `moveForward` (self)
- def `turnRight` (self)
- def `turnLeft` (self)
- def `look` (self)
- def `cmdInventory` (self)
- def `broadcast` (self, str message="Hello")
- def `connectNbr` (self)
- def `fork` (self)
- def `eject` (self)
- def `take` (self, str resource="food")
- def `set` (self, str resource="food")
- def `incantation` (self)
- def `none` (self)
- def `updateVision` (self, str vision)
- def `updateInventory` (self, str inventory)
- def `updateBroadcastReceived` (self, str message)
- def `updateEjectionReceived` (self, str message)
- def `updateLevel` (self, int level)
- def `handleElevation` (self, str response)
- def `hasSomethingHappened` (self, str response)
- def `handleResponse` (self, str response)
- def `chooseAction` (self)

Public Attributes

- **inventory**
- **level**
- **currentAction**
- **currentCommand**
- **callback**
- **vision**
- **broadcastReceived**
- **ejectionReceived**
- **isLeader**
- **unusedSlots**
- **currentlyElevating**

3.9.1 Detailed Description

Player class

A class to handle the player

Attributes :

```
    inventory : Inventory
        the inventory of the player
    level : int
        the level of the player
    currentAction : Action
        the current action of the player
    currentCommand : str
        the current command of the player
    callback : function
        the callback function
    vision : list
        the vision of the player
    broadcastReceived : list
        the broadcast received by the player
    ejectionReceived : list
        the ejection received by the player
    isLeader : bool
        if the player is the leader
    unusedSlots : int
        the unused slots
    currentlyElevating : bool
        if the player is currently elevating
```

Methods :

```
    __init__(isLeader : bool)
        Constructor of the Player class
    __str__()
        Print the player
    moveForward()
        Move the player forward
    turnRight()
        Turn the player right
    turnLeft()
        Turn the player left
    look()
        Look around the player
    cmdInventory()
        Get the inventory of the player
    broadcast(message : str = "Hello")
        Broadcast a message
    connectNbr()
        Connect to the number of players
    fork()
        Fork the player
```

```

eject()
    Eject the player
take(resource : str = "food")
    Take a resource
set(resource : str = "food")
    Set a resource
incantation()
    Start the incantation
none()
    Do nothing
updateVision(vision : str)
    Update the vision of the player
updateInventory(inventory : str)
    Update the inventory of the player
updateBroadcastReceived(message : str)
    Update the broadcast received by the player
updateEjectionReceived(message : str)
    Update the ejection received by the player
updateLevel(level : int)
    Update the level of the player
handleElevation(response : str)
    Handle the elevation
hasSomethingHappened(response : str)
    Check if something happened
handleResponse(response : str)
    Handle the response
chooseAction()
    Choose the action of the player

```

3.9.2 Constructor & Destructor Documentation

3.9.2.1 __init__()

```

def Player.Player.__init__ (
    self,
    bool isLeader )

```

Constructor of the Player class

Parameters :

- isLeader : bool
if the player is the leader

3.9.3 Member Function Documentation

3.9.3.1 __str__()

```

def Player.Player.__str__ (
    self )

```

Print the player

3.9.3.2 broadcast()

```
def Player.Player.broadcast (
    self,
    str message = "Hello" )
```

Set the current action to broadcast

Parameters :

- message : str
the message to broadcast

3.9.3.3 chooseAction()

```
def Player.Player.chooseAction (
    self )
```

Choose the action of the player

TODO: Implement the logic to choose the action of the player

3.9.3.4 cmdInventory()

```
def Player.Player.cmdInventory (
    self )
```

Set the current action to inventory

3.9.3.5 connectNbr()

```
def Player.Player.connectNbr (
    self )
```

Set the current action to connect_nbr

3.9.3.6 eject()

```
def Player.Player.eject (
    self )
```

Set the current action to eject

3.9.3.7 fork()

```
def Player.Player.fork (
    self )
```

Set the current action to fork

3.9.3.8 handleElevation()

```
def Player.Player.handleElevation (
    self,
    str response )
```

Handle the response of the elevation command

Parameters :
 response : str
 the response from the server

3.9.3.9 handleResponse()

```
def Player.Player.handleResponse (
    self,
    str response )
```

Handle the response from the server

Parameters :
 response : str
 the response from the server

3.9.3.10 hasSomethingHappened()

```
def Player.Player.hasSomethingHappened (
    self,
    str response )
```

Check if something happened to the player
Look if the player is dead, if he received a message or if he was ejected

Parameters :
 response : str
 the response from the server

3.9.3.11 incantation()

```
def Player.Player.incantation (
    self )
```

Set the current action to incantation

3.9.3.12 look()

```
def Player.Player.look (
    self )
```

Set the current action to look

3.9.3.13 moveForward()

```
def Player.Player.moveForward (
    self )
```

Set the current action to forward

3.9.3.14 none()

```
def Player.Player.none (
    self )
```

Set the current action to none

3.9.3.15 set()

```
def Player.Player.set (
    self,
    str resource = "food" )
```

Set the current action to set

Parameters :
resource : str
the resource to set

3.9.3.16 take()

```
def Player.Player.take (
    self,
    str resource = "food" )
```

Set the current action to take

Parameters :
 resource : str
 the resource to take

3.9.3.17 turnLeft()

```
def Player.Player.turnLeft (
    self )
```

Set the current action to left

3.9.3.18 turnRight()

```
def Player.Player.turnRight (
    self )
```

Set the current action to right

3.9.3.19 updateBroadcastReceived()

```
def Player.Player.updateBroadcastReceived (
    self,
    str message )
```

Update the broadcast received by the player

Parameters :
 message : str
 the message from the server

3.9.3.20 updateEjectionReceived()

```
def Player.Player.updateEjectionReceived (
    self,
    str message )
```

Update the ejection received by the player

Parameters :

- message : str
the message from the server

3.9.3.21 updateInventory()

```
def Player.Player.updateInventory (
    self,
    str inventory )
```

Update the inventory of the player with the data from the inventory command

Parameters :

- inventory : str
the inventory from the server

3.9.3.22 updateLevel()

```
def Player.Player.updateLevel (
    self,
    int level )
```

Update the level of the player

Parameters :

- level : int
the level of the player

3.9.3.23 updateVision()

```
def Player.Player.updateVision (
    self,
    str vision )
```

Update the vision of the player with the data from the look command

Parameters :

- vision : str
the vision from the server

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/ai/src/Player/Player.py

3.10 PlayerException.PlayerDeathException Class Reference

Inheritance diagram for PlayerException.PlayerDeathException:

Collaboration diagram for PlayerException.PlayerDeathException:

Public Member Functions

- `def __init__ (self, message)`
- `def __init__ (self, message)`

3.10.1 Detailed Description

PlayerDeathException class

A class to handle the death of the player
The PlayerDeathException class inherits from the PlayerException class

Attributes :
 message : str
 the message of the exception

3.10.2 Constructor & Destructor Documentation

3.10.2.1 `__init__()`

```
def PlayerException.PlayerDeathException.__init__ (
    self,
    message )
```

Constructor of the PlayerDeathException class

Reimplemented from [PlayerException.PlayerException](#).

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

- `/home/tjerome-rocher/Desktop/Tek2/Zappy/ai/src/Player/PlayerException.py`

3.11 PlayerException.PlayerException Class Reference

Inheritance diagram for PlayerException.PlayerException:

Collaboration diagram for PlayerException.PlayerException:

Public Member Functions

- `def __init__(self, message)`

3.11.1 Detailed Description

PlayerException class

A class to handle exceptions that can occur in the Player
The PlayerException class inherits from the IError class

Attributes :
 message : str
 the message of the exception

3.11.2 Constructor & Destructor Documentation

3.11.2.1 `__init__()`

```
def PlayerException.PlayerException.__init__ (
    self,
    message )
```

Constructor of the PlayerException class

Reimplemented in [PlayerException.PlayerDeathException](#).

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

- `/home/tjerome-rocher/Desktop/Tek2/Zappy/ai/src/Player/PlayerException.py`

Index

- `__eq__`
 - Inventory.Inventory, [14](#)
 - `__init__`
 - AI.AI, [6](#)
 - API.API, [8](#)
 - ArgsException.ArgsException, [10](#)
 - LError.LError, [11](#)
 - Inventory.Inventory, [13](#)
 - Player.Player, [18](#)
 - PlayerException.PlayerDeathException, [24](#)
 - PlayerException.PlayerException, [25](#)
 - `__repr__`
 - LError.LError, [11](#)
 - `__str__`
 - LError.LError, [12](#)
 - Inventory.Inventory, [14](#)
 - Player.Player, [18](#)
- Action.Action, [5](#)
- addAnObject
 - Inventory.Inventory, [14](#)
- AI.AI, [6](#)
 - `__init__`, [6](#)
 - run, [7](#)
- API.API, [7](#)
 - `__init__`, [8](#)
 - close, [8](#)
 - connect, [8](#)
 - initConnection, [8](#)
 - receiveData, [9](#)
 - sendData, [9](#)
- APIException.APIException, [10](#)
- ArgsException.ArgsException, [10](#)
 - `__init__`, [10](#)
- broadcast
 - Player.Player, [18](#)
- chooseAction
 - Player.Player, [19](#)
- close
 - API.API, [8](#)
- cmdInventory
 - Player.Player, [19](#)
- connect
 - API.API, [8](#)
- connectNbr
 - Player.Player, [19](#)
- eject
 - Player.Player, [19](#)
- fork
 - Player.Player, [19](#)
- handleElevation
 - Player.Player, [20](#)
- handleResponse
 - Player.Player, [20](#)
- hasSomethingHappened
 - Player.Player, [20](#)
- LError.LError, [10](#)
 - `__init__`, [11](#)
 - `__repr__`, [11](#)
 - `__str__`, [12](#)
- incantation
 - Player.Player, [20](#)
- initConnection
 - API.API, [8](#)
- Inventory.Inventory, [12](#)
 - `__eq__`, [14](#)
 - `__init__`, [13](#)
 - `__str__`, [14](#)
 - addAnObject, [14](#)
 - removeAnObject, [14](#)
 - updateCaseContent, [15](#)
 - updateInventory, [15](#)
- Item.Item, [15](#)
- look
 - Player.Player, [21](#)
- moveForward
 - Player.Player, [21](#)
- none
 - Player.Player, [21](#)
- Player.Player, [16](#)
 - `__init__`, [18](#)
 - `__str__`, [18](#)
 - broadcast, [18](#)
 - chooseAction, [19](#)
 - cmdInventory, [19](#)
 - connectNbr, [19](#)
 - eject, [19](#)
 - fork, [19](#)
 - handleElevation, [20](#)
 - handleResponse, [20](#)
 - hasSomethingHappened, [20](#)

- incantation, [20](#)
- look, [21](#)
- moveForward, [21](#)
- none, [21](#)
- set, [21](#)
- take, [21](#)
- turnLeft, [22](#)
- turnRight, [22](#)
- updateBroadcastReceived, [22](#)
- updateEjectionReceived, [22](#)
- updateInventory, [23](#)
- updateLevel, [23](#)
- updateVision, [23](#)
- PlayerException.PlayerDeathException, [24](#)
 - __init__, [24](#)
- PlayerException.PlayerException, [24](#)
 - __init__, [25](#)
- receiveData
 - API.API, [9](#)
- removeAnObject
 - Inventory.Inventory, [14](#)
- run
 - AI.AI, [7](#)
- sendData
 - API.API, [9](#)
- set
 - Player.Player, [21](#)
- take
 - Player.Player, [21](#)
- turnLeft
 - Player.Player, [22](#)
- turnRight
 - Player.Player, [22](#)
- updateBroadcastReceived
 - Player.Player, [22](#)
- updateCaseContent
 - Inventory.Inventory, [15](#)
- updateEjectionReceived
 - Player.Player, [22](#)
- updateInventory
 - Inventory.Inventory, [15](#)
 - Player.Player, [23](#)
- updateLevel
 - Player.Player, [23](#)
- updateVision
 - Player.Player, [23](#)