

## Detritus 2 - Systems

Generated by Doxygen 1.9.4



<b>1 requirements_syntax</b>	<b>1</b>
<b>2 Namespace Index</b>	<b>3</b>
2.1 Namespace List . . . . .	3
<b>3 Hierarchical Index</b>	<b>5</b>
3.1 Class Hierarchy . . . . .	5
<b>4 Class Index</b>	<b>7</b>
4.1 Class List . . . . .	7
<b>5 File Index</b>	<b>9</b>
5.1 File List . . . . .	9
<b>6 Namespace Documentation</b>	<b>11</b>
6.1 systems Namespace Reference . . . . .	11
6.2 systems.database Namespace Reference . . . . .	11
6.3 systems.game Namespace Reference . . . . .	11
6.4 systems.gameobject Namespace Reference . . . . .	11
6.5 systems.inventory Namespace Reference . . . . .	12
6.6 systems.item Namespace Reference . . . . .	12
6.7 systems.person Namespace Reference . . . . .	12
6.8 systems.player Namespace Reference . . . . .	12
6.9 systems.quests Namespace Reference . . . . .	12
6.10 systems.recipe Namespace Reference . . . . .	12
<b>7 Class Documentation</b>	<b>13</b>
7.1 systems.database.Database Class Reference . . . . .	13
7.1.1 Detailed Description . . . . .	14
7.1.2 Constructor & Destructor Documentation . . . . .	14
7.1.2.1 __init__() . . . . .	14
7.1.3 Member Function Documentation . . . . .	14
7.1.3.1 get_item() . . . . .	14
7.1.3.2 get_quest() . . . . .	15
7.1.3.3 get_recipe() . . . . .	15
7.1.3.4 load_items_from_file() . . . . .	15
7.1.3.5 load_recipes_from_file() . . . . .	15
7.1.4 Member Data Documentation . . . . .	15
7.1.4.1 items . . . . .	15
7.1.4.2 recipes . . . . .	16
7.1.4.3 recipes_fp . . . . .	16
7.2 systems.game.Game Class Reference . . . . .	16
7.2.1 Detailed Description . . . . .	17
7.2.2 Constructor & Destructor Documentation . . . . .	17

7.2.2.1 <code>__init__()</code>	17
7.2.3 Member Function Documentation	17
7.2.3.1 <code>game_event()</code>	17
7.2.3.2 <code>setup()</code>	18
7.2.3.3 <code>start_quest()</code>	18
7.2.3.4 <code>update()</code>	19
7.2.3.5 <code>update_requests()</code>	19
7.2.4 Member Data Documentation	20
7.2.4.1 <code>db</code>	20
7.2.4.2 <code>location</code>	20
7.2.4.3 <code>people</code>	20
7.2.4.4 <code>player</code>	20
7.3 <code>systems.gameobject.GameObject</code> Class Reference	21
7.3.1 Detailed Description	21
7.3.2 Constructor & Destructor Documentation	22
7.3.2.1 <code>__init__()</code>	22
7.3.3 Member Function Documentation	22
7.3.3.1 <code>__eq__()</code>	22
7.3.3.2 <code>__hash__()</code>	22
7.3.3.3 <code>__ne__()</code>	22
7.3.3.4 <code>__repr__()</code>	23
7.3.4 Member Data Documentation	23
7.3.4.1 <code>description</code>	23
7.3.4.2 <code>id</code>	23
7.3.4.3 <code>name</code>	23
7.4 <code>systems.inventory.Inventory</code> Class Reference	23
7.4.1 Detailed Description	24
7.4.2 Constructor & Destructor Documentation	24
7.4.2.1 <code>__init__()</code>	24
7.4.3 Member Function Documentation	24
7.4.3.1 <code>__contains__()</code>	24
7.4.3.2 <code>__getitem__()</code>	24
7.4.3.3 <code>add_item()</code>	25
7.4.3.4 <code>get_current_item()</code>	25
7.4.3.5 <code>get_item_count()</code>	25
7.4.3.6 <code>get_items()</code>	25
7.4.3.7 <code>remove_item()</code>	26
7.4.3.8 <code>set_current_item()</code>	26
7.4.4 Member Data Documentation	26
7.4.4.1 <code>current_item</code>	26
7.4.4.2 <code>items</code>	26
7.5 <code>systems.item.InventoryItem</code> Class Reference	27

7.5.1 Detailed Description . . . . .	28
7.5.2 Constructor & Destructor Documentation . . . . .	28
7.5.2.1 <code>__init__()</code> . . . . .	29
7.5.3 Member Function Documentation . . . . .	29
7.5.3.1 <code>__iadd__()</code> . . . . .	29
7.5.3.2 <code>__isub__()</code> . . . . .	30
7.5.3.3 <code>from_dict()</code> . . . . .	30
7.5.3.4 <code>from_item()</code> . . . . .	30
7.5.3.5 <code>set_count()</code> . . . . .	31
7.5.4 Member Data Documentation . . . . .	31
7.5.4.1 <code>count</code> . . . . .	31
7.6 <code>systems.item.Item</code> Class Reference . . . . .	31
7.6.1 Detailed Description . . . . .	32
7.6.2 Constructor & Destructor Documentation . . . . .	32
7.6.2.1 <code>__init__()</code> . . . . .	33
7.6.3 Member Function Documentation . . . . .	33
7.6.3.1 <code>__str__()</code> . . . . .	33
7.6.3.2 <code>from_dict()</code> . . . . .	34
7.6.4 Member Data Documentation . . . . .	34
7.6.4.1 <code>image_path</code> . . . . .	34
7.7 <code>systems.person.Person</code> Class Reference . . . . .	34
7.7.1 Detailed Description . . . . .	35
7.7.2 Constructor & Destructor Documentation . . . . .	35
7.7.2.1 <code>__init__()</code> . . . . .	36
7.7.3 Member Function Documentation . . . . .	36
7.7.3.1 <code>__str__()</code> . . . . .	36
7.7.3.2 <code>decrease_affinity()</code> . . . . .	37
7.7.3.3 <code>get_affinity()</code> . . . . .	37
7.7.3.4 <code>increase_affinity()</code> . . . . .	37
7.7.3.5 <code>set_affinity()</code> . . . . .	37
7.7.4 Member Data Documentation . . . . .	37
7.7.4.1 <code>affinity</code> . . . . .	37
7.7.4.2 <code>base_affinity</code> . . . . .	38
7.7.4.3 <code>location</code> . . . . .	38
7.8 <code>systems.item.PlaceholderInventoryItem</code> Class Reference . . . . .	38
7.8.1 Detailed Description . . . . .	39
7.8.2 Constructor & Destructor Documentation . . . . .	39
7.8.2.1 <code>__init__()</code> . . . . .	40
7.9 <code>systems.player.Player</code> Class Reference . . . . .	41
7.9.1 Detailed Description . . . . .	41
7.9.2 Constructor & Destructor Documentation . . . . .	42
7.9.2.1 <code>__init__()</code> . . . . .	42

7.9.3 Member Function Documentation	42
7.9.3.1 craft()	42
7.9.3.2 get_craftable()	42
7.9.3.3 get_uncraftable()	42
7.9.3.4 start_quest()	43
7.9.4 Member Data Documentation	43
7.9.4.1 inventory	43
7.9.4.2 questlog	43
7.9.4.3 recipebook	43
7.10 systems.quests.Quest Class Reference	44
7.10.1 Detailed Description	45
7.10.2 Constructor & Destructor Documentation	45
7.10.2.1 __init__()	45
7.10.3 Member Function Documentation	46
7.10.3.1 from_dict()	46
7.10.3.2 get_stage()	46
7.10.3.3 next_stage()	46
7.10.3.4 start()	47
7.10.4 Member Data Documentation	47
7.10.4.1 complete	47
7.10.4.2 current_stage	47
7.10.4.3 final_stage	47
7.10.4.4 first_stage	47
7.10.4.5 stages	48
7.10.4.6 started	48
7.11 systems.quests.QuestLog Class Reference	48
7.11.1 Detailed Description	49
7.11.2 Constructor & Destructor Documentation	49
7.11.2.1 __init__()	49
7.11.3 Member Function Documentation	49
7.11.3.1 __getitem__()	49
7.11.3.2 add_quest()	50
7.11.3.3 get_active()	50
7.11.3.4 get_quest()	50
7.11.4 Member Data Documentation	50
7.11.4.1 quests	50
7.12 systems.quests.QuestStage Class Reference	51
7.12.1 Detailed Description	52
7.12.2 Constructor & Destructor Documentation	52
7.12.2.1 __init__()	52
7.12.3 Member Function Documentation	53
7.12.3.1 __str__()	53

7.12.3.2 from_dict()	53
7.12.4 Member Data Documentation	53
7.12.4.1 complete	53
7.12.4.2 requirements	53
7.12.4.3 substages	53
7.13 systems.recipe.Recipe Class Reference	54
7.13.1 Detailed Description	55
7.13.2 Constructor & Destructor Documentation	55
7.13.2.1 __init__()	55
7.13.3 Member Function Documentation	56
7.13.3.1 __eq__()	56
7.13.3.2 __hash__()	56
7.13.3.3 from_dict()	56
7.13.3.4 is_craftable()	56
7.13.4 Member Data Documentation	56
7.13.4.1 item	57
7.13.4.2 requirements	57
7.14 systems.recipe.RecipeBook Class Reference	57
7.14.1 Detailed Description	58
7.14.2 Constructor & Destructor Documentation	58
7.14.2.1 __init__()	58
7.14.3 Member Function Documentation	58
7.14.3.1 add_recipe() [1/2]	58
7.14.3.2 add_recipe() [2/2]	59
7.14.3.3 craft()	59
7.14.3.4 get_craftable()	59
7.14.3.5 get_uncraftable()	59
7.14.3.6 remove_recipe()	60
7.14.4 Member Data Documentation	60
7.14.4.1 current_recipe	60
7.14.4.2 recipes	60
<b>8 File Documentation</b>	<b>61</b>
8.1 systems/__init__.py File Reference	61
8.2 __init__.py	61
8.3 systems/database.py File Reference	61
8.4 database.py	61
8.5 systems/game.py File Reference	62
8.6 game.py	62
8.7 systems/gameobject.py File Reference	63
8.8 gameobject.py	63
8.9 systems/inventory.py File Reference	64

8.10 inventory.py . . . . .	64
8.11 systems/item.py File Reference . . . . .	65
8.12 item.py . . . . .	65
8.13 systems/person.py File Reference . . . . .	66
8.14 person.py . . . . .	67
8.15 systems/player.py File Reference . . . . .	67
8.16 player.py . . . . .	67
8.17 systems/quests.py File Reference . . . . .	68
8.18 quests.py . . . . .	68
8.19 systems/recipe.py File Reference . . . . .	69
8.20 recipe.py . . . . .	69
8.21 systems/requirements_syntax.md File Reference . . . . .	70
<b>Index</b>	<b>71</b>



## Chapter 1

# requirements\_syntax



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">systems</a>	11
<a href="#">systems.database</a>	11
<a href="#">systems.game</a>	11
<a href="#">systems.gameobject</a>	11
<a href="#">systems.inventory</a>	12
<a href="#">systems.item</a>	12
<a href="#">systems.person</a>	12
<a href="#">systems.player</a>	12
<a href="#">systems.quests</a>	12
<a href="#">systems.recipe</a>	12



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

systems.inventory.Inventory . . . . .	23
object	
systems.database.Database . . . . .	13
systems.game.Game . . . . .	16
systems.gameobject.GameObject . . . . .	21
systems.item.Item . . . . .	31
systems.item.InventoryItem . . . . .	27
systems.item.PlaceholderInventoryItem . . . . .	38
systems.person.Person . . . . .	34
systems.quests.Quest . . . . .	44
systems.quests.QuestStage . . . . .	51
systems.recipe.Recipe . . . . .	54
systems.player.Player . . . . .	41
systems.quests.QuestLog . . . . .	48
systems.recipe.RecipeBook . . . . .	57



## Chapter 4

# Class Index

### 4.1 Class List

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

<a href="#">systems.database.Database</a>	13
<a href="#">systems.game.Game</a>	16
<a href="#">systems.gameobject.GameObject</a>	21
<a href="#">systems.inventory.Inventory</a>	23
<a href="#">systems.item.InventoryItem</a>	27
<a href="#">systems.item.Item</a>	31
<a href="#">systems.person.Person</a>	34
<a href="#">systems.item.PlaceholderInventoryItem</a>	38
<a href="#">systems.player.Player</a>	41
<a href="#">systems.quests.Quest</a>	44
<a href="#">systems.quests.QuestLog</a>	48
<a href="#">systems.quests.QuestStage</a>	51
<a href="#">systems.recipe.Recipe</a>	54
<a href="#">systems.recipe.RecipeBook</a>	57





## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

systems/ <a href="#">__init__.py</a>	61
systems/ <a href="#">database.py</a>	61
systems/ <a href="#">game.py</a>	62
systems/ <a href="#">gameobject.py</a>	63
systems/ <a href="#">inventory.py</a>	64
systems/ <a href="#">item.py</a>	65
systems/ <a href="#">person.py</a>	66
systems/ <a href="#">player.py</a>	67
systems/ <a href="#">quests.py</a>	68
systems/ <a href="#">recipe.py</a>	69



## Chapter 6

# Namespace Documentation

### 6.1 systems Namespace Reference

#### Namespaces

- namespace [database](#)
- namespace [game](#)
- namespace [gameobject](#)
- namespace [inventory](#)
- namespace [item](#)
- namespace [person](#)
- namespace [player](#)
- namespace [quests](#)
- namespace [recipe](#)

### 6.2 systems.database Namespace Reference

#### Classes

- class [Database](#)

### 6.3 systems.game Namespace Reference

#### Classes

- class [Game](#)

### 6.4 systems.gameobject Namespace Reference

#### Classes

- class [GameObject](#)

## 6.5 systems.inventory Namespace Reference

### Classes

- class [Inventory](#)

## 6.6 systems.item Namespace Reference

### Classes

- class [InventoryItem](#)
- class [Item](#)
- class [PlaceholderInventoryItem](#)

## 6.7 systems.person Namespace Reference

### Classes

- class [Person](#)

## 6.8 systems.player Namespace Reference

### Classes

- class [Player](#)

## 6.9 systems.quests Namespace Reference

### Classes

- class [Quest](#)
- class [QuestLog](#)
- class [QuestStage](#)

## 6.10 systems.recipe Namespace Reference

### Classes

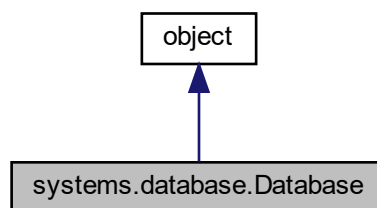
- class [Recipe](#)
- class [RecipeBook](#)

## Chapter 7

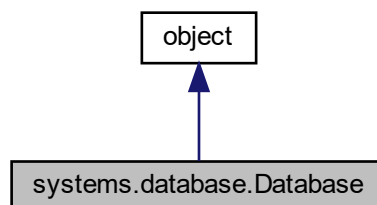
# Class Documentation

### 7.1 `systems.database.Database` Class Reference

Inheritance diagram for `systems.database.Database`:



Collaboration diagram for `systems.database.Database`:



## Public Member Functions

- def `__init__` (self, directory='data', item\_fp='items.json', recipe\_fp='recipes.json')
- def `verify` (self)
- def `load_items_from_file` (self, file)
- def `load_recipes_from_file` (self, file)
- def `get_item` (self, item\_id="")
- def `get_recipe` (self, recipe\_id="")
- def `get_quest` (self, quest\_id)
- def `set_items` (self, items)
- def `__getitem__` (self, key)

## Public Attributes

- `recipes_fp`
- `items`
- `recipes`
- `quests`
- `people`
- `locations`

### 7.1.1 Detailed Description

Definition at line 5 of file [database.py](#).

### 7.1.2 Constructor & Destructor Documentation

#### 7.1.2.1 `__init__()`

```
def systems.database.Database.__init__ (
    self,
    directory = 'data',
    item_fp = 'items.json',
    recipe_fp = 'recipes.json' )
```

Definition at line 6 of file [database.py](#).

### 7.1.3 Member Function Documentation

#### 7.1.3.1 `__getitem__()`

```
def systems.database.Database.__getitem__ (
    self,
    key )
```

Definition at line 57 of file [database.py](#).

#### 7.1.3.2 `get_item()`

```
def systems.database.Database.get_item (
    self,
    item_id = '' )
```

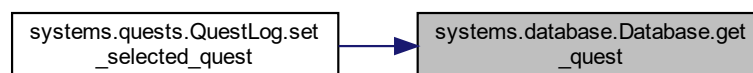
Definition at line 43 of file [database.py](#).

#### 7.1.3.3 `get_quest()`

```
def systems.database.Database.get_quest (
    self,
    quest_id )
```

Definition at line 49 of file [database.py](#).

Here is the caller graph for this function:



#### 7.1.3.4 `get_recipe()`

```
def systems.database.Database.get_recipe (
    self,
    recipe_id = '' )
```

Definition at line 46 of file [database.py](#).

#### 7.1.3.5 load\_items\_from\_file()

```
def systems.database.Database.load_items_from_file (
    self,
    file )
```

Definition at line 29 of file [database.py](#).

#### 7.1.3.6 load\_recipes\_from\_file()

```
def systems.database.Database.load_recipes_from_file (
    self,
    file )
```

Definition at line 36 of file [database.py](#).

#### 7.1.3.7 set\_items()

```
def systems.database.Database.set_items (
    self,
    items )
```

Definition at line 53 of file [database.py](#).

#### 7.1.3.8 verify()

```
def systems.database.Database.verify (
    self )
```

Definition at line 22 of file [database.py](#).

### 7.1.4 Member Data Documentation

#### 7.1.4.1 items

```
systems.database.Database.items
```

Definition at line 14 of file [database.py](#).



#### 7.1.4.2 locations

`systems.database.Database.locations`

Definition at line 18 of file [database.py](#).

#### 7.1.4.3 people

`systems.database.Database.people`

Definition at line 17 of file [database.py](#).

#### 7.1.4.4 quests

`systems.database.Database.quests`

Definition at line 16 of file [database.py](#).

#### 7.1.4.5 recipes

`systems.database.Database.recipes`

Definition at line 15 of file [database.py](#).

#### 7.1.4.6 recipes\_fp

`systems.database.Database.recipes_fp`

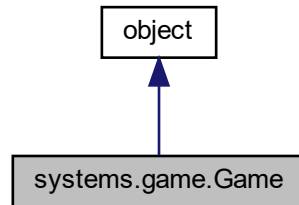
Definition at line 12 of file [database.py](#).

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

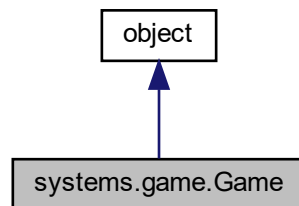
- [systems/database.py](#)

## 7.2 systems.game.Game Class Reference

Inheritance diagram for systems.game.Game:



Collaboration diagram for systems.game.Game:



### Public Member Functions

- def `__init__` (self)
- def `setup` (self, `people`=None, `locations`=None, `from_db`=False, `fps`={}, `items`=[], `recipes`=[], `quests`=[])
- def `start_quest` (self, `quest_id`=None, `quest_object`=None)
- def `update` (self)
- def `update_quests` (self)
- def `set_selected_quest` (self, `quest_id`)
- def `get_selected_quest` (self)
- def `add_item` (self, `item`, `count`=1)
- def `remove_item` (self, `item`, `count`=1)
- def `craft` (self, `recipe`)
- def `get_craftable` (self)
- def `get_uncraftable` (self)

## Public Attributes

- [people](#)
- [db](#)
- [player](#)
- [location](#)

### 7.2.1 Detailed Description

Definition at line 5 of file [game.py](#).

### 7.2.2 Constructor & Destructor Documentation

#### 7.2.2.1 `__init__()`

```
def systems.game.Game.__init__ (  
    self )
```

Definition at line 6 of file [game.py](#).

### 7.2.3 Member Function Documentation

#### 7.2.3.1 `add_item()`

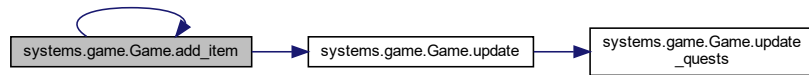
```
def systems.game.Game.add_item (  
    self,  
    item,  
    count = 1 )
```

Adds an item to the Player's inventory.

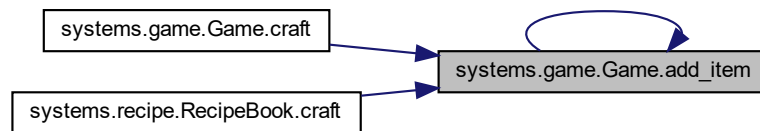
```
Keyword Arguments:  
item -- id or Item object to add  
count -- amount of the item to add (default: 1)
```

Definition at line 141 of file [game.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 7.2.3.2 `craft()`

```
def systems.game.Game.craft (  
    self,  
    recipe )
```

Crafts Recipe.

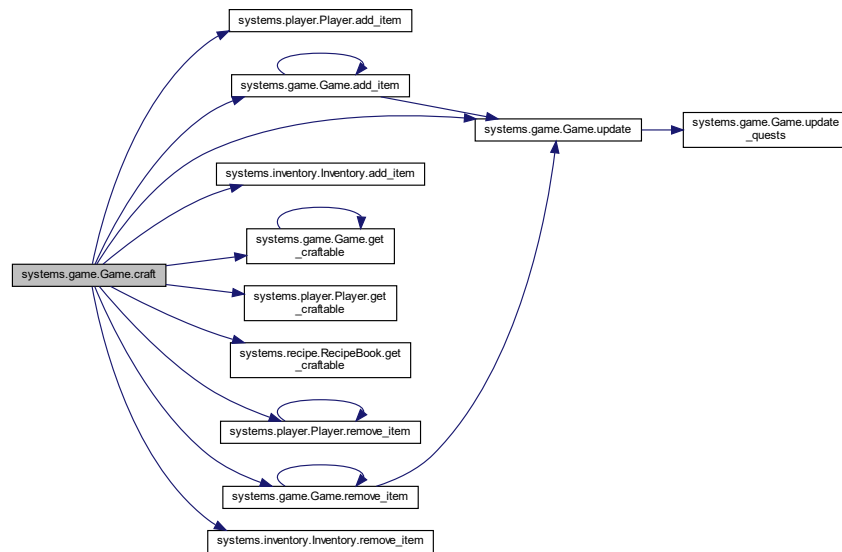
Checks player's inventory for required Item(s), removes them from the player's inventory, adds the cra

Arguments:

recipe -- id or Recipe object to craft

Definition at line 162 of file [game.py](#).

Here is the call graph for this function:



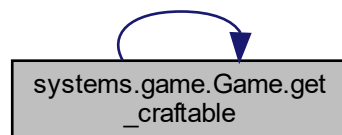
### 7.2.3.3 get\_craftable()

```
def systems.game.Game.get_craftable (
    self )
```

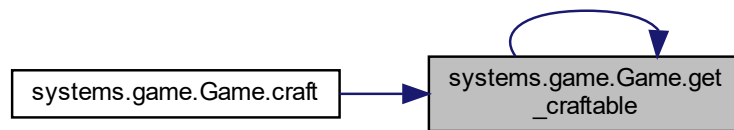
Returns a list of craftable recipes.

Definition at line 189 of file [game.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



#### 7.2.3.4 `get_selected_quest()`

```
def systems.game.Game.get_selected_quest (
    self )
```

Returns the currently selected Quest

Definition at line 137 of file [game.py](#).

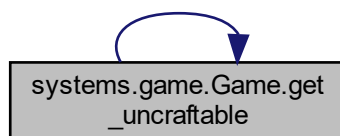
#### 7.2.3.5 `get_uncraftable()`

```
def systems.game.Game.get_uncraftable (
    self )
```

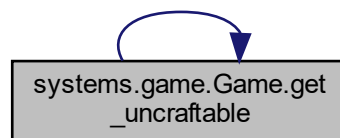
Returns a list of uncraftable recipes.

Definition at line 193 of file [game.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 7.2.3.6 remove\_item()

```
def systems.game.Game.remove_item (
    self,
    item,
    count = 1 )
```

Removes an Item from the Player's Inventory.

Keyword arguments:

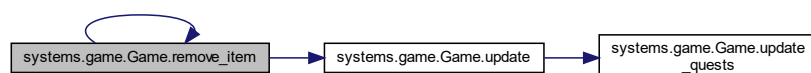
Keyword Arguments:

item -- id or Item object to remove

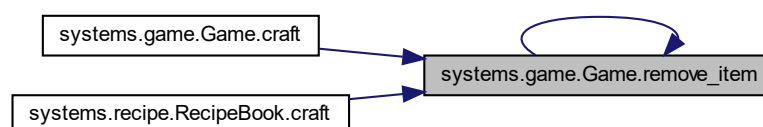
count -- amount of the item to add (default: 1)

Definition at line 150 of file [game.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 7.2.3.7 set\_selected\_quest()

```
def systems.game.Game.set_selected_quest (
    self,
    quest_id )
```

Returns Quest with given ID

Definition at line 133 of file [game.py](#).

### 7.2.3.8 setup()

```
def systems.game.Game.setup (
    self,
    people = None,
    locations = None,
    from_db = False,
    fps = {},
    items = [],
    recipes = [],
    quests = [] )
```

Setup the Game Object

Keyword Arguments:

- people -- a list of people objects
- locations -- a list of locations
- from\_db -- Whether or not to setup game from a Database - Currently unimplemented
- fps -- A dict of file-like objects used to initialize the Database
- items -- a list of Item objects
- recipes -- a list of Recipe objects
- quests -- a list of Quest objects

Definition at line 13 of file [game.py](#).

### 7.2.3.9 start\_quest()

```
def systems.game.Game.start_quest (
    self,
    quest_id = None,
    quest_object = None )
```

Starts a Quest, the updates the game object's state.

Keyword Arguments

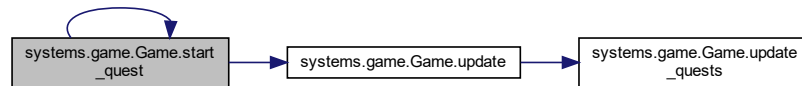
- quest\_id -- the ID of the quest
- quest\_object -- a quest object

Only one is required.

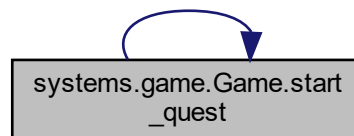


Definition at line 52 of file [game.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



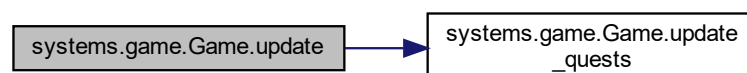
### 7.2.3.10 update()

```
def systems.game.Game.update (
    self )
```

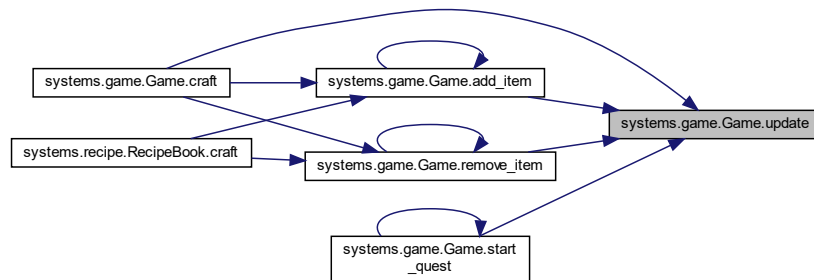
Updates the Game object's state

Definition at line 78 of file [game.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 7.2.3.11 update\_queries()

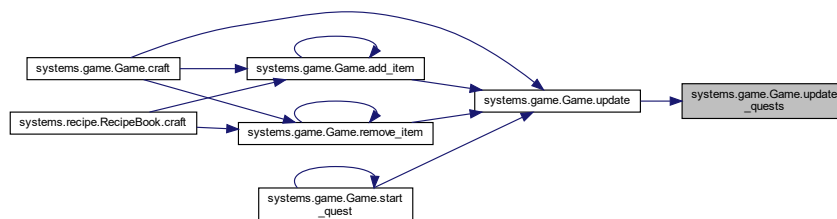
```
def systems.game.Game.update_queries (
    self )
```

Updates all active queries.

More to be implemented soon.

Definition at line 85 of file [game.py](#).

Here is the caller graph for this function:



## 7.2.4 Member Data Documentation

### 7.2.4.1 db

```
systems.game.Game.db
```

Definition at line 8 of file [game.py](#).

#### 7.2.4.2 location

`systems.game.Game.location`

Definition at line 42 of file [game.py](#).

#### 7.2.4.3 people

`systems.game.Game.people`

Definition at line 7 of file [game.py](#).

#### 7.2.4.4 player

`systems.game.Game.player`

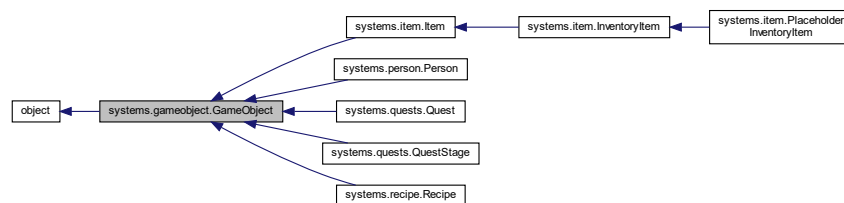
Definition at line 9 of file [game.py](#).

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

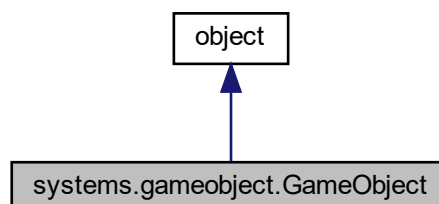
- [systems/game.py](#)

## 7.3 systems.gameobject.GameObject Class Reference

Inheritance diagram for `systems.gameobject.GameObject`:



Collaboration diagram for `systems.gameobject.GameObject`:



## Public Member Functions

- `def __init__ (self, id, name="", description="")`
- `def __repr__ (self)`
- `def __hash__ (self)`
- `def __eq__ (self, other)`
- `def __ne__ (self, other)`

## Public Attributes

- `id`
- `name`
- `description`

### 7.3.1 Detailed Description

Generic GameObject class.  
Inherited by all obtainable "things" in the game.

Definition at line 1 of file [gameobject.py](#).

### 7.3.2 Constructor & Destructor Documentation

#### 7.3.2.1 \_\_init\_\_()

```
def systems.gameobject.GameObject.__init__ (
    self,
    id,
    name = '',
    description = '' )
```

Reimplemented in [systems.item.PlaceholderInventoryItem](#), [systems.person.Person](#), [systems.quests.Quest](#), [systems.item.InventoryItem](#), [systems.item.Item](#), [systems.recipe.Recipe](#), and [systems.quests.QuestStage](#).

Definition at line 6 of file [gameobject.py](#).

### 7.3.3 Member Function Documentation

#### 7.3.3.1 `__eq__()`

```
def systems.gameobject.GameObject.__eq__ (
    self,
    other )
```

Reimplemented in [systems.recipe.Recipe](#).

Definition at line 21 of file [gameobject.py](#).

#### 7.3.3.2 `__hash__()`

```
def systems.gameobject.GameObject.__hash__ (
    self )
```

Reimplemented in [systems.recipe.Recipe](#).

Definition at line 18 of file [gameobject.py](#).

#### 7.3.3.3 `__ne__()`

```
def systems.gameobject.GameObject.__ne__ (
    self,
    other )
```

Definition at line 24 of file [gameobject.py](#).

#### 7.3.3.4 `__repr__()`

```
def systems.gameobject.GameObject.__repr__ (
    self )
```

Definition at line 10 of file [gameobject.py](#).

### 7.3.4 Member Data Documentation

#### 7.3.4.1 `description`

```
systems.gameobject.GameObject.description
```

Definition at line 9 of file [gameobject.py](#).

#### 7.3.4.2 id

`systems.gameobject.GameObject.id`

Definition at line 7 of file [gameobject.py](#).

#### 7.3.4.3 name

`systems.gameobject.GameObject.name`

Definition at line 8 of file [gameobject.py](#).

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

- [systems/gameobject.py](#)

## 7.4 systems.inventory.Inventory Class Reference

### Public Member Functions

- `def __init__ (self)`
- `def add_item (self, item_id, count=1)`
- `def remove_item (self, item, count=1)`
- `def __getitem__ (self, item_id)`
- `def __contains__ (self, key)`
- `def get_items (self)`  
*To be rewritten.*
- `def get_item_count (self, item)`
- `def set_current_item (self, item)`
- `def get_current_item (self)`

### Public Attributes

- `items`
- `current_item`

#### 7.4.1 Detailed Description

An Inventory.

Organizes items. It's basically a glorified list.

Definition at line 4 of file [inventory.py](#).

## 7.4.2 Constructor & Destructor Documentation

### 7.4.2.1 `__init__()`

```
def systems.inventory.Inventory.__init__ (
    self )
```

Definition at line 10 of file [inventory.py](#).

## 7.4.3 Member Function Documentation

### 7.4.3.1 `__contains__()`

```
def systems.inventory.Inventory.__contains__ (
    self,
    key )
```

Checks if Item is in Inventory.

```
Key can be:
str: returns True if str matches any ids in list of items
dict {'id': str, 'count': int}: if str is in list of items, returns True if count is greater than Item
tuple (str, int): if str in list, returns int >= item.count, else False
list [str, int]: same as tuple
list [[str, int]]: for every list, same as tuple. Returns whether all are True.
```

Definition at line 44 of file [inventory.py](#).

### 7.4.3.2 `__getitem__()`

```
def systems.inventory.Inventory.__getitem__ (
    self,
    item_id )
```

Definition at line 41 of file [inventory.py](#).

#### 7.4.3.3 add\_item()

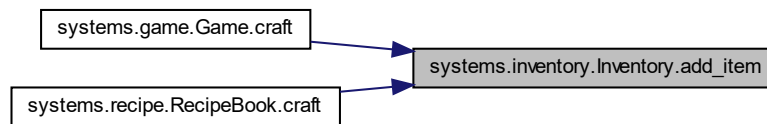
```
def systems.inventory.Inventory.add_item (
    self,
    item_id,
    count = 1 )

    Adds item to the Inventory.

    Arguments:
    item_id -- id or Item object of the Item to add
    count -- Amount of the Item to add (default: 1)
```

Definition at line 14 of file [inventory.py](#).

Here is the caller graph for this function:



#### 7.4.3.4 get\_current\_item()

```
def systems.inventory.Inventory.get_current_item (
    self )
```

Definition at line 98 of file [inventory.py](#).

#### 7.4.3.5 get\_item\_count()

```
def systems.inventory.Inventory.get_item_count (
    self,
    item )
```

Definition at line 90 of file [inventory.py](#).



#### 7.4.3.6 get\_items()

```
def systems.inventory.Inventory.get_items (
    self )
```

To be rewritten.

No tests written for these

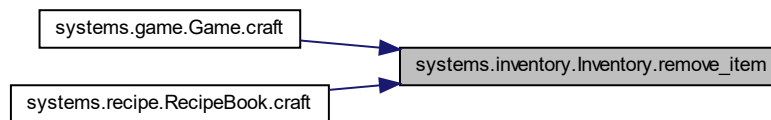
Definition at line 86 of file [inventory.py](#).

#### 7.4.3.7 remove\_item()

```
def systems.inventory.Inventory.remove_item (
    self,
    item,
    count = 1 )
```

Definition at line 37 of file [inventory.py](#).

Here is the caller graph for this function:



#### 7.4.3.8 set\_current\_item()

```
def systems.inventory.Inventory.set_current_item (
    self,
    item )
```

Definition at line 95 of file [inventory.py](#).

### 7.4.4 Member Data Documentation

#### 7.4.4.1 `current_item`

`systems.inventory.Inventory.current_item`

Definition at line 12 of file [inventory.py](#).

#### 7.4.4.2 `items`

`systems.inventory.Inventory.items`

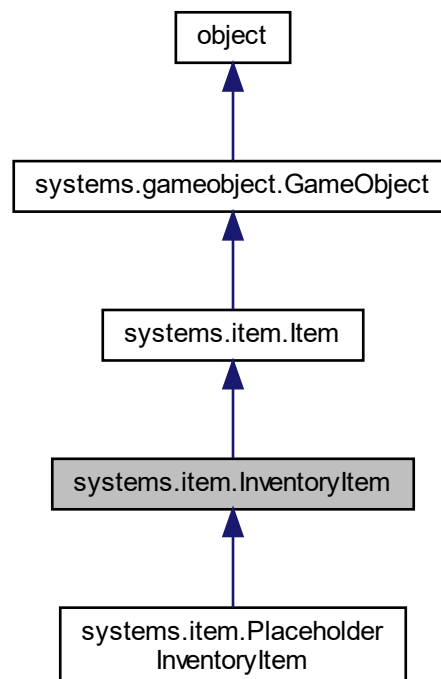
Definition at line 11 of file [inventory.py](#).

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

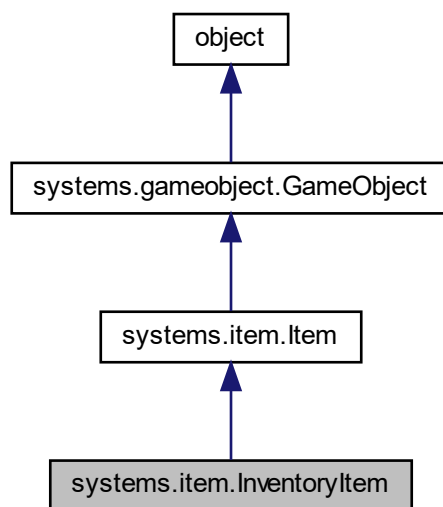
- [systems/inventory.py](#)

## 7.5 `systems.item.InventoryItem` Class Reference

Inheritance diagram for `systems.item.InventoryItem`:



Collaboration diagram for systems.item.InventoryItem:



### Public Member Functions

- `def __init__ (self, id, name, description="", image_path="", count=0)`
- `def set_count (self, count)`
- `def __iadd__ (self, other)`
- `def __isub__ (self, other)`

### Static Public Member Functions

- `def from_item (item, count=0)`
- `def from_dict (d)`

### Public Attributes

- `count`

#### 7.5.1 Detailed Description

Definition at line 23 of file `item.py`.

#### 7.5.2 Constructor & Destructor Documentation

### 7.5.2.1 `__init__()`

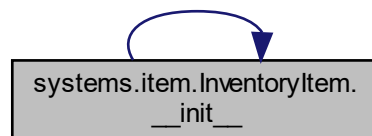
```
def systems.item.InventoryItem.__init__ (
    self,
    id,
    name,
    description = '',
    image_path = '',
    count = 0 )
```

Reimplemented from [systems.item.Item](#).

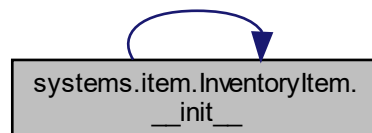
Reimplemented in [systems.item.PlaceholderInventoryItem](#).

Definition at line 24 of file [item.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



## 7.5.3 Member Function Documentation

### 7.5.3.1 `__iadd__()`

```
def systems.item.InventoryItem.__iadd__ (
    self,
    other )
```

Definition at line 37 of file [item.py](#).

### 7.5.3.2 `__isub__()`

```
def systems.item.InventoryItem.__isub__ (
    self,
    other )
```

Definition at line 47 of file [item.py](#).

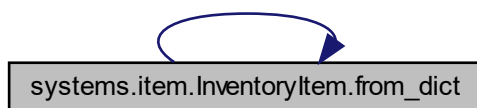
### 7.5.3.3 `from_dict()`

```
def systems.item.InventoryItem.from_dict (
    d ) [static]
```

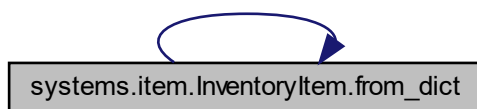
Reimplemented from [systems.item.Item](#).

Definition at line 67 of file [item.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 7.5.3.4 `from_item()`

```
def systems.item.InventoryItem.from_item (
    item,
    count = 0 ) [static]
```

Definition at line 60 of file [item.py](#).

### 7.5.3.5 set\_count()

```
def systems.item.InventoryItem.set_count (
    self,
    count )
```

Definition at line 30 of file [item.py](#).

## 7.5.4 Member Data Documentation

### 7.5.4.1 count

```
systems.item.InventoryItem.count
```

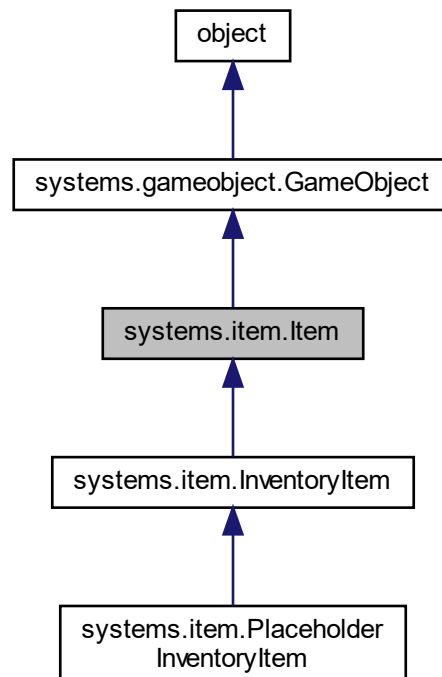
Definition at line 28 of file [item.py](#).

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

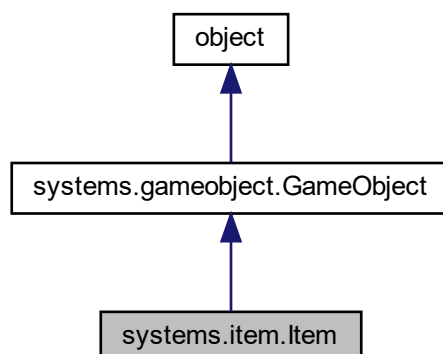
- [systems/item.py](#)

## 7.6 systems.item.Item Class Reference

Inheritance diagram for systems.item.Item:



Collaboration diagram for systems.item.Item:



### Public Member Functions

- `def __init__(self, id, name, description="", image_path="", icon_path="")`
- `def __str__(self)`

### Static Public Member Functions

- `def from_dict(d)`

### Public Attributes

- `image_path`
- `icon_path`

#### 7.6.1 Detailed Description

Definition at line 3 of file [item.py](#).

#### 7.6.2 Constructor & Destructor Documentation

### 7.6.2.1 `__init__()`

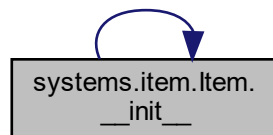
```
def systems.item.Item.__init__ (
    self,
    id,
    name,
    description = '',
    image_path = '',
    icon_path = "" )
```

Reimplemented from [systems.gameobject.GameObject](#).

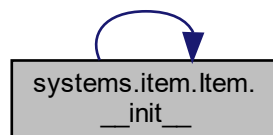
Reimplemented in [systems.item.PlaceholderInventoryItem](#), and [systems.item.InventoryItem](#).

Definition at line 4 of file [item.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



## 7.6.3 Member Function Documentation

### 7.6.3.1 `__str__()`

```
def systems.item.Item.__str__ (
    self )
```

Definition at line 8 of file [item.py](#).



### 7.6.3.2 from\_dict()

```
def systems.item.Item.from_dict (
    d ) [static]
```

Reimplemented in [systems.item.InventoryItem](#).

Definition at line 11 of file [item.py](#).

## 7.6.4 Member Data Documentation

### 7.6.4.1 icon\_path

`systems.item.Item.icon_path`

Definition at line 7 of file [item.py](#).

### 7.6.4.2 image\_path

`systems.item.Item.image_path`

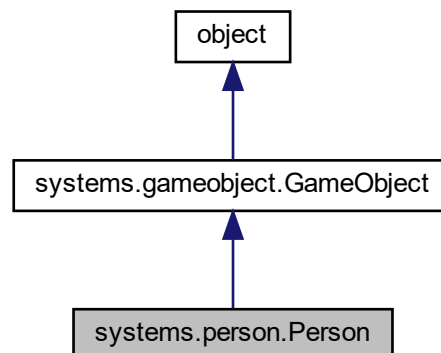
Definition at line 6 of file [item.py](#).

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

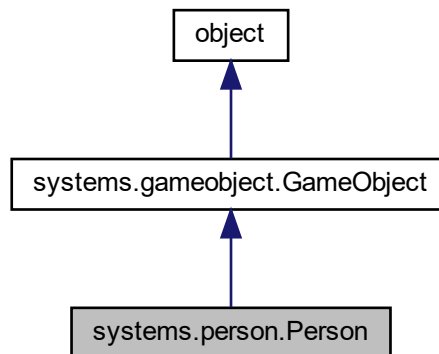
- [systems/item.py](#)

## 7.7 systems.person.Person Class Reference

Inheritance diagram for `systems.person.Person`:



Collaboration diagram for `systems.person.Person`:



## Public Member Functions

- `def __init__(self, id, name, description, base_affinity)`
- `def set_affinity(self, value)`
- `def increase_affinity(self, value=1)`
- `def decrease_affinity(self, value=1)`
- `def get_affinity(self)`
- `def __str__(self)`

## Public Attributes

- `base_affinity`
- `affinity`
- `location`

### 7.7.1 Detailed Description

Definition at line 3 of file `person.py`.

### 7.7.2 Constructor & Destructor Documentation

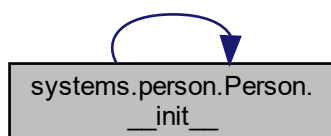
### 7.7.2.1 `__init__()`

```
def systems.person.Person.__init__ (
    self,
    id,
    name,
    description,
    base_affinity )
```

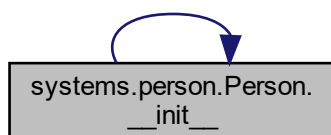
Reimplemented from [systems.gameobject.GameObject](#).

Definition at line 4 of file [person.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



## 7.7.3 Member Function Documentation

### 7.7.3.1 `__str__()`

```
def systems.person.Person.__str__ (
    self )
```

Definition at line 17 of file [person.py](#).

#### 7.7.3.2 decrease\_affinity()

```
def systems.person.Person.decrease_affinity (
    self,
    value = 1 )
```

Definition at line 13 of file [person.py](#).

#### 7.7.3.3 get\_affinity()

```
def systems.person.Person.get_affinity (
    self )
```

Definition at line 15 of file [person.py](#).

#### 7.7.3.4 increase\_affinity()

```
def systems.person.Person.increase_affinity (
    self,
    value = 1 )
```

Definition at line 11 of file [person.py](#).

#### 7.7.3.5 set\_affinity()

```
def systems.person.Person.set_affinity (
    self,
    value )
```

Definition at line 9 of file [person.py](#).

### 7.7.4 Member Data Documentation

#### 7.7.4.1 affinity

```
systems.person.Person.affinity
```

Definition at line 7 of file [person.py](#).

#### 7.7.4.2 base\_affinity

`systems.person.Person.base_affinity`

Definition at line 6 of file [person.py](#).

#### 7.7.4.3 location

`systems.person.Person.location`

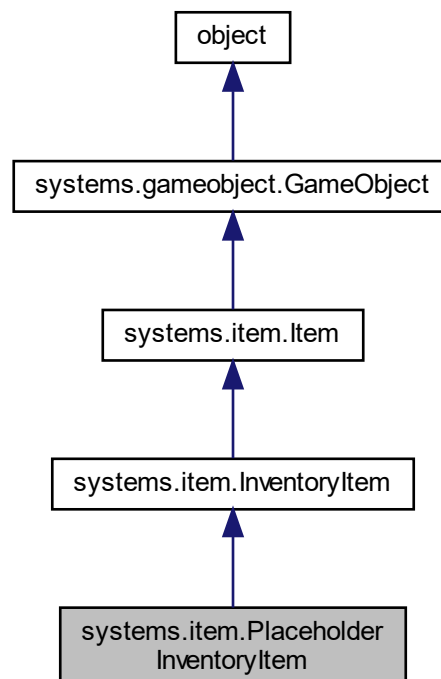
Definition at line 8 of file [person.py](#).

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

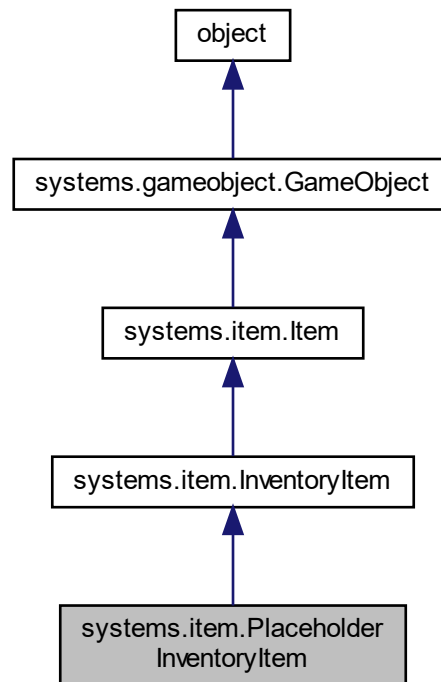
- [systems/person.py](#)

## 7.8 systems.item.PlaceholderInventoryItem Class Reference

Inheritance diagram for `systems.item.PlaceholderInventoryItem`:



Collaboration diagram for `systems.item.PlaceholderInventoryItem`:



## Public Member Functions

- `def \_\_init\_\_(self, id, count=0)`

## Additional Inherited Members

### 7.8.1 Detailed Description

Definition at line [72](#) of file [item.py](#).

### 7.8.2 Constructor & Destructor Documentation

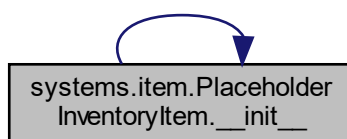
### 7.8.2.1 `__init__()`

```
def systems.item.PlaceholderInventoryItem.__init__ (
    self,
    id,
    count = 0 )
```

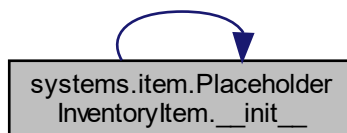
Reimplemented from [systems.item.InventoryItem](#).

Definition at line 73 of file [item.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:

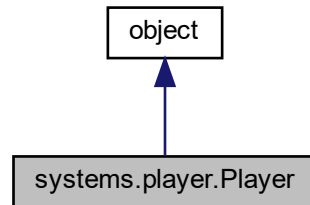


The documentation for this class was generated from the following file:

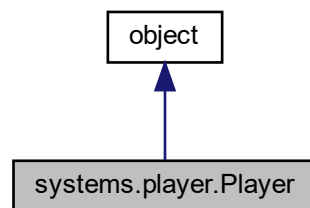
- [systems/item.py](#)

## 7.9 systems.player.Player Class Reference

Inheritance diagram for systems.player.Player:



Collaboration diagram for systems.player.Player:



### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self)
- def [get\\_craftable](#) (self)
- def [get\\_uncraftable](#) (self)
- def [craft](#) (self, recipe)
- def [add\\_item](#) (self, item, count)
- def [remove\\_item](#) (self, item, count)
- def [start\\_quest](#) (self, quest\_id, quest\_object)
- def [add\\_recipe](#) (self, recipe)
- def [get\\_quest](#) (self)
- def [set\\_selected\\_quest](#) (self, quest\_id)
- def [get\\_selected\\_quest](#) (self)
- def [get\\_items](#) (self)

### Public Attributes

- [inventory](#)
- [questlog](#)
- [recipebook](#)



## 7.9.1 Detailed Description

Definition at line 7 of file [player.py](#).

## 7.9.2 Constructor & Destructor Documentation

### 7.9.2.1 `__init__()`

```
def systems.player.Player.__init__ (  
    self )
```

Definition at line 8 of file [player.py](#).

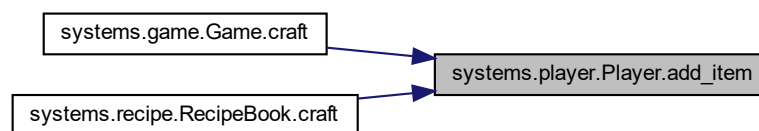
## 7.9.3 Member Function Documentation

### 7.9.3.1 `add_item()`

```
def systems.player.Player.add_item (  
    self,  
    item,  
    count )
```

Definition at line 22 of file [player.py](#).

Here is the caller graph for this function:

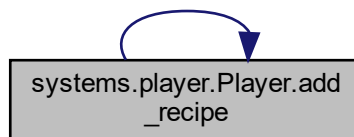


### 7.9.3.2 add\_recipe()

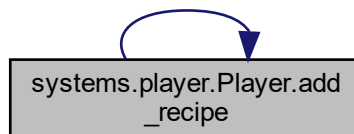
```
def systems.player.Player.add_recipe (
    self,
    recipe )
```

Definition at line 30 of file [player.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 7.9.3.3 craft()

```
def systems.player.Player.craft (
    self,
    recipe )
```

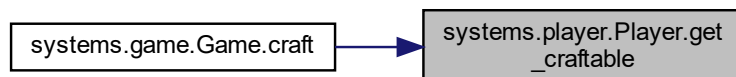
Definition at line 19 of file [player.py](#).

#### 7.9.3.4 get\_craftable()

```
def systems.player.Player.get_craftable (
    self )
```

Definition at line 13 of file [player.py](#).

Here is the caller graph for this function:



#### 7.9.3.5 get\_items()

```
def systems.player.Player.get_items (
    self )
```

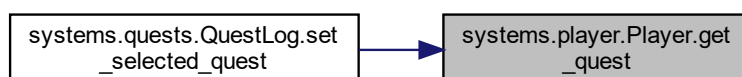
Definition at line 42 of file [player.py](#).

#### 7.9.3.6 get\_quest()

```
def systems.player.Player.get_quest (
    self )
```

Definition at line 33 of file [player.py](#).

Here is the caller graph for this function:



### 7.9.3.7 get\_selected\_quest()

```
def systems.player.Player.get_selected_quest (
    self )
```

Definition at line 39 of file [player.py](#).

### 7.9.3.8 get\_uncraftable()

```
def systems.player.Player.get_uncraftable (
    self )
```

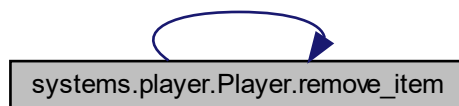
Definition at line 16 of file [player.py](#).

### 7.9.3.9 remove\_item()

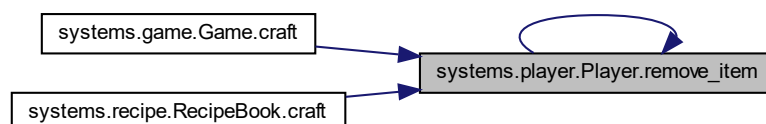
```
def systems.player.Player.remove_item (
    self,
    item,
    count )
```

Definition at line 24 of file [player.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



#### 7.9.3.10 set\_selected\_quest()

```
def systems.player.Player.set_selected_quest (
    self,
    quest_id )
```

Definition at line 36 of file [player.py](#).

#### 7.9.3.11 start\_quest()

```
def systems.player.Player.start_quest (
    self,
    quest_id,
    quest_object )
```

Definition at line 26 of file [player.py](#).

### 7.9.4 Member Data Documentation

#### 7.9.4.1 inventory

```
systems.player.Player.inventory
```

Definition at line 9 of file [player.py](#).

#### 7.9.4.2 questlog

```
systems.player.Player.questlog
```

Definition at line 10 of file [player.py](#).

#### 7.9.4.3 recipebook

```
systems.player.Player.recipebook
```

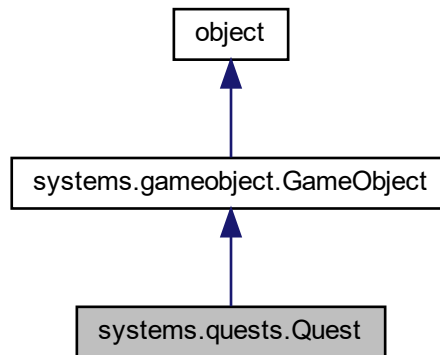
Definition at line 11 of file [player.py](#).

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

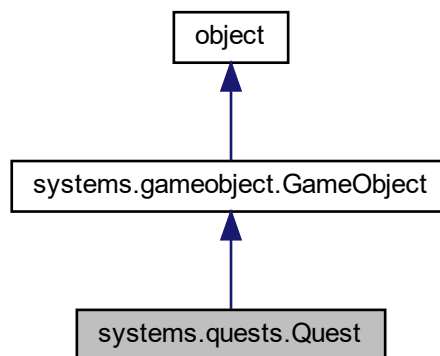
- [systems/player.py](#)

## 7.10 systems.quests.Quest Class Reference

Inheritance diagram for systems.quests.Quest:



Collaboration diagram for systems.quests.Quest:



### Public Member Functions

- def `__init__` (self, id, name, description, stages=[], first\_stage=None, final\_stage=None)
- def `start` (self, stage=None)
- def `get_stage` (self, stage=None)
- def `next_stage` (self, stage=None)
- def `substages_completed` (self, All=False)
- def `get_substages` (self)

## Static Public Member Functions

- def [from\\_dict](#) (d)

## Public Attributes

- [stages](#)
- [current\\_stage](#)
- [started](#)
- [complete](#)
- [first\\_stage](#)
- [final\\_stage](#)

### 7.10.1 Detailed Description

Definition at line 3 of file [quests.py](#).

### 7.10.2 Constructor & Destructor Documentation

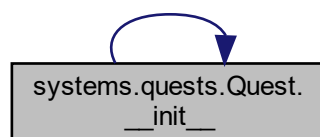
#### 7.10.2.1 `__init__()`

```
def systems.quests.Quest.__init__ (
    self,
    id,
    name,
    description,
    stages = [],
    first_stage = None,
    final_stage = None )
```

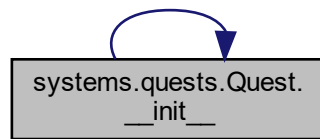
Reimplemented from [systems.gameobject.GameObject](#).

Definition at line 4 of file [quests.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 7.10.3 Member Function Documentation

#### 7.10.3.1 `from_dict()`

```
def systems.quests.Quest.from_dict (
    d ) [static]
```

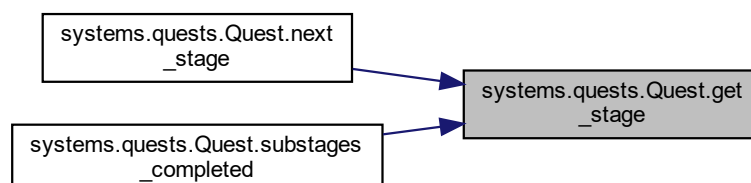
Definition at line 61 of file [quests.py](#).

#### 7.10.3.2 `get_stage()`

```
def systems.quests.Quest.get_stage (
    self,
    stage = None )
```

Definition at line 34 of file [quests.py](#).

Here is the caller graph for this function:





### 7.10.3.3 get\_substages()

```
def systems.quests.Quest.get_substages (
    self )
```

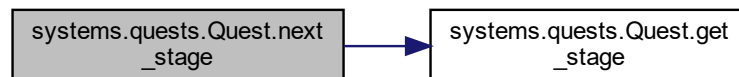
Definition at line 57 of file [quests.py](#).

### 7.10.3.4 next\_stage()

```
def systems.quests.Quest.next_stage (
    self,
    stage = None )
```

Definition at line 40 of file [quests.py](#).

Here is the call graph for this function:



### 7.10.3.5 start()

```
def systems.quests.Quest.start (
    self,
    stage = None )
```

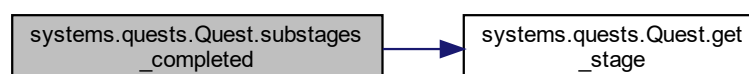
Definition at line 27 of file [quests.py](#).

### 7.10.3.6 substages\_completed()

```
def systems.quests.Quest.substages_completed (
    self,
    All = False )
```

Definition at line 50 of file [quests.py](#).

Here is the call graph for this function:



## 7.10.4 Member Data Documentation

### 7.10.4.1 complete

`systems.quests.Quest.complete`

Definition at line 19 of file [quests.py](#).

### 7.10.4.2 current\_stage

`systems.quests.Quest.current_stage`

Definition at line 17 of file [quests.py](#).

### 7.10.4.3 final\_stage

`systems.quests.Quest.final_stage`

Definition at line 21 of file [quests.py](#).

### 7.10.4.4 first\_stage

`systems.quests.Quest.first_stage`

Definition at line 20 of file [quests.py](#).

### 7.10.4.5 stages

`systems.quests.Quest.stages`

Definition at line 14 of file [quests.py](#).

#### 7.10.4.6 started

`systems.quests.Quest.started`

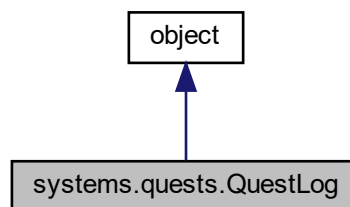
Definition at line 18 of file [quests.py](#).

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

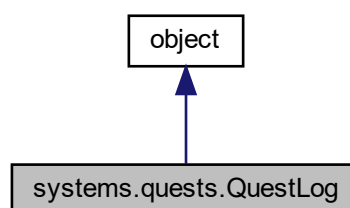
- [systems/quests.py](#)

## 7.11 systems.quests.QuestLog Class Reference

Inheritance diagram for `systems.quests.QuestLog`:



Collaboration diagram for `systems.quests.QuestLog`:



### Public Member Functions

- `def \_\_init\_\_ (self)`
- `def set\_selected\_quest (self, id)`
- `def get\_quest (self, id)`
- `def add\_quest (self, id, quest)`
- `def get\_active (self)`
- `def \_\_getitem\_\_ (self, key)`

## Public Attributes

- [quests](#)
- [selected\\_quest](#)

### 7.11.1 Detailed Description

Definition at line [98](#) of file [quests.py](#).

### 7.11.2 Constructor & Destructor Documentation

#### 7.11.2.1 `__init__()`

```
def systems.quests.QuestLog.__init__ (
    self )
```

Definition at line [99](#) of file [quests.py](#).

### 7.11.3 Member Function Documentation

#### 7.11.3.1 `__getitem__()`

```
def systems.quests.QuestLog.__getitem__ (
    self,
    key )
```

Definition at line [125](#) of file [quests.py](#).

#### 7.11.3.2 `add_quest()`

```
def systems.quests.QuestLog.add_quest (
    self,
    id,
    quest )
```

Definition at line [110](#) of file [quests.py](#).

### 7.11.3.3 get\_active()

```
def systems.quests.QuestLog.get_active (
    self )
```

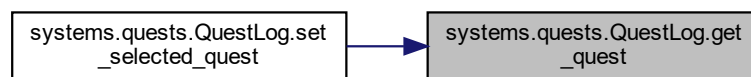
Definition at line 117 of file [quests.py](#).

### 7.11.3.4 get\_quest()

```
def systems.quests.QuestLog.get_quest (
    self,
    id )
```

Definition at line 107 of file [quests.py](#).

Here is the caller graph for this function:

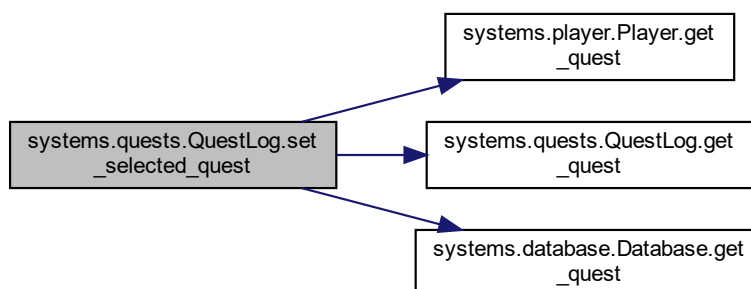


### 7.11.3.5 set\_selected\_quest()

```
def systems.quests.QuestLog.set_selected_quest (
    self,
    id )
```

Definition at line 103 of file [quests.py](#).

Here is the call graph for this function:



## 7.11.4 Member Data Documentation

### 7.11.4.1 `quests`

`systems.quests.QuestLog.quests`

Definition at line 100 of file [quests.py](#).

### 7.11.4.2 `selected_quest`

`systems.quests.QuestLog.selected_quest`

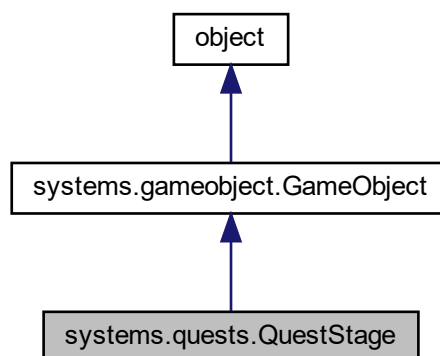
Definition at line 101 of file [quests.py](#).

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

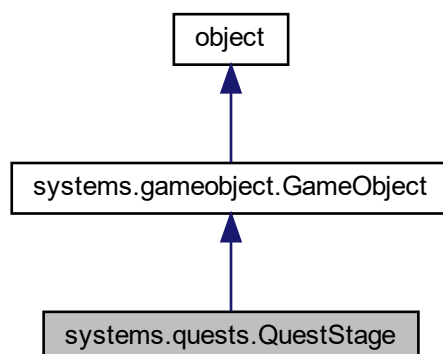
- [systems/quests.py](#)

## 7.12 `systems.quests.QuestStage` Class Reference

Inheritance diagram for `systems.quests.QuestStage`:



Collaboration diagram for systems.quests.QuestStage:



### Public Member Functions

- `def __init__(self, id, parent_quest_id, name="", description="", next_stage=0, substages=[], requirements={})`
- `def __str__(self)`

### Static Public Member Functions

- `def from_dict(d)`

### Public Attributes

- `complete`
- `substages`
- `requirements`
- `next_stage`

#### 7.12.1 Detailed Description

Definition at line 71 of file `quests.py`.

#### 7.12.2 Constructor & Destructor Documentation

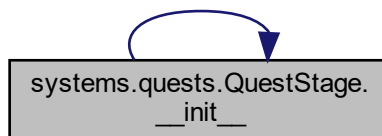
### 7.12.2.1 `__init__()`

```
def systems.quests.QuestStage.__init__ (
    self,
    id,
    parent_quest_id,
    name = '',
    description = '',
    next_stage = 0,
    substages = [],
    requirements = {} )
```

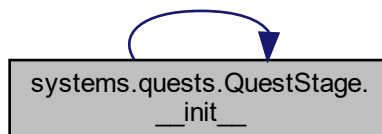
Reimplemented from [systems.gameobject.GameObject](#).

Definition at line 72 of file [quests.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



## 7.12.3 Member Function Documentation

### 7.12.3.1 `__str__()`

```
def systems.quests.QuestStage.__str__ (
    self )
```

Definition at line 90 of file [quests.py](#).



### 7.12.3.2 from\_dict()

```
def systems.quests.QuestStage.from_dict (
    d ) [static]
```

Definition at line 95 of file [quests.py](#).

## 7.12.4 Member Data Documentation

### 7.12.4.1 complete

```
systems.quests.QuestStage.complete
```

Definition at line 83 of file [quests.py](#).

### 7.12.4.2 next\_stage

```
systems.quests.QuestStage.next_stage
```

Definition at line 88 of file [quests.py](#).

### 7.12.4.3 requirements

```
systems.quests.QuestStage.requirements
```

Definition at line 87 of file [quests.py](#).

### 7.12.4.4 substages

```
systems.quests.QuestStage.substages
```

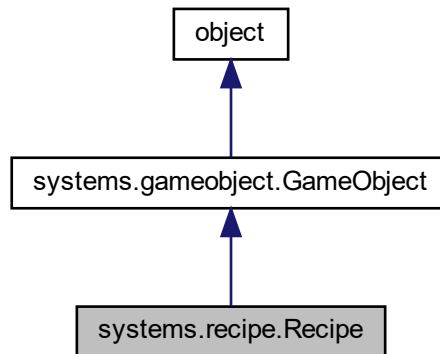
Definition at line 86 of file [quests.py](#).

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

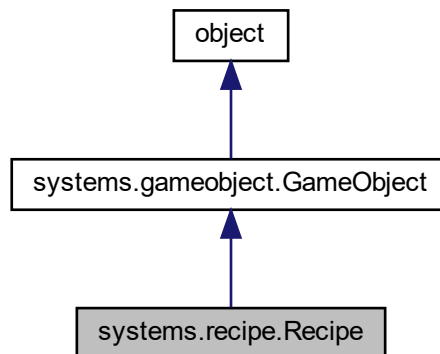
- [systems/quests.py](#)

## 7.13 systems.recipe.Recipe Class Reference

Inheritance diagram for systems.recipe.Recipe:



Collaboration diagram for systems.recipe.Recipe:



### Public Member Functions

- def `__init__` (self, id, name, requirements, items, description=)
- def `is_craftable` (self, item\_list)
- def `__hash__` (self)
- def `__eq__` (self, other)

### Static Public Member Functions

- def `from_dict` (d)

## Public Attributes

- [requirements](#)
- [items](#)

### 7.13.1 Detailed Description

Definition at line 3 of file [recipe.py](#).

### 7.13.2 Constructor & Destructor Documentation

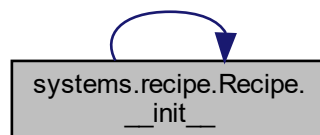
#### 7.13.2.1 `__init__()`

```
def systems.recipe.Recipe.__init__ (
    self,
    id,
    name,
    requirements,
    items,
    description = '' )
```

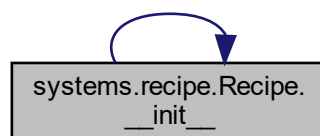
Reimplemented from [systems.gameobject.GameObject](#).

Definition at line 4 of file [recipe.py](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 7.13.3 Member Function Documentation

#### 7.13.3.1 `__eq__()`

```
def systems.recipe.Recipe.__eq__ (
    self,
    other )
```

Reimplemented from [systems.gameobject.GameObject](#).

Definition at line 25 of file [recipe.py](#).

#### 7.13.3.2 `__hash__()`

```
def systems.recipe.Recipe.__hash__ (
    self )
```

Reimplemented from [systems.gameobject.GameObject](#).

Definition at line 22 of file [recipe.py](#).

#### 7.13.3.3 `from_dict()`

```
def systems.recipe.Recipe.from_dict (
    d ) [static]
```

Definition at line 14 of file [recipe.py](#).

#### 7.13.3.4 `is_craftable()`

```
def systems.recipe.Recipe.is_craftable (
    self,
    item_list )
```

Definition at line 8 of file [recipe.py](#).

### 7.13.4 Member Data Documentation

#### 7.13.4.1 items

`systems.recipe.Recipe.items`

Definition at line 7 of file [recipe.py](#).

#### 7.13.4.2 requirements

`systems.recipe.Recipe.requirements`

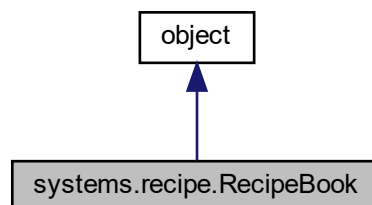
Definition at line 6 of file [recipe.py](#).

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

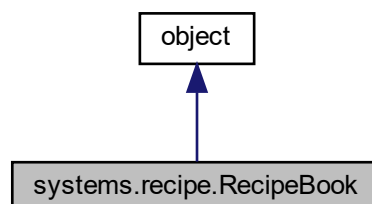
- [systems/recipe.py](#)

## 7.14 systems.recipe.RecipeBook Class Reference

Inheritance diagram for `systems.recipe.RecipeBook`:



Collaboration diagram for `systems.recipe.RecipeBook`:



## Public Member Functions

- `def __init__ (self)`
- `def add\_recipe (self, recipe)`
- `def remove\_recipe (self, recipe)`
- `def get\_recipe (recipe_id)`
- `def get\_craftable (self)`
- `def get\_uncraftable (self)`
- `def craft (self, recipe)`

## Public Attributes

- [recipes](#)
- [current\\_recipe](#)

### 7.14.1 Detailed Description

Definition at line [28](#) of file [recipe.py](#).

### 7.14.2 Constructor & Destructor Documentation

#### 7.14.2.1 `__init__()`

```
def systems.recipe.RecipeBook.__init__ (  
    self )
```

Definition at line [29](#) of file [recipe.py](#).

### 7.14.3 Member Function Documentation

#### 7.14.3.1 `add_recipe()`

```
def systems.recipe.RecipeBook.add_recipe (  
    self,  
    recipe )
```

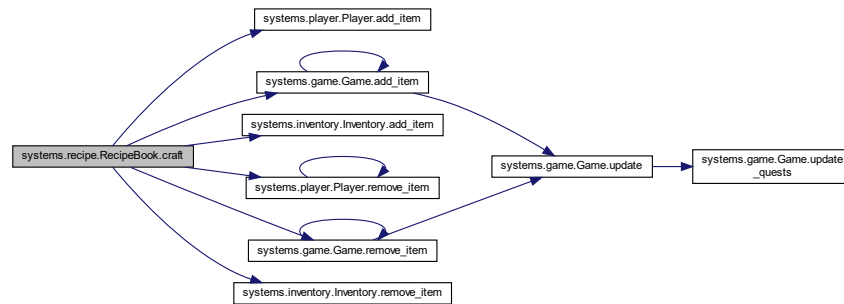
Definition at line [33](#) of file [recipe.py](#).

### 7.14.3.2 craft()

```
def systems.recipe.RecipeBook.craft (
    self,
    recipe )
```

Definition at line 46 of file [recipe.py](#).

Here is the call graph for this function:

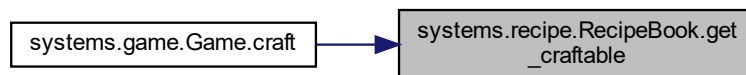


### 7.14.3.3 get\_craftable()

```
def systems.recipe.RecipeBook.get_craftable (
    self )
```

Definition at line 40 of file [recipe.py](#).

Here is the caller graph for this function:



### 7.14.3.4 get\_recipe()

```
def systems.recipe.RecipeBook.get_recipe (
    recipe_id )
```

Definition at line 38 of file [recipe.py](#).

#### 7.14.3.5 `get_uncraftable()`

```
def systems.recipe.RecipeBook.get_uncraftable (
    self )
```

Definition at line 43 of file [recipe.py](#).

#### 7.14.3.6 `remove_recipe()`

```
def systems.recipe.RecipeBook.remove_recipe (
    self,
    recipe )
```

Definition at line 35 of file [recipe.py](#).

### 7.14.4 Member Data Documentation

#### 7.14.4.1 `current_recipe`

```
systems.recipe.RecipeBook.current_recipe
```

Definition at line 31 of file [recipe.py](#).

#### 7.14.4.2 `recipes`

```
systems.recipe.RecipeBook.recipes
```

Definition at line 30 of file [recipe.py](#).

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

- [systems/recipe.py](#)



## Chapter 8

# File Documentation

### 8.1 systems/\_\_init\_\_.py File Reference

### 8.2 \_\_init\_\_.py

[Go to the documentation of this file.](#)

### 8.3 systems/database.py File Reference

#### Classes

- class [systems.database.Database](#)

#### Namespaces

- namespace [systems](#)
- namespace [systems.database](#)

### 8.4 database.py

[Go to the documentation of this file.](#)

```
00001 import json
00002 import os.path as path
00003 from .item import Item
00004 from .recipe import Recipe, RecipeBook
00005 class Database(object):
00006     def __init__(
00007         self,
00008         directory='data',
00009         item_fp='items.json',
00010         recipe_fp='recipes.json'
00011     ):
00012         self.directory, self.items_fp, self.recipes_fp = directory, item_fp, recipe_fp
00013         # self.item_list = self.load_items_from_file()
00014         self.items = {}
00015         self.recipes = {}
00016         self.quests = {}
00017         self.people = {}
```

```

00018         self.locations = {}
00019         # self.recipe_list = self.load_recipes_from_file()
00020
00021
00022     def verify(self):
00023         all_ids = [i.id for i in self.items.values() + self.recipes.values() + self.quests.values() +
self.people.values() + self.locations.values()]
00024         dupes = [(i, all_ids.count(i)) for i in all_ids if all_ids.count(i) > 1]
00025         if len(dupes) > 0:
00026             raise ValueError("\n".join(["ID: {} occurs {} times in Database".format(i[0], i[1]) for i
in dupes]))
00027
00028
00029     def load_items_from_file(self, file):
00030         data = json.load(file)
00031         items = {}
00032         for i in data:
00033             items[i['id']] if 'id' in i else i['name'].lower() = Item.from_dict(i)
00034         self.items = items
00035
00036     def load_recipes_from_file(self, file):
00037         data = json.load(file)
00038         recipes = {}
00039         for r in data:
00040             recipes[r['id']] if 'id' in r else r['name'].lower() = Recipe.from_dict(r)
00041         self.recipes = recipes
00042
00043     def get_item(self, item_id=""):
00044         return self.item_list[item_id]
00045
00046     def get_recipe(self, recipe_id=""):
00047         return self.recipes[recipe_id]
00048
00049     def get_quest(self, quest_id):
00050         pass
00051
00052
00053     def set_items(self, items):
00054         for i in items:
00055             self.items[i.id] = i
00056
00057     def __getitem__(self, key):
00058         return [i for i in self.items.values() + self.recipes.values() + self.quests.values() +
self.people.values() + self.locations.values() if i.id == key][0]

```

## 8.5 systems/game.py File Reference

### Classes

- class [systems.game.Game](#)

### Namespaces

- namespace [systems](#)
- namespace [systems.game](#)

## 8.6 game.py

[Go to the documentation of this file.](#)

```

00001 from .database import Database
00002 from .player import Player
00003 from .quests import Quest
00004 from .recipe import Recipe
00005 class Game(object):
00006     def __init__(self):
00007         self.people = {}
00008         self.db = Database()
00009         self.player = Player()
00010         # characters = {

```

```

00011     # # list of character and info
00012     # }
00013     def setup(self,
00014         people = None,
00015         locations = None,
00016         from_db = False,
00017         fps = {},
00018         items = [],
00019         recipes = [],
00020         quests = []
00021     ):
00022         """Setup the Game Object
00023
00024         Keyword Arguments:
00025         people -- a list of people objects
00026         locations -- a list of locations
00027         from_db -- Whether or not to setup game from a Database - Currently unimplemented
00028         fps -- A dict of file-like objects used to initialize the Database
00029         items -- a list of Item objects
00030         recipes -- a list of Recipe objects
00031         quests -- a list of Quest objects
00032         """
00033         if from_db:
00034             # Todo: make this actually work. Issue is, opening Files in RENPY is annoying.
00035             return
00036             self.db.load_items_from_file(fps['items'])
00037             self.db.load_recipes_from_file(fps['recipes'])
00038         else:
00039             if people:
00040                 self.people = people
00041             if locations:
00042                 self.location = locations
00043             if items:
00044                 self.db.set_items(items)
00045             if recipes:
00046                 for i in recipes:
00047                     self.db.recipes[i.id] = i
00048             if quests:
00049                 for i in quests:
00050                     self.db.quests[i.id] = i
00051
00052     def start_quest(self, quest_id= None, quest_object = None):
00053         """
00054         Starts a Quest, the updates the game object's state.
00055
00056         Keyword Arguments
00057         quest_id -- the ID of the quest
00058         quest_object -- a quest object
00059         Only one is required.
00060         """
00061         if not (isinstance(quest_id, str) or type(quest_id) != None):
00062             raise TypeError("quest_id is not a <str>, is type {}".format(type(quest_id)))
00063         if not (isinstance(quest_object, Quest) or type(quest_object) != None):
00064             raise TypeError("quest_object is not a <Quest>")
00065         if quest_id and not quest_object:
00066             self.player.start_quest(quest_id, self.db.get_quest(quest_id))
00067         elif quest_id and quest_object:
00068             self.player.start_quest(quest_id, quest_object)
00069         elif quest_object:
00070             self.player.start_quest(quest_object.id, quest_object)
00071         else:
00072             raise ValueError()
00073         self.update()
00074
00075
00076
00077
00078     def update(self):
00079         """
00080         Updates the Game object's state
00081         """
00082         self.update_quests()
00083         pass
00084
00085     def update_quests(self):
00086         """
00087         Updates all active quests.
00088
00089         More to be implemented soon.
00090         """
00091         for q in self.player.questlog.get_active(): # Get all active quests
00092             stage = q.get_stage() # Get all of q's stages
00093             stage_complete = True # intermediate variable
00094
00095             # TODO: implement stage check logic
00096             for substage in stage.substages:
00097                 r = substage.requirements

```

```

00098         complete = True
00099         if "item" in r:
00100             # print(reqs['item'])
00101             items_present = [(i in self.player.inventory) for i in r['item']]['ids']]
00102             if r['item']['all']:
00103                 if not all(items_present):
00104                     complete = False
00105             else:
00106                 if not any(items_present):
00107                     complete = False
00108         if complete:
00109             substage.complete = True
00110
00111         reqs = stage.requirements
00112         if "substage" in reqs:
00113             completed_stages = [s.complete for s in stage.substages]
00114             if reqs['substage'] == 'all':
00115                 if not all(completed_stages):
00116                     stage_complete = False
00117             if reqs['substage'] == 'any':
00118                 if not any(completed_stages):
00119                     stage_complete = False
00120         if "item" in reqs:
00121             # print(reqs['item'])
00122             items_present = [(i in self.player.inventory) for i in reqs['item']]['ids']]
00123             if reqs['item']['all']:
00124                 if not all(items_present):
00125                     stage_complete = False
00126             else:
00127                 if not any(items_present):
00128                     stage_complete = False
00129         if stage_complete:
00130             stage.complete = True
00131             q.next_stage()
00132
00133     def set_selected_quest(self, quest_id):
00134         """Returns Quest with given ID"""
00135         self.player.questlog.set_selected_quest(quest_id)
00136
00137     def get_selected_quest(self):
00138         """Returns the currently selected Quest"""
00139         return self.player.questlog.selected_quest
00140
00141     def add_item(self, item, count=1):
00142         """Adds an item to the Player's inventory.
00143
00144         Keyword Arguments:
00145         item -- id or Item object to add
00146         count -- amount of the item to add (default: 1)
00147         """
00148         self.player.add_item(item, count)
00149         self.update()
00150     def remove_item(self, item, count=1):
00151         """
00152         Removes an Item from the Player's Inventory.
00153
00154         Keyword arguments:
00155         Keyword Arguments:
00156         item -- id or Item object to remove
00157         count -- amount of the item to add (default: 1)
00158         """
00159         self.player.remove_item(item, count)
00160         self.update()
00161
00162     def craft(self, recipe):
00163         """
00164         Crafts Recipe.
00165
00166         Checks player's inventory for required Item(s), removes them from the player's inventory, adds
00167         the crafted Item(s), then updates the Game object.
00168
00169         Arguments:
00170         recipe -- id or Recipe object to craft
00171         """
00172         if isinstance(recipe, str):
00173             if recipe in [r.id for r in self.get_craftable()]: # Check if ID is in craftable recipe
00174                 list
00175                 for req in recipe.requirements:
00176                     self.remove_item(req[0], req[1])
00177                 for i in recipe.items:
00178                     self.add_item(self.db[i[0]], i[1])
00179             elif isinstance(recipe, Recipe):
00180                 if recipe in self.get_craftable(): # Check if Recipe is in the craftable recipe list
00181                     for req in recipe.requirements:
00182                         self.remove_item(req[0], req[1])
00183                     for i in recipe.items:
00184                         self.add_item(self.db[i[0]], i[1])

```

```

00183
00184         else:
00185             raise TypeError("Must be of type <str> or <Recipe>, not {}".format(type(recipe)))
00186         self.update()
00187
00188
00189     def get_craftable(self):
00190         """Returns a list of craftable recipes."""
00191         return self.player.get_craftable()
00192
00193     def get_uncraftable(self):
00194         """Returns a list of uncraftable recipes."""
00195         return self.player.get_uncraftable()
00196
00197
00198

```

## 8.7 systems/gameobject.py File Reference

### Classes

- class [systems.gameobject.GameObject](#)

### Namespaces

- namespace [systems](#)
- namespace [systems.gameobject](#)

## 8.8 gameobject.py

[Go to the documentation of this file.](#)

```

00001 class GameObject(object):
00002     """
00003     Generic GameObject class.
00004     Inherited by all obtainable "things" in the game.
00005     """
00006     def __init__(self, id, name="", description=""):
00007         self.id = id
00008         self.name = name
00009         self.description = description
00010     def __repr__(self):
00011         return "<{} id='{}' name='{}' desc='{}'>".format(
00012             self.__class__.__name__,
00013             self.id,
00014             self.name,
00015             self.description
00016         )
00017
00018     def __hash__(self):
00019         return hash((self.name, self.description))
00020
00021     def __eq__(self, other):
00022         return (self.id, self.name, self.description, self.__class__.__name__) == (other.id, other.name,
other.description, other.__class__.__name__)
00023
00024     def __ne__(self, other):
00025         return not(self == other)
00026

```

## 8.9 systems/inventory.py File Reference

### Classes

- class [systems.inventory.Inventory](#)

## Namespaces

- namespace [systems](#)
- namespace [systems.inventory](#)

## 8.10 inventory.py

[Go to the documentation of this file.](#)

```

00001 from .item import InventoryItem, Item, PlaceholderInventoryItem
00002 from .recipe import Recipe
00003
00004 class Inventory():
00005     """
00006     An Inventory.
00007
00008     Organizes items. It's basically a glorified list.
00009     """
00010     def __init__(self):
00011         self.items = {}
00012         self.current_item = InventoryItem("Nothing", "I should look at something first...", 0)
00013
00014     def add_item(self, item_id, count=1):
00015         """
00016         Adds item to the Inventory.
00017
00018         Arguments:
00019         item_id -- id or Item object of the Item to add
00020         count -- Amount of the Item to add (default: 1)
00021         """
00022         if not isinstance(count, int): # if type(count) != int:
00023             raise TypeError("{} is of type {}, not {}".format(count, type(count), int))
00024         if isinstance(item_id, str) or isinstance(item_id, unicode):
00025             if item_id in self.items:
00026                 self.items[item_id] += count
00027             else:
00028                 self.items[item_id] = PlaceholderInventoryItem(item_id, count)
00029         elif isinstance(item_id, Item): # elif type(item_id) == (Item):
00030             if item_id.id not in self.items:
00031                 self.items[item_id.id] = InventoryItem.from_item(item_id, count)
00032             else:
00033                 self.items[item_id.id] += count
00034         else:
00035             raise TypeError("{} is of type {}, not {} or {}".format(item_id, type(item_id), str,
Item))
00036
00037     def remove_item(self, item, count=1):
00038         if item in self.items:
00039             self.items[item] -= count
00040
00041     def __getitem__(self, item_id):
00042         return self.items[item_id]
00043
00044     def __contains__(self, key):
00045         """Checks if Item is in Inventory.
00046
00047         Key can be:
00048         str: returns True if str matches any ids in list of items
00049         dict {'id': str, 'count': int}: if str is in list of items, returns True if count is greater
than Item.count, else False
00050         tuple (str, int): if str in list, returns int >= item.count, else False
00051         list [str, int]: same as tuple
00052         list [[str, int]]: for every list, same as tuple. Returns whether all are True.
00053         """
00054         if isinstance(key, str):
00055             return key in self.items
00056         elif isinstance(key, dict):
00057             if key['id'] in self.items:
00058                 return self.items[key['id']].count >= key['count']
00059             else:
00060                 return False
00061         elif isinstance(key, tuple): #type(key) == tuple:
00062             if key[0] in self.items:
00063                 return self.items[key[0]].count >= key[1]
00064             else:
00065                 return False
00066         elif isinstance(key, list): # type(key) == list:
00067             if all([isinstance(i, tuple) for i in key]):
00068                 if all([len(i) == 2 for i in key]):
00069                     if all([i[0] in self.items for i in key]):
00070                         return all([self.items[i[0]].count >= i[1] for i in key])

```

```

00071         else:
00072             return False
00073         else:
00074             raise ValueError("All tuples must be of types (<str>, <int>")
00075         elif len(key) == 2:
00076             if key[0] in self.items:
00077                 return self.items[key[0]].count >= key[1]
00078             else:
00079                 return False
00080         else:
00081             raise ValueError("Must be a list of tuples of types (<str>, <int>")
00082     else:
00083         raise ValueError("Input must be of types <str>, <dict>, <tuple>, list[<str>, <int>] or
list[(<str>,<int>)], not {}".format(type(key)))
00084
00085
00086     def get_items(self):
00087         return [i for i in self.items.values() if i.count > 0]
00088         # return [{"{} x{}".format(i[0], i[1]), i[0]} for i in list(filter(lambda i: i[1] > 0,
self.items.items()))]
00089
00090     def get_item_count(self, item):
00091         if self.items.has_key(item):
00092             return self.items[item]
00093         return 0
00094
00095     def set_current_item(self, item):
00096         self.current_item = item
00097
00098     def get_current_item(self):
00099         return self.current_item

```

## 8.11 systems/item.py File Reference

### Classes

- class [systems.item.Item](#)
- class [systems.item.InventoryItem](#)
- class [systems.item.PlaceholderInventoryItem](#)

### Namespaces

- namespace [systems](#)
- namespace [systems.item](#)

## 8.12 item.py

[Go to the documentation of this file.](#)

```

00001 from .gameobject import GameObject
00002
00003 class Item(GameObject):
00004     def __init__(self, id, name, description= "", image_path="", icon_path = ""):
00005         super(Item, self).__init__(id, name, description)
00006         self.image_path = image_path
00007         self.icon_path = icon_path
00008     def __str__(self):
00009         return self.name
00010     @staticmethod
00011     def from_dict(d):
00012         if 'id' in d:
00013             id = d['id']
00014         else:
00015             id = d['name']
00016
00017         if 'image_path' in d:
00018             ip = d['image_path']
00019         else:
00020             ip = "items/{}".format(id)

```

```

00021         return Item(id, d['name'], d['description'], image_path=ip)
00022
00023 class InventoryItem(Item):
00024     def __init__(self, id, name, description="", image_path="", count = 0):
00025         super(InventoryItem, self).__init__(id, name, description, image_path)
00026         if count < 0:
00027             raise ValueError("Count cannot be less than 0")
00028         self.count = count
00029
00030     def set_count(self, count):
00031         if not isinstance(count, int):
00032             raise TypeError("{} is of type {}, not {}".format(count, type(count), int))
00033         if count < 0:
00034             raise ValueError("Count cannot be less than 0")
00035
00036         self.count = count
00037     def __iadd__(self, other):
00038         if not isinstance(other, int):
00039             raise TypeError("{} is of type {}, not {}".format(other, type(other), int))
00040
00041         if other < 0:
00042             raise ValueError("Can't add negative number of items".format(other, self.count))
00043
00044         self.count += other
00045         return self
00046
00047     def __isub__(self, other):
00048         if not isinstance(other, int):
00049             raise TypeError("{} is of type {}, not {}".format(other, type(other), int))
00050
00051         if other < 0:
00052             raise ValueError("Can't remove a negative number of items".format(other, self.count))
00053         if self.count - other < 0:
00054             raise ValueError("Removing {} to current count ({} will result in a count less than
00055 0".format(other, self.count))
00056
00057         self.count -= other
00058         return self
00059
00060     @staticmethod
00061     def from_item(item, count = 0):
00062         if not isinstance(item, Item):# if type(item) != Item:
00063             raise TypeError("{} is of type {}, not {}".format(item, type(item), Item))
00064
00065         return InventoryItem(item.id, item.name, item.description, item.image_path, count)
00066
00067     @staticmethod
00068     def from_dict(d):
00069         if not isinstance(d, dict):
00070             raise TypeError("{} is of type {}, not {}".format(d, type(d), dict))
00071         return InventoryItem.from_item(super(InventoryItem, InventoryItem).from_dict(d))
00072
00073 class PlaceholderInventoryItem(InventoryItem):
00074     def __init__(self, id, count=0):
00075         super(PlaceholderInventoryItem, self).__init__(id, "placeholder", "placeholder",
"placeholder", count=count)

```

## 8.13 systems/person.py File Reference

### Classes

- class `systems.person.Person`

### Namespaces

- namespace `systems`
- namespace `systems.person`



## 8.14 person.py

[Go to the documentation of this file.](#)

```
00001 from .gameobject import GameObject
00002
00003 class Person(GameObject):
00004     def __init__(self, id, name, description, base_affinity):
00005         super(Person, self).__init__(id, name, description)
00006         self.base_affinity = base_affinity
00007         self.affinity = base_affinity
00008         self.location = ""
00009     def set_affinity(self, value):
00010         self.affinity = value
00011     def increase_affinity(self, value=1):
00012         self.affinity += value
00013     def decrease_affinity(self, value=1):
00014         self.affinity -= value
00015     def get_affinity(self):
00016         return self.affinity
00017     def __str__(self):
00018         return self.name
00019
```

## 8.15 systems/player.py File Reference

### Classes

- class [systems.player.Player](#)

### Namespaces

- namespace [systems](#)
- namespace [systems.player](#)

## 8.16 player.py

[Go to the documentation of this file.](#)

```
00001 from .gameobject import GameObject
00002 from .item import Item, InventoryItem
00003 from .inventory import Inventory
00004 from .quests import QuestLog
00005 from .recipe import RecipeBook, Recipe
00006
00007 class Player(object):
00008     def __init__(self):
00009         self.inventory = Inventory()
00010         self.questlog = QuestLog()
00011         self.recipebook = RecipeBook()
00012
00013     def get_craftable(self):
00014         return [r for r in self.recipebook.recipes.values() if r.requirements in self.inventory]
00015
00016     def get_uncraftable(self):
00017         return [r for r in self.recipebook.recipes.values() if not r.requirements in self.inventory]
00018
00019     def craft(self, recipe):
00020         pass
00021
00022     def add_item(self, item, count):
00023         self.inventory.add_item(item, count)
00024     def remove_item(self, item, count):
00025         self.inventory.remove_item(item, count)
00026     def start_quest(self, quest_id, quest_object):
00027         self.questlog.add_quest(quest_id, quest_object)
00028         # self.questlog
00029
00030     def add_recipe(self, recipe):
```

```

00031         self.recipebook.add_recipe(recipe)
00032
00033     def get_quest(self):
00034         pass
00035
00036     def set_selected_quest(self, quest_id):
00037         pass
00038
00039     def get_selected_quest(self):
00040         pass
00041
00042     def get_items(self):
00043         return self.inventory.get_items()

```

## 8.17 systems/quests.py File Reference

### Classes

- class [systems.quests.Quest](#)
- class [systems.quests.QuestStage](#)
- class [systems.quests.QuestLog](#)

### Namespaces

- namespace [systems](#)
- namespace [systems.quests](#)

## 8.18 quests.py

[Go to the documentation of this file.](#)

```

00001 from .gameobject import GameObject
00002
00003 class Quest(GameObject):
00004     def __init__(
00005         self,
00006         id,
00007         name,
00008         description,
00009         stages=[],
00010         first_stage=None,
00011         final_stage=None,
00012     ):
00013         super(Quest, self).__init__(id, name, description)
00014         self.stages = {}
00015         for qs in stages:
00016             self.stages[qs.id] = qs
00017         self.current_stage = 0
00018         self.started = False
00019         self.complete = False
00020         self.first_stage = first_stage
00021         self.final_stage = final_stage
00022         # if final_stage == None:
00023         #     self.final_stage = len(stages)
00024         # else:
00025         #     self.final_stage = final_stage
00026
00027     def start(self, stage=None):
00028         self.started = True
00029         if stage == None:
00030             self.current_stage = self.first_stage
00031         else:
00032             self.current_stage = stage
00033
00034     def get_stage(self, stage=None):
00035         if stage == None:
00036             return self.stages[".".join([self.id, str(self.current_stage)])]
00037         else:
00038             return self.stages[".".join([self.id, str(stage)]]

```

```

00039
00040     def next_stage(self, stage = None):
00041         if stage:
00042             self.current_stage = stage
00043         elif self.get_stage().next_stage:
00044             self.current_stage = self.get_stage().next_stage
00045         elif self.get_stage().id == self.final_stage:
00046             self.complete = True
00047         else:
00048             print("Not sure what you want from me here. Moving on...")
00049
00050     def substages_completed(self, All=False):
00051         if All:
00052             return all([q.complete for q in self.get_stage().substages])
00053         else:
00054             return any([q.complete for q in self.get_stage().substages])
00055
00056
00057     def get_substages(self):
00058         return self.substages
00059
00060     @staticmethod
00061     def from_dict(d):
00062         return Quest(
00063             d['id'],
00064             d['name'],
00065             d['description'],
00066             stages=[QuestStage.from_dict(s) for s in d['stages']],
00067             first_stage=d['first_stage'],
00068         )
00069
00070
00071 class QuestStage(GameObject):
00072     def __init__(
00073         self,
00074         id,
00075         parent_quest_id,
00076         name="",
00077         description="",
00078         next_stage=0,
00079         substages = [],
00080         requirements= {}
00081     ):
00082         super(QuestStage, self).__init__(". ".join([parent_quest_id, str(id)]), name, description)
00083         self.complete = False
00084         for ss in substages:
00085             pass
00086         self.substages = substages
00087         self.requirements = requirements
00088         self.next_stage = next_stage
00089
00090     def __str__(self):
00091         return self.name
00092
00093
00094     @staticmethod
00095     def from_dict(d):
00096         return QuestStage(name=d['name'], description=d['description'])
00097
00098 class QuestLog(object):
00099     def __init__(self):
00100         self.quests = {}
00101         self.selected_quest = Quest("placeholder", "Quests", "Things to do!",
00102             stages=[QuestStage("0", "")])
00103
00104     def set_selected_quest(self, id):
00105         self.selected_quest = self.get_quest(id)
00106
00107     def get_quest(self, id):
00108         return self.quests[id]
00109
00110     def add_quest(self, id, quest):
00111         self.quests[id] = quest
00112         self.quests[id].start()
00113         if self.selected_quest.id == "placeholder":
00114             self.selected_quest = quest
00115
00116
00117     def get_active(self):
00118         return [q for q in self.quests.values() if q.started and not q.complete]
00119         pass
00120         # qs = [q.get_stage() for q in self.quests.values() if (q.started and not q.complete)]
00121         # sq = [q.substages for q in qs]
00122         # # python is objectively a stupid language. This is just to flatten a list. Fuck's sake.
00123         # return qs + [s for sublist in sq for s in sublist]
00124

```

```

00125     def __getitem__(self, key):
00126         return self.quests[key]
00127
00128

```

## 8.19 systems/recipe.py File Reference

### Classes

- class [systems.recipe.Recipe](#)
- class [systems.recipe.RecipeBook](#)

### Namespaces

- namespace [systems](#)
- namespace [systems.recipe](#)

## 8.20 recipe.py

[Go to the documentation of this file.](#)

```

00001 from .gameobject import GameObject
00002
00003 class Recipe(GameObject):
00004     def __init__(self, id, name, requirements, items, description=""):
00005         super(Recipe, self).__init__(id, name, description)
00006         self.requirements = requirements
00007         self.items = items
00008     def is_craftable(self, item_list):
00009
00010         pass
00011         # return all([item_list[i] >= c for i, c in self.requirements.items()])
00012
00013     @staticmethod
00014     def from_dict(d):
00015         return Recipe(
00016             d['name'],
00017             item=d['item'],
00018             requirements=d['requirements'],
00019             description=d['description'],
00020         )
00021
00022     def __hash__(self):
00023         return hash((self.name, self.item))
00024
00025     def __eq__(self, other):
00026         return (self.name, self.requirements, self.item) == (other.name, other.requirements,
00027             other.item)
00028
00029 class RecipeBook(object):
00030     def __init__(self):
00031         self.recipes = {}
00032         self.current_recipe = Recipe("", "Nothing here", [("nothing", 0)], [(None, 0)])
00033
00034     def add_recipe(self, recipe):
00035         self.recipes[recipe.id] = recipe
00036
00037     def remove_recipe(self, recipe):
00038         self.recipes.remove(recipe)
00039
00040     def get_recipe(recipe_id):
00041         return self.recipes[recipe_id]
00042
00043     def get_craftable(self):
00044         return [i for i in self.recipes if i.is_craftable(self.items)]
00045
00046     def get_uncraftable(self):
00047         return [i for i in self.recipes if not i.is_craftable(self.items)]
00048
00049     def craft(self, recipe):
00050         if recipe in self.recipes:
00051             if recipe.is_craftable(self.items):
00052                 self.add_item(recipe.item)
00053                 for i, c in recipe.requirements.items():
00054                     self.remove_item(i, c)

```

## 8.21 systems/requirements\_syntax.md File Reference



# Index

- `__contains__`
    - `systems.inventory.Inventory`, [24](#)
  - `__eq__`
    - `systems.gameobject.GameObject`, [22](#)
    - `systems.recipe.Recipe`, [56](#)
  - `__getitem__`
    - `systems.inventory.Inventory`, [24](#)
    - `systems.quests.QuestLog`, [49](#)
  - `__hash__`
    - `systems.gameobject.GameObject`, [22](#)
    - `systems.recipe.Recipe`, [56](#)
  - `__iadd__`
    - `systems.item.InventoryItem`, [29](#)
  - `__init__`
    - `systems.database.Database`, [14](#)
    - `systems.game.Game`, [17](#)
    - `systems.gameobject.GameObject`, [22](#)
    - `systems.inventory.Inventory`, [24](#)
    - `systems.item.InventoryItem`, [28](#)
    - `systems.item.Item`, [32](#)
    - `systems.item.PlaceholderInventoryItem`, [39](#)
    - `systems.person.Person`, [35](#)
    - `systems.player.Player`, [42](#)
    - `systems.quests.Quest`, [45](#)
    - `systems.quests.QuestLog`, [49](#)
    - `systems.quests.QuestStage`, [52](#)
    - `systems.recipe.Recipe`, [55](#)
    - `systems.recipe.RecipeBook`, [58](#)
  - `__isub__`
    - `systems.item.InventoryItem`, [29](#)
  - `__ne__`
    - `systems.gameobject.GameObject`, [22](#)
  - `__repr__`
    - `systems.gameobject.GameObject`, [22](#)
  - `__str__`
    - `systems.item.Item`, [33](#)
    - `systems.person.Person`, [36](#)
    - `systems.quests.QuestStage`, [53](#)
- `add_item`
  - `systems.inventory.Inventory`, [24](#)
- `add_quest`
  - `systems.quests.QuestLog`, [49](#)
- `add_recipe`
  - `systems.recipe.RecipeBook`, [58](#)
- `affinity`
  - `systems.person.Person`, [37](#)
- `base_affinity`
  - `systems.person.Person`, [37](#)
- `complete`
  - `systems.quests.Quest`, [47](#)
  - `systems.quests.QuestStage`, [53](#)
- `count`
  - `systems.item.InventoryItem`, [31](#)
- `craft`
  - `systems.player.Player`, [42](#)
  - `systems.recipe.RecipeBook`, [59](#)
- `current_item`
  - `systems.inventory.Inventory`, [26](#)
- `current_recipe`
  - `systems.recipe.RecipeBook`, [60](#)
- `current_stage`
  - `systems.quests.Quest`, [47](#)
- `db`
  - `systems.game.Game`, [20](#)
- `decrease_affinity`
  - `systems.person.Person`, [36](#)
- `description`
  - `systems.gameobject.GameObject`, [23](#)
- `final_stage`
  - `systems.quests.Quest`, [47](#)
- `first_stage`
  - `systems.quests.Quest`, [47](#)
- `from_dict`
  - `systems.item.InventoryItem`, [30](#)
  - `systems.item.Item`, [33](#)
  - `systems.quests.Quest`, [46](#)
  - `systems.quests.QuestStage`, [53](#)
  - `systems.recipe.Recipe`, [56](#)
- `from_item`
  - `systems.item.InventoryItem`, [30](#)
- `game_event`
  - `systems.game.Game`, [17](#)
- `get_active`
  - `systems.quests.QuestLog`, [50](#)
- `get_affinity`
  - `systems.person.Person`, [37](#)
- `get_craftable`
  - `systems.player.Player`, [42](#)
  - `systems.recipe.RecipeBook`, [59](#)
- `get_current_item`
  - `systems.inventory.Inventory`, [25](#)
- `get_item`
  - `systems.database.Database`, [14](#)
- `get_item_count`
  - `systems.inventory.Inventory`, [25](#)

- get\_items
  - systems.inventory.Inventory, 25
- get\_quest
  - systems.database.Database, 14
  - systems.requests.QuestLog, 50
- get\_recipe
  - systems.database.Database, 15
- get\_stage
  - systems.requests.Quest, 46
- get\_uncraftable
  - systems.player.Player, 42
  - systems.recipe.RecipeBook, 59
- id
  - systems.gameobject.GameObject, 23
- image\_path
  - systems.item.Item, 34
- increase\_affinity
  - systems.person.Person, 37
- inventory
  - systems.player.Player, 43
- is\_craftable
  - systems.recipe.Recipe, 56
- item
  - systems.recipe.Recipe, 56
- items
  - systems.database.Database, 15
  - systems.inventory.Inventory, 26
- load\_items\_from\_file
  - systems.database.Database, 15
- load\_recipes\_from\_file
  - systems.database.Database, 15
- location
  - systems.game.Game, 20
  - systems.person.Person, 38
- name
  - systems.gameobject.GameObject, 23
- next\_stage
  - systems.requests.Quest, 46
- people
  - systems.game.Game, 20
- player
  - systems.game.Game, 20
- questlog
  - systems.player.Player, 43
- quests
  - systems.requests.QuestLog, 50
- recipebook
  - systems.player.Player, 43
- recipes
  - systems.database.Database, 15
  - systems.recipe.RecipeBook, 60
- recipes\_fp
  - systems.database.Database, 16
- remove\_item
  - systems.inventory.Inventory, 25
- remove\_recipe
  - systems.recipe.RecipeBook, 59
- requirements
  - systems.requests.QuestStage, 53
  - systems.recipe.Recipe, 57
- set\_affinity
  - systems.person.Person, 37
- set\_count
  - systems.item.InventoryItem, 30
- set\_current\_item
  - systems.inventory.Inventory, 26
- setup
  - systems.game.Game, 17
- stages
  - systems.requests.Quest, 47
- start
  - systems.requests.Quest, 46
- start\_quest
  - systems.game.Game, 18
  - systems.player.Player, 42
- started
  - systems.requests.Quest, 48
- substages
  - systems.requests.QuestStage, 53
- systems, 11
- systems.database, 11
- systems.database.Database, 13
  - \_\_init\_\_, 14
  - get\_item, 14
  - get\_quest, 14
  - get\_recipe, 15
  - items, 15
  - load\_items\_from\_file, 15
  - load\_recipes\_from\_file, 15
  - recipes, 15
  - recipes\_fp, 16
- systems.game, 11
- systems.game.Game, 16
  - \_\_init\_\_, 17
  - db, 20
  - game\_event, 17
  - location, 20
  - people, 20
  - player, 20
  - setup, 17
  - start\_quest, 18
  - update, 18
  - update\_requests, 19
- systems.gameobject, 11
- systems.gameobject.GameObject, 21
  - \_\_eq\_\_, 22
  - \_\_hash\_\_, 22
  - \_\_init\_\_, 22
  - \_\_ne\_\_, 22
  - \_\_repr\_\_, 22
  - description, 23
  - id, 23



- name, 23
- systems.inventory, 12
- systems.inventory.Inventory, 23
  - \_\_contains\_\_, 24
  - \_\_getitem\_\_, 24
  - \_\_init\_\_, 24
  - add\_item, 24
  - current\_item, 26
  - get\_current\_item, 25
  - get\_item\_count, 25
  - get\_items, 25
  - items, 26
  - remove\_item, 25
  - set\_current\_item, 26
- systems.item, 12
- systems.item.InventoryItem, 27
  - \_\_iadd\_\_, 29
  - \_\_init\_\_, 28
  - \_\_isub\_\_, 29
  - count, 31
  - from\_dict, 30
  - from\_item, 30
  - set\_count, 30
- systems.item.Item, 31
  - \_\_init\_\_, 32
  - \_\_str\_\_, 33
  - from\_dict, 33
  - image\_path, 34
- systems.item.PlaceholderInventoryItem, 38
  - \_\_init\_\_, 39
- systems.person, 12
- systems.person.Person, 34
  - \_\_init\_\_, 35
  - \_\_str\_\_, 36
  - affinity, 37
  - base\_affinity, 37
  - decrease\_affinity, 36
  - get\_affinity, 37
  - increase\_affinity, 37
  - location, 38
  - set\_affinity, 37
- systems.player, 12
- systems.player.Player, 41
  - \_\_init\_\_, 42
  - craft, 42
  - get\_craftable, 42
  - get\_uncraftable, 42
  - inventory, 43
  - questlog, 43
  - recipebook, 43
  - start\_quest, 42
- systems.quests, 12
- systems.quests.Quest, 44
  - \_\_init\_\_, 45
  - complete, 47
  - current\_stage, 47
  - final\_stage, 47
  - first\_stage, 47
  - from\_dict, 46
  - get\_stage, 46
  - next\_stage, 46
  - stages, 47
  - start, 46
  - started, 48
- systems.quests.QuestLog, 48
  - \_\_getitem\_\_, 49
  - \_\_init\_\_, 49
  - add\_quest, 49
  - get\_active, 50
  - get\_quest, 50
  - quests, 50
- systems.quests.QuestStage, 51
  - \_\_init\_\_, 52
  - \_\_str\_\_, 53
  - complete, 53
  - from\_dict, 53
  - requirements, 53
  - substages, 53
- systems.recipe, 12
- systems.recipe.Recipe, 54
  - \_\_eq\_\_, 56
  - \_\_hash\_\_, 56
  - \_\_init\_\_, 55
  - from\_dict, 56
  - is\_craftable, 56
  - item, 56
  - requirements, 57
- systems.recipe.RecipeBook, 57
  - \_\_init\_\_, 58
  - add\_recipe, 58
  - craft, 59
  - current\_recipe, 60
  - get\_craftable, 59
  - get\_uncraftable, 59
  - recipes, 60
  - remove\_recipe, 59
- systems/\_\_init\_\_.py, 61
- systems/database.py, 61
- systems/game.py, 62
- systems/gameobject.py, 63
- systems/inventory.py, 64
- systems/item.py, 65
- systems/person.py, 66, 67
- systems/player.py, 67
- systems/quests.py, 68
- systems/recipe.py, 69
- systems/requirements\_syntax.md, 70
- update
  - systems.game.Game, 18
- update\_quests
  - systems.game.Game, 19