# Detritus 2 - Systems

Generated by Doxygen 1.9.4

# Chapter 1

# requirements_syntax

# Chapter 2

# Namespace Index

## 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

# Chapter 3

# Hierarchical Index

## 3.1 Class Hierarchy

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

# Chapter 4

# Class Index

## 4.1 Class List

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

# Chapter 5

# File Index

## 5.1 File List

Here is a list of all files with brief descriptions:

# 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

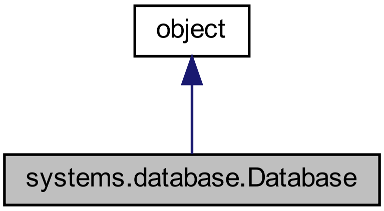
- class Recipe
- class RecipeBook

# Chapter 7

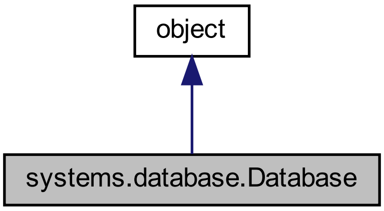
# Class Documentation

## 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 = '' )
```

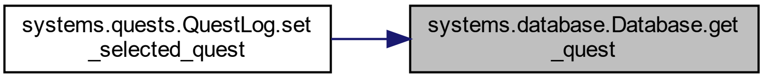
Definition at line 43 of file database.py.

#### 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
```

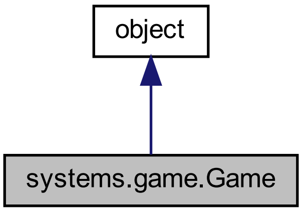
Definition at line 12 of file database.py.

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

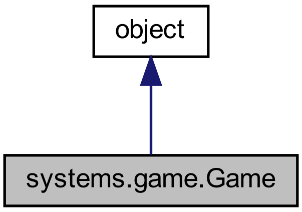
- systems/database.py

## 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

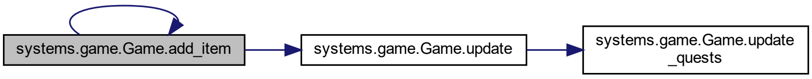
### 7.2.3.1 add_item()

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

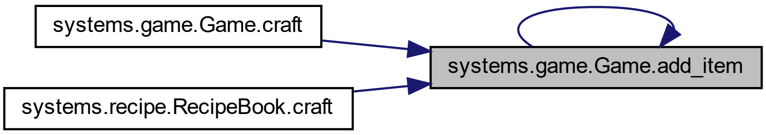
```
Adds an item to the Player’s inventory.

        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:



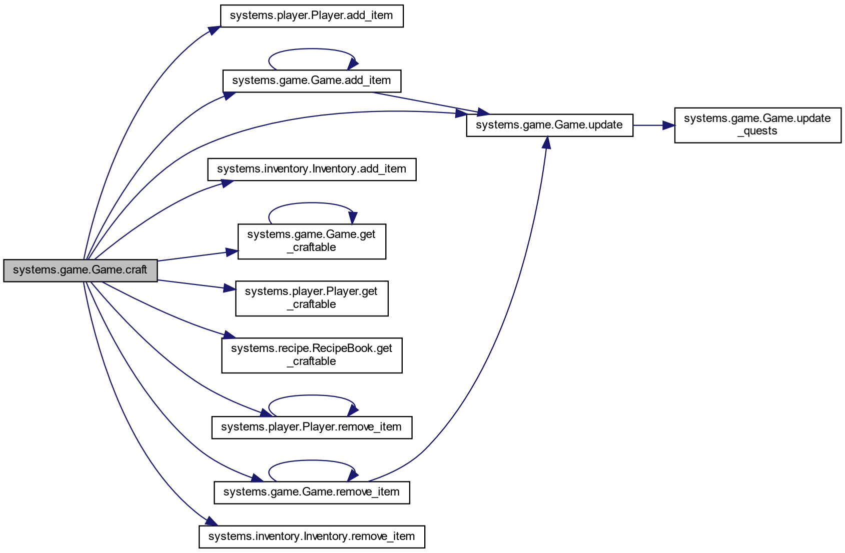
### 7.2.3.2 craft()

```
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:


systems.player.Player.add_item
systems.game.Game.add_item
systems.game.Game.update
systems.game.Game.update_quests
systems.inventory.Inventory.add_item
systems.game.Game.get_craftable
systems.game.Game.craft
systems.player.Player.get_craftable
systems.recipe.RecipeBook.get_craftable
systems.player.Player.remove_item
systems.game.Game.remove_item
systems.inventory.Inventory.remove_item

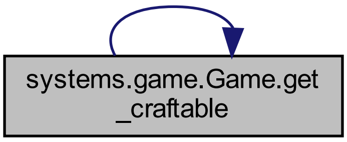
#### 7.2.3.3 get_craftable()

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

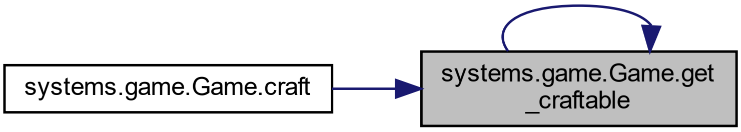
```
Returns a list of craftable recipes.
```

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.

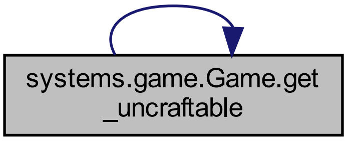
#### 7.2.3.5 get_uncraftable()

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

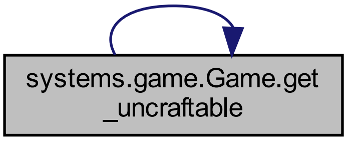
```
Returns a list of uncraftable recipes.
```

Definition at line 193 of file game.py.

Here is the call graph for this function:



Here is the caller graph for this function:



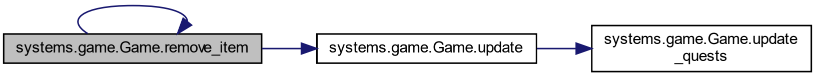
### 7.2.3.6 remove_item()

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

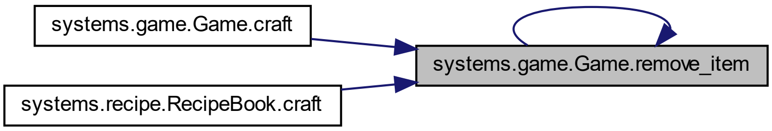
```
        Removes an Item from the Player's Inventory.

        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:


systems.game.Game.craft
systems.recipe.RecipeBook.craft
systems.game.Game.remove_item

#### 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.

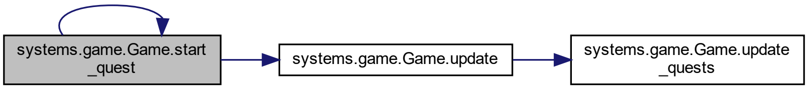
#### 7.2.3.9 start_quest()

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

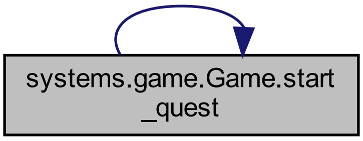
```
        Starts a Quest, the updates the game object's state.

        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:


systems.game.Game.start_quest
systems.game.Game.update
systems.game.Game.update_quests

Here is the caller graph for this function:



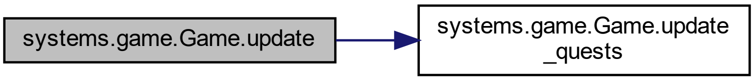
### 7.2.3.10 update()

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

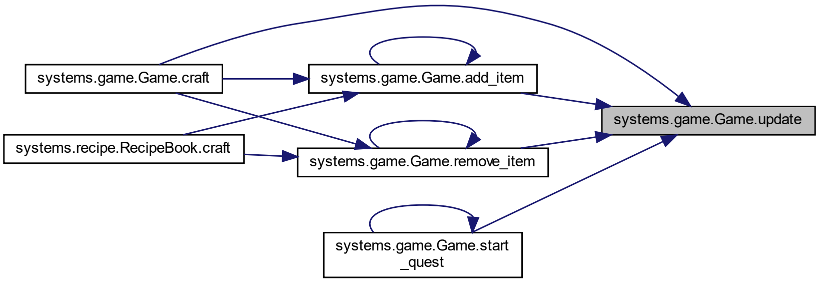
```
        Updates the Game object's state
```

Definition at line 78 of file game.py.

Here is the call graph for this function:



Here is the caller graph for this function:


systems.game.Game.craft
systems.game.Game.add_item
systems.game.Game.update
systems.recipe.RecipeBook.craft
systems.game.Game.remove_item
systems.game.Game.start
_quest

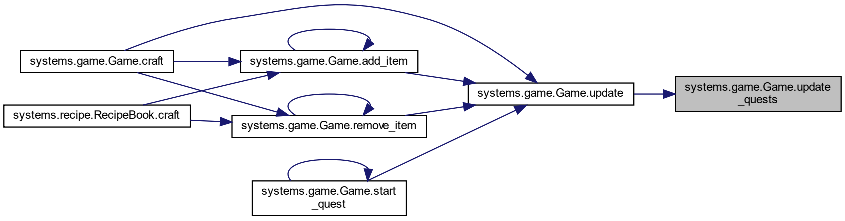
#### 7.2.3.11 update_quests()

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

```
        Updates all active quests.

        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
```

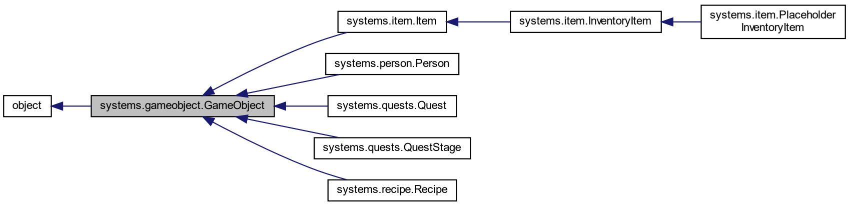
Definition at line 9 of file game.py.

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

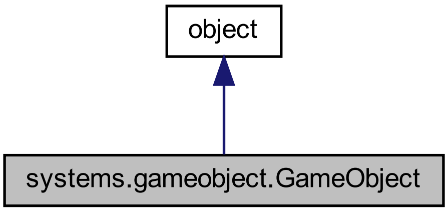
- systems/game.py

## 7.3 systems.gameobject.GameObject Class Reference

Inheritance diagram for systems.gameobject.GameObject:


systems.item.Item
systems.item.InventoryItem
systems.item.Placeholder InventoryItem
systems.person.Person
object
systems.gameobject.GameObject
systems.quests.Quest
systems.quests.QuestStage
systems.recipe.Recipe

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.

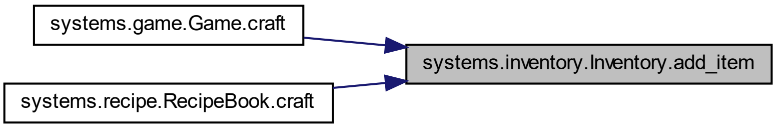
#### 7.4.3.3 add_item()

```
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:


systems.game.Game.craft
systems.recipe.RecipeBook.craft
systems.inventory.Inventory.add_item

#### 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

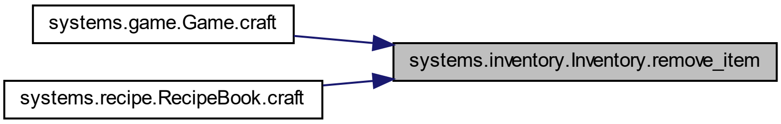
Definition at line 86 of file inventory.py.

#### 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:


systems.game.Game.craft
systems.recipe.RecipeBook.craft
systems.inventory.Inventory.remove_item

#### 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
```

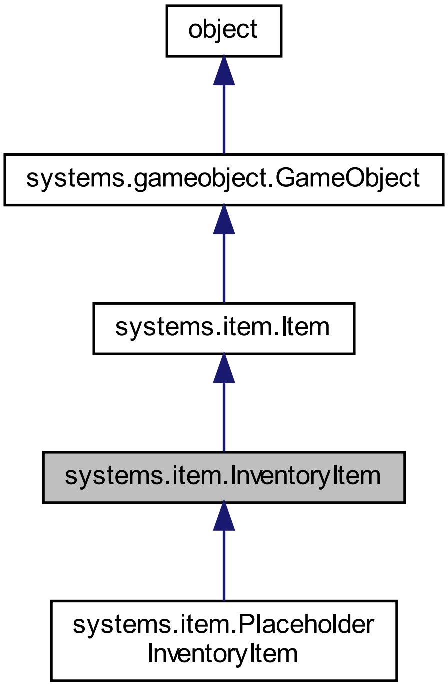
Definition at line 11 of file inventory.py.

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

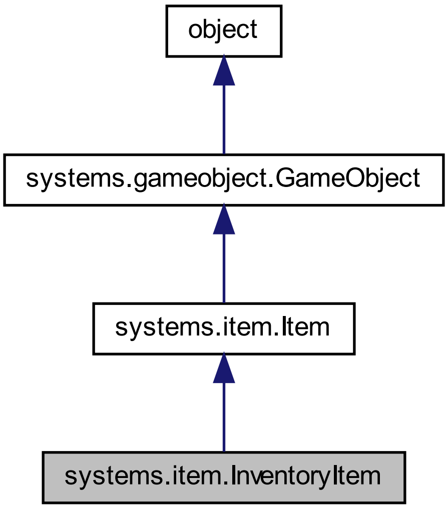
- systems/inventory.py

## 7.5 systems.item.InventoryItem Class Reference

Inheritance diagram for systems.item.InventoryItem:


object
systems.gameobject.GameObject
systems.item.Item
systems.item.InventoryItem
systems.item.Placeholder 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__()

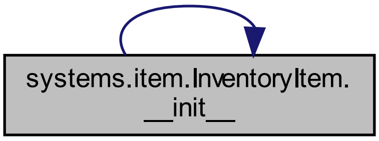
```
def systems.item.InventoryItem.__init__ (
              self,
              id,
              name,
              description = '',
              image_path = '',
              count = 0 )
```

Reimplemented from systems.item.Item.

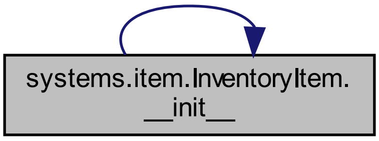
Reimplemented in systems.item.PlaceholderInventoryItem.

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.

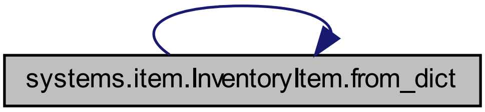
#### 7.5.3.3 from_dict()

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

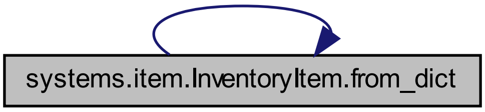
Reimplemented from systems.item.Item.

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
```

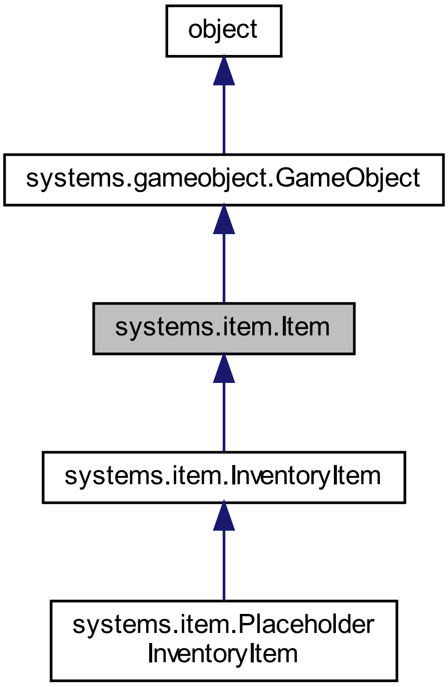
Definition at line 28 of file item.py.

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

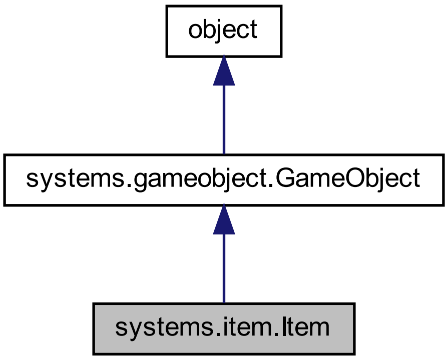
- systems/item.py

## 7.6 systems.item.Item Class Reference

Inheritance diagram for systems.item.Item:


object
systems.gameobject.GameObject
systems.item.Item
systems.item.InventoryItem
systems.item.Placeholder InventoryItem

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__()

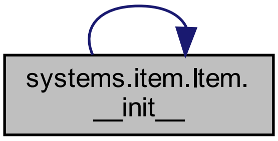
```
def systems.item.Item.__init__ (
              self,
              id,
              name,
              description = '',
              image_path = '',
              icon_path = "" )
```

Reimplemented from systems.gameobject.GameObject.

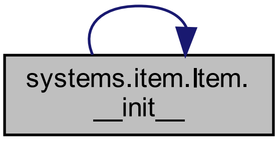
Reimplemented in systems.item.PlaceholderInventoryItem, and systems.item.InventoryItem.

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
```

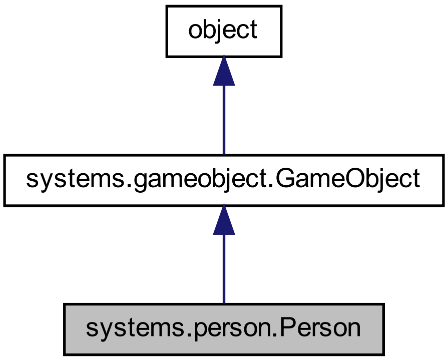
Definition at line 6 of file item.py.

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

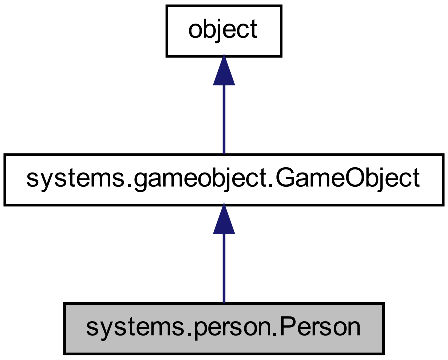
- systems/item.py

## 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

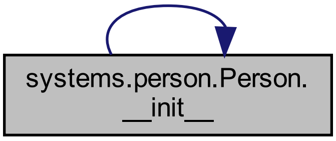
#### 7.7.2.1 __init__()

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

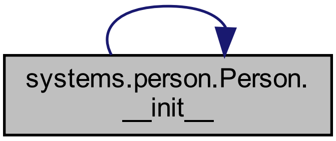
Reimplemented from systems.gameobject.GameObject.

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
```

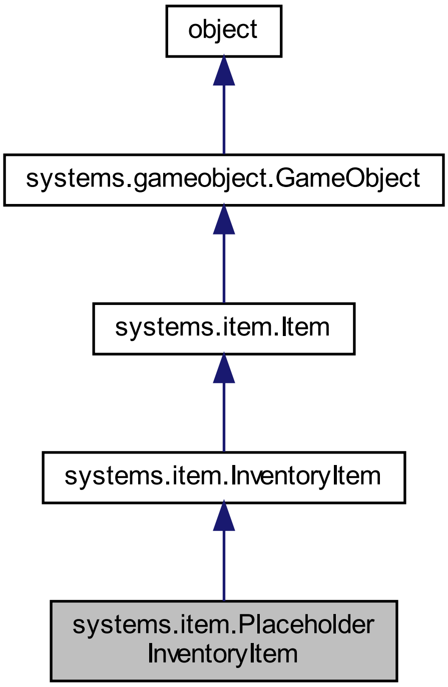
Definition at line 8 of file person.py.

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

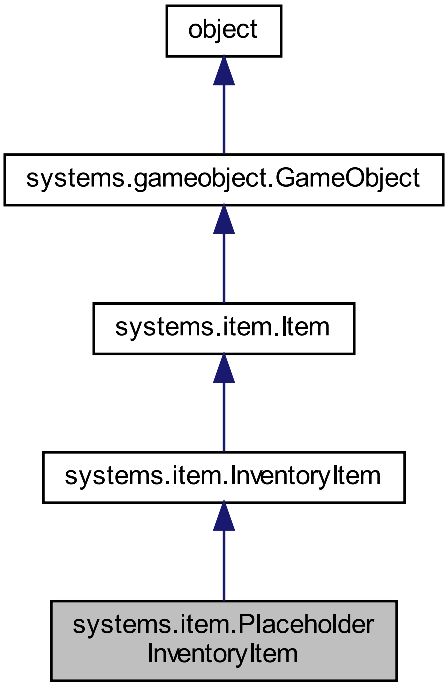
- systems/person.py

## 7.8 systems.item.PlaceholderInventoryItem Class Reference

Inheritance diagram for systems.item.PlaceholderInventoryItem:


object
systems.gameobject.GameObject
systems.item.Item
systems.item.InventoryItem
systems.item.Placeholder InventoryItem

Collaboration diagram for systems.item.PlaceholderInventoryItem:


object
systems.gameobject.GameObject
systems.item.Item
systems.item.InventoryItem
systems.item.Placeholder InventoryItem

## 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

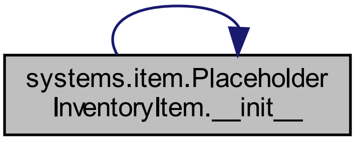
### 7.8.2.1 __init__()

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

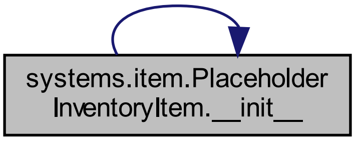
Reimplemented from systems.item.InventoryItem.

Definition at line 73 of file item.py.

Here is the call graph for this function:



Here is the caller graph for this function:

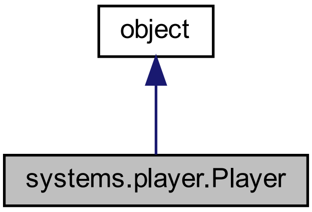


The documentation for this class was generated from the following file:

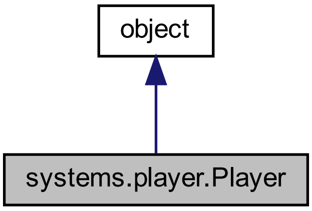
- systems/item.py

## 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.

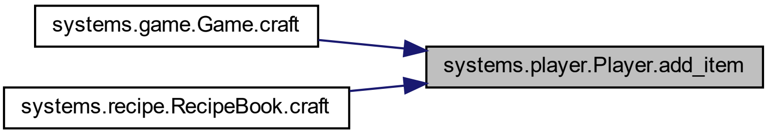
## 7.9.3 Member Function Documentation

### 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:

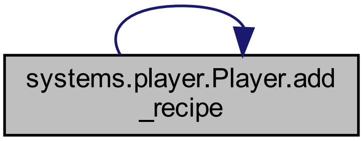

systems.game.Game.craft
systems.recipe.RecipeBook.craft
systems.player.Player.add_item

### 7.9.3.2 add_recipe()

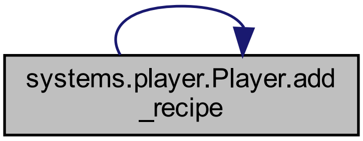
```
def systems.player.Player.add_recipe (
              self,
              recipe )
```

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 )
```

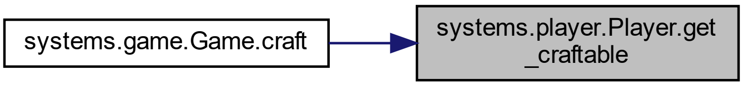
Definition at line 19 of file player.py.

#### 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 )
```

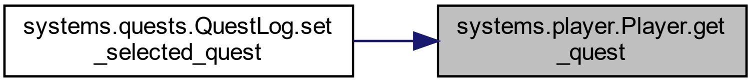
Definition at line 42 of file player.py.

#### 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 )
```

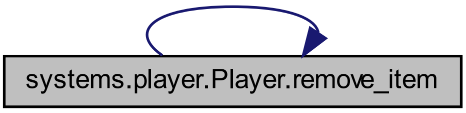
Definition at line 16 of file player.py.

### 7.9.3.9 remove_item()

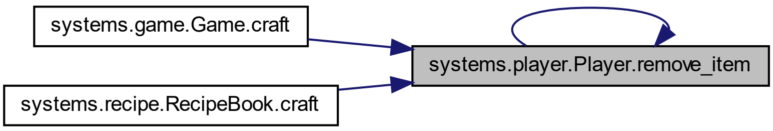
```
def systems.player.Player.remove_item (
              self,
              item,
              count )
```

Definition at line 24 of file player.py.

Here is the call graph for this function:



Here is the caller graph for this function:


systems.game.Game.craft
systems.recipe.RecipeBook.craft
systems.player.Player.remove_item

#### 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
```

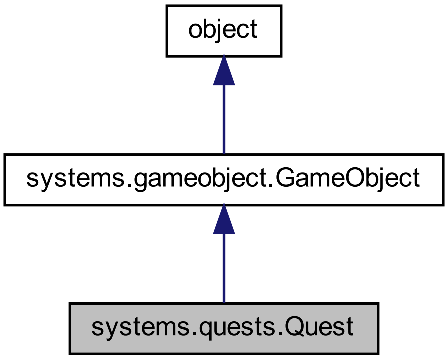
Definition at line 11 of file player.py.

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

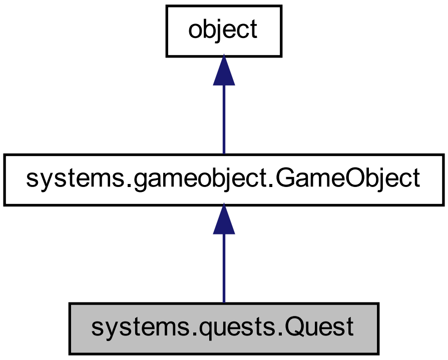
- systems/player.py

## 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

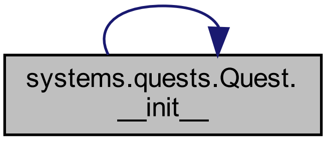
### 7.10.2.1 __init__()

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

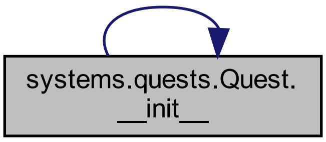
Reimplemented from systems.gameobject.GameObject.

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]
```

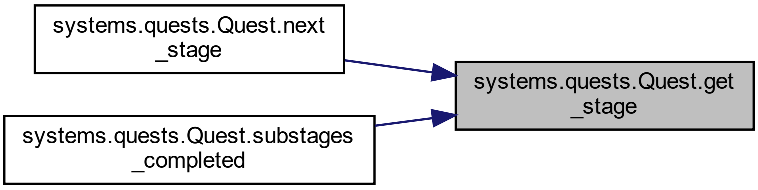
Definition at line 61 of file quests.py.

#### 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 )
```

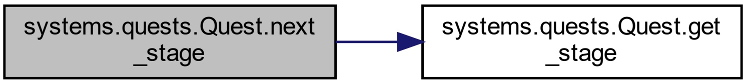
Definition at line 57 of file quests.py.

#### 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:


systems.quests.Quest.next_stage
systems.quests.Quest.get_stage

#### 7.10.3.5 start()

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

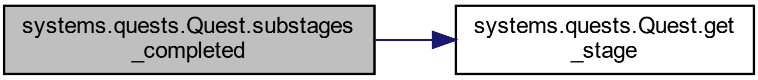
Definition at line 27 of file quests.py.

#### 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:


systems.quests.Quest.substages_completed
systems.quests.Quest.get_stage

## 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
```

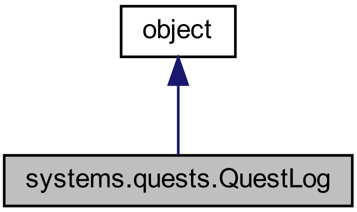
Definition at line 18 of file quests.py.

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

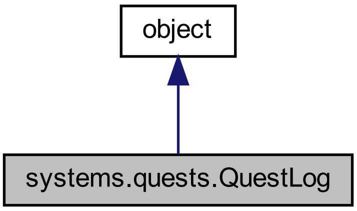
- systems/quests.py

## 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 )
```

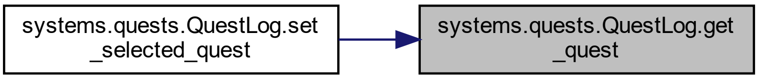
Definition at line 117 of file quests.py.

#### 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:

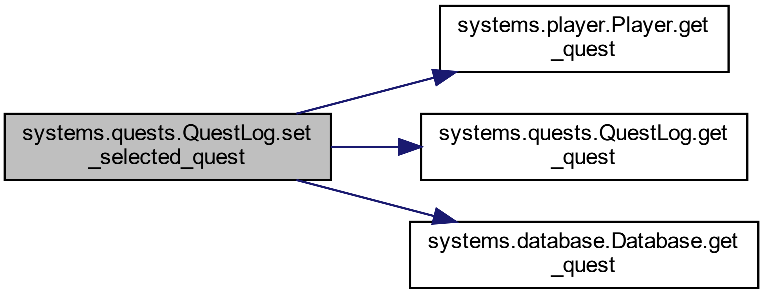


#### 7.11.3.5 set_selected_quest()

```
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:


systems.player.Player.get
_quest
systems.quests.QuestLog.set
_selected_quest
systems.quests.QuestLog.get
_quest
systems.database.Database.get
_quest

### 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
```

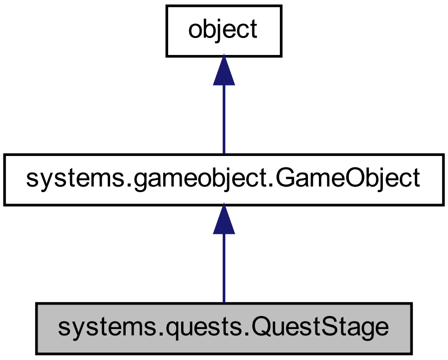
Definition at line 101 of file quests.py.

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

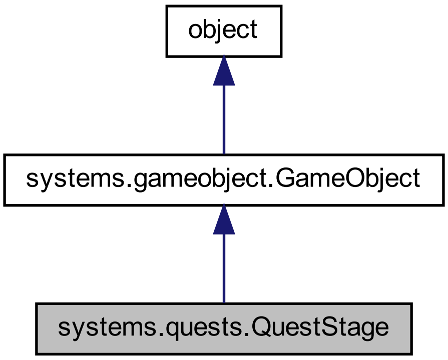
- systems/quests.py

## 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

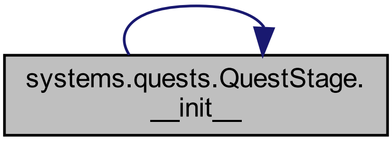
#### 7.12.2.1 __init__()

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

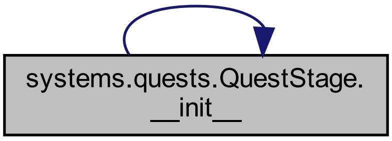
Reimplemented from systems.gameobject.GameObject.

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
```

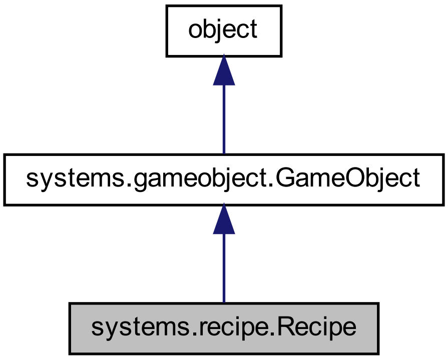
Definition at line 86 of file quests.py.

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

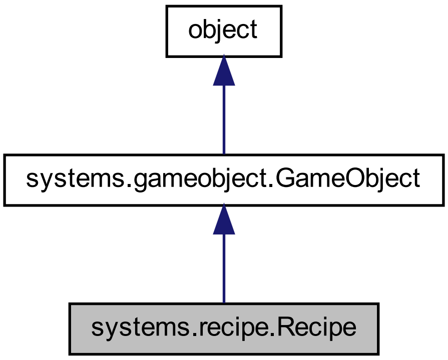
- systems/quests.py

## 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

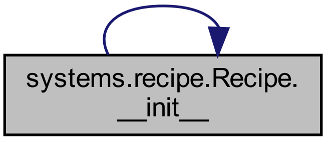
### 7.13.2.1 __init__()

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

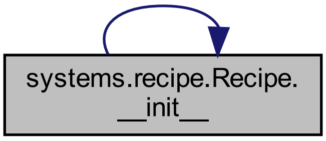
Reimplemented from systems.gameobject.GameObject.

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
```

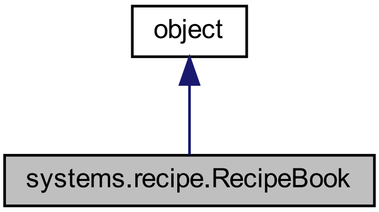
Definition at line 6 of file recipe.py.

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

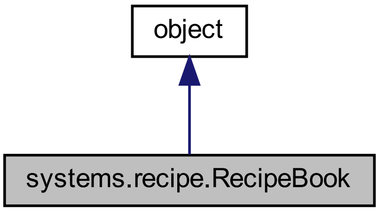
- systems/recipe.py

## 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 )
```

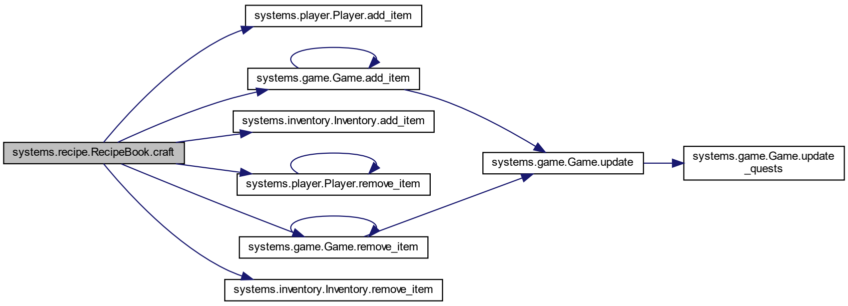
Definition at line 33 of file recipe.py.

#### 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:

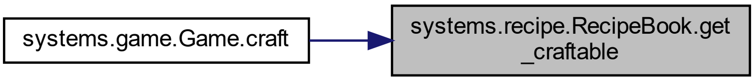

systems.player.Player.add_item
systems.game.Game.add_item
systems.inventory.Inventory.add_item
systems.recipe.RecipeBook.craft
systems.game.Game.update
systems.game.Game.update_quests
systems.player.Player.remove_item
systems.game.Game.remove_item
systems.inventory.Inventory.remove_item

#### 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.

```
import json
import os.path as path
from .item import Item
from .recipe import Recipe, RecipeBook
class Database(object):
    def __init__(
        self,
        directory=’data’,
        item_fp=’items.json’,
        recipe_fp=’recipes.json’
    ):
        self.directory, self.items_fp, self.recipes_fp = directory, item_fp, recipe_fp
        # self.item_list = self.load_items_from_file()
        self.items = {}
        self.recipes = {}
        self.quests = {}
        self.people = {}
        self.locations = {}
        # self.recipe_list = self.load_recipes_from_file()


    def verify(self):
        all_ids = [i.id for i in self.items.values() + self.recipes.values() + self.quests.values() + self.people.values() + self.locations.values()]
        dupes = [(i, all_ids.count(1)) for i in all_ids if all_ids.count(i) > 1]
        if len(dupes) > 0:
            raise ValueError("\n".join(["ID: {} occurs {} times in Database".format(i[0], i[1]) for i in dupes]))


    def load_items_from_file(self, file):
        data = json.load(file)
        items = {}
        for i in data:
            items[i[’id’] if ’id’ in i else i[’name’].lower()] = Item.from_dict(i)
        self.items = items

    def load_recipes_from_file(self, file):
        data = json.load(file)
        recipes = {}
        for r in data:
            recipes[r[’id’] if ’id’ in r else r[’name’].lower()] = Recipe.from_dict(r)
        self.recipes = recipes

    def get_item(self, item_id=”):
        return self.item_list[item_id]

    def get_recipe(self, recipe_id=”):
        return self.recipes[recipe_id]

    def get_quest(self, quest_id):
        pass


    def set_items(self, items):
        for i in items:
            self.items[i.id] = i

    def __getitem__(self, key):
        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.

```
from .database import Database
from .player import Player
from .quests import Quest
from .recipe import Recipe
class Game(object):
    def __init__(self):
        self.people = {}
        self.db  = Database()
        self.player = Player()
    # characters = {
```

```
    #   # list of character and info
    # }
    def 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
        """
        if from_db:
            # Todo: make this actually work. Issue is, opening Files in RENPY is annoying.
            return
            self.db.load_items_from_file(fps[’items’])
            self.db.load_recipes_from_file(fps[’recipes’])
        else:
            if people:
                self.people = people
            if locations:
                self.location = locations
            if items:
                self.db.set_items(items)
            if recipes:
                for i in recipes:
                    self.db.recipes[i.id] = i
            if quests:
                for i in quests:
                    self.db.quests[i.id] = i

    def 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.
        """
        if not (isinstance(quest_id, str) or type(quest_id) != None):
            raise TypeError("quest_id is not a <str>, is type {}".format(type(quest_id)))
        if not (isinstance(quest_object, Quest) or type(quest_object) != None):
            raise TypeError("quest_object is not a <Quest>")
        if quest_id and not quest_object:
            self.player.start_quest(quest_id, self.db.get_quest(quest_id))
        elif quest_id and quest_object:
            self.player.start_quest(quest_id, quest_object)
        elif quest_object:
            self.player.start_quest(quest_object.id, quest_object)
        else:
            raise ValueError()
        self.update()




    def update(self):
        """
        Updates the Game object’s state
        """
        self.update_quests()
        pass

    def update_quests(self):
        """
        Updates all active quests.

        More to be implemented soon.
        """
        for q in self.player.questlog.get_active(): # Get all active quests
            stage = q.get_stage() # Get all of q’s stages
            stage_complete = True # intermediate variable

            # TODO: implement stage check logic
            for substage in stage.substages:
                r = substage.requirements
                complete = True
                if "item" in r:
                    # print(reqs['item'])
                    items_present = [(i in self.player.inventory) for i in r['item']['ids']]
                    if r['item']['all']:
                        if not all(items_present):
                            complete = False
                    else:
                        if not any(items_present):
                            complete = False
                if complete:
                    substage.complete = True

            reqs = stage.requirements
            if "substage" in reqs:
                completed_stages = [s.complete for s in stage.substages]
                if reqs['substage'] == 'all':
                    if not all(completed_stages):
                        stage_complete = False
                if reqs['substage'] == 'any':
                    if not any(completed_stages):
                        stage_complete = False
            if "item" in reqs:
                # print(reqs['item'])
                items_present = [(i in self.player.inventory) for i in reqs['item']['ids']]
                if reqs['item']['all']:
                    if not all(items_present):
                        stage_complete = False
                else:
                    if not any(items_present):
                        stage_complete = False
            if stage_complete:
                stage.complete = True
                q.next_stage()

    def set_selected_quest(self, quest_id):
        """Returns Quest with given ID"""
        self.player.questlog.set_selected_quest(quest_id)

    def get_selected_quest(self):
        """Returns the currently selected Quest"""
        return self.player.questlog.selected_quest

    def 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)
        """
        self.player.add_item(item, count)
        self.update()
    def 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)
        """
        self.player.remove_item(item, count)
        self.update()

    def craft(self, recipe):
        """
        Crafts Recipe.

        Checks player's inventory for required Item(s), removes them from the player's inventory, adds the crafted Item(s), then updates the Game object.

        Arguments:
        recipe -- id or Recipe object to craft
        """
        if isinstance(recipe, str):
            if recipe in [r.id for r in self.get_craftable()]: # Check if ID is in craftable recipe list
                for req in recipe.requirements:
                    self.remove_item(req[0], req[1])
                for i in recipe.items:
                    self.add_item(self.db[i[0]], i[1])
        elif isinstance(recipe, Recipe):
            if recipe in self.get_craftable(): # Check if Recipe is in the craftable recipe list
                for req in recipe.requirements:
                    self.remove_item(req[0], req[1])
                for i in recipe.items:
                    self.add_item(self.db[i[0]], i[1])

        else:
            raise TypeError("Must be of type <str> or <Recipe>, not {}".format(type(recipe)))
        self.update()


    def get_craftable(self):
        """Returns a list of craftable recipes."""
        return self.player.get_craftable()

    def get_uncraftable(self):
        """Returns a list of uncraftable recipes."""
        return self.player.get_uncraftable()



```

## 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.

```
class GameObject(object):
    """
    Generic GameObject class.
    Inherited by all obtainable "things" in the game.
    """
    def __init__(self, id, name=”, description=”):
        self.id = id
        self.name = name
        self.description = description
    def __repr__(self):
        return "<{} id=’{}’ name=’{}’ desc=’{}’>".format(
            self.__class__.__name__,
            self.id,
            self.name,
            self.description
            )

    def __hash__(self):
        return hash((self.name, self.description))

    def __eq__(self, other):
        return (self.id,self.name, self.description, self.__class__.__name__) == (other.id,other.name, other.description, other.__class__.__name__)

    def __ne__(self, other):
        return not(self == other)

```

## 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.

```
from .item import InventoryItem, Item, PlaceholderInventoryItem
from .recipe import Recipe

class Inventory():
    """
    An Inventory.

    Organizes items. It’s basically a glorified list.
    """
    def __init__(self):
        self.items = {}
        self.current_item = InventoryItem("Nothing", "I should look at something first...", 0)

    def 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)
        """
        if not isinstance(count, int): # if type(count) != int:
            raise TypeError("{} is of type {}, not {}".format(count, type(count), int))
        if isinstance(item_id, str) or isinstance(item_id, unicode):
            if item_id in self.items:
                self.items[item_id] += count
            else:
                self.items[item_id] = PlaceholderInventoryItem(item_id, count)
        elif isinstance(item_id, Item): # elif type(item_id) == (Item):
            if item_id.id not in self.items:
                self.items[item_id.id] = InventoryItem.from_item(item_id, count)
            else:
                self.items[item_id.id] += count
        else:
            raise TypeError("{} is of type {}, not {} or {}".format(item_id, type(item_id), str, Item))

    def remove_item(self, item, count=1):
        if item in self.items:
            self.items[item] -= count

    def __getitem__(self, item_id):
        return self.items[item_id]

    def __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.count, else False
        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.
        """
        if isinstance(key, str):
            return key in self.items
        elif isinstance(key, dict):
            if key[’id’] in self.items:
                return self.items[key[’id’]].count >= key[’count’]
            else:
                return False
        elif isinstance(key, tuple): #type(key) == tuple:
            if key[0] in self.items:
                return self.items[key[0]].count >= key[1]
            else:
                return False
        elif isinstance(key, list): # type(key) == list:
            if all([isinstance(i, tuple) for i in key]):
                if all([len(i) == 2 for i in key]):
                    if all([i[0] in self.items for i in key]):
                        return all([self.items[i[0]].count >= i[1] for i in key])
                    else:
                        return False
                else:
                    raise ValueError("All tuples must be of types (<str>, <int>")
            elif len(key) == 2:
                if key[0] in self.items:
                    return self.items[key[0]].count >= key[1]
                else:
                    return False
            else:
                raise ValueError("Must be a list of tuples of types (<str>, <int>)")
        else:
            raise ValueError("Input must be of types <str>, <dict>, <tuple>, list[<str>, <int>] or list[(<str>,<int>)], not {}".format(type(key)))


    def get_items(self):
        return [i for i in self.items.values() if i.count > 0]
        # return [["{} x{}".format(i[0], i[1]), i[0]] for i in list(filter(lambda i: i[1] > 0, self.items.items()))]

    def get_item_count(self, item):
        if self.items.has_key(item):
            return self.items[item]
        return 0

    def set_current_item(self, item):
        self.current_item = item

    def get_current_item(self):
        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.

```
from .gameobject import GameObject

class Item(GameObject):
    def __init__(self, id, name, description= ”, image_path=”, icon_path = ""):
        super(Item, self).__init__(id, name, description)
        self.image_path = image_path
        self.icon_path = icon_path
    def __str__(self):
        return self.name
    @staticmethod
    def from_dict(d):
        if ’id’ in d:
            id = d[’id’]
        else:
            id = d[’name’]

        if ’image_path’ in d:
            ip = d[’image_path’]
        else:
            ip = "items/{}".format(id)
        return Item(id, d[’name’], d[’description’], image_path=ip)

class InventoryItem(Item):
    def __init__(self,id,  name, description=”, image_path=”, count = 0):
        super(InventoryItem, self).__init__(id, name, description, image_path)
        if count < 0:
            raise ValueError("Count cannot be less than 0")
        self.count = count

    def set_count(self, count):
        if not isinstance(count, int):
            raise TypeError("{} is of type {}, not {}".format(count, type(count), int))
        if count < 0:
            raise ValueError("Count cannot be less than 0")

        self.count = count
    def __iadd__(self, other):
        if not isinstance(other, int):
            raise TypeError("{} is of type {}, not {}".format(other, type(other), int))

        if other < 0:
            raise ValueError("Can’t add negative number of items".format(other, self.count))

        self.count += other
        return self

    def __isub__(self, other):
        if not isinstance(other, int):
            raise TypeError("{} is of type {}, not {}".format(other, type(other), int))

        if other < 0:
            raise ValueError("Can’t remove a negative number of items".format(other, self.count))
        if self.count - other < 0:
            raise ValueError("Removing {} to current count ({}) will result in a count less than 0".format(other, self.count))

        self.count -= other
        return self

    @staticmethod
    def from_item(item, count = 0):
        if not isinstance(item, Item):# if type(item) != Item:
            raise TypeError("{} is of type {}, not {}".format(item, type(item), Item))

        return InventoryItem(item.id, item.name, item.description, item.image_path, count)

    @staticmethod
    def from_dict(d):
        if not isinstance(d, dict):
            raise TypeError("{} is of type {}, not {}".format(d, type(d), dict))
        return InventoryItem.from_item(super(InventoryItem, InventoryItem).from_dict(d))

class PlaceholderInventoryItem(InventoryItem):
    def __init__(self,id, count=0):
        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.

```
from .gameobject import GameObject

class Person(GameObject):
    def __init__(self, id, name, description, base_affinity):
        super(Person, self).__init__(id, name, description)
        self.base_affinity = base_affinity
        self.affinity = base_affinity
        self.location = ""
    def set_affinity(self, value):
        self.affinity = value
    def increase_affinity(self, value=1):
        self.affinity += value
    def decrease_affinity(self, value=1):
        self.affinity -= value
    def get_affinity(self):
        return self.affinity
    def __str__(self):
        return self.name

```

## 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.

```
from .gameobject import GameObject
from .item import Item, InventoryItem
from .inventory import Inventory
from .quests import QuestLog
from .recipe import RecipeBook, Recipe

class Player(object):
    def __init__(self):
        self.inventory = Inventory()
        self.questlog = QuestLog()
        self.recipebook = RecipeBook()

    def get_craftable(self):
        return [r for r in self.recipebook.recipes.values() if r.requirements in self.inventory]

    def get_uncraftable(self):
        return [r for r in self.recipebook.recipes.values() if not r.requirements in self.inventory]

    def craft(self, recipe):
        pass

    def add_item(self, item, count):
        self.inventory.add_item(item, count)
    def remove_item(self, item, count):
        self.inventory.remove_item(item, count)
    def start_quest(self, quest_id, quest_object):
        self.questlog.add_quest(quest_id, quest_object)
        # self.questlog

    def add_recipe(self, recipe):
        self.recipebook.add_recipe(recipe)

    def get_quest(self):
        pass

    def set_selected_quest(self, quest_id):
        pass

    def get_selected_quest(self):
        pass

    def get_items(self):
        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.

```
from .gameobject import GameObject

class Quest(GameObject):
    def __init__(
            self,
            id,
            name,
            description,
            stages=[],
            first_stage=None,
            final_stage=None,
        ):
        super(Quest, self).__init__(id, name, description)
        self.stages = {}
        for qs in stages:
            self.stages[qs.id] = qs
        self.current_stage = 0
        self.started = False
        self.complete = False
        self.first_stage = first_stage
        self.final_stage = final_stage
        # if final_stage == None:
        #   self.final_stage = len(stages)
        # else:
        #   self.final_stage = final_stage

    def start(self, stage=None):
        self.started = True
        if stage == None:
            self.current_stage = self.first_stage
        else:
            self.current_stage = stage

    def get_stage(self, stage= None):
        if stage == None:
            return self.stages[".".join([self.id, str(self.current_stage)])]
        else:
            return self.stages[".".join([self.id, str(stage)])]
    def next_stage(self, stage = None):
        if stage:
            self.current_stage = stage
        elif self.get_stage().next_stage:
            self.current_stage = self.get_stage().next_stage
        elif self.get_stage().id == self.final_stage:
            self.complete = True
        else:
            print("Not sure what you want from me here. Moving on...")

    def substages_completed(self, All=False):
        if All:
            return all([q.complete for q in self.get_stage().substages])
        else:
            return any([q.complete for q in self.get_stage().substages])


    def get_substages(self):
        return self.substages

    @staticmethod
    def from_dict(d):
        return Quest(
            d[’id’],
            d[’name’],
            d[’description’],
            stages=[QuestStage.from_dict(s) for s in d[’stages’]],
            first_stage=d[’first_stage’],

            )

class QuestStage(GameObject):
    def __init__(
        self,
        id,
        parent_quest_id,
        name=”,
        description=”,
        next_stage=0,
        substages = [],
        requirements= {}
    ):
        super(QuestStage, self).__init__(".".join([parent_quest_id, str(id)]),name, description)
        self.complete = False
        for ss in substages:
            pass
        self.substages = substages
        self.requirements = requirements
        self.next_stage = next_stage

    def __str__(self):
        return self.name


    @staticmethod
    def from_dict(d):
        return QuestStage(name=d[’name’], description=d[’description’])

class QuestLog(object):
    def __init__(self):
        self.quests = {}
        self.selected_quest = Quest("placeholder", "Quests", "Things to do!",
 stages=[QuestStage("0","")])

    def set_selected_quest(self, id):
        self.selected_quest = self.get_quest(id)


    def get_quest(self, id):
        return self.quests[id]

    def add_quest(self, id, quest):
        self.quests[id] = quest
        self.quests[id].start()
        if self.selected_quest.id == "placeholder":
            self.selected_quest = quest


    def get_active(self):
        return [q for q in self.quests.values() if q.started and not q.complete]
        pass
        # qs = [q.get_stage() for q in self.quests.values() if (q.started and not q.complete)]
        # sq = [q.substages for q in qs]
        # # python is objectively a stupid language. This is just to flatten a list. Fuck’s sake.
        # return qs + [s for sublist in sq for s in sublist]
    def __getitem__(self, key):
        return self.quests[key]


```

## 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.

```
from .gameobject import GameObject

class Recipe(GameObject):
    def __init__(self, id, name, requirements, items, description=”):
        super(Recipe, self).__init__(id, name, description)
        self.requirements = requirements
        self.items = items
    def is_craftable(self, item_list):

        pass
        # return all([item_list[i] >= c for i, c in self.requirements.items()])

    @staticmethod
    def from_dict(d):
        return Recipe(
            d[’name’],
            item=d[’item’],
            requirements=d[’requirements’],
            description=d[’description’],
        )

    def __hash__(self):
        return hash((self.name, self.item))

    def __eq__(self, other):
        return (self.name, self.requirements, self.item) == (other.name, other.requirements,
 other.item)

class RecipeBook(object):
    def __init__(self):
        self.recipes = {}
        self.current_recipe = Recipe("", "Nothing here", [("nothing", 0)], [(None, 0)])

    def add_recipe(self, recipe):
        self.recipes[recipe.id] = recipe
    def remove_recipe(self, recipe):
        self.recipes.remove(recipe)

    def get_recipe(recipe_id):
        return self.recipes[recipe_id]
    def get_craftable(self):
        return [i for i in self.recipes if i.is_craftable(self.items)]

    def get_uncraftable(self):
        return [i for i in self.recipes if not i.is_craftable(self.items)]

    def craft(self, recipe):
        if recipe in self.recipes:
            if recipe.is_craftable(self.items):
                self.add_item(recipe.item)
                for i, c in recipe.requirements.items():
                    self.remove_item(i, c)
```

## 8.21 systems/requirements_syntax.md File Reference

# Index