

Help on module item_model:

NAME

item_model - Represent existents with Item classes, instantiated by games/stories.

FILE

c:\nickmontfort-curveship-814de6e\item_model.py

CLASSES

__builtin__.object

Item

Actor

Door

Room

Substance

Thing

SharedThing

class Actor(Item)

| Any Item that can initiate action, whether human-like or not.

|

| Features of interest:

|

| alive: True | False

| Actors can only act and react if alive. If not specified, this feature
| will always be True. Things can also have an alive feature, but it must
| be set when needed. It should probably be set on a subclass created
| for a particular Thing that can react and prevent.

|

| refuses: list of (string, when.function(world), string)

| Determines what an actor will refuse to do when commanded. The first
| string is matched against actions; the function determines whether or
| not the refusal will take place given a match; and the final string
| is a template used to generate a message explaining the refusal.

|

| Method resolution order:

|

| Actor

|

| Item

|

| __builtin__.object

|

| __init__(self, tag_and_parent, **keywords)

|

| act(self, command_map, concept)

|

| The default act method runs a script, if there is one.

|

| do_command(self, command_words, command_map, concept)

|

| Return the Action that would result from the provided command.

```
exits(self, concept)
    Return this Actor's current Room's exit dictionary.

-----
Methods inherited from Item:

__eq__(self, item)

__ne__(self, item)

__str__(self)

add_child(self, link, item, making_change=True)
    Add (or remove) a child from this Item.

blank(self)
    Erase an Item when nothing is known about it by an Actor.

get_called(self)

get_gender(self)

get_hearing(self)

get_number(self)

get_qualities(self)

get_referring(self)

get_sight(self)

get_smell(self)

get_taste(self)

get_touch(self)

noun_phrase(self, discourse=None, entire=True, extra_adjs='', length=0.0)
    Return the noun phrase representing this Item.

place(self, world)
    Returns the Room this Item is located in, according to World.

prevent(self, _, __)
    By default, items do not prevent actions Subclasses can override.

react(self, _, __)
    By default, items do nothing when reacting. Subclasses can override.
```

```

| react_to_failed(self, _, __)
|     By default, items do nothing when reacting to a failed action.
|
| remove_child(self, link, item, making_change=True)
|     Remove (or add) a child from this Item.
|
| set_called(self, string)
|
| set_gender(self, string)
|
| set_hearing(self, string)
|
| set_number(self, string)
|
| set_qualities(self, quality_list)
|
| set_referring(self, string)
|
| set_sight(self, string)
|
| set_smell(self, string)
|
| set_taste(self, string)
|
| set_touch(self, string)
|     Setter. Needed because strings must be reformatted before being set.
|
| -----
| Data descriptors inherited from Item:
|
| __dict__
|     dictionary for instance variables (if defined)
|
| __weakref__
|     list of weak references to the object (if defined)
|
| called
|
| children
|     Return the children of this Item.
|
| gender
|
| hearing
|
| number
|
| qualities
|
| referring

```

sight

smell

taste

touch

```
class Door(Item)
```

```
    An Item representing a doorway, portal, or passage between two places.
```

```
    Features of interest
```

```
    connects: list of two strings
```

```
        Each string is the tag of a Room; This Door connects the two.
```

```
    Method resolution order:
```

```
        Door
```

```
        Item
```

```
        __builtin__.object
```

```
    Methods defined here:
```

```
    __init__(self, tag, **keywords)
```

```
    -----  
    Methods inherited from Item:
```

```
    __eq__(self, item)
```

```
    __ne__(self, item)
```

```
    __str__(self)
```

```
    add_child(self, link, item, making_change=True)
```

```
        Add (or remove) a child from this Item.
```

```
    blank(self)
```

```
        Erase an Item when nothing is known about it by an Actor.
```

```
    get_called(self)
```

```
    get_gender(self)
```

```
    get_hearing(self)
```

```
    get_number(self)
```

```
    get_qualities(self)
```

`get_referring(self)`

`get_sight(self)`

`get_smell(self)`

`get_taste(self)`

`get_touch(self)`

`noun_phrase(self, discourse=None, entire=True, extra_adjs='', length=0.0)`
Return the noun phrase representing this Item.

`place(self, world)`
Returns the Room this Item is located in, according to World.

`prevent(self, _, __)`
By default, items do not prevent actions Subclasses can override.

`react(self, _, __)`
By default, items do nothing when reacting. Subclasses can override.

`react_to_failed(self, _, __)`
By default, items do nothing when reacting to a failed action.

`remove_child(self, link, item, making_change=True)`
Remove (or add) a child from this Item.

`set_called(self, string)`

`set_gender(self, string)`

`set_hearing(self, string)`

`set_number(self, string)`

`set_qualities(self, quality_list)`

`set_referring(self, string)`

`set_sight(self, string)`

`set_smell(self, string)`

`set_taste(self, string)`

`set_touch(self, string)`
Setter. Needed because strings must be reformatted before being set.

Data descriptors inherited from Item:

`__dict__`
dictionary for instance variables (if defined)

`__weakref__`
list of weak references to the object (if defined)

`called`

`children`
Return the children of this Item.

`gender`

`hearing`

`number`

`qualities`

`referring`

`sight`

`smell`

`taste`

`touch`

```
class Item(__builtin__.object)
| Abstract base class for items.
|
| Methods defined here:
|
| __eq__(self, item)
|
| __init__(self, tag_and_parent, category, **keywords)
|
| __ne__(self, item)
|
| __str__(self)
|
| add_child(self, link, item, making_change=True)
|     Add (or remove) a child from this Item.
|
| blank(self)
|     Erase an Item when nothing is known about it by an Actor.
```

`get_called(self)`

`get_gender(self)`

`get_hearing(self)`

`get_number(self)`

`get_qualities(self)`

`get_referring(self)`

`get_sight(self)`

`get_smell(self)`

`get_taste(self)`

`get_touch(self)`

`noun_phrase(self, discourse=None, entire=True, extra_adjs='', length=0.0)`
Return the noun phrase representing this Item.

`place(self, world)`
Returns the Room this Item is located in, according to World.

`prevent(self, __, __)`
By default, items do not prevent actions Subclasses can override.

`react(self, __, __)`
By default, items do nothing when reacting. Subclasses can override.

`react_to_failed(self, __, __)`
By default, items do nothing when reacting to a failed action.

`remove_child(self, link, item, making_change=True)`
Remove (or add) a child from this Item.

`set_called(self, string)`

`set_gender(self, string)`

`set_hearing(self, string)`

`set_number(self, string)`

`set_qualities(self, quality_list)`

`set_referring(self, string)`

set_sight(self, string)

set_smell(self, string)

set_taste(self, string)

set_touch(self, string)

Setter. Needed because strings must be reformatted before being set.

Data descriptors defined here:

__dict__

dictionary for instance variables (if defined)

__weakref__

list of weak references to the object (if defined)

called

children

Return the children of this Item.

gender

hearing

number

qualities

referring

sight

smell

taste

touch

class Room(Item)

An Item representing a physical location.

Features that are particular to Rooms:

exits: dictionary of string: string

The key is a direction; the value is the tag of the Door or Room in that direction.

shared: list of strings

Each string is the tag of a SharedThing; That Item is present in this room and all other rooms that list it.

view: dictionary of string: (float, string)

The key is the tag of a Room which is visible from this one; the tuple that is the value has the visibility of that room (a floating point number in (0, 1)) and a string which is used to generate a textual description of the direction of that room.

Method resolution order:

Room

Item

__builtin__.object

Methods defined here:

__init__(self, tag, **keywords)

exit(self, direction)

Return the Room or Door that lies in this direction, if there is one.

Methods inherited from Item:

__eq__(self, item)

__ne__(self, item)

__str__(self)

add_child(self, link, item, making_change=True)

Add (or remove) a child from this Item.

blank(self)

Erase an Item when nothing is known about it by an Actor.

get_called(self)

get_gender(self)

get_hearing(self)

get_number(self)

get_qualities(self)

get_referring(self)

```
get_sight(self)

get_smell(self)

get_taste(self)

get_touch(self)

noun_phrase(self, discourse=None, entire=True, extra_adjs='', length=0.0)
    Return the noun phrase representing this Item.

place(self, world)
    Returns the Room this Item is located in, according to World.

prevent(self, _, __)
    By default, items do not prevent actions Subclasses can override.

react(self, _, __)
    By default, items do nothing when reacting. Subclasses can override.

react_to_failed(self, _, __)
    By default, items do nothing when reacting to a failed action.

remove_child(self, link, item, making_change=True)
    Remove (or add) a child from this Item.

set_called(self, string)

set_gender(self, string)

set_hearing(self, string)

set_number(self, string)

set_qualities(self, quality_list)

set_referring(self, string)

set_sight(self, string)

set_smell(self, string)

set_taste(self, string)

set_touch(self, string)
    Setter. Needed because strings must be reformatted before being set.
```

```
-----
Data descriptors inherited from Item:
```

`__dict__`
 dictionary for instance variables (if defined)

`__weakref__`
 list of weak references to the object (if defined)

`called`

`children`
 Return the children of this Item.

`gender`

`hearing`

`number`

`qualities`

`referring`

`sight`

`smell`

`taste`

`touch`

```
class SharedThing(Thing)
```

A special sort of (large) Thing that appears in more than one room.

Note that SharedThing is a subclass of Thing and shares the same category: `example.thing` is True for a SharedThing; there is no 'sharedthing' category. However, all SharedThings will have an attribute "sharedthing" that is set to True. Testing `hasattr(item, 'sharedthing')` will determine if the item is a SharedThing.

SharedThing is provided to allow implementation of things like the sky, the sun, or a massive wall of the sort the United States has erected along the US/Mexico border. Because shared things are meant to represent these sorts of entities, they have an `allowed` expression that always returns False. Nothing can be placed in one, on one, through one, be part of one, or be held by one. If it were possible, for instance, to place a sticker on a massive border wall, this implementation would make the sticker visible in every room along the border, which makes no sense.

A SharedThing does **not** have a shared feature. The Rooms that it is located in have shared features which are lists containing the tags of each shared item.

Method resolution order:

SharedThing
Thing
Item
__builtin__.object

Methods defined here:

`__init__(self, tag_and_parent, **keywords)`

Methods inherited from Item:

`__eq__(self, item)`

`__ne__(self, item)`

`__str__(self)`

`add_child(self, link, item, making_change=True)`
Add (or remove) a child from this Item.

`blank(self)`
Erase an Item when nothing is known about it by an Actor.

`get_called(self)`

`get_gender(self)`

`get_hearing(self)`

`get_number(self)`

`get_qualities(self)`

`get_referring(self)`

`get_sight(self)`

`get_smell(self)`

`get_taste(self)`

`get_touch(self)`

`noun_phrase(self, discourse=None, entire=True, extra_adjs='', length=0.0)`
Return the noun phrase representing this Item.

`place(self, world)`

Returns the Room this Item is located in, according to World.

`prevent(self, _, __)`

By default, items do not prevent actions Subclasses can override.

`react(self, _, __)`

By default, items do nothing when reacting. Subclasses can override.

`react_to_failed(self, _, __)`

By default, items do nothing when reacting to a failed action.

`remove_child(self, link, item, making_change=True)`

Remove (or add) a child from this Item.

`set_called(self, string)`

`set_gender(self, string)`

`set_hearing(self, string)`

`set_number(self, string)`

`set_qualities(self, quality_list)`

`set_referring(self, string)`

`set_sight(self, string)`

`set_smell(self, string)`

`set_taste(self, string)`

`set_touch(self, string)`

Setter. Needed because strings must be reformatted before being set.

Data descriptors inherited from Item:

`__dict__`

dictionary for instance variables (if defined)

`__weakref__`

list of weak references to the object (if defined)

`called`

`children`

Return the children of this Item.

`gender`

hearing
number
qualities
referring
sight
smell
taste
touch

```
class Substance(Item)
```

```
    Includes powders and liquids; must be in a source or vessel.
```

```
    Method resolution order:
```

```
        Substance
```

```
        Item
```

```
        __builtin__.object
```

```
    Methods defined here:
```

```
    __init__(self, tag, **keywords)
```

```
-----  
    Methods inherited from Item:
```

```
    __eq__(self, item)
```

```
    __ne__(self, item)
```

```
    __str__(self)
```

```
    add_child(self, link, item, making_change=True)
```

```
        Add (or remove) a child from this Item.
```

```
    blank(self)
```

```
        Erase an Item when nothing is known about it by an Actor.
```

```
    get_called(self)
```

```
    get_gender(self)
```

```
    get_hearing(self)
```

```
get_number(self)

get_qualities(self)

get_referring(self)

get_sight(self)

get_smell(self)

get_taste(self)

get_touch(self)

noun_phrase(self, discourse=None, entire=True, extra_adjs='', length=0.0)
    Return the noun phrase representing this Item.

place(self, world)
    Returns the Room this Item is located in, according to World.

prevent(self, _, __)
    By default, items do not prevent actions Subclasses can override.

react(self, _, __)
    By default, items do nothing when reacting. Subclasses can override.

react_to_failed(self, _, __)
    By default, items do nothing when reacting to a failed action.

remove_child(self, link, item, making_change=True)
    Remove (or add) a child from this Item.

set_called(self, string)

set_gender(self, string)

set_hearing(self, string)

set_number(self, string)

set_qualities(self, quality_list)

set_referring(self, string)

set_sight(self, string)

set_smell(self, string)

set_taste(self, string)
```

```
| set_touch(self, string)
|     Setter. Needed because strings must be reformatted before being set.
|
| -----
| Data descriptors inherited from Item:
|
| __dict__
|     dictionary for instance variables (if defined)
|
| __weakref__
|     list of weak references to the object (if defined)
|
| called
|
| children
|     Return the children of this Item.
|
| gender
|
| hearing
|
| number
|
| qualities
|
| referring
|
| sight
|
| smell
|
| taste
|
| touch
```

```
class Thing(Item)
|     An item that is not a room, has no concept, and cannot act.
|
|     Method resolution order:
|         Thing
|         Item
|         __builtin__.object
|
|     Methods defined here:
|
|     __init__(self, tag_and_parent, **keywords)
|
|     -----
|     Methods inherited from Item:
```



```
| __eq__(self, item)
|
| __ne__(self, item)
|
| __str__(self)
|
| add_child(self, link, item, making_change=True)
|     Add (or remove) a child from this Item.
|
| blank(self)
|     Erase an Item when nothing is known about it by an Actor.
|
| get_called(self)
|
| get_gender(self)
|
| get_hearing(self)
|
| get_number(self)
|
| get_qualities(self)
|
| get_referring(self)
|
| get_sight(self)
|
| get_smell(self)
|
| get_taste(self)
|
| get_touch(self)
|
| noun_phrase(self, discourse=None, entire=True, extra_adjs='', length=0.0)
|     Return the noun phrase representing this Item.
|
| place(self, world)
|     Returns the Room this Item is located in, according to World.
|
| prevent(self, _, __)
|     By default, items do not prevent actions Subclasses can override.
|
| react(self, _, __)
|     By default, items do nothing when reacting. Subclasses can override.
|
| react_to_failed(self, _, __)
|     By default, items do nothing when reacting to a failed action.
|
| remove_child(self, link, item, making_change=True)
|     Remove (or add) a child from this Item.
```

```
set_called(self, string)

set_gender(self, string)

set_hearing(self, string)

set_number(self, string)

set_qualities(self, quality_list)

set_referring(self, string)

set_sight(self, string)

set_smell(self, string)

set_taste(self, string)

set_touch(self, string)
    Setter. Needed because strings must be reformatted before being set.

-----
Data descriptors inherited from Item:

__dict__
    dictionary for instance variables (if defined)

__weakref__
    list of weak references to the object (if defined)

called

children
    Return the children of this Item.

gender

hearing

number

qualities

referring

sight

smell

taste
```

|
| touch

FUNCTIONS

`check_attributes(identifier, required, impossible, attributes)`
Raise errors if required attributes are missing or impossible ones present.

`determine_called(called)`
Using the called string, determine the name triple.

`set_features(item, category, keywords)`
Sets features modeling the current state (condition) of an Item.

These data attributes represent the item's important features within the fiction/game world. The values held in particular entries are changed by Modify actions. The location and configuration of an item in the overall context of the world (what parent it has and what its link to the parent is) is not represented here. That aspect of the world is changed by Configure instead of Modify. These features model the item itself, not where it is located.

The built-in features, represented by data attributes, are:

`article: string`
The initial article to use. "a" will be adjusted to "an" if necessary when text is generated.

`called: string`
What the item can be called when text is generated.

`referring: string | None`
Additional words that can be typed to refer to the item. Optional words first, separated by spaces; then "|", then space-delimited names. If referring is the '' (the empty string, which is the default), there are no special words added, but the item can still be referred to by words derived from its category and qualities. For the special case in which there should be no way to refer to an item, set referring to None.

`qualities: list of strings`
Terms describing the item; synonyms of these are used in recognition and the terms can be used in simulation.

`gender: 'female' | 'male' | 'neuter'`
Whether the item should be referred to as a she, he, or it.

`glow: float`
How much light the item is radiating, typically in (0, 1).
Can be set outside of (0, 1) for supernatural reasons.

`number: 'singular' | 'plural'`

Whether the item should be referred to in the singular or plural.

prominence: float

How visible or noticeable an item is, typically in (0, 1).

transparent: True | False

Can one see through the item and see what is inside?

mention: True | False

Should the item ever be mentioned (for instance, in lists)? Almost everything should be, but not, for instance, part of another item that is described in the main description of the parent item.

allowed: can.function(tag, link, world)

Determines what an item can contain. Specifically, returns whether the item 'tag' can be a child of the parent, in the specified link relationship, given the situation of world.

shared: list of strings

If a Room, the tags of SharedThings that this Room has; otherwise [].

sight: string

Template that produces a description of what an agent sees when looking at an item.

touch: string

Template that produces a description of what an agent feels when touching at an item. Should be able to complete the sentence "The adventurer feels ..." or "You feel ..."

hearing: string

Similar template for hearing.

smell: string

Similar template for hearing.

taste: string

Similar template for taste.

DATA

```
__author__ = 'Nick Montfort'
__copyright__ = 'Copyright 2011 Nick Montfort'
__license__ = 'ISC'
__status__ = 'Development'
__version__ = '0.5.0.0'
```

VERSION

0.5.0.0

AUTHOR

