```
Help on module world_model:

NAME
    world_model - World and Concept classes for instantiation by interactive fictions.

FILE
    c:\nickmontfort-curveship-814de6e\world_model.py

CLASSES
    __builtin__.object
        WorldOrConcept
            Concept
            World

    class Concept(WorldOrConcept)
     |  An Actor's theory or model of the World, which can be used in telling.
     |
     |  Method resolution order:
     |      Concept
     |      WorldOrConcept
     |      __builtin__.object
     |
     |  Methods defined here:
     |
     |  __init__(self, item_list, actions, cosmos=None)
     |
     |  copy_at(self, time)
     |      Return a new Concept based on this one, but from an earlier time.
     |
     |  item_at(self, tag, time)
     |      Return the Item from this moment in the Concept.
     |
     |  roll_back_to(self, time)
     |      Go back to a previous state of this Concept.
     |
     |  update_item(self, item, time)
     |      After perception, change an Item within this Concept.
     |
     |  ----------------------------------------------------------------------
     |  Methods inherited from WorldOrConcept:
     |
     |  __str__(self)
     |
     |  accessible(self, actor)
     |      List all Items an Item can access.
     |
     |  ancestors(self, tag)
     |      List all Items hierarchically above an Item.
     |
     |  compartment_of(self, tag)
```

```
 |      Return the opaque compartment around the Item.
 |
 |  descendants(self, tag, stop='bottom')
 |      List all Items hierarchically under "tag".
 |
 |      If stop='bottom', descend all the way. If stop='closed', go to down to
 |      closed children, but not inside those; for stop='opaque', stop at
 |      opaque ones.
 |
 |  doors(self, tag)
 |      Returns a list of the Item's Doors; [] if there are none.
 |
 |  has(self, category, tag)
 |      Does the tag represent an Item of this category in this World/Concept?
 |
 |  respondents(self, action)
 |      Return a list: the cosmos, the Room of the agent, (living) contents.
 |
 |      These are all the Items that can prevent or react to an Action by
 |      the agent of the action. If the Item has an "alive" feature, it is only
 |      added if alive is True.
 |
 |      A special case: If the agent has configured itself to a new Room, the
 |      new Room and (living) contents have a chance to respond, too.
 |
 |  room_of(self, tag)
 |      If the Item exists and is in a Room, return the Room.
 |
 |  show_descendants(self, tag, padding='')
 |      Return the tree rooted at this Item.
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors inherited from WorldOrConcept:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)

class World(WorldOrConcept)
 |  The simulated world; it has Items and Actions.
 |
 |  Method resolution order:
 |      World
 |      WorldOrConcept
 |      __builtin__.object
 |
 |  Methods defined here:
 |
```

```
|  __init__(self, fiction)
|
|  advance_clock(self, duration)
|      Move the time forward a specified number of ticks.
|
|  back_up_clock(self, target_time)
|      Roll the time back to a particular tick.
|
|  can_see(self, actor, tag)
|      Is the item identified by "tag" visible to "actor"?
|
|  light_level(self, tag)
|      Determines the light level (not just glow) in the Item's compartment.
|
|  light_within(self, tag)
|      Returns the light illuminating an Item, inherently and within.
|
|  prevents_sight(self, actor, tag)
|      Returns a reason (if there are any) that "actor" cannot see "tag".
|
|  reset(self)
|      Revert the World and Concepts to their initial states.
|
|  set_concepts(self, actors)
|      Set initial information in all Actors' Concepts.
|
|  transfer(self, item, actor, time)
|      Place an appropriate version of an Item in the Actor's Concept.
|
|  transfer_out(self, item, actor, time)
|      Remove the Item from the Actor's Concept.
|
|  undo(self, action_id)
|      Revert the World back to the start time of the specified Action.
|
|  ----------------------------------------------------------------------
|  Methods inherited from WorldOrConcept:
|
|  __str__(self)
|
|  accessible(self, actor)
|      List all Items an Item can access.
|
|  ancestors(self, tag)
|      List all Items hierarchically above an Item.
|
|  compartment_of(self, tag)
|      Return the opaque compartment around the Item.
|
|  descendants(self, tag, stop='bottom')
```

```
 |      List all Items hierarchically under "tag".
 |
 |      If stop='bottom', descend all the way. If stop='closed', go to down to
 |      closed children, but not inside those; for stop='opaque', stop at
 |      opaque ones.
 |
 |  doors(self, tag)
 |      Returns a list of the Item's Doors; [] if there are none.
 |
 |  has(self, category, tag)
 |      Does the tag represent an Item of this category in this World/Concept?
 |
 |  respondents(self, action)
 |      Return a list: the cosmos, the Room of the agent, (living) contents.
 |
 |      These are all the Items that can prevent or react to an Action by
 |      the agent of the action. If the Item has an "alive" feature, it is only
 |      added if alive is True.
 |
 |      A special case: If the agent has configured itself to a new Room, the
 |      new Room and (living) contents have a chance to respond, too.
 |
 |  room_of(self, tag)
 |      If the Item exists and is in a Room, return the Room.
 |
 |  show_descendants(self, tag, padding='')
 |      Return the tree rooted at this Item.
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors inherited from WorldOrConcept:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)

class WorldOrConcept(__builtin__.object)
 |  Abstract base class for the World and for Concepts.
 |
 |  Methods defined here:
 |
 |  __init__(self, item_list, actions)
 |
 |  __str__(self)
 |
 |  accessible(self, actor)
 |      List all Items an Item can access.
 |
 |  ancestors(self, tag)
```

```
 |      List all Items hierarchically above an Item.
 |
 |  compartment_of(self, tag)
 |      Return the opaque compartment around the Item.
 |
 |  descendants(self, tag, stop='bottom')
 |      List all Items hierarchically under "tag".
 |
 |      If stop='bottom', descend all the way. If stop='closed', go to down to
 |      closed children, but not inside those; for stop='opaque', stop at
 |      opaque ones.
 |
 |  doors(self, tag)
 |      Returns a list of the Item's Doors; [] if there are none.
 |
 |  has(self, category, tag)
 |      Does the tag represent an Item of this category in this World/Concept?
 |
 |  respondents(self, action)
 |      Return a list: the cosmos, the Room of the agent, (living) contents.
 |
 |      These are all the Items that can prevent or react to an Action by
 |      the agent of the action. If the Item has an "alive" feature, it is only
 |      added if alive is True.
 |
 |      A special case: If the agent has configured itself to a new Room, the
 |      new Room and (living) contents have a chance to respond, too.
 |
 |  room_of(self, tag)
 |      If the Item exists and is in a Room, return the Room.
 |
 |  show_descendants(self, tag, padding='')
 |      Return the tree rooted at this Item.
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors defined here:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)

FUNCTIONS
    check_for_reserved_tags(items)
        Raise an error if a reserved tag, such as @cosmos, is in the list.

    sight_culprit(prominence, view, lit)
        Which of the three factors is mostly to blame for the lack of visibility?
```

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

VERSION
    0.5.0.0

AUTHOR
    Nick Montfort
```