WEATBAG Documentation

Release 1.0

Thomas Kluyver et al.

June 25, 2016

Contents

You don't just play WEATBAG, you can take part in creating it too. These docs explain how to write new tiles for the game.

The idea is that anything goes, but we do need a few house rules:

1. Don't mess with the real life user or their computer. Deleting files, accessing personal information, sending spam and executing untrusted code are all off limits.
2. The content is in English. If you want to work in another language, you're welcome to take our code and start your own fork.
3. For now, stick to what's provided in the Python 3 standard library. We might allow some extra dependencies later on.

CHAPTER 1

The Tile Interface

1.1 Module name

Your tile will have a module name such as `weatbag.tiles.s3e11`, which determines its position on the grid. The co-ordinates use compass directions: n/s followed by e/w. If one co-ordinate is 0, it should be left out. So the tiles around the centre are named:

n2w2	n2w1	n2	n2e1	n2e2
n1w2	n1w1	n1	n1e1	n1e2
w2	w1	centre	e1	e2
s1w2	s1w1	s1	s1e1	s1e2
s2w2	s2w1	s2	s2e1	s2e2

1.2 The Tile class

Your module should define a class called `Tile`, with the following interface:

class `Tile`

`describe``()`
Describe the scene and anything in it. This is called when the player enters the tile, and again if they look around.

`action`(*player*, *do*)
Called when the player does something inside the tile. You need to interpret the action and carry it out. Tile `e1` has a simple example.

Parameters

- **`player`** – the active `Player` instance
- **`do`** – list of strings - the player's input, split up into words, lowercase and with prepositions removed

`leave`(*player*, *direction*)
Optional. Called when the player tries to leave the square. To let them leave, return *True*, or to prevent them, print an explanation and return *False*. If the method isn't found, the player can always leave. Tile `w1` has a simple example.

Parameters

- **`player`** – the active `Player` instance

- **direction** – the direction the player is trying to leave, as a single lowercase character, n/e/s/w

no_path_msg
Optional. The message to display if the player attempts to leave in a direction where there is no tile. The default message says "The undergrowth in that direction is impassable. You turn back."

CHAPTER 2

Other API components

2.1 The player

2.2 Understanding commands

2.3 Combining items

To allow items to be combined to form other items, add a module in `weatbag.items` with the same name as one of the items, replacing spaces with underscores. There, define a function called `combine`, which will take the name of a second object. You only need to define this for one object; the game will try to find it for both objects being combined.

If the objects can be combined, it should describe what has happened, and return a list of the new objects created. Otherwise, it should return `None`. For example, this function is in `weatbag.items.unlit_torch`:

```
def combine(other):
    if other == 'match':
        print("The torch sputters into life.")
        return ['flaming torch']

    return None
```

2.4 Utility functions

CHAPTER 3

Testing tiles

To save you repeatedly playing through the game to test your latest changes, there's a script to skip to any given tile:

```
./test_tile.py n2
```

If the player needs specific items to test the tile, add a list called test_items to the module. The test script will add these to the player's inventory. For example, tile n2 defines:

```
test_items = ['unlit torch']
```

If you prefer to learn by example, have a look at the tiles already in the game.

Once you've made a tile or a group of tiles, submit a pull request to add it to the game. Don't hang about - if someone else adds tiles in the same location, the first one ready to merge gets the spot.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

A

D

L

N

T