Memit Documentation

Release 0.2.1

Carlo E. T. Oliveira

April 16, 2013

CONTENTS

Contents:

CHAPTER

ONE

MEMIT - INTRODUCTION

Memit is serious game designed to explore memetics entrenched deep in the subconcious. This game enables investigations into the human imaginary through a refined computer input capture, assigning values that correlates with symbolic representations.

CHAPTER

TWO

MEMIT - MODULES

Memit is developed in Brython

All Memit functionality is alloted to single module, documented in *Memit - Core*

CHAPTER

THREE

MEMIT - CORE

See Also:

Module `memit`

Note: Aggregates factory, control and interface units in this single module

---

3.1 Dialog

Modal support for inquiring screens.

See Also:

Class `memit.Dialog`

Note: Interface Unit.

3.2 GUI

A factory class wrapping and hiding the detais of actual HTML and Scale Vector Graphics resources required to draw on the top of browser canvas.

See Also:

Class `memit.GUI`

Note: Factory Unit.

3.3 Marker

A projection of the selected dropping spot on each of three dimensions orthogonal planes. This projection helps to localize the selection placement by reducing the dimensionality to a two dimensional surface.

See Also:

Class `memit.Marker`

Note: Interface Unit.

3.4 Piece

Represents the badge to be placed inside the cube. In this cube, it associates the idea accessed by the badge to a collecion of memetic predicates.

See Also:

Class `memit.Piece`

Note: Interface Unit.

3.5 House

A spatial cell representing a three dimentional coordinate system. The cell is the target where a piece can be dropped. The house determines the final association of an idea with the corresponding memetic predicates.

See Also:

Class `memit.House`

Note: Interface Unit.

3.6 Cube

The memetic space. Each dimension or wall of the cube is a memetic dimension divided int three levels of magnitude.

See Also:

Class `memit.Cube`

Note: Interface Unit.

3.7 Form

Input screen devised to collect demographic data.

See Also:

Class `memit.Form`

Note: Interface Unit.

3.8 Phase

Fourth memetic dimention. Translate the cube to the next memetic space.

See Also:

Class `memit.Phase`

Note: Control Unit.

3.9 Board

Basic game support bindind all the features together.

See Also:

Class `memit.Board`

Note: Interface Unit.

CHAPTER

FOUR

MEMIT - CODE DOCS

4.1 Memit - All

Author *Carlo E. T. Oliveira*

Contact carlo@nce.ufrj.br

Date $Date: 2013/03/17 $

Status This is a "work in progress"

Revision $Revision: 0.1 $

Home Labase

Copyright 2011, GPL.

Serious Game in cavalier projection for memetics.

__author__ = "Carlo E. T. Oliveira (carlo@nce.ufrj.br) $Author: carlo $" __version__ = "0.2 $Revision$"[10:-1] __date__ = "2013/03/17 $Date$"

class memit.**Board**(*gui*)
A meme game board with a 3D cube, some pieces, score and puzzle. *Board*

drag(*p=None*)
Enable placement of pieces. Arg p is the piece being dragged

next_jig()
Remove the next piece from the puzzle.

place(**a*)
Placement state method. Assumes _place (active) or _busy states

remove(*piece*)
acts as a default null house

tick()
Time tick updates pump display value and makes the drops fall

class memit.**Cube**(*gui*, *bottom_image*, *rear_image*, *side_image*)
A 3D game memetic space represented in a cavalier projection. *Cube*

hide()

show()

class memit.**Dialog**(*gui*, *img='/studio/paje.png'*, *text=''*, *act=<function <lambda> at 0x30c1f50>*)
Floating panel holding an editable text area. *Dialog*

action(*event*)

get_text()

hide()

set_text(*text*)

show()

class memit.**Form**(*gui=None*)
Collects demographic info and send results to the server. *Form*

class memit.**GUI**(*panel*, *data*)
Factory creating SVG elements, unpacking extra arguments. *GUI*

avatar()

clear()

click(*handler*)

cling(*level*, *element*)

dialog(*text*, *img='/studio/paje.png'*, *act=<function <lambda> at 0x30f55f0>*)

ellipse(*href*, *cx=0*, *cy=0*, *rx=100*, *ry=50*, *style={}*, ***kw*)

get_args()

group(*group=None*, *layer=0*)

handler(*key*, *handle*)

image(*href*, *x=0*, *y=0*, *width=100*, *height=50*, ***kw*)

over(*handler*)

path(*d*, *style={}*, *onMouseOver='noop'*, *onMouseOut='noop'*)

rect(*x=0*, *y=0*, *width=100*, *height=50*, *style={}*)

remove(*element*)

request(*url='/rest/studio/jeppeto?type=2'*, *action=None*, *data=''*)

set(*element*)

text(*text*, *x=150*, *y=25*, *font_size=22*, *text_anchor='middle'*, *style={}*)

textarea(*text*, *x*, *y*, *w*, *h*, *style={}*)

up(*element*)

class memit.**House**(*gui*, *i*, *j*, *k*, *fill*, *r*, *g*, *b*, *board*)
marks a 3D location inside the cube where a piece can be deployed. *House*

on_click(*ev*)

on_out(*ev*)

on_over(*ev*)
Projects three guiding shadows on the orthogonal cube walls

remove(*piece*)
Remove a piece from the house and set state to receive a new piece

class memit.**Marker**(*gui*, *x*, *y*, *fill*, *face*)
Colored shadow on the walls helping the user to deploy a piece in 3D. *Marker*

hide()

on_over(*ev*, *i*, *j*, *k*)

show(*x*, *y*)

class memit.**Phase**(*gui*, *back_layer*, *puzzle*, *component*)
A game stage with a particular scenario and pieces. *Phase*

cube = None
The 3D cube for this phase.

hide()

next_jig()
Remove the next piece from the puzzle.

piece_places = None
Original placement of pieces at phase startup.

pieces = None
Set of pieces to play in this phase.

reset()
Rearrange all pieces into original placement.

show()

class memit.**Piece**(*gui*, *x*, *y*, *fill*, *r*, *g*, *b*, *board*, *pid*)
Bases: memit.Marker

Represents the user choice when deployed insde the 3D open cube. *Piece*

do_markers(**a*)

next_jig()
Remove the next piece from the puzzle.

on_click(*ev*)

on_out(*ev*)

on_over(*ev*)

place(*z*, *y*, *x*, *house*)

reset(*x*, *y*)

show(*x*, *y*)

memit.**logger**(**a*)

memit.**main**(*dc*, *pn*, *gui*, *repo*)
Starting point

memit.**setinterval**(*a*, *b*)

CHAPTER

FIVE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

m