Leap Documentation

Release 0.1

Kunal Tyagi, Pradyumna Paruchuri

Nov 20, 2016

Contents

1 Contents: **3**
1.1 src 3

2 Indices and tables **7**

Python Module Index **9**

Leap is a bare-basic implementation of how future CAD interfaces would be like. It consists of

- input from a 3D device (currently Leap Motion) which is converted to **Gestures**. Each gesture signifies an action to the application.
- application which accepts the data in 3D unlike the 2D input from a mouse. The application here is a simple Cylinder being controlled in OpenGL

CHAPTER 1

Contents:

1.1 src

1.1.1 Leap package

Subpackages

Leap.leap package

Submodules

Leap.leap.leap_start module

Module contents

Leap.ogl package

Submodules

Leap.ogl.application module

Leap.ogl.element module

Leap.ogl.interface module

Wrapper around PyOpenGL and PyGame

class `Leap.ogl.interface.`**`GLInterface`**
Bases: `object`

Creates a screen, and allows easy evaluation of events

`display_loop`()
Main display function, called every iteration

`event_handler`(*event*)
A big switch case function for input event

Parameters **event** – pygame.Event, used to check for events

Returns handler for event.type

init()
Sets the env settings for the application in the beginning

init_loop()
It is called before drawing on the screen each time

key_down(*key*)

Parameters **key** – key pressed

key_up(*key*)

Parameters **key** – key unpressed

main_loop()
Blocking loop, as per the convention of opengl

mouse_button_down(*event*)

Parameters **event** – directly passes from event-handler

mouse_button_up(*event*)

Parameters **event** – directly passes from event-handler

mouse_motion()
Detect motion of mouse

stop_and_exit()
Quit the application properly

unimplemented(*something*)
Simple wrapper around print for ease :param something: any printable object

Leap.ogl.quaternion module

Leap.ogl.sample_elements module

Leap.ogl.world module

Analogue to Stage in a typical application

class Leap.ogl.world.**World**
holds all the elements, and interface to modify any element

add_element(*element*)
adds an element onto the screen

Parameters **element** – adds the element to the list

clear()
removes all the elements

display()
displays the elements to the screen

transform(*pose=None*)
modifies the elements selected

Parameters **pose** – `Leap.ogl.element.Element.transform()`

Module contents

Submodules

Leap.main module

Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

L

Leap, 5
Leap.leap, 3
Leap.ogl, 5
Leap.ogl.interface, 3
Leap.ogl.world, 4

A

add_element() (Leap.ogl.world.World method), 4

C

clear() (Leap.ogl.world.World method), 4

D

display() (Leap.ogl.world.World method), 4
display_loop() (Leap.ogl.interface.GLInterface method), 3

E

event_handler() (Leap.ogl.interface.GLInterface method), 3

G

GLInterface (class in Leap.ogl.interface), 3

I

init() (Leap.ogl.interface.GLInterface method), 4
init_loop() (Leap.ogl.interface.GLInterface method), 4

K

key_down() (Leap.ogl.interface.GLInterface method), 4
key_up() (Leap.ogl.interface.GLInterface method), 4

L

Leap (module), 5
Leap.leap (module), 3
Leap.ogl (module), 5
Leap.ogl.interface (module), 3
Leap.ogl.world (module), 4

M

main_loop() (Leap.ogl.interface.GLInterface method), 4
mouse_button_down() (Leap.ogl.interface.GLInterface method), 4
mouse_button_up() (Leap.ogl.interface.GLInterface method), 4
mouse_motion() (Leap.ogl.interface.GLInterface method), 4

S

stop_and_exit() (Leap.ogl.interface.GLInterface method), 4

T

transform() (Leap.ogl.world.World method), 4

U

unimplemented() (Leap.ogl.interface.GLInterface method), 4

W

World (class in Leap.ogl.world), 4