Class Diagram for tactus-navigator and Direct Dependencies

Ryan Pavlik <ryan.pavlik@snc.edu>

PyMT Python Multi-Touch Modules

These modules are developed by the PyMT project (third-party), and are downloaded from the latest development code and installed by the rp-mt-scripts package

MTWallpaperWindow
provided by PyMT
For use:
"from pymt import MTWallpaperWindow"

Purpose: Window (fullscreen optional) for multi-touch widgets - subclass of MTWindow with an image displayed as a background.

Inherits from: MTWindow
Aggregates: MTWidget
Used by:
- mainpymt.py - sets up multi-touch environment for application, aggregating an MTButton and the PileCanvas (both of which are MTWidget subclasses)

Loose Aggregation

Loose Aggregation (self relationship)

Loose Aggregation of PileCanvas is not explicit in MTWallpaperWindow: see note futher along this line

Loose Aggregation (Implied by MTWidget's loose aggregation self-relationship)

MTWidget
provided by PyMT
For direct use (uncommon):
"from pymt import MTWidget"

Purpose: Generic, base class for a multi-touch widget, which can aggregate references to child widgets and dispatch touch-related events to instances or subclass methods
Inherited by: all widgets in PyMT
Used by:
- all widgets and consumers of widgets

- added to a MTWindow or subclass to use

MTScatterWidget
provided by PyMT
For direct use (uncommon):
"from pymt import MTScatterWidget"

Purpose: Multi-touch widget offering translation, zoom, and scale with one-finger (drag) and two finger gestures (turn, pinch/spread)

Inherits from: MTWidget - generic widget

Used by:
- CanvassedWindow, CanvassedPile, and PileCanvas indirectly through use of child classes (MTScatterSvg, MTScatterImage)

MTKinetic
provided by PyMT
For use:
"from pymt import MTKinetic"

Purpose: "Container" widget that enables kinetic manipulation ("flick" and similar glide/friction based interaction) of child widgets with configurable parameters.

Inherits from: MTWidget
Used by:
- mainpymt.py - the PileCanvas subclasses MTKinetic to be the parent widget of all Canvassed Windows and CanvassedPiles

MTScatterImage
provided by PyMT
For use:
"from pymt import MTScatterImage"

Purpose: Multi-touch widget offering translation, zoom, and scale manipulations and display of a bitmap image.

Inherits from: MTScatterWidget
Inherited by: CanvassedWindow
Used by:
- CanvassedWindow (subclass) as a widget showing a window's icon manipulating the window itself

MTScatterPlane
provided by PyMT
For use:
"from pymt import MTScatterPlane"

Purpose: Multi-touch widget offering translation, zoom, and scale manipulations of all child widgets when children do not handle those events themselves.

Inherits from: MTScatterWidget
Inherited by: CanvassedPile
Used by:
- CanvassedPile (subclass) to manipulate representations of multiple grouped windows when not touching an individual window.

The package providing this module is distributed by Ubuntu Linux

wnck.Window
provided by libwnck's Python bindings
For direct use:
"import wnck"

Purpose: Provide access to and modification of window properties (minimization status, icon, title, etc) through an object-oriented representation of a window - for use in creation of window navigators, task lists, etc.

Used by:
- ManagedWindow: A ManagedWindow is an enhanced, higher-level object composed of a single wnck.Window, event handlers, and miscellaneous methods.

Composition

ManagedWindow
py/managedwindow.py

Purpose: Each instance handles events from a single wnck.Window and provide a higher-level interface to the window data/methods needed for tactus-navigator

Inherits from: None
Inherited by: CanvassedWindow
Composes ("has-a") wnck.Window

Used by:
- demomain.py (directly for demo interface to window manipulation)

- WindowPile (aggregation of one or more related ManagedWindow instances)

- CanvassedWindow (as parent class)

Loose Aggregation

WindowPile
py/windowpile.py

Purpose: Aggregates one or more ManagedWindow instances for collective manipulation (maximization/minimization as a group, etc). Does not "own" those instances - a ManagedWindow can belong to 0 or 1 WindowPiles, and those ManagedWindows persist even if the WindowPile is destructed.

Inherits from: None
Inherited by: CanvassedPile
Aggregates: ("has a reference to") ManagedWindow

Used by:
- demomain.py (directly for demo interface to window manipulation)

- CanvassedPile (as parent class)

CanvassedWindow
py/canvassedwindow.py

Purpose: Perform direct manipulation on windows through the ManagedWindow class using touch widgets in a multi-touch environment.

Inherits from: ("is-a") ManagedWindow, MTScatterImage (multi-touch widget - zoom and rotate disabled in constructor)
Inherited by: None

Used by:
- CanvassedPile (aggregation of one or more related ManagedWindow instances)

- PileCanvas (multi-touch parent widget in "main" - displays CanvassedWindows on nav screen)

CanvassedPile
py/canvassedpile.py

Purpose: Extends WindowPile through multiple inheritance to permit collective manipulation of the aggregated CanvassedWindows (ManagedWindows) using touch widgets in a multi-touch environment, displaying a visual indicator of that association.

Inherits from: WindowPile, MTScatterPlane (multi-touch widget - zoom and rotate disabled in constructor)
Inherited by: CanvassedPile
Composes ("has-a") MTSvg (for background image)
Aggregates: ManagedWindow (by inheritance)

Used by:
- demomain.py (directly for demo interface to window manipulation)

- PileCanvas (multi-touch parent widget - displays CanvassedPiles on nav screen)

Loose Aggregation of PileCanvas in MTWallpaperWindow implied by inheritance of MTWidget: application only creates a single instance of PileCanvas

Composition

Composition

PileCanvas
py/mainpymt.py

Purpose: Serves as the main application, enumerating windows as CanvassedWindows (into "minimized" and a "visible" initial CanvassedPiles) and setting up the initial status of the multi-touch display/system. Extends MTKinetic to serve as a parent widget, provide natural motion, and provide a natural place of extension for future developers. Use of piles is optional/configurable.

Inherits from: MTKinetic
Composes ("has-a") CanvassedWindow - an instance for every normal window, CanvassedPile - all existing instances in use
Used by:
- mainpymt.py "main" section - Created as child widget to a "MTWallpaperWindow" (root-level widget) and initialized to begin application interactivity

tactus-navigator main application
py/mainpymt.py

PileManagerDemo
py/demomain.py

Purpose: Basic demo scaffolding using a standard graphical interface (GTK+) to test non-multi-touch classes: implementation incomplete due to ease of multi-touch testing. Serves as the main application, enumerating windows as ManagedWindows (into "minimized" and a "visible" initial WindowPiles), and handling button events. Not used in normal operation.

Composes ("has-a") ManagedWindow - an instance for every normal window, WindowPile - creates one for initially-restored windows and one for initially-minimized windows.
Used by:
- demomain.py "main" section - Entire application runs in one instance of PileManagerDemo.

base classes test interface (non-multi-touch)
py/demomain.py

tactus-navigator

These modules were all designed and built expressly for the tactus-navigator component of this CSCI-460 project.

Key:

Class origins:

Provided by a third-party library

Written for the tactus-navigator project

Class relationships:

Inheritance: A (subclass) "is-a" B
(note: A also inherits all relationships of B, not necessarily shown separately)

Composition: A "has-a" B

"has" 0+ instances of B

"has" 1 instance of B

Aggregation (loose): A"has a reference to" B

"has references to" 0+ instances of B

"has a reference to" 1 instance of B