Effects Plugin

Description

Apply animation effects to components and windows. Inspired by script.aculo.us effects.

Installation

The current version of **griffon-effects-plugin** is **0.4**
To install just issue the following command

```
griffon install-plugin effects
```

Usage

Effects can be applied in two ways:

- create a new Effect instance and run it.
- use the convenience methods exposed by `griffon.effects.Effects` (each method matches the effect's name in lower case, Move => Effects.move)

Every effect takes the following parameters:

- `component` - the component to be animated
- `params` - a map with effect options, may be empty or null
- `callback` - a closure to be called at the end of the effect, optional.

All effects share the following options unless otherwise specified:

- `duration` - ong, how long should the animation take. default: 500l
- `delay` - long, wait time before the animation starts. default: 0l
- `ease` - TimelineEase. default: Linear
- `wait` - boolean. Forces the current thread to wait until the effect has finished running default: false

> Make sure the calling thread is not the UI thread when setting `wait:` to true

Some effects accept an `anchor` option, whose value is defined by the `griffon.effects.Anchor` enum. Additional valid values are string literals in lower case, with underscores substituted by spaces; literals from SwingConstants are also valid. Examples

- Anchor.TOP == 'top' == 'TOP' == 'NORTH' == 'north
- Anchor.TOP_LEFT == 'top left == 'top_left' == 'NORTH EAST' == 'north east'

Basic Effects

Move

Animates the `location` property of the target component

Parameters:

- `x` - int, in pixels. default: 0i
- `y` - int, in pixels. default: 0i
- `mode` - String, if movement should be 'relative' or 'absolute'. default: 'relative'

Resize

Animates the `size` property of the target component
Parameters:

- `w` - int, in pixels. default: 0i
- `h` - int, in pixels. default: 0
- `mode` - String, if the update should be 'relative' or 'absolute'. default: 'relative'

Bounds

Animates the `bounds` property of a the target component
Parameters:

- `x` - int, in pixels. default: 0i
- `y` - int, in pixels. default: 0i
- `w` - int, in pixels. default: 0i
- `h` - int, in pixels. default: 0
- `mode` - String, if the update should be 'relative' or 'absolute'. default: 'relative'

Scale

Animates the `bounds` property of the target component by calculating a scale factor
Parameters:

- `scaleX` - boolean, if the x coorinate should scale. default: true
- `scaleY` - boolean, if the y coorinate should scale. default: true
- `from` - float, starting value in percentage. default: 100.0f
- `to` - float, ending value in percentage. default: 0.0f
- `anchor` - Anchor, anchoring point. default: Anchor.CENTER

Opacity

Animates a window's `opacity` property
Parameters:

- `from` - float, starting value. default: 0.0f
- `to` - float, ending value. default: 1.0f

Fade

Animates a window's `opacity` from its current value or 1.0f to 0.0f

Appear

Animates a window's `opacity` from its current value or 0.0f to 1.0f

Composite Effects

Shake

Moves a component from right to left a few times
Parameters:

- `distance` - int, in pixels. default: 20i

Puff

Fades and blows up a window

DropOut

Fades and moves a window out of the screen
Paremeters:

- `anchor` - Anchor, anchoring point. default: Anchor.BOTTOM

DropIn

Appears and moves a window to the center of the screen
Paremeters:

- `anchor` - Anchor, anchoring point. default: Anchor.TOP

Chained events

Effects can be chained in a sequential manner by using the `chain(List<Effect>)` method provided by `griffon.effects.Effects` utility class.

History

Version	Date	Notes
0.4	06-27-10	Release sync with Griffon 0.9.3-beta-2
0.3	05-06-11	Upgraded dependencies
0.2.1	02-22-11	Upgraded dependencies
0.2	12-21-10	Release sync with Griffon 0.9.2
0.1	07-22-10	First release