Cocos

API Documentation

March 24, 2008

Contents

Contents 1

1 Package cocos 2
- 1.1 Modules . . . 2
- 1.2 Functions . . . 2
- 1.3 Variables . . . 2

2 Module cocos.actions 3
- 2.1 Functions . . . 5
- 2.2 Class ForwardDir . . . 6
- 2.3 Class BackwardDir . . . 6
- 2.4 Class PingPongMode . . . 6
- 2.5 Class RestartMode . . . 6
- 2.6 Class ActionSprite . . . 6
  - 2.6.1 Methods . . . 6
  - 2.6.2 Properties . . . 7
  - 2.6.3 Class Variables . . . 8
- 2.7 Class Action . . . 8
  - 2.7.1 Methods . . . 8
  - 2.7.2 Properties . . . 9
- 2.8 Class IntervalAction . . . 10
  - 2.8.1 Methods . . . 10
  - 2.8.2 Properties . . . 11
- 2.9 Class Place . . . 11
  - 2.9.1 Methods . . . 11
  - 2.9.2 Properties . . . 12
- 2.10 Class Hide . . . 12
  - 2.10.1 Methods . . . 13
  - 2.10.2 Properties . . . 13
- 2.11 Class Show . . . 13
  - 2.11.1 Methods . . . 14
  - 2.11.2 Properties . . . 14
- 2.12 Class Blink . . . 14
  - 2.12.1 Methods . . . 15
  - 2.12.2 Properties . . . 15
- 2.13 Class Rotate . . . 16
  - 2.13.1 Methods . . . 16
  - 2.13.2 Properties . . . 17

2.14 Class Scale . . . 17
2.14.1 Methods . . . 17
2.14.2 Properties . . . 18
2.15 Class Goto . . . 18
2.15.1 Methods . . . 19
2.15.2 Properties . . . 19
2.16 Class Move . . . 20
2.16.1 Methods . . . 20
2.16.2 Properties . . . 21
2.17 Class Jump . . . 21
2.17.1 Methods . . . 22
2.17.2 Properties . . . 22
2.18 Class Bezier . . . 23
2.18.1 Methods . . . 23
2.18.2 Properties . . . 24
2.19 Class Spawn . . . 24
2.19.1 Methods . . . 25
2.19.2 Properties . . . 26
2.20 Class Sequence . . . 26
2.20.1 Methods . . . 26
2.20.2 Properties . . . 27
2.21 Class Repeat . . . 28
2.21.1 Methods . . . 28
2.21.2 Properties . . . 29
2.22 Class CallFunc . . . 29
2.22.1 Methods . . . 30
2.22.2 Properties . . . 30
2.23 Class CallFuncS . . . 30
2.23.1 Methods . . . 31
2.23.2 Properties . . . 31
2.24 Class Delay . . . 32
2.24.1 Methods . . . 32
2.24.2 Properties . . . 32
2.25 Class RandomDelay . . . 33
2.25.1 Methods . . . 33
2.25.2 Properties . . . 33
2.26 Class FadeOut . . . 34
2.26.1 Methods . . . 34
2.26.2 Properties . . . 35
2.27 Class FadeIn . . . 35
2.27.1 Methods . . . 35
2.27.2 Properties . . . 36

3 Module cocos.director **37**
3.1 Variables . . . 39

4 Module cocos.effect **40**
4.1 Class Effect . . . 40
4.1.1 Methods . . . 40
4.1.2 Properties . . . 41
4.2 Class TextureFilterEffect . . . 41
4.2.1 Methods . . . 41
4.2.2 Properties . . . 42

4.3 Class ColorizeEffect . . . 42
4.3.1 Methods . . . 42
4.3.2 Properties . . . 43
4.4 Class RepositionEffect . . . 43
4.4.1 Methods . . . 44
4.4.2 Properties . . . 44

5 Module cocos.framegrabber **45**
5.1 Functions . . . 45

6 Module cocos.layer **46**
6.1 Class Layer . . . 46
6.1.1 Methods . . . 46
6.1.2 Properties . . . 48
6.1.3 Class Variables . . . 48
6.2 Class MultiplexLayer . . . 48
6.2.1 Methods . . . 48
6.2.2 Properties . . . 49
6.2.3 Class Variables . . . 49
6.3 Class ColorLayer . . . 50
6.3.1 Methods . . . 50
6.3.2 Properties . . . 50
6.3.3 Class Variables . . . 51

7 Module cocos.menu **52**
7.1 Variables . . . 52
7.2 Class Menu . . . 52
7.2.1 Methods . . . 53
7.2.2 Properties . . . 54
7.2.3 Class Variables . . . 54
7.2.4 Instance Variables . . . 54
7.3 Class MenuItem . . . 54
7.3.1 Methods . . . 55
7.3.2 Properties . . . 55
7.4 Class ToggleMenuItem . . . 56
7.4.1 Methods . . . 56
7.4.2 Properties . . . 56

8 Module cocos.scene **58**
8.1 Class Scene . . . 58
8.1.1 Methods . . . 58
8.1.2 Properties . . . 59

9 Module cocos.transitions **60**
9.1 Class TransitionScene . . . 60
9.1.1 Methods . . . 60
9.1.2 Properties . . . 61
9.1.3 Class Variables . . . 61
9.2 Class Quad2DTransition . . . 61
9.2.1 Methods . . . 61
9.2.2 Properties . . . 62
9.2.3 Class Variables . . . 62
9.3 Class FadeTransition . . . 63

9.3.1 Methods . . . 63
9.3.2 Properties . . . 63
9.3.3 Class Variables . . . 64
9.4 Class SlideLRTransition . . . 64
9.4.1 Methods . . . 64
9.4.2 Properties . . . 64
9.4.3 Class Variables . . . 65
9.5 Class SlideRLTransition . . . 65
9.5.1 Methods . . . 65
9.5.2 Properties . . . 66
9.5.3 Class Variables . . . 66
9.6 Class SlideBTTransition . . . 66
9.6.1 Methods . . . 66
9.6.2 Properties . . . 67
9.6.3 Class Variables . . . 67
9.7 Class SlideTBTransition . . . 67
9.7.1 Methods . . . 67
9.7.2 Properties . . . 68
9.7.3 Class Variables . . . 68
9.8 Class MoveInTTransition . . . 68
9.8.1 Methods . . . 69
9.8.2 Properties . . . 69
9.8.3 Class Variables . . . 69
9.9 Class MoveInBTransition . . . 70
9.9.1 Methods . . . 70
9.9.2 Properties . . . 70
9.9.3 Class Variables . . . 70
9.10 Class MoveInLTransition . . . 71
9.10.1 Methods . . . 71
9.10.2 Properties . . . 71
9.10.3 Class Variables . . . 72
9.11 Class MoveInRTransition . . . 72
9.11.1 Methods . . . 72
9.11.2 Properties . . . 73
9.11.3 Class Variables . . . 73
9.12 Class GrowTransition . . . 73
9.12.1 Methods . . . 73
9.12.2 Properties . . . 74
9.12.3 Class Variables . . . 74
9.13 Class CornerMoveTransition . . . 74
9.13.1 Methods . . . 74
9.13.2 Properties . . . 75
9.13.3 Class Variables . . . 75
9.14 Class ShrinkAndGrow . . . 75
9.14.1 Methods . . . 76
9.14.2 Properties . . . 76
9.14.3 Class Variables . . . 76

1 Package cocos

Los Cocos: An extension for Pyglet

http://code.google.com/p/los-cocos/

Version: 0.2.0

Author: PyAr Team

1.1 Modules

- **actions**: Classes to manipulate sprites.
  (Section 2, p. 3)
- **director**: Singleton that handles the logic behind the Scenes
  (Section 3, p. 37)
- **effect**: Effects that can be applied to layers.
  (Section 4, p. 40)
- **framegrabber**: Utility classes for rendering to a texture.
  (Section 5, p. 45)
- **layer**: Layer class and subclasses
  (Section 6, p. 46)
- **menu**: A menu layer for los-cocos.
  (Section 7, p. 52)
- **scene**: Scene class and subclasses
  (Section 8, p. 58)
- **transitions**: Transitions between Scenes
  (Section 9, p. 60)

1.2 Functions

check_pyglet_version()

1.3 Variables

Name	Description
version	**Value:** `'0.2.0'`

2 Module cocos.actions

Classes to manipulate sprites.

Animating a sprite

To animate an sprite you need to execute an action.

Actions that modifies the sprite's properties:

- `Move` ((x,y), duration)
- `Goto` ((x,y), duration)
- `Rotate` (degrees, duration)
- `Scale` (zoom_factor, duration)
- `Jump` (height, x, number_of_jumps, duration)
- `Bezier` (bezier_configuration, duration)
- `Place` ((x,y))
- `FadeIn` (duration)
- `FadeOut` (duration)
- `Blink` (times_to_blink, duration)
- `Show` ()
- `Hide` ()

Composite actions:

- `Repeat` (action)
- `Spawn` (list_of_actions)
- `Sequence` (list_of_actions)

Misc actions:

- `CallFunc` (function)
- `CallFuncS` (function)
- `Delay` (seconds)
- `RandomDelay` (lo_seconds, hi_seconds)

To execute any action you need to create an action:

```
move = Move( (50,0), 5 )
```

In this case, `move` is an action that will move the sprite 50 pixels to the right (`x` coordinate), 0 pixel in the `y` coordinate, and 0 pixels in the `z` coordinate in 5 seconds.

And now tell the sprite to execute it:

```
sprite.do( move )
```

Interval Actions

An interval action is an action that takes place within a certain period of time. It has an start time, and a finish time. The finish time is the parameter `duration` plus the start time.

These `IntervalAction` have some interesting properties, like:

- They can run Forward (default)
- They can run Backwards
- They can alter the speed of time

For example, if you run an action in a Forward direction and the you run it again in a Backward direction, then you are simulation a PingPong movement.

These actions has 3 special parameters:

`dir` (**direction**)

It can be `ForwardDir` or `BackwardDir` . Default is: `ForwardDir`

`mode` (**repeat mode**)

It can be `PingPongMode` or `RestartMode` . Default is : `PingPongMode` .

`time_func` (**a function. The format of the function is f(runtime, duration)**)

If you want to alter the speed of time, you should provide your onw time_func or use the `accelerate` function. Default : None. No alter-time function is used.

Available IntervalActions

- `Goto`
- `Move`
- `Jump`
- `Bezier`
- `Blink`
- `Rotate`
- `Scale`

- `FadeOut`
- `FadeIn`

Examples:

```
move = Move( (200,0), 5 )     # Moves 200 pixels to the right in 5 seconds.
                              # Direction: ForwardDir (default)
                              # RestartMode:  PingPongMode (default)
                              # time_func:  No alter function (default)

rmove = Repeat( move )        # Will repeat the action *move* forever
                              # The repetitions are in PingPongMode
                              # times: -1 (default)

move2 = Move( (200,0), 5, time_func=accelerate )
                             # Moves 200 pixels to the right in 5 seconds
                             # time_func=accelerate. This means that the
                             # speed is not linear. It will start to action
                             # very slowly, and it will increment the speed
                             # in each step. The total running time will be
                             # 5 seconds.

move3 = Move( (200,0), 5, dir=BackwardDir )
                             # Moves 200 pixels to the **left** in 5 seconds
                             # But when you use this direction (BackwardDir)
                             # the starting coords and the finishing coords
                             # are inverted
```

2.1 Functions

accelerate(*t*, *duration*)

Function that simulates an acceleration from 0 to duration seconds. Use this function to alter the time of some IntervalActions. **Parameters**

`t:` Elapsed time since the start of the action. *(type=float)*

`duration:` Duration time in seconds. *(type=float)*

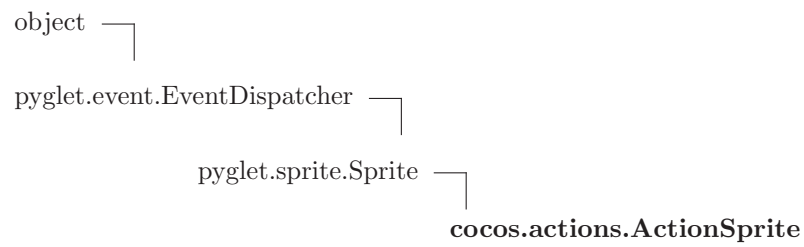
2.2 Class ForwardDir

2.3 Class BackwardDir

2.4 Class PingPongMode

2.5 Class RestartMode

2.6 Class ActionSprite



ActionSprites are sprites that can execute actions.

Example:

```
sprite = ActionSprite('grossini.png')
```

2.6.1 Methods

__init__(*self*, **args*, ***kwargs*)

Create a sprite. **Parameters**

`img:`	Image or animation to display.
`x:`	X coordinate of the sprite.
`y:`	Y coordinate of the sprite.
`blend_src:`	OpenGL blend source mode. The default is suitable for compositing sprites drawn from back-to-front.
`blend_dest:`	OpenGL blend destination mode. The default is suitable for compositing sprites drawn from back-to-front.
`batch:`	Optional batch to add the sprite to.
`group:`	Optional parent group of the sprite.

Overrides: object.__init__ extit(inherited documentation)

do(*self*, *action*)

Executes an *action*. When the action finished, it will be removed from the sprite's queue.
Parameters
`action:` Action that will be executed. *(type=an* `Action` *instance)*
Return Value
A clone of *action (type=*`Action` *instance)*

remove(*self*, *action*)

Removes an action from the queue **Parameters**
`action:` Action to be removed *(type=Action)*

stop(*self*)

Removes running actions from the queue

step(*self*, *dt*)

This method is called every frame. **Parameters**
`dt:` The time that elapsed since that last time this functions was called.
(type=delta_time)

Inherited from pyglet.sprite.Sprite

__del__(), delete(), draw(), on_animation_end(), set_position()

Inherited from pyglet.event.EventDispatcher

dispatch_event(), event(), pop_handlers(), push_handlers(), register_event_type(), remove_handler(), remove_handlers(), set_handler(), set_handlers()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.6.2 Properties

Name	Description
Inherited from pyglet.sprite.Sprite batch, group, height, image, opacity, position, rotation, scale, visible, width, x, y	

continued on next page

Name	Description
Inherited from object	
__class__	

2.6.3 Class Variables

Name	Description
Inherited from pyglet.sprite.Sprite	
event_types	

2.7 Class Action

object
└
cocos.actions.Action

Known Subclasses: cocos.actions.IntervalAction, cocos.actions.CallFunc, cocos.actions.Delay, cocos.actions.Hide, cocos.actions.Place, cocos.actions.Repeat, cocos.actions.Sequence, cocos.actions.Show, cocos.actions.Spawn

2.7.1 Methods

__init__(*self*, **args*, ***kwargs*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

init(*self*)

done(*self*)

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur.

restart(*self*)

Called before an action si restarted.

Called before action is restarted. This happens when an action is being repeated.

step(*self*, *dt*)

get_runtime(*self*)

Returns the runtime in seconds.

__add__(*self*, *action*)

Is the Sequence Action

__or__(*self*, *action*)

Is the Spawn Action

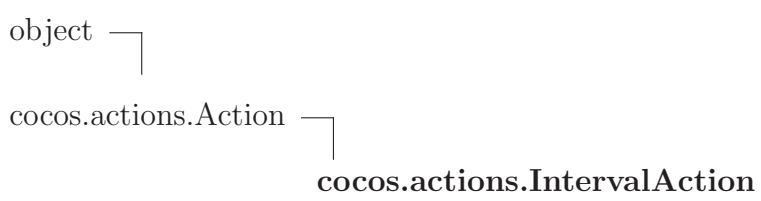
Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.7.2 Properties

Name	Description
Inherited from object	
__class__	

2.8 Class IntervalAction



Known Subclasses: cocos.actions.Bezier, cocos.actions.Blink, cocos.actions.FadeOut, cocos.actions.Goto, cocos.actions.Jump, cocos.actions.Rotate, cocos.actions.Scale

IntervalAction(dir=ForwardDir, mode=PingPongMode, time_func=None)

Abstract Class that defines the direction of any Interval Action. Interval Actions are the ones that can go forward or backwards in time.

For example: `Goto` , `Move` , `Rotate` are Interval Actions. `CallFunc` is not.

dir can be: `ForwardDir` or `BackwardDir` mode can be: `PingPongMode` or `RestartMode` time_func can be any function that alters the time. `accelerate` , a time-alter function, is provided with this lib.

2.8.1 Methods

__init__(*dir*=`ForwardDir`, *mode*=`PingPongMode`, *time_func*=`None`)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

restart(*self*)

Called before an action si restarted.

Called before action is restarted. This happens when an action is being repeated. Overrides: cocos.actions.Action.restart extit(inherited documentation)

done(*self*)

Overrides: cocos.actions.Action.done

get_runtime(*self*)

Returns the runtime in seconds. Overrides: cocos.actions.Action.get_runtime extit(inherited documentation)

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __or__(), init(), start(), step()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.8.2 Properties

Name	Description
Inherited from object	
__class__	

2.9 Class Place

object

cocos.actions.Action

cocos.actions.Place

Place the sprite in the position x,y.

Example:

```
action = Place( (320,240,0) )
sprite.do( action )
```

2.9.1 Methods

init(*self*, *position*)

Init method. **Parameters**

`position:` Coordinates where the sprite will be placed *(type=(x,y))*

Overrides: cocos.actions.Action.init

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

done(*self*)

Overrides: cocos.actions.Action.done

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __init__(), __or__(), get_runtime(), restart(), step()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.9.2 Properties

Name	Description
Inherited from object	
__class__	

2.10 Class Hide

object

cocos.actions.Action

cocos.actions.Hide

Hides the sprite. To show it again call the `Show` () action

Example:

```
action = Hide()
sprite.do( action )
```

2.10.1 Methods

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

done(*self*)

Overrides: cocos.actions.Action.done

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __init__(), __or__(), get_runtime(), init(), restart(), step()

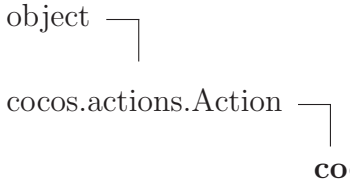
Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.10.2 Properties

Name	Description
Inherited from object	
__class__	

2.11 Class Show



Shows the sprite. To hide it call the `Hide` () action

Example:

```
action = Show()
sprite.do( action )
```

2.11.1 Methods

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

done(*self*)

Overrides: cocos.actions.Action.done

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __init__(), __or__(), get_runtime(), init(), restart(), step()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.11.2 Properties

Name	Description
Inherited from object	
__class__	

2.12 Class Blink

object

cocos.actions.Action

cocos.actions.IntervalAction

cocos.actions.Blink

Blinks the sprite a Number_of_Times, for Duration seconds

Example:

```
action = Blink( 10, 2 ) # Blinks 10 times in 2 seconds
```

```
sprite.do( action )
```

2.12.1 Methods

init(*self*, *times*, *duration*)

Init method. **Parameters**

`times:` Number of times to blink *(type=integer)*

`duration:` Duration time in seconds *(type=float)*

Overrides: cocos.actions.Action.init

step(*self*, *dt*)

Overrides: cocos.actions.Action.step

Inherited from cocos.actions.IntervalAction(Section 2.8)

__init__(), done(), get_runtime(), restart()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __or__(), start()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.12.2 Properties

Name	Description
Inherited from object	
__class__	

2.13 Class Rotate

object
cocos.actions.Action
cocos.actions.IntervalAction
cocos.actions.Rotate

Rotates a sprite counter-clockwise in degrees

Example:

```
action = Rotate( 180, 2 )       # rotates the sprite 180 de-
grees in 2 seconds
sprite.do( action )
```

2.13.1 Methods

init(*self*, *angle*, *duration*=5)

Init method. **Parameters**

`angle:` Degrees that the sprite will be rotated. Positive degrees rotates the sprite conter-clockwise. *(type=float)*

`duration:` Duration time in seconds *(type=float)*

Overrides: cocos.actions.Action.init

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

step(*self*, *dt*)

Overrides: cocos.actions.Action.step

Inherited from cocos.actions.IntervalAction(Section 2.8)

__init__(), done(), get_runtime(), restart()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __or__()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.13.2 Properties

Name	Description
Inherited from object	
__class__	

2.14 Class Scale

object

cocos.actions.Action

cocos.actions.IntervalAction

cocos.actions.Scale

Scales the sprite

Example:

```
action = Scale( 5, 2 )       # scales the sprite 5x in 2 seconds
sprite.do( action )
```

2.14.1 Methods

init(*self*, *zoom*, *duration*=5)

Init method. **Parameters**

`zoom:` scale factor *(type=float)*

`duration:` Duration time in seconds *(type=float)*

Overrides: cocos.actions.Action.init

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

step(*self*, *dt*)

Overrides: cocos.actions.Action.step

Inherited from cocos.actions.IntervalAction(Section 2.8)

__init__(), done(), get_runtime(), restart()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __or__()

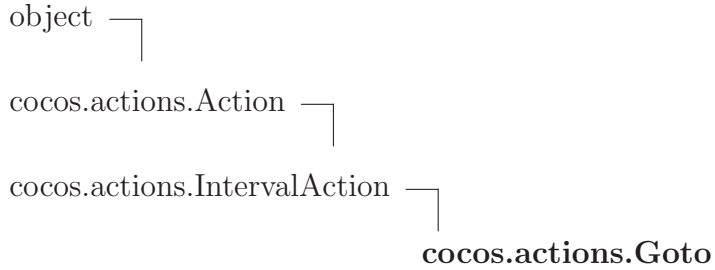
Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.14.2 Properties

Name	Description
Inherited from object	
__class__	

2.15 Class Goto



Known Subclasses: cocos.actions.Move

Moves a sprite to the position x,y. x and y are absolute coordinates.

Example:

```
action = Goto( (50,10), 8 )       # Move the sprite to co-
ords x=50, y=10 in 8 seconds
sprite.do( action )
```

2.15.1 Methods

init(*self*, *dst_coords*, *duration*=5)

Init method. **Parameters**

`dst_coords`: Coordinates where the sprite will be placed at the end of the action *(type=(x,y))*

`duration`: Duration time in seconds *(type=float)*

Overrides: cocos.actions.Action.init

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

step(*self*, *dt*)

Overrides: cocos.actions.Action.step

Inherited from cocos.actions.IntervalAction(Section 2.8)

__init__(), done(), get_runtime(), restart()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __or__()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.15.2 Properties

Name	Description
Inherited from object	
__class__	

2.16 Class Move

object
└ cocos.actions.Action
└ cocos.actions.IntervalAction
└ cocos.actions.Goto
└ **cocos.actions.Move**

Mve a sprite x,y pixels. x and y are relative to the position of the sprite. Duration is is seconds.

Example:

```
action = Move( (-
50,0), 8 )  # Move the sprite 50 pixels to the left in 8 seconds
sprite.do( action )
```

2.16.1 Methods

init(*self*, *delta*, *duration*=5)

Init method. **Parameters**

`delta:` Delta coordinates *(type=(x,y))*

`duration:` Duration time in seconds *(type=float)*

Overrides: cocos.actions.Action.init

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

Inherited from cocos.actions.Goto(Section 2.15)

step()

Inherited from cocos.actions.IntervalAction(Section 2.8)

__init__(), done(), get_runtime(), restart()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __or__()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.16.2 Properties

Name	Description
Inherited from object	
__class__	

2.17 Class Jump

object

cocos.actions.Action

cocos.actions.IntervalAction

cocos.actions.Jump

Moves a sprite simulating a jump movement.

Example:

```
action = Jump(50,200, 5, 6)    # Move the sprite 200 pix-
els to the right
sprite.do( action )            # in 6 seconds, doing 5 jumps
                               # of 50 pixels of height
```

2.17.1 Methods

init(*self*, *y*=150, *x*=120, *jumps*=1, *duration*=5)

Init method **Parameters**

- `y:` Height of jumps *(type=integer)*
- `x:` horizontal movement relative to the startin position *(type=integer)*
- `jumps:` quantity of jumps *(type=integer)*
- `duration:` Duration time in seconds *(type=float)*

Overrides: cocos.actions.Action.init

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

step(*self*, *dt*)

Overrides: cocos.actions.Action.step

Inherited from cocos.actions.IntervalAction(Section 2.8)

__init__(), done(), get_runtime(), restart()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __or__()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.17.2 Properties

Name	Description
Inherited from object	
__class__	

2.18 Class Bezier

object
└
cocos.actions.Action
└
cocos.actions.IntervalAction
└
cocos.actions.Bezier

Moves a sprite through a bezier path

Example:

```
action = Bezier( bezier_conf.path1, 5 )   # Moves the sprite us-
ing the
sprite.do( ac-
tion )                        # bezier path 'bezier_conf.path1'
                                          # in 5 seconds
```

2.18.1 Methods

init(*self*, *bezier*, *duration*=5)

Init method **Parameters**

`bezier:` A bezier configuration *(type=bezier_configuration instance)*

`duration:` Duration time in seconds *(type=float)*

Overrides: cocos.actions.Action.init

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

step(*self*, *dt*)

Overrides: cocos.actions.Action.step

Inherited from cocos.actions.IntervalAction(Section 2.8)

__init__(), done(), get_runtime(), restart()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __or__()

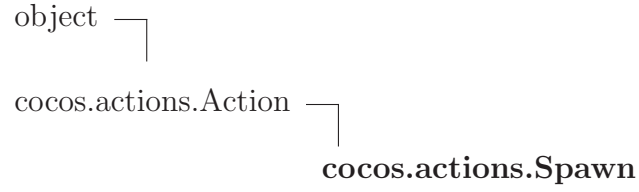
Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.18.2 Properties

Name	Description
Inherited from object	
__class__	

2.19 Class Spawn



Spawn a new action immediately. You can spawn actions using:

- the Spanw() class
- the overriden | operator

- call sprite.do() many times

Example:

```
action = Spawn( action1, action2, action3 )
sprite.do( action )

or:

sprite.do( action1 | action2 | action3 )

or:

sprite.do( action1 )
sprite.do( action2 )
sprite.do( action3 )
```

2.19.1 Methods

init(*self*, **actions*)

Init method **Parameters**

`actions`: The list of actions that will be spawned *(type=list of actions)*

Overrides: cocos.actions.Action.init

done(*self*)

Overrides: cocos.actions.Action.done

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __init__(), __or__(), get_runtime(), restart(), step()

Inherited from object

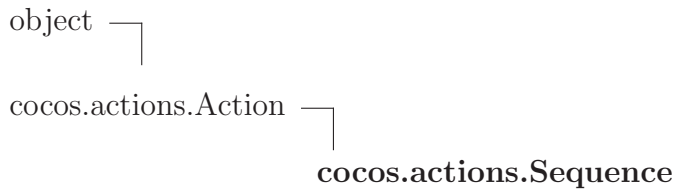
__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),

__repr__(), __setattr__(), __str__()

2.19.2 Properties

Name	Description
Inherited from object	
__class__	

2.20 Class Sequence



Run actions sequentially: One after another You can sequence actions using:

- the Sequence() class
- the overriden + operator

Example:

```
action = Sequence( action1, action2, action3 )
sprite.do( action )

or:

sprite.do( action1 + action2 + action3 )
```

2.20.1 Methods

init(*self*, **actions*, ***kwargs*)

Init method **Parameters**
`actions`: List of actions to be sequenced *(type=list of actions)*

Overrides: cocos.actions.Action.init

restart(*self*)

Called before an action si restarted.

Called before action is restarted. This happens when an action is being repeated. Overrides: cocos.actions.Action.restart extit(inherited documentation)

instantiate(*self*)

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

done(*self*)

Overrides: cocos.actions.Action.done

step(*self*, *dt*)

Overrides: cocos.actions.Action.step

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __init__(), __or__(), get_runtime()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.20.2 Properties

Name	Description
Inherited from object	
__class__	

2.21 Class Repeat

object
└
cocos.actions.Action
└
cocos.actions.Repeat

Repeats an action. It is is similar to Sequence, but it runs the same action every time

Example:

```
action = Jump( 50,200,3,5)
repeat = Repeat( action, times=5 )
sprite.do( repeat )
```

2.21.1 Methods

init(*self*, *action*, *times*=`-1`)

Init method. **Parameters**

`action`: The action that will be repeated *(type=`Action` instance)*

`times`: The number of times that the action will be repeated. -1, which is the default value, means *repeat forever* *(type=integer)*

Overrides: cocos.actions.Action.init

restart(*self*)

Called before an action si restarted.

Called before action is restarted. This happens when an action is being repeated. Overrides: cocos.actions.Action.restart extit(inherited documentation)

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

instantiate(*self*)

done(*self*)

Overrides: cocos.actions.Action.done

step(*self*, *dt*)

Overrides: cocos.actions.Action.step

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __init__(), __or__(), get_runtime()

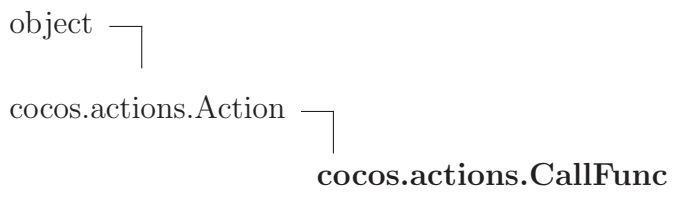
Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.21.2 Properties

Name	Description
Inherited from object	
__class__	

2.22 Class CallFunc



Known Subclasses: cocos.actions.CallFuncS

An action that will call a function.

Example:

```
def my_func():
    print "hello baby"

action = CallFunc( my_func )
sprite.do( action )
```

2.22.1 Methods

init(*self*, *func*, **args*, ***kwargs*)

Overrides: cocos.actions.Action.init

done(*self*)

Overrides: cocos.actions.Action.done

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __init__(), __or__(), get_runtime(), restart(), step()

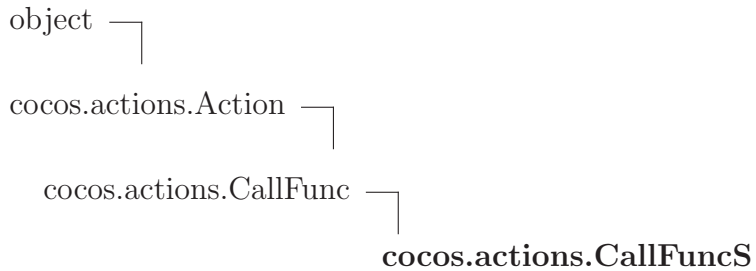
Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.22.2 Properties

Name	Description
Inherited from object	
__class__	

2.23 Class CallFuncS



An action that will call a funtion with the target as the first argument

Example:

```
def my_func( sprite ):
    print "hello baby"

action = CallFuncS( my_func )
sprite.do( action )
```

2.23.1 Methods

start(*self*)

Called before the action starts to execute

When this method is called, the variable self.target will contain a reference to the sprite. If you want to use this variable before this method is called, an unexpected error will occur. Overrides: cocos.actions.Action.start extit(inherited documentation)

Inherited from cocos.actions.CallFunc(Section 2.22)

done(), init()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __init__(), __or__(), get_runtime(), restart(), step()

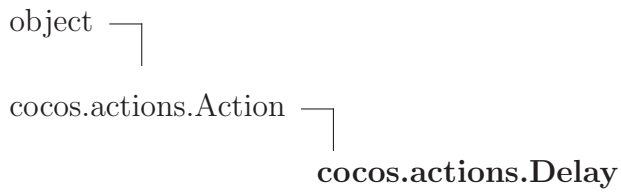
Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.23.2 Properties

Name	Description
Inherited from object	
__class__	

2.24 Class Delay



Known Subclasses: cocos.actions.RandomDelay

Delays the action a certain ammount of seconds

Example:

```
action = Delay(2.5)
sprite.do( action )
```

2.24.1 Methods

init(*self*, *delay*)

Init method **Parameters**

`delay`: Seconds of delay *(type=float)*

Overrides: cocos.actions.Action.init

done(*self*)

Overrides: cocos.actions.Action.done

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __init__(), __or__(), get_runtime(), restart(), start(), step()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.24.2 Properties

Name	Description
Inherited from object	
__class__	

2.25 Class RandomDelay

object
cocos.actions.Action
cocos.actions.Delay
cocos.actions.RandomDelay

Delays the actions between *min* and *max* seconds

Example:

```
action = RandomDelay(2.5, 4.5)      # delays the action be-
tween 2.5 and 4.5 seconds
sprite.do( action )
```

2.25.1 Methods

init(*self*, *low*, *hi*)

Init method **Parameters**

`low:` Minimun seconds of delay *(type=float)*

`hi:` Maximun seconds of delay *(type=float)*

Overrides: cocos.actions.Action.init

Inherited from cocos.actions.Delay(Section 2.24)

done()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __init__(), __or__(), get_runtime(), restart(), start(), step()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.25.2 Properties

Name	Description
Inherited from object	

continued on next page

Name	Description
__class__	

2.26 Class FadeOut

object

cocos.actions.Action

cocos.actions.IntervalAction

cocos.actions.FadeOut

Known Subclasses: cocos.actions.FadeIn

FadeOut(duration) Fades out an sprite

Example:

```
action = FadeOut( 2 )
sprite.do( action )
```

2.26.1 Methods

init(*self*, *duration*)

Init method. **Parameters**
`duration`: Seconds that it will take to fade *(type=float)*

Overrides: cocos.actions.Action.init

step(*self*, *dt*)

Overrides: cocos.actions.Action.step

Inherited from cocos.actions.IntervalAction(Section 2.8)

__init__(), done(), get_runtime(), restart()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __or__(), start()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.26.2 Properties

Name	Description
Inherited from object	
__class__	

2.27 Class FadeIn

object

cocos.actions.Action

cocos.actions.IntervalAction

cocos.actions.FadeOut

cocos.actions.FadeIn

FadeIn(duration) Fades in an sprite

Example:

```
action = FadeIn( 2 )
sprite.do( action )
```

2.27.1 Methods

step(*self*, *dt*)

Overrides: cocos.actions.Action.step

Inherited from cocos.actions.FadeOut(Section 2.26)

init()

Inherited from cocos.actions.IntervalAction(Section 2.8)

__init__(), done(), get_runtime(), restart()

Inherited from cocos.actions.Action(Section 2.7)

__add__(), __or__(), start()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

2.27.2 Properties

Name	Description
Inherited from object	
__class__	

3 Module cocos.director

Singleton that handles the logic behind the Scenes

Director

Initializing

The director is the singleton that creates and handles the main `Window` and manages the logic behind the `Scenes`.

The first thing to do, is to initialize the `director`:

```
from cocos.director import *
director.init( list_of_arguments )
```

This will initialize the director, and will create a display area (a 640x480 window by default). The parameters that are supported by director.init() are the same parameters that are supported by pyglet.window.Window().

Some of the supported parameters are:

- `fullscreen`: Boolean. Window is created in fullscreen. Default is False
- `resizable`: Boolean. Window is resizable. Default is False
- `vsync`: Boolean. Sync with the vertical retrace. Default is True
- `width`: Integer. Window width size. Default is 640
- `height`: Integer. Window height size. Default is 480
- `caption`: String. Window title.
- `visible`: Boolean. Window is visible or not. Default is True.

For example:

```
director.init( caption="Hello World", fullscreen=True )
```

For a complete list of the supported parameters, see the pyglet Window documentation.

Running a Scene

Once you have initialized the director, you can run your first `Scene`:

```
director.run( Scene( MyLayer() ) )
```

This will run a scene that has only 1 layer: `MyLayer()`. You can run a scene that has multiple layers. For more information about `Layers` and `Scenes` refer to the `Layers` and `Scene` documentation.

`cocos.director.Director`

Once a scene is running you can do the following actions:

- `director.replace( new_scene )`: Replaces the running scene with the new_scene
- `director.push( new_scene )`: The running scene will be pushed to a queue of scenes to run, and new_scene will be executed.
- `director.pop()`: Will pop out a scene from the queue, and it will replace the running scene.
- `director.end( end_value )`: Finishes the current scene with an end value of `end_value`. The next scene to be run will be popped from the queue.

Other functions you can use are:

- `director.get_window_size()`: Returns an (x,y) pair with the _logical_ dimensions of the display. The display might have been resized, but coordinates are always relative to this size. If you need the _physical_ dimensions, check the dimensions of `director.window`
- `get_virtual_coordinates(self, x, y)`: Transforms coordinates that belongs the real (physical) window size, to the coordinates that belongs to the virtual (logical) window. Returns an x,y pair in logical coordinates.

The director also has some useful attributes:

- `director.return_value`: The value returned by the last scene that called `director.end`. This is useful to use scenes somewhat like function calls: you push a scene to call it, and check the return value when the director returns control to you.
- `director.window`: This is the pyglet window handled by this director, if you happen to need low level access to it.
- `self.show_FPS`: You can set this to a boolean value to enable, disable the framerate indicator.

- `self.scene`: The scene currently active

3.1 Variables

Name	Description
director	The singleton; check `cocos.director.Director` for details on usage. Don't instantiate Director(). Just use this singleton. **Value:** `Director()`

4 Module cocos.effect

Effects that can be applied to layers.

Effect

Effect are visual transformations that can be applied to layers. You normally use them by calling the `cocos.layer.Layer.set_effect` method on a layer, passing an effect as argument (or None to disable the effect).

This module provides some Effect classes which are ready to use:

- ColorizeEffect: Multiplicative filter of color/alpha
- RepositionEffect: Resize/relocate layer

And some abstract base classes to define your own effects:

- Effect: The base class for every effect
- **TextureFilterEffect: Captures the layer into a texture, allowing you to** display it in any way that a texture can be used.

An effect object should not be applied to several layers at once.

4.1 Class Effect

object
└ **cocos.effect.Effect**

Known Subclasses: cocos.effect.TextureFilterEffect

Abstract base class for effects. Effects are applied to layers (or anything that is shown with a draw() method). Useful effects can inherit this one, which is just the identity effect

4.1.1 Methods

prepare(*self*, *target*)

Advance target in dt, preparing effect display.

show(*self*)

Show layer+effect on screen

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __init__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

4.1.2 Properties

Name	Description
Inherited from object	
__class__	

4.2 Class TextureFilterEffect

object

cocos.effect.Effect

cocos.effect.TextureFilterEffect

Known Subclasses: cocos.effect.ColorizeEffect, cocos.effect.RepositionEffect

Base class for texture based effects. Prepare captures layer in `self.texture`, with a window sized capture. Show just blits the texture, override to do more interesting things

4.2.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

prepare(*self*, *target*)

Advance target in dt, preparing effect display. Overrides: cocos.effect.Effect.prepare extit(inherited documentation)

show(*self*)

`self.texture` contains the layer; redefine this method to show the texture applying the effect you want. Overrides: cocos.effect.Effect.show

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

4.2.2 Properties

Name	Description
Inherited from object	
__class__	

4.3 Class ColorizeEffect

object
└
cocos.effect.Effect
└
cocos.effect.TextureFilterEffect
└
cocos.effect.ColorizeEffect

Applies recoloring (multiplication) and alpha blending.

4.3.1 Methods

__init__(*self*, *color*=`(1, 1, 1, 1)`)

New colorize effect. `color` is stored in `self.color`, which can be modified later. **Parameters**

`color`: The color that will be multiplied by the layer colors. *(type=a 4-uple (red, green, blue, alpha) of floats in 0.0-1.0 range)*

Overrides: object.__init__

show(*self*)

Blits `self.texture` calling glColor4f before with `self.color`. Overrides: cocos.effect.Effect.show

Inherited from cocos.effect.TextureFilterEffect(Section 4.2)

prepare()

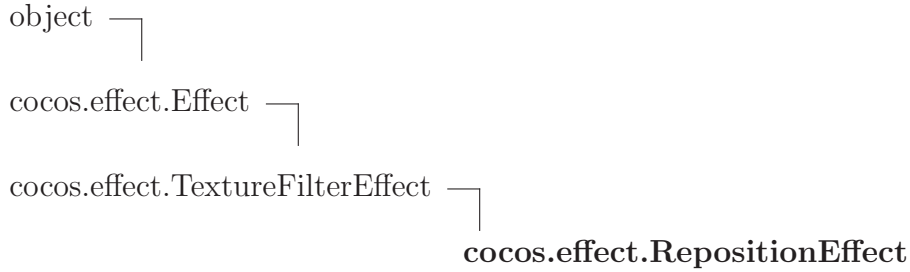
Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

4.3.2 Properties

Name	Description
Inherited from object	
__class__	

4.4 Class RepositionEffect



Applies repositioning and scaling

4.4.1 Methods

__init__(*self*, *x*=0, *y*=0, *width*=None, *height*=None)

New reposition effect. Parameters are stored in `self.x`, `self.y`, `self.width`, `self.height`, which can be modified later. **Parameters**

`x:` The horizontal shift for the layer (default=0) *(type=Integer)*

`y:` The vertical shift for the layer (default=0) *(type=Integer)*

`width:` The horizontal size for the layer (default=0) *(type=Integer)*

`height:` The vertical size for the layer (default=0) *(type=Integer)*

Overrides: object.__init__

show(*self*)

Blits `self.texture` at the rectangle defined by `self.x`, `self.y`, `self.width`, `self.height` Overrides: cocos.effect.Effect.show

Inherited from cocos.effect.TextureFilterEffect(Section 4.2)

prepare()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

4.4.2 Properties

Name	Description
Inherited from object	
__class__	

5 Module cocos.framegrabber

Utility classes for rendering to a texture.

It is mostly used for internal implementation of cocos, you normally shouldn't need it. If you are curious, check implementation of effects to see an example.

5.1 Functions

TextureGrabber()
Returns an instance of the best texture grabbing class

6 Module cocos.layer

Layer class and subclasses

A `Layer` has as size the whole drawable area (window or screen), and knows how to draw itself. It can be semi transparent (having holes and/or partial transparency in some/all places), allowing to see other layers behind it. Layers are the ones defining appearance and behavior, so most of your programming time will be spent coding Layer subclasses that do what you need. The layer is where you define event handlers. Events are propagated to layers (from front to back) until some layer catches the event and accepts it.

6.1 Class Layer

object
└
cocos.layer.Layer

Known Subclasses: cocos.layer.ColorLayer, cocos.layer.MultiplexLayer, cocos.menu.Menu

Class that handles events and other important game's behaviors

6.1.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

add(*self*, **o*)

Adds an object to the batch. The batch will draw it. **Parameters**

`o`: Object that supports the 'batch' property, like Sprites, Labels, `ActionSprite` , etc. *(type=list of objects)*

disable_step(*self*)

Disables the step callback

draw(*self*)

Subclasses shall override this method if they want to draw custom objects

enable_step(*self*)

Enables the step callback. It calls the `step()` method every frame

on_draw(*self*)

Draws every object that is in the batch. It then calls `self.draw()`.
Subclassess shall override `self.draw` to draw custom objects.

on_enter(*self*)

Called every time the layer enters into the scene

on_exit(*self*)

Called every time the layer quits the scene

set_effect(*self*, *e*)

Apply effect e to this layer. if e is None, current effect (if any) is removed
Parameters

`e`: The effect that will be applied to the layer *(type=`Effect` instance)*

step(*self*, *dt*)

Called every frame when it is active. By default `step` is disabled. See `enable_step` and `disable_step` **Parameters**

`dt`: Time that elapsed since the last time `step` was called. *(type=float)*

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

6.1.2 Properties

Name	Description
Inherited from object	
__class__	

6.1.3 Class Variables

Name	Description
effects	**Value:** `()`

6.2 Class MultiplexLayer

object

cocos.layer.Layer

cocos.layer.MultiplexLayer

A Composite layer that only enables one layer at the time.

This is useful, for example, when you have 3 or 4 menus, but you want to show one at the time

6.2.1 Methods

__init__(*self*, **layers*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

switch_to(*self*, *layer_number*)

Switches to another Layer that belongs to the Multiplexor. **Parameters**

`layer_number`: MUST be a number between 0 and the quantities of layers - 1. The running layer will receive an "on_exit()" call, and the new layer will receive an "on_enter()" call. *(type=Integer)*

on_enter(*self*)

Called every time the layer enters into the scene Overrides: cocos.layer.Layer.on_enter extit(inherited documentation)

on_exit(*self*)

Called every time the layer quits the scene Overrides: cocos.layer.Layer.on_exit extit(inherited documentation)

draw(*self*)

Subclasses shall override this method if they want to draw custom objects Overrides: cocos.layer.Layer.draw extit(inherited documentation)

Inherited from cocos.layer.Layer(Section 6.1)

add(), disable_step(), enable_step(), on_draw(), set_effect(), step()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

6.2.2 Properties

Name	Description
Inherited from object	
__class__	

6.2.3 Class Variables

Name	Description
Inherited from cocos.layer.Layer (Section 6.1)	

continued on next page

Name	Description
effects	

6.3 Class ColorLayer

object
└
cocos.layer.Layer
└
cocos.layer.ColorLayer

Creates a layer of a certain color. The color shall be specified in the format (r,g,b,a).

For example, to create green layer:

```
l = ColorLayer( (0.0, 1.0, 0.0, 1.0 ) )
```

6.3.1 Methods

__init__(*self*, **color*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

draw(*self*)

Subclasses shall override this method if they want to draw custom objects
Overrides: cocos.layer.Layer.draw extit(inherited documentation)

Inherited from cocos.layer.Layer(Section 6.1)

add(), disable_step(), enable_step(), on_draw(), on_enter(), on_exit(), set_effect(), step()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

6.3.2 Properties

Name	Description
Inherited from object	
__class__	

6.3.3 Class Variables

Name	Description
Inherited from cocos.layer.Layer (Section 6.1)	
effects	

7 Module cocos.menu

A menu layer for los-cocos.

Menu

This module provides a Menu class, which is a layer you can use in cocos apps. Menus can contain regular items (which trigger a function when selected) or toggle items (which toggle a flag when selected).

When you need a menu, you can define a class inheriting `Menu`, and setting some attributes which control the menu appearance. Then you add `MenuItem` s to it, prepare it, and use it as you would use any layer.

There is a menu demo in the samples folder.

7.1 Variables

Name	Description
CENTER	**Value:** `'center'`
LEFT	**Value:** `'left'`
RIGHT	**Value:** `'right'`
TOP	**Value:** `'top'`
BOTTOM	**Value:** `'bottom'`

7.2 Class Menu

object
└ cocos.layer.Layer
└ **cocos.menu.Menu**

Abstract base class for menu layers.

Normal usage is:

- create a subclass
- override __init__ to set all style attributes, then add items using `add_item()`, and then call `build_items()`

- Finally you shall add the menu to a `Scene`

7.2.1 Methods

__init__(*self*, *title=''*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

add_item(*self*, *item*)

Adds an item to the menu.

The order of the list important since the first one will be shown first.
Parameters
`item`: The MenuItem that will part of the `Menu` *(type=a `MenuItem`)*

draw(*self*)

Subclasses shall override this method if they want to draw custom objects
Overrides: cocos.layer.Layer.draw extit(inherited documentation)

build_items(*self*)

Initializes all the menu items

Call this method after you've added all the menu items.

on_key_press(*self*, *symbol*, *modifiers*)

on_mouse_release(*self*, *x*, *y*, *buttons*, *modifiers*)

on_mouse_motion(*self*, *x*, *y*, *dx*, *dy*)

Inherited from cocos.layer.Layer(Section 6.1)

add(), disable_step(), enable_step(), on_draw(), on_enter(), on_exit(), set_effect(), step()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),

__repr__(), __setattr__(), __str__()

7.2.2 Properties

Name	Description
Inherited from object	
__class__	

7.2.3 Class Variables

Name	Description
Inherited from cocos.layer.Layer (Section 6.1)	
effects	

7.2.4 Instance Variables

Name	Description
font_title	Title's font name
font_title_size	Title's font size. Default size is 56
font_title_color	Title's font color. Default color is (192, 192, 192, 255)
font_items	Item's font name
font_items_size	Items' font size when unselected. Default size is 32
font_items_color	Items' font color when unselected. Default color is (192, 192, 192, 255)
font_items_selected_size	Items' font size when selected. Default size is 48
font_items_selected_color	Items' font color when selected. Default color is (255, 255, 255, 255)
menu_halign	Horizontal alignment. Possible options: CENTER, RIGHT or LEFT. Default is CENTER
menu_valign	Vertical alignment. Possible options: CENTER, TOP or BOTTOM. Default is CENTER

7.3 Class MenuItem

object
└ **cocos.menu.MenuItem**

Known Subclasses: cocos.menu.ToggleMenuItem

A menu item triggering a function.

7.3.1 Methods

__init__(*self*, *label*, *activate_func*)

Creates a new menu item **Parameters**

`label:` The label the of the menu item *(type=string)*

`activate_func:` The callback function *(type=function)*

Overrides: object.__init__

get_box(*self*)

Returns the box that contains the menu item. **Return Value**

returns a tuple (a rectangle) that sets the boundaries of the menu item. *(type=(x1,x2,y1,y2))*

draw(*self*)

on_key_press(*self*, *symbol*, *modifiers*)

is_inside_box(*self*, *x*, *y*)

Returns whether the point (x,y) is inside the menu item. **Return Value**

Whether or not the point (x,y) is inside the menu item *(type=Boolean)*

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

7.3.2 Properties

Name	Description
Inherited from object	

continued on next page

Name	Description
__class__	

7.4 Class ToggleMenuItem

object
└
cocos.menu.MenuItem
└
cocos.menu.ToggleMenuItem

A menu item for a boolean toggle option.

When selected, `self.value` is toggled, the callback function is called with `self.value` as argument.

7.4.1 Methods

__init__(*self*, *label*, *value*, *toggle_func*)

Creates a Toggle Menu Item **Parameters**

`label:` Item's label *(type=string)*

`value:` Item's default value: True or False *(type=Boolean)*

`toggle_func:` Callback function *(type=function)*

Overrides: object.__init__

on_key_press(*self*, *symbol*, *modifiers*)

Overrides: cocos.menu.MenuItem.on_key_press

Inherited from cocos.menu.MenuItem(Section 7.3)

draw(), get_box(), is_inside_box()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

7.4.2 Properties

Name	Description
Inherited from object	
__class__	

8 Module cocos.scene

Scene class and subclasses

8.1 Class Scene

object
└
cocos.scene.Scene

Known Subclasses: cocos.transitions.TransitionScene

8.1.1 Methods

__init__(*self*, **layers*)

Creates a Scene with layers, from bottom to top, giving a z-value from 0.0 to len(layers)-1 **Parameters**

`layers:` Layers that will be part of the scene. *(type=list of* ***`Layer`****)*

Overrides: object.__init__

add(*zvalue*, *layer*, *layer_name*=...)

Adds a layer at z-value depth, naming it 'name' if given. (name can be used later to remove the layer) **Return Value**

None

end(*self*, *value*=`None`)

Ends the current scene setting director.return_value with `value` **Parameters**

`value:` The return value. It can be anything. A type or an instance. *(type=anything)*

on_draw(*self*)

Called every time the scene can be drawn.

on_enter(*self*)

Called every time just before the scene is run.

on_exit(*self*)

Called every time just before the scene leaves the stage

remove(*self*, *layer_name*)

Removes a layer from the scene given the layer_name. If the layer can't be removed, and exception will be risen **Parameters**

`layer_name: Layer` name *(type=string)*

remove_layer(*self*, *layer*)

Removes a layer from the scene given a layer's reference. **Parameters**

`layer: Layer` reference *(type=`Layer`)*

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

8.1.2 Properties

Name	Description
Inherited from object	
__class__	

9 Module cocos.transitions

Transitions between Scenes

9.1 Class TransitionScene

object
cocos.scene.Scene
cocos.transitions.TransitionScene

Known Subclasses: cocos.transitions.Quad2DTransition

A Scene that takes two scenes and makes a transition between them

9.1.1 Methods

__init__(*self*, *out_scene*, *in_scene*, *duration*=2)

Creates a Scene with layers, from bottom to top, giving a z-value from 0.0 to len(layers)-1 **Parameters**

`layers`: Layers that will be part of the scene.

Overrides: object.__init__ extit(inherited documentation)

disable_step(*self*)

Disables the step callback

enable_step(*self*)

Enables the step callback. It calls the 'step' method every frame

step(*self*, *dt*)

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_draw(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

9.1.2 Properties

Name	Description
Inherited from object	
__class__	

9.1.3 Class Variables

Name	Description
in_texture	**Value:** `None`
out_texture	**Value:** `None`

9.2 Class Quad2DTransition

object

cocos.scene.Scene

cocos.transitions.TransitionScene

cocos.transitions.Quad2DTransition

Known Subclasses: cocos.transitions.CornerMoveTransition, cocos.transitions.FadeTransition, cocos.transitions.GrowTransition, cocos.transitions.MoveInBTransition, cocos.transitions.MoveInLTransiti cocos.transitions.MoveInRTransition, cocos.transitions.MoveInTTransition, cocos.transitions.ShrinkAndG cocos.transitions.SlideBTTransition, cocos.transitions.SlideLRTransition, cocos.transitions.SlideRLTransit cocos.transitions.SlideTBTransition

Slides out one scene while sliding in onother.

9.2.1 Methods

__init__(*self*, **args*, ***kwargs*)

Creates a Scene with layers, from bottom to top, giving a z-value from 0.0 to len(layers)-1 **Parameters**

`layers`: Layers that will be part of the scene.

Overrides: object.__init__ extit(inherited documentation)

step(*self*, *dt*)

Overrides: cocos.transitions.TransitionScene.step

on_draw(*self*)

Called every time the scene can be drawn. Overrides: cocos.scene.Scene.on_draw extit(inherited documentation)

draw_scenes(*self*)

blit_texture(*self*, *texture*, *p*)

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

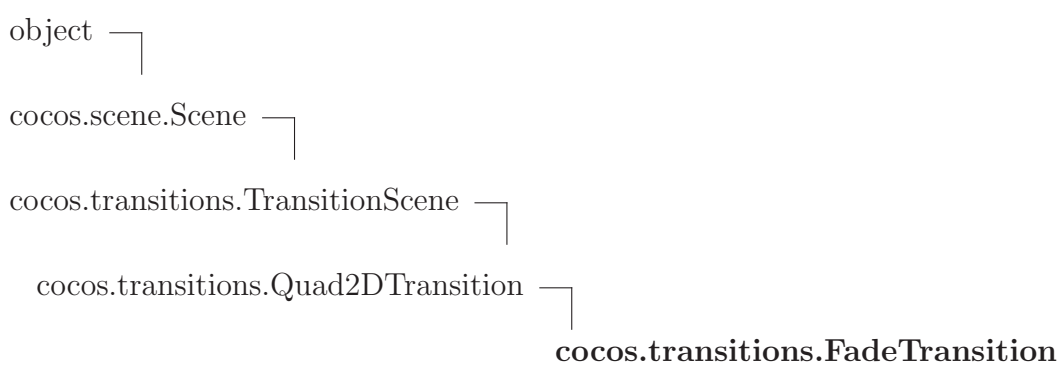
9.2.2 Properties

Name	Description
Inherited from object	
__class__	

9.2.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.3 Class FadeTransition



9.3.1 Methods

draw_scenes(*self*)

Overrides: cocos.transitions.Quad2DTransition.draw_scenes

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

9.3.2 Properties

Name	Description
Inherited from object	
__class__	

9.3.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.4 Class SlideLRTransition

object

cocos.scene.Scene

cocos.transitions.TransitionScene

cocos.transitions.Quad2DTransition

cocos.transitions.SlideLRTransition

9.4.1 Methods

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), draw_scenes(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

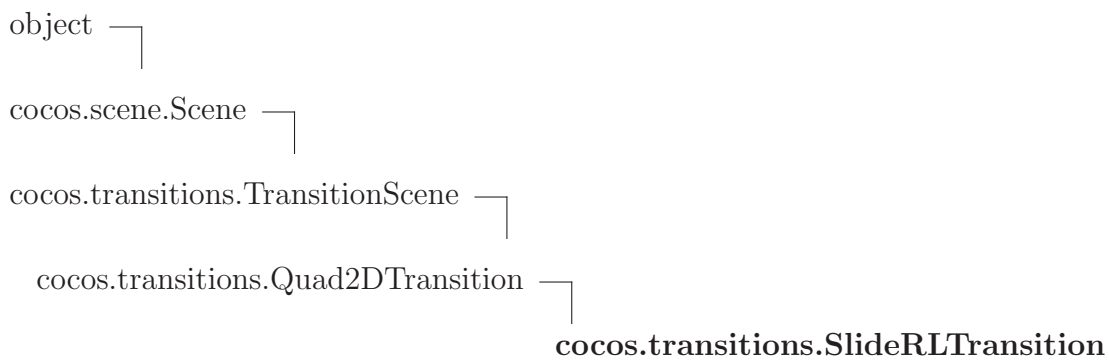
9.4.2 Properties

Name	Description
Inherited from object	
__class__	

9.4.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.5 Class SlideRLTransition



9.5.1 Methods

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), draw_scenes(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

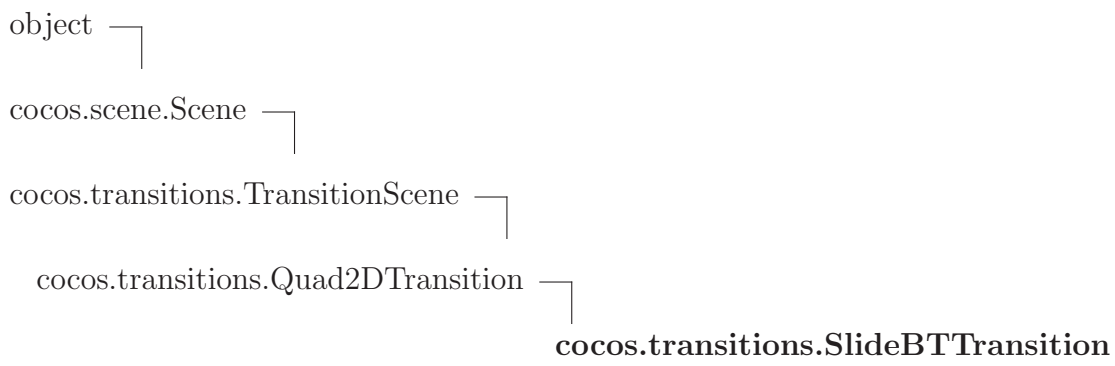
9.5.2 Properties

Name	Description
Inherited from object	
__class__	

9.5.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.6 Class SlideBTTransition



9.6.1 Methods

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), draw_scenes(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

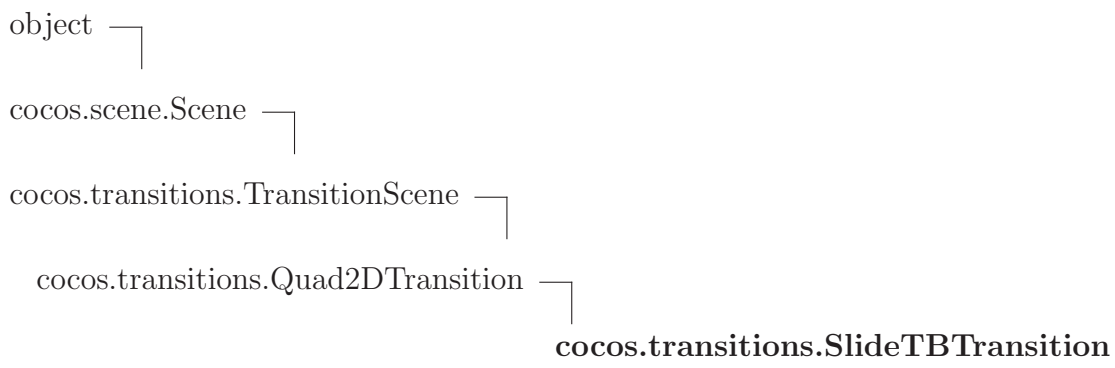
9.6.2 Properties

Name	Description
Inherited from object	
__class__	

9.6.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.7 Class SlideTBTransition



9.7.1 Methods

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), draw_scenes(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

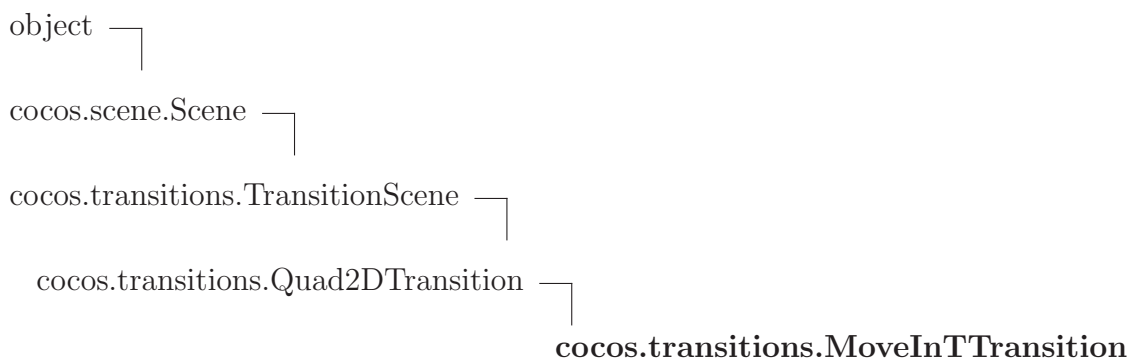
9.7.2 Properties

Name	Description
Inherited from object	
__class__	

9.7.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.8 Class MoveInTTransition



9.8.1 Methods

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), draw_scenes(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

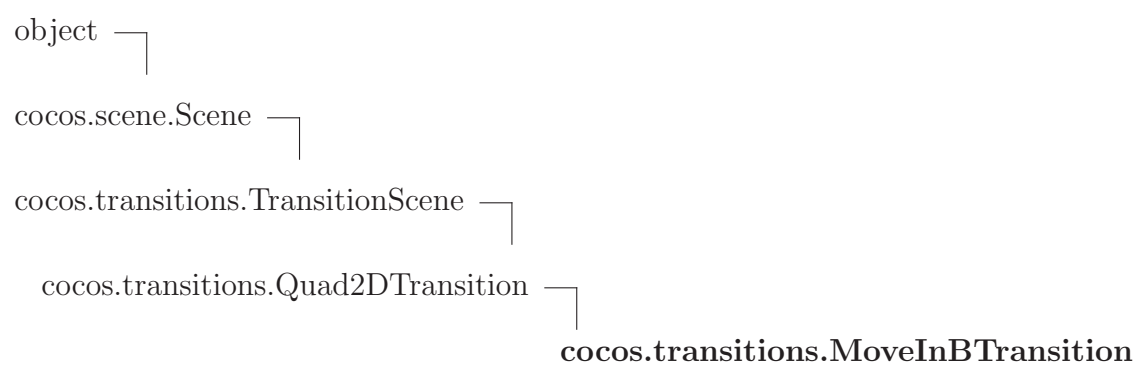
9.8.2 Properties

Name	Description
Inherited from object	
__class__	

9.8.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.9 Class MoveInBTransition



9.9.1 Methods

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), draw_scenes(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

9.9.2 Properties

Name	Description
Inherited from object	
__class__	

9.9.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.10 Class MoveInLTransition

object
└ cocos.scene.Scene
└ cocos.transitions.TransitionScene
└ cocos.transitions.Quad2DTransition
└ **cocos.transitions.MoveInLTransition**

9.10.1 Methods

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), draw_scenes(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

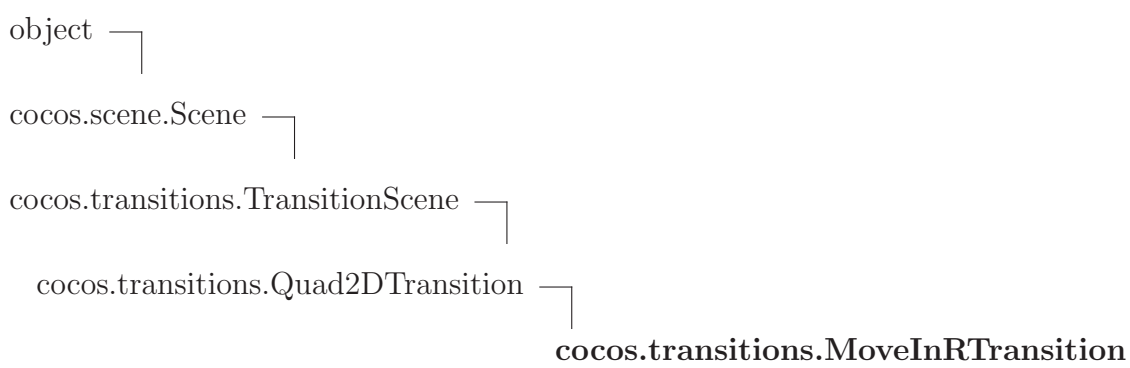
9.10.2 Properties

Name	Description
Inherited from object	
__class__	

9.10.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.11 Class MoveInRTransition



9.11.1 Methods

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), draw_scenes(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

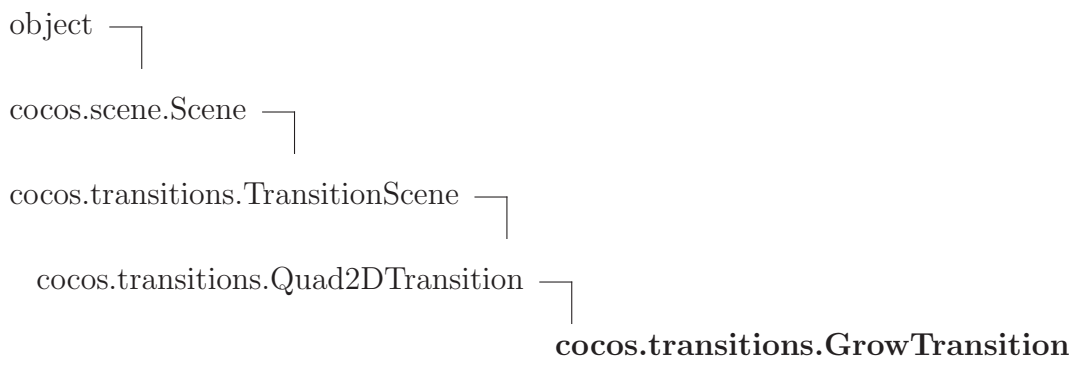
9.11.2 Properties

Name	Description
Inherited from object	
__class__	

9.11.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.12 Class GrowTransition



9.12.1 Methods

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), draw_scenes(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

9.12.2 Properties

Name	Description
Inherited from object	
__class__	

9.12.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.13 Class CornerMoveTransition

object
└ cocos.scene.Scene
└ cocos.transitions.TransitionScene
└ cocos.transitions.Quad2DTransition
└ **cocos.transitions.CornerMoveTransition**

9.13.1 Methods

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), draw_scenes(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

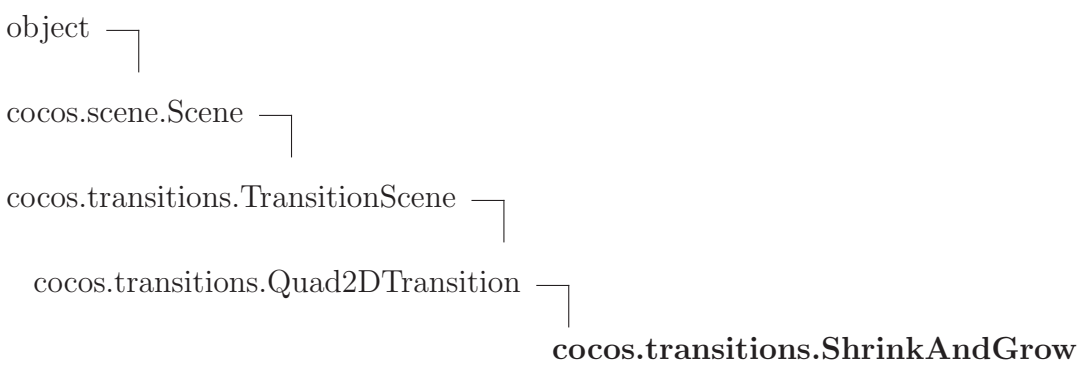
9.13.2 Properties

Name	Description
Inherited from object	
__class__	

9.13.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

9.14 Class ShrinkAndGrow



9.14.1 Methods

draw_scenes(*self*)

Overrides: cocos.transitions.Quad2DTransition.draw_scenes

out_proyect(*self*, *x*, *y*)

in_proyect(*self*, *x*, *y*)

Inherited from cocos.transitions.Quad2DTransition(Section 9.2)

__init__(), blit_texture(), on_draw(), step()

Inherited from cocos.transitions.TransitionScene(Section 9.1)

disable_step(), enable_step()

Inherited from cocos.scene.Scene(Section 8.1)

add(), end(), on_enter(), on_exit(), remove(), remove_layer()

Inherited from object

__delattr__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __str__()

9.14.2 Properties

Name	Description
Inherited from object	
__class__	

9.14.3 Class Variables

Name	Description
Inherited from cocos.transitions.TransitionScene (Section 9.1)	
in_texture, out_texture	

Index

cocos *(package)*, 2
  cocos.actions *(module)*, 3–36
    cocos.actions.accelerate *(function)*, 5
    cocos.actions.Action *(class)*, 8–9
    cocos.actions.ActionSprite *(class)*, 6–8
    cocos.actions.BackwardDir *(class)*, 6
    cocos.actions.Bezier *(class)*, 23–24
    cocos.actions.Blink *(class)*, 14–15
    cocos.actions.CallFunc *(class)*, 29–30
    cocos.actions.CallFuncS *(class)*, 30–31
    cocos.actions.Delay *(class)*, 31–32
    cocos.actions.FadeIn *(class)*, 35–36
    cocos.actions.FadeOut *(class)*, 34–35
    cocos.actions.ForwardDir *(class)*, 5–6
    cocos.actions.Goto *(class)*, 18–20
    cocos.actions.Hide *(class)*, 12–13
    cocos.actions.IntervalAction *(class)*, 9–11
    cocos.actions.Jump *(class)*, 21–23
    cocos.actions.Move *(class)*, 20–21
    cocos.actions.PingPongMode *(class)*, 6
    cocos.actions.Place *(class)*, 11–12
    cocos.actions.RandomDelay *(class)*, 32–34
    cocos.actions.Repeat *(class)*, 27–29
    cocos.actions.RestartMode *(class)*, 6
    cocos.actions.Rotate *(class)*, 15–17
    cocos.actions.Scale *(class)*, 17–18
    cocos.actions.Sequence *(class)*, 26–27
    cocos.actions.Show *(class)*, 13–14
    cocos.actions.Spawn *(class)*, 24–26
  cocos.check_pyglet_version *(function)*, 2
  cocos.director *(module)*, 37–39
  cocos.effect *(module)*, 40–44
    cocos.effect.ColorizeEffect *(class)*, 42–43
    cocos.effect.Effect *(class)*, 40–41
    cocos.effect.RepositionEffect *(class)*, 43–44
    cocos.effect.TextureFilterEffect *(class)*, 41–42
  cocos.framegrabber *(module)*, 45
    cocos.framegrabber.TextureGrabber *(function)*, 45
  cocos.layer *(module)*, 46–51
    cocos.layer.ColorLayer *(class)*, 50–51
    cocos.layer.Layer *(class)*, 46–48
    cocos.layer.MultiplexLayer *(class)*, 48–50
  cocos.menu *(module)*, 52–57
    cocos.menu.Menu *(class)*, 52–54
    cocos.menu.MenuItem *(class)*, 54–56
    cocos.menu.ToggleMenuItem *(class)*, 56–57
  cocos.scene *(module)*, 58–59
    cocos.scene.Scene *(class)*, 58–59
  cocos.transitions *(module)*, 60–76
    cocos.transitions.CornerMoveTransition *(class)*, 74–75
    cocos.transitions.FadeTransition *(class)*, 62–64
    cocos.transitions.GrowTransition *(class)*, 73–74
    cocos.transitions.MoveInBTransition *(class)*, 69–71
    cocos.transitions.MoveInLTransition *(class)*, 71–72
    cocos.transitions.MoveInRTransition *(class)*, 72–73
    cocos.transitions.MoveInTTransition *(class)*, 68–69
    cocos.transitions.Quad2DTransition *(class)*, 61–62
    cocos.transitions.ShrinkAndGrow *(class)*, 75–76
    cocos.transitions.SlideBTTransition *(class)*, 66–67
    cocos.transitions.SlideLRTransition *(class)*, 64–65
    cocos.transitions.SlideRLTransition *(class)*, 65–66
    cocos.transitions.SlideTBTransition *(class)*, 67–68

cocos.transitions.TransitionScene *(class)*,
60–61