
bouquet

Release 0.4.0.dev0

mak8kammerer

May 09, 2024

CONTENTS

1	Bouquet	1
2	Gradients	3
3	Indices and tables	7
	Python Module Index	9
	Index	11

BOUQUET

Bouquet library is collection of awesome widgets and helpful tools for Kivy framework.

Tiny reference:

GRADIENTS

`bouquet.gradients` module provides various types of gradients. Currently, only linear (*LinearGradient*) and radial (*RadialGradient*) gradients are supported. Other types of gradients will be added later.

2.1 How to use gradients with Vertex Instructions?

This is possible to use gradient textures with Vertex Instructions via the `render_texture()` function. You can take a look at code examples in the *Examples* section below.

2.2 Examples

LinearGradient:

```
from kivy.app import App
from bouquet.gradients import LinearGradient

class MyApp(App):

    def build(self):
        return LinearGradient(
            top_left_color='purple',
            top_right_color='purple',
            bottom_left_color='orange',
            bottom_right_color='orange',
        )

MyApp().run()
```

RadialGradient:

```
from kivy.app import App
from kivy.lang import Builder

import bouquet.gradients

KV = """
RadialGradient:
    center_color: (1.0, 0.0, 0.0, 1.0)
```

(continues on next page)

(continued from previous page)

```

        border_color: (1.0, 0.0, 0.0, 0.0)
    """

    class MyApp(App):

        def build(self):
            return Builder.load_string(KV)

    MyApp().run()

```

LinearGradient.render_texture():

```

from kivy.app import App
from kivy.lang import Builder

KV = """
#: import LinearGradient bouquet.gradients.LinearGradient
Widget:
    canvas:
        Color:
            rgb: 1.0, 1.0, 1.0
        Ellipse:
            pos: self.pos
            size: self.size
            texture: LinearGradient.render_texture(size=self.size)
"""

class MyApp(App):

    def build(self):
        return Builder.load_string(KV)

MyApp().run()

```

RadialGradient.render_texture():

```

from kivy.app import App
from kivy.graphics import Ellipse
from kivy.uix.widget import Widget

from bouquet.gradients import RadialGradient

class EllipseWithRadialGradient(Widget):

    def __init__(self, **kwargs):
        super().__init__(**kwargs)
        self.center_color = "#ff0000" # red
        self.border_color = "#ffff00" # yellow
        with self.canvas:
            self.ellipse = Ellipse(
                pos=self.pos,
                size=self.size,

```

(continues on next page)

(continued from previous page)

```

        texture=RadialGradient.render_texture(
            size=self.size,
            center_color=self.center_color,
            border_color=self.border_color
        )
    )

def on_size(self, widget, size):
    self.ellipse.size = size
    self.ellipse.texture = RadialGradient.render_texture(
        size=self.size,
        center_color=self.center_color,
        border_color=self.border_color
    )

class MyApp(App):

    def build(self):
        return EllipseWithRadialGradient()

MyApp().run()

```

class bouquet.gradients.**LinearGradient**(**kwargs)

Bases: [AnchorLayout](#)

Widget for creating a linear gradient background with customizable colors for each corner.

Hint: [LinearGradient](#) is an [AnchorLayout](#) subclass, so you can put any widget inside it.

bottom_left_color

Color of the top bottom corner of gradient.

[bottom_left_color](#) is an [ColorProperty](#) and defaults to *black*.

bottom_right_color

Color of the bottom right corner of gradient.

[bottom_right_color](#) is an [ColorProperty](#) and defaults to *red*.

static render_texture(**kwargs) → [Texture](#)

Renders gradient at FBO and returns the texture.

Parameters

kwargs – Any [LinearGradient](#) properties.

top_left_color

Color of the top left corner of gradient.

[top_left_color](#) is an [ColorProperty](#) and defaults to *green*.

top_right_color

Color of the top right corner of gradient.

[top_right_color](#) is an [ColorProperty](#) and defaults to *yellow*.

class bouquet.gradients.RadialGradient(**kwargs)

Bases: Widget

Widget for creating a radial gradient effect, in which color is linearly interpolated from the center to the edges of the widget.

Note: Currently, it is possible to set only two colors: the color of the center (*center_color*) and the color of the border (*border_color*).

border_color

Color of the widget borders.

border_color is an *ColorProperty* and defaults to *black*.

center_color

Color of the widget center.

center_color is an *ColorProperty* and defaults to *white*.

static render_texture(**kwargs) → Texture

Renders gradient at FBO and returns the texture.

Parameters

kwargs – Any *RadialGradient* properties.

INDICES AND TABLES

- `modindex`
- `search`

PYTHON MODULE INDEX

b

bouquet, ??

bouquet.gradients, 3

INDEX

B

`border_color` (*bouquet.gradients.RadialGradient* attribute), 6

`bottom_left_color` (*bouquet.gradients.LinearGradient* attribute), 5

`bottom_right_color` (*bouquet.gradients.LinearGradient* attribute), 5

`bouquet`
module, 1

`bouquet.gradients`
module, 3

C

`center_color` (*bouquet.gradients.RadialGradient* attribute), 6

L

`LinearGradient` (class in *bouquet.gradients*), 5

M

module
 bouquet, 1
 bouquet.gradients, 3

R

`RadialGradient` (class in *bouquet.gradients*), 5

`render_texture()` (*bouquet.gradients.LinearGradient* static method), 5

`render_texture()` (*bouquet.gradients.RadialGradient* static method), 6

T

`top_left_color` (*bouquet.gradients.LinearGradient* attribute), 5

`top_right_color` (*bouquet.gradients.LinearGradient* attribute), 5