

Philips Hue CLI-Controller

Projektpräsentation

Eduard Belan

Fachinformatiker für Anwendungsentwicklung

Table of contents

1 Overview

2 Prior Knowledge

- External Module: colorgram
- Extractor Module
- JSON Data

3 Random Hues in Detail

- Part 1: get_random_picks
- Part 2: prepare_hue_for_light_input

4 Set Light Scene in Detail

- Set Light Scene Algorithm
- Light Scene Algorithm Explained

5 Summary

Table of contents

1 Overview

2 Prior Knowledge

- External Module: colorgram
- Extractor Module
- JSON Data

3 Random Hues in Detail

- Part 1: get_random_picks
- Part 2: prepare_hue_for_light_input

4 Set Light Scene in Detail

- Set Light Scene Algorithm
- Light Scene Algorithm Explained

5 Summary

External Module: colorgram



```
colorgram.extract('test.jpg', 10)
```



Extractor Module

Extractor Module

```
def extractor(picture_name, number_of_colors):  
    """  
    Returns a List of Tuples with RGB-Values.  
    Params: picture_name = Full Picture Name with Extension  
    Params: number_of_colors = Number of Colors to extract  
    """  
  
    extracted_colors = []  
    raw_extract = colorgram.extract(picture_name, number_of_colors)  
    for color in raw_extract:  
        rgb_tuple = (color.rgb.r, color.rgb.g, color.rgb.b)  
        extracted_colors.append(rgb_tuple)  
    return extracted_colors
```

Extractor Module

```
def extractor(picture_name, number_of_colors):  
    """  
    Returns a List of Tuples with RGB-Values.  
    Params: picture_name = Full Picture Name with Extension  
    Params: number_of_colors = Number of Colors to extract  
    """  
  
    extracted_colors = []  
    raw_extract = colorgram.extract(picture_name, number_of_colors)  
    for color in raw_extract:  
        rgb_tuple = (color.rgb.r, color.rgb.g, color.rgb.b)  
        extracted_colors.append(rgb_tuple)  
    return extracted_colors
```

Example (Output)

```
raw_extract:      [<colorgram.py Color: Rgb(r=70, g=27, b=11), 39.87431390312046%>, <colorgram.py Color: Rgb(r=153, g=77, b=33), 30.315014578369205%>]  
extracted_colors: [(70, 27, 11), (153, 77, 33), (234, 142, 62), (131, 36, 14), (251, 217, 93), (252, 243, 181), (221, 91, 47), (214, 127, 27), (111, 5
```

JSON Data

JSON Data

Data Sample

```
{  
  "christmas": [  
    [  
      70,  
      27,  
      11  
    ],  
    [  
      153,  
      77,  
      33  
    ],  
    [  
      234,  
      142,  
      62  
    ],  
  ],  
}
```

JSON Data

Data Sample

```
{
  "christmas": [
    [
      70,
      27,
      11
    ],
    [
      153,
      77,
      33
    ],
    [
      234,
      142,
      62
    ]
  ],

```

Data Sample

```
],
[
  201,
  141,
  103
],
"themartian": [
  [
    147,
    88,
    31
  ],
  [
    61,
    36,
    6
  ]
],

```

Table of contents

1 Overview

2 Prior Knowledge

- External Module: colorgram
- Extractor Module
- JSON Data

3 Random Hues in Detail

- Part 1: get_random_picks
- Part 2: prepare_hue_for_light_input

4 Set Light Scene in Detail

- Set Light Scene Algorithm
- Light Scene Algorithm Explained

5 Summary

Random Hues in Detail

Part 1: get_random_picks

Random Hues in Detail

Part 1: get_random_picks

```
def get_random_picks(scene_name):  
    from scenes import extractor  
    get_data = extractor.read_data("scenes\\colors.json")  
    get_rgb_values_from_data = get_data[scene_name]  
    random_picks = random.sample(get_rgb_values_from_data, 5)  
    return random_picks
```

Example (Output)

[[r, g, b], [r, g, b], [r, g, b], [r, g, b], [r, g, b]]

Random Hues in Detail

Part 2: prepare_hue_for_light_input

Random Hues in Detail

Part 2: prepare_hue_for_light_input

```
def prepare_hue_for_light_input(scene_name):  
    picks = get_random_picks(scene_name)  
    prepped_hues = []  
    for i in picks:  
        prepped_hues.append(rgb_to_hue_saturation(i))  
    return prepped_hues
```

Example (Input & Output)

Input = [[r, g, b], [r, g, b], [r, g, b], [r, g, b], [r, g, b]]

Output = [(h, s), (h, s), (h, s), (h, s), (h, s)]

Table of contents

1 Overview

2 Prior Knowledge

- External Module: colorgram
- Extractor Module
- JSON Data

3 Random Hues in Detail

- Part 1: get_random_picks
- Part 2: prepare_hue_for_light_input

4 Set Light Scene in Detail

- Set Light Scene Algorithm
- Light Scene Algorithm Explained

5 Summary

Set Light Scene in Detail

```
def set_light_scene_random_hues(scene_name):  
    prepped_hues = prepare_hue_for_light_input(scene_name)  
    lights = b.get_light_objects("id")  
    for i in lights:  
        light = b.get_light(i)  
        if not light["state"]["on"]:  
            b.set_light(i, "on", True)  
        if i == 1:  
            lights[i].brightness = 50  
        else:  
            lights[i].brightness = 125  
        i_list_item = 0  
        lights[i].hue = prepped_hues[i - 1][i_list_item]  
        i_list_item += 1  
        lights[i].saturation = prepped_hues[i - 1][i_list_item]
```

Light Scene Algorithm Explained

For-Loop in Detail

```
for i in lights:
    i_list_item = 0
    lights[i].hue = prepped_hues[i - 1][i_list_item]
    i_list_item += 1
    lights[i].saturation = prepped_hues[i - 1][i_list_item]
```

Light Scene Algorithm Explained

For-Loop in Detail

```
for i in lights:  
    i_list_item = 0  
    lights[i].hue = prepped_hues[i - 1][i_list_item]  
    i_list_item += 1  
    lights[i].saturation = prepped_hues[i - 1][i_list_item]
```

Iterating over prepped_hues

[(h, s), (h, s), (h, s), (h, s), (h, s)]

Light Scene Algorithm Explained

For-Loop in Detail

```
for i in lights:  
    i_list_item = 0  
    lights[i].hue = prepped_hues[i - 1][i_list_item]  
    i_list_item += 1  
    lights[i].saturation = prepped_hues[i - 1][i_list_item]
```

Iterating over prepped_hues

```
[(h, s), (h, s), (h, s), (h, s), (h, s)]  
[(0, 1),
```

Light Scene Algorithm Explained

For-Loop in Detail

```
for i in lights:  
    i_list_item = 0  
    lights[i].hue = prepped_hues[i - 1][i_list_item]  
    i_list_item += 1  
    lights[i].saturation = prepped_hues[i - 1][i_list_item]
```

Iterating over prepped_hues

```
[(h, s), (h, s), (h, s), (h, s), (h, s)]  
[(0, 1), (0, 1), (0, 1), (0, 1), (0, 1)]
```

Table of contents

1 Overview

2 Prior Knowledge

- External Module: colorgram
- Extractor Module
- JSON Data

3 Random Hues in Detail

- Part 1: get_random_picks
- Part 2: prepare_hue_for_light_input

4 Set Light Scene in Detail

- Set Light Scene Algorithm
- Light Scene Algorithm Explained

5 Summary

Summary

- 1 Get a Raw Extract of Colors with the colorgram Module.
 - colorgram-Module

Summary

- 1 Get a Raw Extract of Colors with the colorgram Module.
 - colorgram-Module
- 2 Extract the separate Color Values and add to a List.
 - extractor-Module

Summary

- 1 Get a Raw Extract of Colors with the colorgram Module.
 - colorgram-Module
- 2 Extract the separate Color Values and add to a List.
 - extractor-Module
- 3 Store the Color Values with a Scene Name in a JSON File.
 - dict-Format

Summary

- 1 Get a Raw Extract of Colors with the colorgram Module.
 - colorgram-Module
- 2 Extract the separate Color Values and add to a List.
 - extractor-Module
- 3 Store the Color Values with a Scene Name in a JSON File.
 - dict-Format
- 4 Get Random RGB Samples from the JSON File.
 - random-Module, sample-Method
 - Get as many Random Samples as you have Lights.

Summary

- 1 Get a Raw Extract of Colors with the colorgram Module.
 - colorgram-Module
- 2 Extract the separate Color Values and add to a List.
 - extractor-Module
- 3 Store the Color Values with a Scene Name in a JSON File.
 - dict-Format
- 4 Get Random RGB Samples from the JSON File.
 - random-Module, sample-Method
 - Get as many Random Samples as you have Lights.
- 5 Convert the Samples to Hue and Saturation Values.
 - colorsys-Module, rgb_to_hsl-Method

Summary

- 1 Get a Raw Extract of Colors with the colorgram Module.
 - colorgram-Module
- 2 Extract the separate Color Values and add to a List.
 - extractor-Module
- 3 Store the Color Values with a Scene Name in a JSON File.
 - dict-Format
- 4 Get Random RGB Samples from the JSON File.
 - random-Module, sample-Method
 - Get as many Random Samples as you have Lights.
- 5 Convert the Samples to Hue and Saturation Values.
 - colorsys-Module, rgb_to_hsl-Method
- 6 Distribute the prepped Samples to all Hue Lights.
 - set_light_scene_random_hues-Method

Summary

- 1 Get a Raw Extract of Colors with the colorgram Module.
 - colorgram-Module
- 2 Extract the separate Color Values and add to a List.
 - extractor-Module
- 3 Store the Color Values with a Scene Name in a JSON File.
 - dict-Format
- 4 Get Random RGB Samples from the JSON File.
 - random-Module, sample-Method
 - Get as many Random Samples as you have Lights.
- 5 Convert the Samples to Hue and Saturation Values.
 - colorsys-Module, rgb_to_hsl-Method
- 6 Distribute the prepped Samples to all Hue Lights.
 - set_light_scene_random_hues-Method

The End