# 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

LUTZ+ GRUB

## **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**





### **Extractor Module**



### **Extractor Module**

```
def extractor(picture name, number of colors):
    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):
    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)



## **JSON Data**

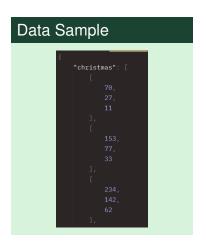


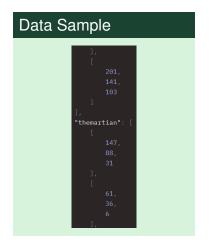
## **JSON Data**

```
Data Sample
         "christmas":
```



## **JSON Data**







## **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



Part 1: get\_random\_picks



#### 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]]
```



Part 2: prepare\_hue\_for\_light\_input



#### 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)
            lights[i].brightness = 50
            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]
```



#### 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]
```



#### 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)]
```



#### 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),
```



#### 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



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



- **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



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



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



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



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



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



