

# Harmonize

## User Documentation

### In this document

|  |   |
|--|---|
| About.....                               | 1 |
| Usage.....                               | 1 |
| First launch.....                        | 1 |
| Configure Hue.....                       | 3 |
| Troubleshooting.....                     | 4 |
| Where do I find the logfile?.....        | 4 |
| Nothing is responding to Streamlabs..... | 4 |
| Changelog.....                           | 5 |

### About

Harmonize is mainly aimed at music streamers, but might have features others enjoy as well.

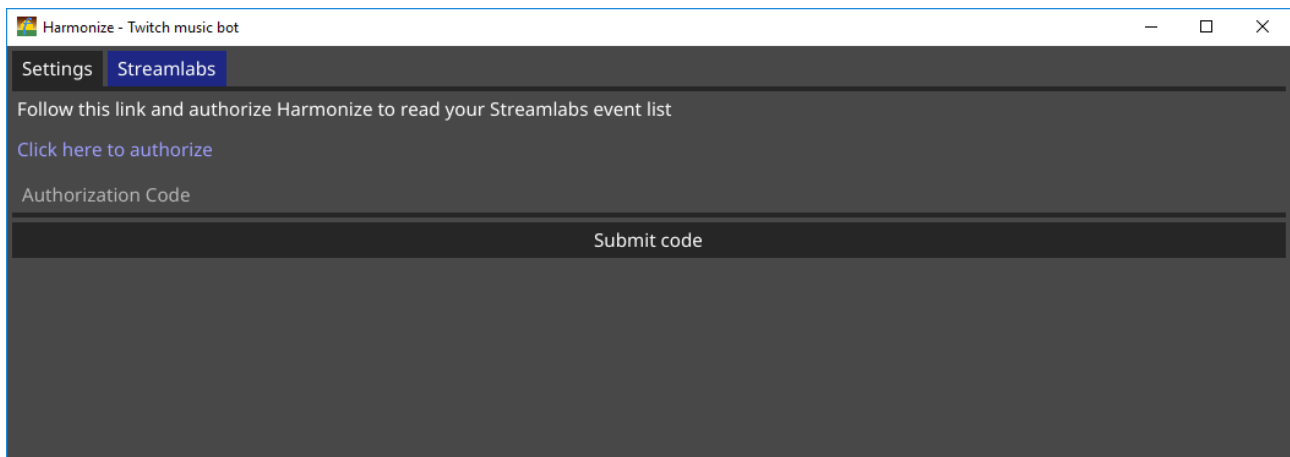
### Usage

Unpack the zip in a directory of your choice. The zip should contain *harmonize.exe*, *harmonize.license* and a directory called *web*.

When you launch *harmonize.exe*, Windows will probably warn you about running an unrecognized app, in a later version I hope to add proper Windows licensing, but for now you'll have to click the 'More info' and ignore the warning.

### First launch

On first launch it should bring you to a screen like this:



This is to authorize the app, so it can access your Streamlabs account.

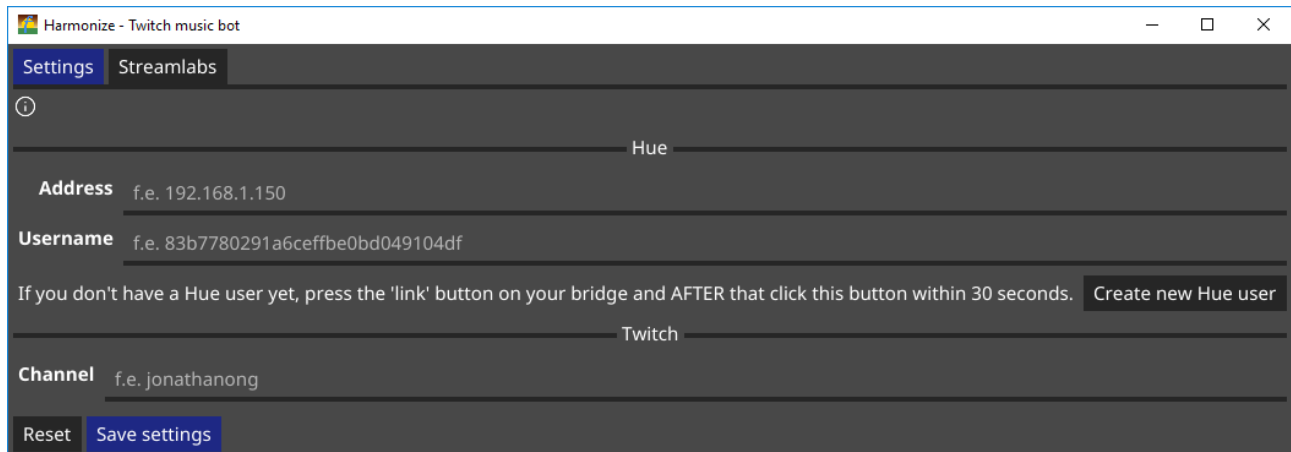
1. Click *Click here to authorize*. You'll be brought to streamlabs
2. If needed: login
3. Streamlabs will ask if you want to authorize Harmonize
4. You'll end up on an ugly page with a big authorization code
5. Paste the authorization code in the window of Harmonize where it says *Authorization Code*
6. Submit

You have now succesfully authorized Harmonize to listen to donation and bits events!

# Hue integration

## Configuring the connection

Click on the *Settings* tab. You'll find a window like this:



Harmonize - Twitch music bot

Settings Streamlabs

Hue

Address f.e. 192.168.1.150

Username f.e. 83b7780291a6ceffbe0bd049104df

If you don't have a Hue user yet, press the 'link' button on your bridge and AFTER that click this button within 30 seconds. Create new Hue user

Twitch

Channel f.e. jonathanong

Reset Save settings

Here you can configure the Hue connection. The app should be running in the same network as Hue for this to work!

Fill in IP address of your Hue bridge, if you don't know it, you should be able to find it here: <https://www.meethue.com/api/nupnp>

You probably don't have a username yet, so **after** you filled in the IP address, press the *link* button on your bridge and click the *Create new Hue user* button.

Don't forget to click *Save settings*!

You are now ready to test it out!

## Configuring the Hue rules

The app should have opened a webpage for you (if not, open <http://localhost:9000> in a browser). I hope it mainly speaks for itself.

# Troubleshooting

## Where do I find the logfile?

In case things don't go as planned, please send me the logfile which can be found at: `%AppData%\Roaming\wjzijderveld\harmonize\harmonize.log`.

To get to *AppData*, Simple click the address bar in Explorer and type `%appdata%` where you will find the *Roaming* folder.

## Nothing is responding to Streamlabs

Although I don't expect this to happen anymore, we somehow lost access to Streamlabs. To fix this we need to reauthorize to Streamlabs, in the future this will be done semi-automatic. For now remove the file `%AppData%\Roaming\wjzijderveld\harmonize\streamlabs-accessToken.json`.

To get to *AppData*, Simple click the address bar in Explorer and type `%appdata%` where you will find the *Roaming* folder.

# Changelog

## **Version 0.1.0**

- Connect to Streamlabs and listen to donation events
- Connects to the Hue API
- A GUI to configure the connections
- When a donation happens (works fine with the test donations) the lights in **Group 1** (hardcoded for now) is set to a random color.

## **Version 0.1.2**

- First version of a web-interface to configure rules
- Added possibility to choose a color for a specific lamp on donation

## **Version 0.2.0**

- Completely redid the web interface to make more sense (hopefully)
- Added option to use a random color
- Added option to change an entire group
- Added support for bits

## **Version 0.2.1** - *bugfix release*

## **Version 0.2.2**

- Added possibility to test an action directly from the interface
- Increased logging output

## **Version 0.3.0**

- Added possibility to revert the effect after a specified interval.  
**NOTE** How this works, is that it checks the current state before it applies the effect, and reverts back to that after the specified time. But this could have unexpected consequences<sup>1</sup>.

---

<sup>1</sup> See [Example](#)

## Unexpected reverting example

Lights 1,2, and 3 are white. Rule **A** triggers light 2 to **blue** to revert after 2 seconds. Rule **B** is triggered to set all lights to **red** and revert after 2 seconds. 2 seconds after rule **A** is triggered, light 2 is reverted to **white**, but 2 seconds after **B** is triggered light 2 is reverted to **blue**, because that's what the color of the light was when it was triggered.

Shown in a table to see what happens in every step. **Bold** colors are the colors that changed in that second.

| Second | Rule triggered | Light 1      | Light 2      | Light 3      |
|--------|----------------|--------------|--------------|--------------|
| 0      |                | white        | white        | white        |
| 1      | A              | white        | <b>blue</b>  | white        |
| 2      | B              | <b>red</b>   | <b>red</b>   | <b>red</b>   |
| 3      | A (revert)     | red          | <b>white</b> | red          |
| 4      | B (revert)     | <b>white</b> | <b>blue</b>  | <b>white</b> |