NAME

blight – GUI frontend to busylight

SYNOPSIS

[**wish8.6**] **blight.tcl**

DESCRIPTION

The **blight.tcl** script is a simple GUI front-end to the **busylight**(1) program. It provides an on-screen indication of the light status (helpful if you can't physically see the LEDs) and a row of buttons to click on to easily change the light status.

All of its operations are carried out via the rbserver**(1)** web service API.

Every 5 minutes, it will query the state of the lights and update its on-screen indicator accordingly, but it will also do this when you manually change the status or click the **refresh** button.

Buttons are provided for each of the defined status values in your **~/.busylight/config.json** file (except the **start** and **stop** statuses).

ACTIVITIES

Blight adds another feature not otherwise provided by **busylight**. It allows for activity tracking. To use this feature, you will need to create a file called **~/.busylight/activities.json** which contains a JSON array of objects, each of which has the following fields:

Name This should be a short (preferably one-word) name of the activity to be tracked.

Lights A string of light codes to light up when this status should be shown. Note that Busylight units cannot illuminate more than one light at once, so this should normally be a single light code (but you can accent it by combining with a strobe pattern or use a flash sequence instead); Readerboards can illuminate any combination of lights at once.

If the string of light codes begins with a star ("*"), then the codes are taken to be a flash sequence (each light in the list is lit up in turn).

Strober A string of light codes which are set as the strober sequence on the status display, in addition to the steady-burning or flashing lights.

Unless A list of light code strings. If any of them match the set of illuminated lights currently displayed on the unit, then this status will not be set because what's already on the display is a higher-priority status. If a string begins with a star ("*"), then it matches the current flash sequence. If a string begins with two stars ("**"), then it matches the current strober sequence.

Elapsed An integer value giving the number of minutes spent so far on that activity.

An example **activities.json** file might look like this:

```
[
 {"Name":"Meetings", "Status":"R", "Unless":["*Rr"], "Elapsed":0},
 {"Name":"Idle", "Status":"G", "Unless":["Y","R","*Rr"], "Elapsed":12},
 {"Name":"Games", "Status":"Y", "Strober":"G", "Elapsed":120}
]
```

The **blight** script will update this file with new **Elapsed** values while activities are active.

To start an activity, click on its button. That will set the lights for the activity and start the timer to track time for that activity. Clicking other buttons will change lights but not change the running timer. Clicking another activity will stop the current one before starting the new activity. Clicking the **(stop activity)** button will stop the activity timer without starting a new one.

AUTHOR

Steve Willoughby <*steve@madscience.zone*>