

## plot Design

`plot` gets `greenerthumb` JSON messages from STDIN (5).

## UI

`plot` will convert input messages to points on a line-graph. The expected message format is:

```
{
  "Name": <message_name>,
  "Timestamp": <timestamp>,
  <name>: <value>, ...
}
```

Messages don't necessarily need to be `greenerthumb` messages. They just need to fit this format.

Each non-ID and non-timestamp field will become a line in the graph. The name of the line will be determined by concatenating the `<message_name>` and the field's `<name>`. The line's will be assigned unique random colors which will be displayed in a legend with the message name's on the right side of the plot.

Each line will be overlayed to allow trend comparison. To do this, each line will have units normalized to each other. The x-axis will have units of hours scaled to the period of all the received messages. Ranges of units are presented in the legend to account for the normalization.

If received messages have the same timestamp, the newest message will overwrite the older messages.

A save button makes screenshots.

`plot` only closes once commanded to close instead of once STDIN is closed.

## Performance

`plot` is expected to be able to render 2-weeks worth of 5 kinds of data at a sample rate of 1 instance per second in less than 1 second per frame.