

OC_manager specification

Dimitri Fichou

February 21, 2017

Introduction

Document to list the specification needed for OC_manager. So far, we have a prototype that is working but it's just a prototype. We must make it so it can evolve to integrate other functionality and step of the OC pipeline.

Current state

Software is written in R shiny, could be better in Python but hey, R is also interesting as it is the data analysis language and we are producing data. So we can see the OC as a data producing tool which transform the samples into analytical result. We use the shinydashboard package, it allows to split the steps easily even though at the end, all steps must be together.

The software is made to be hosted on a raspberry pi which in turn communicate to the arduino. It also works great on a laptop but is harder to install, advantage is about the computing time but the pi can do it if functions are optimize.

The steps are produced on the fly in the software and any order can be created with any number of step, so far, they can be save and reload.

To create a Gcode (that can be executed already), the user, select a step, access an option table, update the step to make a plot and the Gcode. There is a few bug but it works. The Gcode creation and plot creation is supported by functions that take the option table(s) as input.

The full method is stored in a list with steps as elements. Each step is also a list with several element from title to option table.

The method are stored in a folder **methods**. There is a **tables** folder where the default option table are stored. A **gcode** folder to store the gcode files that are then executed, we will have a problem of memory here. There is a **log** folder that contain a csv file where every actions are recorded, it could use a bit of cleaning. The server call the **server_X.R** files that in turn give a lot of renderUI. There is a **config.R** file. There is a **login.csv** file to put user login and passwords, which enable protection and in particular better log for GMP.

Next step

Put the functions directly in the step list. and take the step list as argument of the function. this way, change are easier to implement and stuff that works keep working.

Clean the log file append.

Stop button in method.

Make check box, select and text input in option tables, test rhandsontable

put origin and position bias somewhere (config.R ???)