One Step Closer to Version 1.0

Rohit Das

***Email ID**: rohit.das950@gmail.com*

***Freenode IRC Nick**: mouri11*

***Location**: Serampore, West Bengal, India UTC+5:30*

---

1. Motivation

GCompris is a software suite comprising educational software for children aged 2 to 10. Protected under the GNU General Public License, GCompris is free for developers to work with and contribute to. Having more than 137 activities, developers old and new can have their hands full with fun things to play around. Hence, GCompris is not only educational for the kids, but also for their creators.

Teaching kids and making them more imaginative is what GCompris aims at, and I intend to help with the process of its development. Since GCompris was originally written in GTK+, there are many activities to work with, from porting old ones, to completing and creating activities anew in Qt-Quick. The activity I intend to work on is: *Analog Electricity*.

2. Project Goals

I wish to opt for porting Analog Electricity from scratch from its Gtk+ version for my intended work as a student for Google Summer of Code. By the end of the period, I intend to achieve the following:

- **Analog Electricity activity:** This activity is to be started from scratch, and discussions on this can be found at: https://phabricator.kde.org/T5954.

3. Implementation Details

3.1. Analog Electricity

Analog Electricity, as the name suggests is aimed at teaching children the basics of analog electric circuits, the various components involved in a circuit, how electricity flows and the likes.

Main Goals:

■ **Creating separate modes of learning and free play:**

- ☐ The Gtk+ version simply has a free play mode, where children can simply play around with the components, make a circuit, and complete a level when the bulb glows. This could be a bit counter-intuitive, as a child wouldnt know how to construct a basic circuit, what each newer components and their functions are and how to use them, etc.

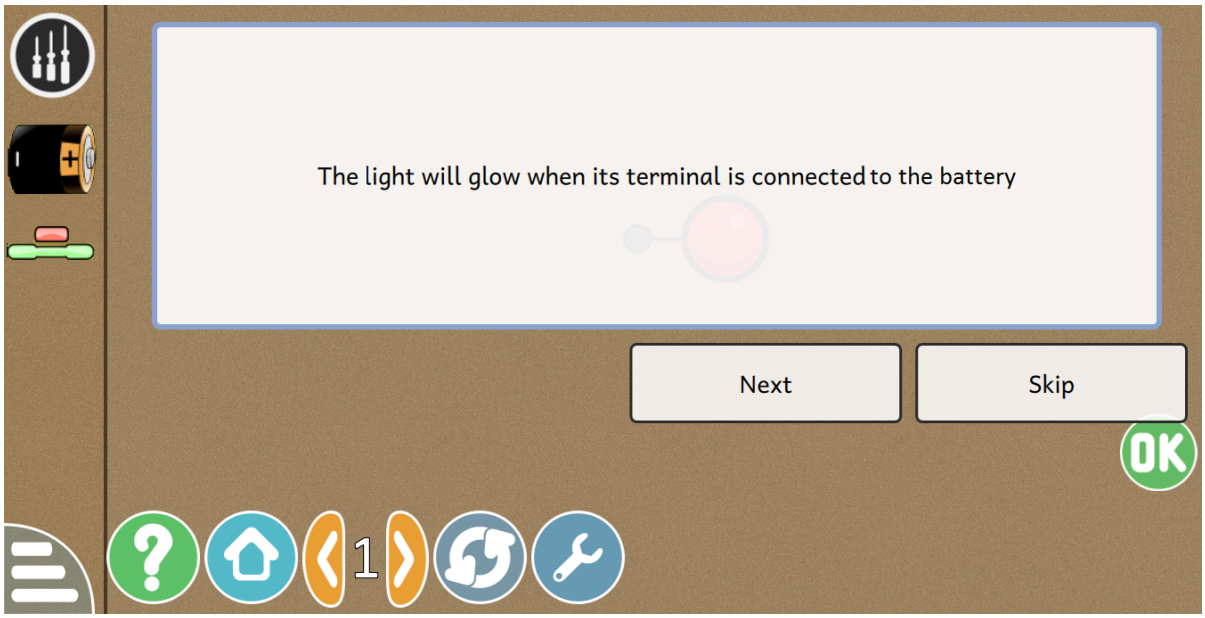

Figure 1: Mockup for possible layout

- ☐ So, two separate modes shall be created, where in one, i.e. the tutorial mode, the child can be guided through each tutorial, teaching him/her the basics of analog circuits and components. In the free play mode, the child can then use their new-found knowledge to construct circuits. Some ideas on the modes are:
  - In the initial levels, children will be introduced to the two sources of current, and how can they observe and relate to it in real life. For a DC source, a simple dry cell will be used, and for AC, a plug analogous to ones in our homes can be used.
  - They will use the above sources in a circuit and see how it works. Separate levels shall be dedicated initially to teaching how the sources work and how they differ (explained in next point). The arrows showing direction of current shall be very helpful in this case. In later levels, both sources shall be kept(in separate levels) to teach them how components work with each source.
  - To teach the differences between the sources, waveforms will be helpful to show the physical difference. The arrows, as mentioned above, shall show how the direction changes with time in AC, but not in DC. The origin of the sources, i.e., dry cells for DC and power plugs for AC will also be helpful for children to distinguish between them.
  - In higher levels, as components like capacitors, solenoids are introduced, they can be taught about what to expect when connected with the two sources. The whys and hows of it can be left out since it will need a lot more than primary school knowledge to understand.
  - For components utilising magnetic properties of current like solenoids, children will be taught about the relation between current and magnetism(tutorial on magnetism explained later).
  - The free mode will contain all of the components discussed here for free implementation, experimentation and learning of the child. Levels wont be necessary here.

* In free mode, for showing the resistance, current and voltage flowing across circuits, Kirchoffs law(for DC) will be used. All the components will be iterated through before calculating individual voltages and currents. Details of the calculation will be beyond the comprehension of the children, and hence hidden from users.
* For solving complicated mathematical equations to obtain current and voltages across circuits(complex numbers for AC), QtMath (http://doc.qt.io/qt-5/qtmath.html) header can be used for basic exponential, logarithmic and trigonometric calculations. For more complex calculations (and matrix manipulations if necessary), math.js (http://mathjs.org/) is a good alternative.
* A resistor can be explained analogous to a water tap, which can restrict/allow water flow.
* Capacitors can be explained as components having a capacity to hold current, and how they work with DC and AC. Capacitors shall be made to be used with a dry cell as well as an AC source(a plug) and a resistor in tutorial mode, and the child will be free to experiment in free play.
* For teaching magnetism and why it occurs in current-carrying components, a superficial explanation will be quite helpful. In the tutorials, a magnet can be depicted to be attracting iron nails kept nearby. Children will be encouraged to try out using small magnets and iron objects(under adult supervision of course).
* A solenoid shall be implemented in the same level as above-mentioned tutorial, and hence would need some graphical cues, like when a solenoid is switched on, some nearby iron nails will be animated to move(attracted) towards it, and stick to it, thus displaying magnetic behaviour. The solenoid is being introduced only to teach that current can produce magnetism, and its applications in real life.

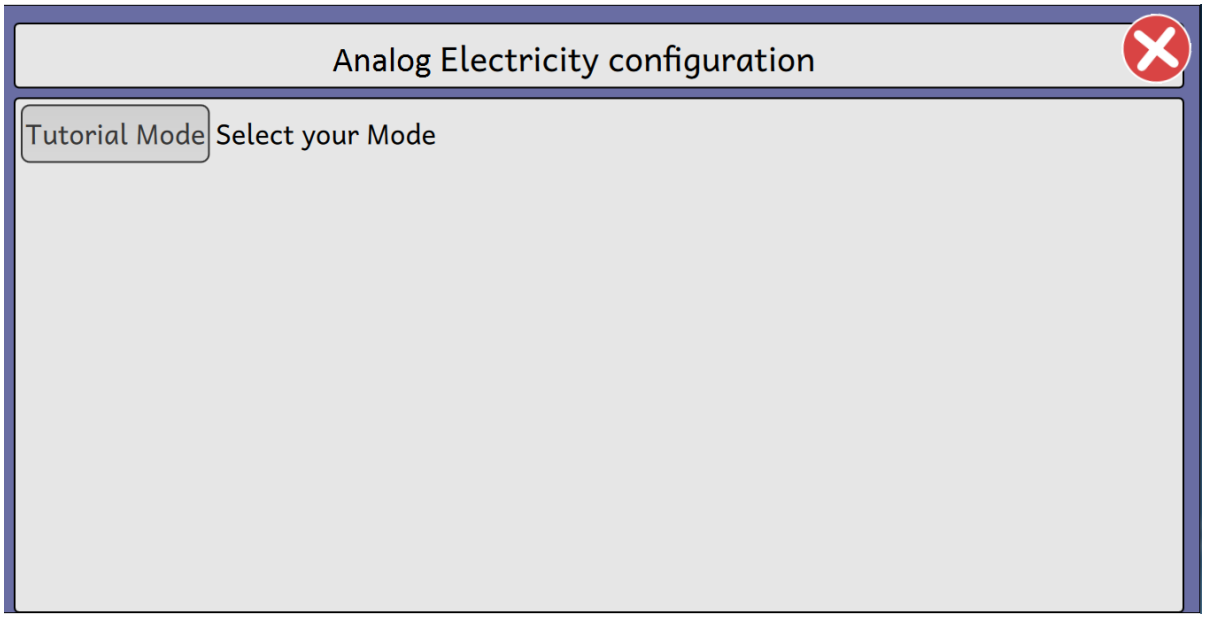

Figure 2: Mockup for modes

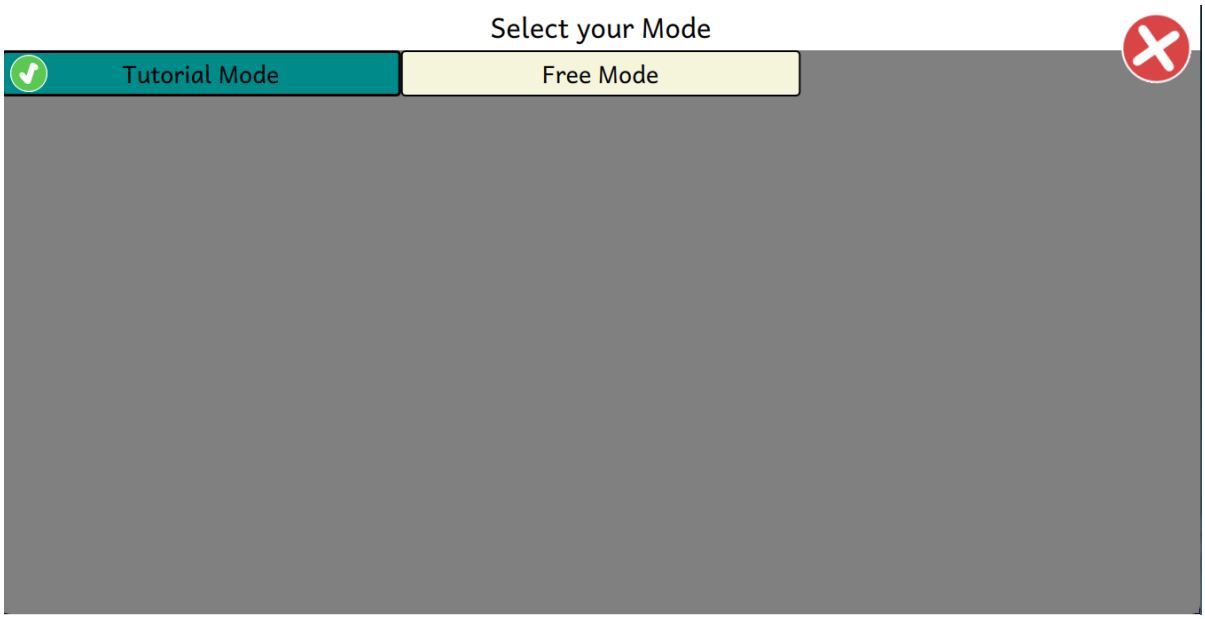

Figure 3: Mockup for choice between modes

■ **Simulator for analog components and circuits:**

- ☐ A sandbox for making circuits and testing out components will be created, which will enable adding/removing components, connecting various circuits and testing them out.
- ☐ Components will be modeled as closely as possible to represent how they actually look and work in real life. Unlike digital electronic components, which work on discrete values, analog components shall be modeled to work with any voltage values and give correct or close to correct results. Clicking on them will show their names, and tutorials shall show how they work.

Figure 4: Mockup showing pinchArea and components

☐ The external library, available at https://github.com/Aerlinger/maxwell, although implements various analog components, cannot be directly used here, as Qt Quick does not support js classes. So this library shall be solely used for technical lookups and taking ideas on how to better implement components. Most of the ideas below have been formulated after a thorough observation of the repository.

* Since js classes arent available in Qt, each component will be implemented with its own separate js file, and exported as a module to be used in circuits.
* A base circuit component file shall be shared among all other components, containing basic features like resistance, voltage, etc. and functions to calculate the values of the attributes, as well as rendering them on screen.
* The interface shall be similar to digital electricity, easy for children to interact and understand.
* For the oscilloscope, Qt Charts can be used(optional: may be configured to use OpenGL for rendering waveforms in OpenGL devices). The other options are:
  - Using smoothie (http://smoothiecharts.org/)(js charting library)
  - Using d3.js (https://d3js.org/) along with Qml to render graphs(the external library uses d3.js and rickshaw (https://www.npmjs.com/package/rickshaw), relying on d3.js).
* As in the external library, we can use marquee text to denote the direction of flow of current, with different colours for various directions of current flow, or determine the direction of current by the polarity of the source, and assign simple arrows at the ends of connections(later option looks better).

☐ Lastly, all of the components(js files, including wires) shall be brought together using Qml.

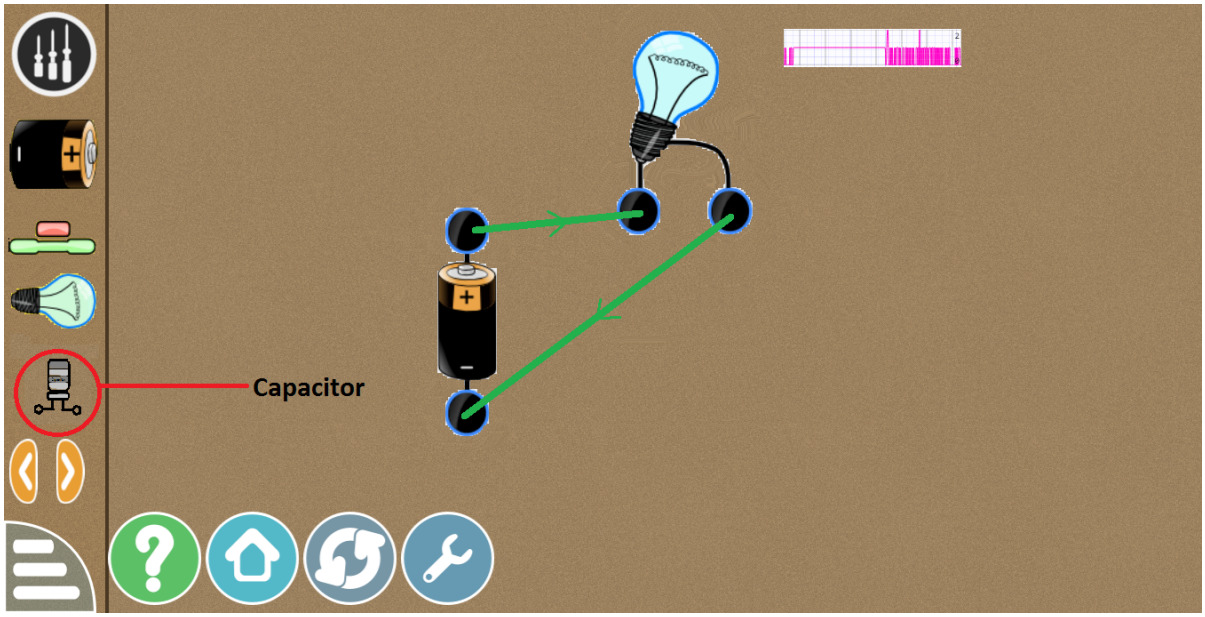

Figure 5: Mockup showing circuits with current direction and waveform

■ **Add more components:**

- ☐ Components like capacitors and solenoids will be added for improved and advanced knowledge on important analog components.
- ☐ Addition of components like capacitors, etc. may also be made interesting by adding a simple pre-adjusted oscilloscope(since the children will need a separate tutorial to handle the controls) to show how the waves of an analog signal varies, and a speaker can be added for audio to show how waves sound, with a volume slider.

■ **Improvements in UI:**

- ☐ For the layout, keeping it similar to that of digital electricity will help children easily understand how both are or arent related and help them differentiate between them. The dark background will also help in clearer depiction of images and text.
- ☐ Displaying voltage, current in case of power source and bulbs, and resistance as well in case of resistors, etc. under the components rather than on them.
- ☐ Using a darker background to enable better legibility of text displayed.
- ☐ Wires to show arrows to display direction of current flow(helpful in case of capacitors, etc.)
- ☐ To prevent cluttering of various components as levels increase, pinchArea shall be implemented to enable navigation among various components in free mode.

■ **Increasing difficulty with levels:**

- ☐ Levels shall be presented in tutorial mode only, so that children can get help to progressively learn the various components and their functions. In free mode, all components shall be available for practice and experimentation.
- ☐ Components mentioned earlier will be introduced into the activity at a much later stage, since these will require a much thorough knowledge even for using in a basic circuit.

4. Timeline

Time Period	**Activity for the Period**
April 24 - August 14, 2018	*Analog Electricity*
April 24 - May 13	*Community Bonding*
	• Bond with the mentors, go through further documentation and look up useful libraries. • Take notes and suggestions from mentors and bond with other GSoC candidates.

May 14 - June 12	*Development of Basic Layout and Simulator*
May 14 - May 20	• Build the basic main layout and layouts for tutorial and free mode, similar to digital electricity, without components(generalize code between activities as much as possible)
May 21 - May 24	• Submit work for review • Add improvements based on suggestions
May 27 - June 10	• Test out codes from external libraries, and start implementing basic components without classes – Using json files to depict various features of components like voltage, resistance,etc will make it easier to handle and change while development and testing(much like in the external library) – Functioning of components will be better handled by separate js files for each type of component(initially implementing simple ones like resistors) – Basic components like bulbs, small resistors, switches, potentiometers and DC sources shall be implemented, using simple circuits.
June 11 - June 12	• Submit work for possible improvements before final submission for phase 1 review

June 13 - June 15	*Submission and Review - Phase I*
	• Submit work for review by mentors • Make changes as suggested in reviews • Submit work for final evaluation of Phase I

June 16 - July 10	*Addition of tutorials and advanced components*
June 16 - June 25	• Work on developing tutorials easy enough to be understood by children. The tutorials shall be built first so as to give an insight on how to proceed to make future components next. – Tutorials for resistors shall consist of pictures depicting what and why of a resistor, followed by how to use them, and analogies to taps being used to restrict water flow similar to resistors restricting current flow. – Concepts of serial and parallel circuits shall be introduced next. Levels shall be particularly allotted for this purpose. – For capacitors, children shall be taught what a capacitor does, i.e. store electricity, and hence how it is different from a battery. A simple circuit can be depicted in images, consisting of a bulb which will slowly dim out on disconnecting from a DC cell, as it will be connected to a capacitor in parallel.

	– For solenoids, a tutorial level will be included where magnetism will be exhibited as follows: a solenoid connected with a battery will be shown. An image of iron nails kept nearby shall be animated to move nearer the solenoid, depicting attraction. In the tutorial, a brief description of a magnet and why it attracts only certain materials.
June 26 - July 1	• Submit above work for review • Discuss possible improvements and fixes
July 2 - July 9	• Work on modelling additional components like capacitors as well as increase levels and their difficulty. – Start working on AC source and capacitors(partial implementation)
July 10 - July 13	*Submission and Review - Phase II*
	• Submit work for review by mentors • Make changes as suggested and submit for evaluation in Phase II.
July 14 - August 7	*Final touches to Analog activity*
	• Complete implementing capacitors. • Implement solenoids(and possibly, oscilloscope and speaker). • Occasional submission of code for review and suggested improvements. • Fixing further issues and implementing final suggestions
August 8 - August 14	*Submission for final evaluation*
	• Submission of entire work for final review and suggestions • Handing over final works for evaluation

Note: I will have to slow down my work during the month of May and/or June(mid May - End of May probably) due to my final semester exams.

5. About Me

I am currently in my final year at Maulana Abul Kalam Azad University of Technology (formerly WBUT), Kolkata, pursuing B. Tech. in Computer Science and Engineering. I have been contributing to the Qt version of GCompris since September 2017. Some of my contributions are:

- Removed dependency of ball direction on mouse buttons in Penalty activity: https://cgit.kde.org/gcompris.git/commit/?id=c94d7454c5a05364a1cbeae5b0bd1bed552ae5ed
- Progress bar corresponding to click zone only will be animated in Penalty activity: https://cgit.kde.org/gcompris.git/commit/?id=e5b68f68657cf1ac25a6aa31924acfaf4a180f0b

- Added easy mode in Letter-In-Word activity:
  https://cgit.kde.org/gcompris.git/commit/?id=c48f7057c27166edc6b8297bb3518a10ec2ee31b
- Improved layout in Hangman activity:
  https://cgit.kde.org/gcompris.git/commit/?id=6a66cd44106015bd369616574d656a8dd19d15be
- Fixed coding style issues for Ascending-Order activity:
  https://cgit.kde.org/gcompris.git/commit/?id=ade955e042ceb9774459a1181382159606a31502
- Added option to change case of letter to find in Letter-In-Word activity:
  https://commits.kde.org/gcompris/7c769594a5372e560904511404e2d1e156904050

Website: mouri11.github.io
Contact Info:
Email: Rohit Das [rohit.das950@gmail.com]
IRC Nick: mouri11