

# GSoC 2022 Dev Log

## Pre-Bonding Period

Pre-Bonding Period: I made the automatic Windows build script. Check it out here ([https://github.com/LibreCAD/LibreCAD\\_3/wiki/Building#windows](https://github.com/LibreCAD/LibreCAD_3/wiki/Building#windows) )

## Community Bonding Period (May 20 - June 12)

3 of June - Participation in Google Summer of Code 2022 Contributor Summit

## Week 1 (June 13 - June 19)

13 of June - Tried different ways to build LibreCAD 3 without VisualStudio. These are:

- Using a "CMakePresets.json" (Translated from CMakeSettings.json which is the proprietary version of Microsoft).
- Using the command appears on building on Visual Studio ( <https://pastebin.com/2vuNsH3w>. )
- Creating a Solution file and building it with MSBuild.
- Creating my own command.

14 of June - Improved CmakeSettings and trying to separately "Generate a Project Buildsystem"→"Build a Project"→"Install a Project"

15 of June - day off

16 of June - Windows build succesful. Working on Cmake→CPack→NSIS for the installer

17 of June - Investigating how to make CPack with relative paths (Requires CMakeLists.txt modification) or using Infrastructure of LibreCAD2 (Which should require modification of the scripts. I think I need to modify the NSIS script and the winbuild or postwindbuild)

18 of June - Testing CPack while changing all the possible paths in CMakeLists.txt to relative. It doesn't work. Using the infrastructure of LibreCAD2 seems difficult because of conan dependencies.

19 of June - day off

## Week 2 (June 20 - June 26)

20 June- Managed to get an installer with nsis and cmake. Solved the path problem with path.lua. The installer works succesfully. Currently working in adding the Qt libraries and others with bundle, so it won't need Qt installed in target platform.

21 June - Packaged QT5 with the installer. Also added licence, question to the user whether they want

to add LibreCAD 3 to path, create a shortcut or in the startup taskbar.

22 June - Solved issue [https://github.com/CRiSTiK24/AddCI-CDLibreCAD\\_3/issues/4](https://github.com/CRiSTiK24/AddCI-CDLibreCAD_3/issues/4). Working on finding where is RelWithDeblInfo created (in build process) to solve the other [https://github.com/CRiSTiK24/AddCI-CDLibreCAD\\_3/issues/3](https://github.com/CRiSTiK24/AddCI-CDLibreCAD_3/issues/3). [https://github.com/CRiSTiK24/AddCI-CDLibreCAD\\_3/issues/5](https://github.com/CRiSTiK24/AddCI-CDLibreCAD_3/issues/5) is in process

23 June - Issues with paths and folder location (creating in ProgramFiles instead of where user says)

24 June - Still trying to solve the path issues. Learned a bit of regex to use it for excluding exes more naturally but it fails to compile.

25 June - Day off

26 June - Something that isn't path.lua is calling files from build root folder. A functional path.lua also doesn't work well.

## Week 3 (June 27 - July 3)

27 June - Started building with Ubuntu WSL

28 June - Linux Build completed, managed to create an AppImage. It needs some extra things for it to work well (Qt inside && other files that I can copy from LibreCAD 2)

29 June - Continuing to solve AppImage problems and advancing with windows release path problems

30 June - Day off

1 July - Trying out paths for OpenGL (none functional meanwhile) and continued developing AppImage (it needs Qt plugins inside, though linuxdeployqt doesn't detect them)

2 July - Day off

3 July - Tried more relative paths for OpenGL and absolute paths. When changed, librecad.exe doesn't create at build.

## Week 4 (July 4 - July 10)

4 July - Advanced with appImage building, linuxdeployqt doesn't work as a standalone because it requires an older linux than Ubuntu 20.04 LTS. So i'm trying to use another tool to deploy appimages (app-image builder)

5 July - Tried out more ways to get the windows installer independent from source code. Stuck so far

6 July - Solved some problems with the windows installer. There is one problem remaining

7 July - Day Off

8 July - General revision of the Linux Appimage process.

9 July - Day off

10 July - First appimage and Solved issues with windows installer (shortcut works, automatic path.lua, VC redistribuible inside

## Week 5 (July 11 - July 17)

11 July - Third appimage.

12 July - Changing the Linux Operative System from Ubuntu LTS 22.04 to 18.04 and trying to build it there

13 July - Day off

14 July - Wrote the exact commands for Linux Build & AppImage. 4th Appimage this time from Ubuntu 18.04 . This one manages to get executed though has same green screen and icon problems and the windows one.

15 July - Trying out different way to get the some qt plugins. Managed to get a snap, though it has same problems

16 July - linuxdeployqt gives some strange error. I'll have to find a way to get it working, since all other ways I've tried dont' work (copying files manually, etc.)

17 July - Managed to get linuxdeployqt working. Now trying to fix SettingsPath so it isn't looking for outside the AppDir

## Week 6 (July 18 - July 24)

July 18 - day off

July 19 - Tried out ways to create the JSON and to find them with Path settings, but so far, either using relative paths causate strage behaviour ( looking at "bin" instead of "--/App/bin" or the program failing to compile

July 20 - Tried more ways to do settings path. Windows and linux are using different ways to find paths (linux one isn't able to find relative paths). This should be solved by setting the paths inside if( UNIX ) elseif(..)

July 21 - Tried to inspect the AppImage tmp mount to see if the files are organized differently when AppImage is executed, and that is why path fail. It is not the case

July 22 - Found why path weren't working (path need to be absolute, adding the returned path of "readlink /proc/self/exe" before the relative path. Also will try using Qt's <https://doc.qt.io/qt-5/qcoreapplication.html#applicationFilePath> to do the same.

July 23 - Qt's application file path is succesful for the lCUIlua paths, but for settings ones in lCkernel, I need to implement some extra methods to get the path from

July 24 - Created a Method from lcUI to get the path from. This way, lckernel can get the path with the qt library. However, this provokes a deadlock!

## Week 7 (July 25 - July 31)

July 25 - Instead of having the method in lcUI i've moved to lckernel, to not create the deadlock. I'm having some trouble due to having to use the object. I've noticed that there is an "extern lc::storage::settings::ModuleSettings viewerSettings" created, so I'm trying to use it.

July 26 - investigation in the "extern" keyword. I'm not really sure where to get the instance of ModuleSettings, I need more investigation to see how can I get it (viewersettings.h, settings.h).

July 27 - In a hesitation about how to proceed because of "Even if I set the correct path to the settings from the mainwindow.h , that wouldn't allow getting the right path when using the kernel without lcUI ( If the code inlcviewernoqt/viewersettings.cpp is the equivalent without lcUI, then I guess it should be implemented another way setting the path)"

July 28 - I've managed to add the absolute path for the settings directly in lcUI/managers/uiseettings.cpp without having to modify lckernel, and it works. The applmage now has no segmentation fault and it launches correctly (still has some issues which i mention next)

July 29 - Tried different ways to change res.cpp in lcviewernoqt so it has the absolute path from lcUI (Right now it has c++ error c2373 redefinition different type modifiers )

July 30 - day off

July 31 - day off

## Week 8 (August 1 - August 8)

August 1 - day off

August 2 - Passed all the scripts to the format in yaml format and began testing it for Github Actions

August 3 - Tried out more ways to add the path to lcviewernoqt, errors of missing links, namespace errors, variable defined more than needed...

August 4 - Investigated more in depth the code, and tried out more ways to get it working (still failure compiling)

August 5 - Managed to add a code to modify the path from lcUI, but right now, even if it compiles , doesn't change the path (some kind of bug, or the design is not correct

August 6 - Found a way using functions instead of directly modifying. Tried to verify with the applmage, but seems a file stopped being created.

August 7 - Managed to get the snap and Applmage functional ( At the beginning the painter is not working, but when opening a new screen, the painter works)

## Week 9 (August 8 - August 14)

August 8 - Solved the issue with the painter creating before the path change

August 9 - day off

August 10 - Restructured the github yaml towards using scripts with the commands inside instead of using the commands directly

August 11 - Solving various issues with the yaml (there are still some)

August 12 - More little changes solved (It still doesn't work)

August 13 - Day off

August 14 - Day off

## Week 10 (August 15 - August 21)

August 15 - solved winqt action problems

August 16 - Modified YAML with github variables

August 17 - Solved small errors in the scripts ( mostly related to paths and cloning recursively)

August 18 - Windows artifact works, now working on appimage and correcting errors with libdrfx library (it's not found)

August 19 - Managed to get the AppImage, Also began working on modifications of the snapcraft script. Managed to get the snap too!

August 20 - Modified the AppImage and Snapcraft to Ubuntu 20.04 as 18.04 was going to be deprecated. Now all artifacts work!

August 21 - Strange error when installing Librecad on Windows VMs.

=== Week 11 ( August 22 - August 28 )

August 22 - day off

August 23 - The errors look related to OpenGL, though it's strange. I don't know if this error needs correction or if it's expected librecad to fail in VM with such old OpenGL versions

From:  
<https://dokuwiki.librecad.org/> - Wiki

Permanent link:  
<https://dokuwiki.librecad.org/doku.php/gsoc:log2022?rev=1661286890>

Last update: **2022/08/23 20:34**



