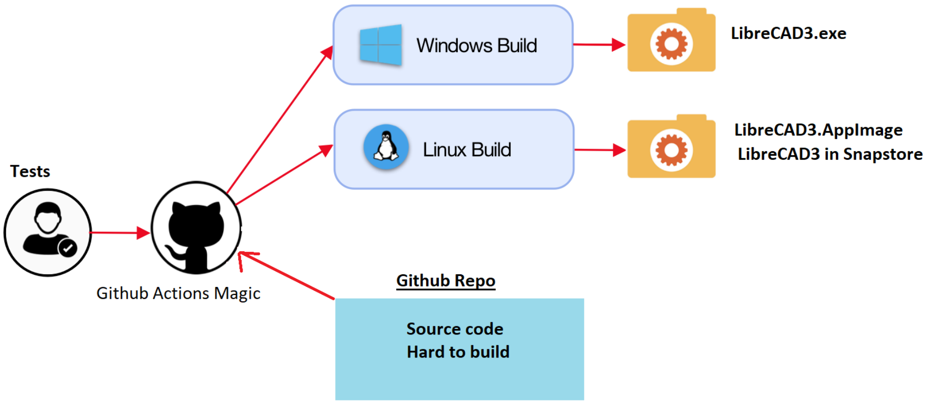
LibreCAD 3 automatic releases

Name: Cristian Bezerdic Stoica

April 18, 2022



Contents

1 Brief Proposal Summary

This is a copy-paste of the summary written on Google's web page in Proposal Summary

Right now to install and use LibreCAD3 on your computer you have to get all the dependencies and build it yourself from the source code in the Github repository. There are some building instructions, but those are incomplete or difficult to follow for a non-developer user.

My objective is to use Github Actions to provide LibreCAD3 with regular releases in Windows and Linux. The windows release will consist of a .exe on the Github releases page of the Repository, while the Linux one will be released on Canonical's Snap Store as a snap and Github releases page as an AppImage. These releases will be tested before they are uploaded to Github/Snap Store.

If you are interested in how I plan to execute this objective feel free to open the Pdf, where there is my personal information, the detailed implementation of this automatization of releases and more!

I've done two drawings describing it overall, the first one is the user vision which is the one I've described here `https://imgur.com/a/XGOmMM6` and the other one is the implementation diagram `https://imgur.com/a/KMgPU9V`. The first image is found on the first page, while the implementation one is found in "Detailed Proposal".

It's the proposal of the organization found in `https://github.com/opencax/GSoC/issues/59`.

2 Personal Information

- Name: Cristian Bezerdic Stoica
- E-mail address: c.bezerdik@gmail.com
- Zulip username: CRiSTiK24 in https://librecad.zulipchat.com/
- Github Username: CRiSTiK24
- Timezone: UTC +1 Barcelona timezone

2.1 Brief background info

Hi, my name is Cristian! I'm a second-year university student of Computer Science in UdG, Spain. I've done 1'5 years of c++ on programming subjects and right now I'm learning shell and c in Operative systems. I know basic git (pulling, pushing...) and I've used a Linux laptop since 2 years ago with Arch Linux, so I'd say I'm familiar with using Linux and having a decent understanding of its inner works. I've never contributed to open-source before, but I've used a lot of open-source before, such as 7z, qBittorrent and most of the tools I use in my Linux Laptop.

2.2 Time availability

During this summer I still don't have any special plans.

I'll do 2-hour classes every day for the English ”B2” exam for a week and a half at the beginning of July and I might study for the theoretical car licence. I also practice MMA and take piano lessons, though I should be able to do those in my free time without any trouble.

Other than that I'll finish my Univerity exams in mid-June.

3 Project Information

3.1 Project Title

The project title is ”LibreCAD 3 automatic releases”, which matches the original Github proposal made by the mentor.

3.2 Pre-proposal contact with LibreCAD3

I managed to contact the mentor (feragon in GitHub) of this project to get familiarized with it. I went through the Windows install process wiki here https://github.com/LibreCAD/LibreCAD_3/wiki/Building, but it was not updated, so I tried to solve any problem I faced and when stuck asked the mentor for help, who changed the source code based on the errors.

Once I finally build it, I updated the install process wiki so it was up to date again and added some verbose for inexperienced people. After that, I began my contribution to the project, which was a script to automatically install LibreCAD3 on a Windows computer through CLI (this should be useful for the automatic releases in this proposal). I managed to code it in a batch file and sent it to my mentor. After that, he recommended separating every command and updating the building wiki with it, with some explanations so future devs can maintain it easily. Here you can check any changes I made https://github.com/LibreCAD/LibreCAD_3/wiki/Building/_history.

Here is the batch script for the windows installation:

3.2.1 Windows Build Script

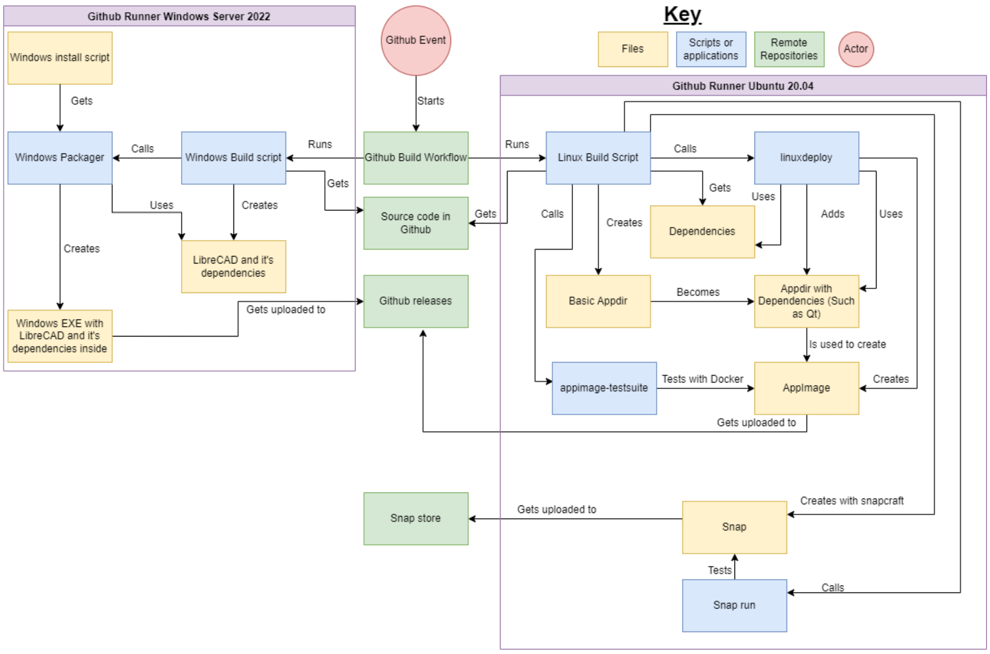
```
REM This script has to be used in an elevated cmd

REM Install Chocolatey

@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat
↪  None -ExecutionPolicy Bypass -Command
↪  "[System.Net.ServicePointManager]::SecurityProtocol = 3072; iex ((New-Object
↪  System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))"
↪  && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"

choco feature enable -n=allowGlobalConfirmation
&::Get rid of confirmation prompts

REM Install Git

choco install git

REM Install VisualStudio with all the needed components

choco install visualstudio2019community --package-parameters='"--add
↪  Microsoft.VisualStudio.Component.VC.CMake.Project
↪  Microsoft.VisualStudio.Component.VC.Tools.x86.x64
↪  Microsoft.VisualStudio.Component.Windows10SDK.19041"'
```

```

REM Install Python for pip and use pip for Conan

choco install python
pip install Conan

choco feature disable -n=allowGlobalConfirmation &::Enabling again confirmation prompts

REM Cloning repository

git clone --recursive https://github.com/LibreCAD/LibreCAD_3.git
cd LibreCAD_3

REM Creating and moving to Conan dir. If it fails, modify the profile compiler version and try again

mkdir conan
cd conan
conan install ..
if NOT %errorlevel% == "0" (
conan profile update settings.compiler.version=16  default
conan install ..
)

REM Getting the Qt5 installer

cd ..\..
curl -O https://d13lb3tujbc8s0.cloudfront.net/onlineinstallers/qt-unified-windows-x86-4.3.0-1-online.exe
mkdir Qt

:: These arguments will have to be added and specified below since the license needs a Qt account.
::--email <email> --password <password>
REM Installing qt5

qt-unified-windows-x86-4.3.0-1-online.exe ^
  --accept-licenses ^
  --default-answer ^
  --accept-obligations ^
  --root "%cd%\Qt" ^
  --confirm-command install ^
  qt.qt5.5152.win64_msvc2019_64

:: directories should be dictated to the user or hardcoded as relative paths

REM Adding new env Variables and PATH appends
setx QTDIR "%cd%\Qt\5.15.2\msvc2019_64" & setx QT_QPA_PLATFORM_PLUGIN_PATH "%QTDIR%\plugins\platforms\"

:: Here can't use setx since if PATH is long, it will be truncated, therefore, Powershell
:: %cd%\Qt\5.15.2\msvc2019_64\bin is not in the wiki, I had to add it so it detected Qt5Widgets.dll
```

```
Powershell.exe -Command "[Environment]::SetEnvironmentVariable('Path', $env:Path + ';%cd%\Qt\5.15.2\msvc2019_64;%cd%\LibreCAD_3\out\build\x64-Debug\lib;%cd%\Qt\5.15.2\msvc2019_64\bin', 'User')"

:: Modify CMakeSettings Doesn't work
::copy /y CMakeSettings.json LibreCAD_3\CMakeSettings.json
REM Last step is not Automatic, Please Visit https://github.com/LibreCAD/LibreCAD_3/wiki/Building#visual-studio-cmake
```

3.3 Detailed Proposal

I think the best way to explain the proposal is by following the diagram:



I'll follow the vocabulary dictated in https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions.

3.3.1 GitHub Event

GitHub Event is an actor that represents a LibreCAD3 dev or merges/pushes into the repository. It's the trigger that will begin the building and releasing of the project. I think that the event that triggers the workflow should be on pushing since this is not a finished software yet, and being able to check it works correctly after every commit is very helpful for reproducing issues. In the future, it should be better to be for the workflow to be triggered manually by a developer, since stability should be more important than features once there is a large pool of users.

3.3.2 Github Build Workflow

This will be the workflow that will set the two runners(A kind of Virtual Machine hosted by GitHub). The first runner will be Windows, while the second will be Linux(Ubuntu).

Each runner will have a job, for the windows one, it'll consist of getting LibreCAD3 installed, packaging it with a Windows install script into a .exe and uploading it intro GitHub releases.

On the other side, the Linux one will also have to build it and then it'll divide into two parts, the first one will be the creation of an AppImage and its upload to GitHub releases. The other one will consist in Uploading a Snap of LibreCAD3 into the Snap store.

3.3.3 Source code in GitHub

This is the main repository where code is pushed, and where we'll get the current code from. In this case, I'll be from https://github.com/LibreCAD/LibreCAD_3.

3.3.4 Github Releases

Here will be listed all releases of the project if they pass the test. It'll consist of a Windows .exe and a Linux .appimage for each release.

3.3.5 Windows build script

This will be the windows script that will clone the source code and install any dependency needed. I managed to automate this step as mentioned in the Pre-proposal contact, you can check the batch script by clicking here.

3.3.6 LibreCAD and dependencies(Windows)

Right now this consists of LibreCAD3 folder which is the one we clone from GitHub and Qt5 folder which is one of the dependencies we install separately. Conan dependencies may be needed too for the installation on the user's device.

3.3.7 Windows install script

This will be a script that will be packaged in the installer, it will be responsible for when a user executes the .exe to extract all the contents from inside and install it on the computer.

3.3.8 Windows Packager

This will be the next script that the job will execute after the building. It will merge Librecad3 build with its dependencies along the windows install script to make a .exe. It should be done with something like https://adamtheautomator.com/bat-to-exe/.

3.3.9 Windows Exe with Librecad and dependencies

This will be the file that will be uploaded to GitHub Releases. There should be a step before uploading that should consist of it being tested. It seems that a GitHub action can be assigned to check if it works correctly.

3.3.10 Linux Build Script

On the Linux side, this will the script that will build the project and get the dependencies so we can create the AppImage and Snap. It should be similar to the Windows one, but the dependencies should be easier to get from the apt.

More information in Basic Appdir and Snap.

3.3.11 Dependencies

This is an abstract file representing the dependencies needed for the project to work, in this case, it mostly meant Qt. As in Linux, it should be installed with the package manager, we'll add some arguments to the tools to capture them in the snap or AppImage.

3.3.12 Basic Appdir

As in AppImage documentation, it says that a dir should be created to store the project. In LibreCAD3 we are using CMake to build the project, so as it is written in the guide, adding a parameter to the building should create the Basic app dir.

3.3.13 linuxdeploy

Following the AppImage guide, now it's written that an extra step is needed to build the appImage. It mentions that some things like Qt plugins can be added with their tool ”linuxdeploy”.

Also, I'll be used to create the AppImage.

3.3.14 Appdir with dependencies

This will be ”Basic Appdir” with the Qt plugins inside. It's the last step before creating the AppImage.

3.3.15 AppImage

This will be the file that will be uploaded to GitHub Releases after testing. I'll be created with linuxdeploy, with the –output argument.

3.3.16 appimage-testsuite

Before uploading the file we must make sure it works in the many Linux Distros. For this we'll use the tool appimage-testsuite. I'll test it in many distros using Docker.

3.3.17 Snap

On the other Linux side the ”Linux Build Script” will also have created a Snap. As LibreCAD uses Qt5, this will be created using the procedure in qt5/KDE apps. Here it describes that it can be built using ”snapcraft” if we have a ”snapcraft. yaml” in the repository.

3.3.18 Snap run

The same documentation as before mentions that we can use "snap run" to test our snap. If it passes the test, then it should be sent to the Snap Store. Down below at the same documentation, it says that a dev profile should be created and the snap's name registered to be uploaded. The command used will be "snapcraft login", "snapcraft register snap" and " snapcraft upload"

3.3.19 Last thing

All of this process is the result of a few-days search. Feel free to contact me to change anything as I have no experience in building and deploying things as complex as LibreCAD.

3.4 Deliverables and Development schedule

Development officially begins on June 13 and finishes on September 12 based on `https://developers.google.com/open-source/gsoc/timeline`. So I've decided to use the 13 weeks in between as timelines.

The creation of this proposal will have the following parts:

3.4.1 Coding of the Linux script (Week 1-2)

I have only built the project for windows, so the next step will be trying to build it for Linux. Best should be using Ubuntu as a distro since that is what will be used in the Linux Github runner. So I'll probably use WSL or an equivalent to try out the installation.

Once I get the project running, I can begin the automatization of the building. As in the windows one, I'll backtrack my steps to code it. Then I shall use the snap and AppImage to create the basic Appdir and Snap. I've noticed there is a similar process in the current version of Librecad in the scripts folder. So it'll be useful to not start from scratch.

3.4.2 Packaging the Windows install (Week 3-4)

As the automatization of the Windows build script currently works (only the last step which is using Visual's studio Gui to build is not, but using Cmake should be able to replace it), I'll have to code the Windows install script. Same as before, it seems that in Librecad's current version there is a Windows install script, and it seems that it uses Qt to deploy create the .exe.

3.4.3 Packaging the Linux Install (Week 5-6)

Here we'll use the documentation that I've stated in the Detailed Proposal subsection to create the AppImage and the Snap. Once I manage to get them I'll backtrack again and automate this step.

3.4.4 Github Actions (Week 7-9

Once I've managed to create the automatization scripts and a .exe, .AppImage and a snap, I'll be time to merge everything inside Github Actions. There I'll test every that script works inside the runners I'll code the Build workflow.

3.4.5 Testing and deploying (Week 10-11)

Once The workflow works, I'll add the final testing and will search on how to integrate it. I guess it should be something like (test app; if the last command line ended successfully: then deploy).

Also here Snap org profile should be created and LibreCAD3 name reserved to be able to push into the store.

3.4.6 Extra time for any unexpected event (Week 12-13)

During this time I'll have extra time to test everything that works well. I might be doing extra documentation if everything works fine or use this time to fix anything unexpected.