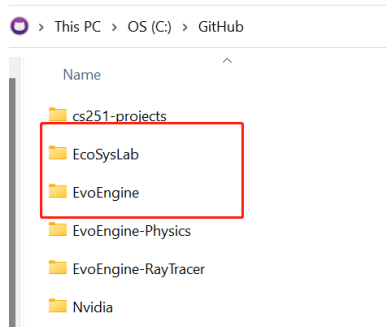


Here's the instruction for deploying the framework on Linux or Windows PC.

First, you need to download the code:

1. Please remove any previous repo you have for my framework (EvoEngine, RayTracerFacility, EcoSysLab, PlantArchitect., etc.)
2. Please clone the following repositories, under the same folder:
 - a. [GitHub - edisonlee0212/EvoEngine: An early-stage, cross-platform interactive application and rendering framework.](#)
 - b. [GitHub - edisonlee0212/EcoSysLab: A procedural vegetation framework based on EvoEngine.](#)

What you have after this, should be:



3. Please unzip ./EvoEngine/3rdParty/shaderc/shaderc.zip*. The result folder should look like:



4. (Optional) If you also want to have ray tracer rendering module, please clone the following repository, under the same folder (So EvoEngine, EcoSysLab, and EvoEngine-RayTracer should all be in the same folder):
 - a. [GitHub - edisonlee0212/EvoEngine-RayTracer: The path tracer module for EvoEngine. Using Nvidia OptiX 8.](#)

Next, you need to setup the building environment:

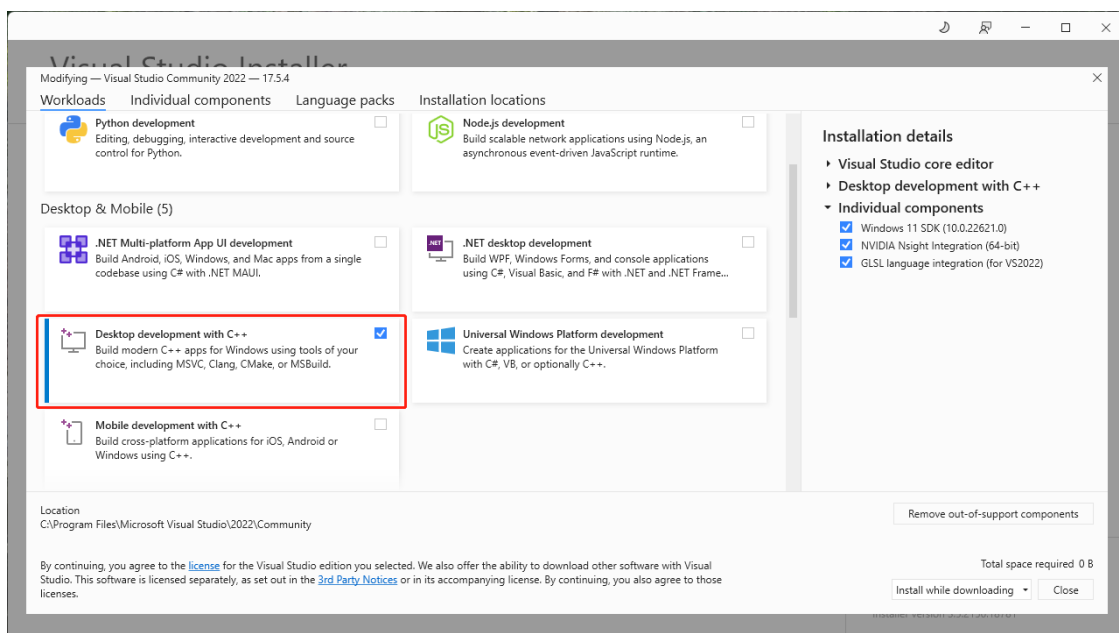
- For Windows PC, you need to have the following SDK/Software/Components installed:
 - CMake – Latest version
 - Latest Vulkan SDK ([LunarXchange \(lunarg.com\)](https://lunarg.com))
 - Python – Version >= 3.9
 - (Optional) OptiX – 8.0.0 (Please uninstall all previous versions you have on your PC, perform the uninstallation from Control Panel.) This is optional and you only need this for ray traced rendering module for the framework. The following should also be installed:
 - CUDA – Version >= 12
 - Nvidia Graphics Driver – Version >= 535
- For Linux PC, you need to have the following SDK/libraries installed:
 - Note: the framework is only tested on Ubuntu 22.04 LTS. You may face compatibility issues if you use other Linux distributions.*
 - Nvidia Graphics Driver – Version >= 525
 - Latest Vulkan SDK ([LunarXchange \(lunarg.com\)](https://lunarg.com))
 - Python – Version >= 3.9
 - The following libraries (apt-get):
 - build-essential
 - cmake
 - git
 - ninja-build
 - libassimp-dev
 - libglm-dev
 - libglfw3-dev
 - libxi-dev
 - libxrandr-dev

Now, to setup the development environment:

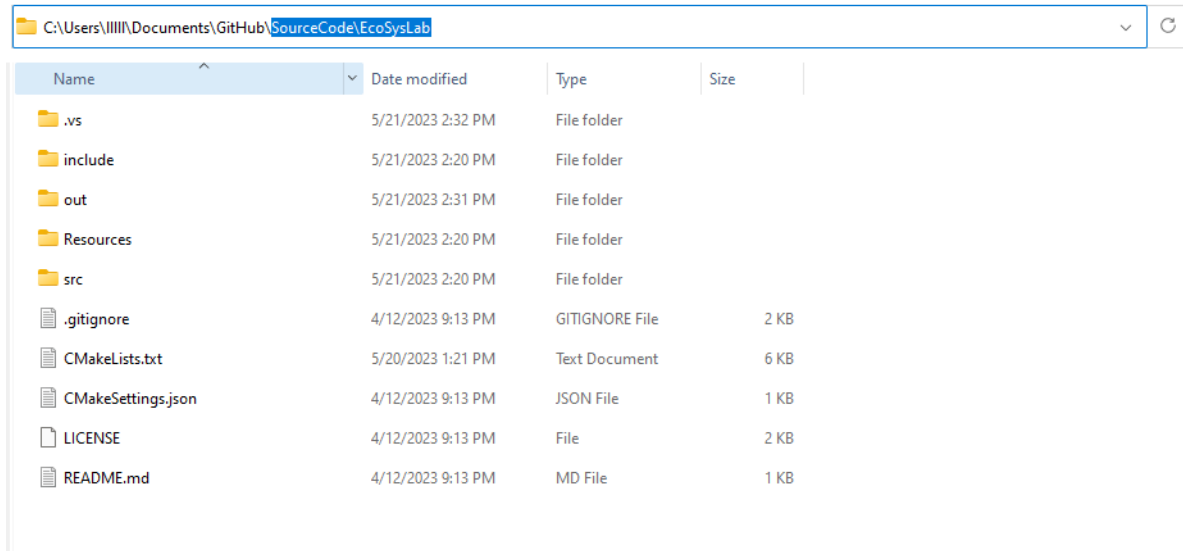
For Windows PC:

- Please make sure the following components are installed before building process:

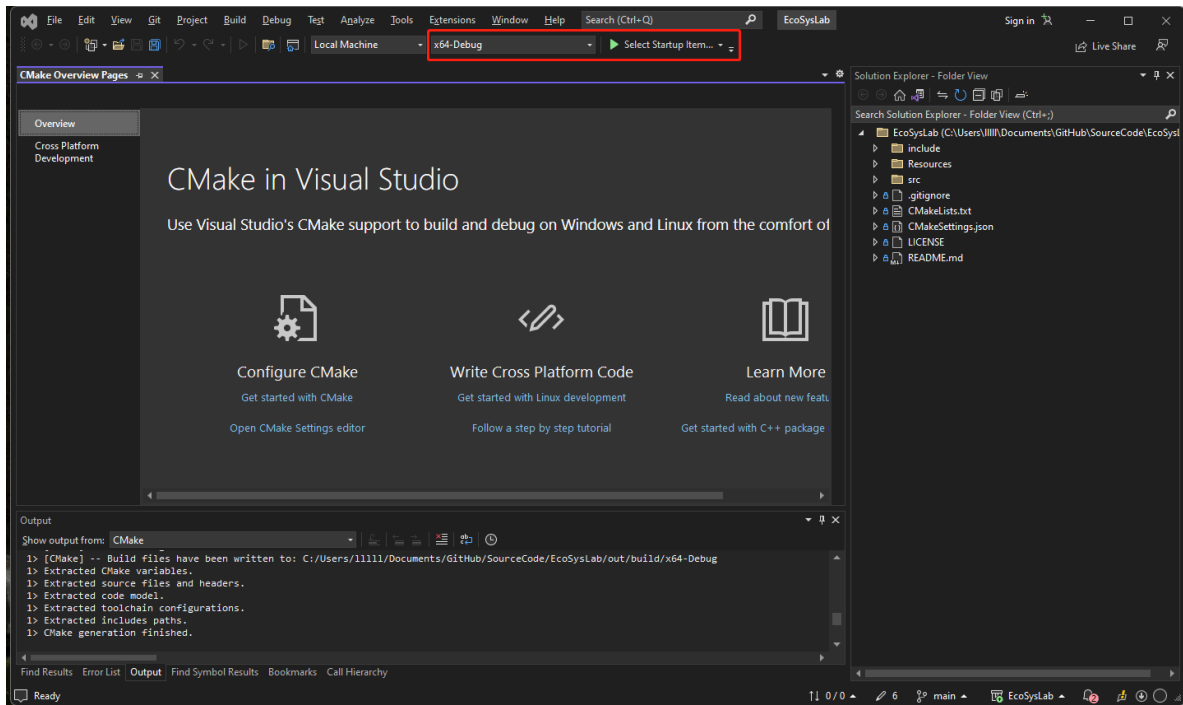
Visual Studio 2022 with “Desktop development with C++” selected:



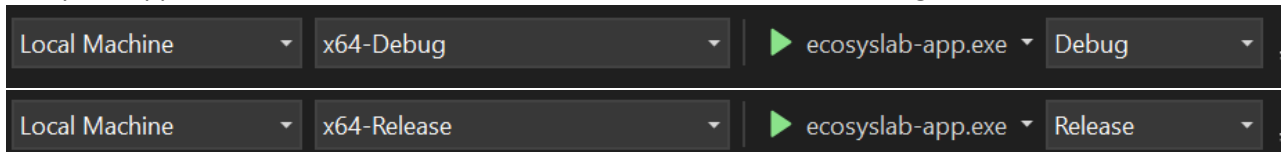
- Open the EcoSysLab folder as a project in Visual Studio 2022. The Visual Studio will automatically recognize it as a CMake project.



- Wait for a little while until Visual Studio loaded the CMake project, and you should see building configurations are set:

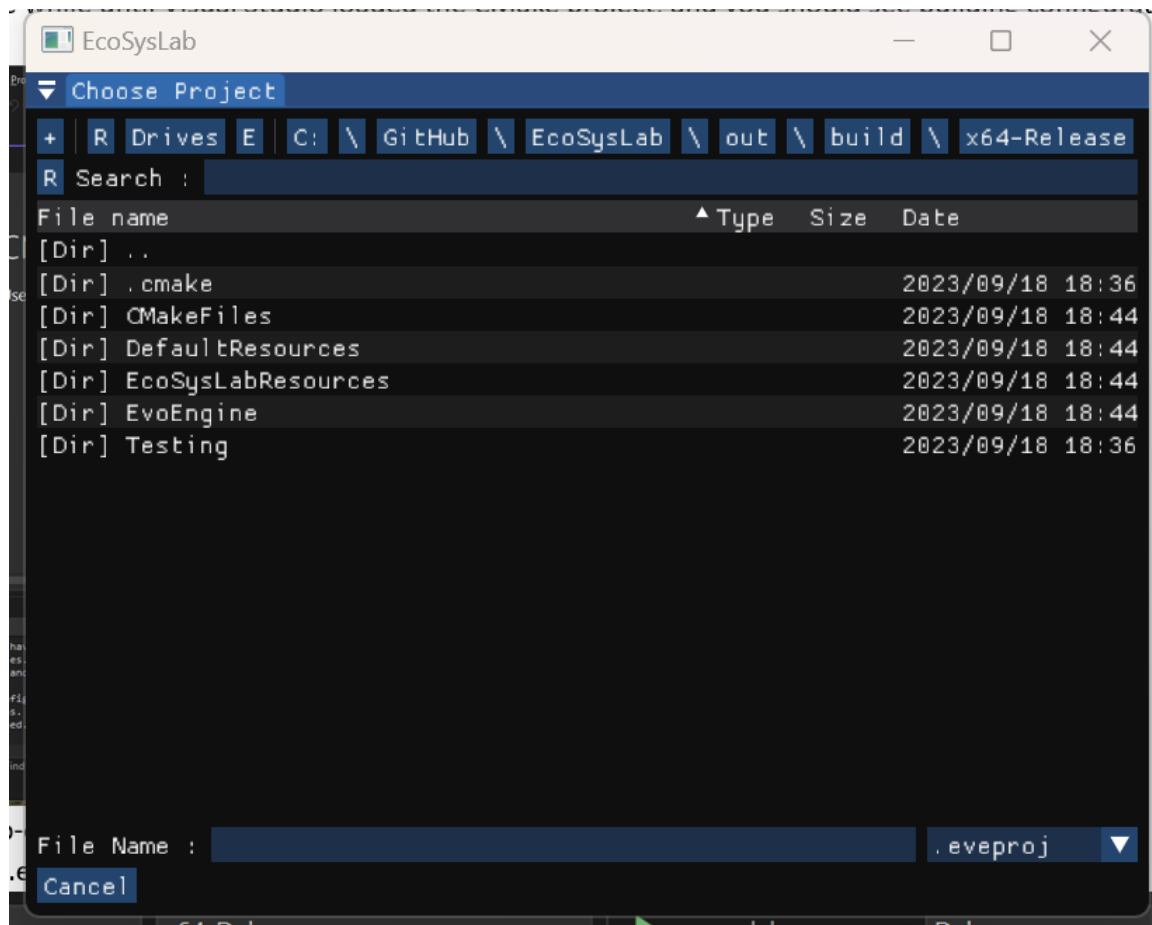


- Open the drop-down menu for building target by clicking the button highlighted with blue box, and select `evosyslab-app.exe` and click start button marked with red box to start building:



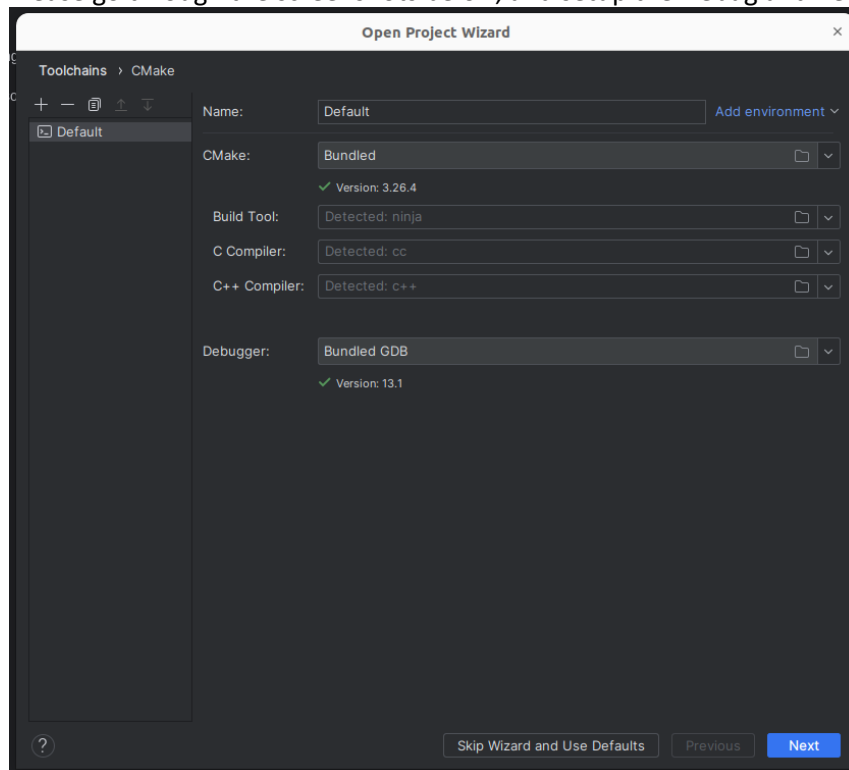
Note: The x64-Debug allows you to add breakpoint and debug the code, but it runs tens or hundreds of times slower than the x64-Release. Please make sure you use x64-Release when you want to use the framework.

- Once the build is finished, the program will run automatically. You will see a small window as the welcoming window of the framework.

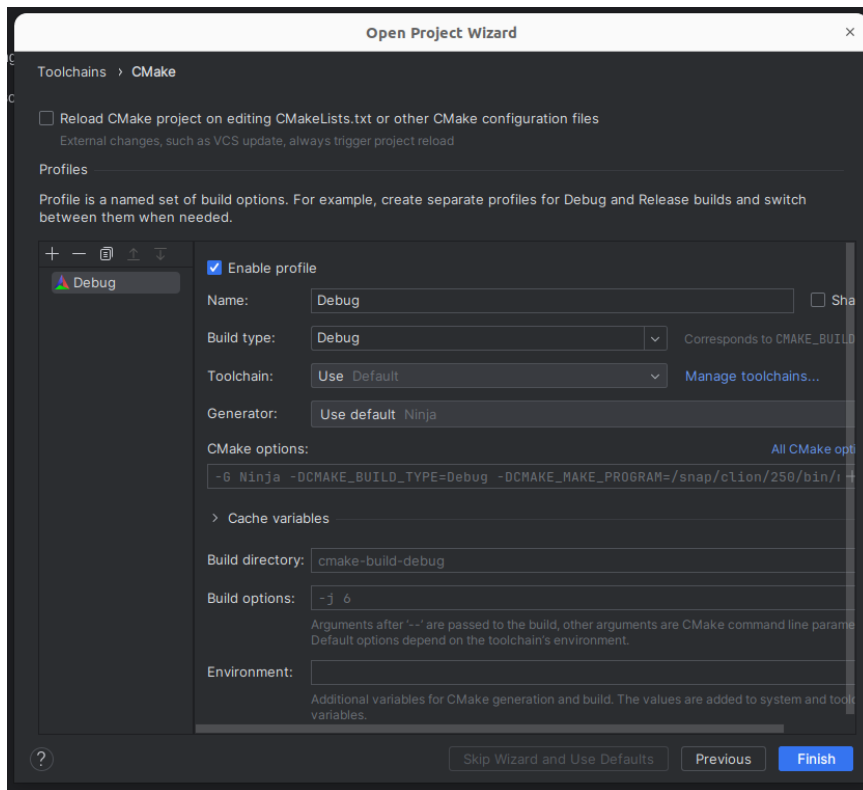


For Linux PC:

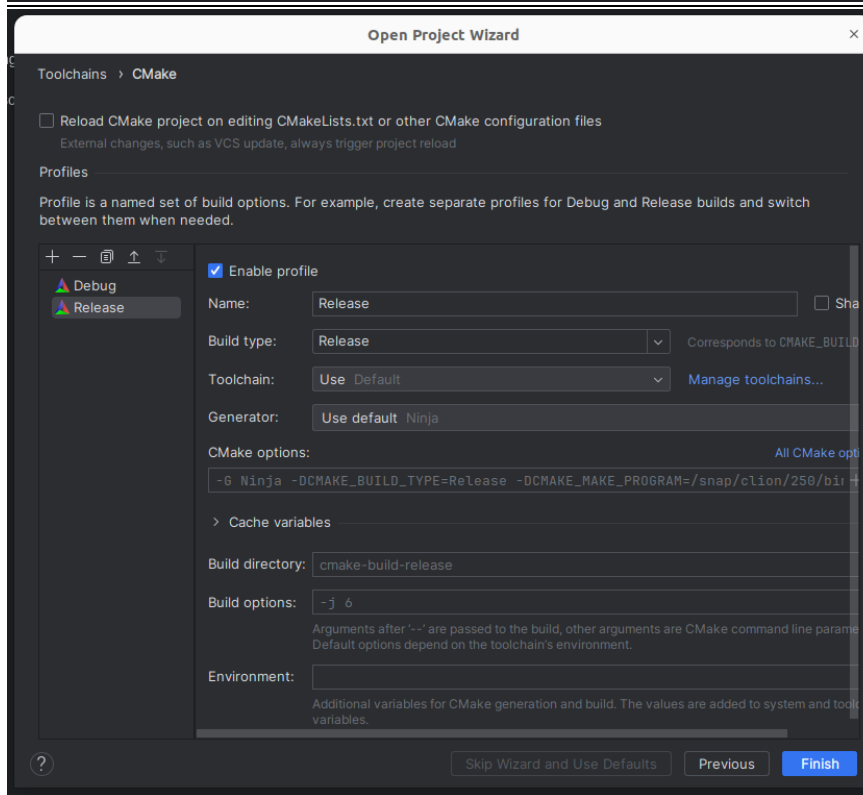
1. Please make sure you installed CLion on your computer.
2. Open the project with CLion, and you should be prompted to configure the project.
3. Please go through the screenshots below, and setup the Debug and Release profiles for the project:



4.

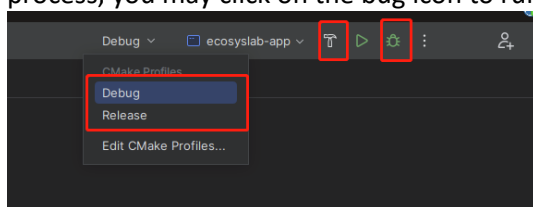


5.



6.

7. Then, select the profile based on your need, and build the project by clicking the hammer icon. After the building process, you may click on the bug icon to run the framework.

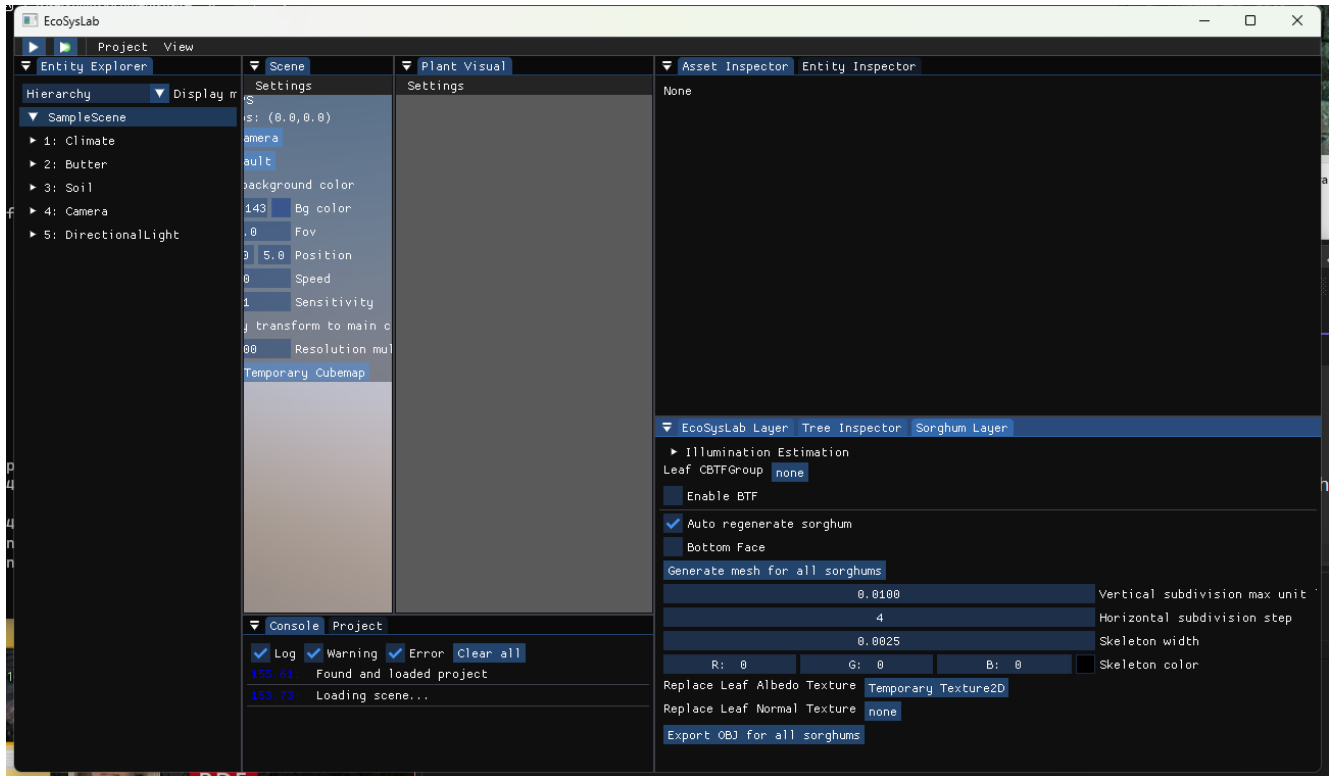


8.

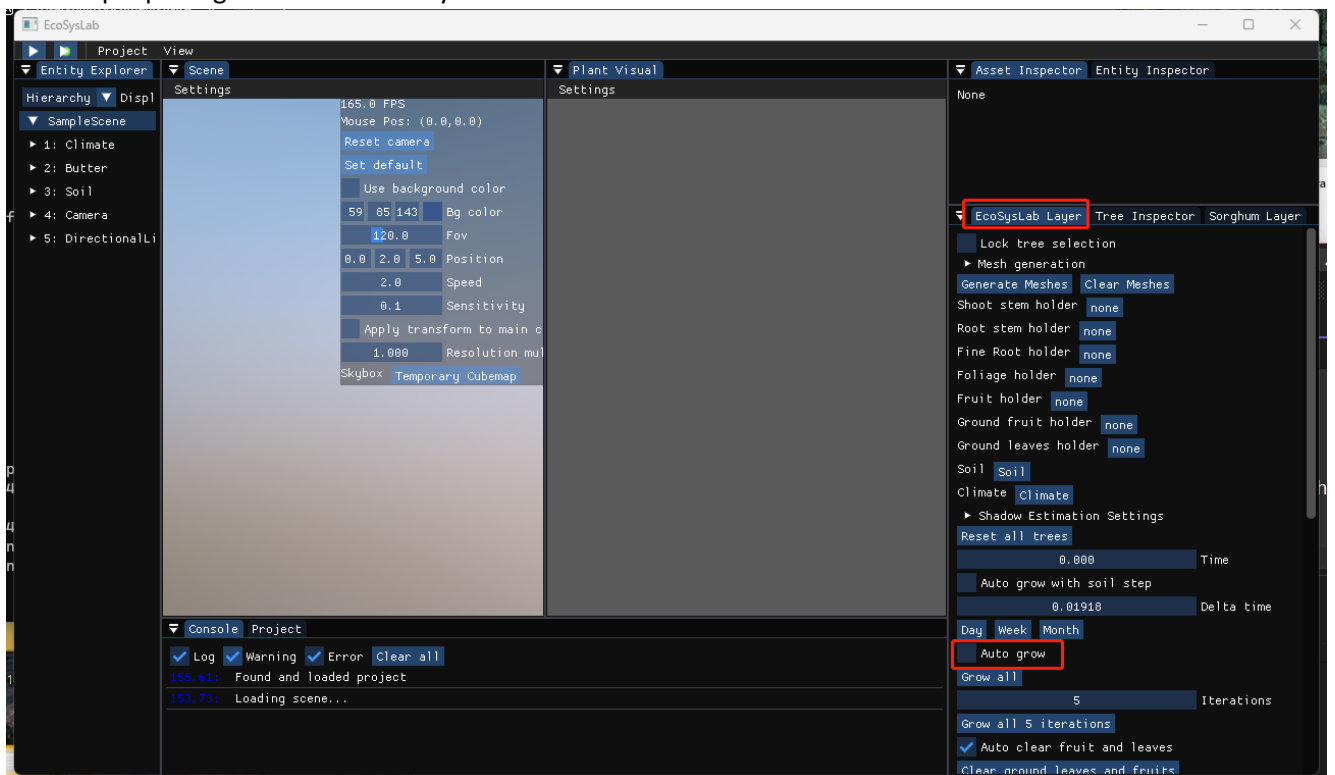
9. You should now see the same startup window for the framework.

Finally, some basic instructions on the framework:

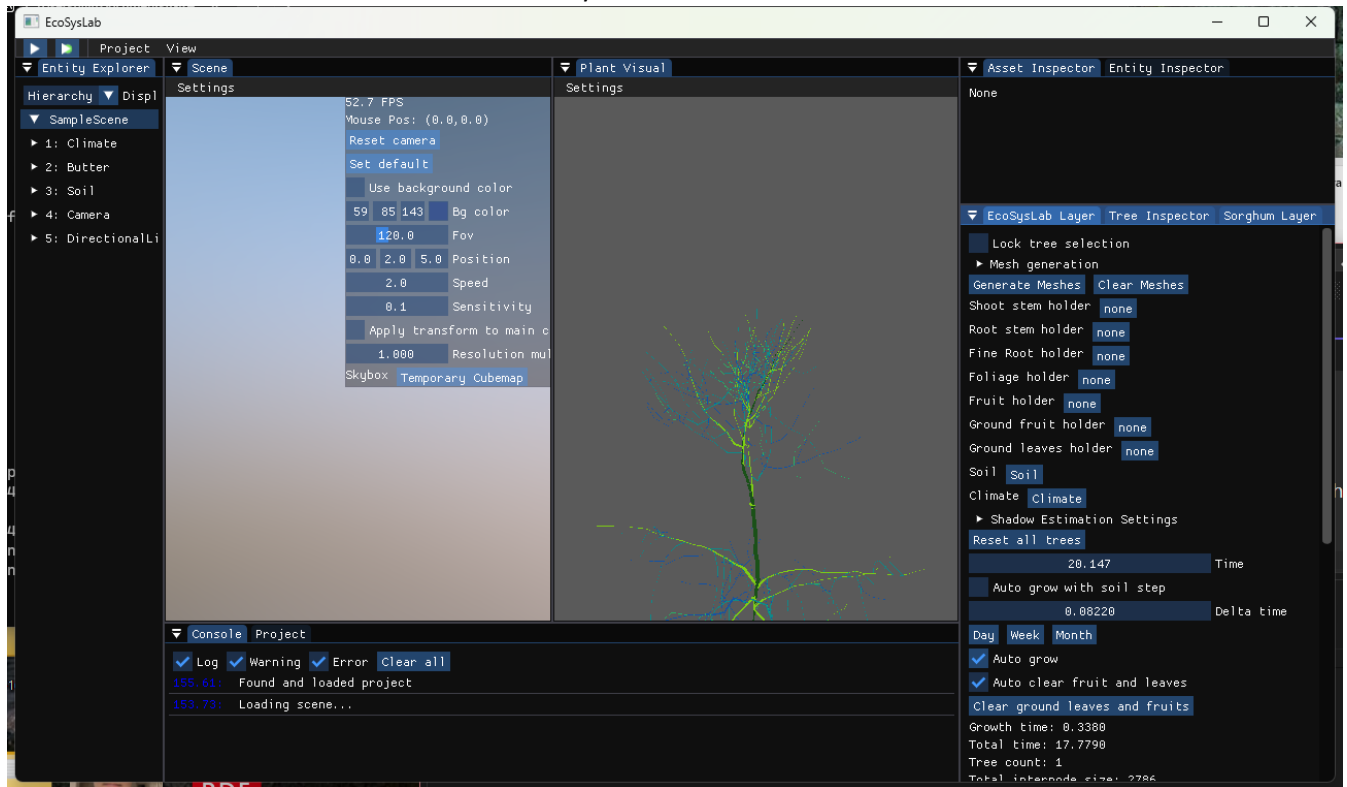
1. You should be able to see the project is loaded and the framework's running:



2. The project folder comes with a sample scene. Select EcoSysLab Layer panel and check "Auto grow" box to see the sample plant grows automatically:



3. You should be able to see the tree with its root system:



4. Click “Generate Meshes” to generate mesh so the tree will appear in the Scene window:

