

Installation User Guide

This guide serves as a supplementary installation manual for our project, 'A Visualization Tool to Help Java Developers Design, Implement, and Test Geometric Algorithms.' It is designed to complement the video installation guide provided with the project, offering detailed written instructions to ensure smooth setup and use

About

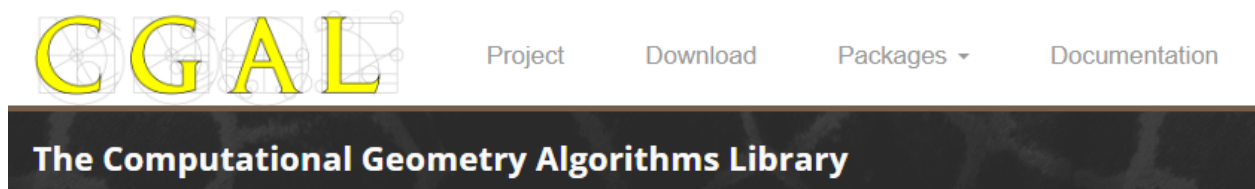
Our visualization tool addresses the current shortage of geometric algorithms libraries in Java. It leverages CGAL, an extensive C++ library renowned for its comprehensive collection of data structures and geometric algorithms. By integrating CGAL's robust capabilities with Java's strengths in visualizing geometric objects, our tool provides Java developers with a powerful resource for designing, implementing, and testing geometric algorithms. This integration not only enhances functionality but also improves accessibility for developers seeking advanced geometric computational solutions.

Getting Started

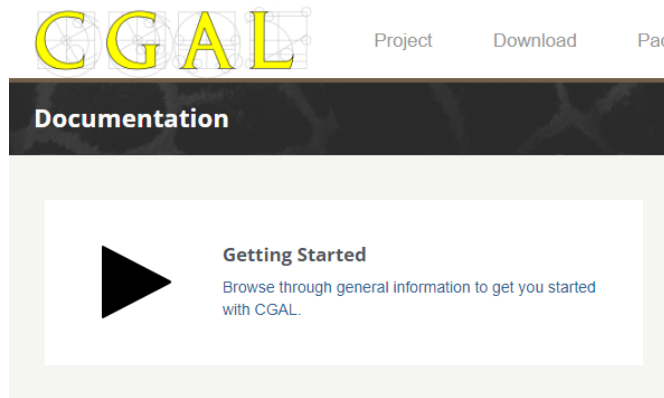
Step1 :

To get started you need to install CGAL instructions on building and installing CGAL and its necessary dependencies are on the CGAL website. <https://www.cgal.org/index.html>

Step 1a. Open CGAL website and click Documentation

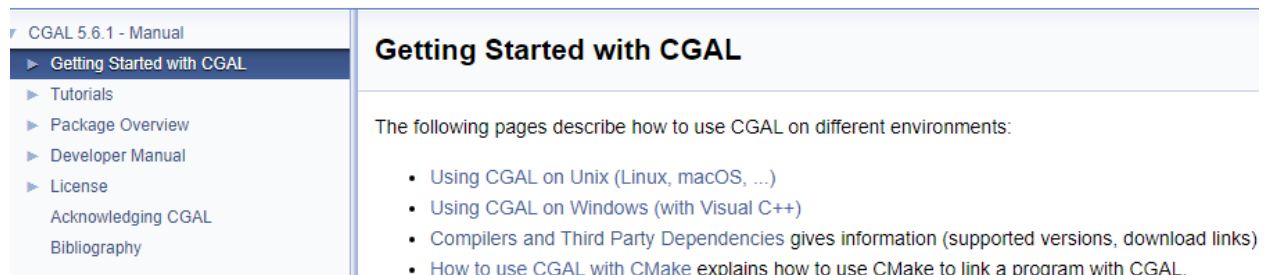


Step 1b. Click "Getting Started"



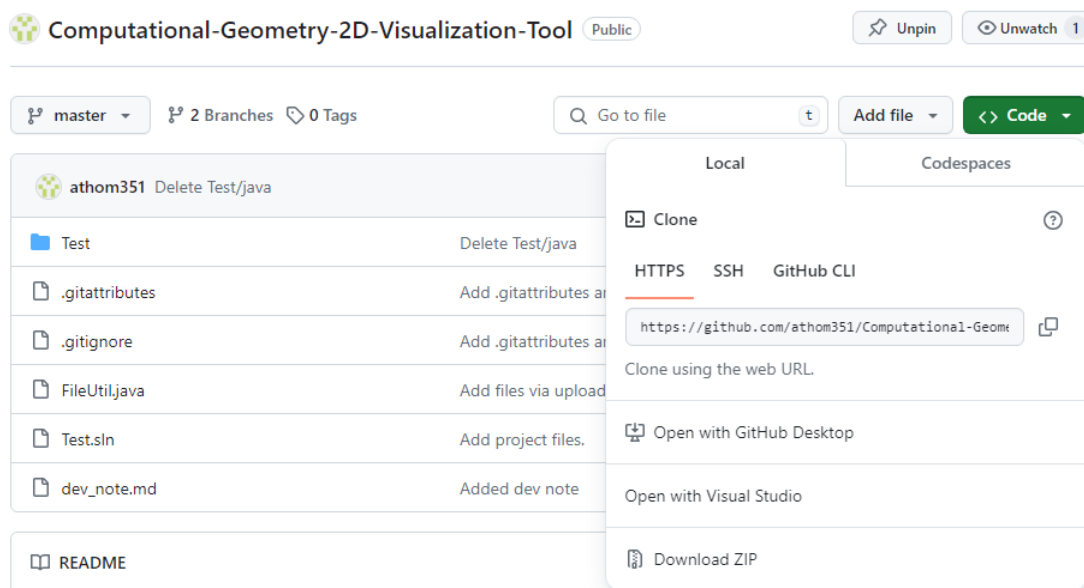
Step 1c. Following page describes how to use and Install CGAL on different environments.

CGAL 5.6.1 - Manual

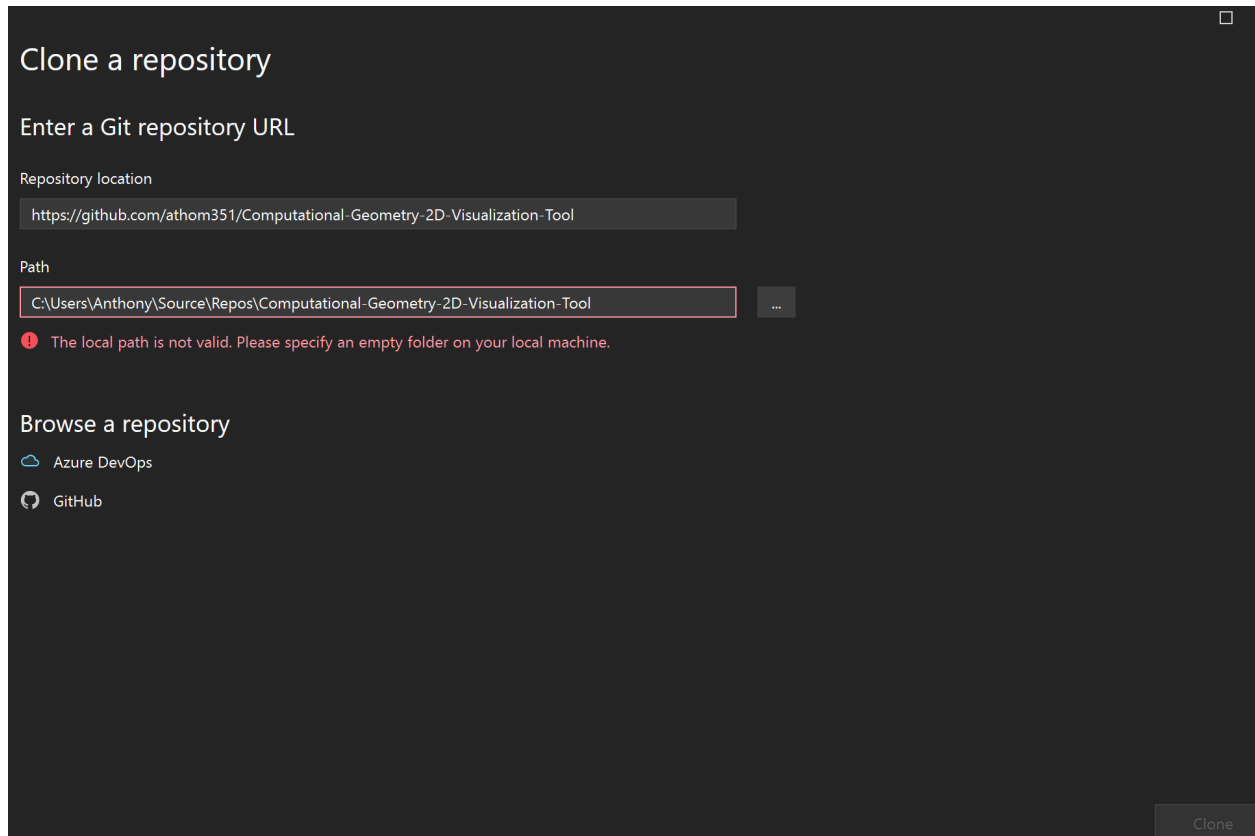


Step 2:

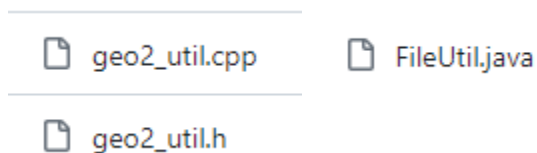
Now that CGAL is installed you will need to install our project there a two ways of installation either downloading the zip file or cloning from github



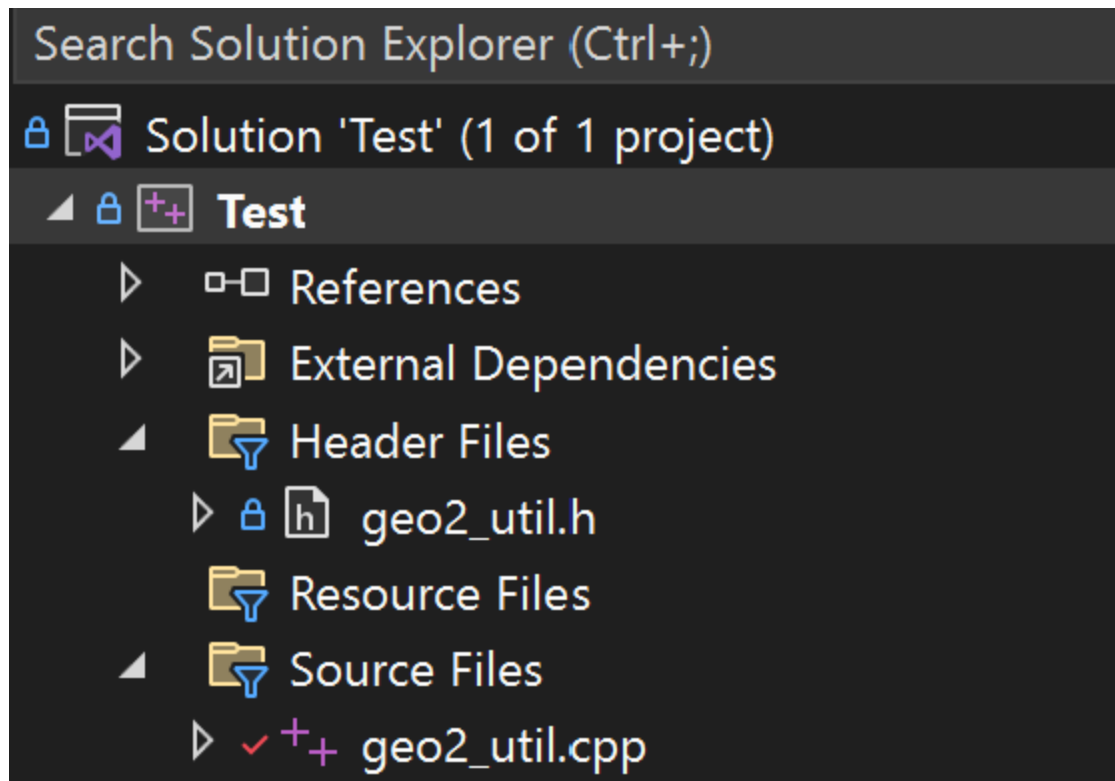
Step 2a: Installation from this point depends on your method of downloading and your IDE you run CGAL on. CGAL is commonly run on Visual Studio or VS Code the easiest installation in our case would be Visual Studio as it allows cloning directly from GitHub. Simple press open with visual studio and press clone on the bottom right and you should have the C++ code ready to execute.



Step 2b: If you could not clone you will have to download from a zip file the three main files needed are `geo2_util.cpp` , `geo2_util.h` , and `FileUtil.java` for C++ and Java respectively.



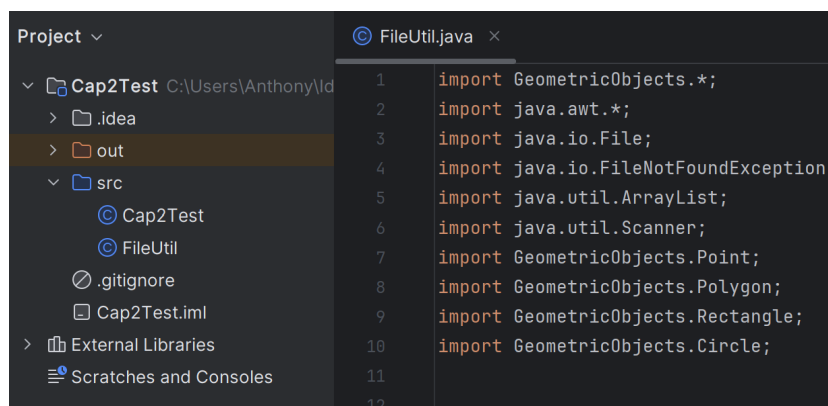
Step 2c: After cloning or downloading through the zip file you should have two C++ files one called `geo2_util.cpp` and another called `geo2_util.h`. These files are all that are needed to run our program on the C++ side. Due to the nature of our project being a resource for developers looking to explore geometric objects we do not provide a main class as it is left up to the user to create what they want. It is assumed that the user will be at least somewhat familiar with CGAL and some of the objects associated with the library.



Step 3:

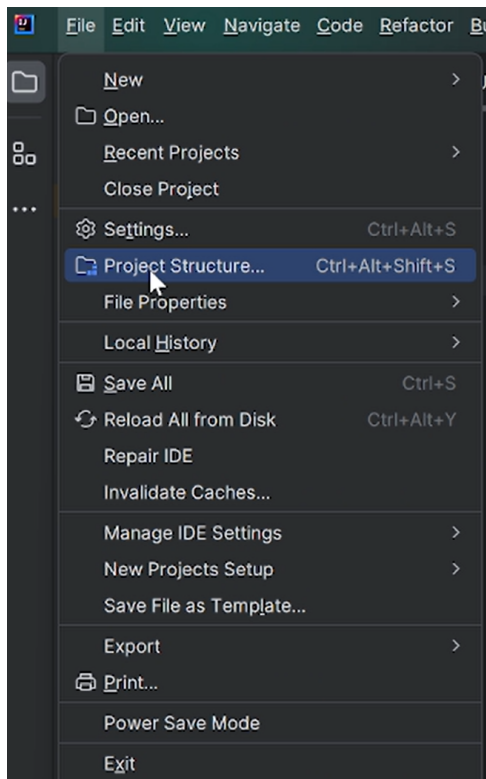
The third step is where we install the java portion of our application. For this you will need `FileUtil.java` which was previously downloaded along with the C++ code. You will also need a Java IDE. Almost any IDE should work as long as it allows the addition of libraries via `.jar` file. For this guide I will be using IntelliJ, but I have seen it work on Eclipse as well as NetBeans.

Step 3a: Add our `FileUtil.java` to newly created project

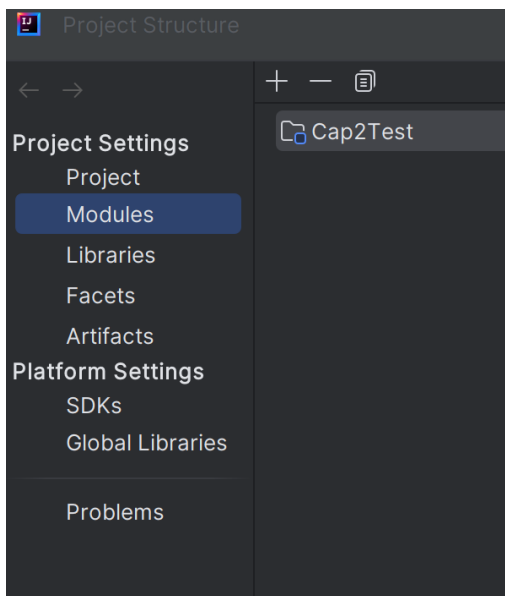


Step 3b: When FileUtil is added you may see many errors this is due to needing the GeometricsObjects library. This library is owned by our product owner and essential for running our java code that being said we can not release it due it not being our work. Even though we may not be able to share it we will still detail how to add the library if you happen to acquire it.

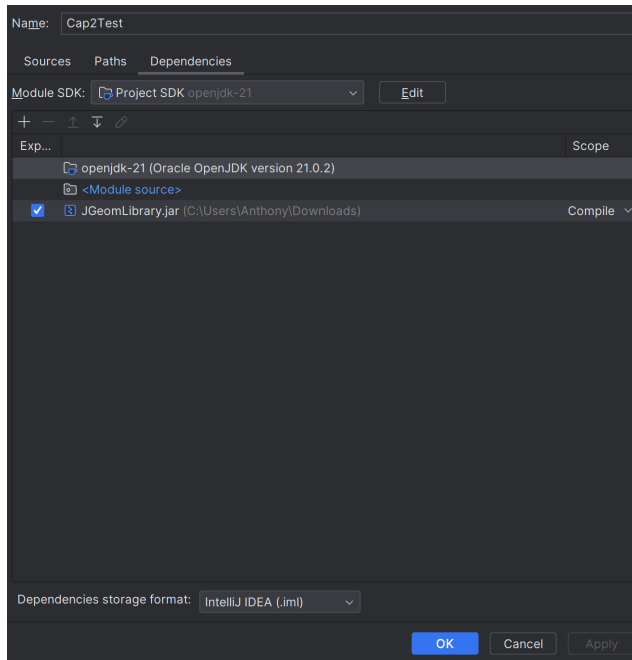
Step 3c: For IntelliJ you will need to enter “File” then click “Project Structure”.



Step 3d: Under project structure enter “Modules”



Step 4d: In modules check the jar file and hit apply then hit ok.



Step 5:

After these steps you are now ready to create a main file for java and C++ to create geometric objects. Note that this tool is intended for developers and you are expected to create your own main classes. In our demo we give examples of how our tool can be used to show functionality but in general our tool is intended for creation.