

Geometric Visualization Tool

Students: Jinpan Zheng

Mentor: Antonio Hernandez

Instructor/Faculty: Masoud Sadjadi, Florida International University

PROBLEM

Geometric computation can be computational expensive. **CGAL** offers efficient and reliable geometric algorithms, yet it lacks flexible visualization functionalities. This module of the project aims to utilize the power of CGAL to handle the computation, meanwhile, provide users essential object data with visual information for further processing.

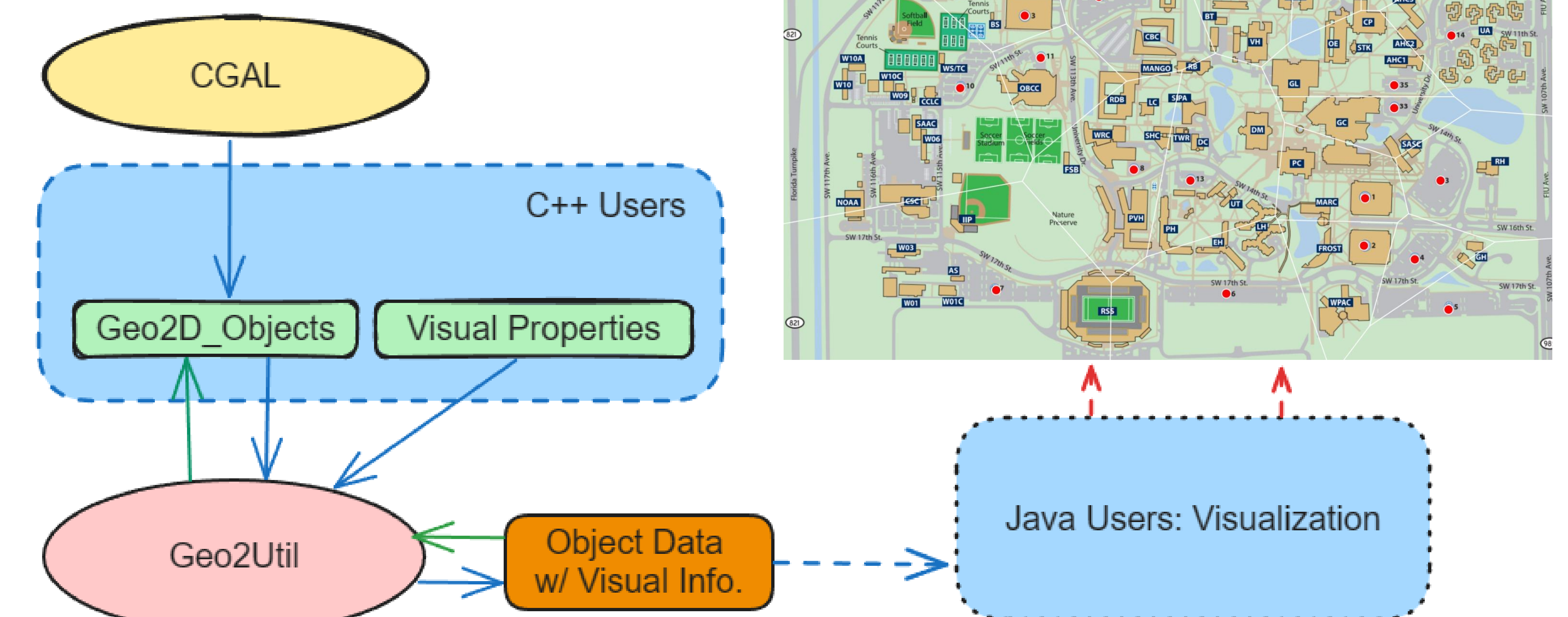
CURRENT SYSTEM

The current system (**Geo2Util**) provides export and import object data functions for 8 types of 2D geometric objects: **point**, **segment**, **circle**, **triangle**, **rectangle**, **polygon**, **line** and **ray**. Also, it offers two versions (**default & customized**) of functions to convert 2D geometric objects to string representation. C++ users can create 2D geometric objects with preferred visual effects for objects' boundary color, boundary type and interior color. Other users (e.g., Java users) can use the exported object data for further processing, such as visualization.

IMPLEMENTATION

Geo2Util is implemented as a utility library, containing a set of methods to create string representations of geometric objects with visual properties, export object data to files, and import object data from files.

- Visual properties: boundary color, boundary type, interior color.
- Export: object data with visual information are exported to text-based file in dedicated format.
- Import: extract pure geometric objects from files while discarding visual information.



SUMMARY

As one of the pivotal members in this project, my responsibility is to create the C++ utility library (**Geo2Util**), which utilizes efficient algorithms from CGAL to generate geometric object data (with visual information), for the Java subsystem to visualize the objects. Geo2Util offers two-way data processing that users can export visual object data to files and retain objects from files. The main goal of this module is to remain the ease-to-use of a Java program while lessen the cost of geometric computation. Ultimately, Geo2Util aims to acquire ideal efficiency of handling high volume of object data to support sophisticated geometric visualization.

REFERENCES

[The Computational Geometry Algorithms Library \(cgal.org\)](https://www.cgal.org/)

REQUIREMENTS

CGAL

