Topologic Calculator

https://github.com/INoodleI/TopologicCircles
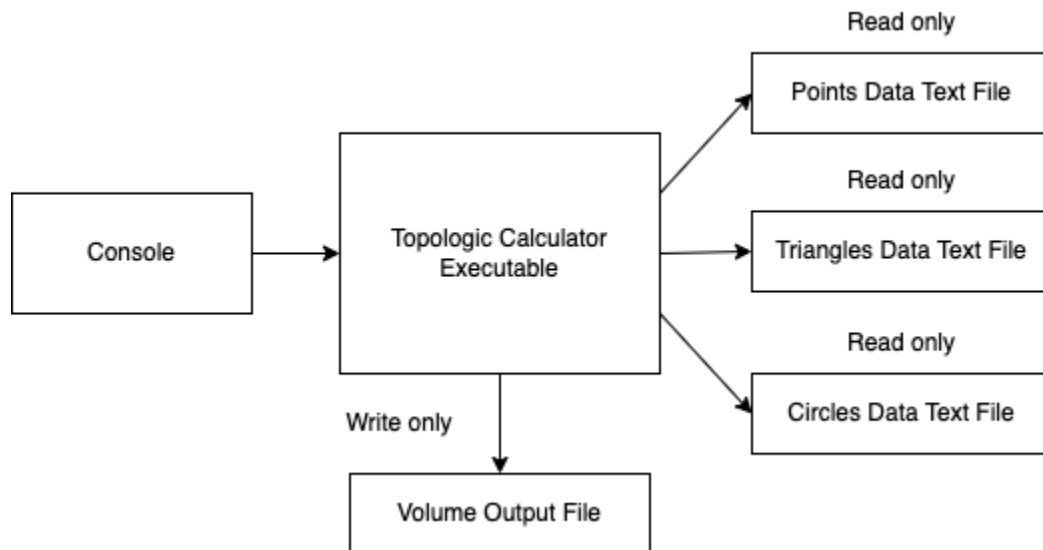
Ryan Porter: r_porter4@u.pacific.edu

Last Edited: Feb 5, 2022

System Architecture

This project would be compiled as a stand-alone executable. The intended running environment for this executable is an 8-core Windows Dell Laptop. The program is to be run from the command line, and needs four file paths as inputs (Points, Triangles, Circles, OutputFile).
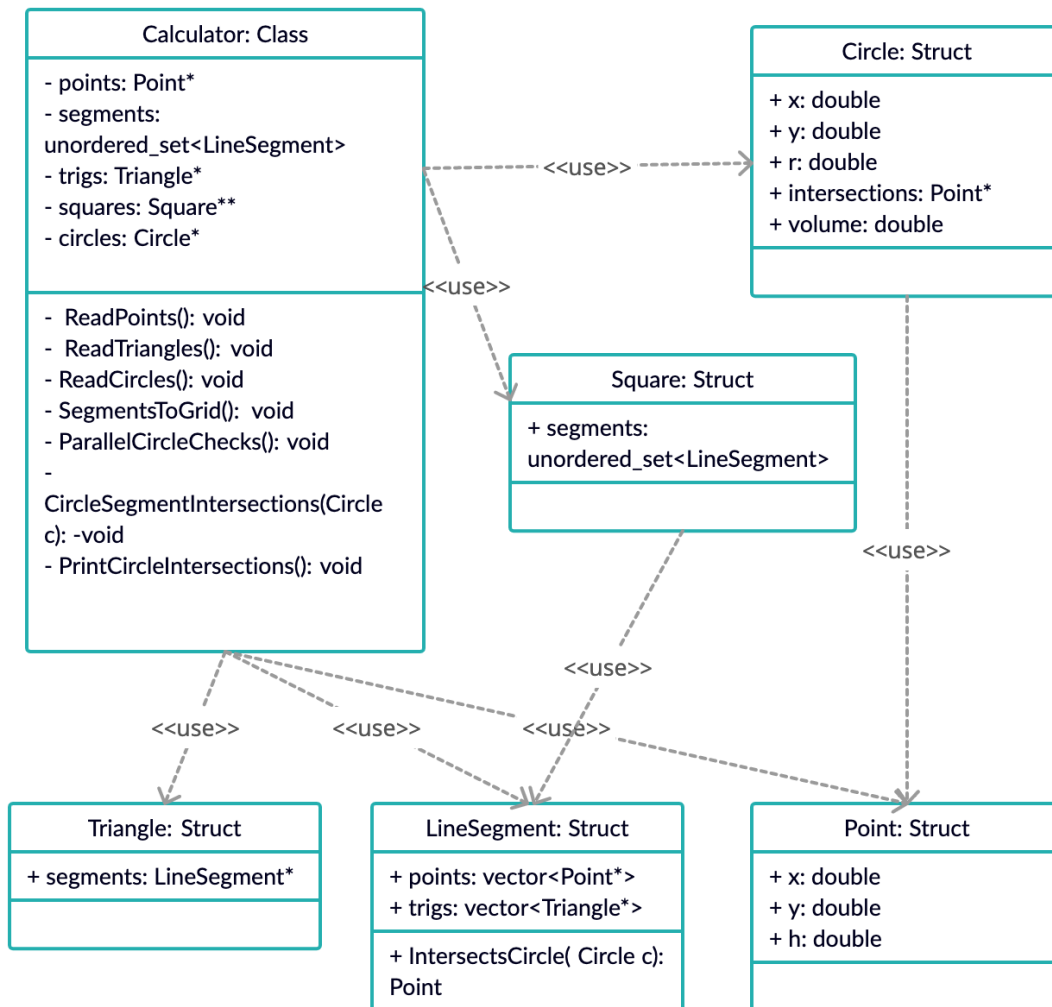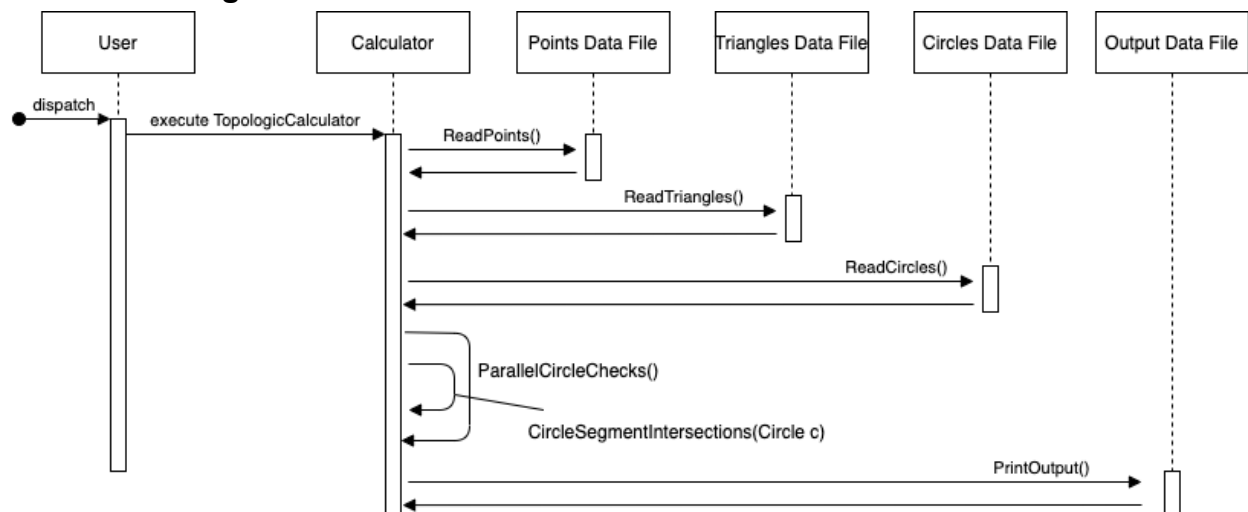

System Requirements

Windows Operating System
8+ Core CPU

UML Class Diagram

Calculator: Class

- points: Point*
- segments: unordered_set<LineSegment>
- trigs: Triangle*
- squares: Square**
- circles: Circle*

- ReadPoints(): void
- ReadTriangles(): void
- ReadCircles(): void
- SegmentsToGrid(): void
- ParallelCircleChecks(): void
- CircleSegmentIntersections(Circle c): -void
- PrintCircleIntersections(): void

Circle: Struct

+ x: double
+ y: double
+ r: double
+ intersections: Point*
+ volume: double

Square: Struct

+ segments: unordered_set<LineSegment>

Triangle: Struct

+ segments: LineSegment*

LineSegment: Struct

+ points: vector<Point*>
+ trigs: vector<Triangle*>

+ IntersectsCircle(Circle c): Point

Point: Struct

+ x: double
+ y: double
+ h: double

<<use>> relationships connect Calculator to Circle, Square, Triangle, LineSegment, and Point; Circle to Point.

Interaction Diagram

Lifelines: User, Calculator, Points Data File, Triangles Data File, Circles Data File, Output Data File

- dispatch
- execute TopologicCalculator
- ReadPoints()
- ReadTriangles()
- ReadCircles()
- ParallelCircleChecks()
- CircleSegmentIntersections(Circle c)
- PrintOutput()

Design Considerations

Due to the large amount of data this program will be dealing with, time efficiency is the highest priority, with a secondary priority on memory usage efficiency. Because of this, the program has been designed to calculate each circle individually, so that parallel processing can be used. Also, references are used instead of variables, to save on memory space, and data structures such as sets are used to leverage the power of hashing for constant time look-up.

Additional research will be done on time and memory efficiency, to validate my choices above, or to change them if I find something better.

User Interface

Users will communicate with the program through text files and the command line terminal. Before running the program, the user must have three data files prepared.

Points Data File:
- Each row will represent one point, in the format xPosition yPosition Height

Triangles Data File
- Each row represents one triangle, in the format Vertex1 Vertex2 Vertex3.
- Each vertex is represented by the array index number of the corresponding point

Circles Data File
- First line represents the radii of each circle, giving the maximum radius and the radius step size. For example 8 0.5 will make the calculator compute the circles with radius 0.5, 1, 1.5 … 7.5, 8.
- After the first row, each row represents one circle, in the format centerXPosition centerYPosition

References:
There were no references used at this point in time.