Code Challenge: C++ Geometry

Scope	Compiler Team
Status	Not started
Created time	@June 13, 2024 8:35 AM
Last edited time	@June 14, 2024 7:20 AM
Tags	

Prompt

We intend for this challenge to take about an hour of your time.

Please write a program which takes as an input a CSV file that contains the coordinates of a simple, concave polygon, and as an output generates the triangulation of that polygon, as well as it's area.

For this program, we would like to see a written implementation of the "Ear clipping method": https://en.wikipedia.org/wiki/Polygon_triangulation

Roughly speaking, the algorithm should look something like this:

1. load the polygon
2. while the number of vertices of the polygon > 3:
    a. find the next "ear"
        i. given a vertex (v), find the previous and next vertices (vprev, vnext)
        ii. create a line segment from vprev to vnext and check if that line segment is within the polygon
        iii. if the line segment is contained by the polygon, this triangle is an ear

    b. save the coordinates of the triangle, compute it’s area
    c. “clip” the ear by deleting the vertex v from the polygon
3. output all triangle coordinates, and the sum of areas

Language: C++, Rust, plain C, or another low level language preferred

Bonus points: if we can build and run your solution (e.g. on Ubuntu 22.04 LTS)

Context:

- We write a lot of C++ and Python (interop with Pybind11)
- We use the Google C++ style guide*
- We follow the ABSL Tips-of-the-week*
- We use as much automated tooling as possible (asan/ubsan, formatting, linting, etc.)
- *(where applicable to a small, scrappy startup)

We’re looking for:

- Design: Using the simplest tools and structures
- Knowledge: vector math and numerical methods
- Testing: concise and complete unit tests
- Readability: Good naming, concise comments for non-obvious code
- Collaboration: You’ll walk us through your solution and design choices

You can use any library you want for basic geometry, but we are looking for you to implement the ear clipping algorithm yourself.

No need to go overboard and write a unit test library or full featured vector math library. This is an exercise to start a conversation, not a control system to fly a spacecraft.

Keep it short and sweet! Nobody likes a 5000-line code review :-)

Sample input files are attached. You should find that the area of the simple concave poly is 0.65 and the concave poly is 1269.1874283009124.

simple_concave_poly.csv

concave_poly.csv