

//README>

16-Sep-15

Structure

I have set up a simple polymorphic structure which can be easily extendable.

The VERTEX class keeps track of a single vertex.

The EDGE class is an abstract class.

All curves are of type EDGE (inherit from). This particular example has just two children LINE and CIRCLE. All child class must override the two pure virtual functions;

- computeMachineCost(const unordered_map <int, VERTEX> &)=0;
- computeMinMaxDim(const unordered_map <int, VERTEX> &)=0;

The key class is the PROFILE class; each profile (user input) is converted into an object of type PROFILE.

A vector of pointers to EDGES represent the profile, this can be made up of LINES or CIRCLES using polymorphism.

The computeQuote function calls the computeTotal MachineCost and computeTotalMaterialCost.

computeTotalMachineCost calls the computeMachineCost on all elements of the ProfileEdges.

Similarly computeMinMaxDim computes the minimum and maximum X, Y locations for each element of the ProfileEdges.

This structure provides easy extensibility, new curves types can be added as inherited child class of edge and PROFILE will remain unaffected.

Assumptions

I have made the following assumptions.

- The input json is error free.
- The input json lists all the edges in a clockwise manner.
- The curve is a closed curve.
- The circular arcs make an interior angle of 90 degrees or 180 degrees. (See What can be improved Sec.)
- The rectangle cut is always axis aligned.

Dataset

I have added a circle.json dataset.

Running the program

The program uses **Boost Libs** to read json,

It also uses some c++11 features. (Please turn on `-std=c++11` option.)

I am using VisualStudio2012 on a windows machine.

Here is a sample run;

Enter file name (.json)

Rectangle.json

Total Quote for your object is 14.10 Dollars Only!

What can be improved?

The machine cost is computed correctly for the two test cases below,

The material cost is not computed optimally, for both the cases it will give the same material cost.

Ideally the material cost for the left case should be less.

I have an idea how this can be fixed.

Also the class should be made templated.

