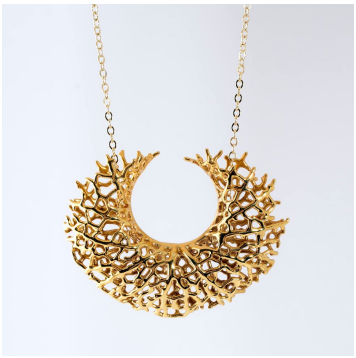
Program 2 - Procedural Modeling

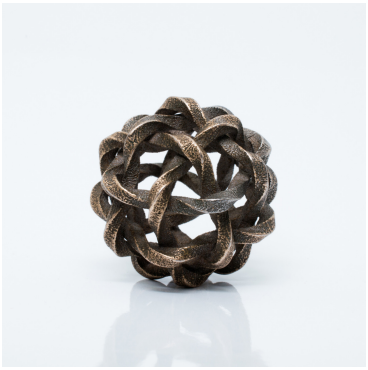
You must work individually on this assignment. To receive credit, push all required materials to the master branch of your BitBucket repository by March 30th.

Summary

Procedural modeling is a term used in computer graphics to describe methods for generating geometry using a set of rules rather than using manual modeling techniques. Some advanced examples of procedural modeling are L-systems and fractals. High end procedural modeling systems are used in animated films and video games to generate natural landscapes, forests, cityscapes and more. Modern manufacturing techniques such as 3D printing and CNC machining can be used to bring unique procedurally modeled products into the real world.



Nervous System's 3D printed hyphae pendant.



3D printed knot ball generated with Freakin' Sweet Knots.

In this assignment, you will be procedurally modeling something of your choice. One option that meets the requirements for this assignment is generating a brick wall. You must provide several customizable options to adjust features of your model. If you choose a brick wall, there are suggestions below for what your customizable options can be below. If you want to implement something other than a brick wall, you MUST come talk to me about your option choices so I can approve them.

Required Materials

Your program2 directory must include:

- All source code for the completed program
- A screenshot of your program displaying your procedurally modeled object
- A readme text file with any necessary instructions for using your program, along with a description of what your procedurally generated object is

Detailed Requirements

Your program must

Be an original program written by you. You may use code from labs as a starting point. You may talk with other students about the program, but looking at their code is not allowed.

Implement a virtual trackball. Use your virtual trackball code from the labs to provide an intuitive interface for viewing your model. You are encouraged to include other means of viewing your model, such as a first person control or a fly mode.

Shade your object using diffuse lighting. Diffuse lighting is a standard way to preview models in any 3D modeling software. Use a light source (or multiple sources) that provide a good visual preview of your model. You are encouraged to incorporate other lighting and shading features, such as specular lighting, textures, bump mapping, and/or normal mapping.

Have four configurable options to adjust your model. If your model is a brick wall, the four options can include brick size (width, height and depth), wall height (number of layers of bricks), brick spacing (the amount of space between bricks) and wall shape (a single wall, a complete square wall, a circular wall, etc. pick at least 2). If you have other ideas for your four options, come talk to me so I can approve them beforehand. If you choose something other than a brick wall, you MUST come talk to me about what your four configurable options are so I can approve them. I recommend talking to me well before the deadline to make sure your idea qualifies. It is up to you how the user is able to adjust your options. Qt sliders are an intuitive way to scrub options, see lab 4 for an example of using QSliders. Keyboard and mouse events are also ways you can adjust your options. Be sure to document how to adjust each option in your README file.

A readme file. Write up what your program does, and how to use it, along with any extra information you want me to know about your program.

Demonstrate creativity and self expression. Put effort into your program, beyond the minimum requirements. Experiment, put extra features in, organize your code exquisitely, it's really up to you. Explain what you did in your readme file.

Point breakdown

20 - Correct virtual trackball behavior
20 - Correct diffuse shading
40 - Four configurable options (10 each)
20 - General (code style, execution, creativity, self expression and readme)
---
100 total points