

Project Overview

This project is meant to allow users to more easily visualize 3D surfaces by creating 3D objects from 3D equations and other parameters. The user will enter a 3D equation for a surface and also necessary parameters to help create a 3D object (name of the object, resolution of the object, bounds for evaluating the equation). After the surface is approximated over the bound at the given resolution, the system will produce a STL file labelled with the name of the object that will be ready for 3D printing.

Terms

The following terms and acronyms are defined and used within this document.

Term/Acronym	Definition
tessellation /triangulation	the process of dividing a surface/shape by arranging closely fit shapes (triangles) in repeating patterns without gaps or overlaps (https://en.wikipedia.org/wiki/Tessellation)
STL file	STereoLithography file; commonly used in CAD software and 3D printing to describe the surface geometry of 3D objects (https://en.wikipedia.org/wiki/STL_(file_format))
resolution	the “choppiness” of the approximated 3D surface; how well the 3D object should approximate the 3D surface; high resolution makes the approximation closer to the surface
bound	a 2D shape on the XY plane over which the 3D surface given by the user’s equation will be evaluated

Requirements

Mandatory requirements are stated using *shall*, while optional requirements are stated using *should*. The requirements are in no particular order, but they are numbered for easy reference.

Functional

1. This system shall allow the user to enter and store a valid name for the solid, a valid equation for a 3D surface, a valid resolution for the approximation of the surface, a valid bound on the XY plane (circle, rectangle, or triangle), and valid information for the bound (vertices, radius). This bound shall describe the projection of the 3D surface onto the XY plane. This system shall validate input data as outlined below, and notify the user of invalid data.
 - 1.1. The name of the solid:
 - 1.1.1. A valid solid name shall be a non-empty String made up of any combination of spaces, dashes, underscores, numbers, and letters that is less than 80 bytes long, but does not solely consist of spaces. E.g. “My_1st Solid”, “1-234”
 - 1.2. The equation defining the 3D surface:
 - 1.2.1. A valid equation shall be of the form $z=f(x,y)$ in infix notation. It shall consist of only combinations of x and y variables with valid operators between them, the constant π , correctly matching parentheses, the math operators (+, -, *, /, ^), and functions ($\sqrt{}$, abs , \sin , \cos , \tan , \ln , \exp). In addition, this equation shall describe a 3D surface above the XY plane at at least some points. E.g. “ $x*\{-y/\ln(60-x)^2\}$ ”, “300”
 - 1.3. The resolution of the approximation:

- 1.3.1. A valid resolution shall only consist of valid integer values between (and including) 1 and 10. E.g. **"10"**, **"5"**
 - 1.4. The bound representing the lower bound on the XY plane to sit under the surface:
 - 1.4.1. A valid bound shall be a circle, a rectangle or a triangle.
 - 1.4.1.1. A circle bound consists of a valid radius and vertex (aka center point).
 - 1.4.1.2. A rectangle bound consists of four valid vertices that form a rectangle.
 - 1.4.1.3. A triangle bound consists of three valid vertices that form a triangle.
 - 1.4.2. A valid radius shall be a positive number. E.g. **"200"**
 - 1.4.3. A valid vertex/center point shall be a pair of numbers separated by a comma. Spaces shall only be accepted between a number and the comma and/or before or after the number. E.g. **"300,400"**, **"200 ,200"**, **" 100 , 0 "**
2. This system shall implement a tessellation/triangulation algorithm to use on the 3D model.
3. This system shall build a STL file of a 3D model representing the space between a valid 3D surface and a valid bound on the XY plane. The system shall produce and save in the application directory a 3D-printer-ready file in ASCII format that can be used to print the model. The format for a STL file is described here: http://www.fabbers.com/tech/STL_Format

Unimplemented Functional

4. The system should implement maximum height, surface depth, and flat/curvy bottom. These parameters may modify width/height and approximation of the surface.
5. This system should save the current 3D model data using the equation and the user's specifications in a database or file.
6. This system should save user profile settings in a database or file.
7. This system should load previously saved 3D models so that the user can modify characteristics of the surface and print the new model.
8. This system should allow the user to specify whether they would like an ASCII or binary STL file.
9. This system should deal with undefined singularities in the equation. E.g. **"ln(0)"**
10. This system should allow user equations below the XY plane.

Non-functional

1. This system shall be open-access source code so that it can be available for everybody, especially for learning purposes and demonstrative classes.
2. This system shall be developed using Java.
3. User interface of this system shall be intuitive and understandable for both faculty and students of all learning levels. If there are technical terms, those have to be explained to users through messages on screen in order for them to be able to respond.