

Joshua Learn (jrl2196)

11/22/2019

Parallel Functional Programming Project Proposal

Polygon Rendering

For my final project I am proposing to tackle the well-known problem of polygon rendering in computer graphics. This project will aim to take a scene of 3-dimensional polygons and their associated properties and render them on a 2-dimensional image using a simplistic Haskell graphics library. The computational work for this problem will come from transforming the abstract data about the polygons into a 2-dimensional image and then rasterizing that image with the appropriately colored pixels that will reflect the properties and shading of the polygons. There will be smaller subproblems that go along with these tasks, such as shading and computing what surfaces are hidden. I plan on using the Z-buffer method for hidden surface removal and the Phong specular reflection model for shading. These are tentative decisions and are subject to change if I find another model that better fits into a parallel functional paradigm.

My approach to this problem will need some method of taking a scene of polygons as input and producing a viewable image as output. I plan on taking polygon file format (PLY) files as input to this algorithm. I will need to be able to properly parse this input to turn it into some abstract representation. Once this has been done, I can proceed to run the transformation on the polygons and convert them to a 2-dimensional image. Finally I can rasterize the image and output it as a matrix of pixels. The key parallel portions will be transformation and rasterization, as this often involves billions of arithmetic operations for considerably complex images. For the implementation of the output, I am leaning in the direction of using a package like gloss (<http://hackage.haskell.org/package/gloss>) for outputting the final image.