

Research Fellow: Rosalie “Rosie” Coyle (2020)

Concentration: Undeclared

Faculty Mentor: Elodie Fourquet

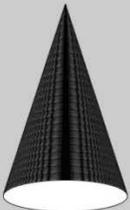
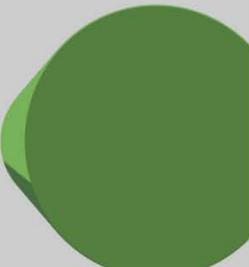
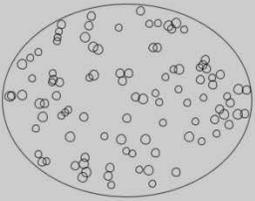
Department: Computer Science

Title of Project: Perspective Tiled Floors in Paintings for a 3D Blueprint in Processing

Project Summary:

I spent this summer writing tutorials for future students to use when working in a program called Processing. Processing is a computer software program that is based in a programming language called Java. This particular software is designed with artists in mind because it is easy to create many different shapes, such as ellipses, rectangles, and triangles, in any color that appear in a graphics window—a small window on the screen on for the images to be displayed. When using Processing, it’s simple to create works of art on a computer. It’s also simple to recreate works of art, which is the basis of the assignment that originated this research project. Our faculty advisor had, in the past, assigned students to recreate famous paintings using the Processing software. There were many difficulties that arose with this, as recreating 3-dimensional perspective using a 2-dimensional platform can be very tricky. To help future students struggle less when completing this assignment my colleague and I recognized ways that we struggled when recreating paintings and wrote tutorials of how to best tackle these complicated situations, along with some general advice on completing the assignment. In addition to writing tutorials, I also spent some of my summer research time learning the Java programming language.

I wrote seven tutorials this summer. The first of which was on coding cylinders so they appeared to be in perspective. To do this, the students need to think of the cylinder as a stack of ellipses that change size and positions slightly as they are stacked. By thinking about the cylinder like this, the student can write a for loop or a while loop that draws a new cylinder each time and changes the width, height, and the x and y of the center position. In following these directions, the end result should look something like Fig. 1. The second tutorial was on coding cones in perspective. This used a similar method to coding cylinders, but as the cylinders are stacked, the width and height change in such a way that either the first or last ellipse drawn has a width and height of 0. By doing this, it’s like a cylinder that comes to a point. The result of following these directions will result in an image like Fig. 2. The third tutorial I wrote was about creating shaded cylinders, that is, cylinders that are colored to appear as though there is shadow on them. To do this, the student follows the cylinder tutorial, but also codes arcs in different colors to make up the shaded outer wall of the cylinder. The fourth tutorial involved using Adobe Illustrator and described a method of outlining faces using its pen tool. This is a useful resource in conjunction with a tutorial my colleague wrote describing how to turn Illustrator creations into Processing files by saving them as SVG files. It can be difficult to translate the curves of face shapes into the simple shapes that are available in Processing, but using these two tutorials, it can become much simpler. My fifth tutorial explained how to use a Cylinder class that I had made that allows the user to create Cylinder objects and draw them with and without shading. My sixth and seventh tutorials explained how to constrain randomly generated shapes to an elliptical boundary. One demonstrated how to achieve this using Point and Dimension objects and the other explained how to achieve this without using them. The result of this process can be seen in Fig. 4.

			
<p>Fig. 1 This is the result of following my cylinder tutorial.</p>	<p>Fig.2 This is the result of following my cone tutorial.</p>	<p>Fig. 3 This is the result of following my shading cylinders tutorial.</p>	<p>Fig. 4 This is the result of following my elliptical boundary tutorial.</p>

Source of Support: AHUM Div. NASC Div. SOSC Div. UNST Div.
 Other (specify): Science and Math Initiative-SMI (NASC Division)