

Connector Editor

Benjamin Lentz

Bima Rakhman

CSS 450

Final Project Proposal

Overview

Our program will be a simulation to demonstrate several ways to transform primitives based on their structure in the world.

Initially, the user is presented with a “workspace” and multiple primitives on the side of the application; these primitives can be placed into the world via mouse clicks. The caveat of this program is when primitives are overlapped: a pivot will be created at a point of intersection and the objects will be “connected.”

Transformation Options

There are two types of modes of transformation to objects inside the workspace: “part” and “whole”. Essentially, the user will be able to modify all connected primitives or individual ones that they select. These modes are governed by mouse input to the main view:

Input	Functionality
Left-mouse click	Translate the last selected primitive to this point and modify connected primitives accordingly
Left-mouse click + drag	Translate selected primitive & connected primitives to current mouse point
Right-click	Select a primitive part
Right-click and drag	Rotate this selected part to the mouse position

World Views

Main view

The main view will be of variable length, according to a slider bar value. This way, users are able to have more/less room to construct and connect objects.

Secondary view

The smaller view will be slightly more complex. As the main view gets bigger and more objects are added to the screen, it may be difficult to move individual parts in all of the clutter.

Secondary viewing will show a zoomed in view of a box around the mouse pointer location. The idea behind this is to allow users to inspect the connections between primitives more closely.

Additional Objects

In addition to the pre-defined shapes of primitives, the application will also support our own “custom” primitives such as (but not limited to):

- Arms, legs (rectangles and circles)
- An entire “person”

These primitives will have increased functionality with their transformation behavior (i.e. moving the arm will move the palm with it).

Application Semantics

Since this application is a primitive editor, we plan to include a save / load function that saves all of the primitive information and their current state. This information will include:

- Geometry information
- Transformation information
- Render state information

