

CIS 510 Final Project

Areej Alghamdi and Jacob Lambert

December 1, 2015

Although we each implemented only certain features, we met several times to discuss the design and implementation of all the features. Jacob primarily worked on creating the face mesh, designing the Qt GUI, and setting up the callbacks in python. Areej primarily worked on creating individual deformers for each metric, and ensuring that the deformers worked together as required.

Maya Face Mesh

To create the face mesh, I first found a crude drawing of a human head to use as an image plane in Maya. The image plane helped with modeling realistic facial dimensions.

To begin the modeling process, I first created a polygon from the side perspective using the image plane. This polygon served as an outline of the head. After deleting extraneous faces, I extruded this outline to begin modeling.

For the eye and mouth areas, I created a cylinder and deleted unneeded surfaces. I then combined this cylinder with the outline mesh.

Throughout the modeling process, I attempted to use only quad faces. This required adding edge-loops and creating new polygons. Most of the polygons were created by extruding edges and merging vertices. I also used the "append-to-polygon" tool.

[face-mesh picture here]

Self-critique

While the face does look somewhat human, it needs some improvement. Currently the model has no ears. Also, when smoothing the mesh, I did not distinguish between "soft" and "hard" edges, resulting in over-smoothing in some areas.

Maya Individual Deformations

(Areej)

Self-critique

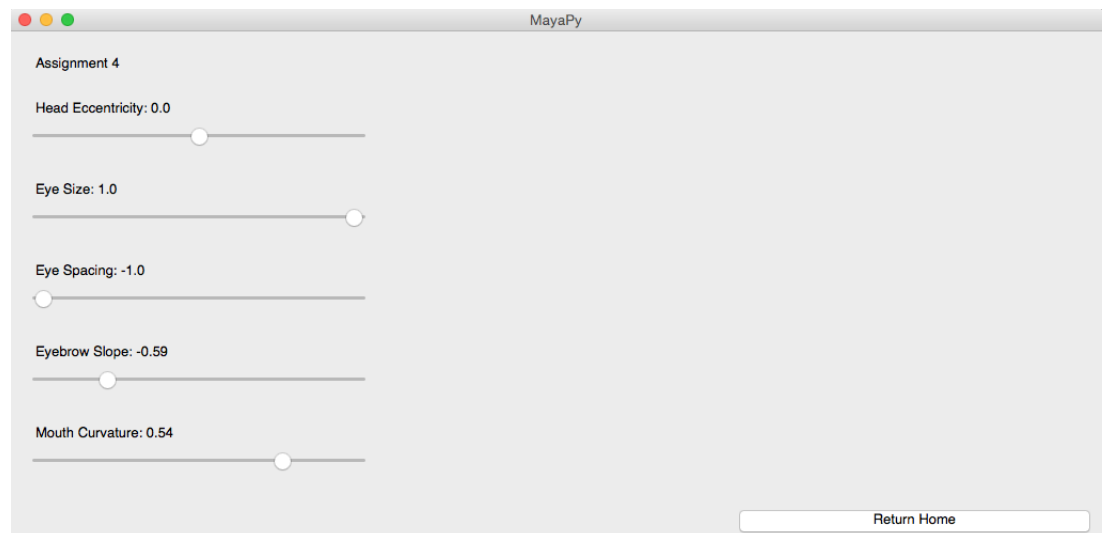
Maya Linking Deformations

(Areej)

Self-critique

Qt GUI

The Qt GUI consists is relatively simple. It consists of a text label and slider for each modifiable attribute. The label contains the name of the attribute and the current value assigned to the attribute. The slider ranges from -1 to 1, and is initialized to zero. The value of the slider represents the value of the associated attribute.



Self-critique

The GUI could allow users to manually input values for the attributes in addition to using the slider. This would require a more MVC based design to keep the slider and user-input values consistent.

Python

(Jacob)

Self-critique