Honglu Xu, Lei Li

CIS 410 Graphics Programming in Python

Final Project

Report

I. divide labor/responsibilities

Briefly:

Honglu handle modeling and deformers creating. (maya part)

Lei Li handle interface design and coding.(PyCharm part)

Together: online searching and update repository

How did we cooperate together:

we will set up a goal together, then we will do some research from the books and internet, then we think about the detail and implementation step-by-step.

Mostly, Honglu xu is responsible for modelling the face and creating the deformers.

In the meantime, Lei Li will consider the UI Layout and creating the new Widgets.

when Honglu Xu finish the face modelling, Lei Li will respond to connect the deformers between Maya and Qt by PyCharm.

Finally, we will adjust the parameters carefully until it achieve our expectations.

II. What aspects of the project are regarded as high priority and what are low priority

High priority aspects:

Creating a good interface is not only a good thing for the user, but also can be an efficient tool for the developer, so “QTdesigner” and the interface we built will be a high priority thing. Also, since a good start is really important, we will need a good 3D model to test our work at the beginning and in the future. Thus, we will build up the face model first, that’s the first priority.

Low priority aspects

Details are really costly to satisfy, especially for the 3D model. We might need to spend a large amount of hours just to build a good face that we feel perfect or even just good. Thus, the details for the 3D model and the artistic aspects for the interface will be our lower priority.

III.outline

New widgets:

Widget names	functions
emotion	1.set a surprise face, 2.set a sad face, 3.go back to home page, 4.reset the face to original face

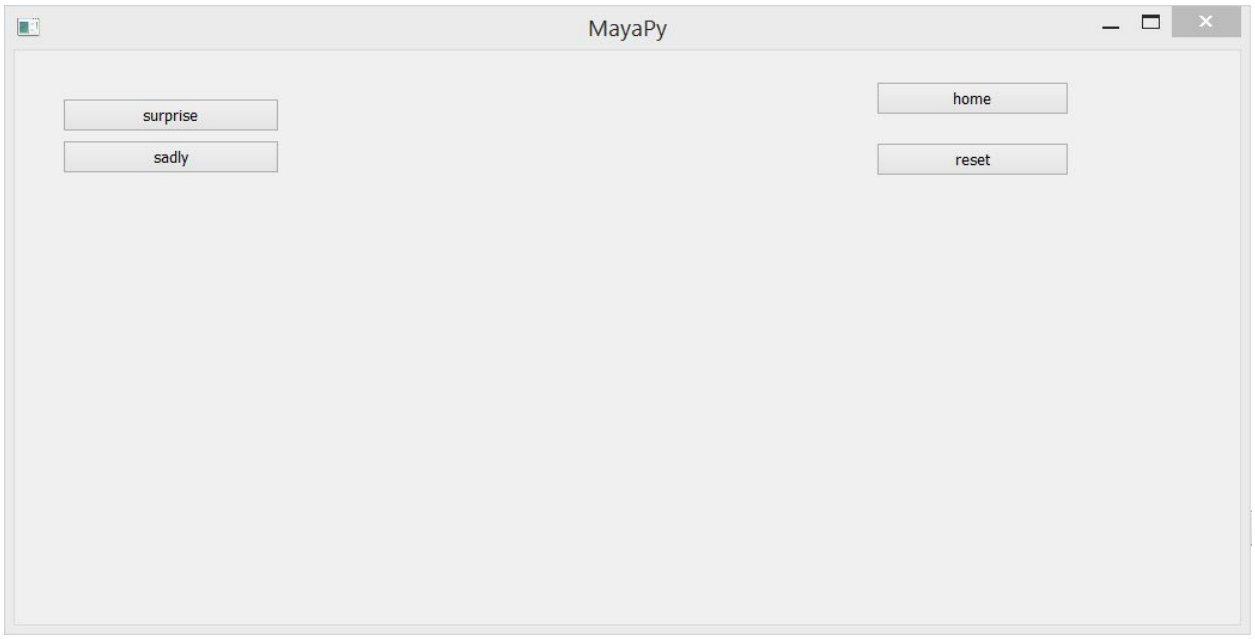
smile	1.set adjust the smile of mouth, 2.set adjust the smile of eyes, 3.set adjust the eyebrows, 4.set adjust the mouth open, 5.set adjust the angry of mouth, 6.switch to next page, 7.go back to home page
smile2	1.set adjust the head high, 2.set adjust the face width, 3.set adjust the chin size, 4.set adjust the nose size, 5.switch to previous page, 6.go back to home page

screenshot:

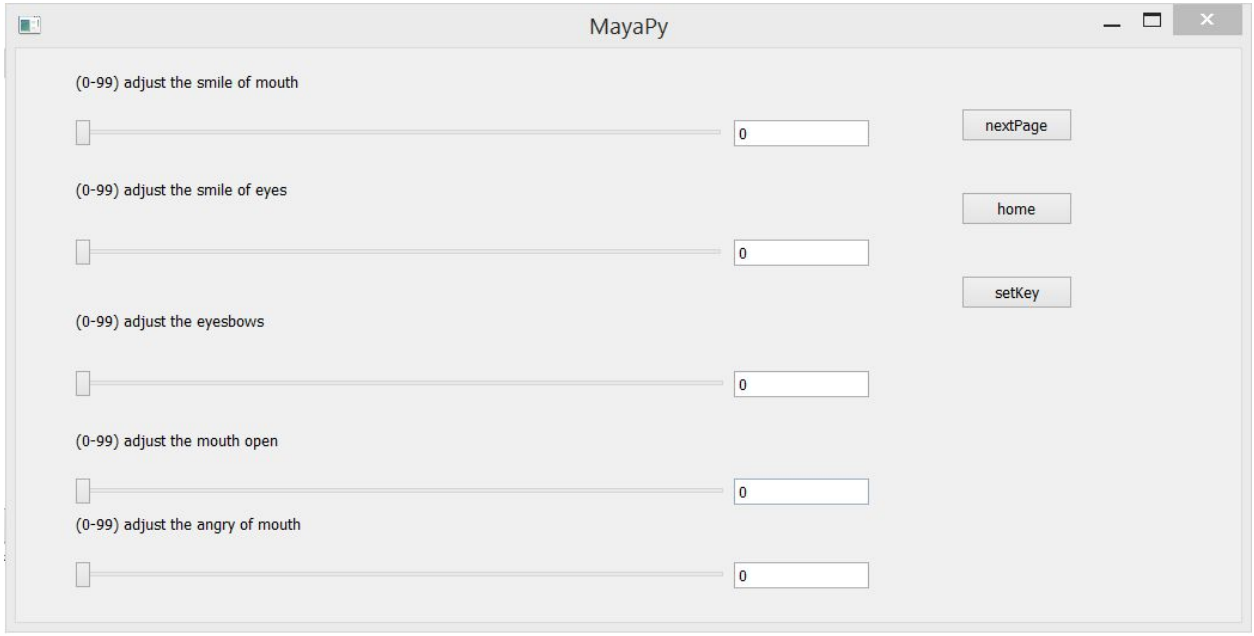
main control

:

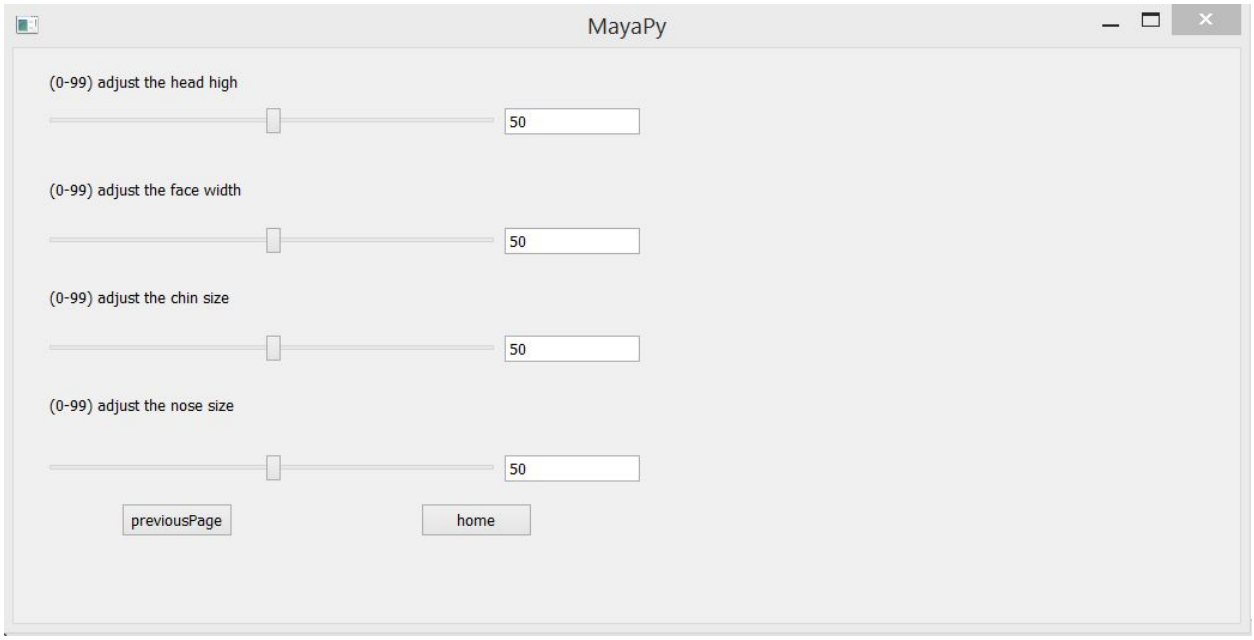
emotion:



smile:



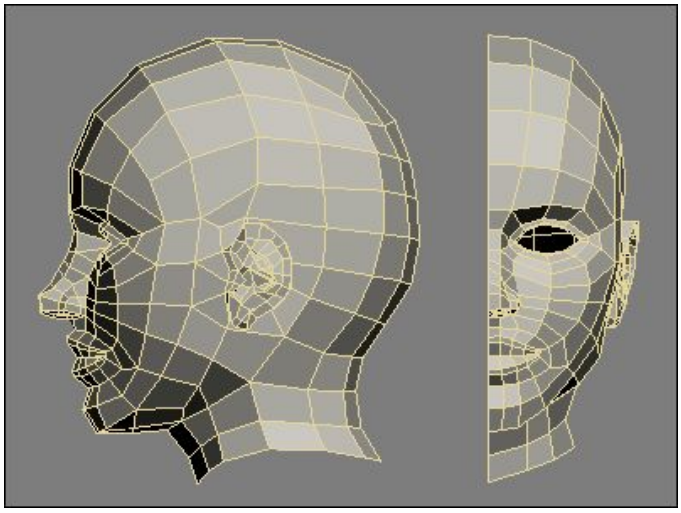
smile2:



IV. Details

Face:

Firstly, we created a face by ourselves. It is not so smooth to do the job but we found a way to complete it. We download a face picture online and used it to complete a face. This is our source picture:



We used this side face to create a side face ourselves and then use front face to adjust the vertices to create a 3D half-model. This is a long but fun part. Then, we used the mirror instance to create the other half and everything is good for a model. Lastly, we used the merge function to fix the cracks between polygons and it looks perfect right now.

Eyebrows is our next step. it is really not easy to implement. We learned a lot of methods to create a eyebrows, but at last, we used a wire tool to accomplish this. We draw a cv curve and a circle, and use the circle to create a surface through the path of that curve. So, when we adjust the control vertex of the curve, the whole eyebrow will be reshaped which is very nice to implement the eyebrow.

Lasty for the modeling, we created the eyeballs to make it looks better. For the eyeball, we created a sphere and added a new material which is an image on it. This is our image for the eyeball:

We downloaded it from the web site and adjusted it so it can fit the sphere.

Smile:

After we created a nice face. The first we do is make it smile. So we add several cluster deformers to accomplish it. We select the right corners of the mouth as the center, and apply a cluster there, using the Envelope = 1.0; then since the force will be weaker in other parts of the face, we select a outer circle to apply the cluster and using Envelope = 0.8; and do it again and again for the outer side vertices. Thus, at last, when we move up all the clusters we created for the mouth, the mouth will smile.

For the eyes , basically we did the same as the mouth. And since we also need to widen the eyes when we smile, we also set a cluster for the X-transfer at the corner so the eyes can be widened.

Also for the eyebrows, since it can not be attached to the face simply. We need to adjust each control vertex on the curve so the eyebrow will move with the face moving.

QT:

Then we trying to set an interface to make it easier for the user to adjust the smile face:

firstly, create a new folder in MayaPy/resources/widget/smile, then create a Widget.ui file with Qtdesiner, and put it to the smile folder. secondly, we create a new file which named “smile.py” and put it in src/mayapy/views/assignment2, In “smile.py”, we will create a class called smile. such as:

```
class smile(PyGlassWidget):
```

and in `def __init__(self, parent, **kwargs):` we will connect all the elements which contains in Widget.ui.

finally, we will go to “MayaPyMainWIndow.py” file, import `from mayapy.views.assignment2.smile import smile` for the header, and register our new class “smile” in _init_ function:

```
def __init__(self, **kwargs):
  PyGlassWindow.__init__(
      self,
      widgets={  ...../// some class we already have before
```

```
    'smile':smile, // we register our new class here
title='MayaPy',
**kwargs )....../// the rest code
```

now our new window are ready to run in PyCharm.

Going on:

After we finished the smile action. We set up to create more functions for our model face. We used the control vertices on curve and the clusters on the mouth to adjust the eyebrows and the mouth, the user can create an angry face very easy. Also, we separated the up and low lips and add the many clusters on both of them. In the QT, we move the up lips cluster to up and the down lips cluster to downside and the face can open its face very nicely.

(Our face with mouth opened)

Shape:

After we created many interfaces for the emotions, we started to build another interface for adjust the shape of the face. The QT layout is same as before. For the face wide, we created adjust the clusters on the side. So it can be dragged to widen the face. Nose size is a little complicated:

enlarge nose:

nose top Z -> +1.5; nose upside Y -> +1.5; nose upside Z -> +1.5; nose right-side X -> +1.5; nose right-side Y -> +1.5; nose second upside Y -> +1.0; nose second upside Z -> +1.0 ……

shrink will be the same but change the number to negative.

Fixed emotion:

To make user feel more easier to use our interface. We created two emotions by our own. We dragged the vertices and clusters to make the face looks more like it is sad or happy. Also a reset button will set all the clusters to the normal, so the users will no worry about clicking on the wrong button.

KeyFrame:

Lastly, we set a button to create a keyframe on the Maya. So the users can create the keyframe and make animation for with our face anytime.