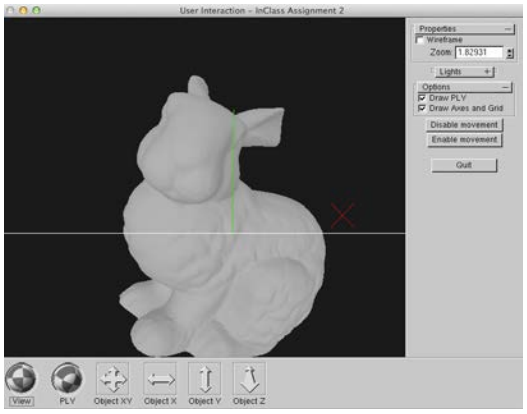
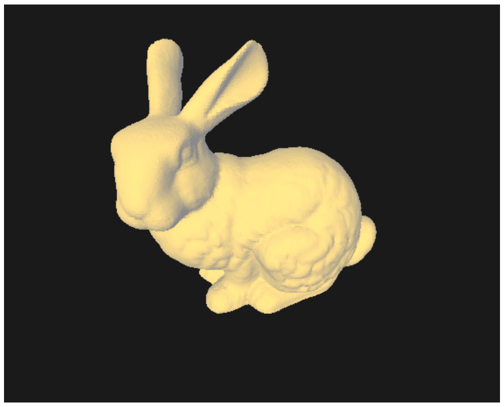
User Interactions in OpenGL




From the screenshots above, you can imagine how a tool with widgets can help an artist get the scene just right.

Description:

Previously we've rendered 3D shapes with an interactive interface. Today we will be exploring more of the functionality of the GLUI library. Our goals are to understand how event-based systems work. This lab will serve as an introduction to building interfaces and building tools useful for interactive graphics applications.

Your Task:

- You will start with the interface displayed in the image above. Use your imagination to then add other widgets in the scene.
  - View http://glui.sourceforge.net/ to see examples of what is possible.
- Work with mouse positions and keyboard
  - Three functions to modify: myGlutKeyboard(), myGlutMouse(), and main() as you add widgets.
    - Left/Right Mouse should scale or zoom into the model
    - Transform the camera with 'w','a','s','d' keys. (Move the camera forward back, and side to side)
- Work with (control_cb)
  - Understand live variables versus callbacks in the GLUI system
- Spend some time looking at the source code to see how things are done.
- Be creative, this lab is more open ended.

Files Given:

main.cpp – You will be working in the main.cpp. Within the main() function you will find lots of examples of the GLUI interface.
ply.cpp, ply.h, entity.cpp, entity.h, geometry.h – These are all helper files you do not need to modify.

Compiling: (On the Mac)

```
g++ -Wall -Wextra main.cpp ply.cpp entity.cpp -
I/Library/Frameworks/GLUI.framework/Headers/ -framework OpenGL -
framework GLUT -framework GLUI -o ui
```

Running:

```
./ui
```

C++ Refresh – Using an API --GLUI:

How Live Variables Work (References)

1. Live variables in automatically synchronized to a control. As you update some property in the control, the variable will be updated.

Callback (How these work)

2. A callback is a function that is passed as an argument in another function. In glut we have functions that catch keyboard and mouse events for example. When they are triggered, we can call into a function to perform an appropriate action.

Finished early?

- For this lab, consider how you might load multiple PLY objects into a listbox, and then be able to select which ones properties you are editing.
- Add some other manipulations so you can move the camera around (Have a camera look at function)
- Understand and work out by hand some of the camera transformations (glMatrixMult versus glTranslate)
- Look for some GLUI examples (provided with the GLUI library) and find interesting widgets to add.
- Try to make the keyboard movements smooth.
  - (Try adding acceleration or deceleration)
- Add some textboxes so we can input the global position, rotation, or scale of our model.
- Thing about your favorite videogame and what each mouse or keyboard command could do.
  - Not a PC gamer? How would you simulate a joystick or other device using the mouse, keyboard, or other widgets?

Going Further:

There are many similar windowing libraries to GLUI for OpenGL such as wxWidgets, QT that offer many other widgets for your interface. Beyond the user interface, you may also consider other input devices (joysticks, controllers, wiimotes, haptic devices, gesture based movements, and more).

Reference:

[1] GLUI Manual [link]