Visualisation Course 2008, UG4/M.Sc

Handout 1 – Introduction to VTK on Linux

If you are familiar with Linux and with computer graphics, you'll want to skip most of this exercise. However you may want to play around a bit with VTK just to get used to the syntax.

Firstly, VTK (The visualisation toolkit) is installed on the DICE machines. You will need to register for an account if you don't already have one. If you have a Windows PC at home you can obtain a windows version of VTK from the site: http://www.vtk.org. The syntax is the same, but you may need to set the paths for the data directories up carefully.

VTK is an object-oriented library of commonly used operations in Visualisation as well as a framework for representing and manipulating data. There are a number of interfaces to the library. If you prefer you can perform the exercises using the C++ or Java interface to VTK. It has its advantages, but the principle drawback is the very long and slow compile and link process for each minor change made to the program.

For this course the examples will use the Tcl/Tk interface to VTK. Tcl is an interpreted language, so there is no compile time – changes to the program can even be made interactively. VTK objects have been compiled for speed, but their interfaces are interpreted. You're quite welcome to use C++ or Java if you want to as the bindings are there for both. If you decide to use one of those, you can skip the rest of this exercise which refers to Tcl.

The Tcl interpreter for VTK on DICE is started by simply typing `vtk` in the command shell. The libraries are in `/usr/lib/vtk-5.0` and the headers are in `/usr/include/vtk-5.0`. Examples, data and documentation will be uploaded to the course website soon.

The TCL / VTK Interpreter

Download and save the source file `cone.tcl` from http://public.kitware.com/cgi-bin/viewcvs.cgi/*checkout*/Examples/Tutorial/Step1/Tcl/Cone.tcl?rev=1.6
Look at the file in an editor (or with the `less` command):

```
# create an actor and give it cone geometry

vtkConeSource cone
      cone SetResolution 8

vtkPolyDataMapper coneMapper
      coneMapper SetInput [cone GetOutput]

vtkActor coneActor
      coneActor SetMapper coneMapper
```

You can see there is a vtkConeSource object created called 'cone' and its resolution is set to 8. There is also an actor and a mapper object created to display the cone.

Start up vtk and load up the cone file. You can either do this by typing `vtk cone.tcl` in the linux command shell or you can load the file up after vtk has started by typing `source cone.tcl.` Ignore the error messages that are produced for the moment.

You should now see a 3D window appear with the cone present. You can rotate the cone with the left mouse button and zoom in or out with the right. You can also type Tcl commands into the command interpreter window and they will be applied immediately without any tedious re-compile or similar operation. Press 'u' to bring up the command interpreter window. Note there is a history mechanism in the command interpreter window below the command prompt.

The cone has 8 sides (i.e. polygon resolution 8). Try changing the resolution of the cone to 12, either by editing the cone.tcl file or entering a new resolution setting into the interpreter window and note what happens. You have to update the display for changes to take effect; the easiest way of doing this is to rotate the model slightly.

Like other computer graphics applications, vtk represents surfaces of objects by planar polygons – usually triangles. In order to represent a smooth surface, such as the side of a cone, many polygons are used to approximate the smooth surface in a piecewise-linear manner. Try increasing to resolution to 100 and note how much smoother the cone appears. Also note the penalty – the cone may rotate in a much more jerky and sluggish manner – depending on your graphics hardware (most modern DICE boxes should cope OK, however).

To display all the methods associated with an object, type the name of the object followed by 'ListMethods' into the command window. For the cone, you can see there are a number of methods associated with parameters of the cone, such as the height, radius and angle. Try experimenting with varying these in the same manner as resolution. You can, of course, also make changes to the cone by editing the script file.

Interactive Change of Resolution

Using the SetUserMethod method of the vtkRenderWindowInteractor object (presented in lectures) can you alter your vtk code for `cone.tcl` to increase the resolution of the cone with every press of the 'u' key, instead of displaying the command interpreter as this action does at the moment. The method 'Render' of a vtkRenderWindow object forces the VTK display to be redrawn without the need to rotate the object.

The SetUserMethod is now (or soon to be) depreciated in VTK and as such produces an error message as encountered in the example in lectures. To avoid this problem the following method syntax can be used as a drop in replacement:

```
iren AddObserver UserEvent { # code to execute on press of key 'u' goes here }
```

Together with the "Render' of a vtkRenderWindow object edit your example to increase the resolution, and automatically re-display (i.e. render) the with every user key press 'u'. Additionally you could try mapping it to increasing in cone height and radius.

Documentation

Documentation for tcl/tk is available at http://www.tcl.tk/man/tcl8.4/TclCmd/contents.htm
Documentation for VTK is available at http://www.vtk.org/doc/release/5.0/html/

There are several copies of the book in the library, but note these refer to VTK 2.4, although the changes are minor. There are many examples available, as well as documentation on individual methods on the www.vtk.org website.

Taku Komura 07/01/08 (based on previous exercise by Toby Breckon)