Interactive 3D Graphics Project 1

Claudio Desideri   110059
Luca Foschiani   109623

The user can choose which form of table data to display, 3D bar chart, or pie chart, or area chart. He can also choose which sv file to open.

The application is able to parse a csv file obtained from a server and automatically visualize the corresponding graph. The only file to be run is index.html and has to be run on an http server. Our project has been tested with Chrome and Apache / Linux environment.

The camera is initialized in a position where all the graph is visible (initially), when the window is resized the projection and viewport mapping are adapted accordingly. Camera uses TrackballControls directly taken on threejs folder where OrbitControls are.

The user can rotate the camera around the displayed 3D graph, zoom in and zoom out. We chose to use trackball controls for a smoother rotation and zooming.

We use PhongMaterial because it can interact (reflect, emit etc...) with light sources.

The geometries that we use are:
-bar chart: PlaneGeometry (background structure), CubeGeometry (bars), we use a BoxHelper for the bars' edges and Line for comparison between bars.
-area chart: PlaneGeometry (background structure), for the areas, we use an extruded ShapeGeometry with blunt edges.
-pie chart: the pie is generated through extruded ShapeGeometry, each of which represents a slice. For each slice, we use Line to draw the slice's edges

All the charts use transparencies, the colors of all the charts are pre-made and the application just use colors in a fixed order, restarting from the first when the last is used.

In the area and bar charts, we added a background structure divided in levels, each of which is labelled with a value based on the maximum value of the bars contained in the graph.

The application adds labels to che charts displaying the attribute associated to every row and displaying the value of a certain bar or part of an area when the mouse is over that part of the graph.
We wrote a function that generates TextGeometry aligned as preferred (center, left, right) in order to use them in labelling phase.

In the bar chart, we added 4 lines per bar, which can be displayed by moving the mouse over them, this helps the comparison between bars.
In the pie chart, the user can click on a slice to slightly move it outwards and display the value associated with that slice.

In order to make the load/unload possible in real time without destroying the whole scene/renderer we make use of dat.GUI for our GUI system in conjunction with JQuery that helps us a lot with event management. When change of chart is performed we simply destroy every object created and reload everything except lights, camera, controls.

We wrote the code taking advangate of mercurial as DCVS. Link with all changes is here:
https://code.google.com/p/int3dp1/source/list