Interactive 3D Graphics 2014 - University of Udine
Roberto Ranon

First coding project

Goals of the project

Your task is to develop a web application that displays 3D graphs from numeric data in form of a 2D table. The application should be able to:

- visualise the table data in the form of a 3D bar chart, or pie chart, or area chart;
- allow the user to rotate the camera around the displayed 3D graph, zoom in and zoom out;
- allow the user to highlight one particular data (or row of data, in the case of area charts) by clicking with the mouse over it.

Requirements

- the 2D table numeric data can be directly hard-coded in the javascript code. Ideally, our application should be able to read a *csv* file from the server, parse it, and visualise the graph, but this is not mandatory;
- the user should be able to choose which kind of chart to visualise through a menu in the page;
- the chart scene can be constructed using any kind of three.js geometries;
- colors in the chart can be decided by the application. Choose a suitable set of colors so that the visualisation is clear. Consider the use of transparencies when they are useful to better understand the graph;
- the camera should be initialised in a position where all the graph is initially visible, and the projection and viewport mapping should be adapted when the browser window is resized to avoid distortions.

Suggestions

- to implement menus in the page, you can take inspiration from this three.js example (http://threejs.org/examples/#webgl_geometry_subdivision) for a simple solution, or use the dat.GUI library (https://code.google.com/p/dat-gui/); an example of using dat.GUI with three.js can be found at http://www.realtimerendering.com/udacity/?load=demo/unit3-tessellation-demo.js
- to rotate the camera around the graph, you can use the javascript code provided with three.js and demonstrated by the example http://threejs.org/examples/#misc_controls_orbit[1]
- to implement the highlight, you can: (i) use the technique demonstrated at http://mrdoob.github.io/three.js/examples/canvas_interactive_cubes.html to figure out which object was selected; (ii) on selection, change its material to a brighter one.

[1] this code does not fully work on my Mac with Safari. It works ok with Chrome. You are required to test your project in Chrome before submitting it.

How to submit the project

- all relevant files should be zipped to a file named with your surnames and sent by email to roberto.ranon@uniud.it by ***tuesday, april 8***. The subject of the email must be "Interactive 3D graphics project 1 - Student1 surname - Student 2 surname"
- include also in the zip file a brief english description (in a text file) to report which of the required goals and constraints your application satisfies, or any other additional capabilities you have implemented
- code should be adequately commented (in english)

Grading policy

- a web application that full fills all goals and requirements, works reasonably well, and with reasonably readable and logical code will get a 30/30 mark
- each goal or requirement not fulfilled, or bug, or problem in the application will decrease the evaluation by 3/30. You cannot resubmit corrections
- additional capabilities could increase the mark (even above 30/30), but this will be considered on each case basis
- a delay of 1-3 days will decrease the mark by 3/30. Any further delay means you will have to do the full exam in any of the official dates (but remember that project must be submitted by that date in any case).