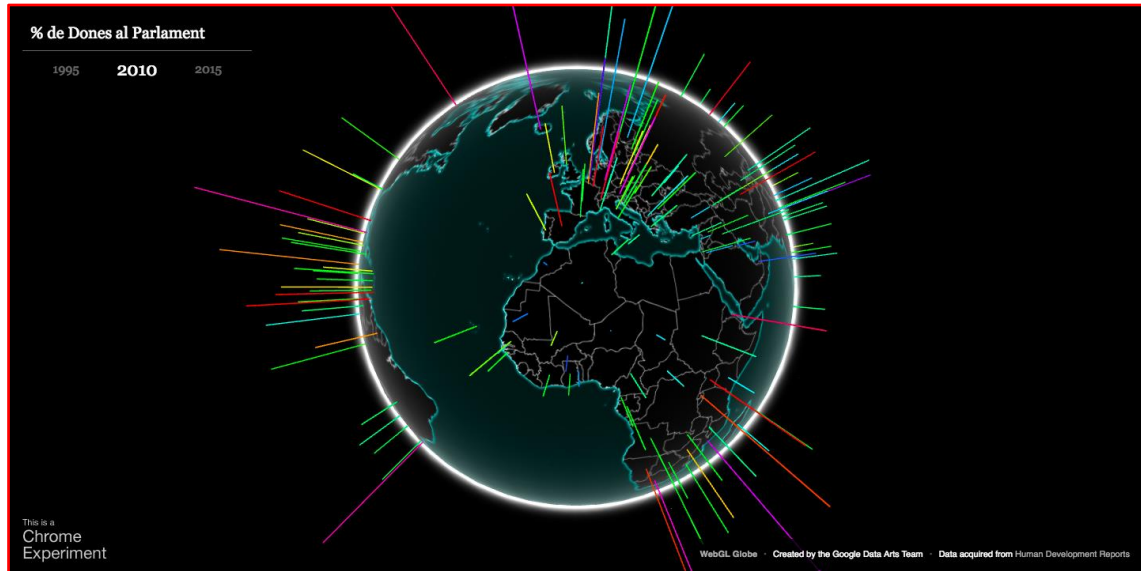


Pràctica 3

Visual Analytics



Blai Ras, Jordi Piñana i Alex Campos

Objectius

Aprendre a visualitzar BigData geolocalitzada en una aplicació WebGL. Concretament, com visualitzar en un navegador dades 3D geolocalitzades procedents de diferents fonts en un globus del món. Aquesta visualització es basa en WebGL, utilitza shaders i està incrustada en un html mitjançant javascript.

També volem treballar la forma de visualitzar diferents dades, canviar algun dels continguts de la plana html i alguns aspectes de la visualització (com el color de les dades mostrades o les textures associades al globus terraquí).

4. Tasques a realitzar per a practicar:

4.1. Canvia el color de visualització de les barres associades a cada dada

La definició d'un color per a cada barra es realitza amb la funció ColorFn. De totes maneres, el codi base usa aquesta funció de manera que el color depèn de les dades. Concretament, agafa una tonalitat o una altre segons el valor de la dada. Per exemple, els valors més baixos són blavosos mentre que els més alts són vermells/liles. La paleta de colors es pre-defineix en la funció.

Per tant, si volem un altre color canviaríem aquesta paleta i posaríem en hexadecimal el color desitjat. Nosaltres ens hem limitat a canviar la paleta de colors, ja que ja ens està bé que el color depengui del valor de la dada.

4.2. Canvia l'alçada de la barra a la que es refereix cada dada

Tal i com podem veure a la línia 209, l'alçada d'una barra és directament proporcional al valor de la dada, és a dir, s'escala segons les dades. Per tant, si volem canviar aquesta alçada, podem fer que no depengui de les dades (seria estípit) o canviar el factor d'escalació, que és el que hem fet. Concretament, de multiplicar per 95 (on algunes barres de valors molt alts no es veien) hem passat a 50.

4.3. Canvia la textura del globus terraquí. Pots agafar alguna de les textures utilitzades a la pràctica 2 o de la que trobis per internet.

La textura del globus és defineix en la línia 107, on creem una nova textura passant-li el *path* de la imatge desitjada. Aquesta més tard es passa al *shader*.

Ens hem baixat una textura de la terra més "guay", anomenada *worldGuay* que pots trobar a la carpeta *images/*. Canviem el *path* i ja podem visualitzar la textura.

4.4. Canvia el json per a que es visualitzin altres dades geolocalitzades.

Hem seguit el tutorial del enunciat de la pràctica. Les dades que hem triat les hem tret de la mateixa web que aquest tutorial, i són el percentatge de dones al parlament des de l'any 1995 fins el 2015. Per veure una diferència significativa, hem triat els anys 1995, 2010 i 2015. Aquestes dades estan en format CSV. Per adaptar les dades al format que necessita...:

```
[  
    'seriesA', [ latitude, longitude, magnitude, latitude, longitude, magnitude, ... ]  
],
```

Hem fet els següents passos:

1. Reduir la llista de països
2. Anar a la API de Google Maps, crear un projecte nou i crear una *Geocoding API*. Agafar la *key* d'aquesta.
3. Realitzar un *script* amb python que usa la llibreria Geopy. Registrar la nostra *key* a Geopy. Per cada país, obtenir la seva latitud i longitud. Per obtenir la llista de països, usem *pandas* per importar l'Excel.
4. Posem aquestes dades a un Excel nou, de manera que tenim 5 columnes: latitud, longitud, percentatge de l'any 1995, del 2010 i 2015.
5. Guardem aquest Excel com un .CSV. Tornem a python. Ara, importem aquest CSV, i per cada *row*, posem en una llista la latitud, longitud i valor. Se'ns creem 3 llistes, una per cada any. Imprimim les llistes com si ja es tractes d'un *json*. Copiem l'*output* i creem el *json* final.

4.5. Canvia el html per a que es correspongui amb les dades que està visualitzant amb aquest nou JSON.

A la línia 164 de *index.html*, especifiquem el *path* del *json* que hem creat i llestos. Si es visualitza alguna línia en una localització errònia és degut a la *Geocoding API* que a vegades s'equivoca.

5. Per al lliurament escull alguna de les següents opcions:

5.1. Explora el projecte globe-search on es visualitza la informació per colors dels llocs segons el llenguatge que es parla. Fixa't com s'utilitza globe.js de l'apartat anterior i com es canvia el comportament de la funció colorFn per a poder tenir una paleta de colors. Podries canviar la visualització de les teves dades per a utilitzar una paleta de colors?

El codi té una variable 'opts' que recull a l'hora de crear un nou 'Globe' les funcions que li passem com a paràmetre. A globe no li passem aquesta funció al html i llavors carrega els colors definits al globe.js.

A globe-search es passa com a paràmetre la funció colorFn amb unes etiquetes i la paleta de colors.

```
{
  colorFn: function(label) {
    return new THREE.Color([
      0xd9d9d9, 0xb6b4b5, 0x9966cc, 0x15adff, 0x3e66a3,
      0x216288, 0xff7e7e, 0xff1f13, 0xc0120b, 0x5a1301, 0xffcc02,
      0xedb113, 0x9fce66, 0x0c9a39,
      0xfe9872, 0x7f3f98, 0xf26522, 0x2bb673, 0xd7df23,
      0xe6b23a, 0x7ed3f7][label]);
  }
});
```

Index.html de globe modificat

```
var years = ['1990', '1995', '2000'];
var container = document.getElementById('container');
var globe = new DAT.Globe(container, {
  colorFn: function(label) {
    return new THREE.Color([
      0xd9d9d9, 0xb6b4b5, 0x9966cc, 0x15adff, 0x3e66a3,
      0x216288, 0xff7e7e, 0xff1f13, 0xc0120b, 0x5a1301, 0xffcc02,
      0xedb113, 0x9fce66, 0x0c9a39,
      0xfe9872, 0x7f3f98, 0xf26522, 0x2bb673, 0xd7df23,
      0xe6b23a, 0x7ed3f7][label]);
  }
});
```

```
DAT.Globe = function(container, opts) {
  opts = opts || {};

  var colorFn = opts.colorFn || function(x) {
    var c = new THREE.Color();
    c.setHSL( ( 0.6 - ( x * 0.5 ) ), 1.0, 0.5 );
    return c;
  };
};
```

En el nostre cas no podem fer aquesta modificació ja que les nostres dades són percentatges i difícilment n'hi ha dos iguals. En llengües té sentit aplicar-ho.

5.3. Explora com canviar la forma associada a les dades. Ara es pinta una línia de l'alçada proporcional al valor que es pinta en aquella posició del món. Com canviaries aquesta línia per una altra forma?

Es crea la forma de les dades amb THREE.*Geometry(params), en aquest cas hem canviat les barres per uns escuradents utilitzant un cilindre (la part que esta comentada).

```
mesh = new THREE.Mesh(geometry, material);
mesh.scale.set( 1.1, 1.1, 1.1 );
scene.add(mesh);

geometry = new THREE.BoxGeometry(0.75,0.75,1);
//geometry = new THREE.CylinderGeometry(1,0.2,0.2);
geometry.applyMatrix(new THREE.Matrix4().makeTranslation(0,0,-0.5));

point = new THREE.Mesh(geometry);
```

Captures de pantalla

