

Gràfics i Visualització de Dades:

Pràctica 3



Albert Morales

Sergi Martorell

Arnau Vancells

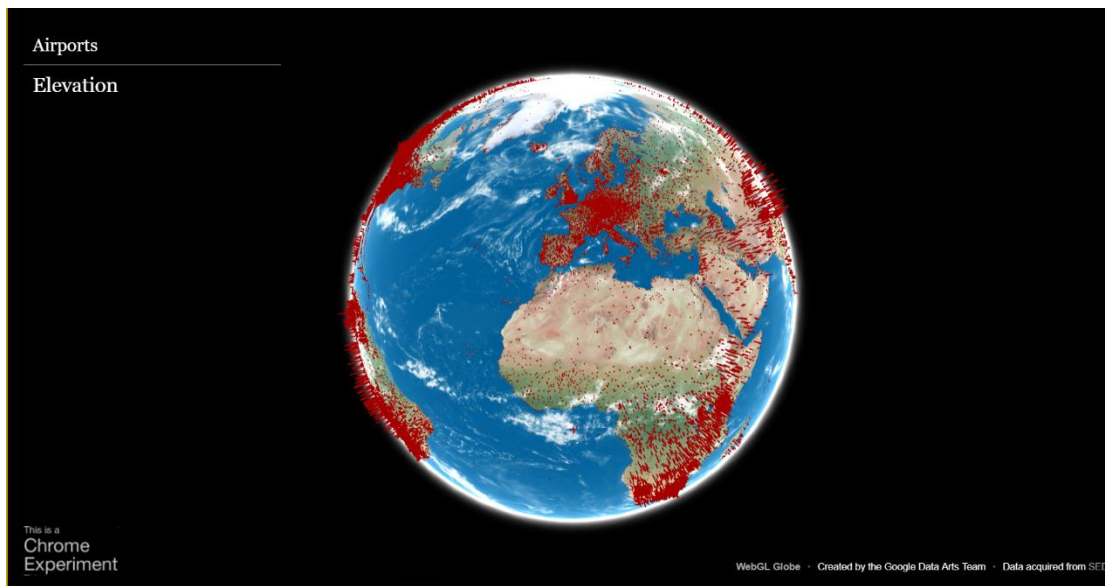
04-06-2018

Introducció

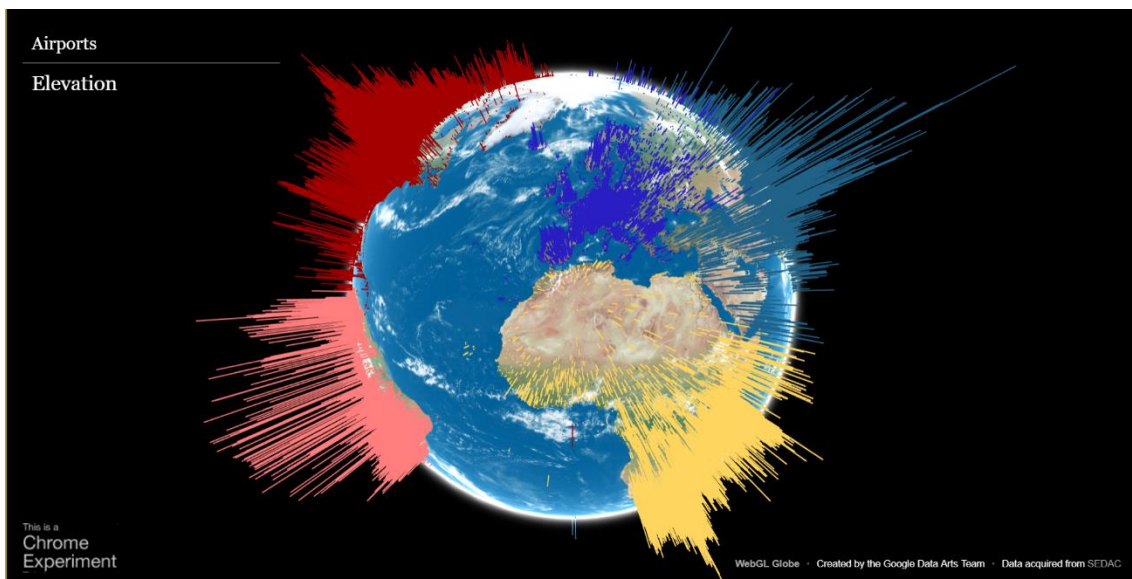
En aquesta pràctica farem servir Three.js per tal de representar un globus terràqui, amb informació en forma de gràfics representat sobre aquest.

Apartats realitzats

4.1- Per a canviar el color de les barres tan sols hem de modificar la funció colorFn.

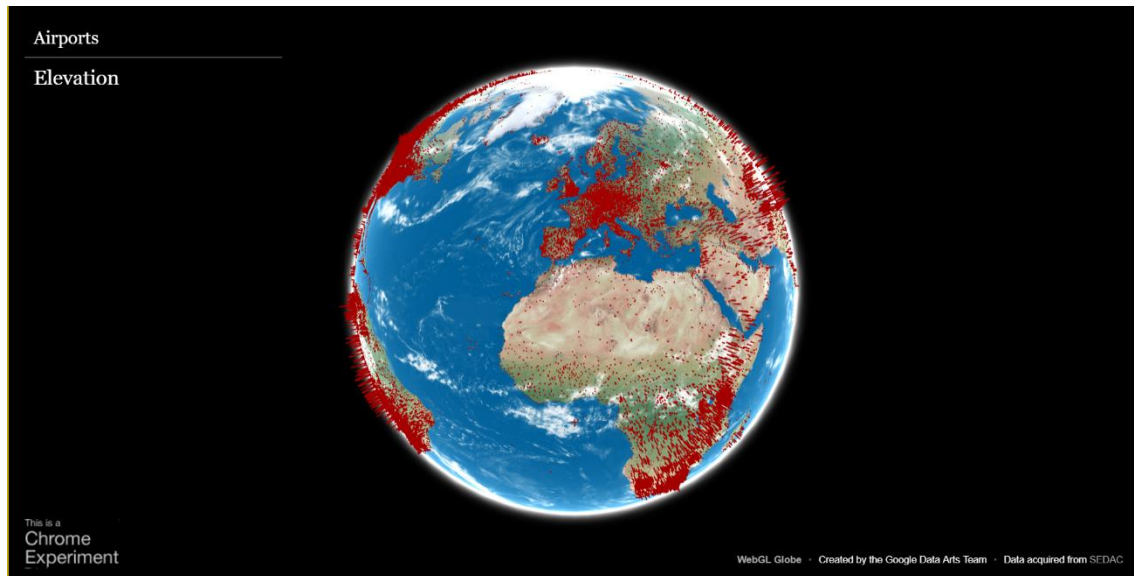


4.2- Per a canviar l'alçada modifiquem size, just abans de crear la barra a addData.



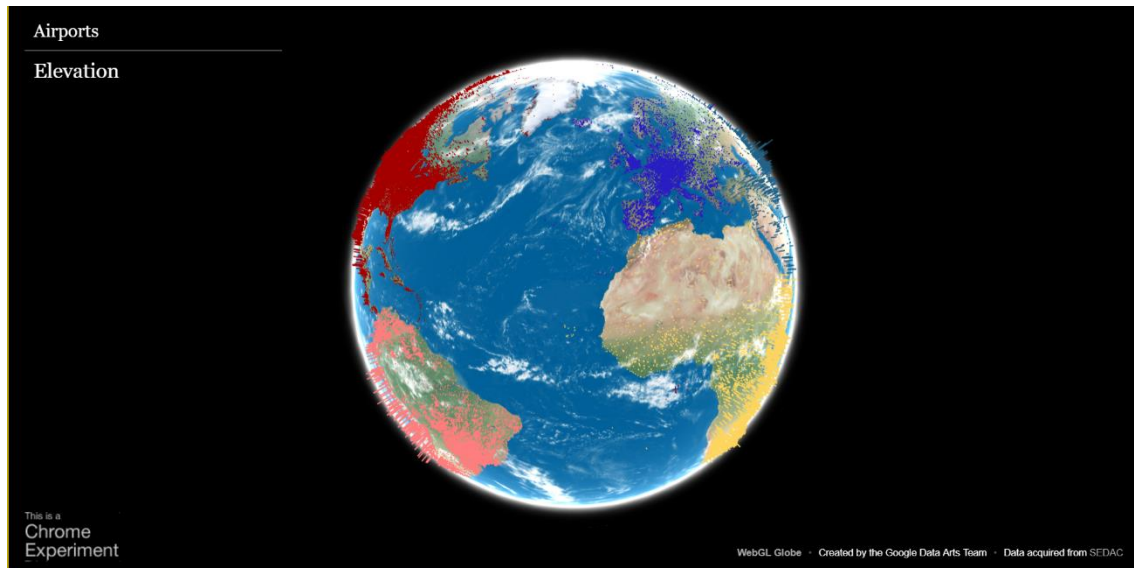
4.3- Per a modificar la textura del globus terraqui, només hem de canviar el path del fitxer a l'hora de fer la crida a loadTexture.

4.4- Hem descarregat d'internet, un fitxer csv que conté informació sobre tots els aeroports del món. Hem creat un script que transforma les dades del csv a .json, de la forma que ens interessa. Així hem guardat al .json la posició de cada aeroport, i la seva altitud.



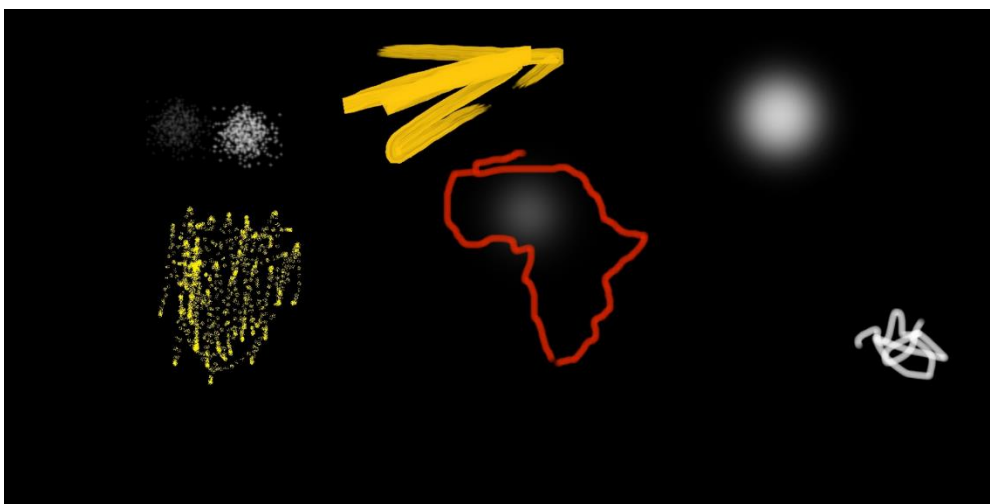
4.5- Hem modificat el Html per a que els títols de les dades visualitzades sigui el corresponent. Com es pot observar a la imatge anterior hem canviat el títol per el corresponent

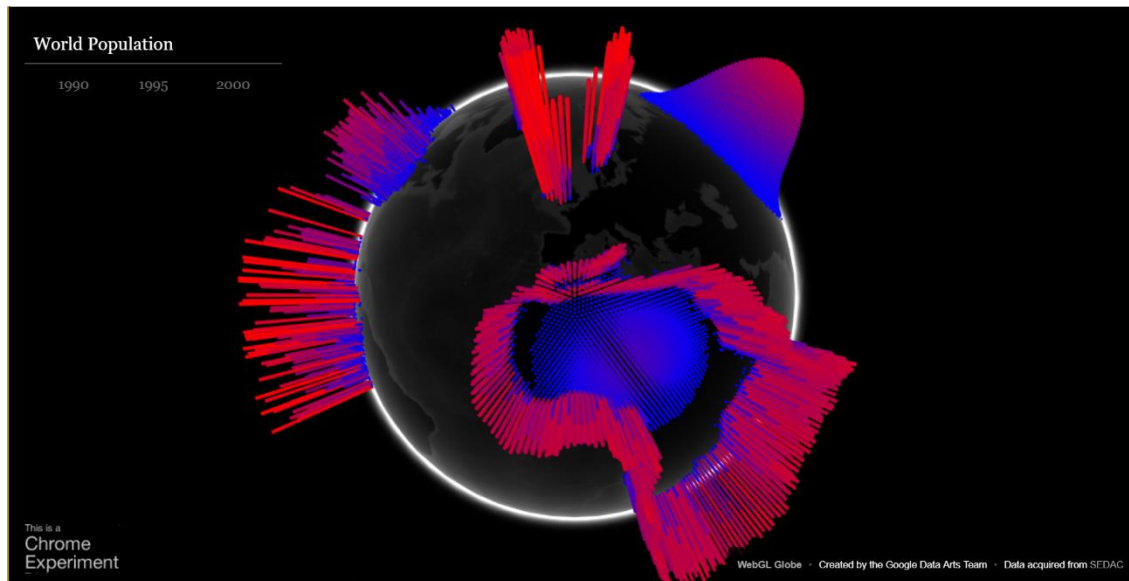
5.1- Per a poder afegir una paleta de colors, hem modificat el nostre fitxer de dades .json, per tal de tenir en compte el continent en el que està situat l'aeroport. A cada continent li hem assignat un número, i a partir d'aquest, pintarem cada dada amb un color diferent, fent un overwrite de la funció colorFn, per a que el color sigui basat en el nombre del continent.



5.2- Les barres de dades es visualitzen amb el shader “data”. El fragment shader utilitza un threshold per a descartar les dades que tenen un valor massa petit, de forma que si no se supera aquest valor, el shader no mostrarà les dades.

Modificant el fitxer de textura, hem pintat a diferents parts d'aquesta, i això s'ha vist reflectit en la representació de les dades del globus.

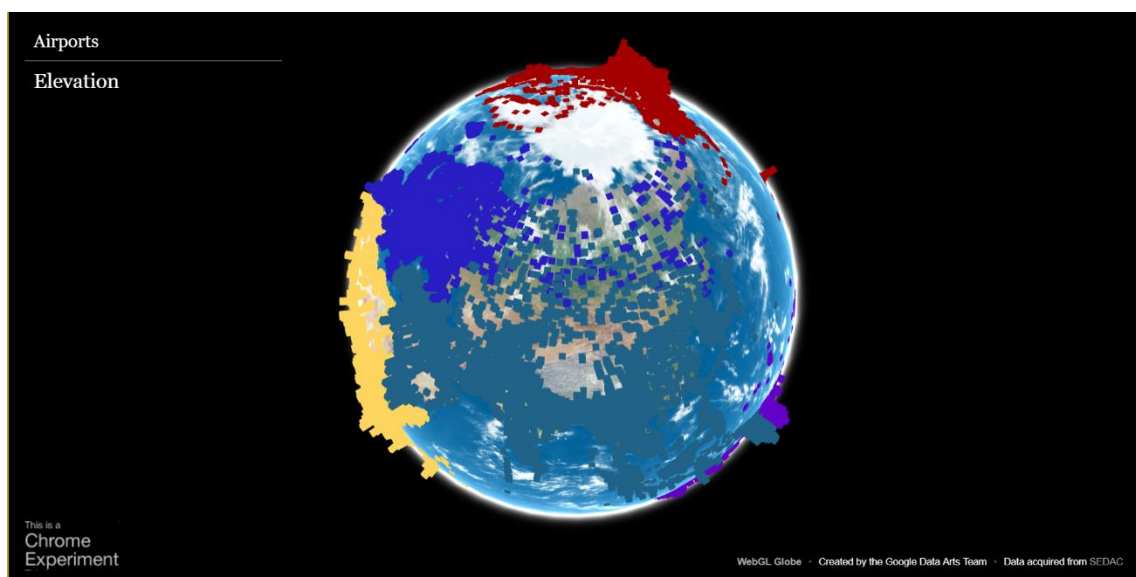




Com podem veure, depenent de la quantitat de color que hi hagi a la textura, la informació serà representada amb més o menys alçada en els gràfics.

Per adaptar això al nostre projecte inicial(amb les dades dels aeroports) hauríem de processar les dades que tenim, per tal de crear un fitxer de textura, que representi, a partir de diferents intensitats de color, la altitud de cada aeroport. Així, i aprofitant el codi de `webgl-globe-texture`, podríem representar les mateixes dades fent servir textures.

5.3- Per a modificar la forma associada a les dades, hem de canviar la geometria d'aquesta, així podríem assignar un mètode diferent de `BoxGeometry` a la geometria de les gràfiques, per tal de representar les dades, per exemple, en forma de `Cone`. Com que `three.min.js` no té la funció `ConeGeometry`, el que hem fet és modificar la forma de `BoxGeometry` per a que més quadrada.



Hem intentat representar-ho amb esferes, fent servir SphereGeometry, però degut a que tenim una quantitat de dades molt gran (de l'ordre de 50000 entrades) el navegador deixa de funcionar al carregar tantes esferes.

Com a afegit, hem modificat tant index.html com globe.js, per tal d'afegir una imatge de fons. Per a fer això hem afegit transparència en la representació del globe.



Animació

La animació que hem fet, té una duració de 4 minuts, però es tracta de la repetició de la rotació del globe.

Per a fer-la, hem modificat la funció onDocumentKeyDown per tal que, al prémer les fletxes esquerra i dreta, es mogui la càmera, per tal de poder "girar" el globe.

A l'animació hi ha representades les elevacions de cada aeroport del món, pintades a partir del color del continent en el que està situat (o és propietat) l'aeroport.

Link al vídeo: youtu.be/xWnbQY9RT7U