# **12/01**:Cesium

<http://cesiumjs.org/demos.html>

# Tiles

Obtener lat y long máximas y mínimas para establecer bien el tile correspondiente. Revisar si *omnivore.gpx* me da los valores máximos y mínimos.

Ajustar imagen mapbox como textura para el terreno con las elevaciones obtenidas de CesiumJS

Imagen de mapbox ponerla sobre el terreno a modo de prueba. Tiene que ser una imagen guardada en el servidor para que sea reconocida como textura.

Guardar una imagen en el pc, Ajax.

<https://ausdemmaschinenraum.wordpress.com/2012/12/06/how-to-save-a-file-from-a-url-with-javascript/>

blob

<http://msdn.microsoft.com/en-us/library/ie/hh779016%28v=vs.85%29.aspx>

<http://weworkweplay.com/play/saving-html5-canvas-as-image/>

**Problemas:** Para añadir una textura a un objeto el fichero debe existir en local(servidor), mapbox no deja importar los static map a ficheros, solución?

<https://www.mapbox.com/help/export-image-pdf/>

**solución**?: <http://stackoverflow.com/questions/6306935/php-copy-image-to-my-server-direct-from-url>

Three.js texture

<http://solutiondesign.com/webgl-and-three-js-texture-mapping/>

Cesium ver los índices con three.js, sino ver el quantized mesh.

<http://threejs.org/examples/obj/male02/Male02_dds.js>

# Crear mesh con vértices + faces.

Problemas:



El problema es que al añadir la geometría a la escena esta no puede leer una posición del array de vértices o faces ya que seguramente la posición exceda el tamaño de uno de los arrays.

**Solucionado**: no se estaban creando bien los puntos de la geometría (x,y,z) ni las caras. Cada vertice debe ser del tipo *Vector3* y cada cara debe ser del tipo *Face* ambos proporcionado por Three.js

# Crear mesh con datos de QuantizedMeshTerrainData

**Solucionado**: pero la salida no corresponde a lo esperado.

**Solucionado**: los datos anteriormente no estaban siendo pasados correctamente.

# **19/01**: Tile obtenido representado en un mesh(malla).

Se ha obtenido un mesh correspondiente al tile en esa coordenada. Según Wikipedia un tile es la parte [gráfica](http://es.wikipedia.org/wiki/Gr%C3%A1fica) de cada [videojuego](http://es.wikipedia.org/wiki/Videojuego) que puede ser utilizada para completar partes de un [fondo](http://es.wikipedia.org/wiki/Fondo) por medio de un [tileset](http://es.wikipedia.org/w/index.php?title=Tileset&action=edit&redlink=1) (conjunto de tiles). Cesium nos devuelve un tileset que es un *quantized-mesh,* un árbol cuaternario piramidal que contiene las alturas (heightmaps) según el TMS. (Tile map service layout and global geographic profile).

**19/01**:La idea es descargar varios tiles y crear sus respectivas mallas para obtener una mayor representación del terreno.

**19/01**: Luego de obtener varios tiles unirlos en un solo mesh con three.js

**19/01**: Añadir textura *Mapbox* al mesh con la ruta *gpx*.

~~¿Tamaño de cada tile proporcionado por cesium?~~

No es necesario en un principio ya que sabiendo el tile que seleccionamos sería suficiente con seleccionar el siguiente.

~~¿Coordenadas que abarca, dirección del tile, escala de los datos?~~

Antes de añadir un mesh a la escena le indico la posición o cualquier atributo que quiera modificar.

# Mapbox <-> Cesium

Relación entre el nivel del tile y el zoom *z* en los mapas de Mapbox.

Coordenadas para el ejemplo: ~~latitud~~-longitud 2.002 y ~~longitud~~ latitud 43.317







Imagen de San Sebastián con nivel de zoom 12 y mesh obtenida con las mismas coordenadas con nivel de detalle 12, se puede ver que no son niveles comparables.

Cesium posee un objecto Rectangle que contiene los valores en radianes de las latitudes y longitudes. Se pedirán las 4 esquinas para saber las coordenadas en esos puntos y así obtener el tamaño de nuestro tile en *real*.

*var positionLonLat = Cesium.Cartographic.fromDegrees(longitude, latitude);*

*positionTileXY = aCesiumTerrainProvider.tilingScheme.positionToTileXY(positionLonLat,12);*

*var rectangleTileXY = aCesiumTerrainProvider.tilingScheme.tileXYToRectangle(positionTileXY.x, positionTileXY.y, 12);*

Con esas instrucciones obtenemos el rectángulo con los datos.

Transformar de radianes a grados, cesium tiene una función pero al intentar ejecutarla igual que todas me da error.

Como es fácil la conversión creare una función propia .



Cada tile es de aproximadamente 4,9km por lado así que habrá que buscar una relación con esos datos.

<http://en.wikipedia.org/wiki/Decimal_degrees>

<http://cesiumjs.org/Cesium/Build/Documentation/Rectangle.html>

# **20/01:** Añadir varios Tiles en la escena

Se deberán añadir varios Tiles y unirlos para crear un terreno que concuerde con las coordenadas, se debe tener en cuenta la ruta que realizo la persona para que la cantidad de tiles abarque la ruta especificada.

**Problemas: ¿**posiciones de los tiles? ¿Cómo ponerlos bien? ¿Orden en la escena?



Tiles añadidos manualmente ahora hay que agregar tres más y crear una estructura con todos, es decir un solo mesh.

# Merge tiles en un mesh

**Problemas**

****

**Solucionado:** Para llamar a la función *merge()* se debe hacer desde una variable del tipo THREE.Geometry y los argumentos deben ser del tipo THREE.Mesh.geometry y THREE.mesh.matrix. Antes de llamar a la función *merge()* se debe actualizar la matriz del objeto THREE.Mesh con la función *updateMatrix()*.

http://stackoverflow.com/questions/24353756/migrating-geometryutils-merge-to-geometry-merge

Posible solución a error: <http://stackoverflow.com/questions/24422289/three-js-merging-geometries-and-mesh>

# Añadir textura

**Problemas:**  Al añadir texturas salta el siguiente error:

[.WebGLRenderingContext]GL ERROR :GL\_INVALID\_OPERATION : glDrawElements: attempt to access out of range vertices in attribute 2

Possible solución: <http://stackoverflow.com/questions/21200386/webgl-gl-error-gl-invalid-operation-gldrawelements-attempt-to-access-out-of>

# Uniendo el pfc

Expresiones regulares: <http://webintenta.com/validacion-con-expresiones-regulares-y-javascript.html>

<http://www.w3schools.com/html/html5_webstorage.asp>

Variables de sesión, luego habrá que borrarlas.