Módulo Tangram

1. Definición

Proyecto para visualizar mapas, basado en Teselas vectoriales Este proyecto, permite implementar diseños en tiempo real y proporcionan interactividad al usuario final.

1. Componentes

* WebGL🡪 API para procesar mapas tipo 2D y 3D.
* GeoJSON🡪 Lectura de información en formato Geographic JSON
* TopoJSON🡪 Lectura de información en formato Topographic JSON
* Binary Vector Data🡪
* Leaflet🡪 Funcionalidad básica para zoom y desplazamiento sobre la región en el mapa.
* YAML 🡪Estructura de componentes que se cargan en el mapa de la vista. En el API se tiene el archivo principal denominado scene.yaml
* Main Library 🡪 tangram.debug.js

Coordenadas URL

Se procesan por la tupla en la definición del URL en la barra de direcciones del Navegador: http://<localhost>/index.html#mapzen, <latitud>,<longitud>,<zoom>.

Donde

Mapzen: Es el servicio de carga de los mapas implementados con Tangram.

Propiedades

* Lights

Establece como se iluminan los objetos que se cargan en el mapa. Se establecen parámetros como dirección, tipo de iluminación, modalidad de difusión los cuales permiten aplicar aspectos de iluminación sobre los mismos.

* Cameras

Establece como se visualizan los objetos del mapa. Es decir, establece la forma de los objetos que se cargan en el mapa. Existen 3 tipos de vistas: perspectiva (perspective), isométrica (Isometric), y plana (flat). Cada uno de estos, permite personalizar distintos aspectos como tipo (type), ejes (axis), longitud focal (focal length) y punto base (vanishing point), así como si se encuentra activa o no la vista.

* Scene

Determina el fondo del escenario donde se carga el mapa. Para el demo, se define el color del fondo.

* Textures

Define la visualización de sitios como hospitales, árboles, hoteles basado en una serie de iconos que se encuentran encapsulados (sprites) en el archivo images/poi\_icons\_32.png. Determina la manera de filtrar los elementos en el mapa. En ésta directiva, permite definir la animación de los iconos (sprites) con base a un arreglo (array) que representa los siguientes elementos: [Origen coordenada x, Origen coordenada y, longitud, alto].

* Styles

Representa el estilo de visualización de elementos del mapa. Están los siguientes: water animation, rainbow, pop-up, elevator, environment map, halftone.

En la tabla 1 se describen cada uno de los estilos de visualización, junto con la imagen que lo representa.

**Tabla 1. Estilos de visualización del experimento 1**

|  |  |  |
| --- | --- | --- |
| **ESTILO** | **DESCRIPCIÓN** | **IMAGEN RELACIONADA** |
| Water | Estilo donde los bloques se mueven como si se observara las corrientes de un rio. |  |
| Elevator | Estilo donde se visualizan los bloques de edificios en movimiento |  |
| PopUp | Vista de perspectiva de los bloques de edificios al desplazarse a lo largo del mapa |  |
| HalfTone | Vista plana de puntos en color negro |  |
| Windows | Vista de forma fotográfica formato negativo del mapa |  |
| Environment | Vista del mapa en tonos solidos de grises y negros |  |
| Rainbow | Vista dinámica de los elementos del mapa con colores que cambian de manera sucesiva |  |

* Layers

Permite activar o desactivar las capas que se encuentran en el mapa. Los elementos de las capas se describen en tabla 2.

**Tabla 2. Elementos de las capas del mapa**

|  |  |
| --- | --- |
| **CAPA** | **DESCRIPCIÓN** |
| Earth | Capa de tierra sobre el mapa |
| Landuse | Parques y espacios al público |
| Water | Cuerpos de agua |
| Roads | Calles principales, secundarias |
| Buildings | Construcciones |
| Road label | Títulos correspondientes a las calles principales, secundarias |
| Poi\_icons | Iconos que describen hospitales, parques, cadenas de hipermercado |
| Places | OJO |
| Point\_labels | Títulos correspondientes a la opción Poi\_icons |

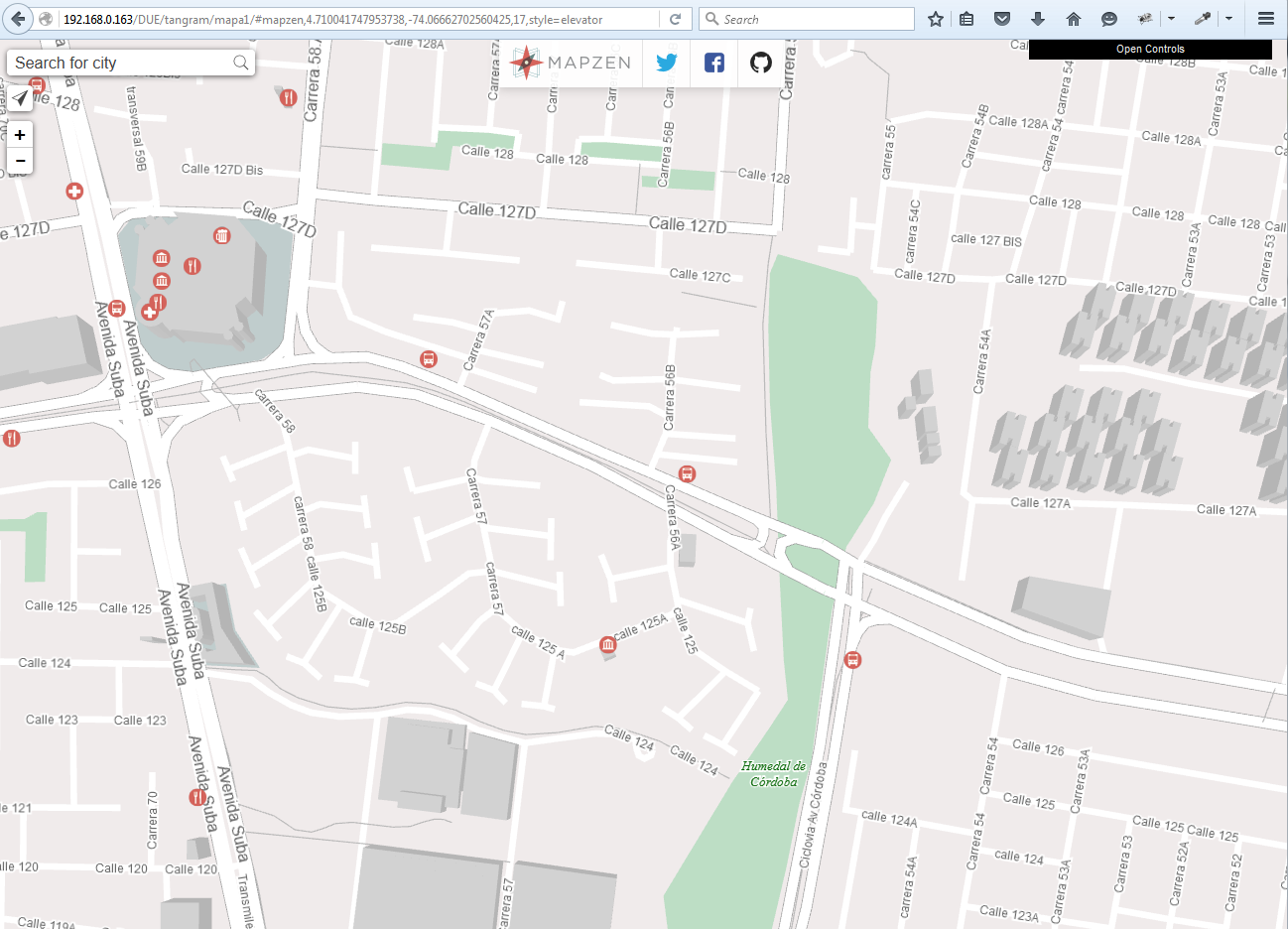
Uso de la propiedad iluminación

Bajo el archivo scene.yaml, se realiza un experimento de iluminación. En la figura 1, se visualiza la iluminación definida según tabla 3.

**Tabla 3. Parámetros de iluminación**

| **PARAMETRO** | **VALOR** |
| --- | --- |
| Type | Direccional |
| Direction | [.1, .5, -1] |
| Diffuse | .2 |
| Ambient | 1.1 |

**Figura 1. Iluminación predeterminada**



FUENTE: DANE. Aplicación prueba

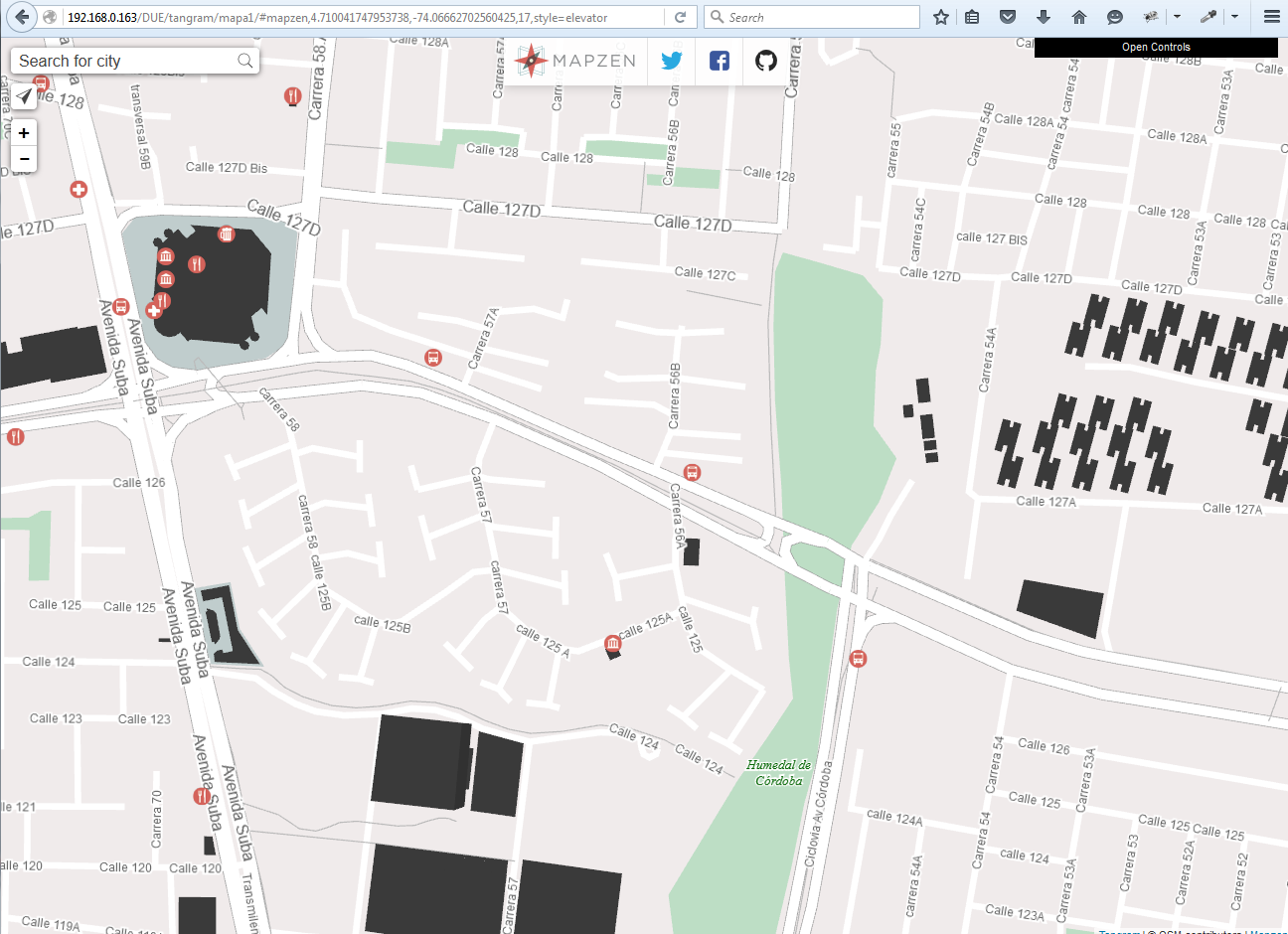
Para el ejemplo, se actualizó la iluminación según se muestra en tabla 3a.

**Tabla 3a. Nuevos valores de iluminación**

|  |  |
| --- | --- |
| **PARAMETRO** | **VALOR** |
| visible | true |
| type | point |
| position | [-74.0170, 40.7031, 100] |
| origin | world |
| ambient | 0.3 |
| diffuse | 1 |
| specular | .2 |

Al aplicar éstos valores de la configuración, se obtiene el mapa visualizado en figura ¿?.

**Figura ¿?. Iluminación del nuevo mapa**



FUENTE: DANE. Aplicación prueba

Como se visualiza en la figura, la iluminación se aplica sobre los bloques sólidos del mapa.

Navegadores compatibles

* Opera: Todas las versiones.
* Chrome: Todas las versiones.
* Firefox: No funciona en versiones 29 e inferiores.

Webgrafía

1. Mapzen Tangram. Disponible: <https://mapzen.com/projects/vector-tiles/>
2. GitHub. Real-Time WebGL Maps. Disponible: <https://github.com/tangrams/tangram>
3. Mapzen. Tangram Render 2D and 3D maps with fine control over almost every aspect of the map-making process. Disponible: [https://mapzen.com/documentation/tangram/](https://mapzen.com/documentation/tangram/%20)