Documentación Práctica 4

Miembros del grupo:

Alejandro Serrano López, bq0100

Diego Torres Aranha, bq0383

Gabriel Gil García, bq0162

Juan José Urioste, bt0492

Miguel Laredo Barbadillo, br0449

Iván Lumbano Vivar, br0097

Qué hemos hecho:

Hemos modificado el programa de la práctica 3 para que utilize el mapa de BingMaps. Para ello uno de nosotros se ha registrado en BingMaps con la cuenta de usuario de la UPM y obtuvo un API-token para tener acceso a la API REST.

Luego, el observador traduce la longitud y latitud a grados, se la pase al Canvas y luego dicho Canvas modificado toma una url de BingMaps y la carga. También se ha modificado el Canvas para que la flecha siempre esté en el centro de la imagen, como en los sistemas de muestra GPS de verdad, y porque el BingMaps admite poner una dirección como centro de la imagen.

La estructura de la url es esta (nótese hemos reemplazado la key por ?????):

```
https://dev.virtualearth.net/REST/v1/Imagery/Map/Aerial/"+this.getLatitudGrados()+","+this.getLongitudGrados()+"?mapSize=950,950&zoomlevel=18&key=?????
```

Posteriormente probamos esto y veíamos que funcionaba, pero tenía muchas ineficiencias, entre ellas que había un flash constante y muchas peticiones. Así que, para prevenir LAG y que la API nos bloquee, inicialmente decidimos tomar una imagen del cuádruple de tamaño y un búffer de imágenes en forma de matriz cuadrada de 3x3, en la que la imagen central se renderiza por encima de las demás, y se mueven todas arriba, abajo, a la izquierda o la derecha si la variación de posición total dividada por la relación metro * píxel (que según BingMaps, es `(156543.04 * Math.cos(latitudGrados) / (Math.pow(2, this.zoomLevel) ))`) superaba la mitad del ancho de pantalla, en tal momento la diferencia se reseteaba. Si la imagen central de dicha posición falta, se pide y así se previene cargar demasiado. Luego las imágenes no nulas se renderizaban de forma que las más cercanas ortogonalmente a la imagen central estuvieran por encima de las más lejanas.

Esto requirió una modificación extensiva del Canvas, así como una ligera modificación del Observer para prevenir que se entregase un dato corrupto.

Para los tests, procedimos a realizar el máximo aumento posible (descubrimos que aunque el las diapositivas dice que 19 es el máximo aumento, 20 también es posible) para poder detectar errores. Alejandro realizó un test inicial en el que un piloto condujo por Rivas-Vaciamadrid, pero hubo un problema con las imágenes, en la que el Toolkit recibía la url correcta pero todas las imágenes eran la misma, y en la que no importaba como, siempre se quedaba atascado con la

primera imagen que cargaba. Luego tras realizar la operación para que la matriz se reemplazara correctamente ese problema desapareció.

Posteriormente procedimos a realizar una optimización para evitar que la pantalla parpadeara, en la que usamos una clase especial de uno de los tutoriales de java llamada DrawingPanel. Esto requirió aún más cambios en el "MyCanvas", así como ligeras modificaciones en el Observer.

Durante este período de mejora nos dimos cuenta de que el método de matriz que habíamos realizado no era suficiente puesto que tendría que almacenar también la posición de desplazamiento. Por lo tanto consideramos dos opciones: una en la que que se guardara la información del desplazamiento de la matriz, y otro en el que simplemente se cargara una imagen cuando fuese necesario (es decir, cuando se acercase a los bordes de la imagen en la que estaba), sin un búffer. Por simplicidad decidimos la segunda, y gracias a ello descubrimos que BingMaps recorta las imágenes a partir de cierto tamaño, haciendo que el centro esté en otro sitio, por lo que decidimos utilizar las herramientas de la clase g2d para escalar a 2:1 una imagen que Bing nos había dado con la mitad de zoom. Además, vimos que la fórmula seguía siendo algo aproximada, por lo que los píxeles por metro seguían debiendo de ajustarse en cierta medida.

También por todos los contratiempos que nos daba el cable roto, Alejandro decidió comprar uno nuevo, que costó 36.30 € + 19 del adaptador serial-USB, de la tienda la casa del GPS de Rivas. Ya quedan pocos de esos cables, así que, por favor, trátenlos con cuidado.

Finalmente, para embellecerlo, se decidió modificar el Canvas y Observer aún más para que se diese por pantalla las coordenadas UTM, la latitud y longitud en grados, diera un pitido del sistema si la velocidad es roja, y se indicara la velocidad del mapa de pista más cercana.

Links:

Tutorial de java: https://www.youtube.com/watch?v=HbN78cFcGIo