

Démarches et recherches projet Leaflet

BLARD Flavie – FOURNEYRON Axel – THOMAS Bénédicte

Fonctionnalités de l'application

Grâce à notre application développée en JavaScript, HTML et CSS, l'utilisateur pourra observer différentes informations relatives à la Métropole de Lyon :

- Le taux de chômage par IRIS
- La localisation des lignes de métro et leurs directions respectives
- La localisation des stations d'auto – partage

Par ailleurs, une fonction d'édition et de création de polyligne est présente.

Processus utilisateur :




- Sélection des couches à visualiser
- Sélection du fond de carte souhaité
- Possibilité de créer une polyligne et/ou de la modifier

Réponse interface :

- Ajustement visualisation en fonction du choix des couches à afficher
- Si affichées, les lignes de métro sont représentées par des polygones et logos animés. L'animation s'arrête dès lors que l'utilisateur passe la souris sur l'icône.

Méthodologie construction application

Pour établir notre application nous avons mobilisé différents langages de programmations :

Langage Programmation	Fonction
HTML 	Définit la position des éléments de la page (Titre, image, corps du texte)
JavaScript 	Permet d'afficher des interactions entre la page et l'utilisateur
CSS 	Définit le style graphique de la page

Par ailleurs pour programmer notre application cartographique, nous mobilisons la librairie **Leaflet**. Celle-ci codée en JavaScript nous permet de programmer l'ensemble des géo-traitements nécessaires aux analyses spatiales d'informations géographiques. En outre, cette librairie open source nous permet également d'obtenir de l'interactivité entre la carte et l'utilisateur.

Détaillons par étape notre démarche :

A. Récupération – création des scripts

- 1) Récupération d'un template CSS gratuit (<https://html5up.net/>)
- 2) Récupération de la librairie Leaflet avec fichiers associés (CSS, JS, images etc)
- 3) Nettoyage des fichiers css et html du template pour conserver les éléments de mise en page souhaités
- 4) Création d'un fichier Javascript

B. Création des données en GeoJSON

Pour exploiter les données avec Leaflet, nous convertissons avec un logiciel SIG (ici Qgis), les données en format GeoJSON avec un système de coordonnées de type WGS 84. Une fois les données converties, nous les sauvegardons dans un fichier Javascript appelées « data_geojson.js ».

C. Lier les scripts à la page HTML

- 1) Dans la page HTML, nous appelons les scripts css :
 - Du template
 - De la librairie Leaflet
- 2) Puis, nous appelons, les scripts JavaScript :
 - Comportant les données en GeoJSON
 - Le fichier .js créé lors de l'étape un qui nous servira à coder l'application avec la librairie Leaflet.

D. Programmation des interactions

Dans un premier temps, nous avons affiché le fond de carte, ajouté les données et personnalisé l'interface pour avoir la possibilité de choisir par des cases à cocher les couches à visualiser. Une fois cette étape terminée, nous nous sommes concentrés sur les différentes options à ajouter :

- *Afficher les données – Menu choix des couches à afficher*
Nous avons ici débuté par construire le js avec une variable stockant l'identifiant de la carte. Puis, nous avons appelé un fond de carte OSM avant de préciser le canevas de la carte. Lors de la déclaration de l'affichage des données, pour les iris, nous précisons qu'il faut mobiliser le script du plug-in carte choroplete. Nous indiquons la variable à représenter « tchom » avec un nombre de classe de 8.
- *Ajouter un marqueur animé le long des lignes de métro*
Pour cette option, nous avons utilisé un plug-in : (NOM PLUG-IN + LIEN). De la même manière que pour les cartes choroplete, le code a été récupéré sur Github et lié au code html. Une fois implémenté, une icône, format png, a été récupérée. Puis, elle a été animée pour se déplacer selon le tracé des lignes de métro.
- *Ajouter des zones tampons de 300m au survol des stations auto-partage + mise en valeur de la station (par un changement de taille, de couleur ou de représentation)*
- *Proposer une fonction de dessin de ligne*

Ici, le code d'un plug-in a été mobilisé. Seulement, cette fois-ci nous avons directement appelé dans le html les url sur lesquels les codes Javascript et css du plug-in étaient disponibles. Il s'agit d'un plug-in de création et d'édition d'entités de différents types : rectangle, marqueurs, cercle, lignes, points... Celui-ci a donc été nettoyé pour ne conserver que la création et l'édition de polygones. Ainsi, l'utilisateur peut créer et modifier une polygone sur la carte.

Mise en page du site générale

Consigne et objectif du projet

A partir des données fournies (en Shapefile) :

Créer une webcarto représentant, les lignes de métro, les stations auto-partage et le taux de chômage (attribut « tchom ») par IRIS.

La webcarto devra comporter une gestion des calques et les couches devront être stylisées.

Votre page (outre la webcarto) comportera DU STYLE.

l'objectif finale étant de mettre un peu de fun dans l'game, ainsi que apprendre en s'amusant avec un trinôme de travaille qui rayon de mille feu pour celà nous souhaitons :



Code l'application cartographique



Pouvoir télécharger un fichier



Regarder les bibliothèque de template



Boire une petite bière pendant le projet



Créer des actions dynamiques



Travailler avec une équipe d'enfert !



Projet WebCarto

Réalisé.e.s par :

- Bénédicte Thomas
- Flavie Blard
- Axel Fourneyron

Les objectifs du projet

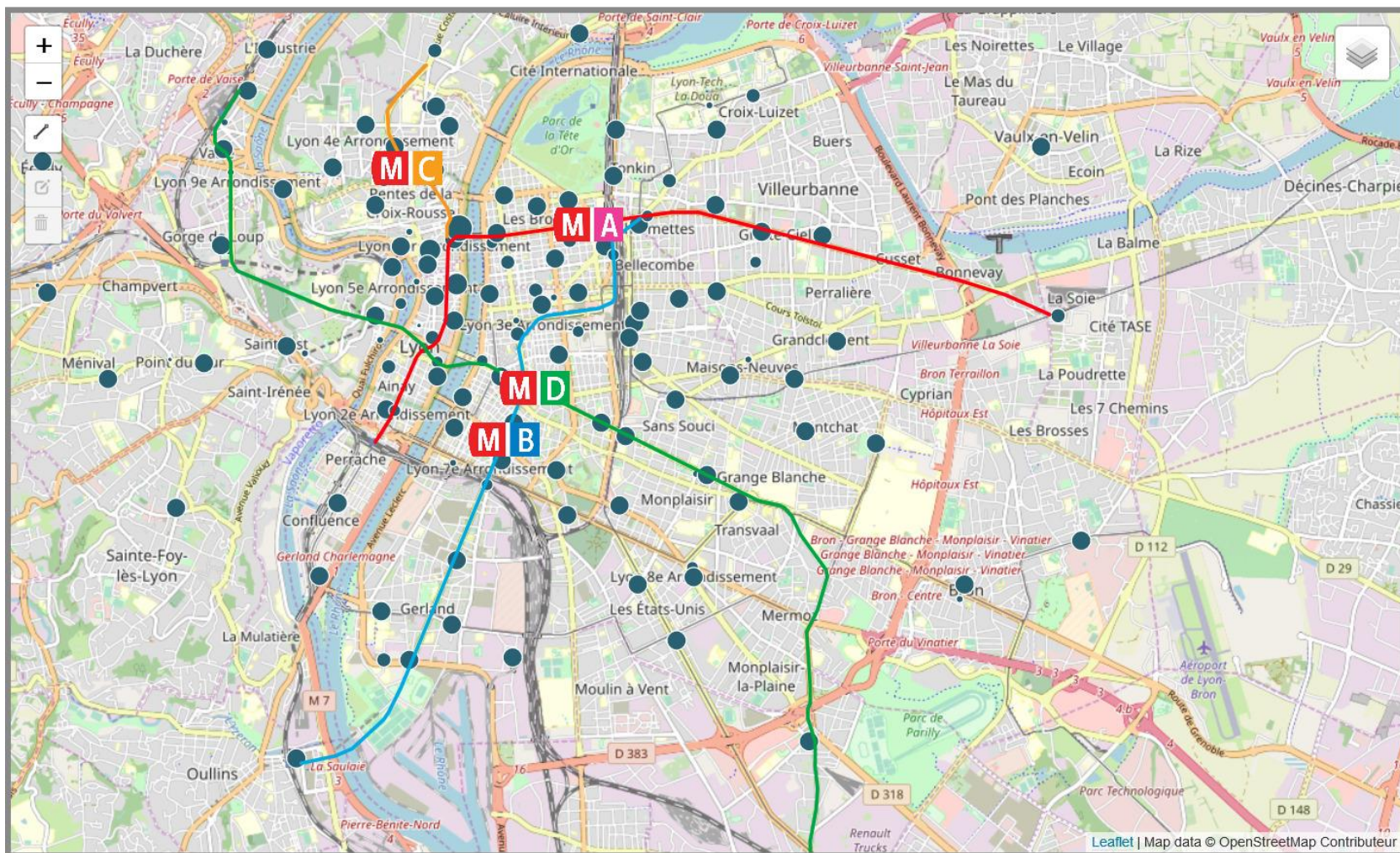
La cartographie

Explication du code



Document récapitulatif

Cartographie des transports de métro et des points d'autopartages



Projet WebCarto

Réalisé.e.s par :

- Bénédicte Thomas
- Flavie Blard
- Axel Fourneyron

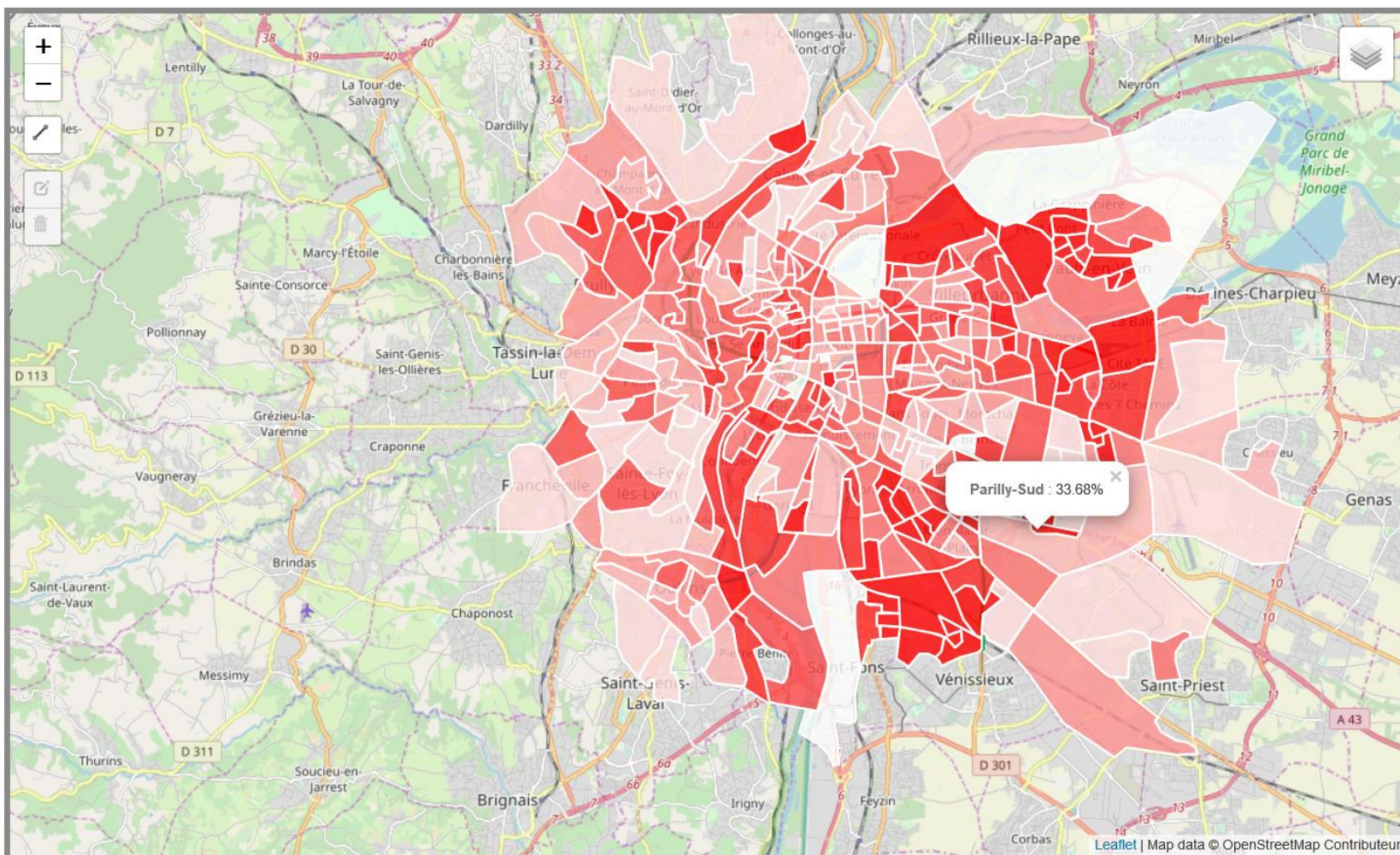
Les objectifs du projet

La cartographie

Explication du code

 Document récapitulatif

Cartographies des Iris de Lyon



Centre-Nord : 25.33%



Projet WebCarto

Réalisé.e.s par :

- Bénédicte Thomas
- Flavie Blard
- Axel Fourneyron

Les objectifs du projet

La cartographie

Explication du code

 Document récapitulatif

Explication dans les grandes lignes du code liée à la cartographie

Option 1

Ajouter un marqueur animé le long des lignes de métro

code.js

```
// Fonction marqueur animé
// Récupération des coordonnées des lignes de métros
function addMarker(feature, layer) {
  var coordinates = [];
  for (var i=0; i < feature.geometry.coordinates[0].length; i++) {
    coordinates.push(feature.geometry.coordinates[0][i].reverse())
  }
  var polyligne = L.polyline(coordinates);
  // Ajout d'une icône personnalisée pour chaque ligne (concaténation selon le nom de la ligne)
  var animatedMarker = L.Marker.movingMarker(polyligne.getLatLngs(), 10000, {
    autostart: true,
    loop: true,
    icon: L.icon({
      iconUrl: layer.feature.properties.ligne+'.png',
      iconSize : [58,28]
    })
  }).addTo(animation);
}
animation.addTo(map);
```



Projet WebCarto

Réalisé.e.s par :

- Bénédicte Thomas
- Flavie Blard
- Axel Fourneyron

Les objectifs du projet

La cartographie

Explication du code

 Document récapitulatif