

# Compte-rendu des démarches et recherche pour l'évaluation webcarto



*Auteurs : Goullencourt Paul  
Membrede Rémi*

# I. Préparation du projet

## 1. Choix des données

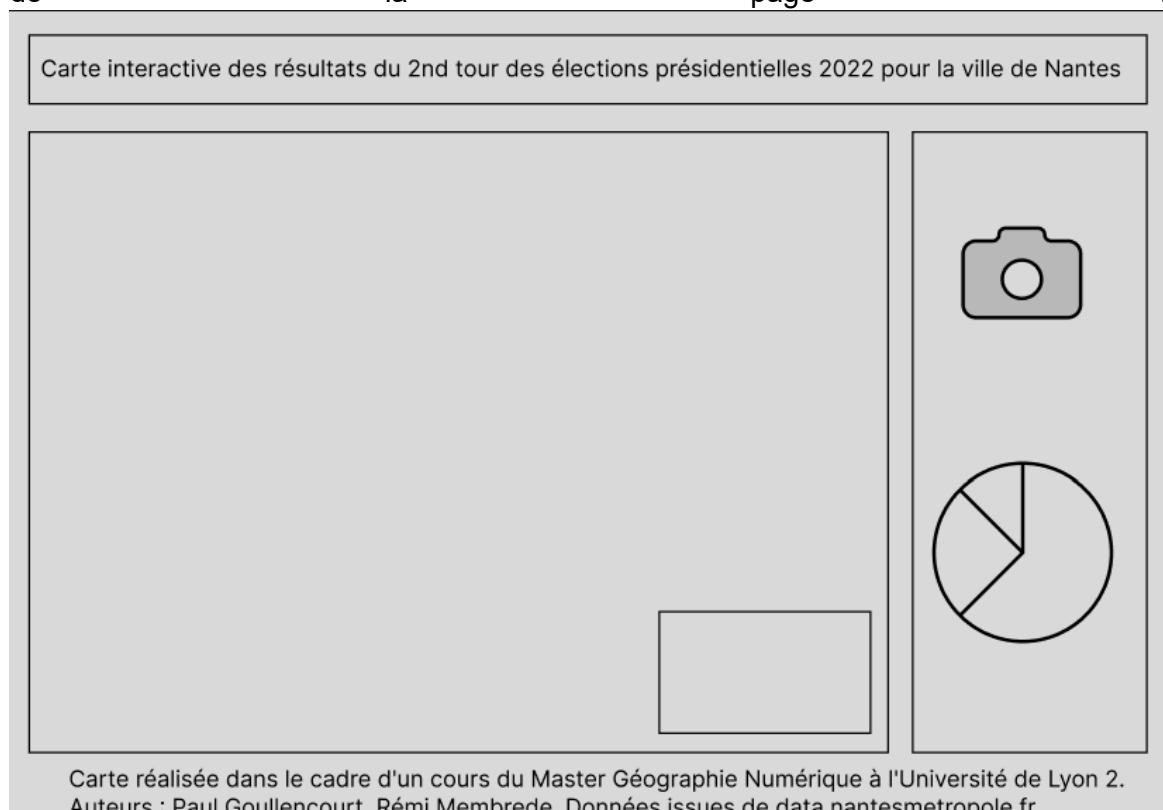
Dans un premier temps, nous avons défini, les données que nous allions utiliser ainsi que la zone géographique d'étude. Après discussion, nous avons décidé de travailler sur les **résultats électoraux du second tour de l'élection présidentielle de 2022**. Ce choix nous a permis :

- De limiter le nombre de candidats à deux, simplifiant ainsi l'analyse et la visualisation des données.
- De changer de cadre géographique en optant pour la **ville de Nantes**, offrant un contexte différent de celui de Lyon, souvent utilisé dans d'autres projets.

Les données ont ensuite été téléchargées sur le site de nantes.data. Voici les données brutes que nous avons récupérées :

- Découpage géographique des bureaux de vote de la ville de Nantes pour 2022
- Election présidentielle 2022 à Nantes - Second tour (infographie)
- Circuits des transports en commun Naolib de Nantes Métropole

Ensuite, nous avons défini les objectifs du projet en nous basant sur les consignes. Nous avons ensuite réalisé une maquette avec FIGMA pour visualiser la structure et l'organisation de la page web.



Il a été décidé de représenter le gagnant pour chaque zone géographique par bureau de vote. Lorsqu'un utilisateur clique sur un polygone, la photo du gagnant apparaît, accompagnée d'un graphique qui détaille les pourcentages précis obtenus par chaque candidat. Le vote blanc a également été intégré au graphique pour offrir une vision complète des résultats.

## 2. Traitement des données

Une fois la maquette du projet finalisée, nous avons pu travailler sur les données. Cependant, il nous manquait les données ponctuelles, qui n'étaient pas encore prêtes à être utilisées.

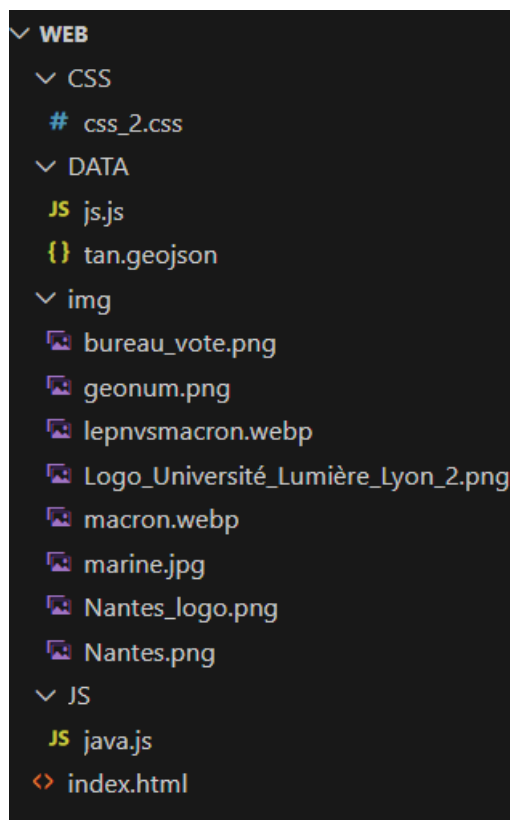
Dans un premier temps, les données fournies par Nantes Métropole ont été analysées et préparées dans QGIS. Cela a impliqué l'analyse de leur structure, la correction des incohérences éventuelles et leur conversion dans des formats compatibles avec le projet. Les centroïdes des polygones ont ensuite été extraits pour créer une couche de points représentant les bureaux de vote. Les données des lignes de tram ont été nettoyées et converties au format GeoJSON pour permettre leur intégration dans le projet. Enfin, des attributs supplémentaires ont été ajoutés aux couches afin de faciliter leur exploitation lors des interactions avec la carte

Nous avons donc nos 3 jeux de données

- ponctuel (points), → Bureau de vote
- linéaire (lignes), → Ligne de tram
- 1 surfacique (polygones) → découpage des bureau de vote

## 3. Organisation des fichiers

Tous les fichiers nécessaires (GeoJSON, images) ont été regroupés dans une structure de dossier claire comprenant des sous-dossiers pour les données, les scripts, et les styles CSS. Cette organisation a permis un accès rapide aux éléments lors des phases de développement et de tests.



- data/ : Contient les fichiers GeoJSON
- img/ : Contient les images des candidats et les pictogrammes
- js/ : Contient le fichier js
- css/ : Contient les styles pour la mise en page

## II. Développement du projet

### 1. Chargement des données

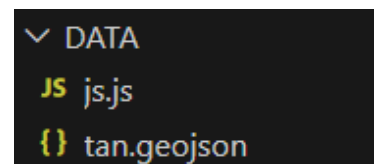
Le projet a pu être lancé en commençant par l'élément central : la carte.

La première étape a consisté à charger les différentes données nécessaires. Pour les lignes de tram, la fonction fetch a été utilisée pour récupérer les données de manière dynamique depuis un fichier GeoJSON externe. Cette approche a été choisie car les données des tramways sont susceptibles d'évoluer régulièrement. En les chargeant dynamiquement, il est facile de mettre à jour le fichier source sans devoir modifier le code de la carte.

En revanche, les données des bureaux de vote et de leur découpage, ont été intégrées directement dans le fichier JavaScript au format GeoJSON. Cela simplifie la gestion des données statiques

Il y'a donc 2 variables dans le fichier JS.JS :

- *Vote*
- *Bureau\_vote*

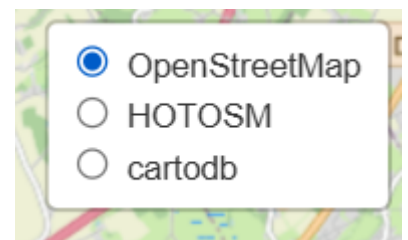


### 2. Gestion des couches et des fonds de carte

Une gestion des couches dynamiques a été mise en place avec **L.control.layers()**, permettant à l'utilisateur d'afficher ou de masquer les différents couches selon ses besoins. Cela simplifie l'exploration des données en évitant une surcharge visuelle et en rendant la carte plus claire et personnalisable.

Les points représentant les bureaux de vote ont été clusterisés avec **leaflet.markercluster** pour améliorer la lisibilité, notamment à faible niveau de zoom. Cette fonctionnalité regroupe les points proches pour éviter les chevauchements et déploie les clusters à mesure que l'utilisateur zoome, offrant une navigation fluide et intuitive

Une gestion des fonds de carte a également été ajoutée, offrant à l'utilisateur le choix entre trois fonds différents provenant d'OpenStreetMap



### 3. Ajout des fonctionnalités avancées

#### Donnée ponctuelle

Pour améliorer la lisibilité et la personnalisation de la carte, les points représentant les bureaux de vote ont été remplacés par un pictogramme spécifique. Ce changement a été implémenté en définissant une icône personnalisée dans le code. Voici comment cela a été réalisé :

1. Une image représentant le pictogramme d'un bureau de vote a été téléchargée et placée dans le dossier img du projet.
2. Lors de l'ajout des bureaux de vote à la carte, une icône personnalisée a été définie à l'aide de **L.icon**, où les propriétés telles que la taille de l'icône (**iconSize**), son point

d'ancrage (**iconAnchor**) et le positionnement des popups (**popupAnchor**) ont été spécifiées.

3. Le pictogramme a ensuite été appliqué à chaque bureau de vote via l'option **pointToLayer** de **L.geoJson**.

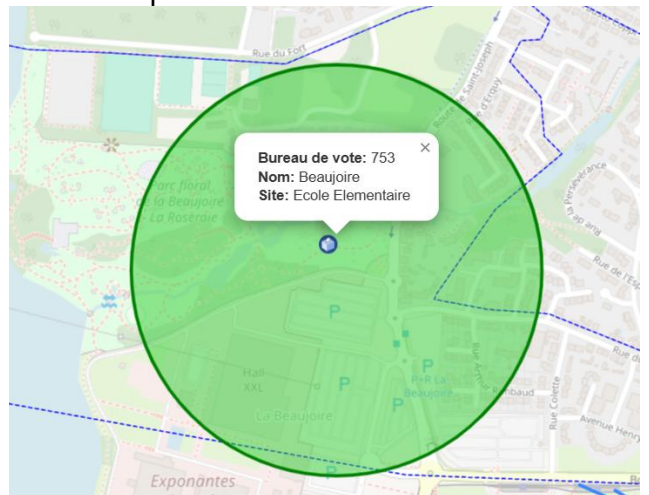
- **Tampons dynamiques :**

Lors du survol d'un bureau de vote, un tampon de 300 mètres s'affiche autour de celui-ci grâce à **L.circle**. Ce cercle est ajouté à la carte au survol et retiré lorsque la souris quitte le point, permettant de visualiser rapidement la zone d'influence de chaque bureau.

- **Popups des bureaux de vote :**

Chaque bureau de vote est associé à un popup interactif qui affiche des informations spécifiques telles que le numéro du bureau, son nom et le site correspondant. Ces popups permettent à l'utilisateur de consulter rapidement les détails des bureaux de vote en cliquant sur un point ou un pictogramme.

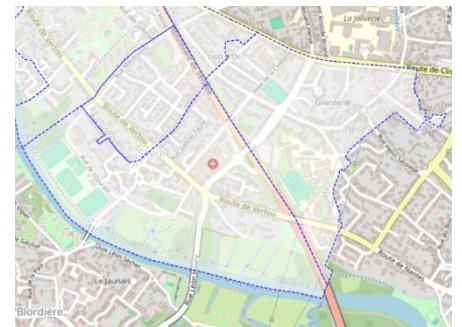
Cette fonctionnalité a été implémentée via l'option **bindPopup** de Leaflet



## Données surfacique

### **Style des polygones :**

Les données surfaciques, représentant le découpage des bureaux de vote, ont été stylisées pour une meilleure lisibilité. Un style par défaut a été défini avec **L.geoJson**, en attribuant des propriétés comme une couleur de bordure (color), une épaisseur (weight), et une transparence (fillOpacity). Ce style met en évidence les limites des bureaux de vote tout en maintenant une présentation claire et homogène.



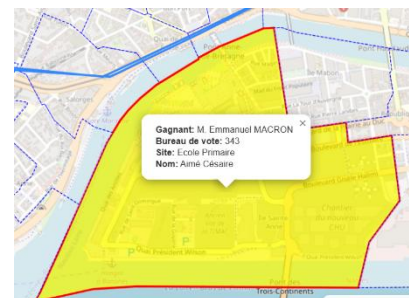
### **Popups sur les polygones :**

Chaque polygone est associé à un popup interactif affichant des informations spécifiques, telles que le nom du bureau, son site et le gagnant des élections. Les popups permettent d'explorer facilement les données associées à chaque zone géographique en cliquant sur un polygone. Ils ont été implémentés via l'option **bindPopup**, en générant dynamiquement le contenu à partir des propriétés des polygones.



### Mise en évidence des polygones sélectionnés :

Pour améliorer l'interactivité, un style de surbrillance est appliqué lorsqu'un polygone est cliqué. La bordure et la couleur de remplissage changent pour mettre en évidence la zone sélectionnée. Cette fonctionnalité, implémentée avec `onEachFeature`, permet de distinguer clairement le polygone actif tout en offrant un retour visuel à l'utilisateur.



### Ajustement de la vue :

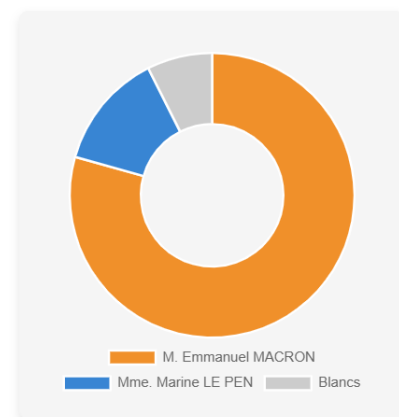
Une fois les données ajoutées, la carte est automatiquement centrée et zoomée pour inclure l'ensemble des polygones. Cette fonctionnalité, réalisée avec `map.fitBounds`, garantit que toutes les zones géographiques sont visibles dès l'ouverture de la carte.

### Données linéaires :

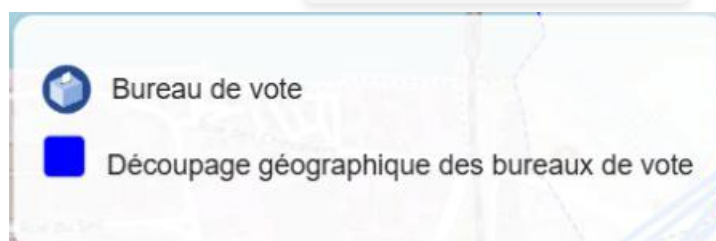
Les données linéaires, représentant les lignes de tramway, n'apportaient pas de réelle plus-value à la thématique principale de notre projet. Après discussion, nous avons décidé de les conserver dans leur état brut, sans personnalisation ni traitement supplémentaire. Cela permet de visualiser leur emplacement en arrière-plan, tout en mettant davantage l'accent sur les autres couches plus pertinentes pour notre analyse.

### Graphiques et légende interactive

Pour réaliser le graphique en donut permettant de visualiser les résultats des bureaux de vote, j'ai utilisé l'IA de ChatGPT ainsi que des ressources disponibles en ligne. Ces outils nous ont permis de concevoir un graphique interactif intégré à la carte. Deux bibliothèques ont été utilisées pour ce graphique : **Chart.js** et son plugin **Datalabels**, offrant des visualisations dynamiques et personnalisables.



La légende a été créée avec **L.control** de Leaflet et positionnée en bas à droite de la carte. Elle a été définie en HTML dans la méthode `onAdd`.



### Affichage de l'image du gagnant :

Une fonctionnalité a été implémentée pour afficher la photo du candidat gagnant de chaque bureau de vote lorsqu'un polygone est cliqué sur la carte. Cette interaction est gérée à l'aide de la fonction `afficherPhotoGagnant`.

Lorsqu'un utilisateur clique sur un polygone, les propriétés associées au gagnant (par exemple, M. Emmanuel MACRON ou Mme. Marine LE PEN) sont récupérées. En fonction de cette information, la fonction met à jour dynamiquement la source de l'image affichée



dans un élément HTML dédié (identifié par winner-photo). Si aucune information sur le gagnant n'est disponible, aucune image n'est affichée.

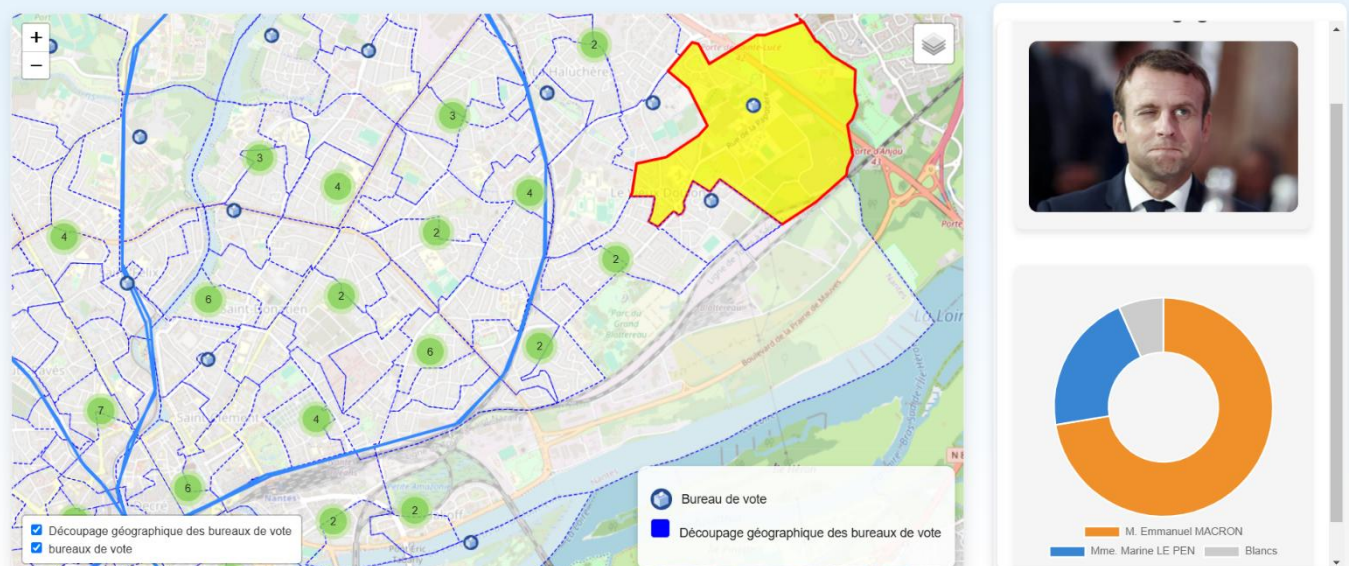
#### 4. Modification du CSS

Les premiers résultats obtenus avec un CSS basique n'étaient pas satisfaisants visuellement. Il a donc été décidé d'améliorer l'apparence de la page web pour la rendre plus attrayante. Pour cela, j'ai utilisé l'IA de ChatGPT pour générer des idées et du code CSS adapté à mes besoins. En complément, d'autres sources externes ont été consultées pour affiner certains détails esthétiques et s'assurer d'un rendu moderne et harmonieux. Ces modifications se sont concentrées uniquement sur l'aspect visuel de la page, sans toucher aux fonctionnalités existantes. L'objectif était de rendre la navigation plus agréable et intuitive pour les utilisateurs.

#### 5. Hébergement du projet :

Ce projet est hébergé sur le dépôt GitHub de l'utilisateur Paulgll, dans le dossier webcarto. Il est accessible publiquement via le lien suivant : [webcarto-project](#).

#### Carte interactive des résultats du 2nd tour des élections présidentielles 2022 pour la ville de Nantes



Carte réalisée dans le cadre d'un cours du Master Géographie Numérique à l'Université de Lyon 2. Auteurs : Paul Goullencourt, Rémi Membrede. Données issues de [data.nantesmetropole.fr](#).

