

Documentation Développeur

Cette documentation décrit le fonctionnement de chaque programme du projet développement CamaYeux.

1/ Fonctionnement global

Le projet CamaYeux est divisé en 2 cartes web. On accède d'abord à une page d'accueil, par le fichier *index.html*. Deux liens sur cette page permettent d'accéder aux deux cartes.

La première est destinée aux daltoniens, et des fonctions permettent d'y modifier les couleurs de la couche actuellement visible sur la carte web. Ces fonctions, ainsi que le code de mise en place du fond de carte, sont comprises dans le fichier *dalto.js*.

La deuxième, destinée aux personnes malvoyantes, regroupe 4 fonctionnalités différentes, et est donc plus complexe. Ces 4 fonctionnalités sont : la mise en évidence d'objets recherchés, l'activation d'un cadre de couleurs, l'aide vocale et la lecture vocale. Cette deuxième carte web consiste donc en un fond de carte de base, à partir duquel les 4 fonctions peuvent être activées ou désactivées en cochant ou décochant des boutons.

Ici, nous avons donc une carte 'de base', codée dans les fichiers *index.js* et *malvoyants.html*, qui est munie de boutons. En plus de cette carte, les boutons appellent les autres fonctionnalités, codées sur les fichiers *surbrillance.js*, *cadre.js*, *texttospeech.js* et *annyang.js*. Ces fonctions s'occupent de faire les mises à jour nécessaires sur la carte web de base, notamment en modifiant les div html du fichier *malvoyants.html*. Chacune de ces fonctionnalités est donc codée séparément et accessible dans son fichier respectif.

Exemple: le fichier surbrillance.js lance un EventListener sur le bouton 'Rechercher une gare' de la carte web. Quand l'utilisateur clique sur ce bouton, le fichier surbrillance.js est appelé, et va créer une couche graphique qui viendra s'ajouter

2/ Description exhaustive

Page d'Accueil :

- **index.html**

HTML de la page d'accueil. Le body est divisé en 3 div principales :

Div 'haut' : bandeau du haut de la page, titre et liens vers les images des logos

Div 'content' : partie principale composée de 3 panneaux images (classe *image*) et 3 panneaux tournants (classe *flip-card*). Au survol, les panneaux tournants se retournent (*.flip-card:hover* dans *accueil.css*) et révèlent des boutons activant leur lien.

Cette div principale est séparée en 3 sous-div ('gauche', 'milieu' et 'droite'), et chaque panneau a une sous-div ('ghaut', 'gbas', 'mhaut', 'mbas', 'dhaut', 'dbas') pour pouvoir accéder à chaque élément en fonction du besoin.

Div 'resume' : texte explicatif du projet CamaYeux, présentant les acteurs et la bibliographie. Le texte est en bas de la page, et un des liens des flip-cards décale la fenêtre pour y accéder.

- **accueil.js**

Gère l'activation des liens 'flip cards' redirigeant de la page d'accueil vers les cartes web.

Deux EventListener écoutent les panneaux Daltoniens et Malvoyants. Une fois ceux-ci déclenchés au clic, la fenêtre est redirigée vers la page correspondante (*window.location.href = ...*).

- **accueil.css**

Fichier de style de la page d'accueil. Positionnement des panneaux en affichage flex, définition des polices, de la taille des images et des fonds d'écrans.

Partie 'flipping cards' : gestion de l'apparence des panneaux tournants, qui ont chacun 2 faces (*flip-card-front*, *flip-card-back*). Au survol, les panneaux sont retournés de 180° (lignes 155- 157).

Carte Daltoniens :

- **dalto.html**

HTML de la carte web pour daltoniens. Le fond de carte épuré ainsi qu'un fond plus détaillé ArcGIS sont chargés, et des boutons activant ou désactivant l'ensemble des fonctionnalités suivantes sont chargés sur le bandeau supérieur.

Le body contient une grande div (*viewDiv*), sur laquelle est chargée le fond de carte ArcGIS. Par-dessus la carte on trouve une div destinée au volet de paramétrage de la carte (*volet-clos*, qui s'étend dans la div *volet*).

Volet contient 4 boutons, qui activent respectivement trois choix de couleurs prédéterminées pour la carte, et un mode de couleurs personnalisables.

Au début du body, il y a aussi une petite div correspondant à l'outil loupe (*widgetLoupe*), activable sur la carte.

- **dalto.js**

Fonctionnement de la modification des couleurs de la carte pour les daltoniens. L'utilisateur sélectionne une des trois options de base (deutéranopie, protanopie ou tritanopie), ou bien le choix personnalisé, depuis le volet ouvrant. Des EventListeners attendent cette sélection, et affectent les couleurs appropriées au renderer de la carte web.

- **dalto.css**

Fichier de style pour la carte web des daltoniens.

Carte Malvoyants :

- **malvoyant.html**

Contient le code HTML de la carte web pour les malvoyants.

On retrouve une grande div *ViewDiv* qui permet de charger le fond de carte de base, ainsi qu'une grande div pour les fonctionnalités du bandeau.

Cette page contient un bouton de retour à la page d'accueil

- **index.js**

Gère le fonctionnement général de la carte web pour malvoyants.

Création du fond de carte et gestion du bandeau supérieur, permettant d'activer les autres fonctionnalités : Mise en Evidence, Cadre, aide Vocale, Lecture Vocale.

- **surbrillance.js**

Gère le fonctionnement de l'outil de recherche de gares et d'affichage du résultat qui clignote. Lorsque l'option 'recherche de gare' est activée, le volet de la carte s'active et cette fonction est appelée.

L'utilisateur fait une recherche de gare, soit par un clic sur la ligne de train ou de métro désirée, soit par l'entrée du nom d'une gare en particulier. La gestion du volet de sélection est faite ici. La sélection des gares correspondantes, ainsi que l'affichage des suggestions de gares, sont faites par des requêtes SQL sur la couche active de la carte. La fonction utilisée est *QueryFeature* d'*ArcGIS Online*, qui va interroger la couche pour trouver les gares que l'on recherche, soit par l'attribut de leur nom, soit de leur ligne.

Les gares sélectionnées sont ensuite mises en évidence sur la carte par un clignotement : une nouvelle couche est rajoutée sur la carte, avec deux *renderers* qui alternent, pour la faire apparaître successivement en jaune puis en bleu foncé.

- **cadre.js**

Gère le fonctionnement du cadre de couleurs.

Quand le bouton 'cadre' est activé, le cadre est d'abord rajouté sur la carte web. (modification du *display none* en *flex* pour l'activation).

La mise à jour du cadre est effectuée automatiquement dès que la vue actuelle est modifiée. Alors l'ensemble des points de la couche active de la carte est traité par la fonction *RefreshCadre*. Les coordonnées de l'emprise actuelle de la carte sont converties en Lambert 93 (les coordonnées des gares). Chaque gare est ensuite positionnée par rapport à l'emprise de la carte.

Les points positionnés dans un certain rayon du centre de la carte et à l'extérieur de l'emprise actuelle sont comptabilisés. Ici, le rayon est calé à 3 fois l'emprise actuelle de la carte, ce qui permet de ne pas prendre en compte les points relativement éloignés de l'utilisateur.

5 paliers de couleurs sont déterminés selon le nombre total de points détectés, allant du jaune pâle au rouge foncé. Si une direction contient par exemple entre 2/5 et 3/5 des points détectés dans la zone, la barre correspondante du cadre sera en orange. Les couleurs à affecter aux paliers successifs sont définies dans une liste. Enfin le style CSS de chaque barre du cadre est ensuite modifié en fonction de la densité de points qui y sont attribués.

- **texttospeech.js**

Gère l'activation des fichiers audio de la lecture vocale.

Si le bouton 'lecture audio' est coché, des EventListener activent les fichiers audios au survol ou au clic des boutons et des options de la carte. Sur les boutons du bandeau supérieur et du volet de recherche, les fichiers enregistrés par l'équipe CamaYeux sont chargés (dossier Audio). Pour le nom des 923 gares, lues lors d'une recherche de gare, les fichiers chargés proviennent de l'API mespeak.js.

- **mespeak.js et mespeak_config.json**

Fichiers de configuration et de gestion de la lecture vocale.

Fichiers tirés d'une API libre de lecture de texte sur le site suivant :

<https://www.masswerk.at/mespeak/>

- **annyang.js**

Gère la reconnaissance vocale.

Quand la reconnaissance vocale est activée, différentes commandes prononcées dans le micro sont reconnues grâce à l'API gratuite Annyang.js. Les commandes (haut, bas, nord, sud, plus, moins, activation du cadre, de la recherche...) sont retranscrites et modifient le code html de la carte de base, ou bien la vue actuelle de la carte.

- **style.css**

Fichier de style de la page malvoyant.

Définit le style de la vue : le cadre de couleur, et le volet de recherche de gares.

Dossiers annexes :

- **Audio :**

Dossier contenant les fichiers audios pour l'option d'aide vocale (voix enregistrée)

- **Berlin-sans-fb-demi-cufonfonts :**

Dossier de ressources pour la police berlin.

- **Mespeak :**

Fichier de ressources pour l'API de reconnaissance vocale meSpeak.

- **Pictures :**

Dossier contenant les images, logos et captures d'écran appelés dans le projet.

- **Voices :**

Dossier de ressources pour la lecture automatique du nom des gares (voix générée).