

Objectifs

- Continuer le prototype individuel du projet synthèse.
- Prototyper l'adaptation du script démo qui permet d'afficher les épigraphes sur la carte Google Maps. Ajouter aussi la navigation entre les pages fiches, carte et chasse.

Consignes

- **Faire le prototype à l'aide des spécifications fournies.**
- **Remise sommative de 15%**

0. Préparation à l'exercice

Terminer le prototype du dernier cours.

Avant de commencer, le prototype a déjà :

- Une page fiche avec des données dynamiques (données JSON). Le nom de la page varie selon le secteur d'activité.
- Cette fiche dynamique récupère le numéro de la fiche à afficher à partir de la Querystring qui a été tapée à la main dans l'URL.
- Cette fiche possède un formulaire de soumission des éléments et un lien pour débiter la chasse.
- Une page chasse permet de débiter la chasse à l'aide d'un bouton. Ceci avec des numéros de fiches tirés au sort dans chacune des catégories d'éléments.
- La page fiche permet la soumission d'un élément. Une vérification des éléments (dans le "localStorage") s'effectue et déclenche l'affichage des messages de réussite ou d'échec appropriés.

1.1 Continuer le prototypage avec le Récit #11

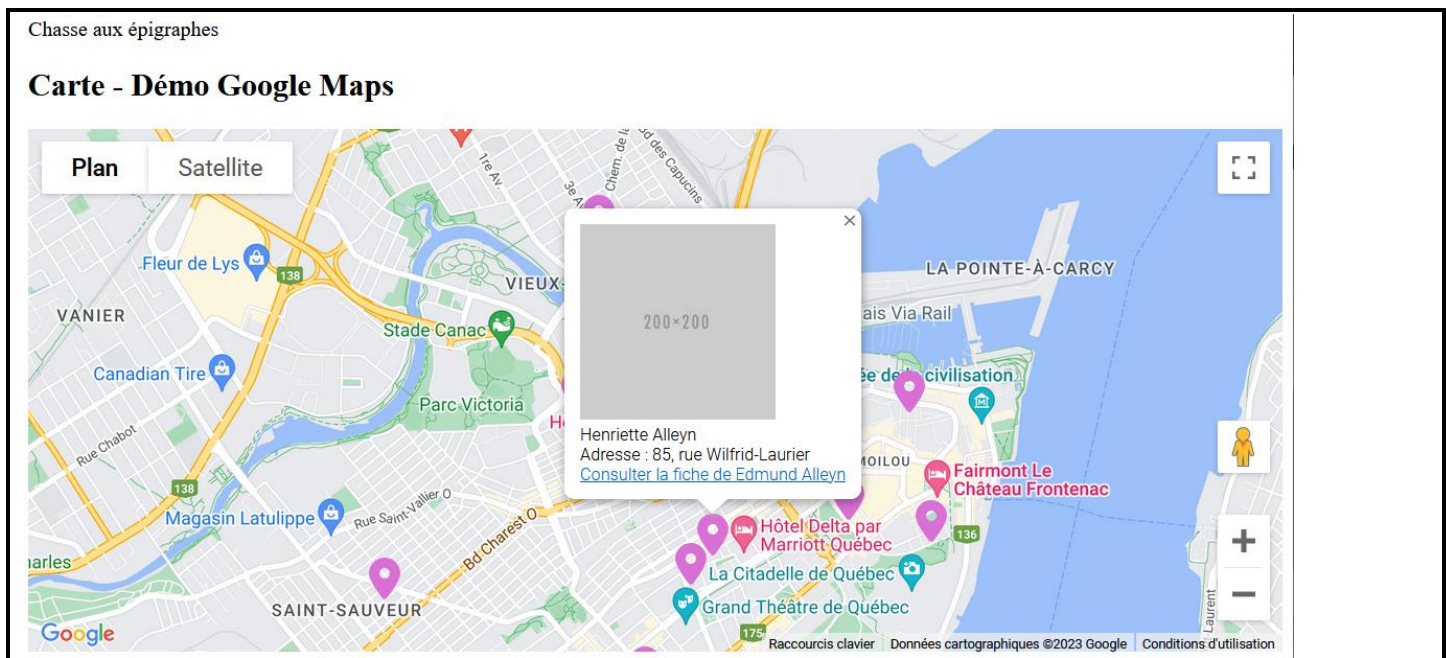
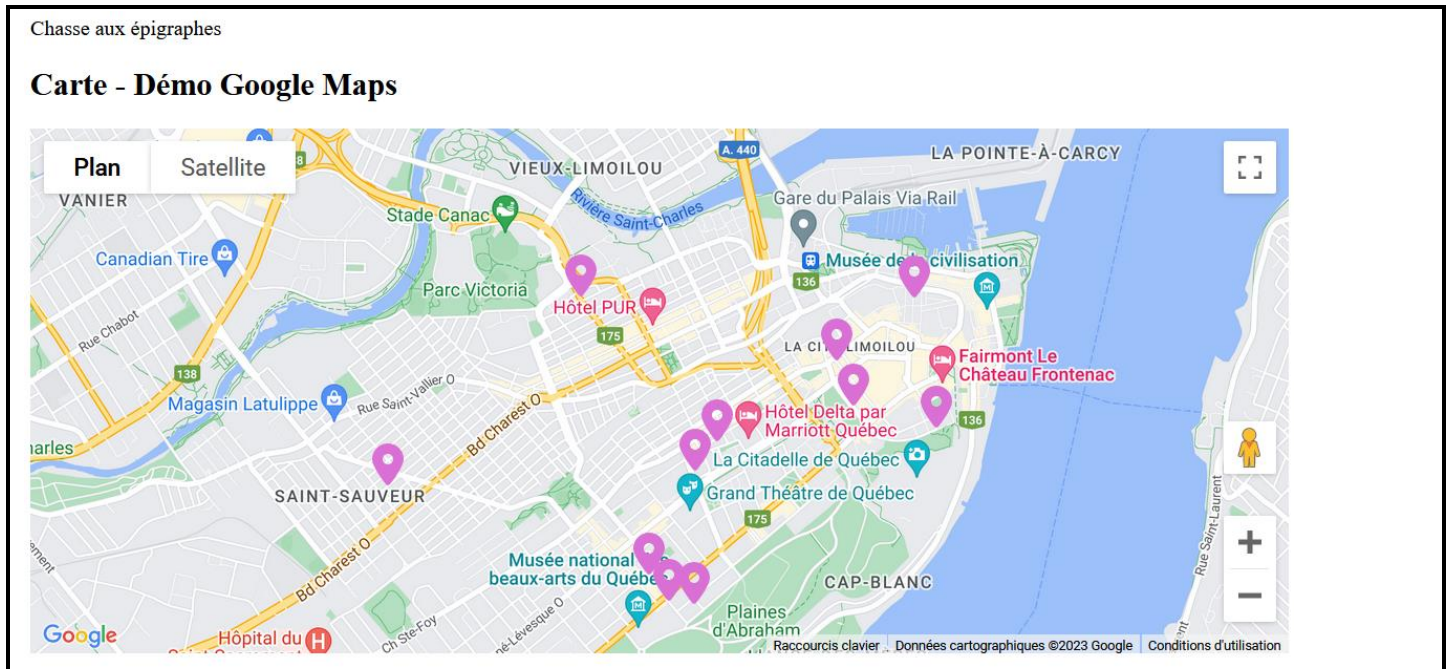
Explorer les épigraphes à partir de la carte
En tant qu'internaute :
<i>Je souhaite utiliser une carte (API de Google Maps) Afin de situer l'emplacement géographique des plaques commémoratives</i>
Critères d'acceptation
<i>Étant donné que l'API de Google Maps permet de situer sur la carte du centre-ville de Québec chaque plaque d'après ses coordonnées de latitude et de longitude. On affichera des marqueurs interactifs pour signaler leurs positions. Lorsque l'INTERNAUTE clique l'un des marqueurs Alors une petite fenêtre s'affiche pour donner l'image du lieu ainsi que le nom complet du personnage et le lien d'accès à sa fiche.</i>

1. Pour le 3^e cours de prototypage, tout le monde doit créer sa page carte.html dans son arborescence de prototype individuel.
2. Nous allons étudier, utiliser et adapter le script de démo de l'API Google Maps présenté en classe. Celui-ci va créer les repères sur la carte à partir de la lecture des épigraphes du fichier JSON, il te restera seulement à faire le html pour faire apparaître les informations au survol du repère.

-> ATTENTION #1 - Lors de tes tests et modifications au JavaScript, il faut recharger la page du navigateur en appuyant sur la touche majuscule (shift) afin qu'il n'y ait rien de vieux conservé en cache.

-> ATTENTION #2 – Vous devez utiliser **votre propre clé** de l'API de Google Maps ou celle du professeur si vous ne possédez pas de carte de crédit pour en faire la demande. La procédure pour créer votre clé d'API est disponible dans un autre pdf présenté en classe.

Voici un aperçu de la carte Google Maps dans la page carte.html :



3. La fenêtre qui s'ouvre sur le clic d'un repère doit contenir :
 - une vignette (Google Street view)
 - prénom et nom du personnage
 - l'adresse de l'épigraphie
 - un lien "En savoir plus sur Prénom et nom du personnage".

Voici une saisie d'écran du prototype de départ où l'on doit améliorer le html de l'infobulle afin qu'il adapte aux données de la fiche d'épigraphie qui a été choisie sur la carte Google :

```

creerMarqueurs: function () {
  for (let strIdEpigraphe in objJSONEpigraphe) {
    const objEpigrapheCourant = objJSONEpigraphe[strIdEpigraphe];

    const strGabaritContenuInfobulle = `<div class="infobulle">
      <div class="image"></div>
      <div class="titre">${objEpigrapheCourant.PRENOM} Alleyn</div>
      <div class="adresse">Adresse : 85, rue Wilfrid-Laurier</div>
      <a class="btn" href="fiche.html?id=e0001&titre=Edmund-Alleyn">Consulter la fiche de Edmund Alleyn</a>
    </div>`;

    console.log(strGabaritContenuInfobulle);

    const objInfobulle = new google.maps.InfoWindow({
      content: strGabaritContenuInfobulle
    });

    const objMarqueur = new google.maps.Marker({
      position: new google.maps.LatLng(objEpigrapheCourant.LATITUDE, objEpigrapheCourant.LONGITUDE),
      title: objEpigrapheCourant.IMAGE.TITRE,
      map: this.objMap,
      icon: `${this.strUrlImages}marqueurs/lieu.png`,
      infowindow: objInfobulle
    });

    objMarqueur.addListener('click', function () {
      objCarte.fermerToutesInfobulles();
      this.infowindow.open(objCarte.objMap, this);
    });

    this.arrMarqueurs.push(objMarqueur);
  }
},

```

Tu peux remarquer l'ajout d'un `console.log(strGabaritContenuInfobulle)` qui te permet de visualiser et vérifier ce que tu as créé par JS comme html afin de repérer les erreurs potentielles (html incorrect, mauvais dossier d'images, etc.).

4. Commencer par aller chercher ce qui est facile dans le JSON : Le nom et l'adresse.
5. Le Lien "En savoir plus sur Prénom et nom du personnage" appellera la page de fiche (renommée) et avec la bonne `Querystring`.
 - Exemples de nom de page et de `Querystring` adéquats - écrire seulement un chemin relatif :
 - <http://localhost:63342/fiches/arts-visuels.html?id=e0001&titre=Edmund-Alleyn>
 - <http://localhost:63342/fiches/economie-et-politique.html?id=e0022&titre=Maurice-Pollack>
 - À noter que nous ajoutons le nom du personnage afin de favoriser le référencement
 - C'est `?cle=valeur&cle=valeur`
 - Le `&` permet d'ajouter d'autres combinaisons de clé/valeur.
6. Pour l'image de Google Street View, il faudra aller chercher sur le réseau les images qui correspondent à vos fiches et les mettre dans un dossier. Vous avez le droit de les renommer. Possible aussi d'ajouter un `width="200px"` (pour rapetisser l'image) au html généré en attendant d'avoir du css dans le projet d'équipe.

1.2 Mettre les 3 parties ensemble

7. Installer la navigation entre les pages `carte.html`, celles de fiches d'épigraphe et la page `chasse.html`
8. Tester si tout fonctionne bien.

1.3 Continuer le prototypage avec le Récit #12

Explorer les épigraphes à partir de la carte / Repérer les fiches visitées

En tant qu'internaute :

Je souhaite voir sur la carte (API de Google Maps) dans un état modifié les marqueurs des fiches visitées

Afin de discerner quelles sont les fiches qu'il reste à visiter

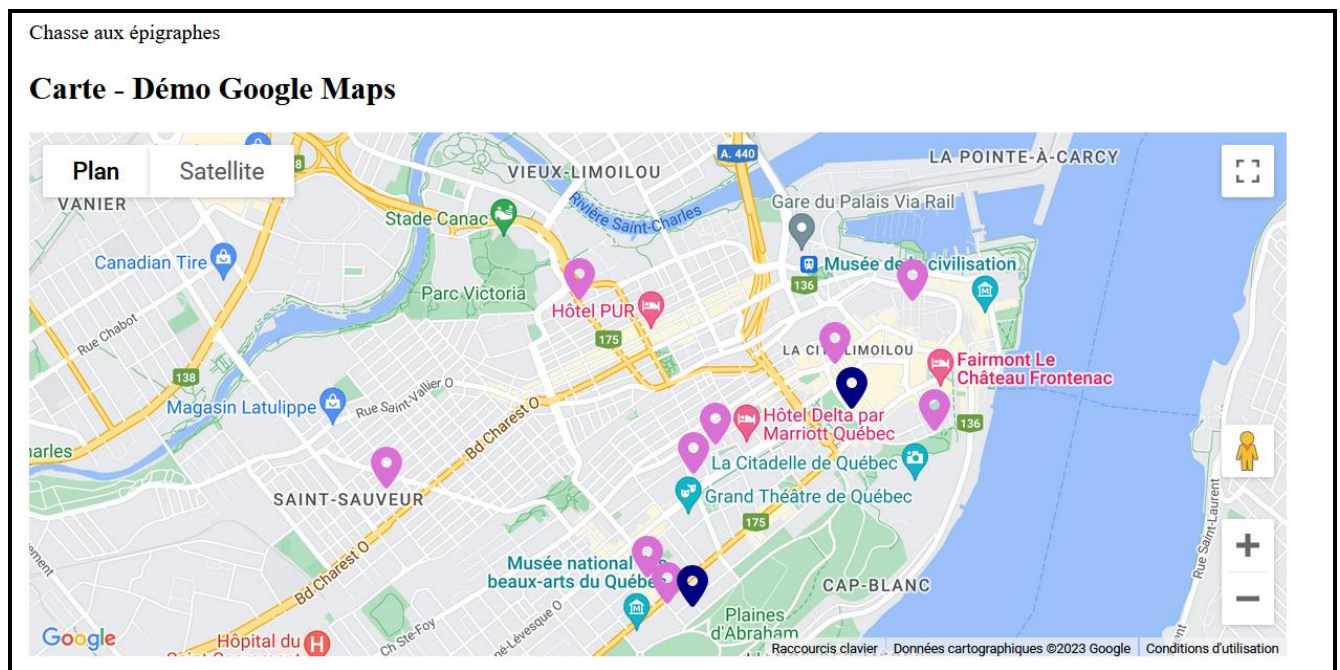
Critères d'acceptation

Étant donné que l'API de Google Maps permet de personnaliser les marqueurs

Lorsqu'une fiche a été visitée

Alors, programmer un état visité pour chaque repère dont la fiche a été consultée.

9. Vous devez afficher les marqueurs d'une couleur différente si la fiche a été visitée. Dans l'exemple ci-dessous, j'ai changé la couleur de 2 marqueurs pour indiquer qu'ils ont été visités.



10. Vous devrez modifier le code JavaScript de votre fiche pour enregistrer dans le localStorage si elle a été visitée ou non.
11. Vous pouvez maintenant utiliser ce que vous avez enregistré ou non dans le localStorage pour faire afficher le marqueur de la bonne couleur dans la carte. J'ai choisi de mettre les marqueurs non visités *orchid* et les marqueurs visités *navy* mais vous pouvez choisir des couleurs qui s'harmonisent mieux avec votre site.

*** Attention! N'oubliez pas de tester sur localhost pour que les variables du localStorage soient accessibles sur toutes les pages! ***

12. Avant de faire la remise individuelle par Léa, vérifier si les normes d'écriture du code sont respectées. Par exemple l'indentation, les préfixes et la documentation des fonctions. Les critères d'évaluation sont décrits plus bas.
13. Vérifier aussi que le minimum de balises id a été ajouté au balisage html. Utiliser modérément le `document.getElementById`. Utiliser au moins à un endroit le `document.querySelectorAll`.

Pour tester ta remise, il suffit d'utiliser la carte Google Maps afin d'accéder au contenu des différentes fiches (personnage ou objet ou lieu) de ton secteur d'activité. En visitant chacune des fiches de ton mandat, il te sera possible de soumettre un élément via le bouton de soumission ou d'utiliser le lien, pour aller à la page de chasse pour débiter une chasse, si elle n'est pas encore démarrée. Lors des soumissions d'éléments, un message de rétroaction approprié apparaîtra.

4 types de messages sont possibles :

- Aucune chasse démarrée
- Aucun bouton radio sélectionné
- Message de bonne réponse
- Message de mauvaise réponse.

Les exemples de construction des messages personnalisés sont donnés dans l'énoncé du dernier cours. Il sera aussi possible de naviguer facilement d'une page à l'autre du prototype.

1.4 Critères d'évaluation qui seront pris en compte :

Remise individuelle du prototype pour le prochain cours :

- Le prototype est séparé et indépendant du projet d'équipe.
- Aucun design du cours de design n'est présent dans le prototype (juste la programmation sur le HTML)
- Le prototype respecte l'architecture de dossier demandée.
- Les récits utilisateurs demandés dans les trois énoncés ont été prototypés.
- Le prototype est fonctionnel.
- La présence et qualité de la documentation dans le JavaScript.
- Le respect des normes d'écriture en JavaScript. Bonne indentation, préfixe de type et camelCase.

Modalité de remise :

Remise sommative par Léa de ton prototype personnel

Toute l'arborescence de dossiers de ton prototype sera à zipper et remettre par Léa. Cependant, **mettre seulement les images requises par tes pages**, dans le dossier images (un prototype n'a pas besoin de toutes les images du site en entier!). **Il doit rester hyper léger** et on y traite seulement le JavaScript. **Pas de CSS et de mise en page**. Ces aspects seront faits dans le projet d'équipe seulement.

→ Chaque personne sera évaluée selon son secteur d'activité choisi.