

VI - Information Visualization

Projet final de cours



MiGraph

Léonard FAVRE, Nicolas FEYER, Yael ISELI

décembre 2021

Table des matières

1	Introduction	2
2	Données sur la migration	2
3	Intention et public cible	2
4	Maquettes fonctionnelles	2
5	Résultat final	5
5.1	La carte	5
5.2	Les graphiques « Line »	7
5.3	Le graphique « Bubble »	7
6	Perspectives	8
6.1	Améliorations	9
6.2	Fonctionnalités	9
7	Conclusion	9
	Annexe : utilisation/installation	10

1 Introduction

Dans le cadre du cours de la visualisation de l'information, nous développons un projet autour de la visualisation de la migration mondiale au fil des années.

Dans ce document, nous décrivons le choix des données, l'intention et le public cible, ainsi que les *wireframes*. Puis nous présentons le résultat final, tout en apportant un avis critique quant aux maquettes initialement proposées et aux outils utilisés. Nous énonçons quelques améliorations et fonctionnalités supplémentaires qui sont intéressantes dans ce projet, pour finalement conclure ce travail.

2 Données sur la migration

L'idée initiale était de pouvoir tracer la migration mondiale tout en indiquant les tracés d'émigration, c'est-à-dire le pays d'origine et de destination pour une donnée de migration. N'ayant pas trouvé un tel *dataset*¹, nous avons trouvé des données² avec le format suivant :

- Données par pays/région et par année. À noter que les années ne sont pas forcément consécutives. .
- Évolution de la population et de sa densité
- Évolution de la migration nette et sa densité.
- Échelle salariale : *Low, Lower middle, Upper middle, High*.

À noter que les informations indiquées ci-dessus sont celles que nous avons utilisées pour le projet, et elles ne sont donc pas complètes par rapport au *dataset* existant.

3 Intention et public cible

Les écoles du secondaire II et les écoles supérieures, enseignants inclus, sont le public cible de notre visualisation. En effet, cet outil peut être intéressant pour un professeur afin de l'aider dans un cours de géographie ou d'histoire. L'enseignant peut s'aider de cet outil pour aborder des sujet tels que les migrations de population ou la géopolitique.

De plus, l'outil peut être utilisé pour renforcer un sujet enseigné en montrant l'impact qu'il a eu dans le monde. Cela peut être au niveau de l'évolution démographique mais aussi selon la migration.

Il est important de mettre de mettre en avant l'aspect dynamique et interactif de l'outil, mais aussi son côté comparatif. En effet, il est parfois difficile de mettre en perspective un nombre (e.g. : « Est-ce que ce nombre implique-t-il une forte hausse de la migration ou non ? ») et peut rapidement se sentir submergé lorsque de grands nombres se succèdent. Avec une comparaison, l'étudiant peut facilement attester à quel endroit l'impact est plus marqué qu'ailleurs.

4 Maquettes fonctionnelles

Avant de commencer à implémenter une solution logicielle concrète, nous nous sommes attelés à la création de maquettes fonctionnelles qui nous ont permis de garder la même ligne directrice tout au long du projet. En effet, ces maquettes fonctionnelles (*wireframes* en anglais) sont utilisées lors de la conception d'une interface utilisateur pour définir les zones et composants qu'elle doit contenir³.

1. Après une discussion avec le professeur, il se trouve qu'un tel *dataset* existe. Comme nous avons déjà commencé notre projet, nous avons décidé de continuer sur notre lancée.

2. Source : <https://www.kaggle.com/eliasdabbas/migration-data-worldbank-1960-2018>

3. [https://fr.wikipedia.org/wiki/Wireframe_\(design\)](https://fr.wikipedia.org/wiki/Wireframe_(design))

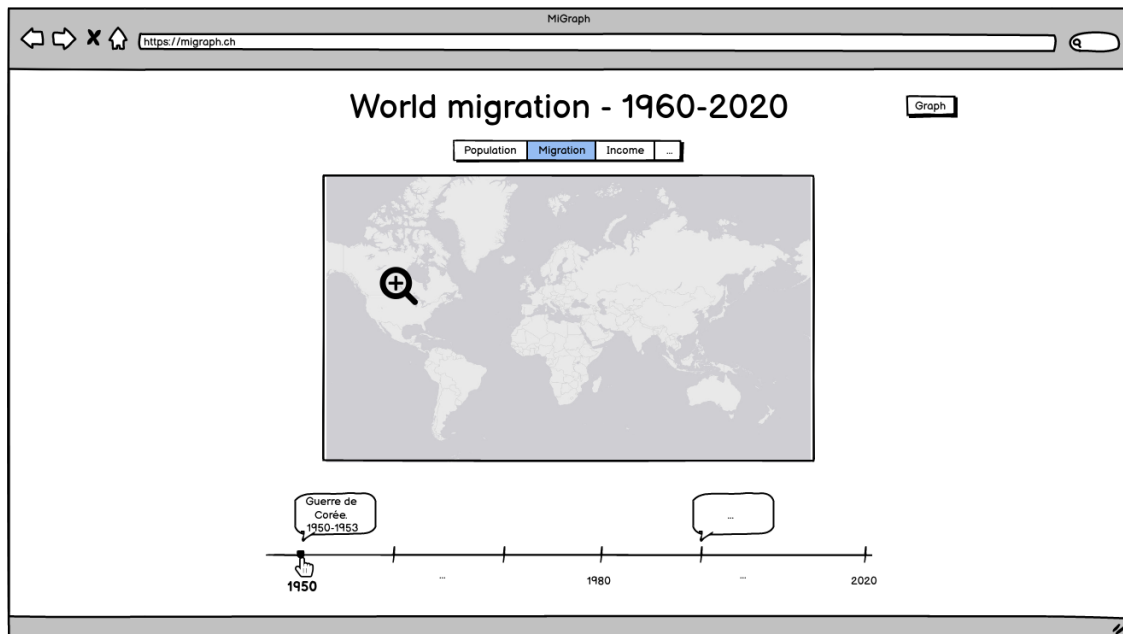


FIGURE 1 – Vue principale

Comme nous le voyons sur la maquette 1 qui présente la vue principale, nous avons trois éléments principaux (de haut en bas) :

- Différents boutons pour changer les données présentées
- Une carte du monde où sont présentées les données
- Un *slider* pour changer l'année

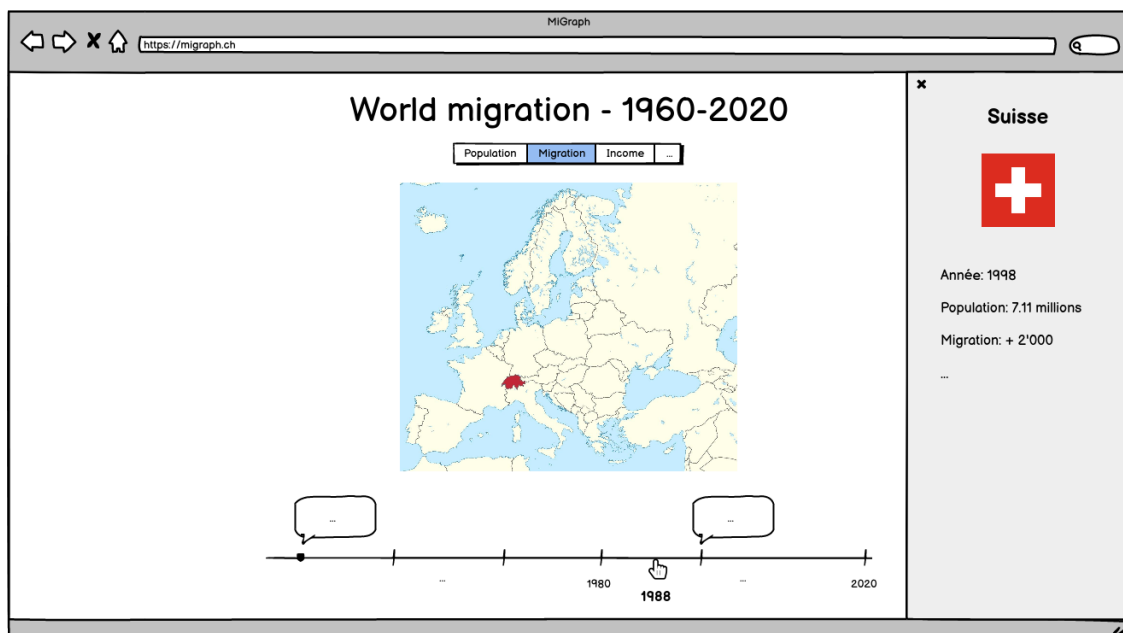


FIGURE 2 – Vue des détails d'un pays

La vue qui s'affiche lorsque nous cliquons sur un pays est présentée dans la figure 2. Diverses informations générales concernant le pays ainsi que son drapeau y sont affichées.

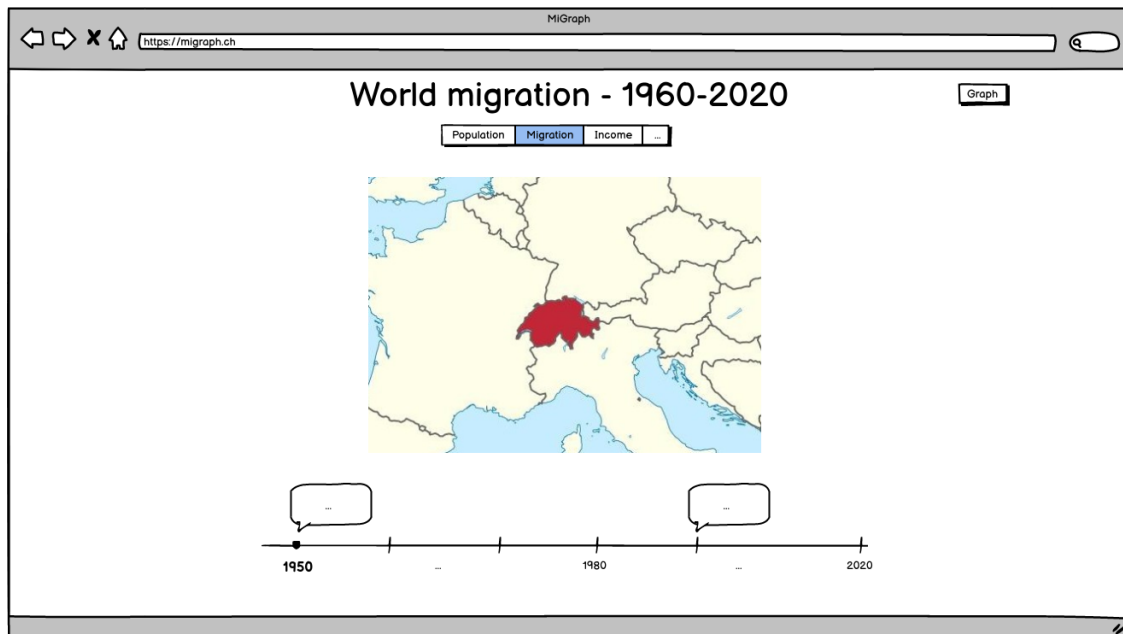


FIGURE 3 – Vue de la carte zoomée

Il est possible de zoomer la carte afin de voir plus précisément un pays, comme le montre la figure 3.

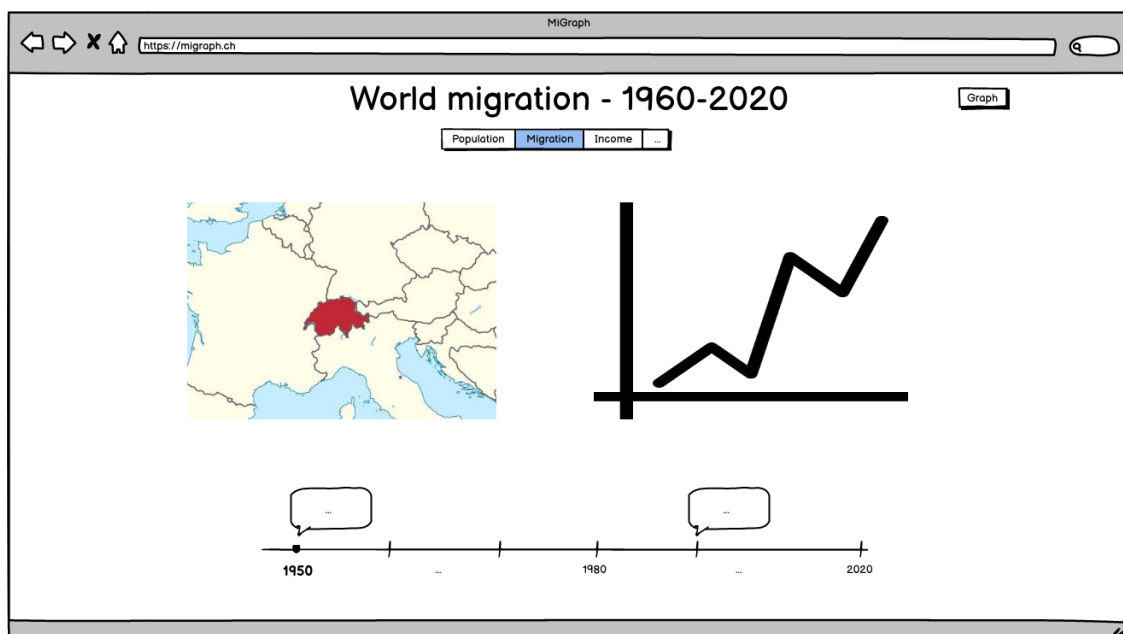


FIGURE 4 – Vue des graphique

Finalement, il est possible d'afficher des graphiques concernant le pays en cliquant sur le bouton situé en haut à droite comme nous pouvons le voir sur la maquette de la figure 4.

5 Résultat final

Pour l'implémentation, nous avons choisi d'utiliser la technologie `Vue.js` et d'y ajouter les bibliothèques `laue`⁴, `d3`⁵ et `vue-simple-svg-map`⁶ pour les composants graphiques (graphes, carte du monde, etc.). Pour ce qui est du design, nous avons utilisé `BootstrapVue`⁷. La disposition finale des éléments n'est pas totalement identique aux maquettes, et ce pour des raisons esthétiques, pratiques et d'utilisabilité. En particulier, il nous a semblé plus judicieux d'utiliser tout l'espace disponible de l'écran pour afficher les graphiques, afin d'augmenter la visibilité et les détails, sans trop charger une seule vue.

Notre résultat final est composé de trois représentations différentes :

- Le mode **carte** : représentation des données sur une carte du monde
- Le mode graphique « **Line** » : représentation des données par pays sous forme de graphique linéaire, avec la possibilité de combiner plusieurs pays dans un seul graphique.
- Le mode graphique « **Bubble** » : représentation des données sous forme de graphique à bulles regroupant tous les pays.

5.1 La carte

Cette représentation est l'élément central de notre visualisation. Elle permet à l'utilisateur de visualiser les données de manière géographique, aidant notamment à situer dans le monde les divers pays. De plus, le but de cette visualisation est d'inciter à la comparaison et de mettre en avant les différences, que ce soit entre les pays, ou les changements au fil des années. Ce n'est donc pas la valeur en soi qui est importante, mais sa valeur en fonction des autres éléments.

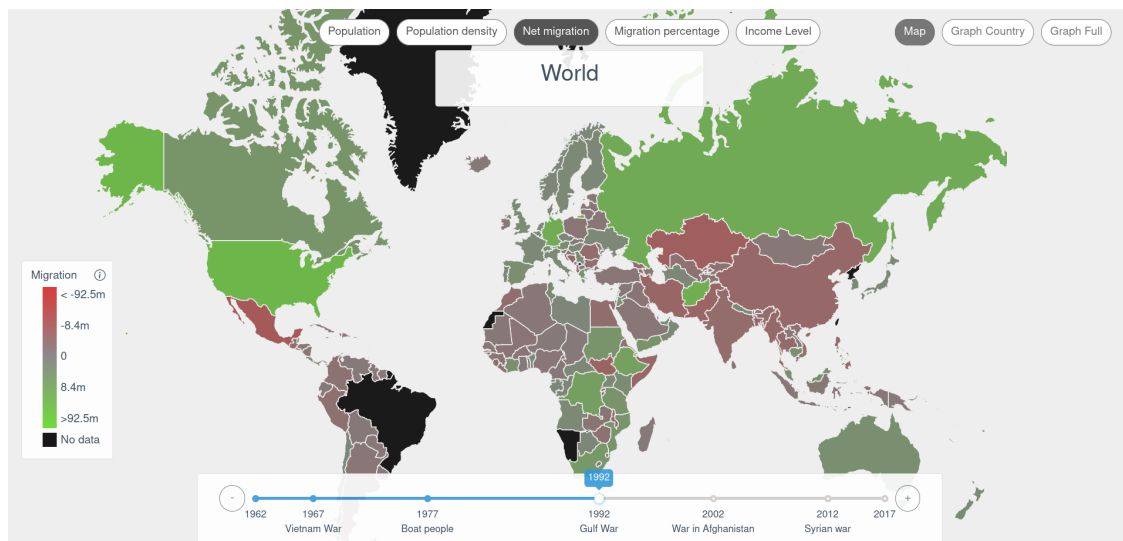


FIGURE 5 – Vue de la carte initiale

Comme le montre la figure 5, nous avons choisi pour cette représentation de visualiser les données à l'aide de couleurs, pour une donnée et une année définie. L'utilisateur peut choisir parmi les cinq types de données disponibles (population, migration, etc.) et utiliser un *slider* pour changer l'année représentée. L'utilisation de couleur a été choisie pour ne pas surcharger la carte tout en fournissant une information pertinente en termes de comparaison entre les pays. Une échelle

4. <https://laue.js.org/>

5. <https://d3js.org/>

6. <https://www.npmjs.com/package/vue-simple-svg-map>

7. <https://bootstrap-vue.org/>

au bord de la carte permet de se rendre compte de la signification des couleurs par rapport aux quantités.

Toutes les données représentées n'ont pas été colorées de manière identique. Les choix de coloration sont les suivants :

- *Population* : Évolution des couleurs en suivant une échelle logarithmique, du vert au rouge en passant par le orange. La valeur la plus faible (vert) correspond à la population la plus petite du *dataset*. Ces choix permettent de mieux distinguer les subtilités des pays avec une population de taille moyenne.
- *Densité de population* : Évolution des couleurs en suivant une échelle logarithmique, du vert au rouge en passant par le orange. Cela permet de mieux distinguer les différences entre les écarts conséquents en terme d'échelle pour ces valeurs.
- *Migration nette* : Évolution des couleurs en suivant une échelle logarithmique, du gris au vert pour les valeurs positives et du gris au rouge pour les valeurs négatives. Une valeur de décalage a été ajoutée afin de mieux distinguer un gris négatif d'un gris positif. Ainsi, les valeurs proches de zéro sont distinguables.
- *Pourcentage de migration* : Évolution des couleurs en suivant une échelle linéaire. Les autres choix sont les mêmes que pour la migration nette.
- *Niveau de revenu* : Quatre valeurs possibles pour cette donnée, nous avons assigné une couleur par valeur, en suivant l'ordre des revenus (vert : haut, jaune : moyen-haut, orange : moyen-bas, rouge : bas).

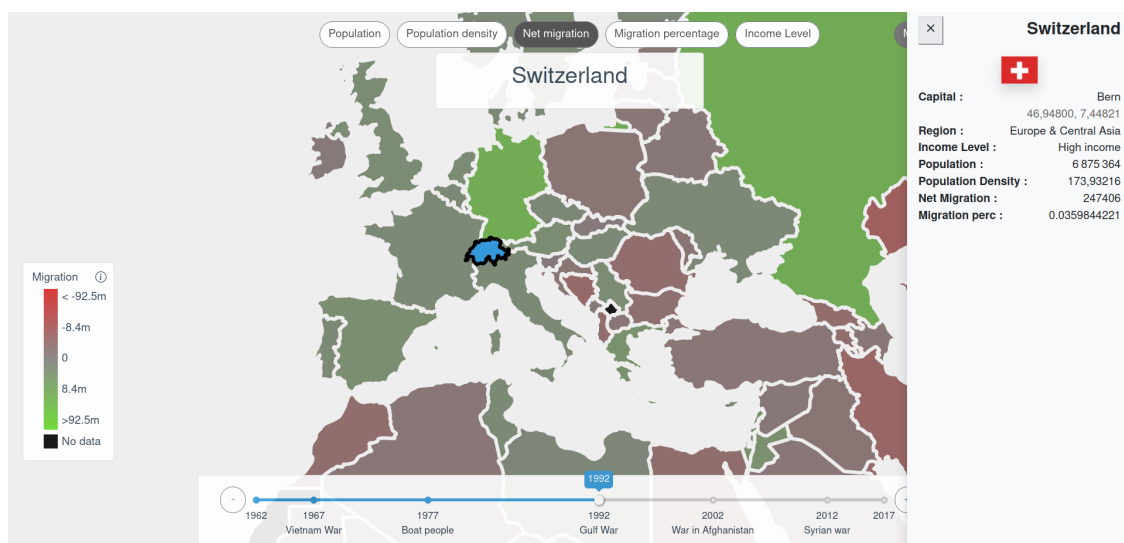


FIGURE 6 – Vue de la carte agrandie avec un pays sélectionné

En cas d'intérêt pour un pays, l'utilisateur peut cliquer dessus pour faire apparaître une barre latérale contenant des informations précises du pays en question (cf. figure 6).

Les informations de la barre latérale et de la carte se mettent à jour automatiquement en cas de changement d'année par l'utilisateur. La carte se met également à jour en cas de changement de catégorie de données. Grâce à la conception JavaScript de notre solution, tous les changements s'exécutent directement dans le navigateur du visiteur et ne nécessitent pas de recharger la page, rendant ainsi l'expérience plus agréable et interactive.

5.2 Les graphiques « Line »

L'utilisateur peut choisir de changer l'affichage pour montrer les données en graphique. Nous avons choisi la Suisse comme étant le pays sélectionné par défaut.

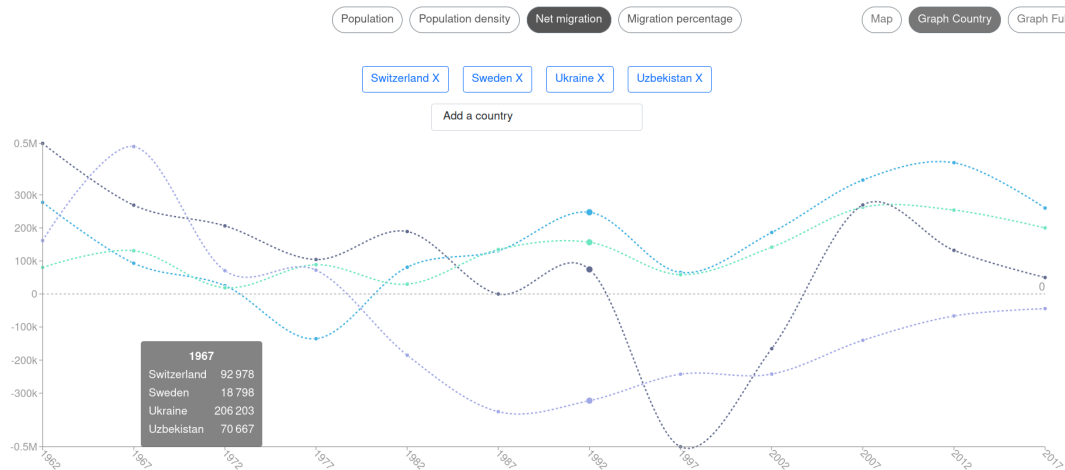


FIGURE 7 – Vue graphique « Line »

Les années sont représentées sur l'axe horizontal et sur l'axe vertical c'est l'une des quatre catégories de données qui est affichée (cf. figure 7). Comme le niveau de revenu n'évolue pas selon le temps dans notre *dataset*, cela n'avait pas de sens de le mettre en graphique évolutif. L'utilisateur peut ensuite choisir d'ajouter d'autres pays au graphique afin de comparer leurs données.

Pour cette partie, nous avons utilisé la librairie `laue`, mais elle s'est avérée limitante pour ce que nous voulions faire. Pour cette raison, le nombre maximum de pays affichés est limité et la modification en direct des données provoque parfois des erreurs d'affichage. Nous nous sommes rendu compte des problèmes trop tard dans le développement et la solution est fonctionnelle en l'état, raison pour laquelle nous avons gardé cet outil.

5.3 Le graphique « Bubble »

Dans cette dernière partie, nous avons choisi une représentation permettant d'avoir une vue globale de la situation mondiale.

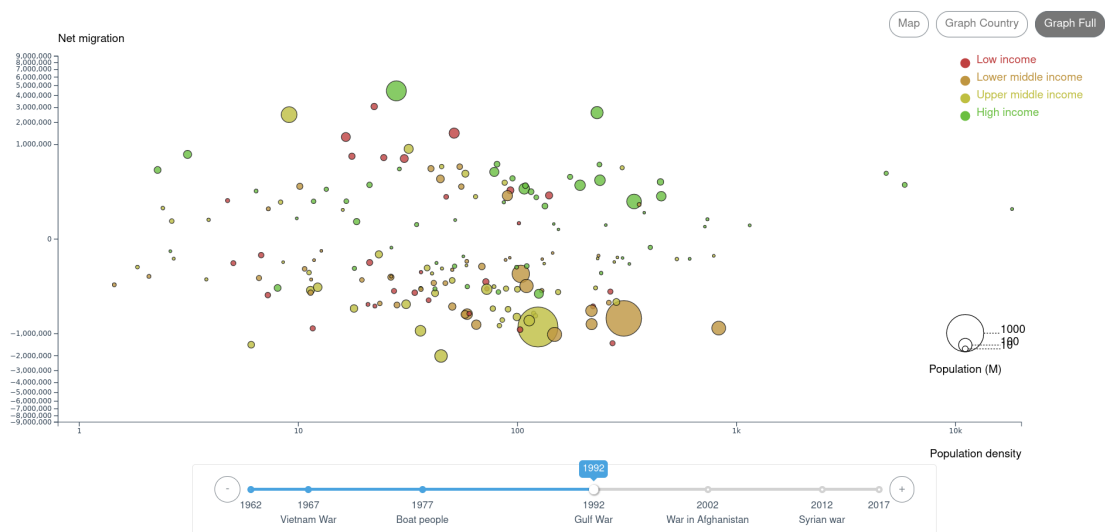


FIGURE 8 – Vue graphique « Bubble »

Comme présenté dans la figure 8, nous avons donc intégré un graphique de type « Bubble », afin de pouvoir afficher quatre données en une seule représentation :

- L'axe horizontal représente la densité de population
- L'axe vertical représente la migration nette
- Le rayon des bulles représente la population
- La couleur représente le niveau de revenu

Nous avons choisi d'utiliser une échelle exponentielle pour la migration nette, car cela permet d'avoir une visibilité plus précise des petites valeurs. Ainsi, nous notons clairement quelles bulles sont au-dessus ou en-dessous du zéro. De plus, les grandes valeurs sont moins éloignées les unes des autres, donnant un résultat plus lisible et compréhensible.

Pour la densité de population, nous avons choisi d'utiliser une échelle logarithmique, car les données diffèrent grandement d'un extrême à l'autre, alors que beaucoup se situent dans un intervalle plus concis au milieu. Cette représentation permet ainsi de distinguer clairement les valeurs ne se situant pas aux extrêmes.

Ainsi, tous les pays sont représentés sur ce graphique. Au passage de la souris, le nom du pays est révélé afin de pouvoir l'identifier. Un *slider* permet de faire évoluer l'année afin de voir l'évolution des pays au fil du temps.

Pour cette partie, nous avons utilisé la librairie **d3**, car la solution utilisée auparavant dans les graphiques « Line » ne permettait pas de faire des graphiques de type « Bubble ». Nous avons adapté un exemple⁸ trouvé en ligne afin de correspondre à notre situation et faire évoluer les données dynamiquement.

6 Perspectives

Dans ce chapitre, nous décrivons les améliorations possibles ainsi que les fonctionnalités que nous n'avons pas implémentées.

8. <https://www.d3-graph-gallery.com/>

6.1 Améliorations

Les améliorations que nous présentons sont considérées comme étant une meilleure version de ce qui existe déjà. Les nouvelles fonctionnalités sont décrites dans le chapitre suivant.

Les améliorations proposées sont les suivantes :

- Comme déjà mentionné précédemment, la librairie `laue` (pour les graphiques de type « Line ») semble contenir plusieurs erreurs et limitations quant à la personnalisation des graphiques. En effet, il semble y avoir un problème au niveau de la mise à jour de la légende lorsque nous ajoutons ou supprimons des pays à visualiser. Nous avons fait le choix de ne pas montrer une information erronée, ce qui implique qu'il faut aller chercher la correspondance entre une ligne du graphique et le pays concerné dans le *tooltip* (lorsque nous passons la souris sur la ligne). Cette solution n'est bien évidemment pas optimale, puisque cela implique une action supplémentaire de la part de l'utilisateur.
- Dans la vue des graphiques « Line », le sélecteur ne contient pas le symbole qui indique que c'est une liste déroulante. Après avoir tenté plusieurs solutions, nous pensons que cela est dû à une erreur de la librairie de design `BootstrapVue`. Nous ne voulions pas adapter tout notre code à un nouveau design, mais l'utilisation de `Vuetify` (outil de UI créé spécifiquement pour `Vuejs`) pourrait être une solution.
- Concernant les couleurs, nous n'avons pas pris en compte les problèmes que pourraient avoir les daltoniens. En effet, un daltonien de type protanope ou deutéranope confond le jaune, le vert et le rouge. Nous pourrions donc adapter notre choix des couleurs.

6.2 Fonctionnalités

Il y a deux fonctionnalités qui n'ont pas été implémentées par manque de temps. Tout d'abord, nous avons inclus quelques événements historiques et géopolitiques afin de montrer ce à quoi ressemble l'intégration de telles informations. Nous avons choisi ces événements car ils sont liés à une grande migration dans quelques pays. Néanmoins, il serait plus intéressant et complet d'utiliser un *dataset* existant qui contient toutes ces informations.

Une autre alternative, voire une fonctionnalité supplémentaire, serait d'intégrer un formulaire pour ajouter soi-même les événements qui nous intéressent. Cela aurait une grande valeur pour le professeur puisqu'il pourrait personnaliser l'outil et l'adapter à son enseignement. Le point négatif de cette alternative est la nécessité d'une base de données pour persister les données, ainsi que d'un compte pour identifier le professeur et sa personnalisation de l'outil.

7 Conclusion

Pour conclure, ce projet constitue un outil d'appui pour les enseignants d'histoire, de géographie, de géopolitique ou autres. En effet, l'enseignant peut cibler la région ou le pays ainsi que l'époque pour appuyer les notions vues en cours. De plus, le côté interactif de notre représentation apporte un aspect dynamique, voire ludique.

L'avantage de cette visualisation est sa représentation globale à travers le monde entier. Ainsi, l'outil n'est pas spécifique à un sujet et peut donc être utilisé régulièrement. Bien évidemment, il serait intéressant d'avoir des données plus anciennes, afin de couvrir une plus grande période de l'histoire.

Plus nous avançons dans le projet, plus nous réalisons que les librairies utilisées pour la carte du monde et les graphiques nous confrontaient à certaines limites. En effet, il est très compliqué de trouver une librairie qui apporte assez de liberté et de fonctionnalités pour pouvoir implémenter tous les éléments que nous avons prévus. Cela est aussi dû au fait que nous avons des besoins spécifiques. Néanmoins, nous estimons que de manière générale, ces librairies offrent un grand panel de choix de personnalisation. Cela est aussi un avantage de l'utilisation du `JavaScript` (et de `Vuejs` qui est aujourd'hui en plein essor), puisque sa forte utilisation dans le monde des applications Web encourage le développement régulier de nouvelles fonctionnalités.

Annexe : utilisation/installation

Le projet est disponible à l'URL suivante : <https://nicolasfeyer.ch/migraph/>

Pour installer le projet en local :

```
1  # clone repository
2  git clone git@github.com:nicolasfeyer/mse-vi-projet.git
3  cd mse-vi-projet
4
5  # install dependencies
6  npm i
7
8  # run vuejs
9  npm run serve
```

Le projet est ensuite accessible à l'URL suivante : <http://localhost:8080/migraph/>