|  |
| --- |
|  |
| Projet IHM-Geo |
|  |

|  |
| --- |
|  |

Erwan MAILHARRO

Arthur LECERT

Table des matières

[Introduction 2](#_Toc23452757)

[I. Description des personas 2](#_Toc23452758)

[a. L’élève 2](#_Toc23452759)

[b. Le professeur d’histoire-géographie 2](#_Toc23452760)

[II. Maquettes papiers 3](#_Toc23452761)

[a. Version Google-like 3](#_Toc23452762)

[b. Version avec une roue 4](#_Toc23452763)

[c. Version retenue 5](#_Toc23452764)

[III. Justification des choix retenus (maquette et rendu) 6](#_Toc23452765)

[IV. Difficultés rencontrées pendant l’implémentation 6](#_Toc23452766)

[Conclusion 6](#_Toc23452767)

# Introduction

Ce rapport se situe dans le contexte du cours « IHM, ergonomie et prototypage ». Il rassemble l’ensemble du travail que nous avons réalisé pour le projet final en Java. Ce dernier prend la forme d’une réponse à un appel d’offres pour la réalisation d’une application dont le sujet est donné ci-dessous :

« Le responsable informatique des lycées d’Annecy vous a demandé de réaliser une application d’aide à l’analyse de données géographiques.

L’idée est de pouvoir offrir un outil aux professeurs (d’histoire géographie plutôt) et aux élèves pour afficher dynamiquement certaines données selon des critères définis par eux.

Les contraintes sont d’avoir un client lourd, de fonctionner sans connexion et de proposer une vue cartographique.

Vous n’avez pas pu obtenir d’accréditation pour rencontrer des professeurs ou des élèves. »

Pour la réalisation de l’application, des jeux de données sont fournis. Le premier prend la forme d’un fichier CSV et contient les indicateurs de développement dans le monde de la Banque Mondiale avec leurs descriptions et leurs valeurs pour chaque pays entre 1960 et 2015. Le second se présente sous la forme d’un fichier JSON qui rassemble les coordonnées géographiques des régions et de tous les pays au format GeoJSON.

De plus, certains scripts sont fournis et permettent de lire et d’écrire les données dans la base de données à l’aide d’un Data Manager.

# Description des personas

Afin de développer une interface d’application qui corresponde au mieux au sujet, nous devions récapituler les différents futurs utilisateurs de l’application et leurs besoins à l’aide de personas.

# L’élève

Il s’appelle Kevin-Paul Martin. Il a 16 ans. Il utilise son ordinateur pour Facebook, Instagram, Youtube et Fortnite principalement.

Aisance numérique : 2/4

Expertise domaine : 0/4

Fréquence d’usage : 1 fois par semaine pour faire ses devoirs

Il travaille le moins possible et à la dernière minute donc il recherche une application épurée et simple d’utilisation pour pouvoir finir rapidement ses devoirs. De plus, il aimerait pouvoir l’utiliser pour des exposés ce qui demande une bonne esthétique.

# Le professeur d’histoire-géographie

Dominique Chenavas est un professeur d’histoire-géographie qui exerce son métier depuis 30 ans. Il en a maintenant 58. Passionné par son travail, il cherche à transmettre son savoir à ses élèves à l’aide d’une application ludique et intéractive.

Aisance numérique : ¼

Expertise domaine : 4/4

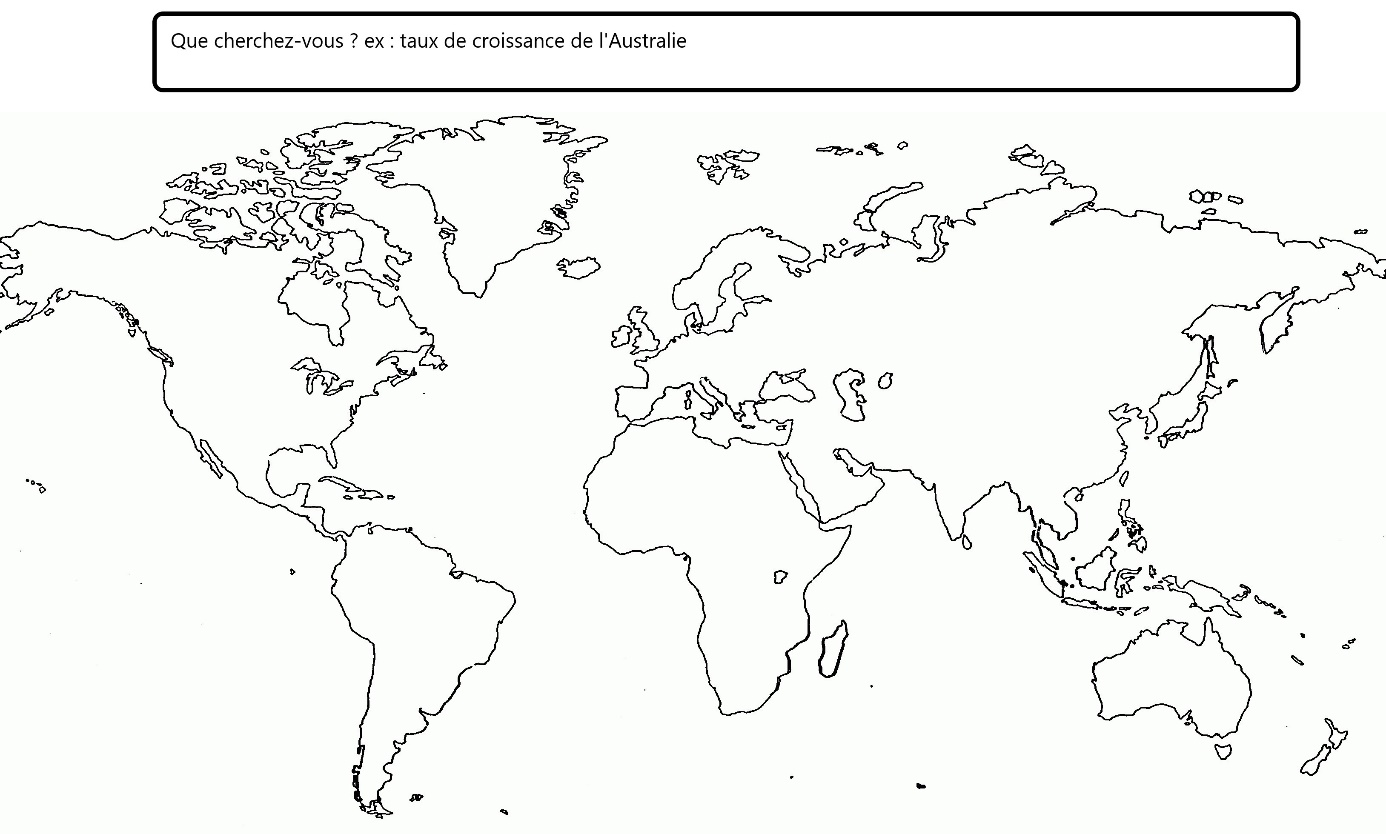
Fréquence d’usage : tous les jours

Ce professeur n’est pas très à l’aise avec les technologies. Il faut aussi que l’application lui permette de rentrer dans les détails de la base de données afin de faciliter son cours et s’il veut ajouter des anecdotes.

# Maquettes papiers

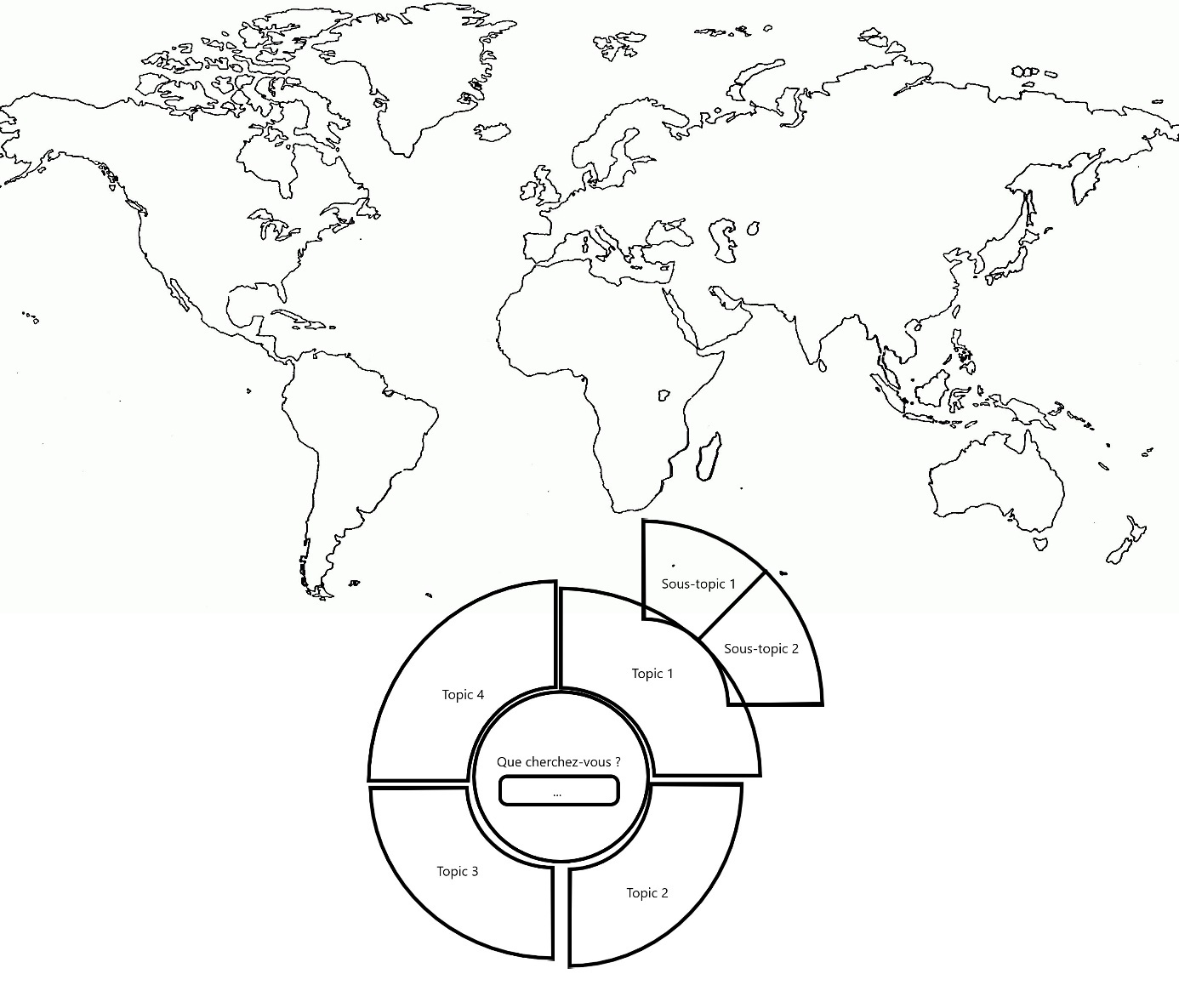
Cette partie contient toutes les idées d’interface que nous avons eues ainsi que leurs avantages et inconvénients.

# Version Google-like



Cette version très épurée de l’interface permet de trouver rapidement ce que l’on cherche sur l’application. Cependant, si l’utilisateur ne connaît pas le nom d’un pays ou d’un indicateur, il lui est difficile de le retrouver.

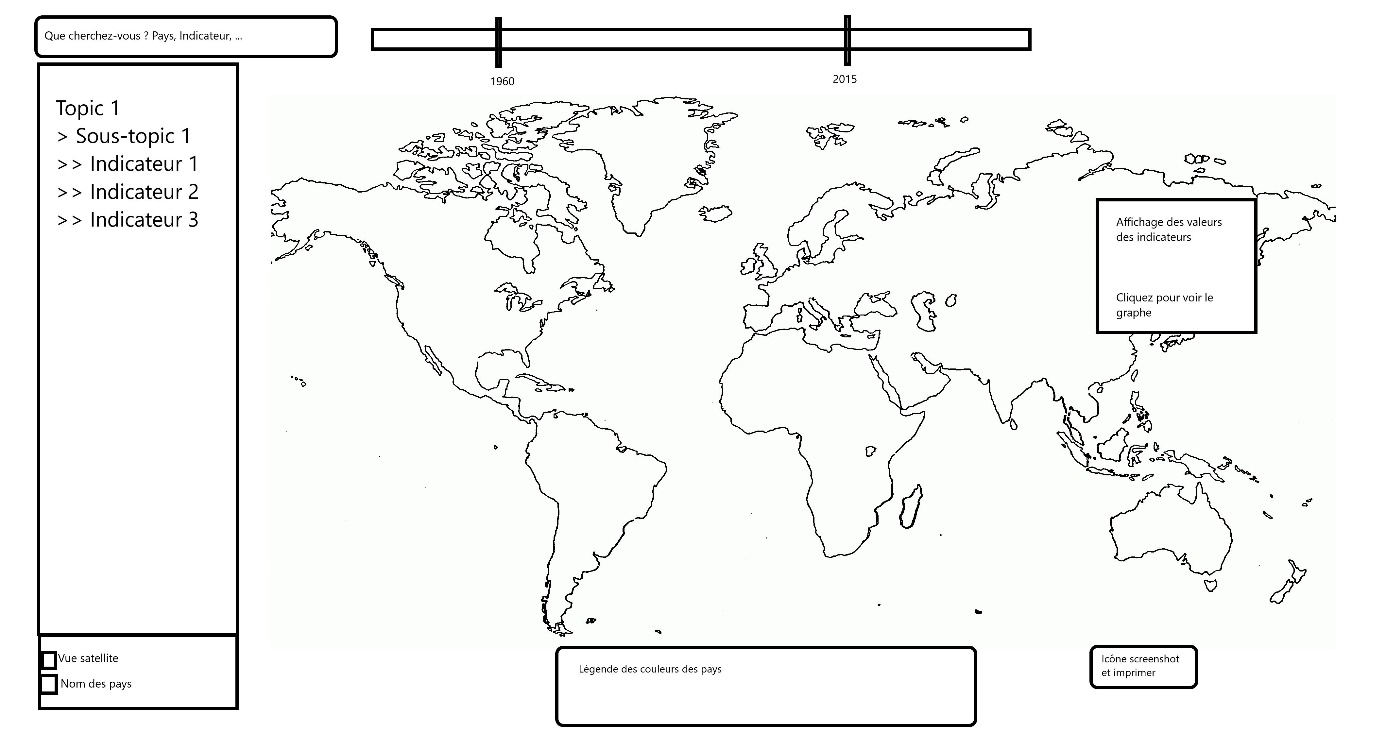
# Version avec une roue

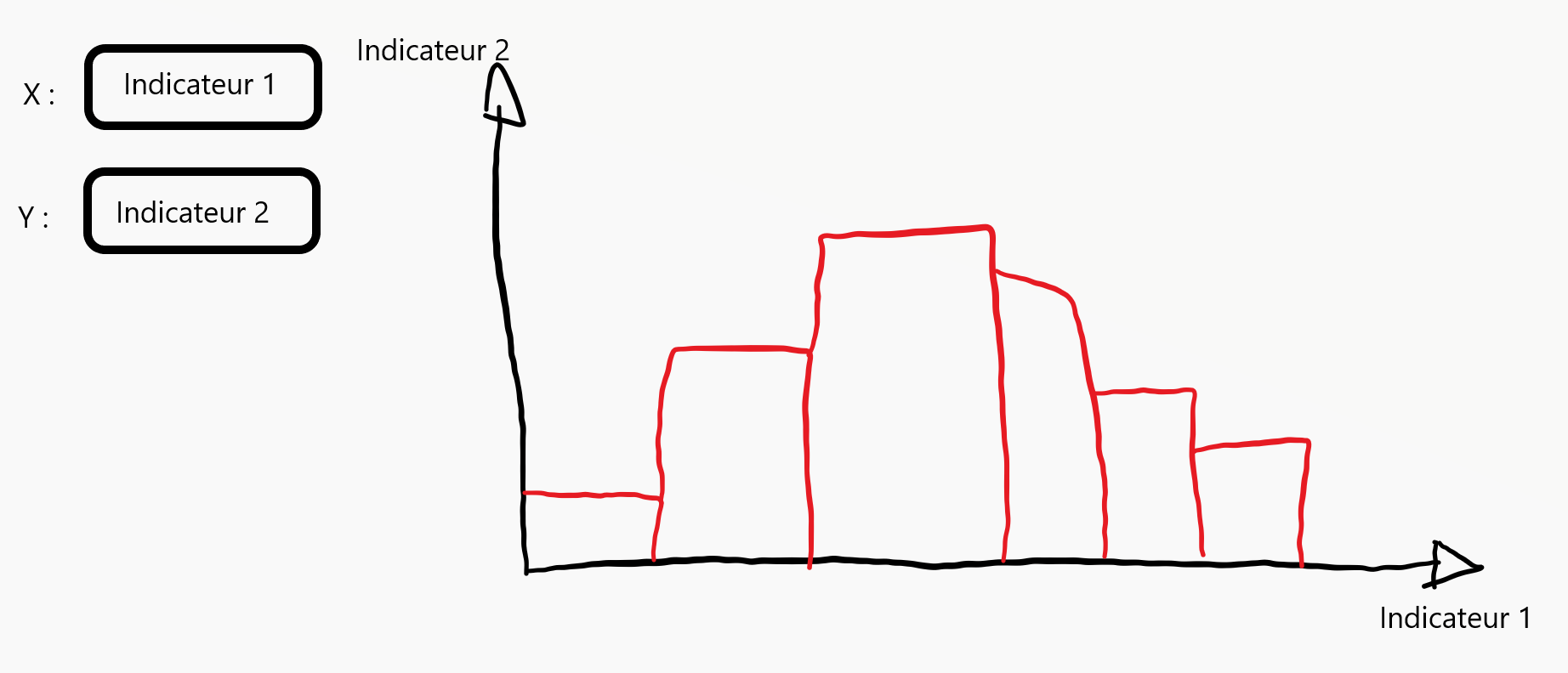


Cette interface a pour avantage d’être très intuitive à utiliser. De plus, elle permet de centraliser toutes les fonctionnalités de l’application dans une roue. On accède aux topics en les survolant avec la souris ce qui fait apparaître les sous-topics puis les indicateurs.

Cependant, la bibliothèque javafx ne propose pas d’éléments déjà implémentés de ce type ce qui complique grandement le développement de cette interface. Nous avons donc décidé de se concentrés sur une interface plus classique et facile à développer.

# Version retenue





Il s’agit de la version retenue dans la suite du rapport et que nous avons commencée à implémenter. Celle-ci contient deux pages : la première qui contient la vue générale des différents pays et la seconde qui permet l’affichage de différents graphiques.

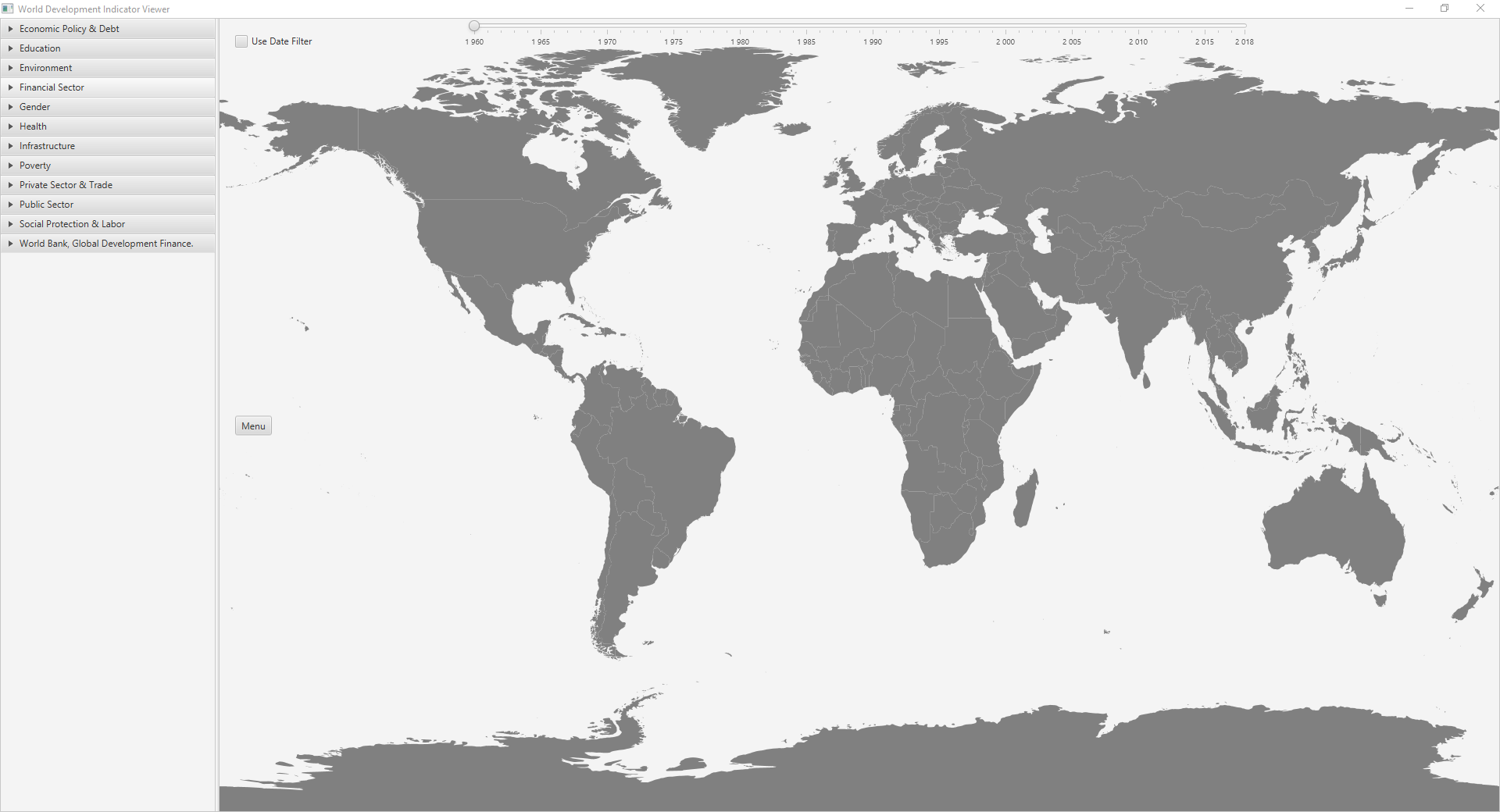
Sur la première page se trouve tout d’abord une barre de recherche semblable à celle de la version Google-like bien qu’elle ne demande pas autant de travail car d’autres éléments de l’interface comblent ses faiblesses. Le volet de navigation sur la gauche permet de sélectionner les indicateurs voulus ce qui colorie la carte du monde selon les valeurs des indicateurs. Les indicateurs sont ainsi représentés sous forme d’arbre ce qui est intuitif et facile d’utilisation. Juste en-dessous se trouve les paramètres de la carte avec notamment la vue satellite et le nom des pays qui sont affichés si les cases sont cochées. La légende en-dessous apparaît une fois un indicateur sélectionné et affiche ainsi les correspondances entre les couleurs et les valeurs des indicateurs. Lorsque l’on clique sur un pays, une fenêtre apparaît affichant la valeur des indicateurs courants. De plus, un lien y figure permettant d’ouvrir la page des graphiques pour rentrer dans les détails. La barre en haut de l’écran permet de sélectionner une date ou une période temporelle.

# Justification des choix retenus (maquette et rendu)

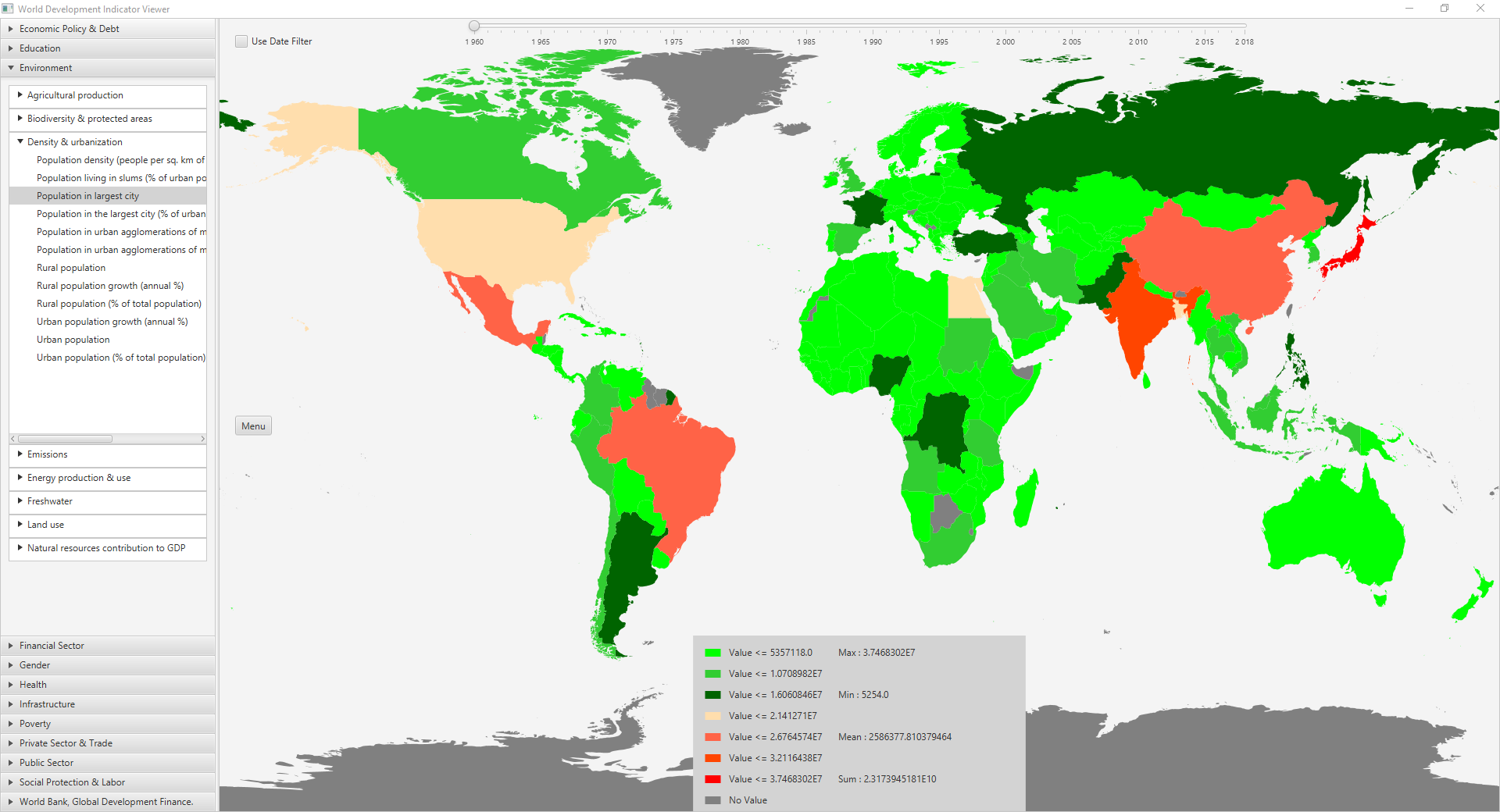
Dans cette partie du rapport, nous allons justifier les choix d’implémentation de l’application par rapport au choix d’interface retenu dans la partie II.

Tout d’abord, nous avons implémenté uniquement la page de la vue générale des différents pays avec la sélection des indicateurs. En effet, à cause de contraintes de temps, il nous a été impossible de réaliser la page qui affiche différents graphiques.

L’image ci-dessous représente la première page sur laquelle arrive l’utilisateur.



Cette interface contient le panneau de sélection des indicateurs sur la gauche de l’interface avec un affichage sous forme d’arbre avec les topics puis les sous-topics puis les indicateurs. Un slider est présent en haut de celle-ci avec une checkbox permettant d’afficher uniquement les dates d’une certaine année. Le slider avec les doubles curseurs n’étant pas présent dans la bibliothèque javafx, nous avons décidé de nous concentrer sur d’autres aspects de l’interface. Un bouton menu sur la gauche permet de fermer ou d’ouvrir le volet de sélection de l’indicateur. Lorsque l’utilisateur clique sur un indicateur, les pays sont coloriés automatiquement selon la légende qui s’affiche en bas de l’interface tel que représenté dans l’image ci-dessous.



La légende affichée indique quelles plages de valeurs correspondent à quelles couleurs ainsi que la valeur minimale, maximale, moyenne et la somme des différentes valeurs correspondantes.



Lorsque l’utilisateur sélectionne une année pour laquelle aucune valeur n’est disponible dans la base de données, un message est affiché l’invitant à en essayer une autre.

La barre de recherche ainsi que les paramètres de la carte pour avoir une vue détaillée ou les noms des pays, les boutons d’impression et d’enregistrement et le panneau d’information qui s’affiche lors du clic de l’utilisateur sur un pays sont des fonctionnalités qui ne sont pas implémentées dans la version rendue et constituent des améliorations possibles dans le futur.

# Difficultés rencontrées pendant l’implémentation

Nous avons rencontré différentes difficultés au cours de l’implémentation de l’application. La première était de comprendre la structure même des données par rapport aux classes déjà disponibles lors du début du projet. La différence entre Indicator reste l’exemple le plus marquant IndicatorData. De plus, le même problème se retrouvait dans la différence entre la classe Polygon fournie et la classe Polygon de la bibliothèque javafx permettant d’afficher les différents points sur la carte.

# Conclusion