|  |
| --- |
| Plot that lines |

Table des matières

[1 Analyse préliminaire 3](#_Toc180662353)

[1.1 Introduction 3](#_Toc180662354)

[1.2 Objectifs 3](#_Toc180662355)

[1.3 Gestion de projet 4](#_Toc180662356)

[2 Analyse / Conception 4](#_Toc180662357)

[2.1 Domaine 4](#_Toc180662358)

[2.2 Concept 4](#_Toc180662359)

[2.3 Analyse fonctionnelle 4](#_Toc180662360)

[2.4 Stratégie de test 4](#_Toc180662361)

[3 Réalisation 5](#_Toc180662362)

[3.1 Points de design spécifiques 5](#_Toc180662363)

[*3.1.1* *…* 5](#_Toc180662364)

[*3.1.2* *…* 5](#_Toc180662365)

[*3.1.3* *…* 5](#_Toc180662366)

[3.2 Déroulement 5](#_Toc180662367)

[3.3 Mise en place de l’environnement de travail 6](#_Toc180662368)

[3.4 Description des tests effectués 6](#_Toc180662369)

[3.5 Erreurs restantes 6](#_Toc180662370)

[4 Conclusions 6](#_Toc180662371)

[5 Annexes 7](#_Toc180662372)

[5.1 Journal de travail 7](#_Toc180662373)

*NOTE L’INTENTION DES UTILISATEURS DE CE CANEVAS :*  
*Toutes les parties en italiques sont là pour aider à comprendre ce qu’il faut mettre dans cette partie du document. Elles n’ont donc aucune raison d’être dans le document final.*

*De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n’aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l’alourdir inutilement.*

# Analyse préliminaire

## Introduction

Ce projet est réalisé dans le cadre du module I323 afin d’apprendre la programmation fonctionnelle en créant une application console affichant des graphiques.

Dans ce projet l’application console permet d’afficher des graphiques concernant les dépenses militaires de chaque pays dans la devise décidé par l’utilisateur afin de comparer les dépenses.

*Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.*

*Ces éléments peuvent être repris des spécifications de départ.*

## Objectifs

* Création d’une interface graphique permettant de filtrer l’affichage des données.
  + Filtre sur les dates
  + Au moins 3 courbes
  + L’axe X correspond à la temporalité

*Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.*

*Ces éléments peuvent être repris des spécifications de départ.*

## Gestion de projet

***Ce chapitre décrit la méthode de gestion de projet utilisée, ainsi que les éventuelles particularités requises par le contexte et/ou le chef de projet***

La méthode de projet utilisé est la méthode agile. Le chef de projet est Xavier Carrel.

La méthode Agile est une méthode de gestion de projet centré sur la flexibilité et la collaboration avec les membres de l’équipe et le client.

Elle fonctionne avec des cycles de développement court appelés « sprints ».

# Analyse / Conception

## Domaine

Les données récoltées sont les dépenses militaires de plusieurs pays selon leurs devises locales.

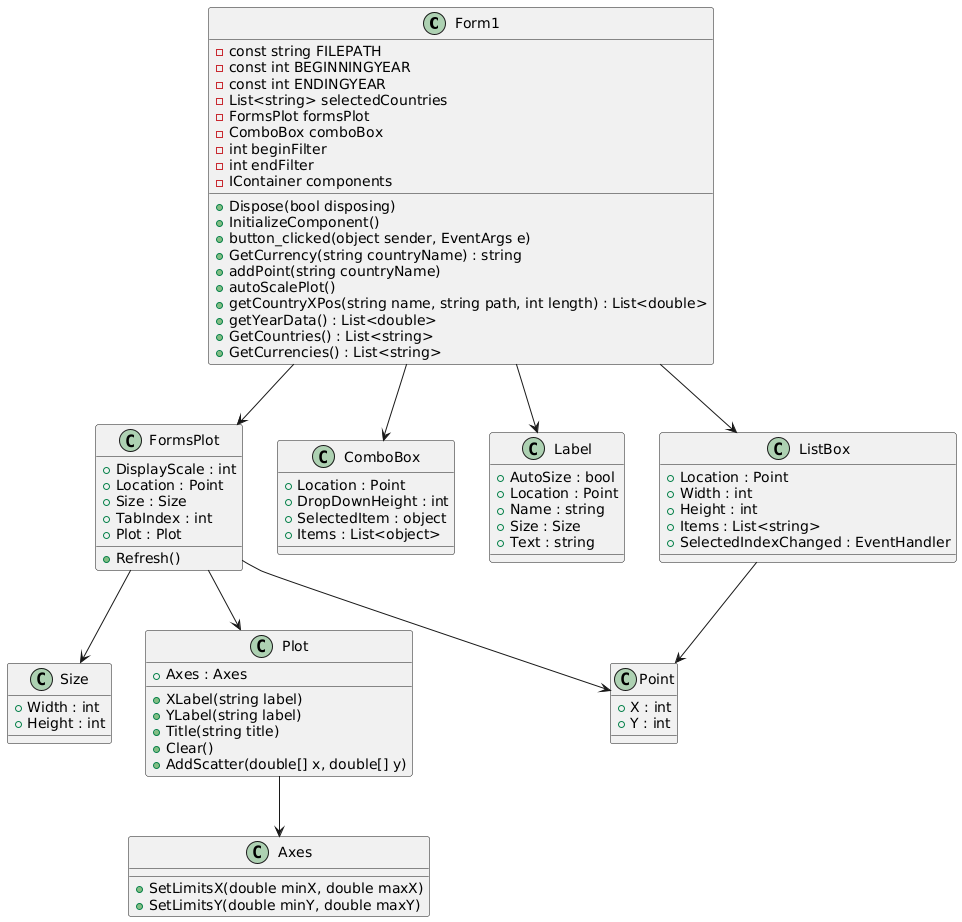
Le domaine des données est militaire et économique.

L’échelle de temps utilisé est une année.

Le public destiné contient des économistes mais aussi n’importe quelle personne curieuse de connaître les dépenses militaires d’un pays en particulier et de les comparer à d’autres pays.

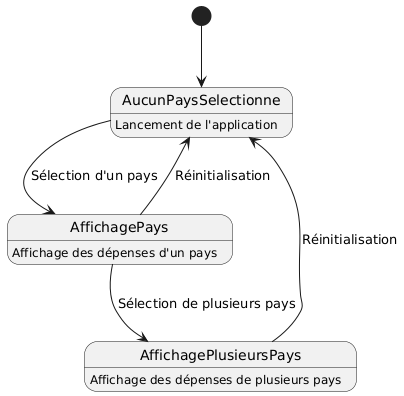
* *Quel est le domaine auquel appartiennent vos données ?*
* *Que représentent-elles ?*
* *Quelles échelles de temps sont pertinentes pour ces données ?*
* *A quel public sont-elles destinées ?*

## Concept



TODO : update

1 - Diagramme de classe



TODO : update

2 - Diagramme d'état

## Analyse fonctionnelle

***Reprendre le contenu des User Stories d’IceScrum : Story + tests d’acceptance (avec IceTools) + maquettes***

## Stratégie de test

Des tests unitaires sont effectués avec MSTest, ils vérifient que les méthodes utilisées renvoient des valeurs correctes. Les méthodes faisant appel à une API ne sont pas testées ainsi que celles ne retournant aucune valeur.

**Ordre des tests :**

1. Tests unitaires :

Les tests unitaires vérifieront que les méthodes fonctionnent indépendamment du reste.

1. Tests d’intégration :

Les tests d’intégration sont effectués pour vérifier que l’appel à l’API fonctionne correctement.

1. Tests de validation :

Finalement les tests de validation seront effectués pour vérifier que le programme correspond bien aux exigences qui ont été spécifié dans les User Story.

**Moyen à mettre en œuvre :**

Des données de test pertinentes doivent être préparées.

Par exemple des noms de pays, des dates ou un fichier comportant des données.

Pour mettre en place ces données il faudra modifier les valeurs des constantes dans le code ou mettre en place un fichier .env.

Les constantes actuelles sont :  
FILEPATH qui contient l’emplacement du fichier contenant les données.

BEGINNINGYEAR qui contient la date de départ des données récoltés.

ENDINGYEAR qui contient la date de fin des données récoltés.

Afin de réduire le temps et l’effort nécessaire il serait préférable d’automatiser les tests dans un outil de CI/CD.

**Testeurs :**

Les testeurs éventuels de l’application seraient des utilisateurs finaux ou des camarades de classe, donc une communauté de développeur.

**Sprint Review :**

La sprint review peut se dérouler en présentiel comme en visioconférence.

Elle concernera toutes les personnes impliquées dans ce projet.

# Réalisation

## Points de design spécifiques

### Gestion des erreurs lors de l’appel à l’API

### Conversion des devises

### Validation des données d’entrées

### Gestion des appels asynchrones

### Utilisation de secret pour la clef API

### Structure des classes et des méthodes

### Implémentation des tests unitaires

### Design de l’interface utilisateur

### Traitement des devises non supportés

### Création des commentaires

***Ce chapitre est constitué de plusieurs sous-chapitre.***

***Chaque sous-chapitre explique un point de design technique particulier, quelque chose que vous avez dû inventer pour répondre au besoin et qui ne peut pas s’expliquer par de simples commentaires dans le code.***

***Il s’agit d’explications techniques sur le fonctionnement du système. Les explications sont appuyées par des diagrammes, ou de très brefs éléments de code.***

***NE PAS mettre ici des pratiques usuelles que tout professionnel de la branche connaît déjà. Par exemple, n’EXPLIQUEZ PAS ICI CE QU’EST LE PATTERN MVC.***

***Exemple (simplifié à l’extrême) : Protection contre des formulaires mal intentionnés ou modifiés***

* ***Au moment de générer le formulaire, le script php :***
  + ***Concatène les noms de tous les champs contenus dans le formulaire***
  + ***Calcule un hash SHA256 de la chaîne obtenue***
  + ***Ajoute un input nommé « CSRF » de type hidden dans le form***
* ***A la réception du POST du fromulaire***
  + ***Concatène les noms des indices de $\_POST***
  + ***Calcule un hash SHA256 de la chaîne obtenue***
  + ***Vérifie que la valeur du champ CSRF correspond***

### *…*

### *…*

### *…*

## Déroulement

***Résumer comment s’est passé la réalisation de chaque story, ses difficultés, les alternatives envisagées mais rejetées, ses surprises, …***

## Mise en place de l’environnement de travail

* ***Comment accéder au code source***
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*

***Ce chapitre décrit précisément comment un employé qualifié peut recréer l’environnement dans lequel vous avez effectué ce travail***

## Description des tests effectués

***Reprendre les tests d’acceptance d’IceScrum au moyen de la feuille ad hoc d’IceTools***

## Erreurs restantes

*S'il reste encore des erreurs :*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

***Reporter la*** [***dette technique***](https://www.premaccess.com/qu-est-ce-que-dette-technique-comment-la-maitriser/#:~:text=La%20dette%20technique%20survient%20quand,de%20plus%20en%20plus%20fr%C3%A9quents.) ***connue. S’appuyer sur la pratique des // TODO***

# Conclusions

*Développez en tous cas les points suivants :*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## Journal de travail