|  |
| --- |
| Plot that line |

Pour projet Plot That Line avec XCL

Table des matières

[1 Analyse préliminaire 3](#_Toc175917886)

[1.1 Introduction 3](#_Toc175917887)

[1.2 Objectifs 3](#_Toc175917888)

[1.3 Gestion de projet 4](#_Toc175917889)

[2 Analyse / Conception 4](#_Toc175917890)

[2.1 Gameplay 4](#_Toc175917891)

[2.2 Concept 4](#_Toc175917892)

[2.3 Analyse fonctionnelle 4](#_Toc175917893)

[2.4 Stratégie de test 4](#_Toc175917894)

[3 Réalisation 4](#_Toc175917895)

[3.1 Points de design spécifiques 4](#_Toc175917896)

[*3.1.1* *…* 5](#_Toc175917897)

[*3.1.2* *…* 5](#_Toc175917898)

[*3.1.3* *…* 5](#_Toc175917899)

[3.2 Déroulement 5](#_Toc175917900)

[3.3 Mise en place de l’environnement de travail 5](#_Toc175917901)

[3.4 Description des tests effectués 6](#_Toc175917902)

[3.5 Erreurs restantes 6](#_Toc175917903)

[4 Conclusions 6](#_Toc175917904)

[5 Annexes 7](#_Toc175917905)

[5.1 Manuel de référence 7](#_Toc175917906)

[5.2 Journal de travail 7](#_Toc175917907)

*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

Le projet a été réalisé dans le cadre du module Programmation Fonctionnel I323. Le but est de faire afficher un graphique sur un jeux de données libre. Pour cela les technologies choisies sont libres. Il a seulement été imposé de faire le tout en c#. J’ai personnellement choisi WPF pour l’affichage et Scott Plot pour les graphiques.

## Objectifs

Les objectifs du projet sont :

* Afficher minimum 3 courbes
* Filtrer les données selon les dates

PS : les objectifs en vert = réussi

## Gestion de projet

La méthode de gestion choisit est agile avec le logiciel IceScrum. Il faut faire des User Stories et des tests d’acceptance pour chacune des fonctionnalités.

# Analyse / Conception

## Domaine

Le domaine de mon jeu de données sont les échecs. Pour cela le graphique va afficher sur axe x le temps en année et sur l’axe y l’Elo (unité utilisée pour évaluer le niveau d’un joueur). La courbe représentant un joueur. Le but est de pouvoir comparer le niveau de plusieurs joueurs d’échec en regardant leur évolution.

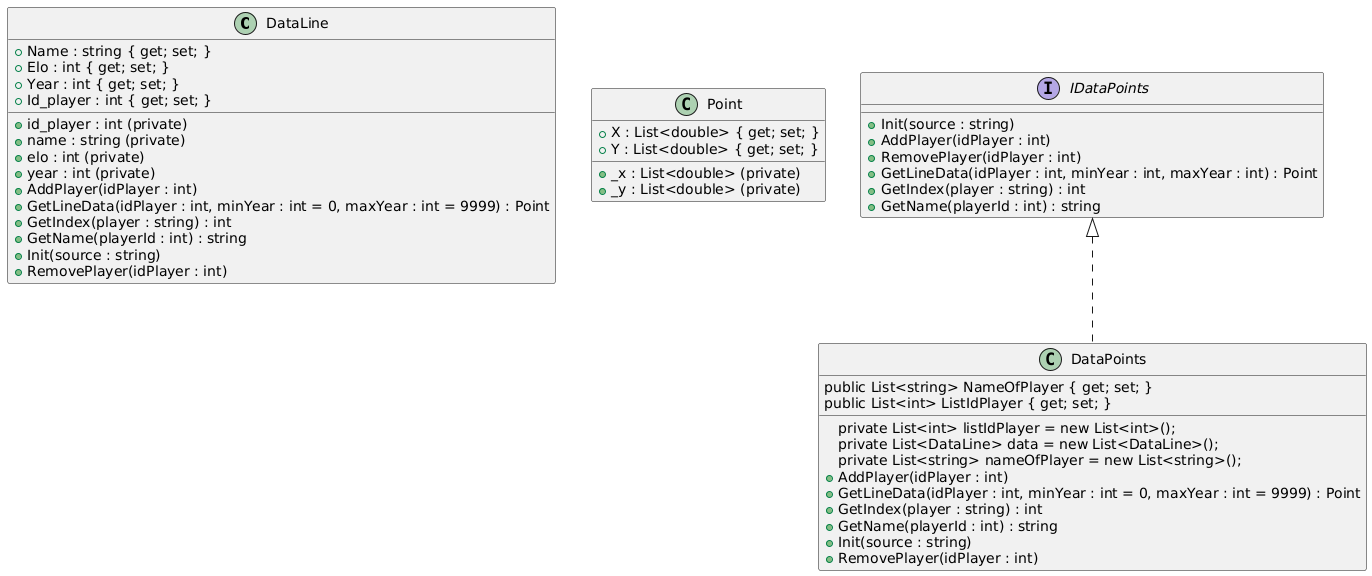
C’est l’utilisateur qui doit fournir le csv. Celui-ci doit respecter plusieurs critères :

* Contenir les **colonnes** suivantes
  + **ELO**
  + **Name**
  + **Date** (en année)
* Chaque élément est séparé par un « ; »
* L’élément ne doive pas être entourée par des guillemets double ou simple
* Il ne doit pas contenir dans une valeur le « ; »

Cette application peut être utilisé par des personnes qui veulent voir l’évolution du niveau de joueurs d’échec (à titre privé ou public). Le but est que l’application soit accessible.

## Concept

* *Diagramme(s) d’état*

**

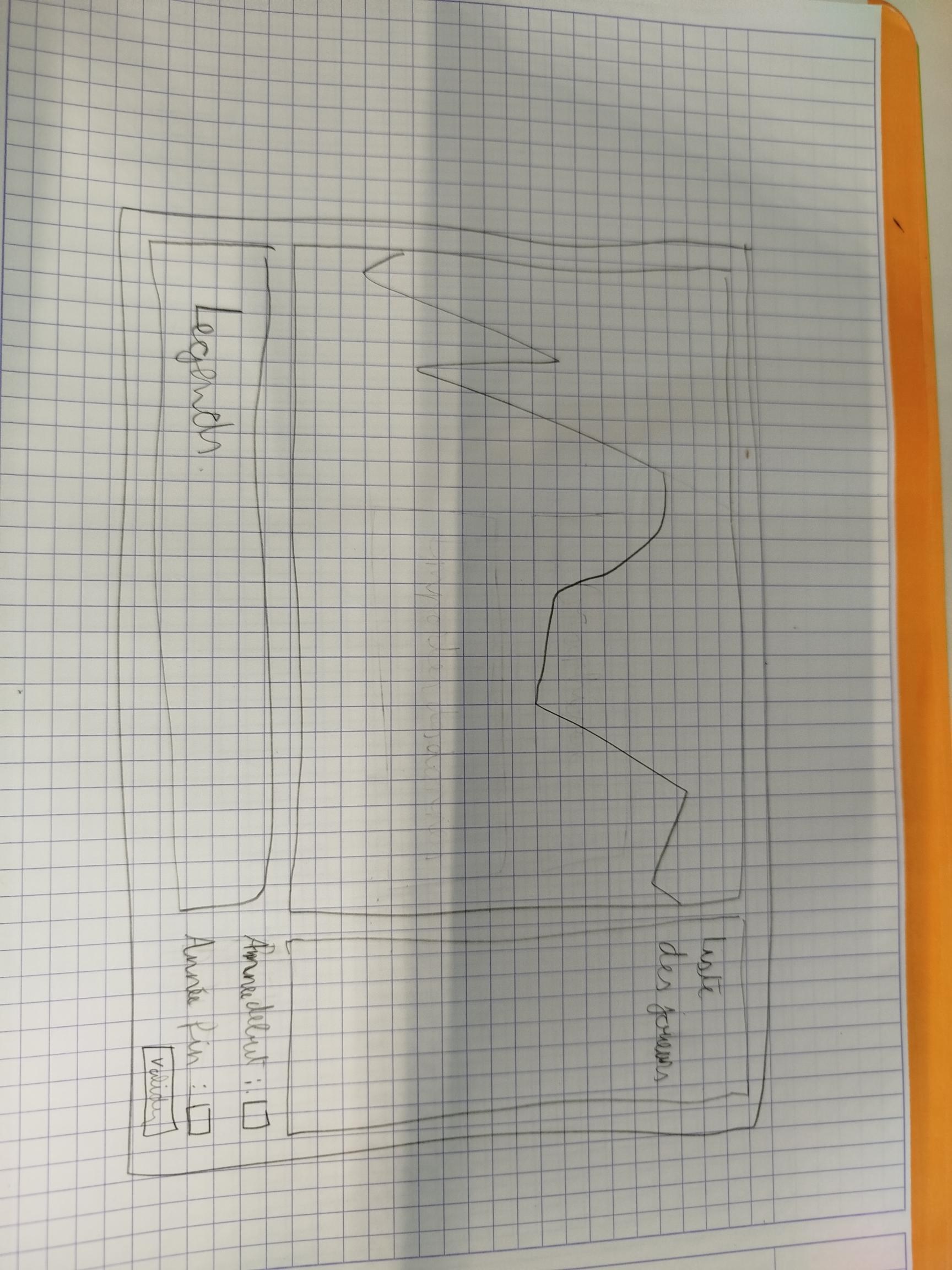
## Analyse fonctionnelle

### Affichage du graphique

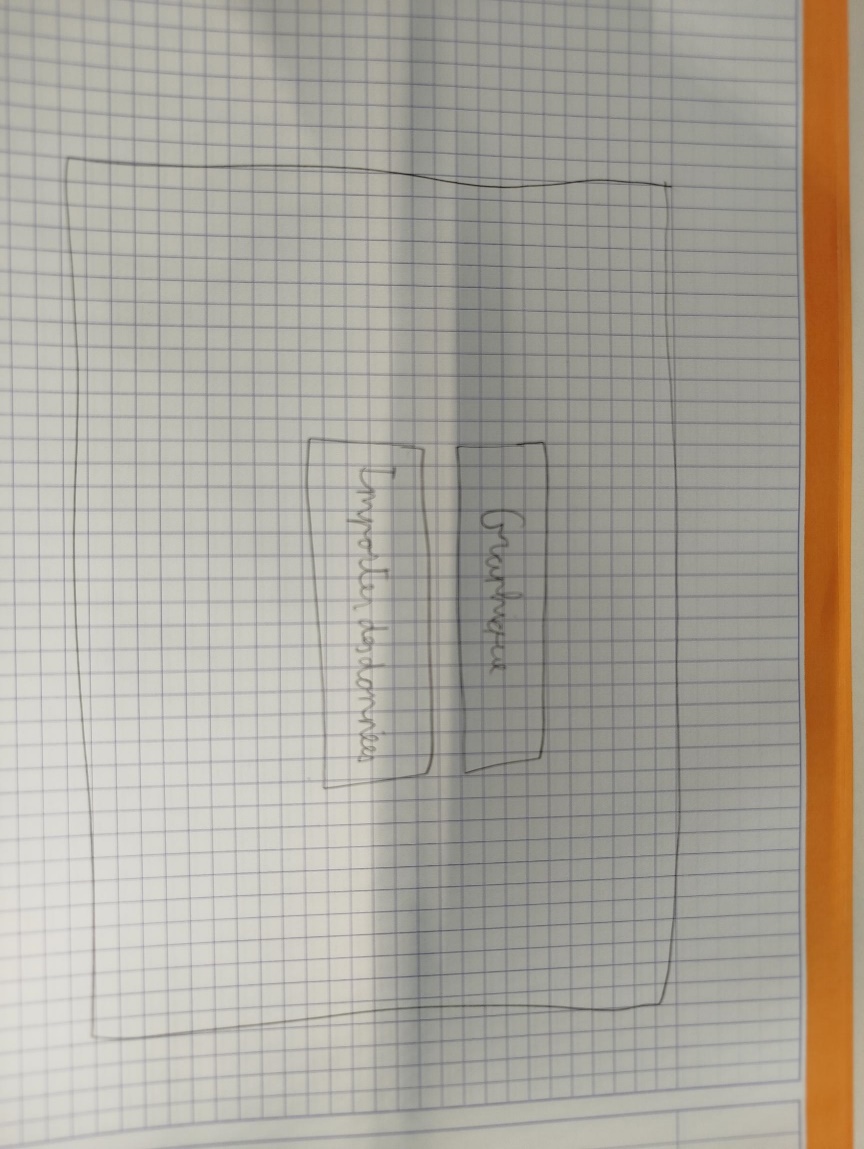
(Auteur : Tiago Rodrigues Sousa)

|  |
| --- |
| En tant que passionné d'échec,  Je veux pouvoir afficher un graphique sur l'Elo des joueurs au fil du temps en utilisant un jeu de données.  Afin de pouvoir voir l'évolution du niveau des échec selon le jeu de données. |
| Tests d’acceptance :   |  |  | | --- | --- | | Choisir un graphique | Sur mon bureau, lorsque je lance l'application une fenêtre apparait avec 2 boutons dont les contenus sont : - Graphique - Importer des données (regarder maquette menu) | | Choisir Graphique | Dans le menu, lorsque je clique sur Graphique, une nouvelle fenêtre apparaît avec un graphique dans lequel on voit des courbes d'Elo des joueurs (regarder maquette graphique) | | Appuyer sur bouton import | Dans le menu, lorsque j'appuie sur le bouton importer des données, une nouvelle fenêtre d’explorateur de fichier apparait | | Importer des données juste | Dans l'explorateur de fichier, lorsque je sélectionne un fichier csv qui est correctement formaté un message avec "importation réussi" apparaît | | Importer données fausse | Dans l'explorateur de fichier, lorsque je sélectionne un fichier csv qui n'est pas correctement formaté alors un message d'erreur avec "l'importation a échoué" apparait | |

Maquette graphique :



Maquette menu :



### Filtrer le graphique

(Auteur : Tiago Rodrigues Sousa)

|  |
| --- |
| En tant que joueur d'échec,  Je veux pouvoir choisir les courbes selon les critères suivants : - joueur que je veux afficher - la date  Afin de pouvoir seulement voir que les informations qui m'intéresse |
| Tests d’acceptance :   |  |  | | --- | --- | | Sélectionner un joueur | Dans la page du graphique, Lorsque j'appuie sur une case vide à côté d'un joueur, Alors la case se rempli et une courbe apparaît sur le graphique | | Désélectionner un joueur | Dans la page du graphique, Lorsque j'appuie sur une case remplie à côté d'un joueur, Alors la case se vide et une courbe disparaît sur le graphique | | Choisir une date (début) | Dans la page du graphique, Lorsque je mets l'année 2005 dans la case "date début", l'affichage des données ne commence que à partir de 2005 | | Choisir une date (fin) | Dans la page du graphique, Lorsque je mets l'année 2020 dans la case "date fin", l'affichage des données se termine en 2020 | | Choisir fausse date | Dans la page du graphique, Lorsque je mets l'année 2020 dans la case "date début" et lorsque je mets l'année 2005 dans la case "date fin", cela affiche une nouvelle page avec une erreur | | Aucun joueur | Dans la page du graphique, Lorsque aucun des joueurs n'est sélectionné alors le graphique est vide | |

## Stratégie de test

Des tests manuels seront effectués en se basant sur les tests d’acceptances. Le but étant de reproduire ce qui est prévue par l’analyse fonctionnel. Le jeu de données utilisé est celui par défaut (Chess.csv).

Des tests automatisés seront mis en place pour tester les fonctions et faire en que du nouveau code ne casse pas l’ancien.

Des testeurs extérieurs utiliseront l’application avec leur propre jeu de données personnels. Le but est de voir si l’application gère correctement d’autre jeu de données et de voir si l’application est utilisable (UI/UX).

# Réalisation

## Points de design spécifiques

***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

### Affichage du graphique

J’ai commencé par faire en forms sans librairie sauf que c’était assez compliqué. Donc j’ai décidé de passer sur du WPF avec la librairie ScottPlot. La suite c’est bien déroulé jusqu’au moment de d’importer le csv de l’utilisateur. J’ai laissé une dette technique concernant la gestion des erreurs sur le fichier csv. J’ai été agréablement surpris de la librairie ScottPlot qui était assez complète.

### Filtrer le graphique

Il y a eu quelque questionnement sur la manière d’afficher la liste des joueurs, mais finalement cela a vite été réglé en utilisant des Checkbox. Au niveau du filtrage par date, cela c’est bien passé.

## 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***

**GitHub repos :** <https://github.com/TiaSous/Plot-That-Line>

**Visual Studio 2022:** Visual Studio Community 2022 (64 bits) - Version 17.9.2

**Data/Chess.csv :** Jeu de données utilisée pour tester l’application

**Fichier projet :**

* DataLine.cs : Classe représentant une ligne de données
* DataPoints.cs : Classe représentant le fichier csv traité
* Point.cs : Classe représentant un point sur le graphique
* GraphicPage.xaml ou .cs : Représente le view du graphique
* Menu.xaml ou .cs : Représente le menu (visuel)

## 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