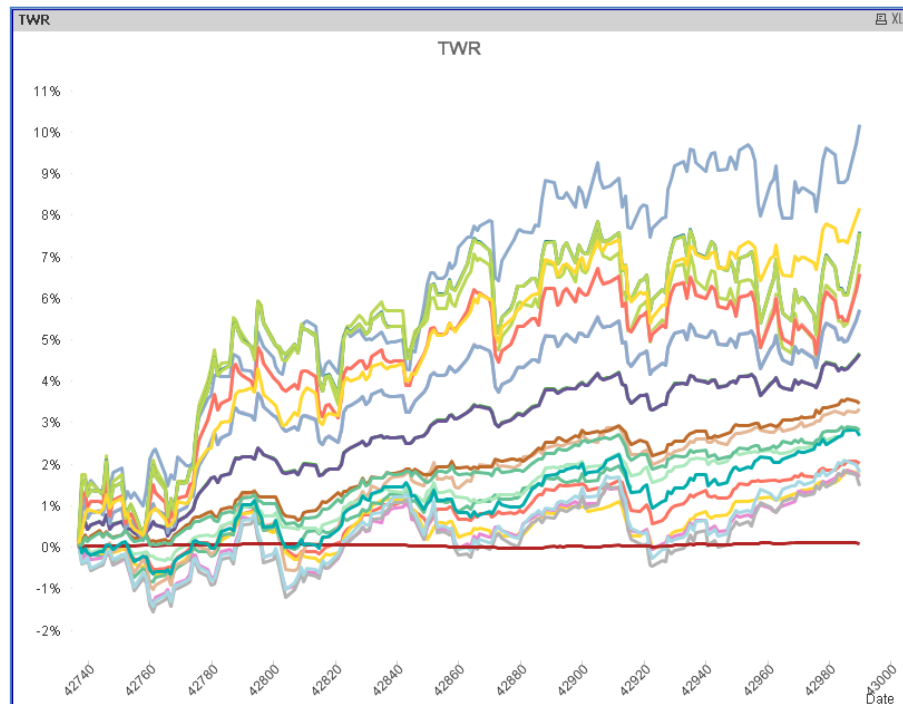


Plot Those Lines



Samuel Sallaku – CID3B
Section Informatique - Vennes
32p
Xavier Carrel

Table des matières

1	SPÉCIFICATIONS.....	3
1.1	DESCRIPTION.....	3
1.2	OBJECTIFS PRODUIT.....	3
1.3	OBJECTIFS PÉDAGOGIQUES	4
1.4	DOMAINE D'APPLICATION	4
2	PLANIFICATION	5
3	ANALYSE FONCTIONNELLE/USER STORIES.....	5
4	MAQUETTES.....	7
5	RAPPORT DE TESTS	9
5.1	TABLEAU DE TESTS	9
6	SCHEMA.....	10
7	JOURNAL DE TRAVAIL.....	11
8	USAGE DE L'IA	11
9	CONCLUSION.....	12
10	SOURCES	13

1 SPÉCIFICATIONS

1.1 Description

Projet en C# ayant pour but de créer un programme qui affiche un graphique avec une importation de données et qui permet d'en ajouter tout en gardant les anciennes données

Plus précisément, il va afficher des jeux vidéo, ainsi que le nombre de ventes pour chaque année depuis sa sortie au public.

Il faut des fichiers CSV avec un format comme : Game, Year, Sales

S'il ne trouve pas cela, il va afficher une erreur.

On peut regarder les différentes années et zoomer/dézoomer dessus.

1.2 Objectifs produit

Le but du programme est de permettre à l'utilisateur de voir et d'analyser l'évolution des ventes de jeux vidéo dans les différentes années.

Les fonctionnalités principales sont :

- Importer des données depuis un fichier CSV (colonnes Game, Year, Sales).
- Afficher ces données sous forme de graphique dynamique à l'aide de ScottPlot.
- Ajouter de nouvelles données sans perdre les anciennes informations.
- Naviguer entre différentes années et effectuer des zooms sur le graphique.
- Grande flexibilité d'affichage donc, possibilité de montrer plusieurs jeux vidéo ou mettre celles qui nous intéressent.

Donc, les objectifs produit visent à :

- Fournir un outil/programme qui est simple et intuitif pour comprendre les données des ventes de jeux vidéo
- Permettre d'analyser rapidement et visuellement les tendances ou les peaks des ventes selon les années
- Permettre de lire les données facilement grâce au graphique

Public visé :

- Des gamers, étudiants, apprentis en informatique qui veulent apprendre à lier la manipulation de données, c# et la visualisation graphique
- Des passionnés de jeux vidéo (gamers) ou des analystes de marché, qui désirent explorer les ventes des jeux vidéo à travers les années

L'usage typique de l'application consiste à importer un fichier CSV contenant les ventes de jeux, puis à visualiser les résultats sous forme de graphiques interactifs. L'utilisateur peut aussi comparer les ventes entre plusieurs titres, observer les tendances et extraire des conclusions sur l'évolution du marché.

1.3 Objectifs pédagogiques

Ce projet a pour objectif pédagogique de renforcer mes compétences en développement C# et d'explorer de nouvelles technologies liées à la visualisation de données comme ScottPlot

Il m'a permis de :

- Découvrir et utiliser **WinForms** pour créer une interface graphique.
- Manipuler et filtrer des données à l'aide de **LINQ**, avec des Select, Where, etc.
- Apprendre à exploiter des bibliothèques externes comme **ScottPlot** pour la création de graphiques.
- Approfondir la compréhension de la gestion de fichiers CSV et de la validation de données.
- Mieux comprendre les **User Stories** et leur utilité pour la création des fonctionnalités à implémenter dans le programme

Ce projet permet également d'approfondir les compétences en LINQ et en programmation fonctionnelle (module en parallèle), meilleure manipulation de fichiers CSV pour le filtrage des données et la création d'interfaces utilisateurs avec WinForms, ce qui est une nouvelle librairie graphique pour moi.

Par exemple, certaines bonnes pratiques dont je n'avais pas eu le réflexe d'implémenter, comme l'instanciation d'une classe dans un fichier appart. J'avais créé la classe GameData avec tout le reste du code.

Il permet de remettre en pratique les principes du POO, ce que j'ai appris en 2^{ème}. Cela m'a permis d'approfondir les connaissances du POO comme l'encapsulation, héritage ou bien polymorphisme.

De plus, un objectif était d'ajouter plus de commentaires pour comprendre certains aspects compliqués à s'en rappeler dans le code, notamment la lecture d'un fichier CSV depuis le programme. Meilleure gestion d'exceptions aussi, comme le try catch qui se trouve dans mon programme pour la lecture des fichiers CSV.

1.4 Domaine d'application

Le thème que j'ai choisi est les achats des jeux à travers les années, que ces jeux soient en ligne ou hors ligne.

Le but derrière est de pouvoir visualiser comment les années et différentes technologies ont impacté l'achat des différents types de jeux à travers les années.

Cela va donc permettre de comparer les différents genres, les différents éditeurs ou bien développeurs et les différentes régions du monde

Je cherche également à mettre le nombre de joueurs à travers les années si le temps le permet. (Optionnel)

Plus précisément :

- Les achats d'un certain jeu
- Différentes années d'achat

- Liste de nom et d'années à mettre sur le graphique
- Le nombre de joueurs (Optionnel)

Les moyens de récupérer ces données seront les suivantes :

- <https://www.kaggle.com/>
- https://developer.valvesoftware.com/wiki/Steam_Web_API
- ChatGPT

2 PLANIFICATION

Les tâches principales sont sur GitHub Projects, [lien ici](#) .

Une tâche principale sera créée et elle aura une ou des User Story pour l'expliquer.

Voici l'ordre dont je pense faire les tâches :

- Création du repo, tâches, User Stories
- Recherches sur où je peux trouver des données plus au moins réelles.
- Maquettes principales
- Page Principale, donc là où on retrouve le graphique
- Endroit et bouton pour l'importation
- Possibilité d'importer des fichiers CSV etc.
- Flexibilité d'affichage
- Mode fonctions

Tout au long du projet, il y aura des tâches qui seront réalisées tous les jours comme le remplissage du journal de travail (gitjournal) et le remplissage du rapport.

3 ANALYSE FONCTIONNELLE/USER STORIES

Séries temporelles - [lien](#)

En tant qu'utilisateur,

Je veux afficher une représentation graphique de plusieurs séries temporelles (time serie) simultanément,

Afin de comparer et analyser l'évolution des données que j'importe dans le temps.

Test d'acceptance

- Dès qu'on lance le programme, on arrive sur l'écran principal (voir maquette 1)
- Quand je clique sur une des cases à cocher, la série temporelle correspondante s'affiche, et si je reclique, elle disparaît (voir maquette 1)

- Quand je clic sur une des cases à cocher sous la catégorie "Année", l'année correspondante s'affiche ou disparaît pour toutes les séries(voir maquette 3)
- Si plusieurs cases sont cochées, alors il y aura plusieurs années affichées(celles qui sont cochées)
- Chaque série temporelle a une couleur ou style dinstinct (voir maquette 1)

Importation des données - [lien](#)

En tant qu'utilisateur,

Je veux être capable d'importer mes propres données

Afin d'ajouter les données que je veux au graphique

Test d'acceptance

- Quand je clique sur "Importer des données", Il y a une fenêtre qui s'ouvre, elle me donne le choix du fichier que je veux importer
- Je peux choisir un fichier (CSV ou autre)
- Après la validation, il y a une nouvelle série dans la liste de gauche (voir maquette 2)
- Si l'import a échoué, un message d'erreur clair s'affiche indiquant l'erreur et on revient sur la maquette 1
- Si les données existent déjà, un message s'affiche indiquant qu'elles seront mises à jour

Timeline – [lien](#)

En tant qu'utilisateur,

Je veux une timeline(axe temporel) sur le graphique

Afin de comparer facilement les données entre plusieurs années d'un certain sujet

Test d'acceptance

- Quand j'insère des données, alors il y a un graphique qui s'affiche avec une timeline claire et lisible
- Timeline est par année (repères clairs sur le temp, voir maquette 1)
- Plusieurs années sont affichées au même temps, quand je fais des clics sur les cases correspondantes des données (voir maquettes 1, 2 & 3)
- Il y a une flexibilité lors de l'affichage du timeline, donc les années ne se suivent pas forcément. (ex: peut aller de 2004 à 2010)

Flexibilité d'affichage - [lien](#)

En tant qu'utilisateur,

Je veux une grande flexibilité d'affichage sur le graphique,

Afin de pouvoir changer l'affichage à mes préférences et analyser les données plus efficacement

Test d'acceptance

- Quand je zoom/dézoom sur le graphique, il y a des changements de la taille et l'échelle du graphique
- Possibilité de changer la taille ou zone d'affichage en agrandissant, réduisant la fenêtre
- Zoomer/Dézoomer pour mieux analyser les données, en faisant zoom/dezoom avec la molette
- Ajouter plusieurs time series simultanément et tous les voir sur le même graphique (voir maquette 1)

Mode fonctions - [lien](#)

En tant qu'utilisateur,

Je veux avoir un mode d'affichage des fonctions plutôt que des séries temporelles (onglet, option, ...)

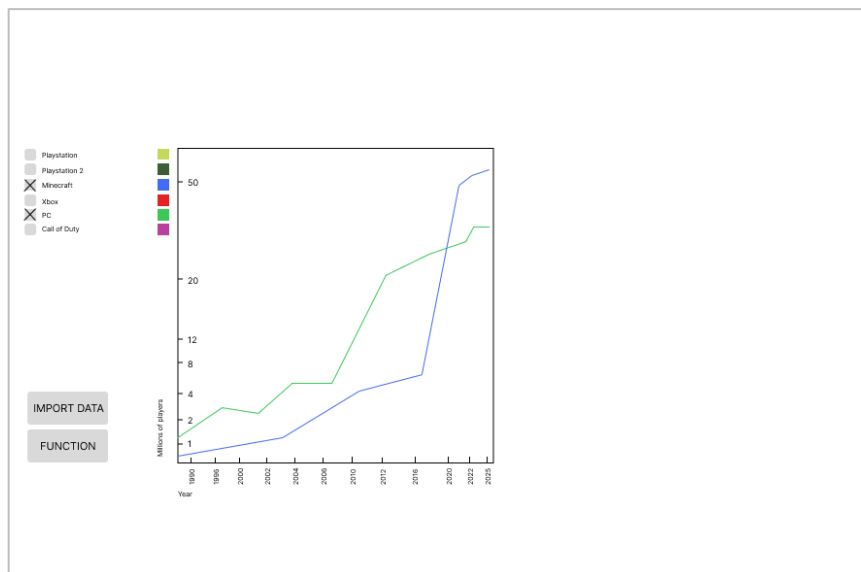
Afin de pouvoir changer l'affichage

Test d'acceptance

- Choix d'utiliser des fonctions au lieu des séries temporelles
- Affichage de fonctions différentes ($2x$, $\sin(x)$, etc..)
- Choix de quelles fonctions sont utilisées sur le graphique (une ou plusieurs au même temps)

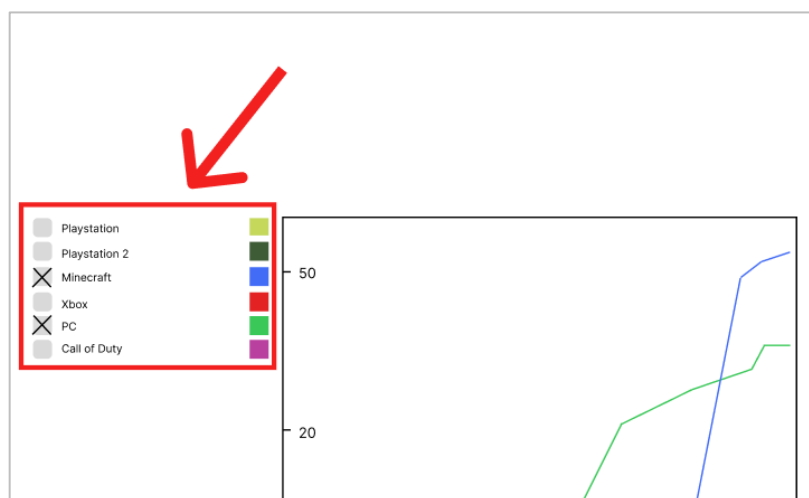
4 MAQUETTES

Maquette principale



Ici on montre la page principale, ce que l'utilisateur verra dès le lancement du programme (sans les données, ce sont juste un exemple)

Maquette 2



Pour cette maquette, on cherche à montrer l'option de changer les différentes séries temporelles donc, la possibilité d'ajouter ou d'enlever des jeux et leurs achats par année du graphique

Maquette 3



Ici le but est de montrer qu'il y a la possibilité d'ajouter ou bien supprimer des années sur l'axe X

5 RAPPORT DE TESTS

5.1 Tableau de tests

Voici des tests de fonctionnalités du programme qui sont basés sur les tests d'acceptance faites avec les User Stories.

On nomme la fonctionnalité puis on la décrit comment elle devrait se dérouler.

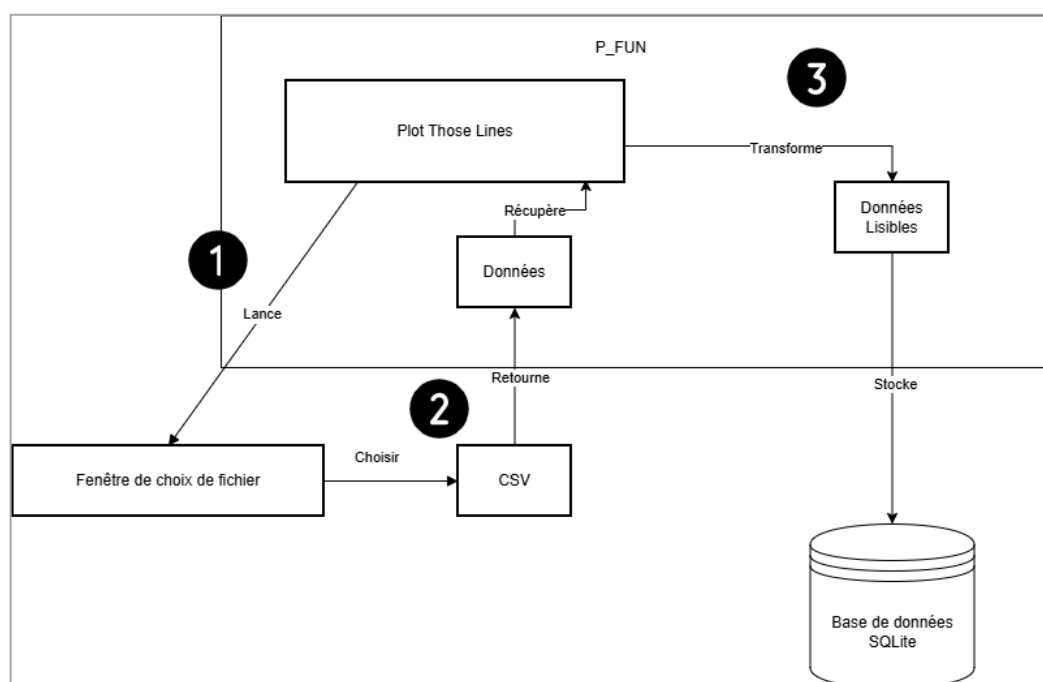
Si le programme fait ce que la description dit, alors on met OK. Sinon KO.

Fonctionnalité	Description du test	Résultat
Zoom/Dézoom	Zoomer/Dézoomer pour mieux analyser les données, en faisant zoom/dezoom avec la molette	OK
Changement de taille de fenêtre	Changer la taille ou zone d'affichage en agrandissant, réduisant la fenêtre	OK
Changer les années sur l'axe x	Ajouter plusieurs time series simultanément et tous les voir sur le même graphique	OK
Importation de données	"Importer des données" va ouvrir une fenêtre pour choisir un fichier et quand le fichier est choisi, les données sont importées	OK
Choix de fichier	Je peux choisir un fichier	OK
Personnalisation du graphique	Quand on importe des données, la liste des années et les jeux permet de sélectionner les données voulues sur le graphique	OK
Message d'erreur lors de l'importation	Si on importe un fichier CSV qui ne correspond pas aux	OK

	colonnes « Name, Year, Sales » il y a une erreur qui s'affiche	
Fusionner les données	Si on importe un nouveau fichier il garde les anciennes données et ajoute les nouvelles données sans doublons	OK
Insertion des données	Quand j'insère des données, alors il y a un graphique qui s'affiche avec une timeline claire et lisible	OK
Persistence des données	Si j'importe un fichier, et que je ferme le programme mais je relance plus tard, il y a toujours les mêmes données	OK
Suppression du fichier games.db	Si on supprime le fichier de DB et qu'on relance le programme et qu'on ajoute des nouvelles données, le fichier se recrée et nous trouvons toutes les années	KO Le fichier se recrée mais, quand on importe un fichier il faut relancer l'app pour que les données apparaissent
Mode fonctions	Choix d'utiliser des fonctions au lieu des séries temporelles	KO Pas eu le temps d'implémenter cette fonctionnalité à mon programme

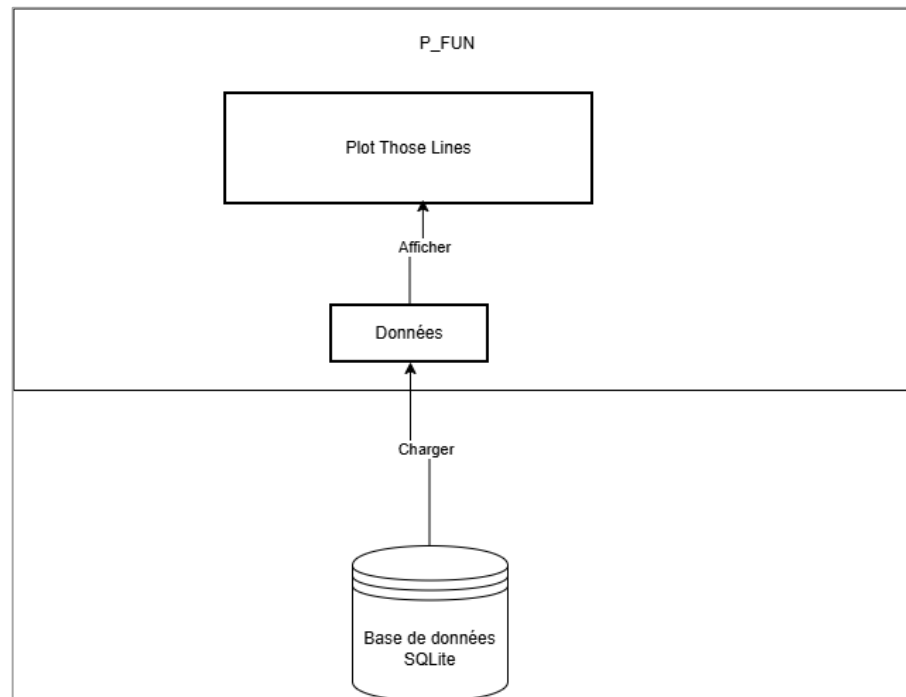
6 SCHEMA

Cette partie consiste à expliquer par des schémas le fonctionnement interne du programme.



Sur ce schéma, on montre le fonctionnement principal du programme, donc dès que nous ouvrons le programme. Si l'utilisateur clique sur « Import CSV », alors une fenêtre s'ouvre et l'utilisateur doit fournir un fichier CSV avec les spécifications qui se trouvent au chapitre 1.1 et 1.2. Dès qu'un fichier CSV valide est choisi, le programme va récupérer les données et les transformer en données lisibles, qui seront affichées sur le programme.

Puis finalement, ces données sont automatiquement stockées dans une base de données SQLite.



Sur ce schéma, on veut simplement expliquer ce qu'il se passe quand on lance le programme après avoir déjà importé un fichier CSV valide. Le programme va lire les données de la DB et va les afficher sur le programme.

7 JOURNAL DE TRAVAIL

Gitjournal - <https://github.com/samumuku/gitjournal>

Pour y accéder, il faut avoir les accès repo en générant un token(classic) et l'ajouter dans le fichier .config.js

Il faut également préciser le repo, étant https://github.com/samumuku/P_FUN-SamuelSallaku

Il existe un pdf du journal de travail complet dans la dernière release du projet.

8 USAGE DE L'IA

J'ai en effet utilisé l'IA pour ce projet pour de raisons différentes, étant :

- Documentation
 - Certains aspects de la documentation comme les objectifs produit/pédagogiques et le domaine d'application au début. Je n'avais aucune idée de quelles données utiliser pour ce projet mais vu que j'aime bien les jeux vidéo et je suis de temps en temps les actualités par rapport aux jeux, j'ai demandé à l'IA pour plus d'infos et comment je pouvais utiliser ces données pour créer un graphique avec des données.

L'IA a pu m'expliquer et a également pu me créer des données fictives à utiliser sur mon graphique.
- Code
 - Pour le code, l'usage de l'IA m'a permis de compléter des parties de code dont j'avais de la difficulté à trouver par moi-même en ligne. Notamment la partie où j'ai perdu un Joker, dont :

```
int nameIndex = Array.FindIndex(headers, h => h.Equals("Game", StringComparison.OrdinalIgnoreCase));
```
 - ```
int yearIndex = Array.FindIndex(headers, h => h.Equals("Year", StringComparison.OrdinalIgnoreCase));
```
  - ```
int salesIndex = Array.FindIndex(headers, h => h.Equals("Sales", StringComparison.OrdinalIgnoreCase));
```
 - Mais cela m'a appris que mon programme a quelque chose de plus avantageux, c'est qu'il peut trouver les colonnes mêmes si elles sont mélangées et va les mettre dans l'ordre.
 - De plus, l'IA m'a permis de mieux comprendre du code que j'ai dû chercher ou bien m'inspirer en ligne comme par exemple les docs de la DB Sqlite. Certaines choses n'étaient pas claires et donc l'IA m'a permis de mieux comprendre le fonctionnement.
- Meilleure compréhension du code et rappels
 - Comme précise sur le point d'avant, l'IA a pu me faire comprendre le fonctionnement de certaines parties pas clair de Sqlite mais, certaines compétences acquises dans d'autres modules de 2ème année comme le module POO, j'ai dû demander et me rappeler de leur fonctionnement.

9 CONCLUSION

Bilan technique

En général, le projet a été bien terminé et il a 90% des fonctionnalités prévues depuis le début, lors de la création des User Stories et la planification des tâches. La seule fonctionnalité est le mode fonctions. Vu le temps que nous avons eu pour ce projet, je n'ai pas eu le temps de terminer la partie Fonctions. S'il y avait plus de temps je pense ce serait intéressant à essayer d'implémenter cette fonctionnalité à mon programme, pour voir comment les courbes elles changeront.

Néanmoins, toutes les fonctionnalités ont été faites et terminés dans le temps imparti, le programme n'a pas de bugs ou bien jusqu'à présent, pas connues. Les fonctionnalités ont été testées une par une et elles ont passé le test, sauf bien sûr pour les Fonctions.

Les conventions de nommage et de codage ont été respectées, il y a des commentaires partout dans le code, qui aide à mieux comprendre le code, surtout les blocs de code qui peuvent paraître compliqué.

LinQ a été utilisé le plus possible sauf quelques endroits où je n'avais pas le choix mais d'utiliser des boucles foreach. Il existe également une fonction Func, qui complète un peu plus la mise en pratique des compétences acquises au module 323, Programmation fonctionnelle.

De plus, quelques librairies ont été ajoutées et installées pour ne pas devoir coder des choses compliquées dont je ne maîtrise pas. Une librairie pour le stockage persistant des données a été ajoutée, étant Microsoft.Data.Sqlite et une autre par exemple, ScottPlot ce qui a été utilisé pour de la facilité à créer et remplir des graphiques avec des données.

Finalement, j'ai pu approfondir toutes mes compétences en Programmation Fonctionnelle, POO et appris des nouvelles librairies. Ceci fut ma première fois où j'utilise Windows Forms pour un projet en C#.

Bilan personnel

Personnellement, j'ai trouvé ce projet intéressant et utile pour ma formation et future profession car je trouve il est important de connaître et maîtriser plusieurs manières de coder, mais aussi une qui aide à éviter de se répéter. Je me suis senti efficace dans ma manière de coder et développer cette application car malgré les 2-3 premières semaines où j'ai dû être absent à cause d'un entretien pour mon stage ainsi qu'une absence de deux périodes où j'ai dû partir, j'ai pu bien avancer dans mon projet et même si j'ai dû passer 30% du temps à documenter.

J'ai moins apprécié les User Stories car cela m'a pris beaucoup de temps à réaliser puisque je les faisais toujours faux, après 1 an de pratique. Mais d'un point de vue positif, j'ai pu mieux comprendre le but des User Stories et comment en faire pour qu'ils aient une utilité à mon application, ainsi que les tests d'acceptance.

J'ai bien aimé le fait de faire un programme où il y a du graphique, et non pas juste une console comme la plupart des projets de programmation en C# à l'ETML.

Si c'était à refaire, je commencerai à me bien concentrer sur les User Stories au début comme ça je n'aurais pas besoin d'y revenir plusieurs fois et les corriger. J'aurai également moins utilisé l'IA ou je serai allé plus loin pour comprendre les nouvelles choses que j'apprends durant le développement de mon application, par exemple les « `int nameIndex = Array.FindIndex(headers, h => h.Equals("Game", StringComparison.OrdinalIgnoreCase));` » pour éviter une autre perte de Joker.

10 SOURCES

ChatGPT

<https://learn.microsoft.com/en-us/dotnet/standard/data/sqlite/transactions>

<https://www.instructables.com/Reading-and-Writing-Data-Into-SQLite-Database-Usin/>

<https://stackoverflow.com/questions/75525848/reading-all-records-from-sqlite-local-file-using-c-sharp>

Kaggle – données

<https://scottplot.net/>

<https://scottplot.net/cookbook/5/>

<https://learn.microsoft.com/en-us/dotnet/standard/data/sqlite/types>

<https://learn.microsoft.com/en-us/dotnet/standard/data/sqlite/connection-strings>

<https://learn.microsoft.com/en-us/dotnet/standard/data/sqlite/?tabs=net-cli>

Doc officielle de Microsoft pour Win Forms