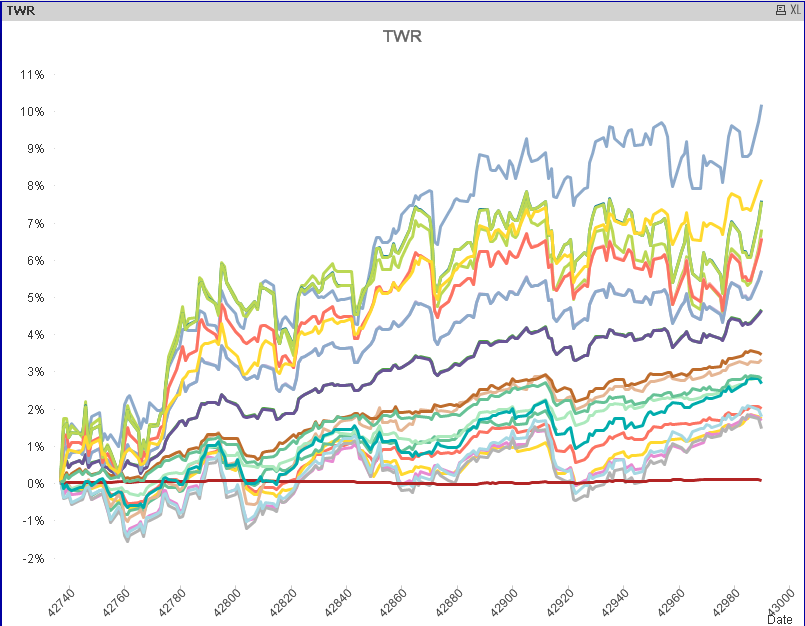
Plot Those Lines



Samuel Sallaku – CID3B

Section Informatique - Vennes

32p

Xavier Carrel

Table des matières

[1 Spécifications 3](#_Toc210996188)

[1.1 Description 3](#_Toc210996189)

[1.2 Objectifs produit 3](#_Toc210996190)

[1.3 Objectifs pédagogiques 3](#_Toc210996191)

[1.4 Domaine d’application 3](#_Toc210996192)

[2 Planification 3](#_Toc210996193)

[3 User Stories 4](#_Toc210996194)

[4 Maquettes 6](#_Toc210996195)

[5 rAPPORT DE TESTS 7](#_Toc210996196)

[5.1 Tableau de tests 7](#_Toc210996197)

[6 journal de travail 7](#_Toc210996198)

[7 usage de l’ia 7](#_Toc210996199)

[8 conclusion 7](#_Toc210996200)

[9 SOURCES 7](#_Toc210996201)

# Spécifications

## 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.

## 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.

## 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 reflexe d’implémenter, comme l’instanciation d’une classe dans un fichier appart. J’avais créé la classe GameData avec tout le reste du code.

J’ai aussi pu remettre en pratique les principes du POO, ce que j’ai appris en 2ème. Cela m’a permit d’approfondir les connaissances du POO comme l’encapsulation, héritage ou bien polymorphisme.

De plus, j’ai ajouté plus de commentaires pour comprendre certains aspects compliqués à s’en rappeler de mon 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.

## 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

# Planification

Les tâches principales sont sur GitHub Projects, [lien ici](https://github.com/users/samumuku/projects/7/views/1) .

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.

# User Stories

**Séries temporelles -** [**lien**](https://github.com/users/samumuku/projects/7/views/1?pane=issue&itemId=127586871&issue=samumuku%7CP_FUN-SamuelSallaku%7C7)

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**](https://github.com/users/samumuku/projects/7/views/1?pane=issue&itemId=127587743&issue=samumuku%7CP_FUN-SamuelSallaku%7C8)

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**](https://github.com/users/samumuku/projects/7/views/1?pane=issue&itemId=127587955&issue=samumuku%7CP_FUN-SamuelSallaku%7C9)

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**](https://github.com/users/samumuku/projects/7/views/1?pane=issue&itemId=127588189&issue=samumuku%7CP_FUN-SamuelSallaku%7C10)

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**](https://github.com/users/samumuku/projects/7/views/1?pane=issue&itemId=127588427&issue=samumuku%7CP_FUN-SamuelSallaku%7C11)

En tant qu’utilisateur,

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

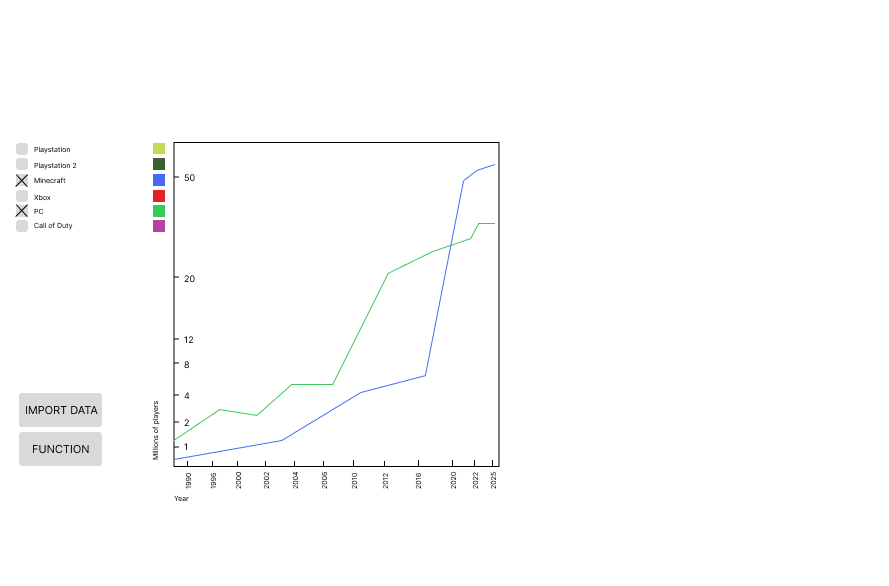
Afin de pouvoir changer l'affichage

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)

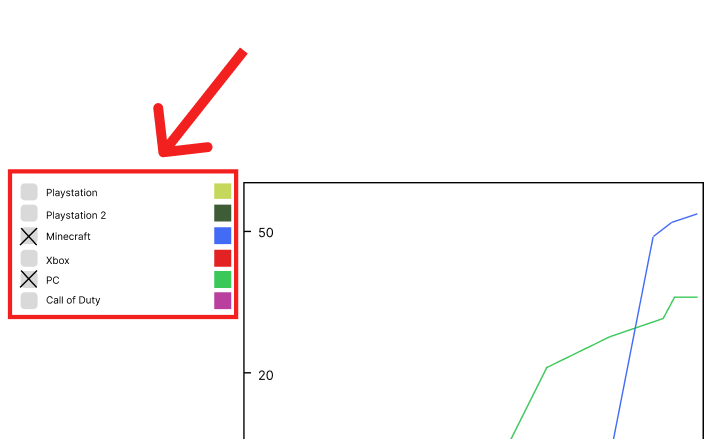
# Maquettes

**Maquette principale**



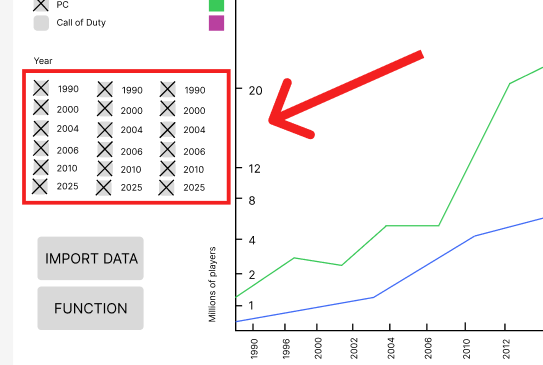
Ici on montre la page principale, ce que l’utilisateur verra dès le lancement du programme (sans les données, ce sont juste un exemple)

**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

# rAPPORT DE TESTS

## Tableau de tests

# journal de travail

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

# usage de l’ia

# conclusion

# SOURCES