



# Rapport de projet - Virtual Trader

UE

IF3A

Noms

Matis MICHEL

François Hietel

Enzo PAPE

Langages

HTML

CSS

JAVASCRIPT

PHP

SQL

GitHub

<https://github.com/Mika-42/VirtualTrader>

# Sommaire

---

Introduction .....	1
Les fonctionnalités générales .....	1
Fonctionnalités propres au jeu .....	1
Schéma d'interaction des pages .....	2
Schéma relationnel .....	3
Diagramme Entité Association .....	4
Page d'accueil .....	5
Page de création de compte .....	5
Page de connexion .....	6
Page de réinitialisation de mot de passe .....	6
Page de menu .....	7
Page du jeu .....	8
I. Interface graphique .....	8
II.mécanique d'évolution des prix .....	9
III.Tableau d'équilibrage .....	10
Page de fin de partie .....	10
Organisation .....	11
Conclusion .....	11
Sources .....	11

---

# Introduction

Dans ce projet Virtual Trader, nous devons réaliser un jeu reprenant le concept des sites de trading en utilisant les langages de balisage **HTML**, **CSS** et de programmation JavaScript, PHP et **SQL**. La partie SQL est administrée par le logiciel **XAMPP**. Nous devons donc implémenter un nombre défini de fonctionnalités.

## Les fonctionnalités générales

- Inscription (email, mot de passe)
- Connexion
- Modifier son mot de passe
- Rechercher un joueur et suivre un joueur  
(cela permet de voir ses derniers achats/ventes)

## Fonctionnalités propres au jeu

- Initialiser le jeu
- reprendre le jeu
- Rechercher et filtrer les actions disponibles
  - par nom
  - par prix
  - par pourcentage de progression
- Visualiser l'évolution du prix d'une action sur les 12 derniers mois
- Visualiser l'état du jeu
  - quantité d'argent possédé
  - liste des actions possédées
  - évolution de la valeur du portefeuille sur les 12 derniers mois)
- Visualiser le classement des joueurs (en fonction de la valeur totale du portefeuille)
- Dérouler les tours de jeu
- Acheter/vendre une action

Le déroulement du jeu est le suivant. Le jeu fonctionne en pseudo temps réel. Les joueurs peuvent effectuer les actions de leur choix et les conséquences de ces actions sont appliquées immédiatement. A pas de temps fixe, le jeu est mis à jour, c'est-à-dire que :

- La date du jeu augmente d'un mois
- Le joueur perçoit les dividendes de ses actions s'il y en a (une action ne verse qu'un seul dividende par an, à condition de posséder l'action à la date de distribution du dividende)

- Les valeurs des actions sont mises à jour, cela affecte le prix auquel nous pouvons acheter les actions, mais aussi le prix auquel nous pouvons les vendre.

Pour mettre à jour le prix d'une action, nous utilisons le principe suivant :

l'évolution du prix est calculée à partir de l'évolution

du prix du mois précédent auquel on ajoute aléatoirement une valeur comprise entre -3 et +3

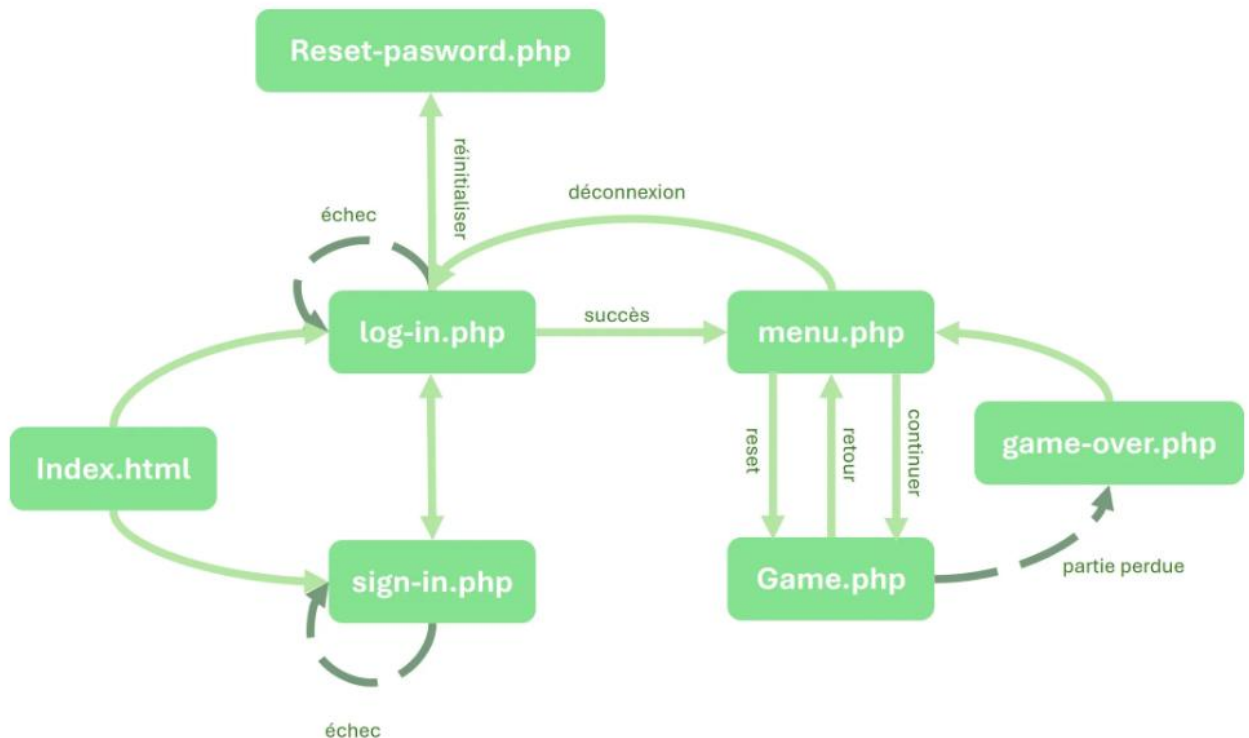
points, sans pouvoir dépasser les bornes -10% et +10% et sans pouvoir descendre en dessous

de 1€ par action. Par exemple, si le mois précédent l'action avait monté de +5%, pour le mois

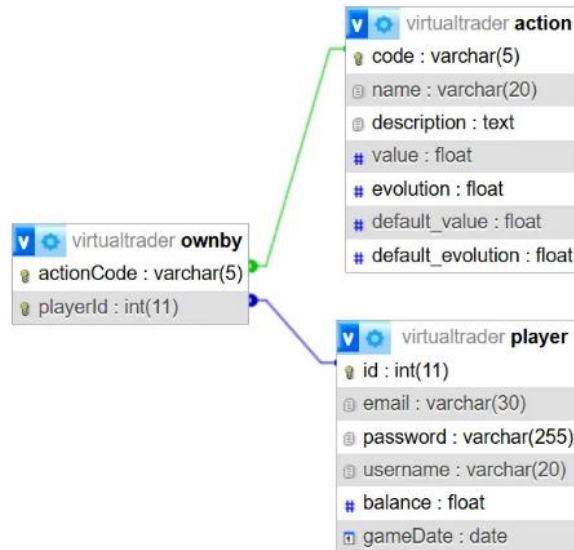
en cours le prix de l'action évoluera selon une valeur aléatoire comprise entre +2% et +8%.

A cela nous avons ajouté la probabilité qu'un krach boursier ai lieu.

## Schéma d'interaction des pages



# Schéma relationnel



- action

Dans la table **action** nous stockons les données qui permettent d'identifier une action, c'est à dire, son *code*, son *nom*, sa *description*, sa *valeur* et son *évolution*.

nous avons décider de rajouter deux attributs *default\_value* et *default\_evolution*, ceux-ci gardent en mémoire le prix initiale des actions. Ainsi lorsque nous réinitialisons le jeux, nous assignons aux attributs *value* et *evolution* les valeurs de *default\_value* et *default\_evolution*.

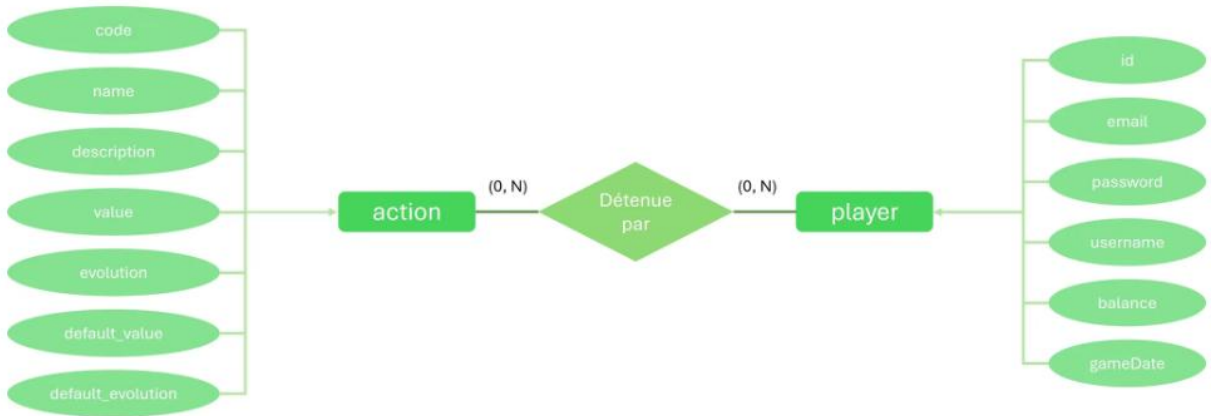
- player

Cette table nous sert à enregistrer les joueurs, ils sont identifié par un numéro unique *id*, leur email est stocké et le hash de leur mot de passe également. chaque joueur possède un pseudonyme et des liquidités (*balance*) fixée au départ à 10 000€. Egalement à chaque joueur est relié la date de son jeu a pour que chacun puisse joueur à son rythme.

- ownby

Enfin, cette dernière table est présente afin de faire l'inventaire des actions possédés par les différents joueurs, elle comporte deux clés étrangère chacune référençant l'unique identifiant des tables **action** et **player**.

# Diagramme Entité Association



La logique de fonctionnement de notre base de donnée est la suivante:

“Un joueur peut avoir entre 0 et  $N$  actions, une action peut être détenue par  $N$  joueurs”

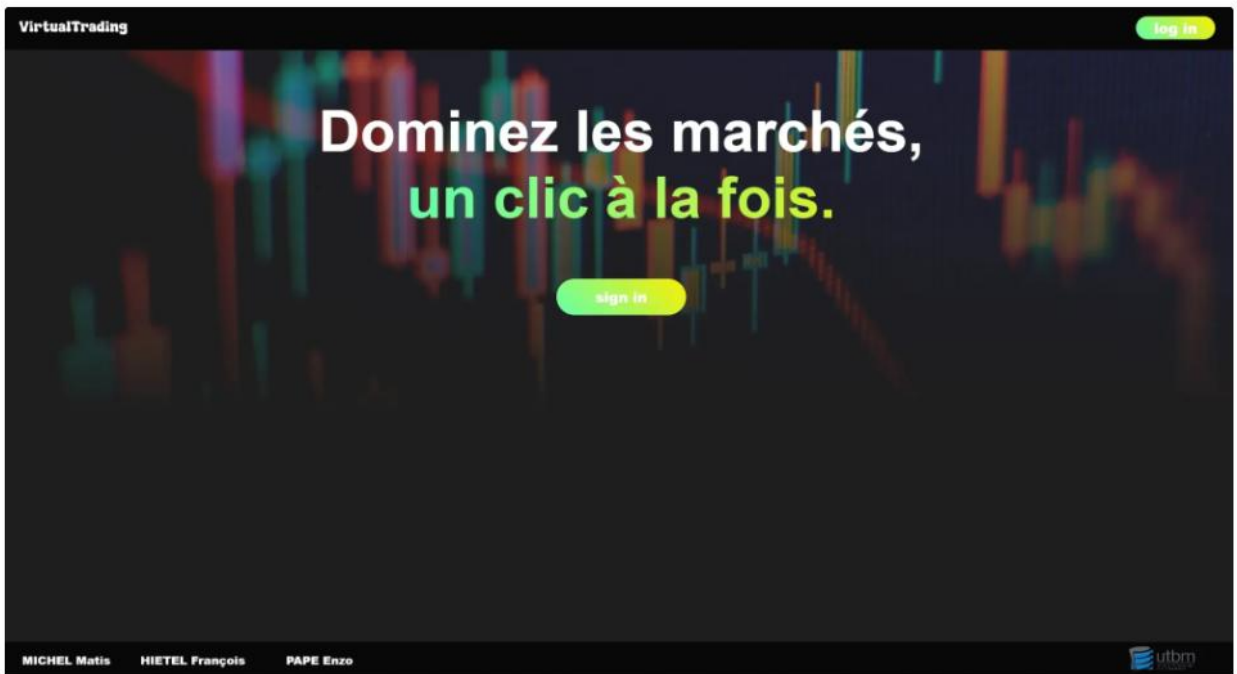
De ce fait,

- Un joueur peut posséder
  - aucune action → cardinalité minimale de 0.
  - jusqu'à  $N$  actions → cardinalité maximale de  $N$ .
- Une action peut être détenue par
  - aucun joueur → cardinalité minimale de 0.
  - jusqu'à  $N$  joueurs → cardinalité maximale de  $N$ .

La relation “Détenue Par” est intrinsèquement une table de jointure contenant deux clés étrangères, l'une sur l'identifiant du joueur et l'autre sur le code d'une action. Elle forme une association  $N:N$ .



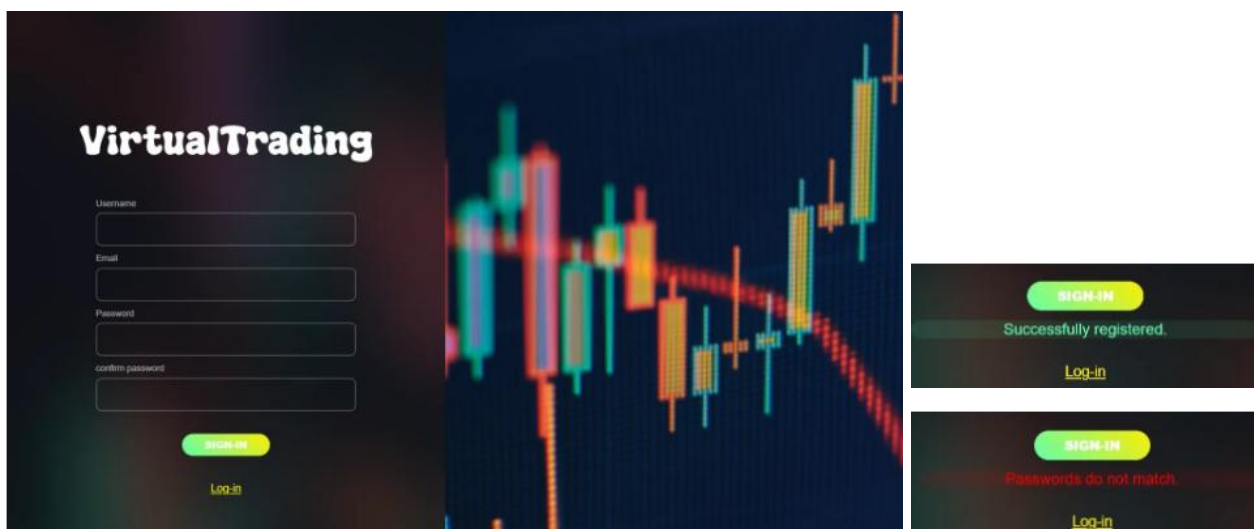
## Page d'accueil



Depuis cette page nous pouvons décider de créer un compte ou de se connecter. Le design de toutes les pages sont fortement inspirés du site web du distributeur de musique TuneCore.

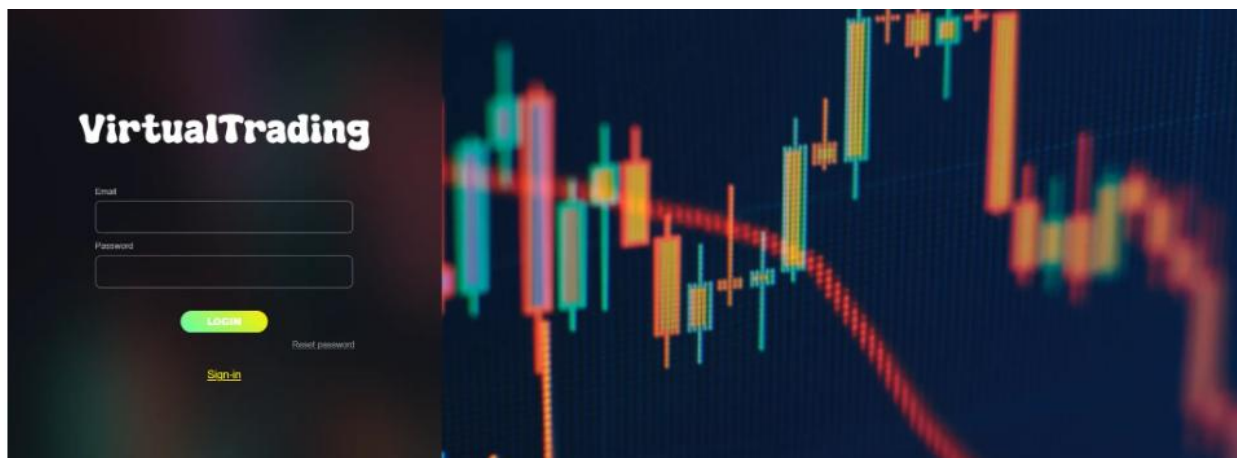
Nous avons choisi le slogan "Dominez les marchés, un clic à la fois" dans la perspective de susciter un intérêt chez les futur joueur pas encore inscrit.

## Page de création de compte



Sur cette page nous avons la possibilité de créer un compte. en arrière plan le programme vérifiera si le compte existe déjà ou si le mot de passe ne correspond pas, il affichera un message d'erreur en rouge. Si aucune erreur n'est présente alors le message de réussite en vert sera affiché. Nous avons également le choix de se diriger vers la page de connexion via le lien hypertexte "Log-in"

## Page de connexion



Email or password is incorrect.

Ici, nous pouvons nous connecter à notre compte, si le mots de passe ou l'email ne correspond pas un message d'erreur est affiché en rouge. Depuis cette page nous pouvons soit être redirigé sur la page de création de compte ou bien sur la page de réinitialisation de mot de passe.

Pour vérifier de manière sécurisé les mots de passes, nous faisons directement le hash du mot de passe saisi puis nous vérifions à l'aide d'une fonction PHP s'il correspond au hash du joueur.

## Page de réinitialisation de mot de passe



Nous pouvons réinitialiser notre mot de passe en remplissant les champs de saisie. Deux type d'erreur peuvent être occasionné:

- cas n°1: le compte que l'on essaye d'actualiser n'existe pas ou l'ancien mot de passe ne correspond pas avec le hash présent dans la base de donnée
- cas n°2: le nouveau mot de passe et sa confirmations ne sont pas identique

Si aucun de ces cas de figure n'est présent alors le changement s'effectue et un message d'information est affiché en vert pour nous confirmer que le changement a bien été effectué.

A la suite de cela nous pouvons retourner sur la page de connexion.



## Page de menu



Une fois arrivé sur cette page, un message personnalisé est affiché pour salué le joueur.

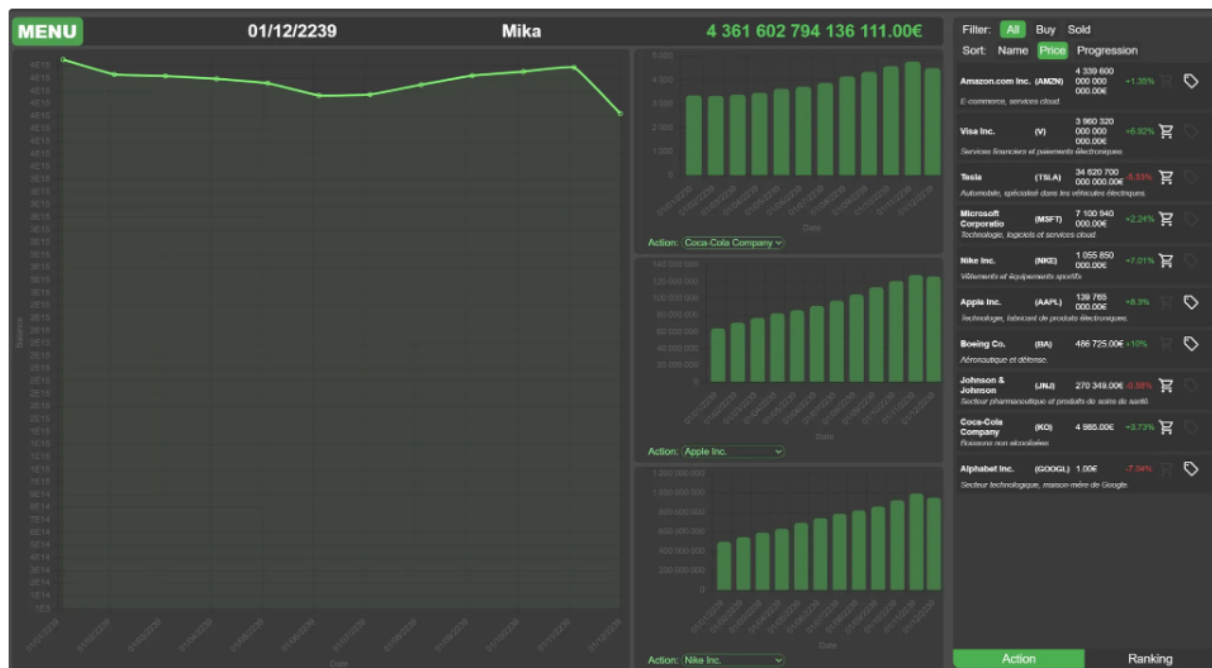
Trois choix s'offrent à lui, il peut:

- continuer sa partie → ses données sont extraite de la base de données et chargée pour lancer le jeu
- réinitialiser sa partie → ses données de jeu sont écrasées et remplacées par les données initiale, suite à cela le jeu est lancé.
- se déconnecter → sa session est détruite et il est renvoyer vers la page de connexion

Afin de pouvoir suivre le joueur au fil de ses déplacement entre les pages, son identifiant est transmis de pages en pages à travers l'url tant que celui-ci ne s'est pas déconnecté.

# Page du jeu

## I. Interface graphique



L'interface du jeu se décompose en six parties. Tout d'abord la première partie tout en haut où sont affichées les informations concernant le joueur qui sont, la date actuel du joueur, son pseudonyme et son portefeuille virtuel qui est la somme de ses liquidité et des valeurs des actions qu'il possède. De plus à gauche de cette section se trouve un bouton qui met en pause le jeu et nous redirige vers la page de menu

En dessous du bandeau informatif, à gauche se trouve le graphique principal permettant de suivre l'évolution mensuelle du portefeuille en euros.

A droite de ce graphique se trouve trois histogrammes permettant de suivre simultanément l'évolution mensuelle du prix d'une action en euros.

Enfin tout à droite, nous retrouvons une sections proposant deux vues. La première permet de visualiser et d'acheter des actions. Elle permet aussi de filtrer et trier les actions pour consulter uniquement celles que l'on possède ou non, ou bien toutes les apercevoir. Nous pouvons également les trier par ordre alphabétique, par prix croissant ou par évolution croissante. La deuxième vue nous permet de visualiser le classement des joueurs les plus fortunés.

Les actions que nous avons sélectionner pour notre projet sont les suivantes;

- **Apple Inc. (AAPL)** – *Technologie, fabricant de produits électroniques.*
- **Tesla Inc. (TSLA)** – *Automobile, spécialisé dans les véhicules électriques.*
- **Amazon.com Inc. (AMZN)** – *E-commerce, services cloud.*
- **Microsoft Corporation (MSFT)** – *Technologie, logiciels et services cloud.*
- **Alphabet Inc. (GOOGL)** – *Secteur technologique, maison mère de Google.*
- **Coca-Cola Company (KO)** – *Boissons non alcoolisées.*
- **Nike Inc. (NKE)** – *Vêtements et équipements sportifs.*
- **Johnson & Johnson (JNJ)** – *Secteur pharmaceutique et produits de soins de santé.*
- **Boeing Co. (BA)** – *Aéronautique et défense.*
- **Visa Inc. (V)** – *Services financiers et paiements électroniques.*
- **NVIDIA Corporation (NVDA)** – *Technologie, processeurs graphiques et intelligence artificielle.*
- **Procter & Gamble Co. (PG)** – *Biens de consommation, produits d'hygiène et ménagers.*
- **McDonald's Corporation (MCD)** – *Restauration rapide.*
- **Berkshire Hathaway (BRK)** – *Géant financier dirigé par Warren Buffett.*

## II. mécanique d'évolution des prix

Afin de faire fluctuer les prix nous appliquons l'algorithme suivant donné par la consignes que nous avons améliorer en rajoutant une probabilité qu'un krach boursier ai lieu

- La nouvelle évolution est la somme de l'évolution du mois précédent et d'une valeur aléatoire entre  $[-3 ; 3]$
- Restreindre la nouvelle évolution obtenue entre  $[-10 ; 10]$
- tirer aléatoirement un nombre entre  $[1 ; 100]$ , s'il est inférieur ou égale à 3.5 alors c'est un krach boursier
- Dans le cas d'un krach boursier la nouvelle évolution est située entre  $[-30 ; -20]$
- Ajouter au prix du mois précédent la nouvelle variation calculée
- Restreindre le prix final entre  $[1, 1\ 000\ 000]$

### III. Tableau d'équilibrage

durée de jeu (années) - recherche d'échec		durée d'un mois (secondes)	durée de jeu réelle (secondes) recherche d'échec	durée de jeu réelle (minutes) recherche d'échec
4		0,5	15	0,25
3		0,6	18	0,3
2		0,7	21	0,35
3		0,8	24	0,4
2		0,9	27	0,45
2		1	30	0,5
2		1,1	33	0,55
2		1,2	36	0,6
1		1,3	39	0,65
4		1,4	42	0,7
		1,5	45	0,75
		1,6	48	0,8
Moyenne	2,5			

Suite à ces ajustements nous avons dressé un tableau afin d'évaluer la durée optimale d'un mois pour que le jeu soit correctement équilibré. Ici, nous avons évalué le temps d'une partie où le joueur ferait absolument tout pour perdre. Nous avons observé que il fallait en moyenne 2.5 ans dans le jeu pour perdre une partie. A partir de cela, nous avons fait le calcul suivant pour déterminer la durée temporelle idéale d'un mois:

$$x \cdot 12 \cdot 2.5 = y$$

- $x$  : durée d'un mois en secondes
- $y$  : durée du jeu réelle en secondes

### Page de fin de partie



Lorsque le portefeuille du joueur passe en dessous de 1 000€, celui ci a perdu la partie et une page s'ouvre pour lui indiquer qu'il a perdu et le rediriger vers le menu où il pourras réinitialiser sa partie

## Organisation

Afin de pouvoir nous organiser dans la réalisation de ce projet nous avons mis en place un dépôt sur GitHub afin d'avoir une gestion simple du projet et de ses différentes versions. Cela nous a été grandement utile lorsque nous voulions changer l'architecture du projet, nous avons créé une branche sur laquelle nous pouvions tester des améliorations sans risquer d'endommager le projet principal. Lorsque les tests d'améliorations prirent fin, nous avons juste à supprimer la branche si ceux-ci n'avaient pas été concluents. Dans le cas contraire nous fusionnions la branche de test avec la branche principale. Le second avantage de l'utilisation de Git et GitHub a été le suivi des actions de chaque membre du groupe, ainsi nous pouvions faire des revues de code et valider ou non des modifications.

Dans un second temps, nous avons mis en place un serveur Discord avec différents salons dans le but de partager les visuels de l'application mais aussi nos sources, les schémas et autres informations.

## Conclusion

En somme, ce projet nous a permis d'acquérir des compétences organisationnelles et de gestion. Nous avons pu nous familiariser avec des outils professionnels tels que Git et GitHub. De plus nous nous avons appris à élaborer l'architecture du projet avec d'autres outils comme phpMyAdmin dans lequel nous avons conçu la base de données. Également, l'usage de design pattern afin de récupérer des informations dans la base de données depuis PHP et les envoyer vers le JavaScript de manière sécurisée nous a permis de comprendre l'importance de l'architecture d'un projet. De même nous avons pu exprimer notre créativité en personnalisant les pages web selon notre convenance.

## Sources

- inspiration pour le visuel du site → <https://web.tunecore.fr/login?check=1>
- image d'arrière plan: [https://stock.adobe.com/ch\\_fr/search?k=forex+chart&asset\\_id=486071336](https://stock.adobe.com/ch_fr/search?k=forex+chart&asset_id=486071336)
- police utilisée pour le logo: <https://www.fontspace.com/super-dream-font-f101954>
- icônes utilisées: <https://fonts.google.com/icons>