

Propositions de différents projets POO

A l'attention de : M. CREGUT

Par : FRANCOIS Yoann, CHAVEROUX Pierre, SILVESTRE Thomas, PRADIER
Corentin, ZIUZIN Nikita, LOPINET Stéphane

Date : Vendredi 21 Mars 2025

1 - BomberManN7

Contexte du projet long : L'idée consiste à développer un jeu multijoueur local compétitif en 2D, inspiré de Bomberman, avec une approche stratégie/action. L'application permettra aux joueurs de :

- Affronter jusqu'à 4 joueurs ou plus tour à tour, avec un menu de configuration avancé (choix de la carte, mode de jeu, nombre de joueurs).
- Utiliser des mécaniques de base intuitives : déplacements, pose de bombes, destruction de blocs.
- Profiter de 3 power-ups stratégiques pour dynamiser nos parties (portée, vitesse, bouclier).
- Expérimenter plusieurs modes de jeu (Classique et Chaos) pour maximiser la rejouabilité.

Systèmes de Jeu Dynamiques

- **Mécaniques de base :**
 - Déplacements au tour par tour, pose de bombes avec délai d'explosion, destruction de blocs destructibles.
 - Collisions gérées via un système de grille pour une précision optimale.
- **Power-ups :**
 - Portée des bombes (+1 case), Vitesse de déplacement (+20%), Bouclier (protège contre 1 explosion).
 - Apparition aléatoire après la destruction de blocs.
- **Carte interactive :**
 - Blocs fixes (indestructibles) et destructibles (cachent parfois des pièges ou power-ups).
 - Pièges tactiques : zones de glace (dérapages), sables mouvants (ralentissements), zones de lave.

Interface Utilisateur

- Design épuré avec accès rapide aux modes de jeu et à la configuration globale.
- Affichage des scores, temps restant, et power-ups actifs.
- Mini-carte pour localiser les adversaires.
- Animations simples pour les explosions et pièges.

Notre valeur ajoutée

Pour résumer, notre futur jeu BomberManN7 vise à offrir une expérience multijoueur unique et compétitive. Le gameplay sera à la fois accessible et intuitif, permettant à tous les joueurs de s'immerger rapidement dans l'action. La profondeur stratégique apportée par les "power-ups" et les pièges tactiques maintiendront l'intérêt sur le long terme, tandis que les différents modes de jeu et cartes variées assureront une rejouabilité élevée.

L'objectif est avant tout de moderniser un classique tout en conservant son essence, BomberManN7 attirera ainsi à la fois les fans nostalgiques et les nouveaux joueurs, tout en restant accessible. Nous comptons sur vous pour choisir ce projet !

2 - Java Embarqué : Récupération de donnée en temps réel

• Contexte du projet :

Ce projet vise à développer un programme Java pour l'acquisition de données critiques en temps réel, en particulier dans les secteurs de la mobilité (automobile, avionique, spatial). L'objectif est de créer un système capable de collecter les informations provenant de divers capteurs d'un véhicule. Initialement, les données seront simulées avant d'être capturées à partir de capteurs réels.

• Description technique et avantages concurrentiels

Fonctionnalités proposées :

- Acquisition de données en temps réel (Donnée simulée ou réelle)
- Type de donnée récupérée : Vitesse, Accélération, Position, ...
- Traitement de la donnée et publication des payloads sur serveur MQTTS
- Applicatif robuste : automate minimal déterministe
- Raspberry Pi Pico pour une solution compacte et économique
- Interface graphique pour visualisation de la donnée

Avantages concurrentiels :

- Coût réduit de la solution
- Adaptabilité à différents types de capteurs
- Modularité à l'ensemble du spectre de cas d'usage nécessitant de la récupération de donnée en temps réel.

• Applications de test

Applications potentielles :

- Tests sur voitures/hélicoptères téléguidés
- Tests sur circuit de vitesse moto

Analyse de marché :

- Croissance du marché : Essor du spatial et de l'eco-mobilité en Europe
- Opportunités de partenariats au sein de l'ENSEEIH : N7RacingTeam et Astre

• Plan de Développement

1. Conception et prototypage
2. Développement du programme
3. Tests et validation sur donnée simulée
4. Intégration des premiers capteurs physiques
5. Tests et validation sur donnée réelle

En résumé, ce projet offre à l'équipe une belle opportunité de renforcer ses compétences en Java et dans le domaine de l'embarqué. Il s'inscrit parfaitement dans les enseignements de ce semestre, couvrant des sujets tels que les automates. La véritable valeur ajoutée de ce projet se trouve dans la mise en situation réelle de la solution proposée, avec des tests concrets et des opportunités d'application au sein des associations de l'école.

3 - Java Finance : Plateforme de Simulation de Portefeuille d'Investissement

Contexte : L'idée consiste à créer une Plateforme de Simulation de Portefeuille d'Investissement en Java. L'application permettra aux utilisateurs de :

- Créer et gérer des portefeuilles fictifs en y ajoutant divers actifs (actions, obligations, ETF, etc.).
- Simuler l'évolution des marchés financiers en temps réel ou via des scénarios prédéfinis.
- Analyser la performance du portefeuille grâce à des indicateurs clés (rendement, volatilité, ratio de Sharpe, etc.).
- Tester différentes stratégies d'investissement dans un environnement sans risques.

Un système de conseils intelligents

- **Recommandations basées sur le profil** : Un moteur de règles suggère des ajustements (ex : réduire les actions tech si l'aversion au risque est élevée).
- **Analyse de données historiques** : Intégration d'API de marchés financiers pour simuler des stratégies passées.

Une gestion des profils d'investisseurs

- **Profils prédéfinis** :
 - **Conservateur** : Priorité aux obligations et ETF stables.
 - **Équilibré** : Mix actions/obligations avec diversification sectorielle.
 - **Audacieux** : Exposition élevée aux actions volatiles, crypto, etc.
- **Questionnaire d'évaluation** : Détermination automatique de la tolérance au risque via des questions (horizon temporel, objectifs financiers).

Une interface intuitive et personnalisable

- **GUI** : Une interface graphique interactive permet de visualiser en temps réel la performance du portefeuille, avec des graphiques (courbes d'évolution, répartition des actifs).
- **Tableau de bord personnalisé** : Affichage d'indicateurs clés adaptés au profil de risque de l'utilisateur.

Une pédagogie et prise de décision éclairée

- **Comparaison de stratégies** : Visualisation côte à côte des résultats de différentes approches (ex : Buy & Hold vs Trading actif).
- **Journal d'activité** : Historique des décisions prises et de leur impact sur le rendement, pour un apprentissage par l'erreur.

Bilan : En conclusion, cette Plateforme de Simulation de Portefeuille d'Investissement en Java offre une solution complète et innovante pour les investisseurs débutants et expérimentés. Elle combine des fonctionnalités avancées de gestion de portefeuille, d'analyse de performance, et de simulation de marché avec un système de conseils intelligents et une interface utilisateur intuitive. La plateforme permet non seulement de tester diverses stratégies d'investissement sans risque réel, mais aussi d'apprendre et de s'améliorer grâce à des outils pédagogiques intégrés.

4 - Java Cybersécurité : Worm Rootkit

Ce projet ambitieux consiste à développer un ver informatique de type rootkit en Java, un programme malveillant qui se propage à travers les réseaux et s'installe sur les systèmes cibles pour obtenir un accès privilégié.

Ce projet est purement éducatif et vise à comprendre les mécanismes de sécurité informatique et les techniques de programmation orientée objet (POO) appliqués à la cybersécurité.

Fonctionnalités Principales

1. **Propagation** : Le ver doit pouvoir se propager à travers un réseau, via des périphériques USB
2. **Dissimulation** : Le rootkit doit se cacher des outils de détection classiques et des utilisateurs. Il devra donc passer outre les antivirus les plus faibles.
3. **Commande et contrôle** : Le rootkit doit permettre le contrôle d'une machine infectée à distance depuis une interface graphique claire.
4. **Modules Extensibles** : Ajouter des fonctionnalités supplémentaires comme la capture de frappes, l'enregistrement d'écran, etc.

Architecture globale du Projet

1. **Classe Ver** : Gère la propagation du ver à travers le réseau.
2. **Classe Rootkit** : Responsable de l'installation et de la dissimulation sur le système cible.
3. **Interface Module** : Interface pour ajouter des fonctionnalités supplémentaires.
4. **Classe Utilitaire** : Fournit des outils pour la manipulation du système et la gestion des ressources.
5. **GUI** : Une interface graphique pour le contrôle du rootkit et la visualisation de la propagation du ver dans le réseau.

Conclusion

Ce projet permettra de comprendre les mécanismes de sécurité informatique et d'appliquer les concepts de la programmation orientée objet en Java. Il est essentiel de rappeler que ce projet est purement éducatif et doit être réalisé dans le respect des lois et de l'éthique.