

PROJET NSI TERMINALE

REPRODUIRE LE JEU DE SOCIÉTÉ STAR REALMS

Bienvenue dans ce projet où vous allez programmer une version numérique simplifiée du jeu **Star Realms** !
Ce projet mettra en pratique vos compétences en programmation orientée objet (POO), gestion des données et logique de jeu.

Objectifs du projet

1. **Appliquer la POO** : Créer des classes pour représenter les éléments du jeu (cartes, joueurs, etc.).
2. **Structurer un jeu** : Implémenter les règles de Star Realms en plusieurs étapes.
3. **Travailler en équipe** : Décomposer le travail et collaborer efficacement.
4. (Bonus) **Créer une interface utilisateur** : En mode console ou graphique.

Les bases du jeu

Star Realms est un jeu de deck-building. Chaque joueur commence avec un deck basique de cartes et doit acheter de nouvelles cartes pour améliorer son deck. Le but est de réduire les points de vie de l'adversaire à 0.

Règles simplifiées

1. Chaque joueur commence avec **50 points de vie**.
2. Les cartes se divisent en deux types principaux :
 - **Vaisseaux** : Ils produisent du **commerce** (pour acheter des cartes) ou du **combat** (pour attaquer l'adversaire).
 - **Bases** : Restent en jeu pour offrir des bonus chaque tour.
3. À chaque tour, un joueur :
 - Pioche 5 cartes.
 - Joue des cartes pour obtenir des ressources.
 - Utilise le commerce pour acheter des cartes.
 - Utilise le combat pour attaquer l'adversaire ou ses bases.
 - Termine son tour en défaussant toutes les cartes jouées et pioche à nouveau.

Plan de travail

1. Analyse et conception

Travail attendu :

- Identifiez les éléments principaux du jeu et leurs interactions.
- Proposez une structure sous forme de classes.
 - Exemple de classes : Carte, Vaisseau, Base, Joueur, Jeu.
- Réalisez un diagramme UML simplifié (à la main ou avec un outil en ligne).

2. Développement progressif

Avancez par étapes pour construire le jeu :

Étape 1 : Création des classes de base

- Identifiez les attributs et comportements nécessaires pour les cartes et les joueurs.
- Concevez les classes Carte et Joueur selon vos idées.
 - Exemple d'attributs pour **Carte** : nom, coût, type, effets.
 - Exemple d'attributs pour **Joueur** : points de vie, deck (pioche, main, défausse).
- Implémentez les fonctionnalités de base comme la pioche de cartes ou la gestion des points de vie.
-

Étape 2 : Mécanique de tour

- Implémentez les phases de jeu :
 1. Pioche au début du tour.
 2. Jeu des cartes et activation des effets.

3. Phase d'achat.
 4. Fin du tour et défausse.
- Vérifiez que les actions respectent les règles (par exemple, vérifier qu'un joueur a assez de commerce pour acheter une carte).

Étape 3 : Intégration des cartes

- Ajoutez des cartes avec des effets variés :
 - Commerce : Gagnez des points pour acheter.
 - Combat : Infligez des dégâts à l'adversaire.
 - Bases : Restez en jeu pour donner des bonus chaque tour.
- Testez les cartes une par une.

Étape 4 : Interface utilisateur

- **Console :**
 - Affichez la main du joueur, le deck de cartes disponibles et l'état des joueurs.
 - Permettez de choisir quelles cartes jouer ou acheter.
- **Graphique (bonus) :** Utilisez une bibliothèque comme Tkinter ou Pygame pour afficher les cartes et les interactions.

3. Tests et finalisation

- Testez chaque fonctionnalité séparément.
- Ajoutez des règles avancées (combo de factions, bases indestructibles, etc.).
- Préparez une démonstration.

Livrables attendus

1. Code source bien structuré, commenté et fonctionnel.
2. Diagramme UML des classes.
3. Une démonstration du jeu (en console ou avec interface).

Conseils pour réussir

- **Travaillez par étapes :** Ne cherchez pas à tout faire d'un coup. Construisez le jeu brique par brique.
- **Testez fréquemment :** Assurez-vous que chaque partie fonctionne avant d'ajouter des nouveautés.
- **Documentez votre code :** Utilisez des commentaires pour expliquer vos choix.
- **Collaborez efficacement :** Répartissez les tâches et communiquez régulièrement.

Timeline suggérée

1. **Jour 1-2 :** Analyse et conception (règles, classes, UML).
2. **Jour 3-5 :** Implémentation des bases (classes principales, mécanique de tour).
3. **Jour 6-8 :** Ajout des cartes et des effets.
4. **Jour 9-10 :** Interface utilisateur et finalisation.

Bonne chance ! Soyez créatifs et amusez-vous en programmant ce jeu.