

Programmation Orientée Objets avec Java

Projet POO Jeu de conquête : *JavaQuest*

**Stéphane Malandain / Yassin Rekik
Semestre d'automne 2025**

Introduction

Dans le cadre du cours de POO / Java, vous allez réaliser un projet entier. Ce travail se divise en 3 étapes. Chaque étape fait l'objet d'un rendu à une date fixe, avec une évaluation en présentiel et une note. La moyenne des 3 notes vous donne une note de projet.

Le projet est individuel. Plagiat ou copie seront sévèrement puni ! Les modalités de dépôt vous seront communiquées sur le Git ultérieurement.

A. Présentation du concept

1. Concept général

- **Titre:** *JavaQuest*
- **Genre :** Stratégie, Gestion, Tour par tour
- **But du jeu :** Chaque joueur commence avec quelques territoires et ressources. Le but est de contrôler un maximum de terrains à la fin de la partie.

2. Règles principales

- Le jeu est **tour par tour** sur une map.
- La map est constituée de **tuiles** (hexagonales ou carrées).
- Chaque tuile appartient à un joueur ou est neutre.
- Les joueurs exploitent leurs tuiles pour produire des ressources, puis utilisent ces ressources pour construire, recruter et conquérir.

3. Ressources

Types de ressources :

1. **Nourriture** – produite par les plaines/fermes, sert à nourrir la population et les armées.

2. **Bois** – produit par les forêts, sert à la construction et aux unités légères.
3. **Pierre/Métal** – produit par les montagnes, sert aux fortifications et unités lourdes.
4. **Or** – produit par certaines tuiles spéciales ou bâtiments, sert de monnaie universelle pour échanges.

Gestion des ressources :

- Les ressources sont collectées chaque tour.
- Certaines unités/bâtiments coûtent un **entretien** en ressources chaque tour (ex. nourriture pour soldats).
- Manque de ressources = malus (famine, désertions, bâtiments inactifs).

4. Terrains et cartes

Types de tuiles :

- **Plaine** → + nourriture
- **Forêt** → + bois, bonus défensif léger
- **Montagne** → + pierre/métal, bonus défensif fort, mais difficile à conquérir
- **Eau** → barrière, sauf avec unités spéciales (bateaux)
- **Désert** → faible production, risque d'événements négatifs
- **Tuile spéciale** → ruines, mines d'or, oasis (bonus rare)

5. Bâtiments

Chaque tuile peut contenir un bâtiment qui améliore sa production ou sa défense.

- **Ferme** → + nourriture
- **Scierie** → + bois
- **Mine** → + pierre/métal
- **Marché** → échange de ressources avec perte
- **Fort** → bonus défensif élevé
- **Palissade** → bonus défensif

6. Armée et conquête

Recrutement d'unités :

- **Soldat léger** (coût nourriture + bois) → rapide, faible puissance.
- **Soldat lourd** (coût nourriture + métal) → lent, puissant.
- **Archers** (coût nourriture + bois) → bonus en forêt.
- **Cavalerie** (coût nourriture + or) → bonus en plaine, mobilité.

Conquêtes :

- Pour attaquer une tuile ennemie, il faut y envoyer une armée.
- Résolution du combat :
 - **Force = unités + bonus terrain + fortifications.**
 - Si attaquant > défenseur → conquête réussie.
 - Sinon → défenseur conserve la tuile, pertes pour les deux camps.

7. Événements et challenges

- **Catastrophes naturelles** : sécheresse, inondation, tremblement de terre.
- **Rébellion interne** : si le joueur gère mal ses ressources.
- **Découvertes** : mine d'or, oasis, ruines (bonus unique).
- **Quêtes secondaires** : contrôler X forêts pour obtenir des archers spéciaux, etc.

8. Conditions de victoire

- **Victoire par territoire** : celui qui possède le plus de tuiles après un nombre de tours fixé.
- **Victoire militaire** : élimination de tous les adversaires.
- **Victoire économique** : atteindre un certain stock d'or.
- **Victoire spéciale (optionnelle)** : accomplir un objectif secret donné au début.

9. Progression et rejouabilité

- Parties configurables (taille de carte, nombre de joueurs, durée en tours).
- Ressources et événements aléatoires pour renouveler les parties.
- Différents styles de jeu possibles : expansion militaire, économie, défense.

10. Aspects techniques (Java)

- **Représentation de la carte** : grille (2D array ou collection de tuiles).
- **Classe Tuile** : type de terrain, propriétaire, bâtiment, ressources produites.
- **Classe Joueur** : ressources, armée, tuiles possédées.
- **Classe Armée** : unités, force totale, position.
- **Gestion des tours** : boucle de jeu où chaque joueur joue successivement.
- **IA (plus tard)** : adversaire basique qui choisit entre expansion, défense ou production.

B : Première étape : les bases du jeu

1. Réalisez les classes et les structures de données pour gérer l'ensemble des éléments décrits précédemment

2. Générez une Map initiale aléatoire de 20*20 avec les tuiles qui possèdent leurs ressources, un type, etc...
3. Affectez 5 tuiles aux 2 joueurs avec un solde d'argent au départ.
4. Réaliser le marché où l'on peut acheter des ressources
5. Réalisez la représentation en mode console de la Map
6. Réalisez le Game-Manager permettant aux 2 joueurs de jouer tour à tour. A chaque tour un joueur peut :
 - a. soit investir dans une zone, acheter des ressources ou des unités
 - b. soit se déplacer et conquérir

C : Seconde étape :

Utilisation de l'héritage pour diversifier les types de soldats et de ressources

Utilisation des Interfaces et des classes Abstraites pour améliorer la structure du projet

Utilisation des Collections Java pour optimiser la gestion des ressources

D : Dernière étape :

Passage à un mode GUI

Intégration des tests unitaires

Utilisation des STREAMS