

## Introduction

Vous êtes chargé.e.s de développer les bases d'un jeu de stratégie en Java. Le jeu doit permettre à un joueur de gérer des ressources, des habitants et des bâtiments, tout en cherchant à maximiser leur économie. La stratégie du jeu est à votre discrétion. Nous suggérons que le but du jeu est de réussir à faire prospérer une ville sur un temps donné. D'autres stratégies peuvent être définies comme par exemple réussir à produire une certaine quantité d'un produit noble, ou encore jouer à deux joueurs ou plus en concurrence sur un même espace et dans un même temps, être limité par les ressources limitées, etc.

Chaque bâtiment contient des habitants, consomme des ressources et en produit d'autres. Le but à atteindre est de réussir à faire prospérer l'économie d'une ville sans mettre les bâtiments de production en faillite. Le message éthique d'un tel jeu est intéressant à étudier, mais ce n'est pas notre sujet.

Ce jeu s'inspire librement d'éditions comme *Anno 1602* de Sunflowers ou *Forge Of Empire* d'*InnoGames* dont je vous laisse chercher des références.

Le nombre de joueurs, la stratégie et les éléments du jeu peuvent être modifiés à votre guise. Vous pouvez envisager de fournir en plus d'une version *ligne de commande* une version avec une interface utilisateur graphique (GUI) pour permettre aux joueurs de gérer leur économie plus facilement.

Vous pouvez aussi ajouter une fonctionnalité de jeu en temps réel en plaçant l'interface utilisateur dans un processus parallèle.

Mais ces deux ajouts ne sont pas demandés et risquent de prendre trop de temps. De plus cette partie du programme ne sera pas essentielle dans l'évaluation du projet qui reste un prétexte à évaluer vos connaissances en matière de programmation orientée objet.

## Description du Projet

Vous devrez créer un jeu de stratégie en Java, en vous concentrant sur les éléments suivants :

### Gestion des Ressources

- Concevoir une classe `Resource` pour représenter les différentes ressources disponibles dans le jeu.
- Les ressources disponibles dans le jeu de base sont: `Gold`, `Food`, `Wood`, `Stone`, `Coal`, `Iron`, `Steel`, `Cement`, `Lumber`, `Tools`.
- Quelques ressources sont disponibles au début du jeu (de l'argent et des matériaux pour construire les premiers bâtiments, de la nourriture pour faire vivre les premiers habitants). Le but du jeu est de faire prospérer ces ressources afin qu'elles ne viennent pas à manquer.

### Gestion des Bâtiments

- Concevoir une classe `Building` pour représenter les bâtiments dans le jeu.

- Chaque bâtiment abrite des habitants et des travailleurs. À la construction du bâtiment, le nombre de ces personnes est défini (voir le tableau ci-dessous), et ensuite le joueur peut le faire évoluer. On pourra ajouter un minimum et un maximum d'habitants et de travailleurs pour chaque type de bâtiment.
- La construction d'un bâtiment a un coût en or et en matériaux. Une nouvelle construction demande une durée avant d'être disponible (voir le tableau ci-dessous).
- Un bâtiment produit des ressources et en consomme d'autres en proportion du nombre de travailleurs.
- Le tableau en fin de sujet contient un exemple de valeurs initiales, vous pouvez modifier ces valeurs comme bon vous semble pour rendre le jeu plus jouable.

### Gestion des habitants

- Les habitants sont logés dans des maisons (**House**, **ApartmentBuilding**) ou des entreprises (**WoodenCabin**, **Farm**, **Quarry**). Qu'ils soient travailleurs ou non, ils consomment tous et toutes de la nourriture au fil du temps (une unité de nourriture par personne et par unité de temps) et ont tous besoin d'un logement.

### Mécanique du Jeu

- Concevoir une classe **Manager** pour gérer l'ensemble des bâtiments et des habitants.
- Permettre aux joueurs d'ajouter ou de retirer des bâtiments et d'ajouter ou de retirer des travailleurs dans chaque bâtiment,
- Créer un mécanisme de consommation et de production des ressources par les bâtiments.
- Créer un mécanisme de consommation de nourriture par les habitants.

### Interface Utilisateur

Afin de rendre votre application conviviale et interactive, créez une interface utilisateur simple basée sur les entrées clavier et les sorties écran. Cette interface permet à l'utilisateur d'interagir avec le système de gestion que vous avez développé. Chaque saisie au clavier correspond à une unité de temps ou une action.

### Livrables

Chaque projet est réalisé par 2 personnes. Vous déposerez sur moodle votre travail sous la forme d'une archive .gz dont le nom commencera par le numéro de votre groupe de TD suivi du nom de chaque membre du binôme. L'archive comprendra :

1. Un code source bien documenté pour votre jeu de stratégie. Votre archive contiendra les fichiers source `.java`.
2. Un rapport de projet au format `.pdf` réalisé en **Latex** et contenant
  - (a) Une page de garde avec le nom de chaque participant.

- (b) un chapitre **Analyse** qui reprendra la problématique du projet et présentera les questions auxquelles il faut répondre. Un chapitre **Conception** qui donnera l'architecture de vos classes, les différents modèles Design Patterns que vous aurez éventuellement utilisés et/ou les concepts DDD si vous les avez mis en oeuvre dans votre code.

## Evaluation

- La qualité de la conception du programme sera particulièrement appréciée. On pourra juger par exemple de la facilité avec laquelle des ajouts au jeu seront faciles. Par exemple le jeu pourra évoluer avec l'ajout d'autres bâtiments, l'ajout d'autres ressources comme la monnaie, l'ajout d'éléments de stratégie comme la satisfaction de la population ou l'économie de la ville ou encore l'interface entre joueurs.
- Si vous avez inclus des *Design Pattern*, la mise en oeuvre et la pertinence de ces éléments seront appréciées. N'oubliez pas de justifier ces choix dans le rapport.
- La qualité du code comptera également beaucoup (code lisible qui compile).
- En revanche, les aspects ergonomiques et graphiques, la qualité du jeu, la jouabilité ou l'intérêt du jeu de stratégie ne seront pas déterminants, mais nous encourageons ceux qui peuvent ajouter ces avantages à leur projet de le faire pour leur propre intérêt et la curiosité que cela suscite.

## Exemple de configuration initiale

	Nombre d'habitants	Nombre de travailleurs	Coût	Matériaux de construction	Consommation (/unité de temps)	Production (/unité de temps)	Temps de construction (/unité de temps)
WoodenCabin	2	2	1 Gold	1 Wood	0	Wood 2, Food 2	2
House	4	0	1 Gold	2 Wood, 2 Stone	0	0	4
ApartmentBuilding	60	0	4 Gold	50 Wood, 50 Stone	0	0	6
Farm	5	3	4 Gold	5 Wood, 5 Stone	0	Food 10	2
Quarry	2	30	4 Gold	50 Wood	0	Stone 4, Iron 4, Coal 4, Gold 2	2
LumberMill	0	10	6 Gold	50 Wood, 50 Stone	Wood 4	Lumber 4	4
CementPlant	0	10	6 Gold	50 Wood, 50 Stone	Stone 4, Coal 4	Cement 4	4
SteelMill	0	40	6 Gold	100 Wood, 50 Stone	Iron 4, Coal 2	Steel 4	6
ToolFactory	0	12	8 Gold	50 Wood, 50 Stone	Steel 4, Coal 4	Tool 4	8