

Rapport de projet

# Jeu Tycoon Unity

BUT3 Informatique

GORGODIAN Valentin;TURLURE Nael  
09/01/2026

## Table des matières

Présentation générale du projet.....	2
Déplacement et caméra du joueur.....	2
Génération de pièces (système Tycoon).....	2
Economie et Wallet .....	3
Pads interactifs .....	3
Pads d'achat.....	3
Pads téléporteurs .....	3
Objets interactifs .....	3
Clé.....	3
Coffre.....	3
Zone mortelle (eau).....	4
Effets et immersion.....	4
Choix de conception et scalabilité du système de générateurs .....	4
Répartition du travail.....	5
Méthodologie.....	5

## Présentation générale du projet

Le projet consiste en la réalisation d'un jeu de type Tycoon en 3D avec le moteur Unity. Le joueur contrôle un personnage dans un environnement où il peut générer des pièces, acheter des générateurs, se téléporter, récupérer des objets, et interagir avec différents éléments du monde.

## Déplacement et caméra du joueur

Le joueur est contrôlé via un Rigidbody et un CapsuleCollider, garantissant une physique stable. Un modèle 3D animé (importé depuis [Mixamo](#)) est attaché en tant qu'enfant visuel, sans influencer la physique.

La caméra est une caméra orbitale à la troisième personne, centrée sur le joueur, avec rotation à la souris, zoom à la molette et verrouillage du curseur en jeu.

Les animations (Idle / Run) sont déclenchées automatiquement en fonction de la vitesse du joueur.

## Génération de pièces (système Tycoon)

Le cœur du jeu repose sur des générateurs de pièces composés de :

- Un Dropper (spawn de pièces),
- Un Conveyor (déplacement automatique),
- Un Collector (récupération et ajout au Wallet).

Ces éléments sont regroupés dans un Prefab GeneratorModule, ce qui permet de :

- Dupliquer facilement les générateurs,
- Activer les générateurs progressivement,
- Gérer différents niveaux de puissance (valeur des pièces, vitesse de génération).

Un GeneratorManager contrôle l'ordre de déblocage et les coûts associés.

## Economie et Wallet

Le joueur possède un Wallet qui stocke les pièces gagnées.

Le Wallet permet d'ajouter des pièces (collectors, coffres), de dépenser des pièces (achat de générateurs) et d'appliquer des pénalités (mort dans l'eau).

L'interface utilisateur affiche en permanence le nombre de pièces.

## Pads interactifs

### Pads d'achat

Les pads d'achat permettent d'acheter de nouveaux générateurs.

Un texte s'affiche au-dessus du pad indiquant la touche d'interaction et le prix du prochain générateur.

### Pads téléporteurs

Les pads téléporteurs fonctionnent par paire :

- Marcher sur un pad téléporte le joueur vers l'autre,
- Un système de cooldown empêche les boucles infinies.

## Objets interactifs

### Clé

La clé est un objet flottant, tournant sur lui-même et récupérable en passant dessus.

Une fois ramassée, elle est stockée dans l'inventaire du joueur.

### Coffre

Le coffre fonctionne en deux étapes :

1. Le joueur doit posséder la clé.
2. En entrant dans la zone du coffre, un panel UI s'ouvre automatiquement.

Le joueur peut récupérer la récompense soit en cliquant sur le bouton Récupérer, soit en appuyant sur la touche E.

Le coffre donne des pièces et ne peut être ouvert qu'une seule fois.

## Zone mortelle (eau)

Une zone d'eau mortelle est implémentée :

- Entrer dans l'eau déclenche une mort instantanée,
- Le joueur est téléporté au point de respawn,
- Une pénalité de pièces est appliquée,
- Un flash rouge à l'écran indique la mort.

## Effets et immersion

Pour améliorer l'immersion des sons de pas sont joués lors des déplacements, des particules de poussière apparaissent aux pieds du joueur et des effets visuels sont utilisés lors de la mort et des interactions.

## Choix de conception et scalabilité du système de générateurs

Le game design n'a volontairement pas été approfondi dans ce projet.

La majorité du temps de développement a été consacrée à la conception d'un système de générateurs de type Tycoon robuste et évolutif, qui constitue le cœur du jeu.

Les générateurs (composés de dropper, conveyor et collector) ont été pensés comme des modules indépendants, regroupés sous forme de prefabs et gérés par un manager centralisé.

Cette architecture permet de créer autant de générateurs que souhaité, sans modifier la logique globale du jeu.

Pour ajouter un nouveau générateur, il suffit :

- de créer ou dupliquer un prefab de générateur,
- de le référencer dans le GameManager / GeneratorManager,
- de définir son coût et ses paramètres (vitesse, valeur des pièces, etc.).

L'ensemble du reste du système (économie, interface, pads d'achat, progression) s'adapte automatiquement en fonction du nombre de générateurs disponibles. Aucune modification structurelle du code n'est nécessaire pour faire évoluer le contenu.

Ce choix de conception a demandé un investissement important en temps, notamment pour :

- Garantir une communication propre entre les générateurs et le Wallet,

- Éviter les duplications de logique,
- Rendre le système facilement extensible.

En contrepartie, l'aspect game design reste volontairement minimal.

Le projet se concentre avant tout sur la qualité de l'architecture logicielle et la démonstration d'un système Tycoon scalable, plutôt que sur une expérience de jeu complète.

## Répartition du travail

Nous avons pratiquement travaillé tous les deux sur les mêmes choses.

Valentin a travaillé un peu plus sur les générateurs de pièces, et Nael un peu plus sur tout le reste.

Mais globalement, rien n'a été fait seul, on a travaillé tous les deux sur chaque partie.

## Méthodologie

Des outils d'intelligence artificielle ont été utilisés lors de ce projet afin de comprendre plus facilement certains points. Ces derniers ont servi notamment à ne pas perdre du temps lors de l'apprentissage du Csharp, en fournissant les types.