# 

# 

# **Technical Game Design**

SAÉ Jeu Vidéo

## 

Groupe : VANDENKOORNHUYSE Lorenzo, MALLEVAEY Adrien, CIOLKOWSKI Yann

Projet : Clash of Piglin

## 

## 

## 

[**Technical Game Design**](#_38n23ng6nrfj) **1**

[0. Cadre technique](#_c5ffxqc4y9ij) 3

[1. Map & Terrain](#_5b27c4lohg7) 3

[2. Unités](#_rnci0ikd2qfb) 4

[4. Économie](#_o8e9irz3pckr) 6

[5. Gameplay & UX](#_cke0qn3uqsun) 6

[6. Conditions de victoire & rythme](#_4na1rsqpgxk0) 6

## 

## **0. Cadre technique**

* **Langage :** Python 3.11+
* **Librairies principales :**
  + **Pygame** → affichage 2D, gestion des événements, boucle de jeu.
  + **NumPy** (optionnel) → gestion des matrices (grille).
  + **Algorithme A\*** (implémentation interne) → pathfinding des unités terrestres.

## **1. Map & Terrain**

* **Grille :** matrice 24×24, affichage 2D.
* **Cases :** chaque case contient :  
  + type : netherrack, soulsand, lave, netherrack bleu, netherrack rouge.
  + unités terrestres : liste max 4.
  + unité volante : slot unique.
* **Bastions :** position fixe (coin supérieur gauche et coin inférieur droit).
* **Chemins :** 1 ou 2 générés en priorité, garantissant connectivité.
* **Zones spéciales :** générées par expansion, ordre = lave → soulsand → bleu → rouge.
* **Remplissage final :** netherrack normal.
* **Effets de terrain :**
  + Lave = infranchissable.
  + Soulsand = vitesse ÷ 2 pour unités terrestres.

**Classes principales :**

* MapManager → génère et stocke la grille.
* Tile → définit type de terrain, liste d’unités et règles de traversabilité.

## **2. Unités**

* **Types disponibles :**
  + **Barbare** : terrestre, mêlée.
  + **Arbalétrier** : terrestre, distance, seule unité anti-air.
  + **Ghast** : volant, cible uniquement les structures, mouvement linéaire.
* **Attributs communs :** PV, dégâts, vitesse, portée, cadence d’attaque, coût, temps de production.
* **Mécaniques :**
  + **Production :** file d’attente par joueur, temps de spawn.
  + **Déplacement :**
    - Terrestres → pathfinding A\* en tenant compte du terrain.
    - Ghasts → trajectoire droite vers la cible.
  + **Combat :** détection portée, application dégâts, changement de cible si un ennemi plus proche entre dans la zone.
  + **Caps :** 4 terrestres / case, 1 volant / case.

**Classes principales :**

* Unit (classe mère) → gère attributs de base et méthodes communes.
* Barbarian, Crossbowman, Ghast → spécialisations avec règles spécifiques.
* Pathfinder → implémentation A\* pour unités terrestres.
* FightManager → applique règles d’attaque et de ciblage.

**3. Structures**

* **Bastion :**
  + PV = 1000, destruction = fin de partie.
  + Ne peut pas attaquer.
* **Balise :**
  + PV = 150, coût 600.
  + Max 2 par joueur.
  + Effets activables :  
    - **Vitesse** (+25%) → effet constant.
    - **Régénération** (+10 PV / 3s).
  + Placement interdit sur lave/soulsand.
  + Ciblée en priorité par les Ghasts.

**Classes principales :**

* Structure (classe mère) → gestion PV et interactions.
* Bastion → hérite de Structure, déclenche victoire/défaite.
* Beacon → applique effets dans son rayon aux unités alliées.

## **4. Économie**

* **Revenus automatiques :**
  + 8/s (0–1 min), 10/s (1–2 min), 12/s (2–3 min), 15/s (3–4 min), 18/s (4–5 min).
  + Cap d’or : 1500.
* **Revenus de combat :** +10% du coût de l’unité tuée.
* **Mécaniques :**
  + Tick économique chaque seconde.
  + Attribution automatique des gains de kills.

**Classe principale :**

* EconomyManager → gère ticks, cap d’or, revenus de kills.

## **5. Gameplay & UX**

* **Commandes :**
  + Sélection simple, multiple, ajout via Shift+clic.
  + Clic droit = déplacement / attaque.
* **Interface :**
  + Haut-droit → or total, temps, revenus/s.
  + Bas-centre → boutons de production avec coûts et timers.
* **Feedback visuel :**
  + Changement de couleur après sélection
  + Rayon visible pour balises.
  + Barres PV flottantes.

## **6. Conditions de victoire & rythme**

* **Victoire :** destruction du bastion ennemi.
* **Défaite :** son propre bastion détruit.
* **Durée cible :** 7–12 minutes.

**Classe principale :**

* GameManager → gère boucle principale, transitions de victoire/défaite.