

# **Game Design Document : Introduction à la programmation**

***Projet réalisé par : Samtmann Guillaume***

***Contexte : SAE 501 apprenti DW***

## **Sommaire :**

- 1. Introduction**
- 2. Gameplay**
  - a. Mécaniques principales
  - b. Progression
- 3. Univers et ambiance**
- 4. Musique et son**
- 5. HUD**
- 6. Réalisation**
- 7. Amélioration**
- 8. Inspiration**

# **1. Introduction**

**Titre du jeu :** Finch

- **Genre :** Aventure
- **Plateforme :** PC
- **Moteur de jeu :** Unity
- **Langage utilisé :** C++
- **Modélisation :** Blender
- **Type de jeu :** Vue à la première personne
- **Durée estimée de développement :** Quelques jours

**Pitch :** Inspiré du film *Finch* ([https://www.allocine.fr/film/fichefilm\\_gen\\_cfilm=260221.html](https://www.allocine.fr/film/fichefilm_gen_cfilm=260221.html)).

Dans un monde post-apocalyptique, le joueur incarne un ingénieur, Finch, tentant de reconstruire un robot avancé à l'aide des ressources disponibles dans son laboratoire. Il devra assembler des pièces et comprendre des manuels de programmation pour coder et donner vie à son robot. La fin du jeu est atteinte lorsque le robot est entièrement fonctionnel.

## **2. Gameplay**

### **2.1 Mécaniques principales**

- **Exploration du centre scientifique :** Le joueur doit se déplacer dans un laboratoire abandonné pour récupérer des pièces et des manuels de programmation.
- **Assemblage du robot :** Trouver et assembler correctement les composants électroniques et mécaniques nécessaires.
- **Programmation guidée :** À l'aide des manuels trouvés, le joueur doit écrire du code (simplifié, le langage de programmation sera inspiré de la réalité mais modifié pour être accessible à un large public) pour activer certaines fonctionnalités du robot.
- **Problèmes à résoudre :** Des erreurs de code peuvent empêcher le robot de fonctionner, incitant le joueur à déboguer et tester ses solutions.

### **2.2 Progression**

1. **Tutoriel :** Introduction au centre scientifique (cinématique), premiers composants à récupérer, lecture des manuels de base.
2. **Phase 1 - Assemblage mécanique :** Trouver les différentes parties du robot (corps, circuits, capteurs, moteurs).
3. **Phase 2 - Premiers scripts :** Écrire des instructions simples (ex : faire bouger un bras, allumer un capteur).
4. **Phase 3 - Débogage et amélioration :** Ajouter des conditions, des boucles et optimiser le comportement du robot.

5. **Phase finale - Activation du robot** : Si le robot fonctionne correctement, une cinématique de fin montre sa mise en marche.

### **3. Univers et ambiance**

- Graphismes en low-poly pour un rendu accessible et pour limiter le temps de modélisation des éléments.
- L'interface du côté programmation du robot sera inspirée des terminaux informatiques que nous connaissons.

### **4. Musique et Son**

Musique d'ambiance tout au long du jeu pour renforcer l'immersion.  
Bruitage d'assemblage des pièces, du clavier pour la programmation.

### **5. HUD**

Il n'y aura pas d'inventaire à proprement parler : le joueur pourra prendre une seule pièce à la fois (l'idée d'un inventaire plus complexe pourrait être une amélioration future).

Fenêtre de programmation pour écrire du code.

Aide contextuelle affichant les manuels en cas de blocage (inspirée du manuel de craft de *The Forest*).

### **6. Réalisation**

Le jeu sera entièrement réalisé sur Unity en C++. Afin d'aller plus vite, certains assets seront téléchargés depuis des banques d'assets mises en place par la communauté. Néanmoins, la plupart du projet sera modélisée par mes soins sur Blender.

### **7. Amélioration**

L'une des améliorations principales que j'aimerais apporter à mon jeu de base serait le développement de la ville pour rechercher du matériel. Cela permettrait de créer des mécaniques de survie, d'ajouter plus d'enjeux avec des ennemis et des conditions climatiques.

## 8. Inspiration

