

# Rapport de projet – Coconut-Party

**Filière :** préING1 — 2024-2025.

**Sujet :** Création d'un jeu 2D de type Tower Défense.

**Groupe :** Tiago CHARETTE, Matthieu VANNEREAU, James NDEUGOUE.

## 1. Présentation du sujet et les raisons de notre choix

Le projet "Coconut-Party" a pour objectif de créer un jeu de type Tower Defense, dans un univers humoristique où des singes protègent la couronne de leur roi contre des crabes envahisseurs.

Le jeu se déroule sur une matrice 2D dans laquelle chaque case représente un élément (chemin, singe, crabe, vide, couronne...). L'utilisateur place des singes pour repousser des vagues de crabes. Le joueur peut améliorer ses défenses entre les vagues et doit empêcher les crabes d'atteindre la couronne, ce qui par conséquent mettrait fin à la partie.

Nous avons choisi ce projet car :

- Il offrait un bon équilibre entre challenge technique et faisabilité.
- Il permettait d'explorer des notions fondamentales : gestion de la mémoire, structures de données, logique de boucle principale, et surtout programmation graphique via SDL2.
- Le thème fun et original nous a également motivé à aller plus loin dans le gameplay et l'esthétique du jeu.
- Enfin, ce projet représentait une excellente expérience de travail en équipe, avec gestion de version via GitHub et coordination via Discord et Whatsapp.

## 2. Présentation de l'équipe

Notre groupe est composé de trois étudiants — Tiago CHARETTE, Matthieu VANNEREAU et James NDEUGOUE — qui ont travaillé de manière collaborative, sans attribution de rôles fixes.

Chacun a participé aux différentes étapes du développement (programmation, débogage, conception graphique...), en fonction des besoins du projet et des disponibilités de chacun.

Nous avons travaillé en collaboration continue à l'aide de :

- GitHub pour la gestion du code source
- Whatsapp pour la communication rapide et la planification des séances de travail.
- Discord a été utilisé pour ses fonctionnalités, notamment les appels vidéo, ce qui a considérablement amélioré notre efficacité.

### **3. Suivi hebdomadaire**

**Période : Semaine 1**

**Objectifs : Génération procédurale & déplacement d'ennemis**

#### **Travail effectué :**

- Tests de génération de carte avec différents algorithmes : chemin droit, serpenté, aléatoire.
- Mise en place de la matrice de jeu (2D), marquage visuel du chemin en rouge dans le terminal.
- Déplacement d'un ennemi (crabe) le long du chemin avec mise à jour dynamique de sa position.

#### **Problèmes rencontrés :**

- Chemins parfois non continus ou bloqués.
- La position de l'ennemi ne changeait parfois pas.

#### **Résultats :**

- Génération d'une carte jouable fonctionnelle.
- Ennemi qui suit correctement le chemin sans sortir des limites.

**Période : Semaine 2 Objectifs :**

**Génération améliorée, sons, déplacement des ennemis**

### **Travail effectué :**

- Passage à un **nouvel algorithme de génération procédurale**, plus fluide et structuré.
- Changement du style de carte pour un rendu plus lisible.
- Intégration des effets sonores (tir, déplacement, fin de partie).
- Poursuite du développement du comportement des ennemis (mouvement, interaction avec le chemin).

### **Problèmes rencontrés :**

- Difficulté à intégrer le son sur tous les systèmes (bibliothèques différentes selon l'OS).
- Débogage du déplacement ennemi encore en cours (gestion des collisions et transitions entre cases).

### **Résultats :**

- Génération de cartes plus **variées et fonctionnelles**.
- Premiers **sons** joués avec succès dans le jeu.
- Déplacement ennemi plus fluide, bien intégré au système de boucle.

### **Période : Semaine 3**

#### **Objectifs : Tourelles, chemin, interface graphique**

### **Travail effectué :**

- Création des **premières tourelles** (placement, propriétés de tir).
- Génération plus robuste des **chemins**.
- Début de la mise en place d'un **Framework** pour la création de fenêtres avec SDL2, en vue d'une interface graphique future.

### **Problèmes rencontrés :**

- Le calcul des portées de tir des tourelles n'était pas toujours juste.
- Prise en main de SDL2 lente au départ (structure des fenêtres, événements).

### **Résultats :**

- Tourelles fonctionnelles à placer sur la carte.
- Chemins générés correctement et compatibles avec le système d'ennemis.
- Première fenêtre affichée avec SDL2.

### **Période : Semaine 4**

**Objectifs :** Finalisation du jeu, intégration complète, tests et corrections

### **Travail effectué :**

- **Intégration de tous les modules** (carte, ennemis, tourelles, affichage, sons).
- **Nettoyage du code** : suppression des doublons, uniformisation des structures et noms de variables.
- Utilisation du **Makefile** pour compiler tout le projet facilement.
- **Tests complets du jeu** pour identifier les bugs (crashes, incohérences, erreurs mémoire).

### **Problèmes rencontrés :**

- Conflits entre certaines structures partagées.
- Comportements imprévus dus à des appels concurrents dans les boucles.
- Difficultés à suivre l'origine de certains bugs dus à des dépendances croisées.

### **Résultats :**

- Malgré les difficultés, **tout fonctionne à la fin de cette semaine** : le jeu est **complet, jouable et stable**.
- Tous les modules communiquent correctement, le programme compile sans erreur, et le gameplay est fluide.
- Objectif atteint pour cette dernière semaine : projet terminé avec succès.

Ce projet nous a permis de concevoir un jeu Tower Defense complet et fonctionnel, de l'idée initiale à la version finale. En quatre semaines, nous avons relevé des défis techniques variés : génération procédurale, déplacements d'ennemis, logique de tir, intégration de sons et interface SDL. Malgré les bugs et problèmes d'intégration, nous avons su organiser notre travail, collaborer efficacement et livrer un jeu stable, fluide et fidèle au cahier des charges. Ce projet a été une excellente occasion de renforcer nos compétences en programmation, gestion de projet et travail en équipe