

# Tower Defense

Lilian Briaut et Thomas Lesieux

Septembre - Décembre 2025

## 1 Semaine du 1<sup>er</sup> au 14 septembre

### 1.1 Introduction

Nous avons commencé par lire attentivement le sujet afin d'identifier les étapes clés du projet. Conscients que certaines parties du développement prendront plus de temps que d'autres, nous avons choisi de ne pas attribuer de dates fixes à chaque étape. Nous avons plutôt établi un planning préliminaire structuré en grandes phases, ce qui nous permet de rester flexibles tout en gardant une vision claire de l'avancement.

Pour clarifier nos idées et disposer d'une base de référence, nous avons demandé à Gemini de produire un compte rendu synthétique du sujet. Cela nous permet de nous y référer facilement tout au long du projet.

#### 1.1.1 Interprétation du sujet et recherche

La première étape consiste à interpréter le sujet et à effectuer des recherches sur les solutions existantes, notamment en matière de *pathfinding* (comme l'algorithme A\*) et de jeux de type *Tower Defense* en C++. Ces recherches nous permettront de réutiliser des concepts existants et d'optimiser notre développement.

#### 1.1.2 Prise en main

Après avoir analysé et comparé les différents codes trouvés, nous nous concentrerons sur les bases de SFML : la création de la carte et son quadrillage, ainsi que les interactions entre la carte et la souris. L'objectif est de permettre à l'utilisateur de placer des tours de manière intuitive.

#### 1.1.3 Gameplay

Nous étudierons ensuite les bases du *gameplay*, en particulier :

- les entités (tours et ennemis),
- leur fonctionnement,
- les types de tours et d'ennemis à intégrer.

#### **1.1.4 Affichage**

Nous fusionnerons les éléments développés en SFML pour afficher les tours et les ennemis. Dans un premier temps, nous nous concentrerons sur le placement des tours et le déplacement basique des ennemis, sans gestion des collisions ou des chemins optimisés.

#### **1.1.5 Gestion des vagues**

Nous mettrons en place le système de déplacement des ennemis (*pathfinding*) et l'ajout de collisions.

#### **1.1.6 Gameplay des tourelles**

Nous ajouterons :

- le système d'attaque des tours,
- l'amélioration des tours,
- la création de 4 à 5 tours différentes.

#### **1.1.7 Finitions du gameplay**

Enfin, nous implémenterons :

- le système de points de vie,
- la gestion de l'argent pour l'achat et amélioration des tours,
- les finitions graphiques.

### **1.2 Interprétation du sujet et recherche**

Pour nos recherches, nous avons principalement utilisé GitHub afin de trouver des codes open source dont nous pourrions nous inspirer. Afin d'élargir notre champ d'investigation, nous avons également utilisé l'outil de recherche de Mistral en appliquant des filtres spécifiques. Cela nous a permis d'obtenir une vision globale des solutions existantes.

Bien que les résultats fournis ne soient pas toujours optimaux ou pertinents, cette méthode nous sert de point de départ. Si les informations sont insuffisantes ou non adaptées, nous complétons nous-mêmes les recherches. Cette approche nous permet de gagner du temps tout en garantissant la qualité des ressources sélectionnées.

Vous trouverez le rapport détaillé de ces recherches à l'adresse suivante : [/IA/rapport\\_mistral.pdf](#).