

---

# Cahier des charges

---

## Objectif

Réaliser une version en C++ du jeu de société sur plateau *Tac-Tik*.

## Description du Jeu

### Présentation du Jeu *Tac-Tik*

*Tac-Tik* est un jeu de plateau stratégique, par équipes de 2, se jouant à 4 ou 6 joueurs où les joueurs doivent utiliser des cartes pour faire sortir leurs 4 pions de leur réserve et les faire avancer sur le parcours de cases, afin de les faire rentrer dans leur Maison.

### Règles du Jeu (Voir annexe)

- ❖ Chaque joueur possède 4 pions de la même couleur : bleu, jaune, vert, rouge (noir et blanc).
- ❖ Une main est constituée de 4 cartes, chaque joueur en défaut une à tour de rôle.
- ❖ Au début d'une manche, les joueurs d'une même équipe doivent s'échanger une carte.
- ❖ Les pions se déplacent selon les cartes jouées.
- ❖ Il est possible de capturer ou de bloquer les pions adverses.
- ❖ La victoire revient à la première équipe qui parvient à remplir ses Maisons.

### Modes de Jeu

- ❖ Solo contre l'IA.
- ❖ Multijoueur local.
- ❖ Possibilité d'une version en ligne ultérieure (optionnel).

## Fonctionnalités

- ❖ Reproduire les mécaniques du jeu *Tac-Tik* de manière fidèle.
- ❖ Implémenter une interface graphique pour la visualisation du plateau et des déplacements des pions.
- ❖ Permettre un mode de jeu contre une IA ou en multijoueur local, voire en ligne (optionnel).

## Solutions techniques

### **Contraintes**

- ❖ Le jeu doit fonctionner sous Windows et Linux.
- ❖ Performances optimisées pour un rendu fluide en 2D.
- ❖ Documentation du code et commentaires clairs.

### **Interface Graphique**

- ❖ Utilisation de bibliothèques graphiques C++.
- ❖ Affichage du plateau de jeu avec cases et pions.
- ❖ Indicateurs visuels pour les actions possibles.
- ❖ Interface minimalistique et intuitive.

### **Mécaniques du Jeu**

- ❖ Utilisation de la Programmation Orientée Objet (cf. diagramme des classes).
- ❖ Gestion des pions et des règles du jeu.
- ❖ Système de tours et d'actions valides.
- ❖ Gestion des événements de capture et des conditions de victoire.
- ❖ Implémentation d'une IA avec des algorithmes heuristiques.

## Moyens à mettre en œuvre

- ❖ C++
- ❖ SDL2
- ❖ Diagramme des classes (cf. annexe)
- ❖ Diagramme de Gantt (cf. annexe)
- ❖ GitLab
- ❖ Editeur de code (VSCode, CLion)
- ❖ Débogueur : gdb
- ❖ Profiler : valgrind
- ❖ Ressources humaines : Marius et Valentin

# Planification du Projet

## Analyse et conception

- ❖ Étude des règles du jeu et formalisation.
- ❖ Choix des technologies et des bibliothèques.

## Développement du moteur de jeu

- ❖ Gestion du plateau, des pions et des règles.
- ❖ Implémentation du système de tour et d'actions.

## Ajout de l'interface graphique

- ❖ Affichage du plateau et des animations.
- ❖ Ajout des interactions utilisateur.

## Implémentation de l'IA et tests

- ❖ Développement d'une IA simple.
- ❖ Tests et corrections.

## Finalisation et optimisation

- ❖ Correction des bugs et amélioration des performances.
- ❖ Documentation du projet.

## Délais

Finir avant les vacances scolaires

Voir diagramme de Gantt



**Ciserane Marius (Chef d'équipe) & Laporte Valentin**

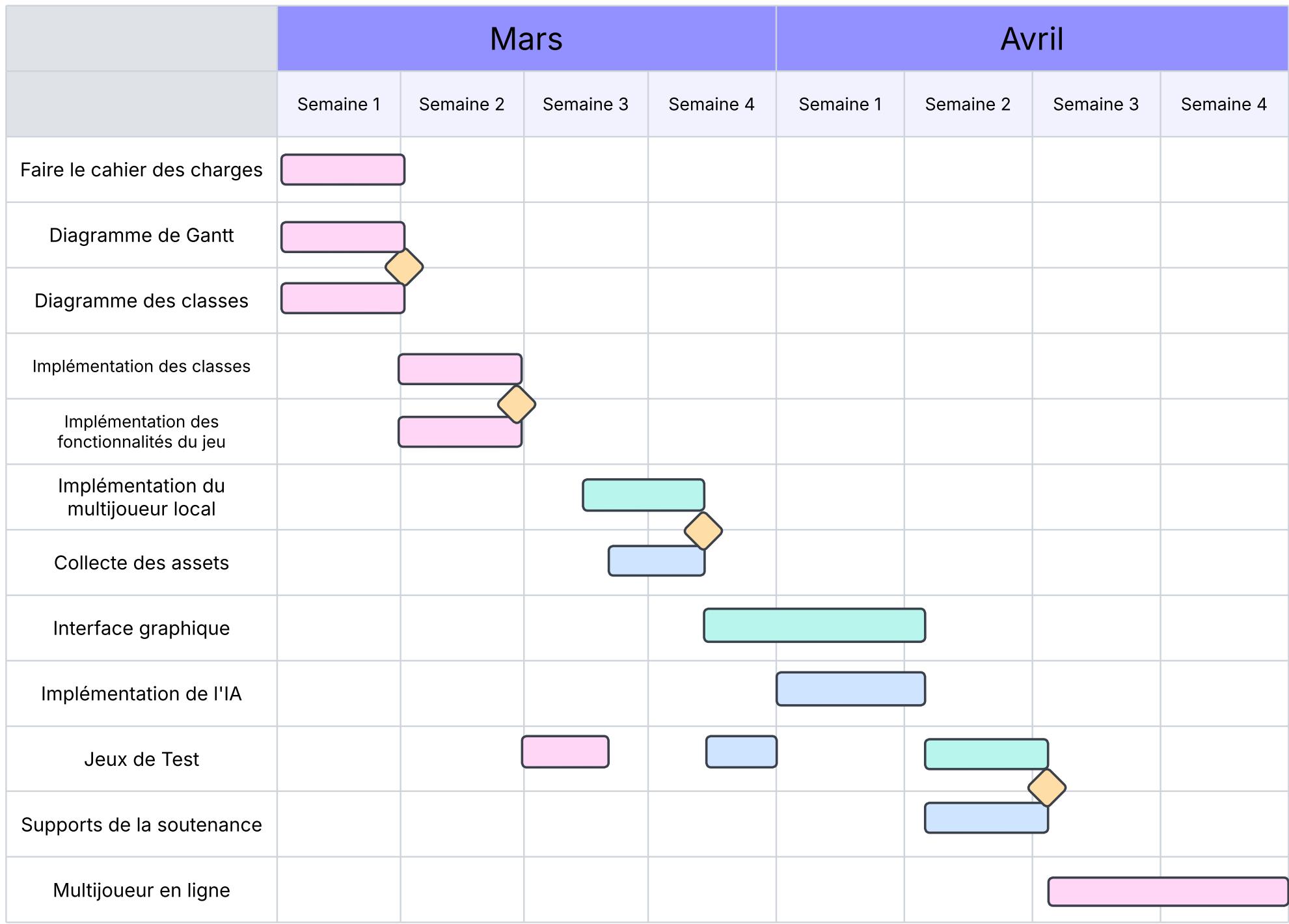
Projet LIFAPCD – Cahier des charges

Page 3/3

# Diagramme de Gantt pour le projet LIFAPCD : Tac-Tik

## Légende

- Équipe
- Marius
- Valentin
- ◆ Objectif



# Diagramme des classes pour le projet LIFAPCD : Tac-Tik

