

---

# 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**

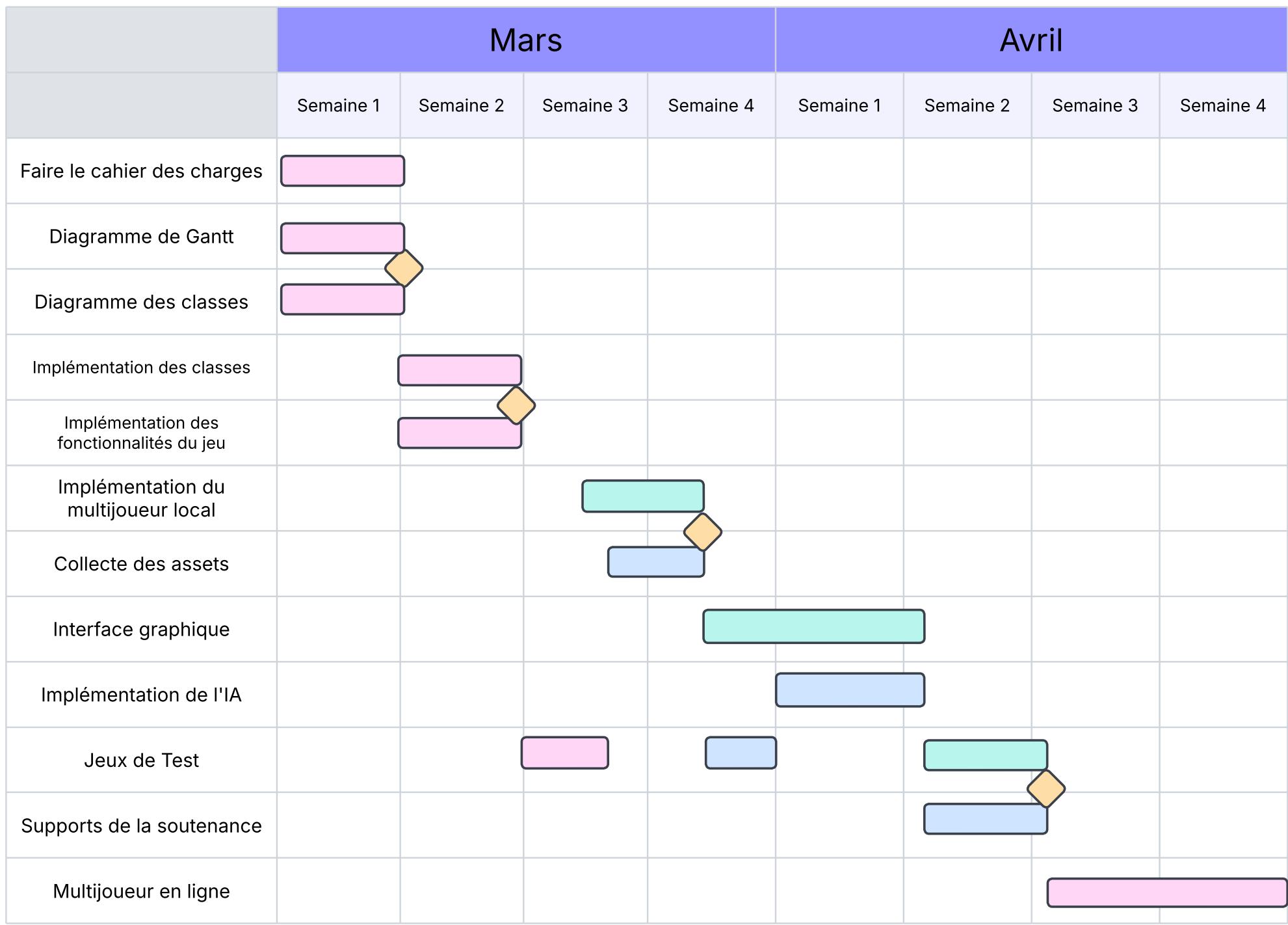
Projet LIFAPCD – Cahier des charges

Page 3/3

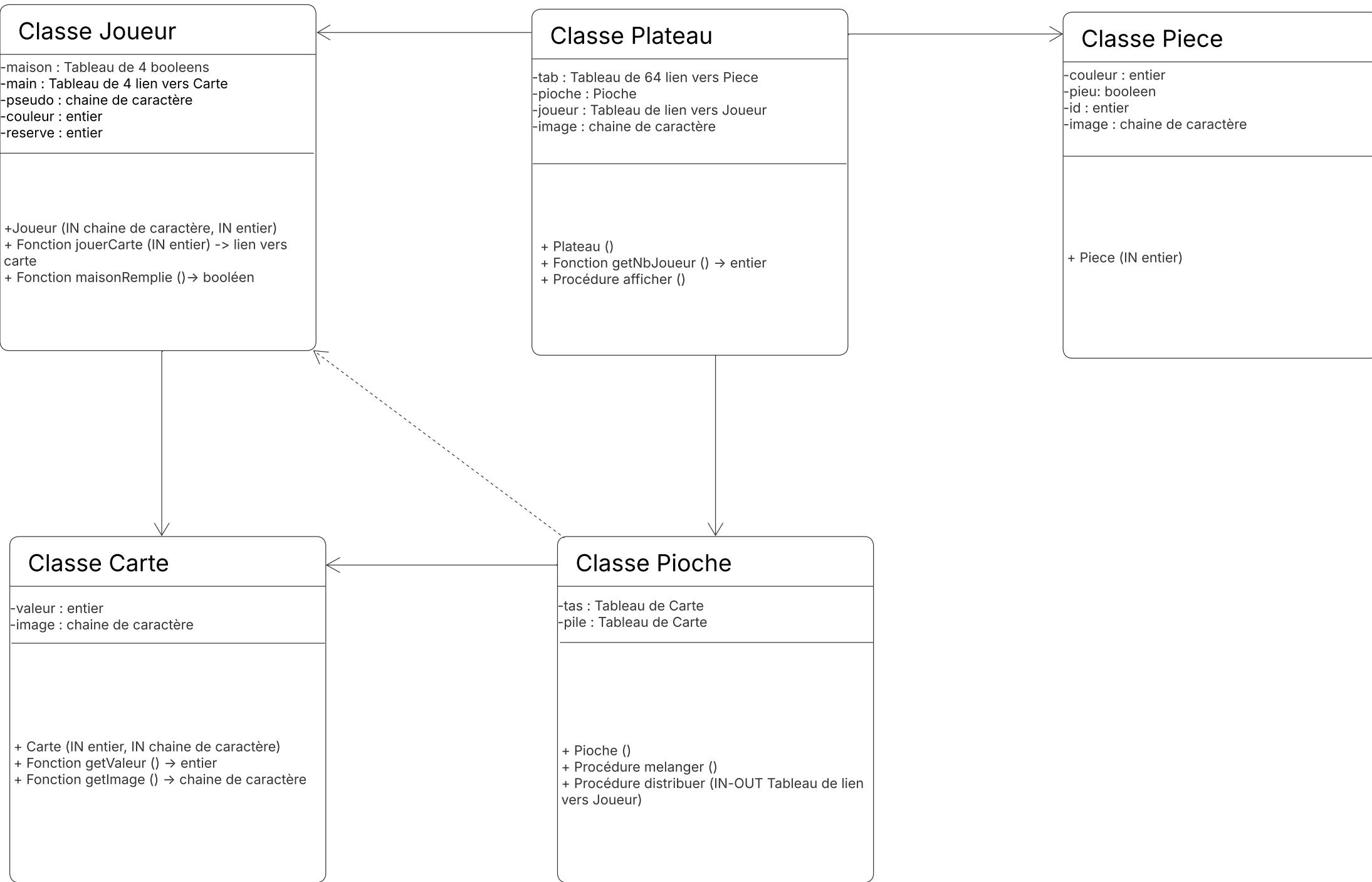
# Diagramme de Gantt pour le projet LIFAPCD : Tac-Tik

## Légende

- Équipe
- Marius
- Valentin
- ◆ Objectif



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



Âge 6+

2 à 6  
joueurs

# TAC-TIK

## Règles du jeu



[jeux-de-cartes-personnalises.fr](http://jeux-de-cartes-personnalises.fr)



## **But du jeu et début de partie**

Faire sortir (démarrer) et déplacer ses 4 pions sur le parcours de cases, dans le sens horaire, afin de les rentrer dans sa *Maison*. Les joueurs déplacent leurs pions en jouant des cartes. Chaque joueur pioche 4 cartes en début de partie et en utilise une à chaque fois que vient son tour. Lorsque tous les joueurs ont joué toutes leurs cartes, ils en piochent de nouvelles. Quand le talon est épuisé, on mélange les cartes et on distribue à nouveau. Les joueurs peuvent former des équipes de deux joueurs qui se font face. Les partenaires s'échangent une carte face cachée après chaque distribution de cartes.

## **Réserve**

Au début de la partie, les pions sont placés dans leur *Réserve* (de même couleur que les pions).

## **Démarrer**

Pour commencer à déplacer un pion, il faut jouer une carte *Démarrer* et prendre un pion dans sa *Réserve* pour le placer sur la case *Démarrer* (Case qui est entourée de la couleur du pion). Le pion qui démarre devient un *Pieu* (perforation visible).

## **Maison**

C'est l'arrivée, représentée par 4 cases de couleur :



## **Pieu**

Un pion qui démarre est un *Pieu* et il doit être posé sur la case *Démarrer* (entourée de sa couleur) avec sa partie percée visible. Il demeure un *Pieu* tant qu'il n'est pas déplacé ou permué. Un *Pieu* ne peut pas rentrer dans sa *Maison*. Il ne peut pas être mangé, dépassé et bloque le passage de tous les autres pions. Il peut être permué uniquement par son propriétaire. Lorsqu'un *Pieu* est déplacé ou permué, il devient un pion. Il est alors retourné (face percée vers le bas).

## **Dépasser**

Un pion peut en dépasser un autre sauf si ce dernier est un *Pieu* ou s'il se trouve dans sa *Maison*.

## **Rentrer dans sa Maison**

Un pion rentre dans sa *Maison* s'il finit son déplacement sur une des 4 cases de sa couleur. Lorsque la valeur de la carte est supérieure au nombre nécessaire pour arriver dans la maison, le pion n'entre pas dans la maison et doit alors continuer d'avancer jusqu'à l'utilisation complète de la valeur de la carte. Ce pion devra refaire un tour pour rentrer dans sa maison... Un pion dans sa *Maison* ne peut pas être mangé, dépassé, permué ou ressortir. Chaque pion doit donc, être déplacé vers le fond, pour que les autres puissent rentrer.

Lorsqu'un joueur a rentré tous ses pions, il continue à jouer en déplaçant les pions de son partenaire.

## **Sauter (ou manger) un pion**

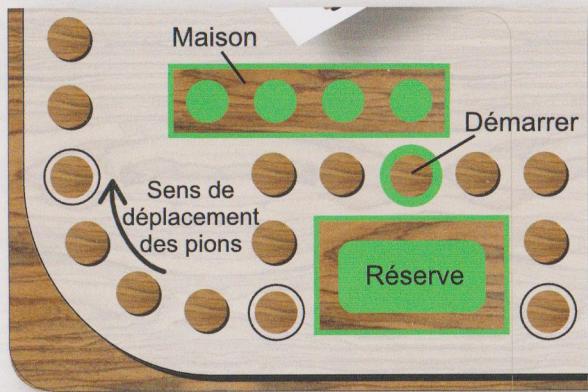
Lorsqu'un pion finit son tour sur un autre pion, ce dernier est sauté (ou mangé) et il est replacé dans sa *Réserve*.

## **Variante pour 2 joueurs**

Pour jouer à 2, vous pouvez piocher 6 cartes et prendre 2 couleurs chacun.

## Les cases

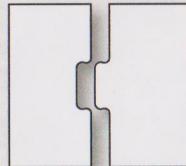
Les pions se déplacent sur les cases rondes dans le sens horaire



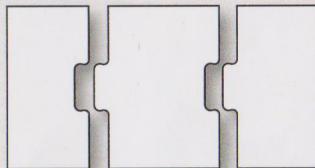
## Le plateau

Emboitez les parties du plateau selon le nombre de joueurs

2 à 4 joueurs



6 joueurs



## Les cartes

La valeur d'une carte doit être utilisée entièrement. Un joueur est toujours obligé de jouer une carte quand vient son tour. Mais s'il ne peut jouer aucune de ses cartes, il se défausse et passe son tour.

### Avancer 2, 3, 5, 6, 8, 9 et 12

Ces cartes permettent de déplacer un pion de la valeur indiquée dans le sens horaire.



### Règles du jeu & Accessoires

Jeux, Pions, Cartes, Sacs et Extensions de plateau Tac-Tik pour 8 et 10 joueurs

[jeux-de-cartes-personnalises.fr](http://jeux-de-cartes-personnalises.fr)



### Avancer ou Démarrer 1 et 10

Ces cartes permettent soit de démarrer un pion (placer un pieu sur la case *Démarrer* qui est entourée de sa couleur) soit de déplacer un pion de la valeur indiquée dans le sens horaire.



### Avancer 7 x 1

Cette carte permet de déplacer un ou plusieurs de ses pions d'une case jusqu'à un total de 7 cases dans le sens horaire. Cette carte permet de sauter (manger) plusieurs pions. La valeur de cette carte ne se partage pas entre des pions de couleurs différentes.



### Reculer -4

Cette carte permet de déplacer un pion dans le sens inverse des aiguilles d'une montre de 4 cases. Elle ne permet pas à un pion de rentrer directement dans sa *Maison*. Elle permet de reculer un pieu qui devient un pion (retournez-le). Elle permet à un pion ou à un *Pieu*, de sauter un pion se trouvant 4 cases derrière lui.



### Permuter

Cette carte permet de permute un de ses pions avec un autre pion, à condition que les deux se trouvent déjà sur le parcours et pas dans les *Maisons*.

Permet aussi de permute son pieu avec un de ses pions qui se trouvent sur le parcours.



### Joker

Cette carte compte comme une autre carte au choix du joueur.