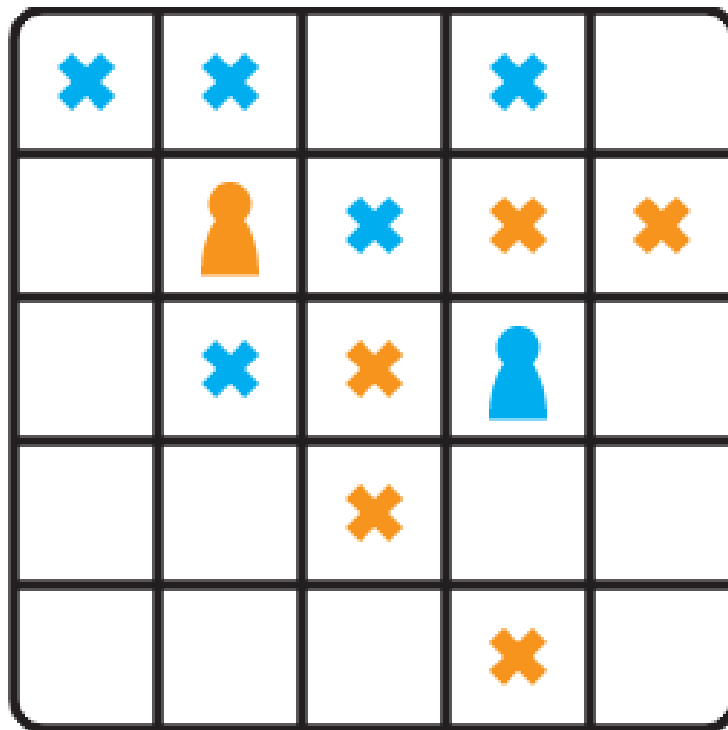


# BLOCUS - Projet APL

## Décembre 2020

Yannis SAGAL - Maxime Hauet

---



### Introduction

Notre mission était de concevoir le jeu du Blocus en le codant de A à Z en C89. Notre jeu ne naquit pas en un jour, mais aujourd'hui, après des heures de travail malgré ces temps difficiles nous avons fini notre jeu.

---

---

## Présentation & Organisation du code

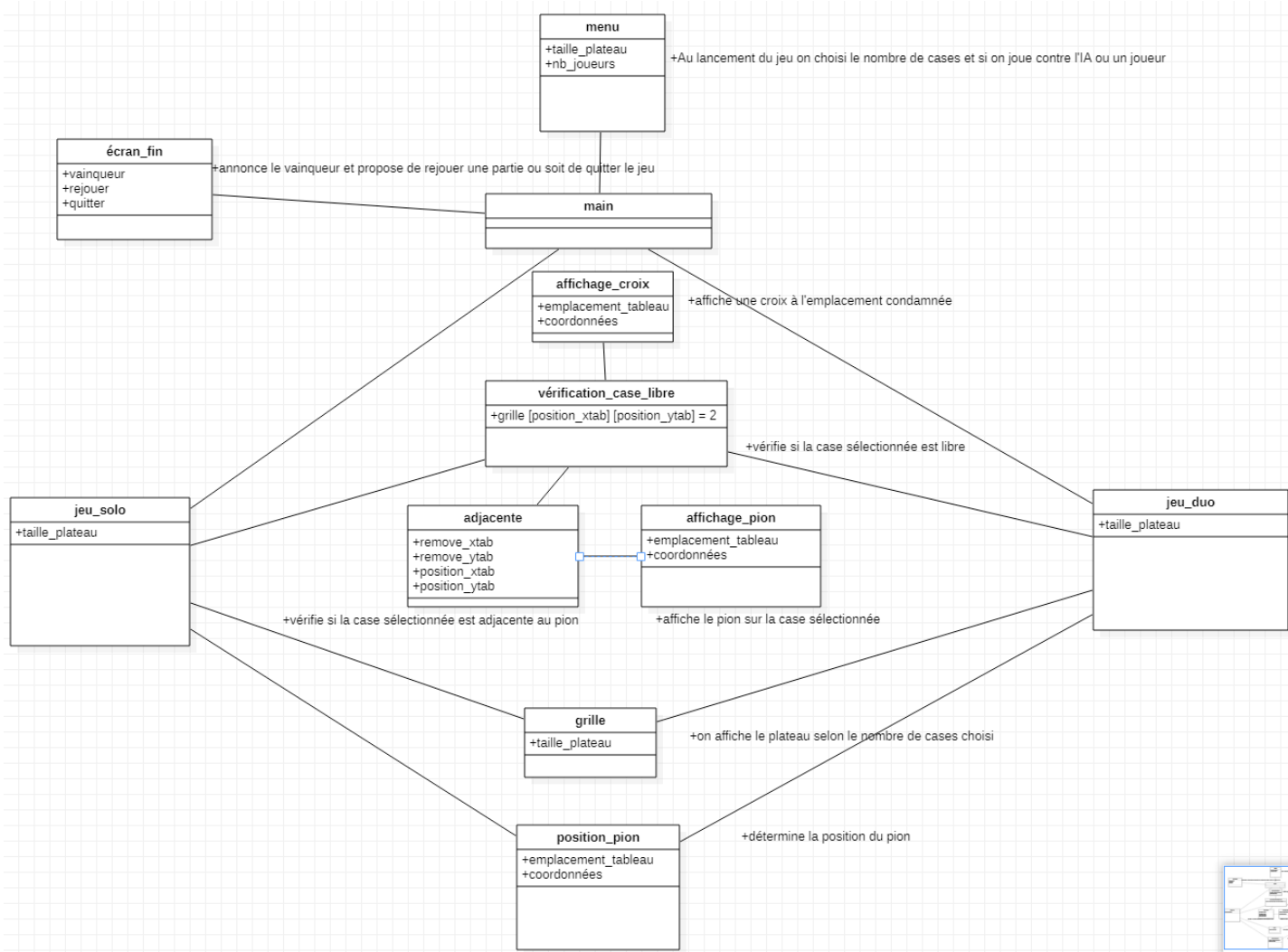
Nous avons d'abord pensé notre Blocus, nous avons décidé de séparer tout notre code en plusieurs fichier, tout en restant cependant organisé dans notre vision du jeu.

Nous avons donc créé un Makefile qui permet grâce à la commande Makerun de compiler tout le programme.

Pour l'aspect graphique, nous avons créé la grille en C et pour ce qui est des pions, des croix et des flèches nous les avons importés sous formes d'image (jpeg et png).

## Structure du code

Voici la structure de notre programme sous la forme d'un diagramme :



---

## Description des fonctionnalités

Tout d'abord nous avons défini la largeur et hauteur du plateau.

Premièrement, tout le plateau est représenté par un tableau multidimensionnel rempli d'entiers. Au départ, on initialise chaque case du tableau à 2. 2 étant l'état représentant une case vide, 1 une case avec un pion dessus et 0 une case bloquée.

On utilise la Fonction `SourisCliquee()` pour récupérer les coordonnées du clique. À l'aide d'un calcul, on détermine l'emplacement dans le tableau multidimensionnel (calcul basé sur la taille d'une case et le décalage x/y du plateau avec les bordures de la fenêtre) et si cette case est libre ( $==2$ ).

Ensuite on test si la case est adjacente au pion grâce à la fonction "adjacente". Si la case est adjacente et libre, on pose le pion. Les seules restrictions pour le blocage d'une case sont que la case soit libre et dans les limites du plateau. Pour le début de la partie, j'ai créé des conditions qui font que les pions doivent partir obligatoirement de la première ligne (ligne du bas pour le pion blanc et ligne du haut pour le pion noir).

Le lancement du jeu se déroule comme ceci, on affiche l'écran d'accueil, on choisit la taille et contre qui on joue. On appui sur le bouton "start". On ferme l'écran d'accueil. Ensuite l'écran du jeu s'affiche, le jeu se déroule comme ceci : Joueur 1 pose son pion, joueur 1 pose sa croix, joueur 2 pose son pion, joueur 2 pose sa croix.

Quand un joueur à gagner on ferme l'écran du jeu pour ouvrir l'écran de fin dans lequel on affiche le nom du vainqueur (joueur 1/joueur2(le bot s'appelle aussi joueur 2)).

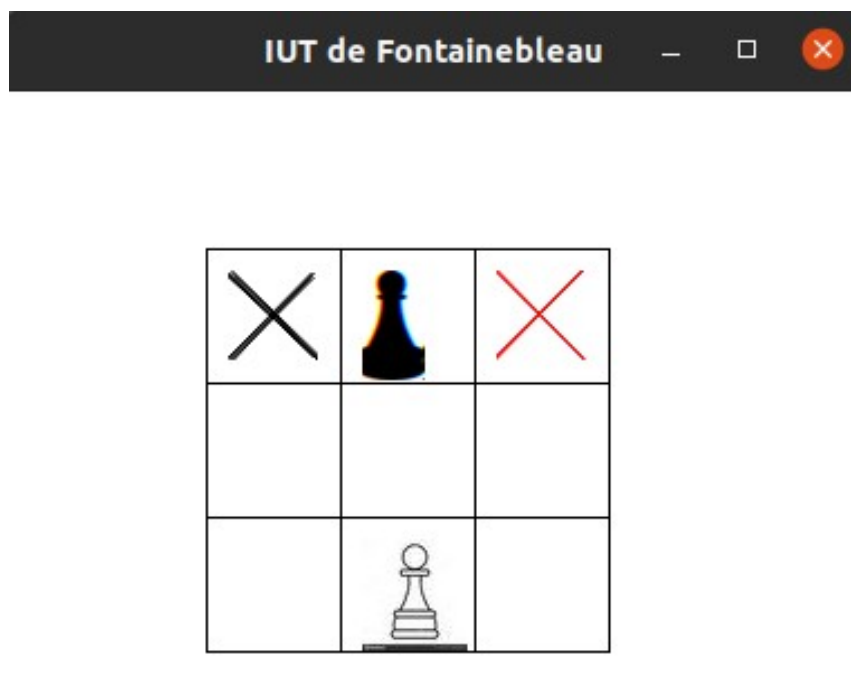
On propose ensuite soit de rejouer, soit de quitter.

## Représentation graphique du jeu

### Ecran du menu principal

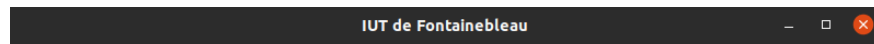


**Ecran d'une partie en cour (taille 3x3)**



---

## Ecran de fin lorsqu'un joueur gagne la partie



Victoire du joueur 1

rejouer

quitter

---

## Conclusion

**Yannis** : Pour ma part, je me suis focalisé principalement sur la façon de repérer le pion sur le plateau et pour ça j'ai directement pensé au tableau multidimensionnel. Malheureusement je ne sais pas pourquoi je ne pouvais plus utiliser git mais je pouvais toujours envoyer mon travail à Maxime et je l'en remercie. Malgré la relation que j'ai avec le C89 j'ai pris du plaisir à travailler sur ce projet avec Maxime, je suis fier de notre travail, et heureux d'avoir eu l'occasion de travailler avec lui.

**Maxime** : Pour ma part, ce projet m'a aidé à bien maîtriser les bases du c. En effet, nous avons entièrement codé sans nous servir d'allocations de mémoire, de listes chaînées, de structures... J'ai enfin vraiment compris comment fonctionne le Makefile. Je pense néanmoins que j'aurais pu rendre mon code beaucoup plus propre. Peut-être créer plus de fonctions... En tout cas je suis content d'avoir fait ce projet puisqu'il m'a permis de me familiariser avec le langage et de voir ce qu'on pouvait concrètement faire.