



BLOCUS

Aléxia BERNARD & Florian CUNSOLO



15 JANVIER 2021

Groupe 2

Table des matières

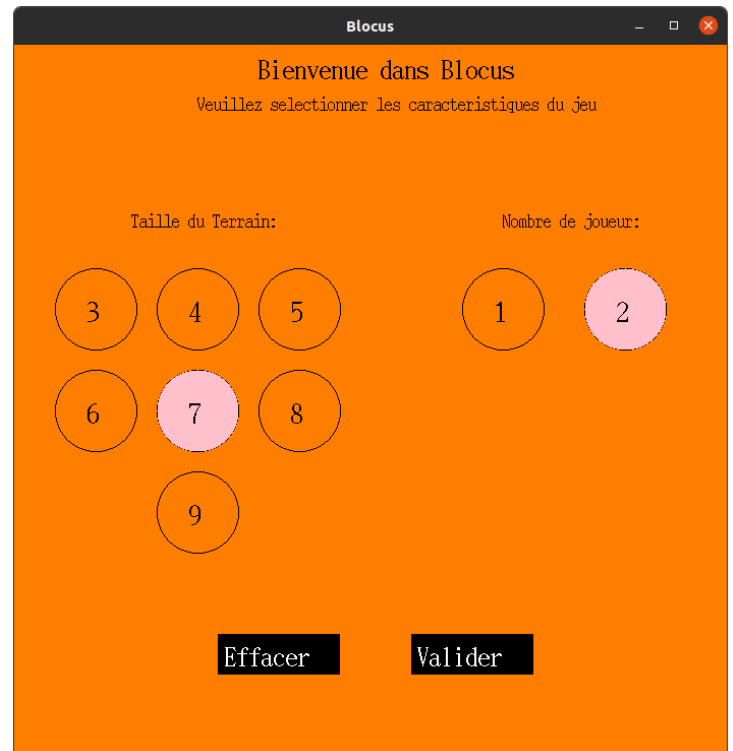
Introduction :-----	2
Description : -----	2
1 : Le premier écran -----	2
2 : Le deuxième écran-----	2
3 : Le troisième écran -----	3
Présentation :-----	3
Explication :-----	4
Conclusion Florian :-----	5
Conclusion Aléxia :-----	5

Introduction :

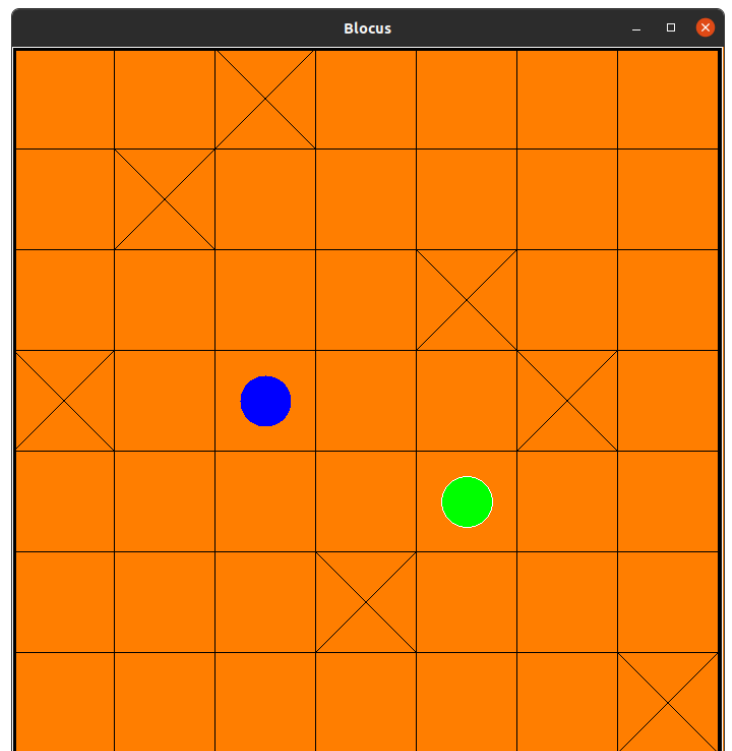
Le projet qui nous a été donné est la réalisation d'un jeu « Blocus », consistant à déplacer deux pions sur une grille pour ensuite tenter de bloquer le pion adverse tout essayant de ne pas se faire bloquer et cela uniquement à l'aide de la souris. Nous avons utilisé uniquement la bibliothèque graphique de l'IUT ainsi que du C89.

Description :

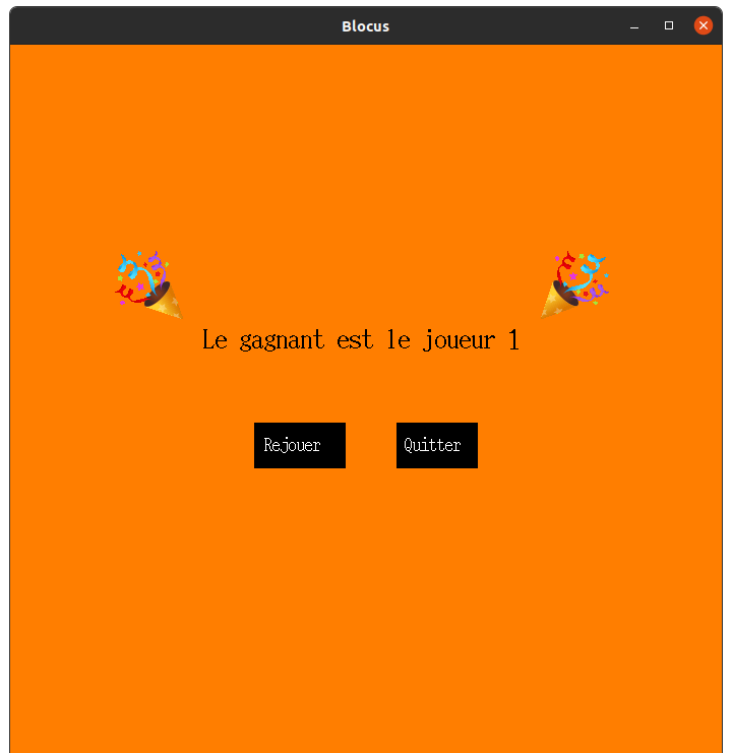
1 : Le premier écran permet au(x) joueur(s) de choisir la taille de la grille et le nombre de joueurs, une fois ces deux caractéristiques soigneusement choisies il suffit de cliquer sur le bouton valider afin de passer à l'écran suivant.



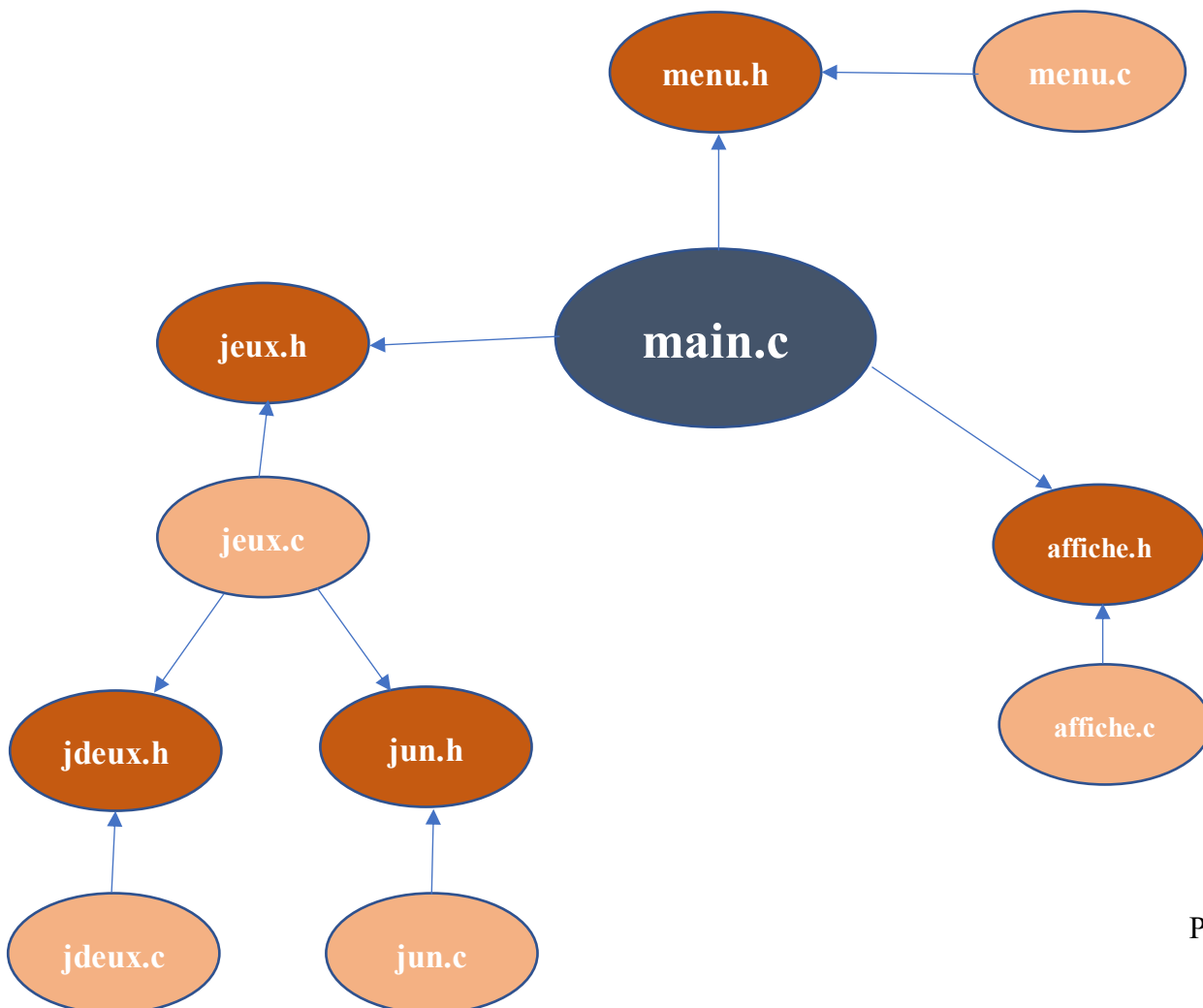
2 : Le deuxième écran est le terrain de jeu, créé en fonction de la taille de la grille choisie au premier écran. Le joueur 1 pose en premier son pion laissant ensuite le joueur 2 faire de même. La partie peut ensuite enfin démarrer : le premier joueur déplace son pion dans une case autour de ce dernier et libre puis place une croix dans n'importe quelle case libre du terrain, c'est maintenant au tour du deuxième joueur de faire de même. La partie se termine donc lorsque l'un des deux joueurs est bloqué (entouré de cases occupées) et passe au troisième écran.



3 : Le troisième écran affiche le gagnant et permet au(x) joueur(s) de rejouer ou de quitter le jeu.



Présentation :



Explication :

Nous allons maintenant vous expliquer chaque fichier et fonctions. Nous dirons, durant ces explications, qu'il n'y a qu'un seul utilisateur par simplification. Nous utilisons dans toutes ses fonctions, exceptées le main, une variable appelée c permettant de choisir puis d'y stocker une couleur.

Tout d'abord le fichier main.c initialise le graphique et crée la fenêtre « Blocus » où va s'afficher les différents écrans. La première variable utilisée est n (initialisée à 1) qui permet de savoir quand arrêter le jeu. Ensuite nous avons fait un appel à la fonction menu (dans le fichier menu.c) qui permet d'afficher le premier écran et renvoie une valeur, stockée dans x, qui nous permet dans la suite de déterminer la taille et le nombre de joueur choisit. Cette valeur est donc ensuite analysée afin d'en sortir la taille et le nombre de joueur, stocké dans joueur. Une fois ces valeurs sorties nous appelons la fonction jeux (dans le fichier jeux.c) qui prend en argument la taille et le nombre de joueur. Cette fonction permet d'afficher l'écran jeux et de permettre à l'utilisateur de jouer. Cette fonction renvoie le gagnant de la partie, stocké dans la variable « gagnant », qui est dans la suite mise en argument, avec le nombre de joueur, à la fonction affiche (dans le fichier affiche.c). Cette fonction renvoie un entier, stocké dans n, qui permet de savoir si l'utilisateur veut rejouer ou quitter le jeu.

Revenons sur le fichier menu.c. Tout d'abord dans ce fichier il existe deux fonctions. Nous allons vous les expliquer l'une après l'autre.

Commençons par DessinerScene1. Cette fonction prend en argument un entier (allant de 1 à 9) stocké dans la variable action, permettant d'entrer dans la boucle correspondante et de mettre un bouton de taille, ou de nombre de joueur, en question de couleur rose lorsque l'utilisateur clique. Cette fonction ne renvoie rien.

Passons maintenant à la fonction principale : menu, appelée dans main.c. Cette fonction ne prend rien en argument. Nous commençons par mettre une image de fond, écrire le nom du jeu ainsi qu'une explication pour l'utilisateur. Nous dessinons ensuite grâce à DessinerArc les boutons tailles que nous numérotions juste après. Nous faisons de même pour le nombre de joueur et des boutons effacer et valider. Ensuite une longue boucle contenant plusieurs conditions nous a permis de faire en sorte que l'utilisateur puisse choisir sa taille et son nombre de joueur en cliquant à plusieurs reprises. En cliquant dans un bouton, cela appelle donc la fonction DessinerScene1 avec en argument le numéro de l'action cliquée et met la variable taille ou joueur, en fonction du bouton cliqué, à la valeur sélectionnée. En revanche, si l'utilisateur appuie sur le bouton Effacer, cela enlèvera tous ses clics. Le bouton valider quant à lui va renvoyer une valeur en fonction des valeurs de taille et joueur, mais cela uniquement si l'utilisateur a choisi une taille et un nombre de joueur.

Passons maintenant au fichier jeux.c. Dans ce fichier il n'y a qu'une seule et unique fonction, appelée dans le main, jeux. Cette fonction prend en argument la taille et le nombre de joueur. Elle dessine tout d'abord un contour blanc et noir à la grille. Ensuite nous testons le nombre de joueur, stocké dans joueur. Si le nombre de joueur est 1 alors nous appelons la fonction un_joueur (dans le fichier jun.c), avec pour argument la taille, et qui renvoie le joueur gagnant stocké dans la variable « gagnant ». Au contraire si le

nombre de joueur est 2 alors nous appelons la fonction `deux_joueurs` (dans le fichier `jdeux.c`), avec pour argument la taille, et qui renvoie le joueur gagnant stocké dans la variable « gagnant ». La fonction `jeux` est ensuite finie et renvoie la valeur de gagnant (le joueur gagnant).

Penchons-nous maintenant sur le sujet du fichier `jun.c` et `deux.c`. Dans chacun de ces fichiers existent deux fonctions, `DessinerSceneUN` et `un_joueur` ou `DessinerSceneDEUX` et `deux_joueurs`.

Commençons par `DessinerSceneUN` et `DessinerSceneDEUX`. Ces fonctions, très similaires) prennent en argument un pion (1 à 4) qui correspond pour 1 et 3 au pion du joueur 1 et 2 et 4 au pion du joueur 2, ainsi que des coordonnées `x` et `y` et la taille de la grille. Par la suite les fonctions testent la valeur de pion. Si pion est égal à 1 (2 respectivement) alors la fonction va permettre d'afficher un rond bleu (vert respectivement) à l'emplacement spécifié en argument. Si pion est égal à 3 (4 respectivement) alors la fonction va permettre d'afficher un cercle blanc à l'emplacement spécifié en argument. Cette fonction ne renvoie rien.

Passons maintenant aux fonctions principales `un_joueur` et `deux_joueurs`, appelées dans `jeux.c`, qui prennent en argument la taille de la grille. Tout d'abord `tab[][]` va permettre de stocker les valeurs de chaque case en les numérotant et `occupe[][]` va permettre de stocker les valeurs de chaque case occupée. La variable `i` (`k` réciproquement) nous permet de stocker les nouveaux numéros de case du pion 1 (2 réciproquement) lorsque le pion 1 (2 réciproquement) bouge. Pour le robot, uniquement dans la fonction `un_joueur`, nous avons décidé de faire appel à une variable aléatoire qui va choisir les déplacements du pion 2 et donc stocker la position dans `k`. Si l'un des deux joueurs ne peut plus bouger (les cases adjacentes sont occupées) alors la fonction renvoie le gagnant.

Pour finir nous allons vous expliquer les deux fonctions du fichier `affiche.c`.

La première fonction est `DessinerScene2` qui prend en argument une action. Cette fonction permet de mettre l'écriture du bouton cliqué, rejouer ou quitter, en rouge. Cette fonction ne renvoie rien.

La principale fonction de ce fichier est `affiche` (appelée dans le `main`) qui prend en argument le gagnant et le nombre de joueur. Cette fonction permet d'afficher le gagnant de la partie. Puis elle permet de choisir de rejouer ou de quitter. La fonction renvoie une valeur en fonction du choix de l'utilisateur (rejouer ou quitter).

Conclusion Florian :

Ce projet m'a permis d'appliquer les connaissances que j'ai pu apprendre en cours et de solidifier certains aspects du C89. Avec Aléxia, nous avons utilisé au maximum la bibliothèque graphique mise à disposition par l'IUT ; en effet il s'agit de ma première utilisation d'une bibliothèque graphique.

Malgré certaines erreurs qui nous ont fait perdre beaucoup de temps pour pas grand-chose, avoir eu l'occasion de travailler en duo est une excellente expérience puisque nous avançons beaucoup plus vite avec un partage d'idées et une autre vision des mêmes connaissances.

Conclusion Aléxia :

Pour conclure, je dirais que malgré les nombreuses difficultés pointilleuses, tel qu'un manque de parenthèse ou même d'un 4 à la place d'un 1 qui faisait un core dumped, j'ai apprécié faire ce programme. Je ne pensais pas au début en voyant l'énoncé être capable de faire un blocus. Ce projet m'a permis d'apprendre à utiliser la bibliothèque graphique de l'IUT mais également m'a permis d'améliorer ma réflexion et recherche en petites fautes de frappes glissées par-ci par-là. J'ai également apprécié travailler en équipe car cela m'a permis de voir le point de vue, parfois différent, de mon binôme et de s'entre-aider.