

Ce projet est un jeu de type "Blocus".

Le terrain de jeu est une grille de taille 3x3 à 9x9 choisie en début de partie, le but du jeu est de faire en sorte que l'adversaire ne puisse plus bouger son joueur, on peut jouer à deux ou seul contre un robot.

Les joueurs jouent à tour de rôle, un tour est divisé en deux parties :

- _d'abord le joueur place son pion dans une case adjacente (diagonale comprise).

- _Ensuite, le joueur choisit une case à bloquer dans toute la grille, qui ne sera plus accessible par n'importe quel joueur.

Et ainsi de suite, jusqu'à qu'un joueur ne puisse plus bouger.

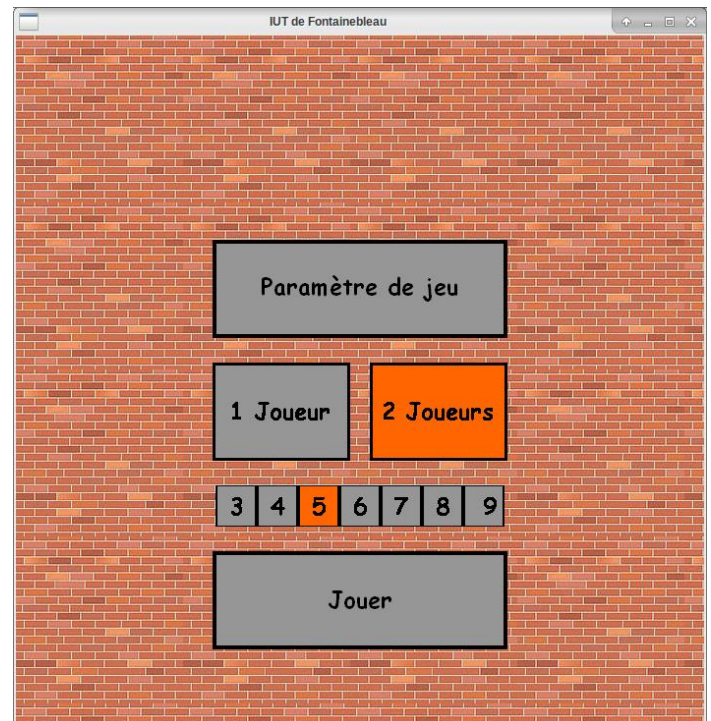
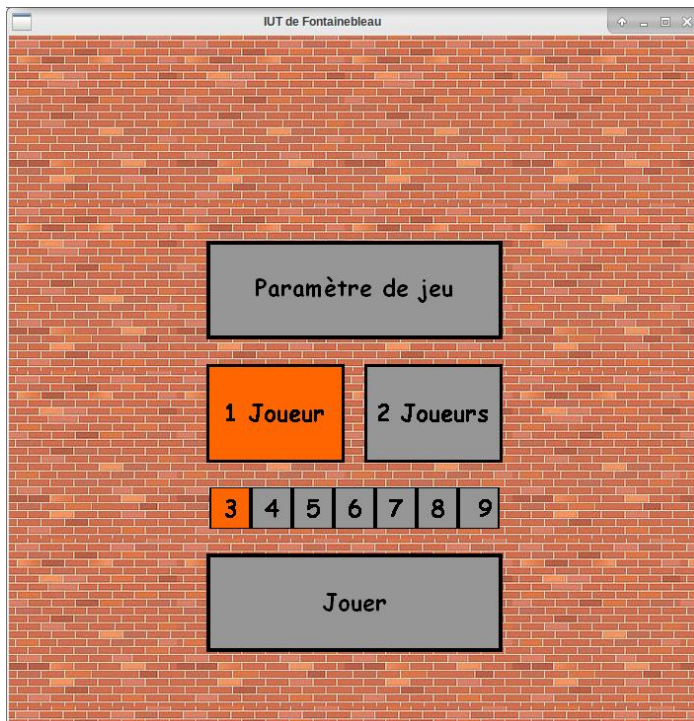
À la fin de la partie, le gagnant est affiché et deux choix nous sont proposés soit recommencer une partie ou quitter le jeu.

Description de fonctionnalité:

Au début du programme, les paramètres de jeu nous sont demandés.

Par défaut, le nombre de joueurs est de 1 et la taille de la grille est de 3.

Le choix voulu est représenté avec la couleur orange.



Paramètre: ici: 1 joueur et une grille de 3x3 ici: 2 joueurs et une grille de 5x5

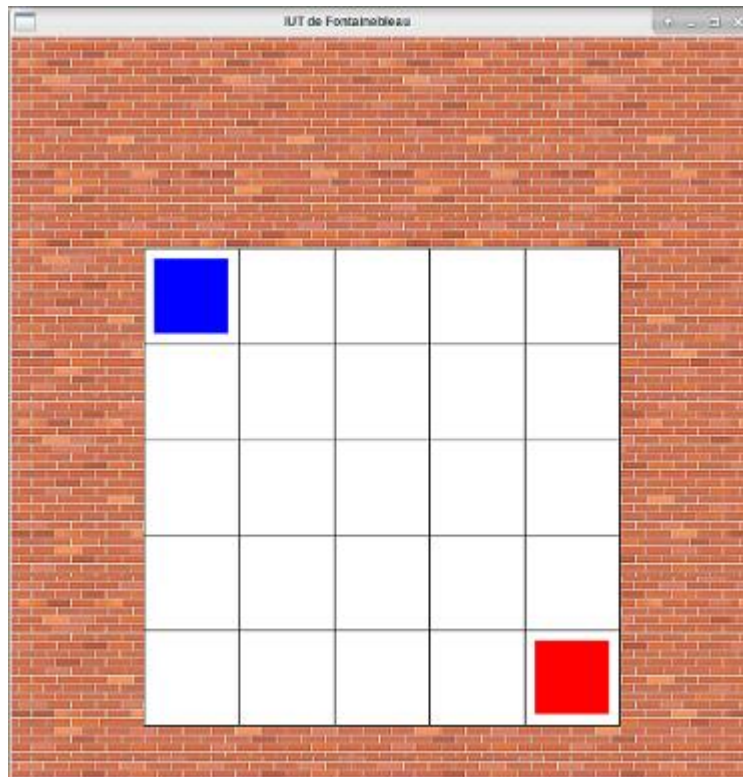
Pour valider les paramètres, il faut cliquer sur jouer.

Ensuite, on le terrain de jeu est affiché (la taille de la grille varie selon les paramètres) et le jeu commence.

Le joueur 1 est le carré bleu et le joueur 2 est le carré rouge, ils sont placés aux extrémités en début de partie.

C'est le joueur 1 (bleu) qui commence la partie.

Si il n'y a qu'un joueur, le robot est le joueur 1 (bleu).

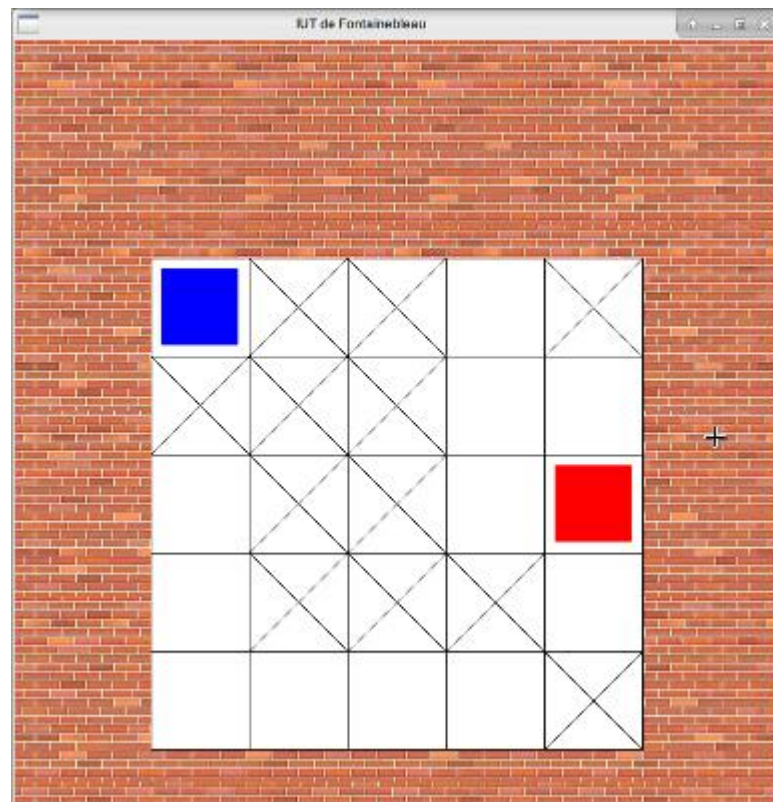
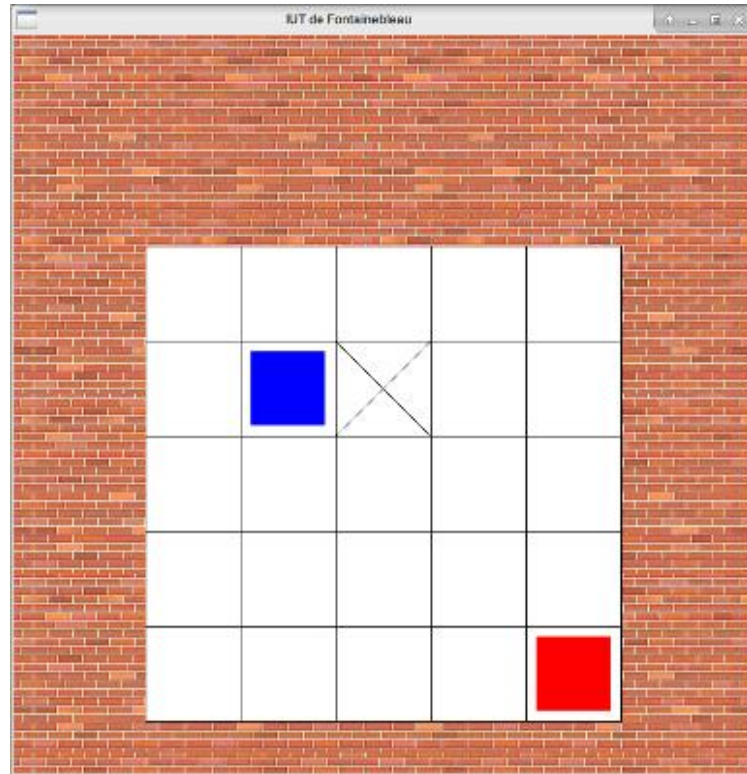


Début de la partie

Au cours de la partie, un joueur peut se déplacer que dans les case adjacente, puis condamner une case aux choix.

Dans l'image, le joueur 1 a déplacé son pion et
et
condamner une case, c'est donc au joueur 2 de jouer.

Les joueurs alternent leurs tour de jeu jusqu'à qu'un des joueur ne puisse plus déplacer son pion.

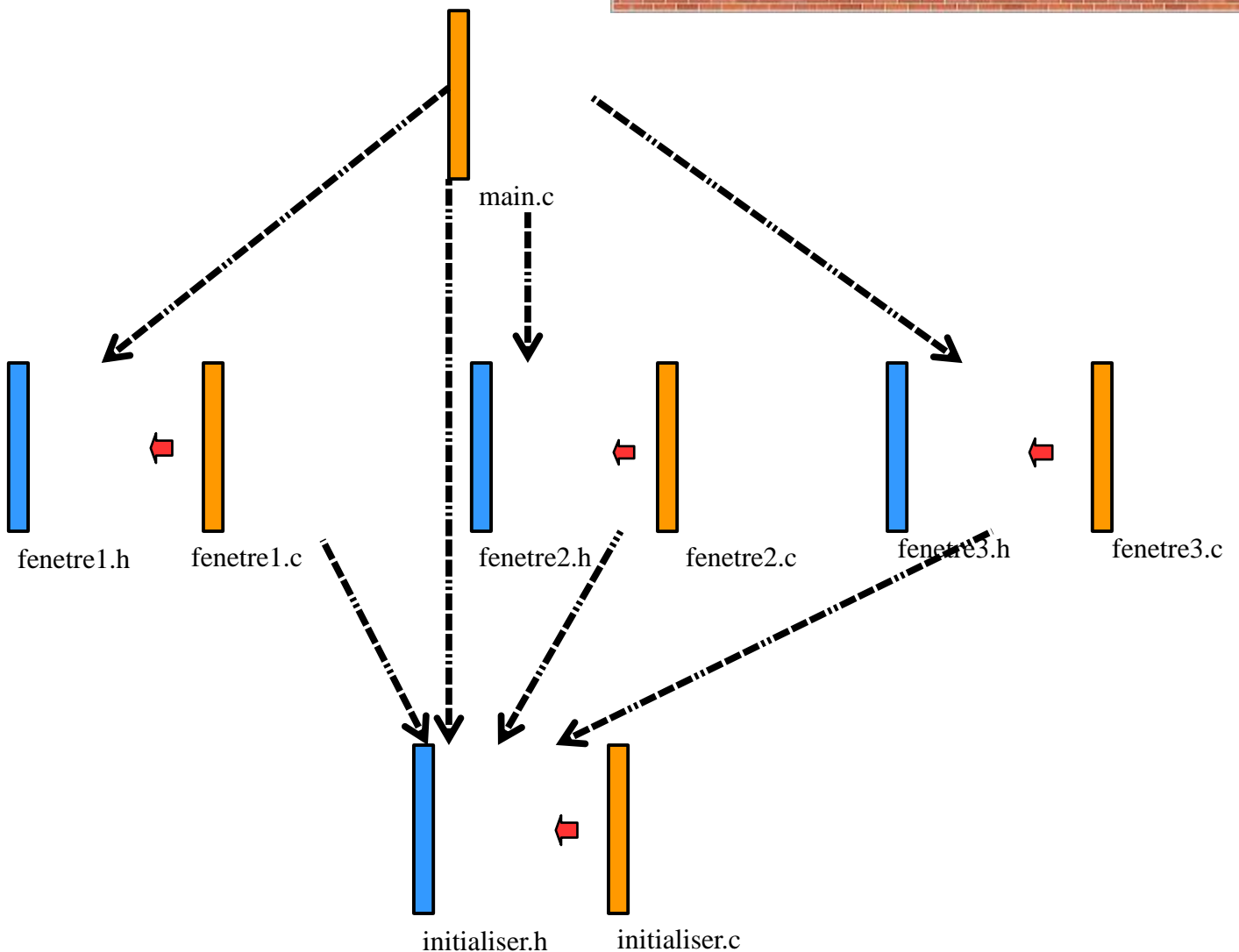
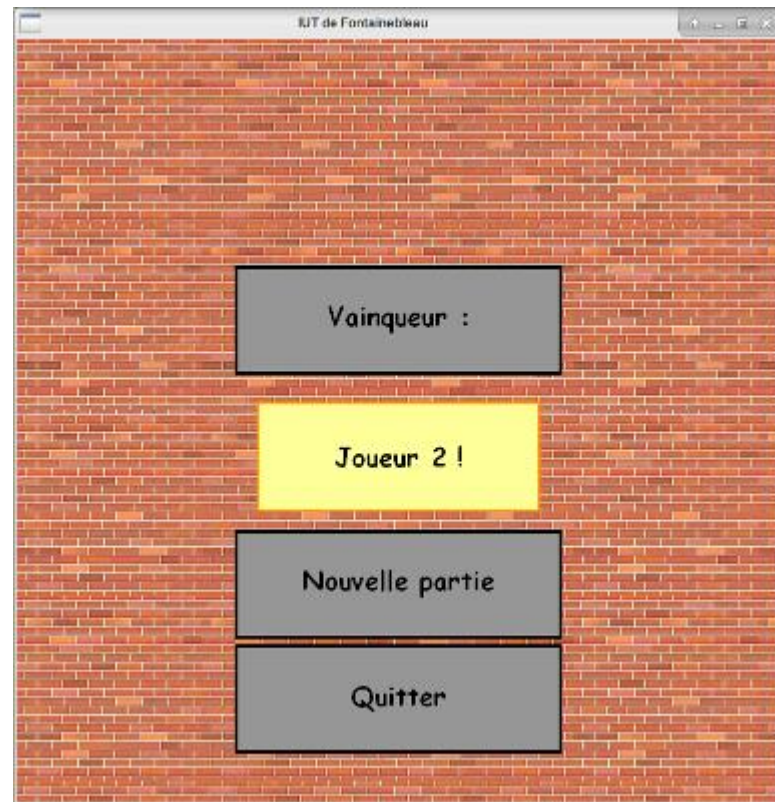


Le joueur 1 est bloquer, le
joueur 2 a gagner.

Une fois la partie terminée, le gagnant est affiché à l'écran, et on peut recommencer une partie ou quitter le jeu.

Affichage du vainqueur=>

Structure interne du programme:



Le fichier fenetre1.c contient les fonction qui servent à:

- _Afficher l'écran de paramètre de jeu.
- _Choisir les paramètre de jeu.

Le fichier fenetre2.c contient les fonction qui servent à:

- _afficher la grille de jeu.
- _exécuter les tour de joueur et robot .
- _verifier la condition de fin de jeu.

Le fichier fenetre3.c contient les fonctions qui servent à:

- _afficher le vainqueur.
- _choisir de quitter ou recommencer une partie.

Le fichier initialiser.c contient toute les constantes (#define) et les fonction qui servent à :

- _initialiser la position des joueurs et des case bloquées.
- _temporiser.
- _afficher le fond d'écran.

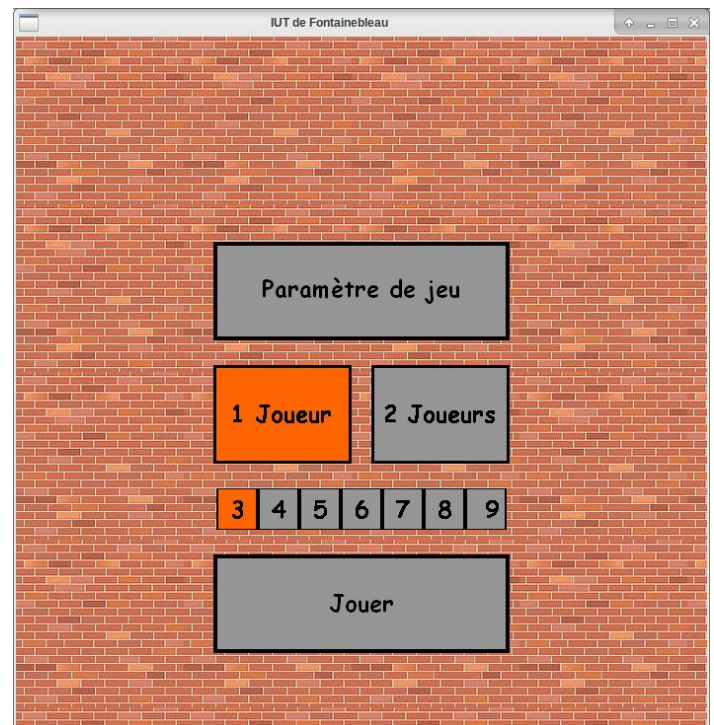
Le fichier main.c contient le programme qui permet de faire fonctionner le jeu.

Données représentant l'état d'une partie en cours

Les données du paramètre de jeu sont stockées dans un tableau, appelé `Choix`, dans la première case il y a le nombre de joueur (1 pour 1 joueur et 2 pour 2 joueurs) et la taille de la grille dans la deuxième case (3 pour une grille 3x3, 4 pour une grille 4x4 etc...).

Ici: `Choix[2]={1,3};`

Un clic sur "Jouer" permet de commencer la partie.



Écran: Choix des paramètres

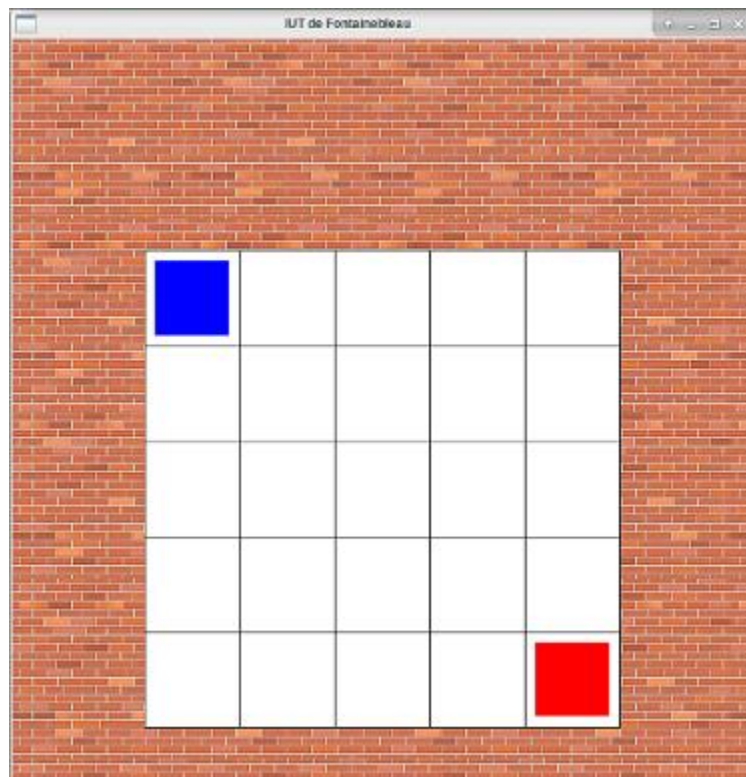
La position des différents éléments est stockée dans un tableau multidimensionnel de 9x9, appelé `tab_pos`.

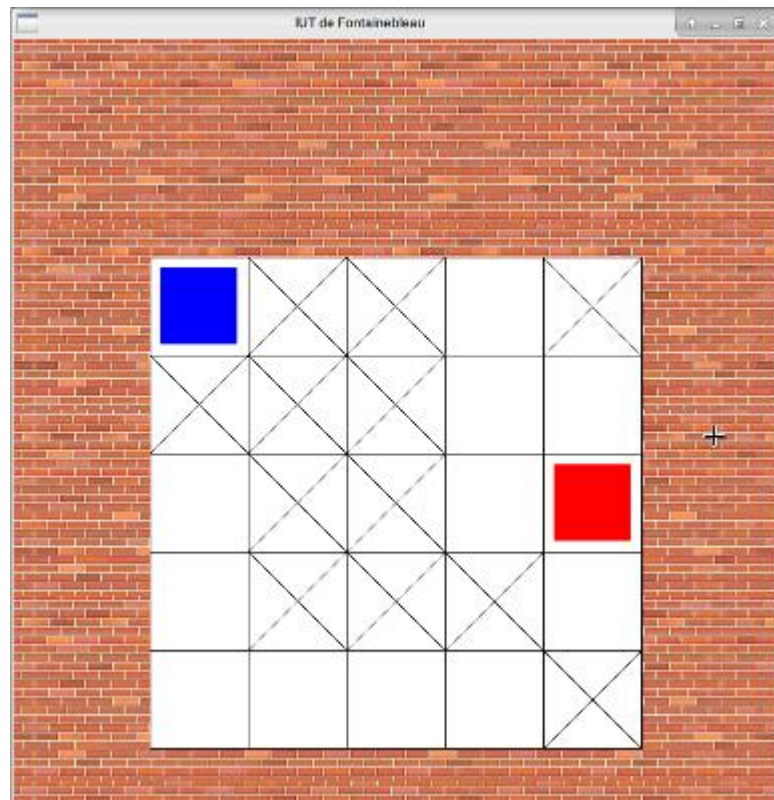
Le joueur 1 (bleu) est représenté par le chiffre 2, le joueur 2 (rouge) par le chiffre 3, une case bloquée par le chiffre 1 et une case vide par 0.

Quand un joueur est déplacé, son ancienne position est remplacée par 0.

ici `tab_pos=`

2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	3	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0





0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Ici, tab_pos=

2	1	1	0	1	0	0	0	0
1	1	1	0	0	0	0	0	0
0	1	1	0	3	0	0	0	0
0	1	1	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Pour verifier qu'un joueur est bloqué, la fonction FinDeJeu, parcourt le tableau et fixe le joueur qu'il trouve en premier, puis parcourt les case autour du joueur, si il y a que des 1 alors le joueur a perdu et la fonction renvoie 1.

2	1	1	0	1	0	0	0	0
1	1	1	0	0	0	0	0	0
0	1	1	0	3	0	0	0	0

0	1	1	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

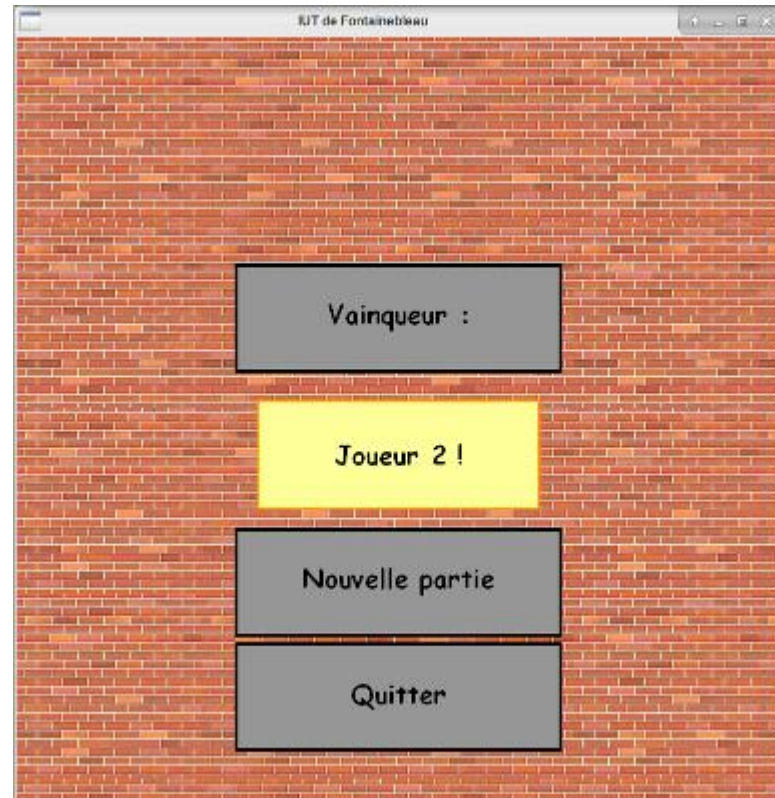
Pour vérifier que le joueur clique sur une case adjacente, on fait deux calcul:

$$(x_ancienne\ position - x_nouvelle\ position)^2$$

$$(y_ancienne\ position - y_nouvelle\ position)^2$$

si le resultat de ces calculs est inférieur a 2 alors le joueur a cliquer sur une case adjacente.

Une fois la partie fini, cette écran s'affiche, un clique sur "Nouvelle partie" et la valeur 1 est renvoyer par la fonction ChoixFin, un clique sur "Quitter" et la fonction ChoixFin renvoie -1.



Conclusion:

Le programme fonctionne mais le code peut être optimiser.

Je n'ai pas réussi a faire fonctionner la fonction ChargerImageFond de la bibliothèque graph.h sur les ecran virtuelle , j'ai du faire autrement.