#include <stdlib.h>

#include <stdio.h>

#include <SDL/SDL.h>

#include <conio.h>

typedef struct \_point

{

int casex;

int casey;

}Case;

Case c\_i,c\_f,c;

int test,i,j;

int comteur=0,comteur2=0;

int joueur1=1;

int joueur2=0;

int merelle[7][7];

int voisinnage[24][4];

SDL\_Surface \*ecran = NULL, \*imageDeFond = NULL,\*fondnoir = NULL, \*menu = NULL, \*instructions = NULL,\*vide = NULL, \*pionrouge1 = NULL,\*pionrouge2 = NULL,\*pionrouge3 = NULL,\*pionrouge4 = NULL,\*pionrouge5 = NULL,\*pionrouge6 = NULL,\*pionrouge7 = NULL,\*pionrouge8 = NULL,\*pionrouge9 = NULL, \*pionblanc1 = NULL, \*pionblanc2 = NULL, \*pionblanc3 = NULL, \*pionblanc4 = NULL, \*pionblanc5 = NULL, \*pionblanc6 = NULL, \*pionblanc7 = NULL, \*pionblanc8 = NULL, \*pionblanc9 = NULL;

SDL\_Rect positionFond = {0, 0, 0, 0};

SDL\_Rect positionCase={0,0,0,0};

SDL\_Rect positionCase1={0,0,0,0};

void affiche(int merelle[7][7],SDL\_Surface\* ecran,SDL\_Rect positionFond);

int Test\_moulin(int merelle[7][7],int j\_c,Case c);

int code(int i,int j);

void initialiser(int (\*merelle)[7]);

int score2(int merelle[7][7]);

int score1(int merelle[7][7]);

int move\_ci\_cf(int voisinnage[24][4],Case c\_i,Case c\_f);

void case\_voisin(int (\*voisin)[4]);

int main(int argc, char \*argv[])

{

if (SDL\_Init(SDL\_INIT\_VIDEO) == -1) // Démarrage de la SDL. Si erreur :

{

fprintf(stderr, "Erreur d'initialisation de la SDL : %s\n", SDL\_GetError()); // Écriture de l'erreur

return EXIT\_FAILURE; // On quitte le programme

}

SDL\_Surface \*ecran = NULL, \*imageDeFond = NULL,\*fondnoir = NULL, \*menu = NULL, \*instructions = NULL, \*pionrouge1 = NULL,\*pionrouge2 = NULL,\*pionrouge3 = NULL,\*pionrouge4 = NULL,\*pionrouge5 = NULL,\*pionrouge6 = NULL,\*pionrouge7 = NULL,\*pionrouge8 = NULL,\*pionrouge9 = NULL, \*pionblanc1 = NULL, \*pionblanc2 = NULL, \*pionblanc3 = NULL, \*pionblanc4 = NULL, \*pionblanc5 = NULL, \*pionblanc6 = NULL, \*pionblanc7 = NULL, \*pionblanc8 = NULL, \*pionblanc9 = NULL;

SDL\_Event event; SDL\_WaitEvent(&event);

int continuer = 1;

SDL\_Rect positionFond,positioninstructions,positionfondnoir, positionpionrouge1, positionpionblanc1, positionpionrouge2, positionpionblanc2, positionpionrouge3, positionpionblanc3, positionpionrouge4, positionpionblanc4, positionpionrouge5, positionpionblanc5, positionpionrouge6,positionpionblanc6, positionpionblanc7, positionpionrouge7, positionpionblanc8, positionpionrouge8, positionpionblanc9, positionpionrouge9, positionmenu;

positionfondnoir.x=0;

positionfondnoir.y=0;

positioninstructions.x=0;

positioninstructions.y=0;

positionmenu.x = 0;

positionmenu.y = 0;

positionFond.x = 70;

positionFond.y = 70;

positionpionrouge1.x = 0;

positionpionrouge1.y = 0;

positionpionblanc1.x = 550;

positionpionblanc1.y = 0;

positionpionrouge2.x =0;

positionpionrouge2.y = 50;

positionpionblanc2.x = 550;

positionpionblanc2.y = 50;

positionpionrouge3.x = 0;

positionpionrouge3.y = 100;

positionpionblanc3.x = 550;

positionpionblanc3.y = 100;

positionpionrouge4.x = 0;

positionpionrouge4.y = 150;

positionpionblanc4.x = 550;

positionpionblanc4.y = 150;

positionpionrouge5.x = 0;

positionpionrouge5.y = 200;

positionpionblanc5.x = 550;

positionpionblanc5.y = 200;

positionpionrouge6.x = 0;

positionpionrouge6.y = 250;

positionpionblanc6.x = 550;

positionpionblanc6.y = 250;

positionpionrouge7.x = 0;

positionpionrouge7.y = 300;

positionpionblanc7.x = 550;

positionpionblanc7.y = 300;

positionpionrouge8.x = 0;

positionpionrouge8.y = 350;

positionpionblanc8.x = 550;

positionpionblanc8.y = 350;

positionpionrouge9.x = 0;

positionpionrouge9.y = 400;

positionpionblanc9.x = 550;

positionpionblanc9.y = 400;

SDL\_Init(SDL\_INIT\_VIDEO);

// SDL\_WM\_SetIcon(SDL\_LoadBMP("cb.bmp"), NULL);

ecran = SDL\_SetVideoMode(600, 600, 32, SDL\_HWSURFACE); // Ouverture de la fenêtre

if( ecran == NULL ){

return EXIT\_FAILURE;

}

SDL\_WM\_SetCaption("Jeu de Moulin", NULL);

menu = SDL\_LoadBMP("MENU1.bmp");

fondnoir = SDL\_LoadBMP("fondnoir.bmp");

imageDeFond = SDL\_LoadBMP("moulin.bmp");

pionrouge1 = SDL\_LoadBMP("pionrouge1.bmp");

pionblanc1 = SDL\_LoadBMP("pionblanc1.bmp");

pionrouge2 = SDL\_LoadBMP("pionrouge1.bmp");

pionblanc2 = SDL\_LoadBMP("pionblanc1.bmp");

pionrouge3 = SDL\_LoadBMP("pionrouge1.bmp");

pionblanc3 = SDL\_LoadBMP("pionblanc1.bmp");

pionrouge4 = SDL\_LoadBMP("pionrouge1.bmp");

pionblanc4 = SDL\_LoadBMP("pionblanc1.bmp");

pionrouge5 = SDL\_LoadBMP("pionrouge1.bmp");

pionblanc5 = SDL\_LoadBMP("pionblanc1.bmp");

pionrouge6 = SDL\_LoadBMP("pionrouge1.bmp");

pionblanc6 = SDL\_LoadBMP("pionblanc1.bmp");

pionrouge7 = SDL\_LoadBMP("pionrouge1.bmp");

pionblanc7 = SDL\_LoadBMP("pionblanc1.bmp");

pionrouge8 = SDL\_LoadBMP("pionrouge1.bmp");

pionblanc8 = SDL\_LoadBMP("pionblanc1.bmp");

pionrouge9 = SDL\_LoadBMP("pionrouge1.bmp");

pionblanc9 = SDL\_LoadBMP("pionblanc1.bmp");

SDL\_BlitSurface(menu, NULL, ecran, &positionmenu);

while (continuer)

{

SDL\_WaitEvent(&event);

switch(event.type)

{

case SDL\_QUIT:

continuer = 0;

break;

case SDL\_MOUSEBUTTONUP: // Calcule la position de la souris pourafficher le cadre de selection!

if ( event.button.x > 0 && event.button.x <225

&& event.button.y > 200 && event.button.y <265 )

{

initialiser(merelle);

case\_voisin(voisinnage);

// fondnoir = SDL\_LoadBMP("fondnoir.bmp");

SDL\_BlitSurface(fondnoir, NULL, ecran, &positionfondnoir);

// imageDeFond = SDL\_LoadBMP("moulin.bmp");

SDL\_BlitSurface(imageDeFond, NULL, ecran, &positionFond);

// pionrouge1 = SDL\_LoadBMP("pionrouge1.bmp");

SDL\_BlitSurface(pionrouge1, NULL, ecran, &positionpionrouge1);

// pionblanc1 = SDL\_LoadBMP("pionblanc1.bmp");

SDL\_BlitSurface(pionblanc1, NULL, ecran, &positionpionblanc1);

// pionrouge2 = SDL\_LoadBMP("pionrouge1.bmp");

SDL\_BlitSurface(pionrouge2, NULL, ecran, &positionpionrouge2);

// pionblanc2 = SDL\_LoadBMP("pionblanc1.bmp");

SDL\_BlitSurface(pionblanc2, NULL, ecran, &positionpionblanc2);

// pionrouge3 = SDL\_LoadBMP("pionrouge1.bmp");

SDL\_BlitSurface(pionrouge3, NULL, ecran, &positionpionrouge3);

// pionblanc3 = SDL\_LoadBMP("pionblanc1.bmp");

SDL\_BlitSurface(pionblanc3, NULL, ecran, &positionpionblanc3);

// pionrouge4 = SDL\_LoadBMP("pionrouge1.bmp");

SDL\_BlitSurface(pionrouge4, NULL, ecran, &positionpionrouge4);

// pionblanc4 = SDL\_LoadBMP("pionblanc1.bmp");

SDL\_BlitSurface(pionblanc4, NULL, ecran, &positionpionblanc4);

//pionrouge5 = SDL\_LoadBMP("pionrouge1.bmp");

SDL\_BlitSurface(pionrouge5, NULL, ecran, &positionpionrouge5);

// pionblanc5 = SDL\_LoadBMP("pionblanc1.bmp");

SDL\_BlitSurface(pionblanc5, NULL, ecran, &positionpionblanc5);

// pionrouge6 = SDL\_LoadBMP("pionrouge1.bmp");

SDL\_BlitSurface(pionrouge6, NULL, ecran, &positionpionrouge6);

//pionblanc6 = SDL\_LoadBMP("pionblanc1.bmp");

SDL\_BlitSurface(pionblanc6, NULL, ecran, &positionpionblanc6);

// pionrouge7 = SDL\_LoadBMP("pionrouge1.bmp");

SDL\_BlitSurface(pionrouge7, NULL, ecran, &positionpionrouge7);

//pionblanc7 = SDL\_LoadBMP("pionblanc1.bmp");

SDL\_BlitSurface(pionblanc7, NULL, ecran, &positionpionblanc7);

// pionrouge8 = SDL\_LoadBMP("pionrouge1.bmp");

SDL\_BlitSurface(pionrouge8, NULL, ecran, &positionpionrouge8);

//pionblanc8 = SDL\_LoadBMP("pionblanc1.bmp");

SDL\_BlitSurface(pionblanc8, NULL, ecran, &positionpionblanc8);

// pionrouge9 = SDL\_LoadBMP("pionrouge1.bmp");

SDL\_BlitSurface(pionrouge9, NULL, ecran, &positionpionrouge9);

// pionblanc9 = SDL\_LoadBMP("pionblanc1.bmp");

SDL\_BlitSurface(pionblanc9, NULL, ecran, &positionpionblanc9);

initialiser(merelle);

affiche(merelle,ecran,positionFond);

//affiche(merelle); // 2- on affiche d'abord le damier et les pions

// SDL\_Flip(ecran); /\* Mise à jour de l'écran avec sa nouvelle couleur \*/

SDL\_FreeSurface(imageDeFond); // Libération de la surface

while (continuer)

{

SDL\_WaitEvent(&event);

switch(event.type)

{

case SDL\_QUIT:

continuer = 0;

break;

case SDL\_MOUSEBUTTONUP:

if(comteur2!=1){

if(joueur1==1 && comteur!=9)

{

for(i=0;i<7;i++){

for(j=0;j<7;j++){

positionCase.x=i\*450/7 +70;

positionCase.y=j\*450/7 +70;

if ( event.button.x >(i\*(450/7))+70 && event.button.x <((i+1)\*(450/7))+70

&& event.button.y > (j\*(450/7))+70 && event.button.y <((j+1)\*(450/7))+70 ){

test=Test\_moulin(merelle,1,c);

if(test==0){

if(merelle[i][j]==0){

c.casex=i;

c.casey=j;

merelle[i][j]=1;

// pionrouge1 = SDL\_LoadBMP("pionrouge1.bmp");

//SDL\_BlitSurface(pionrouge1, NULL, ecran, &positionCase);

// SDL\_Flip(ecran);

affiche(merelle,ecran,positionFond);

test=Test\_moulin(merelle,1,c);

if(test==0){

joueur1=0;

joueur2=1;

comteur++;

printf("joueur2, votre tour ");

}

}

}

else{

if(merelle[i][j]==-1){

printf("joueur1, vous avez droit de supprimmer un pion blanc ");

merelle[i][j]=0;

// vide = SDL\_LoadBMP("vide.bmp");

//SDL\_BlitSurface(vide, NULL, ecran, &positionCase);

//SDL\_Flip(ecran);

affiche(merelle,ecran,positionFond);

joueur1=0;

joueur2=1;

comteur++;

printf("joueur2, votre tour ");

}

}

}

}

}

}

else

{

if(joueur2==1 )

{

for(i=0;i<7;i++){

for(j=0;j<7;j++){

positionCase.x=i\*450/7 +70;

positionCase.y=j\*450/7 +70;

if ( event.button.x >(i\*(450/7))+70 && event.button.x <((i+1)\*(450/7))+70

&& event.button.y > (j\*(450/7))+70 && event.button.y <((j+1)\*(450/7))+70 ){

test=Test\_moulin(merelle,-1,c);

if(test==0){

if(merelle[i][j]==0){

c.casex=i;

c.casey=j;

merelle[i][j]=-1;

// pionblanc1 = SDL\_LoadBMP("pionblanc1.bmp");

// SDL\_BlitSurface(pionblanc1, NULL, ecran, &positionCase);

//SDL\_Flip(ecran);

affiche(merelle,ecran,positionFond);

test=Test\_moulin(merelle,-1,c);

if(test==0){

joueur2=0;

joueur1=1;

if(comteur==9){

comteur2=1;

joueur1=1;

joueur2=0;

}

}

}

}

else{

if(merelle[i][j]==1){

printf("joueur2, vous avez droit de supprimmer un pion blanc ");

// vide = SDL\_LoadBMP("vide.bmp");

//SDL\_BlitSurface(vide, NULL, ecran, &positionCase);

//SDL\_Flip(ecran);

merelle[i][j]=0;

affiche(merelle,ecran,positionFond);

joueur1=1;

joueur2=0;

if(comteur==9){

comteur2=1;

joueur1=1;

joueur2=0;

}

}

}

}

}

}

}

}

}

else{

if(joueur1==1 && score1(merelle)>2) //ayant merelle i j = 1 pion rouge

{

for(i=0;i<7;i++){

for(j=0;j<7;j++){

positionCase.x=i\*450/7 +70;

positionCase.y=j\*450/7 +70;

if ( event.button.x >(i\*(450/7))+70 && event.button.x <((i+1)\*(450/7))+70

&& event.button.y > (j\*(450/7))+70 && event.button.y <((j+1)\*(450/7))+70 ){

if(joueur2==0){

test=Test\_moulin(merelle,1,c);

if(test==1){

printf("joueur1, vous avez droit de supprimmer un pion blanc ");

if(merelle[i][j]==-1){

// vide = SDL\_LoadBMP("vide.bmp");

// SDL\_BlitSurface(vide, NULL, ecran, &positionCase);

// SDL\_Flip(ecran);

merelle[i][j]=0;

affiche(merelle,ecran,positionFond);

joueur1=0;

joueur2=1;

printf("joueur2, votre tour ");

break;

}

}

else{

if(merelle[i][j]==1){

c\_i.casex=i;

c\_i.casey=j;

printf("joueur1, choisissez mtn une case destiné");

// vide = SDL\_LoadBMP("vide.bmp");

// SDL\_BlitSurface(vide, NULL, ecran, &positionCase);

// SDL\_Flip(ecran);

merelle[i][j]=0;

affiche(merelle,ecran,positionFond);

joueur1=1;

joueur2=0.5;

break;

}

}

}

else{

c\_f.casex=i;

c\_f.casey=j;

if(merelle[i][j]==0 && move\_ci\_cf(voisinnage,c\_i,c\_f)==1 ){

c.casex=i;

c.casey=j;

// pionrouge1 = SDL\_LoadBMP("pionrouge1.bmp");

// SDL\_BlitSurface(pionrouge1, NULL, ecran, &positionCase);

positionCase1.x=c\_i.casex\*450/7 +70;

positionCase1.y=c\_i.casey\*450/7 +70;

// vide = SDL\_LoadBMP("vide.bmp");

// SDL\_BlitSurface(vide, NULL, ecran, &positionCase1);

// SDL\_Flip(ecran);

merelle[i][j]=1;

affiche(merelle,ecran,positionFond);

joueur1=0;

joueur2=1;

printf("joueur2, votre tour ");

break;

}

else{

printf("mouvement nom valide,rechoisissez la case à deplacer");

//positionCase.x=c\_i.casex\*450/7 +70;

//positionCase.y=c\_i.casey\*450/7 +70;

//pionrouge1 = SDL\_LoadBMP("pionrouge1.bmp");

//SDL\_BlitSurface(pionrouge1, NULL, ecran, &positionCase);

//SDL\_Flip(ecran);

merelle[c\_i.casex][c\_i.casey]=1;

affiche(merelle,ecran,positionFond);

joueur1=1;

joueur2=0;

break;

}

}

}

}

}

}

else {

if(joueur2==1 && score2(merelle)>2){ //joueur2 ayant pion blanc ou merelle[i][j]==-1

for(i=0;i<7;i++){

for(j=0;j<7;j++){

positionCase.x=i\*450/7 +70;

positionCase.y=j\*450/7 +70;

if ( event.button.x >(i\*(450/7))+70 && event.button.x <((i+1)\*(450/7))+70

&& event.button.y > (j\*(450/7))+70 && event.button.y <((j+1)\*(450/7))+70 ){

if(joueur1==0){

test=Test\_moulin(merelle,-1,c);

if(test==1){

if(merelle[i][j]==1){

printf("joueur2, vous avez droit de supprimmer un pion blanc ");

//vide = SDL\_LoadBMP("vide.bmp");

// SDL\_BlitSurface(vide, NULL, ecran, &positionCase);

// SDL\_Flip(ecran);

merelle[i][j]=0;

affiche(merelle,ecran,positionFond);

joueur1=1;

joueur2=0;

printf("joueur1, votre tour ");

break;

}

}

else{

if(merelle[i][j]==-1){

printf("joueur2, choisissez la destination de votre pion ");

c\_i.casex=i;

c\_i.casey=j;

//vide = SDL\_LoadBMP("vide.bmp");

//SDL\_BlitSurface(vide, NULL, ecran, &positionCase);

merelle[i][j]=0;

affiche(merelle,ecran,positionFond);

joueur2=1;

joueur1=0.5;

break;

}

}

}

else{

c\_f.casex=i;

c\_f.casey=j;

if(merelle[i][j]==0 && move\_ci\_cf(voisinnage,c\_i,c\_f)==1 ){

c.casex=i;

c.casey=j;

//pionblanc1 = SDL\_LoadBMP("pionblanc1.bmp");

//SDL\_BlitSurface(pionblanc1, NULL, ecran, &positionCase);

positionCase1.x=c\_i.casex\*450/7 +70;

positionCase1.y=c\_i.casey\*450/7 +70;

//vide = SDL\_LoadBMP("vide.bmp");

// SDL\_BlitSurface(vide, NULL, ecran, &positionCase1);

//SDL\_Flip(ecran);

merelle[i][j]=-1;

affiche(merelle,ecran,positionFond);

joueur2=0;

joueur1=1;

printf("joueur1, votre tour ");

break;

}

else{

merelle[c\_i.casex][c\_i.casey]=-1;

affiche(merelle,ecran,positionFond);

joueur2=1;

joueur1=0;

printf("mouvement nom valide,rechoisissez la case à deplacer");

break;

}

}

}

}

}

}

}

}

break;

}

//Affichage:

}

}

else

{

fondnoir = SDL\_LoadBMP("fondnoir.bmp");

SDL\_BlitSurface(fondnoir, NULL, ecran, &positionfondnoir);

instructions = SDL\_LoadBMP("instructions.bmp");

SDL\_BlitSurface(instructions, NULL, ecran, &positioninstructions);

SDL\_Flip(ecran); /\* Mise à jour de l'écran avec sa nouvelle couleur \*/

SDL\_FreeSurface(imageDeFond); // Libération de la surface

}

break;

}

SDL\_Flip(ecran);

}

SDL\_Quit();

return EXIT\_SUCCESS; // Fermeture du programme

}

///fonction test de validité de mouvement de la case initiale vers celle finale qui verifie si la cases finale est voisine a celle initiale

int move\_ci\_cf(int voisinnage[24][4],Case c\_i,Case c\_f)

{

int test=0,ligne\_i,ligne\_f,k;

ligne\_i=code(c\_i.casex,c\_i.casey);

ligne\_f=code(c\_f.casex,c\_f.casey);

for(k=0;k<4;k++){

if(voisinnage[ligne\_i][k]==ligne\_f){

test=1;

return test;

}

}

return test;

}

// FONCTION QUI CALCULE LE NOMBRE DE PIONS DU JOUEUR1 ET QUI REPRESENTE AUSSI SON SCORE

int score1(int merelle[7][7])

{

int score1=0;

int i,j;

for(i=0;i<7;i++){

for(j=0;j<7;j++){

if(merelle[i][j]==1)

score1++;

}

}

return score1;

}

// FONCTION QUI CALCULE LE NOMBRE DE PIONS DU JOUEUR2 ET QUI REPRESENTE AUSSI SON SCORE

int score2(int merelle[7][7])

{

int score2=0;

int i,j;

for(i=0;i<7;i++){

for(j=0;j<7;j++){

if(merelle[i][j]==-1)

score2++;

}

}

return score2;

}

///// initialiser la matrice

void initialiser(int (\*merelle)[7])

{

int i ,k;

for (i=0; i<7; i++){ // case nom utilisable contient 8

for (k=0; k<7; k++){

merelle[i][k] = 8;

}

}

merelle[0][0]=0;

merelle[3][0]=0; // les cases contenant 0 sont vides du moulin et peuvent etre remplises

merelle[6][0]=0;

merelle[0][6]=0;

merelle[3][6]=0;

merelle[6][6]=0;

merelle[0][3]=0;

merelle[6][3]=0;

//

merelle[1][1]=0;

merelle[3][1]=0;

merelle[5][1]=0;

merelle[1][5]=0;

merelle[3][5]=0;

merelle[5][5]=0;

merelle[1][3]=0;

merelle[5][3]=0;

//

merelle[2][2]=0;

merelle[3][2]=0;

merelle[4][2]=0;

merelle[2][4]=0;

merelle[3][4]=0;

merelle[4][4]=0;

merelle[2][3]=0;

merelle[4][3]=0;

}

//initialisation de la matrice voisinage

// fonction qui remplie la matrice voisinnage qui definie les positions (par ordre horizontale) des cases voisines de chaque case pouvant contenir un pion ..

// marquée au debut de l'initialisation par \* pourrant etre remplie

void case\_voisin(int (\*voisin)[4])

{

voisin[0][0]=1;

voisin[0][1]=9;

voisin[0][2]=24;

voisin[0][3]=24;

voisin[1][0]=0;

voisin[1][1]=2;

voisin[1][2]=4;

voisin[1][3]=24;

voisin[2][0]=1;

voisin[2][1]=14;

voisin[2][2]=24;

voisin[2][3]=24;

voisin[3][0]=4;

voisin[3][1]=10;

voisin[3][2]=24;

voisin[3][3]=24;

voisin[4][0]=1;

voisin[4][1]=3;

voisin[4][2]=5;

voisin[4][3]=7;

voisin[5][0]=4;

voisin[5][1]=13;

voisin[5][2]=24;

voisin[5][3]=24;

voisin[6][0]=7;

voisin[6][1]=11;

voisin[6][2]=24;

voisin[6][3]=24;

voisin[7][0]=4;

voisin[7][1]=6;

voisin[7][2]=8;

voisin[7][3]=24;

voisin[8][0]=7;

voisin[8][1]=12;

voisin[8][2]=24;

voisin[8][3]=24;

voisin[9][0]=0;

voisin[9][1]=21;

voisin[9][2]=10;

voisin[9][3]=24;

voisin[10][0]=3;

voisin[10][1]=9;

voisin[10][2]=18;

voisin[10][3]=11;

voisin[11][0]=6;

voisin[11][1]=15;

voisin[11][2]=10;

voisin[11][3]=24;

voisin[12][0]=8;

voisin[12][1]=17;

voisin[12][2]=13;

voisin[12][3]=24;

voisin[13][0]=12;

voisin[13][1]=5;

voisin[13][2]=20;

voisin[13][3]=14;

voisin[14][0]=2;

voisin[14][1]=13;

voisin[14][2]=23;

voisin[14][3]=24;

voisin[15][0]=11;

voisin[15][1]=16;

voisin[15][2]=24;

voisin[15][3]=24;

voisin[16][0]=15;

voisin[16][1]=17;

voisin[16][2]=19;

voisin[16][3]=24;

voisin[17][0]=12;

voisin[17][1]=16;

voisin[17][2]=24;

voisin[17][3]=24;

voisin[18][0]=10;

voisin[18][1]=19;

voisin[18][2]=24;

voisin[18][3]=24;

voisin[19][0]=16;

voisin[19][1]=18;

voisin[19][2]=20;

voisin[19][3]=22;

voisin[20][0]=13;

voisin[20][1]=19;

voisin[20][2]=24;

voisin[20][3]=24;

voisin[21][0]=9;

voisin[21][1]=22;

voisin[21][2]=24;

voisin[21][3]=24;

voisin[22][0]=19;

voisin[22][1]=21;

voisin[22][2]=23;

voisin[22][3]=24;

voisin[23][0]=22;

voisin[23][1]=14;

voisin[23][2]=24;

voisin[23][3]=24;

}

//fonction codage de case en matrice merelle -> equivalent en matrice environ

int code(int i,int j)

{

int merelle[7][7];

int l ,k;

for (l=0; l<7; l++){ // case nom utilisable contient 8

for (k=0; k<7; k++){

merelle[l][k] = 8;

}

}

merelle[0][0]=0;

merelle[3][0]=0; // les cases contenant 0 sont vides du moulin et peuvent etre remplises

merelle[6][0]=0;

merelle[0][6]=0;

merelle[3][6]=0;

merelle[6][6]=0;

merelle[0][3]=0;

merelle[6][3]=0;

//

merelle[1][1]=0;

merelle[3][1]=0;

merelle[5][1]=0;

merelle[1][5]=0;

merelle[3][5]=0;

merelle[5][5]=0;

merelle[1][3]=0;

merelle[5][3]=0;

//

merelle[2][2]=0;

merelle[3][2]=0;

merelle[4][2]=0;

merelle[2][4]=0;

merelle[3][4]=0;

merelle[4][4]=0;

merelle[2][3]=0;

merelle[4][3]=0;

int conteur=-1;

for (l=0; l<7; l++){

for (k=0; k<7; k++){

if((merelle[l][k]!=8)){

conteur++;

if((l==i)&&(k==j)){

return conteur;

}

}

}

}

}

// FONCTION POUR tester si un moulin existe en faveur du joueur en cour j\_c la fonction retourne 1 si il existe un moulin pour j\_c et 0 sinn

int Test\_moulin(int merelle[7][7],int j\_c,Case c)

{

int test=0;

switch(c.casex){

case 0 :

if((merelle[0][0]==j\_c)&&(merelle[0][3]==j\_c)&&(merelle[0][6]==j\_c)&&(((c.casex==0)&&(c.casey==0))||((c.casex==0)&&(c.casey==3))||((c.casex==0)&&(c.casey==6)))){

test++;

}

break;

case 1 :

if((merelle[1][1]==j\_c)&&(merelle[1][3]==j\_c)&&(merelle[1][5]==j\_c)&&(((c.casex==1)&&(c.casey==1))||((c.casex==1)&&(c.casey==3))||((c.casex==1)&&(c.casey==5)))){

test++;

}

break;

case 3 :

if(((merelle[3][0]==j\_c)&&(merelle[3][1]==j\_c)&&(merelle[3][2]==j\_c)&&(((c.casex==3)&&(c.casey==0))||((c.casex==3)&&(c.casey==1))||((c.casex==3)&&(c.casey==2))))||((merelle[3][4]==j\_c)&&(merelle[3][5]==j\_c)&&(merelle[3][6]==j\_c)&&(((c.casex==3)&&(c.casey==4))||((c.casex==3)&&(c.casey==5))||((c.casex==3)&&(c.casey==6))))){

test++;

}

break;

case 2 :

if((merelle[2][2]==j\_c)&&(merelle[2][3]==j\_c)&&(merelle[2][4]==j\_c)&&(((c.casex==2)&&(c.casey==2))||((c.casex==2)&&(c.casey==3))||((c.casex==2)&&(c.casey==4)))){

test++;

}

break;

case 4 :

if((merelle[4][2]==j\_c)&&(merelle[4][3]==j\_c)&&(merelle[4][4]==j\_c)&&(((c.casex==4)&&(c.casey==2))||((c.casex==4)&&(c.casey==3))||((c.casex==4)&&(c.casey==4)))){

test++;

}

break;

case 5 :

if((merelle[5][1]==j\_c)&&(merelle[5][3]==j\_c)&&(merelle[5][5]==j\_c)&&(((c.casex==5)&&(c.casey==1))||((c.casex==5)&&(c.casey==3))||((c.casex==5)&&(c.casey==5)))){

test++;

}

break;

case 6 :

if((merelle[6][0]==j\_c)&&(merelle[6][3]==j\_c)&&(merelle[6][6]==j\_c)&&(((c.casex==6)&&(c.casey==0))||((c.casex==6)&&(c.casey==3))||((c.casex==6)||(c.casey==6)))){

test++;

}

break;

default:

break;

}

switch(c.casey){

case 0 :

if((merelle[0][0]==j\_c)&&(merelle[3][0]==j\_c)&&(merelle[6][0]==j\_c)&&(((c.casex==0)&&(c.casey==0))||((c.casex==3)&&(c.casey==0))||((c.casex==6)||(c.casey==0))))

test++;

break;

case 1 :

if((merelle[1][1]==j\_c)&&(merelle[3][1]==j\_c)&&(merelle[5][1]==j\_c)&&(((c.casex==1)&&(c.casey==1))||((c.casex==3)&&(c.casey==1))||((c.casex==5)||(c.casey==1))))

test++;

break;

case 3 :

if(((merelle[0][3]==j\_c)&&(merelle[1][3]==j\_c)&&(merelle[2][3]==j\_c)&&(((c.casex==0)&&(c.casey==3))||((c.casex==1)&&(c.casey==3))||((c.casex==2)||(c.casey==3))))||((merelle[4][3]==j\_c)&&(merelle[5][3]==j\_c)&&(merelle[6][3]==j\_c)&&(((c.casex==4)&&(c.casey==3))||((c.casex==5)&&(c.casey==3))||((c.casex==6)||(c.casey==3)))))

test++;

break;

case 2 :

if((merelle[2][2]==j\_c)&&(merelle[3][2]==j\_c)&&(merelle[4][2]==j\_c)&&(((c.casex==2)&&(c.casey==2))||((c.casex==3)&&(c.casey==2))||((c.casex==4)||(c.casey==2))))

test++;

break;

case 4 :

if((merelle[2][4]==j\_c)&&(merelle[3][4]==j\_c)&&(merelle[4][4]==j\_c)&&(((c.casex=2)&&(c.casey==4))||((c.casex==3)&&(c.casey==4))||((c.casex==4)||(c.casey==4))))

test++;

break;

case 5 :

if((merelle[1][5]==j\_c)&&(merelle[3][5]==j\_c)&&(merelle[5][5]==j\_c)&&(((c.casex==1)&&(c.casey==5))||((c.casex==3)&&(c.casey==5))||((c.casex==5)||(c.casey==5))))

test++;

break;

case 6 :

if((merelle[0][6]==j\_c)&&(merelle[3][6]==j\_c)&&(merelle[6][6]==j\_c)&&(((c.casex==0)&&(c.casey==6))||((c.casex==3)&&(c.casey==6))||((c.casex==6)||(c.casey==6))))

test++;

break;

default:

break;

}

return test;

}

void affiche(int merelle[7][7],SDL\_Surface\* ecran,SDL\_Rect positionFond)

{

int i, j;

SDL\_Rect positionCase={0,0,0,0};

imageDeFond = SDL\_LoadBMP("moulin.bmp");

pionrouge1 = SDL\_LoadBMP("pionrouge1.bmp");

pionblanc1 = SDL\_LoadBMP("pionblanc1.bmp");

vide = SDL\_LoadBMP("vide.bmp");

SDL\_BlitSurface(imageDeFond, NULL, ecran, &positionFond);

for (i = 0; i < 7; i++)

{

for (j = 0; j < 7; j++)

{

positionCase.x=i\*450/7 +70;

positionCase.y=j\*450/7 +70;

switch(merelle[i][j])

{

case 1:

SDL\_BlitSurface(pionrouge1, NULL, ecran, &positionCase);

break;

case -1:

SDL\_BlitSurface(pionblanc1, NULL, ecran, &positionCase);

break;

case 0:

SDL\_BlitSurface(vide, NULL, ecran, &positionCase);

break;

default:

break;

}

}

}

SDL\_Flip(ecran);

}