Juego Dots and Boxes

MATRIZ:

Para la carga del tablero-matriz de tipo char utilizamos un for junto a un if la cual condiciona que en las posiciones pares de fila y columna

se coloque el punto (en este caso utilizamos un asterisco) y lo que resta de la matriz con ceros.

Leemos o generamos n que seria el tamaño de la matriz, utilizamos n=n+(n-1); para que el programa entienda que la N que se ingreso

sea la cantidad de puntos que se va mostrar.

VECTOR PARA VERIFICAR POSICIONES DISPONIBLES PARA ALEATORIO

Para los puntos random de filas y columnas de la maquina utilizamos la condicion que si la matriz es igual a '0' (espacio libre)

el juego continua porque siguen puntos libres por conectar luego verificamos que tenemos un punto libre y correcto, entonces esas 2 posiciones de fila y columna almacenamos

en un vector la cual al ingresar el punto 1 random de la maquina, se verifica de acuerdo al vector generado con posiciones validas.

VECTOR PARA ALMACENAR PUNTAJES DE LOS JUGADORES

Los puntajes del usuario y maquina almacenamos en un vector donde en cada jugada se actualiza y se compara para volver a jugar en el caso que completo un cuadrado

y para saber quien gano, perdio o si se genero un empate.

VARIABLES

int i, j,n=0;// contador, tamaño tablero

int aux=0;// validacion para punto 2

int x=0;// validacion para finalizar juego

int turno=1;//carga de matriz

int user=0;//usuario

int user1[2]={0,0};//vector de usuario para puntajes

int m1=0;// validacion para mensaje de bienvenida para tamaño tablero

int m2=0;// validacion para mensaje de bienvenida quien inicia y color

char matriz[30][30];//

int fila[1000],columna[1000];//2vectores para la seleccion random

int pf1=1,pc1=1,pf2,pc2;

int aux1=0;//validacion para punto 1

int aux2;//guardamos el valor del puntaje para verificar si aumenta

int lf,lc;//guardar punto

int band=0;//contador de conexiones

int aux3=0,aux4=0;//verificador de lineas ocupadas

int fin=0;//fin del juego, si no hay espacios disponibles

int y,z,w;//para menu inicializar

Para el menu inicial, esta función muestra en pantalla el mensaje de bienvenida al juego:

void inicio(void){

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\*\t\t\t\t\t\t\*");

printf("\n\*\t\tBIENVENIDO AL JUEGO!!!\t\t\*");

printf("\n\*\t\tDOTS AND BOXES\t\t\t\*");

printf("\n\*\t\t\t\t\t\t\*");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

Con esta función leemos o generamos n, utilizamos n=n+(n-1); para que el programa entienda que la N que se ingreso sea la cantidad de puntos que se va mostrar:

void tablero(void){

if (m1==0){

while (y<1 || y>2 ){

printf("\n\nPresione 1, si desea ingresar por teclado el tamaño del tablero: ");

printf("\n\nPresione 2, si desea generar aleatoriamente el tamaño del tablero: \n");

scanf("%d",&y);//leer 1 o 2

if (y==1){

while (n<3 || n>15 ){

printf("\n\nIngrese el tamaño de la matriz: ");

scanf("%d",&n);

}

}else if(y==2){

while (n<3){ //|| n>15

n=rand()%16;//genera del 1 al 15

}

}

//printf("n es %d",n);

n=n+(n-1);//para tablero real

//printf(",n tablero %d",n);

m1=1;

}

}

}

Con esta función lo que hacemos es elegir 3 para ingresar por teclado quien inicia el juego y 4 si queremos generar aleatoriamente quien inicia el juego, además elegimos el botón 5, si deseamos ingresar por teclado el color o 6 si deseamos generar aleatoriamente los colores, verificando el ingreso por teclado con un while para que el numero este entre 3 y 4 ademas con otro while verificamos que sea el usuario o la computadora (1 y 2) y en el caso de seleccino aleatoria sumamos uno al random para limitarlo entre 1 user o 2 compu:

void jugador(void){

while (z<3 || z>4){

printf("\n\nPresione 3, si desea ingresar por teclado quien inicia el juego: ");

printf("\n\nPresione 4, si desea generar aleatoriamente quien inicia el juego: \n");

scanf("%d",&z);

if(z==3){

while (user<1 || user>2){

printf("Seleccionar jugador inicial\n");

printf("\n1: Jugador, 2:Computadora\n");

scanf("%d",&user);//leer user 1 o 2

}

}else if(z==4){

user=1+rand()%2;//sumar uno al random para limitarlo entre 1 user o 2 compu

}

}

while (w<5 || w>6){

printf("\n\nPresione 5, si desea ingresar por teclado su color: ");

printf("\n\nPresione 6, si desea generar aleatoriamente los colores: \n");

scanf("%d",&w);

y=0;

if(w==5){

while (y<1 || y>2){

printf("Seleccione su color\n");

printf("\npresione 1 para Rojo\npresione 2 para Azul\n");

scanf("%d",&y);

}

}else if(w==6){

y=1+rand()%2;//sumar uno al random para limitarlo entre 1 rojo o 2 azul

}

m2=1;

}

}

Con esta función lo que hacemos es imprimir el color que le corresponde a cada jugador, la lógica seria que 1 es el color rojo y 2 el azul para el usuario y lo contrario se aplica para la computadora:

void turnos(void){

if (pf1!=99){

printf("\n\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\nTurno del Jugador:%d",user);

if(user==1 && y==1){//y color rojo

printf("\nCOLOR: ROJO");

} else if(user==1 && y==2){// y color azul

printf("\nCOLOR: AZUL");

} else if(user==2 && y==1){

printf("\nCOLOR: AZUL");

} else if(user==2 && y==2){

printf("\nCOLOR: ROJO");

}

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

}

void carga\_matriz(void){

if (turno==1){

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

printf("\n");

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

if (i%2==0 && j%2==0){//colocar los astericos en las posiciones de fila y columna que son multiplos de 2

matriz[i][j]='\*';//carga astericos

}else matriz[i][j]='0';//cargado de la matriz

printf("%c",matriz[i][j]);//imprimir matriz

}

}

}

turno++;

printf("\n");

}

Con esta función lo que hacemos es, para los puntos random de filas y columnas utilizamos la condicion que si la matriz es igual a '0', el juego continua porque siguen puntos libres por conectar, luego verificamos que tenemos un punto libre y correcto, entonces esas 2 posiciones de fila y columna almacenamos en un vector la cual al ingresar el punto 1 random de la maquina, se genera de acuerdo al vector generado con posiciones validas:

void verificar\_aleatorio(void){

aux1=0;//para que solicite la carga de P1

fin=0;//para que cambie en 1 si solo si en el siguente for encuentra una jugada posible

z=0;//contador de los posibles puntos a seleccionar por random

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

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

if (matriz[i][j]=='0'){

fin=1;//condicion para que el juego continue

}

//verificacion para seleccion de P1 para random

if (matriz[i][j]=='\*'){

if(matriz[i+1][j]=='0'){//conexion abajo

fila[z]=i;//guardar esa posicion en vector fila

columna[z]=j;//guardar esa posicion en vector columna

z++;

}

if(matriz[i][j+1]=='0'){//conexion derecha

fila[z]=i;

columna[z]=j;

z++;

}

if(matriz[i-1][j]=='0'){//conexion arriba

fila[z]=i;

columna[z]=j;

z++;

}

if(matriz[i][j-1]=='0'){//conexion izq

fila[z]=i;

columna[z]=j;

z++;

}

}

}

}

}

Con esta función lo que hacemos es terminar el juego, en el caso que no se encuentre algun espacio en la matriz, termina el juego con un mensaje y puntajes de cada jugador, como tambien tenemos la posibilidad de terminar el juego seleccionando 99 como fila:

void game\_over(void){

if (fin==0){//condicion GAME OVER

printf("GAME OVER");

x=1;//condicion para que no vuelva a repetir turnos

aux1=1;// condicion para que no pida p1

pf1=99; //condicion para que no pida p2

//agregar ganador

if (user1[0]>user1[1]){//compara vector de puntaje

printf("\nyou win");

}else if(user1[0]<user1[1]){

printf("\nyou lose");

}else{

printf("\nit's a tie");

}

}

}

Con esta función, leemos el punto 1 y lo traducimos pf1=pf1-(2-pf1); para que el programa pueda entender en que posicion queremos estar, para la computadora se genera aleatoriamente entre los puntos validos almacenado en el vector fila-columna y el mismo paso para las columnas, se verifica que los puntos sean par y esten dentro del tablero y con un conjunto de if y aux verificamos que el punto 1 se pueda volver a leer en caso que se vuelva a seleccionar un punto que ya no tiene como conectarse con otro punto:

void punto1(void){

if(user==1){

printf("Ingrese el punto 1: \n");

printf("\nEn el caso que quiera terminar el juego presione 99 \n");

printf("fila: ");

scanf("%d",&pf1);

if (pf1!=99){

pf1=pf1-(2-pf1);//traducir fila a posiciones

}

if (pf1==99){

x=1;

aux1=1;

}

}else{

w=rand()%(z);//z se carga al final de cada carga de losvectores, por ende el ultimo z++ no llevara ningun valor util.

pf1=fila[w];//compu

}

if(user==1 && pf1!=99){

printf("columna: ");

scanf("%d",&pc1);//user

pc1=pc1-(2-pc1);//traducir columna a posiciones

}else if(pf1!=99){

pc1=columna[w];//compu

}

/\*printf("\nP1(%d,%d) ",pf1,pc1);

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

printf("\nV%d(%d,%d) ",i,fila[i],columna[i]);

}\*/

if (pf1%2==0 && pc1%2==0 && pf1>=0 && pf1<=n && pc1>=0 && pc1<=n){//verificar fila o columna sea par, y no se encuentre fuera

/////////para que pida cargar de nuevo el punto 1//////

if(pf1-1>=0){//verificar que haya una jugada posible

aux4++;//contador jugadas posibles

if(matriz[pf1-1][pc1]=='|'){

aux3++; //jugadas realizadas arriba

}

}

if(pc1+1<=(n-1)){//verificar que haya una jugada posible derecha

aux4++;

if(matriz[pf1][pc1+1]=='-'){

aux3++;

}

}

if(pf1+1<=(n-1)){//verificar que haya una jugada posible aabajo

aux4++;//contador posiciones posibles

if(matriz[pf1+1][pc1]=='|'){

aux3++;

}

}

if(pc1-1>=0){//verificar que haya una jugada posible izquierda

aux4++;//contador posiciones posibles

if(matriz[pf1][pc1-1]=='-'){

aux3++;

}

}

if(aux4!=aux3){//punto ocupado,vuelve a pedir ingresar

aux1=1;

}

// printf("\njuagadas posibles:%d jugada hechas:%d",aux4,aux3);

aux3=0;

aux4=0;

//printf("\npunto 1 (%d , %d)",pf1 ,pc1);

}else if(pf1==99){

printf("Game over\n");

printf("\nPuntaje del user 1:%d",user1[0]);

printf("\nPuntaje del user 2:%d",user1[1]);

if (user1[0]>user1[1]){//compara vector de puntaje

printf("\n\nyou win");

}else if(user1[0]<user1[1]){

printf("\n\nyou lose");

}else{

printf("\n\nit's a tie");

}

//////////////////////////if

}

}

Con esta función, leemos los puntos a conectar para usuario y generamos aleatoriamente para la maquina en este caso, cada vez que leemos el punto 2 ya conectamos verticalmente u horizontalmente con un conjunto de if que verifican si esta libre, si es par, si esta dentro de la matriz, si esta a su derecha,arriba,abajo o izquierda del punto 1.

y se conectan

void punto2(void){

if(user==1){

printf("Ingrese el punto 2: \n");

printf("fila: ");

scanf("%d",&pf2);

pf2=pf2-(2-pf2);

}else{

pf2=rand()%(n);

//printf("\nfila random %d",pf2);

}

if(user==1){

printf("columna: ");

scanf("%d",&pc2);

pc2=pc2-(2-pc2);

}else{

pc2=rand()%(n);

//printf("\ncolumna random %d",pc2);

}

if (pf2==pf1-2 && pc2==pc1 && pf2>=0){//arriba

if(matriz[pf1-1][pc1]=='0' ){

matriz[pf1-1][pc1]='|';//conexion de puntos

lf=pf1-1;

lc=pc1;

aux=1;

// printf("\npunto 2 (%d,%d)",pf2,pc2);

}

}else if(pf2==pf1 && pc2==pc1+2 && pc2<=n){//derecha

if(matriz[pf1][pc1+1]=='0' ){

matriz[pf1][pc1+1]='-';//conexion de puntos

lf=pf1;

lc=pc1+1;

aux=1;

//printf("\npunto 2 (%d,%d)",pf2,pc2);

}

}else if(pf2==pf1+2 && pc2==pc1 && pf2<=n){//abajo

if(matriz[pf1+1][pc1]=='0' ){

matriz[pf1+1][pc1]='|';//conexion de puntos

lf=pf1+1;

lc=pc1;

aux=1;

//printf("\npunto 2 (%d,%d)",pf2,pc2);

}

}else if(pf2==pf1 && pc2==pc1-2 && pc2>=0){//izquierda

if(matriz[pf1][pc1-1]=='0' ){

matriz[pf1][pc1-1]='-';//conexion de puntos

lf=pf1;

lc=pc1-1;

aux=1;

//printf("\npunto 2 (%d,%d)",pf2,pc2);

}

}

}

Con esta función lo que hacemos es verificar, si la ultima conexion que hicimos fue una horizontal, verifica que abajo o arriba, derecha o izquierda contengan una línea en ese caso suma 10 al jugador o la maquina o si la ultima conexion fue vertical, de acuerdo a su posicion verifica las conexiones para sumar 10 al jugador que haya completado y muestra en pantalla quien completo el cuadrado junto a su color los puntajes del usuario y maquina almacenamos en un vector donde cada jugada se actualiza y se compara:

void verificar\_cuadrado(void){

//abajo

band=0;

if (matriz[lf][lc]=='-'){//si lo ultimo que conectaste fue un guion

if(matriz[lf-1][lc-1]=='|'){

band++;

}

if(matriz[lf-2][lc]=='-'){

band++;

}

if(matriz[lf-1][lc+1]=='|'){

band++;

}

if (band==3 && pf1!=99){

user1[user-1]=user1[user-1]+10;

printf("se completo un cuadrado");

if(user==1 && y==1){

matriz[lf-1][lc]='R';

} else if(user==1 && y==2){

matriz[lf-1][lc]='A';

} else if(user==2 && y==1){

matriz[lf-1][lc]='A';

} else if(user==2 && y==2){

matriz[lf-1][lc]='R';

}

}

band=0;

/// verificar cuadrados arriba

if(matriz[lf+1][lc-1]=='|'){

band++;

}

if(matriz[lf+2][lc]=='-'){

band++;

}

if(matriz[lf+1][lc+1]=='|'){

band++;

}

if (band==3 && pf1!=99){

user1[user-1]=user1[user-1]+10;

printf("\nse completo un cuadrado\n");

if(user==1 && y==1){

matriz[lf+1][lc]='R';

} else if(user==1 && y==2){

matriz[lf+1][lc]='A';

} else if(user==2 && y==1){

matriz[lf+1][lc]='A';

} else if(user==2 && y==2){

matriz[lf+1][lc]='R';

}

}

band=0;

// verificar conexion derecha vertical

}else if(matriz[lf][lc]=='|'){//si lo ult que pusiste fue linea vertical

if(matriz[lf-1][lc-1]=='-'){

band++;

}

if(matriz[lf][lc-2]=='|'){

band++;

}

if(matriz[lf+1][lc-1]=='-'){

band++;

}

if (band==3 && pf1!=99){

user1[user-1]=user1[user-1]+10;

printf("\nse completo un cuadrado\n");

if(user==1 && y==1){

matriz[lf][lc-1]='R';

} else if(user==1 && y==2){

matriz[lf][lc-1]='A';

} else if(user==2 && y==1){

matriz[lf][lc-1]='A';

} else if(user==2 && y==2){

matriz[lf][lc-1]='R';

}

}

band=0;

//verificar conexion izquierda vert

if(matriz[lf-1][lc+1]=='-'){

band++;

}

if(matriz[lf][lc+2]=='|'){

band++;

}

if(matriz[lf+1][lc+1]=='-'){

band++;

}

if (band==3 && pf1!=99){

user1[user-1]=user1[user-1]+10;

printf("\nse completo un cuadrado\n");

if(user==1 && y==1){

matriz[lf][lc+1]='R';

} else if(user==1 && y==2){

matriz[lf][lc+1]='A';

} else if(user==2 && y==1){

matriz[lf][lc+1]='A';

} else if(user==2 && y==2){

matriz[lf][lc+1]='R';

}

}

band=0;

}

}

Esta función imprime la matriz

void imprimir\_matriz(){

if (pf1!=99){

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

printf("\n");

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

printf("%c",matriz[i][j]);

}

}

}

}

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

Llamamos a las funciones:

inicio();

srand(time(NULL));

while (x==0){

tablero();

while (m2==0){

jugador();

}

carga\_matriz();

verificar\_aleatorio();

game\_over();

turnos();

while(aux1==0){

punto1();

}

if (pf1!=99){

aux=0;//condicion para que no pida el P2

}else{

aux=1;//condicion para que pida el p2

}

printf("\n");

while(aux==0){

punto2();

}

aux2=user1[user-1];//guardamos el valor del puntaje para verificar si aumenta

verificar\_cuadrado();

imprimir\_matriz();

if (user==1){//cambio de turno

if(aux2==user1[user-1]){// si puntaje anterior es igual al puntaje nuevo, juega maquina, si cambia entonces vuelve a jugar

user=2;

}

}else{

if(aux2==user1[user-1]){// si puntaje anterior es diferente al puntaje nuevo, juega usuario, si cambia entonces vuelve a jugar

user=1;

}

}

if (pf1!=99){

printf("\nPuntaje del user 1:%d",user1[0]);

if (user1[0]==user1[1]){//para empate

printf("\n\tEmpate");

}

printf("\nPuntaje del user 2:%d",user1[1]);

}

}

}

Fin del programa.