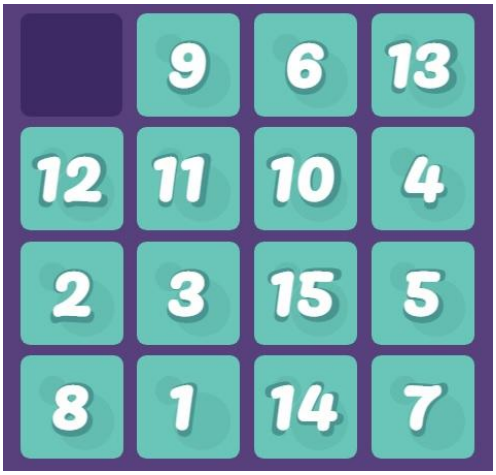
Documentación de ejercicio propuesto: Puzzle15

Índice

Objetivo

El objetivo del ejercicio es programar el juego Puzzle15, el cual consiste en ordenar numéricamente de menor a mayor un tablero que es otorgado con números del 1 al 15 inicialmente ordenados al azar.
En dicho tablero de 4 x 4 hay un espacio libre, el cual deberá ser ocupado por las fichas adyacentes para que sean ordenadas.
El juego termina cuando en el tablero se encuentran ordenados del 1 al 15 todos los números y el espacio libre se encuentra al final del tablero.

Inicio de partida:



Partida avanzada:

Partida finalizada:

Práctica en C

En este ejercicio se emplean los siguientes elementos de programación en C:

- Directivas de procesador: macros
- Librerías externas auxiliares: stdio.h y stdlib.h
- Array multi-dimensionales

El programa consta de dos archivos: globales.h, que contiene las macros y todas las variables que fueron definidas de manera global, y main.c, que utiliza las variables definidas en globales.h, en el cual se definen todas las funciones necesarias para hacer que el juego funcione.

Estructura de datos:

El programa inicia mostrando el tablero a resolver con los números ordenados de manera aleatoria (dentro de las posibilidades que se le dan) y consultándole al usuario en cuántos de movimientos en cree que resolverá el puzle, dándole un rango numérico.

Una vez recibido este dato, se almacena para dar comienzo con el reto, el cual, al finalizar la partida servirá como referencia para mostrar el puntaje final, dependiendo también de la otra variable que se actualiza con los movimientos del usuario.

Con cada tecla que se oprime, el programa reemplaza la celda vacía por el número de una celda adyacente y transforma en 0 al número de dicha celda, haciendo que esta quede vacía. A su vez, se reescribe la variable “movimiento” con el valor que tenía + 1, y se actualiza la cantidad de movimientos que le queda al usuario para llegar al número apostado por él.

Al terminar de ordenar los números de menor a mayor, el programa reconoce la posición final como la ganadora, mostrándote tu puntaje (que depende de la cantidad de movimientos realizados y la diferencia entre estos y la apuesta del usuario) y el puntaje histórico de las partidas anteriores.

En cualquier momento del juego está la posibilidad de dejar de jugar, cortando así el bucle principal que contiene todas las funciones correspondientes al movimiento de las fichas y actualizaciones de variables. También se puede reiniciar el juego con una variante de números al azar diferente a la que había salido en esa partida, haciendo que se reinicie el bucle principal sin salir de este.

Algoritmo y funciones importantes:

La función principal, main(), es la cual le da inicio al juego está conformada por dos iteraciones, una dentro de la otra. En ellas se incluyen la mayoría de las funciones que son desarrolladas posteriormente en el código.

```
int main(){
  int result = 1;
  do {                              // Este do representa el inicio de una partida desde o
    x = 0;                          // Se inicia la partida con la x y la y igual a o para que la ubicación
    y = 0;                          // matriz[0][0] sea reemplazada por los corchetes vacíos.
    indPartida++;
    obtenerPartida();               // Elije de manera random uno de los posibles inicios de partida
    mostrarTablero();               // Imprime el tablero con la partida a resolver
    apuesta = leerApuesta();  // Toma la apuesta que realiza el usuario y la guarda en la variante apuesta
    movimientos = 0;
    do {                            // Este do evalúa qué acción realizó el usuario y según cuál sea, sigue
                                    //  iterando o sale para volver al do anterior.
      mostrarTableroCompleto();   //Limpia lo que haya en pantalla para mostrar la versión final del
                                  //  tablero, la apuesta, cant de mov y la dif entre apuesta y mov.
                                  //  A su vez, se va a actualizando a medida que el usuario juega.
      if (result == 3)
        printf("El caracter es incorrecto \n");
      result = ejecutarMov();       //Esta función retorna un valor, el cual será guardado en result y a
                                    //partir de ahí se ejecutará un movimiento
      if (result == 1)
        movimientos++;
      if (iguales()) {          // La función iguales() compara si la matriz resuelta coincide con la del usuario.
                                // Lo que hace este if es decir "si se cumple la función iguales, mostrar el
                                // tablero resulto, con el puntaje correspondiente (mostrarPuntaje()), guardar
                                // los datos de la partida (guardarPartida()) y mostrarlo en el historial de
                                //partida(mostrarHistorial()).
        mostrarTableroCompleto();
        printf("Puzzle resuelto!\n");
        mostrarPuntaje();
        guardarPartida();
        mostrarHistorial();
        result = contJuego();           //consulta al usuario si quiere volver a jugar, según lo que devuelve,
                                        //vuelve a entrar al primer do o sale del programa
      }
    } while (result!= 0 && result!=-1);
  } while(result!=0);
  return 0;
}
```

Compilación y ejecución

Para complicar el programa realice las siguientes acciones:

1- Descomprimir en un directorio el archivo **puzzle**
2- Acceda al directorio generado
   # cd Puzzle15
3- Compile
   # gcc -Wall main.c -o main
4- Ejecute
   # ./main

- El programa se debería ver, en una primera instancia de esta forma:

```
[  ][ 9][11][ 3]
[ 5][14][ 7][ 1]
[13][ 8][15][ 2]
[ 4][10][ 6][12]

Cantidad de mov para resolver el puzzle (entre 40 y 200):...
```

- Una vez ingresado el número, la pantalla pasa a estar de esta forma:

```
[  ][ 9][11][ 3]
[ 5][14][ 7][ 1]
[13][ 8][15][ 2]
[ 4][10][ 6][12]

Apuesta:...      Movimientos:...       Quedan:...

Utiliza las teclas W A S D para moverte por el tablero. 0 para
finalizar, N para nuevo juego.
```

- Al terminar la partida se debería ver de esta forma:

```
[ 1][ 2][ 3][ 4]
[ 5][ 6][ 7][ 8]
[ 9][10][11][12]
[13][14][15][  ]

Apuesta:...      Movimientos:...       Quedan:...

Utiliza las teclas W A S D para moverte por el tablero. 0 para
finalizar, N para nuevo juego.

Puzzle resuelto!

Puntaje obtenido:... puntos

Part nro:  |Apuesta:  |Movimientos:  |Puntaje:
    1      |  ...     |    ...       |  ...
```