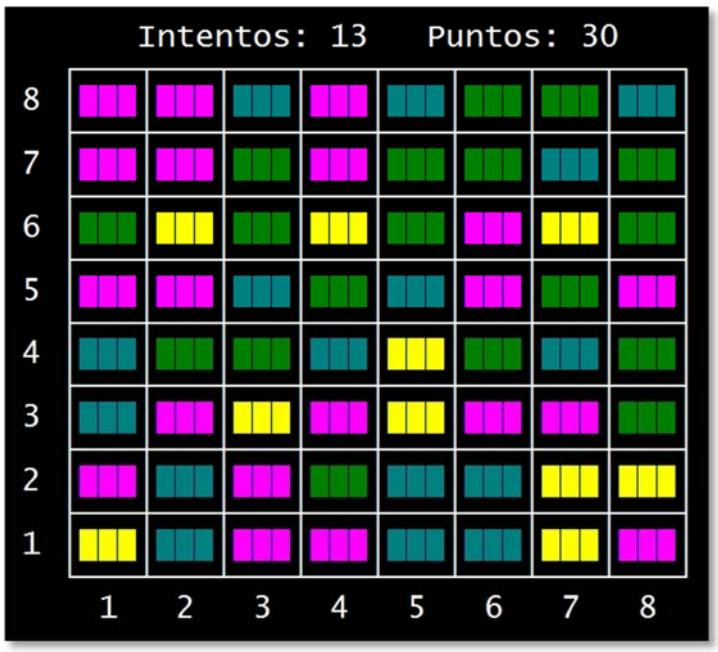
Práctica 3
Color Crush

Fecha de entrega: 30 de marzo de 2014



Color Crush es una versión muy simplificada de un famoso juego de Internet. Consiste en un tablero de N x N fichas de colores. El objetivo es conseguir el mayor número de puntos posible en el número de intentos disponible. Los puntos se consiguen alineando 3 o más fichas de un mismo color en horizontal o en vertical.

0. Mecánica del juego

En cada jugada se intercambian dos fichas consecutivas de forma que se consiga un *grupo* de fichas (un grupo de fichas son 3 o más fichas de un mismo color que están alineadas en horizontal o en vertical). Se añaden los puntos conseguidos y se eliminan del tablero las fichas que se han conseguido alinear. Al eliminarse las fichas quedan huecos en los que *caen* las fichas de la fila superior, dejando otros huecos que se llenan igualmente. Los huecos que quedan en la fila superior se llenan con fichas de colores generados aleatoriamente.

Al caer las fichas para tapar los huecos se pueden formar nuevos grupos de 3 o más fichas iguales, que se eliminan del tablero, contabilizándose los puntos, y se vuelven a rellenar los huecos.

1. Versión básica del programa

En su primera versión, el programa deberá permitir al usuario jugar una partida de *Color Crush*. La dimensión del tablero y el máximo de intentos serán constantes (por defecto, 8 y 15). Las fichas pueden ser de uno de cuatro colores posibles: magenta, amarillo, azul y verde.

El programa comenzará generando el tablero, asignando aleatoriamente un color a cada ficha. A continuación lo mostrará y lo procesará eliminando posibles grupos que se hayan formado inicialmente. El usuario tendrá entonces la posibilidad de realizar tantas jugadas como máximo de intentos haya.

En cada jugada se pregunta al usuario qué ficha quiere intercambiar, indicando la posición (fila y columna) y la dirección (arriba, abajo, izquierda o derecha):

Una vez comprobado que el movimiento es válido (se forma algún grupo de fichas) se realiza el intercambio, se muestra la nueva situación del tablero y se procesa éste en busca de grupos de fichas. El usuario puede introducir un 0 en cualquier momento para dar por terminado el juego.

1.1. Datos del programa

Las estructuras de datos son muy sencillas: el tablero es un array bidimensional de colores (`tTablero`). Utiliza un tipo enumerado `tFicha` con los colores posibles (añade un color *neutro* que represente que la celda está vacía).

En cada momento tenemos el tablero con una determinada disposición, un número de intentos restantes y una serie de puntos acumulados. Declara un tipo de estructura `tJuego` que contenga el tablero, el número de intentos restantes y los puntos conseguidos hasta el momento.

Cada movimiento del jugador consiste en una fila, una columna y una dirección. Declara un tipo enumerado `tDireccion` (`arriba`, `abajo`, `derecha`, `izquierda`) y un tipo de estructura `tMovimiento` que contenga la fila, la columna y la dirección.

Utilizarás otro tipo enumerado para los colores de la pantalla, pero eso te lo explicamos un poco más adelante.

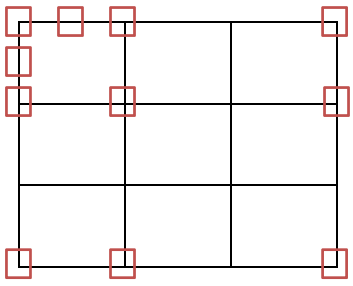
1.2. Subprogramas

El programa contendrá, al menos, subprogramas para las siguientes tareas:

- Generar aleatoriamente un tablero.
- Conducir una partida del juego.
- Comprobar si hay algún grupo a partir de una posición, en horizontal o en vertical.
- Comprobar si un movimiento de usuario es válido (fila, columna y dirección válidas, no se sale del tablero y genera algún grupo de fichas).
- Procesar el tablero eliminando los grupos de fichas existentes y contabilizando los puntos correspondientes. Cada vez que elimina un grupo de fichas debe volver a procesar el tablero desde el principio, ya que se pueden haber formado nuevos grupos de fichas al *caer* unas llenando los huecos de las eliminadas.
- Llenar los huecos de las fichas eliminadas dejando *caer* las fichas superiores sobre los huecos.
- Mostrar el tablero en la pantalla.

1.3. Visualización del tablero

El tablero se *dibujará* utilizando para las líneas caracteres de la parte extendida del código ASCII. Estos son los códigos que vas a necesitar: 218, 196, 194, 191, 179, 195, 197, 180, 192, 193 y 217. De arriba hacia abajo y de izquierda a derecha:



Para mejorar la visualización del tablero vamos a hacer uso de algunas utilidades de la biblioteca `Windows.h`. Por ejemplo, para mostrar los distintos colores.

Al incluir la biblioteca `Windows.h` podemos usar `SetConsoleTextAttribute()`, una rutina que permite establecer algunas propiedades de los caracteres que se muestren en la pantalla, tales como el color. Requiere que se le pase un *handle*, un

manejador de salida que haremos corresponder con la pantalla y un color, el que queramos establecer para los siguientes caracteres que se visualicen.

Para crear el *handle* disponemos del tipo incluido en `Windows.h` `HANDLE` y la función `GetStdHandle()` que lo crea asociado a un dispositivo de salida:

```
HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
```

`STD_OUTPUT_HANDLE` representa la pantalla.

Una vez creado el *handle* podemos utilizar `SetConsoleTextAttribute()` para establecer el color en el que queremos que se muestren los siguientes caracteres:

```
SetConsoleTextAttribute(handle, color);
```

Donde `color` será un valor de un tipo enumerado `tColor` declarado de la siguiente forma:

```
typedef enum { // Screen colors
   black,           //  0
   dark_blue,       //  1
   dark_green,      //  2
   dark_cyan,       //  3
   dark_red,        //  4
   dark_magenta,    //  5
   dark_yellow,     //  6
   light_gray,      //  7
   dark_gray,       //  8
   light_blue,      //  9
   light_green,     // 10
   light_cyan,      // 11
   light_red,       // 12
   light_magenta,   // 13
   light_yellow,    // 14
   white            // 15
} tColor;
```

Así, podemos crear una rutina para cambiar el color de los caracteres:

```
void setColor(tColor color) {
   HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
   SetConsoleTextAttribute(handle, color);
}
```

En la biblioteca `Windows.h` también hay alguna otra rutina que puede resultar interesante, como `Sleep()`, que detiene la ejecución del programa durante el número de milisegundos que se le pase como argumento.

Para contemplar cómo se van produciendo los acontecimientos sobre el tablero puedes volver a visualizar el tablero cada vez que se produce un cambio en su disposición (*ver caer las fichas*). Puedes ajustar el tiempo de actualización del tablero utilizando la función `Sleep()`.

El programa incluirá otra constante `Debug` que cuando esté a `true` hará que en lugar de ejecutarse `Sleep()` se ejecute `system("pause")`, de forma que la visualización quede detenida hasta que el usuario pulse una tecla, permitiendo ver con detenimiento todo lo que va ocurriendo en el tablero.

1.4. Puntuación

La puntuación en cada jugada será la suma de los puntos de cada grupo de fichas conseguido, directa o indirectamente: 3 puntos si se han alineado 3 fichas, 8 puntos si se han alineado 4 y 15 puntos para grupos de 5 fichas alineadas. Recuerda que en el tablero inicial puede haber grupos de fichas que deben ser igualmente puntuados.

2. Posibles mejoras (elige al menos dos)

La versión final del programa deberá incluir la versión básica [8 puntos] y al menos dos mejoras entre las posibles mejoras que se indican continuación:

- [1 punto] Contemplar la existencia de una *gelatina* en cada casilla que se elimina de aquellas en las que se ha formado un grupo de fichas. Cambiar el objetivo del juego a eliminar toda la *gelatina*.
- [0,25 puntos] Mantener un archivo con las puntuaciones de los distintos jugadores. Solicitar el nombre del jugador al principio y actualizar los datos de cada usuario que juegue.
- [0,5 puntos] Permitir salvar partidas y recuperarlas.
- [0,25 puntos] Cargar tableros de archivo. Preguntar si se quiere cargar un tablero (solicitar el nombre del archivo) o que se genere aleatoriamente.

3. Entrega de la práctica

La práctica se entregará a través del Campus Virtual. Se habilitará una nueva tarea **Entrega de la Práctica 3** que permitirá subir el archivo con el código fuente y otros posibles de puntuaciones, juegos guardados o tableros para cargar.

Fin del plazo de entrega: **30 de marzo a las 23:55**