

Fundamentos de la programación

Grado en Desarrollo de Videojuegos

Examen parcial Final, Junio 2016

Indicaciones generales:

- Se entregará un único archivo `Program.cs` (generado con MonoDevelop) con el programa completo.
 - La línea 1 será un comentario de la forma `// Nombre Apellido1 Apellido2`.
 - La línea 2 será un comentario de la forma `// Numero de laboratorio, Puesto`.
 - Lee **atentamente** el enunciado e implementa el programa tal como se pide, con los métodos, parámetros y requisitos que se especifican.
 - El programa debe ser correcto, estar bien estructurado y comentado. Se valorarán la claridad, la concisión y la eficiencia.
 - **Entrega:** la entrega se hará a través del servidor FTP de los laboratorios.
 - **Plantilla:** en el servidores FTP puede encontrarse la plantilla para programa pedido, que incorpora la clase de las listas enlazadas vista en clase (la utilizada en la última práctica del *snake*).
-

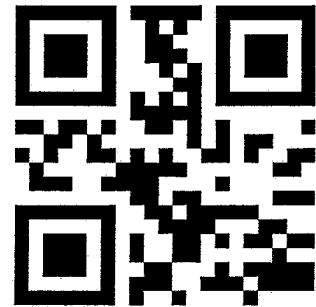
Los *códigos QR* son matrices cuadradas de puntos que sirven para almacenar información diversa. Estas matrices pueden ser distintos tamaños, pero a nosotros nos interesarán las de 21×21 . Para que el lector pueda detectar la posición del código, tal como se muestra en la figura, hay tres *cuadrados de control* en las esquinas superiores y en la esquina inferior izquierda.

En nuestro caso vamos a representar estos códigos QR mediante una matriz cuadrada de booleanos de dimensión 21×21 , donde `true` representa negro y `false` blanco. Implementaremos algunas operaciones con ellos. Para empezar, en la plantilla ya está definido el método

```
static bool [,] generaQRAleatorio()
```

que genera un código QR aleatorio válido, i.e., con los cuadrados de control correspondientes. Este método servirá para probar el resto de métodos pedidos. Se pide, implementar los siguientes métodos:

- [1 pt] `void dibuja(bool [,] qr)`: dibuja en pantalla el código QR con cuadrados blancos y negros, tal como se ha explicado. Para obtener una figura proporcionada, cada cuadrado de la matriz se dibujará con dos posiciones horizontales en pantalla (tal como hemos hecho en algunas prácticas de laboratorio).
- [1 pt] `bool [,] generaPatron()`: genera una matriz de tamaño 7×7 con el patrón de un cuadrado de control tal como se muestra:



Este patrón es un cuadrado negro de 3×3 , inscrito en otro cuadrado blanco de 5×5 , a su vez inscrito en otro negro de 7×7 .

- [2 pt] `bool estaPatron(bool [,] pat, bool [,] qr, int x, int y)`: comprueba si el patrón de la matriz `pat`, se encuentra en la matriz `qr` en las coordenadas (x,y) . Por ejemplo, el patrón obtenido con el método anterior (`generaPatron`) se encuentra en el código `qr` del ejemplo inicial en las esquinas esperadas, i.e., en las coordenadas $(0,0)$ y también en $(0,14)$ y $(14,0)$.
- [1 pt] `bool qrValido(bool [,] qr)`: comprueba si el código `qr` es válido utilizando los dos métodos anteriores, es decir, comprueba si `qr` contiene los tres cuadrados de control en las esquinas correspondientes. Además debe comprobar que dichos cuadrados de control estén separados del resto del código por franjas blancas, tal como se muestra en la figura inicial.
- [1 pt] `void escribeSalida(bool [,] qr, string file)`: salva el código `qr` en el archivo `file`. Para ello escribe cada uno de los valores de la matriz `qr` en una línea de dicho archivo. Recordemos que los archivos de salida abren de la forma `StreamWriter f = new StreamWriter (file)`.
- [1 pt] `bool [,] leeEntrada(string file)`: lee del archivo `file` la información de un código QR en el formato salvado por el método anterior lo devuelve la matriz correspondiente. Recordemos que los archivos de entrada abren de la forma `StreamReader f = new StreamReader (file)` y los booleanos se pueden leer directamente de archivo con `bool.Parse(f.ReadLine ())`.
- [2 pt] Diseña un método `convierte` que tome como entrada una matriz de booleanos (correspondiente a un código QR) y obtenga una lista enlazada que contenga las coordenadas de los puntos negros de ese código (no importa el orden y no se almacena nada para los blancos). Deben decidirse los parámetros del método y la forma de paso de los mismos. Por ejemplo, una matriz de la forma:



producirá una lista enlazada con las coordenadas $(0,0)$, $(0,2)$, $(0,3)$, ..., $(1,1)$, $(1,3)$, ...

La clase de las listas enlazadas ya está implementada en la plantilla proporcionada y no hay que hacer ninguna modificación sobre dicha clase.

- [1 pt] `int cuentaRec(ListaEnlazada l)`: dada una lista enlazada `l` con las coordenadas de los puntos negros de un código QR, **calcula recursivamente** el número de puntos negros de ese código. Puede asumirse que sobre dicha lista ya se ha realizado la operación de `iniciaRecorrido`.

Para subir nota: implementar una versión recursiva final para de este mismo método.

Las soluciones iterativas (con bucles) no puntuarán nada en este apartado.