
SISTEMAS MULTIMEDIA

Examen de la convocatoria de Junio

CURSO 2023-2024

Nombre:

DNI:

Para realizar este examen se creará un proyecto en NetBeans de nombre "ExamenJunio2024". Cada ejercicio irá ampliando el anterior, de forma que el proyecto final englobe la totalidad de las preguntas.

El examen se divide en dos bloques: gráficos e imágenes. Para aprobar el examen es necesario superar cada bloque de forma independiente. Para cada pregunta, se indica el peso (%) en su bloque¹.

BLOQUE I. Gráficos

1. Crear una aplicación que permita dibujar elipses, cada una con sus atributos propios de color, relleno y grosor. Para ello, se creará una *VentanaPrincipal* que incluirá en su interior un lienzo (para lo cual se creará una clase propia); además, la ventana tendrá una barra de herramienta que incluya (1) tres botones de colores, (2) un botón de dos posiciones para el relleno y (3) un *deslizador* para el grosor. El lienzo mantendrá todas las elipses que se vayan dibujando, pintándose cada una de ellas con el color, relleno y grosor que haya seleccionados en cada momento en la barra de herramientas. En el caso del relleno, será relleno liso. 40%

Las clases relativas a formas y atributos de dibujo que se definan para resolver este ejercicio deberán tener exactamente la misma estructura que se ha diseñado para la práctica de evaluación (en caso contrario, se considerará erróneo el ejercicio en su totalidad)².
2. Incluir un botón "edición" de dos posiciones, que permita activar/desactivar el modo edición. Por defecto, estará desactivado, un cuyo caso se dibujarán elipses como se indica en el ejercicio 1. Si se activa el modo edición, se podrán modificar los atributos de la primera figura del vector sin más que usar los correspondientes botones de la barra de herramientas; en este modo no se podrán dibujar elipses 35%
3. Añadirle al lienzo una imagen que se muestre como fondo; inicializarlo con una imagen de tamaño 400x400 y de color blanco. Incluir un botón en la barra de herramientas que, al pulsarlo, vuelque la primera figura del vector en la imagen (y la elimine del vector)³. 25%

¹ En caso de que un bloque no se apruebe, la nota final del examen será el mínimo de las notas de ambos bloques. En caso de que ambos bloques se superen, la nota final será la media ponderada de ambos bloques, con un peso de 0.45 para el bloque de gráficos y de 0.55 para el de imágenes.

² Aunque solo será necesario implementar la clase asociada a la elipse, deberá de hacerse siguiendo la jerarquía y estructura de clases y métodos entregada en la práctica de evaluación. Para los atributos, bastará centrarse en los exigidos en el ejercicio (color, relleno y grosor); de nuevo, habrá que implementarlos (clases, métodos, etc.) de la misma forma que se hiciera en la práctica de evaluación.

³ Si se pulsa este botón de forma consecutiva, si irán volcando figuras hasta que, finalmente, todo el vector esté vacío (y las figuras dibujadas en la imagen)

BLOQUE II. Imágenes

4. Incluir en la ventana principal un botón “abrir” que, al pulsarlo, lance el diálogo abrir y, una vez seleccionada la imagen, ésta se muestre en el lienzo de la ventana. Añadir dos botones en la barra de herramientas: uno que permita aumentar el brillo en 10 y otro que intercambie las bandas roja y azul. 10%
5. Implementar un operador LookupOp que permita el siguiente efecto: invertir el color de las zonas oscuras manteniendo el de las zonas claras. Incluir un comentario en la cabecera del método que justifique por qué la solución responde a la operación que se solicita. Incluir en la ventana principal un botón que, al pulsarlo, aplique el operador anterior a la imagen que se esté mostrando en ese momento en el lienzo. 20%
6. Implementar un filtro de convolución que simule el efecto “movimiento vertical oscurecido”. El filtro de movimiento se define como un filtro de emborronamiento con una dirección dada (horizontal, vertical, etc.); el objetivo de este ejercicio es combinar, en una sola máscara, ese emborronamiento con un efecto de oscurecimiento. Incluir un comentario en la cabecera del método que justifique por qué la solución responde a la operación que se solicita.

Incluir un deslizador que permita variar el tamaño de la máscara (p.e., entre 1 y 30). Al desplazar el deslizador, deberá de aplicarse el filtro anterior a la imagen original usando el correspondiente tamaño de máscara; en la ventana se irá mostrando el resultado de aplicar el filtro sobre la imagen original. Mostar en la consola de salida el valor de la máscara aplicada⁴. 30%
7. Definir una clase correspondiente a un nuevo operador “SegmentaOp”, con parámetro $T \in [0,255]$, que haga lo siguiente: para aquellos píxeles cuya intensidad supere el umbral T , mantendrá su color original; para aquellos que sean inferiores al umbral, le asignará un color C predeterminado (segundo parámetro del operador). Formalmente: 40%

$$g(x,y) = \begin{cases} f(x,y) & \text{si } I(x,y) \geq T \\ C & \text{si } I(x,y) < T \end{cases}$$

con $f(x,y)$ la imagen original, $g(x,y)$ la imagen resultado, $T \in [0,255]$ el umbral, $C = [r_c, g_c, b_c]$ un color, e $I(x,y)$ la intensidad del píxel calculada como la media de sus componentes $I(x,y) = (r(x,y) + g(x,y) + b(x,y))/3$

Incluir en la ventana principal un deslizador (con rango $[0,255]$) asociado al parámetro T ; al desplazar el deslizador, deberá de aplicarse el operador anterior a la imagen original usando el correspondiente valor de umbral (en la ventana se irá mostrando el resultado de aplicar la operación). Como color (segundo parámetro) usar el color seleccionado en la barra de herramientas (ejercicio 1).

■ Anexo: copia de una imagen

El siguiente código, visto en la práctica 9, crea una copia de la imagen “imagen” y la asigna a la variable “imagenFuente”:

```
ColorModel cm = imagen.getColorModel();
WritableRaster raster = imagen.copyData(null);
boolean alfaPre = imagen.isAlphaPremultiplied();
imgFuente = new BufferedImage(cm,raster,alfaPre,null);
```

⁴ El siguiente código muestra el vector “m” en la salida: `System.out.println(Arrays.toString(m))`;