

Sistemas Multimedia - Práctica Final. Documentación.

Adrián Portillo Sánchez

Junio 2017

Índice

1. Introducción	2
2. Funcionalidades principales de la aplicación	2
2.1. Funcionalidades principales.	3
2.2. Reproductor de audio.	6
2.3. Reproductor de vídeo.	6
3. Análisis de requisitos de la aplicación	7
4. Funcionalidades opcionales desarrolladas.	9
5. Operación propia randomizerOp	9
6. Clase propias del sistema: Config.	10
7. Conclusiones sobre la asignatura.	10

Índice de figuras

1. Vista principal de la aplicación, con un dibujo simple en ella. . .	2
2. Reproductor de audio de la aplicación.	6
3. Reproductor de vídeo de la aplicación.	6
4. Diagrama de clases de la aplicación.	8

1. Introducción

En este segundo cuatrimestre del curso 2016-2017 he cursado la asignatura Sistemas Multimedia, en la cual se tiene como objetivo enseñar la programación y desarrollo de distintos sistemas multimedia, sistemas que se encargan de reproducir, transmitir, mostrar y editar medios digitales, como pueden ser imágenes, audio o vídeo.

En las prácticas de esta asignatura se ha realizado una aplicación completa de multimedia, con la capacidad de dibujar y procesar imágenes, de reproducir y grabar audio, y de reproducir y grabar vídeo desde una webcam; la cual está documentada en este documento, así como en su respectivo javadoc y será mostrada para la evaluación de la asignatura.

Dicha aplicación se ha realizado en java utilizando las bibliotecas de Graphics2D e Image2D para los sistemas de dibujo y procesamiento de imágenes, así como diversas librerías para los sistemas de audio y vídeo. Veamos la aplicación.

2. Funcionalidades principales de la aplicación

A continuación veremos el aspecto principal de la aplicación y sus funcionalidades.

Esta es la ventana principal de la aplicación, en ella tenemos abiertas dos imágenes, una dibujada por la misma aplicación con algo de texto y unas figuras sencillas y otra de una imagen obtenida del pc para editar mediante las opciones de edición de imágenes de la aplicación.

A continuación veremos, partiendo de esta ventana principal, las funcionalidades de nuestra aplicación.

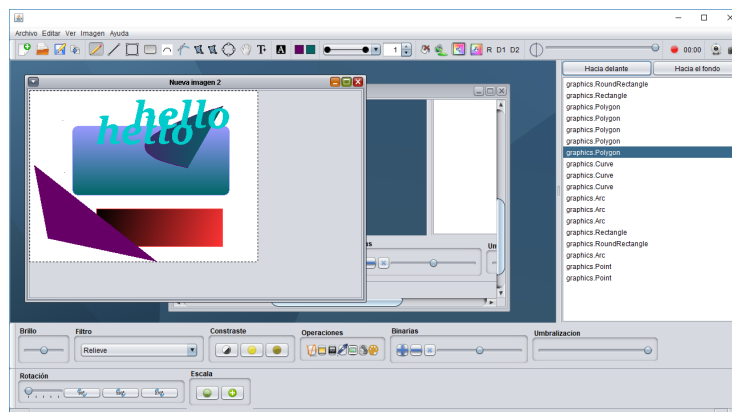


Figura 1: Vista principal de la aplicación, con un dibujo simple en ella.

2.1. Funcionalidades principales.

En la ventana principal de la aplicación podemos apreciar las distintas funcionalidades que posee:

1. En la barra superior, de izquierda a derecha:

- **Archivo.**
 - Nueva imagen.
 - Abrir medio (Imagen, audio o vídeo).
 - Guardar imagen.
 - Duplicar imagen.
- **Formas y herramientas de dibujo.**
 - Forma: Punto.
 - Forma: Línea.
 - Forma: Rectángulo.
 - Forma: Rectángulo curvo.
 - Forma: Arco.
 - Forma: Curva.
 - Forma: Polilínea.
 - Forma: Elipse.
 - Herramienta: Mano.
 - Herramienta: Texto: introducimos el texto al pulsar.
- **Fuente:** Abre un diálogo para seleccionar fuente y formato del texto.
- **Color:** A la izquierda color frontal y a la derecha color trasero.
- **Estilo de línea.**
 - Seleccionar estilo: continuo, discontinuo suave o discontinuo fuerte.
 - Seleccionar grosor de línea.
- **Relleno.**
 - Sin relleno.
 - Con relleno plano.
 - Degradado izquierda a derecha.
 - Degradado arriba a abajo.
 - Degradado radial.
 - Degradado diagonal 1.
 - Degradado diagonal 2.

- **Suavizado.**
 - Antialiasing, activa/desactiva el suavizado de bordes.
 - Slider de transparencia.
 - **Grabación de audio.**
 - Botón de grabar: pulsar para empezar a grabar y pulsar de nuevo para terminar grabación.
 - Tiempo de grabación actual.
 - **Grabación de vídeo.**
 - Grabar webcam: activa la webcam por defecto y comienza a grabar vídeo.
 - Snapshot: toma una captura del vídeo o imagen actuales.
2. **En el centro tenemos las ventanas con las imágenes editables.**
 3. **A la derecha el histograma, que sigue un registro de los objetos dibujados en la imagen.**
 4. **Por último, en la parte inferior, de izquierda a derecha:**
 - **Slider de brillo:** Ajusta el brillo de la imagen.
 - **Filtros:**
 - Emborronamiento medio: genera un efecto de desenfoque suave.
 - Emborronamiento binomial: genera un efecto de desenfoque algo más fuerte.
 - Enfoque: Aumenta la nitidez del enfoque.
 - Relieve: Crea un efecto relieve en los bordes.
 - Detector de fronteras laplaciano: Elimina las fronteras y los espacios en blanco.
 - **Contraste.**
 - Contraste: añade contraste a la imagen.
 - Contraste brillante: añade contraste entre objetos brillantes a la imagen.
 - Contraste oscuro: añade contraste entre objetos oscuros a la imagen.

- **Operaciones.**
 - Seno: altera la imagen matemáticamente usando una función sinusoidal.
 - Sepia: altera la imagen para dejarla sepia.
 - Gradiente Sobel: altera la imagen en un gradiente blanco y negro para detección de bordes.
 - Tintado: aplica un tintado del color seleccionado a la imagen.
 - Negativo: pasa la imagen a negativo.
 - Blanco y negro: aplica un filtro de blanco y negro.
 - Operación propia: Randomizador: aplica un filtro donde cada punto es modificado aleatoriamente un poco de su color original, excepto blancos y negros que diferirán mucho más.
- **Operaciones binarias.**
 - Suma: Realiza una operación de suma entre las dos imágenes seleccionadas.
 - Resta: Realiza una operación de resta entre las dos imágenes seleccionadas.
 - Multiplicación: Realiza una operación de multiplicación entre las dos imágenes seleccionadas.
 - Slider de combinación: combina las dos imágenes en una combinación ponderada según la posición del slider.
- **Umbralización:** Permite realizar una umbralización de color ajustada al slider.
- **Rotación**
 - Slider de rotación para rotación libre.
 - Rotar 90°.
 - Rotar 180°.
 - Rotar 270°.
- **Aumentar y disminuir zoom.**



Figura 2: Reproductor de audio de la aplicación.

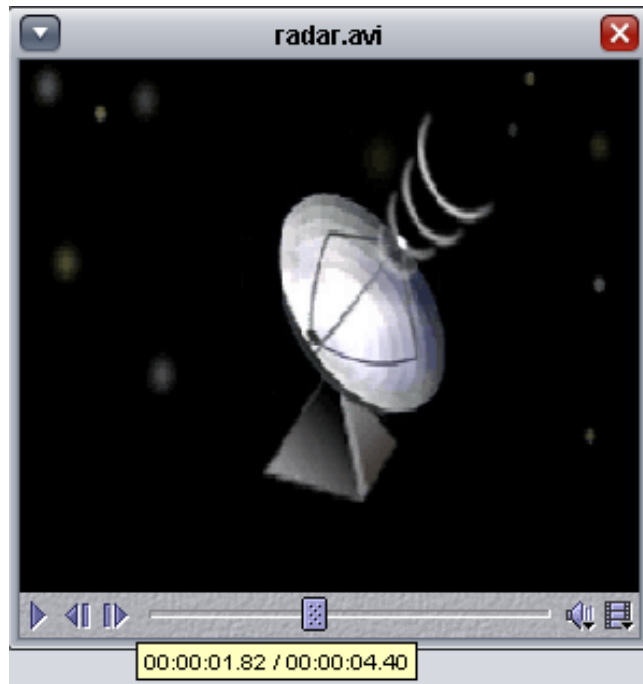


Figura 3: Reproductor de vídeo de la aplicación.

2.2. Reproductor de audio.

Una vez grabemos un audio, podemos reproducirlo o abrir también un audio, entonces se abrirá el siguiente reproductor sencillo.

Su funcionalidad es muy básica, pulsamos play para reproducir el archivo y se parará cuando termine su reproducción, no tenemos opción de pausa ni de avance rápido ni nada por ese estilo.

2.3. Reproductor de vídeo.

También podremos abrir vídeos, actualmente sólo en .avi pues mi aplicación aún no está muy pulida, y se abrirá en el siguiente reproductor.

Este posee algo más de funcionalidad, podremos pausar y reiniciar, podremos saltar a distintos puntos del vídeo, cambiar el volumen del vídeo y ver información relativa al mismo; así como avanzar y retroceder un fotograma con los dos botones de al lado del play.

3. Análisis de requisitos de la aplicación

1. RF1 Crear ventana principal del programa.
2. RF2 Crear ventana de imagen que contenga un lienzo.
3. RF2.1 Duplicar la ventana de imagen seleccionada y su lienzo.
4. RF3 Permitir dibujar figuras en el lienzo.
5. RF3.1 Permitir dibujar figuras geométricas básicas (Rectángulo, Elipse, Línea...).
6. RF3.2 Permitir dibujar figuras avanzadas (Curva, Polígono, Línea...).
7. RF3.3 Permitir seleccionar una figura y cambiar sus parámetros (color, trazo).
8. RF3.4 Permitir seleccionar una figura y moverla sobre el lienzo.
9. RF4 Permitir la introducción de texto con formato en el lienzo.
10. RF5 Permitir abrir imágenes desde archivo.
11. RF5.1 Permitir guardar imágenes en archivo.
12. RF6 Permitir aplicar operaciones sobre la imagen seleccionada (Negativo, Brillo, Relieve...).
13. RF7 Crear ventana de visualización de webcam.
14. RF7.1 Permitir realizar una instantánea de la ventana webcam.
15. RF8 Crear ventana de vídeo.
16. RF8.1 Permitir realizar una instantánea de la ventana de vídeo.
17. RF8.2 Permitir controlar la reproducción de video (Play, Stop...).
18. RF9 Crear ventana de audio.
19. RF9.1 Reproducir audio en dicha ventana.
20. RF10 Permitir la grabación de audio.

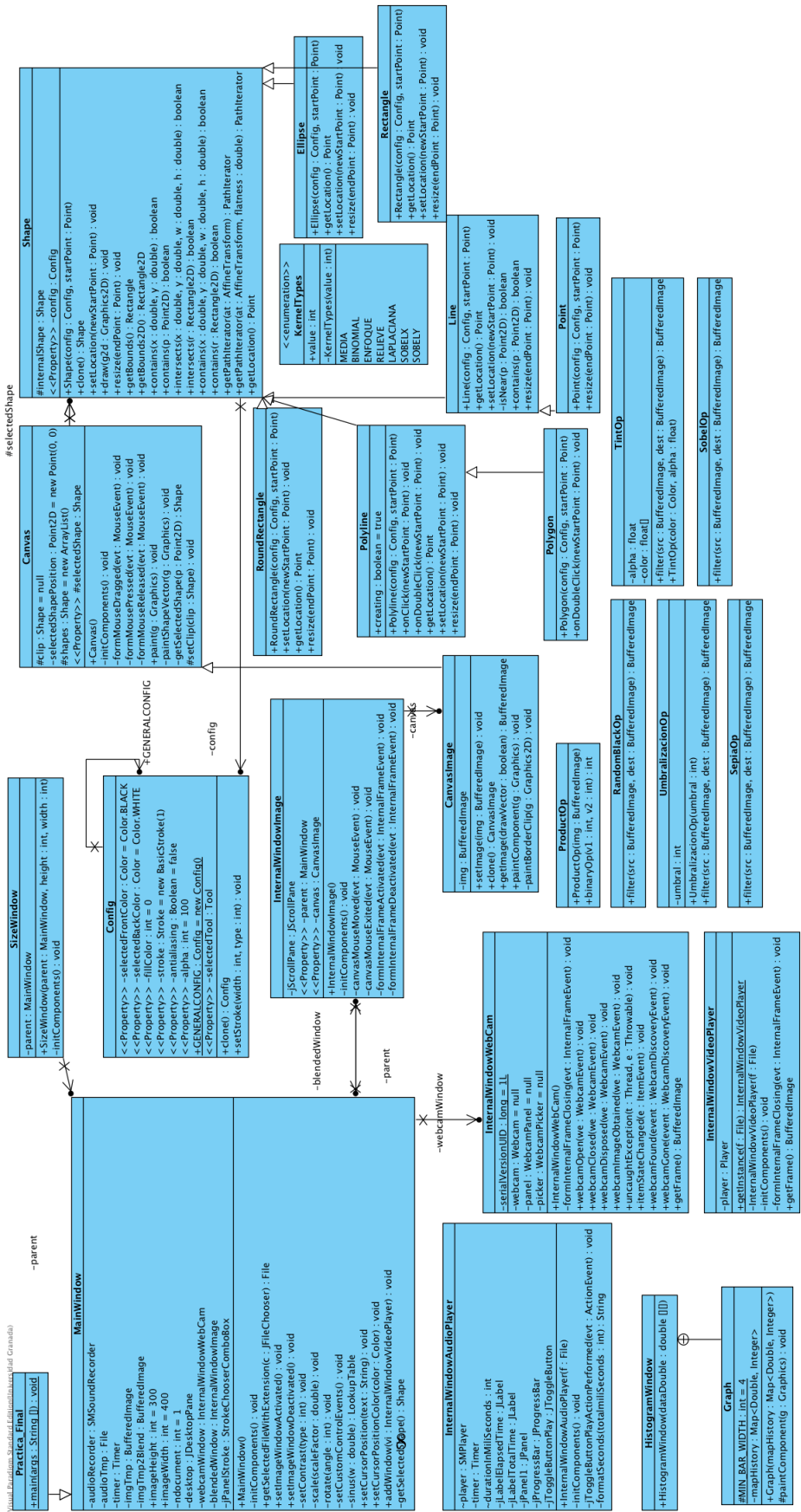


Figura 4: Diagrama de clases de la aplicación.

4. Funcionalidades opcionales desarrolladas.

Ya que algunas funcionalidades no eran obligatorias, me tomo la libertad de exponer cuales de las funcionalidades desarrolladas opcionales he desarrollado:

1. Formas:

- Arco.
- Polilínea.
- Polígono.

2. Dos tipos de trazos discontinuos, trazos largos y cortos.

3. Relleno:

- Degradado radial.
- Dos degradados diagonales.

4. Operaciones sobre imagen:

- Detección de fronteras sobel.
- Histograma.

5. Operaciones binarias:

- Suma de imágenes
- Resta de imágenes
- Mezcla mediante deslizador

5. Operación propia randomizerOp

```
for (BufferedImagePixelIterator it = new BufferedImagePixelIterator(src); it.hasNext();)
{
    pixel = it.next();

    pixel.sample[0] = pixel.sample[0] +
        (int)((5 * Math.random()) - (5 * Math.random()));
    pixel.sample[1] = pixel.sample[1] +
        (int)((5 * Math.random()) - (5 * Math.random()));
    pixel.sample[2] = pixel.sample[2] +
        (int)((5 * Math.random()) - (5 * Math.random()));

    destRaster.setPixel(pixel.col, pixel.row, pixel.sample);
}
```

Esta sencilla operación le suma a cada canal del color de cada pixel un valor en el intervalo [-5,5], dando así resultados bastante curiosos especialmente en fotografías, ya que los colores extremos como el blanco y el negro cambian muchísimo, mientras que colores intermedios cambian muy poco, además es fácilmente aplicable esta operación varias veces a la misma imagen para así llegar a un resultado deseado.

6. Clase propias del sistema: Config.

Una clase propia es la llamada Config donde se almacena la configuración de las figuras en el momento actual de tiempo, dicha instancia se clona cada vez que se crea una figura nueva con lo que cada "Shape" tiene su propia instancia de config, así es muy sencillo alterar las propiedades dependiendo de cual sea la figura seleccionada.

De esta forma la implementación de la figura queda separada de la implementación de las configuraciones e unida simplemente mediante instanciación de esa clase. Config facilita también la configuración del texto y otras configuraciones generales como antialiasing o brillo.

7. Conclusiones sobre la asignatura.

Creo que ha sido una asignatura en la que he aprendido mucho sobre el procesamiento de imágenes, aunque pienso que en los aspectos de reproducción y grabación de audio la parte de programación ha quedado muy limitada, pero lo veo lógico, ya que un reproductor es una aplicación creo yo demasiado concreta para trabajar en profundidad.

Pienso que los conocimientos prácticos aprendidos de Java especialmente sobre las librerías Graphics2D e Image2D pueden ser útiles en un futuro, aunque por mi parte la mayor utilidad obtenida de dicha asignatura está en la parte práctica ya que he aprendido mucho sobre cosas que conocía tan sólo superficialmente como son los codecs, los formatos de imagen, etc.

Referencias

- [1] Librería utilizada para la conexión con la webcam: <http://webcam-capture.sarxos.pl/>
- [2] Selección de trazo: <http://www.jfree.org/jcommon/api/org/jfree/ui/StrokeChooserPanel.html>
- [3] Otras dudas: <http://stackoverflow.com/questions/tagged/java>