Gráficos avanzados - Ejercicios

Índice

1. Dibujo básico

Vamos a probar una aplicación básica que dibuje contenido en la pantalla utilizando la API de bajo nivel. En el directorio `DibujoBasico` de las plantillas de la sesión tenemos implementada esta aplicación.

a) Consultar el código y posteriormente compilar y probar la aplicación.

b) Modificar el anchor de posicionamiento del texto para probar los distintos alineamientos existentes.

c) Probar a cambiar las propiedades de los objetos dibujados (fuente, color, etc).

d) Añadir animación. Para ello crearemos un hilo que modifique las variables `x` e `y` a lo largo del tiempo y llame a `repaint` para dibujar en la nueva posición. Utilizar el evento `showNotify` para poner en marcha este hilo.

2. TeleSketch (I)

Vamos a implementar una aplicación similar al juego que se conocía como *"TeleSketch"*. Esta aplicación nos deberá permitir dibujar en la pantalla utilizando las teclas de los cursores.

La idea es dibujar en *offscreen* (en una imagen mutable), de forma que no se pierda el contenido dibujado. En cada momento conoceremos la posición actual del cursor, donde dibujaremos un punto (puede ser un círculo o rectángulo de tamaño reducido). Al pulsar las teclas de los cursores del móvil moveremos este cursor por la pantalla haciendo que deje rastro, y de esta manera se irá generando el dibujo.

Tenemos una plantilla en el directorio `TeleSketch`. Sobre esta plantilla deberemos realizar lo siguiente:

a) En el constructor de la clase deberemos crear una imagen mutable donde dibujar, con el tamaño del `Canvas`.

b) En el método `actualiza` deberemos dibujar un punto en la posición actual del cursor y llamar a `repaint` para repintar el contenido de la pantalla.

c) En `paint` deberemos volcar el contenido de la imagen *offscreen* a la pantalla.

d) Deberemos definir los eventos `keyPressed` y `keyRepeated` para mover el cursor cada vez que se pulsen las teclas arriba, abajo, izquierda y derecha. Podemos utilizar las acciones de juegos (game actions) para conocer cuáles son estas teclas.

3. TeleSketch (II)

Ampliar la aplicación anterior permitiendo cambiar el color y el grosor del lápiz. Para hacer esto podemos añadir una serie de comandos con un conjunto de colores y grosores preestablecidos.

4. TeleSketch (III)

Si dibujamos usando los eventos de repetición el cursor muchas veces se moverá muy lentamente. En lugar de utilizar este evento para dibujar podemos crear un hilo que ejecute un bucle infinito. Dentro de este hilo mientras la tecla siga pulsada se irá moviendo el cursor, de esta manera se actualizará con la frecuencia que nosotros queramos sin tener que esperar al evento de repetición. Sabremos que una tecla se mantiene pulsada desde que se invoca un evento `keyPressed` hasta que se invoca un evento `keyReleased` para la misma tecla.

5. TeleSketch (IV)

Utilizar la API de Nokia para dibujar en un *canvas* a pantalla completa.

6. Gráficos 3D

En el directorio `Prueba3D` de las plantillas de la sesión podemos encontrar un ejemplo de aplicación M3G. En ella podemos ver un ejemplo de modo inmediato y otro de modo *retained*. Consultar el código y probar el ejemplo. Para poderlo probar necesitaremos al menos WTK 2.5.

Una vez probada la aplicación, vamos a realizar una serie de modificaciones en la parte del modo *retained* (`Visor3DRetained`). Se pide:

a) Realizar una función recursiva que explore e imprima en la consola todos los elementos del grafo de escena cargado. Deberá mostrarse para cada elemento el nombre de la clase a la que pertenece y la lista de sus hijos. De cada uno de estos hijos deberá mostrarse la misma información, y así recursivamente hasta explorar todo el arbol.

b) Hacer que la aplicación cargue el contenido del fichero `mundo.m3g` que se encuentra entre los recursos de la aplicación, en lugar de `caja.m3g`. Comprobar que la nueva escena se visualiza correctamente.

c) La escena se ve algo oscura. Añadir luz ambiente con intensidad 0.8 para mejorar la visibilidad. Para ello se deberá añadir el correspondiente objeto `Light` al grafo de la escena.

d) Dar la posibilidad de navegar por el mundo de forma subjetiva, utilizando para ello las teclas del cursor. Para hacer esto deberemos objeter la cámara activa del mundo, y

transformar su posición según las pulsaciones de las teclas. Para girar la cámara deberemos hacerlo alrededor del eje Y, ya que este eje es la altura en el mundo 3D y esto nos hará girar a la izquierda o a la derecha.

Recomendación

Será conveniente tener nuestro ángulo actual almacenado en un campo de nuestra clase. Podemos inicializar la cámara con ángulo 0º alrededor de Y, utilizando para ello el método `setOrientation`. Cada vez que se pulsen los cursores izquierdo o derecho, se modificará el ángulo actual, y se establecerá este ángulo en la cámara utilizando ese mismo método.

Para avanzar o retroceder, deberemos hacerlo en la dirección en la que estemos mirando. Para ello se deberán utilizar las funciones `cos` y `sin` de la clase `Math` sobre nuestro ángulo actual, para avanzar la correspondiente distancia en los ejes X y Z.

Nota

Se debe llevar cuidado, ya que la función `setOrientation` trabaja con ángulos en grados, mientras que `cos` y `sin` lo hacen en radianes. Se deberán utilizar los métodos `toRadians` y/o `toDegrees` de la clase `Math` para convertir entre ambas unidades.

e) En lugar de manejar la cámara subjetiva, vamos a hacer que lo que se maneje mediante los cursores sea el coche que aparece en la escena. El grupo de objetos que componen el coche está almacenado en el fichero M3G con el identificador 101. Podemos localizar un determinado elemento mediante su identificador utilizando el método `find` del mundo.

Localizar el objeto del coche, y hacer que las transformaciones se apliquen a este objeto, en lugar de a la cámara activa, para conseguir de esta forma que lo que se mueva sea el coche.

Nota

Es posible que haya que intercambiar las funciones `sin` y `cos` y sus signos para que el coche se mueva en la dirección y sentido correctos, ya que la orientación con la que está almacenado el coche en el fichero es diferente de la de la cámara.

f) El grupo del coche, a parte de las piezas que lo componen, incorpora una cámara con la que se ve el coche desde detrás. Vamos a hacer que la cámara siga el coche conforme lo movemos. Para ello estableceremos como cámara activa la cámara del coche. Esta cámara tiene un identificador 102.

g) Al movernos por el mundo podemos ver que el plano de corte posterior de la cámara es demasiado cercano, y esto produce que los objetos sólo aparezcan cuando nos acercamos lo suficiente. Hacer que el plano de corte posterior sea más lejano (utilizando el método `setPerspective` de la cámara), para que podamos ver los tres molinos al mismo tiempo desde un punto lo suficientemente lejano.

h) Cambiar el fondo del mundo por un fondo de color anaranjado.