



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Un caso práctico de configuración y desarrollo de aplicaciones *MultiTouch* con MT4j

Apellidos, nombre	Agustí Melchor, Manuel ¹ (magusti@disca.upv.es) Torres Pardo, Miguel Angel (mitorpars@eui.upv.es)
Departamento	¹ Dpto. De Ing. De Sistemas y Computadores
Centro	Universidad Politécnica de Valencia



1 Introducción

Este trabajo está enfocado en la tecnología *multitouch* y las posibilidades que nos brinda su desarrollo. En un principio dedicado a investigar las posibilidades del *touchpad* de un portátil y la incorporación de múltiples ratones. Con los resultados obtenidos se va a profundizar con una tableta *Bamboo Touch CTT-460* y el paquete *Mt4j* desarrollado en Java que permite ejecutar aplicaciones comprobando y explorando la arquitectura *multitouch*.

2 Objetivos

El objetivo marcado en una primera instancia es la configuración y adaptación del *touchpad* de un ordenador portátil, en este caso un acer aspire 4310, para que se comporte como un dispositivo *multitouch*. Pero a medida que se va avanzando en el proyecto el objetivo marcado va variando ante la imposibilidad de conseguir el objetivo propuesto. Primero en el uso de MPX, uso simultáneo de dos punteros en Unix y posteriormente en la configuración y adaptación de una tableta *Bamboo Touch CTT-460*.

En la segunda parte vamos a tratar de mejorar y realizar nuevas funciones en la aplicación SVG del paquete *Mt4j* versión 09. Partiremos con el *touchpad* de un ordenador portátil, en este caso un acer aspire 4310 y de una tableta *Bamboo Touch CTT-460*, aunque se podrán añadir hasta dos ratones más y aprovechar la función de puntero múltiple.

3 Desarrollo

En un principio el objetivo era el de configurar el *touchpad* del portátil para que respondiera de forma *multitouch* asemejando al comportamiento que tiene el *iphone* y una vez conseguido partir desde ese punto para explotar este recurso. En primer lugar instalamos el driver *synaptics_v10.2.4.0_allOS_modded_b2* el cual nos dará esta posibilidad, pero no observamos cambios ni que se modifique el driver instalado en ese momento, probamos incluso en 3 portátiles diferentes (sony vaio vgn-fz18m, acer aspire 4310 y acer aspire 4320) con MS/Windows XP y Vista y con diferentes formas de proceder sin la respuesta esperada. Nos damos cuenta de ello porque hay una aplicación de *synaptics* que muestra las pulsaciones que se hacen en el *touchpad* que se llama *Moodpad* el cual si pulsamos con dos dedos obviamente entra en conflicto.

Probando en sistemas Unix, mas concretamente con Ubuntu 9.10 Karmic Koala, procedemos editando el fichero `/etc/hal/fdi/policy/11-x11-synaptics.fdi` que controla el dispositivo del *touchpad* del ratón y reseteamos el servicio para que los cambios surtan efecto. Ejecutando la orden *synclient* en un Terminal obtendríamos las coordenadas de pulsación en el *touchpad* y nos identificaría incluso con cuantos puntos estamos haciendo contacto (*multitouch*), pero el sistema operativo nos dice que no tenemos habilitado el acceso a memoria oculta, el cual se controla con la variable *SHMConfig* en *xorg.conf* pero nuevamente nos encontramos con otra negativa tras varios intentos y pruebas diferentes.



Figura 1. Versión touchPad V6.5 original



Figura 2. Versión touchPad 6.3 que ofrece multitouch

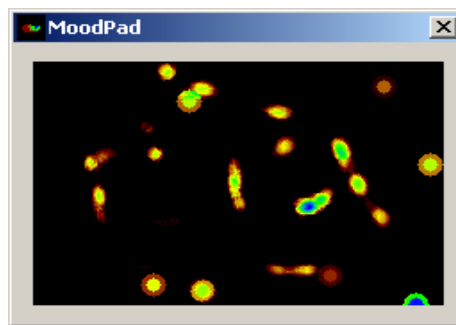


Figura 3. MoodPad de synaptics

La orden exacta es "synclient -l" y la respuesta del sistema es exactamente "Can't access shared memory area. SHMConfig disabled?". La respuesta esperada debía haber sido semejante a la siguiente imagen:

```
time  x  y  zf  w  l  r  u  d  m  multi  gl  gm  gr  gdx  gdy
0.000  1 5855  0 0 0 0 0 0 0 00000000  0 0 0 0 0
18.685 2482 2250 44 1 8 0 0 0 0 00000000  0 0 0 0 0
18.735 2489 2247 43 1 8 0 0 0 0 00000000  0 0 0 0 0
18.785 2656 2233 39 1 10 0 0 0 0 00000000  0 0 0 0 0
```

Figura 4. Respuesta de la orden synclient -l

Donde cada columna representa un valor como las coordenadas o la cantidad de dedos que tocan el *touchpad* al mismo tiempo.

Después de estas experiencias negativas nos damos cuenta de la complejidad que tiene configurar el *touchpad* y que el problema puede estar en que el hardware del que disponemos no está preparado para soportar esta función. Mirando en los foros vemos que le pasa a mucha gente el mismo problema con la variable SHMConfig y no han podido resolverlo.

Por el motivo anterior cambiamos de objetivo, que será el uso de 2 ratones simultáneos con 2 punteros independientes, de forma que podamos abrir varias aplicaciones y trabajar en la misma pantalla como si se tratara de 2 usuarios diferentes. Obviamente nos centraremos en sistemas Unix ya que en windows parece una labor muy compleja por razones obvias, así que procedemos a

modificar el archivo `xorg.conf` que controla el servidor X de sistemas Unix, probando en el mismo Ubuntu 9.10 que anteriormente. El problema que aparece es que este archivo no existe ya que desde Ubuntu 7.04 no se utiliza, se usa `hal` (*the Hardware Abstraction Layer*), pero si se edita y se crea `xorg.conf` el sistema operativo lo utilizará. La forma de proceder es crearlo y arrancar el sistema para que así de forma automática Ubuntu configura y crea las líneas del fichero acorde con todos los dispositivos hardware que tiene nuestra máquina. Editando el fichero nos vamos a la sección de los dispositivos y añadimos la línea del nuevo ratón y a continuación creamos el dispositivo con sus propias opciones. Finalmente arrancamos el sistema y comprobamos que no aparecen 2 punteros y entre los 2 ratones hardware que usamos entre ellos se crean conflicto. Después de diferentes configuraciones y probar múltiples opciones observamos que los resultados siguen sin ser los esperados.

Ya que hasta ahora todas las pruebas han sido con resultado negativo conseguimos una *Bamboo Touch modelo CTT-460* la cual ya está preparada para soportar acciones *multitouch*. La instalamos y configuramos en empezamos a explorar con aplicaciones que soporten *multitouch*.

Una de las características y funcionalidades más destacadas es el manejo del navegador web Mozilla Firefox, el cual podemos movernos por la web con dos dedos desplazándolos tanto vertical como horizontalmente y también ampliar o alejar la web juntando o alejando dos dedos.

Otra de las funciones que hemos experimentado en MS/Windows es el zoom y volteo de imágenes con el visualizar que viene por defecto, para voltear colocamos los dedos separados y dibujamos media circunferencia con ellos

Entre los diferentes paquetes que encontramos para probar esta tecnología en MS/Windows encontramos `Mt4j`, `TuioSimulator` y `reactIVision 1.4`, nombrar que existen otros paquetes como `gkaindl` o `tbeta` de *Nuigroup* para MacOS que no profundizaremos por no disponer del hardware necesario. En el caso de `reactIVision` se requiere una cámara para poder ejecutarlo.

Conseguimos el paquete `MT4j` el cual conseguimos editar con Eclipse y compilamos los archivos de *examples*, más concretamente nos centramos en *ModestMapsMT*. En esta aplicación tenemos un mapamundi con total funcionalidad *multitouch*, hacemos zoom alejando o separando los dedos y nos movemos usando también dos dedos.

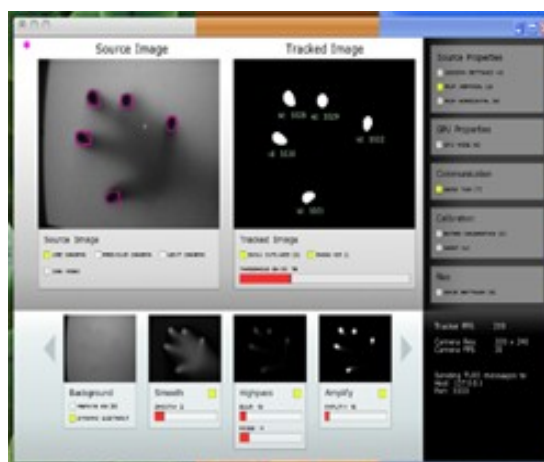


Figura 5. Aplicación *tbeta* en MacOS

Centrándonos en el uso de la tableta Bamboo en sistemas Unix comprobamos en primer lugar como en los requerimientos del hardware no aparece por ningún sitio éste sistema operativo, siendo requisitos MS/Windows XP, Vista (con SP2), 7 y MacOS X 10.4 o superior y proporcionándonos los drivers en un CD. Visto la negativa por parte del fabricante en la web conseguimos un enlace a una web externa <http://linuxwacom.sourceforge.net> que nos proporciona el paquete linuxwacom-0.8.4-4.tar.bz2 con fecha 2009-11-13 soportado por kernels 2.6.31 o superiores y por Xorg 1.6.0 o superiores también. Descomprimos y ejecutamos makefile y el configure y obtenemos otra vez errores con el archivo xorg.conf.

Con los resultados obtenidos nos centraremos con el paquete Mt4j que podemos descargar desde la web <http://www.mt4j.org>, más en concreto emplearemos la aplicación SGV, la cual nos permite mover y manipular imágenes SVG (*Scalable Vector Graphics*), donde para más información podemos visitar la siguiente página web <<http://www.w3.org/Graphics/SVG/>>. Trabajaremos con la versión v09 y utilizando el entorno de programación *Eclipse*.

Una vez tenemos instalado *Eclipse*, lo ejecutamos e importamos el proyecto desde un fichero, donde seleccionamos el paquete. En una primera instancia al compilar para probar la aplicación nos da errores y no podemos avanzar, investigando y observando nos damos cuenta de que faltan librerías. En concreto las librerías MT4j del paquete src, hay que descargarse la versión full y no la binaria para poder compilar los ejemplos. Destacar que nos descargamos un ZIP y podemos trabajar directamente en él o bien descomprimirlo y trabajar en la carpeta descomprimida.

Trabajando ya con la aplicación SVG procedemos a ejecutar el paquete teniendo ya la tableta *multitouch* conectada al computador, en una primera instancia aparecen las 3 imágenes por defecto y observamos que la característica de *multitouch* no es reconocida ya que hacemos movimientos específicos y observamos un solo puntero sin obtener la respuesta que esperamos.

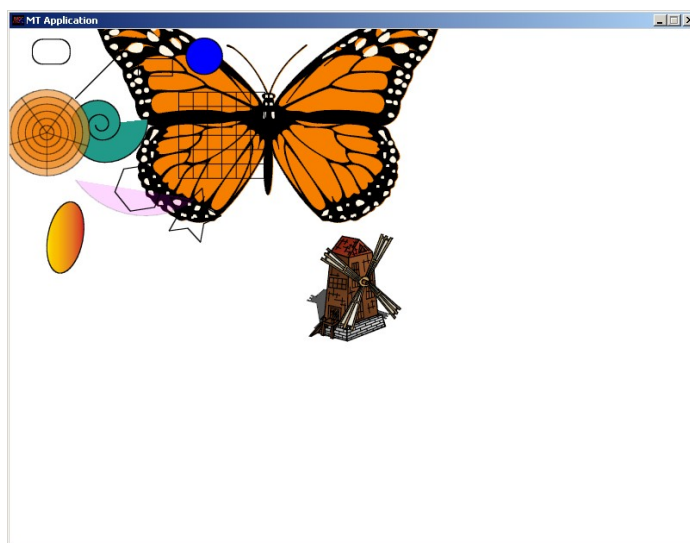


Figura 6. Vista aplicación original

En primer lugar para la tardanza de carga de la aplicación introduciremos una barra de carga, así el usuario no estará con la pantalla blanca y sabrá que la aplicación se está cargando correctamente. Para ello crearemos PantallaCargandoMain que será una clase con las funciones y constructor correspondientes y que además contendrá la imagen que será mostrada mientras la barra vaya cargándose.

A continuación creamos la clase PantallaCargando que contiene la programación y configuración de la barra, en la función velocidadDeCarga personalizamos la velocidad con la que aumentará el porcentaje de carga y si queremos que se muestre algún mensaje al mismo tiempo. El archivo está comentado para una rápida comprensión.

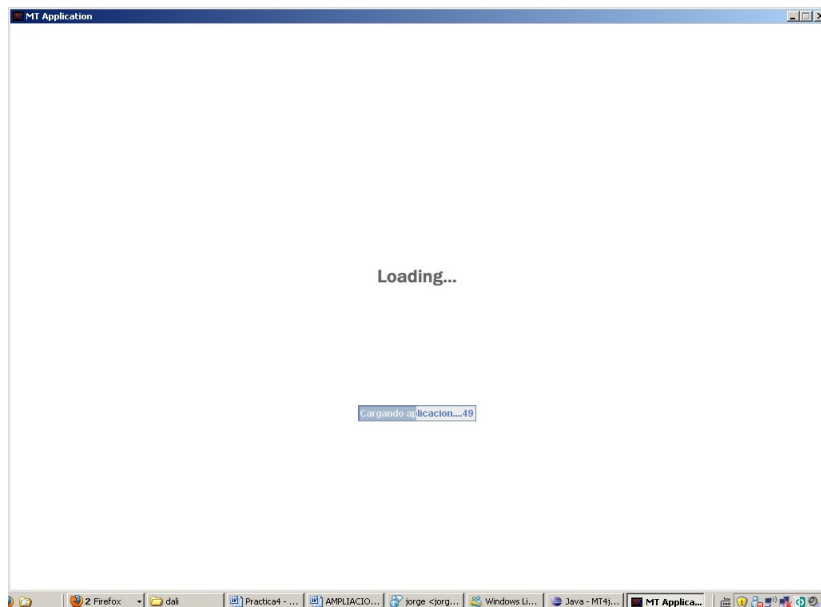


Figura 7. Pantalla de carga

El paso siguiente será ampliar la funcionalidad, actualmente tan sólo carga las 3 imágenes que están predefinidas limitando mucho al usuario así que nuestro objetivo será que cargue todas las imágenes que estén dentro de la carpeta data. Para ello modificaremos el archivo SVGScene que es el encargado de la carga de los archivos.

La idea principal es declarar una variable de tipo File con la ruta donde almacenaremos los archivos y recorrerla con un bucle for, ejecutando la instrucción facilitada por la aplicación en cada archivo, teniendo en cuenta que para obtener el nombre de la foto invocaremos al método toString que cada objeto de java hereda de la clase Object.

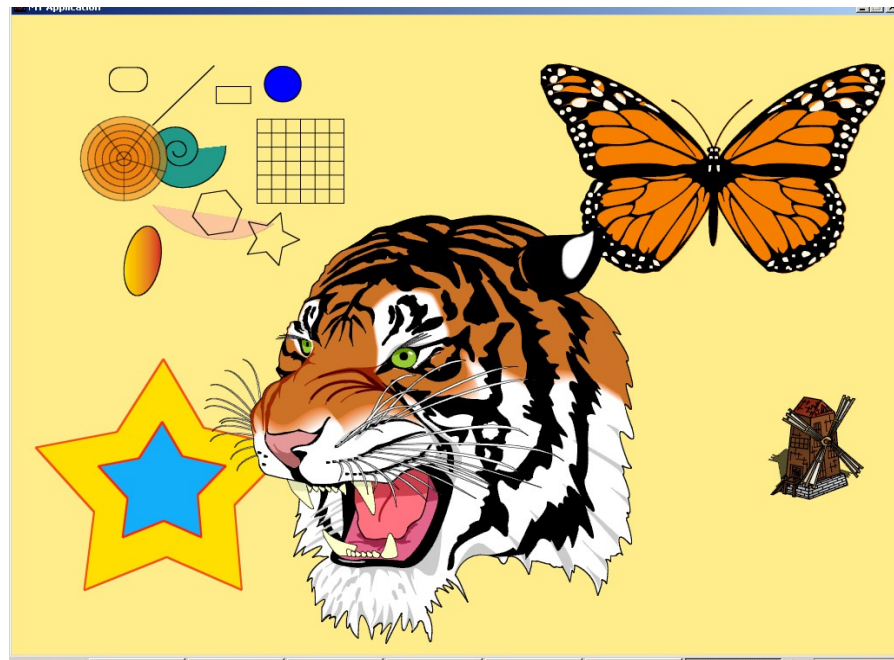


Figura 8. Vista final aplicación modificada

Podemos mencionar que con la variable `setClearColor` podemos personalizar el color de fondo de la aplicación siendo de tipo RGB y con el canal alpha.

Probando la aplicación apreciamos en primer lugar que no se comporta de forma *multitouch*, lo que nos lleva a explorar todos los archivos de texto que incluye el paquete y encontramos en que existe un fichero de nombre `settings.txt` que controla los siguientes parámetros:

```
#Wed Jan 30 04:24:13 CET 2008
```

```
Frametitle=MT Application
```

```
#Renderer: OpenGL: 1 P3D: 2
```

```
Renderer=1
```

```
#Fullscreen:'true' or 'false'
```

```
Fullscreen=false
```

```
#Switch for using fullscreen exclusive mode
```

```
 #(if not exclusive, mt4j just uses an undecorated window, other windows can overlap it)
```

```
FullscreenExclusive=false
```

```
#Resolution
```

```
DisplayWidth=1024
```

```
DisplayHeight=768
```

```
#Which display to use for fullscreen
```

```
Display=1
```

```
#Framerate maximum
```



```
MaximumFrameRate=60
#Vertical_sync:'true' or 'false'
Vertical_sync=false
#Enables multiple mice use (on windows)
MultiMiceEnabled=true
#Anti aliasing multisampling - '0' disables it - 2,4,8 determines the quality of
the a.a.
OpenGLAntialiasing=0
```

Entre otros valores podemos controlar si queremos ejecutar la aplicación a pantalla completa, con `true` o `false`. También podemos definir la resolución de ejecución que por defecto es 1024x768 ó el máximo `FrameRate` para las aplicaciones que simulan movimiento. Pero el valor con más relevancia sin duda es el de `MultiMiceEnabled` ya que si no cambiamos el valor de `False` que viene por defecto a `True` no podremos utilizar varios punteros en ningún ejemplo del paquete.

Concluyendo podemos decir que con esta ampliación hemos conseguido ampliar el ejemplo SVG del paquete `Mt4j v09` con el que podríamos seguir añadiendo funciones dispares para programar una aplicación compleja y lo más importante a saber como habilitar la tecnología *multitouch* del paquete para explotar todos los ejemplos que incluye.

4 Cierre

Podemos concluir con todas estas experiencias que la tecnología *multitouch* ya está implantada hoy día en muchos dispositivos aunque todavía queda un largo camino por recorrer en su desarrollo. El hardware fabricado con anterioridad a esta tecnología representa graves problemas para soportar esta funcionalidad ya que los drivers no están preparados en su mayoría.

En el uso de dos punteros simultáneos sólo vemos que sea posible en sistemas Unix y pese que no hemos podido conseguir el objetivo creemos que con el hardware y versión de sistema operativo apropiados si que sería posible obtener nuestro objetivo.

Con la tableta *Bamboo* y los drivers adecuados en MS/Windows XP no encontramos ningún problema y podemos explotar la tecnología *multitouch*, la cual cada día está cobrando más importancia.

Ha sido una sorpresa comprobar como el fabricante da la espalda a sistemas Unix y como la instalación pueda ser tan tediosa y problemática, todo lo contrario que con windows o MacOS.

Respecto a la aplicación *multitouch* `Mt4j` podemos concluir que se pueden añadir multitud de características y propiedades nuevas utilizando las herramientas adecuadas y multiplataforma ya que está programado en Java.



5 Bibliografía

<http://www.taringa.net/posts/linux/2365776/Truco:-habilitar-multitouch-en-Linux.html>

<http://www.mt4j.org/mediawiki/index.php/Examples>

<http://www.tuio.org/?software>

<http://www.bulma.net/body.phtml?nIdNoticia=1823&nIdPage=2>

http://en.wikipedia.org/wiki/Multi-Pointer_X

<http://multipointtuio.codeplex.com/>

<http://www.kulturafree.com.ar/kultura/?p=70>