

DISEÑO DE INTERFACES CON PROCESSING

Miguel Angel Ruiz Gálvez

Visita: miguelo.me

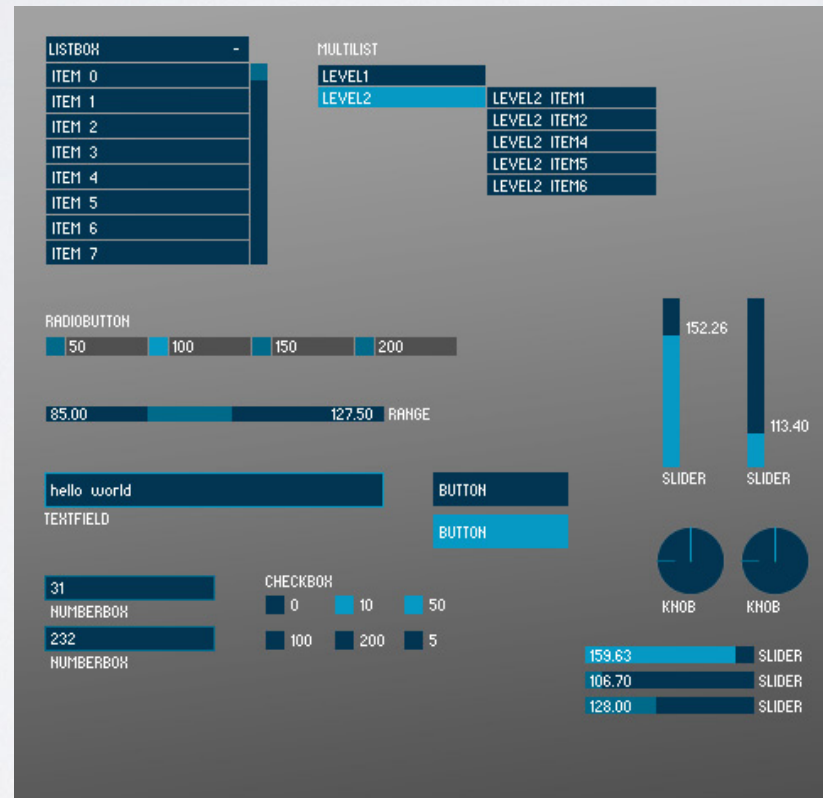
Material en:

SOMEFI

Este documento está licenciado bajo la Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita <http://creativecommons.org/licenses/by-nc-sa/4.0/>.

¿QUE ES UNA INTERFAZ?

La interfaz gráfica de usuario o GUI (graphical user interface) es un programa que actúa como puente entre el usuario y el programa, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles.



{ JSON }

- JSON es el acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos.
- La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML. Una de las ventajas de JSON es que es mucho más sencillo escribir y analizar.

```
{ "menu": {  
  "id": "5",  
  "platillos": "3",  
  "Comida": {  
    "PlatoFuerte": [  
      { "platillo": "pollo al cilantro", "precio": "70" },  
      { "platillo": "albondigas de res", "precio": "60" },  
      { "platillo": "filete de pescado", "precio": "90" }  
    ]  
  }  
}
```

{ JSON }

```
JSONObject json = new JSONObject();  
JSONArray values = new JSONArray();
```

```
saveJSONObject(json, "data/new.json");  
json = loadJSONObject("data.json");
```

```
json.setInt("count", 88);  
json.setFloat("weight", 35.432);  
json.setString("name", "celery");  
json.setBoolean("isFruit", false);
```

```
int count = json.getInt("count");  
float weight = json.getFloat("weight");  
String name = json.getString("name");  
boolean isFruit = json.getBoolean("isFruit");
```

- Processing permite crear, leer, escribir y guardar JSON utilizando los tipos de dato **JSONObject** y **JSONArray**
- Es posible cargar y guardara archivos con terminación **.json** utilizando la función **load** y **save** respectivamente.
- Se utilizan las funciones **.get** para leer datos almacenados en un id.
- Se utiliza la función **.set** para escribir un dato con un id característico.

SONIDOS

Minim es una librería de audio que utiliza la API JavaSound, un poco de Tritonus y MP3SPI de Javazoom para proporcionar un fácil utilizar la biblioteca de audio para processing. La filosofía detrás de la API es hacer que la integración de audio en sus programas sea tan simple como sea posible sin dejar de ofrecer una cantidad razonable de flexibilidad para los usuarios más avanzados..

Estas son algunas de las características de Minim:

- [AudioPlayer](#): reproducción mono y estéreo de archivos WAV, AIFF, AU, SND, y MP3.
- [AudioMetaData](#): Un objeto lleno de metadatos de un archivo, archivos ID3.
- [Audiorecorder](#): grabación de audio mono y estéreo.
- [AudioInput](#): monitorización de entrada mono y estéreo.
- [AudioOutput](#): síntesis de sonido mono y estéreo.
- [FFT](#): realizar una transformada de Fourier de los datos de audio para generar un espectro de frecuencia.
- [BeatDetect](#): una clase para hacer la detección de ritmo.

SONIDOS

```
import ddf.minim.*;  
Minim minim;  
minim = new Minim(this);
```

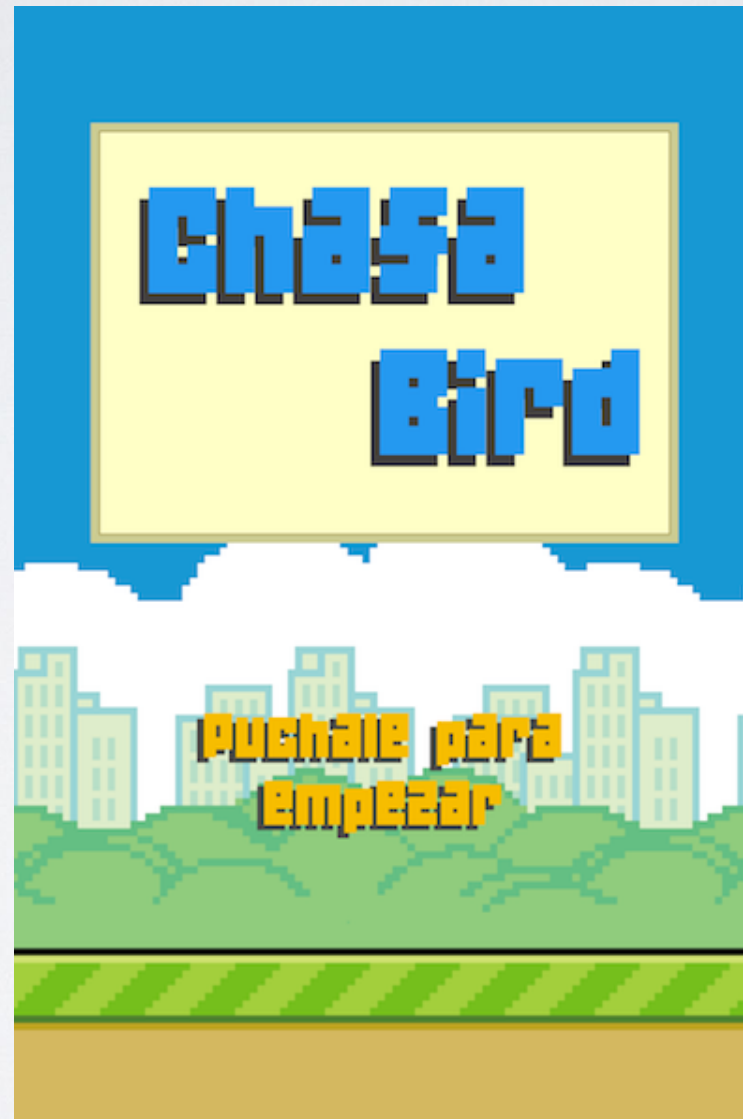
```
AudioPlayer track;  
track = minim.loadFile("Sounds/theme.mp3");  
track.play();  
track.pause();  
track.rewind();
```

```
AudioSample efecto;  
efecto = minim.loadSample("Sounds/hit.mp3");  
efecto.trigger();
```

- Se utiliza los **AudioPlayer** para elemento sonidos de larga duración como canciones, fondos o pistas.
- **AudioSample** se utilizan para efectos de audio de poca duración.
- Para cargar los archivos se utiliza la función load.
- Es posible modificar el estado de un **AudioPlayer** con los comandos **play()**, **puse()** y **rewind()** entre otros.
- Para activar un efecto de sonido (**AudioSample**) se utiliza la función **trigger()**.

Nota: Más información de esta librería en: <http://code.compartmental.net/tools/minim/>

¡CHAFA BIRD!



GRAFICA

Grafica es una librería de uso sencillo y configurable para Processing. Ayuda a crear fácilmente gráficas, tablas, diagramas.

- Gráficas de dispersion y líneas en tiempo real.
- Histogramas verticales y horizontales.
- Añadir varias capas con propiedades diferentes a la misma grafica.
- Funciona tanto con escalas lineales y logarítmicos.
- Ejes automáticos.
- Función de zoom y paneo Interactivo.
- Añadir etiquetas a tus puntos y mostrarlos con un solo clic.
- Es de código abierto. **grafica** está bajo la Licencia Pública General GNU. Usted puede encontrar el código fuente completo en: <https://github.com/jagracar/grafica>

GRAFICA

```
import grafica.*;  
GPlot plot = new GPlot(this);
```

```
plot.setPos(x, y);  
plot.setDim(ancho, alto);  
plot.setXLim(min, max); plot.setYLim(min, max);  
plot.setTitleText("Titulo");  
plot.getYAxis().setAxisLabelText("Eje Y");  
plot.getXAxis().setAxisLabelText("Eje X");  
plot.setPoints(arreglo GPoints);
```

```
plot.defaultDraw();  
plot.beginDraw();  
plot.drawBackground();  
plot.drawBox();  
plot.drawXAxis();  
plot.drawYAxis();  
plot.drawTitle();  
plot.drawPoints();  
plot.drawFilledContours(GPlot.HORIZONTAL, 0);  
plot.drawLabels();  
plot.endDraw();
```

- La clase GPlot corresponde a los parámetros de la gráfica, es posible modificar multiples características para lograr un grafico personalizado.

GRAFICA

```
GPointsArray points = new GPointsArray(tamaño);
```

```
points.add(indice,x, y);  
points.removeRange(inicio,fin);  
points.remove(indice);  
float x = points.getX( );  
float y = points.getY( );
```

- La clase GPointArray corresponde a un arreglo de valores X y Y que serán requeridos por la clase GPlot para poder graficar.
- Tiene un gran parecido al manejo de arrays en Processing.

XLS READER (EXCEL)

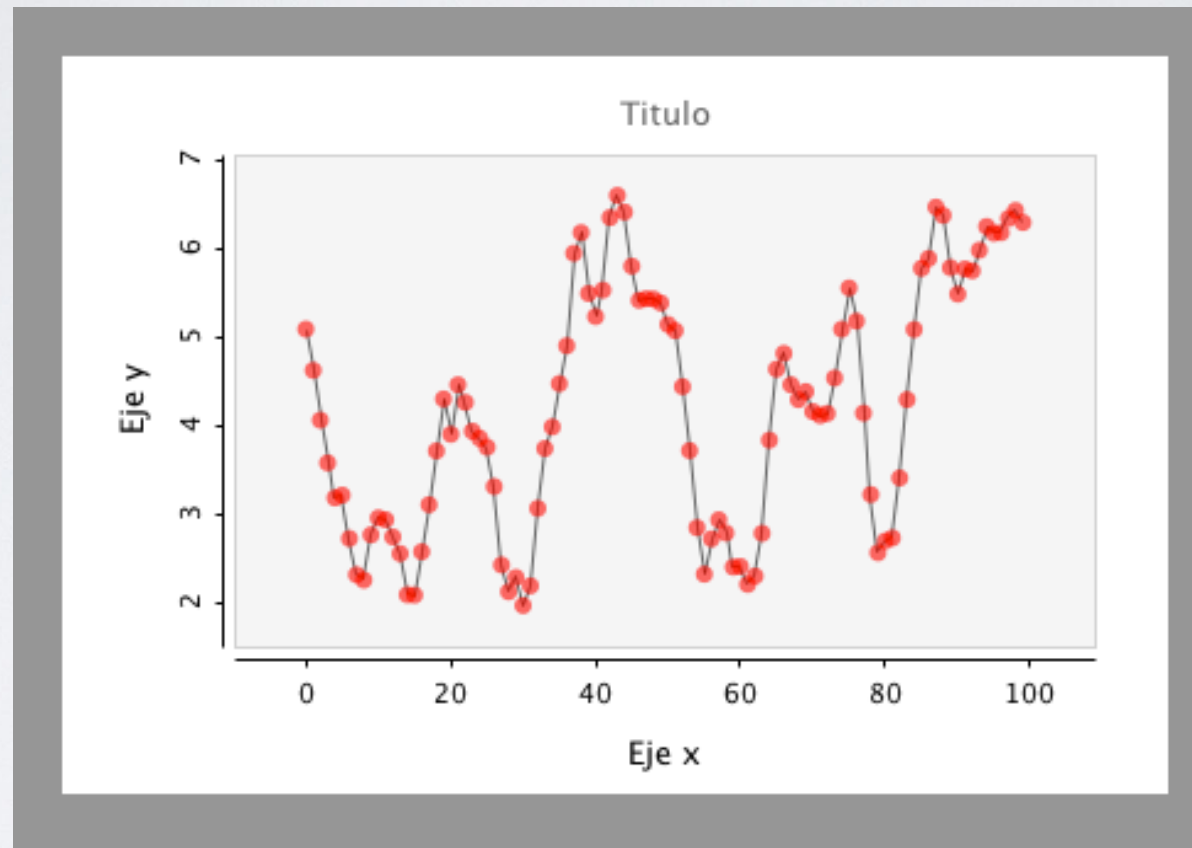
```
import de.bezier.data.*;  
XlsReader reader;  
reader = new XlsReader(this, "puntos.xls" );
```

```
reader.firstRow();  
reader.hasMoreRows();  
reader.nextRow();  
reader.nextCell();
```

```
reader.getFloat();  
reader.getInt();  
reader.getString();
```

- La librería de XlsReader permite abrir, navegar y leer un archivo .xls (Excel de 97 a 2009)
- No es posible escribir dentro del archivo.
- Esta librería carece de referencias.

GRATIFICADOR PARA EXCEL



COMUNICACIÓN SERIAL

```
import processing.serial.*;
```

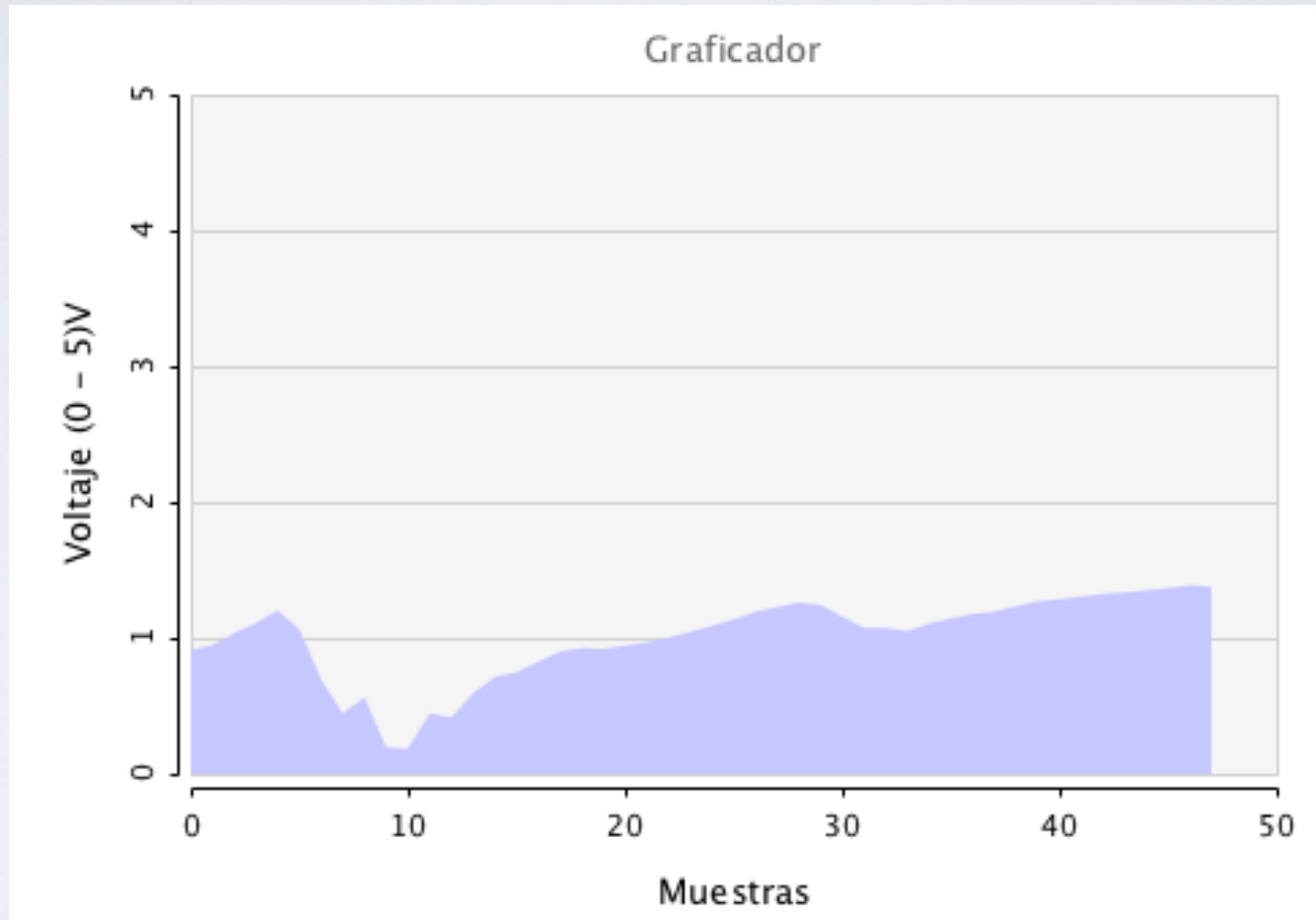
```
Serial myPort;  
println(Serial.list());  
myPort = new Serial(this, Serial.list()[0], 9600);
```

```
void serialEvent(Serial p) {  
    inString = p.readString();  
}
```

- La librería de Serial lee y escribe datos un byte desde y hacia dispositivos externos.
- Esta librería permite la comunicación con dispositivos externos como microcontroladores y utilizarlos como entradas o salidas.
- El puerto serial es un puerto de nueve pines I / O que existe en muchos PCs y puede ser emulado a través de USB.

Nombre	Acción
<code>int val = myPort.available()</code>	Regresa el numero de bytes que esperan en el buffer.
<code>int val = myPort.read();</code>	Regresa el primer byte en la cola del biffer.
<code>String cadena = myPort.readString();</code>	Regresa todo el contenido del buffer en un string.
<code>println(Serial.list());</code>	Regresa una lista de todos los puertos disponibles.
<code>myPort.clear();</code>	Elimina todo el contenido del buffer.
<code>myPort.write(int/char/byte);</code>	Envia un byte.
<code>myPort.bufferUntil(valor);</code>	Establece un byte específico al almacenar en el buffer hasta llamar al serialEvent ().
<code>string myString = myPort.readStringUntil(valor);</code>	Le una cadena hasta encontrar un byte específico.
<code>serialEvent()</code>	Funcion que se activa cada vez que llega un nuevo dato al buffer.

GRATIFICADOR DE SENSOR



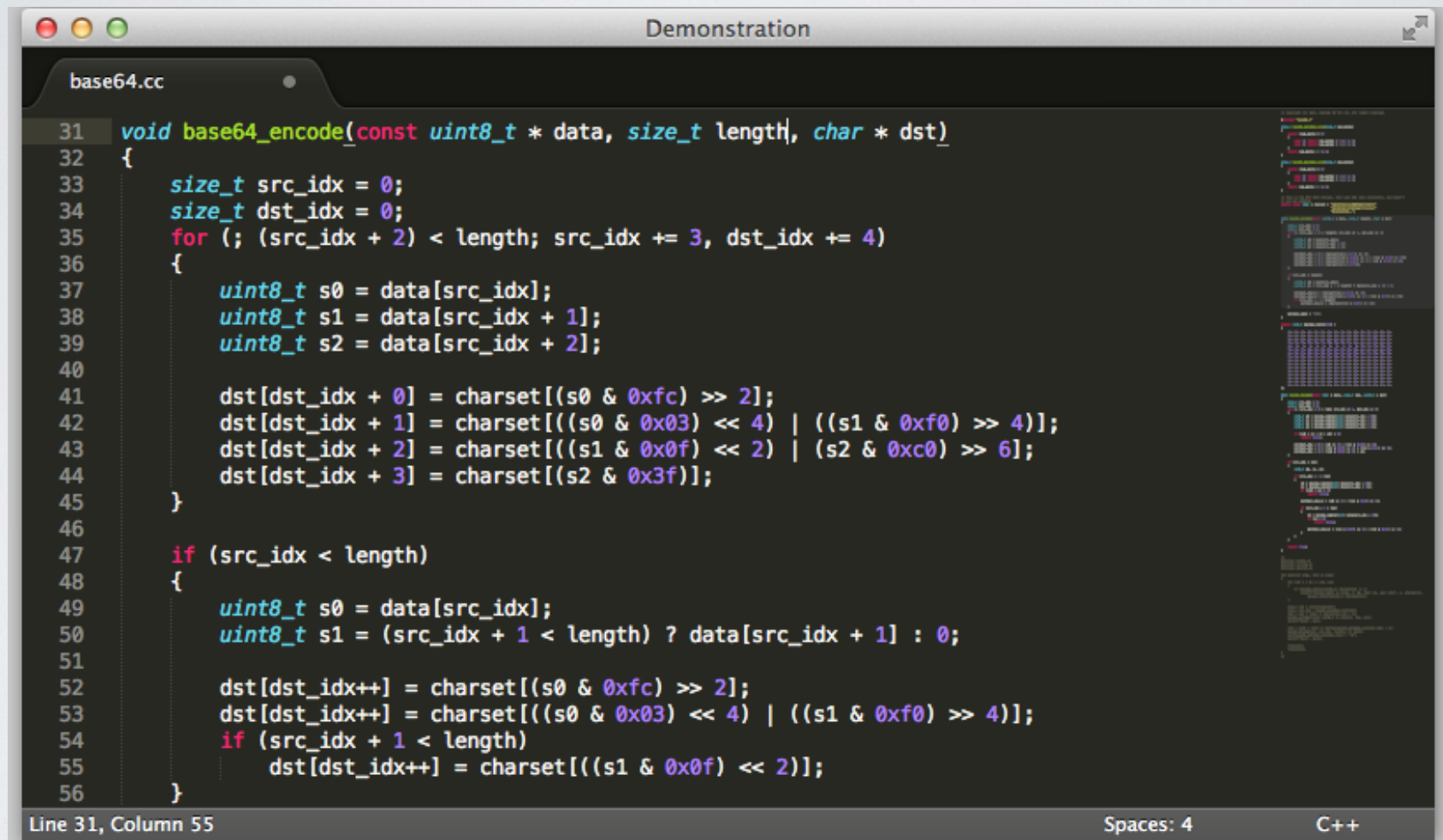
SUBLIME TEXT

Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia.

Se distribuye de forma gratuita, sin embargo no es software libre o de código abierto, se puede obtener una licencia para su uso ilimitado, pero el no disponer de ésta no genera ninguna limitación más allá de una alerta cada cierto tiempo. Sitio oficial : <http://www.sublimetext.com>.

- **Minimapa**
- **Multi Selección**
- **Multi Cursor**
- **Multi Layout**
- **Soporte nativo para infinidad de lenguajes**
- **Búsqueda Dinámica**
- **Auto completado y marcado de llaves**
- **Soporte de Snippets y Plugins**
- **Configuración total de Keybindings**
- **Acceso rápido a línea o archivo (Cmd+P o Ctrl+P)**
- **Coloreado y envoltura de sintaxis**
- **Pestañas**
- **Resaltado de paréntesis e indentación.**

SUBLIME TEXT



```
31 void base64_encode(const uint8_t * data, size_t length, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < length; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < length)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < length) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < length)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
```

Line 31, Column 55

Spaces: 4

C++

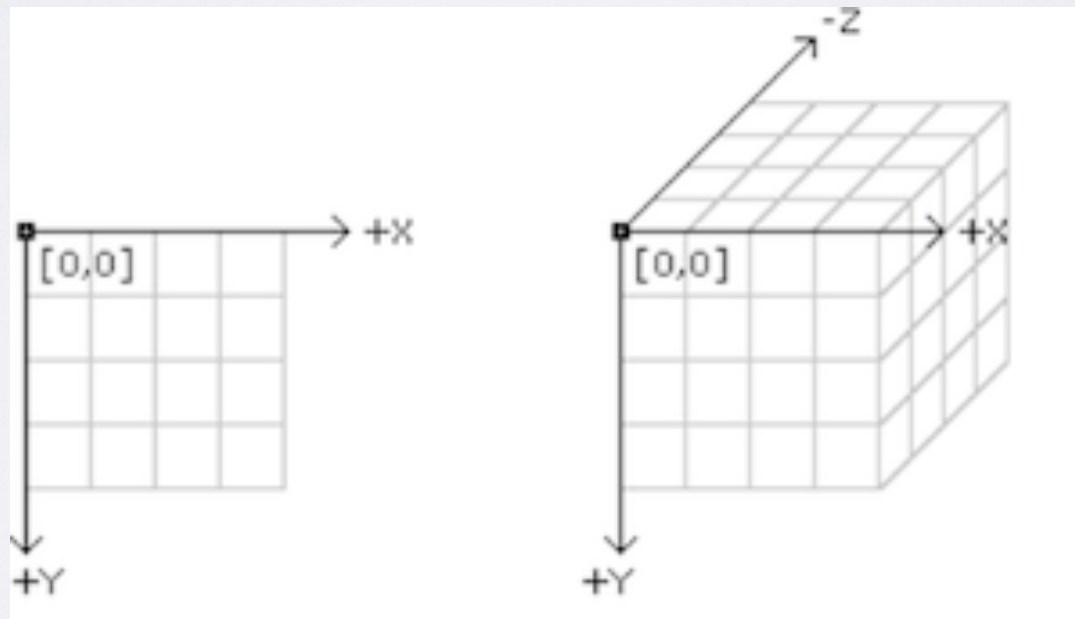
Comando (⌘/Ctrl)	Acción
Ctrl + L Ctrl + ⌘ + L Ctrl + ⌘ + K Ctrl + KK (igual en Mac) Ctrl + K + ⌘ (igual en Mac) Ctrl + ⌘ + D	Manejo de lienas
Ctrl + X	Corta la linea completa
Ctrl + ↵ Ctrl + ⌘ + ↵	Añadir lineas
Ctrl + ↑ Ctrl + ↓ Ctrl + ⌘ + ↑ Ctrl + ⌘ + ↓	Selección completa
Ctrl + D Ctrl + Click	Multicursor con selección
Ctrl + M Ctrl + ⌘ + M (igual en Mac)	Manejode contenido con corchetes

Comando (⌘/Ctrl)	Acción
Ctrl + P Ctrl + ⇧ + P	Abrir archivos y panel de control
Ctrl + R	Buscador de funciones
Ctrl + F2 F2 ⇧ + F2 Ctrl + ⇧ + F2	Bookmarks
Ctrl + KU Ctrl + KL	Mayúsculas/Minúsculas
Ctrl + / Ctrl + ⇧ + /	Comentarios
Ctrl + Space	Navegación en auto-compleat

PROCESSING 3D

En Procesamiento de 2.0, hay 3 modos de interpretación: P2D, P3D y OPENGLE. Esto debe de ser especificado en la función `size()`.

```
void setup() {  
  size(200,200,P3D);  
}
```



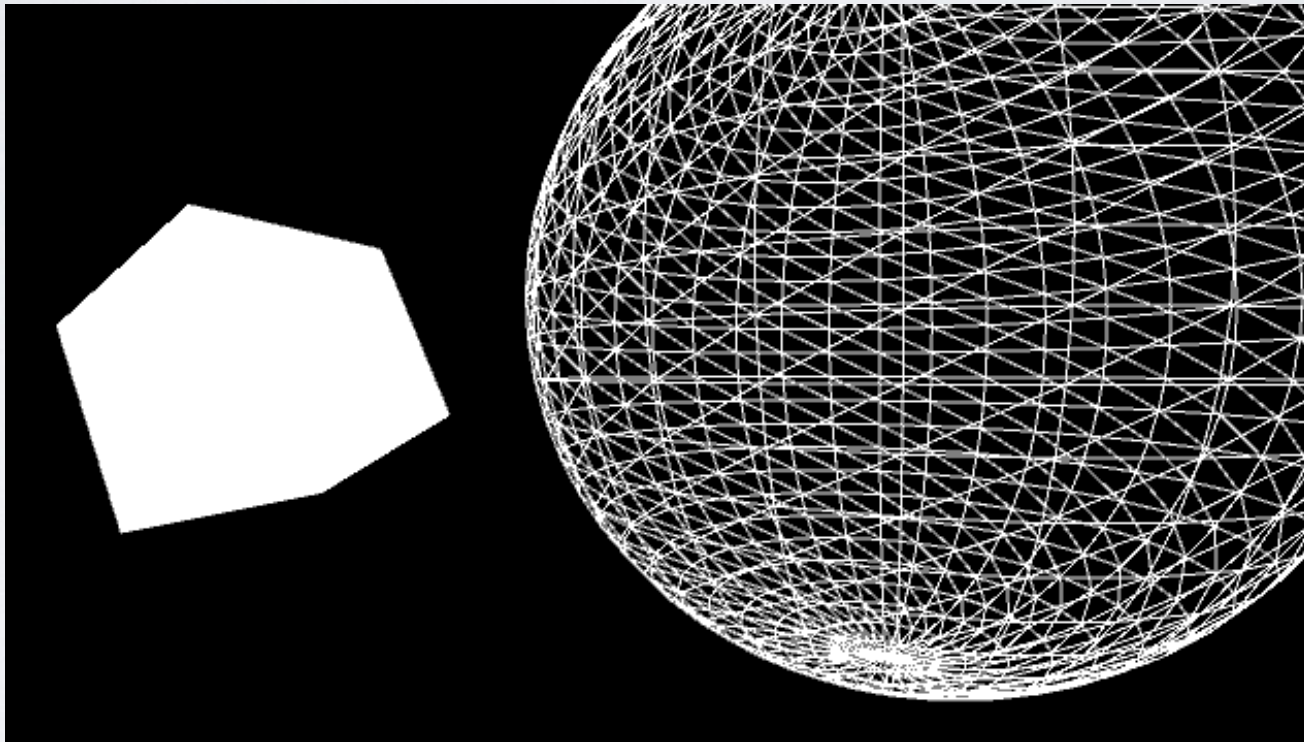
TRANSFORMACIONES

`translate(x, y)`
`translate(x, y, z)`

`rotateX(angle)`
`rotateY(angle)`
`rotateZ(angle)`

- El parámetro X especifica izquierda o derecha, el parámetro Y especifica hasta arriba o abajo, y el parámetro Z especifica traducciones adelante o atrás.
- Utilizando esta función con el parámetro z requiere el uso de P3D.
- Las transformaciones son acumulativas.
- Esta función se puede controlar aún más mediante el uso de `pushMatrix()` y `popMatrix()`.

P3D



PEASYCAM

```
import peasy.*;
```

```
PeasyCam cam;  
cam = new PeasyCam(this, 1200);
```

- Es una librería escrita por de Jonathan Feinberg.
- PeasyCam ofrece manejo de cámara de forma muy simple.
- Se basa en el excelente paquete geometría Apache Commons Matemáticas para sus rotaciones.
- El PeasyCam se coloca sobre una esfera cuyo radio es la distancia dada de la ventana y el centro del objeto.
- Mas información en: <http://mrfeinberg.com/peasycam/>

Creado por: Miguel Angel Ruiz Gálvez
Contacto: miguelo.me

Agradecimientos a:

- Ben Fry
- Casey Reas
- Javier Graciá Carpio
- Jonathan Feinberg

Esta obra está licenciada bajo la Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita <http://creativecommons.org/licenses/by-nc-sa/4.0/>.