



PPEM 2022

Creación de GUI

Creación de GUI



Librerías para crear GUI

Controladores

Agrupaciones de controladores

Visualización de datos en la GUI

Cómo guardar valores y controlar controladores

Librerías

Libraries

Modes

Tools

Examples

Updates



All



Status

Name

Author



ControlP5 | A GUI library to build custom user interfaces f...

Andreas Schlegel



Dawesome Toolkit | Convenient utilities for some everyd...

Brendan Dawes



G4P | Provides a set of 2D GUI controls and multiple wind...

Peter Lager



Guido | A simple cross mode GUI library

Florian Jenett



Interfascia | Build simple yet gorgeous user interfaces.

Brendan Berg



SketchMapper | A GUI tool to map sketches onto surfaces.

J. Taylor OConnor

Librerías

The screenshot shows a web interface for browsing libraries. At the top, there are navigation tabs: 'Libraries' (selected), 'Modes', 'Tools', 'Examples', and 'Updates'. Below the tabs is a search bar containing the text 'gui' and a dropdown menu set to 'All'. The main content area displays a list of search results with columns for 'Status', 'Name', and 'Author'. Each result includes a green checkmark icon, the library name, a brief description, and the author's name.

| Status | Name ▲ | Author |
|--------|--|--------------------------------|
| | ComposingForEveryone ComposingForEveryone gives support f... | Guido Kramann |
| ✓ | ControlP5 A GUI library to build custom user interfaces for desk... | Andreas Schlegel |
| ✓ | Dawesome Toolkit Convenient utilities for some everyday tasks i... | Brendan Dawes |
| ✓ | G4P Provides a set of 2D GUI controls and multiple window supp... | Peter Lager |
| ✓ | Guido A simple cross mode GUI library | Florian Jenett |
| ✓ | Interfascia Build simple yet gorgeous user interfaces. | Brendan Berg |
| ✓ | SketchMapper A GUI tool to map sketches onto surfaces. | J. Taylor OConnor |
| ✓ | UiBooster Creates fast and easy GUI components for your sketch... | Nick 'Milchreis' Müller |

Librerías

ControlP5 - buena documentación, muchas clases de controladores.

Dawsome Toolkit - colocar cosas alrededor de una esfera, hacer cuadrículas, diseños en espiral de vogel, loguear a un archivo, generar paletas de colores simples,... - NO sirve para GUI

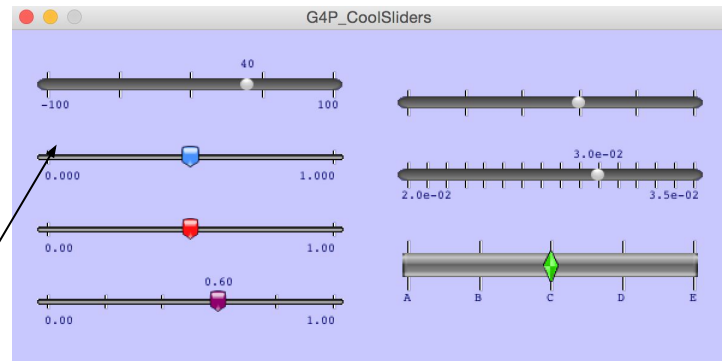
G4P - CoolSlider poco cool + “G4P is a large complex library, and sometimes there are issues with Processing that affect it adversely.”

Guido - pocos ejemplos, sin documentación. Botón, checkbox, slider.

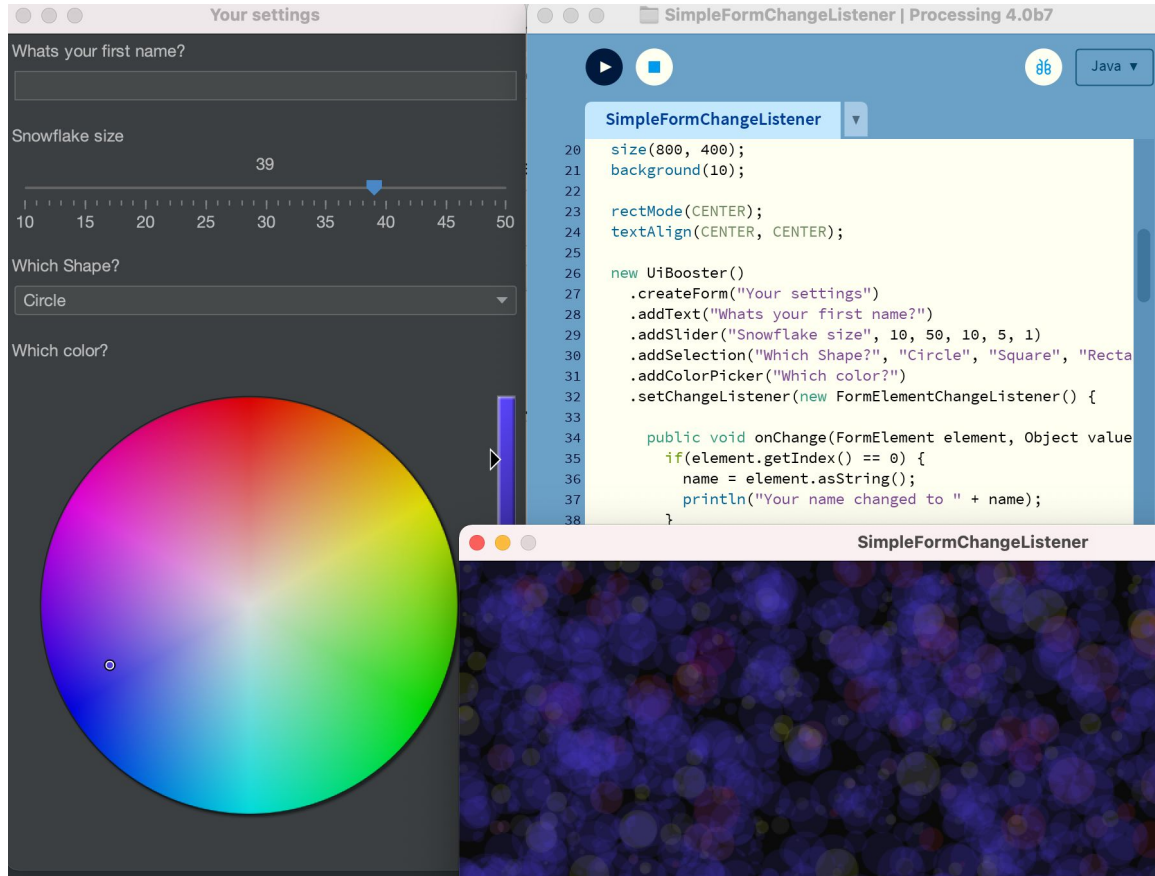
Interfascia - poca cosa: botón, radio botón, checkbox, campo de texto.

SketchMapper - para mapping.

UiBooster -notificaciones tipo pop-up, slider, elegir opción, elegir color, panel externo



UiBooster



The image displays a Processing 4.0b7 IDE window titled "SimpleFormChangeListener | Processing 4.0b7". The code in the editor is as follows:

```
20 size(800, 400);
21 background(10);
22
23 rectMode(CENTER);
24 textAlign(CENTER, CENTER);
25
26 new UiBooster()
27   .createForm("Your settings")
28   .addText("Whats your first name?")
29   .addSlider("Snowflake size", 10, 50, 10, 5, 1)
30   .addSelection("Which Shape?", "Circle", "Square", "Recta")
31   .addColorPicker("Which color?")
32   .setChangeListener(new FormElementChangeListener() {
33
34     public void onChange(FormElement element, Object value)
35     {
36       if(element.getIndex() == 0) {
37         name = element.asString();
38         println("Your name changed to " + name);
39       }
40     }
41   });
```

The IDE shows a preview of the application. On the left, a window titled "Your settings" contains a text input field for "Whats your first name?", a slider for "Snowflake size" set to 39, a dropdown menu for "Which Shape?" set to "Circle", and a color picker for "Which color?". On the right, a window titled "SimpleFormChangeListener" displays a dark blue background with a field of small, semi-transparent blue and purple snowflakes.

ControlP5



Creamos un botón

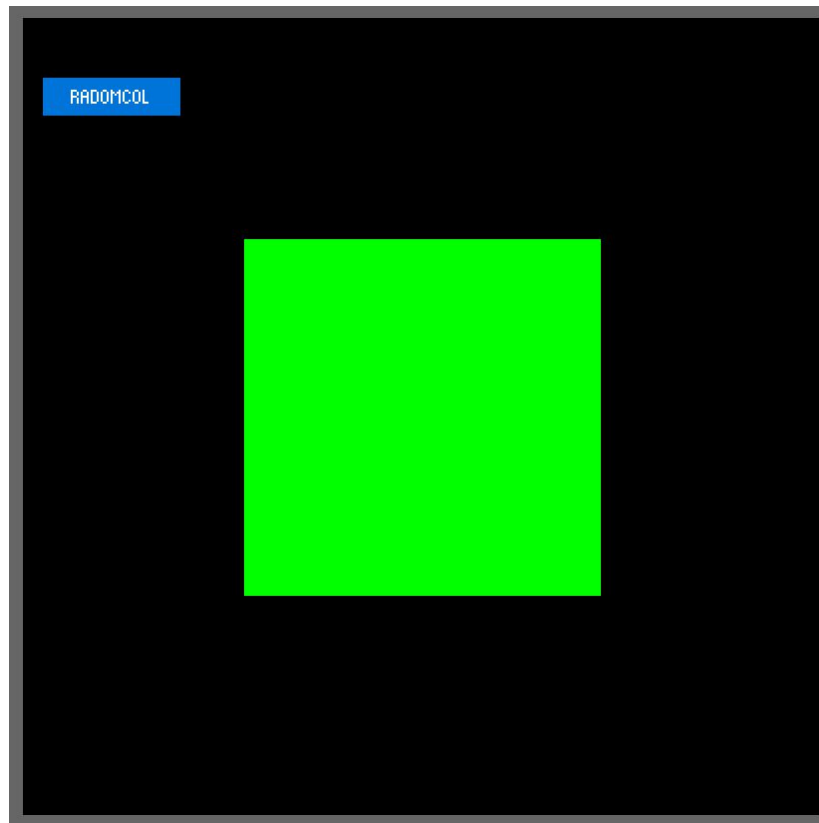
```
import controlP5.*;
ControlP5 cp5;
float randomFill = 0;
```

```
void setup() {
  size(400,400);
  colorMode(HSB);
  cp5 = new ControlP5(this);
  cp5.addButton("randomCol");
}
```

```
void draw() {
  background(0);
  fill(randomFill,255,255);
  rect(110,110,180,180);
}
```

```
void randomCol() {
  randomFill = random(255);
}
```

Se posiciona sólo!



Creamos un slider

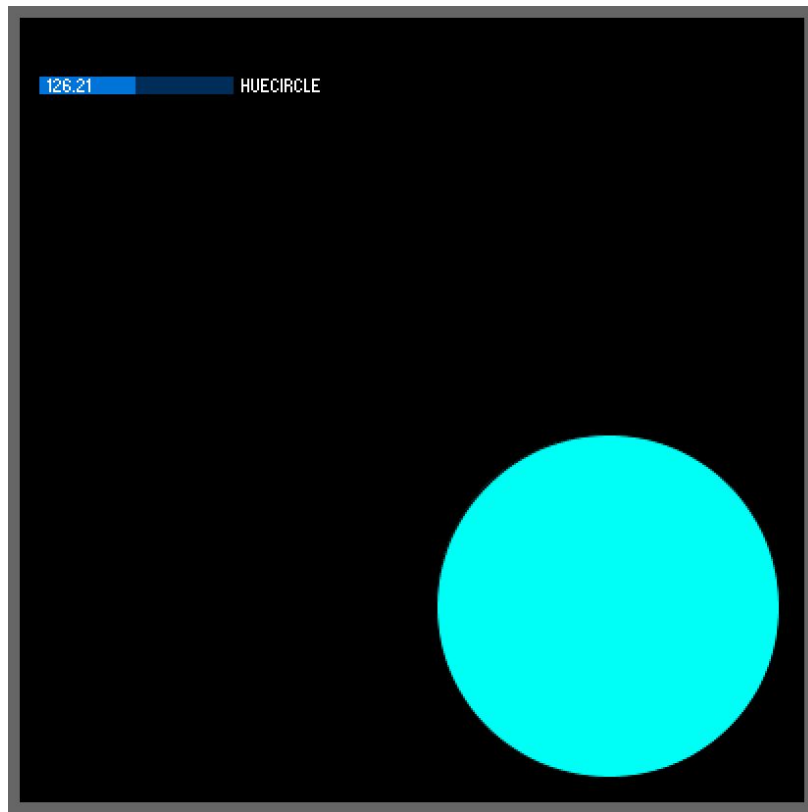
```
import controlP5.*;
ControlP5 cp5;
float hueCircle = 0;

void setup() {
  size(400,400);
  colorMode(HSB);
  cp5 = new ControlP5(this);
  cp5.addSlider("hueCircle",0,255);
}

void draw() {
  background(0);

  fill(hueCircle,255,255);
  ellipse(300,300,175,175);
}
```

Se posiciona sólo!



Rápido posicionamiento con linebreak

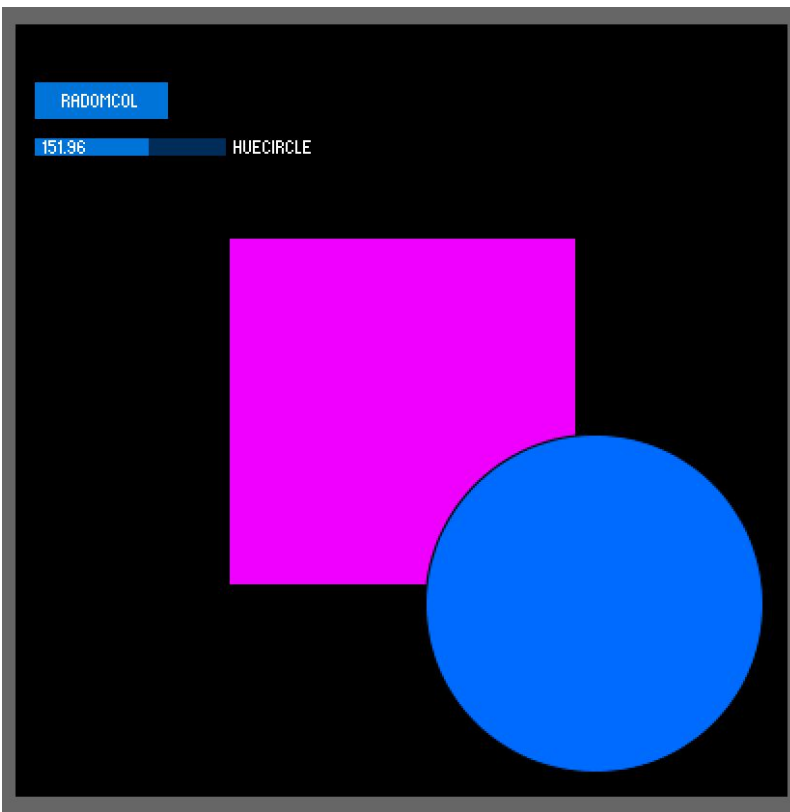
```
import controlP5.*;
ControlP5 cp5;
float randomFill = 0;
float hueCircle = 0;

void setup() {
  size(400,400);
  colorMode(HSB);
  cp5 = new ControlP5(this);
  cp5.addButton("randomCol").linebreak();
  cp5.addSlider("hueCircle",0,255);
}

void draw() {
  background(0);
  fill(randomFill,255,255);
  rect(110,110,180,180);

  fill(hueCircle,255,255);
  ellipse(300,300,175,175);
}

void randomCol() {
  randomFill = random(255);
}
```

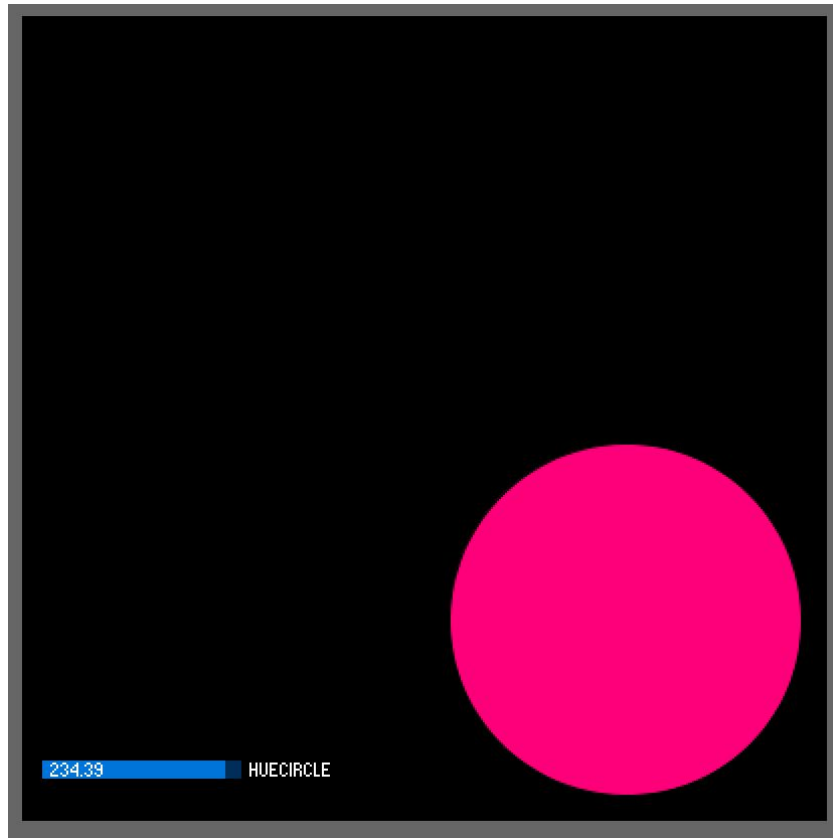


Posicionamiento pasando la posición

```
import controlP5.*;
ControlP5 cp5;
float hueCircle = 0;

void setup() {
  size(400,400);
  colorMode(HSB);
  cp5 = new ControlP5(this);
  cp5.addSlider("hueCircle",0,255)
    .setPosition(10,370);
}

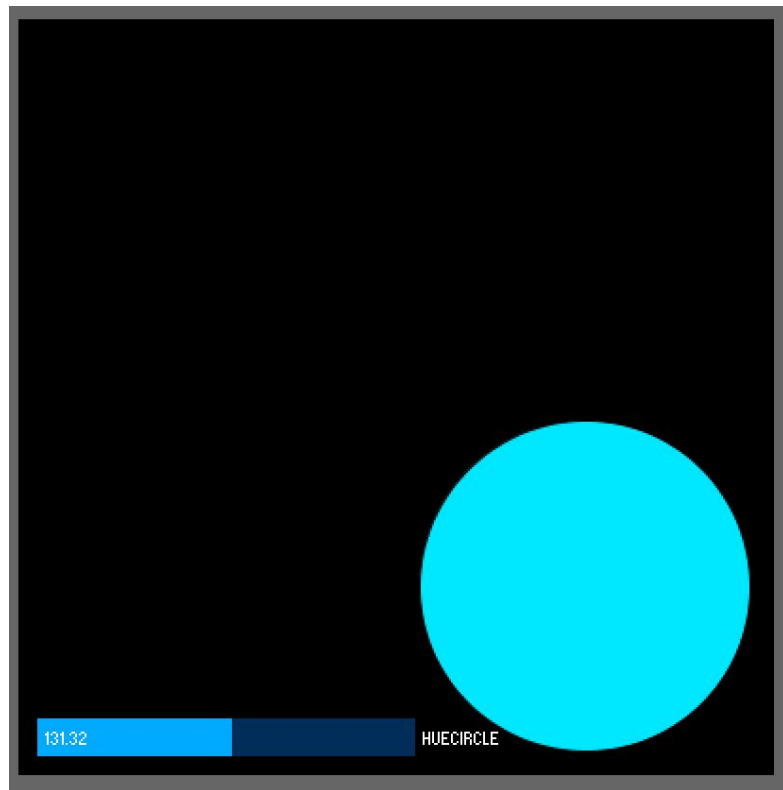
void draw() {
  background(0);
  fill(hueCircle,255,255);
  ellipse(300,300,175,175);
}
```



El tamaño del controlador

```
import controlP5.*;
ControlP5 cp5;
float hueCircle = 0;

void setup() {
  size(400,400);
  colorMode(HSB);
  cp5 = new ControlP5(this);
  cp5.addSlider("hueCircle",0,255)
    .setPosition(10,370)
    .setSize(200,20);
}
void draw() {
  background(0);
  fill(hueCircle,255,255);
  ellipse(300,300,175,175);
}
```



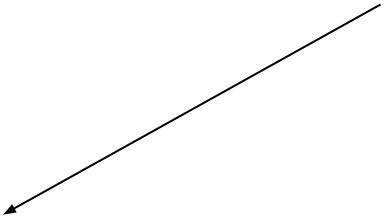
Cómo acceder a los valores de los controladores?

- Una función asociada al controlador (como randomCol)
- Una variable asociada al controlador (como hueCircle)
- getValue
- plugTo (conecto con una variable)
- Evento de control
 - theEvent.isController() + theEvent.getController().getName()
 - theEvent.isController() + theEvent.getController().getId()
 - Definición de un controllerX + theEvent.isFrom(controllerX)

getValue

```
import controlP5.*;
ControlP5 cp5;
float hueCircle = 0;

void setup() {
  size(400,400);
  colorMode(HSB);
  cp5 = new ControlP5(this);
  cp5.addSlider("hueCircle",0,255)
    .setPosition(10,370)
    .setSize(200,20);
}
void draw() {
  background(0);
  fill(cp5.getController("hueCircle").getValue(),255,255);
  ellipse(300,300,175,175);
}
```



plugTo

```
import controlP5.*;
ControlP5 cp5;
float hueCircle = 0;

void setup() {
  size(400,400);
  colorMode(HSB);
  cp5 = new ControlP5(this);
  cp5.addSlider("hueCircleController",0,255)
    .plugTo( this,"hueCircle")
    .setPosition(10,370)
    .setSize(200,20);
}
void draw() {
  background(0);
  fill(hueCircle,255,255);
  ellipse(300,300,175,175);
}
```

Evento de control: getController+getName

```
import controlP5.*;
ControlP5 cp5;
float randomFill = 0;
float hueCircle = 0;

void setup() {
  size(400,400);
  colorMode(HSB);
  cp5 = new ControlP5(this);
  cp5.addButton("randomCol").linebreak();
  cp5.addSlider("hueCircleController",0,255);
}

void draw() {
  background(0);
  fill(randomFill,255,255);
  rect(110,110,180,180);
  fill(hueCircle,255,255);
  ellipse(300,300,175,175);
}
```

```
void controlEvent(ControlEvent theEvent) {
  if(theEvent.isController()) {
    if(theEvent.getController().getName()=="randomCol"){
      randomFill = random(255);
    } else
    if(theEvent.getController().getName()=="hueCircleController")
    {
      hueCircle = theEvent.getController().getValue();
    }
  }
}
```


Evento de control: getController+getId

```
import controlP5.*;
ControlP5 cp5;
float randomFill = 0;
float hueCircle = 0;

void setup() {
  size(400,400);
  colorMode(HSB);
  cp5 = new ControlP5(this);
  cp5.addButton("randomCol")
    .setId(0).linebreak();
  cp5.addSlider("hueCircleController",0,255)
    .setId(1);
}

void draw() {
  background(0);
  fill(randomFill,255,255);
  rect(110,110,180,180);
  fill(hueCircle,255,255);
  ellipse(300,300,175,175);
}
```

```
void controlEvent(ControlEvent theEvent) {
  if(theEvent.isController()) {
    if(theEvent.getController().getId()==0){
      randomFill = random(255);
    } else if(theEvent.getController().getId()==1){
      hueCircle = theEvent.getController().getValue();
    }
  }
}
```

Evento de control y isFrom

```
import controlP5.*;
ControlP5 cp5;
Button randomBtn;
Slider hueSlider;
float randomFill = 0;
float hueCircle = 0;
```

```
void setup() {
  size(400,400);
  colorMode(HSB);
  cp5 = new ControlP5(this);
  randomBtn = cp5.addButton("randomCol").linebreak();
  hueSlider = cp5.addSlider("hueCircleController",0,255);
}

void draw() {
  background(0);
  fill(randomFill,255,255);
  rect(110,110,180,180);
  fill(hueCircle,255,255);
  ellipse(300,300,175,175);
}
```

```
void controlEvent(ControlEvent theEvent) {
  if(theEvent.isFrom(randomBtn)){
    randomFill = random(255);
  }else if(theEvent.isFrom(hueSlider)){
    hueCircle = theEvent.getController().getValue();
  }
}
```

Controladores



Button: se activa al soltar. Se puede asociar con un valor mediante setValue()

Bang: se activa al apretar

Toggle: tiene dos estados: true o false

Matrix: una matriz de botones de tipo toggle con un recorrido en el tiempo

Slider: permite recorrer un rango de valores. Puede ser: Slider, Slider2d o Knob

ColorPicker y **ColorWheel:** permiten definir un color o seleccionar un color

Numberbox: para cambiar el número con arrastrando el mouse

Range: permite definir un mínimo y máximo en un rango de valores

Textfield: para ingresar texto

Agrupaciones



Checkbox: array de toggles con 2 estados: 0 o 1. Cada toggle puede tener asociado un valor interno

ButtonBar: botonera donde uno puede responder a clicks en cada botón (se devuelve el índice del botón)

RadioButton: devuelve estado de todos los items (`theEvent.getGroup().getArrayValue()[i]`) o el valor del botón activo (`theEvent.getValue()`)

Group: para agrupar varios controladores. Se puede colapsar

Accordion: para agrupar varios Group

ScrollableList: listas con barra de scroll

Tab: permite generar pestañas para agrupar funcionalidades

***** ControlFrame:** para generar una ventana aparte usando PApplet (ejemplo: `extra/ControlP5frame`)

Visualización de datos



Chart: para visualizar series de datos

Textarea: para visualizar texto (un uso interesante: desviar los prints en consola: `extra/controlp5console`)

TextLabel: para escribir texto (...?)

Guardar los valores



Guardar y cargar los valores de los controladores (ejemplo `use/ControlP5propertiesCustom`) en un json en la carpeta del sketch. Disponible para:

Slider, Knob, Numberbox, Toggle, Checkbox, RadioButton, Textlabel, Matrix, Range, Textarea, ColorPicker.

Control de los controladores



Se puede modificar los valores que tienen los controladores con:

- Directamente usando las funciones “set” // ejemplo:
use/ControlP5sliderSetValue
- Mapeando las teclas a las acciones de los controladores en el setup con la función “mapKeyFor” // ejemplo:
controllers/ControlP5accordion

Texturas con controladores

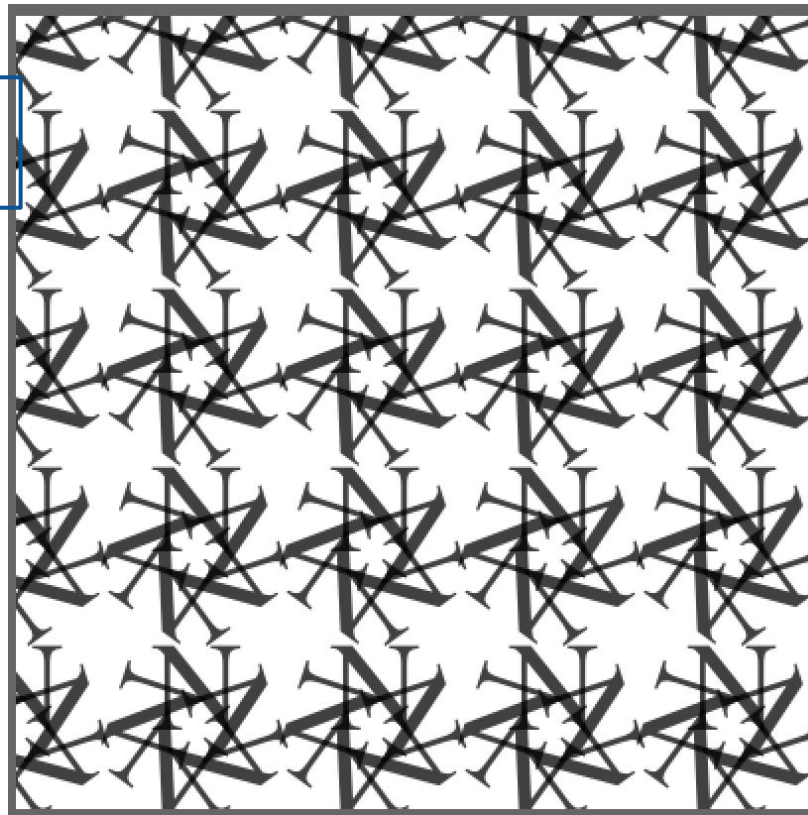
+ En ControlWindow

```
PFont f = createFont("Serif",64,true);
int repeatNr = 5;
textFont(f);
size(400,400);
background(255);
fill(0,190);
textAlign(CENTER);
for(int i=0;i<width;i+=90){
  for(int j=0;j<height;j+=90){
    for(int r=0;r<repeatNr;r++){
      pushMatrix();
      translate(i,j);
      rotate(radians(360/repeatNr)*r);
      text("N",0,0);
      popMatrix();
    }
  }
}
```

radioButton con
opciones 1,2,3,4,5,6

slider2D

Campo para ingresar texto



Agregar controladores al texto

+ En ControlWindow

```
String message = "hola!mundo!";  
float rotationDegrees;  
float theta;  
void setup(){  
  size(400,400);  
  textSize(32);  
  rotationDegrees = 360/message.length();  
  theta = 0;  
  background(0);  
  textAlign(CENTER);  
}  
void draw(){  
  fill(0,20);  
  rect(0,0,width,height);  
  fill(255);  
  translate(width/2,height/2); // nos movemos al centro  
  rotate(theta); // para que haya un giro constante  
  for(int i =0;i<message.length();i++){  
    pushMatrix();  
    rotate(radians(rotationDegrees*i)); // rotar el sistema de coordenadas  
    translate(100,0); // nos alejamos del centro  
    rotate(radians(90)); // para rotar la letra  
    text(message.charAt(i),0,0); // dibujo la letra en 0,0  
    popMatrix();  
  }  
  theta+=0.01;  
}
```

Campo para ingresar texto

Un slider de 0 a 100

Knob de 0 a 360

Numberbox de 0 a 1



Entrega 3



Texturas tridimensionales compuestas con letras, objetos tridimensionales con imágenes que responden al sonido con una GUI para controlar parámetros de la visualización.

Tiene que incluir:

1. **Texto o letras** como elementos decorativos en el espacio tridimensional + repetición
2. Por lo menos 2 objetos **tridimensionales distintos** (box, shpera, PShape o formas compuestas por vértices)
3. Por lo menos un objeto con una **textura** hecha con una imagen
4. Cambio de **color** de algún elemento mediante **controlador**
5. Cambio de alguna **letra o texto** mediante **controlador** (usando **Textfield**)
6. Cambio de **color** de algún elemento mediante **sonido**
7. Cambio de **ubicación o rotación** de algún elemento mediante **sonido**
8. Una GUI con por lo menos **4 controladores distintos**. Puede haber más de 4 controladores si lo desean.

Fecha de entrega: miércoles **20.04.2022** hasta las 23:59.