

# Interactivity and GUI in Processing



Technische Universiteit  
**Eindhoven**  
University of Technology

Where innovation starts

# Content

- User input in Processing
  - Mouse
  - Keyboard
- controlP5, a GUI library
  - Bang
  - Button
  - Toggle
  - Radio buttons
  - Numberbox
  - Slider
  - Knob
  - Textfield
  - Textlabel

# User input in Processing

## Mouse

- System variables: mouseX and mouseY
  - contain the current coordinate of the mouse.
  - Load example: Mouse\mouseXmouseY:

```
void draw() {  
    background(128);  
    line(0, 0, mouseX, mouseY);  
}
```

# User input in Processing

## Mouse

- System variables: pmouseX and pmouseY
  - contain the coordinate of the mouse in the frame previous to the current one.
  - Load example: Mouse\pmouseXpmouseY:

```
void draw() {  
    background(204);  
    line(mouseX, 20, pmouseX, 80);  
}
```

# User input in Processing

## Mouse

- System variable: mousePressed
  - true if a mouse button is pressed; false if a button is not pressed
  - Load example: Mouse\mousePressedVariable :

```
void draw() {  
    background(204);  
    if (mousePressed == true) {  
        fill(0);  
    } else {  
        fill(255);  
    }  
    rect(mouseX-10, mouseY-10, 20, 20);  
}
```

# User input in Processing

## Mouse

- System variable: mouseButton
  - either **LEFT**, **RIGHT**, or **CENTER** depending on which button is pressed.
  - Load example: Mouse\mouseButton :

```
void draw() {  
    background(204);  
    if (mousePressed == true && mouseButton == LEFT) {  
        fill(0,255,0);  
    }  
    else if (mousePressed == true && mouseButton == RIGHT) {  
        fill(255,0,0);  
    }  
    else {  
        fill(255,255,255);  
    }  
    rect(mouseX-10, mouseY-10, 20, 20);  
}
```

# User input in Processing

## Mouse

- Event: `mousePressed()`
  - called once after every time a mouse button is pressed.
  - Load example: `Mouse\mousePressedEvent` :

```
void draw() {  
    background(204);  
    rect(mouseX-10, mouseY-10, 20, 20);  
}  
  
void mousePressed() {  
    if (mouseButton == LEFT) {  
        fill(0,255,0);  
    }  
    else if (mouseButton == RIGHT) {  
        fill(255,0,0);  
    }  
    else {  
        fill(255,255,255);  
    }  
}
```

Should have the same behavior as previous one.

Something is different.

# User input in Processing

## Mouse

- Event: `mouseReleased()`
  - called every time a mouse button is released.
  - Load example: `Mouse\mouseReleased` :

```
void draw() {  
    background(204);  
    rect(mouseX-10, mouseY-10, 20, 20);  
}
```

```
void mousePressed() {  
    if (mouseButton == LEFT) {  
        fill(0,255,0);  
    }  
    else if (mouseButton == RIGHT) {  
        fill(255,0,0);  
    }  
}
```

```
void mouseReleased() {  
    fill(255,255,255);  
}
```

# User input in Processing

## Mouse

- Event: mouseMoved
  - called every time the mouse moves and a mouse button is not pressed.
  - Load example: Mouse\mouseMoved :

```
int value = 0;

void draw() {
    fill(value);
    rect(25, 25, 50, 50);
}

void mouseMoved() {
    value = value + 5;
    if (value > 255) {
        value = 0;
    }
}
```

# User input in Processing

## Mouse

- Event: mouseDragged
  - called once every time the mouse moves and a mouse button is pressed.
  - Load example: Mouse\mouseDragged :

```
int value = 0;

void draw() {
    fill(value);
    rect(25, 25, 50, 50);
}

void mouseDragged()
{
    value = value + 5;
    if (value > 255) {
        value = 0;
    }
}
```

# User input in Processing

## Keyboard

System Variables		
key	keyCode: if(Key == CODED)	keyPressed
„A“, „B“, ... „Z“, „a“, „b“, ... „z“, „0“, „1“, ... „9“, „~“, „!“, „@“, ... „?“ etc.	BACKSPACE, TAB, ENTER, RETURN, ESC, DELETE, UP, DOWN, LEFT, RIGHT, ALT, CONTROL, SHIFT.	true false

Events
keyPressed()
keyReleased()

# User input in Processing

## Keyboard

- Load example: Keyboard\keyPressedVariable

```
void draw() {  
    if(keyPressed) {  
        if (key == 'b' || key == 'B') {  
            fill(0);  
        }  
    }  
    else {  
        fill(255);  
    }  
    rect(25, 25, 50, 50);  
}
```

# User input in Processing

## Keyboard

- Load example: Keyboard\keyPressedEvent

```
int fillVal = 126;

void draw() {
    fill(fillVal);
    rect(25, 25, 50, 50);
}

void keyPressed() {
    if (key == CODED) {
        if (keyCode == UP) {
            fillVal = fillVal<255 ? fillVal+5 : 255;
        }
        else if (keyCode == DOWN) {
            fillVal = fillVal>0 ? fillVal-5 : 0;
        }
    }
    else {
        fillVal = 126;
    }
}
```

# GUI in Processing Libraries

- See <http://processing.org/reference/libraries/index.html#interface>
  - controlP5 : “highly recommended”.
  - Interfascia: Not really completed.
  - MyGUI: Poor documentation
  - SpringGUI: Based on Java AWT. Problems in the “stage”.

# GUI in Processing Libraries

- See <http://processing.org/reference/libraries/index.html#interface>
  - controlP5 : “highly recommended”.
  - Interfascia: Not really completed.
  - MyGUI: Poor documentation
  - SpringGUI: Based on Java AWT. Problems in the “stage”.

Notice: if using Processing 1.5.1, you have to download an earlier version of ControlP5: controlP5-1.5.2.zip

# controlP5 Framework

```
//first import the controlP5 library
import controlP5.*;

//Define an ControlP5 variable.
ControlP5 controlP5;

void setup() {
    //create a top level control manager
    controlP5 = new ControlP5(this);

    //add GUI components to the manager
    controlP5.add<Component>("nameOfTheComponent", param1, param2, ... paramn);

    //and if necessary, change the default properties of the component
    controlP5.controller("nameofTheComponent").setLabel("This is the new label");
}

void draw() {
    //draw as usual
}

// callback when an action is performed with the component "nameOfTheComponent"
void nameOfTheComponent([<Type> <value>]) {
    //do something...
}
```

nameOfTheComponent:

- identifies the component,
- is the default label
- defines the name of the callback function

# controlP5

## Bang

- addBang(theName, theX, theY, theWidth, theHeight);
- Load example: controlP5\firstBang

```
import controlP5.*;
ControlP5 controlP5;
float x=200, y = 200;

void setup() {
    size(400, 400);

    controlP5 = new ControlP5(this);
    controlP5.addBang("firstBang", 10, 10, 40, 20 );
}

void draw() {
    background(0);
    fill(255);
    ellipse(x,y, 40, 40);
}

void firstBang() {
    x=random(400); y=random(400);
}
```

# controlP5

## Bang

- Load example: controlP5\secondBang

```
import controlP5.*;  
  
ControlP5 controlP5;  
color c = color(0,0,0);  
void setup() {  
    size(400, 400);  
    controlP5 = new ControlP5(this);  
    controlP5.addBang("firstBang", 10, 10, 40, 20 );  
    controlP5.addBang("secondBang", 10, 60, 40, 20 );  
    controlP5.controller("firstBang").setLabel("Red");  
    controlP5.controller("secondBang").setLabel("Blue");  
}  
  
void draw() {  
    background(c);  
}  
  
void firstBang() {  
    c = color(255,0,0);  
}  
  
void secondBang() {  
    c = color(0,0,255);  
}
```

# controlP5

## Button

- addButton(theName, **theValue**, theX, theY, theW, theH);
- Load example: controlP5\button

```
import controlP5.*;

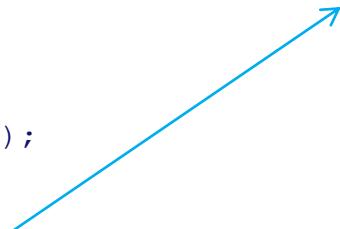
ControlP5 controlP5;
int c = 128;
void setup() {
    size(400, 400);
    controlP5 = new ControlP5(this);
    controlP5.addButton("black", 0, 10, 10, 40, 20 );
    controlP5.addButton("grey", 128, 10, 50, 40, 20);
    controlP5.addButton("white", 255, 10, 90, 40, 20 );
}

void draw() {
    background(c);
}

void black(float value) { c = (int)value; }

void grey(float value) { c = (int)value; }

void white(float value){ c = (int)value; }
```



# controlP5

## Toggle

- addToggle(theName, **theDefaultValue**, theX, theY, theWidth, theHeight);
- Load example: controlP5\toggle

```
import controlP5.*;

ControlP5 controlP5;
int c = 0;
boolean isRound = false;
void setup() {
    size(400, 400);
    controlP5 = new ControlP5(this);
    controlP5.addToggle("on", false, 10, 10, 10, 10 );
    controlP5.addToggle("round", false, 50,10,10,10 );
}

void draw() {
    background(204);
    fill(c);
    if(isRound)
        ellipse(200,200,200,200);
    else
        rect(100,100,200,200);
}
```

```
void on(boolean value) {
    if(value) c = 255;
    else c = 0;
}

void round(boolean value) {
    isRound = value;
}
```



# controlP5

## Radio buttons

- addRadio(theName, theX, theY);
- Load example: controlP5\radio

```
import controlP5.*;
ControlP5 controlP5;
color c = color(255,0,0);
void setup() {
    size(400, 400);
    controlP5 = new ControlP5(this);
    Radio r = controlP5.addRadio("myradio", 10,10);
    r.addItem("red", 0);
    r.addItem("green", 1);
    r.addItem("blue", 2);
}
void draw() {
    fill(c); rect(100,100,200,200);
}
void myradio(int value) {
    switch(value){
        case 0: c = color(255,0,0); break;
        case 1: c = color(0,255,0); break;
        case 2: c = color(0,0,255); break;
    }
}
```



# controlP5

## Numberbox

- addNumberbox(theName, **theDefaultValue**, theX, theY, theWidth, theHeight);
- Load example: controlP5\numberbox

```
import controlP5.*;

ControlP5 controlP5;
int c = 128;
void setup() {
    size(400, 400);
    controlP5 = new ControlP5(this);
    controlP5.addNumberbox("wierdbox", 128, 10, 10, 80, 15);
}

void draw() {
    fill(c); rect(100,100,200,200);
}

void wierdbox(int value) {
    if(value>255) c = 255;
    else if(value<0) c = 0;
    else c = value;
}
```

# controlP5

## Slider

- addSlider(theName, **theMin**, **theMax**, **theDefaultValue**, theX, theY, theW, theH);
- Load example: controlP5\slider

```
import controlP5.*;

ControlP5 controlP5;
int c = 128;
void setup() {
    size(400, 400);
    controlP5 = new ControlP5(this);
    controlP5.addSlider("mygod", 0, 255, 128, 10, 10, 200, 15);
}

void draw() {
    fill(c); rect(100,100,200,200);
}

void mygod(int value) {
    c = value;
}
```

# controlP5

## Knob

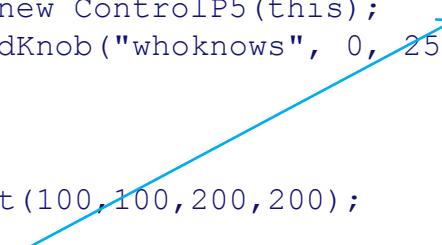
- addKnob(theName, theMin, theMax, theDefaultValue, theX, theY, theDiameter);
- Load example: controlP5\knob

```
import controlP5.*;

ControlP5 controlP5;
int c = 128;
void setup() {
    size(400, 400);
    smooth();
    controlP5 = new ControlP5(this);
    controlP5.addKnob("whoknows", 0, 255, 128, 175, 175, 50);
}

void draw() {
    fill(c); rect(100,100,200,200);
}

void whoknows(int value) {
    c = value;
}
```



# controlP5

## Textfield

- addTextfield(theName, theX, theY, theW, theH);
- Load example: controlP5\textfield

```
import controlP5.*;

ControlP5 controlP5;
int c = 128;
void setup() {
    size(400, 400);
    controlP5 = new ControlP5(this);
    controlP5.addTextfield("eindhoven", 10, 10, 200, 20);
}

void draw() {
    fill(c); rect(100,100,200,200);
}

void eindhoven(String value) {
    println(value);
    c = int(value);
}
```

# controlP5

## Textlabel

- addTextlabel(theName, theText, theX, theY);
- Load example: controlP5\textlabel

```
import controlP5.*;

ControlP5 controlP5;
int c = 128;
Textlabel label;
void setup() {
  size(400, 400);
  controlP5 = new ControlP5(this);
  controlP5.addTextfield("eindhoven", 10, 10, 200, 20);
  label = controlP5.addTextlabel("delft","Now, type something, or try a number", 10, 60);
}

void draw() {
  background(64);
  fill(c); rect(100,100,200,200);
}

void eindhoven(String value) {
  label.setValue(value);
  c = int(value);
}
```