LUÍS GONÇALO
&& JÉSSICA PARENTE
(CDV LAB)

08.12.2021
11H00

DEI — FCTUC
(ROOM E4.1)
&& REMOTE

WORKSHOP
PROCESSING FOR ANDROID

PROCESSING
COMMUNITY
DAY **COIMBRA**
2021

MORE INFO
@pcdcoimbra
pcdcoimbra.dei.uc.pt

1 2 9 0 FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE D COIMBRA

dei departamento de engenharia informática

PROCESSING FOR ANDROID
ANDROID MODE

A **programming mode** for Processing that adds all the options needed to **run Processing sketches** on **Android devices** and also in the **emulator**.

https://android.processing.org/

CONS

– Creating **android apps** easily, creating live wallpapers, watch faces, and VR/AR apps.
– Running **Processing sketches** on **Android devices,** smartphones, tablets, smartwatches.
– Using **Android hardware** to **retrieve data**, for instance **sensor data.**
– Exporting the sketch as a **signed package.**

IN THE WORKSHOP
WE WILL

– Explore the use of the **Android hardware**, **read sensor data,** use **different types of touches** and the **camera**.

– Use **ke:tai**, an extensive library to Processing that gives you **straight-forward access** to **android hardware**.

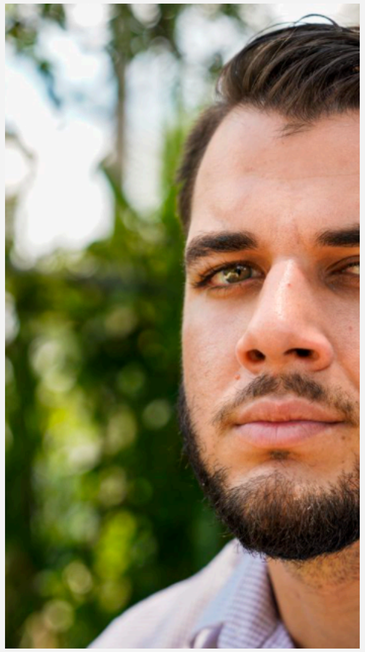
http://ketai.org/

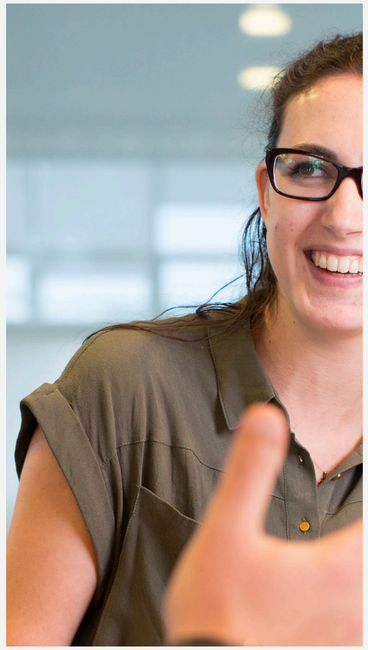
IN THE WORKSHOP
WE WILL

Develop an application that makes **a representation of a photograph composed with circles**. To do that, the will:

– Take a **picture** with the smartphone camera.
– Then, use the **accelerometer** to draw circles based on the color of the picture taken.

IN THE WORKSHOP
EXAMPLES





IN THE WORKSHOP
YOU NEED TO HAVE

– **Processing 3 installed**.
– An **Android smartphone** and a **cable** to connect to the computer or optionally run on an **emulator.**

https://github.com/pdc-coimbra/workshop-2021-Processing-for-Android

HANDS ON.

ANDROID MODE
INSTALLATION

– **Open Processing**.
– In the upper right corner click on the **arrow** where it says Java (the default for Processing) and click on “**Add Mode**”.
– On the panel that appears, select **Android Mode** and install.
– Then, click again on the **arrow** in the upper right corner and select **Android**.

ANDROID MODE
INSTALLATION

– Then, on the panel that appears, click on “**Download SDK automatically**”. Wait a few minutes.
– Then, you have to **accept** the **SDK license**.
– Then, click “**OK”** to the confirmation that the Android mode has been installed successfully in Processing.

#APP 0

```
import ketai.sensors.*;
KetaiSensor sensor;

void setup(){
  sensor = new KetaiSensor(this);
  println(sensor.list());
}
```

#APP 1

```
import ketai.sensors.*;
KetaiSensor sensor;
float accelerometerX, accelerometerY, accelerometerZ, light;

void setup(){
  sensor = new KetaiSensor(this);
  sensor.start();
  orientation(PORTRAIT);
  textAlign(CENTER, CENTER);
  textSize(72);
  fill(255, 0, 0); }
```

#APP 1

```
void draw(){
  float value = map(light, 0, 600, 0, 255);
  background(value);
  text("Accelerometer: \n" +
    "x: " + accelerometerX + "\n" +
    "y: " + accelerometerY + "\n" +
    "z: " + accelerometerZ + "\n" +
    "light: " + light, width/2, height/2);
}
```

#APP 1

```
void onAccelerometerEvent(float x, float y, float z){
  accelerometerX = x;
  accelerometerY = y;
  accelerometerZ = z;
}

void onLightEvent(float l){
  light = l;
}
```

#APP 2 BASIS

```
void setup(){
   fullScreen();
   orientation(LANDSCAPE);
}

void draw(){

}
```

#APP 2 CAMERA

```
import ketai.camera.*;
KetaiCamera cam;

void setup(){
  (…)
  cam = new KetaiCamera(this, width, height, 24);
  cam.start();
}
```

#APP 2 CAMERA

```
void draw(){
    if(cam.isStarted()){
        image(cam,0,0);
    }
}

void onCameraPreviewEvent(){
    cam.read();
}
```

```
void exit(){
    cam.stop();
}
```

CAMERA PERMISSION: ANDROID -> SKETCH PERMISSIONS -> CHECK “CAMERA” -> OK

#APP 2 SAVE CAMERA IMG

```
import ketai.ui.*;
import android.view.MotionEvent;
KetaiGesture gesture;
PImage img;

void setup(){
  (…)
  gesture = new KetaiGesture(this);
}
```

#APP 2 SAVE CAMERA IMG

```
public boolean surfaceTouchEvent(MotionEvent event) {
  return gesture.surfaceTouchEvent(event);}

void onDoubleTap(float x, float y){
  img = createImage(width,height,RGB);
  img.copy(cam,0,0,width,height,0,0,width,height);
  cam.stop();
  background(255);
  tint(255, 50);
  image(img,0,0);
}
```

#APP 2 ADD ACCELEROMETER

```
import ketai.sensors.*;
KetaiSensor sensor;

PVector accelerometer;
PVector position;
float ball_size = 20;

void onAccelerometerEvent(float x, float y, float z, long time, int
accuracy){
  accelerometer.set(x, y, z);
}
```

#APP 2 ADD ACCELEROMETER

```
void setup(){
  (…)
  position = new PVector(width/2, height/2);
  sensor = new KetaiSensor(this);
  sensor.start();
  accelerometer = new PVector();
}
```

#APP 2 ADD ACCELEROMETER

```
void draw(){
  if(cam.isStarted()){ image(cam,0,0);}
  }else if (img != null){
    PVector rollDirection = new PVector(accelerometer.y * 2,
    accelerometer.x * 2);
    position.add(rollDirection);
    position.x = constrain(position.x, ball_size/2, width-(ball_size/2));
    position.y = constrain(position.y, ball_size/2, height-(ball_size/2));
    ellipse(position.x,position.y,ball_size,ball_size);
  }
}
```

#APP 2 ADD PIXEL COLOUR

```
void draw(){
  if(cam.isStarted()){ image(cam,0,0);}
  }else if (img != null){
     (…)
     img.loadPixels();
     int x = int(position.x);
     int y = int(position.y);
     int loc = x + y*img.width;
     fill(img.pixels[loc]);
     ellipse(position.x,position.y,ball_size,ball_size);}
 }
```

```
void setup(){
   (…)
   noStroke();
}
```

#APP 2 CHANGE BALL SIZE

```
void onPinch(float x, float y, float d){
    ball_size += d;
    ball_size = constrain(ball_size, 10, 200);
}
```

THANK YOU.

Submit your results, here:

https://forms.gle/ge2gBzEQaKFvH4mD8

Jéssica Parente jparente@dei.uc.pt
Luís Gonçalo lgoncalo@dei.uc.pt