The code for iProcessing presents some differences from the code for Processing.
We divided the main code in three different tabs:

- -main
- -logic
- -graphics.

The logic tab manages the actions, such as calling each menu and screens.
The graphic tab loads and unloads the images and holds the functions that are called by the logic tab.

This is the logic tab which loads all the different screens of the application in a hierarchy that reflects the order of apparance.
For example this is the first part of the application in which we load the background and the image of the logo. Also we load a little animation in which we want to simulate the loading of application.
At first we get the name of the image, so in "void loadLogo()" we load the image by the data folder, finally in "void drawLogo()" we put the image in the screen and get the coordinates x, y.

```
PImage sfondo;
PImage logo;

float x =18.1;
float y =250;
float s =6.8;
float g = 17;

void loadLogo()
{
  sfondo = loadImage("griglia.jpg");
  logo = loadImage("logo.png");
  imageMode (CENTER);
  rectMode (CENTER);
  frameRate (10);
}

void drawLogo()
{
  image(sfondo, 240, 160);
  image(logo, 236,160);
}

////////////////////////////////////////////////////////////////////////////////
```

In the same way we load every part of application, name the images, load the image and put the image on the screen with the coordinates.
Now we want to explain the relevant part of the application that compares in the logic tab.
For example, in the map part we must pulse the place dots, then we want to ensure that the dots are clickable twice;
once:

- where is the place;
- if the performance is already started;
- if we can add at performance.

twice:

- select the place and go to the next screen.

Now we below identified this part of code.

```
PImage mappa;
PImage pallino;
PImage pallino2;
PImage pallino3;
PImage pallino4;
PImage pallino5;
PImage pallino6;

PImage stefano;
PImage apostoli;
PImage margherita;
PImage bartolomeo;
PImage elena;
PImage polo;
PImage croce;

int pulse=1;
int sceltaAct=0;
int oldSceltaAct=0;

int mappaAct=0;
int oldMappaAct=0;

void loadMappa()
{
  sfondo = loadImage("griglia.jpg");
  mappa = loadImage("mappa.png");
  pallino = loadImage ("pallino.png");
  pallino2 = loadImage ("pallino2.png");
  pallino3 = loadImage ("pallino3.png");
  pallino4 = loadImage ("pallino4.png");
  pallino5 = loadImage ("pallino5.png");
  pallino6 = loadImage ("pallino6.png");

  stefano = loadImage ("stefano.png");
  apostoli = loadImage ("apostoli.png");
  margherita = loadImage ("margherita.png");
  bartolomeo = loadImage ("bartolomeo.png");
  elena = loadImage ("elena.png");
  polo = loadImage ("polo.png");
  croce =loadImage ("cruz.png");
  imageMode (CENTER);
}

void drawMappa()
{

  image(sfondo, 240, 160);
  float a=190;
  float pulse= abs(sin(frameCount/30.0))*85.0;                     to define the frame count of the pulse
  noTint();
  image(sfondo, 240, 160);
  image(mappa, 242.802,159);
  image(croce,81,209);
  noTint();
  tint(a+pulse);                                                   to have pulse of dots
  image(pallino, 257,93);
  image(pallino2,291,126);
  image(pallino3,190,159);
  image(pallino4,156,192);
  image(pallino5,248.6,192);
  image(pallino6,450.23,225);
  noTint();

  if(mappaAct == 1) {                                              to click twice
    image (apostoli,257,76.9);
  }

  if(mappaAct == 2) {
    image (bartolomeo,291.5,109.5);
  }
  if(mappaAct == 3) {
    image (polo,190.5,142);
  }
  if(mappaAct == 4) {
    image (margherita,156,175);
  }
  if(mappaAct == 5) {
    image (stefano,249.5,175.9);
  }
  if(mappaAct == 6) {
    image (elena,450,208);
  }
}


///////////////////////////////////////////////////////////////////////////////////////////
```

This part of the code we have creates a sequencer that is able to draws the icon of a button and plays the relevant sound when I press on any beat of the positioning grid.
So we comment below on a single part on the code.

```
float a =  212;
float b =  60;
float c= 59;
float d= 50;

boolean simbolo_status;                                   this part defines the name of the boolean that we use as a
                                                          sequencer function
boolean[] onOff;

PImage schermata;
PImage bit1;
PImage strumentolight5;
PImage bitsingolo;
int position = 0;

IPhone myIPhone = new IPhone();                           this command we use for the sound
PSound suono;

void loadComposizione() {


  suono = myIPhone.loadSound("alto basso.mp3");           we load the sound
  frameRate(30);                                          we define a frame rate for the speed of the metronome
  sfondo = loadImage ("griglia.jpg");

  bit1 = loadImage ("bit1.png");
  schermata = loadImage ("schermata.png");
  strumento5 = loadImage ("5.png");
  strumentoblu5 = loadImage ("5b.png");
  strumentolight5 =loadImage ("strumentolight5.png");
  bitsingolo = loadImage ("++.png");
  indietro  = loadImage("indietro.png");

  imageMode(CENTER);
  simbolo_status = false;                            we say that the status of the symbol at the start is false this to
                                                     start the icon inactive
  onOff = new boolean[16];                           we define this boolean as valid for the other sixteen booleans

  for(int i=0; i<ONOFF.length; i++) {                this loop controls the position that we give to all beats
    onOff[i] = false;                                we say that the status of the "onOff" at the start is false
                                                     to start the icon inactive
  }
}

void drawComposizione() {

  image(sfondo,240,160);
  image (schermata, 312,160);

  image(bitsingolo,a,b,c,d);
  image (indietro,13,160);
  a +=67;
  if (a >= 467) {
    a= 212;
    b+= 66;
  }
  if ( b >258) {
    a =212;
    b=60;
  }



  if (checkNumberActive() > 0) {                                This controls the pulse of the icon depending on the
                                                                position of the beats in the composition grid
    image (strumentolight5,89,159);
  }
  else {
    image (strumento5, 89, 159);
  }
  if(onOff[position]) {                                         when we click on one or the all beats
                                                                the icon becomes active and reproduces the sound.
    suono.rewind();
    suono.play();
    image (strumentoblu5,89,159);
  }
  if(position < 15 ) {
    position++;
  }
  else {
    position = 0;
  }

  if(onOff[0]) {                                                when we click on the positioning grid we activate the beat
                                                                selected
    image (bit1,210,60);
  }
  if(onOff[1]) {
    image (bit1,277,60);
  }
  if(onOff[2]) {
    image (bit1,345,60);
  }
  if(onOff[3]) {
    image (bit1,412,60);
  }
  if(onOff[4]) {
    image (bit1,210,126);
  }
  if(onOff[5]) {
    image (bit1,277,126);
  }
  if(onOff[6]) {
    image (bit1,345,126);
  }
  if(onOff[7]) {
    image (bit1,412,126);
  }
  if(onOff[8]) {
    image (bit1,210,192);
  }
  if(onOff[9]) {
    image (bit1,277,192);
  }
  if(onOff[10]) {
    image (bit1,345,192);
  }
  if(onOff[11]) {
    image (bit1,412,192);
  }
  if(onOff[12]) {
    image (bit1,210,258);
  }
  if(onOff[13]) {
    image (bit1,277,258);
  }
  if(onOff[14]) {
    image (bit1,345,258);
  }
  if(onOff[15]) {
    image (bit1,412,258);
  }
}

int checkNumberActive() {
```

this part of the code we defines the condition to turn on the beats and to turn off the beat by the selection

```
  int num = 0;
  for(int i=0; i<ONOFF.length; i++) {
    if(onOff[i] == true) {
      num++;
    }
  }
  return num;
}
```

///

This is the part of code in which we control the instrument during the performance. so we have some control:

 - sound
 - tonality
 - volume
 - filter1
 - filter2
 - interaction

 We comment below the relevant function of this code part

```
PImage interazione;


int numFrames = 8;
int frameAnimation1;
int frameAnimation2;
```

we define the frame number of the animation of the filter
we get the name of the two different animations of the filter

```
PImage[] imagesMod = new PImage[numFrames];
PImage sfondo;
PImage interaction;

Boolean touchCheck = false;
```

this is the name of boolean that we use to change the tonality of the instrument’s sound

```
void loadStrumento2()
{

  suono2 =myIPhone.loadSound("suono2.mp3");

  strumentoblu2   = loadImage ("2b.png");
  sfondo = loadImage ("griglia.jpg");
  interazione =loadImage ("inter.png");
  indietro  = loadImage("indietro.png");
  imageMode(CENTER);

  imagesMod[0]  = loadImage("mod1.png");
```

we load the image needed to draw the filter animation

```
  imagesMod[1]  = loadImage("mod2.png");
  imagesMod[2]  = loadImage("mod3.png");
  imagesMod[3]  = loadImage("mod4.png");

  imagesMod[5]  = loadImage("mod5.png");
  imagesMod[6]  = loadImage("mod6.png");
  imagesMod[7]  = loadImage("mod7.png");
  imagesMod[8]  = loadImage("mod8.png");

  myIPhone.startAccelerometer();
```

we load the accelerometer function

```
}
void drawStrumento2()
{
  image(sfondo, 240, 160);

  image (interazione,452,41.5);
  image (strumentolight2,e1,160);
  image (indietro,13,160);

  if(onOff2[position]) {
```

we put in the instrument the same sound that we have previously created in the positioning grid

```
    suono2.rewind();
    suono2.play();
    image (strumentoblu2,e1,160);                             Also we give to the instrument the pulse which corresponds to
in corrispondence
                                                              the sound beat
  }
  if(position < 15 ) {
    position++;
  }
  else {
    position = 0;
  }

  float xA = myIPhone.getAcceleration().x;                    we define that the animation starts only when we turn forward
                                                              and back(filter1), and left and right(filter2)the iPhone

  float yA = myIPhone.getAcceleration().y;
  if(xA > 0) {                                                we define the condition that controls the animation start
    if(xA < 0.5) {
      frameAnimation2 = int(map(xA, 0, 0.5, 1, 5));
    }
    else {
      frameAnimation2 = 4;
    }
  }
  else {
    if(xA > -0.5) {
      frameAnimation2 = int(map(xA, 0, -0.5, 1, 5));
    }
    else {
      frameAnimation2 = 4;
    }
  }

  if(yA < 0) {
    if(yA > -0.5) {
      frameAnimation1 = int(map(yA, 0, -0.5, 5, 10));
    }
    else {
      frameAnimation1 = 9;
    }
  }
  else {
    if(yA < 0.5) {
      frameAnimation1 = int(map(yA, 0, 0.5, 5, 10));
    }
    else {
      frameAnimation1 = 9;
    }
  }


  if(frameAnimation1 >= 5 && frameAnimation1 <= 9) {
    for(int i = 5; i < frameAnimation1; i++) {
      image(imagesMod[i], e1,160);
    }
  }

  if(frameAnimation2 >= 1 && frameAnimation2 <= 4) {
    for(int i = 1; i < frameAnimation2; i++) {
      image(imagesMod[i], e1,160);
    }
  }
}
```

///

This is the logic tab of the code, which loads every screen in the application.
In the first part we recall all the screens that we have draw in the graphic part.
In the void setup part we load these screens

```
int SCREEN_LOGO = 0;
int SCREEN_MAPPA= 1;
int SCREEN_GOTOMARGHERITA = 2;
int SCREEN_GOTOBARTOLOMEO = 3;
int SCREEN_GOTOSTEFANO = 4;
int SCREEN_ALREADY =5;
int SCREEN_OKMARGHERITA = 6;
int SCREEN_OKBARTOLOMEO = 7;
int SCREEN_OKSTEFANO = 8;
int SCREEN_SCELTA =9;
int SCREEN_COMPOSIZIONE =10;
int SCREEN_STRUMENTO2 =11;
int SCREEN_COMPOSIZIONE_RULLO =12;
int SCREEN_STRUMENTO5 =13;
int SCREEN_INTERAZIONE = 14;
int SCREEN_INTERAZIONE2 = 15;

int screenMode = SCREEN_LOGO;
```

```
void setup()
{
  size(480,320);

  loadLogo();
  loadMappa();
  loadGotomargherita();
  loadGotobartolomeo();
  loadGotostefano();
  loadAlready();
  loadOkmargherita();
  loadOkbartolomeo();
  loadOkstefano();
  loadScelta();
  loadComposizione();
  loadStrumento2();
  loadComposizioneRullo();
  loadStrumento5();
  loadInterazione();
  loadInterazione2();
}

////////////////////////////////////////////////////////////////////////////////////////////
```

This part draws all the screens and we gives the main command. We comment below
on the most relevant functions in this code part

```
void draw()
{
  if(screenMode == SCREEN_LOGO) {          in this part we draw the first screen with logo and the simulation of application
                                           loading
    drawLogo();
    noStroke();
    fill(223,223,223);
    rect(x,y, s, g);
    x += 8.38;

    if (x >= 454) {                        we define that when the loading animation arrives at the last position the
                                           application loads the next screen
      x= 454;
      drawMappa();
      screenMode = SCREEN_MAPPA;
    }
  }
```

This part of the code identifies the function of tonality, and permits us to move the instrument icon on the screen.
We have decided a range of movement only on the X axis, so we use the constrain function.

```
  float e1 = 240;
  float e2 = 240;

  void touch1Moved() {
    if (screenMode==SCREEN_STRUMENTO2 ) {
      e1 = constrain(touch1X,100,380);
    }
    if (screenMode==SCREEN_STRUMENTO5 ) {
      e2 = constrain(touch1X,100,380);
    }
  }


  ////////////////////////////////////////////////////////////////////////////////////////////
```

In this part," void mousePressed()", we have defined that when we click on a preset area we load another screen or return to the preceding screen.
we have use this function every time we must return to another screen.
In this first part, in the void mousePressed, we have many events .
For example in the screen mappa we write the code to enable twice click on the dots;
in the same way, in the part of instrument choice we use this function and add in the code the part relative the sound.
So below we report few part of code in which compare this function.

```
  void mousePressed()
  {
    if(screenMode == SCREEN_MAPPA) {

      if (mouseX >  257 -20  && mouseX <  257 + 20  && mouseY >  94 - 20 && mouseY <  94 + 20 ) {
        mappaAct = 1;
      }

      if (mouseX >  291 -20  && mouseX <  291 + 20  && mouseY >  127 - 20 && mouseY <  127 + 20 ) {
        mappaAct = 2;
      }

      if (mouseX >  190 -20  && mouseX <  190 + 20  && mouseY >  160 - 20 && mouseY <  160 + 20 ) {
        mappaAct = 3;
      }

      if (mouseX >  156 -20  && mouseX <  156 + 20  && mouseY >  193 - 20 && mouseY <  193 + 20 ) {
        mappaAct = 4;
      }

      if (mouseX >  248.6 -20  && mouseX <  248.6 + 20  && mouseY >  193 - 20 && mouseY <  193 + 20 ) {
        mappaAct = 5;
      }

      if (mouseX >  450.23 -20  && mouseX <  450.23 + 20  && mouseY >  226 - 20 && mouseY <  226 + 20 ) {
        mappaAct = 6;
      }



      if(oldMappaAct == mappaAct) {
        if(mouseX >  156 -20  && mouseX <  156 + 20  && mouseY >  193 - 20 && mouseY <  193 + 20 ) {
          screenMode = SCREEN_GOTOMARGHERITA;
        }
        if(mouseX >  291 -20  && mouseX <  291 + 20  && mouseY >  127 - 20 && mouseY <  127 + 20) {
          screenMode = SCREEN_GOTOBARTOLOMEO;
        }
        if(mouseX >  248.6 -20  && mouseX <  248.6 + 20  && mouseY >  193 - 20 && mouseY <  193 + 20 ) {
          screenMode = SCREEN_GOTOSTEFANO;
        }
        if (mouseX >  257 -20  && mouseX <  257 + 20  && mouseY >  94 - 20 && mouseY <  94 + 20 ) {
          screenMode = SCREEN_ALREADY;
        }
        if (mouseX >  190 -20  && mouseX <  190 + 20  && mouseY >  160 - 20 && mouseY <  160 + 20 ) {
          screenMode = SCREEN_ALREADY;
        }
        if (mouseX >  450.23 -20  && mouseX <  450.23 + 20  && mouseY >  226 - 20 && mouseY <  226 + 20 ) {
          screenMode = SCREEN_ALREADY;
        }
      }

      oldMappaAct = mappaAct;
    }

    //////////////////////////////////////////////////////////////////////////////////////////////////////////////

    if(screenMode == SCREEN_SCELTA) {

      if (mouseX >  97 -50  && mouseX <  97 + 50  && mouseY >  94 - 50 && mouseY <  94 + 50 ) {
        sceltaAct = 1;
      }
      else if (mouseX >  240 -50  && mouseX <  240 + 50  && mouseY >  94 - 50 && mouseY <  94 + 50 ) {
        sceltaAct = 2;

        if(oldSceltaAct != sceltaAct) {
          suono2.rewind();
          suono2.play();
        }
      }
```