

ADEPTS - PROCESSING CODE

```
// *****
// Some variables
// *****

PImage logo; // image of the homepage
PImage menuAlert, page1, page1_politics, page2, alert1map, page3, page_map, page_start_bubble;
PImage aiuto1, aiuto2;

int advise;

boolean numB;
PImage prova1, prova2;

TopMenu topMenu;
BottomMenu bottomMenu;

int PAGE_TITLE = 0;
int PAGE_MAIN = 1;
int PAGE_ALERT = 2;
int PAGE_BUBBLE_BIG = 3;
int PAGE_PHRASE = 4;
int PAGE_MAP = 5;
int PAGE_START_BUBBLE = 6;

int page = PAGE_TITLE;

boolean SEND_BUBBLE;
boolean aiuto;

int num_foto;
String[] imageName;
String path;
PImage[] foto;
int n;

// *****
// Setup
// *****

void setup(){
  size(320,480); // iPhone resolution
  background(255);
  advise = 0;
  page = 0;
  topMenu = new TopMenu();
  bottomMenu = new BottomMenu();
  logo = loadImage("logo.png");
  setupSpeech();
  menuAlert = loadImage("menu_alert2.png");
  page1 = loadImage("page1.png");
  page1_politics = loadImage("page1_politics.png");
  page2 = loadImage("page2.png");
  alert1map = loadImage("alert1map.png");
  page3 = loadImage("page3.png");
  page_map = loadImage("page_map.png");
  page_start_bubble = loadImage("page_start_bubble.png");
  aiuto1 = loadImage("aiuto1.png");
  aiuto2 = loadImage("aiuto2.png");

  path = sketchPath+"/data/bubble/";
  imageName = listFileNames(path);
  num_foto = imageName.length - 1;
  println("Animation frames loaded: " + num_foto);
  foto = new PImage[num_foto];
  for(int i=1; i<imageName.length; i++){
    foto[i-1] = loadImage(path+imageName[i]);
  }
}
```

```

}

// *****
SEND_BUBBLE = true;
// true : sending tag mode
// false : visualization mode
// *****

prova1 = loadImage("prova1.png");
prova2 = loadImage("prova2.png");

numB = true;
aiuto = true;

setupFloatingBubble();
}

// *****
// Show pages
// *****

void showAlert(int section){
  page = PAGE_ALERT;
  bottomMenu.setButtons(1,2,0,1,1,false);
  topMenu.setButtons(1,1,1);
}

void showMain(){
  page = PAGE_TITLE;
  topMenu.setButtons(0,1,1);
  bottomMenu.setButtons(1,1,0,1,1,false);
  advise = 0;
}

void showPageBubbleBig(){
  page = PAGE_BUBBLE_BIG;
  topMenu.setButtons(1,1,1);
  bottomMenu.setButtons(1,1,1,0,1,false);
}

void showPagePhrase(){
  page = PAGE_PHRASE;
  topMenu.setButtons(1,1,1);
  bottomMenu.setButtons(1,1,1,0,1,false);
}

void showPageMap(){
  page = PAGE_MAP;
  topMenu.setButtons(1,1,1);
  bottomMenu.setButtons(1,1,2,0,1,false);
}

void showStartBubble(){
  page = PAGE_START_BUBBLE;
  n = 0;
  if(advise == 0){
    advise = 1;
  }
  else if(advise == 1){
    advise = 2;
  }
  else if(advise == 2){
    advise = 0;
  }
}
}

```

```

// *****
// Events listener
// *****

void mousePressed(){ // manage mouse event

    if(page == PAGE_PHRASE){
        if(mouseX >= 116 && mouseX <= 174 && mouseY >=425 && mouseY <=480){
            showPageMap();
        }
    }

    if(page == PAGE_BUBBLE_BIG){
        if(mouseX >= 0 && mouseX <= 320 && mouseY >=55 && mouseY <=425){
            showPagePhrase();
        }
    }

    if(page == PAGE_MAIN){
        if(mouseX >= 0 && mouseX <= 320 && mouseY >=55 && mouseY <=425){
            showPageBubbleBig();
        }
    }

    // swich to the main screen
    if(page == PAGE_TITLE){
        page = PAGE_MAIN;
    }

    // open/close the menu
    if(page >= PAGE_MAIN){
        if(!topMenu.isOpen && !bottomMenu.isOpen){
            if(mouseX >= 265 && mouseX <= 320 && mouseY >=0 && mouseY <=55){
                topMenu.show();
                bottomMenu.show();
            }
        }
        else {
            if(mouseX >= 265 && mouseX <= 320 && mouseY >=0 && mouseY <=55){
                topMenu.hide();
                bottomMenu.hide();
            }
        }
    }

    // when I click somewhere on the botton menu
    if(bottomMenu.isOpen){
        bottomMenu.onPressBottom(); // this function check all the click to icons
    }

    // when the user click the speech recognizer button
    if(topMenu.isOpen){
        topMenu.onMousePressed(); // this function check all the click to icons
    }

    //if(!topMenu.isOpen){
        if(mouseX >= 265 && mouseX <= 320 && mouseY >=425 && mouseY <=480){
            showStartBubble();
        }
    //}

}

```

```

}

void keyPressed(){

if(keyPressed) {
  if (key == 'b') {
    SEND_BUBBLE = true;
  }
  if (key == 'n') {
    SEND_BUBBLE = false;
  }
  if(key == 'm'){
    showStartBubble();
  }
}

}

}

```

```

// *****
// Bottom Menu Class
// *****

```

```

class BottomMenu
{
  float y = 480;
  float yTarget = 480;
  PImage img;
  boolean isOpen = false;

  int[] statusMenu;

  PImage alert0;
  PImage alert1;
  PImage alert2;
  PImage home0;
  PImage home1;
  PImage home2;
  PImage map0;
  PImage map1;
  PImage map2;
  PImage modeTag0;
  PImage modeTag1;
  PImage modeTag2;
  PImage modeUser0;
  PImage modeUser1;
  PImage modeUser2;
  PImage sestiere0;
  PImage sestiere1;
  PImage sestiere2;

  boolean user;

  BottomMenu(){
    hide();
    img = loadImage("bottomMenu.png");
    alert0 = loadImage("ic_alert0.png");
    alert1 = loadImage("ic_alert1.png");
    alert2 = loadImage("ic_alert2.png");
    home0 = loadImage("ic_home0.png");
    home1 = loadImage("ic_home1.png");

```

```

home2 = loadImage("ic_home2.png");
map0 = loadImage("ic_map0.png");
map1 = loadImage("ic_map1.png");
map2 = loadImage("ic_map2.png");
modeTag0 = loadImage("ic_modeTag0.png");
modeTag1 = loadImage("ic_modeTag1.png");
modeTag2 = loadImage("ic_modeTag2.png");
modeUser0 = loadImage("ic_modeUser0.png");
modeUser1 = loadImage("ic_modeUser1.png");
modeUser2 = loadImage("ic_modeUser2.png");
sestiere0 = loadImage("ic_sestiere0.png");
sestiere1 = loadImage("ic_sestiere1.png");
sestiere2 = loadImage("ic_sestiere2.png");
statusMenu = new int[5];
setButtons(1,1,0,1,1,false);
topMenu.setButtons(0,1,1);
}

void move(float y){
  this.yTarget = y;
}

void show(){
  move(425);
  isOpen = true;
}

void hide(){
  move(480);
  isOpen = false;
}

void update(){
  y += (yTarget - y) * 0.2;
  image(img,0,y);
  drawButtons();
}

boolean mouseIsTouchingHome(){
  if(mouseX >= 0 && mouseX <= 50 && mouseY >=435 && mouseY <=480){
    return true;
  }
  return false;
}

boolean mouseIsTouchingAlert(){
  if(mouseX >= 55 && mouseX <= 110 && mouseY >=435 && mouseY <=480){
    return true;
  }
  return false;
}

void onPressBottom(){
  if(mouseIsTouchingHome()){ // home
    if(bottomMenu.statusMenu[0] == 1){
      bottomMenu.statusMenu[0] = 2;
      showMain();
    }
  }
  if(mouseIsTouchingAlert()){ // alert
    if(bottomMenu.statusMenu[1] == 1){
      bottomMenu.statusMenu[1] = 2;
      showAlert(1);
    }
  }
}

void setButtons(int a, int b, int c, int d, int e, boolean user){
  /*
  user = mode_type (user/tag) [true, false] (if user==true then we're in user mode)
  */
  this.statusMenu[0] = a;
}

```

```
this.statusMenu[1] = b;
this.statusMenu[2] = c;
this.statusMenu[3] = d;
this.statusMenu[4] = e;
this.user = user;
}
```

```
void drawButtons(){
```

```
int x_home = 0;
int x_alert = 58;
int x_map = 116;
int x_sestiere = 174;
int x_mode = 242;
```

```
switch(statusMenu[0]) {
case 0:
    image(home0,x_home,y);
    break;
case 1:
    image(home1,x_home,y);
    break;
case 2:
    image(home2,x_home,y);
    break;
}
```

```
switch(statusMenu[1]) {
case 0:
    image(alert0,x_alert,y);
    break;
case 1:
    image(alert1,x_alert,y);
    break;
case 2:
    image(alert2,x_alert,y);
    break;
}
```

```
switch(statusMenu[2]) {
case 0:
    image(map0,x_map,y);
    break;
case 1:
    image(map1,x_map,y);
    break;
case 2:
    image(map2,x_map,y);
    break;
}
```

```
switch(statusMenu[3]) {
case 0:
    image(sestiere0,x_sestiere,y);
    break;
case 1:
    image(sestiere1,x_sestiere,y);
    break;
case 2:
    image(sestiere2,x_sestiere,y);
    break;
}
```

```
switch(statusMenu[4]) {
case 0:
    if(user){
        image(modeUser0,x_mode,y);
    }
    else{
        image(modeTag0,x_mode,y);
    }
    break;
case 1:
    if(user){
        image(modeUser1,x_mode,y);
    }
}
```

```

else{
    image(modeTag1,x_mode,y);
}
break;
case 2:
if(user){
    image(modeUser2,x_mode,y);
}
else{
    image(modeTag2,x_mode,y);
}
break;

}
}

}

// *****
// Top Menu Class
// *****

class TopMenu
{
float y = -55;
float yTarget = -55;
boolean isOpen = false;
PImage img;
PImage openMenu;

PImage back0;
PImage back1;
PImage back2;
PImage close1;
PImage speech1;

int[] statusMenu;

TopMenu(){
hide();
img = loadImage("topMenu.png");
openMenu = loadImage("openMenu.png");
back0 = loadImage("ic_back0.png");
back1 = loadImage("ic_back1.png");
back2 = loadImage("ic_back2.png");
close1 = loadImage("ic_close1.png");
speech1 = loadImage("ic_speech1.png");
statusMenu = new int[3];
setButtons(0,1,1);

}

void move(float y){
this.yTarget = y;
}

void show(){
move(0);
isOpen = true;
}

void hide(){
move(-55);
isOpen = false;
}

void update(){
y += (yTarget - y) * 0.2;
image(img,0,y);
drawButtons();
}

```

```

}

void showOpenMenu(){
    image(openMenu,0,0);
}

boolean mouseIsTouchingBack(){
    if(mouseX >= 0 && mouseX <= 50 && mouseY >=0 && mouseY <=55){
        return true;
    }
    return false;
}

boolean mouseIsTouchingSpeech(){
    if(mouseX >= 87 && mouseX <= 232 && mouseY >=0 && mouseY <=55){
        return true;
    }
    return false;
}

void onMousePressed(){
    if(mouseIsTouchingBack()){
        if(topMenu.statusMenu[0] == 1){
            topMenu.statusMenu[0] = 2;
            topMenu.setButtons(2,1,1);
        }
    }

    if(mouseIsTouchingSpeech()){
        work=!work;
    }

}

void setButtons(int a, int b, int c){
    this.statusMenu[0] = a;
    this.statusMenu[1] = b;
    this.statusMenu[2] = c;
}

void drawButtons(){

    int x_back = 0;
    int x_speech = 0;
    int x_close = 265;

    if(topMenu.isOpen){
        stroke(130);
        line(87, y+23, 232, y+23);
    }
    if(work){
        showBubble2(y);
    }

    switch(statusMenu[0]) {
        case 0:
            image(back0,x_back,y);
            break;
        case 1:
            image(back1,x_back,y);
            break;
        case 2:
            image(back2,x_back,y);
            break;
    }
    switch(statusMenu[1]) {

```

```

    case 1:
        image(speech1,x_speech,y);
        break;
    }
    switch(statusMenu[2]) {
        case 1:
            image(close1,x_close,y);
            break;
        }
    }
}
}

```

```

// *****
// Load images for the animation
// *****

```

```

String[] listFileNames(String dir) {
    File file = new File(dir);
    if (file.isDirectory()) {
        String names[] = file.list();
        return names;
    } else {
        // If it's not a directory
        return null;
    }
}

```

```

// *****
// Draw
// *****

```

```

void draw(){

    background(255);
    numB = !numB;
    aiuto = !aiuto;

    if(page==PAGE_TITLE){
        image(logo,0,0);
    }

    if(page == PAGE_START_BUBBLE){

        background(255);
        if(advise==1){
            image(foto[0],0,0);
            drawFloatingBubble(false);
        }
        else if(advise==2){

            if(n < 176){
                image(foto[n],0,0);
                n++;

                // animation speed management
                if(n>0 && n<20){ delay(40); }
                if(n>20 && n<30){ delay(30); }
                if(n>=30){ delay(15); }
                if(n>150 && n<=160){ delay(20); }
                if(n>160 && n<=162){ delay(20); }
                if(n>162 && n<176){ delay(30); }
            }
        }
    }
}

```

```

else if(n==num_foto){
    image(foto[n-1],0,0);
}

drawFloatingBubble(false);

}

}

if(page == PAGE_ALERT){
    background(255);
    // draw alert page
    image(alert1 map,0,0);
    if(topMenu.isOpen){
        image(menuAlert, 0, 50);
    }

    if(aiuto){
        image(aiuto1, 0, 0);
        delay(500);
    }
    else{
        image(aiuto2, 0, 0);
        delay(500);
    }

}

}

if(page == PAGE_MAIN){
    background(255);
    println(SEND_BUBBLE);
    if(SEND_BUBBLE){
        //image(page1, 0, 0);
        drawFloatingBubble(false);
    }
    else{
        //image(page1_politics, 0, 0);
        drawFloatingBubble(true);
    }

}

}

if(page == PAGE_BUBBLE_BIG){
    background(255);
    image(page2, 0, 0);
}
if(page == PAGE_PHRASE){
    background(255);
    image(page3, 0, 0);
}

if(page == PAGE_MAP){
    background(255);
    image(page_map, 0, 0);
}

if(page >= PAGE_MAIN){
    topMenu.update();
    bottomMenu.update();
    if(!topMenu.isOpen){
        topMenu.showOpenMenu();
    }
    /*
    if(topMenu.isOpen){
        stroke(130);

```

```

    line(87, 23, 232, 23);
  }
  */
}

// speech recognizer

}

// *****
// Floating bubble animation
// *****

PImage be; //bolla ethics
PImage bm; //bolla music
PImage bph; //bolla physics
PImage bpo; //bolla politics

float bubbleXbe;
float bubbleOffsetXbe;
float bubbleRangeXbe;

float bubbleXbm;
float bubbleOffsetXbm;
float bubbleRangeXbm;

float bubbleXbph;
float bubbleOffsetXbph;
float bubbleRangeXbph;

float bubbleXbpo;
float bubbleOffsetXbpo;
float bubbleRangeXbpo;

float bubbleProgress;

void setupFloatingBubble(){
  size(320, 480);
  bubbleOffsetXbe = 52;
  bubbleRangeXbe = 10;
  bubbleOffsetXbm = 185;
  bubbleRangeXbm = 5;
  bubbleOffsetXbph = 43;
  bubbleRangeXbph = 4;
  bubbleOffsetXbpo = 95;
  bubbleRangeXbpo = 16;
  bubbleProgress = 0;
  be=loadImage("SCREAN_ STOP MOTION-19.png");
  bm=loadImage("SCREAN_ STOP MOTION-20.png");
  bph=loadImage("SCREAN_ STOP MOTION-18.png");
  bpo=loadImage("SCREAN_ STOP MOTION-21.png");
}

void drawFloatingBubble(boolean politics){

  bubbleXbe = bubbleOffsetXbe + (sin(bubbleProgress) * bubbleRangeXbe);
  bubbleXbm = bubbleOffsetXbm + (cos(bubbleProgress) * bubbleRangeXbm);
  bubbleXbph = bubbleOffsetXbph + (sin(bubbleProgress) * bubbleRangeXbph);
  bubbleXbpo = bubbleOffsetXbpo + (cos(bubbleProgress) * bubbleRangeXbpo);
  bubbleProgress = bubbleProgress + .01;
  if(bubbleProgress >= 2 * PI){
    bubbleProgress = 0;
  }
  // ellipse(bubbleX, 100, 50, 50);
  image(be,bubbleXbe,51); //original position 52,51
  image(bm,bubbleXbm,160);
  image(bph,bubbleXbph,182);
  if(politics){
    image(bpo,bubbleXbpo,214);
  }
}
}

```

```

// *****
// Speech recognizer animation
// *****

float[] r;
int Max;//max ellipse size
int Min;//max ellipse size
int t; // array lenght
float[] xPosition;
boolean work;

void setupSpeech(){
  smooth();
  Max=20;
  Min=5;
  t=15;
  r = new float[t];
  xPosition= new float[t];
  xPosition[0]=87;

  for(int count=0; count<t; count++){
    r[count]=random(Min,Max);
    if(count>0){
      xPosition[count]= (xPosition[count-1])+r[count-1]/2)+(r[count]/2);
    }
  }
}

void showBubble(){
  for(int count=0; count<t; count=count+1){
    fill(130);
    noStroke();
    ellipse(xPosition[count], 23, r[count] , r[count] );
    xPosition[count] = xPosition[count] + 0.8;
    if(xPosition[count] > 232){
      xPosition[count] = 87;
    }
  }
}

void showBubble2(float y){
  for(int count=0; count<t; count=count+1){
    fill(130);
    noStroke();
    ellipse(xPosition[count], y+23, r[count] , r[count] );
    xPosition[count] = xPosition[count] + 0.8;
    if(xPosition[count] > 232){
      xPosition[count] = 87;
    }
  }
}

```