```
/*
TidingsReceiver
IUAV IxD Lab1 2010/2011
Philip Tabor with Gillian Crampton Smith
*/
////////////////////////// TAB tidingsReceiver /////////////////////////
// Wave coordinates
float y1= 200;
float y2= 480;
float curveX = 300;
float a;
float movimentoC;

PImage home;
PImage wave;
PImage tag;
PImage received;
PImage map;
PImage bottle_map;

ArrayList ripples;

int timer2 =0;

// Screen number
int screenMode = 0;

void setup() {
  size(320, 480);
  ripples = new ArrayList();
  loadBackground();

  wave= loadImage(“onda.png”);
  tag= loadImage(“tag.png”);
  received= loadImage(“messaggereceived.png”);
  map = loadImage(“mappa.png”);
  bottle_map = loadImage(“bottiglia_mappa.png”);
}


void draw() {
  //// Home navigation ////
  if(screenMode == 0) {
    updateBackground();
    drawBackground();

    // Tag- Rising wave (when you receive a message) //
    if (y1<480) {
      a = a+0.05;
      movimentoC = 100*sin(a/1.2);
      image(wave,(-1*curveX)+movimentoC,y1,1449/1.8,917/1.8);
      y1= y1+5;
    }
```

```
else {
      a = a+0.05;
      movimentoC = 100*sin(a/1.2);
      image(wave,(-1*curveX)+movimentoC,y1,1449/1.8,917/1.8);
      if (y2>200) {
        a = a+0.05;
        movimentoC = 100*sin(a/1.2);
        image(tag,(-1*curveX)+movimentoC,y2,1073/1.2,698/1.2);
        y2= y2-5;
      }
      else {
        a = a+0.05;
        movimentoC = 100*sin(a/1.2);
        image(tag,(-1*curveX)+movimentoC,y2,1073/1.2,698/1.2);
      }
    }
  }
  //// Received message ////
  else if(screenMode == 1) {
    image(received, 0, 0);
  }
  //// Home ////
  else if(screenMode == 2) {
    updateBackground();
    drawBackground();
    a = a+0.05;
    movimentoC = 100*sin(a/1.2);
    image(wave,(-1*curveX)+movimentoC,200,1449/1.8,917/1.8);
  }
  //// Map ////
  else if(screenMode == 3) {
    drawMap();
  }
}

//////// Screen change ////////
void mousePressed() {
  /// Received message ///
  if(screenMode == 0) {
    // See the message
    if(mouseX > 0 && mouseX < width && mouseY > 185 && mouseY < 330) {
      screenMode = 1;
    }
    mousePressedBackground();
  }
  /// See the message ///
  else if(screenMode == 1) {
    // Go back
    if(mouseX > 20 && mouseY > 20 && mouseX < 80 && mouseY < 80) {
      screenMode = 2;
    }
    // Map
    if(mouseX > 220 && mouseY > 220 && mouseX < 260 && mouseY < 260) {
      screenMode = 3;
    }
  }
  /// Map ///
  if(screenMode == 3) {
    // Go back
    if(mouseX > 20 && mouseY > 20 && mouseX < 80 && mouseY < 80) {
      screenMode = 1;
    }
  }
}

////////////////////////// TAB background /////////////////////////
/// Position & Animation Variables
float friction = 0.95;
float x_velocity = -1;
float x_position = 640;
float x_offset = -1;
float x_refresh = 640;
float y=0;
float percentage = 0;

/// Background
void loadBackground() {
  home = loadImage (“menu.jpg”);
}

void updateBackground() {
  // Horizontal dragging
  if(mousePressed) {
    x_velocity = (mouseX + x_offset) - x_position;
    x_position = mouseX + x_offset;
  }
  else {
    x_velocity = x_velocity * friction;
    x_position = x_position + x_velocity;
  }

  ///// stoppig edges
  if (x_position > 0) {
    x_position = 0;
  }
  if (x_position < (-320)) {
    x_position = -320;
  }
}
```

```
void drawBackground() {
  image (home, x_position, y);
}


void movetocenter() {
  if(x_position != -320) {
    x_velocity = (x_refresh - x_position) * 0.054;
  }
}


void mousePressedBackground() {
  if((mouseX > x_position - 1600)&&(mouseX <  x_position + 1600)) {
    x_offset = x_position - mouseX;
  }
}


/////////////////////////////// TAB map ////////////////////////////////
void drawMap() {
  image(map, 0, 0, 320, 480);
  image(bottle_map, 215, 230, 41, 84);

  for(int i = ripples.size()-1; i >= 0;i--) {
    image(bottle_map, 215, 230, 41, 84);
    Ripple ripple = (Ripple) ripples.get(i);
    ripple.resize();
    ripple.display();
    if(!ripple.on)ripples.remove(i);
  }
  timer2++;
  if(timer2%50 == 0) {
    int which = round(random(1)); // pick a color, any color (or the
number associateing to that color in the array)
    ripples.add(new Ripple(230, 305, which));
  }
}
```

```
///////////////////////////// TAB ripples //////////////////////////////
// Color array
int c1 = color (255,255,255);
int c2 = color (255,255,255);
int c3 = color (255,255,255);

color[] c = {
  c1, c2, c3
};

class Ripple { // The ripple class
  float x, y;
  float diameter, alph;
  boolean on;
  boolean finished = false;
  color colors;

  public Ripple(float xpos, float ypos, color shade) {
    x = xpos;
    y = ypos;
    on = true;
    diameter = 20;
    colors = shade;
  }

  void resize() {
    if (on) {
      if (diameter < 200) {
        diameter++;
        alph = abs((diameter*1.275)-255);
      }
      else {
        on = false;
      }
    }
  }

  void display() {
    noFill();
    stroke(c[colors], alph);
    strokeWeight(3);
    ellipse(x, y, diameter, diameter);
  }
}
```