

```
/* IUAV design dell'interazione 07-08
```

```
Sardiniamo
```

```
An application that allow the user to locate his contacts in venice map and send them messages and invites to meet.
```

```
*/
```

```
/******
```

```
*****
```

```
* Graphics Section
```

```
* This section of the program is for the graphics.
```

```
* You can think of it as the section for code that provides feedback to the user (behaviors)
```

```
*****
```

```
*****/
```

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

```
// Names Of Graphics & How Load Them
```

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

```
// named references to graphics
```

```
PImage sardinamodeMenu;
```

```
PImage messagesMenu;
```

```
PImage contactsMenu;
```

```
PImage settingsMenu;
```

```
PImage sardinaModeMessage;
```

```
PImage map;
```

```
PImage mapModeBar;
```

```
PImage sardinaModeBar;
```

```
PImage grey;
```

```
PImage arrow;
```

```
PImage messagesModeBar;
```

```
PImage contactsModeBar;
```

```
PImage settingsModeBar;
```

```
PImage pallino;
```

```
PImage sardinaVERDE;
```

```
PImage Lelisboa;
```

```
PImage CarlaGinella;
```

```
PImage Antonella;
```

```
PImage Abdul;
```

```
PImage barra_arancia;
```

```
PImage button_check;
```

```
PImage button_uncheck;
```

```
PImage box_message;
```

```
PImage box_contacts;
```

```
PFont fontWhite;
```

```
PFont fontBlack;
```

```
// function to load all the images for the interface
```

```

void loadImage()
{
    sardinamodeMenu = loadImage("sardinamodeMenu.png");
    sardinaModeBar = loadImage ("sardinaModeBar.png");
    messagesMenu = loadImage("messagesMenu.png");
    contactsMenu = loadImage("contactsMenu.png");
    settingsMenu = loadImage("settingsMenu.png");

    map = loadImage("map.png");
    mapModeBar = loadImage ("mapModeBar.png");
    grey = loadImage ("grey.png");
    arrow = loadImage ("arrow.png");
    messagesModeBar = loadImage("messagesModeBar.png");
    contactsModeBar = loadImage("contactsModeBar.png");
    settingsModeBar = loadImage("settingsModeBar.png");

    pallino = loadImage("pallino.png");
    sardinaVERDE = loadImage("sardinaVERDE.png");

    Lelisboa = loadImage("Lelisboa.png");
    Antonella = loadImage("Anto.nella.png");
    Abdul = loadImage("Abdul.png");
    CarlaGinella = loadImage("CarlaGinella.png");

    barra_arancia = loadImage("barra_arancia.png");
    button_check = loadImage("button_check.png");
    button_uncheck = loadImage("button_uncheck.png");
    box_message = loadImage("box_message.png");
    box_contacts = loadImage("box_contacts.png");
}

////////////////////////////////////
// Screen Drawing Functions
////////////////////////////////////

// function to draw the menu with specified focus
void drawMenu(int focus)
{
    if(focus == MENUFOCUS_SARDINAMODE){
        image(sardinamodeMenu, 0, 262);
        image(sardinaModeBar,0,0);
    }
    else if(focus == MENUFOCUS_MESSAGES){
        image(messagesModeBar,0,0);
        image(messagesMenu, 0, 262);
    }
    else if(focus == MENUFOCUS_CONTACTS){
        image(contactsModeBar,0,0);
        image(contactsMenu, 0, 262);
    }
    else if(focus == MENUFOCUS_SETTINGS){

```

```

    image(settingsModeBar,0,0);
    image(settingsMenu, 0, 262);
}
if(drawnOnce == false){
    drawnOnce = true;
    image(grey,0,22);
}
}

```

```
boolean drawnOnce = false;
```

```

void beforeMenu(){
    modeBeforeMenu = mode;
    drawnOnce = false;
}

```

```

void drawMessages(int focus, String message, int contactsFocus)
{
    // draw background
    if(focus == MESSAGES_WRITE){
        image(box_message, 0, 0);
        image(sardinaModeBar,0,0);
    }
    else if(focus == MESSAGES_CONTACTS){
        image(box_contacts, 0, 0);
        image(sardinaModeBar,0,0);
    }

    // draw our message
    textFont(fontBlack);
    text(message, 20, 60);

    // if we are writing a message...
    if(focus == MESSAGES_WRITE)
    {
        // draw everyone unselected...
        drawContact("Abdul", 130, checkBoxValues[0]);
        drawContact("andrea", 145, checkBoxValues[1]);
        drawContact("Anto.nella", 160, checkBoxValues[2]);
        drawContact("CCCCCAAAAA", 175, checkBoxValues[3]);
        drawContact("Carla Ginella", 190, checkBoxValues[4]);
        drawContact("Diego", 205, checkBoxValues[5]);
        drawContact("enrico de napoli", 220, checkBoxValues[6]);
        drawContact("Federicaaaaa", 235, checkBoxValues[7]);
        drawContact("Guillermo", 250, checkBoxValues[8]);
        drawContact("Lelisboa", 265, checkBoxValues[9]);
        drawContact("marina", 280, checkBoxValues[10]);
    }
    // if we are choosing contacts...
    else if(focus == MESSAGES_CONTACTS){

```

```

// if abdul is selected...
if(0 == contactsFocus){
    // draw abdul selected
    drawSelectedContact("Abdul", 130, checkBoxValues[0]);
}
else{
    // draw abdul unselected
    drawContact("Abdul", 130, checkBoxValues[0]);
}

// if andrea is selected...
if(1 == contactsFocus){
    // draw andrea selected
    drawSelectedContact("andrea", 145, checkBoxValues[1]);
}
else{
    drawContact("andrea", 145, checkBoxValues[1]);
}

// if antonella is selected...
if(2 == contactsFocus){
    // draw Anto.nella selected
    drawSelectedContact("Anto.nella", 160, checkBoxValues[2]);
}
else{
    drawContact("Anto.nella", 160, checkBoxValues[2]);
}

// if CCCCCAAAAA is selected...
if(3 == contactsFocus){
    // draw CCCCCAAAAA selected
    drawSelectedContact("CCCCCAAAAA", 175, checkBoxValues[3]);
}
else{
    drawContact("CCCCCAAAAA", 175, checkBoxValues[3]);
}

// if Carla Ginella is selected...
if(4 == contactsFocus){
    // draw Carla Ginella selected
    drawSelectedContact("Carla Ginella", 190, checkBoxValues[4]);
}
else{
    drawContact("Carla Ginella", 190, checkBoxValues[4]);
}

// if Diego is selected...
if(5 == contactsFocus){
    // draw Diego selected
    drawSelectedContact("Diego", 205, checkBoxValues[5]);
}
}

```

```

else{
    drawContact("Diego", 205, checkBoxValues[5]);
}

// if enrico de napoli is selected...
if(6 == contactsFocus){
    // draw enrico de napoli selected
    drawSelectedContact("enrico de napoli", 220, checkBoxValues[6]);
}
else{
    drawContact("enrico de napoli", 220, checkBoxValues[6]);
}

// if Federicaaaaaa is selected...
if(7 == contactsFocus){
    // draw Federicaaaaaa selected
    drawSelectedContact("Federicaaaaaa", 235, checkBoxValues[7]);
}
else{
    drawContact("Federicaaaaaa", 235, checkBoxValues[7]);
}

// if Guilermo is selected...
if(8 == contactsFocus){
    // draw Guilermo selected
    drawSelectedContact("Guilermo", 250, checkBoxValues[8]);
}
else{
    drawContact("Guilermo", 250, checkBoxValues[8]);
}

// if Lelisboa is selected...
if(9 == contactsFocus){
    // draw Lelisboa selected
    drawSelectedContact("Lelisboa", 265, checkBoxValues[9]);
}
else{
    drawContact("Lelisboa", 265, checkBoxValues[9]);
}

// if marina is selected...
if(10 == contactsFocus){
    // draw marina selected
    drawSelectedContact("marina", 280, checkBoxValues[10]);
}
else{
    drawContact("marina", 280, checkBoxValues[10]);
}
}
}
}

```

```

void drawSelectedContact(String name, int yPosition, boolean checked)

```

```

{
image(barra_arancia, 15, yPosition);
if(checked){
image(button_check, 20, yPosition +1);
}
else{
image(button_uncheck, 20, yPosition +1);
}
textFont(fontWhite);
text(name, 40, yPosition + 12);
}

```

```

void drawContact(String name, int yPosition, boolean checked)

```

```

{
if(checked){
image(button_check, 20, yPosition +1);
}
else{
image(button_uncheck, 20, yPosition +1);
}
textFont(fontBlack);
text(name, 40, yPosition + 12);
}

```

```

void drawMap()

```

```

{
// this is true as long as a key is still held down.
if(keyPressed)
{
// Make sure this is within the main draw loop to get continous pan on keypress.
// Otherwise, you would need to press the keys over and over to move.

if(keyCode == DOWN){
mapY=mapY-pan_speed;           // Pan up
}
else if (keyCode == UP){
mapY=mapY+pan_speed;           // Pan down
}
else if (keyCode == RIGHT){
mapX=mapX-pan_speed;           // Pan left
}
else if (keyCode == LEFT) {
mapX=mapX+pan_speed;           // Pan right
}

mapX = constrain(mapX,(width-map.width),0); // Prevent image from panning beyond width
mapY = constrain(mapY,(height-map.height),0); // Prevent image from panning beyond height
}

image(map, mapX, mapY);           // Display image
image(mapModeBar,0,0);

```

```

    image(arrow,109,297);
}

//parte anna pallini
void drawPallino()
{
    image(pallino, mapX + 100, mapY + 600);
    image(pallino, mapX + 170, mapY + 700);
    image(pallino, mapX + 190, mapY + 760);
    image(pallino, mapX + 320, mapY + 770);
    image(pallino, mapX + 350, mapY + 785);
    image(pallino, mapX + 280, mapY + 720);
    image(pallino, mapX + 350, mapY + 620);
    image(sardinaVERDE, mapX + 330, mapY + 730);

}

void drawNames ()
{
    image(Lelisboa,mapX +329, mapY +764);
    image(Abdul,mapX +363, mapY +783);
    image(CarlaGinella,mapX +293, mapY +719);
    image(Antonella,mapX +363, mapY +618);

}

////////////////////////////////////
// Setup, Executes Once When Started, Prepares Program to Run (Logic Initialization)
////////////////////////////////////

import processing.phone.*; // import phone library to go fullscreen
Phone myPhone;           // named reference to phone instance

void setup() // happens only once, when the program starts...
{
    checkBoxValues = new boolean[11];
    checkBoxValues[0] = false;
    checkBoxValues[1] = false;
    checkBoxValues[2] = false;
    checkBoxValues[3] = false;
    checkBoxValues[4] = false;
    checkBoxValues[5] = false;
    checkBoxValues[6] = false;
    checkBoxValues[7] = false;
    checkBoxValues[8] = false;
    checkBoxValues[9] = false;
    checkBoxValues[10] = false;

    loadImages();
    fontWhite = loadFont("Helvetica-Bold-12.mv1w", color(255, 255, 255));
    fontBlack = loadFont("Helvetica-Bold-12.mv1w", color(0, 0, 0));
}

```

```

// go fullscreen
myPhone = new Phone(this); // create new phone instance/controller
myPhone.fullscreen(); // tell phone to go fullscreen
}

////////////////////////////////////
// Mode, Focus & Option Names (Constants)
////////////////////////////////////
// names for each possible mode
// tip: one mode for each screen is easy to program for a demo but less easy to make into a real
application
// one mode per logical group/flow of screens (e.g. registration, menu) is difficult to program but
easier to make real

int MODE_MENU = 0;
int MODE_SARDINAMODE = 1;
int MODE_MAP = 2;
// names for each possible focus option of the main menu
int MENUFOCUS_SARDINAMODE = 0;
int MENUFOCUS_MESSAGES = 1;
int MENUFOCUS_CONTACTS = 2;
int MENUFOCUS_SETTINGS = 3;

int MESSAGES_WRITE = 0;
int MESSAGES_CONTACTS = 1;

////////////////////////////////////
// State, Information Collected From Use (Variables)
////////////////////////////////////
// main menu information
int mode = MODE_MAP; // initially, we are showing the main menu
// main menu information
int menuFocus = MENUFOCUS_SARDINAMODE; // initially, the main menu focus is on the first
item "sardinamode"
int modeBeforeMenu = MODE_MAP;

int mapX = -240;
int mapY = -585;

int pan_speed = 7;

int modeMessages = MESSAGES_WRITE;
String message = "Sardiniamo?";
int contactsFocus = 0;
boolean[] checkBoxValues;

int menuCountDown = 4;

////////////////////////////////////
// Draw, Executes Forever, Provides User Feedback (Logic Repetition)
////////////////////////////////////

```

```

void draw() // happens repeatedly (according to framerate)...
{
  if(mode == MODE_MENU){ // if we are in "Menu" mode...
    drawMenu(menuFocus); // draw image for menu with a variable focus
  }
  else if(mode == MODE_MAP){ // if we are in "SARDINA MODE" mode...
    drawMap(); // draw map
    drawPallino();
    drawNames();

  }
  else if(mode == MODE_SARDINAMODE){ // if we are in "SARDINA MODE" mode...
    //drawsardinaModeMessage(); // draw image for the sestiere selector
    drawMessages(modeMessages, message, contactsFocus);
  }
  if(menuCountDown > 0){
    menuCountDown--;
    if(menuCountDown == 1){
      mode = MODE_MENU;
    }
  }
}

////////////////////////////////////
// Keypad Event (User Input Capture & Interpretation)
////////////////////////////////////

void keyReleased() // whenever a key is pressed...
{
  if(mode == MODE_MENU){
    // go back to map mode
    if(key == '0'){ // if the user pressed 0...
      mode = modeBeforeMenu; // put the menu focus on "mapmode"
    }
    // if focused on sardina
    if(menuFocus == MENUFOCUS_SARDINAMODE) // if the menu is focused on
"sardinamode"...
    {
      if(keyCode == RIGHT){ // if the user pressed down...
        menuFocus = MENUFOCUS_MESSAGES; // put the menu focus on "messages"
      }
      else if(keyCode == LEFT){ // if the user pressed right...
        menuFocus = MENUFOCUS_SETTINGS; // put the menu focus on "settings"
      }
      else if(keyCode == FIRE){ // if the user pressed fire/center...
        mode = MODE_SARDINAMODE; // put the menu focus on "sardinamode"
      }
    }
  }
  else if(menuFocus == MENUFOCUS_MESSAGES) // if the menu is focused on "messages"...
  {
    if(keyCode == RIGHT){ // if the user pressed up...

```

```

    menuFocus = MENUFOCUS_CONTACTS;    // put the menu focus on "contacts"
}
else if(keyCode == LEFT){             // if the user pressed right...
    menuFocus = MENUFOCUS_SARDINAMODE; // put the menu focus on "sardinamode"
}
}
else if(menuFocus == MENUFOCUS_CONTACTS) // if the menu is focused on "contacts"...
{
    if(keyCode == RIGHT){              // if the user pressed up...
        menuFocus = MENUFOCUS_SETTINGS; // put the menu focus on "settings"
    }
    else if(keyCode == LEFT){          // if the user pressed right...
        menuFocus = MENUFOCUS_MESSAGES; // put the menu focus on "messages"
    }
}
else if(menuFocus == MENUFOCUS_SETTINGS) // if the menu is focused on "settings"...
{
    if(keyCode == RIGHT){              // if the user pressed up...
        menuFocus = MENUFOCUS_SARDINAMODE; // put the menu focus on
"sardinamode"
    }
    else if(keyCode == LEFT){          // if the user pressed right...
        menuFocus = MENUFOCUS_CONTACTS; // put the menu focus on "contacts"
    }
}
}
else if(mode == MODE_MAP){
    if(key == '0'){ // if the user pressed 0...
        beforeMenu();
        mode = MODE_MENU; // put the menu focus on "menumode"
    }
    if(keyCode == RIGHT){              // if the user pressed up...
        mapX = mapX - pan_speed ;     // put the menu focus on "sardinamode"
    }
    else if(keyCode == LEFT){          // if the user pressed right...
        mapX = mapX + pan_speed ;     // put the menu focus on "contacts"
    }
    else if(keyCode == UP){            // if the user pressed right...
        mapY = mapY + pan_speed ;     // put the menu focus on "contacts"
    }
    else if(keyCode == DOWN){          // if the user pressed right...
        mapY = mapY - pan_speed ;     // put the menu focus on "contacts"
    }
}
}
else if(mode == MODE_SARDINAMODE) // if we are at messages
{
    if(key == '0'){ // if the user pressed 0...
        beforeMenu();
        mode = MODE_MENU; // put the menu focus on "menumode"
    }
    if(keyCode == SOFTKEY2){ // if the user pressed "back"...

```

