

# VENICELIVE | MOBILE PROCESSING CODE

```
/******  
  
VENICE LIVE  
Variables Initialization  
*****/  
  
////////////////////////////////////  
// Libraries - Additional libraries that your program requires  
////////////////////////////////////  
  
import processing.phone.*;  
import processing.video.*;  
import processing.sound.*;  
  
Phone myPhone; // Named reference to your phone  
Sound s;  
  
// using camera to take a picture of the user  
Capture cap;  
PImage pic;  
boolean showingCamera = false;  
boolean pictureTaken = false;  
  
// State, Describes the current state of the program (Logic State)  
  
int currentScreen = 0;  
int totalScreen = 9;  
PImage[] screens;  
PImage nickname_over;  
PImage photo_over;  
PImage no_key1;  
PImage no_key2;  
PImage border_photo;  
PImage calendar;
```

```
UIImage day_arrow_left;
UIImage day_arrow_left_ok;
UIImage day_arrow_right;
UIImage day_arrow_right_ok;
UIImage event_arrow_left;
UIImage event_arrow_left_ok;
UIImage event_arrow_right;
UIImage event_arrow_right_ok;
```

```
boolean go_day_sx;
boolean go_day_dx;
boolean go_event_sx;
boolean go_event_dx;
UIImage[] day_selected;
int n_scroll_days;
int currentDay;
UIFont font;
UIFont font_name_artist;
UIFont font_details_event;
UIFont font_ask;
UIFont font_code;
```

```
UIImage border_cover;
UIImage current_cover;
int currentEvent;
boolean events;
UIImage no_event;
```

```
int finalDay;
int finalEvent;
```

```
// images for the genre
```

```
UIImage[] genre_img;
UIImage[] genre_img_selected;
```

```
// menu status
int MENU_OPTION_NAME = 1;
int MENU_OPTION_PICTURE = 2;
int MENU_OPTION_GENRE = 3;
int menuFocus;

int menuFocusCalendar;
int focusDay = 1;
int focusEvent = 2;

// manage genre of music i'd like
String direction_genre="";
boolean moving_genre;

// nickname, public variable
String nickname_ok="";
int genreStatus;
boolean view_select_screen1;
String[][][] live;
int oneday;
int event;
int details;

boolean modeInfo;
boolean modePlayer;
PImage str_place;
PImage str_ticketprice;
PImage star;
PImage star_ok;
PImage bg_info;
PImage bg_player;
int currentStars;
int x_star;
String ask1;

PImage bg_ticket;
```

```
PIImage bg_reservation;
PIImage less;
PIImage less_ok;
PIImage more;
PIImage more_ok;
int ticketNumber;
String numTicketStr;
String totalPrice;
PIImage buy;
PIImage book;
boolean arrow_more;
boolean arrow_less;
String code;
boolean soundPlaying = false;
PIImage coverBig;
boolean zoom;

void showCamera()
{
    showingCamera = true;
    cap.show(50, 90, 139, 136);
    noLoop();
}

void hideCamera()
{
    showingCamera = false;
    pictureTaken = true;
    byte[] data = cap.read();
    pic = loadImage(data);
    cap.hide();
    myPhone.vibrate(500);
    loop();
}
```

```

// *****
// Redraw "image" function
// *****

void image(PImage sourceImage, int xPos, int yPos, int newWidth, int newHeight) {
    int oldWidth = sourceImage.width;
    int oldHeight = sourceImage.height;

    PImage2 inputData = new PImage2(sourceImage);
    PImage2 outputData = new PImage2(newWidth, newHeight);

    int YD = (oldHeight / newHeight - 1) * oldWidth;
    int YR = oldHeight % newHeight;
    int XD = oldWidth / newWidth;
    int XR = oldWidth % newWidth;
    int outOffset = 0;
    int inOffset = 0;

    for (int y = newHeight, YE = 0; y > 0; y--) {
        for (int x = newWidth, XE = 0; x > 0; x--) {
            outputData.pixels[outOffset++] = inputData.pixels[inOffset];
            inOffset += XD;
            XE += XR;
            if (XE >= newWidth) {
                XE -= newWidth;
                inOffset++;
            }
        }
        inOffset += YD;
        YE += YR;
        if (YE >= newHeight) {
            YE -= newHeight;
            inOffset += oldWidth;
        }
    }
}

```

```

    image(outputData, xPos, yPos);
}
/*****
* Graphics Section - Code that provides feedback to the user (behaviors)
*****/

import processing.image2.*;

// ****
// Draw - Executes forever, provides user feedback (Logic Repetition)
// ****

void draw()
{
    drawScreen(currentScreen);

    String nickname = "";

    if(currentScreen==1){
        textFont(font);
        if(menuFocus == MENU_OPTION_NAME) {
            image(nickname_over,14,45);

            multitap(); // enable multitap

            for(int i = 0; i < multitapBufferIndex; i++){
                nickname += multitapBuffer[i];
            }

            text(nickname, 22, 65);

            nickname_ok = nickname;

            if(pictureTaken){
                image(pic, 50, 90, 139, 136);
            }
        }
    }

    if(menuFocus == MENU_OPTION_PICTURE){
        text(nickname_ok, 22, 65); // quando arrivo qui il nickname deve rimanere dov'è !! ma con sfondo scuro !

        if(pictureTaken){
            image(border_photo,48,88);

            image(pic, 50, 90, 139, 136);
        }
    }
}

```

```

}
else{
    image(border_photo,48,88);
    image(photo_over,50,88);
}
}

// direction_genre
genreDraw();

if(menuFocus == MENU_OPTION_GENRE){
    text(nickname_ok, 22, 65);
    if(pictureTaken){
        image(pic, 50, 90, 139, 136);
    }
    genreDrawSelected();
}

if(view_select_screen1 == false){
    image(no_key2, 176, 297);
}
if(nickname_ok!=""){
    view_select_screen1 = true;
}

} // end currentScreen == 1

if(currentScreen==2){

if(menuFocusCalendar == focusDay){
    // qui scorro i giorni della settimana
    image(calendar,27-((currentDay)*62),31);
    image(day_arrow_left,0,31);
    image(day_arrow_right,width-27,31);
    if(go_day_sx){

```

```

    image(day_arrow_left_ok,0,31);
}
else if(go_day_dx){
    image(day_arrow_right_ok,width-27,31);
}

// carico l'img per il giorno selezionato
image(day_selected[currentDay],0,31);

// prima cover del giorno sempre visibile, se c'è almeno un evento per quel giorno
// altrimenti mostra un'immagine di default che fa capire che per quel giorno non ci sono eventi in programma
if(live[currentDay][1][0]!=""){
    events = true;
    current_cover = loadImage(live[currentDay][1][1]);
    textAlign(CENTER);
    textFont(font_name_artist);
    text(live[currentDay][1][0],10,100,220,20);
    image(current_cover,70,130);
    text("H "+live[currentDay][1][3],10,240,220,20);
    textFont(font_ask);
    text("Participants: "+live[currentDay][1][10],10,270,220,20);
}
else{
    events = false;
    image(no_event, 70,130);
}
}

if(menuFocusCalendar == focusEvent){
    // scrollo gli eventi disponibili in quel giorno

    // ridisegno il calendario senza focus
    image(calendar,27-((currentDay)*62),31);
    image(day_arrow_left_ok,0,31);
    image(day_arrow_right_ok,width-27,31);
    image(day_selected[currentDay],0,31);
}

```



```
// ciclo
current_cover = loadImage(live[currentDay][currentEvent][1]);
textFont(font_name_artist);
textAlign(CENTER);

//text(live[currentDay][currentEvent][0],10,70,220,20);
image(border_cover,68,128); // focus nell'evento

if(go_event_sx){ // evidenzio il pulsante sinistro di movimento
  image(event_arrow_left_ok, 28, 167);
}else{
  image(event_arrow_left, 28, 167);
}

if(go_event_dx){ // evidenzio il pulsante destro di movimento
  image(event_arrow_right_ok, 198, 167);
}else{
  image(event_arrow_right, 198, 167);
}

// current cover deve cambiare ...
image(current_cover,70,130);

textAlign(CENTER);
textFont(font_name_artist);
text(live[currentDay][currentEvent][0],10,100,220,20);
text("H "+live[currentDay][1][3],10,240,220,20);
textFont(font_ask);
text("Participants: "+live[currentDay][currentEvent][10],10,270,220,20);

if(zoom){
  // disegna lo zoom
  image(coverBig, 0, 0);
}
```

```
}
```

```
}
```

```
if(currentScreen == 3){  
    finalDay = currentDay;  
    finalEvent = currentEvent;  
    // disegno la testata  
    image(current_cover,10,10,74,73);  
    textFont(font_name_artist);  
    text(live[finalDay][finalEvent][0],98,26); // 0 = nome artista  
    textFont(font_details_event);  
    text(live[finalDay][finalEvent][11],98,42); // 11 = nome del giorno  
    text(live[finalDay][finalEvent][3],98,58); // 3 = ora  
  
    // modeInfo  
    if(modeInfo){  
        image(bg_info,6,60);  
  
        if((live[finalDay][finalEvent][0] == "the killers")){  
            s.stop();  
            soundPlaying = false;  
        }  
  
        // stelle  
        x_star = 174;  
        currentStars = Integer.parseInt(live[finalDay][finalEvent][6]);  
        for(int i=1;i<5;i++){  
            if(currentStars >= i){ image(star_ok,x_star,100); }  
            else{ image(star,x_star,100); }  
            x_star = x_star+12;  
        }  
        // place y = 150  
        textFont(font_name_artist);  
        textAlign(CENTER);  
        text(live[finalDay][finalEvent][4],10,145, 220, 20); // 4 = location name
```

```

text(live[finalDay][finalEvent][5],10,165, 220, 20); // 4 = location address
text(live[finalDay][finalEvent][7],10,220, 220, 20); // 7 = ticket price
textFont(font_ask);

// in base al costo scelgo la domanda da porre all'utente
/*
1. Euro 10
2. Euro 0 - Reservation required
3. Free entry
*/
if(live[finalDay][finalEvent][7] == "Euro 10"){
    ask1 = "Do you want to buy a ticket?";
}
else if(live[finalDay][finalEvent][7] == "Euro 0 - Reservation required"){
    ask1 = "Do you want a reservation?";
}
else if(live[finalDay][finalEvent][7] == "Free entry"){
    ask1 = ""; // no questions, the user goes immediately in the "pre party" screen
}

if(ask1!=""){ text(ask1,10,260,220,18); }

textAlign(LEFT);
}

// modePlayer
if(modePlayer){
    image(bg_player,6,60);
    textFont(font_name_artist);
    textAlign(CENTER);
    text(live[finalDay][finalEvent][12],10,145, 220, 20); // 4 = location name
    textAlign(LEFT);

    if(live[finalDay][finalEvent][0] == "the killers"){

        if(soundPlaying == false)

```

```

    {
        s.play();
    }

    if(s.duration() >= s.time())
        {soundPlaying = true;}
    else if(s.duration() == s.time())
        {soundPlaying = false; }
    }

}

}

if(currentScreen==4){
    // disegno la testata
    image(current_cover,10,10,74,73);
    textFont(font_name_artist);
    text(live[finalDay][finalEvent][0],98,26); // 0 = nome artista
    textFont(font_details_event);
    text(live[finalDay][finalEvent][11],98,42); // 11 = nome del giorno
    text(live[finalDay][finalEvent][3],98,58); // 3 = ora
    // in base al prezzo del biglietto (costo oppure ingresso gratuito ma con prenotazione) carico il resto della pagina
    if(live[finalDay][finalEvent][7] == "Euro 10"){
        image(bg_ticket,6,60);
        image(buy,177,295); // pulsante buy
    }
    if(live[finalDay][finalEvent][7] == "Euro 0 - Reservation required"){
        image(bg_reservation,6,60);
        image(book,177,295); // pulsante book
    }
    //disegno le frecce
    if(arrow_less){

```

```

    image(less_ok,54,171);
}else{
    image(less,54,171);
}
if(arrow_more){
    image(more_ok,174,171);
} else {
    image(more,174,171);
}
textFont(font_details_event);
numTicketStr = Integer.toString(ticketNumber);
text(numTicketStr, 115, 188);
textFont(font_name_artist);
text(Integer.toString(ticketNumber*10), 113, 267);

}

if(currentScreen == 5) {
    // disegno la testata
    image(current_cover,10,10,74,73);
    textFont(font_name_artist);
    text(live[finalDay][finalEvent][0],98,26); // 0 = nome artista
    textFont(font_details_event);
    text(live[finalDay][finalEvent][11],98,42); // 11 = nome del giorno
    text(live[finalDay][finalEvent][3],98,58); // 3 = ora
    textFont(font_code);
    text(code, 152, 136);

}

if(currentScreen == 6) {
    // disegno la testata
    image(current_cover,10,10,74,73);
    textFont(font_name_artist);
    text(live[finalDay][finalEvent][0],98,26); // 0 = nome artista
    textFont(font_details_event);
    text(live[finalDay][finalEvent][11],98,42); // 11 = nome del giorno

```

```

text(live[finalDay][finalEvent][3],98,58); // 3 = ora
// scrivo le info sul meeting point
textAlign(CENTER);
textFont(font_code);
text(live[finalDay][finalEvent][8],10,185, 220, 20); // 4 = location preparty name
text(live[finalDay][finalEvent][9],10,205, 220, 20); // 4 = location preparty address
textAlign(LEFT);
}

if(currentScreen == 7) {
textAlign(LEFT);
textFont(font_code);
text(live[finalDay][finalEvent][8],10,7, 220, 20); // 4 = location preparty name
text(live[finalDay][finalEvent][9],10,27, 220, 20); // 4 = location preparty address
}

} // end of draw()

// *****
// Function to draw screen
// *****
void drawScreen(int screenNumber)
{
image(screens[screenNumber], 0, 0);
}

// *****
// Other functions
// *****
void genreNext(){
genreStatus = genreStatus+1;
genreStatus = genreStatus % 6;
}

```

```
}
```

```
void genrePrev(){  
    genreStatus = genreStatus-1;  
    if(-1 == genreStatus){  
        genreStatus = 5;  
    }  
}
```

```
void genreDraw(){  
    image(genre_img[genreStatus],31,253);  
}
```

```
void genreDrawSelected(){  
    image(genre_img_selected[genreStatus],31,253);  
}
```

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

```
STORAGE EVENTS INFORMATIONS IN A TRI-DIMENSIONAL ARRAY
```

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

```
// LIVE[giorni][eventi per giorno][informazioni per un determinato evento]
```

```
// LIVE[][][] = new String[7][5][13];
```

```
void fillLive(){
```

```
    String a = "Euro 10";
```

```
    String b = "Euro 0 - Reservation required";
```

```
    String c = "Free entry";
```

```
//
```

```
*****  
*****
```

```
//
```

```
*****  
***** day 8
```

```
// day 8
```

```
// event 1
```

```
live[0][0][0] = "8"; // numer of the day
```

```
live[0][1][0] = ""; // name of the artist
```

```
live[0][1][1] = ""; // path of the event's cover
live[0][1][2] = ""; // path of the mp3
live[0][1][3] = ""; // time when event starts
live[0][1][4] = ""; // name of the location where the event happens
live[0][1][5] = ""; // address of the location
live[0][1][6] = ""; // reputation of the ocation
live[0][1][7] = ""; // price of the event
live[0][1][8] = ""; // name of the pre-party location
live[0][1][9] = ""; // address of the pre-party location
live[0][1][10] = ""; // number of partecipants confirmed for the event
live[0][1][11] = ""; // name of the day
live[0][1][12] = ""; // name of the song
```

```
// event 2
```

```
live[0][0][0] = "8";
live[0][2][0] = "";
live[0][2][1] = "";
live[0][2][2] = "";
live[0][2][3] = "";
live[0][2][4] = "";
live[0][2][5] = "";
live[0][2][6] = "";
live[0][2][7] = "";
live[0][2][8] = "";
live[0][2][9] = "";
live[0][2][10] = "";
live[0][2][11] = "";
live[0][2][12] = "";
```

```
// event 3
```

```
live[0][0][0] = "8";
live[0][3][0] = "";
live[0][3][1] = "";
live[0][3][2] = "";
live[0][3][3] = "";
live[0][3][4] = "";
```



```
live[0][3][5] = "";
live[0][3][6] = "";
live[0][3][7] = "";
live[0][3][8] = "";
live[0][3][9] = "";
live[0][3][10] = "";
live[0][3][11] = "";
live[0][3][12] = "";
```

```
// event 4
```

```
live[0][0][0] = "8";
live[0][4][0] = "";
live[0][4][1] = "";
live[0][4][2] = "";
live[0][4][3] = "";
live[0][4][4] = "";
live[0][4][5] = "";
live[0][4][6] = "";
live[0][4][7] = "";
live[0][4][8] = "";
live[0][4][9] = "";
live[0][4][10] = "";
live[0][4][11] = "";
live[0][4][12] = "";
```

```
// event 5
```

```
live[0][0][0] = "8";           // numer of the day
live[0][5][0] = "";           // name of the artist
live[0][5][1] = "";           // path of the event's cover
live[0][5][2] = "";           // path of the mp3
live[0][5][3] = "";           // time when event starts
live[0][5][4] = "";           // name of the location where the event happens
live[0][5][5] = "";           // address of the location
live[0][5][6] = "";           // reputation of the ocation
live[0][5][7] = "";           // price of the event
live[0][5][8] = "";           // name of the pre-party location
```

```

live[0][5][9] = ""; // address of the pre-party location
live[0][5][10] = ""; // number of participants confirmed for the event
live[0][5][11] = "";
live[0][5][12] = "";

//
*****
*****

//
*****
***** day 9

// day 9

// event 1
live[1][0][0] = "9"; // numer of the day
live[1][1][0] = "van she"; // name of the artist
live[1][1][1] = "vanshe_cover.png"; // path of the event's cover
live[1][1][2] = "mp3"; // path of the mp3
live[1][1][3] = "22:00"; // time when event starts
live[1][1][4] = "Auditorium Santa Margherita"; // name of the location where the event happens
live[1][1][5] = "Dorsoduro 3689"; // address of the location
live[1][1][6] = "0"; // reputation of the ocation
live[1][1][7] = a; // price of the event
live[1][1][8] = "Osteria alla Bifora"; // name of the pre-party location
live[1][1][9] = "Dorsoduro 2930, 21:30"; // address of the pre-party location
live[1][1][10] = "100"; // number of participants confirmed for the event
live[1][1][11] = "tuesday";
live[1][1][12] = "";

// event 2
live[1][0][0] = "9"; // numer of the day
live[1][2][0] = "radiohead"; // name of the artist
live[1][2][1] = "radiohead_cover.png"; // path of the event's cover
live[1][2][2] = "mp3 2"; // path of the mp3
live[1][2][3] = "23:00"; // time when event starts
live[1][2][4] = "Paradiso Perduto"; // name of the location where the event happens
live[1][2][5] = "Cannaregio 2540"; // address of the location
live[1][2][6] = "4"; // reputation of the ocation

```

```
live[1][2][7] = a;           // price of the event
live[1][2][8] = "Cantina Vecia Carbonera"; // name of the pre-party location
live[1][2][9] = "Cannaregio 2329, 22:30"; // address of the pre-party location
live[1][2][10] = "900";     // number of participants confirmed for the event
live[1][2][11] = "tuesday";
live[1][2][12] = "";

// event 3
live[1][0][0] = "9";       // numer of the day
live[1][3][0] = "";       // name of the artist
live[1][3][1] = "";       // path of the event's cover
live[1][3][2] = "";       // path of the mp3
live[1][3][3] = "";       // time when event starts
live[1][3][4] = "";       // name of the location where the event happens
live[1][3][5] = "";       // address of the location
live[1][3][6] = "";       // reputation of the ocation
live[1][3][7] = "";       // price of the event
live[1][3][8] = "";       // name of the pre-party location
live[1][3][9] = ""; // address of the pre-party location
live[1][3][10] = "";      // number of participants confirmed for the event
live[1][3][11] = "tuesday";
live[1][3][12] = "";

// event 4
live[1][0][0] = "9";       // numer of the day
live[1][4][0] = "";       // name of the artist
live[1][4][1] = "";       // path of the event's cover
live[1][4][2] = "";       // path of the mp3
live[1][4][3] = "";       // time when event starts
live[1][4][4] = "";       // name of the location where the event happens
live[1][4][5] = "";       // address of the location
live[1][4][6] = "";       // reputation of the ocation
live[1][4][7] = "";       // price of the event
live[1][4][8] = "";       // name of the pre-party location
live[1][4][9] = "";       // address of the pre-party location
live[1][4][10] = "";      // number of participants confirmed for the event
live[1][4][11] = "tuesday";
```

```

live[1][4][12] = "";

// event 5
live[1][0][0] = "9";           // numer of the day
live[1][5][0] = "";           // name of the artist
live[1][5][1] = "";           // path of the event's cover
live[1][5][2] = "";           // path of the mp3
live[1][5][3] = "";           // time when event starts
live[1][5][4] = "";           // name of the location where the event happens
live[1][5][5] = "";           // address of the location
live[1][5][6] = "";           // reputation of the ocation
live[1][5][7] = "";           // price of the event
live[1][5][8] = "";           // name of the pre-party location
live[1][5][9] = "";           // address of the pre-party location
live[1][5][10] = "";          // number of participants confirmed for the event
live[1][5][11] = "tuesday";
live[1][5][12] = "";

//
*****
*****

//
*****
***** day 10

// day 10

// event 1
live[2][0][0] = "10";         // numer of the day
live[2][1][0] = "justice";    // name of the artist
live[2][1][1] = "justice_cover.png"; // path of the event's cover
live[2][1][2] = "mp3";        // path of the mp3
live[2][1][3] = "19:30";      // time when event starts
live[2][1][4] = "Teatro Goldoni"; // name of the location where the event happens
live[2][1][5] = "San Marco 4650"; // address of the location
live[2][1][6] = "3";          // reputation of the ocation
live[2][1][7] = b;            // price of the event
live[2][1][8] = "Osteria Bancogiro"; // name of the pre-party location
live[2][1][9] = "San Polo 122, 19:00"; // address of the pre-party location

```

```

live[2][1][10] = "130";           // number of participants confirmed for the event
live[2][1][11] = "wednesday";
live[2][1][12] = "";

// event 2
live[2][0][0] = "10";           // numer of the day
live[2][2][0] = "digitalism";   // name of the artist
live[2][2][1] = "digitalism_cover.png"; // path of the event's cover
live[2][2][2] = "mp3 2";       // path of the mp3
live[2][2][3] = "23:30";       // time when event starts
live[2][2][4] = "Magazzini del Sale"; // name of the location where the event happens
live[2][2][5] = "Dorsoduro 259"; // address of the location
live[2][2][6] = "1";           // reputation of the ocation
live[2][2][7] = c;             // price of the event
live[2][2][8] = "Osteria alla Bifora"; // name of the pre-party location
live[2][2][9] = "Dorsoduro 2930, 23:00"; // address of the pre-party location
live[2][2][10] = "200";        // number of participants confirmed for the event
live[2][2][11] = "wednesday";
live[2][2][12] = "";

// event 3
live[2][0][0] = "10";           // numer of the day
live[2][3][0] = "";           // name of the artist
live[2][3][1] = "";           // path of the event's cover
live[2][3][2] = "";           // path of the mp3
live[2][3][3] = "";           // time when event starts
live[2][3][4] = "";           // name of the location where the event happens
live[2][3][5] = "";           // address of the location
live[2][3][6] = "";           // reputation of the ocation
live[2][3][7] = "";           // price of the event
live[2][3][8] = "";           // name of the pre-party location
live[2][3][9] = "";           // address of the pre-party location
live[2][3][10] = "";          // number of participants confirmed for the event
live[2][3][11] = "";
live[2][3][12] = "";

```

```
// event 4
live[2][0][0] = "10";           // numer of the day
live[2][4][0] = "";           // name of the artist
live[2][4][1] = "";           // path of the event's cover
live[2][4][2] = "";           // path of the mp3
live[2][4][3] = "";           // time when event starts
live[2][4][4] = "";           // name of the location where the event happens
live[2][4][5] = "";           // address of the location
live[2][4][6] = "";           // reputation of the ocation
live[2][4][7] = "";           // price of the event
live[2][4][8] = "";           // name of the pre-party location
live[2][4][9] = "";           // address of the pre-party location
live[2][4][10] = "";          // number of partecipants confirmed for the event
live[2][4][11] = "";
live[2][4][12] = "";
```

```
// event 5
live[2][0][0] = "10";           // numer of the day
live[2][5][0] = "";           // name of the artist
live[2][5][1] = "";           // path of the event's cover
live[2][5][2] = "";           // path of the mp3
live[2][5][3] = "";           // time when event starts
live[2][5][4] = "";           // name of the location where the event happens
live[2][5][5] = "";           // address of the location
live[2][5][6] = "";           // reputation of the ocation
live[2][5][7] = "";           // price of the event
live[2][5][8] = "";           // name of the pre-party location
live[2][5][9] = "";           // address of the pre-party location
live[2][5][10] = "";          // number of partecipants confirmed for the event
live[2][5][11] = "";
live[2][5][12] = "";
```

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

```

//
*****
***** day 11

// day 11

// event 1
live[3][0][0] = "11";           // numer of the day
live[3][1][0] = "the killers";  // name of the artist
live[3][1][1] = "thekillers_cover.png"; // path of the event's cover
live[3][1][2] = "song.mp3";     // path of the mp3
live[3][1][3] = "19:30";        // time when event starts
live[3][1][4] = "La vecchia malvasia"; // name of the location where the event happens
live[3][1][5] = "Dorsoduro 2548"; // address of the location
live[3][1][6] = "3";           // reputation of the ocation
live[3][1][7] = a;             // price of the event
live[3][1][8] = "Bar al Casanova"; // name of the pre-party location
live[3][1][9] = "Dorsoduro 1200, 18:30"; // address of the pre-party location
live[3][1][10] = "130";        // number of partecipants confirmed for the event
live[3][1][11] = "thursday";
live[3][1][12] = "Sam's town";

// event 2
live[3][0][0] = "11";           // numer of the day
live[3][2][0] = "klaxons";     // name of the artist
live[3][2][1] = "klaxons_cover.png"; // path of the event's cover
live[3][2][2] = "mp3 2";       // path of the mp3
live[3][2][3] = "23:30";        // time when event starts
live[3][2][4] = "Teatro Fondamenta Nuove"; // name of the location where the event happens
live[3][2][5] = "Cannaregio 5013"; // address of the location
live[3][2][6] = "3";           // reputation of the ocation
live[3][2][7] = b;             // price of the event
live[3][2][8] = "Cantina Vecia Carbonera"; // name of the pre-party location
live[3][2][9] = "Cannaregio 2329, 23:00"; // address of the pre-party location
live[3][2][10] = "200";        // number of partecipants confirmed for the event
live[3][2][11] = "thursday";
live[3][2][12] = "";

```

```
// event 3
live[3][0][0] = "11";           // numer of the day
live[3][3][0] = "";           // name of the artist
live[3][3][1] = "";           // path of the event's cover
live[3][3][2] = "";           // path of the mp3
live[3][3][3] = "";           // time when event starts
live[3][3][4] = "";           // name of the location where the event happens
live[3][3][5] = "";           // address of the location
live[3][3][6] = "";           // reputation of the ocation
live[3][3][7] = "";           // price of the event
live[3][3][8] = "";           // name of the pre-party location
live[3][3][9] = "";           // address of the pre-party location
live[3][3][10] = "";          // number of partecipants confirmed for the event
live[3][3][11] = "";
live[3][3][12] = "";
```

```
// event 4
live[3][0][0] = "11";           // numer of the day
live[3][4][0] = "";           // name of the artist
live[3][4][1] = "";           // path of the event's cover
live[3][4][2] = "";           // path of the mp3
live[3][4][3] = "";           // time when event starts
live[3][4][4] = "";           // name of the location where the event happens
live[3][4][5] = "";           // address of the location
live[3][4][6] = "";           // reputation of the ocation
live[3][4][7] = "";           // price of the event
live[3][4][8] = "";           // name of the pre-party location
live[3][4][9] = "";           // address of the pre-party location
live[3][4][10] = "";          // number of partecipants confirmed for the event
live[3][4][11] = "";
live[3][4][12] = "";
```

```
// event 5
live[3][0][0] = "11";           // numer of the day
live[3][5][0] = "";           // name of the artist
live[3][5][1] = "";           // path of the event's cover
```



```

live[3][5][2] = "";           // path of the mp3
live[3][5][3] = "";           // time when event starts
live[3][5][4] = "";           // name of the location where the event happens
live[3][5][5] = "";           // address of the location
live[3][5][6] = "";           // reputation of the ocaion
live[3][5][7] = "";           // price of the event
live[3][5][8] = "";           // name of the pre-party location
live[3][5][9] = "";           // address of the pre-party location
live[3][5][10] = "";          // number of participants confirmed for the event
live[3][5][11] = "";
live[3][5][12] = "";

//
*****
*****

//
*****
***** day 12

// day 12

// event 1
live[4][0][0] = "12";         // numer of the day
live[4][1][0] = "";           // name of the artist
live[4][1][1] = "";           // path of the event's cover
live[4][1][2] = "";           // path of the mp3
live[4][1][3] = "";           // time when event starts
live[4][1][4] = "";           // name of the location where the event happens
live[4][1][5] = "";           // address of the location
live[4][1][6] = "";           // reputation of the ocaion
live[4][1][7] = "";           // price of the event
live[4][1][8] = "";           // name of the pre-party location
live[4][1][9] = "";           // address of the pre-party location
live[4][1][10] = "";          // number of participants confirmed for the event
live[4][1][11] = "";
live[4][1][12] = "";

// event 2
live[4][0][0] = "12";         // numer of the day

```

```
live[4][2][0] = "";           // name of the artist
live[4][2][1] = "";           // path of the event's cover
live[4][2][2] = "";           // path of the mp3
live[4][2][3] = "";           // time when event starts
live[4][2][4] = "";           // name of the location where the event happens
live[4][2][5] = "";           // address of the location
live[4][2][6] = "";           // reputation of the ocation
live[4][2][7] = "";           // price of the event
live[4][2][8] = "";           // name of the pre-party location
live[4][2][9] = ""; // address of the pre-party location
live[4][2][10] = "";          // number of participants confirmed for the event
live[4][2][11] = "";
live[4][2][12] = "";
```

// event 3

```
live[4][0][0] = "12";         // numer of the day
live[4][3][0] = "";           // name of the artist
live[4][3][1] = "";           // path of the event's cover
live[4][3][2] = "";           // path of the mp3
live[4][3][3] = "";           // time when event starts
live[4][3][4] = "";           // name of the location where the event happens
live[4][3][5] = "";           // address of the location
live[4][3][6] = "";           // reputation of the ocation
live[4][3][7] = "";           // price of the event
live[4][3][8] = "";           // name of the pre-party location
live[4][3][9] = ""; // address of the pre-party location
live[4][3][10] = "";          // number of participants confirmed for the event
live[4][3][11] = "";
live[4][3][12] = "";
```

// event 4

```
live[4][0][0] = "12";         // numer of the day
live[4][4][0] = "";           // name of the artist
live[4][4][1] = "";           // path of the event's cover
live[4][4][2] = "";           // path of the mp3
live[4][4][3] = "";           // time when event starts
```

```
live[4][4][4] = "";           // name of the location where the event happens
live[4][4][5] = "";           // address of the location
live[4][4][6] = "";           // reputation of the ocation
live[4][4][7] = "";           // price of the event
live[4][4][8] = "";           // name of the pre-party location
live[4][4][9] = ""; // address of the pre-party location
live[4][4][10] = "700";        // number of participants confirmed for the event
live[4][4][11] = "";
live[4][4][12] = "";
```

// event 5

```
live[4][0][0] = "12";         // numer of the day
live[4][5][0] = "";           // name of the artist
live[4][5][1] = "";           // path of the event's cover
live[4][5][2] = "";           // path of the mp3
live[4][5][3] = "";           // time when event starts
live[4][5][4] = "";           // name of the location where the event happens
live[4][5][5] = "";           // address of the location
live[4][5][6] = "";           // reputation of the ocation
live[4][5][7] = "";           // price of the event
live[4][5][8] = "";           // name of the pre-party location
live[4][5][9] = ""; // address of the pre-party location
live[4][5][10] = "";          // number of participants confirmed for the event
live[4][5][11] = "";
live[4][5][12] = "";
```

//

```
*****
*****
```

//

```
*****
***** day 13
```

// day 13

// event 1

```
live[5][0][0] = "13";         // numer of the day
```

```

live[5][1][0] = "bloc party";           // name of the artist
live[5][1][1] = "bloccparty_cover.png"; // path of the event's cover
live[5][1][2] = "mp3";                 // path of the mp3
live[5][1][3] = "23:00";               // time when event starts
live[5][1][4] = "Il Boldù";            // name of the location where the event happens
live[5][1][5] = "Cannaregio 6000";     // address of the location
live[5][1][6] = "2";                   // reputation of the ocaction
live[5][1][7] = a;                     // price of the event
live[5][1][8] = "Osteria Bancogiro";   // name of the pre-party location
live[5][1][9] = "San Polo 122, 22:30"; // address of the pre-party location
live[5][1][10] = "700";                // number of partecipants confirmed for the event
live[5][1][11] = "saturday";
live[5][1][12] = "";

// event 2
live[5][0][0] = "13";                  // numer of the day
live[5][2][0] = "mstrkrft";           // name of the artist
live[5][2][1] = "mstrkrft_cover.png"; // path of the event's cover
live[5][2][2] = "mp3";                 // path of the mp3
live[5][2][3] = "23:00";               // time when event starts
live[5][2][4] = "Centro Sociale Morion"; // name of the location where the event happens
live[5][2][5] = "Castello 2951";       // address of the location
live[5][2][6] = "1";                   // reputation of the ocaction
live[5][2][7] = a;                     // price of the event
live[5][2][8] = "Cantina Vecia Carbonera"; // name of the pre-party location
live[5][2][9] = "Cannaregio 2329, 22:30"; // address of the pre-party location
live[5][2][10] = "700";                // number of partecipants confirmed for the event
live[5][2][11] = "saturday";
live[5][2][12] = "";

// event 3
live[5][0][0] = "13";                  // numer of the day
live[5][3][0] = "the preset";          // name of the artist
live[5][3][1] = "thepreset_cover.png"; // path of the event's cover
live[5][3][2] = "mp3";                 // path of the mp3
live[5][3][3] = "23:00";               // time when event starts

```

```

live[5][3][4] = "Devil's Forest Pub";          // name of the location where the event happens
live[5][3][5] = "San Marco 5185";           // address of the location
live[5][3][6] = "1";                        // reputation of the ocaation
live[5][3][7] = b;                          // price of the event
live[5][3][8] = "Osteria Bancogiro";       // name of the pre-party location
live[5][3][9] = "San Polo 122, 22:30";     // address of the pre-party location
live[5][3][10] = "700";                    // number of partecipants confirmed for the event
live[5][3][11] = "saturday";
live[5][3][12] = "";

// event 4
live[5][0][0] = "13";                      // numer of the day
live[5][4][0] = "";                        // name of the artist
live[5][4][1] = "";                        // path of the event's cover
live[5][4][2] = "";                        // path of the mp3
live[5][4][3] = "";                        // time when event starts
live[5][4][4] = "";                        // name of the location where the event happens
live[5][4][5] = "";                        // address of the location
live[5][4][6] = "";                        // reputation of the ocaation
live[5][4][7] = "";                        // price of the event
live[5][4][8] = "";                        // name of the pre-party location
live[5][4][9] = ""; // address of the pre-party location
live[5][4][10] = "";                       // number of partecipants confirmed for the event
live[5][4][11] = "";
live[5][4][12] = "";

// event 5
live[5][0][0] = "13";                      // numer of the day
live[5][5][0] = "";                        // name of the artist
live[5][5][1] = "";                        // path of the event's cover
live[5][5][2] = "";                        // path of the mp3
live[5][5][3] = "";                        // time when event starts
live[5][5][4] = "";                        // name of the location where the event happens
live[5][5][5] = "";                        // address of the location
live[5][5][6] = "";                        // reputation of the ocaation
live[5][5][7] = "";                        // price of the event

```

```

live[5][5][8] = ""; // name of the pre-party location
live[5][5][9] = ""; // address of the pre-party location
live[5][5][10] = ""; // number of participants confirmed for the event
live[5][5][11] = "";
live[5][5][12] = "";

//
*****
*****

//
*****
***** day 14

// day 14

// event 1
live[6][0][0] = "14"; // numer of the day
live[6][1][0] = "twenty six"; // name of the artist
live[6][1][1] = "twentysix_cover.png"; // path of the event's cover
live[6][1][2] = "mp3"; // path of the mp3
live[6][1][3] = "23:00"; // time when event starts
live[6][1][4] = "Magazzini del Sale"; // name of the location where the event happens
live[6][1][5] = "Dorsoduro 259"; // address of the location
live[6][1][6] = "4"; // reputation of the ocation
live[6][1][7] = c; // price of the event
live[6][1][8] = "Osteria alla Bifora "; // name of the pre-party location
live[6][1][9] = "Dorsoduro 2930, 22:30"; // address of the pre-party location
live[6][1][10] = "700"; // number of participants confirmed for the event
live[6][1][11] = "sunday";
live[6][1][12] = "";

// event 2
live[6][0][0] = "14"; // numer of the day
live[6][2][0] = ""; // name of the artist
live[6][2][1] = ""; // path of the event's cover
live[6][2][2] = ""; // path of the mp3
live[6][2][3] = ""; // time when event starts
live[6][2][4] = ""; // name of the location where the event happens

```

```
live[6][2][5] = "";           // address of the location
live[6][2][6] = "";           // reputation of the ocation
live[6][2][7] = "";           // price of the event
live[6][2][8] = "";           // name of the pre-party location
live[6][2][9] = ""; // address of the pre-party location
live[6][2][10] = "";          // number of participants confirmed for the event
live[6][2][11] = "";
live[6][2][12] = "";
```

// event 3

```
live[6][0][0] = "14";         // numer of the day
live[6][3][0] = "";           // name of the artist
live[6][3][1] = "";           // path of the event's cover
live[6][3][2] = "";           // path of the mp3
live[6][3][3] = "";           // time when event starts
live[6][3][4] = "";           // name of the location where the event happens
live[6][3][5] = "";           // address of the location
live[6][3][6] = "";           // reputation of the ocation
live[6][3][7] = "";           // price of the event
live[6][3][8] = "";           // name of the pre-party location
live[6][3][9] = ""; // address of the pre-party location
live[6][3][10] = "";          // number of participants confirmed for the event
live[6][3][11] = "";
live[6][3][12] = "";
```

// event 4

```
live[6][0][0] = "14";         // numer of the day
live[6][4][0] = "";           // name of the artist
live[6][4][1] = "";           // path of the event's cover
live[6][4][2] = "";           // path of the mp3
live[6][4][3] = "";           // time when event starts
live[6][4][4] = "";           // name of the location where the event happens
live[6][4][5] = "";           // address of the location
live[6][4][6] = "";           // reputation of the ocation
live[6][4][7] = "";           // price of the event
live[6][4][8] = "";           // name of the pre-party location
```

```

live[6][4][9] = ""; // address of the pre-party location
live[6][4][10] = ""; // number of participants confirmed for the event
live[6][4][11] = "";
live[6][4][12] = "";

// event 5
live[6][0][0] = "14"; // numer of the day
live[6][5][0] = ""; // name of the artist
live[6][5][1] = ""; // path of the event's cover
live[6][5][2] = ""; // path of the mp3
live[6][5][3] = ""; // time when event starts
live[6][5][4] = ""; // name of the location where the event happens
live[6][5][5] = ""; // address of the location
live[6][5][6] = ""; // reputation of the ocation
live[6][5][7] = ""; // price of the event
live[6][5][8] = ""; // name of the pre-party location
live[6][5][9] = ""; // address of the pre-party location
live[6][5][10] = ""; // number of participants confirmed for the event
live[6][5][11] = "";
live[6][5][12] = "";

}

```

```

String secretCode(){
    String[] lettere = new String[16];
    lettere[0] = "a";
    lettere[1] = "b";
    lettere[2] = "c";
    lettere[3] = "d";
    lettere[4] = "e";
    lettere[5] = "f";
    lettere[6] = "g";
    lettere[7] = "h";
    lettere[8] = "i";
    lettere[9] = "l";
}

```



```

lettere[10] = "m";
lettere[11] = "n";
lettere[12] = "o";
lettere[13] = "p";
lettere[14] = "q";
lettere[15] = "r";

int x = random(15);

String let = lettere[x];
String n1 = Integer.toString(random(9));
String n2 = Integer.toString(random(9));
String n3 = Integer.toString(random(9));
String n4 = Integer.toString(random(9));

code = let+n1+n2+n3+n4;
return code;
}
/*****
* Logic Section - Code that captures and interprets user input
*****/

// *****/
// Setup - Executes only once, prepares program to run (Logic Initialization)
// *****/
void setup()
{
    myPhone = new Phone(this);
    myPhone.fullscreen();
    s = new Sound("venicelive.wav"); // ridurre il file, è troppo pesante (1.1 MB)

    no_key1 = loadImage("no_key1.png");
    no_key2 = loadImage("no_key2.png");

    cap = new Capture(this); // comment to enable the demo in WirelessToolkit

```

```
menuFocus = MENU_OPTION_NAME;
```

```
menuFocusCalendar = focusDay;
```

```
screens = new PImage[8];
```

```
screens[0] = loadImage("screen0.png");
```

```
screens[1] = loadImage("screen1.png");
```

```
screens[2] = loadImage("screen2.png");
```

```
screens[3] = loadImage("screen3.png");
```

```
screens[4] = loadImage("screen4.png");
```

```
screens[5] = loadImage("screen5.png");
```

```
screens[6] = loadImage("screen6.png");
```

```
screens[7] = loadImage("screen7.png");
```

```
genre_img = new PImage[6];
```

```
genre_img[0] = loadImage("rect_rock.png");
```

```
genre_img[1] = loadImage("rect_pop.png");
```

```
genre_img[2] = loadImage("rect_jazz.png");
```

```
genre_img[3] = loadImage("rect_electro.png");
```

```
genre_img[4] = loadImage("rect_alternative.png");
```

```
genre_img[5] = loadImage("rect_all.png");
```

```
genreStatus = 0;
```

```
moving_genre = false;
```

```
genre_img_selected = new PImage[6];
```

```
genre_img_selected[0] = loadImage("rect_rock_select.png");
```

```
genre_img_selected[1] = loadImage("rect_pop_select.png");
```

```
genre_img_selected[2] = loadImage("rect_jazz_select.png");
```

```
genre_img_selected[3] = loadImage("rect_electro_select.png");
```

```
genre_img_selected[4] = loadImage("rect_alternative_select.png");
```

```
genre_img_selected[5] = loadImage("rect_all_select.png");
```

```
day_selected = new PImage[7];
```

```
n_scroll_days = 0;
```

```
for(int i=8;i<15;i++){
```

```
    day_selected[n_scroll_days] = loadImage("day"+i+".png");
```

```
    n_scroll_days++;
```

```

}

nickname_over = loadImage("nickname_over.png");
photo_over = loadImage("photo_over.png");
border_photo = loadImage("border_photo.png");
calendar = loadImage("calendar.png");
day_arrow_left = loadImage("day_arrow_left.png");
day_arrow_left_ok = loadImage("day_arrow_left_ok.png");
day_arrow_right = loadImage("day_arrow_right.png");
day_arrow_right_ok = loadImage("day_arrow_right_ok.png");
event_arrow_left = loadImage("event_arrow_left.png");
event_arrow_left_ok = loadImage("event_arrow_left_ok.png");
event_arrow_right = loadImage("event_arrow_right.png");
event_arrow_right_ok = loadImage("event_arrow_right_ok.png");

currentDay = 1; // inizialmente il primo giorno selezionato è il secondo tra quelli disponibili, così da riempire tutti gli
spazi del calendario e non avere buchi

go_day_sx = false;
go_day_dx = false;
go_event_sx = false;
go_event_dx = false;
events = true;
font = loadFont("Verdana-14.mv1w", color(0,0,0));
textFont(font);
font_name_artist = loadFont("Verdana-20.mv1w", color(203,201,201));
font_code = loadFont("Verdana-16.mv1w", color(203,201,201));
font_details_event = loadFont("Verdana-16.mv1w", color(35,34,35));
font_ask = loadFont("Verdana-12.mv1w", color(203,201,201));

view_select_screen1 = false;

// LIVE[giorni][eventi per giorno][informazioni per un determinato evento]
oneday = 7;
event = 6;
details = 13;
live = new String[oneday][event][details];
fillLive();

```

```
border_cover = loadImage("border_cover.png");
no_event = loadImage("no_event.png");
currentEvent = 1;

// variables screen 3
modeInfo = true;
modePlayer = false;
str_place = loadImage("str_place.png");
str_ticketprice = loadImage("str_ticketprice.png");
star = loadImage("star.png");
star_ok = loadImage("star_ok.png");
bg_info = loadImage("bg_info.png");
bg_player = loadImage("bg_player.png");

bg_ticket = loadImage("bg_ticket.png");
bg_reservation = loadImage("bg_reservation.png");

less = loadImage("less.png");
less_ok = loadImage("less_ok.png");
more = loadImage("more.png");
more_ok = loadImage("more_ok.png");
ticketNumber = 1;
numTicketStr = "";
totalPrice = "";
buy = loadImage("buy.png");
book = loadImage("book.png");
arrow_more = false;
arrow_less = false;
zoom = false;
coverBig = loadImage("killersBig.png");

code = secretCode();
}
```

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

```

// KeyPressed, Input - Executes whenever a key is pressed...
// *****
void keyPressed()
{
  if(keyCode == SOFTKEY1){
    if(currentScreen==1){
      exit();
    }
    else if(currentScreen == 5) {
      currentScreen = 2;
    }
    else if(currentScreen == 6) {
      currentScreen = 2;
    }
    else if(currentScreen == 7) {
      currentScreen = 2;
    }
    else{
      currentScreen = currentScreen-1;
    }
  }

  if((keyCode == FIRE) && (currentScreen==0)){
    currentScreen = currentScreen + 1;
  }

  if(keyCode == SOFTKEY2){
    if((currentScreen==0)) {
      currentScreen = currentScreen + 1;
    }
    if(currentScreen==1){
      if(view_select_screen1){
        currentScreen = currentScreen+1;
      }
    }
    else if(currentScreen==2){

```

```

if(events){ currentScreen = currentScreen+1; }
}
else if(currentScreen==3){
    // se è totalmente gratis +2
    // altrimenti +1
    if((live[finalDay][finalEvent][7] == "Euro 10") || live[finalDay][finalEvent][7] == "Euro 0 - Reservation
required"){
        currentScreen = currentScreen+1;
    }
    else if(live[finalDay][finalEvent][7] == "Free entry"){
        currentScreen = currentScreen+2;
    }
}
else if(currentScreen == 7) {
    exit();
}
else{
    currentScreen = currentScreen+1;
}
}

// screen 1
if((currentScreen==1)) {

if(menuFocus == MENU_OPTION_NAME){
    if(keyCode == DOWN){
        menuFocus = MENU_OPTION_PICTURE;
    }
}
else if(menuFocus == MENU_OPTION_PICTURE){
    if(keyCode == FIRE){
        if(showingCamera){
            hideCamera();
        }
        else {
            showCamera();
        }
    }
}
}
}

```

```

}

// UP e DOWN non possono essere premuti durante lo scatto della foto o il funzionamento della fotocamera
if(!showingCamera){
    if(keyCode == UP) {
        menuFocus = MENU_OPTION_NAME;
    }
    if(keyCode == DOWN) {
        menuFocus = MENU_OPTION_GENRE;
    }
}
}

else if(menuFocus == MENU_OPTION_GENRE){
    if(keyCode == LEFT) {
        genrePrev();
    }
    if(keyCode == RIGHT) {
        genreNext();
    }
    if(keyCode == UP) {
        menuFocus = MENU_OPTION_PICTURE;
    }
}
}

} // end currentScreen=1

if(currentScreen == 2){
    // controllo il focus sul calendario
    if(menuFocusCalendar == focusDay){
        currentEvent = 1; // azzero il currentEvent quando torno in questo menu dopo aver scrollato gli eventi di un altro
        giorno perchè possono esserci problemi con gli indici dell'array

        // gestione dello scroll del calendario
        if(keyCode == LEFT) {
            // currentDay può essere un numero compreso tra 0 e 6
            if(currentDay!=0){
                currentDay = currentDay-1;
                go_day_sx = true;
            }
        }
    }
}
}

```

```

    }
}
if(keyCode == RIGHT) {
    if(currentDay!=6){
        currentDay = currentDay+1;
        go_day_dx = true;

    }
}
if(events){
    if(keyCode == DOWN) { menuFocusCalendar = focusEvent; }
}
}
if(menuFocusCalendar == focusEvent){
    // gestione scroll eventi all'interno di un giorno
    if(keyCode == UP) { menuFocusCalendar = focusDay; }
    if(live[currentDay][currentEvent][0]=="the killers"){
        if(keyCode == FIRE) { zoom = !zoom; }
    }
    if(keyCode == LEFT) {
        // currentEvent può essere un numero compreso tra 1 e 5 estremi compresi
        if(currentEvent>1 && currentEvent < 6){ //
            if(live[currentDay][currentEvent-1][0]!=""){
                currentEvent = currentEvent-1;
                go_event_sx = true;
            }
        }
    }
}

if(keyCode == RIGHT) {
    // currentEvent può essere un numero compreso tra 1 e 5 estremi compresi
    if(currentEvent>=1 && currentEvent < 5){ //
        if(live[currentDay][currentEvent+1][0]!=""){
            currentEvent = currentEvent+1;
            go_event_dx = true;
        }
    }
}

```



```
    }  
  }  
  
}  
  
}  
  
if(currentScreen == 3){  
  if(keyCode == LEFT) {  
    modePlayer = true;  
    modeInfo = false;  
  }  
  if(keyCode == RIGHT) {  
    modePlayer = false;  
    modeInfo = true;  
  }  
}  
  
if(currentScreen == 4){  
  if(keyCode == LEFT) {  
    if(ticketNumber > 1) {  
      arrow_less = true;  
      ticketNumber = ticketNumber - 1;  
    }  
  }  
  if(keyCode == RIGHT) {  
    arrow_more = true;  
    ticketNumber = ticketNumber + 1;  
  }  
}  
  
} // end keyPressed  
  
void keyReleased()
```

```
{  
  if(currentScreen == 2){  
    if(menuFocusCalendar == focusDay){  
      if(go_day_sx) { go_day_sx = false; }  
      if(go_day_dx) { go_day_dx = false; }  
    }  
    if(menuFocusCalendar == focusEvent){  
      if(go_event_sx) { go_event_sx = false; }  
      if(go_event_dx) { go_event_dx = false; }  
    }  
  }  
}
```