

```
/*
Created by Alessandro Carraretto based on the Mobile Processing code developed for
the IUAV Interaction Design Programme by David Mellis, Vinay Ventrakamen
and Nicholas Zambetti 2005-07. See www.interaction-venice.com/resources/?page_id=5.
```

The prototype uses the following code fragments:

1 - Basic structure of a fullscreen mobile processing program that loads and displays an image using functions with a parameter for image selection (focus) and constants for making image selection easier for people remember, and a variable to make the image selection controllable via logic.

2 - Functions for ease animation developed by David Mellis, Vinay Ventrakamen and Nicholas Zambetti 2005-07.

3 - Camera functions developed with Vinay Ventrakamen and Nicholas Zambetti.  
\*/

```
/******
 * masQuest 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
////////////////////////////////////

//fonts
PFont verdana11 = loadFont("Verdana-11.mvlw");
PFont verdanaBold11 = loadFont("Verdana-Bold-11.mvlw");

// named references to graphics
PImage logo;
PImage menuLinguaStripe;
PImage bars;
PImage masQuestTop;
PImage ok;
PImage cancel;
PImage leftArrow;
PImage rightArrow;
PImage q;
PImage welcome;
PImage welcome2;
PImage genreStripe;
PImage fantasyTxt;
PImage tvAndMoviesTxt;
PImage superHeroesTxt;
PImage traditionalMasksTxt;
PImage selectGameBackground;
PImage selectGameSelect;
PImage selectGameTxt;
PImage groupStripe;
PImage hobbitsTxt;
PImage orcsTxt;
PImage elvesTxt;
PImage demonsTxt;
PImage hobbitsDescription;
PImage hobbitsRole;
PImage confirm;
PImage gameStarted;
PImage hobbits1stClue;
PImage duringGameStripe;
PImage gameProgress;
PImage menuTxt;
PImage gameMenu;
PImage gameMenuSelect;
PImage gameMenuTxt;
PImage cluesMap;
PImage messagesBackground;
PImage messagesTxt;
PImage sendMessage;
PImage helpBackground;
PImage helpTxtStripe;
PImage helpVerticalBar;
PImage helpBars;
PImage quit;
PImage semacode;
PImage hobbits1stComplete;
PImage hobbits2ndFailed;
```

```

PImage sendSemacode;
PImage minutesLeft;
PImage timeUp;
PImage winnerBars;
PImage winnerStripe;

Sound fruscio;

// function to load all the images for the interface
void loadImages()
{
  // menu graphics

  //initial logo
  logo = loadImage("logo.png");

  //language menu
  /*
  menuLinguaItalian = loadImage("menuLinguaItalian.png");
  menuLinguaFrench = loadImage("menuLinguaFrench.png");
  menuLinguaEnglish = loadImage("menuLinguaEnglish.png");
  */

  menuLinguaStripe = loadImage("menuLinguaStripe.png");
  bars = loadImage("bars.png");
  q = loadImage("q.png");
  masQuestTop = loadImage("masQuestTop.png");
  ok = loadImage("ok.png");
  cancel = loadImage("cancel.png");
  leftArrow = loadImage("leftArrow.png");
  rightArrow = loadImage("rightArrow.png");
  welcome = loadImage("welcome.png");
  welcome2 = loadImage("welcome2.png");
  genreStripe = loadImage("genreStripe.png");
  fantasyTxt = loadImage("fantasyTxt.png");
  tvAndMoviesTxt = loadImage("tvAndMoviesTxt.png");
  superHeroesTxt = loadImage("superHeroesTxt.png");
  traditionalMasksTxt = loadImage("traditionalMasksTxt.png");
  selectGameBackground = loadImage("selectGameBackground.png");
  selectGameSelect = loadImage("selectGameSelect.png");
  selectGameTxt = loadImage("selectGameTxt.png");
  groupStripe = loadImage("groupStripe.png");
  hobbitsTxt = loadImage("hobbitsTxt.png");
  orcsTxt = loadImage("orcsTxt.png");
  elvesTxt = loadImage("elvesTxt.png");
  demonsTxt = loadImage("demonsTxt.png");
  hobbitsDescription = loadImage("hobbitsDescription.png");
  hobbitsRole = loadImage("hobbitsRole.png");
  confirm = loadImage("confirm.png");
  gameStarted = loadImage("gameStarted.png");
  hobbits1stClue = loadImage("hobbits1stClue.png");
  duringGameStripe = loadImage("duringGameStripe.png");
  gameProgress = loadImage("gameProgress.png");
  menuTxt = loadImage("menuTxt.png");
  gameMenu = loadImage("gameMenu.png");
  gameMenuSelect = loadImage("gameMenuSelect.png");
  gameMenuTxt = loadImage("gameMenuTxt.png");
  cluesMap = loadImage("cluesMap.png");
  messagesBackground = loadImage("messagesBackground.png");
  messagesTxt = loadImage("messagesTxt.png");
  sendMessage = loadImage("sendMessage.png");
  helpBackground = loadImage("helpBackground.png");
  helpTxtStripe = loadImage("helpTxtStripe.png");
  helpVerticalBar = loadImage("helpVerticalBar.png");
  helpBars = loadImage("helpBars.png");
  quit = loadImage("quit.png");
  semacode = loadImage("semacode.png");
  hobbits1stComplete = loadImage("hobbits1stComplete.png");
  hobbits2ndFailed = loadImage("hobbits2ndFailed.png");
  sendSemacode = loadImage("sendSemacode.png");
  minutesLeft = loadImage("minutesLeft.png");
  timeUp = loadImage("timeUp.png");
  winnerBars = loadImage("winnerBars.png");
  winnerStripe = loadImage("winnerStripe.png");
}

void loadSounds(){
  fruscio = new Sound("fruscio.mp3", "audio/mpeg");
}

```



```

//draw the Q in the top left corner
image(q, 0, 0, 14, 17, 6, 2);
}

void drawGenreSelection(int focus){
println("GENREFOCUS: " + focus);
//draw the background
background(0);

//draw the circles
ellipseMode(CENTER);
fill(255);
noStroke();
ellipse(58, 166, 4, 4);
ellipse(81, 166, 4, 4);
ellipse(101, 166, 4, 4);
ellipse(121, 166, 4, 4);
//---

//draw the genre icons
if(focus == GENRE_FOCUS_FANTASY){
genreX = easeTo(genreX, POSITION_FANTASY, easeSpeed);
image(genreStripe, genreX, 0);

image(fantasyTxt, 0, 144);

//draw the selected circle
fill(255);
noStroke();
ellipse(58, 166, 10, 10);
//---
}
else if(focus == GENRE_FOCUS_TVANDMOVIES){
genreX = easeTo(genreX, POSITION_TVANDMOVIES, easeSpeed);
image(genreStripe, genreX, 0);

image(tvAndMoviesTxt, 0, 144);

//draw the selected circle
fill(255);
noStroke();
ellipse(81, 166, 10, 10);
//---
}
else if(focus == GENRE_FOCUS_SUPERHEROES){
genreX = easeTo(genreX, POSITION_SUPERHEROES, easeSpeed);
image(genreStripe, genreX, 0);

image(superHeroesTxt, 0, 144);

//draw the selected circle
fill(255);
noStroke();
ellipse(101, 166, 10, 10);
//---
}
else if(focus == GENRE_FOCUS_TRADITIONALMASKS){
genreX = easeTo(genreX, POSITION_TRADITIONALMASKS, easeSpeed);
image(genreStripe, genreX, 0);

image(traditionalMasksTxt, 0, 144);

//draw the selected circle
fill(255);
noStroke();
ellipse(121, 166, 10, 10);
//---
}

//draw the arrows
//left
image(leftArrow, 0, 0, 6, 12, 5, 96);
//right
image(rightArrow, 0, 0, 6, 12, 165, 96);

//draw the menubars
image(bars, 0, 0);
image(ok, 141, 190);

```

```

image(cancel, 26, 190);

//draw the title
image(masQuestTop, 0, 0);

//draw the Q in the top left corner
image(q, 0, 0, 14, 17, 6, 2);
}

void drawSelectGame(int focus){
background(0);

image(selectGameBackground, 0, 0);

if(focus == GAME_FOCUS_1){
selectGameY = easeTo(selectGameY, POSITION_GAME_1, easeSpeed-3);
image(selectGameSelect, 2, selectGameY);
}
else if(focus == GAME_FOCUS_2){
selectGameY = easeTo(selectGameY, POSITION_GAME_2, easeSpeed-3);
image(selectGameSelect, 2, selectGameY);
}
else if(focus == GAME_FOCUS_3){
selectGameY = easeTo(selectGameY, POSITION_GAME_3, easeSpeed-3);
image(selectGameSelect, 2, selectGameY);
}

image(selectGameTxt, 0, 0);
}

void drawSelectGroup(int focus){
//draw the background
background(0);

//draw the circles
ellipseMode(CENTER);
fill(255);
noStroke();
ellipse(58, 166, 4, 4);
ellipse(81, 166, 4, 4);
ellipse(101, 166, 4, 4);
ellipse(121, 166, 4, 4);
//---

//draw the group icons
if(focus == GROUP_FOCUS_HOBBITS){
groupX = easeTo(groupX, POSITION_HOBBITS, easeSpeed);
image(groupStripe, groupX, 0);

image(hobbitsTxt, 0, 144);

//draw the selected circle
fill(255);
noStroke();
ellipse(58, 166, 10, 10);
//---
}
else if(focus == GROUP_FOCUS_ORCS){
groupX = easeTo(groupX, POSITION_ORCS, easeSpeed);
image(groupStripe, groupX, 0);

image(orcsTxt, 0, 144);

//draw the selected circle
fill(255);
noStroke();
ellipse(81, 166, 10, 10);
//---
}
else if(focus == GROUP_FOCUS_ELVES){
groupX = easeTo(groupX, POSITION_ELVES, easeSpeed);
image(groupStripe, groupX, 0);

image(elvesTxt, 0, 144);

//draw the selected circle
fill(255);
noStroke();
ellipse(101, 166, 10, 10);
}
}

```

```

    //---
}
else if(focus == GROUP_FOCUS_DEMONS){
    groupX = easeTo(groupX, POSITION_DEMONS, easeSpeed);
    image(groupStripe, groupX, 0);

    image(demonsTxt, 0, 144);

    //draw the selected circle
    fill(255);
    noStroke();
    ellipse(121, 166, 10, 10);
    //---
}

//draw the arrows
//left
image(leftArrow, 0, 0, 6, 12, 5, 96);
//right
image(rightArrow, 0, 0, 6, 12, 165, 96);

//draw the menubars
image(bars, 0, 0);
image(ok, 141, 190);
image(cancel, 26, 190);

//draw the title
image(masQuestTop, 0, 0);

//draw the Q in the top left corner
image(q, 14, 0, 14, 17, 6, 2);
}

void drawDuringGame(int focus){
    //draw the background
    background(0);

    //draw the circles
    ellipseMode(CENTER);
    fill(255);
    noStroke();
    ellipse(68, 166, 4, 4);
    ellipse(91, 166, 4, 4);
    ellipse(114, 166, 4, 4);
    //---

    //draw the during game screen
    if(focus == DURING_GAME_FOCUS_1){
        //image(menuLinguaItalian, 0, 0);
        duringGameX = easeTo(duringGameX, POSITION_DURING_GAME_1, easeSpeed);
        image(duringGameStripe, duringGameX, 0);

        //draw the selected circle
        fill(255);
        noStroke();
        ellipse(68, 166, 10, 10);
        //---
    }
    else if(focus == DURING_GAME_FOCUS_2){
        //image(menuLinguaFrench, 0, 0);
        duringGameX = easeTo(duringGameX, POSITION_DURING_GAME_2, easeSpeed);
        image(duringGameStripe, duringGameX, 0);

        //draw the selected circle
        fill(255);
        noStroke();
        ellipse(91, 166, 10, 10);
        //---
    }
    else if(focus == DURING_GAME_FOCUS_3){
        //image(menuLinguaEnglish, 0, 0);
        duringGameX = easeTo(duringGameX, POSITION_DURING_GAME_3, easeSpeed);
        image(duringGameStripe, duringGameX, 0);

        //draw the selected circle
        fill(255);
        noStroke();
        ellipse(114, 166, 10, 10);
        //---
    }
}

```

```

}

//draw the arrows
//left
image(leftArrow, 0, 0, 6, 12, 5, 96);
//right
image(rightArrow, 0, 0, 6, 12, 165, 96);

//draw the menubars
image(bars, 0, 0);
image(ok, 141, 190);
image(menuTxt, 0, 0);
image(cancel, 26, 190);

//draw the title
image(gameProgress, 0, 0);

//draw the Q in the top left corner
image(q, 14, 0, 14, 17, 6, 2);
}

void drawMenu(int focus){
    image(gameMenu, 1, 0);

    if(focus == GAME_MENU_FOCUS_CLUES){
        gameMenuSelectY = easeTo(gameMenuSelectY, GAME_MENU_CLUES_POSITION, easeSpeed-3);
        image(gameMenuSelect, 1, gameMenuSelectY);
    }
    else if(focus == GAME_MENU_FOCUS_MESSAGES){
        gameMenuSelectY = easeTo(gameMenuSelectY, GAME_MENU_MESSAGES_POSITION, easeSpeed-3);
        image(gameMenuSelect, 1, gameMenuSelectY);
    }
    else if(focus == GAME_MENU_FOCUS_HELP){
        gameMenuSelectY = easeTo(gameMenuSelectY, GAME_MENU_HELP_POSITION, easeSpeed-3);
        image(gameMenuSelect, 1, gameMenuSelectY);
    }
    else if(focus == GAME_MENU_FOCUS_QUIT){
        gameMenuSelectY = easeTo(gameMenuSelectY, GAME_MENU_QUIT_POSITION, easeSpeed-3);
        image(gameMenuSelect, 1, gameMenuSelectY);
    }
}

    image(gameMenuTxt, 0, 0);
}

void drawMessageMenu(int focus){
    image(messagesBackground, 1, 0);

    if(focus == MESSAGE_FOCUS_INBOX){
        messageMenuY = easeTo(messageMenuY, MESSAGE_INBOX_POSITION, easeSpeed-3);
        image(gameMenuSelect, 1, messageMenuY);
    }
    else if(focus == MESSAGE_FOCUS_SEND){
        messageMenuY = easeTo(messageMenuY, MESSAGE_SEND_POSITION, easeSpeed-3);
        image(gameMenuSelect, 1, messageMenuY);
    }
}

    image(messagesTxt, 0, 0);
}

void drawHelp(int focus){
    //background(0);
    image(helpBackground, 1, 0);

    if(focus == HELP_FOCUS_PAGE1){
        helpY = easeTo(helpY, HELP_PAGE1_POSITION, easeSpeed-3);
        image(helpTxtStripe, 1, helpY);
    }
    else if(focus == HELP_FOCUS_PAGE2){
        helpY = easeTo(helpY, HELP_PAGE2_POSITION, easeSpeed-3);
        image(helpTxtStripe, 1, helpY);
    }
}

    image(helpBars, 0, 0);
}

void drawWinner(int focus){
    background(0);

    //draw the circles

```

```

ellipseMode(CENTER);
fill(255);
noStroke();
ellipse(68, 166, 4, 4);
ellipse(91, 166, 4, 4);
ellipse(114, 166, 4, 4);
//---

//draw the flags
if(focus == WINNER_FOCUS_1){
  //image(menuLinguaItalian, 0, 0);
  winnerX = easeTo(winnerX, WINNER_POSITION_1, easeSpeed);
  image(winnerStripe, winnerX, 0);

  //draw the selected circle
  fill(255);
  noStroke();
  ellipse(68, 166, 10, 10);
  //---
}
else if(focus == WINNER_FOCUS_2){
  //image(menuLinguaFrench, 0, 0);
  winnerX = easeTo(winnerX, WINNER_POSITION_2, easeSpeed);
  image(winnerStripe, winnerX, 0);

  //draw the selected circle
  fill(255);
  noStroke();
  ellipse(91, 166, 10, 10);
  //---
}
else if(focus == WINNER_FOCUS_3){
  //image(menuLinguaEnglish, 0, 0);
  winnerX = easeTo(winnerX, WINNER_POSITION_3, easeSpeed);
  image(winnerStripe, winnerX, 0);

  //draw the selected circle
  fill(255);
  noStroke();
  ellipse(114, 166, 10, 10);
  //---
}

//draw the arrows
//left
image(leftArrow, 0, 0, 6, 12, 5, 96);
//right
image(rightArrow, 0, 0, 6, 12, 165, 96);

//draw the menubars
image(winnerBars, 0, 0);
image(ok, 141, 190);

//draw the title
image(masQuestTop, 0, 0);

//draw the Q in the top left corner
image(q, 0, 0, 14, 17, 6, 2);
}

void drawMessage(){
  rect(0,0,width,height);
  text(render_text, 15, 100);
}

////////////////////////////////////
// Camera Functions
////////////////////////////////////

void showCamera(int x, int y, int w, int h, PImage imageBackground)
{
  image(imageBackground, 0, 0);
  noLoop();
  myCapture.show(x,y,w,h);
}

void hideCamera()
{
  myCapture.hide();
}

```



```

//myCapture.close();
loop();
}

void destroy()
{
  myCapture.close();
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Movement Functions
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

int easeTo(int currentOriginal, int targetOriginal, int speed)
{
  // make values large for math
  int current = currentOriginal * 1000; // make current value large for math (e.g. 2 becomes 2000)
  int target = targetOriginal * 1000; // make target value large for math (e.g. 33 becomes 33000)

  // do math to calculate our next value
  int change = target - current; // find out how much change there is (e.g. 33000 - 2000 = 31000)
  int changeLittle = change / speed; // make the change a little change (e.g. 31000 / 4 = 7750)
  int next = current + changeLittle; // change the current value a little (e.g. 2000 + 7750 = 9750)

  // make next value small for screen
  next = next / 1000;

  // if our little change was so little that we didn't move...
  if(next == currentOriginal){
    next = targetOriginal; // our next step is our target
  }

  return next; // return our next value (e.g. 9750 / 1000 = 9, remember that we started with 2)
}

/*****
* ZóGame Logic Section
* This section of the program is for the logic; how we decide what graphics to show.
* You can think of it as the section for code that captures and interprets user input (actions)
*****/

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

import processing.video.*; //import video library for use the camera
import processing.phone.*; // import phone library to go fullscreen
import processing.sound.*;
Phone myPhone; // named reference to phone instance
Capture myCapture; //to use a capture object

void setup() // happens only once, when the program starts...
{
  // go fullscreen
  myPhone = new Phone(this); // create new phone instance/controller
  myPhone.fullscreen(); // tell phone to go fullscreen

  myCapture = new Capture(this); //for using the camera

  loadImages(); // load images
  loadSounds(); // load sounds
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// 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_LOGO = 0;
int MODE_LANGUAGE_MENU = 1;
int MODE_WELCOME = 2;
int MODE_WELCOME_2 = 3;
int MODE_SELECT_GENRE = 4;
int MODE_SELECT_GAME = 5;
int MODE_SELECT_GROUP = 6;
int MODE_DESCRIPTION = 7;

```

```

int MODE_ROLE = 8;
int MODE_CONFIRM = 9;
int MODE_GAME_STARTED = 10;
int MODE_HOBBITS_1ST_CLUE = 11;
int MODE_MENU = 12;
int MODE_DURING_GAME = 13;
int MODE_SEMACODE = 14;
int MODE_SHOW_CLUES = 15;
int MODE_SHOW_MESSAGES = 16;
int MODE_SHOW_HELP = 17;
int MODE_QUIT = 18;
int MODE_INBOX = 19;
int MODE_SEND_MESSAGE = 20;
int MODE_HOBBITS_1ST_COMPLETE = 21;
int MODE_DURING_GAME2 = 22;
int MODE_HOBBITS_2ND_FAILED = 23;
int MODE_MINUTES_LEFT = 24;
int MODE_TIME_UP = 25;
int MODE_WINNER = 26;

// names for each possible focus option of the main menu
int LANGUAGE_FOCUS_ITALIAN = 0;
int LANGUAGE_FOCUS_ENGLISH = 1;
int LANGUAGE_FOCUS_FRENCH = 2;

//position for each position of language menu
int POSITION_ITALIAN = 0;
int POSITION_ENGLISH = -176;
int POSITION_FRENCH = -352;

//names for each position of the genre selection
int GENRE_FOCUS_FANTASY = 1;
int GENRE_FOCUS_TVANDMOVIES = 2;
int GENRE_FOCUS_SUPERHEROES = 3;
int GENRE_FOCUS_TRADITIONALMASKS = 4;

//position for each position of genre menu
int POSITION_FANTASY = 0;
int POSITION_TVANDMOVIES = -176;
int POSITION_SUPERHEROES = -352;
int POSITION_TRADITIONALMASKS = -528;

//names for each position of the game selection
int GAME_FOCUS_1 = 0;
int GAME_FOCUS_2 = 1;
int GAME_FOCUS_3 = 2;

//names for each position of the group selection
int POSITION_GAME_1 = 30;
int POSITION_GAME_2 = 70;
int POSITION_GAME_3 = 113;

//names for each position of the group selection
int GROUP_FOCUS_HOBBITS = 1;
int GROUP_FOCUS_ORCS = 2;
int GROUP_FOCUS_ELVES = 3;
int GROUP_FOCUS_DEMONS = 4;

//position for each position of group menu
int POSITION_HOBBITS = 0;
int POSITION_ORCS = -176;
int POSITION_ELVES = -352;
int POSITION_DEMONS = -528;

//names for each possible group description
int DESCRIPTION_FOCUS_HOBBITS = 1;
int DESCRIPTION_FOCUS_ORCS = 2;
int DESCRIPTION_FOCUS_ELVES = 3;
int DESCRIPTION_FOCUS_DEMONS = 4;

//names for each possible focus in the role screen
int ROLE_FOCUS_HOBBITS = 1;
int ROLE_FOCUS_ORCS = 2;
int ROLE_FOCUS_ELVES = 3;
int ROLE_FOCUS_DEMONS = 4;

//during game focus
int DURING_GAME_FOCUS_1 = 0;
int DURING_GAME_FOCUS_2 = 1;

```

```

int DURING_GAME_FOCUS_3 = 2;

//during game positions
int POSITION_DURING_GAME_1 = 0;
int POSITION_DURING_GAME_2 = -176;
int POSITION_DURING_GAME_3 = -352;

//menu focus
int GAME_MENU_FOCUS_CLUES = 0;
int GAME_MENU_FOCUS_MESSAGES = 1;
int GAME_MENU_FOCUS_HELP = 2;
int GAME_MENU_FOCUS_QUIT = 3;

//menu positions
int GAME_MENU_CLUES_POSITION = 34;
int GAME_MENU_MESSAGES_POSITION = 63;
int GAME_MENU_HELP_POSITION = 90;
int GAME_MENU_QUIT_POSITION = 118;

//message menu focus
int MESSAGE_FOCUS_INBOX = 0;
int MESSAGE_FOCUS_SEND = 1;

//message menu positions
int MESSAGE_INBOX_POSITION = 40;
int MESSAGE_SEND_POSITION = 76;

//help menu focus
int HELP_FOCUS_PAGE1 = 0;
int HELP_FOCUS_PAGE2 = 1;

//help menu positions
int HELP_PAGE1_POSITION = 0;
int HELP_PAGE2_POSITION = -145;

//winner menu focus
int WINNER_FOCUS_1 = 0;
int WINNER_FOCUS_2 = 1;
int WINNER_FOCUS_3 = 2;

//winner menu positions
int WINNER_POSITION_1 = 0;
int WINNER_POSITION_2 = -176;
int WINNER_POSITION_3 = -352;

//variables for show the selected arrow
int arrowCount = 1;

////////////////////////
// State, Information Collected From Use (Variables)
////////////////////////

// main menu information
int mode = MODE_LOGO; // initially, we are showing the main menu

// main menu information
int menuFocus = LANGUAGE_FOCUS_ITALIAN; // initially, the main menu focus is on the first item "Gioca"
int genreFocus = GENRE_FOCUS_FANTASY;
int gameFocus = GAME_FOCUS_1;
int groupFocus = GROUP_FOCUS_HOBBITS;
int descriptionFocus = DESCRIPTION_FOCUS_HOBBITS;
int roleFocus;
int duringGameFocus = DURING_GAME_FOCUS_1;
int gameMenuFocus = GAME_MENU_FOCUS_CLUES;
int previousMode;
int messageMenuFocus = MESSAGE_FOCUS_INBOX;
int helpFocus;
int winnerFocus = WINNER_FOCUS_1;

int languageX = 0;
int genreX = 0;
int selectGameY = 30;
int groupX = 0;
int duringGameX = 0;
int gameMenuSelectY = 35;
int messageMenuY = 40;
int helpY = 0;
int winnerX = 0;

```

```

int easeSpeed = 5;

String render_text="";

/*
void keyPressed(){
  if(mode == MODE_LANGUAGE_MENU){ // if we are at the language menu...
    if(keyCode == RIGHT){ // if the user pressed right...
      image(rightArrow, 6, 0, 6, 12, 165, 96);
    }
    else if(keyCode == LEFT){ // if the user pressed left...
      image(leftArrow, 6, 0, 6, 12, 5, 96);
    }
  }
}
*/

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

void draw() // happens repeatedly (according to framerate)...
{
  println("MODE: " + mode);
  if(mode == MODE_LOGO){ // if we are in "Logo" mode...
    //drawLogo(); // draw image for Logo
    drawStatic(logo);
  }
  else if(mode == MODE_LANGUAGE_MENU){ // if we are in "Language Menu" mode...
    drawLanguageMenu(menuFocus); // draw image for the language according to the "menuFocus"
  }
  else if(mode == MODE_WELCOME){ //if we are in mode_welcome
    drawStatic(welcome); //show the welcome image
  }
  else if(mode == MODE_WELCOME_2){ //if we are in mode_welcome_2
    drawStatic(welcome2); //show the welcome2 image
  }
  else if(mode == MODE_SELECT_GENRE){
    drawGenreSelection(genreFocus);
  }
  else if(mode == MODE_SELECT_GAME){
    drawSelectGame(gameFocus);
  }
  else if(mode == MODE_SELECT_GROUP){
    drawSelectGroup(groupFocus);
  }
  else if(mode == MODE_DESCRIPTION){
    if(descriptionFocus == DESCRIPTION_FOCUS_HOBBITS){
      drawStatic(hobbitsDescription);
    }
    /* DA INSERIRE LE IMMAGINI
    else if(descriptionFocus == DESCRIPTION_FOCUS_ORCS){
      drawStatic(orcsDescription);
    }
    else if(descriptionFocus == DESCRIPTION_FOCUS_ELVES){
      drawStatic(elvessDescription);
    }
    else if(descriptionFocus == DESCRIPTION_FOCUS_DEMONS){
      drawStatic(demonsDescription);
    }
    */
  }
  else if(mode == MODE_ROLE){
    if(roleFocus == ROLE_FOCUS_HOBBITS){
      drawStatic(hobbitsRole);
    }
    /* DA INSERIRE LE IMMAGINI
    else if(roleFocus == ROLE_FOCUS_ORCS){
      drawStatic(orcsRole);
    }
    else if(roleFocus == ROLE_FOCUS_ELVES){
      drawStatic(elvesRole);
    }
    else if(roleFocus == ROLE_FOCUS_DEMONS){
      drawStatic(demonsRole);
    }
    */
  }
}

```

```

else if(mode == MODE_CONFIRM){
    drawStatic(confirm);
}
else if(mode == MODE_GAME_STARTED){
    drawStatic(gameStarted);
}
else if(mode == MODE_HOBBITS_1ST_CLUE){
    drawStatic(hobbits1stClue);
}
else if(mode == MODE_DURING_GAME){
    drawDuringGame(duringGameFocus);
}
else if(mode == MODE_MENU){
    drawMenu(gameMenuFocus);
}
else if(mode == MODE_SHOW_CLUES){
    drawStatic(cluesMap);
}
else if(mode == MODE_SHOW_MESSAGES){
    drawMessageMenu(messageMenuFocus);
}
else if(mode == MODE_SHOW_HELP){
    drawHelp(helpFocus);
}
else if(mode == MODE_QUIT){
    drawStatic(quit);
}
else if(mode == MODE_SEMACODE){
    showCamera(5, 23, 166, 116, sendSemacode);
}
else if(mode == MODE_HOBBITS_1ST_COMPLETE){
    drawStatic(hobbits1stComplete);
}
else if(mode == MODE_DURING_GAME2){
    drawDuringGame(duringGameFocus);
}
else if(mode == MODE_HOBBITS_2ND_FAILED){
    drawStatic(hobbits2ndFailed);
}
else if(mode == MODE_MINUTES_LEFT){
    drawStatic(minutesLeft);
}
else if(mode == MODE_TIME_UP){
    drawStatic(timeUp);
}
else if(mode == MODE_WINNER){
    drawWinner(winnerFocus);
}
else if(mode == MODE_SEND_MESSAGE){
    drawMessage();
}
}

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

void keyReleased() // whenever a key is pressed...
{
    //we are in the logo screen-----
    if(mode == MODE_LOGO){
        if(keyCode == SOFTKEY2){
            mode = MODE_LANGUAGE_MENU;
        }
    }
    //we are in the language selection screen-----
    else if(mode == MODE_LANGUAGE_MENU){ // if we are at the language menu...
        if(menuFocus == LANGUAGE_FOCUS_ITALIAN){ // if the menu is focused on ITALIAN...
            if(keyCode == RIGHT){ // if the user pressed right...
                fruscio.play();
                menuFocus = LANGUAGE_FOCUS_ENGLISH; // put the menu focus on ENGLISH
            }
            else if(keyCode == LEFT){ // if the user pressed left...
                fruscio.play();
                menuFocus = LANGUAGE_FOCUS_FRENCH; // put the menu focus on FRENCH
            }
        }
    }
    else if(menuFocus == LANGUAGE_FOCUS_FRENCH){ // if the menu is focused on FRENCH..
        if(keyCode == RIGHT){ // if the user pressed right...

```

```

    fruscio.play();
    menuFocus = LANGUAGE_FOCUS_ITALIAN; // put the menu focus on ITALIAN
}
else if(keyCode == LEFT){ // if the user pressed left...
    fruscio.play();
    menuFocus = LANGUAGE_FOCUS_ENGLISH; // put the menu focus on ENGLISH
}
}
else if(menuFocus == LANGUAGE_FOCUS_ENGLISH){
    if(keyCode == SOFTKEY2){
        mode = MODE_WELCOME; //go to the welcome screen
    } // if the menu is focused on ENGLISH...
    else if(keyCode == RIGHT){ // if the user pressed right...
        fruscio.play();
        menuFocus = LANGUAGE_FOCUS_FRENCH; // put the menu focus on FRENCH
    }
    else if(keyCode == LEFT){ // if the user pressed left...
        fruscio.play();
        menuFocus = LANGUAGE_FOCUS_ITALIAN; // put the menu focus on ITALIAN
    }
}
}
}
//we are in welcome screen-----
else if(mode == MODE_WELCOME) { //if we are in the welcome screen...
    if(keyCode == SOFTKEY2){ //and we press fire or 3
        mode = MODE_WELCOME_2; //show the 2nd welcome screen
    }
    else if(keyCode == SOFTKEY1){ // if the user pressed 1...
        mode = MODE_LANGUAGE_MENU; //back in the language selection
        /*
        menuFocus = LANGUAGE_FOCUS_ITALIAN; // put the menu focus on ITALIAN
        languageX = 0;
        */
    }
}
//we are in welcome 2 screen-----
else if(mode == MODE_WELCOME_2) { //if we are in the welcome screen...
    if(keyCode == SOFTKEY2){ //and we press fire or 3
        mode = MODE_SELECT_GENRE; //show the 2nd welcome screen
    }
    else if(keyCode == SOFTKEY1){ // if the user pressed 1...
        mode = MODE_WELCOME; //back in the language selection
    }
}
//we are in genre selection screen-----
else if(mode == MODE_SELECT_GENRE) { //if we are in the welcome 2 screen
    if(genreFocus == GENRE_FOCUS_FANTASY){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            mode = MODE_SELECT_GAME; //go to the select game screen
        }
        else if(keyCode == RIGHT){ // if the user pressed right...
            fruscio.play();
            genreFocus = GENRE_FOCUS_TVANDMOVIES; // put the menu focus on ENGLISH
        }
        else if(keyCode == LEFT){ // if the user pressed left...
            fruscio.play();
            genreFocus = GENRE_FOCUS_TRADITIONALMASKS; // put the menu focus on FRENCH
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_WELCOME_2; //go to the select game screen
        }
    }
}
else if(genreFocus == GENRE_FOCUS_TVANDMOVIES){ // if the menu is focused on ITALIAN...
    if(keyCode == RIGHT){ // if the user pressed right...
        fruscio.play();
        genreFocus = GENRE_FOCUS_SUPERHEROES; // put the menu focus on ENGLISH
    }
    else if(keyCode == LEFT){ // if the user pressed left...
        fruscio.play();
        genreFocus = GENRE_FOCUS_FANTASY; // put the menu focus on FRENCH
    }
    else if(keyCode == SOFTKEY1){
        mode = MODE_WELCOME_2; //go to the select game screen
    }
}
}
else if(genreFocus == GENRE_FOCUS_SUPERHEROES){ // if the menu is focused on ITALIAN...
    if(keyCode == RIGHT){ // if the user pressed right...
        fruscio.play();
        genreFocus = GENRE_FOCUS_TRADITIONALMASKS; // put the menu focus on ENGLISH
    }
}
}

```

```

    }
    else if(keyCode == LEFT){ // if the user pressed left...
        fruscio.play();
        genreFocus = GENRE_FOCUS_TVANDMOVIES; // put the menu focus on FRENCH
    }
    else if(keyCode == SOFTKEY1){
        mode = MODE_WELCOME_2; //go to the select game screen
    }
}
else if(genreFocus == GENRE_FOCUS_TRADITIONMasks){ // if the menu is focused on ITALIAN...
    if(keyCode == RIGHT){ // if the user pressed right...
        fruscio.play();
        genreFocus = GENRE_FOCUS_FANTASY; // put the menu focus on ENGLISH
    }
    else if(keyCode == LEFT){ // if the user pressed left...
        fruscio.play();
        genreFocus = GENRE_FOCUS_SUPERHEROES; // put the menu focus on FRENCH
    }
    else if(keyCode == SOFTKEY1){
        mode = MODE_WELCOME_2; //go to the select game screen
    }
}
}
//we are in select game screen-----
else if(mode == MODE_SELECT_GAME){
    if(gameFocus == GAME_FOCUS_1){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            mode = MODE_SELECT_GROUP; //go to the select game screen
        }
        else if(keyCode == DOWN){ // if the user pressed right...
            gameFocus = GAME_FOCUS_2; // put the menu focus on ENGLISH
        }
        else if(keyCode == UP){ // if the user pressed left...
            gameFocus = GAME_FOCUS_1; // put the menu focus on FRENCH
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_SELECT_GENRE; //go to the select game screen
        }
    }
    else if(gameFocus == GAME_FOCUS_2){ // if the menu is focused on ITALIAN...
        if(keyCode == DOWN){ // if the user pressed right...
            gameFocus = GAME_FOCUS_3; // put the menu focus on ENGLISH
        }
        else if(keyCode == UP){ // if the user pressed left...
            gameFocus = GAME_FOCUS_1; // put the menu focus on FRENCH
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_SELECT_GENRE; //go to the select game screen
        }
    }
    else if(gameFocus == GAME_FOCUS_3){ // if the menu is focused on ITALIAN...
        if(keyCode == DOWN){ // if the user pressed right...
            gameFocus = GAME_FOCUS_3; // put the menu focus on ENGLISH
        }
        else if(keyCode == UP){ // if the user pressed left...
            gameFocus = GAME_FOCUS_2; // put the menu focus on FRENCH
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_SELECT_GENRE; //go to the select game screen
        }
    }
}
//we are in select group screen-----
else if(mode == MODE_SELECT_GROUP){
    if(groupFocus == GROUP_FOCUS_HOBBITS){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            mode = MODE_DESCRIPTION; //go to the select game screen
        }
        else if(keyCode == RIGHT){ // if the user pressed right...
            fruscio.play();
            groupFocus = GROUP_FOCUS_ORCS; // put the menu focus on ENGLISH
        }
        else if(keyCode == LEFT){ // if the user pressed left...
            fruscio.play();
            groupFocus = GROUP_FOCUS_DEMONS; // put the menu focus on FRENCH
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_SELECT_GAME; //go to the select game screen
        }
    }
}

```

```

}
else if(groupFocus == GROUP_FOCUS_ORCS){ // if the menu is focused on ITALIAN...
if(keyCode == RIGHT){ // if the user pressed right...
fruscio.play();
groupFocus = GROUP_FOCUS_ELVES; // put the menu focus on ENGLISH
}
else if(keyCode == LEFT){ // if the user pressed left...
fruscio.play();
groupFocus = GROUP_FOCUS_HOBBITS; // put the menu focus on FRENCH
}
else if(keyCode == SOFTKEY1){
mode = MODE_SELECT_GAME; //go to the select game screen
}
}
else if(groupFocus == GROUP_FOCUS_ELVES){ // if the menu is focused on ITALIAN...
if(keyCode == RIGHT){ // if the user pressed right...
fruscio.play();
groupFocus = GROUP_FOCUS_DEMONS; // put the menu focus on ENGLISH
}
else if(keyCode == LEFT){ // if the user pressed left...
fruscio.play();
groupFocus = GROUP_FOCUS_ORCS; // put the menu focus on FRENCH
}
else if(keyCode == SOFTKEY1){
mode = MODE_SELECT_GAME; //go to the select game screen
}
}
else if(groupFocus == GROUP_FOCUS_DEMONS){ // if the menu is focused on ITALIAN...
if(keyCode == RIGHT){ // if the user pressed right...
fruscio.play();
groupFocus = GROUP_FOCUS_HOBBITS; // put the menu focus on ENGLISH
}
else if(keyCode == LEFT){ // if the user pressed left...
fruscio.play();
groupFocus = GROUP_FOCUS_ELVES; // put the menu focus on FRENCH
}
else if(keyCode == SOFTKEY1){
mode = MODE_SELECT_GAME; //go to the select game screen
}
}
}
//we are in group description screen-----
else if(mode == MODE_DESCRIPTION){
if(descriptionFocus == DESCRIPTION_FOCUS_HOBBITS){ // if the menu is focused on ITALIAN...
if(keyCode == SOFTKEY2){
mode = MODE_ROLE;
roleFocus = ROLE_FOCUS_HOBBITS;
//go to the select game screen
}
else if(keyCode == SOFTKEY1){
mode = MODE_SELECT_GROUP; //go to the select game screen
}
}
else if(descriptionFocus == DESCRIPTION_FOCUS_ORCS){ // if the menu is focused on ITALIAN...
if(keyCode == SOFTKEY2){
mode = MODE_ROLE;
roleFocus = ROLE_FOCUS_ORCS;
//go to the select game screen
}
else if(keyCode == SOFTKEY1){
mode = MODE_SELECT_GROUP; //go to the select game screen
}
}
}
else if(descriptionFocus == DESCRIPTION_FOCUS_ELVES){ // if the menu is focused on ITALIAN...
if(keyCode == SOFTKEY2){
mode = MODE_ROLE;
roleFocus = ROLE_FOCUS_ELVES;
//go to the select game screen
}
else if(keyCode == SOFTKEY1){
mode = MODE_SELECT_GROUP; //go to the select game screen
}
}
}
else if(descriptionFocus == DESCRIPTION_FOCUS_DEMONS){ // if the menu is focused on ITALIAN...
if(keyCode == SOFTKEY2){
mode = MODE_ROLE;
roleFocus = ROLE_FOCUS_DEMONS;
//go to the select game screen
}
}
}

```



```

        else if(keyCode == SOFTKEY1){
            mode = MODE_SELECT_GROUP;                //go to the select game screen
        }
    }
}
//we are in role screen-----
else if(mode == MODE_ROLE){
    if(roleFocus == ROLE_FOCUS_HOBBITS){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            mode = MODE_CONFIRM;
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_DESCRIPTION;                //go to the select game screen
        }
    }
    else if(roleFocus == ROLE_FOCUS_ORCS){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            mode = MODE_CONFIRM;
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_DESCRIPTION;                //go to the select game screen
        }
    }
    else if(roleFocus == ROLE_FOCUS_ELVES){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            mode = MODE_CONFIRM;
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_DESCRIPTION;                //go to the select game screen
        }
    }
    else if(roleFocus == ROLE_FOCUS_DEMONS){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            mode = MODE_CONFIRM;
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_DESCRIPTION;                //go to the select game screen
        }
    }
}
//we are in the confirm screen-----
else if(mode == MODE_CONFIRM){
    if(keyCode == SOFTKEY2){
        mode = MODE_GAME_STARTED;
    }
    if(keyCode == SOFTKEY1){
        mode = MODE_ROLE;                //go to the select game screen
    }
}
//we have started the game-----
else if(mode == MODE_GAME_STARTED){
    if(keyCode == SOFTKEY2){
        mode = MODE_HOBBITS_1ST_CLUE;
    }
    else if(keyCode == FIRE){
        previousMode = MODE_GAME_STARTED;
        mode = MODE_MENU;                //go to the select game screen
    }
}
//we show the first hobbits clue-----
else if(mode == MODE_HOBBITS_1ST_CLUE){
    if(keyCode == SOFTKEY2){
        mode = MODE_DURING_GAME;
    }
    else if(keyCode == FIRE){
        previousMode = MODE_HOBBITS_1ST_CLUE;
        mode = MODE_MENU;                //go to the select game screen
    }
    else if(keyCode == SOFTKEY1){
        mode = MODE_DURING_GAME;
    }
}
//we are in the during game screen
else if(mode == MODE_DURING_GAME){
    if(duringGameFocus == DURING_GAME_FOCUS_1){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            //showCamera(5, 23, 166, 116, sendSemacode);
            mode = MODE_SEMACODE;                //go to the select game screen
        }
    }
}

```

```

else if(keyCode == RIGHT){          // if the user pressed right...
    fruscio.play();
    duringGameFocus = DURING_GAME_FOCUS_2; // put the menu focus on ENGLISH
}
else if(keyCode == LEFT){          // if the user pressed left...
    fruscio.play();
    duringGameFocus = DURING_GAME_FOCUS_3; // put the menu focus on FRENCH
}
else if(keyCode == FIRE){
    previousMode = MODE_DURING_GAME;
    mode = MODE_MENU;              //go to the select game screen
}
if(keyCode == SOFTKEY1){
    mode = MODE_HOBBITS_1ST_CLUE;          //go to the select game screen
}
}
else if(duringGameFocus == DURING_GAME_FOCUS_2){ // if the menu is focused on ITALIAN...
    if(keyCode == SOFTKEY2){
        mode = MODE_SEMACODE;            //go to the select game screen
    }
    else if(keyCode == RIGHT){          // if the user pressed right...
        fruscio.play();
        duringGameFocus = DURING_GAME_FOCUS_3; // put the menu focus on ENGLISH
    }
    else if(keyCode == LEFT){          // if the user pressed left...
        fruscio.play();
        duringGameFocus = DURING_GAME_FOCUS_1; // put the menu focus on FRENCH
    }
    else if(keyCode == FIRE){
        previousMode = MODE_DURING_GAME;
        mode = MODE_MENU;              //go to the select game screen
    }
}
else if(duringGameFocus == DURING_GAME_FOCUS_3){ // if the menu is focused on ITALIAN...
    if(keyCode == SOFTKEY2){
        mode = MODE_SEMACODE;            //go to the select game screen
    }
    else if(keyCode == RIGHT){          // if the user pressed right...
        fruscio.play();
        duringGameFocus = DURING_GAME_FOCUS_1; // put the menu focus on ENGLISH
    }
    else if(keyCode == LEFT){          // if the user pressed left...
        fruscio.play();
        duringGameFocus = DURING_GAME_FOCUS_2; // put the menu focus on FRENCH
    }
    else if(keyCode == FIRE){
        previousMode = MODE_DURING_GAME;
        mode = MODE_MENU;              //go to the select game screen
    }
}
}
}
//we are in the menu screen-----
else if(mode == MODE_MENU){
    if(gameMenuFocus == GAME_MENU_FOCUS_CLUES){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            mode = MODE_SHOW_CLUES;          //go to the select game screen
        }
        else if(keyCode == DOWN){          // if the user pressed right...
            gameMenuFocus = GAME_MENU_FOCUS_MESSAGES; // put the menu focus on ENGLISH
        }
        else if(keyCode == UP){          // if the user pressed left...
            gameMenuFocus = GAME_MENU_FOCUS_QUIT; // put the menu focus on FRENCH
        }
        else if(keyCode == SOFTKEY1){
            mode = previousMode;            //go to the select game screen
        }
    }
}
else if(gameMenuFocus == GAME_MENU_FOCUS_MESSAGES){ // if the menu is focused on ITALIAN...
    if(keyCode == SOFTKEY2){
        mode = MODE_SHOW_MESSAGES;          //go to the select game screen
    }
    else if(keyCode == DOWN){          // if the user pressed right...
        gameMenuFocus = GAME_MENU_FOCUS_HELP; // put the menu focus on ENGLISH
    }
    else if(keyCode == UP){          // if the user pressed left...
        gameMenuFocus = GAME_MENU_FOCUS_CLUES; // put the menu focus on FRENCH
    }
}
else if(keyCode == SOFTKEY1){
    mode = previousMode;            //go to the select game screen
}
}
}

```

```

    }
}
else if(gameMenuFocus == GAME_MENU_FOCUS_HELP){ // if the menu is focused on ITALIAN...
    if(keyCode == SOFTKEY2){
        mode = MODE_SHOW_HELP; //go to the select game screen
    }
    else if(keyCode == DOWN){ // if the user pressed right...
        gameMenuFocus = GAME_MENU_FOCUS_QUIT; // put the menu focus on ENGLISH
    }
    else if(keyCode == UP){ // if the user pressed left...
        gameMenuFocus = GAME_MENU_FOCUS_MESSAGES; // put the menu focus on FRENCH
    }
    else if(keyCode == SOFTKEY1){
        mode = previousMode; //go to the select game screen
    }
}
else if(gameMenuFocus == GAME_MENU_FOCUS_QUIT){ // if the menu is focused on ITALIAN...
    if(keyCode == SOFTKEY2){
        mode = MODE_QUIT; //go to the select game screen
    }
    else if(keyCode == DOWN){ // if the user pressed right...
        gameMenuFocus = GAME_MENU_FOCUS_CLUES; // put the menu focus on ENGLISH
    }
    else if(keyCode == UP){ // if the user pressed left...
        gameMenuFocus = GAME_MENU_FOCUS_HELP; // put the menu focus on FRENCH
    }
    else if(keyCode == SOFTKEY1){
        mode = previousMode; //go to the select game screen
    }
}
}
//we are in the clues screen-----
else if(mode == MODE_SHOW_CLUES){
    if(keyCode == SOFTKEY2){
        mode = MODE_MENU; //go to the select game screen
    }
    else if(keyCode == SOFTKEY1){
        mode = previousMode; //go to the select game screen
    }
}
//we are int the message menu-----
else if(mode == MODE_SHOW_MESSAGES){
    if(messageMenuFocus == MESSAGE_FOCUS_SEND){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            mode = MODE_SEND_MESSAGE; //go to the select game screen
        }
        else if(keyCode == DOWN){
            messageMenuFocus = MESSAGE_FOCUS_INBOX;
        }
        else if(keyCode == UP){
            messageMenuFocus = MESSAGE_FOCUS_INBOX;
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_MENU;
        }
    }
    else if(messageMenuFocus == MESSAGE_FOCUS_INBOX){
        if(keyCode == SOFTKEY2){
            mode = MODE_INBOX; //go to the select game screen
        }
        else if(keyCode == DOWN){
            messageMenuFocus = MESSAGE_FOCUS_SEND;
        }
        else if(keyCode == UP){
            messageMenuFocus = MESSAGE_FOCUS_SEND;
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_MENU;
        }
    }
}
}
//show the compose message screen-----
else if(mode == MODE_SEND_MESSAGE){
    String user_text = textInput("Message", "", 160);
    if(user_text != null){ // Check to make sure text is not null or will cause an exception
        render_text = user_text;
    }
}
}
//show the help screen-----

```

```

else if(mode == MODE_SHOW_HELP){
    if(helpFocus == HELP_FOCUS_PAGE1){
        if(keyCode == SOFTKEY2){
            mode = MODE_MENU;                //go to the select game screen
        }
        else if(keyCode == DOWN){
            helpFocus = HELP_FOCUS_PAGE2;
        }
        else if(keyCode == UP){
            helpFocus = HELP_FOCUS_PAGE1;
        }
        else if(keyCode == SOFTKEY1){
            mode = previousMode;
        }
    }
}
else if(helpFocus == HELP_FOCUS_PAGE2){
    if(keyCode == SOFTKEY2){
        mode = MODE_MENU;                //go to the select game screen
    }
    else if(keyCode == DOWN){
        helpFocus = HELP_FOCUS_PAGE2;
    }
    else if(keyCode == UP){
        helpFocus = HELP_FOCUS_PAGE1;
    }
    else if(keyCode == SOFTKEY1){
        mode = previousMode;
    }
}
}
//we are in the quit confirmation menu-----
else if(mode == MODE_QUIT){
    if(keyCode == SOFTKEY2){
        exit();
    }
    else if(keyCode == SOFTKEY1){
        mode = MODE_MENU;
    }
}
//we are in the semacode shoot screen-----
else if(mode == MODE_SEMACODE){
    if(keyCode == SOFTKEY2){
        hideCamera();
        mode = MODE_HOBBITS_1ST_COMPLETE;    //insert the code to quit application
    }
    else if(keyCode == FIRE){
        hideCamera();
        previousMode = MODE_SEMACODE;
        mode = MODE_MENU;                //go to the select game screen
    }
}
//we show the current player positions-----
else if(mode == MODE_HOBBITS_1ST_COMPLETE){
    if(keyCode == SOFTKEY2){
        mode = MODE_DURING_GAME2;    //insert the code to quit application
        duringGameFocus = DURING_GAME_FOCUS_1;
    }
}
//during game screen-----
else if(mode == MODE_DURING_GAME2){
    if(duringGameFocus == DURING_GAME_FOCUS_1){ // if the menu is focused on ITALIAN...
        if(keyCode == SOFTKEY2){
            mode = MODE_HOBBITS_2ND_FAILED;    //insert the code to quit application
        }
        else if(keyCode == RIGHT){ // if the user pressed right...
            fruscio.play();
            duringGameFocus = DURING_GAME_FOCUS_2; // put the menu focus on ENGLISH
        }
        else if(keyCode == LEFT){ // if the user pressed left...
            fruscio.play();
            duringGameFocus = DURING_GAME_FOCUS_3; // put the menu focus on FRENC
        }
        else if(keyCode == FIRE){
            previousMode = MODE_DURING_GAME2;
            mode = MODE_MENU;                //go to the select game screen
        }
        else if(keyCode == SOFTKEY1){
            mode = MODE_HOBBITS_1ST_COMPLETE;
        }
    }
}

```

```

}
else if(duringGameFocus == DURING_GAME_FOCUS_2){ // if the menu is focused on ITALIAN...
    if(keyCode == SOFTKEY2){
        mode = MODE_HOBBITS_2ND_FAILED; //insert the code to quit application
    }
    else if(keyCode == RIGHT){ // if the user pressed right...
        fruscio.play();
        duringGameFocus = DURING_GAME_FOCUS_3; // put the menu focus on ENGLISH
    }
    else if(keyCode == LEFT){ // if the user pressed left...
        fruscio.play();
        duringGameFocus = DURING_GAME_FOCUS_1; // put the menu focus on FRENCH
    }
    else if(keyCode == FIRE){
        previousMode = MODE_DURING_GAME2;
        mode = MODE_MENU; //go to the select game screen
    }
    else if(keyCode == SOFTKEY1){
        mode = MODE_HOBBITS_1ST_COMPLETE;
    }
}
else if(duringGameFocus == DURING_GAME_FOCUS_3){ // if the menu is focused on ITALIAN...
    if(keyCode == SOFTKEY2){
        mode = MODE_HOBBITS_2ND_FAILED; //insert the code to quit application
    }
    else if(keyCode == RIGHT){ // if the user pressed right...
        fruscio.play();
        duringGameFocus = DURING_GAME_FOCUS_1; // put the menu focus on ENGLISH
    }
    else if(keyCode == LEFT){ // if the user pressed left...
        fruscio.play();
        duringGameFocus = DURING_GAME_FOCUS_2; // put the menu focus on FRENCH
    }
    else if(keyCode == FIRE){
        previousMode = MODE_DURING_GAME2;
        mode = MODE_MENU; //go to the select game screen
    }
    else if(keyCode == SOFTKEY1){
        mode = MODE_HOBBITS_1ST_COMPLETE;
    }
}
}
else if(mode == MODE_HOBBITS_2ND_FAILED){
    if(keyCode == SOFTKEY2){
        mode = MODE_MINUTES_LEFT; //insert the code to quit application
    }
}
else if(mode == MODE_MINUTES_LEFT){
    if(keyCode == SOFTKEY2){
        mode = MODE_TIME_UP; //insert the code to quit application
    }
    else if(keyCode == FIRE){
        previousMode = MODE_MINUTES_LEFT;
        mode = MODE_MENU; //go to the select game screen
    }
}
else if(mode == MODE_TIME_UP){
    if(keyCode == SOFTKEY2){
        mode = MODE_WINNER; //insert the code to quit application
    }
}
else if(mode == MODE_WINNER){
    if (winnerFocus == WINNER_FOCUS_1){
        if(keyCode == SOFTKEY2){
            mode = MODE_LOGO; //insert the code to quit application
        }
        else if(keyCode == RIGHT){
            fruscio.play();
            winnerFocus = WINNER_FOCUS_2;
        }
        else if(keyCode == LEFT){
            fruscio.play();
            winnerFocus = WINNER_FOCUS_3;
        }
    }
}
else if (winnerFocus == WINNER_FOCUS_2){
    if(keyCode == SOFTKEY2){
        mode = MODE_LOGO; //insert the code to quit application
    }
}
}

```

```
else if(keyCode == RIGHT){
    fruscio.play();
    winnerFocus = WINNER_FOCUS_3;
}
else if(keyCode == LEFT){
    fruscio.play();
    winnerFocus = WINNER_FOCUS_1;
}
}
else if (winnerFocus == WINNER_FOCUS_3){
    if(keyCode == SOFTKEY2){
        mode = MODE_LOGO;        //insert the code to quit application
    }
    else if(keyCode == RIGHT){
        fruscio.play();
        winnerFocus = WINNER_FOCUS_1;
    }
    else if(keyCode == LEFT){
        fruscio.play();
        winnerFocus = WINNER_FOCUS_2;
    }
}
}
}
```