

/\*\*

**Ready? Set... Go!**

**Mobile Processing Code**

Silvia Boscolo . Maria Gabriella Astolfo . Erika Rossi

thanks to Vinay Venkatraman and Nicholas Zambetti

\*/

/\*\*

\* Logic Section - Code that captures and interprets user input

\*/

////////////////////////////////////

// Libraries - Additional libraries that your program requires

////////////////////////////////////

import processing.sound.\*;

Sound s;

import processing.phone.\*;

Phone myPhone; // Named reference to your phone

////////////////////////////////////

// Modes - Named states of the program to make code more readable (Constants)

////////////////////////////////////

// Names for each possible screen

int SCREEN\_LOGO =0;

int SCREEN\_STUFF = 1;

int SCREEN\_HELP = 2;

int SCREEN\_SETTING =3;

int SCREEN\_AIM =4;

int SCREEN\_MERMAID =5;

int SCREEN\_ALIEN =6;

int SCREEN\_PIRATE =7;

int SCREEN\_NEW =8;

int SCREEN\_CHOOSE =9;

int SCREEN\_INTROMERMAID =10;

int SCREEN\_INTROMERMAID2 =11;

int SCREEN\_OBJECTS =12;

int SCREEN\_STARTMERMAID=13;

int SCREEN\_LANGUAGE =14;

int SCREEN\_SANMARCO =15;

```
int SCREEN_INTROSANMARCO =16;
int SCREEN_SANMARCOQ1 =17;
int SCREEN_WORKINPROGRESS =18;
int SCREEN_STANDBY =19;
int SCREEN_STARTMERMAID2 =20;
int SCREEN_EXCELLENT=21;
int SCREEN_ACTIVITY1=22;
int SCREEN_EXCELLENT1=23;
int SCREEN_ERROR=24;
int SCREEN_CODE=25;
int SCREEN_COMPLETED =26;
int SCREEN_ALLOBJECTS =27;
int SCREEN_REPEAT =28;
int SCREEN_RECOVER =29;

int PREV_MODE = 0;

// Names for each possible menu focus selection
int STUFF_NEW = 0;
int STUFF_RESUME = 1;
int STUFF_SAVE = 2;
int STUFF_HELP = 3;
int STUFF_SETTING = 4;
int STUFF_QUIT = 5;

int LANGUAGE_FRENCH =0;
int LANGUAGE_SPANISH =1;
int LANGUAGE_ITALIAN =2;
int LANGUAGE_ENGLISH =3;
int LANGUAGE_GERMAN =4;
int LANGUAGE_JAPANESE =5;

int HELP_AIM = 0;
int HELP_RULES = 1;
int HELP_ITEMS = 2;

int AIM_MERMAID =0;
int AIM_ALIEN =1;
int AIM_PIRATE =2;

int SETTING_SOUND = 0;
int SETTING_VIBRATION = 1;
int SETTING_LANGUAGE = 2;
int SETTING_CHARACTER = 3;
int SETTING_BLUETOOTH = 4;
```

```
int SANMARCO_YES =0;
int SANMARCO_NO =1;

int SANMARCOQ1_A = 0;
int SANMARCOQ1_B = 1;
int SANMARCOQ1_C = 2;
int SANMARCOQ1_D = 3;

int REPEAT_YES=0;
int REPEAT_NO=1;

/////////////////////////////////////////////////////////////////
// State - Information collected from use (Variables)
/////////////////////////////////////////////////////////////////

int screenMode =SCREEN_LOGO;
int stuffFocus = STUFF_NEW; // Focus state of the stuff menu
int helpFocus = HELP_AIM; // Focus state of the help menu
int settingFocus = SETTING_SOUND; // Focus state of the help menu
int aimFocus =AIM_MERMAID; // Focus state of the aim menu
int languageFocus =LANGUAGE_FRENCH;
int sanmarcoFocus =SANMARCO_YES;
int sanmarcoq1Focus =SANMARCOQ1_A;
int repeatFocus =REPEAT_YES;

//variabili aggiornamento barra
String puntiJolly = "x0";
String puntiOggetti = "x0";
int n=1;

// Named references to images
PImage sfondo_grigio;
PImage logo;
PImage bandiera;
PImage botton_start;
PImage botton_ok;
PImage stuff_menu;
PImage stuff_fasce;
PImage barra;
PImage help_fasce;
PImage help_menu;
PImage stuff_botton;
PImage setting_sfondo;
PImage setting_testo;
PImage aim_sfondo;
PImage aim_testo;
```

```
PImage mermaid;  
PImage botton_back;  
PImage alien;  
PImage pirate;  
PImage nickname;  
PImage sfondo_viola;  
PImage stuff_ok_viola;  
PImage barra_mermaid;  
PImage testo_intro_mermaid;  
PImage stuff_goon_viola;  
PImage testo_intro_mermaid2;  
PImage mermaid_start_testo;  
PImage sanmarco_yesno;  
PImage yesno;  
PImage barre_yesno;  
PImage selezione_yesno;  
PImage ok_viola;  
PImage introsanmarco;  
PImage sanmarco;  
PImage sanmarcoq1;  
PImage workinprogress;  
PImage standby;  
PImage mermaid_start_testo2;  
PImage mermaid_aim;  
PImage alien_aim;  
PImage pirate_aim;  
PImage sanmarco_barrel;  
PImage sanmarco_selezione1;  
PImage excellent;  
PImage sanmarco_al;  
PImage code;  
PImage codice;  
PImage completed;  
PImage repeat;  
PImage fine;  
PImage back_goon_viola;  
PImage stuff_viola;  
PImage goon_viola;  
PImage error;  
PImage barra_nera;  
PImage sfondo_viola_question;  
PImage sfondo_viola_activity;  
PImage comb_icon;
```

```
// Function to load all the images  
void loadImages()
```

```
{
sfondo_grigio =loadImage ("sfondo_grigio.png");
logo =loadImage ("logo.png");
bandiera = loadImage ("bandiera_animata.png");
botton_start =loadImage ("botton_start.png");
botton_ok =loadImage ("botton_ok.png");
stuff_menu = loadImage("stuff_menu.png");
stuff_fasce = loadImage ("stuff_fasce.png");
barra = loadImage("barra.png");
help_fasce = loadImage ("help_fasce.png");
help_menu = loadImage ("help_menu.png");
stuff_botton =loadImage ("stuff_botton.png");
setting_sfondo = loadImage ("setting_sfondo.png");
setting_testo =loadImage ("setting_testo.png");
aim_sfondo =loadImage ("aim_sfondo.png");
aim_testo =loadImage ("aim_testo.png");
mermaid = loadImage ("mermaid.png");
botton_back = loadImage ("botton_back.png");
alien =loadImage ("alien.png");
pirate =loadImage ("pirate.png");
nickname =loadImage ("nickname.png");
sfondo_viola =loadImage ("sfondo_viola.png");
stuff_ok_viola =loadImage ("stuff_ok_viola.png");
barra_mermaid =loadImage ("barra_mermaid.png");
testo_intro_mermaid =loadImage ("testo_intro_mermaid.png");
stuff_goon_viola =loadImage ("stuff_goon_viola.png");
testo_intro_mermaid2 =loadImage ("testo_intro_mermaid2.png");
mermaid_start_testo =loadImage ("mermaid_start_testo.png");
sanmarco_yesno =loadImage ("sanmarco_yesno.png");
yesno =loadImage ("yesno.png");
barre_yesno =loadImage ("barre_yesno.png");
selezione_yesno =loadImage ("selezione_yesno.png");
ok_viola =loadImage ("ok_viola.png");
introsanmarco =loadImage ("introsanmarco.png");
sanmarco =loadImage ("sanmarco.png");
sanmarcoq1 =loadImage ("sanmarcoq1.png");
workinprogress =loadImage ("workinprogress.png");
standby =loadImage ("standby.png");
mermaid_start_testo2 =loadImage ("mermaid_start_testo2.png");
mermaid_aim =loadImage ("mermaid_aim.png");
alien_aim =loadImage ("alien_aim.png");
pirate_aim =loadImage ("pirate_aim.png");
sanmarco_barrel =loadImage ("sanmarco_barrel.png");
sanmarco_selezione1 =loadImage ("sanmarco_selezione1.png");
excellent =loadImage ("excellent.png");
sanmarco_a1 =loadImage ("sanmarco_a1.png");
```

```
code =loadImage ("code.png");
codice =loadImage ("codice.png");
completed = loadImage ("completed.png");
repeat =loadImage ("repeat.png");
fine =loadImage ("fine.png");
back_goon_viola =loadImage ("back_goon_viola.png");
stuff_viola =loadImage ("stuff_viola.png");
goon_viola =loadImage ("goon_viola.png");
error =loadImage ("error.png");
barra_nera =loadImage ("barra_nera.png");
sfondo_viola_question=loadImage ("sfondo_viola_question.png");
sfondo_viola_activity=loadImage ("sfondo_viola_activity.png");
comb_icon=loadImage ("comb_icon.png");

//////////////////////menu language
flags = new PImage[6];
flags_render = new PImage[6];
flags_big = new PImage[6];
flags_render_big = new PImage[6];

flags[0] = loadImage("gb.png");
flags[1] = loadImage("ge.png");
flags[2] = loadImage("jp.png");
flags[3] = loadImage("fr.png");
flags[4] = loadImage("sp.png");
flags[5] = loadImage("it.png");

flags_big[0] = loadImage("big_gb.png");
flags_big[1] = loadImage("big_ge.png");
flags_big[2] = loadImage("big_jp.png");
flags_big[3] = loadImage("big_fr.png");
flags_big[4] = loadImage("big_sp.png");
flags_big[5] = loadImage("big_it.png");

flags_render[0] = flags[0];
flags_render[1] = flags[1];
flags_render[2] = flags[2];
flags_render[3] = flags[3];
flags_render[4] = flags[4];
flags_render[5] = flags[5];

flags_render_big[0] = flags_big[0];
flags_render_big[1] = flags_big[1];
flags_render_big[2] = flags_big[2];
flags_render_big[3] = flags_big[3];
flags_render_big[4] = flags_big[4];
```

```

flags_render_big[5] = flags_big[5];
//////////////////////////////////// end menu language

////////////////////////////////////menu choose character
character = new PImage[3];
character_render = new PImage[3];
character_big = new PImage[3];
character_render_big = new PImage[3];

sfondo_grigio =loadImage("sfondo_grigio.png");
character[0] = loadImage("mermaid.png");
character[1] = loadImage("alien.png");
character[2] = loadImage("pirate.png");

character_big[0] = loadImage("spotviola.png");
character_big[1] = loadImage("spotazz.png");
character_big[2] = loadImage("spotarancio.png");

character_render[0] = character[0];
character_render[1] = character[1];
character_render[2] = character[2];

character_render_big[0] = character_big[0];
character_render_big[1] = character_big[1];
character_render_big[2] = character_big[2];
////////////////////////////////////end of menu choose character

////////////////////////////////////start of objects menu //////////////////////////////////
objects = new PImage[7];
objects_render = new PImage[7];
objects_big = new PImage[7];
objects_render_big = new PImage[7];

objects[0] = loadImage("shell.png");
objects[1] = loadImage("comb.png");
objects[2] = loadImage("map.png");
objects[3] = loadImage("mirror.png");
objects[4] = loadImage("fish.png");
objects[5] = loadImage("necklace.png");
objects[6] = loadImage("jolly.png");

objects_big[0] = loadImage("big_shell1.png");
objects_big[1] = loadImage("big_comb1.png");
objects_big[2] = loadImage("big_map1.png");
objects_big[3] = loadImage("big_mirror1.png");
objects_big[4] = loadImage("big_fish1.png");

```

```
objects_big[5] = loadImage("big_pearls1.png");
objects_big[6] = loadImage("big_jolly1.png");
```

```
objects_render[0] = objects[0];
objects_render[1] = objects[1];
objects_render[2] = objects[2];
objects_render[3] = objects[3];
objects_render[4] = objects[4];
objects_render[5] = objects[5];
objects_render[6] = objects[6];
```

```
objects_render_big[0] = objects_big[0];
objects_render_big[1] = objects_big[1];
objects_render_big[2] = objects_big[2];
objects_render_big[3] = objects_big[3];
objects_render_big[4] = objects_big[4];
objects_render_big[5] = objects_big[5];
objects_render_big[6] = objects_big[6];
```

```
objects_c = new PImage[7];
objects_render_c = new PImage[7];
objects_big_c = new PImage[7];
objects_render_big_c = new PImage[7];
```

```
objects_c[0] = loadImage("shell_c.png");
objects_c[1] = loadImage("comb_c.png");
objects_c[2] = loadImage("map_c.png");
objects_c[3] = loadImage("mirror_c.png");
objects_c[4] = loadImage("fish_c.png");
objects_c[5] = loadImage("pearls_c.png");
objects_c[6] = loadImage("jolly_c.png");
```

```
objects_big_c[0] = loadImage("big_shell1_c.png");
objects_big_c[1] = loadImage("big_comb1_c.png");
objects_big_c[2] = loadImage("big_map1_c.png");
objects_big_c[3] = loadImage("big_mirror1_c.png");
objects_big_c[4] = loadImage("big_fish1_c.png");
objects_big_c[5] = loadImage("big_pearls1_c.png");
objects_big_c[6] = loadImage("big_jolly1_c.png");
```

```
objects_render_c[0] = objects_c[0];
objects_render_c[1] = objects_c[1];
objects_render_c[2] = objects_c[2];
objects_render_c[3] = objects_c[3];
objects_render_c[4] = objects_c[4];
objects_render_c[5] = objects_c[5];
```

```

objects_render_c[6] = objects_c[6];

objects_render_big_c[0] = objects_big_c[0];
objects_render_big_c[1] = objects_big_c[1];
objects_render_big_c[2] = objects_big_c[2];
objects_render_big_c[3] = objects_big_c[3];
objects_render_big_c[4] = objects_big_c[4];
objects_render_big_c[5] = objects_big_c[5];
objects_render_big_c[6] = objects_big_c[6];

////////////////////////////////////end of objects menu //////////////////////////////////////
}

////////////////////////////////////
// Events - Timed/Triggered events to keep the program running (Execution)
////////////////////////////////////

// Setup - Executes only once, prepares program to run (Logic Initialization)
void setup()
{
    myPhone = new Phone(this); // Creates a phone controller
    myPhone.fullscreen(); // Use the entire screen
    loadImages();
    //sound "You're in ..."
    s = new Sound("mid_warn.wav");

    //nikename//code
    size(176, 208); // Set the screen size for Nokia 6630
    font = loadFont("Verdana-14.mvlw"); // Load the font file from the data folder
    textFont(font); // Use the font for rendering text
    multitap();
}

////////////////////////////////////

void keyPressed()
{
    /* always
    if(keyCode == '5'){
        screenMode = SCREEN_ALLOBJECTS;
    }*/

    // if showing logo screen
    if(screenMode == SCREEN_LOGO){
        keyPressedLogo();
    }
}

```

```
// if showing screen language
else if(screenMode == SCREEN_LANGUAGE){
    keyPressedMenuflag ();
}

// if showing stuff menu
else if(screenMode == SCREEN_STUFF){
    keyPressedStuff();
}

// if showing setting menu
else if(screenMode == SCREEN_SETTING){
    keyPressedSetting();
}

// if showing help menu
else if(screenMode == SCREEN_HELP){
    keyPressedHelp();
}

// if showing aim menu
else if(screenMode == SCREEN_AIM){
    keyPressedAim();
}

// if showing mermaid menu
else if(screenMode == SCREEN_MERMAID){
    keyPressedMermaid();
}

// if showing alien menu
else if(screenMode == SCREEN_ALIEN){
    keyPressedAlien();
}

// if showing pirate menu
else if(screenMode == SCREEN_PIRATE){
    keyPressedPirate();
}

// if showing new (nickname) menu
else if(screenMode == SCREEN_NEW){
    keyPressedNew();
    keyPressedNickname();
}
```

```
// if showing choose menu
else if(screenMode == SCREEN_CHOOSE){
    keyPressedChoose();
    keyPressedMenucharacter();
}

// if showing intro mermaid
else if(screenMode == SCREEN_INTROMERMAID){
    keyPressedIntromermaid();
}

// if showing intro mermaid2
else if(screenMode == SCREEN_INTROMERMAID2){
    keyPressedIntromermaid2();
}

// if showing screen objects
else if(screenMode == SCREEN_OBJECTS){
    keyPressedObjects();
    keyPressedObjectsMenu ();
}

// if showing screen start mermaid
else if(screenMode == SCREEN_STARTMERMAID){
    keyPressedStartmermaid();
}

// if showing screen start mermaid
else if(screenMode == SCREEN_STARTMERMAID2){
    keyPressedStartmermaid2();
}

// if showing screen san marco
else if(screenMode == SCREEN_SANMARCO){
    keyPressedSanmarco();
}

// if showing screen intro san marco
else if(screenMode == SCREEN_INTROSANMARCO){
    keyPressedIntrosanmarco();
}

// if showing screen san marco q1
else if(screenMode == SCREEN_SANMARCOQ1){
    keyPressedSanmarcoq1();
}
```

```
}

// if showing screen work in progress
else if(screenMode == SCREEN_WORKINPROGRESS){
    keyPressedWorkinprogress();
}

// if showing screen standby
else if(screenMode == SCREEN_STANDBY){
    keyPressedStandby();
}

// if showing screen excellent
else if(screenMode == SCREEN_EXCELLENT){
    keyPressedExcellent();
}

// if showing screen activity1
else if(screenMode == SCREEN_ACTIVITY1){
    keyPressedActivity1();
}

// if showing screen activity1
else if(screenMode == SCREEN_EXCELLENT1){
    keyPressedExcellent1();
}

// if showing screen code
else if(screenMode == SCREEN_CODE){
    keyPressedCode();
    //keyPressedCode2();
}

// if showing screen code
else if(screenMode == SCREEN_COMPLETED){
    keyPressedCompleted();
}

// if showing screen code
else if(screenMode == SCREEN_REPEAT){
    keyPressedRepeat();
}

// if showing screen code
else if(screenMode == SCREEN_RECOVER){
```



```
}
// if focused on option B
else if(stuffFocus == STUFF_RESUME){
    // if pressed up
    if(keyCode == UP){
        stuffFocus = STUFF_NEW; // move focus to option A
    }
    // if pressed down
    else if(keyCode == DOWN){
        stuffFocus = STUFF_SAVE; // move focus to option C
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        screenMode = SCREEN_WORKINPROGRESS; // show screen B
    }
}
// if focused on option C
else if(stuffFocus == STUFF_SAVE){
    // if pressed up
    if(keyCode == UP){
        stuffFocus = STUFF_RESUME; // move focus to option B
    }
    // if pressed down
    else if(keyCode == DOWN){
        stuffFocus = STUFF_HELP; // move focus to option D
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        screenMode = SCREEN_WORKINPROGRESS;
    }
}
// if focused on option D
else if(stuffFocus == STUFF_HELP){
    // if pressed up
    if(keyCode == UP){
        stuffFocus = STUFF_SAVE; // move focus to option C
    }
    // if pressed down
    else if(keyCode == DOWN){
        stuffFocus = STUFF_SETTING; // move focus to option E
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        screenMode = SCREEN_HELP; // show screen D
    }
}
}
```

```

// if focused on option E
else if(stuffFocus == STUFF_SETTING){
    // if pressed up
    if(keyCode == UP){
        stuffFocus = STUFF_HELP; // move focus to option D
    }
    // if pressed down
    else if(keyCode == DOWN){
        stuffFocus = STUFF_QUIT; // move focus to option F
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        screenMode = SCREEN_SETTING; // show screen E
    }
}
// if focused on option F
else if(stuffFocus == STUFF_QUIT){
    // if pressed up
    if(keyCode == UP){
        stuffFocus = STUFF_SETTING; // move focus to option E
    }
    // if pressed down
    else if(keyCode == DOWN){
        stuffFocus = STUFF_NEW; // move focus to option A
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        exit();
    }
}
PREV_MODE=SCREEN_STUFF;
}

// if showing nickname screen_new
void keyPressedNew()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_STUFF;
    }
    if (keyCode == SOFTKEY2){
        screenMode = SCREEN_CHOOSE;
    }
    PREV_MODE=SCREEN_NEW;
}

```

```

// if showing screen_choose
void keyPressedChoose()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_NEW;
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        //screenMode = SCREEN_INTROMERMAID;
    }
    PREV_MODE=SCREEN_CHOOSE;
}

```

```

// if showing screen_intromermaid
void keyPressedIntromermaid()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_STUFF;
    }
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_INTROMERMAID2;
    }
    PREV_MODE=SCREEN_INTROMERMAID;
}

```

```

// if showing screen_intromermaid2
void keyPressedIntromermaid2()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_INTROMERMAID;
    }
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_OBJECTS;
    }
    PREV_MODE=SCREEN_INTROMERMAID2;
}

```

```

// if showing screen_objects
void keyPressedObjects()
{
    // if pressed in

```

```

if(keyCode == SOFTKEY1){
    screenMode = SCREEN_INTROMERMAID2;
}
// if pressed in
if(keyCode == SOFTKEY2){
    screenMode = SCREEN_STARTMERMAID;
}
PREV_MODE=SCREEN_OBJECTS;
}

// if showing screen_start mermaid
void keyPressedStartmermaid()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_OBJECTS;
    }
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_STARTMERMAID2;
    }
    PREV_MODE=SCREEN_STARTMERMAID;
}

// if showing screen_start mermaid2
void keyPressedStartmermaid2()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_STARTMERMAID;
    }
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_STANDBY;
    }
    PREV_MODE=SCREEN_STARTMERMAID2;
}

// if showing screen_san marco
void keyPressedSanmarco()
{
    // if focused on option A
    if(sanmarcoFocus == SANMARCO_YES){
        // if pressed up
        if(keyCode == UP){
            sanmarcoFocus = SANMARCO_NO; // move focus to option B

```

```

    println(sanmarcoFocus);
}
// if pressed down
else if(keyCode == DOWN){
    sanmarcoFocus = SANMARCO_NO; // move focus to option B
    println(sanmarcoFocus);
}
else if(keyCode == SOFTKEY1){
    screenMode =SCREEN_STUFF; // show screen B
}
else if(keyCode == SOFTKEY2){
    screenMode =SCREEN_INTROSANMARCO; // show screen B
}
}
// if focused on option B
else if(sanmarcoFocus == SANMARCO_NO){
    // if pressed up
    if(keyCode == UP){
        sanmarcoFocus = SANMARCO_YES; // move focus to option A
    }
    // if pressed down
    else if(keyCode == DOWN){
        sanmarcoFocus = SANMARCO_YES; // move focus to option A
    }
    // if pressed in
    else if(keyCode == SOFTKEY1){
        screenMode =SCREEN_STANDBY; // show screen B
    }
    else if(keyCode == SOFTKEY2){
        screenMode =SCREEN_STANDBY; // show screen B
    }
}
PREV_MODE=SCREEN_SANMARCO;
}

// if showing screen_ intro san marco
void keyPressedIntrosanmarco()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_SANMARCO;
    }
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_SANMARCOQ1;
    }
}

```

```

PREV_MODE=SCREEN_INTROSANMARCO;
}

// if showing help menu
void keyPressedHelp()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_STUFF; // show screen C
    }
    // if focused on option A
    if(helpFocus == HELP_AIM){
        // if pressed up
        if(keyCode == UP){
            helpFocus = HELP_ITEMS; // move focus to option C
            println(helpFocus);
        }
        // if pressed down
        else if(keyCode == DOWN){
            helpFocus = HELP_RULES; // move focus to option B
            println(helpFocus);
        }
        // if pressed in
        else if(keyCode == SOFTKEY2){
            screenMode = SCREEN_AIM; // show screen A
        }
    }
    // if focused on option B
    else if(helpFocus == HELP_RULES){
        // if pressed up
        if(keyCode == UP){
            helpFocus = HELP_AIM; // move focus to option A
        }
        // if pressed down
        else if(keyCode == DOWN){
            helpFocus = HELP_ITEMS; // move focus to option C
        }
        // if pressed in
        else if(keyCode == SOFTKEY2){
            screenMode = SCREEN_WORKINPROGRESS;
        }
    }
    // if focused on option C
    else if(helpFocus == HELP_ITEMS){
        // if pressed up
        if(keyCode == UP){

```

```

    helpFocus = HELP_RULES; // move focus to option B
}
// if pressed down
else if(keyCode == DOWN){
    helpFocus = HELP_AIM; // move focus to option A
}
// if pressed in
else if(keyCode == SOFTKEY2){
    screenMode = SCREEN_WORKINPROGRESS;
}
}
PREV_MODE=SCREEN_HELP;
} // end of Screen help

// if showing aim menu
void keyPressedAim()
{
    // if pressed in softkey1
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_HELP;
    }
    // if focused on option A

    if(aimFocus == AIM_MERMAID){
        // if pressed up
        if(keyCode == UP){
            aimFocus = AIM_PIRATE; // move focus to option C
            println(aimFocus);
        }
        // if pressed down
        else if(keyCode == DOWN){
            aimFocus = AIM_ALIEN; // move focus to option B
            println(aimFocus);
        }
        // if pressed in
        else if(keyCode == SOFTKEY2){
            screenMode = SCREEN_MERMAID; // show screen A
        }
    }
    // if focused on option B
    else if(aimFocus == AIM_ALIEN){
        // if pressed up
        if(keyCode == UP){
            aimFocus = AIM_MERMAID; // move focus to option A
        }
        // if pressed down

```

```

else if(keyCode == DOWN){
    aimFocus = AIM_PIRATE; // move focus to option C
}
// if pressed in
else if(keyCode == SOFTKEY2){
    screenMode = SCREEN_ALIEN; // show screen B
}
}
// if focused on option C
else if(aimFocus == AIM_PIRATE){
    // if pressed up
    if(keyCode == UP){
        aimFocus = AIM_ALIEN; // move focus to option B
    }
    // if pressed down
    else if(keyCode == DOWN){
        aimFocus = AIM_MERMAID; // move focus to option A
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        screenMode = SCREEN_PIRATE; // show screen C
    }
}
}
PREV_MODE=SCREEN_AIM;
} // end of Screen aim

// if showing screen mermaid
void keyPressedMermaid()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_AIM;
    }
    PREV_MODE=SCREEN_MERMAID;
}

// if showing screen alien
void keyPressedAlien()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_AIM;
    }
    PREV_MODE=SCREEN_ALIEN;
}
}

```

```

// if showing screen pirate
void keyPressedPirate()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_AIM;
    }
    PREV_MODE=SCREEN_PIRATE;
}

// if showing setting menu
void keyPressedSetting()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_STUFF; // show screen C
    }
    // if focused on option A
    if(settingFocus == SETTING_SOUND){
        // if pressed up
        if(keyCode == UP){
            settingFocus = SETTING_BLUETOOTH; // move focus to option E
        }
        // if pressed down
        else if(keyCode == DOWN){
            settingFocus = SETTING_VIBRATION; // move focus to option B
        }
        // if pressed in
        else if(keyCode == SOFTKEY2){
            screenMode = SCREEN_WORKINPROGRESS;
        }
    }
    // if focused on option B
    else if(settingFocus == SETTING_VIBRATION){
        // if pressed up
        if(keyCode == UP){
            settingFocus = SETTING_SOUND; // move focus to option A
        }
        // if pressed down
        else if(keyCode == DOWN){
            settingFocus = SETTING_LANGUAGE; // move focus to option C
        }
        // if pressed in
        else if(keyCode == SOFTKEY2){
            screenMode = SCREEN_WORKINPROGRESS;
        }
    }
}

```

```

}
// if focused on option C
else if(settingFocus == SETTING_LANGUAGE){
    // if pressed up
    if(keyCode == UP){
        settingFocus = SETTING_VIBRATION; // move focus to option B
    }
    // if pressed down
    else if(keyCode == DOWN){
        settingFocus = SETTING_CHARACTER; // move focus to option D
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        screenMode = SCREEN_LANGUAGE; // show screen C
    }
}
// if focused on option D
else if(settingFocus == SETTING_CHARACTER){
    // if pressed up
    if(keyCode == UP){
        settingFocus = SETTING_LANGUAGE; // move focus to option C
    }
    // if pressed down
    else if(keyCode == DOWN){
        settingFocus = SETTING_BLUETOOTH; // move focus to option E
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        screenMode = SCREEN_CHOOSE; // show screen D
    }
}
// if focused on option E
else if(settingFocus == SETTING_BLUETOOTH){
    // if pressed up
    if(keyCode == UP){
        settingFocus = SETTING_CHARACTER; // move focus to option D
    }
    // if pressed down
    else if(keyCode == DOWN){
        settingFocus = SETTING_SOUND; // move focus to option A
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        screenMode = SCREEN_WORKINPROGRESS;
    }
}
}

```

```

PREV_MODE=SCREEN_SETTING;
} //end of screen setting

// if showing screen_ san marco q1
void keyPressedSanmarcoq1()
{
  // if focused on option A
  if(sanmarcoq1Focus == SANMARCOQ1_A){
    // if pressed up
    if(keyCode == UP){
      sanmarcoq1Focus = SANMARCOQ1_D; // move focus to option D
    }
    // if pressed down
    else if(keyCode == DOWN){
      sanmarcoq1Focus = SANMARCOQ1_B; // move focus to option A
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
      screenMode = SCREEN_ERROR;
    }
  }
  // if focused on option B
  else if(sanmarcoq1Focus == SANMARCOQ1_B){
    // if pressed up
    if(keyCode == UP){
      sanmarcoq1Focus = SANMARCOQ1_A; // move focus to option D
    }
    // if pressed down
    else if(keyCode == DOWN){
      sanmarcoq1Focus = SANMARCOQ1_C; // move focus to option A
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
      screenMode = SCREEN_EXCELLENT;
    }
  }
  // if focused on option C
  else if(sanmarcoq1Focus == SANMARCOQ1_C){
    // if pressed up
    if(keyCode == UP){
      sanmarcoq1Focus = SANMARCOQ1_B; // move focus to option D
    }
    // if pressed down
    else if(keyCode == DOWN){
      sanmarcoq1Focus = SANMARCOQ1_D; // move focus to option A
    }
    // if pressed in

```

```

else if(keyCode == SOFTKEY2){
    screenMode = SCREEN_ERROR;
}
}
// if focused on option D
else if(sanmarcoq1Focus == SANMARCOQ1_D){
    // if pressed up
    if(keyCode == UP){
        sanmarcoq1Focus = SANMARCOQ1_C; // move focus to option D
    }
    // if pressed down
    else if(keyCode == DOWN){
        sanmarcoq1Focus = SANMARCOQ1_A; // move focus to option A
    }
    // if pressed in
    else if(keyCode == SOFTKEY2){
        screenMode = SCREEN_ERROR;
    }
}
PREV_MODE=SCREEN_SANMARCOQ1;
}
// if showing screen_ work in progress
void keyPressedWorkinprogress()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = PREV_MODE;
    }
    PREV_MODE=PREV_MODE;
}

// if showing screen_ standby
void keyPressedStandby()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_STARTMERMAID2;
    }
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_SANMARCO;
    }
    PREV_MODE=SCREEN_STANDBY;
}

//if showing excellent screen

```

```

void keyPressedExcellent ()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_STUFF;
    }
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_ACTIVITY1;
    }
    PREV_MODE=SCREEN_EXCELLENT;
}

//if showing activity1 screen
void keyPressedActivity1 ()
{
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_EXCELLENT1;
    }
    PREV_MODE=SCREEN_ACTIVITY1;
}

//if showing excellent1 screen
void keyPressedExcellent1 ()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_STUFF;
    }
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_CODE;
    }
    PREV_MODE=SCREEN_EXCELLENT1;
}

//if showing code screen
void keyPressedCode ()
{
    okPuntiOggetti();
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_COMPLETED;
    }
    PREV_MODE=SCREEN_CODE;
}

```

```

}

//if showing completed screen
void keyPressedCompleted ()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_STUFF;
    }
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_ALLOBJECTS;
    }
    PREV_MODE=SCREEN_COMPLETED;
}

//if showing allobjects screen
void keyPressedAllobjects ()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = PREV_MODE;
    }
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_REPEAT;
    }
    PREV_MODE=SCREEN_ALLOBJECTS;
}

//if showing repeat screen
void keyPressedRepeat ()
{
    // if focused on option A
    if(repeatFocus == REPEAT_YES){
        // if pressed up
        if(keyCode == UP){
            repeatFocus = REPEAT_NO; // move focus to option B
            println(repeatFocus);
        }
        // if pressed down
        else if(keyCode == DOWN){
            repeatFocus = REPEAT_NO; // move focus to option B
            println(repeatFocus);
        }
        else if(keyCode == SOFTKEY1){
            screenMode =SCREEN_STUFF; // show screen B

```

```

    }
    else if(keyCode == SOFTKEY2){
        //screenMode =SCREEN_STANDBY; // show screen B
    }
}
// if focused on option B
else if(repeatFocus == REPEAT_NO){
    // if pressed up
    if(keyCode == UP){
        repeatFocus = REPEAT_YES; // move focus to option A
    }
    // if pressed down
    else if(keyCode == DOWN){
        repeatFocus = REPEAT_YES; // move focus to option A
    }
    // if pressed in
    else if(keyCode == SOFTKEY1){
        screenMode =SCREEN_STUFF; // show screen B
    }
    else if(keyCode == SOFTKEY2){
        screenMode =SCREEN_RECOVER; // show screen B
    }
}
PREV_MODE=SCREEN_REPEAT;
}

//if showing recover screen
void keyPressedRecover ()
{
    // if pressed in
    if(keyCode == SOFTKEY1){
        screenMode = SCREEN_STUFF;
    }
    PREV_MODE=SCREEN_RECOVER;
}

//if showing error screen
void keyPressedError ()
{
    // if pressed in
    if(keyCode == SOFTKEY2){
        screenMode = SCREEN_SANMARCOQ1;
    }
    PREV_MODE=SCREEN_ERROR;
}

```

```

/*****
* Graphics Section - Code that provides feedback to the user (behaviors)
*****/

// Function to draw the STUFF SCREEN
int menuFocusYPosition = 50;

// Function to draw LOGO SCREEN
void drawLogoMenu()
{
    //draw menu background
    image (sfondo_grigio, 0, 0);
    // draw menu options text
    image(logo,0,0);
    image (botton_start, 3, height-20);
    drawLogoAnimation();
}

// Function to draw language
void drawLanguageMenu()
{
    //draw menu background
    image (sfondo_grigio, 0, 0);
    // draw menu options text
    image (botton_ok, 3, height-20);
    for(int i=0;i<=5; i++)
    {
        image(flags_render[i], xcor[i], ycor[i]);
        image(flags_render_big[0], 48, 25);
    }
}

void drawStuffMenu()
{
    // draw menu background
    image (sfondo_grigio, 0, 0);
    image (stuff_botton, 3, height-20);
    image (stuff_fasce,0,0);
    // draw focus bar
    if(stuffFocus == STUFF_NEW){
        image(barra, 17, 57);
    }
    else if(stuffFocus == STUFF_RESUME){
        image(barra, 17, 76);
    }
}

```

```

else if(stuffFocus == STUFF_SAVE){
    image(barra, 17, 95);
}
else if(stuffFocus == STUFF_HELP){
    image(barra, 17, 114);
}
else if(stuffFocus == STUFF_SETTING){
    image(barra, 17, 133);
}
else if(stuffFocus == STUFF_QUIT){
    image(barra, 17, 152);
}
// draw menu options text
image(stuff_menu,0, 0);
}

```

// Function to draw HELP SCREEN

```

void drawHelpMenu()
{
    //draw menu background
    image (sfondo_grigio, 0, 0);
    image (stuff_botton, 3, height-20);
    image(help_fasce, 0, 0);
    // draw focus bar
    if(helpFocus == HELP_AIM){
        image(barra, 17, 57);
    }
    else if(helpFocus == HELP_RULES){
        image(barra, 17, 76);
    }
    else if(helpFocus == HELP_ITEMS){
        image(barra, 17, 95);
    }
    // draw menu options text
    image(help_menu,0,0);
}

```

// Function to draw AIM SCREEN

```

void drawAimMenu()
{
    //draw menu background
    image (sfondo_grigio, 0, 0);
    image (stuff_botton, 3, height-20);
    image(aim_sfondo, 0, 0);
    // draw focus bar
    println("focus:"+aimFocus);
}

```

```

if(aimFocus == AIM_MERMAID){
    image(barra, 17, 57);
}
else if(aimFocus == AIM_ALIEN){
    image(barra, 17, 76);
}
else if(aimFocus == AIM_PIRATE){
    image(barra, 17, 95);
}
// draw menu options text
image(aim_testo,0,0);
}

// Function to draw MERMAID SCREEN
void drawMermaidMenu()
{
    //draw menu background
    image (sfondo_grigio, 0, 0);
    image (botton_back, 3, height-20);
    // draw menu options text
    image(mermaid_aim,0,0);
}

// Function to draw ALIEN SCREEN
void drawAlienMenu()
{
    //draw menu background
    image (sfondo_grigio, 0, 0);
    image (botton_back, 3, height-20);
    // draw menu options text
    image(alien_aim,0,0);
}

// Function to draw PIRATE SCREEN
void drawPirateMenu()
{
    //draw menu background
    image (sfondo_grigio, 0, 0);
    image (botton_back, 3, height-20);
    // draw menu options text
    image(pirate_aim,0,0);
}

// Function to draw the SETTING SCREEN
void drawSettingMenu()
{

```

```

// draw menu background
image (sfondo_grigio, 0, 0);
image (stuff_botton, 3, height-20);
image (setting_sfondo,0,0);
// draw focus bar
if(settingFocus == SETTING_SOUND){
    image(barra, 17, 57);
}
else if(settingFocus == SETTING_VIBRATION){
    image(barra, 17, 76);
}
else if(settingFocus == SETTING_LANGUAGE){
    image(barra, 17, 95);
}
else if(settingFocus == SETTING_CHARACTER){
    image(barra, 17, 114);
}
else if(settingFocus == SETTING_BLUETOOTH){
    image(barra, 17, 133);
}
// draw menu options text
image(setting_testo,0, 0);
}

//draw new screen
void drawNewMenu()
{
    //draw menu background
    image (sfondo_grigio, 0, 0);
    // draw menu options text
    image(nickname,0,0);
    image (stuff_botton, 3, height-20);
    drawNickname ();
} // end of new screen

// Function to draw choose_screen
void drawChooseMenu()
{
    //draw menu background
    image (sfondo_grigio,0,0);
    image (stuff_botton, 3, height-20);
    // draw menu options text
    for(int i=0;i<=2; i++)
    {
        image(character_render[i], xcorcharacter[i], ycorcharacter[i]);
        image(character_render_big[0], 36, 15);
    }
}

```

```

    }
}

// Function to draw intromermaid_screen
void drawIntromermaidMenu()
{
    //draw menu background
    image (sfondo_viola,0,0);
    //topbar
    //image (topbar_mermaid, 0,0);
    image (barra_mermaid, 0,0);
    //elementi barra
    textFont (font);
    text (puntiJolly, 70, 30); //punti jolly sulla barra
    text (puntiOggetti, 110, 30); //punti oggetti sulla barra
    // draw menu options text
    image (testo_intro_mermaid,0,0);
    //botton
    image (stuff_goon_viola, 3, height-20);
}

// Function to draw intromermaid2_screen
void drawIntromermaid2Menu()
{
    //draw menu background
    image (sfondo_viola,0,0);
    //topbar
    // image (topbar_mermaid, 0,0);
    image (barra_mermaid, 0,0);
    //elementi barra
    textFont (font);
    text (puntiJolly, 70, 30); //punti jolly sulla barra
    text (puntiOggetti, 110, 30); //punti oggetti sulla barra
    // draw menu options text
    image (testo_intro_mermaid2,0,0);
    drawMermaidIntroAnimation();
    //botton
    image (back_goon_viola, 3, height-20);
}

// Function to draw objects_screen
void drawObjectsMenu()
{
    //draw menu background
    image (sfondo_viola,0,0);
    //topbar

```

```

//image (topbar_mermaid, 0,0);
image (barra_mermaid, 0,0);
//elementi barra
textFont (font);
text (puntiJolly, 70, 30); //punti jolly sulla barra
text (puntiOggetti, 110, 30); //punti oggetti sulla barra
// draw menu options text
for(int i=0;i<=6; i++)
{
    image(objects_render[i], xcorobjects[i], ycorobjects[i]);
    image(objects_render_big[0], 44, 47);
}
//botton
image (back_goon_viola, 3, height-20);
}

```

```

// Function to draw start mermaid_screen

```

```

void drawStartmermaidMenu()
{
    //draw menu background
    image (sfondo_viola,0,0);
    //topbar
    //image (topbar_mermaid, 0,0);
    image (barra_mermaid, 0,0);
    //elementi barra
    textFont (font);
    text (puntiJolly, 70, 30); //punti jolly sulla barra
    text (puntiOggetti, 110, 30); //punti oggetti sulla barra
    // draw menu options text
    image (mermaid_start_testo,0,0);
    //botton
    image (back_goon_viola, 3, height-20);
}

```

```

// Function to draw start mermaid2_screen

```

```

void drawStartmermaid2Menu()
{
    //draw menu background
    image (sfondo_viola,0,0);
    //topbar
    //image (topbar_mermaid, 0,0);
    image (barra_mermaid, 0,0);
    //elementi barra
    textFont (font);
    text (puntiJolly, 70, 30); //punti jolly sulla barra
    text (puntiOggetti, 110, 30); //punti oggetti sulla barra

```

```

// draw menu options text
image (mermaid_start_testo2,0,0);
drawCoda();
//botton
image (back_goon_viola, 3, height-20);
}

// Function to draw san marco screen
boolean drawSanmarcoMenuAlready = false;
void drawSanmarcoMenu()
{
  if(!drawSanmarcoMenuAlready){
    s.play();
    drawSanmarcoMenuAlready = true;
  }
  //draw menu background
  image (sfondo_viola,0,0);
  //topbar
  //image (topbar_mermaid, 0,0);
  image (barra_mermaid, 0,0);
  //elementi barra
  textFont (font);
  text (puntiJolly, 70, 30); //punti jolly sulla barra
  text (puntiOggetti, 110, 30); //punti oggetti sulla barra
  // draw menu options text
  image (barre_yesno,0,20);
  image (sanmarco_yesno, 0, 0);
  // draw focus bar
  if(sanmarcoFocus == SANMARCO_YES){
    image (selezione_yesno,42,125);
  }
  else if(sanmarcoFocus == SANMARCO_NO){
    image (selezione_yesno,42,144);
  }
  //options
  image (yesno,0,20);
  //botton
  image (stuff_ok_viola, 3, height-20);
}

// Function to draw intro san marco screen
void drawIntrosanmarcoMenu()
{
  //draw menu background
  image (sfondo_viola,0,0);
  image (sanmarco, 0, 0);
}

```

```

//topbar
// image (topbar_mermaid, 0,0);
image (barra_mermaid, 0,0);
//elementi barra
textFont (font);
text (puntiJolly, 70, 30); //punti jolly sulla barra
text (puntiOggetti, 110, 30); //punti oggetti sulla barra
//text
image (introsanmarco,0,0);
//botton
image (back_goon_viola, 3, height-20);
}

```

```

// Function to draw san marco q1 screen

```

```

void drawSanmarcoq1Menu()
{
//draw menu background
image (sfondo_viola_question,0,0);
//topbar
image (barra_mermaid, 0,0);
image (comb_icon,140,10);
//elementi barra
textFont (font);
text (puntiJolly, 70, 30); //punti jolly sulla barra
text (puntiOggetti, 110, 30); //punti oggetti sulla barra
//text
image (sanmarco_barrel,0,0);
// draw focus bar
if(sanmarcoq1Focus == SANMARCOQ1_A){
    image (sanmarco_selezione1,42,105);
}
else if(sanmarcoq1Focus == SANMARCOQ1_B){
    image(sanmarco_selezione1, 42, 124);
}
else if(sanmarcoq1Focus == SANMARCOQ1_C){
    image(sanmarco_selezione1, 42, 143);
}
else if(sanmarcoq1Focus == SANMARCOQ1_D){
    image(sanmarco_selezione1, 42, 162);
}
image (sanmarcoq1,0,0);
//botton
image (ok_viola, 3, height-20);
}

```

```

// Function to draw work in progress

```

```

void drawWorkinprogressMenu()
{
    //draw menu background
    image (sfondo_grigio, 0, 0);
    //text
    image (workinprogress,0,0);
    //botton
    image (botton_back, 3, height-20);
}

// Function to draw standby
void drawStandbyMenu()
{
    //draw menu background
    image (standby, 0, 0);
    image (barra_nera,3, height-20);
}

// Function to draw excellent
void drawExcellentMenu()
{
    //draw menu background
    image (sfondo_viola, 0, 0);
    image (barra_mermaid, 0,0);
    image (comb_icon,140,10);
    //elementi barra
    textFont (font);
    text (puntiJolly, 70, 30); //punti jolly sulla barra
    text (puntiOggetti, 110, 30); //punti oggetti sulla barra
    //text
    image (excellent,0,0);
    //botton
    image (stuff_goon_viola, 3, height-20);
}

// Function to draw activity1
void drawActivity1Menu()
{
    //draw menu background
    image (sfondo_viola_activity, 0, 0);
    image (barra_mermaid, 0,0);
    image (comb_icon,140,10);
    //elementi barra
    textFont (font);
    text (puntiJolly, 70, 30); //punti jolly sulla barra
    text (puntiOggetti, 110, 30); //punti oggetti sulla barra
}

```

```

//text
image (sanmarco_a1,0,0);
//botton
image (ok_viola, 3, height-20);
}

// Function to draw excellent1
void drawExcellent1Menu()
{
//draw menu background
image (sfondo_viola, 0, 0);
image (barra_mermaid, 0,0);
image (comb_icon,140,10);
//elementi barra
textFont (font);
text (puntiJolly, 70, 30); //punti jolly sulla barra
text (puntiOggetti, 110, 30); //punti oggetti sulla barra
//text
image (excellent,0,0);
//botton
image (stuff_goon_viola, 3, height-20);
}

// Function to draw code
void drawCodeMenu()
{
//draw menu background
image (sfondo_viola, 0, 0);
image (barra_mermaid, 0,0);
image (comb_icon,140,10);
//elementi barra
textFont (font);
text (puntiJolly, 70, 30); //punti jolly sulla barra
text (puntiOggetti, 110, 30); //punti oggetti sulla barra
//text
image (code,0,0);
image (codice,0,32);
//botton
image (ok_viola, 3, height-20);
}

// Function to draw completed
void drawCompletedMenu()
{
//draw menu background

```

```

image (sfondo_viola, 0, 0);
image (barra_mermaid, 0, 0);
//elementi barra
textFont (font);
text (puntiJolly, 70, 30); //punti jolly sulla barra
text (puntiOggetti, 110, 30); //punti oggetti sulla barra
//text
image (completed,0,0);
drawPettine ();
//botton
image (stuff_goon_viola, 3, height-20);
}

```

// Function to draw repeat

```

void drawRepeatMenu()
{
//draw menu background
image (sfondo_viola, 0, 0);
image (barra_mermaid, 0,0);
//elementi barra
textFont (font);
text (puntiJolly, 70, 30); //punti jolly sulla barra
text (puntiOggetti, 110, 30); //punti oggetti sulla barra
//text
image (barre_yesno,0,1);
// draw focus bar
if(repeatFocus == REPEAT_YES){
    image (selezione_yesno,42,106);
}
else if(repeatFocus == REPEAT_NO){
    image (selezione_yesno,42,125);
}
image (repeat, 0, 0);
//botton
image (stuff_ok_viola, 3, height-20);
}

```

// Function to draw recover

```

void drawRecoverMenu()
{
//draw menu background
image (sfondo_viola, 0, 0);
image (barra_mermaid, 0,0);
//elementi barra
textFont (font);
text (puntiJolly, 70, 30); //punti jolly sulla barra

```

```

text (puntiOggetti, 110, 30); //punti oggetti sulla barra
//text
image (fine,0,0);
drawBye ();
//botton
image (stuff_viola, 3, height-20);
}

// Function to draw error
void drawErrorMenu()
{
//draw menu background
image (sfondo_viola, 0, 0);
image (barra_mermaid, 0,0);
//elementi barra
textFont (font);
text (puntiJolly, 70, 30); //punti jolly sulla barra
text (puntiOggetti, 110, 30); //punti oggetti sulla barra
//text
image (error,0,0);
//botton
image (goon_viola, 3, height-20);
}

// Function to draw allobjects_screen
void drawAllobjectsMenu()
{
colorObject();

background (255);
//draw menu background
image (sfondo_viola,0,0);
//topbar
//image (topbar_mermaid, 0,0);
image (barra_mermaid, 0,0);
//elementi barra
textFont (font);
text (puntiJolly, 70, 30); //punti jolly sulla barra
text (puntiOggetti, 110, 30); //punti oggetti sulla barra
// draw menu options text
for(int i=0;i<=6; i++)
{
image(objects_render[i], xcorobjects[i], ycorobjects[i]);
image(objects_render_big[0], 44, 47);
}
//botton

```

```

image (back_goon_viola, 3, height-20);

sortListObjects();
sortListbigObjects();
}

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

// Draw - Executes forever, provides user feedback (Logic Repetition)
void draw()
{
  //println(SCREEN_STUFF);
  if(screenMode == SCREEN_LOGO){
    drawLogoMenu();
  }
  else if(screenMode == SCREEN_STUFF){
    drawStuffMenu();
  }
  else if(screenMode == SCREEN_LANGUAGE){
    println(screenMode);
    drawLanguageMenu();
  }
  else if(screenMode == SCREEN_HELP){
    println(screenMode);
    drawHelpMenu();
  }
  else if(screenMode == SCREEN_AIM){
    drawAimMenu();
  }
  else if(screenMode == SCREEN_MERMAID){
    drawMermaidMenu();
  }
  else if(screenMode == SCREEN_ALIEN){
    drawAlienMenu();
  }
  else if(screenMode == SCREEN_PIRATE){
    drawPirateMenu();
  }
  else if(screenMode == SCREEN_SETTING){
    drawSettingMenu();
  }
  else if(screenMode == SCREEN_NEW){
    drawNewMenu();
  }
}

```

```
else if(screenMode == SCREEN_CHOOSE){
    drawChooseMenu();
}
else if(screenMode == SCREEN_INTROMERMAID){
    drawIntromermaidMenu();
}
else if(screenMode == SCREEN_INTROMERMAID2){
    drawIntromermaid2Menu();
}
else if(screenMode == SCREEN_OBJECTS){
    drawObjectsMenu();
}
else if(screenMode == SCREEN_STARTMERMAID){
    drawStartmermaidMenu();
}
else if(screenMode == SCREEN_STARTMERMAID2){
    drawStartmermaid2Menu();
}
else if(screenMode == SCREEN_SANMARCO){
    drawSanmarcoMenu();
}
else if(screenMode == SCREEN_INTROSANMARCO){
    drawIntrosanmarcoMenu();
}
else if(screenMode == SCREEN_SANMARCOQ1){
    drawSanmarcoq1Menu();
}
else if(screenMode == SCREEN_WORKINPROGRESS){
    drawWorkinprogressMenu();
}
else if(screenMode == SCREEN_STANDBY){
    drawStandbyMenu();
}
else if(screenMode == SCREEN_EXCELLENT){
    drawExcellentMenu();
}
else if(screenMode == SCREEN_ACTIVITY1){
    drawActivity1Menu();
}
else if(screenMode == SCREEN_EXCELLENT1){
    drawExcellent1Menu();
}
else if(screenMode == SCREEN_CODE){
    drawCodeMenu();
}
else if(screenMode == SCREEN_COMPLETED){
```

```

    drawCompletedMenu();
}
else if(screenMode == SCREEN_REPEAT){
    drawRepeatMenu();
}
else if(screenMode == SCREEN_RECOVER){
    drawRecoverMenu();
}
else if(screenMode == SCREEN_ERROR){
    drawErrorMenu();
}
else if(screenMode == SCREEN_ALLOBJECTS){
    drawAllobjectsMenu();
}
}

//*****
//nickname
//*****

PFont font; // Declare font
String[] render_text; // Set initial text
String[] user_text;
String multitapRead;

void keyPressedNickname()
{
    if (keyCode == FIRE) {
        //user_text = textInput("Name", "", 20); // Get the string from the device input
method
        if(user_text != null){ // Check to make sure text is not
null or will cause an exception
            render_text = user_text;
        }
    }
    else if(keyCode == '1')
    {
        render_text[0] = "a";
    }
}

void drawNickname() {
    multitapRead = "";
    for (int i=0; i<=multitapBufferIndex; i++)
    {

```

```

    multitapRead += multitapBuffer[i];
}
if(render_text != null){
    text(render_text[0], 15, 102);           // Render the input text on screen
}
if(multitapRead != null){
    text(multitapRead, 15, 102);           // Render the input text on screen
}
}

```

```

//*****
//menu
//*****

```

```

//////////menu language

```

```

int[] xcor = {
    60, 129, 120, 76, 32, 23};
int[] ycor = {
    38, 91, 119, 139, 119, 91};
PImage[] flags;
PImage[] flags_render;
PImage[] flags_big;
PImage[] flags_render_big;
int counterFlag=0;

```

```

void keyPressedMenuflag()
{
    if(keyCode == RIGHT)
    {
        counterFlag --;
    }
    else if (keyCode == LEFT)
    {
        counterFlag ++;
    }
    if (counterFlag == 0){
        // if pressed in
        if (keyCode == SOFTKEY2){
            screenMode = SCREEN_STUFF;
        }
    }
}

```

```

//counter = constrain(counter, 0, 5);
if(counterFlag > 5)
{

```

```
    counterFlag = 0;
}
else if (counterFlag < 0)
{
    counterFlag = 5;
}
println("counterFlag:"+counterFlag);
sortList();
sortListbig();
}
```

```
void sortList()
{
    if(counterFlag == 0)
    {
        flags_render[0] = flags[0];
        flags_render[1] = flags[1];
        flags_render[2] = flags[2];
        flags_render[3] = flags[3];
        flags_render[4] = flags[4];
        flags_render[5] = flags[5];
    }

    else if(counterFlag == 1)
    {
        flags_render[0] = flags[5];
        flags_render[1] = flags[0];
        flags_render[2] = flags[1];
        flags_render[3] = flags[2];
        flags_render[4] = flags[3];
        flags_render[5] = flags[4];
    }

    else if(counterFlag == 2)
    {
        flags_render[0] = flags[4];
        flags_render[1] = flags[5];
        flags_render[2] = flags[0];
        flags_render[3] = flags[1];
        flags_render[4] = flags[2];
        flags_render[5] = flags[3];
    }

    else if(counterFlag == 3)
    {
        flags_render[0] = flags[3];
```

```

    flags_render[1] = flags[4];
    flags_render[2] = flags[5];
    flags_render[3] = flags[0];
    flags_render[4] = flags[1];
    flags_render[5] = flags[2];
}

else if(counterFlag == 4)
{
    flags_render[0] = flags[2];
    flags_render[1] = flags[3];
    flags_render[2] = flags[4];
    flags_render[3] = flags[5];
    flags_render[4] = flags[0];
    flags_render[5] = flags[1];
}

else if(counterFlag == 5)
{
    flags_render[0] = flags[1];
    flags_render[1] = flags[2];
    flags_render[2] = flags[3];
    flags_render[3] = flags[4];
    flags_render[4] = flags[5];
    flags_render[5] = flags[0];
}
}

void sortListbig()
{
    if(counterFlag == 0)
    {
        flags_render_big[0] = flags_big[0];
        flags_render_big[1] = flags_big[1];
        flags_render_big[2] = flags_big[2];
        flags_render_big[3] = flags_big[3];
        flags_render_big[4] = flags_big[4];
        flags_render_big[5] = flags_big[5];
    }

    else if(counterFlag == 1)
    {
        flags_render_big[0] = flags_big[5];
        flags_render_big[1] = flags_big[0];
        flags_render_big[2] = flags_big[1];
        flags_render_big[3] = flags_big[2];
    }
}

```

```

    flags_render_big[4] = flags_big[3];
    flags_render_big[5] = flags_big[4];
}

else if(counterFlag == 2)
{
    flags_render_big[0] = flags_big[4];
    flags_render_big[1] = flags_big[5];
    flags_render_big[2] = flags_big[0];
    flags_render_big[3] = flags_big[1];
    flags_render_big[4] = flags_big[2];
    flags_render_big[5] = flags_big[3];
}

else if(counterFlag == 3)
{
    flags_render_big[0] = flags_big[3];
    flags_render_big[1] = flags_big[4];
    flags_render_big[2] = flags_big[5];
    flags_render_big[3] = flags_big[0];
    flags_render_big[4] = flags_big[1];
    flags_render_big[5] = flags_big[2];
}

else if(counterFlag == 4)
{
    flags_render_big[0] = flags_big[2];
    flags_render_big[1] = flags_big[3];
    flags_render_big[2] = flags_big[4];
    flags_render_big[3] = flags_big[5];
    flags_render_big[4] = flags_big[0];
    flags_render_big[5] = flags_big[1];
}

else if(counterFlag == 5)
{
    flags_render_big[0] = flags_big[1];
    flags_render_big[1] = flags_big[2];
    flags_render_big[2] = flags_big[3];
    flags_render_big[3] = flags_big[4];
    flags_render_big[4] = flags_big[5];
    flags_render_big[5] = flags_big[0];
}
}

```

//////////menu choose character//////////

```

int[] xcorcharacter = {
    65, 112, 18};
int[] ycorcharacter = {
    32, 115, 115};
PImage[] character;
PImage[] character_render;
PImage[] character_big;
PImage[] character_render_big;

int counterCharacter=0;

void keyPressedMenucharacter()
{ //println("counter:"+counterCharacter);
  if(keyCode == RIGHT)
  {
    counterCharacter ++;
  }
  else if (keyCode == LEFT)
  {
    counterCharacter --;
  }

  if (counterCharacter == 0){
    // if pressed in
    if (keyCode == SOFTKEY2){
      screenMode= SCREEN_INTROMERMAID;
    }
  }

  //counterCharacter = constrain(counterCharacter, 0, 2);

  if(counterCharacter > 2)
  {
    counterCharacter = 0;
  }
  else if (counterCharacter < 0)
  {
    counterCharacter = 2;
  }

  //println("counterCharacter:"+counterCharacter);
  sortListCharacter();
  sortListCharacterbig();
}

```

```

void sortListCharacter()
{
    if(counterCharacter == 0)
    {
        character_render[0] = character[0];
        character_render[1] = character[1];
        character_render[2] = character[2];
    }

    else if(counterCharacter == 1)
    {
        character_render[0] = character[2];
        character_render[1] = character[0];
        character_render[2] = character[1];
    }

    else if(counterCharacter == 2)
    {
        character_render[0] = character[1];
        character_render[1] = character[2];
        character_render[2] = character[0];
    }
}

void sortListCharacterbig()
{
    if(counterCharacter == 0)
    {
        character_render_big[0] = character_big[0];
        character_render_big[1] = character_big[1];
        character_render_big[2] = character_big[2];
    }

    else if(counterCharacter == 1)
    {
        character_render_big[0] = character_big[2];
        character_render_big[1] = character_big[0];
        character_render_big[2] = character_big[1];
    }

    else if(counterCharacter == 2)
    {
        character_render_big[0] = character_big[1];
        character_render_big[1] = character_big[2];
        character_render_big[2] = character_big[0];
    }
}

```

```

    }
}
//////////endo of menu choose character//////////

//////////menu choose objects//////////
int[] xcorobjects = {
    76, 20, 28, 60, 95, 126, 134 };
int[] ycorobjects = {
    73, 105, 138, 160, 160, 138, 105 };
PImage[] objects;
PImage[] objects_render;
PImage[] objects_big;
PImage[] objects_render_big;
PImage[] objects_c;
PImage[] objects_render_c;
PImage[] objects_big_c;
PImage[] objects_render_big_c;

int counterObject=0;

void keyPressedObjectsMenu (){

    if(keyCode == RIGHT)
    {
        counterObject ++;
    }
    else if (keyCode == LEFT)
    {
        counterObject --;
    }

    //counter = constrain(counter, 0, 6);
    if(counterObject > 6)
    {
        counterObject = 0;
    }
    else if (counterObject < 0)
    {
        counterObject = 6;
    }
    println("counterObject:"+counterObject);
    sortListObjects();
    sortListbigObjects();
}

```

```
void sortListObjects()
{
    if(counterObject == 0)
    {
        objects_render[0] = objects[0];
        objects_render[1] = objects[1];
        objects_render[2] = objects[2];
        objects_render[3] = objects[3];
        objects_render[4] = objects[4];
        objects_render[5] = objects[5];
        objects_render[6] = objects[6];
    }

    else if(counterObject == 1)
    {
        objects_render[0] = objects[6];
        objects_render[1] = objects[0];
        objects_render[2] = objects[1];
        objects_render[3] = objects[2];
        objects_render[4] = objects[3];
        objects_render[5] = objects[4];
        objects_render[6] = objects[5];
    }

    else if(counterObject == 2)
    {
        objects_render[0] = objects[5];
        objects_render[1] = objects[6];
        objects_render[2] = objects[0];
        objects_render[3] = objects[1];
        objects_render[4] = objects[2];
        objects_render[5] = objects[3];
        objects_render[6] = objects[4];
    }

    else if(counterObject == 3)
    {
        objects_render[0] = objects[4];
        objects_render[1] = objects[5];
        objects_render[2] = objects[6];
        objects_render[3] = objects[0];
        objects_render[4] = objects[1];
        objects_render[5] = objects[2];
        objects_render[6] = objects[3];
    }
}
```

```
else if(counterObject == 4)
{
    objects_render[0] = objects[3];
    objects_render[1] = objects[4];
    objects_render[2] = objects[5];
    objects_render[3] = objects[6];
    objects_render[4] = objects[0];
    objects_render[5] = objects[1];
    objects_render[6] = objects[2];
}

else if(counterObject == 5)
{
    objects_render[0] = objects[2];
    objects_render[1] = objects[3];
    objects_render[2] = objects[4];
    objects_render[3] = objects[5];
    objects_render[4] = objects[6];
    objects_render[5] = objects[0];
    objects_render[6] = objects[1];
}

else if(counterObject == 6)
{
    objects_render[0] = objects[1];
    objects_render[1] = objects[2];
    objects_render[2] = objects[3];
    objects_render[3] = objects[4];
    objects_render[4] = objects[5];
    objects_render[5] = objects[6];
    objects_render[6] = objects[0];
}
}

void sortListbigObjects()
{
    if(counterObject == 0)
    {
        objects_render_big[0] = objects_big[0];
        objects_render_big[1] = objects_big[1];
        objects_render_big[2] = objects_big[2];
        objects_render_big[3] = objects_big[3];
        objects_render_big[4] = objects_big[4];
        objects_render_big[5] = objects_big[5];
        objects_render_big[6] = objects_big[6];
    }
}
```

```
else if(counterObject == 1)
{
    objects_render_big[0] = objects_big[6];
    objects_render_big[1] = objects_big[0];
    objects_render_big[2] = objects_big[1];
    objects_render_big[3] = objects_big[2];
    objects_render_big[4] = objects_big[3];
    objects_render_big[5] = objects_big[4];
    objects_render_big[6] = objects_big[5];
}
```

```
else if(counterObject == 2)
{
    objects_render_big[0] = objects_big[5];
    objects_render_big[1] = objects_big[6];
    objects_render_big[2] = objects_big[0];
    objects_render_big[3] = objects_big[1];
    objects_render_big[4] = objects_big[2];
    objects_render_big[5] = objects_big[3];
    objects_render_big[6] = objects_big[4];
}
```

```
else if(counterObject == 3)
{
    objects_render_big[0] = objects_big[4];
    objects_render_big[1] = objects_big[5];
    objects_render_big[2] = objects_big[6];
    objects_render_big[3] = objects_big[0];
    objects_render_big[4] = objects_big[1];
    objects_render_big[5] = objects_big[2];
    objects_render_big[6] = objects_big[3];
}
```

```
else if(counterObject == 4)
{
    objects_render_big[0] = objects_big[3];
    objects_render_big[1] = objects_big[4];
    objects_render_big[2] = objects_big[5];
    objects_render_big[3] = objects_big[6];
    objects_render_big[4] = objects_big[0];
    objects_render_big[5] = objects_big[1];
    objects_render_big[6] = objects_big[2];
}
```

```
else if(counterObject == 5)
```

```

{
  objects_render_big[0] = objects_big[2];
  objects_render_big[1] = objects_big[3];
  objects_render_big[2] = objects_big[4];
  objects_render_big[3] = objects_big[5];
  objects_render_big[4] = objects_big[6];
  objects_render_big[5] = objects_big[0];
  objects_render_big[6] = objects_big[1];
}
else if(counterObject == 6)
{
  objects_render_big[0] = objects_big[1];
  objects_render_big[1] = objects_big[2];
  objects_render_big[2] = objects_big[3];
  objects_render_big[3] = objects_big[4];
  objects_render_big[4] = objects_big[5];
  objects_render_big[5] = objects_big[6];
  objects_render_big[6] = objects_big[0];
}
}
//////////////////////////////////end of screen choose objects//////////////////////////////////

//*****
//score and objects
//*****

///punti oggetti///
void okPuntiOggetti()
{
  puntiOggetti= "x"+n; //update the score in the toolbar
  n=n+1;
  println(n);
}

///oggetti vinti///
void colorObject() //color the conquered object
{
  objects[1]=objects_c[1];
  objects_big[1]=objects_big_c[1];
}

//*****
//animation
//*****

///// animazione bandiera

```

```

int xbandiera = 93;           // flag's horizontal position
int ybandiera = 18;          // flag's vertical position
int logoAnimationFrame = 0;

void drawLogoAnimation(){
    // draw flag, only showing a single frame of the sprite sheet
    image(bandiera, (logoAnimationFrame / 20) * 71, 0, 71, bandiera.height, xbandiera,
ybandiera);
    // show the next frame when we draw again
    logoAnimationFrame++;
    // stay on the sheet!
    if(60 < logoAnimationFrame){ // if we are past the last frame
        logoAnimationFrame = 0; // go to the first frame
    }
}

/////animazione sirena intro
PImage mermaidAnimationIntro;
int xmermaidAnimationIntro = 25;           // mermaid's horizontal position
int ymermaidAnimationIntro = 100;          // mermaid's vertical position
int framemermaidintro = 0;                 // which frame of mermaid's sprite sheet to show
int framemermaidintroWidth = 121; // the width (in pixels) of a single frame in
mermaid's sprite sheet

void drawMermaidIntroAnimation(){
    mermaidAnimationIntro = loadImage("strip_mermaidAnimationIntro.png");
                                                // load mermaid's sprite animation graphics
    framerate(4);                             // slow framerate so we can see mermaid moving
    // draw mermaid, only showing a single frame of the sprite sheet
    image(mermaidAnimationIntro, framemermaidintro * framemermaidintroWidth, 0,
framemermaidintroWidth, mermaidAnimationIntro.height, xmermaidAnimationIntro,
ymermaidAnimationIntro);
    // show the next frame when we draw again
    framemermaidintro++;
    // stay on the sheet!
    if(5 < framemermaidintro){ // if we are past the last frame
        framemermaidintro = 0; // go to the first frame
    }
}

///// animazione coda (start mission)
PImage coda;
int xcoda = 22;           // mermaid's horizontal position
int ycoda = 100;          // mermaid's vertical position
int frameCoda = 0;        // which frame of mermaid's sprite sheet to show

```

```

int frameCodaWidth = 129; // the width (in pixels) of a single frame in mermaid's sprite
sheet

void drawCoda(){
    coda = loadImage("strip_coda.png"); // load mermaid's sprite animation graphics
    framerate(4); // slow framerate so we can see mermaid
moving
    // draw mermaid, only showing a single frame of the sprite sheet
    image(coda, frameCoda * frameCodaWidth, 0, frameCodaWidth, coda.height, xcoda, ycoda);
    // show the next frame when we draw again
    frameCoda++;
    // stay on the sheet!
    if(4 < frameCoda){ // if we are past the last frame
        frameCoda = 0; // go to the first frame
    }
}

///// animazione pettine
PImage animation_comb;

int xanimation_comb = 38; // mermaid's horizontal position
int yanimation_comb = 100; // mermaid's vertical position
int frameAnimation_comb = 0; // which frame of mermaid's sprite sheet to show
int frameAnimation_combWidth = 100; // the width (in pixels) of a single frame in
mermaid's sprite sheet

void drawPettine(){
    animation_comb = loadImage("animation_comb.png"); // load mermaid's sprite animation
graphics
    framerate(5); // slow framerate so we can see mermaid
moving
    // draw mermaid, only showing a single frame of the sprite sheet
    image(animation_comb, frameAnimation_comb * frameAnimation_combWidth, 0,
frameAnimation_combWidth, animation_comb.height, xanimation_comb, yanimation_comb);
    // show the next frame when we draw again
    frameAnimation_comb++;
    // stay on the sheet!
    if(5 < frameAnimation_comb){ // if we are past the last frame
        frameAnimation_comb = 0; // go to the first frame
    }
}

/////animazione saluto
PImage bye;
int xbye = 20; // mermaid's horizontal position
int ybye = 125; // mermaid's vertical position

```

```
int frameBye = 0;          // which frame of mermaid's sprite sheet to show
int frameByeWidth = 125; // the width (in pixels) of a single frame in mermaid's sprite
sheet

void drawBye(){
    bye = loadImage("strip_saluto.png"); // load mermaid's sprite animation graphics
    framerate(4);                        // slow framerate so we can see mermaid
moving
    // draw mermaid, only showing a single frame of the sprite sheet
    image(bye, frameBye * frameByeWidth, 0, frameByeWidth, bye.height, xbye, ybye);
    // show the next frame when we draw again
    frameBye++;
    // stay on the sheet!
    if(4 < frameBye){ // if we are past the last frame
        frameBye = 0; // go to the first frame
    }
}

////////// end of code //////////
```