```
    instRect.centerx=HALF_WINWIDTH
    instRect.centery=HALF_WINHEIGTH-80
    DISPLAYSURF.blit(instSurf,instRect)
    DISPLAYSURF.blit(SEQ_SURFS[numtest-1],SEQ_RECTS[numtest-1])
    index_sec=NumSec[numtest-1]

    GAME=button(' START ', HALF_WINWIDTH-100, HALF_WINHEIGTH, 200,100, BUTTONCOLOR_OFF,
BUTTONCOLOR_ON, WHITE, BASICFONT,"instructions")
    ENDTRIAL=button(' TRIAL END ', HALF_WINWIDTH+400, HALF_WINHEIGTH+100, 200,100,
BUTTONCOLOR_OFF, BUTTONCOLOR_ON, WHITE, BASICFONT,"instructions")

    if GAME==1:
        if int(index_sec)>10:
            nameTrialFile='f'+nameTrialDirec+'T'+code_sec+index_sec+dateStr+'.txt'
        else:
            nameTrialFile='f'+nameTrialDirec+'T'+code_sec+'0'+index_sec+dateStr+'.txt'
        print(nameTrialFile)
        arduino.reset_output_buffer()
        arduino.write(str(nameTrialFile).encode('ascii'))

    if ENDTRIAL==1:
        inst=0
        GAME=0
        sec=0
        READY=0
        info_count=0
        count_full=0
        index_ok=0
        COUNT=0

    pygame.display.update()
    FPSCLOCK.tick(15)
    checkForQuit()


def demoScreen():
    global DEMO_SURFS,DEMO_RECTS
    global GAME
    global inst
    global nameDemoFile,nameTrialDirec,nameTrialFile
    global dateStr
    global info_count,count_full,COUNT,READY,index_ok

    msg='Enter Demo index'

    NUMBERFONT=pygame.font.Font('freesansbold.ttf',25)

    DISPLAYSURF.fill(BGCOLOR)

    INSTRUCCION_COUNT = 'Enter the Demo index: '
    countSurf = BASICFONT.render(INSTRUCCION_COUNT,1,TEXTCOLOR)
    countRect = countSurf.get_rect()
    countRect.centerx = HALF_WINWIDTH-200
    countRect.centery = HALF_WINHEIGTH+60

    DISPLAYSURF.blit(countSurf, countRect)

    for i in range(len(DEMO_SURFS)):
        DISPLAYSURF.blit(DEMO_SURFS[i], DEMO_RECTS[i])

    if index_ok==0:
```

```
        index_ok=CountTrialScreen(msg,40)
    elif index_ok==1:

buttontext(str(info_count),HALF_WINWIDTH,HALF_WINHEIGTH+40,300,40,WHITE,NUMBERFONT,"info"
)
        GAME=button(' READY ', HALF_WINWIDTH-100, HALF_WINHEIGTH+120, 200,100,
BUTTONCOLOR_OFF, BUTTONCOLOR_ON, WHITE, BASICFONT,"instructions")

    if GAME==1:
        if info_count>10:
            nameDemoFile=nameDemoFile+str(info_count)+dateStr+'.txt'
        else:
             nameDemoFile=nameDemoFile+'0'+str(info_count)+dateStr+'.txt'
        print(nameDemoFile)
        arduino.reset_output_buffer()
        arduino.write(str(nameDemoFile).encode('ascii'))
        time.sleep(0.1)

    Menu=button(' MENU', HALF_WINWIDTH+450, HALF_WINHEIGTH+150, 150,60, BUTTONCOLOR_OFF,
BUTTONCOLOR_ON, WHITE, BASICFONT,"instructions")

    if Menu==1:
        READY=0
        info_count=0
        count_full=0
        COUNT=0
        index_ok=0
        inst=0

    pygame.display.update()
    FPSCLOCK.tick(15)
    checkForQuit()


def getDistanceAngle(angle):
    L1 = 90#distancia del punto medio de la pantalla al piso
    L2 = 60#distancia de los gluteos de la persona al piso

    weight_cm=43.6#ancho de la pantalla en cm
    height_cm=23.5#alto de la pantalla en cm
    R = L1-L2
    distx = R*math.sin(angle*(math.pi/180))*(WINWIDTH/weight_cm)
    #disty = R*(1-math.cos(angle*(math.pi/180)))*(WINHEIGTH/height_cm)

    return distx


def PlayGame():
    global inst,GAME,myangles
    global trial,demo,numtest,sec
    global info_count,count_full,COUNT,READY,index_ok
    global nameDemoFile,nameTrialDirec,nameTrialFile

    TESTFONT = pygame.font.Font('freesansbold.ttf',20)

    #Bloques para actualizar secciones de pantalla
    bargoal_rect=pygame.Rect(0,0,WINWIDTH,30)
    object_rect=pygame.Rect(0,250,WINWIDTH,270)
    boton_rect=pygame.Rect(WINWIDTH-250,WINHEIGTH-150,125,50)

    #Inicializa variables
```

```
    posx=0
    posy=0
    neg=-1
    #offset=1.35

    test_bt=0

    color_bar=RED

    despx=0
    despy=0
    angle_imu=10

    time_limit_bar=WINWIDTH

    arduino.reset_input_buffer()

    NUMBERFONT = pygame.font.Font('freesansbold.ttf',200)

    pygame.time.set_timer(pygame.USEREVENT,1000)

    for i in range(3):
        second = 0

        if i == 0:
            number="3"
        elif i== 1:
            number="2"
        else:
            number="1"

        numberSurf = NUMBERFONT.render(number,1,TEXTCOLOR)
        numberRect = numberSurf.get_rect()
        numberRect.centerx = HALF_WINWIDTH
        numberRect.centery = HALF_WINHEIGTH

        DISPLAYSURF.fill(BGCOLOR)
        DISPLAYSURF.blit(numberSurf, numberRect)
        pygame.display.update()
        FPSCLOCK.tick(15)

        while second == 0:
            for event in pygame.event.get():
                if event.type == pygame.USEREVENT:
                    second=1
                elif event.type == pygame.QUIT:
                    terminate()

    #Modificando para actualizar barra cada 10ms
    time_timer=10#ms
    time_target=5000#ms
    count_timer=time_target/time_timer
    rate=(WINWIDTH*time_timer)/time_target
    pygame.time.set_timer(pygame.USEREVENT,time_timer)#time_target(time_timer*count)
    time.sleep(0.001)

    code = 'Y'#Enable interrupts
    arduino.reset_output_buffer()
    arduino.write(str(code).encode('ascii'))

    if demo==1:
```

```
        i=random.randint(1,len(PRUEBA_A))
        print(i)
        ANGLES=PRUEBA_A[i-1]
    elif trial==1:
        ANGLES=PRUEBA[numtest-1]

    ang='0'

    width_target=250
    heigth_target=250
    width_imu=200
    heigth_imu=200

    targetRect=pygame.Rect(0,0,width_target,heigth_target)#130,250
    imuRect=pygame.Rect(0,0,width_imu,heigth_imu)#120,200

    targetRect.centery=HALF_WINHEIGTH
    imuRect.centery=HALF_WINHEIGTH

    color_bg=BLACK

    DISPLAYSURF.fill(color_bg)
    pygame.display.flip()

    for i in range(len(ANGLES)):
        contpos=1

        cont=1
        cinco=0
        read=0
        angle = ANGLES[i]

        if angle == 0:
            targetRect.centerx = HALF_WINWIDTH
        else:
            posx = getDistanceAngle(angle)
            targetRect.centerx = HALF_WINWIDTH+posx

        imuRect.centerx = HALF_WINWIDTH+despx

        if angle == 0:
            code='0'
        elif angle < 0:
            ang=str(angle*(-1))
            code='N'+ang
        elif angle > 0:
            ang=str(angle)
            code='P'+ang


        arduino.reset_output_buffer()
        arduino.write(str(code).encode('ascii'))

        posangle=0
        while cinco == 0 and test_bt==0:
            #Lectura del dato
            if (arduino.inWaiting()>0):
                texto=arduino.readline()
                texto=texto.split()
                angle_imu=neg*(float(texto[0])-offset)
                despx=getDistanceAngle(angle_imu)
```

```
                imuRect.centerx = HALF_WINWIDTH+despx#Show only the projection in x axis

            #Anàlisis logro de posicòn
            if angle==0:
                if (angle_imu<(angle+0.075)) and (angle_imu>(angle-0.075)):#0.1
                    posangle=1
                else:
                    posangle=0
            elif angle>0:
                if (angle_imu<(angle+angle*0.05)) and (angle_imu>(angle-angle*0.05)):#0.1
                    posangle=1
                else:
                    posangle=0
            elif angle<0:
                if (angle_imu<(angle-angle*0.05)) and (angle_imu>(angle+angle*0.05)):#0.1
                    posangle=1
                else:
                    posangle=0

            test_bt=button('TRIAL END', WINWIDTH-250,WINHEIGTH-
150,125,50,BUTTONCOLOR_OFF,BUTTONCOLOR_ON,WHITE,TESTFONT,"instructions")

            if test_bt==1:
                arduino.reset_input_buffer()
                code='Z'#Disable timer
                arduino.reset_output_buffer()
                arduino.write(str(code).encode('ascii'))
                time.sleep(0.005)
                i=len(ANGLES)+1
                READY=0
                info_count=0
                count_full=0
                COUNT=0
                index_ok=0
                nameTrialFile='T'

            pygame.draw.rect(DISPLAYSURF,color_bar, (0,0,time_limit_bar,30))#GREEN
            pygame.draw.ellipse(DISPLAYSURF,GREEN,targetRect,5)
            pygame.draw.ellipse(DISPLAYSURF,RED_BRIGTH,imuRect,0)

            pygame.display.update(bargoal_rect)
            pygame.display.update(object_rect)
            pygame.display.update(boton_rect)

            arduino.reset_output_buffer()
            arduino.write(str('Q').encode('ascii'))


            pygame.draw.rect(DISPLAYSURF,BLACK, (0,0,time_limit_bar,30))#GREEN
            pygame.draw.ellipse(DISPLAYSURF,color_bg,targetRect,5)
            pygame.draw.ellipse(DISPLAYSURF,color_bg,imuRect,0)
            pygame.draw.rect(DISPLAYSURF,color_bg,(WINWIDTH-250,WINHEIGTH-150,125,50))

            for event in pygame.event.get():
                if event.type == pygame.USEREVENT:
                    time_limit_bar=time_limit_bar-rate
                    if posangle==1:
                        color_bar=GREEN
                    else:
                        color_bar=RED
```

```
                    if cont==count_timer:
                        cinco=1
                        time_limit_bar=WINWIDTH
                    else:
                        cont=cont+1
                elif event.type == pygame.QUIT:
                    terminate()

    if test_bt == 0:
        time.sleep(0.1)
        arduino.reset_input_buffer()
        code='Z'#Disable timer
        arduino.reset_output_buffer()
        arduino.write(str(code).encode('ascii'))
        time.sleep(0.005)
        READY=0
        info_count=0
        count_full=0
        COUNT=0
        index_ok=0
        nameTrialFile='T'

    if demo==1 or numtest==len(PRUEBA):
        inst = 0
        GAME = 0
        trial=0
        demo=0
        sec=0
        numtest=1
        READY=0
        info_count=0
        count_full=0
        COUNT=0
        index_ok=0
        nameDemoFile=''
        nameTrialDirec=''
        nameTrialFile=''
    else:
        numtest=numtest+1
        GAME=0


def TextRect(text,font,color):
    textSurf = font.render(text,True,color)
    textRect = textSurf.get_rect()
    return (textSurf,textRect)


def button(msg,x,y,w,h,ic,ac,tc,font,action=None):
    var = 0

    mouse=0
    click=0

    mouse = pygame.mouse.get_pos()
    click = pygame.mouse.get_pressed()

    if x+w > mouse[0] > x and y+h > mouse[1] > y:
        pygame.draw.rect(DISPLAYSURF,ac, (x,y,w,h))
        if click[0] == 1 and action != None:
```

```
            if action == "instructions" or action == "start" or action=="number" or
action=="back" or action=="info" or action=="complete":
                var = 1
    else:
        pygame.draw.rect(DISPLAYSURF,ic, (x,y,w,h))

    textSurf, textRect = TextRect(msg, font,tc)
    textRect.center = (x+(w/2),y+(h/2))
    DISPLAYSURF.blit(textSurf,textRect)
    return var


def checkForQuit():
    for event in pygame.event.get():
        if event.type == QUIT:
            terminate()

def terminate():
    pygame.quit()
    sys.exit()

if __name__ == '__main__':
    main()
```

BIBLIOGRAFÍA

[1] "smaffer_vgax_ VGA library for Arduino UNO". [En línea]. Disponible en: https://github.com/smaffer/vgax. [Consultado: 17-jul-2018].

[2] "RPi-Distro_RTIMULib_ RTIMULib is a C++ and Python library that makes it easy to use 9-dof and 10-dof IMUs with embedded Linux systems (especially the Raspberry PI and Intel Edison!)". [En línea]. Disponible en: https://github.com/RPi-Distro/RTIMULib. [Consultado: 12-jul-2018].

[3] "pygame". [En línea]. Disponible en: https://www.pygame.org/news. [Consultado: 20-sep-2018].