

Proyecto Programado

Tabla de contenido

Contenido

Tabla de contenido	1
Introduccion:.....	2
Descripción del Problema:	3
Dificultades Encontradas:.....	4
Análisis de resultados:	5
Bitácora de Actividades:	7
Estadística de tiempos:	9
Conclusión Personal:	10

Introducción:

Un robot se deriva del checo robota que significa literalmente trabajo o labor y figurativamente trabajo pesado. Un robot es un maquina programable que puede manipular objetos y realizar operaciones. Estos robots pueden ser entidades físicos o virtuales a estos últimos se le llama “bots”. Pero ambos tienen características en común como:

No son naturales, es decir han sido creados artificialmente.

- Son programable.
- Pueden realizar movimientos coordinados.
- Pueden realizar operaciones matemáticas o sencillas a una velocidad mayor a la de un humano
- Toman decisiones basados en su programación

Para los agentes o robots físicos, existen diferentes tipos y divisiones; se clasifican normalmente por sus generaciones la cual llega hasta la quinta.

Pero para efectos de este trabajo vamos a clasificarlos por sus clases (o capacidades):

- Androides: Robots con forma humana. Imitan el comportamiento de las personas
- Móviles: Se desplazan mediante ruedas o algún otro mecanismo que asegure el transporte.
- Zoomórficos: Imitan a los animales.
- Polis articulados: Para uso industrial, normalmente en tareas repetitivas.

A finales de los 90 se lanza el Lego Mindstorms, una nueva línea de lego enfocada en el desarrollo de robótica y conocimiento en programación para los más pequeños de la casa. Esto impulsa la implementación de la robótica en la educación complementaria. A su vez en el 2004 se celebran las primeras olimpiadas de robótica con tan solo 12 países participando, haciendo un llamado más fuerte a las demás compañías a desarrollar sus propios equipos para desarrollar robots funcionales y fáciles de armar pues en su mayoría son manipulados por niños y hasta son comercializados como juguetes. De hecho, lego saca una segunda versión de Mindstorms la cuales WeDo la cual es una plataforma para niños aún más pequeños en edad. Algunas de esas empresas son: Arduino, adafruit, robotis, Mowayduin, sparkfun y demás.

En el caso de la empresa Arduino, esta empresa nace bajo el principio de software libre y hardware accesible y cómodo para realizar proyecto, es un microcontrolador con pines digitales y analógicos programables, conexión serial, bluetooth y la mayoría con la capacidad de conexión vía wifi, ethernet y rfa.

Descripción del Problema:

El proyecto programado consiste en crear un robot virtual el cual va a estar determinado por un nivel de energía. Por lo tanto, el robot va a iniciar una energía de 99 el cual va a ser su nivel máximo de energía y su nivel mínimo será de 0.

El robot va a permitir recibir ciertos comandos los cuales son:

- hello: el robot debe saludar y dar su nombre.
- built: el robot debe dar su fecha de creación
- #n: el robot recibe n de energía.
- status: el robot indica cuánto tiene de energía.
- goahead: el robot camina "hacia adelante". Reduce en uno la energía.
- goback: el robot camina "hacia atrás". Reduce en uno la energía.
- right: el robot gira a la derecha. Reduce en uno la energía.
- left: el robot gira a la izquierda. Reduce en uno la energía.
- dance: el robot baila. Reduce en dos la energía.
- music-on: el robot reproduce música. Reduce en uno la energía.
- music-off: el robot para la reproducción de música.
- smile: el robot ríe. No se modifica la energía.
- cry: el robot llora. Reduce en uno la energía.
- exercise: el robot realiza ejercicios de pesas
- preguntas: el robot responde a ciertas preguntas como que día es hoy, que hora es y cómo se llama

El robot debe dar una alerta al llegar a un valor menor a 20 en energía y va a simular una muerte al llegar a un valor de 0 en energía, en todo momento a valor menores a 20 el robot dará alertas sobre su energía baja.

Se dejó un botón de help, donde se especifican las instrucciones para utilizar el robot. Y otro botón para cerrar por completo la ventana del robot (en caso de cerrar la ventana por la X posiblemente de un error)

Toda la estructura del proyecto debe realizarse con un lenguaje de programación llamado Python 3.6 y la interfaz gráfica debe realizarse con la librería tkinter.

Dificultades Encontradas:

Durante el planteamiento del problema, fue la decisión de crear una animación estilo gif o crear cada uno de los frames y con cual programa iba a realizar este proceso porque en lo personal nunca había diseñado nada. Al final y con la ayuda de Roger Mora utilice el programa de Adobe flash Cs4, Roger se tomó una hora y media de su tiempo para explicarme el funcionamiento del programa.

El segundo problema fue encontrar la mejor forma de simular el Shell, que al final me decidí por varios label y una caja de texto. Además de encontrar la forma de asignarle una función a la tecla "enter" y no utilizar un botón para ingresar los datos.

El tercer problema, el que más tiempo me llevo fue realizar la animación pues había encontrado diversas librerías para Python como PIL, huppy, gifmaker y demás, pero algunas de estas librerías eran compatibles solo una versión de Python o en el caso de PIL simplemente no podía exportar los métodos de la misma por lo tanto no podía usarla. Luego encontré un método llamado spreets que cumplía con todos los requisitos con la limitante que para su mejor rendimiento se debía implementar en una clase y/o utilizar pygame. Ya había logrado llamar de manera muy ineficiente imágenes usando recursividad e hilos, pero al terminar la secuencia esto se caía inmediatamente (dejaba de funcionar). Al final en una tutoría extra Santiago Gamboa el martes 11 explico a 5 personas más la forma correcta y eficiente de llamar invocar una secuencia de imágenes, lo demás fue asignarle lo hilos.

El cuarto problema, fue la implementación de la energía, tenía muy clara que tenía que ser un hilo y que no podía ser una variable global porque tenía que almacenarme el valor aun cuando el programa fuera cerrado y abierto múltiples veces con o sin modificación, por lo que opte por colocarlo en un archivo .txt en la primera línea, para que fuera lo primero que leyera y lo primero que reescribiera. Además, reutilizara este archivo para escribir los datos del robot.

El quinto problema fue la utilización de sonidos .wav y el peso de los mismo, para estos vi unos 30 artículos y tutoriales hasta lograr encontrar uno donde explicaban la frecuencia y el paquete de bits que podía ser cambiado utilizando un programa llamado audacity

El sexto problema es que al final de cada animación la pantalla se queda en negro, esto es porque el hilo termina, y no encontré una forma eficaz de resolverlo, más que colocándole un comando extra que me retornara nuevamente la imagen del robot.

El ultimo problema que tuve fue que no sabía que comandos propios podía realizar, así que postee una foto y un video del funcionamiento del robot y les pregunte a mis amigos que cosas le incluirían y que cosas le quitaría al robot y de ahí surgieron los comandos propios.

Análisis de resultados:

✚ Para las pruebas de las imágenes con hilos (primera vez):

```
def move ():
    contenedor_principal.move("DER",10,0)
def imprimir():
    i = 6
    while(i>0):
        print("funcion1" + str(i))
        imagen1 = cargar_img("canvas2.gif")
        contenedor_principal.create_image(100,100,image=imagen1,tags="DER",anchor = NW)
        i -= 1
        time.sleep(1)
def imprimir2():
    i = 7
    while(i>0):
        print("funcion2" + str(i))
        imagen2 = cargar_img("canvas4.png")
        contenedor_principal.create_image(100,100,image=imagen2,tags = "DER",anchor = NW)
move()
f2= Thread(target = imprimir2, args = ())
f2.start()
f1 = Thread(target = imprimir, args = ())
f1.start()
```

✚ Prueba de imagenes con hilos(edición final):

```
def imagenesl():
    time.sleep(1)
    for i in range(4):
        imagen2 = cargar_img("atleta" + str(i) + ".png")
        label_imagen.config(image = imagen2)
        #print("img"+ str(i))
        time.sleep(0.5)
d = Thread(target = imagenesl,args =())
d.start()
```

✚ Pruebas con txt sin hilos(primera vez):

```
archiv_robot = open("robot_ayuda.txt", "r+")#solo leer "r" y leer y escribir "r+"
#contenido = archiv_robot.read()
final_linea = archiv_robot.tell()

linea1 = archiv_robot.readline()
#linea1 = float(linea1) - 1
archiv_robot.seek(0)
archiv_robot.write("100ii")
archiv_robot.writelines("\nsegunda linea")
#lista = ['1\n']
#lista[0]= int(lista[0])+30
#archiv_robot.seek(0)
#archiv_robot.writelines(str(linea1//1))
archiv_robot.seek(final_linea)
nueva_linea = archiv_robot.read()
#nuevo_contenido = archiv_robot.read()
print (nueva_linea)
```

🚦 Pruebas con txt con hilos(edición final):

```
def power():
    archiv_robot = open("robot.txt","r+")
    final_linea = archiv_robot.tell()
    linea1 = archiv_robot.readline()
    linea2 = float(linea1)- 1
    archiv_robot.seek(0)
    archiv_robot.writelines(str(linea2//1))
    archiv_robot.seek(final_linea)
    nuevo_contenido = archiv_robot.read()
    return(linea2)
    print(nuevo_contenido)

def hilo_energia():
    e = Thread(target = power, args = ())
    e.start()
    status_energia = power()
    label_estatus = Label(ventana_principal,text= "energia = "+str(status_energia+1),bg="black",fg="white")
    label_estatus.place(x=410,y=300)
    if status_energia < 20:
        messagebox.showinfo("alerta","reinicie la energia porque esta muriendo")

    if status_energia < 0:
        murio()
        messagebox.showinfo("alerta","reinicie la energia su robot acaba de morir")
        print ("murio")
```

🚦 Intentos de usar librería para las imagenes:

```
from PIL import Image
imagen = Image.open("prueba.jpg")
imagen.show()
RESTART: C:\Users\Brenes\Dropbox\TEC\intro progra\Ejemplo Tkinter\proyecto robot\pruebas\prueba_pillow.py
Traceback (most recent call last):
  File "C:\Users\Brenes\Dropbox\TEC\intro progra\Ejemplo Tkinter\proyecto robot\pruebas\prueba_pillow.py",
line 1, in <module>
    from PIL import Image
  File "C:\Users\Brenes\AppData\Local\Programs\Python\Python36-32\lib\site-packages\PIL\Image.py", line 56,
in <module>
    from . import _imaging as core
ImportError: DLL load failed: No se encontró el proceso especificado.
```

```
from apng import APNG
t1 = 200
t2 = 1000
atletas = [('atleta-01.png', t1),
            ('atleta-02.png', t1),
            ('atleta-03.png', t1),
            ('atleta-04.png', t1),
            ('atleta-05.png', t2)]

anima = APNG()
for archivo, delay in atletas:
    anima.append(archivo, delay=delay)
anima.save('animatleta2.png')
File "C:\Users\Brenes\AppData\Local\Programs\Python\Python36-32\lib\site-packages\apng\__init__.py", line
69, in chunks
    PIL.Image.open(png).save(f2, "PNG", optimize=True)
NameError: name 'PIL' is not defined
```

Bitácora de Actividades:

Fecha	Actividad	Responsable	Duración
4 de abril	Recibí las indicaciones del programa, verlas, analizarlas y descargar programas para diseñar	Bertha Brenes	2 horas
5 de abril	Consultas y aclaración de la tarea con el profe. Además de creación de la interfaz del Shell. Ver figura #1	Bertha Brenes Jeff Schmidt	3 horas
7 de abril	Consulta con el profe Acerca del Shell. Implementación de la tecla enter Investigación y descargas de librería para crear animaciones en Python. Pruebas con spreets	Bertha Brenes Jeff Schmidt	8 horas
8 de abril	Seguir con pruebas de las animaciones, logre hacer interposición de imágenes con bastante ineficiencia. Empezar a diseñar las animaciones en Paint y anime studio	Bertha Brenes	5 o 6 horas
9 de abril	Seguir probando con la animación. Seguir trabajado con el diseño de la animación	Bertha Brenes	4 horas
10 de abril	Pruebas con varias librerías para gif Seguir trabajando en el diseño	Bertha Brenes	2 horas
11 de abril	Taller de tkinter con Santiago. Eficiencia de las animaciones. Explicación de CS4 por Roger. Por lo	Bertha Brenes Santiago Gamboa Roger Mora	8 horas

	tanto creación de Eva.0		
12 de Abril	Implementación del txt en el programa y la función de energía. Creación de animaciones para 4 comandos Avance de documentación	Bertha Brenes	10 horas
13 de abril	Creación de animaciones finalizando con los comandos principales y la unificación de todas las funciones en la interfaz teniendo terminado el proyecto en una etapa 0.1. Publicación del proyecto en Instagram para coleccionar ideas. Ver Figura2	Bertha Brenes	10 horas
15 de abril	Creación de las funciones propias, mejoramiento de errores e implementación de cosas extras. Intentos de comprimir imágenes y audios Audios se logró, imágenes no bajo menos de 55kb.	Bertha Brenes	8 horas
16 de abril	Terminar documentación Hacer pruebas con familia con Eva	Bertha Brenes Familia de Bertha	4 horas

Estadística de tiempos:

Función	Tiempo
Análisis de requerimientos	4 horas
Diseño de aplicación	8-16 horas
Investigación de funciones	8-12 horas
Programación	20 horas
Documentación Interna	2 hora
Pruebas	5 horas
Elaboración de documento	4 horas
Total	64 horas

Imagen1:

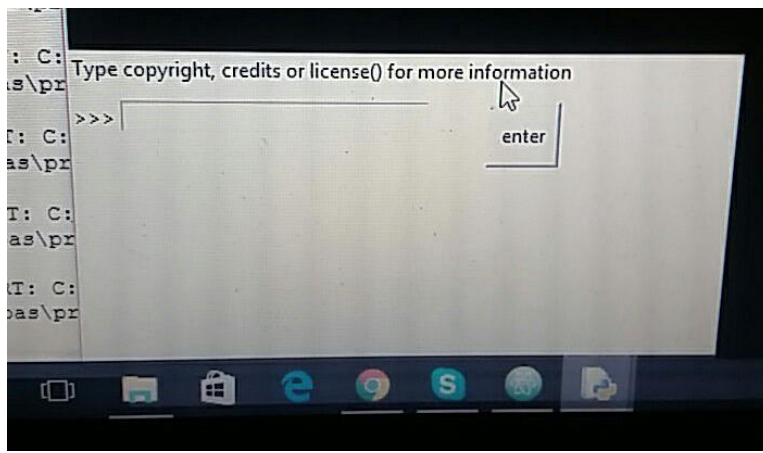
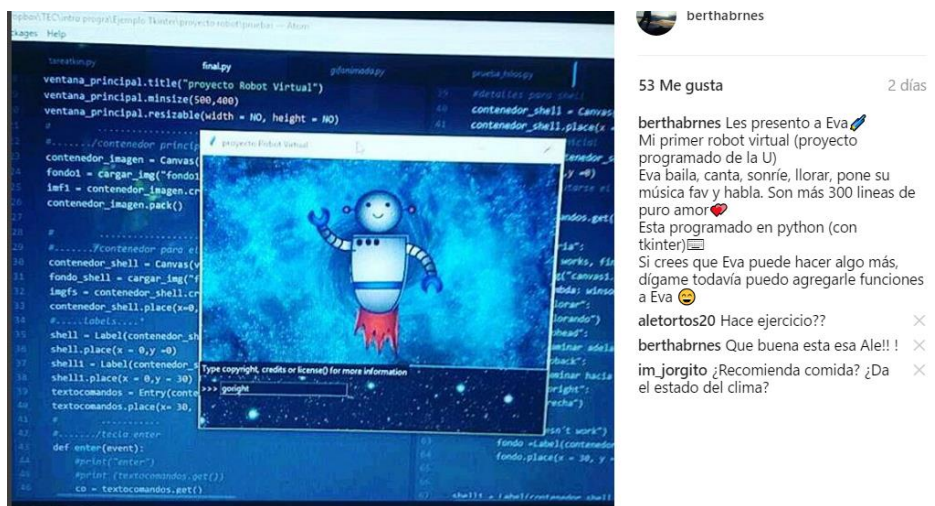


Imagen2:



Conclusión Personal:

Me llevo un aprendizaje muy grande de este proyecto, porque logre entender por completo el funcionamiento de los Thread y tkinter. Pero considero que tkinter es muy limitado y tieso para este tipo de proyectos. Además de que aprendí a diseñar porque nunca había diseñado nada, siempre me consideré una persona poco diestra para el dibujo.

El lunes libero el código en mi cuenta de github para que todo aquel que quiera usarlo y/o modificarlo pueda hacerlo