

Instituto Tecnológico de Costa Rica

Sede Central

Taller de programación

Primer proyecto programado

Robot Virtual

Profesor:

Jeff Schmidt

Estudiante:

Jasson Rodríguez

Tabla de contenidos

Introducción	3
Descripción del problema	4
Dificultades encontradas	5
Análisis de resultados	6
Bitácora de actividades	9
Estadística de tiempos	11
Conclusión	12
Bibliografía	13

Introducción

Como seres humanos, la búsqueda eterna por comprender y mejorar nuestro entorno, por facilitar las cosas o solucionar problemáticas tanto complejas como sencillas de la vida diaria, ha sido el motor que nos ha llevado a renovarnos y desarrollar los avances en la ciencia y tecnología que vemos hoy en día (Hernández, Ortiz, Calles & Rodríguez, 2015).

En la actualidad, los robots son utilizados a diario en una gran cantidad de industrias para la automatización de tareas que de otra manera resultan tediosas, difíciles o peligrosas para los empleados y que además cuestan dinero a la empresa. Sin embargo, los robots no son tan nuevos como parecen:

“El uso del robot industrial, que se identificó como dispositivo único en la década de 1960, junto con los sistemas de diseño asistido por computadora (CAD) y manufactura asistida por computadora (CAM), caracteriza las tendencias más recientes en la automatización del proceso de manufactura (Craig, 2006).

Según Ollero (2007): “Los robots son máquinas en las que se integran componentes mecánicos, eléctricos, electrónicos y de comunicaciones, y dotadas de un sistema informático para su control en tiempo real, percepción del entorno y programación.” A pesar de esto, los robots no siempre han estado constituidos por una máquina mecánica. Según Prieto (2015): Los robots virtuales ya están siendo utilizados en áreas tan diferentes como la atención al cliente, psicoterapia, compra y venta de acciones, y reclutamiento de mano de obra.

Además de los usos industriales que se le dan a los robots, existe una gran variedad de kits de robótica para el aprendizaje y entretenimiento. Esto debido a que la robótica se ha popularizado como una actividad educativa a nivel internacional y que una gran cantidad de centros educativos están implementando. De esta manera, ofrecen a los niños y jóvenes la oportunidad de construir sus propios robots controlados por computadora (Pittí, 2011).

Descripción del problema

El problema consiste en el diseño, programación y verificación de un robot virtual programado en Python con la utilización de la interfaz gráfica Tkinter. El robot es capaz de ejecutar una cierta cantidad de comandos ingresados mediante una consola siempre y cuando cuente con la energía necesaria para realizar las acciones.

El robot realiza las acciones ingresadas de forma animada y con sonidos siempre que sea necesario. Además, guarda antes de salir en un archivo de texto los datos básicos como su nombre, imagen, fecha de creación y energía restante.

Los comandos aceptados por el robot son:

- ✱ hello: el robot debe saludar y dar su nombre.
- ✱ built: el robot debe dar su fecha de creación.
- ✱ power(n): el robot recibe n de energía. La energía es un valor entre 0 y 100 y debe mostrarse en todo momento.
- ✱ status: el robot indica cuánto tiene de energía. Debe dar alguna alerta si la energía es menor a 20.
- ✱ 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.
- ✱ jump: el robot salta. Reduce en uno la energía.
- ✱ sing: el robot canta.

Dificultades encontradas

La principal dificultad encontrada durante la programación del robot virtual fue que al cambiar la imagen del robot dentro de una función utilizando una variable local, Python desecha la variable local para evitar la acumulación de basura y la imagen es borrada. Para solucionarlo se siguieron los pasos descritos por Lundh (2005):

Note: When a PhotoImage object is garbage-collected by Python (e.g. when you return from a function which stored an image in a local variable), the image is cleared even if it's being displayed by a Tkinter widget.

To avoid this, the program must keep an extra reference to the image object.

A simple way to do this is to assign the image to a widget attribute, like this:

```
label = Label(image=photo)
```

```
label.image = photo # keep a reference!
```

De esta manera se creó un atributo de la clase robot llamado image donde se guardó la variable con la imagen a utilizar.

Otro problema que surgió durante la verificación del programa fue que no todos los tipos de errores retornaban “Comando no válido” para solucionar esto fue necesario cambiar la excepción NameError por la excepción Exception que abarca un rango más amplio de errores.

Análisis de resultados

Las pruebas fueron llevadas a cabo introduciendo una serie de comandos preestablecidos, uno a la vez en la consola, utilizando como parámetro inicial un valor de energía de 1; asimismo la salida del programa fue copiada sin realizar modificaciones y se anotó aquellos casos en que exista salida de audio o animación. Además, en caso de que se presentara un error se procedió a reiniciar la instancia del programa. A continuación se observan los resultados de las ejecuciones de prueba:

```
>>>hello
```

Hola, mi nombre es Nohe (Reproduce audio)

```
>>>built
```

Fui creado el día 5 de abril de 2017 (Reproduce audio)

```
>>>power(20)
```

(Añadió la cantidad indicada de energía)

```
>>>status
```

La energía restante es del 21%

```
>>>goahead
```

(Reproduce la animación)

```
>>>goback
```

(Reproduce la animación)

```
>>>left
```

(Reproduce la animación)

```
>>>left
```

(Reproduce la animación)

```
>>>goahead
```

(Reproduce la animación)

```
>>>goback
```

(Reproduce la animación)

```
>>>right
```

(Reproduce la animación)

```
>>>right
(Reproduce la animación)
>>>goahead
(Reproduce la animación)
>>>goback
(Reproduce la animación)
>>>left
(Reproduce la animación)
>>>dance
(Reproduce la animación)
>>>music_on
(Reproduce audio)
>>>music_off
(Detiene la reproducción del audio)
>>>smile
(Reproduce la animación y el audio)
>>>cry
(Reproduce la animación y el audio)
>>>jump
(Reproduce la animación)
>>>sing
(Reproduce la animación y el audio)
>>>status
Batería baja! Por favor conecte a la corriente.
4% restante. (Reproduce audio)
>>>hello()
Hola, mi nombre es Nohe (Reproduce audio)
>>>hello(hello)
(TypeError)
>>>power(-1)
Valor de energía no aceptado>>>
```

```
>>>hola
```

Comando no válido

```
>>>baila
```

Comando no válido

```
>>>built(
```

(Error de sintaxis)

```
>>>power(43
```

(Error de sintaxis)

Después de arreglar los errores los resultados obtenidos fueron los siguientes:

```
>>>hello(hello)
```

Comando no válido

```
>>>power(-1)
```

Valor de energía no aceptado

```
>>>hola
```

Comando no válido

```
>>>baila
```

Comando no válido

```
>>>built(
```

Comando no válido

```
>>>power(43
```

Comando no válido

Bitácora de actividades

5/04/17 9:30 a 10:30: Análisis de requerimientos

5/04/17 10:50 a 11:15: Investigación

5/05/17 9:00 a 9:10: Programación: Importación de librerías y definición de la ventana principal

5/05/17 9:20 a 9:40: Investigación de manejo de strings

5/05/17 9:40 a 11:00: Programación: algoritmo de serialización de datos

5/05/17 1:40 a 2:40: Programación

6/05/17 11:40 a 7/05/17 12:10: Análisis de requerimientos

7/05/17 12:10 a 1:30: Diseño de los gráficos del robot

7/05/17 1:30 a 2:00: Diseño: Búsqueda de canción

9/05/17 12:00 a 12:50: Programación para ejecutar comandos desde el shell

9/05/17 5:45 a 6:30: Programación para reproducir música y sonidos

10/05/17 9:50 a 9:55 Programación de la función built

10/05/17 3:40 a 5:15: Programación de la función power y del indicador de batería

10/05/17 5:15 a 5:25: Investigación de protocolos

10/05/17 9:00 a 9:15: Programación del indicador de la batería

10/05/17 9:45 a 10:15: Programación de las funciones status y smile. Problema con cambiar la imagen del robot.

12/05/17 3:00 a 5:30: Programación de las funciones de movimiento

12/05/17 10:00 a 10:45 Programación de la función dance

12/05/17 10:45 a 10:50 Análisis de requerimientos

12/05/17 11:15 a 11:20: Programación de la función cry

13/05/17 11:30 a 12:10: Diseño de las funciones propias

13/05/17 2:15 a 2:40: Programación de la función jump

13/05/17 4:30 a 5:30: Programación de la función sing y creación de la ventana ayuda

13/05/17 8:40 a 9:10: Programación del texto del cuadro de ayuda

13/05/17 10:55 a 11:05: Grabación de las voces

14/05/17 9:20 a 10:00: Búsqueda de los efectos de sonidos y grabación de sing

14/05/17 1:40 a 2:00: Programación para la muerte del robot

14/05/17 2:30 a 3:00: Diseño de robot muerto

14/05/17 3:00 a 3:30: Programación adicional para la muerte del robot

14/05/17 5:20 a 7:40: Redacción de la introducción

15/05/17 10:45 a 12:15: Redacción de la descripción del problema y de los problemas encontrados

15/05/17 8:25 a 9:30: Verificación del programa

16/05/17 12:20 a 12:30: Programación para la corrección de errores

16/05/17 12:40 a 1:40: Documentación interna

16/05/17 2:00 a 2:20: Bitácora y estadística de tiempos

16/05/17 10:30 a 11:30: Investigación de reconocimiento de voz

16/05/17 4:50 a 6:30: Programación de reconocimiento de voz

16/05/17 7:40 a 8:00: Programación de ajustes finales

16/05/17 8:00 a 8:25: Redacción de la conclusión

16/05/17 8:25 a 8:30: Actualización de la bitácora y estadística de tiempos.

Estadística de tiempos

Función	Tiempo (h)
Análisis de requerimientos	1.6
Diseño de la aplicación	3.3
Investigación de funciones	1.9
Programación	14.9
Documentación interna	1
Pruebas	0.1
Elaboración documento	4.7
Total	27.5

Conclusión

Se construyó un robot virtual cumpliendo con las especificaciones planteadas, incluyendo también la función de reconocimiento de voz, lo que le permite recibir comandos orales y escritos.

Bibliografía

Craig, J. J. (2006). Robótica. Mexico: Pearson educación.

Hernández, M., Ortiz, M., Calles, C. & Rodríguez, J. (2015). Robótica: Análisis, modelado, control e implementación. Tapaliupas, México: OmniaScience

Lundh, F. (2005). An introduction to Tkinter. Recuperado el 15 de abril de 2017 de:
<http://effbot.org/tkinterbook/>

Ollero, B. A. (2007). Robótica: Manipuladores y robots móviles. Barcelona, España: Marcombo Boixareu.

Pittí, K. (2011). Metodologías de aprendizaje colaborativo a través de las tecnologías. Salamanca, España: Ediciones Universidad de Salamanca

Prieto, E. (2015). De cerebro a civilización. Bloomington, Indiana: Palibrio