Instituto Tecnológico de Costa Rica	Primer Proyecto Programado
Escuela de Ingeniería en Computación	ROBOT VIRTUAL
CE-1102 Taller de Programación	Prof: Jeff Schmidt Peralta
I Semestre 2017	Consultas: grupo facebook
4 de abril de 2017	correo: jschmidtcr@gmail.com

1. Introducción.

Se va a desarrollar un robot virtual muy básico, implementando múltiples funciones utilizando los conceptos de listas, archivos planos y una interfaz gráfica, por medio de programación recursiva.

2. Los robots virtuales.

Un robot es una entidad virtual o mecánica artificial, un sistema electrónico y mecánico realiza acciones o movimientos. La palabra robot normalmente se refiere a mecanismos físicos, pero también a sistemas virtuales de software, por ejemplo, sistemas de backup automatizados.

Existen diferentes tipos y clases de robots, por sus capacidades normalmente se clasifican en 4 formas:

- 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.
- **Poliarticulados:** para uso industrial, normalmente en tareas repetitivas.

El uso de robots en educación ha tenido un gran impulso, se puede profundizar el tema investigando sobre herramientas como LEGO WeDo, LEGO Mindstorms, Robotis y Mowayduino.

El robot virtual que el estudiante va a crear (debe definir un nombre y una imagen para SU robot) es una herramienta sencilla que permite el entretenimiento de aquellos que la utilicen y que integra:

- Manejo de una consola (Shell).
- Manejo de unidades de tiempo.
- Gráficos, sonidos y otros

3. Descripción Funcional Programa.

La presentación de la interfaz con el usuario es **totalmente libre** y será un elemento importante dentro de la calificación del proyecto. Se debe utilizar Tkinter como interfaz gráfica base para el proyecto.

El programa debe mostrar al robot y algún tipo de consola que permita darle órdenes básicas, va a funcionar por medio de comandos, se verá una pequeña gramática (forma de escribir las expresiones en el lenguaje).

1. Descripción del robot:

El programa va a ser interactivo, se va a ejecutar desde una consola (puede ser similar a la de Python), de forma que cuando el usuario escriba alguna orden en la ventana de comandos, el resultado se mostrará en esa misma consola.



Los datos básicos para el robot deben almacenarse en un archivo de texto llamado "robot.txt". A continuación los datos *básicos* del robot.

Nombre: nombre del robot definido por su creador

Imagen: imagen del robot (ruta del archivo)
Fecha de creación: fecha en que se creó el robot
Energía: con valores discretos entre 0 y 100.

2. Comandos permitidos:

La ventana de comandos (consola) permite interactuar al usuario con el robot. Por medio del indicador o prompt (ejemplo >>> en el caso de Pyhton), se le indica al usuario que se espera su comando.

Comandos permitidos en la Command Window:

 $\verb|hello: el robot debe saludar y dar su nombre.|$

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.

own1: el robot hace una labor a definir por el programador.

own2: el robot hace una labor a definir por el programador.

En caso que se logren implementar comandos de voz para que el robot realice las acciones, se va a considerar un opcional con 10 puntos adicionales.

Las respuestas básicas pueden darse por medio de la misma consola, pero preferiblemente por medio de sonidos. Las acciones que implican movimiento, deben presentarse por medio de alguna animación.

3. La ventana de ayuda (Help)

Permite obtener ayuda sobre la utilización del programa y sobre las funciones que realiza el robot.

Consideraciones a realizar.

• Debe guardarse el estado del robot y recuperarse la próxima vez que se ejecute el programa.

3

4. Funciones a investigar.

Para mejorar la calidad y presentación de la tarea, debe **investigarse** el uso de algunas funciones referentes a validaciones de datos y despliegue de información. Las funciones que podrían utilizarse, entre otras son:

- Utilización de multimedia: integración de animaciones, sonidos y otros.
- Generación de números aleatorios
- Manejo de archivos de texto

5. Documentación.

La documentación interna en el programa fuente, debe contener antes de definir cada función, al menos una explicación de lo que realiza la función, las entradas, salidas y restricciones consideradas.

La documentación externa debe incluir:

- Tabla de contenidos
- Introducción
- Descripción del problema.
- Dificultades encontradas: problemas en el desarrollo y que se hizo para corregirlos
- Análisis de resultados. (incluyendo corridas de ejemplo)
- Bitácora de actividades: se deben ir anotando todas las actividades, tipo de actividad, su descripción, responsable y duración.
- Estadística de tiempos: un cuadro que muestre un resumen de la Bitácora de Actividades en cuanto las horas **REALES** invertidas. Ejemplo:

FUNCION	Tiempo
Análisis de requerimientos	xx horas
Diseño de la aplicación	xx horas
Investigación de funciones	xx horas
Programación	xx horas
Documentación interna	xx horas
Pruebas	xx horas
Elaboración documento	xx horas
TOTAL	xx horas

• Conclusión personal.

6. Evaluación.

Documentación 15%
Interna 5%
Externa 10%

Resultados (ejecución, eficiencia, manejo correcto de estructuras planteadas, presentación)

Funciones:

Nombre e imagen del robot 5 % Manejo interfase general 15 % (creatividad-presentación) Manejo de la consola 10% Manejo de archivos 5% Comandos (3 puntos c/u) 45% hello built power status goahead goback right left dance music-on music-off smile cry own1 own2 5% Pantalla de ayuda

7. Aspectos Administrativos.

- La tarea es individual y se debe entregar hasta el día 16/04/2017 hasta las 11:59 pm, en forma electrónica, en un archivo comprimido con el nombre del estudiante, que contenga TODO lo necesario para poder ejecutarla. Luego se asignará una cita de revisión del programa. No se aceptarán tareas después de la fecha y hora indicadas. Debe enviarse un archivo readme.txt con la versión de Python a utilizar para la revisión y alguna otra indicación que se considere importante. Se debe enviar al correo tareasintrotaller.ce@gmail.com
- La defensa o revisión del proyecto es indispensable. En esta revisión se preguntará sobre aspectos relacionados con funcionalidad, así como sobre el código. El estudiante debe mostrar TOTAL dominio de estos dos temas, de lo contrario, el proyecto puede ser considerado como una copia.

- En caso de probarse algún tipo de fraude en la elaboración de la tarea se aplicarán todas las medidas indicadas al inicio del curso, incluyendo una carta al expediente del estudiante.
- Se debe incluir en el archivo comprimido la documentación solicitada. Debe entregarse en formato electrónico (archivo .doc .odt o .pdf).
- No se aceptarán tareas cuyo archivo sobrepase 2.0 mb de espacio en disco.
- Cualquier falta a los aspectos aquí enunciados implicará pérdida de puntos.
- El profesor se reserva el derecho de calificar forma y fondo de las actividades tomando como referencia la mejor actividad presentada

7. Bibliografía.

Documentación técnica Python

8. Consultas.

Puede dirigir cualquier consulta a jschmidtcr@gmail.com o al grupo del curso en facebook.