Computer Engineering

CE-1102 Taller de Programación

Project 2

Sensor Bot

(Inspired on Prof Jeff Schmidt Project)

Prof. Milton Villegas Lemus (G2)

T.A. Marcelo Truque Montero

Team work up to 2 students

Costa Rica,October 20, 2022

Introducción

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.

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, Robotics y Mowayduino.

El robot virtual que el estudiante va a crear (debe definir el nombre e imagen para su robot) a crear es una herramienta sencilla que integra:

- Definición de objetos, sus datos y métodos.
- Manejo de unidades de tiempo.
- Generación de eventos aleatorios.
- Gráficos, sonidos y otros

Descripción

Se debe montar una simulación virtual de movimientos de un robot, con una interfaz donde se expondrá cómo funcionará el robot.

Primeramente se tendrá un menú donde el usuario podrá escribir características propias del robot, **el cual debe ser modelado con programación orientada a objetos**. Estas características incluyen el nombre, la fecha de creación del robot, el modelo y la empresa fabricante. Además, el robot deberá llevar internamente un registro de la cantidad de mediciones efectuadas durante su vida útil. Después se pasará a una pantalla del diseño de escenarios.

Para simular sus movimientos el escenario funcionará como una cuadrícula, de modo que el robot pueda moverse por posiciones o coordenadas similar a un juego de batalla naval. El mismo escenario deberá contar con obstáculos, de modo que el

robot no pueda ocupar ciertas casillas de la cuadrícula, en la figura 1 se puede ver un ejemplo del escenario.

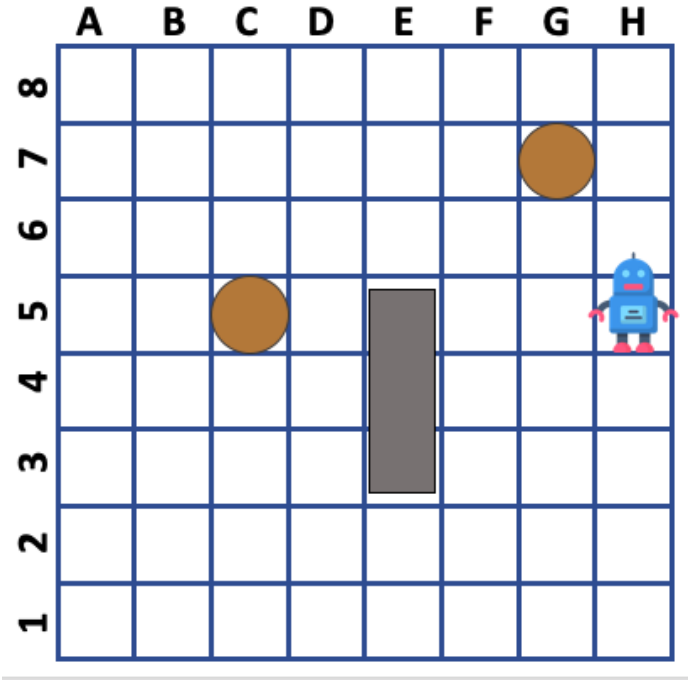

Figura 1. Ejemplo escenario para el funcionamiento de la interfaz

El escenario debe ser diseñado por el usuario de modo que se deben definir mínimo 3 tipos diferentes de obstáculos. Debe tomar en cuenta que el usuario no puede bloquear la aparición del robot con un obstáculo. Se recomienda al estudiante utilizar una matriz binaria representativa del escenario. La matriz equivalente al escenario de la figura 1 se puede observar en la figura 2.

```
((0,0,0,0,0,0,0,0),
 (0,0,0,0,0,0,1,0),
 (0,0,0,0,0,0,0,0),
 (0,0,1,0,1,0,0,0),
 (0,0,0,0,1,0,0,0),
 (0,0,0,0,1,0,0,0),
 (0,0,0,0,0,0,0,0),
 (0,0,0,0,0,0,0,0),)
```

Figura 2. Matriz a modo de ejemplo de la representación interna del escenario

La cuadrícula debe de tener internamente un valor de alguna medición definida por el estudiante (humedad, temperatura, iluminación…), el cual también se recomienda sea guardado en una matriz equivalente. El diseño del escenario no debe ser necesariamente utilizando programación orientada a objetos pero se recomienda utilizar la misma.

El robot tendrá dos modos de funcionamiento, mediante órdenes o piloto automático; mediante órdenes el usuario le indica al robot todos los pasos que debe seguir, y donde debe recolectar el valor interno de cada cuadrícula para generar un reporte. Los posibles movimientos que el robot puede hacer es avanzar hacia adelante, girar hacia la derecha, girar hacia la izquierda 90 grados o efectuar la medición del sensor. El reporte será el promedio de mediciones recolectadas (simuladas es decir generadas en forma aleatoria en intervalos de valores definidos y en intervalos de tiempo establecidos), el cual debe ser guardado en un txt junto con la fecha y hora de la generación del reporte y la magnitud de cada medición . El robot debe de poder identificar obstáculos y notificar al usuario que no puede seguir sus órdenes si estos se lo impiden. Si se encuentra en modo piloto automático el usuario le indica la posición en la cuadrícula donde el robot debe andar y recolectar el dato, en modo piloto automático no se genera reporte sino que directamente en la interfaz se indica la magnitud de la medición efectuada, en este modo el robot debe llegar ahí evadiendo por sí solo los obstáculos y debe notificar al usuario si la posición elegida es inaccesible (se eligió una posición con un obstáculo encima).

Adicionalmente, la interfaz debe contar con un modo de reset que limpie el documento de reportes y permita al usuario redefinir el escenario. También debe contar con un botón de presentación donde se imprimirá en consola una ficha de presentación del robot en la cual debe aparecer su nombre, su modelo, empresa fabricante, su fecha de creación y la cantidad total de mediciones efectuadas. A continuación se muestra un ejemplo de cómo puede verse la ficha:

"*Hola! Soy Marbot modelo 1.3 y trabajo para TEC. He efectuado un total de 100 mediciones de temperatura*""

EVALUACIÓN

ASPECTOS DEL PROGRAMA 62 pts

Pantalla Inicial ● Captura de la información del robot ● Botón creación escenario	5
Pantalla de creación escenario ● Diferentes tipos de obstáculo ● Personalización ● Reponerse a errores	10
Pantalla Acerca de ● Información completa ● Retorno a pantalla de inicio	5
Pantalla Simulación ● Cambio modos ● Funciones básicas del robot ● Botón presentación del robot	20
Manejo Obstáculos ● Notificación choque en modo manual ● Evasión obstáculos modo automático ● Se mantiene en los límites del escenario	12
Generación reportes ● Reporte en un archivo txt ● Promedio mediciones ● Medición en modo manual	10

Documentación 38pts

Introducción	1
Conclusiones	5
● Recomendaciones	4
Análisis de Resultados ● Diagrama de módulos ● Plan de Pruebas	4

Bitácora	6
Literatura o Fuentes	1
Módulo de Generación de Autodocumentación de los Módulos Principales	2
Auto documentación de módulos sigue formato	3
Documentación Interna en partes de los módulos más importantes	2
Documentación de Reglas de Grupo y Roles	5
Actividades, fechas de entrega y coevaluación	5

Notas

● El proyecto programado es en parejas y se debe entregar a más tardar el viernes de 4 noviembre 2022 hasta las 23:00 en forma electrónica. Cada grupo debe incluir un archivo readme.txt con la versión de Python (>=3.3) a utilizar para la revisión y alguna otra indicación que se considere importante. Se debe subir al en la asignación de Classroom abierta al efecto. Cada archivo debe enviarse con el nombre construido de la siguiente forma: Inicial del nombre en mayúscula, primer apellido y segundo apellido del primer integrante, guión bajo, inicial del nombre en mayúscula, primer apellido, y segundo apellido del segundo integrante y extensiones py, txt y pdf.

● Está prohibido el uso de iteración, se debe utilizar la recursión en para cualquier ciclo

● Cualquier duda, omisión o contradicción en la especificación se debe aclarar con el profesor y se difundirá por el Classroom (Google) del grupo de Taller y con el asistente por medio del grupo de Telegram.

● Se debe de enviar la documentación (archivo .pdf) ésta debe contar con lo siguiente: Introducción, conclusiones, recomendaciones, diagramas de módulos, plan de pruebas, bitácora y literatura o fuentes consultadas.

● Además, debe adjuntar un programa que genere la auto-documentación con el método print de Python para los principales módulos de su programa. El resto de la auto-documentación debe estar disponible en la defensa.

● El código debe estar suficientemente documentado de tal forma que el grupo se pueda orientar en él fácilmente durante la defensa. Debe contar con auto documentación interna para ser generada mediante el método print de Python.

● El proyecto se debe defender previa cita con el profesor. El profesor publicará unas fechas de defensa. El grupo debe inscribirse para defender el proyecto, de no asistir a la defensa, únicamente obtendrá la nota correspondiente a la documentación del proyecto.

● Cualquier clase de copia de código será sancionada de acuerdo con el reglamento vigente y se llevará hasta la consecuencia de amonestación con carta al expediente. Código adoptado para el manejo de interfaz debe especificarse claramente la fuente y reconocer los créditos. Está prohibida la copia de código que involucre la solución lógica general del algoritmo.

● Si en la defensa no demuestra su autoría con el dominio propio de esa calidad solo se le otorgarán los puntos correspondientes a los obtenidos en la documentación.