Programando el hub RI5 con Python

Hola, soy Shah. Soy un formador profesional especializado en software y hardware MacOS e iOS, y también imparto clases sobre desarrollo de aplicaciones iOS y LEGO Mindstorms.

En la última lección, aprendiste cómo crear su primer programa Python. Hoy, aprenderás más sobre la interfaz de usuario de programación de Python, aprenderás sobre el programa que creó y verás la documentación de programación en la aplicación Robot Inventor. También aprenderás cómo hacer que tu robot se mueva hacia adelante y hacia atrás. ¿Estás listo? ¡Estupendo! ¡Vamos!

Que necesitas:

- LEGO Mindstorms Robot Inventor (set no. 51515)
- Una computadora (Mac / PC), tableta (iOS / Android) o teléfono (iOS / Android) con la aplicación Mindstorms Robot Inventor instalada

Antes de que empieces

Para esta lección, necesitarás construir el robot Tricky. Asegúrese de que las baterías de su bloque programable y su dispositivo estén cargadas.

La interfaz de usuario de programación de Python

- Escribe el código en el área de edición.
- Al hacer clic en el icono de la Guía de programación, se muestran los detalles de los comandos de Python que puede utilizar.
- Los controles de zoom le permiten establecer el nivel de zoom predeterminado, alejar y acercar.
- Los botones Deshacer / Rehacer le permiten deshacer o rehacer sus acciones.
- La consola muestra mensajes de error y otra información, y al hacer clic en el botón Consola, se muestra.
- El botón de ranura de programa le permite elegir qué ranura del concentrador se utilizará para almacenar el programa.
- El botón Detener detiene cualquier programa en ejecución.
- El botón Reproducir copia su programa en la ranura seleccionada y lo ejecuta.
- Ahora que está familiarizado con la interfaz de usuario, aprendamos un poco más sobre el programa que creó en la última lección.

Sobre su programa

En la última lección, simplemente ejecutó el programa que creó sin aprender mucho sobre él. Veámoslo con más detalle.

Las primeras cuatro líneas de su programa importan las diversas bibliotecas y otro código necesario para que su robot funcione.

El comando hub = MSHub () crea una representación de su robot, Tricky en este caso. Necesita hacer esto para poder darle comandos.

La última línea de código le dice a Tricky que emita un pitido.

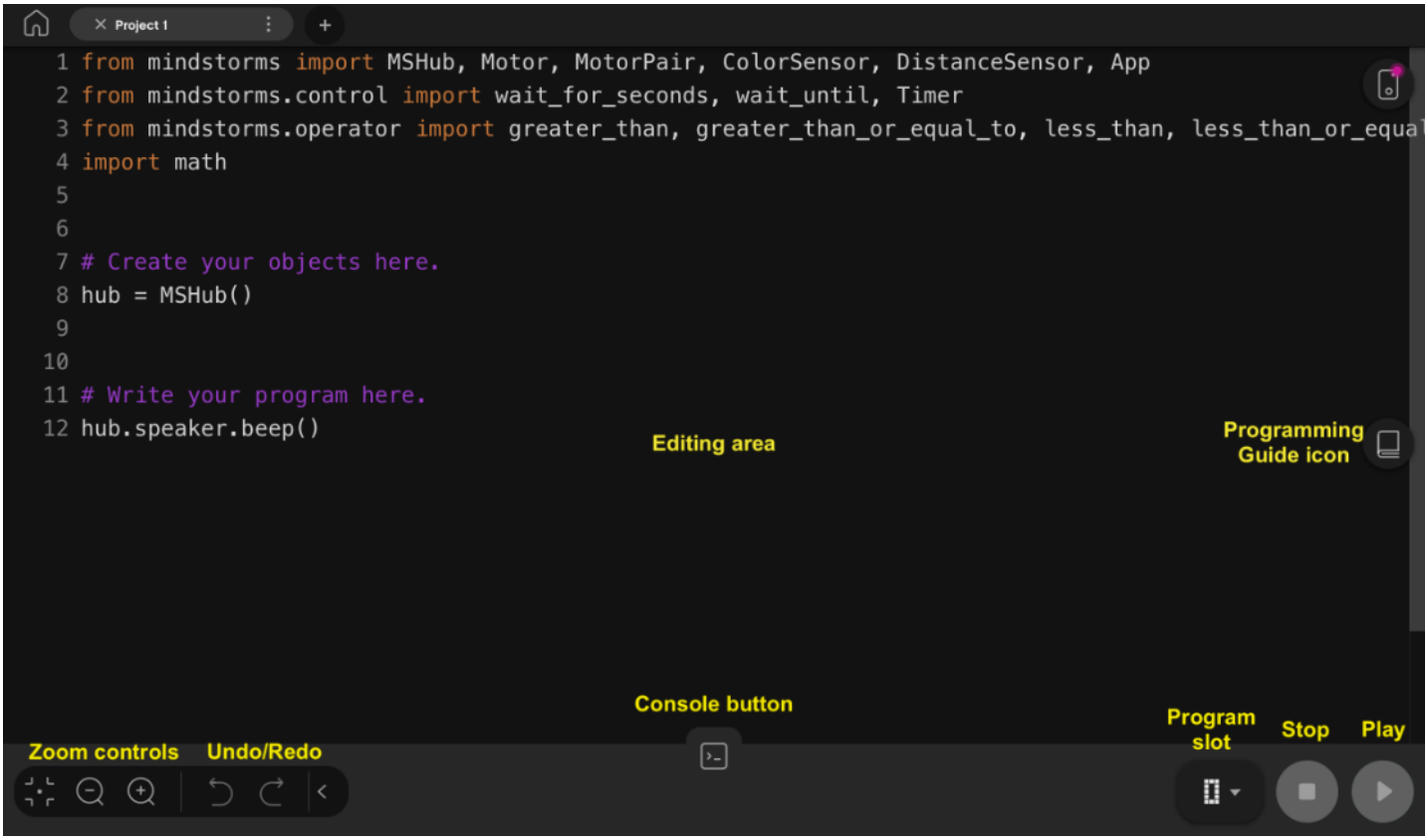
Ahora modifiquemos su programa para que Tricky avance y retroceda.

Haciendo que su robot se mueva recto

Modificarás el programa que usaste en la última lección para hacer que Tricky avance y retroceda. Sigue estos pasos.

1. Haga clic en el icono de la Guía de programación y busque en la documentación detalles sobre los pares de motores. Lea sobre cómo inicializar pares de motores y cómo emitir comandos de pares de motores. Está bien si no lo entiende, ya que se explicará en los pasos siguientes.

2. Elimine la línea de código que hace que el robot emita un pitido. Agregue la siguiente línea de código a su programa para crear un objeto motorPair y asígnelo a una variable motorPair:



```
motorPair = motorPair( A , B )
```

El objeto mototrPair representa la configuración del motor en Tricky. Tenga en cuenta que Tricky tiene dos motores con una rueda conectada a cada uno. El motor izquierdo está conectado al puerto A y el motor derecho está conectado al puerto B. Una vez que se ha creado el objeto motorPair, puede programarlo para que su robot se mueva y gire.

3. Agregue las siguientes líneas de código a su programa para hacer que Tricky avance:

```
motorPair.set_default_speed(30)
motorPair.move(2, rotations , steering=0)
```

La primera línea establece las velocidades de los motores en 30. Puede establecerlas en cualquier velocidad entre -100 y 100. La segunda línea hace que Tricky avance hasta que las ruedas hayan completado dos rotaciones. Establecer la dirección en 0 hace que ambos motores se muevan a la misma velocidad.

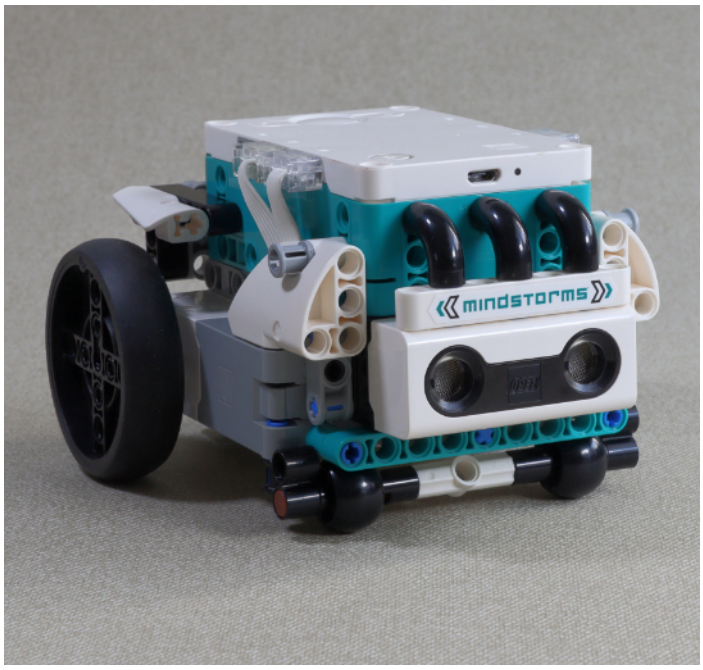
4. Agregue la siguiente línea de código para hacer que Tricky se detenga por un segundo:

```
wait_for_seconds(1)
```

5. Agregue las siguientes líneas de código para hacer que Tricky conduzca al revés:

```
motorPair.set_default_speed(-30)
motorPair.move(720, degrees , steering=0)
wait_for_seconds(1)
```

La primera línea establece las velocidades de los motores en -30. Dado que establecer la velocidad predeterminada en un valor positivo hace que Tricky conduzca hacia adelante, establecerlo en un valor negativo hace que Tricky conduzca hacia atrás. La segunda línea hace que Tricky conduzca hacia atrás hasta que las ruedas hayan girado 720 grados (2 rotaciones completas). Como antes, poner la dirección en 0 hace que ambos motores se muevan a la misma velocidad. La tercera línea hace que Tricky se detenga por un segundo como lo hizo antes.



6. Agregue las siguientes líneas de código para hacer que Tricky avance nuevamente:

```
motorPair.set_default_speed(30)
motorPair.move(2, seconds , steering=0)
```

La primera línea vuelve a establecer las velocidades de los motores en 30, y la segunda línea hace que Tricky avance durante dos segundos.

7. Verifique que el programa completo se parezca a la captura de pantalla a continuación. Ejecute su programa. Tricky debe avanzar dos rotaciones, retroceder dos rotaciones y volver a avanzar durante dos segundos.

¡Acabas de programar a Tricky para que avance y retroceda! ¡Hurra!
En la siguiente lección, harás que Tricky gire mientras te mueves.
¡Hasta entonces!