El reto simula un cuarto de dimensiones 1200 x 700. Dentro del cuarto se encuentra un robot en la esquina inferior izquierda y una meta en la esquina superior derecha, tal como se muestra en la Figura 1. El objetivo del reto consiste en mover el robot desde su punto inicial hasta la meta, evitando los obstáculos que se puedan encontrar en el cuarto.

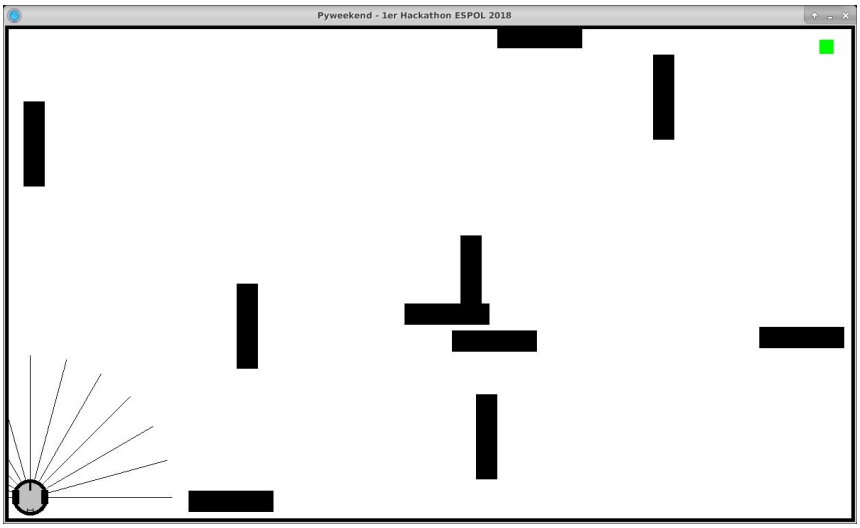


Figura 1. Ambiente de pruebas del reto.

El robot tiene forma circular (diámetro = 51) y cuenta con los siguientes métodos (funciones):

Percepción

`robot.get_pos()`
Retorna un tupla indicando la ubicación del centro del robot dentro del cuarto, según el sistema de coordenadas de la Figura 2. Por ejemplo, en su posición inicial robot.get_pos() retorna: (35, 665).

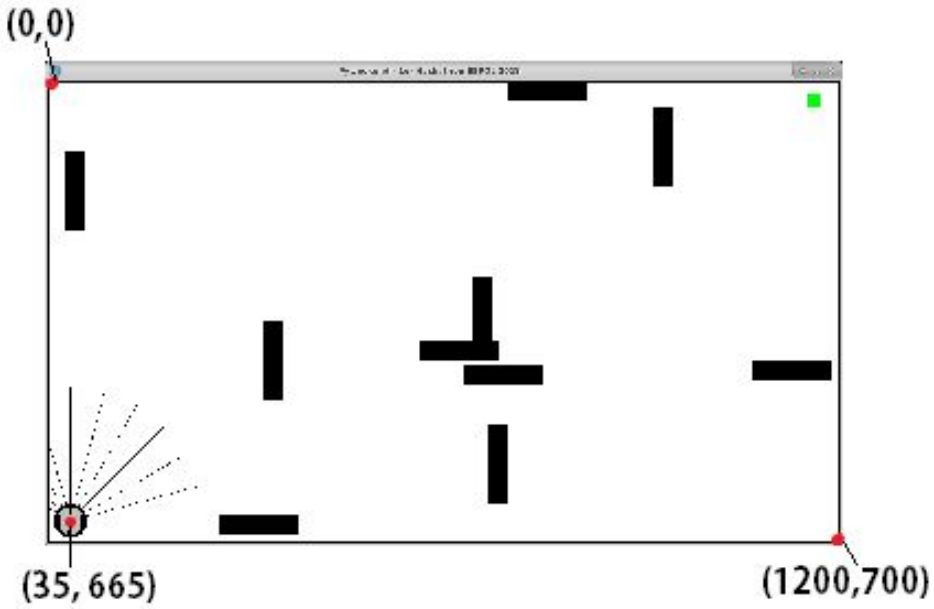
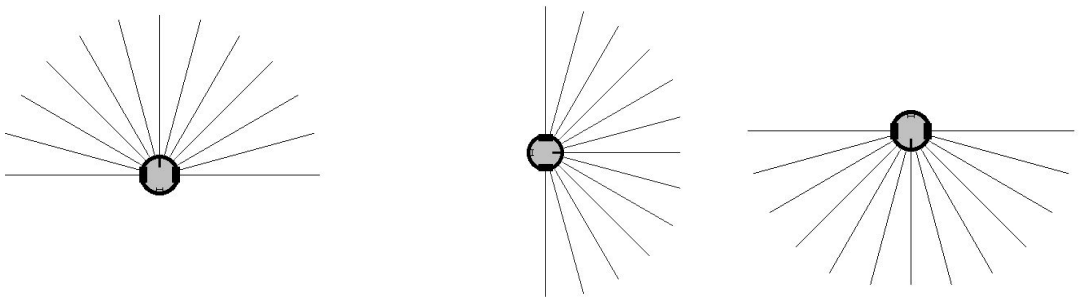


Figura 2. Sistema de Coordenadas

`robot.get_angle()`
Retorna el ángulo en el que se encuentra orientado el robot. El robot puede retornar valores positivos o negativos dependiendo de hacia dónde realizó el giro partiendo desde cero. El valor cero representa que el robot se encuentra orientado hacia arriba, a partir de esta posición se consideran como positivos los giros hacia la izquierda y como negativos los giros hacia la derecha (Figura 3). Los valores de ángulo siempre se mantienen entre -360 y +360.



0 -90 -180

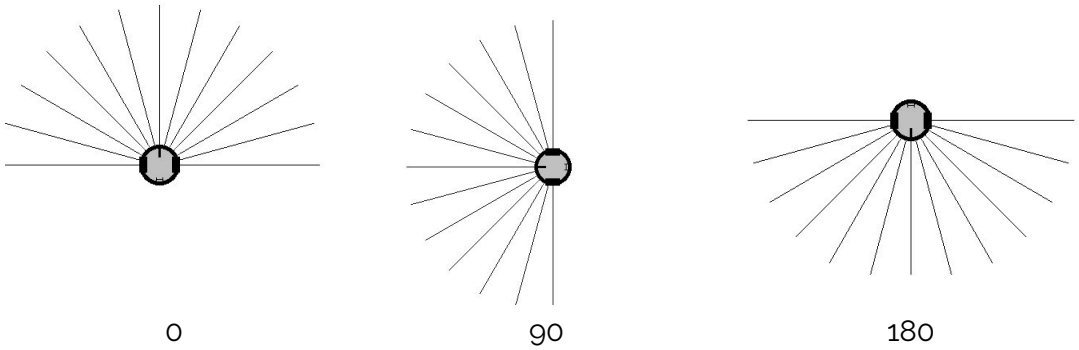


Figura 3. Valor de ángulo del robot cuando realiza:
giro a la derecha (fila superior), giro a la izquierda (fila inferior)

`robot.get_collision()`
Retorna un valor booleano que indica si el robot ha chocado contra algún obstáculo.

`robot.read_sensors()`
El robot consta de un anillo de sonares que le permiten conocer la distancia a los objetos en su ambiente. El anillo se sonares está compuesto por 13 sensores de distancia ubicados cada 15 grados, comenzando desde la izquierda del robot y finalizando a la derecha. El máximo alcance de cada sensor es de 200. El método read_sensors() retorna una lista con los valores de cada uno de los sensores. Por ejemplo, para el robot de la Figura 4, robot.read_sensors() retorna [170, 175, 195, 200, 200, 200, 200, 200, 200, 200, 115, 135, 200]

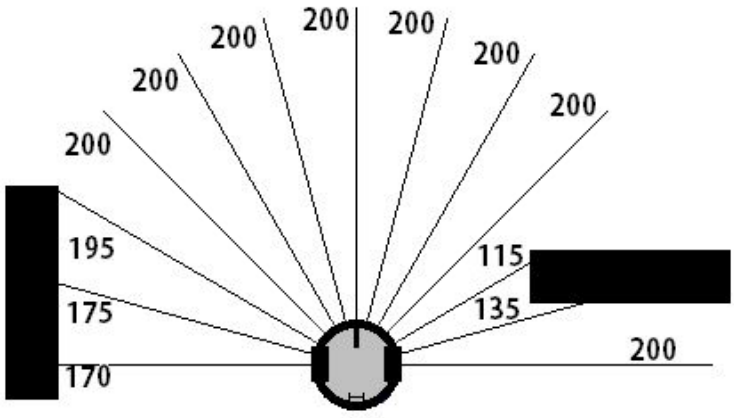


Figura 4. Lectura de sonares del robot

`robot.get_traces()`
Retorna una lista de tuplas con las posiciones recorridas por el robot. En la Figura 5 se representa esta lista por medio de puntos en azul.

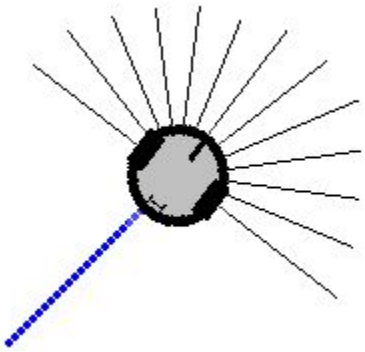


Figura 5. Recorrido del robot (traces)

Actuadores

En el reto se simula a un robot de manejo diferencial (2 motores, uno de cada lado) sin embargo se ha limitado el control de los motores, por lo cual el movimiento del robot se puede realizar por medio de 2 funciones solamente.

`robot.spin(r_step_theta)`
Permite girar al robot un número de grados determinados a partir de su orientación actual. Si el parámetro es negativo, el robot girará hacia la derecha (horario), y si es positivo girará hacia la izquierda (anti horario). La variable global r_step_theta tiene un valor por defecto de 7.5 grados.

`robot.move_fwd()`
Permite mover al robot hacia adelante (en la orientación a la cual se encuentra actualmente). El robot se desplazará 5 unidades en la pantalla.

Implementación de la solución

Al iniciar el reto se proveerá a su equipo de un código que carga la simulación y un ambiente de prueba. Su misión es implementar la función start() que se encarga de mover el robot para que logre llegar a la meta. El programa se detendrá automáticamente cuando el robot alcance la meta y llamará a la función report(). En la función report, su equipo

puede almacenar todos los datos que crea necesarios para ser utilizados en la exposición de su respuesta. Además, puede realizar optimizaciones de la ruta generada.

```
def start(robot): # <<<<==================== YOUR CODE  HERE ======================


def report(robot): # <<<<==================== YOUR CODE  HERE ======================
```

En la exposición de la solución se requiere que se describa el algoritmo que utilizaron para lograr el objetivo y que muestren una gráfica del camino que el robot debe seguir para llegar a la meta en base a la ejecución de su programa. La meta se encuentra en las coordenadas (1150,20).

Código del Proyecto

El código para ejecutar el simulador del robot lo puede encontrar en: http://bit.ly/2uL6d9d. Para ejecutarlo, previamente debe tener instalado Pygame. Si no lo tiene instalado en su máquina, por favor instalarlo utilizando el siguiente comando: **`pip install pygame.`**

Entregables

Subir en la dirección X un archivo rar con el formato ***NombreGrupo_Paralelo.rar*** lo siguiente:

1. Código utilizado
2. Una presentación visual para ser presentada en 5 min (Ejm: Power Point). Esta presentación, además de la descripción del algoritmo que presentó, debe incluir:
    a. Capturas de 2 escenarios en los que se utiliza su algoritmo (traces y tiempo).

En la presentación

Además de lo solicitado, realizar una demo del robot simulado utilizando su algoritmo.