



MASTERMIND

OBR 2024: MODALIDADE RESGATE

O ROBÔ:

Bem-vindos à apresentação do nosso robô, desenvolvido para competir na OBR (Olimpíada Brasileira de Robótica). A competição exige que o robô siga linhas, passe por encruzilhadas, detecte valores, desvie de objetos, e o principal, concluir o resgate.

Estrutura do Robô

Base Motriz

A base motriz é a estrutura que permite que o robô se mova. Ela é composta por dois motores localizados em lados opostos.

Módulo de Sensores

O módulo de sensores é responsável por coletar informações sobre o ambiente. Ele é composto por três sensores de cor/luminosidade e um sensor ultrassônico.

Base Motriz

1 Controladores

Os controladores fornecem a potência necessária para mover o robô. Eles são motores do kit EV3, que são de uso prático e fácil.

2 Esteiras

As esteiras, localizadas nas laterais do robô, junto aos motores, permitem que o robô se mova sobre o terreno. As esteiras são basicamente rodas agrupadas, que se movem com a rotação do motor



Módulo de Sensores

Sensores de Distância

Os sensores de distância permitem que o robô detecte objetos ao seu redor. O sensor que utilizamos é o ultrassônico, no qual mede distância com a vibração sonora.

Sensores de Luz

Os sensores de luz que utilizamos permitem que o robô reflita a luz do ambiente. Nós escolhemos utilizar os sensores de cor/reflexão.



Programação do Robô

1

Linguagem Python

A linguagem Python é a escolha perfeita para programação de robôs devido à sua simplicidade, legibilidade e vasto conjunto de bibliotecas.

2

Controle do Robô

O código Python controla a base motriz, o módulo de sensores e os atuadores do robô, permitindo que ele execute tarefas específicas.

```
def resgate():
    robot.pointTo(PontoInicial[2])
    robot.motors.move_tank(3000,250,250)
    robot.back_goTo(45,35)
    robot.back_goTo(Center[0],Center[1])
    safe = FindSafe(AreaResgate)
    if not safe:
        robot.goTo(out[0], out[1])
        robot.pointTo(out[2])
    else:
        robot.back_goTo(safe[0], safe[1])
        robot.goTo(Center[0],Center[1])
        robot.goTo(45,75)
        robot.goTo(out[0], out[1])
        robot.pointTo(out[2])
        robot.motors.move_tank(1000,250,250)
    print(robot.map.points)
```

```
#creating a function to detect if the robot is in t
def checarResgate(u_value):
```

```
    r = False
    if u_value > 300 and u_value < 500:
        motors.move_tank(500,-250,250)
        if u2.distance() < 500 and u2.distance() >
            motors.move_tank(700,250,-250)
            r = True
    else:
        motors.move_tank(1000,250,-250)
        if u2.distance()
            r = True
```

Linguagem Python



Classe "IntersectionSolver"

A interseção, uma classe que utiliza do módulo de sensores, mais especificamente os sensores de luz, basicamente nos diz se estamos em uma encruzilhada, e retorna um valor. Se o valor for falso, ele apenas segue o caminho. Se for verdadeiro, ele vê se o valor é verde, e então vira para seguir outro caminho.



Classe "MotorPair"

MotorPair é uma classe que agrupa dois motores de movimento, para facilitar na hora de programar. Ela também define comandos de movimentação e rotação, que possibilitam o robô de se mover com velocidade e tempos alternados



Classe "FinishLine"

Essa classe é a única que não é sempre executada no robô, pois funciona apenas no final do percurso. Ela vê se a cor é vermelho, ou seja, diferente das outras, e retorna um valor. Se o valor for verdadeiro, o robô para, e o percurso termina.



Versionamento de Código com Git

1

Controle de Versões

O Git permite que acompanhemos as mudanças no código, garantindo que o desenvolvimento do robô seja organizado e eficiente.

2

Histórico de Alterações

O Git registra todas as alterações feitas no código, permitindo que revivamos versões anteriores se necessário.

3

Trabalho em Equipe

O Git facilita o trabalho colaborativo, permitindo que múltiplos desenvolvedores trabalhem em conjunto no código do robô.





OBR

Demonstração do Funcionamento do Robô

Movimentos

ele se movimenta de acordo com sua correção de eixo, para seguir na linha, e seu alinhamento proporcional, permitindo com que avance quando estiver totalmente alinhado

Detecção de Cor

para poder se mover e detectar a linha, é preciso que ele detecte ela primeiro. Aí entram os sensores, que detectam a cor da linha para seguir, e a cor das interseções.

Detecção de Distância

Ela serve para duas coisas: A primeira é a detecção de obstáculos á frente, para conseguir desviar deles. A segunda é detectar o resgate, missão principal do robô.



Conclusão e Próximos Passos

Esta apresentação demonstrou a estrutura, a programação e o funcionamento do nosso robô. No futuro, pretendemos melhorá-lo, para conseguirmos atuar de maneira mais eficiente, e com isso obter resultados melhores.