

## Relatório do Laboratório 1 - Máquina de Estados Finita e *Behavior Tree*

### 1 Breve Explicação em Alto Nível da Implementação

#### 1.1 Máquina de Estados Finita

##### 1.1.1 Mover para frente

Usa-se atributos de tempo, *self.t*<sub>2</sub>, velocidade angular nula e velocidade linear padrão.

##### 1.1.2 Mover em espiral

Análogo ao anterior, utiliza-se o *self.t*<sub>1</sub>. Segue os passos do executor (cálculo da velocidade):

---

```
self.radius = constants.INITIAL_RADIUS_SPIRAL + self.t*self.b
self.angularSpeed = self.linearSpeed/self.radius
```

##### 1.1.3 Ir para trás

Utiliza-se a velocidade de ir para trás padrão e o tempo *self.t*<sub>3</sub> para a transição.

##### 1.1.4 Rotacionar

O acompanhamento do ângulo é análogo à evolução do tempo. Segue seleção do ângulo aleatório:

---

```
self.randomAngle = random.uniform(-math.pi, math.pi)
```

#### 1.2 *Behavior Tree*

Lógica análoga à da máquina de estados, construiu-se a árvore e, para as folhas:

---

```
def __init__(self):
    # Inicia as variaveis]
def enter(self, agent):
    # Reinicia o estado ser seguido
def execute(self, agent):
    # Tarefas a serem executadas
    ...
    # Checagem
    # Retorno de RUNNING, SUCCESS
    # Retorno do FAILURE (bumper indica parede)
```

## 2 Figuras Comprovando Funcionamento do Código

### 2.1 Máquina de Estados Finita

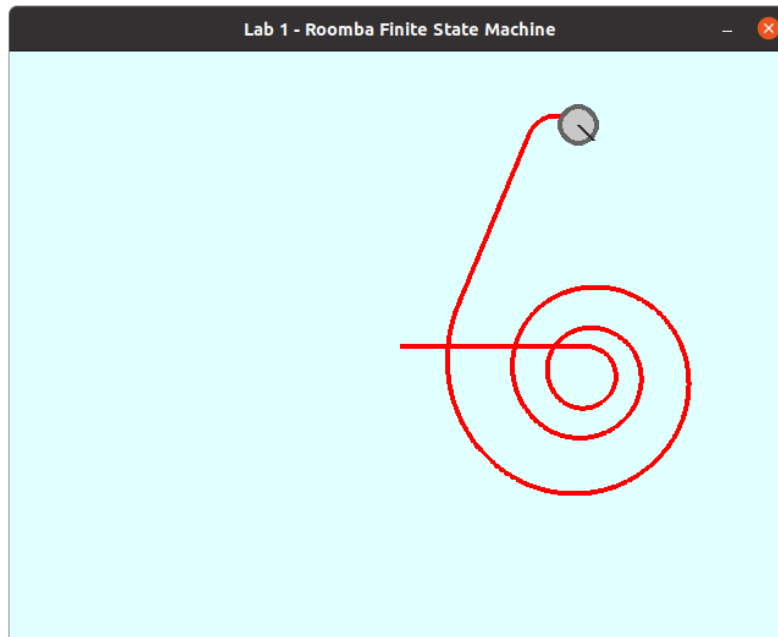


Figura 1: Máquina de estados, início do trajeto.

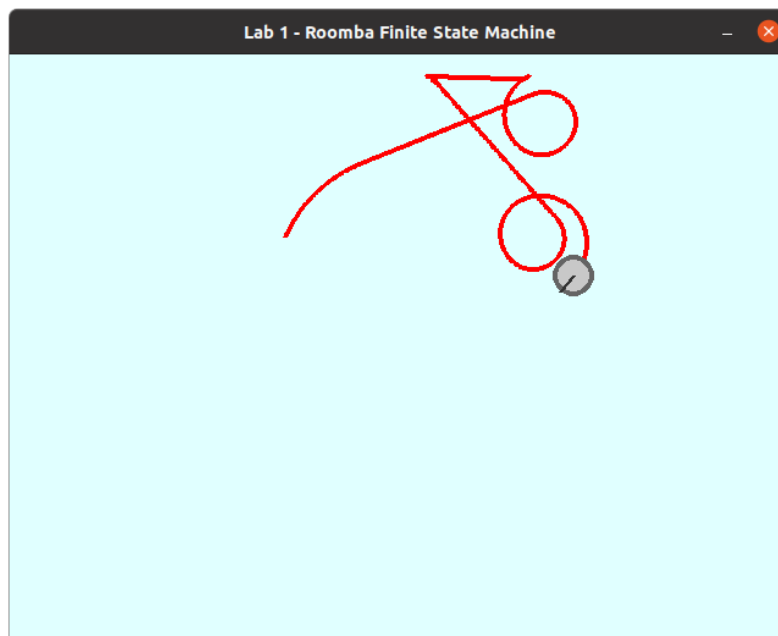


Figura 2: Máquina de estados, após um tempo longo e algumas colisões com as paredes.

## 2.2 *Behavior Tree*

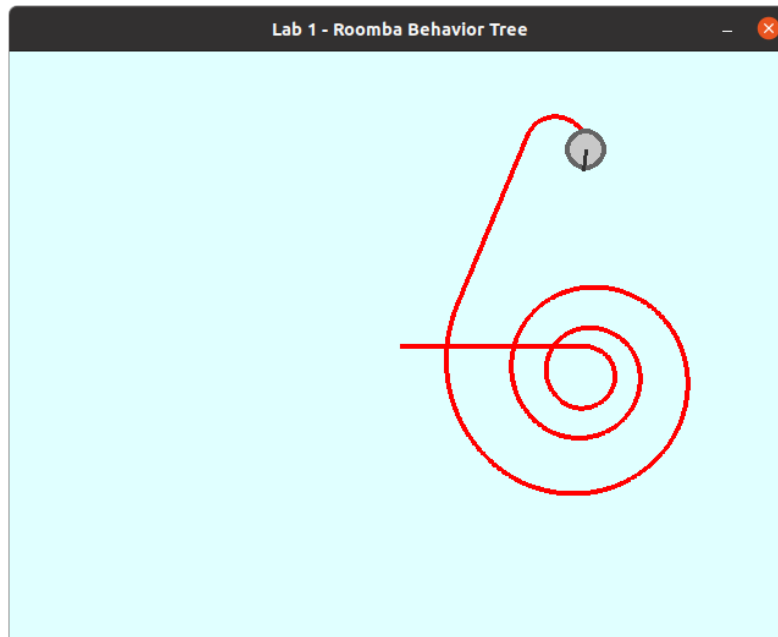


Figura 3: Árvore de comportamentos, início do trajeto.

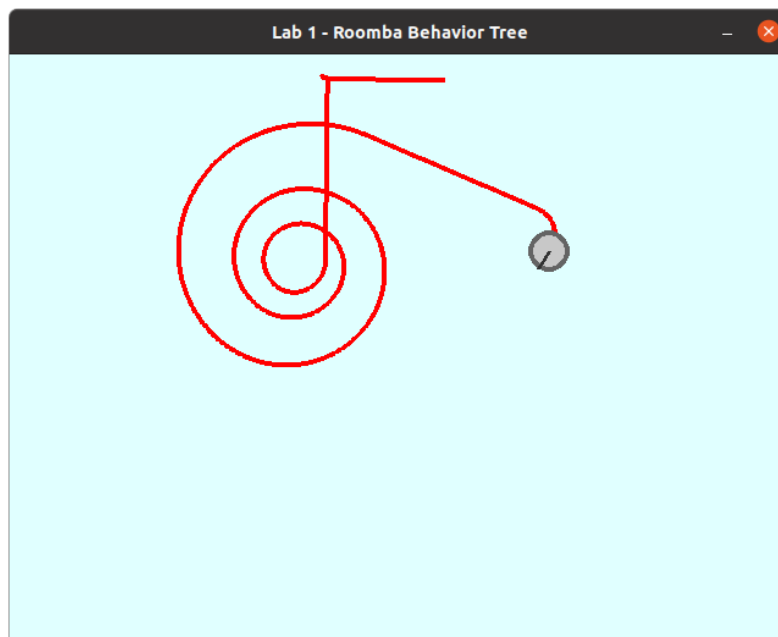


Figura 4: Árvore de comportamentos, após um tempo longo e algumas colisões com as paredes.