Instituto Tecnológico de Aeronáutica - ITA
Inteligência Artificial para Robótica Móvel - CT213
Aluno: Pedro Pinheiro Borges

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

A máquina de Estados Finitas funciona como um objeto que tem como atributo a classe `state`. Tal classe deriva em quatro classes:

- *Move Forward State*
- *Move in Spiral State*
- *Go Back State*
- *Rotate State*

Para cada `state` exite um tempo definido de duração antes de mudar para outro (caso o Roomba não colida com uma parede), com exceção do *Rotate State* cujo tempo de execução é aleatório.

Em cada um dos `state`, existem os mesmos métodos:

- *check transition*
- *execute*

O método *check transition* é usado para verificar se deve-se mudar ou manter o `state` atual, baseado no tempo decorrido (calculado utilizando a quantidade de vezes que o método execute foi chamado) e se o Roomba colidiu com uma parede. Já o *execute* é usado para alterar a velocidade do Roomba, de tal forma que ele consiga executar o movimento requisitado pelo `state` em questão.

1.2 *Behavior Tree*

A *Behavior Tree* é uma árvore binária que tem o nó raiz um *Selector Node*, o qual retorna falha se todos os filhos retornarem falha (funcionamento semelhante a uma porta lógica *OR*). Os filhos do nó raiz são dois *Sequence Node*, os quais retornam falha se pelo menos um filho retornar falha (funcionamento semelhante a uma porta lógica *AND*).

Os filhos do primeiro *Sequence Node* são:

- *Move Forward Node*
- *Move in Spiral Node*

E os filhos do segundo *Sequence Node* são:

- *Go Back Node*
- *Rotate Node*

Uma melhor visualização da *Behavior Tree* pode ser vista na imagem 1:

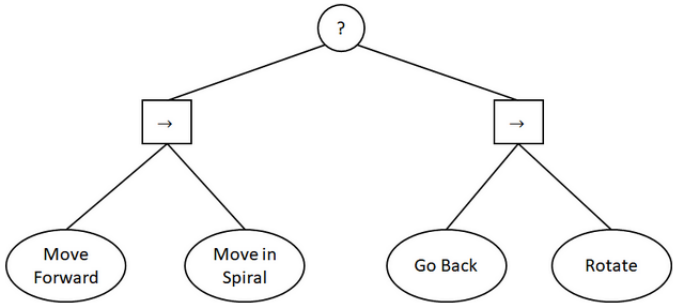

Figura 1: *Behavior Tree* do Roomba

Em cada nó, existem os seguintes métodos:

- *enter*
- *execute*

O método *enter* chamado sempre que se entra pela primeira vez em um nó, ele serve para iniciar os atributos necessários para o nó funcionar (como o tempo de execução). Já o *execute* é usado para alterar a velocidade do Roomba, de tal forma que ele consiga executar o movimento requisitado pelo `node` em questão e ele retorna se a tarefa realizada pelo nó foi um sucesso ou uma falha, baseado no fato de o Roomba ter colidido ou não com uma parede.

2 Figuras Comprovando Funcionamento do Código

2.1 Máquina de Estados Finita

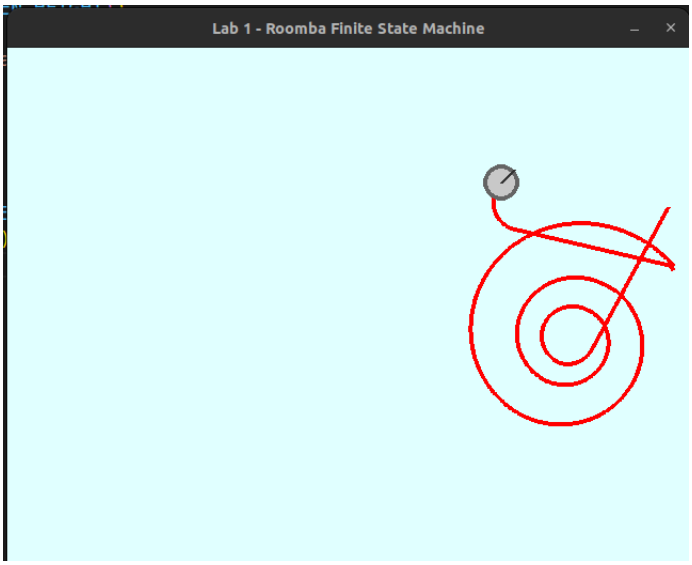

Figura 2: Trajetória do Roomba com *behavior* de FSM

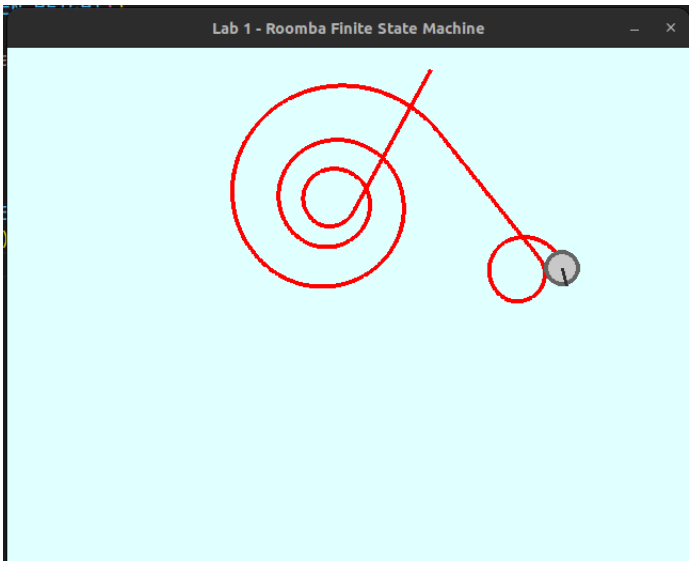

Figura 3: Trajetória do Roomba com *behavior* de FSM

2.2 *Behavior Tree*

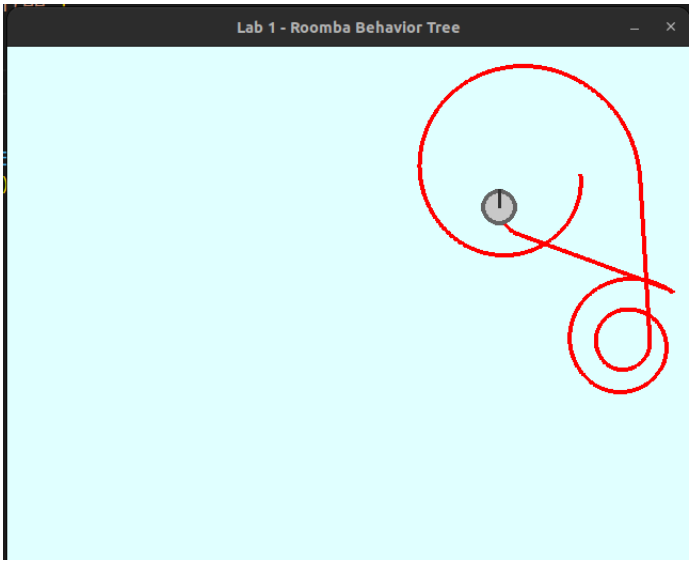

Figura 4: Trajetória do Roomba com *behavior* de BT

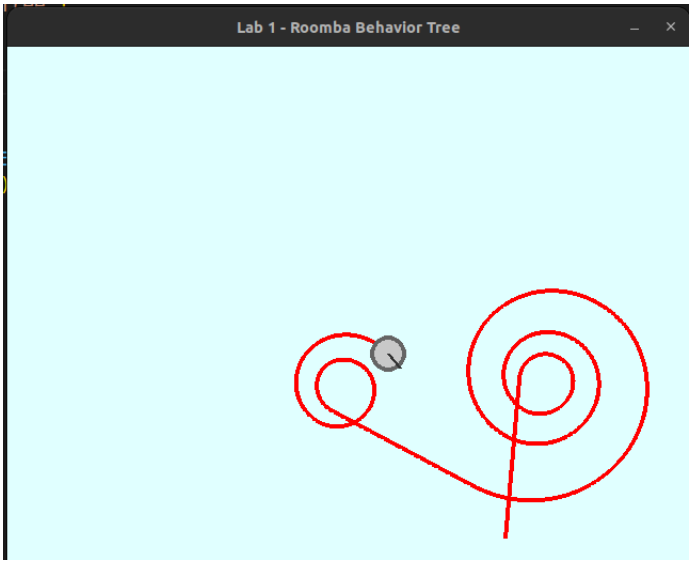

Figura 5: Trajetória do Roomba com *behavior* de BT