

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

Para a medição do tempo de simulação em cada um dos estados, criou-se uma variável `execution_count`, inicializada como nula e aumentada de 1 a cada vez que a função `execute` fosse executada, podendo-se obter o tempo de simulação através da operação `execution_count * SAMPLE_TIME`. Para que o robô pudesse girar em espiral, criou-se uma variável `spiral_radius`, que armazena o tamanho do raio da espiral a cada execução e sendo inicializada como `INITIAL_RADIUS_SPIRAL`. A cada execução, tal variável era incrementada de `SPIRAL_FACTOR * SAMPLE_TIME`. A velocidade angular ω , que é um dos argumentos da função `set_velocity` da classe `Roomba`, é calculada a cada execução por $\omega = \frac{v}{r(t)}$, em que v é a velocidade linear e $r(t)$ é o raio da espiral naquele instante.

A transição para o `GoBackState` ocorria quando nos estados `MoveForwardState` ou `MoveInSpiralState` a função de verificação `get_bumper_state` retornasse `True`. Para a determinação do ângulo de rotação do robô (`rotate_angle`) assim que movimentasse para trás, utilizou-se a função `uniform` do módulo `random` para retornar um `float` no intervalo $(-\pi, \pi)$. O tempo de execução do movimento é obtido por meio da operação `rotate_angle / ANGULAR_SPEED`.

1.2 *Behavior Tree*

Todas as operações implementadas na Máquina de Estados Finita foram basicamente mantidas no código da *Behavior Tree*, sendo também implementados procedimentos na lógica de funcionamento da *Behavior Tree*. A classe `RoombaBehaviorTree` foi inicializada instanciando os nós da árvore conforme o esquema da Figura 1, sendo o nó `root` um nó composto `Selector`, cujos filhos são dois nós `Sequence` referentes ao “modo limpeza” e ao “modo colisão” do Roomba e havendo, por fim, os nós folhas referentes aos 4 movimentos do robô.

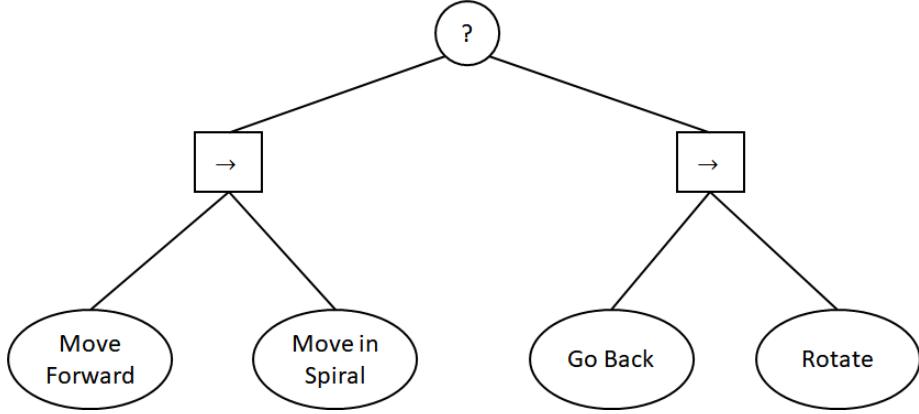


Figura 1: *Behavior Tree* do comportamento do Roomba

Os nós do modo limpeza retornavam *Failure* quando o robô colidia com a parede e *Success* quando o tempo de simulação era maior que o tempo de execução do movimento; caso contrário, retornavam *Running*. Quando retornavam *Failure*, o próximo nó a ser executado era o *GoBackNode*, que retornava *Success* quando terminava de executar o movimento e, na sequência, executava-se o nó *RotateNode*, retornando *Success* da mesma forma. Quando os últimos dois nós retornavam *Success*, o segundo nó *Sequence* (referente ao modo colisão) retornava *Success*, fazendo o nó *Selector* retornar *Success* e, portanto, recomeçando a árvore devido ao *update*, executando a função `enter` do primeiro filho.

2 Figuras Comprovando Funcionamento do Código

2.1 Máquina de Estados Finita

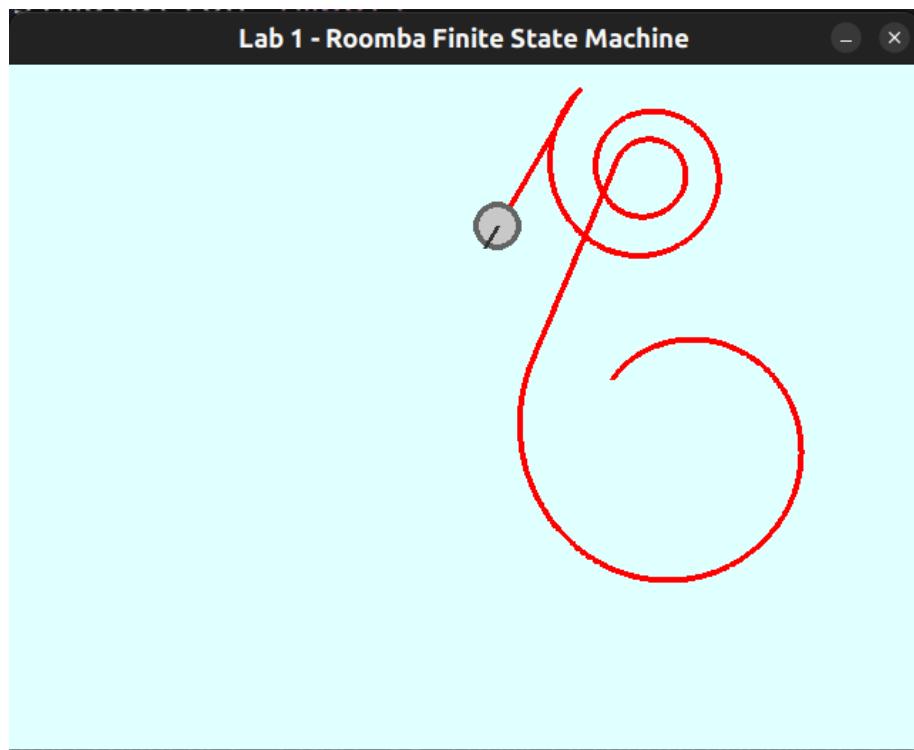


Figura 2: Simulação usando Máquina de Estados Finita

2.2 Behavior Tree

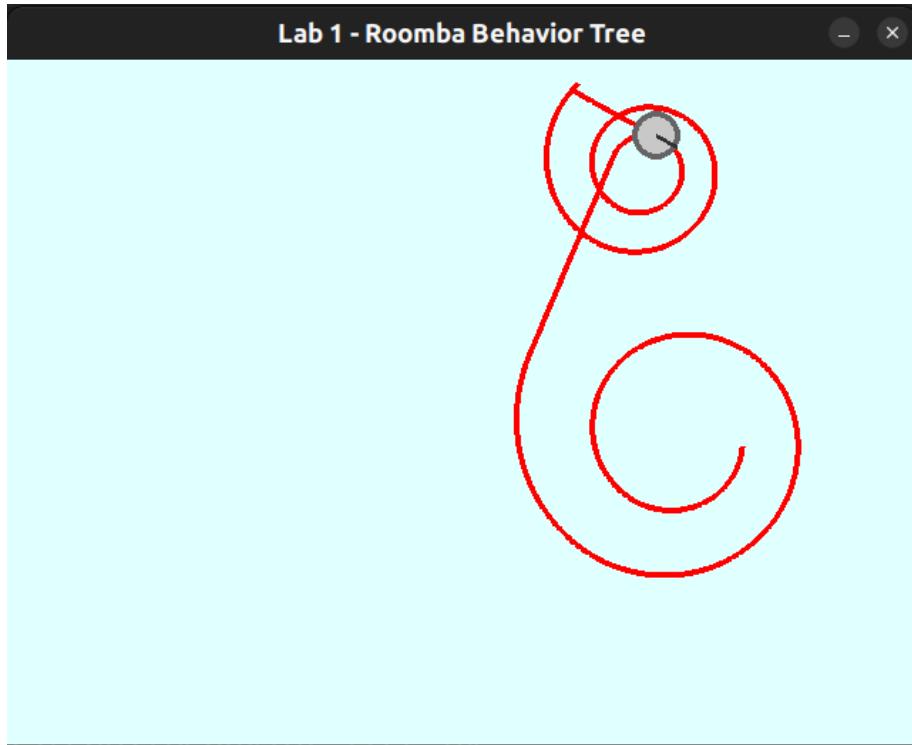


Figura 3: Simulação usando *Behavior Tree*