

Instituto Tecnológico de Aeronáutica - ITA
Inteligência Artificial para Robótica Móvel - CT-213
Aluno:

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

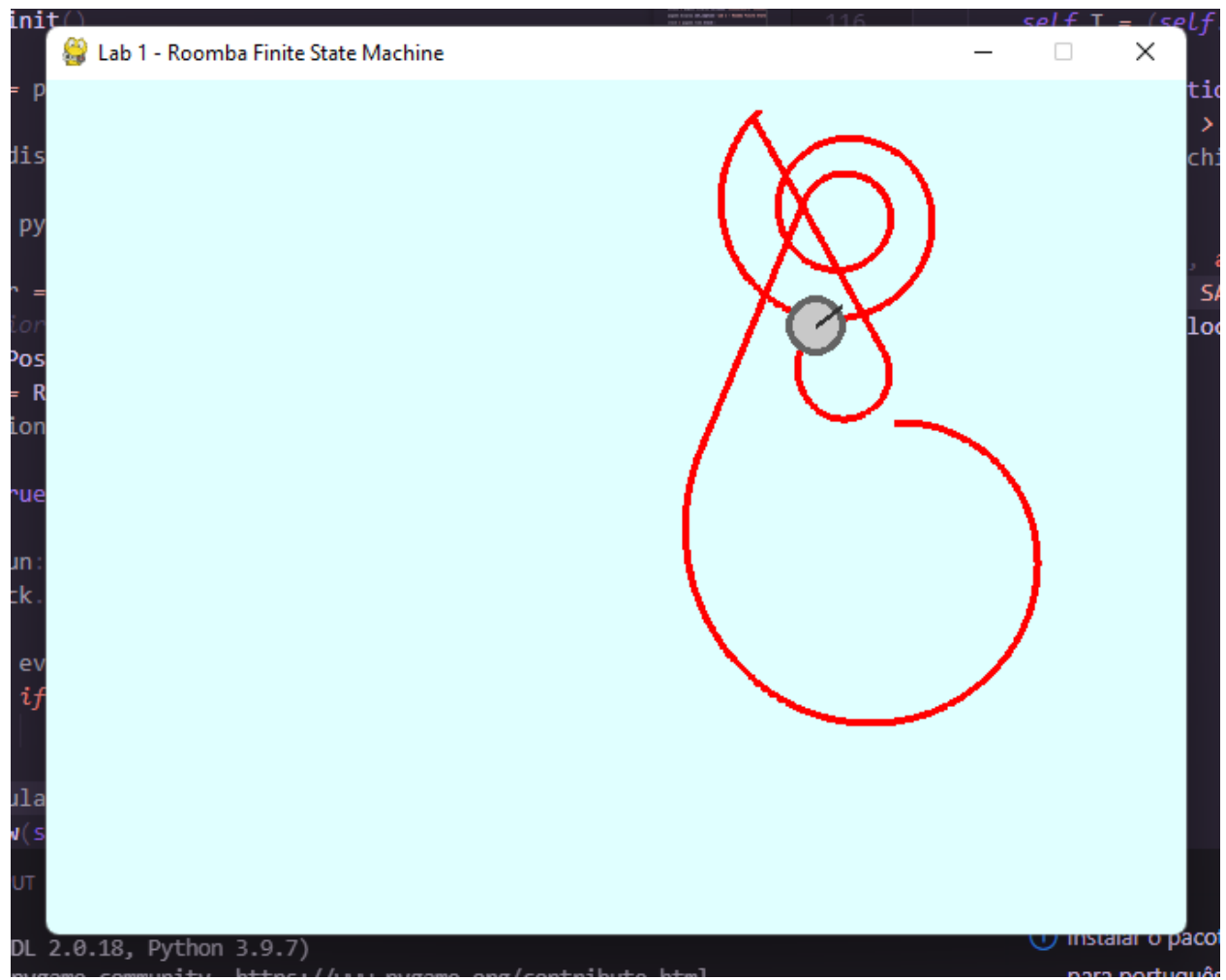
Para a implementação da máquina de estados finitas, bastou que, para cada estado, fossem implementadas uma `_init_`, em que inicializa-se variáveis iniciais como o tempo ou ângulo de giro, uma `check_transition`, que checa a condição de mudança de estado, e uma `execute`, que determina a tarefa que deve ser realizada enquanto não houver troca de estado. Dessa forma, bastou observar as condições de mudança e as tarefas a serem realizadas, descritas no fluxograma de máquina de estados finita do roteiro (Figura 2). Dessa forma, de tempos em tempos o robô alternava entre movimento retilíneo para frente e movimento espiral, mudando para o estado de voltar para trás após colidir com a parede. Isso é seguido pelo estado de rotação parado para determinar uma nova direção de giro, até que, rodado o ângulo de giro, o estado passa a alternar entre retilíneo e espiral.

1.2. Behavior Tree

Primeiramente foi necessário construir a `RoombaBehaviorTree`, que teve que ser composta de um `SelectorNode` com dois filhos adicionados pela função `add_child()` que eram `SequenceNode`, como indicado no roteiro (Figura 3). No nó da esquerda tínhamos o `MoveForwardMode` seguido pelo `MoveInSpiralNode`. Para o `SelectorNode` reiniciar, é necessário que apenas um dos nós retorne `SUCCESS`. Por isso, para alterar entre movimento espiral e movimento retilíneo enquanto não havia colisão, retornava-se `SUCCESS` sempre que a duração tempo dos movimentos era excedida. Já quando havia colisão, retornava-se `FAILURE`, para que o `SelectorNode` passe para o nó da direita, que contém os nós `GoBackNode` e o `RotateNode`. Como queríamos que o processo se repetisse, retornamos `SUCCESS` sempre assim que as funções desses nós eram executadas. Enquanto os nós não retornava nem `SUCCESS` nem `FAILURE`, retornavam `RUNNING`. Foi necessário, além disso, sempre inicializar o tempo no `enter`, para que as funções exercidas pelos nós pudessem se repetir. Na `_init_` foram inicializadas variáveis como o tempo ou o ângulo de rotação e na `execute` foram atualizadas variáveis como tempo e velocidade e foram escritas as condições para o retorno dos nós.

2. Figuras Comprovando Funcionamento do Código

2.1. Máquina de Estados Finita



2.2. Behavior Tree

