UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

GABRIEL ODIA DA ROZA
LUCAS SANTOS WEISS
VICTOR DUARTE HENNEMANN

Edubot

Professor: Renato Ventura Bayan Henriques

Porto Alegre
2025

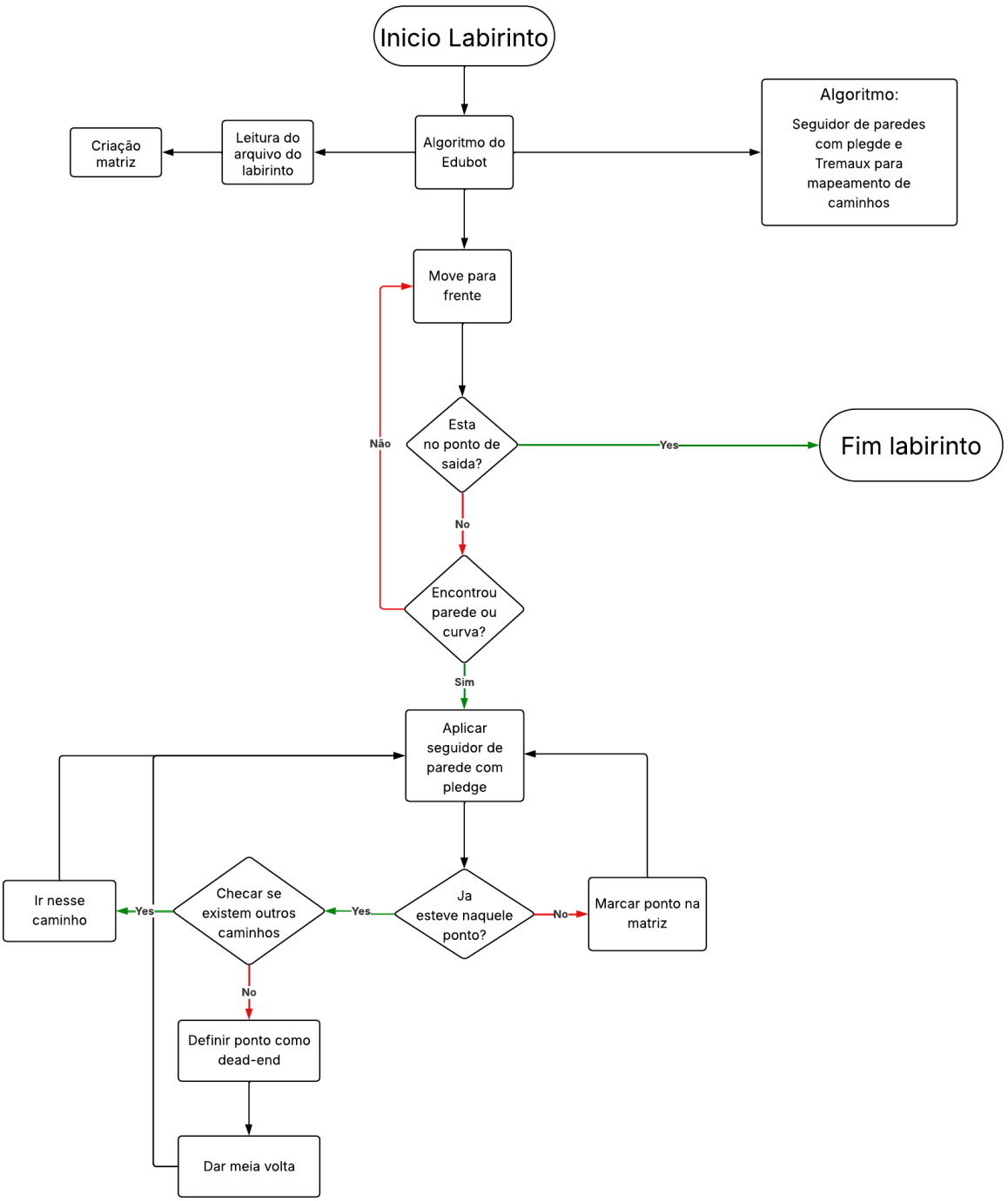
DESCRIÇÃO DO TRABALHO

A primeira etapa do trabalho foi pensar em como será feito o algoritmo do robô para a saída. As opções consideradas foram: Seguidor de paredes, Pledge, A*, Distância de Manhattan, Tremaux. A escolha foi primeiro fazer um seguidor de paredes com Pledge para resolução do labirinto e depois a implementação de Trémaux para aumentar a eficiência do algoritmo(IMPLEMENTAÇÃO TREMAUX FALHA). A* foi considerada no processo por ser a mais eficiente, sendo utilizada no GPS, mas por apresentar um alto nível de dificuldade foi descartada.
Alguns desafios enfrentados foram:

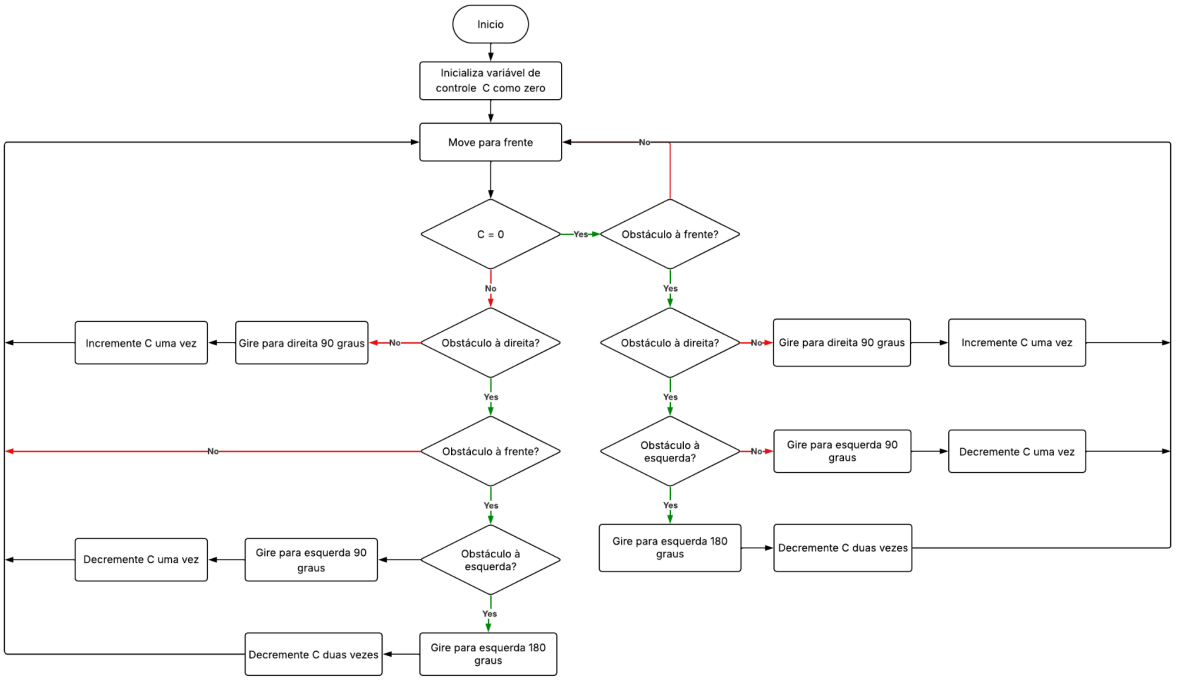
- Leitura do labirinto e representação em uma matriz
- Imprecisão do movimento do Edubot em algumas situações

O principal desafio enfrentado foi quando verificar os sonares, pois ou eles são checados o tempo inteiro ou são checados quando o bot se encontra no meio de uma célula, mas como o edubot não tem uma precisão exata, ele acaba batendo na parede em algum momento. No mapa maze isso não se tornou um problema pois as paredes são bem distantes, mas no pacman é mais complicado pois as paredes são bem estreitas.
O código acabou sendo muito eficiente nos dois mapas, mas é nítido que isso aconteceu por um pouco de sorte, pois se fosse o mapa fosse um pouco diferente, o robô facilmente cairia em um loop. Como melhoria, só falta implementar tremaux para marcar células já visitadas e evitar loops

FLUXOGRAMA

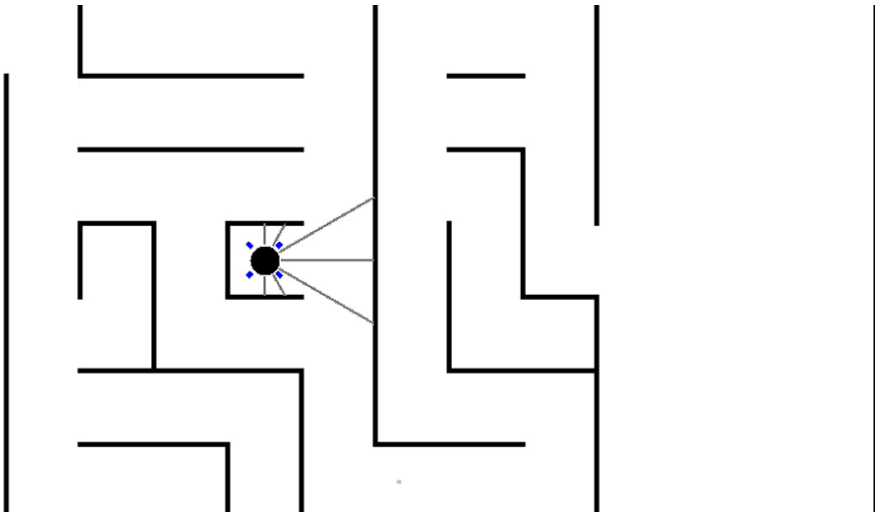

Inicio Labirinto
Algoritmo:
Seguidor de paredes com plegde e Tremaux para mapeamento de caminhos
Algoritmo do Edubot
Leitura do arquivo do labirinto
Criação matriz
Move para frente
Esta no ponto de saida?
Yes
Fim labirinto
Não
No
Encontrou parede ou curva?
Sim
Aplicar seguidor de parede com pledge
Ja esteve naquele ponto?
No
Marcar ponto na matriz
Yes
Checar se existem outros caminhos
Yes
Ir nesse caminho
No
Definir ponto como dead-end
Dar meia volta

Explicação de seguidor de paredes para direita com Pledge para fluxograma

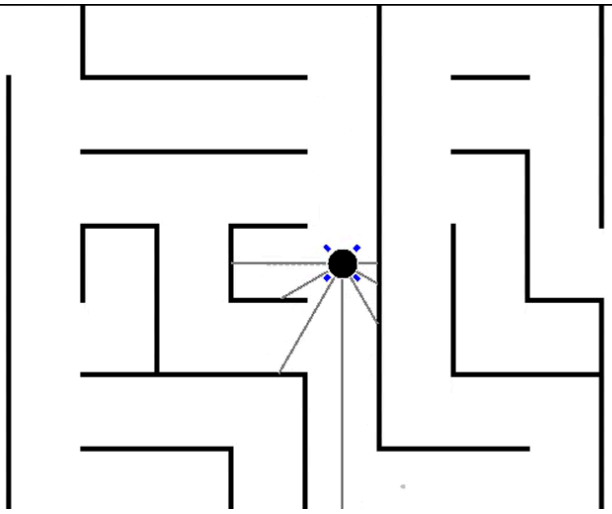


Algoritmos de Navegação				
Rato Aleatório:	**Seguidor de Parede**:	**Pledge**:	**Tremaux**:	**A*:**
Sem memória, caminha aleatoriamente até encontrar obstáculo e escolhe nova direção (sempre evitando retornar 180° a menos que esteja em dead-end) . **CONTRAS** *Sem heurística *Movimentos aleatórios, sem lógica	Mantém contato constante com a parede de um lado (direito ou esquerdo), girando 90° toda vez que a parede some ou encontra obstrução à frente . **CONTRAS** *Sem heurística *Paredes soltas prendem o robô	Aprimora o seguidor de parede ao contar rotações e abandonar a parede apenas quando o "contador de curvas" (diferença entre curvas esquerda e direita) zera, permitindo ignorar paredes soltas no meio do labirinto . **CONTRAS** ***Não garante a solucao do labirinto caso comece do meio dele**	Mapeamento incremental: deixa marcações virtuais (migalhas) nas células; nunca repete caminho já marcado "duas vezes" e, em cruzamentos, escolhe caminhos sem marcações; em celas esgotadas, trata como obstáculo . **CONTRAS** ***Depende de um algoritmo de navegação**	Algoritmo de busca de caminho que encontra o caminho mais curto entre dois pontos em um grafo ou mapa. Utilizando da distância de Manhattan **CONTRAS** *Difícil implementação

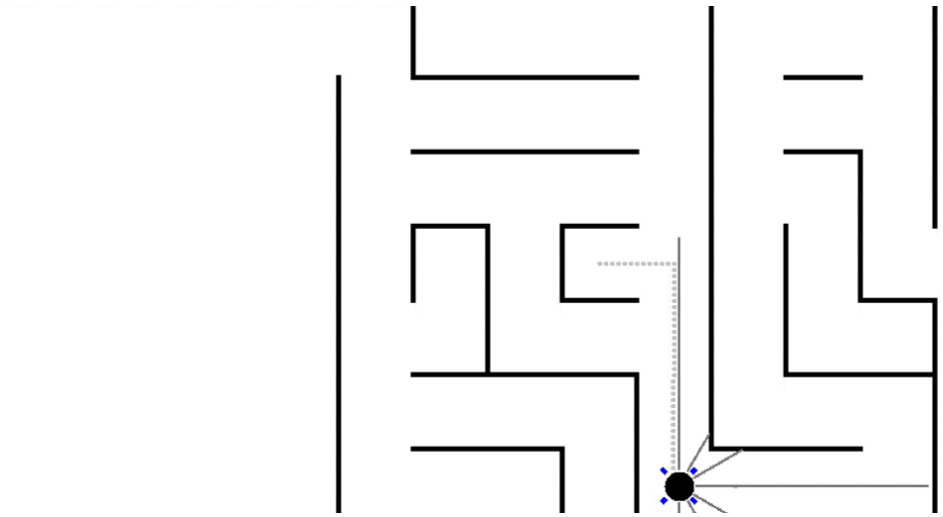
SCREENSHOTS NAVEGAÇÃO MAZE



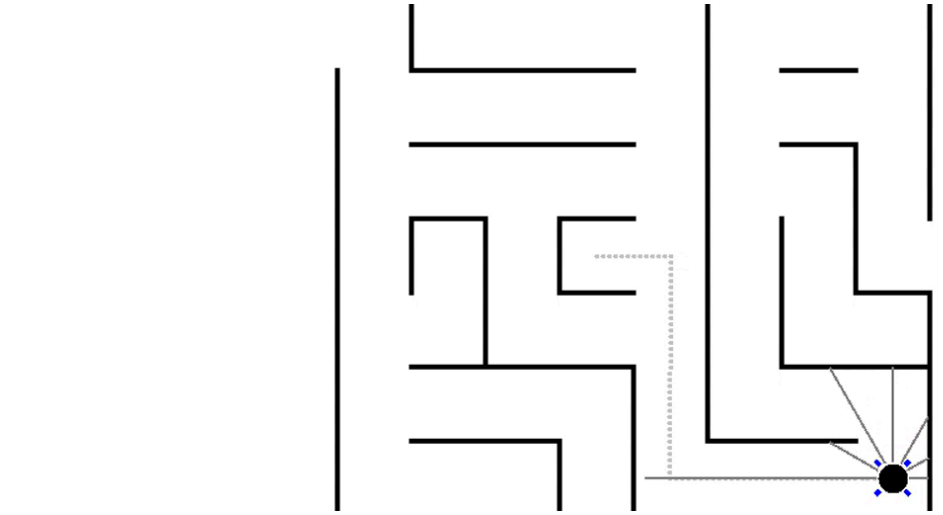
Contador:0
Descrição:Robô irá seguir reto até encontrar uma parede.



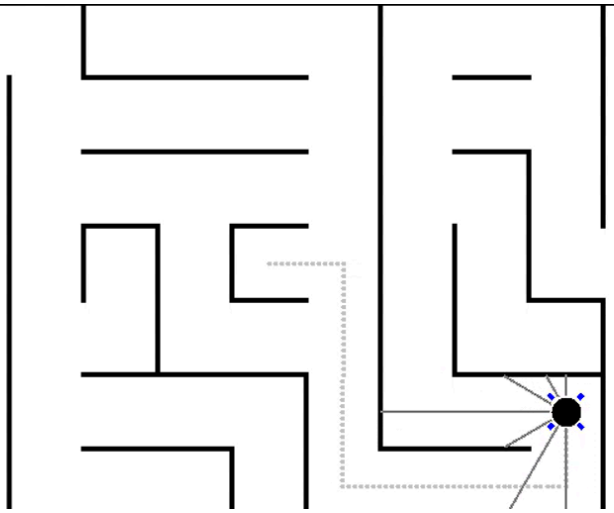
Contador:1
Descrição: Encontrou parede, fez curva para direita e Contador+1
Robô irá seguir para frente até encontrar uma parede ou ter a oportunidade de zerar o contador(fazer curva para esquerda)



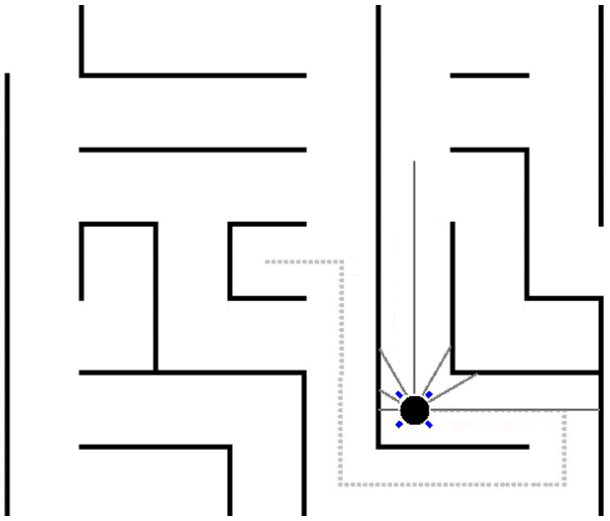
Contador:0
Descrição:Robô teve oportunidade de zerar o contador e fez curva para esquerda Contador-1, Robô irá seguir para frente até encontrar parede



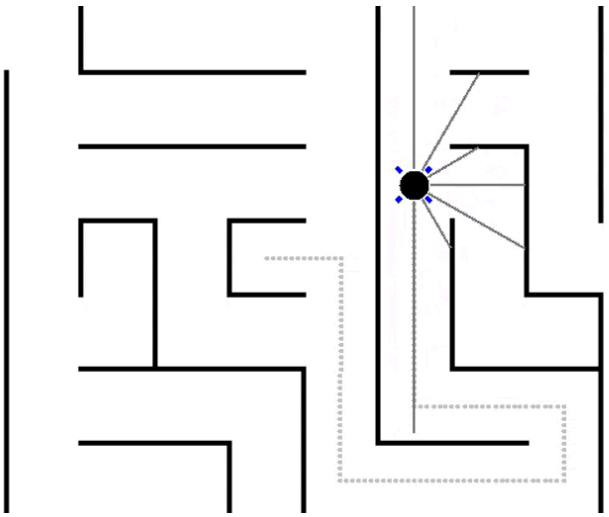
Contador:-1
Descrição:Robô encontrou parede, curvaria para direita, mas como estava bloqueada, curvou para esquerda Contador-1
Robô irá seguir para frente até encontrar uma parede ou ter a oportunidade de zerar o contador(fazer curva para direita)



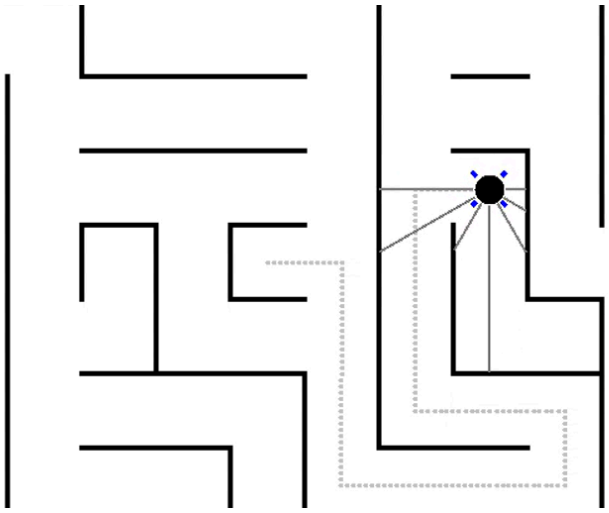
Contador:-2
Descrição: Robô não pode seguir em frente nem virar direita, então virou esquerda Contador-1
Robô irá seguir para frente até encontrar uma oportunidade de virar a direita ou a esquerda(direita como preferência)



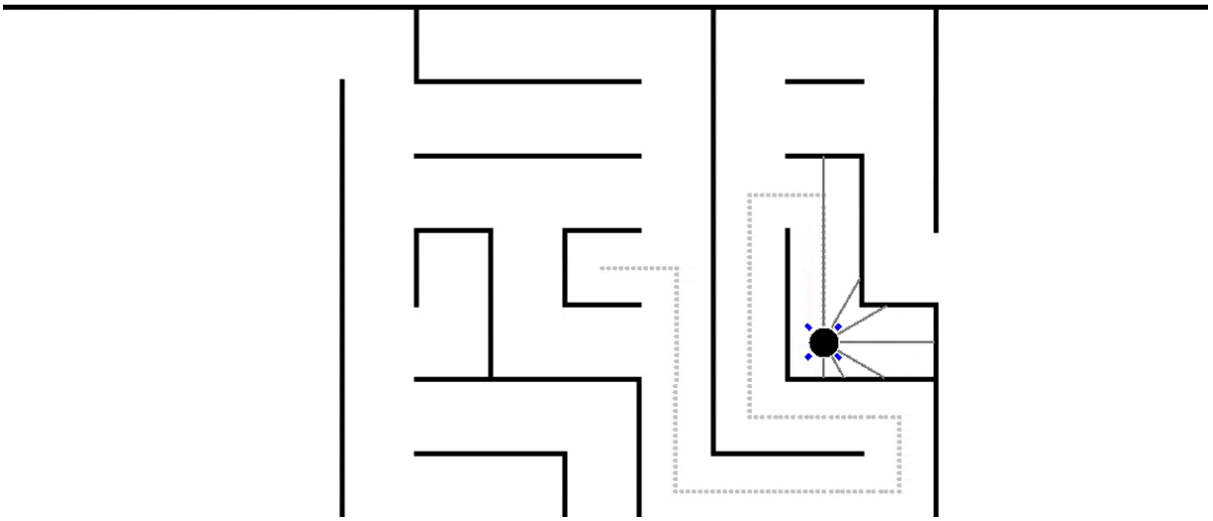
Contador:-1
Descrição:Robô teve oportunidade de virar direita Contador+1
Robô irá seguir para frente até encontrar uma parede ou ter a oportunidade de zerar o contador(fazer curva para direita)



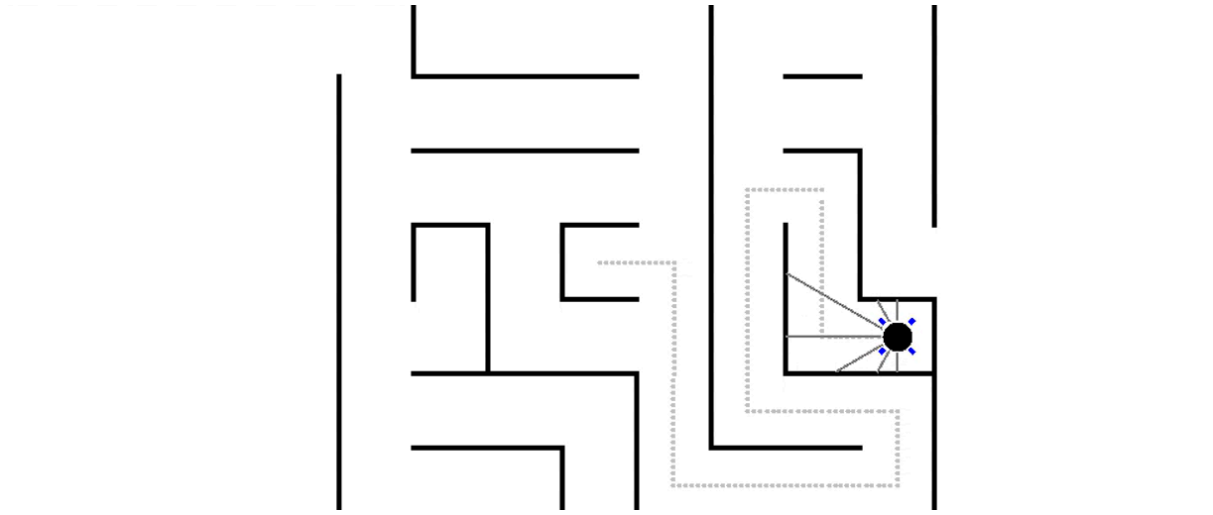
Contador:0
Descrição:Robô teve oportunidade de virar direita Contador+1 e zerar o contador
Robô irá seguir reto até encontrar uma parede.



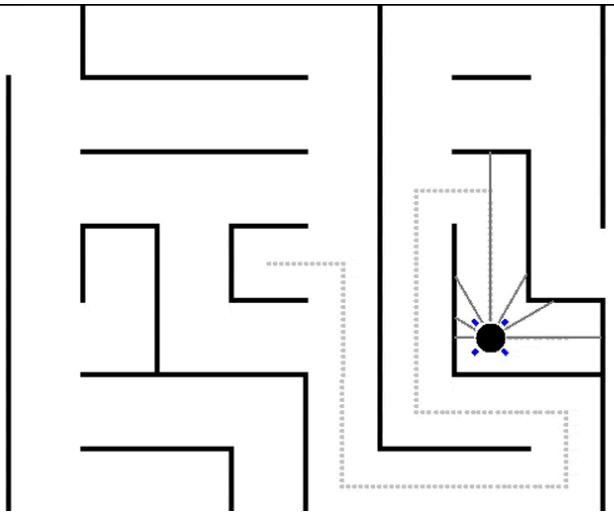
Contador:1
Descrição:Robô encontrou parede e virou direita Contador+1



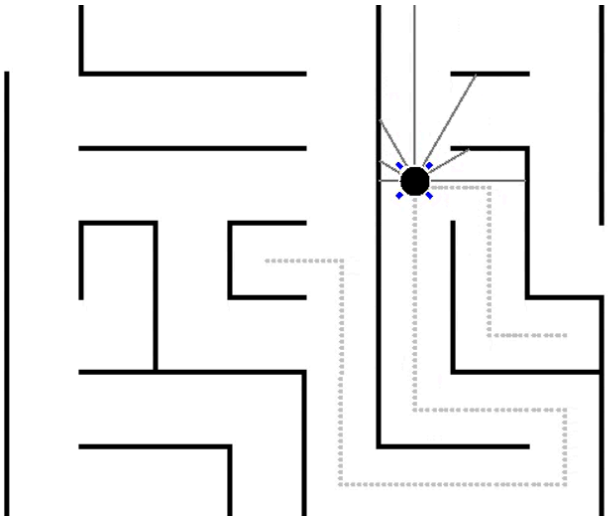
Contador:0
Descrição:Robô teve oportunidade de zerar o contador e virou esquerda
Contador-1



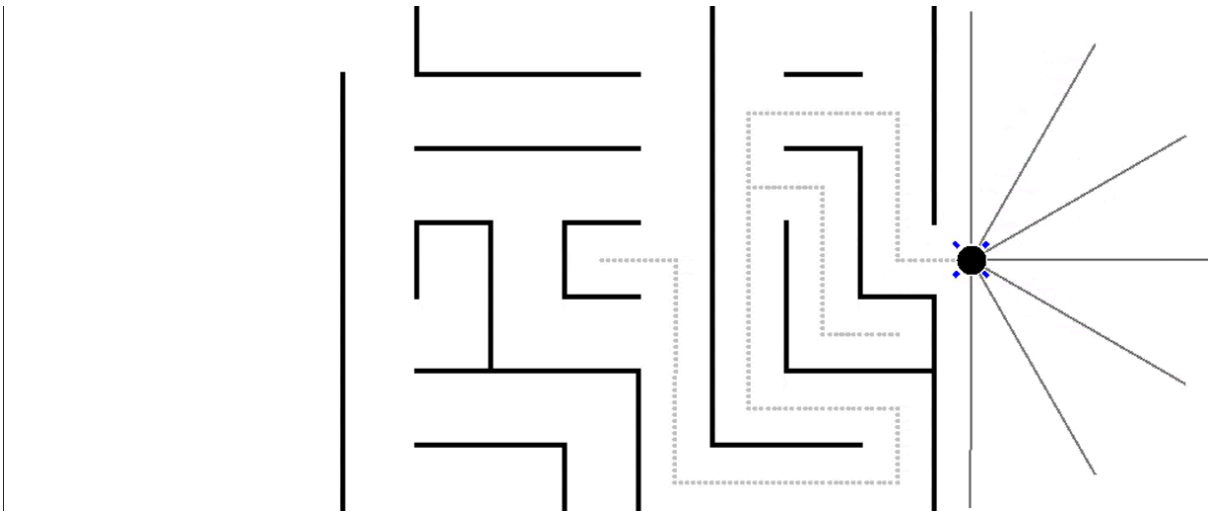
Contador:-2
Descrição:Robô encontrou dead-end e virou para trás Contador - 2



Contador:-1
Descrição:Robô virou direita Contador+1, caso ele virasse esquerda, o contador seria somado +3, para revitalizar o padrão de sul, contador=1

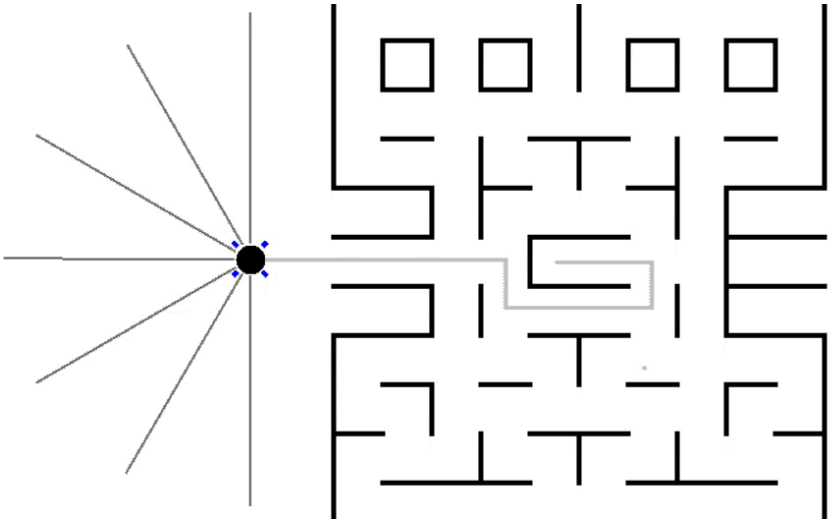


Contador:-1
Descrição:Robô virou direita ao invés de esquerda porque era sua preferência
Contador+1



Robô achou a saída seguindo os comportamentos descritos antes.

SCREENSHOTS NAVEGAÇÃO PACMAN



Mudança do código para mapa Pacman:Robô dá uma rotação de 180 no início para redefinir sentido de preferência(Se torna a esquerda a partir da rotação), além de algumas mudanças de velocidade, tempo de parada e distancia de sonares.
O Mapa foi terminado com a maior eficiência possível sem heurística, mas por apenas sorte das direção de preferência que foi imposta no robô.
Quero esclarecer também que houve várias tentativas pela imprecisão dos movimentos do robô.
Segue o vídeo no github para ver a navegação.