

Lab 02 - Principais abordagens em IA - abordagem simbólica

Ex 1: Converta projeto com if...else

Ex 2: Navegação IA Simbólica do Drone Entre Posições

Ex 3: Navegação IA Simbólica do Drone Entre Posições com obstáculo

Ex 1: Converta projeto com if...else

Converta o Código do ficheiro `ex02_1.py` em IA simbólica.

Ex 2: Navegação IA Simbólica do Drone Entre Posições

- **Problema:** Como controlar o movimento de um drone para que ele percorra uma série de pontos específicos numa sequência definida num espaço bidimensional?
- **Símbolos:**
 - (S1): O drone está na posição "Base".
 - (S2): O drone está na posição "A".
 - (S3): O drone está na posição "B".
 - (S4): O drone está na posição "C".
 - (S5): O drone está na posição "D".
 - (R1): O drone está em "Descolagem".
 - (R2): O drone está "Em Rota".
 - (D1): O destino atual do drone é o ponto "A".
 - (D2): O destino atual do drone é o ponto "B".
 - (D3): O destino atual do drone é o ponto "C".
 - (D4): O destino atual do drone é o ponto "D".
 - (D5): O destino atual do drone é a "Base".
 - (M1): O movimento para o destino foi completado.
- **Regras:**
 - Se (R1) (o drone está em "Descolagem") e (M1) (movimento completo), então o estado do drone muda para (R2) ("Em Rota").
 - Se (R2) e (S1) (o drone está na "Base") e (M1), então o destino seguinte é (D1) (ponto "A").
 - Se (R2) e (S2) (o drone está no ponto "A") e (M1), então o destino seguinte é (D2) (ponto "B").
 - Se (R2) e (S3) (o drone está no ponto "B") e (M1), então o destino seguinte é (D3) (ponto "C").
 - Se (R2) e (S4) (o drone está no ponto "C") e (M1), então o destino seguinte é (D4) (ponto "D").
 - Se (R2) e (S5) (o drone está no ponto "D") e (M1), então o destino seguinte é (D5) (a "Base").
- **Exemplo de Decisão:**
 - **Estado Inicial:** O drone está na "Base" ((S1)) e o estado é "Descolagem" ((R1)).
 - **Regra Ativada:** Como o drone está em "Descolagem" e o movimento foi completado ((M1)), o estado muda para "Em Rota" ((R2)) e o próximo destino é o ponto "A" ((D1)).
 - **Ação:** O drone começa a mover-se em direção ao ponto "A". Ao atingir (M1), o estado do drone confirma que o movimento foi completado, e o próximo destino é atualizado para o ponto "B" ((D2)).

Funcionamento Passo a Passo:

1. Inicialização:

- O sistema é inicializado com o drone na "Base" ((S1)) e o destino é configurado como o ponto "A" ((D1)). O estado inicial do drone é "Descolagem" ((R1)).

2. Movimento do Drone:

- O drone começa a mover-se da "Base" para o ponto "A". O sistema calcula a direção e move o drone em pequenos passos até que ele atinja o destino. Quando o movimento está completo ((M1)), o estado muda para "Em Rota" ((R2)) e o próximo destino é atualizado para "B".

3. Sequência de Navegação:

- Este ciclo repete-se: o drone move-se de "A" para "B", depois para "C", "D", e finalmente retorna à "Base". A cada etapa, o sistema atualiza o destino conforme as regras definidas.

4. Atualização do Estado do Obstáculo:

- Embora o obstáculo seja atualizado a cada iteração, no contexto deste exemplo específico, ele não interfere diretamente com o movimento do drone. O foco está na navegação do drone entre os pontos predefinidos.

5. Conclusão:

- O drone completa a sua rota, visitando todos os pontos definidos numa sequência lógica, controlado pelas regras simbólicas que determinam o próximo destino com base na posição atual e estado do movimento.

Auxílio: ver as regras no Documento de Regras:

Ex 3: Navegação IA Simbólica do Drone Entre Posições com obstáculo

Atualização do Código: Introdução de Obstáculos Dinâmicos

Neste exercício, vamos expandir o código existente para incluir a detecção e evasão de obstáculos dinâmicos durante a navegação do drone. A inteligência simbólica será responsável por garantir que o drone evita colisões com obstáculos que se movem verticalmente no espaço.

Objetivo:

Atualizar o sistema de navegação do drone para que ele seja capaz de detectar e evitar obstáculos durante o seu movimento entre pontos.

Descrição das Modificações:

1. Migre o código para dentro do simulador gráfico:

- Coloque as regras no código **Lab 02 - ex3.py**

2. Detecção de Colisão com Obstáculos:

- **Símbolos Adicionados:**
 - (O1): Posição vertical do obstáculo.
 - (C1): Verificação de colisão iminente.
 - (A1): Rota alternativa para evitar o obstáculo.
 - (E1): O drone está a evitar um obstáculo.
- **Regra de Detecção de Colisão:**
 - Se o drone está a mover-se e a sua trajetória passa pela posição atual do obstáculo ((O1)), então (C1) é ativado, indicando uma colisão iminente.

3. Evasão de Obstáculos:

- **Símbolos Adicionados:**
 - (E1): O drone está a seguir uma rota alternativa para evitar o obstáculo.
- **Regras para Evasão:**
 - Se (C1) é verdadeiro (colisão iminente) e (E1) não foi ainda ativado, então a IA calcula uma (A1) (rota alternativa).
 - O drone segue (A1) até que a colisão seja evitada, e depois retoma a rota original.

4. Movimento Continuado:

- **Regras:**
 - Após evitar o obstáculo, (E1) é desativado e o drone continua em direção ao seu destino original.

- Se o drone completa o movimento para o destino ((M1)), o próximo destino é definido conforme as regras pré-existentes.

Implementação no Código:

Aqui estão as principais modificações e adições ao código:

1. Função `condicao_verifica_colisao`:

- Verifica se o drone, no seu trajeto entre dois pontos, cruzará a área ocupada pelo obstáculo. Se sim, ativa a condição de colisão.

2. Função `regra_calcular_rota_alternativa`:

- Calcula uma rota que evita o obstáculo ao desviar lateralmente para a direita ou para a esquerda, dependendo de onde o drone está em relação ao obstáculo.

3. Modificações na Função `regra_mover_drone`:

- Implementa a lógica para o drone seguir a rota alternativa, caso uma colisão seja prevista, e retomar a rota original após evitar o obstáculo.

4. Atualização do Estado do Obstáculo:

- A posição vertical do obstáculo é atualizada continuamente, simulando o movimento do obstáculo no ambiente.

Exemplo de Decisão:

- **Situação Inicial:** O drone está a mover-se do ponto "A" para o ponto "B". O obstáculo está a mover-se verticalmente no meio do espaço.
- **Deteção de Colisão:** Durante o trajeto, a IA deteta que o drone irá colidir com o obstáculo se continuar no caminho atual.
- **Ação de Evasão:** A IA calcula uma rota alternativa que desvia o drone para a esquerda, contornando o obstáculo.
- **Retorno à Rota Original:** Após evitar o obstáculo, o drone retoma a sua rota original em direção ao ponto "B".

Auxílio: ver as regras no Documento de Regras: