

Relatorio das Atividades - Dividido por Etapa

Estudantes: Bernardo Nicoletti, Enrico B R Santos, Felipe Clzanosky, Guilherme Piva, Pedro K Albuquerque.

Etapa 1: Agente que descobre seus Limites

Na primeira etapa do projeto, o objetivo principal foi fazer o robô explorar os limites do ambiente e reconhecer as bordas do grid. A lógica consistiu em mover o agente em quatro direções fixas, direita, baixo, esquerda e cima, até que ele colidisse com uma parede. Durante esse percurso, cada posição visitada era exibida no console, permitindo acompanhar em tempo real o deslocamento. Ao encontrar uma parede, o robô registrava a última posição válida antes do impacto e prosseguia para a próxima direção. Esse comportamento simples, mas essencial, marcou o primeiro passo para que o agente começasse a interagir ativamente com o ambiente.

O ambiente foi definido como um grid quadrado de tamanho variável, fornecido pelo usuário no início da execução. Além disso, a posição inicial do robô também era escolhida pelo usuário, com verificações para garantir que estivesse dentro dos limites da matriz. Esse detalhe permitiu testar diferentes cenários de exploração, variando tanto o tamanho do ambiente quanto a localização inicial do agente. O uso de uma função auxiliar para desenhar a matriz ajudou a visualizar as células já percorridas e a posição final do robô após completar o ciclo de movimentos.

Os resultados mostraram que o agente conseguiu encontrar corretamente as quatro paredes em todos os experimentos, confirmando a efetividade da lógica implementada. Embora simples, essa etapa foi crucial para consolidar a noção de deslocamento e percepção de fronteiras dentro do grid. Ela também trouxe uma visão didática de como estruturar o fluxo de decisões do agente: mover-se, registrar estados percorridos, detectar condições de parada e seguir para a próxima ação. Em resumo, a Etapa 1 representou a base da exploração, preparando o caminho para as fases seguintes, nas quais seriam introduzidos obstáculos e objetivos mais complexos.

Etapa 2: Agente que analisa células disponíveis

Na segunda etapa, o agente evoluiu de um simples explorador de fronteiras para um explorador completo do ambiente, capaz de visitar todas as células acessíveis do grid, mesmo na presença de obstáculos. O algoritmo central foi a Busca em Largura (BFS), utilizada aqui para encontrar sempre a célula não visitada mais próxima e calcular o caminho até ela. Dessa forma, o robô passou a se mover de maneira sistemática, explorando o mapa passo a passo sem deixar regiões descobertas. A cada movimento, a posição do agente e as células já visitadas eram plotadas em tempo real, o que tornava a execução bastante visual e didática.

O ambiente foi configurado em um grid 10x10 com obstáculos fixos, representados como células bloqueadas que o robô precisava contornar. A posição inicial era sorteada aleatoriamente, garantindo variedade entre diferentes execuções. Durante a simulação, o agente repetia um ciclo de percepção e deslocamento: identificava a célula não visitada mais próxima por meio da BFS, seguia até ela e atualizava a lista de posições já exploradas. Esse processo se repetia até que todas as áreas acessíveis do grid tivessem sido visitadas, respeitando os bloqueios impostos pelos obstáculos.

Os resultados dessa etapa mostraram que o robô foi capaz de percorrer 100% das células acessíveis em todos os testes realizados, o que validou a efetividade da lógica implementada. Além disso, foram calculadas métricas importantes como número total de passos, passos redundantes e porcentagem de completude. Embora nem sempre o agente seguisse a rota mais eficiente em termos de distância, ele garantiu a exploração total do ambiente, demonstrando sucesso na tarefa de desvio de obstáculos. Essa fase representou um marco importante no projeto, pois estabeleceu as bases do conceito de exploração sistemática, preparando o terreno para a transição futura ao planejamento de rotas com objetivos específicos.

Etapa 3: Agente Baseado em Objetivos

Na terceira etapa do projeto, o agente deixou de ser apenas um explorador reativo e passou a agir com propósito definido, sendo capaz de planejar rotas até um objetivo. Para isso, utilizamos o algoritmo de Busca em Largura (BFS), escolhido por garantir o caminho mais curto em um ambiente não ponderado, como o grid proposto. A lógica do algoritmo se baseia em explorar todos os vizinhos de um nó antes de avançar para níveis mais profundos, utilizando uma fila para organizar os próximos passos. Para reconstruir a rota final, foi implementado um dicionário de predecessores, permitindo que o caminho fosse reconstituído do destino até a origem de forma eficiente.

O ambiente foi configurado em duas versões. Primeiro, um grid estático e totalmente observável, sem obstáculos, onde o agente precisava apenas navegar até o ponto final. Em seguida, foram introduzidos os obstáculos já utilizados na etapa anterior, exigindo que o algoritmo desviasse de bloqueios e encontrasse uma rota válida. Em ambos os cenários, o agente tinha conhecimento completo do mapa (limites e obstáculos) antes de iniciar o planejamento. Para validar a corretude do algoritmo, a simulação contou com uma animação que mostrava o robô percorrendo o caminho passo a passo, deixando claro o processo de decisão até alcançar o destino.

Os resultados foram bastante satisfatórios, atendendo plenamente às métricas de avaliação. Na métrica de Sucesso na Tarefa, o agente sempre conseguiu atingir o objetivo sempre que existia um caminho possível, tanto no ambiente livre quanto no com obstáculos. Quanto ao Comprimento do Caminho, o desempenho foi ótimo, já

que o BFS retorna, por definição, a rota mais curta em termos de número de passos. De forma geral, essa etapa representou uma transição muito didática entre os conceitos de exploração (da etapa anterior) e de planejamento. A adaptação do BFS, que antes encontrava a célula não visitada mais próxima e agora buscava um alvo específico, mostrou como um mesmo algoritmo pode ser aplicado em diferentes contextos com pequenas modificações.

Etapa 4: Agente Baseado em Utilidade

Na quarta etapa, o agente passou a lidar com um desafio mais sofisticado: não bastava encontrar o caminho mais curto, mas sim o de menor custo total. Para isso, implementamos o algoritmo A* (A-Estrela), uma técnica de busca informada que combina o custo real de cada trajeto já percorrido ($g(n)$) com uma estimativa do custo restante até o objetivo ($h(n)$). Utilizamos a distância de Manhattan como heurística, ideal para grids, garantindo eficiência no cálculo. Com essa abordagem, o A* prioriza os caminhos mais promissores ao avaliar $f(n) = g(n) + h(n)$, explorando o mapa de forma muito mais inteligente que o BFS, que se limita a contar passos sem considerar os custos do terreno.

O ambiente de testes foi configurado em duas versões. Na primeira, totalmente observável, o agente tinha conhecimento de todos os custos do grid desde o início, o que permitia calcular a rota ótima de forma direta. Já na segunda, parcialmente observável, o agente só descobria os custos das células vizinhas à medida que avançava, simulando um verdadeiro “nevoeiro de guerra”. Nessa situação, o agente precisou adotar um ciclo contínuo de percepção, planejamento e ação, reavaliando sua rota a cada passo. Essa dinâmica trouxe um nível mais realista de tomada de decisão, aproximando o comportamento do agente de aplicações práticas em cenários de exploração e jogos.

Os resultados mostraram claramente a força do A*. Em ambos os ambientes, o agente atingiu o objetivo com Sucesso na Tarefa, e a métrica de Custo Total do Caminho foi otimizada, com o algoritmo escolhendo rotas mais longas em número de passos quando isso significava reduzir o custo total — por exemplo, evitando terrenos rochosos (custo 3) e arenosos (custo 2). Essa foi a etapa mais desafiadora e gratificante, especialmente pela implementação do replanejamento no ambiente parcialmente observável, que exigiu adaptação constante do agente. A visualização foi fundamental para validar o comportamento, deixando clara a diferença entre o caminho “curto” que o BFS escolheria e o caminho “barato” encontrado pelo A*.