

Tugas Robotika Week 5

Nama : Nur Ihsan Ibrahim Abdul Fattah

Kelas :TK45G09

NIM : 1103210191

Analisis dijkstra_planning.py

Algoritma Dijkstra untuk menemukan jalur terpendek dari titik awal ke titik tujuan dalam graf berbobot yang direpresentasikan sebagai dictionary bernama graph. Fungsi dijkstra menggunakan heapq untuk mengelola antrian prioritas, di mana node dengan biaya terendah selalu diproses terlebih dahulu. Fungsi ini menyimpan node yang sudah diproses dalam set seen dan mencatat jarak minimum dari titik awal ke setiap node dalam dictionary min_distance. Dalam setiap iterasi, fungsi memeriksa tetangga dari node saat ini; jika jarak baru yang dihitung lebih rendah dari jarak minimum sebelumnya, jarak tersebut diperbarui, dan node tersebut ditambahkan ke antrian. Jika tujuan tercapai, fungsi mengembalikan biaya total dan jalur yang ditempuh; jika tidak ada jalur yang ditemukan, fungsi mengembalikan nilai tak terhingga. Contoh graf di akhir menunjukkan penggunaan fungsi ini untuk mencari jalur dari 'A' ke 'D' dengan hasil biaya dan jalur terpendek yang ditampilkan.

Analisis a_star_planning.py

Algoritma A* untuk menemukan jalur terpendek dari titik awal ke titik tujuan dalam grid berbentuk matriks 2D, di mana nilai 0 menunjukkan area yang dapat dilalui dan 1 menunjukkan rintangan. Fungsi a_star menggunakan dua list: open_list untuk node yang perlu diperiksa dan closed_list untuk node yang sudah diproses. Jarak antara node dihitung dengan fungsi heuristik heuristic, yang menggunakan jarak Manhattan. Untuk setiap node, tetangga yang dapat dilalui dihitung oleh fungsi get_neighbors. Algoritma terus memproses node dengan nilai f terendah (jumlah jarak dari awal dan estimasi jarak ke tujuan) hingga mencapai tujuan atau kehabisan node yang dapat diperiksa. Jika jalur ditemukan, algoritma mengembalikan jalur dari start ke goal; jika tidak, mengembalikan list kosong. Pada contoh grid, jalur yang ditemukan dari (0, 0) ke (3, 3) ditampilkan.

Analisis cell_decompotision.py

Metode cell decomposition untuk mencari jalur dari titik awal ke tujuan dalam lingkungan yang memiliki rintangan. Fungsi cell_decomposition membagi area menjadi sel-sel yang bebas dari rintangan berdasarkan posisi rintangan yang diberikan, kemudian mencari jalur melalui sel-sel ini dari start ke goal. Fungsi create_cells bertanggung jawab membuat sel-sel tersebut, tetapi saat ini hanya mengembalikan contoh sel sederhana yang dapat dikembangkan lebih lanjut. Fungsi find_path_through_cells kemudian mencari jalur melalui sel-sel ini, meskipun dalam contoh ini ia hanya mengembalikan jalur langsung dari start ke goal. Dalam contoh yang diberikan, rintangan berada di posisi (1, 1, 2, 2), dan jalur dari (0, 0) ke (3, 3) dihasilkan serta ditampilkan di output.