Grid_World_problem_p8

November 2, 2025

Simple Grid World Problem: Design a custom 2D grid world where the agent navigates from a start position to a goal, avoiding obstacles. Environment: Custom grid (easily implemented in Python)

[9]:

```python
import numpy as np
import os
import time

# Step 1: Create Grid World
grid = np.array([
    ['S', ' ', ' ', 'X', ' ', ' '],
    [' ', 'X', ' ', ' ', 'X', ' '],
    [' ', ' ', ' ', ' ', ' ', 'G']
])

# Step 2: Define start and goal
start_pos = (0, 0)
goal_pos = (2, 5)

# Step 3: Movement directions
moves = {
    'W': (-1, 0),  # Up
    'S': (1, 0),   # Down
    'A': (0, -1),  # Left
    'D': (0, 1)    # Right
}

# Step 4: Function to print grid with agent position
def print_grid(agent_pos):
    os.system('cls' if os.name == 'nt' else 'clear')
    display = grid.copy()
    display[start_pos] = 'S'
    display[goal_pos] = 'G'
    if agent_pos != start_pos and agent_pos != goal_pos:
        display[agent_pos] = 'A'  # Agent
    for row in display:
        print(' '.join(row))
    print("\nControls: W = Up | S = Down | A = Left | D = Right | Q = Quit")
```

```python
# Step 5: Function to move agent
def move_agent(position, direction):
    if direction not in moves:
        return position
    new_pos = (position[0] + moves[direction][0], position[1] + moves[direction][1])
    # Check boundaries and obstacles
    if (0 <= new_pos[0] < grid.shape[0]) and (0 <= new_pos[1] < grid.shape[1]) and grid[new_pos] != 'X':
        return new_pos
    return position  # Invalid move

# Step 6: Interactive game loop
agent_pos = start_pos
print_grid(agent_pos)

while True:
    move = input("Move (W/A/S/D): ").upper()
    if move == 'Q':
        print("  Exiting the game.")
        break
    agent_pos = move_agent(agent_pos, move)
    print_grid(agent_pos)
    if agent_pos == goal_pos:
        print("\n  Goal Reached! Congratulations!")
        break
```

```
S     X
  X     X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  A

S     X
  X     X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  S

S     X
A X     X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  A
```

```
S     X
A X     X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  W

S     X
  X     X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  A

S     X
  X     X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  D

S A   X
  X     X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  D

S   A X
  X     X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  S

S     X
  X A   X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  D

S     X
  X   A X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit
```

```
Move (W/A/S/D):  S

S     X
  X     X
      A   G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  D

S     X
  X     X
        A G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

Move (W/A/S/D):  D

S     X
  X     X
          G

Controls: W = Up | S = Down | A = Left | D = Right | Q = Quit

 Goal Reached! Congratulations!
```

[ ]:

```

```

[ ]:

```

```