

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Вступ до Штучного Інтелекту
Лабораторна робота №2
«Інтелектуальні агенти»

Виконав:
студент групи ІО-05
Кобилинський Ф. О.
Перевірів:
Трочун Є.В.

Київ - 2022 р.

Мета роботи: розробити інтелектуального агента-машину, що рухається по графу-дорозі з попередньої лабораторної роботи за заданими правилами та метою. Отримати практичні навички роботи з інтелектуальними агентами.

Завдання:

- Отримати навички роботи з інтелектуальними агентами.
- Розробити раціонального агента-автомобіль, що рухається по дорозі з попередньої лабораторної роботи.

Теоретичні відомості

Агент — це будь-який об'єкт чи програма, що може розглядатися як такий, що сприймає своє оточення за допомогою датчиків і діє на це оточення за допомогою певних виконавчих механізмів. Ми використовуємо термін сприйняття для позначення перцептивних вхідних даних агента в будь-який момент. Послідовність сприйняття агента - це повна історія всього, що агент коли-небудь сприймав. Загалом вибір дії агента в будь-який момент може залежати від усієї послідовності попередньо сприйнятої інформації, але не від того, що він ще не сприйняв. Ми можемо сказати, що поведінка агента описується функцією агента, яка перетворює будь-яку послідовність сприйняття на дію.

Раціональний агент – це агент, який виконує раціональні дії, зважаючи на свою послідовність сприйняття та мету. Для того, щоб зробити висновок, чи вчинив агент правильно, нам потрібен показник ефективності, на якому будуть базуватись його наступні дії для сприйняття середовища та досягнення мети. Як правило, краще розробляти показники ефективності відповідно до того, що потрібно отримати в середовищі, а не відповідно до того, як, на вашу думку, повинен поводитися агент.

Те, що саме є раціональним у будь-який момент часу, залежить від наступних показників:

- Показник ефективності, який визначає критерій успіху
- Попередні знання агента про середовище
- Дії, які може виконувати агент
- Послідовність сприйняття агента на даний момент

Отже, раціональний агент для кожної можливої послідовності сприйняття повинен вибрати дію, яка, як очікується, максимізує показник ефективності, враховуючи докази, надані послідовністю сприйняття, і будь-які вбудовані знання, якими володіє агент.

Опис алгоритму:

Дана лабораторна робота розроблена на основі першої (map_drawing), тому опису другу функцію пошуку цілі (achieve_goal). Спочатку подаємо на ввід координати початку та фінішу вершин та викликаємо функцію пошуку. Визначаємо змінні старту, фінішу та поточної вершини. Пошук працює на основі алгоритму пошуку в глибину. В словнику visited ми змінюємо false на true, якщо ми відвідали відповідну вершину. В stack закидуємо можливі вершини, в які можна перейти з поточної. Також як тільки ми перейшли на іншу вершинну, то з стеку видаляється вершина куди пішли і фарбуємо вершину в чорний. Коли ж координати фінішу є в списку результатів(пройдених вершин), цикл завершується.

Програмний код:

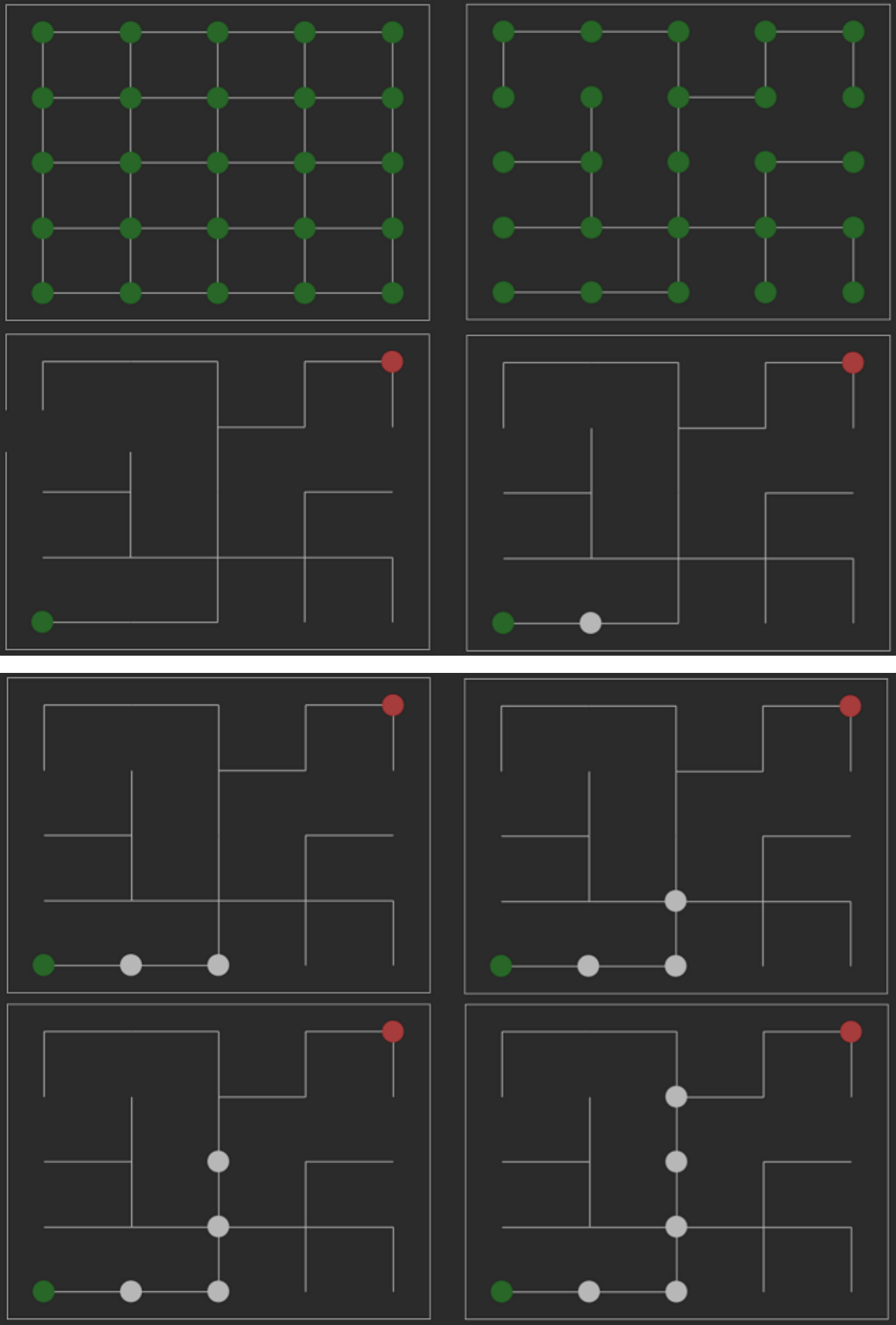
```
In 5 1 import matplotlib.pyplot as plt
      2 import networkx as nx
      3 import random
```

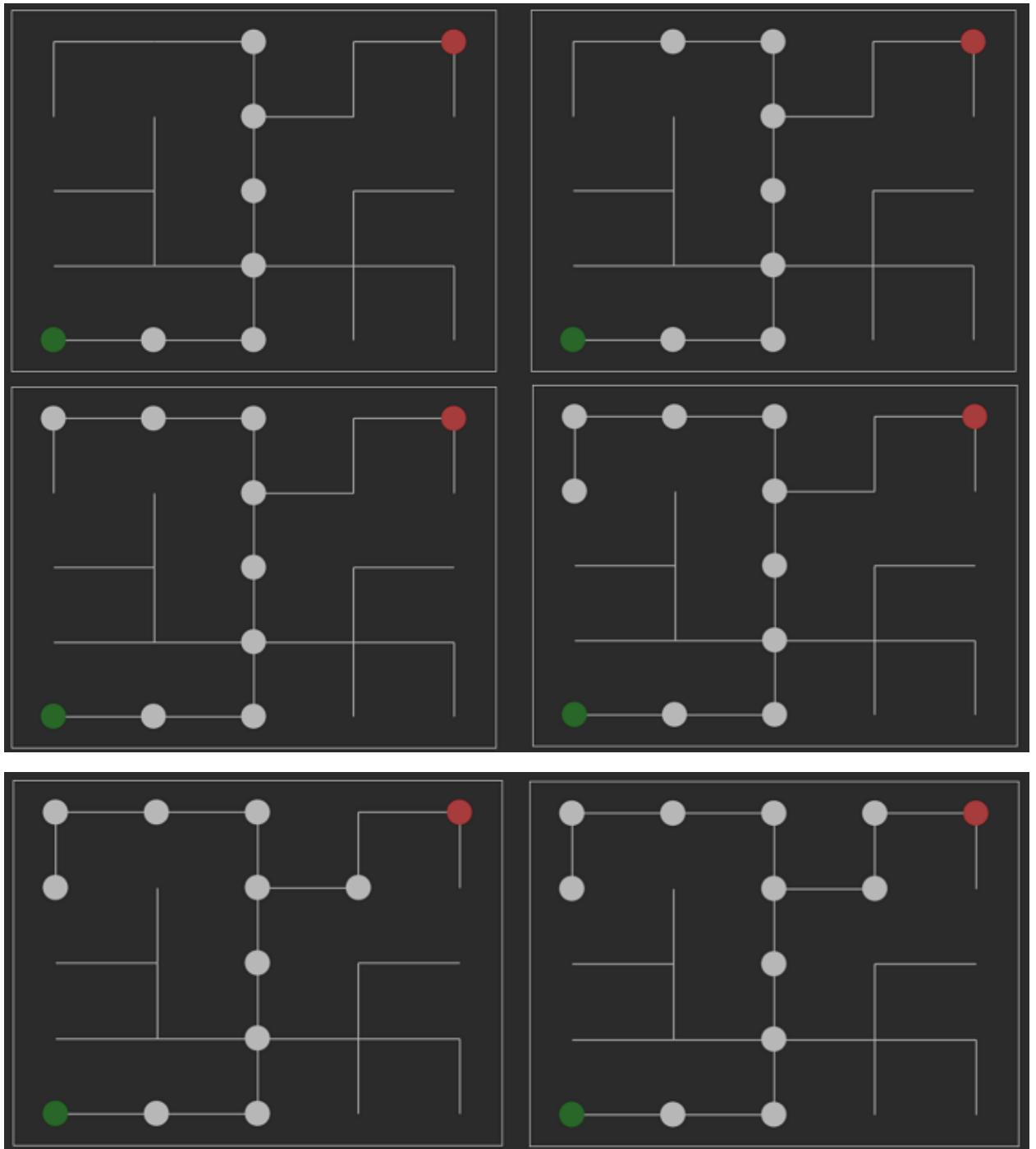
```
In 6 1 def map_drawing(delete, N=5):
      2     exeption = (N - 1) ** 2
      3
      4     if delete > exeption:
      5         print("Ви ввели число більше допустимого! Граф не буде зв'язним.")
      6     else:
      7         global G
      8         global pos
      9         G = nx.grid_2d_graph(N, N)
     10         pos = dict((n, n) for n in G.nodes())
     11         nx.draw_networkx(G, pos=pos, with_labels=False, node_size=300, node_color="PaleGreen")
     12         plt.show()
     13
     14         while delete > 0:
     15             edges = list(G.edges)
     16             chosen_edge = random.choice(edges)
     17             G.remove_edge(chosen_edge[0], chosen_edge[1])
     18
     19             if nx.is_connected(G):
     20                 delete -= 1
     21             else:
     22                 G.add_edge(chosen_edge[0], chosen_edge[1])
     23         nx.draw_networkx(G, pos=pos, with_labels=False, node_size=300, node_color="PaleGreen")
     24         plt.show()
```

```
In 7 1 def achieve_goal(x, y, i, j):
      2     try:
      3         active = (x, y)
      4         start = active
      5         finish = (i, j)
      6
      7         visited = {node: False for node in G.nodes}
      8         stack = [start]
      9         result = []
     10         colors = {active: "PaleGreen", finish: "Red"}
     11         while stack:
     12             nx.draw_networkx(G, pos=pos, with_labels=False, nodelist=colors,
     13                             node_color=colors.values(), node_size=300)
     14             plt.show()
     15             curently = stack.pop()
     16             if not visited[curently]:
     17                 result.append(curently)
     18                 visited[curently] = True
     19                 colors[curently] = 'black'
     20             for node in G.neighbors(curently):
     21                 if not visited[node]:
     22                     stack.append(node)
     23             if finish in result:
     24                 break
     25         except:
     26             print("Помилка вводу координат")
```

```
In 8 1 if __name__ == '__main__':
      2     N = int(input("Введіть розмірність матриці: "))
      3     delete = int(input("Введіть кількість ребер на видалення: "))
      4     x = int(input("Введіть координату початку пошуку(x, y) x: "))
      5     y = int(input("Введіть координату початку пошуку(x, y) y: "))
      6     i = int(input("Введіть координату фінішу(i, j) i: "))
      7     j = int(input("Введіть координату фінішу(i, j) j: "))
      8
      9     map_drawing(delete, N)
     10     achieve_goal(x, y, i, j)
```

Результат роботи програми:





Висновок: в даній лабораторній роботі, я доповнив свою попередню роботу функціоналом пошуку маршруту. Отримав навички роботи з інтелектуальними агентами, закріпив свої знання з бібліотеками `network` та `matplotlib`.