Як ви сказали в лекції, очевидним недоліком агента s3 є його довге блукання на одному і тому самому місці, відповідно удосконалений код штрафує робота який залишається довго на одному і тому самому місці.

**Пояснення внесених змін:**

1. **Додано історію позицій для агента s3:**

s3.body.position\_history = [] # Історія позицій

s3.body.lingering = False # Прапорець нерухомості

1. **Оновлено Q-таблицю для додаткового стану:**

s3.body.Q=[[0, 0], [0, 0], [0, 0], [0, 0]] # Додано новий стан

Тепер Q-таблиця має 4 стани:

* + Стан 0: Нічого в секторі
  + Стан 1: Об'єкт s1 в секторі
  + Стан 2: Антиоб'єкт s2 в секторі
  + Стан 3: Агент знаходиться на одному місці (нерухомість)

1. **Оновлено функцію strategy2:**
   * **Оновлення історії позицій та перевірка на нерухомість:**

історію позицій

if canvas.frame % 10 == 0:

b.position\_history.append((x, y))

if len(b.position\_history) > 500: # Зберігаємо позиції за останні 10 секунд (500 кадрів \* 0.02 с)

b.position\_history.pop(0)

# Перевірка на нерухомість

if len(b.position\_history) >= 2:

recent\_positions = b.position\_history[-250:] # Останні 5 секунд

avg\_x = sum(pos[0] for pos in recent\_positions) / len(recent\_positions)

avg\_y = sum(pos[1] for pos in recent\_positions) / len(recent\_positions)

avg\_dist = getDist(x, y, avg\_x, avg\_y)

if avg\_dist < 20: # Якщо агент знаходиться в межах 20 пікселів від середньої позиції

b.lingering = True

else:

b.lingering = False

* + **Визначення стану та винагороди з урахуванням нерухомості:**

if b.lingering:

state=3 # Новий стан - нерухомість

reward=-1 # Штраф за нерухомість

elif inS:

state=1

reward=1 if b.action==0 else -1

elif inS2:

state=2

reward=-1 if b.action==0 else 1

else:

state=0; reward=0

b.Q[state][b.action] += reward # оновлюємо Q таблицю

* + **Решта коду в strategy2 залишається майже без змін, але тепер агент враховує новий стан і може приймати рішення, щоб уникнути штрафів за нерухомість.**

1. **Додаткові пояснення:**
   * Ми оновлюємо історію позицій агента кожні 10 кадрів і зберігаємо її за останні 500 кадрів (що відповідає 10 секундам, оскільки кожен кадр - це 0.02 секунди).
   * Перевіряємо, чи середня відстань від поточної позиції до середньої позиції за останні 5 секунд менша за певний поріг (20 пікселів). Якщо так, то вважаємо, що агент "топчеться на місці".
   * Якщо агент довго знаходиться на одному місці, він переходить у новий стан (стан 3) і отримує штраф за нерухомість.
   * Q-таблиця оновлюється відповідно до нового стану та винагороди, що дозволяє агенту навчитися уникати нерухомості.

**Тепер агент s3 буде штрафуватися за залишання на одному і тому ж місці, що стимулюватиме його досліджувати нові області простору і уникати "топтання на місці".**