```
!pip install "pettingzoo[mpe]" supersuit gymnasium torch numpy tqdm
```

```
Collecting supersuit
  Downloading supersuit-3.10.0-py3-none-any.whl.metadata (3.1 kB)
Requirement already satisfied: gymnasium in /usr/local/lib/python3.12/dist-packages (1.2.1)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.8.0+cu126)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (2.0.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (4.67.1)
Collecting pettingzoo[mpe]
  Downloading pettingzoo-1.25.0-py3-none-any.whl.metadata (8.9 kB)
Requirement already satisfied: pygame>=2.3.0 in /usr/local/lib/python3.12/dist-packages (from pettingzoo[mpe]) (2.6.1)
Collecting tinyscaler>=1.2.6 (from supersuit)
  Downloading tinyscaler-1.2.8-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.2 kB)
Requirement already satisfied: cloudpickle>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from gymnasium) (3.1.1)
Requirement already satisfied: typing-extensions>=4.3.0 in /usr/local/lib/python3.12/dist-packages (from gymnasium) (4.15.0)
Requirement already satisfied: farama-notifications>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from gymnasium) (0.0.
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch) (3.20.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch) (3.5)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch) (9.10.2.
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.4.
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch) (11.3.0.4
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch) (10.3.7
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch) (11.7.
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch) (12.5.
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in /usr/local/lib/python3.12/dist-packages (from torch) (2.27.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch) (1.11.1.
Requirement already satisfied: triton==3.4.0 in /usr/local/lib/python3.12/dist-packages (from torch) (3.4.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch) (1.
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch) (3.0.3)
Downloading supersuit-3.10.0-py3-none-any.whl (50 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 50.2/50.2 kB 2.0 MB/s eta 0:00:00
Downloading tinyscaler-1.2.8-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (563 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 563.6/563.6 kB 13.9 MB/s eta 0:00:00
Downloading pettingzoo-1.25.0-py3-none-any.whl (852 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 852.5/852.5 kB 29.3 MB/s eta 0:00:00
Installing collected packages: tinyscaler, supersuit, pettingzoo
Successfully installed pettingzoo-1.25.0 supersuit-3.10.0 tinyscaler-1.2.8
```

```
import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim
from pettingzoo.mpe import simple_spread_v3
from tqdm import trange

# --- Q-Network ---
class QNetwork(nn.Module):
    def __init__(self, obs_size, act_size):
        super().__init__()
        self.net = nn.Sequential(
            nn.Linear(obs_size, 128),
            nn.ReLU(),
            nn.Linear(128, act_size)
        )
    def forward(self, x):
        return self.net(x)

# --- Hyperparameters ---
EPISODES = 200
GAMMA = 0.95
LR = 1e-3
EPSILON = 0.2
BATCH_SIZE = 64
MEMORY_SIZE = 5000

# --- Experience Replay Buffer ---
class ReplayBuffer:
    def __init__(self, capacity):
        self.capacity = capacity
        self.memory = []
    def push(self, data):
        if len(self.memory) >= self.capacity:
            self.memory.pop(0)
```

```
            self.memory.append(data)
        def sample(self, batch_size):
            idx = np.random.choice(len(self.memory), batch_size)
            return [self.memory[i] for i in idx]
        def __len__(self):
            return len(self.memory)

    # --- Environment ---
    env = simple_spread_v3.env(N=3, local_ratio=0.5, max_cycles=25)
    env.reset()

    # --- Initialize agents ---
    agents = env.possible_agents
    obs_size = env.observation_space(agents[0]).shape[0]
    act_size = env.action_space(agents[0]).n

    q_nets = {a: QNetwork(obs_size, act_size) for a in agents}
    optims = {a: optim.Adam(q_nets[a].parameters(), lr=LR) for a in agents}
    buffers = {a: ReplayBuffer(MEMORY_SIZE) for a in agents}

    # --- Training Loop ---
    for ep in trange(EPISODES, desc="Training MARL Agents"):
        env.reset()
        episode_rewards = {a: 0 for a in agents}

        for agent in env.agent_iter():
            obs, rew, term, trunc, info = env.last()
            if term or trunc:
                env.step(None)
                continue

            obs_t = torch.tensor(obs, dtype=torch.float32).unsqueeze(0)
            if np.random.rand() < EPSILON:
                action = env.action_space(agent).sample()
            else:
                with torch.no_grad():
                    q_vals = q_nets[agent](obs_t)
                    action = torch.argmax(q_vals).item()

            env.step(action)
            next_obs, next_rew, next_term, next_trunc, _ = env.last()
            episode_rewards[agent] += rew

            buffers[agent].push((obs, action, next_rew, next_obs, next_term))

            # --- Train ---
            if len(buffers[agent]) >= BATCH_SIZE:
                batch = buffers[agent].sample(BATCH_SIZE)
                obs_b, act_b, rew_b, next_obs_b, done_b = zip(*batch)
                obs_b = torch.tensor(obs_b, dtype=torch.float32)
                act_b = torch.tensor(act_b, dtype=torch.int64).unsqueeze(1)
                rew_b = torch.tensor(rew_b, dtype=torch.float32).unsqueeze(1)
                next_obs_b = torch.tensor(next_obs_b, dtype=torch.float32)
                done_b = torch.tensor(done_b, dtype=torch.float32).unsqueeze(1)

                q_vals = q_nets[agent](obs_b).gather(1, act_b)
                with torch.no_grad():
                    next_q = q_nets[agent](next_obs_b).max(1, keepdim=True)[0]
                    target = rew_b + GAMMA * (1 - done_b) * next_q

                loss = nn.MSELoss()(q_vals, target)
                optims[agent].zero_grad()
                loss.backward()
                optims[agent].step()

        total_reward = sum(episode_rewards.values()) / len(agents)
        if ep % 20 == 0:
            print(f"Episode {ep}, Avg Reward per Agent: {total_reward:.3f}")

    print("Training Complete ✅")

    # --- Visualization ---
    env = simple_spread_v3.env(render_mode="human")
    env.reset()
    for agent in env.agent_iter():
        obs, _, term, trunc, _ = env.last()
        if term or trunc:
            env.step(None)
            continue
        obs_t = torch.tensor(obs, dtype=torch.float32).unsqueeze(0)
        with torch.no_grad():
            q_vals = q_nets[agent](obs_t)
```

```
            action = torch.argmax(q_vals).item()
        env.step(action)
```

```
/tmp/ipython-input-2980010755.py:5: DeprecationWarning: The environment `pettingzoo.mpe` has been moved to `mpe2` and will b
  from pettingzoo.mpe import simple_spread_v3
Training MARL Agents:   0%|          | 0/200 [00:00<?, ?it/s]/tmp/ipython-input-2980010755.py:85: UserWarning: Creating a te
  obs_b = torch.tensor(obs_b, dtype=torch.float32)
Episode 0, Avg Reward per Agent: -21.048
Training MARL Agents:  10%|▉         | 21/200 [00:05<00:56,  3.19it/s]Episode 20, Avg Reward per Agent: -13.961
Training MARL Agents:  20%|██        | 41/200 [00:10<00:40,  3.90it/s]Episode 40, Avg Reward per Agent: -21.742
Training MARL Agents:  30%|███       | 61/200 [00:15<00:33,  4.16it/s]Episode 60, Avg Reward per Agent: -10.108
Training MARL Agents:  40%|████      | 81/200 [00:20<00:29,  4.07it/s]Episode 80, Avg Reward per Agent: -39.688
Training MARL Agents:  50%|█████     | 101/200 [00:25<00:25,  3.91it/s]Episode 100, Avg Reward per Agent: -24.195
Training MARL Agents:  60%|██████    | 121/200 [00:32<00:22,  3.56it/s]Episode 120, Avg Reward per Agent: -29.672
Training MARL Agents:  70%|███████   | 141/200 [00:37<00:14,  4.13it/s]Episode 140, Avg Reward per Agent: -31.276
Training MARL Agents:  80%|████████  | 161/200 [00:42<00:13,  2.88it/s]Episode 160, Avg Reward per Agent: -36.310
Training MARL Agents:  90%|█████████ | 181/200 [00:47<00:04,  4.38it/s]Episode 180, Avg Reward per Agent: -34.693
Training MARL Agents: 100%|██████████| 200/200 [00:52<00:00,  3.80it/s]
Training Complete ✅
```