In [1]:
```
import torch
import pygame
import gymnasium as gym
```

```
pygame 2.1.0 (SDL 2.0.16, Python 3.9.12)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

In [2]:
```
# Run the code below to test the environment with keyboard input
env_name = "LunarLander-v3"

actions = {
    pygame.K_DOWN: 0,     # do nothing
    pygame.K_LEFT: 1,     # fire left orientation engine
    pygame.K_UP: 2,       # fire main engine
    pygame.K_RIGHT: 3,    # fire right orientation engine
}

pygame.init()
env = gym.make(env_name, render_mode="human")

# Reset the environment with a seed
env.reset()

# Game loop
done = False
while not done:
    # Render the environment
    env.render()

    # Check for events (keyboard input)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:  # If the window close button is pressed
            done = True
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_ESCAPE:  # If ESC is pressed
                done = True

    # Default action (no key pressed)
    action = 0

    # Check for key press and map it to action
    keys = pygame.key.get_pressed()
```

```
    for key, action_value in actions.items():
        if keys[key]:
            action = action_value
            break

    # Step through the environment with the chosen action
    observation, reward, terminated, truncated, info = env.step(action)

    # Check if the episode is over
    if terminated or truncated:
        observation, info = env.reset()

# Close the environment after the game loop
env.close()
pygame.quit()
```

In [3]:
```
class PolicyNet(torch.nn.Module):
    def __init__(self, state_dim, action_dim):
        super(PolicyNet, self).__init__()
        NotImplemented

    def forward(self, x):
        NotImplemented

class REINFORCE:
    def __init__(self, states, actions, device):
        self.device = device
        self.action = actions
        self.policy_net = PolicyNet(states, actions).to(device)
        NotImplemented

    def take_action(self, state):
        NotImplemented

    def update(self):
        NotImplemented
```

In [4]:
```
device = torch.device("cuda") if torch.cuda.is_available() else torch.device("cpu")

env = gym.make(env_name, render_mode="rgb_array")

obs_space = env.observation_space.shape[0]   # 8, continuous
action_space = env.action_space.n   # 4, discrete
```

```
state, _ = env.reset()

episodes = 500

agent = REINFORCE(obs_space, action_space, device)

# Training Code Below
```

In [22]:

```
### 落地了还会喷气我也不知道为什么那就这样吧
# Training code
import torch.nn as nn
import torch.optim as optim
import numpy as np
import matplotlib.pyplot as plt
from tqdm import tqdm

class PolicyNet(nn.Module):
    def __init__(self, state_dim, action_dim):
        super(PolicyNet, self).__init__()
        self.fc1 = nn.Linear(state_dim, 64)
        self.fc2 = nn.Linear(64, action_dim)

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = self.fc2(x)
        return x

class REINFORCE:
    def __init__(self, state_dim, action_dim, device, gamma=0.99, lr=0.002):
        self.device = device
        self.gamma = gamma

        self.policy_net = PolicyNet(state_dim, action_dim).to(device)
        self.optimizer = optim.Adam(self.policy_net.parameters(), lr=lr)

        self.log_probs = []
        self.rewards = []

    def take_action(self, state):
        state_tensor = torch.tensor(state, dtype=torch.float32).unsqueeze(0).to(self.device)
        logits = self.policy_net(state_tensor)
        dist = torch.distributions.Categorical(logits=logits)
```

```
        action = dist.sample()
        self.log_probs.append(dist.log_prob(action))
        return action.item()

    def update(self):
        returns = []
        G = 0
        for r in reversed(self.rewards):
            G = r + self.gamma * G
            returns.insert(0, G)

        returns = torch.tensor(returns, dtype=torch.float32).to(self.device)
        returns = (returns - returns.mean()) / (returns.std() + 1e-9)

        loss = 0
        for log_prob, Gt in zip(self.log_probs, returns):
            loss += -log_prob * Gt
        loss = loss / len(self.rewards)

        self.optimizer.zero_grad()
        loss.backward()
        self.optimizer.step()

        self.log_probs.clear()
        self.rewards.clear()

        return loss.item()

env_name = "LunarLander-v3"
env = gym.make(env_name)
device = torch.device("cuda") if torch.cuda.is_available() else torch.device("cpu")

obs_space = env.observation_space.shape[0]  # 8
action_space = env.action_space.n  # 4

total_episodes = 15000
episodes_per_iteration = 100
iterations = total_episodes // episodes_per_iteration

agent = REINFORCE(obs_space, action_space, device, gamma=0.99, lr=0.0012) # 实际上和0.001也差不多

return_list = []
length_list = []
loss_list = []
```

```python
for i_iter in range(iterations):
    with tqdm(range(episodes_per_iteration), desc=f"Iteration {i_iter}", ncols=100) as pbar:
        for i_episode_in_iter in pbar:
            i_episode = i_iter * episodes_per_iteration + i_episode_in_iter
            state, _ = env.reset(seed=i_episode)
            done = False
            episode_reward = 0
            episode_length = 0

            while not done:
                action = agent.take_action(state)
                next_state, reward, terminated, truncated, info = env.step(action)
                done = terminated or truncated

                agent.rewards.append(reward)
                state = next_state
                episode_reward += reward
                episode_length += 1

            loss = agent.update()
            return_list.append(episode_reward)
            length_list.append(episode_length)
            loss_list.append(loss)

        # 平均回报
    avg_return = np.mean(return_list[-episodes_per_iteration:])
    print(f"Episode: {(i_iter+1)*episodes_per_iteration}, Average Return: {avg_return:.2f}")

env.close()

fig, axes = plt.subplots(1, 3, figsize=(15, 5))

axes[0].plot(return_list)
axes[0].set_title("Episode Rewards")
axes[0].set_xlabel("Episode")
axes[0].set_ylabel("Reward")
axes[0].grid(True)

axes[1].plot(length_list)
axes[1].set_title("Episode Lengths")
axes[1].set_xlabel("Episode")
axes[1].set_ylabel("Length")
axes[1].grid(True)
```

```python
axes[2].plot(loss_list)
axes[2].set_title("Training Error")
axes[2].set_xlabel("Episode")
axes[2].set_ylabel("Loss")
axes[2].grid(True)

plt.tight_layout()
plt.show()

torch.save(agent.policy_net.state_dict(), "reinforce_lunarlander.pth")
```

```
Iteration 0: 100%|| 100/100 [00:04<00:00, 20.59it/s]
Episode: 100, Average Return: -169.24
Iteration 1: 100%|| 100/100 [00:05<00:00, 19.06it/s]
Episode: 200, Average Return: -153.75
Iteration 2: 100%|| 100/100 [00:05<00:00, 18.32it/s]
Episode: 300, Average Return: -138.76
Iteration 3: 100%|| 100/100 [00:05<00:00, 17.75it/s]
Episode: 400, Average Return: -120.21
Iteration 4: 100%|| 100/100 [00:06<00:00, 15.93it/s]
Episode: 500, Average Return: -108.95
Iteration 5: 100%|| 100/100 [00:05<00:00, 17.27it/s]
Episode: 600, Average Return: -89.69
Iteration 6: 100%|| 100/100 [00:07<00:00, 14.24it/s]
Episode: 700, Average Return: -78.90
Iteration 7: 100%|| 100/100 [00:09<00:00, 10.54it/s]
Episode: 800, Average Return: -89.26
Iteration 8: 100%|| 100/100 [00:09<00:00, 10.08it/s]
Episode: 900, Average Return: -73.95
Iteration 9: 100%|| 100/100 [00:15<00:00,  6.59it/s]
Episode: 1000, Average Return: -91.99
Iteration 10: 100%|| 100/100 [00:21<00:00,  4.75it/s]
Episode: 1100, Average Return: -108.50
Iteration 11: 100%|| 100/100 [00:25<00:00,  3.99it/s]
Episode: 1200, Average Return: -85.88
Iteration 12: 100%|| 100/100 [00:25<00:00,  3.98it/s]
Episode: 1300, Average Return: -62.89
Iteration 13: 100%|| 100/100 [00:36<00:00,  2.77it/s]
Episode: 1400, Average Return: -57.18
```

```
Iteration 14: 100%|████████████████████████████████████████| 100/100 [00:25<00:00,  3.85it/s]
Episode: 1500, Average Return: -19.31
Iteration 15: 100%|████████████████████████████████████████| 100/100 [00:37<00:00,  2.68it/s]
Episode: 1600, Average Return: -47.90
Iteration 16: 100%|████████████████████████████████████████| 100/100 [00:39<00:00,  2.51it/s]
Episode: 1700, Average Return: -34.86
Iteration 17: 100%|████████████████████████████████████████| 100/100 [00:39<00:00,  2.52it/s]
Episode: 1800, Average Return: -19.72
Iteration 18: 100%|████████████████████████████████████████| 100/100 [00:39<00:00,  2.52it/s]
Episode: 1900, Average Return: -14.99
Iteration 19: 100%|████████████████████████████████████████| 100/100 [00:40<00:00,  2.47it/s]
Episode: 2000, Average Return: 2.52
Iteration 20: 100%|████████████████████████████████████████| 100/100 [00:42<00:00,  2.35it/s]
Episode: 2100, Average Return: 8.95
Iteration 21: 100%|████████████████████████████████████████| 100/100 [00:43<00:00,  2.30it/s]
Episode: 2200, Average Return: 13.52
Iteration 22: 100%|████████████████████████████████████████| 100/100 [00:44<00:00,  2.25it/s]
Episode: 2300, Average Return: 12.03
Iteration 23: 100%|████████████████████████████████████████| 100/100 [00:44<00:00,  2.27it/s]
Episode: 2400, Average Return: 25.23
Iteration 24: 100%|████████████████████████████████████████| 100/100 [00:46<00:00,  2.17it/s]
Episode: 2500, Average Return: 12.97
Iteration 25: 100%|████████████████████████████████████████| 100/100 [00:41<00:00,  2.41it/s]
Episode: 2600, Average Return: 36.54
Iteration 26: 100%|████████████████████████████████████████| 100/100 [00:40<00:00,  2.46it/s]
Episode: 2700, Average Return: 43.24
Iteration 27: 100%|████████████████████████████████████████| 100/100 [00:43<00:00,  2.28it/s]
Episode: 2800, Average Return: 49.16
Iteration 28: 100%|████████████████████████████████████████| 100/100 [00:45<00:00,  2.20it/s]
Episode: 2900, Average Return: 69.96
Iteration 29: 100%|████████████████████████████████████████| 100/100 [00:46<00:00,  2.13it/s]
Episode: 3000, Average Return: 81.56
Iteration 30: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.06it/s]
Episode: 3100, Average Return: 99.49
Iteration 31: 100%|████████████████████████████████████████| 100/100 [00:45<00:00,  2.22it/s]
Episode: 3200, Average Return: 80.77
Iteration 32: 100%|████████████████████████████████████████| 100/100 [00:47<00:00,  2.11it/s]
Episode: 3300, Average Return: 103.94
```

```
Iteration 33: 100%|| 100/100 [00:48<00:00,  2.08it/s]
Episode: 3400, Average Return: 109.07
Iteration 34: 100%|| 100/100 [00:46<00:00,  2.13it/s]
Episode: 3500, Average Return: 116.00
Iteration 35: 100%|| 100/100 [00:44<00:00,  2.27it/s]
Episode: 3600, Average Return: 117.26
Iteration 36: 100%|| 100/100 [00:43<00:00,  2.32it/s]
Episode: 3700, Average Return: 115.61
Iteration 37: 100%|| 100/100 [00:50<00:00,  1.97it/s]
Episode: 3800, Average Return: 122.16
Iteration 38: 100%|| 100/100 [00:52<00:00,  1.89it/s]
Episode: 3900, Average Return: 123.40
Iteration 39: 100%|| 100/100 [00:54<00:00,  1.85it/s]
Episode: 4000, Average Return: 116.92
Iteration 40: 100%|| 100/100 [00:49<00:00,  2.03it/s]
Episode: 4100, Average Return: 125.87
Iteration 41: 100%|| 100/100 [00:49<00:00,  2.02it/s]
Episode: 4200, Average Return: 125.66
Iteration 42: 100%|| 100/100 [00:46<00:00,  2.14it/s]
Episode: 4300, Average Return: 134.17
Iteration 43: 100%|| 100/100 [00:46<00:00,  2.13it/s]
Episode: 4400, Average Return: 125.62
Iteration 44: 100%|| 100/100 [00:49<00:00,  2.01it/s]
Episode: 4500, Average Return: 119.17
Iteration 45: 100%|| 100/100 [00:50<00:00,  1.98it/s]
Episode: 4600, Average Return: 131.97
Iteration 46: 100%|| 100/100 [00:46<00:00,  2.16it/s]
Episode: 4700, Average Return: 132.25
Iteration 47: 100%|| 100/100 [00:37<00:00,  2.65it/s]
Episode: 4800, Average Return: 163.66
Iteration 48: 100%|| 100/100 [00:39<00:00,  2.53it/s]
Episode: 4900, Average Return: 157.63
Iteration 49: 100%|| 100/100 [00:41<00:00,  2.43it/s]
Episode: 5000, Average Return: 158.81
Iteration 50: 100%|| 100/100 [00:44<00:00,  2.27it/s]
Episode: 5100, Average Return: 154.95
Iteration 51: 100%|| 100/100 [00:35<00:00,  2.81it/s]
Episode: 5200, Average Return: 165.59
```

```
Iteration 52: 100%|| 100/100 [00:35<00:00,  2.79it/s]
Episode: 5300, Average Return: 184.20
Iteration 53: 100%|| 100/100 [00:30<00:00,  3.32it/s]
Episode: 5400, Average Return: 186.05
Iteration 54: 100%|| 100/100 [00:28<00:00,  3.53it/s]
Episode: 5500, Average Return: 161.96
Iteration 55: 100%|| 100/100 [00:34<00:00,  2.91it/s]
Episode: 5600, Average Return: 181.97
Iteration 56: 100%|| 100/100 [00:29<00:00,  3.37it/s]
Episode: 5700, Average Return: 162.06
Iteration 57: 100%|| 100/100 [00:35<00:00,  2.79it/s]
Episode: 5800, Average Return: 166.09
Iteration 58: 100%|| 100/100 [00:43<00:00,  2.31it/s]
Episode: 5900, Average Return: 146.57
Iteration 59: 100%|| 100/100 [00:48<00:00,  2.05it/s]
Episode: 6000, Average Return: 138.08
Iteration 60: 100%|| 100/100 [00:51<00:00,  1.93it/s]
Episode: 6100, Average Return: 125.74
Iteration 61: 100%|| 100/100 [00:43<00:00,  2.28it/s]
Episode: 6200, Average Return: 142.24
Iteration 62: 100%|| 100/100 [00:43<00:00,  2.29it/s]
Episode: 6300, Average Return: 144.08
Iteration 63: 100%|| 100/100 [00:45<00:00,  2.22it/s]
Episode: 6400, Average Return: 143.94
Iteration 64: 100%|| 100/100 [00:46<00:00,  2.16it/s]
Episode: 6500, Average Return: 148.65
Iteration 65: 100%|| 100/100 [00:50<00:00,  1.98it/s]
Episode: 6600, Average Return: 135.62
Iteration 66: 100%|| 100/100 [00:50<00:00,  1.96it/s]
Episode: 6700, Average Return: 132.49
Iteration 67: 100%|| 100/100 [00:50<00:00,  1.96it/s]
Episode: 6800, Average Return: 139.17
Iteration 68: 100%|| 100/100 [00:49<00:00,  2.03it/s]
Episode: 6900, Average Return: 141.42
Iteration 69: 100%|| 100/100 [00:49<00:00,  2.00it/s]
Episode: 7000, Average Return: 139.96
Iteration 70: 100%|| 100/100 [00:46<00:00,  2.15it/s]
Episode: 7100, Average Return: 142.67
```

```
Iteration 71: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.05it/s]
Episode: 7200, Average Return: 135.31
Iteration 72: 100%|████████████████████████████████████████| 100/100 [00:51<00:00,  1.95it/s]
Episode: 7300, Average Return: 135.12
Iteration 73: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.00it/s]
Episode: 7400, Average Return: 137.75
Iteration 74: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.99it/s]
Episode: 7500, Average Return: 132.50
Iteration 75: 100%|████████████████████████████████████████| 100/100 [00:51<00:00,  1.95it/s]
Episode: 7600, Average Return: 127.12
Iteration 76: 100%|████████████████████████████████████████| 100/100 [00:51<00:00,  1.95it/s]
Episode: 7700, Average Return: 127.65
Iteration 77: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.97it/s]
Episode: 7800, Average Return: 139.91
Iteration 78: 100%|████████████████████████████████████████| 100/100 [00:51<00:00,  1.96it/s]
Episode: 7900, Average Return: 136.45
Iteration 79: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.96it/s]
Episode: 8000, Average Return: 144.13
Iteration 80: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.98it/s]
Episode: 8100, Average Return: 140.37
Iteration 81: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.06it/s]
Episode: 8200, Average Return: 140.94
Iteration 82: 100%|████████████████████████████████████████| 100/100 [00:47<00:00,  2.11it/s]
Episode: 8300, Average Return: 135.37
Iteration 83: 100%|████████████████████████████████████████| 100/100 [00:51<00:00,  1.94it/s]
Episode: 8400, Average Return: 136.87
Iteration 84: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.05it/s]
Episode: 8500, Average Return: 137.66
Iteration 85: 100%|████████████████████████████████████████| 100/100 [00:51<00:00,  1.95it/s]
Episode: 8600, Average Return: 134.74
Iteration 86: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.98it/s]
Episode: 8700, Average Return: 134.85
Iteration 87: 100%|████████████████████████████████████████| 100/100 [00:51<00:00,  1.95it/s]
Episode: 8800, Average Return: 130.90
Iteration 88: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.96it/s]
Episode: 8900, Average Return: 133.07
Iteration 89: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.96it/s]
Episode: 9000, Average Return: 134.93
```

```
Iteration 90: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.99it/s]
Episode: 9100, Average Return: 137.12
Iteration 91: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.02it/s]
Episode: 9200, Average Return: 143.69
Iteration 92: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.08it/s]
Episode: 9300, Average Return: 144.09
Iteration 93: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.01it/s]
Episode: 9400, Average Return: 140.34
Iteration 94: 100%|████████████████████████████████████████| 100/100 [00:47<00:00,  2.09it/s]
Episode: 9500, Average Return: 145.00
Iteration 95: 100%|████████████████████████████████████████| 100/100 [00:37<00:00,  2.65it/s]
Episode: 9600, Average Return: 122.68
Iteration 96: 100%|████████████████████████████████████████| 100/100 [00:29<00:00,  3.38it/s]
Episode: 9700, Average Return: 154.82
Iteration 97: 100%|████████████████████████████████████████| 100/100 [00:38<00:00,  2.59it/s]
Episode: 9800, Average Return: 156.63
Iteration 98: 100%|████████████████████████████████████████| 100/100 [00:36<00:00,  2.72it/s]
Episode: 9900, Average Return: 122.02
Iteration 99: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.03it/s]
Episode: 10000, Average Return: 135.35
Iteration 100: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.04it/s]
Episode: 10100, Average Return: 152.33
Iteration 101: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.02it/s]
Episode: 10200, Average Return: 147.81
Iteration 102: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.06it/s]
Episode: 10300, Average Return: 143.99
Iteration 103: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.04it/s]
Episode: 10400, Average Return: 145.90
Iteration 104: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.99it/s]
Episode: 10500, Average Return: 147.36
Iteration 105: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.97it/s]
Episode: 10600, Average Return: 145.75
Iteration 106: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.99it/s]
Episode: 10700, Average Return: 136.30
Iteration 107: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.04it/s]
Episode: 10800, Average Return: 148.58
Iteration 108: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.99it/s]
Episode: 10900, Average Return: 144.44
```

```
Iteration 109: 100%|████████████████████████████████████████| 100/100 [00:47<00:00,  2.12it/s]
Episode: 11000, Average Return: 148.75
Iteration 110: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.01it/s]
Episode: 11100, Average Return: 146.15
Iteration 111: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.96it/s]
Episode: 11200, Average Return: 142.39
Iteration 112: 100%|████████████████████████████████████████| 100/100 [00:47<00:00,  2.11it/s]
Episode: 11300, Average Return: 146.88
Iteration 113: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.06it/s]
Episode: 11400, Average Return: 146.43
Iteration 114: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.07it/s]
Episode: 11500, Average Return: 142.81
Iteration 115: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.00it/s]
Episode: 11600, Average Return: 147.92
Iteration 116: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.98it/s]
Episode: 11700, Average Return: 141.18
Iteration 117: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  2.00it/s]
Episode: 11800, Average Return: 141.44
Iteration 118: 100%|████████████████████████████████████████| 100/100 [00:47<00:00,  2.09it/s]
Episode: 11900, Average Return: 148.31
Iteration 119: 100%|████████████████████████████████████████| 100/100 [00:46<00:00,  2.17it/s]
Episode: 12000, Average Return: 149.57
Iteration 120: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.99it/s]
Episode: 12100, Average Return: 144.12
Iteration 121: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.99it/s]
Episode: 12200, Average Return: 143.86
Iteration 122: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.05it/s]
Episode: 12300, Average Return: 143.22
Iteration 123: 100%|████████████████████████████████████████| 100/100 [00:45<00:00,  2.21it/s]
Episode: 12400, Average Return: 145.15
Iteration 124: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.05it/s]
Episode: 12500, Average Return: 141.00
Iteration 125: 100%|████████████████████████████████████████| 100/100 [00:47<00:00,  2.12it/s]
Episode: 12600, Average Return: 142.18
Iteration 126: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.03it/s]
Episode: 12700, Average Return: 141.88
Iteration 127: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.99it/s]
Episode: 12800, Average Return: 141.60
```

```
Iteration 128: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.06it/s]
Episode: 12900, Average Return: 146.47
Iteration 129: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.99it/s]
Episode: 13000, Average Return: 155.89
Iteration 130: 100%|████████████████████████████████████████| 100/100 [00:48<00:00,  2.06it/s]
Episode: 13100, Average Return: 142.04
Iteration 131: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.02it/s]
Episode: 13200, Average Return: 150.60
Iteration 132: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.97it/s]
Episode: 13300, Average Return: 148.87
Iteration 133: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.96it/s]
Episode: 13400, Average Return: 143.17
Iteration 134: 100%|████████████████████████████████████████| 100/100 [00:51<00:00,  1.95it/s]
Episode: 13500, Average Return: 145.71
Iteration 135: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.01it/s]
Episode: 13600, Average Return: 149.98
Iteration 136: 100%|████████████████████████████████████████| 100/100 [00:31<00:00,  3.20it/s]
Episode: 13700, Average Return: 173.48
Iteration 137: 100%|████████████████████████████████████████| 100/100 [00:27<00:00,  3.60it/s]
Episode: 13800, Average Return: 162.73
Iteration 138: 100%|████████████████████████████████████████| 100/100 [00:36<00:00,  2.74it/s]
Episode: 13900, Average Return: 179.76
Iteration 139: 100%|████████████████████████████████████████| 100/100 [00:33<00:00,  3.02it/s]
Episode: 14000, Average Return: 197.27
Iteration 140: 100%|████████████████████████████████████████| 100/100 [00:35<00:00,  2.82it/s]
Episode: 14100, Average Return: 182.34
Iteration 141: 100%|████████████████████████████████████████| 100/100 [00:40<00:00,  2.45it/s]
Episode: 14200, Average Return: 161.05
Iteration 142: 100%|████████████████████████████████████████| 100/100 [00:45<00:00,  2.19it/s]
Episode: 14300, Average Return: 161.92
Iteration 143: 100%|████████████████████████████████████████| 100/100 [00:47<00:00,  2.11it/s]
Episode: 14400, Average Return: 155.32
Iteration 144: 100%|████████████████████████████████████████| 100/100 [00:46<00:00,  2.13it/s]
Episode: 14500, Average Return: 152.15
Iteration 145: 100%|████████████████████████████████████████| 100/100 [00:47<00:00,  2.11it/s]
Episode: 14600, Average Return: 147.94
Iteration 146: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.98it/s]
Episode: 14700, Average Return: 145.99
```

```
Iteration 147: 100%|████████████████████████████████████████| 100/100 [00:49<00:00,  2.00it/s]
Episode: 14800, Average Return: 143.64
Iteration 148: 100%|████████████████████████████████████████| 100/100 [00:50<00:00,  1.99it/s]
Episode: 14900, Average Return: 147.03
Iteration 149: 100%|████████████████████████████████████████| 100/100 [00:51<00:00,  1.96it/s]
Episode: 15000, Average Return: 144.24
```