# MOON LANDER PROBLEM

**Aim:** To demonstrate the capability of reinforcement learning, particularly the Advantage Actor-Critic (A2C) algorithm, in autonomously landing a moon lander with precision and reliability, showcasing its potential for real-world applications in space exploration and robotic control.

**Description**

The Advantage Actor-Critic (A2C) algorithm is a reinforcement learning technique that combines elements of policy-based and value-based methods. It involves two main components:

**Actor:** The actor network learns a policy directly mapping states to actions. This network is responsible for selecting actions that maximize expected rewards based on the current state of the environment.

**Critic:** The critic network evaluates state-action pairs by estimating the value or expected return associated with taking a specific action in a given state. The critic's role is to provide feedback to the actor by assessing the quality of chosen actions.

In A2C, the actor and critic networks are trained simultaneously. The actor's policy is updated based on the advantages calculated from the critic's evaluations, allowing the agent to learn effective decision-making strategies in continuous action spaces. This approach enables efficient learning by leveraging both the policy gradient from the actor and the value estimation from the critic.

**Source code:**  
# Import necessary libraries

from stable\_baselines3 import A2C

import gym

# Define the environment

environment\_name = 'LunarLander-v2'

env = gym.make(environment\_name)

# Set parameters

episodes = 40

# Main loop for training

for episode in range(episodes):

state = env.reset()

score = 0

while True:

# Render environment (optional)

env.render()

# Choose action based on current state

action = env.action\_space.sample() # Example: randomly sample action space

next\_state, reward, done, info = env.step(action)

# Update score

score += reward

# Print episode score

print(f"Episode: {episode + 1}, Score: {score}")

if done:

break

# Initialize A2C model

model = A2C("MlpPolicy", env, verbose=1)

# Train the model

model.learn(total\_timesteps=100000)

# Result

print("Successfully Implemented the A2C algorithm for the Moon Lander task.")  
  
**Result:**

successfully implemented moon lander using A2C