**PGM2 : CART-POLE BALANCING USING A RANDOM POLICY**

Aim : To implement a RL agent using a random policy to play a game in an environment with a continuous state space namely the Cart-Pole balancing environment.

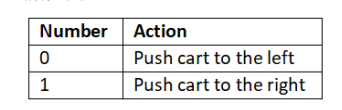
**Model Dynamics:**

1. **State Space:**

Each state is defined using four continuous variables namely

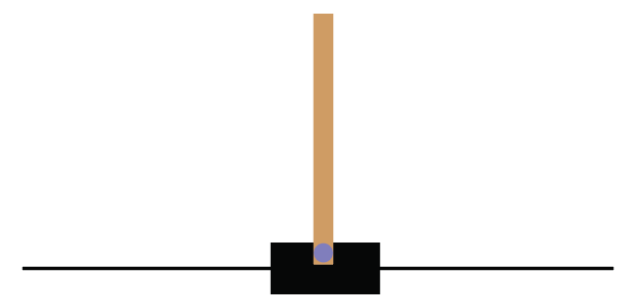
cart position, cart velocity, pole angle and pole velocity at the tip.

1. The value of the **cart position** ranges from -4.8 to 4.8.
2. The value of the **cart velocity** ranges from -Inf to Inf ( −∞ to ∞ ).
3. The value of the **pole angle** ranges from -0.418 radians to 0.418 radians.
4. The value of the **pole velocity** at the tip ranges from -Inf to Inf.
5. **Action Space :**



1. **Reward:** A +1 reward for every move of the agent with a threshold of 500.
2. **Initial state** : a random value for the 4 continuous variables (cart position, cart velocity, pole angle, pole velocity) in the range of -0.05 to +0.05
3. **Terminal state** : An episode ends when one of the following condition is true
4. Pole Angle is greater than ±12°
5. Cart Position is greater than ±2.4 (center of the cart reaches the edge of the display)
6. Episode length is greater than 500 (200 for v0)
7. **Goal of the agent:**

To keep the pole stand straight on the cart



import gymnasium as gym

#from gymnasium.wrappers.monitoring import video\_recorder

#from gymnasium.wrappers.monitoring.video\_recorder import VideoRecorder

env = gym.make("CartPole-v1", render\_mode = "human")

env.reset()

env.render()

#print state space which is continuous

print(env.observation\_space)

#print action space

print(env.action\_space)

env.reset()

#implement with a random policy and print the return after every 10th

#episode

n\_episodes = 50

n\_timesteps = 50

for i in range(n\_episodes):

Return = 0

for t in range(n\_timesteps):

env.render()

rnd\_action = env.action\_space.sample()

next\_state, reward, done, infor, prob = env.step(rnd\_action)

Return = Return + reward

if done:

env.reset()

break

if i%10 == 0:

print("Episode : {}, Return : {}".format(i+1, Return))

env.close()