```python
import numpy as np

def epsilon_greedy_policy_generator(action_low, action_high_plus):

    def epsilon_greedy_policy(state, model, epsilon = 0):
        '''
        Simple policy which either gets the next action or random action with chance epsilon in cartpole
        '''

        if np.random.rand() < epsilon:
            return np.random.randint(action_low, high=action_high_plus)
        else:
            Q_values = model.predict(state[np.newaxis])
            return np.argmax(Q_values[0])

    return epsilon_greedy_policy

def random_policy(*args):
    '''
    Randomly chooses an action in cartpole
    '''

    return np.random.randint(2)

def epsilon_episode_decay(initial_epsilon, min_epsilon, rate=500):
    '''
    Linearly decays epsilon
    '''

    # Decay epsilon to some min value according to episode
    return lambda episode: max(initial_epsilon - episode / rate, min_epsilon)

def acrobot_epsilon_decay(initial_epsilon, dropoff):

    def func(episode):

        if episode >= dropoff:
            return .01
        else:
            return initial_epsilon

    return func
```