Algorithm 1: Deep Deterministic Policy Gradient (DDPG)

Input: Initial actor parameters θ, critic parameters ϕ, target actor parameters $\bar{\theta} \leftarrow \theta$, target critic parameters $\bar{\phi} \leftarrow \phi$, polyak averaging coefficient τ, empty replay buffer $\mathcal{D}$, action bounds $a_{\min}$ and $a_{\max}$

for *each iteration* **do**
- Reset environment and observe initial state s_0;
- Reset Ornstein-Uhlenbeck noise process $\mathcal{N}_t$;
- **while** *not terminal* **do**
  - Select action with exploration: $a_t = \pi_\theta(s_t) + \mathcal{N}_t$;
  - Clip action: $a_t \leftarrow \mathrm{clip}(a_t, -c, c)$;
  - Execute a_t in environment;
  - Observe reward r_t, next state s_{t+1}, and terminal signal d_t;
  - Store $(s_t, a_t, r_t, s_{t+1}, d_t)$ in replay buffer $\mathcal{D}$;
  - **if** d_t *is True* **then**
    - Reset environment and observe new initial state s_{t+1};
  - **end**
  - $s_t \leftarrow s_{t+1}$;
- **end**
- **for** *each gradient step* **do**
  - Sample a mini-batch of N transitions (s, a, r, s', d) from $\mathcal{D}$;
  - Compute target Q-value:
    $$y = r + \gamma(1-d)Q_{\bar{\phi}}(s', \pi_{\bar{\theta}}(s'))$$
  - Update critic by gradient descent:
    $$\phi \leftarrow \phi - \lambda_Q \nabla_\phi \frac{1}{N} \sum \left(Q_\phi(s,a) - y\right)^2$$
  - Update actor by gradient ascent:
    $$\theta \leftarrow \theta + \lambda_\pi \nabla_\theta \frac{1}{N} \sum Q_\phi(s, \pi_\theta(s))$$
  - Update target networks:
    $$\bar{\phi} \leftarrow \tau\phi + (1-\tau)\bar{\phi}, \quad \bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$$
- **end**

end