Algorithm 1: Twin Delayed Deep Deterministic Policy Gradient (TD3)

Input: Initial actor parameters θ, critic parameters ϕ_1, ϕ_2, target actor parameters $\bar{\theta} \leftarrow \theta$, target critic parameters $\bar{\phi}_1 \leftarrow \phi_1$, $\bar{\phi}_2 \leftarrow \phi_2$, polyak averaging coefficient τ, policy delay K, empty replay buffer $\mathcal{D}$

for *each iteration* **do**
  Reset environment and observe initial state s_0;
  while *not terminal* **do**
    Select action with exploration: $a_t = \pi_\theta(s_t) + \mathcal{N}_t$;
    Clip action: $a_t \leftarrow \text{clip}(a_t, -c, c)$;
    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 $t = 1$ **to** *number of gradient steps* **do**
    Sample a mini-batch of N transitions (s, a, r, s', d) from $\mathcal{D}$;
    Add clipped noise to target action:

$$\tilde{a}' = \pi_{\bar{\theta}}(s') + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c)$$

    Compute target Q-value:

$$y = r + \gamma(1 - d) \min_{i=1,2} Q_{\bar{\phi}_i}(s', \tilde{a}')$$

    Update critics by gradient descent:

$$\phi_i \leftarrow \phi_i - \lambda_Q \nabla_{\phi_i} \frac{1}{N} \sum \left(Q_{\phi_i}(s, a) - y\right)^2, \quad i = 1, 2$$

    if $t \bmod K = 0$ **then**
      Update actor by gradient ascent:

$$\theta \leftarrow \theta + \lambda_\pi \nabla_\theta \frac{1}{N} \sum Q_{\phi_1}(s, \pi_\theta(s))$$

      Update target networks:

$$\bar{\phi}_i \leftarrow \tau\phi_i + (1 - \tau)\bar{\phi}_i, \quad \bar{\theta} \leftarrow \tau\theta + (1 - \tau)\bar{\theta}$$

    end
  end
end