```
Initialize learning rate $\alpha$
Initialize $\tau$
Initialize number of batches per training step, $B$
Initialize number of updates per batch, $U$
Initialize batch size $N$
Initialize experience replay memory with max size $K$
Initialize target network update frequency $F$
Randomly initialize the network parameters $\theta$
Initialize the target network parameters $\varphi = \theta$
for $m = 1 \ldots MAX\_STEPS$ do
    Gather and store $h$ experiences $(s_i, a_i, r_i, s'_i)$ using the current policy
    for $b = 1 \ldots B$ do
        Sample a batch, $b$, of experiences from the experience replay memory
        for $u = 1 \ldots U$ do
            for $i = 1 \ldots N$ do
                # Calculate target $Q$-values for each example
                $y_i = r_i + \delta_{s'_i} \gamma Q^{\pi_\varphi}(s'_i, \max_{a'_i} Q^{\pi_\theta}(s'_i, a'_i))$ where $\delta_{s'_i} = 0$
                $\hookrightarrow$    if $s'_i$ is terminal, 1 otherwise
            end for
            # Calculate the loss, for example using MSE
            $L(\theta) = \frac{1}{N} \sum_i (y_i - Q^{\pi_\theta}(s_i, a_i))^2$
            # Update the network's parameters
            $\theta = \theta - \alpha \nabla_\theta L(\theta)$
        end for
    end for
    Decay $\tau$
    if $(m \mod F) == 0$ then
        # Update the target network
        $\varphi = \theta$
    end if
end for
```