Page 3, Tabular Q-Learning

1. Start with empty table mapping states to values of actions,
2. By interacting with the environment, obtain tuple (s, a, r, s').
3. Update Q(s, a) value using Bellman approximation: $Q(s,a) \leftarrow r + \gamma \max_{a' \in A} Q(s', a')$
4. Repeat step 2.

...using learning rate α with value from 0 to 1:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a' \in A} Q(s', a'))$$

The final version of the algorithm is here:

1. Start with empty table for Q(s, a).
2. Obtain (s, a, r, s') from the environment.
3. Make Bellman update: $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a' \in A} Q(s', a'))$
4. Check convergence conditions, if not met, repeat from step 2.

Page 8, Deep Q-learning
With this in mind, let's make modifications to the Q-learning algorithm:

1. Initialize Q(s, a) with some initial approximation,
2. By interacting with the environment, obtain tuple (s, a, r, s').
3. Calculate loss: $\mathcal{L} = (Q(s,a) - r)^2$ if episode has ended or $\mathcal{L} = (Q(s,a) - (r + \gamma \max_{a' \in A} Q_{s',a'}))^2$ otherwise.
4. Update Q(s, a) using SGD algorithm by minimizing the loss in respect to model parameters.
5. Repeat step 2 until converged.

Page 11, Final form of DQN training

1. Initialize parameters for Q(s, a) and $\hat{Q}(s,a)$ with random weights, $\epsilon \leftarrow 1.0$, and empty replay buffer
2. With probability ϵ select a random action a, otherwise $a = \arg\max_a Q(s,a)$
3. Execute action a in emulator and observe reward r and next state s'.
4. Store transition (s, a, r, s') in the replay buffer.
5. Sample random minibatch of transitions from replay buffer.

6. For every transition in the buffer calculate target $y = r$ if episode has ended at this step or $y = r + \gamma \max_{a' \in A} \hat{Q}(s', a')$
7. Calculate loss: $\mathcal{L} = (Q(s, a) - y)^2$
8. Update Q(s, a) using SGD algorithm by minimizing the loss in respect to model parameters.
9. Every N steps copy weights from Q to $\hat{Q}$
10. Repeat step 2 until converged.

Page 24

As a reminder, there is the loss expression we need to calculate: $\mathcal{L} = (Q(s, a) - (r + \gamma \max_{a' \in A} \hat{Q}(s', a')))^2$ for steps which wasn't at the end of the episode or $\mathcal{L} = (Q(s, a) - r)^2$ for final steps.