3.6) ~~Reward~~ for a continuous task $(R) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \cdots$
Return

" " " episodic task $(R) = r_{t+1} + r_{t+2} + r_{t+3} + \cdots$

In this case since we also introduce discounts in the episodic task, the ~~reward~~ return will be the same as a continuous task:

$$R = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots$$

Return at each time will be 0, except for a failure where it will be $= -\gamma^{k-1}$

3.7) No, we have not introduced any negative reward on taking a wrong move, meaning a short path (5 grids) and a longer path (8 grids) will give it the same reward (+1) on escape, so the agent doesn't learn anything.
By introducing negative reward (say -1), the rewards will be -4 and -7 respectively instead, according to my example. So it receives negative reinforcement, such that it learns from the policy / path it took to escape, and can learn and differentiate between good and better paths.

3.14) Figure 3.2

Bellman equation:-

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right], \text{ for all } s \in S$$

Here, given :-

$p(s',r|s,a) = 1$ (transition probability)

$\gamma = 0.9$

$r = 0$

$\pi(a|s) = 1/4 = 0.25$ (4 choices)

Therefore:-

$$v_\pi(0.7) = 0.25 \times 1 (0 + (0.9 \times 0.23))$$
$$+ 0.25 \times 1 (0 + (0.9 \times 0.4))$$
$$+ 0.25 \times 1 (0 + (0.9 \times -0.4))$$
$$+ 0.25 + 1 (0 + (0.9 \times 0.7))$$

$$= 0.25 (2.07 + 0.36 - 0.36 + 0.63)$$

$$= 0.675 \approx 0.7$$

3.22) Bellman equation:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\left[r + \gamma v_\pi(s')\right]$$

In this case:

$\pi(a|s) = 0.5$ (two choices)

$p(s', r|s, a) = 1$

for $\gamma = 0$:

~~$v_\pi(s) = 0.5 \times 1$~~ (

$v^*_\pi(s) = \max(q_L, q_R)$, for state s, L = left policy, R = right policy

where:

$$q_L = \sum p(s', r|s, a)\left[0 + \gamma v_\pi(s')\right]$$
$$= 1\left[1 + 0 \times 0\right]$$
$$= 1$$

$$q_R = 1\left[0 + 0 \times 2\right]$$
$$= 0$$

$\therefore v^*(s)$ for $\gamma = 0$ $= q_L$

Similarly for $\gamma = 0.9$ & $\gamma = 0.5$:-

γ	Q_L	Q_R	
0	(1)	0	→ Optimal policy = left path
0.9	1	(1.8)	→ Optimal policy = right path
0.5	(1)	(1)	→ Optimal policy = either path is fine

3.23) Bellman equations for Q^* :-

given 2 states → low, high
and 3 actions → search, wait, recharge

$$Q^*(\text{high}, \text{search}) = \alpha(R_{\text{search}} + \gamma V^*(\text{high})) + (1-\alpha)(R_{\text{search}} + \gamma V^*(\text{low})) \quad -(1)$$

$$Q^*(\text{high}, \text{wait}) = R_{\text{wait}} + \gamma V^*(\text{high}) \quad -(2)$$

$$Q^*(\text{low}, \text{search}) = \beta(R_{\text{search}} + \gamma V^*(\text{low})) + (1-\beta)(-3 + \gamma V^*(\text{high})) \quad -(3)$$

$$Q^*(\text{low}, \text{recharge}) = \gamma V^*(\text{high}) \quad -(4)$$

$$Q^*(\text{low}, \text{wait}) = R_{\text{wait}} + \gamma V^*(\text{low}) \quad -(5)$$

3.25) V^* in terms of Q^* :-

$V^*(\text{high}) = \max(Q^*(\text{high}, \text{search}), Q^*(\text{high}, \text{wait}))$.

$V^*(\text{low}) = \max(Q^*(\text{low}, \text{search}), Q^*(\text{low}, \text{wait}),$
$Q^*(\text{low}, \text{recharge}))$.