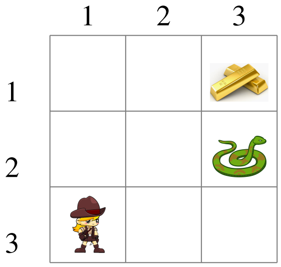
Exercise 1 :

Consider the following variation of the example from the lecture:



The only difference is that the reward is 1 point for the gold, -20 points for the snake, and 0 everywhere else.

(i) Compute the first 3 iterations of value iteration with $\gamma = 0.9$

(ii) Consider the value function after the 3th iteration. Does the optimal policy change with respect to the version in the lecture? Justify your answer.

Solution:

Iteration 0:

	1	2	3
1	0	0	0
2	0	0	0
3	0	0	0

Iteration 1:

	1	2	3
1	0	0	1
2	-0	0	-20
3	0	0	0

Iteration 2:

	1	2	3
1	0	0.9	1
2	0	-3.6	-22.88
3	0	0	0

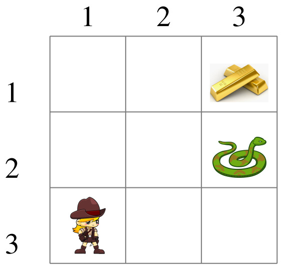
Iteration 3:

	1	2	3
1	0.81	0.9	1
2	-0.65	-3.47	-23.40
3	0	0	0

The optimal policy will always remain in the bottom row and never attempt to get the gold. This is different as in the lecture, where the optimal policy was to get the gold going as far from the snake as possible. This makes sense, as here, the adventurer is more scared of the snake so it won't risk her life to get the gold.

Exercise 2 :

Consider the following variation of the example from the lecture:



But now the snake will move to an adjacent square every time. The following rules apply:

- If the adventurer is where the snake is, the snake will remain there. In that case, the adventurer cannot move with probability 0.5, else it moves where they wanted to.
- If the adventurer is next to the snake, then the snake will jump to the adventurer with probability 0.5 (and the adventurer cannot move) or remain where it is.

- If the adventurer is not next to the snake, then the snake will move at random (0.25 percent in each direction).
- As in the original problem, when the adventurer moves there is a small chance (20%) that she falls to the right.

Getting the gold provides the adventurer 10 points. Being in the same square as the snake provides -5 points. Everything else provides -0.1 points. Formalize this problem as an MDP. Provide a formal description of the following.

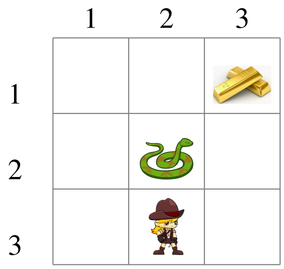
(i) Set of states

(ii) Actions

(iii) Reward function

(iv) Transition probability function. In particular, describe the possible outcomes for the initial state where we apply the north action.

(v) Transition probability function. In particular, describe the possible outcomes for the following state where we apply the east action.



Solution:

(i) We one state for each position of the adventurer (x_a, y_a) and each position of the snake (x_s, y_s). We describe each state as a 4-tuple (x_a, y_a, x_s, y_s). Therefore, the set of states contains 81 elements: $\{(1, 1, 1, 1), (1, 1, 1, 2), \dots\}$

(ii) The actions are as in the original problem: $\{N, E, W, S\}$

(iii) If $(x_a, y_a) = (3, 1)$ then $R(s) = 10$. Otherwise, if $(x_a, y_a) = (x_s, y_s)$, then $R(s) = -5$. Otherwise $R(s) = -0.1$

(iv) The possible outcomes are:

- $(2, 3, 3, 1)$, with probability $0.2 \cdot 0.25 = 0.05$
- $(2, 3, 3, 2)$, with probability $0.2 \cdot 0.5 = 0.1$
- $(2, 3, 3, 3)$, with probability $0.2 \cdot 0.25 = 0.05$

- $(1, 2, 3, 1)$, with probability $0.8 \cdot 0.25 = 0.05$
- $(1, 2, 3, 2)$, with probability $0.8 \cdot 0.5 = 0.1$
- $(1, 2, 3, 3)$, with probability $0.8 \cdot 0.25 = 0.05$

(v) The possible outcomes are:

- $(2, 3, 2, 3)$, with probability 0.5
- $(3, 3, 2, 2)$, with probability 0.5

Exercise 3 :

Consider the game of Tetris: https://en.wikipedia.org/wiki/Tetris_(NES_video_game).

Formalize this problem as an MDP. Provide a description of the following.

(i) Set of states

(ii) Actions

(iii) Reward function

(iv) Transition probability function

(v) Initial state

(vi) Set of terminal states

The description can be just a short text in English textual and not necessarily entirely formal.

Solution:

(i) We can describe a state in terms of the following variables:

- For each cell, a boolean variable indicating whether there is a block there or not.
- Which piece is falling
- The position of the piece that is falling

We have a state for each combination of the values of these variables.

Optionally, one can also model the "next-piece", as well as the number of lines.

(ii) There are nine actions: $\{\mathsf{moveleft}, \mathsf{don'tmove}, \mathsf{moveright}\} \times \{\mathsf{rotateleft}, \mathsf{don'trotate}, \mathsf{rotateright}\}$.

Optionally, one could also model push-down.

(iii) The reward function corresponds to the score in the game. Every time lines are cleared, the player achieves the corresponding reward.

(iv) The transition function is almost always deterministic. The state can be computed following the rules of the game, where the falling piece is moved according to the player's action. The transition funciton is not deterministic when deciding the next piece, which is chosen uniformly at random.

(v) In the initial state the board is clear and no piece has been selected yet.

(vi) The terminal states are those where the player “dies”, i.e., the next piece does not have any space to be put into. Optionally, one could put a line cap too.