# Assignment 1 - Wumpus World

**Group:** Tomass Lacis (tola18), Alexandr Silonosov (alsi18)

The group decided to solve Wumpus World assignment using reinforcement learning approach (Grade B). This was done using “Q-Learning” algorithm (Wikipedia gives a good brief summary about the algorithm: https://en.wikipedia.org/wiki/Q-learning ).

For Q-learning to work you have to choose and tune viable actions, state structure, reward function and training process. We used following configuration for our QStates and training method:

- **Actions (7 in total):** MOVE_UP, MOVE_RIGHT, MOVE_DOWN, MOVE_LEFT, SHOOT, GRAB, CLIMB
    - **Additional functionality:** we have hardcoded conditions when each listed action is available to use in a given state to ensure proper behavior
- **State structure:** Special flags, player X/Y, 16 tile data
    - **Special flags:** Safe exploration toggle, is on gold, is on wumpus, is in pit, has arrow
        - Safe exploration on by default, which means that agent is not allowed to take unnecessary risks, when it is off, it is free to dive into potential pits (but still avoid wumpus)
    - **Player X,Y**: player location X,Y coordinates
    - **16 tile data:** represents 4x4 tile world state
- **Reward function:** Normal actions without consequences give 0 reward, while exploring unexplored tiles give 1 and grabbing gold gives 10. These values were found through trial and error
- **Training method:** We use epsilon-greedy Q-learning training where we generally try to do random actions (epsilon-large probability to do random action) at the start and note their rewards, which are used to calculate Q-values. Gradually we lower epsilon to make agent use QTable knowledge more for best actions rather than random actions

**Notes:**

- Reinforcement learning part is actually navigating through the map with at-the-time-available actions
- Overfitting: the assignment task is to solve 7 pre-made maps which was done, however, the agent will perform poorly on maps with previously unseen states
- Percepts will not be ignored due to action availability logic
- Future improvement can be using neural networks to create function approximization for Q-learning, therefore, Deep-Q-Learning algorithm (https://arxiv.org/pdf/1312.5602v1.pdf). This would allow better training results from random maps