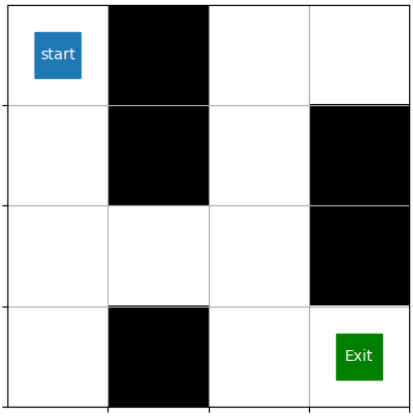
Tutorial 4 - Monte Carlo Methods

Exercise N°1:

We consider the problem of solving a Maze. The agent begins at the start cell and must find the exit cell by moving through the maze

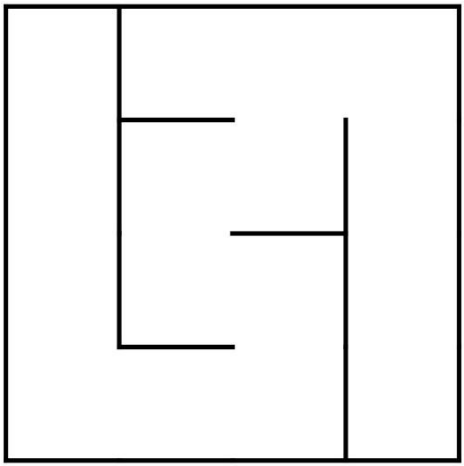
- The maze is represented as a 2D grid with three types of cells: blocked cells, free cells, and exit cell.
- To get to the exit, the agent moves through the maze in a series of steps. The are exactly 4 actions encoded as integers 0-3: [0 - up, 1 - down, 2 - left, 3 - right]
    - The possible actions deterministically cause the agent to move one cell in the respective directions
    - If the agent decides to take a non-permitted action (moving to blocked cells or moving outside the maze), it stays in the same state
- The reward function is defined as follow:
    - The agent receives a reward of (+50) for reaching the exit cell
    - The agent receives a penalty of (-10) for colliding with a wall (i.e. for trying to enter a blocked cell or moving out of the maze)
    - A penalty of (-1) is applied for a move which did not result in finding the exit cell
- The figure below shows an example of a maze environment
    - The start state is located in the (0, 0) cell (blue cell).
    - The exit state is located in the (3, 3) cell (green cell).
    - Black color indicates blocked cells.
    - White color indicates a free cell.
    - A cell is identified by (index_row, index_col) and takes the value of “0” if it is free and “1” if it is blocked.
    - The agent wins if it reaches the exit cell and fails if it exceeds the time limit.



1. Formalize the problem as a Markov decision process (MDP).
2. Implement the reward () function that returns the scalar value the agent receives after performing a given action
3. Implement the next_state() function that returns the next state after the agent performs an action
4. Implement **step()** function that runs one time-step of the environment's dynamics using the agent's actions. The function returns the following: the next state, the reward, and whether the episode has ended (due to reaching either the exit cell or the time limit.
5. We consider the policy $\pi 1$ defined as equidistributed directions with a probability of 1/4 for all states. Apply the **First-Visit Monte Carlo** prediction algorithm to evaluate $\pi 1$.
6. Apply the Monte Carlo Control methods covered in the course.
    a. Monte Carlo Exploring Starts
    b. On-policy first-visit MC control
    c. Off-policy MC control
7. What is the main difference between MC algorithms and the DP algorithms?

Exercise N°2 (Additional):

We consider the problem of solving a maze with a different representation of the state of environment (cell of the maze). In this new presentation, each cell may be surrounded by walls to the north, east, south, or west. The figure below provides an example of a maze with this new representation.



1. Formalize the problem as a Markov decision process (MDP).
2. Adapt the **reward(), next_state(),** and **step()** functions from the first exercise to work with the new maze representation
3. We consider the policy $\pi 1$ defined as equidistributed directions with a probability of $1/4$ for all states. Apply the **First-Visit Monte Carlo** prediction algorithm to evaluate $\pi 1$.
4. Apply the Monte Carlo Control methods covered in the course.
    a. Monte Carlo Exploring Starts
    b. On-policy first-visit MC control
    c. Off-policy MC control