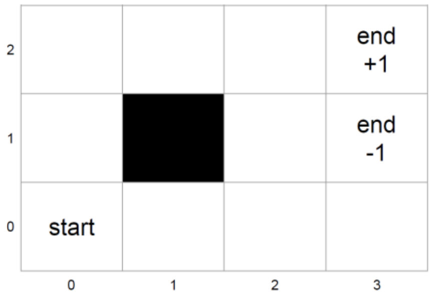
Workshop Lecture 2

Part A: A simple maze environment

In the Blackboard folder for this week, you will see the python script “`q-learning.py`” under the workshop materials. Save locally, and run with (in terminal, `cd` to location of script):

```
python lecture2-simulation.py
```

This script defines a simple maze (image below), and a simple agent, with the task of learning how to get from the start to the end goal. Take a few minutes to understand the structure of the code (link to base code provided in script). You will notice that there are two ‘agents’ implemented: `Agent` instantiates a standard q-learning algorithm, and `IRLAgent` implements a basic interactive reinforcement learning variant.



Part B: Q-learning Agent

At the bottom of the script, in `main`, you will see an `Agent` instantiated, and 40 completions of the maze (this can be changed). Run this simulation and see what the resultant state values are (shown in console). Vary parameters (leave the environment size unchanged) and explore the impact.

Part C: IRL Agent

To enable the `IRLAgent`, uncomment the relevant line in `main`, and run the simulation again. For this agent, you will be asked for user input on every action taken by the `IRLAgent`. Provide this reward (this is very simplified – see source).

Compare the operation of `Agent` with `IRLAgent`, both in terms of how they work, and in terms of the results they come up with (in terms of the state values). What are the advantages/disadvantages of each approach (also wrt much larger problems)?