

Assignment 3: Wumpus Quest

AI-2 Systems Project (Summer Semester 2026)

Jan Frederik Schaefer

Friedrich-Alexander-Universität Erlangen-Nürnberg, Department Informatik

Topic: Markov Decision Processes

Due on: August 5, 2026

Version from: June 9, 2026

Author: Jan Frederik Schaefer

Make sure you sign up before working on this assignment.^a

Using someone else's solution code, even as inspiration, is not allowed!

Sharing your solution code with other AISysProj students is not allowed.

^aYou can still decide to postpone the assignment. Signing up includes an eligibility check, which avoids situations where you invest work into an assignment that you are not supposed to take.

1 Task summary

Your goal is to collect gold in the Wumpus cave and bring it back safely. There are many dangers in the cave, but, fortunately, you have a map that can help you decide on the best course of action. You can test and evaluate your implementation on the server¹.

Didactic objectives

1. Gain experience modelling a complex problem as a Markov Decision Process.

Prerequisites and useful methods

1. Basics of working with probabilities,
2. Markov Decision Processes (and value iteration) as discussed in the lecture.

2 Detailed problem description

You have a 14×12 map of the Wumpus cave (Figure 1 shows an example). Your goal is to guide the agent to collect the gold and leave the cave. Depending on the environment,

¹<https://aisysproj.kwarc.info>

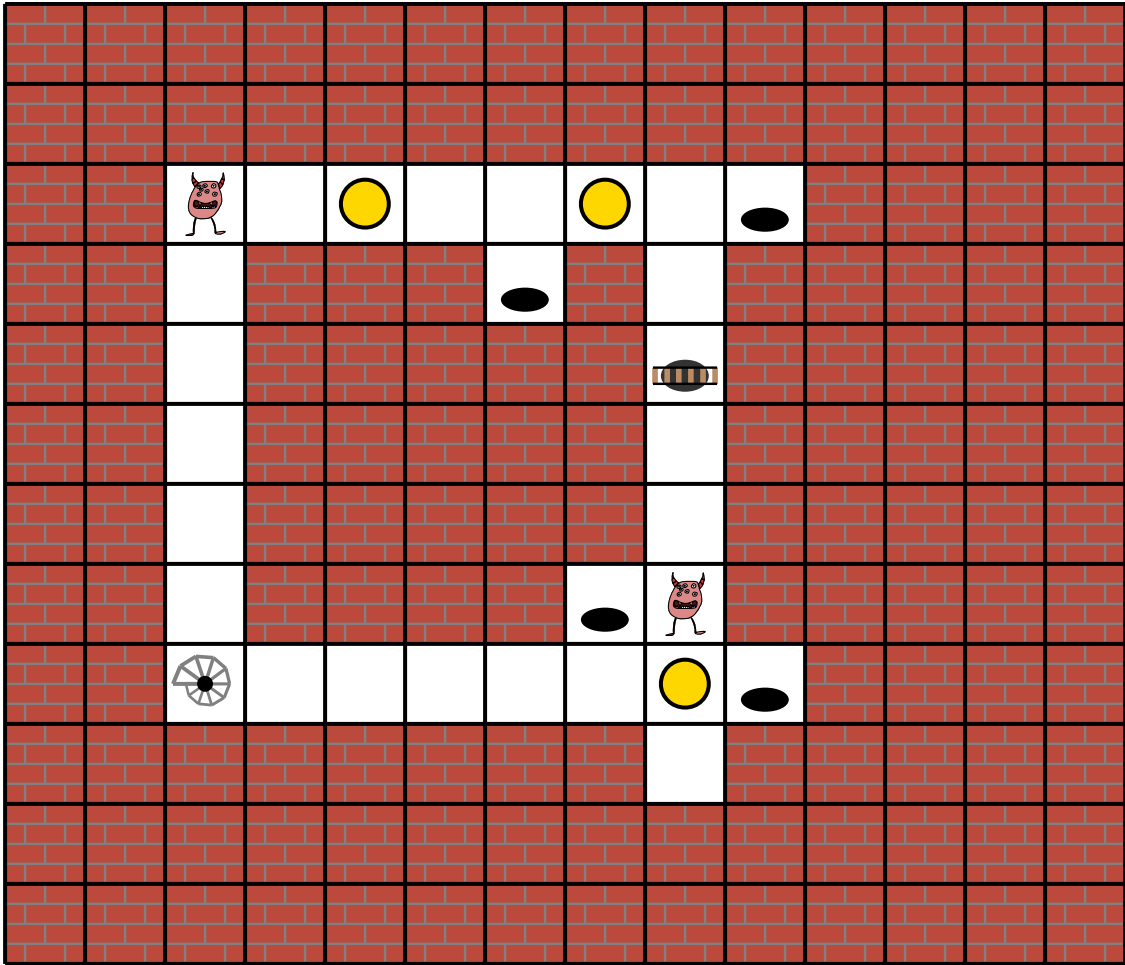


Figure 1: The Wumpus cave.

there are different obstacles that might kill your agent. Before entering the cave, your agent can train, which means that you can allocate a certain number of skill points, which will e.g. improve its chances when fighting the Wumpus. Your agent only has enough time for 100 actions until it runs out of food (in practice, that is plenty of time and mostly serves as a mechanism to stop runs where your agent somehow got stuck).

Your agent will be rated based on how much gold it carries out of the cave (0 if it dies). Note that your agent starts with 1 gold, so, sometimes, the best strategy might be to take no risk and exit the cave right away.

In the following subsections, we will explore what actions your agent can perform and what obstacles it faces in the cave.

2.1 Maps

The map is represented as a string, where each line corresponds to a row of the map and each character describes the properties of a cell. For example, the map shown in Figure 1 would have the following string representation:

```
XXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXX
XXW_G_G_PXXXXX
XX_XXXPX_XXXXX
XX_XXXXXBXXXXX
XX_XXXXX_XXXXX
XX_XXXXX_XXXXX
XX_XXXXPWXXXXX
XXS_G_G_G_GPXXXX
XXXXXXXXX_XXXXX
XXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXX
```

The letters have the following meanings:

- A space: An empty square.
- **S**: The stairs (your agent will start there, and it is the only exit).
- **G**: Gold.
- **P**: A pit.
- **W**: A Wumpus.

- **B**: A bridge.

Section 2.4 will describe some of the obstacles in more detail.

2.2 Skill points

Depending on the environment, you have a certain number of skill points that you can allocate to two different skills: **agility** and **fighting**. For example, if you can distribute 8 skill points, you could allocate 3 points for **agility** and the remaining 5 points for **fighting**. If you need to use a skill, your agent will roll as many dice as it has allocated to the relevant skill. Your agent then gets a **score**, which is the sum of the three best results. If the score is sufficiently high, your agent succeeds – otherwise it fails.

For example, if your agent has to fight the Wumpus, it will need to use the **fighting** skill. With 5 allocated skill points, it would roll 5 dice. Let us say the result is $\{3, 3, 1, 3, 1\}$. Then the three best results are $\{3, 3, 3\}$, which gives us a sum of 16. To win against the Wumpus, the sum has to be at least 13, so, in this case, your agent would win against the Wumpus.

2.3 Actions

Depending on where you are, you can perform one of the following actions:

- **NORTH/EAST/SOUTH/WEST**: Walk one step in the direction (you cannot walk in the direction of a wall, and you also cannot walk away from the Wumpus without fighting – the server would reject such actions). It is dark in the cave, so sometimes the agent walks in the wrong direction. If there is no wall to the left, there is a 10% chance that it will walk there instead. Similarly, there is a 10% chance that it will walk to the right square if possible. With the remaining probability, it will walk in the requested direction. So, if there are walls to the left and right, the agent will definitely walk into the requested direction. Whenever the agent walks to a square with gold, it will automatically collect it.
- **EXIT**: Exit the cave (only possible if you are at the stairs).
- **FIGHT**: If you are on a square with a Wumpus, you have to fight it. To win, you have to use the **fighting** skill and get a score of at least 13.

2.4 Obstacles

Depending on the environment, your agent can encounter the following deadly obstacles:

- **Pit:** If your agent falls into a pit, it will not survive.
- **Wumpus:** If your agent meets the Wumpus, it will have to fight (see Section 2.3). If your agent wins, the Wumpus will not bother you anymore, but if it loses, your agent dies.
- **Bridge:** If your agent crosses a bridge, it will have to use its **agility** skill and get a score of at least 12. If successful, it can continue its adventure. Otherwise, it will fall into the pit below the bridge and not survive.

3 Competing on the server

You should test and evaluate your agent by competing on the server. The rating of your agent is the average amount of gold that your agent carried out of the cave in 1000 consecutive runs. Your agent can communicate with the server via HTTP requests. A Python implementation of the protocol is provided in the assignment repository [AR]. The details of the protocol are described in [CSP], but you will only need them if you want to create your own implementation (e.g. in a different programming language).

3.1 Action requests

The server will send you action requests, which contain an identifier for the request and a JSON object describing your current adventure. Concretely, it has the following fields:

- **"map":** The map of the cave.
- **"history":** The actions you have performed so far and what the outcome was. The "outcome" fields are the most relevant ones as they help you re-construct the current state. You can ignore the "expl" fields.
- **"skill-points":** Your skill point allocation (if you have already allocated them).
- **"free-skill-points":** The number of skill points you can allocate in this environment (if you haven't allocated them already).

Here is an example (apologies for the poor line breaking):

```
{
  "map": "XXXXXXXXXXXXXXXX\nXXXXXXXXXXXXXXXX\nXXW G G PXXXX\nXX XXXPX
  XXXXX\nXX XXXXXBXXXXX\nXX XXXXX XXXXX\nXX XXXXX XXXXX\nXX
  XXXXPWXXXXX\nXXS GPXXXX\nXXXXXXXXXX XXXXX\nXXXXXXXXXXXXXXXX\n
  nXXXXXXXXXXXXXXXX",
```

```

"history": [
  {"action": {"agility": 0,"fighting": 12},"outcome": {"agility": 0,"fighting": 12}},
  {"action": "NORTH","outcome": {"position": [2,7]}},
  {"action": "NORTH","outcome": {"position": [2,6]}},
  {"action": "NORTH","outcome": {"position": [2,5]}},
  {"action": "NORTH","outcome": {"position": [2,4]}},
  {"action": "NORTH","outcome": {"position": [2,3]}},
  {"action": "NORTH","outcome": {"position": [2,2]}},
  {"action": "FIGHT","outcome": {"killed-wumpus-at": [2,2],"expl": "dice: 553415214222"
    }},
  {"action": "EAST","outcome": {"position": [3,2]}},
  {"action": "EAST","outcome": {"position": [4,2],"collected-gold-at": [4,2],"current-
    amount-of-gold": 3}}
],
"skill-points": {
  "agility": 0,
  "fighting": 12
}
}

```

3.2 Sending Actions

Your agent should respond to an action request by sending an action. The first action should be an allocation of skill points in the form of a JSON object (e.g. {"agility": 0, "fighting": 12}).

For the remaining actions, you should send the desired action as a string (Section 2.3 lists the available actions).

4 What to submit

Your solution should be submitted to your team's repository. It should contain:

1. All your code for solving this assignment.
2. A README.md file explaining

- i. dependencies (programming language, version, external libraries and how to get them),
 - ii. how to run your code on different environments,
 - iii. the repository structure,
 - iv. anything else we should know.
3. A solution summary (see [SoS] for more details – it should describe the main ideas, not document the code).

5 Points

You can get up to 80 points for the strength of your agent according to the server (assuming it is reproducible). Assuming you have at least a partial solution, you can additionally get up to 20 points for the quality of the submission (README, explanation, ...). The maximum number of points is therefore 100. If the grading scheme doesn't seem to work well, we might adjust it later on (likely in your favor).

Figure 2 summarizes how many points you can get for each environment according to the rating on the server. When grading, we will only consider the environment in which you would get the most points.

References

- [AR] *Repository for Assignment 3: Wumpus Quest*. URL: <https://gitlab.rrze.fau.de/wrv/AISysProj/ss26/a2.3-wumpus-quest/assignment>.
- [CSP] Jan Frederik Schaefer. *AISysProj server – Clients and server protocol*. URL: <https://aisysprojserver.readthedocs.io/en/latest/clients.html>.
- [SoS] *Solution Summary*. URL: <https://gitlab.rrze.fau.de/wrv/AISysProj/admin/general/-/blob/main/solution-summary.md>.

Config	Obstacles	Skill points	Points
ss26-wquest-1.json		6	$\left\{ \begin{array}{l} 0 \quad \text{if rating} < 2.0 \\ 15.0 \quad \text{if rating} \geq 2.0 \\ 25.0 \quad \text{if rating} \geq 3.0 \end{array} \right.$
ss26-wquest-2.json	B	6	$\left\{ \begin{array}{l} 0 \quad \text{if rating} < 1.3 \\ 35.0 \quad \text{if rating} \geq 1.3 \\ 45.0 \quad \text{if rating} \geq 1.6 \end{array} \right.$
ss26-wquest-3.json	B, P	8	$\left\{ \begin{array}{l} 0 \quad \text{if rating} < 1.3 \\ 45.0 \quad \text{if rating} \geq 1.3 \\ 70.0 \quad \text{if rating} \geq 1.8 \end{array} \right.$
ss26-wquest-4.json	B, P, W	12	$\left\{ \begin{array}{l} 0 \quad \text{if rating} < 1.4 \\ 65.0 \quad \text{if rating} \geq 1.4 \\ 73.0 \quad \text{if rating} \geq 1.7 \\ 80.0 \quad \text{if rating} \geq 1.95 \end{array} \right.$

Figure 2: Number of points that you can get.