ASSIGNMENT 2 – WUMPUS WORLD

The Assignment

- The Wumpus World is a simple maze-like game where the player agent has to take decisions based on conflicting information.
- Your task is to implement a player agent.
- The agent should be able to solve the game on all existing maps as well as on a good number of randomly generated ones.

The Wumpus World

- A cave consisting of rooms connected vertically and horizontally.
- Somewhere in the cave lurks the Wumpus.
- The Wumpus can be killed by the player, but the player only has one arrow.
- Some rooms have bottomless pits.
- Goal is to find the gold treasure!
- Wumpus world is a well-known testbed for logic, first is from 1972.

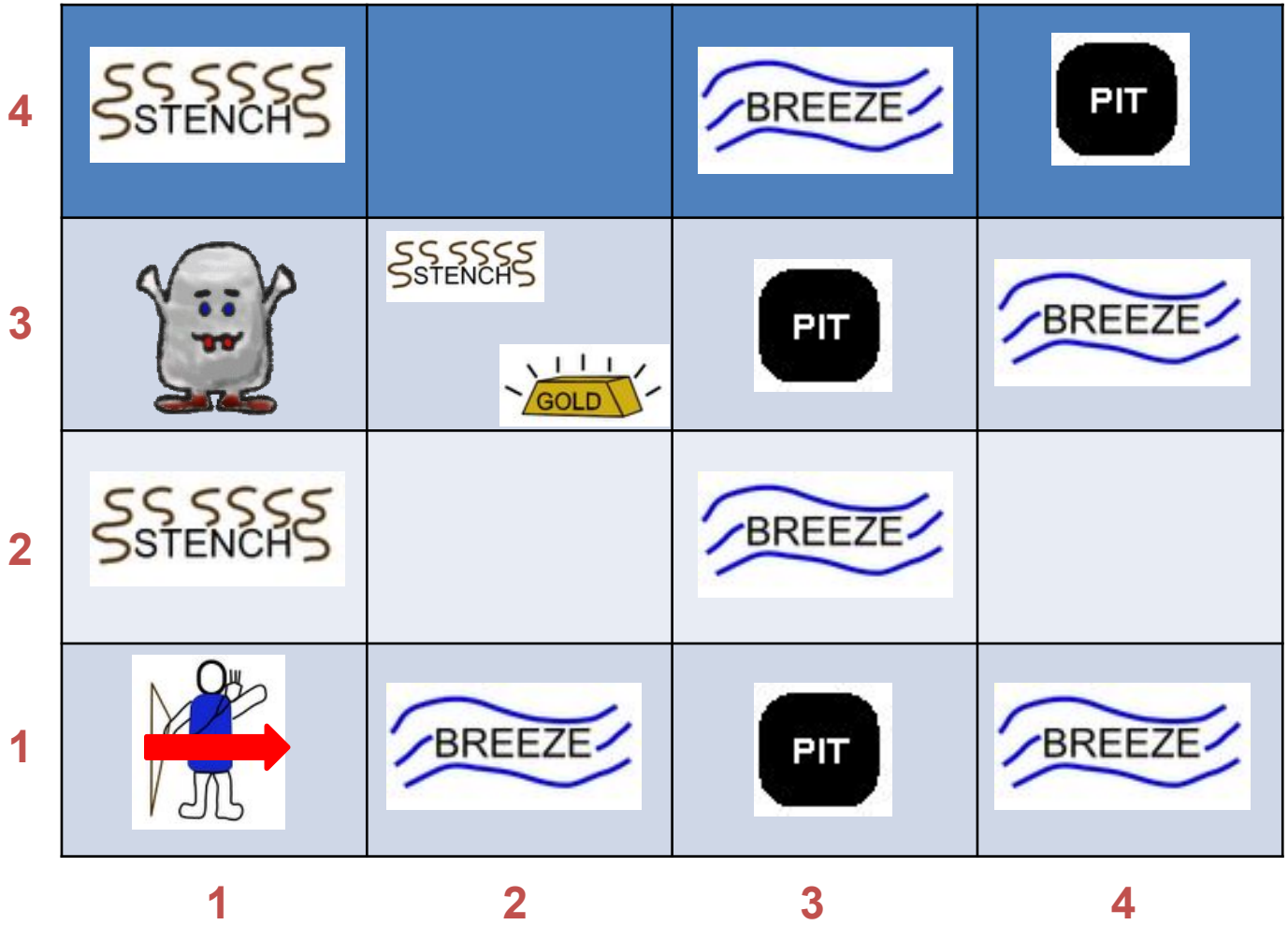
The Wumpus World

- Score:
  - +1000 for picking up the gold.
  - -1000 for falling into a pit or getting eaten by the Wumpus.
  - -1 for each action taken.
  - -10 for shooting the arrow.
- Environment:
  - 4x4 grid in our example.
  - Player starts at (1,1), facing right.
  - Randomly placed pits, Wumpus and gold.
- Actions:
  - Turn 90^0 left or right
  - Move forward
  - Shoot
  - Grab

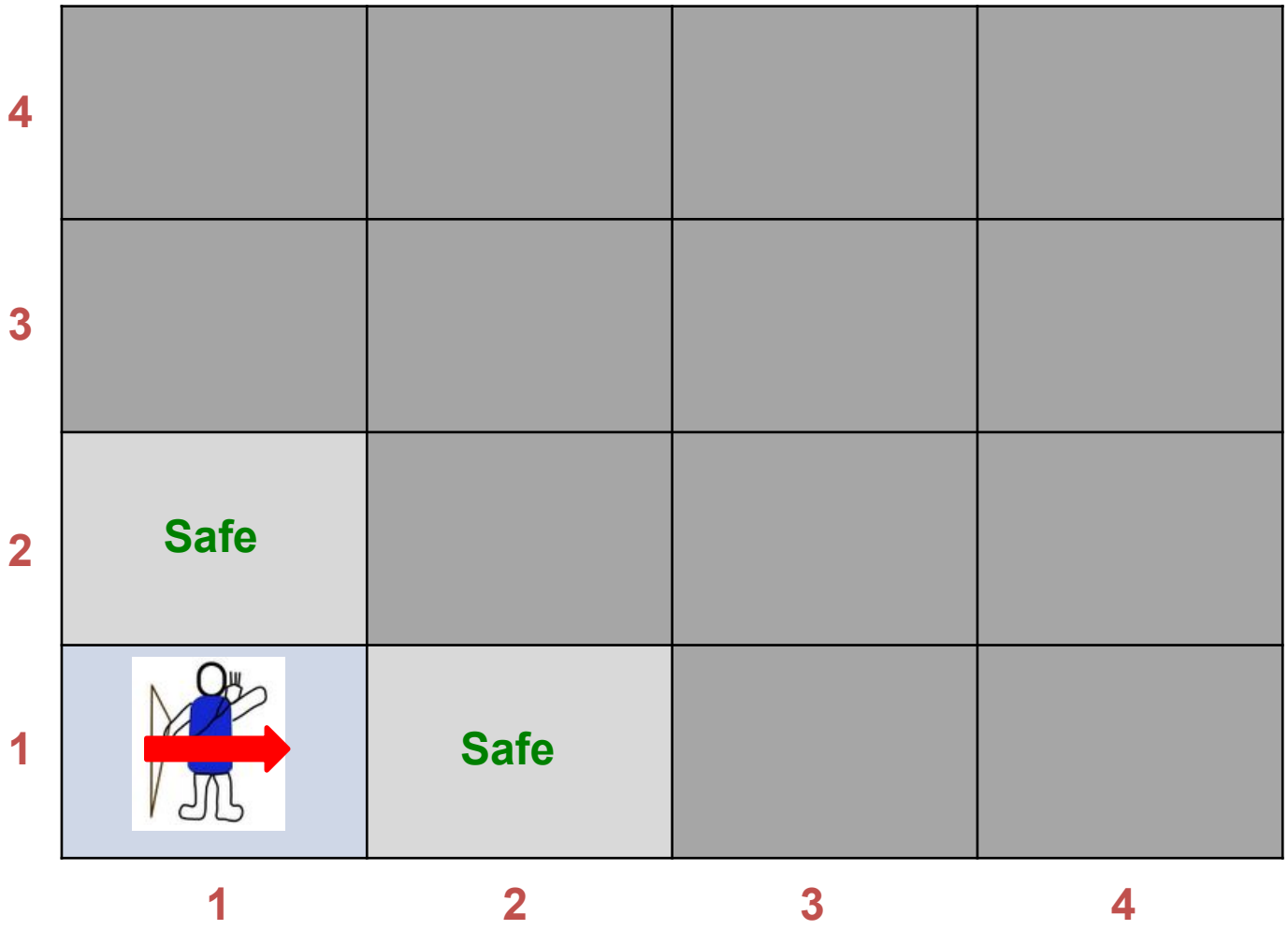
The Wumpus World

- Sensors:
  - In squares next to the Wumpus the player perceives a stench (not diagonally).
  - In the squares next to a pit the player perceives a breeze (not diagonally).
  - In the square with the gold treasure, the player perceives a glitter.
  - If the Wumpus is killed, a scream is heard all over the cave.
  - Percepts: [Stench, Breeze, Glitter, Bump, Scream]
  - Example: [Stench, Breeze, None, None, None]

Example Wumpus World

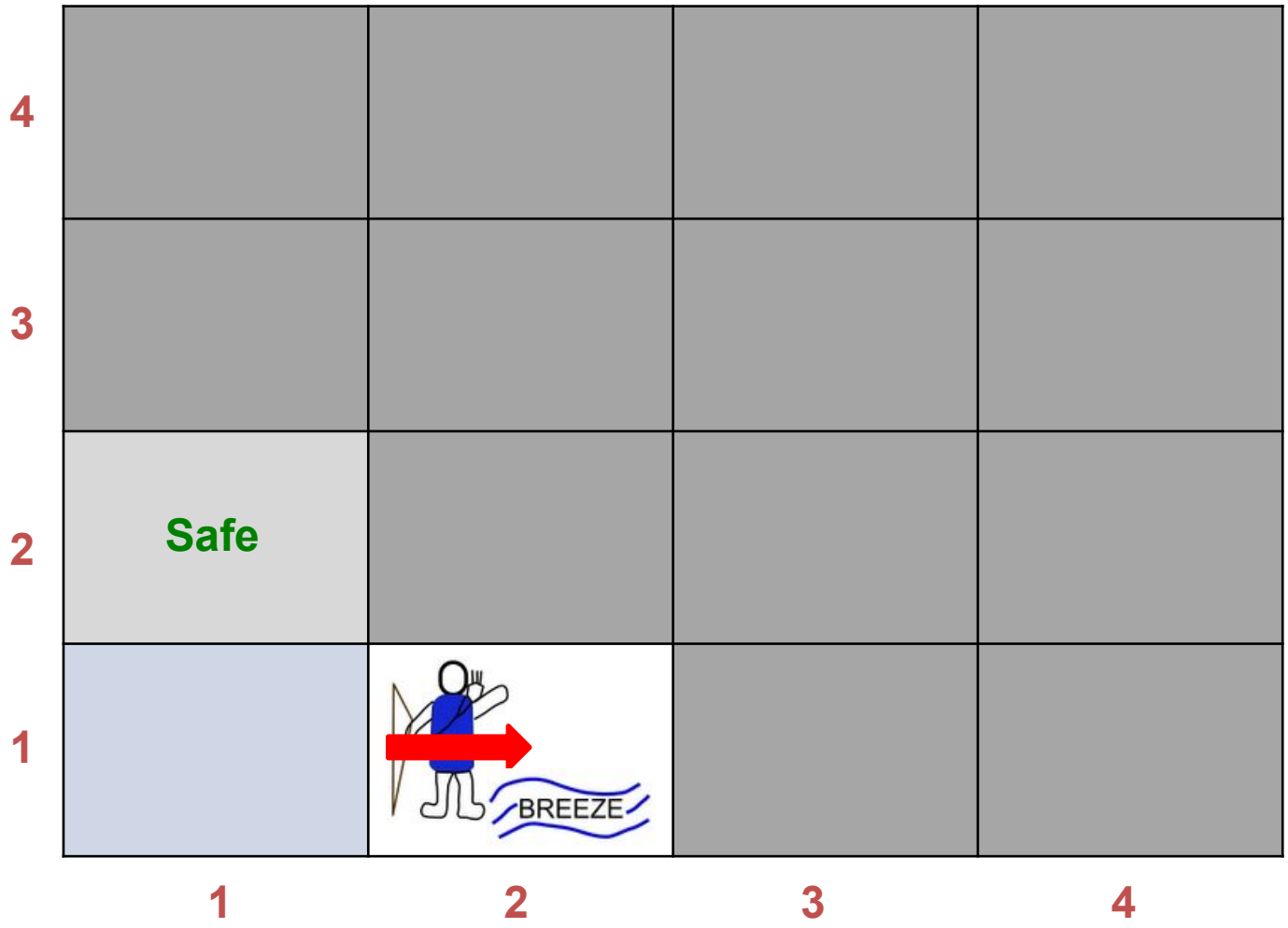


$\text{Percept}_{(1,1)}$ = [None, None, None, None, None]



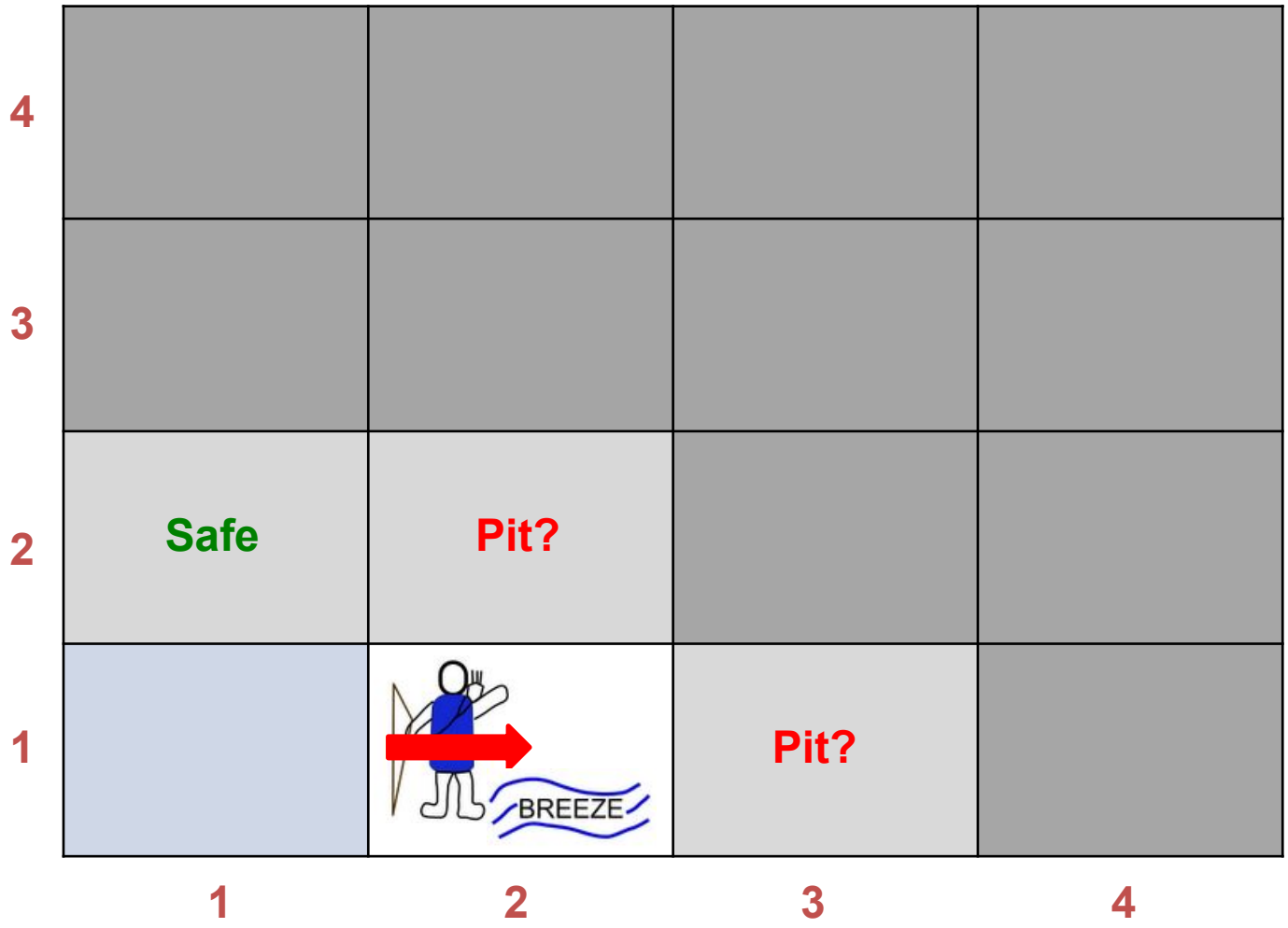
Action: Move Forward

$\text{Percept}_{(2,1)}$ = [None, Breeze, None, None, None]



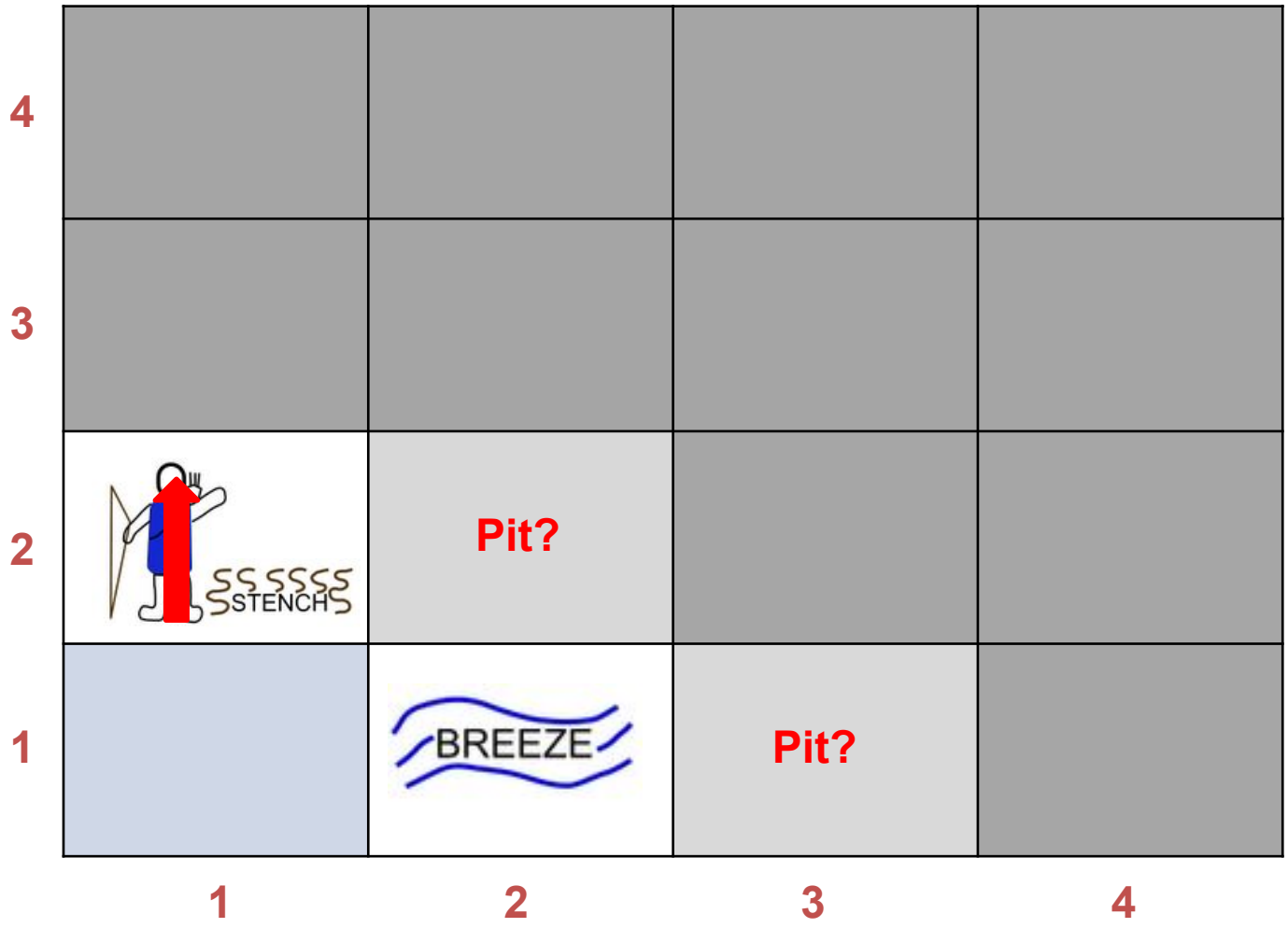
What conclusions can we make?

$\text{Percept}_{(2,1)}$ = [None, Breeze, None, None, None]



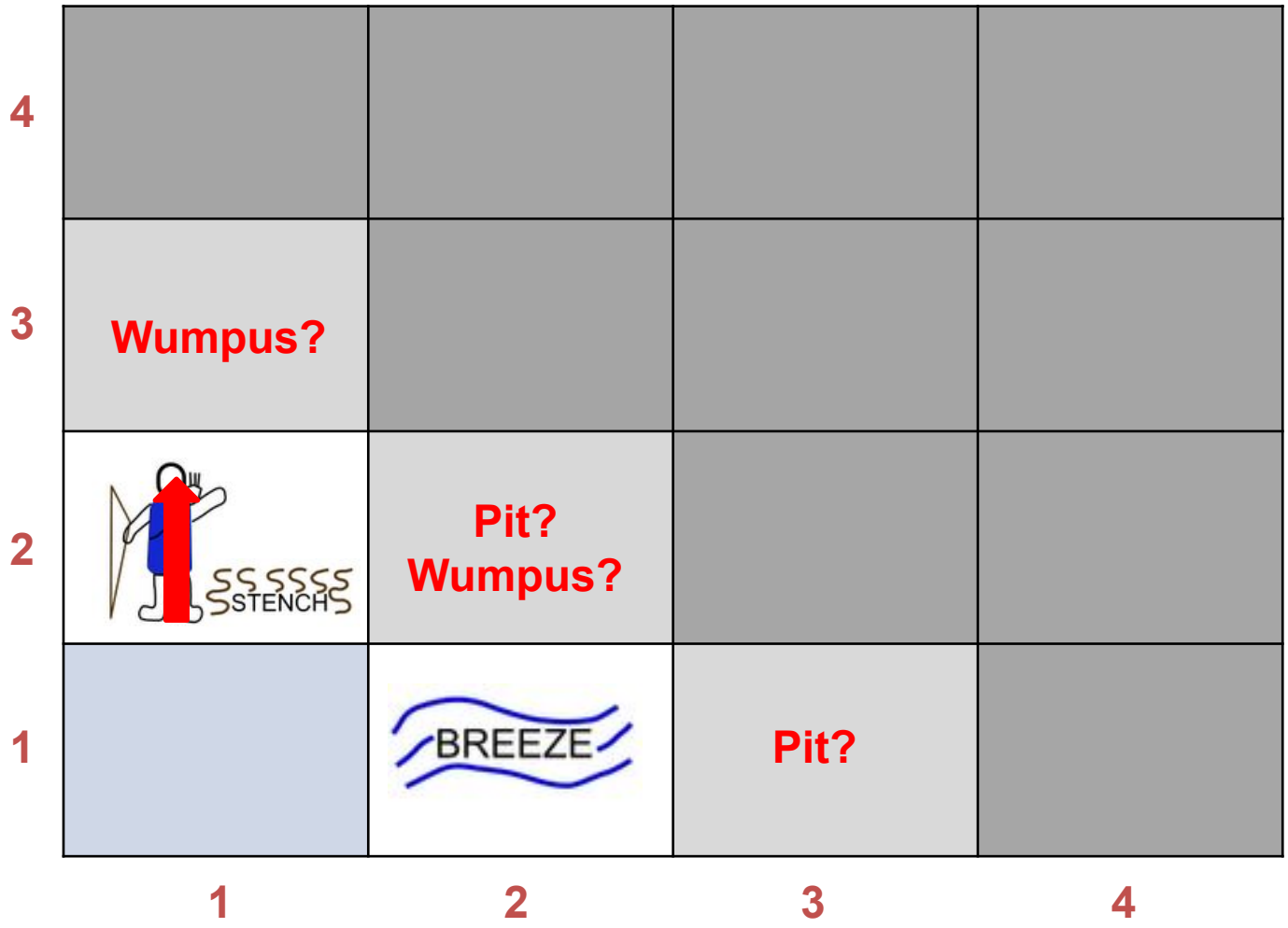
We need more information…

$\text{Percept}_{(1,2)}$ = [Stench, None, None, None, None]



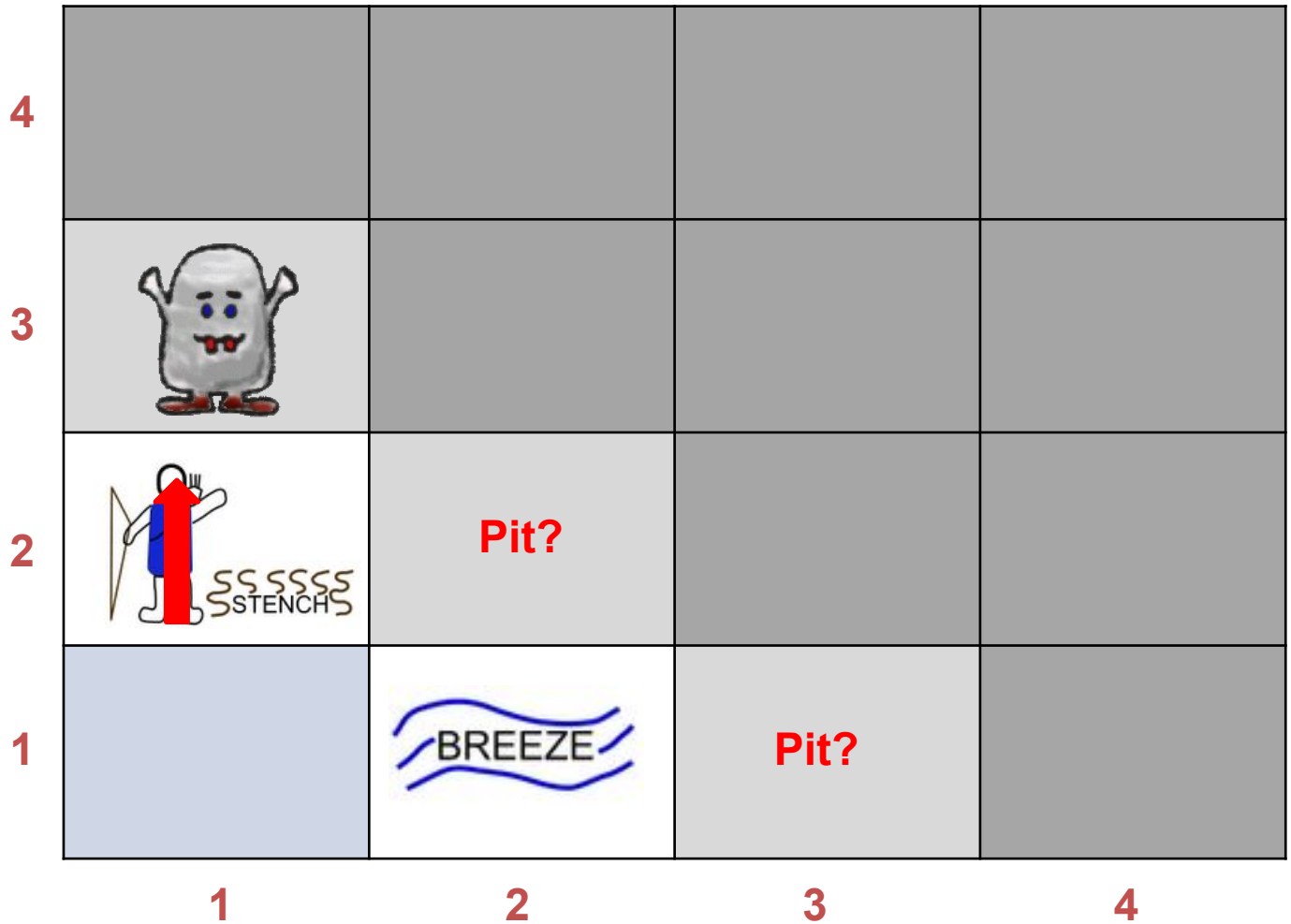
What conclusions can we make?

$\text{Percept}_{(1,2)}$ = [Stench, None, None, None, None]



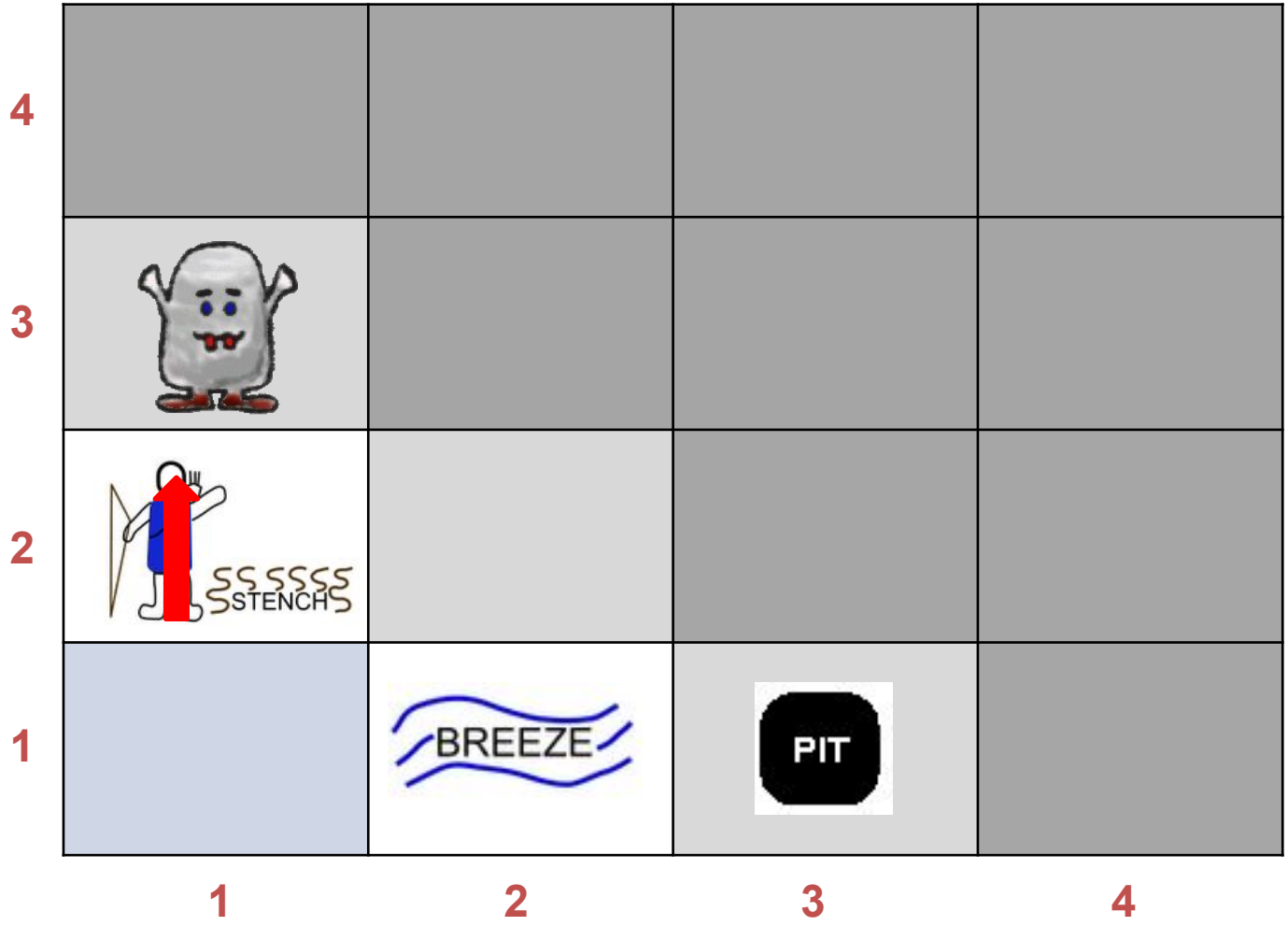
The Wumpus is nearby, but where?

$\text{Percept}_{(1,2)}$ = [Stench, None, None, None, None]



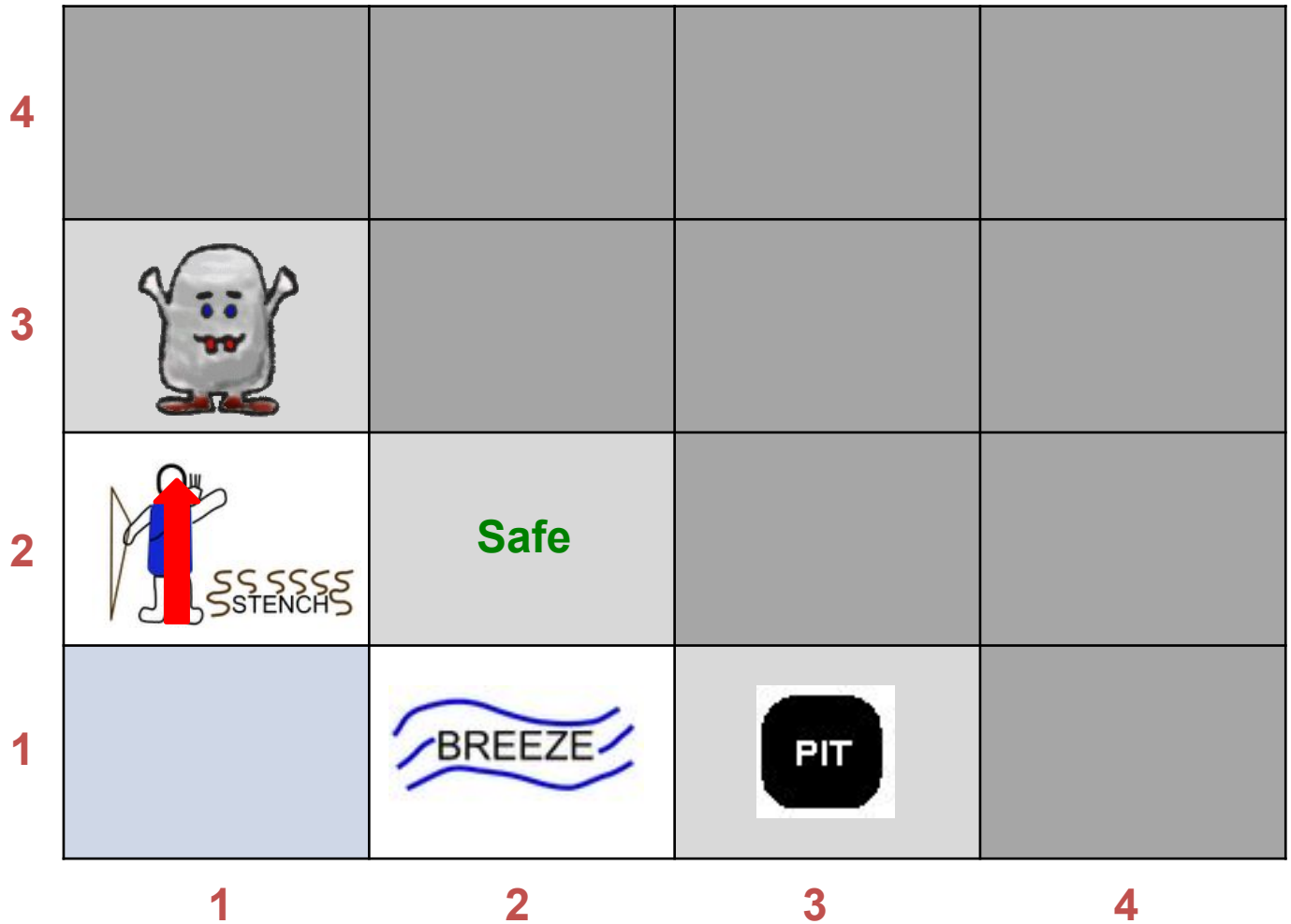
Wumpus must be in (1,3), since no stench was perceived in (2,1)

$\text{Percept}_{(1,2)}$ = [Stench, None, None, None, None]



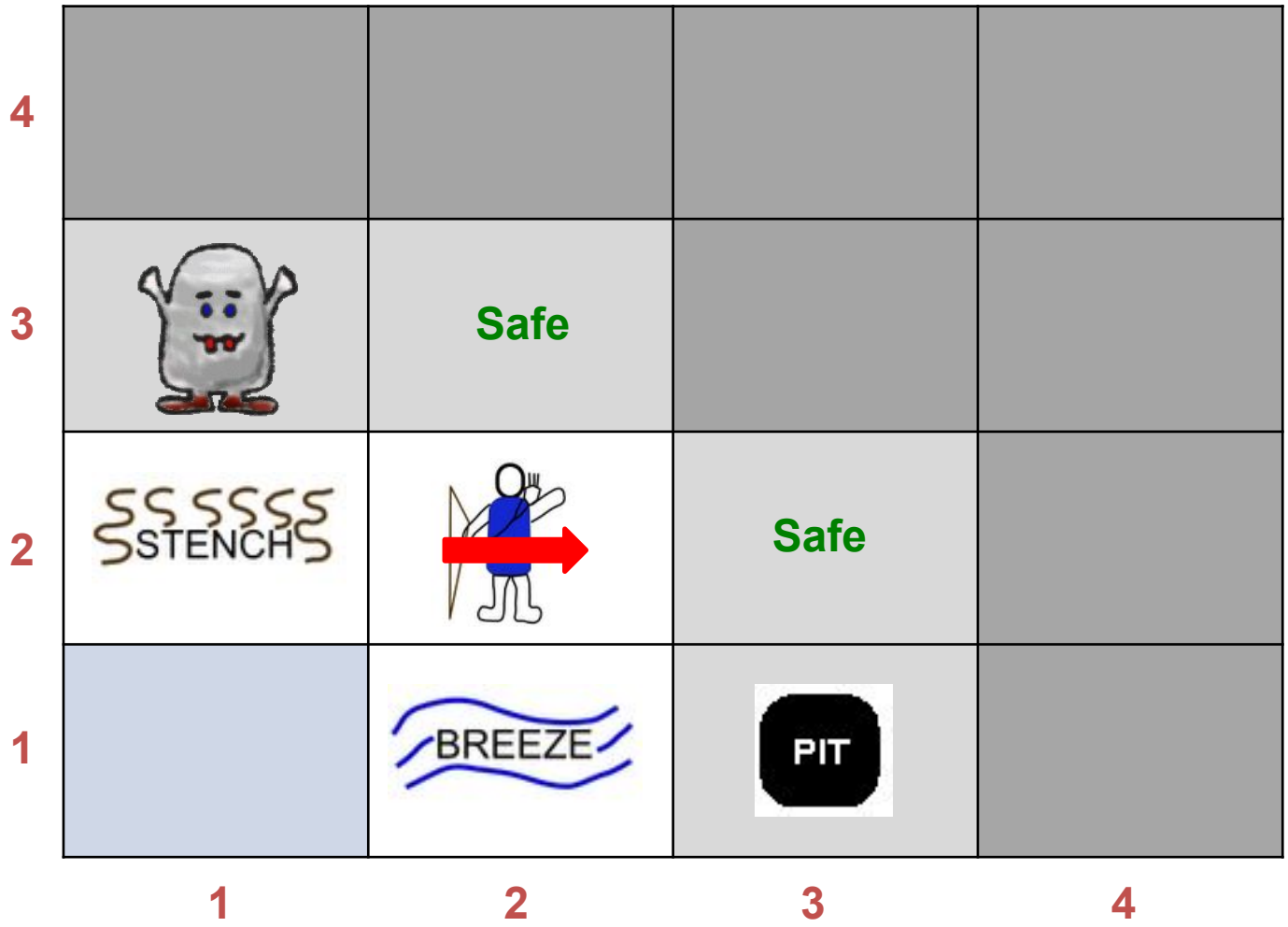
We can also conclude that the Pit must be in (3,1), since no Breeze is perceived in (1,2).

$\text{Percept}_{(1,2)}$ = [Stench, None, None, None, None]



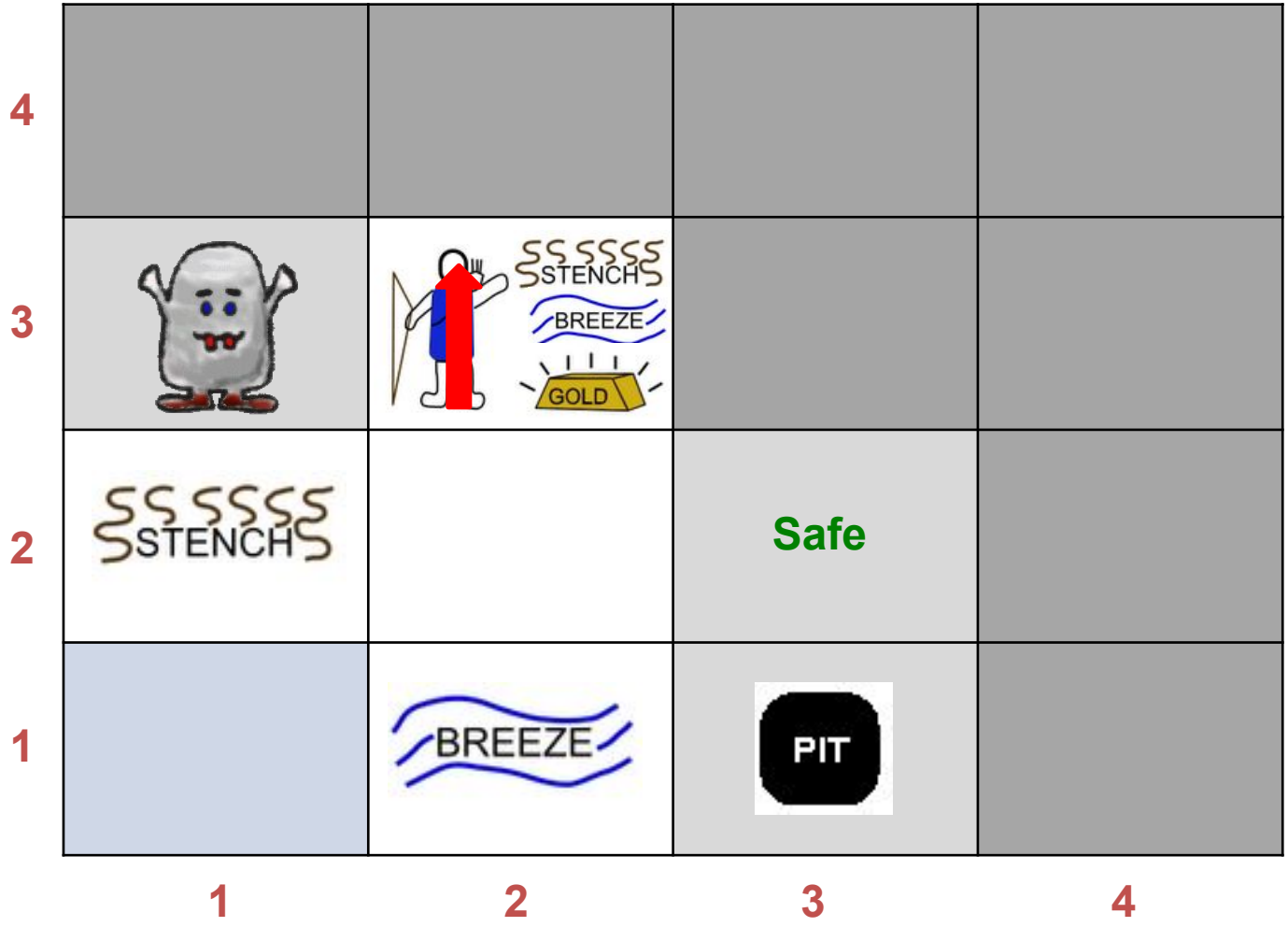
And that (2,2) is safe, since no Breeze is perceived and we know where Wumpus is.

$\text{Percept}_{(2,2)}$ = [None, None, None, None, None]



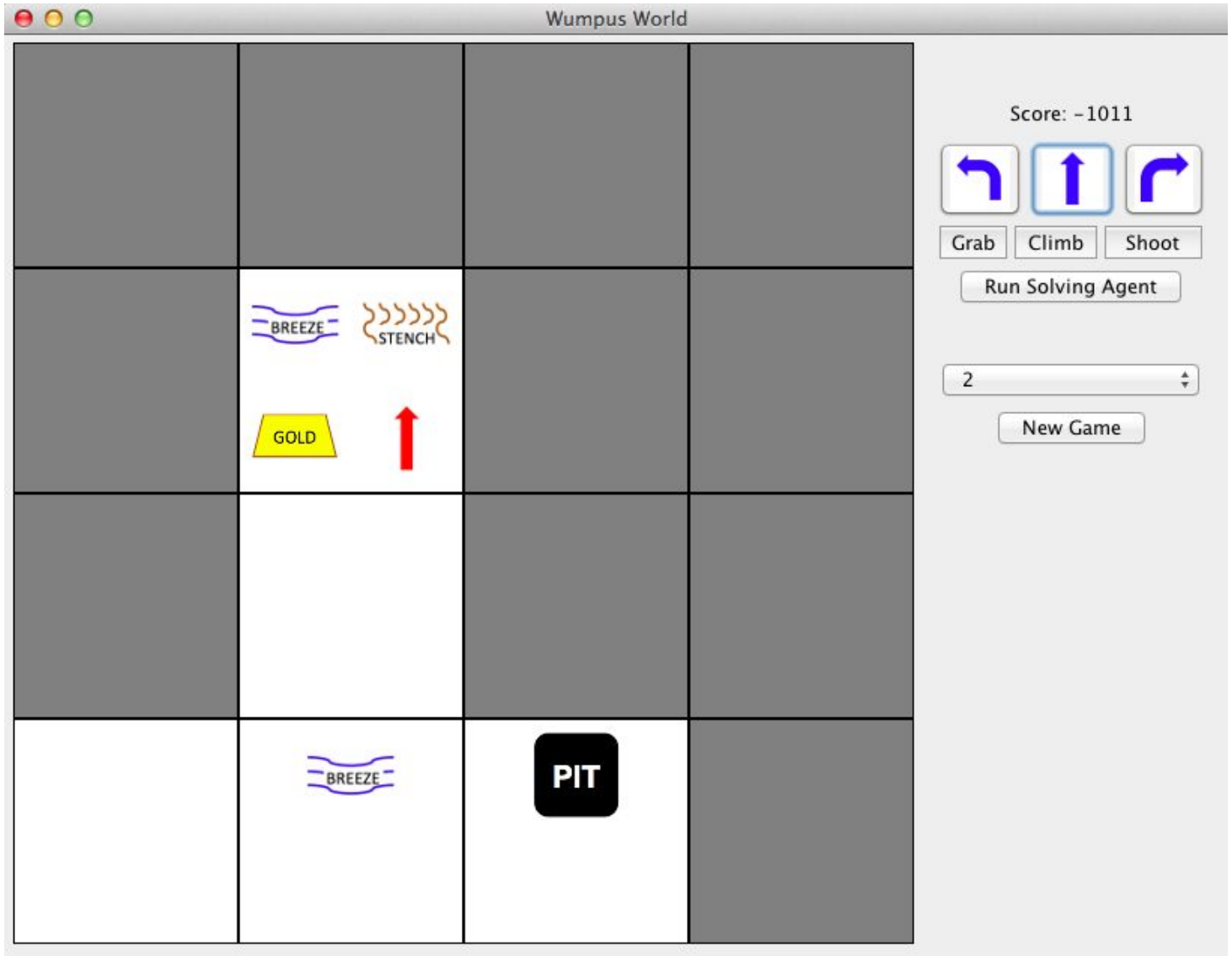
No Stench or Breeze, so (2,3) and (3,2) must be safe! We choose (2,3).

$Percept_{(2,3)}$ = [Stench, Breeze, Glitter, None, None]



We sense Glitter, so lets dig up the treasure!

The Application

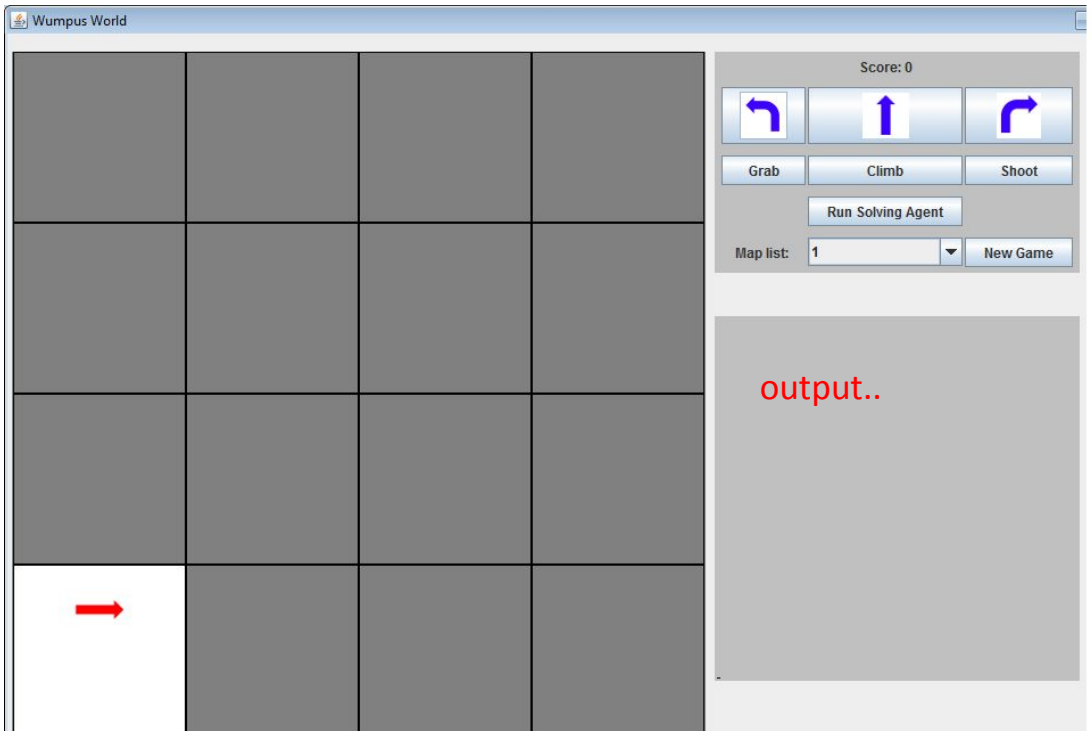


The Application

- The application is written in Java.
- Your task is to add code to the MyAgent.java class to create an ”intelligent” player for game.
- The class contains some examples of basic methods you need to use.
- Your code is called by clicking the *Run Solving Agent* button in the GUI.

The Application

- You may create additional classes and add additional UI elements to GUI.java.
- You may **not** edit the original application code.



Basic methods

```
//Location of the player
int cX = w.getPlayerX();
int cY = w.getPlayerY();
```

```
//Basic action:
//Grab Gold if we can.
if (w.hasGlitter(cX, cY))
{
    w.doAction(World.A_GRAB);
    return;
}
```

```
//Test the environment
if (w.hasBreeze(cX, cY))
{
    System.out.println("I am in a Breeze");
}
if (w.hasStench(cX, cY))
{
    System.out.println("I am in a Stench");
}
if (w.getDirection() == World.DIR_LEFT)
{
    System.out.println("I am facing Left");
}
```

```
//Move actions:
w.doAction(World.A_TURN_LEFT);
w.doAction(World.A_TURN_RIGHT);
w.doAction(World.A_MOVE);
```

Requirements

- Grade E:
  - Rule-based if-then system.
- Grade D:
  - Solution based on A* algorithm.
- Grade C:
  - Solution using neural networks **with** external neural network libraries.
- Grade B:
  - Solution based on Naïve Bayes approach.
- Grade A:
  - Solution using neural networks. Own implementation of neural network required, **without** usage of external libraries.
- A solution that does not behave well (by for example missing obvious percepts) will receive an Fx grade.

Requirements

- All solutions have to be compatible with the given code
- Comment your code to make the grading task easier

Grade D

- One possible solution
  - Model the problem as a graph.
  - Define terminal state.
  - Come up with heuristic function.
  - Apply A* search.

Grade B

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- B - known data (percepts, ...)
- A - queried parameter after known data are taken into account.
- P(B|A) - likelihood of known data with the parameter - can be calculated from dataset.

Grade A and C

- Simplify the problem, help the neural network on the way
  - For example, it doesn’t necessarily have to decide when to shoot
- The neural network should play a significant part in the decision making
- Additional logic may be employed

What to submit

- The complete source code for the Wumpus World program containing your AI agent. The program should be runnable.
- A report explaining your approach to solve the problem and who has done what.
- Submit to It's Learning no later than **the deadline on canvas**