

Saving Shipwrecked Sailors with Bayes' Rule

Clara García Latorre
Noemí Serna Martín
Jimena Arnau Martínez

Summary

1. Project Objective
2. Methodology
3. Theoretical Background
4. Search Probabilities
5. Technologies Used
6. Game Mechanics
7. Search Algorithm
8. Simulation Results
9. Conclusion

Project Objective

In this project we have create an interactive simulation game where a robot explores caves to find lost sailors using Bayesian inference. The main objectives are:

- Demonstrate how Bayesian reasoning can be applied in a game context.
- Visualize probability updates and robot behavior dynamically through a GUI.
- Integrate logic, statistics, and user interface design to deliver an Captivating and educational experience.

Methodology



1. Design of game logic with Bayesian updating.
2. GUI development using Tkinter.
3. Creation and integration of visual assets (cave and robot images).
4. Randomized assignment of sailor location at game start.

Theoretical Background

The robot uses Bayes' Theorem to update the probability of each cave containing the lost sailors:

$$p(H | E) = \frac{p(E | H) p(H)}{p(E)}$$

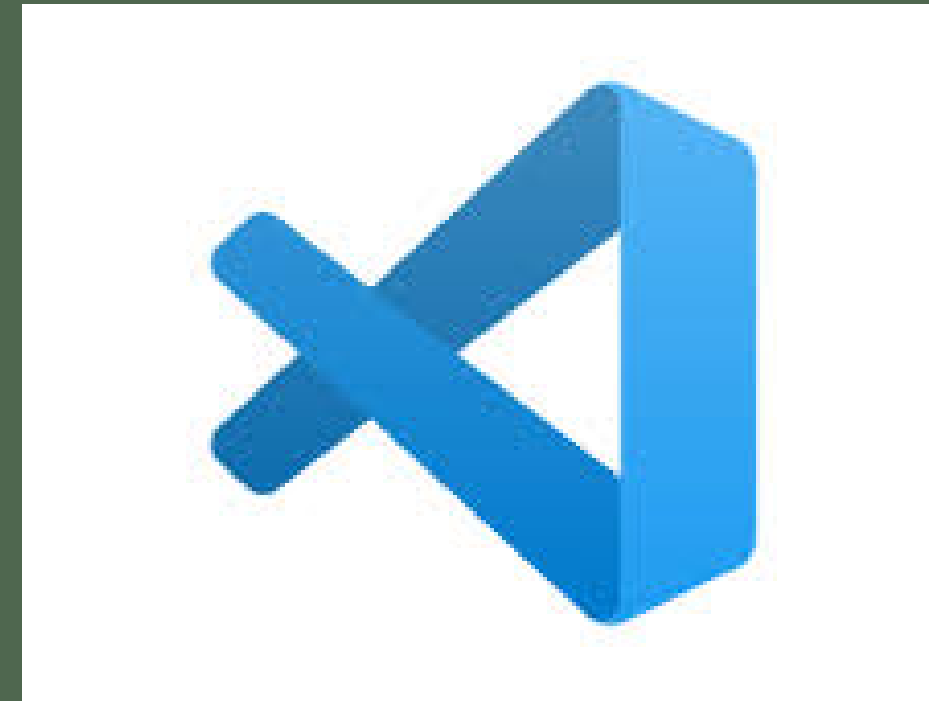
H (Hypothesis) - E (Evidence)

After each search, probabilities are updated, helping the robot focus on the most likely cave.

Search Probabilities

In this model, if we search the right cave, there's a 90% chance we'll find the target. But there's also a 10% chance that searching a wrong cave will wrongly tell us the target is there. This makes things tricky, and we have to use logic and update what we believe based on each search result to make better decisions

Technologies Used



Game Mechanics

- Objective: Find the hidden sailors using intelligent search.
- Maximum of 10 search steps.
- Game ends when: Probability of the correct cave exceeds 0.999, or 10 searches have been performed.

Search Algorithm

- Initial probabilities: $1/3$ for each cave (A, B, C)
- Sailors hidden in one random cave
- Robot searches one cave per step
- Outcome is uncertain (90% true positive, 10% false positive)
- Probabilities updated using Bayesian logic

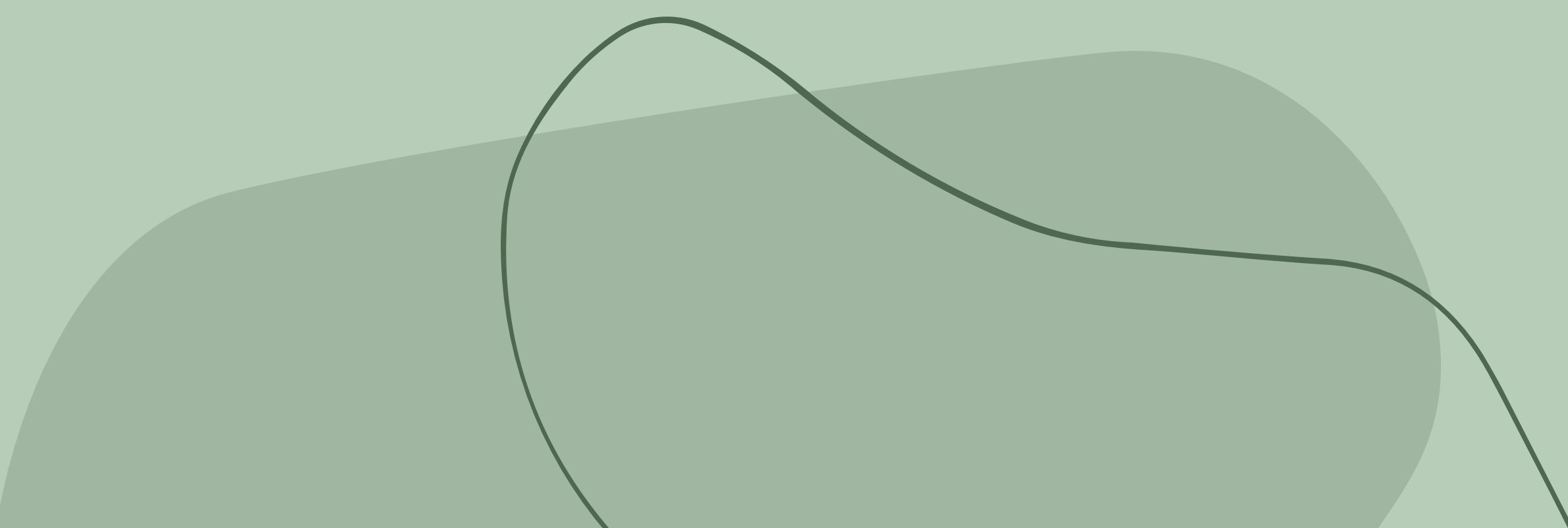
Movement Constraint

- Robot moves only among caves A, B, and C
- Always goes to the cave with the highest probability
- Stops searching if:
 - A cave reaches 99.9% certainty
 - 10 steps have passed

Simulation Results

Conclusion

The project successfully achieved its goal by simulating a search using Bayesian inference. The robot intelligently selects the most probable cavern at each step, and the GUI clearly displays the robot's progress and probability updates. Despite some challenges with image handling and smooth updates, these were resolved. Overall, the project provided valuable experience in applying Bayesian reasoning and integrating logic with a graphical interface using Tkinter.

A decorative abstract graphic in the bottom left corner, consisting of a dark green wavy line and a light green semi-transparent circular shape.