Are Bugs Intelligent?
Student Workbook

How This Workbook Works

In this project, you will explore a simple but surprisingly deep question:

Are bugs intelligent?

You will:

- solve and design mazes,
- think about intelligence and memory,
- model bug movement using probability,
- compare your own maze-solving strategies to a "very dumb bug."

This workbook is written to you, not about you. You will read, think, sketch, calculate, and reflect as you go.

Day 0: Solving a Tricky Maze

Today you will start by solving a maze and thinking about how you solve it.

Part 1: Solve the Maze

Your teacher will give you a maze. Work quietly at first and try to solve it.

Time started: ______________ **Time finished:** ______________

As you worked, what was your strategy? Did you:

- trace the walls?
- look ahead for dead ends?
- try different branches and backtrack?
- something else?

Describe your approach in a few sentences:

Part 2: Before and After

Before solving the maze:

1. Do you think you are more intelligent than a bug? Why or why not?

2. Do you think you can solve a maze faster than a bug? Why?

After solving the maze:

1. Do you still think you can solve a maze faster than a bug? Explain.

2. What do you think gave you this (in)ability? List at least two reasons.

Day 1: Designing Your Own Maze

Today you will switch roles: you are the maze designer.

Part 1: What Makes a Maze Hard?

Think about the maze you solved yesterday and other mazes you have seen.

1. List at least three things that can make a maze harder.

2. List at least two things that can make a maze easier.

Part 2: Draw Your Maze

Your job is to create a maze on graph paper.

- It should have one clear start and one clear finish.
- It should have at least a few decision points (places where a path splits).
- It should not be impossible.

Use the space below or separate graph paper (if provided).

Figure 1: Sketch of my maze (rough draft)

Part 3: Reflect

1. What did you find harder: solving a maze or creating one? Why?

Day 2–3: Building Mazes in 3D (OpenSCAD)

You will now learn how to model simple 3D shapes with a tool called OpenSCAD.

Part 1: Basic Shapes

Your teacher will show you how to create:

- cubes,
- spheres,
- cylinders.

Write down one example command for each:

1. Cube command example:

2. Sphere command example:

3. Cylinder command example:

Part 2: Transformations

You can also:

- move shapes (translate),
- rotate shapes,
- scale shapes,
- mirror shapes.

Your teacher will show you examples. Write down one transformation that you think is especially useful for building a maze and explain why:

Part 3: 3D Version of Your Maze

You will start creating a 3D version of your maze.

1. What is one advantage of having your maze in 3D instead of just on paper?

Day 4: Brains, Memory, and Maps

Today you will connect mazes to brains and memory.

Part 1: Storing a Map

Imagine you want to store a map in a computer or in a brain.

1. List at least two ways a map can be represented (for example, as a picture, as a list of connections, etc.).

2. What might be an advantage of representing a map as a list of connections (which places are connected to which)?

Part 2: Tiny Brains

Your teacher will talk about how many neurons a small animal (like an ant or a bee) might have.

1. Do you think a bug can store a detailed map of a maze in its brain? Why or why not?

Day 5: Designing an Experiment

Today you will think like a scientist: how could we test whether bugs are intelligent?

Part 1: Fair Tests

A fair experiment has to:

- control outside influences,
- use a clear measurement,
- be repeatable.

Your teacher will share part of a talk by Richard Feynman about good and bad experiments.

1. Suppose you put food at the end of a maze for a bug. Why might this not be a good test of "intelligence"?

2. In our project, what might we use as a measure or score for "bug performance" in a maze?

Part 2: A Very Dumb Bug

We are going to compare real bugs (which may have some intelligence) to a **very dumb bug** that just moves randomly.

1. Describe, in your own words, what a "very dumb bug" does at a decision point in a maze.

Day 6: Random Walks (A Very Dumb Bug in Action)

Today you will perform a simple random-walk experiment.

Part 1: Coin Flip Walk

Your teacher will set up an experiment, for example:

- flip a coin to move left or right along a board,
- or use another simple rule.

As you run the experiment, record what happens.

1. Describe the rules of your random walk experiment here:

2. What patterns (if any) did you notice in where the marker ended up after several steps?

Part 2: Fractional Bugs (Ensembles)

Instead of following just one bug, imagine you could follow many identical bugs at once.

1. Why might it be helpful to think about many bugs at once, instead of just one?

Day 7: Probability Tables (How a Bug Decides Where to Move)

Now we will describe the bug's movement using **probability tables**.

Part 1: A Simple Example

Imagine a tiny maze with three important points: 3, 4, and 5.

From point 4, the bug can go to 3 or 5, each with probability 1/2:

Current Node	Next Node	Probability
4	3	1/2
4	5	1/2

From point 5, suppose the bug can go to 3, 5, or 6, each with probability 1/3:

Current Node	Next Node	Probability
5	3	1/3
5	5	1/3
5	6	1/3

Part 2: Building Your Own Probability Table

Your teacher will show you a simple graph or part of a maze with labeled points.

1. For each point, list all possible next points and assign probabilities that add up to 1.

Use the table layout below for one of the nodes:

Current Node	Next Node	Probability

Repeat for other nodes as needed.

Day 8: Multi-Step Probabilities and Probability Trees

Today you will see how probabilities change as the bug takes more than one step. To help with this, we will use something called a **probability tree**.

Part 1: The Bug World (Movement Rules)

We will use a tiny "bug world" with five important nodes: 2, 4, 3, 5, and 6. The bug always starts at node 4.

The table below shows all possible moves the bug can make and the probability of each move.

Current Node	Next Node	Probability
4	3	1/2
4	5	1/2
3	4	1/2
3	5	1/2
5	4	1/3
5	3	1/3
5	6	1/3

Part 2: What Is a Probability Tree?

A probability tree is a diagram that shows:

- all possible paths the bug can take,
- the probability of each step,
- how those probabilities multiply along a path.

Each level of the tree represents one step. A tree of depth 2 shows all possible 2-step paths.

Figure 2: Sketch your probability tree here

The probability of a path is the **product** of the probabilities along its branches:

$$P(4 \to 3 \to 2) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}.$$

Part 3: Build Your Own Two-Step Probability Tree

Now it's your turn to **draw the probability tree yourself**. Use the movement rules from the table above. Your tree should:

- start at node 4,
- show every possible move after 1 step,
- show every possible move after 2 steps,
- label each branch with its probability.

This is a hand-drawn tree (much simpler than the computer-generated one you saw earlier). Make sure each branch splits according to the probabilities in the table.

Draw your 2-step probability tree below:

Figure 3: Your 2-step probability tree

Part 4: List All Two-Step Paths

Each path is a sequence of nodes the bug might visit. For example: $4 \to 3 \to 5$.

List all the two-step paths from your tree:

1.

2.

3.

4.

5.

Part 5: Compute the Probability of Each Path

Multiply the probabilities along each branch.

Example:

$$P(4 \to 3 \to 5) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}.$$

Compute the probability of each path you listed:

1.

2.

3.

4.

5.

Part 6: Where Does the Bug End Up?

To find the probability of ending at a certain node, add the probabilities of all paths that end there.

1. Probability of ending at node 3:

2. Probability of ending at node 5:

3. Probability of ending at node 6:

Part 7: From Trees to Distributions

A probability tree shows every path. A probability distribution summarizes the same information in a table.

Worked Example: From a Tree to a Distribution

Here is a small example using a different tiny "bug world" so you can see how to convert a probability tree into a probability distribution without giving away the answers to your own tree.

Suppose a bug moves among three nodes: A, B, and C, as depicted in the tree that follows the table.

Current Node	**Next Node**	**Probability**
A	B	1/2
A	C	1/2
B	A	1
C	C	1

Figure 4: Draw your two-step probability tree inside this box.

If the bug starts at A, the **Step 0 distribution** is:

$$[\,P(A) = 1,\ P(B) = 0,\ P(C) = 0\,].$$

Step 1

From A, the bug can go to B or C with probability 1/2 each. So the **Step 1 distribution** is:

$$[\,0,\ \tfrac{1}{2},\ \tfrac{1}{2}\,].$$

Step 2

Now update each part:

From B (probability 1/2), the bug goes to A with probability 1.
From C (probability 1/2), the bug stays at C with probability 1.

So:

$$P(A) = \tfrac{1}{2} \cdot 1 = \tfrac{1}{2}, \qquad P(B) = 0, \qquad P(C) = \tfrac{1}{2} \cdot 1 = \tfrac{1}{2}.$$

Thus the **Step 2 distribution** is:

$$[\,\tfrac{1}{2},\ 0,\ \tfrac{1}{2}\,].$$

This example shows how a probability tree and a probability distribution describe the same information: the tree lists all paths, and the distribution summarizes the results.

Using your tree:

1. Using your probability tree from Part 3, write the distribution after 1 step.

2. Using your tree, write the distribution after 2 steps.

3. Explain how the distribution method and the tree method give the same results.

The key idea:

> A probability tree shows all possible paths. A probability distribution summarizes the same information without drawing the whole tree.

Day 9: Turning Your Maze into a Graph

Now you will turn **your** maze into a graph and a set of probability tables.

Part 1: Label Decision Points

1. On your maze drawing, circle every place where the path splits (a choice point).
2. Give each decision point a number: 1, 2, 3, . . .

Sketch the graph version of your maze below (just circles and lines):

Figure 5: Graph of my maze

Part 2: Build Probability Tables for Your Maze

For each node (decision point), decide:

- What nodes can the bug move to from here?
- What probabilities should each move have?

If the bug just chooses randomly among available corridors, then each move has equal probability.

Fill in at least one table for a node in your maze:

Current Node	Next Node	Probability

Day 10: How Likely Is the Bug to Finish?

Today you will estimate how likely it is that your very dumb bug finishes the maze after a certain number of steps.

Part 1: Starting the Bug

Pick:

- one node as the start,
- one node as the exit.

1. Write down your starting distribution (100% at the start node, 0% everywhere else).

Part 2: Updating the Distribution

For each step, you can update the probabilities using this idea:

> New probability at a node = sum over all previous nodes of
> (previous probability at that node) × (probability of moving into the new node).

Your teacher may give you a table layout or a digital tool to help with this.

1. Compute the distribution after 1 step.

2. Compute (or estimate) the distribution after a few more steps.

3. For each step, record the probability that the bug is at the exit node.

Part 3: Reflecting on Bugs and Intelligence

1. How does the probability of the bug finishing the maze change as the number of steps increases?

2. Compare your own maze-solving process to the bug's random walk. List at least two ways your thinking is different.

3. After this project, how would you answer the question:
   "Are bugs intelligent?"
   Write a short paragraph with your current thoughts.

Day 11: Running the "Very Dumb Bug" Experiment

Today you will run a real experiment to see how a **very dumb bug** moves through a maze. This experiment will help you compare **physical randomness** to the **probability tables** and **trees** you have been building.

Part 1: Set Up Your Maze

Your teacher will give you a maze model. It might be:

- a 3D maze you can tilt,
- a flat maze on paper,
- or a simple graph with nodes and connections.

Choose a small object to be your **bug** (a bead, a token, or a small ball).

Part 2: The Bug's Rule

Your bug is **very dumb**. At every decision point, it chooses randomly between the available paths.

Use one of these tools to make the choice:

- **2 choices:** flip a coin
- **3 choices:** roll a die (1–2 = first option, 3–4 = second, 5–6 = third)
- **4+ choices:** use a spinner or numbered cards

You must follow the random choice—no thinking ahead.

Part 3: Run a Single Trial

Start your bug at the **start node**. Each time it reaches a decision point:

1. Look at the available paths.
2. Use your random tool to choose one.
3. Move the bug along that path.
4. Keep going until:
   - the bug reaches the **exit**, or
   - you reach **15 steps**, whichever comes first.

Record what happened.

Trial Record

- **Path taken (list the nodes):**

- **Did the bug reach the exit?** ☐ Yes ☐ No
- **Number of steps:**

Part 4: Run More Trials

Run **at least 5 trials**. Each trial starts fresh at the beginning of the maze.

Trial #	Path Taken	Reached Exit?	Steps
1			
2			
3			
4			
5			

Part 5: What Did You Notice?

1. How often did your bug reach the exit?

2. Did some paths show up more often than others? Why do you think that happened?

3. Did your bug ever get "stuck" in part of the maze? Describe what happened.

Part 6: Connecting to Probability

Look back at your results.

1. Pick one short path your bug took (for example: $4 \to 3 \to 5$). Write it here:

2. Using your probability table, calculate the theoretical probability of that path.

3. Compare your calculation to how often that path actually happened. Were they close? Why or why not?

Part 7: What Does This Experiment Tell You?

Write a short reflection:

- What did you learn about randomness?
- How does the physical experiment connect to the probability tables and trees you've been building?
- What surprised you?