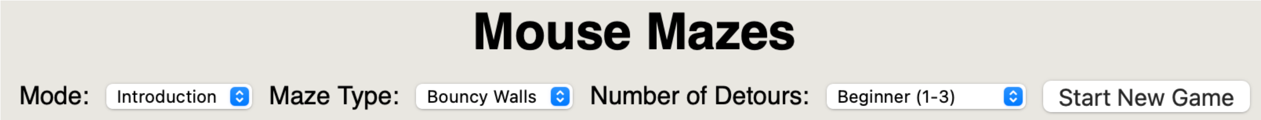
Mouse Maze

A game of inference & hypothesis testing

Mouse Maze

This program was inspired by the program Blockers and Finders, an educational game designed by Tom O'Brien, programmed by Jim Thomas, and distributed by Sunburst Communications in 1987. Blockers and Finders was based on Baffles by James D. Spain, which was inspired by Black Box, an even older public domain program. (This reads a bit like 0:55-1:10 of Danny Kaye's Lobby Number).

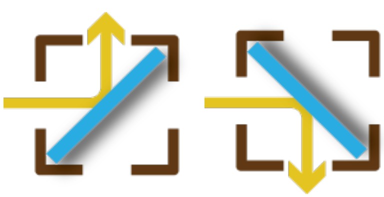


Mouse Mazes

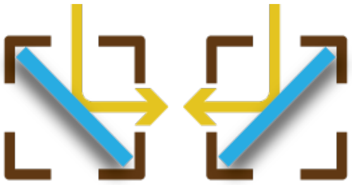
Mode: Introduction Maze Type: Bouncy Walls Number of Detours: Beginner (1-3)

There are two different activities. The program begins in Introduction mode. This activity is designed to introduce students to the mechanics of the mouse mazes, which consist of a 4-by-4 grid of rooms that can be empty or have one of two types of detour (bouncy wall or arrow) inside them. Mice leave their mouse hole and head horizontally if their hole is on a side of the grid (mouse holes 5-8 and 13-16) and vertically if it is on the top or bottom of the grid (holes 1-4 and 9-12). If a mouse enters a room with a tilted bouncy wall, it exits 90° from its entering direction bouncing the way light hitting a mirror would. Here are the behaviors coming into a room...

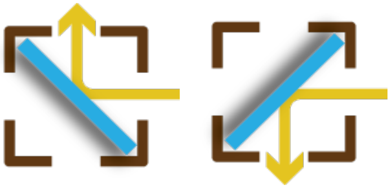
From the left:



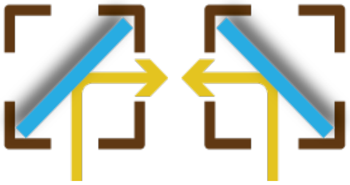
From above:



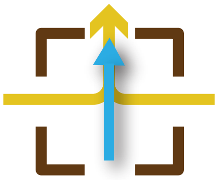
From the right:



From below:



If a mouse enters a room with an arrow, it exits the room in the direction of the arrow regardless of its original path. To the right is an example. If the arrow points back in the direction from which the mouse came, it appears to bounce back.



In the Introduction mode, students can see the mouse's actions and are given eight questions that ask them to predict where a mouse will end up based on where they start or where a mouse started based on where they ended up. They answer questions by clicking on the correct numbered mouse hole along the outside of the grid. For harder problems involving several arrows, it is possible for a mouse to never emerge and there is a button they can use to predict that as well. With arrows, there is often a number of possible answers to the question about where a mouse may have started in order to end up in a particular hole. Any correct answer is accepted and the other options are also shown to emphasize the lack of a 1-to-1 match between starting and ending holes in the arrow mode.

Students can change the maze type from Bouncy Wall to Arrow. They can change the difficulty level which increases the number of possible bouncy walls or arrows that are randomly placed in the rooms. Each time they change the game Mode, Maze Type, or Level, they should click on the Start New Game button.

As noted above, the Introduction mode is less of a game and more of a here-is-how-it-works quick activity. It is fine if a student jumps around with new games with different levels and maze types without even finishing the eight questions, if they are confident they can see how things work in that mode.

The Inference mode is the main point of the game. The computer still places detours in the rooms, but both the layout and the path of the mouse are hidden from view. Students click on a mouse hole to send a mouse into the grid. The mouse will leave its hole, disappear into the grid, and then (if it isn't trapped by arrows) appear at the hole where it exited. That same information, starting and exit hole (if any), is recorded in the Probe Log to the right. Mice always do the same thing, so there is no need to ever probe the maze with the same mouse twice for any given game.

As students see where the mouse enters and leaves, they should click on the grid squares to record their guess about where walls or arrows might be. They are building their model of how they think the computer has arranged the detours. They can click repeatedly to cycle through the wall and arrow types and a blank room. If a mouse probe produces an outcome that they did not expect, it may cause them to re-evaluate a prior hypothesis and change the positions of the detours in the grid. Once they think they are likely to have everything correct (e.g., they are in beginner mode and already have three bouncy walls placed), they should press the blue **Test My Model** button.

When a test of a model is requested, the program will take these steps:

- Simulate all 16 mice and compare how the mice behave in both the student's model and the computer's maze.
- Report how many mouse results matched between the two versions.
- If all 16 mice in the student model end up in the same place as in the hidden maze, the computer will report whether the student model exactly matches the computer's layout or differs but still succeeds by producing the same behaviors. If they differ, which happens quite often when there are many detours, the user will be given the opportunity to see the

hidden model. It is often surprising and fun to see how different the layouts can be that yield the same outcomes.

- Report how many probes were used.

The first goal as students work on these challenges is to have a correct model – one where all 16 mice go to the same place as they do for the computer's layout. As they get more sophisticated, a second goal is to have a successful model with as few mouse probes as seems reasonable. For example, it may be possible to get a correct bouncy wall model with only a few probes. An arrow model at the Expert level may require sending all 16 mouses into the grid to be sure of your model.

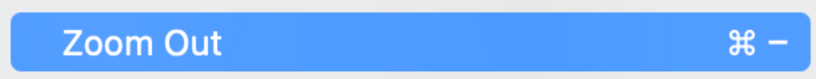
Teaching Notes

Mouse Maze is designed to support:

- Logical reasoning & deductive inference
- Experimentation and data-driven modeling
- Discussion of multiple models that fit the same evidence
- The opportunity to discuss mappings that are 1-to-1 functions (bouncy walls) and non-invertible functions (arrows).

Technical Setup

- Mouse Maze can be run from the makingmath.org website. If the game does not fully display in the browser, you can make it smaller with command-negative sign:



Introduction Mode – Teaching Suggestions

1. **Whole-class demonstration**

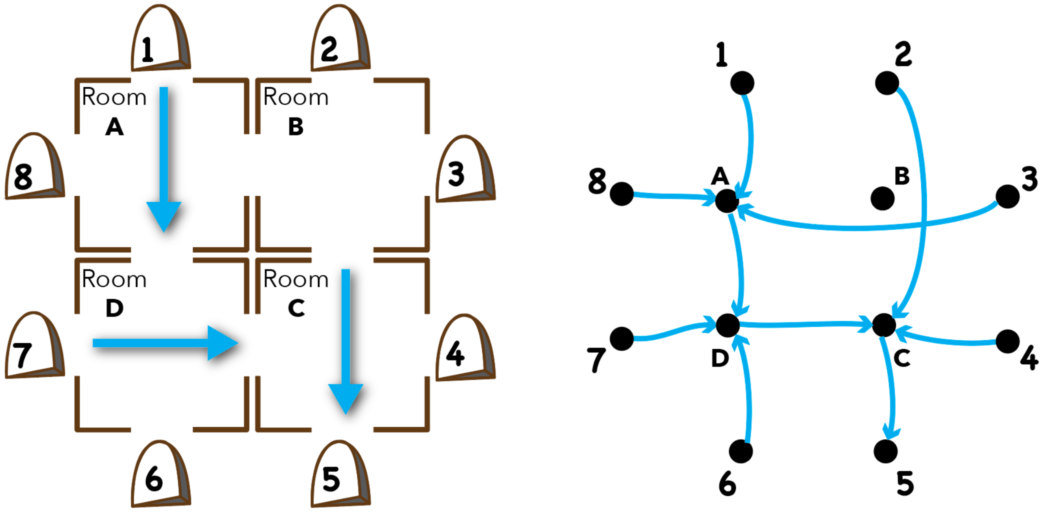
    - Display the game on a projector.
    - Start with:
        - Mode: **Introduction**
        - Maze type: **Bouncy Walls**
        - Level: **Beginner**
    - Model several questions as a group:
        - Have students suggest where they think the mouse will exit.
        - Then click on their chosen mouse hole and reveal the animation and discuss.

2. **Small-group or individual practice**

    - Let students choose Bouncy Walls or Arrows.
    - Encourage them to **verbalize rules**:
        - For walls: "How does each diagonal turn the mouse?"
        - For arrows: "Why does the direction from which the mouse enters **not** matter?"

- Ask them to record data.
    - For example, they might generate a mini table of *start hole → end hole* relationships for a single maze and describe observations and patterns. They might see the inverse relationship that Hole A → Hole B means Hole B → Hole A for the bouncy walls version. They may report that knowing where a mouse exits may not tell you where it began for the arrows model (it is non-invertible)

3. **Traps (arrows mode)**
    - For the harder levels, the arrow mode can produce **cycles** that trap a mouse:
        - Students can be shown how to represent a grid as a directed graph where:
            - Each room is a node; arrows are directed edges. Rooms without an arrow can be ignored (no edge attached to them) since how a mouse leaves such a room depends on how it enters. Just draw directed edges between holes and rooms with arrows.
            - A trapped mouse corresponds to a cycle with no path to the boundary.
        - Here is an example for a smaller 2x2 grid:



        - Note that each node has at most one exiting edge in the graph.

Inference Mode – Teaching Suggestions

1. **Experimental Design**

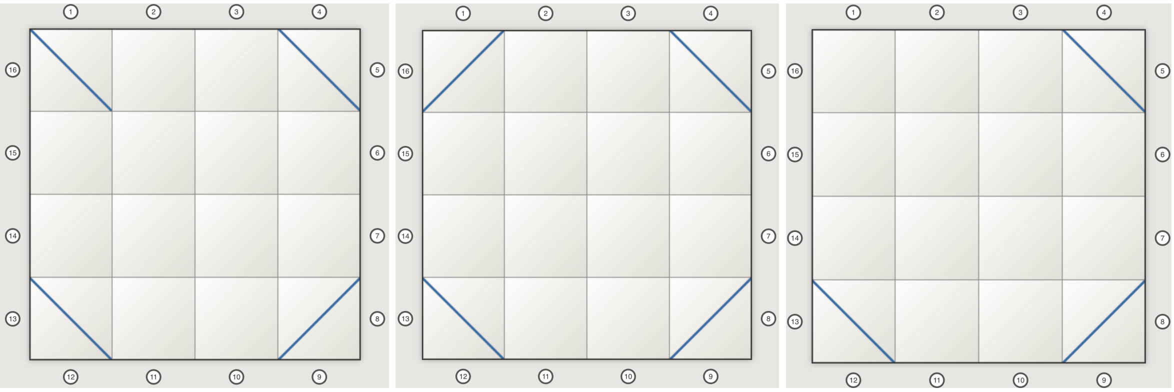
    - Each click on a hole is an experiment.
    - Students are choosing which experiment to run next.

        - "Which probe would give us the **most information**?"
        - "Is there a hole we haven't used yet that would help distinguish between two possible layouts?"

2. **Evidence → Model**

    - They're building a model to fit observed (input → output) data.

- Good prompts:
  - “What do we know for sure about this room from these probes?”
  - “What are two different layouts that would explain the same data?”
  - Sometimes more than one arrangement produces identical outcomes. Their model may not match the computer’s set-up, but the mice may all end up in the same mouse holes in both models.
  - Can they show two different models with identical outcomes on paper (perhaps for a smaller 2x2 or 3x3 grid)? Below is one of my favorite simple examples for which all three mazes produce the same outcomes. They are **equivalent**. The only difference is how far each mouse probe travels.



3. **Efficiency & Meta-cognition: Probe Count**

- The probe counter encourages reflection:
  - Did they use 6 probes? 12? 25?
  - “Could you have done this with fewer experiments?”
  - “If you repeat the activity, can you beat your previous ‘mice used’ score?”

4. **Functions & Non-uniqueness**

- The mapping from holes to holes is still a function: each starting hole has exactly one behavior (a specific exit or stuck in a loop).
- But different internal mechanisms (different mazes) can define the **same function**.
- This is a valuable moment for talking about the difference between matching data and discovering “the true” internal mechanism for a phenomenon. In the case of Mouse Mazes, once there are more than a small number of detours, there is no mechanism for determining the true internal model. We can just match the behavior.

First time using Inference mode

1. **Whole-class demo**

   1. Choose a simple hidden maze (Beginner).
   2. As a group, decide:
      - Which hole to test first, second, etc.
   3. As data points are known, ask:
      - “Can we place any walls/arrows that can achieve this result?”

    - “Are there other layouts that would also work?”
4. Let the class suggest a model, build it on screen, and press **Test My Model**.

2. **Small-group challenge**

   - Groups work with their own devices or time-shared rotations.
   - Give them goals like:

     - “Try to solve a maze using ≤ 12 probes.”
     - “Get 16/16 correct with fewer probes than last time.”

3. **Compare strategies**

   - After a couple of rounds, ask each group to share any strategies or insights.
   - Encourage students to **name their strategies**: “Corner probes,” “opposite-side probes,” “follow the symmetry,” etc.

Extensions

- **Student-designed mazes:**
  - Have student pairs each create a layout on graph paper and play each other in the fashion of Battleship taking turns sending a mouse in and being told where it came out. They only get two chances to test their model and the first one who has all 16 mouse destinations (but not necessarily an identical layout) correct wins.
- **Formal reasoning tasks:**
  - Ask students to write:
    - A short proof or explanation: Why do we never need to send more than 8 probes for a layout with bouncy walls? Can students develop a notation or diagram that helps to make a convincing argument?
    - A short proof or explanation: In bouncy wall mode, all mice end up in separate mouse holes (all 16 are exit points for one mouse).
    - A short proof or explanation: In bouncy wall mode, no mouse ends up back in its own hole. Why is that not possible?
    - A description of why some mice may never emerge in arrow mode.

Classroom tips

- If a student gets frustrated:
  - Suggest they focus on the **Probe Log** patterns and consider restarting their model from scratch.
  - Temporarily switch them back to **Introduction mode** to reinforce mechanics.
- Make it social:
  - Pair students with different roles:
    - “Mouse launcher” (chooses probes),
    - “Recorder” (updates a paper log or builds a graph or other diagram),
    - “Model builder” (clicks the grid).

Instructions for Students (Not needed if there is a classroom demonstration)

The Game's Layout

- **Top controls:**
  - **Mode**: Introduction / Inference
  - **Maze Type**: Bouncy Walls / Arrows
  - **Level**: Beginner / Intermediate / Expert
  - **Start New Game**: creates a new maze with your current settings.
- **Left side:**
  - **Square house** drawn as a 4×4 grid of rooms.
  - **Numbered mouse holes (1–16)** arranged around the outside walls.
  - In Introduction mode, inside the house you'll see:
    - **Diagonal blue lines** = bouncy walls (/ or \) or
    - **Blue arrows**
  - In Inference mode, you see only your **own model** (your guesses).
- **Right side:**
  - **Prompt box** – tells you what question or task you're on.
  - **Feedback** – how your last answer or test went.
  - **Score** (Introduction mode) – how many correct out of 8.
  - **Inference panel** (Inference mode):
    - **Test My Model** button
    - **Probe Log** – records what happened to each mouse you launched.
- **Button under the board (Introduction mode only):**
  "The Mouse Doesn't Come Out!" – use this when you believe the mouse gets trapped forever inside the house.

Introduction Mode

What you're doing

You're practicing tracing mice through the house when you can **see** the obstacles. You'll always get **8 questions** about the **same** maze.

Step-by-step

1. At the top, choose:
   - Mode: **Introduction**
   - Maze Type: **Bouncy Walls** or **Arrows**
   - Level: **Beginner / Intermediate / Expert**
2. Click **Start New Game**.
3. The board shows a maze and the prompt on the right shows something like:
   - "Question 3 of 8"
   - In **bold color**:
     - **"A mouse leaves hole 13."**
       or
     - **"A mouse ends up in hole 7."**
4. Read the question carefully:
   - **Forward question (common in all levels):**
     - Example: **"A mouse leaves hole 13. Where will it come out?"**
     - You answer by clicking the **hole** where you think the mouse will **emerge**.

        - In arrow mode, the text also reminds you that you may use **“The Mouse Doesn’t Come Out!”** if you think the mouse is stuck.
    - **Reverse question – Bouncy Walls only (Intermediate & Expert):**
        - Example: **“A mouse ends up in hole 7. From which hole did it start?”**
        - You click the hole you think was the **starting** hole.
    - **Reverse question – Arrows (Intermediate & Expert):**
        - Example: **“A mouse ends up in hole 10. Click on one hole that could have been its starting hole.”**
        - There might be **more than one** correct starting hole; any one valid starting hole counts as correct.
    - **Trapped mouse (Arrows only):**
        - If you think the arrow arrangement makes the mouse circulate forever and **never** return to any hole, choose the **“The Mouse Doesn’t Come Out!”** button.
5. After you answer:
    - You see an animation showing the mouse’s **path inside the house**.
    - You get feedback:
        - If you’re right: a positive message.
        - If you’re wrong: an “Oops — ...” message that tells you what actually happens.
6. Your **score** (correct answers out of 8) is shown and updated after each question.
7. After 8 questions, you’ll see a final message:
    - “You made N out of 8 correct predictions.”

You can then:

- Change mode, maze type, or level and click **Start New Game** again;
- Or just launch mice from any holes to explore how the maze works (even after the question set is done).

Inference Mode

What you’re doing

Now you **can’t see** the true internal layout. Your job is to **infer** where the bouncy walls or arrows might be, using experiments and logic.

You have:

- A hidden **real maze** (used by the program).
- Your own **model maze**, which you build on top of the grid.
- A **probe log** recording what happens when you send in mice.
- A **Test My Model** button to check how well your model matches the real maze.

Step-by-step

1. Set:
    - Mode: **Inference**
    - Maze Type: **Bouncy Walls** or **Arrows**

   - Level: **Beginner / Intermediate / Expert**
2. Click **Start New Game**.
   - A **new hidden maze** is created.
   - The main grid is blank (no obstacles).
   - The Probe Log is cleared.
   - Your **probe count** is reset (0 mice sent).
3. **Send in mice (probes)**
   - Click any **hole button** (1–16) around the house.
   - The program:
     - Simulates what the hidden maze does to that mouse.
     - Shows an animation **outside the house only**:
       - A larger mouse marker starts on top of the hole button,
       - moves slowly to the wall,
       - disappears into the house,
       - and (if it eventually emerges) appears on the exit wall and continues to the exit button.
     - If the mouse is trapped in a loop:
       - You'll only see the entry segment. It never reappears.
   - The **feedback** text explains:
     - "The mouse from hole *X* came out of hole *Y*."
       or
     - "The mouse from hole *X* is stuck in the house forever."
   - The **Probe Log** adds a line:
     - From 3 → 11
     - or From 6 → (stuck inside)
   - A **probe counter** tracks how many mice you've sent in this game.
4. **Build your model of the maze**
   - Click inside **rooms** (squares) of the grid to cycle your model through the different detour options.
   - The maze drawn on the board is **your model**, not the hidden original.
   - Try to make your model consistent with all the data in the Probe Log.
5. **Test My Model**
   - When you think your model might work, click **Test My Model**.
   - The program:
     - Simulates all 16 starting holes with the **real** maze and your **model**.
     - Counts how many behave exactly the same:
       - Same exit hole, or
       - Both trapped forever.
   - It then reports:
     - How many are correct:
       - "*N* of your mice arrive in the right place for that model."
     - Whether your model:
       - Is **exactly** the same as the hidden layout, or
       - Produces exactly the same behavior but looks different (another valid solution) in which case it will offer to show you its model.
     - And also:
       - **"You have sent *M* mice into the maze."**
         so you can see how many probes you needed.
       -

6. You can now:
   - Refine your model (change some walls/arrows) and **Test My Model** again.
   - Keep using more probes (the probe count keeps increasing each time you launch a mouse).
   - Or click **Start New Game** to reset with a new hidden maze and reset the probe count.

Making Math

Developing Student Mathematicians