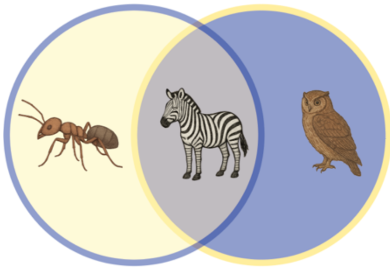


Zoe's Zoo

A game of logic and problem solving

Welcome to Zoe's Zoo! You can play this game at makingmath.org/zoeszoo. This game is designed to provide:

- Problem solving experiences
- Learning about the logic operations And, Or, and Not, which are commonly used in conditional statements (including definitions and theorems) in mathematics
- Learning about the binary operators, which for computer science help with constructing more complex conditional (if-then) statements and which, when modeled with transistors, are central to how computers perform calculations and make decisions at their most basic electrical level.
- Learning some interesting information about animals!

Our setting: Zoe runs a small zoo. She has many animals, but there isn't room for them at the zoo every day. Each day, they either hang out at the retreat or visit the zoo. Your job is to make sure that the animals whose turn it is to be at the zoo get selected from a larger group. You will make that selection using sensors and logic gates to identify the desired animals. Each sensor reacts to one of the characteristics of the animals:

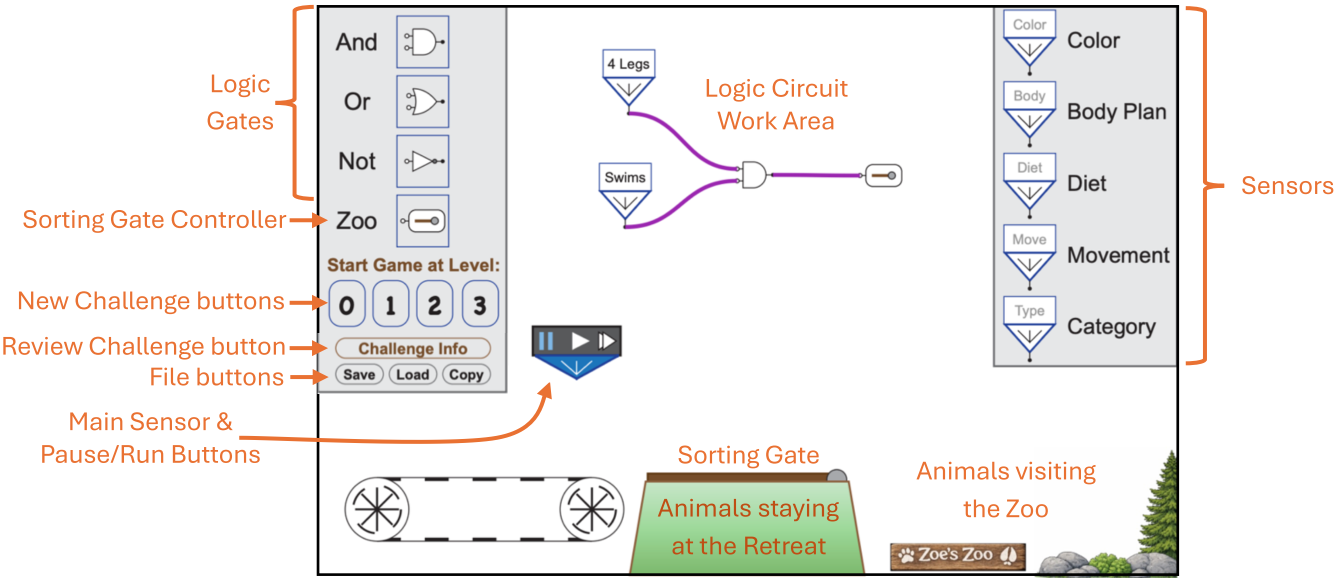
- Their main **color**(s): Black, Blue, Brown, Gray, Green, Orange, Pink, Red, White, Yellow, and Striped (oh, ok, that is not a color, but this pattern is also an option).
- Their **body plan**: 2 Legs, 4 Legs, or Radial. A radial shape looks the same even after you spin it a certain amount. For example, the flower at right appears unchanged if rotated 60°, so it has *radial symmetry*. Animals with legs typically have reflection or *bilateral symmetry*: they have two halves that look essentially the same if flipped over a center line, as seen in the zebra here.

- Their **diet**: Carnivore (meat-eating), Herbivore (plant-eating), and Omnivore (omni- means "all", so an animal eats a mix of meat and plant material).
- Some of the ways they can **move**, with only swimming and flying as options.
- The type of animal they are within the animal kingdom: Amphibian, Bird, Crustacean, Fish, Insect, Mammal, Mollusk, Reptile, and Vertebrate (has a spine) and Invertebrate (lacks a spine).

Note: not every color, diet, or movement is an option for sensing. For example, one of the animals is a detritivore, which means it eats detritus (which is the decayed organic matter from past plants and animals in the soil), but that is not an option for the diet sensor.

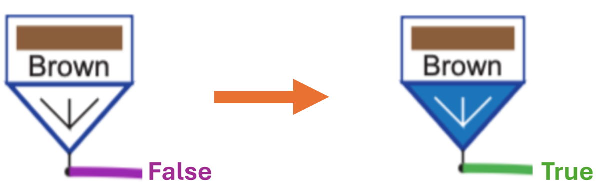
All sensors look like the one shown here. To choose which version of a sensor you want (for example, Yellow, if you have a Color sensor, or carnivore, if you have a Diet sensor), click on the rectangular region. To move the sensor in the work area (see Main Screen below), click on the triangular region and drag it to where you want it.

The Main Screen



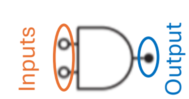
Here are the components of the main screen:

Sensors: the sensors, as described above, can be dragged from the panel on the right. The property they sense is chosen from a window when the rectangular region is clicked (see the Diet window at right as an example). When an animal has a property that matches what the sensor is looking for, the triangular portion of the sensor turns blue (the exact shade is "true blue"). Additionally, the wire leaving the sensor, which was purple showing that it was in off or *False* mode because the sensor was not sensing a brown animal, turns green to show it is on and is in *True* mode – it is now sensing a brown animal.



To remove a sensor, drag it to either side panel.

Wires: the purple connectors seen above in the work area connect sensors to logic gates and logic gates to the Sorting Gate Controller. When no logic gates are needed (for example, if you are only separating out animals that swim from animals that don't), a sensor will attach with a single wire directly to the Sorting Gate Controller. All information in a logic circuit is either *True* or *False* (or yes or no) and each wire passes along that information. The wires are one-directional: they pass that information from an output attachment point (the closed circles on the sensors and gates) to an input attachment point (the

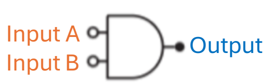


open circles). Wires can be created by clicking on attachment point and dragging to a point or the opposite type. Each wire must connect an output to an input and no cycles or loops may be created. To remove a wire, click on either end of it.

The Sorting Gate Controller: This device sends its output wirelessly to the Sorting Gate. All circuits are designed to tell the controller whether to send an animal to the zoo by staying flat as shown in the controller or to open so that an animal will return to the retreat for the day. If the input to the controller is *False* (purple), the gate will swing open. If the input to the controller is *True* (green), the gate will stay closed so that the animal can reach the zoo. You will always need to put a controller into each circuit. Only one controller is allowed so that the Sorting Gate doesn't get conflicting messages about whether or not to open.

Logic Gates: The logic gates, And, Or, and Not, help you make distinctions between the animals. They can be dragged from the panel on the left into your working area. Gates can be removed by dragging them back to a panel on either side. Gates can get input from sensors or other gates and they output their status (*True* or *False*) to other gates or to the controller.

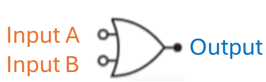
- The **And** gate has two inputs. If either of the inputs is *False*, it will output *False*. If both inputs are *True*, the gate will output *True*. So they have to be *True* **and** *True* to output *True*. When studying logic, we often list all possibilities in a logic table using *True* and *False* or 1 (for *True*) and 0 (for *False*). Here are the logic tables for the And gate:



Input A	Input B	Output
F	F	F
F	T	F
T	F	F
T	T	T

Input A	Input B	Output
0	0	0
0	1	0
1	0	0
1	1	1

- The **Or** gate has two inputs. If either of the inputs is *True*, it will output *True*. If both inputs are *False*, the gate will output *False*. So they have to be *True* **or** *True* (or both, since in logic Or includes And) to output *True*. Here are the logic tables for the Or gate:



Input A	Input B	Output
F	F	F
F	T	T
T	F	T
T	T	T

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	1

- The **Not** gate has one input, and it reverses its value. If the input is *True*, the gate will output *False*. If the input is *False*, it will output *True*. It outputs the opposite (e.g., not true *is* false). Here are the logic tables for the Not gate:

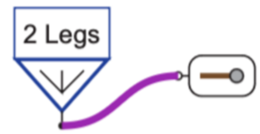
Input	Output
F	T
T	F

Input	Output
0	1
1	0

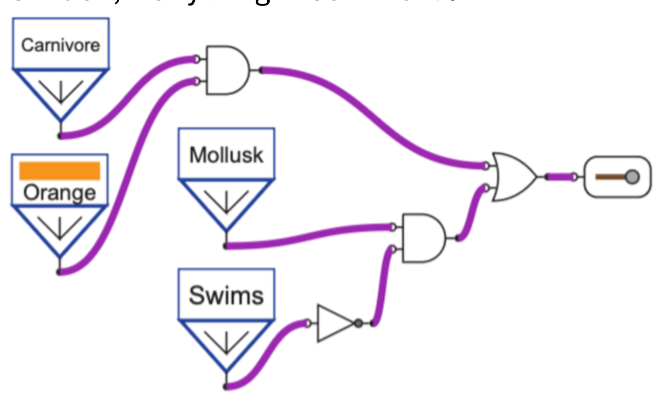
Ways to Use the Gates: The And gate is useful when we want to limit a selection to an animal with two required characteristics. If we want more than two requirements, we can link more than one And gate. The Or gate is useful when we want to include more than one type of animal, so it is typically used when we want to include animals without shared properties (or at least ones that are also not shared with retreat-bound animals). The Not gate is useful when the characteristic we want is not available in a sensor or when the retreat animals have a shared characteristic we want to avoid (e.g., if all of the retreat animals have 2 legs and the zoo animals have 0, 4, and more legs, then Not (2 Legs) serves the purpose).

The Challenge Buttons: The four buttons labeled 0 through 3 start a new challenge with a new selection of animals. The animals that should be selected for the Zoo are highlighted in green.

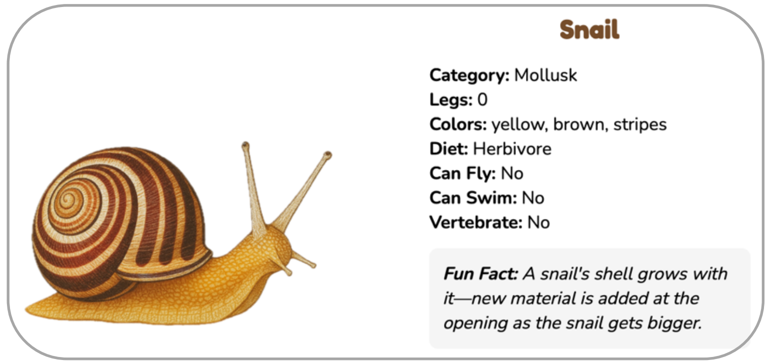
- Level 0 challenges do not require any of the logic gates. They can be solved using just a sensor and the controller. This means there is some characteristic (color, diet, etc.) that the zoo-bound animals share that is not true for the animals that will spend their day at the retreat. At right is an example of a Level 0 solution.



- Level 1 challenges can be solved with a single logic gate. When the Zoes Zoo program is first launched, it will give instructions for the first three level 1 challenges – it will, for example, have a note that the And gate can be used to solve the problem. After the first three Level 1 challenges, students will need to figure out which gates will help them solve the problem.
- Level 2 and 3 challenges involve more than one gate and get progressively harder. Sometimes, the zoo-bound animals do not have much, if anything in common. In these cases, solutions might involve an And gate to make sure a specific animal is targeted fully and an Or gate to connect to a completely different set of criteria for another animal. For example, to include the snail below and a tiger, we might use the circuit at right, which selects for an orange carnivore or a non-swimming mollusk. This will work so long as none of the animals we don't want at the zoo share these features. The animal category invertebrate ("not vertebrate") is not available at these levels, but can be achieved using the vertebrate category.



When a challenge is presented with the pictures of the animals, students can **mouse over the animal and the cursor will be become a question mark** (?). They can then click on that animal to get an **information window that will list the animal's characteristics** according to the five sensors as well as an interesting fact about the animal. This is a very helpful tool for finding common characteristics between the zoo-bound animals and distinct characteristics from the animals staying at the retreat. When a student is testing out their logic circuit (see below), they can always pause the test and mouse over an animal for the question mark cursor and an information window. Any animal shown on the main screen can be investigated.



For some challenges, certain sensors may be disabled (they are "broken" for the day). This limitation means students have to use different sensors and learn about different aspects of the animals.

Typically, the challenges will have multiple solutions, some less complicated than others. It is possible that a Level 2 or 3 challenge can be solved with only a single gate.

The Challenge Info Button: At any point in a challenge, the animal involved can be reviewed by clinking here.

The Pause/Run Buttons: To test out a logic circuit, click on the white "play" triangle button. The animals will start appearing on the conveyor belt and will be scanned by the main sensor one at a time. To make the test run more quickly, click on the double triangle "speed" button. The speed button is not a play button; it is a switch that can be turned on and off that adjusts how fast the animals will move through the process. When a test is running, as an animal passes below the main sensor, special sensing rays (of uncertain scientific meaning :-) will flash and the sensors and wires in the logic circuit will show where true values are in the circuit. These results can be studied more closely by hitting the pause button at any point. The space bar can also be pressed to pause or restart a test.

The Save, Load, and Copy Buttons: The Save button will save the current challenge so that a student or teacher can share it with one or more students to see how many different approaches arise or to preserve a particularly interesting challenge. Saved files will have the .zoo suffix, but, because the program runs in a browser, clicking on the file will not launch Zoes Zoo. All .zoo files must be used through the Load button. Save does not preserve any circuit or solution, just the challenge. The copy button will place a graphic with both the challenge and any logic circuit in the clipboard so that students can document their solutions that way.