Human Logic Gate: Object Filter (3–5 learners per team)

Type: Icebreaker (15–30 mins)

Overview

Learners form a living **logic gate** to sort objects. Objects are simple shape–colour cards (red/green circles/triangles). Inputs check one feature (colour or shape). Operators apply a rule (AND, OR, NOT). The output announces the decision loudly. The whole team works together to sort objects quickly. However, if anyone makes a mistake, the object may be sorted wrongly, just like errors in real systems.

This activity shows how simple yes/no rules can be chained to filter information, just like inside a computer.

Key concepts

- **AND:** Both conditions must be true to pass
- **OR:** Either condition being true is enough to pass
- **NOT:** Flips a single condition (yes → no, no → yes)
- **Sorting:** One gate alone isn't enough: but combining them lets you separate all 4 objects

Materials

- **Minimal:** Paper + pencils/crayons (red and green). Each learner draws red/green circles/triangles: three of each type.
- **Optional:** Printable set of the 4 objects (3 copies of each).
- **Optional:** role cards for inputs, operators, outputs.

Roles

- **Input A** – checks for *colour* (e.g. "Is it red?")
- **Input B** – checks for *shape* (e.g. "Is it a circle?")
- **Operators (1–2 learners)** – listen to the inputs, apply the gate's rule, and agree on the decision. Must announce it loudly together.
- **Output** – repeats the operators' decision loudly and sorts the object into the correct pile.

How to play

1. Learners form **gate teams** of 3–5. Each team represents 1 gate.
2. **Assign conditions**: Input A checks colour, input B checks shape.
3. The facilitator (or a learner) presents an object card (e.g. red circle).
4. **Inputs respond**:
    - If the object matches their condition, they raise their hands and shout **"YES!"**
    - If not, they keep hands down and shout **"NO!"**
5. **Operators listen** to both inputs, apply the rule (AND, OR, NOT), and loudly announce the result: **"YES!"** or **"NO!"**
    - If two operators are present, they must reach **consensus** before announcing.
6. The **output** repeats the operators' decision loudly and sorts the card into the **YES pile** or **NO pile**.
7. Continue until all objects are sorted.
8. **Challenge round**: Time how fast the team can sort all objects. Mistakes count against their time — if an object is placed wrongly, they must stop and correct it before finishing.

Leader tips: Sorting the objects

One gate alone can *never* sort all four. What happens with only 1 gate:

- **AND (red AND circle):** Only lets through red circles. Everything else is lumped together.
- **OR (red OR circle):** Lets through red circles **plus** extras (red triangles, green circles).
- **NOT (NOT red):** Splits by colour (greens vs reds) but ignores shape. (Or vice-versa.)

Sorting the objects fully always requires 3 gates.

Three NOT gates:

1. First split by **colour** (red vs NOT red).
2. Then split your red group by **shape** (circle vs NOT circle).
3. Then split your green group by **shape** (circle vs NOT circle).

Result: all 4objects in their correct piles.

Other ways creators may try

- **3x AND:** Isolate individual object types one at a time. Red AND Circle, then Red AND Triangle, etc.
- **OR then AND:** Works eventually but looks messy to start, as you will have some green and red mixed after the first sort. Sort again with a NOT, or a specific AND to split the colours properly. Then another NOT or AND.

Leader tips:

- **Praise working solutions** even if they're not optimal: "You found red circles! Nice."
- **Challenge them to refine:** "But can you get *all 4 types* sorted into piles?"
- **Spot teachable moments:**
  - OR is too loose (false positives).
  - AND is strict but incomplete.
  - NOT is useful for exclusion, but not enough on its own.
- **Push toward the two-phase solution if they are stuck**: "What if we first separate by colour, then by shape?"

Core lesson

"One simple check isn't enough, but combining two gives you a complete sort. Computers work the same way: millions of tiny YES/NO checks chained together to do complex tasks."

Variations

- **Sorting race:** Two gate teams compete to sort the same set of objects. The fastest correct team wins.
- **Circuit chain:** Connect two gates in sequence. An object must pass through the first (YES) before being tested again in the second.
- **Error injection:** The facilitator deliberately miscalls an object's feature once: how does the system handle an error?