

Nuclear Avoidant Non-Detonators (NAND)

Grading on a curve has never been more explosive!

Authors: Kate Lyman, Maya Burke, Andrew Chung

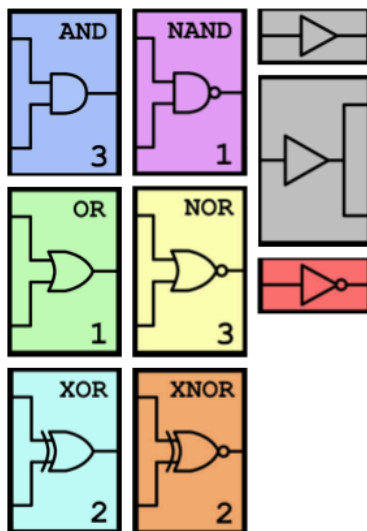
Description: This turn-based card game is about competitively diffusing bombs in a bomb-defusal academy. The players are completing their final exam to get their Nuclear Avoidant Non-Detonator license while screwing over their rival students.

of Players: 3-4 players

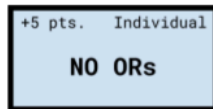
Duration of Game: 20-30 minutes

Audiences: Middle schoolers & early high-schoolers. Beginner programmers.

List of Materials:



Logic Gate Cards
(cut into above pieces)



Goal Cards
(cut into above pieces)

A	B	AND	NAND	OR	NOR	XOR	XNOR	NOT	BUFFER	SPLITTER
0	0	0	1	0	1	0	1			
0	1	0	1	1	0	1	0			
1	0	0	1	1	0	1	0			
1	1	1	0	0	1	0	1			
0								1	0	0
1								0	1	1

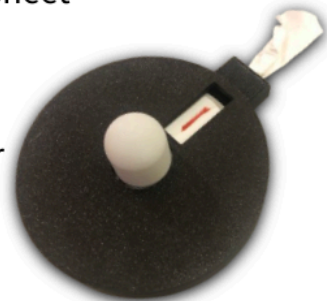
IN YOUR TURN, TAKE ANY TWO OF THE FOLLOWING ACTIONS:
1. **PLAY:** Place one logic gate card from your hand on your system.
2. **NOT:** Place one NOT card from the NOT deck on your system.
3. **SABOTAGE:** Flip one of your own history tokens to flip an opponent's token.
4. **DRAW:** Take one card from the main logic gate deck to be in your hand.
5. **RESCUE:** Discard your entire hand. Draw the same number of cards from the main logic gate deck.
6. **EQUIVALENCE:** Use at least one card in your hand, the NOT deck, and/or the splitter deck to replace at least one card in your system that is logically equivalent to the used cards. "Bank" the replaced card(s) on the side.
In your turn, you may use as many buffer and splitter cards as you want without costing an action.

Cheat Sheet



Binary Tokens
(fold across the middle)

Round Counter
(make your own!)

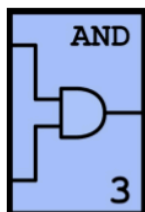


What are Logic Gates?:

A logic gate is a circuit that takes in one or more values (called inputs) and gives one result (called an output). These inputs and outputs are either 1 or 0. For example, an AND gate has an output of 1 if both inputs are 1, and an output of 0 if that is not the case.

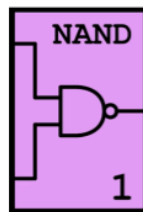
Each logic gate will output different values depending on the inputs. The separate cheat sheet is a reference for what each logic gate does for every possible pair of inputs.

Each Logic Gate Explained:



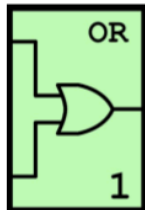
AND Logic Gate:

- Checks if inputs A and B are both 1.
- If so, output 1. If not, output 0.
- Worth 3 points for having 3 outputs of 0 out of all 4 possible scenarios.



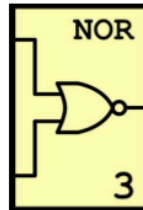
NAND (NOT AND) Logic Gate:

- Checks if inputs A and B are NOT both 1.
- If so, output 1. If not, output 0.
- Worth 1 point for having 1 output of 0 out of all 4 possible scenarios.
- The opposite of AND.



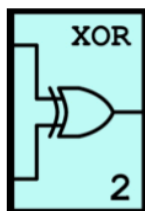
OR Logic Gate:

- Checks if at least one input out of both inputs A and B are 1.
- If so, output 1. If not, output 0.
- Worth 1 point for having 1 output of 0 out of all 4 possible scenarios.



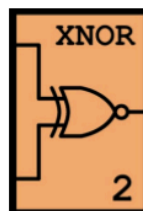
NOR (NOT OR) Logic Gate:

- Checks if none of inputs A and B are 1.
- If so, output 1. If not, output 0.
- Worth 3 points for having 3 outputs of 0 out of all 4 possible scenarios.
- The opposite of OR.



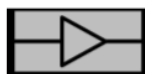
XOR (Exclusive OR) Logic Gate:

- Checks if only one input out of both inputs A and B is 1.
- If so, output 1. If not, output 0.
- Worth 2 points for having 2 outputs of 0 out of all 4 possible scenarios.



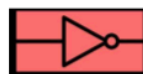
XNOR (Exclusive NOR) Logic Gate:

- Checks if inputs A and B are both only 0 or both only 1.
- If so, output 1. If not, output 0.
- Worth 2 points for having 2 outputs of 0 out of all 4 possible scenarios.
- The opposite of XOR.



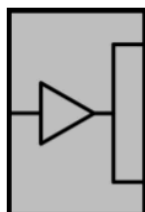
Buffer Logic Gate:

- Outputs the same value as the gate's one input.
- Worth 0 points.
- In this game, this only serves as a visual connection between your layout's materials.



NOT Logic Gate:

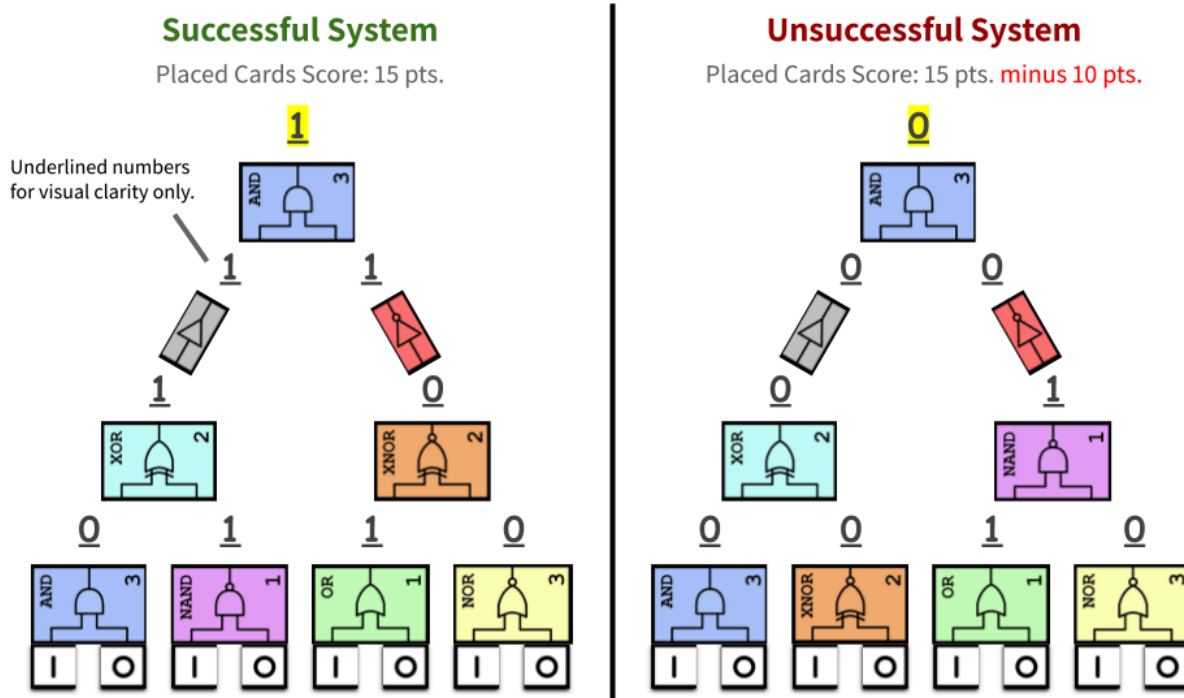
- Outputs the opposite value as the gate's one input.
- Worth 0 points.



Splitter Logic Gate:

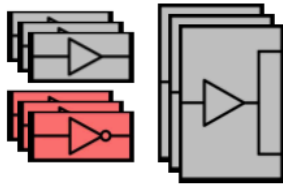
- Produces two outputs that are the same value as the gate's one input.
- Worth 0 points.

Goal: Defuse your bomb to pass the exam: this means that when you submit your individual system of your logic gates, its final output (given all eight binary tokens) should be a value of 1 to pass. If the final output is not a 1... BOOM! (deduct 10 points). All will have their points be tallied up based on what logic gate cards they played, what logic gate cards they "banked", and what individual and group goals they satisfied. The player with the highest score wins and earns their Nuclear Avoidant Non-Detonator License!

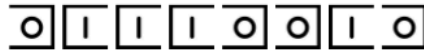


Setup:

Select at least one person to handle the setup.

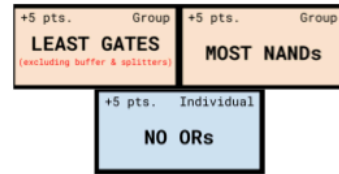


The buffer, splitter, and NOT cards are in separate decks.



(example)

Deal 8 binary tokens to each player. Shuffle them and line them up.

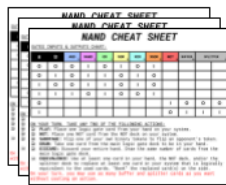


(example)

Randomly reveal 2 group goals and deal 1 individual goal to each player.



Shuffle the deck with the main logic gate cards. Deal 3 cards to each player.



Each player gets a cheat sheet for reference.

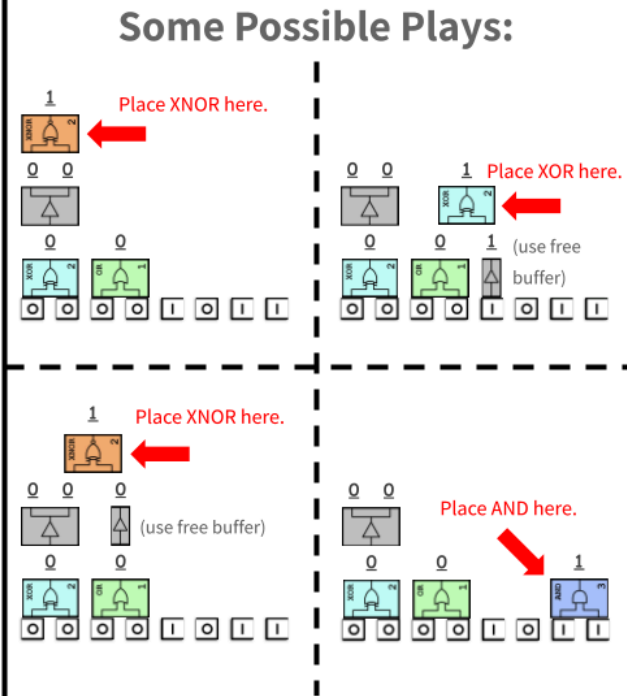
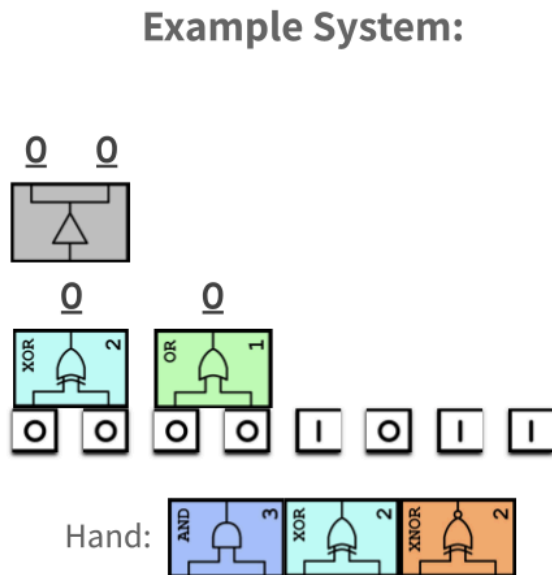


The youngest player's turn marks Round 1. The next player is determined clockwise. The youngest player's new turn marks a new round. Track the round number with the counter. The game ends after Round 10.

On Your Turn:

During each player's turn, they will take any **two** of the following **actions**. *They can also draw and place any number of buffer and splitter cards on their system without spending an action.*

- **PLAY:** Play a main logic gate card from your hand onto your system. A valid placement takes exactly two inputs, and has nothing yet connected to the card's output wire. An input can be a binary token or the output of another logic gate that you have already played. Once you have played a logic gate card, it cannot be moved and its inputs cannot be changed, except through an **EQUIVALENCE** action.



- **NOT:** Take a NOT logic gate card from the NOT deck and place it on your system. A valid placement takes exactly one input, and has nothing yet connected to the card's output wire. An input can be a binary token or the output of another logic gate that you already played. Once you have played a NOT card, it cannot be moved and its input cannot be changed, except through an **EQUIVALENCE** action.
- **SABOTAGE:** Flip one of your own binary tokens to flip any one of another player's binary tokens. This action cannot be taken during Round 10 of the game.
- **DRAW:** Draw a card from the main logic gate deck and add it to your hand.
- **DISCARD:** Discard your hand to the bottom of the deck. Draw the same number of cards from the logic gate deck.
- **EQUIVALENCE:** Replace a circuit in your system with a "logically equivalent" circuit made up of cards from your hand, the NOT deck, and/or the splitter deck. The new circuit must be connected to the

same inputs and output as the circuit that you are replacing. The cards you replaced are now “banked”. Set your banked cards to the side of your system. See the next section for more on what “logically equivalent” means.

Defining Logical Equivalence

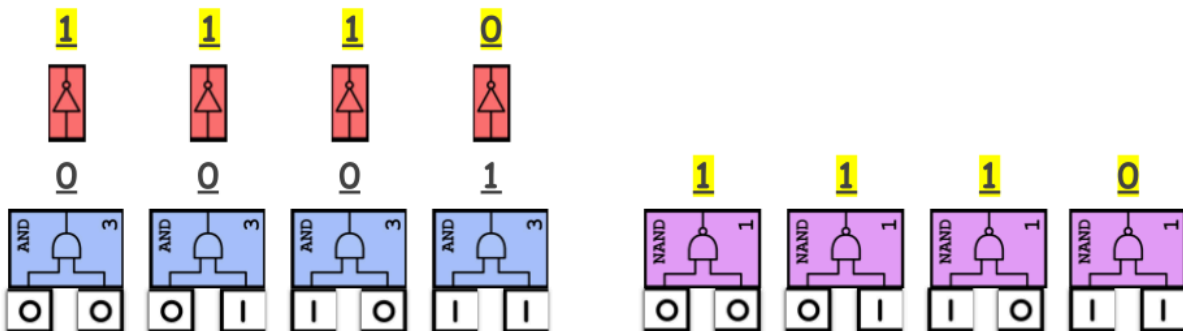
Recall that a “circuit” takes in one or more inputs and gives one output value. In this game, a circuit is at least one logic gate. A circuit can be one logic gate card, multiple logic gate cards that feed into each other, or even your completed system!

One circuit is logically equivalent to another circuit if:

- the two circuits have the same number of inputs, AND
- for every possible combination of inputs, the outputs of the two sets will be the same.

For planning complex exchanges and convincing opponents that an exchange is indeed logically equivalent, it may help to draw a truth table like the one on the cheat sheets and the ones used in the examples below.

- EXAMPLE: an AND gate feeding into a NOT gate is logically equivalent to one NAND gate. This exchange would earn or bank 3 points.



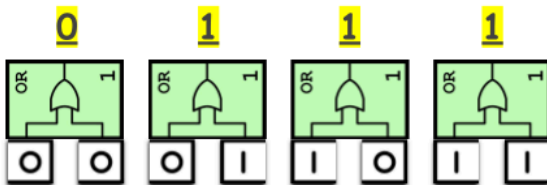
A	B	AND	NOT FOR AND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

=

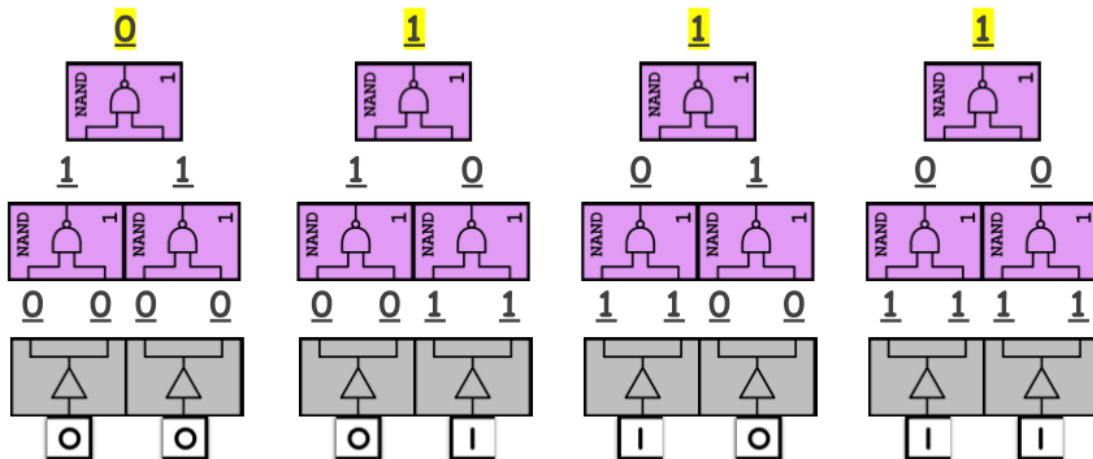
A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0

- EXAMPLE: An OR gate is logically equivalent to these three NAND gates. This exchange would earn or bank 3 points. Fun fact! NAND is a universal logic gate. This means that for any logic gate, there exists a logically equivalent circuit made up of only NAND gates!

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

=


A	B	NAND FOR SPLITTED A	NAND FOR SPLITTED B	NAND FOR BOTH NANDs
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1



Game End:

When Round 10 is over, every student submits their system.

Everyone tallies up their points according to the point values of all of the logic gate cards in their system. Any “banked” cards’ point values also count towards your total score, but no longer count towards any individual and group goals unless specified by the goals themselves. Also add the point value of any individual goals you fulfilled.

Determine as a group which players also fulfilled the group goals. These players add the indicated number of points to their total. In the event of a tie, all players who tie can add the points to their total.

Every player’s submitted system must begin with their initial 8 binary tokens and end with one single final output of 1 to have their bomb be defused. If a player’s exam does not meet this criteria, the bomb explodes! That player loses 10 points. Players who do meet this criteria survive and do not have their points be affected.

The player with the best grade (i.e. the highest total number of points) wins! Players are encouraged to verify each other’s system to ensure everyone’s points were calculated correctly. This can be accomplished in whatever reasonably works, like a player checking the system of the player to their right.

Learning Objectives:

- NAND seeks to teach boolean logic, or boolean algebra. It is a fundamental concept in computer science, used in both software and hardware. The only values in this system are true and false, which are represented respectively by 1 and 0. Boolean algebra uses logical operators, like AND, OR, and NOT.
- NAND's logic gate cards exemplifies the boolean logic needed for players to use to output a value of 1.
- Through the fundamental card-laying mechanic of NAND, players are exposed to the idea of a binary system that uses only two values, and they should become more familiar with logical operators by the end of the game.

Debrief Questions:

Students can discuss the following questions upon completing a game:

- What did you learn about logic gates?
- How did this game make you think about using your logic gates?
- Did you follow a specific strategy playing this game? Did it work and why or why not?
- How can the problem-solving mindset used in this game be similarly used in real-life when it comes to programming and computers?