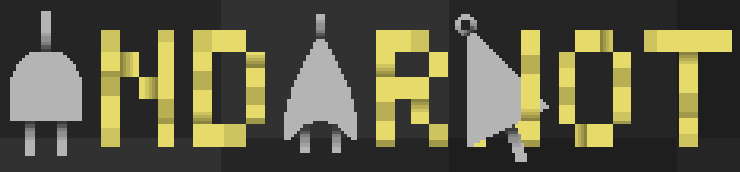


User Guide

Made by the Developers of the Game “ANDORNOT”:

Lagunsad II, Michael Ricardo

Racelis, Sebastian

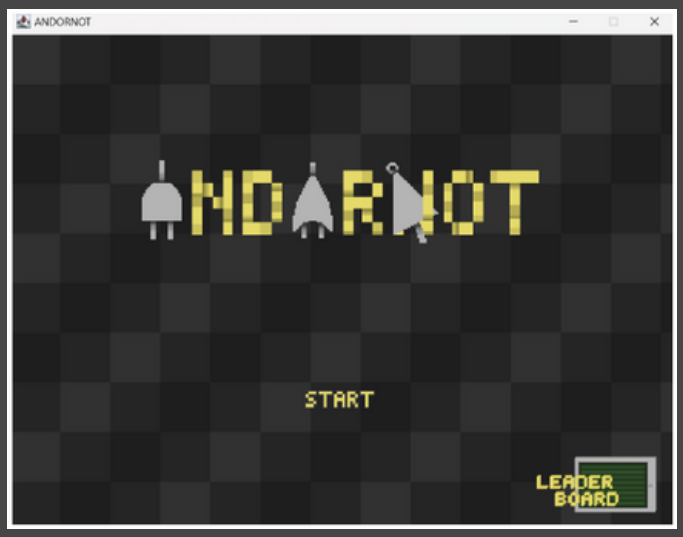
Mendoza, John Paul

Created in 2025

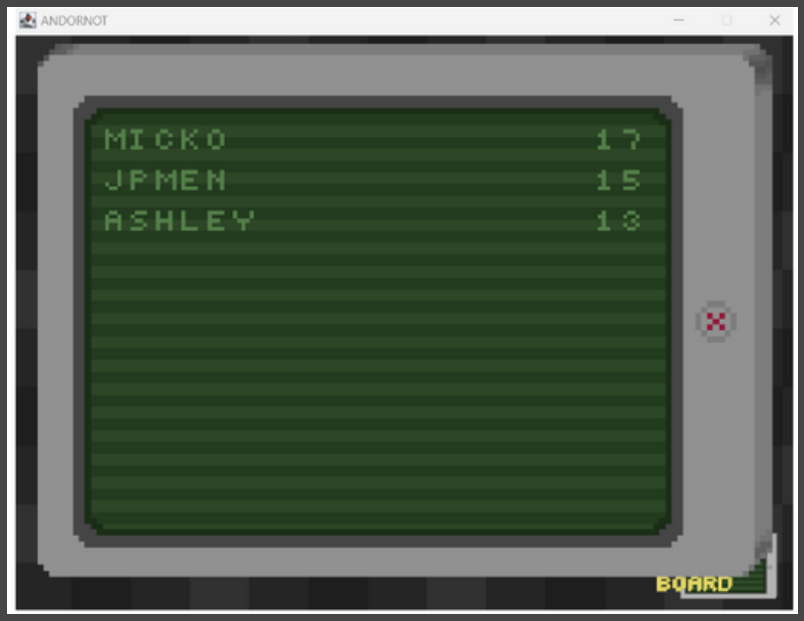
Introduction

ANDORNOT is a puzzle game based on Logic Diagrams, graphical representations of Boolean Arithmetic (Logical Expressions) to make outputs. Drag the right logic gates to the slots and test your guess and understanding of each logic gates' behaviors! Compete with your friends and family with the leaderboard!

User Interface - Title Screen



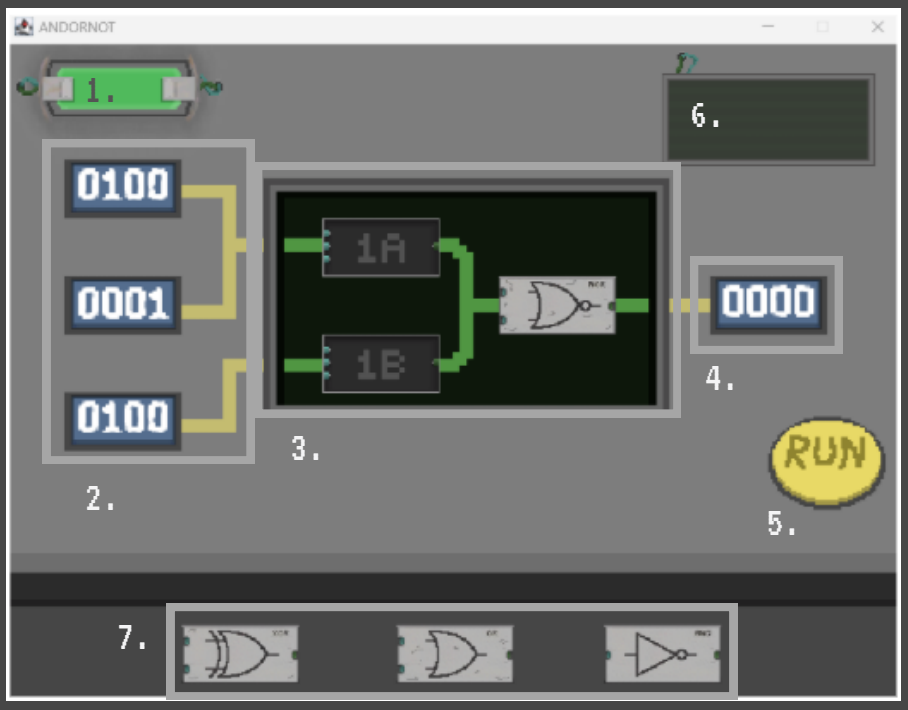
Press "Start" to start the game.



Pressing "Leaderboard" opens the leaderboard tab, showing all the names and their finished levels after multiple times of playing this. Press the red "X" to go back to the Title Screen.

User Interface - Game / Level Screen

Each button and section are numbered.



1.Player Life Counter

There are only three chances to make mistakes. When testing the circuit (6.) and an error occurs, the indicator changes color.

3 Lives (Green) - The default / starting amount

2 Lives (Yellow)

1 Life (Red) - Ends the game after one more error

2.Input Section

This section consists of strings of “1s” and “0s” (known as binary strings). These 4-length strings (known as nibble) are the starting point before changing after running through logic gates.

3.Gate Diagram Section

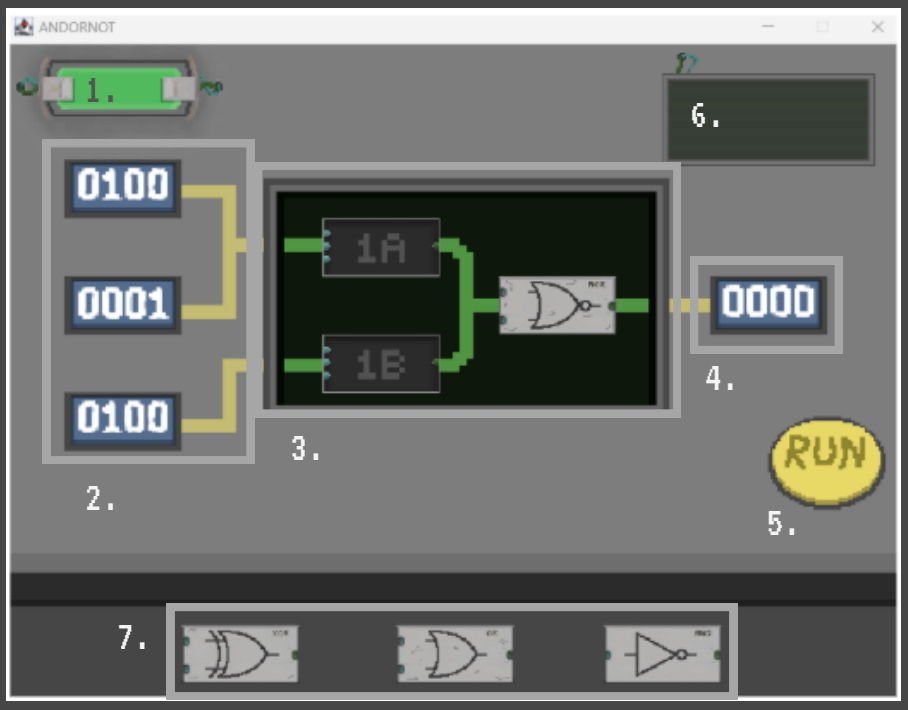
Where the logic gates of (Section) “7.” are placed to make the output on (Section) “4.”. Logic gates already placed there are immovable- use them in the computations for reaching the output.

4.Output Section

Where the desired output is placed. When computing, after reaching the second layer in (Section) “3.”, look at this section to find which logic gates lead to that string.

User Interface - Game / Level Screen

Each button and section are numbered.



5.Run Button - Verifies if the logic gates put on (Section) “3.” are correct. If not, it affects (Sections) “1.” and “6.” The same button creates a new level if the answer is correct.

6.Error Section (Clue)

States which part (or slot) within (Section) “3.” is wrong. However, it can only give a clue to a single slot. Be wise and take time, there is no time limit in checking.

7.Logic Gate Section

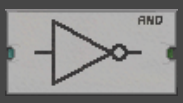
The section where the player picks a logic gate from any of the choices to be dragged onto the slots of (Section) “3.” There are always three choices to pick from. What each logic gate does is on the next page.

Logic Gates

Logic Gates are the fundamental blocks for digital electronics, to form logical operations. Each have a specific logical expression and truth table. For Logic Diagrams, we focus on the truth table.

NOT Gate

Symbol



Logical Expression

$F = x^1$

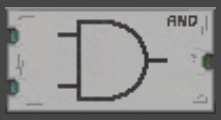
Truth Table

x	F
0	1
1	0

NOT Gates is one of the simplest logic gates. With only one input, it reverses the '1s' and '0s' into the other. If someone asks to "negate" or "invert" an input, NOT gates and logic gates with a similar function is immediately what we want to call.

AND Gate

Symbol



Logical Expression

$F = x \cdot y$

Truth Table

x	y	F
0	0	0
0	1	0
1	0	0
1	1	1

AND gates use two and more inputs with only one condition to keep a "1": Have both inputs be equal to 1. Within Boolean Algebra, this is known as Logical Multiplication. Even in Arithmetic Multiplication, there has been no time that something multiplied by 0 is not 0.

NAND Gate

Symbol

Logical Expression

$F = (xy)^1$

Truth Table

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

Our first logic gate with a NOT gate built within it. With what was stated before that Not gates "invert" the output, what NAND is doing now is getting the output of the AND gate and inverting it. And now, every input having 0 now leads to an output of 1.

OR Gate

Symbol

Logical Expression

$F = x + y$

Truth Table

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1

OR Gates use the same number of inputs as AND gates to function. Within Boolean Algebra, this gate is known for doing Logical Addition. So, whenever there's a 0, as long as there's a single 1, the output will be equal to 1. And no carrying to the next left digit on this folks.

NOR Gate

Symbol

Logical Expression

$F = (x+y)^1$

Truth Table

x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

NOR gates, just like NAND gates, invert the output of OR Gates after its initial output. Therefore, only when all the inputs are "0" that a NOR gate will make a "1".

XOR Gate

Symbol

Logical Expression

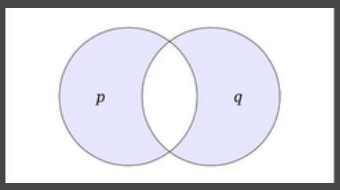
$F = xy^1 + x^1y = x \oplus y$

Truth Table

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

XOR gates are known as Exclusive OR Gates. Why "Exclusive"? XOR gates removes the rule that Or gates can give out "1s" if there are more than one "1". Only when both inputs are the opposites of each other when the output becomes 1.

Tip: This Venn Diagram helps visualize XOR gate's function. This is the Exclusive variant of Logical Disjunction (Or), only allowing the difference between p and q (1 and 0)



XNOR Gate

Symbol

Logical Expression

$F = xy + x^1y^1 = (x \odot y)$

Truth Table

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

And lastly, the XNOR gate inverts the output of an XOR gate. And so, when all inputs at a certain time are 0 or 1, the output will trigger a 1.

User Interface - Game Over Screen

Each button and section are numbered.

1. Retry Button

During game over; resets the game

---

2. Home Button

During game over; redirects player into the title screen. This is the only time when the game asks for the player's name as well; to let them record their personal best without clutter.

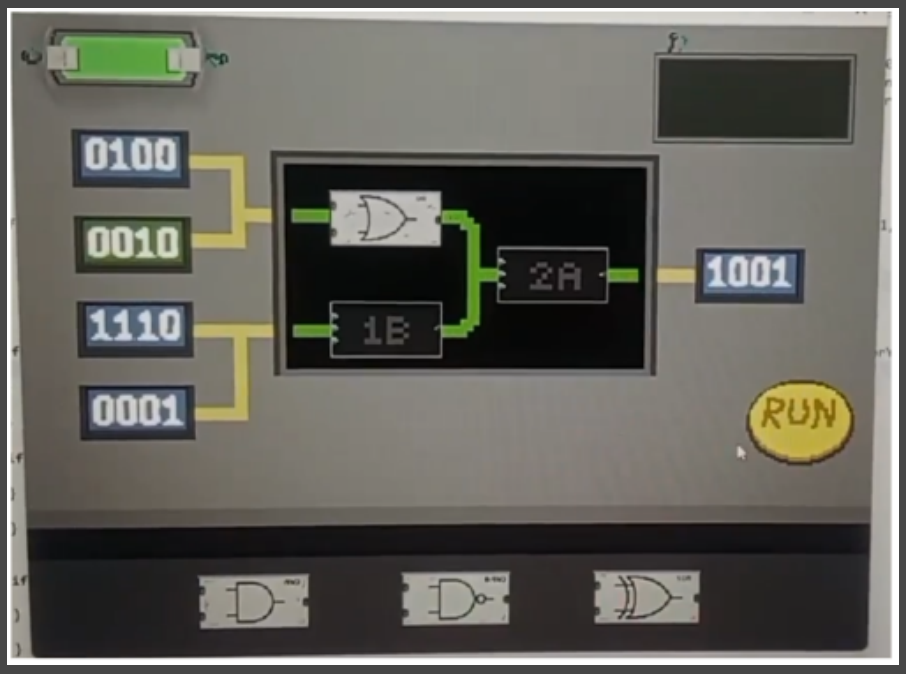
---

3. Total Score

The amount of levels or circuits a player successfully finishes. This is the number that will show up in the leaderboard.

Test Run

Example:



Steps:

1.Make tables of the inputs and separate each 1 and 0 into rows.

Top Inputs	
A	B
0	0
1	0
0	1
0	0

Bottom Inputs	
A	B
0	0
1	0
0	1
0	0

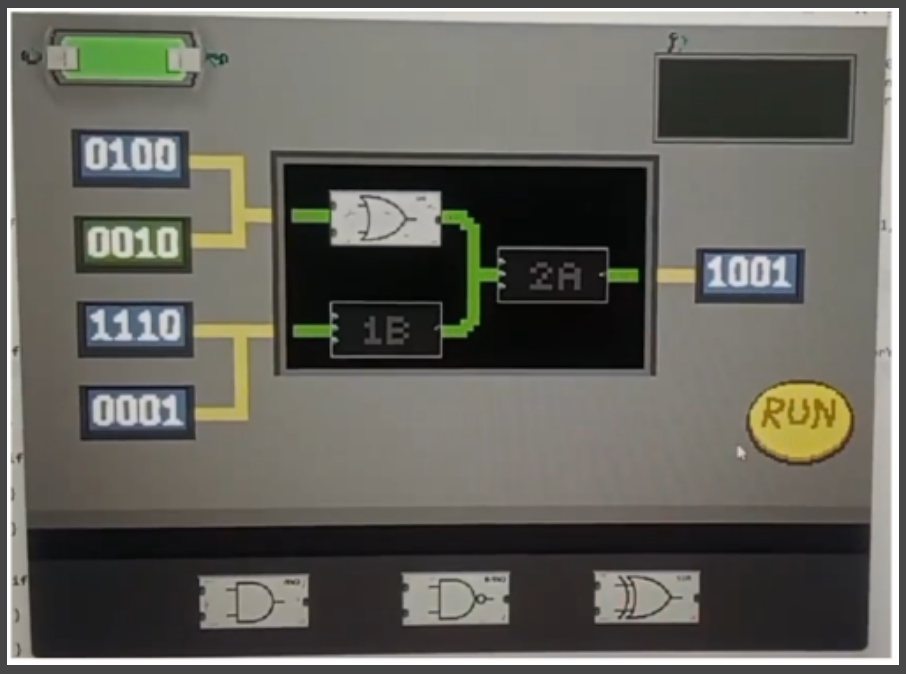
2. Make new column for the result of each possible gate. And then, with the truth table, solve each row

Top Inputs		
A	B	Or Gate
0	0	0
1	0	1
0	1	1
0	0	0

Bottom Inputs				
A	B	And Gate	Nand Gate	XOR Gate
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1
0	1	0	1	1

Test Run

Example:



Steps:

3. Make another table again, combining the result of the given output to the missing outputs. Since the output this time is the answer, that is our target for the answer.

For Output 0000:

Output Layer 1				
A	B	And Gate	Nand Gate	XOR Gate
0	0	0	1	0
1	0	0	1	1
1	0	0	1	1
0	0	0	1	0

For Output 1111:

Output Layer 1				
A	B	And Gate	Nand Gate	XOR Gate
0	1	0	1	1
1	1	1	0	0
1	1	1	0	0
0	1	0	1	1

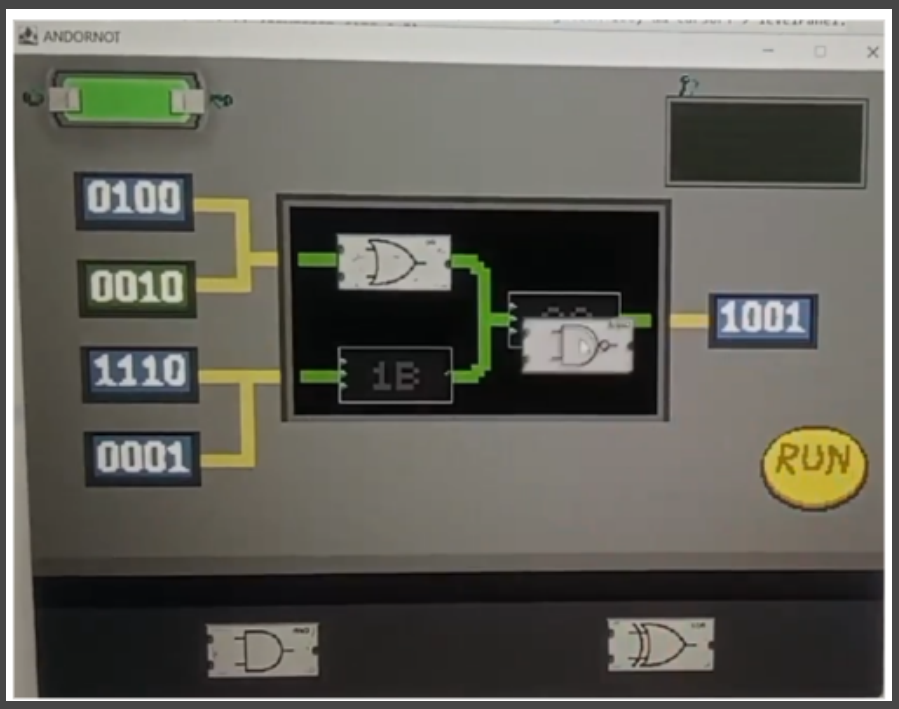
Therefore, our answer is NAND + XOR or XOR + NAND.

Test Run

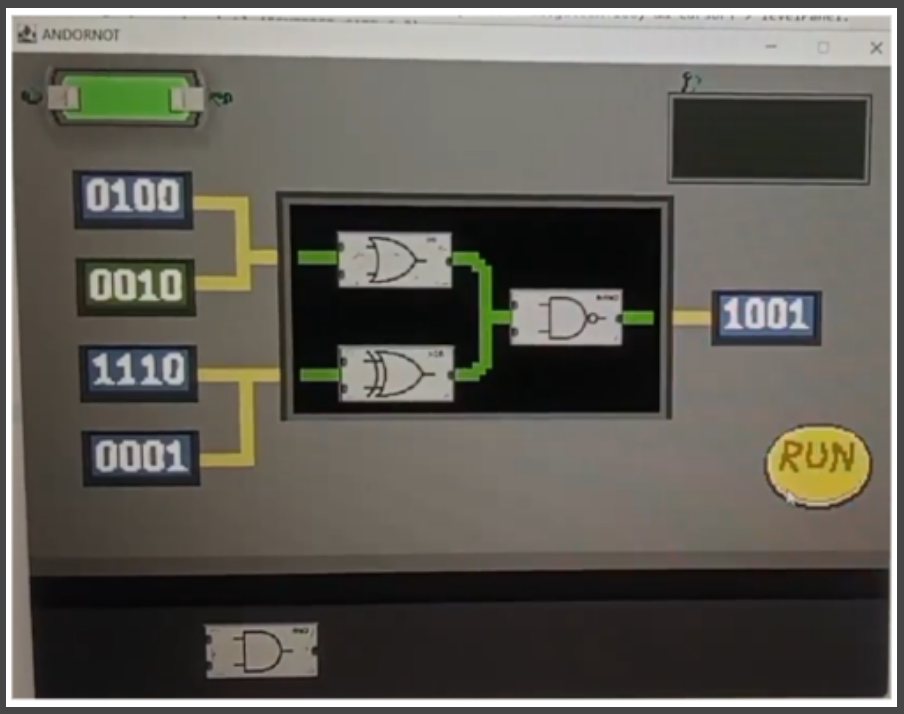
Example:

Steps:

4. Drag the logic gates based on our answer



5. Press Run



6. Repeat until all lives are lost