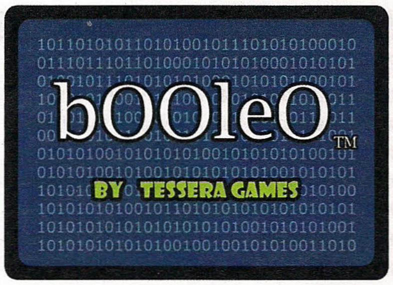


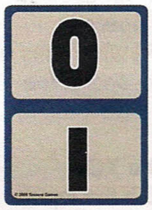
Introduction

bOOleO is a card game which combines strategy card play with the concepts of Boolean Logic. Players must race to resolve an intial binary number to a single specific bit by creating a logic pyramid. This is accomplished through the use of logical gates.

Cards

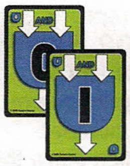
A standard **bOOleO** deck contains 64 total cards.

- 2 - Cheat Sheet Cards
- 6 - Initial Binaries
- 8 - Not Cards
- 48 - Logical Gate Cards

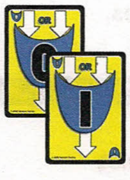


Initial Binary cards are used to determine the binary numbers that form the base upon which the players will apply their Logic Gates.

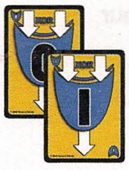
Logic Gate cards are used to resolve two adjacent input bits into a single output bit. There are three different Logic Gate cards in **bOOleO**; AND, OR and XOR. Each of them has the two different output versions, 0 and 1.



AND Gates		
IN 1	IN 2	Can Play
0	0	AND 0
0	1	AND 0
1	0	AND 0
1	1	AND 1



OR Gates		
IN 1	IN 2	Can Play
0	0	OR 0
0	1	OR 1
1	0	OR 1
1	1	OR 1



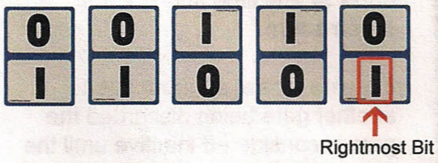
XOR Gates		
IN 1	IN 2	Can Play
0	0	XOR 0
0	1	XOR 1
1	0	XOR 1
1	1	XOR 0

Not cards can be used to change the state of a bit in the initial binary once the game has begun.

Game Setup

First set the initial binaries which will be the base numbers to play from. Shuffle the deck of six binary cards and randomly lay out five or six of them so that the 1 side is facing one player and the 0 side is facing the other player. This will result in a different binary number for each player.

To determine the end result required, each player looks at the rightmost bit in their initial binary. This is the result which the player must resolve to. Keep in mind that each player will have their own required output. Also keep in mind that through the course of the game the required output may change if the initial binary is changed through the play of the Not cards.



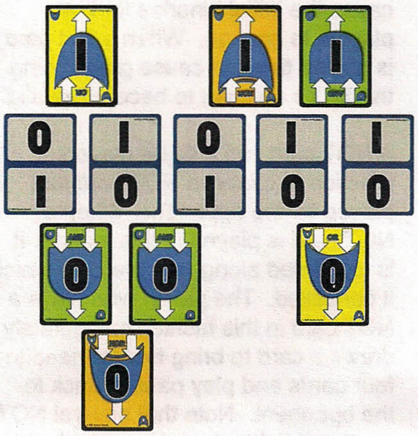
Determine by any convenient means which player will be the dealer and have them shuffle the deck and deal four cards face-down to each player. The non-dealing player then starts the game play.

Game Play

During each turn the player first draws the top card from the deck and then either plays a card or discards a card. Only one card may be played (or discarded) during each turn. At the end of each turn the player should have a hand of four cards. Play then passed to the other player who performs the same two steps. If the deck runs out, reshuffle the discards into a new deck and continue play.

During play the players will each work on completing a tableau of gates with the outputs of the gates forming new binaries. The diagram above and to the right shows a typical game in play. Each new binary (each row is a new binary) will consist of one less bit than the previous binary. In this way, to win the round a player must place fifteen gates (five in the first binary, then four, then three, then two and finally the last one which must resolve to the required output).

To play a Logical Gate the player places the gate face up underneath and between two bits from the previous binary which form the inputs of that gate. When placed the gate must resolve to a valid output.

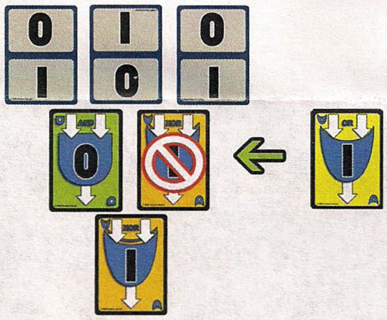


For example if the two bits forming the inputs are 1 and 0 then the following gates are legal to place there – AND(0), OR(1) and XOR(1). If a player places an invalid gate then the offending gate is discarded.

Gates which have been previously placed may be may be replaced by simply playing a new gate in the same position and discarding the original gate. When a gate is replaced in this manner it must resolve to a valid output; this may cause gates below this one to become invalid.

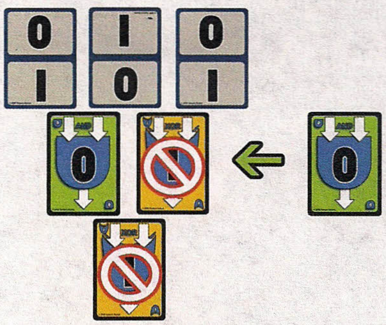
Valid Replacement

(Only the Top XOR 1 must be discarded)



Invalid Replacement

(Both XOR 1s must be discarded)



NOT Usage

To play a Not card the player declares which initial bit is the target and reverses that bit card, the Not card is then removed from play (or placed under the initial binary). This will cause the initial binaries for both players to change. When a Not card is played this can cause gates using this bit as an input to become invalid.

A NOT card can also be played as a reaction to cancel a NOT card just played by the opponent. When a NOT card is played in this manner it is discarded along with the card which it cancelled. The player who plays a NOT card in this fashion immediately draws a card to bring his/her hand to four cards and play passes back to the opponent. Note that several NOT cards could be played in a row in this manner, cancelling each one out in turn.

For example, Amber plays a NOT card to reverse an initial binary bit. Esme plays a NOT card to cancel this action. Both NOT cards are discarded and the initial binary bit is not reversed. Esme then draws a card to bring her hand back up to four cards and play resumes with Amber's next turn.

(Not Card is placed and rotated)

2 Players Ages 10+

What is bOOleO?

bOOleO is a fun, fast-paced strategy game that is based on Boolean logic. Each player races to complete their logic pyramid before their opponent .... while avoiding the dreaded NOT card. You don't know Boolean Logic? No Worries! Play bOOleO and you will learn it, maybe even master it.

Why is bOOleO fun?

bOOleO stretches your brain by using the fundamental concepts of Boolean logic. bOOleO uses four different types of gates; AND, OR, XOR and NOT. You use these gates to strategically build your Logic pyramid, carefully watching the movements of your opponent, while planning your next move to destroy their pyramid. The game play can be lightening fast and the tension can blow your mind.

www.tesseragames.com\booleo

Invalid Gates

When one or more inputs to a gate are changed this may cause the gate to become invalid and be discarded.

For example if an AND(1) gate (inputs of 1 and 1) has one of its inputs changed to 0 then it is no longer valid and must be discarded. Removing a gate will cause any gates which rely on the removed gate for an input to become inactive or invalid. As long a gate has one valid input it is not discarded, but is considered inactive and cannot be used as an input for placing any gates. Any gates which rely on inactive gates for input are also considered inactive. This can cascade down a structure rendering many gates inactive until the missing input can be restored. When restoring a gate any gates which rely on this new gate as an input must be checked to ensure that they are still valid. If any gate with two inputs is ever found to resolve incorrectly it is invalid and must be discarded immediately. This may occur at any time during either player's turn.

If a gate looses an input due to another gate being discarded the gate is considered inactive until the correct input is restored. If the input is restored but this causes the inactive gate to become invalid it is discarded. Inactive gates cannot be used as inputs to place new gates. The rule of thumb is that a gate is only discarded if the inputs do not resolve to the proper output for that gate, or if the gate no longer has any inputs.

Winning the Game

The first player to completely resolve the initial binary down to a single bit wins the round. In order to place the final gate the output value must match the current right-most bit in the initial binary. In a standard two-player game this will require the legal placement of fifteen logical gates.

More About Boolean Logic

Boolean algebra is a subject in math which deals with logic; it is used extensively in the field of computers. It is similar to regular algebra but also different in many ways.

One way that it is different is the values which are allowed. In Boolean algebra all values are logical rather than numeric – that is the values can have one of two states such as 0 and 1 or sometimes the values are stated as either true (1) or false (0).

More About Boolean Logic

This makes Boolean logic ideal for manipulating binary numbers and computer circuits. A binary number is a number which is represented using only the digits 0 and 1 using the base-2 number system. Each place in a binary number is a power of 2 instead of a power of ten like in the decimal number system. Each binary digit is called a bit (the term bit was derived by taking the first two and the last letters of the term binary digit).

Like regular arithmetic Boolean algebra has operators which are used to evaluate expressions. The three base operators in Boolean algebra are AND, OR and NOT. AND is similar to multiplication in regular algebra; therefore if either of the inputs is 0 then the result is 0 and the only way the result is 1 is when both inputs are 1. OR is kind of similar to addition in that if either of the inputs is 1 then the result is 1 and the only way the result is 0 is if both inputs are 0. NOT is simply the negation of the input, if the input is 1 then the result is 0, if the input is 0 then result is 1. In addition there is another derived operator called exclusive or (XOR) which has a result of 1 if one and only one of the inputs is 1.

Another way to look at the XOR operator is the result is 0 if the inputs are the same and the result is 1 if the inputs are different.

A logical gate is a device which performs Boolean operation on one or more inputs and returns a single output, thus making a decision based on the inputs and returning the answer. Inputs and outputs are represented by bits; either a one or a zero. In **bOOleO** three of the operators are represented by gates: AND, OR and XOR. The NOT operator is represented as a special card which can negate one of the bits in the initial binary number. The charts in the next section show the outputs of the various gates given all possible input combinations.

bOOleO on the Web

For optional game play rules and other products offered by Tessera Games, visit our website at:

www.tesseragames.com