# *Computer Algorithms: Tic-Tac-Toe*

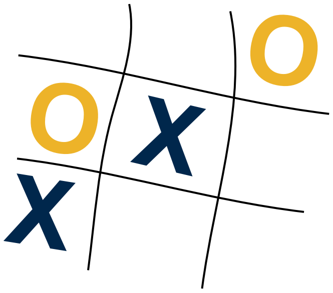
## *Summary:*

In this activity, you'll work to create your very own Tic-Tac-Toe algorithm! An ***algorithm*** is a set of instructions a computer follows to solve a problem. An algorithm is like a recipe you may follow in the kitchen, except instead of baking a cake, you are trying to win a game of Tic-Tac-Toe! After creating your Tic-Tac-Toe algorithm, you can use it to play against humans or against other algorithms.

At the end of this activity, you'll have the chance to play against the master Tic-Tac-Toe algorithm that has never lost a game!

## *Materials:*

- Paper, pencils (To play tic-tac-toe)
- Set of random move cards (Page 6)
- Instruction cards for Player 1 and Player 2 (Pages 7-8)
- Instruction board for Player 1 and Player 2 (Pages 9-10)



## *Can computers beat humans at Tic-Tac-Toe?*

Many chess players used to scoff at the idea that a computer could one day beat the humans at chess. They thought that computers could never have the same intelligence as the best chess players. But, as technology advanced, computers became better and better at playing chess. In 1997, a computer named Deep Blue beat the world chess champion Garry Kasparov. Was Deep Blue more intelligent than humans? This depends on how you define "intelligence." Deep Blue was a computer specially build to play chess. It was just following instructions written by computer programmers. How did it win by just following instructions? In this activity, we will explore how computers can win at games by creating a computer algorithm for Tic-Tac-Toe.

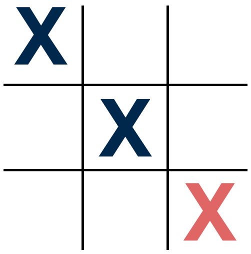
In this activity, you'll create two algorithms: one for Player 1 (the person who goes first), and one for Player 2 (the person who goes second). You'll create your algorithm by assembling instruction cards on your instruction board. When you're ready to test your computer algorithm, play a game of Tic-Tac-Toe against a friend. You are playing as the algorithm, so make sure you follow the instruction cards *exactly*, because this is what a computer does! If you think the computer is making a bad move, rearrange your algorithm! Computer programmers go through *many* rounds of editing when they develop computer algorithms.

Try to create an algorithm that *never loses* a game of Tic-Tac-Toe. Tying a game is OK! Then, see how your algorithm stacks up against the master algorithm.

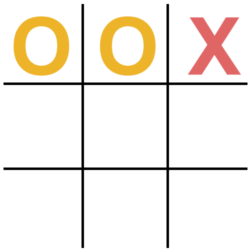
# *Computer Algorithms: Tic-Tac-Toe*

## *Instructions:*

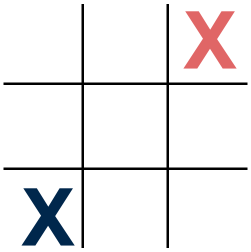
**Step 1.** Choose whether you want to be Player 1 or Player 2. Player 1 always makes the first move during the game. Cut out the instruction cards (Page 7-8) for the Player of your choice. The moves that can be made are restricted to those on the provided instruction cards. The instruction cards include the following moves:



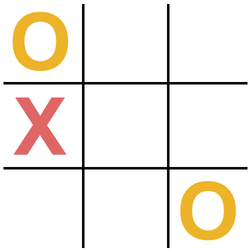
**Go for 3 in a line—** If you have two spaces in a row and the third space is free, go in the third space



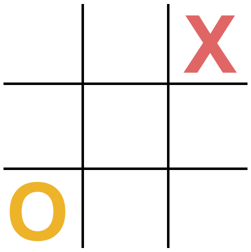
**Block 3 in a line—** If your opponent (O's) has two spaces in a row, go in the third space.



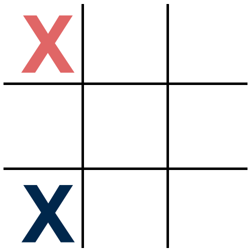
**Go in the opposite corner to my last corner move—** If you went in the corner on the previous turn, go in the opposite corner



**If the other player holds opposite corners, then go in a free edge—** If the other player has two opposite corners, draw a random edge



**Go in the opposite corner to the other player's last corner move—** If your opponent (O's) went in the corner on the previous turn, go in the opposite corner



**Go in the corner on the same side to my last corner move—** If you went in the corner on the previous turn, go in another corner that is *not* the opposite corner

**Go in a *random* free space—** Draw a random number. If that space is free, go in that space. Otherwise, draw again.

**Go in a *random* free corner—** Draw a random number. If that space is free and is a corner (1, 3, 7, or 9), go in that space. Otherwise, draw again.

**Go in a *random* free edge—** Draw a random number. If that space is free and is an edge (2, 4, 6, 8), go in that space. Otherwise, draw again.

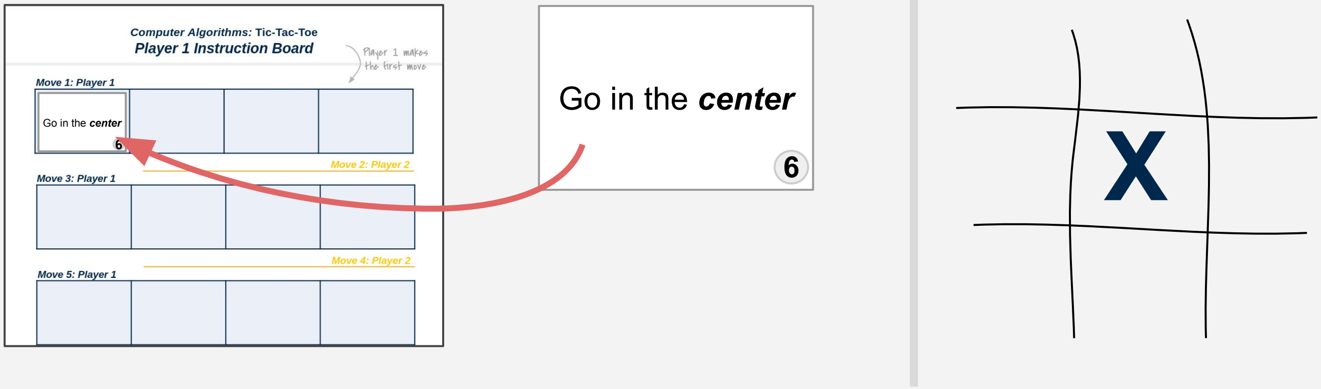
**Go in the center—** Go in the center space if it is free

# Computer Algorithms: Tic-Tac-Toe

**Step 2.** Create your Tic-Tac-Toe algorithm by placing the instruction squares on the Instruction Board for your player. Use at least one instruction square per move. Make sure to include *backup instructions* just in case the first instruction is not valid.

**Example Board.** If Player 1 wants to go in the center of the board, they should place the "Go in the center" instruction square under Move 1. This is the first move of the game.

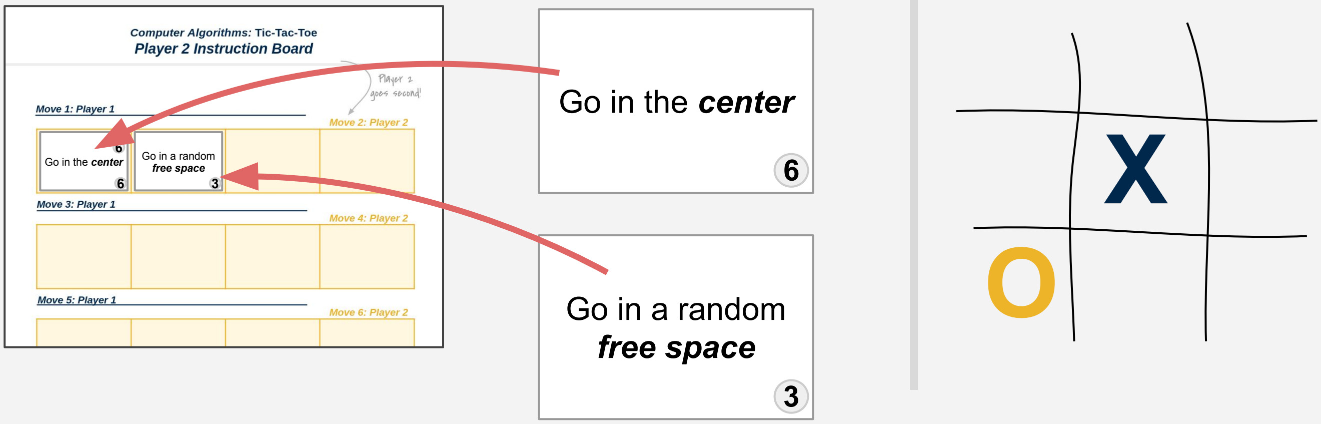
**Instruction Board:** **Instruction Squares:** **Tic-Tac-Toe Board:**



If Player 2 also wants to go in the center of the board, they can place the "Go in the center" instruction square under Move 2.

Since this is the *second move* in the game, the center square may be taken! Player 2 should choose a **backup instruction** just in case their first instruction does not work. Here, Player 2 chooses "Go in a random free space" and places it next to their first instruction. Go to Page 4 for more information on random moves.

**Instruction Board:** **Instruction Squares:** **Tic-Tac-Toe Board:**



**Step 3.** Once there is at least one Instruction Square for each move on your Instruction Board, test out your algorithm by playing Tic-Tac-Toe!

For each move of the algorithm, take the first card for that move. If that move is possible, make it. This ends your turn. Otherwise, try the next instruction card.

**Choosing Random Moves.** When playing your algorithm, you may come across an instruction that asks you to choose a **random** space:

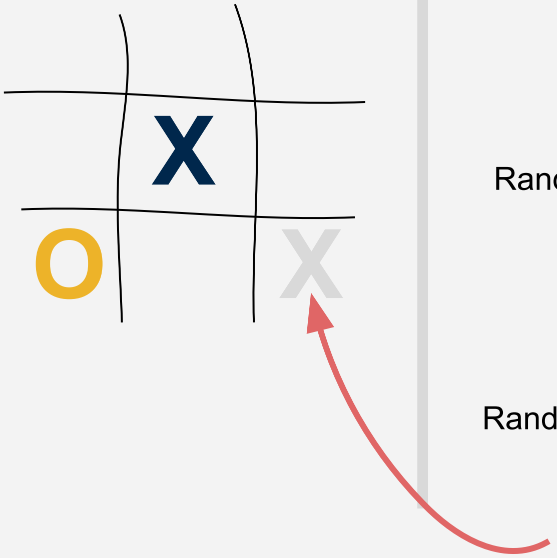
- Go in a random ***free space***
- Go in a random ***free corner***
- Go in a random ***free edge***
- If the other player holds opposite corners, then go in a ***free edge***

How does a computer decide which space to choose?

It uses a ***random number generator***. This is similar to rolling a dice or picking a number out of the hat. For this activity, cut out the numbers on Page 6 and pick a number out of a hat or bowl.

Then, the computer uses the random number to decide which space to go in. Page 6 shows the board position you should go to after drawing a random number. If the space is taken, pick a new random number and try again!

**Tic-Tac-Toe Board:**



**Next move: "Go in a random free corner"**

Random Number Generator: 2

Space 2 is not a corner! Try again.

Random Number Generator: 7

Space 7 is a taken corner! Try again.

Random Number Generator: 9

Space 9 is a free corner! Go there!

**Step 4.** Think about how your algorithm played the game. Did it win? Did it do something you didn't expect? Did it make one move when you thought it should do something different?

Take this chance to change your algorithm as you see fit! Real computer programmers make lots of changes before their algorithm is ready.

**Step 5.** Try playing against the *Master Algorithm* (Page 11-12). How did you do? Remember, your goal is to win or to tie!

Compare your algorithm to the Master Algorithm. How are they similar? How are they different? There are a lot of different ways to solve the same problem. Even if the two algorithms are different, they may be equally successful!

## Helpful Tips:

- Try to develop a strategy on how to play. What is the best first move to make? How can you stop the other player from winning on the next move?
- The best way to develop a strategy is to think about how *you* play Tic-Tac-Toe! Start by playing Tic-Tac-Toe against another person and think about how your moves match the instruction cards.
- If you cannot complete a move (i.e., there are no more valid instructions for your turn), the player must forfeit! It may be a good idea to include "Go in a free space" as the last instruction of every list. It ensures you can always make a move.
- The instructions "Go for 3 in a line" and "Block 3 in a line" refer to the situations where there is only one move before a possible win. These cards are used when either you or your opponent has two spaces in a row and the third space is free. Of course, these cards may not help if your opponent manages to guarantee a win first!

## Wrap-Up:

You've just created an algorithm (list of instructions) for playing Tic-Tac-Toe! A human or computer can use these instructions to play Tic-Tac-Toe and, if the instructions are good, they can play well! In fact, this is all a computer does: follows the instructions written by programmers. To solve problems or create something new, *you* have to come up with the instructions to program the computer. The programmers of Deep Blue did exactly this in order to create the best chess player on the planet.

## *Computer Algorithms:* Tic-Tac-Toe
# *Random Moves*

## *Board positions for randomly chosen numbers:*

After drawing a random number, use this board to determine what space you should go in (e.g., if you choose a 5 you must go in the center space). If the space you choose is taken or does not meet the conditions of the move, just draw another number!

| | | |
|---|---|---|
| **1** | **2** | **3** |
| **4** | **5** | **6** |
| **7** | **8** | **9** |

***Free Space:***
**1-9**

***Free Corner:***
**1, 3, 7, or 9**

***Free Edge:***
**2, 4, 6, or 8**

## *Number cards for random moves:*

Cut out and place in a bowl or hat to use as your random number generator. Ensure the number is not visible through the paper (or close your eyes when drawing from the bowl).

| | | | | |
|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** |
| **6** | **7** | **8** | **9** | |

## *Computer Algorithms:* Tic-Tac-Toe

# *Player 1 Instruction Set*

| | | | |
|---|---|---|---|
| Go for ***3 in a line***<br>1 | Go for ***3 in a line***<br>1 | Go for ***3 in a line***<br>1 | Go for ***3 in a line***<br>1 |
| Block ***3 in a line***<br>2 | Block ***3 in a line***<br>2 | Block ***3 in a line***<br>2 | Block ***3 in a line***<br>2 |
| Go in a random ***free space***<br>3 | Go in a random ***free space***<br>3 | Go in a random ***free space***<br>3 | Go in a random ***free space***<br>3 |
| Go in a random ***free corner***<br>4 | Go in a random ***free corner***<br>4 | Go in a random ***free corner***<br>4 | Go in a random ***free corner***<br>4 |
| Go in a random ***free edge***<br>5 | Go in a random ***free edge***<br>5 | Go in a random ***free edge***<br>5 | Go in a random ***free edge***<br>5 |
| Go in the ***center***<br>6 | Go in the ***center***<br>6 | Go in the ***center***<br>6 | Go in the ***center***<br>6 |
| Go in the ***opposite corner*** to my last corner move<br>7 | Go in the ***opposite corner*** to the other player's last corner move<br>8 | Go in the ***corner on the same side*** to my last corner move<br>9 | If the other player holds opposite corners, then go in a ***free edge***<br>10 |

# *Computer Algorithms:* Tic-Tac-Toe
## ***Player 2 Instruction Set***

| | | | |
|---|---|---|---|
| Go for ***3 in a line***<br>1 | Go for ***3 in a line***<br>1 | Go for ***3 in a line***<br>1 | Go for ***3 in a line***<br>1 |
| Block ***3 in a line***<br>2 | Block ***3 in a line***<br>2 | Block ***3 in a line***<br>2 | Block ***3 in a line***<br>2 |
| Go in a random ***free space***<br>3 | Go in a random ***free space***<br>3 | Go in a random ***free space***<br>3 | Go in a random ***free space***<br>3 |
| Go in a random ***free corner***<br>4 | Go in a random ***free corner***<br>4 | Go in a random ***free corner***<br>4 | Go in a random ***free corner***<br>4 |
| Go in a random ***free edge***<br>5 | Go in a random ***free edge***<br>5 | Go in a random ***free edge***<br>5 | Go in a random ***free edge***<br>5 |
| Go in the ***center***<br>6 | Go in the ***center***<br>6 | Go in the ***center***<br>6 | Go in the ***center***<br>6 |
| Go in the ***opposite corner*** to my last corner move<br>7 | Go in the ***opposite corner*** to the other player's last corner move<br>8 | Go in the ***corner on the same side*** to my last corner move<br>9 | If the other player holds opposite corners, then go in a ***free edge***<br>10 |

## *Computer Algorithms:* Tic-Tac-Toe

# *Player 1 Instruction Board*

Player 1 makes the first move

### *Move 1: Player 1*

Placeholder for Player 2

*Move 2: Player 2*

### *Move 3: Player 1*

*Move 4: Player 2*

### *Move 5: Player 1*

*Move 6: Player 2*

### *Move 7: Player 1*

*Move 8: Player 2*

### *Move 9: Player 1*

Game over! Who won?

## *Computer Algorithms:* Tic-Tac-Toe

# *Player 2 Instruction Board*

Player 2 goes second!

***Move 1: Player 1***

***Move 2: Player 2***

***Move 3: Player 1***

***Move 4: Player 2***

***Move 5: Player 1***

***Move 6: Player 2***

***Move 7: Player 1***

***Move 8: Player 2***

***Move 9: Player 1***

## ***Computer Algorithms:*** **Tic-Tac-Toe**

# ***Player 1 Master Algorithm***

### ***Move 1: Player 1***

| | | | |
|---|---|---|---|
| Go in a random ***free corner***<br>4 | | | |

### ***Move 2: Player 2***

### ***Move 3: Player 1***

| | | | |
|---|---|---|---|
| Go in the ***opposite corner*** to my last corner move<br>7 | Go in a random ***free corner***<br>4 | | |

### ***Move 4: Player 2***

### ***Move 5: Player 1***

| | | | |
|---|---|---|---|
| Go for ***3 in a line***<br>1 | Block ***3 in a line***<br>2 | Go in a random ***free corner***<br>4 | Go in a random ***free space***<br>3 |

### ***Move 6: Player 2***

### ***Move 7: Player 1***

| | | | |
|---|---|---|---|
| Go for ***3 in a line***<br>1 | Block ***3 in a line***<br>2 | Go in a random ***free corner***<br>4 | Go in a random ***free space***<br>3 |

### ***Move 8: Player 2***

### ***Move 9: Player 1***

| | | | |
|---|---|---|---|
| Go in a random ***free space***<br>3 | | | |

# *Computer Algorithms:* Tic-Tac-Toe

# *Player 2 Master Algorithm*

### *Move 1: Player 1*

### *Move 2: Player 2*

| | | | |
|---|---|---|---|
| Go in the ***center***<br>6 | Go in a random ***free corner***<br>4 | | |

### *Move 3: Player 1*

### *Move 4: Player 2*

| | | | |
|---|---|---|---|
| Block ***3 in a line***<br>2 | If the other player holds opposite corners, then go in a ***free edge***<br>10 | Go in the ***opposite corner*** to the other player's last corner move<br>8 | Go in a random ***free corner***<br>4 |

### *Move 5: Player 1*

### *Move 6: Player 2*

| | | | |
|---|---|---|---|
| Go for ***3 in a line***<br>1 | Block ***3 in a line***<br>2 | Go in a random ***free corner***<br>4 | Go in a random ***free space***<br>3 |

### *Move 7: Player 1*

### *Move 8: Player 2*

| | | | |
|---|---|---|---|
| Go for ***3 in a line***<br>1 | Block ***3 in a line***<br>2 | Go in a random ***free corner***<br>4 | Go in a random ***free space***<br>3 |

### *Move 9: Player 1*