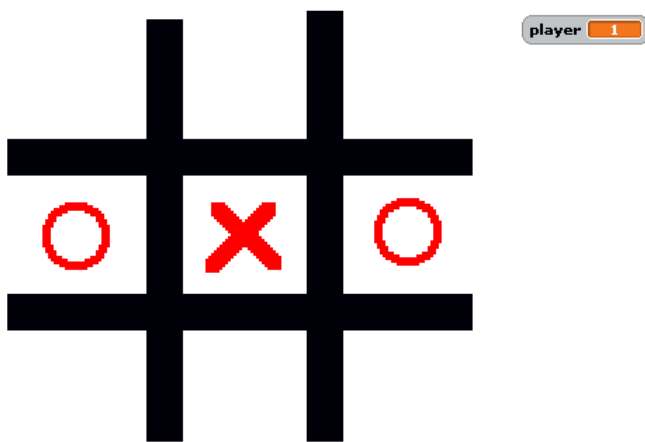


This project explores how to create the popular game of Tic Tac Toe otherwise known as Noughts and Crosses. The aim of this Sushi card is to help you understand how to construct the game and learn some new coding skills along the way! Ready for a challenge?

Lets start with a two player game, where each player takes a turn to put an X or a O into a box and if you get a line of X's or O's you win!



Some key things that need to be worked out:

- Draw The Board (it could be the background or it could be a sprite)
- How about some Sprites for the O and X to appear
- Resetting for the next game
- How do you know whose go it is
- How do you know if a player has a winning line

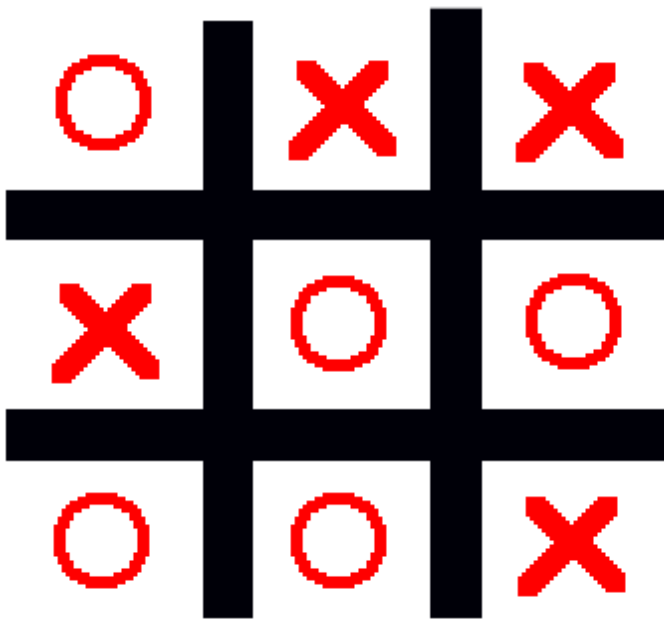
Some hints:

- Visually you can see what is happening but how can your program test if you have filled in a line – think of the squares as a grid of 1-9 squares – how would you represent that in Scratch?
- Test your program with one line
- Then build it out for three lines

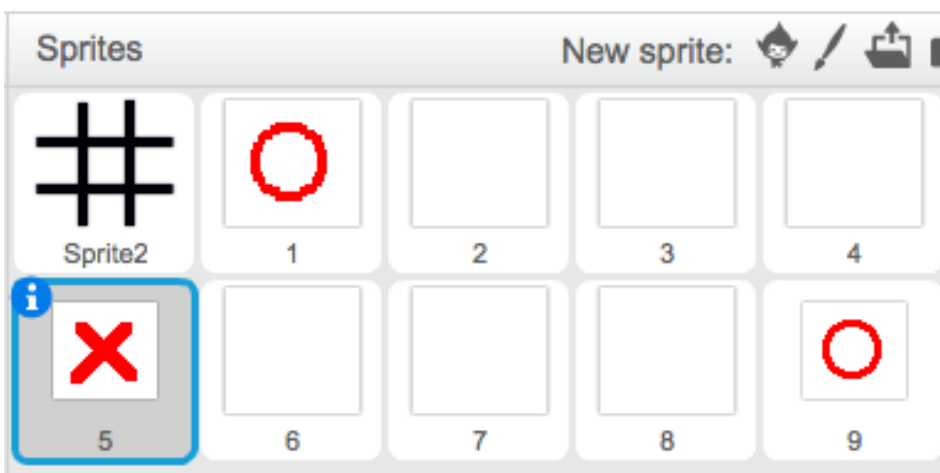
Don't forget to load the project back to the Scratch site and show the class!

### The Basics

**1** The Board: You can draw this any way you want, but the key here is to have a space where a player can make their move. When the player makes



**2** Making choices should be easy enough to figure out. A player clicks on a space. How do we know the player clicked a space? One way is to use sprites....



# Scratch

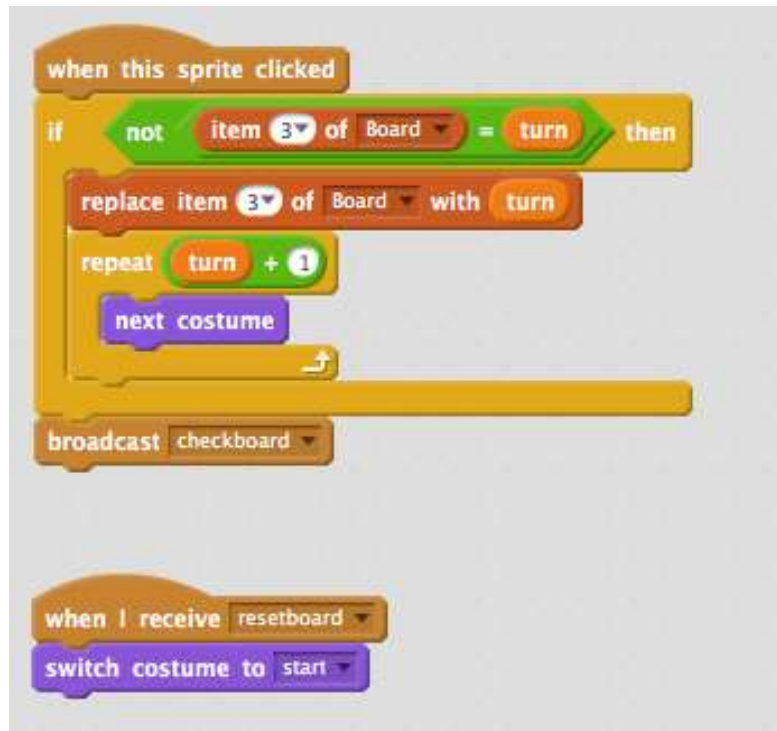
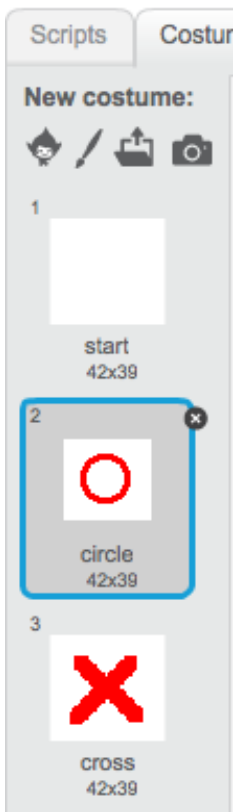


I'm Learning about

Tic Tac Toe Game

Card 3 of 7

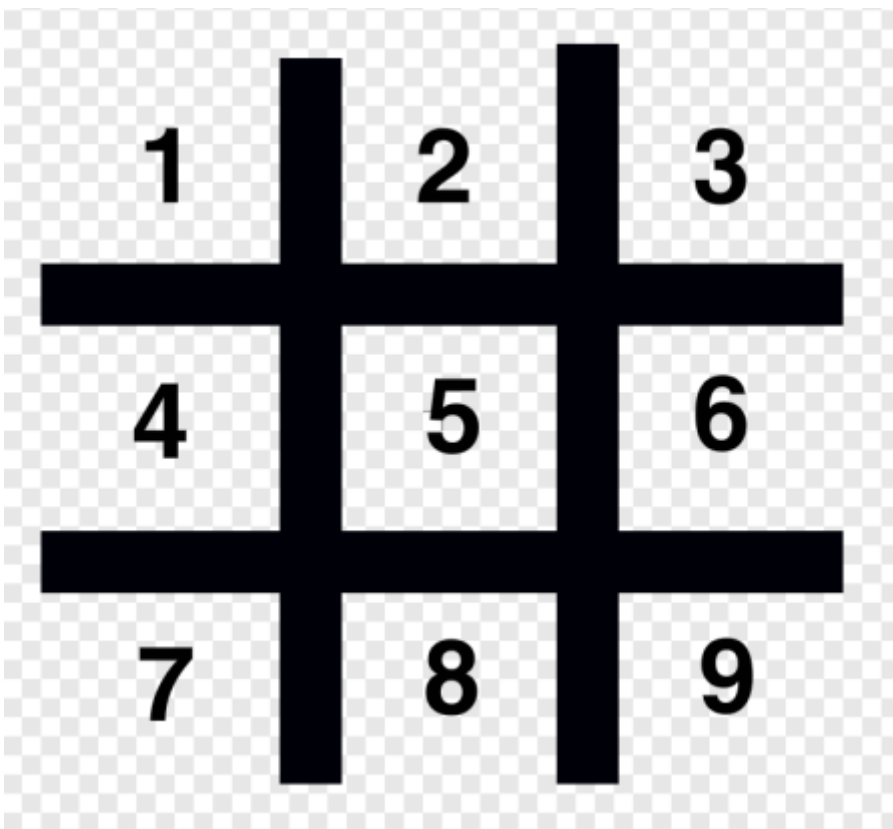
And using costumes and with some code to check if the sprite has been clicked we can select the right costume.



## Playing History

**4** As each play is made it will be necessary for you to track what squares have been played, not only to ensure a player doesn't play that square again, but also because you need to know if there is a winning line.

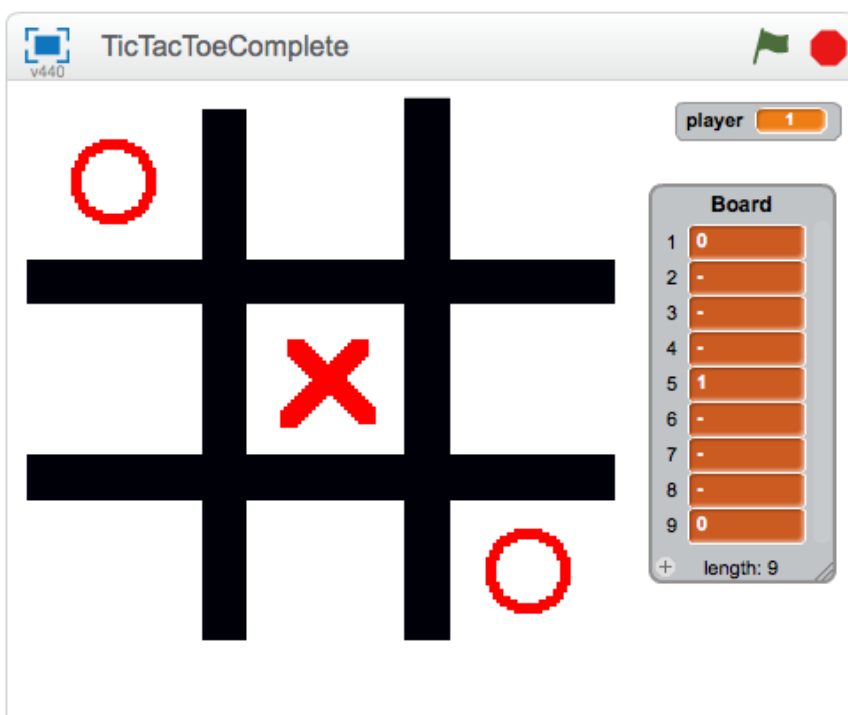
One approach for this is to store what squares have been played. The way to imagine this is to number each square to form a grid.



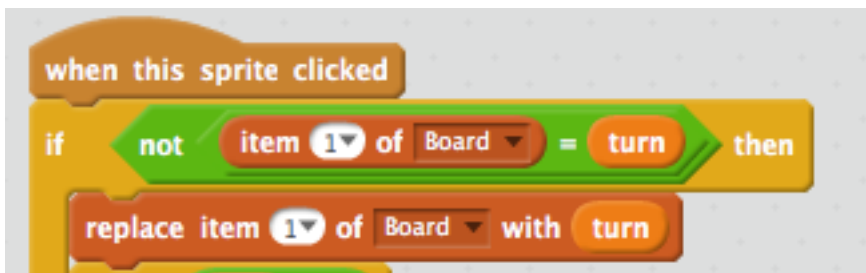
Now you can have a grid, you can see there are 9 possible entries. This grid can also be thought of a list.

Knowing what has been played

5 As each play is made it you can store what has been played in a list. Create a list to represent a board using the Data Palatte



5 As each play is made record which player used which square.



It would be good to check if the square has been played!

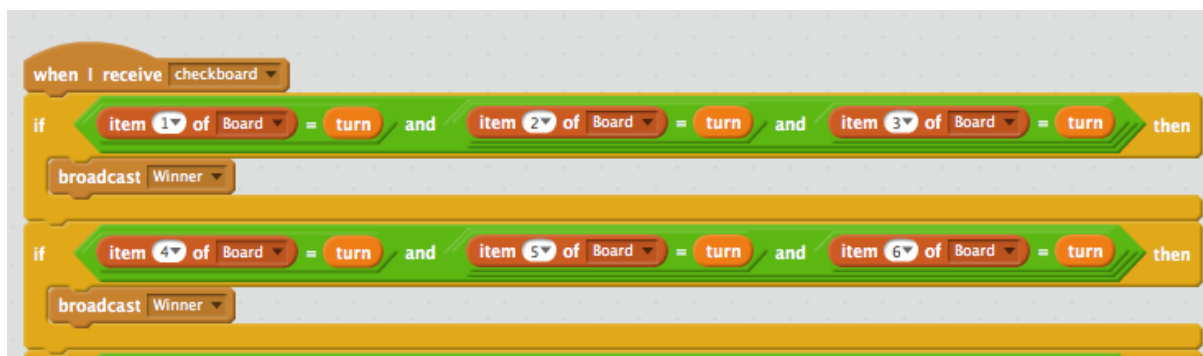
### Checking for a winning line

**6** Now we have to check if we have a Winner. The code here is simpler than you think. We can directly code for which combinations are valid, for example:

- 1,2,3 - winner
- 4,5,6 - winner
- 7,8,9 - winner
- 1,4,7 - winner
- 2,5,8 - winner
- 3,6,9 - winner

*Are all winning combinations covered here?*

Lets see how that might look....



```
when I receive checkboard
  if (item 1 of Board = turn and item 2 of Board = turn and item 3 of Board = turn) then
    broadcast Winner
  if (item 4 of Board = turn and item 5 of Board = turn and item 6 of Board = turn) then
    broadcast Winner
```

Remember to check on each turn if you have a winner.

*You will also need to make sure there's not a draw condition which will also end the game!*

## Improving the Game

**7** How do you think we can improve the game? Here are some suggestions:

- Draw a line over the winning squares!
- Scoreboard – keep tabs on who is winning
- Make a sound if the wrong square is selected
- Can you reduce the number of sprites used?
- And if that's all too easy – Make the second player 2 the computer!