# CSC 108
# Spring 2010

Pre-lab06 Tic-Tac-Toe

Dr. Yue-Ling Wong

# Pre-lab 06 Tic-Tac-Toe

- Due: 2:00pm Monday
- Upload to Blackboard
- No late submission for any pre-lab

- Estimated time to complete the pre-lab: no more than an hour

# Step 1.

Before Coding:

Visual Content Preparation

# Visual Content Preparation

- Create one movieclip. (See the sample .fla file posted on the course Web site.)
  In this movieclip, create 3 keyframes:
  - frame 1: how you want each cell in your game board to look by default (without marking)
  - frame 2: player 1's token, for example, an "X"
  - frame 3: player 2's token, for example, an "O"

  Give frame 2 a frame label "player1" and frame 3 a frame label "player2"

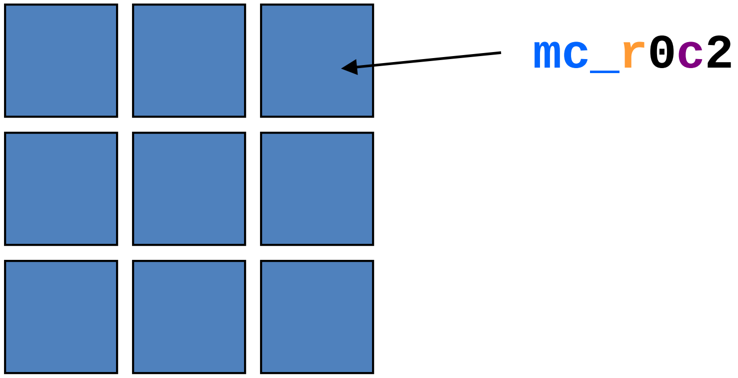
- Don't forget to add code to stop this movieclip from looping.

- Put 9 copies of this movieclip on stage.  Arrange them like the tic-tac-toe game board.

- Add a dynamic text on stage, and name it, say, `txt_feedback`.

# MovieClip Instance Naming

- Give each of the 9 copies of the movieclip an instance name using the following rules:
  - start each name with **mc_** or end with **_mc**
  - the rest of the name starts with "r", then the row number, then "c", and then the column number, like this:



# MovieClip Instance Naming

- Be sure to use the correct row and column numbers.

- The numbering follows how an array is indexed, i.e. starting with 0, not 1.

- The purpose of this part is to make you practice thinking in the way of how an array is indexed.

- Points will be taken off if the naming does not correctly correspond to the row and column numbers of the block, according to how an array is indexed!!

# Step 2.

Coding:

Create a variable to keep track of which player's turn

# Use a Variable to Keep Track of Which Player's Turn

Create a variable, `currPlayer`, to keep track of which player's turn.

- Because there will be two players in the game, we will use the value of 1 and 2 to track the turn.
- What should the data type be?
- What should `currPlayer` be initialized to?

# Step 3.

Coding:

Create two 2D arrays to model the tic-tac-toe game board and keep track of the markings on the board

# Modeling the 2d Game Board

- Create a 2-dimensional array, called `board`, to store the movieclip names in the array.

- It starts like this:

```
var board:Array =
```

- Review the 2-dimensional arrays lecture if you don't know how to create a 2-d array.

| | | |
|---|---|---|
| mc_r0c0 | mc_r0c1 | mc_r0c2 |
| mc_r1c0 | mc_r1c1 | mc_r1c2 |
| mc_r2c0 | mc_r2c1 | mc_r2c2 |

# A Second 2-d Array

Create a second 2-dimensional array, called `markings`, with a 3x3 of zeros.

```
var markings:Array =
```

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |

# Step 4.

Coding:

Write event listener code to allow the player to mark a cell on the game board by clicking on it

# Mouse Event Handling

Add code (5 statements) in the mouse up event handler function for the cell:

1. assign the value of `currPlayer` to the corresponding element in the `markings` array, for example, if you click on `mc_r0c0`, then

   ```
   markings[0][0] = currPlayer;
   ```

2. a `trace` statement to output the array `markings`, so you can see the values of the array

3. make the playhead of this block to go to and stop at the frame that represents the current player

   ```
   board[0][0].gotoAndStop("player" + currPlayer);
   ```

4. switch player by switching the value of `currPlayer`

   Pseudocode:

   ```
   if currPlayer is 1 then
       set currPlayer to 2
   else
       set currPlayer to 1
   ```

5. display the message saying whose turn in the dynamic text box
   (Review how you display the score in the Pong game if you don't remember how to display text in a textbox on stage.)

# Total 9 Sets of Mouse Event Listener Code

- Once your mouse event listener code is working for a cell, create 8 more sets of the mouse event listener code, so that you have total 9 sets of mouse event listener code, each for a cell.