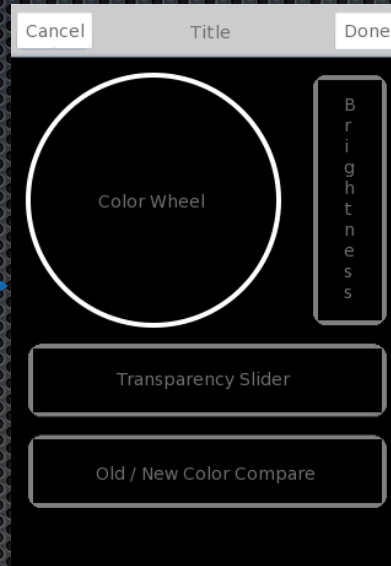


# Mobile Application Programming

## TicTacToe



# The Iterative Design Process



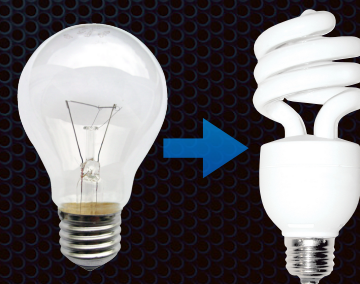
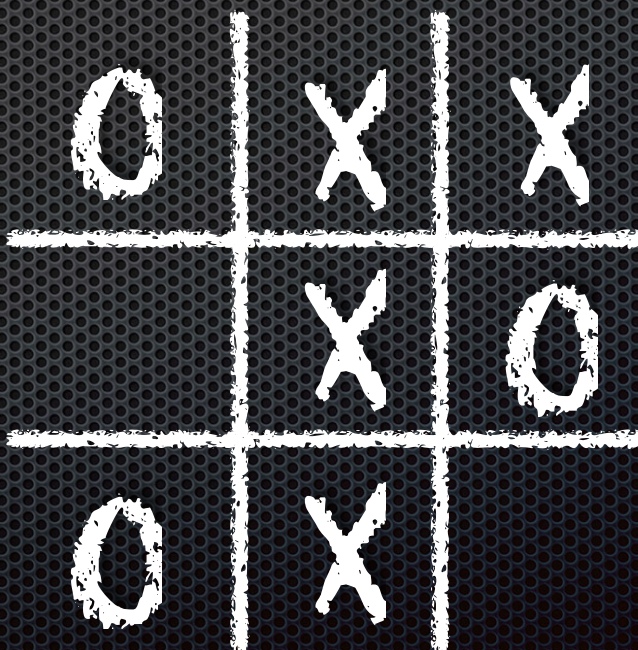
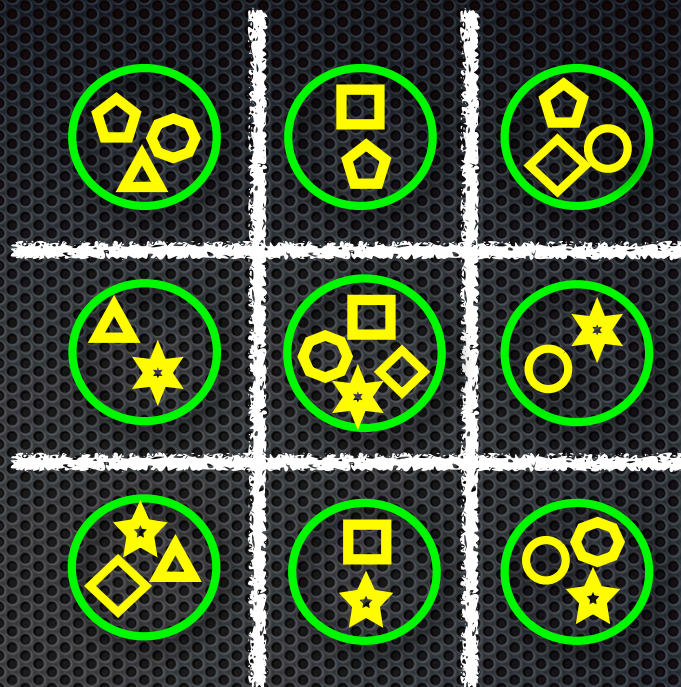
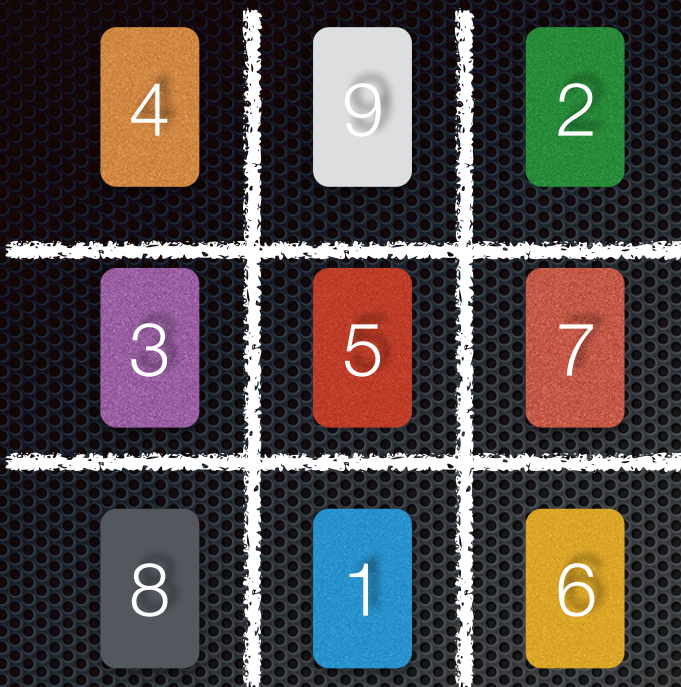
```
let quoteListJson: NSData? = NSData(c
if (quoteListJson == nil)
{
    return
}

let quoteListJsonString: NSString = N
println(quoteListJsonString)

let quoteAttributions: NSArray? = NS
allZeros, error: nil) as NSArray?
if (quoteAttributions == nil)
{
    return
}
}
for quoteDictionary in quoteAttribut
{
    let quoteID: String? = quoteDicti
    let quoteAttribution: String? = c
    if (quoteID != nil && quoteAttrib
    {
        _quoteIDs.append(quoteID!)
    }
}
```









Matt Stokar

Mar/6/2017

Tic-Tac-Toe

### Abstract

An implementation of the paper & pencil game Tic-Tac-Toe with support for in-progress and completed games.

### Game List Screen

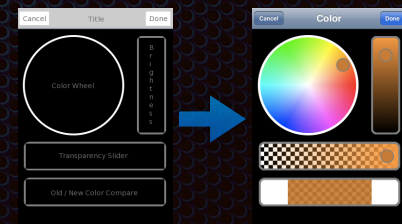
Tic-Tac-Toe	+
X's Turn	2 moves
O's Turn	3 moves
O Won	6 moves
Tie	9 moves

- Tapping '+' opens a new game
- Tapping a row opens that game
- Games can be deleted by right-swipe

### Game Screen

Back	O's Turn	Delete
		X
	X	O

- Tapping a cell selects that location for a move
- The title updates to reflect who's turn it is
- Tapping 'Back' returns to the game List Screen
- Tapping 'Delete' deletes this game
- When a player wins, tapping a cell informs the user that the game has ended







```
let quoteListJson: NSData? = NSData(contentsOfFile: quoteListJsonPath)
if (quoteListJson == nil)
{
    return
}

let quoteListJsonString: NSString = NSString(data: quoteListJson!, encoding: UTF8StringEncoding)
println(quoteListJsonString)

let quoteAttributions: NSArray? = NSData(contentsOfFile: quoteAttributionsPath)
if (quoteAttributions == nil)
{
    return
}

for quoteDictionary in quoteAttributions
{
    let quoteID: String? = quoteDictionary["quoteID"]
    let quoteAttribution: String? = quoteDictionary["quoteAttribution"]
    if (quoteID != nil && quoteAttribution != nil)
    {
        _quoteIDs.append(quoteID!)
    }
}
```

