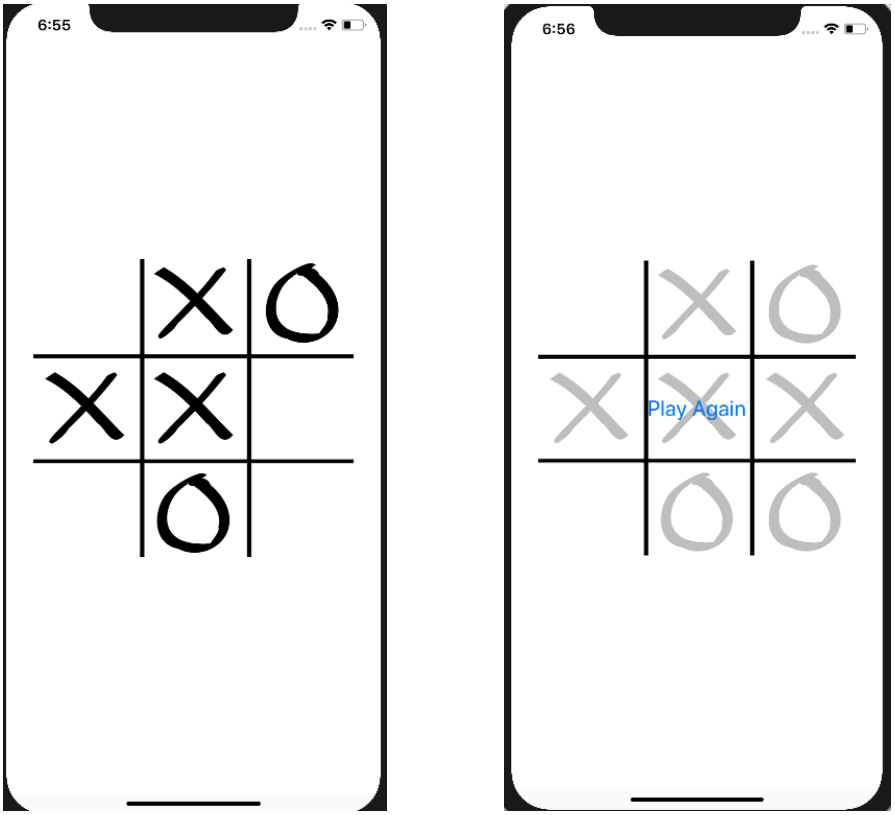
Innotech
Mohamm
April 8, 2018

Swift Document

Udemy iOS 10(TicTacToe App)

We all know TicTacToe and in this document we gonna make this game together and learn to make little animation moves.



if you have finished your graphical view like the left picture above, add the Play Again button to pass them to ViewController.(Empty their pictures)

* note that in this game all of the X and Os I chose, are UIButton type.

```
@IBOutlet weak var playAgainButton: UIButton!
@IBAction func buttonPressed(_ sender: Any) {
```

for buttonPressed, at first drag one of the buttons and after making the outlet, drag all of the others.(except Play Again)

```
@IBAction func toPlayAgain(_ sender: Any) {
```

My algorithm for this game is that I have the winStates and an array which is full of 9 zeros that changes when each player touches buttons.

```
var player = 1  // 1 is nought and 2 is cross
var gameState = [0,0,0,0,0,0,0,0,0]
let winState = [[0,1,2],[3,4,5],[6,7,8],[0,3,6],[1,4,7],[2,5,8],[0,4,8],[2,4,6]]
```

I want to make an animation which when our game finished Play Again button comes in, so at first it must be at the left of the screen and hidden.

```
override func viewDidLoad() {
  super.viewDidLoad()
  playAgainButton.center = CGPoint(x: playAgainButton.center.x - 500, y:
      playAgainButton.center.y)
  playAgainButton.isHidden = true
}
```

to work with buttons separately in ViewController, usually we assign them tag numbers.

* 0 is for default.

* My 9 buttons numbered from 1 to 9 and Play Again is 10.

Now for example Jack and John are playing and when Jack clicks, its Johns turn. As shown before I change their turns with the help of a globally variable called player.(and also you if they weren't clicked before.)

```
@IBAction func buttonPressed(_ sender: Any) {

    if gameState[(sender as AnyObject).tag - 1] == 0{

        if player == 1 {
            gameState[(sender as AnyObject).tag - 1] = 1
            (sender as AnyObject).setImage(X, for: [])
            player = 2
        }else {
            gameState[(sender as AnyObject).tag - 1] = 2
            (sender as AnyObject).setImage(O, for: [])
            player = 1
        }
    }
```

When John clicks, the state of game will changed and Image of that button has to be changed from empty to nought.

There must be a for loop which always check if anyone win the game:

- first checks that the state shouldn't be zero.(it means that button didn't pressed)
- then like the pattern of winState, states must be same.

```
for winstate in winState{
        if gameState[winstate[0]] != 0
            && gameState[winstate[0]] == gameState[winstate[1]]
            && gameState[winstate[1]] == gameState[winstate[2]]{
```

Above if, means: if we have a winner.

Then all of the buttons must be disabled and Play Again button have to flow in with the help of UIView.animation function.

```
for winstate in winState{
    if gameState[winstate[0]] != 0
        && gameState[winstate[0]] == gameState[winstate[1]]
        && gameState[winstate[1]] == gameState[winstate[2]]{

        var tempButton : UIButton
        for i in 1..<10{
            tempButton = view.viewWithTag(i) as! UIButton
            tempButton.alpha = 0.5
            tempButton.isEnabled = false
        }
        playAgainButton.isHidden = false
        UIView.animate(withDuration: 1, animations: {
            self.playAgainButton.center = CGPoint(x: self.playAgainButton.center.x +
                500, y: self.playAgainButton.center.y+100)
        })
    }
}
```

* reason of using self keyword is because those lines are in closure:)

Now its time for implementing our last function to start over the game by clicking on Play Again button.

Pictures of all of the buttons must be empty and they have to be enabled again, their alpha have to be 1 again , gameStates must be refreshed and Play Again button should be hidden.

```
@IBAction func toPlayAgain(_ sender: Any) {
    var tempButton : UIButton
    for i in 1..<10{
        tempButton = view.viewWithTag(i) as! UIButton
        tempButton.setImage(nil, for: [])
        tempButton.isEnabled = true
        tempButton.alpha = 1
        gameState[i-1] = 0
    }
    playAgainButton.isHidden = true
}
```