**Final Project Report**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\***

TicTacToe Game on iOS

**Student Name: Minh Le**

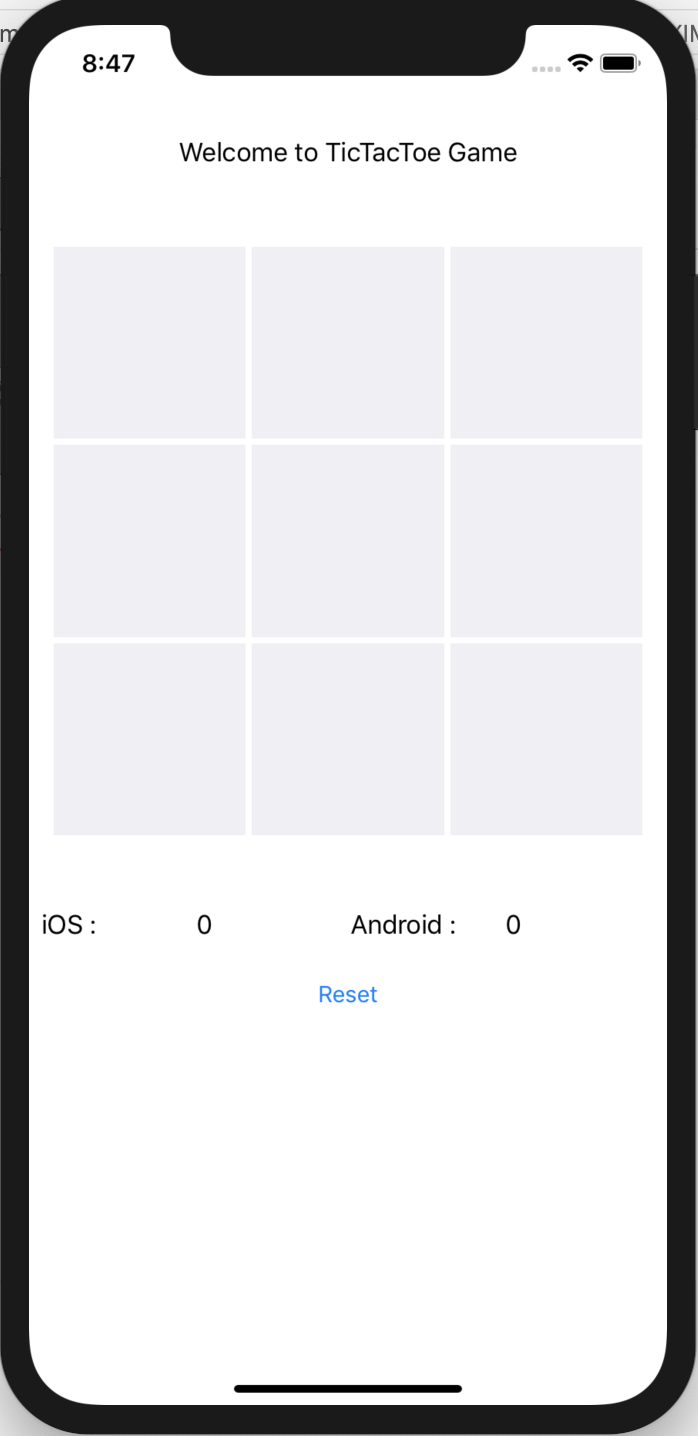
**Student Number: 1834968**

Platform: iOS

Language: Swiff

***I/ User interface:***

The game has only 1 screen:



***II/Select a cell (func “Section”)***

When a cell is selected, we will call function “Section”:

1. To set an corresponding image for the cell

2. Then disable the cell

3. Then change the parameter “player” so the next selection will be his/her opponent’s move.

4. Call the function “Check” to see if this move matches the winning conditions (or draw if this is the last remaining cell)

**@IBAction** **func** Section(**\_** sender: UIButton) {

sender.isEnabled = **false**

**if** (player == 1){

sender.setImage(UIImage(named: "apple.png"), for: UIControl.State.normal)

board[sender.tag] = 1

check(player: player)

player = 2

}

**else**{

sender.setImage(UIImage(named: "android.png"), for: UIControl.State.normal)

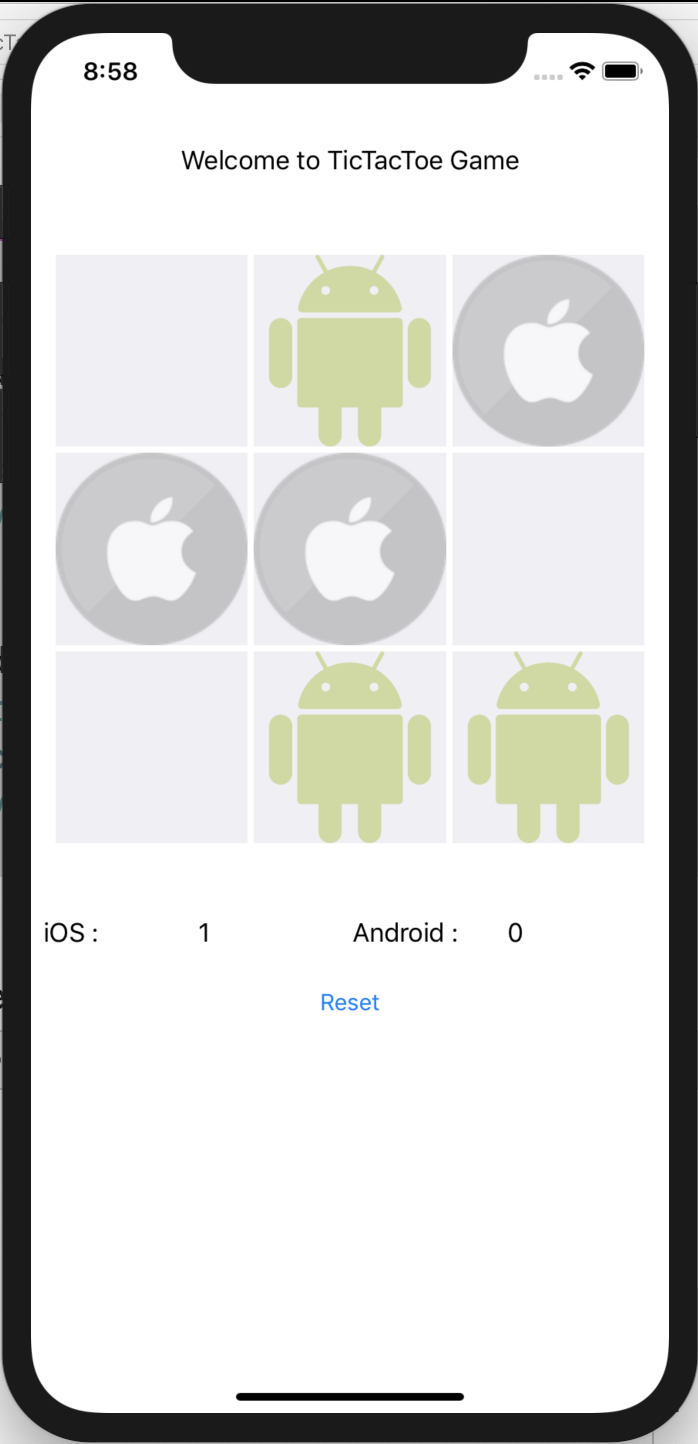
board[sender.tag] = 2

check(player: player)

player = 1

}

}



***III/ Check a move (func check)***

The function will be called when a cell is selected. It will get the an integer parameter “player” to identify the players (iOS is “1” and Android is “2”).

**III.A When the board still have unselected cell or cells.**

This is winning conditions: [[0,1,2],[3,4,5],[6,7,8],[0,3,6],[1,4,7],[2,5,8],

[0,4,8],[2,4,6]] . The numbers are the cell positions (begin with 0)

We will do a loop in the winning conditions Array to check if the move match any winning condition. If yes, display an alert result, increase winning count for the winner by 1 and display it, if no, do nothing.

**func** check(player: Int){

winOrdraw = 2

*//check when board still have moves to play*

**if** (board.contains(0) == **true**){

**if**(player == 1){

**for** winingline **in** winningcondition{

**if** board[winingline[0]] != 0 && board[winingline[0]] == board[winingline[1]] && board[winingline[1]] == board[winingline[2]]{

**self**.createAlert(title: "Result!", message: "iOS player win!")

player1winCount = player1winCount + 1

player1win.text = String(player1winCount)

}

}

}

**else** **if** (player == 2)

{

**for** winingline **in** winningcondition{

**if** board[winingline[0]] != 0 && board[winingline[0]] == board[winingline[1]] && board[winingline[1]] == board[winingline[2]]{

**self**.createAlert(title: "Result!", message: "Android player win!")

player2winCount = player2winCount + 1

player2win.text = String(player2winCount)

}

}

}

}

**III.B When the board has no move cell to select**

Similar to III.A but at this time: If yes, display an alert result, increase winning count for the winner by 1 and display it, if no, display an draw alert.

**else**{

**if**(player == 1){

**for** winingline **in** winningcondition{

print(winingline[0])

myCondition: **if** (board[winingline[0]] != 0 && board[winingline[0]] == board[winingline[1]] && board[winingline[1]] == board[winingline[2]]){

**self**.createAlert(title: "Result!", message: "iOS player win!")

player1winCount = player1winCount + 1

player1win.text = String(player1winCount)

winOrdraw = 1

**break** myCondition

}

}

}

**else** **if** (player == 2)

{

**for** winingline **in** winningcondition{

myCondition: **if** (board[winingline[0]] != 0 && board[winingline[0]] == board[winingline[1]] && board[winingline[1]] == board[winingline[2]]){

**self**.createAlert(title: "Result!", message: "Android player win!")

player2winCount = player2winCount + 1

player2win.text = String(player2winCount)

winOrdraw = 1

**break** myCondition

}

}

}

**if** (winOrdraw == 2)

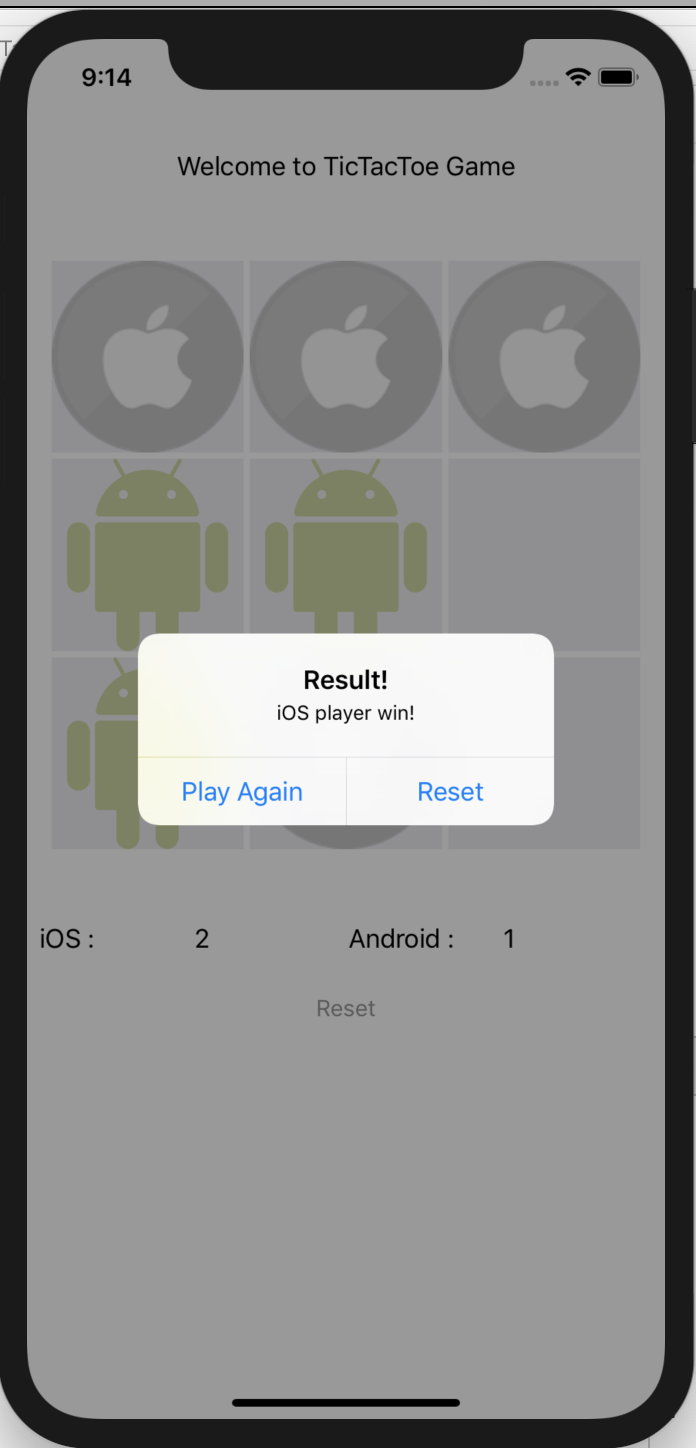
{

**self**.createAlert(title: "Result!", message: "What a draw!")

}

}

}



***IV/Reset (func reset())***

Just reset winning counts to 0, enable disabled cells, clear the board.

**@IBAction** **func** Reset(**\_** sender: UIButton) {

reset()

player1winCount = 0

player2winCount = 0

**self**.player1win.text = "0"

**self**.player2win.text = "0"

}

**func** reset(){

winOrdraw = 1

board = [0,0,0,

0,0,0,

0,0,0]

**for** UIButtion **in** Section{

UIButtion.setImage(UIImage(named: ""), for: UIControl.State.normal)

UIButtion.isEnabled = **true**

}

}

***V/Alert (func createAlert)***

Just the create an alert with 2 buttons: Play again (Play without reset winning counts) and reset (Play again but winning counts are 0 again)

**func** createAlert(title: String, message: String){

**let** alert = UIAlertController(title: title, message: message, preferredStyle: UIAlertController.Style.alert)

alert.addAction(UIAlertAction(title: "Play Again", style: .default, handler: { (action) **in**

alert.dismiss(animated: **true**, completion: **nil**)

**self**.reset()

}))

alert.addAction(UIAlertAction(title: "Reset", style: .default, handler:{ (action) **in**

alert.dismiss(animated: **true**, completion: **nil**)

**self**.reset()

**self**.player1winCount = 0

**self**.player2winCount = 0

**self**.player1win.text = "0"

**self**.player2win.text = "0"

}))

**self**.present(alert, animated: **true**,completion: **nil**)

}