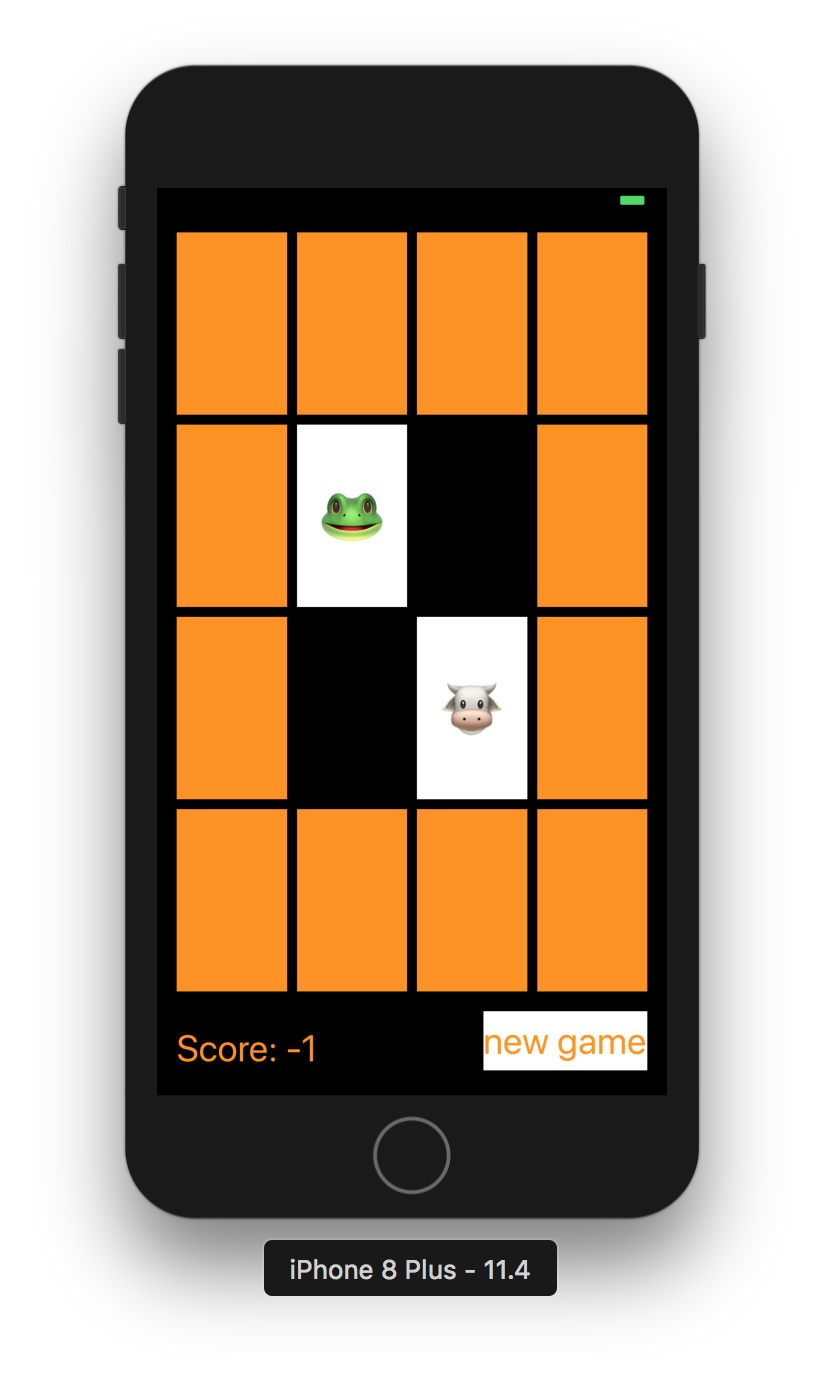
实验编号： 7 **四川师大《IOS》实验报告 2018** 年 **10** 月 **24** 日

### **计算机科学学院** 2016 级 4 班 实验名称： Game单MVC \_

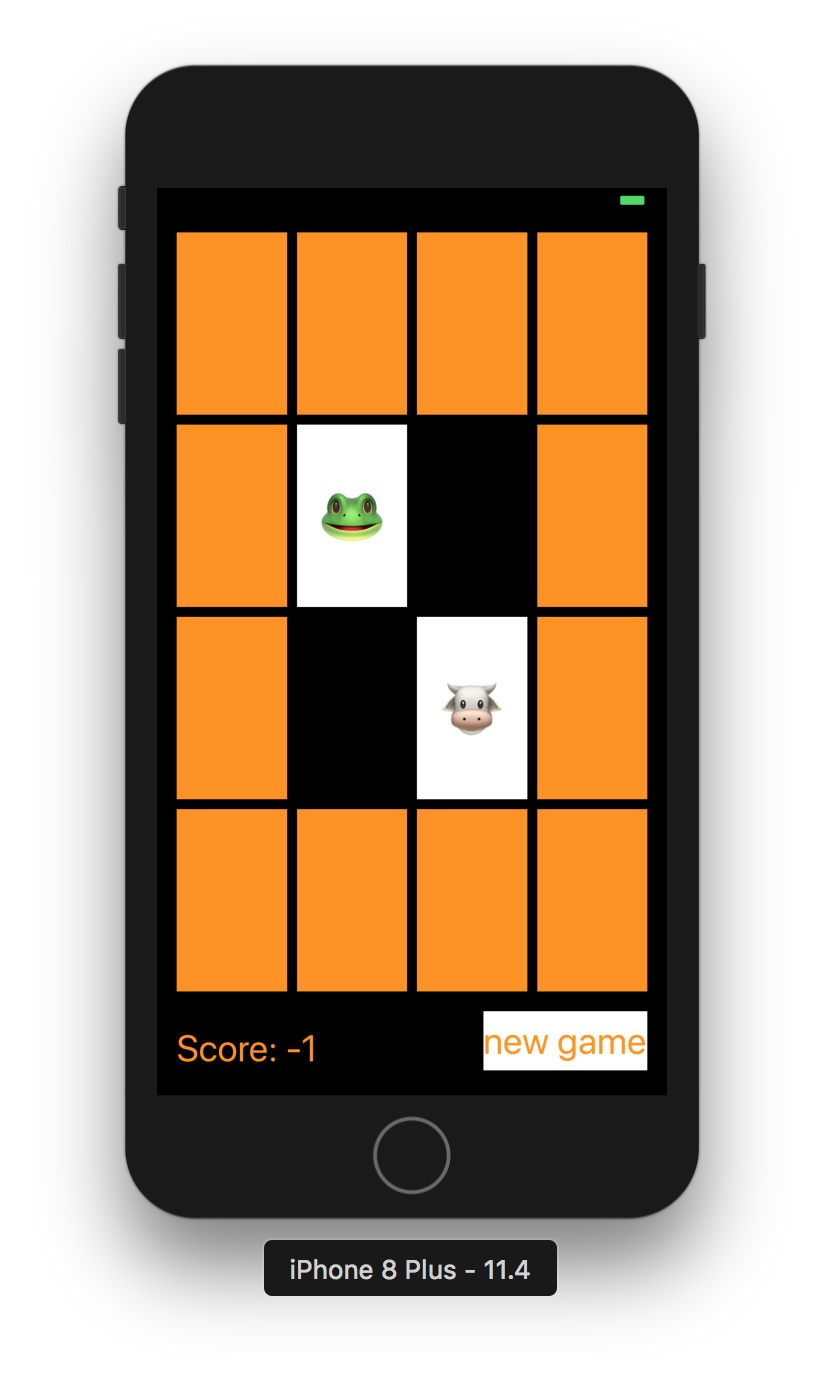
姓名：\_樊琳\_\_ 学号：\_2016110408指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验 七 \_\_\_\_\_\_** Game单MVC **\_\_\_\_\_\_\_\_**

1. 实验目的及要求
2. 实现一款功能完整的game（Concentration）；
3. 掌握单MVC的主要思想；
4. 实验内容
5. 参照Stanford视频1和2完成一个game（Concentration）的制作；
6. 在(1)的基础上进一步完成Stanford Assignment 1的完整要求；
7. 采用autolayout布局解决横竖屏自适应如下所示；



1. 实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）
2. 参照Stanford视频1和2完成一个game（Concentration）的制作；
3. 在(1)的基础上进一步完成Stanford Assignment 1的完整要求；
4. 采用autolayout布局解决横竖屏自适应如下所示；



* 程序代码：

//

// ViewController.swift

// Concentration

//

// Created by student 2018/12/13.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

class ViewController: UIViewController {

lazy var game = Concentration(numberOfPairsOfCards: (cardButtons.count + 1) / 2)

@IBOutlet weak var scoreLabel: UILabel!

@IBOutlet var cardButtons: [UIButton]!

@IBAction func touchCard(\_ sender: UIButton) {

if let cardNumber = cardButtons.index(of: sender) {

game.chooseCard(at: cardNumber)

updateViewFromModel()

} else {

print("choose card was not in cardButtons")

}

}

func updateViewFromModel() {

for index in cardButtons.indices {

let button = cardButtons[index]

let card = game.cards[index]

if card.isFaceUp {

button.setTitle(emoji(for: card), for: UIControlState.normal)

button.backgroundColor = colorLiteral(red: 0.9999960065, green: 1, blue: 1, alpha: 1)

} else {

button.setTitle("", for: UIControlState.normal)

button.backgroundColor = card.isMatched ? colorLiteral(red: 1, green: 1, blue: 1, alpha: 0) : colorLiteral(red: 1, green: 0.5763723254, blue: 0, alpha: 1)

}

}

scoreLabel.text = "Score: \(game.score)"

}

var themes = [0:["🎃","👻","🐶","🐱","🐭","🐹","🦊","🐼","🐵","🐮","🐸"],

1:["🍏","🍎","🍐","🍊","🍋","🍌","🍉","🍇","🍑","🍒","🍓"],

2:["⚽️","🏀","🏈","⚾️","🎱","🏉","🏐","🎾","🏓","🏸","🏒"],

3:["🚗","🚕","🚙","🚌","🚑","🚓","🏎","🚎","🚒","🚚","🛵"],

4:["⌚️","📱","💻","🖨","🖥","⌨️","💽","🗜","🕹","💾","☎️"],

5:["🇦🇱","🇩🇿","🇦🇫","🏳️‍🌈","🇦🇷","🇦🇪","🇦🇼","🇴🇲","🇮🇪","🇪🇹","🇪🇬"]]

lazy var emojiChoices = themes[0]!

var emoji = [Int: String]()

func emoji(for card: Card) -> String {

if emoji[card.identifier] == nil, emojiChoices.count > 0 {

let randomIndex = Int(arc4random\_uniform(UInt32(emojiChoices.count)))

emoji[card.identifier] = emojiChoices.remove(at: randomIndex)

}

return emoji[card.identifier] ?? "?"

}

@IBAction func newGame(\_ sender: UIButton) {

game = Concentration(numberOfPairsOfCards: (cardButtons.count + 1) / 2)

let them = Int(arc4random\_uniform(UInt32(themes.keys.count)))

emojiChoices = themes[them]!

updateViewFromModel()

}

}

//

// Concentration.swift

// Concentration //

// Created by student 2018/12/13.

// Copyright © 2018年 fl. All rights reserved.

//

import Foundation

class Concentration

{

var score = 0

var cards = [Card]()

var indexOfOneAndOnlyFaceUpCard: Int?

func chooseCard(at index: Int) {

if !cards[index].isMatched {

if let matchIndex = indexOfOneAndOnlyFaceUpCard, matchIndex != index {

if cards[matchIndex].identifier == cards[index].identifier {

cards[matchIndex].isMatched = true

cards[index].isMatched = true

score += 2

} else {

if cards[matchIndex].isSeen {

score -= 1

}

if cards[index].isSeen {

score -= 1

}

}

cards[index].isFaceUp = true

indexOfOneAndOnlyFaceUpCard = nil

} else {

for flipDownIndex in cards.indices {

cards[flipDownIndex].isFaceUp = false

}

cards[index].isFaceUp = true

indexOfOneAndOnlyFaceUpCard = index

}

}

}

init(numberOfPairsOfCards: Int) {

for \_ in 1...numberOfPairsOfCards {

let card = Card()

cards += [card,card]

}

// TODO: Shuffle the cards

cards.sort { \_,\_ in arc4random\_uniform(2) > 0 }

}

}

//

// Card.swift

// Concentration //

// Created by student 2018/12/13.

// Copyright © 2018年 fl. All rights reserved.

//

import Foundation

struct Card

{

var isFaceUp = false {

didSet {

if oldValue && !isFaceUp {

isSeen = true

}

}

}

var isMatched = false

private(set) var isSeen = false

var identifier: Int

static var identifierFactory = 0

static func getUniqueIdentifier() -> Int {

identifierFactory += 1

return identifierFactory

}

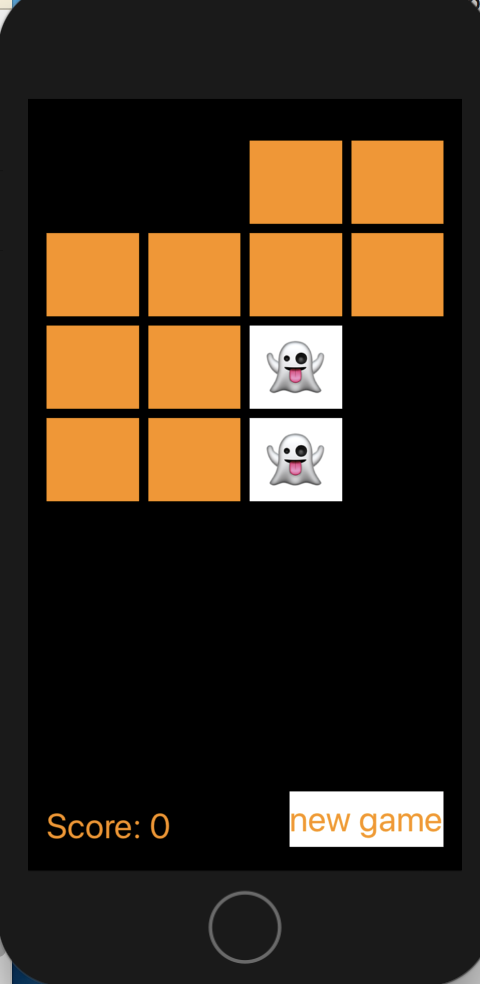
init() {

self.identifier = Card.getUniqueIdentifier()

}

}

* 运行截图：



1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

这次实验对于我来说是比较难的，我就跟着老师的视频和代码打了一遍，加深自己对这一章节内容的理解。

通过这一章内容，我对单mvc有了更加充分的理解。

Github地址：

注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。