实验编号：07 **四川师大《IOS高级开发技术》实验报告 2018** 年**10**月**30** 日

**计算机科学学院** 2016 级4班 实验名称： 作业七 \_

姓名：\_邹琳\_ 学号：\_\_2016110458\_ 指导老师：李贵洋 实验成绩:\_\_\_\_\_

**实验\_07\_ \_\_\_\_\_\_\_\_作业七\_\_\_\_\_\_\_\_\_\_**

1. 实验目的及要求

1、实验目的

1. 实现一款功能完整的game（Concentration）；
2. 掌握单MVC的主要思想；

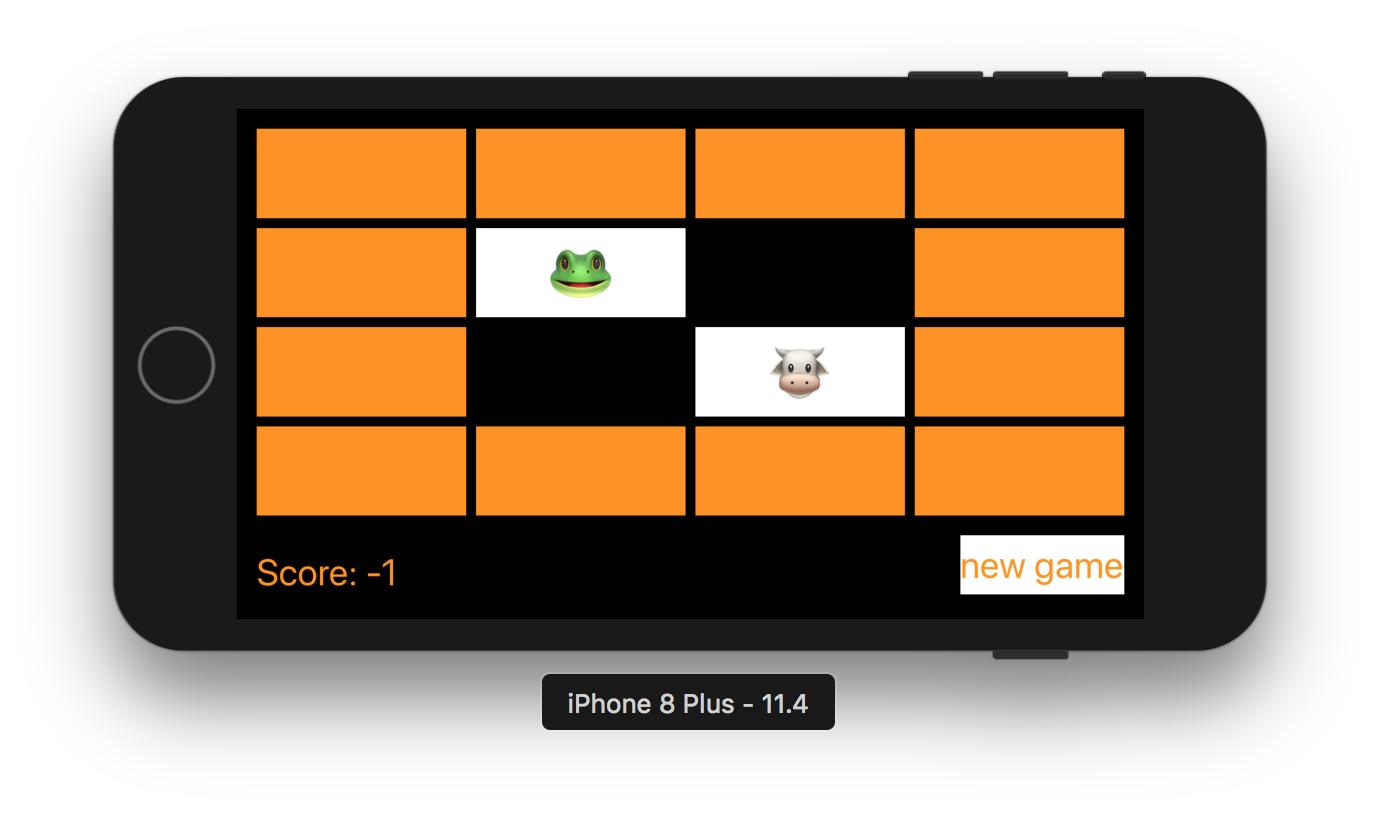
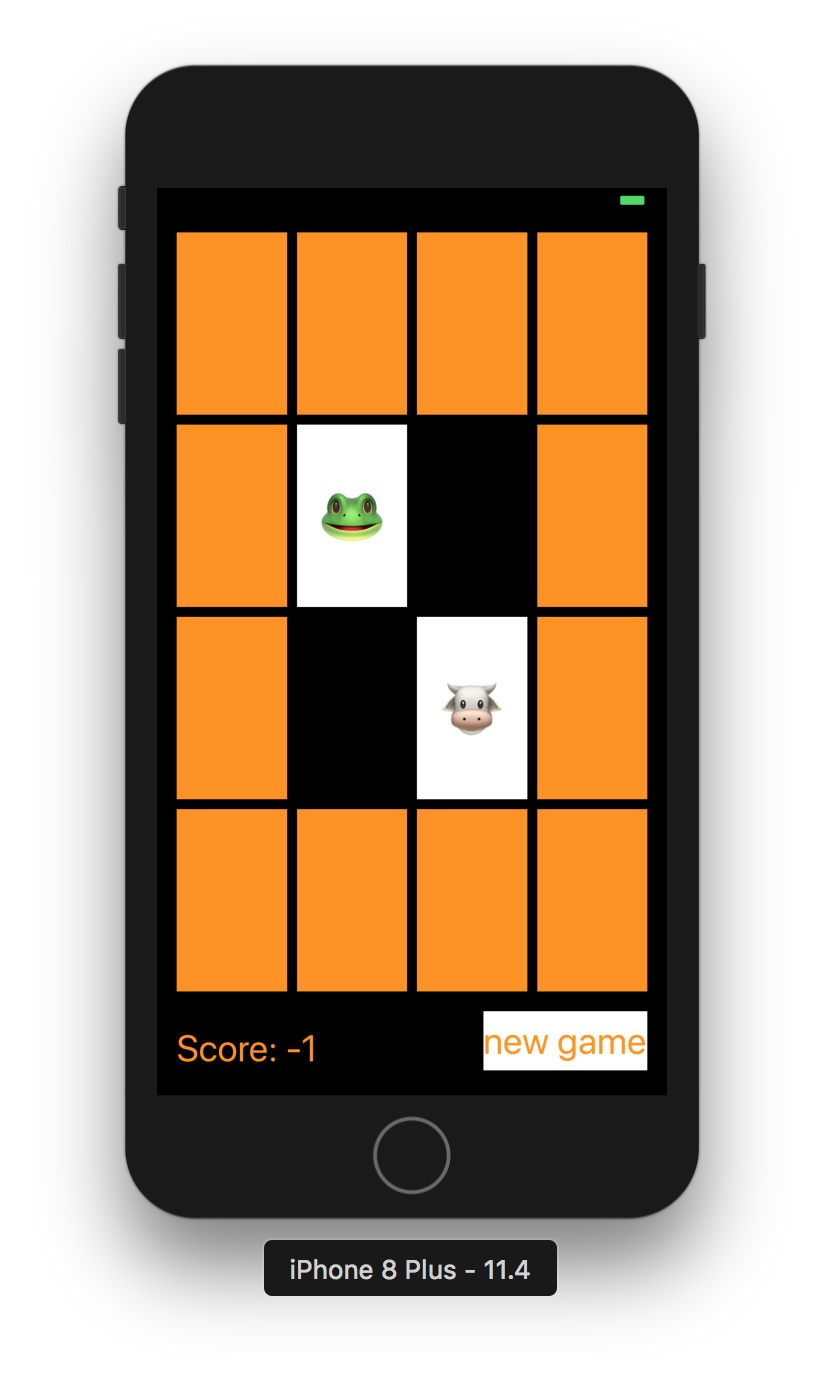
2、实验要求

1. 认真填写实验报告，要求附加部分运行界面和主要代码；
2. 对设计好的程序，检查输出是否符合预期，如有错请分析错误原因并解决；
3. 实验内容

(1)参照Stanford视频1和2完成一个game（Concentration）的制作；

(2)在(1)的基础上进一步完成Stanford Assignment 1的完整要求；

(3)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）

import Foundation

struct Card {

var isFaceUp = false {

didSet {

if oldValue && !isFaceUp {

isSeen = true

}

}

}

//初始为不匹配

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()

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//

// Concentation.swift

// seven

//

// Created by student on 2018/11/22.

// Copyright © 2018年 2016110458. All rights reserved.

//

import Foundation

class Concentation{

//初始化分数为0

var score = 0

//卡片用数组存储

var cards = [Card]()

var indexOfOneAndOnlyFaceUpCard: Int?

//判断卡片内容是否匹配

func chooseCard(at index: Int){

if !cards[index].isMatched{

if let matchIndx = indexOfOneAndOnlyFaceUpCard,matchIndx != index {

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

cards[matchIndx].isMatched = true

cards[index].isMatched = true

score += 2

}else{

if cards[matchIndx].isSeen {

score -= 1

}

if cards[index].isSeen {

score -= 1

}

}

cards[index].isFaceUp = true

indexOfOneAndOnlyFaceUpCard = nil

}else{

for flipDownTndex in cards.indices {

cards[flipDownTndex].isFaceUp = false

}

cards[index].isFaceUp = true

indexOfOneAndOnlyFaceUpCard = index

}

}

}

init(numberOfPairsOfCards: Int){

for \_ in 1...numberOfPairsOfCards{

let card = Card()

cards += [card,card]

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//

// ViewController.swift

// seven

//

// Created by student on 2018/11/24.

// Copyright © 2018年 2016110458. All rights reserved.

//

import UIKit

class ViewController: UIViewController {

//卡片组

@IBOutlet var cardButons: [UIButton]!

//得分显示

@IBOutlet weak var Score: UILabel!

//游戏

lazy var game = Concentation(numberOfPairsOfCards: (cardButons.count + 1) / 2)

//统计游戏得分

var flipCount = 0 {

didSet{

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

}

}

//卡片点击事件

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

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

game.chooseCard(at: cardNumber)

updateViewFromModel()

}else{

print("choose card was not in cardButonds")

}

}

//卡片翻牌事件

func updateViewFromModel(){

for index in cardButons.indices{

let button = cardButons[index]

let card = game.cards[index]

if card.isFaceUp{

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

button.backgroundColor = #colorLiteral(red: 0.2588235438, green: 0.7568627596, blue: 0.9686274529, alpha: 1)

}else{

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

button.backgroundColor = card.isMatched ? #colorLiteral(red: 0.9411764741, green: 0.4980392158, blue: 0.3529411852, alpha: 0) : #colorLiteral(red: 0.1953536868, green: 0.6936447024, blue: 0.9727748036, alpha: 1)

}

}

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

}

//卡片内容

// var emojiChoices = ["👻","🐶","🐯","🐶","🦊","🐸","🦊","🐸","😆","🐵","🐣","🐵","🐣"]

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

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

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

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

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

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

//从0开始选择，每新开始一局游戏就重新选择

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] ?? "?"

}

// var emoji = [Int:String]()

//新游戏

@IBAction func new(\_ sender: Any) {

game = Concentation(numberOfPairsOfCards: (cardButons.count + 1) / 2)

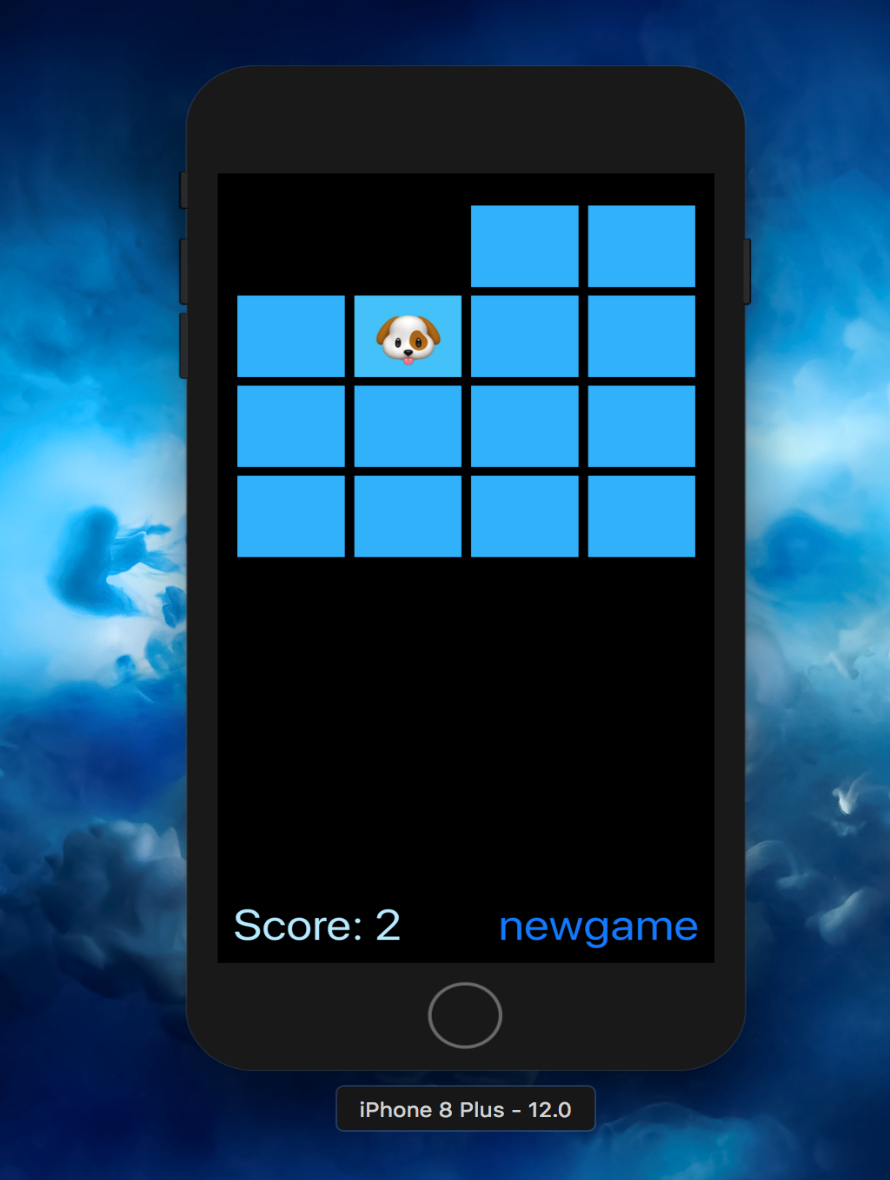
let them = Int(arc4random\_uniform(UInt32(themes.keys.count)))

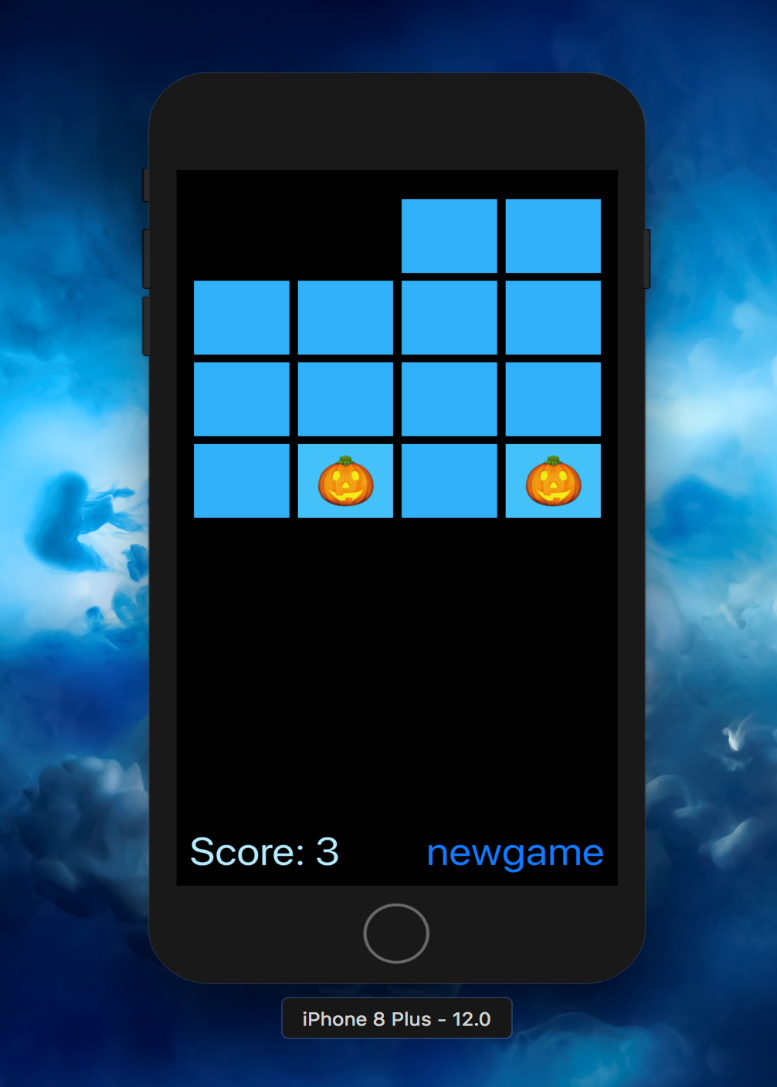
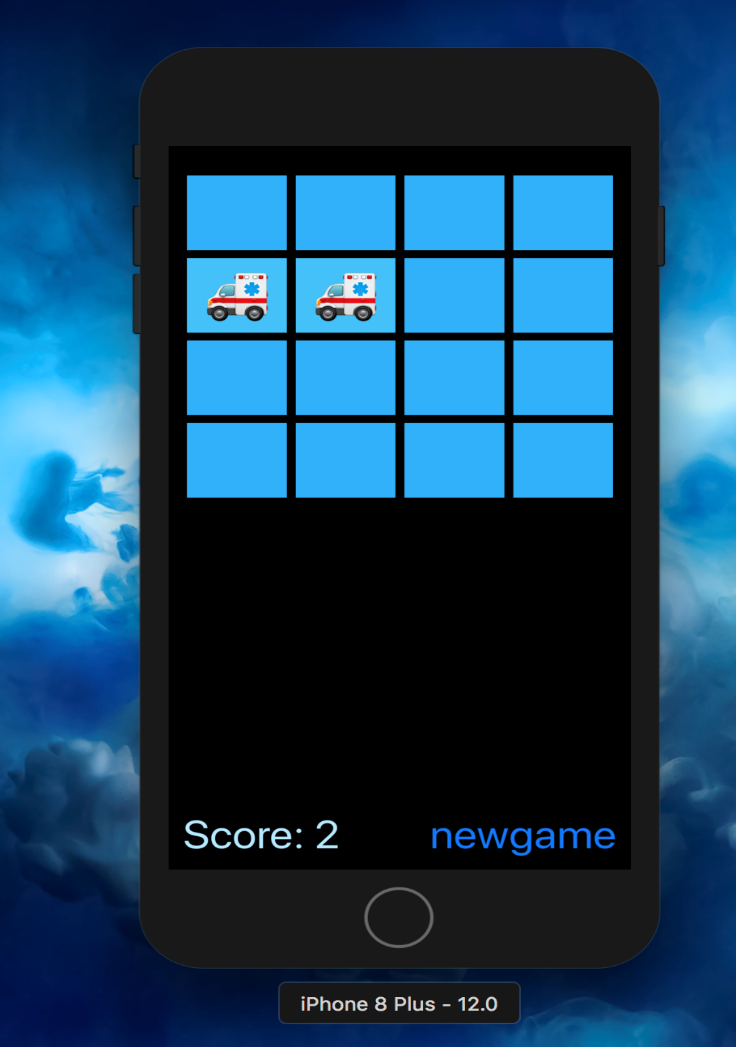
emojiChoices = themes[them]!

updateViewFromModel()

}

}

1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

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。