# Программирование для платформы iOS

## Задание

Необходимо реализовать игру Memorize - на поиск одинаковых карточек среди случайно расположенных карт колоды. Реализация должна включать в себя использование технологии **SwiftUI** и архитектурного паттерна **MVVM**.

## Реализованная функциональность приложения

### 1. Реализовать игру, которую мы делали на протяжении семестра.

Увидеть работу приложения можно на скриншотах ниже.

### 2. Пусть игра начинается со случайного количества пар карт в диапазоне от 2-х пар карт до 5-ти пар карт.

```
func newGame() {
    model.clearModel()
    Int.random(in: 2...5)
        .times { index in
            generateCards(with: "\(index)", number: model.gameRules.requiredCardsCount)
        }

    shuffle()
}
```

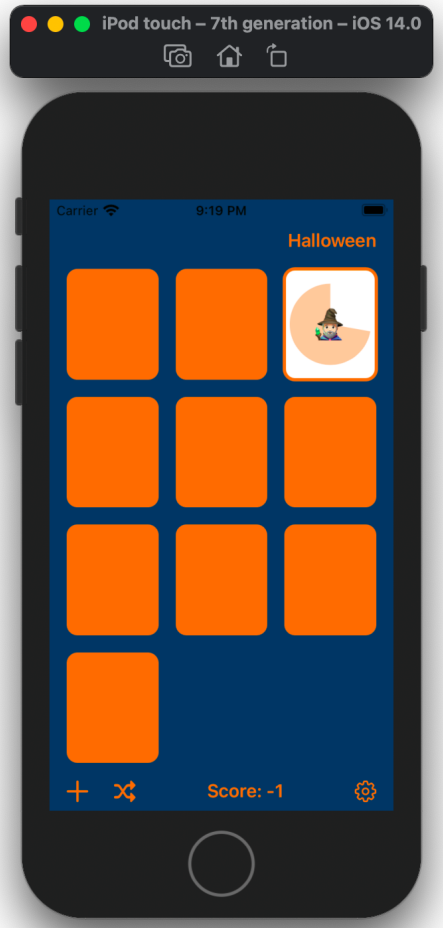
### 3. Добавить в игру кнопки создания новой игры, перемешивания карт.

Соответствующие кнопки находятся в левом нижнем углу


## 4. Формировать счет в игре добавлением 2-х очков за каждое совпадение и штрафом в 1 очко за каждое несовпадение ранее увиденной карты.

```
let hitPrize: Int = 2
let missPenalty = 1

mutating func applyResult(result: GameRulesResolution<CardContent>) {
    switch result {
        case .miss(let missedCards):
            missedCards.forEach { card in
                score -= card.isSeenBefore ? missPenalty : 0
                let choosenIndex = cards.firstIndex(matching: card)!
                cards[choosenIndex].isJustSeen = true
                cards[choosenIndex].isSeenBefore = true
                cards[choosenIndex].isFaceUp = false
            }
```

```
            case .hit(let hittedCards): do {
                score += hitPrize
                hittedCards.forEach { card in
                    let choosenIndex = cards.firstIndex(matching: card)!
                    cards[choosenIndex].isMatched = true
                    cards[choosenIndex].isJustSeen = true
                    cards[choosenIndex].isFaceUp = true
                }
            }
            default: do {
                // nothing to do
            }
        }
    }
```


## 5. Добавлять на ваш UI метку для счета в игре (score label).

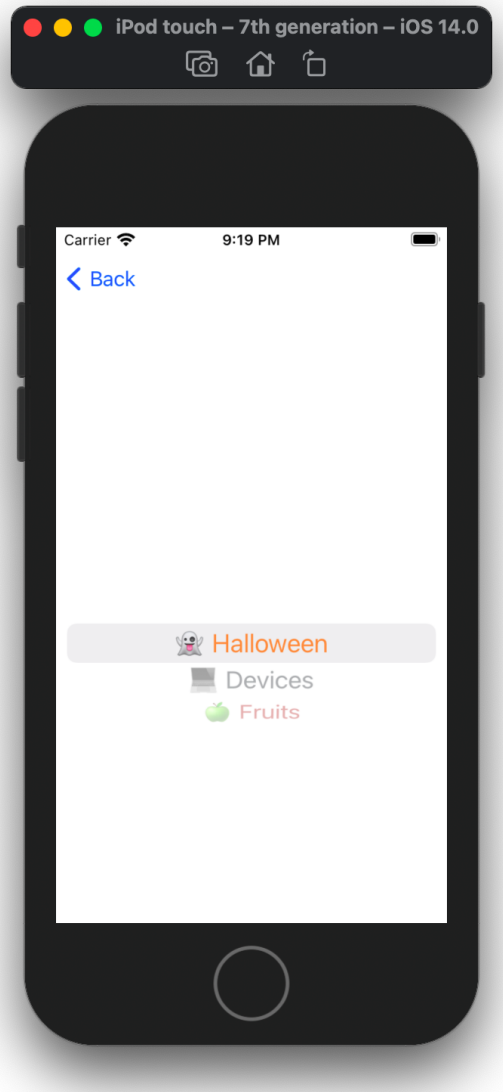

## 6. Ввести в игру концепцию “Тема”. Тема состоит из смены эмоджи, цвета фона, оборотной стороны карты, фрагмента круговой диаграммы.

```
struct Theme: Hashable {
    let name: String
    let emojiTheme: EmojiTheme
    let background: ColorTheme
    let rubashkaColor: ColorTheme
    let pieConfig: PieConfig
}

enum GameThemes: CaseIterable {
    case HALLOWEEN
    case DEVICES
    case FRUITS
```

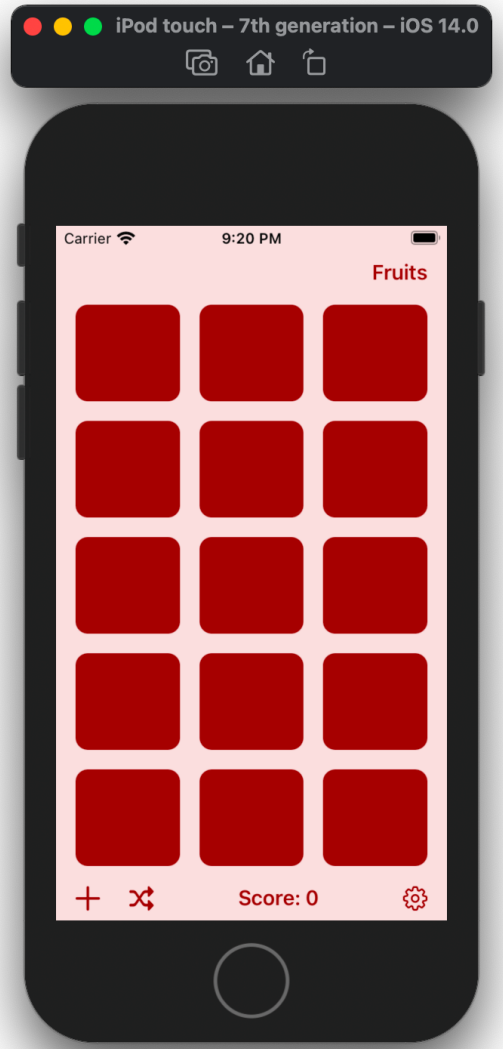
```
    ...
}
```

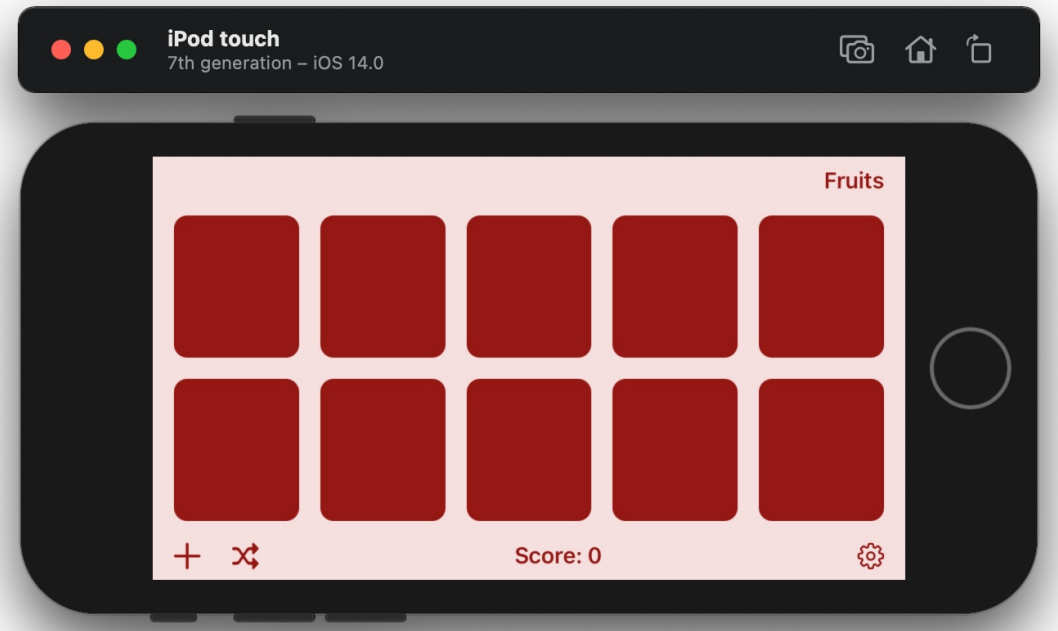
## 7. Предусмотреть не менее 3х тем в игре. В правом верхнем углу отображать отображать название текущей темы.

В приложении реализованы 3 темы: хэллоуин, устройства и фрукты. Выбор темы происходит через экран настроек.




**8. Интерфейс игры должен одинаково хорошо смотреться в любой ориентации.**

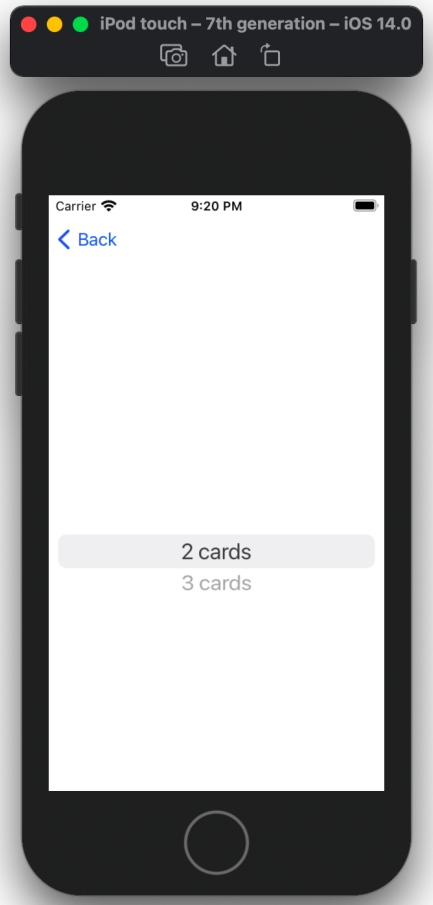

iPod touch – 7th generation – iOS 14.0
Carrier
9:20 PM
Fruits
Score: 0

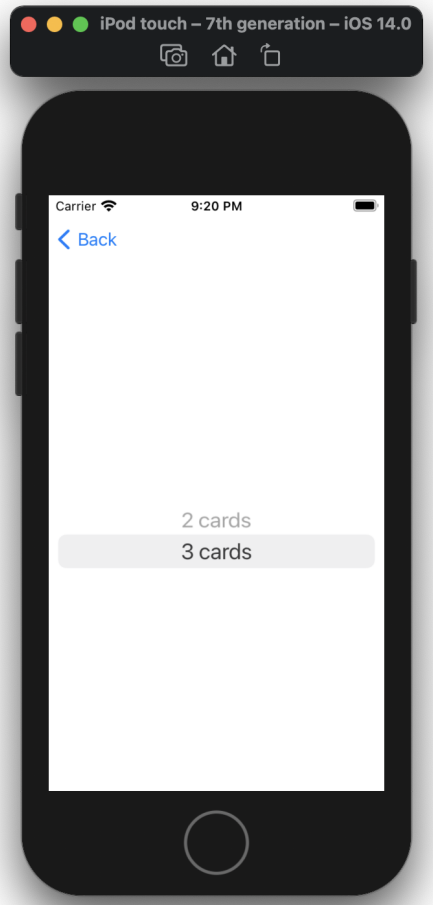



## 9. В рамках того же проекта реализовать аналогичную игру, но с выбором и удалением по 3 карты.

```
class EmojiGameRules {
    static let twoCardGameRule = GameRule<String>(requiredCardsCount: 2)
    static let threeCardGameRule = GameRule<String>(requiredCardsCount: 3)

    static let allRules = [twoCardGameRule, threeCardGameRule]
}

...

func newGame() {
    model.clearModel()
    Int.random(in: 2...5)
        .times { index in
            generateCards(with: "\(index)", number: model.gameRules.requiredCardsCount)
        }

    shuffle()
}
```

## 10. Добавить кнопку выбора правил игры (с 2 картами, с 3 картами).




## Реализация:

- Яппаров Артём. Профиль на github:

https://github.com/ArtJapp

- Рябков Алексей. Профиль на github:

https://github.com/blessedbyjobs