

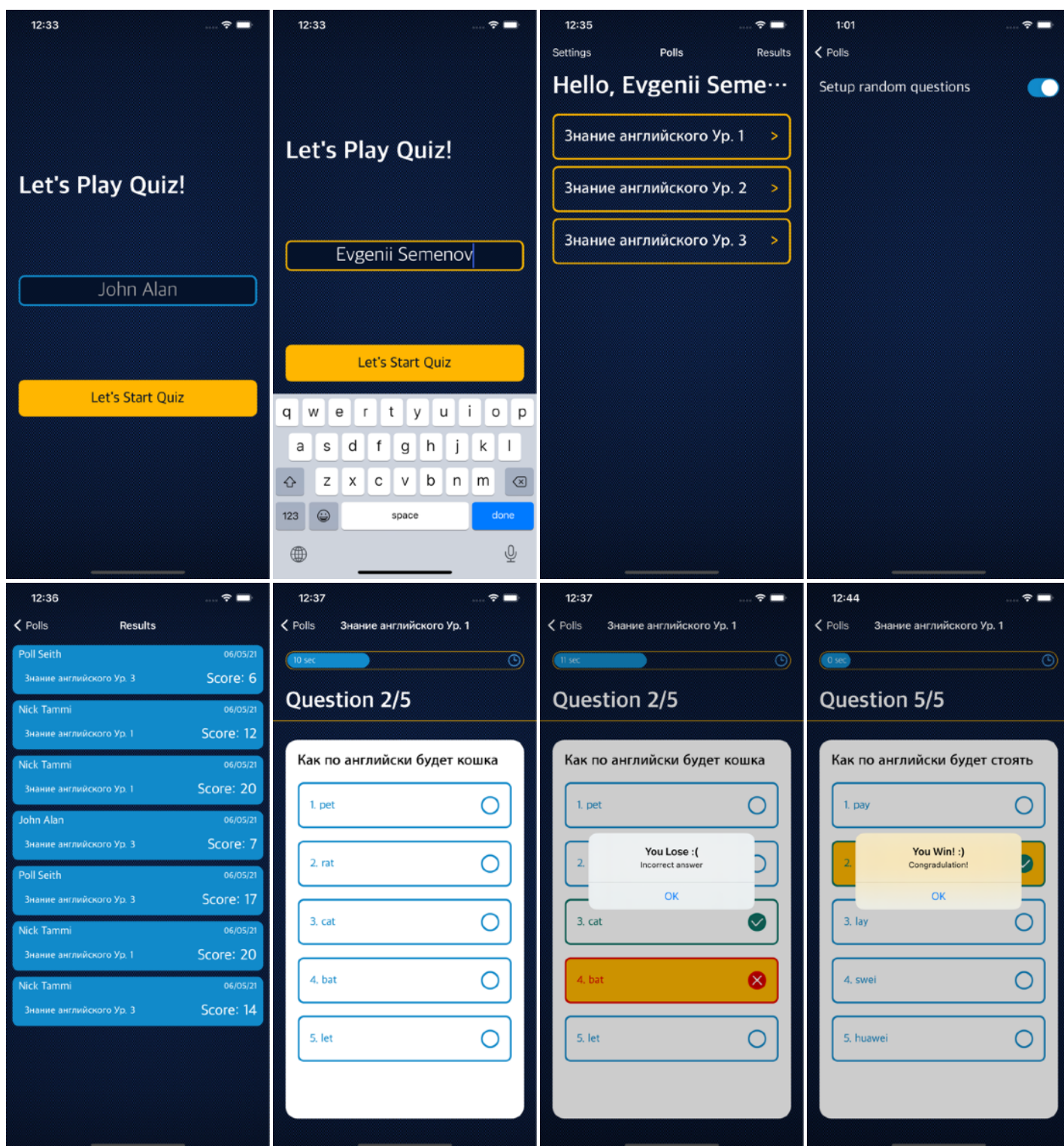
Программа курса "Введение в разработку. Создание приложений с UIKit"

1. ООП, Протоколы, SOLID
2. Git, Sourcetree
3. Интерфейс Xcode, Storyboards, Xib
4. MVC, UIKit, UIViewController, UIView,
5. UILabel, UIButton, UIImage, UIImageView
6. Жизненный цикл приложения, контроллера, вью
7. UITextField, UITextView, UITextFieldDelegate
8. UIStackView, Constraints, Priorities, ViewHierarchyDebugger
9. Промежуточные итоги. Причесывание приложения
- 10.Keyboard notifications
- 11.UISegmentedControl, UIImage, UIImageView, UIActivityIndicator
- 12.UIScrollView, UITableView, UITableViewDelegate/Datasource
- 13.UITableViewCell
- 14.UICollectionView, UICollectionViewLayout
- 15.TabBar, Navigation
- 16.Итоги. Финальная доработка приложения

Приложение

На уроках последовательно рассматривается создание приложение Quize. В приложении зашиты различные тесты/опросы, которые можно проходить в случайном порядке или последовательно. Прохождение тестов/опросов осуществляется за отведенное время. Результаты игры сохраняются в течении времени работы с приложением. По итогу курса вы научитесь использовать Storyboard для верстки адаптивных приложений, создавать компоненты с индивидуальным дизайном, а также грамотно использовать базовые UI-элементы.

Скрины экранов



Домашние задания

1. Готовим структуру проекта
 - A. Создать проект для будущего приложения Quiz (опросник)
 - B. Создать модель данных (структуру) для вопроса с полями:
 - Заголовок
 - Варианты ответа (массив)
 - Корректный ответ (или его индекс)
 - C. Создать модель данных (структуру) для опроса с полями:
 - Название
 - Вопросы (массив вопросов)
 - D. Создать класс для хранения опросов и зашить как минимум один опрос с несколькими вопросами (примеры вопросов можно найти в сети)
2. Подключаем систему контроля версий Git для проекта
 - A. Создать удаленный репозиторий с .gitignore и ReadMe.md файлами
 - B. Клонировать репозиторий локально
 - C. Добавить проект в репозиторий в отдельной feature ветке
 - D. Сделать commit и отправить изменения на удаленный репозиторий
 - E. Сделать pull request с feature ветки проекта в master (или main) ветку
3. Создаем каркас экранов проекта
 - A. В сториборде создать цепочку из экранов:
 - Экран входа
 - Главный экран
 - Экран опроса
 - B. Задать идентификаторы для всех экранов
 - C. Создать xib и swift файлы для view прогресса (ProgressView)
 - D. Настроить связи между xib и swift файлами кастомной view
4. Структурируем проект по архитектуре MVC
 - A. Для каждого экрана сториборда создать классы viewController и view
 - B. Связать классы их в коде и в сториборде
 - C. Создать пустые классы моделей для каждого экрана
5. Верстаем основные экраны приложения
 - A. Добавить заголовок с кнопкой на экран входа. Добавить переход на главный экран по нажатию кнопки
 - B. Добавить заголовок и кнопку начала прохождения опроса на главном экране. Добавить переход на экран опроса по нажатию кнопки опроса
 - C. Добавить ProgressView на экран опроса

6. Реализуем логику прохождения опроса
 - A. Добавить на экране опроса view карточки вопроса (с заголовком и вариантами ответов в виде кнопок)
 - B. При переходе на экран опроса выбираем случайный опрос (если их несколько) и подставляем данные из первого вопроса в карточку вопроса
 - C. Создать AlertController и два варианта его вызова при окончании игры (ответил верно на все вопросы либо ответил не верно)
 - D. После нажатия на вариант ответа проверяем его корректность и либо отображаем следующий вопрос (если вопросы кончились, отображаем сообщение о выигрыше), либо сообщение о проигрыше
7. Добавляем игрока в приложение
 - A. На экране входа добавить текстовое поле для ввода имени
 - B. Запретить начало игры без ввода корректного имени
 - C. Передать имя игрока на главный экран для заголовка приветствия
 - D. Передать имя игрока на экран опроса для отображения в сообщениях
8. Настраиваем поддержку разных устройств
 - A. Добавить все элементы главного экрана в stackView, настроить их расположение вверху экрана, над клавиатурой
 - B. На главном экране добавить кнопки опросов в stackView (если опрос один, добавить еще несколько опросов в приложение), настроить расположение заголовка приветствия и кнопок на экране
 - C. Задать констрейнты для ProgressView на экране опроса, настроить корректное отображение карточки вопроса на разных экранах
9. Полируем приложение
 - A. Добавить изображение на фон для экранов игры
 - B. Задать основные цвета приложения и использовать их на экранах
 - C. Добавить поддержку темной темы
 - D. Добавить таймер на экране опроса для изменения ProgressView
 - E. Сбрасывать таймер после ответа и останавливать в конце игры
 - F. Добавить завершение игры, если время вышло
10. Поддержка клавиатуры на экране входа
 - A. Разместить stackView экрана входа по центру
 - B. Добавить подъем stackView при отображении клавиатуры
11. Улучшаем логику игры
 - A. Добавить экран настроек приложения и переход с главного экрана
 - B. Добавить переключатель для выбора последовательного или случайного вывода вопросов опроса

- C. Создать глобальное перечисление Settings для хранения настроек приложения (статическое свойство)
- D. Применить настройку на экране опроса (перемешивать вопросы, если выбран случайный порядок)

12. Добавляем результаты игр

- A. Добавить в приложение модель результата игры с полями:
 - Имя игрока
 - Счет игры
 - Дата и время игры
- B. Добавить в Settings статическое свойство для хранения результатов игр (массив)
- C. На экране опроса реализовать добавление результата после прохождения опроса
- D. Добавить экран результатов (TableViewCell) и переход с главного экрана
- E. Добавить на экране результатов отображение сохраненных результатов

13. Кастомизируем результаты игры

- A. Добавить xib и swift файлы для ячейки результата, сверстать ее
- B. Использовать кастомную ячейку для отображения результатов игр

14. Улучшаем приложение

- A. Заменить stackView опросов на главном экране на TableView
- B. Создать xib и swift файлы для ячейки опроса, сверстать ее и использовать ее в таблице главного экрана
- C. Заменить кнопки вариантов ответов в карточке вопроса на collectionView
- D. Создать xib и swift файлы для ячейки варианта ответа, сверстать ячейку и использовать для отображения вариантов ответа
- E. Добавить смену дизайна ячейки при выборе варианта ответа

15. Навигация в приложении

- A. Добавить UINavigationController для экрана входа в приложение
- B. Добавить заголовки для каждого экрана
- C. Кнопки настроек и результатов главного экрана перенести в NavigationBar
- D. Поменять отображение всех экранов на полноэкранное

16. Финальные штрихи

- A. Проверяем корректность работы приложения, исправляем ошибки