

**В.Н. Леванцов, В.С. Давыдов**

**УО «Гомельский государственный университет  
имени Франциса Скорины», Гомель, Беларусь**

## **РАЗРАБОТКА WEB-ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ XCODE**

### **Введение**

Высокие технологии и микрокомпьютеры заняли значительную часть жизни человека. С движением технологического прогресса составные части компьютера становятся всё меньше, что позволяет уменьшать и форм-фактор компьютеров. В данный момент на рынке активное распространение получают планшетные компьютеры – устройства, совмещающие в одном корпусе системный блок, монитор и

устройства ввода-вывода. Главным отличием планшетных компьютеров от настольных является отсутствие у них таких устройств ввода-вывода как мышь и клавиатура – они заменены сенсорным экраном. В таких компьютерах пользователь интуитивно прикасается к элементам интерфейса для взаимодействия с ними, при необходимости ввода текста – на экране появляется виртуальная клавиатура, у которой за кнопки ответственны отдельные участки экрана.

Безусловным лидером рынка планшетных компьютеров в текущее время является Apple iPad, работающий под управлением операционной системы iOS. Для разработки приложений используется набор инструментов iOS SDK, включающий в себя среду разработки Xcode, редактор интерфейса Interface Builder, симулятор iOS Simulator и ряд инструментов для отладки, оптимизации и публикации приложений, а также обширную документацию.

### **1. Интерфейс программирования приложений на платформе Xcode**

Cocoa Touch – это интерфейс программирования приложений на платформе Xcode для создания программ, исполняемых на платформах iPhone, iPod Touch и iPad компании Apple.

Cocoa Touch представляет из себя ступень абстракции операционной системы iOS. Он базируется на Mac OS X Cocoa API и, также как и он, в основном написан на Objective-C. Cocoa Touch позволяет использовать аппаратные средства не доступные на компьютерах Mac OS X, которые являются специфическими для мобильных устройств (акселерометр, компас, вспышка и пр.). Cocoa Touch использует шаблон проектирования Model-View-Controller (MVC) в своей архитектуре. Технологии операционной системы iOS могут быть представлены как набор уровней абстракции с Cocoa Touch на самом высоком уровне и Core OS вместе с ядром Mac OS X – на самом низком. Это делает удобным доступ к различным сервисам и технологиям, уменьшает затраты на разработку и освобождает разработчика от решения задач низкоуровневого программирования. Cocoa Touch включает в себя широкий спектр библиотек (фреймворков) наиболее важными из которых являются:

- Foundation Kit Framework – базовый набор объектов языка (строки, коллекции, математические функции и пр.);

- UIKit Framework – набор инструментов для создания пользовательского интерфейса;

- Game Kit Framework – компоненты для организации игровых приложений используя специальный сервис от компании Apple – Game Center;

- Map Kit Framework – набор классов для работы с картами

местности и их отображения.

В фреймворке UIKit определено множество классов, необходимых для построения и управления пользовательским интерфейсом программы.

В нем представлены: объект самого приложения UIApplication, объект события, окна, представления, кнопки и многие другие элементы интерфейса.

Для управления поведением приложения определён протокол UIApplicationDelegate. В нём объявлены методы, которые реализуются синглтон-объектом класса UIApplication. Эти методы предоставляют разработчику информацию о ключевых событиях исполнения приложения, таких как окончание загрузки, момента, когда выполнение приложения будет прекращено, при нехватке памяти и некоторых других важных изменениях в поведении приложения.

Одной из самых главных задач делегата приложения является отслеживание состояния приложения. До появления операционной системы iOS 4.0 приложение имело только три состояния: активное, неактивное и нерабочее. В iOS 4.0 и более поздних приложения также могут работать в фоновом режиме и быть приостановленными.

Задачи, реализуемые наследниками протокола:

- наблюдение за изменениями состояния приложения;
- открытие URL во встроенном браузере;
- управление изменениями панели статуса устройства;
- ответ на системные уведомления;
- обработка удалённых уведомлений;
- обработка локальных уведомлений;
- ответ на изменения состояния защиты.

Стандартным для iPhone-приложения является построение по шаблону MVC (модель – представление – контроллер). В качестве представления обычно используются объекты класса UIView и его наследники.

Класс UIView, как и все классы с префиксом UI в названии (указывает на принадлежность в фреймворку UIKit), отвечает за представление данных на экране устройства.

Объект класса определяет прямоугольную область на экране и интерфейсы для управления содержимым этой области. Во время выполнения программы объект управляет отрисовкой своего содержимого, также он обрабатывает любое взаимодействие с этим содержимым.

Класс UIView определяет для себя самого в качестве базового поведения заполнение области, им занимаемой, фоновым цветом.

Поскольку объекты UIView являются главным способом для взаимодействия приложения и пользователя, на них возложен ряд

обязанностей.

Рисование и анимация:

- представление отрисовывает содержимое с использованием технологий UIKit, Core Graphics, и OpenGL ES;

- некоторые свойства представления могут менять свои значения анимированно.

Размещение и управление субпредставлениями:

- представление может иметь от нуля до огромного количества субпредставлений (subview);

- каждое представление определяет своё собственное поведение по умолчанию по изменению размеров в соответствии с событиями на родительском представлении;

- представление может напрямую изменять размеры и положение своих субпредставлений по необходимости.

Обработка событий:

- представления является так называемым респондером (ответчиком) и может обрабатывать касания экрана и другие события, определённые в родительском классе UIResponder.

Представления могут использовать метод `addGestureRecognizer`: для добавления к своему поведению просмотрщика, опознающего наиболее стандартные жесты пользователя (например, двойное касание, щипок, жест прокручивания и т. д.)

На представление можно добавлять другие представления, создавая при этом сложную иерархическую визуальную структуру. Между представлением и его субпредставлением создаётся отношение родитель – потомок. В терминах языка главное представление называется `Superview`, а субпредставление – `Subview`. Как правило, видимая область субпредставления может выходить за границы родительского представления, однако можно использовать свойство `clipsToBounds` – обрезать по границам, изменяющего это поведение на противоположное. Родительское представление может содержать любое количество субпредставлений, однако у субпредставления может быть только одно родительское, которое отвечает за размещение своих субпредставления должным образом.

## 2 Примеры реализации приложений

### 2.1 Приложение для обучения детей счёту

Разработанное приложение является игровым, его целевой аудиторией являются дети дошкольного возраста. Это выдвигает требования к простому и максимально понятному пользовательскому интерфейсу, красочному дизайну и интуитивным взаимодействием с пользователем.

Экран приложения состоит из игрового поля, на котором

отображаются счётные палочки, меню создания новых объектов, панели заданий, панели «Share», кнопок выбора режима, смены фона и переключения на экран общей информации.

Процесс взаимодействия с приложением представляет собой расширенный для режима множественных прикосновений Drag-and-Drop-интерфейс (поднял и положил). Пользователь, используя меню создания объектов, вытаскивает на игровое поле необходимые ему счётные палочки. Каждой палочке на экране соответствует спрайт, имеющий положение, определённый угол поворота и связанное с ним физическое тело, которое в процессе симуляции физического мира подвергается силам со стороны прикосновений пользователя, сталкивается с другими телами и реагирует на события игрового поля. В результате симуляции тело меняет своё положение и это немедленно отображается на игровом поле.

## 2.2 Приложение для организации и проведения мероприятий

Данное приложение предназначено для поиска интересующих человека мероприятий, просмотре информации по данному мероприятию, а также просмотр участников данного мероприятия. Если данное событие заинтересовало человека, то пользователь приложения всегда может добавить данное событие в календарь, и коммуникатор по мере наступления данного события будет напоминать человеку о нем.

Данное приложение взаимодействует с популярными социальными сетями: facebook, twitter, foursquare. Приложение получает данные от сервера, основанного на платформе drupal.

Для работы с приложением не обязательно регистрироваться, но для работы с функциями программы, завязанными на социальные сети необходимо авторизироваться в данных социальных сетях.

Организаторы мероприятия должны подготовить данные о проводимом мероприятии на сервере drupal. При запуске приложения на iPhone оно отправляет запрос на сервер и получает с него данные. После получения данных программа отображает их.

Программа также обладает и другими функциями. Программа может расшифровывать QR коды, нанесенные на различные предметы и здания. Также реализована прокладка маршрута между текущим положением пользователя и местом, куда пользователю необходимо добраться. В программу также добавлена функция расширенной реальности, т. е. программа на экране iPhone должна показывать направление к требуемому объекту. Также программа позволяет отмечаться на местах зарегистрированных в системе foursquare а также добавит возможность размещать фотографии на стене facebook.

## 2.3 Приложение для просмотра видеофайлов из социальных сетей

Это приложение построено по принципу tab-based, то есть его

интерфейс основан на использовании вкладок с представлениями, переключение между которыми производится посредством tab bar'a – полосы управления вкладками, расположенной в нижней части экрана приложения.

Поскольку iPhone имеет встроенный датчик, отслеживающий изменение положения устройства (акселерометр), имеется возможность разрабатывать приложения, изменяющие свой интерфейс при изменении положения (повороте) устройства. Всего положений интерфейса два – ландшафтное (landscape) и портретное (portrait). Ландшафтная ориентация – это когда устройство расположено своей более широкой стороной параллельно горизонту, портретная – более узкой. Поскольку, с точки зрения разработчика, наиболее удобно управлять работой приложения в ландшафтном режиме, то приложение SMPlayer поддерживает только эту ориентацию интерфейса. При повороте устройства из ландшафтного режима в портретный интерфейс не изменяется, при повороте же из ландшафтного режима на 180 градусов, интерфейс также разворачивается на такой же угол, снова оказываясь в ландшафтной ориентации по отношению к пользователю.

При первоначальном запуске приложения пользователь вначале проходит процедуру аутентификации, после чего, в случае удачи, приложение переходит в основной режим работы.

Всего в приложении имеется четыре вкладки: поиск, мои записи, плейлисты, профиль. Также на таббаре имеется кнопка переключения между режимами аудио и видео. Бар можно убрать с экрана, потянув вниз пальцем, при этом в нижнем правом углу появляется закладочка, за которую бар можно вытянуть обратно.

### **Заключение**

Разработанные приложения на платформе XCODE, ориентированны на любого пользователя который умеет пользоваться программами на смартфонах фирмы Apple. Они просты в употреблении, не требуют много сетевого трафика для получения данных и позволяет в последующем работать с основными функциями без Интернета. Для многих функций Интернет все же требуется.

### **Литература**

1. Лебедев, В.А. Объектно-ориентированное программирование: учеб. пособие / В.А. Лебедев, А.О. Ноготков. – Уфа: [б. и.], 1995. – 56 с.
2. Хиллегасс, А. Программирование в среде Сосоа для Mac OS X: обучение для профессионалов / А. Хиллегасс. – М.: ЭКОМ, 2009. – 433 с.