iOS Human Interface Guidelines — Part 1 | Part 2 | Part 3

ОГЛАВЛЕНИЕ

Основы UI дизайна

- Создание дизайна для iOS
- Строение приложения на iOS
- Адаптивность и layout
- Запуск и выход из приложения
- Навигация
- Модальный контекст
- Интерактивность и фидбек
- Анимация
- Брендирование
- Цвет и типография
- Иконки и графика
- Терминология и формулировки
- Интеграция с iOS

Дизайн стратегии

- Принципы дизайна
- От концепции к продукту
- Реальный опыт: от десктопной версии к iOS

Основы UI дизайна

Создание дизайна для iOS

Платформа iOS основана на трех основных темах:

- **Отношение с почтением.** Пользовательский интерфейс помогает людям понимать контент и взаимодействовать с ним, но никогда не соревнуется с контентом в полезности.
- **Ясность.** Текст легко читается при любом размере, иконки четкие и понятные, графические элементы не режут глаз и сочетаются с общей темой. Усиленный фокус на функциональности является основополагающей частью разработки приложения.
- **Глубина.** Заметные невооруженным глазом слои и реалистичное движение придают приложению «живой» вид и позволяют людям восхищаться дизайном и понимать его.

Неважно, создаете ли вы новое приложение или обновляете имеющееся, начинайте работу со следующих шагов:

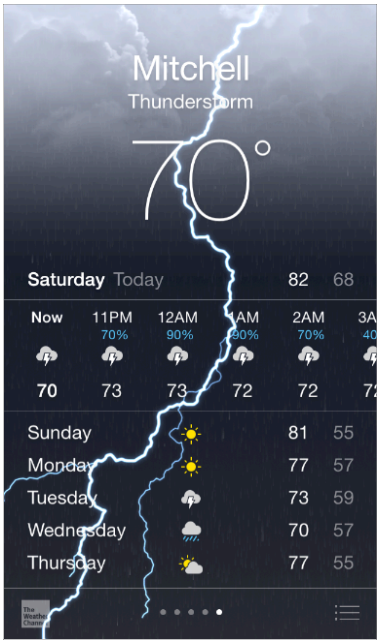
- Шаг 1. Взгляните дальше, чем пользовательский интерфейс сам по себе. Взгляните на основные функции приложения и удостоверьтесь, что этот функционал все еще актуален.
- Шаг 2. Используйте темы iOS для информации о разработке пользовательского интерфейса и опыта пользователя приложения. Добавляйте детали и графические элементы с осторожностью и никогда не делайте этого без особой причины.
- Шаг 3. Удостоверьтесь, что созданный пользовательский интерфейс адаптируется к различным устройствам и режимам работы – так пользователи будут с удовольствием пользоваться вашим приложением в самых различных ситуациях.

В процессе разработки приложения будьте готовы бросать вызов старым правилам, задавать вопросы и дать возможность фокусу на контенте и функциональности мотивировать каждое ваше решение касательно дизайна приложения.

Почтительное отношение к контенту

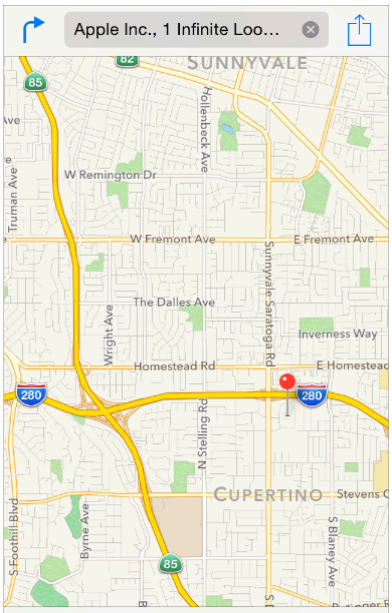
Несмотря на то, что живой и красивый пользовательский интерфейс и плавные движения являются одной из особенностей работы на iOS, контент пользователя является основой всего.

Вот несколько способов убедиться в том, что ваш дизайн улучшает функционал и с почтением относится к контенту.

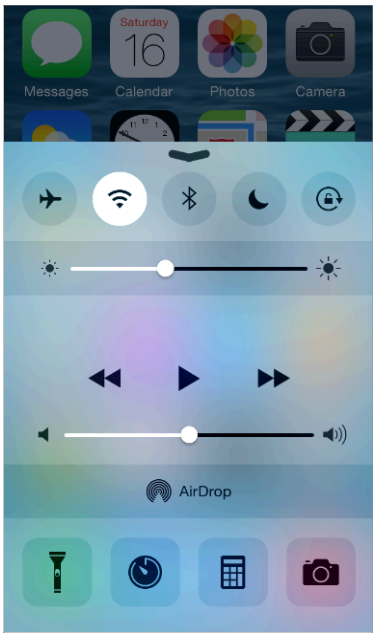


Используйте экран на полную. Приложение “Погода” - отличный пример такого подхода: красивое отображение текущей погоды в определенном местоположении во весь экран моментально передает нужную информацию и освобождает место на экране для информации о ежечасном изменении погоды.

Пересмотрите визуальные указатели, когда речь идет о телесности и реализме. Обрамления, градиентная заливка и отбрасываемые тени иногда “утяжеляют” элементы пользовательского интерфейса и могут пересиливать или соревноваться с контентом. Вместо этого сфокусируйтесь на



контенте и позвольте пользовательскому интерфейсу играть роль второго плана.



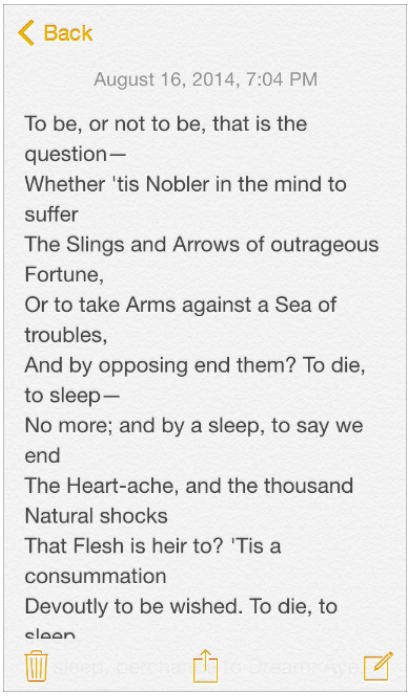
Позвольте полупрозрачным элементам пользовательского интерфейса намекать на спрятанный за ними контент. Полупрозрачные элементы пользовательского интерфейса - например, как в Центре Управления - обеспечивают контекст, помогают пользователю убедиться, что дополнительный контент также доступен и сигнализируют возможную быстроту действий. В iOS полупрозрачные элементы затеняют только контент, находящийся непосредственно за элементом - это создает впечатление, что пользователь смотрит на них через рисовую бумагу. Остальные элементы на экране остаются незатемненными.

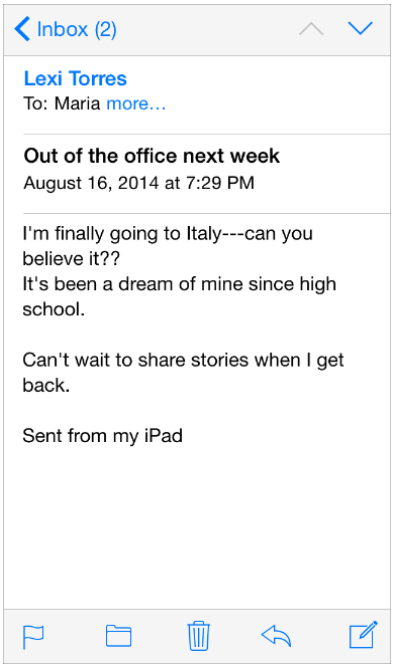
Делаем приложение ясным и понятным

Ясность и понятность - еще один способ удостовериться в том, что контент лежит в основе вашего приложения. Вот несколько способов сделать самую важную часть контента и функционала понятной и удобной в использовании.

Используйте много негативного отражающего пространства. Негативное пространство делает важный контент и функционал более заметными и понятными. Негативное пространство также часто придает ощущение спокойствия и безмятежности. Оно также наделяет приложение такими качествами как целостность и эффективность.

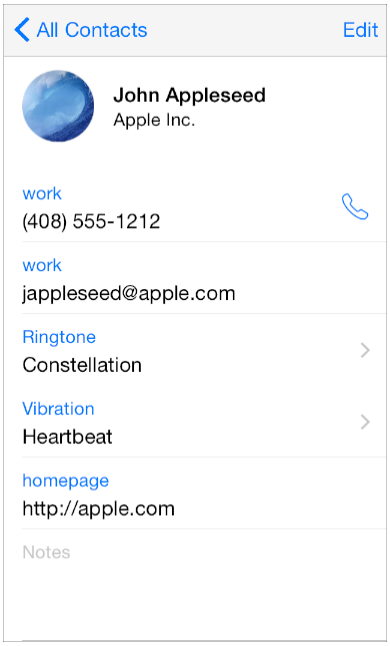
Позвольте цветам упростить пользовательский интерфейс. Основной цвет - например желтый в приложении Notes - выделяет важные части структуры и ненавязчиво указывает на способы взаимодейтсвия. Он также создает целостную цветовую тему приложения. Встроенные приложения в iOS используют серию отчетливых и ясных цветов, которые хорошо видны в любых оттенках и на светлом, и тёмном фоне.





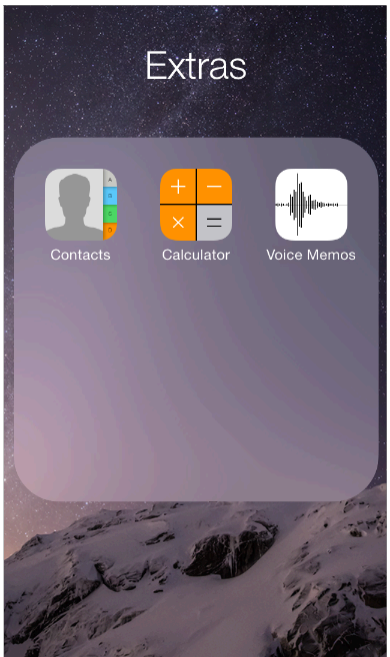
Удостоверьтесь в том, что текст легко читается, используя предустановленные в системе шрифты. Предустановленные в iOS шрифты автоматически подстраивают размер букв и расстояние между ними, чтобы они легко читались и красиво смотрелись при любом размере шрифта. Независимо от того, используете ли вы предустановленные или дополнительные шрифты, обязательно используйте шрифты Динамического типа, чтобы приложение реагировало на выбор разного размера текста пользователем.

Активно используйте кнопки без видимых границ.
По умолчанию кнопки на панели не имеют визуальных границ. В зоне расположения контента кнопки обозначаются контекстом, цветом и названием, которые призывает к действию. Все это указывает на возможность взаимодействия с кнопкой. Когда в этом есть необходимость, кнопка в зоне расположения контента может иметь тонкую границу или слегка подкрашенный фон, чтобы выделяться из контента.



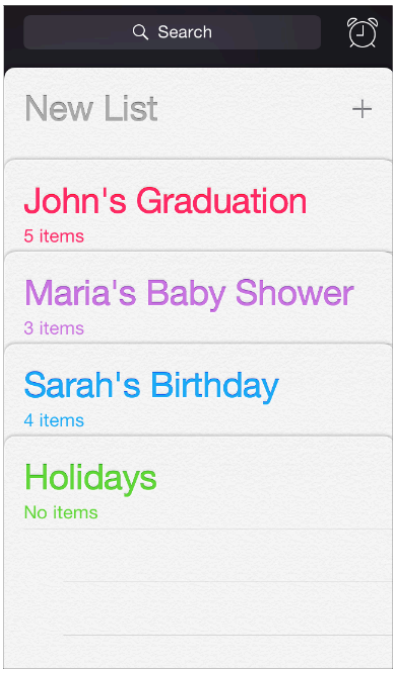
Используйте глубину для передачи информации

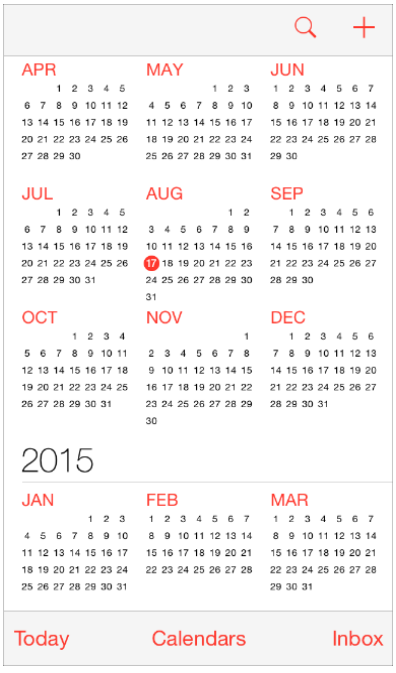
iOS часто представляет контент в виде нескольких видимых невооруженным взглядом слоёв, которые символизируют иерархию и позиционирование. Это помогает пользователям понять взаимосвязь между объектами, изображенными на экране.



С помощью полупрозрачного фона и эффекта “плавания” над главным меню, содержимое папки отделяется от остальной части экрана.

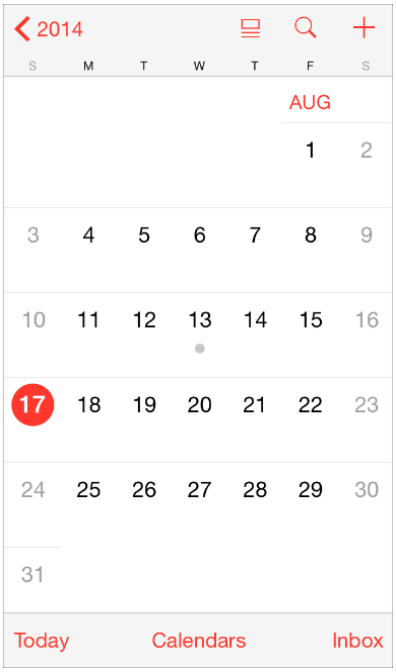
Напоминания отображаются в виде слоёв, как показано на картинке. Когда пользователь работает с одним из списков, остальные списки собираются в один в нижней части экрана.

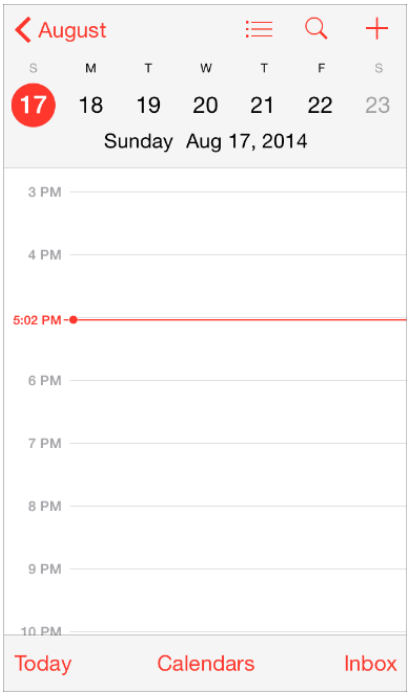




В Календаре используются усиленные переходы, которые позволяют пользователю видеть иерархию и глубину дизайна, когда они перемещаются между отображаемыми годами, месяцами и днями. При просмотре года, как показано на картинке, пользователи могут сразу видят сегодняшний день и могут использовать другие функции календаря.

Когда пользователь выбирает определенный месяц, обзор года приближается до отображения месяца. Сегодняшний день остается подсвеченным, а год указан на кнопке “Назад”. Таким образом пользователь знает, что они смотрят сейчас, откуда они пришли на эту страничку и как вернуться назад.



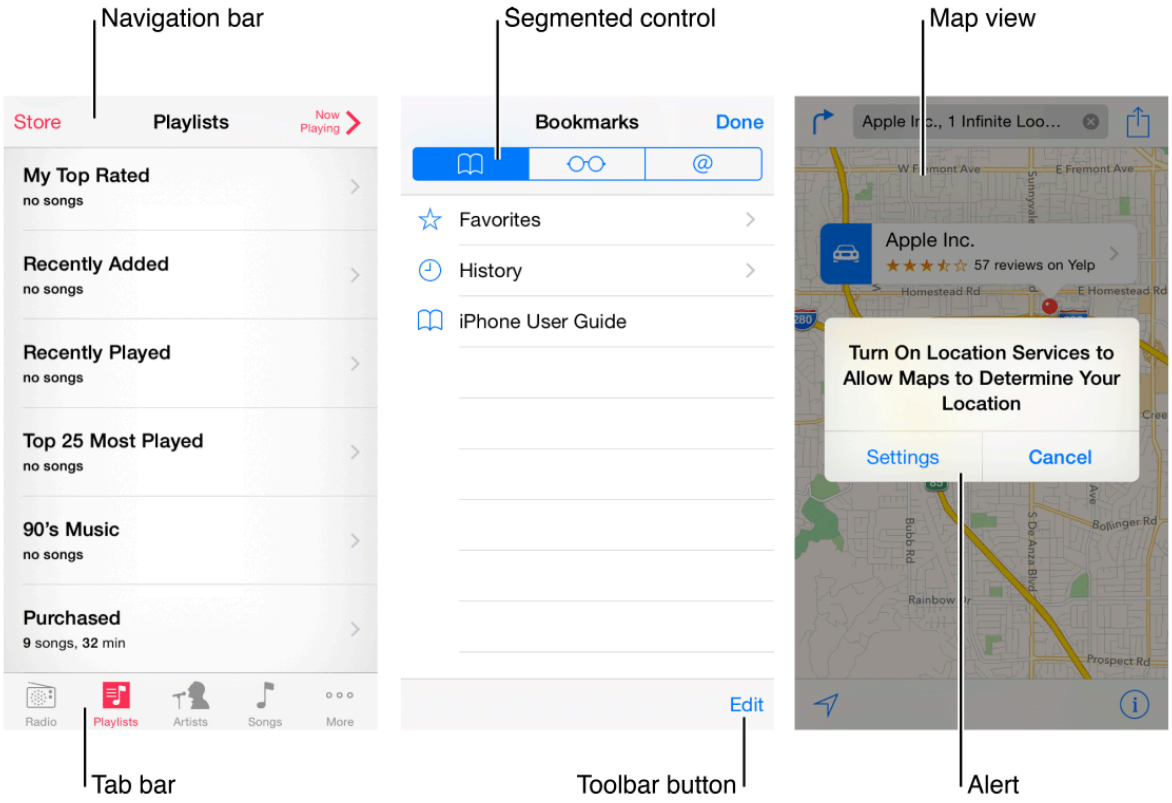


Похожий переход происходит, когда пользователь выбирает определенный день: отображаемый месяц как будто раскалываться и выталкивает текущую неделю на верхнюю часть экрана и открывая почасовое отображение выбраного дня. С каждым переходом Календарь усиливает отношения иерархии между годами, месяцами и днями.

< Назад к оглавлению

Строение приложения на iOS

Практически все приложения на iOS в той или иной мере используют компонтнты пользовательского интерфейса, описанные во фреймфорке UIKit. Знание названий, ролей и возможностей базовых компонентов поможет вам принимать обдуманные решения при разработки пользовательского интерфейса вашего приложения.



Компоненты пользовательского интерфейса, предлагаемые UIKit можно разделить на 4 большие группы:

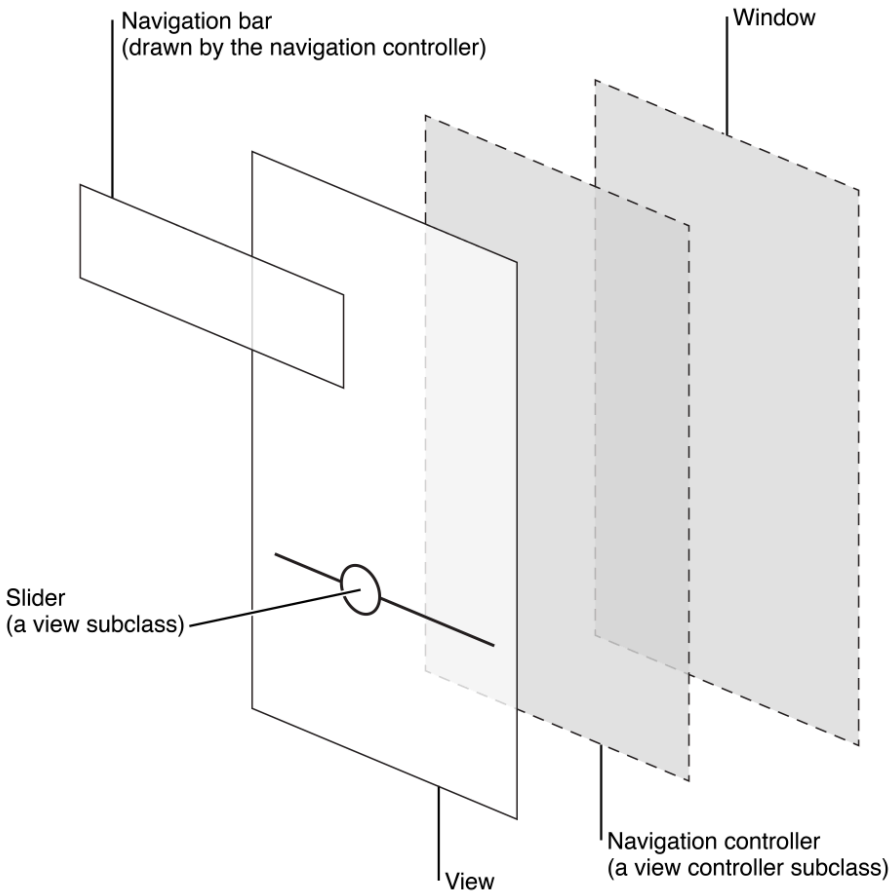
- **Панели (bars).** Панели содержат в себе вытекающую из контекста информацию, которая помогает пользователю сориентироваться, где он находится и элементы управления, которые помогают пользователю перемещаться по приложению или инициировать действия.
- **Content views.** Content views позволяют взаимодействовать с контентом приложения, например пролистывать его, вставлять, удалять и переставлять части контента.
- **Controls.** Controls выполняют действия или отображают информацию.
- **Temporary views**. Temporary views эелементы открываются на короткий период времени с целью сообщить пользователю важную информацию или предложить дополнительные опции и функционал.

В дополнение к основополагающим элементам пользовательского интерфейса, UIKit также включает в себя объекты, которые запускают различные функции: распознавание жестов, рисование, доступность и поддержка печати.

С точки зрения программирования, каждый элемент пользовательского интерфейса - это представление данных, они унаследовали это от системы UIView. Каждое представление данных знает, как отображаться на экране. Элементы управления (такие как кнопки или слайдеры), элементы просмотра контента (такие как просмотр колекций или просмотр таблиц) и элементы временного просмотра (такие как напоминания и списки действий) являются способами представления данных.

Чтобы управлять всей этой иерархией представлений данных в приложении, чаще всего используется view controller. View controller координирует отображение представлений, осуществляет функционал, следующий из действий пользователя, и осуществляет переходы от одного экрана к другому. Например, “Настройки” использует navigation controller, чтобы отобразить иерархию представлений.

Вот пример того, как представления данных и view controller могут сочетаться между собой, создавая пользовательский интерфейс в приложении для iOS.



Хотя разработчики думают в масштабах пресдтавлений данных и view controllers, пользователи воспринимают приложение на iOS как коллекцию экранов. С этой точки зрения, под экраном понимается отличительный визуальный вид или режим приложения.

> Примечание:
> В приложениях для iOS есть окна. Но в отличии от окон в компьютерных приложениях, окна в iOS не имеют видимых очертаний и не могут быть передвинуты в другую часть экрана. Большинство приложений для iOS имеют только один экран; приложения, которые поддерживают выход на внешний экран, чаще всего имеют больше одного экрана.

В *iOS Human Interface Guidelines*, слово *экран* используется в привычном для пользователя смысле. Как разработчик, вы возможно будете читать об экране в другом контексте, когда это слово используется в отношении объекта *UIScreen*, который можно использовать для доступа к внешнему экрану.

< Назад к оглавлению

Адаптивность и layout

Встроенная адаптивность

Обычно люди предпочитают пользоваться своими любимыми приложениями на разных устройствах и при разной ориентации экрана. В iOS 8 и выше, используйте классификацию размеров и *Auto Layout*, чтобы оправдать ожидания пользователей и предусмотреть, как меняются макеты экрана, view controllers и представления данных при изменении среды дисплея. (**Display environment** относится к экрану всего устройства или к его части, например, часть экрана для всплывающего уведомления и первоначальное представление данных в контроллере разделено во view controller).

В iOS предусмотрено два класса размеров: обычный и компактный. **Regular** класс размеров ассоциируется со способным к расширению пространством, а **compact** размер ассоциируется с ограниченным пространством. Для того, чтобы охарактеризовать **display environment**, нужно выбрать между горизонтальным классом размеров и вертикальным классом размеров. Как вы уже догадались, устройство на iOS может использовать один класс размеров для *portrait* ориентации и другой класс размеров для *landscape* ориентации.

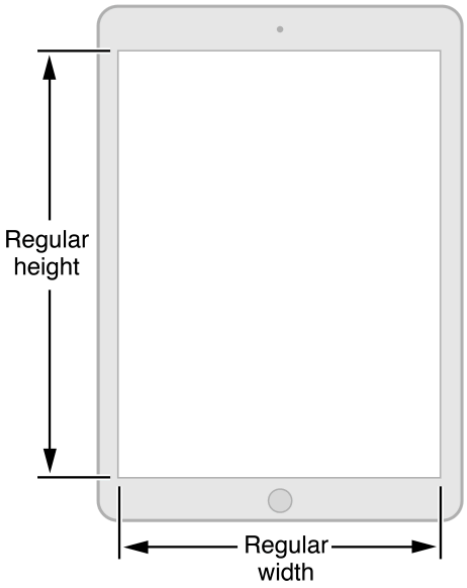
iOS автоматически делает изменения в layout, когда меняется класс размера в display environment. Например, когда вертикальный класс размеров меняется с compact на regular, панели навигации и панели управления автоматически становятся выше.

Если вы полагаетесь на классы размеров при изменении макета, ваше приложение будет отлично смотреться в любой display environment. Чтобы разобраться, как работать с классами размеров в программе Interface Builder, смотрите *Size Classes Design Help*.

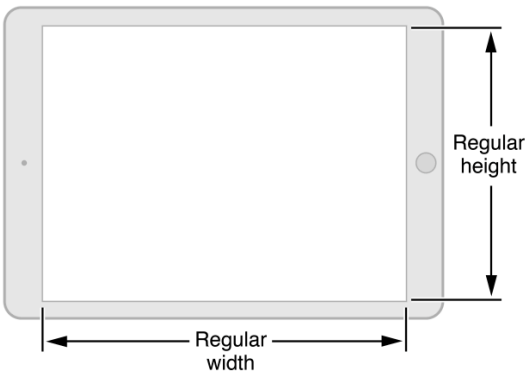
> Примечание:
> В рамках одного класса размеров, продолжайте использовать Automatic Layout для небольших изменений в layout, таких как расширение или сужение контента.

Следующие конкретные примеры помогут вам визуально представить, как классы размеров характеризуют display environment на различных устройствах. Например, на iPad используется обычный класс размеров во всех положениях и ориентациях. Другими словами, на iPad display environment всегда горизонтальный regular и вертикальный regular.

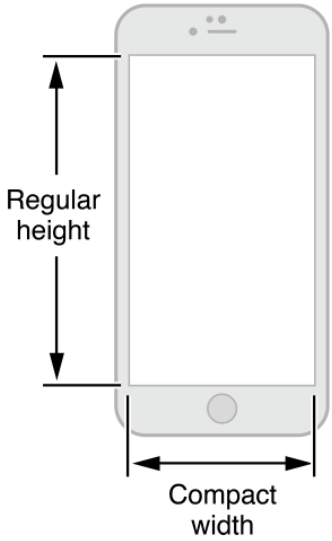
Классы размеров на iPad в portrait ориентации



Классы размеров на iPad в landscape ориентации

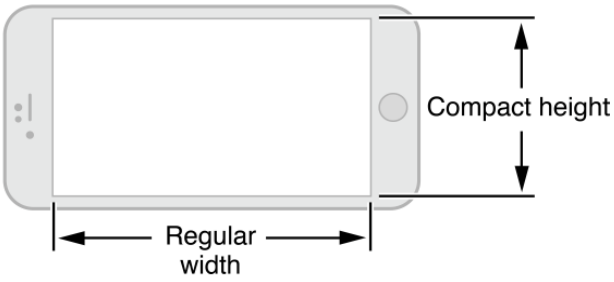


Display environment в iPhone зависит от устройства и ориентации дисплея.

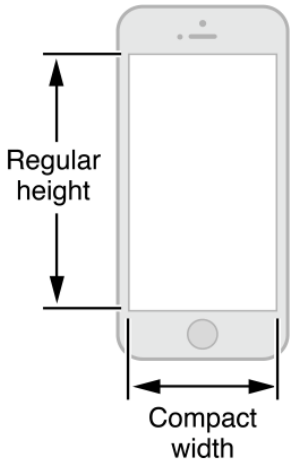


В portrait ориентации iPhone 6 Plus использует компактный горизонтальный и regular вертикальный класс размеров.

В landscape ориентации iPhone 6 Plus использует regular горизонтальный и compact вертикальный класс размеров.

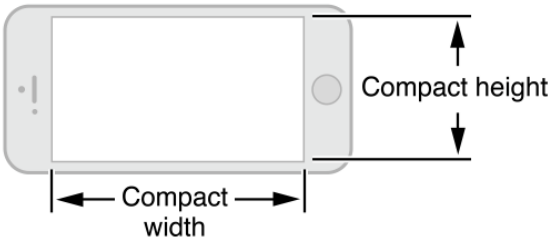


Все другие модели iPhone, включая iPhone 6, используют одинаковый набор классов размеров.



В portrait ориентации, iPhone 6, iPhone 5 и iPhone 4s используют compact горизонтальный и regular вертикальный класс размеров.

В landscape ориентации, эти устройства используют compact размеры и в вертикальном, и горизонтальном положении.



Устройте незабываемый опыт работы с приложением в любом environment

Когда вы хотите улучшить приложение благодаря адаптивности, вы должны удостовериться, что пользовательский интерфейс правильно реагирует на изменения в display environment. Следуйте этим правилам, чтобы пользователи наслаждались работой с вашим приложением независимо от ориентациии экрана.

В любом environment фокусируйтесь на основном контенте. Это ваш главный приоритет. Люди пользуются вашим приложением, чтобы просматривать и взаимодествовать с важным для них контентом. Изменение фокуса приложения при смене ориентации экрана может сбить пользователя с толку и дать ощущение, что он потерял контроль над приложением.

Избегайте беспричинного изменения layout. Похожий опыт работы с приложением в любой ориентации позволяет пользователям сохранять их привычки работы с приложением, когда они поворачивают устройство или открывают приложение на другом устройстве. Например, если вы показываете изображения в виде сетки в regular горизонтальном environment, не нужно показывать ту же самую информацию в виде списка в горизонтальной compact environment, достаточно лишь изменить размеры сетки.

Если приложение работает только в одном типе ориентации, проинформируйте об этом. Пользователи ожидают, что приложением можно будет пользоваться в любой ориентации экрана и будет лучше, если вы удовлетворите эти ожидания. Но если приложение работает только в одном типе ориентации, выполните следующие правила:

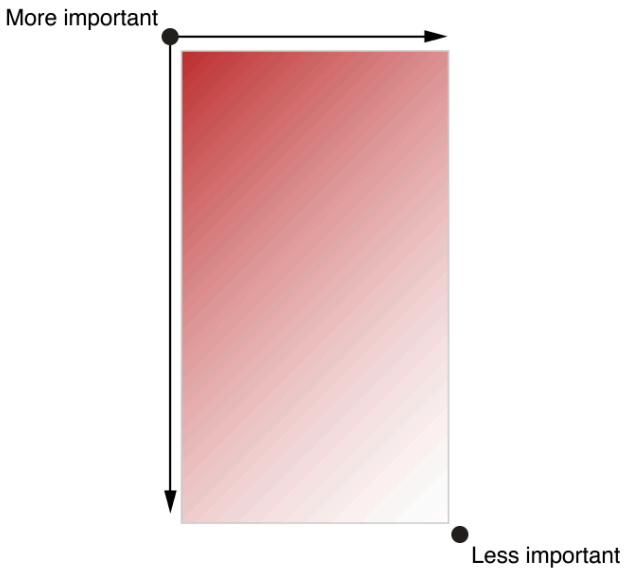
- **Старайтесь не показывать элементы пользовательского интерфейса, которые предлагают пользователям повернуть устройство**. Работа в поддерживаемой ориентации открыто предлагает людям повернуть экран, если возможно, без добавления лишних элементов к пользовательскому интерфейсу.
- **Поддерживайте оба варианта одного типа ориентации**. Например, если приложение работает только в landscape ориентации экрана, люди должны иметь возможность пользоваться им, если кнопка “Главное меню” находится и справа, и слева. И если люди поворачивают устройство на 180 градусов при запущенном приложении, будет лучше если приложение отреагирует на это поворотом контента на 180 градусов.

Если приложение воспринимает изменение в ориентации экрана как нацеленное действие пользователя, относитесь к поворотам в зависимости от настроек приложения. Например, если в игре поворот экрана используется для движения объектов в игре, экран не должен поворачиваться. В таком случае, приложение должно запускать в обоих вариантах необходимой ориентации и предлагать пользователю возможность выбрать между вариантами перед тем, как запустится основная функция приложения. Как только пользователи запустят основную функцию приложения, приложение начнет отвечать на движение устройства в запрограммированном формате.

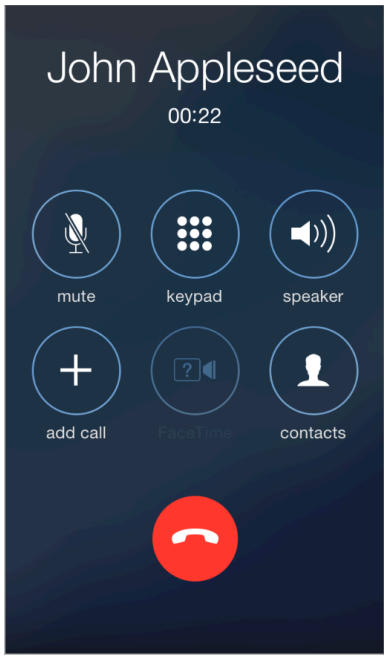
Используйте layout для передачи информации

Layout заключает в себе не только то, как элементы пользовательского интерфейса смотрятся на экране. Благодаря layout, вы показываете пользователю, что является самым важным, какие у него есть возможности и как элементы связаны между собой.

Позвольте пользователю легко концентрироваться на главном, размещая важный контент и функционал сверху. Один из способов сделать это - размещать самые важные элементы в верхней части экрана и - в странах, где читают слева на право - ближе к левой части экрана:



Используйте визуальный вес и баланс, чтобы продемонстрировать пользователю относительную важность различных элементов. Крупные значки привлекают внимание и выглядят более важными, чем маленькие. На крупные значки также легче нажать, что особенно полезно в приложениях, которыми люди пользуются на ходу - например, Телефон и Часы.



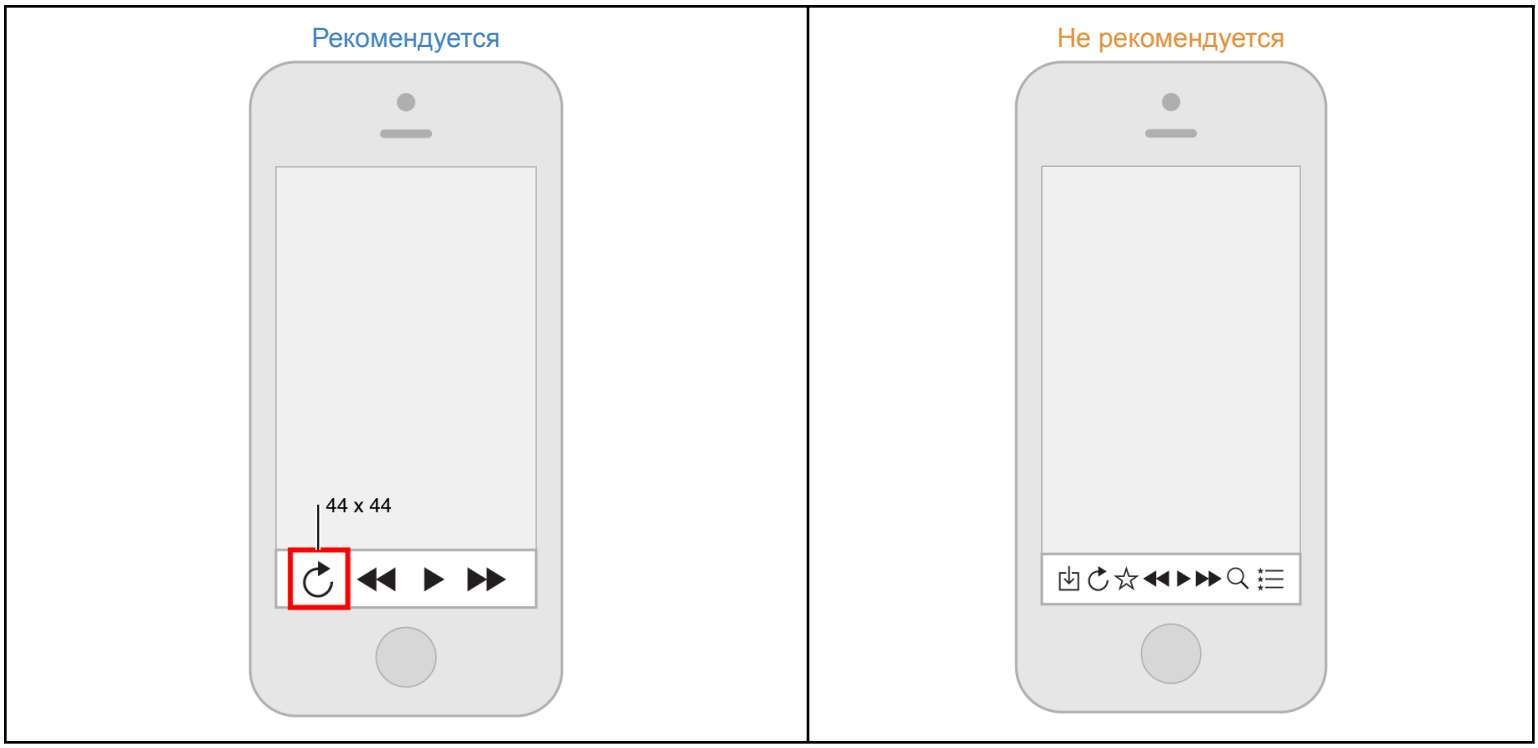
Используйте выравнивание, чтобы упростить беглый просмотр и дать представление о группировке или иерархии. Выравнивание делает приложение более аккуратным и правильно организованным. Оно выделяет места, на которых нужно сфокусироваться при беглом просмотре экрана, полного информации. Выравнивание и отступы между разными группами информации показывают, как эти группы взаимосвязаны между собой и упрощают поиск определенной информации.

Убедитесь в том, что пользователи могут понять основную часть контента в его первоначальном размере. Например, пользователь не должен двигать изображение, чтобы дочитать важный текст или увеличивать изображение, чтобы толком рассмотреть его.

Будьте готовы к изменению размера текста. Выбирая другой размер текста в Настройках, пользователи ожидают, что и приложение отреагирует на это должным образом. Чтобы предусмотреть изменения в размере текста, нужно подстроить макет под возможные изменения; для более подробной информации о тексте в приложении смотрите статью *Text Should Always Be Legible*.

Как можно чаще старайтесь исключить несогласованность между элементами пользовательского интерфейса. В целом, элементы, отвечающие за схожие функции, должны выглядеть похоже. Люди чаще всего предполагают, что разница и несогласованность между элементами в приложении неспроста и тратят время на то, что понять, почему это так.

Сделайте работу с контентом и контрольными элементами проще, предоставляя достаточно пространства каждому элементу. Целевой размер для контрольных элементов, на которые нужно нажимать, составляет 44 x 44 points.



< Назад к оглавлению

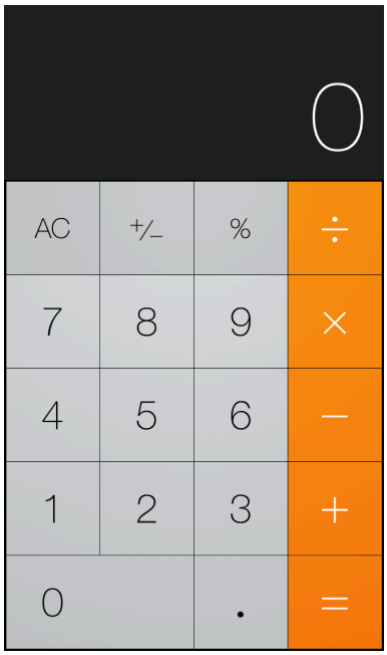
Запуск и выход из приложения

Начинайте без промедлений

Часто говорят, что люди тратят не более 1-2 минут, чтобы оценить новое приложение. Постарайтесь воспользоваться этим коротким промежутком времени по максимуму, предоставляя пользователю полезный контент незамедлительно. Таким образом вы заинтересуете новых пользователей и предоставите лучший опыт работы с приложением.

> **Важно:** Не предлагайте пользователям перезагрузить устройство после установки приложения. Перезагрузка занимает время и придает ощущение, что ваше приложение ненадежное и сложное в использовании.
> Если ваше приложение требует много оперативной памяти телефона и без перезагрузки устройства будет плохо работать, нужно решить эту проблему. Возможные пути решения и способы создания эффективных приложений описаны в разделе “Use Memory Efficiently” в ”Guidance on developing a well-tuned app”.

Как можно чаще старайтесь избегать экранов приветствия и других заставок во время запуска. Будет лучше, если пользователь сможет начать работу с приложением незамедлительно.

Рекомендуется	Не рекомендуется
	

Избегайте необходимости в необходимости задавать настройки для приложения. Вместо этого вы можете:

- **Сфокусироваться на нуждах 80% ваших пользователей.** Если вы сделаете это, то большинству пользователей и не придется задавать никаких настроек, потому что приложение будет уже оптимизировано под их нужды. Если существуют функции, которые будут интересны только некоторым пользователям или которыми они воспользуются только пару раз, не стоит включать их в приложение.
- **Постарайтесь собрать как можно больше информации из сторонних источников.** По возможности старайтесь использовать информацию, уже предоставленную пользователями во встроенных приложениях или настройках устройства, поищите эти данные в системе. Не просите пользователей снова вводить эти данные.
- **Если нужно ввести информацию для запуска приложения, делайте это внутри приложения.** Затем, сохраните эту информацию как можно скорее (например, в настройках приложения). Таким образом людям не придется переходить в меню Настроек для того, чтобы воспользоваться вашим приложением. Если им будет нужно изменить внесенную информацию, они в любое время смогут сделать это прямо в настройках приложения.

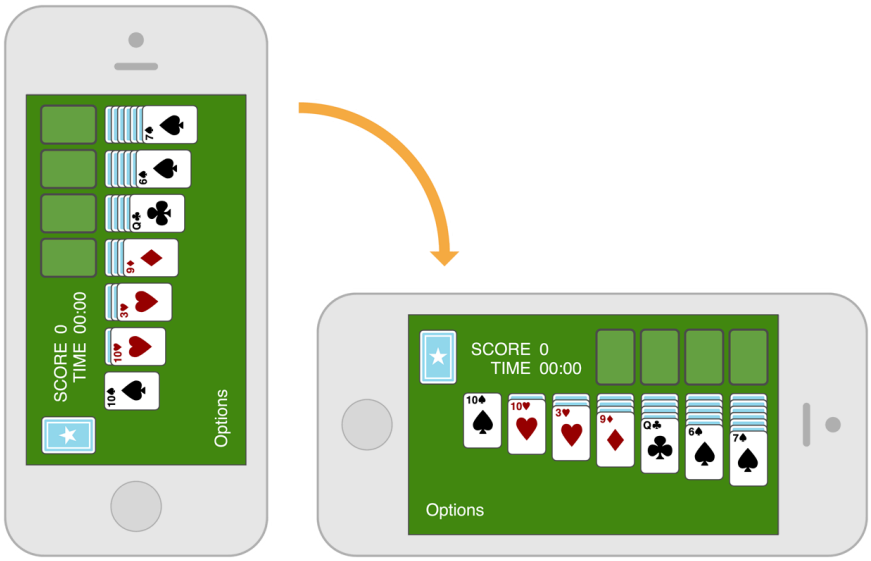
Откладывайте необходимость в регистрации или логине как можно дольше. Будет лучше, если люди смогут просмотреть большую часть приложения и часть его функционала без необходимости в регистрации или логине. Например, AppStore не требует логина до тех пор, пока пользователь не соберется что-либо купить. Пользователи часто решают прекратить пользоваться приложением, которое требует регистрации или логина до того, как они получили от приложения какую-либо полезную информацию.
Если пользователь обязательно должен залогиниться, коротко опишите в окошке логина, в чем заключается эта необходимость и какую пользу это принесет пользователю.

Тщательно продумайте период знакомства с приложением. (Знакомство с приложением заключается в демонстрации основных функций приложения перед началом работы с объяснением, как они работают). Перед тем как решить, стоит ли включать знакомство с приложением в разработку, постарайтесь разработать приложение таким образом, что все его функции интуитивны понятны пользователю и их легко найти в приложении. *Этап знакомства с приложением не заменяет собой необходимость качественного дизайна приложения*. Если вы считаете, что этап знакомства все-таки нужен, следуйте следующим советам:

- **Предоставляйте только информацию, необходимую для начала работы**. Хороший период знакомства с приложением показывает пользователю только то, что нужно сделать в первую очередь или кратко демонстрирует работу функций, в которых заинтересовано большинство пользователей. Если на этом этапе вы предоставите слишком много информации до того, как пользователь сам попробует работу с приложением, вы возлагаете на него ответственность запомнить все рассказанные вами детали, которые ему все равно сначала не понадобятся. Это может создать ощущение, что приложением сложно пользоваться. Если дополнительные инструкции необходимы для определенных задач, демонстрируйте их только когда пользователь начинает выполнять эти задачи.
- **Используйте анимацию и интерактивные элементы, чтобы вовлечь пользователя и помочь ему учиться на ходу.** Добавляйте текстовые подсказки редко и только если они действительно полезны; не думайте, что пользователь будет тратить время на чтение длинных и сложных текстов. Например, не делайте описание функции в виде последовательности действий, если гораздо легче показать принцип работы при помощи анимации. Чтобы помочь пользователю со сложными функциями, попробуйте добавить временные overlay views, которые вкратце будут описывать следующее действие, перед тем как пользователь его должен сделать. Избегайте использования скриншотов на этапе знакомства с приложением, так как они не интерактивны и пользователи могут спутать их с пользовательским интерфейсом приложения.
- **Сделайте пропуск или отмену этапа знакомства с приложением простыми и быстрыми.** После того, как пользователи прошли этап знакомства с приложением, они вряд ли заходят проходить его снова. Некоторые пользователи не захотят проходить его вообще. Убедитесь, что приложение «запомнило» выбор пользователя касательно ознакомительного периода и не заставляйте проходить его каждый раз, когда пользователь открывает приложение.

Не просите пользователей оценить ваше приложение слишком быстро. Просьба оценить приложение слишком скоро после его установки обычно раздражает пользователей и это снизит количество полезного фидбека, который вы можете получить от пользователей. Чтобы фидбек был рациональным и полезным, дайте пользователям шанс сформировать мнение о приложении перед тем как вы попросите их поделиться этим мнением. Например, подождите пока пользователь увидит определенное количество экранов или выполнит определенный набор действий.

Запускайте приложение в текущей ориентации экрана устройства. В случае, если приложение поддерживается только в одной ориентации, всегда запускайте его в этой ориентации и позвольте пользователю повернуть устройство, если это необходимо. Например, если игра и просмотр медиафайлов работает только в landscape ориентации, будет правильнее запустить приложение сразу же в landscape ориентации, даже если само устройство в данных момент находится в portrait ориентации. Таким образом, если люди запустят приложение, когда устройство находится в portrait ориентации, они буду знать, что его следует повернуть в landscape ориентацию, чтобы хорошо видеть контент.



Примечание.
Будет лучше, если приложение, работающее только в landscape ориентации, будет поддерживаться в обоих видах landscape ориентации – то есть при кнопке Home справа и слева от пользователя. Если устройство уже находится в landscape ориентации, приложение должно запуститься в таком же типе landscape ориентации (если то возможно). В противном случае, запускайте приложение всегда в landscape ориентации с положением кнопки Home справа. (Для более подробной информации о поддержке различных типах ориентации экрана, смотрите раздел “Respond to Changes in Device Orientation”).

Используйте лого или изображение при запуске приложения. iOS показывает изображение при запуске вашего приложения – это дает пользователю ощущение, что приложение работает быстро и дает время приложению, чтобы загрузиться. Научитесь создавать файлы и изображения для запуска приложения в разделе Launch Images.

Если возможно, освободите пользователя от необходимости читать дисклеймер или принимать лицензионные соглашения использования до того, как им удалось испробовать приложение. Вместо этого вы можете разрешить AppStore показывать ваш дисклеймер или лицензионное соглашение, чтобы пользователь могли испробовать приложение до того, как скачают ваше приложение. Если без дикслеймера в приложении не обойтись, убедитесь, что он гармонично смотрится в пользовательском интерфейсе и представляет собой баланс между бизнес-необходимостью и впечатлениями пользователя о приложении.

Когда приложение перезапускается, приведите его в то же состояние, в котором пользователь закончил с ним работу. Люди не должны повторять уже пройденные шаги, чтобы оказаться в том же месте приложения. Подробнее о сохранении статуса и восстановлении приложения, смотрите раздел “State Preservation and Restoration”.

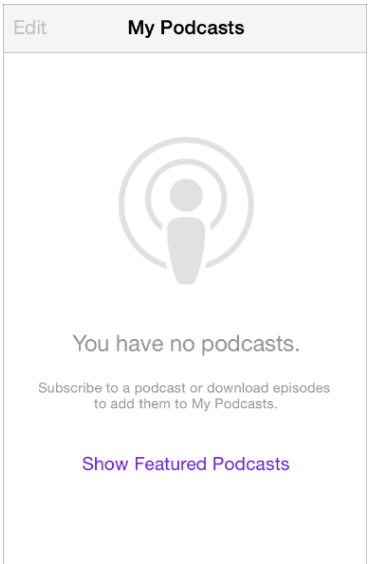
Всегда будьте готовы остановить работу приложения

В приложения на iOS никогда не бывает кнопок «Закрыть» или «Выйти». Люди заканчивают работу с приложением, когда они начинают работать с другим приложением, нажимают на кнопку Home или переключают устройство в спящий режим.
Когда люди переключаются с вашего приложения на другое приложение, мультизадачность на iOS переключает его в фоновый режим и заменяет его пользовательский интерфейс на пользовательский интерфейс другого приложения. Чтобы ваше приложение было готово к такой ситуации, нужно сделать следующее:

- **Сохраняйте данные пользователей как можно скорее и часто (в пределах разумного).** Это нужно делать, потому что приложение, работающее в фоновом режиме, может быть принудительно выключено в любой момент.
- **Сохраняйте текущий статус приложения во время остановки работы приложения до мельчайших деталей.** Таким образом люди не будут терять контекст происходящего, когда вернутся к работе с приложением. Например, если приложение предоставляет просмотр данных со скроллингом, сохраняйте место, до которого дошел пользователь. Вы можете подробнее узнать об эффективных способах сохранения статуса и восстановления приложения, смотрите “State Preservation and Restoration”.

Некоторые приложения должны иметь возможность работать в фоновом режиме, пока пользователь работает с другим приложением. Например, пользователь хочет продолжать слушать музыку в одном приложении, в то время как проверяет свой список дел на день или играет в игру в другом приложении. Узнайте подробнее о том, как правильно и красиво подготовить приложение к мультизадачности в разделе Multitasking.

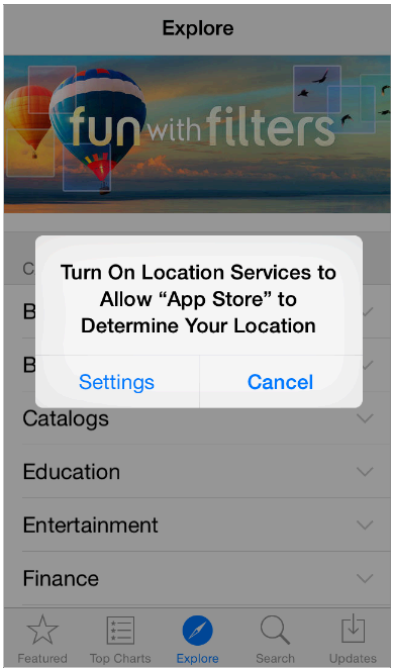
Никогда не завершайте приложение на iOS принудительно. Люди часто воспринимают это как ошибку системы. Если что-то не дает вашему приложению работать, как положено, вы должны рассказать пользователю о ситуации и объяснить, что они могут сделать, чтобы исправить это. Вот несколько хороших способов сделать это:



Если все функции приложения недоступны, отобразите экран, на котором будет объяснение ситуации и предлагаемое решение проблемы. Эта информация дает фидбек пользователю и убеждает их в том, что с вашим приложением все в порядке. Это также дает пользователю

контроль над ситуацией, давая им возможность выбрать между решением проблемы, использованием приложения в неполном формате и переключением на другое приложение.

Если недоступны только некоторые функции приложения, покажите экран или уведомление, когда люди пытаются воспользоваться недоступной функцией. В противном случае, люди должны иметь возможность работать с остальными функциями приложения. Если вы хотите использовать уведомление, убедитесь, что оно появляется *только* когда пользователь пытается воспользоваться недоступной функцией.



< Назад к оглавлению

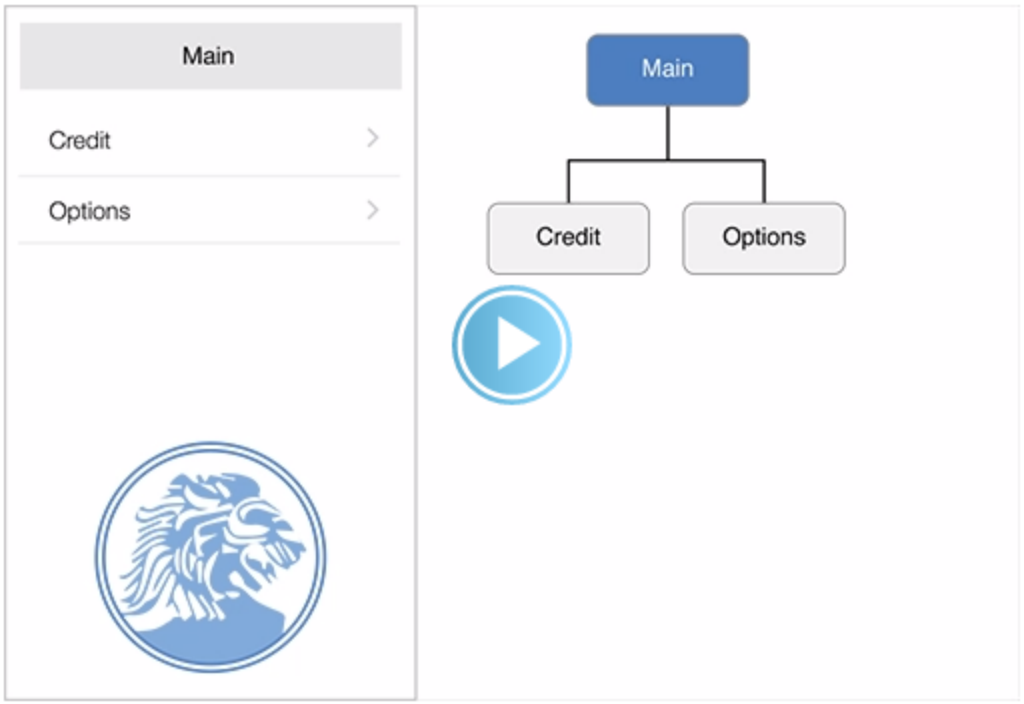
Навигация

Люди не сильно обращают внимание на опыт навигации по приложению до тех пор, пока он их устраивает. Ваша работа заключается в том, чтобы создать навигацию по приложению таким образом, чтобы она поддерживала структуру и основную цель приложения, но не привлекала к себе лишнего внимания.

Обобщая, есть три основных типа навигации, каждый из которых хорошо подходит для приложений с определенной структурой:

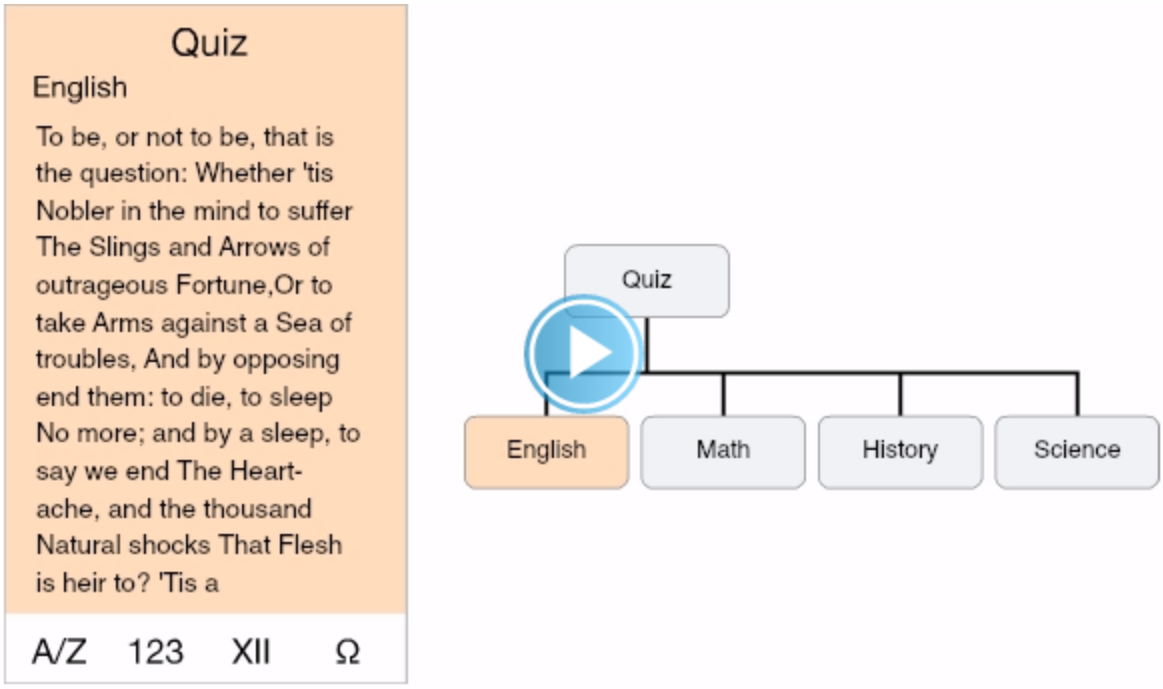
- Hierarchical структура
- Flat структура
- Приложение, построенное на основе контента и опыта использования

В hierarchical приложении, пользователь перемещается по приложению, делая по одному выбору на каждом экране до тех пор, пока не доберется до места назначения. Чтобы добраться до другого места назначения в приложении, пользователь должен отменить несколько предыдущих выборов – или начать сначала – и выбрать другие функции. Настройки и Почта – хорошие примеры приложений, использующих иерархичную структуру.



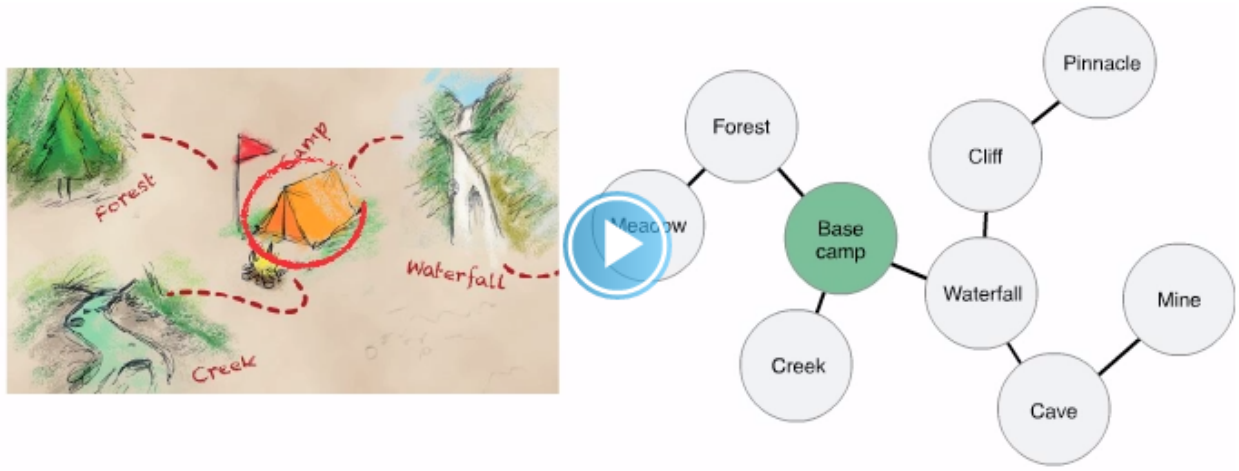
Link to this video

В приложении с flat структурой, пользователь может перейти от одной широкой категории к другой, потому что все основные категории доступны с главного меню. Музыка и AppStore – хорошие примеры приложений, использующих такую структуру.



Link to this video

Неудивительно, что приложения, структура которых основана на информации о контенте или опыте использования, используют навигацию, также основанную на контенте или опыте использования. Например, пользователь перемещается по книге переходя со страницы на страницу или выбирая определенную страницу в содержании; в играх навигация также играет немаловажную роль.



Link to this video

В некоторых случаях совмещение нескольких типов навигации в одном приложении делает его удобнее. Например, в приложении с плоской структурой для основных категорий, внутри одной из категорий функции могут располагаться в виде списка (то есть, иерархически).

Пользователь всегда должен знать, в какой части приложения он находится и как перейти к нужной ему части приложения. Независимо от выбранного вами стиля навигации, подходящего структуре приложения, самая главная задача – сделать путь пользователя по контенту приложения логичным, предсказуемым и простым для понимания.

UIKit определяет некоторые стандарты пользовательского интерфейса, которые делают внедрение иерархичной и прямой навигации простым. Кроме того, есть некоторые элементы, которые помогают создать зависимую от контента навигацию, например, навигацию в стиле книги и просмотра медиафайлов. Игры и другие приложения чаще всего используют навигацию, основанную на опыте использования и для этого нужны кастомизированные элементы пользовательского интерфейса.

Используйте Navigation bar, чтобы пользователи могли легко отследить иерархию данных в приложении. Заголовок Navigation bar's может указывать на то, где пользователь находится сейчас; кнопка «назад» позволяет легко вернутся на прежний уровень навигации. Подробнее смотрите в разделе Navigation bar.

Используйте Tab bar, чтобы показать несколько похожих и равноценных категорий контента или функционала. Tab bar хорошо подходит для приложений с flat структурой, ее постоянное положение позволяет пользователю переключаться между категориями независимо, где они находятся прямо сейчас. Подробнее смотрите в разделе Tab bar.

Используйте Page Control, когда каждый экран приложения отображает индивидуальное отображение одного и того же типа содержимого или страницы. Page Control – отличный инструмент, чтобы показать пользователю какие страницы или объекты доступны и какой из них отображается в данный момент. Например, приложение Погода использует page control, чтобы показать, сколько прогнозов погоды в различных местоположениях открыл пользователь. Подробнее смотрите в разделе Page Control.

Чаще всего, наилучшим решением является один путь к одному конкретному экрану. Если один из экранов пользователь хочет видеть в разных контекстах, используйте temporary views, такие как modal view, action sheet или уведомления. Подробнее смотрите в разделах Modal view, Action sheet и Alert.

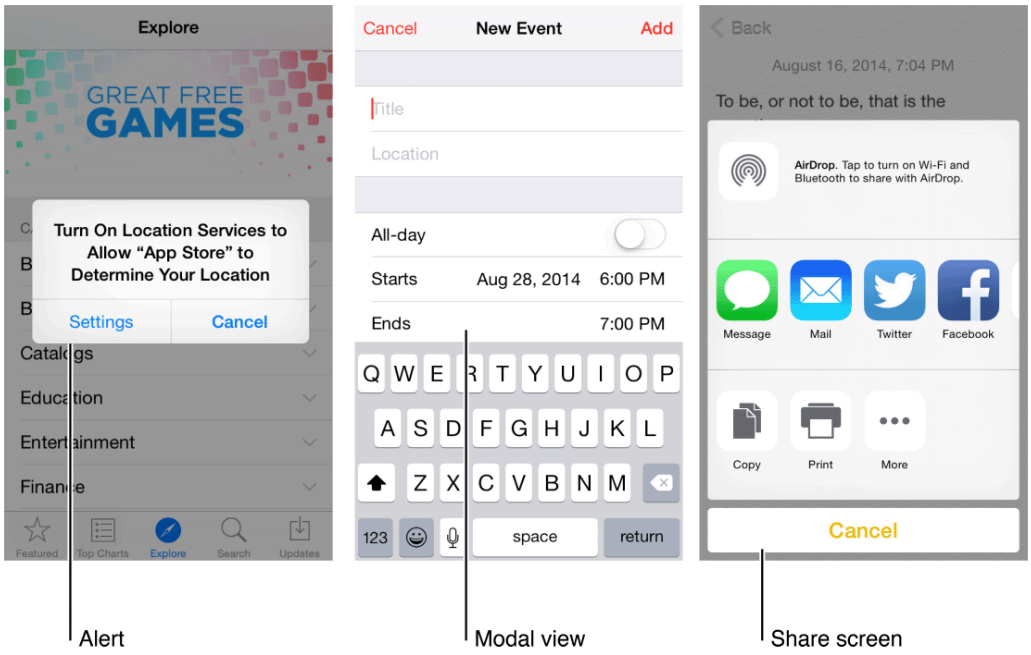
UIKit также предоставляет следующие похожие контроллеры:

- Segmented Control – позволяет пользователям видеть различные категории или аспекты контента на экране, но не дает возможности переходить к ним.
- Toolbar – выглядит похоже с tab bar и navigation bar, но не дает возможность навигации. Вместо этого дает пользователю возможность контролировать контент на текущем экране.

< Назад к оглавлению

Модальный контекст

Модальность - то есть режим или способ, благодаря которому что-то работает или существует - имеет свои плюсы и минусы. Она дает пользователям возможность завершить действие или получить информацию без отвлекающих факторов , но в то же время она не дает им в этот момент взаимодействовать с остальной частью приложения.



При идеальном раскладе, люди должны пользоваться приложениями на iOS в нелинейном формате, поэтому лучше минимизировать количество modular experiences в вашем приложении. Используйте модельный контекст только в следующих ситуациях:

- Необходимо срочно привлечь внимание пользователя
- Самостоятельная, автономная задача должна быть выполнена или отменена, чтобы данные пользователя не пребывали в неоднозначном состоянии

Модальные задачи должны быть простыми, короткими и чётко сфорусированными на проблеме. Пользователь не должен воспринимать modal view как мини-приложение внутри вашего приложения. Если под-задача слишком сложная , люди могут потерять нить действий в главной задаче, которая приостановилась при открытии модального контекста. Будьте особенно осторожны при создании модальных задач, которые включают в себя иерархию views, потому что пользователь может потеряться в приложении и забыть, как вернуться в предыдущее местоположение в приложении. Если модальная задача включает в себя под-задачи в отдельных views, дайте пользователю один понятный путь по этой иерархии и избегайте цикличности. Подробнее о modal views смотрите раздел Modal view.

Всегда делайте выход из модальной задачи простым и понятным. Люди всегда должны понимать, что произойдет, если они закроют окно с модальной задачей.

Если задача предполагает наличие иерархии из modal views, убедитесь, что пользователи могут понять, что случится, если они нажмут на кнопку Done на одном из не главных views в иерархии. Проанализируйте задачу и решите, должна ли кнопка Done отвечать за часть задачи или за всю задачу. Из-за возможной неразберихи, избегайте добавления кнопки Done в окно подзадач.

Используйте предупреждения (alerts) только для самых главных, и приводящих в действию, оповещений. Alert прерывает использование приложения и требует прикосновения к экрану, чтобы его отключить, поэтому важно сделать так, чтобы alert-ы не казались назойливыми и оставались полезными. Подробнее смотрите в разделе Alert.

Уважайте предпочтения пользователей касательно получения оповещений. В Настройках пользователи указывают, как часто они хотят получать оповещения от вашего приложения. Подчиняйтесь этим предпочтениям, чтобы пользователям не захотелось отключить ваши уведомления насовсем.

< Назад к оглавлению

Интерактивность и Фидбек

Пользователи уже знают стандартные жесты

Люди используют жесты – такие как tap, drag, pinch - для управления и взаимодействия с устройством на iOS. Использование жестов создает близкую персональную связь с их устройствами и усиливает ощущение контроля над объектами на экране. Люди чаще всего ожидают, что привычные жесты будут исполнять те же функции во всех приложениях, что они используют.

Прикосновение (Tap) – Нажать или выбрать контроллер или объект.

Перетащить (Drag) – Прокрутить или повернуть что-то с одной стороны на другую. Перетащить элемент.

Перелистывание (Flick) – Прокрутить или повернуть с одной стороны на другую быстро.

Смахивание (Swipe) – Одним пальцем: вернуться к предыдущему экрану, открыть скрытый просмотр в split view controller или кнопка. Удалить в строках при просмотре таблицы. Четырьмя пальцами: переключиться между приложениями на iPad.

Двойное прикосновение (Double tap) – Увеличить и поместить в центр экрана часть контента или фотографии. Уменьшить (если уже увеличено).

Сведение и разведение пальцев (Pinch) – Развести чтобы увеличить, свести чтобы уменьшить.

Прикоснуться и удержать (Touch and hold) – в редактируемом или выделяемом тексте – для открытия лупы с целью установки курсора

Встряхнуть устройство (Shake) – Чтобы отменить или повторить последнее действие

Кроме стандартных и знакомых пользователю жестов, в iOS запрограммированы жесты, одинаковые во всей оперативной системе, например, открывающие Control Center или Notification Center. Пользователи считают эти жесты универсальными и считают, что они сработают независимо от того, каким приложением они пользуются.

Избегайте связи различных действий в приложении со стандартными жестами. Если только ваше приложение не игра, изменение значения стандартного жеста запутает пользователя и сделает приложение более сложным в использовании.

Не придумывайте новые жесты для действий, для которых уже предусмотрены стандартные жесты. Люди привыкли к тому, как работают стандартные жесты, и они вряд ли будут в восторге от необходимости запоминать новые способы для совершения тех же самых действий.

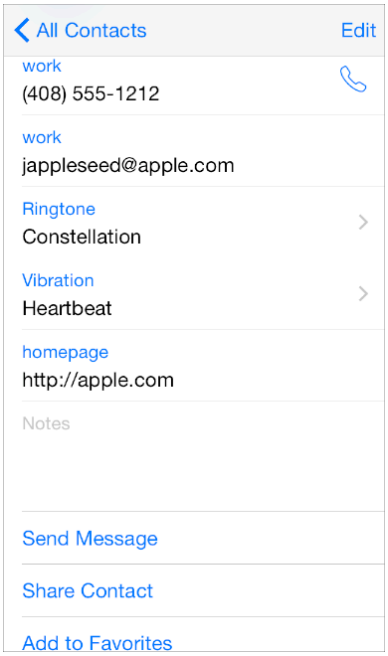
Используйте сложные жесты как быструю альтернативу для совершения действия, но не единственный способ совершения этого действия. Всегда давайте пользователям простой и понятный способ выполнить действия, даже если это значит 2-3 дополнительных прикосновения к экрану. Простые жесты позволяют пользователю сконцентрироваться на опыте использования приложения и контенте, а не на взаимодействии с ним.

Избегайте новых жестов, если только ваше приложение не игра. В играх и других затягивающих приложениях кастомизированные жесты могут стать веселой частью опыта использования приложения. Но в приложениях, которые делают жизнь пользователей проще и помогают в важных делах, лучше использовать стандартные жесты, так как люди не хотят тратить время и усилия на запоминание новых жестов.

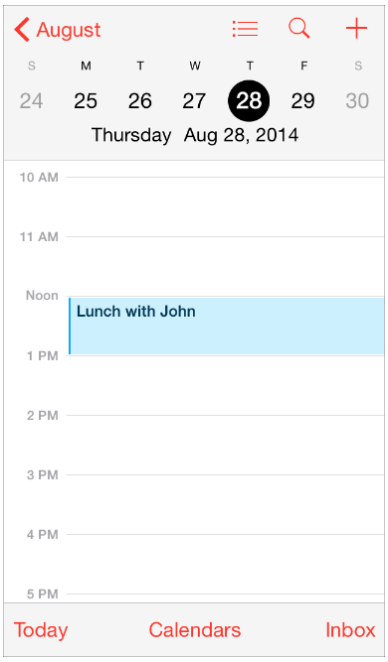
В regular environment, попробуйте жесты с использованием нескольких пальцев руки. Хотя сложные жесты не подходят для каждого приложения, они могут улучшить опыт работы с приложениями, в которых люди проводят много времени (игры, создание контента). Но помните, что сложные жесты не всегда интуитивно понятны, поэтому они не могут быть единственным способом совершения действия.

К интерактивным элементам хочется прикоснуться

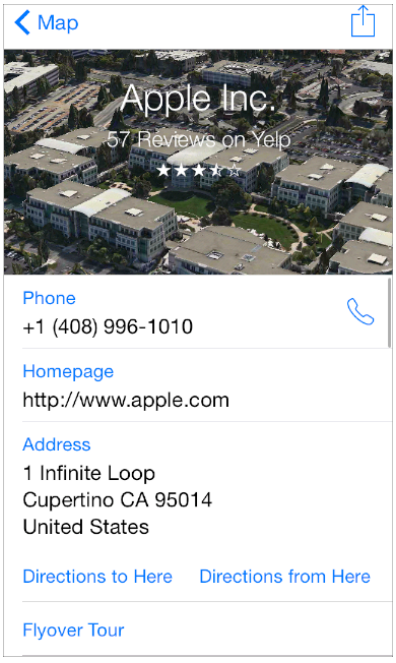
Чтобы указать на интерактивность элемента, встроенные приложения использует много указателей, таких как цвет, местоположение, контекст и «говорящие» иконки и лейблы. Пользователи редко нуждаются в дополнительных указаниях на интерактивность элемента на экране или объяснениях, за что этот элемент отвечает.



Основной цвет дает пользователю четкий сигнал интерактивности, особенно в приложениях, которые не используют много разных цветов. В Контактах голубой цвет указывает на интерактивность элементов и придает приложению упорядоченную и узнаваемую визуальную тему.

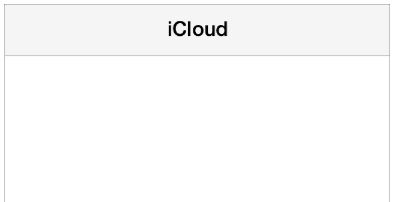


Кнопка «Назад» использует несколько указателей на интерактивность и объяснений ее функционала. Она появляется в ответ на навигацию, на ней изображена стрелка в обратном направлении, она чаще всего окрашена в основной цвет и ее заголовок часто указывает на название предыдущего экрана.



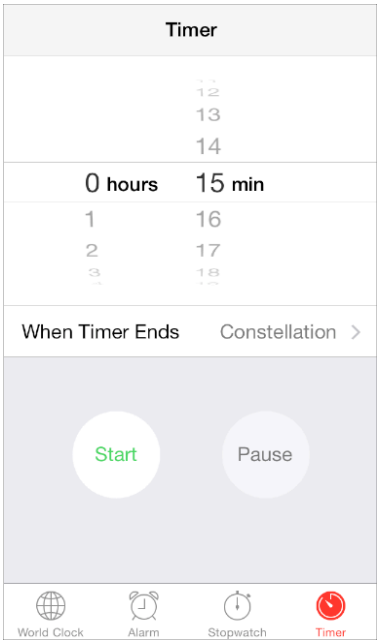
Иконка или название, которые представляют собой четкий призыв к действию открыто приглашает пользователей нажать на нее. Например, заголовки в Картах, такие как “Вид с высоты птичьего полета” или “Как добраться сюда” демонстрируют собой соответствующие действия. Вместе с выделение ключевым цветом, призывающие к действию заголовки отменяют необходимость в ярко выраженных границах кнопок и прочих графических украшений.

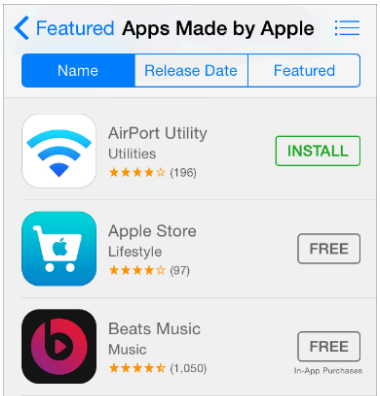
В зоне контента, добавляйте границы кнопок или фон только по необходимости. Кнопки в панелях, action sheets и предупреждениях не нуждаются в границах поскольку пользователи и так знают, что элементы в этих зонах экрана интерактивны. В зоне контента, с другой стороны, кнопке может понадобится граница или фон, чтобы отличить ее от остального контента. Например, приложения Музыка, Часы и AppStore используют такие кнопки в некоторых контекстах.



“Фотографии” используют границу кнопки, чтобы отличить кнопку “Start Sharing” от поясняющего текста над ней.

“Часы” использую фон кнопок на экранах “Секундомер” и “Таймер”, чтобы привлечь внимание ко кнопкам “Старт” и “Пауза” и сделать нажатие на них простым и легким даже если пользователь находится в помещении, полном отвлекающих моментов.





AppStore использует границу кнопок в строках таблицы, чтобы подчеркнуть разницу между прикосновением к строке таблицы для получения дополнительной информации и прикосновением к кнопке для начала покупки или установки.

Фидбэк способствует пониманию

Фидбэк помогает пользователям знать, что приложение делает сейчас, узнать, что они могут сделать позже и понять результаты их действий. UIKit controls и views позволяют давать различный фидбэк.

Как можно чаще интегрируйте статус и другую уместную информацию-фидбек в ваш пользовательский интерфейс. Будет лучше, если пользователь будет получать эту информацию без совершения каких-либо действий и без отвлечения от работы с приложением. Например, Почта показывает текущий статус почтового ящика в toolbar и не закрывает собой контент, важный для пользователя.

Избегайте ненужных предупреждений. Предупреждение - сильный инструмент фидбэка , но он должен использоваться только для донесения важной информации, в идеале с возможностью сразу же совершить действие. Если использовать слишком много предупреждений, которые не содержат важной информации, со временем пользователи учатся их не игнорировать. Подробнее смотрите в разделе “Предупреждения.”

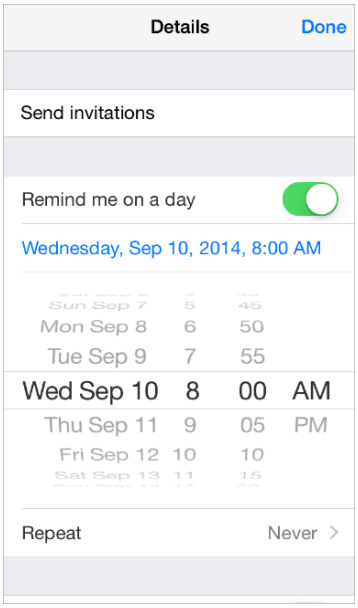
Ввод информации должен быть простым

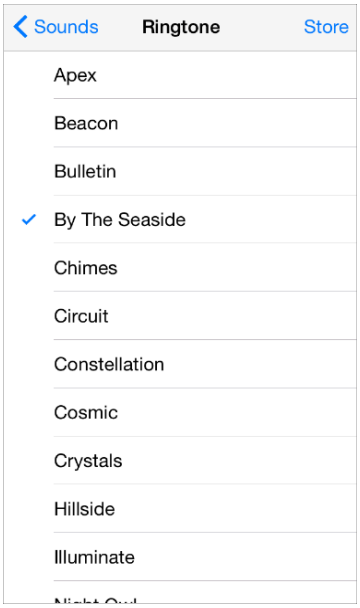
Ввод информации требует времени и внимания, независимо от того используют ли пользователи tap controls или клавиатуру. Когда приложение работает медленно из-за того, что пользователи должны ввести много информации перед тем, как приложение сделает что-то полезное, люди теряют охоту работать с этим приложением.

Сделайте выбор простым для пользователя. Например, вы можете использовать инструмент выбора или просмотр в виде таблицы вместо сплошного текста, потому что большинство людей предпочитают возможность выбрать один из вариантов, а не введение текста вручную.

Date picker в Напоминаниях

Список опций в Settings





Получайте информацию напрямую из iOS, когда это целесообразно. Люди хранят много информации о себе на их устройствах. Когда в этом есть смысл, не заставляйте пользователей предоставлять вам ту информацию, которую вы и сами можете найти в устройстве. Примером такой информации могут быть контакты или информация из календаря.

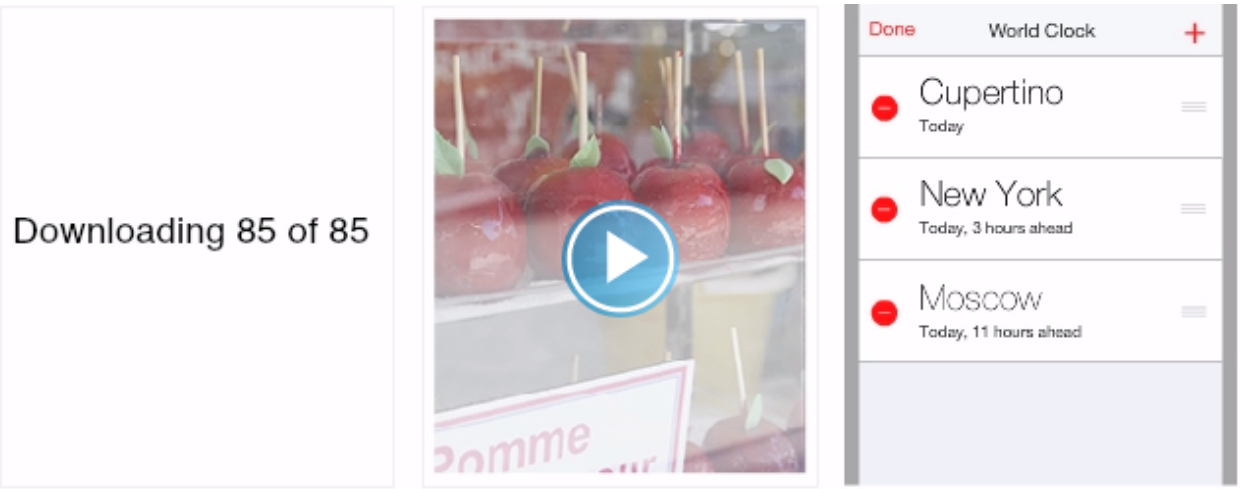
Создайте баланс между необходимостью вводить информацию и получением полезной информации от приложения. Ощущение, что они не только дают информацию, но и получают что-то взамен дает пользователям ощущение, что они движутся вперед, пользуясь вашим приложением.

< Назад к оглавлению

Анимация

Красивая, ненавязчивая анимация пронизывает весь пользовательский интерфейс iOS и делает работу с приложениями более захватывающей и динамичной. Подходящая анимация может:

- Сообщать статус и давать фидбэк.
- Усиливать чувство прямого контроля над устройством
- Помогать людям визуально представлять результаты их действий



Link to this video

Добавляйте анимацию осторожно, особенно если приложение не предполагает погружения в виртуальную среду. Анимация, которая кажется избыточной или неуместной может затруднять работу приложения, уменьшать его производительность и отвлекает пользователя от работы.

В частности, используйте эффект движения и UIKit dynamic behaviors только с четкой целью и ограничениями, а также не забывайте протестировать результат. Когда эти эффекты использованы к месту, они могут улучшить понимание и восторг пользователя от работы с приложением. Слишком частое их использование наоборот сделает приложение запутанным и неподдающимся контролю пользователя.

Если это уместно, делайте собственную анимацию похожей на встроенную анимацию. Люди привыкли к ненавязчивой анимации, используемой во встроенных приложениях на iOS. На самом деле, люди считают не резкие переходы между views, плавный переход в связи с изменением ориентации экрана и основанный на законах физики плавный скроллинг, как что-то само собой разумеющееся в приложениях на iOS. Если только вы не разрабатываете приложение, погружающее пользователя в виртуальную реальность - например игру - разработанная вами анимация должна быть похожа на встроенную анимацию.

Используйте анимацию в приложении последовательно. Как и с другими инструментами кастомизации приложения, важно использовать анимацию последовательно, чтобы пользователи могли сформировать полное мнение об опыте работы с вашим приложением.

В целом, стремитесь к реализму и достоверности в созданной вами анимации. Людям нравится творческий подход к внешнему виду приложения, но они могут чувствовать себя потеряно, когда движения не имеют смысла или противоречат законам физики. Например, если view открывается с помощью его переноса с верхней части экрана в нижнюю, то закрываться он должен поднятием его обратно вверх.Таким образом пользователь запоминает, откуда появляется view. Если вы будете закрывать view опуская его в самую нижнюю часть экрана, вы сломаете парадигму пользователя - какой же view доступен в верхней части экрана?

< Назад к оглавлению

Брендирование

Успешный брендинг включает в себя не только добавление элементов вашего бренда в приложение. Лучшие приложения объединяют в себе существующие элементы с уникальными ощущениями и впечатлениями от использования и именно это дает пользователю запоминающийся и восхитительный опыт работы с приложением.

iOS позволяет использовать созданные вами иконки, цвета и шрифты, которые помогут вам создать отличительный пользовательский интерфейс, не похожие на никакие другие приложения. Во время дизайна этих элементов, помните о двух вещах:

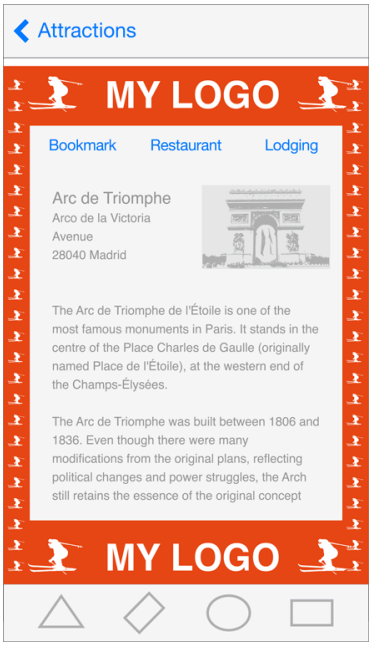
- Каждый отдельно созданный элемент должен хорошо выглядеть и работать сам по себе, но он так же должен сочетаться с остальными элементами приложения, независимо от того, стандартные они или разработанные вами.
- Чтобы хорошо смотреться на iOS, приложение не должно выглядеть как встроенные приложения, но оно должно включать в себя принципы почтительного отношения к контенту, ясности и глубины (узнайте больше об этих принципах в разделе Designing for iOS).Потратьте время на то, чтобы понять, как работают эти принципы и воплотите их в разработанных вами элементах приложения.

Если с помощью приложения вы хотите напомнить пользователю об уже существующем бренде, следуйте этим советам.

Включайте элементы бренда в интерфейс в изящном и не создающем для контента препятствий формате. Люди пользуются приложениями, чтобы решать задачи или развлекаться. И при этом они не хотят, чтобы их заставляли смотреть рекламу. Для наилучшего опыта работы с приложением, следует незаметно напоминать пользователю о сущности бренда благодаря шрифтам, цветам или фото.

Рекомендуется

Не рекомендуется



Не используйте для брендинга пространство, предназначенное для важного пользователям контента. Например, отображение второй не убирающейся панели в верхней части экрана, которая только отображает элементы бренда, занимает место более важного для пользователя контента. Вместо этого отнеситесь к контенту с почтением и поищите менее навязчивый способ показа элементов бренда, используя особенное обрамление или шрифт или слегка изменяя фон экрана.

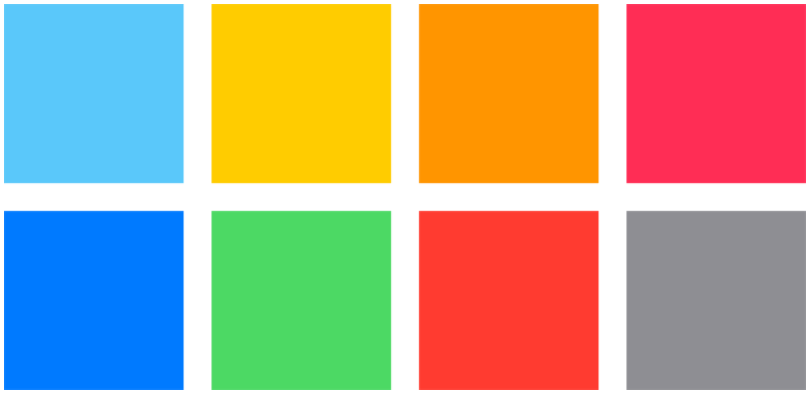
Избегайте соблазна показывать ваш логотип по всему приложению. Экраны мобильных устройств сравнительно небольшие и каждое изображение логотипа забирает место у контента, который пользователи хотели бы видеть. Более того, изображение логотипа в приложении играет не такую же роль, как изображение логотипа на веб-странице: когда пользователи заходит на страницу, он может не знать, кто ее владелец. Такая ситуация менее вероятна с приложением: вряд ли пользователь откроет приложение, не посмотрев на его иконку.

< Назад к оглавлению

Цвет и типография

Цвет усиливает коммуникацию

В iOS цвет помогает указать на интерактивность, придать жизненность и показать визуальную целостность. Встроенные приложения используют палитру чистых, ясных цветов, которые отлично смотрятся по одиночке и в комбинациях друг с другом как на светлом, так и на темном фоне.



Если вы используете несколько разработанных вами цветов, убедитесь, что они сочетаются между собой. Например, если в основе вашего приложения лежат пастельные оттенки, нужно создать палитру сочетающихся пастельных оттенков, которые могут использоваться в других элементах приложения.

Обратите внимание на контрасты цветов в различном контексте. Например, если фон панели навигации и фон кнопок на панели недостаточно контрастны между собой, кнопки будут плохо видны. Быстрый и несложный способ узнать, достаточен ли контраст между различными цветами это посмотреть на приложение в различном освещении, включая солнечный день на улице.

Хотя просмотр приложения при различном освещении может выявить некоторые слабые стороны вашего приложения, этого недостаточно, чтобы объективно оценить приложение и его возможные результаты работы. Более научный подход заключается в измерении соотношений яркости основного цвета и фонового цвета. Чтобы посчитать это соотношение, используйте онлайн-калькулятор соотношения контрастов или посчитайте это соотношение сами. с помощью формулы из стандартов WCAG 2.0. В идеале соотношение контрастов должно быть 4,5:1 или выше.

Не забывайте о полупрозрачности панели и контенте приложения, если решите слегка подкрасить панель. Если вы хотите создать цветную панель определенного цвета, например цвета вашего бренда, вам придется поэкспериментировать прежде чем вы получите желаемый результат. Полупрозрачность панели, характерная для iOS, и цвет контента, расположенного за панелью, повлияют на окончательный цвет панели.

> Примечание для API
> Чтобы подкрасить на панели, используйте свойство **tintColor**, чтобы подкрасит саму панель, используйте свойство **barTintColor**. Узнайте подробнее об этих свойствах панели в разделах UINavigationBar Class Reference, UITabBar Class Reference, UIToolbar Class Reference и UISearchBar Class Reference.

Примите во внимание дальтонизм. Большинство людей, страдающих дальтонизмом, не различают зеленый и красный цвета. Во время тестирования вашего приложения, убедитесь, что в вашем приложении нет мест, где выбор между двумя

свойствами или значениями заключается в выборе между красным и зеленым цветами (некоторые графические редакторы имеют функции проверки на готовность приложения к пользователям, не различающим цвета). В целом, лучше всего использовать больше одного индикатора, чтобы выделить интерактивность элемента. (Подробнее об этом вы можете узнать в разделе Interactive Elements Invite Touch)

Попробуйте выбрать один основной цвет, который будет указывать на интерактивность и статус. Основные цвета во встроенных приложениях включают в себя желтый в Notes и красный в Календаре. Если вы выберите один основной цвет для обозначения статуса и интерактивности элементов, убедитесь, что другие цвета не соперничают с ним и не перекрывают его.

Избегайте использования одинаковых цветов в интерактивных и неинтерактивных элементах. Цвет - один из способов обозначения интерактивности в пользовательском интерфейсе. Если интерактивные и неинтерактивные элементы окрашены в один цвет, пользователю будет сложнее разобраться, куда нажимать.

Цвет передает информацию, но не всегда так, как вы хотите. Каждый человек интерпретирует цвет по-своему и многие страны приписывают определенные значения разным цветам. Потратьте время на то, чтобы разобраться, как ваша цветовая схема может быть интерпретирована в разных странах и культурах. Удостоверьтесь по максимуму, что цветовая схема вашего приложения дает верный посыл пользователям.

В большинстве случаев, не давайте цвету отвлекать пользователей. Если только цвет - не один из главных аспектов вашего приложения, будет лучше если он будет не резким усилением дизайна.

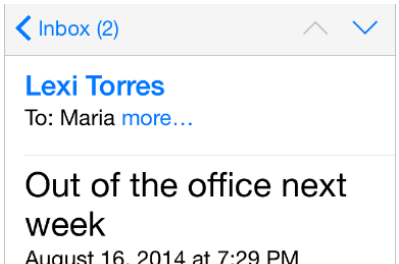
Текст всегда должен легко читаться

Прежде всего, текст должен легко читаться. Если пользователь не может прочитать текст в вашем приложении, то неважно, насколько красиво написание этого текста. Если вы применяете Динамические шрифты, вы получаете следующие преимущества:

- Автоматическое регулирование расстояния между буквами и высоты букв при любом размере шрифта
- Возможность устанавливать разные стили текста для семантически разных блоков текста (таких как Body, Footnote и Headline).
- Текст будет адаптироваться под изменения, внесенные пользователем в настройках размера шрифтов (в том числе тексты для слабовидящих).

> Примечание.
> Если вы используете разработанный вами шрифт, вы также можете изменять его размер в зависимости от пользовательских настроек размера текста. Вы ответственны за то, чтобы приложение правильно реагировало на эти изменения.

Применение Динамических шрифтов требует некоторых усилий с вашей стороны. Чтобы научиться использовать эти шрифты и убедиться, что приложение получает уведомление в момент, когда пользователь меняет настройки, смотрите раздел Text Styles в Text Programming Guide for iOS.



Расставляйте приоритеты в контенте, когда меняете размер. Не весь контент одинаково важен для пользователей. Когда пользователи выбирают больший размер шрифта, они хотят чтобы важный для них контент читался легче, но это не значит, что каждое слово на экране должно стать больше.

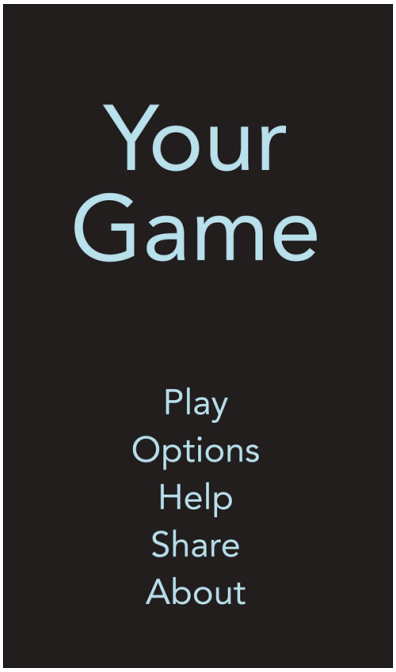
Например, когда пользователь выбирает увеличенный текст для слабовидящих, Почта отображает заголовок и текст письма более крупно, но дата и отправитель отображают более мелко, так как являются менее важным контентом.

Когда это уместно, подстраивайте макет приложения под разные размеры текста. Например, вы можете заменить макет с текстом в одну колонку на макет с текстом в две колонки, когда пользователь выбирает более мелкий шрифт текста. Если вы решите изменять макет в зависимости от размера шрифта, делайте это не для каждого конкретного размера шрифта, а для категорий шрифтов - маленький, средний и большой.

Убедитесь, что созданные вами шрифты легко читаются, независимо от того, какой шрифт выбрал пользователь. Один из способов сделать это - протестировать различные условия, в которых iOS показывает шрифты.
Например:

- Текст никогда не должен быть меньше 11 типографских пунктов, даже если пользователь выбирает очень маленький размер шрифта. Для сравнения, в разделе body при выборе крупного шрифта текст отображается в размере 17 типографских пунктов, это настройка по умолчанию.
- Чаще всего размер шрифта и размер отступа между строками отличаются на один типографский пункт при каждом изменении размера текста в настройках. Исключением являются подписи к объектам, которые всегда имеют одинаковый размер текста, расстояние между строками и расстояние между буквами при любых настройках текста - очень маленький, маленький, средний и большой.
- В трех самых мелких шрифтах расстояние между буквами относительно большое, в трех самых крупных шрифтах - относительно маленькое
- Заголовок и основной текст (body) используют один и тот же размер текста. Чтобы выделить заголовок на фоне основного текста, используется более широкий шрифт.
- Текст в navigation controller использует тот же размер шрифта, что и основной текст (body) в размере по умолчанию при большом шрифте выбранном в настройкам, то есть 17 типографских пунктов.

Старайтесь использовать один стиль шрифта во всем приложении. Смешивание нескольких шрифтов придает вашему приложению небрежный и разбитый на части вид. Вместо этого используйте один шрифт и несколько стилей и размеров. Используйте API UIfont, чтобы определить различные зоны текста по семантическому использованию, такие как основной текст и заголовок.

Рекомендуется	Не рекомендуется
	

< Назад к оглавлению

Иконки и графика

Иконка приложений

Каждому приложению нужна красивая иконка. Нередко пользователь формирует первое впечатление о качестве, назначении и надежности вашего приложения по иконке.

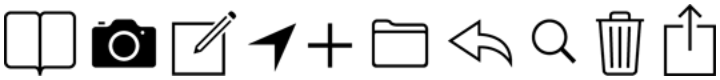
Вот несколько моментов, о которых нужно помнить, разрабатывая иконку приложения. Когда идея будет готова и вы захотите приступить к созданию иконки, обратитесь к разделу App icons для детального руководства и уточнений.

- Иконка приложения - это важная часть бренда приложения. Отнеситесь к ее дизайну как возможности рассказать историю вашего приложения и создать эмоциональную связь с пользователями.

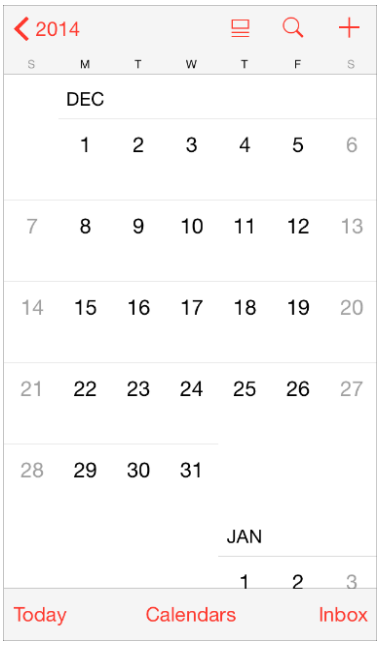
- Лучшие иконки уникальные, не загруженные лишними деталями, захватывающие и запоминающиеся
- Иконка приложения должна хорошо смотреться в разных размерах и на разных фонах. Интересные детали, которые хорошо смотрятся в крупном размере, могут казаться неясными и путанными в мелком размере.

Иконки на Bar-панели

iOS предоставляет много небольших иконок или типов контента, с помощью которых можно создать собственные иконки для bar-панелей, панелей управления и панелей навигации. Использовать встроенные иконки как можно чаще - хорошая идея, потому что пользователи уже с ними хорошо знакомы.



Если вам нужны иконки для других действий или типов контента, вы можете создать собственные иконки. Дизайн этих небольших обтекаемых иконок сильно отличается от дизайна иконки приложения. Если вы хотите создать свои иконки для bar-панелей, подробные инструкции описаны в разделе Bar button icons.



Обратите внимание, что вместо иконок вы можете использовать текст в панелях навигации или панелях управления. Например, календарь использует текст “Сегодня”, “Календари” и “Входящие” вместо иконок в панели управления.

Чтобы решить, подходят вам больше иконки или текстовая навигация, подумайте, сколько иконок будут отображаться на экране одновременно. Слишком много иконок на экране делают приложение трудным для понимания. Также не забывайте, что это решение зависит от того, как приложение будет выглядеть в горизонтальной ориентации, так как в горизонтальной regular environment обычно есть больше места для текста в bar-панелях.

Графика

Приложения на iOS чаще всего богаты графикой. Независимо от того, показываете ли вы фото пользователя или создаете оригинальные иллюстрации, вот несколько советов, которым нужно следовать.

Поддерживайте дисплей Retina. Убедитесь, что вы поддерживаете высокое разрешение для иллюстраций и графики в вашем приложении. В частности, для iPhone 6 Plus нужны @3x assets, а для всех остальных устройств с высоким разрешением на iOS - @2x assets.

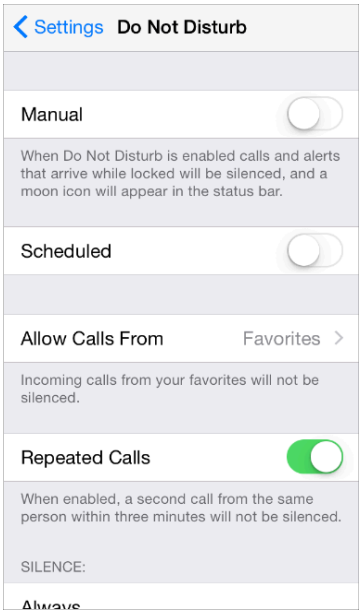
Показывайте фото и графику в их оригинальном масштабе и не увеличивайте их больше, чем на 100%. Вы же не хотите, чтобы иллюстрации и графика выглядели слишком растянутыми или большими. Позвольте пользователю самому решить, хочет он увеличивать изображение или нет.

Не используйте в вашем приложении изображения, которые копируют продукты Apple. Это изображения защищены копирайтом и дизайн продуктов может часто меняться.

< Назад к оглавлению

Терминология и Формулировки

Каждое слово, которое отображается на экране вашего приложения - это часть вашего диалога с пользователем. Используйте этот диалог, чтобы сделать работу с приложением более ясной и комфортной.



Settings - одно из самых необходимых приложений для любого пользователя, поэтому в нем используется простой и понятный язык, объясняющий пользователю, что он может сделать. Например, Settings => Do Not Disturb объясняет собой эффект от различных опций без использования технического сленга, который может оказаться сложным для не очень продвинутого пользователя.

Используйте терминологию, в знании пользователем которой вы точно уверены. Используйте свои знания о потенциальных пользователях, чтобы определить, уместны ли для них те слова и

фразы, которые вы планируете использовать. Например, технический сленг вряд ли будет уместен в приложении, ориентированном на не продвинутых в технологиях пользователей, но в приложении для технически подкованных людей он может даже создавать особую ценность.

Используйте неформальный и дружелюбный тон, но не фамильярничайте. Старайтесь не звучать неестественно и сковано, но и не впадайте в другую крайность - не звучите слишком беззаботно или покровительственно. Помните, что пользователь скорее всего не один раз прочитает текст в вашем приложении и то, что звучало умно при первом прочтении, может начать раздражать в будущем.

Думайте, как редактор в газете: избегайте лишних слов. Когда текст в вашем пользовательском интерфейсе краток и понятен, пользователи поймут его просто и быстро. Определите самую важную информацию, преподнесите ее в сжатом виде и сделайте заметной на экране. Таким образом пользователям не придется много читать, чтобы найти то, что они искали или разобраться, что делать дальше.

Давайте элементам управления короткие названия или понятные иконки. Люди должны с первого взгляда понимать, за что отвечает элемент управления.

Будьте точными, описывая даты. Использование в приложении дружелюбных и понятных слов как *сегодня* и *завтра* только приветствуется. Однако это может запутать пользователя, если ваше приложение не отслеживает местоположение пользователя. Например, представьте что мероприятие начинается незадолго до полуночи. Для пользователя в одном часовом поясе мероприятие будет начинаться “сегодня”, но для человека в смежном часовом поясе это мероприятие началось уже “вчера”.

Используйте возможность создать диалог с потенциальными пользователями с помощью хорошего описания приложения в AppStore. Кроме точного описания приложения и обращения внимания пользователей на качества, которые скорее всего привлекут больше всего внимания, убедитесь сделать следующее:

- **Проверьте все грамматические, пунктуационные и орфографические ошибки.** Хотя не все обращают внимание на эти ошибки, некоторых пользователей они могут оттолкнуть и создать негативное впечатление о приложении.
- **Как можно реже пишите слова заглавными буквами.** Некоторые слова, написанные заглавными буквами, помогают привлечь внимание людей, но когда целый абзац написан таким образом, его становится тяжело читать. Это так же может создать впечатление, что вы кричите на пользователя.
- **Возможно, стоит описать улучшения в работе приложения.** Если новая версия приложения содержит в себе решения старых технических проблем, которые пользователи очень ждали, упомяните это в описании приложения.

< Назад к оглавлению

Интеграция с iOS

Интеграция с iOS значит захватывающий и восхитительный опыт работы, который смотрится “родным” на платформе iOS. Но это не значит, что приложение должно выглядеть как копия встроенного приложения.

Лучший способ интегрировать ваше оригинальное приложение в платформу - это разобраться в идеологических основах iOS - они подробно описаны в разделе Designing for iOS - и разобраться, как ваше приложение может проявить эти принципы. Делая это, следуйте советам в этом разделе, чтобы удовлетворить ожидания пользователей.

Правильно используйте стандартные UI элементы

Как можно чаще старайтесь использовать стандартные элементы пользовательского интерфейса, доступные в UIKit. Когда вы используете стандартные элементы вместо создания собственных, то выгоду получаете и вы, и пользователь:

- Стандартные элементы пользовательского интерфейса автоматически получают обновления, если iOS представляет новый дизайн - созданные вами элементы не получают обновлений.
- Стандартные элементы пользовательского интерфейса предлагают различные способы кастомизации их внешнего вида и функций. Например, все views (то есть объекты, которые произошли от UIView) имеют возможность затемнения оттенка, поэтому добавлять цвета в ваше приложения становится проще. Чтобы узнать подробнее о добавлении цвета в элементы пользовательского интерфейса, смотрите раздел “Using Tint Color” в iOS 7 UI Transition Guide.
- Людям проще со стандартными элементами пользовательского интерфейса, так как они сразу же понимают как пользоваться ими в вашем приложении.

Чтобы воспользоваться всеми преимуществами использования стандартный элементов пользовательского интерфейса, необходимо выполнять следующие условия:

Следуйте указаниям при работе с каждым элементом пользовательского интерфейса. Когда элемент пользовательского интерфейса выглядит и работает так, как люди к этому привыкли, их прошлый опыт поможет им быстрее начать работать с вашим приложением. Вы можете найти инструкции к элементам пользовательского интерфейса в следующих разделах: Bars, Content Views, Controls и Temporary Views.

Не смешивайте стили элементов пользовательских интерфейсов из разных версий iOS. Пользователи вероятнее всего запутаются, если увидят элементы пользовательского интерфейса, которые принадлежат не к той версии iOS, которая установлена на их устройстве.

Избегайте создания собственных элементов пользовательского интерфейса, которые выполняют стандартные действия. Для начала спросите себя, зачем вы создаете собственный элемент пользовательского интерфейса, который делает то же самое, что и стандартный элемент. Если вы хотите просто изменить внешний вид элемента, советуем вам воспользоваться функцией изменения вида стандартного элемента. Это можно сделать с помощью API для изменения внешнего вида в UIKit или опцией затемнения оттенка цвета. Если вы хотите немного изменить поведение элемента, попробуйте узнать, может ли стандартный элемент делать то же самое с помощью небольших изменений в свойствах и атрибутах. Если вам нужно разработать совершенно новое действие, лучше создать элемент, который внешне полностью отличается от стандартного.

> Совет.
> Interface Builder позволяет легко получать стандартные элементы пользовательского интерфейса, использовать API для изменения внешнего вида элемента, доступ к свойствам и атрибутам элементов. Он также позволяет применять стандартные и разработанные вами иконки на элементы контроля. Подробности об Interface Builder смотрите в разделе XCode Overview.

Не используйте заданные системой кнопки и иконки в других значениях. iOS предоставляет множество иконок и кнопок, которые вы можете использовать в своих приложениях. Удостоверьтесь, что вы правильно поняли значение этих иконок и кнопок, не опирайтесь на интерпретацию их внешнего вида. (Вы можете найти описание значения каждой иконки в разделе Toolbar и Navigation Bar Buttons и Tab Bar Icons.

Если вы не можете найти кнопку или иконку среди предоставленных системой, которая подходит для определенной функции в вашем приложении, вы можете создать свою. Некоторые советы и инструкции по создание иконок вы можете найти в разделе Bar Button Icons.

Если ваше приложение предполагает вовлекающие задачи или опыт вовлечения в виртуальную реальность, создание полностью новых и оригинальных методов управления может быть к месту. Создание уникальной environment и постепенное изучение, как ее контролировать - это то, что пользователи ожидают найти в такого рода приложениях.

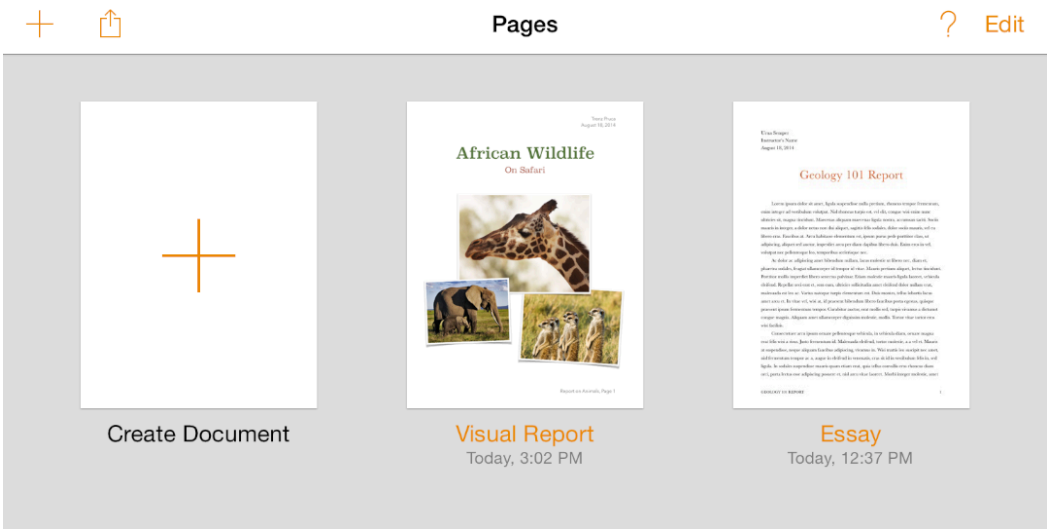
Уменьшение файлов и Работа с документами

Приложения на iOS дают людям возможность создавать документы и управлять ими, но это не значит, что пользователи должны задумываться, о системе файлов на их устройстве на iOS.

Если ваше приложение позволяет пользователям создавать и редактировать документы, будет полезно создать какой-то режим библиотеки внутри приложения, который позволит им открывать существующие документы или создавать новые. В идеале такой режим библиотеки должен иметь следующие характеристики:

- **Насыщенная графика**. Люди должны без проблем определять нужный документ, просматривая уменьшенные изображения.
- **Люди должны делать минимум движений, чтобы получить то, что они хотят.** Например, люди должны прокрутить карусель из документов или пролистать сетку документов, чтобы открыть нужный документ одним прикосновением.
- **Включите функцию создания нового документа.** Вместо того, чтобы размещать функцию создания документа в другом месте, разместите кнопку создания документа прямо в режиме библиотеки.

Например, в Pages документы пользователя отображаются в удобном формате вместе с простым способом создавать новые документы в режиме графической библиотеки.



Совет
Вы можете использовать функцию Quick Look Preview для предварительного просмотра документов в вашем приложении, даже если само приложение не может их открывать. Подробнее об этой функции в вашем приложении, смотрите раздел Quick Look.

Если ваше приложение позволяет использовать документы, которые пользователи создали в других документах, вы можете показывать модальный document picker view controller, чтобы упростить процесс доступа к документам. Document picker view controller может отображать документы из iCloud Drive и расширения Document Provider, которые связаны с другими приложениями для создания или хранения документов. Чтобы узнать больше о расширении Document Provider, смотрите раздел Document Provider Extension. Чтобы узнать больше о document picker view controller, смотрите Document Picker Programming Guide.

Уверьте пользователей, что из работа всегда будет сохранена, если только они намеренно не отменят ее или не удалят. Если ваше приложение предполагает создание и редактирование документов, не просите людей каждый раз нажимать кнопку “Сохранить”. Приложение на iOS должны сами позаботиться о сохранении действий пользователя - периодически и при выборе другого документа или приложения.

Если создание документов не является главной функцией приложения - но вы позволяете пользователям переключаться между просмотром информации и ее редактированием - стоит уточнять, нужно ли сохранять внесенные изменения. В таком случае, обычно полезно иметь кнопку “Редактировать” в режиме просмотра информации. Когда люди нажимают на кнопку “Редактировать”, она заменяется на две другие кнопки - “Сохранить” и “Отменить”. Это изменение напоминает людям, что они перешли в режим редактирования и им нужно будет сохранить внесенные изменения, а кнопка “Отменить” дает им возможность выйти без сохранения изменений.

Если необходимо, станьте конфигурируемыми

Некоторые приложения должны давать пользователю возможность изменить настройки работы приложения, но большинство приложений могут избежать этого или отложить этот момент. Успешные приложения хорошо работают для большинства пользователей, но также предлагают подстроить приложение под привычки работы пользователя.

Когда вы разрабатываете приложение так, чтобы оно подходило большинству пользователей, вы снижаете необходимость в дополнительных настройках. Если вам нужна информация о пользователе, сделайте поиск по системе устройства, а не спрашивайте эти данные у пользователя. Если вы считаете, что вам обязательно нужно внедрить настройки приложения, которые пользователь редко будет менять, смотрите раздел The Settings Bundle в iOS App Programming Guide for iOS, чтобы научиться поддерживать ихв коде вашего приложения.

Как можно чаще предлагайте настроить работу приложения в главном пользовательском интерфейсе. Размещение доступных опций в главном пользовательском интерфейсе создает впечатление, что настройка этих опций - важная задача и их нужно регулярно обновлять. Если эти настройки вряд ли нуждаются в частном обновлении, лучше запусть их в отдельном view.

Если необходимо, позвольте пользователю переходить прямо к настройкам вашего приложения в Settings. В частности, если вы показываете сообщение, которое указывает на путь к настройкам (например, Перейдите в Настройки - Мое Приложение - Личные данные - Сервисы геолокации”), замените это описание кнопкой, которая открывает это же меню сразу в Settings. Узнать, как это сделать, вы можете в разделе Settings Launch URL в UIApplication Class Reference.

Используйте технологии iOS по максимуму

iOS предлагает множество технологических решений, которые поддерживают частые задачи и варианты развития событий в том виде, в котором их ожидают увидеть пользователи. Эти ожидание означают, что практически в любой ситуации интеграция поддерживаемых системой технологий в ваше приложение будет лучше, чем разработка с нуля.

Некоторые технологии iOS - например Multitasking и VoiceOver - являются системными требованиями ко всем приложениям. Другие технологии активируют определенную функцию в приложении, например, использование купонов и подарочных карт (Passbook), возможность совершения покупки в приложении (In-App Purchase), отображение рекламы в приложении (iAd Rich Media Ads), интеграция с Game center и поддержка iCloud.

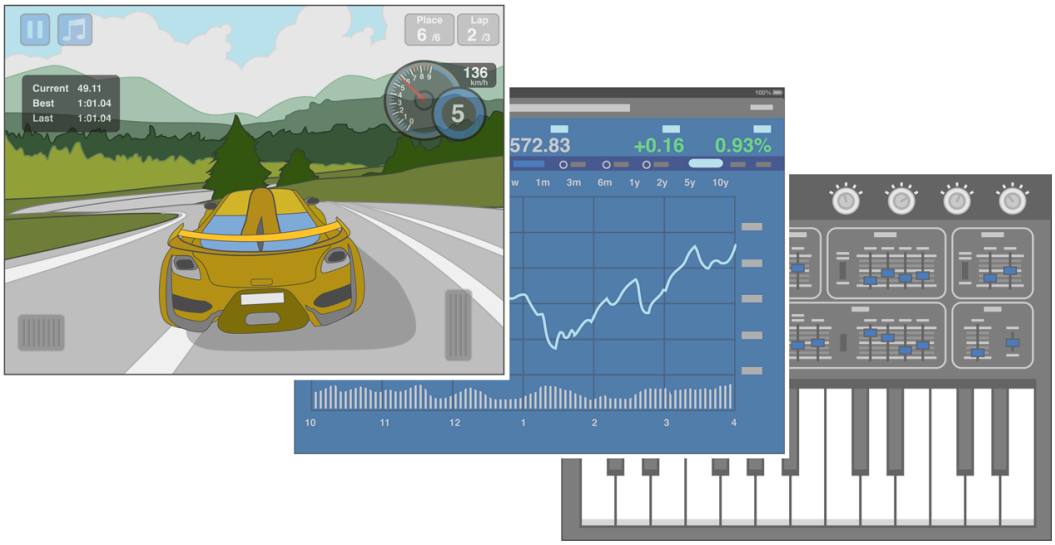
< Назад к оглавлению

Дизайн Стратегии

Принципы дизайна

Эстетическая целостность

Эстетическая целостность не измеряет красоту оформления приложения и не характеризует его стиль. Она скорее определяет, насколько хорошо внешний вид приложения и поведение сочетаются с его функциями и дает осмысленный посыл пользователю.



Людям важно, чтобы приложение выполняло обещанные функции, но также сильно – порой на подсознательном уровне – их мнение о приложении зависит от его внешнего вида и поведения. Например, приложение, помогающее решать сложные задачи, может фокусировать на главной задаче, оставляя декоративные элементы едва заметные и не отвлекающими внимание с помощью использования стандартных контроллеров и ожидаемого поведения. Приложение передает ясное и целостное сообщение о своей цели и сущности, которые помогают пользователям доверять ему. Но если бы это же приложение создавало смешанное впечатление путем использования слишком яркого, отвлекающего или невыдержанный, то пользователи бы засомневались, можно ли считать это приложение надежным и вселяющим доверие.

С другой стороны, от приложения, которое представляет собой увлекательный процесс - например, игры – пользователи ожидают захватывающего внешнего вида, который обещает веселье и восторг и стимулирует желание открывать что-то новое. Люди не ожидают, что это приложение поможет им решить сложные задачи, но они ожидают, что внешний вид и поведение игры будут соответствовать ее цели.

Последовательность

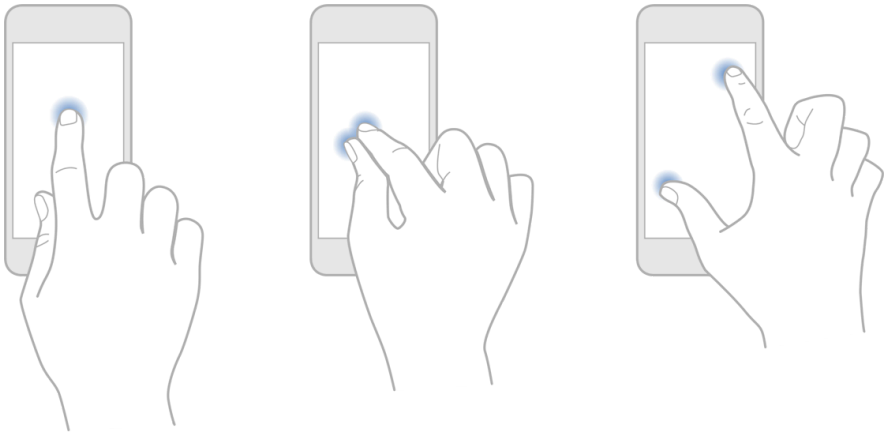
Последовательность помогает людям переносить их знания с одной части пользовательского интерфейса в другую и из одного приложения в другое. Последовательное приложение не жалкая копия других приложений и не устаревшее с точки зрения дизайна. Наоборот, оно уделяет внимание стандартам и парадигмам, к которым привыкли пользователи и представляет внутренне последовательный опыт работы с приложением.

Чтобы определить, соответствует ли ваше приложение на iOS приниципам последовательности, ответьте на следующие вопросы:

- Соответствует ли ваше приложение стандартам iOS? Использует ли оно предлагаемые системой элементы контроля, views или иконки правильно? Использует ли она встроенные в устройство функции так, как это ожидают пользователи?
- Последовательно ли работает приложение? Используется ли одниковая терминология и стиль текста? Всегда ли одинаковая иконка имеет одно и то же значение? Могут ли люди предсказать, что произойдет, если они совершат одно и то же действие в разных частях приложения? Работает ли разработанный вами пользовательский интерфейс одинаково по всему приложению?
- Похоже ли приложение на его предыдущие версии? Сохранились ли условия и значения теми же? Основные концепции и основные функции в большинстве своем остались те же?

Прямое управление

Когда люди напрямую управляют объектами на экране вместо того, чтобы использовать отдельные контроллеры для управления ими, они больше вовлечены в процесс и им легче понять результаты их действий.



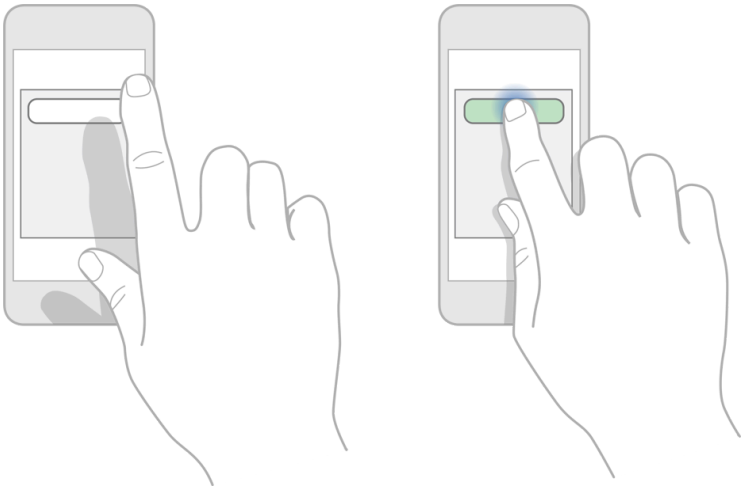
С использованием интерфейса Multi-Touch , люди могут развести пальцы на экране и увеличить конкретный контент на экране. В игре, пользователи двигаются и взаимодействуют с объектами на экране напрямую - например, в игре может быть цифровой замок, который пользователь прокручивает для того, чтобы открыть его.

В приложениях на iOS, люди занимаются прямым управлением, когда они

- Поворачивают или каким-либо образом двигают устройство, чтобы управлять объектами на экране
- Используют жесты для управления объектами на экране
- Могут видеь результаты их действий ясно и немедленно

Фидбэк

Фидбэк подтверждает действия пользователей, показывает им результаты действий и дает им обновленную информацию о прогрессе выполнения задания.



Встроенные приложения на iOS предоставляют воспринимаемый фидбэк в ответ на каждое действие пользователя. Список объектов и объектов управления слегка подсвечивается, когда люди нажимают на них и - во время операций, которые длятся более нескольких секунд - элемент управления показывает ход прогресса.

Ненавязчивая анимация может давать людям значимый фидбэк, который поможет прояснить результат их действий. Например, списки могут использовать анимацию при добавлении новой строки, чтобы люди могли отслеживать их действия визуально.

Звуковые оповещения тоже могут быть полезным фидбэком, но они не должны быть единственной формой фидбэка, так как пользователи не всегда могут слышать их устройства.

Метафоры

Когда виртуальные объекты и действия в приложении похожи на знакомые действия - привычные в цифровом мире или в реальном мире - пользователи быстрее понимают, как работает приложение.

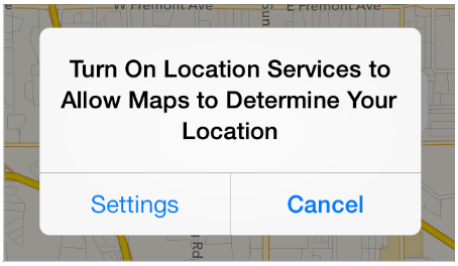
Лучше, если приложение будет использовать метафоры, намекая на способ использования или опыт, но не позволяя метафоре усилить негативные ограничения объекта или действия в приложении по сравнению с реальными.

Приложения на iOS предлагают большой простор для метафор, так как люди физически взаимодействуют с экраном. Метафоры в iOS включают в себя:

- Сдвигание слоев views, чтобы просмотреть контент под ними
- Перелистывание, смахивание и перетаскивание объектов в игре
- Нажатие на переключатели, сдвигание слайдов и кружение “каруселей” выбора
- Перелистывание страниц журнала

Пользовательский контроль

Люди - а не приложения - должны начинать и контролировать действия. Приложение может предложить ход действий или предупредить о негативных последствиях, но будет ошибкой давать контроль принятия решений приложению, а не пользователю. Лучшие приложения находят баланс между передачей контроля и возможностью пользователю и оберегания его от нежелательных последствий.



Пользователи чувствуют больший контроль над приложением, когда его поведение и элементы управления привычные и предугадываемые. И когда действия просты, пользователи могут легко понять их и запомнить.

Люди ожидают, что у них будет достаточно возможностей отменить операцию, пока она еще не началась и они ожидают, что смогут подтвердить свое намерение совершить потенциально пагубное действие. Наконец, люди ожидают, что они смогут остановить операцию, которая находится в процессе выполнения.

< Назад к оглавлению

От концепции к продукту

Охарактеризуйте ваше приложение

Характеристика приложения - это короткое и конкретное определение главной цели приложения и его предполагаемой целевой аудитории.

Создание характеристики приложения на раннем этапе процесса разработки поможет вам превратить идею и список функций в связный продукт, который людям захочется иметь. В течение процесса разработки, используйте характеристику приложения, чтобы решить, подходят ли приложению определенные функции и поведение. Проделайте следующие шаги, чтобы создать сильную характеристику приложения.

1. Создайте список всех функций и характеристик, которые могут понравиться пользователям

Начинайте мозговой штурм. На этом этапе постарайтесь собрать все задачи, которые подходят к вашей основной идее приложения. Если список слишком длинный, не переживайте, вы сократите его позже.

Представим, что ваша первоначальная идея - это разработка приложения, которое поможет людям ходить в магазин за продуктами. Когда вы думаете об этом занятии, вы придумываете список похожих задач - то есть, потенциальных функций - в который пользователь может заинтересоваться. Например:

- Создание списков
- Получение рецептов
- Сравнение цен
- Нахождение магазинов
- Добавление примечаний в рецептах
- Получение и использование купонов
- Просмотр демо-роликов с готовкой
- Знакомство с кухнями народов мира
- Поиск ингредиентов-заменителей

2. Определите, кто ваши пользователи

Теперь нужно определить, что отличает пользователей вашего приложения от всех других пользователей устройств на iOS. В контексте вашей основной идеи, что важно для них? Используя пример с покупкой продуктов, вы можете определить, каким образом ведут себя пользователи из вашей целевой аудитории:

- Обычно готовят дома или предпочитают готовые блюда?
- Часто используют купоны или считают их пустой тратой времени?
- Людям искать необычные ингредиенты или редко покупают что-то кроме самого главного?
- Строго следуют рецептам или используют их только как вдохновение перед экспериментом?
- Делают небольшие покупки часто или редко закупаются большим количеством продуктов?
- Хотят вести несколько списков для различных целей или хотят простое напоминание, что нужно купить по дороге домой?
- Предпочитают определенные бренды или смиряются с самыми удобными альтернативами?
- Покупают примерно то же самое каждый раз или покупаю то, что написано в рецепте

После анализа этих вопросов, представьте, что вам нужно остановиться на трех характеристиках, которые лучше всего характеризуют вашу аудиторию: любят экспериментировать, часто спешат и экономят, если это не занимает слишком много времени.

3. Отфильтруйте список функций с помощью характеристик аудитории

Если после определения ключевых характеристик пользователей в вашем списке осталось всего несколько функций приложения, вы на правильном пути. Отличные приложения для iOS нацелены на конкретную задачу, в выполнении которой они хотят помочь.

Например, вспомните длинный список характеристик приложения из Шага 1. Все они полезны, но не все будут интересны аудитории, которую вы выделили в Шаге 2.

Когда вы просмотрите список функций в контексте целевой аудитории, вы поймете, что приложение должно сфокусироваться на трех функциях: создание списков, получение и использование купонов и получение рецептов.

После этого вы можете отредактировать свою характеристику приложения, кратко изложив, что делает приложение и для кого. Хорошая характеристика приложения для походов за покупками может быть: “Инструмент для создания списков покупок для бережливых людей, которые любят готовить”.

4. Не останавливайтесь на этом

Используйте вашу характеристику приложения на пути всего процесса разработки, чтобы определить, подходят ли определенные функции, элементы управления и терминология. Например:

Когда вы решаете добавить новую функцию, спросите себя, является ли она обязательной для основной цели вашего приложения и для вашей целевой аудитории. Если нет, отложите эту функцию, она может стать основой для другого приложения. Например, если вы решили, что пользователи заинтересованы в необычной готовке, то вряд ли они оценят функцию, дающую информацию о готовых смесях для тортов или замороженных готовых блюдах.

Когда вы думаете над внешним видом и поведением пользовательского интерфейса, спросите себя, понравится ли пользователям простой и лаконичный дизайн или более открытый тематический дизайн. Руководствуйтесь тем, что люди ожидают достичь с использованием вашего приложения: решить сложную задачу, получить быстрый ответ, вникнуть в подробный контент или развлечься. Например, если приложение для списка покупок должно быть понятным и быстрым в использовании, вашей аудитории скорее всего понравится тематический пользовательский интерфейс с множеством красивых картинок продуктов и блюд.

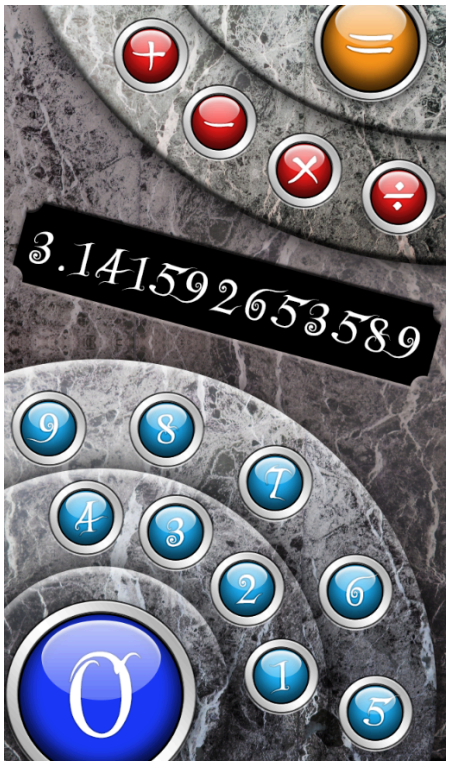
Когда вы думаете над используемой терминологией, стремитесь соответствовать уровню знаний вашей аудитории. Например, хоть ваша аудитория и не состоит из шеф-поваров, вы можете быть уверены, что они хотят видеть правильные названия ингредиентов и способов приготовления блюд.

Используйте кастомизацию в зависимости от задачи

Лучшие приложения для iOS представляют собой баланс между кастомизированным пользовательским интерфейсом и ясностью цели и простотой использования. Чтобы достичь этого баланса в вашем приложении, подумайте над кастомизацией на раннем этапе процесса разработки. Поскольку заботы о брендинге, оригинальности и годности для успешной продажи влияют на решения о кастомизации, сложно сфокусироваться только на том, как кастомизация повлияет на пользовательский опыт работы с приложением.

Начните с рассмотрения отдельных задач в вашем приложении: Как часто люди будут ими пользоваться и в каких условиях?

Например, представьте приложение-калькулятор, в котором используется сложный и артистичный стиль и выдуманный макет для отображения привычных элементов калькулятора. Тщательно проработанная графика и выдуманный макет не мешают людям в понимании, как нажимать на кнопки и получить результат. Но для людей, которым нужно просто выполнить работу, новизна такого опыта быстро приедается и со временем красивый и уникальный пользовательский интерфейс становится препятствием.



Для сравнения, вспомните о приложении GarageBand. GarageBand мог бы позволять людям писать музыку без отображения красивых и реалистичных музыкальных инструментов, но это сделало бы приложение менее интуитивным и менее развлекательным в использовании. В GarageBand кастомизированный пользовательский интерфейс не только показывает, как пользоваться приложением, но и делает основную задачу - то есть создание музыки - более простой для совершения.

Когда вы думаете о том, как кастомизация может улучшить или ухудшить задачу, которую решает приложение, помните о следующих вещах:

Всегда делайте кастомизацию только если на неё есть причина. В идеале кастомизация пользовательского интерфейса делает решение задач проще и улучшает пользовательский опыт работы с приложением. Как можно чаще давайте задачам быть причиной кастомизацией в приложении.

Как можно чаще избегайте увеличение смысловой нагрузки на пользователя. Люди привыкли к внешнему виду и поведению стандартных элементов пользовательского интерфейса, поэтому они пользуются ими, не тратя время на то, чтобы разобраться в их принципе работы. Когда пользователи встречают элементы интерфейса, с которыми они прежде не работали, пользователи теряют преимущество наличия прошлого опыта. Если только разработанные вами элементы не делают работу еще проще, пользователи могут быть не рады факту того, что им придется изучать новые павила, которые потом не смогут быть применены к другим приложениям.

Будьте внутренне последовательны. Чем больше изменений вы вносите в пользовательский интерфейс, тем важнее последовательность в дизайне и поведении элементов во всём приложении. Если пользователи тратят время, чтобы разобраться в элементах управления, созданных вами, они ожидают, что эти знания пригодятся им во всем приложении.

Всегда относитесь к контенту с почтением. Поскольку стандартные элементы пользовательского интерфейса привычны пользователю, они не соревнуются с контентом за внимание пользователя. Когда вы кастомизируете пользовательский интерфейс, убедитесь что он не затмевает собой важный для пользователей контент. Например, если ваше приложение позволяет пользователям просматривать видео, вы можете создать кастомизированные элементы управления просмотром. Но независимо от того, разрабатываете ли вы элементы управления сами или используете стандартные, важнее то, чтобы они исчезали в момент, когда пользователь начинает просматривать контент, и появлялись после повторного прикосновения к экрану.

Подумайте дважды перед тем, как менять дизайн стандартного элемента управления. Если ваши планы пошли дальше, чем кастомизация стандартного элемента управления, убедитесь что переделанный элемент контроля предоставляет не меньше информации, чем стандартный. Например, если вы создали switch control, который не указывает на наличие противоположного состояния, люди могут не понять, что это two-state control.

Обязательно тщательно протестируйте на пользователях созданные вами элементы пользовательского интерфейса. Во время тестирования, внимательно наблюдайте за пользователями, что понять, могут ли они

предварительно понять, что делают созданные вами элементы и просто ли пользователям взаимодействовать с ними. Если, например, вы создали элемент управления, который занимает пространство меньше чем 44x44, у людей могут проблемы с его активацией. Или, если вы создаете view, функционирования которого отличается при прикосновении и смахивании, убедитесь, что функционал этого view стоит дополнительного усилия при взаимодействии с ним.

Создайте прототип и повторите снова

До того, как вы инвестируете большие инженерные ресурсы в воплощение вашей разработки в жизнь, будет лучше создать прототип для тестирования пользователями.Если даже несколько ваших коллег протестируют прототип, вы получите пользу от их свежего взгляда на приложение и опыт работы.

На ранних стадиях разработки вы можете делать прототип на бумаге или с помощью
“каркаса” приложения (wireframe), чтобы показать главные views и элементы управления и переходы между экранами. Вы можете получить полезный фидбэк после тестирования на каркасе, но их незаполненность может ввести тестировщиков в заблуждение. Это происходит потому, что людям сложно представить, как будет работать приложение, когда каркас будет заполнен реальным контентом.

Вы получите еще более ценный фидбэк, создав прототип “во плоти”, который будет самостоятельно работать на устройстве. Когда люди могут испробовать приложение на устройстве, они с большей вероятностью обнаружат места, в который приложение работает не так, как им хотелось бы или где процесс работы слишком сложен.

Самый простой способ создать заслуживающий доверия прототип - это использование шаблона XCode на основе эскизов раскадровки для построения базового приложения и наполнить его подходящим заполняющим контентом. (**Файл раскадровки (storyboard)** захватывает весть пользовательский интерфейс приложения, учитывая переходы между экранами). Затем установите прототип на устройство, чтобы тестировщики могли получить наиболее реалистичный опыт использования приложения.

Прототип приложения не нужно заполнять большим количеством контента или активировать все элементы управления, но нужно предоставить достаточно контекста для формирования реалистичного опыта использования. Нацеливайтесь на баланс между типичным опытом пользователя и более необычными критическими ситуациями. Например, если приложение вероятнее всего будет работать с длинными списками объектов, будет неправильно использовать в прототипе списки из 1-2 объектов. Что касается тестирования пользовательского взаимодействия, до тех пор пока тестировщики могут прикоснуться к части экрана для перехода следующий логически view или совершения основной задачи, они смогут даль ценный фидбэк.

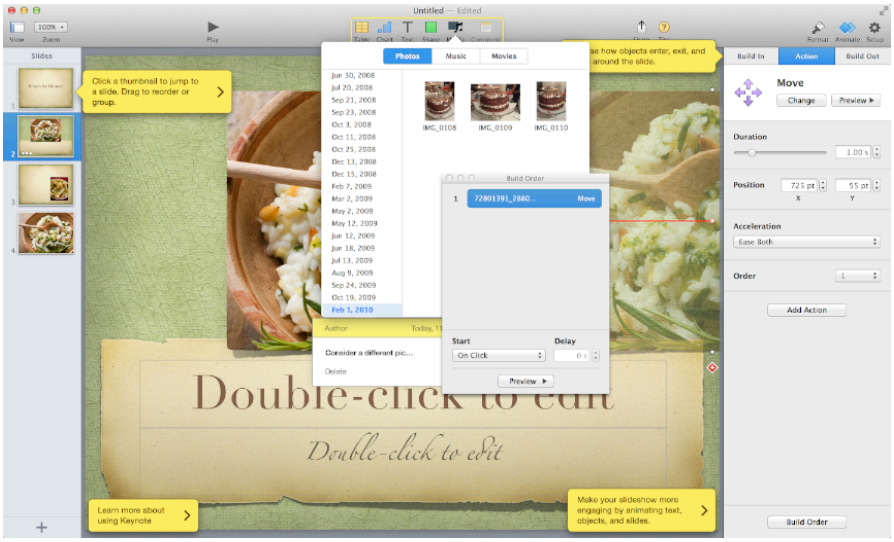
Когда вы основываете свой прототип на шаблоне из приложения Xcode, вы получаете много бесплатного функционала и вносить изменения по результатам фидбэка довольно просто. За короткий период времени вы сможете протестировать несколько версий прототипа перед тем, как вы укрепите свою разработку и потратите ресурсы на ее превращение в жизнь. Чтобы разобраться в Xcode лучше, смотрите раздел Xcode Overview.

< Назад к оглавлению

Реальный опыт: от десктопной версии к iOS

Keynote на iPad

Десктопная версия Keynote - мощное и гибкое приложения для создания презентаций высшего класса. Люди любят Keynote за совмещение простоты использования и тонкого контроля за бесчисленным количеством деталей, таких как анимация и свойства текста.



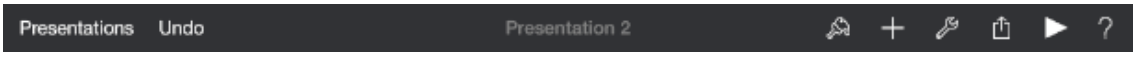
Keynote на iPad сохраняет основу десктопной версии Keynote и заставляет пользователей чувствовать себя как дома, создавая пользовательский опыт, который

- Фокусируется на контенте пользователя
- Снижает сложность без уменьшения количества возможностей
- Предлагает быстрые клавиши, которые дают больше контроля и восторга от работы
- Приспосабливает в себе отличительные свойства десктопной версии
- Дает фидбэк и сообщения с помощью красноречивой анимации

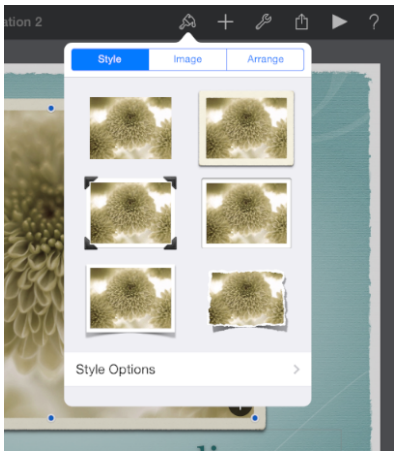
Пользователи Keynote сразу же понимают, как пользоваться приложением на iPad, потому что ожидаемый функционал передается с помощью привычных для iPad инструментов. Новые пользователи легко учатся пользоваться Keynote на iPad, потому что они могут напрямую управлять контентом в простой и привычной форме.

Трансформация Keynote с десктопной версии до iPad основана на множестве модификаций и изменений в дизайне - от еле заметных до очень больших. Вот несколько самых видных изменений:

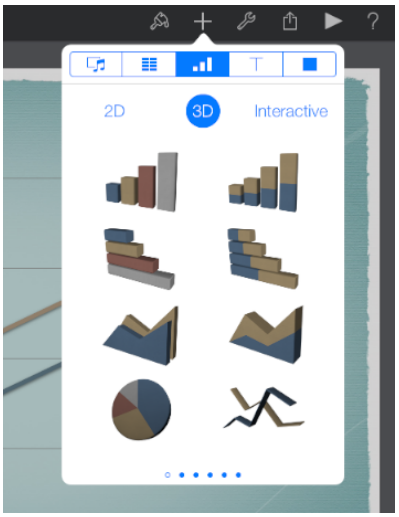
Более очевидная панель управления. На панели управления показаны только несколько объектов, но они дают пользователю постоянный доступ ко всем функциям, необходимым для создания контента.



Упрощенный, приоритизированный инструмент проверки контента, который реагирует на фокус пользователя. Инструмент проверки контента в Keynote на iPad автоматически содержит в себе инструменты и свойства, которые нужны людям для редактирования выбранного объекта. Часто люди могут совершить все необходимые изменения уже в первом открывшемся окне инструмента проверки. Если им нужно изменить более редкие свойства, они могут опуститься до других режимов просмотра в инструменте.



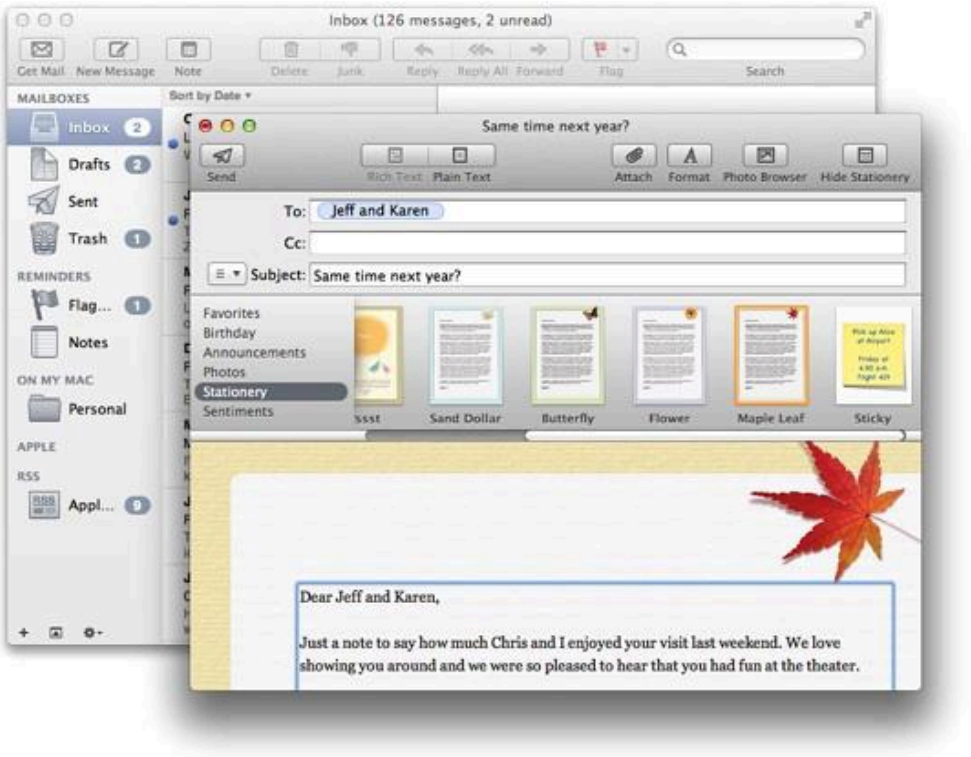
Множество подготовленных разных коллекций стилей. Люди могут легко изменить внешний вид объектов (таких как графики и таблицы) с помощью встроенных стилей. Кроме цветовых схем каждая коллекция включается в себя предустановленные свойства - такие как заголовки таблиц и разделение осей, которые скоординированы с основной темой.



Прямое управление контентом, усиленное значимой анимацией. В Keynote на iPad, пользователь перетаскивает слайд в новую позицию, закручивает объект, чтобы повернуть его и прикосается к объекту, чтобы выбрать его. Впечатление прямого контроля усилено ответной анимацией в Keynote. Например, слайд слегка пульсирует, когда пользователь передвигает его и, когда его помещают в новое место,слайды вокруг него разъезжаются, чтобы для нового слайда появилось место.

Почта на iPhone

Почта - это одно из самых видных, удобных в использовании и ценимых приложений в операционной системе OS. Это также мощная программа, позволяющая пользователям создавать, получать, приоритизировать и хранить электроннную почту, отслеживать необходимые действия и события и создавать заметки и приглашения. Почта на рабочем столе предлагает мощный функционал в нескольких окнах.



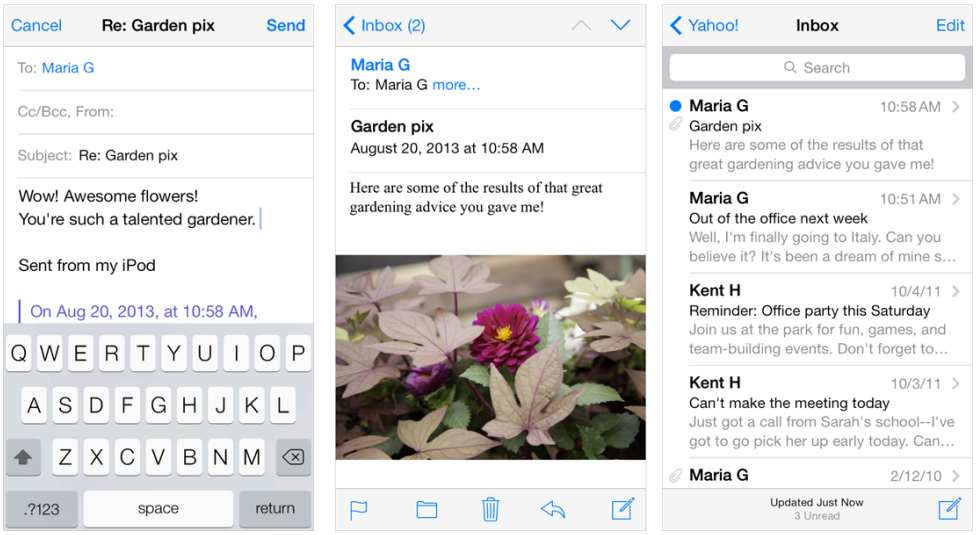
Почта на iPhone сфокусировалась на основном функционале десктопной версии Почты, помогая людям получать, создавать, отправлять и организовывать их сообщения. Почта на iPhone предлагает более сжатый функционал в пользовательском интерфейсе, адаптированном для опыта работы на мобильных устройствах, включающего в себя:

- Более рациональный внешний вид, который размещает контент пользователя впереди и в центре
- Разные views для работы над разными задачами
- Интуитивно понятная структура информации, которая легко расширяется
- Мощные инструменты редактирования и упорядочивания, доступные по мере необходимости
- Ненавязчивая, но выразительная анимация, которая предает информацию и дает фидбэк

Важно понимать, что Почта на iPhone не является более хорошим приложением, что десктопная версия Почты. Скорее, это Почта, переделанная под пользователей мобильных устройств. Концентрируясь на некоторых функциях десктопной версии и представляя их в привлекательном обтекаемом пользовательском интерфейсе, Почта на iPhone дет пользователям основной опыт работы с Почтой в мобильном формате.

Чтобы адаптировать опыт работы с Почтой в мобильный формат, в Почте на iPhone были сделаны следующие нововведения в пользовательском интерфейсе.

Отдельные и четко сфокусированные экраны. Каждый экран отображает один аспект Почты: список аккаунтов, список почтовых ящиков, список сообщений, просмотр письма и окно создания нового письма. На одном экране люди прокручивают содержимое, чтобы увидеть весь контент.



Простая и предсказуемая навигация. Одним нажатием на экран люди переходят от общего (список аккаунтов) к частному (письмо). Каждый экран показывает заголовок, объясняющий пользователю, где он находится и кнопку “Назад”, которая делает возврат к предыдущим шагам проще.

Простые элементы контроля, на которые можно нажать и доступные в любое время. Поскольку написание письма и проверка почтового ящика - основные действия, которыми люди хотят пользоваться в любом контексте, Почта на iPhone делает их доступными на нескольких экранах. Когда люди просматривают сообщение, такие функции как ответить, переместить и отправить в корзину доступны, поскольку они взаимодействуют с просматриваемым сообщением.

Разным типа фидбэка для разных задач. Когда люди удаляют письмо, оно перемещается анимацией на иконку корзины. Когда люди отправляют сообщение, они видят прогресс отправки и слышат специальный звук, когда отправка завершена. Неброский текст на панели управления в списке сообщений позволяет сразу понять, когда почтовый ящик был обновлен в последний раз.

Веб-контент на iOS

Safari на iOS предлагает выдающийся опыт просмотра веб-контента на iOS устройствах. Люди ценят четкий текст, контрастные изображения и возможность регулировать просмотр с помощью поворота устройства, сведения или разведения пальцев на экране или простого прикосновения.

Стандартные веб-сайты хорошо отображаются на iOS устройствах. В частности, вебсайты, которые определяют тип устройства и не используют плагины, отображаются отлично на iPhone и iPad с совсем небольшими изменения или вообще без них.

Кроме этого, наиболее успешные вебсайты обычно:

- Устанавливают подходящую область просмотра для устройства, если ширина страницы должна соответствовать ширине экрана устройства.
- Избегают фиксирования позиций в CSS, чтобы контент не съезжал с экрана во время увеличения масштаба страницы
- Включение пользовательского интерфейса для сенсорного экрана, который не реагирует на привычные клики

Иногда другие изменения также должны быть сделаны. Например, веб-приложения всегда устанавливают нужную ширину области просмотра и поэтому часто закрывают собой пользовательский интерфейс Safari на iOS. Чтобы узнать больше о такого рода изменениях, смотрите раздел Configuring the Viewpoint в Safari Webcontent Guide и Configuring Web Applications в Safari Webcontent Guide.

Вебсайты могут адаптировать десктопную версию к Safari на iOS и другими способами:

Разместить клавиатуру в Safari на iOS. Когда отображаются клавиатура и форма поддержки, Safari на iPhone откроет вебстраницу в пространстве между строкой ввода URL и клавиатурой и формой поддержки.

Разместить control всплывающего меню в Safari на iOS. В десктопной версии Safari, всплывающее меню, которое содержит в себе много объектов, отображается также, как и в приложении на OS X. То есть, меню открывается, чтобы показать все объекты, расширяясь за границы окна, если необходимо. В Safari на iOS всплывающее меню отображается с использование привычных для iOS элементов, которые предлагают лучший опыт использования приложения. Например, на iPhone инструмента подбора - пикера, который представляет собой список вариантов, из которого пользователь может выбрать один. (Подробнее о picker control смотрите раздел Picker.)

< Назад к оглавлению