(Слайд 1) Добрый день, коллеги. Сегодня хотел бы рассказать Вам о некоторых основных библиотеках, которые используются для создания и настройки анимаций базового компонента вида модели MVC

(Слайд 2) Передо мной стояла следующая цель – провести анализ библиотек для проведения анимаций для реализации анимаций представлений в рамкках модели MVC, а также задача - найти и изучить основные библиотеки, сравнить их достоинства и недостатки. Понять, что необходимо сделать, чтобы получить удобную библиотеку для анимирования, которую можно было бы использовать как на работе, так и в личных целях

(Слайд 3) План рассказа будет следующий – расскажу, как реализованы анимации в iOS, про базовый компонент и его свойства. После перейду к основным интересным на мой взгляд библиотекам – JHChainableAnimations & Presentr. В конце подведу итоги

(Слайд 4) Так что же такое анимация? Это видимые изменения элемента, происходящие с течением времени, например, изменение положения, смена цвета, или размеров. Для чего это нужно – ранее использовалось для привлечения внимания, сейчас же – один их способов взаимодействия с пользователем

(Слайд 5) Реализация анимации происходит следующим образом – имеется базовый компонент, от которого наследуются все остальные элементы, которые мы видим на экране нашего телефона. Он имеет положение, стоит сразу оговорить, что на самом деле он имеет два положения – в своей системе координат и в родительской (в том элементе, в котором он расположен). Конечно же, имеет цвет фона и размеры

(Слайд 6) Этот элемент имеет центр анимации или иначе называемый якорем, относительно которого будут происходить видимые преобразования, такие как повороты, сдвиги, масштабирование. Для этих целей мы используем так называемую матрицу преобразования – матрицу, которая дает однозначный новые значения местоположения, размеров элемента при применении к текущим значениям

(Слайд 7) Возьмем небольшой пример – необходимо повернуть квадрат на 45 градусов, уменьшить его в два раза, и передвинуть на 100 точек влево и вправо

(Слайд 8) Этапы для анимации выглядят следующим образом: во-первых, получение матрицы преобразования. В большинстве случаев её значение равно исходной матрице преобразования, при которой преобразования и вовсе нет. Во-вторых, следует работа с матрицей. Ниже показан пример, каким образом это удается достичь.

(Слайд 9) На помощь приходит первая библиотека – JHChainableAnimation – библиотека для совершения связанных анимаций простым и элегантным способом. Среди достоинств – цепочки вызовов, что позволяет писать меньше кода. Расширенный набор функций времени – это такие функции, которые позволяют имитировать поведение реальных объектов

(Слайд 10 и 11) Посмотрите каким элегантным способом можно решить данную проблему.

(Слайд 12) Ещё один важный аспект – модальность. Модальность очень важно сохранять при работе с пользователем через ваш интерфейс. Модальность может быть несколько видов, отлично сочетается с 3DTouch – но главный ареал применения – моменты, когда важно чтобы пользователь не потерял контекст текущей задачи. За это все необходимо заплатить свою цену – разобраться в API, который предоставляет компания, а также грамотно и правильно настроить её выполнение

(Слайд 13) Тут на помощь приходит другая библиотека – Presentr, которая упрощает решение этой задачи

(Слайд 14 & 15) Вот несколько примеров её работы

(Слайд 16) Среди основных достоинств можно отметить следующие: удобство, гибкость настройки предоставленных возможностей, но в то же время ограниченный функционал

(Слайд 17) В данной таблице представлены основные критерия для сравнения двух библиотек (озвучить их)

(Слайд 18) (Озвучить выводы) Большое спасибо за внимание, я готов ответить на Ваши вопросы