Фоновая анимация в Android приложении

Добавлено [Vadik](http://src-code.net/author/vadik/" \o "Posts by Vadik) дата July 5, 2013

[**0**](http://src-code.net/fonovaya-animaciya-v-android-prilozhenii/#respond)

Покадровая анимация (в документации Google – frame-by-frame animation) очень проста. Она представляет собой набор кадров, которые проигрываются в определенном порядке с заданными интервалами. Анимация такого рода реализуется в виде подклассов от AnimationDrawable.

Объекты Ani mationDrawable, будучи подклассами Drawable, могут использоваться во всех тех контекстах, что и любые другие Drawable. Но анимирующий их механизм сам не является частью Drawable. Для осуществления анимации экземпляр AnimationDrawable пользуется внешним механизмом сервисов – реализацией интерфейса Drawable. Call back.

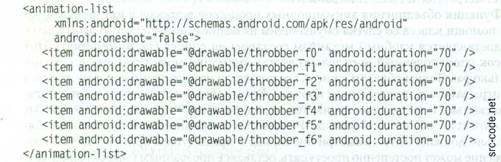
Класс View реализует этот интерфейс и может использоваться для анимирования Ani mationDrawable. К сожалению, в таком случае анимация будет применяться только к тому объекту Drawable, который задан в качестве фона View.

Хорошая новость заключается в том, что этого, вполне возможно, будет достаточно. Фоновая анимация имеет доступ ко всему холсту виджета. Все рисуемое этой фоновой анимацией будет казаться нарисованным под содержимым, создаваемым методом View. onDraw. Поэтому такой способ плохо подходит для реализации полнофункциональных независимых графических объектов, свободно перемещающихся по экрану (спрайтов). И все же, умело пользуясь классом DrawableContan пег (позволяющим одновременно анимировать несколько различных рисунков) и тем, что фон в любой момент можно изменить, вы сможете достаточно многого достичь и без реализации собственного анимационного фреймворка.

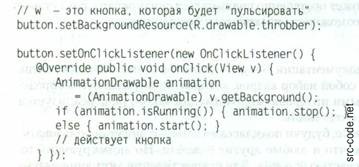
Ani mationDrawable на фоне вида – вот и все, что вам потребуется, чтобы сделать практически что угодно: и указать, что в данный момент происходит какая-то долговременная операция (например, нарисовать крылатые конвертики, летящие от телефона к сотовой вышке), и просто заставить фон кнопки на экране пульсировать.

Пример с пульсирующей кнопкой в виджете 6 – наглядный, и при этом он очень прост в реализации. В примерах 9.14 и 9.15 показан весь необходимый код. Анимация определяется как ресурс, а код применяет анимацию к кнопке. Можно задать Drawable в качестве фона при помощи setBackgroundDrawable или setBackgroundResource.

Пример 9.14. Покадровая анимация (ресурс)



Пример 9.15. Покадровая анимация (код)



Здесь есть пара уловок. Не видно никакого способа, который позволил бы запустить фоновую анимацию из метода Activity. onCreate: строка ((AnimationDrawable) view.getBackground()). start() этого не сделает. Если фон приложения должен анимироваться все то время, пока он виден, то для запуска анимации вам придется схитрить. В данной конкретной реализации для запуска анимации применяется обработчик onClіck. В Вебе встречаются упоминания и о том, что анимацию можно успешно запускать из потока, который ненадолго приостанавливается перед вызовом AnimationDrawablе.start.

Кроме того, если вам приходилось работать с другими фреймворками пользовательских интерфейсов, вы, возможно, привыкли рисовать фон вида в двух первых строках метода onDraw (или его эквивалента). Если сделать так, то рисование пойдет поверх анимации! Вообще, стоит приучить себя использовать setBackground для управления фоном View независимо от того, применяется ли в качестве фона однотонный цвет, градиент, изображение или анимация.

Загружать DrawableAni mation из ресурса – гибкое решение. Можно указать список отрисовываемых ресурсов (любых изображений на ваш выбор), которые составляют анимацию. Если анимация должна быть динамической, то Ani mationDrawable – простейший инструмент для создания динамического отрисовываемого объекта, который можно анимировать на фоне View.