Создаем управляемое движение объектом с помощью перетаскивания

|  |
| --- |
| [Главная](http://learn-android.ru/index.html) » Создаем управляемое движение объектом с помощью перетаскивания |
| В этом уроке мы научимся создавать движение объекта по экрану с полным контролем этого движения прикосновением пальца. У нас на экране будет находится всеми любимое изображение ic\_launcher, которое можно будет передвигать по всему экрану в любую точку.  Начнем с создания нового проекта, выбираем Blank Activity, минимальная версия Android  будет 2,2+.  Создадим нехитрый интерфейс приложения. Открываем файл **activity\_main.xml** и добавим туда элемент **ImageView**:  <**RelativeLayout** xmlns:android="schemas.android.com/apk/res/android"  android:id="@+id/move"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <**ImageView**  android:id="@+id/ImageView"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true"  android:src="@drawable/ic\_launcher" />  </**RelativeLayout**>  Теперь переходим к написанию самой программы в MainActivity.java. Все происходящее будет вертеться вокруг использования двух вещей - **RelativeLayout.LayoutParams**и класса **MotionEvent**. С помощью RelativeLayout.LayoutParams мы запихнем наше изображение в определенные рамки (сделаем его размером 100\*100) и будем изменять положение изображения, отслеживая новые координаты положения изображения на экране.  Класс MotionEvent описывает движение с помощью набора значений координат. Он состоит из определенных команд и числовых значений по осям координат, определяющим положение объекта. Например, когда пользователь впервые прикасается к экрану, вызывается команда **ACTION\_DOWN**, а также набор значений X, Y, включающих координаты точки прикосновения.  Из класса MotionEvent у нас будет использовано 2 команды: **MotionEvent.ACTION\_DOWN** - отслеживает факт прикосновения к объекту и начало движения, здесь будут определяться начальные координаты объекта, далее в работу вступает **MotionEvent.ACTION\_MOVE**, в которой происходит динамическое изменение положения объекта за время жеста прикосновения и содержится точка последней остановки, установленная после завершения действия MotionEvent.ACTION\_DOWN.  Добавляем в файл **MainActivity.java** следующий код:  import **android.app.Activity**;  import **android.os.Bundle**;  import **android.view.MotionEvent**;  import **android.view.View**;  import **android.view.ViewGroup**;  import **android.widget.ImageView**;  import **android.widget.RelativeLayout**;  **public** **class** MainActivity **extends** Activity **implements** View.OnTouchListener {  **private** **ImageView** mImageView;  **private** **ViewGroup** mMoveLayout;  **private** **int** mX;  **private** **int** mY;  **@Override**  **public** **void** onCreate(**Bundle** *savedInstanceState*) {  super.onCreate(savedInstanceState);  setContentView(**R**.layout.activity\_main);  *//Связываемся с нашими объектами, определяя изображение через заданный ViewGroup:*  mMoveLayout = (**ViewGroup**) findViewById(**R**.id.move);  mImageView = (**ImageView**) mMoveLayout.findViewById(**R**.**id.ImageView**);  *//Создаем программно RelativeLayout с параметрами 100\*100:*  **RelativeLayout**.**LayoutParams** layoutParams = new **RelativeLayout**.**LayoutParams**(100, 100);  *//Применяем эти параметры к нашему изображению:*  mImageView.setLayoutParams(layoutParams);  *//И настраиваем ему слушателя (обработчик) прикосновений:*  mImageView.setOnTouchListener(this);  }  *//Обрабатываем прикосновения к объекту:*  **public** **boolean** onTouch(**View** *view*, **MotionEvent** *event*) {  *//Определение координат через getRawX() и getRawY() дает*  *//координаты по отношению к размерам экрана устройства:*  **final** **int** **X** = (**int**) event.getRawX();  **final** **int** **Y** = (**int**) event.getRawY();  switch (event.getAction() & **MotionEvent**.ACTION\_MASK) {  *//ACTION\_DOWN срабатывает при прикосновении к экрану,*  *//здесь определяется начальное стартовое положение объекта:*  case **MotionEvent**.ACTION\_DOWN:  **RelativeLayout**.**LayoutParams** lParams = (**RelativeLayout**.**LayoutParams**) view.getLayoutParams();  mX = **X** - lParams.leftMargin;  mY = **Y** - lParams.topMargin;  break;  *//ACTION\_MOVE обрабатывает случившиеся в процессе прикосновения изменения, здесь*  *//содержится информация о последней точке, где находится объект после окончания действия прикосновения ACTION\_DOWN:*  case **MotionEvent**.ACTION\_MOVE:  **RelativeLayout**.**LayoutParams** layoutParams = (**RelativeLayout**.**LayoutParams**) view  .getLayoutParams();  layoutParams.leftMargin = **X** - mX;  layoutParams.topMargin = **Y** - mY;  layoutParams.rightMargin = -250;  layoutParams.bottomMargin = -250;  view.setLayoutParams(layoutParams);  break;  }  return true;  }  }  Ну а теперь проверим, работает ли программа, получилось ли желаемое управляемое движение. Запускаем приложение и любуемся результатом:  Объект в одном положении  Успешно передвинутый Android на другую сторону  Наш Андруша успешно двигается по всему экрану, надеюсь у вас также :). |