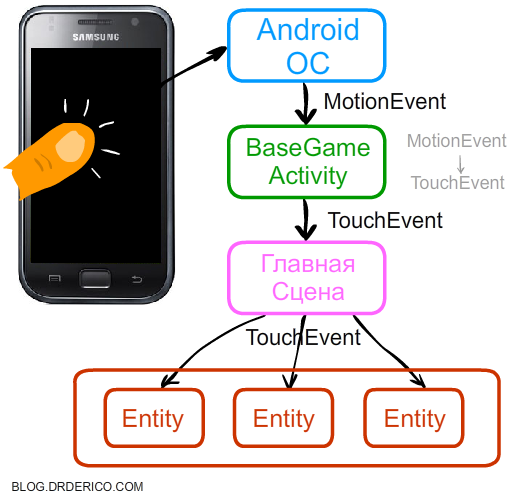
*Обработка касаний экрана*

[AndEngine](http://drderico.ru/category/android/andengine/), [Android](http://drderico.ru/category/android/)  //  [24 коммент.](http://drderico.ru/andengine-obrabotka-kasanij-ekrana/#comments)

AndEngine позволяет с лёгкостью ловить нажатия на экран и на спрайты.

Когда касаешься экрана, Android посылает уведомление (называется **MotionEvent**) в Activity. AndEngine ловит эти события, кэширует и отправляет дальше в Главную Сцену.



Главная Сцена рассылает событие всем объектам, которые добавлены в список слушателей этого события.

Чтобы добавить объект в этот "волшебный" список, нужно в коде Сцены написать:

|  |  |
| --- | --- |
| 1 | registerTouchArea(\_Sprite); |

После этого спрайт будет получать вызов метода **onAreaTouched**.

**ВАЖНО**: он будет его получать, даже если он не добавлен на Сцену.

*MotionEvent и TouchEvent*

Что между ними общего и в чём разница?

**MotionEvent** — объект, создаваемый самой системой Android. Не вдаваясь в глубокие подробности, скажу, что он содержит координаты касания экрана и вид касания. Координаты получаются АБСОЛЮТНЫЕ в пределах физического экрана устройства.

**TouchEvent** — объект, создаваемый движком AndEngine. Он также содержит координаты, вид касания и ещё ссылку на исходный MotionEvent. То есть через TouchEvent можно обратиться к нативному MotionEvent. Координаты у TouchEvent получаются в пределах игровой СЦЕНЫ и зависят от координат Камеры.

Из TouchEvent мы можем узнать вид касания. Он может быть:

* **ACTION\_DOWN** — произошло касание экрана — нажали пальцем на экран. Событие при первом касании экрана вызывается ВСЕГДА первым;
* **ACTION\_UP** — касание закончилось — убрали палец с экрана;
* **ACTION\_MOVE** — начали водить пальцем по экрану после нажатия, либо просто держим палец на месте. ACTION\_MOVE всегда происходит между ACTION\_DOWN и ACTION\_UP;
* **ACTION\_OUTSIDE** — нажатие произошло за пределами Activity (*я ни разу его не использовал*);
* **ACTION\_CANCEL** — произошла отмена нажатия — это не то же самое, что убрали палец с экрана. Я думаю это событие может произойти, когда вы нажали на экран и в это время кто-то вам звонит, и игра прерывается (*тоже пока не использовал*).

*Примеры*

Пример 1. Ловим первое нажатие на объект (спрайт):

/\*\*

\* Создаем кнопку

\*/

\_PauseBtn = new Sprite(755, 11, GfxAssets.\_Pause\_BTN\_TR)

{

    public boolean onAreaTouched(TouchEvent pSceneTouchEvent, float pTouchAreaLocalX, float pTouchAreaLocalY)

    {

        if (pSceneTouchEvent.isActionDown())

        {

            //Пишем код реакции при нажатии на конпку

            ...

            return true; //возвращаем TRUE, если событие отработали

        }

        return false; //возвращаем FALSE, если событие не отработали

    };

};

/\* Добавляем кнопку в слушатели\*/

registerTouchArea(\_PauseBtn);

Пример 2. Ловим нажатие типа КЛИКА на объект (спрайт):

// Флаг для определения того, что коснулись кнопки.

private boolean \_BtnTouched = false;

...

\_Play\_btn = new Sprite(100, 100, GfxAssets.\_Play\_BTN\_TR)

{

    public boolean onAreaTouched(TouchEvent pSceneTouchEvent, float pTouchAreaLocalX, float pTouchAreaLocalY)

    {

        if (pSceneTouchEvent.isActionDown())

        {

            //Касание экрана произошло на кнопке. Это хорошо

            \_BtnTouched = true;

            return true;

        } else

        if (pSceneTouchEvent.isActionUp() && \_BtnTouched)

        {

            //Отпускание пальца произошло на кнопке.

            //Да к тому же и нажатие произошло на кнопке.

            //Значит точно кликнули на кнопке

            \_BtnTouched = false;

            ...

            return true;

        }

        return false;

    };

};

/\*\*

\* Не забываем добавлять кнопку в список слушателей

\*/

registerTouchArea(\_Play\_btn);

...

/\*\*

\* Это - слушатель нажатий для Сцены.

\* При любом касании экрана сбрасываем флажок для кнопки (кнопок),

\* потому что касание экрана может случиться не обязательно

\* на кнопке.

\* Если этого не делать, то могут случаться ложные срабатывания нажатий на кнопки

\*/

@Override

public boolean onSceneTouchEvent(TouchEvent pSceneTouchEvent) {

    if (pSceneTouchEvent.isActionDown())

    {

        /\*\*

         \* Сбрасываем нажатия кнопок

         \*/

        \_BtnTouched = false;

        //\_BtnTouched2 = false;

        //\_BtnTouched3 = false;

    }

    return super.onSceneTouchEvent(pSceneTouchEvent);

}

Переменная \_BtnTouched используется для защиты от ложных нажатий, т.е. исключает такие варианты касаний, как:

* Нажали пальцем на кнопке, а отпустили вне кнопки;
* Нажали пальцем вне кнопки, а отпустили на кнопке.

Другие примеры обработки касаний можете посмотреть в [AndEngineExamples](http://code.google.com/p/andengineexamples/" \t "_blank)...