第一步：构建一个ClearEditText类继承EditText

package com.minxing.view;

import com.zhumingmin.vmsofminxing.R;

import android.content.Context;

import android.graphics.drawable.Drawable;

import android.text.Editable;

import android.text.TextWatcher;

import android.util.AttributeSet;

import android.view.MotionEvent;

import android.view.View;

import android.view.View.OnFocusChangeListener;

import android.view.animation.Animation;

import android.view.animation.CycleInterpolator;

import android.view.animation.TranslateAnimation;

import android.widget.EditText;

public class ClearEditText extends EditText implements OnFocusChangeListener,

        TextWatcher {

    /\*\*

     \* 删除按钮的引用

     \*/

    private Drawable mClearDrawable;

    /\*\*

     \* 控件是否有焦点

     \*/

    private boolean hasFoucs;

    public ClearEditText(Context context) {

        this(context, null);

    }

    public ClearEditText(Context context, AttributeSet attrs) {

        // 这里构造方法也很重要，不加这个很多属性不能再XML里面定义

        this(context, attrs, android.R.attr.editTextStyle);

    }

    public ClearEditText(Context context, AttributeSet attrs, int defStyle) {

        super(context, attrs, defStyle);

        init();

    }

    private void init() {

        // 获取EditText的DrawableRight,假如没有设置我们就使用默认的图片

        mClearDrawable = getCompoundDrawables()[2];

        if (mClearDrawable == null) {

            // throw new

            // NullPointerException("You can add drawableRight attribute in XML");

            mClearDrawable = getResources().getDrawable(

                    R.drawable.delete\_selector);

        }

        mClearDrawable.setBounds(0, 0, mClearDrawable.getIntrinsicWidth(),

                mClearDrawable.getIntrinsicHeight());

        // 默认设置隐藏图标

        setClearIconVisible(false);

        // 设置焦点改变的监听

        setOnFocusChangeListener(this);

        // 设置输入框里面内容发生改变的监听

        addTextChangedListener(this);

    }

    /\*\*

     \* 因为我们不能直接给EditText设置点击事件，所以我们用记住我们按下的位置来模拟点击事件 当我们按下的位置 在 EditText的宽度 -

     \* 图标到控件右边的间距 - 图标的宽度 和 EditText的宽度 - 图标到控件右边的间距之间我们就算点击了图标，竖直方向就没有考虑

     \*/

    @Override

    public boolean onTouchEvent(MotionEvent event) {

        if (event.getAction() == MotionEvent.ACTION\_UP) {

            if (getCompoundDrawables()[2] != null) {

                boolean touchable = event.getX() > (getWidth() - getTotalPaddingRight())

                        && (event.getX() < ((getWidth() - getPaddingRight())));

                if (touchable) {

                    this.setText("");

                }

            }

        }

        return super.onTouchEvent(event);

    }

    /\*\*

     \* 当ClearEditText焦点发生变化的时候，判断里面字符串长度设置清除图标的显示与隐藏

     \*/

    @Override

    public void onFocusChange(View v, boolean hasFocus) {

        this.hasFoucs = hasFocus;

        if (hasFocus) {

            setClearIconVisible(getText().length() > 0);

        } else {

            setClearIconVisible(false);

        }

    }

    /\*\*

     \* 设置清除图标的显示与隐藏，调用setCompoundDrawables为EditText绘制上去

     \*

     \* @param visible

     \*/

    protected void setClearIconVisible(boolean visible) {

        Drawable right = visible ? mClearDrawable : null;

        setCompoundDrawables(getCompoundDrawables()[0],

                getCompoundDrawables()[1], right, getCompoundDrawables()[3]);

    }

    /\*\*

     \* 当输入框里面内容发生变化的时候回调的方法

     \*/

    @Override

    public void onTextChanged(CharSequence s, int start, int count, int after) {

        if (hasFoucs) {

            setClearIconVisible(s.length() > 0);

        }

    }

    @Override

    public void beforeTextChanged(CharSequence s, int start, int count,

            int after) {

    }

    @Override

    public void afterTextChanged(Editable s) {

    }

    /\*\*

     \* 设置晃动动画

     \*/

    public void setShakeAnimation() {

        this.setAnimation(shakeAnimation(5));

    }

    /\*\*

     \* 晃动动画

     \*

     \* @param counts

     \*            1秒钟晃动多少下

     \* @return

     \*/

    public static Animation shakeAnimation(int counts) {

        Animation translateAnimation = new TranslateAnimation(0, 10, 0, 0);

        translateAnimation.setInterpolator(new CycleInterpolator(counts));

        translateAnimation.setDuration(1000);

        return translateAnimation;

    }

}

第二步：在需要引用该自定义控件的地方加入：

  < com.minxing.view.ClearEditText

  android:id= "@+id/TianXieZhangHao"

  android:layout\_height= "wrap\_content"

  android:layout\_marginRight= "10dp"

  android:digits= "0123456789"

  android:hint ="输入用户名"

  android:maxLength= "18"

  android:textColor= "#ffffff"

  android:textSize= "20sp" />

参考文献：

[Android 带清除功能的输入框控件ClearEditText,仿IOS的输入框](http://blog.csdn.net/xiaanming/article/details/11066685)