```
package com.sysfore.SimpleCalculator;

import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnKeyListener;
import android.widget.Button;
import android.widget.EditText;
import java.net.*;

public class SimpleCalculator extends Acticvity {

    private final int MENUITEM_CLOSE = 300;
    private EditText txtCalc=null;
    private Button btnZero=null;
    private Button btnOne=null;
    private Button btnTwo=null;
    private Button btnThree=null;
    private Button btnFour=null;
    private Button btnFive=null;
    private Button btnSix=null;
    private Button btnSeven=null;
    private Button btnEight=null;
    private Button btnNine=null;
    private Button btnPlus=null;
    private Button btnMinus=null;
    private Button btnMultiply=null;
    private Button btnDivide=null;
    private Button btnEquals=null;
    private double num = 0;
    private double memNum = 0;
    private int operator = 1;
    private boolean readyToClear = false;
    private boolean hasChanged = false;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setTheme(android.R.style.Theme_Black);
        setContentView(R.layout.main);

        this.setTitle("CS630Calculator");

        initControls();
        initScreenLayout();
        reset();
    }
    private void initScreenLayout() {

        DisplayMetrics dm = new DisplayMetrics();
        getWindowManager().getDefaultDisplay().getMetrics(dm);

        int height = dm.heightPixels;
        int width = dm.widthPixels;

        if (height < 400 || width < 300) {
            txtCalc.setTextSize(20);
        }

        if (width < 300) {
            btnDivide.setTextSize(18);
            btnPlus.setTextSize(18);
            btnMinus.setTextSize(18);
            btnMultiply.setTextSize(18);
            btnEquals.setTextSize(18);
            btnNine.setTextSize(18);
            btnEight.setTextSize(18);
            btnSeven.setTextSize(18);
            btnSix.setTextSize(18);
            btnFive.setTextSize(18);
            btnFour.setTextSize(18);
            btnThree.setTextSize(18);
            btnTwo.setTextSize(18);
            btnOne.setTextSize(18);
            btnZero.setTextSize(18);

        }

        btnZero.setTextColor(Color.BLACK);
        btnOne.setTextColor(Color.BLACK);
        btnTwo.setTextColor(Color.BLACK);
        btnThree.setTextColor(Color.BLACK);
        btnFour.setTextColor(Color.BLACK);
        btnFive.setTextColor(Color.BLACK);
        btnSix.setTextColor(Color.BLACK);
        btnSeven.setTextColor(Color.BLACK);
        btnEight.setTextColor(Color.BLACK);
        btnNine.setTextColor(Color.BLACK);

    }

    private void initControls() {
        txtCalc = (EditText) findViewById(R.id.txtCalc);
        btnZero = (Button) findViewById(R.id.btnZero);
        btnOne = (Button) findViewById(R.id.btnOne);
        btnTwo = (Button) findViewById(R.id.btnTwo);
        btnThree = (Button) findViewById(R.id.btnThree);
        btnFour = (Button) findViewById(R.id.btnFour);
        btnFive = (Button) findViewById(R.id.btnFive);
        btnSix = (Button) findViewById(R.id.btnSix);
        btnSeven = (Button) findViewById(R.id.btnSeven);
        btnEight = (Button) findViewById(R.id.btnEight);
        btnNine = (Button)findViewById(R.id.btnNine);
        btnPlus = (Button) findViewById(R.id.btnPlus);
        btnMinus = (Button) findViewById(R.id.btnMinus);
        btnMultiply = (Button) findViewById(R.id.btnMultiply);
        btnDivide = (Button) findViewById(R.id.btnDivide);
        btnEquals = (Button) findViewById(R.id.btnEquals);


        txtCalc.setOnKeyListener(new OnKeyListener() {
            public boolean onKey(View v, int i, android.view.KeyEvent e) {
                if (e.getAction() == KeyEvent.ACTION_DOWN) {
                    int keyCode = e.getKeyCode();

                    switch (keyCode) {
                    case KeyEvent.KEYCODE_0:
                        handleNumber(0);
                        break;

                    case KeyEvent.KEYCODE_1:
                        handleNumber(1);
                        break;

                    case KeyEvent.KEYCODE_2:
                        handleNumber(2);
                        break;

                    case KeyEvent.KEYCODE_3:
                        handleNumber(3);
                        break;

                    case KeyEvent.KEYCODE_4:
                        handleNumber(4);
                        break;

                    case KeyEvent.KEYCODE_5:
                        handleNumber(5);
                        break;

                    case KeyEvent.KEYCODE_6:
                        handleNumber(6);
                        break;

                    case KeyEvent.KEYCODE_7:
                        handleNumber(7);
                        break;

                    case KeyEvent.KEYCODE_8:
                        handleNumber(8);
                        break;

                    case KeyEvent.KEYCODE_9:
                        handleNumber(9);
                        break;

                    case 43:
                        handleEquals(1);
                        break;

                    case KeyEvent.KEYCODE_EQUALS:
                        handleEquals(0);
                        break;

                    case KeyEvent.KEYCODE_MINUS:
                        handleEquals(2);
                        break;

                    case KeyEvent.KEYCODE_PERIOD:
                        handleDecimal();
                        break;

                    case KeyEvent.KEYCODE_C:
                        reset();
                        break;

                    case KeyEvent.KEYCODE_SLASH:
                        handleEquals(4);
                        break;

                    case KeyEvent.KEYCODE_DPAD_DOWN:
                        return false;
                    }
                }

                return true;
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        menu.add(0,1, MENUITEM_CLOSE, "Close");

        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
        case MENUITEM_CLOSE:
            finish();
            break;
        }

        return super.onOptionsItemSelected(item);
    }

    private void handleEquals(int newOperator) {
        if (hasChanged) {
            switch (operator) {
            case 1:
                num = num + Double.parseDouble(txtCalc.getText().toString());
                break;
            case 2:
                num = num - Double.parseDouble(txtCalc.getText().toString());
                break;
            case 3:
                num = num * Double.parseDouble(txtCalc.getText().toString());
                break;
            case 4:
                num = num / Double.parseDouble(txtCalc.getText().toString());
                break;
            }

            String message=Double.toString(num);
            int server_port = 9999;
            DatagramSocket s = new DatagramSocket();
            InetAddress local = InetAddress.getByName("192.168.0.104");
            int msg_length=message.length();
            byte[] message = message.getBytes();
            DatagramPacket p = new DatagramPacket(message, msg_length,local,server_port);
            s.send(p);

            String txt;
            byte[] message2 = new byte[1500];
            DatagramPacket p2 = new DatagramPacket(message2, message2.length);
            DatagramSocket s2 = new DatagramSocket(server_port);
            s.receive(p2);
            txt = new String(message2, 0, p2.getLength());
            s2.close();

            txtCalc.setText(txt);
            txtCalc.setSelection(txt.length());

            readyToClear = true;
            hasChanged = false;
        }

        operator = newOperator;
    }

    private void handleNumber(int num) {
        if (operator == 0)
            reset();

        String txt = txtCalc.getText().toString();
        if (readyToClear) {
            txt = "";
            readyToClear = false;
        } else if (txt.equals("0"))
            txt = "";

        txt = txt + Integer.toString(num);

        txtCalc.setText(txt);
        txtCalc.setSelection(txt.length());

        hasChanged = true;
    }

    private void setValue(String value) {
        if (operator == 0)
            reset();

        if (readyToClear) {
            readyToClear = false;
        }

        txtCalc.setText(value);
        txtCalc.setSelection(value.length());

        hasChanged = true;
    }

    private void handleDecimal() {
        if (operator == 0)
            reset();

        if (readyToClear) {
            txtCalc.setText("0.");
            txtCalc.setSelection(2);
            readyToClear = false;
            hasChanged = true;
        } else {
            String txt = txtCalc.getText().toString();

            if (!txt.contains(".")) {
                txtCalc.append(".");
                hasChanged = true;
            }
        }
    }

    private void handleBackspace() {
        if (!readyToClear) {
            String txt = txtCalc.getText().toString();
            if (txt.length() > 0) {
                txt = txt.substring(0, txt.length() - 1);
                if (txt.equals(""))
                    txt = "0";
                txtCalc.setText(txt);
                txtCalc.setSelection(txt.length());
            }
        }
    }

    private void handlePlusMinus() {
        if (!readyToClear) {
            String txt = txtCalc.getText().toString();
            if (!txt.equals("0")) {
                if (txt.charAt(0) == '-')
                    txt = txt.substring(1, txt.length());
                else
                    txt = "-" + txt;

                txtCalc.setText(txt);
                txtCalc.setSelection(txt.length());
            }
        }
    }

    private void reset() {
        num = 0;
        txtCalc.setText("0");
        txtCalc.setSelection(1);
        operator = 1;
    }
}
```