插入 Texview

- ID 為 textView1

插入 button

- Text 為 Open camera
- On Click 為 btn_test1

插入 button

- Text 為 close

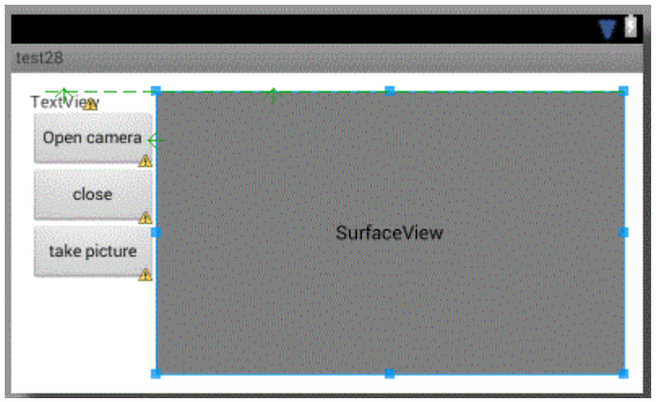
On Click 為 btn_test2

插入 button

- Text 為 take picture
- On Click 為 btn_test2

插入 SurfaceView

- ID 為 surfaceView1



需要設權限 AndroidManifest.xml

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

全域變數

```
SurfaceHolder sfvhoder;
Camera cam1;
SurfaceView sfv1;
```

OnCreate

```
//setRequestedOrientation(0);
sfv1=(SurfaceView)findViewById(R.id.surfaceView1);
sfvhoder=sfv1.getHolder();
sfvhoder.setType(sfvhoder.SURFACE_TYPE_PUSH_BUFFERS);
```

```
        //sfvhoder.addCallback( this);
```

MainActivity.java 加入以下 code,按 ctrl+shift+o 自動補 import

```
    public void btn_test1(View v) {
        Camera.Parameters cam_param;
        TextView tv1;
        tv1=(TextView)this.findViewById(R.id.textView1);
        String st1;
        // TODO Auto-generated method stub

        try{
            cam1 = Camera.open();
            cam_param=cam1.getParameters();
//          cam_param.setPreviewSize(320, 240);

            List psizelist = cam_param.getSupportedPictureSizes();
            if (null != psizelist && 0 < psizelist.size()) {
                int heights[] = new int[psizelist.size()];
                Map<Integer, Integer> map = new HashMap<Integer, Integer>();
                st1=Integer.toString( psizelist.size());
                for (int i = 0; i < psizelist.size(); i++) {
                    Size size = (Size) psizelist.get(i);
                    int sizehieght = size.height;
                    int sizewidth = size.width;
                    heights[i] = sizehieght;
                    map.put(sizehieght, sizewidth);
                    st1=st1+' '+Integer.toString(size.width)+'x'+Integer.toString(size.height);
                }
                Arrays.sort(heights);// set mini size
                cam_param.setPictureSize(map.get(heights[0]),heights[0]);
                //st1=Integer.toString(heights[0])+'x'+Integer.toString(map.get(heights[0]));
                tv1.setText(st1);
            }
            //cam1.setDisplayOrientation(90);

            cam1.setParameters(cam_param);
            cam1.setPreviewDisplay(sfvhoder);
```

```
            cam1.startPreview();
        }catch (Exception e)
        {
            Toast.makeText(this, "preview fail", 1000);
            e.printStackTrace();
        }
}
public void btn_test2(View v) {
        cam1.stopPreview();
        cam1.release();

}

PictureCallback jpeg =new PictureCallback(){

public void onPictureTaken(byte[] data, Camera camera) {

    Bitmap bmp=BitmapFactory.decodeByteArray(data, 0, data.length);
    FileOutputStream fop;
    try {
        fop=new FileOutputStream("/sdcard/test.jpg");
        bmp.compress(Bitmap.CompressFormat.JPEG, 100, fop);
        fop.close();
        System.out.println("take picture ok");
    } catch (FileNotFoundException e) {
        e.printStackTrace();
        System.out.println("FileNotFoundException");
    } catch (IOException e) {
        e.printStackTrace();
        System.out.println("IOException");
    }
        camera.startPreview();
    }
};
public void btn_test3(View v) {
```

```
        cam1.takePicture(null, null, jpeg);
    }
```