



ANDROID STUDIO -SURFACE & THREAD-

SOULSEEK

SURFACE

- onDraw()와 invalidate()가 필요 없다
- 독립적인 그리기 작업을 할 수 있는 공간
- Thread 설치가 필요하다

SurfaceView

Surface

고속메모리(버퍼)이며 lockCanvas 메소드에 의해서 사용자의 문자나 이미지가 그려지게 되는 곳이다.

SurfaceHolder

SurfaceHolder는 서피스뷰에서 전용으로 사용되는 인터페이스이다. lockCanvas()와 unlockCanvasAndPost()를 가지고 있다.

- **unlockCanvasAndPost()** : 서피스에 픽셀(그림이나 그림자)을 표시하는 것으로 끝내고 서피스에 있는 화면을 실제화면에 복사한다.
- **lockCanvas()** : 서피스에 픽셀(그림이나 그림자)을 그린다.

THREAD

- 응용프로그램이 구동되고 있는 작업공간.
- **Main Thread**에서 모든 처리를 기다리면서 넘어가기 힘들다면 나누어라.
- **Surface**와 연결하여 그리는 부분과 인터페이스 동작부분을 나누어 줄 수 있다.

Thread에 SurfaceView연결하기

1. 서피스뷰를 상속받고 **SurfaceHolder.Callback**을 구현하는 클래스를 만든다.
 - **public class MySurface extends Surface implements SurfaceHolder.Callback**
2. 만든 **Surface** 클래스안에 **Thread**를 상속받은 내부 클래스를 만든다.
 - **Class MyThread extends Thread**
3. 만든 **Surface** 클래스 멤버 필드로 **Thread** 객체 및 **Surface Holder**를 선언한다.
 - **MyThread myThread;** // 쓰레드 객체 선언
 - **SurfaceHolder mHolder;** //서피스 홀더 객체 선언
4. 만든 **Surface** 클래스의 생성자 안에서 **Thread**를 생성한다. 이때 생성자 인수로 **SurfaceHolder**객체 및 **Context**객체를 넘겨준다.
 - **myThread = new Thread(holder, context);**
5. **surfaceCreated()** 메소드 안에 **start()**메소드를 넣는다.(쓰레드를 가동한다)
 - **myThread.start();**
6. 만든 **Thread** 클래스 안에 **run()** 메소드를 만들고 안에 다음 내용을 적는다.

```
public void run() {
    Canvas canvas = null;
    while(true) {
        canvas = mHolder.lockCanvas();

        try {
            synchronized(mHolder) \{
                drawEverything(canvas);
            }
        }
        finally {
            if(canvas != null) {
                mHolder.unlockCanvasAndPost(canvas);
            }
        }
    }
}
```

SURFACE연결하기

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

```
    <com.soulseek.surfacegame.MySurface  
        android:id="@+id/mySurface"  
        android:layout_width="fill_parent"  
        android:layout_height="fill_parent"  
    />
```

```
</LinearLayout>
```



