

# Android Game Framework 使用說明

## 將多個動作製成物件

練習 4：將貼圖與移動等動作製作成一個物件。

1. 請先完成練習 3A 或 3B。
2. 新增一個 Class 名為 Practice，並且實作 GameObject 的 move()、show() 和 release()

```
public class Practice implements GameObject {
    private MovingBitmap mPractice;
    private int px, py;

    public Practice(){
        mPractice = new MovingBitmap(R.drawable.ntut_csie);
        px = 250;
        py = 50;
    }

    public void initialize(){
        mPractice.setLocation(px, py);
    }

    @Override
    public void release() {
        mPractice.release();
        mPractice = null;
    }

    @Override
    public void move() {
        if(px < 350) {
            px += 5;
        } else {
            py += 5;
        }
        mPractice.setLocation(px, py);
    }

    @Override
    public void show() {
        mPractice.show();
    }
}
```

遊戲框架要求 show() 只負責顯示圖像，不應該包含任何變更物件內容

的程式邏輯，因此 `move()` 負責改變座標，而 `show()` 負責顯示圖像。

3. 在 `StateRun.java` 中宣告物件 `Practice`：

```
public class StateRun extends GameState {  
    private Practice mPractice;
```

4. 在 `initialize()` 中初始物件 `Practice`

```
@Override  
public void initialize(Map<String, Object> data) {  
    mPractice = new Practice();  
    mPractice.initialize();
```

5. 接著在 `show()`、`move()` 中加入物件 `Practice` 的副程式：

```
@Override  
public void move() {  
    ...  
    mPractice.move();  
}  
  
@Override  
public void show() {  
    ...  
    mPractice.show();
```

6. 在每個 `State` 結束時，釋放記憶體空間：

```
@Override  
public void release() {  
    mPractice.release();  
    mPractice = null;
```

7. 執行。