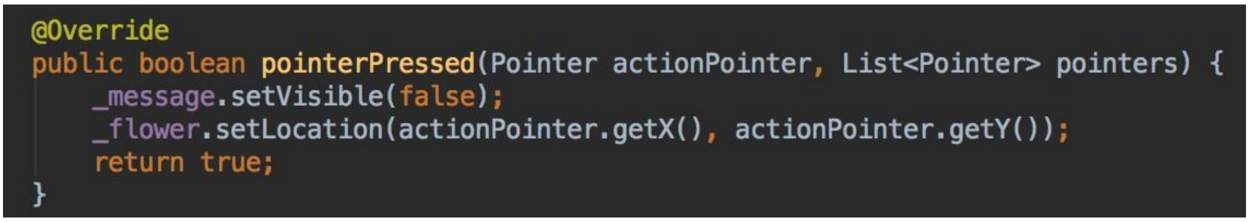
Android Game Framework 使用說明

單點及多點觸控

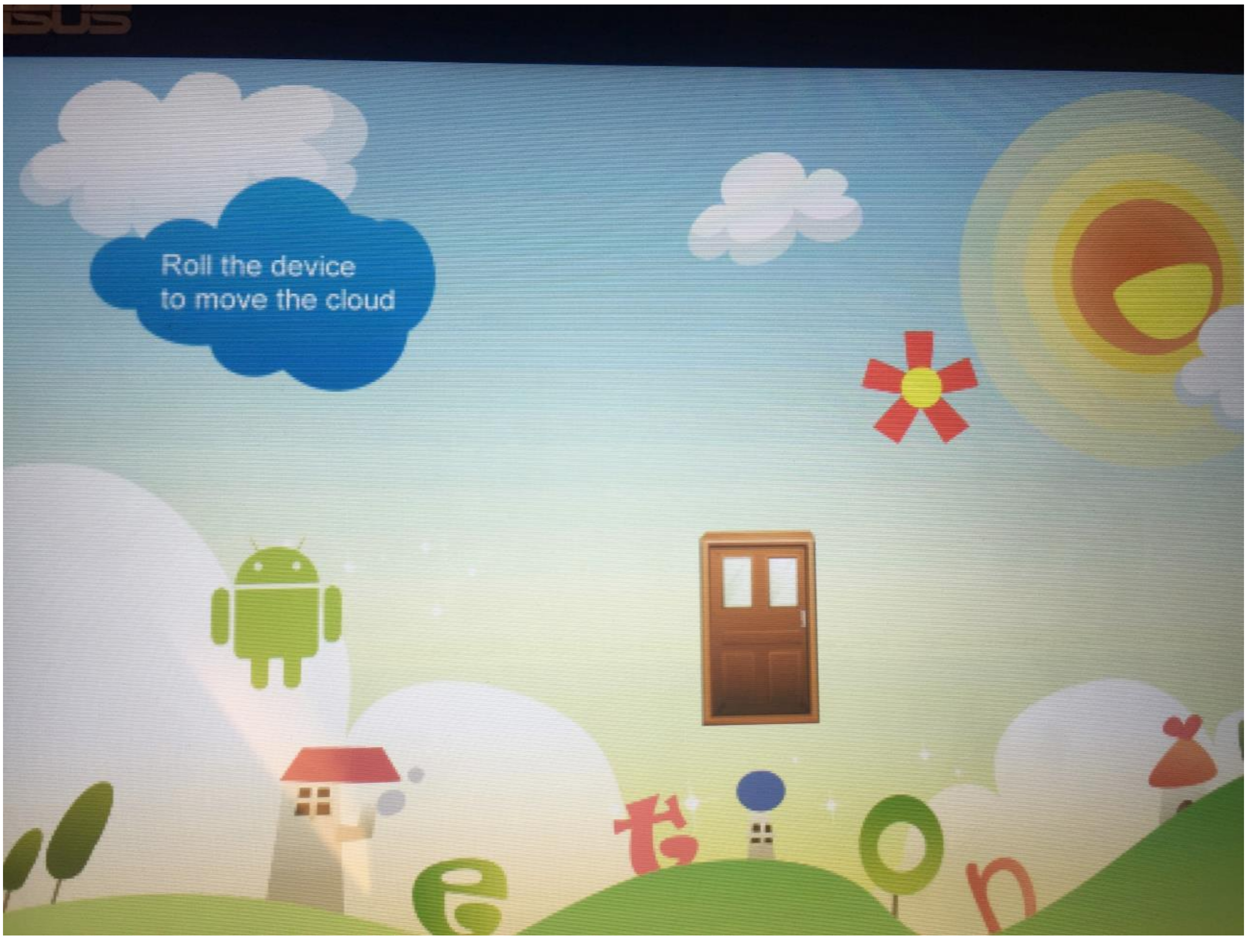
練習 6A ： 單點觸控

1. 在 StateRun 中 pointerPressed 改成以下程式碼



```
@Override
public boolean pointerPressed(Pointer actionPointer, List<Pointer> pointers) {
    _message.setVisible(false);
    _flower.setLocation(actionPointer.getX(), actionPointer.getY());
    return true;
}
```

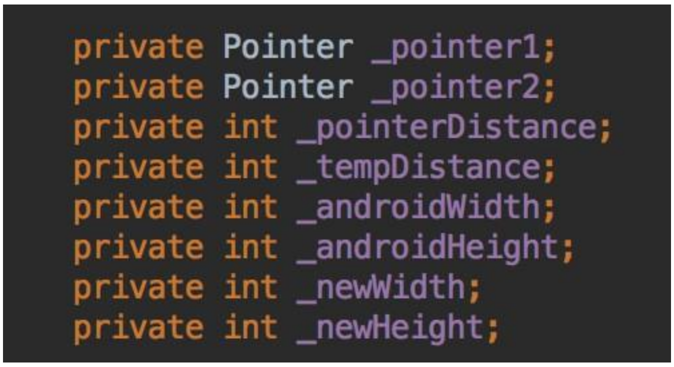
2. 執行，執行結果如下(隨意觸碰螢幕)



花朵會出現在觸碰的位置上

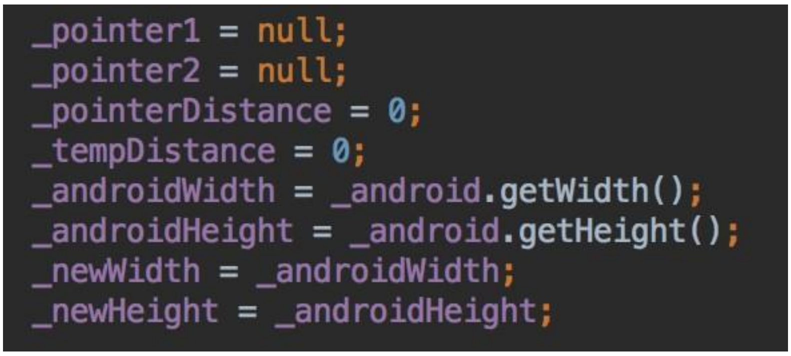
練習 6B : 多點觸控

1. 在 StateRun 中增加以下變數



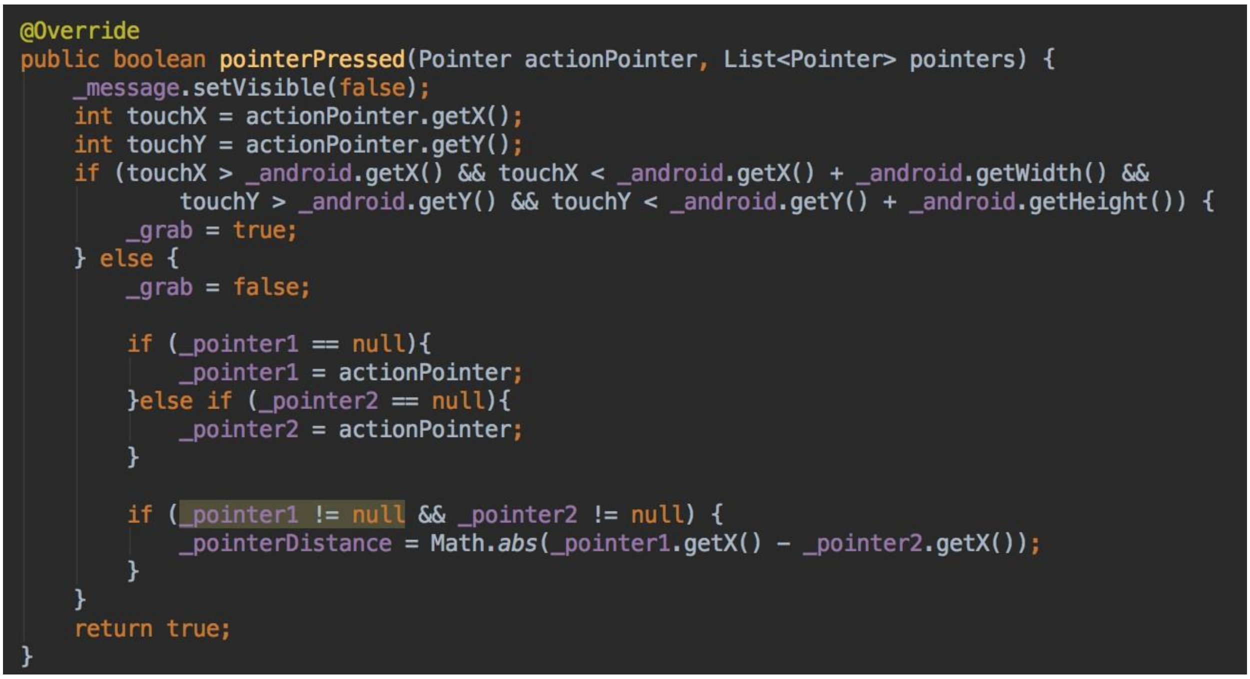
```
private Pointer _pointer1;
private Pointer _pointer2;
private int _pointerDistance;
private int _tempDistance;
private int _androidWidth;
private int _androidHeight;
private int _newWidth;
private int _newHeight;
```

2. initialize 加入以下程式碼使數值初始化



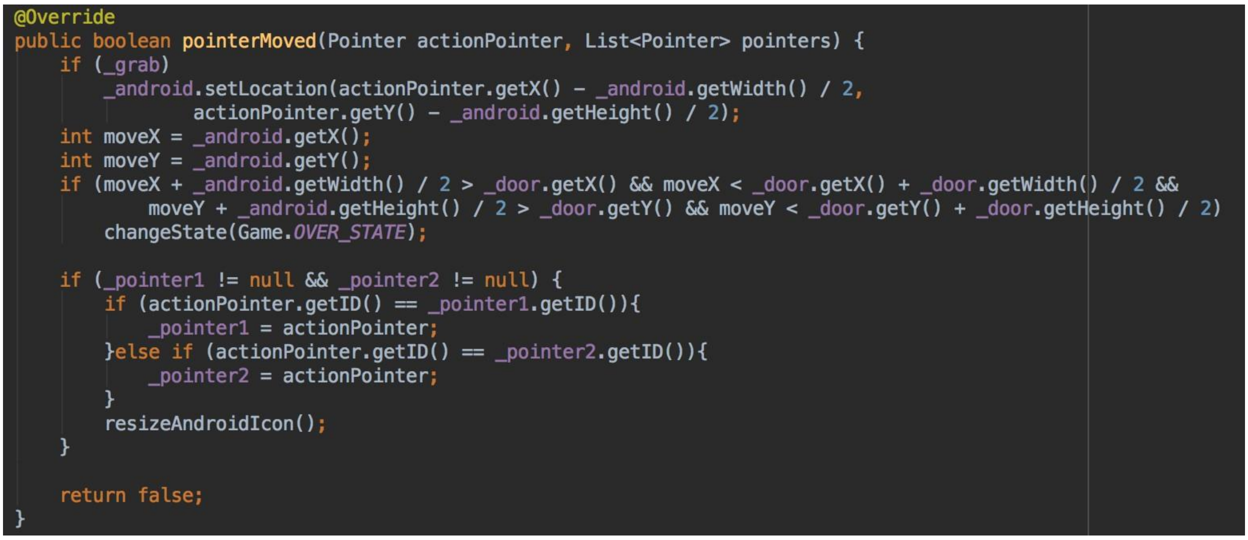
```
_pointer1 = null;
_pointer2 = null;
_pointerDistance = 0;
_tempDistance = 0;
_androidWidth = _android.getWidth();
_androidHeight = _android.getHeight();
_newWidth = _androidWidth;
_newHeight = _androidHeight;
```

3. pointerPressed 修改如下，當螢幕被觸控時會執行此 function



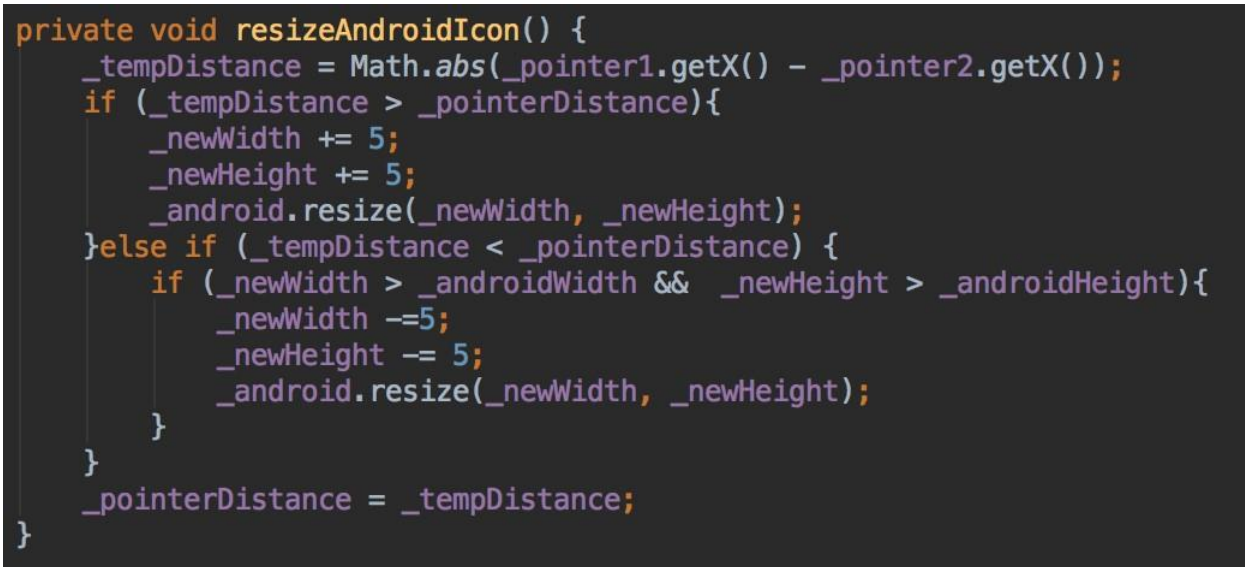
```
@Override
public boolean pointerPressed(Pointer actionPointer, List<Pointer> pointers) {
    _message.setVisible(false);
    int touchX = actionPointer.getX();
    int touchY = actionPointer.getY();
    if (touchX > _android.getX() && touchX < _android.getX() + _android.getWidth() &&
            touchY > _android.getY() && touchY < _android.getY() + _android.getHeight()) {
        _grab = true;
    } else {
        _grab = false;

        if (_pointer1 == null){
            _pointer1 = actionPointer;
        }else if (_pointer2 == null){
            _pointer2 = actionPointer;
        }

        if (_pointer1 != null && _pointer2 != null) {
            _pointerDistance = Math.abs(_pointer1.getX() - _pointer2.getX());
        }
    }
    return true;
}
```

4. pointerMove 修改如下，當手指在螢幕上移動或按著不放時執行此 function



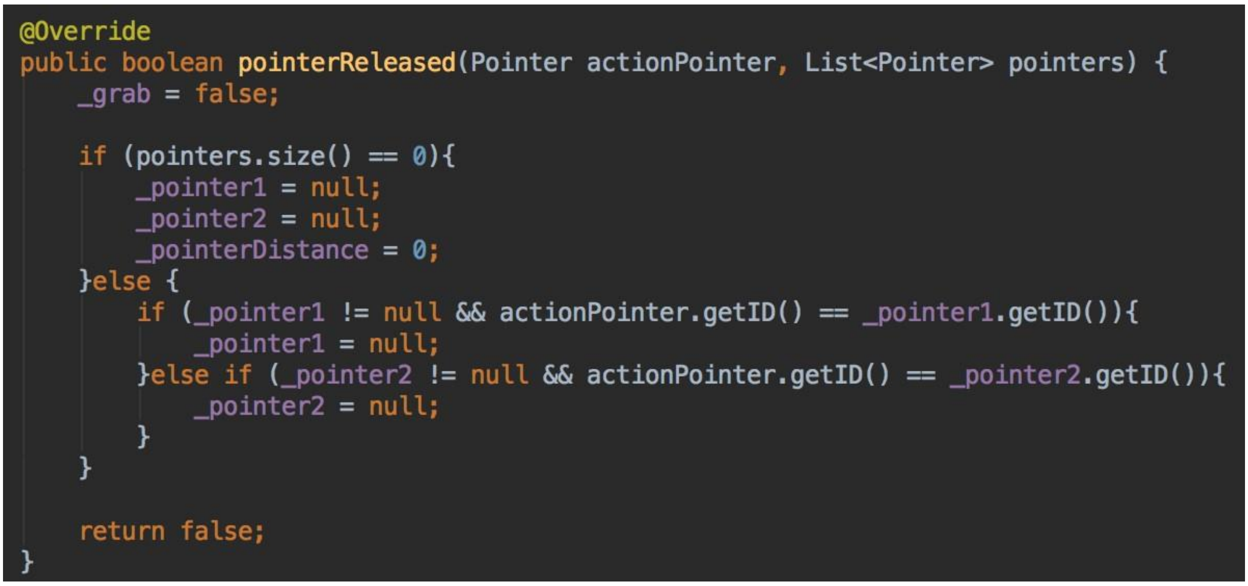
```
@Override
public boolean pointerMoved(Pointer actionPointer, List<Pointer> pointers) {
    if (_grab)
        _android.setLocation(actionPointer.getX() - _android.getWidth() / 2,
                actionPointer.getY() - _android.getHeight() / 2);
    int moveX = _android.getX();
    int moveY = _android.getY();
    if (moveX + _android.getWidth() / 2 > _door.getX() && moveX < _door.getX() + _door.getWidth() / 2 &&
            moveY + _android.getHeight() / 2 > _door.getY() && moveY < _door.getY() + _door.getHeight() / 2)
        changeState(Game.OVER_STATE);

    if (_pointer1 != null && _pointer2 != null) {
        if (actionPointer.getID() == _pointer1.getID()){
            _pointer1 = actionPointer;
        }else if (actionPointer.getID() == _pointer2.getID()){
            _pointer2 = actionPointer;
        }
        resizeAndroidIcon();
    }

    return false;
}
```

5. 新增加此 function



```
private void resizeAndroidIcon() {
    _tempDistance = Math.abs(_pointer1.getX() - _pointer2.getX());
    if (_tempDistance > _pointerDistance){
        _newWidth += 5;
        _newHeight += 5;
        _android.resize(_newWidth, _newHeight);
    }else if (_tempDistance < _pointerDistance) {
        if (_newWidth > _androidWidth &&  _newHeight > _androidHeight){
            _newWidth -=5;
            _newHeight -= 5;
            _android.resize(_newWidth, _newHeight);
        }
    }
    _pointerDistance = _tempDistance;
}
```

6. pointerReleased 修改如下，當手指離開螢幕時執行此 function



```
@Override
public boolean pointerReleased(Pointer actionPointer, List<Pointer> pointers) {
    _grab = false;

    if (pointers.size() == 0){
        _pointer1 = null;
        _pointer2 = null;
        _pointerDistance = 0;
    }else {
        if (_pointer1 != null && actionPointer.getID() == _pointer1.getID()){
            _pointer1 = null;
        }else if (_pointer2 != null && actionPointer.getID() == _pointer2.getID()){
            _pointer2 = null;
        }
    }

    return false;
}
```

7. 執行結果(手指不碰觸到 Android icon 情況下，2 隻手指做縮放，Android icon 會縮放）