

導讀：

創作一個手機App，聽起來不容易，如果真的做出來，想必十分有成就感。

其實，現階段就有一個低門檻的手機App開發工具，即便你對程式設計苦手，依然可以玩的很開心，這套工具就是由 M.I.T（麻省理工學院）開發的 App Inventor。

我們來看看究竟如何使用 App Inventor 開發App吧！

App Inventor 介紹及優劣分析

簡介

App Inventor 是由麻省理工學院 (MIT) 所開發的程式設計軟體，因為 App Inventor 具有免費、簡單的特色，所以相當合適非程式專業領域的人們投入開發。

App Inventor 在開發程式時，以拖曳與拼接積木方塊的方式來撰寫程式。利用 App Inventor 所撰寫出來的程式，可以輕鬆的在 Android 手機上安裝執行。而且，App Inventor 提供雲端平台及專案存放的空間，方便於任意不同電腦繼續專案的開發，也是讓許多人選擇它進行手機 App開發的原因。

要使用 App Inventor 開發程式，必須先[連上 App Inventor 網站](#)，並登入帳號才可以開始程式開發。

優缺點分析

App Inventor	
優點	提供繁體中文介面 函式庫的精簡積木化 雲端保存及易於協作
缺點	受限於積木定義，有些功能無法輕鬆實現 一定要連網路才能開發 在Android平台上能很好執行，但並不適用於iOS平台

第一支程式

新建專案

登入帳號後，在「我的專案」頁面，找到「新增專案」，點擊並建立專案：



專案開啟後，預設介面如下：



新增元件 - 按鈕、標籤

接著從左方元件面板處，新增一個按鈕、一個標籤：

改變按鈕的「屬性」-「寬度」規則，觀察其元件變化：

修改按鈕文字：

修改標籤文字：

修改標籤「屬性」-「可見性」：

測試專案

在 App Inventor 中，測試程式的方法有 4 種，分別如下：

- AI Companion: 在 Android 手機中，開啟 Play Store 搜尋 MIT AI2 Companion，再透過 App 來執行程式。
- 模擬器: 在電腦中安裝模擬器，再以模擬器方式執行測試。
- USB: 手機透過 USB 線與電腦連線，即可從手機執行程式。
- 打包apk: 下載 apk 檔案至手機中並安裝，會直接將程式裝於手機儲存空間中，以此方法執行程式。

其中使用 MIT AI2 Companion 的測試方式，速度最快，且能在修改程式碼後立即更新執行畫面。

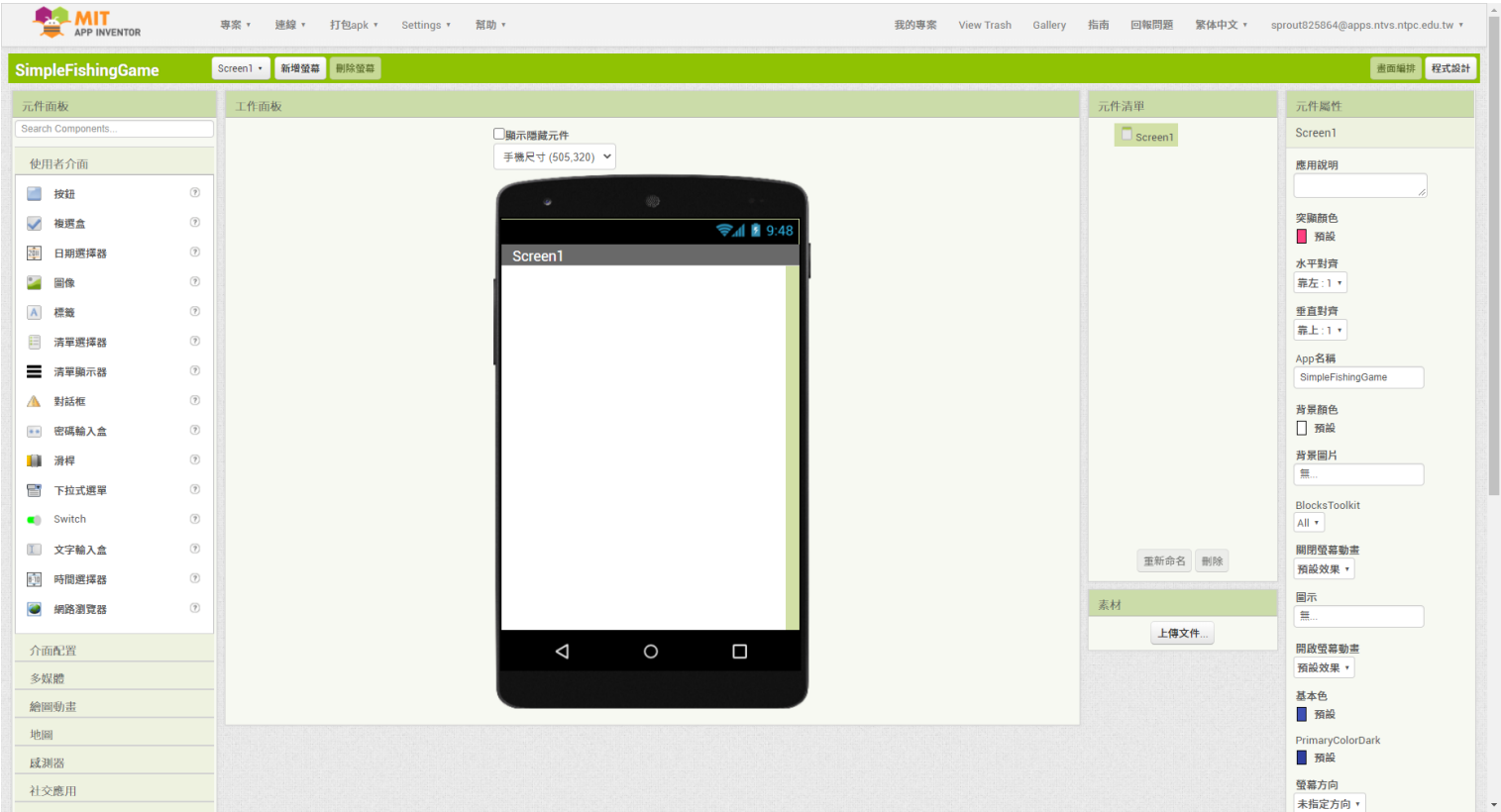
因此，以下將針對此方法進行示範。

試做個小遊戲

由於使用素材有版權疑慮，因此僅作為學習交流使用，勿作商業用途。

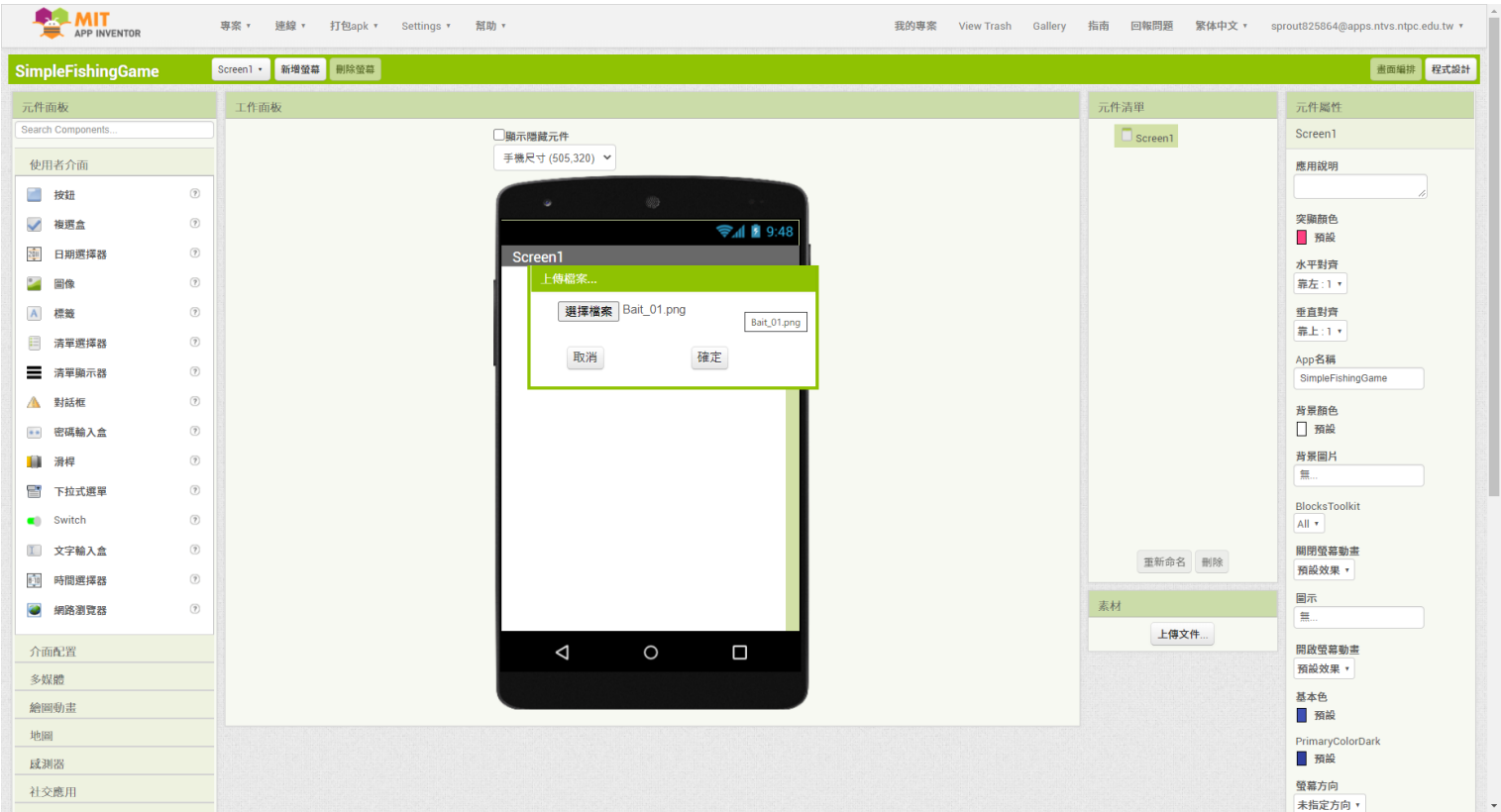
創建一個新專案

專案可隨意取名

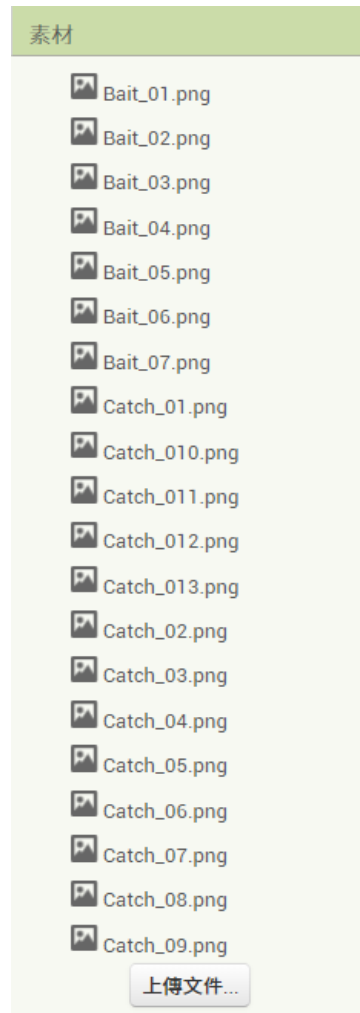


動畫作法

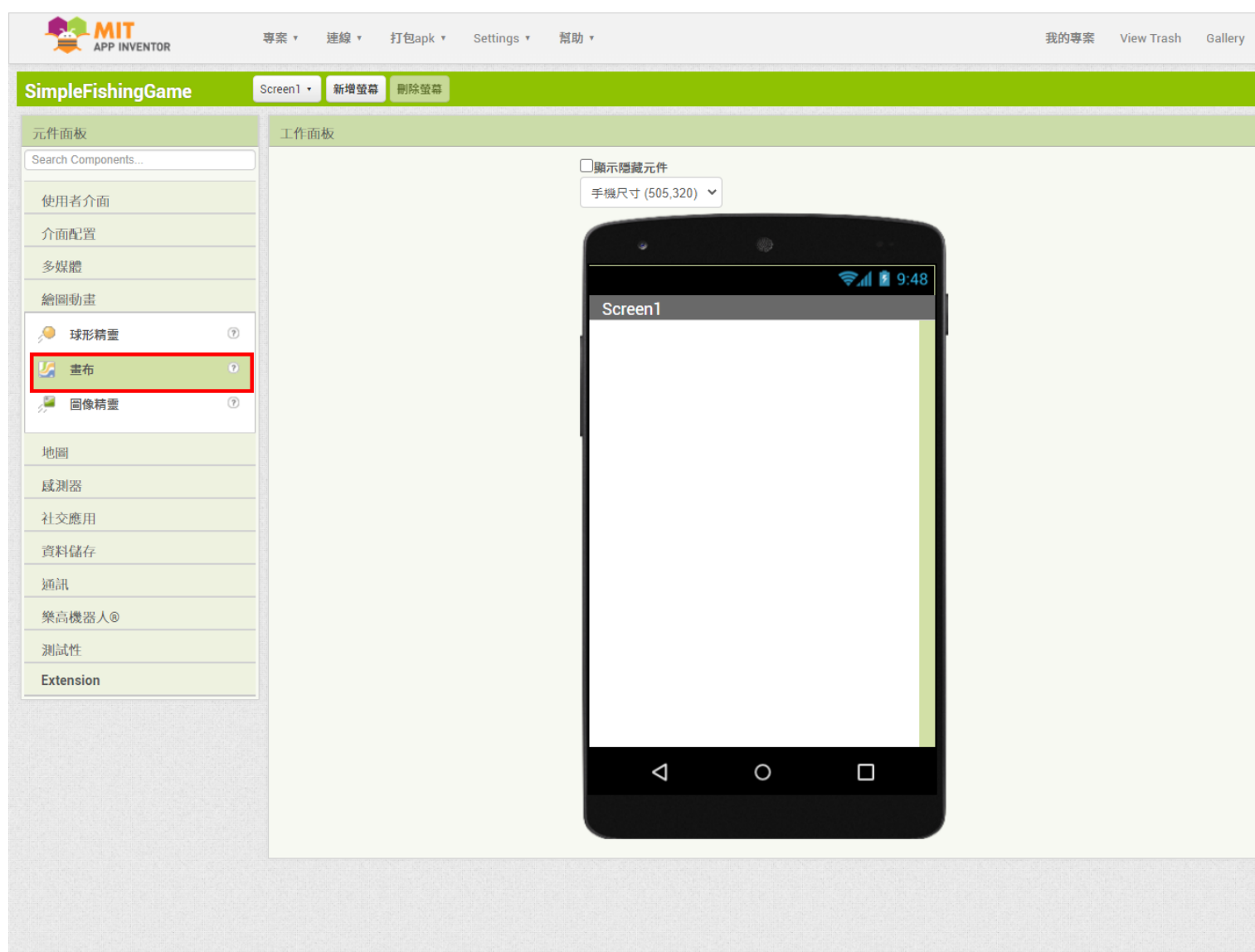
先將素材一一上傳完畢(需要耐心)

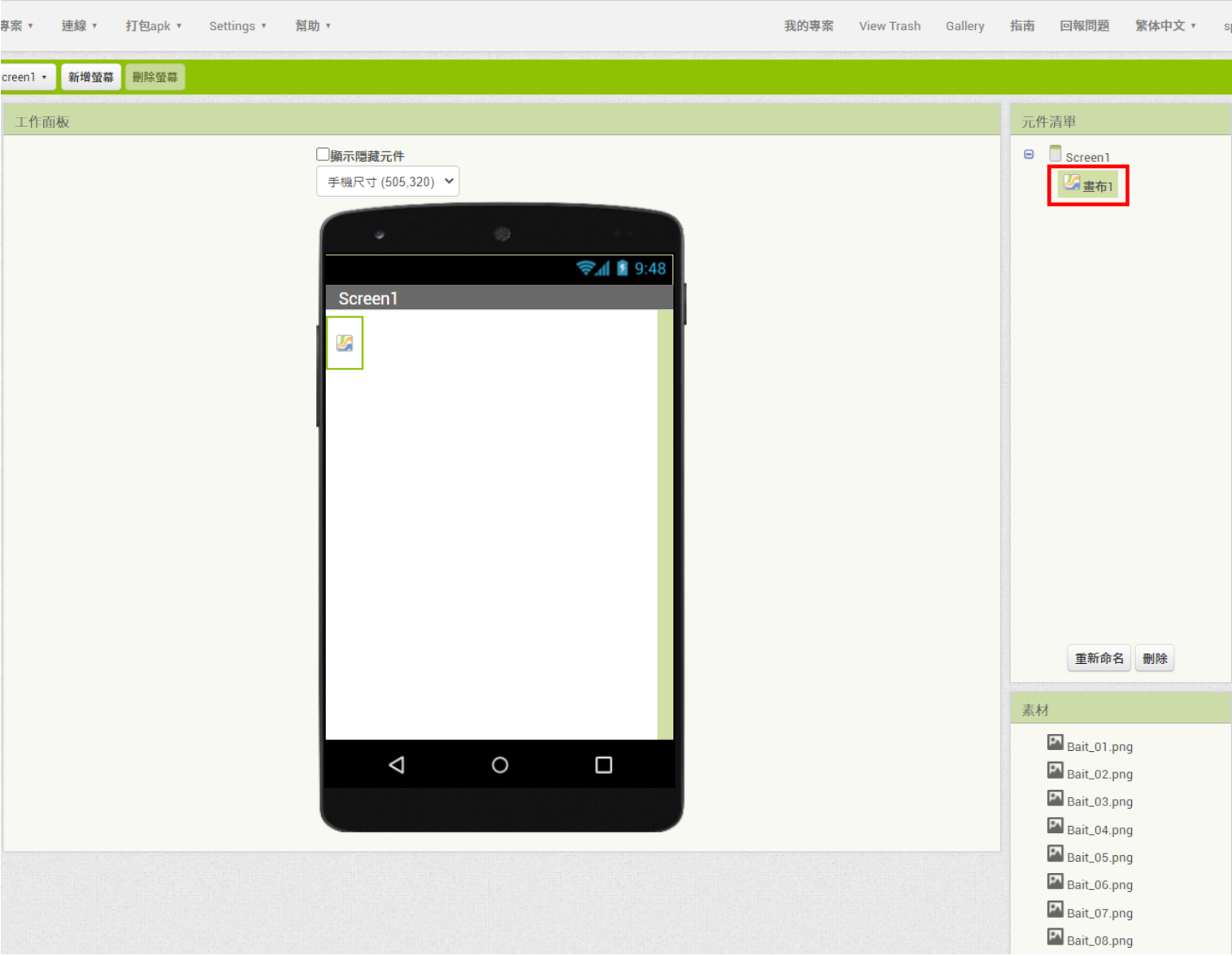


上傳挺費時的，所以如果自覺不是有耐性的人，就選擇別的任務做吧！

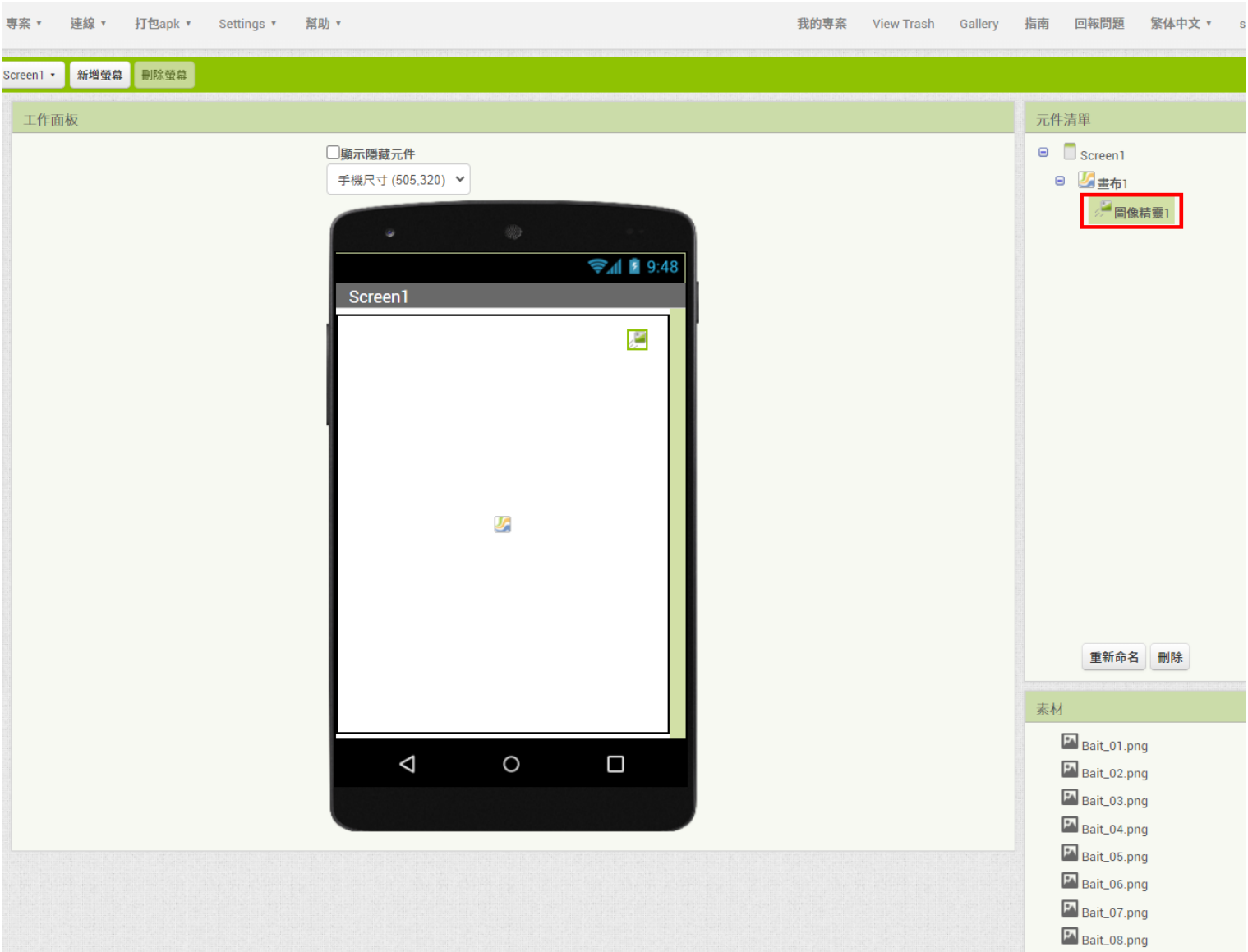
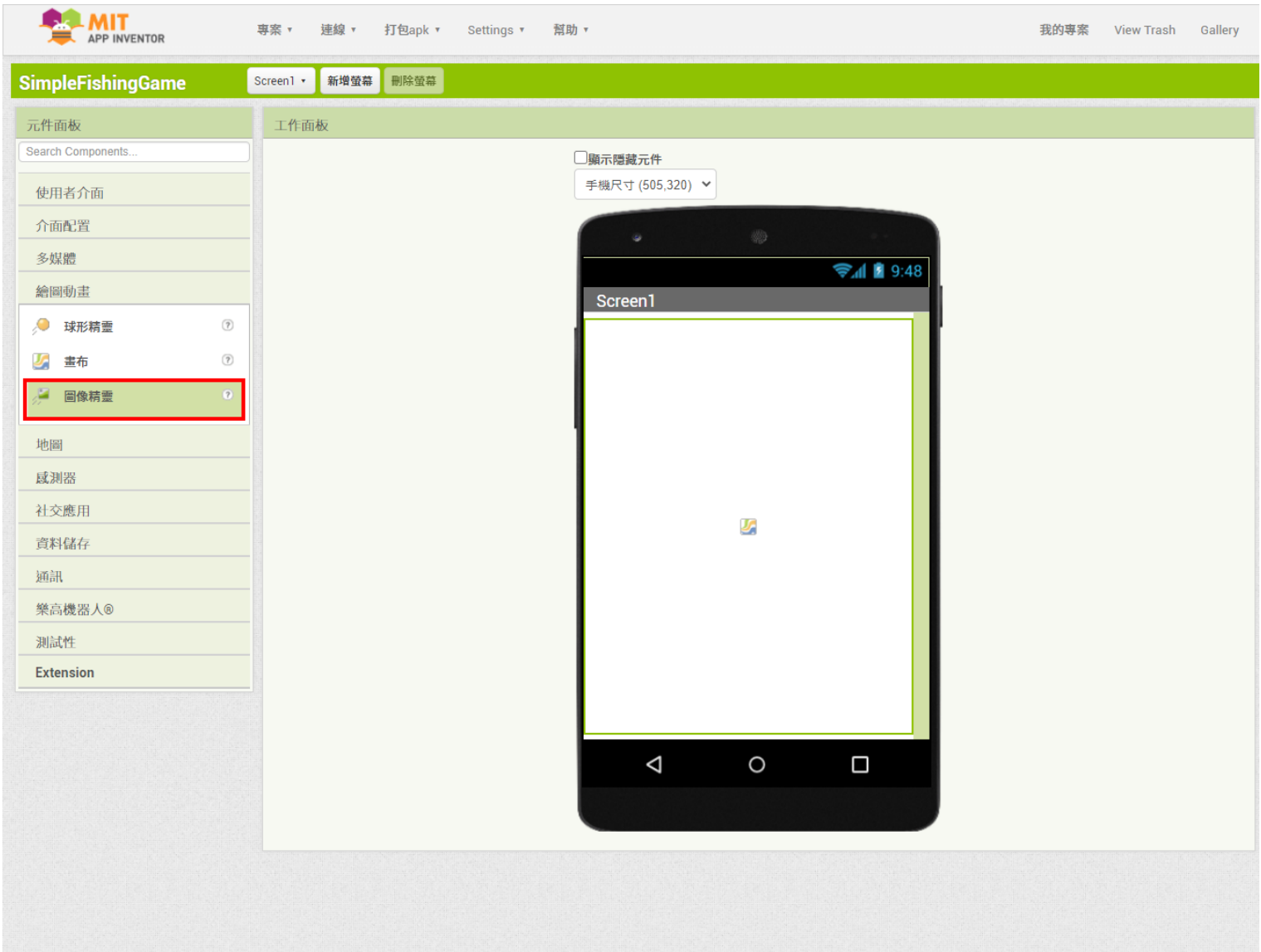


加入「繪圖動畫」-「畫布」





再加入「圖像精靈」元件：



接著「屬性」-「圖片」改為 初始圖像，此處我們選 Bait_01.png：

元件清單

Screen1

畫布1

圖像精靈1

重新命名

刪除

素材

Bait_01.png

Bait_02.png

Bait_03.png

Bait_04.png

Bait_05.png

Bait_06.png

Bait_07.png

Bait_08.png

元件屬性

圖像精靈1

啟用

☒

指向

0

高度

自動...

寬度

自動...

間隔

160

圖片

無

Bait_01.png

Bait_02.png

Bait_03.png

Bait_04.png

Bait_05.png

Bait_06.png

Bait_07.png

Bait_08.png

Bait_09.png

Catch_01.png

上傳文件...

取消

確定

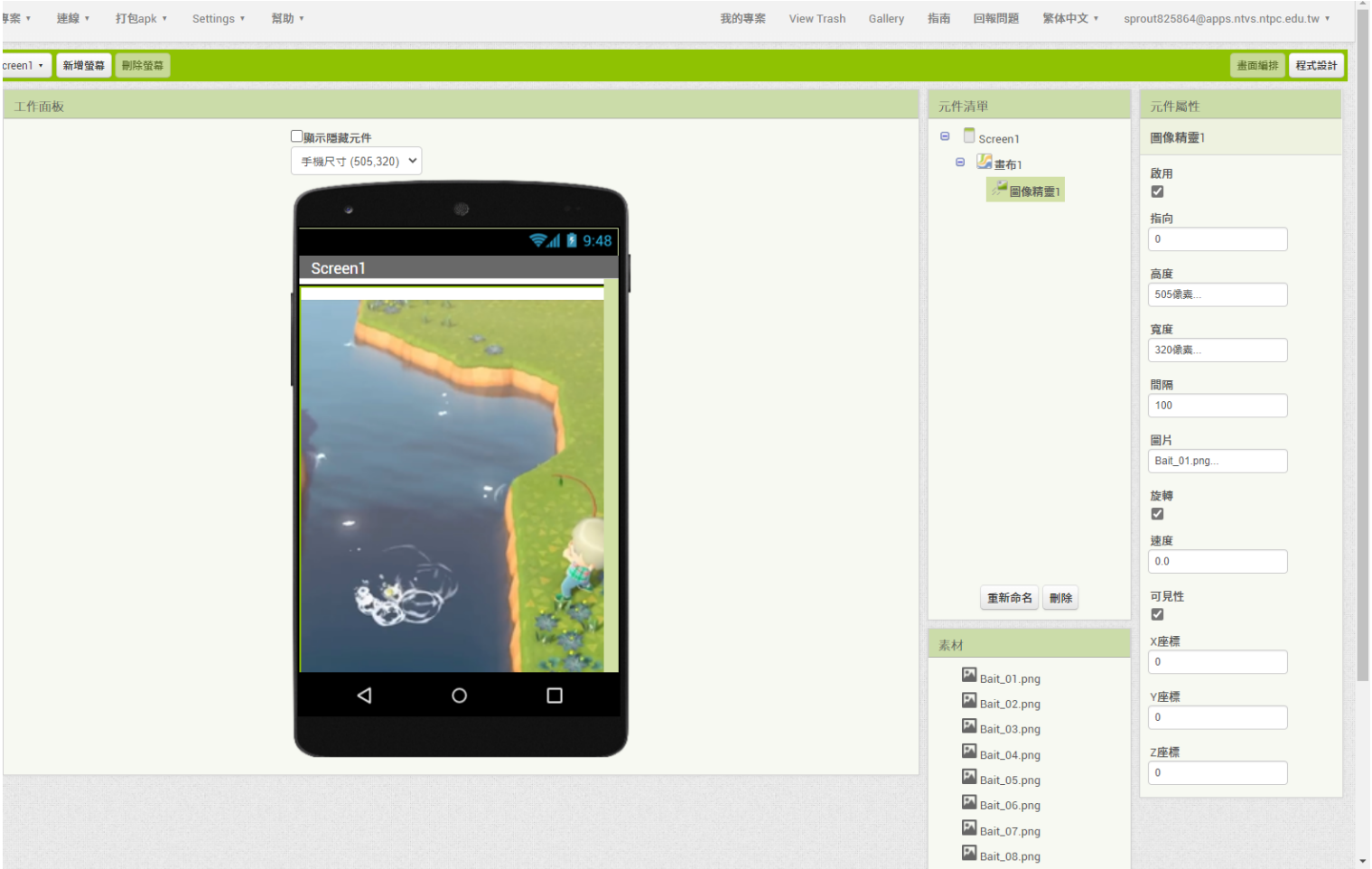
Y座標

13

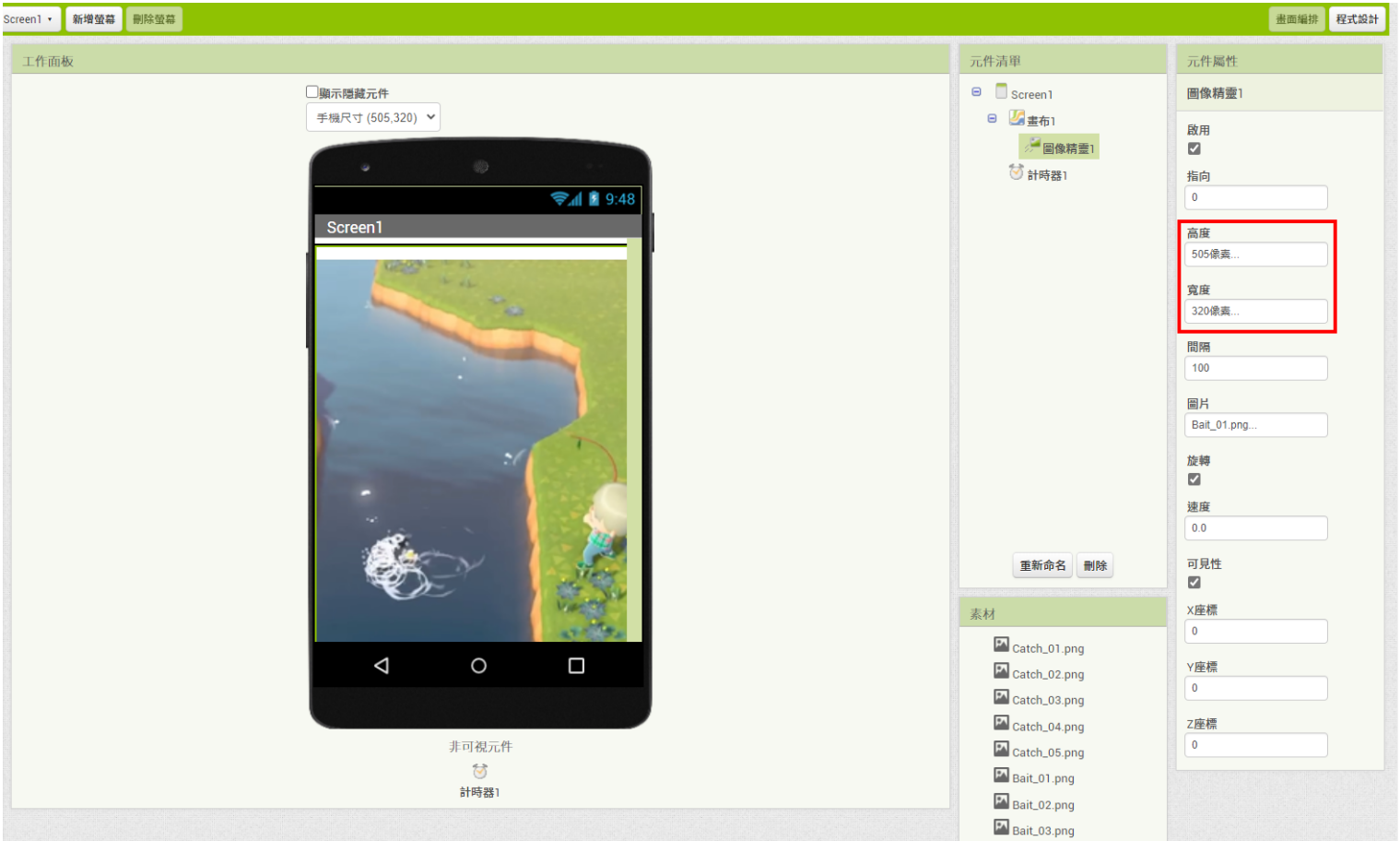
Z座標

1.0

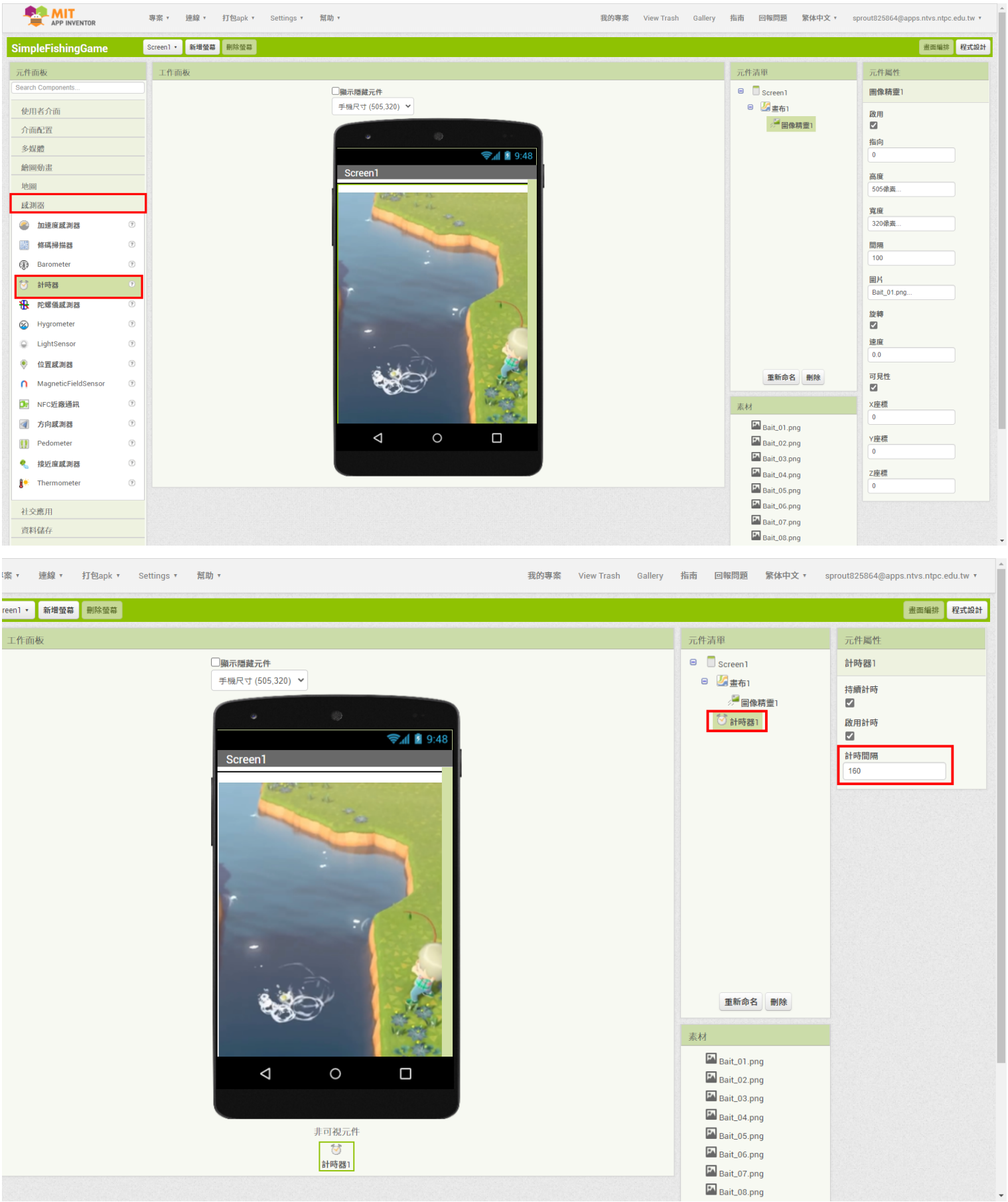
由於此處素材取自市售遊戲《集合吧！動物森友會》的片段內容，因此僅能作為教學練習用。



而且，由於 App Inventor 的限制，我們得將畫布尺寸及圖像精靈尺寸壓縮到 505 * 320。

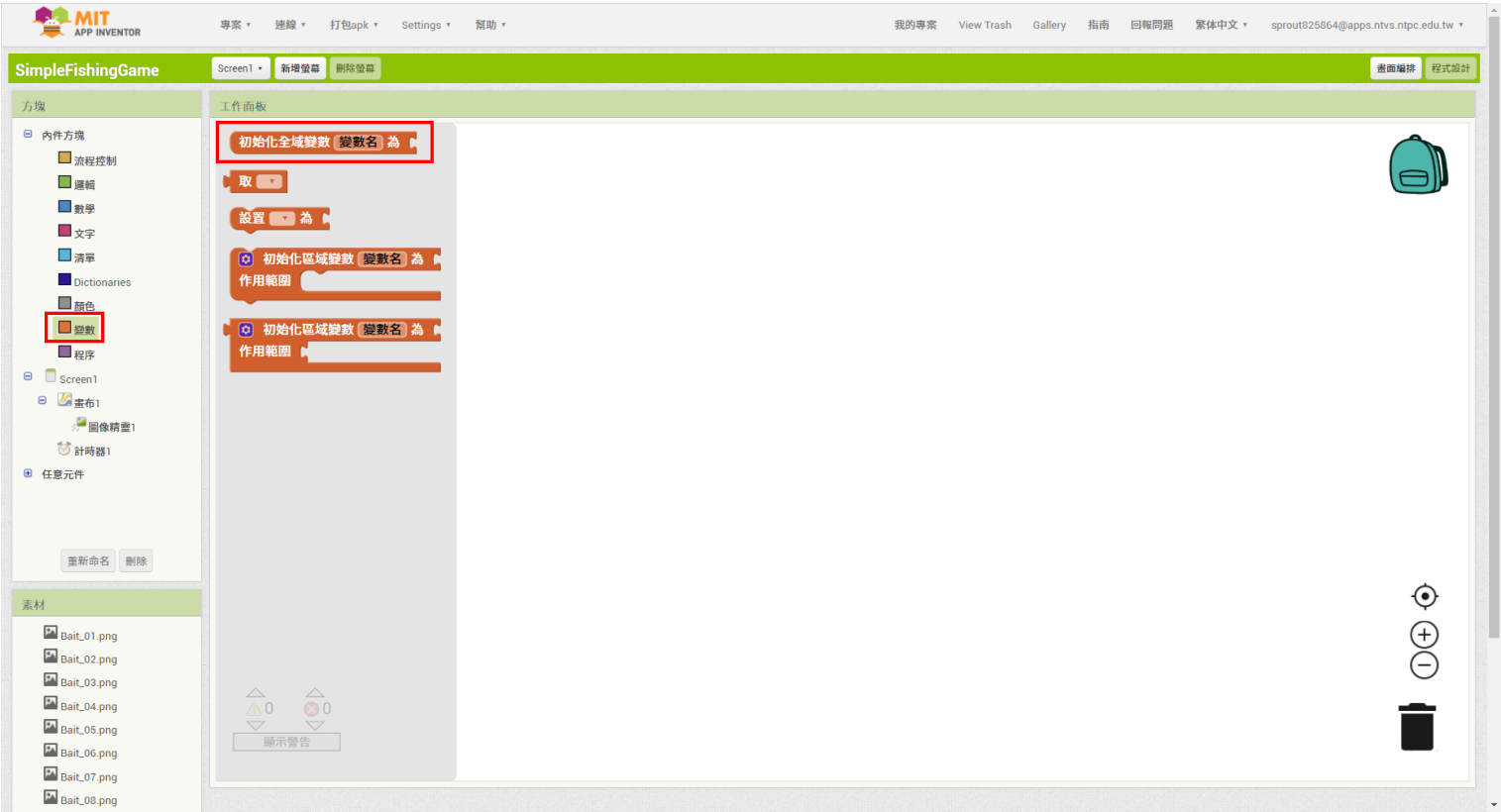


接著加入「感測器」-「計時器」，我們要設定每0.16秒（160毫秒）變換一次圖片：

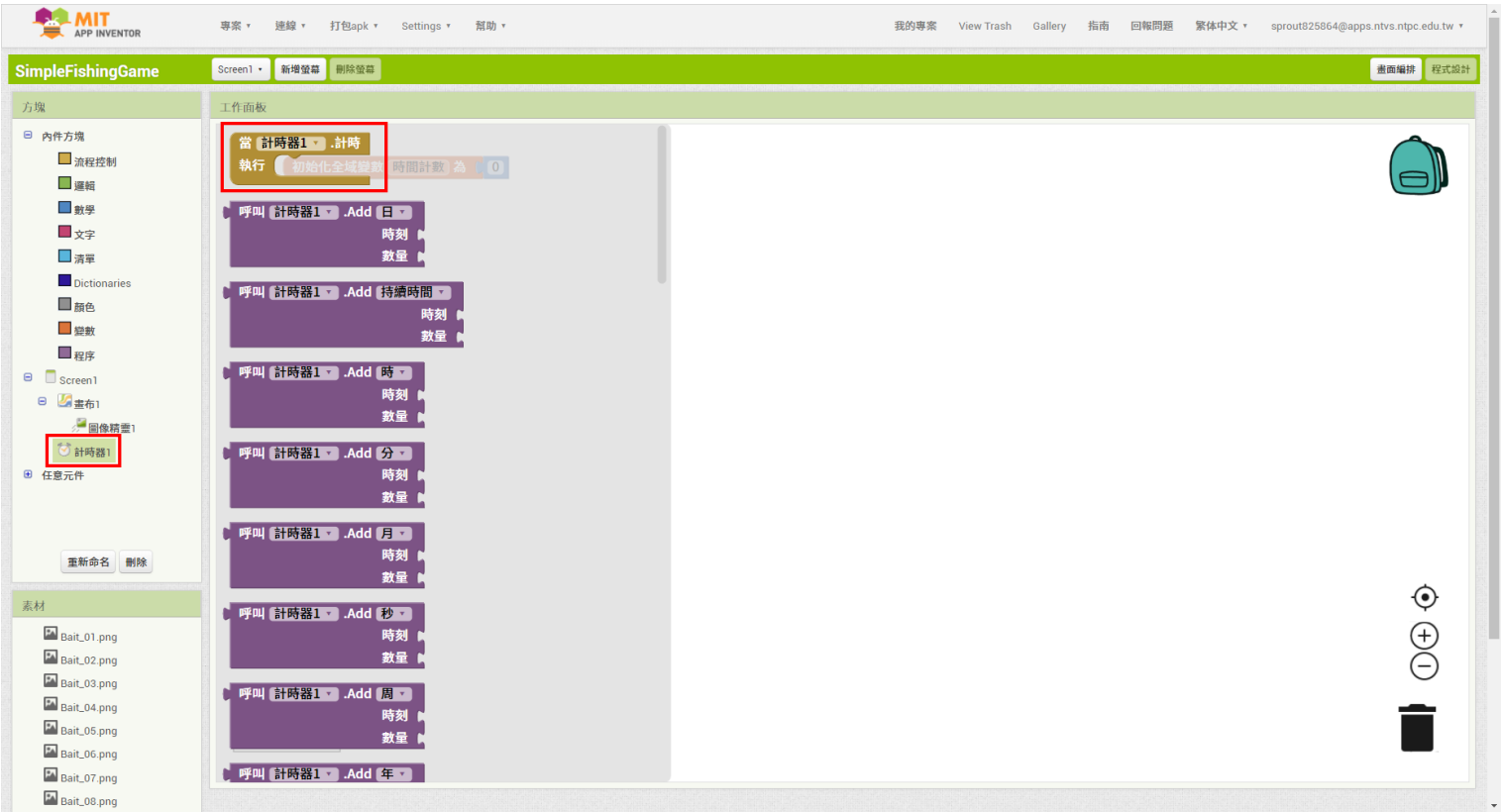


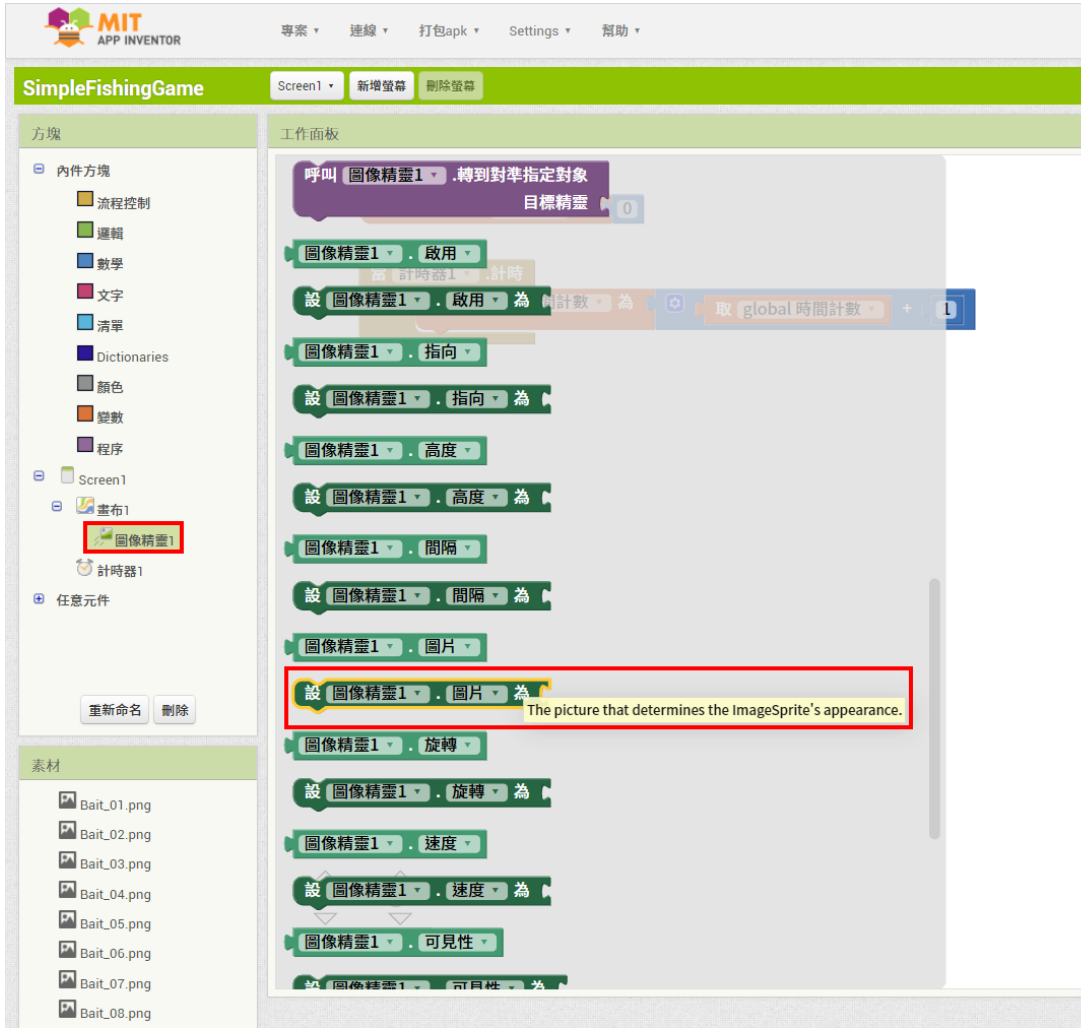
接著進入「程式設計」。

首先設定「變數」-「初始化全域變數」, 時間計數為0:

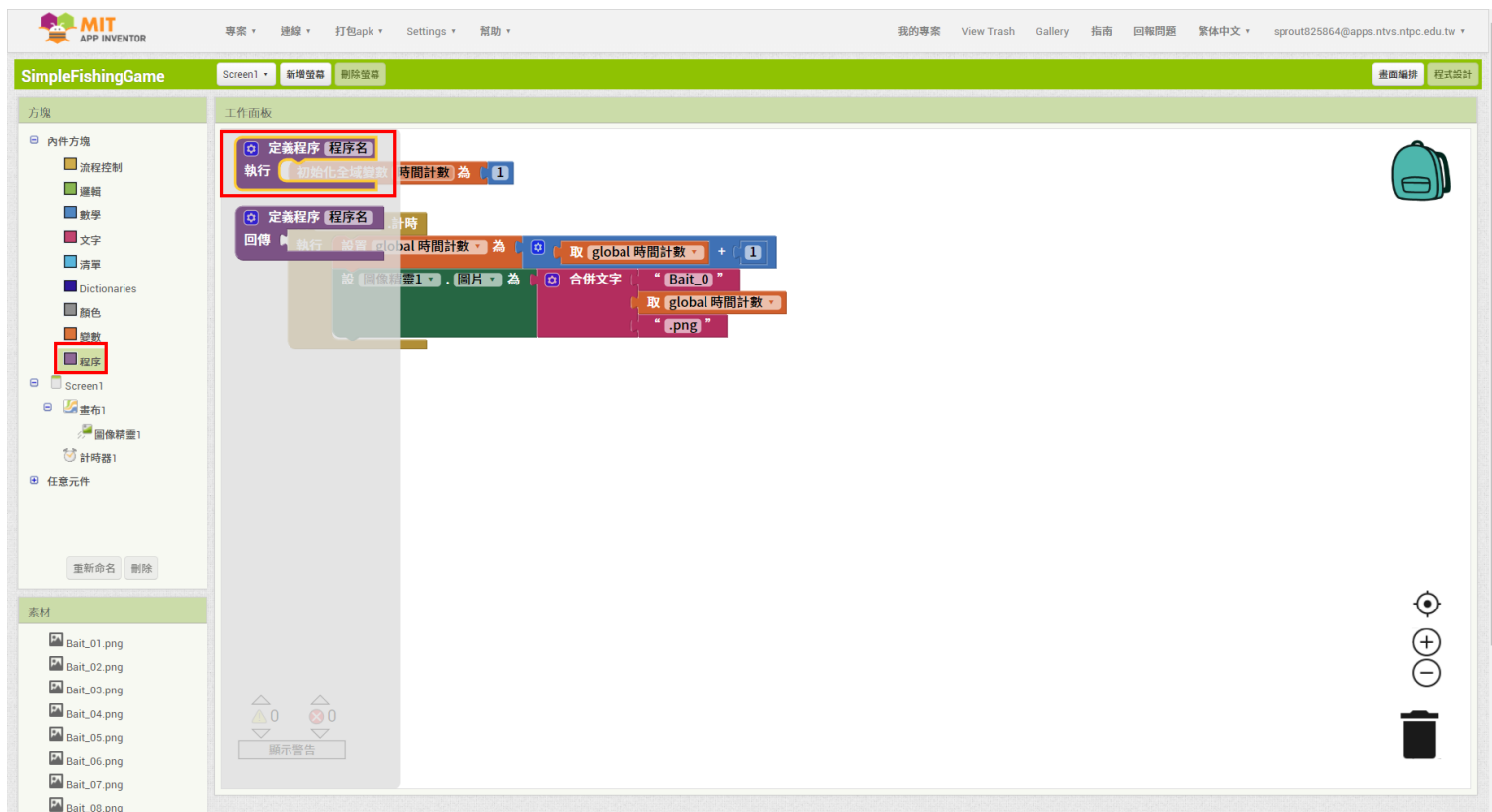


當計時器元件啟動後, 每經過0.16秒, 時間計數+1, 而且圖片名稱也+1, 就達成切換下一張圖片。

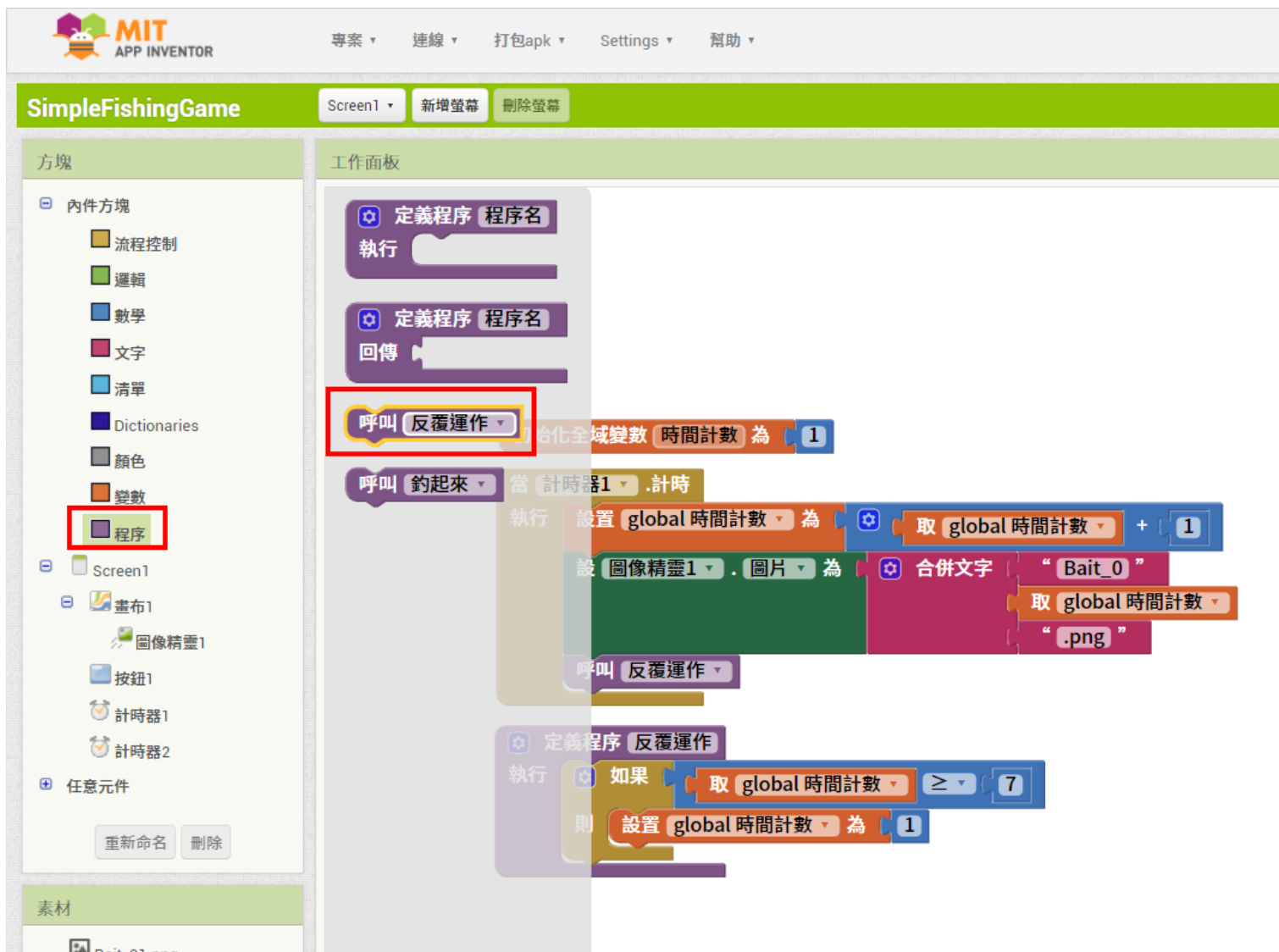




自訂程序，在其他程式語言普遍叫做「副程式」，我們現在自訂一個程序讓圖片反覆運作：



接著將反覆運作加入計時器程式：



接著快來試試當前成果吧！

測試結果

影片流程

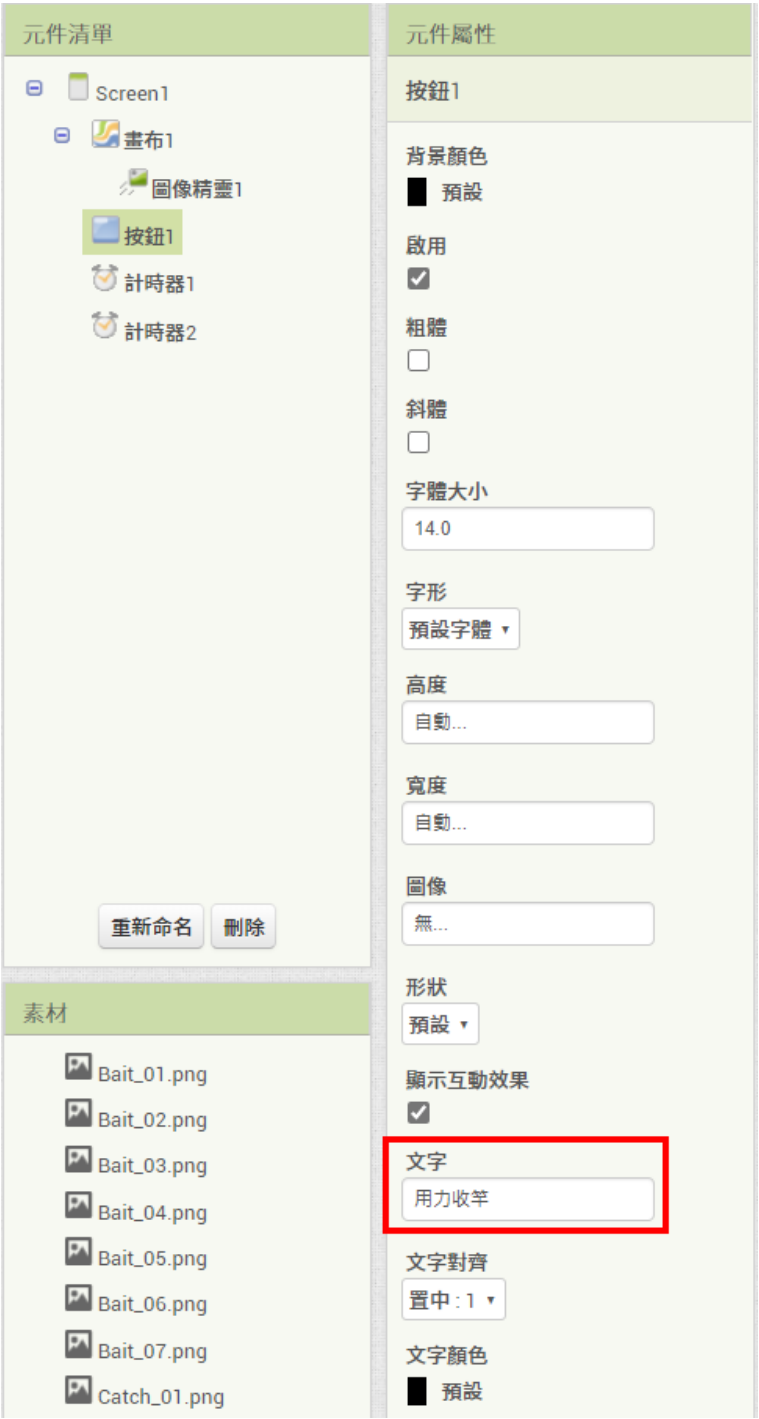


釣起事件

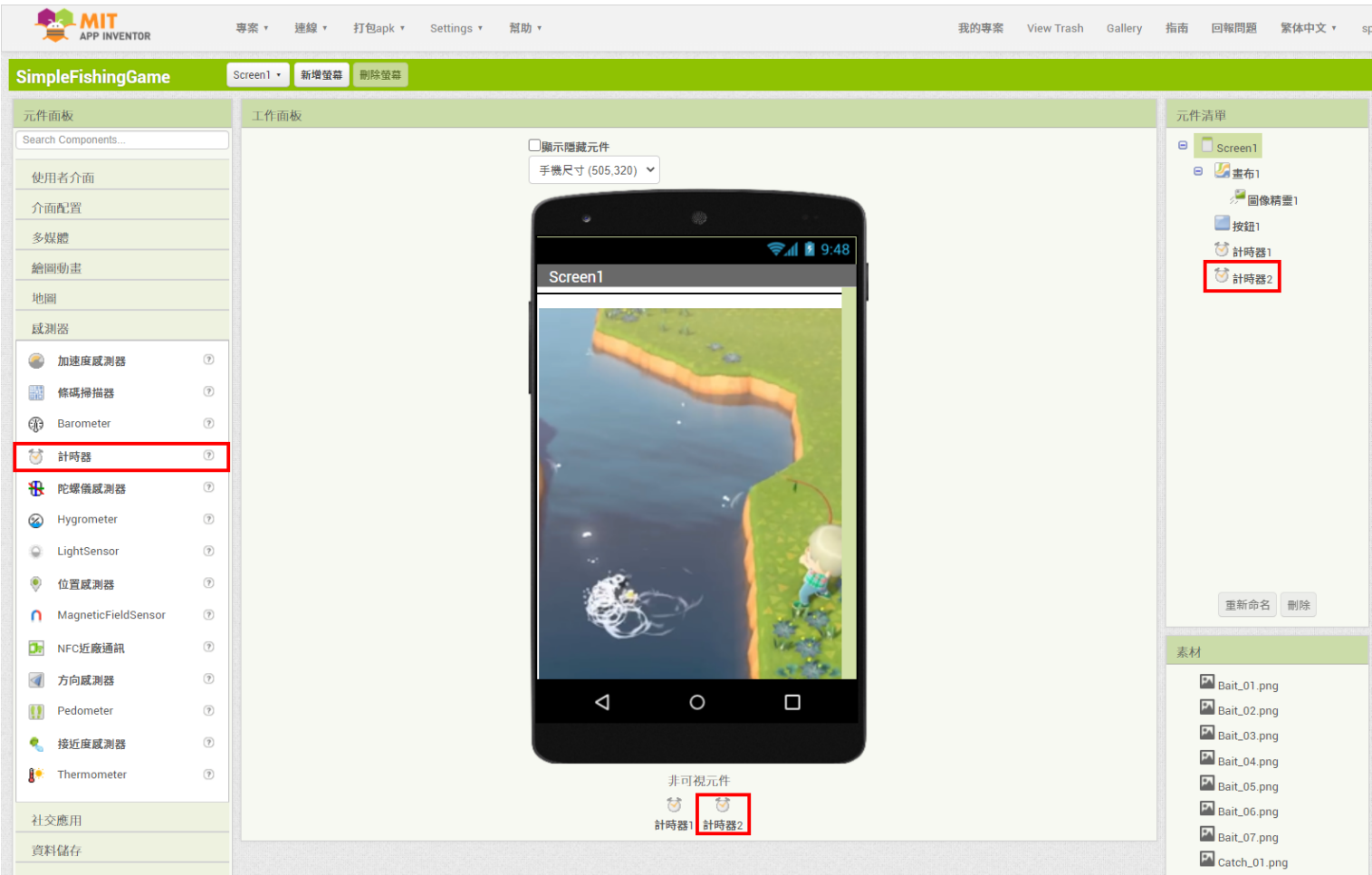
接著我們新增一個按鈕，讓我們能把這隻魚釣起來：



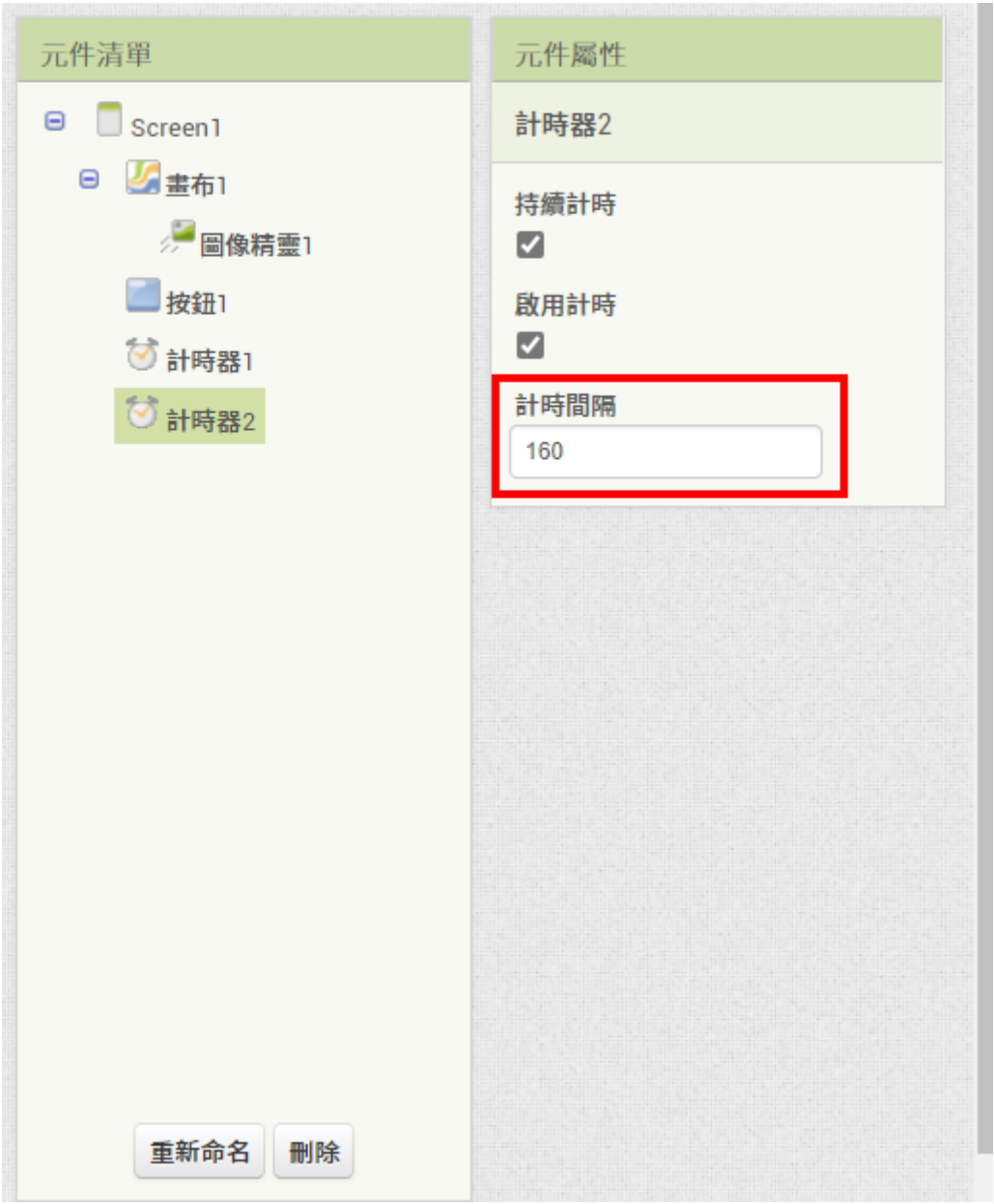
按鈕的文字可修改：



也加入計時器2：



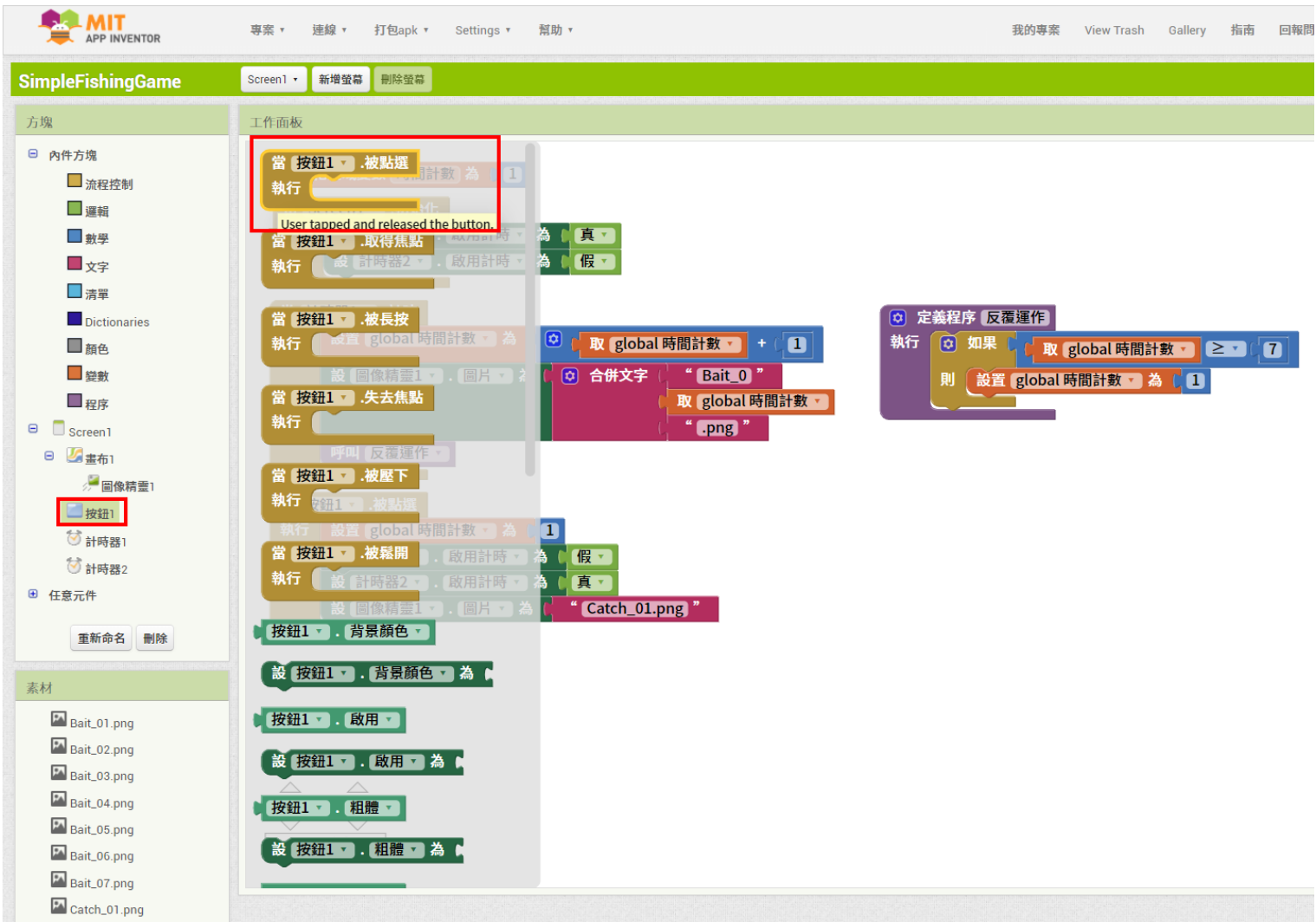
計時器2的計時間隔也設定為：



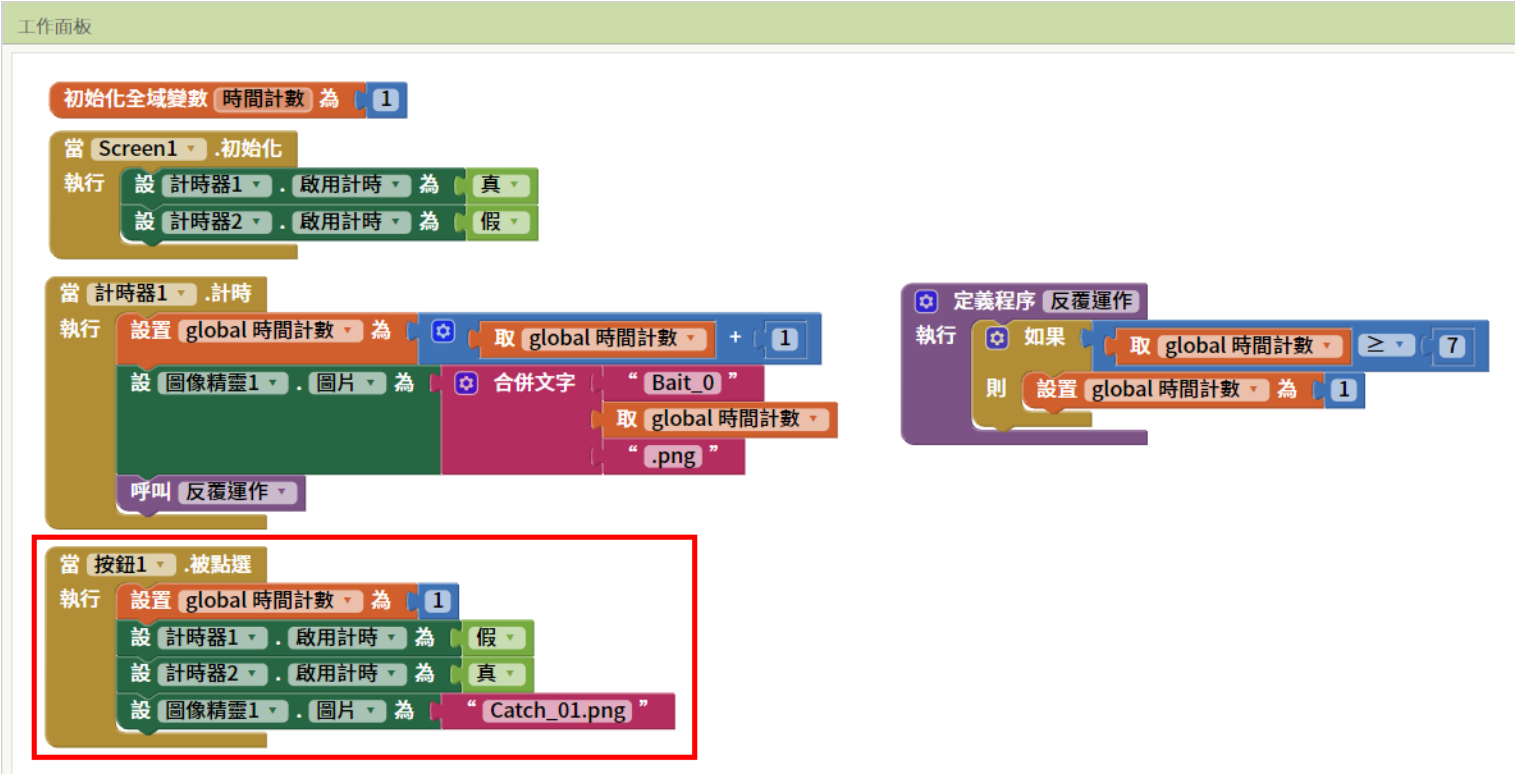
接著程式設計中需要增加的項目會比較多：
我們需要在初始化時，定義「計時器1」先開啓；「計時器2」先關閉：



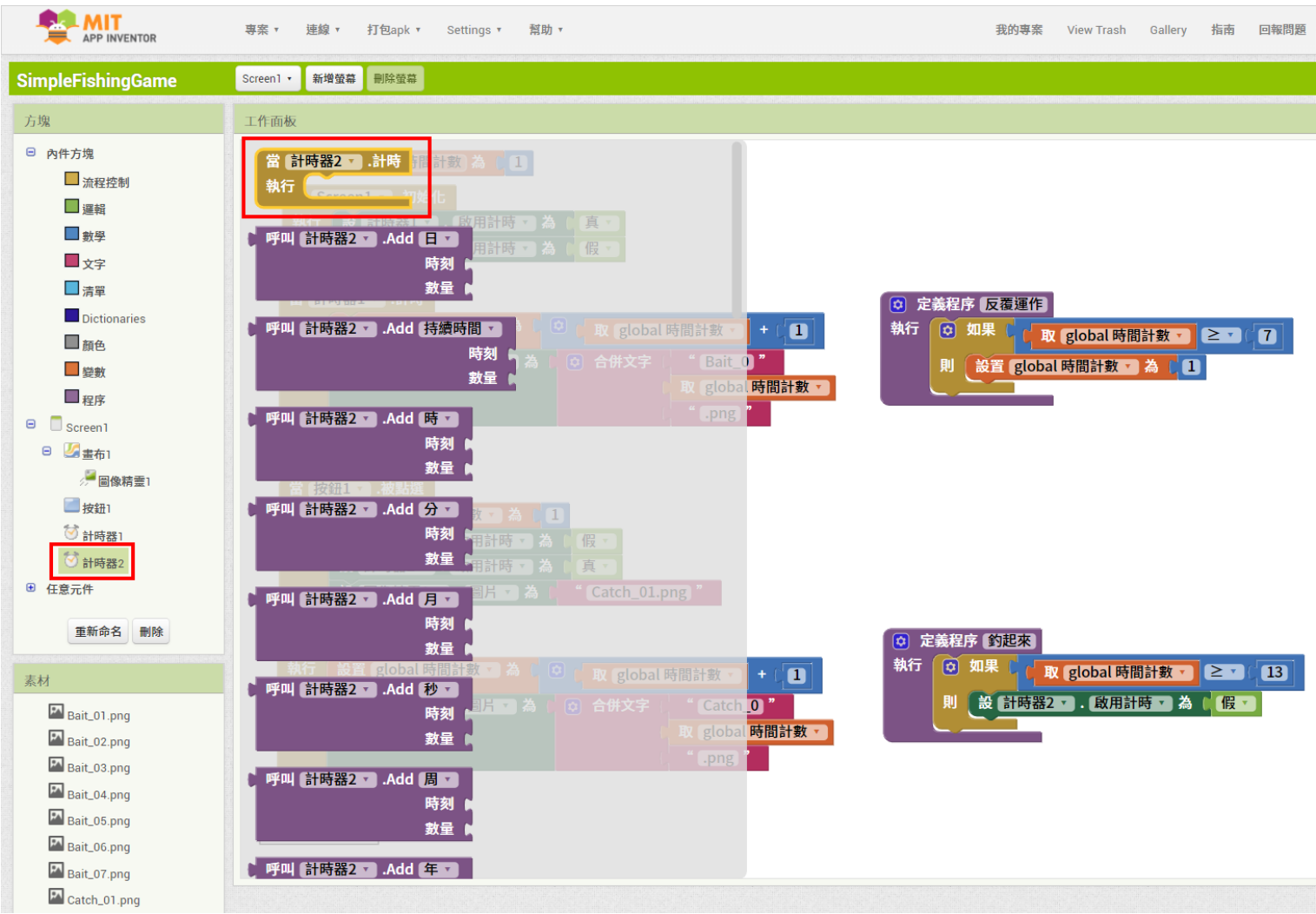
接著增加按鈕1的功能：



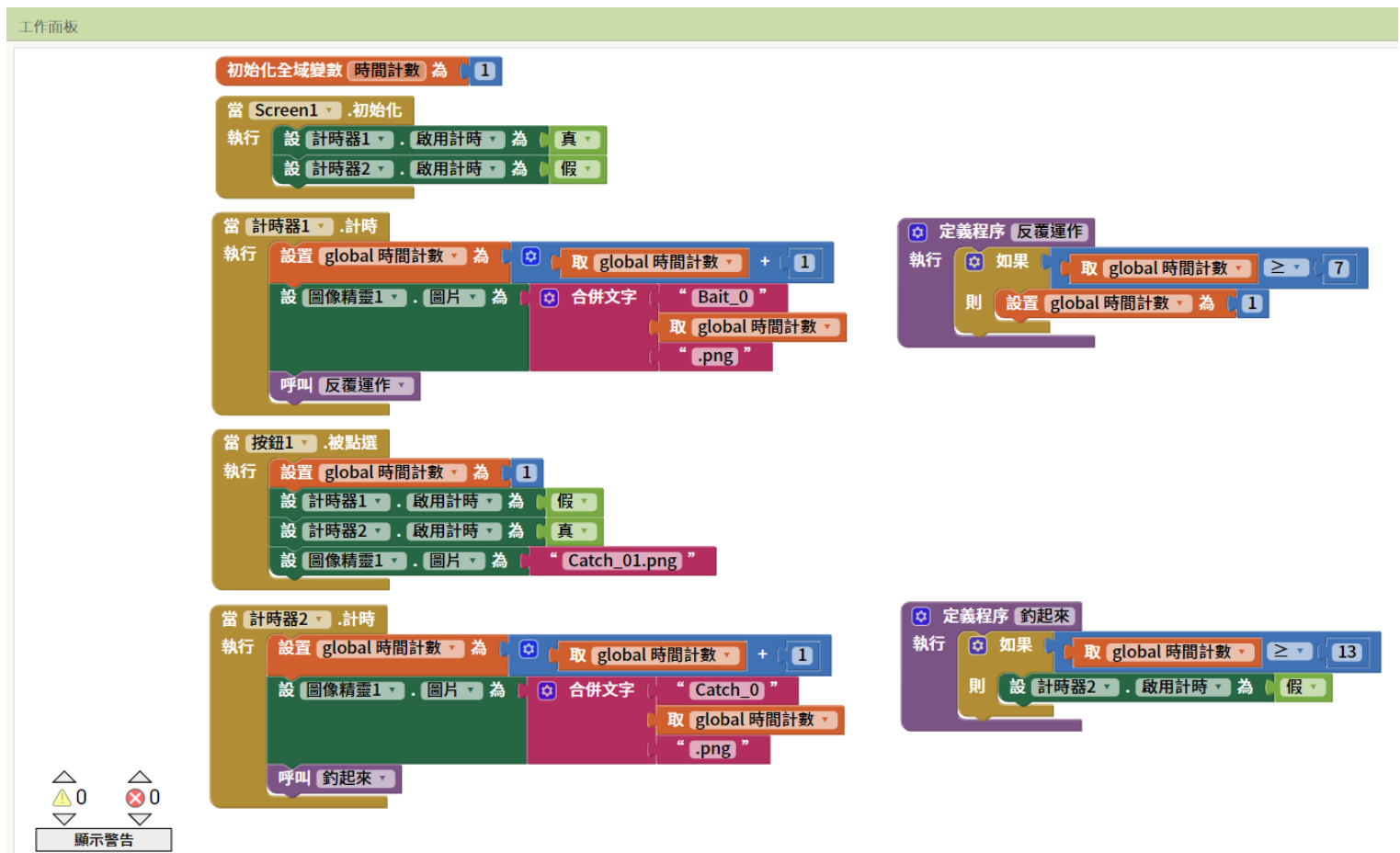
當按鈕1按下的時候,「計時器1」關閉;「計時器2」開啓:



接著是「計時器2」的功能設定:



因為和「計時器1」相似，因此不多作說明：



注意到右方又多出了一個「釣起來」副程式吧？這次不用反覆執行：



因為不用反覆執行，所以超出範圍就讓「計時器2」停下：



如此就完成釣起事件，乾淨利落。
你的第一支手機遊戲就這麼問世了！
動動腦，懂得舉一反三的你甚至可以為這遊戲加功能！

可附加的功能提案	
重新釣魚	如果能理解程式流程，你就能很輕易的在釣上魚後，重新再出竿釣第二隻、第三隻。
魚類百科	釣上來的究竟是什麼魚呢？ 用任意形式（標籤、方塊或其他）說明這隻魚的來歷吧！
隨機魚種	承上案，如果想要每次釣上來的魚都不同，你能不能做的出來？（撇除動畫部分魚無法輕鬆變更，百科魚類資訊的不同很容易吧！）
隨機大小	如果上面都不會做，但是懂得隨機數的使用，你應該能做出「讓每次釣上來的魚」，量測尺寸都不一樣，這個再容易不過了吧？
其他	你可以再思考一下，有什麼有趣的點子可以添加，盡情揮灑你的創意。

如果想挑戰看看附加功能，可以閱讀[【程式語言】釣魚遊戲功能擴增 - App Inventor 2](#)。

科目辨別/英文字彙測驗 App

已經有過在其他程式語言中，開發這個測驗的經驗，因此同學們理應能使用自己的邏輯思考，撰寫出這支程式，在此不作贅述，直接看實例：

至於之前很痛苦，讓各位邊宣告陣列(會計科目及意義)邊心底碎碎念的步驟，我們此時可以藉由「csv轉list輕鬆達成」：

[會計csv下載點](#)

[英文字彙csv下載點](#)

