



Android專題 — 踩地雷

508062334 軟創三乙 陳彥志

511172176 軟創二乙 李則霖

目錄

一.起源

二.畫面

三.程式

四.延伸

五.總結



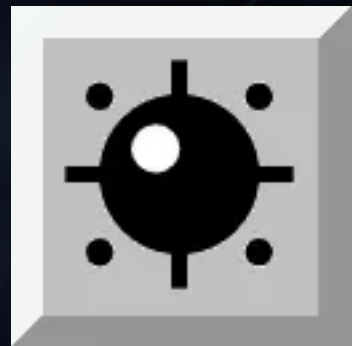
一.起源



學習APP製作

提出需求:想要一有款隨時隨地
可以動腦的小遊戲

解決辦法:利用現有專案學習並
練習製作開發

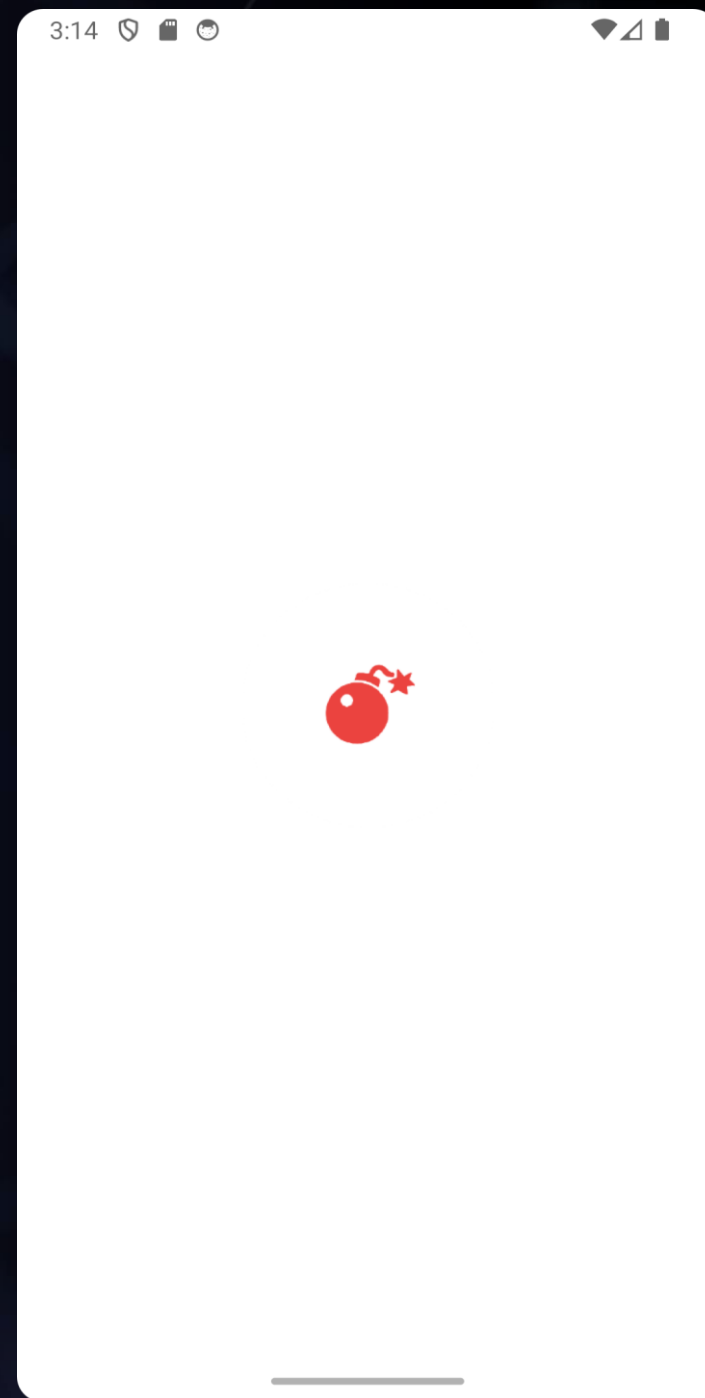


二.畫面



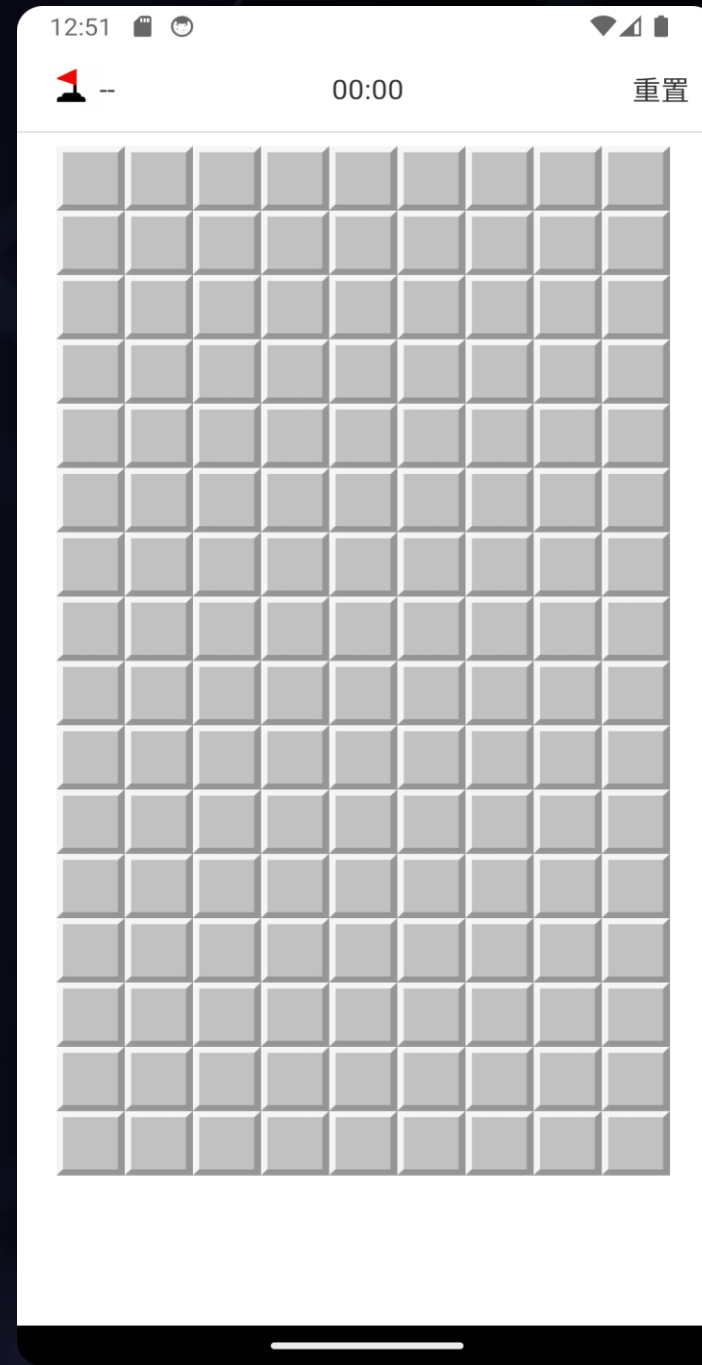
畫面0

開始畫面



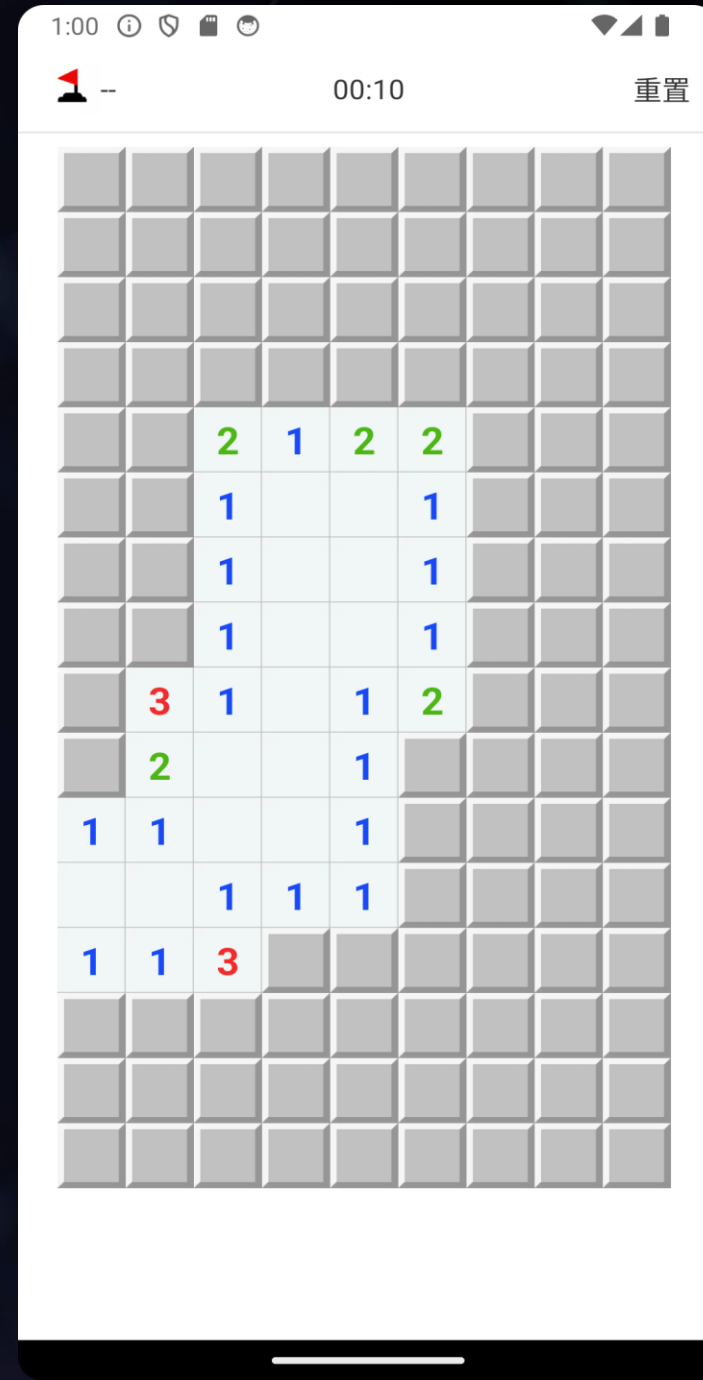
畫面1

預設9 x 16格格



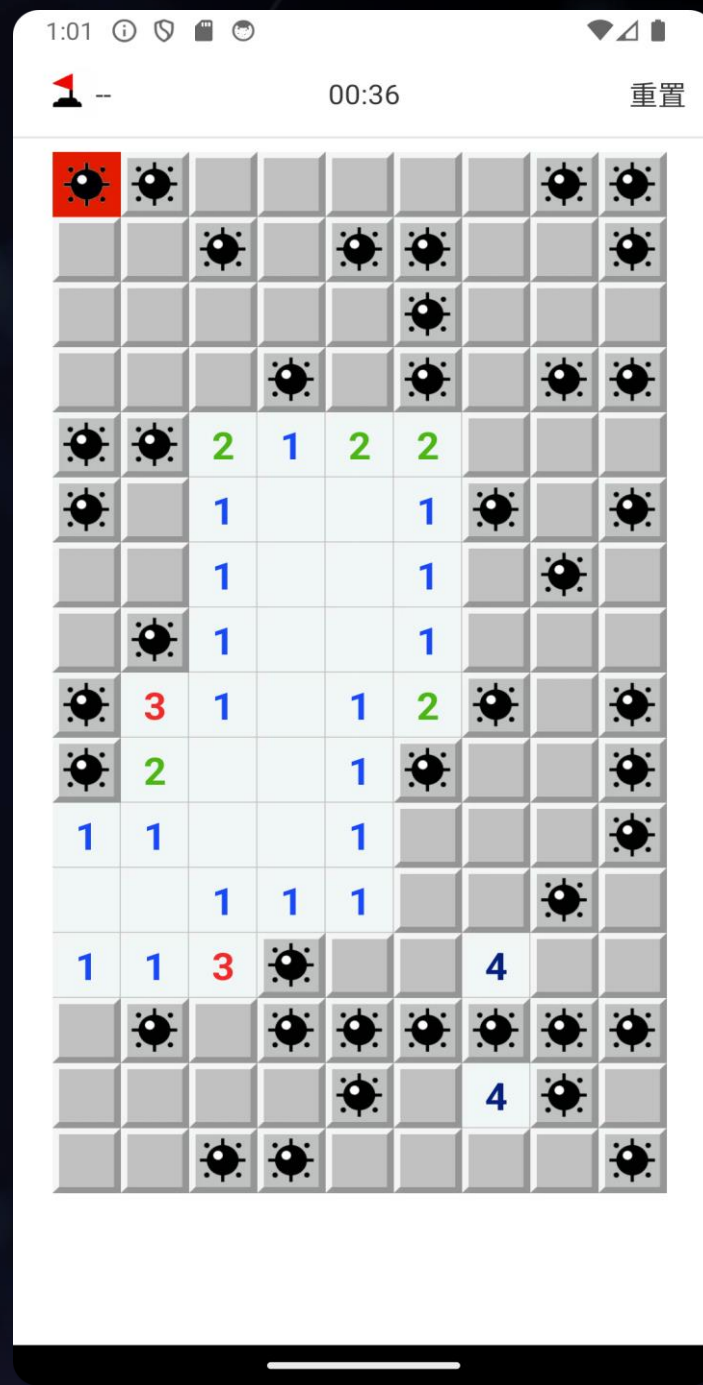
畫面2

沒踩到地雷格子打開



畫面3

踩到地雷遊戲結束



畫面4

設置旗子及問號



畫面5

遊戲勝利畫面

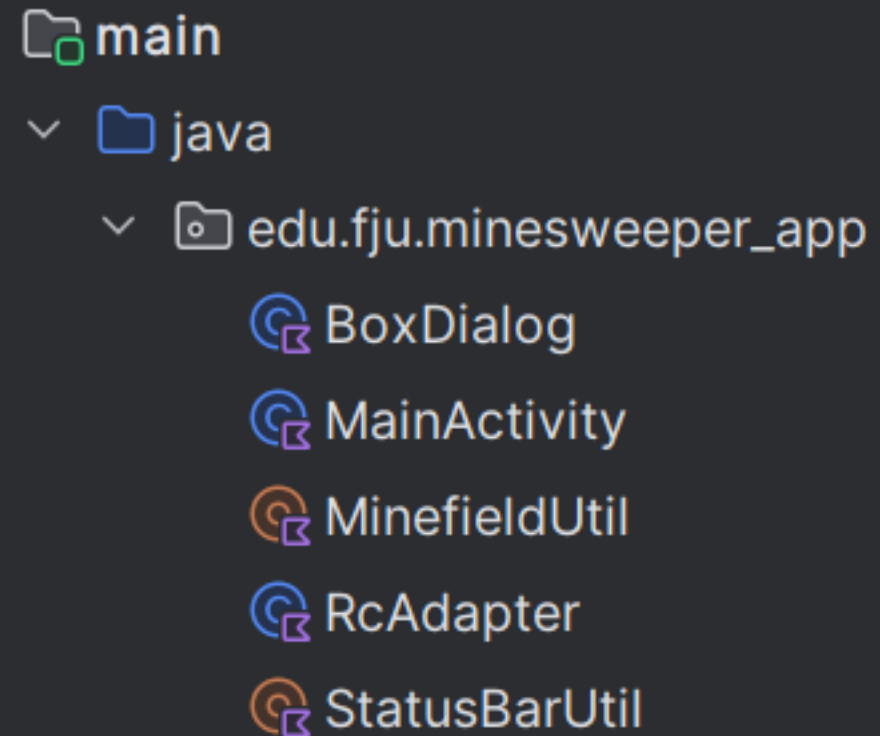


三.程式



main

主程式及其他相關程式放置處



BoxDialog.kt

顯示遊戲勝利、遊戲花費時間、再來一局視窗。

fun onCreate：初始化對話框，設定內容視圖，配置對話框行為。

MainActivity.kt

主程式，負責初始化遊戲介面、處理遊戲邏輯、以及更新遊戲狀態。

fun onCreate：應用程式啟動時被調用，用於設定活動的佈局、初始化UI控制項、配置狀態列樣式，以及設定重置遊戲的點擊事件監聽器。

fun initView：初始化遊戲介面，定義遊戲開始、結束、使用旗幟、取消旗幟、和獲勝時的邏輯。

fun dip2px：用於將dp單位轉換為px單位，以適應不同螢幕密度。

MinefieldUtil.kt

踩地雷遊戲的工具類，用於管理管理遊戲的地圖狀態、追蹤玩家的操作、處理遊戲的開始和重置邏輯。

fun establish：初始化遊戲地圖，根據玩家首次點擊的位置佈置地雷，確保玩家第一次點擊不會踩到地雷。

fun createTrapsNumber：計算並更新每個格子周圍地雷的數量。

fun createSpecialCoordinates：創建一個特殊坐標列表，這些坐標是根據玩家第一次點擊的位置確定的，確保這些區域不會佈置地雷。

fun reset：重置遊戲的狀態，包括地圖資料和玩家操作記錄，準備開始新的遊戲。

RcAdapter.kt

管理如何將資料展示在每個格子內，以及回應使用者的點擊事件。顯示遊戲格子、處理使用者互動、遊戲狀態更新。

fun onCreateViewHolder：持有中每View個小格子的信息。

fun getItemCount：渲染多少個格子的View。

fun onBindViewHolder：根據遊戲的狀態配置每個格子的顯示，設定背景、文字，並處理點擊事件。

fun exploitation：遞迴方法，當使用者點擊到數字0的格子時，自動打開周圍的空白區域，直到周圍有數字的格子為止。

fun setDataCallback：設定回調函數，用於遊戲狀態的更新，遊戲開始、結束。

interface DataCallback：定義了回調方法 `gameStart`、`gameOver`、`gameWins`、`cancelFlag`、`useFlag`，用於在外部處理遊戲狀態的改變，顯示遊戲勝利、結束遊戲、使用旗子、取消使用旗子。

StatusBarUtil.kt

工具方法，用於改變狀態列的顏色、透明度、文字顏色以及圖示顏色。

fun immersive：設定狀態欄，指定顏色和透明度。

fun darkMode：根據傳入的值來設定狀態列文字和圖示的顏色模式，true 為深色文字和圖示。

fun setPadding & fun setPaddingSmart：調整View的paddingTop使內容不被狀態列遮擋。

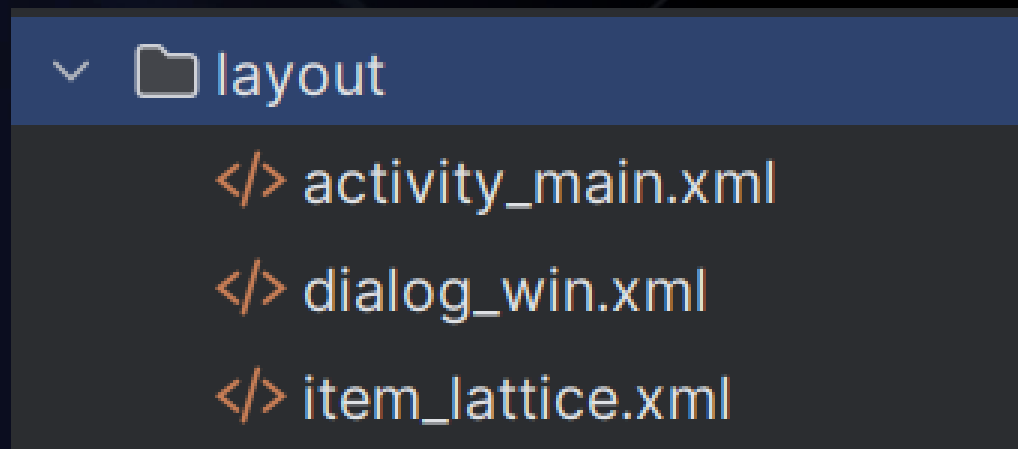
fun setHeightAndPadding：同時調整View的高度和paddingTop。

fun setMargin：調整View的marginTop防止內容被狀態列遮蔽。

fun setTranslucentView：在容器中新增與狀態列同高的透明視圖，可用於建立自訂的沉浸式效果。

layout

圖形介面設置檔案放置處



activity_win.xml

LinearLayout：整個布局的容器，方向設置為垂直，背景設定為白色。

LinearLayout (titleBar)：頂部的標題欄區域，包含旗子數量圖標和時間。

ImageView：展示旗子圖標，圖標來自@mipmap/icon_explain_flag。

TextView (flagNum)：顯示旗子的數量，初始化為"--"。

TextView (time)：標題欄中心顯示時間，初始化為"00:00"。

分隔線View：在標題欄下方，區分標題欄和內容區。

內容區LinearLayout：左右設置22dp的外邊距，允許動態調整大小。



dialog_win.xml

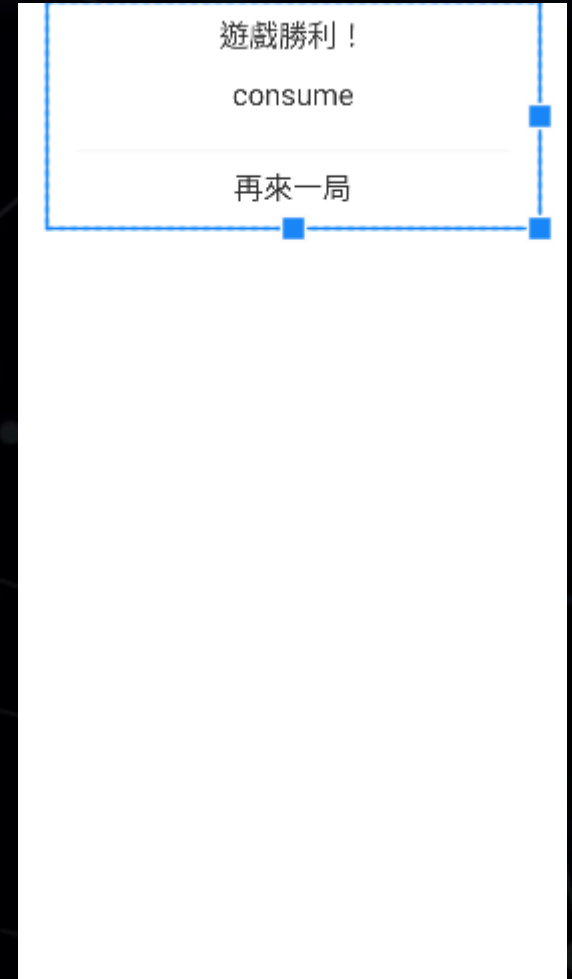
LinearLayout：最外層的容器，以垂直方式排列其中的子視圖。

子LinearLayout：顯示格子底圖，使用@drawable/dialog_bg。以垂直方式排列，包括遊戲勝利的文字和一條分隔線。

分隔線：左右邊距為22dp，分隔按鈕。

Consume：顯示遊戲所用時間。

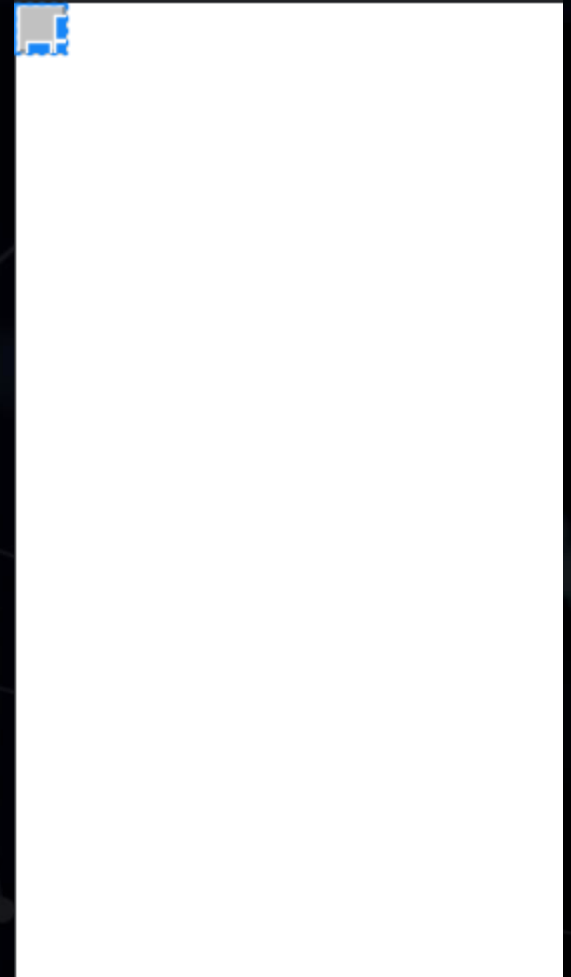
再來一局按鈕：顯示再來一局的文字，重新開始遊戲的按鈕。



item_lattice.xml

LinearLayout：創建一個寬高38dp的方形容器。

TextView (itemText)：未踩地雷格子，圖案來源
@mipmap/icon_lattice。



mipmap-xxhdpi

格子、地雷、旗子圖標檔案放置處

▼ mipmap-xxhdpi

- ic_launcher.webp
- ic_launcher_round.webp
- icon_doubt.webp
- icon_empty.webp
- icon_explain_flag.webp
- icon_flag.webp
- icon_flag_error.webp
- icon_lattice.webp
- icon_trap.webp
- icon_trap_open.webp

四.延伸



延伸應用

希望利用此專案延伸再參考地雷破壞者專案，製作出能自己解地雷的APP。

Minesweeper spoiled by AI

😏 "I do the *brain-testing* part for you" 🧠

😬 "You do the *luck-testing* part" 🍀



Level:

五.總結



參考來源

踩地雷專案:

<https://github.com/ThirdGoddess/trap>

地雷破壞者專案:

<https://github.com/mame/minesweeper-spoiled-by-ai>

感謝大家的聆聽

