

Computer Organization and Assembly Languages Final Project

# POP TANKS !!

跑跑坦克車



B95902016 黃彥翔

B95902027 王璽承

B95902058 郭明鑫

B95902088 梁逸安

# ***INDEX!***

---

目錄.....	<b>1</b>
遊戲製作靈感.....	<b>2</b>
設計概述.....	<b>3</b>
實作內容.....	<b>5</b>



## 遊戲製作靈感

我們製作此遊戲的靈感，是小時候相當流行的一款電視遊樂器，任天堂遊戲機(俗稱紅白機)，當中的一款經典遊戲「坦克大戰 Battle City」。坦克大戰是一款平面射擊遊戲，在 1985 年由任天堂公司推出，遊戲中並沒有劇情路線，當玩家破完所有的關卡後，便回到第一關繼續遊戲。



【坦克大戰的遊戲畫面】

在遊戲中，玩家控制的坦克以消滅敵軍坦克為目標，並要守護畫面下方的基地，不被敵軍摧毀，若是坦克失去了三次生命，或是基地被敵軍摧毀，玩家便落敗，遊戲便結束。



【遊戲結束畫面】

## 設計概述

我們所設計的遊戲，一開始玩家的坦克會在畫面下方的國王標誌旁出現，玩家的目標是摧毀所有的敵人，而這次凶狠的敵人不再是敵軍的威猛坦克，取而代之的是我們熟悉的卡通人物－HelloKitty、Keroro 和 Giroro。玩家在他們可愛的外表下，是否會心軟而無法達成任務？在面善心惡的敵人們夾攻之下，玩家又能否達成任務前往下一個關卡？



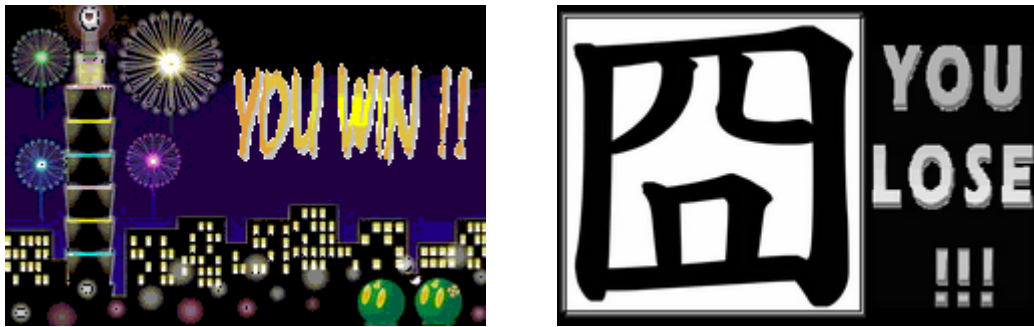
【遊戲進入畫面】

我們的關卡一共有兩關，當你在第一關消滅了全部的敵人之後，便可以進入到第二關，若是不幸被擁有可愛外表的敵人給擊敗了，遊戲就結束了，可說是一款容易上手的小品遊戲！



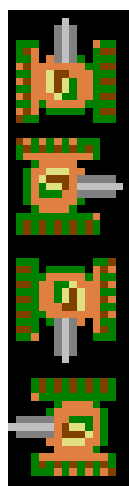
【兩關的遊戲地圖畫面】

遊戲中，最下面的 KINGS 便是國王標誌，玩家的坦克由此處開始移動，花草和泥土地是坦克可以通過的區域，而樹木、岩石和水地是坦克不能通過的區域，另外還有橋樑讓坦克可以越過水地障礙。

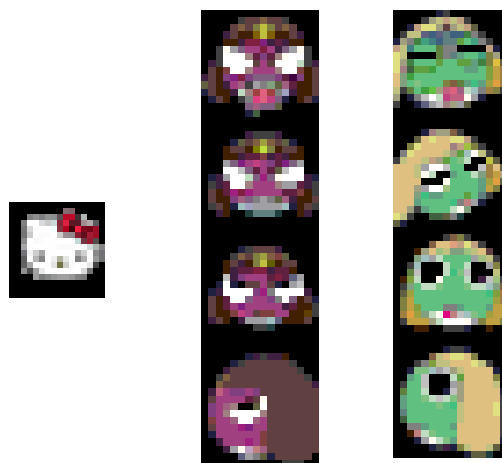


【遊戲中的過關畫面和失敗畫面】

遊戲中玩家的坦克若被敵人擊中或是觸碰到敵人，便會被摧毀，而玩家最多可以復活三次，若超過三次遊戲便結束了。而每一個敵人則是有三到五滴血（被擊中三次以上才會被摧毀），而敵人可以復活的次數也是三次。



【玩家所操控的坦克】



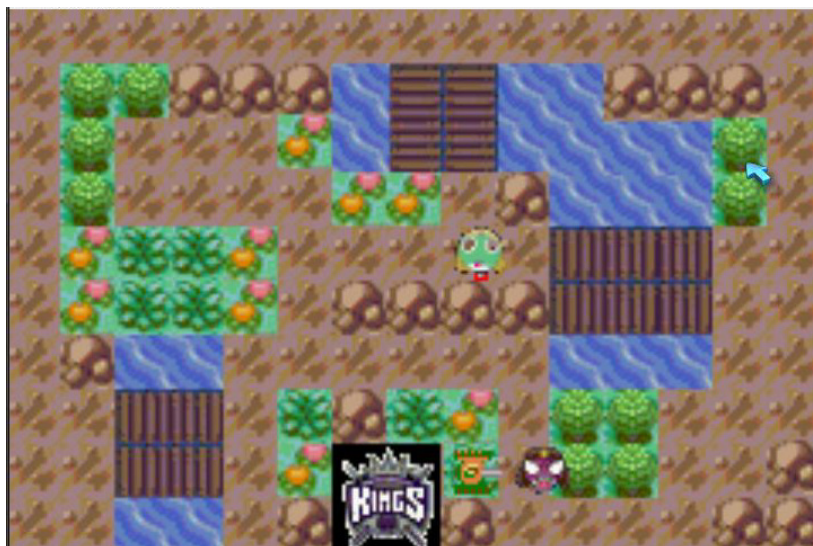
【遊戲中的敵人—Kitty·Giroro·Keroro】

## 實作內容

整個遊戲的內容其實不會太過複雜，一開始大致可分成三個部分，第一個部分是 `background`，這部分需要包含的就是每一個畫面的製作，像是一開始跑跑坦克車的畫面、`stage1`、地圖一、地圖二等等，而每一個圖都必須經過小畫家去調整我們所需的大小，再由 `gfx2gba` 這個程式來產生.c 檔讓 `ham` 可以幫我們 load 進去，在要 load 一個檔時我們利用 `ham_InitMapFragment` 這個函式。

此外，我們在組成這個 `160*240` 的地圖時，爲了達到讓電腦判定此塊區域可不可以通過，我們以 `16*16` 的圖形爲一個單位，再利用一個 `maparray[2][10][15]` 來建表；第二部分，是有關 `GBA` 的 `input`，也可以說是我們如何控制我們的坦克，`query_buttons` 這個函式裡面記錄了如何對按下的按鍵做出反應，這一個部份特別的是我們必須讓坦克在遇到障礙物無法繼續前進，所以要設下適當的條件使得按鍵無效，這方面利用之前建好的 `maparray` 在加上對座標的運算(也有很大一部分在計算 `pixel`)來做判定；

第三部分電腦的 `AI`，這部分可以說是整份作業最讓人煩惱的部分，如何控制電腦出現、死亡、往我方坦克走來，這是這部分的大目標，出現和死亡和我方坦克的實作方法大致相同，而電腦自行前進的方法，我們嘗試了幾個部分，第一部分是使用 `greedy` 演算法，直接計算最短的方向往前進，不過這個方式我們在加入了障礙物條件後就遇到了挫折，再來有助教所提示的 `BFS`，這邊由於 `coding` 能力不足，加上時間的問題作罷，再來是嘗試 `rand` 的方式 `rand` 前進的方向，不過這個由於不知名的原因沒有達到理想效果而放棄，最後在 `code` 裡面的是加了一個 `hit` 的參數記錄射不中的次數，來改變方向前進，在加上每次前進時判斷是否可通過的條件，大致上就可以做出電腦往坦克進攻的樣子。而在電腦發射子彈的部分，還是用已知的電腦位置、我方位置，計算出較適合的射擊方向，一樣是用 `if statement` 來達到效果。



### 【遊戲中的畫面】

整個 GBA 大致上可分成幾個函式來看，main 裡面處理背景，還有一些全域變數；update\_plane\_pos()，更新我方坦克的位置，還有一些被子彈打到、被魔王撞到的判斷；update\_plane\_shit\_pos()，設定我方坦克發射的子彈飛行的方式；update\_kitty\_pos()、update\_gero\_pos()是電腦行走的 AI 部分，當然也有包括被坦克打到的判定。GBA 遊戲的撰寫，真的是一個很好玩的經驗，從沒有基本的認識，但大致懂得一個 GBA 遊戲的產生，它所需要的、所用的、所克服的大概會是哪些部分，這一次自己嘗試去學個新東西的經驗，真的是一個很好的磨練。