

Computer Organization and Assembly Languages

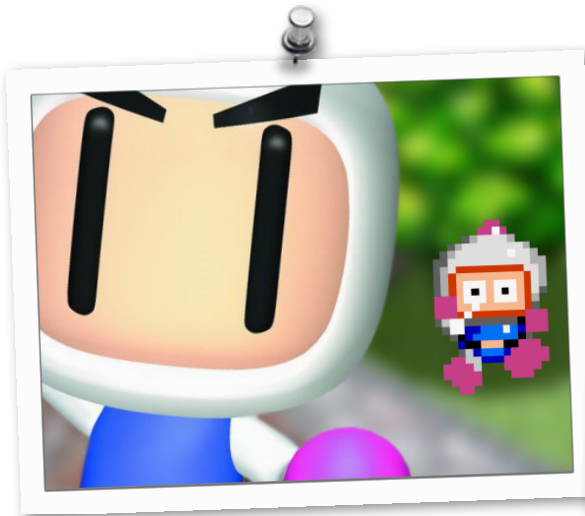
Final Project

QQ Bomber man



楊育宗 B96902108
楊詠翔 B96902076
沈暉原 B96902068
林煒皓 B96902074

INDEX



設計理念

遊戲介紹



實作內容

程式碼中的函式



圖片設計

遭遇到的困難

參考文獻及感謝名單

設計理念



炸彈超人系列的遊戲，
從我們小時候就屬於非常經典的遊戲；
一直到今天，
雖然市面上大多是充斥各種絢麗特效的遊戲，
但例如「彈水阿給」、「爆爆王」等等類似
炸彈超人而且畫面簡單的網路遊戲，
人氣度卻不減反增。
藉此，我們想利用這次機會，
製作一個屬於我們自己的炸彈超人遊戲。



遊戲介紹

遊戲的背景是在西元9002年，主角是一個型號為QQ-B96076的棒波麵機器人（BOMBER MAN）。這一年的棒波界並不和平，多年前與QQ棒波麵有著深仇大恨的CC棒波麵，竟然又東山再起？為了平息這場風波、為了打敗CC棒波麵在棒波界的邪惡勢力，扛著八代嫡傳QQ波棒麵的名號。QQ-B96076將要帶著他的炸彈，踏上他冒險的旅程。



操作介紹

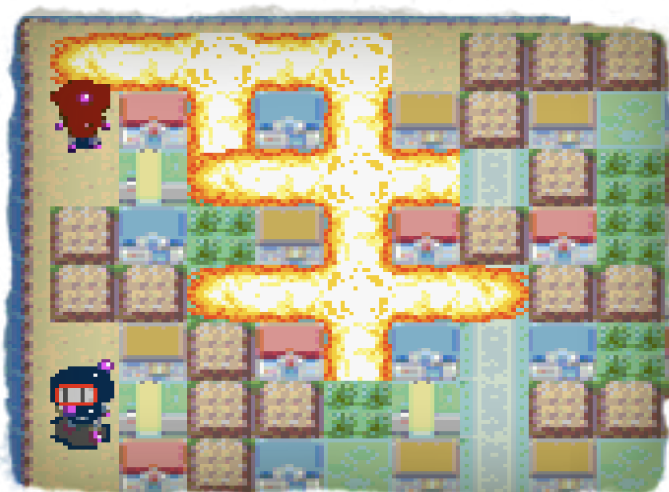
↑↓←→ 移動遊戲人物
A(鍵盤上的Z) 放置炸彈

特殊道具

遊戲中有兩種道具

第一種吃到可以增加放炸彈的數量

第二種吃到可以增加炸藥的火力



獲勝條件

將所有敵人用炸彈全數殲滅
但若不慎被自己放置的炸彈炸到
或是被敵方抓到
遊戲就 **GAME OVER**

實作內容

顯示模式：

我們採用 mode 1 來顯示畫面，背景的部份使用 bitmap，炸彈超人、炸彈以及可以被炸掉的箱子和箱子裡面的道具都是使用 sprite。

前景物件：

以下幾種遊戲中出現的人物或物品是用 sprite：

<code>sprite_bomber</code>	炸彈超人的圖片
<code>sprite_bomber_bad</code>	敵人的圖片
<code>box_pic1</code>	第一關可以炸的箱子
<code>box_pic2</code>	第二關可以炸的箱子
<code>min_gia_pic1</code>	道具
<code>min_gia_pic2</code>	道具
<code>bomb_pic</code>	炸彈
<code>fire_mid(row,col,up,down,left,right)</code>	火焰

上述物件在程式碼裡面大多是用 structure 來代表。

並且在 structure 裡面給予該物件的編號以及在螢幕上顯示的位置。

程式碼：以下列出 function 與其功能

`void vbl_func()`

每1/60秒更新一次此function的內容。

`void query_buttons()`

每當按下按鈕的時候，所應該對應的動作。

包括人物的移動（上下左右），以及放炸彈。

`void update_man_pos()`

根據玩家角色的位置，更新人物在畫面上的位置。

`void update_man_gfx()`

根據玩家角色方向的參數，更新人物在畫面上的出現的圖片。

可以讓人物看起來像真的在跑步。

`void update_bad_pos(int i)`

與`update_man_pos()`同功能，改變敵方角色的位置。

`void update_bad_gfx(int i)`

與`update_man_gfx()`同功能，改變敵方角色的圖片。

void check_deadline()

檢查炸彈是否已經爆炸，並且加入連鎖爆炸的功能。

void update_bomblist(struct bomb *b)

檢查炸彈的linked list，確保此linked list正確

void update_firelist(struct fire *b)

檢查火焰的linked list，確保此linked list正確

void blast_fire(struct fire *f)

當炸彈爆炸的時候，往四面發射出火焰。

並且確保障礙物跟邊界可以擋住火焰。

void init_map()

在遊戲開始或者接關的時候，初始mark[10][14]裡面的值。

mark[][]的內容為該格子的狀態（牆壁、箱子、可走的道路、炸彈）

此函式以及陣列存放在 map.h

int right_pos(int x, int y, int dir)

在人物移動前，火焰生成時，檢查位置是否超過邊界。

int bongbong(int x, int y, int dir)

在人物移動前，火焰生成時，檢查欲移動處是否為可以走的道路。

void delete_fire()

當炸彈爆炸之後，將四周的火焰消除掉。

void init_bad()

初始化敵人的參數。

void move_bad()

此函數令所有敵人可隨機移動。

void live_or_die()

判斷玩家角色目前狀態（正常遊戲或Game Over）。

int bomb_X(int x, int y)

微調放置炸彈的位置，使其在規定的格子上

int bomb_Y(int x, int y)

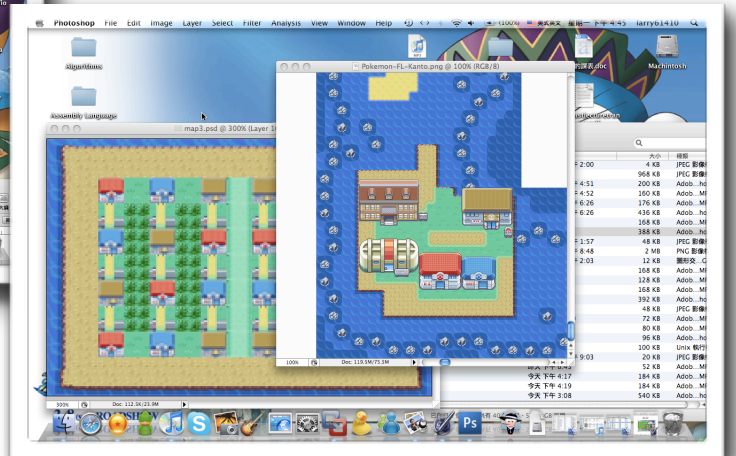
微調放置炸彈的位置，使其在規定的格子上

圖片設計

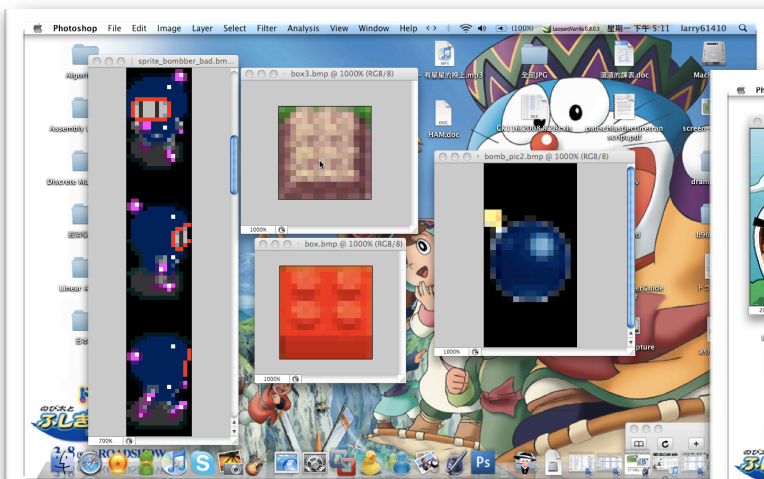
首先是遊戲地圖，我們總共製作了三張地圖，挑出了其中兩張作為最後的遊戲關卡。



上下邊界8 pixel，左右邊界16 pixel
中間是13×9的（16×16 pixel）格子。



- 樹木、海域
當作整個地圖的邊界
- 綠樂高、柏油路草地、漠地
當作可以行走的道路
- 三色房子
當作固定的障礙物，可以用來躲炸彈
- 紅樂高、土堆
當作可以炸開的暫時障礙物



再來是遊戲人物部份，玩家控制的角色是紅色的炸彈超人，對抗的敵人是藍色的炸彈超人。
為了使人物移動較為連續和生動，四個方向各有四張不同動作的圖。
人物長相是依照炸彈超人原作的模型來修改。

最後是幾個遊戲開始、勝利、失敗的畫面。我們是利用組員楊詠翔的相片來作為主題。
不管是勝利還是失敗他都會出現，鼓勵你繼續加油喔。

遭遇到的困難

1. 調色盤的使用

一開始沒辦法顯示全部人物的顏色,調色盤不知道怎麼正確使用
發現應該將所有檔案一次匯入調色盤中,而非分別處理各個圖檔。

2. 無法順利切換背景

因為函式用法都不太熟悉,在背景的載入與切換上遇到了一些困難,不斷嚐試後才發現正確的使用方法。

3. 放炸彈的速度

無法控制放炸彈的速度,很容易按一次A鍵後就會在一格連續放無數顆炸彈,造成記憶體不足,後來運用一個變數的改變來控制不能在短時間放多顆炸彈才加以解決。

4. 人物的移動與障礙物

因為要配合背景地圖上的障礙物,所以必須使得有些地方人物是無法到達的,但一開始的時候一直無法順利的讓人物被障礙物阻擋,後來增加了一些判斷人物位置的函式,才能順利行走在正確的道路上。

5. 炸彈的放置

一開始我們使用的函式檢查放置位置比較不嚴謹,所以常常會放置在人物左邊的格子上,造成遊戲中很容易死亡,後來增寬放炸彈位置的限制,讓人就算站在那一格稍微偏左或稍微偏右時,都可以將炸彈放置在正確的位置。

6. 爆炸後的火焰

地圖上固定的障礙物應該是火焰無法通過的,但一開始火燄沒辦法因為碰到障礙物而不產生,所以我們增加一個函式來判定這格是否已經有障礙物,來決定爆炸後的火焰是否應該生成。

7. 連鎖爆炸

遊戲中很重要的一點就是炸彈會連鎖爆炸,我們使用的是linked list來儲存每一顆炸彈,但我們所寫出來檢查的函式只要爆炸順序稍有不正確,就會造成炸彈被切斷而遺留在地圖中,後來我們增加了一個檢查的判斷式,來解決這個問題。

8. 道具的使用

為了更進一步實現炸彈超人的功能，所以我們增加了道具的功能，但在要讓箱子消失同時讓道具出現，以及吃到道具之後道具便要消失上遇到了一些困難，程式運行始終有錯誤，無法順利判斷道具是否已經被吃掉，後來增加一個陣列來儲存道具現在的狀態來改善這個問題。

9. 敵人的行動

一開始我們的敵人只會沿著道路行走，直到碰到牆壁才轉向，這樣使得有許多地點會無法走到，後來我們增加了一個亂數來不定時決定是否轉向，讓敵人會移動的範圍更加不固定增加遊戲的難度。

參考文獻

HAM TUTORIAL BY [AARON ROGERS](http://www.aaronrogers.com/ham)

[HTTP://WWW.AARONROGERS.COM/HAM](http://www.aaronrogers.com/ham)

HAM DOCUMENTATION

[HTTP://WWW.ENGINE.DE/HAMDOC/INDEX.HTML](http://www.ngine.de/hamdoc/index.html)

特別感謝

CYY

ADOBE PHOTOSHOP CS3

COREL ULEAD PHOTOIMPACT X3

VISUAL HAM

GFX2GBA