

# 目錄

一、	簡介.....	- 1 -
1.	動機.....	- 1 -
2.	分工.....	- 1 -
二、	遊戲介紹.....	- 2 -
1.	遊戲說明.....	- 2 -
a.	遊戲內容.....	- 2 -
b.	遊戲操作.....	- 2 -
c.	遊戲劇情.....	- 3 -
d.	角色介紹.....	- 3 -
2.	遊戲圖形.....	- 5 -
3.	遊戲音效.....	- 9 -
三、	程式設計.....	- 10 -
1.	程式架構.....	- 10 -
2.	程式類別.....	- 11 -
3.	程式技術.....	- 12 -
四、	結語.....	- 13 -
1.	問題及解決方法.....	- 13 -
2.	時間表.....	- 15 -
3.	貢獻比例.....	- 16 -
4.	檢核表.....	- 16 -
5.	收穫.....	- 17 -
6.	心得、感想.....	- 18 -
7.	對於本課程的建議.....	- 19 -
五、	附錄.....	- 21 -
	Mygame.h .....	- 21 -
	Mygame.cpp.....	- 24 -
	Kirby.h .....	- 41 -
	Kirby.cpp.....	- 44 -
	Enemy.h .....	- 79 -
	Enemy.cpp.....	- 80 -
	Map.h .....	- 84 -
	Map.cpp .....	- 85 -
	Door.h.....	- 90 -
	Door.cpp.....	- 91 -

## 一、 簡介

### 1. 動機

經過期初的討論，從眾多的遊戲中，我們想到卡比之星這部由任天堂出品的經典作品，其中鏡之大迷宮這個系列陪伴我們度過不少的童年時光，對這部作品很熟悉也很有興趣，起初是抱著嘗試看看的想法，沒想到這個主題非常適合老師的 framework，便以此做為物件導向實習的主題。

### 2. 分工

每個部份我們都有分配負責人，但非完全由個人完成，是經過兩人多次討論、不斷修正的結果。

以下列出兩人負責的項目：

周宇天：地圖系統、閃電怪、火焰怪、樹王、鏡子碎片、開始畫面。

林彥廷：卡比、閃電卡比、火焰卡比、飛機王、音效、操作說明。

## 二、 遊戲介紹

### 1. 遊戲說明

#### a. 遊戲內容

此遊戲是模仿星之卡比鏡之大迷宮，以練習物件導向程式設計為目的，利用一學期的時間撰寫，達到實作與理論並重的課程練習，我們使用陳偉凱老師提供的 framework 為架構，加上兩人合力完成的 Kirby、Map 等等物件，經過多次的討論與修正，最終完成整個遊戲。

#### b. 遊戲操作

↑：吸氣飛翔、進入傳送門

↓：蹲下、變身

←：左移

→：右移

Z：吐氣、攻擊、吸怪

X：跳躍

S：血量補滿

↓ + Z：踢擊

→ + C：向右跑步

← + C：向左跑步

c. 遊戲劇情

此遊戲包含一張主地圖、兩張普通地圖與兩張魔王地圖，主地圖用來顯示鏡子碎片的收集狀況與連通各個地圖的傳送門；普通地圖包含各種小怪，提供玩家練習操作與變身，兩張普通地圖有各自的主題，一張是洞窟，另一張是凹凸草原，讓玩家能體驗不同的遊戲風景；最後是魔王地圖，這是全遊戲的關鍵，需要打敗兩隻魔王奪回鏡子碎片才能過關。

d. 角色介紹

遊戲裡包含兩隻魔王，分別是塔王與飛機王，塔王的地圖中會隨機落下石塊攻擊卡比，但卡比也能利用吸入落下的石塊，吐出星星攻擊塔王，算是比較容易攻破的魔王；飛機王是相對比較困難的，它會在地圖中上下飛行，讓玩家不好攻擊，撞到卡比也會讓卡比受到傷害，更恐怖的是它會發射飛彈，不僅僅是碰到飛彈會受傷，更要小心它的爆炸範圍，建議玩家多利用空氣砲來攻擊，即便是變身過後，飛機王也不是個好對付的對手。

小怪部分有普通的小怪、閃電怪和火焰怪，普通的小怪會在地圖中來回走動，用踢擊可以近距離將它踢死，玩家可以利用它來練習操作，吃掉它是不能變身的喔！閃電怪會追蹤卡比，並且發出電場攻擊卡比，值得注意的是它會跳躍，不是跳得高高的就能高枕無憂唷，如此強大的對手與其殺掉它不如把它吃了，變身成閃電卡比吧！火焰怪會噴出火焰，讓玩家很難靠近它，但它也不能永無止盡的噴下去，把握它的攻擊間隔時間打敗它吧！當然你也能選擇吃掉它變成華麗的火焰卡比，享受火烤怪物的樂趣，火焰怪也會追蹤卡比，沒有錯我們的怪獸都是有智慧的，為了考驗玩家們的智商，太無腦是沒辦法過關的喔！。

卡比總共有三種型態，普通卡比、閃電卡比、火焰卡比，攻擊部分，普通卡比可以將怪物吸進嘴裡，吐出星星攻擊或是吞下去變身，也可以把飛行吸入的氣體吐出，作為攻擊的手段喔；閃電卡比是發出強大電場，電擊範圍內的怪物，火焰卡比則是吐出熊熊烈火，火烤就是美味。各種型態的卡比都是能使用踢擊的，玩家可以每一種都是嘗試看看，找出屬於自己的卡比吧！

## 2. 遊戲圖形

開始畫面	說明
	<p>操作：</p> <ul style="list-style-type: none"> <li>↑: 吸氣飛翔、進入傳送門 ↓: 蹲下、變身</li> <li>←: 左移 →: 右移</li> <li>Z: 攻擊、吸怪 ↓+Z: 踢擊</li> <li>X: 跳躍</li> <li>C: 跑步(按住加左右移)</li> <li>Esc: 關閉遊戲</li> </ul> <p>密技：</p> <ul style="list-style-type: none"> <li>S: 血量補滿</li> </ul> <p>打敗兩隻魔王，拿回碎片拯救世界吧！卡比 (按下空白鍵開始)</p>
卡比	
	
行走動畫	
	
跑步動畫	
	
翻滾動畫	
	
飛行動畫	
	
跳躍	落下
	
吐氣&星星	踢擊
	
壓扁	

吸氣&吸怪動畫



吸怪後的行走動畫



火焰卡比



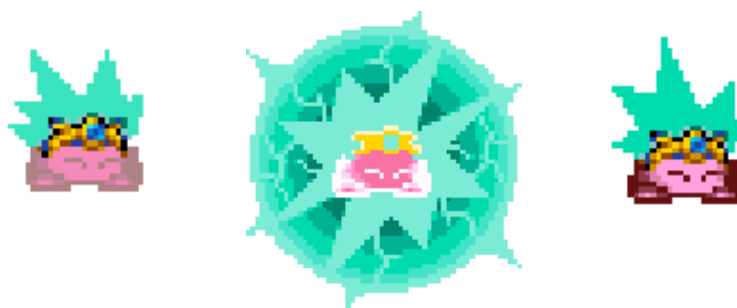
火焰卡比攻擊動畫



閃電卡比



閃電卡比攻擊動畫



普通小怪行走動畫



普通小怪被吸



火焰怪行走動畫



火焰怪攻擊動畫



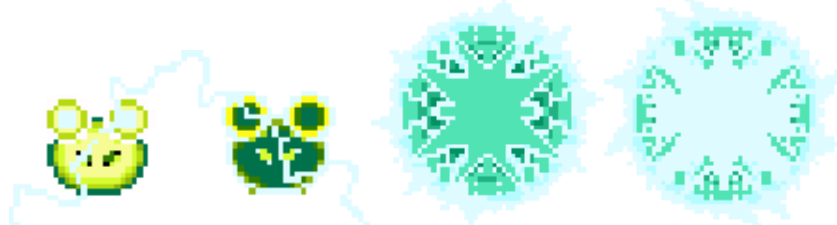
火焰怪被吸動畫



閃電怪跳躍動畫



閃電怪攻擊動畫



閃電怪被吸動畫

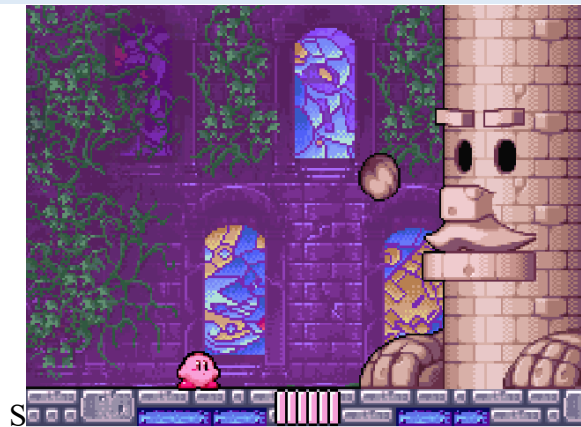




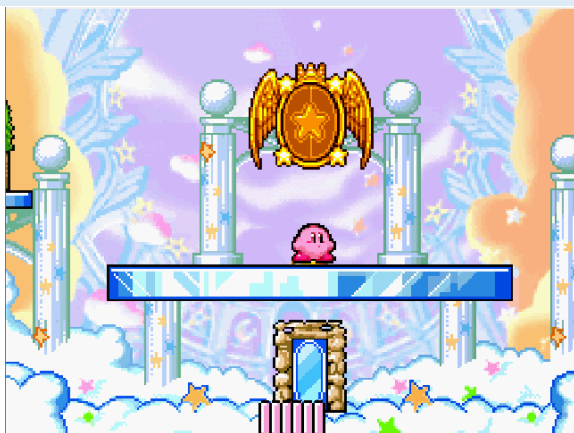
飛機王飛行動畫



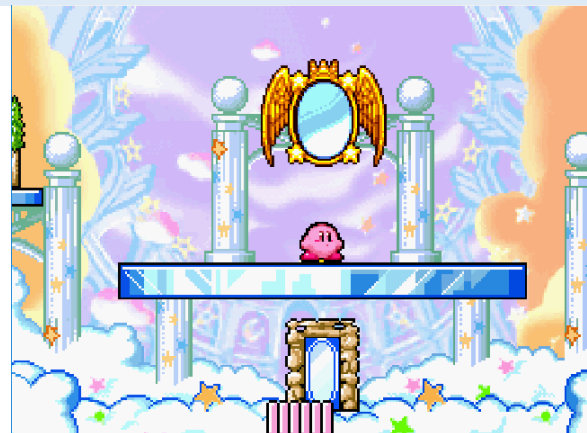
塔王



初始地圖



破關後的地圖



過關動畫

恭喜過關！

製作者 Producers：

林彥廷 Lin,Yan-Ting

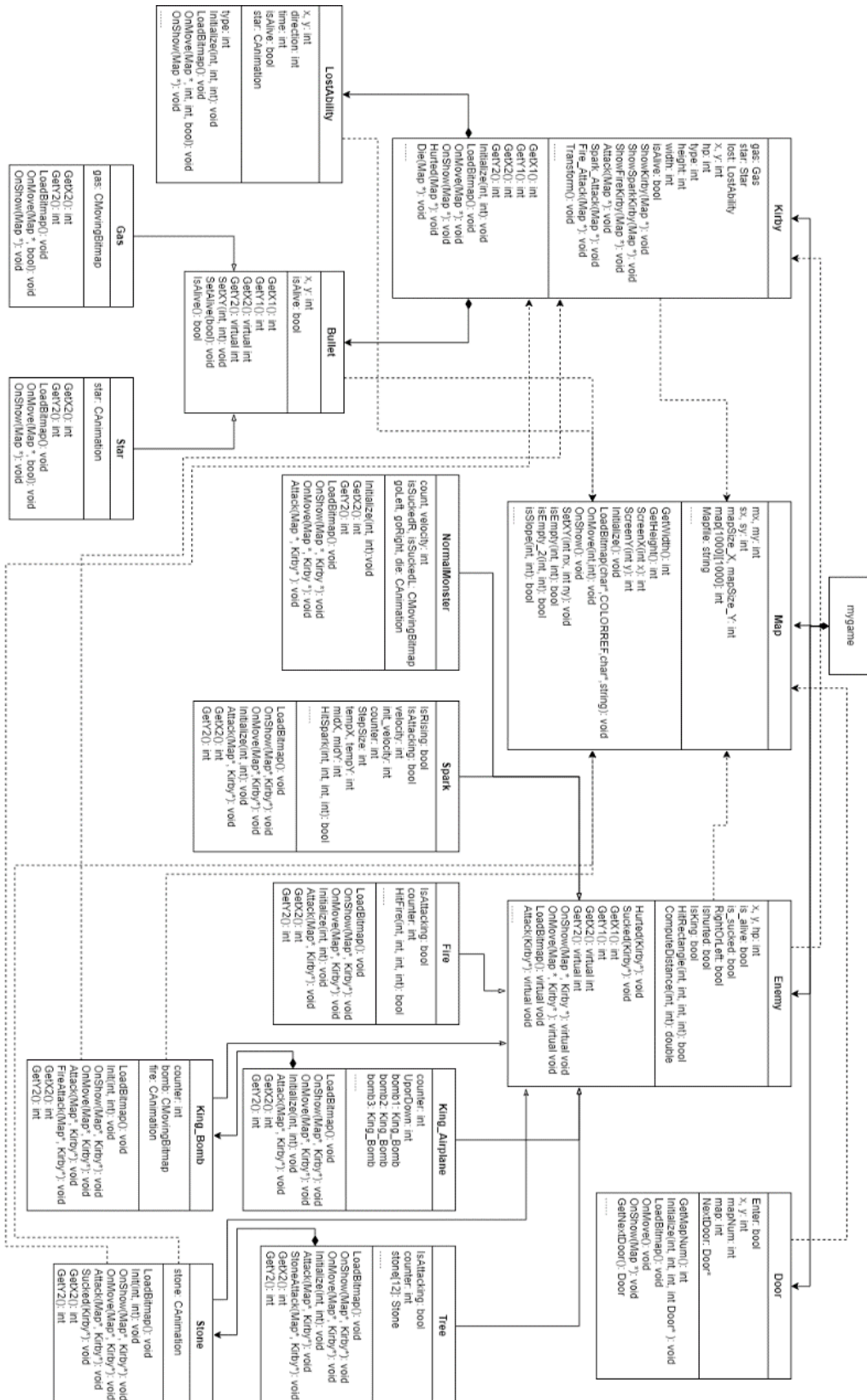
周宇天 Chou,You-Tian

## 3. 遊戲音效

音效	說明
die.wav	卡比死亡音效
ending.mp3	結束動畫背景音樂
enemyDie.wav	怪物死亡音效
fire.wav	火焰攻擊音效
fly.wav	飛行音效
gas.wav	吐氣音效
hurt.wav	受到傷害的音效
jump.wav	跳躍音效
kick.wav	踢擊音效
Kirby_background.mp3	遊戲時的背景音樂
landing.wav	落地音效
lostAbility.wav	失去能力時的音效
run.wav	跑步音效
spark.wav	閃電攻擊音效
star.wav	星星撞擊時的音效
start.mp3	開始畫面的背景音樂
suck.wav	吸怪時的音效
swallow.wav	吞怪時的音效

### 三、 程式設計

#### 1. 程式架構



## 2. 程式類別

類別名稱	.h 檔行數	.cpp 檔行數	說明
<b>mygame</b>	106	658	整個遊戲流程
<b>Bullet</b>	19	49	發射物的基礎內容(父類別)
<b>Door</b>	43	118	傳送門
<b>Enemy</b>	54	137	怪物的基本內容(父類別)
<b>Fire</b>	28	235	火焰怪，繼承 Enemy
<b>Gas</b>	18	48	空氣砲，繼承 Bullet
<b>King_Airplane</b>	26	175	飛機王，繼承 Enemy
<b>King_Bomb</b>	24	87	飛機王的炸彈，繼承 Enemy
<b>King_Stone</b>	23	100	塔王的石頭，繼承 Enemy
<b>Kirby</b>	133	1726	卡比的所有行為(變身、攻擊...)
<b>LostAbility</b>	46	113	卡比受傷時失去的變身能力
<b>Map</b>	39	197	地圖
<b>NormalMonster</b>	24	163	普通小怪，繼承 Enemy
<b>Spark</b>	35	261	閃電怪，繼承 Enemy
<b>Star</b>	17	54	卡比吐出的星星，繼承 Bullet
<b>Tree</b>	32	128	塔王，繼承 Enemy

### 3. 程式技術

我們的遊戲由 Kirby、Enemy、Map 這三個主要的物件交織而成，運用不少繼承與多型來擴充程式功能，並針對需求增加新物件供主要的物件使用，各功能主要由大量的判斷式與迴圈來完成，對於每種情況都進行嚴格的檢測。

Kirby 部分，原先的計畫是將各種卡比切割成多個物件，但考慮到建構、解構與資源佔用的問題，決定將卡比結合在一個物件，並新增 LostAbility、Bullet 兩個物件，還有 Gas 與 Star，兩者繼承 Bullet 並運用多型形成不同的攻擊方式與傷害。

Enemy 是每隻怪物的基礎，包含兩隻王與它們丟出的石頭或飛彈，同樣運用繼承與多型，每隻怪物在基礎的 function 中再延伸出各自的功能，並且可以透過一個 Enemy 指標陣列進行統一控管。

Map 是貫穿整個遊戲的物件，與地圖或是位子相關的判斷都透過 Map 來完成，運用 1000\*1000 的陣列儲存地圖的地形資訊，可以將地形資訊設定到幾乎吻合我們的地圖，此外我們有設計一個設定地形資訊的 function，用滑鼠點擊遊戲畫面就能設定，並且將地形資訊儲存到各自的文件檔，切換地圖時就能馬上載入新的地形資訊，每張地圖透過 Door 物件連通，Door 物件將地圖做轉換，並實作轉場動畫讓地圖轉換更流暢，不會顯得突兀。

## 四、 結語

### 1. 問題及解決方法

#### (1)斜坡的判斷式不夠精準

解決方法：由於地圖系統是多個正方形組成的關係，沒辦法很精準地完成斜坡功能，起初試了許多種判斷式，都沒有辦法克服，最終只好重新規劃地形資料，以達到幾乎精準的斜坡判定。

#### (2)不知道是否該將不同型態的卡比切割成多個物件

解決方法：經過仔細思考之後，我們決定不切割卡比，一方面避免建構與解構時發生問題，另一方面也可以減少不必要的程式複雜度，因為卡比本來就是單一的物件，即便他變身了還是卡比。

#### (3)動態配置的結構不穩定

解決方案：改用靜態的陣列處理動態配置的物件，雖然耗費效能，但能有效解決動態配置不穩定的問題。

#### (4)音效重複讀取或讀取太慢導致播放時跳例外

解決方案：充分了解 framework 功能後，修改 CAudio 中的部分程式碼，以符合我們自己程式的需求。

#### (5)企劃不夠周詳

解決方案：針對負責的項目與完成先後順序做更詳細的討論，避免彼此衝突或是資訊不流通的狀況。

(6)程式撰寫規劃不足

解決方案：在開始設計程式前先進行功能規劃，以免更改程式時造成程式邏輯的前後矛盾。

(7)大量檔案的引入

解決辦法：一開始對圖片的引入是手動在 resource 裡建立新的點陣圖，但在後期處理大量圖檔時太過耗時且無法直接對圖檔進行更動，後來改以圖檔的相對位置開啟。

(8)素材來源格式不一

解決辦法：一開始的人物與地圖的比例非常奇怪，後來篩選圖片的來源並對圖檔統一放大。

## 2. 時間表

週	每週工作時間(小時)		總和	工作規劃
	周宇天	林彥廷		
0	3	3	6	Tutorial
1	4	14	18	完成地圖
2	5.5	14	19.5	角色在地圖上移動
3	4.5	2	6.5	怪物與卡比的互動
4	6.5	11	17.5	攻擊怪物與被攻擊
5	5	2	7	完成傳送門
6	3.5	4	7.5	吃怪、吐星星、吐氣....
7	7.5	7	14.5	新增怪物與卡比變身
8	5	7	12	第二關卡完成
9	8.5	9	17.5	變身火焰卡比與魔王關卡完成
10	2	3	5	新增音效
11	3	5	8	增加關卡、關卡難度與複雜度
12	4	6.5	10.5	製作過關動畫&通關方式
13	3	5	8	製作報告
14	10	5	15	製作報告
總和	75	97.5	172.5	





### 3. 貢獻比例

林彥廷：50%

周宇天：50%

### 4. 檢核表

項目	完成與否
1 解決 Memory leak	<input checked="" type="checkbox"/> 已完成 <input type="checkbox"/> 未完成
2 自定遊戲 Icon	<input checked="" type="checkbox"/> 已完成 <input type="checkbox"/> 未完成
3 全螢幕啟動	<input checked="" type="checkbox"/> 已完成 <input type="checkbox"/> 未完成
4 修改 Help->About	<input checked="" type="checkbox"/> 已完成 <input type="checkbox"/> 未完成
5 初始畫面說明按鍵及滑鼠之用法與密技	<input checked="" type="checkbox"/> 已完成 <input type="checkbox"/> 未完成
6 上傳 setup 檔	<input checked="" type="checkbox"/> 已完成 <input type="checkbox"/> 未完成
7 報告字型、點數、對齊、行距、頁碼等格式正確	<input checked="" type="checkbox"/> 已完成 <input type="checkbox"/> 未完成
8 報告封面、側邊格式正確	<input checked="" type="checkbox"/> 已完成 <input type="checkbox"/> 未完成

## 5. 收穫

林彥廷：

經過這一個學期的歷練，讓我對物件導向的概念有更多體悟，也有更多的程式撰寫經歷，對於程式的架構也有更深入的了解，由於這是第一次寫這種較大型的專案，有了這次經驗以後，相信再面對這類型的專案時，我就不會太畏懼，也能更有自信的估計耗費的時間或資源，此外這也是我第一次運用這麼多繼承與多型，經過多次失敗與搜尋，讓我更清楚如何正確地使用繼承與多型，來簡化複雜的物件關係或豐富程式功能，本次課程讓我學到最多的是團隊合作，有過各式各樣的合作經驗，合作完成一個較大型的專案倒是第一次，了解到合作完成一個程式的所有過程，也更知道如何和隊友清楚表達自己的程式或需求，表達正是我所欠缺的能力，這是個珍貴的練習經驗。

周宇天：

這學期的實作再次加深的我對物件導向的觀念與能力，專案裡大量使用了繼承與多型，不再像以前是以練習語法為目的的作業，這次我們扎扎實實的練習了編寫大型專案的感覺與技巧。不僅如此，透過大量閱讀別人的 code 了解程式的脈絡與結構，讓我有能力對程式進行優化與改寫，以別人的程式為基礎做出自己需要的功能。最重要的是在這個專案學會了與夥伴溝通合作，單憑一個人是很難做出一個完善的作品，不單單是因為每個人擅長的方面不同，憑一己之力是很難做出像樣的作品，我們都站在巨人的肩膀上，使用了別人的心血結晶，使用了老師的 framework 為基礎才能完成。

## 6. 心得、感想

林彥廷：

實作的過程中經歷了許多挫折與失敗，一次又一次的修改，一遍又一遍的尋找資料，正所謂失敗為成功之母，每次挫敗都讓我學到更多，像是繼承與多型的部分，就讓我收穫很多，但有些部分直到最後還是沒有辦法完成，像是動態生成一些非常駐的物件，起初空氣砲與星星的部分我是打算用動態完成，但卡在一些觀念不完整，且時間也不足夠，最後只好用比較不理想的方法完成這些功能，這是讓我比較遺憾的，但也讓我更清楚自己有哪部分需要加強，整體來說，這個專案我很滿意，雖然成品還是有些小缺失，但整個過程讓我成長許多，程式邏輯的思緒也更清晰，表達能力也有加強，感謝辛苦的老師與助教，帶我們完成這個專案，更感謝我的夥伴，讓我有這麼棒的合作經驗。

周宇天：

以前作業大部分都是自己一個人完成，但這次的實作讓我有機會能與夥伴合作一起完成專案，讓我初步認識該怎麼與人合作一同撰寫程式。在與其他人合作時我覺得最困難的地方在於溝通與閱讀，我們時常會無法理解對方需要的程式功能就是因為溝通不良所導致；而在編寫與維護程式時時常需要繼承或改寫別人的程式碼，如果註解與說明不夠清楚與明確將會造成很大的麻煩，這部分我應要改進，不要因為懶惰而造成夥伴的困擾。這學期我們從無到有做出了令自己滿意的作品，體驗了北科誠樸精勤實作精神。

感謝這學期教授與助教的用心教導

感謝夥伴的包容與忍耐

## 7. 對於本課程的建議

林彥廷：

希望在 Demo 的時候可以將畫面廣播到台下的電腦上，讓台下的同學能看得更清楚，如果 Android 不行，至少 Windows、HTML 的可以。

周宇天：

希望能讓 Demo 的流程與效率增加，以及讓 git 更穩定不然網站上的專案進度、編輯次數等功能就這麼白費了。



## 五、 附錄

### Mygame.h

```

1      #include "Kirby.h"
2      #include "Map.h"
3      #include "NormalMonster.h"
4      #include "Door.h"
5      #include "Gas.h"
6      #include "Star.h"
7      #include "LostAbility.h"
8      #include "Spark.h"
9      #include "Fire.h"
10     #include "Tree.h"
11     #include "King_Airplane.h"
12     namespace game_framework {
13         enum AUDIO_ID {                                // 定義各種音效的編號
14             AUDIO_BACKGROUND,                          // 0
15             jump,                                       // 1
16             landing,                                   // 2
17             kick,                                       // 3
18             die,                                        // 4
19             fly,                                        // 5
20             hurted,                                    // 6
21             run,                                        // 7
22             suck,                                       // 8
23             gasSound,                                  // 9
24             starSound,                                 // 10
25             swallow,                                   // 11
26             spark,                                     // 12
27             fire,                                       // 13
28             stone,                                     // 14
29             start,                                     // 15
30             enemyDie,                                  // 16
31             lostAbility,                               // 17
32             ending                                     // 18
33         };
34         class CGameStateInit : public CGameState {

```

```

35     public:
36         CGameStateInit(CGame *g);
37         void OnInit(); // 遊戲的初值及圖形設定
38         void OnBeginState(); // 設定每次重玩所需的變數
39         void OnKeyUp(UINT, UINT, UINT); // 處理鍵盤 Up 的動作
40         void OnLButtonDown(UINT nFlags, CPoint point); // 處理滑鼠的動作
41     protected:
42         void OnShow(); // 顯示這個狀態的遊戲畫面
43     private:
44         int page;
45         CAnimation StartAnimation;
46         CMovingBitmap background, pressStart;
47 };
48
49 class CGameStateRun : public CGameState {
50     public:
51         CGameStateRun(CGame *g);
52         ~CGameStateRun();
53         void OnBeginState(); // 設定每次重玩所需的變數
54         void OnInit(); // 遊戲的初值及圖形設定
55         void OnKeyDown(UINT, UINT, UINT);
56         void OnKeyUp(UINT, UINT, UINT);
57         void OnMouseMove(UINT nFlags, CPoint point); // 處理滑鼠的動作
58         void OnLButtonDown(UINT nFlags, CPoint point); // 處理滑鼠的動作
59         void OnLButtonUp(UINT nFlags, CPoint point); // 處理滑鼠的動作
60         void OnRButtonDown(UINT nFlags, CPoint point); // 處理滑鼠的動作
61         void OnRButtonUp(UINT nFlags, CPoint point); // 處理滑鼠的動作
62         void ResetMonster();
63     protected:
64         void OnMove(); // 移動遊戲元素
65         void OnShow(); // 顯示這個狀態的遊戲畫面
66     private:
67         Kirby kirby; // 卡比
68         Map map[5]; // 地圖
69         Map *index;
70         Enemy *monster[10]; // 怪物
71         Enemy *Boss;
72         NormalMonster normalMonster1[5], normalMonster4[1];

```

```

73         Spark spark1[4],spark4[1];
74         Fire fire1[2], fire4[1];
75         Tree tree;
76         King_Airplane airplane;
77         Door door[10];
78         Door door1[2];
79         Door door2;
80         Door door3;
81         Door door4[2];
82         Door *gate;
83         int mapNum; //設定現在為第幾號地圖
84         int Mirror_L_Y,Mirror_R_Y;
85         int end_Y;
86         CAnimation Transition;
87         CAnimation Mirror_L, Mirror_R;
88         CMovingBitmap end;
89         bool Istransiting = false;
90         bool MovingMirror = false;
91         bool Show_Mirror_L = false;
92         bool Show_Mirror_R = false;
93         bool isEnd = false;
94     };
95     class CGameStateOver : public CGameState {
96     public:
97         CGameStateOver(CGame *g);
98         void OnBeginState(); // 設定每次重玩所需的變數
99         void OnInit();
100    protected:
101        void OnMove(); // 移動遊戲元素
102        void OnShow(); // 顯示這個狀態的遊戲畫面
103    private:
104        int counter; // 倒數之計數器
105    };

```



**Mygame.cpp**

```

1  #include "stdafx.h"
2  #include "Resource.h"
3  #include <mmsystem.h>
4  #include <ddraw.h>
5  #include "audio.h"
6  #include "gamelib.h"
7  #include "mygame.h"
8
9  namespace game_framework {
10 CGameStateInit::CGameStateInit(CGame *g): CGameState(g)
11 {
12 }
13 void CGameStateInit::OnInit()
14 {
15     StartAnimation.AddBitmap(".\\RES\\Start\\title_0.bmp", RGB(255, 255, 255));
16     StartAnimation.AddBitmap(".\\RES\\Start\\title_1.bmp", RGB(255, 255, 255));
17     StartAnimation.AddBitmap(".\\RES\\Start\\title_2.bmp", RGB(255, 255, 255));
18     StartAnimation.AddBitmap(".\\RES\\Start\\title_3.bmp", RGB(255, 255, 255));
19     StartAnimation.AddBitmap(".\\RES\\Start\\title_4.bmp", RGB(255, 255, 255));
20     StartAnimation.AddBitmap(".\\RES\\Start\\title_5.bmp", RGB(255, 255, 255));
21     StartAnimation.AddBitmap(".\\RES\\Start\\title_6.bmp", RGB(255, 255, 255));
22     StartAnimation.AddBitmap(".\\RES\\Start\\title_7.bmp", RGB(255, 255, 255));
23     StartAnimation.AddBitmap(".\\RES\\Start\\title_8.bmp", RGB(255, 255, 255));
24     StartAnimation.AddBitmap(".\\RES\\Start\\title_9.bmp", RGB(255, 255, 255));
25     StartAnimation.AddBitmap(".\\RES\\Start\\title_10.bmp", RGB(255, 255, 255));
26     StartAnimation.AddBitmap(".\\RES\\Start\\title_11.bmp", RGB(255, 255, 255));
27     StartAnimation.AddBitmap(".\\RES\\Start\\title_12.bmp", RGB(255, 255, 255));
28     StartAnimation.AddBitmap(".\\RES\\Start\\title_13.bmp", RGB(255, 255, 255));
29     StartAnimation.AddBitmap(".\\RES\\Start\\title_14.bmp", RGB(255, 255, 255));
30     StartAnimation.AddBitmap(".\\RES\\Start\\title_15.bmp", RGB(255, 255, 255));
31     StartAnimation.AddBitmap(".\\RES\\Start\\title_16.bmp", RGB(255, 255, 255));
32     StartAnimation.AddBitmap(".\\RES\\Start\\title_17.bmp", RGB(255, 255, 255));
33     StartAnimation.AddBitmap(".\\RES\\Start\\title_18.bmp", RGB(255, 255, 255));
34     StartAnimation.AddBitmap(".\\RES\\Start\\title_19.bmp", RGB(255, 255, 255));
35     background.LoadBitmap(".\\RES\\Start\\background.bmp");
36     pressStart.LoadBitmap(".\\RES\\Start\\Press_start.bmp", RGB(248, 248, 248));
37 }

```

```

38 void CGameStateInit::OnBeginState()
39 {
40     page = 0;
41     if(!CAudio::Instance()->IsLoaded(start))
42         CAudio::Instance()->Load(start, "Sounds\\start.mp3");
43     CAudio::Instance()->Play(start, true);
44 }
45 void CGameStateInit::OnKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags)
46 {
47     const char KEY_ESC = 27;
48     const char KEY_SPACE = ' ';
49     if (nChar == KEY_SPACE)
50     {
51         if (page == 0)
52             page++;
53         else
54         {
55             CAudio::Instance()->Stop(start);
56             GotoGameState(GAME_STATE_RUN); // 切換至 GAME_STATE_RUN
57         }
58     }
59     else if (nChar == KEY_ESC) // Demo 關閉遊戲的方法
60         PostMessage(AfxGetMainWnd()->m_hWnd, WM_CLOSE, 0, 0); // 關閉遊戲
61 }
62 void CGameStateInit::OnLButtonDown(UINT nFlags, CPoint point)
63 {
64     if (page == 0)
65         page++;
66     else
67     {
68         CAudio::Instance()->Stop(start);
69         GotoGameState(GAME_STATE_RUN); // 切換至 GAME_STATE_RUN
70     }
71 }
72 void CGameStateInit::OnShow()
73 {
74     if (page == 0)
75     {

```

```

76         background.SetTopLeft(0, 0);
77         StartAnimation.SetTopLeft(0, 0);
78         pressStart.SetTopLeft(60, 380);
79         background.ShowBitmap();
80         StartAnimation.OnMove();
81         StartAnimation.OnShow();
82         pressStart.ShowBitmap();
83     }
84     else
85     {
86         //*Demo 螢幕字型的使用，不過開發時請盡量避免直接使用字型，改用 CMovingBitmap 比較好*/
87         CDC *pDC = CDDraw::GetBackCDC();           // 取得 Back Plain 的 CDC
88         CFont f,*fp;
89         f.CreatePointFont(160,"Times New Roman");    // 產生 font f; 160 表示 16 point 的字
90         fp=pDC->SelectObject(&f);                   // 選用 font f
91         pDC->SetBkColor(RGB(0, 0, 0));
92         pDC->SetTextColor(RGB(255, 255, 0));
93         pDC->TextOut(35, 50, "操作 :");
94         pDC->TextOut(65, 80, "↑ : 吸氣飛翔、進入傳送門   ↓ : 蹲下、變身");
95         pDC->TextOut(65, 110, "← : 左移   → : 右移");
96         pDC->TextOut(65, 140, "Z : 攻擊、吸怪   ↓ + Z : 踢擊");
97         pDC->TextOut(65, 170, "X : 跳躍");
98         pDC->TextOut(65, 200, "C : 跑步(按住加左右移)");
99         pDC->TextOut(65, 230, "Esc : 關閉遊戲");
100        pDC->TextOut(35, 260, "密技 :");
101        pDC->TextOut(65, 290, "S : 血量補滿");
102        pDC->TextOut(35, 350, "打敗兩隻魔王，拿回碎片拯救世界吧！卡比");
103        pDC->TextOut(220, 380, "(按下空白鍵開始)");
104        pDC->SelectObject(fp);                         // 放掉 font f (千萬不要漏了放掉)
105        CDDraw::ReleaseBackCDC();                     // 放掉 Back Plain 的 CDC*/
106    }
107 }
108 CGameStateOver::CGameStateOver(CGame *g): CGameState(g)
109 {
110 }
111 void CGameStateOver::OnMove()
112 {
113     if(counter==0)

```

```

114         GotoGameState(GAME_STATE_INIT);
115     }
116     void CGameStateOver::OnBeginState()
117     {
118         counter = 300;
119     }
120     void CGameStateOver::OnInit()
121     {
122     }
123     void CGameStateOver::OnShow()
124     {
125         counter--;
126         CDC *pDC = CDDraw::GetBackCDC();           // 取得 Back Plain 的 CDC
127         CFont f,*fp;
128         f.CreatePointFont(160,"Times New Roman");    // 產生 font f; 160 表示 16 point 的字
129         fp=pDC->SelectObject(&f);                   // 選用 font f
130         pDC->SetBkColor(RGB(0,0,0));
131         pDC->SetTextColor(RGB(255,255,0));
132         char str[80];                                // Demo 數字對字串的轉換
133         sprintf(str, "Game Over ! (%d)",counter/30);
134         pDC->TextOut(240,210,str);
135         pDC->SelectObject(fp);                         // 放掉 font f (千萬不要漏了放掉)
136         CDDraw::ReleaseBackCDC();                     // 放掉 Back Plain 的 CDC
137     }
138     CGameStateRun::CGameStateRun(CGame *g): CGameState(g)
139     {
140     }
141     CGameStateRun::~CGameStateRun()
142     {
143     }
144     void CGameStateRun::OnBeginState()
145     {
146         map[0].Initialize();
147         map[1].Initialize();
148         map[2].Initialize();
149         map[3].Initialize();
150         map[4].Initialize();
151

```

```

152     Mirror_L.SetTopLeft(608, -100);
153     Mirror_R.SetTopLeft(608, -100);
154     kirby.Initialize(640,400);
155     door[0].Initialize(123, 37, 1, 0, &door1[0]);
156     door[1].Initialize(621, 37, 2, 0, &door2);
157     door[2].Initialize(1118, 37, 4, 0, &door4[0]);
158     door[3].Initialize(83, 328, 4, 0, &door4[1]);
159     door[4].Initialize(248, 369, 4, 0, &door4[0]);
160     door[5].Initialize(993, 367, 1, 0, &door1[0]);
161     door[6].Initialize(1159, 326, 1, 0, &door1[1]);
162     door[7].Initialize(207, 575, 1, 0, &door1[1]);
163     door[8].Initialize(620, 614, 3, 0, &door3);
164     door[9].Initialize(1036, 574, 4, 0, &door4[1]);
165     door1[0].Initialize(30, 425, 0,1, &door[5]);
166     door1[1].Initialize(4450, 350, 0, 1, &door[6]);
167     door2.Initialize(320, 240, 0, 2, &door[1]);
168     door3.Initialize(150, 350, 0, 3, &door[8]);
169     door4[0].Initialize(48, 435, 0, 4, &door[4]);
170     door4[1].Initialize(3140, 240, 0, 4, &door[3]);
171     ResetMonster();
172     tree.Initialize(450, 100);
173     airplane.Initialize(450, 60);
174     Mirror_L_Y = -10;
175     Mirror_R_Y = -10;
176     end_Y = 0;
177     mapNum = 0;
178     index = &map[mapNum];
179     gate = &door[5];
180     CAudio::Instance()->Play(AUDIO_BACKGROUND, true);
181 }
182 void CGameStateRun::OnMove() // 移動遊戲元素
183 {
184     index->OnMove(kirby.GetX1(), kirby.GetY1());
185     Mirror_R.OnMove();
186     Mirror_L.OnMove();
187     if (Istransiting) {
188         Transition.OnMove();
189         Transition.SetDelayCount(4);

```

```
190     }
191     else if (MovingMirror) {
192         map[0].SetXY(320, 160);
193         if (Show_Mirror_R && Mirror_R_Y != 327) {
194             Mirror_R_Y += 1;
195             Mirror_R.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror_R_Y));
196         }
197         else if (Show_Mirror_L && Mirror_L_Y != 327) {
198             Mirror_L_Y += 1;
199             Mirror_L.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror_L_Y));
200         }
201         else {
202             MovingMirror = false;
203         }
204     }
205     else if (isEnd)
206     {
207         end_Y--;
208         if (end_Y == -1)
209         {
210             CAudio::Instance()->Stop(AUDIO_BACKGROUND);
211             CAudio::Instance()->Play(ending);
212         }
213         else if (end_Y == -1050)
214         {
215             CAudio::Instance()->Stop(ending);
216             CAudio::Instance()->Play(AUDIO_BACKGROUND);
217             isEnd = false;
218             end_Y = 1;
219         }
220     }
221     else {
222         if (kirby.IsAlive())
223             kirby.OnMove(index);
```

```

224     if (mapNum == 0)
225     {
226         for (int i = 0; i < 10; i++) {
227             door[i].OnMove();
228             if (door[i].IsEnter(&kirby)) {
229                 door[i].SetEnter(false);
230                 Istransiting = true;
231                 Transition.Reset();
232                 gate = &door[i];
233             }
234         }
235     }
236     if (mapNum == 1) {
237         for (int m = 0; m < 10; m++)
238             monster[m]->OnMove(index, &kirby);
239         for (int i = 0; i < 2; i++) {
240             door1[i].OnMove();
241             if (door1[i].IsEnter(&kirby)) {
242                 door1[i].SetEnter(false);
243                 Istransiting = true;
244                 Transition.Reset();
245                 gate = &door1[i];
246             }
247         }
248     }
249     if (mapNum == 2) {
250         airplane.OnMove(index,&kirby);
251         if (!airplane.IsAlive())
252         {
253             if (!Show_Mirror_R) {
254                 MovingMirror = true;
255                 Show_Mirror_R = true;
256                 Istransiting = true;
257                 Transition.Reset();
258                 Mirror_L.Reset();
259                 Mirror_R.Reset();
260             }
261             else {

```

```

262             door2.OnMove();
263             if (door2.IsEnter(&kirby)) {
264                 door2.SetEnter(false);
265                 Istransiting = true;
266                 Transition.Reset();
267                 gate = &door2;
268             }
269         }
270     }
271 }
272 if (mapNum == 3) {
273     tree.OnMove(index,&kirby);
274     if (!tree.IsAlive()) {
275         if (!Show_Mirror_L) {
276             MovingMirror = true;
277             Show_Mirror_L = true;
278             Istransiting = true;
279             Transition.Reset();
280         }
281         else {
282             door3.OnMove();
283             if (door3.IsEnter(&kirby)) {
284                 door3.SetEnter(false);
285                 Istransiting = true;
286                 Transition.Reset();
287                 gate = &door3;
288             }
289         }
290     }
291 }
292 if (mapNum == 4) {
293     door4[0].OnMove();
294     door4[1].OnMove();
295     for (int m = 0; m < 3; m++)
296         monster[m]->OnMove(index, &kirby);
297     for (int i = 0; i < 2; i++) {
298         if (door4[i].IsEnter(&kirby)) {
299             door4[i].SetEnter(false);

```



```

300             Istransiting = true;
301             Transition.Reset();
302             gate = &door4[i];
303         }
304     }
305 }
306 }
307 if (Transition.IsFinalBitmap()) {
308     Istransiting = false;
309 }
310 if (Transition.GetCurrentBitmapNumber() == 7 && !MovingMirror) {
311     mapNum = gate->GetMapNum();
312     index = &map[mapNum];
313     ResetMonster();
314     kirby.SetXY(gate->GetNextDoor().GetX() - 50, gate->GetNextDoor().GetY2()-kirby.GetHeight());
315 }
316 }
317 void CGameStateRun::OnInit()           // 遊戲的初值及圖形設定
318 {
319     Transition.AddBitmap("../Map//Transition_7.bmp", RGB(0, 0, 0));
320     Transition.AddBitmap("../Map//Transition_6.bmp", RGB(0, 0, 0));
321     Transition.AddBitmap("../Map//Transition_5.bmp", RGB(0, 0, 0));
322     Transition.AddBitmap("../Map//Transition_4.bmp", RGB(0, 0, 0));
323     Transition.AddBitmap("../Map//Transition_3.bmp", RGB(0, 0, 0));
324     Transition.AddBitmap("../Map//Transition_2.bmp", RGB(0, 0, 0));
325     Transition.AddBitmap("../Map//Transition_1.bmp", RGB(0, 0, 0));
326     Transition.AddBitmap("../Map//Transition_0.bmp", RGB(0, 0, 0));
327     Transition.AddBitmap("../Map//Transition_1.bmp", RGB(0, 0, 0));
328     Transition.AddBitmap("../Map//Transition_2.bmp", RGB(0, 0, 0));
329     Transition.AddBitmap("../Map//Transition_3.bmp", RGB(0, 0, 0));
330     Transition.AddBitmap("../Map//Transition_4.bmp", RGB(0, 0, 0));
331     Transition.AddBitmap("../Map//Transition_5.bmp", RGB(0, 0, 0));
332     Transition.AddBitmap("../Map//Transition_6.bmp", RGB(0, 0, 0));
333     Transition.AddBitmap("../Map//Transition_7.bmp", RGB(0, 0, 0));
334     Mirror_L.AddBitmap("../RES//Door//mirror_L_0.bmp", RGB(255, 255, 255));
335     Mirror_L.AddBitmap("../RES//Door//mirror_L_1.bmp", RGB(255, 255, 255));
336     Mirror_L.AddBitmap("../RES//Door//mirror_L_2.bmp", RGB(255, 255, 255));
337     Mirror_L.AddBitmap("../RES//Door//mirror_L_3.bmp", RGB(255, 255, 255));

```

```

338     Mirror_R.AddBitmap("../RES//Door//mirror_R_0.bmp", RGB(255, 255, 255));
339     Mirror_R.AddBitmap("../RES//Door//mirror_R_1.bmp", RGB(255, 255, 255));
340     Mirror_R.AddBitmap("../RES//Door//mirror_R_2.bmp", RGB(255, 255, 255));
341     Mirror_R.AddBitmap("../RES//Door//mirror_R_3.bmp", RGB(255, 255, 255));
342     end.LoadBitmap("../RES//end.bmp");
343     map[0].LoadBitmap("../Map//foreground.bmp", RGB(255, 255, 255), "../Map//background.bmp", "../Map//map.txt");
344     map[1].LoadBitmap("../Map//map1.bmp", RGB(255, 255, 255), "../Map//background_1.bmp", "../Map//map1.txt");
345     map[2].LoadBitmap("../Map//Boss_map.bmp", RGB(255, 255, 255), "../Map//background_2.bmp", "../Map//map2.txt");
346     map[3].LoadBitmap("../Map//King_foreground.bmp", RGB(255, 255, 255), "../Map//King_background.bmp", "../Map//m
347 ap3.txt");
348     map[4].LoadBitmap("../Map//foreground_4.bmp", RGB(255, 255, 255), "../Map//background_4.bmp", "../Map//map4.txt"
349 );
350     kirby.LoadBitmap();
351     monster[0] = &fire1[0];
352     monster[1] = &fire1[1];
353     monster[2] = &normalMonster1[0];
354     monster[3] = &normalMonster1[1];
355     monster[4] = &normalMonster1[2];
356     monster[5] = &normalMonster1[3];
357     monster[6] = &spark1[0];
358     monster[7] = &spark1[1];
359     monster[8] = &spark1[2];
360     monster[9] = &spark1[3];
361     fire4[0].LoadBitmap();
362     spark4[0].LoadBitmap();
363     normalMonster4[0].LoadBitmap();
364     for (int m = 0; m < 10; m++)
365         monster[m]->LoadBitmap();
366     for (int i = 0; i < 10; i++)
367         door[i].LoadBitmap();
368     door1[0].LoadBitmap();
369     door1[1].LoadBitmap();
370     door2.LoadBitmap();
371     door3.LoadBitmap();
372     for (int i = 0; i < 2; i++)
373         door4[i].LoadBitmap();
374     tree.LoadBitmap();
375     airplane.LoadBitmap();

```

```

376     CAudio::Instance()->Load(AUDIO_BACKGROUND, "Sounds\\Kirby_background.mp3"); //背景音樂
377     CAudio::Instance()->Load(jump, "Sounds\\jump.wav");
378     CAudio::Instance()->Load(landing, "Sounds\\landing.wav");
379     CAudio::Instance()->Load(kick, "Sounds\\kick.wav");
380     CAudio::Instance()->Load(die, "Sounds\\die.wav");
381     CAudio::Instance()->Load(fly, "Sounds\\fly.wav");
382     CAudio::Instance()->Load(hurted, "Sounds\\hurted.wav");
383     CAudio::Instance()->Load(run, "Sounds\\run.wav");
384     CAudio::Instance()->Load(suck, "Sounds\\suck.wav");
385     CAudio::Instance()->Load(gasSound, "Sounds\\gas.wav");
386     CAudio::Instance()->Load(starSound, "Sounds\\star.wav");
387     CAudio::Instance()->Load(swallow, "Sounds\\swallow.wav");
388     CAudio::Instance()->Load(spark, "Sounds\\spark.wav");
389     CAudio::Instance()->Load(fire, "Sounds\\fire.wav");
390     CAudio::Instance()->Load(enemyDie, "Sounds\\enemyDie.wav");
391     CAudio::Instance()->Load(lostAbility, "Sounds\\lostAbility.wav");
392     CAudio::Instance()->Load(ending, "Sounds\\ending.mp3");
393 }
394 void CGameStateRun::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
395 {
396     const char KEY_ESC = 27;      // keyboard Esc
397     const char KEY_LEFT = 0x25;   // keyboard 左箭頭
398     const char KEY_UP = 0x26;     // keyboard 上箭頭
399     const char KEY_RIGHT = 0x27;  // keyboard 右箭頭
400     const char KEY_DOWN = 0x28;   // keyboard 下箭頭
401     const char KEY_Jump = 0x58;    // keyboard X 鍵
402     const char KEY_Attack = 0x5A;  // keyboard Z 鍵
403     const char KEY_Run = 0x43;     // keyboard C 鍵
404     const char KEY_Restore = 0x53; // keyboard S 鍵
405     if (nChar == KEY_LEFT)
406         kirby.SetMovingLeft(true);
407     if (nChar == KEY_RIGHT)
408         kirby.SetMovingRight(true);
409     if (nChar == KEY_UP) {
410         kirby.SetMovingUp(true);
411         if(mapNum==0)
412             for (int i = 0; i < 10;i++)
413                 door[i].SetEnter(true);

```

```
414         if (mapNum == 1)
415             for (int i = 0; i < 2; i++)
416                 door1[i].SetEnter(true);
417         if (mapNum == 2)
418             door2.SetEnter(true);
419         if (mapNum == 3)
420             door3.SetEnter(true);
421         if (mapNum == 4)
422             for (int i = 0; i < 2; i++)
423                 door4[i].SetEnter(true);
424     }
425     if (nChar == KEY_DOWN)
426     {
427         kirby.SetMovingDown(true);
428         if (kirby.IsBig())
429             CAudio::Instance()->Play(swallow);
430     }
431     if (nChar == KEY_Attack)
432     {
433         kirby.SetAttack(true);
434         if (!MovingMirror)
435         {
436             if (kirby.IsFly())
437                 CAudio::Instance()->Play(gasSound);
438             else if (kirby.IsDown())
439                 CAudio::Instance()->Play(kick);
440             else if (kirby.GetType() == 0 && !kirby.IsBig())
441                 CAudio::Instance()->Play(suck);
442             else if (kirby.GetType() == 0 && kirby.IsBig())
443                 CAudio::Instance()->Play(starSound);
444             else if (kirby.GetType() == 1)
445                 CAudio::Instance()->Play(spark);
446             else if (kirby.GetType() == 2)
447                 CAudio::Instance()->Play(fire);
448         }
449     }
450     if (nChar == KEY_Run)
451     {
```

```

452         kirby.SetRun(true);
453         if (!kirby.IsFly() && !index->isEmpty((kirby.GetX1()+kirby.GetX2())/2, kirby.GetY2() + 1) && kirby.IsMove())
454             CAudio::Instance()->Play(run);
455     }
456     if (nChar == KEY_Jump && !kirby.IsFly() && (!index->isEmpty(kirby.GetX1(), kirby.GetY2() + 1)
457 || !index->isEmpty(kirby.GetX2(), kirby.GetY2() + 1))) //按下 X,卡比不是在飛行且落地才可跳躍
458         kirby.SetJump(true);
459     if (nChar == KEY_ESC)
460         PostMessage(AfxGetMainWnd()->m_hWnd, WM_CLOSE, 0, 0); // 關閉遊戲
461     if (nChar == KEY_Restore)
462         kirby.Restore();
463 }
464 void CGameStateRun::OnKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags)
465 {
466     const char KEY_LEFT = 0x25;    // keyboard 左箭頭
467     const char KEY_UP = 0x26;      // keyboard 上箭頭
468     const char KEY_RIGHT = 0x27;   // keyboard 右箭頭
469     const char KEY_DOWN = 0x28;    // keyboard 下箭頭
470     const char KEY_Attack = 0x5A;   // keyboard Z 鍵
471     const char KEY_Run = 0x43;      // keyboard C 鍵
472     if (nChar == KEY_LEFT)
473         kirby.SetMovingLeft(false);
474     if (nChar == KEY_RIGHT)
475         kirby.SetMovingRight(false);
476     if (nChar == KEY_UP) {
477         kirby.SetMovingUp(false);
478         if (mapNum==0)
479             for (int i = 0; i < 10; i++)
480                 door[i].SetEnter(false);
481         if (mapNum == 1)
482             for (int i = 0; i < 2; i++)
483                 door1[i].SetEnter(false);
484         if (mapNum == 2)
485             door2.SetEnter(false);
486         if (mapNum==3)
487             door3.SetEnter(false);
488         if (mapNum == 4)
489             for (int i = 0; i < 2; i++)

```

```

490             door4[i].SetEnter(false);
491         }
492         if (nChar == KEY_DOWN)
493             kirby.SetMovingDown(false);
494         if (nChar == KEY_Attack)
495         {
496             kirby.SetAttack(false);
497             kirby.SetSuck(false);
498             CAudio::Instance()->Stop(suck);
499             CAudio::Instance()->Stop(spark);
500             CAudio::Instance()->Stop(fire);
501         }
502         if (nChar == KEY_Run)
503             kirby.SetRun(false);
504     }
505     void CGameStateRun::OnMouseMove(UINT nFlags, CPoint point)    // 處理滑鼠的動作
506     {
507         // 沒事。如果需要處理滑鼠移動的話，寫 code 在這裡
508         index->SetMouse(point.x, point.y);
509     }
510     void CGameStateRun::OnLButtonDown(UINT nFlags, CPoint point) // 處理滑鼠的動作
511     {
512         index->IsLclick(true);
513     }
514     void CGameStateRun::OnLButtonUp(UINT nFlags, CPoint point) // 處理滑鼠的動作
515     {
516         index->IsLclick(false);
517     }
518     void CGameStateRun::OnRButtonDown(UINT nFlags, CPoint point) // 處理滑鼠的動作
519     {
520         index->IsRclick(true);
521     }
522     void CGameStateRun::OnRButtonUp(UINT nFlags, CPoint point) // 處理滑鼠的動作
523     {
524         index->IsRclick(false);
525     }
526     void CGameStateRun::ResetMonster()
527     {

```

```

528     if (mapNum == 1) {
529         normalMonster1[0].Initialize(947, 467);
530         normalMonster1[1].Initialize(2459, 517);
531         normalMonster1[2].Initialize(2903, 517);
532         normalMonster1[3].Initialize(3495, 59);
533         spark1[0].Initialize(675, 393);
534         spark1[1].Initialize(1673, 163);
535         spark1[2].Initialize(3247, 269);
536         spark1[3].Initialize(4045, 477);
537         fire1[0].Initialize(417, 467);
538         fire1[1].Initialize(3745, 477);
539         monster[0] = &fire1[0];
540         monster[1] = &fire1[1];
541         monster[2] = &normalMonster1[0];
542         monster[3] = &normalMonster1[1];
543         monster[4] = &normalMonster1[2];
544         monster[5] = &normalMonster1[3];
545         monster[6] = &spark1[0];
546         monster[7] = &spark1[1];
547         monster[8] = &spark1[2];
548         monster[9] = &spark1[3];
549     }
550     if (mapNum == 4) {
551         fire4[0].Initialize(410, 400);
552         spark4[0].Initialize(1125, 525);
553         normalMonster4[0].Initialize(1600, 525);
554         monster[0] = &fire4[0];
555         monster[1] = &spark4[0];
556         monster[2] = &normalMonster4[0];
557     }
558 }
559 void CGameStateRun::OnShow()
560 {
561     end.SetTopLeft(0, end_Y);
562     index->OnShow();
563     if (mapNum == 0) {
564         for (int i = 0; i < 10; i++)
565             door[i].OnShow(index);

```

```

566         if (Show_Mirror_L) {
567             Mirror_L.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror_L_Y));
568             Mirror_L.OnShow();
569         }
570         if (Show_Mirror_R) {
571             Mirror_R.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror_R_Y));
572             Mirror_R.OnShow();
573         }
574     }
575     else if (mapNum == 1) {
576         for (int i = 0; i < 2; i++)
577             door1[i].OnShow(index);
578         for (int m = 0; m < 10; m++)
579             monster[m]->OnShow(index, &kirby);
580     }
581     else if (mapNum == 2) {
582         airplane.OnShow(index,&kirby);
583         if(!airplane.IsAlive())
584             door2.OnShow(index);
585     }
586     else if (mapNum == 3) {
587         tree.OnShow(index,&kirby);
588         if (!tree.IsAlive()) {
589             door3.OnShow(index);
590         }
591     }
592     else if (mapNum == 4) {
593         for (int i = 0; i < 2; i++)
594             door4[i].OnShow(index);
595         for (int m = 0; m < 3; m++)
596             monster[m]->OnShow(index, &kirby);
597     }
598     if (kirby.IsAlive()) {
599         kirby.OnShow(index);
600     }
601     else if (kirby.GetY1() <= 1)
602     {
603         // 卡比死掉就不用播消失能力的音效

```



```

604         CAudio::Instance()->Stop(lostAbility);
605         // 鏡子重置
606         Show_Mirror_L = false;
607         Show_Mirror_R = false;
608         Mirror_L_Y = -10;
609         Mirror_R_Y = -10;
610         GotoGameState(GAME_STATE_OVER);
611     }
612     else
613         kirby.Die(index);
614     if (MovingMirror && Transition.GetCurrentBitmapNumber()>6) {
615         map[0].SetXY(320, 160);
616         map[0].OnShow();
617         if (Show_Mirror_L) {
618             Mirror_L.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror_L_Y));
619             Mirror_L.OnShow();
620         }
621         if (Show_Mirror_R) {
622             Mirror_R.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror_R_Y));
623             Mirror_R.OnShow();
624         }
625     }
626     if (Istransiting) {
627         Transition.OnShow();
628     }
629     if (Show_Mirror_L && Show_Mirror_R && Mirror_L_Y == 327 && Mirror_R_Y == 327 && end_Y == 0)
630         isEnd = true;
631     if (isEnd)
632         end.ShowBitmap();
633 }
634 }

```

## Kirby.h

```

1  #pragma once
2  #include "Map.h"
3  #include "Gas.h"
4  #include "Star.h"
5  #include "LostAbility.h"
6  namespace game_framework {
7      class Kirby
8      {
9      public:
10         Kirby();
11         int  GetX1();           // 左上角 x 座標
12         int  GetY1();           // 左上角 y 座標
13         int  GetX2();           // 右下角 x 座標
14         int  GetY2();           // 右下角 y 座標
15         int  GetType();         // 取得當前型態
16         int  GetHeight();       // 取得身高
17         int  GetWidth();        // 取得寬度
18         void Initialize(int ,int); // 設定為初始值
19         bool IsAlive();         // 是否活著
20         bool IsFly();           // 是否在飛
21         bool IsKick();          // 是否踢擊
22         bool IsSuck();          // 是否吸怪
23         bool IsAttack();        // 是否按 Z 鍵
24         bool IsRight();         // 面對方向
25         bool IsDown();          // 是否縮小
26         bool IsBig();           // 是否是大隻的
27         bool IsMove();          // 是否在移動
28         void LoadBitmap();      // 載入圖形
29         void OnMove(Map *m);    // 移動
30         void OnShow(Map *m);    // 將圖形貼到畫面
31         void Hurted(Map *m);    // 受傷
32         void Die(Map *m);       // 死去
33         void SetMovingDown(bool flag); // 設定是否正在往下移動
34         void SetMovingLeft(bool flag);  // 設定是否正在往左移動
35         void SetMovingRight(bool flag); // 設定是否正在往右移動

```

```

36         void SetMovingUp(bool flag);           // 設定是否正在往上移動
37         void SetJump(bool flag);               // 設定是否按下 X 鍵
38         void SetAttack(bool flag);             // 設定是否按下 Z 鍵
39         void SetSuck(bool flag);               // 設定是否吸怪
40         void SetBig(bool flag);                // 設定是否含滷蛋
41         void SetRun(bool);                     // 設定是否按下 C
42         void SetXY(int nx, int ny);            // 設定左上角的座標
43         void SetEat(int t);                     // 設定吃了什麼怪物
44         void Restore();                        // 密技:血回到滿
45         Gas* GetGas();
46         Star* GetStar();
47     protected:
48         CMovingBitmap blood6, blood5, blood4, blood3, blood2, blood1, blood0;
49         // 普通卡比
50         CMovingBitmap originR, originL, exhaleR, exhaleL, jumpR, jumpL, downR, downL, landingR, landingL,
51         downAttackR, downAttackL, GG;
52         CAnimation goL, goR, flyR, prepareFlyR, flyL, prepareFlyL, hurtedL, hurtedR, runL, runR, suckR, suckL;
53         // 含東西卡比
54         CMovingBitmap bigOriginR, bigOriginL, bigJumpR, bigJumpL, bigLandingR, bigLandingL, threwR, threwL;
55         CAnimation bigGoL, bigGoR, swallowR, swallowL;
56         // 雷電卡比
57         CMovingBitmap Spark_exhaleR, Spark_exhaleL;
58         CAnimation Spark_originR, Spark_originL, Spark_downR, Spark_downL, Spark_goR, Spark_goL,
59         Spark_jumpR, Spark_jumpL, Spark_landingR, Spark_landingL, Spark_downAttackR, Spark_downAttackL, Spark_flyR,
60         Spark_prepareFlyR, Spark_flyL, Spark_prepareFlyL, Spark_runR, Spark_runL, Spark_attackR, Spark_attackL;
61         // 火焰卡比
62         CMovingBitmap fire_exhaleR, fire_exhaleL;
63         CAnimation fire_originR, fire_originL, fire_downR, fire_downL, fire_goR, fire_goL, fire_jumpR, fire_jumpL,
64         fire_landingR, fire_landingL, fire_downAttackR, fire_downAttackL, fire_flyR, fire_prepareFlyR, fire_flyL, fire_prepareFlyL,
65         fire_runR, fire_runL, fire_attackR, fire_attackL, attack1, attack2, attack3;
66         Gas gas;
67         Star star;
68         LostAbility lost;
69         int x, y;                               // 左上角座標
70         bool isMovingDown;                      // 是否正在往下移動
71         bool isMovingLeft;                     // 是否正在往左移動
72         bool isMovingRight;                    // 是否正在往右移動
73         bool isMovingUp;                       // 是否正在往上移動

```

```

74         bool isExhale;           // 是否吐氣
75         bool isJump;             // 是否按下 X 鍵
76         bool isAttack;           // 是否按下 Z 鍵
77         bool isKick;             // 是否使用踢擊
78         bool isSuck;             // 是否使用吸怪
79         bool isHurted;           // 是否被攻擊
80         bool isAlive;            // 是否活著
81         bool isFly;              // 是否在飛
82         bool isRunning;          // 是否在跑
83         bool isBig;              // 是否含滷蛋
84         bool isSwallow;          // 是否在吞怪
85         bool isLanding;          // 是否降落中
86         bool rightOrLeft;        // 判斷左右
87     private:
88         void ShowKirby(Map *m);   // 顯示普通卡比
89         void ShowSparkKirby(Map *m); // 顯示雷電卡比
90         void ShowFireKirby(Map *m); // 顯示火焰卡比
91         int exhaleDelay;          // 吐氣的時間
92         int jumpDistance;         // 跳躍的距離
93         int kickDistance;         // 踢擊的距離
94         int gasDistance;          // 氣體飛行距離
95         int starDistance;         // 星星飛行距離
96         bool bulletDirection;     // 吐出物體方向
97         bool isInvincible;        // 是否無敵
98         int InvincibleTime;       // 無敵時間
99         void Attack(Map *m);      // 普通卡比攻擊
100        void Spark_Attack(Map *m); // 雷電卡比攻擊
101        void Fire_Attack(Map *m);  // 火焰卡比攻擊
102        void Transform();          // 變身
103        int hp;                    // 血量
104        int type;                  // 型態
105        int eat;                   // 吃了什麼
106        int velocity;              // 重力加速度
107        int count;                 // 計數是否一秒
108        int height;                // 記錄當下的身高
109        int width;                 // 記錄當下的寬度
110    };
111 }
```

## Kirby.cpp

```
1  #include "stdafx.h"
2  #include "Resource.h"
3  #include <mmsystem.h>
4  #include <ddraw.h>
5  #include "audio.h"
6  #include "gamelib.h"
7  #include "Kirby.h"
8  namespace game_framework {
9      Kirby::Kirby()
10     {
11     }
12     int Kirby::GetX1()
13     {
14         return x;
15     }
16     int Kirby::GetY1()
17     {
18         return y;
19     }
20     int Kirby::GetX2()
21     {
22         return x + width;
23     }
24     int Kirby::GetY2()
25     {
26         return y + height;
27     }
28     int Kirby::GetType()
29     {
30         return type;
31     }
32     int Kirby::GetHeight()
33     {
34         return height;
35     }
```

```

36     int Kirby::GetWidth()
37     {
38         return width;
39     }
40     void Kirby::Initialize(int px, int py)
41     {
42         const int X_POS = px;
43         const int Y_POS = py;
44         x = X_POS;
45         y = Y_POS;
46         hp = 6;
47         jumpDistance = 120;
48         kickDistance = 100;
49         exhaleDelay = 10;
50         gasDistance = starDistance = 0;
51         InvincibleTime = 0;
52         type = 0;
53         eat = -1;
54         velocity = 2;
55         count = 0;
56         isMovingLeft = isMovingRight = isMovingUp = isMovingDown = isExhale = isJump = isAttack = false;
57         isKick = isFly = isHurted = isSuck = isBig = isSwallow = isRunning = isInvincible = isLanding = false;
58         isAlive = rightOrLeft = true;
59         blood0.SetTopLeft(SIZE_X / 2 - blood0.Width() / 2, SIZE_Y - blood0.Height());
60         blood1.SetTopLeft(SIZE_X / 2 - blood1.Width() / 2, SIZE_Y - blood1.Height());
61         blood2.SetTopLeft(SIZE_X / 2 - blood2.Width() / 2, SIZE_Y - blood2.Height());
62         blood3.SetTopLeft(SIZE_X / 2 - blood3.Width() / 2, SIZE_Y - blood3.Height());
63         blood4.SetTopLeft(SIZE_X / 2 - blood4.Width() / 2, SIZE_Y - blood4.Height());
64         blood5.SetTopLeft(SIZE_X / 2 - blood5.Width() / 2, SIZE_Y - blood5.Height());
65         blood6.SetTopLeft(SIZE_X / 2 - blood6.Width() / 2, SIZE_Y - blood6.Height());
66     }
67     bool Kirby::IsAlive()
68     {
69         return isAlive;
70     }
71     bool Kirby::IsFly()
72     {
73         return isFly;

```

```

74     }
75     bool Kirby::IsKick()
76     {
77         return isKick;
78     }
79     bool Kirby::IsSuck()
80     {
81         return isSuck;
82     }
83     bool Kirby::IsAttack()
84     {
85         return isAttack;
86     }
87     bool Kirby::IsRight()
88     {
89         return rightOrLeft;
90     }
91     bool Kirby::IsDown()
92     {
93         return isMovingDown;
94     }
95     bool Kirby::IsBig()
96     {
97         return isBig;
98     }
99     bool Kirby::IsMove()
100    {
101        return isMovingLeft || isMovingRight;
102    }
103    void Kirby::LoadBitmap() { . . . . . } //所有圖形的 addbitmap
104    void Kirby::OnMove(Map *m)
105    {
106        //被攻擊且不是在空中也沒超出邊界時
107        if (isHurted)
108        {
109            isFly = false;
110            if (m->isEmpty(GetX2() - width / 2, GetY2() + 1))
111                y++;

```

```

112         if (rightOrLeft && m->isEmpty(GetX1() - STEP_SIZE, GetY1() + height / 2)
113             && x - STEP_SIZE >= 0)
114             x -= STEP_SIZE;
115         else if (!rightOrLeft && m->isEmpty(GetX2() + STEP_SIZE, GetY2() - height / 2) && x +
116 STEP_SIZE <= m->GetWidth() - width)
117             x += STEP_SIZE;
118     }
119     else
120     {
121         if (isMovingLeft && !isSuck && !isSwallow && !isAttack)
122         {
123             rightOrLeft = false;          //設定面向左邊
124             // 上坡
125             if (m->isSlope(GetX1() - 1, GetY2()) && !isMovingDown && (isFly || !isMovingUp)) {
126                 if (isRunning && !isFly)
127                 {
128                     x -= STEP_SIZE * 2;
129                     y -= STEP_SIZE * 2;
130                 }
131                 else
132                 {
133                     x -= STEP_SIZE;
134                     y -= STEP_SIZE;
135                 }
136             }
137             // 下坡，前腳(左腳)離開斜坡(下方不是空的)就不要在下移了，不然會衝進土裡
138             else if (m->isSlope(GetX2(), GetY2() + 1) && m->isEmpty(GetX1(), GetY2() + 1)
139 && !isMovingDown && (isFly || !isMovingUp)) {
140                 if (isRunning && !isFly)
141                 {
142                     x -= STEP_SIZE * 2;
143                     y += STEP_SIZE * 2;
144                 }
145                 else
146                 {
147                     x -= STEP_SIZE;
148                     y += STEP_SIZE;
149                 }

```



```

150     }
151     //先判斷左邊是否可走且沒有按 Down，狀態要是向左飛行中或正常向左走
152     //為了防止變身後卡比變高導致牆壁失效，將判斷撞牆的點設在 Y2-10
153     else if (m->isEmpty(GetX1() - STEP_SIZE, GetY2() - 10) && !isMovingDown && (isFly
154 || !isMovingUp))
155     {
156         if (x <= 0) //邊界
157             x = 0;
158         else if (isRunning && !isFly && !m->isEmpty(GetX1() + width / 2, GetY2() + 1))
159             x -= STEP_SIZE * 2;
160         else
161             x -= STEP_SIZE;
162     }
163 }
164 else if (isMovingRight && !isSuck && !isSwallow && !isAttack)
165 {
166     rightOrLeft = true;          //設定面向右邊
167     // 上坡
168     if (m->isSlope(GetX2() + 1, GetY2()) && !isMovingDown && (isFly || !isMovingUp)) {
169         if (isRunning && !isFly)
170         {
171             x += STEP_SIZE * 2;
172             y -= STEP_SIZE * 2;
173         }
174         else
175         {
176             x += STEP_SIZE;
177             y -= STEP_SIZE;
178         }
179     }
180     // 下坡，前腳(右腳)離開斜坡(下方不是空的)就不要在下移了，不然會衝進土裡
181     else if (m->isSlope(GetX1(), GetY2() + 1) && m->isEmpty(GetX2(), GetY2() + 1)
182 && !isMovingDown && (isFly || !isMovingUp)) {
183         if (isRunning && !isFly)
184         {
185             x += STEP_SIZE * 2;
186             y += STEP_SIZE * 2;
187         }

```

```

188             else
189             {
190                 x += STEP_SIZE;
191                 y += STEP_SIZE;
192             }
193         }
194         //先判斷右邊是否可走且沒有按 Down，狀態要是向右飛行中或正常向右走
195         //為了防止變身後卡比變高導致牆壁失效，將判斷撞牆的點設在 Y2-10
196         else if (m->isEmpty(GetX2() + STEP_SIZE, GetY2() - 10) && !isMovingDown
197             && (isFly || !isMovingUp))
198         {
199             if (x >= m->GetWidth() - width) //邊界
200                 x = m->GetWidth() - width;
201             else if (isRunning && !isFly && !m->isEmpty(GetX2() - width / 2, GetY2() + 1))
202                 x += STEP_SIZE * 2;
203             else
204                 x += STEP_SIZE;
205         }
206     }
207     if (isMovingUp && !isBig)
208     {
209         if (isFly && m->isEmpty_2(GetX1() + flyR.Width() / 2, GetY1() - STEP_SIZE))
210         {
211             if (y <= 0) //邊界
212                 y = 0;
213             else
214                 y -= STEP_SIZE;
215         }
216     }
217     if (isJump && !isSuck)
218     {
219         if (jumpDistance == 120)
220             CAudio::Instance()->Play(jump);
221         jumpDistance -= 5;
222         if (m->isEmpty_2(GetX1() + width / 2, GetY1() - 5) && y - 5 > 0) //會不會撞到頭
223             y -= 5;
224         if (jumpDistance == 0 || y - 5 <= 0)
225         {

```

```

226             jumpDistance = 120;
227             isJump = false;
228         }
229     }
230 }
231 //地吸引力
232 if (!(isMovingUp || isJump) && m->isEmpty(GetX2(), GetY2() + 1) && m->isEmpty(GetX1(), GetY2() + 1))
233 {
234     count++;
235     isLanding = true;
236     if (isFly)
237         y += 1;
238     else
239     {
240         y += velocity; // y 軸下降(移動 velocity 個點，velocity 的單位為 點/次)
241         if (velocity < 5 && count == 30)
242         {
243             velocity++; // 受重力影響，下次的下降速度增加
244             count = 0;
245         }
246     }
247 }
248 else if (!isFly && m->isEmpty(x, y + height + 1) && m->isEmpty(x + width, y + height + 1))
249     y += 1;
250 else
251 {
252     if (isLanding == true && !isFly)
253     {
254         CAudio::Instance()->Play(landing);
255         isLanding = false;
256     }
257     velocity = 2; // 重設重力加速度
258     count = 0;
259 }
260 }

```

```
261 void Kirby::OnShow(Map *m)
262 {
263     if (type == 0)
264     {
265         if (isBig)
266         {
267             height = bigOriginR.Height();
268             width = bigOriginR.Width();
269         }
270         else
271         {
272             height = originR.Height();
273             width = originR.Width();
274         }
275     }
276     else if (type == 1)
277     {
278         height = Spark_originR.Height();
279         width = Spark_originR.Width();
280     }
281     else if (type == 2)
282     {
283         height = fire_originR.Height();
284         width = fire_originR.Width();
285     }
286     if (hp == 6) blood6.ShowBitmap();
287     else if (hp == 5) blood5.ShowBitmap();
288     else if (hp == 4) blood4.ShowBitmap();
289     else if (hp == 3) blood3.ShowBitmap();
290     else if (hp == 2) blood2.ShowBitmap();
291     else if (hp == 1) blood1.ShowBitmap();
292     else blood0.ShowBitmap();
293     if (InvincibleTime != 0) InvincibleTime--;
294     else isInvincible = false;
295     if (type == 0) ShowKirby(m);
296     else if (type == 1) ShowSparkKirby(m);
297     else if (type == 2) ShowFireKirby(m);
298
```

```

299         if (lost.IsAlive())
300         {
301             lost.OnMove(m, x, y, isSuck);
302             lost.OnShow(m);
303         }
304     }
305     void Kirby::Hurted(Map *m)
306     {
307         if (type != 0)
308         {
309             lost.Initialize(type, x, y);
310             type = 0;
311         }
312         if (!isInvincible)
313         {
314             hp--;
315             isInvincible = true;
316             InvincibleTime = 60;
317             if(hp>0)
318                 CAudio::Instance()->Play(hurted);
319         }
320         if (hp > 0)
321             isHurted = true;
322         else
323         {
324             CAudio::Instance()->Play(die);
325             CAudio::Instance()->Stop(AUDIO_BACKGROUND);
326             isAlive = false;
327         }
328     }
329     void Kirby::Die(Map *m)
330     {
331         GG.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
332         GG.ShowBitmap();
333         y -= 3;
334     }
335     void Kirby::SetMovingDown(bool flag)
336     {

```

```
337         isMovingDown = flag;
338     }
339     void Kirby::SetMovingLeft(bool flag)
340     {
341         isMovingLeft = flag;
342     }
343     void Kirby::SetMovingRight(bool flag)
344     {
345         isMovingRight = flag;
346     }
347     void Kirby::SetMovingUp(bool flag)
348     {
349         if (isBig)
350             isMovingUp = false;
351         else
352             isMovingUp = flag;
353     }
354     void Kirby::SetJump(bool flag)
355     {
356         isJump = flag;
357     }
358     void Kirby::SetAttack(bool flag)
359     {
360         isAttack = flag;
361     }
362     void Kirby::SetSuck(bool flag)
363     {
364         isSuck = flag;
365     }
366     void Kirby::SetBig(bool flag)
367     {
368         isBig = flag;
369     }
370     void Kirby::SetRun(bool flag) {
371         isRunning = flag;
372     }
373
374
```

```

375 void Kirby::SetXY(int nx, int ny)
376 {
377     x = nx; y = ny;
378 }
379 void Kirby::Restore()
380 {
381     hp = 6;
382 }
383 void Kirby::SetEat(int t)
384 {
385     eat = t;
386 }
387 Gas* Kirby::GetGas()
388 {
389     return &gas;
390 }
391 Star* Kirby::GetStar()
392 {
393     return &star;
394 }
395 void Kirby::Attack(Map *m)
396 {
397     //----- 空氣彈-----
398     if (isFly && isAttack)
399     {
400         gas.SetXY(x, y);
401         gas.SetAlive(true);
402         gasDistance = 176;
403         bulletDirection = rightOrLeft;
404     }
405     if (gasDistance != 0)
406     {
407         gasDistance -= STEP_SIZE * 2;
408         gas.OnMove(m, bulletDirection);
409         gas.OnShow(m);
410     }
411     else
412         gas.SetAlive(false);

```

```

413         //-----星星-----
414         if (isBig && isAttack && !isSuck && !star.IsAlive())
415         {
416             star.SetXY(x, y);
417             star.SetAlive(true);
418             starDistance = 376;
419             bulletDirection = rightOrLeft;
420         }
421         if (starDistance != 0)
422         {
423             starDistance -= STEP_SIZE * 2;
424             star.OnMove(m, bulletDirection);
425             star.OnShow(m);
426         }
427         else
428             star.SetAlive(false);
429         //-----踢擊-----
430         if (!isFly && !isBig && (isMovingDown || isKick) && (isAttack || isKick) && !m->isEmpty(GetX2() - width /
431 2, GetY2() + 1)) //在地面上蹲下按攻擊
432         {
433             isKick = true;
434             isAttack = false;
435             kickDistance -= 5;
436             if (rightOrLeft && m->isEmpty(GetX2() + 5, GetY2() - 5) && x + 5 <= m->GetWidth() -
437 downAttackR.Width()) //右邊會不會踢牆(y-5 是補償卡比大小)
438                 x += 5;
439             else if (!rightOrLeft && m->isEmpty(GetX1() - 5, GetY2() - 5) && x - 5 >= 0) //左邊會不會踢牆
440 (y-5 是補償卡比大小)
441                 x -= 5;
442             //踢完.踢到邊界.腳沒踏到地都不能使用踢擊
443             if (kickDistance == 0 || x <= 0 || x >= m->GetWidth() || m->isEmpty(GetX2() - width / 2, GetY2() + 1))
444             {
445                 kickDistance = 100;
446                 isKick = false;
447             }
448         }
449     }
450

```



```

451 void Kirby::Spark_Attack(Map *m)
452 {
453     //放電在怪物那邊偵測
454     //-----空氣彈-----
455     if (isFly && isAttack)
456     {
457         gas.SetXY(x, y + 40);
458         gas.SetAlive(true);
459         gasDistance = 176;
460         bulletDirection = rightOrLeft;
461     }
462     if (gasDistance != 0)
463     {
464         gasDistance -= STEP_SIZE * 2;
465         gas.OnMove(m, bulletDirection);
466         gas.OnShow(m);
467     }
468     else
469         gas.SetAlive(false);
470     //-----踢擊-----
471     if (!isFly && !isBig && (isMovingDown || isKick) && (isAttack || isKick) && !m->isEmpty(GetX2() - width /
472 2, GetY2() + 1)) //在地面上蹲下按攻擊
473     {
474         isKick = true;
475         isAttack = false;
476         kickDistance -= 5;
477         if (rightOrLeft && m->isEmpty(GetX2() + 5, GetY2() - 5) && x + 5 <= m->GetWidth() -
478 Spark_downAttackR.Width() + 40) //右邊會不會踢牆(y-5是補償卡比大小)(+40是消除動畫補償)
479             x += 5;
480         else if (!rightOrLeft && m->isEmpty(GetX1() - 5, GetY2() - 5) && x - 5 >= 0) //左邊會不會踢牆
481 (y-5是補償卡比大小)
482             x -= 5;
483         //踢完.踢到邊界.腳沒踏到地都不能使用踢擊
484         if (kickDistance == 0 || x <= 0 || x >= m->GetWidth() || m->isEmpty(GetX2() - width / 2, GetY2() + 1))
485         {
486             kickDistance = 100;
487             isKick = false;
488         }

```

```

489         }
490     }
491     void Kirby::Fire_Attack(Map *m)
492     {
493         //放火在怪物那邊偵測
494         //-----空氣彈-----
495         if (isFly && isAttack)
496         {
497             gas.SetXY(x, y + 40);
498             gas.SetAlive(true);
499             gasDistance = 176;
500             bulletDirection = rightOrLeft;
501         }
502         if (gasDistance != 0)
503         {
504             gasDistance -= STEP_SIZE * 2;
505             gas.OnMove(m, bulletDirection);
506             gas.OnShow(m);
507         }
508         else
509             gas.SetAlive(false);
510         //-----踢擊-----
511         if (!isFly && !isBig && (isMovingDown || isKick) && (isAttack || isKick) && !m->isEmpty(GetX2() - width /
512 2, GetY2() + 1)) //在地面上蹲下按攻擊
513         {
514             isKick = true;
515             isAttack = false;
516             kickDistance -= 5;
517             if (rightOrLeft && m->isEmpty(GetX2() + 5, GetY2() - 5) && x + 5 <= m->GetWidth() -
518 fire_downAttackR.Width() + 30) //右邊會不會踢牆(y-5是補償卡比大小)(+40是消除動畫補償)
519                 x += 5;
520             else if (!rightOrLeft && m->isEmpty(GetX1() - 5, GetY2() - 5) && x - 5 >= 0) //左邊會不會踢牆
521 (y-5是補償卡比大小)
522                 x -= 5;
523             //踢完.踢到邊界.腳沒踏到地都不能使用踢擊
524             if (kickDistance == 0 || x <= 0 || x >= m->GetWidth() || m->isEmpty(GetX2() - width / 2, GetY2() + 1))
525                 {
526                     kickDistance = 100;

```

```

527         isKick = false;
528     }
529 }
530 }
531 void Kirby::Transform()
532 {
533     if (eat == 1)
534     {
535         type = 1;
536         y -= 40;
537     }
538     else if (eat == 2)
539     {
540         type = 2;
541         y -= 40;
542     }
543 }
544 void Kirby::ShowKirby(Map *m)
545 {
546     originR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
547     originL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
548     goL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
549     goR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
550     prepareFlyR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
551     prepareFlyL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
552     flyR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
553     flyL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
554     exhaleR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
555     exhaleL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
556     jumpR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
557     jumpL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
558     downR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
559     downL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
560     landingL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
561     landingR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
562     downAttackL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
563     downAttackR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
564     hurtedL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));

```

```

565         hurtedR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
566         runR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
567         runL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
568         suckL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
569         suckR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
570         bigOriginR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
571         bigOriginL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
572         bigJumpR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
573         bigJumpL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
574         bigLandingR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
575         bigLandingL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
576         threwR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
577         threwL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
578         bigGoL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
579         bigGoR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
580         swallowR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
581         swallowL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
582         Attack(m);
583         if (!isMovingUp && !isFly)    //沒吸氣也沒飛行就要 reset 吸氣動畫
584         {
585             prepareFlyR.Reset();
586             prepareFlyL.Reset();
587         }
588         if (isHurted)    // 被攻擊
589         {
590             isBig = false;
591             if (rightOrLeft)
592             {
593                 hurtedR.OnMove();
594                 hurtedR.OnShow();
595             }
596             else
597             {
598                 hurtedL.OnMove();
599                 hurtedL.OnShow();
600             }
601             if (hurtedL.IsFinalBitmap() || hurtedR.IsFinalBitmap())
602             {

```

```

603             isHurted = false;
604             hurtedL.Reset();
605             hurtedR.Reset();
606         }
607     }
608     else if (isBig)
609     {
610         if(isSuck)
611             isAttack = false;
612         if (isJump)    //跳躍
613         {
614             if (rightOrLeft)
615                 bigJumpR.ShowBitmap();
616             else
617                 bigJumpL.ShowBitmap();
618         }
619         // || (starDistance < 376 && starDistance > 346)是為了讓吐怪的圖能顯示久一點到大概 1 秒(每秒跑 30 次 OnShow 迴圈)
620         else if (isAttack || (starDistance < 376 && starDistance > 346))    //吐怪
621         {
622             // isAttack 要設為 false，避免吐怪完馬上吸怪的 bug
623             isAttack = false;
624             if (rightOrLeft)
625                 threwR.ShowBitmap();
626             else
627                 threwL.ShowBitmap();
628             // 在吐氣的圖顯示結束前將卡比變回原形
629             if (starDistance <= 350)
630                 isBig = false;
631         }
632         else if (isSwallow || (isMovingDown && (!m->isEmpty(GetX1(), GetY2() + 1) || !m->isEmpty(GetX2(),
633             GetY2() + 1))))    //吞食
634         {
635             isSwallow = true;
636             if (rightOrLeft)
637             {
638                 swallowR.OnMove();
639                 swallowR.OnShow();
640             }

```

```

641         else
642         {
643             swallowL.OnMove();
644             swallowL.OnShow();
645         }
646         if (swallowR.IsFinalBitmap() || swallowL.IsFinalBitmap())
647         {
648             isSwallow = false;
649             isBig = false;
650             Transform();
651             swallowR.Reset();
652             swallowL.Reset();
653         }
654     }
655     //自由落體
656     //兩腳地都必須空，且不是走在斜坡，才會顯示降落的圖
657     else if (!isFly && m->isEmpty(GetX1(), GetY2() + 1) && m->isEmpty(GetX2(), GetY2() + 1)
658 && !m->isSlope(GetX1(), GetY2() + 1) && !m->isSlope(GetX2(), GetY2() + 1))
659     {
660         if (rightOrLeft)
661             bigLandingR.ShowBitmap();
662         else
663             bigLandingL.ShowBitmap();
664     }
665     else if (isMovingLeft)    //向左走
666     {
667         if (isRunning)
668             bigGoL.SetDelayCount(2);
669         else
670             bigGoL.SetDelayCount(10);
671         bigGoL.OnShow();
672         bigGoL.OnMove();
673     }
674     else if (isMovingRight)    //向右走
675     {
676         if (isRunning)
677             bigGoR.SetDelayCount(2);
678         else

```

```

679             bigGoR.SetDelayCount(10);
680             bigGoR.OnShow();
681             bigGoR.OnMove();
682         }
683         else if (rightOrLeft)    //面相右
684             bigOriginR.ShowBitmap();
685         else if (!rightOrLeft)  //面相左
686             bigOriginL.ShowBitmap();
687     }
688     else
689     {
690         if (isAttack && !isFly && !isMovingDown) //吸怪
691         {
692             isSuck = true;
693             if (rightOrLeft)
694             {
695                 suckR.OnMove();
696                 suckR.OnShow();
697             }
698             else
699             {
700                 suckL.OnMove();
701                 suckL.OnShow();
702             }
703             if (lost.IsAlive() && (lost.GetX() - x >= -2) && (lost.GetX() - x <= 2) && (lost.GetY() - y >= -2)
704                 && (lost.GetY() - y <= 2))
705             {
706                 lost.SetAlive(false);
707                 SetBig(true);
708                 SetEat(lost.type);
709             }
710         }
711         else if (isJump)    //跳躍
712         {
713             if (rightOrLeft)
714                 jumpR.ShowBitmap();
715             else
716                 jumpL.ShowBitmap();

```

```

717         }
718         else if ((isAttack || isExhale) && isFly)    //吐氣
719         {
720             // isAttack 要設為 false，避免吐氣完馬上吸怪的 bug
721             isAttack = false;
722             isExhale = true;
723             exhaleDelay--;
724             if (rightOrLeft)
725                 exhaleR.ShowBitmap();
726             else
727                 exhaleL.ShowBitmap();
728             if (exhaleDelay == 0)
729             {
730                 exhaleDelay = 10;
731                 prepareFlyR.Reset();
732                 prepareFlyL.Reset();
733                 isFly = false;
734                 isExhale = false;
735             }
736             SetEat(-1);
737         }
738         else if (isMovingUp)
739         {
740             if (prepareFlyR.IsFinalBitmap() || prepareFlyL.IsFinalBitmap())    //飛行前吸氣
741             {
742                 isFly = true;
743                 if (rightOrLeft)
744                 {
745                     if (flyR.GetCurrentBitmapNumber() == 1)
746                         CAudio::Instance()->Play(fly);
747                     flyR.OnShow();
748                     flyR.OnMove();
749                 }
750                 else
751                 {
752                     if (flyL.GetCurrentBitmapNumber() == 1)
753                         CAudio::Instance()->Play(fly);
754                     flyL.OnShow();

```



```

755             flyL.OnMove();
756         }
757     }
758     else
759     {
760         if (rightOrLeft)
761         {
762             prepareFlyR.OnShow();
763             prepareFlyR.OnMove();
764         }
765         else
766         {
767             prepareFlyL.OnShow();
768             prepareFlyL.OnMove();
769         }
770     }
771 }
772 else if (isFly) //飛行
773 {
774     if (rightOrLeft)
775     {
776         if (flyR.GetCurrentBitmapNumber() == 1)
777             CAudio::Instance()->Play(fly);
778         flyR.OnShow();
779         flyR.OnMove();
780     }
781     else
782     {
783         if (flyL.GetCurrentBitmapNumber() == 1)
784             CAudio::Instance()->Play(fly);
785         flyL.OnShow();
786         flyL.OnMove();
787     }
788 }
789 //自由落體
790 //兩腳地都必須空，且不是走在斜坡，才會顯示降落的圖
791 else if (!isFly && m->isEmpty(GetX1(), GetY2() + 1) && m->isEmpty(GetX2(), GetY2() + 1)
792 && !m->isSlope(GetX1(), GetY2() + 1) && !m->isSlope(GetX2(), GetY2() + 1))

```

```

793         {
794             if (rightOrLeft)
795                 landingR.ShowBitmap();
796             else
797                 landingL.ShowBitmap();
798         }
799     else if (isKick && !isFly)    //踢擊
800     {
801         if (rightOrLeft)
802             downAttackR.ShowBitmap();
803         else
804             downAttackL.ShowBitmap();
805     }
806     else if (isMovingDown && !m->isEmpty(GetX1() + width / 2, GetY2() + 1))    //縮小
807     {
808         if (rightOrLeft)
809             downR.ShowBitmap();
810         else
811             downL.ShowBitmap();
812     }
813     else if (!isFly && isRunning && (isMovingLeft || isMovingRight))    //跑
814     {
815         if (!rightOrLeft)
816         {
817             runL.OnShow();
818             runL.OnMove();
819         }
820         else
821         {
822             runR.OnShow();
823             runR.OnMove();
824         }
825     }
826     else if (!isFly && isMovingLeft)    //一般向左走
827     {
828         goL.OnShow();
829         goL.OnMove();
830     }

```

```

831         else if (!isFly && isMovingRight)    //一般向右走
832         {
833             goR.OnShow();
834             goR.OnMove();
835         }
836         else if (rightOrLeft)    //面相右
837             originR.ShowBitmap();
838         else if (!rightOrLeft)    //面相左
839             originL.ShowBitmap();
840     }
841 }
842 void Kirby::ShowSparkKirby(Map *m)
843 {
844     // 雷電卡比圖片.動畫定位及大小補償
845     Spark_exhaleR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
846     Spark_exhaleL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
847     Spark_downR.SetTopLeft(m->ScreenX(x), m->ScreenY(y+20));
848     Spark_downL.SetTopLeft(m->ScreenX(x), m->ScreenY(y+20));
849     Spark_landingR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
850     Spark_landingL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
851     Spark_originR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
852     Spark_originL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
853     Spark_goR.SetTopLeft(m->ScreenX(x), m->ScreenY(y-5));
854     Spark_goL.SetTopLeft(m->ScreenX(x), m->ScreenY(y-5));
855     Spark_jumpR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
856     Spark_jumpL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
857     Spark_downAttackR.SetTopLeft(m->ScreenX(x-40), m->ScreenY(y+30));
858     Spark_downAttackL.SetTopLeft(m->ScreenX(x), m->ScreenY(y+30));
859     Spark_flyR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
860     Spark_prepareFlyR.SetTopLeft(m->ScreenX(x), m->ScreenY(y+5));
861     Spark_flyL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
862     Spark_prepareFlyL.SetTopLeft(m->ScreenX(x), m->ScreenY(y+5));
863     Spark_runR.SetTopLeft(m->ScreenX(x-20), m->ScreenY(y+10));
864     Spark_runL.SetTopLeft(m->ScreenX(x+10), m->ScreenY(y+10));
865     Spark_attackR.SetTopLeft(m->ScreenX(x-40), m->ScreenY(y-20));
866     Spark_attackL.SetTopLeft(m->ScreenX(x-40), m->ScreenY(y-20));
867     hurtedL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
868     hurtedR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));

```

```

869         Spark_Attack(m);
870         if (!isMovingUp && !isFly)    //沒吸氣也沒飛行就要 reset 吸氣動畫
871         {
872             Spark_prepareFlyR.Reset();
873             Spark_prepareFlyL.Reset();
874         }
875         if (isHurted)    // 被攻擊
876         {
877             y += 40;    //還原補償的高度
878             if (rightOrLeft)
879             {
880                 hurtedR.OnMove();
881                 hurtedR.OnShow();
882             }
883             else
884             {
885                 hurtedL.OnMove();
886                 hurtedL.OnShow();
887             }
888             if (hurtedL.IsFinalBitmap() || hurtedR.IsFinalBitmap())
889             {
890                 isHurted = false;
891                 hurtedL.Reset();
892                 hurtedR.Reset();
893             }
894         }
895         else
896         {
897             if (isAttack && !isFly && !isMovingDown)    //放電
898             {
899                 if (rightOrLeft)
900                 {
901                     Spark_attackR.SetDelayCount(5);
902                     Spark_attackR.OnMove();
903                     Spark_attackR.OnShow();
904                 }
905                 else
906                 {

```

```

907             Spark_attackL.SetDelayCount(5);
908             Spark_attackL.OnMove();
909             Spark_attackL.OnShow();
910         }
911     }
912     else if (isJump)    //跳躍
913     {
914         if (rightOrLeft)
915         {
916             Spark_jumpR.OnMove();
917             Spark_jumpR.OnShow();
918         }
919         else
920         {
921             Spark_jumpL.OnMove();
922             Spark_jumpL.OnShow();
923         }
924     }
925     else if ((isAttack || isExhale) && isFly)    //吐氣
926     {
927         // isAttack 要設為 false，避免吐氣完馬上吸怪的 bug
928         isAttack = false;
929         isExhale = true;
930         exhaleDelay--;
931         if (rightOrLeft)
932             Spark_exhaleR.ShowBitmap();
933         else
934             Spark_exhaleL.ShowBitmap();
935         if (exhaleDelay == 0)
936         {
937             exhaleDelay = 10;
938             Spark_prepareFlyR.Reset();
939             Spark_prepareFlyL.Reset();
940             isFly = false;
941             isExhale = false;
942         }
943     }
944     else if (isMovingUp)

```

```
945         {
946             if (Spark_prepareFlyR.IsFinalBitmap() || Spark_prepareFlyL.IsFinalBitmap())    //飛行前吸氣
947             {
948                 isFly = true;
949                 if (rightOrLeft)
950                 {
951                     if (Spark_flyR.GetCurrentBitmapNumber() == 3)
952                         CAudio::Instance()->Play(fly);
953                     Spark_flyR.OnShow();
954                     Spark_flyR.OnMove();
955                 }
956                 else
957                 {
958                     if (Spark_flyL.GetCurrentBitmapNumber() == 3)
959                         CAudio::Instance()->Play(fly);
960                     Spark_flyL.OnShow();
961                     Spark_flyL.OnMove();
962                 }
963             }
964             else
965             {
966                 if (rightOrLeft)
967                 {
968                     Spark_prepareFlyR.OnShow();
969                     Spark_prepareFlyR.OnMove();
970                 }
971                 else
972                 {
973                     Spark_prepareFlyL.OnShow();
974                     Spark_prepareFlyL.OnMove();
975                 }
976             }
977         }
978     else if (isFly)    //飛行
979     {
980         if (rightOrLeft)
981         {
```

```

982         if (Spark_flyR.GetCurrentBitmapNumber() == 3)
983             CAudio::Instance()->Play(fly);
984         Spark_flyR.OnShow();
985         Spark_flyR.OnMove();
986     }
987     else
988     {
989         if (Spark_flyL.GetCurrentBitmapNumber() == 3)
990             CAudio::Instance()->Play(fly);
991         Spark_flyL.OnShow();
992         Spark_flyL.OnMove();
993     }
994 }
995 //自由落體
996 //兩腳地都必須空，且不是走在斜坡，才會顯示降落的圖
997 else if (!isFly && m->isEmpty(GetX1(), GetY2() + 1) && m->isEmpty(GetX2(), GetY2() + 1)
998 && !m->isSlope(GetX1(), GetY2() + 1) && !m->isSlope(GetX2(), GetY2() + 1))
999     {
1000         if (rightOrLeft)
1001         {
1002             Spark_landingR.OnShow();
1003             Spark_landingR.OnMove();
1004         }
1005         else
1006         {
1007             Spark_landingL.OnShow();
1008             Spark_landingL.OnMove();
1009         }
1010     }
1011 else if (isKick && !isFly) //踢擊
1012 {
1013     if (rightOrLeft)
1014     {
1015         Spark_downAttackR.OnMove();
1016         Spark_downAttackR.OnShow();
1017     }

```

```

1018             else
1019             {
1020                 Spark_downAttackL.OnMove();
1021                 Spark_downAttackL.OnShow();
1022             }
1023         }
1024         else if (isMovingDown && !m->isEmpty(GetX1() + width / 2, GetY1() + height + 1)) //縮小
1025         {
1026             if (rightOrLeft)
1027             {
1028                 Spark_downR.OnShow();
1029                 Spark_downR.OnMove();
1030             }
1031             else
1032             {
1033                 Spark_downL.OnShow();
1034                 Spark_downL.OnMove();
1035             }
1036         }
1037         else if (!isFly && isRunning && (isMovingLeft || isMovingRight)) //跑
1038         {
1039             if (!rightOrLeft)
1040             {
1041                 Spark_runL.OnShow();
1042                 Spark_runL.OnMove();
1043             }
1044             else
1045             {
1046                 Spark_runR.OnShow();
1047                 Spark_runR.OnMove();
1048             }
1049         }
1050         else if (!isFly && isMovingLeft) //一般向左走
1051         {
1052             Spark_goL.OnShow();
1053             Spark_goL.OnMove();
1054         }

```



```

1055         else if (!isFly && isMovingRight)    //一般向右走
1056         {
1057             Spark_goR.OnShow();
1058             Spark_goR.OnMove();
1059         }
1060         else if (rightOrLeft)    //面相右
1061         {
1062             Spark_originR.OnMove();
1063             Spark_originR.OnShow();
1064         }
1065         else if (!rightOrLeft)    //面相左
1066         {
1067             Spark_originL.OnMove();
1068             Spark_originL.OnShow();
1069         }
1070     }
1071 }
1072 void Kirby::ShowFireKirby(Map *m)
1073 {
1074     fire_exhaleR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1075     fire_exhaleL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1076     fire_originR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1077     fire_originL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1078     fire_downR.SetTopLeft(m->ScreenX(x), m->ScreenY(y + 20));
1079     fire_downL.SetTopLeft(m->ScreenX(x), m->ScreenY(y + 20));
1080     fire_goR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1081     fire_goL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1082     fire_jumpR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1083     fire_jumpL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1084     fire_landingR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1085     fire_landingL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1086     fire_downAttackR.SetTopLeft(m->ScreenX(x - 30), m->ScreenY(y + 28));
1087     fire_downAttackL.SetTopLeft(m->ScreenX(x), m->ScreenY(y + 28));
1088     fire_flyR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1089     fire_prepareFlyR.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1090     fire_flyL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1091     fire_prepareFlyL.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
1092     fire_runR.SetTopLeft(m->ScreenX(x - 35), m->ScreenY(y + 5));

```

```

1093         fire_runL.SetTopLeft(m->ScreenX(x), m->ScreenY(y + 5));
1094         fire_attackR.SetTopLeft(m->ScreenX(x - 15), m->ScreenY(y + 10));
1095         fire_attackL.SetTopLeft(m->ScreenX(x + 10), m->ScreenY(y + 10));
1096         Fire_Attack(m);
1097         attack1.SetDelayCount(5);
1098         attack2.SetDelayCount(5);
1099         attack3.SetDelayCount(5);
1100         if (!isMovingUp && !isFly) //沒吸氣也沒飛行就要 reset 吸氣動畫
1101         {
1102             fire_prepareFlyR.Reset();
1103             fire_prepareFlyL.Reset();
1104         }
1105         if (isHurted) // 被攻擊
1106         {
1107             y += 40; //還原補償的高度
1108             if (rightOrLeft)
1109             {
1110                 hurtedR.OnMove();
1111                 hurtedR.OnShow();
1112             }
1113             else
1114             {
1115                 hurtedL.OnMove();
1116                 hurtedL.OnShow();
1117             }
1118             if (hurtedL.IsFinalBitmap() || hurtedR.IsFinalBitmap())
1119             {
1120                 isHurted = false;
1121                 hurtedL.Reset();
1122                 hurtedR.Reset();
1123             }
1124         }
1125         else
1126         {
1127             if (isAttack && !isFly && !isMovingDown) //放火
1128             {
1129                 if (rightOrLeft)
1130                 {

```

```

1131         fire_attackR.OnShow();
1132         fire_attackR.OnMove();
1133         attack1.SetTopLeft(m->ScreenX(x + 63), m->ScreenY(y + 30));
1134         attack2.SetTopLeft(m->ScreenX(x + 83), m->ScreenY(y + 10));
1135         attack3.SetTopLeft(m->ScreenX(x + 133), m->ScreenY(y + 10));
1136     }
1137     else
1138     {
1139         fire_attackL.OnShow();
1140         fire_attackL.OnMove();
1141         attack1.SetTopLeft(m->ScreenX(x - 52), m->ScreenY(y + 30));
1142         attack2.SetTopLeft(m->ScreenX(x - 72), m->ScreenY(y + 10));
1143         attack3.SetTopLeft(m->ScreenX(x - 122), m->ScreenY(y + 10));
1144     }
1145     attack1.OnMove();
1146     attack1.OnShow();
1147     attack2.OnMove();
1148     attack2.OnShow();
1149     attack3.OnMove();
1150     attack3.OnShow();
1151 }
1152 else if (isJump)    //跳躍
1153 {
1154     if (rightOrLeft)
1155     {
1156         fire_jumpR.OnMove();
1157         fire_jumpR.OnShow();
1158     }
1159     else
1160     {
1161         fire_jumpL.OnMove();
1162         fire_jumpL.OnShow();
1163     }
1164 }
1165 else if ((isAttack || isExhale) && isFly)    //吐氣
1166 {
1167     // isAttack 要設為 false，避免吐氣完馬上吸怪的 bug
1168     isAttack = false;

```

```

1169         isExhale = true;
1170         exhaleDelay--;
1171         if (rightOrLeft)
1172             fire_exhaleR.ShowBitmap();
1173         else
1174             fire_exhaleL.ShowBitmap();
1175         if (exhaleDelay == 0)
1176         {
1177             exhaleDelay = 10;
1178             fire_prepareFlyR.Reset();
1179             fire_prepareFlyL.Reset();
1180             isFly = false;
1181             isExhale = false;
1182         }
1183     }
1184     else if (isMovingUp)
1185     {
1186         if (fire_prepareFlyR.IsFinalBitmap() || fire_prepareFlyL.IsFinalBitmap()) //飛行前吸氣
1187         {
1188             isFly = true;
1189             if (rightOrLeft)
1190             {
1191                 if (fire_flyR.GetCurrentBitmapNumber() == 3)
1192                     CAudio::Instance()->Play(fly);
1193                 fire_flyR.OnShow();
1194                 fire_flyR.OnMove();
1195             }
1196             else
1197             {
1198                 if (fire_flyL.GetCurrentBitmapNumber() == 3)
1199                     CAudio::Instance()->Play(fly);
1200                 fire_flyL.OnShow();
1201                 fire_flyL.OnMove();
1202             }
1203         }
1204         else
1205         {

```

```

1206         if (rightOrLeft)
1207         {
1208             fire_prepareFlyR.OnShow();
1209             fire_prepareFlyR.OnMove();
1210         }
1211         else
1212         {
1213             fire_prepareFlyL.OnShow();
1214             fire_prepareFlyL.OnMove();
1215         }
1216     }
1217 }
1218 else if (isFly) //飛行
1219 {
1220     if (rightOrLeft)
1221     {
1222         if (fire_flyR.GetCurrentBitmapNumber() == 3)
1223             CAudio::Instance()->Play(fly);
1224         fire_flyR.OnShow();
1225         fire_flyR.OnMove();
1226     }
1227     else
1228     {
1229         if (fire_flyL.GetCurrentBitmapNumber() == 3)
1230             CAudio::Instance()->Play(fly);
1231         fire_flyL.OnShow();
1232         fire_flyL.OnMove();
1233     }
1234 }
1235 //自由落體
1236 //兩腳地都必須空，且不是走在斜坡，才會顯示降落的圖
1237 else if (!isFly && m->isEmpty(GetX1(), GetY2() + 1) && m->isEmpty(GetX2(), GetY2() + 1)
1238 && !m->isSlope(GetX1(), GetY2() + 1) && !m->isSlope(GetX2(), GetY2() + 1))
1239 {
1240     if (rightOrLeft)
1241     {
1242         fire_landingR.OnShow();
1243         fire_landingR.OnMove();

```

```

1244         }
1245         else
1246         {
1247             fire_landingL.OnShow();
1248             fire_landingL.OnMove();
1249         }
1250     }
1251     else if (isKick && !isFly)    //踢擊
1252     {
1253         if (rightOrLeft)
1254         {
1255             fire_downAttackR.OnMove();
1256             fire_downAttackR.OnShow();
1257         }
1258         else
1259         {
1260             fire_downAttackL.OnMove();
1261             fire_downAttackL.OnShow();
1262         }
1263     }
1264     else if (isMovingDown && !m->isEmpty(GetX1() + width / 2, GetY1() + height + 1))    //縮小
1265     {
1266         if (rightOrLeft)
1267         {
1268             fire_downR.OnShow();
1269             fire_downR.OnMove();
1270         }
1271         else
1272         {
1273             fire_downL.OnShow();
1274             fire_downL.OnMove();
1275         }
1276     }
1277     else if (!isFly && isRunning && (isMovingLeft || isMovingRight))    //跑
1278     {
1279         if (!rightOrLeft)
1280         {
1281             fire_runL.OnShow();

```

```

1282             fire_runL.OnMove();
1283         }
1284         else
1285         {
1286             fire_runR.OnShow();
1287             fire_runR.OnMove();
1288         }
1289     }
1290     else if (!isFly && isMovingLeft)    //一般向左走
1291     {
1292         fire_goL.OnShow();
1293         fire_goL.OnMove();
1294     }
1295     else if (!isFly && isMovingRight)    //一般向右走
1296     {
1297         fire_goR.OnShow();
1298         fire_goR.OnMove();
1299     }
1300     else if (rightOrLeft)    //面相右
1301     {
1302         fire_originR.OnMove();
1303         fire_originR.OnShow();
1304     }
1305     else if (!rightOrLeft)    //面相左
1306     {
1307         fire_originL.OnMove();
1308         fire_originL.OnShow();
1309     }
1310     }
1311 }
1312 }
```

## Enemy.h

```

1  #pragma once
2  #include "Map.h"
3  #include "Kirby.h"
4  namespace game_framework {
5      class Enemy
6      {
7      public:
8          Enemy();
9          int GetX1();                // 左上角 x 座標
10         int GetY1();                // 左上角 y 座標
11         void Hurted(Kirby* kirby);  // 被攻擊
12         void Sucked(Kirby* kirby);  // 被吸
13         void SetAlive(bool flag);    // 設定死活
14         bool IsAlive();              // 是否活著
15         virtual int GetX2();          // 右下角 x 座標
16         virtual int GetY2();          // 右下角 y 座標
17         virtual void OnShow(Map *m, Kirby *kirby); // 顯示
18         virtual void OnMove(Map *m, Kirby* kirby); // 移動
19         virtual void LoadBitmap();   // 讀圖
20         virtual void Attack(Kirby* kirby); // 攻擊
21     protected:
22         int x, y, hp;                // 左上角座標.血量
23         bool is_alive;               // 是否活著
24         bool is_sucked;              // 是否被吸
25         bool RightOrLeft;            // 判斷左右
26         bool Ishurted;               // 受傷與否
27         bool IsKing;                 // 是不是王
28         bool HitRectangle(int tx1, int ty1, int tx2, int ty2); // 是否碰到參數範圍的矩形
29         double ComputeDistance(int tx1, int ty1); // 計算距離
30     };
31 }

```



## Enemy.cpp

```

1  #include "stdafx.h"
2  #include "Resource.h"
3  #include <mmsystem.h>
4  #include <draw.h>
5  #include "audio.h"
6  #include "gamelib.h"
7  #include "Enemy.h"
8  namespace game_framework {
9      Enemy::Enemy()
10     {
11     }
12     int Enemy::GetX1()
13     {
14         return x;
15     }
16     int Enemy::GetY1()
17     {
18         return y;
19     }
20     int Enemy::GetX2()
21     {
22         return 0;
23     }
24     int Enemy::GetY2()
25     {
26         return 0;
27     }
28     void Enemy::Hurted(Kirby* kirby)
29     {
30         // 被踢
31         if ((kirby->GetType() == 0 && HitRectangle(kirby->GetX1(), kirby->GetY1(), kirby->GetX2(),
32 kirby->GetY2()) && kirby->IsKick()) || ((kirby->GetType() == 1 || kirby->GetType() == 2) && HitRectangle(kirby->GetX1(),
33 kirby->GetY1() + 40, kirby->GetX2(), kirby->GetY1() + 60) && kirby->IsKick())) {
34             hp -= 10;
35             Ishurted = true;

```

```

36         }
37         // 被空氣砲擊中
38         else if (kirby->GetGas()->IsAlive())&&HitRectangle(kirby->GetGas()->GetX1()+10, kirby->GetGas()->GetY1()
39 + 10, kirby->GetGas()->GetX2() - 10, kirby->GetGas()->GetY2() - 10)) {
40             hp -= 10;
41             Ishurted = true;
42             kirby->GetGas()->SetAlive(false);
43         }
44         // 被星星擊中
45         else if (kirby->GetStar()->IsAlive() && HitRectangle(kirby->GetStar()->GetX1() + 10,
46 kirby->GetStar()->GetY1() + 10, kirby->GetStar()->GetX2() - 10, kirby->GetStar()->GetY2() - 10)) {
47             hp -= 20;
48             Ishurted = true;
49             kirby->GetStar()->SetAlive(false);
50         }
51         // 被電
52         else if (kirby->GetType() == 1 && kirby->IsAttack() && !kirby->IsDown() && HitRectangle(kirby->GetX1()
53 - 40, kirby->GetY1() - 10, kirby->GetX1() + 90, kirby->GetY1() + 110)){ // -10 補償圖片 +145.+158 是雷電圖檔的大小
54             hp -= 1;
55             Ishurted = true;
56         }
57         // 被燒
58         else if (kirby->GetType() == 2 && kirby->IsAttack() && !kirby->IsDown() && ((kirby->IsRight() &&
59 HitRectangle(kirby->GetX1() + 63, kirby->GetY1() + 12, kirby->GetX1() + 203, kirby->GetY1() + 90)) || (!kirby->IsRight()
60 && HitRectangle(kirby->GetX1() - 112, kirby->GetY1() + 12, kirby->GetX1() - 12, kirby->GetY1() + 90)))) {
61             hp -= 1;
62             Ishurted = true;
63         }
64         else {
65             Ishurted = false;
66         }
67         if (kirby->IsSuck() && (x - kirby->GetX1() > 2 || x - kirby->GetX1() < -2 || y - kirby->GetY1() > 2 || y -
68 kirby->GetY1() < -2))
69             is_alive = true;
70         else if (hp <= 0)
71         {
72             CAudio::Instance()->Play(enemyDie);
73             is_alive = false;

```

```

74         }
75     }
76     void Enemy::Sucked(Kirby* kirby)
77     {
78         if (IsKing)
79             return;
80         if (ComputeDistance(kirby->GetX1(), kirby->GetY1()) < 150.0 && kirby->IsSuck() && ((kirby->IsRight()
81 && x - kirby->GetX1() >= 0) || (!kirby->IsRight() && x - kirby->GetX2() <= 0)))
82         {
83             hp -= 10;
84             if (hp <= 0)
85                 is_sucked = true;
86         }
87         else
88             is_sucked = false;
89     }
90     void Enemy::SetAlive(bool flag)
91     {
92         is_alive = flag;
93     }
94     bool Enemy::IsAlive() {
95         return is_alive;
96     }
97     void Enemy::OnShow(Map *m, Kirby *kirby)
98     {
99     }
100    void Enemy::OnMove(Map *m, Kirby* kirby)
101    {
102    }
103    void Enemy::LoadBitmap()
104    {
105    }
106    void Enemy::Attack(Kirby* kirby)
107    {
108    }
109    bool Enemy::HitRectangle(int tx1, int ty1, int tx2, int ty2)
110    {
111        int x1 = GetX1();    // 左上角 x 座標

```

```

112         int y1 = GetY1();    // 左上角 y 座標
113         int x2 = GetX2();    // 右下角 x 座標
114         int y2 = GetY2();    // 右下角 y 座標
115         // 檢測矩形與參數矩形是否有交集
116         return (tx2 >= x1 && tx1 <= x2 && ty2 >= y1 && ty1 <= y2);
117     }
118     double Enemy::ComputeDistance(int tx1, int ty1)
119     {
120         int temp = (x - tx1)*(x - tx1) + (y - ty1)*(y - ty1);
121         return sqrt(temp);
122     }
123 }

```

## Map.h

```

1  #pragma once
2  namespace game_framework {
3      class Map
4      {
5      public:
6          Map();
7          int  GetWidth();           //取得背景寬度
8          int  GetHeight();          //取得背景高度
9          int  ScreenX(int x);        // 螢幕 x 座標
10         int  ScreenY(int y);        // 螢幕 y 座標
11         void Initialize();           // 設定為初始值
12         void LoadBitmap(char*,COLORREF,char*,string); // 載入圖形
13         void OnMove(int,int);        // 地圖移動
14         void OnShow();               // 將圖形貼到畫面
15         void SetXY(int nx, int ny);   // 設定螢幕畫面左上角的座標
16         bool isEmpty(int x, int y);   // 判斷碰壁
17         bool isEmpty_2(int x, int y); // 判斷碰壁
18         bool isSlope(int, int);
19         void SetMouse(int ,int);      //設定滑鼠座標
20         void IsLclick(bool);          //設定左鍵按下
21         void IsRclick(bool);          //設定右鍵按下
22     protected:
23         CMovingBitmap background;     // 地圖
24         CMovingBitmap foreground;
25         CMovingBitmap ball_1,ball_2,ball_3; //顯示地形圖
26     private:
27         bool Lclick=false, Rclick=false; //滑鼠左鍵與右鍵是否按下 預設為否
28         int mx, my;                     //滑鼠座標
29         int sx, sy;                     //左上角座標
30         int mapSize_X, mapSize_Y;       //地圖所切的格數
31         int map[1000][1000];
32         string Mapfile;
33     };
34 }

```

## Map.cpp

```

1  #include "stdafx.h"
2  #include "Resource.h"
3  #include <mmsystem.h>
4  #include <ddraw.h>
5  #include "audio.h"
6  #include "gamelib.h"
7  #include "Map.h"
8  #include <fstream>
9  namespace game_framework {
10     Map::Map()
11     {
12     }
13     void Map::SetMouse(int x, int y) { //設定滑鼠座標
14         mx = x;
15         my = y;
16     }
17     void Map::IsLclick(bool flag) { //設定左鍵是否按下
18         Lclick= flag;
19         ofstream output(Mapfile);
20         for (int i = 0; i < mapSize_Y; i++) {
21             for (int j = 0; j < mapSize_X; j++) {
22                 output << map[i][j] << " ";
23             }
24             output << endl;
25         }
26         output.close();
27     }
28     void Map::IsRclick(bool flag) { //設定右鍵是否按下
29         Rclick= flag;
30         ofstream output(Mapfile);
31         for (int i = 0; i < mapSize_Y; i++) {
32             for (int j = 0; j < mapSize_X; j++) {
33                 output << map[i][j] << " ";
34             }
35             output << endl;

```

```

36         }
37         output.close();
38     }
39     int Map::GetWidth() {                //取得背景寬度
40         return foreground.Width();
41     }
42     int Map::GetHeight() {              //取得背景長度
43         return foreground.Height();
44     }
45     int Map::ScreenX(int x)             //回傳人物在螢幕上座標
46     {
47         return x-sx;
48     }
49     int Map::ScreenY(int y)             //同上
50     {
51         return y-sy;
52     }
53     void Map::Initialize()
54     {
55         //const int X_POS = 320;
56         //const int Y_POS = 240;
57         //sx = X_POS;
58         //sy = Y_POS;
59         ifstream file(Mapfile);        //將資料存入陣列
60         for (int i = 0; i<mapSize_Y; i++)
61         {
62             for (int j = 0; j < mapSize_X; j++) {
63                 file >> map[i][j];
64             }
65         }
66         file.close();
67     }
68     void Map::LoadBitmap(char* fore,COLORREF color,char* back,string file)
69     {
70         foreground.LoadBitmap(fore,color);
71         background.LoadBitmap(back);
72         Mapfile = file;
73         mapSize_X = (foreground.Width() * 10 + 5) / 100;

```

```

74         mapSize_Y = (foreground.Height() * 10 + 5) / 100;
75         ball_1.LoadBitmap("../RES//ball_1.bmp", RGB(0, 0, 0));
76         ball_2.LoadBitmap("../RES//ball_2.bmp", RGB(0, 0, 0));
77         ball_3.LoadBitmap("../RES//ball_3.bmp", RGB(0, 0, 0));
78     }
79     void Map::OnMove(int x, int y)          //地圖隨人物移動
80     {
81         sx = x - SIZE_X / 2;              //讓人物保持在地圖中間
82         sy = y - SIZE_Y / 2;
83         if (x <= SIZE_X / 2) {            //設定當人物接近邊界時地圖移動停止
84             sx = 0;
85         }
86         if (x >= foreground.Width() - SIZE_X / 2) {
87             sx = foreground.Width() - SIZE_X;
88         }
89         if (y <= SIZE_Y / 2) {
90             sy = 0;
91         }
92         if (y >= foreground.Height() - SIZE_Y / 2) {
93             sy = foreground.Height() - SIZE_Y;
94         }
95     }
96     void Map::OnShow()
97     {
98         background.SetTopLeft(0, 0);
99         background.ShowBitmap();
100        foreground.SetTopLeft(-sx, -sy); // 指定第(i, j)這一格的座標
101        foreground.ShowBitmap();
102        //ofstream set(Mapfile);          //第一次建立地圖設立初始值
103        //for (int i = 0; i < mapSize_Y; i++) {
104        //    for (int j = 0; j < mapSize_X; j++) {
105        //        set << 0 << " ";
106        //    }
107        //    set << endl;
108        //}
109        //set.close();
110        //for (int i = 0; i < mapSize_Y; i++) {
111        //    for (int j = 0; j < mapSize_X; j++) {

```



```

112         //          int x = j * 10 - sx; // 算出第(i,j)這一格的 x 螢幕座標
113         //          int y = i * 10 - sy; // 算出第(i,j)這一格的 y 螢幕座標
114         //          if (mx >= x && mx <= x + 10 && my >= y && my <= y + 10) { //判斷滑鼠位置
115         //              if (Lclick) {          //左鍵為設立障礙物
116         //                  map[i][j] = 1;
117         //              }
118         //              if (Lclick) {          //左鍵為設立可穿透的障礙物
119         //                  map[i][j] = 2;
120         //              }
121         //              if (Lclick) {          //左鍵為設立斜坡
122         //                  map[i][j] = 3;
123         //              }
124         //              if (Rclick) {          //右鍵為取消障礙物
125         //                  map[i][j] = 0;
126         //              }
127         //          }
128         //          switch (map[i][j]) {      //顯示格子
129         //              case 1:
130         //                  ball_1.SetTopLeft(x, y); // 指定第(i,j)這一格的座標
131         //                  ball_1.ShowBitmap();
132         //                  break;
133         //              case 2:
134         //                  ball_2.SetTopLeft(x, y);
135         //                  ball_2.ShowBitmap();
136         //                  break;
137         //              case 3:
138         //                  ball_3.SetTopLeft(x, y);
139         //                  ball_3.ShowBitmap();
140         //                  break;
141         //          }
142         //      }
143     //}
144 }
145 void Map::SetXY(int nx, int ny)
146 {
147     sx = nx;
148     sy = ny;
149 }

```

```
150     bool Map::isEmpty(int x, int y) // (x, y) 為地圖的點座標
151     {
152         int gx = x / 10; // 轉換為 X 軸格座標(整數除法)
153         int gy = y / 10; // 轉換為 Y 軸格座標(整數除法)
154         //map[x][y]中的 y 表示 X 軸的格數，x 表示 Y 軸
155         return map[gy][gx] == 0; // 假設 0 代表空的
156     }
157     bool Map::isEmpty_2(int x, int y) // (x, y) 為地圖的點座標
158     {
159         int gx = x / 10; // 轉換為 X 軸格座標(整數除法)
160         int gy = y / 10; // 轉換為 Y 軸格座標(整數除法)
161         //map[x][y]中的 y 表示 X 軸的格數，x 表示 Y 軸
162         return map[gy][gx] == 0 || map[gy][gx] == 2; // 假設 0 代表空的 2 代表可穿越
163     }
164     bool Map::isSlope(int x, int y) {
165         int gx = x / 10; // 轉換為 X 軸格座標(整數除法)
166         int gy = y / 10; // 轉換為 Y 軸格座標(整數除法)
167         //map[x][y]中的 y 表示 X 軸的格數，x 表示 Y 軸
168         return map[gy][gx] == 3; // 假設 3 代表斜坡
169     }
170 }
```

## Door.h

```

1  #pragma once
2  #include"Map.h"
3  #include"Kirby.h"
4  namespace game_framework {
5      class Door
6      {
7      public:
8          Door();
9          int  GetMapNum();           //取得地圖號碼
10         int  GetWidth();             //取得背景寬度
11         int  GetHeight();            //取得背景高度
12         int  GetX();
13         int  GetY();
14         int  GetNextX();
15         int  GetNextY();
16         int  GetX2();
17         int  GetY2();
18         void Initialize(int x,int y,int Num,int,Door* ); // 設定為初始值
19         void LoadBitmap();           // 載入圖形
20         void OnMove();               // 地圖移動
21         void OnShow(Map *);          // 將圖形貼到畫面
22         void SetEnter(bool flag);     // 設定是否正進入傳送門
23         bool IsEnter(Kirby*);        // 是否進入
24         Door GetNextDoor();
25     protected:
26         CAnimation door;             // 地圖
27         CAnimation Gate;
28     private:
29         bool Enter=false;            //是否在門按上 預設為否
30         int x, y;                    //左上角座標
31         int mapNum;                  //記錄此門傳送到幾號地圖
32         int map;                      //現在所在地圖
33         Door* NextDoor;
34     };
35 }

```

## Door.cpp

```
1  #include "stdafx.h"
2  #include "Resource.h"
3  #include <mmsystem.h>
4  #include <draw.h>
5  #include "audio.h"
6  #include "gamelib.h"
7  #include "Door.h"
8  #include <fstream>
9  namespace game_framework {
10     Door::Door() {
11     }
12     int Door::GetMapNum() {
13         return mapNum;
14     }
15     int Door::GetHeight() {
16         return door.Height();
17     }
18     int Door::GetWidth() {
19         return door.Width();
20     }
21     int Door::GetX() {
22         return x;
23     }
24     int Door::GetY() {
25         return y;
26     }
27     int Door::GetX2() {
28         return x + door.Width();
29     }
30     int Door::GetY2() {
31         return y + door.Height();
32     }
33     int Door::GetNextX() { //回傳下一個門的 X 座標
34         return NextDoor->GetX();
35     }
```

```

36     int Door::GetNextY() { //回傳下一個門的 Y 座標
37         return NextDoor->GetY();
38     }
39     Door Door::GetNextDoor() { //回傳指標
40         return *NextDoor;
41     }
42     void Door::Initialize(int px,int py,int num,int m ,Door* h) { //設定門的位置所在地圖下一個門的地圖
43         x = px;
44         y = py;
45         map = m;
46         mapNum = num;
47         NextDoor = h;
48     }
49     void Door::LoadBitmap() { . . . . . } //所有圖形的 addbitmap
50     void Door::OnMove() {
51         door.OnMove();
52         Gate.OnMove();
53         Gate.SetDelayCount(7);
54     }
55     void Door::OnShow(Map *m) {
56         if (map == 0) { //當為地圖 0 時為一般的門
57             door.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
58             door.OnShow();
59         }
60         else { //其餘為有星星的門
61             Gate.SetTopLeft(m->ScreenX(x), m->ScreenY(y));
62             Gate.OnShow();
63         }
64     }
65     void Door::SetEnter(bool flag) { //設定是否按上
66         Enter = flag;
67     }
68     bool Door::IsEnter(Kirby *k) {
69         if (map == 0) { //當為地圖 0 時為一般的門
70             if ((k->GetX1() + k->GetX2()) / 2 >= x && (k->GetX1() + k->GetX2()) / 2 <= x + door.Width() &&
71 (k->GetY1() + k->GetY2()) / 2 >= y && (k->GetY1() + k->GetY2()) / 2 <= y + door.Height() && Enter)
72                 return true;
73         else

```

```
74             return false;
75         }
76         else { //其餘為有星星的門
77             if ((k->GetX1() + k->GetX2()) / 2 >= x && (k->GetX1() + k->GetX2()) / 2 <= x + Gate.Width() &&
78 (k->GetY1() + k->GetY2()) / 2 >= y && (k->GetY1() + k->GetY2()) / 2 <= y + Gate.Height() && Enter)
79                 return true;
80             else
81                 return false;
82         }
83     }
84 }
```