**目錄**

[一、 簡介 - 1 -](#_Toc517234676)

[1. 動機 - 1 -](#_Toc517234677)

[2. 分工 - 1 -](#_Toc517234678)

[二、 遊戲介紹 - 2 -](#_Toc517234679)

[1. 遊戲說明 - 2 -](#_Toc517234680)

[a. 遊戲內容 - 2 -](#_Toc517234681)

[b. 遊戲操作 - 2 -](#_Toc517234682)

[c. 遊戲劇情 - 3 -](#_Toc517234683)

[d. 角色介紹 - 3 -](#_Toc517234684)

[2. 遊戲圖形 - 5 -](#_Toc517234685)

[3. 遊戲音效 - 9 -](#_Toc517234686)

[三、 程式設計 - 10 -](#_Toc517234687)

[1. 程式架構 - 10 -](#_Toc517234688)

[2. 程式類別 - 11 -](#_Toc517234689)

[3. 程式技術 - 12 -](#_Toc517234690)

[四、 結語 - 13 -](#_Toc517234691)

[1. 問題及解決方法 - 13 -](#_Toc517234692)

[2. 時間表 - 14 -](#_Toc517234693)

[3. 貢獻比例 - 15 -](#_Toc517234694)

[4. 檢核表 - 15 -](#_Toc517234695)

[5. 收穫 - 16 -](#_Toc517234696)

[6. 心得、感想 - 17 -](#_Toc517234697)

[7. 對於本課程的建議 - 18 -](#_Toc517234698)

[五、 附錄 - 19 -](#_Toc517234699)

[Mygame.h - 19 -](#_Toc517234700)

[Mygame.cpp - 25 -](#_Toc517234701)

# 簡介

## 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，其中鏡大之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。

## 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。

# 遊戲介紹

## 遊戲說明

### 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。

### 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

### 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。

### 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！

## 遊戲圖形

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 開始畫面 | | | 說明 | | |
| C:\Users\宇天\AppData\Local\Microsoft\Windows\INetCache\Content.Word\開始畫面.png | | | C:\Users\宇天\AppData\Local\Microsoft\Windows\INetCache\Content.Word\說明.png | | |
| 卡比 | | | | | |
| D:\kirby\game4.10\RES\Kirby\KB_R_0.bmp | | | | | |
| 行走動畫 | | | | | |
| D:\kirby\game4.10\RES\Kirby\KB_R_9.bmpD:\kirby\game4.10\RES\Kirby\KB_R_8.bmpD:\kirby\game4.10\RES\Kirby\KB_R_7.bmpD:\kirby\game4.10\RES\Kirby\KB_R_6.bmpD:\kirby\game4.10\RES\Kirby\KB_R_5.bmpD:\kirby\game4.10\RES\Kirby\KB_R_4.bmpD:\kirby\game4.10\RES\Kirby\KB_R_3.bmpD:\kirby\game4.10\RES\Kirby\KB_R_2.bmpD:\kirby\game4.10\RES\Kirby\KB_R_1.bmpD:\kirby\game4.10\RES\Kirby\KB_R_10.bmp | | | | | |
| 跑步動畫 | | | | | |
| D:\kirby\game4.10\RES\Kirby\RUN_R_6.bmpD:\kirby\game4.10\RES\Kirby\RUN_R_5.bmpD:\kirby\game4.10\RES\Kirby\RUN_R_4.bmpD:\kirby\game4.10\RES\Kirby\RUN_R_3.bmpD:\kirby\game4.10\RES\Kirby\RUN_R_2.bmpD:\kirby\game4.10\RES\Kirby\RUN_R_1.bmpD:\kirby\game4.10\RES\Kirby\RUN_R_0.bmpD:\kirby\game4.10\RES\Kirby\RUN_R_7.bmp | | | | | |
| 翻滾動畫 | | | | | |
| D:\kirby\game4.10\RES\Kirby\KB_Hurted_R_3.bmpD:\kirby\game4.10\RES\Kirby\KB_Hurted_R_2.bmpD:\kirby\game4.10\RES\Kirby\KB_Hurted_R_1.bmpD:\kirby\game4.10\RES\Kirby\KB_Hurted_R_4.bmp | | | | | |
| 飛行動畫 | | | | | |
| D:\kirby\game4.10\RES\Kirby\KB_U_R_10.bmpD:\kirby\game4.10\RES\Kirby\KB_U_R_9.bmpD:\kirby\game4.10\RES\Kirby\KB_U_R_8.bmpD:\kirby\game4.10\RES\Kirby\KB_U_R_7.bmpD:\kirby\game4.10\RES\Kirby\KB_U_R_6.bmpD:\kirby\game4.10\RES\Kirby\KB_U_R_11.bmp | | | | | |
| 跳躍 | 落下 | 吐 | | 踢擊 | 壓扁 |
| D:\kirby\game4.10\RES\Kirby\KB_Jump_R.bmp | D:\kirby\game4.10\RES\Kirby\KB_Landing_R.bmp | D:\kirby\game4.10\RES\Kirby\KB_Exhale_R.bmp | | D:\kirby\game4.10\RES\Kirby\KB_DownAttack_R.bmp | D:\kirby\game4.10\RES\Kirby\KB_Down_R.bmp |

|  |
| --- |
| 吸氣&吸怪動畫 |
| D:\kirby\game4.10\RES\Kirby\KB_Suck_R_3.bmpD:\kirby\game4.10\RES\Kirby\KB_Suck_R_2.bmpD:\kirby\game4.10\RES\Kirby\KB_Suck_R_1.bmpD:\kirby\game4.10\RES\Kirby\KB_Suck_R_4.bmp |
| 吸怪後的行走動畫 |
| D:\kirby\game4.10\RES\Kirby\BKB_R_7.bmpD:\kirby\game4.10\RES\Kirby\BKB_R_6.bmpD:\kirby\game4.10\RES\Kirby\BKB_R_5.bmpD:\kirby\game4.10\RES\Kirby\BKB_R_4.bmpD:\kirby\game4.10\RES\Kirby\BKB_R_3.bmpD:\kirby\game4.10\RES\Kirby\BKB_R_2.bmpD:\kirby\game4.10\RES\Kirby\BKB_R_8.bmp |
| 火焰卡比 |
| D:\kirby\game4.10\RES\FireKirby\FireKirby_Stand_R_3.bmp |
| 火焰卡比攻擊動畫 |
| D:\kirby\game4.10\RES\FireKirby\FireKirby_Attack_R_2.bmpD:\kirby\game4.10\RES\FireKirby\FireKirby_Attack_R_1.bmpD:\kirby\game4.10\RES\FireKirby\FireKirby_Attack_R_0.bmpD:\kirby\game4.10\RES\FireKirby\FireKirby_Attack_R_3.bmpD:\kirby\game4.10\RES\FireKirby\attack1_1.bmpD:\kirby\game4.10\RES\FireKirby\attack2_4.bmpD:\kirby\game4.10\RES\FireKirby\attack3_5.bmp |
| 閃電卡比 |
| D:\kirby\game4.10\RES\SparkKirby\Spark_stand_R_3.bmp |
| 閃電卡比攻擊動畫 |
| D:\kirby\game4.10\RES\SparkKirby\Spark_Attack_R_5.bmpD:\kirby\game4.10\RES\SparkKirby\Spark_Attack_R_4.bmpD:\kirby\game4.10\RES\SparkKirby\Spark_Attack_R_6.bmp |

|  |
| --- |
| 普通小怪行走動畫 |
| D:\kirby\game4.10\RES\NormalMonster\NormalMonster_R_6.bmpD:\kirby\game4.10\RES\NormalMonster\NormalMonster_R_5.bmpD:\kirby\game4.10\RES\NormalMonster\NormalMonster_R_4.bmpD:\kirby\game4.10\RES\NormalMonster\NormalMonster_R_3.bmpD:\kirby\game4.10\RES\NormalMonster\NormalMonster_R_2.bmpD:\kirby\game4.10\RES\NormalMonster\NormalMonster_R_1.bmpD:\kirby\game4.10\RES\NormalMonster\NormalMonster_R_7.bmp |
| 普通小怪死亡 |
| D:\kirby\game4.10\RES\NormalMonster\NormalMonster_Sucked_R.bmp |
| 火焰怪行走動畫 |
| D:\kirby\game4.10\RES\Fire\Fire_walk_R_4.bmpD:\kirby\game4.10\RES\Fire\Fire_walk_R_3.bmpD:\kirby\game4.10\RES\Fire\Fire_walk_R_2.bmpD:\kirby\game4.10\RES\Fire\Fire_walk_R_1.bmpD:\kirby\game4.10\RES\Fire\Fire_walk_R_0.bmpD:\kirby\game4.10\RES\Fire\Fire_walk_R_5.bmp |
| 火焰怪攻擊動畫 |
| D:\kirby\game4.10\RES\Fire\Fire_Attack_R_0.bmpD:\kirby\game4.10\RES\Fire\Fire_Attack_R_1.bmpD:\kirby\game4.10\RES\Fire\Fire_Attack_R_2.bmp |
| 火焰怪被吸動畫 |
| D:\kirby\game4.10\RES\Fire\Fire_sucked_L_0.bmpD:\kirby\game4.10\RES\Fire\Fire_sucked_L_1.bmp |
| 閃電怪跳躍動畫 |
| D:\kirby\game4.10\RES\Spirky\Spirky_jump_R_0.bmpD:\kirby\game4.10\RES\Spirky\Spirky_jump_R_1.bmpD:\kirby\game4.10\RES\Spirky\Spirky_jump_R_2.bmpD:\kirby\game4.10\RES\Spirky\Spirky_jump_R_3.bmpD:\kirby\game4.10\RES\Spirky\Spirky_jump_R_4.bmpD:\kirby\game4.10\RES\Spirky\Spirky_jump_R_5.bmpD:\kirby\game4.10\RES\Spirky\Spirky_jump_R_6.bmpD:\kirby\game4.10\RES\Spirky\Spirky_jump_R_7.bmp |
| 閃電怪攻擊動畫 |
| D:\kirby\game4.10\RES\Spirky\Spirky_Attack_8.bmpD:\kirby\game4.10\RES\Spirky\Spirky_Attack_6.bmpD:\kirby\game4.10\RES\Spirky\Spirky_Attack_14.bmpD:\kirby\game4.10\RES\Spirky\Spirky_Attack_17.bmp |
| 閃電怪被吸動畫 |
| D:\kirby\game4.10\RES\Spirky\Spirky_Sucked_R_0.pngD:\kirby\game4.10\RES\Spirky\Spirky_Sucked_R_1.png |

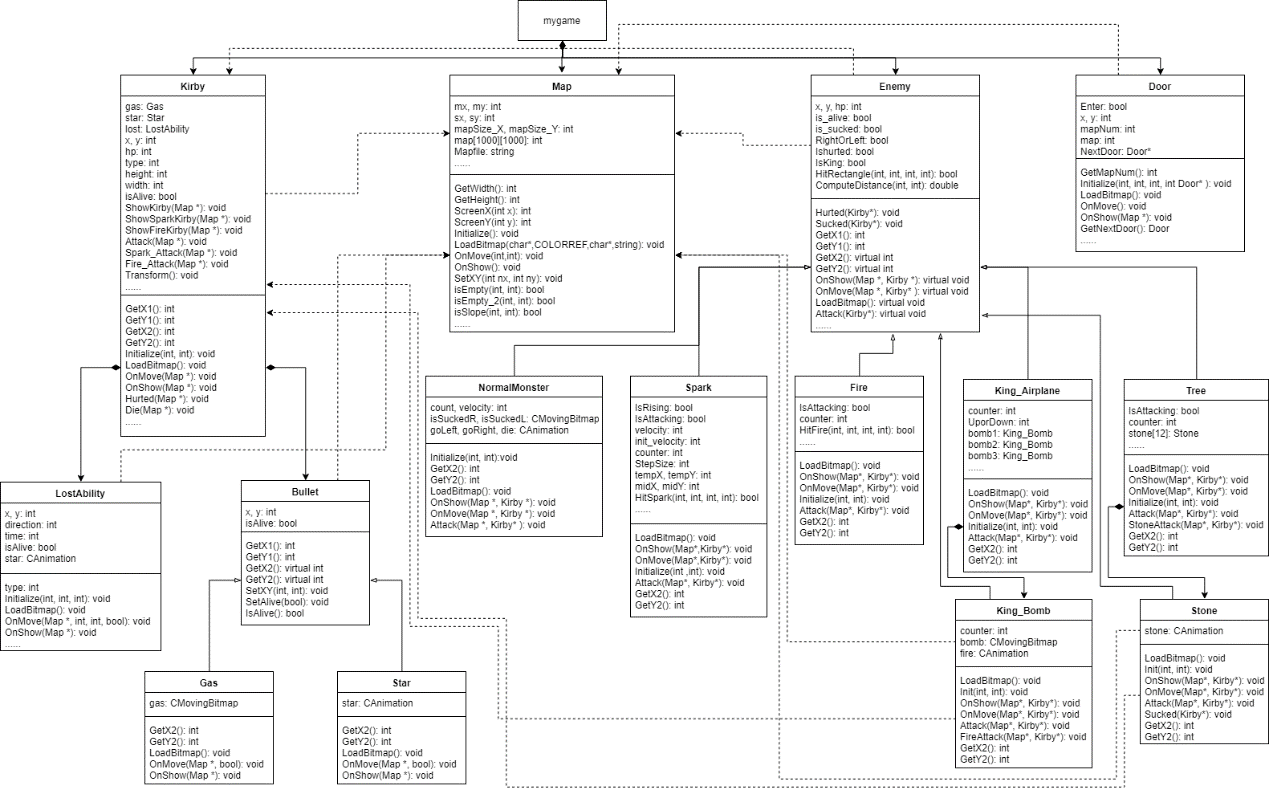
|  |  |
| --- | --- |
| 飛機王飛行動畫 | |
| D:\kirby\game4.10\RES\Bombar\Bombar_goR_3.bmpD:\kirby\game4.10\RES\Bombar\Bombar_goR_4.bmpD:\kirby\game4.10\RES\Bombar\Bombar_goR_6.bmpD:\kirby\game4.10\RES\Bombar\Bombar_goR_7.bmpD:\kirby\game4.10\RES\Bombar\Bombar_goR_9.bmpD:\kirby\game4.10\RES\Bombar\Bombar_goR_11.bmp | |
| 樹王 | |
| SD:\kirby\截圖\未命名.png | |
| 初始地圖 | 破關後的地圖 |
|  |  |
| 過關動畫 | |
|  | |

## 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 |
| hurted.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 |

# 程式設計

## 程式架構



## 程式類別

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

## 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。

# 結語

## 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。

## 時間表

|  |  |  |  |
| --- | --- | --- | --- |
| 週 | 每週工作時間(小時) | | 總和 |
| 周宇天 | 林彥廷 |
| 0 | 3 | 3 | 6 |
| 1 | 4 | 14 | 18 |
| 2 | 5.5 | 14 | 19.5 |
| 3 | 4.5 | 2 | 6.5 |
| 4 | 6.5 | 11 | 17.5 |
| 5 | 5 | 2 | 7 |
| 6 | 3.5 | 4 | 7.5 |
| 7 | 7.5 | 7 | 14.5 |
| 8 | 5 | 7 | 12 |
| 9 | 8.5 | 9 | 17.5 |
| 10 | 2 | 3 | 5 |
| 11 | 3 | 5 | 8 |
| 12 | 4 | 6.5 | 10.5 |
| 13 | 3 | 5 | 8 |
| 14 | 10 | 5 | 15 |
| 總和 | **75** | **97.5** | **172.5** |

## 貢獻比例

林彥廷：５０％

周宇天：５０％

## 檢核表

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

## 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。

## 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

## 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。

# 附錄

## Mygame.h

#include "Kirby.h"

#include "Map.h"

#include "NormalMonster.h"

#include "Door.h"

#include "Gas.h"

#include "Star.h"

#include "LostAbility.h"

#include "Spark.h"

#include "Fire.h"

#include "Tree.h"

#include "King\_Airplane.h"

namespace game\_framework {

enum AUDIO\_ID { // 定義各種音效的編號

AUDIO\_BACKGROUND, // 0

jump, // 1

landing, // 2

kick, // 3

die, // 4

fly, // 5

hurted, // 6

run, // 7

suck, // 8

gasSound, // 9

starSound, // 10

swallow, // 11

spark, // 12

fire, // 13

stone, // 14

start, // 15

enemyDie, // 16

lostAbility, // 17

ending // 18

};

class CGameStateInit : public CGameState {

public:

CGameStateInit(CGame \*g);

void OnInit(); // 遊戲的初值及圖形設定

void OnBeginState(); // 設定每次重玩所需的變數

void OnKeyUp(UINT, UINT, UINT); // 處理鍵盤Up的動作

void OnLButtonDown(UINT nFlags, CPoint point); // 處理滑鼠的動作

protected:

void OnShow(); // 顯示這個狀態的遊戲畫面

private:

int page;

CAnimation StartAnimation;

CMovingBitmap background, pressStart;

};

class CGameStateRun : public CGameState {

public:

CGameStateRun(CGame \*g);

~CGameStateRun();

void OnBeginState(); // 設定每次重玩所需的變數

void OnInit(); // 遊戲的初值及圖形設定

void OnKeyDown(UINT, UINT, UINT);

void OnKeyUp(UINT, UINT, UINT);

void OnMouseMove(UINT nFlags, CPoint point); // 處理滑鼠的動作

void OnLButtonDown(UINT nFlags, CPoint point);// 處理滑鼠的動作

void OnLButtonUp(UINT nFlags, CPoint point); // 處理滑鼠的動作

void OnRButtonDown(UINT nFlags, CPoint point);// 處理滑鼠的動作

void OnRButtonUp(UINT nFlags, CPoint point); // 處理滑鼠的動作

void ResetMonster();

protected:

void OnMove(); // 移動遊戲元素

void OnShow(); // 顯示這個狀態的遊戲畫面

private:

Kirby kirby; // 卡比

Map map[5]; // 地圖

Map \*index;

Enemy \*monster[10]; // 怪物

Enemy \*Boss;

NormalMonster normalMonster1[5],normalMonster4[1];

Spark spark1[4],spark4[1];

Fire fire1[2], fire4[1];

Tree tree;

King\_Airplane airplane;

Door door[10];

Door door1[2];

Door door2;

Door door3;

Door door4[2];

Door \*gate;

int mapNum; //設定現在為第幾號地圖

int Mirror\_L\_Y,Mirror\_R\_Y;

int end\_Y;

CAnimation Transition;

CAnimation Mirror\_L, Mirror\_R;

CMovingBitmap end;

bool Istransiting = false;

bool MovingMirror = false;

bool Show\_Mirror\_L = false;

bool Show\_Mirror\_R = false;

bool isEnd = false;

};

class CGameStateOver : public CGameState {

public:

CGameStateOver(CGame \*g);

void OnBeginState(); // 設定每次重玩所需的變數

void OnInit();

protected:

void OnMove(); // 移動遊戲元素

void OnShow(); // 顯示這個狀態的遊戲畫面

private:

int counter; // 倒數之計數器

};

## Mygame.cpp

#include "stdafx.h"

#include "Resource.h"

#include <mmsystem.h>

#include <ddraw.h>

#include "audio.h"

#include "gamelib.h"

#include "mygame.h"

namespace game\_framework {

// 這個class為遊戲的遊戲開頭畫面物件

CGameStateInit::CGameStateInit(CGame \*g)

: CGameState(g)

{

}

void CGameStateInit::OnInit()

{

StartAnimation.AddBitmap(".\\RES\\Start\\title\_0.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_1.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_2.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_3.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_4.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_5.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_6.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_7.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_8.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_9.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_10.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_11.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_12.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_13.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_14.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_15.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_16.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_17.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_18.bmp", RGB(255, 255, 255));

StartAnimation.AddBitmap(".\\RES\\Start\\title\_19.bmp", RGB(255, 255, 255));

background.LoadBitmap(".\\RES\\Start\\background.bmp");

pressStart.LoadBitmap(".\\RES\\Start\\Press\_start.bmp", RGB(248, 248, 248));

}

void CGameStateInit::OnBeginState()

{

page = 0;

if(!CAudio::Instance()->IsLoaded(start))

CAudio::Instance()->Load(start, "Sounds\\start.mp3");

CAudio::Instance()->Play(start, true);

}

void CGameStateInit::OnKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags)

{

const char KEY\_ESC = 27;

const char KEY\_SPACE = ' ';

if (nChar == KEY\_SPACE)

{

if (page == 0)

page++;

else

{

CAudio::Instance()->Stop(start);

GotoGameState(GAME\_STATE\_RUN); // 切換至GAME\_STATE\_RUN

}

}

else if (nChar == KEY\_ESC) // Demo 關閉遊戲的方法

PostMessage(AfxGetMainWnd()->m\_hWnd, WM\_CLOSE,0,0); // 關閉遊戲

}

void CGameStateInit::OnLButtonDown(UINT nFlags, CPoint point)

{

if (page == 0)

page++;

else

{

CAudio::Instance()->Stop(start);

GotoGameState(GAME\_STATE\_RUN); // 切換至GAME\_STATE\_RUN

}

}

void CGameStateInit::OnShow()

{

if (page == 0)

{

background.SetTopLeft(0, 0);

StartAnimation.SetTopLeft(0, 0);

pressStart.SetTopLeft(60, 380);

background.ShowBitmap();

StartAnimation.OnMove();

StartAnimation.OnShow();

pressStart.ShowBitmap();

}

else

{

// /\*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\*/

CDC \*pDC = CDDraw::GetBackCDC(); // 取得 Back Plain 的 CDC

CFont f,\*fp;

f.CreatePointFont(160,"Times New Roman"); // 產生 font f; 160表示16 point的字

fp=pDC->SelectObject(&f); // 選用 font f

pDC->SetBkColor(RGB(0, 0, 0));

pDC->SetTextColor(RGB(255 , 255, 0));

pDC->TextOut(35, 50, "操作 : ");

pDC->TextOut(65, 80, "↑ : 吸氣飛翔、進入傳送門 ↓ : 蹲下、變身");

pDC->TextOut(65, 110, "← : 左移 → : 右移");

pDC->TextOut(65, 140, "Z : 攻擊、吸怪 ↓ + Z : 踢擊");

pDC->TextOut(65, 170, "X : 跳躍");

pDC->TextOut(65, 200, "C : 跑步(按住加左右移)");

pDC->TextOut(65, 230, "Esc : 關閉遊戲");

pDC->TextOut(35, 260, "密技 : ");

pDC->TextOut(65, 290, "S : 血量補滿");

pDC->TextOut(35, 350, "打敗兩隻魔王，拿回碎片拯救世界吧 ! 卡比");

pDC->TextOut(220, 380, "(按下空白鍵開始)");

pDC->SelectObject(fp); // 放掉 font f (千萬不要漏了放掉)

CDDraw::ReleaseBackCDC(); // 放掉 Back Plain 的 CDC\*/

}

}

CGameStateOver::CGameStateOver(CGame \*g)

: CGameState(g)

{

}

void CGameStateOver::OnMove()

{

if(counter==0)

GotoGameState(GAME\_STATE\_INIT);

}

void CGameStateOver::OnBeginState()

{

counter = 300;

}

void CGameStateOver::OnInit()

{

}

void CGameStateOver::OnShow()

{

counter--;

CDC \*pDC = CDDraw::GetBackCDC(); // 取得 Back Plain 的 CDC

CFont f,\*fp;

f.CreatePointFont(160,"Times New Roman"); // 產生 font f; 160表示16 point的字

fp=pDC->SelectObject(&f); // 選用 font f

pDC->SetBkColor(RGB(0,0,0));

pDC->SetTextColor(RGB(255,255,0));

char str[80]; // Demo 數字對字串的轉換

sprintf(str, "Game Over ! (%d)",counter/30);

pDC->TextOut(240,210,str);

pDC->SelectObject(fp); // 放掉 font f (千萬不要漏了放掉)

CDDraw::ReleaseBackCDC(); // 放掉 Back Plain 的 CDC

}

CGameStateRun::CGameStateRun(CGame \*g)

: CGameState(g)

{

}

CGameStateRun::~CGameStateRun()

{

}

void CGameStateRun::OnBeginState()

{

map[0].Initialize();

map[1].Initialize();

map[2].Initialize();

map[3].Initialize();

map[4].Initialize();

Mirror\_L.SetTopLeft(608, -100);

Mirror\_R.SetTopLeft(608, -100);

kirby.Initialize(640,400);

door[0].Initialize(123, 37, 1, 0, &door1[0]);

door[1].Initialize(621, 37, 2, 0, &door2);

door[2].Initialize(1118, 37, 4, 0, &door4[0]);

door[3].Initialize(83, 328, 4, 0, &door4[1]);

door[4].Initialize(248, 369, 4, 0, &door4[0]);

door[5].Initialize(993, 367, 1, 0, &door1[0]);

door[6].Initialize(1159, 326, 1, 0, &door1[1]);

door[7].Initialize(207, 575, 1, 0, &door1[1]);

door[8].Initialize(620, 614, 3, 0, &door3);

door[9].Initialize(1036, 574, 4, 0, &door4[1]);

door1[0].Initialize(30, 425, 0,1, &door[5]);

door1[1].Initialize(4450, 350, 0, 1, &door[6]);

door2.Initialize(320, 240, 0, 2, &door[1]);

door3.Initialize(150, 350, 0, 3, &door[8]);

door4[0].Initialize(48, 435, 0, 4, &door[4]);

door4[1].Initialize(3140, 240 , 0, 4, &door[3]);

ResetMonster();

tree.Initialize(450, 100);

airplane.Initialize(450, 60);

Mirror\_L\_Y = -10;

Mirror\_R\_Y = -10;

end\_Y = 0;

mapNum = 0;

index = &map[mapNum];

gate = &door[5];

CAudio::Instance()->Play(AUDIO\_BACKGROUND, true);

}

void CGameStateRun::OnMove() // 移動遊戲元素

{

index->OnMove(kirby.GetX1(), kirby.GetY1());

Mirror\_R.OnMove();

Mirror\_L.OnMove();

if (Istransiting) {

Transition.OnMove();

Transition.SetDelayCount(4);

}

else if (MovingMirror) {

map[0].SetXY(320, 160);

if (Show\_Mirror\_R && Mirror\_R\_Y != 327) {

Mirror\_R\_Y += 1;

Mirror\_R.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror\_R\_Y));

}

else if (Show\_Mirror\_L && Mirror\_L\_Y != 327 ) {

Mirror\_L\_Y += 1;

Mirror\_L.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror\_L\_Y));

}

else {

MovingMirror = false;

}

}

else if (isEnd)

{

end\_Y--;

if (end\_Y == -1)

{

CAudio::Instance()->Stop(AUDIO\_BACKGROUND);

CAudio::Instance()->Play(ending);

}

else if (end\_Y == -1050)

{

CAudio::Instance()->Stop(ending);

CAudio::Instance()->Play(AUDIO\_BACKGROUND);

isEnd = false;

end\_Y = 1;

}

}

else {

if (kirby.IsAlive())

kirby.OnMove(index);

if (mapNum == 0)

{

for (int i = 0; i < 10; i++) {

door[i].OnMove();

if (door[i].IsEnter(&kirby)) {

door[i].SetEnter(false);

Istransiting = true;

Transition.Reset();

gate = &door[i];

}

}

}

if (mapNum == 1) {

for (int m = 0; m < 10; m++)

monster[m]->OnMove(index, &kirby);

for (int i = 0; i < 2; i++) {

door1[i].OnMove();

if (door1[i].IsEnter(&kirby)) {

door1[i].SetEnter(false);

Istransiting = true;

Transition.Reset();

gate = &door1[i];

}

}

}

if (mapNum == 2) {

airplane.OnMove(index,&kirby);

if (!airplane.IsAlive())

{

if (!Show\_Mirror\_R) {

MovingMirror = true;

Show\_Mirror\_R = true;

Istransiting = true;

Transition.Reset();

Mirror\_L.Reset();

Mirror\_R.Reset();

}

else {

door2.OnMove();

if (door2.IsEnter(&kirby)) {

door2.SetEnter(false);

Istransiting = true;

Transition.Reset();

gate = &door2;

}

}

}

}

if (mapNum == 3) {

tree.OnMove(index,&kirby);

if (!tree.IsAlive()) {

if (!Show\_Mirror\_L) {

MovingMirror = true;

Show\_Mirror\_L = true;

Istransiting = true;

Transition.Reset();

}

else {

door3.OnMove();

if (door3.IsEnter(&kirby)) {

door3.SetEnter(false);

Istransiting = true;

Transition.Reset();

gate = &door3;

}

}

}

}

if (mapNum == 4) {

door4[0].OnMove();

door4[1].OnMove();

for (int m = 0; m < 3; m++)

monster[m]->OnMove(index, &kirby);

for (int i = 0; i < 2; i++) {

if (door4[i].IsEnter(&kirby)) {

door4[i].SetEnter(false);

Istransiting = true;

Transition.Reset();

gate = &door4[i];

}

}

}

}

if (Transition.IsFinalBitmap()) {

Istransiting = false;

}

if (Transition.GetCurrentBitmapNumber() == 7 && !MovingMirror) {

mapNum = gate->GetMapNum();

index = &map[mapNum];

ResetMonster();

kirby.SetXY(gate->GetNextDoor().GetX() - 50, gate->GetNextDoor().GetY2()-kirby.GetHeight());

}

}

void CGameStateRun::OnInit() // 遊戲的初值及圖形設定

{

Transition.AddBitmap(".//Map//Transition\_7.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_6.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_5.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_4.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_3.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_2.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_1.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_0.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_1.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_2.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_3.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_4.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_5.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_6.bmp", RGB(0, 0, 0));

Transition.AddBitmap(".//Map//Transition\_7.bmp", RGB(0, 0, 0));

Mirror\_L.AddBitmap(".//RES//Door//mirror\_L\_0.bmp", RGB(255, 255, 255));

Mirror\_L.AddBitmap(".//RES//Door//mirror\_L\_1.bmp", RGB(255, 255, 255));

Mirror\_L.AddBitmap(".//RES//Door//mirror\_L\_2.bmp", RGB(255, 255, 255));

Mirror\_L.AddBitmap(".//RES//Door//mirror\_L\_3.bmp", RGB(255, 255, 255));

Mirror\_R.AddBitmap(".//RES//Door//mirror\_R\_0.bmp", RGB(255, 255, 255));

Mirror\_R.AddBitmap(".//RES//Door//mirror\_R\_1.bmp", RGB(255, 255, 255));

Mirror\_R.AddBitmap(".//RES//Door//mirror\_R\_2.bmp", RGB(255, 255, 255));

Mirror\_R.AddBitmap(".//RES//Door//mirror\_R\_3.bmp", RGB(255, 255, 255));

end.LoadBitmap(".//RES//end.bmp");

map[0].LoadBitmap(".//Map//foreground.bmp", RGB(255, 255, 255), ".//Map//background.bmp", ".//Map//map.txt");

map[1].LoadBitmap(".//Map//map1.bmp", RGB(255, 255, 255), ".//Map//background\_1.bmp", ".//Map//map1.txt");

map[2].LoadBitmap(".//Map//Boss\_map.bmp", RGB(255, 255,255),".//Map//background\_2.bmp", ".//Map//map2.txt");

map[3].LoadBitmap(".//Map//King\_foreground.bmp",RGB(255,255,255),".//Map//King\_background.bmp"

,".//Map//map3.txt");

map[4].LoadBitmap(".//Map//foreground\_4.bmp",RGB(255,255,255),".//Map//background\_4.bmp"

,".//Map//map4.txt");

kirby.LoadBitmap();

monster[0] = &fire1[0];

monster[1] = &fire1[1];

monster[2] = &normalMonster1[0];

monster[3] = &normalMonster1[1];

monster[4] = &normalMonster1[2];

monster[5] = &normalMonster1[3];

monster[6] = &spark1[0];

monster[7] = &spark1[1];

monster[8] = &spark1[2];

monster[9] = &spark1[3];

fire4[0].LoadBitmap();

spark4[0].LoadBitmap();

normalMonster4[0].LoadBitmap();

for (int m = 0; m < 10; m++)

monster[m]->LoadBitmap();

for (int i = 0; i < 10;i++)

door[i].LoadBitmap();

door1[0].LoadBitmap();

door1[1].LoadBitmap();

door2.LoadBitmap();

door3.LoadBitmap();

for (int i = 0; i < 2; i++)

door4[i].LoadBitmap();

tree.LoadBitmap();

airplane.LoadBitmap();

CAudio::Instance()->Load(AUDIO\_BACKGROUND, "Sounds\\Kirby\_background.mp3"); //背景音樂

CAudio::Instance()->Load(jump, "Sounds\\jump.wav");

CAudio::Instance()->Load(landing, "Sounds\\landing.wav");

CAudio::Instance()->Load(kick, "Sounds\\kick.wav");

CAudio::Instance()->Load(die, "Sounds\\die.wav");

CAudio::Instance()->Load(fly, "Sounds\\fly.wav");

CAudio::Instance()->Load(hurted, "Sounds\\hurted.wav");

CAudio::Instance()->Load(run, "Sounds\\run.wav");

CAudio::Instance()->Load(suck, "Sounds\\suck.wav");

CAudio::Instance()->Load(gasSound, "Sounds\\gas.wav");

CAudio::Instance()->Load(starSound, "Sounds\\star.wav");

CAudio::Instance()->Load(swallow, "Sounds\\swallow.wav");

CAudio::Instance()->Load(spark, "Sounds\\spark.wav");

CAudio::Instance()->Load(fire, "Sounds\\fire.wav");

CAudio::Instance()->Load(enemyDie, "Sounds\\enemyDie.wav");

CAudio::Instance()->Load(lostAbility, "Sounds\\lostAbility.wav");

CAudio::Instance()->Load(ending, "Sounds\\ending.mp3");

}

void CGameStateRun::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)

{

const char KEY\_ESC = 27; // keyboard Esc

const char KEY\_LEFT = 0x25; // keyboard左箭頭

const char KEY\_UP = 0x26; // keyboard上箭頭

const char KEY\_RIGHT = 0x27; // keyboard右箭頭

const char KEY\_DOWN = 0x28; // keyboard下箭頭

const char KEY\_Jump = 0x58; // keyboard X鍵

const char KEY\_Attack = 0x5A; // keyboard Z鍵

const char KEY\_Run = 0x43; // keyboard C鍵

const char KEY\_Restore = 0x53; // keyboard S鍵

if (nChar == KEY\_LEFT)

kirby.SetMovingLeft(true);

if (nChar == KEY\_RIGHT)

kirby.SetMovingRight(true);

if (nChar == KEY\_UP) {

kirby.SetMovingUp(true);

if(mapNum==0)

for (int i = 0; i < 10;i++)

door[i].SetEnter(true);

if (mapNum == 1)

for (int i = 0; i < 2;i++)

door1[i].SetEnter(true);

if (mapNum == 2)

door2.SetEnter(true);

if (mapNum == 3)

door3.SetEnter(true);

if (mapNum == 4)

for (int i = 0; i < 2; i++)

door4[i].SetEnter(true);

}

if (nChar == KEY\_DOWN)

{

kirby.SetMovingDown(true);

if (kirby.IsBig())

CAudio::Instance()->Play(swallow);

}

if (nChar == KEY\_Attack)

{

kirby.SetAttack(true);

if (!MovingMirror)

{

if (kirby.IsFly())

CAudio::Instance()->Play(gasSound);

else if (kirby.IsDown())

CAudio::Instance()->Play(kick);

else if (kirby.GetType() == 0 && !kirby.IsBig())

CAudio::Instance()->Play(suck);

else if (kirby.GetType() == 0 && kirby.IsBig())

CAudio::Instance()->Play(starSound);

else if (kirby.GetType() == 1)

CAudio::Instance()->Play(spark);

else if (kirby.GetType() == 2)

CAudio::Instance()->Play(fire);

}

}

if (nChar == KEY\_Run)

{

kirby.SetRun(true);

if (!kirby.IsFly()&&!index->isEmpty((kirby.GetX1()+kirby.GetX2())/2, kirby.GetY2() + 1) && kirby.IsMove())

CAudio::Instance()->Play(run);

}

if (nChar == KEY\_Jump && !kirby.IsFly() && (!index->isEmpty(kirby.GetX1(), kirby.GetY2() + 1) || !index->isEmpty(kirby.GetX2(), kirby.GetY2() + 1))) //按下X,卡比不是在飛行且落地才可跳躍

kirby.SetJump(true);

if (nChar == KEY\_ESC)

PostMessage(AfxGetMainWnd()->m\_hWnd, WM\_CLOSE, 0, 0); // 關閉遊戲

if (nChar == KEY\_Restore)

kirby.Restore();

}

void CGameStateRun::OnKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags)

{

const char KEY\_LEFT = 0x25; // keyboard左箭頭

const char KEY\_UP = 0x26; // keyboard上箭頭

const char KEY\_RIGHT = 0x27; // keyboard右箭頭

const char KEY\_DOWN = 0x28; // keyboard下箭頭

const char KEY\_Attack = 0x5A; // keyboard Z鍵

const char KEY\_Run = 0x43; // keyboard C鍵

if (nChar == KEY\_LEFT)

kirby.SetMovingLeft(false);

if (nChar == KEY\_RIGHT)

kirby.SetMovingRight(false);

if (nChar == KEY\_UP) {

kirby.SetMovingUp(false);

if(mapNum==0)

for (int i = 0; i < 10;i++)

door[i].SetEnter(false);

if (mapNum == 1)

for (int i = 0; i < 2; i++)

door1[i].SetEnter(false);

if (mapNum == 2)

door2.SetEnter(false);

if (mapNum==3)

door3.SetEnter(false);

if (mapNum == 4)

for (int i = 0; i < 2; i++)

door4[i].SetEnter(false);

}

if (nChar == KEY\_DOWN)

kirby.SetMovingDown(false);

if (nChar == KEY\_Attack)

{

kirby.SetAttack(false);

kirby.SetSuck(false);

CAudio::Instance()->Stop(suck);

CAudio::Instance()->Stop(spark);

CAudio::Instance()->Stop(fire);

}

if (nChar == KEY\_Run)

kirby.SetRun(false);

}

void CGameStateRun::OnMouseMove(UINT nFlags, CPoint point) // 處理滑鼠的動作

{

// 沒事。如果需要處理滑鼠移動的話，寫code在這裡

index->SetMouse(point.x, point.y);

}

void CGameStateRun::OnLButtonDown(UINT nFlags, CPoint point) // 處理滑鼠的動作

{

index->IsLclick(true);

}

void CGameStateRun::OnLButtonUp(UINT nFlags, CPoint point) // 處理滑鼠的動作

{

index->IsLclick(false);

}

void CGameStateRun::OnRButtonDown(UINT nFlags, CPoint point) // 處理滑鼠的動作

{

index->IsRclick(true);

}

void CGameStateRun::OnRButtonUp(UINT nFlags, CPoint point) // 處理滑鼠的動作

{

index->IsRclick(false);

}

void CGameStateRun::ResetMonster()

{

if (mapNum == 1) {

normalMonster1[0].Initialize(947, 467);

normalMonster1[1].Initialize(2459, 517);

normalMonster1[2].Initialize(2903, 517);

normalMonster1[3].Initialize(3495, 59);

spark1[0].Initialize(675, 393);

spark1[1].Initialize(1673, 163);

spark1[2].Initialize(3247, 269);

spark1[3].Initialize(4045, 477);

fire1[0].Initialize(417, 467);

fire1[1].Initialize(3745, 477);

monster[0] = &fire1[0];

monster[1] = &fire1[1];

monster[2] = &normalMonster1[0];

monster[3] = &normalMonster1[1];

monster[4] = &normalMonster1[2];

monster[5] = &normalMonster1[3];

monster[6] = &spark1[0];

monster[7] = &spark1[1];

monster[8] = &spark1[2];

monster[9] = &spark1[3];

}

if (mapNum == 4) {

fire4[0].Initialize(410, 400);

spark4[0].Initialize(1125, 525);

normalMonster4[0].Initialize(1600, 525);

monster[0] = &fire4[0];

monster[1] = &spark4[0];

monster[2] = &normalMonster4[0];

}

}

void CGameStateRun::OnShow()

{

// 注意：Show裡面千萬不要移動任何物件的座標，移動座標的工作應由Move做才對，

// 否則當視窗重新繪圖時(OnDraw)，物件就會移動，看起來會很怪。換個術語

// 說，Move負責MVC中的Model，Show負責View，而View不應更動Model。

end.SetTopLeft(0, end\_Y);

index->OnShow();

if (mapNum == 0) {

for (int i = 0; i < 10;i++)

door[i].OnShow(index);

if (Show\_Mirror\_L) {

Mirror\_L.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror\_L\_Y));

Mirror\_L.OnShow();

}

if (Show\_Mirror\_R) {

Mirror\_R.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror\_R\_Y));

Mirror\_R.OnShow();

}

}

else if (mapNum == 1) {

for (int i = 0; i < 2;i++)

door1[i].OnShow(index);

for (int m = 0; m < 10; m++)

monster[m]->OnShow(index, &kirby);

}

else if (mapNum == 2) {

airplane.OnShow(index,&kirby);

if(!airplane.IsAlive())

door2.OnShow(index);

}

else if (mapNum == 3) {

tree.OnShow(index,&kirby);

if (!tree.IsAlive()) {

door3.OnShow(index);

}

}

else if(mapNum == 4) {

for (int i = 0; i < 2; i++)

door4[i].OnShow(index);

for (int m = 0; m < 3; m++)

monster[m]->OnShow(index, &kirby);

}

if (kirby.IsAlive()) {

kirby.OnShow(index);

}

else if (kirby.GetY1() <= 1)

{

// 卡比死掉就不用播消失能力的音效

CAudio::Instance()->Stop(lostAbility);

// 鏡子重置

Show\_Mirror\_L = false;

Show\_Mirror\_R = false;

Mirror\_L\_Y = -10;

Mirror\_R\_Y = -10;

GotoGameState(GAME\_STATE\_OVER);

}

else

kirby.Die(index);

if (MovingMirror && Transition.GetCurrentBitmapNumber()>6) {

map[0].SetXY(320, 160);

map[0].OnShow();

if (Show\_Mirror\_L) {

Mirror\_L.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror\_L\_Y));

Mirror\_L.OnShow();

}

if (Show\_Mirror\_R) {

Mirror\_R.SetTopLeft(map[0].ScreenX(608), map[0].ScreenY(Mirror\_R\_Y));

Mirror\_R.OnShow();

}

}

if (Istransiting) {

Transition.OnShow();

}

if (Show\_Mirror\_L && Show\_Mirror\_R && Mirror\_L\_Y == 327 && Mirror\_R\_Y == 327 && end\_Y == 0)

isEnd = true;

if (isEnd)

end.ShowBitmap();

}

}