

CMoney 程式菁英軟體工程師戰鬥營

期中專題成果報告



團隊名稱：SuperDragon

指導老師：賴立本

組員姓名：郭恩宏、鐘怡茜

中華民國 107 年 11 月 14 日

目錄

第一章	專案簡介.....	5
1.1	遊戲設計理念.....	5
1.2	遊戲故事背景.....	5
第二章	開發環境.....	6
第三章	需求分析.....	7
3.1	系統目標.....	7
3.2	系統範圍.....	7
3.3	系統整體描述.....	7
3.4	功能需求.....	8
3.5	非功能需求.....	8
第四章	系統分析與設計.....	9
4.1	系統平台架構.....	9
4.3	活動圖.....	11
4.3 - 1	故事模式活動圖.....	11
4.3 - 2	選單活動圖.....	12
4.3 - 3	第一關活動圖.....	13
4.3 - 4	第二關活動圖.....	13
4.3 - 5	第三關活動圖.....	14

4.4 類別圖	16
第五章 團隊組成與分工.....	19
第六章 實作方法.....	20
6.1 Region 與中心點繪圖之結合	20
6.2 舞台、角色、功能性物件	21
6.3 轉盤設計	22
6.4 火焰對抗設計	23
6.5 平衡木設計	24
第七章 專案成果.....	25
第八章 討論與結論.....	28

圖目錄

圖 1 系統平台架構.....	9
圖 2 使用個案圖	10
圖 3 故事模式活動圖	11
圖 4 選單活動圖	12
圖 5 第一關活動圖.....	13
圖 6 第二關活動圖.....	14
圖 7 第三關活動圖.....	15
圖 8 系統類別圖	16
圖 9 Actor 父類的 Region.....	20
圖 10 製作者模式下能輕易看出繪圖範圍，以及物件的具體數量	21
圖 11 轉盤實作成果(註: 上圖輪盤是賺取隨機的額外生命值 $5 + (5)$)	22
圖 12 火焰對抗實作成果	23
圖 13 平衡遊戲.....	24

表目錄

表 1 系統開發工具表	6
表 2 類別說明	17
表 3 團隊組成與分工表	19
表 4 分工比重表	19

第一章 專案簡介

1.1 遊戲設計理念

遊戲【SuperDragon】發想源自於過去的 Scratch 遊戲專案，整體內容以故事性的方式貫穿整個架構。玩家在體驗的過程中，以輔助故事角色的方式來體驗整個遊戲劇情。【SuperDragon】定位屬於 AVG (冒險遊戲 Adventure Game) 的一種，需要在 PC 上執行。有別於傳統的橫向卷軸模式，因玩家為輔助者，所以無法透過鍵盤控制角色的移動。

遊戲【SuperDragon】全程以滑鼠事件為主，透過**點擊**或**移動**的方式，拚取**最短**的通關成績。過程需要玩家大量點擊滑鼠左(右)鍵，能藉此紓發平日積累之壓力。【SuperDragon】畫風採用小朋友都會喜愛的風格，訴求不分年齡層的設計。本款遊戲主軸，主打可愛的遊戲畫風、鍛鍊手指運動為主要方針來開發。

1.2 遊戲故事背景

在一座充滿各種奇幻生物的島上，一對恐龍兄妹居住在森林湖畔附近。有著鮮綠色的噴火龍 Melon 和一身粉色龍鱗的妹妹 Tiffany 快樂的生活著。

某日，身為妹控的 Melon 回家後發現，鮮少外出的妹妹不見了。Melon 內心開始感到焦躁不安，四處尋找親愛的妹妹，情急的 Melon 向人打聽後，得知

疑似是住在島嶼另一側地下洞窟的黑恐龍 Ken 擄走自己的妹妹，於是 Melon 一股腦的朝目標前進，忽略前途可能遭遇的障礙和困難，無計可施的 Melon 需要玩家的從旁協助，讓 Melon 突破重重關卡，順利的找到自己心愛的妹妹。

第二章 開發環境

本章將介紹開發遊戲的過程中所需使用的開發工具。如表 1

表 1 系統開發工具表

類別	軟體	用途
作業系統	Windows 10	開發遊戲之作業環境
遊戲開發環境	Microsoft Visual Studio 2017	使用 Windows Form 豐富的 UI 控制項開發遊戲。
程式語言	C#	遊戲開發
美工製作	Photoshop、Vectr.com	背景、角色繪製
資料儲存	記事本	遊戲紀錄

第三章 需求分析

本專案最重要的環節，在本章定義產品的走向和主要核心是什麼。

3.1 系統目標

參考既有 Scratch 遊戲專案模式，採用橫向卷軸方式呈現，以第三人稱視角的方式進行遊玩，遊戲系統需具備圖片繪製、呈現舞台等類別，將物件生成到舞台上，舞台放入主要視窗中呈現。其中，遊戲所需的功能性方法，依照特性各自撰寫類別封裝。

3.2 系統範圍

使用對象不限年齡層，以六歲以上皆可玩的方針來設計。簡易的點擊操作讓玩家能輕易上手。

3.3 系統整體描述

請參考章節四。

3.4 功能需求

3.4.1 玩家名稱

3.4.2 排行榜

3.4.3 音量設定

3.4.4 遊玩模式

3.4.5 故事介紹

3.5 非功能需求

3.5.1 回應時間與執行效能：建議須具備至少 2G 以上記憶體。

3.5.2 維護性：

- 具有能力的角色，繼承 `AnimateActor` 類別。
- 新增功能可透過獨立類別封裝後使用。
- 新增舞台類別，可以額外擴增關卡。

3.5.3 使用性：一般玩家約遊玩十次左右即可熟悉遊戲所有流程、操作。

3.5.4 可靠度：系統低概率於重複使用同樣物件導致衝突。

第四章 系統分析與設計

4.1 系統平台架構

本系統之平台架構。如圖 1

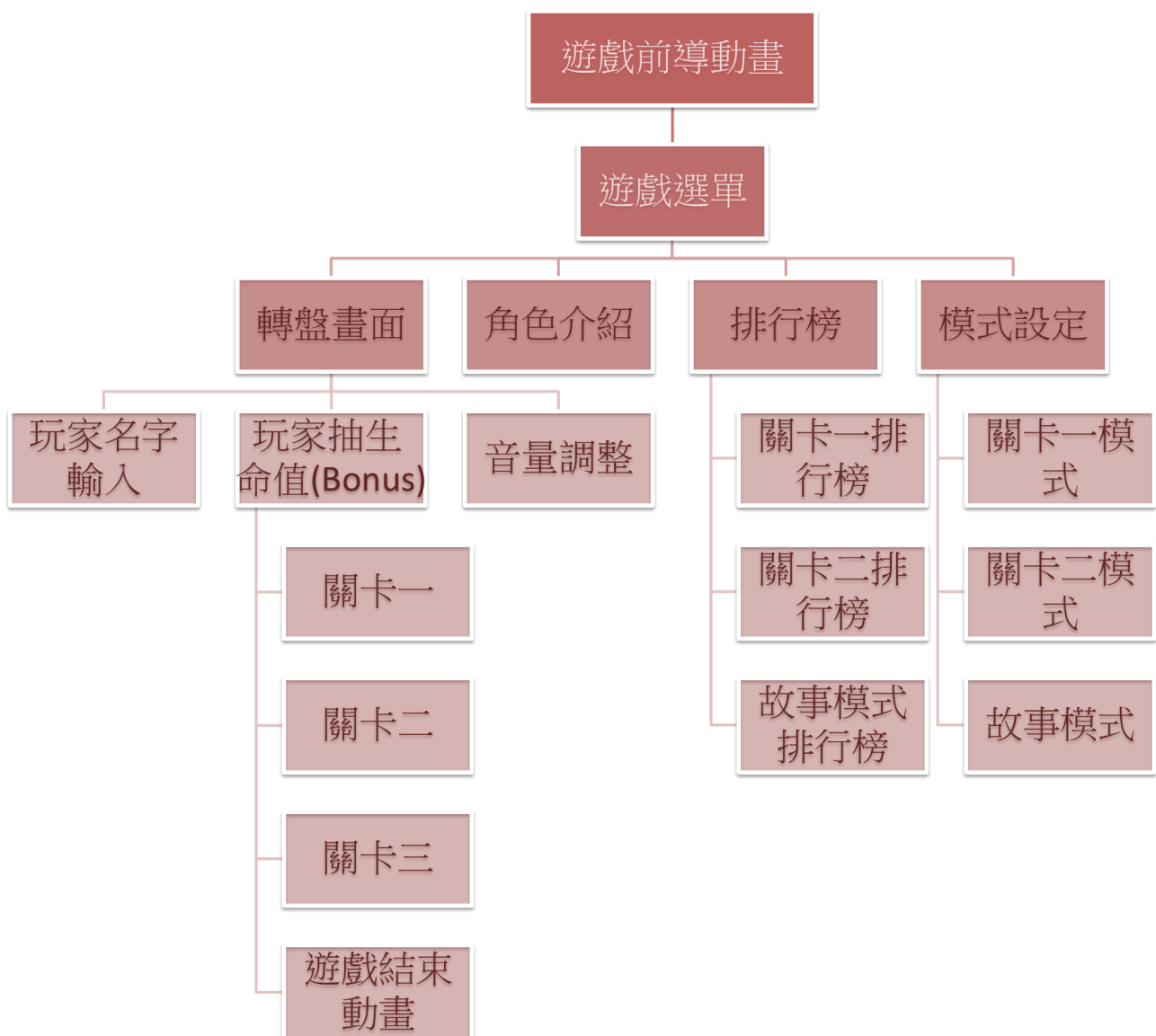


圖 1 系統平台架構

4.2 使用個案圖

玩家可操作或使用之功能如下圖 2。

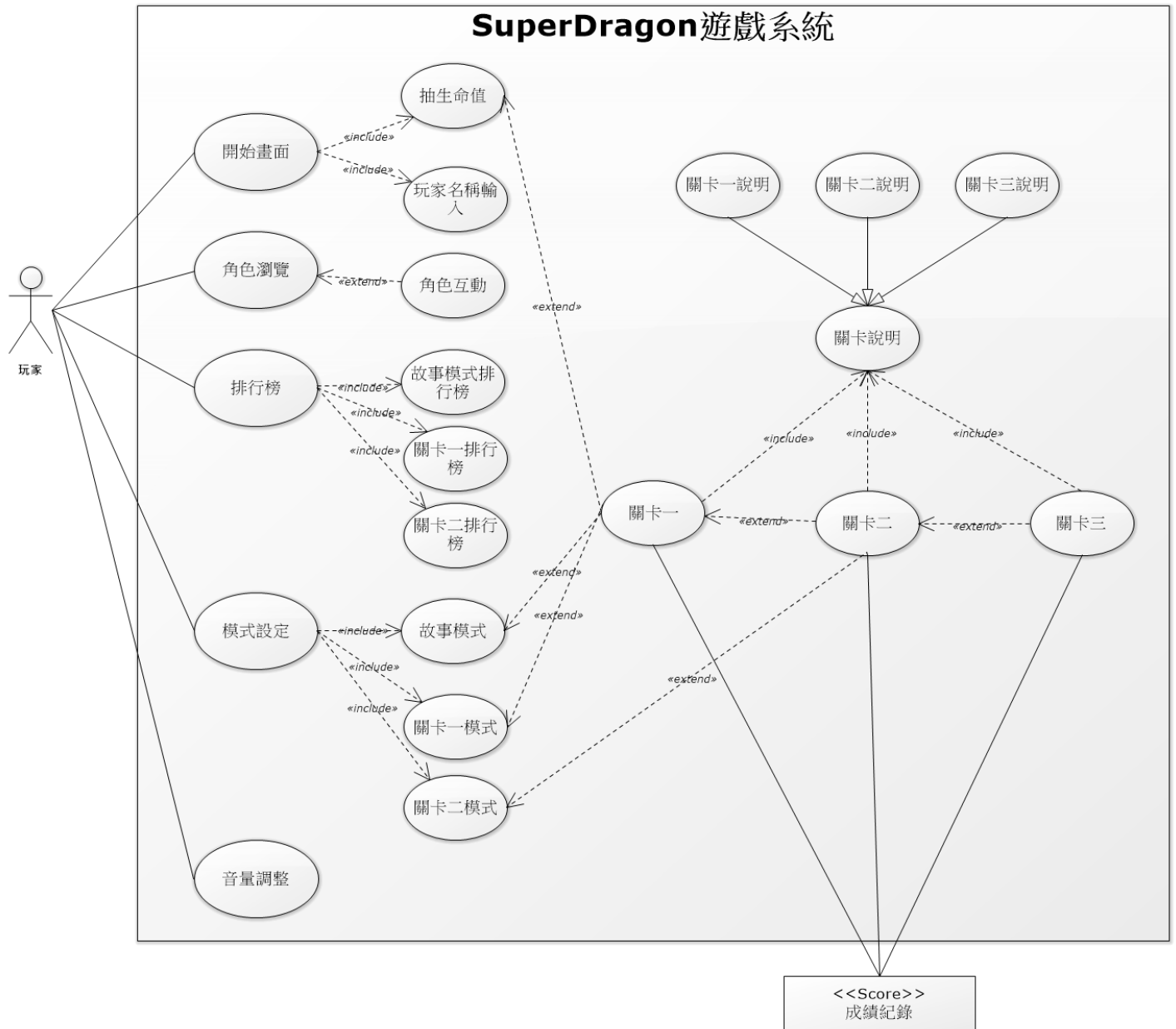
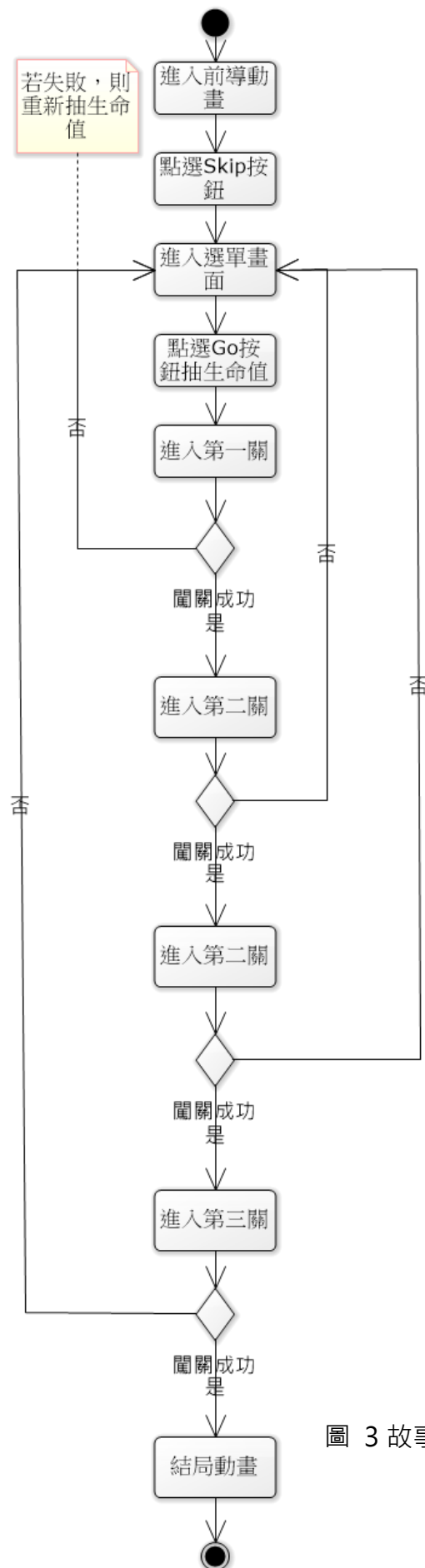


圖 2 使用個案圖

4.3 活動圖

4.3 - 1 故事模式活動圖



4.3 - 2 選單活動圖

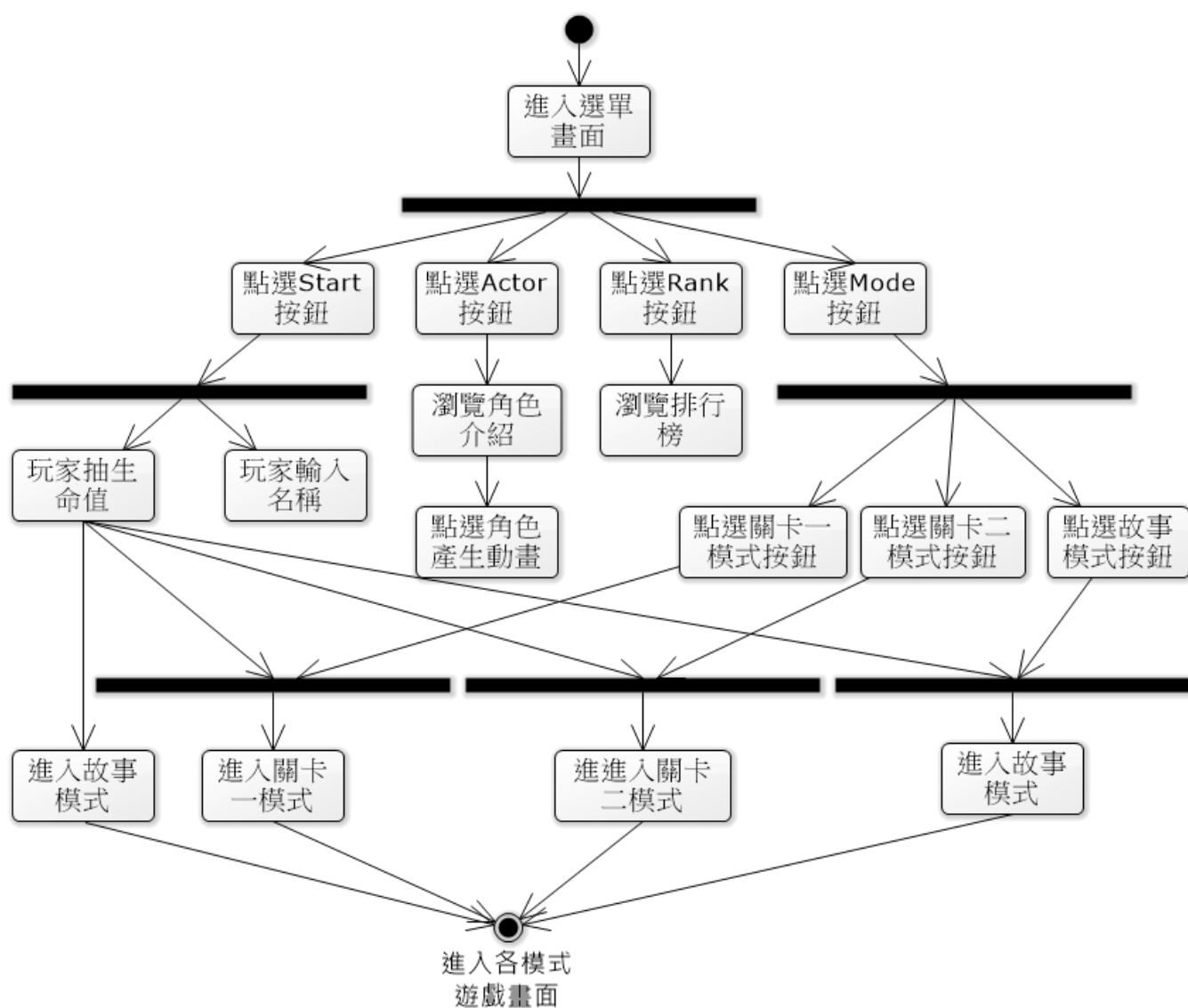


圖 4 選單活動圖

4.3 - 3 第一關活動圖

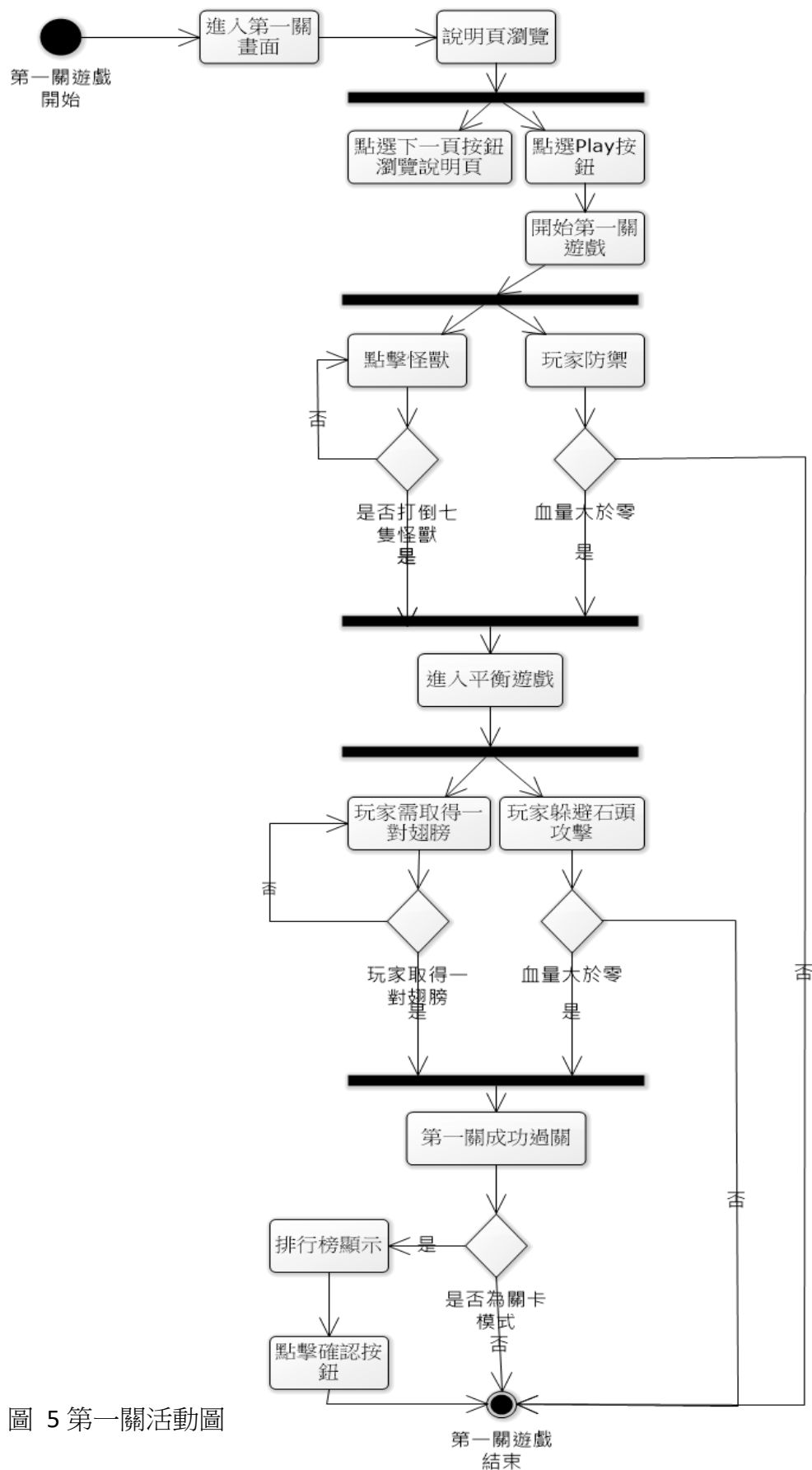


圖 5 第一關活動圖

4.3 - 4 第二關活動圖

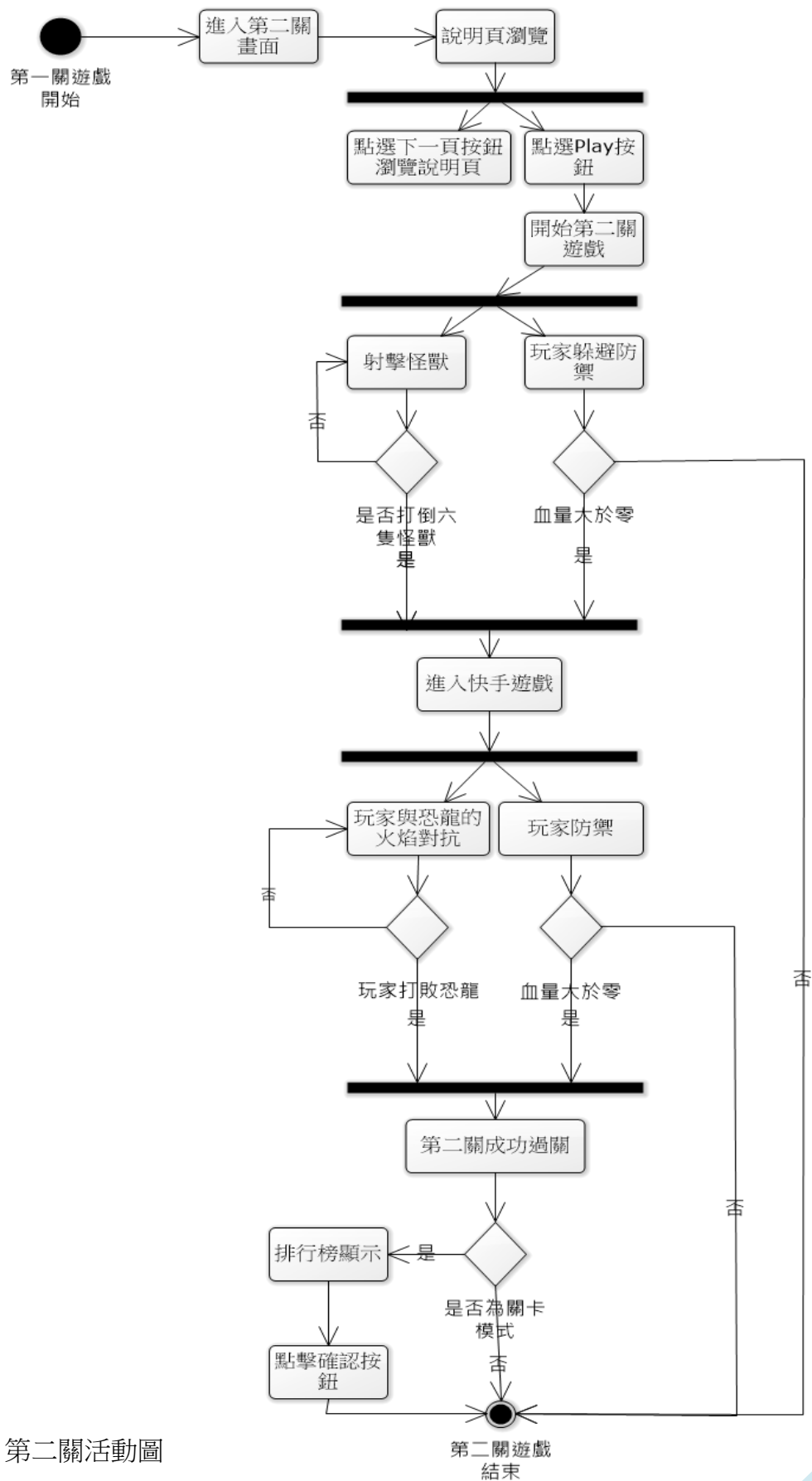


圖 6 第二關活動圖

4.3 - 5 第三關活動圖

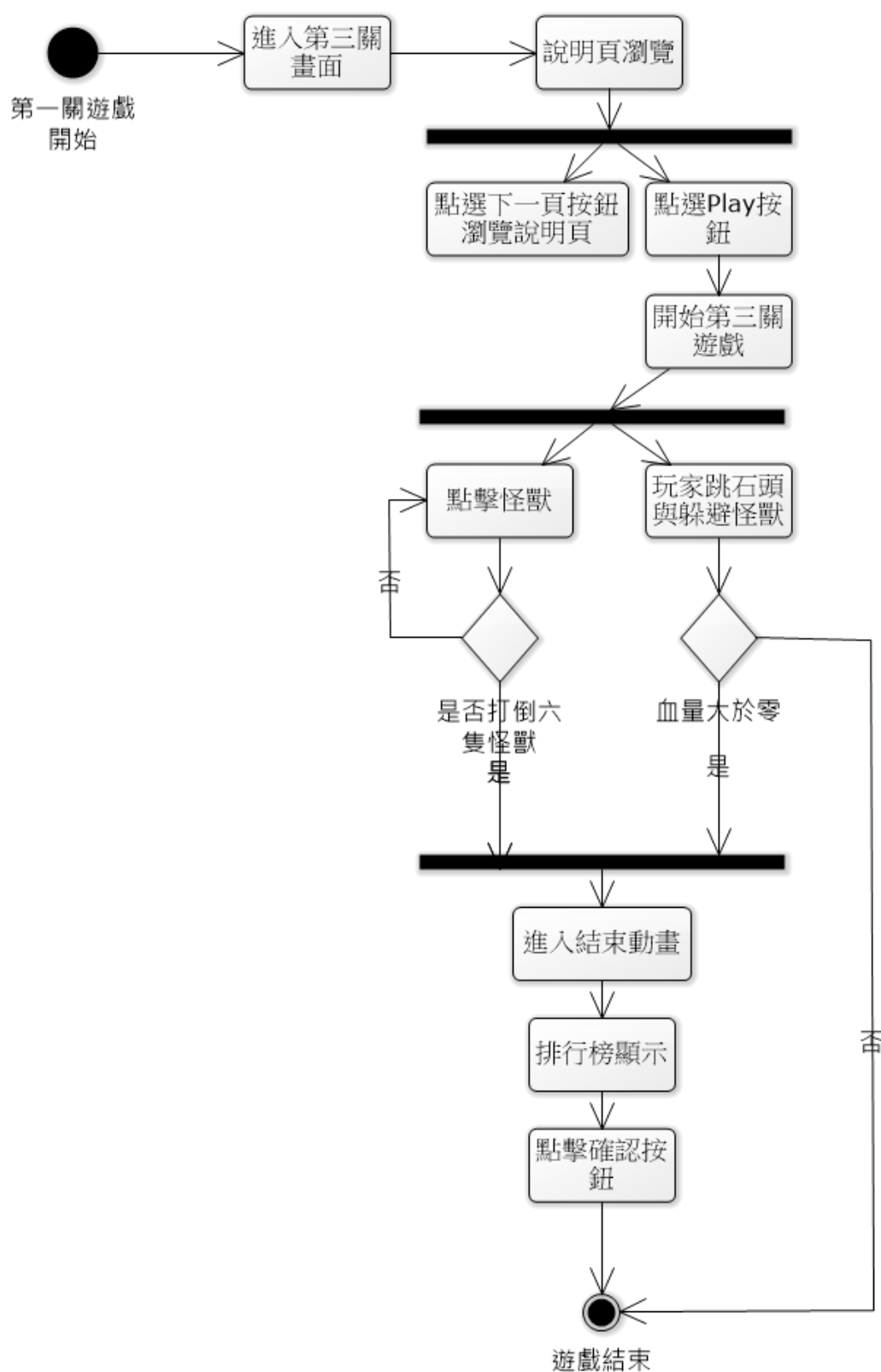


圖 7 第三關活動圖

4.4 類別圖

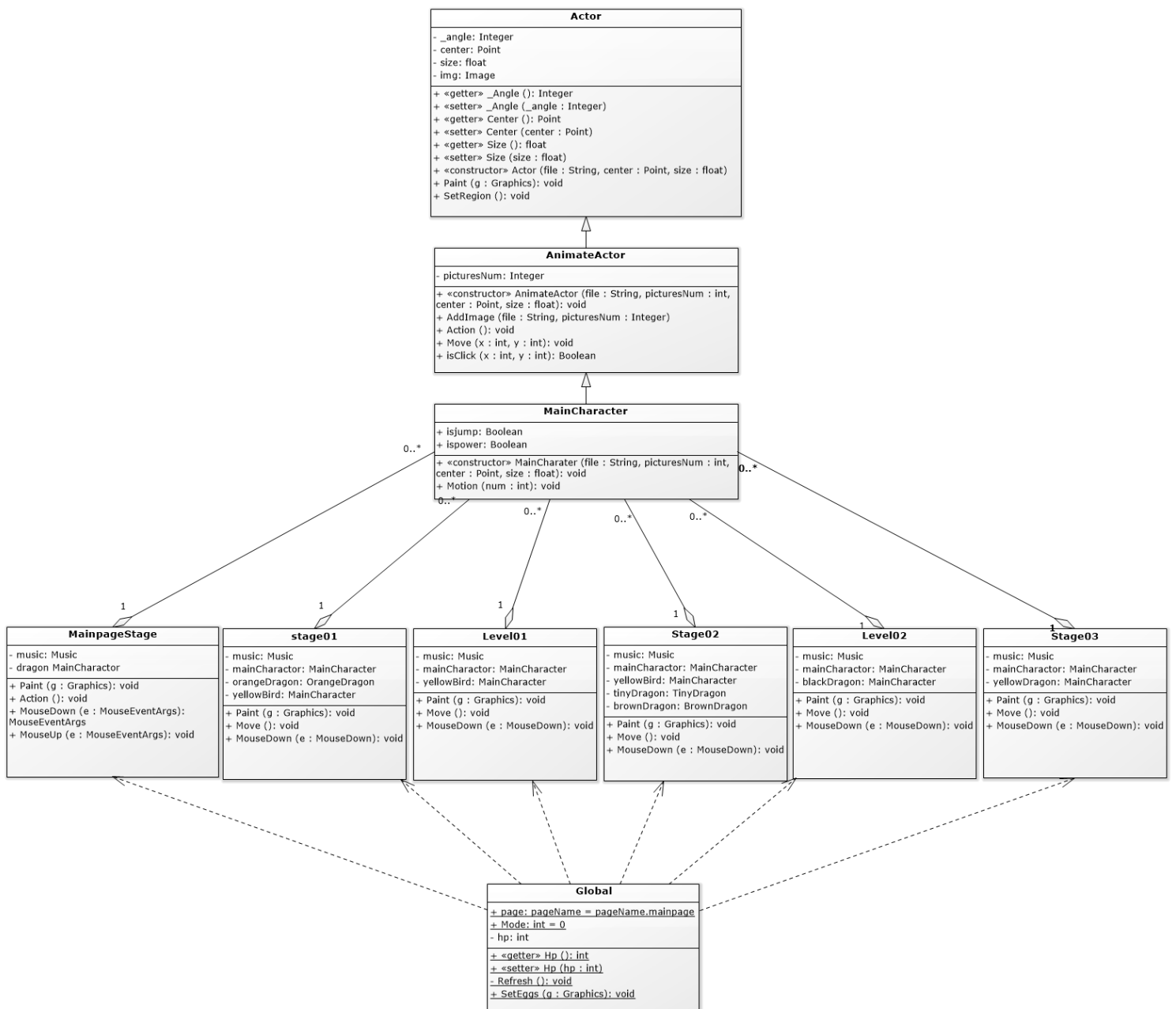


圖 8 系統類別圖

表 2 類別說明

模組名稱	說明
Actor	圖片的基本設定
AnimateActor	動畫的基本設定
HealthPt	血條變換設定
PinkDragon	恐龍屬性設定
TinyDragon	恐龍屬性設定
YellowBird	恐龍屬性設定
BrownDragon	恐龍屬性設定
BlackDragon	恐龍屬性設定
OrangeDragon	恐龍屬性設定
MainCharator	恐龍屬性設定
Button	按鈕屬性設定
Fire	火焰屬性設定
CountTlme	計算時間
Global	生命值設定
TurnTable	轉盤屬性設定
Music	音樂功能設定

Score	分數計算
TurnPage	轉場屬性設定
Foot	腳印動畫顯示
Ranker	排行榜存取
MainpageStage	主畫面顯示
TurntableStage	轉盤頁面顯示
Stage01	關卡一顯示
Level01	關卡一顯示
Stage02	關卡二顯示
Level02	關卡二顯示
Stage03	關卡三顯示
ResultForm	結束畫面顯示

第五章 團隊組成與分工

組員	執行項目
郭恩宏	遊戲核心演算、選單畫面製作、第一關製作、第三關製作、音效配置、排行榜、串關設定
鐘怡茜	遊戲架構、主畫面製作、選單畫面製作、第一關背景製作、第二關製作、說明頁製作

表 3 團隊組成與分工表

	郭恩宏	鐘怡茜
圖片	70%	30%
繪製	角色、場景、說明頁設計	主視覺風格、Logo、按鍵與角色設計
遊戲	20%	80%
內容	內容一致性、故事設定	遊玩方式設計、Scratch 專案原案
程式	60%	40%
撰寫	遊戲功能、關卡設計	主要架構、頁面配置、關卡設計
報告	50%	50%
撰寫	輔助報告撰寫、影片製作	專案文件、專案發想、簡報製作
比重分配	200%	200%

表 4 分工比重表

第六章 實作方法

6.1 Region 與中心點繪圖之結合

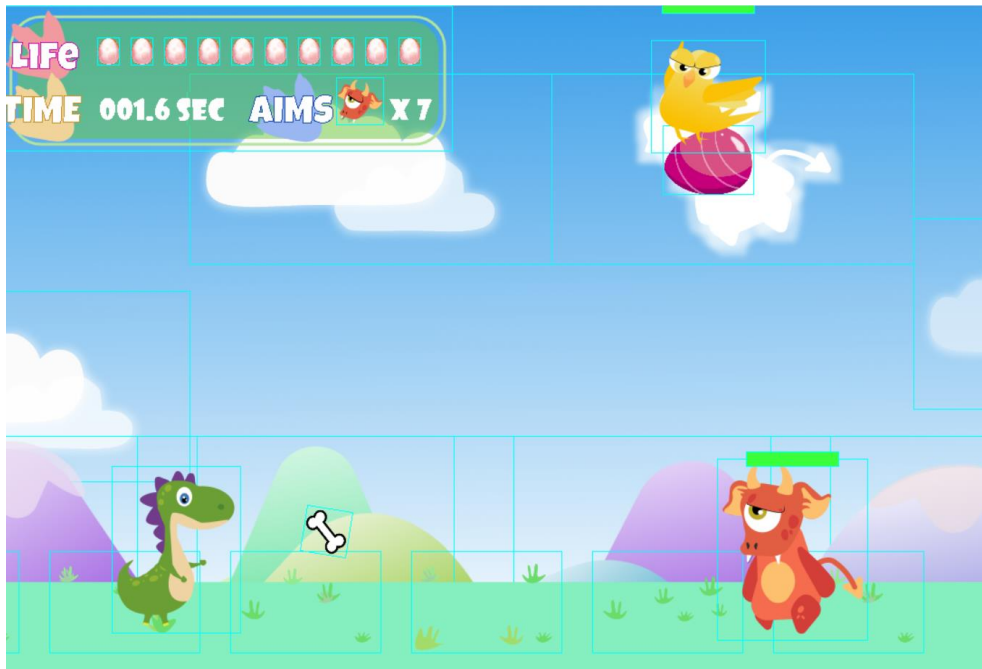
設計時考慮到可能有圓形碰撞、矩形碰撞等需求，這時如能直接取得圖片的中心點，將可省下不少功夫，所以設計了以中心點為基準，透過導入圖片的長、寬計算四個邊角位置(長寬還可依據變數 Size 等比縮放)。

透過 Region 的方式，用四個點構成圖畫的範圍，除了能輕易地進行旋轉碰撞判斷(在算出四個點時，同時放入了旋轉公式，請參考圖 9)，還能夠直接繪出角色的繪圖範圍(圖 10)，在程式撰寫過程相當方便。

圖 9 Actor 父類的 Region

```
//----- Region 設定 -----//
public void SetRegion() {
    // 旋轉座標位置
    // x = (x1 - x2)cos $\theta$  - (y1 - y2)sin $\theta$  + x2 (x1:原始座標點, x2:中心座標點)
    // y = (y1 - y2)cos $\theta$  + (x1 - x2)sin $\theta$  + y2
    sita = Convert.ToSingle(_angle * Math.PI / 180);
    // 點擊區域計算
    myGraphicsPath = new GraphicsPath();
    myRegion = new Region();
    myGraphicsPath.Reset();
    inputpoint1[0] = new Point(Convert.ToInt32(-img.Width * (gdiPercentage / size) / 2.0 * Math.Cos(sita) + ;
    inputpoint1[1] = new Point(Convert.ToInt32(-img.Width * (gdiPercentage / size) / 2.0 * Math.Cos(sita) - ;
    inputpoint1[2] = new Point(Convert.ToInt32(img.Width * (gdiPercentage / size) / 2.0 * Math.Cos(sita) - in
    inputpoint1[3] = new Point(Convert.ToInt32(img.Width * (gdiPercentage / size) / 2.0 * Math.Cos(sita) + in
    myGraphicsPath.AddPolygon(inputpoint1);
    myRegion.MakeEmpty();
    myRegion.Union(myGraphicsPath);
}
```

圖 10 製作者模式下能輕易看出繪圖範圍，以及物件的具體數量



6.2 舞台、角色、功能性物件

除了 WinForm 控件外，所有在舞台上出現的東西，皆源自於 Actor 這個父類別，純靜態背景可直接使用 Actor 類別，但在類橫向卷軸遊戲模式下，所有東西都會移動，因此繼承 Actor，設置 AnimateActor 類別，在此類別中實作了新增、修改、移除、輪播(正反序、翻轉)圖片等功能。

舞台即不同場景、關卡，因為關卡間玩法差異大，所以我們選擇讓登場的所有角色與建構子一起產生，並在 Form1 Class 產生新的靜態舞台，就能輕鬆切換場景，並把不要的場景直接 null 釋出空間，間接實作場景歸零、暫停的效果。

Globe 類別為全域的 Class，功能類同常見的 GameControle Class，用來控制遊戲流程，以靜態變數 page 做為控制每個關卡的開始與結束，並由主角

當前血量決定遊戲是否繼續進行。關卡模式與故事模式的判定，以靜態變數

Mode 作為模式切換的基準。

6.3 轉盤設計



圖 11 轉盤實作成果(註: 上圖輪盤是賺取隨機的額外生命值 5 + (5))

實作轉盤列幾項須具備的要素：隨機性、正確讀取數值。前者最簡單的方式可以透過 Random 生成，再依據該值生成對應圖片位置，但本例作法並非如此。本案轉盤結合計時器(一個撰寫出來的 CountTime 類別，在計時開始後會回傳 false 直至所設定的時間到達瞬間回傳 true)，透過亂數指定不同時間與轉速的組合，可以達成指定輪盤的特定區間，也因為是區間，所以可依電腦執行速度不同，每次結果都還會有些微變化，甚至加入了轉到交界處會自動重轉的功能。其次正確讀取轉盤數值的方法是依據角度來作判定，以%360 取餘數的方式得到當前轉盤位置，並依據區間不同進入不同 case，回傳不同生命值參數。

6.4 火焰對抗設計

火焰對抗是一個簡單的方法所做出，最初構想是採計滑鼠一定期間位移量的方式來進行火焰輸出判定，不過這種方法在滑鼠操作上很容易迴避難易度，導致關卡本身難易度不好調整。

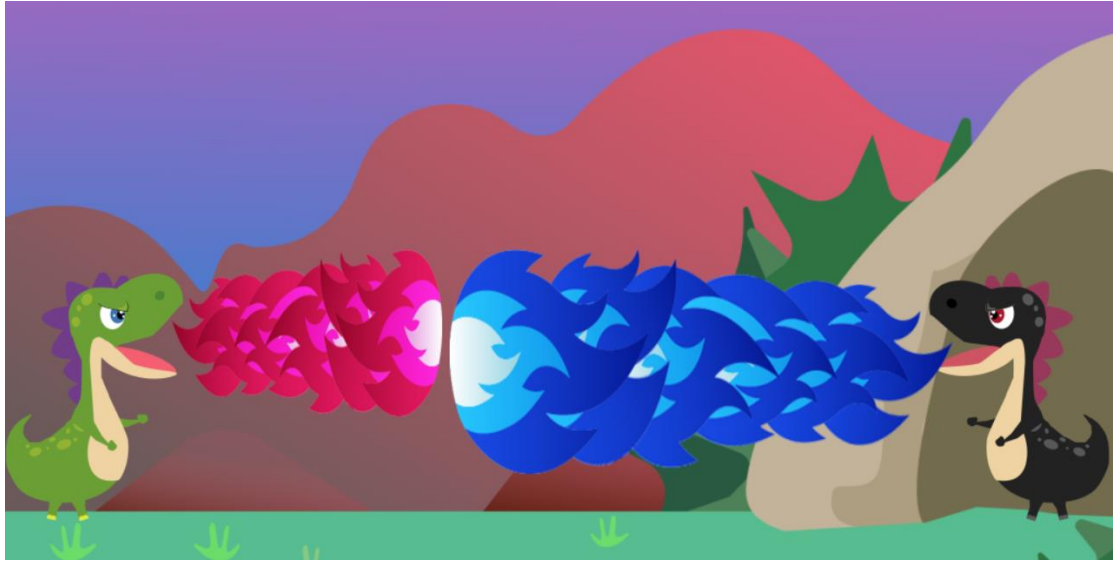
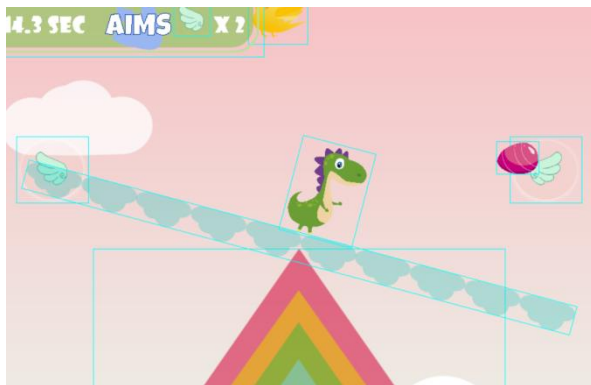


圖 12 火焰對抗實作成果

改變的方法是判斷水平方向移動的折返次數，當折返頻率越高，越能快速輸出高強火力，但不同輸入設備，如：觸控板，可更快疊加快火力輸出。所以本關在難度設計上並沒太難，會讓玩家想要靠自己來通關。

6.5 平衡木設計



如果詢問最花費時間設計的關卡，非本關莫屬，因為它需要底層架構的支持！也算是促成專案改用 Region 作為繪圖範圍的主因。

Region 本身具備 IsVisible 的

方法，可判斷給予的點是否落在範圍內，亦可做多邊形的設定，未來擴充修改也

圖 13 平衡遊戲
較為彈性。

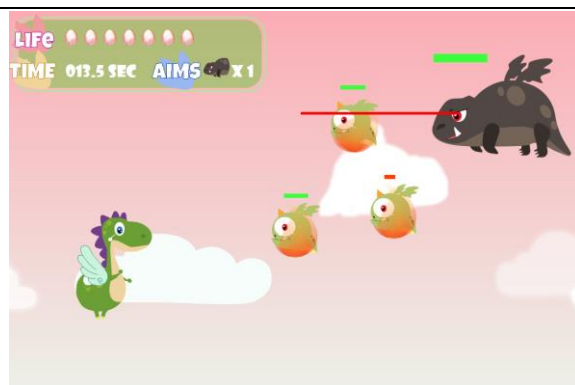
本案雖採四邊形為主，但主要訴求除了判定落點位置，還有進行旋轉的設定，一般透過轉置矩陣進行圖片旋轉，但 Rectangle 本身的座標位置並不會因為旋轉矩陣 Matrix 產生變化，所以碰撞判定與實際繪圖是不符的。改用 Region 除了簡化碰撞判定程式碼的撰寫，亦可自行將中心旋轉公式帶入底層，讓所有角色繪圖與實際框線是一致的。

第七章 專案成果

頁面名稱	圖片呈現
前導動畫主畫面	
轉盤頁面	
角色瀏覽介紹	
排行榜	

<p>模式設定頁面</p>	
<p>遊戲說明</p>	
<p>第一關 – 反應遊戲</p>	
<p>第一關 – 平衡遊戲</p>	

第二關 – 射擊遊戲



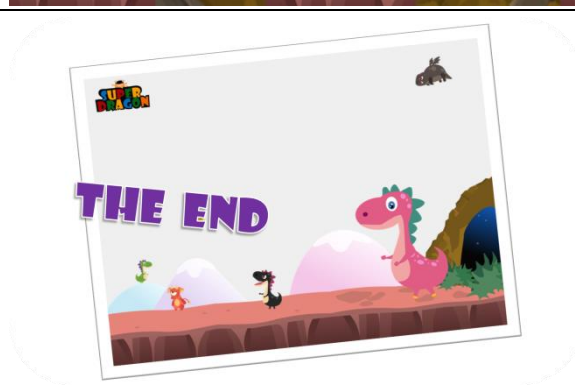
第二關 – 快手遊戲



第三關 – 反應遊戲



結束畫面



第八章 討論與結論

由於成立較晚，專題規劃上較匆促定案，致使後續實作上花費不少時間修改、磨合。所幸兩人合作默契越來越好，不論從繪畫到程式等，各個部分都能相輔相成、互相監督，審慎評估並改善另一人所執行的成果，讓專題成品愈加完善。

遊戲由成員怡茜設計的恐龍為主角，並依照最初設計的 SuperDragon Logo 為美術概念圖，由組員恩宏負責其他角色設計、背景設計，組員怡茜則以小插圖、按鈕元件設計為主，中間不時互相交換工作成果，以確保風格能維持一至。

角色設計部分，由於並非設計專業，所以先參考網路資源，以描繪骨架的方式，進行二次創作，其目的是為了製作角色動畫，另一部分則是為了保持風格的一致性，再次特別感謝 C#班曹凱嵐同學分享 Vectr.com 這樣簡單易上手的繪圖網站，大量縮短摸索專業繪圖軟體的時間，從而產出大量的角色動作與背景圖畫。

程式設計部分，架構的設計者為組員怡茜，實作遊戲邏輯則以組員恩宏為主，但礙於前置作業時間較短與初期溝通並不完善，所以在功能與架構結合上花費不少時間磨合。雖然最後成品並非物件導向，但在擴增關卡與實作上算有相當大的彈性，符合最初專案的核心需求「多樣化的闖關方法」。

怡茜以中心繪圖為主軸的架構相當好用，讓整個遊戲在移動與判斷上變得相當便利，再藉由 Region 取代 Rectengle 的方式，可以更輕鬆地設置旋轉方法，讓所有對中心旋轉的判定變得容易。由恩宏實作的火焰對抗(滑鼠來回移動判定)、翹翹板斜面上滑動、隨機轉盤皆相當有意思。

改善空間大致上是彼此雖然信賴對方的成果，但卻因為信賴而較少溝通已實作的部分，導致有時會有重複對方工作的部分。在遊戲專案最初討論上應再多投資時間，讓整體方向更加明確，避免所實作出來的東西還需要做多次修正。整體而言彼此都對這次合作相當滿意，成果算是超乎最初預期的程度。