

電腦遊戲設計  
分組作業報告說明書

神奇寶貝 VS 瑪利兄弟

第二組

資四 B	96156203	蔡鈺蘋
資四 B	96156208	吳宸緯
資四 B	96156218	劉爾剛
資四 B	96156253	蕭智元
資三 A	97156156	陳芳瑜

## 目 錄

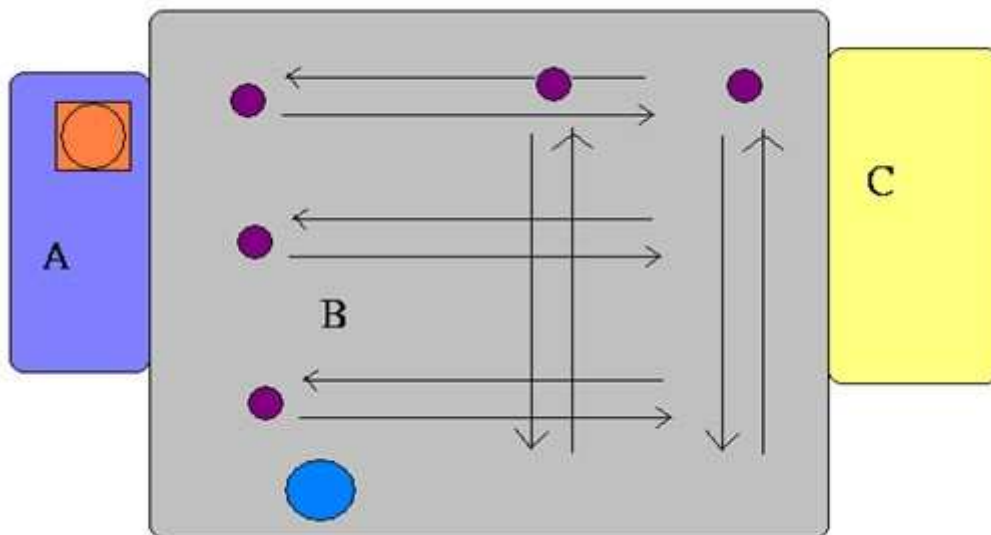
1	遊戲設計動機、目的.....	3
2	遊戲設計雛型.....	3
2.1	介面雛型.....	3
2.2	遊戲需求彙整.....	4
3	遊戲背景介紹.....	4
4	遊戲說明.....	4
4.1	操作說明.....	4
4.2	界面說明.....	5
4.2.1	遊戲主選單.....	5
4.2.2	遊戲角色、物品介紹.....	6
4.2.3	遊戲畫面.....	6
5	開發架構.....	8
5.1	架構.....	8
5.2	遊戲 XNA 技術應用範圍.....	8
6	遊戲設計心得.....	9
6.1	96156203.....	9
6.2	96156208.....	9
6.3	96156218.....	10
6.4	96156253.....	10
6.5	97156156.....	10
7	工作分配.....	11
8	程式碼.....	11
9	參考資料.....	13

## 1 遊戲設計動機、目的

藉著遊戲設計課程所學到的各個程式設計方法。結合設計出遊戲。利用簡單的遊戲方式與精簡的遊戲畫面，訓練靈活腦力與反應力活動。

## 2 遊戲設計雛型

### 2.1 介面雛型



- 橘色的圖形為可操作可移動的圖形(主角)。
- 紫色的圓球為關卡中的障礙物隨機產生，在畫面中移動
- 而遊戲者必須從起點 A 區操控移動的圖形
- 避過隨機產生而移動的圖形和去吃取(碰撞)另一樣隨機產生而移動的圖形

## 2.2 遊戲需求彙整

- 提供主選單畫面讓使用者可選擇進入遊戲、遊戲說明、選擇遊戲關卡(難度)
- 憑藉 XNA 所提供的函數，做出滑鼠游標偵測輸入、鍵盤按鍵輸入來操作遊戲
- 建立遊戲的操作畫面、背景
- 依照討論出的遊戲呈現方式和遊戲規則。設計出遊戲程式邏輯

## 3 遊戲背景介紹

大多數人一輩子都沒有機會操作飛機，更不可能體驗到飛行時突破、穿梭在各個飛彈中高速奔跑、滾動、跳躍。為了追求、獲取最大利益而得到的快感。

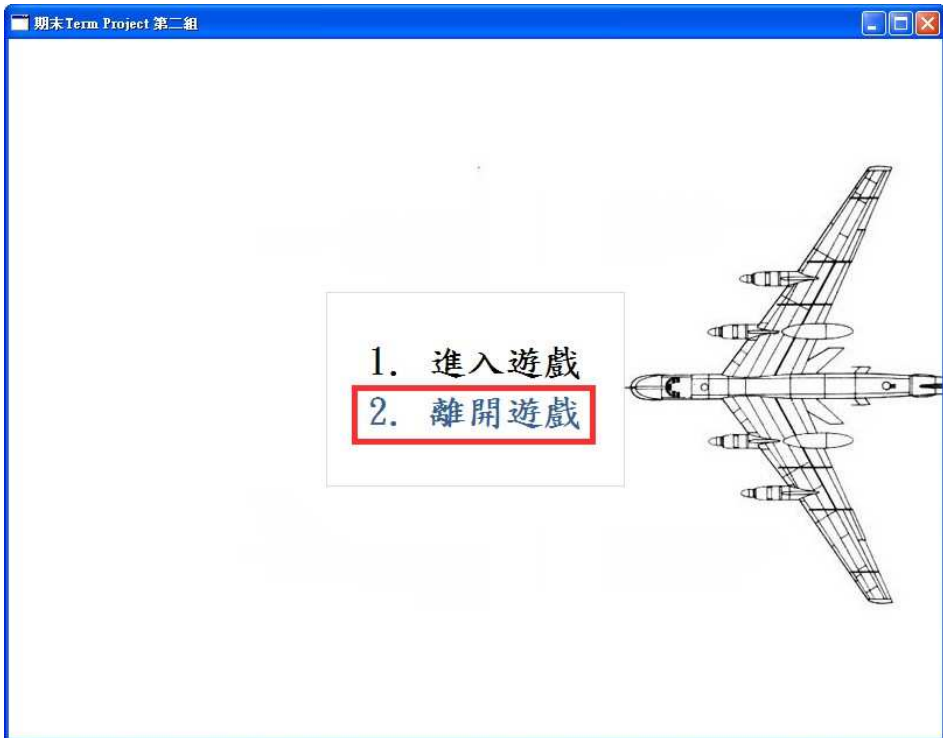
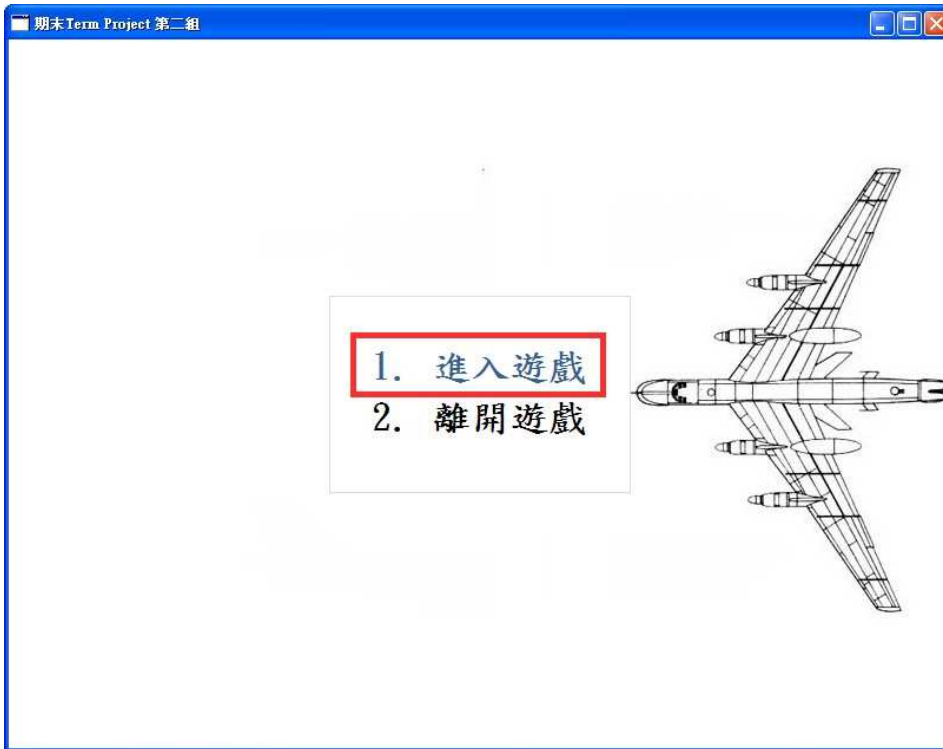
## 4 遊戲說明

### 4.1 操作說明

駕駛飛行物(主角)能上下左右移動，有具體方位、角度，必須一直地躲避導彈。而且要積極地一直補充燃料(蛋糕)。飛行物必須閃避隕石(炸彈)後。收集蛋糕。還會順便補充能量。關卡中補充到足夠的燃料(蛋糕)，到達一定的數量時就可以進入到下一關，戰鬥時所累積的分數可以用來幫飛行物(主角)升級。

## 4.2 界面說明

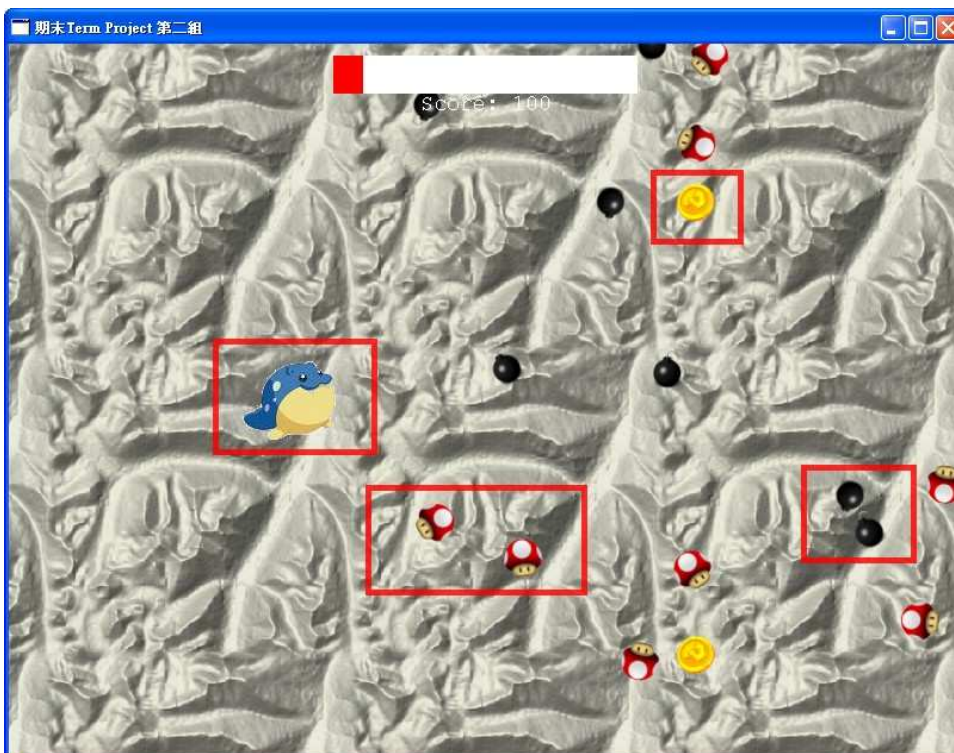
### 4.2.1 遊戲主選單



### 4.2.2 遊戲角色、物品介紹

	主角(玩家操作)
	補充燃料(香菇)
	隕石(炸彈)
	密技飛彈
	機會、(命運)。大量補充燃料、(結束遊戲)

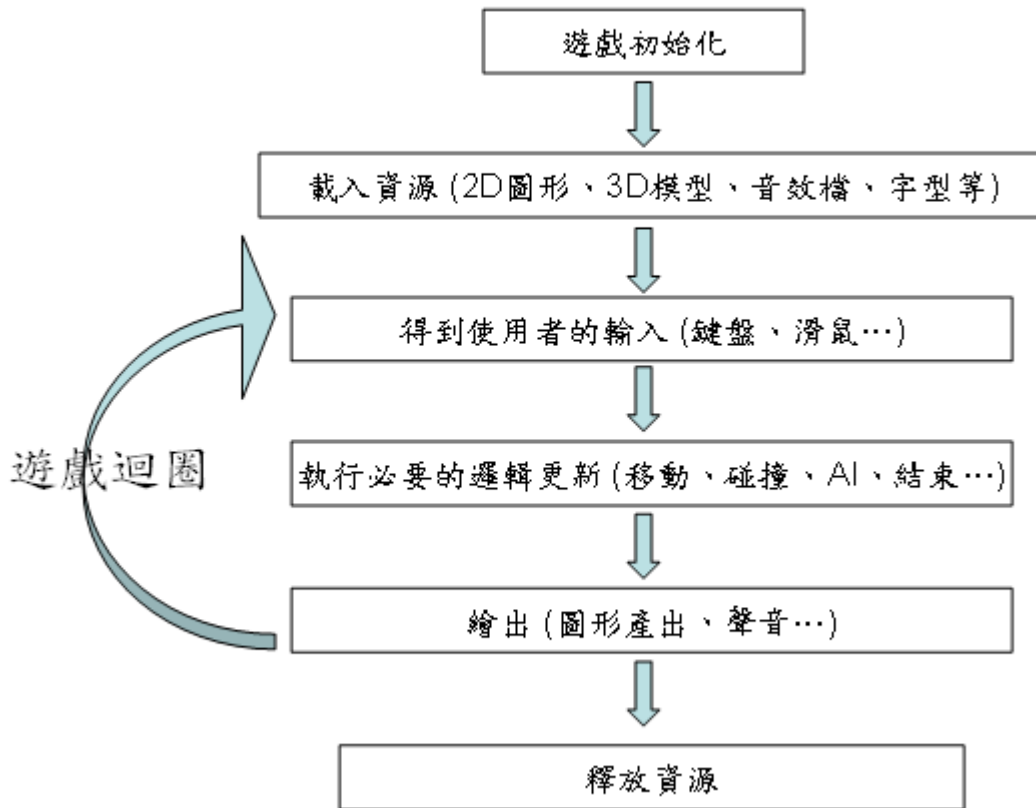
### 4.2.3 遊戲畫面





## 5 開發架構

### 5.1 架構



參考來源: 2D/3D 遊戲程式設計入門：使用 XNA 3.0 與 C#，鄧永傳、何振揚編著

首先在程式撰寫時。對遊戲架構要有更清晰了解。程式在執行時，它生命週期各階段會有相互影響。這是在撰寫程式中會特別去留意。

### 5.2 遊戲 XNA 技術應用範圍

此遊戲設計運用在這學期課堂上所學到的：“無接縫貼圖”、“互相碰撞 2D 圖形”、“行走的小王子”、“遊戲架構流程”、“外部設備輸入(滑鼠、鍵



盤)”等...。創造出簡單、基本能操作的小遊戲。



## 6 遊戲設計心得

### 6.1 96156203

在大二時修習過計算機圖學。現在的遊戲設計也是有跟圖學有更大的密切關係。使用大量的離散數學和線性代數，凸顯出資訊基礎課程的重要性。數學的部份讓我知道如何視角、座標位置，在程式中呈現出來。而程式也是不斷的呼叫函式庫裡的函式。不斷地利用一堆圖，不斷的重新繪圖、更新，重新繪圖、更新。

### 6.2 96156208

我們試著藉著課堂老師所教的課本內容。提供的範例拿來研讀，如此而已。不過，離現今市面上提供的遊戲還有好大的差距！我的夢想是未來的某一天能在知名社交平台上寫個小遊戲讓大家玩。

### 6.3 96156218

XNA 在日前已經公佈到 4.0 版本。而他的特色就是簡單、好用、好學。還有一大關鍵就是不用任何花費。而且是使用 .Net 平台的 C# 程式語言。在 .Net、Java 為當今主流的時代。在資訊科系一定都會有接觸這兩者其一的的經驗。所以只要會 C#，在 .net Framework 架構下，就不用特地再去學其他的 SDK，.Net 架構下用 Framework SDK 寫的 API 都可以通用。XNA 開發後可在 XBOX 上操作。

### 6.4 96156253

在課堂上講解遊戲概念、文件製作、文件內容的，讓我了解遊戲製作的過程以及人員安排等等。由於我專題是在業界實習。會認為上課的這概念不只單單只能放在遊戲上，在以後實際參與專案。開發產品或計劃，都可以用得上。

### 6.5 97156156

在老師的課程中，雖然我沒有 JAVA 的基礎，但是，在第三節課進行上機練習的時候，有老師給的範例，再藉由更改部分程式碼來一步一步發展，最後漸漸的上手了，去實際操作比較容易懂著程式的運行，就像以前在學 C++ 一樣，都是到了實習教室才開竅，這次的環境雖然讓我比較陌生，以前也沒修過視窗 VB，讓我起頭有點困擾，不過在熟悉之後，就有一種成就感，感覺自己比別人多學了一點。

## 7 工作分配

	遊戲邏輯	介面配置	關卡設計	遊戲流程架構
蔡鈺蘋		√	√	
吳宸緯	√		√	√
劉爾剛	√		√	
蕭智元	√	√		
陳芳瑜		√	√	

## 8 程式碼

```

using System;
using System.Collections.Generic;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Storage;
using Microsoft.Xna.Framework.Content;

namespace WindowsGame1
{
    /// <summary>
    /// This is a game component that implements IUpdateable.
    /// </summary>
    public class GameComponent_Plane : Microsoft.Xna.Framework.DrawableGameComponent
    {
        GraphicsDevice device;
        Game game;
        public SpriteFont font;
        // private SpriteBatch spriteBatch;

        Class_BG bg;
        Class_Plane plane;
        Class_Bar bar;

        public GameComponent_Plane(Game game)
            : base(game)
        {
            // TODO: Construct any child components here
            this.game = game;
            device = game.GraphicsDevice;
        }
    }
}

```

```

/// <summary>
/// Allows the game component to perform any initialization it needs to before starting
/// to run. This is where it can query for any required services and load content.
/// </summary>
public override void Initialize()
{
    // TODO: Add your initialization code here
    bg = new Class_BG(game, game.Content.Load<Texture2D>("Background2"));
    plane = new Class_Plane(game, game.Content.Load<Texture2D>("abc"),
        game.Content.Load<Texture2D>("bomb"),
        game.Content.Load<Texture2D>("ddl"),
        game.Content.Load<Texture2D>("whh"),
        game.Content.Load<Texture2D>("whh"),
        game.Content.Load<Texture2D>("bbl"));
    font = game.Content.Load<SpriteFont>("Courier New");
    bar = new Class_Bar(game, game.Content.Load<SpriteFont>("Courier New"));
    bar.Initialize(256, 32); // 產生 bar 紋理圖
    bar.Pos = new Vector2(device.Viewport.Width / 2 - bar.Width / 2, 10);

    base.Initialize();
}

/// <summary>
/// Allows the game component to update itself.
/// </summary>
/// <param name="gameTime">Provides a snapshot of timing values.</param>
public override void Update(GameTime gameTime)
{
    if (GameStateClass.currentGameState != GameStateClass.GameState.Game2)
        return;

    KeyboardState newState; // 宣告一個KeyboardState 結構的變數
    newState = Keyboard.GetState(); // 得到目前鍵盤每一個按鍵的狀況
    if (newState.IsKeyDown(Keys.P)) // Escape -> 轉回 主選單
        GameStateClass.changeState(GameStateClass.GameState.Menu, game);

    bg.Update(gameTime);
    plane.Update(gameTime);
    if (plane.planeHit) bar.score += 50;
    if (plane.bombHit) bar.score -= 30;
    // if (plane.bulletHit) bar.score += 5;
    if (plane.wh_SHit) bar.score += 100;
    if (plane.wh_FHit) bar.score -= 50;
    plane.rockProb = (bar.score+50) / 8000;
    plane.cakeProb = (bar.score+50) / 8000;

    if (bar.score == 1000)
    {
    }
    else if (bar.score == 0)
    {
    }

    base.Update(gameTime);
}

```

```
public override void Draw(GameTime gameTime)
{
    if (GameStateClass.currentGameState != GameStateClass.GameState.Game2)
        return;

    bg.Draw(gameTime);
    plane.Draw(gameTime);
    bar.Draw(gameTime);

    /*string message = "1231fwefihwoj";
    Vector2 FontOrigin = font.MeasureString(message) / 2;
    if (bar.score == 0)
        // spriteBatch.DrawString(font, "Game Over!", new Vector2(device.Viewport.Width / 2, device.Viewport.Height / 2),
        // spriteBatch.DrawString(font, message, new Vector2(device.Viewport.Width / 2, device.Viewport.Height / 2),
        // spriteBatch.DrawString(font, "Game Over!", new Vector2(100, 100), Color.Black);

    */

    if (bar.score == 0) //Escape -> 轉回 主選單
    {
        GameStateClass.changeState(GameStateClass.GameState.over, game);
        // GameStateClass.changeState(GameStateClass.GameState.Menu, game);
        // bar.score = 100;
        Initialize();
    }
    else if ( bar.score == 1000) //Escape -> 轉回 主選單
    {
        GameStateClass.changeState(GameStateClass.GameState.Help, game);
        Initialize();
        //bar.score = 100;
    }
    base.Draw(gameTime);
}
}
```

## 9 參考資料

- XNA Developer Center – <http://msdn.microsoft.com/zh-tw/xna/default>
- XNA 官方網站 – <http://creators.xna.com/>
- XNA PC/XBox 360 C#遊戲程式設計，葉思義、李震宇著，碁峰資訊，2007 年 11 月初版
- 2D/3D 遊戲程式設計入門：使用 XNA 3.0 與 C#，鄞永傳、何振揚編著，文魁資訊，2009 初版二刷