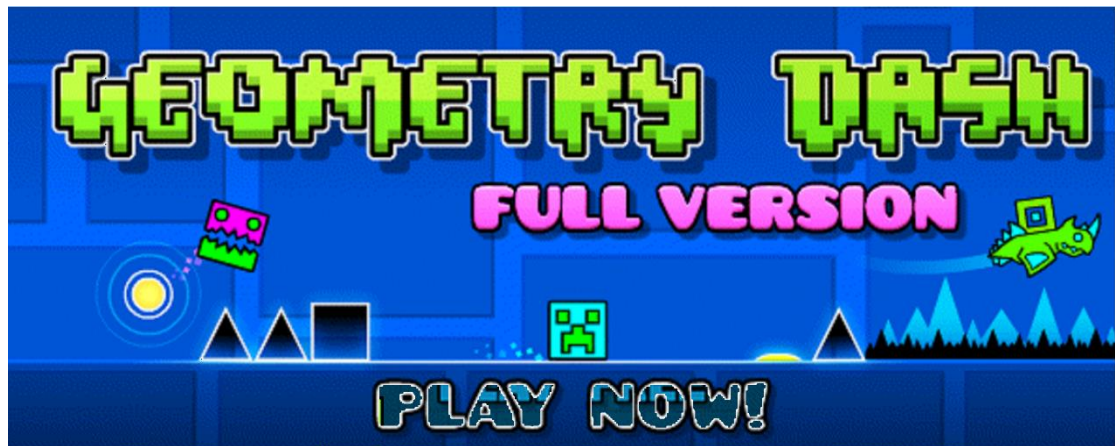


國立臺北科技大學
2020 Spring 資工系物件導向程式實習
期末報告

Geometry Dash 幾何衝刺



第 29 組

108590011 廉博旭

108590450 羅傑

目錄

一、簡介	1
(一)動機.....	1
(二)分工.....	1
二、遊戲介紹	
(一)遊戲說明.....	2
(二)遊戲圖形.....	2
(三)遊戲音效.....	3
三、程式設計.....	4
(一)程式架構.....	4
(二)程式類別.....	5
(三)程式技術.....	6
四、結語	8
(一)問題及解決方法.....	8
(二)時間表.....	9
(三)貢獻比例.....	10
(四)自我檢核表.....	10
(五)收獲.....	11
(六)心得、感想.....	11
(七)對於本課程的建議.....	12

一、簡介

(一)動機

其實也沒什麼太大的動機，在之前剛獲得手機的時候，下載了一款有趣的遊戲，也就是我們這次的主題「幾何衝刺」，玩了一下子就上癮了，那時候沉迷了不少光陰在裏頭，我希望能將這個遊戲介紹給大家。

(二)分工

108590011 廉博旭：

主要致力於遊戲介面方面，也會參與程式碼Debug。

108590450 羅傑：

主要致力於遊戲程式碼開發與Debug。



遊戲進行畫面：



(三)遊戲音效

遊戲音效直接採用原遊戲內對應的選單及關卡音樂。

三、程式設計

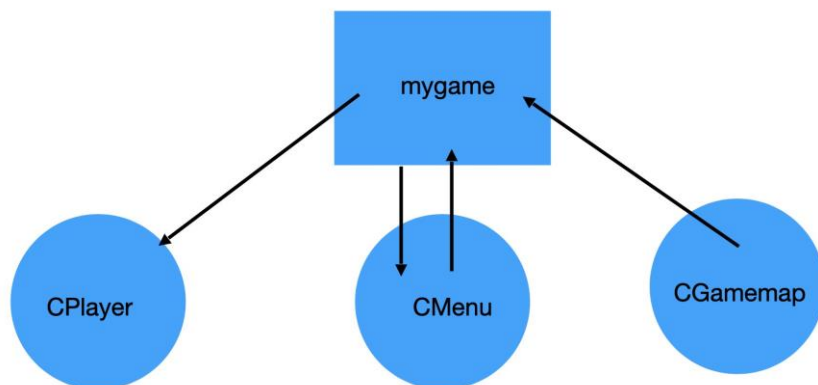
(一)程式架構

CGamemap：該類別實現了地圖的像素與障礙物轉換為陣列，提供mygame偵測障礙物的功能實現與判斷過關與死亡的功能實現。此類別沒有繼承任何類別。

CMenu: 該類別實現了選關與切換關卡，並繼承於CAAnimation，加載了6個LEVEL的bmp與首頁bmp。並且改寫了SwitchImage，使用於mygame中並且改寫了gamelib.cpp中的void CAAnimation::SwitchImage()，使用iterator。

CPlayer: 該類別實現了主遊玩角色方塊的功能，並繼承於CAAnimation，同樣也改寫了gamelib.cpp中的void CAAnimation::SwitchImage()，使用iterator。加載了3個變換跳躍變換的bmp，使用於mygame中。

mygame：該類別實現了所有遊戲運行的統合，包括CGameStateInit CGameStateOver 和CGameStateRun，加載了遊戲需要使用的音樂檔，實現了外掛模式的關卡跳躍(shift)、選關按鈕偵測(left鍵)，與判斷角色死亡的規則、設定回到關卡的實現、音樂關卡的切換功能實現、加載CGamemap的元素資料用於判斷角色的位置、調整關卡進行的角色移動速度與動作、最後是角色死亡後的倒數計時。



(二)程式類別

類別名稱	.h 檔行數	.cpp 檔行數	說明
Mygame	50		音樂檔 enum
Mygame	70	56	關卡切換與外掛 模式
Mygame		149	遊戲結束設定
Mygame		181	角色初始值設定
Mygame		207	音樂檔案加載
Mygame		233	死亡線設定
Mygame		246	選擇結束音樂
Mygame		263	角色移動
Mygame		278	障礙物與地板偵 測
Mygame		314	角色座標移動紀 錄
Mygame		356	角色跳躍偵測
CPlayer	10	35	初始化設定
CPlayer	11	47	Bitmap 加載
CPlayer	12	57	角色跳躍程式
CPlayer	15	74	角色座標值紀錄

CPlayer	14	85	角色切換跳躍圖片
CMenu	7	20	菜單初始化設定
CMenu	9	34	菜單切換圖片設定
CMenu	10	29	菜單顯示
CGameMap		15	遊戲關卡背景加載
CGameMap		23	遊戲關卡背景顯示
CGameMap		40	遊戲關卡座標設定
CGameMap	All		遊戲所有相關陣列值紀錄

(三)程式技術

CGamemap：該類別實現了地圖的像素與障礙物轉換為資料結構二維陣列，並且對應角色的座標與地圖座標來決定是否碰到障礙物並作出反應（死亡等、通關等）使用迴圈。障礙物與地圖分別為陣列數值2與1，背景為數值0，通過不同的數值來表達地圖內部不同的物件偵測。

CPlayer：該類別實現了主遊玩角色方塊的功能，並繼承於CAnimation，改寫了gamelib.cpp中的void CAnimation::SwitchImage()，使用iterator智慧指標。遊戲軌道圖片使用資料結構陣列存取。

遊戲角色動作使用自由落體演算法，提供加速度與速度參數來實現近

似自由落體的物理行為。

橫向速度為背景常數速度移動，使用資料結構陣列，並且記錄於角色中。

自行開發與架構了建立地圖的流程書，在創建關卡時明顯提升開發速度。

<https://lvgl.io/tools/imageconverter>

<https://javl.github.io/image2cpp/>

1) 將遊戲背景的bmp檔案轉至 <https://javl.github.io/image2cpp/>

背景為白色

2) 將轉出來的圖片下載(testbmp.png) 修改成 障礙物為紅色 平台為黑色 背景為白色 如(testbmp.bmp樣)的圖片，此時還是png檔案，並用小畫家將寬高除以20倍，再轉成bmp檔案

3) <https://lvgl.io/tools/imageconverter> 放到這裡 轉成陣列 選項都是預設就行

4) 將轉出來的.c檔案打開 複製第一個陣列來修改 背景白色為0 黑色為1 紅色為2 如 test.c檔樣式

5) 將他修成二維陣列 一個關卡完成

四、結語

(一)問題及解決方法

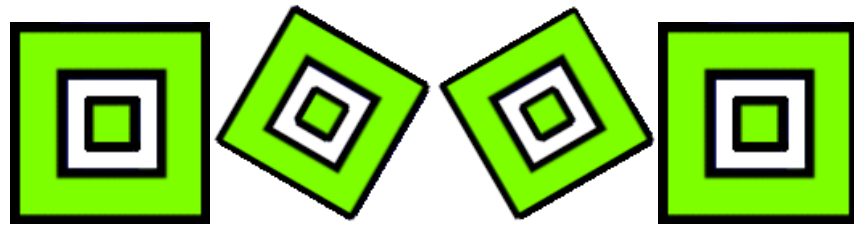
問題一：角色移動方面問題

解決方法：使用卷軸式設計，並非讓角色移動，而是讓背景移動來達到讓角色看起來像是往前動的感覺。

問題二：角色如何旋轉

解決方法：使用老師給予我們的framework當中的Animation來解決，我們將旋轉分成四個動作

，在跳躍的過程中一張一張撥放，來達到像是旋轉的效果，如下圖：



問題三：地圖背景問題

解決方法：這個問題並非完全解決，而是利用截圖來將遊戲內背景障礙物紀錄下來製作關卡，所以背景還是無法像原本遊戲那樣背景變色。

問題四：關卡的背景與實際陣列並非完全吻合

解決方法：這個問題我們還沒有完全解決，我們一個關卡的圖片是大約70000~100000 * 984，所以圖片相當大，若要將每一個像素轉換成陣列，那我們的檔案可能會卡死，所以我們退而求其次的將背景除以20來製作陣列，但這也導致遊戲內有些地方障礙物跳台等等與實際位置並不符合，若是我們的電腦足以負荷原圖陣列，也許這個問題也會隨之而消失。

(二)時間表(兩人合計)

Week	廉博旭	羅傑	Description
3	7	7	練習tutorial、git practice、tooggle track紀錄時間
4	10	10	建立素材，開始開發遊戲角色移動探索框架探索
5	10	10	建立素材，開始開發遊戲背景移動設計並且探索框架內animation方法
6	7	7	建立素材，開始開發遊戲背景移動設計並且探索框架內animation方法
7	8	8	開發遊戲，關卡部分完成度較高已經可以移動跟有看見障礙報告
8	9	9	開發遊戲，期中報告已將角色跳躍動作實現背景圖移動已完成
9	10	10	開發遊戲，開始實作地圖準備地圖陣列的開發流程
10	9	9	開發遊戲，實作地圖發現背景太長無法轉換成陣列 構思地圖陣列開發流程進行中
11	10	10	開發遊戲，實作地圖，請教協助，解決地圖過長問題 縮小地圖尺寸並設計陣列內的值完成
12	9	10	開發遊戲，實作地圖，請教協助，解決地圖過長問題，縮小地圖尺寸並設計陣列內的值，進行報告
13	9	11	開發遊戲，實作地圖，地圖開發流程完畢，製作完成第二關卡與切換選單改進，發現切換關卡的死機問題

14	8	10	開發遊戲，實作地圖第三關完成，切換關卡與死機問題解決
15	10	10	開發遊戲，實作地圖第四關完成，切換關卡完成，正在進行密技開發，遠端與助教討論
16	8	10	開發遊戲，實作地圖第五關完成，切換關卡完成，增加密技完成，進行音樂增加，遠端與助教討論
17	10	10	開發遊戲，實作地圖完成，切換關卡完成，音樂增加完成，遠端與助教討論
15週	134	141	

(三)貢獻比例

108590011 廉博旭 — 40%

108590450 羅傑 — 60%

(四)自我檢核

項目	項目	完成否	無法完成原因
1	Memory leak	是	
2	自訂遊戲icon	是	
3	全螢幕啟動	否	BMP大小與角色座標固定，調整困難
4	About畫面	是	
5	初始畫面說明按鍵及滑鼠之用法與秘技	否	放置於報告書內
6	上傳 setup/apk/source檔	是	
7	setup檔可正確執行	是	

8	報告字形點數..	是	
---	----------	---	--

(五)收獲

廉博旭：

這堂課帶來的收獲比想像中的還要多很多，遊戲開發也是未來我們資工系的一條出路，在現階段就能練習到簡單的遊戲開發可以讓我們更快適應未來工作。

羅傑：

這學期學到最大的收獲就是使用這門課所規定提供的framework進行開發，使用這個framework之前花了很多心力去讀懂文件的規格，還有花了很多時間去請教同學如何改寫。

最大的收獲是在提供的framework之上去學會開發程式，這項經驗將會成為未來開發程式的重要基石。

(六)心得、感想

廉博旭：

物件導向程式設計實習這門課在剛聽到的時候還不知道是要學習甚麼，以為跟上學期差不多，沒想到是要與組員共同開發一個遊戲，學習到了很多東西，例如根據framework來開發遊戲、與組員共同合作分工等等，雖然我們的遊戲完整性並非想像中那麼高，但這些都是難得的經驗，我會好好將這些學以致用。

羅傑：

這次的課程在中途因為疫情的關係變成遠距，有許多的難關與障礙必須克服，像是與組員共同開發時就無法當面溝通，或是改用郵件的方式與助教詢問問題，也無法與身旁較不認識的人詢問跟討論程式與框架的細節，是我比較困擾的地方。最後能夠在有限的能力上開發程式，也非常感謝助教與教授

們對我的協助跟幫忙。

(七)對於本課程的建議

廉博旭：

希望以後能夠更新一下framework，雖然這是練習但其中還是有蠻多不方便的地方，尤其是現在的遊戲越來越進步，舊的framework可能沒辦法完全做到新遊戲的畫面。

羅傑：

如果能夠提供更多關於framework的技術文件就能夠更快速的入手了。

```

#include "stdafx.h"
#include "Resource.h"
#include <mmsystem.h>
#include <ddraw.h>
#include "audio.h"
#include "gamelib.h"
#include "CPlayer.h"
namespace game_framework {

    CPlayer::CPlayer()
    {
        Initialize();
    }

    int CPlayer::GetX1()
    {
        return virtualx;
    }

    int CPlayer::GetY1()
    {
        return y;
    }

    int CPlayer::GetX2()
    {
        return x + player.Width();
    }

    int CPlayer::GetY2()
    {
        return y + player.Height();
    }

```

```

void CPlayer::Initialize()
{
    x = 500;
    virtualx = 500;
    y = 620;
    g = 0;
    vy = 20;
    floorY = 620;
    jumpY = 200;
    jumpflag = false;
}

void CPlayer::LoadBitmap() {
    player.AddBitmap(IDB_C1, RGB(255, 255, 255));
    player.AddBitmap(IDB_C3, RGB(255, 255, 255));
    player.AddBitmap(IDB_C2, RGB(255, 255, 255));
}

void CPlayer::SetX(int x) {
    virtualx = x;
}

void CPlayer::Jump() {
    if (jumpflag) {
        y -= (vy + g);
        if (y <= floorY - jumpY)
            vy = -1 * vy;
        else if (y > floorY) {
            vy = -1 * vy;
            jumpflag = false;
            y = floorY;
        }
        if (y == floorY - jumpY + 40)
            SwitchImage();
        if (y == floorY - jumpY)
            SwitchImage();
    }
}

```

```

void CPlayer::SetXY(int nx, int ny)
{
    if (ny >= 620) ny = 620;
    x = nx; y = ny;
}

void CPlayer::OnShow() {
    player.SetTopLeft(x, y);
    player.OnShow();
}

void CPlayer::SwitchImage() {
    player.SwitchImage();
}

void CPlayer::setFloorY(int y) {
    if (y >= 620) y = 620;
    floorY = y;
}
}

```

```

#include "stdafx.h"
#include "Resource.h"
#include <mmsystem.h>
#include <ddraw.h>
#include "audio.h"
#include "gamelib.h"
#include "CGamemap.h"

namespace game_framework {

CGamemap::CGamemap(int n) : mapnum(n)
{
}

void CGamemap::LoadBitmap() {
    gametrack[0].LoadBitmap(IDB_TRACK_1);
    gametrack[1].LoadBitmap(IDB_TRACK_2);
    gametrack[2].LoadBitmap(IDB_TRACK_3);
    gametrack[3].LoadBitmap(IDB_TRACK_4);
    gametrack[4].LoadBitmap(IDB_TRACK_5);
}

void CGamemap::OnShow() {
    if(mapnum == 1)
        gametrack[0].ShowBitmap();

    if (mapnum == 2)
        gametrack[1].ShowBitmap();

    if (mapnum == 3)
        gametrack[2].ShowBitmap();

    if (mapnum == 4)
        gametrack[3].ShowBitmap();

    if (mapnum == 5)
        gametrack[4].ShowBitmap();
}
}

```



```

void CGamemap::LoadBitmap() {
    gametrack[0].LoadBitmap(IDB_TRACK_1);
    gametrack[1].LoadBitmap(IDB_TRACK_2);
    gametrack[2].LoadBitmap(IDB_TRACK_3);
    gametrack[3].LoadBitmap(IDB_TRACK_4);
    gametrack[4].LoadBitmap(IDB_TRACK_5);
}

void CGamemap::OnShow() {
    if(mapnum == 1)
        gametrack[0].ShowBitmap();

    if (mapnum == 2)
        gametrack[1].ShowBitmap();

    if (mapnum == 3)
        gametrack[2].ShowBitmap();

    if (mapnum == 4)
        gametrack[3].ShowBitmap();

    if (mapnum == 5)
        gametrack[4].ShowBitmap();
}

void CGamemap::SetTopLeft(int x, int y) {
    if (mapnum == 1)
        gametrack[0].SetTopLeft(x, y);
    if (mapnum == 2)
        gametrack[1].SetTopLeft(x, y);
    if (mapnum == 3)
        gametrack[2].SetTopLeft(x, y);
    if (mapnum == 4)
        gametrack[3].SetTopLeft(x, y);
    if (mapnum == 5)
        gametrack[4].SetTopLeft(x, y);
}
}

```

```

#include "stdafx.h"
#include "Resource.h"
#include <mmsystem.h>
#include <ddraw.h>
#include "audio.h"
#include "gamelib.h"
#include "CMenu.h"
namespace game_framework {

    CMenu::CMenu()
    {
        Initialize();
    }

    void CMenu::Initialize()
    {
    }

    void CMenu::LoadBitmap() {
        menu.AddBitmap(IDB_MENU, RGB(255, 255, 255));
        menu.AddBitmap(IDB_MENULEVEL1, RGB(255, 255, 255));
        menu.AddBitmap(IDB_MENULEVEL2, RGB(255, 255, 255));
        menu.AddBitmap(IDB_MENULEVEL3, RGB(255, 255, 255));
        menu.AddBitmap(IDB_MENULEVEL4, RGB(255, 255, 255));
        menu.AddBitmap(IDB_MENULEVEL5, RGB(255, 255, 255));
    }

    void CMenu::OnShow() {
        menu.SetTopLeft(0, 0);
        menu.OnShow();
    }

    void CMenu::SwitchImage() {
        menu.SwitchImage();
    }
}

```

```

void CAnimation::OnShow()
{
    GAME_ASSERT bmp.size() != 0, "CAnimation: Bitmaps must be loaded before they are shown.");
    bmp_iter->SetTopLeft(x,y);
    bmp_iter->ShowBitmap();
}

int CAnimation::Top()
{
    GAME_ASSERT bmp.size() != 0, "CAnimation: Bitmaps must be loaded first.");
    return y;
}

int CAnimation::Width()
{
    GAME_ASSERT bmp.size() != 0, "CAnimation: Bitmaps must be loaded first.");
    return bmp_iter->Width();
}

void CAnimation::SwitchImage() {
    bmp_iter++;
    bmp_counter++;
    if (bmp_iter == bmp.end()) {
        bmp_iter = bmp.begin();
        bmp_counter = 0;
    }
}
}

```

```

#include "stdafx.h"
#include "Resource.h"
#include <mmsystem.h>
#include <ddraw.h>
#include "audio.h"
#include "gamelib.h"
#include "mygame.h"
#include <iostream>
int game_map_num = 0;
bool god = false;
namespace game_framework {
    CMenu menu; // game menu
    CGamemap *gamemap; // I'01I
    // 3o0class~°¹CÄ, äº¹CÄ, ¶}ÄYµe±²«¥ó
    ///////////////////////////////////////////////////

    CGameStateInit::CGameStateInit(CGame *g)
        : CGameState(g)
    {
    }

    void CGameStateInit::OnInit()
    {
        //
        // ·¹¹I«Ü|h®ÉiA0nInit, ũ±J©0{³ºº¹In²á«Ü|h®É¶iiC~°Á×$K²±¹CÄ, äº±H
        // µ¥²º±£@·DiA¹CÄ, ·|¥X²{i uLoading ... i viAÄã¥ÜLoading²º¶i«×iC
        //
        ShowInitProgress(0); // ±@¶}©l²ºloading¶i«×~°0%
        //
        // ¶}©l, ũ±J, ê®Æ
        //
        CAudio::Instance()->Load(AUDIO_MENU, "sounds\\menu.mp3"); // , ũ±J½s, ¹0²ºÄnµ
        CAudio::Instance()->Load(AUDIO_LEVEL1, "sounds\\TRACK_1.mp3"); // , ũ±J½s, ¹1²ºÄnµ
        CAudio::Instance()->Load(AUDIO_LEVEL2, "sounds\\TRACK_2.mp3"); // , ũ±J½s, ¹2²ºÄnµ
        CAudio::Instance()->Load(AUDIO_LEVEL3, "sounds\\TRACK_3.mp3"); // , ũ±J½s, ¹3²ºÄnµ
        CAudio::Instance()->Load(AUDIO_LEVEL4, "sounds\\TRACK_4.mp3"); // , ũ±J½s, ¹4²ºÄnµ
        CAudio::Instance()->Load(AUDIO_LEVEL5, "sounds\\TRACK_5.mp3"); // , ũ±J½s, ¹5²ºÄnµ

        menu.LoadBitmap();
        CAudio::Instance()->Play(AUDIO_MENU, true);
        gamemap = new CGamemap(game_map_num);
        gamemap->LoadBitmap(); // , ũ±J I'ººº¹I$Î
    }
}

```

```

Sleep(300); // 0ñºCíA¥H«K-Y²M·i¶i«×iA¹ê»U¹CÄ,½D$R°£!¹Sleep
//
// ¹OnInit°Ê$@·|±µ"iCGameStaterRun::OnInit()iA©0¥H¶i«×ÄÜ"S"i100%
//
}

void CGameStateInit::OnBeginState()
{
}

void CGameStateInit::OnKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    const char KEY_ESC = 27;
    const char KEY_LEFT_ARROW = 0x27;
    const char KEY_SPACE = ' ';
    const char GOD = 0x10;

    if (nChar == KEY_SPACE) {
        GotoGameState(GAME_STATE_RUN); // ¼Á'«|ÜGAME_STATE_RUN
        CAudio::Instance()->Stop(AUDIO_MENU);
    }
    else if (nChar == KEY_ESC) // Demo Äö³¬¹CÄ,²º¼è²k
        PostMessage(AfxGetMainWnd()->m_hWnd, WM_CLOSE, 0, 0); // Äö³¬¹CÄ
    else if (nChar == KEY_LEFT_ARROW) {
        menu.SwitchImage();
        game_map_num++;
        if (game_map_num > 5) game_map_num = 0;
    }

    if (nChar == GOD) {
        god = true;
    }
}

```

```

void CGameStateInit::OnLButtonDown(UINT nFlags, CPoint point)
{
    GotoGameState(GAME_STATE_RUN); // ¼Á'«|ÜGAME_STATE_RUN
}

void CGameStateInit::OnShow()
{
    //
    // ¶K=Wlogo
    //
    menu.OnShow();

    //
    // Demo¿Ä¹ö¿r«¬²º"İ¥İiA=£¹L¶}µ©É½D²É¶qÁ×$K²½±µ"İ¥İ¿r«¬iA$İ¥İCMovingBitmap=ñ,û¿n
    //

    CDC *pDC = CDDraw::GetBackCDC(); // "ú±o Back Plain ²º CDC
    CFont f, *fp;
    f.CreatePointFont(160, "Times New Roman"); // ²£¥İ font f; 160²i¥Ü16 point²º¿r
    fp = pDC->SelectObject(&f); // ¿i¥İ font f
    pDC->SetBkColor(RGB(0, 0, 0));
    pDC->SetTextColor(RGB(255, 255, 0));

    pDC->SelectObject(fp); // ©ñ±¼ font f (¼d,U¼£n²|¼F©ñ±¼)
    CDDraw::ReleaseBackCDC(); // ©ñ±¼ Back Plain ²º CDC
}

////////////////////////////////////
// ³oÖclass¬º¹CÄ,²ºµ²$ô²¬ºA(Game Over)
////////////////////////////////////

CGameStateOver::CGameStateOver(CGame *g)
: CGameState(g)
{
}

```

```

void CGameStateOver::OnMove()
{
    counter--;
    if (counter < 0)
        GotoGameState(GAME_STATE_INIT);
}

void CGameStateOver::OnBeginState()
{
    counter = 30 * 5; // 5 seconds
}

void CGameStateOver::OnInit()
{
    //CAudio::Instance()->Play(AUDIO_MENU, true);
    //
    // ·i¹İ«Ü|h®ÉiA0nInit, ü±J©Ò|³²²¹İn³á«Ü|h®Éŋi iC¬°Á×$K²±¹CÁ,²²²H
    // µŸ²²²£@·ĐiA¹CÁ,·|ŸX²{iuLoading ...iviAÄ³ŸÜLoading²²ŋi«×iC
    //
    ShowInitProgress(66); // ±µÓ«e±@Ó²¬²A²²ŋi«×iA|¹³Bŋi«×µø¬°66%
    //
    // ŋ}©l, ü±J, ê®Æ
    //
    Sleep(300); // ©ñ²CiAŸH«K¬Ÿ²M·iŋi«×iA¹ê»Ú¹CÁ,½Đ$R°£|¹Sleep
    //
    // ³İ²×ŋi«×¬°100%
    //
    ShowInitProgress(100);
}

```

```

void CGameStateRun::OnMove() // ²³¼°Ê¹CÁ,²²,¬Á
{
    //
    // |p²G$£±±×$İcursor²²¼Ê|i iA«h±N±U±µ{|i²²comment"ú®ø$ŸŸi
    //
    // SetCursor(AfxGetApp()->LoadCursor(IDC_GAMECURSOR));
    //
    //
    // ²³¼°ÊI'²¹İ²²²y¼Đ
    //

    int nowmap = game_map_num - 1;
    int maxl = 0;

    if (nowmap == 0 || nowmap == 1 || nowmap == 2) {
        maxl = -14480;
    }
    else if(nowmap == 3){
        maxl = -15280;
    }
    else if (nowmap == 4) {
        maxl = -14480;
    }

    if (gamemap->gametrack[nowmap].Left() == maxl) {
        if (game_map_num == 1)
            CAudio::Instance()->Stop(AUDIO_LEVEL1); // ¼·©ñ
        if (game_map_num == 2)
            CAudio::Instance()->Stop(AUDIO_LEVEL2); // ¼·©ñ WAVE
        if (game_map_num == 3)
            CAudio::Instance()->Stop(AUDIO_LEVEL3); // ¼·©ñ WAVE
        if (game_map_num == 4)
            CAudio::Instance()->Stop(AUDIO_LEVEL4); // ¼·©ñ WAVE
        if (game_map_num == 5)
            CAudio::Instance()->Stop(AUDIO_LEVEL5); // ¼·©ñ WAVE
        GotoGameState(GAME_STATE_OVER);
        god = false;
    }
}

```

```

void CGameStateRun::OnBeginState()
{
    gamemap->mapnum = game_map_num;

    const int BALL_GAP = 90;
    const int BALL_XY_OFFSET = 45;
    const int BALL_PER_ROW = 7;
    const int HITS_LEFT = 10;
    const int HITS_LEFT_X = 590;
    const int HITS_LEFT_Y = 0;
    const int BACKGROUND_X = 0;
    const int ANIMATION_SPEED = 15;
    for (int i = 0; i < NUMBALLS; i++) {
        int x_pos = i % BALL_PER_ROW;
        int y_pos = i / BALL_PER_ROW;

    }

    if (god == true) {
        gamemap->SetTopLeft(-13000, 0);
    }

    else {
        gamemap->SetTopLeft(0, 0);

    }

    if (game_map_num == 1)
        CAudio::Instance()->Play(AUDIO_LEVEL1, true);
    if (game_map_num == 2)
        CAudio::Instance()->Play(AUDIO_LEVEL2, true);
    if (game_map_num == 3)
        CAudio::Instance()->Play(AUDIO_LEVEL3, true);
    if (game_map_num == 4)
        CAudio::Instance()->Play(AUDIO_LEVEL4, true);
    if (game_map_num == 5)
        CAudio::Instance()->Play(AUDIO_LEVEL5, true);

    //CAudio::Instance()->Play(AUDIO_DING, false);
    //CAudio::Instance()->Play(AUDIO_NTUT, true);
}

```

```

if (gamemap->gametrack[nowmap].Left() > maxl) {
    gamemap->SetTopLeft(gamemap->gametrack[nowmap].Left() - 20, 0);
    c_player.SetX(c_player.GetX1() + 20);
}

gamemap->gametrack[nowmap].SetTopLeft(gamemap->gametrack[nowmap].Left(), gamemap->gametrack[nowmap].Top() + 1);

//
// ²¼°Éplayer
//
c_player.Jump();

//
// °»´úplayer, I¼²²«
//
int temp;
if (nowmap == 0)
    temp = gamemap->map_1[c_player.GetY1() / 20][c_player.GetX1() / 20];

if (nowmap == 1)
    temp = gamemap->map_2[c_player.GetY1() / 20][c_player.GetX1() / 20];

if (nowmap == 2)
    temp = gamemap->map_3[c_player.GetY1() / 20][c_player.GetX1() / 20];

if (nowmap == 3)
    temp = gamemap->map_4[c_player.GetY1() / 20][c_player.GetX1() / 20];

if (nowmap == 4)
    temp = gamemap->map_5[c_player.GetY1() / 20][c_player.GetX1() / 20];

if (temp == 2) {
    if (!god) {
        if (game_map_num == 1)
            CAudio::Instance()->Stop(AUDIO_LEVEL1);
        if (game_map_num == 2)
            CAudio::Instance()->Stop(AUDIO_LEVEL2);
        if (game_map_num == 3)
            CAudio::Instance()->Stop(AUDIO_LEVEL3);
        if (game_map_num == 4)
            CAudio::Instance()->Stop(AUDIO_LEVEL4);
    }
}

```

```

        if (game_map_num == 4)
            CAudio::Instance()->Stop(AUDIO_LEVEL4);           // ¼·ñ WAVE
        if (game_map_num == 5)
            CAudio::Instance()->Stop(AUDIO_LEVEL5);           // ¼·ñ WAVE

        gamemap->SetTopLeft(0, 0);
        c_player.SetX(500);
        c_player.SetXY(500, 620);
        c_player.setFloorY(620);
        GotoGameState(GAME_STATE_OVER);
    }
}
else if (temp == 0) {
    c_player.setFloorY(c_player.GetY1());
    c_player.SetXY(500, c_player.GetY1());
}
else if (temp == 1 && !c_player.jumpflag) {
    if (c_player.GetY1() < 700) {
        c_player.SetXY(500, c_player.GetY1() + 20);
        c_player.setFloorY(c_player.GetY1() + 20);
    }
}
}
}
}

```

```

void CGameStateRun::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    const char KEY_SPACE = 0x20; // keyboard space

    if (nChar == KEY_SPACE) {
        c_player.jumpflag = true;
    }
}

void CGameStateRun::OnKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    const char KEY_SPACE = 0x20; // keyboard space
    if (nChar == KEY_SPACE) {
    }
}
}

```