

小遊戲程式碼結構

大概

- `int main(), void Init()`: 初始化
- `void Display()`: 負責和OpenGL溝通、繪製畫面
- `void System(int)`: 每25ms呼叫一次，處理時間軸
(像是處理球的移動)
- 處理Input的函數：
 - `void SpecialKeyDown(int,int,int)`
 - `void SpecialKeyUp(int,int,int)`
 - `void NormalKeyDown(unsigned char,int,int)`
 - `void NormalKeyUp(unsigned char,int,int)`

```
int main()
```

- 初始化 glut
- 註冊函數：告訴glut遇到哪些事情要呼叫哪個函數

```
glutDisplayFunc(Display);
```

- 進入遊戲：

```
glutMainLoop();
```

void Init()

- 把全域變數初始化

```
float ball_x, ball_y;  
float ball_vx, ball_vy;  
  
float bar_x, bar_y;  
float bar_vx, bar_vy;  
float bar_width, bar_height;  
  
float block_x[100], block_y[100];  
float block_width, block_height;  
int block_cnt;  
  
bool trigger_left_key, trigger_right_key;
```

void Display()

- 負責和OpenGL溝通、繪製畫面

像是

```
glBegin(GL_QUADS);
    glColor3f(255, 0, 0);
    glVertex3f(0.0, 0.0, 0.0);
    glVertex3f(1.0, 0.0, 0.0);
    glVertex3f(1.0, 1.0, 0.0);
    glVertex3f(0.0, 1.0, 0.0);
glEnd();
```

void System(int)

- 每25ms呼叫一次
- 處理時間軸 (像是處理球的移動)

```
glutTimerFunc(25, SystemTimer, 1);
```

處理Input的函數：

- 函數們：

- void SpecialKeyDown(int,int,int)
- void SpecialKeyUp(int,int,int)
- void NormalKeyDown(unsigned char,int,int)
- void NormalKeyUp(unsigned char,int,int)

- 可以注意到 int main() 裡面有

```
glutKeyboardFunc(NormalKeyDown);  
glutKeyboardUpFunc(NormalKeyUp);  
glutSpecialFunc(SpecialKeyDown);  
glutSpecialUpFunc(SpecialKeyUp);
```

這就是告訴glut當鍵盤事件觸發時要呼叫的函數

小遊戲狀態

- 全域變數裡面有

```
int game_state;  
  
const int GAME_STATE_PLAY = 0;  
const int GAME_STATE_END = 1;
```

是什麼？

- 它儲存的是當前的遊戲狀態
- 這遊戲分兩個狀態：
 - 玩遊戲中
 - 遊戲結束
- 使得 `Display`, `SystemTimer` 知道要怎麼處理，像是假如當前是 `GAME_STATE_END`，那 `SystemTimer` 根本不需要更改球的座標，只需要顯示 **Game Over**

繪圖函數

概念

- 主要操作分成兩類
 - 設定畫筆顏色
 - 繪製圖形
- 其實還有更多XD
- 不過小遊戲就用到這兩個

操作--設定顏色

```
glColor3f(R, G, B);
```

- 參數
 - 不同於 $[0, 255]$ 區間，這裏 $R, G, B \in [0, 1]$
 - 換算方式：假如要改成的顏色是(101, 103, 107)那就呼
叫

```
glColor3f(101/255.0, 103/255.0, 107/255.0);
```

操作--繪製圖形

```
glColor3f(255, 0, 0);
glBegin(GL_QUADS);
    glVertex3f(0.0, 0.0, 0.0);
    glVertex3f(1.0, 0.0, 0.0);
    glVertex3f(1.0, 1.0, 0.0);
    glVertex3f(0.0, 1.0, 0.0);
glEnd();
```

- `glBegin(GL_QUADS)` 告訴gl要準備繪製四邊形
- `glVertex3f(x, y, z)` 告訴gl點座標
 - 這裏會發現會有z軸，可是都設定成0
- `glEnd()` 告訴gl指定座標完成。