# Stack

E14092013 陳艾揚

5★
1 則評論

0+
下載次數

3+
3 歲以上

安裝在更多裝置上

這個應用程式與你的每部裝置都相容（大概）

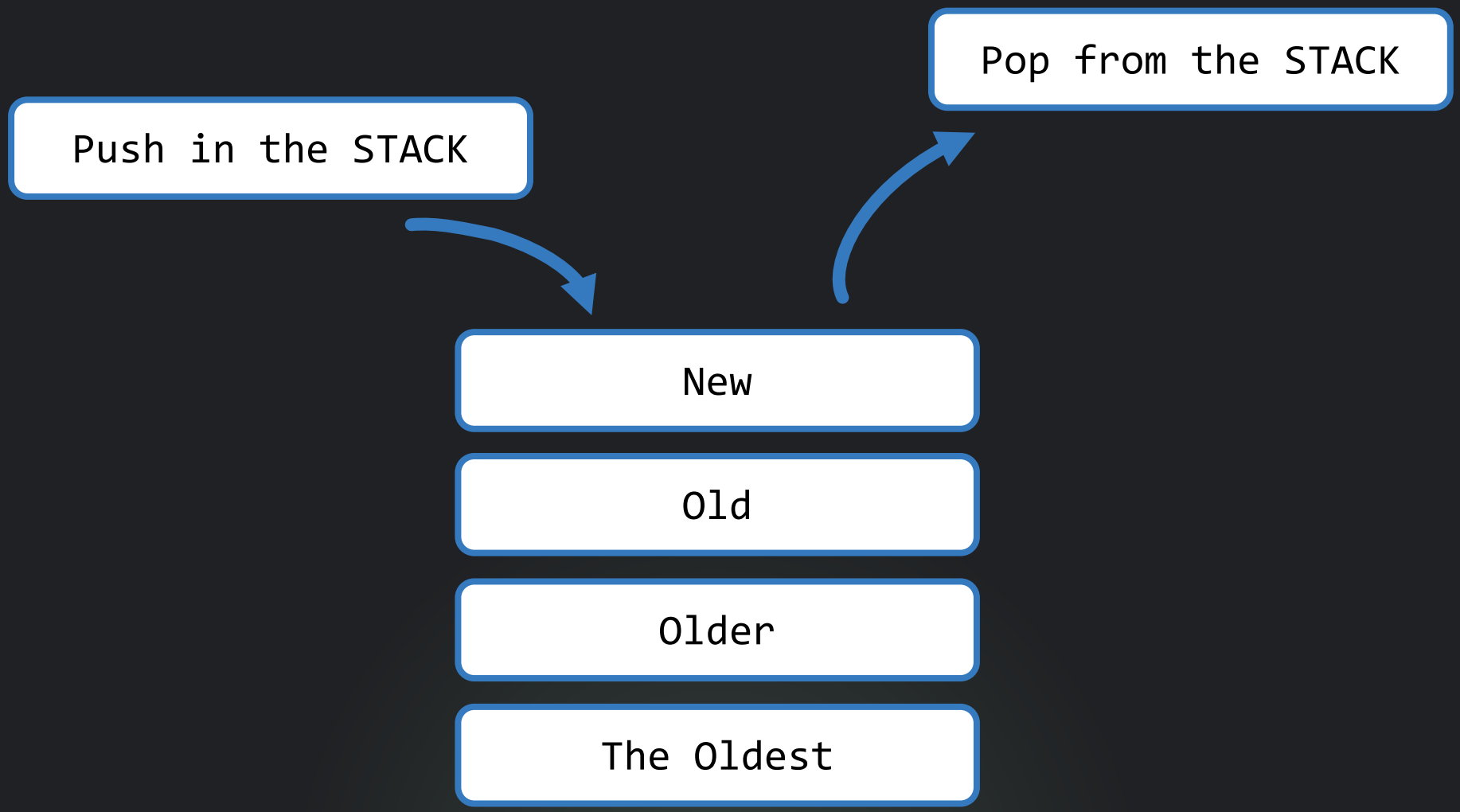
OBJECTIVE.h

```
/*
  1. 能跑。
  2. 展現學到的東東……
  3. 能跑。
*/
```

# Motivation.h

```
bool This_Semaster_is_happy(){

    glPushMatrix();
// everything suffering
    glPopMatrix();

    if(all_pass){ return TURE; }
    else { return FALSE; }
}
```

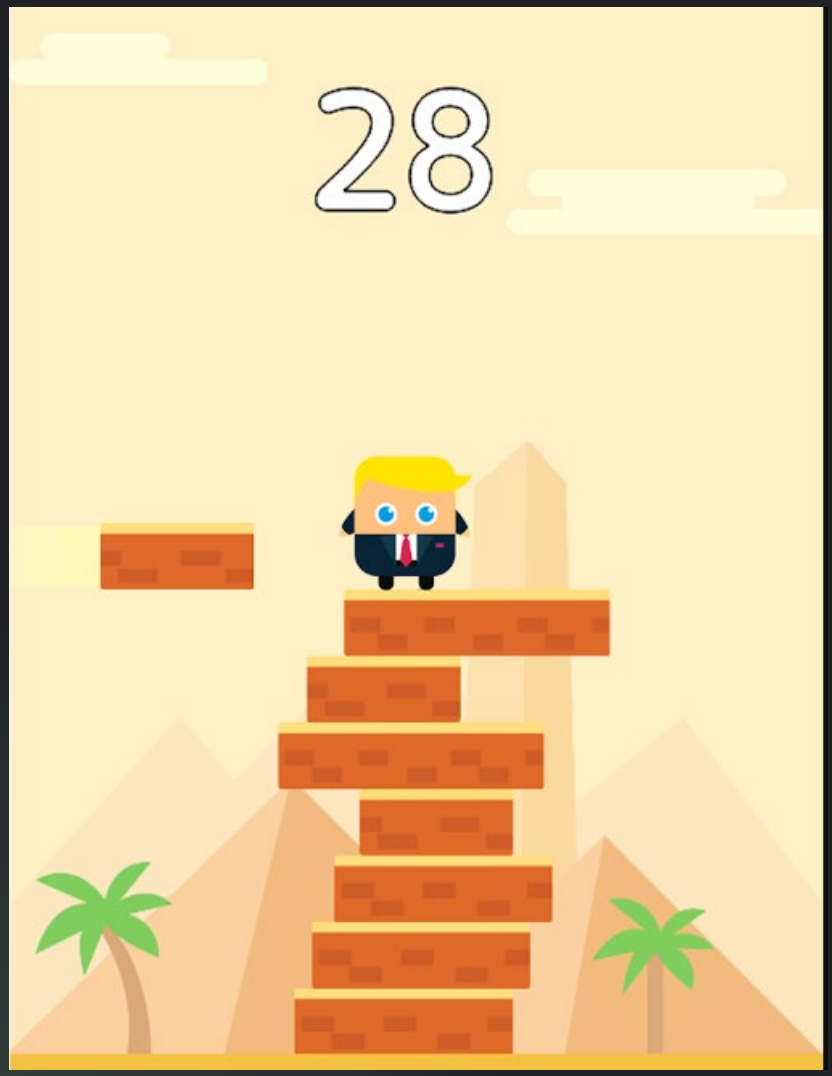
# What_is_STACK.H




LA CAPILLA SIXTINA
JAPON
EL LIBRO DE LOS MISTERIOS Y LAS POTENCIAS OCULTAS
SIDNEY SHELDON NADA ES ETERNO
EL PIANISTA
AQUI EN LA TIERRA

Motivation.h

```
switch(STACK_GAME){
case 1:
/*
  @Advenworks
  Tap Tower
*/
```

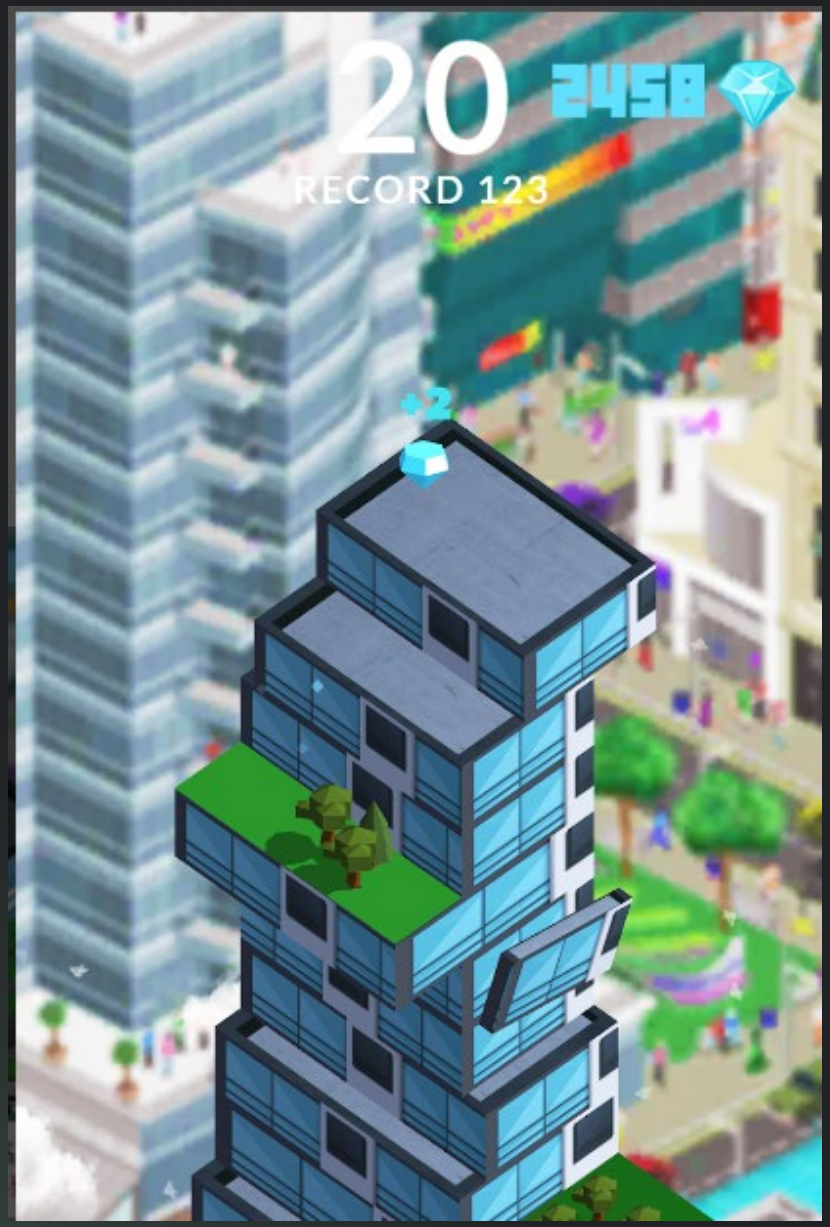


# Motivation.h

```
case 2:
/*
   @MARS.
   Tower Tap
*/
```

# Motivation.h

```
case 3:
/*
  @Artik Games
  TOWER BUILDER
*/
```



# Motivation.h

```
case 4:
/*
@Hyper Master Games
    Tap Tap Tower
    Relaxing Game
*/
```

# Motivation.h

```
case 5:
/*
  @Patrick König
  Tap and Stack-
  Building Tower
*/
```

# Stack

Ketchapp

含廣告內容 · 應用程式內購

4.2★
44.2萬 則評論

5000萬+
下載次數

3+
3 歲以上

安裝在更多裝置上

這個應用程式與你的每部裝置都相容

Motivation.h

/* 這個遊戲的數學好像比較簡單… */

Watch 1

Search (Ctrl+E) Search Depth: 3

| Name | Value | Type |
| --- | --- | --- |
| Can_show_my_study | TRUE | bool |
| Is_3_dimension | TRUE | bool |
| using_Color | TRUE | bool |
| using_Texture | TRUE | bool |
| using_Light | TRUE | bool |
| using_Buffers | TRUE | bool |
| using_Interactive | TRUE | bool |

OBJECTIVE.h

```
/*
    1. 能跑。
    2. 展現學到的東東。
    3. 把STACK複製一個。
    4. 完成度高一點
    5. 能跑。
    6. 期末報告能~~滿分高分~~及格
*/
```

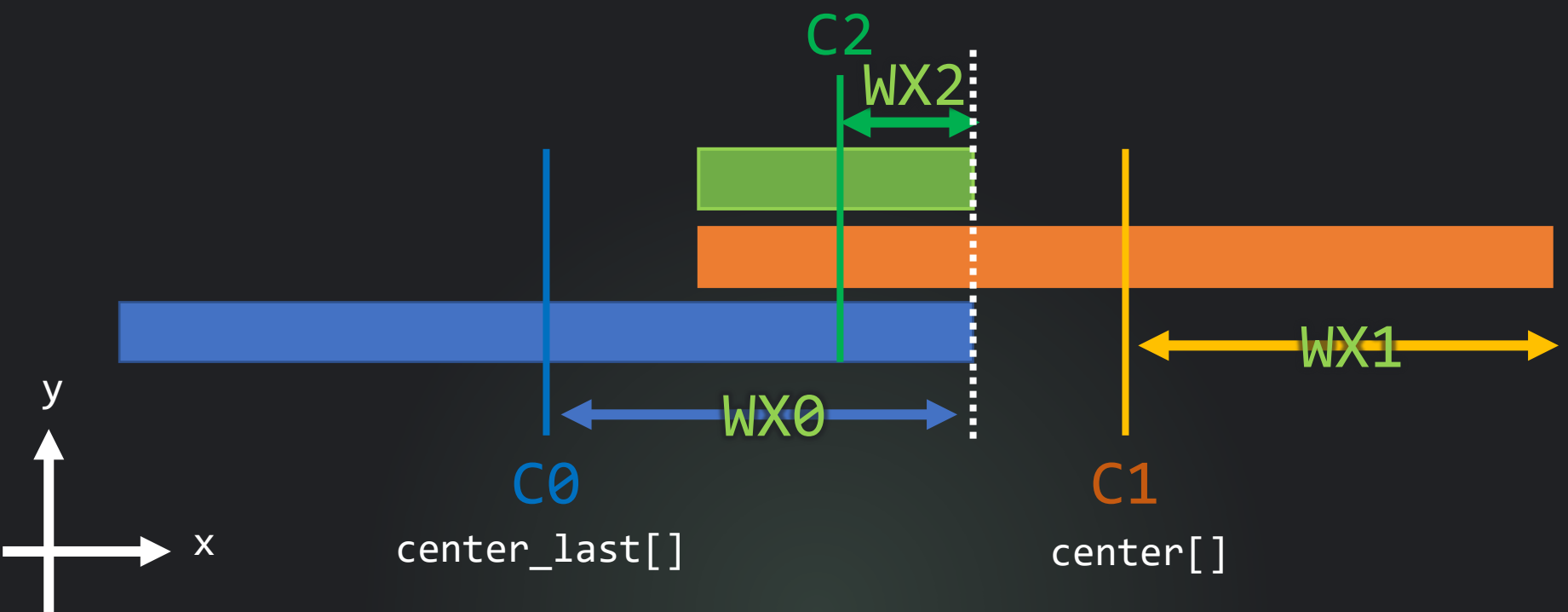
MATHEMATIC.h

// 關於堆疊下一塊的參數

$$WX_0 = WX_1$$

$$C_2 = \frac{(C_0 + C_1)}{2}$$

$$WX_2 = WX_0 - (C_2 - C_0)$$



開始偵錯(G)　　F5

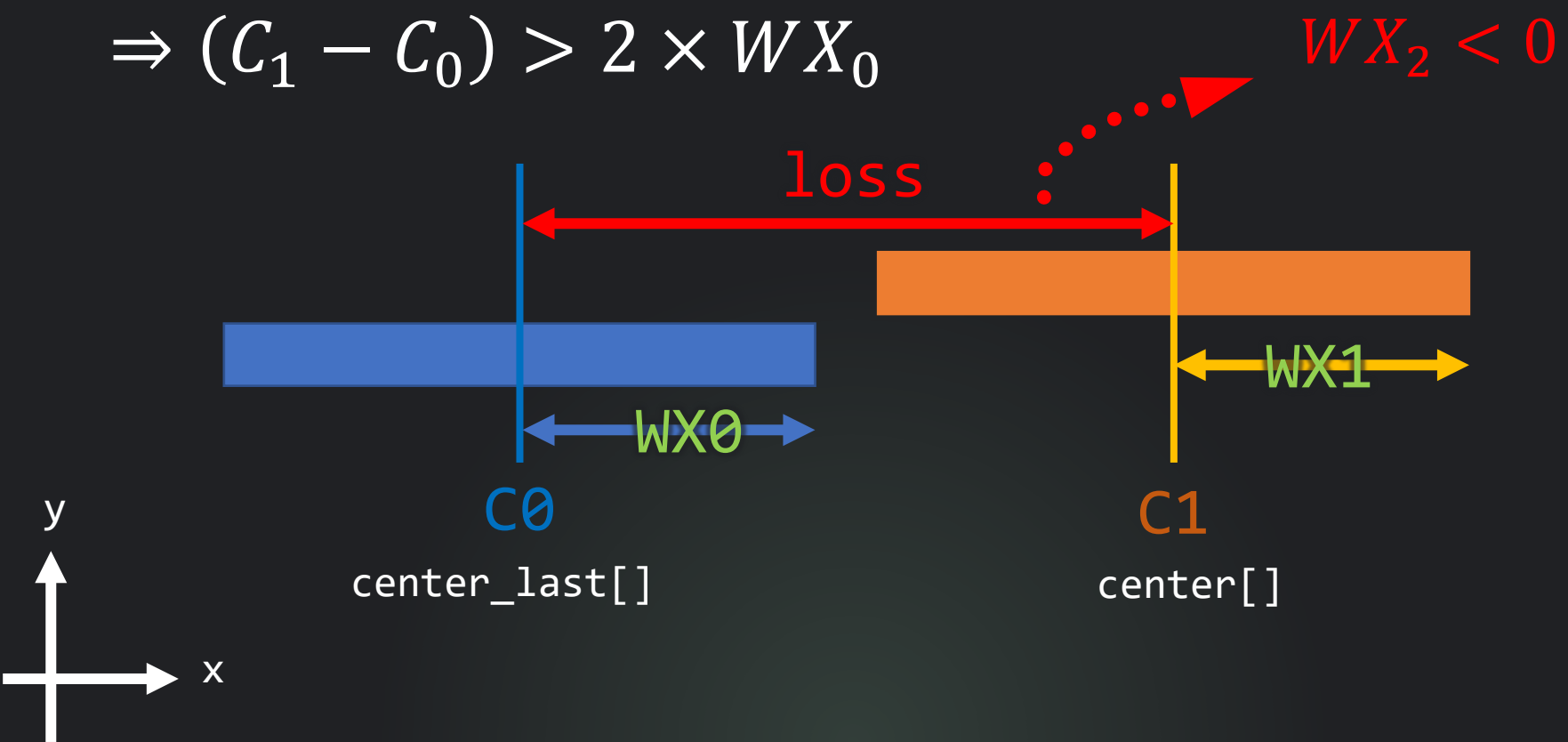
# MATHEMATIC.h

// 關於失敗條件

$$C_2 = \frac{(C_0 + C_1)}{2} \in (C_0 - WX_0, C_0 + WX_0)$$

$$WX_0 = WX_1$$

$$\Rightarrow (C_1 - C_0) > 2 \times WX_0$$



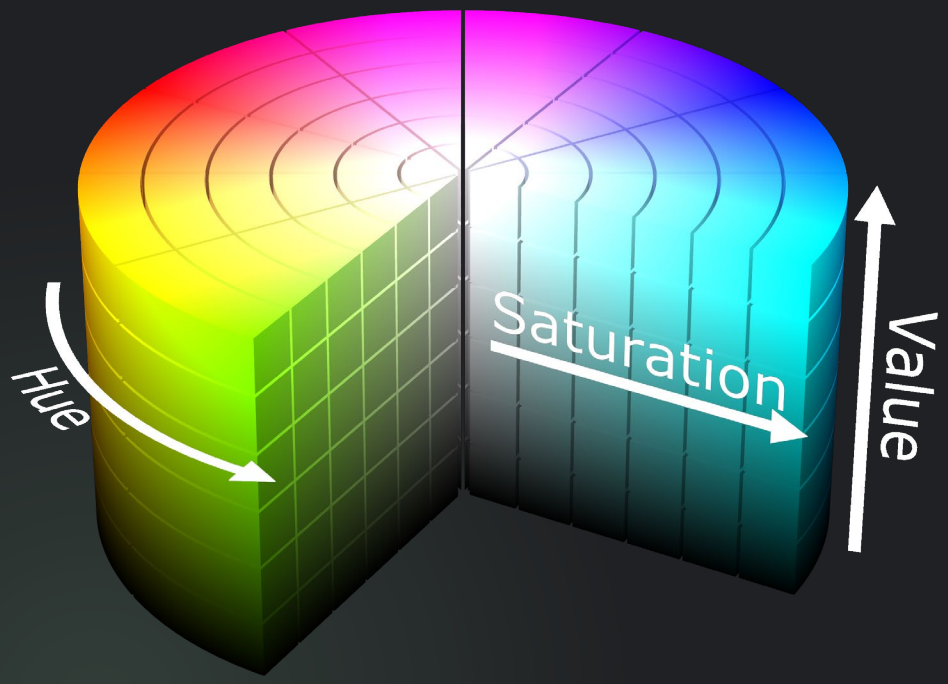
- 啟動但不偵錯(H)

CUBOID.h

```
/* 方塊 */
class CUBOID {
public:
    // 八個頂點座標
    float vertex[8][3];
    // RGB 三個顏色
    GLubyte color[3];
};
```

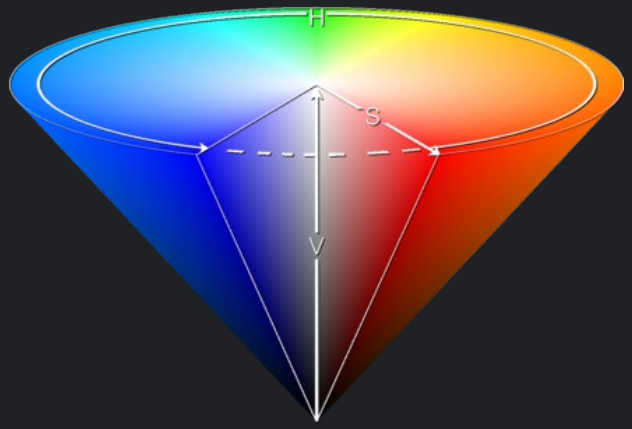
# What_is_HSV.H

```
#define H  Hue
// 色相：  具體顏色
#define S  Saturation
// 飽和度：濃淡
#define V  Value
// 彩度：  多黑
```



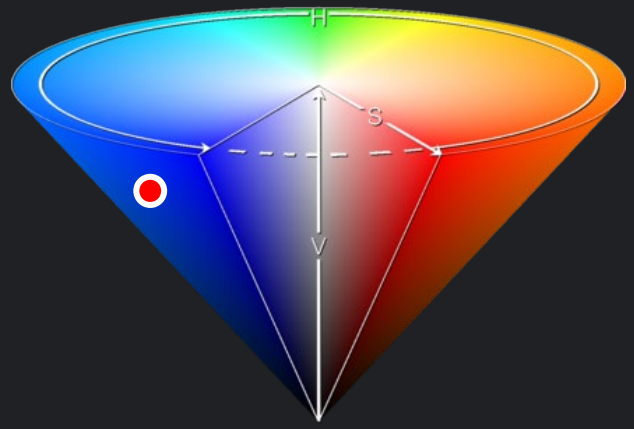
# What_is_HSV.H

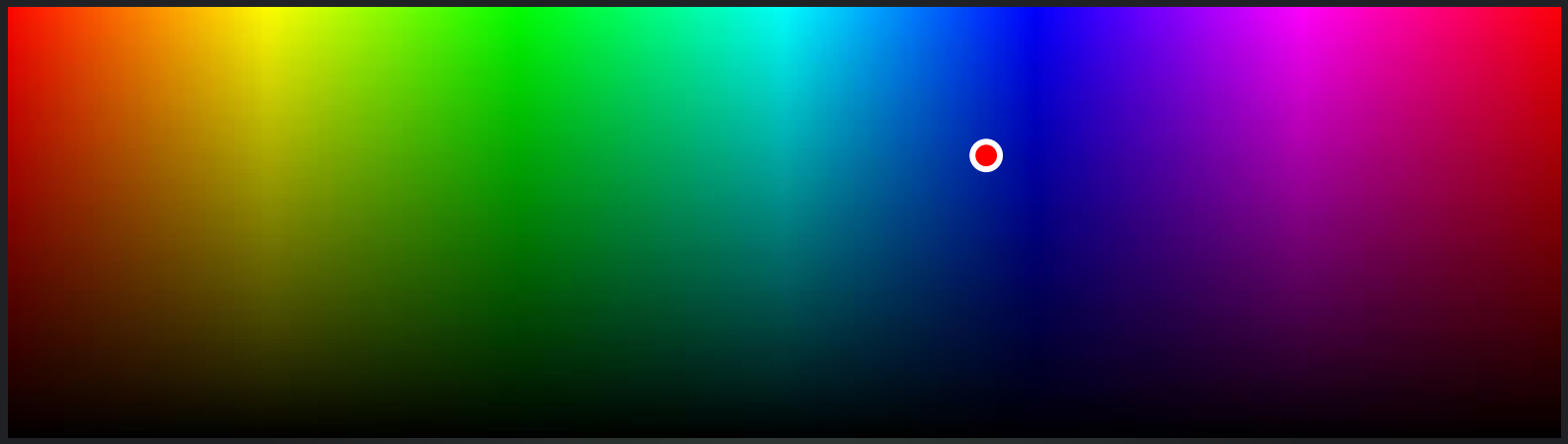
```
#define H  Hue
#define S  Saturation
#define V  Value
```




# Random_color.h

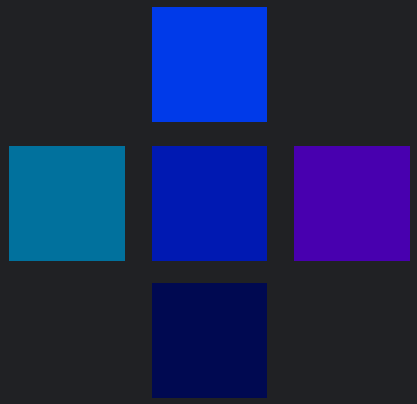
```
// 1.隨機決定起點
```

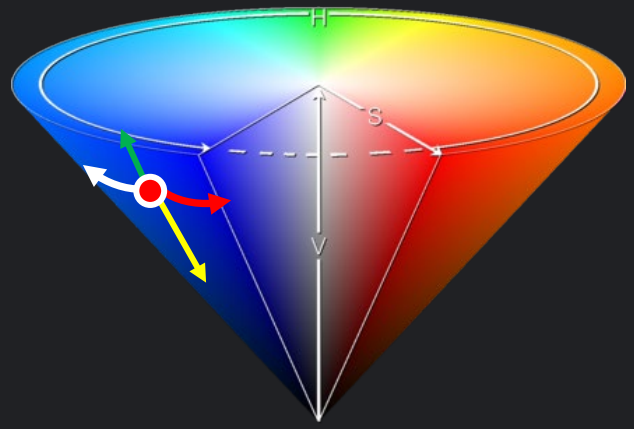


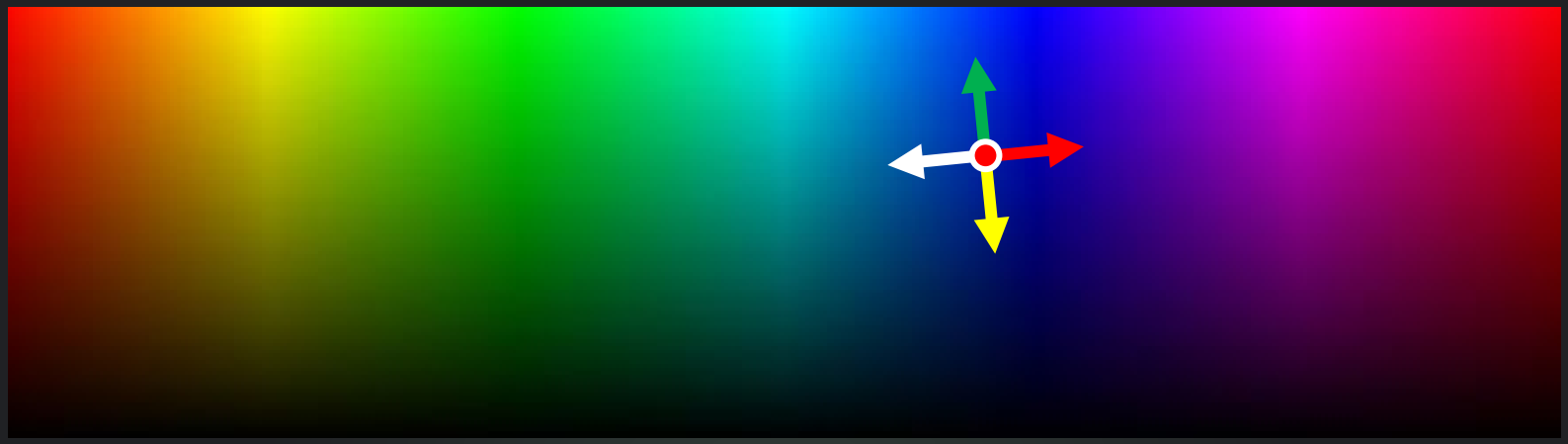



# Random_color.h

// 2.隨機決定方向

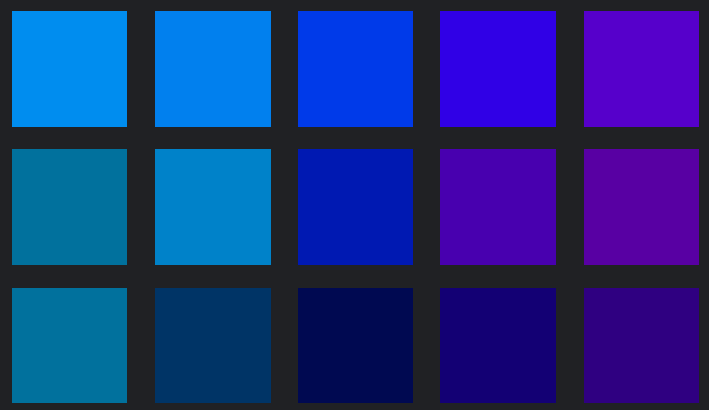


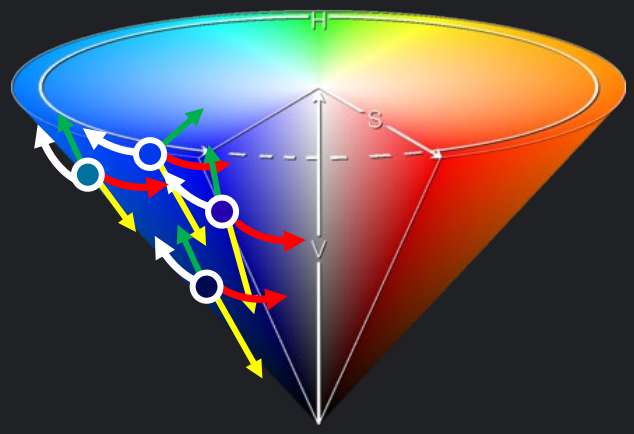


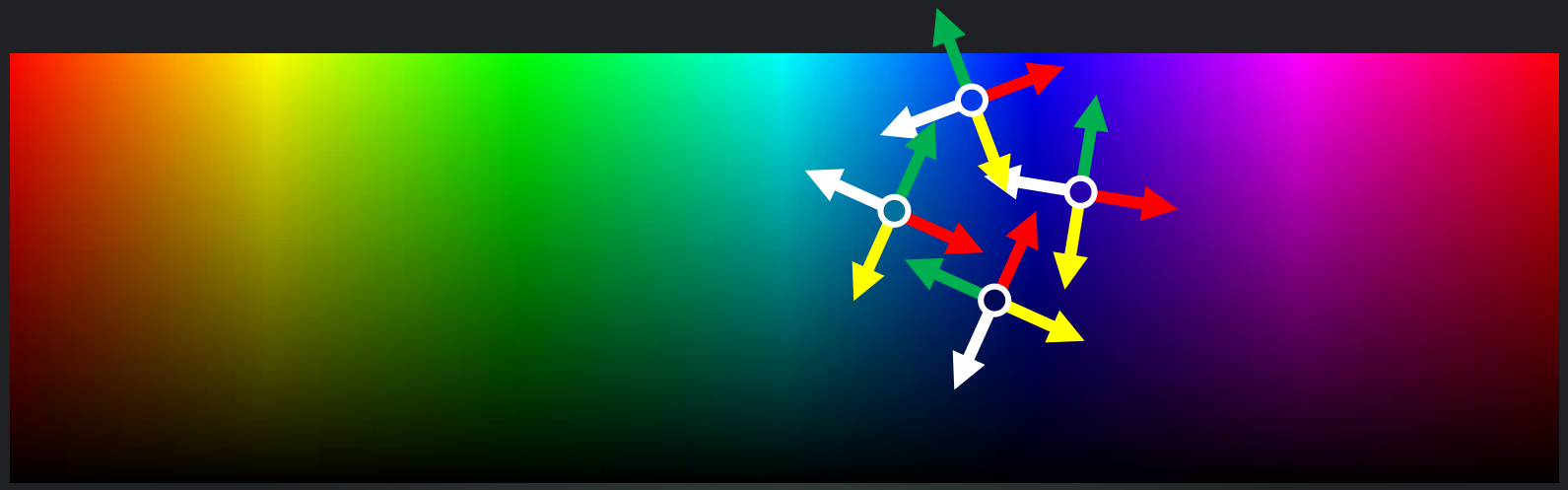



Random_color.h

// 3.作為新起點





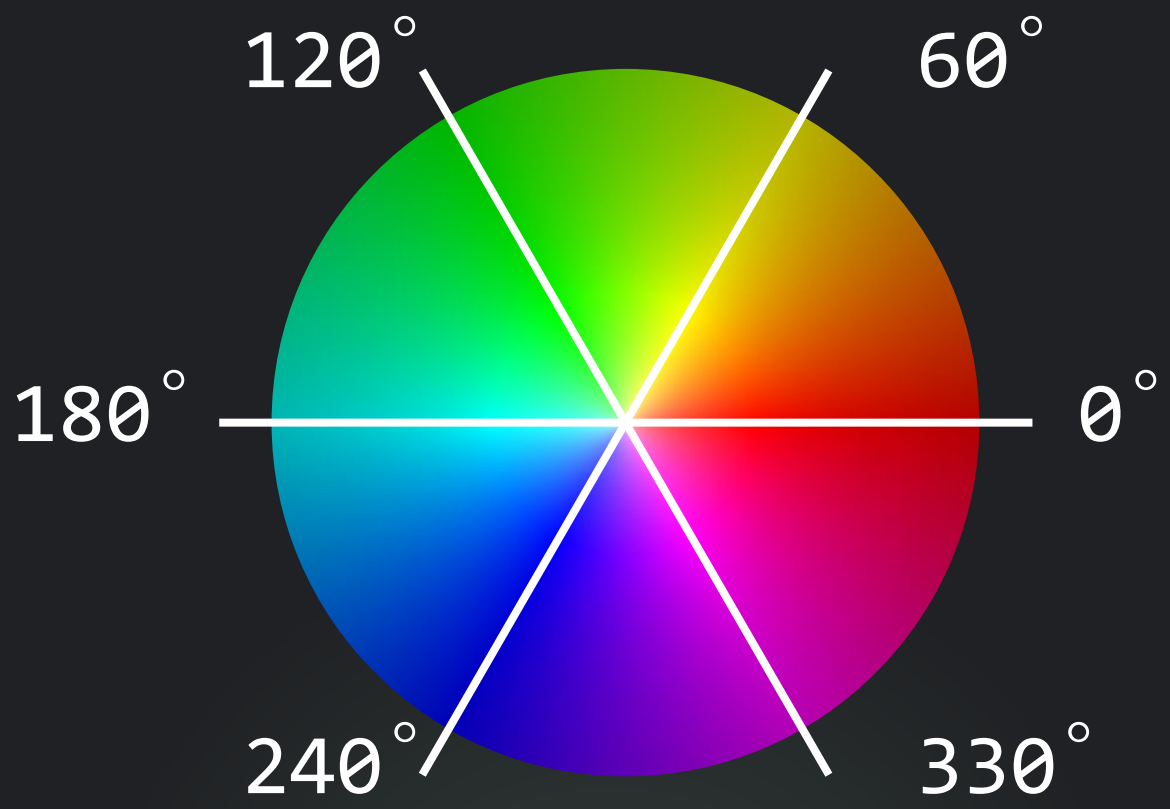



# HSV_2_RGB.h

# HSV_2_RGB.h

```
int H = (CuboidColor_H / 60) % 6;
float v_ = float(CuboidColor_V) / 100.0f;
float s_ = float(CuboidColor_S) / 100.0f;
float f_ = float(CuboidColor_H) / 60.0f - float(H);
float p_ = v_ * (1 - s_);
float q_ = v_ * (1 - f_ *s_);
float t_ = v_ * (1 - (1 - f_)*s_);
float r, g, b;
switch (H){
    case 0:  r = v_; g = t_; b = p_; break;
    case 1:  r = q_; g = v_; b = p_; break;
    case 2:  r = p_; g = v_; b = t_; break;
    case 3:  r = p_; g = q_; b = v_; break;
    case 4:  r = t_; g = p_; b = v_; break;
    case 5:  r = v_; g = p_; b = q_; break;
    default: break;
}
```

# HSV_2_RGB.h



- 啟動但不偵錯(H)

Finish?

No.

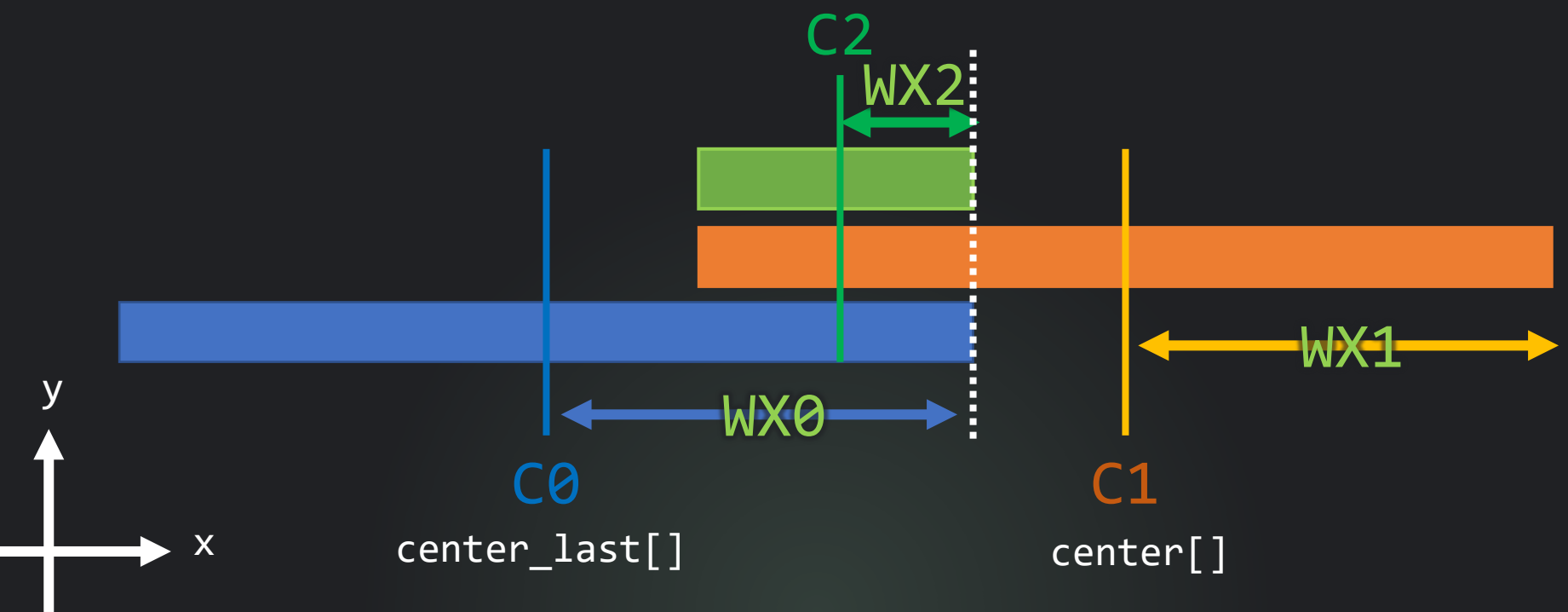
MATHEMATIC.h

// 關於堆疊下一塊的參數

$$WX_0 = WX_1$$

$$C_2 = \frac{(C_0 + C_1)}{2}$$

$$WX_2 = WX_0 - (C_2 - C_0)$$



# MATHEMATIC.h

## // 關於掉落方塊的參數

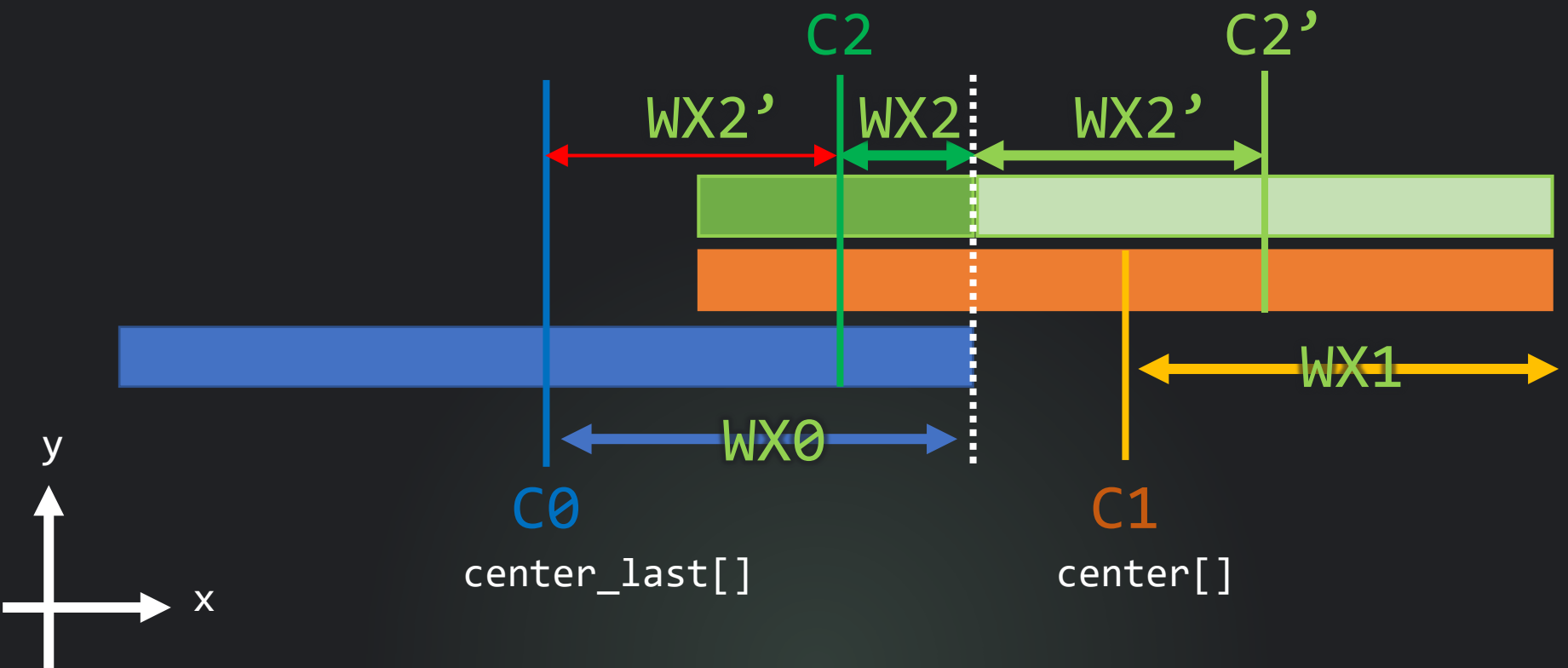
$$WX_0 = WX_1$$

$$C_2 = \frac{(C_0+C_1)}{2}$$

$$WX_2 = WX_0 - (C_2 - C_0)$$

$$C_2' = \frac{(C_1+WX_1)+(C_0+WX_0)}{2} = \frac{C_0+C_1}{2} + WX_0 = C_2 \pm WX_0$$

$$WX_2' = C_2' - (C_0 + WX_0) = C_2 - C_0$$



- 啟動但不偵錯(H)

fallingCUBOID.h

```
class CUBOID {
public:
    float vertex[8][3];
    GLubyte color[3];
};
class fallingCUBOID :public CUBOID{
public:
    bool osci; // 旋轉方向
    float Rot; // 旋轉量
};
```

SLOW_FALLING.h

```
if (t < hy) { // 緩降
    glMatrixMode(GL_MODELVIEW);
    glPushMatrix();
    glTranslatef(0, hy - t, 0);
}
/* other rendering thingy */
if (t < hy) {
    glPopMatrix();
}
```

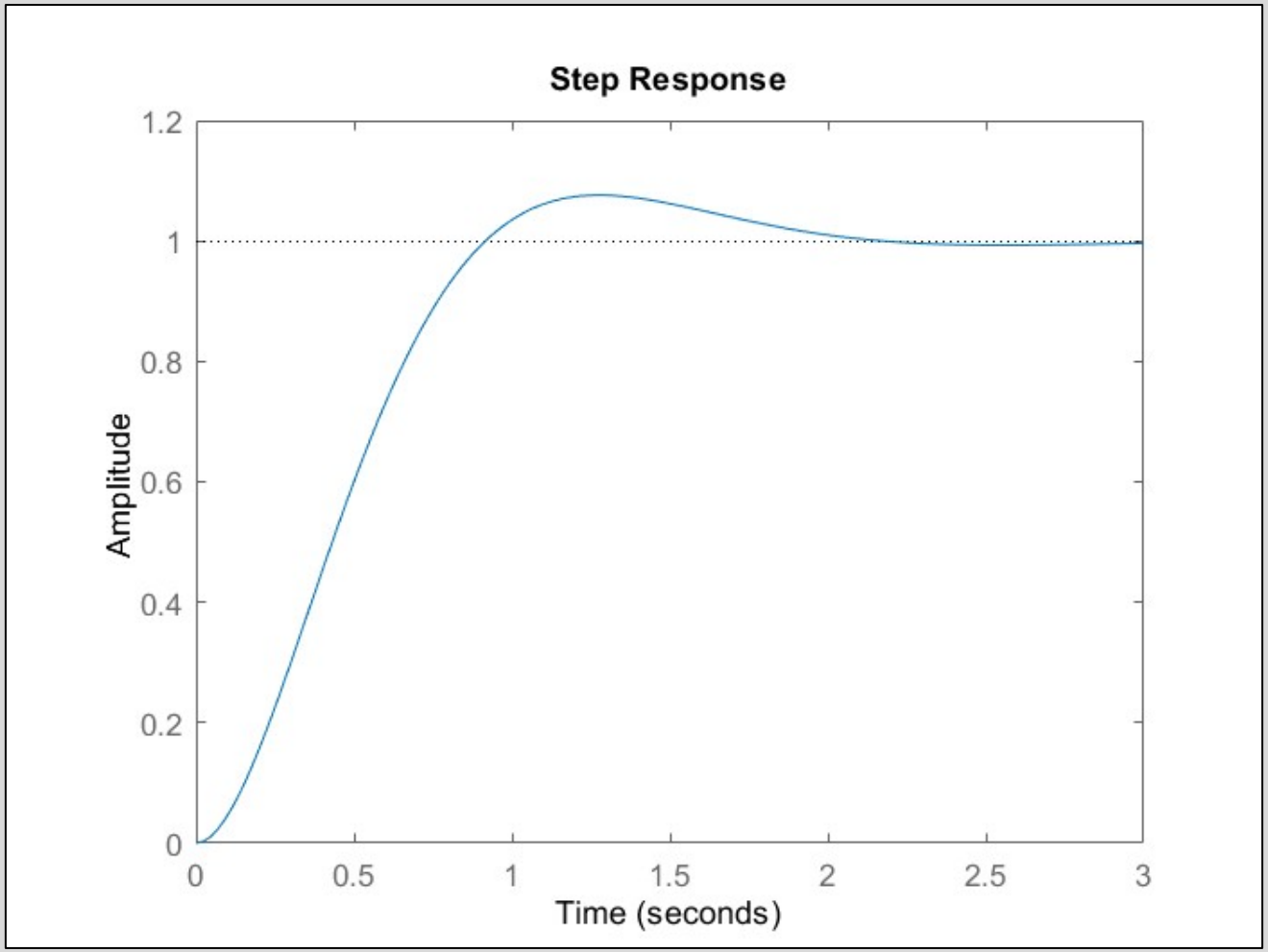
STEP_RESPOND.m

$$R(s) \rightarrow \boxed{\frac{11.1(s+18)}{(s+20)(s^2+4s+10)}} \rightarrow Y(s)$$

```
syms s; s = tf('s');
T = (11.1*(s+18))/((s+20)*(s^2+4*s+10));
T = T/s;  % 步階輸入
[N,D] = tfdata(T);
syms S;
F = poly2sym(N,S)/poly2sym(D,S);
f = ilaplace(F,S,'t'); % 反拉氏轉換
```

STEP_RESPOND.m

```
y = (37*exp(-20*t))/11000 - (5513*exp(-2*t)*(cos(6^(1/2)*t)
    +(48*6^(1/2)*sin(6^(1/2)*t))/149))/5500 % 響應震盪成份
    +999/1000 % 穩態響應
```

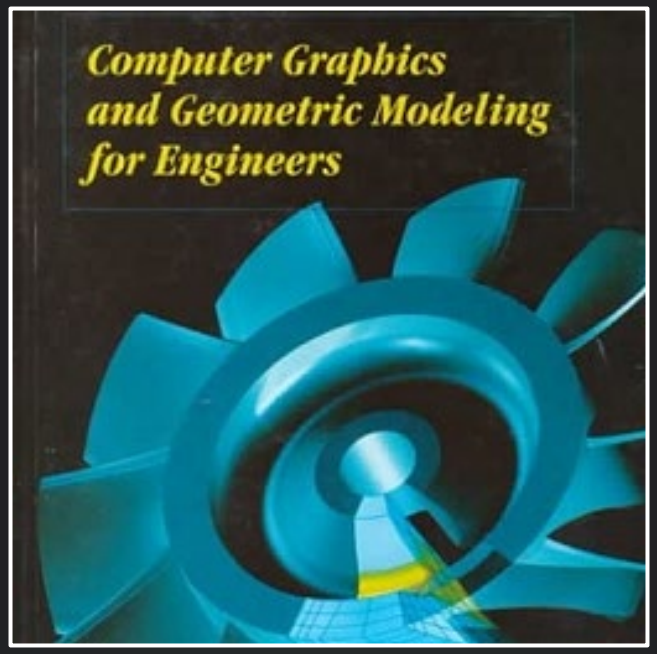


STEP_RESPOND.h

```
float respond = ((37.0f*exp(-20.0f
* t)) / 11000.0f - (5513.0f * exp(-
2.0f * t)*(cos(pow(6.0f, 0.5f)*t) +
(48 * pow(6.0f, 0.5f)*sin(pow(6.0f,
0.5f)*t)) / 149.0f)) / 5500.0f +
999.0f / 1000.0f);
// settling time and steady state
if (abs(respond - 1.0f) < 0.02)
    { return steady_respond; }
else{ return respond; }
```

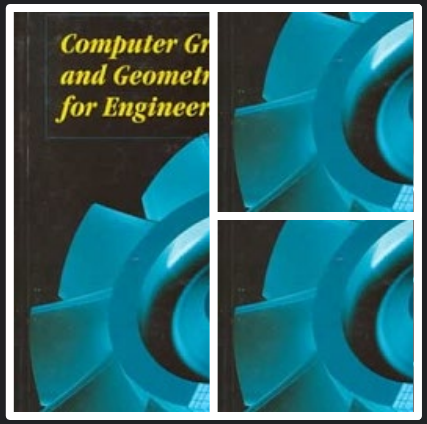
# TEXTURE.h

glTexCoord2f(0,1) glTexCoord2f(1,1)

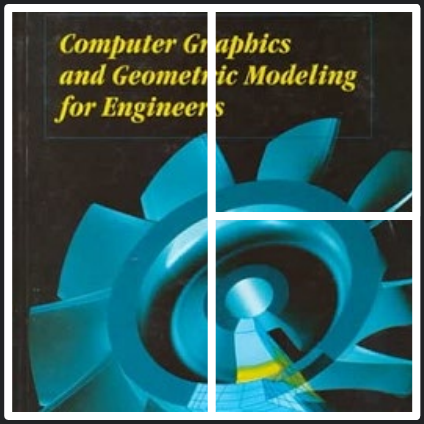



glTexCoord2f(0,0) glTexCoord2f(1,0)

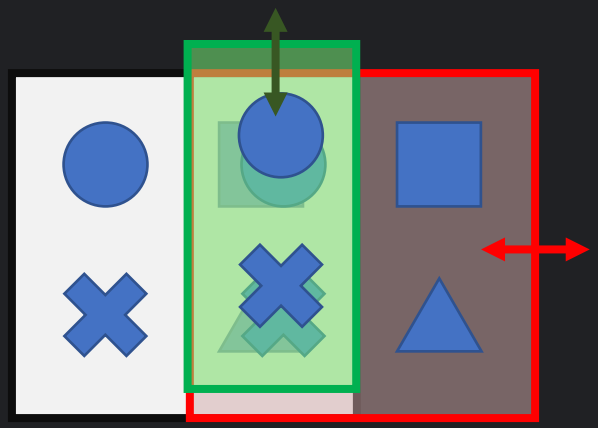
# TEXTURE.h

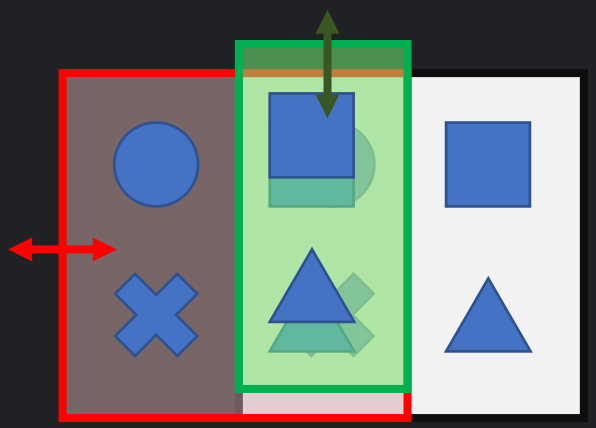

Computer Gr
and Geometr
for Engineer


Computer Graphics
and Geometric Modeling
for Engineers


Computer Graphics
and Geometric Modeling
for Engineers

# TEXTURE.h





World coordinate:

$X_{neg_0}$ $X_{neg_1}$ $X_{pos_0}$ $X_{pos_1}$

$X_{neg_2}$ $X_{pos_2}$

$X_{neg_1}$ $X_{neg_0}$ $X_{pos_1}$ $X_{pos_0}$

$X_{neg_2}$ $X_{pos_2}$

Texture coordinate:

$T_{neg_0}$ $T_{neg_0}$ $T_{pos_0}$ $T_{pos_0}$

$T_{neg_0}$ $T_{pos_2}$

$T_{neg_1}$ $T_{neg_0}$ $T_{pos_1}$ $T_{pos_0}$

$T_{neg_2}$ $T_{pos_2}$

# TEXTURE.h

切掉右邊

$$T_{neg_2} = T_{neg_0}$$

$$T_{pos_2} = \frac{X_{neg_2} - X_{neg_0}}{X_{pos_0} - X_{neg_0}} = T_{neg_2} + \frac{WX1 - WX0}{WX_{initial}}$$

切掉左邊

$$T_{neg_2} = \frac{X_{pos_2} - X_{neg_0}}{X_{pos_0} - X_{neg_0}} = T_{pos_2} - \frac{WX1 - WX0}{WX_{initial}}$$

$$T_{pos_2} = T_{pos_0}$$

TEXTURE.h

```
if (osci_direct) {  // 如果是 x 方向來回
    float center_temp_x = (center_last[0] + center[0]) / 2;   // C2
    wx2_ = abs(center_temp_x - center[0]);                    // wx2'
    if (center[0] > center_last[0]) {                         // 如果切掉的是 + 那邊
        center_fall[0] = center_temp_x + wx; b = TRUE;        // C2'
        face_texture_falling[0][1] = face_texture_falling[1][1] =
        face_texture[2][1] = face_texture[3][1] = face_texture[0][1] + (wx - wx2_) / STACK_CUBOID_INITIAL_WIDTH; /* z+ */
    }
    else {                                                    // 切掉的是 - 那邊
        center_fall[0] = center_temp_x - wx; b = FALSE;
        face_texture_falling[2][1] = face_texture_falling[3][1] =
        face_texture[0][1] = face_texture[1][1] = face_texture[2][1] - (wx - wx2_) / STACK_CUBOID_INITIAL_WIDTH; /* z- */
    }
    wx = wx - wx2_;                                           // wx2
    center_last[0] = center[0] = center_temp_x;
}
else {              // 如果是 z 方向來回
    float center_temp_z = (center_last[2] + center[2]) / 2;   // C2
    wz2_ = abs(center_temp_z - center[2]);                    // wz2'
    if (center[2] > center_last[2]) {                         // 切掉的是 + 那邊
        center_fall[2] = center_temp_z + wz; b = TRUE;        // C2'
        face_texture_falling[0][0] = face_texture_falling[3][0] =
        face_texture[1][0] = face_texture[2][0] = face_texture[0][0] + (wz - wz2_) / STACK_CUBOID_INITIAL_WIDTH; /* x+ */
    }
    else {                                                    // 切掉的是 - 那邊
        center_fall[2] = center_temp_z - wz; b = FALSE;
        face_texture_falling[1][0] = face_texture_falling[2][0] =
        face_texture[0][0] = face_texture[3][0] = face_texture[1][0] - (wz - wz2_) / STACK_CUBOID_INITIAL_WIDTH; /* x- */
    }
    wz = wz - wz2_;                                           // wz2
    center_last[2] = center[2] = center_temp_z;
}
```

CUBOID.h

```
/* 方塊ver.2 */
class CUBOID {
public:
    float vertex[8][3];
    // 貼圖座標
    float texcoord[4][2];
    GLubyte color[3];
};
```

UI.h

```
//==========   操縱方式   ==========
//空白 遊玩堆疊,結束->起始,選擇->起始
// e 強制結束(會儲存紀錄)
//場景: a 起始, f 選擇1, g 選擇2
//→↑起始->選擇1->選擇2->起始
//←↓起始->選擇2->選擇1->起始
// i 模式:純色   // o 模式:骨架
// p 貼圖:隨機   // t 貼圖:校徽
// l 貼圖:系徽   // r 貼圖:實驗室徽
// q 儲存紀錄    // w 載入紀錄
// z 強制重新顏色起點
// x 強制顏色再隨機一次
// h 顯示/不顯示 掉落方塊
// j 顯示/不顯示 掉落方塊旋轉
// v 顯示/不顯示 緩降動畫
// b 顯示/不顯示 按鈕
// m 顯示/不顯示 二階系統彈跳
// y 全顯示
```

```
//========== 貼圖解鎖條件 ==========
// 每疊10層  credit + 1
// 永久使用: PURE COLOR
//           SKELETON
//           RAMDOM
//           STATIC
//           DYNAMIC
// credit > 20   Thermodynamics
// credit > 25   Mechanics of materials
// credit > 30   Munson's fluid mechanics
// credit > 35   Advanced engineering mathmatics
// credit > 40   Complex analysis:
//               A first course with Application
// max > 20      Modern Mechanisms
// max > 25      Electrical Engineering
// max > 30      Electronic Devices
// max > 35      Manufacturing
//               Engineering and Techology
// max > 40      Fundaments of
//               Machine Component Design
// max > 45      Control Systems Engineering
// max > 50      Ivor Hortom's Beginning ANSI C++
// max > 55      Computer Graphics
//               and Geometric modeling
//               for engineers
```

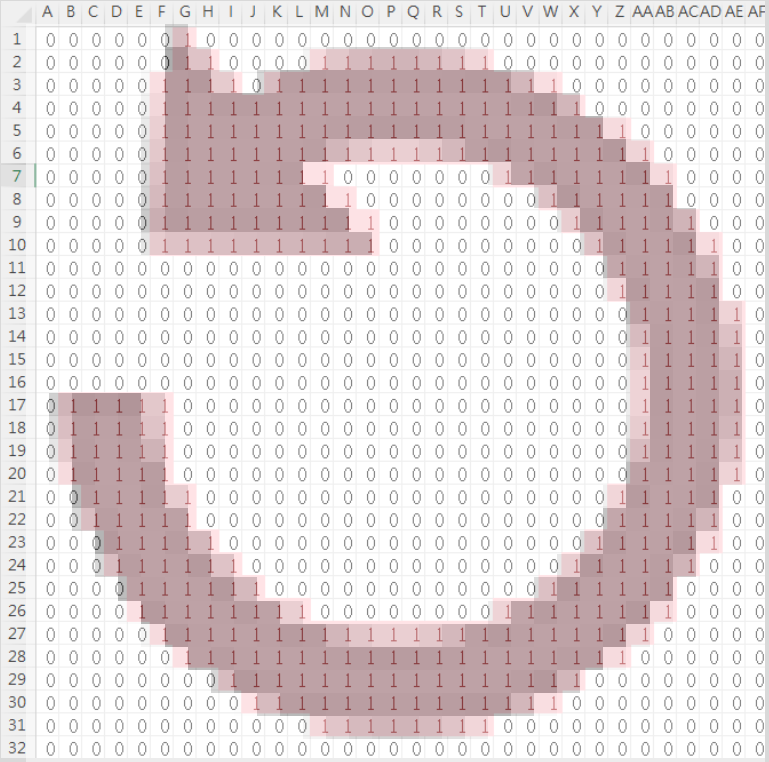
Bitmap_generation.m

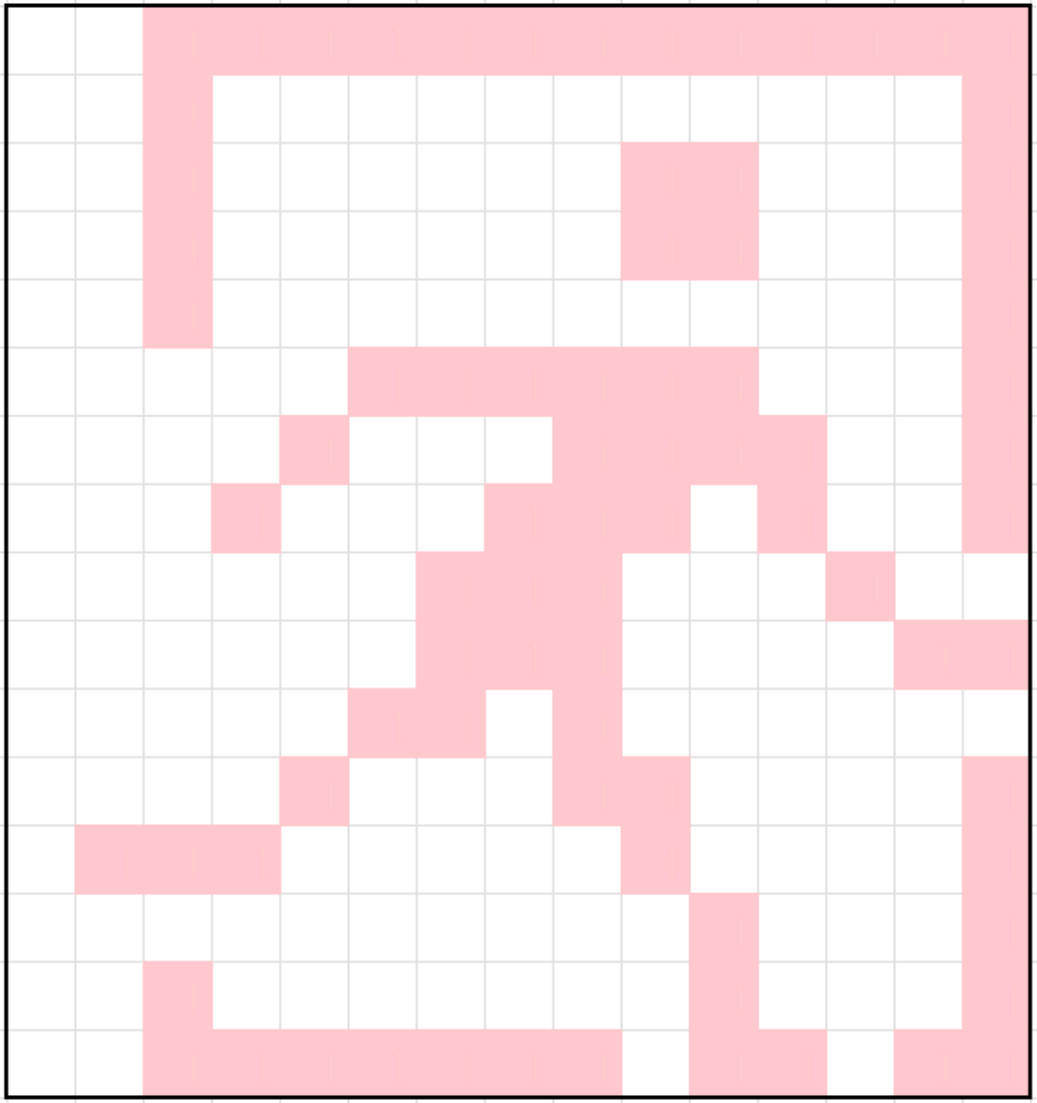
```
im = imread('redo.png','png');
im(im~=255) = 1; % 影像二極化
im(im==255) = 0;
```

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | AA | AB | AC | AD | AE | AF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 17 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 18 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 19 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 20 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 21 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 22 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 23 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 24 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 25 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 27 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 28 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bitmap_generation.m

```
im = imread('redo.png','png');
im(im~=255) = 1; % 影像二極化
im(im==255) = 0;
```





| | |
|---|---|
| 0x1F, 0xFF | 0x1F, 0xDB |
| 0x10, 0x1 | 0x10, 0x11 |
| 0x10, 0x31 | 0x0, 0x11 |
| 0x10, 0x31 | 0x38, 0x21 |
| 0x10, 0x1 | 0x4, 0x61 |
| 0x3, 0xF1 | 0x3, 0x40 |
| 0x4, 0x79 | 0x1, 0xC3 |
| 0x8, 0xE9 | 0x1, 0xC4 |
| 0x1, 0xC4 | 0x8, 0xE9 |
| 0x1, 0xC3 | 0x4, 0x79 |
| 0x3, 0x40 | 0x3, 0xF1 |
| 0x4, 0x61 | 0x10, 0x1 |
| 0x38, 0x21 | 0x10, 0x31 |
| 0x0, 0x11 | 0x10, 0x31 |
| 0x10, 0x11 | 0x10, 0x1 |
| 0x1F, 0xDB | 0x1F, 0xFF |

```
="0x"&BIN2HEX(CONCAT(IF(ISBLANK(A2:H2),0,1)))&", 0x"&BIN2HEX(CONCAT(IF(ISBLANK(I2:P2),(
=INDIRECT("Q"&(QUOTIENT(ROW()-2,16)+1)*16-MOD(ROW()-2,16)+1)
```

16x16

# UI.h

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | AA | AB | AC | AD | AE | AF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 25 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 26 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 29 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 30 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 31 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 32 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

=INDIRECT("16x16!"&ADDRESS($AH$2+CEILING(ROW()/2,1)-1,$AI$2+CEILING(COLUMN()/2,1)-1,4,

Bitmap.h

```
static GLubyte bitmap_exit[] = {
    0x3,  0xFF, 0xF3, 0xCF, 0x3,  0xFF, 0xF3, 0xCF,
    0x3,  0x0,  0x3,  0x3,  0x3,  0x0,  0x3,  0x3,
    0x0,  0x0,  0x3,  0x3,  0x0,  0x0,  0x3,  0x3,
    0xF,  0xC0, 0xC,  0x3,  0xF,  0xC0, 0xC,  0x3,
    0x0,  0x30, 0x3C, 0x3,  0x0,  0x30, 0x3C, 0x3,
    0x0,  0xF,  0x30, 0x0,  0x0,  0xF,  0x30, 0x0,
    0x0,  0x3,  0xF0, 0xF,  0x0,  0x3,  0xF0, 0xF,
    0x0,  0x3,  0xF0, 0x30, 0x0,  0x3,  0xF0, 0x30,
    0x0,  0xC0, 0xFC, 0xC3, 0x0,  0xC0, 0xFC, 0xC3,
    0x0,  0x30, 0x3F, 0xC3, 0x0,  0x30, 0x3F, 0xC3,
    0x0,  0xF,  0xFF, 0x3,  0x0,  0xF,  0xFF, 0x3,
    0x3,  0x0,  0x0,  0x3,  0x3,  0x0,  0x0,  0x3,
    0x3,  0x0,  0xF,  0x3,  0x3,  0x0,  0xF,  0x3,
    0x3,  0x0,  0xF,  0x3,  0x3,  0x0,  0xF,  0x3,
    0x3,  0x0,  0x0,  0x3,  0x3,  0x0,  0x0,  0x3,
    0x3,  0xFF, 0xFF, 0xFF, 0x3,  0xFF, 0xFF, 0xFF
};
glBitmap(32, 32, 16, 16, 0, 0, bitmap_exit);
```

# password.h

```
switch (password) //  password = ncku
{
    case 0: if (key == ‘n’ || key == ‘N’) { password++; break; }
    case 1: if (key == ‘c’ || key == ‘C’) { password++; break; }
    case 2: if (key == ‘k’ || key == ‘K’) { password++; break; }
    case 3:
        if (key == 'u' || key == 'U') {
            password++;
            max_score = INT_MAX;
            credit = 144; //  好熟悉的數字啊...難道是...畢業學分...
            calculate_record();
        }
        break;
    case 4:  break;
    default: password = 0; //  重置
}
```

# Branchless_programming.h

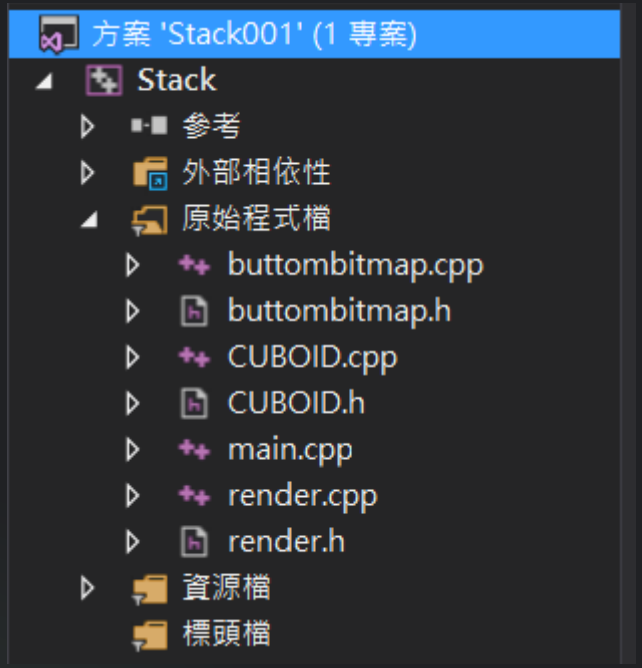
```
void function(){
    if(A)
        B = B + C;
    else
        B = B + D;
}

void function(){
    B = B + A * C + (!A) * D;
}
```

Branchless_programming.h

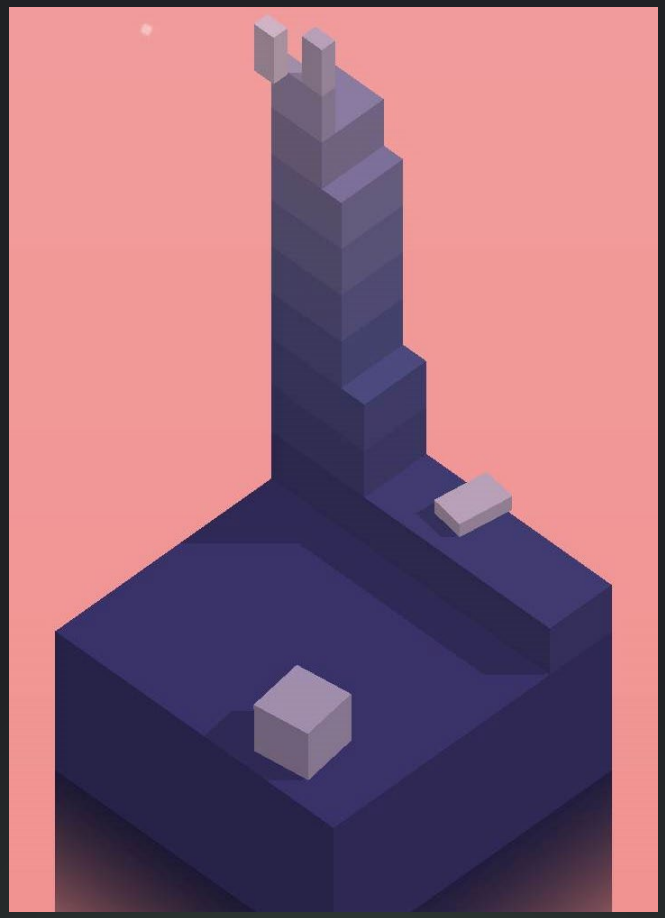
```
return
(abs(respond - 1.0f) < 0.02) *
((37.0f*exp(-20.0f * t)) / 11000.0f
- (5513.0f * exp(-2.0f *
t)*(cos(pow(6.0f, 0.5f)*t) + (48 *
pow(6.0f, 0.5f)*sin(pow(6.0f,
0.5f)*t)) / 149.0f)) / 5500.0f +
999.0f / 1000.0f)
+
(!(abs(respond - 1.0f) < 0.02)) *
steady_respond;
```

# MINUS_POINT.H

```
1359        ADVENGMATH = records[STACK_TEXTURE_ADVENG
1360        MECHANISM = records[STACK_TEXTURE_MECHANI
1361        EE = records[STACK_TEXTURE_EE];
1362        ED = records[STACK_TEXTURE_ED];
1363        MANUFACTURE = records[STACK_TEXTURE_MANUF
1364        DESIGN = records[STACK_TEXTURE_DESIGN];
1365        CONTROL = records[STACK_TEXTURE_CONTROL];
1366        CPP = records[STACK_TEXTURE_CPP];
1367        CG = records[STACK_TEXTURE_CG];
1368        calculate_record();
1369        return;
1370    }
1371
```



# FUTURE_WORK.H



Thanks for listening.