# 打磚塊遊戲（2D 繪圖）

**班級 ： 資工 4 甲**

**學號 ： 406261688**

**姓名 ： 陳君平**

**題目 ：**困難到翻過去的打磚塊

**程式架構 ：**

程式初始化:

glutInit(&argc, argv);

glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);

glutInitWindowSize(600,600); - 設定視窗大小

glutCreateWindow("期中作業"); 檔案上方標題

glutKeyboardFunc(ESC); 按鈕控制

控制方面有”ESC”,”1”,”2”,”3”,”4”,”space”

ESC – 退出遊戲

1 – 玩家變長 2 – 玩家變短 3-球變大 4-球變小

Space - 開始遊戲、準備遊戲、失敗重新開始

glutSpecialFunc(Arrowkeys); 方向鍵控制

方向鍵控制玩家

glutDisplayFunc(RenderScene); 把視窗中的圖形打印出來

1.印出訊息

2.印出玩家的橫槓

3.印出磚塊

4.印出球

5.印出分數

6.區分 1,2 關卡

glutReshapeFunc(ChangeSize); 當視窗改變時觸發

glutTimerFunc(10, TimerFunction, 1); 可讓視窗有動畫效果

分為 1,2 關

每關有各個關卡不同的設定

SetupRC(); 清空

glutMainLoop();重複執行 main

**討論 (遊戲詳細介紹):**

本次作品共分為兩關

第一關 ：玩家進入遊後馬上進入第一關，只要按下 **space** 即可開始，玩家會控制四根棍子，並有技巧地把磚塊打完，第一關過關可得 **120** 分，共 **12** 磚塊，玩家的棍子相鄰控制方向相反(例如玩家按右鍵)，在視窗下方往右跑，上方又跑，左與右的棍子(立著)向左，失敗條件 ：只要觸碰到任一邊遊戲結束，遊戲困難度有點難，因此設計上多了一些小小工具，分別

為 **1~4** 數字鍵，兩關皆可使用，數字 **1** 可讓玩家變長到 **windowWidth** 一半，數字 **2** 最短變成 **70**，數字 **3** 可讓球變成最大半徑 **10**，數字 **4** 變小至 **5**，當失敗時出現 **GAME OVER**，過關時出現 **NEXT LEVEL**，當失敗按 **space**，先進入第一關準備畫面，再按一次進入第一關再次挑戰，若成功按 **space**，進入第二關準備畫面，再按一次進入第二關再次挑戰。

第二關 ：第二關比第一關少 **3** 個橫槓，只有最下方，只有若到下方才算輸，其他三方牆壁接反彈，畫面上共有 **8** 個磚塊，每個磚塊都有特別部分首先 **1,2,5,6** 磚塊的位置使用亂數配置隨機位置，因此四個磚塊位置不固定，在球撞擊到觸發效果

**1,2 –** 當撞擊時，由亂數判斷球是依照水平或垂直進行反彈

**5,6 –** 也是藉由亂數取決，但不同 **1,2** 磚塊在於在取決完水平或垂直反彈時，可能使球速增快

**3 –** 亂數取決球變大或是水平反射

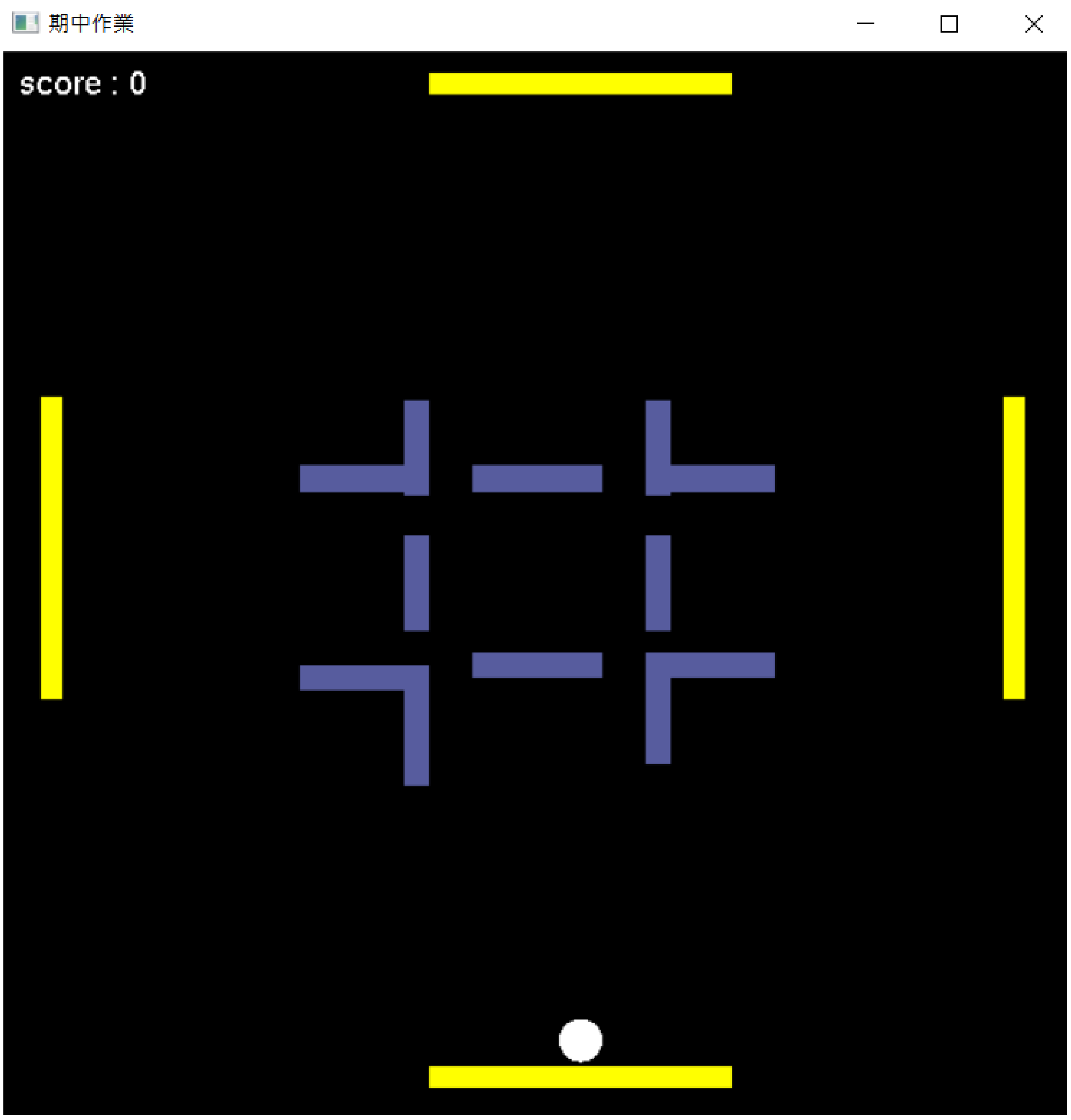
**4 -** 亂數取決球變小或是垂直反射

**7–** 亂數取決玩家橫槓變長或是水平反射

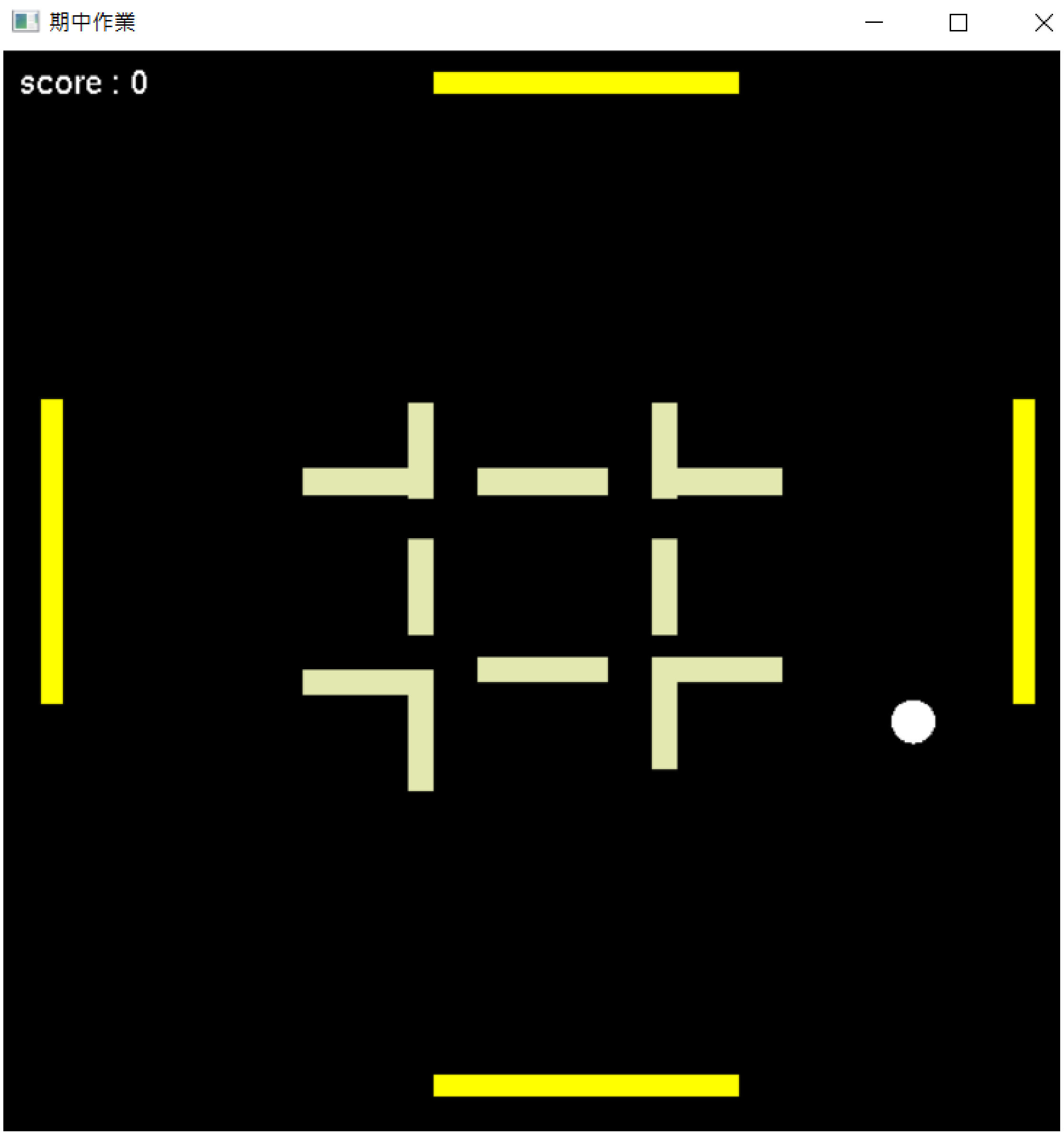
**8 -** 亂數取決玩家橫槓變短或是垂直反射

過關條件 ： 必須在面臨 **1,2,5,6** 磚塊干擾下打掉 **3,4,7,8** 磚塊，每個磚塊 **10** 分，與第一關分數累加，失敗同第一關，可再次挑戰，若通關，畫面顯示 **WIN~~~**

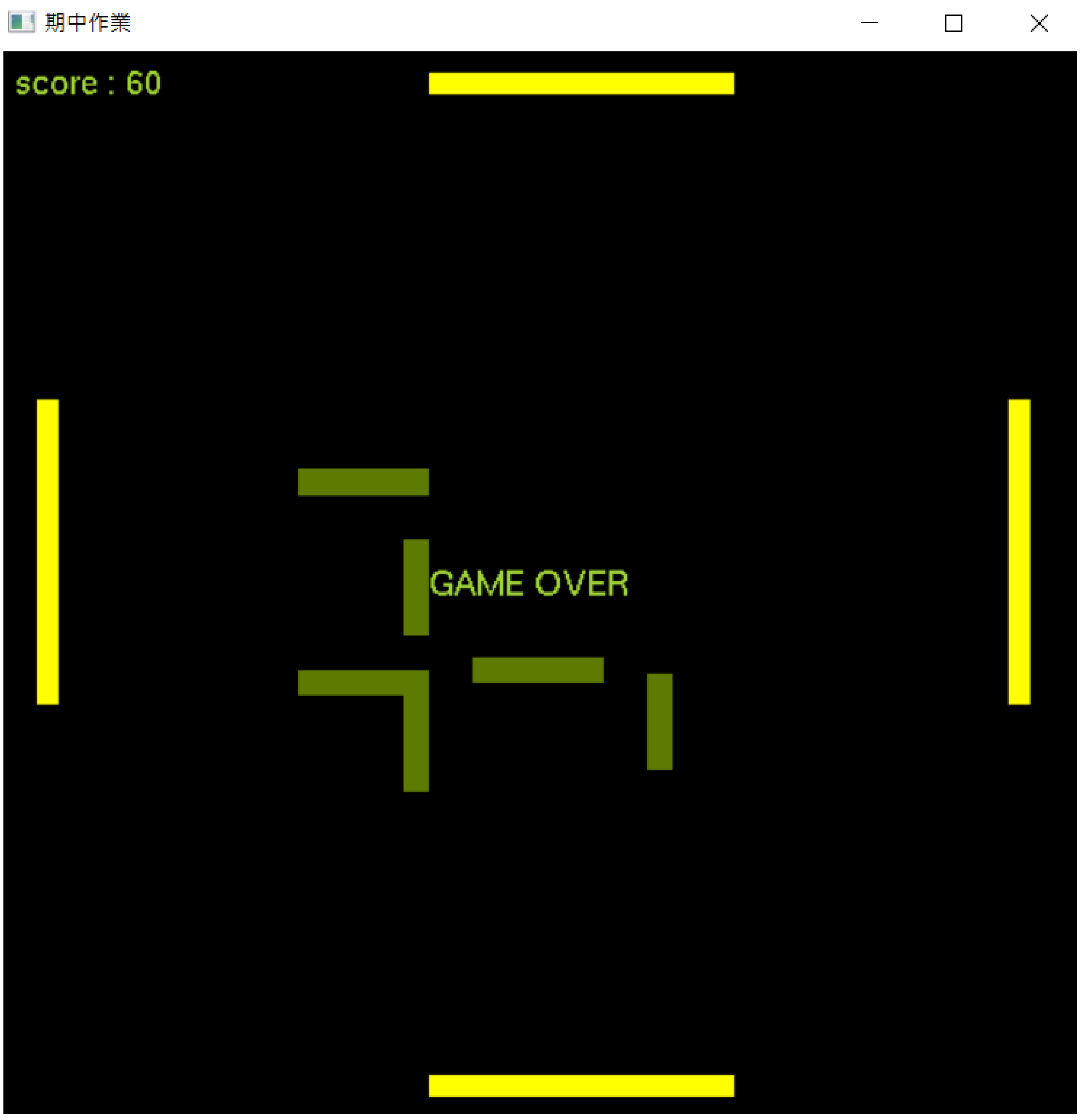
執行畫面 ：



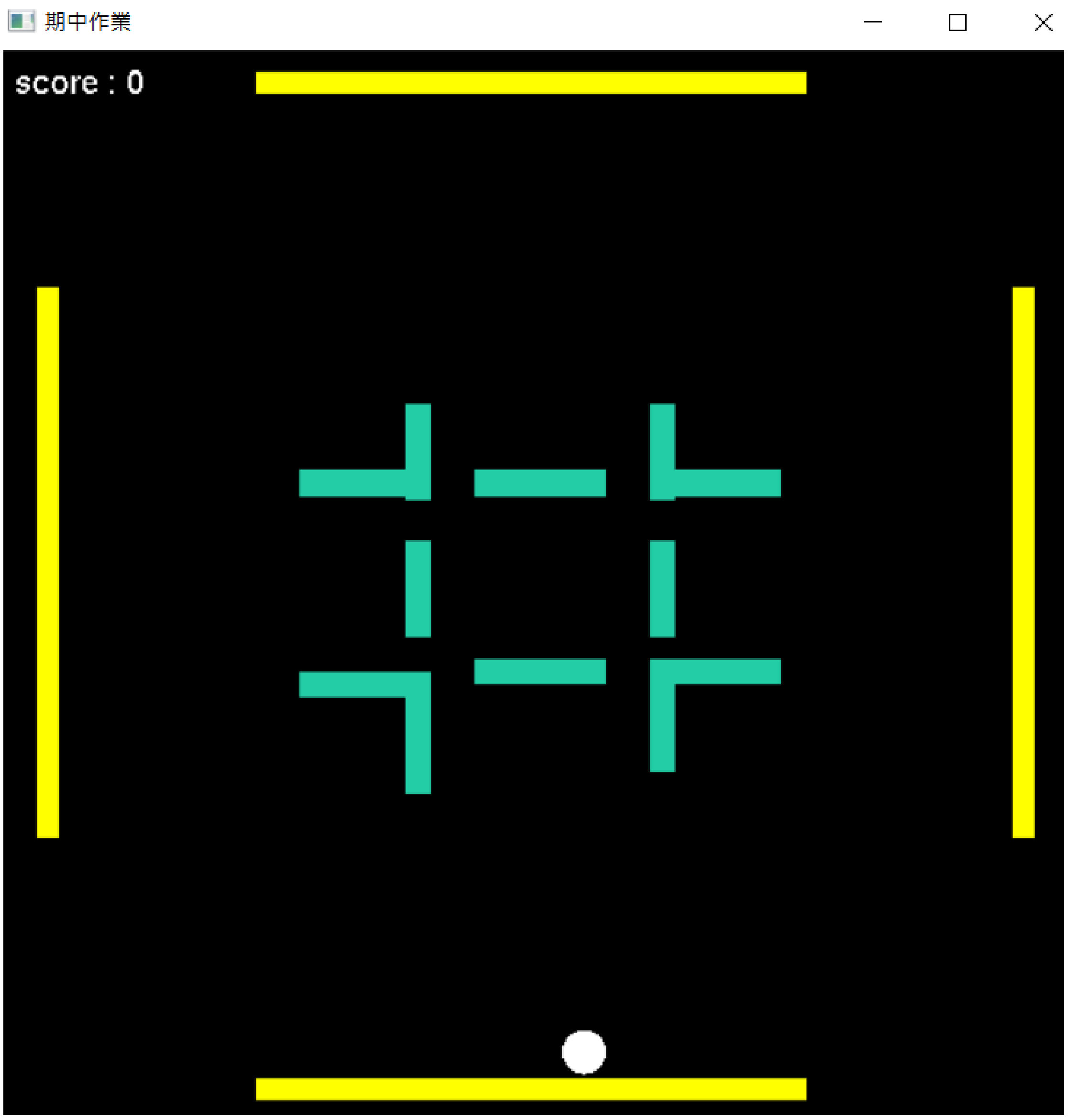
進入遊戲第一關



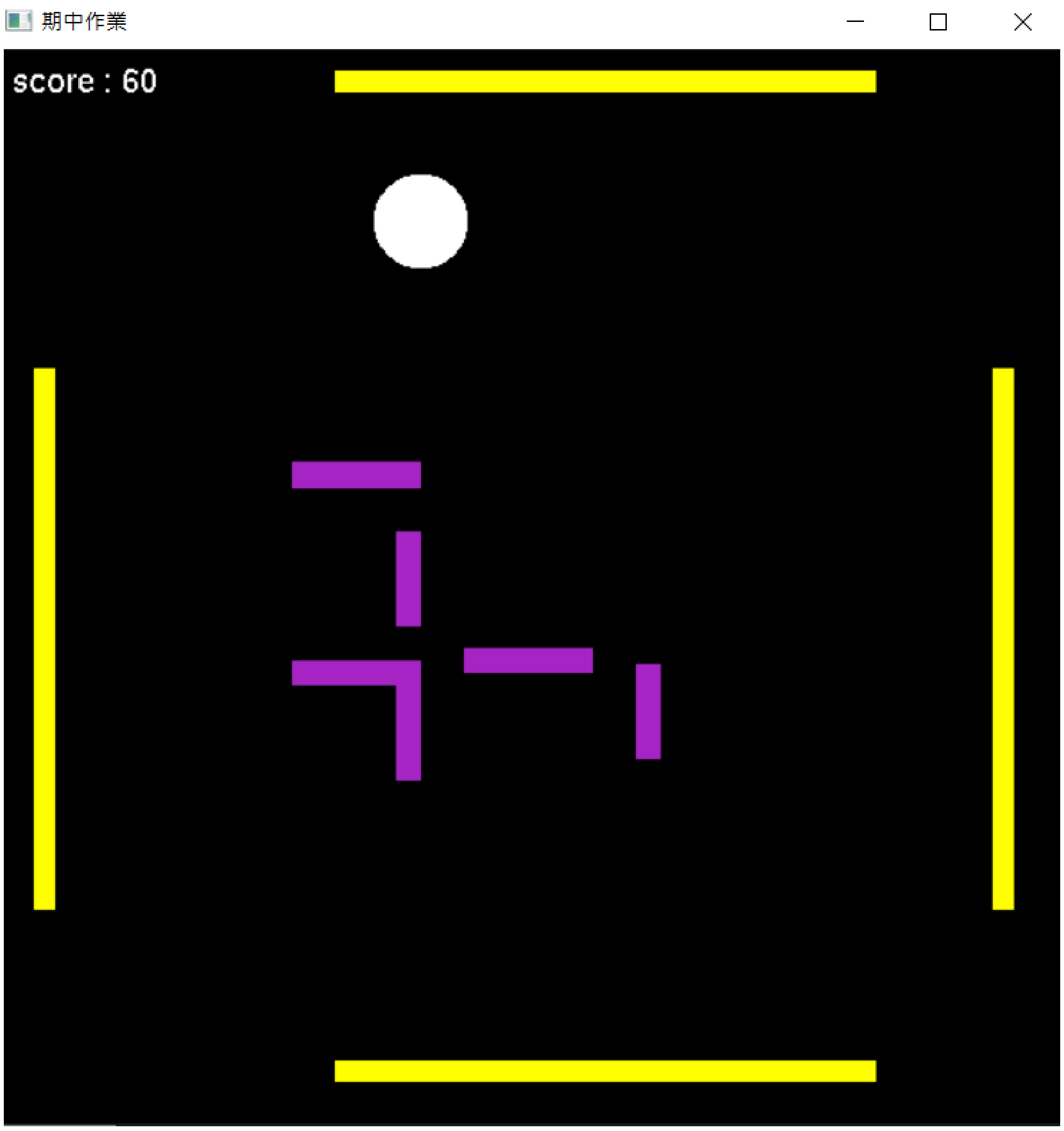
空白鍵射出球



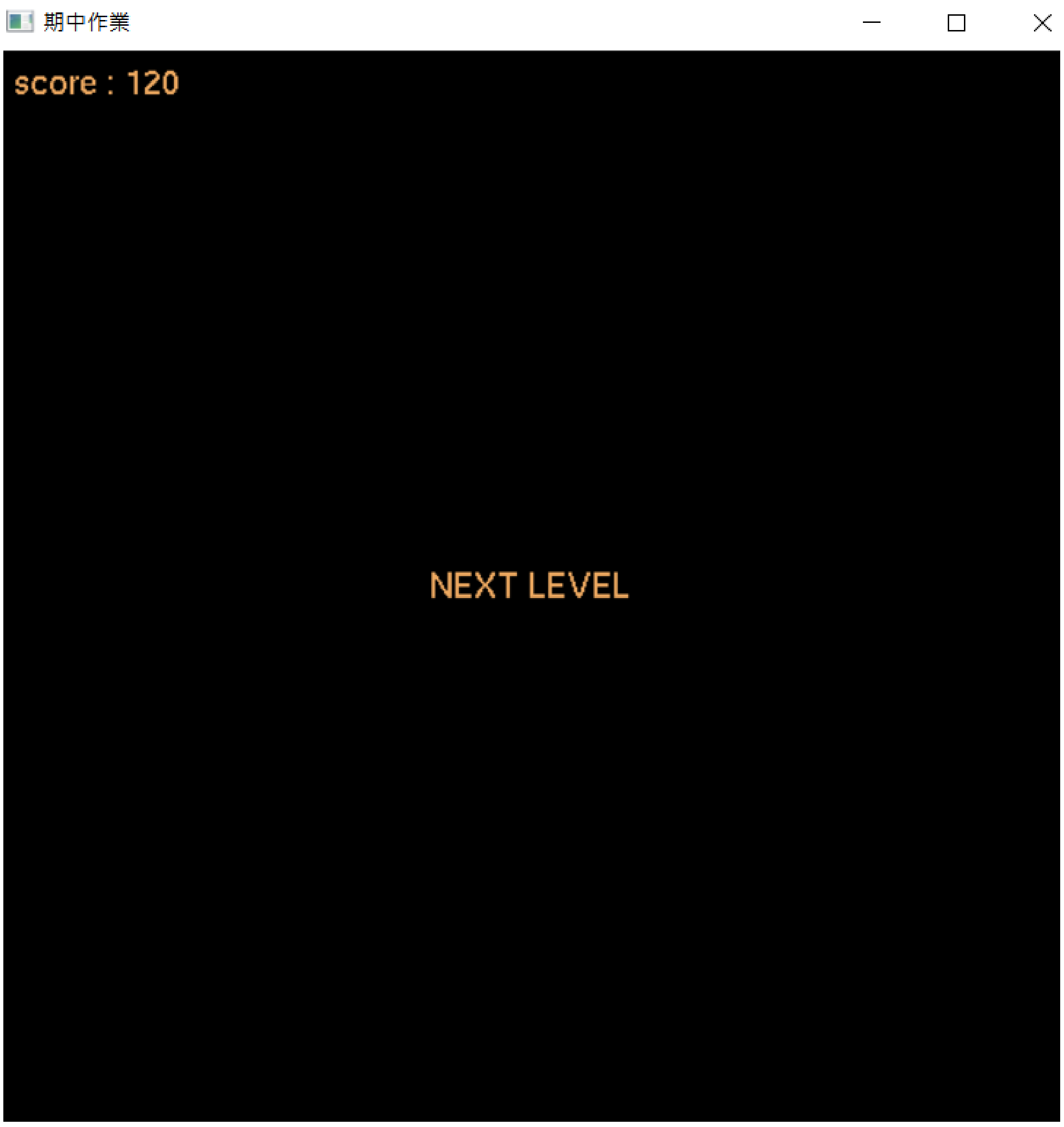
**GAME OVER**



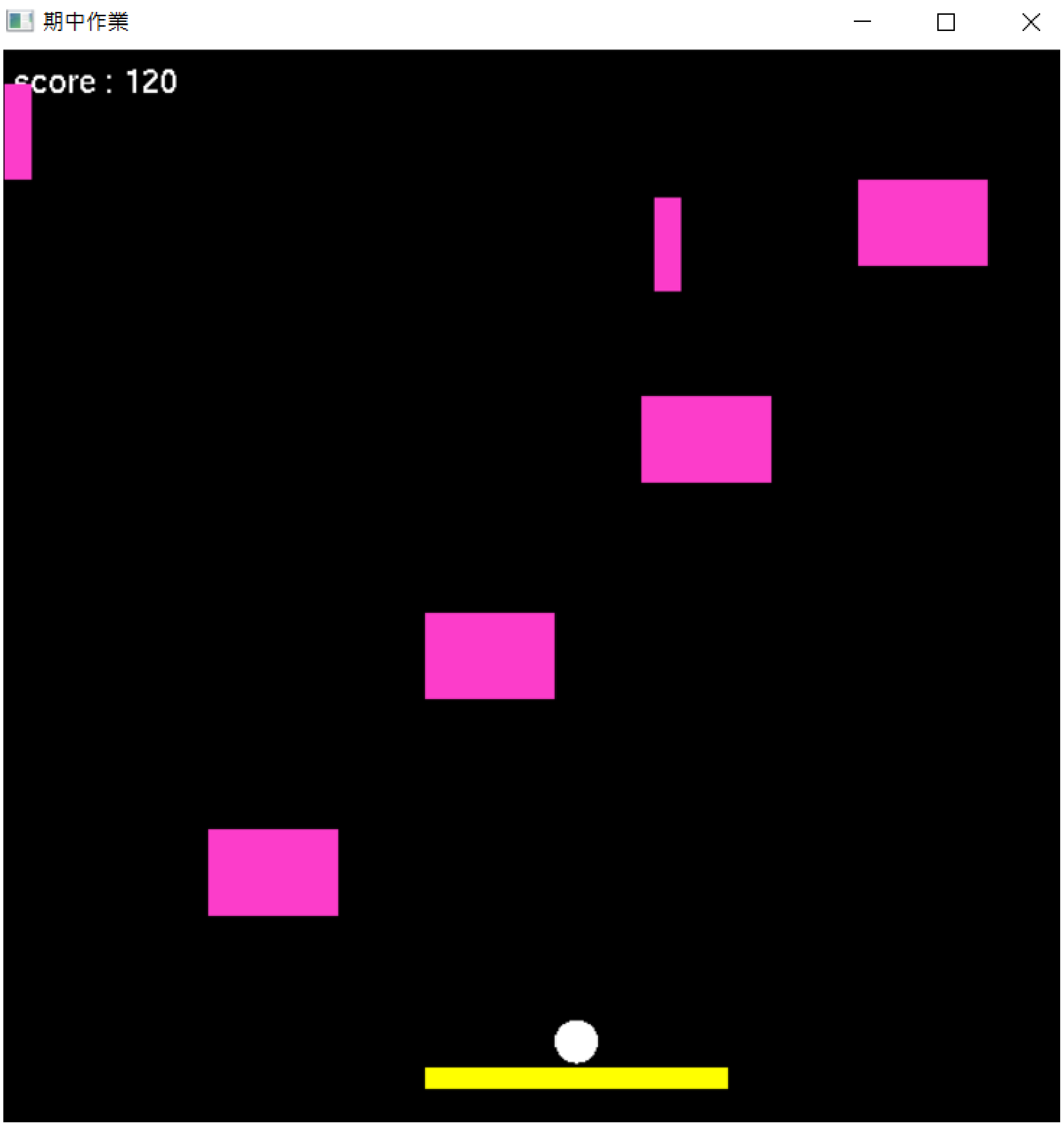
開啟 1or2 按鈕控制橫槓長度



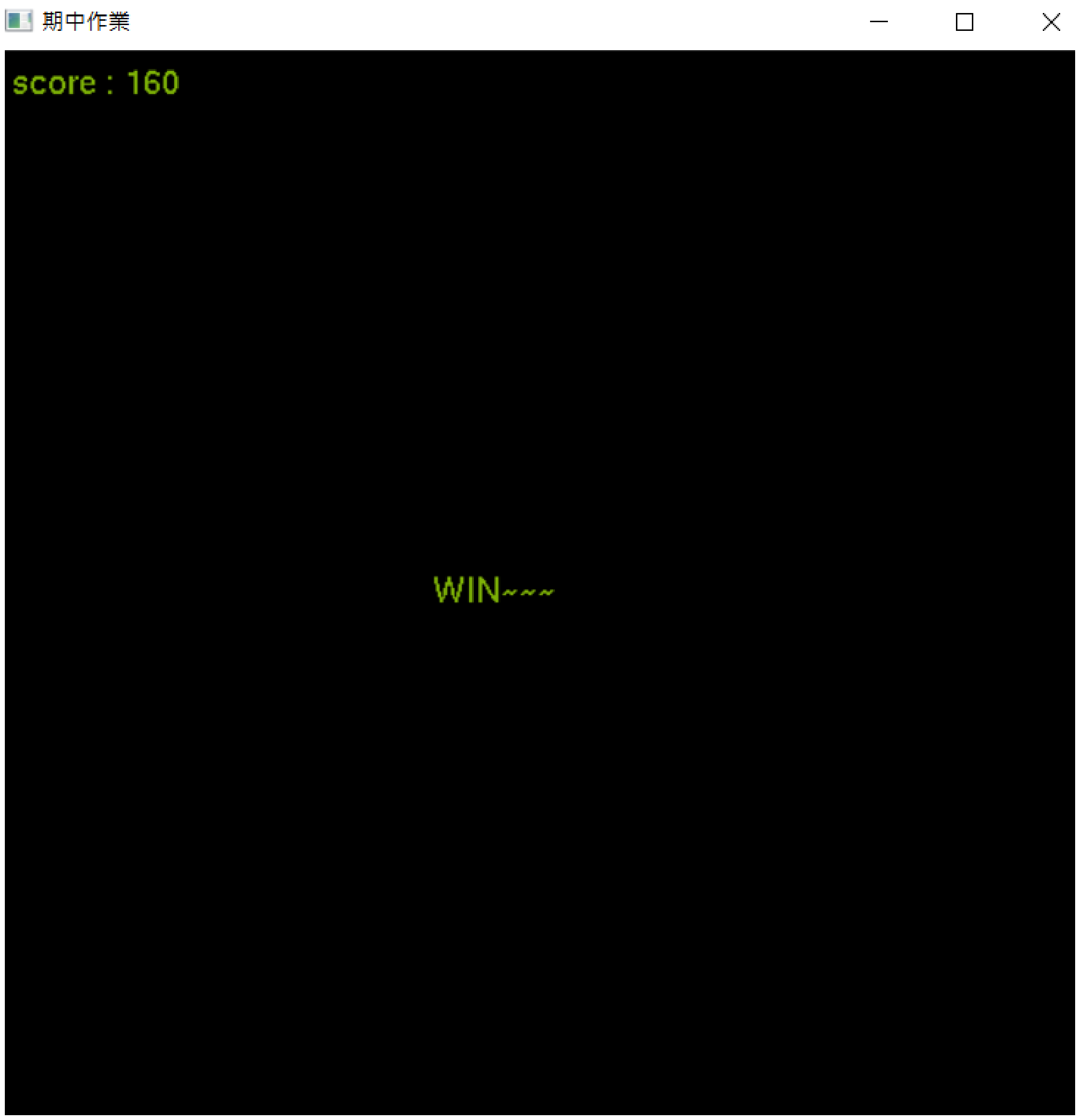
開啟 3or4 按鈕控制球大小



挑戰成功，進下一關



第二關遊戲畫面，因為方塊動很快，因此圖片無法展現效果



全破關

程式碼 ：

```
#define GLUT_DISABLE_ATEXIT_HACK

#include <windows.h>

#include <GL/glut.h>

#define PI 3.14159265

#include <math.h>
```

```
#include <stdlib.h>

#include<ctime>

#include <string>

#include "stdio.h"


int state = 0;

int level = 1;

int xstep = 1.0f;

int ystep = 1.0f;


int windowWidth;

int windowHeight;


int players_w = 70;

int players_h = 5;

int circle = 30;

int r = 5;


int players1_x = 100;
int players1_y = 10;

int players2_x = 100;

int players2_y = 240;


int players3_x = 10;

int players3_y = 100;

int players4_x = 233;

int players4_y = 100;


int ball_x = players1_x + (players_w/2);

int ball_y = players1_y + players_h + r + 1;


int block_count = 0;

int score = 0;


int block_x_1 = 70;

int block_y_1 = 148;

int block_x_2 = 110;

int block_y_2 = 148;
int block_x_3 = 150;
int block_y_3 = 148;

int block_x_4 = 70;
int block_y_4 = 102;
int block_x_5 = 110;
int block_y_5 = 105;
int block_x_6 = 150;
int block_y_6 = 105;

int block_x_7 = 94;
int block_y_7 = 80;
int block_x_8 = 94;
int block_y_8 = 116;
int block_x_9 = 94;
int block_y_9 = 147;

int block_x_10 = 150;
int block_y_10 = 85;
int block_x_11 = 150;

int block_y_11 = 116;

int block_x_12 = 150;

int block_y_12 = 147;







void int2str(int i, char *s) {

    sprintf(s,"score : %d",i);

}


void ESC(unsigned char key, int x, int y)

{

        if(key == 27) exit (0);

        if(key == 49 && players_w <= windowWidth/2)

players_w++;

        if(key == 50 && players_w >= 70) players_w--;

        if(key == 51 && r <= 10) r++;
        if(key == 52 && r >= 5) r--;
        if(key == 32 && state == 0 && (level == 1 || level
== 2)) state = 1;
        if(key == 32 && state == 0 && level ==0 )
      {
          level++;
          state = 0;
          level = 1;
          xstep = 1.0f;
          ystep = 1.0f;

          players_w = 70;
          players_h = 5;
          circle = 30;
          r = 5;

          players1_x = 100;
          players1_y = 10;
          players2_x = 100;
```

```
players2_y = 240;

players3_x = 10;

players3_y = 100;

players4_x = 233;

players4_y = 100;

ball_x = players1_x + (players_w/2);

ball_y = players1_y + players_h + r + 1;

block_count = 0;

score = 0;

block_x_1 = 70;

block_y_1 = 148;

block_x_2 = 110;

block_y_2 = 148;

block_x_3 = 150;

block_y_3 = 148;

block_x_4 = 70;

block_y_4 = 102;

block_x_5 = 110;

block_y_5 = 105;
```

```
    block_x_6 = 150;
    block_y_6 = 105;
    block_x_7 = 94;
    block_y_7 = 80;
    block_x_8 = 94;
    block_y_8 = 116;
    block_x_9 = 94;
    block_y_9 = 147;
    block_x_10 = 150;
    block_y_10 = 85;
    block_x_11 = 150;
    block_y_11 = 116;
    block_x_12 = 150;
    block_y_12 = 147;
}
if(key == 32 && state == 2 && level == 2)
{
    state = 0;
    xstep = 1.0f;
```

```
ystep = 1.0f;

players1_x = 100;

players1_y = 10;


score = 120;

players_w = 70;

players_h = 5;

circle = 30;

r = 5;


ball_x = players1_x + (players_w/2);

ball_y = players1_y + players_h + r + 1;


block_count = 13;

block_x_1 = 70;

block_y_1 = 148;

block_x_2 = 110;

block_y_2 = 148;
```

```
          block_x_3 = 50;

          block_y_3 = 50;

          block_x_4 = 100;

          block_y_4 = 100;


          block_x_5 = 110;

          block_y_5 = 105;

          block_x_6 = 150;

          block_y_6 = 105;


          block_x_7 = 150;

          block_y_7 = 150;

          block_x_8 = 200;

          block_y_8 = 200;

          state = 0 ;

     }

}


void Arrowkeys(int key, int x, int y)
```

```
{
        if(key == GLUT_KEY_RIGHT)
      {
          if(players1_x < windowWidth - players_w)
                players1_x = players1_x + 10;
          if(players2_x < windowWidth - players_w)
                players2_x = players2_x + 10;
          if(players3_y > 0)
                players3_y = players3_y - 10;
          if(players4_y > 0)
                players4_y = players4_y - 10;
      }
        if(key == GLUT_KEY_LEFT)
        {
          if(players1_x > 0)
                players1_x = players1_x - 10;
          if(players2_x > 0)
                players2_x = players2_x - 10;
          if(players3_y < windowHeight - players_w)
```

```
                players3_y = players3_y + 10;
            if(players4_y < windowHeight - players_w)
                players4_y = players4_y + 10;
    }
}

void RenderBitmapString(float x, float y, void
*font,char *string)
{
    char *c;
    glRasterPos2f(x, y);
    for (c=string; *c != '\0'; c++)
    {
       glutBitmapCharacter(font, *c);
    }
}

void RenderScene(void)
{
    glClearColor(0.0, 0.0, 0.0, 1.0);
    glClear(GL_COLOR_BUFFER_BIT);

    if(block_count == 17)
    {
        level=3;
        state = 2;
        glColor3ub(rand()%256, rand()%256,
rand()%256);
            RenderBitmapString(100,windowHeight/2,
GLUT_BITMAP_HELVETICA_18 , "WIN~~~");
    }
    if(block_count == 12)
    {
        level=2;
        state = 2;
        glColor3ub(rand()%256, rand()%256,
rand()%256);
            RenderBitmapString(100,windowHeight/2,
```

```
GLUT_BITMAP_HELVETICA_18 , "NEXT LEVEL");
    }
    if(level ==2 && ball_y <= 0 - r)
    {
        state = 2;
        glColor3ub(rand()%256, rand()%256,
rand()%256);
            RenderBitmapString(100,windowHeight/2,
GLUT_BITMAP_HELVETICA_18 , "GAME OVER");
    }
    if((level == 0 || level == 1)&&(ball_y <= 0 - r ||
ball_y >= windowHeight+r || ball_x <=0-r ||
ball_x>=windowWidth+r))
      {
        level = 0;
            state = 0;
            glColor3ub(rand()%256, rand()%256,
rand()%256);
            RenderBitmapString(100,windowHeight/2,
GLUT_BITMAP_HELVETICA_18 , "GAME OVER");
      }


      glClearColor(0.0, 0.0, 0.0, 1.0);
    char S[64];
    int2str(score, S);
    puts(S);
      RenderBitmapString(5,windowHeight-10,
GLUT_BITMAP_HELVETICA_18 , S);
      if(level == 1 || (state == 0 && level ==0))
      {
        glColor3f(1.0f, 1.0f, 0.0f);
          glRectf(players1_x, players1_y,
players1_x+players_w, players1_y+players_h);
          glRectf(players2_x, players2_y,
players2_x+players_w, players2_y+players_h);
          glRectf(players3_x, players3_y,
players3_x+players_h, players3_y+players_w);
          glRectf(players4_x, players4_y,
players4_x+players_h, players4_y+players_w);


		glColor3ub(rand()%256, rand()%256,
rand()%256);
		glRectf(block_x_1, block_y_1, block_x_1+30,
block_y_1+6);
		glRectf(block_x_2, block_y_2, block_x_2+30,
block_y_2+6);
		glRectf(block_x_3, block_y_3, block_x_3+30,
block_y_3+6);
		glRectf(block_x_4, block_y_4, block_x_4+30,
block_y_4+6);
		glRectf(block_x_5, block_y_5, block_x_5+30,
block_y_5+6);
		glRectf(block_x_6, block_y_6, block_x_6+30,
block_y_6+6);
		glRectf(block_x_7, block_y_7, block_x_7+6,
block_y_7+22);
		glRectf(block_x_8, block_y_8, block_x_8+6,
block_y_8+22);
            glRectf(block_x_9, block_y_9, block_x_9+6,
block_y_9+22);
            glRectf(block_x_10, block_y_10, block_x_10+6,
block_y_10+22);
            glRectf(block_x_11, block_y_11, block_x_11+6,
block_y_11+22);
            glRectf(block_x_12, block_y_12, block_x_12+6,
block_y_12+22);
            glColor3f(1.0f, 1.0f, 1.0f);
            glBegin(GL_POLYGON);
          for(int i=0;i<circle;i++)

     glVertex2f(r*cos(2*PI*i/circle)+ball_x,r*sin(2*PI*i/ci
rcle)+ball_y);
          glEnd();
     }
     if(level == 2 && state != 2)
     {
        glColor3f(1.0f, 1.0f, 0.0f);
            glRectf(players1_x, players1_y,
players1_x+players_w, players1_y+players_h);
            block_x_1 = rand()%500;    block_y_2 =
rand()%80+170;
            block_x_2 = rand()%400;    block_y_2 =
rand()%100+150;
            block_x_5 = rand()%300;    block_y_5 =
rand()%120+130;
            block_x_6 = rand()%200;    block_y_6 =
rand()%140+110;
            glColor3ub(rand()%256, rand()%256,
rand()%256);
            glRectf(block_x_1, block_y_2, block_x_1+30,
block_y_2+6);
            glRectf(block_x_2, block_y_2, block_x_2+30,
block_y_2+6);
            glRectf(block_x_3, block_y_3, block_x_3+30,
block_y_3+20);
```

```
          glRectf(block_x_4, block_y_4, block_x_4+30,
block_y_4+20);
          glRectf(block_x_5, block_y_5, block_x_5+6,
block_y_5+22);
          glRectf(block_x_6, block_y_6, block_x_6+6,
block_y_6+22);
          glRectf(block_x_7, block_y_7, block_x_7+30,
block_y_7+20);
          glRectf(block_x_8, block_y_8, block_x_8+30,
block_y_8+20);

          glColor3f(1.0f, 1.0f, 1.0f);
          glBegin(GL_POLYGON);
        for(int i=0;i<circle;i++)

    glVertex2f(r*cos(2*PI*i/circle)+ball_x,r*sin(2*PI*i/ci
rcle)+ball_y);
        glEnd();
    }
```

```
        glutSwapBuffers();
}

void TimerFunction(int value)
{

      if(state==1 && level == 1)
      {
            if((ball_x == players3_x+players_h+r && ball_y <
players3_y+players_w && ball_y > players3_y) ||
(ball_x == players4_x-r && ball_y <
players4_y+players_w && ball_y > players4_y))
            {xstep = -xstep;}
            if((ball_y == players1_y+players_h+r && ball_x <
players1_x+players_w && ball_x > players1_x)||(ball_y
== players2_y-r && ball_x < players2_x+players_w &&
ball_x > players2_x))
               {ystep = -ystep;}
```

```
          if(ball_y >= block_y_1-r && ball_y <=
block_y_1+6+r && ball_x <= block_x_1+30+r && ball_x
>= block_x_1-r)
          {
            block_x_1 = -500;
                block_y_1 = 500;
                block_count++;
                score+=10;
                ystep = -ystep;
        }
        if(ball_y >= block_y_2-r && ball_y <=
block_y_2+6+r && ball_x <= block_x_2+30+r && ball_x
>= block_x_2-r)
          {
            block_x_2 = -500;
                block_y_2 = 500;
            block_count++;
            score+=10;
                ystep = -ystep;
          }
                 if(ball_y >= block_y_3-r && ball_y <=
block_y_3+6+r && ball_x <= block_x_3+30+r && ball_x
>= block_x_3-r)
            {
              block_x_3 = -500;
                  block_y_3 = 500;
                  block_count++;
                  score+=10;
                  ystep = -ystep;
          }
          if(ball_y >= block_y_4-r && ball_y <=
block_y_4+6+r && ball_x <= block_x_4+30+r && ball_x
>= block_x_4-r)
            {
              block_x_4 = -500;
                  block_y_4 = 500;
                  block_count++;
                  score+=10;
```

```
                ystep = -ystep;
            }
            if(ball_y >= block_y_5-r && ball_y <=
block_y_5+6+r && ball_x <= block_x_5+30+r && ball_x
>= block_x_5-r)
              {
                block_x_5 = -500;
                    block_y_5 = 500;
                    block_count++;
                    score+=10;
                    ystep = -ystep;
            }
            if(ball_y >= block_y_6-r && ball_y <=
block_y_6+6+r && ball_x <= block_x_6+30+r && ball_x
>= block_x_6-r)
              {
                block_x_6 = -500;
                    block_y_6 = 500;
                    block_count++;
                score+=10;
                ystep = -ystep;
            }
            if(ball_y >= block_y_7-r && ball_y <=
block_y_7+30+r && ball_x <= block_x_7+6+r && ball_x
>= block_x_7-r)
              {
                block_x_7 = -500;
                    block_y_7 = 500;
                    block_count++;
                    score+=10;
                    xstep = -xstep;
            }
            if(ball_y >= block_y_8-r && ball_y <=
block_y_8+30+r && ball_x <= block_x_8+6+r && ball_x
>= block_x_8-r)
              {
                block_x_8 = -500;
                    block_y_8 = 500;
                    block_count++;

                    score+=10;

                    xstep = -xstep;

          }

          if(ball_y >= block_y_9-r && ball_y <=
block_y_9+30+r && ball_x <= block_x_9+6+r && ball_x
>= block_x_9-r)

            {

               block_x_9 = -500;

                    block_y_9 = 500;

                    block_count++;

                    score+=10;

                    xstep = -xstep;

          }

          if(ball_y >= block_y_10-r && ball_y <=
block_y_10+30+r && ball_x <= block_x_10+6+r &&
ball_x >= block_x_10-r)

            {

               block_x_10 = -500;
```

```
                block_y_10 = 500;

                block_count++;

                score+=10;

                xstep = -xstep;

            }

            if(ball_y >= block_y_11-r && ball_y <=

        block_y_11+30+r && ball_x <= block_x_11+6+r &&

        ball_x >= block_x_11-r)

             {

              block_x_11 = -500;

                block_y_11 = 500;

                block_count++;

                score+=10;

                xstep = -xstep;

            }

            if(ball_y >= block_y_12-r && ball_y <=

        block_y_12+30+r && ball_x <= block_x_12+6+r &&

        ball_x >= block_x_12-r)

             {
```

```
        block_x_12 = -500;
            block_y_12 = 500;
            block_count++;
            score+=10;
            xstep = -xstep;
    }

    ball_x += xstep;
    ball_y += ystep;
}
if( state==1 && level == 2)
{
    if(ball_x > windowWidth-r || ball_x < r)
    {xstep = -xstep;}

    if(ball_y > windowHeight-r)
    {ystep = -ystep;}

      if((ball_y == players1_y+players_h+r && ball_x
< players1_x+players_w && ball_x > players1_x))
          {ystep = -ystep;}
          if(ball_y >= block_y_1-r && ball_y <=
block_y_1+6+r && ball_x <= block_x_1+30+r && ball_x
>= block_x_1-r)
          {
                if (rand()%2==0) ystep = -ystep; else xstep
= -xstep;
        }
        if(ball_y >= block_y_2-r && ball_y <=
block_y_2+6+r && ball_x <= block_x_2+30+r && ball_x
>= block_x_2-r)
          {
            if (rand()%2==0) ystep = -ystep; else xstep = -
xstep;
        }
        if(ball_y >= block_y_3-r && ball_y <=
block_y_3+20+r && ball_x <= block_x_3+30+r &&
ball_x >= block_x_3-r)
```

```
            {
              block_x_3 = -500;
                  block_y_3 = 500;
                  block_count++;
                  score+=10;
                  if (rand()%2==0) r++; else xstep = -xstep;
          }
          if(ball_y >= block_y_4-r && ball_y <=
  block_y_4+20+r && ball_x <= block_x_4+30+r &&
  ball_x >= block_x_4-r)
            {
              block_x_4 = -500;
                  block_y_4 = 500;
                  block_count++;
                  score+=10;
                  if (rand()%2==0) ystep = -ystep; else xstep
  = r--;
          }
          if(ball_y >= block_y_5-r && ball_y <=
  block_y_5+6+r && ball_x <= block_x_5+30+r && ball_x
  >= block_x_5-r)
            {
              if (rand()%2==0) ystep = -ystep+1; else xstep =
  -xstep-1;
          }
          if(ball_y >= block_y_6-r && ball_y <=
  block_y_6+6+r && ball_x <= block_x_6+30+r && ball_x
  >= block_x_6-r)
            {
              if (rand()%2==0) ystep = -ystep-1; else xstep =
  -xstep+1;
          }
          if(ball_y >= block_y_7-r && ball_y <=
  block_y_7+30+r && ball_x <= block_x_7+22+r &&
  ball_x >= block_x_7-r)
            {
              block_x_7 = -500;
                  block_y_7 = 500;
```

```
                        block_count++;

                        score+=10;

                        if (rand()%2==0) ystep = players_w+=10;

else xstep = -xstep;

            }

            if(ball_y >= block_y_8-r && ball_y <=

block_y_8+30+r && ball_x <= block_x_8+22+r &&

ball_x >= block_x_8-r)

               {

                  block_x_8 = -500;

                        block_y_8 = 500;

                        block_count++;

                        score+=10;

                        if (rand()%2==0) ystep = -ystep; else xstep

= players_w-=10;

            }

               ball_x += xstep;

            ball_y += ystep;

      }
```

```
        glutPostRedisplay();
        glutTimerFunc(3,TimerFunction, 1);
}


///////////////////////////////////////////////////////
//////
// Setup the rendering state
void SetupRC(void)
        {
        // Set clear color to blue
        glClearColor(0.0f, 0.0f, 1.0f, 1.0f);
        }


///////////////////////////////////////////////////////
//////
// Called by GLUT library when the window has
changed size
```

```
void ChangeSize(int w, int h)
        {
        GLfloat aspectRatio;

        // Prevent a divide by zero
        if(h == 0)
                h = 1;

        // Set Viewport to window dimensions
        glViewport(0, 0, w, h);

        // Reset coordinate system
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();

     if (w <= h)
          {
          windowHeight = 250*h/w;
          windowWidth = 250;
          }
        else
          {
          windowWidth = 250*w/h;
          windowHeight = 250;
          }

      // Set the clipping volume
      glOrtho(0.0f, windowWidth, 0.0f, windowHeight,
1.0f, -1.0f);
        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();
        }

//////////////////////////////////////////////////////
//////
// Main program entry point
int main(int argc, char* argv[])
{
}

	srand(time(0));

	glutInit(&argc, argv);

	glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);

	glutInitWindowSize(600,600);

	  glutCreateWindow("期中作業");

	  glutKeyboardFunc(ESC);


	glutSpecialFunc(Arrowkeys);

	glutDisplayFunc(RenderScene);

	  glutReshapeFunc(ChangeSize);

	glutTimerFunc(10, TimerFunction, 1);


	SetupRC();


	glutMainLoop();


	  return 0;

}
```