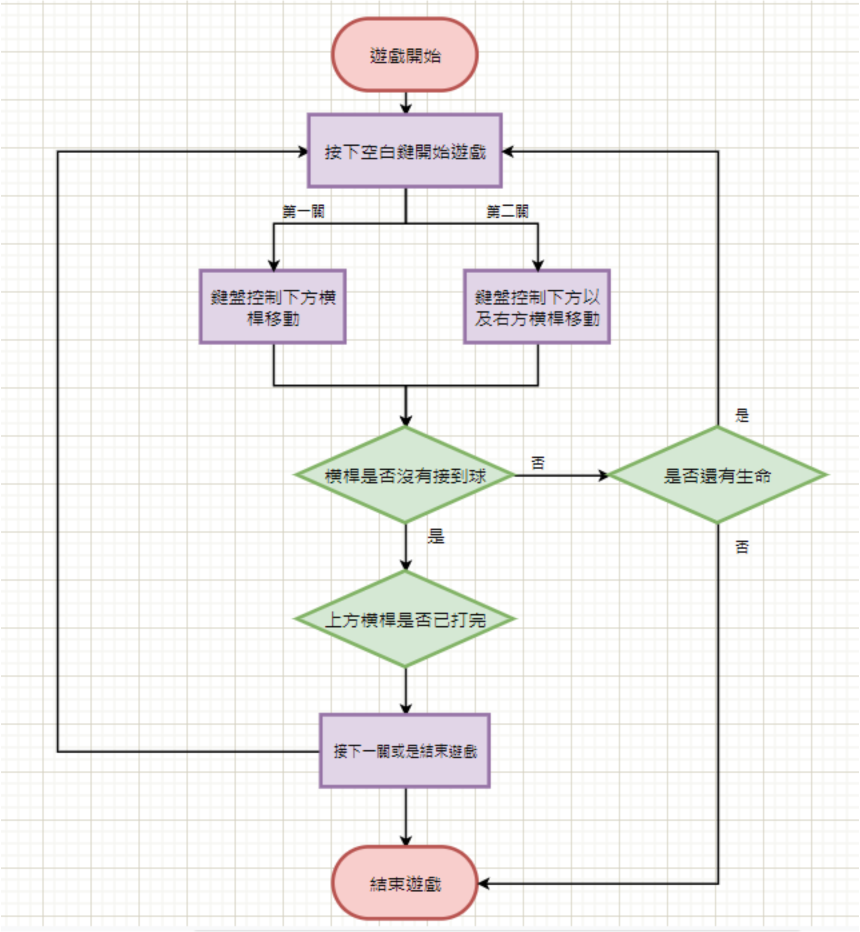
打磚塊遊戲

班級：資工四乙 學號：406262333 姓名：吳佩臻

一、 程式架構



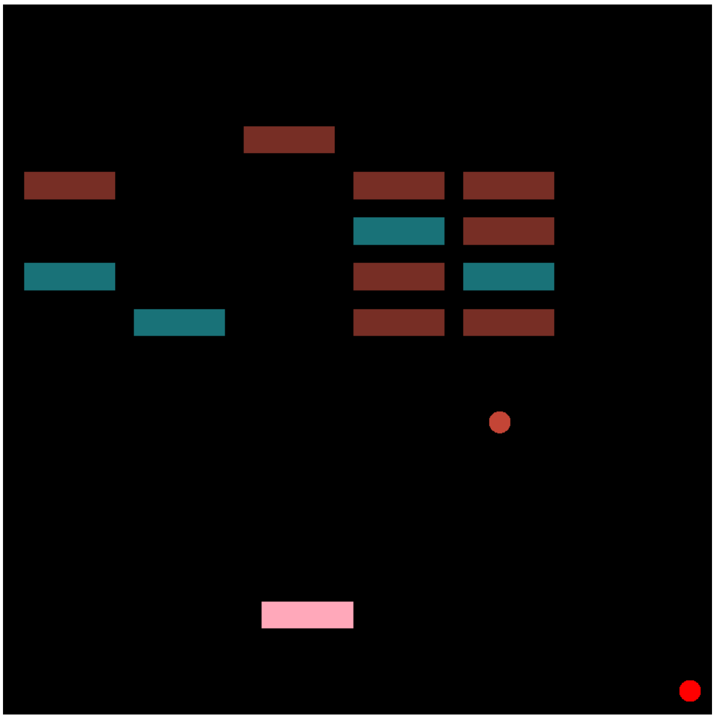
二、 討論

除了最基本的下方有移動板以及球彈跳之外，為了提升遊戲困難度，於第二關時增加了右方的移動板。與同學討論後決定增加右下方的生命提示，方便玩家了解目前的生命狀況。

三、　執行畫面

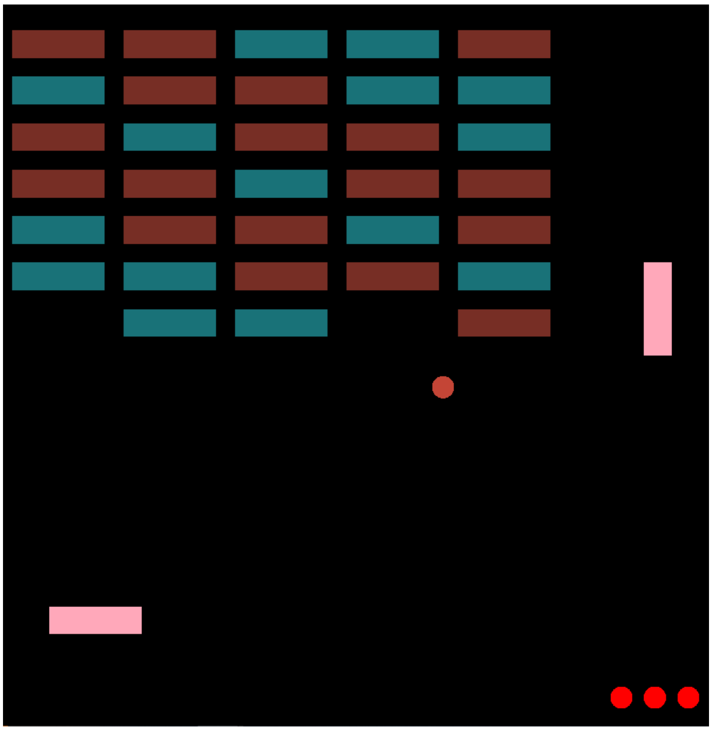
第一關：

第一關只有下方的移動橫桿。右下方有生命提示。



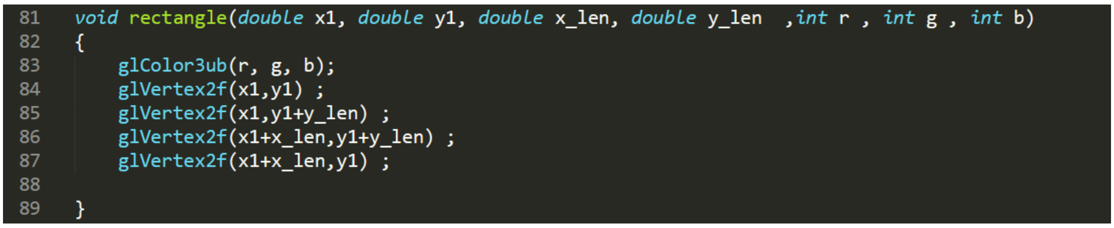
第二關：

第一關下方以及右方都有可移動的橫桿。右下方有生命提示。



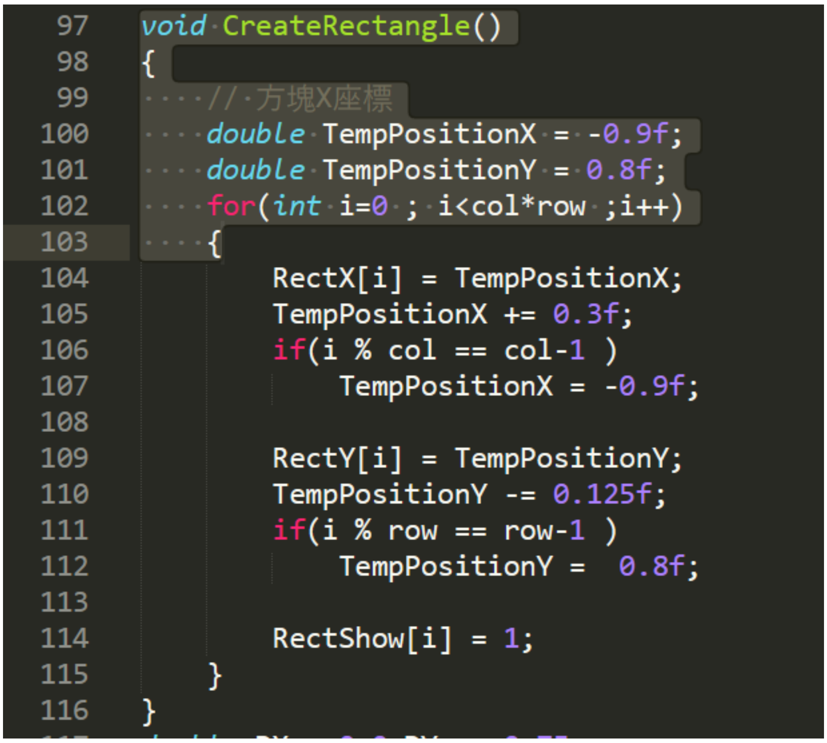
四、 程式碼

1. 產生長方形的副程式



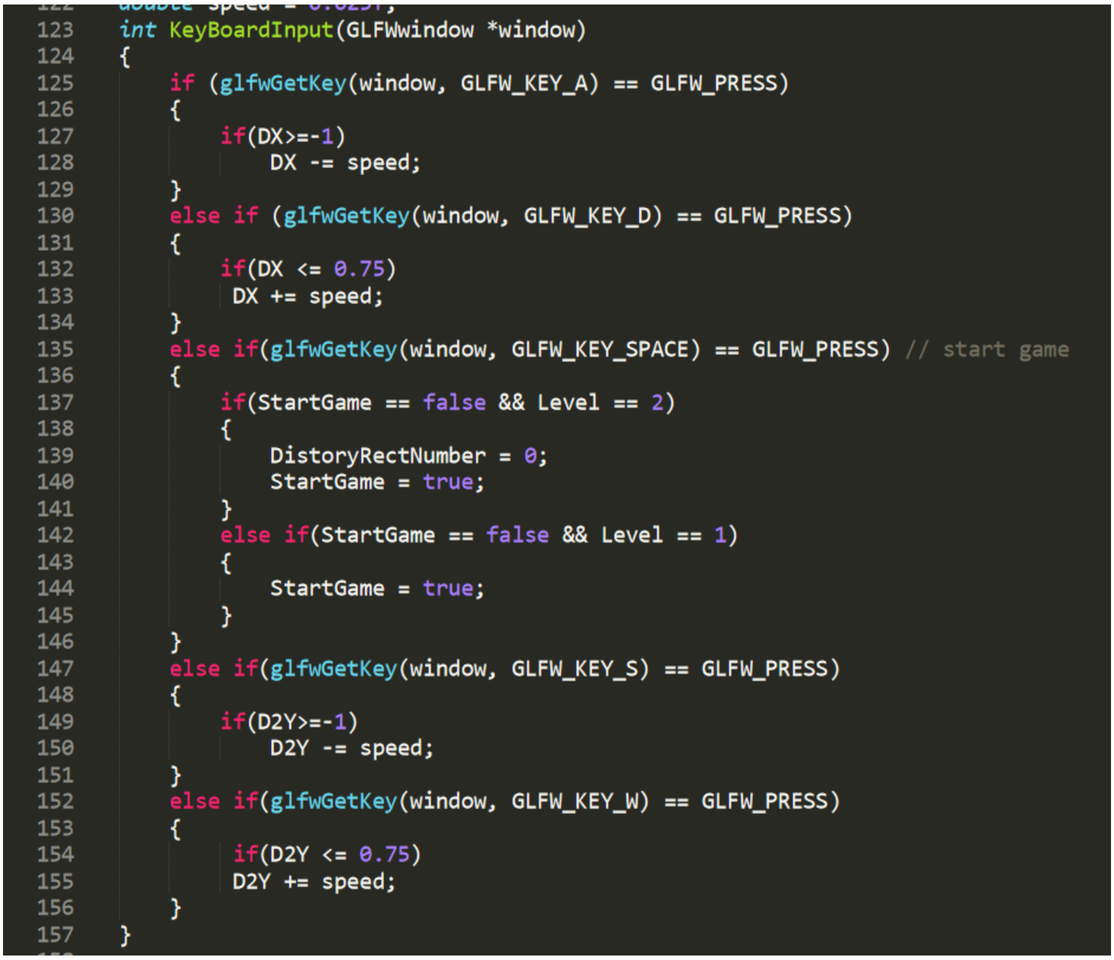
```
void rectangle(double x1, double y1, double x_len, double y_len  ,int r , int g , int b)
{
    glColor3ub(r, g, b);
    glVertex2f(x1,y1) ;
    glVertex2f(x1,y1+y_len) ;
    glVertex2f(x1+x_len,y1+y_len) ;
    glVertex2f(x1+x_len,y1) ;

}
```

2. 用 FOR 迴圈一次產生所有上方的橫條的位置



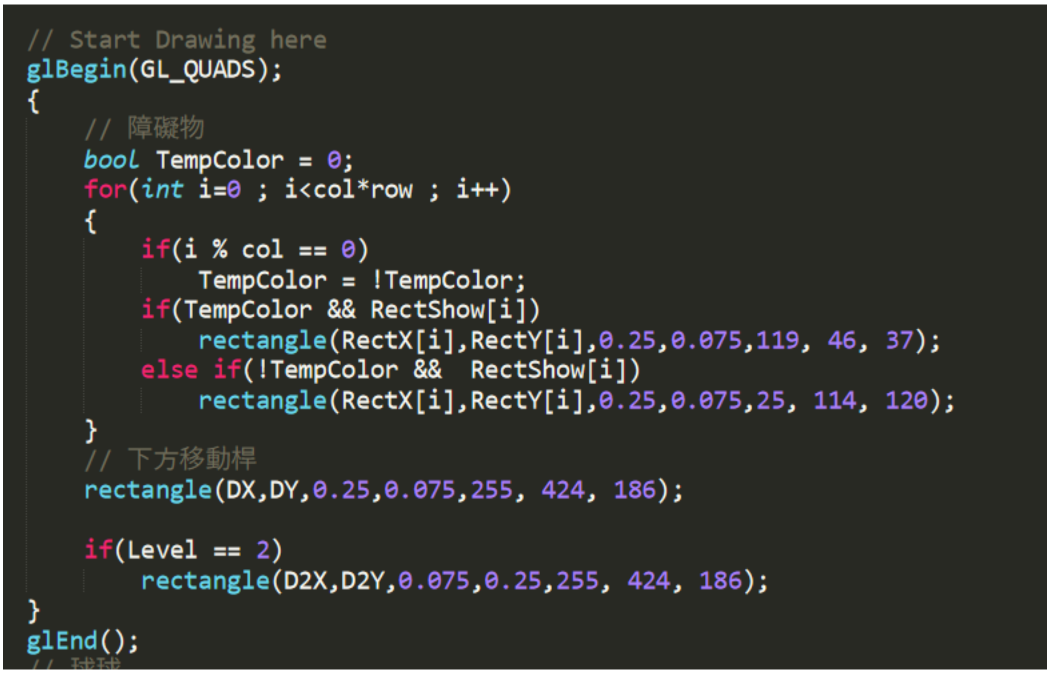
```
void CreateRectangle()
{
    // 方塊X座標
    double TempPositionX = -0.9f;
    double TempPositionY = 0.8f;
    for(int i=0 ; i<col*row ;i++)
    {
        RectX[i] = TempPositionX;
        TempPositionX += 0.3f;
        if(i % col == col-1 )
            TempPositionX = -0.9f;

        RectY[i] = TempPositionY;
        TempPositionY -= 0.125f;
        if(i % row == row-1 )
            TempPositionY =  0.8f;

        RectShow[i] = 1;
    }
}
```

3. 鍵盤輸入事件



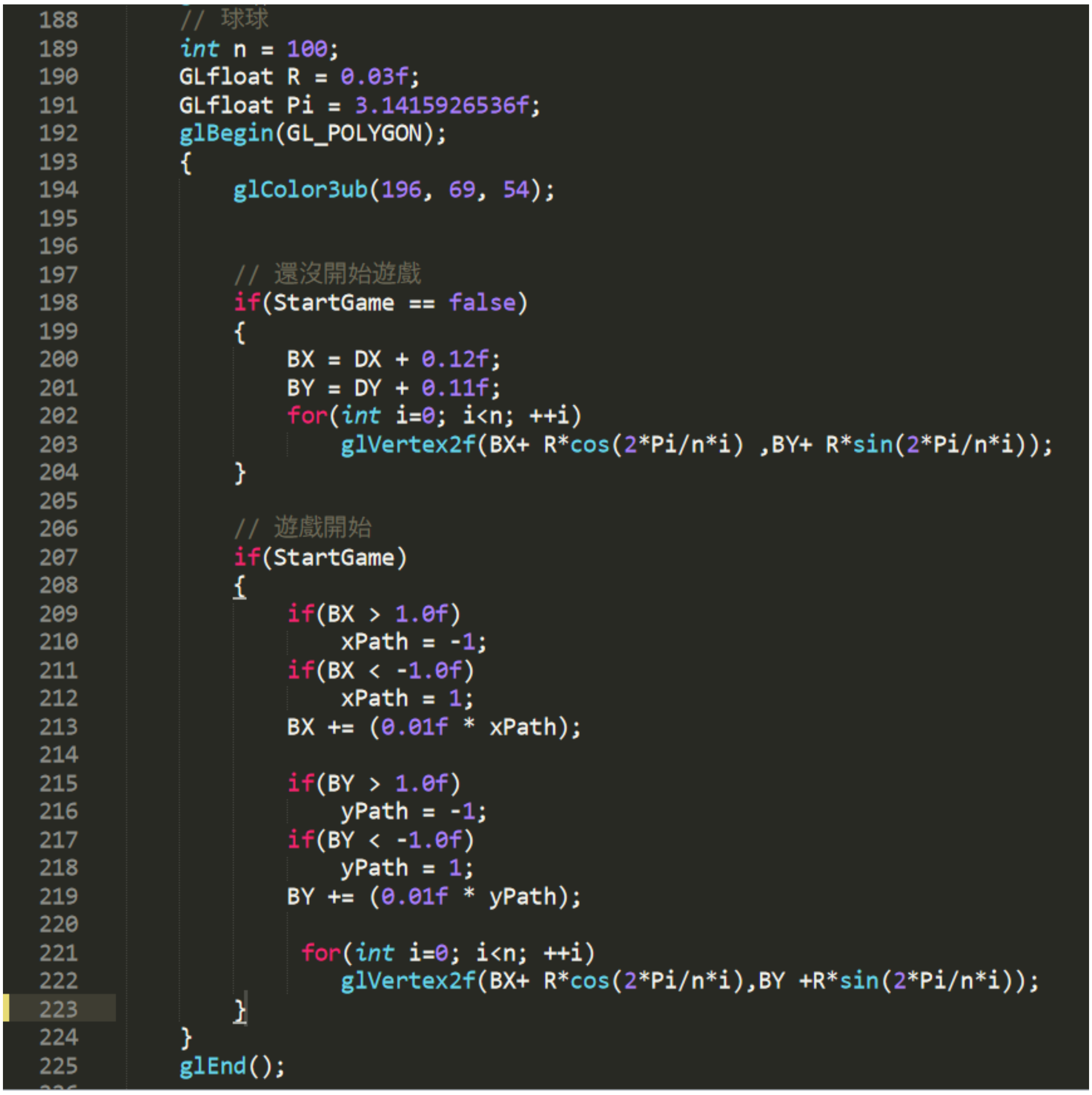
```
double speed = 0.025f;
int KeyBoardInput(GLFWwindow *window)
{
    if (glfwGetKey(window, GLFW_KEY_A) == GLFW_PRESS)
    {
        if(DX>=-1)
            DX -= speed;
    }
    else if (glfwGetKey(window, GLFW_KEY_D) == GLFW_PRESS)
    {
        if(DX <= 0.75)
         DX += speed;
    }
    else if(glfwGetKey(window, GLFW_KEY_SPACE) == GLFW_PRESS) // start game
    {
        if(StartGame == false && Level == 2)
        {
            DistoryRectNumber = 0;
            StartGame = true;
        }
        else if(StartGame == false && Level == 1)
        {
            StartGame = true;
        }
    }
    else if(glfwGetKey(window, GLFW_KEY_S) == GLFW_PRESS)
    {
        if(D2Y>=-1)
            D2Y -= speed;
    }
    else if(glfwGetKey(window, GLFW_KEY_W) == GLFW_PRESS)
    {
         if(D2Y <= 0.75)
         D2Y += speed;
    }
}
```

4. 繪出上方的橫條



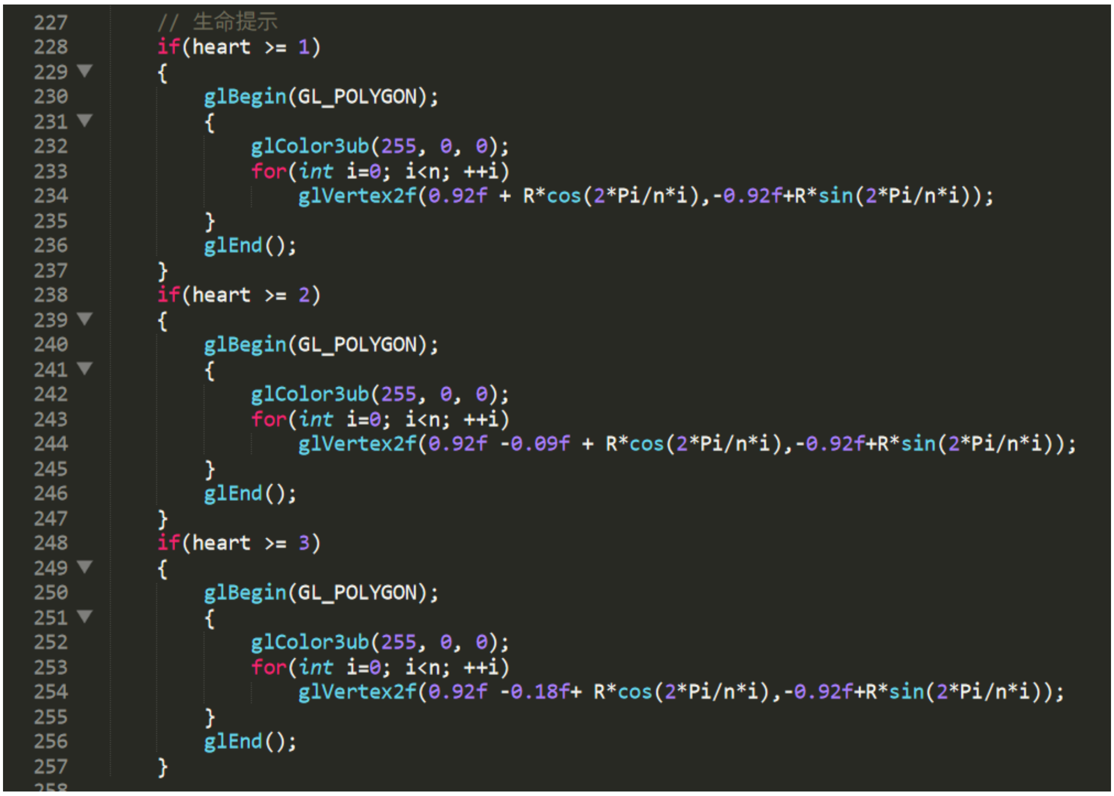
```
// Start Drawing here
glBegin(GL_QUADS);
{
    // 障礙物
    bool TempColor = 0;
    for(int i=0 ; i<col*row ; i++)
    {
        if(i % col == 0)
            TempColor = !TempColor;
        if(TempColor && RectShow[i])
            rectangle(RectX[i],RectY[i],0.25,0.075,119, 46, 37);
        else if(!TempColor &&  RectShow[i])
            rectangle(RectX[i],RectY[i],0.25,0.075,25, 114, 120);
    }
    // 下方移動桿
    rectangle(DX,DY,0.25,0.075,255, 424, 186);

    if(Level == 2)
        rectangle(D2X,D2Y,0.075,0.25,255, 424, 186);
}
glEnd();
// 球球
```

5. 繪出 彈跳球



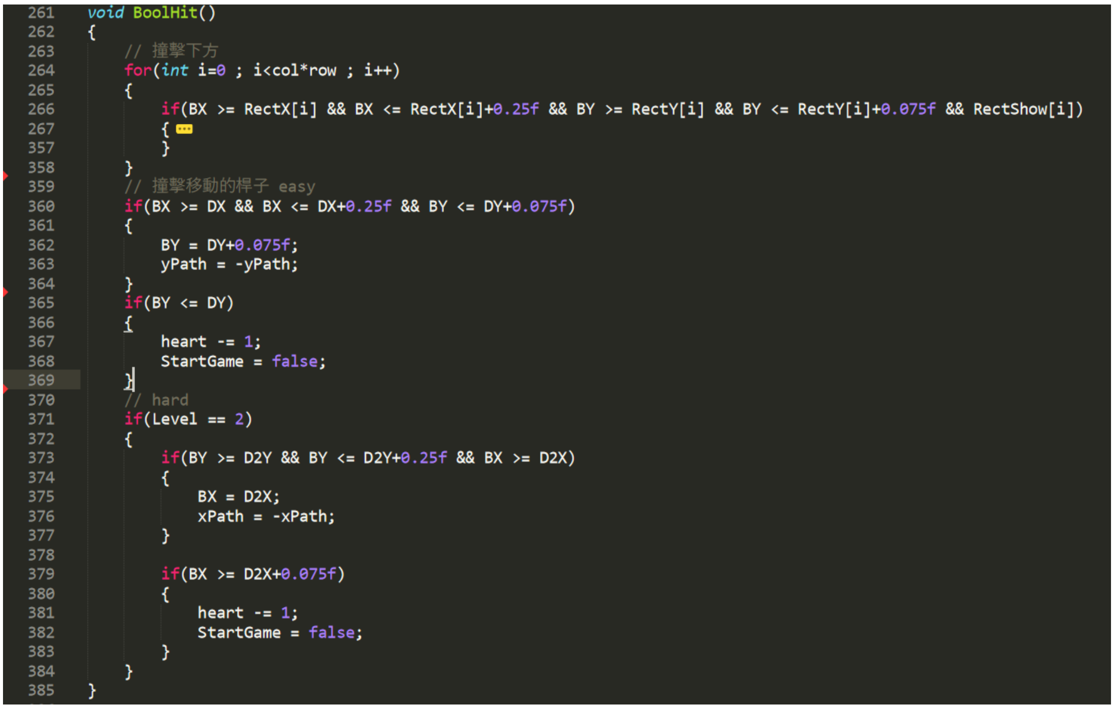
```
    // 球球
    int n = 100;
    GLfloat R = 0.03f;
    GLfloat Pi = 3.1415926536f;
    glBegin(GL_POLYGON);
    {
        glColor3ub(196, 69, 54);


        // 還沒開始遊戲
        if(StartGame == false)
        {
            BX = DX + 0.12f;
            BY = DY + 0.11f;
            for(int i=0; i<n; ++i)
                glVertex2f(BX+ R*cos(2*Pi/n*i) ,BY+ R*sin(2*Pi/n*i));
        }

        // 遊戲開始
        if(StartGame)
        {
            if(BX > 1.0f)
                xPath = -1;
            if(BX < -1.0f)
                xPath = 1;
            BX += (0.01f * xPath);

            if(BY > 1.0f)
                yPath = -1;
            if(BY < -1.0f)
                yPath = 1;
            BY += (0.01f * yPath);

             for(int i=0; i<n; ++i)
                glVertex2f(BX+ R*cos(2*Pi/n*i),BY +R*sin(2*Pi/n*i));
        }
    }
    glEnd();
```

6. 繪出生命提示球



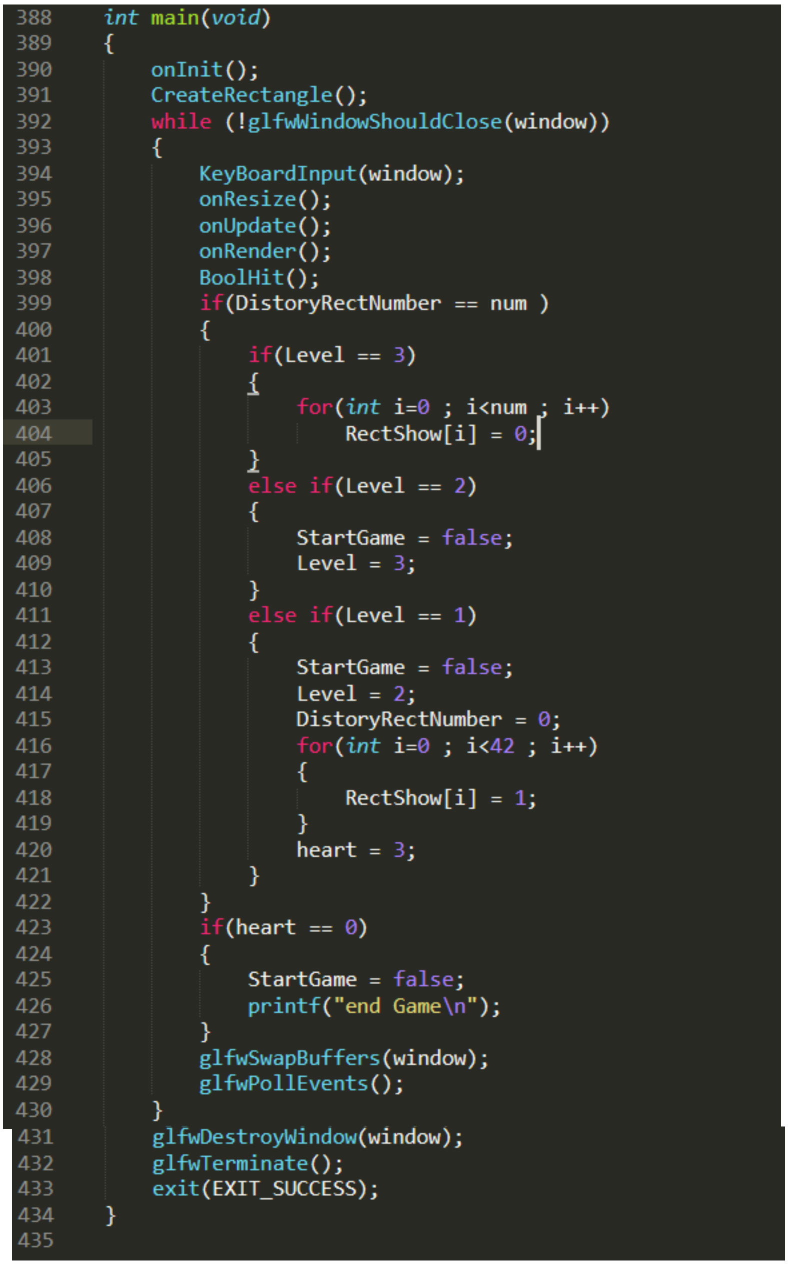
```
    // 生命提示
    if(heart >= 1)
    {
        glBegin(GL_POLYGON);
        {
            glColor3ub(255, 0, 0);
            for(int i=0; i<n; ++i)
                glVertex2f(0.92f + R*cos(2*Pi/n*i),-0.92f+R*sin(2*Pi/n*i));
        }
        glEnd();
    }
    if(heart >= 2)
    {
        glBegin(GL_POLYGON);
        {
            glColor3ub(255, 0, 0);
            for(int i=0; i<n; ++i)
                glVertex2f(0.92f -0.09f + R*cos(2*Pi/n*i),-0.92f+R*sin(2*Pi/n*i));
        }
        glEnd();
    }
    if(heart >= 3)
    {
        glBegin(GL_POLYGON);
        {
            glColor3ub(255, 0, 0);
            for(int i=0; i<n; ++i)
                glVertex2f(0.92f -0.18f+ R*cos(2*Pi/n*i),-0.92f+R*sin(2*Pi/n*i));
        }
        glEnd();
    }
```

7. 判斷球是否撞擊橫桿



```
void BoolHit()
{
    // 撞擊下方
    for(int i=0 ; i<col*row ; i++)
    {
        if(BX >= RectX[i] && BX <= RectX[i]+0.25f && BY >= RectY[i] && BY <= RectY[i]+0.075f && RectShow[i])
        { …
        }
    }
    // 撞擊移動的桿子 easy
    if(BX >= DX && BX <= DX+0.25f && BY <= DY+0.075f)
    {
        BY = DY+0.075f;
        yPath = -yPath;
    }
    if(BY <= DY)
    {
        heart -= 1;
        StartGame = false;
    }
    // hard
    if(Level == 2)
    {
        if(BY >= D2Y && BY <= D2Y+0.25f && BX >= D2X)
        {
            BX = D2X;
            xPath = -xPath;
        }

        if(BX >= D2X+0.075f)
        {
            heart -= 1;
            StartGame = false;
        }
    }
}
```

8. 主程式



```
int main(void)
{
    onInit();
    CreateRectangle();
    while (!glfwWindowShouldClose(window))
    {
        KeyBoardInput(window);
        onResize();
        onUpdate();
        onRender();
        BoolHit();
        if(DistoryRectNumber == num )
        {
            if(Level == 3)
            {
                for(int i=0 ; i<num ; i++)
                    RectShow[i] = 0;
            }
            else if(Level == 2)
            {
                StartGame = false;
                Level = 3;
            }
            else if(Level == 1)
            {
                StartGame = false;
                Level = 2;
                DistoryRectNumber = 0;
                for(int i=0 ; i<42 ; i++)
                {
                    RectShow[i] = 1;
                }
                heart = 3;
            }
        }
        if(heart == 0)
        {
            StartGame = false;
            printf("end Game\n");
        }
        glfwSwapBuffers(window);
        glfwPollEvents();
    }
    glfwDestroyWindow(window);
    glfwTerminate();
    exit(EXIT_SUCCESS);
}
```