**亞洲大學**

**資訊工程學系**

**『程式設計(二)』**

**專題導向學習期末報告**

(桌上冰球)

**指導老師：周永振 老師**

**學生： 黃意中/103021112、 黃致鈞/103021013**

**中華民國 106 年 6 月 19 日(繳交日期16pt/粗體)**

本專題報告符合亞洲大學Asia-Team 1.0 格式規範

目錄

目錄-----------------------------------------------------------------------1

一、專題問題描述 ---------------------------------------------------

二、專題執行步驟-----------------------------------------------------

2-1. 整合管理 (Project Integration Management) ----

要討論在整個專案由計劃開始到執行結束的整個專案生命週期之中，

所不可或缺、「貫穿前後」的重要活動的管理。2-2. 管理範疇界定(Project Scope Management) ------------  
 2-3. 時程與進度管理(例如是以甘特圖的方式來說明)

(Project Time Management) --------------------------------  
2-4. 成本控制與管理(例如是以預算與實際支出比較表來

說明)(Project Cost Management) ------------------------  
**2-5.**品質管理(說明專案成果的品質控管方式)

(Project Quality Management) ----------------------------  
 2-6. 人力資源管理(例如說明團隊分工項目，以及成員是否依計畫貢獻所長)(Project Human Resource Management) --  
2-7. 溝通管理(例如專案的期中進度的發表，或是結案公開發

表)(Project Communication Management) -------------  
2-8. 風險管理(例如說明專案如果未能依計劃進行時的配套措

施)(Project Risk Management) ----------------------------  
2-9. 採購或是資源取得管理(例如說明專案執行過程中，如

何取得必須的資源)(Project Procurement Management) -  
2-10. 利害關係人的管理(例如說明會受到此專案成敗影響的

直接或是間接關係人的溝通，或是如何確保大家的權

益)(Project Stakeholder Management) ------------

三、專題執行成果與績效-----------------------------------

四、專題反思與討論------------------------------------------

五、參考文獻--------------------------------------------------

一、專題問題描述

對於一開始的題目想了許久,結合了這學期的視窗、awt常用元件、視窗事件處理和例外,終於有了點想法,就是關於球類的,但到底是關於哪一種球類我們又想一段時間,如果是籃球或足球、棒球這種大型球類的遊戲肯定會花費不少精力與時間,所以又把範圍縮小到兩人的球類遊戲,最終的選擇就是類似於桌球的桌上冰球。

(桌上冰球影片<https://www.youtube.com/watch?v=VkTpWFFjLFQ>)

在設計桌上冰球上,視窗的架構和一些簡單的排場都是沒甚麼困難的,比較讓我們頭痛的是球碰撞而反彈的角度,但我們折衷用比較簡單的方式去寫反彈這個步驟,第二個問題點在於我們到底要用P2(第二個玩家)還是電腦來當作對手,在設計電腦的上面遇到了許多疑難雜症,例如如何跟著球的移動去作出相等的移動模式,所以我們又折衷做了兩個玩家對戰的模式。

二、專題執行步驟

2-1. 整合管理 (Project Intergration Management)

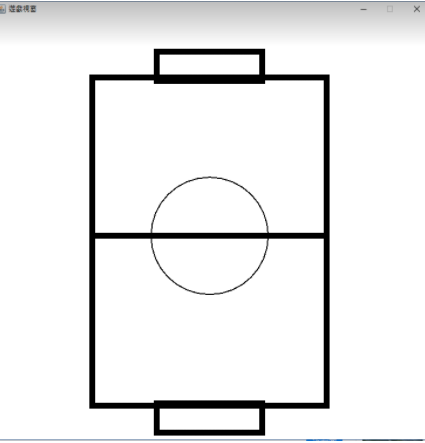
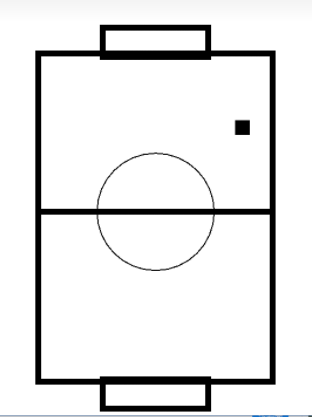
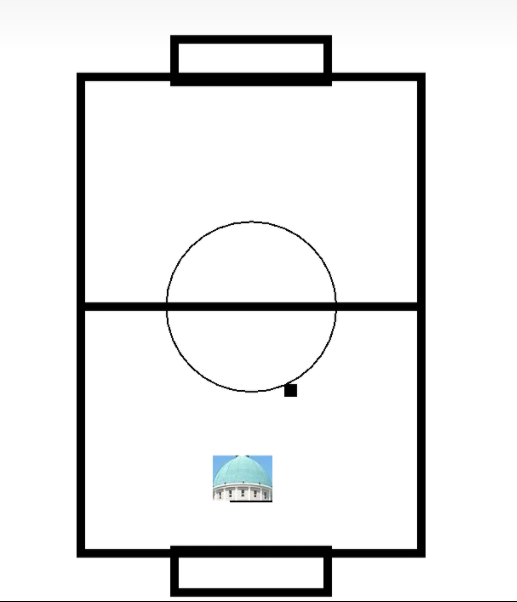
1. 專案計畫規劃: 做出Meun的 1.(玩家間對戰) 2.(電腦對戰-困難、中等、簡單) ,再來就是視窗的架構,球的判定和板子(玩家、電腦)的移動。
2. 專案計畫執行: 先寫出Manu然後視窗的整個架構接下來就是球的移動程序最後就是P1和P2的控制及判定
3. 修改頁面美化及新增小功能

2-2. 管理範疇界定(Project Scope Management)

把這學期最重要的『視窗』給表現出來,再來就是從『球』這個方面去衍生出許多不同的想法 ,

2-3. 時程與進度管理(例如是以甘特圖的方式來說明)

(Project Time Management)

1. 做出整個視窗布局的架構: 
2. 接下來就是主體(球): 
3. 在來就是玩家(P1和P2): 

2-4. 成本控制與管理(例如是以預算與實際支出比較表來

說明)(Project Cost Management)

1. 資訊計畫編制:如果把成本是當作時間來看待的話,我們預計總共花一個月整來完成我們所要達到的最低目標
2. 成本估算:如果在一個月內我們能達到所需的目標,在來就是把視窗美化和新增與電腦對戰
3. 成本安排:一個禮拜之內完成視窗的架構,兩個禮拜內完成球的判定和玩家間的對戰,在來就是剩餘的時間新增Menu和美化視窗和電腦對戰。

**2-5.**品質管理(說明專案成果的品質控管方式)

(Project Quality Management)

對於品質其實我們一開始的期待是信心滿滿的,但在過程中實在是發生了太多疑難雜症導致我們只好把程式簡單化。

2-6. 人力資源管理(例如說明團隊分工項目，以及成員是否依計畫貢獻所長)(Project Human Resource Management)

黃意中:著手視窗的架構和主體(球)的判定。

黃致鈞:負責Menu美化、小功能和玩家對戰以及電腦對戰的部分。

2-7. 溝通管理(例如專案的期中進度的發表，或是結案公開發

表)(Project Communication Management)

基本上我們每天都會見面,也能用通訊軟體交流，所以我們都能掌握對方開發程式的過程,而在每周的星期一都會和教授分享進度已經聆聽教授所給予的意見進行修正。  
2-8. 風險管理(例如說明專案如果未能依計劃進行時的配套措

施)(Project Risk Management)

如果在預期內沒法完成我們所達到的目標,那就先把更進階的功能放一邊,例如與電腦對戰或是連線對戰等難題。  
2-9. 採購或是資源取得管理(例如說明專案執行過程中，如

何取得必須的資源)(Project Procurement Management)

在這個專題當中我們就是四處去取得所要的資源,課本、網路等等在改寫成自己想要的東西。

2-10. 利害關係人的管理(例如說明會受到此專案成敗影響的

直接或是間接關係人的溝通，或是如何確保大家的權

益)(Project Stakeholder Management)

我們的專題算是簡單的,所以沒有所謂的成敗,只有功能的多寡以及要不要去付出去做而已。

三、專題執行成果與績效

遊戲程式碼

Main

**import** java.awt.\*;

**import** java.awt.\*;

**import** javax.swing.JFrame;

**import** java.awt.\*;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.ImageIcon;

**import** javax.swing.JOptionPane;

**import** java.awt.Rectangle;

**import** java.awt.event.KeyAdapter;

**import** java.awt.event.KeyEvent;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**public** **class** Main **extends** JFrame {

**int** score1;

**int** score2;

**int** s1;

**int** s2;

String winner ;

String a = JOptionPane.*showInputDialog*("請輸入玩家1名稱");

String b = JOptionPane.*showInputDialog*("請輸入玩家2名稱");

**int** screen = 1;

**int** OverGameScore = 5;

Image net1image;

Image net2image;

Rectangle startBottom = **new** Rectangle(280, 250, 200, 100);

Rectangle SetOverScore = **new** Rectangle(280, 400, 200, 100);

Rectangle ScoreBord = **new** Rectangle(280, 550, 200, 100);

Rectangle explanation = **new** Rectangle(280, 680, 200, 50);

Rectangle net1 = **new** Rectangle(280, 86, 180, 50);

Rectangle net2 = **new** Rectangle(280, 689, 180, 50);

**static** Ball *ball* = **new** Ball();

**static** P1 *p1* = **new** P1();

**static** P2 *p2* = **new** P2();

**public** Main() {

ImageIcon net = **new** ImageIcon("");

net1image = net.getImage();

net2image = net.getImage();

setTitle("遊戲視窗");

setSize(750, 750);

setResizable(**false**);

setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

setVisible(**true**);

setBackground(Color.*white*);

addKeyListener(**new** KeyHandler());

addKeyListener(**new** KeyHandler2());

addMouseListener(**new** MouseHandler());

}

**public** **void** start() {

screen = 2;

score1 = 0;

score2 = 0;

}

**public** **void** score() {

**if** (*ball*.ball.intersects(net1)) {

score1++;

}

**if** (*ball*.ball.intersects(net2)) {

score2++;

}

}

**public** **class** KeyHandler **extends** KeyAdapter {

**public** **void** keyPressed(KeyEvent e) {

**if** (screen == 2)

*p1*.keyPressed(e);

**if** (screen == 1) {

**if** (e.getKeyCode() == e.*VK\_UP*) {

OverGameScore++;

}

**if** (e.getKeyCode() == e.*VK\_DOWN*) {

OverGameScore--;

}

**if** (OverGameScore <= 0) {

JOptionPane.*showMessageDialog*(**null**, "請調整為大於0的值", "跳出視窗內容",

JOptionPane.*INFORMATION\_MESSAGE*);

}

}

**if** (screen == 3 || screen == 4) {

**if** (e.getKeyCode() == e.*VK\_SPACE*) {

screen = 2;

}

**if** (e.getKeyCode() == e.*VK\_ESCAPE*) {

screen = 1;

}

}

**if** (screen == 5 || screen == 6) {

**if** (e.getKeyCode() == e.*VK\_ESCAPE*) {

screen = 1;

}

}

}

**public** **void** keyReleased(KeyEvent e) {

**if** (screen == 2)

*p1*.keyReleased(e);

}

}

**public** **class** KeyHandler2 **extends** KeyAdapter {

**public** **void** keyPressed(KeyEvent e) {

*p2*.keyPressed(e);

}

**public** **void** keyReleased(KeyEvent e) {

*p2*.keyReleased(e);

}

}

**public** **class** MouseHandler **extends** MouseAdapter {

**public** **void** mousePressed(MouseEvent e) {

**int** x = e.getX();

**int** y = e.getY();

**if** (screen == 1) {

**if** (x > startBottom.x && x < startBottom.x + startBottom.width

&& y > startBottom.y

&& y < startBottom.y + startBottom.height)

start();

**if** (x > explanation.x && x < explanation.x + explanation.width

&& y > explanation.y

&& y < explanation.y + explanation.height)

screen = 5;

**if** (x > ScoreBord.x && x < ScoreBord.x + ScoreBord.width

&& y > ScoreBord.y

&& y < ScoreBord.y + ScoreBord.height)

screen = 6;

}

}

}

**int** i = Integer.*parseInt*(a);

**int** x = Integer.*parseInt*(b);

**public** **void** collision() {

**if** (*ball*.ball.intersects(*p1*.P1)) {

*ball*.YD(-1);

*ball*.x--;

*ball*.speed -= 2;

**if** (*ball*.speed < 2)

*ball*.speed = 2;

}

**if** (*ball*.ball.intersects(*p2*.P2)) {

*ball*.YD(+1);

*ball*.x++;

*ball*.speed -= 2;

**if** (*ball*.speed < 2)

*ball*.speed = 2;

}

}

**public** **void** paint(Graphics g) {

Image doubleBufferImage;

Graphics doubleBufferGraphics;

doubleBufferImage = createImage(getWidth(), getHeight());

doubleBufferGraphics = doubleBufferImage.getGraphics();

paintComponent(doubleBufferGraphics);

g.drawImage(doubleBufferImage, 0, 0, **this**);

}

**public** **void** paintComponent(Graphics g) {

**if** (screen == 1) {

**if** (score1 >= 1)

score1 = 0;

**if** (score2 >= 1)

score2 = 0;

g.setFont(**new** Font("Arial", Font.*BOLD*, 70));

g.drawString("GAME", 270, 150);

g.setColor(Color.*BLUE*);

g.fillRect(startBottom.x, startBottom.y, startBottom.width,

startBottom.height);

g.setFont(**new** Font("Arial", Font.*BOLD*, 30));

g.setColor(Color.*WHITE*);

g.drawString(" START", startBottom.x + 20, startBottom.y + 60);

g.setColor(Color.*RED*);

g.fillRect(SetOverScore.x, SetOverScore.y, SetOverScore.width,

SetOverScore.height);

g.setFont(**new** Font("Arial", Font.*BOLD*, 21));

g.setColor(Color.*WHITE*);

g.drawString("Set Over Score :" + OverGameScore,

SetOverScore.x + 5, SetOverScore.y + 60);

g.setColor(Color.*WHITE*);

g.fillRect(explanation.x, explanation.y, explanation.width,

explanation.height);

g.setFont(**new** Font("Arial", Font.*BOLD*, 25));

g.setColor(Color.*GREEN*);

g.drawString("Control Instructions", explanation.x - 15,

explanation.y + 30);

g.setColor(Color.*BLACK*);

g.fillRect(ScoreBord.x, ScoreBord.y, ScoreBord.width,

ScoreBord.height);

g.setFont(**new** Font("Arial", Font.*BOLD*, 30));

g.setColor(Color.*GREEN*);

g.drawString("history score", ScoreBord.x + 10, ScoreBord.y + 60);

}

**if** (screen == 2) {

Graphics2D g2d = (Graphics2D) g;

BasicStroke stroke = **new** BasicStroke(2.0f);

g2d.setStroke(stroke);

g.drawRect(170, 130, 400, 560);

g.drawRect(net1.x, net1.y, net1.width, net1.height);

g.drawRect(net2.x, net2.y, net2.width, net2.height);

g.drawImage(net1image, 280, 86, **this**);

g.drawString(a + "得分 " + score1, 200, 650);

g.drawString(b + "得分" + score2, 490, 170);

BasicStroke stroke2 = **new** BasicStroke(2.0f);

g2d.setStroke(stroke2);

g.drawOval(270, 300, 200, 200);

g.fillRect(169, 395, 403, 2);

**if** (score1 >= OverGameScore) {

screen = 3;

s1 = score1;

s2 = score2;

score1 = 0;

score2 = 0;

}

**if** (score2 >= OverGameScore) {

screen = 4;

s1 = score1;

s2 = score2;

score1 = 0;

score2 = 0;

}

*ball*.draw(g);

*p1*.draw(g);

*p2*.draw(g);

score();

collision();

}

**if** (screen == 3) {

setBackground(Color.*MAGENTA*);

g.setFont(**new** Font("Arial", Font.*BOLD*, 40));

g.drawString(a + " win!", 260, 100);

g.drawString("press 'SPACE' to continue", 80, 300);

g.drawString("press 'ESC' to menu", 80, 600);

winner = a;

}

**if** (screen == 4) {

setBackground(Color.*ORANGE*);

g.setFont(**new** Font("Arial", Font.*BOLD*, 40));

g.drawString(b + " win!", 260, 100);

g.drawString("press 'SPACE' to continue", 100, 300);

g.drawString("press ' ESC ' to the menu", 100, 600);

winner = b;

}

**if** (screen == 5) {

setBackground(Color.*pink*);

g.setFont(**new** Font("Arial", Font.*BOLD*, 30));

g.drawString("Player 1 controls with 'left' and 'right' keys", 100,

150);

g.drawString("Player 2 controls with 'A' and 'D' keys", 100, 300);

g.drawString("Press keys 'Up' and 'Down' change the over score",

10, 450);

g.drawString("press ' ESC ' to the menu", 150, 600);

}

**if** (screen == 6) {

setBackground(Color.*CYAN*);

g.setFont(**new** Font("Arial", Font.*BOLD*, 40));

g.drawString(winner + " is the winner,score is " + s1 + ":" + s2, 100, 300);

g.drawString("press ' ESC ' to the menu", 120, 600);

}

repaint();

}

**public** **static** **void** main(String arges[]) {

Main main = **new** Main();

Thread t1 = **new** Thread(*ball*);

t1.start();

Thread t2 = **new** Thread(*p1*);

t2.start();

Thread t3 = **new** Thread(*p2*);

t3.start();

main.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

}

}

Ball

**import** java.awt.Graphics;

**import** java.awt.Rectangle;

**public** **class** Ball **implements** Runnable {

**int** x,y,xD,yD;

**int** speed = 10 ;

Rectangle ball;

**public** Ball(){

x=300;

y=300;

**int** xrDirection = (**int**)(2\*Math.*random*()); // 1/2/3

**if**(xrDirection==0)

xrDirection--;

XD(xrDirection);

**int** yrDirection = (**int**)(2\*Math.*random*());

**if**(yrDirection==0)

yrDirection--;

YD(yrDirection);

ball = **new** Rectangle(x,y,15,15); //弄出小正方體

}

**public** **void** XD(**int** xd){

xD=xd;

}

**public** **void** YD(**int** yd){

yD=yd;

}

**public** **void** movingTheBall(){

ball.x +=xD;

ball.y +=yD;

**if**(ball.x<= 170)

XD(+1);

**if**(ball.x>=560)

XD(-1);

**if**(ball.y<= 135){

YD(+1);

speed++;

}

**if**(ball.y>= 675){

YD(-1);

speed++;

}

}

**public** **void** draw(Graphics g){

g.fillRect(ball.x, ball.y, ball.width, ball.height);

}

**public** **void** run(){

**try**{

**for**(;;){

movingTheBall();

Thread.*sleep*(speed);

}

}

**catch**(Exception e){

System.*out*.println("錯誤");

}

}

}

P1

**import** java.awt.Graphics;

**import** java.awt.Image;

**import** java.awt.Rectangle;

**import** java.awt.event.KeyEvent;

**import** javax.swing.ImageIcon;

**public** **class** P1 **implements** Runnable {

**int** x,y,xDirection;

Image MainP1;

Rectangle P1;

**public** P1(){

x = 345;

y = 605;

P1 = **new** Rectangle (x,y,50,50);

ImageIcon MainP1Image = **new** ImageIcon("C:\\Users\\huang chun\\Desktop\\aaa\\ccc");

MainP1 = MainP1Image.getImage();

}

**public** **void** keyPressed(KeyEvent e){

**if**(e.getKeyCode()==e.*VK\_LEFT*){

XDirection(-1);

}

**if**(e.getKeyCode()==e.*VK\_RIGHT*){

XDirection(+1);

}

}

**public** **void** keyReleased(KeyEvent e){

**if**(e.getKeyCode()==e.*VK\_LEFT*){

XDirection(0);

}

**if**(e.getKeyCode()==e.*VK\_RIGHT*){

XDirection(0);

}

}

**public** **void** draw(Graphics g){

g.fillRect(P1.x,P1.y ,P1.width, P1.height);

g.drawImage(MainP1,x,y,**null**);

}

**public** **void** move(){

P1.x += xDirection;

x += xDirection;

**if**(P1.x <= 170)

P1.x = 170;

**if**(P1.x>=525)

P1.x =525;

**if**(x <= 170)

x = 170;

**if**(x>=525)

x =525;

}

**public** **void** XDirection(**int** xdirection){

xDirection = xdirection;

}

**public** **void** run(){

**try**{

**for**(;;){

move();

Thread.*sleep*(3); //延遲控制 數字越高越慢

}

}**catch**(Exception e){

System.*out*.println("Error");

}

}

}

P2

**import** java.awt.Graphics;

**import** java.awt.Image;

**import** java.awt.Rectangle;

**import** java.awt.event.KeyEvent;

**import** javax.swing.ImageIcon;

**public** **class** P2 **implements** Runnable {

**int** x2,y2,yDirection;

Image MainP2;

Rectangle P2;

**public** P2(){

x2 = 345;

y2 = 175;

P2 = **new** Rectangle (x2,y2,50,50);

ImageIcon MainP2Image = **new** ImageIcon("C:\\Users\\huang chun\\Desktop\\aaa\\bbb");

MainP2 = MainP2Image.getImage();

}

**public** **void** keyPressed(KeyEvent e){

**if**(e.getKeyCode()==e.*VK\_A*){

YDirection(-1);

}

**if**(e.getKeyCode()==e.*VK\_D*){

YDirection(+1);

}

}

**public** **void** keyReleased(KeyEvent e){

**if**(e.getKeyCode()==e.*VK\_A*){

YDirection(0);

}

**if**(e.getKeyCode()==e.*VK\_D*){

YDirection(0);

}

}

**public** **void** draw(Graphics g){

g.fillRect(P2.x,P2.y ,P2.width, P2.height);

g.drawImage(MainP2,x2,y2,**null**);

}

**public** **void** move(){

P2.x += yDirection;

x2 += yDirection;

**if**(P2.x <= 170)

P2.x = 170;

**if**(P2.x>=525)

P2.x =525;

**if**(x2 <= 170)

x2 = 170;

**if**(x2>=525)

x2 =525;

}

**public** **void** YDirection(**int** ydirection){

yDirection = ydirection;

}

**public** **void** run(){

**try**{

**for**(;;){

move();

Thread.*sleep*(3);

}

}**catch**(Exception e){

System.*out*.println("錯誤");

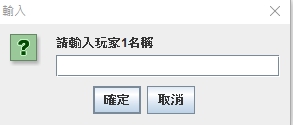
}

}

}

執行程式(圖文)

跳出輸入玩家名稱



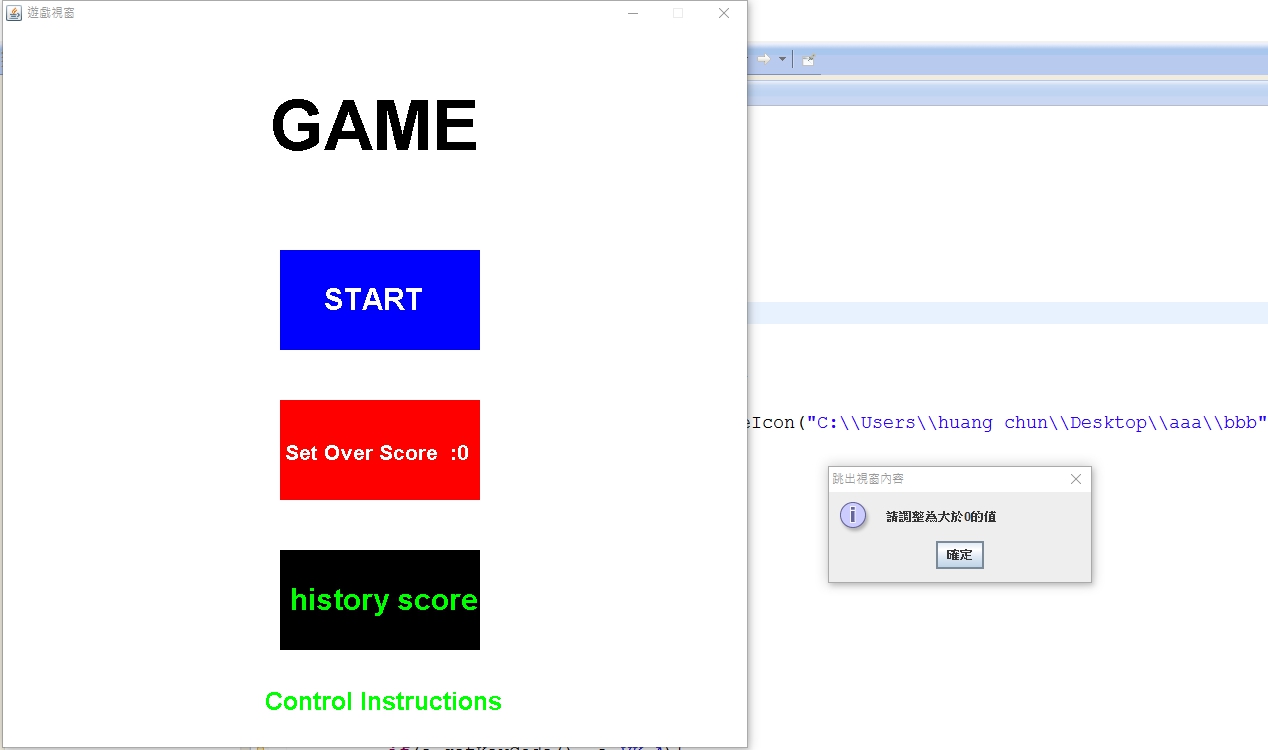
名子隨意輸入

44及55



進入遊戲主菜單

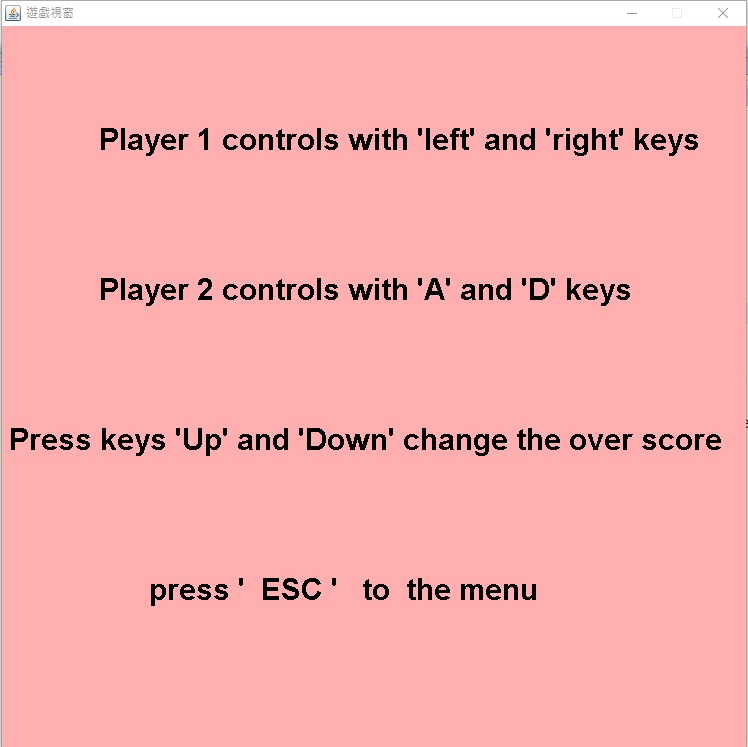
有3個功能鍵 分別是 開始，歷史遊戲分數紀錄，遊戲控制說明以及一個調整遊戲結束分數的欄位



按鍵盤上下調整遊戲結束的分數，分數沒有上限，下限為0，若低於0跳出錯誤小視窗提示分數不能低於1



歷史遊戲紀錄記分板在尚未遊戲前是空值

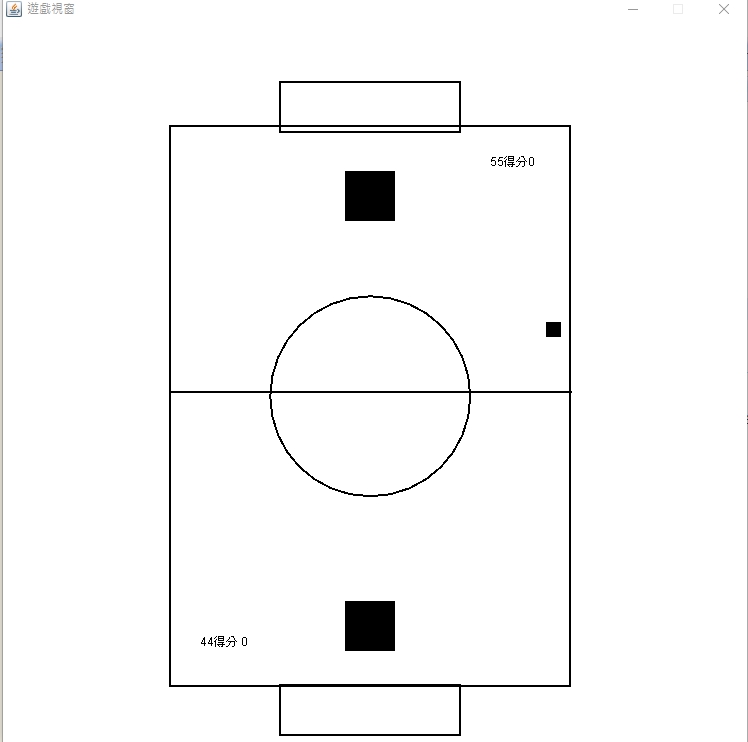


最下方(CONTROL INSTRUCTIONS)按鍵為遊戲控制說明，按下後如上圖

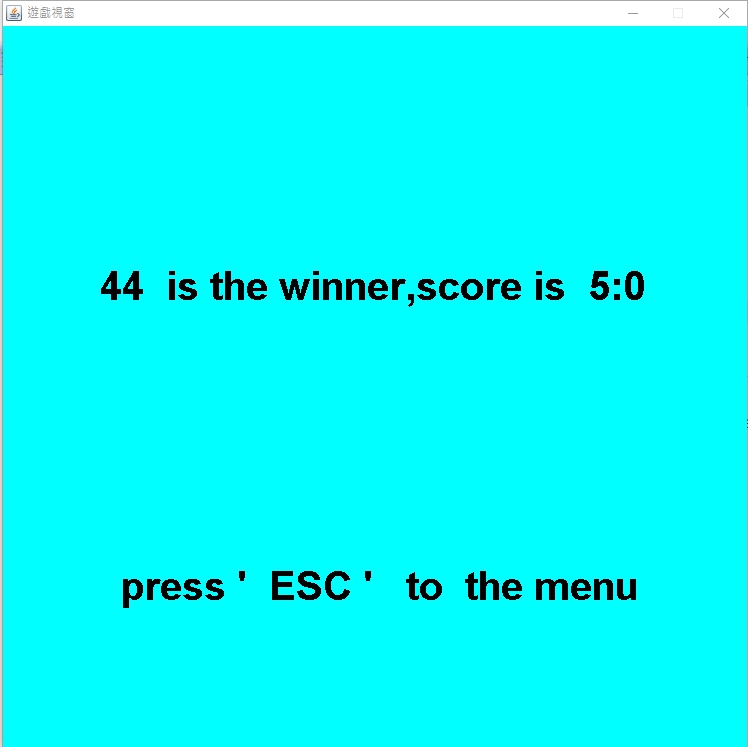
說明玩家1及玩家2控制鍵、主頁面按上下鍵調整遊戲結束分數以及跳出回到主頁面



按下開始鍵(START)後遊戲開始，右上角及左下角標示玩家名稱及玩家現在分數





44達到遊戲結束分數時獲勝，並跳出畫面，按空白鍵繼續遊戲或者按ESC回到主選單

回到主選單後點選遊戲歷史分數紀錄

紀錄為 玩家44獲勝，分數比數為5:0

按ESC鍵跳回主頁面。

以上為遊戲功能。

四、專題反思與討論-

這次專題，我們選擇寫這個小程式雖然感覺比較偏簡單類型，但是在實作上依然遇到某些小瓶頸，我們分工去把自己的部分盡量解決，再把解決不了的小問題提出討論，並一起尋求解決辦法;在整體上小問題雖然很多，也有些到現在尚未能解決，像是上面提到的，原本要新增的與電腦對戰功能，基本上已經完成但是因為會與其他功能衝突到導致其他功能失效，以我們的能力，暫時還無法解決，所以到現在只能先把這功能移除，因為時間有限，在往後可能可以完成，但我們現在還不夠精通，還在學習階段，所以我們都希望能更加進步，共勉之。

五、參考文獻

java7教學手冊 第五版 洪維恩 著

程式語言教學誌 [http://www.kaiching.org/2012/07/java-api-awt.html](https://l.facebook.com/l.php?u=http%3A%2F%2Fwww.kaiching.org%2F2012%2F07%2Fjava-api-awt.html&h=ATN68G9xO68nfk2RxR4cbF9IwB0SP7nwMP_ev35yWuBCoCgVnJ-41egvd1UyGeHhFcZO7L94ERbDahO7Ua5FHEwM1PiTurnLlgo9cWWlev71gh_Le5SBDO8BqtS40XApafJGpIEQq7jqig)

小狐狸事務所 [http://yhhuang1966.blogspot.tw/2014/02/java-swing-jframe.html](https://l.facebook.com/l.php?u=http%3A%2F%2Fyhhuang1966.blogspot.tw%2F2014%2F02%2Fjava-swing-jframe.html&h=ATN68G9xO68nfk2RxR4cbF9IwB0SP7nwMP_ev35yWuBCoCgVnJ-41egvd1UyGeHhFcZO7L94ERbDahO7Ua5FHEwM1PiTurnLlgo9cWWlev71gh_Le5SBDO8BqtS40XApafJGpIEQq7jqig)

Ping Pong 範例 [https://codereview.stackexchange.com/questions/27197/pong-game-in-java](https://l.facebook.com/l.php?u=https%3A%2F%2Fcodereview.stackexchange.com%2Fquestions%2F27197%2Fpong-game-in-java&h=ATN68G9xO68nfk2RxR4cbF9IwB0SP7nwMP_ev35yWuBCoCgVnJ-41egvd1UyGeHhFcZO7L94ERbDahO7Ua5FHEwM1PiTurnLlgo9cWWlev71gh_Le5SBDO8BqtS40XApafJGpIEQq7jqig)

java color類別建立顏色物件中英文術語對照表

<http://www.kaiching.org/2012/06/java-api-color.html>

google翻譯

<https://translate.google.com.tw/>

彈跳出小視窗輸入名子並記錄

<https://tw.answers.yahoo.com/question/index?qid=20070708000015KK06589>

跳出小視窗

<http://run26.tian.yam.com/posts/6858475>

遇到問題就找書或是上網查，一些小問題很多，所以參考的資料太多了記不清楚。