棒球遊戲



班級: 資一A

學號: 01156109

姓名: 陸建綱

**棒球遊戲**

-目錄-

[一、遊戲介紹&劇情 1](#_Toc359801686)

[二、UML圖 2](#_Toc359801687)

[(1) Player UML圖 2](#_Toc359801688)

[(2) Playert UML圖 3](#_Toc359801689)

[(3) Enter UML圖 4](#_Toc359801690)

[(4) Output UML圖 5](#_Toc359801691)

[(5) Rule UML圖 5](#_Toc359801692)

[三、主程式---Playball 7](#_Toc359801693)

[四、球員物件---Player 48](#_Toc359801694)

[五、進化物件---Playert 55](#_Toc359801695)

[六、輸入物件---Enter 57](#_Toc359801696)

[七、規則物件---Rule 70](#_Toc359801697)

[八、輸出物件---Output 82](#_Toc359801698)

# 一、遊戲介紹&劇情

遊戲名稱:棒球游戲

角色人數：9人

局數：1〜9

角色特徵：背號、名稱、防守位置、打擊棒次、打擊能力0~100、防守能力0~100

投手：球路、投球能力、各球路能力、體力1000

消耗體力: 直球0 , 曲球70 , 滑球80 , 變速球90 , 蝴蝶球100 , 指叉球110

降低打者攻擊: 直球0 ,曲球10 , 滑球15 , 變速球20 , 蝴蝶球25 , 指叉球30

一壘手、二壘手、三壘手、游擊手：接滾地球、雙殺

捕手：阻殺

左外野手、中外野手、右外野手：接殺、回傳

遊戲系統：顯示打擊名單系統、自訂選手系統、電腦自訂選手系統、選擇打擊系統、投手控制系統、阻殺系統、對抗系統、盜壘系統、防守位置設定系統、防守位置顯示系統、球路設定系統、語言對話系統、好壞球出局數系統、比分紀錄系統、經驗值累積、升級稱號、投手體力不支、擊球距離

判斷、安打判斷、失誤判斷、盜壘失誤、阻殺失誤、體力增減判斷使用球種造成失投、各種機率判斷、輸入判斷、電腦攻擊以及防守、電腦因應情況投球、跑壘推進系統、對話輸出、雙殺系統、防守失誤造成推進、各種情節合理化、球員資料庫(30位球員+特殊人物一人)、球員連續兩次打擊沒有安打獲得經驗值增加打擊能力、安打獲得經驗值

# 二、UML圖

## (1) Player UML圖

|  |
| --- |
| Player |
| # name:String |
| # number:int |
| # defend:int |
| # attack:int |
| # SticktNumber:int |
| # Location:String |
| - Pitch:String[] |
| # PitchNunber:int |
| - power =int |
| *team*:String[] |
| - *scores*:int[][] |
| - *mis*:int[][] |
| - *hit*:int[] |
| # EXP:int |
| # nonhit:int |
| # Level:int |
| # hitime:int |
| # run:int |
| # method:String[] |
| *hurt*= new int[] |
| *a*:String[] |
| *p*:int[] |
| *h*:int[] |
| Player() {} |
| Player() {name:String,number:int,defend:int,attack:int,StickNumber:int,  Location:String } |
| setscores(a:int, b:int ,c:int) :void |
| getscores():int[][] |
| getmis(a:int):int |
| setmis(a:int , b:int):void |
| gethit( a:int): int |
| sethit(a:int, b:int) :void |
| setpower(a:int) : void |
| setpower() :void |
| setname(name:String) :void |
| setnumber(number:int):void |
| setdefend(defend:int):void |
| setattack(attack:int) :void |
| setSticktNumber(SticktNumber: int):void |
| setLocation(Location:String):void |
| setPitch(Pitch:String[]):void |
| setPitchR():void |
| getname():String |
| getnumber():int |
| getdefend():int |
| getattack():int |
| getSticktNumber():int |
| getLocation():String |
| getPitch():String[] |
| getPitchNunber():int |
| bat():String[] |
| getpower():int |
| hurt(String c):void |
| setEXP(f:boolean): void |
| getEXP():int |
| setLevel():void |
| getLevel():int |
| getbataverage():double |
| gethitime():int |
| sethitime():void |
| getrun():int |
| getsup():String |
| setsup(i:int) :void |
| setrunbase():int |
| Lineout():void |

## (2) Playert UML圖

|  |
| --- |
| Playert |
| a:String[] |
| v:String[] |
| Title:String |
| run:int |
| playert(name:String, number:int,defend:int,attack:int,SticktNumber:int,Location:String,EXP:int, nonhit:int, Level: int,hitime:int,run: int) |
| getsup():String |
| setsup(i:int): void |
| lineout():void |

## (3) Enter UML圖

|  |
| --- |
| Enter |
| random:tring[] |
| all:int[] |
| j:int |
| h:int[] |
| int k = 0 |
| g:int[] |
| t: int |
| c:int[] |
| Enter() |
| nameR():String |
| numberR():int |
| testN(a:int,b: boolean): int |
| testN(a:int):int |
| testN(a:int, b:int):int |
| whichN(a:int):int |
| whichN(a:int,v: Player []):int |
| rest():void |
| StringEnter(a:String[]):String[] |
| coin():String |
| judge19(z: String):int |
| judgeYN(z: String): String |
| judgeN():int |
| judge100():int |
| judgecoin():String |
| judge01():int |
| judgeSB():String |
| judge012():int |
| judge09():int |
| judgeball(ball:String []):String |
| judge0123():int |
| judgestolen(cc: int []):int |

## (4) Output UML圖

|  |
| --- |
| Output |
| *Restore*:int |
| *y*:String |
| *f*:int |
| *c*:String[] |
| ChooseG(s:Player[]):void |
| GT(a:String,b:int):void |
| Default():void |
| ChooseGT():void |
| ChooseE ():void |
| playball():void |
| gamend ():void |
| score(a: String,b: String): void |

## (5) Rule UML圖

|  |
| --- |
| Rule |
| -TeamScore : int[] |
| -strike : int |
| -ball : int |
| -out : int |
| -hit : int |
| -mis : int |
| -Long : String |
| -Early : String |
| -set : int |
| -setnow : int |
| coin : String |
| half : int |
| has:int[]; |
| cc:int |
| game(p: Player [],enter: Enter) : void |
| Rule() |
| setLong(Long: String):void |
| setEarly(Early: String):void |
| setcoin(coin: String): void |
| setset(set:int): void |
| setTeamScoreO(etTeamScore: int): void |
| setTeamScoreI(setTeamScore: int): void |
| setout(out :int): void |
| setball(ball: int): void |
| setstrike(strike: int): void |
| setsetnow(setnow: int): void |
| sethit(hit: int): void |
| setmis(mis: int): void |
| reset():void |
| getLong() : String |
| getEarly() : String |
| getset():int |
| getTeamScoreO() : int |
| getTeamScoreI() :int |
| getout() : int |
| getball() : int |
| getsetnow() : int |
| getstrike() : int |
| gethit() : int |
| getmis() : int |
| getcoin() : String |
| count(a :int, b:String, c:String) : int |
| destination(z:int,cz: int ): double |
| GoodOrBadBall(GoodOrBadBall:int): String |
| wheretofly(a:int): String |
| wheretoground(a:int) : String |
| howmuchbaserun(a: int) : int |
| returnball( a: String) : String |
| isAmistakes(a:int, b:int ) : boolean |
| DefendmistakesO(a:int): int |
| DefendmistakesI(a:int):int |
| batchance():int |
| probability(x: Player,a:int):int |
| counstolen(a: Player, b: Player,c: int, base: int[],rule: Rule, judge: int,who: int,s: int): int |

# 三、主程式---Playball

package Game;

import java.util.Arrays;

import java.util.Scanner;

public class Playball {

/\*\*

\* @param args

\*/

// 判斷最終結果輸贏

public static void WINORLOSS(int a, int b) {

if (a > b)

System.out.println(" WIN");

else if (a < b)

System.out.println(" LOSS");

else

System.out.println(" DEUCE");

}

// 調整輸出長度

public static String adjustment(String a) {

String b = "";

if (a.length() < 5) {

for (int i = 0; i < 5 - a.length(); i++) {

b = b + " ";

}

}

a = a + b;

return a;

}

// 判斷誰先攻擊誰先防守

public static Player[] whofirst(int x, Player c[], Player p[]) {

if (x == 0)

return p;

return c;

}

// 判斷誰先攻擊誰先防守

public static Player[] whosecond(int x, Player c[], Player p[]) {

if (x == 0)

return c;

return p;

}

// 找出誰在哪裡防守

public static int whoAThere(String a, Player x[]) {

int y = 0;

for (int i = 0; i < 9; i++) {

if (x[i].getLocation().equals(a)) {

y = i;

// System.out.println(y);

break;

}

}

return y;

}

// 判斷打擊順序

public static int Sticktimes(int c, Player a[]) {

int d = 1;

for (int i = 0; i < 9; i++) {

if (a[i].getSticktNumber() == c) {

d = i;

break;

}

}

return d;

}

// 打擊位置

public static String pointname(int a) {

String f = "";

String c[] = { "內角偏低", "偏低", "外角偏低", "內測", "紅中", "外測", "內側偏高", "偏高",

"外側偏高" };

for (int i = 0; i < 9; i++) {

if (a == i) {

f = f + c[i];

break;

}

}

return f;

}

// 球種名稱

public static String pichername(String a) {

String f = "";

String b[] = { "A", "B", "C", "D", "E", "F" };

String c[] = { "直球", "曲球", "滑球", "變速球", "蝴蝶球", "指叉球" };

for (int i = 0; i < 6; i++) {

if (b[i].equals(a)) {

f = f + c[i];

break;

}

}

return f;

}

// 球種名稱

public static String pichername(int a) {

String f = "";

String b[] = { "A", "B", "C", "D", "E", "F" };

String c[] = { "直球", "曲球", "滑球", "變速球", "蝴蝶球", "指叉球" };

for (int i = 0; i < 6; i++) {

if (b[i].equals(a)) {

f = f + c[i];

break;

}

}

return f;

}

// 主程式

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner input = new Scanner(System.in);

Rule rule = new Rule();

Player[] p = new Player[9];

Player[] c = new Player[9];

System.out.print("輸入比賽局數(1~9):");

Enter enter = new Enter();

rule.setset(enter.judge19(input.next()));

System.out.print("請問是否提早結束比賽（Y/N):");

rule.setEarly(enter.judgeYN(input.next()));

System.out.print("輸入隊名：");

Player.team[0] = input.next();

// ///////////////////////////////////////////////////////////////

/\* 產生球員名單 \*/

char a = '1';

System.out.println("輸入球員姓名及編號（1~99）");

System.out.print("是否隨機產生球員名單（Y/N):");

String rn = enter.judgeYN(input.next());

for (int i = 0; i < 9; i++) {

p[i] = new Player();

if (rn.equals("N")) {

System.out.print("第" + a + "位球員姓名: ");

p[i].setname(input.next());

System.out.print(" 背號:");

p[i].setnumber(enter.testN(enter.judgeN()));

} else {

p[i].setname(enter.nameR());

p[i].setnumber(enter.numberR());

System.out

.printf("%-10s%-5d", p[i].getname(), p[i].getnumber());

System.out.println();

}

a = (char) (a + 1);

enter.all[i] = p[i].getnumber();

}

enter.c = Arrays.copyOf(enter.all, enter.all.length);

// ///////////////////////////////////////////////////////////////

/\* 設定守備位置 \*/

System.out.println("設定球員守備位置（輸入球員編號）");

System.out.print("是否隨機分配（Y/N):");

rn = enter.judgeYN(input.next());

enter.rest();

String team[] = { "投手", "一壘手", "二壘手", "三壘手", "游擊手", "左外野手", "中外野手",

"右外野手", "捕手" };

int g = 0;

for (int j = 0; j < 9; j++) {

if (rn.equals("N")) {

System.out.print("#" + team[j] + ":");

g = enter.judgeN();

int yy = enter.testN(g, 1);

int fr = enter.whichN(yy);

p[enter.whichN(yy)].setLocation(team[j]);

} else {

p[enter.testN((int) ((Math.random()) \* 100) % 9, false)]

.setLocation(team[j]);

}

}

for (int i = 0; i < 9; i++) {

System.out.printf("%-10s%-5d%-5s", p[i].getname(),

p[i].getnumber(), p[i].getLocation());

System.out.println();

}

// ///////////////////////////////////////////////////////////////

/\* 設定能力 \*/

System.out.println("能力設定 ");

System.out.print("是否隨機分配（Y/N):");

rn = enter.judgeYN(input.next());

enter.rest();

for (int j = 0; j < 9; j++) {

System.out.println(p[j].getLocation() + "(" + p[j].getnumber()

+ ")");

// ////////////////////////////////////

if (p[j].getLocation().equals("投手")) {

System.out.println("輸入請用冒號(:)隔開");

System.out.print("投手球路(直球A(預設)、曲球B、滑球C、變速球D、蝴蝶球E、指叉球F):");

String pitch[] = enter.StringEnter(input.next().split(":"));

p[j].setPitch(pitch);

}

// ////////////////////////////////////

if (rn.equals("N")) {

System.out.print("防守能力(0~100): ");

p[j].setdefend(enter.judge100());

System.out.print("打擊能力(0~100): ");

p[j].setattack(enter.judge100());

} else {

p[j].setdefend((int) (Math.random() \* 100));

p[j].setattack((int) (Math.random() \* 100));

System.out.println("防守能力(0~100): " + p[j].getdefend());

System.out.println("打擊能力(0~100): " + p[j].getattack());

}

}

// ///////////////////////////////////////////////////////////////

/\* 輸入打擊順序 \*/

enter.rest();

System.out.println("打擊順序（輸入編號）");

System.out.print("是否隨機分配（Y/N):");

rn = enter.judgeYN(input.next());

for (int i = 0; i < 9; i++) {

if (rn.equals("N")) {

System.out.print("第" + (i + 1) + "棒: ");

int g1 = enter.judgeN();

p[enter.whichN(enter.testN(g1, 1))].setSticktNumber(i + 1);

} else

p[enter.testN(1, false)].setSticktNumber(i + 1);

}

for (int i = 0; i < 9; i++) {

System.out.printf("%-5d%-10s%-5d%-5s", p[i].getSticktNumber(),

p[i].getname(), p[i].getnumber(), p[i].getLocation());

System.out.println();

}

// ///////////////////////////////////////////////////////////////

/\* 輸出使用者球員名單 \*/

System.out.println("對戰組合");

System.out.println(" " + Player.team[0]);

System.out.println("----------------------------------------");// 40

System.out

.println("棒次 背號 球員 能力（守／打) 守備 ");

System.out.println("----------------------------------------");

for (int i = 0; i < 9; i++) {

System.out.printf("%-5d%-5d%-13s%-2d/%-8d%-10s",

p[i].getSticktNumber(), p[i].getnumber(), p[i].getname(),

p[i].getdefend(), p[i].getattack(), p[i].getLocation());

if (p[i].getLocation().equals("投手")) {

for (int j = 0; j < p[i].getPitchNunber(); j++) {

System.out.print(p[i].getPitch()[j]);

}

}

System.out.println();

System.out.println("----------------------------------------");

}

// ///////////////////////////////////////////////////////////////

/\* 產生電腦球隊 \*/

enter.rest();

for (int i = 0; i < 9; i++) {

c[i] = new Player();

c[i].setname(enter.nameR());

c[i].setnumber(enter.numberR());

}

enter.rest();

for (int i = 0; i < 9; i++) {

c[enter.testN((int) ((Math.random()) \* 100) % 9, false)]

.setLocation(team[i]);

}

enter.rest();

for (int i = 0; i < 9; i++) {

c[i].setdefend((int) (Math.random() \* 100));

if (c[i].getLocation().equals("投手")) {

// System.out.print("投手球路(曲球A、直球B、滑球C、變速球D、蝴蝶球E、指叉球F):");

c[i].setPitchR();

}

}

enter.rest();

for (int i = 0; i < 9; i++) {

c[i].setattack((int) (Math.random() \* 100));

}

enter.rest();

for (int i = 0; i < 9; i++) {

c[enter.testN((int) ((Math.random()) \* 100) % 9, false)]

.setSticktNumber(i + 1);

}

// ///////////////////////////////////////////////////////////////

/\* 輸出電腦球員名單 \*/

Player.team[1] = "Computer";

System.out.println(" " + Player.team[1]

+ " ");

System.out.println("----------------------------------------");// 40

System.out

.println("棒次 背號 球員 能力（守／打) 守備 ");

System.out.println("----------------------------------------");

for (int i = 0; i < 9; i++) {

System.out.printf("%-5d%-5d%-13s%-2d/%-8d%-10s",

c[i].getSticktNumber(), c[i].getnumber(), c[i].getname(),

c[i].getdefend(), c[i].getattack(), c[i].getLocation());

if (c[i].getLocation().equals("投手")) {

for (int j = 0; j < c[i].getPitchNunber(); j++) {

System.out.print(c[i].getPitch()[j]);

}

}

System.out.println();

System.out.println("----------------------------------------");

}

enter.rest();

// ///////////////////////////////////////////////////////////////

/\* 正式遊戲部分 \*/

System.out

.println("各位喜愛棒球的棒球痴棒球狂\n大家好 打給後 胎軋後 我是徐展元 \n今天的比賽一定相當的精采！\n天母棒球場即將展開由 "

+ Player.team[0] + " VS " + "computer" + " 的賽事，預祝兩隊勝利！");

// ///////////////////////////////////////////////////////////////

/\* 用錢幣決定先攻擊或先防守 \*/

String rcoin = enter.coin();

System.out.print("投擲錢幣（正/反）：");

rule.setcoin(enter.judgecoin());

int firstorlast = 0;

if (rule.getcoin().equals(rcoin)) {

System.out.println("恭喜你猜對!!");

System.out.print("決定先攻0/後攻攻1:");

firstorlast = enter.judge01();

} else {

System.out.println("猜錯啦!!先防守吧!");

firstorlast = 1;

}

Output.playball();

// ///////////////////////////////////////////////////////////////

/\* 把兩隊放入新增物件中來判斷攻守 \*/

Player f[] = whofirst(firstorlast, c, p);

Player s[] = whosecond(firstorlast, c, p);

// ///////////////////////////////////////////////////////////////

int Sticktimes[] = { 1, 1 };

int Sticktimes2 = 1;

int runbase = 0;

int fightmethod = 0;

// ///////////////////////////////////////////////////////////////

while (rule.getsetnow() != rule.getset() + 1) {

// ///////////////////////////////////////////////////////////////////////

int restore = 0;

String UD = "";

if (Rule.half % 2 != 0) {

UD = "上";

} else {

UD = "下";

}

int base[] = new int[4];// 壘包

boolean flag = false;

// ///////////////////////////////////////////////////////////////

System.out.print("第" + (rule.getsetnow()) + "局");

System.out.println(UD);

System.out.println("============================");

// ///////////////////////////////////////////////////////////////

/\* 每一局的打擊內容 \*/

while (rule.getout() != 3) {

boolean comtinue = true;// 非常重要的判斷，判斷是否繼續每位打者的打擊

int doubleplay = 0;

boolean F = true;

// ///////////////////////////////////////////////////////////////

/\* 每一個打擊內容 \*/

int stolenrun = 0;

while (comtinue) {

runbase = 0;// 跑壘

if (flag)

System.out.println("----------------------------");

flag = true;

if (f[Sticktimes(Sticktimes[firstorlast], f)].getsup()

.equals("N")) {

} else

System.out.print(f[Sticktimes(Sticktimes[firstorlast],

f)].getsup());

// ///////////////////////////////////////////////////////////////

System.out.print("第"

+ Sticktimes[firstorlast]

+ "棒"

+ " "

+ f[Sticktimes(Sticktimes[firstorlast], f)]

.getname()

+ "(攻擊:"

+ f[Sticktimes(Sticktimes[firstorlast], f)]

.getattack()

+ " 經驗值:"

+ f[Sticktimes(Sticktimes[firstorlast], f)]

.getEXP()

+ "/100"

+ " 等級Lv"

+ f[Sticktimes(Sticktimes[firstorlast], f)]

.getLevel() + " 打擊率:");

System.out.printf("%.2f",

f[Sticktimes(Sticktimes[firstorlast], f)]

.getbataverage());

System.out.println(" 安打比:"

+ f[Sticktimes(Sticktimes[firstorlast], f)]

.getrun()

+ "/"

+ f[Sticktimes(Sticktimes[firstorlast], f)]

.gethitime() + ")");

System.out.println(Player.team[0] + rule.getTeamScoreO()

+ " : " + rule.getTeamScoreI() + "computer");

System.out.println(rule.getstrike() + "S" + " "

+ rule.getball() + "B" + " " + rule.getout()

+ "out");

Output.ChooseG(s);

int Countresult = f[Sticktimes(Sticktimes[firstorlast], f)]

.getattack() - s[whoAThere("投手", s)].getdefend();

// //////盜壘!!!記得寫/////////////////盜壘!!!記得寫/////////////盜壘!!!記得寫//////////

int stolen = 5;

int cc = 0;

String ww = "";

int defendstolen = 5;

if (base[0] != 0 && base[1] != 0 && base[2] != 0) {

System.out.println("滿壘了,還是不要冒險盜壘了,拚一支全壘打吧!!");

} else if (base[0] != 0 || base[1] != 0 || base[2] != 0) {

if (firstorlast == 0)

System.out.print("盜壘(");

else

System.out.print("阻殺(");

for (int i = 0; i < 4; i++) {

if (base[i] != 0) {

System.out.print((i + 1) + ",");// 1,12,\*123,13,2,\*23,3

ww = ww + (i + 1);

cc++;

}

}

int ccc[] = new int[ww.length() + 1];

ccc[ww.length()] = 5;

for (int i = 0; i < ww.length(); i++) {

String ss = "";

ss = ss + ww.charAt(i);

ccc[i] = Integer.parseInt(ss);

}

if (cc > 1) {

if (firstorlast == 0)

System.out.print("4全部盜壘,");

else

System.out.print("4全面阻殺,");

}

if (firstorlast == 0)

System.out.print("5不到壘");

else

System.out.print("5不進行阻殺");

if (firstorlast == 0) {

System.out.print("):");

stolen = enter.judgestolen(ccc);

} else {

System.out.print("):");

stolen = ccc[(int) (Math.random() \* 100)

% ccc.length];

// System.out.print(stolen);

defendstolen = enter.judgestolen(ccc);

// System.out.println(stolen);

if (stolen == 5)

System.out.println("看來對手並沒有要盜壘的打算,專心三振對手吧!!");

}

}

if (stolen != 5) {

if (base[0] != 0 && base[1] == 0 && base[2] == 0

&& stolen == 1) {// 1-->1壘盜壘

if (rule.counstolen(f[(enter.whichN(base[0], f))],

s[whoAThere("捕手", s)], 0, base, rule,

defendstolen, firstorlast, stolen) == 1) {

base[1] = base[0];

base[0] = 0;

System.out.println("盜上2壘");

}

} else if (base[0] != 0 && base[1] != 0 && base[2] == 0) {// 12-->1壘盜壘

if (stolen == 1 || stolen == 4) {

if (rule.counstolen(

f[(enter.whichN(base[1], f))],

s[whoAThere("捕手", s)], 1, base, rule,

defendstolen, firstorlast, stolen) == 1) {

base[2] = base[1];

base[1] = 0;

System.out.println("盜上3壘");

}

if (rule.counstolen(

f[(enter.whichN(base[0], f))],

s[whoAThere("二壘手", s)], 0, base, rule,

defendstolen, firstorlast, stolen) == 1) {

base[1] = base[0];

base[0] = 0;

System.out.println("盜上2壘");

}

} else if (stolen == 2) {

if (rule.counstolen(

f[(enter.whichN(base[1], f))],

s[whoAThere("捕手", s)], 1, base, rule,

defendstolen, firstorlast, stolen) == 1) {

base[2] = base[1];

base[1] = 0;

System.out.println("盜上3壘");

}

}

} else if (base[0] != 0 && base[1] == 0 && base[2] != 0) {// 13

if (stolen == 1) {

if (rule.counstolen(

f[(enter.whichN(base[0], f))],

s[whoAThere("捕手", s)], 0, base, rule,

defendstolen, firstorlast, stolen) == 1) {

base[1] = base[0];

base[0] = 0;

System.out.println("盜上2壘");

}

} else if (stolen == 3) {

if (rule.counstolen(

f[(enter.whichN(base[2], f))],

s[whoAThere("捕手", s)], 2, base, rule,

defendstolen, firstorlast, stolen) == 1) {

// base[3] = base[2];

base[2] = 0;

System.out.println("盜上本壘,得到1分!!");

stolenrun++;

}

} else if (stolen == 4) {

if (rule.counstolen(

f[(enter.whichN(base[2], f))],

s[whoAThere("捕手", s)], 2, base, rule,

defendstolen, firstorlast, stolen) == 1) {

// base[3] = base[2];

base[2] = 0;

System.out.println("盜上本壘,得到1分!!");

stolenrun++;

}

if (rule.counstolen(

f[(enter.whichN(base[0], f))],

s[whoAThere("二壘手", s)], 0, base, rule,

defendstolen, firstorlast, stolen) == 1) {

base[1] = base[0];

base[0] = 0;

System.out.println("盜上2壘");

}

}

} else if (base[0] == 0 && base[1] != 0 && base[2] == 0

&& stolen == 2) {// 2

if (rule.counstolen(f[(enter.whichN(base[1], f))],

s[whoAThere("捕手", s)], 1, base, rule,

defendstolen, firstorlast, stolen) == 1) {

base[2] = base[1];

base[1] = 0;

System.out.println("盜上3壘");

}

} else if (base[0] == 0 && base[1] == 0 && base[2] != 0

&& stolen == 3) {// 3

if (rule.counstolen(f[(enter.whichN(base[2], f))],

s[whoAThere("捕手", s)], 2, base, rule,

defendstolen, firstorlast, stolen) == 1) {

// base[3] = base[2];

base[2] = 0;

System.out.println("盜上本壘,得到1分!!");

stolenrun++;

}

} else if (base[0] == 0 && base[1] != 0 && base[2] != 0) {// 23

if (stolen == 2 || stolen == 4) {

if (rule.counstolen(

f[(enter.whichN(base[2], f))],

s[whoAThere("捕手", s)], 2, base, rule,

defendstolen, firstorlast, stolen) == 1) {

// base[3] = base[2];

base[2] = 0;

System.out.println("盜上本壘,得到1分!!");

stolenrun++;

}

if (rule.counstolen(

f[(enter.whichN(base[1], f))],

s[whoAThere("三壘手", s)], 1, base, rule,

defendstolen, firstorlast, stolen) == 1) {

base[2] = base[1];

base[1] = 0;

System.out.println("盜上3壘");

}

} else if (stolen == 3) {

if (rule.counstolen(

f[(enter.whichN(base[2], f))],

s[whoAThere("捕手", s)], 2, base, rule,

defendstolen, firstorlast, stolen) == 1) {

// base[3] = base[2];

base[2] = 0;

System.out.println("盜上本壘,得到1分!!");

stolenrun++;

}

} else if (base[0] != 0 && base[1] != 0

&& base[2] != 0) {// 滿壘

System.out.println("");

}

}

}

if (rule.getout() < 3) {

// //////盜壘!!!記得寫/////////////////盜壘!!!記得寫/////////////盜壘!!!記得寫//////////

// ///////////////////////////////////////////////////////////////

/\* 使用者選擇好壞球 \*/

if (firstorlast == 0)

System.out.print("預測球路（好球S/壞球B):");

else

System.out.print("選擇投球（好球S/壞球B):");

String gesBall = enter.judgeSB();

int GoodOrBadBall = (int) (Math.random() \* 100) % 10;

// ///////////////////////////////////////////////////////////////

/\* 投手投出壞球 \*/

if (gesBall.equals("B")) {

if (rule.GoodOrBadBall(GoodOrBadBall).equals("B")) {

if (firstorlast == 0)

System.out.println("選得好啊~這是一個換球");

else

System.out.println("打者選掉了這一球,配球在整密一點!");

rule.setball(rule.getball() + 1);

} else if (rule.GoodOrBadBall(GoodOrBadBall)

.equals("S")) {

if (firstorlast == 0)

System.out

.println("這是一顆好球阿!!可惜可惜放棄了一次全壘打的機會");

else

System.out.println("成功引誘打者揮棒!取得一個好球數");

rule.setstrike(rule.getstrike() + 1);

}

// ///////////////////////////////////////////////////////////////

/\* 投手投出好球 \*/

} else if (gesBall.equals("S")) {

// ///////////////////////////////////////////////////////////////

/\* 輸入好球選擇打擊方法 \*/

if (firstorlast == 0) {

System.out.print("打擊方法(");

String bat[] = f[Sticktimes(

Sticktimes[firstorlast], f)].bat();

for (int i = 0; i < bat.length; i++) {

System.out.print(bat[i] + i);

if (i < bat.length - 1)

System.out.print("/");

}

System.out.print("):");

if (bat.length == 3)

fightmethod = enter.judge012();

else

fightmethod = enter.judge0123();

} else {

// ///////////////////////////////////////////////////////////////

/\* 設定電腦打擊的邏輯 \*/

String bat[] = f[Sticktimes(

Sticktimes[firstorlast], f)].bat();

int Fpro = (int) (Math.random() \* 100) % 10;

fightmethod = rule.batchance();

if (base[0] == 0 && base[1] == 0

&& base[2] == 0) {

if (fightmethod == 1 || fightmethod == 2) {

if (Fpro > 2) {

fightmethod = 0;

}

}

}

}

// ///////////////////////////////////////////////////////////////

/\* 輸入打擊方法 \*/

if (firstorlast == 0)

System.out.print("打擊點:");

else

System.out.print("投擲點:");

int d = enter.judge09();

int cz = (int) (Math.random() \* 100) % 9;

if ((s[whoAThere("投手", s)].getpower() >= 100)) {

}

int upOrdown = d - cz;// 大於0高飛球,小於0滾地球

double D = rule.destination(d, cz);

String ballvariety = "";

int chooseVariety = 0;

// ///////////////////////////////////////////////////////////////

/\* 輸入球種 \*/

if (firstorlast == 1) {

System.out.print("選擇球種" + "(");

for (int j = 0; j < s[whoAThere("投手", s)]

.getPitchNunber(); j++) {

System.out

.print(pichername(s[whoAThere("投手",

s)].getPitch()[j])

+ ""

+ s[whoAThere("投手", s)]

.getPitch()[j]);

Player.hurt(s[whoAThere("投手", s)]

.getPitch()[j]);

System.out.print(-Player.hurt[0]);

if (j != s[whoAThere("投手", s)]

.getPitchNunber() - 1)

System.out.print(",");

}

System.out.print("):");

ballvariety = enter.judgeball(s[whoAThere("投手",

s)].getPitch());

}

// ///////////////////////////////////////////////////////////////

/\* 打擊方法選擇0 \*/

if (fightmethod == 0) {

if (firstorlast == 0) {

System.out.print("預測球種" + "(");

for (int j = 0; j < s[whoAThere("投手", s)]

.getPitchNunber(); j++) {

System.out

.print(pichername(s[whoAThere(

"投手", s)].getPitch()[j])

+ ""

+ s[whoAThere("投手", s)]

.getPitch()[j]);

Player.hurt(s[whoAThere("投手", s)]

.getPitch()[j]);

System.out.print("-" + Player.hurt[1]);

if (j != s[whoAThere("投手", s)]

.getPitchNunber() - 1)

System.out.print(",");

}

System.out.print("):");

ballvariety = enter.judgeball(s[whoAThere(

"投手", s)].getPitch());

}

int cb = (int) (Math.random() \* 100)

% s[whoAThere("投手", s)]

.getPitchNunber();

if (firstorlast == 0) {

cb = rule.probability(

s[whoAThere("投手", s)], cb);

}

// ///////////////////////////////////////////////////////////////

/\* 投手投出好球 \*/

if (rule.GoodOrBadBall(GoodOrBadBall).equals(

"S")) {

if (firstorlast == 0) {

System.out.print("投手投出了一顆"

+ pointname(cz)

+ "的"

+ pichername(s[whoAThere("投手",

s)].getPitch()[cb])

+ ",");

} else {

System.out

.print("投手投出了一顆"

+ pointname(d)

+ "的"

+ pichername(ballvariety)

+ ",");

}

// ///////////////////////////////////////////////////////////////

/\* 判斷選中球種 \*/

if (ballvariety

.equals(s[whoAThere("投手", s)]

.getPitch()[cb])) {

if (firstorlast == 0)

System.out.println("成功選中球種!!!");

else

System.out.println("無趣的配球!!!");

chooseVariety = 1;

} else {

if (firstorlast == 0)

System.out.println("猜錯啦!");

else

System.out.println("豐富的配球使對手毫無頭緒!");

}

// ///////////////////////////////////////////////////////////////

/\* 球種影響攻擊和體力 \*/

int decreaseP = 0;

int decreaseA = 0;

int notafford = 0;

String balldecrease = "";

if (firstorlast == 0)

balldecrease = s[whoAThere("投手", s)]

.getPitch()[cb];

else

balldecrease = ballvariety;

// ///////////////////////////////////////////////////////////////

/\* 球種影響攻擊和體力 \*/

Player.hurt(balldecrease);

if (s[whoAThere("投手", s)].getpower() >= Player.hurt[0]) {

decreaseP = -Player.hurt[0];

decreaseA = -Player.hurt[1];

} else

notafford = 1;

if (notafford == 0) {

s[whoAThere("投手", s)]

.setpower(decreaseP);

Countresult = Countresult + decreaseA;

if (firstorlast == 0)

System.out

.println(pichername(s[whoAThere(

"投手", s)]

.getPitch()[cb])

+ "使的投手體力"

+ decreaseP

+ "打者攻擊力"

+ decreaseA);

else

System.out

.println(pichername(ballvariety)

+ "使的投手體力"

+ decreaseP

+ "打者攻擊力"

+ decreaseA);

} else {

System.out

.println("但是投手的體力無法負荷,投成了一個軟弱無力的直球");

if (f[Sticktimes(

Sticktimes[firstorlast], f)]

.getattack() >= 50) {// 0~30

// 30

// ~60

// 60

// ~100

runbase = 4;

if (firstorlast == 0)

System.out

.println("輕鬆得到了這一支全壘打!!再多來幾顆失投球吧!");

else

System.out

.println("被對方輕鬆得到了這一支冤枉得全壘打"

+ "!!注意投手的體力!!!");

} else {

runbase = 3;

if (firstorlast == 0)

System.out.println("輕鬆得到了這一支"

+ runbase + "壘安打");

else

System.out

.println("被對方輕鬆得到了這一支"

+ runbase

+ "壘安打"

+ "!!注意投手的體力!!!");

}

rule.sethit(rule.gethit() + 1);

F = false;

comtinue = false;

}

// ///////////////////////////////////////////////////////////////

if (notafford == 0) {

if (firstorlast == 0) // 防守

System.out.println("你選擇打擊"

+ pointname(d) + "的"

+ pichername(ballvariety));

else

System.out.println("電腦選擇打擊"

+ pointname(cz)

+ "的"

+ pichername(s[whoAThere(

"投手", s)]

.getPitch()[cb]));

int isAfight = 0;

// ///////////////////////////////////////////////////////////////

/\* 判斷打擊位置以決定是否安打 \*/

if (D == 0) {

if (firstorlast == 0)

System.out

.println("完全預料到了捕手的心理!!!真強!打擊出去!");

else

System.out

.println("投球的心態完全被對手掌握了!真是糟糕!打擊出去!");

isAfight = 2;

} else if (D == 1) {

if (firstorlast == 0)

System.out

.println("快抓到了投手的投球模式!打擊出去!");

else

System.out

.println("配球模式快被猜到啦!!再小心一點!打擊出去!");

isAfight = 1;

} else if (1 < D && D < 2) {

if (firstorlast == 0)

System.out

.println("擦棒球啦!!!界外界外");

else

System.out

.println("這將是一場投打對決,會纏鬥多久呢!!!!界外界外");// "強大的尾勁將對手的球棒!!!界外界外"

if (rule.getstrike() == 2)

isAfight = -1;

} else {

if (firstorlast == 0) {

System.out

.println("揮得很大力但只揮到空氣 早安午安晚安 請回去休息");

} else {

System.out

.println("強大的尾勁將對手的球棒狠狠的甩開!!!揮棒落空!!!");

System.out

.println("揮得很大力但只揮到空氣 早安午安晚安 請回去休息");

}

}

// ///////////////////////////////////////////////////////////////

/\* 位置正確 \*/

if (isAfight == 2) {

System.out

.println("攻擊力"

+ (f[Sticktimes(

Sticktimes[firstorlast],

f)]

.getattack() + decreaseA)// 球種導致攻擊力下降

+ "-防守力"

+ s[whoAThere("投手",

s)]

.getdefend()

+ "=" + Countresult);

// ///////////////////////////////////////////////////////////////

/\* 球種正確 \*/

if (chooseVariety == 1) {// 球種正確

// ///////////////////////////////////////////////////////////////

/\* 用打者攻擊力以及投手防禦來判斷是否安打 \*/

if (Countresult >= 20) {

if (firstorlast == 0) {

System.out

.println("這一球 ");

System.out

.println("像是斷了線的風箏 ");

System.out

.println("像是變了心的女朋友 回不來了 全壘打");

} else

System.out

.println("慘阿!!這是一支全美打!!像變了新的女朋友依樣回不去啦!QQ");

runbase = 4;

rule.sethit(rule.gethit() + 1);

F = false;

comtinue = false;

} else {

runbase = rule

.howmuchbaserun(Countresult);

if (firstorlast == 0)

System.out

.println("這是一支"

+ runbase

+ "壘安打!!");

else

System.out

.println("這是一支悲劇"

+ runbase

+ "壘安打!!");

rule.sethit(rule.gethit() + 1);// 得分判斷點

F = false;

comtinue = false;

}

} else {

runbase = rule

.howmuchbaserun(Countresult);

if (firstorlast == 0)

System.out.println("這是一支+"

+ runbase

+ "+安打!!!");

else

System.out

.println("這是一支悲劇"

+ runbase

+ "壘安打!!");

rule.sethit(rule.gethit() + 1);

F = false;

comtinue = false;

}

// ///////////////////////////////////////////////////////////////

/\* 位置很近 \*/

} else if (isAfight == 1) {

// ///////////////////////////////////////////////////////////////

/\* 高飛球 \*/

if (upOrdown > 0) {

System.out

.println("攻擊力"

+ (f[Sticktimes(

Sticktimes[firstorlast],

f)]

.getattack() + decreaseA)

+ "-防守力"

+ s[whoAThere(

"投手", s)]

.getdefend()

+ "="

+ Countresult);

int wheretofly = (int) (Math

.random() \* 100) % 3;// 012

System.out

.println("這是一隻飛往"

+ rule.wheretofly(wheretofly)

+ "("

+ s[whoAThere(

rule.wheretofly(wheretofly),

s)]

.getdefend()

+ ")" + "的高飛球");// 左中右外野

if (f[Sticktimes(

Sticktimes[firstorlast],

f)].getattack() < s[whoAThere(

rule.wheretofly(wheretofly),

s)].getdefend()

|| chooseVariety == 0) {// 沒失誤

System.out

.println("由背號"

+ s[whoAThere(

rule.wheretofly(wheretofly),

s)]

.getnumber()

+ "的"

+ s[whoAThere(

rule.wheretofly(wheretofly),

s)]

.getname()

+ "接殺出局");

if (rule.getout() < 2)

if (base[0] != 0

|| base[1] != 0

| base[2] != 0)

System.out

.println(s[whoAThere(

rule.wheretofly(wheretofly),

s)]

.getname()

+ "迅速回傳給"

+ rule.returnball(rule

.wheretofly(wheretofly))

+ "壘上跑者文風不動~");

rule.setout(rule.getout() + 1);// rule.out++;

// ///////////////////////////////////////////////////////////////

/\* 失誤 \*/

} else {

runbase = rule

.DefendmistakesO(Countresult);

System.out.println(runbase);

System.out

.print("背號"

+ s[whoAThere(

rule.wheretofly(wheretofly),

s)]

.getnumber()

+ "的"

+ s[whoAThere(

rule.wheretofly(wheretofly),

s)]

.getname()

+ "發生了失誤!!");

if (firstorlast == 0)

System.out

.println("水啦!!");

else

System.out

.println(s[whoAThere(

rule.wheretofly(wheretofly),

s)]

.getLocation()

+ "是在發呆嗎!!給我認真點!");

rule.sethit(rule.gethit() + 1);// 得分判斷點

F = false;

rule.setmis(rule.getmis() + 1);

}

comtinue = false;

// ///////////////////////////////////////////////////////////////

/\* 滾地球 \*/

} else if (upOrdown < 0) {

System.out

.println("攻擊力"

+ (f[Sticktimes(

Sticktimes[firstorlast],

f)]

.getattack() + +decreaseA)

+ "-防守力"

+ s[whoAThere(

"投手", s)]

.getdefend()

+ "="

+ Countresult);

int wheretoground = (int) (Math

.random() \* 100) % 5;// 012

System.out

.println("這是一隻飛往"

+ rule.wheretoground(wheretoground)

+ "("

+ s[whoAThere(

rule.wheretoground(wheretoground),

s)]

.getdefend()

+ ")" + "的滾地球");// 一二三壘,游擊手,投手,捕手

if (f[Sticktimes(

Sticktimes[firstorlast],

f)].getattack() < s[whoAThere(

rule.wheretoground(wheretoground),

s)].getdefend()

|| chooseVariety == 0) {

System.out

.print("由背號"

+ s[whoAThere(

rule.wheretoground(wheretoground),

s)]

.getnumber()

+ "的"

+ s[whoAThere(

rule.wheretoground(wheretoground),

s)]

.getname()

+ "轉傳");

// ///////////////////////////////////////////////////////////////

/\* 判斷各種雙殺模式 \*/

if (rule.getout() < 2) {

if (base[0] != 0

&& base[1] == 0

&& base[2] == 0) {// 1壘有人--->滾地---->1,2壘雙殺(壘上無人)

if (firstorlast == 0)

System.out

.println("一,二壘完成雙殺,真是機不逢時~");

else

System.out

.println("一,二壘完成雙殺,美技美技!!每日一好球!!");

runbase = 0;

base[0] = 0;

base[1] = 0;

rule.setout(rule

.getout() + 1);

} else if (base[0] != 0

&& base[1] == 0

&& base[2] != 0) {// 1壘,3壘有人--->滾地---->1,2壘雙殺((3壘有人))

if (firstorlast == 0)

System.out

.println("一,二壘完成雙殺,真是機不逢時~");

else

System.out

.println("一,二壘完成雙殺,美技美技!!每日一好球!!");

runbase = 0;

base[0] = 0;

base[1] = 0;

rule.setout(rule

.getout() + 1);

} else if (base[0] != 0

&& base[1] != 0

&& base[2] == 0) {// 1壘,2壘有人--->滾地---->1,2壘雙殺(3壘有人)

if (firstorlast == 0)

System.out

.println("一,二壘完成雙殺,真是機不逢時~");

else

System.out

.println("一,二壘完成雙殺,美技美技!!每日一好球!!");

doubleplay = 1;

runbase = 1;

rule.setout(rule

.getout() + 1);

} else if (base[0] != 0

&& base[1] != 0

&& base[2] != 0) {// 1壘,2壘,3壘有人--->滾地---->本壘,3壘雙殺(1,2壘有人)

if (firstorlast == 0)

System.out

.println("本壘,三壘完成雙殺,真是機不逢時~少了一分!!恨阿!");

else

System.out

.println("本壘,三壘完成雙殺,阻斷了對手的氣勢,美技美技!!每日一好球!!");

doubleplay = 2;

runbase = 1;

rule.setout(rule

.getout() + 1);

} else {// 1壘封殺

if (firstorlast == 0)

System.out

.println("一壘完成封殺,跑得太慢了吧~打遠一點!!");

else

System.out

.println("一壘完成封殺,恩~守得漂亮");

runbase = 0;

base[0] = 0;

}

} else {// 1壘封殺

if (firstorlast == 0)

System.out

.println("一壘完成封殺,跑得太慢了吧~打遠一點!!");

else

System.out

.println("一壘完成封殺,水~守得漂亮");

}

rule.setout(rule.getout() + 1);// rule.out++;

comtinue = false;

// ///////////////////////////////////////////////////////////////

/\* 發生失誤 \*/

} else {

runbase = rule

.DefendmistakesI(Countresult);

System.out

.println("背號"

+ s[whoAThere(

rule.wheretoground(wheretoground),

s)]

.getnumber()

+ "的"

+ s[whoAThere(

rule.wheretoground(wheretoground),

s)]

.getname()

+ "發生了失誤!!");

if (firstorlast == 0)

System.out

.println("水啦!!");

else

System.out

.println(s[whoAThere(

rule.wheretoground(wheretoground),

s)]

.getLocation()

+ "是在發呆嗎!!給我認真點!");

rule.sethit(rule.gethit() + 1);// 得分判斷點

F = false;

rule.setmis(rule.getmis() + 1);

}

}

comtinue = false;

} else if (isAfight == 0) {// 揮棒落翁

if (firstorlast == 0)

System.out

.println("放鬆不要緊張~慢慢來~!!");

else

System.out

.println("看來這一個出局數可以輕鬆解決~哈哈");

rule.setstrike(rule.getstrike() + 1);

}

}

// ///////////////////////////////////////////////////////////////

/\* 壞球情況 \*/

} else if (rule.GoodOrBadBall(GoodOrBadBall)

.equals("B")) {

if (firstorlast == 0)

System.out

.println("投手投出了一個壞球,你揮棒落空啦!!選球仔細一點");

else

System.out

.println("打者動也不動的放掉了這一球,水啦!!");

rule.setstrike(rule.getstrike() + 1);

}

// ///////////////////////////////////////////////////////////////

/\* 選擇其他的打擊法球 \*/

} else {

if (rule.GoodOrBadBall(GoodOrBadBall).equals(

"S")) {

// ///////////////////////////////////////////////////////////////

/\* 選擇使用打擊方法1 \*/

if (fightmethod == 1) {

if (D < 2.8) {// 2\*1.414

if (rule.getout() != 2) {

if (firstorlast == 0) {

if (base[0] == 0

&& base[1] == 0

&& base[2] == 0)

System.out

.println("壘上沒人阿!!浪費一個出局數!這是什麼情況!?");

else

System.out

.println("犧牲短打成功,成功推進壘包,很好的策略");

} else {

if (base[0] == 0

&& base[1] == 0

&& base[2] == 0)

System.out

.println("壘上沒人阿!!難道這是什麼詭計嗎!?");

else

System.out

.println("對手擺出短棒,這是一隻犧牲短打,看來對手心急了");

}

rule.setout(rule.getout() + 1);// rule.out++;

comtinue = false;

runbase = 1;

} else {

System.out

.println(f[Sticktimes(

Sticktimes[firstorlast],

f)].getname()

+ "被封殺在一壘");

rule.setout(rule.getout() + 1);// rule.out++;

comtinue = false;

}

} else if (Countresult < 0) {

if (firstorlast == 0) // 防守

System.out

.println("差太多啦!!揮棒落空!!!");

else

System.out

.println("強大的尾勁將對手的球棒狠狠的甩開!!!揮棒落空!!!");

rule.setstrike(rule.getstrike() + 1);// rule.strike++;

} else if (Countresult > 0) {

if (firstorlast == 0)

System.out

.println("擦棒球啦!!!界外界外");

else

System.out

.println("這將是一場投打對決,會纏鬥多久呢!!!!界外界外");// "強大的尾勁將對手的球棒!!!界外界外"

if (rule.getstrike() != 2)

rule.setstrike(rule.getstrike() + 1);// rule.strike++;

}

// ///////////////////////////////////////////////////////////////

/\* 選擇使用打擊方法2 \*/

} else if (fightmethod == 2) {

if (D < 2.8) {// 2\*1.414

if (rule.getout() != 2) {

if (firstorlast == 0) {

if (base[0] == 0

&& base[1] == 0

&& base[2] == 0)

System.out

.println("壘上沒人阿!!浪費一個出局數!這是什麼情況!?");

else

System.out

.println("犧牲高遠球成功,成功推進壘包,很好的策略");

} else {

if (base[0] == 0

&& base[1] == 0

&& base[2] == 0)

System.out

.println("壘上沒人阿!!難道這是什麼詭計嗎!?");

else

System.out

.println("對手把球拉的很高,這是一隻犧牲高遠球,看來對手心急了");

}

runbase = 1;

rule.setout(rule.getout() + 1);// rule.out++;

comtinue = false;

} else {

System.out.println("接殺出局~");

rule.setout(rule.getout() + 1);// rule.out++;

comtinue = false;

}

} else if (Countresult < 0) {

if (firstorlast == 0) // 防守

System.out

.println("差太多啦!!揮棒落空!!!");

else

System.out

.println("強大的尾勁將對手的球棒狠狠的甩開!!!揮棒落空!!!");

rule.setstrike(rule.getstrike() + 1);// rule.strike++;

} else if (Countresult > 0) {

if (firstorlast == 0)

System.out

.println("擦棒球啦!!!界外界外");

else

System.out

.println("這將是一場投打對決,會纏鬥多久呢!!!!界外界外");// "強大的尾勁將對手的球棒!!!界外界外"

if (rule.getstrike() != 2)

rule.setstrike(rule.getstrike() + 1);// rule.strike++;

}

// ///////////////////////////////////////////////////////////////

/\* 選擇使用打擊方法3特殊打擊法 \*/

} else {

runbase = f[Sticktimes(

Sticktimes[firstorlast], f)]

.setrunbase();

f[Sticktimes(Sticktimes[firstorlast], f)]

.Lineout();

rule.sethit(rule.gethit() + 1);

F = false;

comtinue = false;

}

} else if (rule.GoodOrBadBall(GoodOrBadBall)

.equals("B")) {

if (firstorlast == 0)

System.out

.println("投手投出了一個壞球,你揮棒落空啦!!選球仔細一點");

else

System.out

.println("打者動也不動的放掉了這一球,水啦!!");

rule.setstrike(rule.getstrike() + 1);

}

}

}

if (rule.getstrike() == 3) {

if (firstorlast == 0)

System.out.println("三振出局啦!!!遜!!再接再厲");

else

System.out.println("三振出局啦!!!全場熱血沸騰(主審一個超大拉弓)");

rule.setout(rule.getout() + 1);

comtinue = false;

}

if (rule.getball() == 4) {

if (firstorlast == 0)

System.out.println("四壞球保送,真不愧是選球達人!!");

else

System.out.println("四壞球保送,難道這是什麼策略嗎!!??");

if (base[0] != 0)

runbase = 1;

else {

runbase = 0;

base[0] = f[Sticktimes(Sticktimes[firstorlast],

f)].getnumber();

}

comtinue = false;

}

} else

comtinue = false;

}

f[Sticktimes(Sticktimes[firstorlast], f)].sethitime();

// ///////////////////////////////////////////////////////////////

/\* 跑壘推進 \*/

int y = 0;

if (runbase != 0) {

for (int j = 1; j <= runbase; j++) {

for (int i = 3; i > 0; i--) {

base[i] = base[i - 1];

base[i - 1] = 0;

}

if (doubleplay == 1) {

base[0] = 0;

base[1] = 0;

}

if (doubleplay == 2) {

base[3] = 0;

base[2] = 0;

}

if (fightmethod == 1 || fightmethod == 2) {

base[0] = 0;

}

if (base[3] != 0) {

y++;

}

}

// ///////////////////////////////////////////////////////////////

/\* 上壘 \*/

if (doubleplay == 0 && fightmethod == 0 || fightmethod == 3)

base[runbase - 1] = f[Sticktimes(

Sticktimes[firstorlast], f)].getnumber();

// ///////////////////////////////////////////////////////////////

/\* 全壘打 \*/

if (runbase == 4) {

y++;

}

if (y != 0) {

System.out.println("得到了" + y + "分");

}

}

if (stolenrun != 0) {

y = stolenrun;

}

if (firstorlast == 0) {

rule.setTeamScoreO(rule.getTeamScoreO() + y);

Player.setscores(0/\* 隊伍編號 \*/, rule.getsetnow()/\* 局數 \*/, y/\* 分數 \*/);

Player.setmis(1, Player.getmis(1) + rule.getmis());

Player.sethit(0, Player.gethit(0) + rule.gethit());

} else {

Player.setscores(1/\* 隊伍編號 \*/, rule.getsetnow()/\* 局數 \*/, y/\* 分數 \*/);

rule.setTeamScoreI(rule.getTeamScoreI() + y);

Player.setmis(0, Player.getmis(0) + rule.getmis());

Player.sethit(1, Player.gethit(1) + rule.gethit());

}

// ///////////////////////////////////////////////////////////////

/\* 圖像輸出 \*/

if (rule.getout() < 3) {

for (int i = 0; i <= 3; i++) {

if (base[i] != 0) {

if (i == 3 && runbase != 0) {

System.out

.println(f[(enter.whichN(base[i], f))]

.getname() + "回到本壘得分");

} else

System.out

.println(f[(enter.whichN(base[i], f))]

.getname()

+ "在"

+ (i + 1)

+ "壘");

Output.GT(f[(enter.whichN(base[i], f))].getname(),

(i));

}

}

base[3] = 0;

Output.ChooseGT();

Output.Default();

}

// ///////////////////////////////////////////////////////////////

/\* 增加經驗值以及升級 \*/

f[Sticktimes(Sticktimes[firstorlast], f)].setEXP(F);

f[Sticktimes(Sticktimes[firstorlast], f)].setLevel();

Sticktimes[firstorlast]++;

if (Sticktimes[firstorlast] == 10)

Sticktimes[firstorlast] = 1;

rule.reset();

}

for (int i = 0; i < 2; i++) {

System.out.println();

}

// ///////////////////////////////////////////////////////////////

/\* 比分失誤安打單局呈現 \*/

System.out.println(" Run Hit Error");

System.out.println(adjustment(Player.team[0]).substring(0, 5)

+ " " + rule.getTeamScoreO() + " " + Player.gethit(0)

+ " " + Player.getmis(0));

System.out.println(adjustment(Player.team[1]).substring(0, 5)

+ " " + rule.getTeamScoreI() + " " + Player.gethit(1)

+ " " + Player.getmis(1));

// ///////////////////////////////////////////////////////////////

/\* 判斷比賽是否繼續 \*/

// 比賽進行達五局以上，雙方差距達15分，比賽提前結束。

// 比賽進行達七局以上，雙方差距達10分，比賽提前結束。

boolean gg = true;

if (rule.getEarly().equals("Y")) {

if (Math.abs(rule.getTeamScoreO() - rule.getTeamScoreI()) >= 15

&& rule.getsetnow() == 5) {

System.out.println("因為雙方在第五局分數相差15分所以比賽提前結束");

rule.setsetnow(rule.getset() + 1);

gg = false;

} else if (Math

.abs(rule.getTeamScoreO() - rule.getTeamScoreI()) >= 10

&& rule.getsetnow() == 7) {

System.out.println("因為雙方在第七局分數相差10分所以比賽提前結束");

rule.setsetnow(rule.getset() + 1);

gg = false;

}

}

if (gg) {

/\* 玩遊戲使球員獲得稱號 \*/

if (rule.getsetnow() == rule.getset() && UD.equals("下"))

System.out.println("");

else

rule.game(p, enter);

rule.reset();

rule.setout(0);

// ///////////////////////////////////////////////////////////////

/\* 攻守交換 \*/

Player D[] = f;

f = s;

s = D;

if (firstorlast == 0)

firstorlast = 1;

else

firstorlast = 0;

if (UD.equals("下"))

rule.setsetnow(rule.getsetnow() + 1);

Rule.half++;

// ///////////////////////////////////////////////////////////////

/\* 投手體力增加 \*/

if (rule.getsetnow() > 1) {

if (s[whoAThere("投手", s)].getpower() <= 700) {

int RES = (int) (Math.random() \* 1000) % 101 + 200;// 隨機產生100~300

s[whoAThere("投手", s)].setpower(RES);

Output.y = "因為休息投手體力增加" + RES;

Output.f = 1;

} else {

s[whoAThere("投手", s)].setpower();

Output.y = "因為休息投手體力回復到1000";

Output.f = 1;

}

}

if (rule.getsetnow() != rule.getset() + 1)

Output.ChooseE();

}

}

// ////////////////////////////////////////////////////////////////

/\* 輸出比賽局果 \*/

Output.gamend();

int S[][] = Player.getscores();

System.out.print(" ");

for (int i = 0; i < 9; i++) {

System.out.print((i + 1) + " ");

}

System.out.println("R H E");

System.out.print(adjustment(Player.team[0]).substring(0, 5) + " ");

for (int i = 0; i < 9; i++) {

System.out.print(S[0][i] + " ");

}

System.out.print(rule.getTeamScoreO() + " " + Player.gethit(0) + " "

+ Player.getmis(0));

WINORLOSS(rule.getTeamScoreO(), rule.getTeamScoreI());

System.out.print(adjustment(Player.team[1]).substring(0, 5) + " ");

for (int i = 0; i < 9; i++) {

System.out.print(S[1][i] + " ");

}

System.out.print(rule.getTeamScoreI() + " " + Player.gethit(1) + " "

+ Player.getmis(1));

WINORLOSS(rule.getTeamScoreI(), rule.getTeamScoreO());

}

}

# 四、球員物件---Player

package Game;

public class Player {

protected String name;

protected int number;

protected int defend;

protected int attack;

protected int SticktNumber;

protected String Location;

private String Pitch[] = new String[6];

protected int PitchNunber = 1;// 直球

private int power = 1000;// 體力

static String team[] = new String[2];

static private int[][] scores = new int[2][9];

static private int mis[] = new int[2];

static private int hit[] = new int[2];

protected int EXP = 0;

protected int nonhit = 0;

protected int Level = 1;

protected int hitime = 0;

protected int run = 0;

protected String method[] = { "一般打擊", "犧牲短打", "犧牲高遠球" };

static int hurt[] = new int[2];

static String a[] = { "A", "B", "C", "D", "E", "F" };

static int p[] = { 0, 70, 80, 90, 100, 110 };// power

static int h[] = { 0, 10, 15, 20, 25, 30 };

Player() {

}

Player(String name, int number, int defend, int attack, int StickNumber,

String Location) {

this.name = name;

this.number = number;

this.defend = defend;

this.attack = attack;

this.SticktNumber = SticktNumber;

this.Location = Location;

}

static void setscores(int a/\* 隊伍編號 \*/, int b/\* 局數 \*/, int c/\* 分數 \*/) {

scores[a][b - 1] = scores[a][b - 1] + c;

}

static int[][] getscores() {

return scores;

}

static int getmis(int a) {

return mis[a];

}

static void setmis(int a, int b) {

mis[a] = b;

}

static int gethit(int a) {

return hit[a];

}

static void sethit(int a, int b) {

hit[a] = b;

}

void setpower(int a) {

this.power = this.power + a;

}

void setpower() {

this.power = 900;

}

void setname(String name) {

this.name = name;

}

void setnumber(int number) {

this.number = number;

}

void setdefend(int defend) {

this.defend = defend;

}

void setattack(int attack) {

this.attack = attack;

}

void setSticktNumber(int SticktNumber) {

this.SticktNumber = SticktNumber;

}

void setLocation(String Location) {

this.Location = Location;

}

void setPitch(String Pitch[]) {

for (int i = 0; i < Pitch.length; i++) {

this.Pitch[i] = Pitch[i];

PitchNunber++;

}

this.Pitch[Pitch.length] = "A";// 直球

}

void setPitchR() {

int y = (int) ((Math.random()) \* 100 + 1) % 4;// 6控制電腦球種

for (int i = 0; i < y; i++) {

String c = "";

c = c + (char) (66 + (int) (Math.random() \* 100) % 5);

while (true) {

boolean flag = false;

for (int j = 0; j < y; j++) {

if (c.equals(this.Pitch[j])) {

flag = true;

break;

}

}

if (flag) {

c = "";

c = c + (char) (66 + (int) (Math.random() \* 100) % 5);

} else

break;

}

this.Pitch[i] = c;

PitchNunber++;

}

this.Pitch[y] = "A";// 直球

}

String getname() {

return this.name;

}

int getnumber() {

return this.number;

}

int getdefend() {

return this.defend;

}

int getattack() {

return this.attack;

}

int getSticktNumber() {

return this.SticktNumber;

}

String getLocation() {

return this.Location;

}

String[] getPitch() {

return this.Pitch;

}

int getPitchNunber() {

return this.PitchNunber;

}

String[] bat() {

return this.method;

}

int getpower() {

return this.power;

}

static void hurt(String c) {

for (int i = 0; i < 6; i++) {

if (c.equals(a[i])) {

hurt[0] = p[i];

hurt[1] = h[i];

}

}

}

void setEXP(boolean f) {

int R = 0;

if (f) {

nonhit++;

} else {

this.run++;// 安打數

nonhit = 0;

R = (int) (Math.random() \* 100) % 21 + 50;// 50~70

System.out.println("因為安打經驗值上升" + R + "!!");

}

if (nonhit == 2) {

R = (int) (Math.random() \* 1000) % 11 + 70;// 70~80

System.out.println("因為連續沒有上壘經驗值增加" + R + "請繼續努力~");//

}

this.EXP += R;

if (this.EXP >= 100) {

this.EXP = this.EXP - 100;

}

}

int getEXP() {

return this.EXP;

}

void setLevel() {

if (this.EXP >= 100) {

this.Level += 1;

this.EXP = 0;

System.out.println("升級啦!!!");

int up = 0;

if (this.attack >= 100) {

System.out.println("攻擊力已經升級到了100");

} else {

if (this.attack <= 50)

up = (int) (Math.random() \* 100) % 11 + 30;// 上升攻擊力30~40

else if (this.attack <= 70 && this.attack > 50)

up = (int) (Math.random() \* 100) % 11 + 20;// 上升攻擊力20~30

else if (this.attack <= 70)

up = (int) (Math.random() \* 100) % 3 + 5;// 上升攻擊力5~7

else

up = (int) (Math.random() \* 100) % 2 + 2;// 上升攻擊力2~3

}

if (up != 0)

System.out.println("攻擊力上升了" + up);

this.attack = this.attack + up;

}

}

int getLevel() {

return this.Level;

}

double getbataverage() {// 打擊率

if (this.hitime != 0)

return (double) this.run / this.hitime;

return 0;

}

int gethitime() {

return this.hitime;

}

void sethitime() {// 打擊次數

this.hitime++;

}

int getrun() {

return this.run;// 安打數

}

String getsup() {

return "N";

}

void setsup(int i) {

}

int setrunbase() {

return 0;

}

void Lineout() {

}

}

# 五、進化物件---Playert

package Game;

public class playert extends Player {

String a[] = { "一般打擊", "犧牲短打", "犧牲高遠球", "特殊打擊" };

String v[] = { "[1壘安打王]", "[2壘安打王]", "[3壘安打王]", "[全壘打王]" };

String title = "";

int run = 0;

playert(String name, int number, int defend, int attack, int SticktNumber,

String Location, int EXP, int nonhit, int Level, int hitime, int run) {// [1壘安打王]

this.name = name;

this.number = number;

this.defend = defend;

this.attack = attack;

this.SticktNumber = SticktNumber;

this.Location = Location;

this.EXP = EXP;

this.nonhit = nonhit;

this.Level = Level;

this.hitime = hitime;

this.run = run;

this.method = a;

}

String getsup() {

return title;

}

void setsup(int i) {

title = v[i];

run = i + 1;

}

int setrunbase() {

return run;

}

// ////////////////////////////////////////////////////////////////

/\* 特殊打擊 \*/

void Lineout() {

if (run == 4) {

System.out.println(this.name + "使用了特殊打擊精確地打出了一隻全壘安打!");

}

System.out.println(this.name + "使用了特殊打擊精確地打出了一隻" + run + "壘安打!");

}

}

# 六、輸入物件---Enter

package Game;

import java.util.Arrays;

import java.util.Scanner;

public class Enter {

String random[] = new String[31];

int all[] = new int[9];

Enter() {

// ////////////////////////////////////////////////////////////////

/\* 球員名單資料 \*/

random[0] = "Lin";

random[1] = "Wang";

random[2] = "Kuo";

random[3] = "Chen";

random[4] = "A-Rod";

random[5] = "Granderson";

random[6] = "Holliday";

random[7] = "Dickey";

random[8] = "Kershaw";

random[9] = "Suzuki";

random[10] = "Lee";

random[11] = "Oswalt";

random[12] = "Bautista";

random[13] = "Kinsler";

// ///////////

random[14] = "Pujols";

random[15] = "Abreu";

random[16] = "Ortiz";

random[17] = "Gonzalez";

random[18] = "Rivera";

random[19] = "Papelbon";

random[20] = "Soriano";

random[21] = "Teixera";

random[22] = "Nathan";

random[23] = "Johnson";

random[24] = "Cook";

random[25] = "Chapman";

random[26] = "Strasberg";

random[27] = "Morse";

random[28] = "Sale";

random[29] = "Cano";

random[30] = "chengangLu";

}

// ////////////////////////////////////////////////////////////////

/\* 隨機產生名字 \*/

int j = 0;

int h[] = { -1, -1, -1, -1, -1, -1, -1, -1, -1 };

String nameR() {

int a = (int) (Math.random() \* 100 % 31);

int y = 0;

while (true) {

for (int i = 0; i <= 8; i++) {

if (a == h[i]) {

y++;

}

}

if (y == 0) {

break;

} else {

a = (int) (Math.random() \* 100 % random.length);

y = 0;

}

}

h[j] = a;

j++;

return random[(int) a];

}

// ////////////////////////////////////////////////////////////////

/\* 隨機產生1~99 \*/

int k = 0;

int g[] = { -1, -1, -1, -1, -1, -1, -1, -1, -1 };

int numberR() {

Scanner input = new Scanner(System.in);

int a = (int) (Math.random() \* 100);

int y = 0;

while (true) {

for (int i = 0; i <= 8; i++) {

if (a == g[i]) {

y++;

}

}

if (y == 0 && a >= 1 && a <= 99) {// 只產生1~99

break;

} else {

a = (int) (Math.random() \* 100);

y = 0;

}

}

g[k] = a;

k++;

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷重複輸入!!!!!!!(第一次輸入)(testNumber) \*/

int t = 0;

int testN(int a, boolean b) {

Scanner input = new Scanner(System.in);

boolean flag = true;

while (flag) {

flag = false;

for (int i = 0; i < 9; i++) {

if (a == g[i]) {

flag = true;

break;

}

}

if (flag) {

a = (int) ((Math.random()) \* 100) % 9;

}

}

g[t] = a;

t++;

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷重複輸入!!!!!!!(第一次輸入)(testNumber) \*/

int testN(int a) {

Scanner input = new Scanner(System.in);

boolean flag = true;

while (flag) {

if (a >= 1 && a <= 99) {

flag = false;

for (int i = 0; i < 9; i++) {

if (a == g[i]) {

flag = true;

System.err.print(g[i] + " " + i);

break;

}

}

}

if (flag) {

System.err.print("重新輸入$: ");

a = judgeN();

}

}

g[t] = a;

t++;

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷重複輸入!!!!!!!(第一次輸入)(testNumber) \*/

int c[] = new int[9];

int testN(int a, int b) {

// for (int i = 0; i < 9; i++)

// System.out.println(c[i] + " ");

Scanner input = new Scanner(System.in);

boolean flag = true;

while (flag) {

flag = false;

for (int i = 0; i < 9; i++) {

if (a == c[i]) {

c[i] = -3;

break;

} else {

if (i == 8)

flag = true;

}

}

if (flag) {

System.err.print("重新輸入!: ");

a = judgeN();

}

}

a = testN(a);

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷是哪一個球員的號碼 \*/

int whichN(int a) {

int z = 0;

for (int i = 0; i < 9; i++) {

if (a == all[i]) {

z = i;

break;

}

}

return z;

}

int whichN(int a, Player v[]) {

for (int i = 0; i < 9; i++) {

}

int z = 0;

for (int i = 0; i < 9; i++) {

if (a == v[i].getnumber()) {

z = i;

break;

}

}

return z;

}

// ////////////////////////////////////////////////////////////////

/\* 每次呼叫完後一定要rest!!!!!!! \*/

void rest() {

for (int i = 0; i < 9; i++) {

g[i] = -1;

h[i] = -1;

}

t = 0;

j = 0;

k = 0;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入的球種(A~F) \*/

String[] StringEnter(String a[]) {

Scanner input = new Scanner(System.in);

for (int i = 0; i < a.length; i++) {

a[i] = a[i].substring(0, 1);

}

while (true) {

boolean flag = false;

if (a.length > 5) {

flag = true;

} else {

for (int i = 0; i < a.length; i++) {

for (int j = i + 1; j < a.length; j++) {

if (a[i].charAt(0) == a[j].charAt(0)) {

flag = true;

break;

}

}

}

for (int i = 0; i < a.length; i++) {

if (a[i].charAt(0) >= 'B' && a[i].charAt(0) <= 'F') {

} else {

flag = true;

break;

}

}

}

if (flag) {

System.err.print("重新輸入\*: ");

a = input.next().split(":");

for (int i = 0; i < a.length; i++) {

a[i] = a[i].substring(0, 1);

}

} else

break;

}

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入錢幣 \*/

String coin() {

String c = "";

double a = Math.random();

if (a >= 0.5)

c = "正";

else

c = "反";

return c;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入1~9 \*/

int judge19(String z) {

int c = 0;

Scanner input = new Scanner(System.in);

boolean flag = true;

while (flag) {

if (z.equals("1") || z.equals("2") || z.equals("3")

|| z.equals("4") || z.equals("5") || z.equals("6")

|| z.equals("7") || z.equals("8") || z.equals("9")) {

break;

} else {

System.err.print("重新輸入:");

z = input.next();

}

}

c = Integer.parseInt(z);

return c;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入YES/NO \*/

String judgeYN(String z) {

Scanner input = new Scanner(System.in);

boolean flag = true;

while (flag) {

if (z.equals("Y") || z.equals("N") || z.equals("n")

|| z.equals("y")) {

break;

} else {

System.err.print("重新輸入:");

z = input.next();

}

}

z = z.toUpperCase();

return z;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入是否是數字 \*/

int judgeN() {

Scanner input = new Scanner(System.in);

boolean flag1 = true;

int c = 0;

String a = "";

while (flag1) {

a = input.next();

for (int i = 0; i < a.length(); i++) {

if (a.charAt(i) >= '0' && a.charAt(i) <= '9') {

} else {

break;

}

if (i == a.length() - 1) {

flag1 = false;

break;

}

}

if (flag1)

System.err.print("重新輸入:");

}

c = Integer.parseInt(a);

return c;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入是否0~100 \*/

int judge100() {

boolean flag1 = true;

int a = judgeN();

while (flag1) {

if (a >= 0 && a <= 100) {

break;

} else {

System.err.print("重新輸入:");

a = judgeN();

}

}

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入錢幣 \*/

String judgecoin() {

Scanner input = new Scanner(System.in);

String a = input.next();

boolean flag1 = true;

while (flag1) {

if (a.equals("正") || a.equals("反")) {

break;

} else {

System.err.print("重新輸入:");

a = input.next();

}

}

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入01 \*/

int judge01() {

boolean flag1 = true;

int a = judgeN();

while (flag1) {

if (a >= 0 && a <= 1) {

break;

} else {

System.err.print("重新輸入:");

a = judgeN();

}

}

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入好壞球 \*/

String judgeSB() {

Scanner input = new Scanner(System.in);

boolean flag1 = true;

String a = input.next();

while (flag1) {

if (a.equals("S") || a.equals("B")) {

break;

} else {

System.err.print("重新輸入:");

a = input.next();

}

}

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入012 \*/

int judge012() {

Scanner input = new Scanner(System.in);

boolean flag1 = true;

int a = judgeN();

while (flag1) {

if (a == 0 || a == 1 || a == 2) {

break;

} else {

System.err.print("重新輸入:");

a = judgeN();

}

}

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入0到8 \*/

int judge09() {

Scanner input = new Scanner(System.in);

boolean flag1 = true;

int a = judgeN();

while (flag1) {

if (a == 0 || a == 1 || a == 2 || a == 3 || a == 4 || a == 5

|| a == 6 || a == 7 || a == 8) {

break;

} else {

System.err.print("重新輸入:");

a = judgeN();

}

}

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入球種 \*/

String judgeball(String ball[]) {

Scanner input = new Scanner(System.in);

boolean flag1 = true;

String a = input.next();

while (flag1) {

for (int i = 0; i < ball.length; i++) {

if (a.equals(ball[i])) {

flag1 = false;

break;

}

}

if (flag1) {

System.err.print("重新輸入:");

a = input.next();

}

}

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入0123 \*/

int judge0123() {

Scanner input = new Scanner(System.in);

boolean flag1 = true;

int a = judgeN();

while (flag1) {

if (a == 0 || a == 1 || a == 2 || a == 3) {

break;

} else {

System.err.print("重新輸入:");

a = judgeN();

}

}

return a;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷輸入盜壘 \*/

int judgestolen(int cc[]) {

boolean flag1 = true;

int c = judgeN();

while (flag1) {

for (int i = 0; i < cc.length; i++) {

if (c == cc[i]) {

flag1 = false;

break;

}

}

if (flag1) {

System.err.print("重新輸入:");

c = judgeN();

}

}

return c;

}

}

# 七、規則物件---Rule

package Game;

import java.util.Arrays;

import java.util.Scanner;

public class Rule {

private int TeamScore[] = new int[2];

private int strike = 0;

private int ball = 0;

private int out = 0;

private int hit = 0;

private int mis = 0;

private String Long = "";

private String Early = "";

private int set = 0;// 現在的局數

private int setnow = 1;// 要比賽的局數

String coin = "";

static int half = 1;

Rule() {

TeamScore[0] = 0;

TeamScore[1] = 0;

}

void setLong(String Long) {

this.Long = Long;

}

void setEarly(String Early) {

this.Early = Early;

}

void setcoin(String coin) {

this.coin = coin;

}

void setset(int set) {

this.set = set;

}

void setTeamScoreO(int setTeamScore) {

TeamScore[0] = setTeamScore;

}

void setTeamScoreI(int setTeamScore) {

TeamScore[1] = setTeamScore;

}

void setout(int out) {

this.out = out;

}

void setball(int ball) {

this.ball = ball;

}

void setstrike(int strike) {

this.strike = strike;

}

void setsetnow(int setnow) {

this.setnow = setnow;

}

void sethit(int hit) {

this.hit = hit;

}

void setmis(int mis) {

this.mis = mis;

}

void reset() {

this.ball = 0;

this.strike = 0;

this.hit = 0;

this.mis = 0;

}

String getLong() {

return this.Long;

}

String getEarly() {

return this.Early;

}

int getset() {

return this.set;

}

int getTeamScoreO() {

return TeamScore[0];

}

int getTeamScoreI() {

return TeamScore[1];

}

int getout() {

return this.out;

}

int getball() {

return this.ball;

}

int getsetnow() {

return this.setnow;

}

int getstrike() {

return this.strike;

}

int gethit() {

return this.hit;

}

int getmis() {

return this.mis;

}

String getcoin() {

return this.coin;

}

static int count(int a, String b, String c) {

return 1;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷擊球點距離 \*/

double destination(int z, int cz) {// 關鍵心臟

int x = z % 3;

int y = z / 3;

int cx = cz % 3;

int cy = cz / 3;

double D = Math.pow((x - cx) \* (x - cx) + (y - cy) \* (y - cy), 0.5);

return D;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷好壞球 \*/

String GoodOrBadBall(int GoodOrBadBall) {

if (GoodOrBadBall <= 2)

return "B";

return "S";

}

// ////////////////////////////////////////////////////////////////

/\* 判斷球往那裡飛由誰處理 \*/

String wheretofly(int a) {

String q = "";

if (a == 0) {

q = q + "左外野手";

} else if (a == 1) {

q = q + "中外野手";

} else if (a == 2) {

q = q + "右外野手";

}

return q;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷球往哪裡滾由誰處理 \*/

String wheretoground(int a) {

String q = "";

if (a == 0) {

q = q + "一壘手";

} else if (a == 1) {

q = q + "二壘手";

} else if (a == 2) {

q = q + "三壘手";

} else if (a == 3) {

q = q + "投手";

} else if (a == 4) {

q = q + "游擊手";

}

return q;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷沒有全壘打但有安打 \*/

int howmuchbaserun(int a) {// a=攻擊-防禦 0 10 20 30 40 50 60 70 80 90 100

int c = 0;

if (a >= 60 && a <= 100) {// 安打率8:2

c = 3;

} else if (a >= 30 && a < 60) {// 安打率4:6

c = 2;

} else if (a >= 0 && a < 30) {// 安打率4:6

c = 1;

} else

c = 1;

return c;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷球由誰處理 \*/

String returnball(String a) {

String r = "";

int y = (int) (Math.random() \* 100) % 2;

if (a.equals("左外野手")) {

if (y == 1)

r = r + "三壘手";

else

r = r + "游擊手";

} else if (a.equals("中外野手")) {

if (y == 1)

r = r + "二壘手";

else

r = r + "游擊手";

} else if (a.equals("右外野手")) {

if (y == 1)

r = r + "一壘手";

else

r = r + "二擊手";

}

return r;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷是否失誤 \*/

boolean isAmistakes(int a, int b) {// 打擊者攻擊a,野手防禦//攻擊-防守(野手)

if (a < b)// 出局,沒有失誤

return false;// 出局,沒有失誤

return true;// 有失誤

}

// ////////////////////////////////////////////////////////////////

/\* 判斷幾壘安打 \*/

int DefendmistakesO(int a) {// 攻擊-防守(投手)

int c = 0;

if (a >= 60 && a <= 100) {// 安打率8:2

c = 3;

} else if (a >= 30 && a < 60) {// 安打率4:6

c = 2;

} else if (a < 30) {// 安打率4:6

c = 1;

}

return c;

}

// ////////////////////////////////////////////////////////////////

/\* 判斷幾壘安打 \*/

int DefendmistakesI(int a) {// 攻擊-防守(投手)

int c = 0;

if (a >= 50) {// 安打率8:2

c = 2;

} else if (a < 50) {

c = 1;

}

return c;

}

// ////////////////////////////////////////////////////////////////

/\* 設定電腦打擊方式選擇 \*/

int batchance() {// 0 1 2 3////5

int y = 0;

int i = (int) (Math.random() \* 100) % 10;

if (i >= 0 && i <= 7)

y = 0;

else if (i == 8)

y = 1;

else

y = 2;

return y;

}

// ////////////////////////////////////////////////////////////////

/\* 設定電腦投球方式選擇 \*/

// 直球不用進來這個method

int probability(Player x, int a) {// 傷體力,體力 0 2 //1

int d = 0;

int power = x.p[a];

if (power < x.getpower()) {

d = a;

} else {

int c = (int) (Math.random() \* 100) % 10;

if (c < 2) {

d = a;

} else {

d = x.getPitchNunber() - 1;

}

}

return d;

}

// ////////////////////////////////////////////////////////////////

/\* 透過小遊戲獲得稱號--->繼承 \*/

int has[] = new int[9];

int cc = 0;

void game(Player p[], Enter enter) {// 升級!!!透過小遊戲來升級

int count = 1;

Scanner input = new Scanner(System.in);

System.out.println();

System.out.println();

System.out.println("-------------");

System.out

.print("中場休息來個小遊戲吧\n透過遊戲增加球員的特術稱號\n透過稱號增加不同的能力\n是否進行挑戰(Y/N):");

String rn = enter.judgeYN(input.next());

if (rn.equals("Y")) {

int R = (int) (Math.random() \* 100) % 100 + 1;// 50

System.out.print("<終極密碼>");

System.out.print("猜個數字:");

boolean flag = true;

int M = 100;

int m = 0;

int a = enter.judgeN();

if (R == a)

flag = false;

while (flag) {

if (a != R) {

if (a > m && a < R) {// 0~R

m = a;

}

if (a > R && a < M) {// R~100

M = a;

}

System.out.print("提示(" + m + "~" + M + "):");

a = enter.judgeN();

count++;

} else {

break;

}

}

if (cc < 9) {

System.out.println("總共猜了" + count + "次"

+ "\n系統將根據你猜的次數來\n決定你指定的球員獲得稱號的機會");

// ///////////////////////

for (int i = 0; i < 9; i++) {

enter.all[i] = p[i].getnumber();

}

enter.c = Arrays.copyOf(enter.all, enter.all.length);

System.out.print("可獲得稱號的球員:");

for (int i = 0; i < 9; i++) {

for (int j = 0; j < 9; j++) {

if (enter.c[i] == has[j])

break;

else if (j == 8)

System.out.print(enter.c[i] + " ");

}

}

System.out.println();

// //////////////////////

int rr = (int) (Math.random() \* 10);

int upper = 0;

if (rr >= 6) {// 6 7 8 9 40%

System.out.println("有獲得[1壘安打王]稱號的機會!");

upper = 0;

} else if (rr <= 5 && rr >= 3) {// 3 4 5 30%

System.out.println("有獲得[2壘安打王]稱號的機會!");

upper = 1;

} else if (rr <= 2 && rr >= 1) {// 1 2 20%

System.out.println("稀有稱號!!\n有獲得[3壘安打王]稱號的機會!");

upper = 2;

} else if (rr == 0) {// 0 10%

System.out.println("真是太幸運了!\n超級稀有稱號!!\n有獲得[全壘打王]稱號的機會!");

upper = 3;

}

System.out.println("機率:" + (100 - count \* 10) + "%");

// ////////////////////////////////

System.out.print("指定獲得編號球員:");

int g = enter.judgeN();

int yy = enter.testN(g, 1);

// ///////////////////////

// 1 2 3 4 5 6 7 8 9 10~1000>10--->機率皆為1/10

int r = (int) (Math.random() \* 100) % 10;// 0~9(10個數字)

if (count < 10) {

if (r >= count - 1) {// 機率100% 1

p[enter.whichN(yy, p)] = new playert(p[enter.whichN(yy,

p)].name, p[enter.whichN(yy, p)].number,

p[enter.whichN(yy, p)].defend, p[enter.whichN(

yy, p)].attack,

p[enter.whichN(yy, p)].SticktNumber,

p[enter.whichN(yy, p)].Location,

p[enter.whichN(yy, p)].EXP, p[enter.whichN(yy,

p)].nonhit,

p[enter.whichN(yy, p)].Level, p[enter.whichN(

yy, p)].hitime,

p[enter.whichN(yy, p)].run);

p[enter.whichN(yy, p)].setsup(upper);

System.out.println("成功的使"

+ p[enter.whichN(yy, p)].getname() + "獲得"

+ p[enter.whichN(yy, p)].getsup());

has[cc] = yy;

cc++;

} else {

System.out.println("很可惜,沒有使球員獲得稱號下次再試試吧!");

}

} else {

if (r == 0) {// 機率100% 1

p[enter.whichN(yy, p)] = new playert(p[enter.whichN(yy,

p)].name, p[enter.whichN(yy, p)].number,

p[enter.whichN(yy, p)].defend, p[enter.whichN(

yy, p)].attack,

p[enter.whichN(yy, p)].SticktNumber,

p[enter.whichN(yy, p)].Location,

p[enter.whichN(yy, p)].EXP, p[enter.whichN(yy,

p)].nonhit,

p[enter.whichN(yy, p)].Level, p[enter.whichN(

yy, p)].hitime,

p[enter.whichN(yy, p)].run);

p[enter.whichN(yy, p)].setsup(upper);// 獲得稱號!!!!!!!!!!!!!

System.out.println("成功的使"

+ p[enter.whichN(yy, p)].getname() + "獲得"

+ p[enter.whichN(yy, p)].getsup() + "升級打擊模式");

has[cc] = yy;

cc++;

} else {

System.out.println("很可惜,沒有使球員獲得稱號下次再試試吧!");

}

}

} else

System.out.println("球員都已經獲得了稱號!!good!!");

System.out.println("-------------");

} else {

System.out.println("真是太可惜了,歡銀下次再戰~");

System.out.println("-------------");

}

}

// ////////////////////////////////////////////////////////////////

/\* 盜壘判斷 \*/

int counstolen(Player a/\* 攻擊 \*/, Player b/\* 防守 \*/, int c, int base[],

Rule rule, int judge/\* 使用者猜 \*/, int who/\* 攻擊或防守 \*/, int s) {// 成功盜壘,捕手回傳但安全回壘,盜壘失敗被觸殺

int RR = 2;// 7成成功率

int HH = 4;// 盜壘失敗率,電腦6成成功

if (who == 1) {

if (judge == s) {

System.out.println("盜壘預測成功!大大的降低了對手盜壘成功率!!!");

RR = 10;// 對手盜壘成功率為2%

HH = 3;// 對手盜壘成功率為2%

} else

System.out.println("對手出乎意料的盜壘!!!");

}

int i = 0;

if (c == 2) {

System.out.println("盜本壘!!??\n看來跑者對自己的腳程很有自信!!!");

RR = 6;// 3成成功率,盜本壘

}

int R = (int) (Math.random() \* 100) % 10;

if (a.getattack() - b.getdefend() > 0) {// 7成機會成功

if (R > RR) {

System.out.println(a.getname() + "成功盜壘");

i = 1;

} else {

System.out

.println("非常的接近!!!\n這到底是主審的誤判還是一個漂亮得美技呢!?\n什麼!居然是盜壘失敗!!!");

rule.setout(rule.getout() + 1);

}

} else if (a.getattack() == b.getdefend()) {

System.out.println("捕手" + b.getname() + "進行阻殺但" + a.getname()

+ "安全回壘");

i = 2;

} else if (a.getattack() - b.getdefend() < 0) {// 6成績會失敗

if (R > HH) {

System.out.println(a.getname() + "盜壘失敗" + b.getLocation() + " "

+ b.getname() + "阻殺成功");

i = 3;

base[c] = 0;

rule.setout(rule.getout() + 1);

} else {

System.out

.println(a.getname()

+ "拚了命的撲上了壘包!\n是safe還是out!?\n上壘了上了上壘了!!!\nwhat an amazing run!!!");

rule.setout(rule.getout() + 1);

i = 1;

}

}

return i;

}

}

# 八、輸出物件---Output

package Game;

public class Output {

static int restore = 0;

static String y = "";

static int f = 1;

static String c[] = { "/", "/", "/", "/" };

public static int whoAThere(String a, Player x[]) {

int y = 0;

for (int i = 0; i < 9; i++) {

if (x[i].getLocation().equals(a)) {

y = i;

break;

}

}

return y;

}

static void ChooseG(Player s[]) {

System.out.print("投手" + s[whoAThere("投手", s)].getname() + "(體力"

+ s[whoAThere("投手", s)].getpower() + ",防守"

+ s[whoAThere("投手", s)].getdefend() + ")");// 投手LEO(體力45,防守32)

if (f == 1) {

if (y.equals("")) {

} else

System.out.println(y);

f++;

} else

System.out.println();

if (s[whoAThere("投手", s)].getpower() < 100) {

restore++;

}

if (restore == 4) {

System.out.println("投手的情況相當不穩定,教練喊出了暫停");

int a = (int) (Math.random() \* 1000) % 301 + 300;// 隨機產生300~700

s[whoAThere("投手", s)].setpower(a);

System.out.println("因為教練的鼓舞投手體力增加" + a);

restore = 0;

}

System.out.println("-------------");

System.out.println("| 6 | 7 | 8 |");

System.out.println("-------------");

System.out.println("| 3 | 4 | 5 |");

System.out.println("-------------");

System.out.println("| 0 | 1 | 2 |");

System.out.println("-------------");

}

static void GT(String a, int b) {

String x = "";

x = x + a.charAt(0);

c[b] = x;

}

static void Default() {

c[0] = "/";

c[1] = "/";

c[2] = "/";

c[3] = "/";

}

static void ChooseGT() {

System.out.println(" [" + c[1] + "]2 ");

System.out.println(" ");

System.out.println("3[" + c[2] + "] [" + c[0] + "]1 ");

System.out.println(" ");

System.out.println(" [" + c[3] + "]4 ");

}

static void ChooseE() {

System.out.println();

System.out.println();

System.out.println("----------------");

System.out.println("| Change Sides |");

System.out.println("----------------");

System.out.println();

System.out.println();

}

static void playball() {

System.out.println();

System.out.println();

System.out.println("----------------");

System.out.println("| Play Ball |");

System.out.println("----------------");

System.out.println();

System.out.println();

}

static void gamend() {

System.out.println();

System.out.println();

System.out.println("----------------");

System.out.println("| Game End |");

System.out.println("----------------");

System.out.println();

System.out.println();

}

static void score(String a, String b) {

}

}