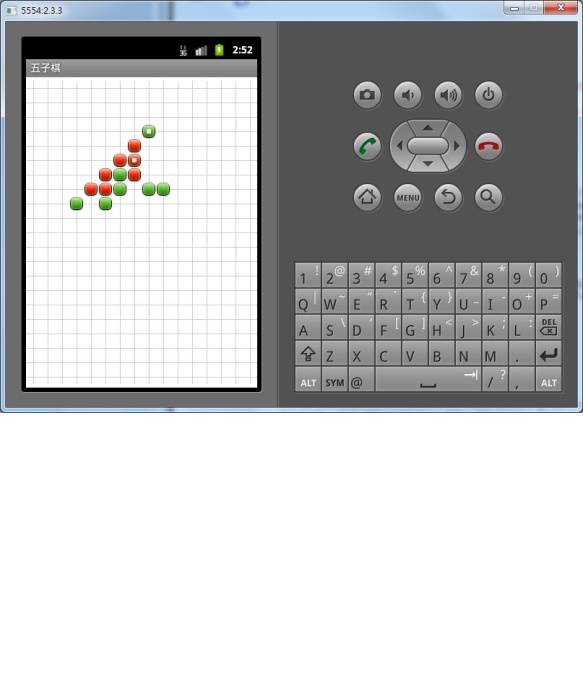
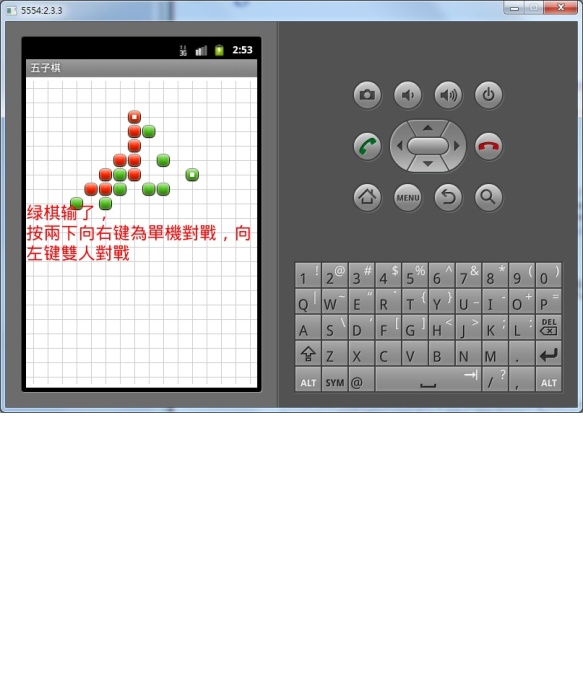
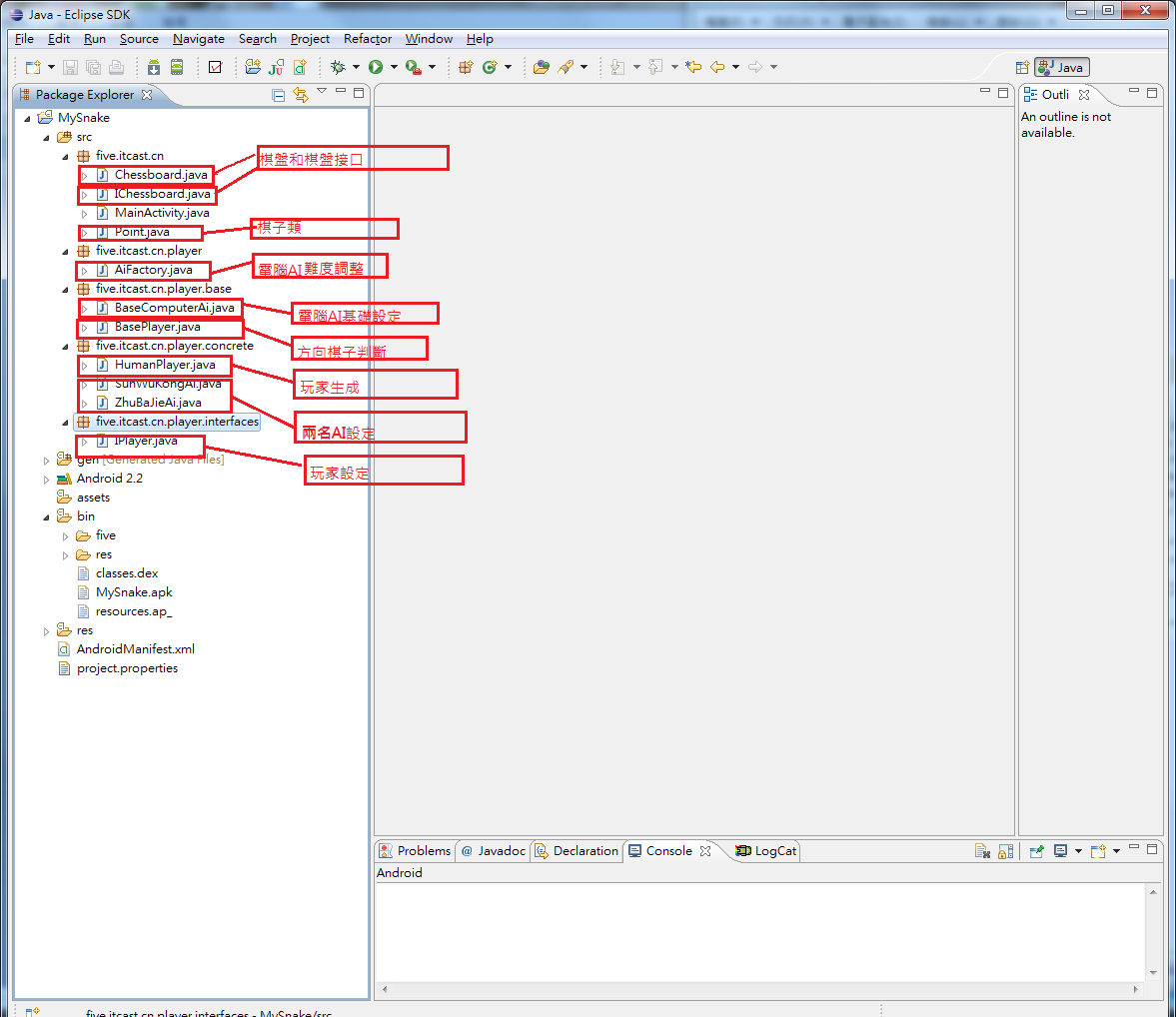
**題目:五子棋  
內容:誰先連續下5子即獲勝  
目標:瞭解裡面程式碼  
參考網址:** **http://apk.tw/thread-2324-1-1.html  
**

****

**(2) 運用哪些工具、設計技巧(資料結構? 演算法?)與重要模組(套件、類別) ...  
-- 分別以實際程式碼來舉例說明  
 重要模組:棋盤和棋盤接口、棋子類、電腦AI難度調整、電腦AI基礎設定、方向棋子判斷、玩家生成、兩名AI設定、玩家設定**//遊戲狀態常量：  
//已準備好，可開始  
private static final int READY = 1;  
//已開始  
private static final int RUNNING = 2;  
//已結束  
private static final int PLAYER\_TWO\_LOST = 3;  
private static final int PLAYER\_ONE\_LOST = 4;

###### (3) 更新-工作項目(必須細分到可衡量各項工作成果)、時程安排與分工狀況(負責) BN98061棋盤和棋盤接口90% 進行中 //兩個玩家 //第一個玩家默認為人類玩家 private IPlayer player1 = new HumanPlayer(); //第二個則根據選擇人機戰還是對戰模式來初始化 private IPlayer player2; //預先初始兩個第二玩家 //電腦玩家 private static final IPlayer computer = AiFactory.getInstance(2); //人類玩家 private static final IPlayer human = new HumanPlayer(); // 所有未下的空白點 private final List<Point> allFreePoints = new ArrayList<Point>(); public Chessboard(Context context, AttributeSet attrs) { super(context, attrs); setFocusable(true); /畫棋盤 private List<Line> lines = new ArrayList<Line>(); private void drawChssboardLines(Canvas canvas){ for (Line line : lines) { canvas.drawLine(line.xStart, line.yStart, line.xStop, line.yStop, paint); //線類 class Line{ float xStart,yStart,xStop,yStop; public Line(float xStart, float yStart, float xStop, float yStop) { this.xStart = xStart; this.yStart = yStart; this.xStop = xStop; this.yStop = yStop; BN98072電腦AI難度調整90% 進行中 //電腦AI工廠類 public class AiFactory { private final static Map<Integer,IPlayer> ais = new HashMap<Integer, IPlayer>(2); //工廠方法，數字越大，難度越高 public static IPlayer getInstance(int level){ IPlayer ai = ais.get(level); if(ai==null){ switch (level) { case 1: ais.put(level, new ZhuBaJieAi()); break; case 2: ais.put(level, new SunWuKongAi()); break; } } return ais.get(level); } } BN98074電腦AI基礎設定90% 進行中

###### 演算法核心類,演算法的主體思想分三個步驟, //第一步:根據雙方的當前的形勢迴圈地假設性的分別給自己和對方下一子(在某個範圍內下子),並判斷此棋子能帶來的形勢上的變化,如能不能沖4,能不能形成我方或敵方雙3等, //第二步:根據上一步結果,組合每一步棋子所帶來的所有結果(如某一步棋子可能形成我方1個活3,1個沖4(我叫它半活4)等),包括敵方和我方的。 //第三步:根據使用者給的規則對上一步結果進行排序,並選子(有進攻形、防守形規則)

###### /標示分析結果當前點位是兩頭通（ALIVE）還是只有一頭通（HALF\_ALIVE），封死的棋子分析過程自動屏蔽，不作為待選棋子 private static final int ALIVE = 1; private static final int HALF\_ALIVE = 0; //private static final int DEAD = -1; //計算範圍，太大的範圍會有性能問題 private class CalcuteRange{ int xStart,yStart,xStop,yStop; private CalcuteRange(int xStart, int yStart, int xStop, int yStop) { this.xStart = xStart; this.yStart = yStart; this.xStop = xStop; this.yStop = yStop;

**(4) 本周個人/團隊工作成果(重點)紀錄**

**剛開始執行遊戲時，發現可以讓遊戲變得更完美，雖然這是參考別人所寫的，但我們目前在程式的執行過程上慢慢著墨，每一行的程式碼意義**

**(5) 遭遇問題與解決方法(或處理方案與進度)  
剛開始解讀每個SRC裡的JAVA檔時遇到了不知道這是代表著程式中哪一區塊，所以開始尋找每個JAVA檔中關鍵字眼和靠著網路收尋的力量慢慢來解釋。**