**Java跳棋報告書**

2019/ 1 / 3

目錄

[前言 1](#_Toc534797777)

[一、程式運作 1](#_Toc534797778)

[1.1 Socket 1](#_Toc534797779)

[1.1.1實作Server端 1](#_Toc534797780)

[1.1.2實作Client端 2](#_Toc534797781)

[二、系統文件 2](#_Toc534797782)

[2.1 系統流程圖 2](#_Toc534797783)

[2.2 程式分類列表 2](#_Toc534797784)

[2.2.1 checker.java 伺服端 2](#_Toc534797785)

[2.2.2 SocketClient.java 3](#_Toc534797786)

[三、技術文件 3](#_Toc534797787)

[3.1 系統環境 3](#_Toc534797788)

[3.2 環境設定 5](#_Toc534797789)

[3.2.1 網路設定 5](#_Toc534797790)

[3.2.2 防火牆設定 5](#_Toc534797791)

[四、使用者文件 6](#_Toc534797792)

[4.1 操作流程 6](#_Toc534797793)

[五、程式分類列表 9](#_Toc534797794)

[5.1 checker.java 9](#_Toc534797795)

[5.2 SocketClient.java 10](#_Toc534797796)

[5.3 board.java 11](#_Toc534797797)

[5.4 win.java 12](#_Toc534797798)

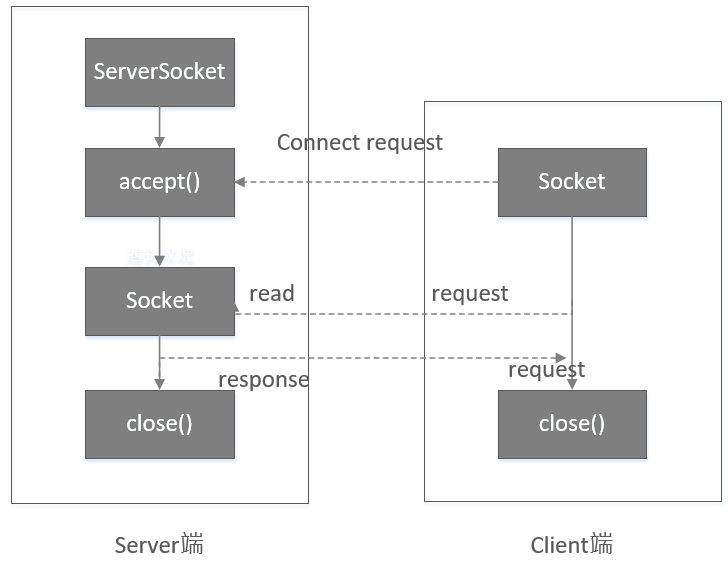
[附錄 13](#_Toc534797799)

[附錄一、程式碼 13](#_Toc534797800)

# 前言

 以 Socket 為基礎，在兩台電腦之間建立網路連線，透過網路溝通的端點，傳送與接收資料。欲建立 Socket 連線，需要IP位址及通訊埠(port)。兩台電腦同時擔任server端和client端。

# 一、程式運作



(圖x) Socket架構圖

## Socket

程式與網路之間的介面。Server端和Client端建立Server socket，連線並傳送資料，一直到雙方都關閉Socket。

## 1.1.1實作Server端

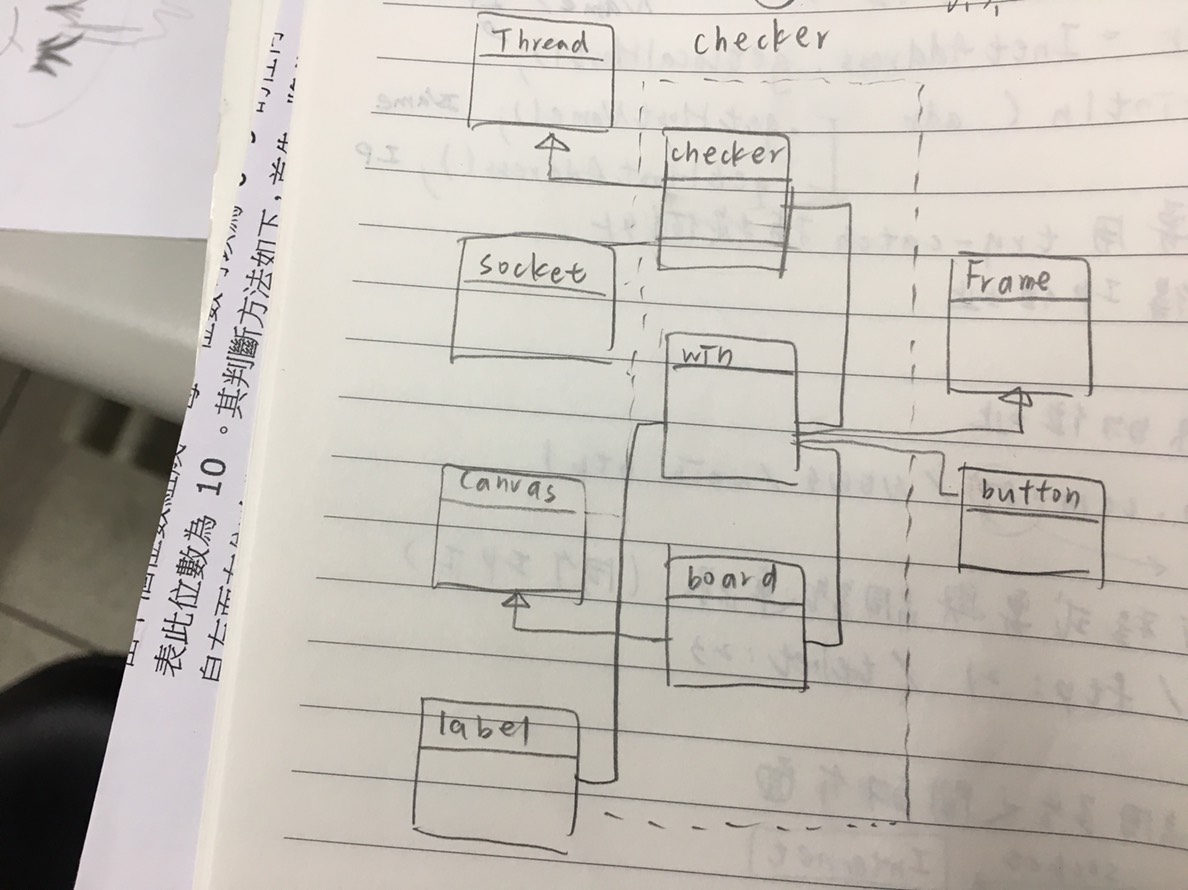
伺服端程式使用 ServerSocket 類別，在物件 server 執行accept()方法監聽客戶端的連線請求，當接受到連線請求，accept()方法傳回socket物件，利用該物件建立連線串流，當連線建立就呼叫 close() 關閉端口。

## 1.1.2實作Client端

客戶端使用socket類別。利用connect()方法與伺服端進行連線，將資料寫入socket，傳回資料至指定連接之IP位址。

# 二、系統文件

## 2.1 系統流程圖



(圖x) 程式架構圖

### 2.2 程式分類列表

## 2.2.1 checker.java 伺服端

用以等待及接收客戶端傳來的資料。

in = **new** java.io.BufferedInputStream(socket.getInputStream())

在socket執行getInputStream()方法，取得輸入之字串流。

socket.setSoTimeout(0)

可取得或設定執行accept()方法的逾時時間，單位為毫秒(ms)。預設值為 0，代表不會逾時，永遠等待。

socket.getInetAddress())

取得客戶端之IP位址並印出。

## 2.2.2 SocketClient.java

將資料傳至伺服端

DataOutputStream out = **new** DataOutputStream(client.getOutputStream())在socket執行getOutputStream()方法，取得輸出之字串流，並建立類別 DataOutputStream 型態之物件 out，將輸出串流轉換為物件 out。

# 三、技術文件

## 3.1 系統環境

|  |  |  |
| --- | --- | --- |
| 項目 | 名稱及版本 | 備註 |
| 開發平台 | Eclipse |  |
| 應用程式 | JVM | Java虛擬機 |
| 連線通訊埠 |  | 兩台電腦之Port : 8765 |

## 3.2 環境設定

### 3.2.1 網路設定

本遊戲利用虛擬IP在區域網路內連線。兩台電腦需先連至同一網路下，才可進行連線。

### 3.2.2 防火牆設定

如果確定網路線已設定好，還是出現錯誤訊息，那就代表是「Windows防火牆」的問題。依照以下步驟關閉防火牆 :

進入「控制台」點選「Windows 防火牆」選擇左方的｢進階設定｣，進入後點選左方的｢輸入規則｣ :

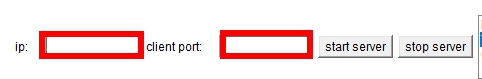
有｢私人｣、｢網域｣、｢公用｣三種設定檔，如果網路是設定｢家用網路｣或｢工作場所網路｣，那就選｢私人｣這個設定檔；如果是設定｢公用網路｣，那就選｢公用｣。

# 四、使用者文件

# 4.1 操作流程

4.1.1**輸入對方IP位址及通訊埠Port**

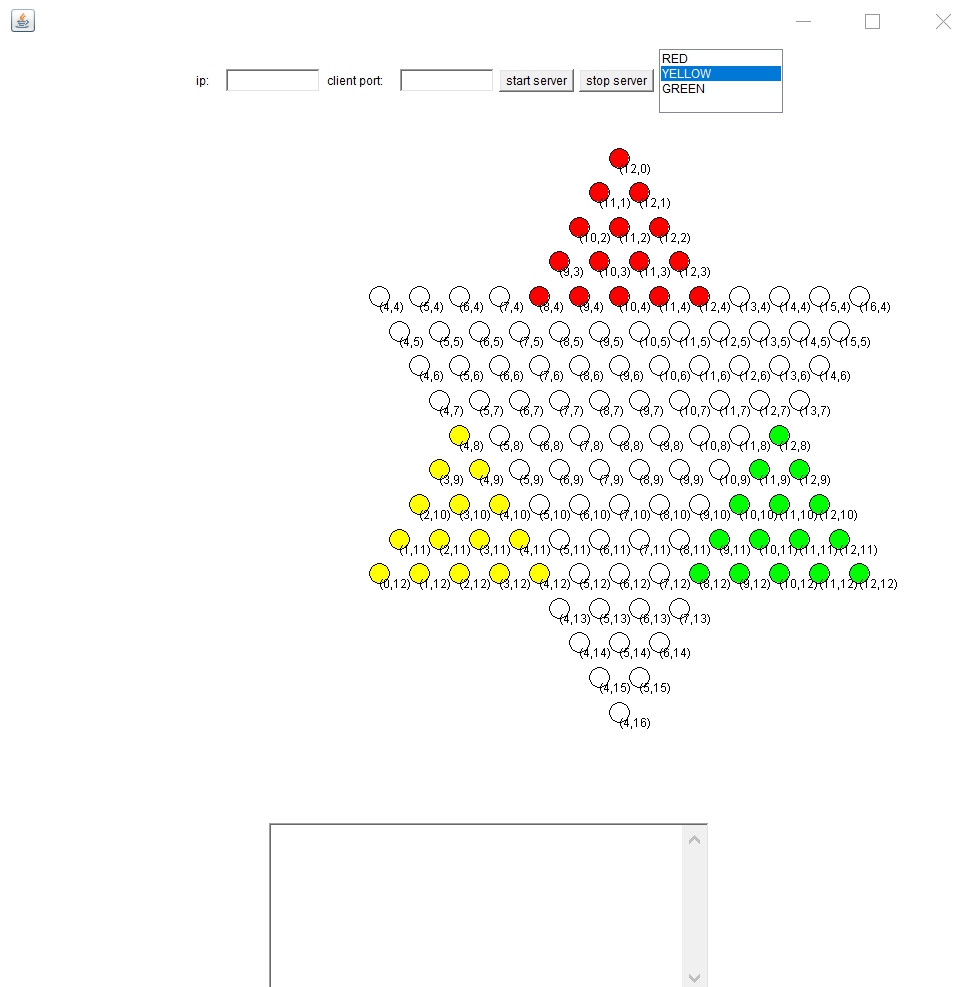
開啟命令題示字元 ( cmd ) 輸入ipconfig可查詢自己電腦的虛擬IP位址。



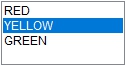
(圖x)IP位址及通訊埠Port輸入框

4.1.2**選擇棋子顏色**

遊戲未進行時，有三個顏色供玩家選擇，當一接收到對方的棋子顏色，即消掉未使用到的顏色棋子。



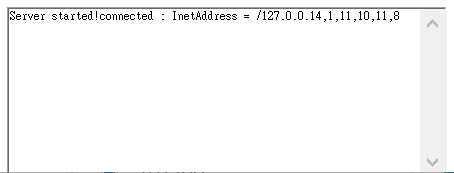
(圖x)開始頁面

****

(圖x) 玩家雙擊欲選之顏色，即成功選取

4.1.3**連線**

點擊start server按鈕，進行連線，下方對話框顯示是否成功連線。

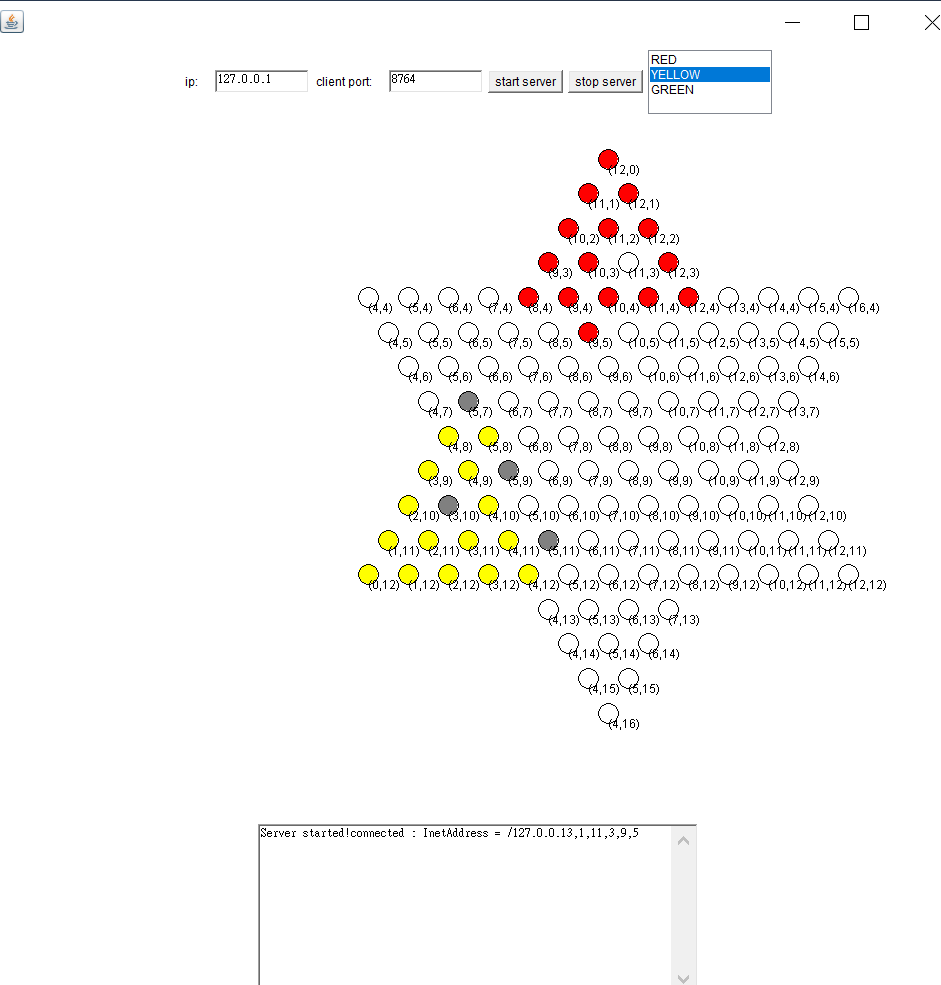


(圖x)當連線成功，即顯示對方IP位址及棋子移動資料

(註: 127.0.0.1為IP位址 ; 4，1，11，10，11，8分別為玩家棋子顏色，移動步數，從 (10,11) 移動到 (11,8)

4.1.4**下棋**

雙方輪流下棋，以滑鼠拖移方式移動棋子，點擊自己顏色的棋子，將顯示可移動範圍，過程中不可移動別人的棋子。

****

(圖x)以灰色棋子顯示可移動範圍

****

(圖x)還未輪到你時，不可移動下一步

****

(圖x)遊戲過程中，不可移動別人的棋子

# 五、程式分類列表

* checker.java 伺服端 (主要執行檔)
* SocketClient.java 客戶端

UI介面

* board.java 畫布重整
* win.java主要顯示介面

## 5.1 checker.java

|  |  |  |
| --- | --- | --- |
| **變數名稱** | **型別** | **備註** |
| Outserver | boolean | 預設值為false。當等於true即代表連線成功 |
| ServerPort | int | 伺服端的通訊埠 (Port) |
| socket | Socket() | Socket()類別中，用以取得或設定伺服端物件狀態 |
| myturn | boolean | 預設值為true。此玩家先下或是對方移動完棋子，輪到此玩家下棋。每次此玩家移動完即改為false |
| redwon | Boolean | 預設值為false。紅色棋子獲勝 |
| yellowwon | boolean | 預設值為false。黃色棋子獲勝 |
| greenwon | boolean | 預設值為false。綠色棋子獲勝 |
| strbuf | String | 顯示在視窗上的字串 |
| in | BufferedInputStream() | 字串緩衝流。取得輸入之字串流，將輸入串流轉換為物件in |
| data | String | 存放輸入字串流的資料 |
| A | String | 以逗號分割接收的data資料，以字串形式存入陣列中 |
| command | ArrayList() | 將分割過後的資料轉成int型態，透過add()方法存入list中 |
| out | DataOutputStream() | 將資料編寫並送出 |

## 5.2 SocketClient.java

|  |  |  |
| --- | --- | --- |
| **變數名稱** | **型別** | **備註** |
| port | int | 伺服端的通訊埠 (Port) |
| address | string | 伺服端的IP位址 |
| client | Socket() | Socket 類別中，用以取得或設定連線狀態的方法 |
| out | DataOutputStream() | 將資料編寫並送出 |
| ijstring | string | 和伺服端連線時，將玩家移動棋子的字串資料傳入 |

## 5.3 board.java

|  |  |  |
| --- | --- | --- |
| **欄位** | **型別** | **備註** |
| A | int | 以二維陣列顯示棋盤 |
| goal | int | 三維陣列表示三顏色需到達的棋盤位置，以判斷輸贏 |
| l | double |  |
| dx | double | 傳回正弦函數值 |
| dy | double | 傳回餘弦函數值 |
| r | int |  |
| mark | int | 全域變數。棋子的顏色 |
| i\_old | int | 棋子移動前的x軸座標 |
| j\_old | int | 棋子移動前的y軸座標 |
| b | Board() |  |
| app | checker() |  |
| w | Win() |  |
| move\_jump | int | 計算棋子連跳的二維陣列(上下左右各加2) |
| move\_step | int | 計算棋子走一步的二維陣列(上下左右各加1) |
| yellowcount | int | Goal[][][]陣列上的黃色棋子數 |
| redcount | int | Goal[][][]陣列上的紅色棋子數 |
| greencount | int | Goal[][][]陣列上的綠色棋子數 |
| c | Color | 一維陣列。共有五個顏色，按陣列索引直為白色、黃色、紅色、綠色、灰色 |

## 5.4 win.java

|  |  |  |
| --- | --- | --- |
| **欄位** | **型別** | **備註** |
| mcolor | string | 存放回傳的字串 |
| List\_color | List |  |
| socketText | TextArea | 對話框顯示的字 |
| mapcolor | HashMap | 利用put()新增元素，再用key – value取得顏色的值 |
| mycolor | int | 將回傳的字串轉成int |