OOP期末專題〈坦克大戰(射擊遊戲)〉書面報告

第14組：黃宇晨、林以鑫、高璿哲

1. 設計內容

我們所製作的射擊遊戲坦克大戰是以物件導向程式設計為基礎，設計許多class如battlefield, player, monster等等，以完成射擊、殺怪物、得分、陷阱等遊戲特效。

Battlefield在遊戲開始時會被建立，它包含三個member：player, bullet, monster, and trap。

Player的功能是紀錄目前玩家所在位置、印出玩家所在位置以及判斷player和bullet, monster, and trap之間的關聯，bullet應朝向player面對的方向擊發，player被monster碰到時應被擊退且損血，player走到trap的位置應被限制移動一段時間。

Bullet的功能是玩家開槍時生成子彈紀錄子彈的位置、印出子彈位置以及判斷bullet和monster, player之間的關聯，bullet打到monster時，bullet和monster應消失，player的得分上升。

Monster的功能是有怪物死掉時生成新怪物，紀錄怪物的位置、印出怪物位置以及判斷monster和player, bullet之間的關聯。

(預設: 怪物最多六隻同時在場；玩家一位；子彈數最多六發同時在場。)

Trap 的功能是有陷阱被觸發過後生成新陷阱，紀錄陷阱的位置、印出陷阱位置以及判斷trap和player之間的關聯。此陷阱觸發需停留在陷阱上一定時間，且須相應時間之後才會解除狀態。

1. 程式架構設計

Health(血量)

(繼承)

monster(怪物)

bullet(子彈)

Player(玩家)

trap(陷阱)

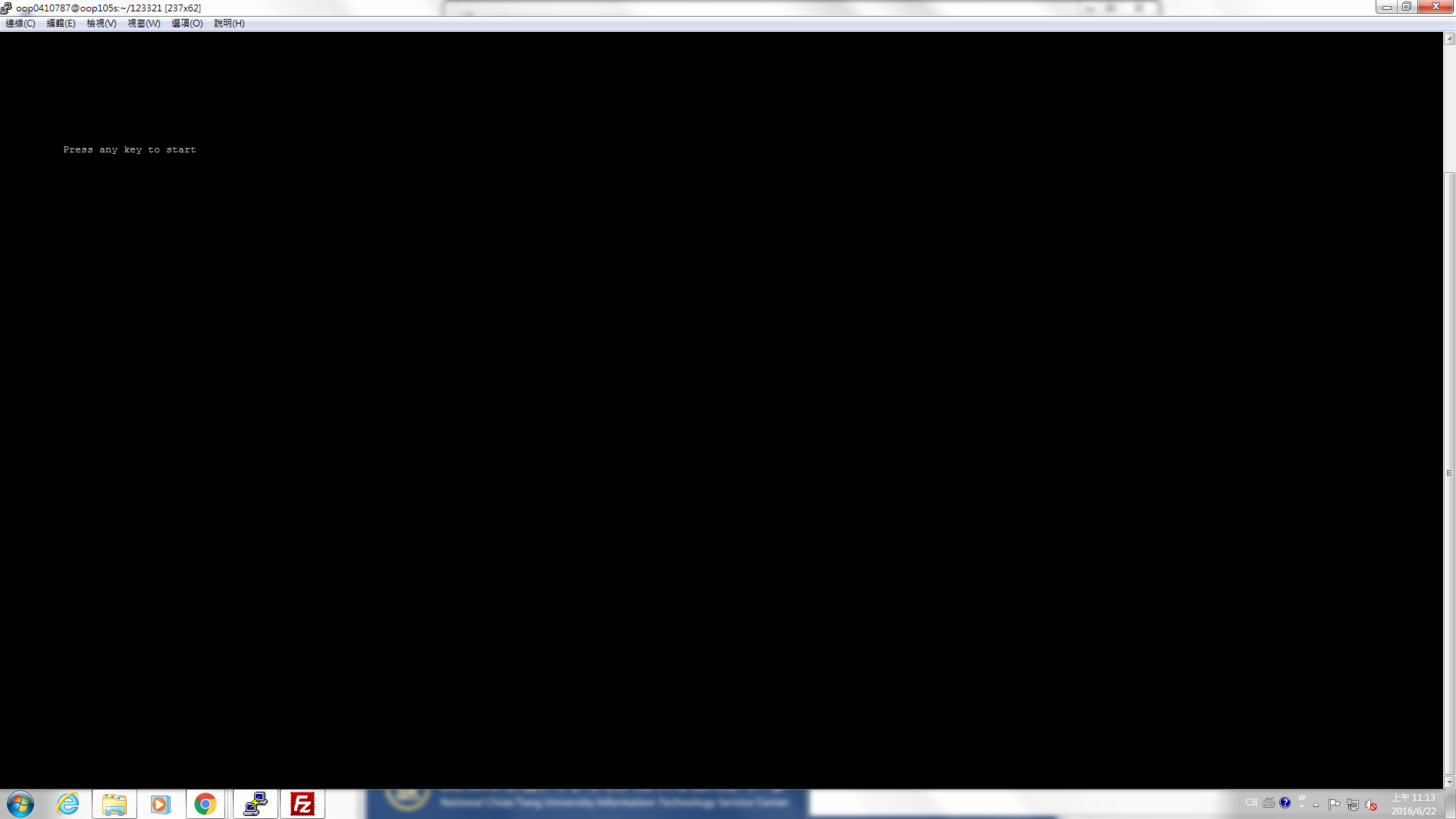
(used as member)

battlefield(遊戲戰場)

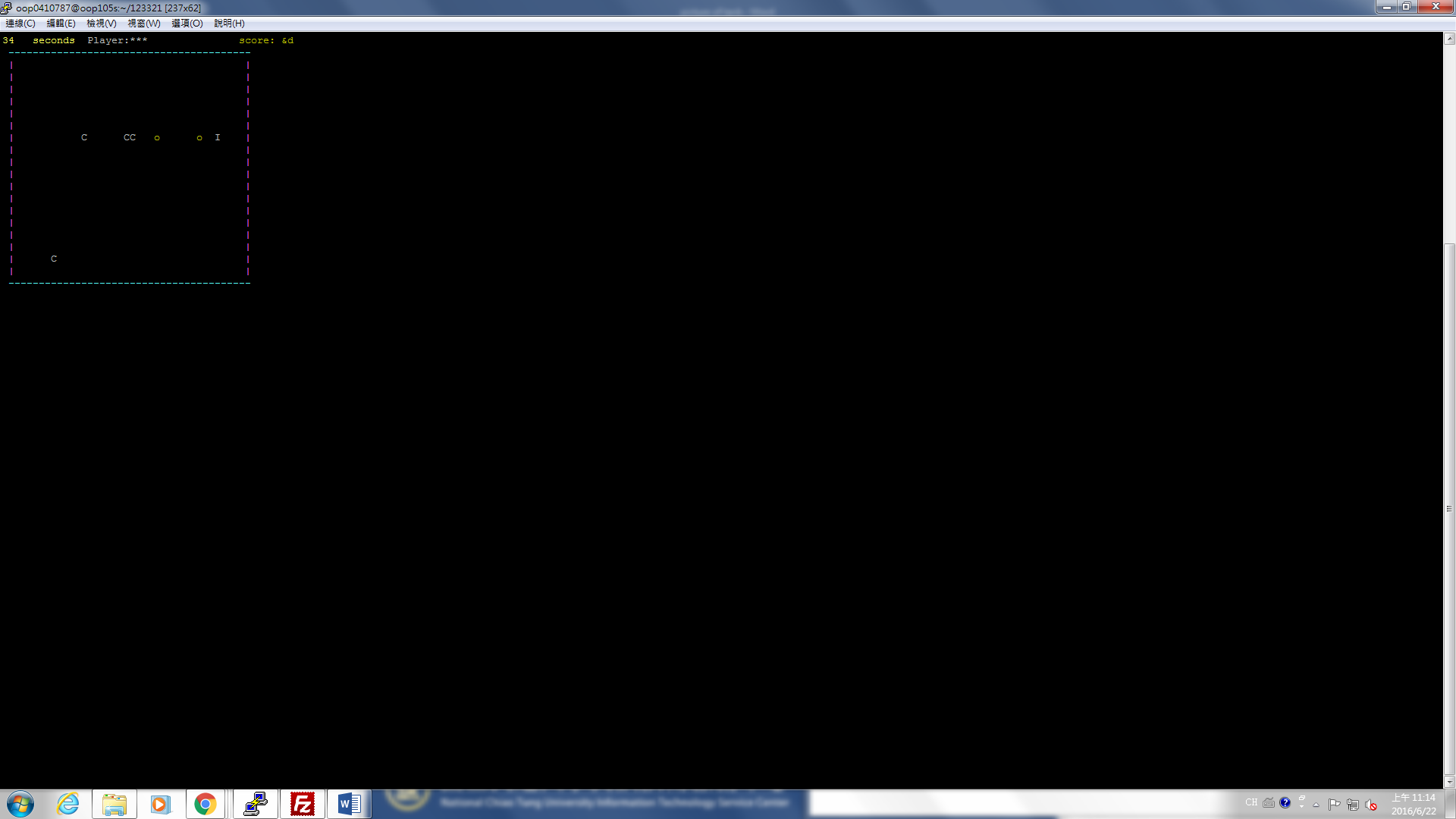
P.S.Class definition皆在原始程式碼中

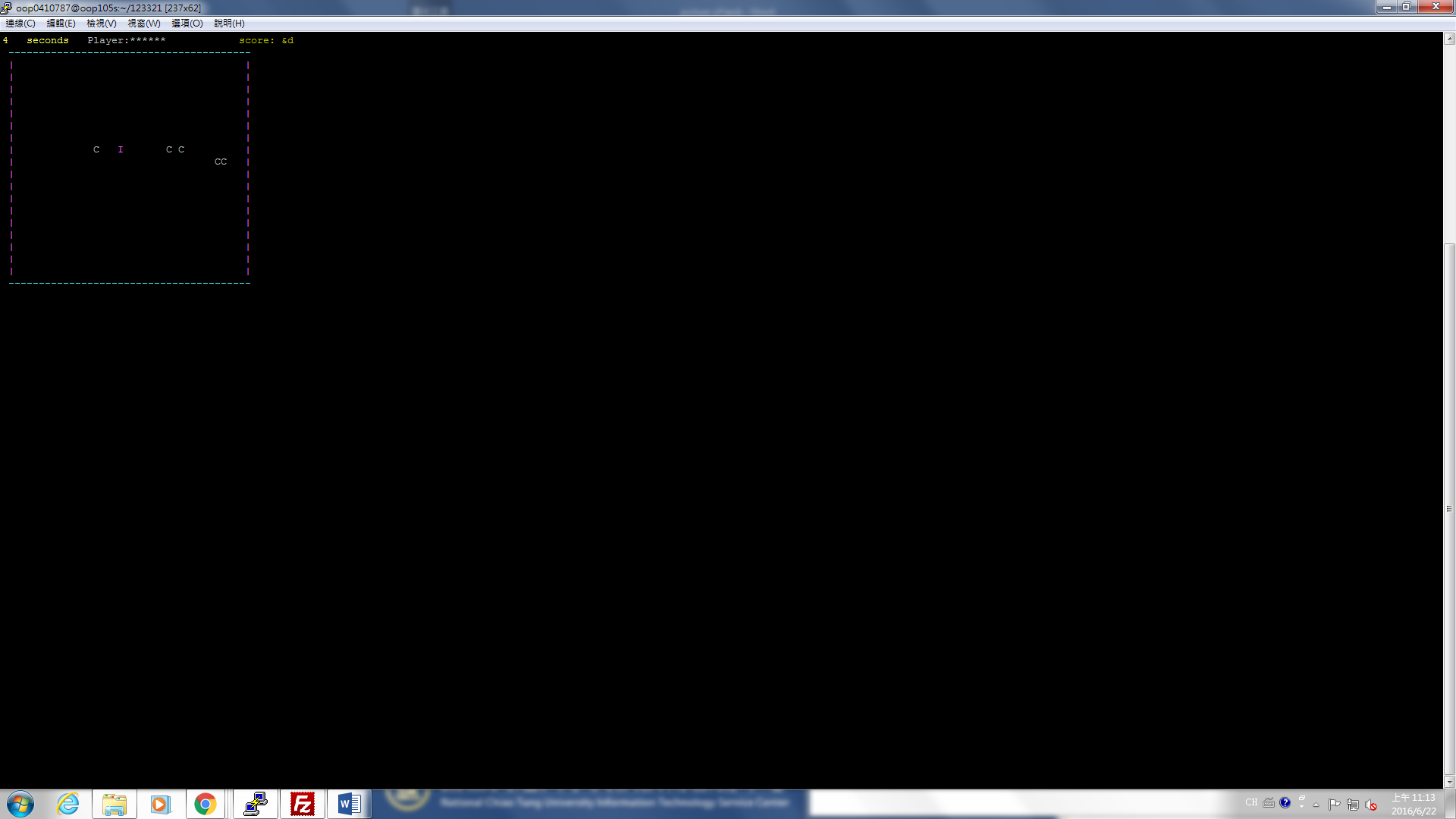
1. 部分畫面擷取圖

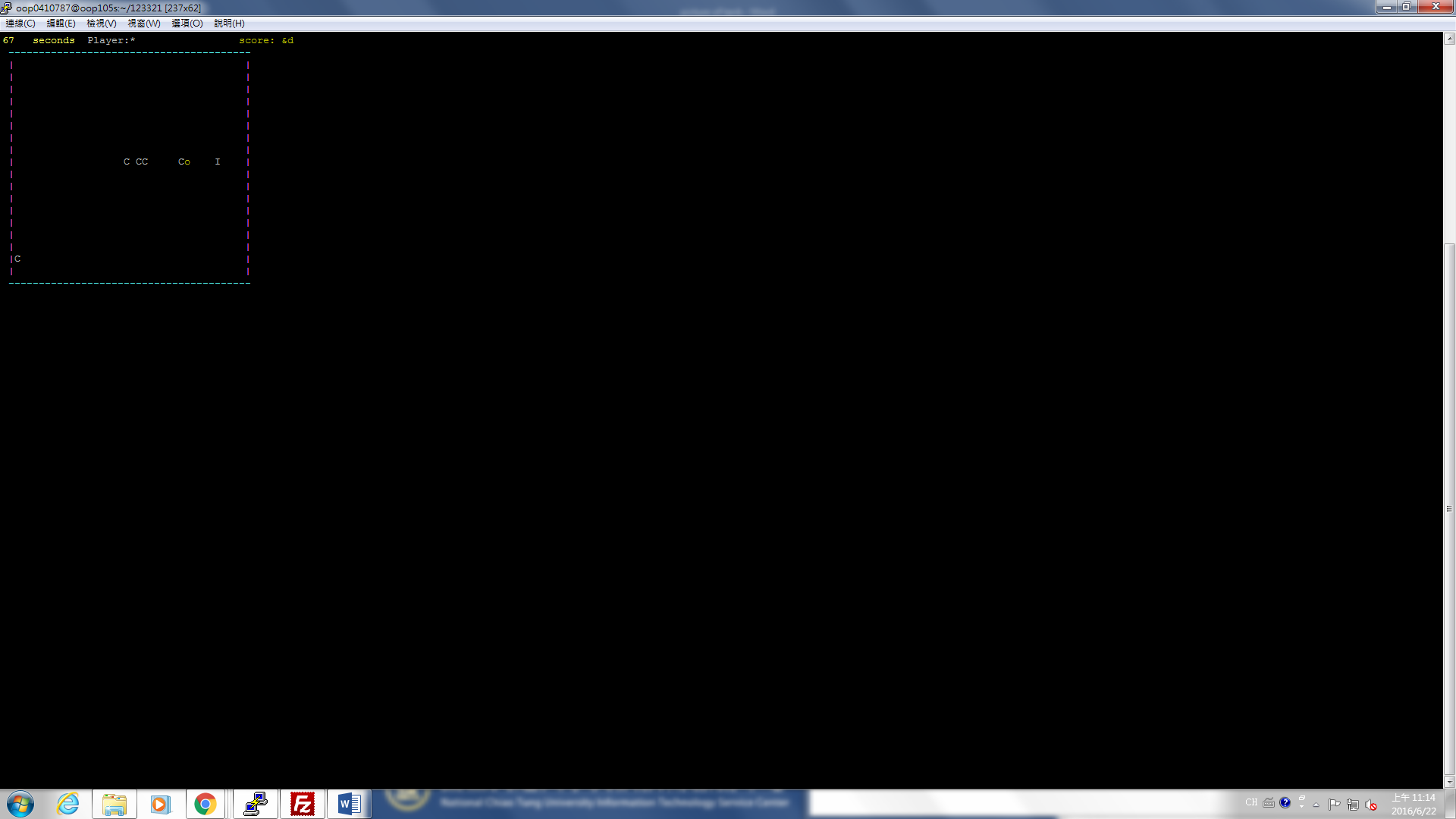
Game start:

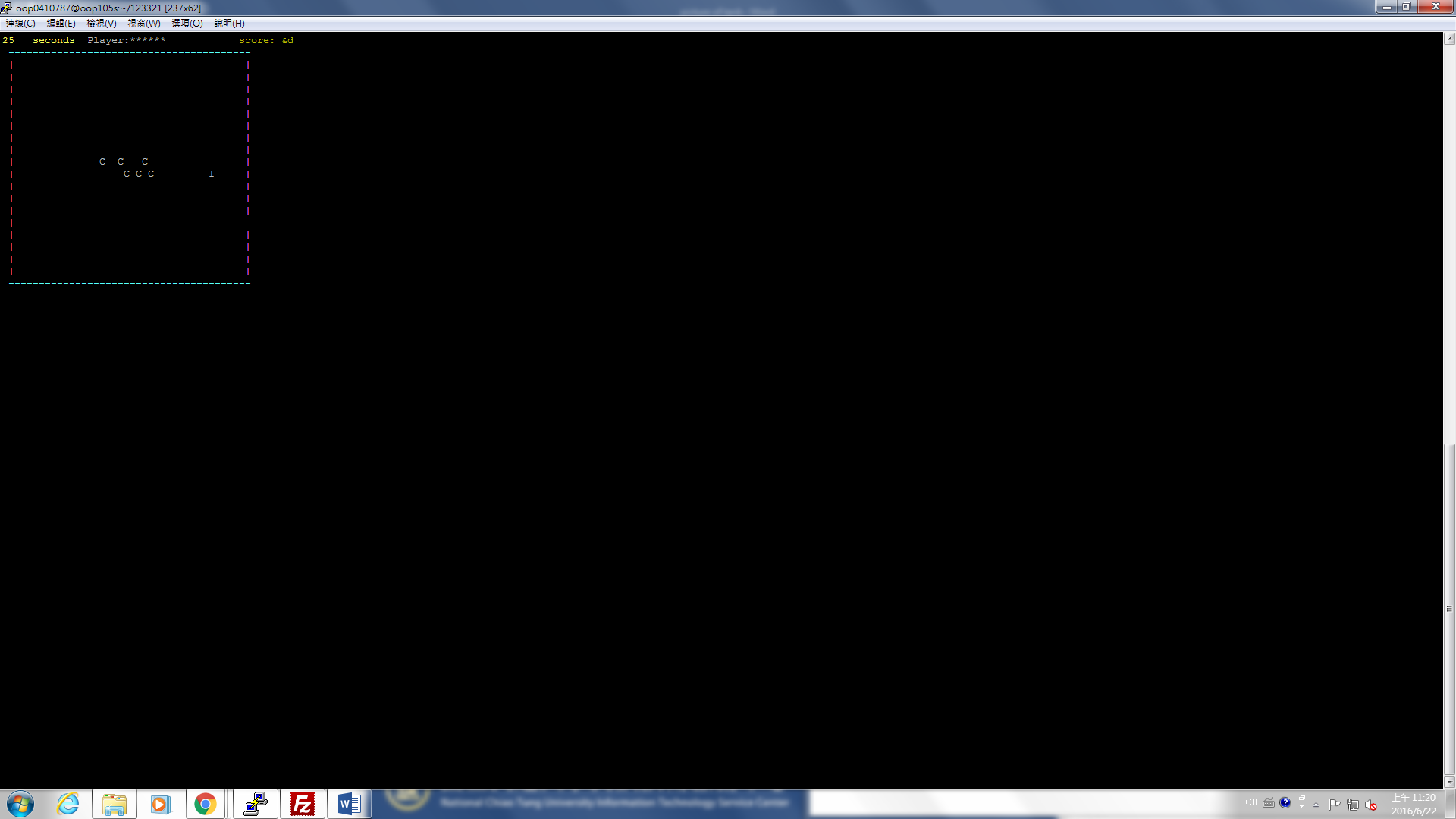


Bullet Shot:

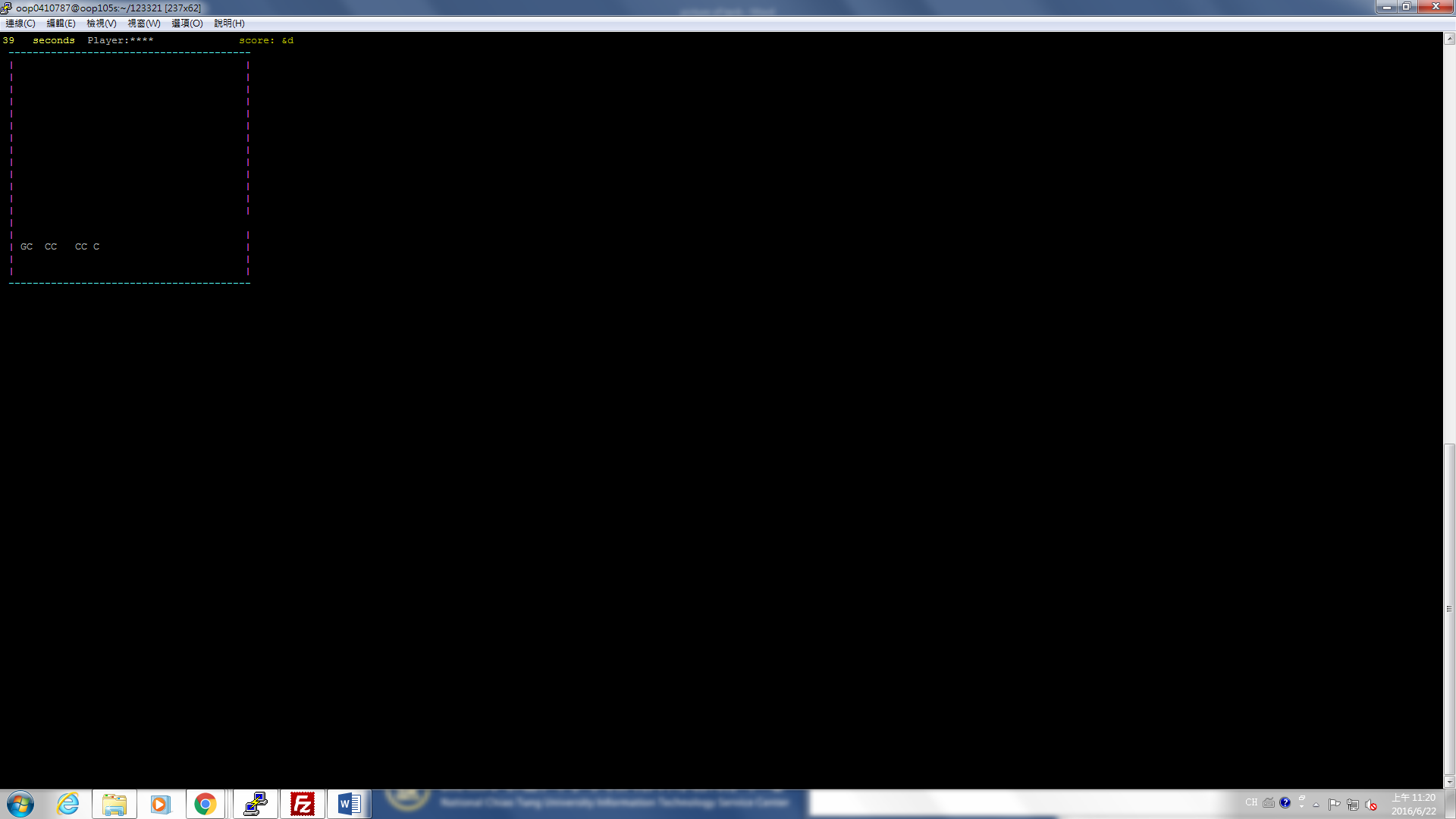




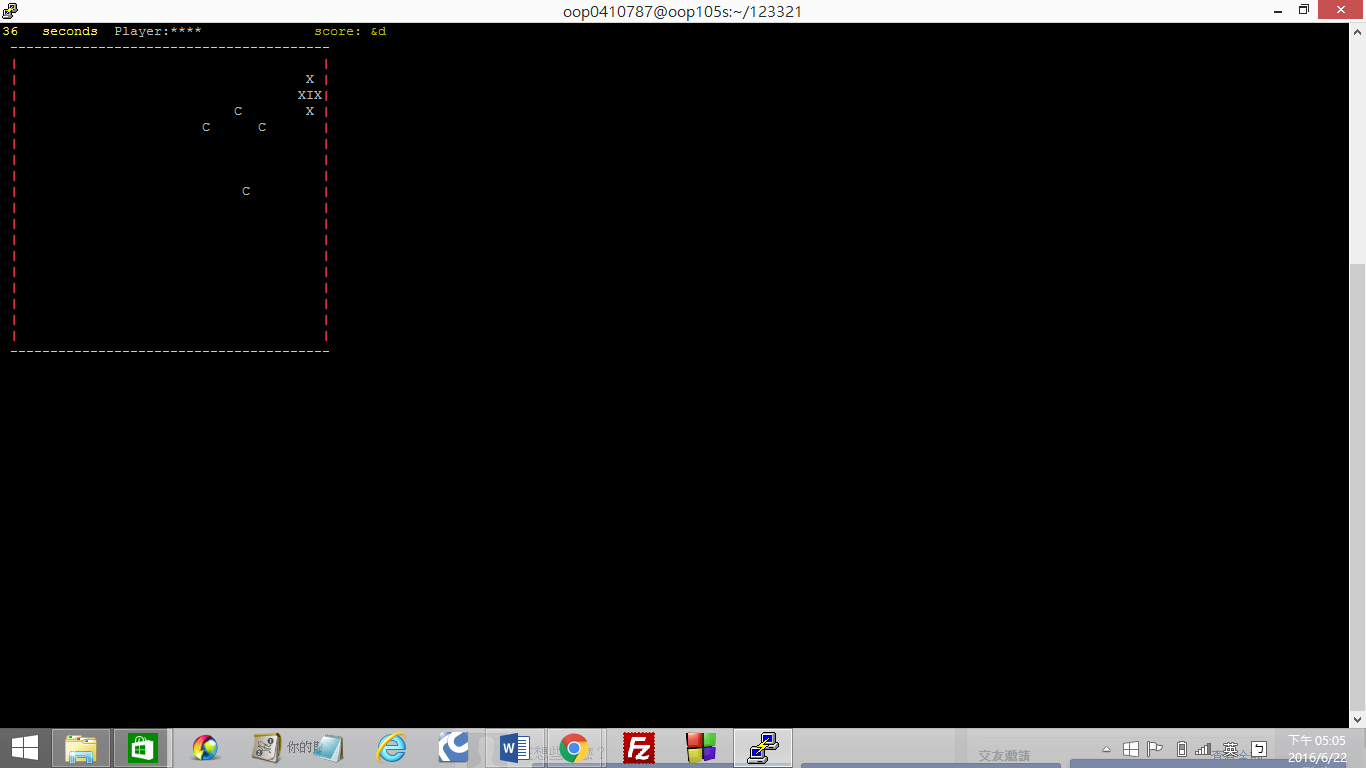




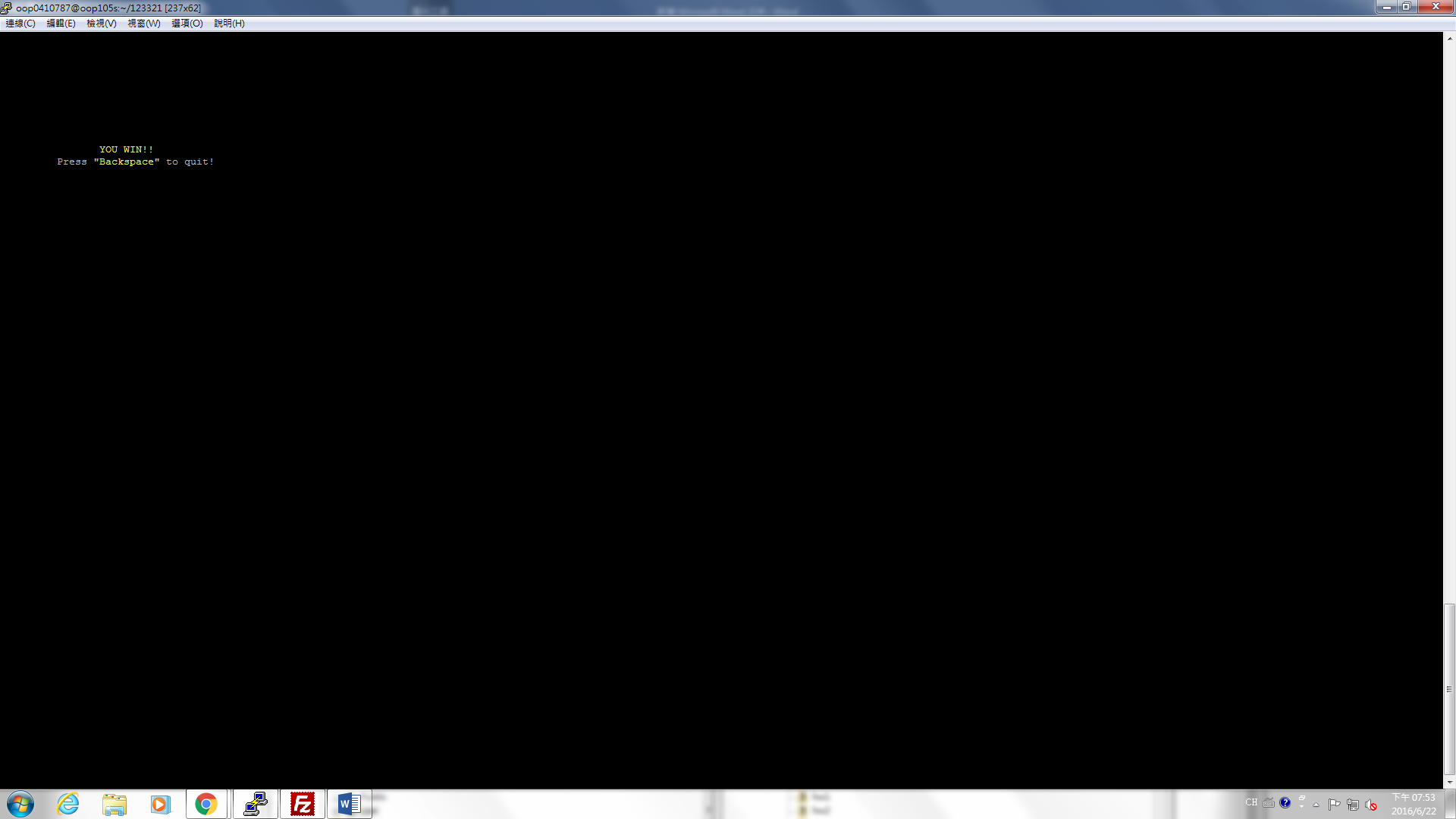
Monster attack player ( I => G)



Trap triggered by player (player halts for three seconds):



When you kill more than 15 monsters (Game End):



When you got hurt 6 times (Game End):



四、分工情形

林以鑫：原始程式結構(因原是在visual studio上編譯，故其結構不可在工

作站上編譯)

黃宇晨：將原始程式結構轉為可在工作站上運行的版本

高璿哲：原始版本與工作站版本的細部class內容的編寫(如 trap等)

五、心得感想

高璿哲：

這次的專題對我而言是很大的一項挑戰，我一向是對吸收新的觀念較為生疏，而這次我們選得主題，除了上課老師有講到的物件導向觀念以外，還需要用到其他的library來處理print的問題，因此在這上面花的時間實在可觀，遇到困難方面，有個蠻大的問題是，我們沒有去特別注意到coding style的問題，而大家安排class和member的方式又不盡相同，造成後期要將原本在visual studio上運行的版本轉成能在工作站上運行的版本時很不順，花了很多時間，還有一點很可惜的就是，我們組員林以鑫原本有設計一個BOSS關卡，另外用class snake設計一條蛇當作魔王，還有以前面的class作為基礎設計各項技能，且在visual studio能成功運行，只不過在要轉成工作站版本時遇到編譯器不支援的問題，卻又找不到有效的解決方案，只好放棄，真的很可惜也很難過，不過我在這次的專題中仍然獲益良多，畢竟我們嘗試了向坦克大戰這樣之前從來都沒見過的程式執行的模式，也成功在團隊合作下，以物件導向的方式，將這樣的概念作活用，相信這會對未來的，不論是與人合作或是獨自開發，都有相當程度的幫助。

林以鑫:

我終於做出拉奇烏拉了!!!(然而LINUX上的編譯器不支援)

這次的專題，我經過一翻考慮，決定拿自己最近在玩的遊戲來做成C++ 版本。於是我先設計了小怪及玩家。由於我是在visual studio做的，許多 函式庫LINUX並不支援，所以我交給其他組員找尋替代函式，最後成功的替 代掉<Windows.h>及<conio.h>中的某些函式。

接下來，我們新增了一些額外的物件，設計了血條、陷阱，還有BOSS。 BOSS主要由我負責，畢竟只有我在玩那款遊戲，於是我找了<thread>、 <mutex>來實現他。而這次我做比較久，快到期了才做完，於是替代函式並 沒有全部找到，沒辦法將<mutex>編譯，於是最後放棄了BOSS。(檔案中附 上編譯後的exe執行檔，有興趣可參考一下)

{ BOSS攻擊模式: 吃(E)、戳(S)、毒(G)、移動(O)

Player行動: 上(W)、下(S)、左(A)、右(D)、攻擊( )}

這次遇到的問題，除了編譯外，組員間不同的coding-style也浪 費了很多時間。由於真的難懂對方寫的東西，光是理解就花了一堆時間， 這讓我瞭解到縮排、命名、以及註解的重要性，畢竟以前都是打CODE給自 己看，完全沒這困擾。

黃宇晟:

這次的專題很感謝跟我同組的組員，從一剛開始的雛型，就用V

isual Studio 寫了出來。只限於當時不知可以實際在工作站上實作的方法。後來，當得知作法後，我負責移植結構到Linux平台之上。但當時我卻非單純的移植，而是選擇了，分檔並修改的步驟。改變了不同型別，物件間的觸發判定，而這又大多是將資料利用各自型別中宣告的function去做判斷，變得極為麻煩。最開始沒能想到這樣的結果，以至於拖累了整組進度。最後熬夜趕工，總算完成了基本的部分。我覺得多虧大家的包容我才能與大家共同完成專題，所以呢非常感謝以鑫，他花了許多時間研究實作的方式，並實際構思出整個程式的主架構。此外覺得璿哲加入的陷阱非常有意思。謝謝大家和我一起徹夜奮鬥。

在寫程式時，當我只是一個人時，就想得不太清楚。幸好這次能有一起工作的夥伴，讓我看到大家其實都很強。這次的經歷，告訴我我不太適合構思太過複雜的事物。此外，若論及工作站外的進度，其實其餘兩人寫了得多了更多。而我只將這些整合起來，但我卻在其中犯了致命的錯誤。

其實我應該維持原有的架構，如此不但能確保程式的正確性，也不用耗費時間debug。而debug 實屬花費了我最多時間的原因，也讓我們這組的進度相應落後。所以說我應該更相信他人的並尊重他們的code，應該先坐進一步地確認，再開始做事。另外，原本易懂的程式碼，經我手後，卻變得不易讀懂，此外又因我未適當地加入註解，造成大家不易讀code的困擾。這總總miss 造成了的失誤，卻是由組員熬夜彌補，這是不應該的，由此我也很感謝這麼棒的隊友。最後實在是想為以鑫報怨，他花了數天完成的boss，卻因為g++compiler 的版本限制，而無法添入，為此而感到憾然。(g++-11 在 邊<multex> 宣告的物件時，無法找到適當的某個東西)

那麼經過這次的事情，我對demo的成品雖然感到有點不安，但這遊戲確實吸引著我。整體上，雖然覺得作品已具備基本的功能，若是時間還有所剩餘的話，應該就會進入增添細節的環節。另外設定檔案的部分，還需要使的作品更加的處理細節，在這之外若還有時間，希望能把所有事件判定的部分改用呼叫global function的方式， 以簡化程式的複雜度。最後想由衷表示沒能引入Boss 到這遊戲中的戰場，實在是不服氣啊。嗯，這次的專題依舊讓我學到不少。

。