2016 Web Programming Final Project

**Group ID:** 8

**Project Title:** IO game (like AoE)

**Team Members:**

電機四 陳俞兆

電機三 黃于瑄

**Contact Information :**

**Mail:** [b03901156@ntu.edu.tw](mailto:b03901156@ntu.edu.tw)

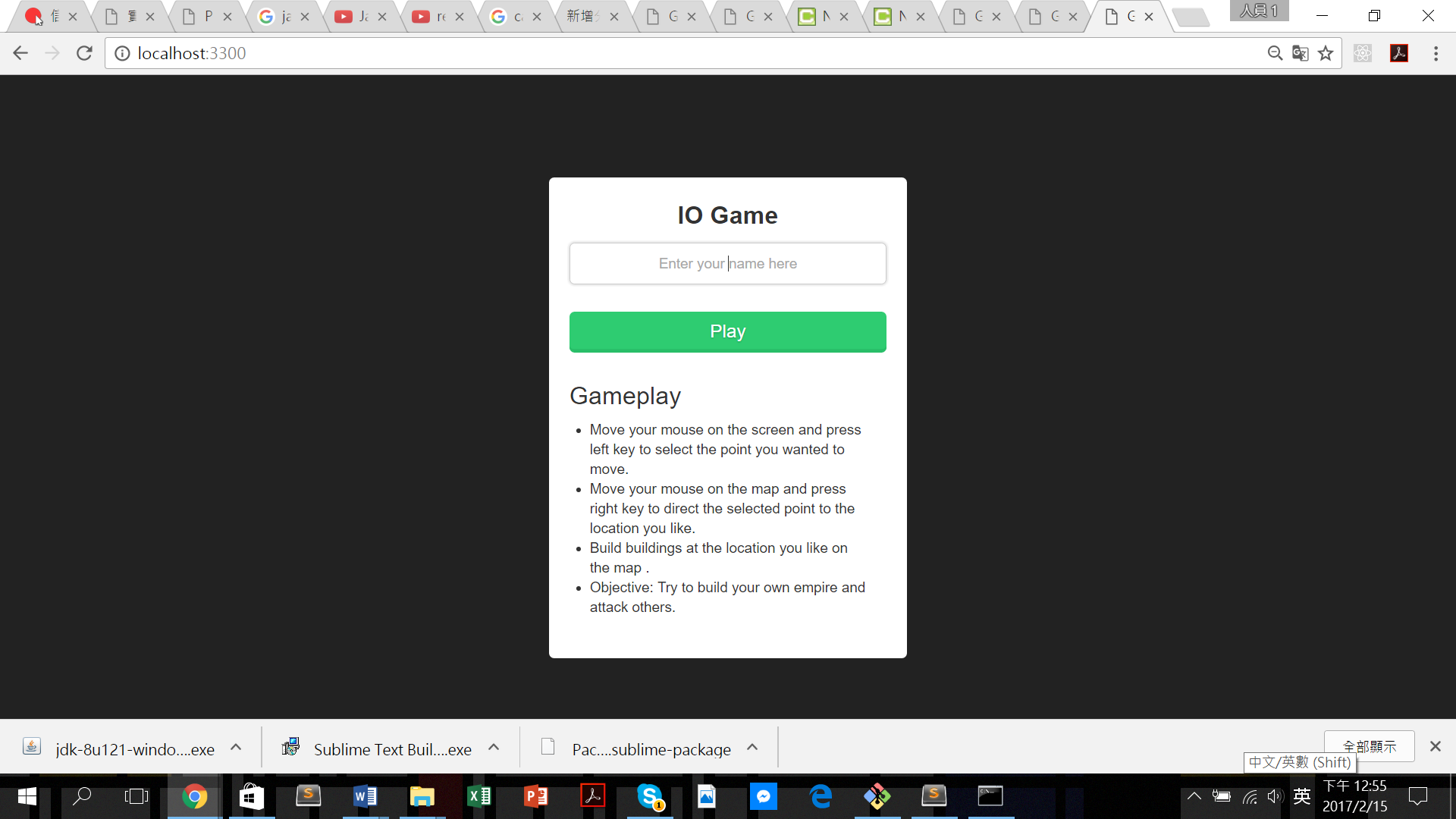
**Mobile:** 0979740757

Content：

1. Function Description
2. IO game 版本的策略遊戲(AoE)
3. 由server (based on Node.js)統一管理所有遊戲的connection information
4. Server將每8個connections (players)編入同一遊戲中隨機分成A、B兩隊，並記錄各隊資訊
5. 遊戲主介面由canvas、pixi、p2、phaser所寫成
6. 遊戲介面提供地圖探索功能 (移動視窗範圍進行探索)、移動功能
7. 提供基本的蓋建築、攻擊功能
8. How to Use

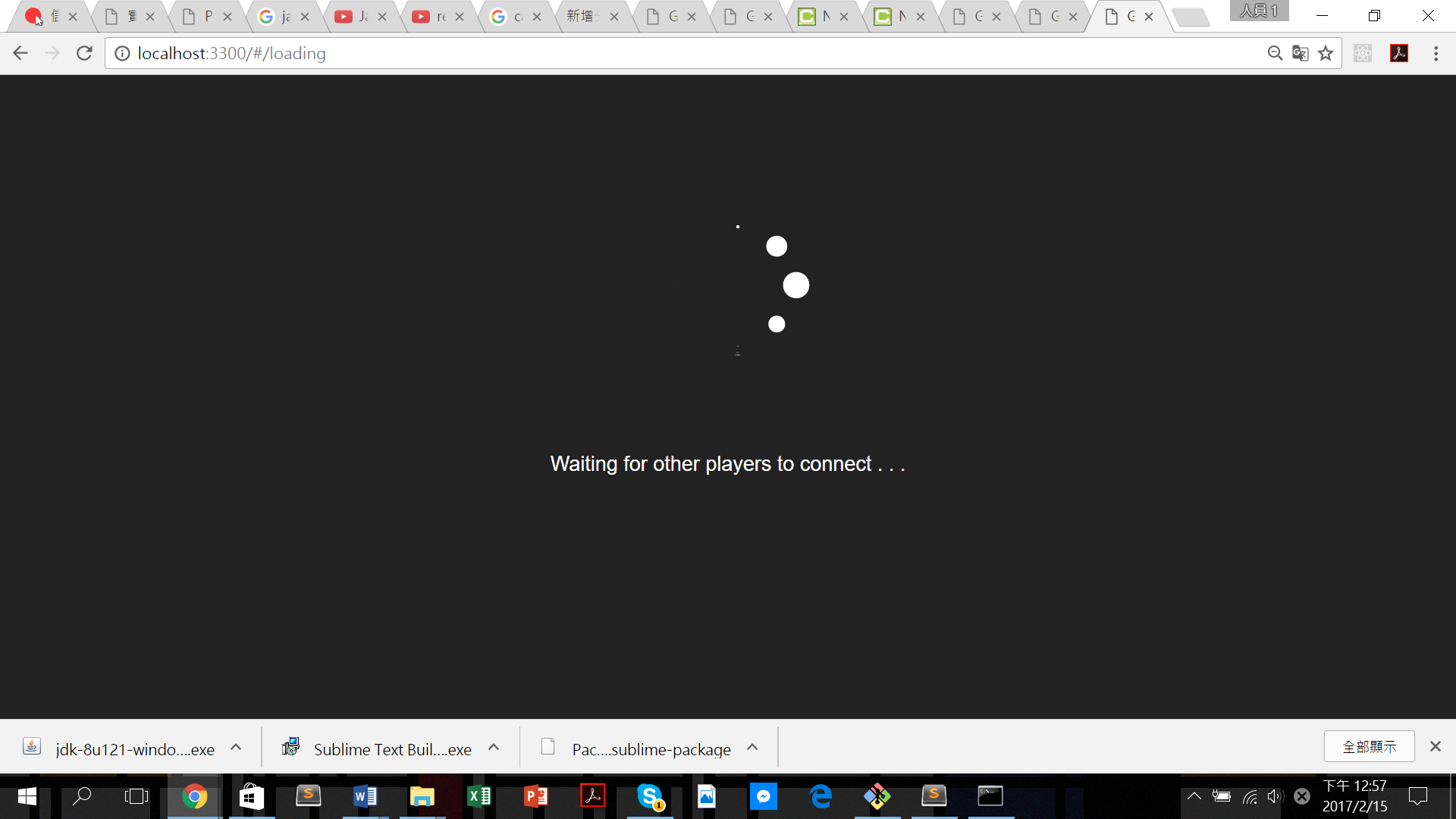
Enter the website

Input user name



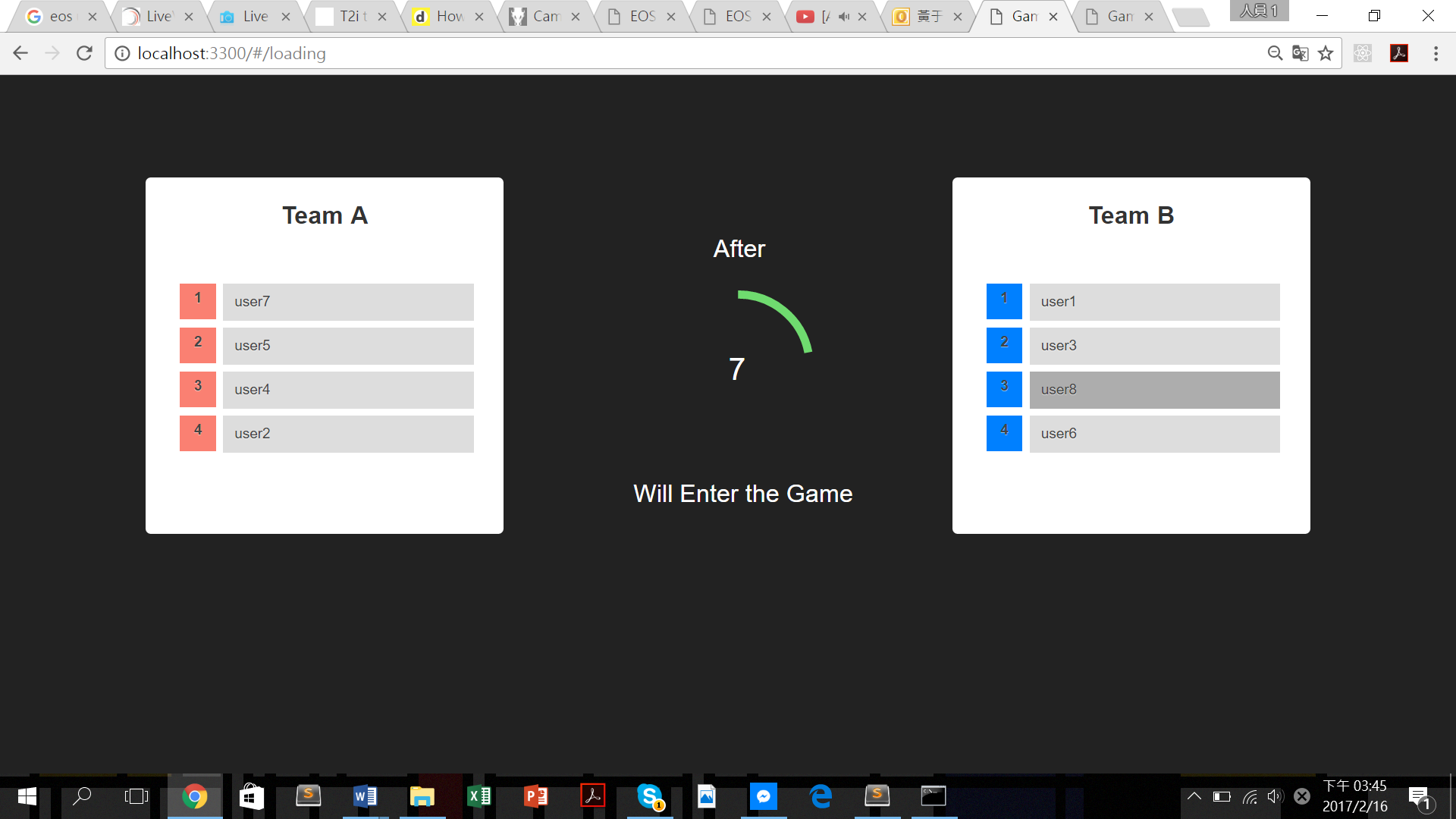
user等待分隊(A/B team, 4 members/team)，server記錄各玩家information

(等待畫面)

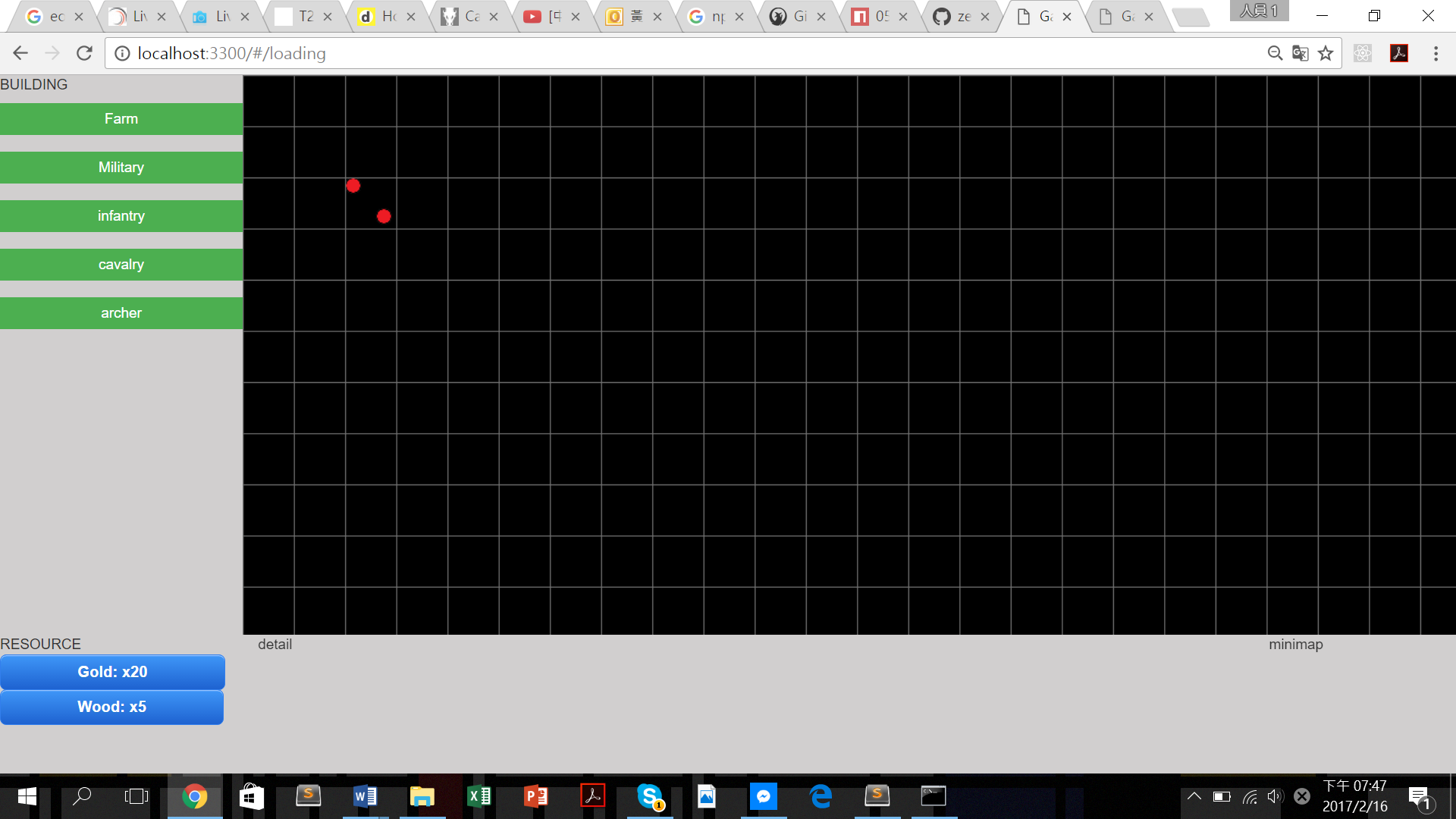


User 將被隨機編入Team A or Team B, 並於頁面中告知該user的所屬隊伍；同時倒數10秒以進入主遊戲畫面

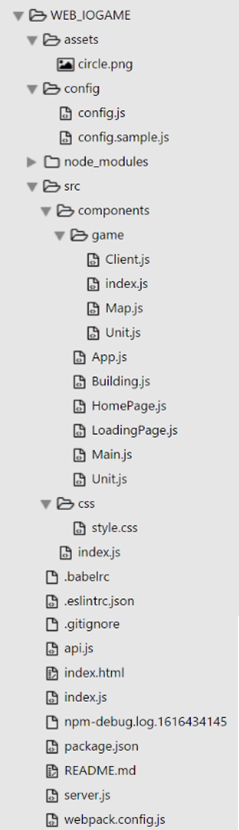
(以下以user8為例)



1. 在地圖上以滑鼠左鍵拖曳出一個涵蓋該兩點的方框(即select)，被選取之點即會以一定速率朝向滑鼠右鍵所觸及之location移動
2. 利用WASD鍵探索周遭地圖



1. Code Design and Implementation



// 資料夾game定義遊戲的主軸世界， 並藉由

render function 在遊戲主畫面中呈現

// 此處的index.js 實現探索地圖、移動至特定地點、蓋建築等基本功能(based on Canvas, phaser, pixi, p2)

// HomePage.js管理遊戲進入畫面並協助user輸入正確的玩家名稱

// loadingPage.js 管理遊戲進入等待分組介面後的頁面切換、A/B 兩隊的資訊傳輸(based on socket.io)及各Page的目標

// index.html管理整個遊戲的render輸出

// server (based on Node.js)利用socket.io管理所有遊戲的connection、information

1. Source Code Link

Link: <https://github.com/zeroHuang0516/WEB_IOGAME>

1. Used techniques
2. html, css and javascriopt
3. Node.js, React, npm, express, wedpack and mongoDB
4. konva
5. p2, phaser, pixi
6. node-uuid
7. socket.io
8. Division of work

電機四 陳俞兆:

遊戲主頁面的設計(前端)，包含地圖探索，移動等功能，能讀取來自server的訊息；並提供遊戲的設計概念

電機三 黃于瑄

(後端)Index.html、Sever、Homepage及LoadingPage的撰寫；負責socket.io的相關功能

1. Reviews

在這一次的期末專題中，我們希望能撰寫類似AoE(世紀帝國)的IO game, 然而在實作過程中卻發現AoE 和 IO game 的遊戲性質差異相當大: AoE 是策略思考遊戲，因為帝國設定的關係所涵蓋層面較廣，因此遊戲較大多的IO game來的複雜；儘管我們已大大簡化遊戲內容的設計，仍因connection的管理，server與game server的資訊傳遞而傷透腦筋。

這一次的實作中，我們也使用了之前所未曾碰過的工具；一邊翻著相關文件一邊寫code雖然不如已學習的知識使用來的得心應手，但在無形中的確學到了不少新知(~~學習的速度也因為即學即用的迫切感而增快不少~~)。

經由這一次的專題實作，最大的收穫在於了解分工合作以及code的架構的重要性: 在分工合作方面，必須適當地將先前僅由一人寫code的思維、習慣捨去，改以團隊合作的態度投入工作中；而在code的架構方面，隨著專題的壯大就越能體會code架構的重要性，每每寫到一半才在開始後悔當初訂架構的不明智，我們想以後應可透過多看看open source 的專案以學習如何寫出架構良好的code.