

國立高雄科技大學
智慧商務系

資料結構期末專題

將 Minecraft Bot 使用類別的方式重構

C109156107 王昱翔

指導教授：謝文川

中華民國 112 年 1 月

第一章、緒論

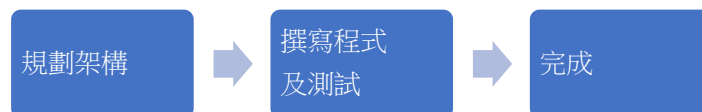
第一節 研究動機與目的

前一陣子我在遊玩 **Minecraft** 這款遊戲時，在多人伺服器因為要想辦法轉遊戲幣，又無法 24 小時在線遊玩，於是我找到了 **Mineflayer** 這個 npm 開源套件，該套件可以撰寫屬於自己的 **Minecraft Bot**，於是我使用了這個套件製作了一個可以 24 小時幫我補捉生物的 **Bot**，在遊戲裡面有一個叫做掠奪者的生物，生成時有低機率會生成掠奪者隊長，差別為背上會有一個白色旗幟，該 **Bot** 會針對隊長進行捕捉。

不過寫好的程式在可讀性上有較大的問題，因此這次我嘗試使用類別的方式重構程式碼，作為本次報告主題。

第二節 研究流程

本次重構重點在於建立 **Mineflayer Bot** 的部分，首先規劃架構再來撰寫程式，並且撰寫的過程會確保執行結果與之前的一樣。



第三節 計畫時程

本次報告於 112 年 1 月 21 日開始規劃架構，1 月 22 至 27 日重構程式碼，最終在 27 日完成。

第二章、實作與展示

第一節 舊程式碼

原先的程式碼是將 create mineflayer bot 與後續動作寫在同一個 function，這邊我預計將他重構成一個 class，包含 create bot 與抓取的功能。

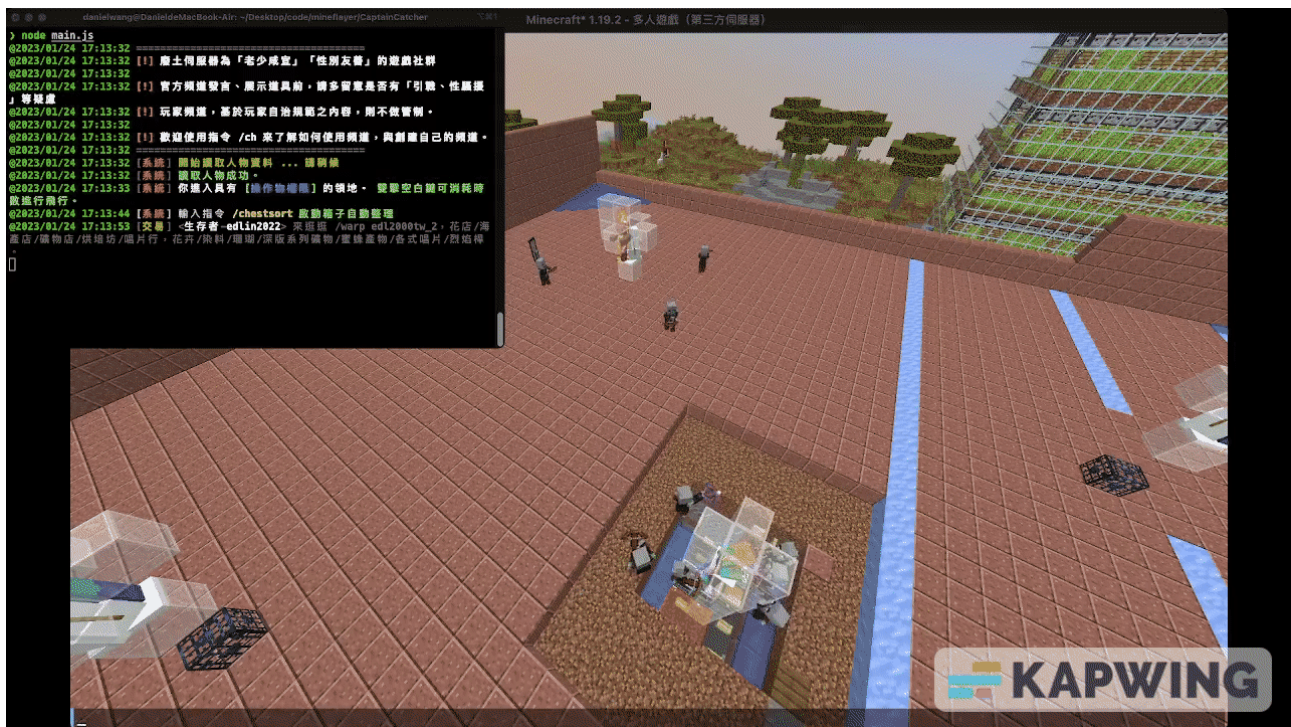
```
21 async function connects() {
22   tokens.use(opt, function (_err, _opts) {
23     if (_err) throw _err
24
25     const bot = mineflayer.createBot(_opts)
26
27     bot.once('spawn', async () => {
28
29       mcData = require('minecraft-data')(bot.version)
30
31       await sleep(5000)
32
33       caught_egg_total = 0
34       start_time = performance.now()
35
36       while (!STOP) {
37
38         // 在bot身上搜尋是否有空蛋
39         let emptyEgg = await bot.inventory.findInventoryItem(891)
40
41         if (emptyEgg != null) {
42
43           // 將空蛋拿在手上
44           await bot.equip(emptyEgg, 'hand')
45
46           for (const mob of Object.values(bot.entities).filter(e => e.type === 'mob' && (e.mobType === 'Pillager')) {
47             for (const element of mob.equipment) {
48               if (element && element.displayName === 'White Banner') {
49                 now_time = performance.now()
50                 await bot.useOn(mob)
51                 await depositCapEgg(bot)
52               }
53             }
54           }
55
56         } else {
57           await takeEmptyEgg(bot)
58         }
59
60         await sleep(500)
61       }
62     })
63   })
64 }
```

修改前程式碼

第二節 重構後的程式碼

```
101 class mineflayerBot {
102   constructor() {
103     this.promise = new Promise((resolve, reject) => {
104       tokens.use(opt, (_err, _opts) => {
105         if (_err) reject(_err);
106         else {
107           this.bot = mineflayer.createBot(_opts);
108           resolve();
109         }
110       });
111     });
112     this.caughtTotal = 0;
113     this.startTime = 0;
114   }
115
116   async catcher() {
117     while(true) {
118       // 搜尋身上是否有空蛋
119       let emptyEgg = await this.bot.inventory.findInventoryItem(891)
120       if (emptyEgg != null) {
121         // console.log('has emptyEgg');
122         await this.bot.equip(emptyEgg, 'hand');
123
124         for (const mob of Object.values(this.bot.entities).filter(e => e.type === 'mob' && (e.mobType === 'Pillager'))) {
125           for (const element of mob.equipment) {
126             if (element && element.displayName === 'White Banner') {
127               await this.bot.useOn(mob);
128               await depositCapEgg(this, this.bot);
129             }
130           }
131         }
132
133       } else {
134         await takeEmptyEgg(this.bot);
135       }
136
137       await sleep(600);
138     }
139   }
140 }
141
142 module.exports = mineflayerBot;
```

修改後程式碼



程式演示畫面

第三章 結論

此次重構利用了類別來實作，我把 Mineflayer Bot 物件寫在 mineflayerBot 這個 class 底下，連同抓取數量、程式啟動時間、抓取的函式...等等一同放在該類別底下，使得程式的可讀性提高了不少，維護起來也更方便。