第十五組
人工智慧-期末報告

鄭璟翰 B093040003

郭晏涵 B093040024

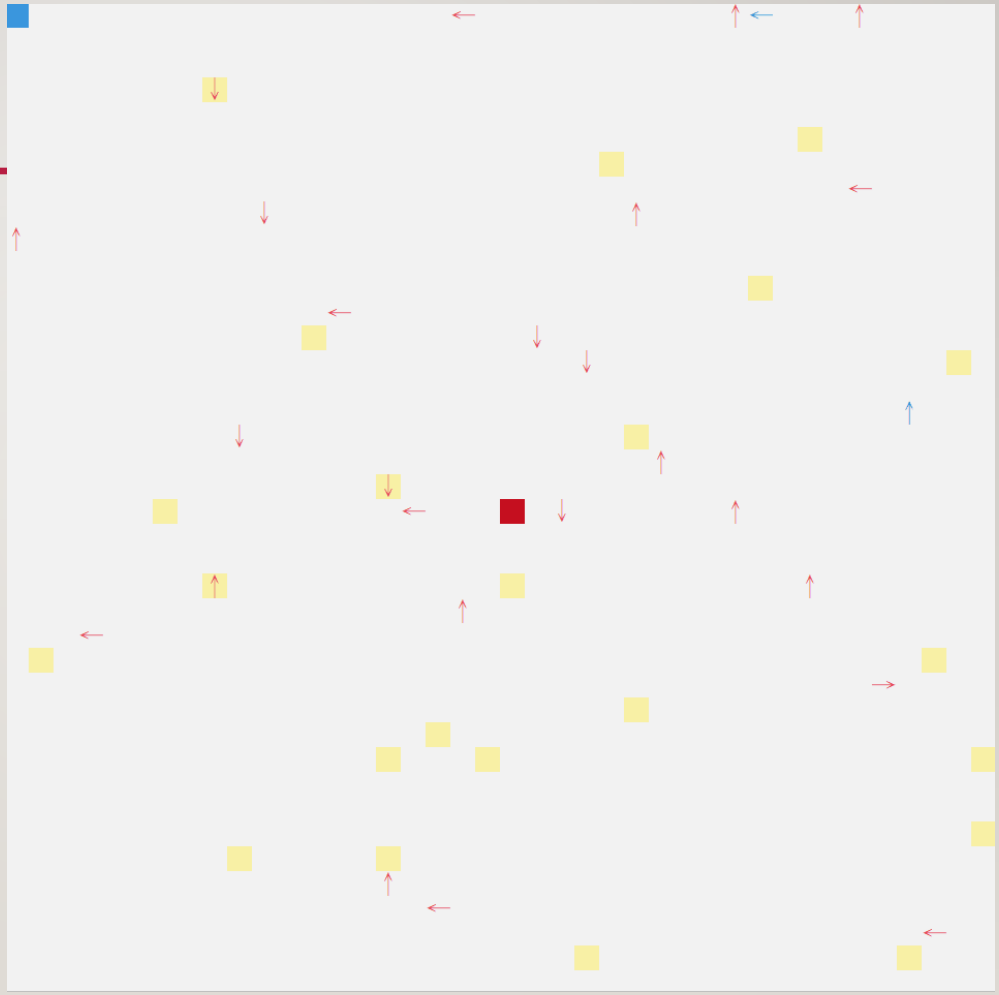
授課教師：蔡崇煒

大綱

- 前言簡介
- 程式設計方式與過程
- 移動策略
- 執行結果
- 結論

前言簡介

- 期中報告
- 用C++實現
- 掉落食物
- 食物搜尋功能



前言簡介-目標

加入費洛蒙

環境食物上限

限制螞蟻壽命

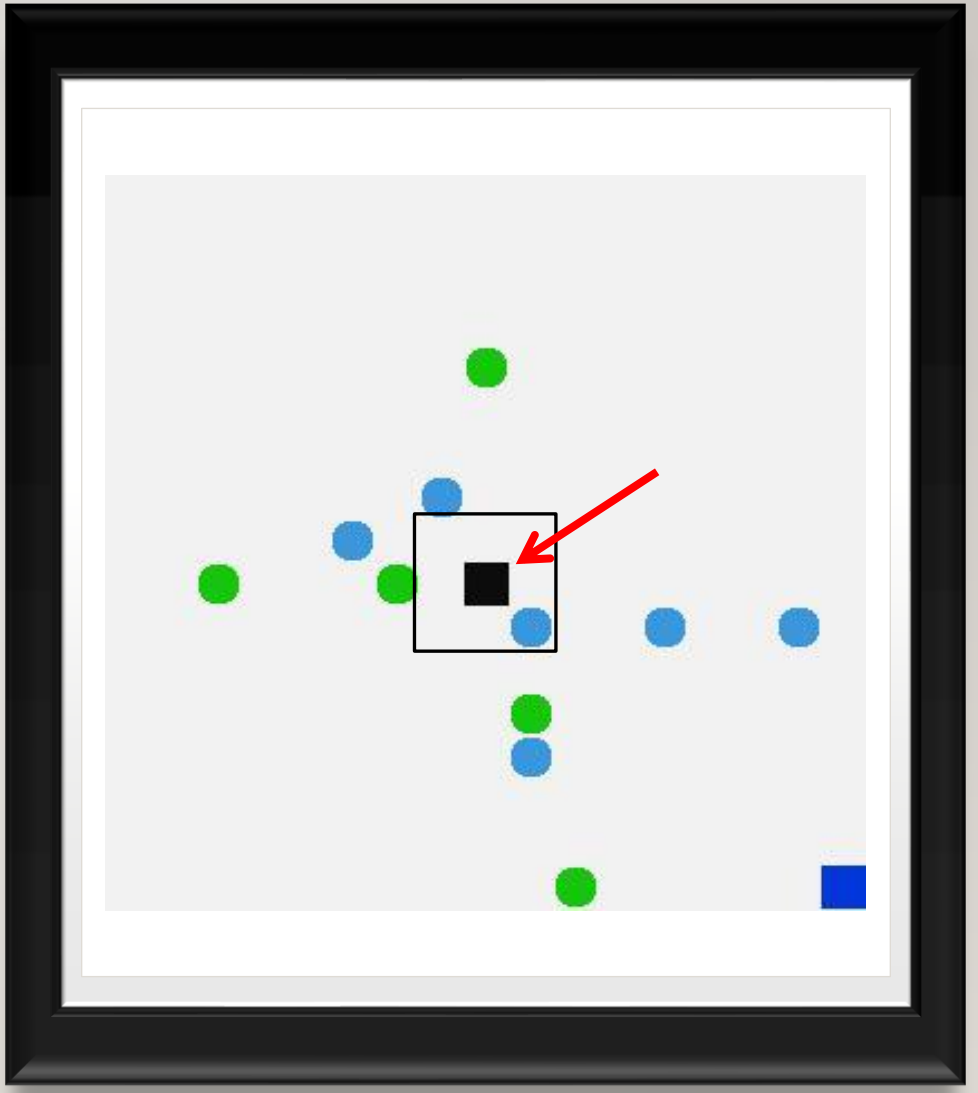
不同移動策略

獎勵機制

程式設計方法與過程

巢穴

- 黑色(表示洞口)
- 相鄰四點為出生點
- 對角四格為歸巢點



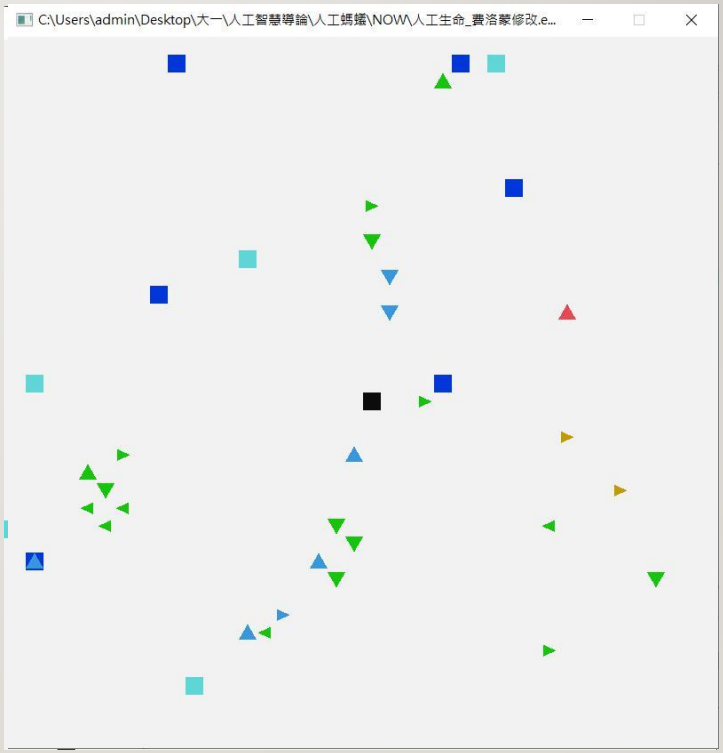
螞蟻(樣式)

- 設計三種型式：

  箭頭、圓形、三角形

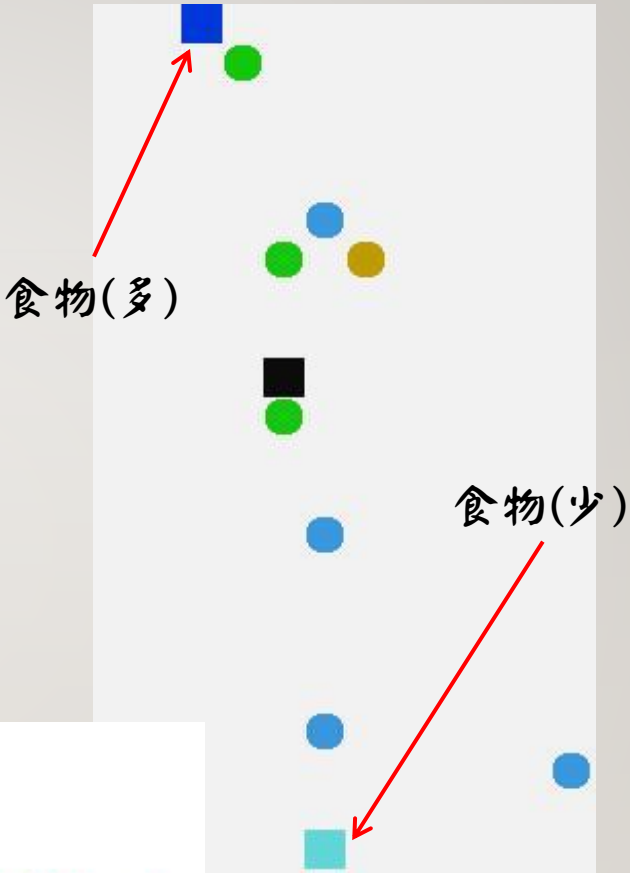
```
#include <io.h>
#define SIZE 40
#define ANT 0                   //2是三角形 1是箭頭 0是圓形
#define YOUNG_COLOR 10
#define MID_COLOR 6
#define OLD_COLOR 12
#define FIND_COLOR 3
#define TEN_FOOD_COLOR 16
#define FIV_FOOD_COLOR 176
#define DEAD_COLOR 128
#define NONE_COLOR 240
#define BORN_COROR 51
#define HOME_COLOR 0
using namespace std;
```

```
715     switch (ANT){
716         case 0:
717             switch (ant){
718             case ' ':
719                 cout << setw(2) << " ";
720                 break;
721             default:
722                 cout << setw(2) << "• ";
723                 break;
724             }
725             break;
726         case 1:
727             switch (ant){
728                 case 'u':
729                 case 'U':
730                     cout << setw(2) << "↑ ";
731                     break;
732                 case 'D':
733                 case 'd':
734                     cout << setw(2) << "↓ ";
735                     break;
```



食物

- 每個點可領取十次
- 環境設有上限(十個)
- 剩餘食物五個以上，深藍
- 剩餘食物五個以下，淺藍




```
151     if(foodnum<=10){    //產生新食物
152         if(food_cycle==9){
153             food_cycle=0;
154             do{
155                 x=rand()%SIZE;
156                 y=rand()%SIZE;
157             }while(color2[x][y]!=0 || produce[x][y]=='x');
158             color2[x][y]=10;
159         }else{
160             food_cycle++;
161         }
162     }
```

螞蟻(壽命)

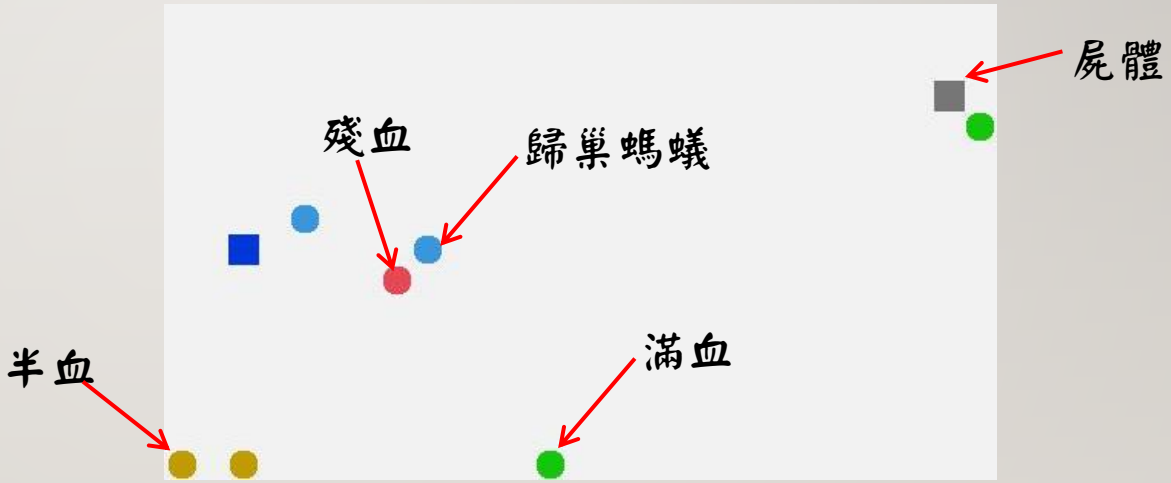
- 每隻螞蟻血量250滴，

  走1步扣1滴
- 滿血(血量250~171)、

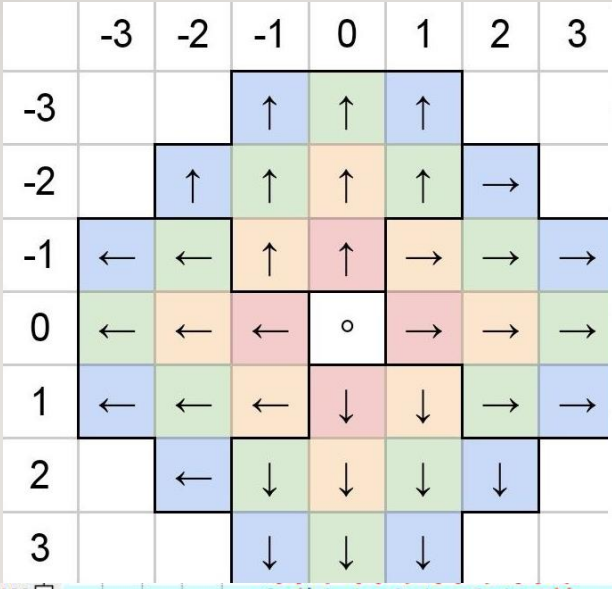
  半血(血量170~91)、

  殘血(血量90~0)
- 死亡：

  血量歸零、留下屍體



食物搜尋範圍及方法



紅色走一步就能到達
橘色兩步
綠色三步
藍色四步

```
if(紅格有食物){
   計算紅格哪裡食物多就往哪裡走
}else if(橘格有食物){
   計算橘格哪裡食物多就往哪裡走
}else if(綠格有食物){
   計算綠格哪裡食物多就往哪裡走
}else if(藍格有食物){
   計算藍格哪裡食物多就往哪裡走
}
```

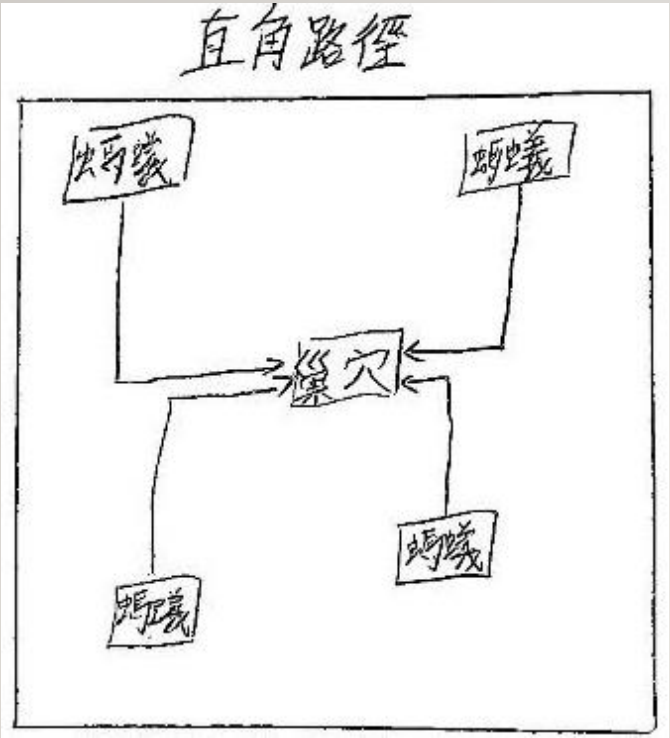
```
196                 for(int step=1;step<=4;step++){          //以周圍食物作為判斷依據
197                     for(int k=-step; k<=step; k++){
198                         if(k==4||k==-4){
199                             continue;
200                         }
201                         for(int l=-step; l<=step; l++){
202                             if(l==4||l==-4){
203                                 continue;
204                             }
205                             if(k+l==step || k+l==-step || k-l==step || k-l==-step){
206                                 if(color1[i+k][j+l]!=0 && i+k>=0 && j+l>=0 && i+k<=SIZE-1 && j+l<=SIZE-1){
207                                     if(k<0&&k+l<0&&k-l<=0){
208                                         d[0]++;
209                                     }else if(k>0&&k+l>0&&k-l>=0){
210                                         d[1]++;
211                                     }else if(l<0&&k+l<=0&&k-l>0){
212                                         d[2]++;
213                                     }else if(l>0&&k+l>=0&&k-l<0){
214                                         d[3]++;
215                                     }
216                                 }
217                             }
218                         }
219                     }
220                     if(d[0]!=0 || d[1]!=0 || d[2]!=0 || d[3]!=0){
276                 }
```

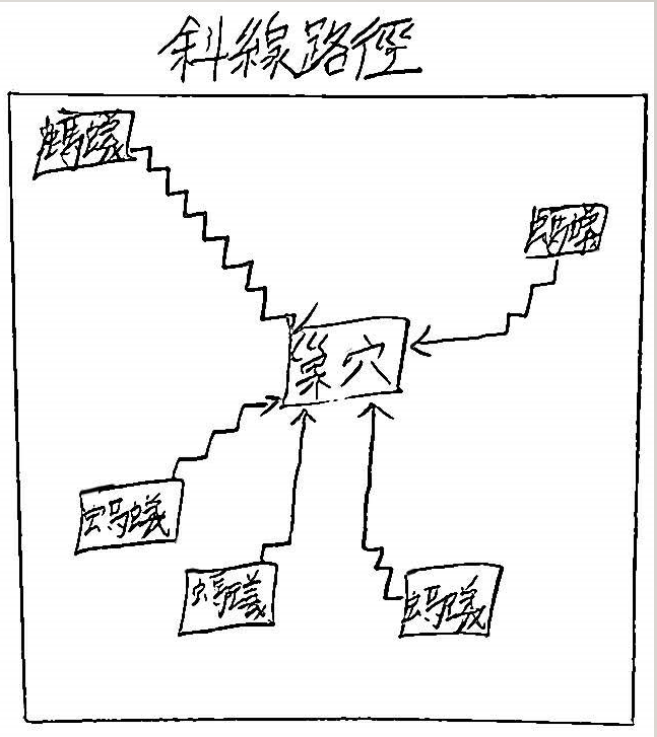
獎勵機制

- 找到食物的螞蟻會獲得能量，獲得另外五十滴血

```
163     }else if((ant1[i][j]=='u' || ant1[i][j]=='d' || ant1[i][j]=='l' || ant1[i][j]=='r') && color1[i][j]!=0){        //螞蟻找到食物
164         if(i>SIZE/2){
173         if(color1[i][j]>1){
176         if(color1[i][j]>0){
183         timeinterval2[i][j]=timeinterval1[i][j]-50;
184         if(timeinterval2[i][j]<0){
187         search2[i][j]=search1[i][j];
188     }else if((ant1[i][j]=='u' || ant1[i][j]=='d' || ant1[i][j]=='l' || ant1[i][j]=='r')){        //螞蟻沒有找到食物
```

```
183         timeinterval2[i][j]=timeinterval1[i][j]-50;
```





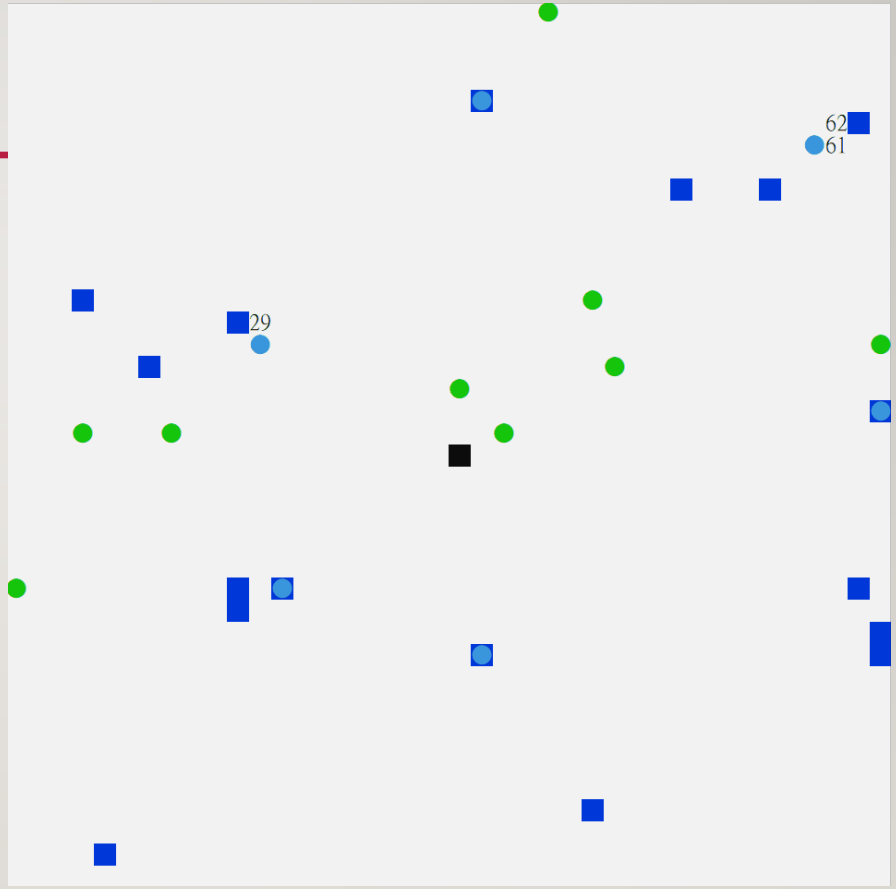
回巢規則

設計兩種：直角路徑、斜線路徑

費洛蒙

- 費洛蒙的初值為

  到巢穴步數的兩倍
- 每走一步費洛蒙釋放的濃度降低一單位
- 每一個時間單位費洛蒙濃度降低一單位
- 搬走最後一個食物或屍體則不會釋放費洛蒙



費洛蒙搜尋範圍與方法

- 和食物搜尋的範圍相同
- 將該方向的濃度加總，再決定前進方向

```
278                 //以費洛蒙作為判斷依據
279                 for(int k=-3; k<=3; k++){       //判斷周圍
280                     for(int l=-3; l<=3; l++){
281                         if((k+l<-4)||(k+l>4)||(k-l<-4)||(k-l>4)||(k==0&&l==0)){
282                             continue;       //中心和菜四角落跳過
283                         }else if(pheromone[i+k][j+l]!=0 && i+k>=0 && j+l>=0 && i+k<=SIZE-1 && j+l<=SIZE-1){
284                             if(k<0&&((l>-2&&k<-1)||l==0||l==-1)){
285                                 d[0]+=pheromone[i+k][j+l];
286                             }else if(k>0&&((l<2&&k>1)||l==0||l==1)){
287                                 d[1]+=pheromone[i+k][j+l];
288                             }else if(l<0&&((k<2&&l<-1)||k==0||k==1)){
289                                 d[2]+=pheromone[i+k][j+l];
290                             }else if(l>0&&((k>-2&&l>1)||k==0||k==-1)){
291                                 d[3]+=pheromone[i+k][j+l];
292                             }
293                         }
294                     }
295                 }
```

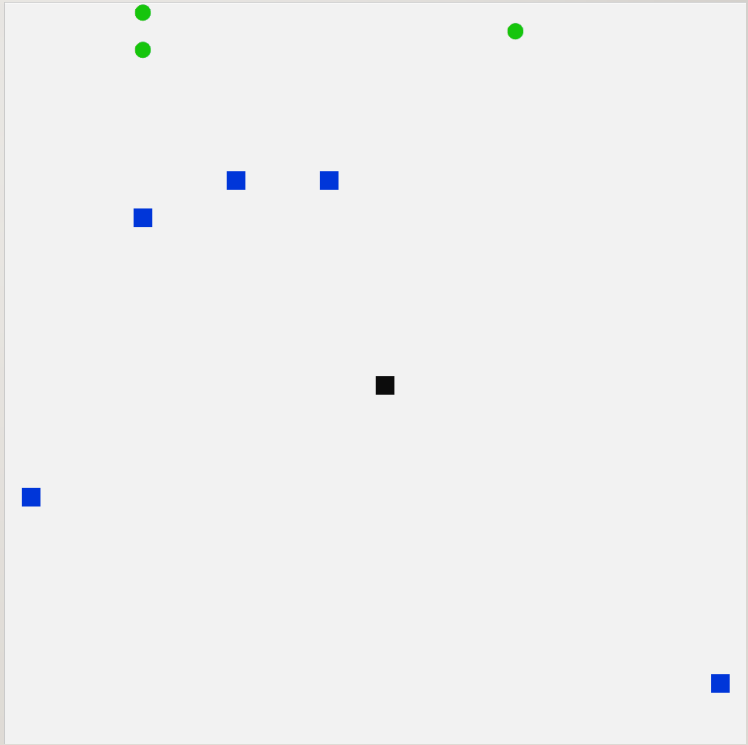
移動策略

個體的移動策略分為以下幾種：

1. 搜尋食物
2. 搜尋費洛蒙
3. 皆無搜尋到
4. 搬運食物回巢的個體

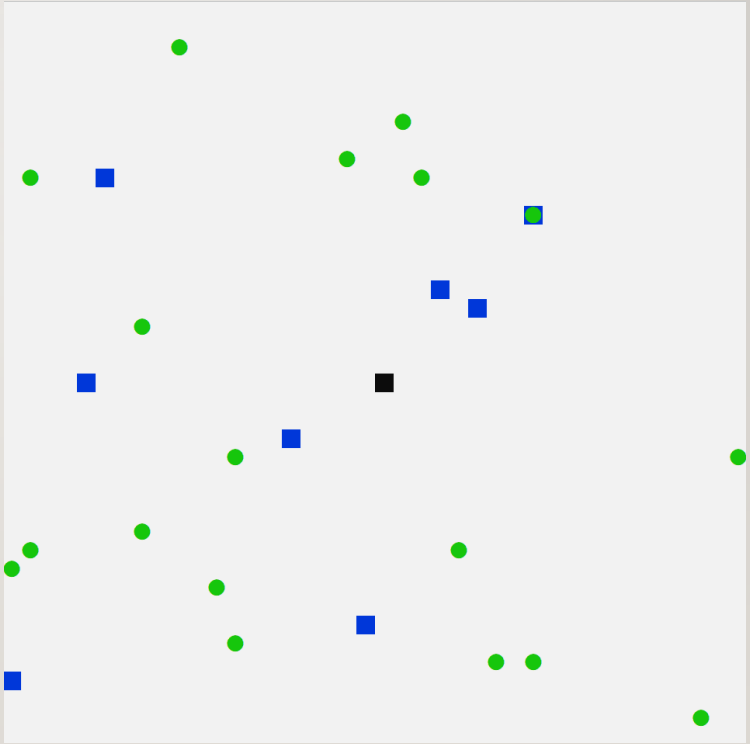
移動策略1-直行

- 個體直線前進，除非前進方向碰到障礙物或是其他個體才會轉彎



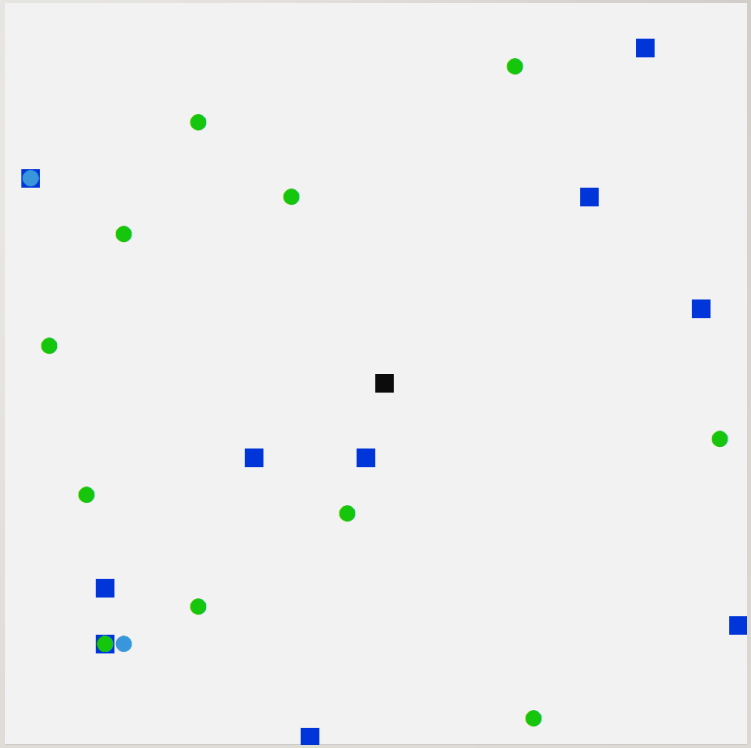
移動策略2-十五步

- 個體每走十五步會轉彎，若直行方向碰到障礙物或是其他個體也會轉彎



移動策略3-隨機

- 個體根據下列機率決定前進方向
- 直行機率：1/2
- 左轉機率：1/5
- 右轉機率：1/5
- 後退機率：1/10

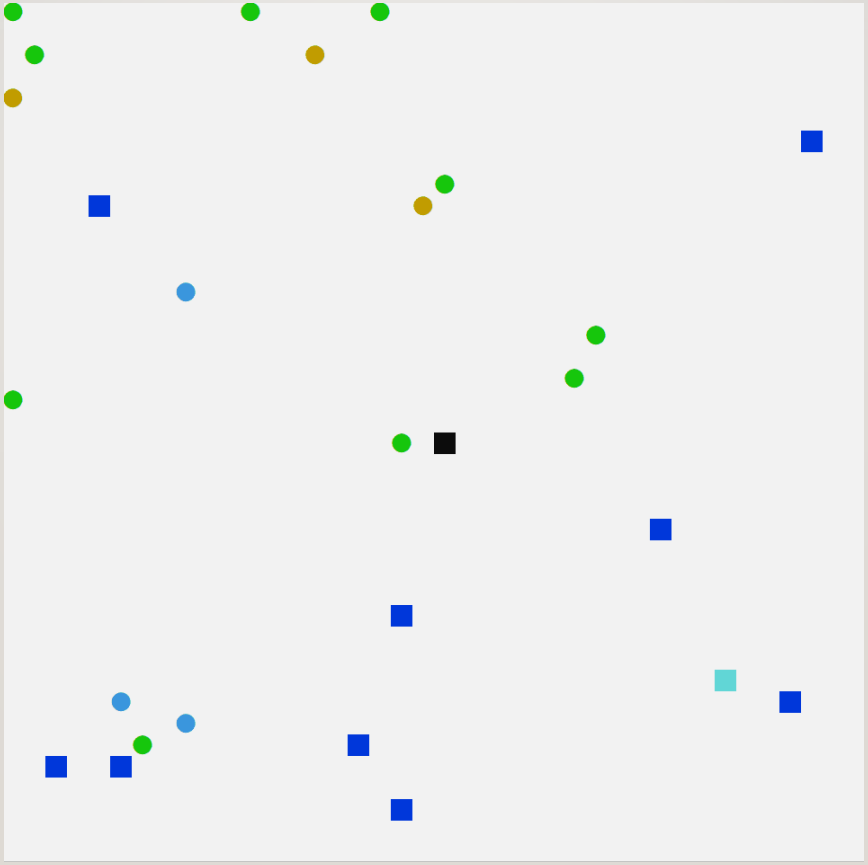


移動策略的改變

- 個體除了會依據不同的狀態改變(滿血、半血、殘血)自身的顏色，同時也會改變移動策略

```
188         }else if((ant1[i][j]=='u' || ant1[i][j]=='d' || ant1[i][j]=='l' || ant1[i][j]=='r')){       //螞蟻沒有找到食
189             if(timeinterval1[i][j]==80||timeinterval1[i][j]==160){   //換一種新的移動策略
190                 do{
191                     a=rand()%3+1;
192                 }while(a==search1[i][j]);
193                 search1[i][j]=a;
194             }
195             d[0]=0, d[1]=0, d[2]=0, d[3]=0, direction=-1;
196             for(int step=1;step<=4;step++){          //以周圍食物作為判斷依據
197                 for(int k=-step; k<=step; k++){
198                     if(k==4||k==-4){
199                         continue;
200                     }
201                     for(int l=-step; l<=step; l++){
202                         if(l==4||l==-4){
203                             continue;
204                         }
```

執行結果



結論、問題

- 開發過程中衍生出以下問題有待解決：
  1.程式編寫方式導致執行速度過慢，畫面呈現效果未達理想
  2.若食物已空，個體會停留在原地直到費洛蒙消失，使搜尋效率降低
- 很充實，但不會推薦沒有任何經驗的學弟妹這堂課。

解決方法

- 另外設定費洛蒙時間，而非讓費洛蒙遞減(未達成)
- 讓領取最後一個食物的個體不要留下費洛蒙(已達成)
- 讓費洛蒙的出值隨巢穴的位置改變(效果有限)

THE END

謝謝聆聽