
CUDA 驅動蟻群模擬器

鄭適其
309612092
土木工程學系
國立陽明交通大學

Contents

1 簡介	2
2 演算法細節	2
2.1 定義	2
2.2 世界更新方式	2
2.2.1 生成移動場	2
2.2.2 費洛蒙殘留	2
2.2.3 費洛蒙揮發	2
2.3 移動方式	3
2.3.1 區域極值問題	3
2.3.2 邊界問題	3
2.3.3 缺乏探索問題	3
2.3.4 LH 和 LF 的差異	3
2.4 蟻群自我調節方式	3
2.4.1 螞蟻的出生	3
2.4.2 螞蟻的死亡	3
2.4.3 螞蟻吃掉食物	3
3 GUI 介面	4
4 實驗記錄	5
4.1 地圖 0	5
4.2 地圖 1	7
4.3 地圖 2	9
5 程式碼	11
5.1 程式簡介	11
5.2 使用方式	11

1 簡介

在夏天的戶外，我們常常可以看見螞蟻們成群結隊、井然有序地把食物搬到牠們的巢穴中，不禁讓人感嘆大自然的奧妙。但是我卻也非常好奇這些小小的螞蟻怎麼會有如此的智慧，在查過資料後，大概了解了螞蟻是依靠在走過的路上留下特殊的費洛蒙，來指引同伴食物或巢穴的位置，並藉由控制留下費洛蒙的量以及運用費洛蒙會揮發的特性，構建出一條往返食物與巢穴的橋樑。

本專案就是要透過 Python 中的 Tkinter 視窗套件視覺化這樣生物的機制，並且在這之中使用 CUDA 平行運算來加速取得人工螞蟻們的行徑方向。不過計算速度在 Python 中仍然是短板，因此本專案對真實的螞蟻仍然做了簡化，並非完整的螞蟻找食物機制，但是可以確保人工螞蟻們能在自訂義的 2D 地圖上找到往返食物與巢穴的路。

2 演算法細節

2.1 定義

人工螞蟻 (以下簡稱**螞蟻**) 有分成兩種狀態:

- (a) **尋找食物狀態 (Look at Food, LF)**，即未找到食物的螞蟻。
- (b) **尋找巢穴狀態 (Look at Home, LH)**，即已找到食物的螞蟻。

費洛蒙也分成兩種:

- (a) **指引食物 (Guide to Food, GF)**
- (b) **指引巢穴 (Guide to Home, GH)**

其他定義:

- **速度**: 包含移動距離與方向。

重點目標: LF 的目標是不斷找尋一定範圍內最大 GF 所在的方向，而 LH 則是不斷尋找一定範圍內最大 GH 所在的方向。

2.2 世界更新方式

2.2.1 生成移動場

在螞蟻們移動前，會依照地圖上殘留的費洛蒙來生成每個位置 (像素) 對應的最大費洛蒙方向，這個動作是由 CUDA 平行運算完成，這也是驅動整個蟻群移動的**核心引擎**。每個位置對應到一組長度為 3 的向量，各元素對應的意義為:

$$[\text{是否附近所有值皆相等}, \text{最大值所在} X, \text{最大值所在} Y] \quad (1)$$

第一個值若是 True，則在那個位置上的螞蟻會隨機移動。而自身位置不會被列入考慮，避免可能出現螞蟻不會移動的情況。另外，地圖邊界外、牆壁內的位置也不考慮。

2.2.2 費洛蒙殘留

每隻螞蟻每次移動時都會留下費洛蒙，LF 會留下 GH，LH 會留下 GF。而每走一步留下的費洛蒙會越來越少，這麼做的目的是為了使越接近食物或巢穴的地方，費洛蒙殘留量越高。所以螞蟻向著費洛蒙愈大的地方走，就可以解釋成離食物或巢穴愈接近。

2.2.3 費洛蒙揮發

每次 iteration 都會讓地圖上的費洛蒙揮發一些，是蟻群能找到最佳的路徑的關鍵。因為根據 2.2.2 可知，路徑長度愈長，費洛蒙補充的越少，因此若已知地圖每個位置的揮發率相同，則長度較長的路徑會越容易消失，而較短路徑會留下。

2.3 移動方式

最主要的走法還是要瞄準相對目前位置的一定範圍內最大費洛蒙方向 (為了簡化計算，此範圍是正方形)，但是單純的這麼做會遇到一些問題。

2.3.1 區域極值問題

首先是最佳化中最惡名昭彰的問題: **Local Maximum** 與 **Local Minimum**，而因為螞蟻的目標是找最大值，所以只有 local maximum 問題。因此本專案參考了深度學習中 optimizer 常用的技巧 **momentum**，也就是藉由將上個 iteration 的速度與本次 iteration 的速度做線性組合做為新的速度在本次 iteration 中使用 (如 eq(2)所示)。從實驗的結果看來，momentum 確實能有效使螞蟻脫離 local maximum。

$$v'_t(s_t) \leftarrow \alpha \cdot v_{t-1}(s_{t-1}) + (1 - \alpha) \cdot v_t(s_t), \quad 0 \leq \alpha \leq 1 \quad (2)$$

v 是速度， α 是權重， s 是位置向量， t 是迭代次數。另外，食物與巢穴位置是 global maximum，因為在此專案中，假設它們的費洛蒙值始終維持無窮大，讓到它們附近的螞蟻可以被快速吸引過去。

2.3.2 邊界問題

由於 Tkinter 的 Canvas 範圍是有限的，而我們不能螞蟻走出邊界，另外為了檢驗演算法的效果，本專案設計了可以在地圖上加上牆壁的機制，而螞蟻不能穿過牆壁。也就是說，螞蟻們是不能隨便走動的，因此所有的螞蟻都具有碰牆反彈的機制。但有得時候螞蟻仍然會走出界或是走到牆壁裡，因此在走之前要檢查 sample 到的速度是否會走出界或進到牆壁，如果有那就讓螞蟻在這個 iteration 靜止不動，但是保留一半的速度作為 momentum。

2.3.3 缺乏探索問題

為了增強蟻群的探索 (exploration)，螞蟻的每次行動都有隨機性，也就是螞蟻會有一定的機率不按照最大費洛蒙方向走。如果所有螞蟻只會按照現有的路線走，那就無法找到路徑更短的新路線以加快食物取得速度，因此行動的隨機性是相當重要的。另外，經由實驗發現 momentum 也有增強蟻群探索速度的效果。故移動速度的期望值可寫成:

$$\mathbb{E}_p\{v_t(s_t)\} \stackrel{def}{=} (1 - p) \cdot v'_t(s_t) + p \cdot \text{Clip}(\text{Random Vector}, -speed, speed) \quad (3)$$

其中 p 為隨機移動機率， $speed$ 是移動距離。

2.3.4 LH 和 LF 的差異

為了加速 LF 找到食物，LF 在 GF 的向量(1)第一個元素為 True 時，會採取 GH 最大值方向的反向 ($-\mathbb{E}_{(1-p)}^{GH}\{v_t(s_t)\}$) 作為移動目標，當附近有出現 GF 時就改用 $\mathbb{E}_p^{GF}\{v_t(s_t)\}$ 。而 LH 沒有相對應的機制，依然只依靠 $\mathbb{E}_p^{GH}\{v_t(s_t)\}$ 作為移動依據。

2.4 蟻群自我調節方式

2.4.1 螞蟻的出生

當收集到的食物達到足夠的量，則可以新出生更多螞蟻擴大蟻群，可以及時補充地圖上的費洛蒙，使蟻群可以找到更遠的食物，或是更加鞏固現有的道路。

2.4.2 螞蟻的死亡

當 LF 太久沒有找到食物，則會有機率餓死。因為食物量不變，每餓死一隻螞蟻就會有一隻新螞蟻在巢穴出生。可以重新補充地圖的費洛蒙。

2.4.3 螞蟻吃掉食物

當 LH 太久沒找到巢穴，則他會把食物吃掉以存活，並且將狀態從 LH 轉成 LF，並在找到新食物前不會再留下任何費洛蒙。

3 GUI 介面



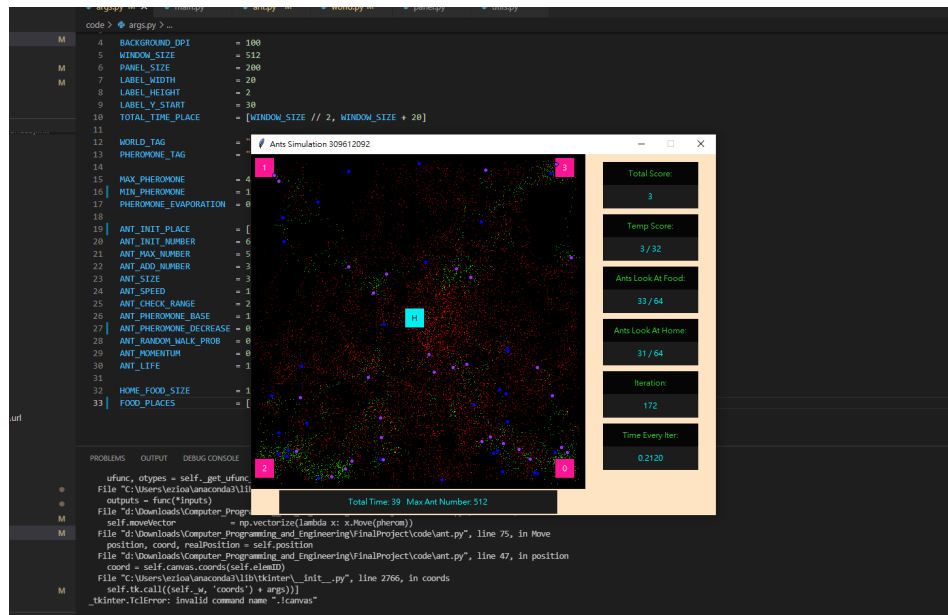
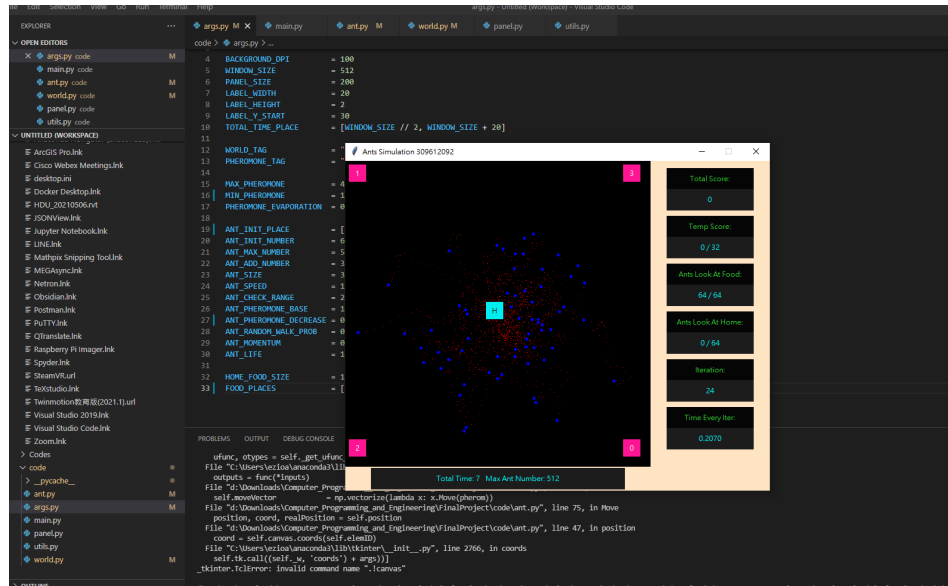
Figure 1: GUI 介面

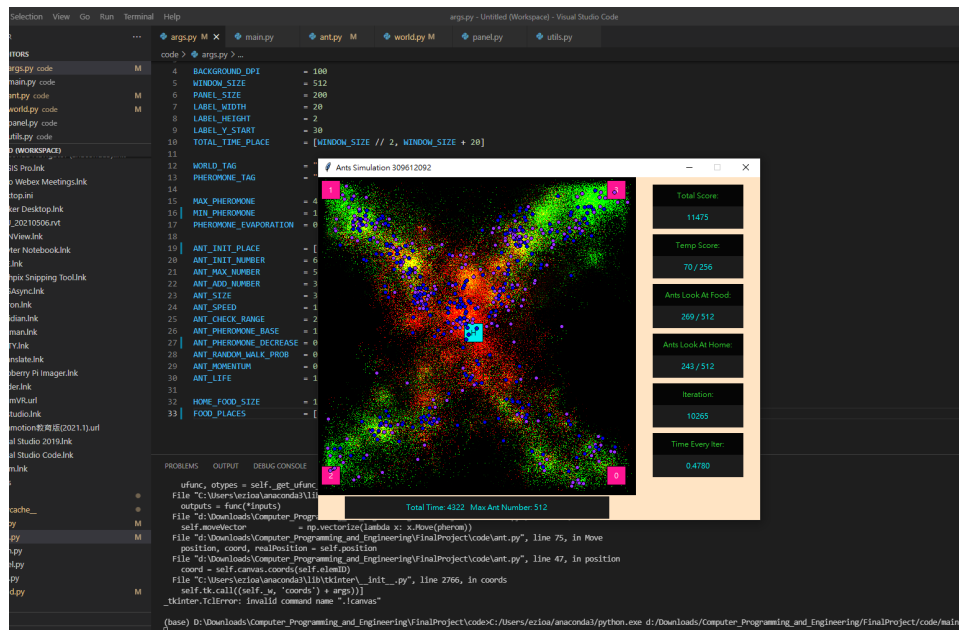
如 Figure 1 所示，程式實際執行起來會出現這樣的 GUI 介面:

- 右方面板:
 - Total Score: 總得分 (每帶一單位食物回巢得一分)
 - Temp Score: 暫時得分，當累積到總螞蟻數的一半會產生一群新螞蟻，並且分數歸零
 - Ants Look At Food: 有多少螞蟻處於 LF 狀態
 - Ants Look At Home: 有多少螞蟻處於 LH 狀態
 - Iteration: 目前迭代次數
 - Time Every Iter: 每次迭代所需時間
- 下方面板:
 - Total Time: 總經過時間
 - Max Ant Number: 最大螞蟻數量
- 中間面板:
 - 藍點: LF
 - 綠點: LH
 - 白色: 牆壁
 - 紅點: GH
 - 綠點: GF
 - 水藍色正方形: 螞蟻巢穴
 - 粉紅色正方形: 食物

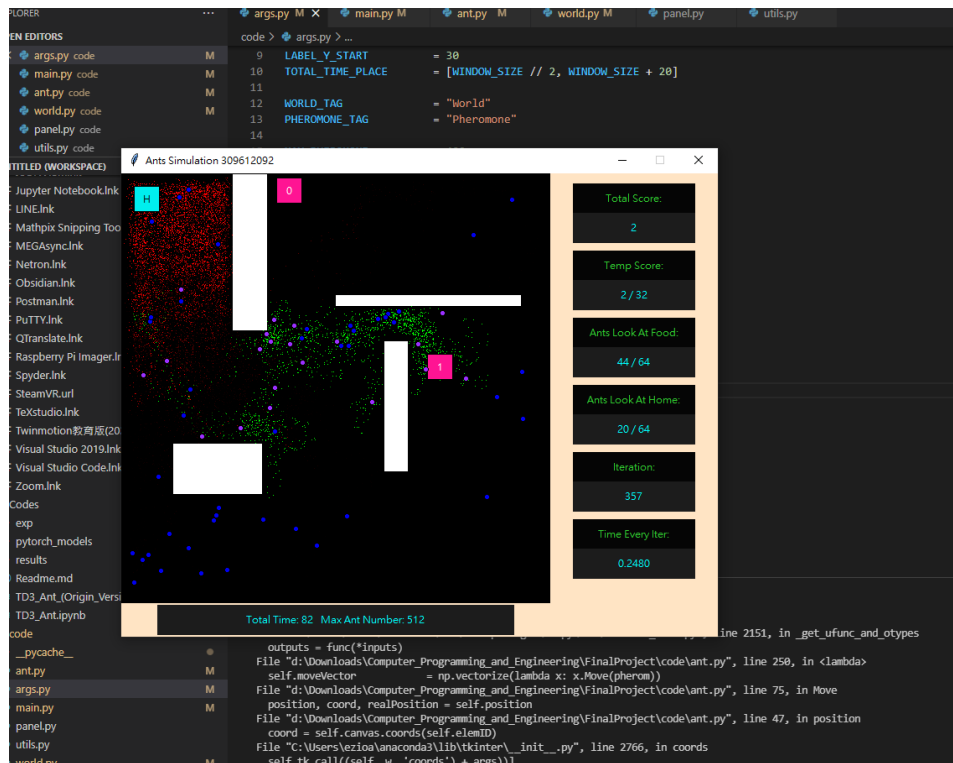
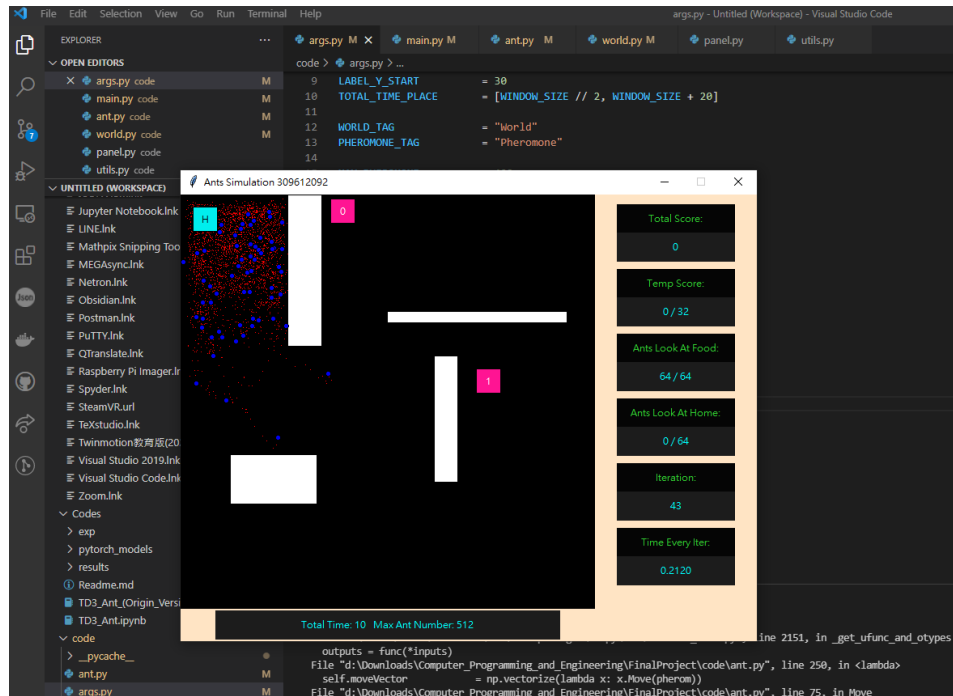
4 實驗記錄

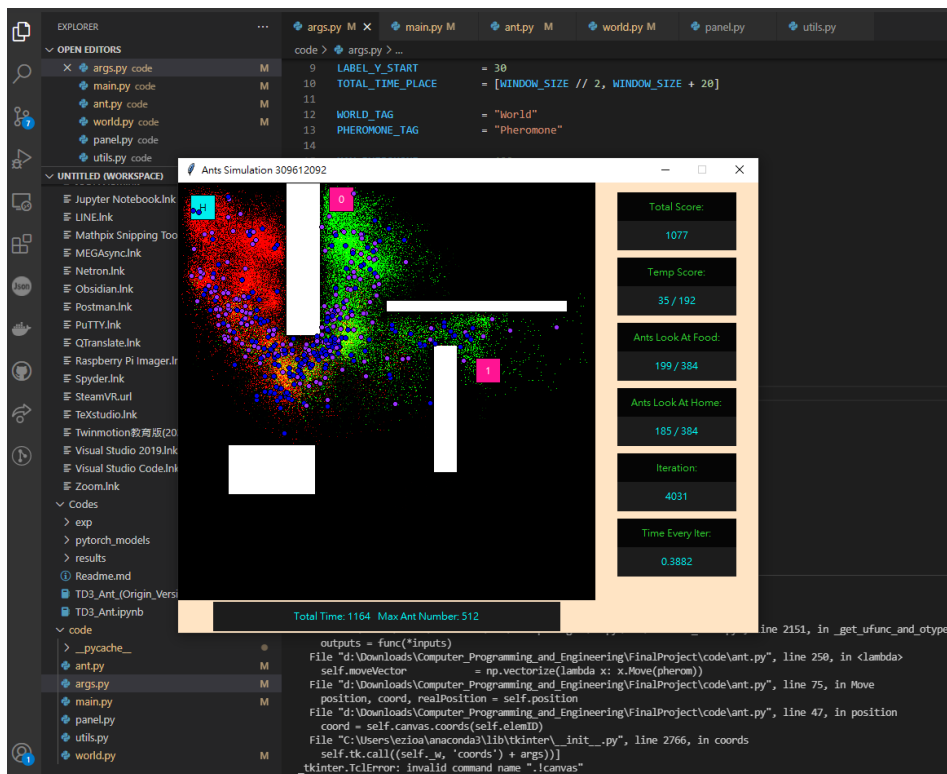
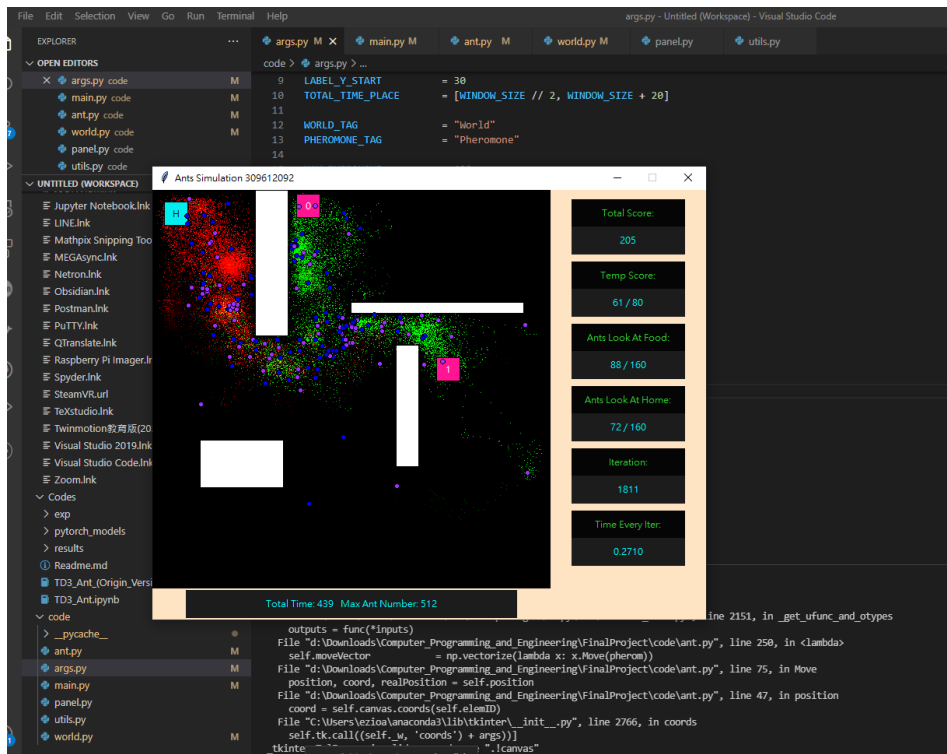
4.1 地圖 0



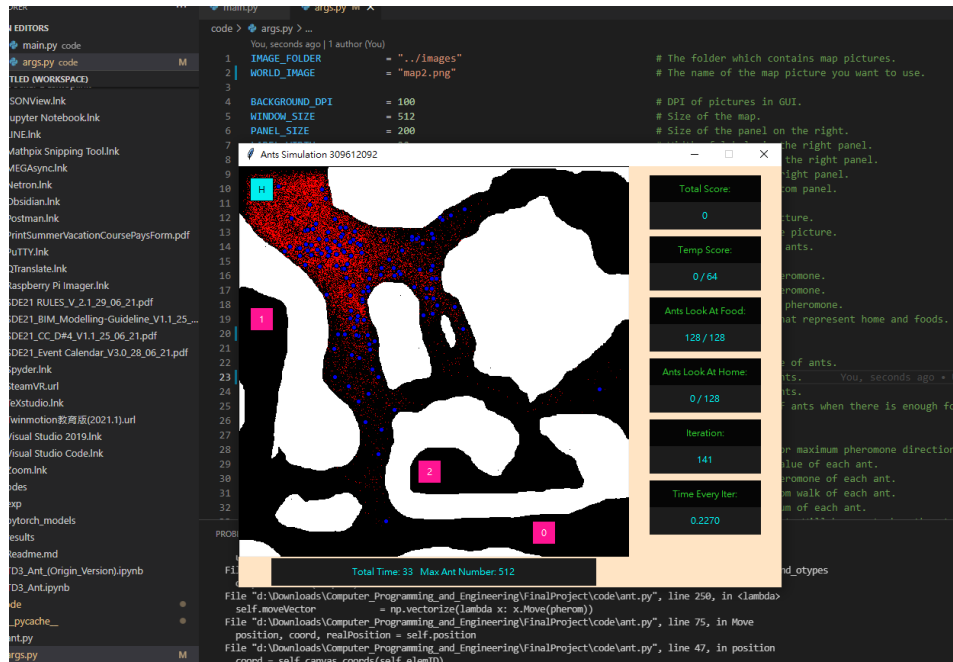


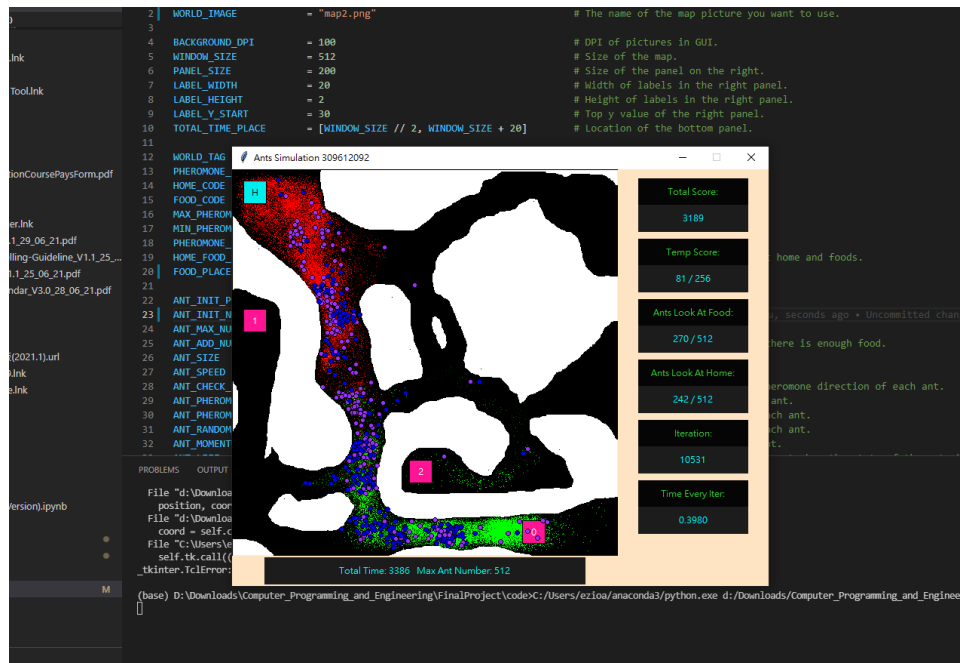
4.2 地圖 1





4.3 地圖 2





5 程式碼

5.1 程式簡介

程式檔案用途:

- **args.py**: 參數設置
- **main.py**: 主程式循環
- **ant.py**: 螞蟻、蟻群相關程式碼
- **world.py**: 世界、費洛蒙地圖相關程式碼
- **panel.py**: GUI 右方與下方 panel 相關程式碼
- **utils.py**: 雜項

自訂義 6 大 class 介紹:

- **BG_Manager**: 背景管理 (World 與 PheromoneMap 的父類別)
- **World**: 世界地圖
- **PheromoneMap**: 費洛蒙地圖
- **Ant**: 人工螞蟻
- **AntColony**: 蟻群統籌管理
- **Panel**: GUI 右方與下方儀表板

5.2 使用方式

- (a) 在 args.py 中調整參數 (參數意義寫在程式碼註解中)。
- (b) 調整參數後執行 main.py。
- (c) 彈出 Tkinter 的視窗即執行成功。

若要自行設計地圖，請注意地圖只能是二值圖片 (即只有黑色與白色)，白色部分會被當作牆壁，黑色部分為螞蟻可走範圍，此設定無法由參數更改。由於簡化計算之故，盡量不要把牆壁畫得太薄，厚度至少大於 args.py 裡的 ANT_SPEED(單位: pixel)，否則螞蟻可能會穿牆。最後記得要調整 args.py 中的 FOOD_PLACES(食物位置，可有多個) 以及 ANT_INIT_PLACE(螞蟻巢穴位置，只能有一個)。