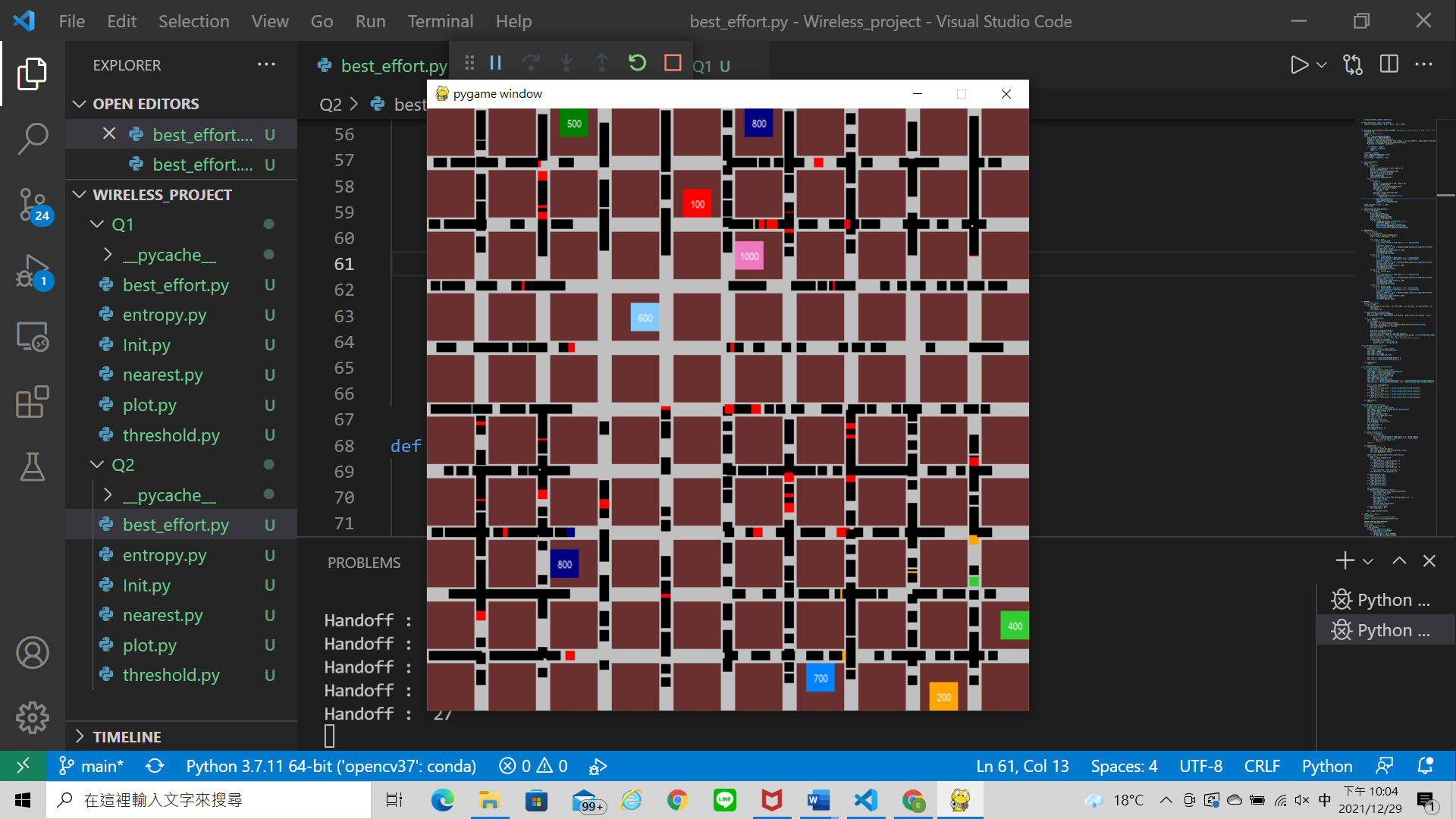
無線通訊網路程式模擬

使用python pygame :



參數設定:

1.FPS = 100，模擬1min。(FPS設置1/3為真實情況(模擬起來太慢)，因此加速300倍🡪FPS=1/3\*300=100)

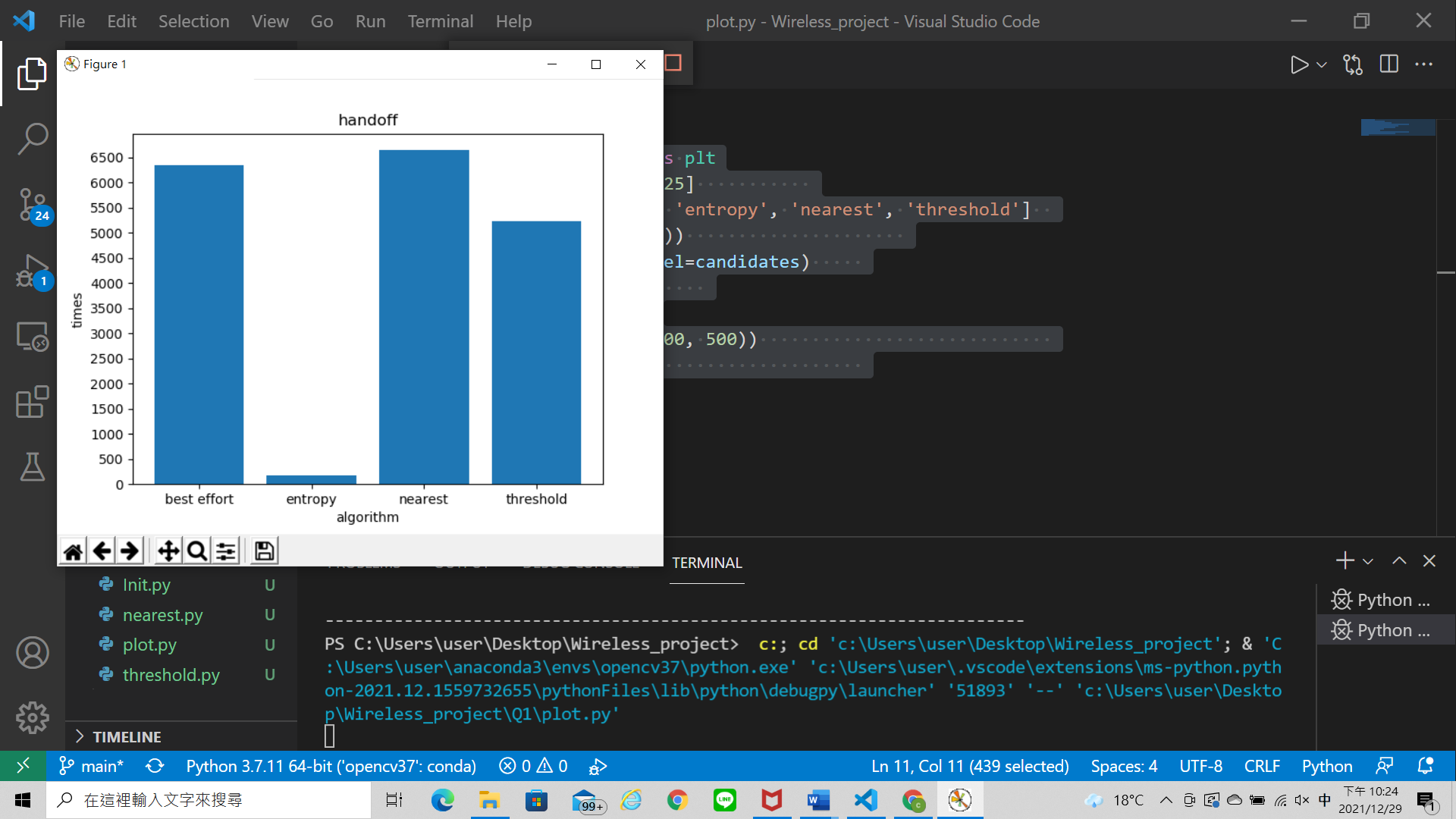
2.Lambda=1/4。(原本題目設定為1/12，若FPS=1/3時(真實情況)，表示每3秒有1/4的arrival rate)

3.Speed=1.2。(原本題目設定為0.02km/sec，但是我的視窗大小為50\*50，題目設定為2.5km\*2.5km，所以比例為20倍，因此0.02\*20=0.4，但是因為FPS=1/3(真實情況)，所以速度要再乘3 🡪 0.4\*3=1.2)

Threshold、entropy、Ptransmit與題目設定相同。

4種algorithm分析:

Q1:



Q2:

一張含有 文字, 螢幕擷取畫面, 監視器 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 監視器 的圖片

自動產生的描述

1.Minium(Threshold):強度夠的時候不切換(Pmin=100dB)

2.Best effort:訊號強者提供訊號

3. Entropy:兩基地台訊號落差值超過閾值時候切換(閾值E=25dB)

4.Nearest:選最近的base station

解析 :

兩題中做最少交換次數的algorithm都是entropy，因為他的交換條件是兩基地台訊號落差超過25db(蠻大的)才會交換，條件最嚴苛；而threshold為交換次數第2少的原因是因為只要訊號強度夠就不切換(符合條件才切換)，所以會比best effort交換次數少(比best effort保守派)；Best effort是訊號強的基地台提供訊號，所以只要車子發現訊號比原本基地台提供的訊號強就會切換，所以會常常handoff；而我自己設定的nearest方法是類似best effort，不過條件變成距離不是訊號 : 永遠選最近的基地台，不管訊號強度如何，所以只要車子開到哪裡，就會連最近的基地台，最常切換。我當初會設計連最近的基地台是因為我以為這是事實(哈哈，原來不是阿)，雖然交換次數最多，但是因為老師說過電信業者在架設基地台的時後統計過近距離的人會比遠距離的人更容易通話(譬如說都在成大校園裡的兩個人打電話機率 > 台南東區打電話給北區)，所以如果都連距離最近的，兩電話間的資料傳輸就不用轉換基地台，會省一些轉換的損耗，因此我認為也是不錯的algorithm。

3個algo的好壞:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Best effort | Threshold | Entropy |
| Switch cause overhead | worse | Medium | better |
| 訊號品質 | best | Medium?(需要看臨界值是多少) | Worse?( 需要看臨界值是多少) |