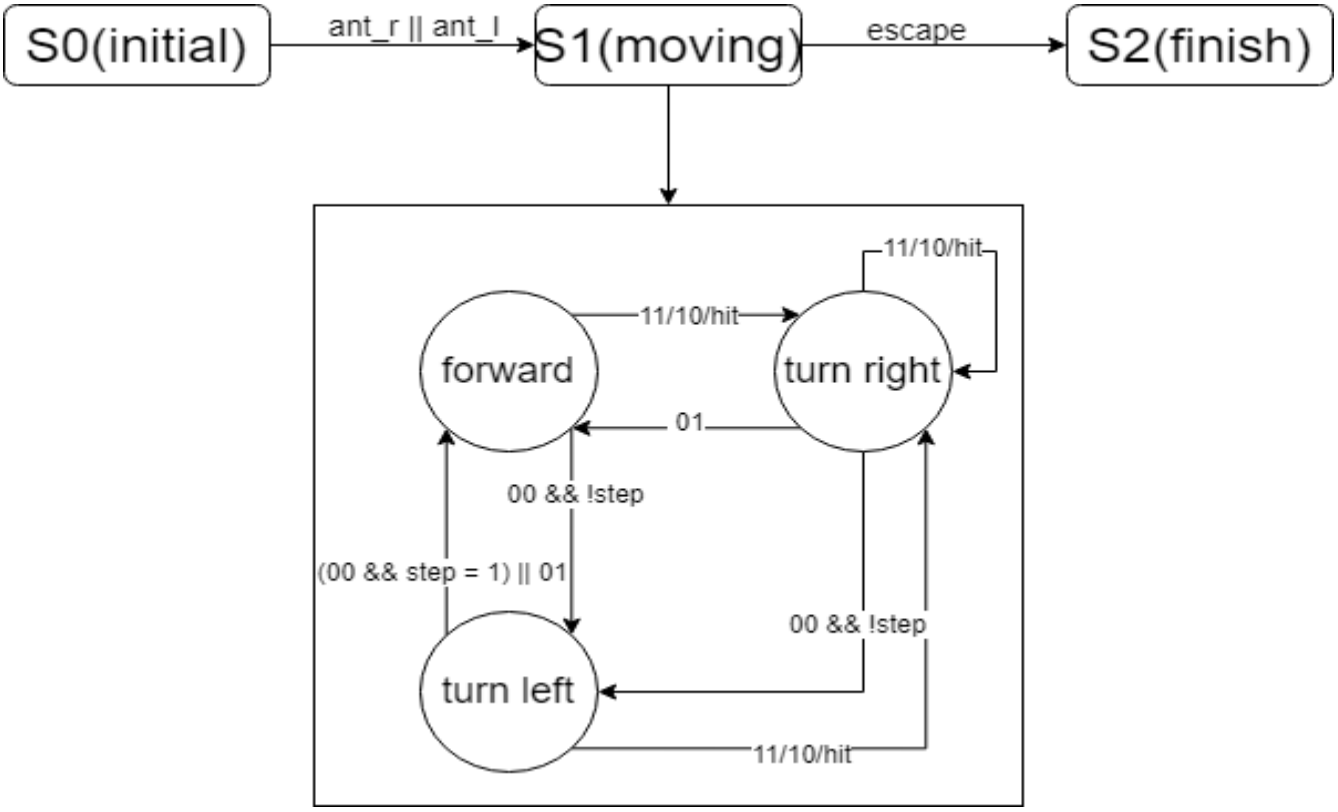
EECS1010 Design Assignment 03

Student ID: 107070002 Name: 莊英暄

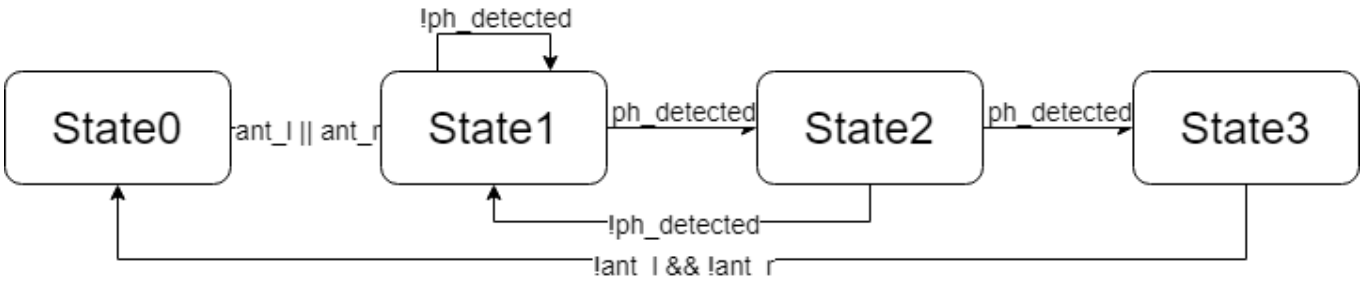
1. Design Concept and Description



我的想法是在一開始的狀態為 initial，在這個 state 時會一直往前走直到觸角感覺到任何東西，也就是 lr != 00，並開始進行 moving 階段。

在 moving 階段時，我盡量以靠左側行走，有三種移動方式，當前方遇到牆或是右側有牆時，會選擇向右轉直到牆壁在左側，而牆壁在左側時則向前走。當遇到轉角的時候會有兩個步驟，會先往前到空地，這時 step = 0 代表在第一步驟，會向左轉然後將 step 設為 1，在 step = 1 時，會向前走並把 step 設回 0，這樣就能在遇到轉角時順利做完轉向以及向前。

最後在到達終點時 escape 會被設為 1，這時就代表順利逃跑，並結束程式。



Challenge2:

因為多了會偵測掉落物的東西，因此只用 3 個 state 不夠，需要新增一個 state 來讓偵測物品更加完整，如圖片所示，state0 為 initial state，只會往前，state1 以及 state2 的行為模式與普通模式的行走一樣，state3 則是背向牆面並回到 state0，重新出發。

2. Simulation and Discussion

第一迷宮

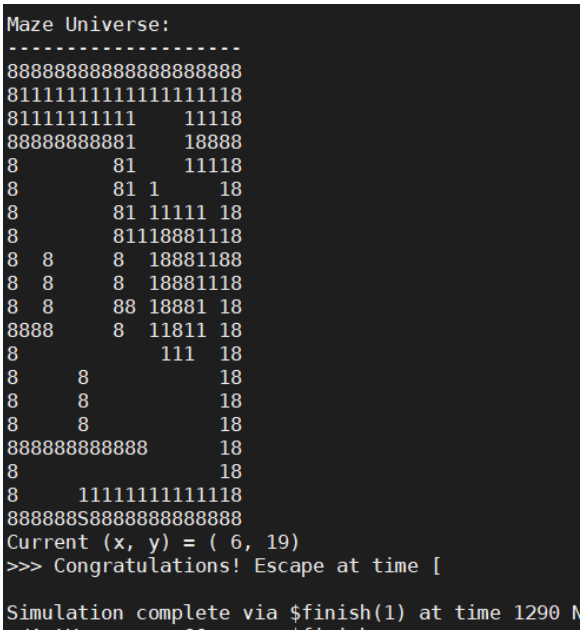
```
Maze Universe:
--------------------
8888888888888888888
8    88           8
8    88           8
8    8         88888
8                 8
8       E         8
888               8
888               8
8                 8
8    8         88888
8  888            8
8  8              8
8  88             8
8   8            88
8   8           888
8  88           888
*  8            888
8  8  8  8     8888
8888888888888888888
Current (x, y) = ( 7,  5)
[Step    1] Move Forward Position ( 8,  5) East   LR=00
[Step    2] Move Forward Position ( 9,  5) East   LR=00
[Step    3] Move Forward Position (10,  5) East   LR=00
[Step    4] Move Forward Position (11,  5) East   LR=00
[Step    5] Move Forward Position (12,  5) East   LR=00
[Step    6] Move Forward Position (13,  5) East   LR=00
[Step    7] Move Forward Position (14,  5) East   LR=00
[Step    8] Move Forward Position (15,  5) East   LR=00
[Step    9] Move Forward Position (16,  5) East   LR=00
[Step   10] Move Forward Position (17,  5) East   LR=11
[Step   11] Halt         Position (17,  5) East   LR=11
[Step   12] Turn Right   Position (17,  5) South  LR=10
[Step   13] Move Forward Position (17,  6) South  LR=10
[Step   14] Move Forward Position (17,  7) South  LR=10
[Step   15] Move Forward Position (17,  8) South  LR=11
[Step   16] Turn Right   Position (17,  8) West   LR=10
[Step   17] Move Forward Position (16,  8) West   LR=10
[Step   18] Move Forward Position (15,  8) West   LR=10
```

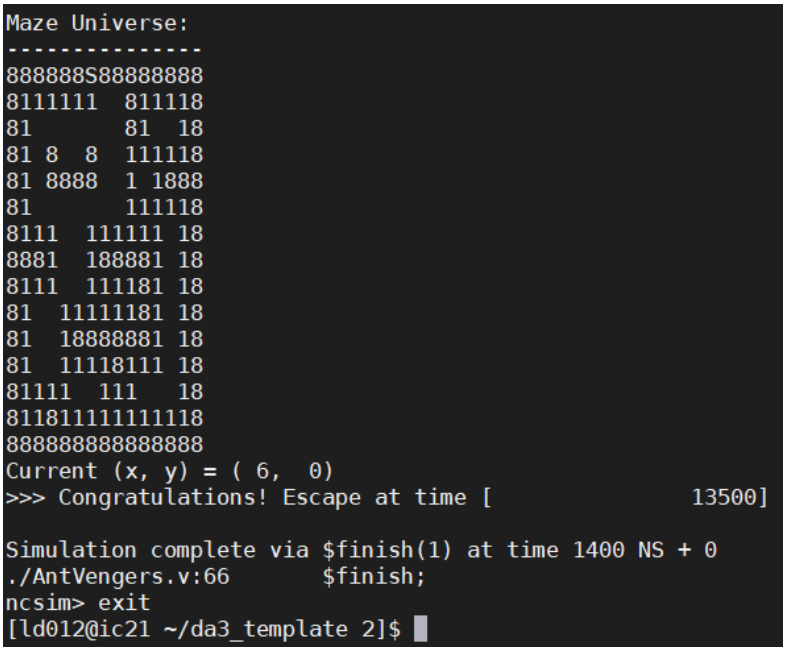
第二迷宮

```
Maze Universe:
--------------------
88888888888888888888
8                   8
8                   8
8888888888      8888
8         8         8
8         8         8
8         8         8
8         8         8
8  8      8        88
8  8      8         8
8  8      88        8
8888      8         8
8                  888
8      8           888
8      8             8
8      8             8
888888888888        8
8  N                8
8                   8
888888*8888888888888
Current (x, y) = ( 3, 17)
[Step    1] Turn Right   Position ( 3, 17) East   LR=10
[Step    2] Halt         Position ( 3, 17) East   LR=10
[Step    3] Move Forward Position ( 4, 17) East   LR=10
[Step    4] Move Forward Position ( 5, 17) East   LR=10
[Step    5] Move Forward Position ( 6, 17) East   LR=10
[Step    6] Move Forward Position ( 7, 17) East   LR=10
[Step    7] Move Forward Position ( 8, 17) East   LR=10
[Step    8] Move Forward Position ( 9, 17) East   LR=10
[Step    9] Move Forward Position (10, 17) East   LR=10
[Step   10] Move Forward Position (11, 17) East   LR=10
[Step   11] Move Forward Position (12, 17) East   LR=00
[Step   12] Turn Left    Position (12, 17) North  LR=00
[Step   13] Move Forward Position (12, 16) North  LR=10
[Step   14] Move Forward Position (12, 15) North  LR=00
[Step   15] Turn Left    Position (12, 15) West   LR=00
[Step   16] Move Forward Position (11, 15) West   LR=10
[Step   17] Move Forward Position (10, 15) West   LR=10
```

最麻煩的部分是經過轉角得時候，因為不能同時轉彎並前進，因此需要拆成兩個階段執行，也就是 step0 為轉彎而 step1 為直走。

Challenge discuss:





一開始先正常行走，當遇到重複走過的點的時候會選擇遠離牆壁並回到 state0 直走直到靠近牆壁。判斷方式為當現在動作為前進並且踏在地點為已經經過的地區時會向右轉直到背向牆壁，接著進入 state0 一直往前直到到達下一個牆壁附近。

我將繞牆拆成兩個部分，state1 代表一般的繞牆，當採到已經走過的點時會先進入 state2 進行判斷，state2 時會判斷你在繼續下一步的時候是否還是在已經踩過的點，如果是的話就代表開始循環的然後進入 state3，如果不是就代表可能只是在轉彎而已，所以回到 state1。

3. Summary

這次作業花了我不少時間思考，也讓我更了解硬體 code 執行的邏輯，最一開始在寫的時候一直覺得很難，但是後來仔細思考後發現其實沒想像中的難，只是在遇到處理轉角時卡了非常的久。最後使用了 reg step 當作判斷條件來執行轉彎動作。

在處理 challenge 上，原本只打算處理 possible solution，但是後來發現好想可以直接寫出來，雖然花了不少時間，但是最後還是打出來了，我覺得最麻煩的是要在甚麼情況下進入 state3 是最複雜的，因為轉向就會使地面留上物品，導致如果只用一個 state 行走會常常跑出 wall island，最後千辛萬苦才終於寫出一個能跑也比較理想也穩定的走法。