



Maze Miner

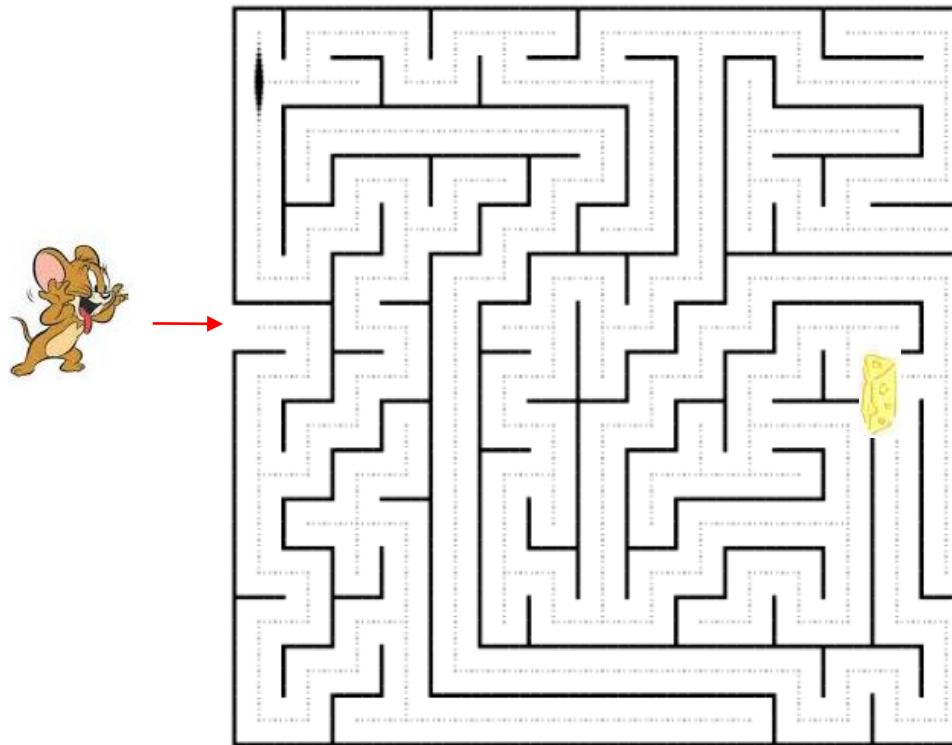
Speaker: Yen Ning Wan

Advisor: Jen-Wei Huang



Introduction

- Move the mouse to find the cheese !!





Maze Example_Representation

12,12

1,1,1,1,1,1,1,1,1,1,1,1
1,1,0,0,0,0,0,0,0,0,0,1
1,0,2,0,1,1,1,1,1,1,0,1
1,1,0,1,0,0,0,0,0,0,0,1
1,1,0,1,1,1,1,0,1,1,1,1
1,0,0,0,0,0,0,0,0,0,0,1
1,0,1,1,1,0,1,1,1,1,0,1
1,0,1,0,1,0,0,0,0,0,0,1
1,0,0,0,1,0,1,1,1,1,1,1
1,0,1,0,1,0,1,0,0,0,0,1
1,1,1,1,1,1,1,1,1,1,1,1

Number	Representation
0	Road
1	Wall
2	Start Point
3	End Point



Maze Example_Array

12,12

```
1,1,1,1,1,1,1,1,1,1,1,1  
1,1,0,0,0,0,0,0,0,0,0,1  
1,0,2,0,1,1,1,1,1,1,0,1  
1,1,0,1,0,0,0,0,0,0,0,1  
1,1,0,1,1,1,1,0,1,1,1,1  
1,0,0,0,0,0,0,0,0,0,0,1  
1,0,1,1,1,0,1,1,1,1,0,1  
1,0,1,0,1,0,0,0,0,0,0,1  
1,0,0,0,1,0,1,1,1,1,1,1  
1,0,1,0,1,0,1,0,0,0,0,1  
1,0,1,0,1,0,0,0,1,0,3,1  
1,1,1,1,1,1,1,1,1,1,1,1
```

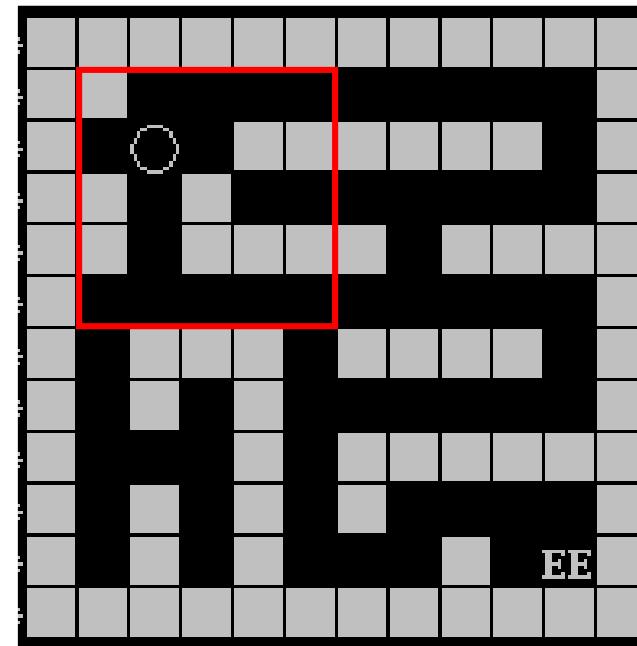
0	1	2	3	4	5
0	(0,0)	(1,0)	...		(5,0)
1	(0,1)				
2	:				
3					
4					
5	(0,5)				



Maze Example_Show

12, 12

```
1,1,1,1,1,1,1,1,1,1,1,1  
1,1,0,0,0,0,0,0,0,0,0,1  
1,0,2,0,1,1,1,1,1,1,0,1  
1,1,0,1,0,0,0,0,0,0,0,1  
1,1,0,1,1,1,1,0,1,1,1,1  
1,0,0,0,0,0,0,0,0,0,0,1  
1,0,1,1,1,0,1,1,1,1,0,1  
1,0,1,0,1,0,0,0,0,0,0,1  
1,0,0,0,1,0,1,1,1,1,1,1  
1,0,1,0,1,0,1,0,0,0,0,1  
1,0,1,0,1,0,0,0,1,0,3,1  
1,1,1,1,1,1,1,1,1,1,1,1
```





Maze Example_Move

Input	Instruction
8	↑
2	↓
4	←
6	→

Maze Miner:

← ↑ → ↓ 4 8 6
 2

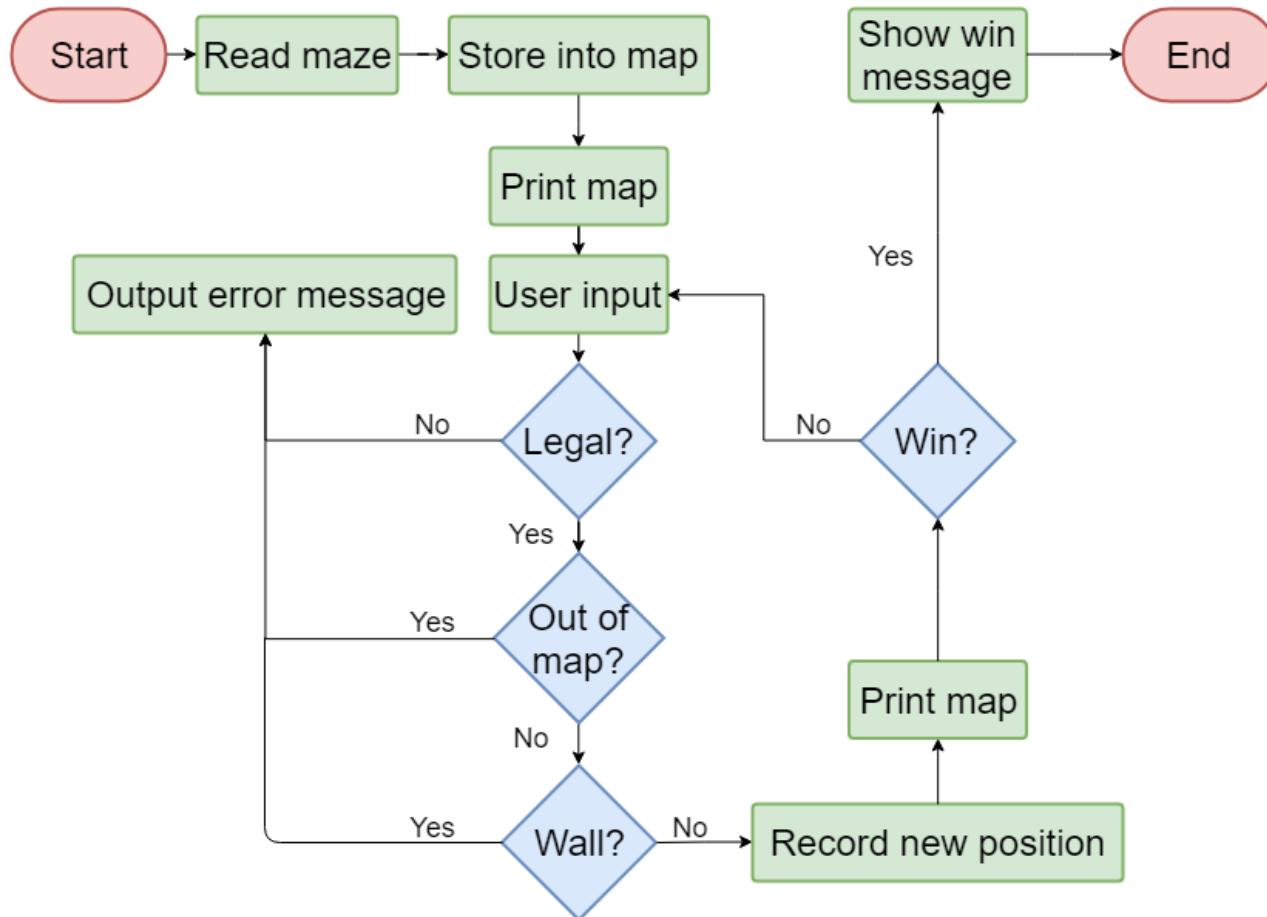
Vision:

```
*****  
*██* *  
* *○ *  
*██* *  
*██* *  
*██* *  
*****
```

Please select the next step: (8:up, 2:down, 4:left, 6:right):



Flow Chart





Uncompleted Program

```
Maze Miner:  
↑ ← → 4 8  
↓ 2 6  
  
Vision:  
*****  
* █ *  
* ○ *  
* █ *  
* █ *  
* █ *  
* █ *  
*****  
Please select the next step: (8:up, 2:down, 4:left, 6:right): 6  
please complete this part: I want to go right!  
請按任意鍵繼續 . . .
```



Main Steps

- Read the map (**DONE**)
- Print Maze (**DONE**)
- Implement uncompleted parts for each direction
 - Prevent users from going beyond the map (**Not Done**)
 - Prevent users from crossing the wall (**Not Done**)
 - Move for four direction (**Not Done**)
- Check if user reach to the destination (**Not Done**)



Your Mission

```
//注意!!!要開始修改程式碼了!!!-----  
//輸入總共有4種情況，選擇其中一種即可  
//往右走  
if (true) // check this line  
{  
  
    /*請刪除或註解掉以下的程式碼*/  
    cout << "please complete this part: I want to go right!" << endl;  
    break;  
    win = true;  
    /*請刪除或註解掉以上的程式碼*/  
  
    //右邊已是地圖外  
    if (true) // check this line  
    {  
  
        /*請刪除或註解掉以下的程式碼*/  
        cout << "please complete this part: do not go out the right map!" << endl;  
        win = true;  
        /*請刪除或註解掉以上的程式碼*/
```



Your Mission

```
//右邊是牆 不能走
else if (true) // check this line
{
    //*****請刪除或註解掉以下的程式碼*****
    cout << "please complete this part: do not across the right wall!" << endl;
    win = true;
    //*****請刪除或註解掉以上的程式碼*****

    //*****請依照註解需求完成以下的程式碼*****
    // please design and output your own error message
    //*****請依照註解需求完成以上的程式碼*****
}
```



Hint

```
//變數宣告-----  
int row; //記錄地圖的row數量  
int column; //記錄地圖的column數量  
int **map; //記錄整個地圖的array  
int start_row, start_col, dest_row, dest_col; //起始row & column, 結束row & column  
bool win = false; //用來判斷遊戲是否結束的boolean值  
string file;  
char temp;  
char comma[200][200];  
  
//可以使用的Function-----  
void loadMazeMap(); //讀取地圖資訊的Function  
void loadStartAndDest(); //讀取起始和結束位置資訊的Function  
void printMaze(int position_row, int position_column); //印出可目前可以看到的地圖範圍的Function
```



THANK YOU

