(Interpreted) requirements of the program

- Create a multithreaded program
- Program is to discover the surroundings of a jungle maze
- Program is to explore as much of the jungle maze terrain as possible, and mark the discovered area as barrier (‘#’), or danger (‘X’) accordingly
- When the program terminate, it should output a map of the explored jungle maze, as well as 1 ‘safe’ path, to traverse from Start to End locations.

Diagram / Illustrations of program design

Main features

```
parallels@ubuntu:~$ cd '/media/psf/Home/Dropbox/Programming/Workspace/212A3'
parallels@ubuntu:/media/psf/Home/Dropbox/Programming/Workspace/212A3$ ./PathFinder.sh

In file included from Maze.h:13:0,
                 from PathFinder.h:4,
                 from PathFinder.cpp:1:
Assignm3_Utils.h: In constructor ‘Point::Point()’:
Assignm3_Utils.h:19:21: warning: converting to non-pointer type ‘int’ from NULL [-Wconversion-null]
Assignm3_Utils.h:19:31: warning: converting to non-pointer type ‘int’ from NULL [-Wconversion-null]


Enter Number of Solutions to be submitted (Leave blank for default of 1): 1

     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
  0  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #
  1  #  S     #        #                                       #
  2  #        #     #  #        #  #        #  #  #     #  #  #
  3  #                 #           #           #        E     #
  4  #  #     #        #     X     #        X        #  #     #
  5  #           X     #  #  #     #  #  #  #  #              #
  6  #     #           #           #        #           #  #  #
  7  #     #  #  #     #  #  #     #  #     #     #  #  #  #  #
  8  #        #                                             #
  9  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #


_length  : 20
_breadth : 10

_startLocation : [ 1, 1 ]
_endLocation   : [ 17, 3 ]
```

Start of program: - Prompts user for number of solutions to be submitted
- Displays loaded maze

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
  0
  1     S
  2
  3                                                       E
  4
  5
  6
  7
  8
  9


_length  : 20
_breadth : 10

_startLocation : [ 1, 1 ]
_endLocation   : [ 17, 3 ]

No. of paths discovered : 0

Thread 'POOH' has been created !!
Total no. of steps : 1

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
  0
  1  #  S
  2
  3                                                       E
  4
  5
  6
  7
  8
  9
```

Start of path finding:

- Thread creation is shown
- Solution maze is shown as empty except for start and end points and the first barrier found by the thread (in this picture)

```
Thread 'TIGGER' has been created !!
Total no. of steps : 6

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
  0     #  #
  1  #  S  5
  2  #  1  4
  3  #  2  3                                              E
  4     #
  5
  6
  7
  8
  9

Total no. of steps : 6

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
  0     #  #
  1  #  S  5  #
  2  #  1  4
  3  #  2  3                                              E
  4     #
  5
  6
  7
  8
  9
```

Creation of another thread and further finding of path

```
Thread 'POOH' hits a dead end near [2, 8] !!
Total no. of steps : 11

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 0     #  #
 1  #  S     #
 2  #  1
 3  #  2  3                                                 E
 4     #  4
 5  #  6  5
 6  #  7
 7  #  8  #
 8  #  9 10
 9     #

Total no. of steps : 11

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 0     #  #
 1  #  S     #
 2  #  1
 3  #  2  3                                                 E
 4     #  4
 5  #  6  5
 6  #  7
 7  #  8  #
 8  #  9 10
 9     #  #
```

A thread finds a dead end

```
Thread 'POOH' stepped into DANGER at [4, 5] !!
Total no. of steps : 9

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 0     #  #
 1  #  S     #
 2  #  1
 3  #  2  3                                                 E
 4     #  4  #
 5  #     5  6  X
 6  #     #  7  8
 7  #     #  #
 8  #        #
 9     #  #

Thread 'POOH' is dead! It's sacrifice shall not be in vain!
Thread 'TIGGER' hits a dead end near [2, 8] !!
Total no. of steps : 9

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 0     #  #
 1  #  S     #
 2  #  1
 3  #  2  3                                                 E
 4     #  4  #
 5  #     5  6  X
 6  #     #  7  8
 7  #     #  #  #
 8  #        #
 9     #  #
```

A thread finds a danger point and is terminated

```
Thread 'ROO' has been created !!
============================================================
Elapsed Time: 0
Latest Update...
============================================================

Dead End Paths Found   : 2
Barriers Discovered    : 21
Danger Area Discovered : 1

Thread 'ROO' hits a dead end near [2, 8] !!
Thread 'ROO' hits a dead end near [5, 1] !!
Total no. of steps : 17

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 0     #  #     #  #
 1  #  S     # 15 16
 2  #  1     # 14
 3  #  2  3    13                                      E
 4     #  4  # 12 11
 5  #     5  6  X 10
 6  #     #  7  8  9
 7  #     #  #  #
 8  #        #
 9     #  #
```

Update is shown and more dead ends found

```
Finished Finding a SAFE PATH !!
Printing submitted maze solution ...

Printing solution for GarciaAnthonyJohnAbril, id : 4321819
-----------------------------------------------------------------------------------------------

Total no. of steps : 32

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 0     #  #     #  #     #  #  #                          #
 1  #  S  1  #        #                                    #
 2  #  3  2  #     #  #        #                #     #  #
 3  #  4  5  6  7  8  #                     # 30 31  E
 4     #     #     9        X              X    29
 5  #           X 10  #  #  #                 # 28 27
 6  #     #       11  #                    # 24 25 26  #
 7  #     #  #  # 12  #  #  #              # 23  #  #     #
 8  #        #    13 14 15 16 17 18 19 20 21 22              #
 9     #  #     #     #  #  #                          #

Total no. of Threads submitting info : 5

Duplicated Paths (to Barriers)    submitted  : 77
Duplicated Paths (to Danger Area) submitted  : 0
Total no. of Barrier     ('#') discovered    : 56 out of 105 !!
Total no. of Danger Area ('X') discovered    : 3 out of 3 !!

Printing Thread Statistics !!
-----------------------------------------------------------------------------------------------

Stats for Thread ID : 140635615037184

Found Solution Path                       : No ...

UNIQUE     Path to barriers discovered    : 20
DUPLICATED Path to barriers submitted     : 0

UNIQUE     Path to danger areas discovered : 1
DUPLICATED Path to danger areas submitted  : 0

*****************************************************************
```

A solution is found and program is terminated. Report is shown.

```
**********************************************************************

Stats for Thread ID : 140635589859072

Found Solution Path                         : No ...

UNIQUE     Path to barriers discovered      : 22
DUPLICATED Path to barriers submitted       : 0

UNIQUE     Path to danger areas discovered  : 1
DUPLICATED Path to danger areas submitted   : 0

**********************************************************************

Stats for Thread ID : 140635573073664

Found Solution Path                         : No ...

UNIQUE     Path to barriers discovered      : 11
DUPLICATED Path to barriers submitted       : 0

UNIQUE     Path to danger areas discovered  : 0
DUPLICATED Path to danger areas submitted   : 0

**********************************************************************

Stats for Thread ID : 140635564680960

Found Solution Path                         : YES !!

UNIQUE     Path to barriers discovered      : 36
DUPLICATED Path to barriers submitted       : 0

UNIQUE     Path to danger areas discovered  : 0
DUPLICATED Path to danger areas submitted   : 0

**********************************************************************
```

Thread information are shown

```
Discoveries
Dead End Paths Found   : 13
Barriers Discovered    : 65
Danger Area Discovered : 3
No Of Threads Started  : 5
No Of Paths Submitted  : 225
No Of Solns Submitted  : 1

Shortest Path Found    :
Total no. of steps : 32

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
 0  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #
 1  #  S  1  #        #                                       #
 2  #  3  2  #     #  #        #  #        #  #  #     #  #  #
 3  #  4  5  6  7  8  #           #           # 30 31  E     #
 4  #  #     #     9  #     X     #        X    29  #  #     #
 5  #           X 10  #  #  #     #  #  #  #  # 28 27        #
 6  #     #       11  #           #        # 24 25 26  #  #  #
 7  #     #  #  # 12  #  #  #     #  #     # 23  #  #  #  #  #
 8  #        #    13 14 15 16 17 18 19 20 21 22              #
 9  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #
```

Further discoveries are shown. Including the shortest path.

```
Thread 'GOLPHER0' just found a solution! Well done!!
Thread 'ROO0' just found a solution! Well done!!
============================================================
Elapsed Time: 10
Latest Update...
============================================================

Dead End Paths Found   : 37
Barriers Discovered    : 92
Danger Area Discovered : 3

Thread 'KANGA0' has been created !!
Thread 'KANGA0' hits a dead end near [18, 8] !!
============================================================
Elapsed Time: 11
Latest Update...
============================================================

Dead End Paths Found   : 38
Barriers Discovered    : 92
Danger Area Discovered : 3

Thread 'LUMPY0' has been created !!
Finished Finding a SAFE PATH !!
Printing submitted maze solution ...
```

Program is run again and is now having 10 solutions submitted

```
Discoveries
Dead End Paths Found   : 38
Barriers Discovered    : 92
Danger Area Discovered : 3
No Of Threads Started  : 15
No Of Paths Submitted  : 821
No Of Solns Submitted  : 10

Shortest Path Found    :
Total no. of steps : 30

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
  0 #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #
  1 #  S     #        #                                      #
  2 #  1     #     #  #        #  #        #  #  #     #  #  #
  3 #  2  3           #           #           # 28 29  E     #
  4 #  #  4  #        #     X     #        X    27  #  #     #
  5 #     5  6  X     #  #  #     #  #  #  #  # 26 25        #
  6 #     #  7  8  9  #           #        # 22 23 24  #  #  #
  7 #     #  #  # 10  #  #  #     #  #     # 21  #  #  #  #  #
  8 #        #    11 12 13 14 15 16 17 18 19 20              #
  9 #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #  #
```

Shortest path among the 10 solution paths submitted is shown

Summary of implementation of each module in your program

PathFinder.h

```
*******************************
int Initialize();
      Created to initialize all variables and load the maze
int Start();
      Start of threading and finding paths
void Output();
      Display and printing to file of final result
int Close();
      Deallocation of memory and joining of all threads
void *FindAPath(void *vptr_args);
      The path finder function where a thread finds a path
void *DisplayInfo(void * vptr_args);
      Displays updates on a consistent time and runs on the third thread
void getStats(bool printToFile);
      Display or prints to file of additional findings
void submitBarrier(Point currLoc, VectorOfPointStructType pathTaken);
      Submission of barrier point and path taken
void submitDanger(Point currLoc, VectorOfPointStructType pathTaken);
      Submission of danger point and path taken
void submitSoln(Point currLoc, VectorOfPointStructType pathTaken);
      Submission of solution path taken
VectorOfPointStructType getShortestPathRev();
      Get the shortest path submitted
```

Reflections on program development

- Assumptions made
  - “mazedata.txt” is the name of the file containing data
  - Duplicated barriers found are acceptable
  - First and second threads are for maze solving and third thread is for constant display of updates
- Difficulties faced
  - Understanding threading and how to use PThread library
- What could have been done better
  - Maybe a better/faster algorithm for finding a path
- Possible enhancements in future
  - Variable time in showing updates
- What have I learnt
  - The use of basic thread creation
  - Algorithm for finding a path