Welcome to NB theme!

Rulebook

Task 0

Task 1

Task 2

Task 3

Task 4

4A - Path Planning

4B - Develop Ball Navigation Algorithm

4C - Theme Analysis and Implementation

Task 5

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

eyantra Engineering a better tomorrow

eYRC 2020-21: Nirikshak Bot (NB)

# Task 4A

# Path Planning

[ *Last Updated on:* ***17th December 2020, 20:00 Hrs*** ]

- 1. Task 4A Folder Layout
- 2. Coordinate System in Grid and Maze
- 3. Given
- 4. Problem Statement
- 5. Instructions
- 6. Running your Solution
- 7. Testing your Solution
- 8. Submitting your Solution

The aim of this task is as follows:

- Read **start** and **end** coordinates for a given maze from a **JSON** file.
- Calculate the **path** between the start and end coordinates for a given maze.

## 1. Task 4A Folder Layout

- Download the following zip file containing the files for Task 1A. Right-click on the hyperlink an select **Save Link As...** option to download.
    - **Windows OS Users**:
        - **task_4a_path_planning_windows.zip**

Welcome to NB theme!

Rulebook ❱

Task 0 ❱

Task 1 ❱

Task 2 ❱

Task 3 ❱

Task 4 ❯

4A - Path Planning

4B - Develop Ball Navigation Algorithm

4C - Theme Analysis and Implementation

Task 5 ❱

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

- **Ubuntu OS Users**:
    - **task_4a_path_planning_ubuntu.zip**
- **Macintosh OS Users**:
    - **task_4a_path_planning_macintosh.zip**

- Following is the **file/folder layout** for **Task 4A**:
    - Inside **task_4a_path_planning** folder, you will find the following **files / folders**:
        - ***task_4a.py*** file - you will implement your solution in this file
        - ***start_end_coordinates.json*** file - it contains start and end coordinates for each maz image in ***test_cases*** folder
        - ***test_task_4a.exe*** OR ***test_task_4a*** file - you will test your solution using this executable file
        - ***test_cases*** folder
            - ***maze00.jpg*** to ***maze09.jpg*** files - maze images to test your solution

## 2. Coordinate System in Grid and Maze

We have defined a coordinate system for the cells of the **Grid** and **Maze**. The coordinate system is i a **row-major** format (as shown in Figure 1(a) and Figure 1(b).

Welcome to NB theme!

Rulebook

Task 0

Task 1

Task 2

Task 3

Task 4

4A - Path Planning

4B - Develop Ball Navigation Algorithm

4C - Theme Analysis and Implementation

Task 5

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

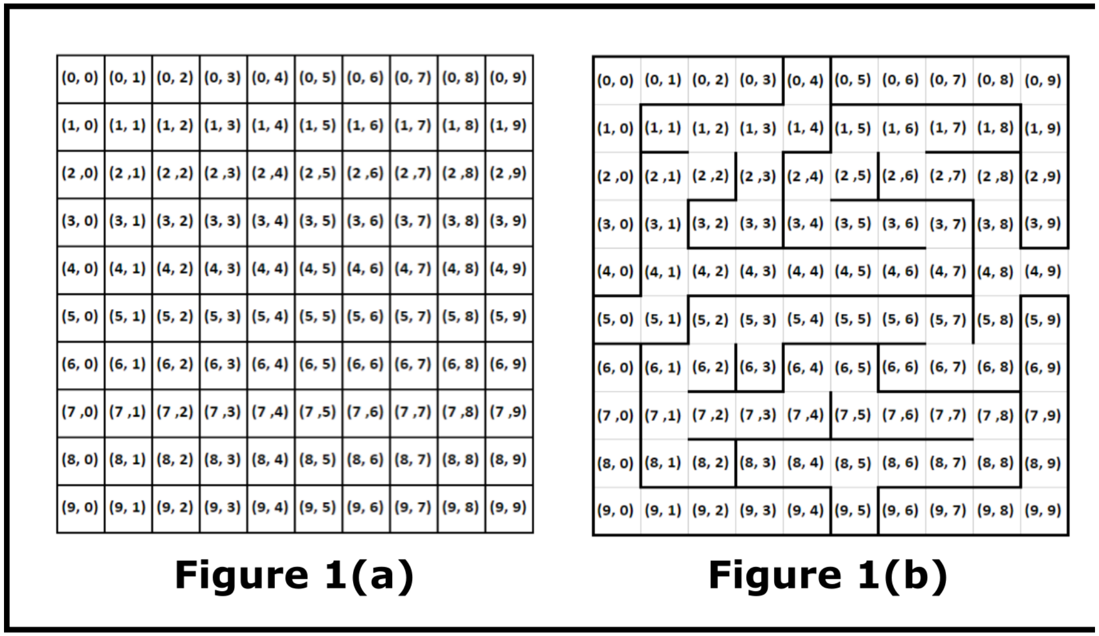
Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| (0, 0) | (0, 1) | (0, 2) | (0, 3) | (0, 4) | (0, 5) | (0, 6) | (0, 7) | (0, 8) | (0, 9) |
| (1, 0) | (1, 1) | (1, 2) | (1, 3) | (1, 4) | (1, 5) | (1, 6) | (1, 7) | (1, 8) | (1, 9) |
| (2 ,0) | (2 ,1) | (2 ,2) | (2 ,3) | (2 ,4) | (2 ,5) | (2 ,6) | (2 ,7) | (2 ,8) | (2 ,9) |
| (3, 0) | (3, 1) | (3, 2) | (3, 3) | (3, 4) | (3, 5) | (3, 6) | (3, 7) | (3, 8) | (3, 9) |
| (4, 0) | (4, 1) | (4, 2) | (4, 3) | (4, 4) | (4, 5) | (4, 6) | (4, 7) | (4, 8) | (4, 9) |
| (5, 0) | (5, 1) | (5, 2) | (5, 3) | (5, 4) | (5, 5) | (5, 6) | (5, 7) | (5, 8) | (5, 9) |
| (6, 0) | (6, 1) | (6, 2) | (6, 3) | (6, 4) | (6, 5) | (6, 6) | (6, 7) | (6, 8) | (6, 9) |
| (7 ,0) | (7 ,1) | (7 ,2) | (7 ,3) | (7 ,4) | (7 ,5) | (7 ,6) | (7 ,7) | (7 ,8) | (7 ,9) |
| (8, 0) | (8, 1) | (8, 2) | (8, 3) | (8, 4) | (8, 5) | (8, 6) | (8, 7) | (8, 8) | (8, 9) |
| (9, 0) | (9, 1) | (9, 2) | (9, 3) | (9, 4) | (9, 5) | (9, 6) | (9, 7) | (9, 8) | (9, 9) |

**Figure 1(a)**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| (0, 0) | (0, 1) | (0, 2) | (0, 3) | (0, 4) | (0, 5) | (0, 6) | (0, 7) | (0, 8) | (0, 9) |
| (1, 0) | (1, 1) | (1, 2) | (1, 3) | (1, 4) | (1, 5) | (1, 6) | (1, 7) | (1, 8) | (1, 9) |
| (2 ,0) | (2 ,1) | (2 ,2) | (2 ,3) | (2 ,4) | (2 ,5) | (2 ,6) | (2 ,7) | (2 ,8) | (2 ,9) |
| (3, 0) | (3, 1) | (3, 2) | (3, 3) | (3, 4) | (3, 5) | (3, 6) | (3, 7) | (3, 8) | (3, 9) |
| (4, 0) | (4, 1) | (4, 2) | (4, 3) | (4, 4) | (4, 5) | (4, 6) | (4, 7) | (4, 8) | (4, 9) |
| (5, 0) | (5, 1) | (5, 2) | (5, 3) | (5, 4) | (5, 5) | (5, 6) | (5, 7) | (5, 8) | (5, 9) |
| (6, 0) | (6, 1) | (6, 2) | (6, 3) | (6, 4) | (6, 5) | (6, 6) | (6, 7) | (6, 8) | (6, 9) |
| (7 ,0) | (7 ,1) | (7 ,2) | (7 ,3) | (7 ,4) | (7 ,5) | (7 ,6) | (7 ,7) | (7 ,8) | (7 ,9) |
| (8, 0) | (8, 1) | (8, 2) | (8, 3) | (8, 4) | (8, 5) | (8, 6) | (8, 7) | (8, 8) | (8, 9) |
| (9, 0) | (9, 1) | (9, 2) | (9, 3) | (9, 4) | (9, 5) | (9, 6) | (9, 7) | (9, 8) | (9, 9) |

**Figure 1(b)**

Figure 1: Coordinate System in Grid and Maze

## 3. Given

- A set of **10 maze images** are given in the **test_cases** folder. There are **two kinds** of maze images in the **test_cases** folder.
- **maze00.jpg** to **maze07.jpg** are mazes with dimensions **8x8** with **1 cell padding** on all sides (shown in Figure 2(a)).
- **maze08.jpg** and **maze09.jpg** are mazes with dimensions **10x10** (shown with Figure 2(b)).

Welcome to NB theme!
Rulebook
Task 0
Task 1
Task 2
Task 3
Task 4
4A - Path Planning
4B - Develop Ball Navigation Algorithm
4C - Theme Analysis and Implementation
Task 5
Practice Task
Instructions for Task 6
Task 6 Scene Details
Coding Standard
Git and GitHub
Live Session 1 - 24th October 2020
Live Session 2 - 21st November 2020
Live Session 3 - 12th December 2020
Live Session 4 - 10th January 2021
Changelog

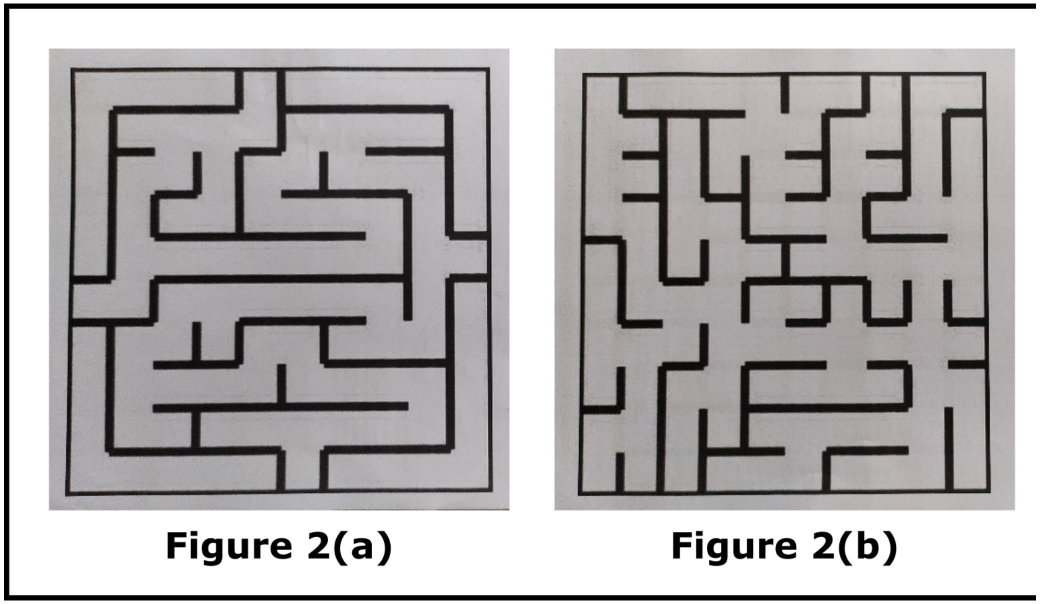


Figure 2: Example Maze Images

- A **JSON** file named **start_end_coordinates.json** is given. This **JSON** file contains a nested dictionary in the following format:

```
{"maze00": {"start_coord": [0, 4], "end_coord": [9, 5]},
 "maze01": {"start_coord": [5, 0], "end_coord": [4, 9]},
 ...
 ...
 "maze09": {"start_coord": [0, 0], "end_coord": [9, 9]}}
```

- The nested dictionary contains the **start** and **end** coordinates for each of the maze images (from ***maze00*** to ***maze09***).

## 4. Problem Statement

- Read the corresponding start and end coordinates from the JSON file for each of the mazes.
- Calculate the path between the start and end coordinates for a maze.

Welcome to NB theme!

Rulebook

Task 0

Task 1

Task 2

Task 3

Task 4

4A - Path Planning

4B - Develop Ball Navigation Algorithm

4C - Theme Analysis and Implementation

Task 5

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

- An example, for ***maze00*** (shown in Figure 2(a)), the start coordinates and end coordinates are given as **(0, 4)** and **(9, 5)**, so the path will be as:

```
path = [(0, 4), (1, 4), (1, 3), (1, 2), (2, 2), (2, 1), (3, 1),
        (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7),
        (3, 7), (3, 6), (3, 5), (3, 4), (2, 4), (2, 5), (1, 5),
        (1, 6), (2, 6), (2, 7), (2, 8), (3, 8), (4, 8), (5, 8),
        (6, 8), (6, 7), (5, 7), (5, 6), (5, 5), (5, 4), (5, 3),
        (5, 2), (6, 2), (6, 1), (7, 1), (7, 2), (7, 3), (7, 4),
        (6, 4), (6, 5), (7, 5), (7, 6), (7, 7), (7, 8), (8, 8),
        (8, 7), (8, 6), (8, 5), (9, 5)]
```

## 5. Instructions

- You will be using the functions ( `applyPerspectiveTransform()` and `detectMaze()` ) that you completed in **Task 1B**. This might be a good time to revisit `task_1b.py` again and recalibrate the code using the new batch of test cases we have provided you.

- You need to copy the `task_1b.py` file into the **task_4a_path_planning** folder.

- There are two functions pre-written in `task_4a.py` which you have to modify.

- The first function is `read_start_end_coordinates()` .

| **Function Name** | `read_start_end_coordinates()` |
|---|---|
| **Purpose** | Reads the corresponding start and end coordinates for each maze image from the specified JSON file |
| **Input Arguments** | `file_name` **: [ str ]**<br>name of JSON file<br>`maze_name` **: [ str ]**<br>specify the maze image for which the start and end coordinates are to be returned. |

Welcome to NB theme!
Rulebook ›
Task 0 ›
Task 1 ›
Task 2 ›
Task 3 ›
Task 4 ⌄
4A - Path Planning
4B - Develop Ball Navigation Algorithm
4C - Theme Analysis and Implementation
Task 5 ›
Practice Task
Instructions for Task 6
Task 6 Scene Details
Coding Standard
Git and GitHub
Live Session 1 - 24th October 2020
Live Session 2 - 21st November 2020
Live Session 3 - 12th December 2020
Live Session 4 - 10th January 2021
Changelog

| **Function Name** | `read_start_end_coordinates()` |
|---|---|
| **Output Arguments** | `start_coord` : [ **tuple** ]<br>start coordinates for the maze image<br>`end_coord` : [ **tuple** ]<br>end coordinates for the maze image<br>The coordinates should be returned in `tuple` format. For example for maze00, the start and end coordinates should be returned as **(0, 4)** and **(9, 5)**. |
| **Example Call** | `start, end = read_start_end_coordinates("start_end_coordinates.json", "maze00")` |

- The second function is `find_path()` .
- This function will be used for finding the path from start coordinates to end coordinates. You will have to explore Path planning algorithms in order to complete this part. There are many path planning algorithms like **BFS, DFS, Djikstra, A-star**. You will be able to find plenty of online resources for the same.

| **Function Name** | `find_path()` |
|---|---|
| **Purpose** | Takes a maze array as input and calculates the path between the start coordinates and end coordinates. |
| **Input Arguments** | `maze_array` : [ **nested list of lists** ]<br>encoded maze array in form of 2D array<br>`start_coord` : [ **tuple** ]<br>start coordinates of the path<br>`end_coord` : [ **tuple** ]<br>end coordinates of the path |
| **Output Arguments** | `path` : [ **list of tuples** ]<br>path between start and end coordinates<br>The path should be generated in form of a list of tuples. Refer the example given in the Problem Statement section<br>The path should be returned as **None** if no path exists between start and end coordinates. |

Welcome to NB theme!

Rulebook

Task 0

Task 1

Task 2

Task 3

Task 4

4A - Path Planning

4B - Develop Ball Navigation Algorithm

4C - Theme Analysis and Implementation

Task 5

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

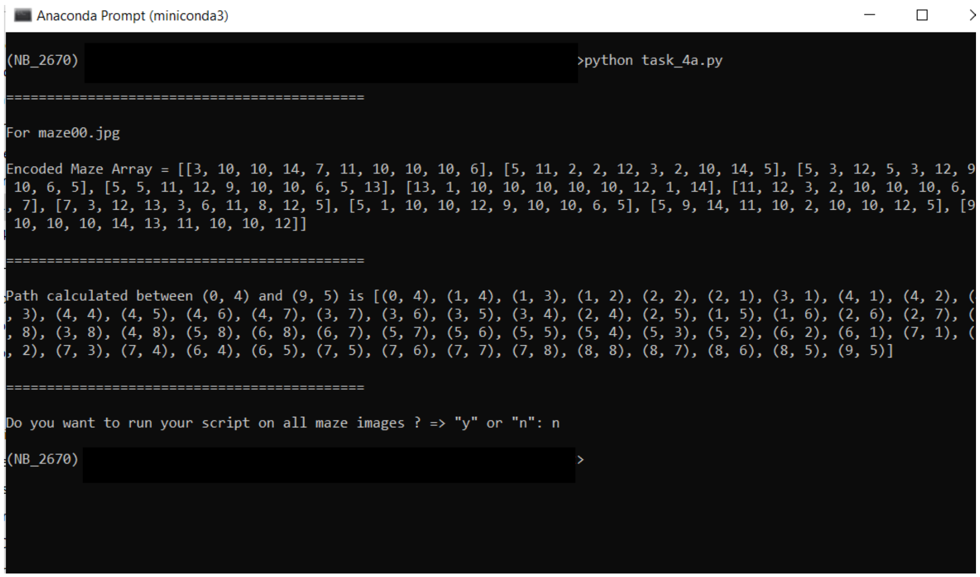
Live Session 4 - 10th January 2021

Changelog

| | |
|---|---|
| **Function Name** | `find_path()` |
| **Example Call** | `path = find_path(maze_array, start_coord, end_coord)` |

## 6. Running your Solution

- To test and run your solution, do the following:
  - Open Anaconda Prompt or Terminal and navigate to the directory / folder **task_4a_path_planning** on your system.
  - Activate the Conda environment.
  - Run the command: `python task_4a.py` to execute your solution. Your should get an output similar to Figure 3.



```
(NB_2670) [redacted] >python task_4a.py

============================================

For maze00.jpg

Encoded Maze Array = [[3, 10, 10, 14, 7, 11, 10, 10, 10, 6], [5, 11, 2, 2, 12, 3, 2, 10, 14, 5], [5, 3, 12, 5, 3, 12, 9
 10, 6, 5], [5, 5, 11, 12, 9, 10, 10, 6, 5, 13], [13, 1, 10, 10, 10, 10, 10, 12, 1, 14], [11, 12, 3, 2, 10, 10, 10, 6,
, 7], [7, 3, 12, 13, 3, 6, 11, 8, 12, 5], [5, 1, 10, 10, 12, 9, 10, 10, 6, 5], [5, 9, 14, 11, 10, 2, 10, 10, 12, 5], [9
 10, 10, 10, 14, 13, 11, 10, 10, 12]]

============================================

Path calculated between (0, 4) and (9, 5) is [(0, 4), (1, 4), (1, 3), (1, 2), (2, 2), (2, 1), (3, 1), (4, 1), (4, 2), (
, 3), (4, 4), (4, 5), (4, 6), (4, 7), (3, 7), (3, 6), (3, 5), (3, 4), (2, 4), (2, 5), (1, 5), (1, 6), (2, 6), (2, 7), (
, 8), (3, 8), (4, 8), (5, 8), (6, 8), (6, 7), (5, 7), (5, 6), (5, 5), (5, 4), (5, 3), (5, 2), (6, 2), (6, 1), (7, 1), (
, 2), (7, 3), (7, 4), (6, 4), (6, 5), (7, 5), (7, 6), (7, 7), (7, 8), (8, 8), (8, 7), (8, 6), (8, 5), (9, 5)]

============================================

Do you want to run your script on all maze images ? => "y" or "n": n

(NB_2670) [redacted] >
```

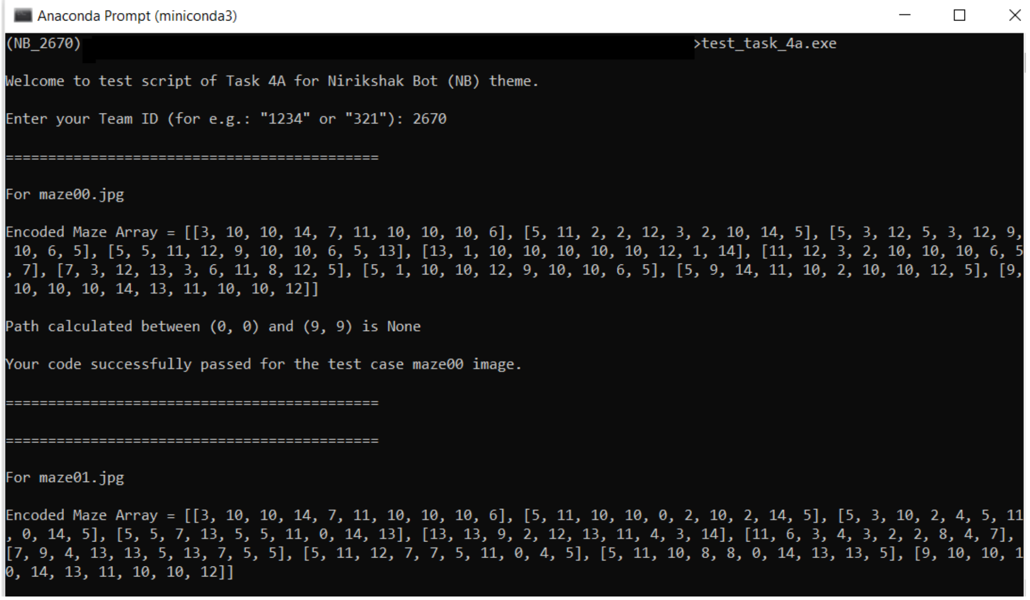
Figure 3: Running task_4a.py

  - You can choose to run your script for rest 9 maze images from **test_cases** folder by providing "**y**" as an input.

Welcome to NB theme!
Rulebook
Task 0
Task 1
Task 2
Task 3
Task 4
4A - Path Planning
4B - Develop Ball Navigation Algorithm
4C - Theme Analysis and Implementation
Task 5
Practice Task
Instructions for Task 6
Task 6 Scene Details
Coding Standard
Git and GitHub
Live Session 1 - 24th October 2020
Live Session 2 - 21st November 2020
Live Session 3 - 12th December 2020
Live Session 4 - 10th January 2021
Changelog

## 7. Testing your Solution

- The final step before submitting **Task 4A** is to test and evaluate your solution.
- For testing your solution, run the `test_task_4a.exe` OR `test_task_4a` in Anaconda Prompt / Terminal with the command: `test_task_4a.exe` (on **Windows**) and `./test_task_4a` (on **Ubuntu** and **Mac**).
- If your script executed without any errors, you will see the output resemble Figure 4.



```
(NB_2670) [redacted] >test_task_4a.exe

Welcome to test script of Task 4A for Nirikshak Bot (NB) theme.

Enter your Team ID (for e.g.: "1234" or "321"): 2670

============================================

For maze00.jpg

Encoded Maze Array = [[3, 10, 10, 14, 7, 11, 10, 10, 10, 6], [5, 11, 2, 2, 12, 3, 2, 10, 14, 5], [5, 3, 12, 5, 3, 12, 9,
 10, 6, 5], [5, 5, 11, 12, 9, 10, 10, 6, 5, 13], [13, 1, 10, 10, 10, 10, 10, 12, 1, 14], [11, 12, 3, 2, 10, 10, 10, 6, 5
, 7], [7, 3, 12, 13, 3, 6, 11, 8, 12, 5], [5, 1, 10, 10, 12, 9, 10, 10, 6, 5], [5, 9, 14, 11, 10, 2, 10, 10, 12, 5], [9,
 10, 10, 10, 14, 13, 11, 10, 10, 12]]

Path calculated between (0, 0) and (9, 9) is None

Your code successfully passed for the test case maze00 image.

============================================

============================================

For maze01.jpg

Encoded Maze Array = [[3, 10, 10, 14, 7, 11, 10, 10, 10, 6], [5, 11, 10, 10, 0, 2, 10, 2, 14, 5], [5, 3, 10, 2, 4, 5, 11
, 0, 14, 5], [5, 5, 7, 13, 5, 5, 11, 0, 14, 13], [13, 13, 9, 2, 12, 13, 11, 4, 3, 14], [11, 6, 3, 4, 3, 2, 2, 8, 4, 7],
[7, 9, 4, 13, 13, 5, 13, 7, 5, 5], [5, 11, 12, 7, 7, 5, 11, 0, 4, 5], [5, 11, 10, 8, 8, 0, 14, 13, 13, 5], [9, 10, 10, 1
0, 14, 13, 11, 10, 10, 12]]
```

Figure 4: Running test_task_4a.exe

- If your code executed without errors, you will also see a new file **task_4a_output.csv** generated in the **task_4a_path_planning** folder. This output file is encoded.

## 8. Submitting your Solution

For **Task_4A submission** you have to upload a **.zip** file. To create the appropriate file please follow instructions given below:

- Create a new folder named `NB_<Team-ID>_Task_4A`.

Welcome to NB theme!

Rulebook

Task 0

Task 1

Task 2

Task 3

Task 4

4A - Path Planning

4B - Develop Ball Navigation Algorithm

4C - Theme Analysis and Implementation

Task 5

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

Live Session 1 - 24th October 2020

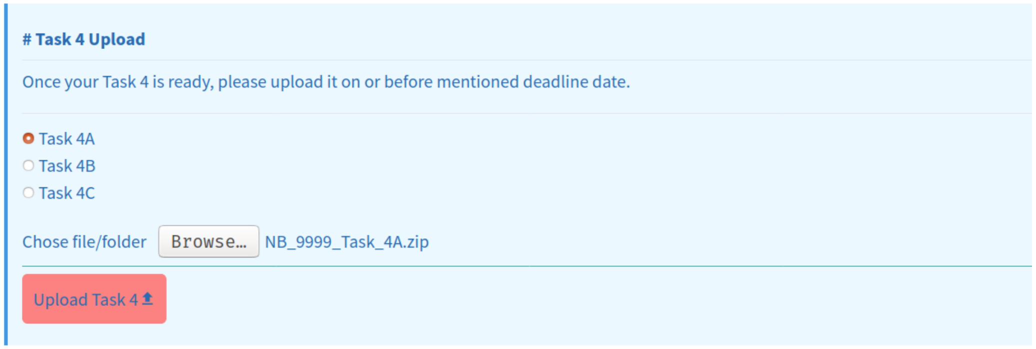
Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog

- For example: if your team ID is **9999** then you need to create a folder named `NB_9999_Task_4A` .
- Now copy and paste following files into this folder:
  - `task_1b.py`
  - `task_4a.py`
  - `task_4a_output.csv` (generated after running `test_task_4a.exe` OR `test_task_4a` )
- Compress this folder into a `NB_9999_Task_4A.zip` file.
- Now go to the eYRC Portal and follow the instructions to upload this **.zip** file for **Task_4A** (shown in Figure 5).



**# Task 4 Upload**

Once your Task 4 is ready, please upload it on or before mentioned deadline date.

● Task 4A
○ Task 4B
○ Task 4C

Chose file/folder Browse... NB_9999_Task_4A.zip

Figure 5: Submission of **NB_9999_Task_4A.zip** file on eYRC portal

---

**NOTE:** File names mentioned are case sensitive. Verify all the file names before creating the zip file.

---

- Congrats, you have successfully completed **Task 4A** ! You can now proceed to **Task 4B**.
- After you have **successfully submitted Task 4A , 4B and 4C files**, you can verify the zip file uploaded from the **'Verify Task 4 Upload'** section on the eYRC portal. The same is shown is Figure 6.

Welcome to NB theme!

Rulebook ›

Task 0 ›

Task 1 ›

Task 2 ›

Task 3 ›

Task 4 ⌄

4A - Path Planning

4B - Develop Ball Navigation Algorithm

4C - Theme Analysis and Implementation

Task 5 ›

Practice Task

Instructions for Task 6

Task 6 Scene Details

Coding Standard

Git and GitHub

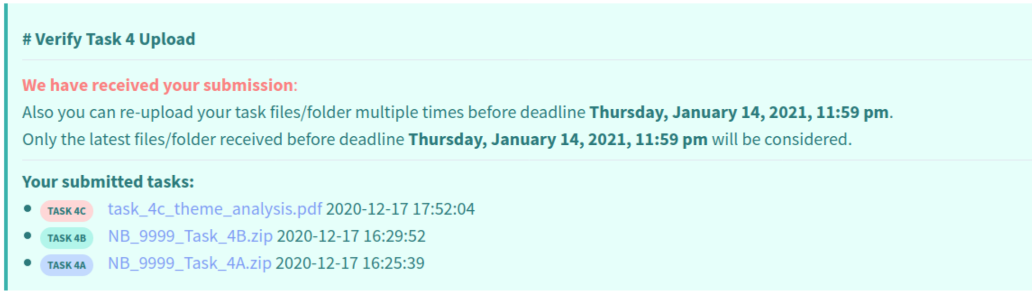
Live Session 1 - 24th October 2020

Live Session 2 - 21st November 2020

Live Session 3 - 12th December 2020

Live Session 4 - 10th January 2021

Changelog



**# Verify Task 4 Upload**

**We have received your submission:**

Also you can re-upload your task files/folder multiple times before deadline **Thursday, January 14, 2021, 11:59 pm**.

Only the latest files/folder received before deadline **Thursday, January 14, 2021, 11:59 pm** will be considered.

**Your submitted tasks:**

- TASK 4C task_4c_theme_analysis.pdf 2020-12-17 17:52:04
- TASK 4B NB_9999_Task_4B.zip 2020-12-17 16:29:52
- TASK 4A NB_9999_Task_4A.zip 2020-12-17 16:25:39

Figure 6: Successful submission of Task 4A, 4B and 4C on eYRC portal

## ALL THE BEST !!