SCHOOL OF COMPUTING (SOC)

Diploma in Applied AI and Analytics

ST1507 DATA STRUCTURES AND ALGORITHMS (AI)

2022/23 SEMESTER 2
ASSIGNMENT TWO (CA2)

~ Pizza Runners (path finding algorithms) ~

Objective of Assignment

For this assignment you will have an opportunity to apply all that you have learnt with regards to data structures, algorithms, and object-oriented programming as to develop an application that can help a pizza delivery company to use drones to navigate a city landscape by means of suitable path finding algorithms.

Overview of the system

Pizza Runners is a startup company that specializes in door-step pizza delivery for the busy office workers in the business district. Since last year they have started to use drones to deliver pizzas at the office workers doorsteps. They are currently using a computerized system that controls the navigation of the drones using a path finding approach that is based on the *left-hand* algorithm (also known as *wall follower* algorithm).

Lately they have received complaints from customers with regards to late deliveries, and occasionally, even completely failed deliveries. A preliminary investigation by the company's IT department, has concluded that the left-hand algorithm that the company is currently using is at fault. They discovered that the left-hand algorithm has some serious shortcomings. For instance, the algorithm appears not always to be able to find a path between two points (even if there is one). This has caused the drone to get lost or trapped on several occasions. They also noticed, that even when the drone reaches its destination, it may not necessary have taken the shortest path, and this has caused the delays in delivery that people have been complaining about.

Your job as team

As a team of AI programmers, you have been roped in to implement a more efficient, and effective path finding solution for Pizza Runners. Your task is to develop a Python application that would allow the company to simulate the navigation of the pizza delivery drones in the business district. You are required to implement the left-hand algorithm, as well as a second algorithm (one of you own choice) that would meet the requirements of the client, namely, an algorithm that is guaranteed to find a path (if there is one), and to come up with the shortest path possible.

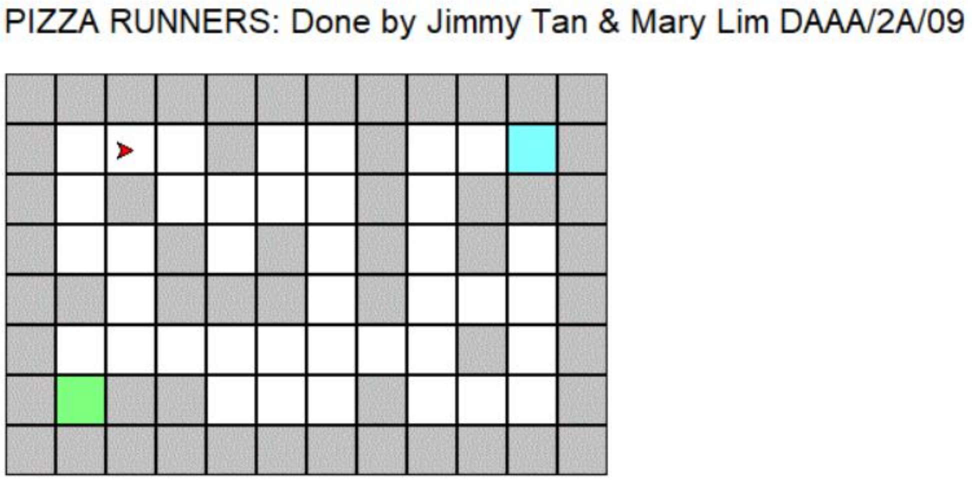
<u>City map</u>

You will need to read the city map from a text file. You must follow the following conventions.

Character	**Description**
X	*Building (drone can't go here)*
.	*Road (drone can fly here)*
s	*The start location of the drone*
e	*The pizza delivery location*

Next is an example of a city map file, together with a screen shot of how the city map would appear in the application.

```
XXXXXXXXXXXX
X...X..X..eX
X.X....X.XXX
X..X.X.X.X.X
XX.XXX.X...X
X........X.X
XsXX...X...X
XXXXXXXXXXXX
```



PIZZA RUNNERS: Done by Jimmy Tan & Mary Lim DAAA/2A/09

Instructions and Guidelines:

1. This is a group assignment (you will work in pairs, only one group of 3 would be allowed if there is an odd number of total students).

2. This assignment accounts for **40%** of your final grade.

3. The assignment comprises a group component (70%) and an individual component (30%).

4. The submission date is **Wednesday 8 February 1:00 pm**.

5. The development will be carried out in Python using Anaconda.

6. The demonstrations/interviews will be conducted during the DSAA lessons in week 17/18. You are expected to explain on your code and program logic. Take note that the interview is compulsory. In case you are absent from the interview without valid reasons, you will not get any marks for the assignment.

7. No marks will be given, if the work is copied, or you have allowed others to copy your work.

8. **50% of marks** will be deducted for submission of assignment within **ONE** calendar day after the deadline. **No marks shall** be awarded for assignments submitted **more than one day** after the deadline.

Warning: Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person.
Plagiarism is a serious offence, and if you are found to have committed, aided, and/or abetted the offence of plagiarism, disciplinary action will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester, or even be liable for expulsion.

Requirements for Group Component (70%):

- Your program should support at least two path finding algorithms:

    1) You must implement the '*left-hand rule*' algorithm. A left-hand algorithm is how a blindfolded person would use his/her left hand to follow the wall in a building to find the exit. Hence this algorithm is also referred to as '*wall follower*' algorithm (pls. do some further research on the internet).

    2) The second can be any algorithm of your own choice. You should stive for an algorithm that is guaranteed to find a path (if there is one), and to come up with the shortest path possible.

- You are required to design and write the Python application using an object-oriented approach (OOP). You should use appropriate functions, classes, and data structures.

- The graphics, and interactivity capabilities of the application must be implemented through the Turtle library.

- You should only make use of those libraries that already ship with Anaconda Jupyter Take note Anaconda already ships with Turtle, you are also allowed to make use of the networkx library. You should not make use of the collection library.

- It is advisable to test your application with various city maps. Take note, we will test your application with some pre-prepared city maps.

- You must display your name and class above the city map in following format:

    PIZZA RUNNERS: Done by Your Names, and Class

- You must display the algorithm name, and the number of steps the application took/has taken, in the window's title bar.

- It should be possible for the user to switch between algorithms with a key press. You should use the tab key for this.

- You application must at least support city maps with dimensions, of 8 rows by 12 columns (as shown in screenshot example)

- The OOP classes that you develop must be placed in separate python files.

- The group will be requested to demonstrate the basic features of the application during the demonstration.

- Take note the group's demonstration should not exceed 15 minutes (including 5 minutes for Q&A).

Requirements for Individual Component (30%):

Each individual team member is required to implement two additional features to be added to the application. These two additional features will need to be presented during the final presentation.

- The features will be graded on technical sophistication, uniqueness, innovativeness, and usability.
- Take note features within the same group must be different. So please check with you group member(s) first before embarking on implementing the extra features.
- Each group member must present his/her features through a short PowerPoint presentation followed by a demonstration of the features (total presentation time not to exceed 10 minutes)
- Your PowerPoint slides must briefly describe what features you have implemented. You must include screen shots demonstrating the features in action. Please explain how the features work, and why they are useful. You are to include your Name, Class and Group Number in the first slide.
- You must submit the PowerPoint Slides (converted to pdf) together with a compulsory Peer Feedback (template will be provide on BlackBoard).

Final Deliverables

Your group's final deliverables must include:

(a) Group Report

A report (as pdf file) with a maximum **of 10** pages. This excludes cover page and the appendix with source listing and references. The report should contain:

a) Cover page with group number (your instructor will assign group numbers) names, ids, and class.
b) Description, and user guidelines, on how to operate your application (please include screen shots of your application in action).
c) Describe how you have made use of the Object-Oriented Programming (OOP) approach. You may elaborate of the classes that you have developed, and discus on issues such as encapsulation, function/operator overloading, polymorphism, inheritance etc. Include a class diagram that displays the relation between the various classes you have developed.
d) Discussion on the data structures and algorithms you have developed for your application. You may discuss on issues such as, the performance of the algorithms in terms of Big (O). Explain why you did develop certain data structures and explain why you deem these data structures suitable for the task(s) at hand. Include a table summarizing all the data structures that you have been using (those that you have developed and those already build in Python).
e) Include a summary of the challenges that the group has faced while developing the application (that should include both technical, as well as group-work challenges). Provide a summary of the key take aways and learning achievements that you have obtained from this project.
f) Include a clear description of the roles and contributions of each member_in the team. Clearly state what each member has been responsible for, and what programming work has been carried out by each member.
g) **All** your python **source code listings** must be included as an appendix at the end of your report. You must clearly indicate in the source code listings who wrote what code. You may include in the appendix, those references from literature or internet that you may have consulted.

(b) Source Code

- You must submit all (*) the python files (.py files) that makes up your application. Ensure code is complete, and that it can run directly from the Anaconda Prompt.

  (*) Take note, that includes the code for all the extra features as well.

Submission instructions

Group Submission:

- Group Leader to submit all the group deliverables (Source Code and Group Report) in the designated Blackboard Drop Box.

- You must submit it as one Zipped folder (RAR will not be accepted, only zip) whereby you label your submission as:

  CA2_GroupNumberClass.zip

  For example: *CA2_GR_10_DAAA_2B08.zip*

- Please ensure that you submit it by the stipulated deadline.

Individual Submission:

- Each individual Group Member is to submit his/her individual deliverables.

- Individual submission to include the PowerPoint slides (converted to pdf) and the Peer Feedback. Please ensure that you submit it in the designated Blackboard Drop Box for individual submissions.

- You must submit it as one Zipped folder (RAR will not be accepted, only zip) whereby you label your submission as:

  CA2_Final_ GroupNumberNameClass.zip

  For example: *CA2_GR_10_JIMMY_TAN_DAAA_2B08.zip*

- Please ensure to submit it by the stipulated deadline.

Assessment Criteria

The group component of the assignment will be assessed based on the following criteria:

Assessment criteria (Group 70 %)	**Marks awarded**
GROUP COMPONENT (70 %)	
File IO & GUI: - Reads city map from file and displays it correctly. - GUI allows switching of algorithms, updates algorithm name, and number of steps.	Max 10
Left-Hand rule algorithm: - Correctness - Efficiency	Max 10
Pathfinding algorithm Two (an algorithm of your own choice): - Guaranteed to find a path (if there is one) - Follows shortest path possible	Max 10
Programming techniques and efficiency: - Appropriate usage of classes and OOP technology. - Appropriate usage of data structures and algorithms.	Max 10
Robustness, and readability of code: - Code is properly commented and neatly structured. - Application is free of crashes.	Max 10
Group Report: - Report follows the prescribed format. - Report is well written and comprehensive.	Max 10
Group's demonstration: - Group effectively demonstrates the basic features. - Group's ability to answer questions raised in Q&A.	Max 10
Group Total	**70**

The individual component of the assignment will be assessed based on the following criteria:

Assessment criteria (Individual 30 %)	**Marks awarded**
INDIVIDUAL COMPONENT (30 %)	
Extra Feature One - Technical sophistication. - Uniqueness/innovativeness/usability.	Max 10
Extra Feature Two - Technical sophistication. - Uniqueness/innovativeness/usability.	Max 10
Presentation & Demonstration: - PowerPoint slides. - Demonstration of features and Q&A.	Max 10
Individual Total	**30**

(*) Take note in case of group members with poor contribution to the group effort a multiplier may be applied (peer feedback taken in consideration).

~ End ~