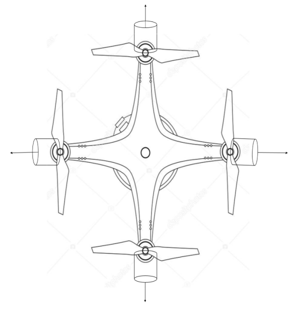
A.I.

First Assignment - Exploration in a simple environment

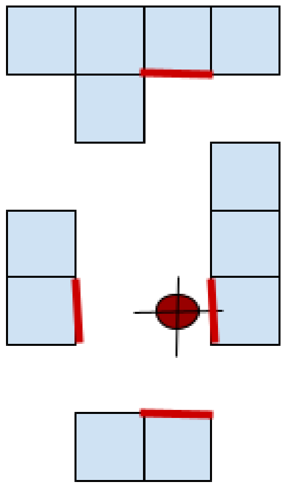
We have a drone, placed in an unknown environment. We want this drone to map this environment into a predefined area.

In order to simplify the problem we consider only the movement in plan.

First step is to model this environment. The drone will explore a certain rectangular area *A*. We divide this area in small squares of size length *u* that will fit loosely the drone (lets say *u = 55* cm). If the square is empty it will be marked zero, if it is occupied it is marked with 1.



The problem modeled like this is reduced to map a simple labyrinth. For this we use 4 of the Ultrasonic Distance Measuring Sensors (we use all 6 of them if we map a 3D space instead of a plan area). The 4 sensors will give us the distances to the obstacles from a point in 4 directions.



So the drone's sensors will detect the 3u, 0, u, and u values. So in order to map the surroundings we will need to move the drone, more or less, on each empty square at least one time.

Consider now an algorithm like DFS, or other of the same type, and we want to map the surroundings with it.

Task

Using a DFS type of algorithm, design a function that can be used by a drone to map a rectangular area of size *n x n* units *u* starting from an empty square. The area is unknown to the drone, the only information will come from the sensors (we will simulate that on the application, using a function *readUDMSensors*), so at every step the drone will receive 4 values - the number of empty squares to the next occupied one. From each position the drone can move in one adjacent empty square.

In the end we can compare the constructed map with the "real" environment.

This "real" environment will be encoded in a matrix in the class Environment. The **only way** the drone receives information from this matrix is through the *readUDMSensors.*

Your detected map is an object from class DMap. in a position on this map for now you can have 3 values:

1. -1   unknown value, unexplored
2. 0 an empty square
3. 1 a wall

You have to write the *moveDSF* function from the Drone class. The function will be called in the main function in the main loop of the application.

For that you have to erase the handler for the keypress event, and add the call for the moveDFS function and the time delay.

ATTENTION!! In one call you have to move the drone one square. Overall the moves should respect the DSF pattern of visit.

When there are no more empty squares to visit (the search is at the end, the stack is empty) the position of the drone will be set to none (*x = None, y = None*).

The application is very simple in beta version, it can have bugs, and errors. Feel free to correct them. The code is not optimum at all!

For this assignment one can get a maximum **100** points.

Extra **50** points can be earned if the application is made more friendly, and with a proper layered architecture.

Due time:

1 week to present an almost finished version of the code

2 weeks for the final solution

IF nothing is done in the first week you will have a penalty of 10 points.

The solution can not be turned in after 2 weeks.