COMP9032 Project

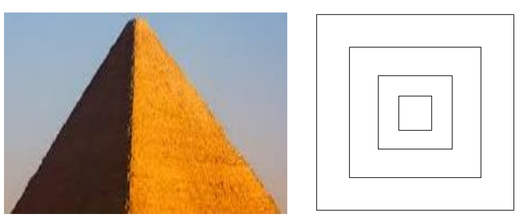
<u>Description</u>

Drones are increasingly used in missions that are difficult and dangerous for human beings.

Figure 1*

In this project, you will be working in a group to develop a simulation system with the lab board, emulating the control of a drone to search an accident scene in a mountainous area, as illustrated in Figure 1.

For simplicity, we assume the mountain has no caves and can be described by a 2D matrix of surface points, H. For each x-y coordinate location, the related surface height, H(x,y), is given in the matrix. A simple example of such representation is shown in Figure 2, where a pyramid on a 7meter-by-7meter area with the peak of 6 meters is represented by a 7x7 matrix.



$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 4 & 4 & 4 & 2 & 0 \\ 0 & 2 & 4 & 6 & 4 & 2 & 0 \\ 0 & 2 & 4 & 4 & 4 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(a) (b) (c)

Figure 2: (a) Pyramid* (b) Top View (c) H matrix

For this project, you can generate your own mountain that should have at least three peaks of different heights. For example, you can use pyramid-like structures and/or other simple structures of different sizes as building blocks to construct the mountain. The accident scene, if any, can be located on some surface point, associated with an element in the H matrix. It is also assumed that x and y are each 15 meters and that the highest peak and the lowest valley of the area are 9 meters and 2 meters, respectively (the height on the edge can be 0). All values in the 2D presentation are measured in meters and rounded to the nearest integers.

The drone is initially grounded. Once launched, it flies at a given speed to a default start search point, for example, H(0,0). During the search, the user can change its speed and flight direction. For simplicity, we assume only six directions the user can choose: east, west, north, south, up and down, and search can only go along east-west, south-north paths. We assume that the drone can automatically adjust its altitude within +/- 1 meter range when flying from one spot to its adjacent spot (given in H matrix). Namely, along a path, the drone can fly up and down freely according to the mountain's contour if the altitude change does not exceed 1 meter related to the current position.

We also assume that the drone can be in four states: 1) fly along a path, 2) hover to change flight direction, 3) return from the search, or 4) crash.
(Note: in the hover state, the drone still runs at a speed, we use the state to change the drone's altitude for more than 1 meter distance.)

The drone's' position is represented by (x, y, z). At a location, the local peak around the drone can block its view to other side of the peak. Therefore, the drone needs to fly close enough over the peaks, valleys and other mountain surfaces in order to see target while keeping safety (i.e. avoiding collision). We define visibility in terms of the distance of drone to a mountain surface spot. The drone can identify an accident scene only if it passes the scene within visibility distance during search.

The accident location can be set after the simulation starts and the visibility and initial search speed can be set before the simulation.

The design offers the following input/output functions:

Input:

- Reset button:
  - To start simulation
- Push buttons (PB0 and PB1):
  - To increase/decrease flight speed.
    - One button for increasing speed.
    - Another button for decreasing speed.
  - A button press changes the speed by 1m/s.
    - For example, if the current speed is 1m/s, changing the drone's altitude from 5 to 10 meters will take 5 seconds; by pressing the increasing button 4 times, the speed is changed to 5m/s and the drone takes 1 second to climb to the 10 meters' level.
- Keypad:
  - Six keys for six flight directions (south, north, east, west, up, down).
  - One key for state change between flight and hover.
    - Press the key to change from flight to hover; press the key again to resume flight.

Output:

- LCD (two-line display):
  - The first line displays the mountain contour on the current flight path which is represented by the related row or column in H matrix (for example, a row [0 2 4 4 4 2 0] in Figure 2(c) can serve as a path and displayed in the first line)
  - The second line is divided into three sections to display the following search information:
    - Flight state:
      - Flight, F
      - Hover, H
      - Return from search, R
      - Crash, C
    - Drone's position:
      - (x,y,z)
    - Flight speed/direction:
      - e.g. 1/W for speed 1m/s and west flight direction
- LED bar
  - On when simulation starts
  - Flash when crash

Some examples of LCD display are given in Figure 3 (a)-(c) and elaborated below.

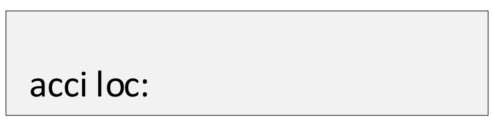


Figure 3(a)

Figure 3(a) shows the display when the simulation starts and the system is waiting for the input from the user for the accident location.

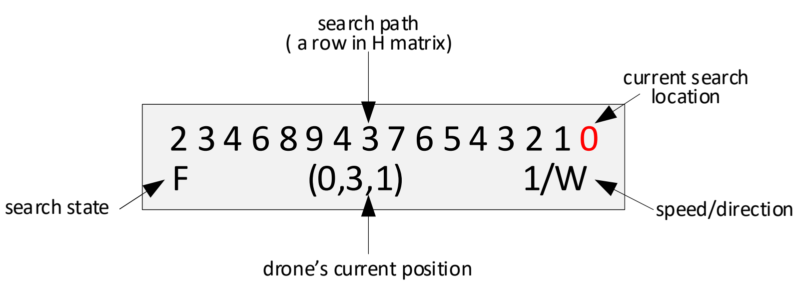


Figure 3(b)

Figure 3(b) shows that the drone is in a flight state at the location (0,3,1), flying towards west at the speed of 1m/s. Assume the right bottom location in H matrix is (0,0), the flight path in this example is related to row 3 in H matrix and the drone is flying 1 meter over the location corresponds to the first element of the row, as highlighted in red in the figure.

When the drone flies along the path, its position and speed can be changed by the user, as demonstrated in Figure 3(c).

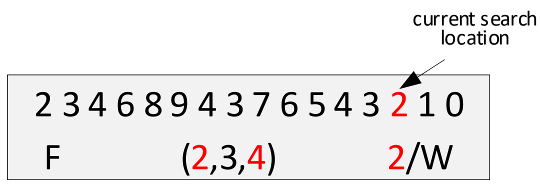


Figure 3(c)

Figure 3(d) demonstrates a display when the search completed, which shows there is an accident and the location is (12, 3, 6). You can use (-,-,-) to indicate no accident found.

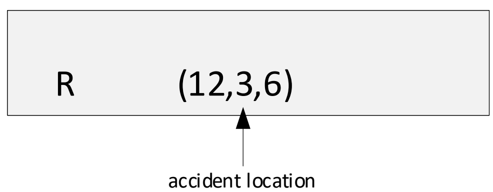


Figure 3(d)

In your design, you need to decide the overall search route, for example, from east to west, from south to north.

To speed up the simulation on the lab board, you can scale down time. For example, 1 second duration can be scaled down to 0.5 seconds.

For any information you think is necessary but not provided here, you are allowed to make reasonable assumptions.

<u>Submission and Assessment</u>

The project submission is divided into two parts: 1) lab demonstration and 2) written report, which are elaborated below.

1. Group lab demonstration (65 marks), due **your lab session in Week 10**

   The demonstration is run in the following way:
   - One member demonstrates the overall design with the lab board.
   - Other members each explain part of the code development.

Your marks from the **demonstration** consist of two components:

- 70% from the group work
  - Based on whether your design implements all functions set in the project specification.
- 30% from the individual presentation
  - Based on your understanding about the project and the design produced by your group.

2. Group project report *give* submission (35 marks), due **Fri., Nov. 17, 6pm**

   The submission consists of two files:

   - Source code (5 marks)
     The code should be well commented and easy to read.

   - Project report (30 marks).

     The report is about six pages long in font size 11. It should provide:

     - (5 marks) the general description about the project development, management, and the contribution of each group member, (5 marks)
     - (20 marks) the overview of the project design, which includes:
       - hardware components used and related interfacing design (5 marks)
       - software code structure and execution flow (5 marks), and
       - how software and hardware interact with each other (10 marks).
     - (5 marks) concluding remarks about the project.

   For a good report, you need to consider appropriate structure, language and supporting graphical illustrations and tables.

   The *give* command for the submission will be available in Week 10, one report per group.

Each member needs to submit a form about the overall contribution of all members (including themselves) in the group. On the submission form, make sure the total contribution of all members adds up to 100%.

Your individual mark of the assignment is based on two components: the whole group's mark (G, that comes from the project demonstration and report) and your relative contribution (C). It is calculated as G*C but capped by 100, as demonstrated in the following two examples:

1. For a group of four members, if everyone contributes equally and the group mark is 90, each member's mark is min{90*(100%/4)/(100%/4), 100}=90.
2. For a group of four members with the group mark of 70, if individual contributions are 20%, 20%, 20%, and 40%, respectively, the one with the highest contribution will receive a mark of min{70*40%/(100%/4), 100}=100.

The *give* submission and the contribution form will be available in Week 10.