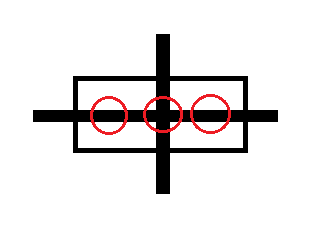
SPUG –

SPUG is the cart that autonomously moves to the product location that the user desires to buy. It calculates the shortest path to all the desired products by avoiding the other SPUG present. SPUG consists of the following components:

1. Infrared sensors: It has 3 IR sensors in the front which are parallelly placed to each other to sense the track. The sensor cluster is perpendicular to the track as shown in figure 1.

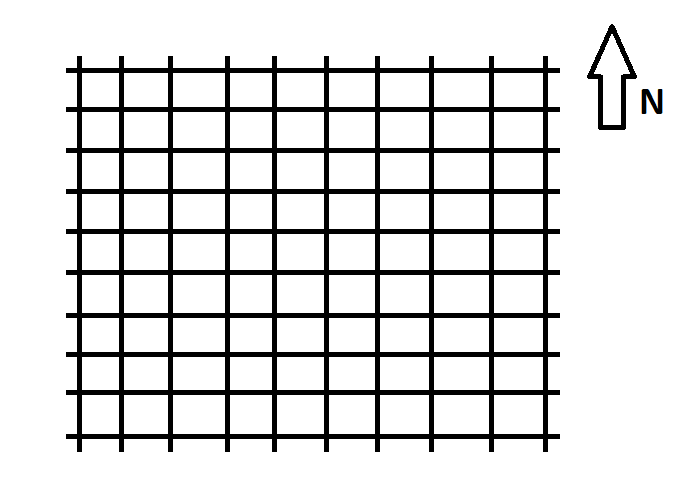


1. Ultrasonic sensor: It has a ultrasonic sensor in the front to check if there are any obstacles present on the path.
2. DC Motors: It has four independently operated DC motors, which allows the cart to move in all the four directions.
3. LED’s: The LED’s glow orange when there is an obstacle present within a threshold distance in the path of the SPUG.
4. Raspberry PI: It is the main controller which collects the data from sensors and controls the actuators. It communicates with the server to receive the product locations and blocking points and sends data to server regarding the current position in the map and if the product location is reached. It runs the AI planner to determine the shortest path to the products and back to the origin.

Track –

The track for movement of SPUG will be similar as shown in figure 2. The following assumptions are made:

1. The products are placed at the nodes.
2. Directions are considered to be absolute.
3. SPUG starts at Origin and if facing North as shown in figure 3.
4. At all the nodes SPUG’s three IR Sensors detect the track, even at the boundaries of the track.
5. Any other obstacles such as humans can be present on track.



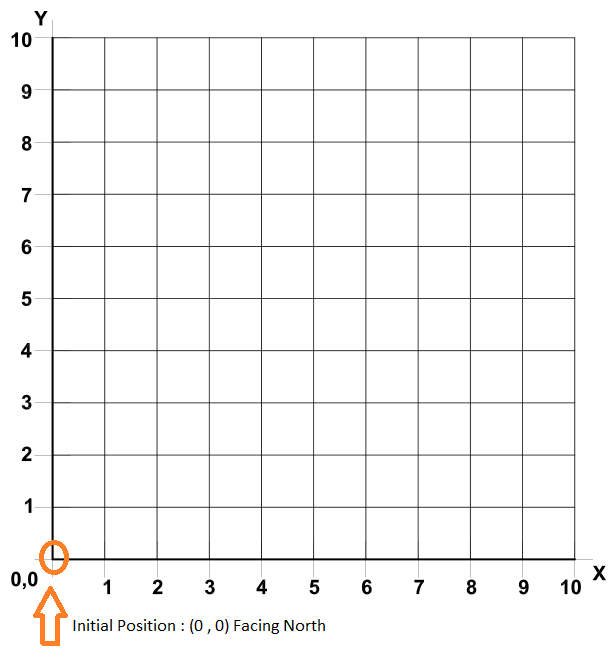
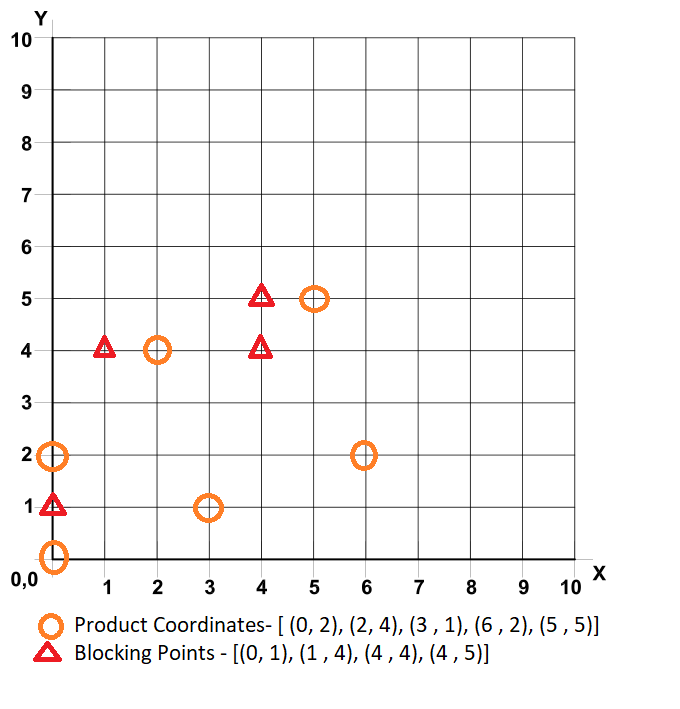
Movement –

SPUG receives the coordinates of the Products that the user has desired to buy. The product location coordinates are sent as JSON messages through MQTT Communication established between the Server and the specific SPUG. Once product locations are received it waits for the Blocking points. Blocking points are the locations which are occupied by the other SPUGs’. Blocking points are considered to be static in our scenario. The messages are sent in the same way as before.

For example, the product location points and the blocking points updated on a cartesian coordinates map is shown in figure 4 for the following example:

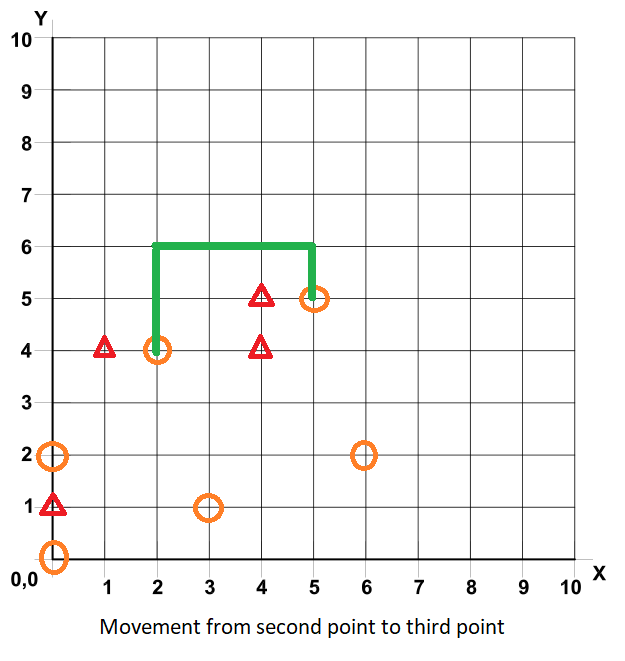
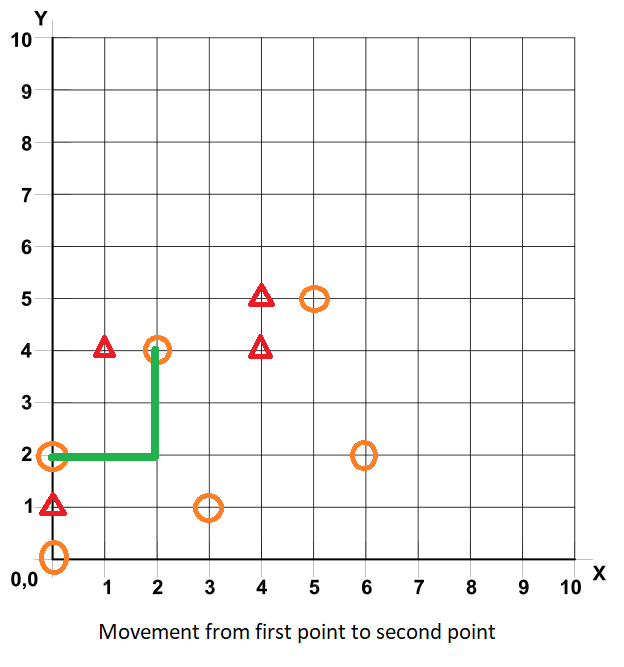
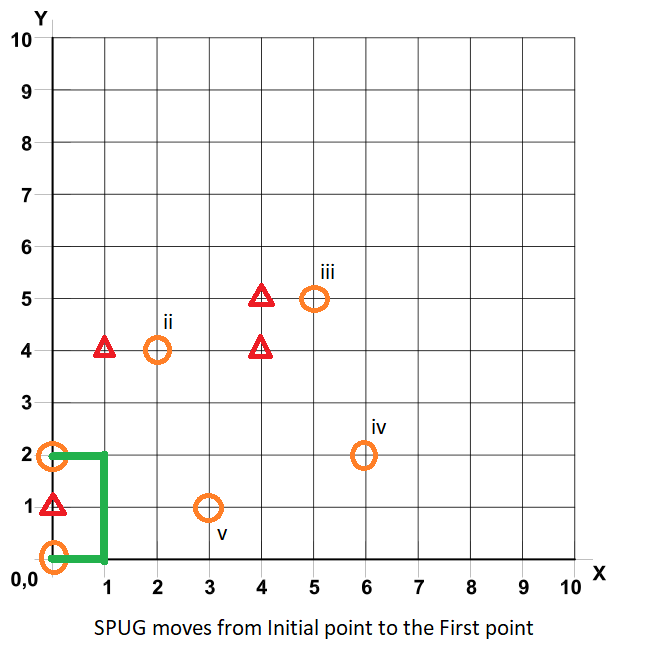
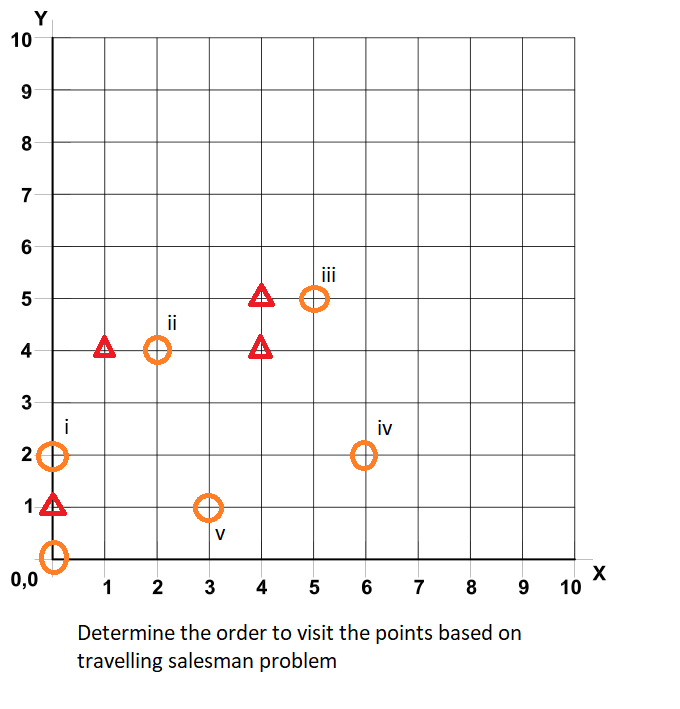
Product Location Points: [(0, 2), (2, 4), (6, 2), (3, 1), (5, 5)]

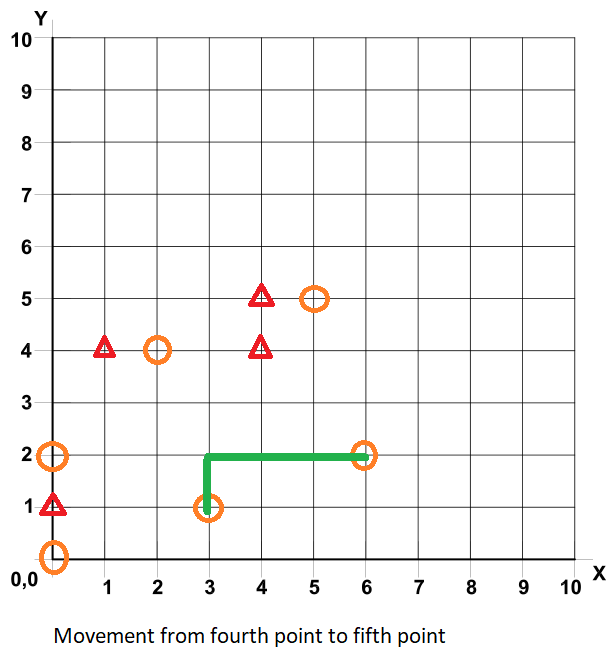
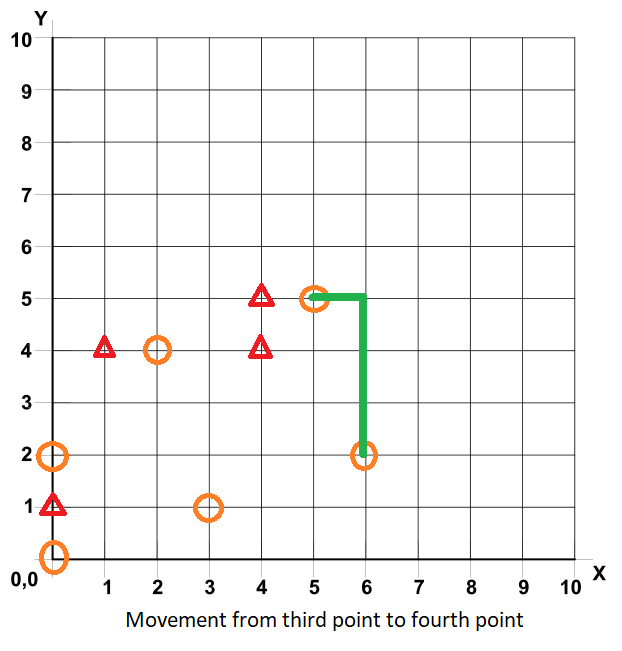
Blocking Points: [(0 ,1), (1, 4), (4, 4), (4, 5)]

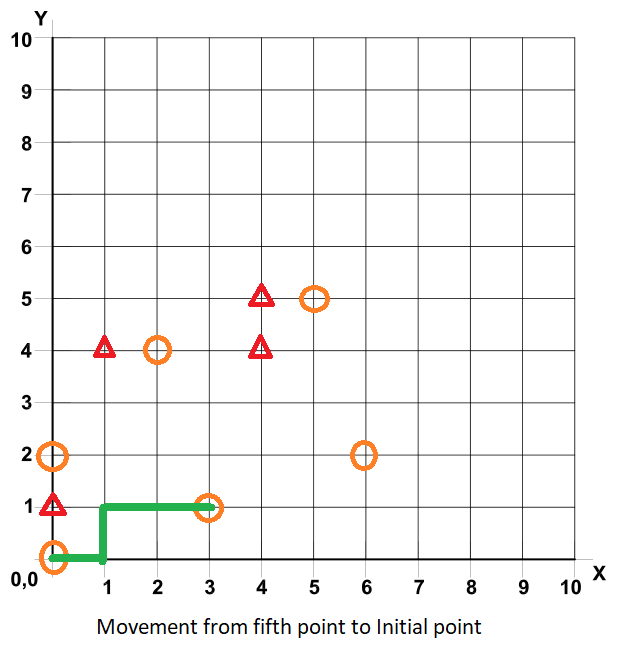
 

Afterwards the shortest path to reach all the product locations and back to the origin point is calculated. The shortest path is calculated according to the “Travelling Salesman Problem” by considering the “Manhattan” distance between the product location points. According to the travelling salesman problem a shortest possible route that visits each city and back to the origin city has to be found given a list of cities and the distances between each pair of cities.

The product location coordinates are ordered according to the shortest path calculated as show in figure 3. The origin and the first product coordinate as per the sequence and the blocking points are given to the AI Planner. AI planner considers the origin as initial point and the first product location as destination point and finds a possible solution to reach the destination by avoiding the blocking points. The SPUG reads the solution from the AI planner and moves accordingly to reach the product location as shown in figure 4. Similarly, a path is calculated between a pair of points as per the shortest path sequence by the AI Planner as shown in figure 5 to figure 10.







Once the SPUG reaches the product location a MQTT message will be sent to the server indicating that the product location is reached. User gets a notification in the mobile application to confirm if he wants to buy the product. If yes, the product will be added to the cart and a message will be sent to the SPUG to move the next product location. If no, SPUG just moves on to the next location. SPUG waits till it gets the response from the user.

While the SPUG is at a particular coordinate a path occupy message is sent to the server as a MQTT message. Server receives this message and sets that particular path as occupied. Once it moves away from the occupied path PUG send a path un-occupy message to the server. Server frees the path which was occupied before. In this was the server gets to know the free paths and the location of all the SPUGs’ in the map.

Obstacles on path-

As mentioned earlier SPUG can sense the distance to the obstacles in the front. If the distance to the obstacle is below a threshold it stops moving and LED’s glow Orange indicating that there are obstacles. If the obstacle is removed then the LED’s turn off and the SPUG starts moving back to the location it was scheduled to go.