Unit Technologies
Group4

For the Path-Planning part:

Overall we will do something about the path-planning simulation in closed scenario. It is something like in a factory, many small robots pick and drop boxes automatically without collision and deadlock with each other.

There are 5 parts in Path-Planning frame:

1. Task Scheduling, it will randomly generate the tasks for the the robots, for this part there will not be algorithms in it, just randomly generate

2. Path_Planning for Multi-vehicles, for this part, the algorithm is more like a 3d A* search, but much more complicated than that.

3. Short-Path Grant Module, this module is used to assign the grant area for each robot so they can only remain in there own area, so that there will not be any collisions.

4. Emulator, this is for the simulation of the above, which makes the demo more convinced.

5. Map, we will build a simple oriented map, it's a map that contains many nodes and we do the Path_Planning in the map.

OpenGL visualizer:

We propose to build a 3D visualizer using OpenGL to conceptualize a path planning algorithm. The idea is to have a path planning algorithm such as A-Star and its modified versions running in the back-end while the visualizer forms the front end of the project.

The visualizer will be written in C++ using pure OpenGL API commands and possibly a GPU implementation if time permits.

The back end path planner with send over the states of the various components(objects) using a TCP connection. A listener will be implemented within the visualizer to get the object states real-time and display them in the 3D visualizer.

Another feature in the visualizer would include a graphical querying system that would allow us to query the states of various objects either by clicking or hovering over them in real-time.

Technologies used - OpenGL, CUDA runtime.