# User Guide

- ## Installation Instructions

The robot runs in an OPENGL environment, on machines without OPENGL support it is required to first install the associated Dynamic Link Library (DLL) files.

As the drive name for CD drives may change from system to system the batch file INSTALL_DLL.BAT resides in the \DLL directory, this ensures that the path to the DLL files is always correct.

This file may be executed by double clicking within a windows environment or from a command prompt by navigating to the \DLL directory and typing INSTALL_DLL

If the system is networked or has restricted access it may not be possible to access the windows directory, if this is the case then the system administrator should be contacted.

The DLL's that are copied to the windows directories are,

| | FILE | DESTINATION DIRECTOY |
|---|---|---|
| o | Mfc42d.dll | c:\windows |
| o | Msvcrtd.dll | c:\windows\system |
| o | Msvcp60d.dll | c:\windows\system |
| o | Glut32.dll | c:\windows\system |

In the unlikely event that the operating system is installed on a drive other than C: or in a directory other than C:\WINDOWS these files may be hand copied into the desired destinations either from the command prompt of from within a windows interface.

The copy method is the basic DOS copy command, if files already exist the system will ask before replacing them. If the files currently on the system are more recent than those supplied with the project then it is recommended that the existing files are not replaced.

- ## Operating Instructions

The executable file may be run directly from the CD, alternatively it may be copied and executed from a local drive. The file is a standard windows executable file and will run by double clicking in a windows environment or from command prompt of a windows environment.

When the program begins a text window displays the following operating instructions,

- Use the Left Mouse Button to Draw Walls and the Right mouse Button to Erase Walls
- Both the Robot and Green Goal can be moved by Dragging with the Left Mouse Button
- The Goal will Appear Crossed-Out when Inactive. Drag the Goal to activate

In addition to the quick start pointers given above, full appreciation of the system may require some further knowledge.

The Map is drawn on a white background with black artefacts, such as walls and objects.

The robot has a fixed starting position at the bottom left of the screen, around the robot will be seen a green hashed area, this reflects the data held in the local fuzzy belief map.

The intensity of the colour represents the level of belief occurring at that location.

Hashing diagonally from top right to bottom left indicates an area that is believed to be clear of obstacles, conversely hashing from the top left to the bottom right indicates regions that are believed to be occupied, often the occupied regions are not entirely visible as the hashing falls onto map areas that obstacles exist in and which obscure the hashed line.

This hashed data is converted into global map data by a fuzzy mechanism; the global map data is shown as red grid segments on the map. Again the intensity of the colour represents the strength of belief in this position being occupied. The indication that an area is clear from obstacles comes from an absence of red markers.

The Global Target is shown as a large green square, this initially is crossed out with a large red cross, this cross indicates that the target is not active, this is to say that the robot does not have this position set as a goal.

In order to activate the target and request the robot proceed toward it, simply drag it with the left mouse button, as soon as this is done the red cross disappears and the target becomes the robots goal.

A smaller green square will periodically appear, this is a sub-goal of the robots own internal methods, it shows the immediate goal of the robot and there is no user control over this.

As the robot progresses across the map, a succession of blue markers will be left, these will be interconnected with lines either blue of red, this is a display of the positions that the robot already knows, the links between them form a collection of traversable routs. Again the user has no direct control over these points.

The blue lines that connect known positions show just the links, red lines show the calculation of routes, by following the red lines the robot is able to find a goal in any known area. This route planning is displayed for the user.

To appreciated the route planning strategy, first look at the blue node closest to the goal position, this node can be seen to have a small black spot in the upper right corner, this indicates that this node is the terminal node of the currently planned route, a black spot can be seen on the bottom left corner of the blue node that the route started from. By looking at the end node it can usually be seen that many red links exist, by following the red links toward the starting position it can be observed exactly how the route plan spread during construction, the reasons why the chosen rout was selected and the route that the robot is actually following then become visible.

- Rebuilding the project from source code

From within MS Visual C++ navigate to the \ARTIFACT directory, open the project file VERSIONONE.DSW Once open navigate the class structure with the ClassView window.

It is possible to edit several files to alter the features of the simulation, the default map may be changed by editing the BOOLMAP class.BOOLMAP.CPP

Adjusting the Membership of the Relations can be achieved by editing the MAKEMAPPINGS class definition in MAKEMAPPINGS.CPP Likewise the MAKERELATIONS class can be changed to affect the nature of the fuzzy rules.

Once edited it is necessary to re-compile the code before changes will take effect, as the project is supplied on a CD, it is first necessary to copy the \ARTIFACT directory onto a local drive.

Rebuilding and running the new executable file is achieved by selecting from the menus:

BUILD - REBUILDALL

BUILD - EXECUTE "VersionOne" .EXE