

A Guide to MOOSIVP Graphical Tools

Andrew Shafer, Michael Benjamin
Dept of Electrical Engineering/Computer Science, MIT
Cambridge MA 02139
<ajshafer@mit.edu>, <mikerb@mit.edu>

September 26, 2007

Abstract—Documentation for some of the MOOS/IVP graphical tools, including pMarineViewer and polyview. Includes instructions for clients using the tools and for programmers who want to use more advanced features.

I. INTRODUCTION

This document is designed for people who want to use the graphical tools included in MOOS/IVP. It assumes that the reader is familiar with starting and running a MOOS mission and with basic usage of a GUI. It also assumes that the tools are fully compiled and functional. In Section II, usage and configuration of pMarineViewer is described. In Section III, the mission planning tool polyview is described. Finally, some of the examples are shown in the appendices.

II. PMARINEVIEWER

For programming reference, the collaboration diagram for pMarineViewer is included in Fig. 1.

A. Configuration

pMarineViewer is launched by calling pMarineViewer file.moos (contrary to the usage instructions, no other arguments are read from the command line). Also, the file must be named .moos, or it will not be loaded by pMarineViewer.

1) *MOOS Configuration Block*: The pMarineViewer MOOS configuration block looks like:

```
//-----  
// pMarineViewer config block  
{  
  AppTick    = 4  
  CommsTick  = 4  
  
  TIF_FILE   = Default.tif  
  VEHICOLOR  = nyak200, darkblue  
  VEHICOLOR  = nyak201, hex, 08, a4, ff  
  VEHICOLOR  = nyak204, .450, .132, .55  
}
```

TIF_FILE: Optional. Default value is “Default.tif”. The path to the image file to be used as the background for the display window.

VEHICOLOR: Optional. Sets the color of a labelled object. The format of this string is “label, {colname OR hex[:], ff, ff, ff OR .050, .071, .125}” See Appendix II for a list of color names.

2) *Background Image Data*: Each tif file should have a .info file with the same name. Comment lines are prefaced with “//”. This text file can have the following entries:

img_centx and img_centy: The pixel that represents the origin of the image. UNITS UNKNOWN.

img_offset_x and img_offset_y: FEATURE UNDOCUMENTED.

centlat and centlon: The latitude and longitude (in +N and +E degrees) of the center of the image.

For example:

```
img_centx    = 0.495850  
img_centy    = 0.509000  
img_meters   = 0.048828  
//img_meters = 0.48828  
img_centlat  = 42.35849  
img_centlon  = -71.08759333
```

B. Menus and Interface

All pMarineViewer keyboard shortcuts are documented in the menu system. The mouse performs no action in pMarineViewer.

C. MOOS Variables

1) *Subscribes*: In all of the requests to plot a figure, if a label is given, requesting to plot that label again with different values will cause the figure to move to that position instead of drawing a duplicate.

AIS_REPORT and AIS_REPORT_LOCAL: An AIS report is a string of the form “NAME=name, TYPE=type, X=valx, Y=valy, SPD=speed, HDG=heading, DEPTH=depth”. The name value must match the sending community’s name. All of the variables except type are required for a valid AIS report. For type, pMarineViewer currently knows how to draw types “kayak” and “auv”. All other types are drawn with a default symbol.

GRID_CONFIG: Configures pMarineViewer to plot a new XYGrid on the display. The string must be a valid XYGrid configuration string (of the form “polygon_string@unit_string[@initial_value]”) A polygon_string is “poly[gon]:[label,LABELNAME:] segment_list”. A segment_list is a colon separated list of comma separated x,y pairs (e.g. 4,5.5:1,2.2). The unit_string is the dimensions of the rectangle to place inside the bounding polygon. It is of the form “x_width, y_width”. Usually, these are the same value. The label is used to uniquely identify the grid.

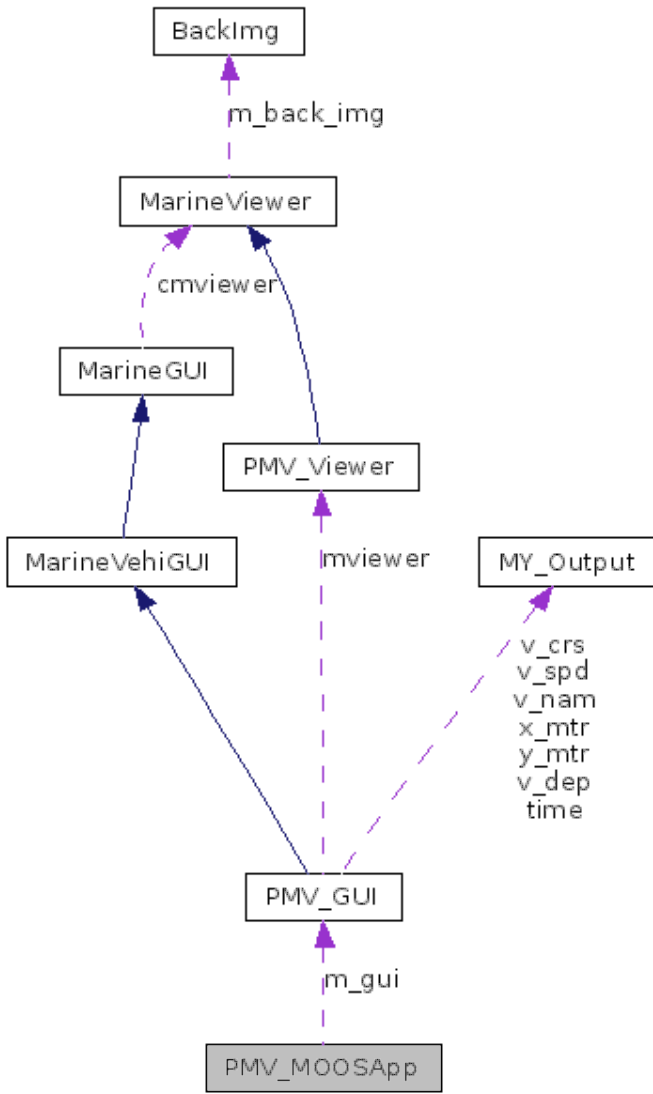


Fig. 1. A collaboration diagram for pMarineViewer

GRID_DELTA: A string to update the grid. It is of the form “LABELNAME@index, old_val, new_val[,old_utility, new_utility][:index, old_val, new_val...]”.

VIEW_POLYGON: Plots the specified polygon. The string is a valid XYPolygon initialization string. Polygons can also be of the form “radial:xval, yval, radius, num_points[,snap_value[,LABELNAME]]” to approximate a circle. (An arc can also be plotted, see XYPolygon.cpp for details.)

VIEW_SEGLIST: Plots the specified seglist. The string is a valid XYSeglist initialization string. It is of the form “[label,LABELNAME:]segment_list”. It can also be of the zigzag form (not described here. See XYSegList.cpp for a description.

VIEW_POINT and VIEW_CIRCLE: VIEW_CIRCLE IS NOT IN SUBSCRIPTION LIST. VIEW_POINT plots a dot. The string is a valid XYCircle initialization string. It is of the

TABLE I
A LIST OF SHORTCUTS TO REFERENCE IMAGE FILES IN POLYVIEW.

mit OR charles	AerialMIT-1024.tif
wmit OR wireframe OR wf	WireFrameMIT-1024.tif
mb OR monterey	Monterey-2048.tif
mbd	Monterey-2048-30-30-100.tif

form “x_val,y_val,radius[,LABELNAME]”.

TRAIL_RESET: Forces pMarineViewer to “forget” the current trails for any vehicles.

2) Publishes: No MOOSDB writes are created by pMarineViewer.

III. POLYVIEW

A. Configuration

polyview is launched by calling `polyview image.tif file1 file2...` with arguments in any order. The image.tif file is the same as the one used by pMarineViewer. A list of shortcuts for common backgrounds is in Table I. The user can also specify “-noimg” to force no background image to load. All of the other arguments are scanned as text files for Polys, Grids, Arcs, Circles, and Hexagons.

All readable lines are non-blank and do not begin with ‘#’. Each line is in the format “key = initialization_string”.

A Polygon is keyed with “polygon”, “points”, or “radial”. e.g. “polygon = polygon:label,A: 0,0: 1,1: 1,0”

A Grid is keyed with “searchgrid” (abbreviated “sgrid”) or “fullgrid” (abbreviated “fgrid”). It is followed by “=” and then the XYGrid initialization string. See XYEncoders.cpp, StringToXYGrid for information on the “fullgrid” implementation.

An Arc is keyed with “arc”. i.e. “arc = x, y, radius, left_angle, right_angle” (both angles in degrees, 0 is straight up).

A Circle is keyed with “circle”. i.e. “circle = x, y, radius[,label]”

A Hexagon is keyed with “hexagon”. i.e. “hexagon = x, y, radius_to_points”

All of the detected objects are loaded and displayed in the polyview window.

B. Menus and Interface

The interface for polyview is very similar to the pMarineViewer interface. In the EditMode menu, the various editing operations change the way that a left-click functions in the interface. The EditMode operations only work on the currently selected object. The current object is changed in the Polygons menu, or by using + and -.

A new polygon can be created by right-clicking in the interface, or by clicking Polygons->Create New.

In the Polygons menu, DumpSpec will output the string that represents the current figure into the terminal window that called polyview. This is useful for cutting and pasting into text files for other programs (such as moos and behavior files).

NOTES: Duplicate doesn’t seem to work. The test modes are also undocumented.

REFERENCES

- [1] Philip E. Agre and David Chapman. What Are Plans For. *Robotics and Autonomous Systems*, 6:17–34, 1990.

APPENDIX I

APPENDIX EXAMPLE

This is an example in the appendix.

APPENDIX II

COLOR NAMES FOR PMARINEVIEWER

antiquewhite, aqua, aquamarine, azure, beige, bisque, black, blanchetalmond, blue, blueviolet, brown, burlywood, cadetblue, chartreuse, chocolate, coral, cornsilk, cornflowerblue, crimson, cyan, darkblue, darkcyan, darkgoldenrod, darkgray, darkgreen, darkkhaki, darkmagenta, darkolivegreen, darkorange, darkorchid, darkred, darksalmon, darkseagreen, darkslateblue, darkslategray, darkturquoise, darkviolet, deeppink, deepskyblue, dimgray, dodgerblue, firebrick, floralwhite, forestgreen, fuchsia, gainsboro, ghostwhite, gold, goldenrod, gray, green, greenyellow, honeydew, hotpink, indianred, indigo, ivory, khaki, lavender, lavenderblush, lawngreen, lemonchiffon, lightblue, lightcoral, lightcyan, lightgoldenrod, lightgray, lightgreen, lightpink, lightsalmon, lightseagreen, lightskyblue, lightslategray, lightsteelblue, lightyellow, lime, limegreen, linen, magenta, maroon, mediumblue, mediumorchid, mediumseagreen, mediumslateblue, mediumspringgreen, mediumturquoise, mediumvioletred, midnightblue, mintcream, mistyrose, moccasin, navajowhite, navy, oldlace, olive, olivedrab, orange, orangered, orchid, palegreen, paleturquoise, palevioletred, papayawhip, peachpuff, pelegoldenrod, peru, pink, plum, powderblue, purple, red, rosybrown, royalblue, saddlebrown, salmon, sandybrown, seagreen, seashell, sienna, silver, skyblue, slateblue, slategray, snow, springgreen, steelblue, tan, teal, thistle, tomato, turquoise, violet, wheat, white, whitesmoke, yellow, yellowgreen