

Technical Specifications

Software Details

This software was created with the purpose to analyze biking trails' difficulty in physical strain. Users can import their own elevation data and polyline data and can create their own polylines within the program. Trail Analyzer is a stand-alone desktop application programmed in C#. It uses the following open-source DLL's and extensions: DotSpatial, ZedGraph, Visual Basic, and GDAL. It was created by Michael Scott and Stephen Duncan. The source code can be downloaded from <https://github.com/GISBikeDragons/TrailAnalyzer2K15>.

Compatibility and Limitations

This program was designed for the Windows operating system. Use is not guaranteed to work on Macs or other operating systems. For any questions, issues, or troubleshooting assistance, contact. Stephen Duncan (stephenduncan64@gmail.com) or Michael Scott (mykolescot@gmail.com)

License

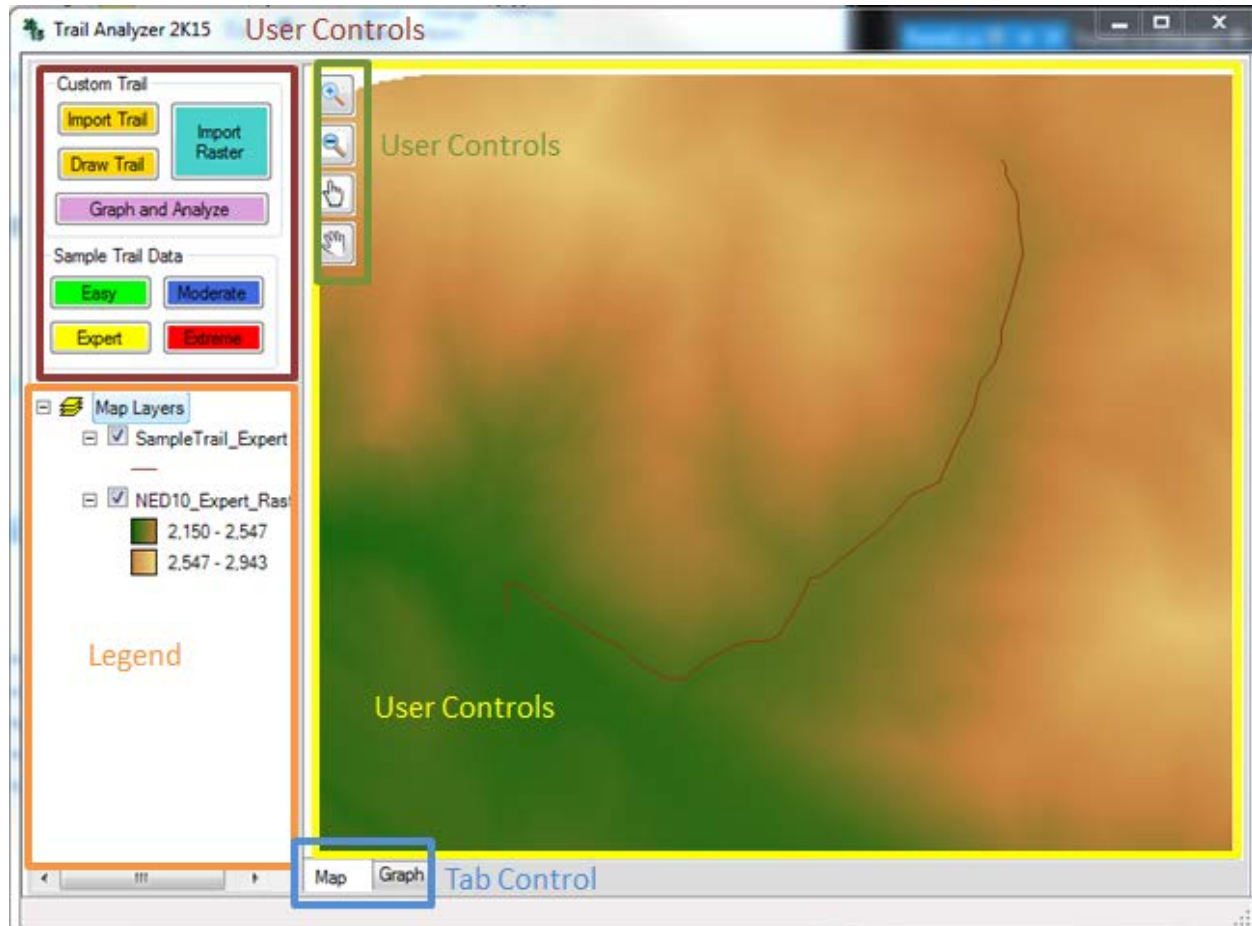
Trail Analyzer 2K15 created by Stephen Duncan and Michael Scott is licensed under the MIT license.

Software Design

Purpose of Design

This programmed was designed to be simple and easy to use. Most of the space is devoted to the map. A small area on the left is devoted to the buttons and the map legend. The graph and analysis data is also a major aspect of the program so it was given a tab to use the same space devoted to the map. The map navigation buttons are important to the user but don't need to be very large so they were tucked up nicely in the corner. The following graphic shows the location of each of these elements.

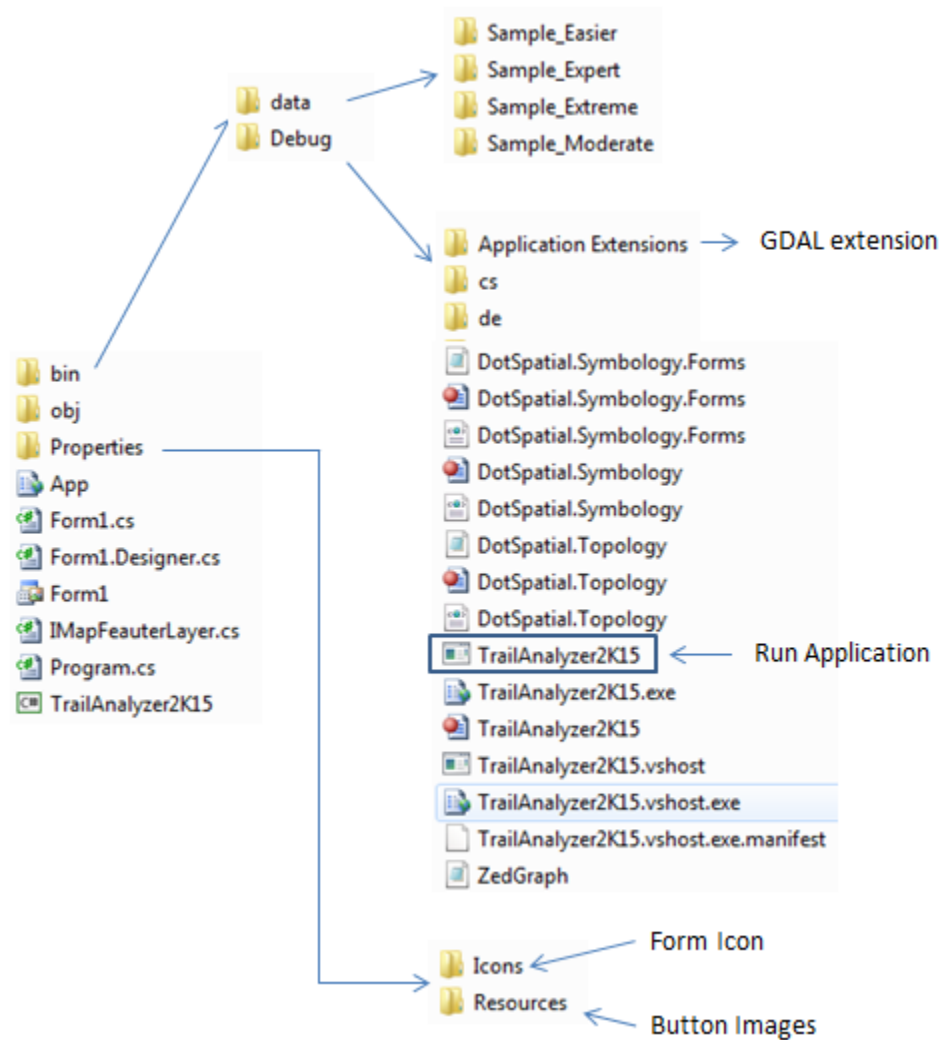
Technical and Design Documentation

**Main Files**

This program is organized by plugins, data (which includes the sample raster data and sample polyline data), application extensions, properties, and the C# files.

The following graphic shows how the files are organized. The executable file is outlined in black.

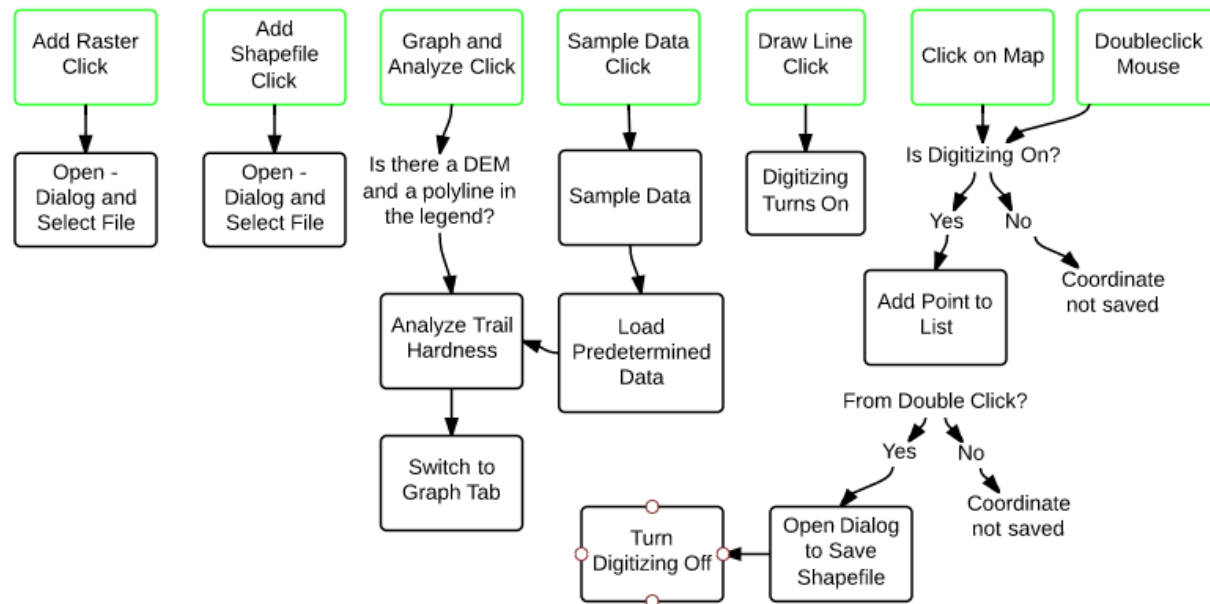
Technical and Design Documentation



Code Structure

The following flowchart shows the basic structure of the code. The top row represents actions that the user can take. The remaining rows represent the functions that are called.

Technical and Design Documentation



Key Functions

Function Name	Details
AnalyzeTrailHardness(IMapRasterLayer, IMapFeatureLayer)	Steps through each vertex in the polyline and pulls out the distance, slope, and elevation of each segment. Calculates values to populate the user-viewable ratings. Plots the elevation graph of line. Determines the physical difficulty of the trail based on the algorithm within the function.
ResizeForm()	Sets the location of both splitcontainers.
btnDrawTrail_Click_1(event)	Gets program ready to digitize line. Changes the selected index of the tab control back to the map. Changes the cursor. Updates the StatusStripLabel.
mapMain_MouseClick(event)	Stores coordinates of each click in a list if Boolean "amDigitizing" is true.
mapMain_MouseDoubleClick(event)	When "amDigitizing" is true this event stores the last coordinate then opens a dialog box for the user to save their line as a shapefile.
btnGraph_Click(event)	Runs the trail analysis for the DEM and Polyline in the legend.
btnAddRaster_Click(event)	Opens dialog box for user to import raster data.
btnAddLayer_Click(event)	Opens dialog box for user to import polyline shapefile data.
sampleData(6 Strings)	This function is used by the 4 sample data buttons to load the sample data and update the graph labels.