

Hexagon Mining Assignment

Version 1.3.0

Contents

History.....	2
Story-1: to query for the closest point id in my point set compared to a query point.....	3
Acceptance Criteria.....	3
Story-2: be able to shift and translate all my points by a vector	3
Acceptance Criteria.....	3
Story-3: be able to specify a number of points as closest point.....	3
Acceptance Criteria.....	3
Story-4: be to interact with the program using a Graphical User Interface	4
Acceptance Criteria.....	4
Appendix I: Source File Format	4
Appendix II: Submitting the Results.....	4

History

Version 1.3 Series:

- 1.3.0 – September 22nd, 2016
 - o Changed header

Version 1.2 Series:

- 1.2.0 – December 11th, 2014
 - o Added a forth story for GUI design

Version 1.1 Series:

- 1.1.2 – November 20th, 2014
 - o Clarified the range of possible values for point data.
- 1.1.1 – November 19th, 2014
 - o Eric Hambright helped corrected typos.
- 1.1.0 – November 17th, 2014
 - o Broken down in terms of stories.

Story-1: to query for the closest point id in my point set compared to a query point

As a User, I want to query for the closest point id in my point set compared to a query point, so that I find locations for building my drill plan.

It is important for me to be able to enter a co-ordinate and compare it against all my known points in the database, and locate the closest point in that database.

Acceptance Criteria

1. I expect to be able to specify a source for my points (see **Appendix I: Source File Format**).
2. I expect to be able to specify a query point.
3. I expect to be able to use positive or negative values of the query point.
 - i.e. X=1000,Y=10,Z=12 or X=-12, Y=2000, Z=-1201
4. I expect that the point ID returned will be the closest point to the query point.
5. I expect that this is documented.

Story-2: be able to shift and translate all my points by a vector

As a User, I want to be able to translate all my points by a vector, so that I can apply a primitive adjustment to my coordinate system to match my query point's coordinate system.

The point I am querying might be in a different coordinate system than the point data itself, I need a way to shift that data.

Acceptance Criteria

1. I expect to be able to enter an X, Y, and Z coordinate, and have my entire source data shift by that.
2. I expect that this is documented.

Story-3: be able to specify a number of points to return closest to the query point

As a User, I want to be able to specify a number of points as closest point, so that I can select the best point to work from.

Acceptance Criteria

1. I expect to be able to specify the number of points to report as closest.

Story-4: be to interact with the program using a Graphical User Interface

As a User, I want to be able to interact with the program using a Graphical User Interface, so that I don't need to understand how a command line works

Acceptance Criteria

1. I expect all functionality to be accessible through the interface
2. I expect to see the results of the program on the interface

Appendix I: Source File Format

The source file is a CSV file.

Line 1 is the header.

Each remaining point is the data.

X	Y	Z	ID	Description
2221.367	4879.073	1971.682	1	
2169.996	5071.648	2321.178	2	

The point data can either have positive values or negative values.

Appendix II: Submitting the Results

The results should be submitted in a zip file containing all the source files.

The zip file should contain at least one file called README that provides the following details,

1. How to “build” the program.
2. How to “start” the program.

Any document format can be used for the README, but one should stick to the common most reasonable format for your end user.

There should a separate document called DOCUMENTATION. This documentation should show how to run the program in order to satisfy the stories.

Any document format can be used for the DOCUMENTATION, but one should stick to the common most reasonable format for your end user.