BestXYZ Processor

Requirements, Design, Implementation, Testing (RDIT)

|  |  |
| --- | --- |
| User Story | Text |
| SETPOSITION | The user was able to specify the GPS Antenna Position as ECEF (E, F, G) coordinates in meters in the main window |
| GETPOSITION | The user was able to load the GPS Antenna Position as ECEF (E, F, G) coordinates in meters from a file using a button in the main window |
|  |  |

1. Requirements
   1. The user was able to type into 3 separate text fields corresponding to E, F, and G coordinates in meters on a GUI interface.
   2. The user was able to load E, F, G coordinates in meters from an input text file by searching for the file on their local computer.
2. Design
   1. Use a framework to create a main GUI dialog
   2. Add in 3 text fields for inputting GPS Antenna position as ECEF(E, F, G) in meters.
   3. Add in a Button to browse using a standard file open dialog.
      1. Browsing for a file fills in the three input text fields for antenna positions.
3. Implementation
   1. Create a QT Window
   2. Use the QTCreator GUI Editor to organize GUI objects
      1. Add three QTextFields text fields onto the main QT Window
      2. Add three Qlabels onto the QT Window
      3. Add a QButton onto the QT Window
         1. Pops up a file selection window
         2. On a valid file selection parses a file that contains the E, F, and G values separated by commas. For example: “1.0,2.5,3.9”
         3. Invalid files with non-numerical inputs fills the E, F, and G values with zeros. For example: “AAA, BBB, CCC”
         4. Malformed files result in an error QMessageBox and don’t fill the values. For example: “3.4, 2.4”
      4. Add Horizontal and Vertical QSpacers to organize the Qlabels and QTextFields
4. Testing  
   Test Setup: A computer with the BestXYZ Processor application loaded and containing at least one reference receiver dataset. An invalid antenna position file containing “AAA, @$#D, FFF”. A malformed antenna position file containing “1.3, 2.5”.
   1. Open the BestXYZProcessor application
   2. Verify that a text entry field is present on the GUI
   3. Verify that the user can type into the text entry field
   4. Verify that a “Load Position” button exists under the text entry fields
   5. Click the “Load Position” button to open the file browser and then click on the cancel button
   6. Verify that no values populated the text entry fields
   7. Browse for the malformed file and verify that a QMessageBox warning prompts the user and no values populate the text entry fields
   8. Browse for the invalid antenna position file and verify that each text entry field is populated with 0
   9. Browse for a valid file and verify that the text entry fields are automatically populated with the specified float values
   10. Run the application and verify in the output data file that the entered coordinates were used. (EAnt = Erecord-deltaE)