# Appraisal

## Completion of Objectives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective** | | **Done** | | **Comments** |
| The system will be able to plot the various loci on an Argand diagram. | | **✓+** | | The program can also plot dual rays: . |
| The system will also shade in inequalities for each of these loci with a translucent colour fill. | | **✓-** | | The program cannot shade sectors. |
| Values of and will be labelled on the diagram. | | **✓-** | | The program labels explicitly entered points, but not points of interest on other loci. |
| The system will plot individual complex numbers as points: | | **✓+** | | The program can solve many more complex equations for points as well. |
| The system will display multiple plots on one diagram. | | **✓** | | The plots cannot be reordered once added. |
| Diagrams can be saved to and loaded from .arg files. | | **✓** | | The Windows installer also allows .arg files to be associated with the program. |
| It will be possible to zoom in on the diagram by scrolling the mouse wheel while hovering over it, or by moving a slider. | | **✓** | | When zooming, the scale of the axes changes according to a logarithmic scale. |
| It will be possible to pan around the diagram, by dragging the mouse over it, or by using a coordinate input. | | **✓** | | If, while panning, the origin moves off-screen, the axes will cling to the edges of the screen so the scale is still visible. |
| All plots will have configurable colours – either from a set collection of swatches or with a colour picker tool. | | **✓** | | The Qt default colour picker is used. |
| All inequalities will have configurable alpha transparency. | | **✓** | | This is configurable via the colour picker. |
| The program will not require a mouse to use – all inputs will have tab indices set correctly. | | **✓** | |  |
| The program will have a text input for entering complex equations. | | **✓** | | Equations are entered in plaintext. When an input does not validate, the background of the input area turns red. |
| The program will be capable of converting strings into equations which can be plotted on the diagram. | | **✓** | |  |
| The program will have built-in information (i.e. a tooltip / popup) about how to enter equations in the correct format. | | **✗** | | There was not enough time to implement this feature. Guidance on the correct format for input is given in the user manual. |
| The program will display current plots in a list, where they can be edited or removed. | | **✓** | |  |
| The minimum resolution of the program will be, at most, 1024x768. | | **✓** | | The resolution is suitably low to be used on any Windows monitor. |
| The diagram will be legible on a 1024x768 resolution projector from the back of an average-sized classroom. | | **✓** | |  |
| The program will run on at least an Intel x64 Windows 7 PC. | | **✓** | | Supported platforms are listed in the user guide. |
| The program will have a configuration menu which allows the following to be altered: | | **✓-** | | The menu exists, but doesn’t include all planned functionality (see below). |
| Whether labels are displayed for points. | | **✓** |  |
| Whether the axes are labelled. | | **✓** |  |
| The font size used for labels. | | **✗** | There was not enough time to implement this. However, the default font size is adequately legible on most displays and projectors. |
| The line thickness used in the diagram. | | **✓** | The thickness is bounded between 1 and 10 pixels. |
| The program will save the diagram configuration state to the .arg file. | | **✗** | | This is no longer planned, as preferences are independent of diagrams. |
| The program will remember the last window size used, and revert to it on start-up. | | **✗** | | This is no longer planned. |

## User Feedback

**Q: How does Argand Plotter compare to your expectations?**

*A: Argand Plotter exceeds my expectations. It is incredibly easy to use and makes a nice link between the algebra and the graphs.*

**Q: Please comment on the usability of Argand Plotter.**

*A: It is very intuitive, with all operations appearing in the way that I would expect. The zoom and drag functions are particularly friendly. It's very easy to enter equations and the real-time nature is excellent for helping students to learn.*

**Q: What issues do you have with Argand Plotter (if any)?**

*A: It would be nice if it would accept either or for the imaginary unit. Also, most students use rather than the decimal equivalent.*

**Q: Do you have any suggestions for improvements to Argand Plotter?**

*A: If possible, it would be nice to extend it to include arg inequalities i.e. as these are graphs which students find particularly challenging.*

My client was also asked to rate how much he agreed with the following statements:

|  |  |
| --- | --- |
| **Statement** | **Response** |
| Argand Plotter is a useful tool for teachers. | Strongly Agree |
| Argand Plotter is a useful tool for students. | Strongly Agree |
| Argand Plotter is easy to use. | Strongly Agree |
| Argand Plotter performs all the functions I need. | Agree |
| I will use .arg files to save diagrams. | Strongly Agree |

## Analysis of User Feedback

Considering my client’s feedback on the intuitiveness of Argand Plotter, I think my objectives for UI design have been effective in making the program easy for both students and teachers to understand.

Overall, based on my objectives and my client’s feedback, I consider Argand Plotter to be a resounding success.

## Improvements And Suggestions

***“It would be nice if it would accept either i or j for the imaginary unit.”***

My original intention was to use for the imaginary unit, but I made the decision to use , since it follows the Python convention. It was simply easier to pass an input term directly to the complex built-in function than to switch the symbol beforehand. It would still be possible to alter the parsing code to make a valid imaginary unit.

***“Also, most students use rather than the decimal equivalent.”***

While my parser does recognise multi-letter words (for example, the sin, cos, and tan functions), it does not currently have a mechanism for treating such a symbol as a number, rather than a function. While having pi as a suitable input is a possible feature, it would probably require a fair amount of work. If done correctly, however, it should also be easy to implement other constants such as , , etc.

***“It would be nice to extend it to include arg inequalities i.e. ”***

This would be much more challenging, as the parser is currently designed for relational statements with two sides and one relational operator. One possibility is the addition of drawing of sectors from the positive x-axis only, and then forming a statement as above by comparing the two simultaneous halves. Currently, the program isn’t capable of drawing sectors, but the main reason for this is the complexity of the required drawing code, rather than parsing (most of what would be needed is already in place).