

LUNAR PIT PATROL

PRESENTS

CRATER STATS



TEAM INTRODUCTION



Trent Hare
Sponsor



Marc Hunter
Sponsor Associate



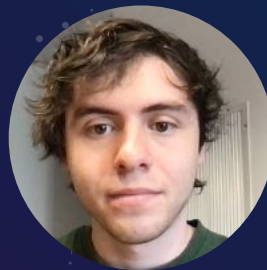
Evan Palmisano
Team Lead



Ibrahim Hmood
Customer Communicator



Caden Tedeschi
Architect



Alden Smith
Release Manager



Levi Watlington
Recorder

PROBLEM STATEMENT.



The usability of Crater Stats needs improvement!

“Scientists should not have to go through a boot camp to use Crater Stats...”

- Trent Hare



PROBLEM STATEMENT

Key Problems

CLI & Old GUI

- Crater Stats is hard to learn
 - CLI is not the easiest and has little documentation...
 - Previous GUI attempts were not user friendly...
- Crater Stats is inefficient in terms of time
 - Rewrite an entire command for a slight change...
 - Old GUI was showing everything everywhere all at once...



PROBLEM STATEMENT.

The usability of Crater Stats needs improvement!

CLI is not user friendly - Designing a GUI is hard

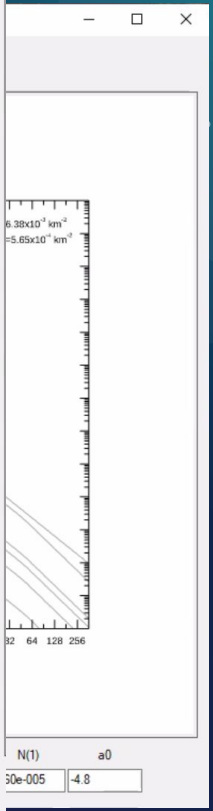
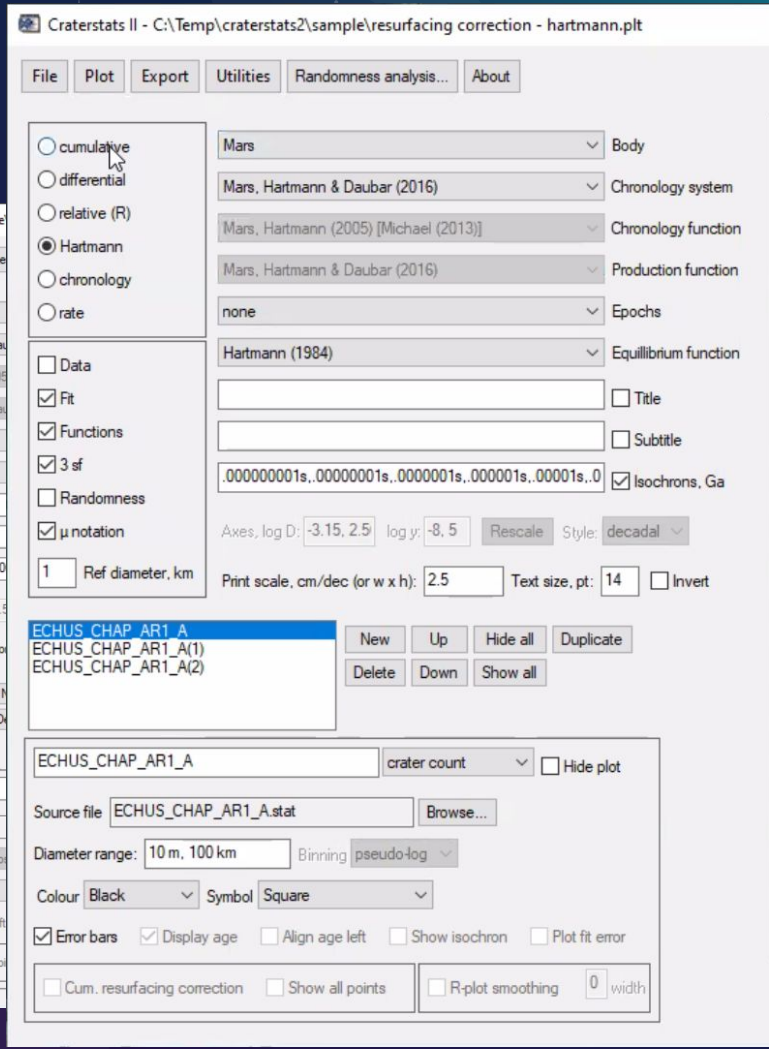
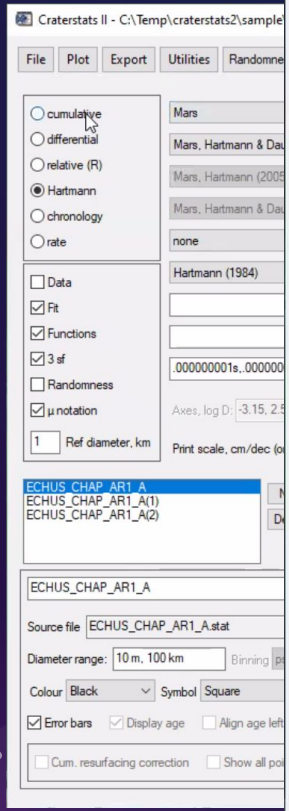
From this...

```
craterstats -cs neukumivanov -p source=%sample%/Pickering.scc,psym=o,binning=10/decade -p type=d-fit,range=[.2,.7],isochron=1 -p range=[2,5],colour=red
```

PROBLEM STATEMENT

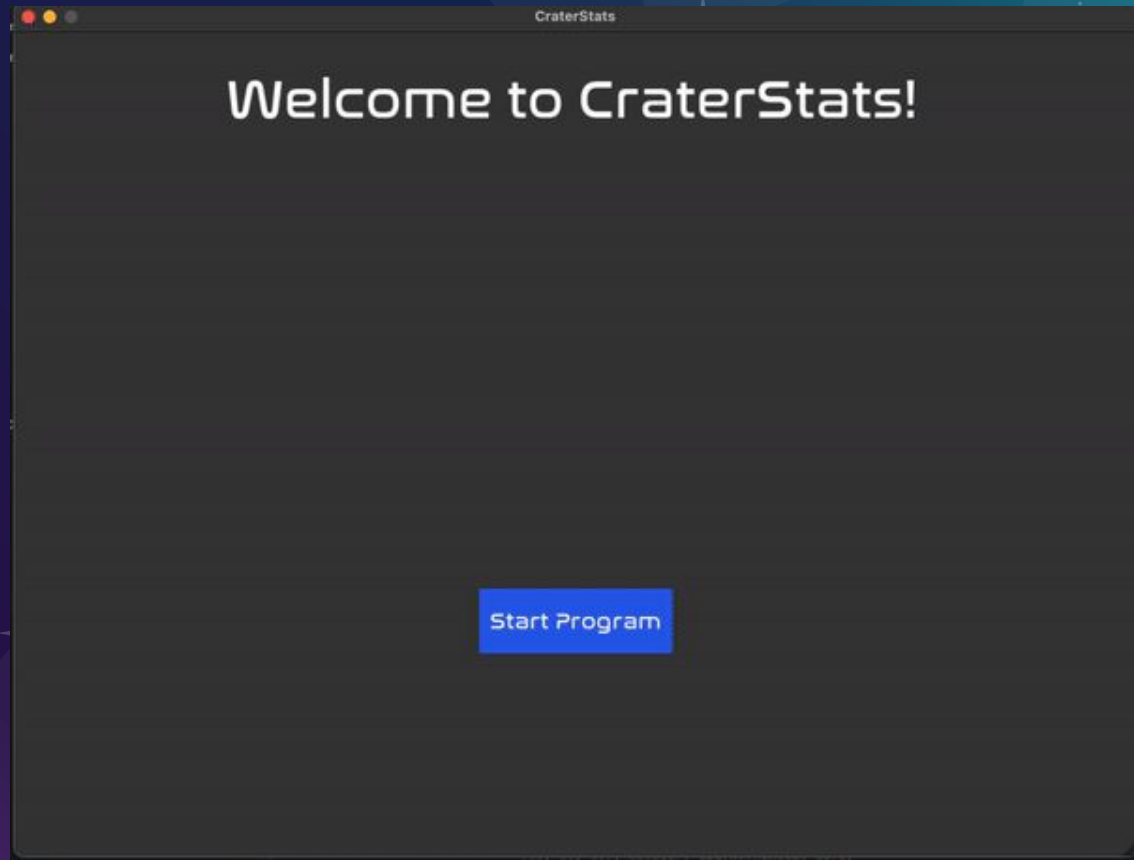
To this...*

*or similar



PROBLEM STATEMENT

Our current prototype...



SOLUTION OVERVIEW

- A GUI with a modern layout featuring:
 - Easily accessible options
 - Simplistic navigation
 - Large plot display
 - Tabbed pages for options



SOLUTION OVERVIEW



Tabs: Format, Stats, Plot

Application Welcome

Logo

Loading Progress Bar

Tabs: Format, Stats, Plot

Format Page

Tool Bar

File Loading & Plot Options

Tabs: Format, Stats, Plot

Statistics Page

Tool Bar

Statistical Inputs &
Calculation Customization

Tabs: Format, Stats, Plot

Plot Page

Tool Bar

Plot View

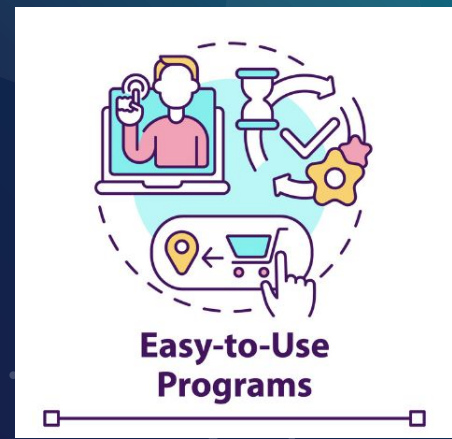
KEY REQUIREMENTS

- Acquiring requirements:
 - Meeting with our sponsor Trent
 - Studying the previous CraterStats GUI application
 - Checking the specifications in our CapStone project description



KEY REQUIREMENTS

- Domain-level requirements:
 - The application needs to be easy to use and navigate by users
 - Display correct graph and information to user
 - Accept graph files and be able to save current graph to a file
 - Able to be used without downloading everything for the application



KEY REQUIREMENTS

- High level functional requirements:
 - Easy navigation through the frames of our GUI
 - Keep the graph correctly updated
 - Ability to upload and download graph files to and from the application
 - Able to be run as a web application with the same functionality as the GUI application



KEY REQUIREMENTS

- Functional requirement deep dive:
 - Requirement: Easy navigation
 - Group the settings options with the same type of purpose together
 - Put the different types of settings options on different frames of the GUI
 - Create tabs to give immediate access to any frame no matter what frame the user is on
 - Give the tabs descriptive names so the user knows what each frame will hold

The performance of all of these lower level requirements is very good since they give good information to the user that makes the use of the application easier

RISKS

- ✦ Change in performance
 - Keeping CLI speed in GUI
- ✦ Incorrectly displayed data
 - Making sure program functions are called correctly for the options selected



FEASIBILITY

- Tkinter → DearPyGUI
 - Switch GUI library for better functions and newer theming
- Better cross-platform compatibility
 - No OS specific themes with the same amount of features
- Integrating current features
 - Python GUI with Python program makes for an easier time

SCHEDULE

	Months									
	Jan	Feb	Mar	Apr	May	Sep	Oct	Nov	Dec	
Task 1	■	■		✦						
Task 2		■	■	■	■	◆				
Task 3										
Task 4							■	■	■	
Task 5						■	■	■	◆	
Task 6										
Task 7										
Task 8										



CONCLUSION



In conclusion,

- Defined direction with established prototype
- Switching python libraries
 - **Tkinter** → **DearPyGUI**
- Next step is to associate stat functions with GUI input options.



THANK YOU FOR
YOUR TIME!

QUESTIONS

