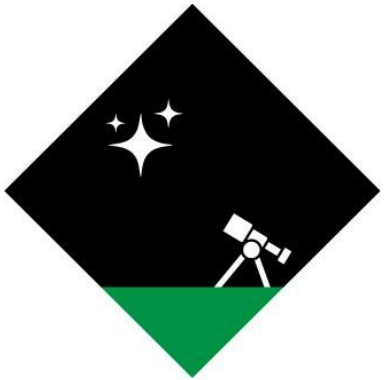
# Team Dark Sky Design Review



**Team: Luke Thompson, Justin Ceccarelli, Jordan Tatum**

Mentor: Daniel Kramer

Faculty Leader: Michael Leverington

# Our Client: The Navy Precision Optical Interferometer (NPOI)


Jim Clark:

Director and
Chief Engineer

Peter Kurtz:

Software Engineer

Henrique Schmitt:

Astronomer and Head
of Interferometry

# Research at the NPOI

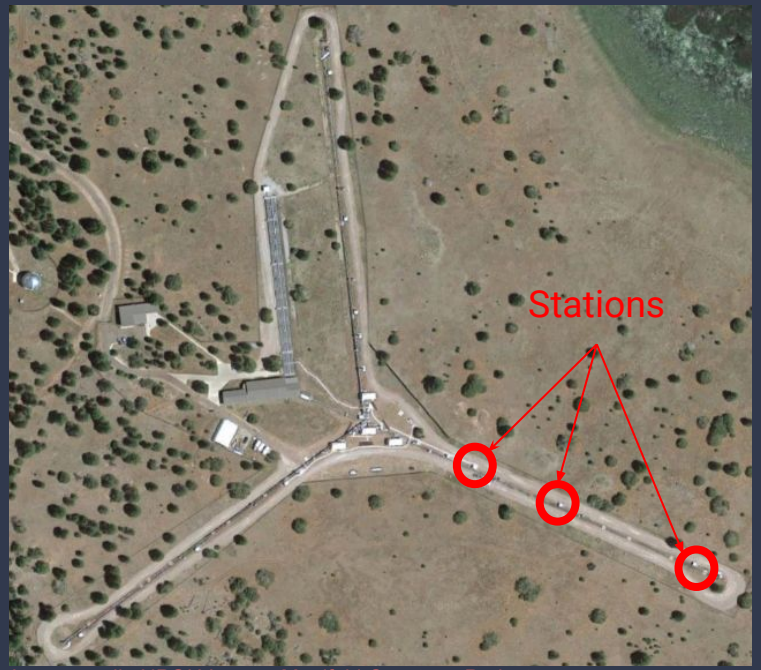



Image credit: NPOI Vacuum Manifold Capstone Project
https://ceias.nau.edu/capstone/projects/ME/2021/21F03_NPOI/About.html

- Interferometry - Creates an interference pattern
- Largest baseline interferometer in the world
- Specializes in studying binary stars
- Stellar navigation

# Problem Statement


Photo credit: Luke Thompson

- A night of observation costs roughly $12,000
- **obsprep** - (Observation Preparation)
- Unnecessary use of man-hours in training and installation
- Not 100% reliable

# Causes

- Built with retired Python 2
- Difficult installation process
- Learning curve
- Critical bug in 2020

# Requirements:

- Acquisition:
  - Original obsprep
  - Software and Astronomy staff
  - Direct input from users
- Functional Requirements:
  - Intuitive, faithful GUI
  - Access a catalog of stars and choose precise observation specifications
  - Straightforward installation and maintenance

# Solution Overview

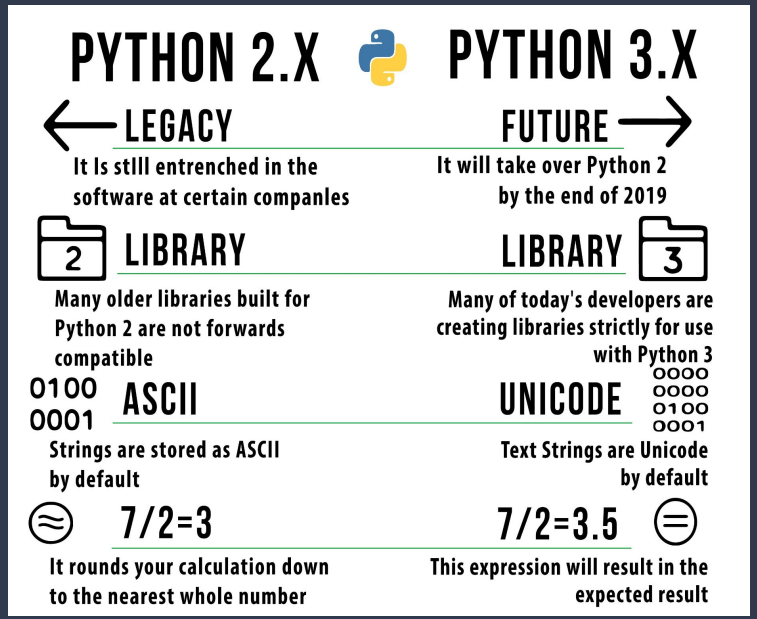
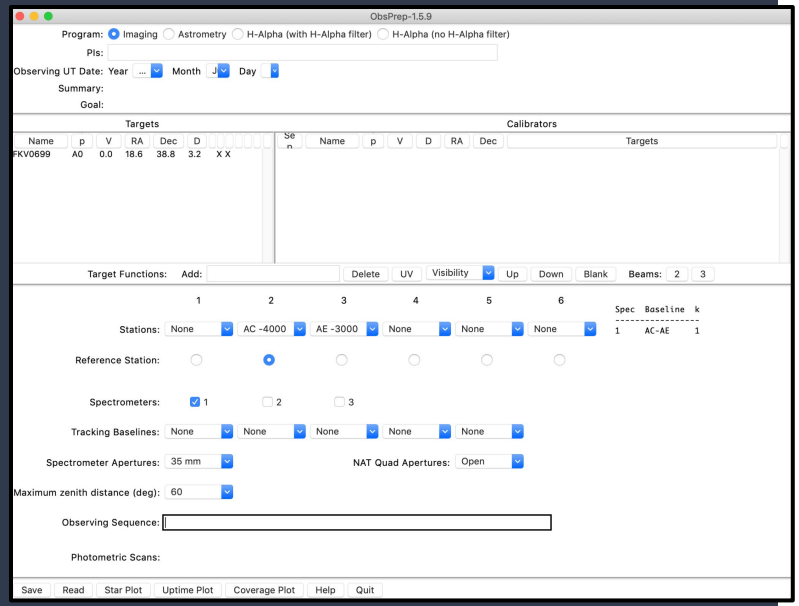



Image credit: GeeksForGeeks
https://www.geeksforgeeks.org/python2-vs-python3-syntax-and-performance-comparison/

- Intuitive, structured, and simple
- Updating obsprep involves:
  - Main GUI file
    - TKinter
  - Client supplied files - math and astronomical functions
  - C wrapped Files

# Update main GUI file

Original GUI




- TKinter, a graphical framework for Python
- Update functions and widgets to match python 3 syntax
- Update matplotlib functions

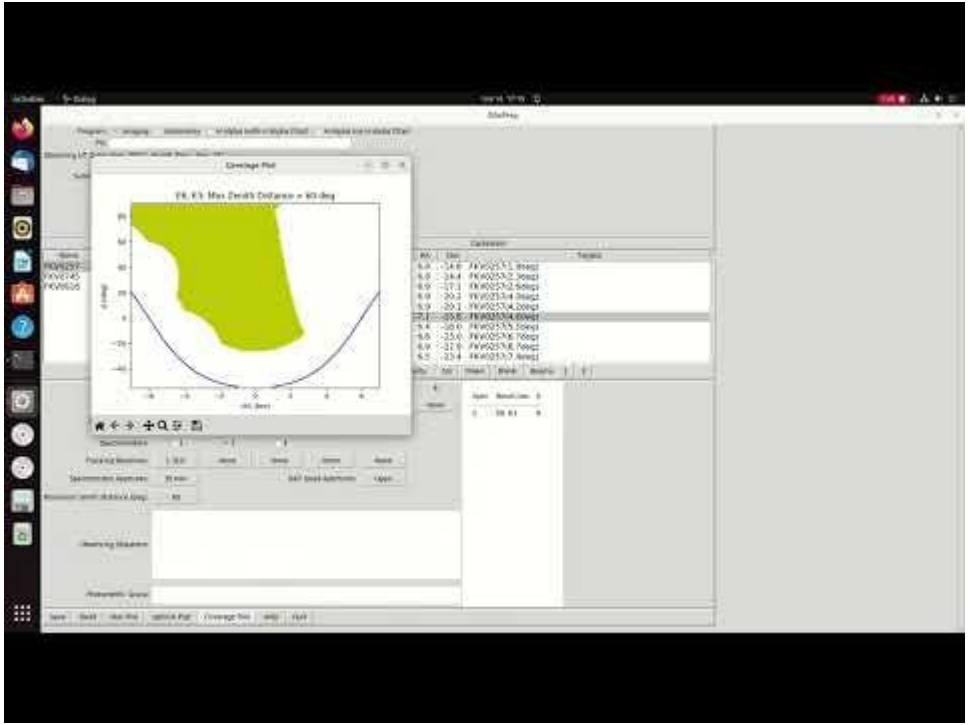
# Update client supplied files

- The original files use Python 2 syntax
- Replace old Python functions
- Update syntax throughout program

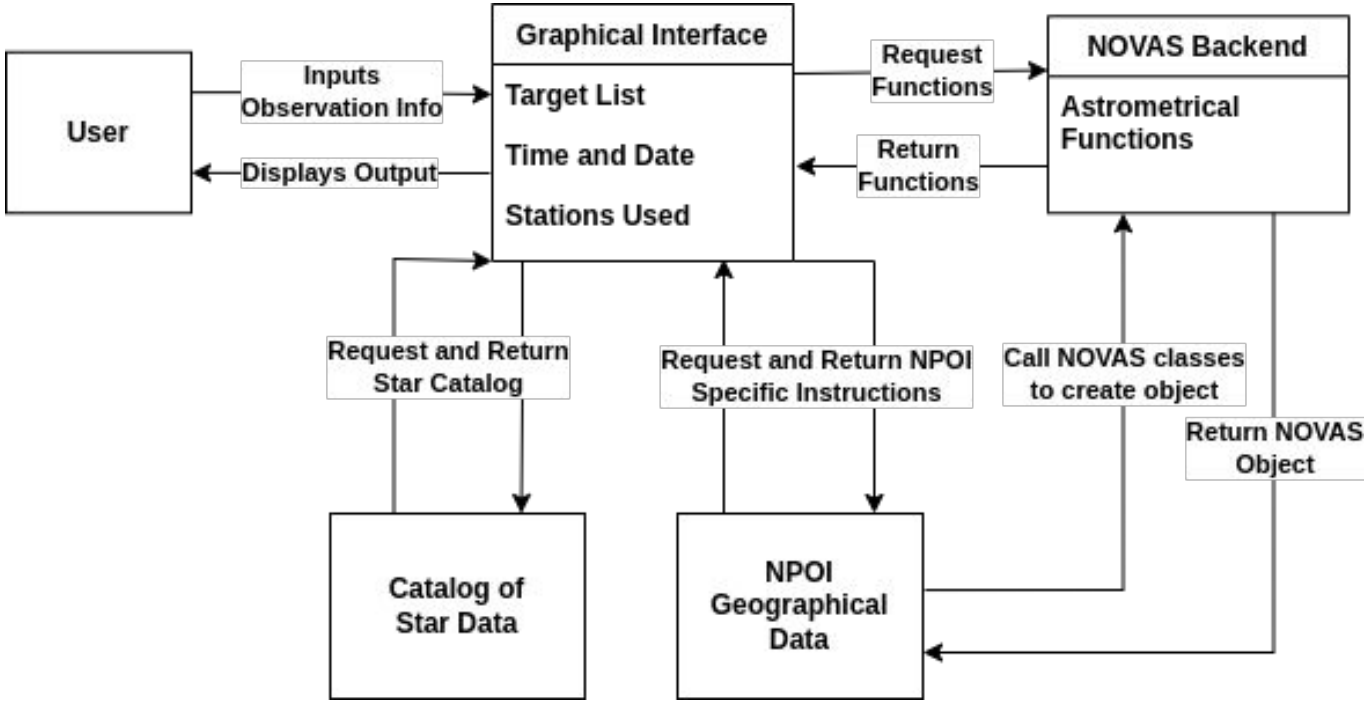
# Update C wrapped Files

- Original C wrapped files used python 2 syntax
- Updated to modern Swig
- Create a new interface file
- Rewrap C files

# Prototype Review



# Architecture Overview



# Challenges and Resolutions

Working with pre-existing code

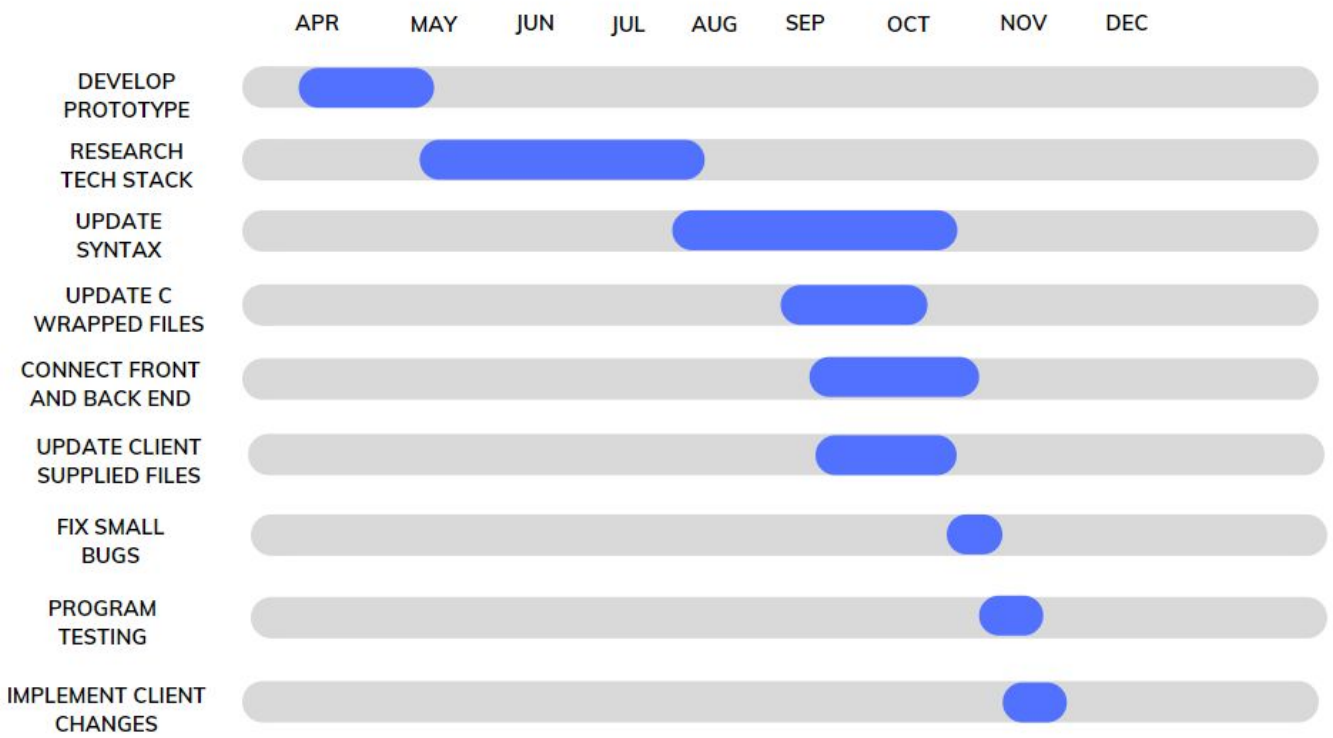
- Python3 Conversion
  - Relearning Python, LOTS of debugging, syntax errors
- Deep Program understanding
  - Extensive code reading
- Hard to understand behavior
  - A consequence of the above

# Challenges and Resolutions

## Communication and Efficiency

- Lost effort due to changing expectations/communication errors
- Install setup, original program, tables
- Testing accuracy in function and value output
    - More communication with with all points of contact at our client

# Schedule


APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
DEVELOP PROTOTYPE
RESEARCH TECH STACK
UPDATE SYNTAX
UPDATE C WRAPPED FILES
CONNECT FRONT AND BACK END
UPDATE CLIENT SUPPLIED FILES
FIX SMALL BUGS
PROGRAM TESTING
IMPLEMENT CLIENT CHANGES

# Testing Plan

- Unit Testing
  - Add star to target table
  - Calculate Calibrator
- Integration Testing
  - 77 functions across 6 classes
  - Target and Calibrator Class
- User Testing
  - Gave client our program to get active feedback

# Future Work

- Little maintenance needed
- Adoptance by the staff at the NPOI
- Current software engineer Peter Kurtz will be maintaining obsprep

# Conclusion

- Our project will save astronomers hours of work every night of observation, thereby enhancing the US Navy's astronomical research
- This project provided our team with interpersonal and technical learning opportunities
- Through updating our program from Python2 to Python3 and rewrapping our backend C files we were able to successfully deliver the product
- Program is currently being used by the NPOI team with no flaws