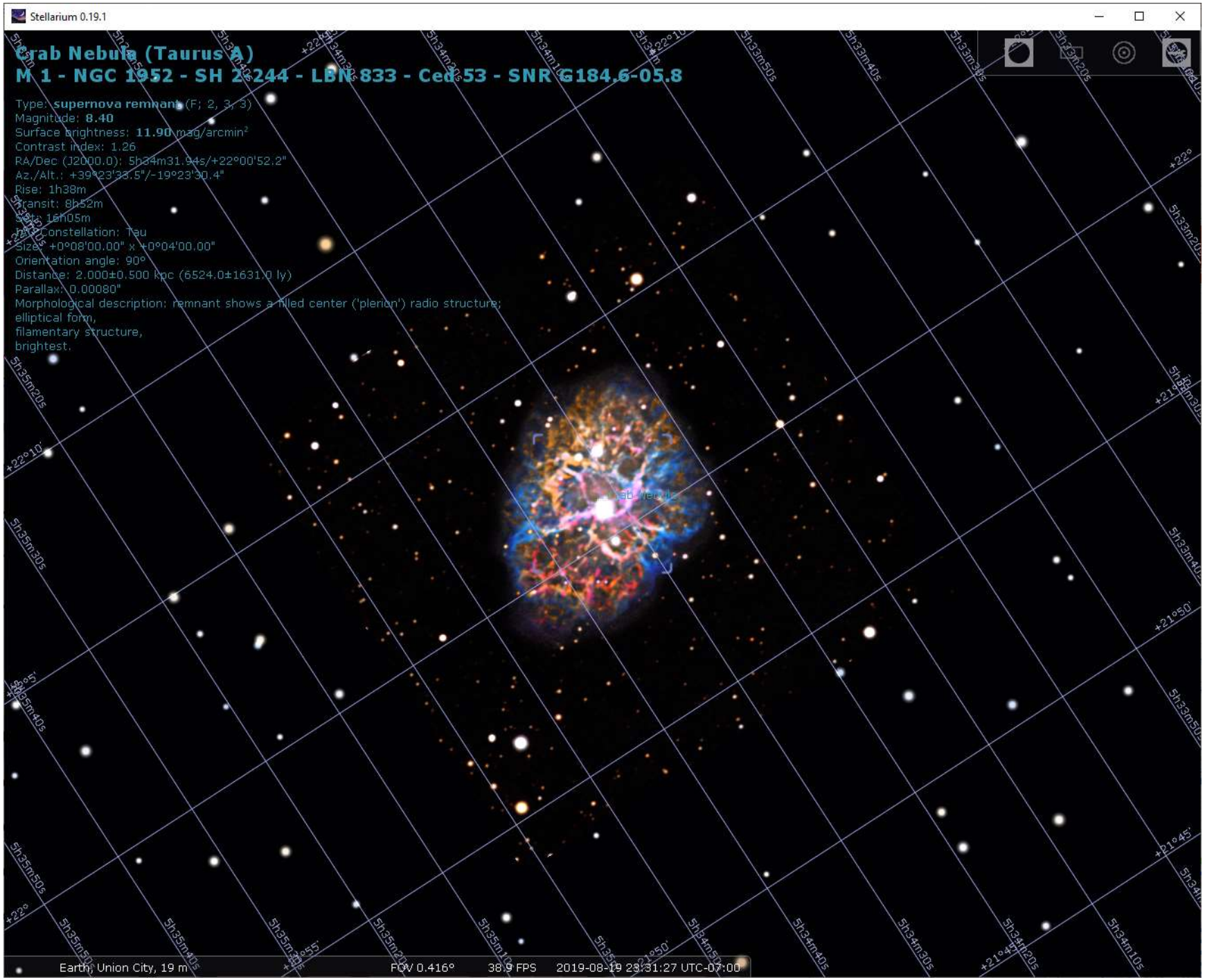
Adding Nebulae Images to Stellarium

"PERFECT" STAR ALIGNMENT EVERY TIME

GLENN NEWELL

8-20-19

HTTPS://DRIVE.GOOGLE.COM/OPEN?ID=1XB-S3YNNHNTKO02FLTAB7MSIO69ZU1WX


Stellarium 0.19.1
Crab Nebula (Taurus A)
M 1 - NGC 1952 - SH 2-244 - LBN 833 - Ced 53 - SNR G184.6-05.8
Type: supernova remnant (F; 2, 3, 3)
Magnitude: 8.40
Surface brightness: 11.90 mag/arcmin²
Contrast index: 1.26
RA/Dec (J2000.0): 5h34m31.94s/+22°00'52.2"
Az./Alt.: +39°23'33.5"/-19°23'30.4"
Rise: 1h38m
Transit: 8h52m
Set: 16h05m
IAU Constellation: Tau
Size: +0°08'00.00" x +0°04'00.00"
Orientation angle: 90°
Distance: 2.000±0.500 kpc (6524.0±1631.0 ly)
Parallax: 0.00080"
Morphological description: remnant shows a filled center ('plerion') radio structure;
elliptical form,
filamentary structure,
brightest.
Earth, Union City, 19 m
FOV 0.416°
38.9 FPS
2019-08-19 23:31:27 UTC-07:00

Why and History

- Circa 2016 I created a spread sheet for getting Nebulae images in Stellarium using data from Astrotortilla
  - Only (free) Plate Solver I knew about at the time
  - Still had to manually tweak things so stars in the image lined up with stars in Stellarium
  - Long and tedious process
- Now in 2019, I went to update the spreadsheet to use nova.astrometry.net, but still had tedious alignment
- Discovered the World Coordinate System that was made by astronomers for this very purpose
- Puts data in .fits files so every pixel can be exactly mapped on the sky despite "distortions" of Telescope, etc.
  - Created for Spitzer data
  - https://www.cs.helsinki.fi/group/goa/viewing/viewtransf/viewTrans.html
  - https://fits.gsfc.nasa.gov/fits_wcs.html
- Now images line up every time – at scales larger than 1 arcsec/pix

Goal

Check and refine these instructions for Update of chapter in official Stellarium User Guide

Before Sept. 15 for Stellarium 19.2

Stretch Goal

georg.viehoever
PTeam Member
PixInsight Jedi Master

Posts: 2129

PCL: Python Module
« **on:** 2012 December 02 09:09:08 »

This is an alpha release of a Python http://www.python.org/download/releases/2.7.3/ language extension for PixInsight. Python is a scripting language that is hugely popular in scientific computing and many other areas. It is well defined, modern and powerful. As such, I hope it is a powerful extension for PixInsight.

I am releasing this module in alpha state in the hope to find collaborators that help to advance development of this module. See section "Development" below.

Features:

- Allows to run Python 2.7 scripts within PixInsight

100% Pixinsight workflow

Plate Solve (not blind) and WCS

Python in Pixinsight

Maybe easier for end users?

Requirements

- Python 3.7 (I use Anaconda)
- Astropy (included in Anaconda)
- Astroquery (latest version, install via pip)
- My WCS_corners.py script
- My Stellarium_Nebulae_Image_Prep.py script
- API Key from your account at nova.astrometry.net
- Images plate solved at nova.astrometry.net (or pixinsight – not yet tested) so that WCS data is created
- Graphics software to flip, rotate, scale, etc. and save as .png
  - I use Photoshop
  - GIMP is a free alternative

Anaconda and AstroQuery Install

- Get the Python 3.7 64bit installer from: https://www.anaconda.com/distribution/
- Follow directions, taking defaults
- Windows
  - Open Anaconda Prompt
- Mac
  - Open Terminal
- pip install --pre –upgrade astroquery

Astrometry API Key

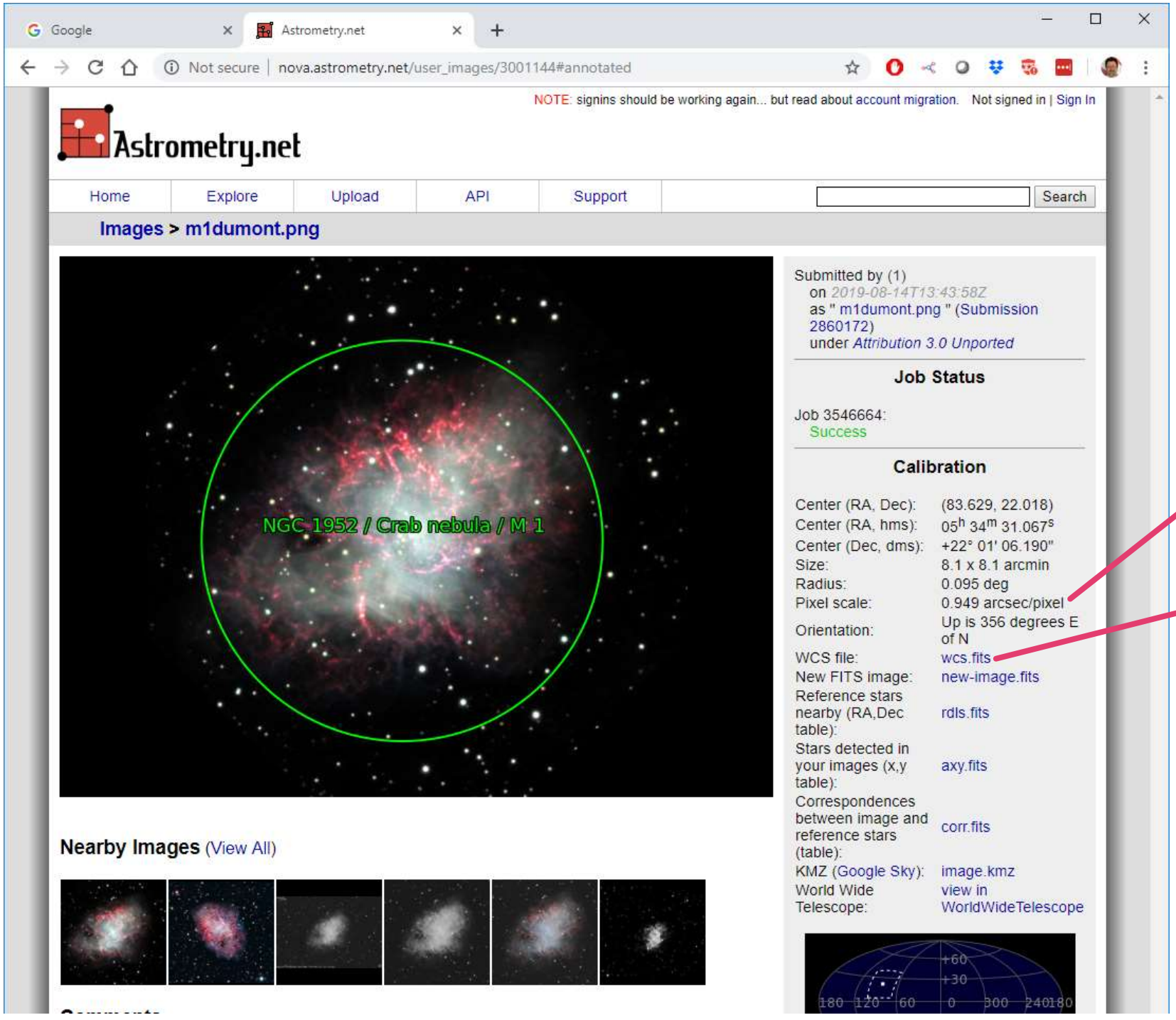
- Create an account (or log in with google, etc.) at nova.astrometry.net
  - On the API tab, copy your api key (shown in **green**)
  - Paste that key into
    - Windows
      - "%USERPROFILE%\Anaconda3\Lib\site-packages\astroquery\astrometry_net__init__.py"
      - Inside the single quotes on line 18 and save the file
    - Mac

Json Lint

- If no images show up in Stellarium, chances are you have introduced an error(s) in the textures.json file
- You can check here: https://jsonlint.com/
- Just paste the entire contents of the file in and press "Validate JSON"
  - However, the original as shipped textures.json file also fails
    - Missing leading zeros in front of decimal points
    - White space (a tab in this case) before http in an infoUrl entry

Workflow

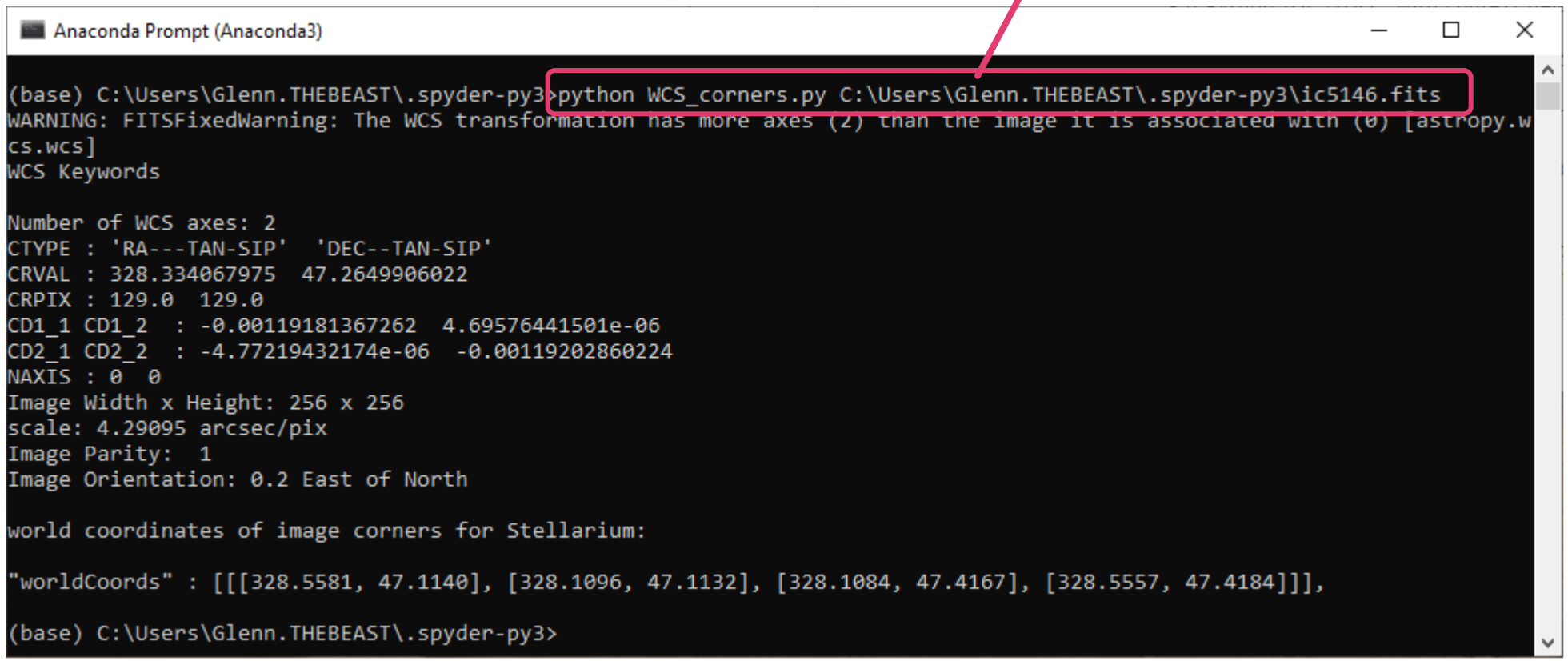
- Plate Solve your image @ nova.astrometry.net and note the "pixel scale" and "orientation".
- Download the wcs.fits file from the “Results” page and run WCS_Corners.py on it
- If Parity is -1, flip your image horizontally (do this before below rotate step)
- If needed, re-scale your image so that the pixel scale is greater than or equal to 1 arcsec/pixel and the portion you want to see in Stellarium would be between 512 and 2048 pixels in width (or height)
- Rotate your image so "up" is exactly "North". i.e. if the orientation of your image was "261 degrees East of North" then rotate your image 261 degrees CW
- Fill in the blank areas of your rotated image with black pixels, and set your “sky” to be black
- Crop your image so that both x and y dimensions are powers of two pixels, e.g. 512x512, 1024x1024, 2048x2048, 1024x2048, etc.
- Save your image as .png file
- Plate Solve your flipped, rotated, and cropped image @ nova.astrometry.net and save the wcs.fits file again
- Copy your image to the \Stellarium\nebulae\default directory
- Run the python script with the full path to the wcs.fits file as an argument
- Edit the Stellarium textures.json with the image file name and the generated world coordinates
- View your image in Stellarium (and adjust min brightness in textures.json if needed)



Pixel scale and Orientation

Download wcs.fits file

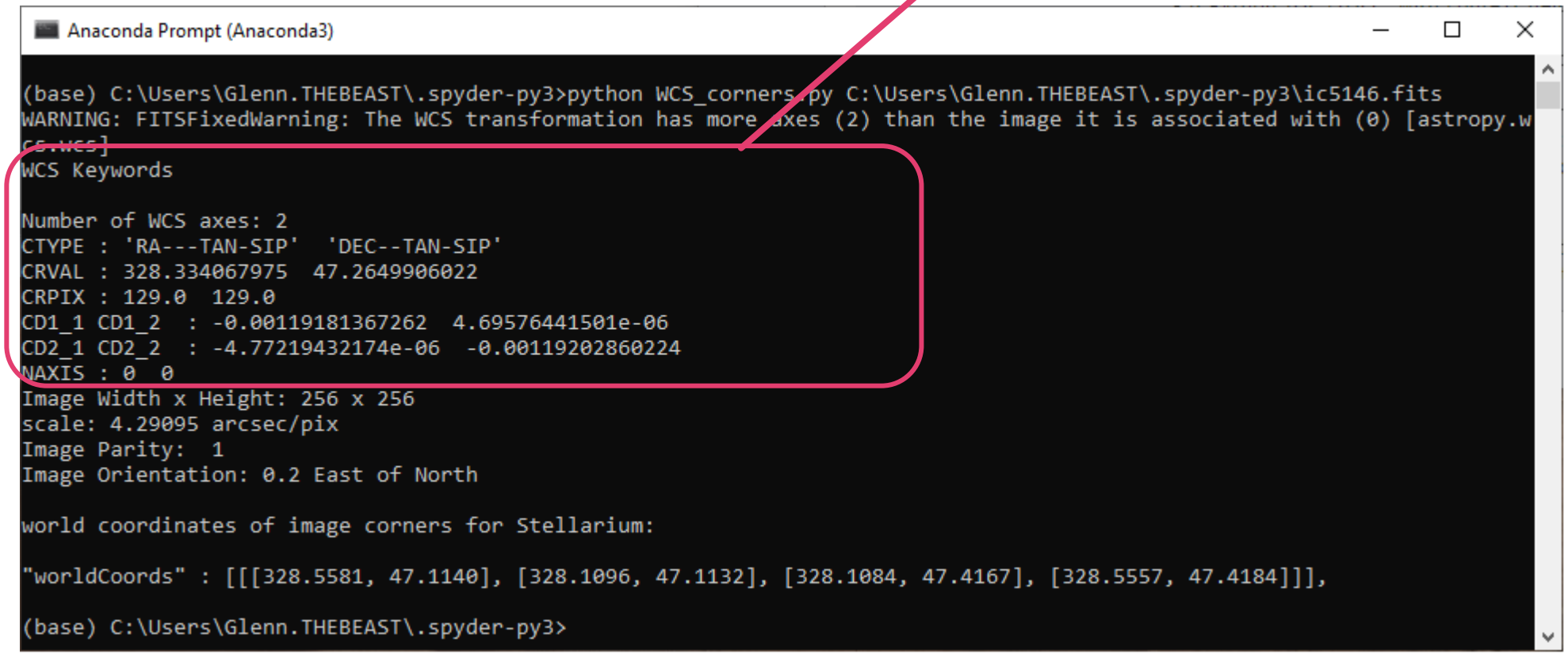
Run command

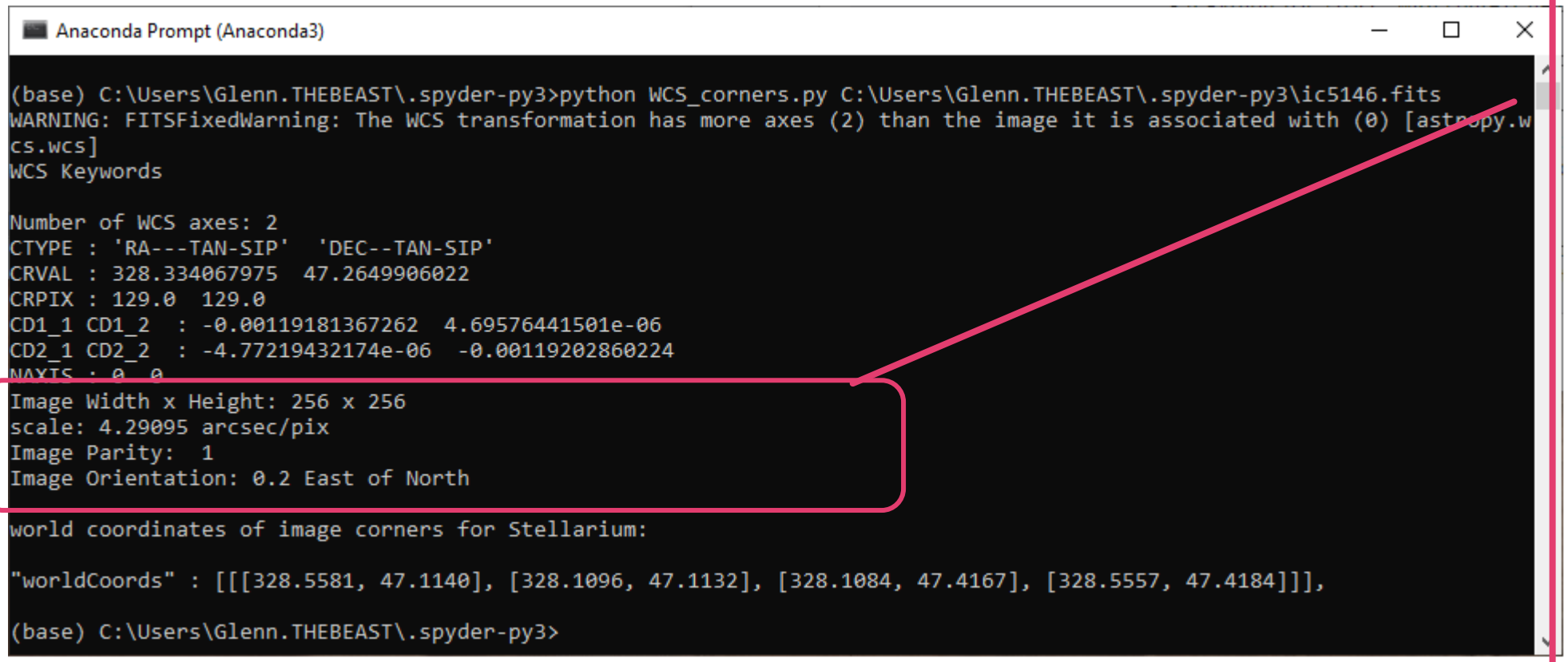


python WCS_corners.py wcs.fits

If you just leave a space after “python WCS_corners.py”
you can drag and drop your wcs.fits file (into Windows’ Anaconda Prompt window)

(Some of) The WCS info





Pixel Dimensions:

Powers of two on a side
512, 1024, or 2048

Scale:

>= 1 arcsec/pix

Parity:

If -1 flip image horizontal

Orientation:

Rotate image so North is Up
i.e. if orientation is 240 degrees East of North rotate image 240 degrees CW

```
    "maxBrightness" : 14.2
},
{
    "imageCredits" : {"short" : "Hewholooks","infoUrl": "https://commons.wikimedia.org/wiki/File:CocoonHunterWilson.jpg"},
    "imageUrl" : "ic5146.png",
    "worldCoords" : [[[328.5582,47.1141], [328.1111,47.1124], [328.1078,47.4172], [328.5582,47.4183]]],
    "textureCoords" : [[[0,0], [1,0], [1,1], [0,1]]],
    "minResolution" : 0.266023154,
    "maxBrightness" : 13.8
},
{
    "imageCredits" : {"Short":"ESO/R. Gendler and Sun Shuwei","infoUrl": "http://www.eso.org/public/images/eso1311c/, https
    "imageUrl" : "lmc_01.png",
    "worldCoords" : [[[90.3108, -66.5911], [85.4304, -66.8003], [85.084, -64.862], [89.5863,-64.6724]]],
```

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\Glenn.THEBEAST\.spyder-py3>python WCS_corners.py C:\Users\Glenn.THEBEAST\.spyder-py3\ic5146.fits
WARNING: FITSFixedWarning: The WCS transformation has more axes (2) than the image it is associated with (0) [astropy.w
cs.wcs]
WCS Keywords

Number of WCS axes: 2
CTYPE : 'RA---TAN-SIP'  'DEC--TAN-SIP'
CRVAL : 328.334067975  47.2649906022
CRPIX : 129.0  129.0
CD1_1 CD1_2  : -0.00119181367262  4.69576441501e-06
CD2_1 CD2_2  : -4.77219432174e-06  -0.00119202860224
NAXIS : 0  0
Image Width x Height: 256 x 256
scale: 4.29095 arcsec/pix
Image Parity:  1
Image Orientation: 0.2 East of North

world coordinates of image corners for Stellarium:

"worldCoords" : [[[328.5581, 47.1140], [328.1096, 47.1132], [328.1084, 47.4167], [328.5557, 47.4184]]],

(base) C:\Users\Glenn.THEBEAST\.spyder-py3>
```

Cut and Paste in textures.json

worldCoords, imageURL, textureCoords and maxBrightness are the only entries that matter

- Copy/Leave textureCoords as is
- Brightness is Stellar Magnitude
- 14 as a starting point

```
    "maxBrightness" : 14.2
},
{
    "imageCredits" : {"short" : "Hewholooks","infoUrl": "https://commons.wikimedia.org/wiki/File:CocoonHunterWilson.jpg"},
    "imageUrl" : "ic5146.png",
    "worldCoords" : [[[328.5582,47.1141], [328.1111,47.1124], [328.1078,47.4172], [328.5582,47.4183]]],
    "textureCoords" : [[[0,0], [1,0], [1,1], [0,1]]],
    "minResolution" : 0.266023154,
    "maxBrightness" : 13.8
},
{
    "imageCredits" : {"Short":"ESO/R. Gendler and Sun Shuwei","infoUrl": "http://www.eso.org/public/images/eso1311c/, https
    "imageUrl" : "lmc_01.png",
    "worldCoords" : [[[90.3108, -66.5911], [85.4304, -66.8003], [85.084, -64.862], [89.5863,-64.6724]]],
```

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\Glenn.THEBEAST\.spyder-py3>python WCS_corners.py C:\Users\Glenn.THEBEAST\.spyder-py3\ic5146.fits
WARNING: FITSFixedWarning: The WCS transformation has more axes (2) than the image it is associated with (0) [astropy.w
cs.wcs]
WCS Keywords

Number of WCS axes: 2
CTYPE : 'RA---TAN-SIP'  'DEC--TAN-SIP'
CRVAL : 328.334067975  47.2649906022
CRPIX : 129.0  129.0
CD1_1 CD1_2  : -0.00119181367262  4.69576441501e-06
CD2_1 CD2_2  : -4.77219432174e-06  -0.00119202860224
NAXIS : 0  0
Image Width x Height: 256 x 256
scale: 4.29095 arcsec/pix
Image Parity:  1
Image Orientation: 0.2 East of North

world coordinates of image corners for Stellarium:

"worldCoords" : [[[328.5581, 47.1140], [328.1096, 47.1132], [328.1084, 47.4167], [328.5557, 47.4184]]],

(base) C:\Users\Glenn.THEBEAST\.spyder-py3>
```

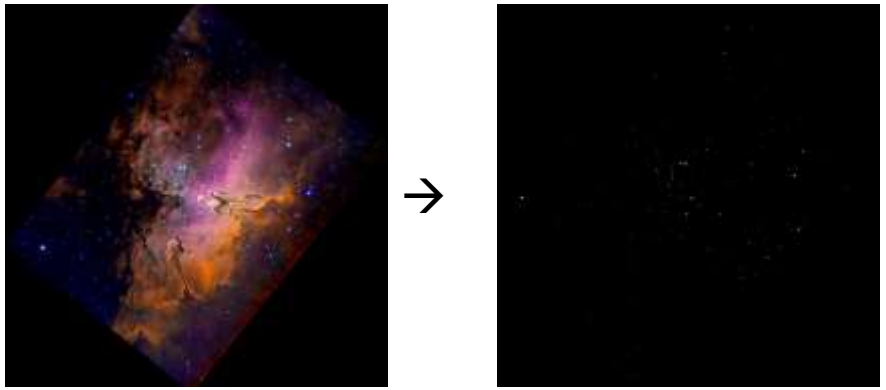
imageCredits and minResolution don't do anything

- At least as of Stellarium version 19.1

But they must be in your entry syntax wise (?)

Additional Info

- Images are NOT tied to an object, just placed on the sky by worldCoords
  - So multiple image can overlap
  - Black is transparent
  - “Transparent” areas of .png show as white
  - E.g. the bubble nebula has overlapping textures
- See the chapter on Directories, in the user guide, for where to put your own copy of the default images + yours, so yours won’t get overwritten by Stellarium Updates
- If plate solve fails
  - Adjust gamma/blackpoint so “only” stars are showing

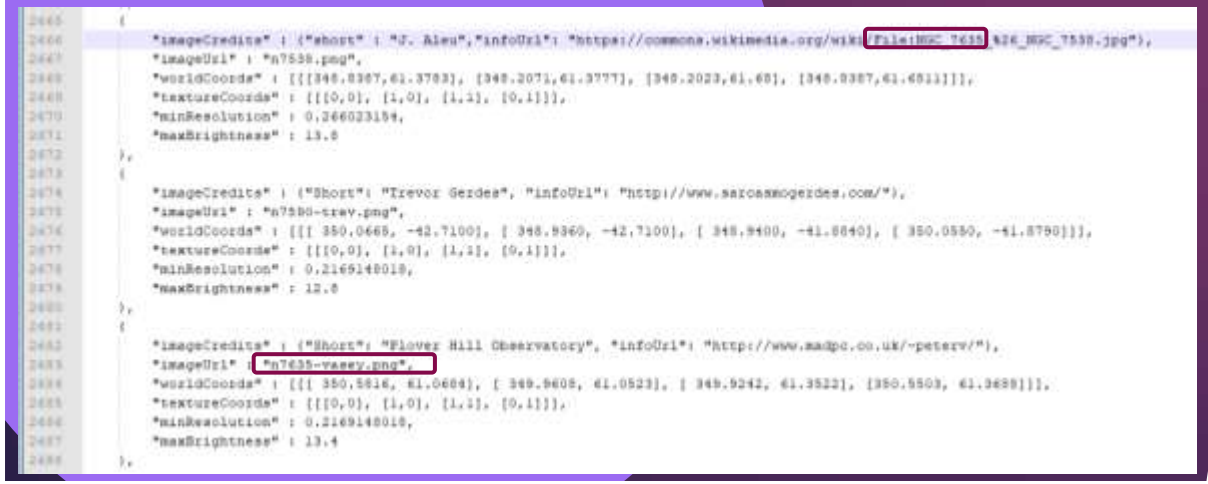


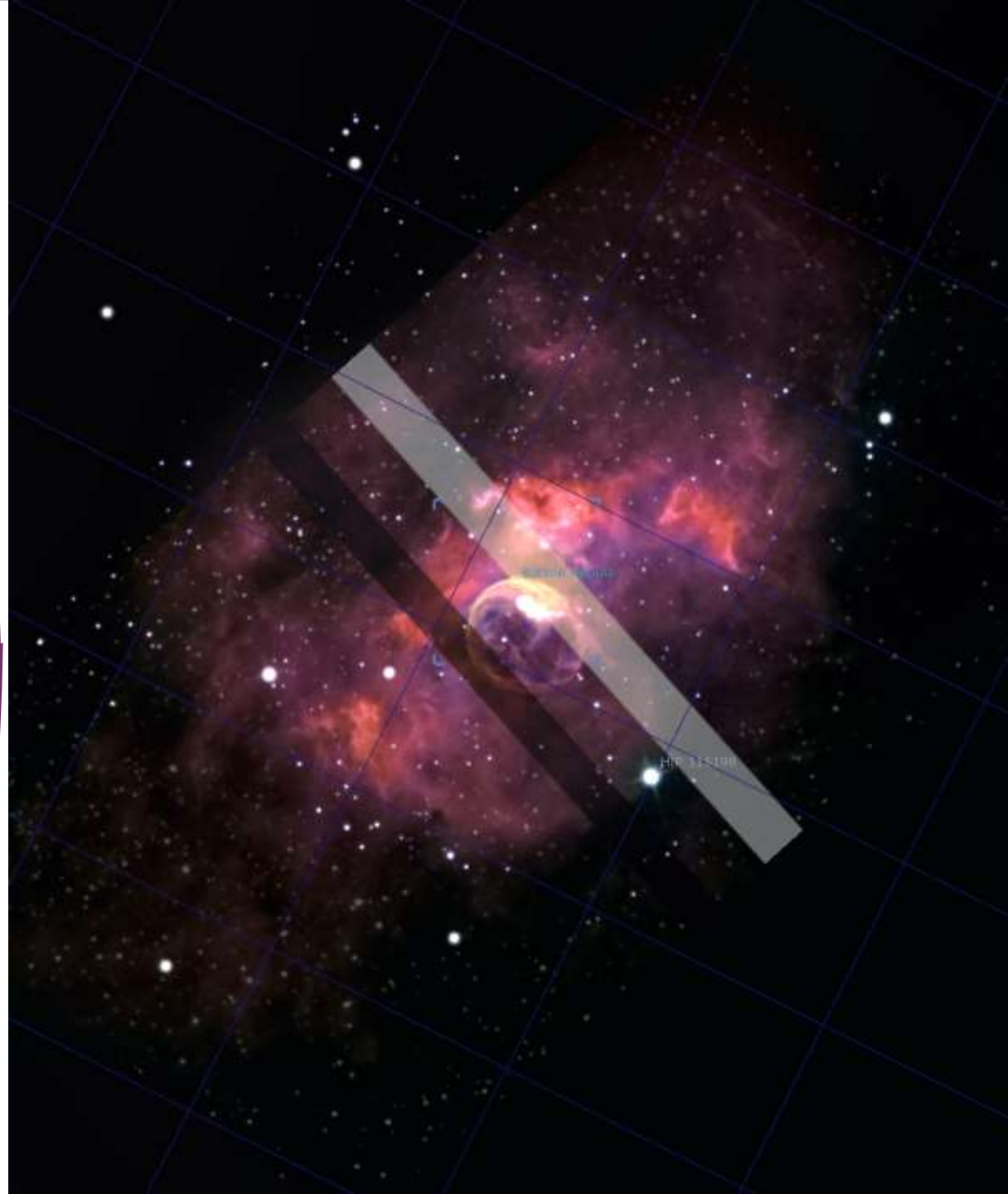
Two Bubble Images Overlaid

Black on top/first image shows image underneath

“Transparent” shows as white (or grey with brightness)

Brighter areas in both images will reinforce each other





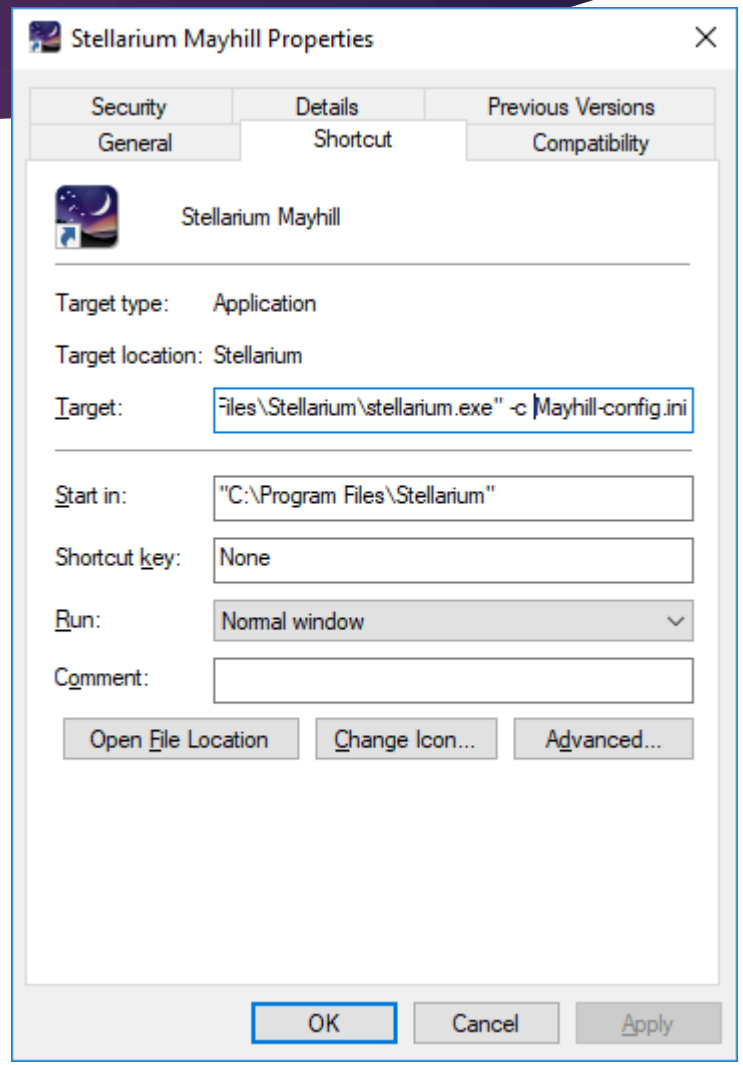
Stellarium Shortcuts

- F3 brings up object search – Tab through search results
- G to remove “ground” (in case your object is currently behind the landscape)
- A to remove “atmosphere” (in case it is daytime)
- I to toggle nebulae “images” on and off – use to test star alignment
- M to toggle milkyway on and off
- / to zoom into selected object
- \ to zoom back out

Shortcuts and custom config.ini files

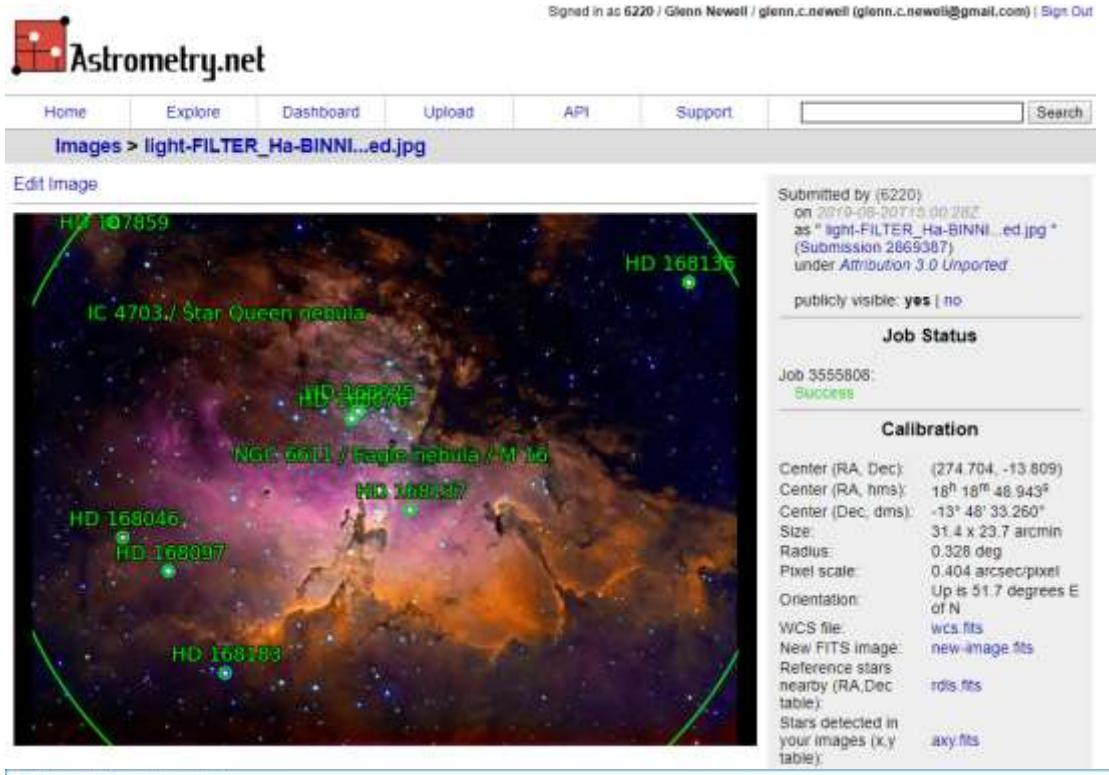
- If you are going to the view your image multiple times, you can save keystrokes
  - Make a copy or your config.ini file and rename it
  - Make a Stellarium shortcut that uses that file
  - Save your zoomed-in view of your object/image
  - View ill be saved only in that config.ini

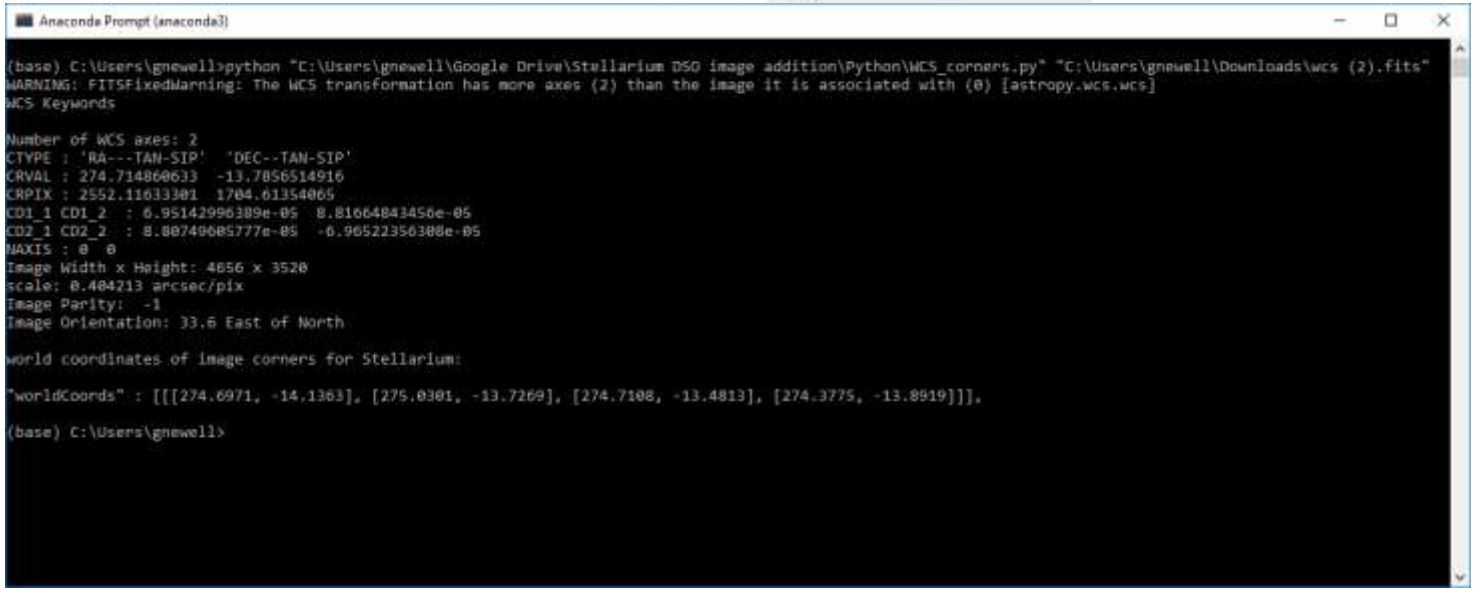
- Similar using Finder Alias on Mac (?)



Example

- Needs to be rescaled to >= 1 arcsec/pixel
- Needs to be flipped left to right
- Needs to be rotated 51.7 degrees CW
- Needs to be cropped so sides are powers of 2 pixels



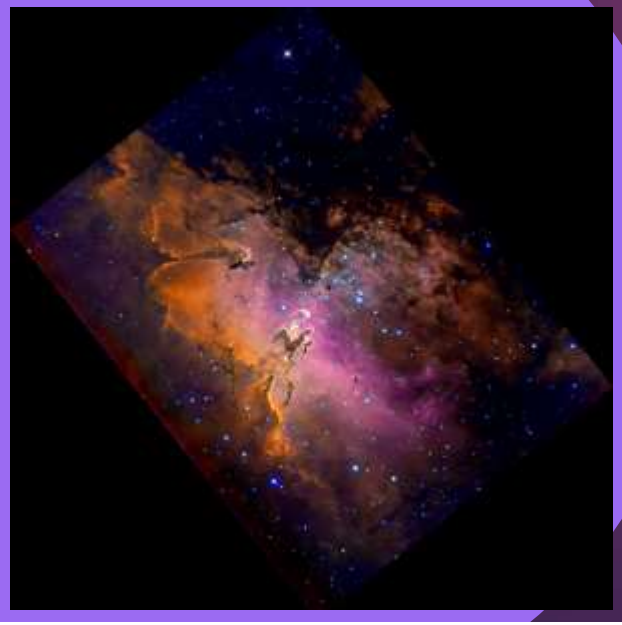


After Adjustments

Parity is correct

Image Scale is correct

Orientation is correct



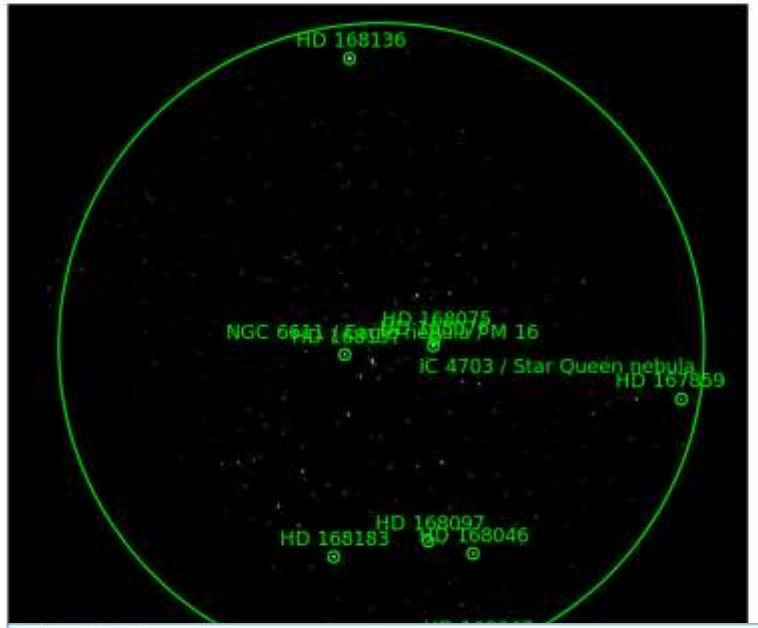
Signed in as 6220 / Glenn Newell / glenn.c.newell (glenn.c.newell@gmail.com) | Sign Out

Astrometry.net

Home | Explore | Dashboard | Upload | API | Support | Search

Images > M16_scaled_flipped_ro...rs.png

Edit Image



Submitted by (6220)
on 2019-08-20T16:22:23Z
as " M16_scaled_flipped_ro...rs.png "
(Submission 2869520)
under *Attribution 3.0 Unported*

publicly visible: **yes** | no

Job Status

Job 3555942:
Success

Calibration

Center (RA, Dec):	(274.704, -13.809)
Center (RA, hms):	$18^h\ 18^m\ 48.920^s$
Center (Dec, dms):	-13° 48' 33.768"
Size:	38.1 x 38.1 arcmin
Radius:	0.449 deg
Pixel scale:	1.12 arcsec/pixel
Orientation:	Up is 0.0139 degrees E of N
WCS file:	wcs.fits
New FITS image:	new-image.fits
Reference stars nearby (RA,Dec table):	rdls.fits
Stars detected in your images (x,y table):	axy.fits
Correspondences between image and reference stars (table):	corr.fits

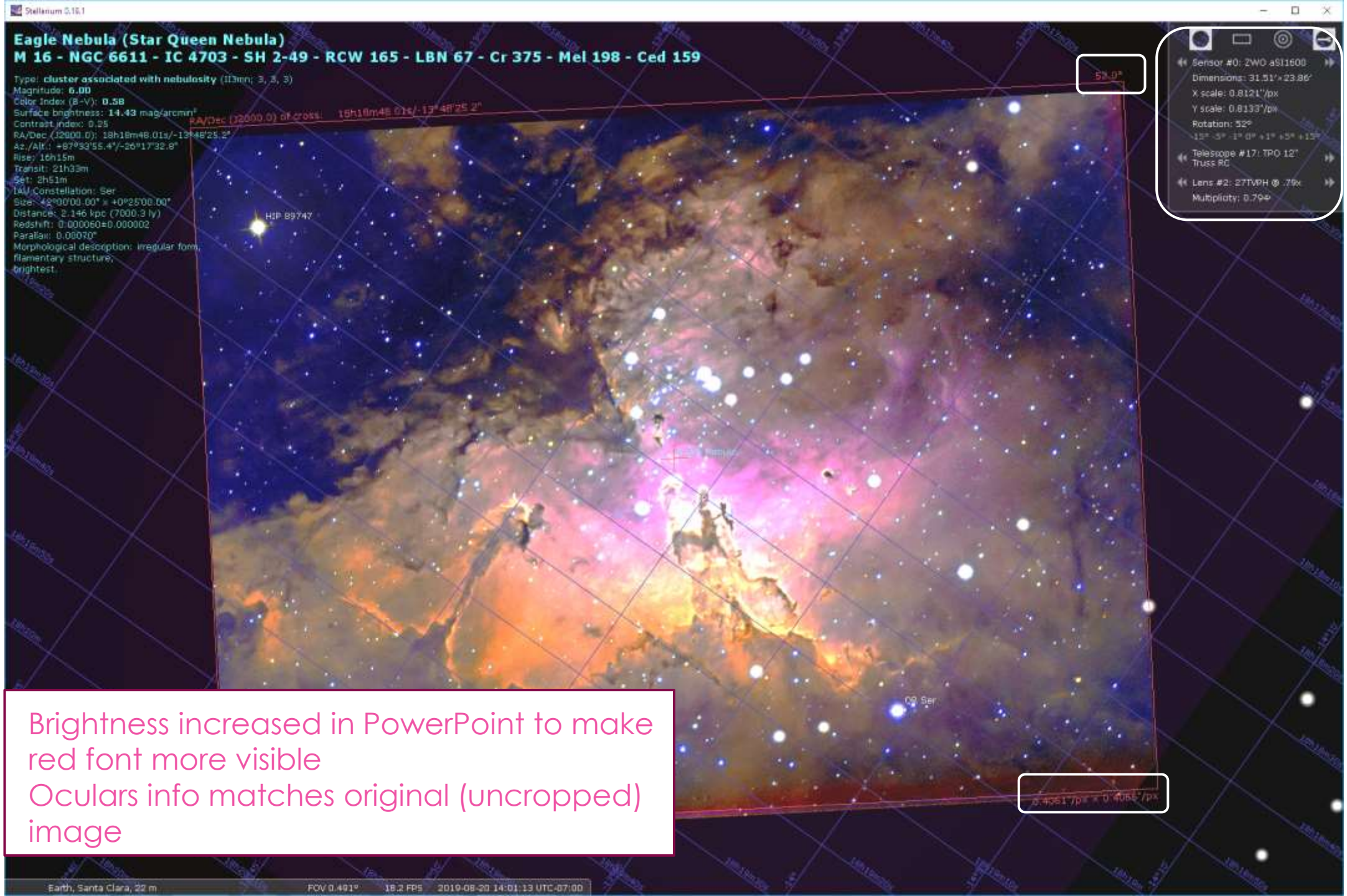
Anaconda Prompt (anaconda3)

```
(base) C:\Users\gnewell>python "C:\Users\gnewell\Google Drive\Stellarium OSD image addition\Python\WCS_corners.py" "C:\Users\gnewell\Downloads\wcs (7).fits"
WARNING: FITSFixedWarning: The WCS transformation has more axes (2) than the image it is associated with (0) [astropy.wcs.wcs]
WCS Keywords

Number of WCS axes: 2
CTYPE : 'RA---TAN-SIP'  'DEC--TAN-SIP'
CRVAL : 274.692951191  -13.7819856898
CRPIX : 1058.58631897  936.118305206
CD1_1 CD1_2  : -0.000309567334484  1.59490321889e-07
CD2_1 CD2_2  : 8.76619776374e-09  -0.000309954019315
NAXIS : 0  0
Image Width x Height: 2048 x 2048
Scale: 1.11514 arcsec/pix
Image Parity:  1
Image Orientation: 0.0 East of North

world coordinates of image corners for Stellarium:

"worldCoords" : [[[275.0320, -14.1256], [274.3759, -14.1272], [274.3789, -13.4913], [275.0284, -13.4926]]],

(base) C:\Users\gnewell>
```

Results

- Stars Match
  - "Blink with "I" key
- Matched Ocular FOV


```
{
    "imageCredits" : {"short" : "Glenn Newell"},
    "imageUrl" : "M16_scaled_flipped_rotated_cropped.png",
    "worldCoords" : [[[275.0320, -14.1256], [274.3759, -14.1272], [274.3789, -13.4913], [275.0284, -13.4926]]],
    "textureCoords" : [[[0,0], [1,0], [1,1], [0,1]]],
    "minResolution" : 0.266023154,
    "maxBrightness" : 13.6
},
```



Brightness increased in PowerPoint to make red font more visible

Oculars info matches original (uncropped) image

WCS_corners_hca.py | WCS_example.py | client2.py | filesindir.py | WCS_corners.py

```python
# -*- coding: utf-8 -*-
from astropy.io import fits
import astropy.wcs as wcs
import sys
import re
import math

filename = sys.argv[-1]
#filename = 'D:\Glenn\Downloads\wcs (2).fits'
hdulist = fits.open(filename)
w = wcs.WCS(hdulist[(0)].header, hdulist)
hdu = hdulist[0]
hdr = hdulist[0].header
hdulist.close()

print(repr(w))

print('Image Width x Height:',hdu.header['IMAGEW'],'x', hdu.header['IMAGEH'])
for comment in hdr['comment']:
    #print(comment)
    if re.match('^scale:',comment):
            print(comment)

cd11 = hdu.header['CD1_1']
cd12 = hdu.header['CD1_2']
cd21 = hdu.header['CD2_1']
cd22 = hdu.header['CD2_2']
if cd11 * cd22 - cd12 * cd21 > 0:
    parity = 1
else:
    parity = -1

print('Image Parity: ',parity)
orientation = math.degrees(math.atan2(cd21-cd12, cd11+cd22))
print('Image Orientation: {0:.1f}'.format(180 + orientation),'East of North')

wx1, wy1 = w.all_pix2world(0., 0., 0)
#print('0,0: {0} {1}'.format(wx1, wy1))

wx2, wy2= w.all_pix2world(hdu.header['IMAGEW']-1,0, 0)
#print(hdu.header['IMAGEW']-1,0,': {0} {1}'.format(wx2, wy2))


wx3, wy3 = w.all_pix2world(hdu.header['IMAGEW']-1, hdu.header['IMAGEH']-1, 0)
#print(hdu.header['IMAGEW']-1, hdu.header['IMAGEH']-1,': {0} {1}'.format(wx3, wy3))

wx4, wy4 = w.all_pix2world(0., hdu.header['IMAGEH']-1, 0)
#print(0, hdu.header['IMAGEH']-1,': {0} {1}'.format(wx4, wy4))
print('\nworld coordinates of image corners for Stellarium:')
print('\n"worldCoords" : [[[{0:.4f}, {1:.4f}], [{2:.4f}, {3:.4f}], [{4:.4f}, {5:.4f}], [{6:.4f}, {7:.4f}]]],'.format(wx4, wy4, wx3, wy3, wx2, wy2, wx1, wy1))
```

Not much to the Python
Thanks to Astropy

Thank You

HTTPS://DRIVE.GOOGLE.COM/OPEN?ID=1XB-S3YNNHNTKO02FLTAB7MSIO69ZU1WX