

Moonn

August 12, 2022

1 Drawing the moon movement in Egypt in Jan 2022 with coordinate system Altitude and Azimuth as a part of our project

1.0.1 We are using data from U.S. Naval Observatory

<https://www.cnmoc.usff.navy.mil/usno/>

```
[1]: !pip install Pillow==9.0.0
      !pip install pandas
      !pip install tk
```

```
Requirement already satisfied: Pillow==9.0.0 in g:\new folder
(4)\anconda\lib\site-packages (9.0.0)
Requirement already satisfied: pandas in g:\new folder (4)\anconda\lib\site-
packages (1.3.4)
Requirement already satisfied: numpy>=1.17.3 in g:\new folder
(4)\anconda\lib\site-packages (from pandas) (1.20.3)
Requirement already satisfied: pytz>=2017.3 in g:\new folder
(4)\anconda\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: python-dateutil>=2.7.3 in g:\new folder
(4)\anconda\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in g:\new folder (4)\anconda\lib\site-
packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
Requirement already satisfied: tk in g:\new folder (4)\anconda\lib\site-packages
(0.1.0)
```

```
[23]: from tkinter import *
      import time
      from PIL import ImageTk,Image, ImageEnhance,ImageGrab
      import pandas as pd
      import numpy as np
```

Some important Constants

```
[24]: win_WIDTH = 1900
      win_HEIGHT = 500
      vision_range_alt = 90
      vision_range_az = 360

      preview_postion_x = 100
```

```

preview_postion_y = 125

bar_x_place = 1600
bar_y_place = 220
c_height = 100
y_stretch = 2 # The highest y = max_data_value * y_stretch
y_gap = 20 # The gap between lower canvas edge and x axis
x_stretch = 20 # Stretch x wide enough to fit the variables
x_width = 40 # The width of the x-axis
x_gap = 40 # The gap between left canvas edge and y axis

```

initiating the window, the background, grid, texts...

```

[25]: try:
        window = Tk()
    except:
        window = Toplevel()

window.resizable(False, False)
window.geometry('%dx%d+%d+%d' % (win_WIDTH, win_HEIGHT, 0, 0))
window.title('Moon movement Jan 2022')

canvas = Canvas(window, width=win_WIDTH, height=win_HEIGHT)
background_photo = PhotoImage(file='sky.png')
background = canvas.create_image(0, 0, image=background_photo, anchor=NW)

for i in range(5):
    canvas.create_line(0, i * 100, win_WIDTH, i * 100, fill="grey", width=1,
        ↪dash=(2, 1))
    canvas.create_text(35, i * 100 - 10, fill="white", font=('Helvetica', '9'),
        text=f"Altitude: {(5-i)*20}")

for i in range(11):
    canvas.create_line(i * 190, 0, i * 190, win_WIDTH, fill="grey", width=1,
        ↪dash=(2, 1))
    canvas.create_text(i * 190, 10, fill="white", font=('Helvetica', '9'),
        text=f"Azimuth: {(i)*36}")

alt = canvas.create_text(bar_x_place+120, bar_y_place
    ↪+90, fill="white", font=('Helvetica', '8', 'bold'), text=f"Altitude")
az = canvas.create_text(bar_x_place+60, bar_y_place
    ↪+90, fill="white", font=('Helvetica', '8', 'bold'), text=f"Azimuth")
vel = canvas.create_text(bar_x_place+90, bar_y_place
    ↪-130, fill="white", font=('Helvetica', '12', 'bold'), text=f"Velocities")

```

```

date = canvas.create_text(70, win_HEIGHT - 50, fill="white", font=('Helvetica', 18, 'bold'), text=f"Day: 1\nMonth: 1\nYear: 2022")
cords = canvas.create_text(win_WIDTH - 50, win_HEIGHT - 25, fill="white", font=('Helvetica', 12, 'bold'), text=f"Altitude: 0\nAzimuth: 0")

speed_x= canvas.create_rectangle(1000, 200, 120, 220, fill="red")
speed_y= canvas.create_rectangle(1000, 100, 220, 120, fill="blue")

canvas.pack()

```

Creating Moon image and phases preview image with suitable sizes

```

[26]: img = (Image.open("image_moon.png"))
moon_size_x = int(win_WIDTH / vision_range_az * 10)
moon_size_y = int(win_WIDTH / vision_range_az * 10)

moon_preview_size_x = 250
moon_preview_size_y = 250

resized_image = img.resize((moon_size_x,moon_size_y),resample=Image.BICUBIC)

resized_moon_preview = img.
    ↳resize((moon_preview_size_x,moon_preview_size_y),resample=Image.BICUBIC)

```

we create a dark version and light version to show the Moon phases

```

[27]: enhancer = ImageEnhance.Brightness(resized_image)
Bright = enhancer.enhance(1.5)
Dark = enhancer.enhance(0.4)

Dark_p = ImageTk.PhotoImage(Dark)
Bright_p = ImageTk.PhotoImage(Bright)

my_image = canvas.create_image(0, 0, image=Dark_p, anchor=NW)
my_image2 = canvas.create_image(0, 0, image=Bright_p, anchor=NW)

enhancer2 = ImageEnhance.Brightness(resized_moon_preview)
Bright_preview = enhancer2.enhance(1)
Dark_preview = enhancer2.enhance(0.2)

Dark_p_preview = ImageTk.PhotoImage(Dark_preview)
Bright_p_preview = ImageTk.PhotoImage(Bright_preview)

moon_D = canvas.create_image(preview_postion_x, preview_postion_y,
    ↳image=Dark_p_preview, anchor=NW)

```

```
moon_B = canvas.create_image(20, 150, image=Bright_p_preview, anchor=NW)
```

Getting the data ready

```
[28]: obs = pd.read_csv('observationsCopy.csv')
obs_cor = obs[['Alt', 'Az']]

#data mapping functions
def mping_alt(n):
    return win_HEIGHT - (n * win_HEIGHT / vision_range_alt)

def mping_az(n):
    return n * win_WIDTH / vision_range_az

cor_alt = list(map(mping_alt, obs_cor['Alt']))
cor_az = list(map(mping_az, obs_cor['Az']))
```

1.0.2 Drawing velocity functions

```
[29]: def draw_vel(velocity):
        for x, y in enumerate(velocity):

            # Bottom left coordinate
            y = abs(y)
            x0 = x * x_stretch + x * x_width + x_gap + bar_x_place

            # Top left coordinates
            y0 = bar_y_place + c_height - (y * y_stretch + y_gap)

            # Bottom right coordinates
            x1 = x * x_stretch + x * x_width + x_width + x_gap + bar_x_place

            # Top right coordinates
            y1 = bar_y_place + c_height - y_gap

            speed = y
            if y == velocity[0]:
                if y > 40:
                    canvas.itemconfig(az, fill="red", font=('Helvetica', '8',
↪'bold'), text=f"Azimuth")
                    time.sleep(0.05)
                    if y > 100:
                        continue
                else:
```

```

        canvas.itemconfig(az, fill="white", font=('Helvetica', '8', ↵
↵'bold'), text=f"Azimuth")
        canvas.coords(speed_y, x0, y0, x1, y1)

    else:
        if y > 100:
            canvas.itemconfig(alt, fill="red", font=('Helvetica', '8', ↵
↵'bold'), text=f"Altitude")
            time.sleep(0.05)
            if y > 100:
                continue
            else:
                canvas.itemconfig(alt, fill="white", font=('Helvetica', ↵
↵'8', 'bold'), text=f"Altitude")

        canvas.coords(speed_x, x0, y0, x1, y1)

```

1.1 Main loop

```

[30]: wait_time = 0.01
graph = 1
i = 0
day = obs.loc[0, 'day']
day_aa = []
while i < len(obs[obs['month'] == 1]):
    velocity = []
    cropped_image_br = Bright.crop([0, 0, moon_size_x * (obs.loc[i, 'Visible']), ↵
↵moon_size_y])
    cropped_preview_image_br = Bright_preview.crop([0, 0, moon_preview_size_x * ↵
↵(obs.loc[i, 'Visible']), moon_preview_size_y])

    Bright_p = ImageTk.PhotoImage(cropped_image_br)
    Bright_p_preview = ImageTk.PhotoImage(cropped_preview_image_br)

    my_image2 = canvas.create_image(0, 0, image=Bright_p, anchor=NW)
    moon_B = canvas.create_image(preview_postion_x, preview_postion_y, ↵
↵image=Bright_p_preview, anchor=NW)
    canvas.pack()
    coordinates = canvas.coords(my_image)
    coordinates2 = canvas.coords(my_image2)
    # print(coordinates)
    velocity.append(cor_az[i] - coordinates[0])
    velocity.append(cor_alt[i] - coordinates[1])
    canvas.move(my_image, velocity[0], velocity[1] )

```

```

        canvas.move(my_image2, cor_az[i] - coordinates2[0], cor_alt[i] -
↪coordinates2[1])

        draw_vel(velocity)

        window.update()
        time.sleep(wait_time)
        canvas.itemconfig(date, text=f"Day: {obs.loc[i, 'day']}\nMonth: {obs.
↪loc[i, 'month']}\nYear: 2022")
        if day != obs.loc[i, 'day']:
            this_day = np.where(obs["day_counter"] == graph)[-1]

            for line in day_aa:
                canvas.create_line(line, fill="green", width=3)
            graph+=1
            day_aa = []
            canvas.itemconfig(cords, text=f"Altitude: {obs.loc[i, 'Alt']}\nAzimuth: {obs.
↪loc[i, 'Az']}")

            if i % 2 == 0:
                day_aa.append((cor_az[i]+ moon_size_x/2, cor_alt[i]+moon_size_y/
↪2, cor_az[i+1]+moon_size_x/2, cor_alt[i+1]+
                    moon_size_y/2))

            # day = obs.loc[i, 'day']
            # ImageGrab.grab().crop((0, 0, 1910, 530)).save(f"images3/img{str(i)}.
↪zfill(10)}.png")
            i += 1

window.mainloop()

```