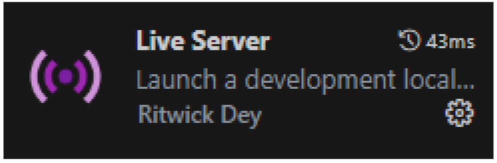
# Animate a Point with a Camera

Zhaoxu Sui
This instruction refers to only the newest version of the code (060623)
Color Code: Topic, File, Description, future direction for improvement

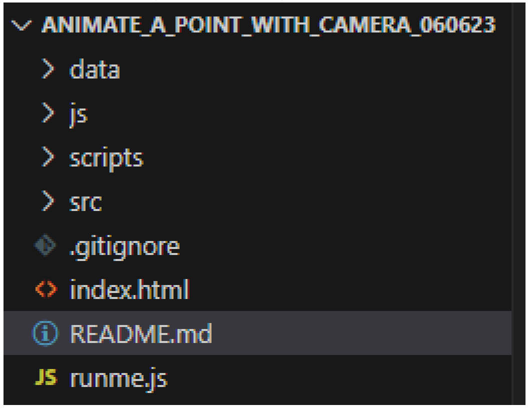
## Environment

Please open and edit it with Visual Studio Code, and don't forget to add the extension of Live Server,



Live Server
43ms
Launch a development local...
Ritwick Dey

Then, use Explorer to open the whole folder, and enable the Go Live option on the bottom-right bar.



```
ANIMATE_A_POINT_WITH_CAMERA_060623
  > data
  > js
  > scripts
  > src
    .gitignore
    index.html
    README.md
    runme.js
```

So, congrats! Now you host this file in your local server. It is fine if you want to use it in other code editor with live editing option, just feel free to use that to avoid any CORS problem.

## html Parameter Query

Default URL with query Params:
http://127.0.0.1:5500/?gender=female&stage=1&prod=false&square=false
'gender' and 'stage' are used to locate which deer we are specifically looking at. In this case, our default data is female and stage 1. I will change the naming conventions in future, and it is also easy to change.
If 'prod' is true, the duration of the whole animation will be longer, and the speed will be slower. When the animation ends, there will be a downloaded file ready to view, but it is broken (as 6/6/23).
If 'square 'is true, the map will be in a fixed dimension of 1080px * 1080px.

## Basic Structure

Blue Arrows refers to where the function gets called and where it is stated.

Index.html

Title, head, style, and body.

In body, js/d3-ease.min.js is called first, and js/script.js is called next.

js/d3-ease.min.js

Refer to GitHub - d3/d3-ease: Easing functions for smooth animation. For more detail.

js/script.js

Catch parameters from URL query parameters.

If Square = true, the web map will be in a square size.

Initialize the Base Map and the Inset Map.

Set Inset Map bounds and add the source and layer for the yellow point on the inset map.

Function: UpdateInsetMap declared here.

Map on "Load"

Call Add 3D, sky, and Fog.

Encoder, Call Play Animation, record

Const: Add 3D, sky, and Fog.

Const: Play Animation, the following will go in sequence.

Call AddPathSourceAndLayer.

Get the Start of line string.

Call FlyInAndRotate: js/fly-in-and-rotate.js.

Call animatePath:

js/animate-path.js

Calculate distance along the path, bearing

Call computeCameraPosition:

js/util.js

ComputeCameraPosition

Set Camera Options

Call UpdateInsetMap

Add the progress bar and measure the dates.

Standardize naming practices to grab the snow data.

Call add_Snow_Source and add_Snow_Layer.

Window.requestAnimationFrame(frame)

Window.requestAnimationFrame(frame)

Fit to the bounds of the map.

setTimeout

Const addPathSourceAndLayer.

Add Pins, Call createGeoJSONCircle:

js/util.js

createGeoJSONCircle

Function add_Snow_Source and add_Snow_Layer.

## How to make the Migration Lines Dataset?

### PREPARING MIGRATION LINES DATASET

1. Format should be shapefile
2. Check on the point data and examine whether the interval of every points taken is consistent
3. Go to line data, and which should be represented as a single polyline before exporting.

### CONVERTING TO JSON

3. Use "Features To JSON" and export it into the default folder within /GIS/Convert_to_GeoJSON folder
4. Leave all settings unchecked except the "Output to GeoJSON"
5. Cut and paste the JSON file to data/Camera_Line
6. Rename it to the following format: Camera_target_SEASONYEAR, e.g. Camera_target_Fall2019

### IMPORT SETTING

11. Open working folder
12. Under" /data", paste the GeoJSON file
13. Open the JSON fille, under the "properties", add the content following the practice below  
 {"start_date": "20XX-XX-XXT00:00:00Z", "end_date": "20XX-XX-XXT00:00:00Z"},  
 Matching the actual start time and end time of the dataset
14. Rename it to match the naming practices.

## How to make the Snow Layer Data?

### PREPROCESSING

1. Go to https://www.nohrsc.noaa.gov/archived_data/, Access Data by FTP
2. under" …\Script_based\data\NOAA_masked_2019",
3. keep the naming number_monthabbr (e.g., 10_Oct) to create a new folder
4. Choose Masked Data 2019, add download the whole month of data in this new folder
5. Unzip it and only select those dates with the name beginning with "us_ssmv11036tS"
6. Unzip those again. Feel free to keep the original zipped files in separate folders like I did.

### ARC NOTEBOOK

7. Under " …\Script_based\data\Notebook", duplicate any file and give it a new name
8. Open the notebook in ArcGIS Pro,
9. and change all the relevant code Lines, corresponds to the current month (e.g., 10_Oct)
10. Be careful about modifying the number of days in that month (28, 29, 30, or 31?)
11. Double check on the paths and naming conventions
12. Make a new folder under " …\Script_based\data\NOAA_masked_2019" as "Exported_GIF_X"
13. Run!

### UPLOAD DATA

14. Look at the "Exported_GIF_X" folder, and delete irrelevant files, only imageries left here.
15. Upload it as a folder to GitHub repository under "WMI/Animation/NOAA_masked_2019/"