The major difference:

1, having trajectory viewing ability

2, using LLpos rather than XYpos.

The output waypoint is in long lat

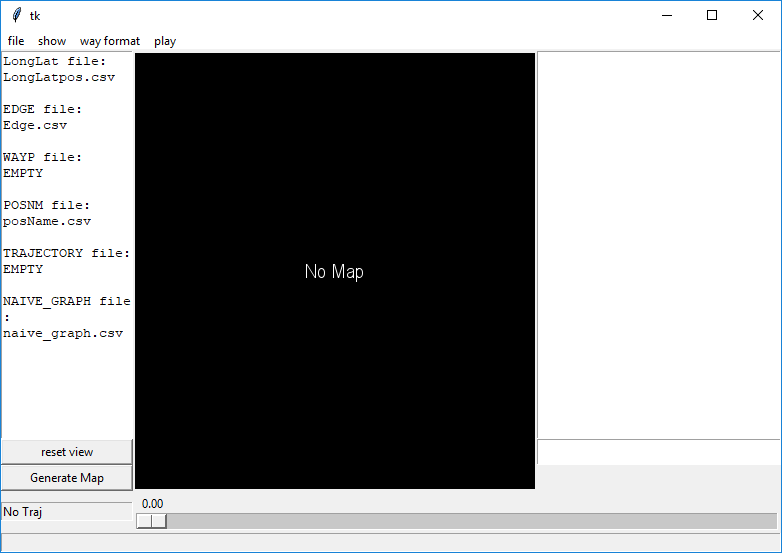
3, Plane.png will be use for display. The center of the image will be at the Long lat in trajectory. The height of the demo image is considered 70(hard coded). So if you want smaller plane use a smaller sized image!

Tutorial of the airport viewer panel

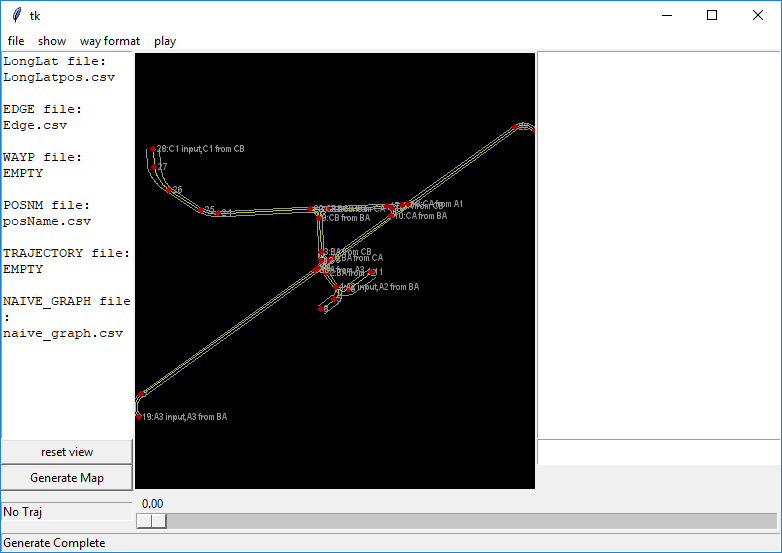
# General description

This software is aimed at generating waypoints from a high level aircraft Taxing plan, as well as visualizing the motion of an aircraft on a 2D map.

First, run **main.py** using python 3.6.2.



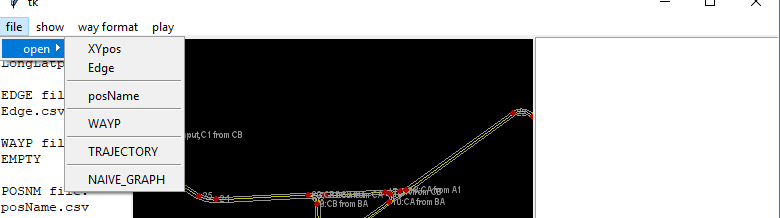
On opening the software panel. Click **Generate Map**



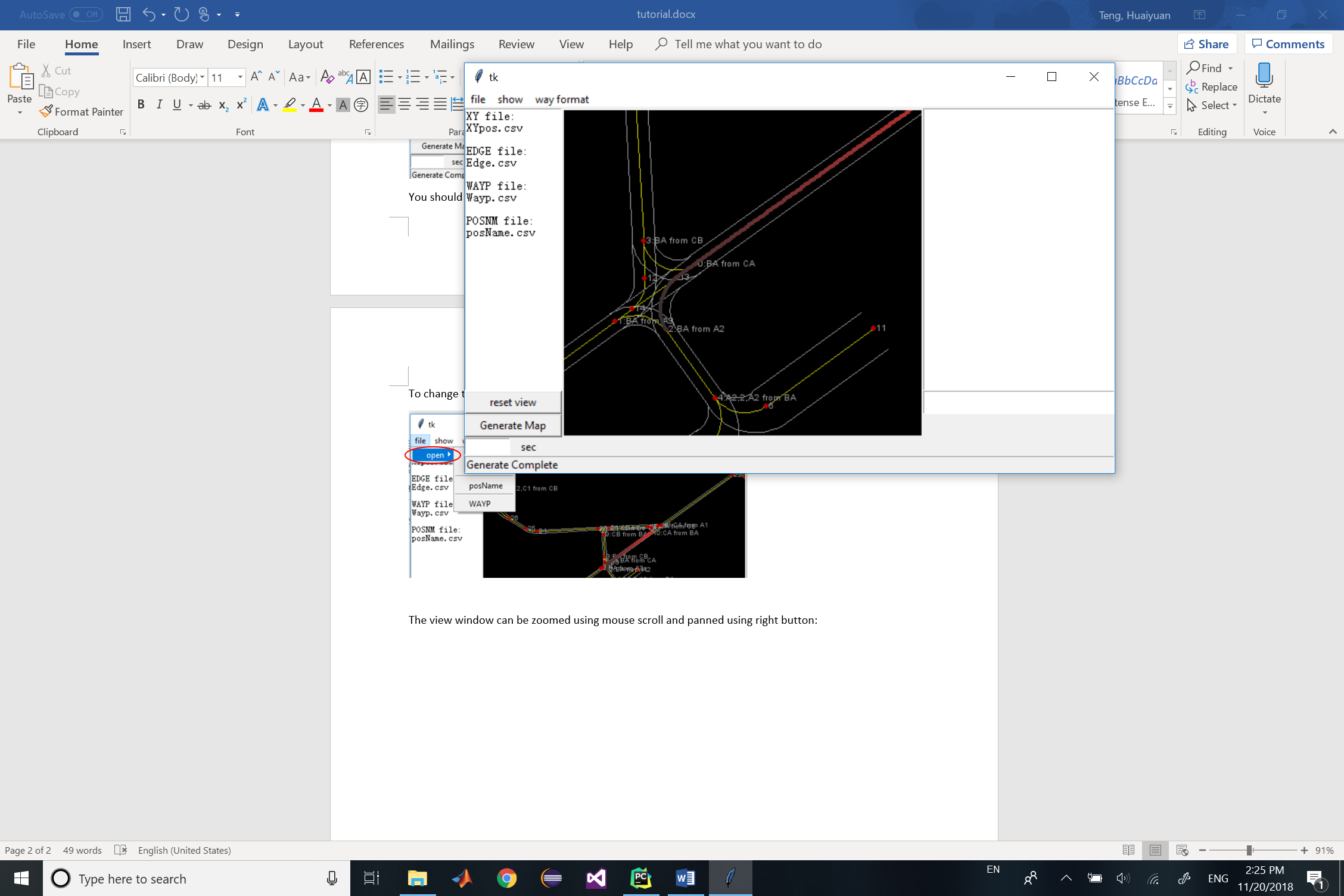
You should see a map has been created from the files listed on the left.

# Menu:

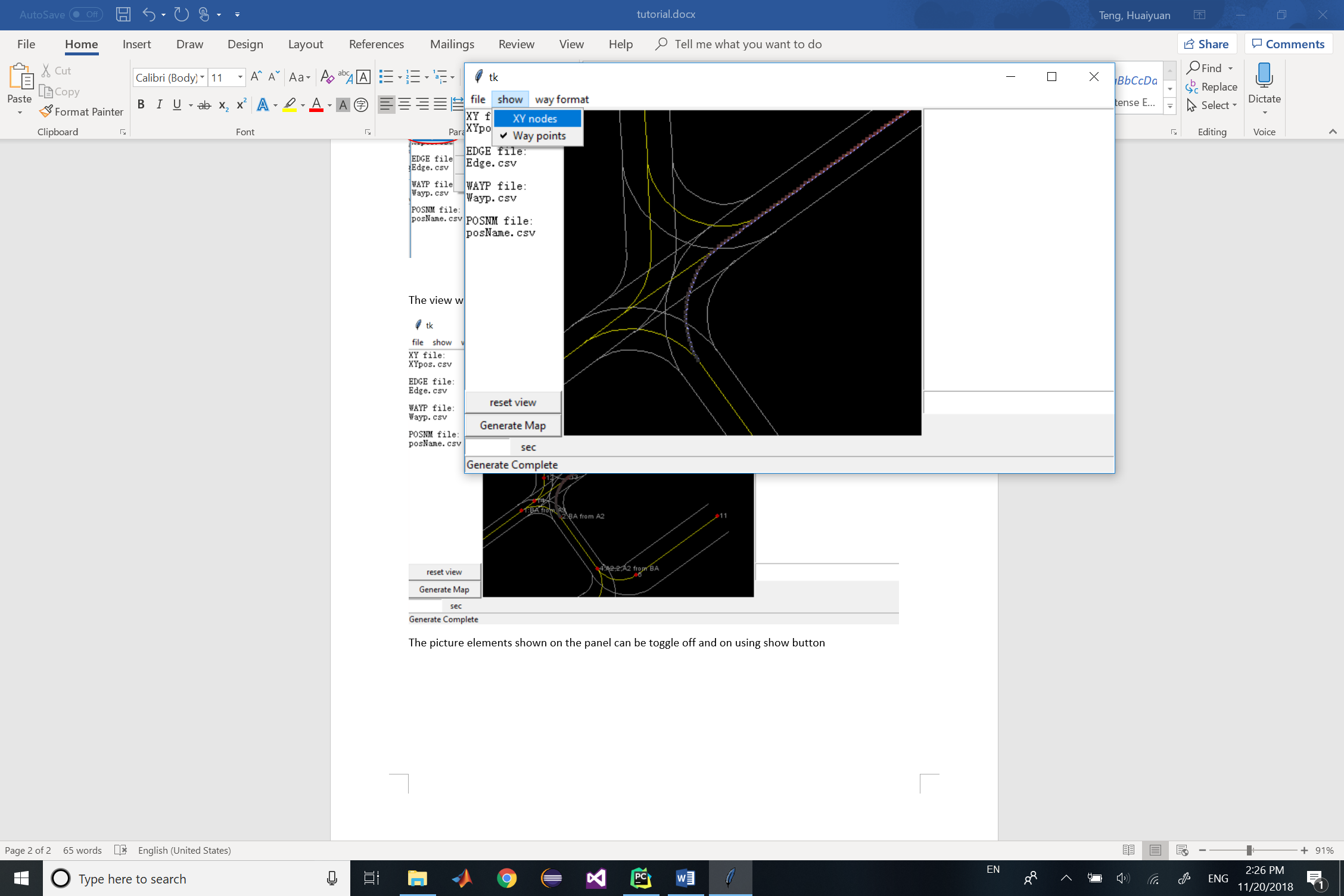
To change the file list on the left, use file->open menu.



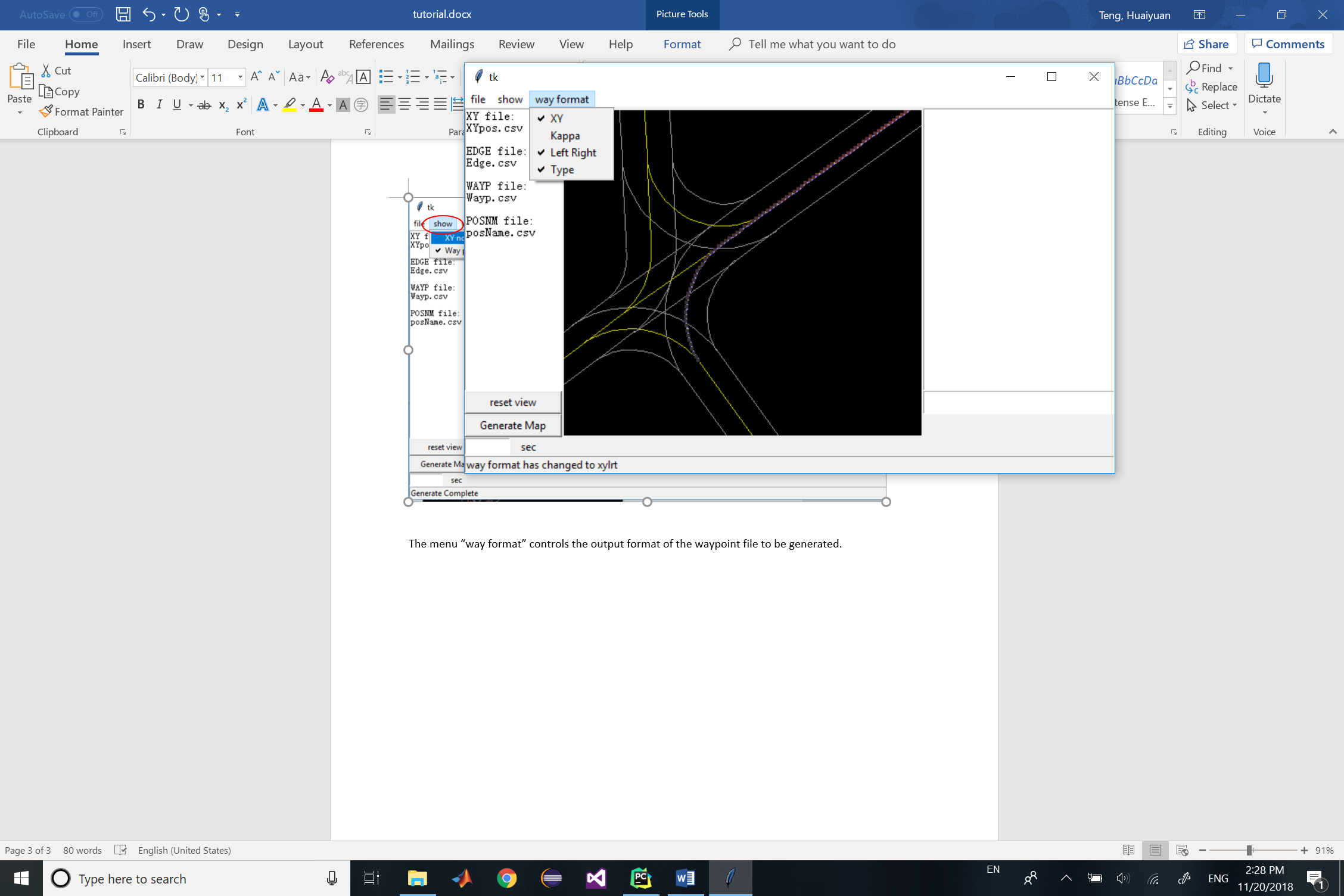
The view window can be zoomed using **mouse scroll** and panned holding **mouse** **right button**:



The picture elements shown on the panel can be toggle off and on using **show** menu.



The menu **way format** controls the output format of the waypoint file to be generated.



Once you toggled a selection here, the **message box** on the bottom shall tell you the latest entries of each row of the waypoint matrix to be generated.

For example, ‘xylrt’ indicates a row is [x,y,left,right,type].

‘xy’ indicates a row is [x,y]

…

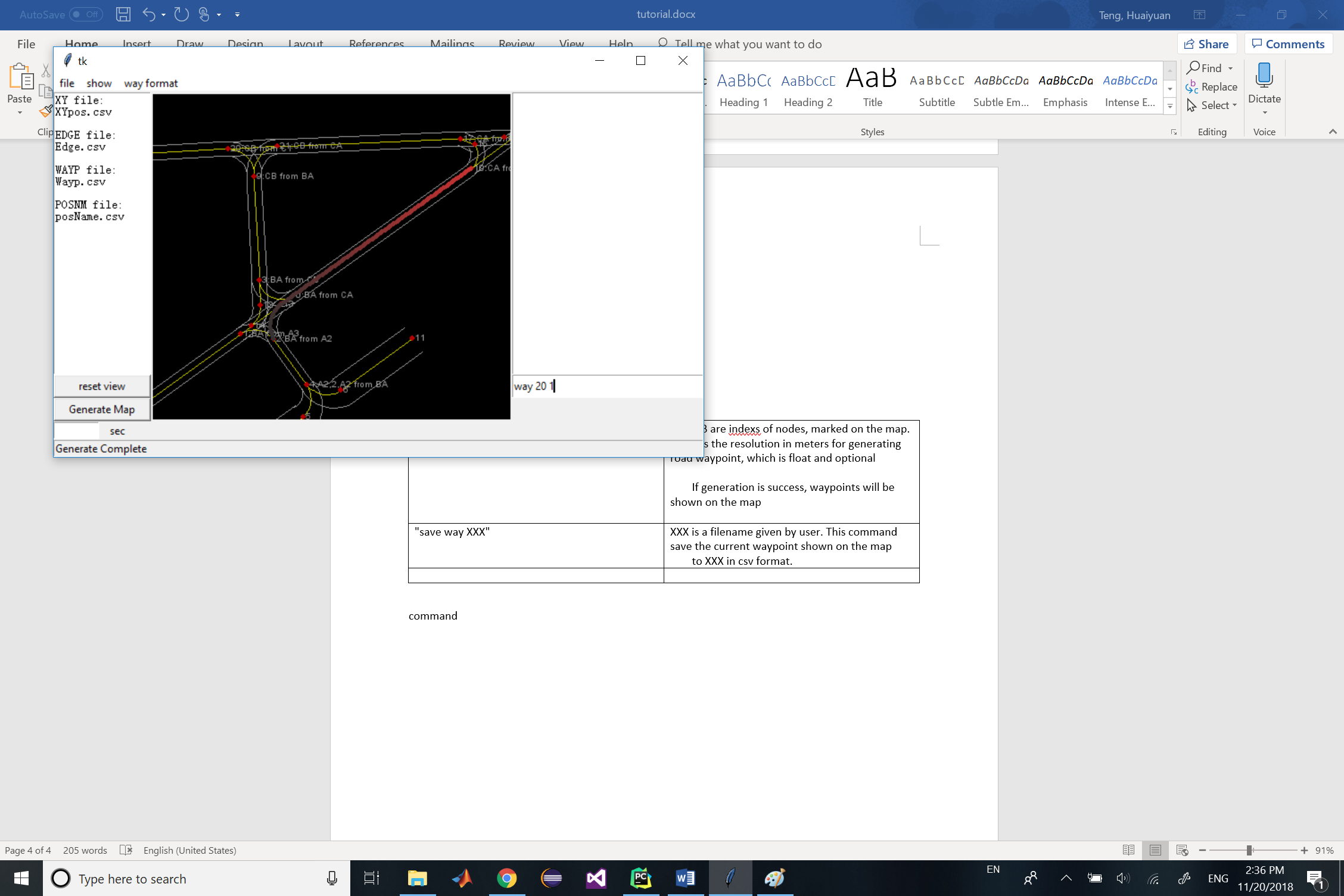
More information of waypoint types can be found in **appendix**.

# Commands

There is a **command box** on the right of the panel:

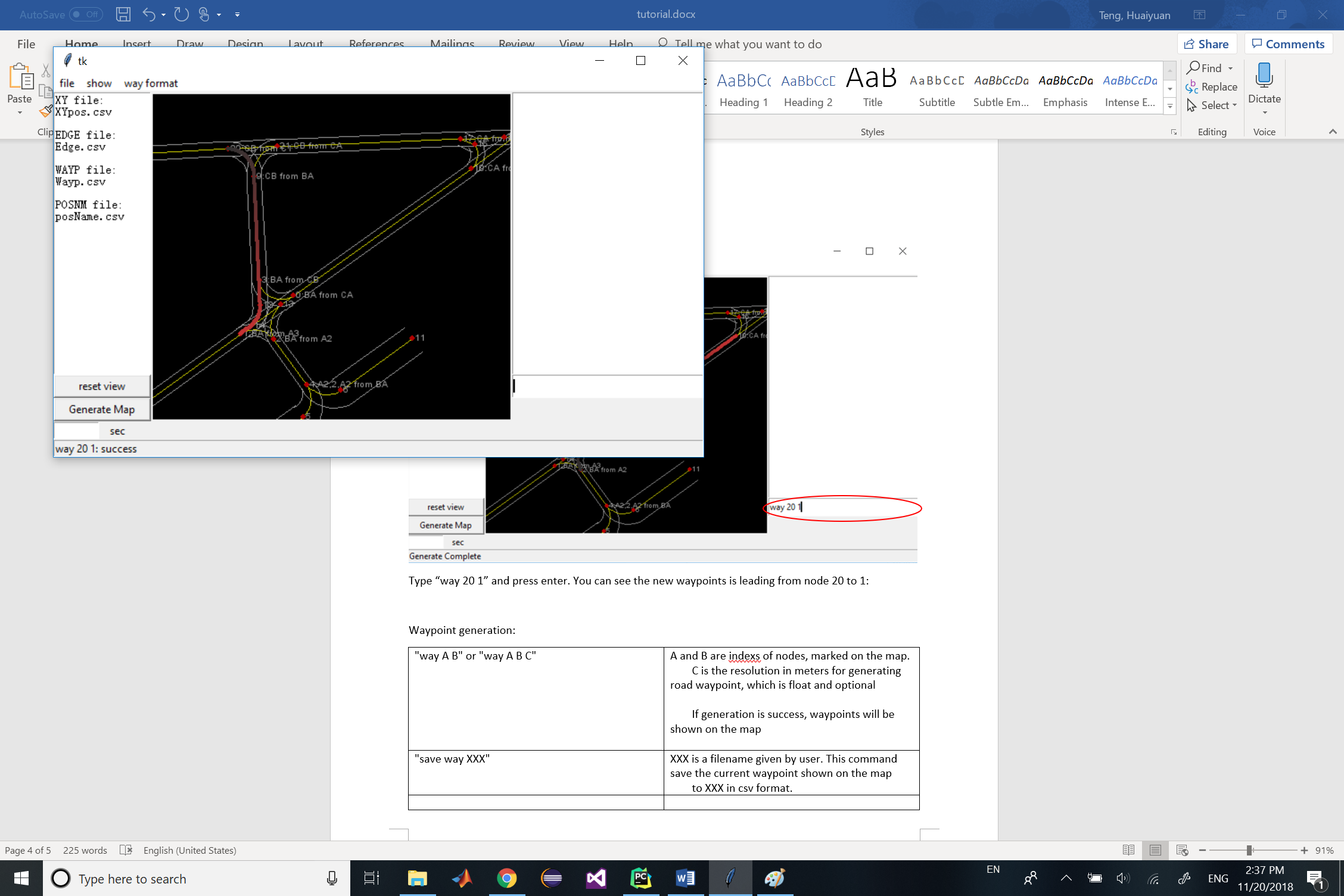
## Commands for path generation

To convert an ATC command into a path that can be used for waypoint generation, use the following command syntax: “command [start\_node] [end\_node] [taxiway 1] [taxiway 2] … [taxiway n]”, where [start\_node] and [end\_node] are the node names in naïve\_graph.csv, and the [taxiway i] are the taxiway names used in naïve\_graph.csv. The result of this command is shown in the command window, and, if a path is found, it is stored as “path1.txt”. This text file can be used in conjunction with the “ways” command as described in the following section.

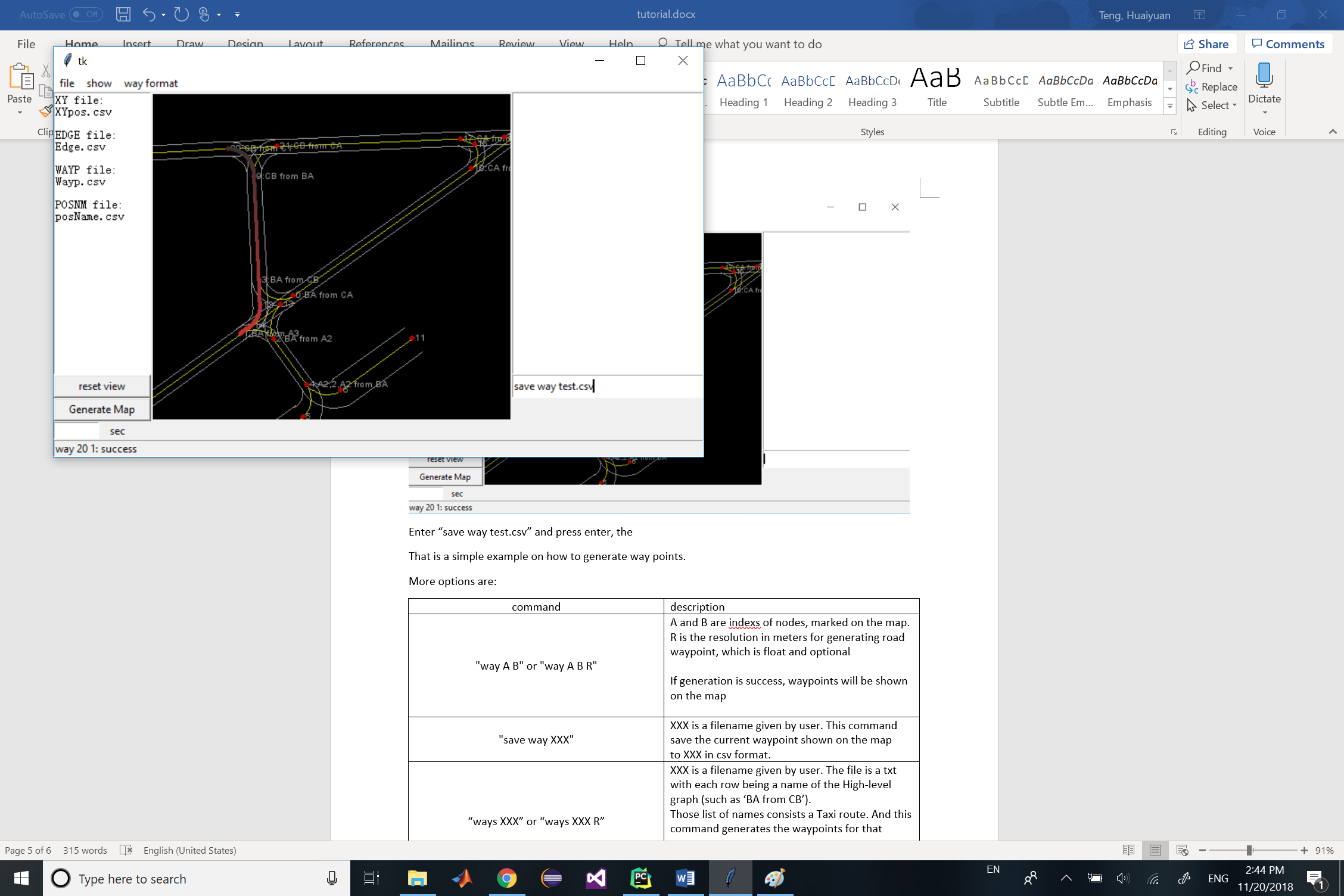


## Commands for Waypoint generation:

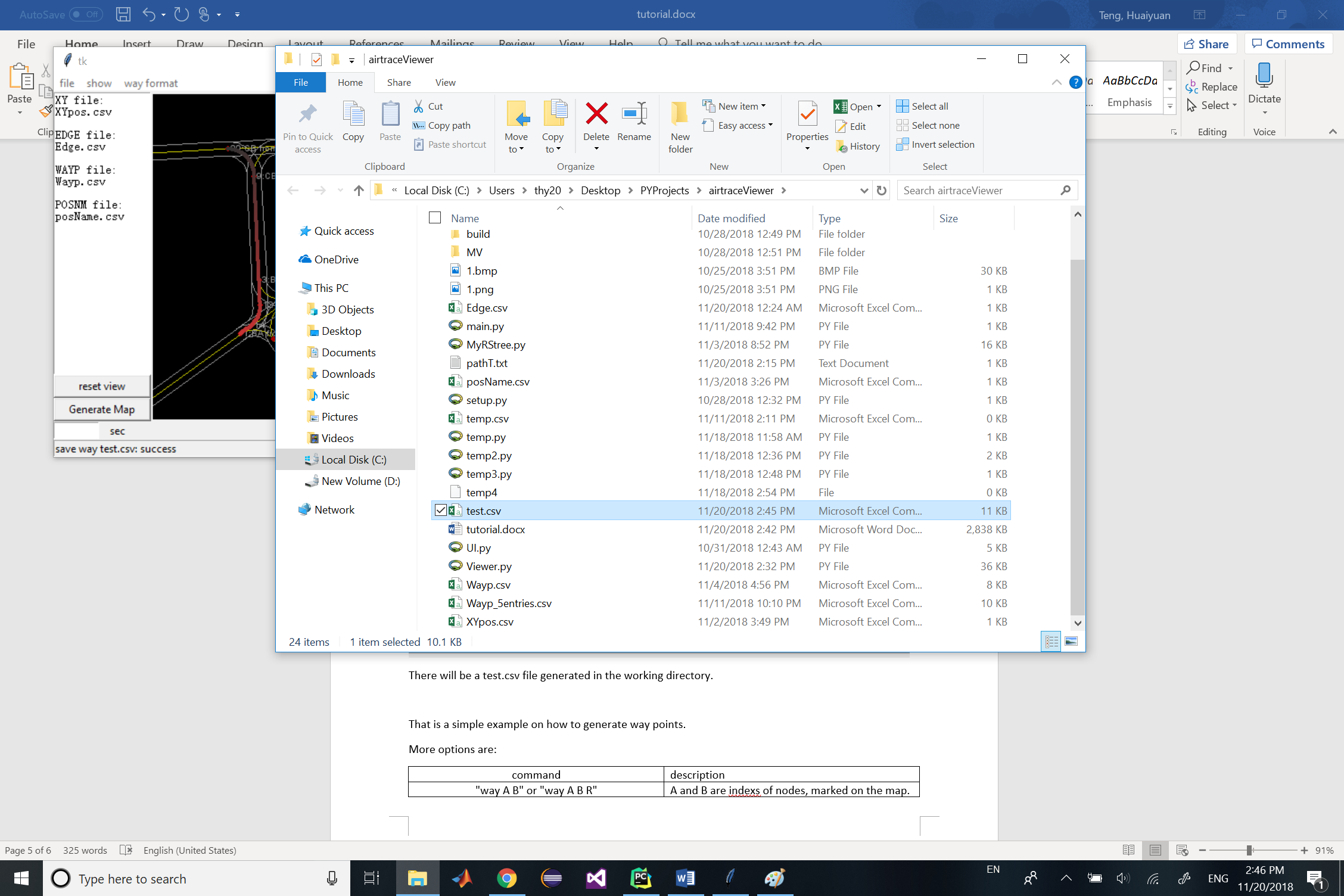
Type “way 20 1” and press enter. You can see the new waypoints is leading from node 20 to 1:



Enter “save way test.csv” and press enter



There will be a test.csv file generated in the working directory.

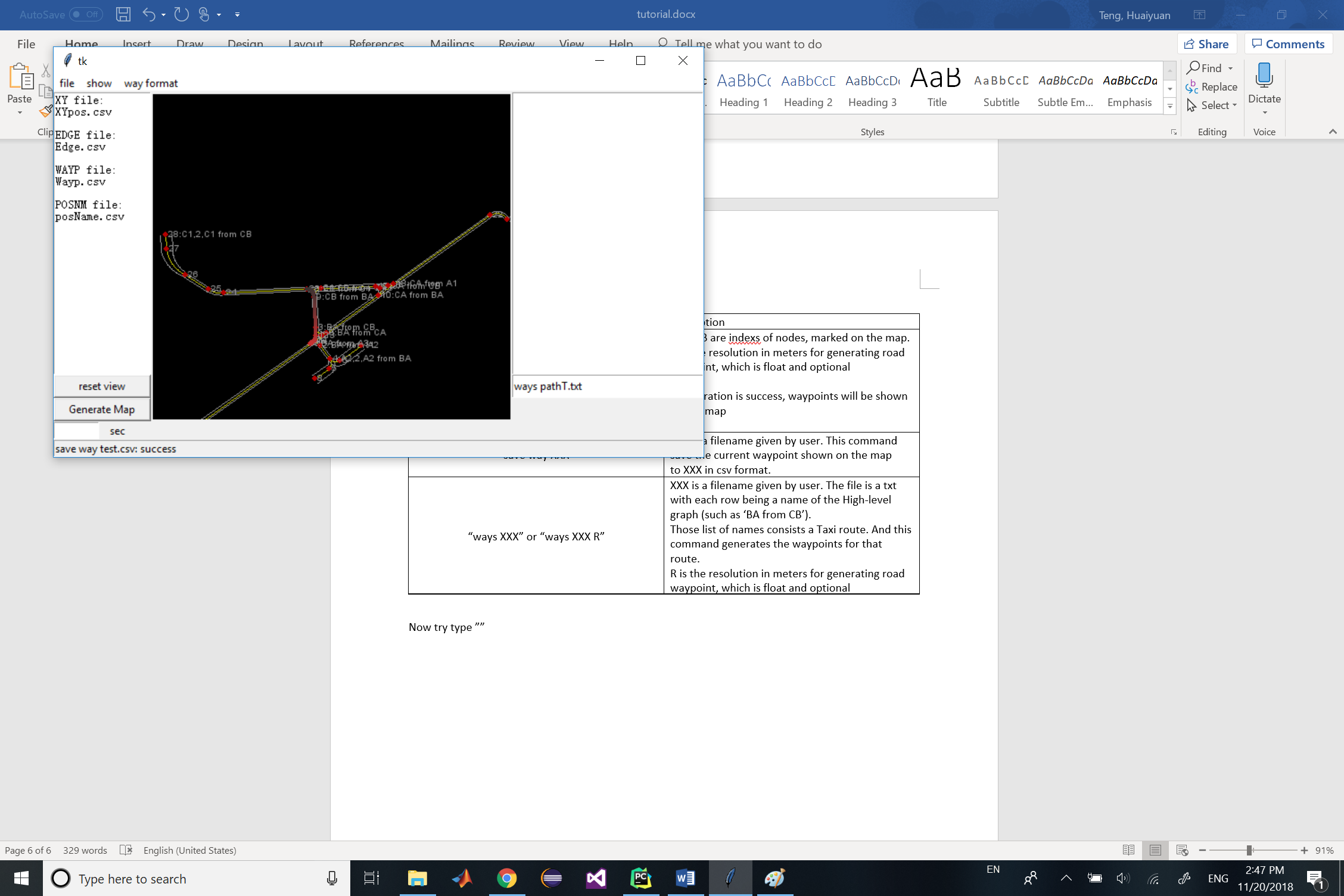


That is a simple example on how to generate way points.

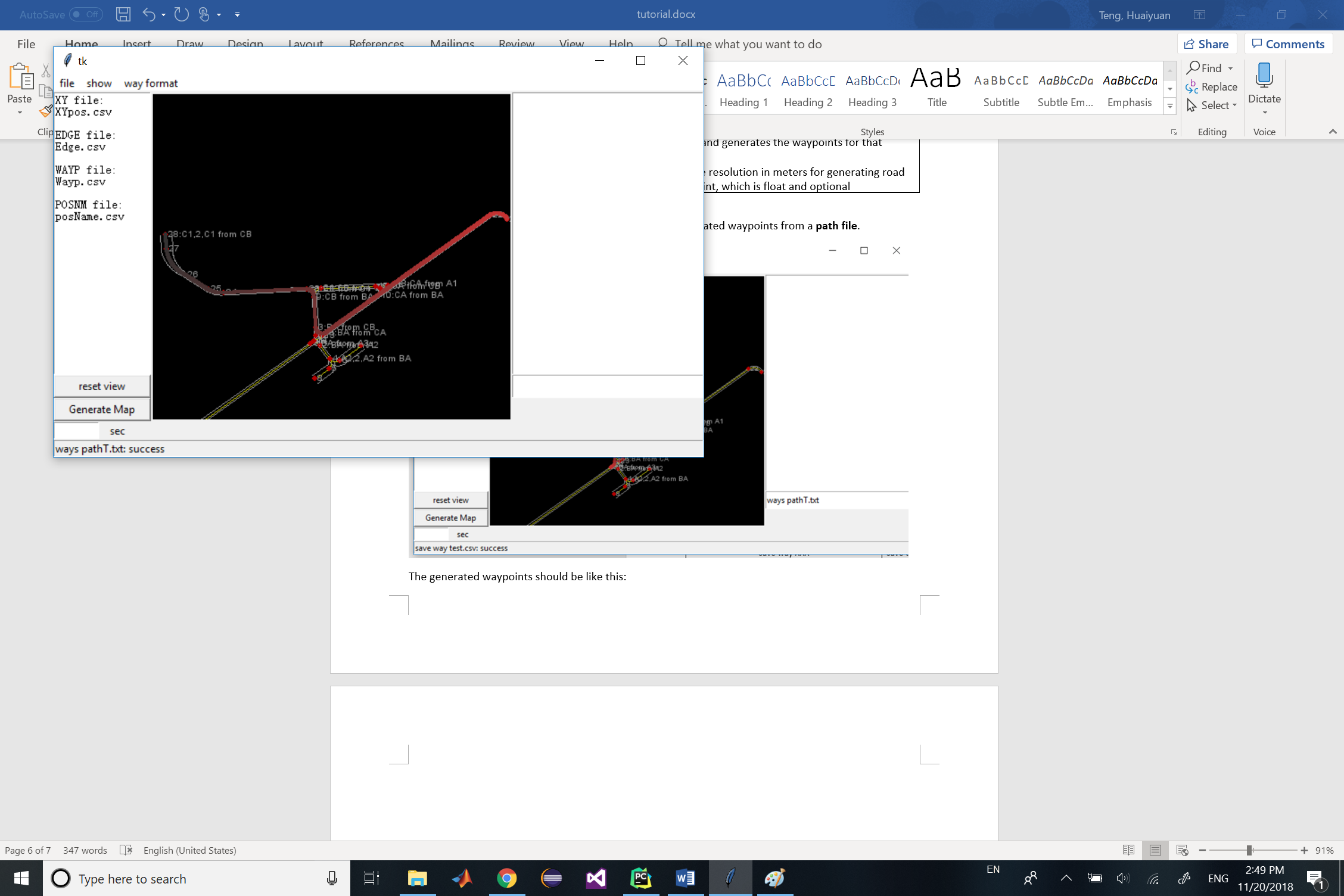
More options are:

|  |  |
| --- | --- |
| command | description |
| "way A B" or "way A B R" | A and B are indexs of nodes, marked on the map.  R is the resolution in meters for generating road waypoint, which is float and optional  If generation is success, waypoints will be shown on the map |
| "save way XXX" | XXX is a filename given by user. This command save the current waypoint shown on the map  to XXX in csv format. |
| “ways XXX” or “ways XXX R” | XXX is a **path file** given by user. The file is a txt with each row being a name of the High-level graph (such as ‘BA from CB’).  Those list of names consists a Taxi route. And this command generates the waypoints for that route.  R is the resolution in meters for generating road waypoint, which is float and optional |

Now try typing ”ways pathT.txt” to generate a more complicated waypoints from a **path file**.



The generated waypoints should be like this:



# Appendix

## Waypoint file types:

Possible entries of a waypoint includes:

|  |  |
| --- | --- |
| Name | description |
| XY | The flat map position of the road center(yellow line) of the waypoint. |
| Kappa | The curvature at the waypoint |
| Left, Right | The road width at the way point. |
| Type | An integer indicating the type of the road:  0: Straight Taxi-way  1: Curved Taxi-way  2: Straight Intersection  3: Curved Intersection  4, Straight Run-way  5, Curved Run-way |