

prepare_data.py

```
001 # Main-Script
002 # #####
003
004 # This python script can be used to generate a trajectory with constraints from
005 # a floorplan or similar geometry.
006
007 # Authors:
008 # Christopher Mahn
009 # Silas Teske
010 # Joshua Wolf
011
012 # #####
013
014 # Import of Libraries
015 # -----
016
017 import main as settings
018 # import string as st
019 # import random as r
020 # import re
021 # from turtle import position
022 # import matplotlib.pyplot as plt
023 # from scipy import interpolate
024 import numpy as np
025 import math as m
026 import json
027 # import sys
028 import os
029 # from scipy.fft import fft, fftfreq
030 # from scipy import signal
031
032
033 # -----
034 # Debugging-Settings
035
036 verbose = True # Shows more debugging information
037
038
039 # Functions
040 # -----
041
042 def import_data(input_filename):
043     # Import of measurements
044     if(verbose):
045         print(f'[INFO] Opening file "{input_filename}"', end="\r")
046     with open(os.path.join("data", input_filename)) as file:
047         if(verbose):
048             print(f'[INFO] Reading file "{input_filename}"', end="\r")
049         data = file.readlines()
050     if(verbose):
051         print(f'[INFO] Read file "{input_filename}" successfully')
052     text = ""
053     for i in data:
054         text = f'{text}{i.strip()}'
055     return(text)
056
057 def convert_walls(geometry):
058     coordinates = geometry["coordinates"][0]
059     lines = []
060     for i, e in enumerate(coordinates):
061         lines.append([coordinates[i-1][0], coordinates[i-1][1], e[0], e[1]])
062     return(lines)
063
064
065 # Classes
066 # -----
067
068
069 # Beginning of the Programm
070 # -----
071
072 if __name__ == '__main__':
073     for i, filename in enumerate(settings.project_filenames):
074         geometry = []
075         data = json.loads(import_data(f'{filename}.geojson'))
076         for feature in data["features"]:
077             if(feature["properties"]["Type"] == "Wall" or feature["properties"]["Type"] == "Fassade" or
078                feature["properties"]["Type"] == "B_Door" or feature["properties"]["Type"] == "Door" or
079                feature["properties"]["Type"] == "Balcony"):
080                 if(feature["geometry"] is not None):
```

```
079         walls = convert_walls(feature["geometry"])
080         for wall in walls:
081             geometry.append(wall)
082     print(f'[INFO] Fetched {len(geometry)} line-segments')
083     with open(os.path.join("data", f'{filename}_lines.csv'), "w") as file:
084         for j, e in enumerate(geometry):
085             file.write(f'{j}; {e[1]}; {e[0]}; {e[3]}; {e[2]}\n')
```