Global Inputs

Data that does not change during implementation

pysim5g run.py

```
PARAMETERS = {
    'indoor_users_percentage': 50,
    'los_breakpoint_m': 500,

    'tx_macro_baseline_height': 3(
    'tx_macro_power': 40,
    'tx_macro_gain': 16,
    'tx_macro_losses': 1,
    'tx_micro_baseline_height': 1(
    'tx_micro_power': 24,
    'tx_micro_gain': 5,
    'tx_micro_losses': 1,

    'rx_gain': 4,
    'rx_losses': 4,
    'rx_misc_losses': 4,
    'rx_height': 1.5,
    'network_load': 50,
    'asset_lifetime': 10,
    'discount_rate': 3.5,
    'opex_percentage_of_capex': 3(
```

Entered by the user

Entered by the user

Still not defined, probably user input

Entered by the user

```
COSTS = {
    #all costs in $USD
    'single_sector_antenna': 1500,
    'single_remote_radio_unit': 4(
    'single_baseband_unit': 10000,
    'tower': 10000,
    'civil_materials': 5000,
    'transportation': 10000,
    'installation': 5000,
    'site_rental': 9600,
    'power_generator_battery_syste
    'high_speed_backhaul_hub': 15(
    'router': 2000,
```

Entered by the user

```
ANT_TYPE = [
    ('macro'),
    ('micro'),
]
```

Selectable

```
CONFIDENCE_INTERVALS = [
    5,
    50,
    95,
]

INCREMENT_MA = (400, 30400, 2000)
INCREMENT_MI = (25, 500, 25)
```

Entered by the user (Default values)

```
SPECTRUM_PORTFOLIO = [
    # frequency, bandwidth,
generation, transmission_type
    (0.7, 10, '5G', '1x1'),
    (0.8, 10, '4G', '1x1'),
    (1.8, 10, '4G', '1x1'),
    (2.6, 10, '4G', '1x1'),
    (3.5, 40, '5G', '1x1'),
    (26, 100, '5G', '1x1'),
]
```

.csv input

```
def run_simulator
environments =[
    'urban',
    #'suburban',
    #'rural'
]
```

Selectable

Sources

Selectable files source
- Spectrum Portfolio.
- antenas.csv

antenas.csv

Contains a list of data, among other possible:

- Cell_id
- Long
- Lat

Variable Data

Using antenas.csv, create a voronoi shape using the position of the cells (long, lat).

For each antena.csv ID, call pysim5G with the global data and for every iteration:

pysim5g run.py line 908

```
unprojected_point = {
    'type': 'Feature',
    'geometry': {
        'type': 'Point',
        'coordinates': (-0.07496, 51.42411),
        },
    'properties': {
        'site_id': 'Crystal Palace
        }
    }
```

long, lat

Cell ID

pysim5g generate_hex.py

Using the Voronoi area, calculate radius using the distance between neighbouring transmitters and use the average result.

I don't know if it will be possible to use the adjacent transmitters as "interfering sites" if they are found instead of the automatically generated ones for the simulation.

pysim5g run.py line 1044

Using the Voronoi area of the transmitter, obtain OSM data

```
PARAMETERS = {

    'building_height': 5,
    'street_width': 20,

    'sectorization': 3,
    'mnos': 2,

}
```

OSM Data

Still not Defined

Output

Right now pysim5G gives tables of results and shapes of one antenna (the running one). For now the output would be the same but adding the id in the filename. It would be nice to use the template in vis folder for each antenna.