**City Simulator class:**

* Struct:

|  |  |  |
| --- | --- | --- |
| **CityConfigStr** | | |
| **Attribute** | **Definition** | **Unit** |
| map\_grid\_size | Discretization resolution | m |
| map\_x\_len | Map width | m |
| map\_y\_len | Map length | m |
| ave\_width | The width of the avenues (main streets) | m |
| st\_width | The width of the streets (lanes) | m |
| blk\_size\_x | The width of each block in the city (between avenues) | m |
| blk\_size\_y | The length of each block in the city (between avenues) | m |
| blk\_size\_small | Average small block size (between streets) | m |
| blk\_size\_min | Average min block size (between streets) | m |
| bld\_height\_avg | Building height average | m |
| bld\_height\_max | Building maximum height | m |
| bld\_height\_min | Building minimum height | m |
| bld\_size\_avg | Average size of the building | m |
| bld\_size\_min | Minimum size of the building | m |
| bld\_dense | The density of the building in each block | ? |

* Constructor

|  |  |
| --- | --- |
| **Input Arguments** | |
| **Name** | **Definition** |
| NoUe | Number of outdoor users to be distributed in the city. If is set to zero, then the user positions provided by argument “ue\_pos” is used, otherwise NoUe outdoor users are randomly and uniformly generated. |
| ue\_pos | Predefined user position |
| gen\_new\_map | If set to true, a new city is generated with random building height and building distribution following the urban\_config input argument.  If set to False, then a city is loaded form the directory provided by city\_file\_name argument. |
| save\_map | If set true, the generated city is saved in the given directory by city\_file\_name argument. |
| urban\_config | City configuration to generate new city buildings |
| city\_file\_name | The directory to save or load the city buildings. |

* Callable variables/Attribute

|  |  |
| --- | --- |
| **Name** | **Definition** |
| urban\_config | City configuration struct |
| buildings | A 3D numpy array containing the coordinates of the buildings in the city |
| num\_blds | Number of buildings in the city |
| grid\_x | 2D numpy array containing the x coordinate of each discrete grid of the city |
| grid\_y | 2D numpy array containing the y coordinate of each discrete grid of the city |
| height\_grid\_map | 2D matrix containing the height of each grid in the city |
| user\_pos | 2D numpy array, each row is the coordinate of each user |
| num\_users | Number of users |

* Functions:

|  |  |
| --- | --- |
| **generate\_city\_topology** | |
| **Output** | |
| **Name** | **Definition** |
| city\_blds | A 3D numpy array containing the coordinates of the buildings in the city |
| map\_x\_len | Map width |
| map\_y\_len | Map length |

|  |  |
| --- | --- |
| **gen\_height\_grid\_map** | |
| **Output** | |
| **Name** | **Definition** |
| grid\_x | 2D numpy array containing the x coordinate of each discrete grid of the city |
| grid\_y | 2D numpy array containing the y coordinate of each discrete grid of the city |
| height\_map | 2D matrix containing the height of each grid in the city |

|  |  |
| --- | --- |
| **user\_scattering** | |
| **Input Argument** | |
| **Name** | **Definition** |
| num\_user | Number of users to be generated randomly and uniformly over the city |
| **Output** | |
| **Name** | **Definition** |
| user\_arr | 2D numpy array, each row is the coordinate of each user |

|  |  |
| --- | --- |
| **user\_scattering** | |
| **Input Argument** | |
| **Name** | **Definition** |
| num\_user | Number of users to be generated randomly and uniformly over the city |
| **Output** | |
| **Name** | **Definition** |
| city\_blds | A 3D numpy array containing the coordinates of the buildings in the city |
| map\_x\_len | Map width |
| map\_y\_len | Map length |