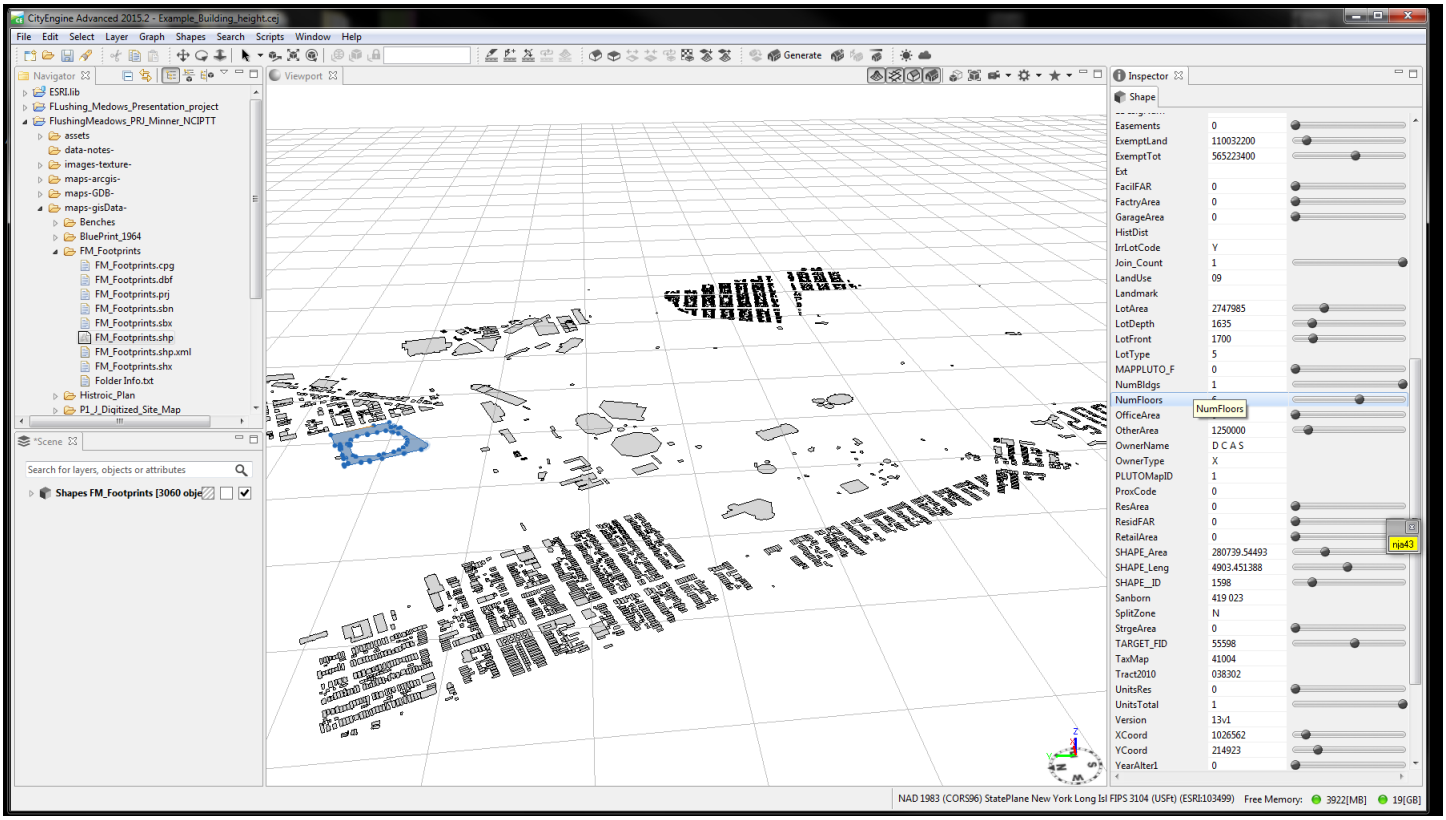
1. **Add your shape file to the City Engine Scene:**
    a. Select a shape and look at “Object attributes” in the inspector to verify that your desired data is associated.
    b. In this case I will be using the ‘NumFloors’ data from a building footprint .shp



2. Start a new .CGA rule file. You can use the following code

a. This rule allows for color and floor height to be specified and takes the number of floors to be placed from the .shp data.

b. Save your rule into the City Engine Workspace

c. to copy/paste code see last page (without color coding)

maps-arcgis-

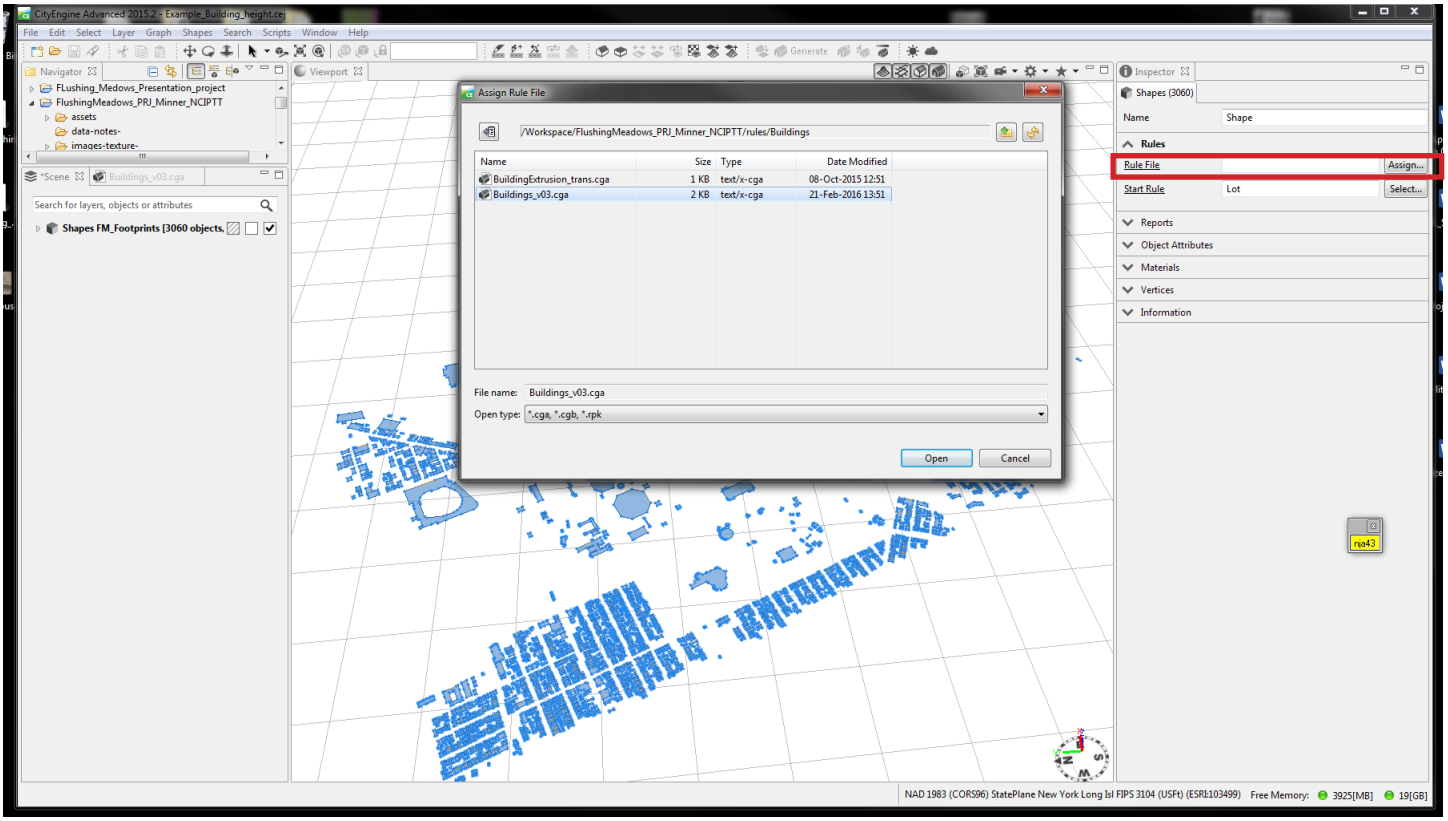
*Scene | Buildings_v03.cga

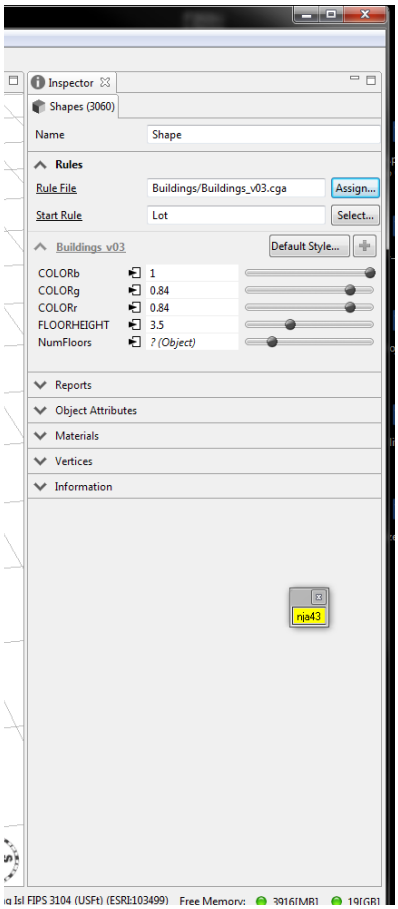
```
/**
 * File:    Buildings_v03.cga
 * Created: 7 Jan 2016 17:16:06 GMT
 * Author:  ??????
 */

version "2015.2"

//declare your variables for the rule [each will get slider of field in insepector]
// each variable is given a defult value here, but can be adjusted latter in the inspectors tab
attr FLOORHEIGHT = 3.5 // hight from floor to celing of building
attr COLORr = .84      // red of RGB
attr COLORg = .84     // green of RGB
attr COLORb = 1       ///  blue of RGB
attr NumFloors = 1    // defult value for NumFloors - will latter be set by object atribute in .shp file
                     // not that this variable has the exact name of the .shp attribute it will inherate



//creates the start rule
Lot -->
                // the extrude() comand will turn our footprint into a volume based on a unit of distance
                //in this case i will you FLOORHEIGHT (size of each story) multipled by NumFloors (number of stories)
   extrude(FLOORHEIGHT*NumFloors)

                // sets material transparincy level  ".68/1 "
   set(material.opacity, .68)

                //set color of object to RGB values choosen
   set(material.color.g, COLORg)
   set(material.color.r, COLORr)
   set(material.color.b, COLORb)
```

NAD 1983 (CORS96) StatePlane New York Long Isl FIPS 3

3. Select all objects on imported .shp and assign your new rule

a. use the inspector tab to assign the new rule file





4. the declared 'attr' variables will now appear within our inspector under the heading of the rule's name.

a. this will be in the inspector for any shapes that have had the rule applied to them

b. use these sliders to adjust the value for any selected shapes

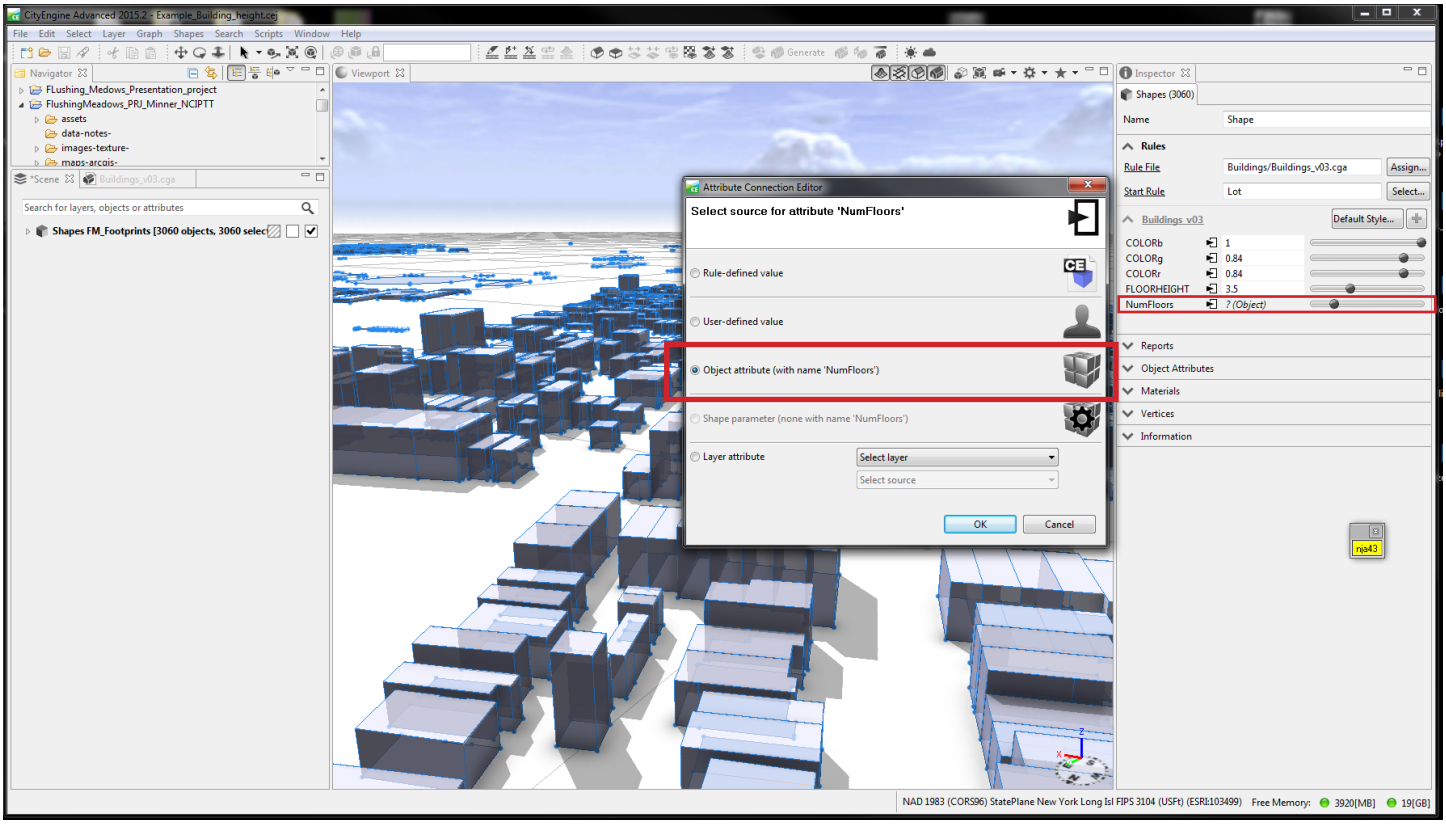
c. click the generate button at the top of the screen to visualize the results of applying the rule

5. Insure 'attr' variables link to .shp attributes for given shape

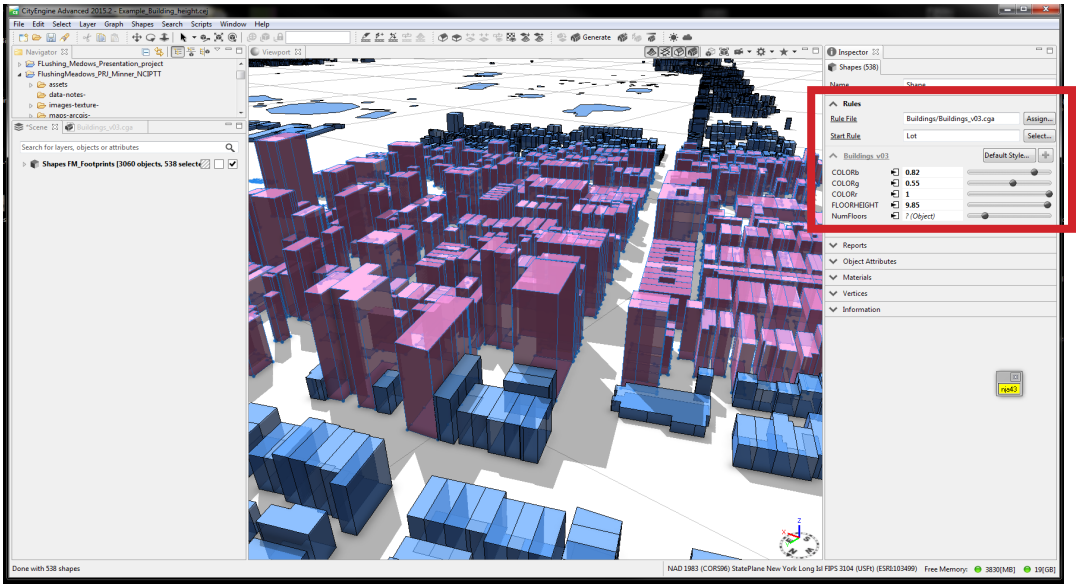
a. next to the 'NumFloors' variable listed with the rule's other attributes and Sliders, click the button with a black triangle pointing to a white box between the label and the slider.

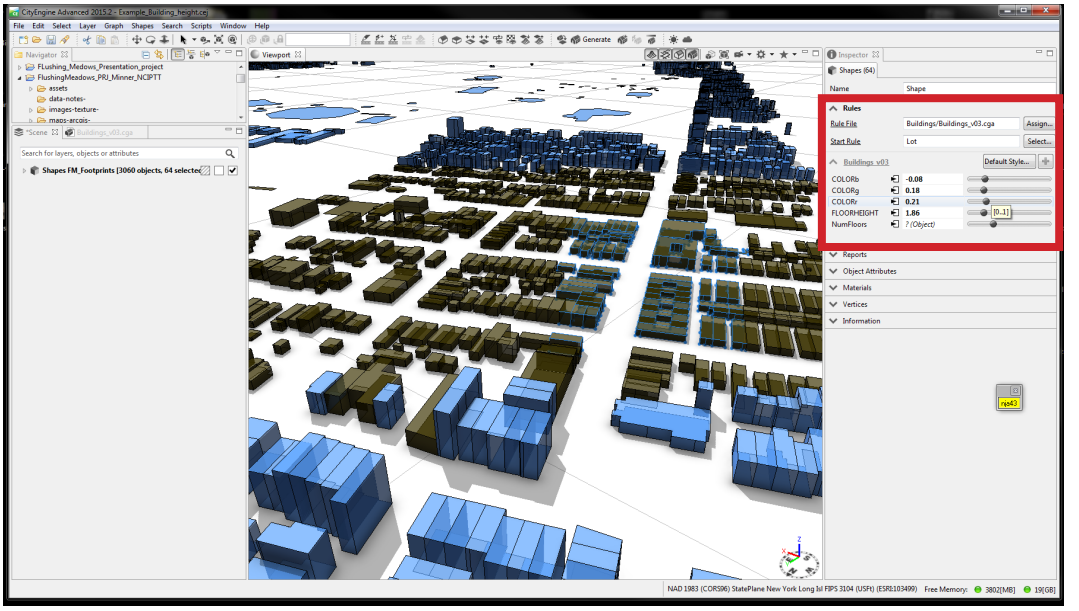
b. choose the option "Object attribute (with name 'NumFloors') and hit 'ok'

c. Re-generate selected layers



6. You can now adjust the ‘attr’ sliders and see the effect





```
/**
 * File:    Buildings_v03.cga
 * Created: 7 Jan 2016 17:16:06 GMT
 * Author:  ??????
 */

version "2015.2"

//Declare your variables for the rule [each will get slider of field in inspector]
// Each variable is given a default value here, but can be adjusted latter in the inspectors tab
attr FLOORHEIGHT = 3.5 // hight from floor to ceiling of building
attr COLORr = .84      // red of RGB
attr COLORg = .84     // green of RGB
attr COLORb = 1              ///  blue of RGB
attr NumFloors = 1    // defult value for NumFloors - will latter be set by object attribute in .shp file
                                                // not that this variable has the exact name of the .shp attribute it will
inherit



//creates the start rule
Lot -->
// the extrude() command will turn our footprint into a volume based on a unit of distance
 //in this case i will you FLOORHEIGHT (size of each story) multiples by NumFloors (number of stories)
   extrude(FLOORHEIGHT*NumFloors)

                                        // sets material transparency level  ".68/1 "
   set(material.opacity, .68)

                                        //set color of object to RGB values choosen
   set(material.color.g, COLORg)
   set(material.color.r, COLORr)
   set(material.color.b, COLORb)
```