

#TOWNHOUSE GRID/COLOR  
#Miranda Leighr, VSFx 319, Nov 2021

#USER NOTES:

#This code will create a grid of houses and add random colors to them.

#You can then adjust each color in the attribute editor shape node by selecting the piece you want to change in the viewer.

#

---

#IMPORT LIBRARIES

import maya.cmds as cmds  
import random as rand

#

---

#TURN ON VISIBILITY & DELETE EXISTING GROUPS

#Turn ON visibility of house\_grp to start in case code is re run multiple times:

```
def VisToggle(objName,vis):  
    cmds.setAttr('house_grp' + ".visibility", vis)
```

```
if cmds.objExists('house_grp'):  
    VisToggle('house_grp',1)
```

#delete new groups except original to reset grid/attributes every time

```
if cmds.objExists("hou1_*"):  
    cmds.delete("hou1_*")
```

```
if cmds.objExists('house_grp_copy'):  
    cmds.delete('house_grp_copy')
```

#

---

#DUPLICATE ORIGINAL HOUSE TO BUILD GRID FROM

#duplicate original house to build grid from and add attributes to this copy

```
cmds.duplicate('house_grp', n= 'house_grp_copy', rc=True)  
cmds.select('house_grp_copy', hi = True) #select entire house_grp_copy
```

#

---

## #SELECT SHAPE NODES AND ADD ATTRIBUTES

#select shape nodes

```
shapes = cmds.listRelatives(shapes=True)
sel= cmds.select(shapes)
sel = cmds.ls(sl = 1, type = "shape")
cmds.select(sel)
```

#add attributes to house\_grp\_copy shapes, one single attribute called ObjColor

```
cmds.addAttr(ln = 'mtoa_constant_ObjColor', usedAsColor=True, attributeType = 'float3')
cmds.addAttr(ln = 'myR', attributeType = 'float', parent = 'mtoa_constant_ObjColor')
cmds.addAttr(ln = 'myG', attributeType = 'float', parent = 'mtoa_constant_ObjColor')
cmds.addAttr(ln = 'myB', attributeType = 'float', parent = 'mtoa_constant_ObjColor')
```

#

---

## #CREATE PROCEDURAL GRID

#Global Variables for grid

#adjust Gridwidth and Griddepth to change the size of the neighborhood. Don't change Houwidth/Houdepth.

```
Gridwidth = 7
Griddepth = 7
Houwidth= 12
Houdepth= 10
```

#create the actual grid

```
for x in range(0,Gridwidth):
    for z in range(0,Griddepth):
        cmds.select("house_grp_copy")
        cmds.duplicate("house_grp_copy", rc=True) #rc will rename
        sel = cmds.ls(sl =1)
        #rename original 'house_grp' to make the duplicates
        cmds.rename(sel[0], "hou1_house_grp_" + str(x))
        cmds.move(x * Houdepth, 0, z * Houwidth, worldSpaceDistance=True)
        cmds.FreezeTransformations()
        cmds.DeleteHistory()
```

#

---

## #ADJUST VISIBILITY TO AVOID ANY GRID GEOMETRY OVERLAP

```

#turn off visibility of original house_grp
def VisToggle(objName,vis):
    cmds.setAttr('house_grp' + ".visibility", vis)

if cmds.objExists('house_grp'):
    VisToggle('house_grp',0)

#turn off visibility of house_grp_copy so it doesn't overlap in the grid
def VisToggle(objName,vis):
    cmds.setAttr('house_grp_copy' + ".visibility", vis)

if cmds.objExists('house_grp_copy'):
    VisToggle('house_grp_copy',0)

#


---




---



#RANDOMIZE HOUSE COLOR IN A LOOP FOR EACH PIECE

#define random value

colorRand = rand.random()

#***SIDING

#select all of each type of house piece

cmds.select('siding*', hi=True, vis = True)
shapes = cmds.listRelatives(shapes=True)
sidingSel= cmds.select(shapes)
sidingSel = cmds.ls(sl = 1, type = "shape")
print sidingSel

#get number of pieces procedurally in a list

sidingList = len(sidingSel)
print sidingList

#loop through each piece in list

for i in range(0,sidingList):
    colorRand = rand.random()
    cmds.setAttr(sidingSel[i] + ".mtoa_constant_ObjColor", 0.2 + (0.3*colorRand), 0.03,
0.01) #orange/red orange

#***WINDOWS

```

```
#select all of each type of house piece
```

```
cmds.select('windows*', hi=True, vis = True)
shapes = cmds.listRelatives(shapes=True)
windowsSel= cmds.select(shapes)
windowsSel = cmds.ls(sl = 1, type = "shape")
print windowsSel
```

```
#get number of pieces procedurally in a list
```

```
windowsList = len(windowsSel)
print windowsList
```

```
#loop through each piece in list
```

```
for i in range(0,windowsList):
    colorRand = rand.random()
    cmds.setAttr(windowsSel[i] + ".mtoa_constant_ObjColor", 1,1,1) #white
```

```
****GRASS
```

```
#select all of each type of house piece
```

```
cmds.select('grass*', hi=True, vis = True)
shapes = cmds.listRelatives(shapes=True)
grassSel= cmds.select(shapes)
grassSel = cmds.ls(sl = 1, type = "shape")
print grassSel
```

```
#get number of pieces procedurally in a list
```

```
grassList = len(grassSel)
print grassList
```

```
#loop through each piece in list
```

```
for i in range(0,grassList):
    colorRand = rand.random()
    cmds.setAttr(grassSel[i] + ".mtoa_constant_ObjColor", 0.1,0.2 + (0.1*colorRand),0.05)
#dark green
```

```
****SIDEWALK
```

```
#select all of each type of house piece
```

```
cmds.select('sidewalk*', hi=True, vis = True)
shapes = cmds.listRelatives(shapes=True)
sidewalkSel= cmds.select(shapes)
sidewalkSel = cmds.ls(sl = 1, type = "shape")
print sidewalkSel
```

```
#get number of pieces procedurally in a list
```

```
sidewalkList = len(sidewalkSel)
print sidewalkList
```

```
#loop through each piece in list
```

```
for i in range(0,sidewalkList):
    colorRand = rand.random()
    cmds.setAttr(sidewalkSel[i] + ".mtoa_constant_ObjColor", 0.5 + (0.1*colorRand), 0.5,
0.5) #gray
```

```
****CHIMNEY
```

```
#select all of each type of house piece
```

```
cmds.select('chimney*', hi=True, vis = True)
shapes = cmds.listRelatives(shapes=True)
chimneySel= cmds.select(shapes)
chimneySel = cmds.ls(sl = 1, type = "shape")
print chimneySel
```

```
#get number of pieces procedurally in a list
```

```
chimneyList = len(chimneySel)
print chimneyList
```

```
#loop through each piece in list
```

```
for i in range(0,chimneyList):
    colorRand = rand.random()
    cmds.setAttr(chimneySel[i] + ".mtoa_constant_ObjColor", 0.5, 0.16 * colorRand, 0.03)
#light orange chimney
```

```
****ROOF
```

```
#select all of each type of house piece
```

```
cmds.select('roof*', hi=True, vis = True)
shapes = cmds.listRelatives(shapes=True)
```

```
roofSel= cmds.select(shapes)
roofSel = cmds.ls(sl = 1, type = "shape")
print roofSel
```

```
#get number of pieces procedurally in a list
```

```
roofList = len(roofSel)
print roofList
```

```
#loop through each piece in list
```

```
for i in range(0,roofList):
    colorRand = rand.random()
    cmds.setAttr(roofSel[i] + ".mtoa_constant_ObjColor", 0.36, 0.08,
0.15*(0.5*colorRand)) #light brown/beige
```

```
****MAINHOUSE
```

```
#select all of each type of house piece
```

```
cmds.select('mainhouse*', hi=True, vis = True)
shapes = cmds.listRelatives(shapes=True)
mainhouseSel= cmds.select(shapes)
mainhouseSel = cmds.ls(sl = 1, type = "shape")
print mainhouseSel
```

```
#get number of pieces procedurally in a list
```

```
mainhouseList = len(mainhouseSel)
print mainhouseList
```

```
#loop through each piece in list
```

```
for i in range(0,mainhouseList):
    colorRand = rand.random()
    cmds.setAttr(mainhouseSel[i] + ".mtoa_constant_ObjColor", 0.02, 0.04, 0.1 +
(0.5*colorRand)) #dark blue base
```

```
****DOOR
```

```
#select all of each type of house piece
```

```
cmds.select('door*', hi=True, vis = True)
shapes = cmds.listRelatives(shapes=True)
doorSel= cmds.select(shapes)
doorSel = cmds.ls(sl = 1, type = "shape")
```

```
print doorSel
```

```
#get number of pieces procedurally in a list
```

```
doorList = len(doorSel)
```

```
print doorList
```

```
#loop through each piece in list
```

```
for i in range(0,sidingList):
```

```
    colorRand = rand.random()
```

```
    cmds.setAttr(doorSel[i] + ".mtoa_constant_ObjColor", 0.1-colorRand,0.02,0) #dark  
brown
```