

# 실전 활용

# Rotate object

```
❶ rollout rotate_obj "Rotate object"                -- rollout
❷ (
❸     -- 버튼 생성
❹     button x1_btn " + X " pos:[40,10]
❺     button x2_btn " - X " pos:[90,10]
❻     button y1_btn " + Y " pos:[40,50]
❼     button y2_btn " - Y " pos:[90,50]
❽     button z1_btn " + Z " pos:[40,90]
❾     button z2_btn " - Z " pos:[90,90]
❿     button reset_btn " Reset " pos:[40,130] width:85
⓫
⓬
⓭
⓮     -- 오브젝트를 회전시키는 함수
⓯     fn rotateObj x y z =
⓰     (
⓱         if selection.count > 0 do                -- 선택이 최소 1개 이상이면 동작
⓲         (
⓳             obj = selection[1]                    -- 첫 번째 오브젝트만 동작
⓴             in coordsys local rotate obj (EulerAngles x y z)    -- 회전
⓵         )
⓶     )
⓷ )
```

```

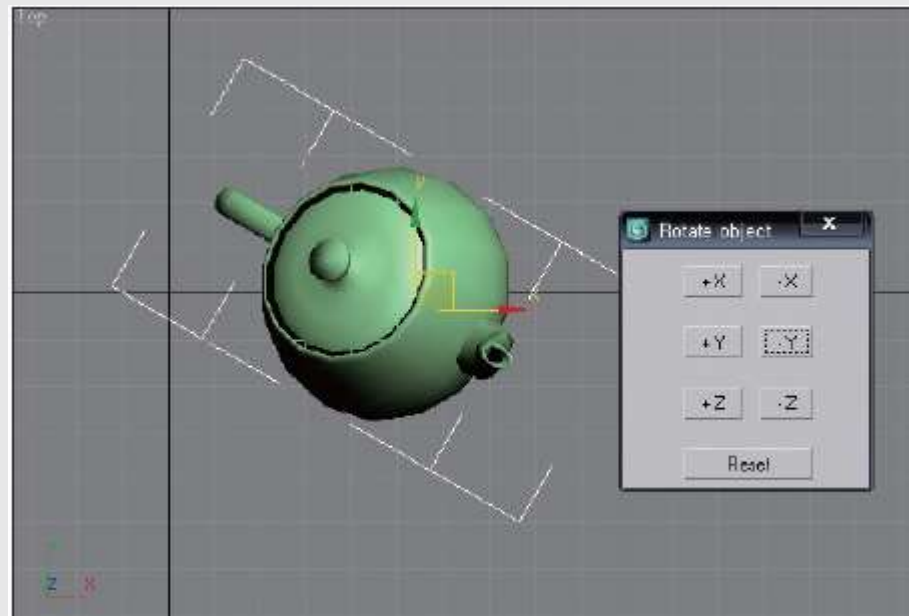
22
23  on x1_btn pressed do                                -- +X 버튼을 눌렀을 때의 이벤트
24  (
25      rotateObj 10 0 0                                -- rotateObj 함수 호출
26  )
27  on x2_btn pressed do
28  (
29      rotateObj -10 0 0
30  )
31
32  on y1_btn pressed do
33  (
34      rotateObj 0 10 0
35  )
36  on y2_btn pressed do
37  (
38      rotateObj 0 -10 0

```

```

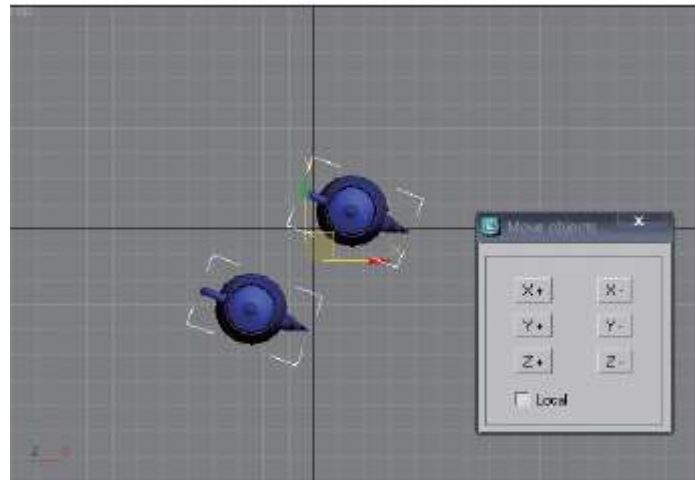
39 )
40
41 on z1_btn pressed do
42 (
43     rotateObj 0 0 10
44 )
45 on z2_btn pressed do
46 (
47     rotateObj 0 0 -10
48 )
49
50 -- 오브젝트를 원래 위치로
51 on reset_btn pressed do
52 (
53     if selection.count > 0 do
54     (
55         obj = selection[1]
56         bak_pos = obj.pos
57         obj.rotation = (eulerAngles 0 0 0)
58         obj.pos = bak_pos
59     )
60 )
61 )

```



62 createdialog rotate\_obj

# Move objects



실행 결과

```

1 rollout moveObjs_rollout "Move objects"           -- rollout 생성
2 (
3     groupBox group1 "" pos:[5,5] width:150 height:150    -- group 만들기
4     button x1_btn "X +" pos:[30,30]
5     -- 버튼 배치
6     button x2_btn "X -" pos:[100,30]
7     button y1_btn "Y +" pos:[30,60]
8     button y2_btn "Y -" pos:[100,60]
9     button z1_btn "Z +" pos:[30,90]
10    button z2_btn "Z -" pos:[100,90]
11    checkbox local_cb "Local" pos:[30,125]             -- local로 이동 체크박스
12
13    local v = 10                                         -- 오브젝트의 이동량
14
15
16    -- 오브젝트를 이동시켜 주는 함수
17    fn MoveObjFn obj pos =
18    (
19        if local_cb.checked == false then
20            obj.pos += pos
21    -- world 축으로 이동
22        else
23            in coordsys local obj.pos = pos             -- local 축으로 이동
24    )

```

```

27  on x1_btn pressed do                                -- X+ 버튼을 눌렀을 때 이벤트
28  (
29      for o in selection do                            -- 선택한것만 적용
30          MoveObjFn o  [v,0,0]                        -- MoveObjFn 함수 호출
31  )
32
33  on x2_btn pressed do
34  (
35      for o in selection do
36          MoveObjFn o  [-v,0,0]
37  )
38
39  on y1_btn pressed do
40  (
41      for o in selection do
42          MoveObjFn o  [0,v,0]
43  )
44
45  on y2_btn pressed do
46  (
47      for o in selection do
48          MoveObjFn o  [0,-v,0]
49  )

```

```

51  on z1_btn pressed do
52      (
53          for o in selection do
54              MoveObjFn o [0,0,v]
55      )
56
57  on z2_btn pressed do
58      (
59          for o in selection do
60              MoveObjFn o [0,0,-v]
61      )
62
63
64 )
65 createdialog moveObjs_rollout      -- 다이얼로그 생성

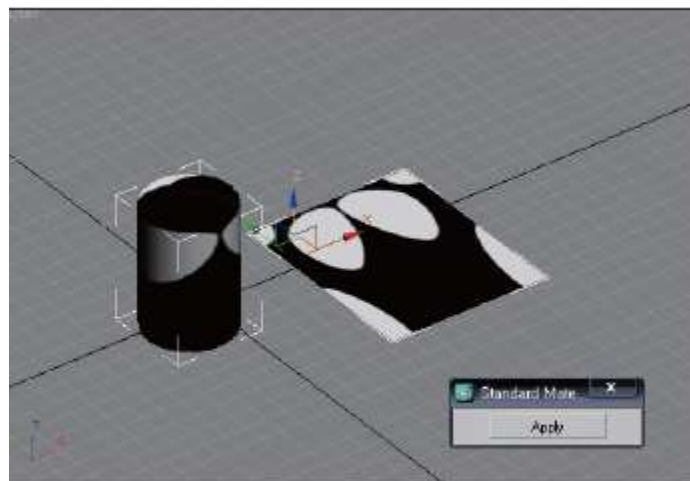
```



# Standard Material

Script의 상세한 실행 순서는 다음과 같습니다.

1. Standard Material을 적용한다.
2. Show Standard Map in Viewport 버튼을 켜다.
3. "3dsmaxRoot\maps\WWBIGLEAF3.TGA"는 파일 이름의 텍스처 생성한다.
4. 생성된 텍스처를 diffusemap과 bumpmap에 적용한다.
5. 텍스처 안의 U,V Offset을 0.5로 설정한다.
6. 텍스처 Output의 Bump Amount를 2로 설정한다.



실행 결과

```

❶ fn ApplyStandardMat =
❷ (
❸     st = StandardMaterial  showInViewport:true           -- Standard Material 적용
❹     file = GetDir #maxroot + "maps\\BIGLEAF3.TGA"        -- file 이름
❺     tex = Bitmaptexture filename:file                    -- bitmap Texture를 file 이름으로 생성
❻     st.diffusemap = tex                                   -- standard Material의 diffuse Color에 tex(bitmapTexture) 적용
❼     st.bumpmap = tex                                     -- standard Material의 bump에 tex(bitmapTexture) 적용
❽     st.twoSided = true                                   -- twoSide On
❾
❿
⓫     objs = selection as array                            -- 선택한 오브젝트들을 배열로
⓬     for i in objs do
⓭         i.material = st                                  -- 선택한 오브젝트에 하나씩 st(standard material)을 적용
⓮
⓯     tex.coords.u_offset = 0.5                            -- texture coordinates의 u_offset을 0.5로
⓰     tex.coords.v_offset = 0.5                            -- texture coordinates의 v_offset을 0.5로
⓱     tex.output.Bump_Amount = 2                           -- texture output의 bump Amount를 2.0으로
⓲ )

```

```

22 rollout StandardMat_roll "Standard Material"
23 (
24     button Apply_btn "Apply" width:100
25
26     on Apply_btn pressed do
27     (
28         ApplyStandardMat()          -- ApplyStandardMat 함수 호출
29     )
30 )
31 createdialog StandardMat_roll

```

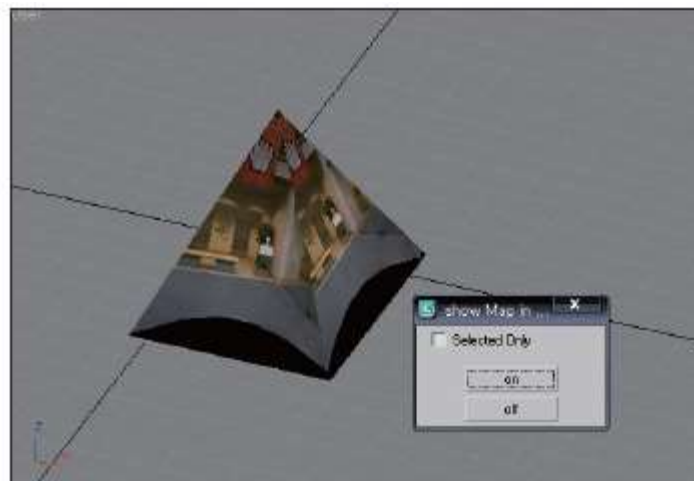
# Show Map in Viewport

```
1 rollout showMapinViewport_roll "show Map in Viewport"
2 (
3     checkbox sel_cb "Selected Only"                -- checkbox 생성
4     button on_btn "on"    width:80  offset:[0,10]    -- on 버튼 생성
5     button off_btn "off"  width:80                  -- off 버튼 생성
6
7     -- showMapViewport 함수
8     fn showMapViewport bool sel =
9         (
10             local objs = #()
11
12             if sel == true then                    -- checkbox의 체크에 따라서 분기
13                 objs = selection as array          -- 선택한 오브젝트를 배열로
14             else
15                 objs = objects as array            -- 모든 오브젝트를 배열로
16
17             -- 오브젝트를 차례대로 loop
18             for i in objs do
19                 (
20                     if classof i.material == Standardmaterial do -- Standard Material 인지
21                         판별
22                         i.material.showInViewport = bool -- showInViewport 버튼 켜기
23                 )
24             )
25         )
26     )
```

```

24
25  on on_btn pressed do          -- on 버튼을 눌렀을 경우 이벤트
26  (
27      showMapViewport true sel_cb.checked    -- showMapViewport 함수 호출
28  )
29
30  on off_btn pressed do        -- off 버튼을 눌렀을 경우 이벤트
31  (
32      showMapViewport false sel_cb.checked
33  )
34
35 )
36 createdialog showMapinViewport_roll

```



실행 결과

# Clean Material

## (메터리얼 슬롯 지우기)

```
① -- Clean Material 해주는 함수
② fn CleanMatSlot =
③ (
④     for i in 1 to 24 do                -- 1부터 24번 슬롯을 loop
⑤     (
⑥         na = formattedprint i format:"02d"        -- formattedprint 함수를 이용하여 숫자 붙이기
⑦         na += " - Default"                        -- 문자열 결합
⑧         medimaterials[i] = standard name:na        -- medimaterials 클리어
⑨     )
⑩ )
⑪ -- 선택한 오브젝트의 material을 지우는 함수
⑫ fn killSelectMaterials =
⑬ (
⑭     selobjs = selection as array
⑮
⑯     for i in selobjs do
⑰     (
⑱         i.material = undefined                -- material 지우기
⑲     )
⑳ )
㉑ )
```

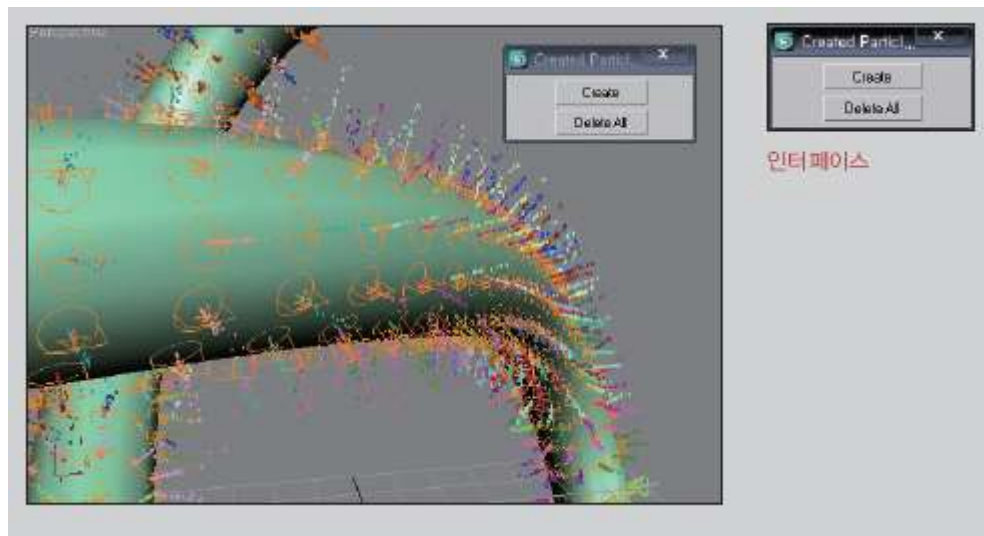


```

23 rollout CleanMat_rollout "Clean Material"
24 (
25     -- button 생성 영역
26     button CleanMat_bnt "Clean Material Slot"    width:100
27     button killMat_btn "Kill Material"            width:100
28
29
30     on CleanMat_bnt pressed do                    -- Clean 버튼을 눌렀을 때의 이벤트
31     (
32         CleanMatSlot()                            -- CleanMatSlot 함수 호출
33     )
34
35     on killMat_btn pressed do                      -- Kill Material 버튼을 눌렀을 때의 이벤트
36     (
37         killSelectMaterials()                      -- killSelectMaterials 함수 호출
38     )
39 )
40 createdialog CleanMat_rollout                    -- 다이얼로그 생성

```

# Created Particle From Selection Object





```

1 -- CreateParticle 함수
2 fn CreateParticle pos myTime normal =
3 (
4     fireParticle = superspray()           -- Super Spray 생성
5     fireParticle.iconSize = 2             -- icon Size
6     fireParticle.Axis_Spread = 10         -- Axis_Spread
7     fireParticle.Plane_Spread = 10        -- Plane_Spread
8     fireParticle.quantityMethod = 1       -- Particle Quantity 모드
9     fireParticle.Total_Number = 1000      -- Total_Number
10    fireParticle.viewPercent = 80         -- Percentage of Particles
11    fireParticle.Speed = 0.2              -- speed
12    fireParticle.Speed_Variation = 50     -- speed Variation
13    nameStr = uniqueName "fireParticle"   -- Particle 이름
14    fireParticle.name = nameStr
15    fireParticle.Emitter_Start = mytime    -- Emitter_Start 설정
16    fireParticle.Emitter_Stop = 1000f     -- Emitter_Stop
17    fireParticle.Display_Until = 1000f     -- Display_Until
18    fireParticle.seed = random 0 25000    -- Seed값은 랜덤으로 설정
19    fireParticle.pos = pos                 -- Particle 위치를 이동
20    fireParticle.dir = normal              -- Particle Z 방향을 설정
21 )

```



```

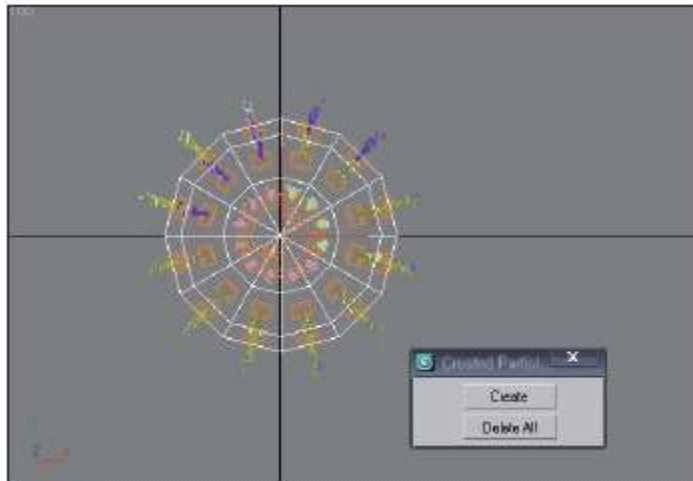
23 -- rollout 생성
24 rollout CreatedParticleFromSelObj_roll "Created Particle From Selection Object"
25 (
26     -- 버튼 생성 영역
27     button Create_btn "Create" width:80
28     button DeleteAll_btn "Delete All" width:80
29
30     on Create_btn pressed do                -- Create 버튼을 누를 때 이벤트
31     (
32         if (selection.count == 1) do        -- 선택한 오브젝트가 1개일 때만 동작
33         (
34             c=selection as array            -- 선택한 오브젝트를 배열로
35             c = convertTopoly c[1]          -- Editable Poly로 변환
36
37             for i in c.faces do             -- faces loop
38             (
39                 -- Face Center 구하기
40                 newPos = polyOp.getFaceCenter c i.index
41
42                 -- Face Normal 구하기
43                 normal = polyOp.getFaceNormal c i.index
44
45                 CreateParticle newPos 1 normal -- CreateParticle 함수 호출
46             )
47         )
48     )

```

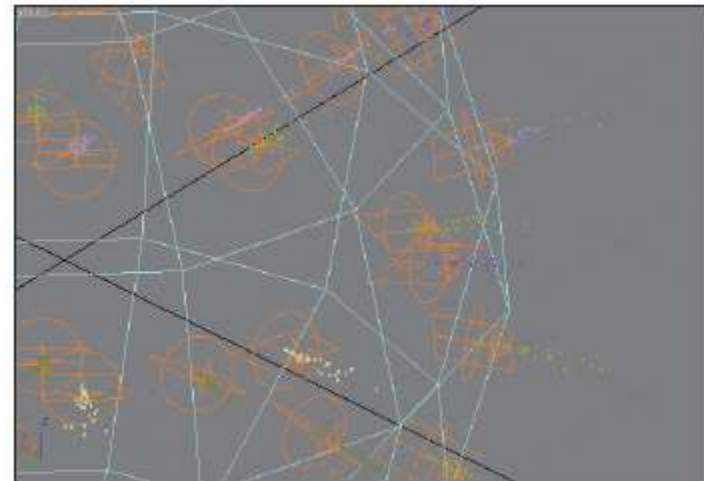
```

49
50  on DeleteAll_btn pressed do          -- Delete All 버튼을 누를 때 이벤트
51      (
52          delete $fireParticle*
53      )
54  )
55  createdialog  CreatedParticleFromSelObj_roll          -- 다이얼로그 생성

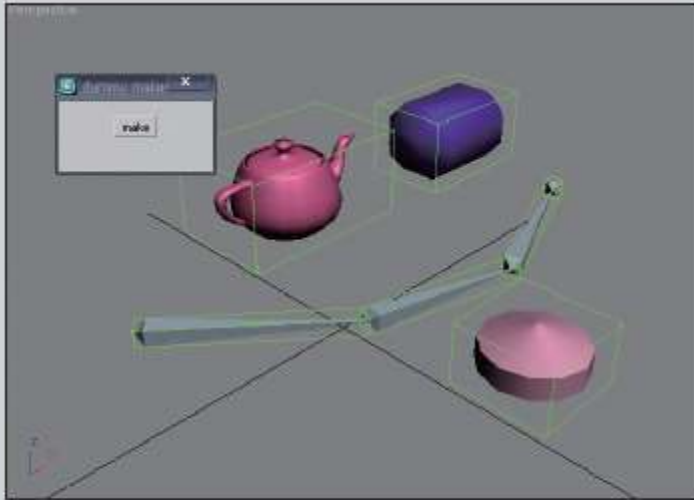
```



실행 결과



## (선택한 오브젝트에 dummy 생성)



이 Script는 주로 더미를 정확히 오브젝트의 위치에 배치하고자 할 때 사용되는데 world 좌표에 의해 실행되는 것이 아니라 오브젝트의 Local 좌표를 베이스로 하기 때문에 오브젝트가 회전되어 있을 경우라도 정확하게 배치됩니다.

```

❶ rollout dummy_maker_rollout "dummy maker"      -- rollout 생성
❷ (
❸     button make_btn "make" offset:[0,10]        -- 버튼 생성
❹
❺     -- makeDummy 함수
❻     fn makeDummy obj =
❼         (
❽             bak_mat = obj.transform              -- 오브젝트 transform을 백업
❾
❿             obj.transform = matrix3 1            -- 오브젝트 transform을 초기화
⓫             size = obj.max - obj.min            -- 오브젝트 크기를 구한다.
⓫             mydummy = dummy boxsize:size        -- dummy 생성
⓫             offset = obj.center - mydummy.center -- 오브젝트와 dummy의 차이를 구한다.
⓫             mydummy.pos += offset               -- 오브젝트 offset을 증가
⓫             mydummy.pivot = obj.pivot           -- 오브젝트 pivot을 dummy pivot으로 복사
⓫             obj.transform = bak_mat              -- 오브젝트를 원래 위치 되돌린다.
⓫             mydummy.transform = bak_mat          -- Dummy도 bak_mat로 보내기
⓫         )
❻     )

```

```

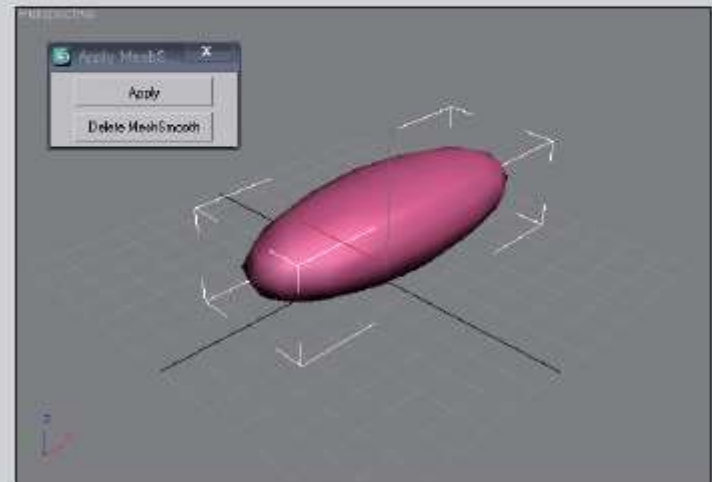
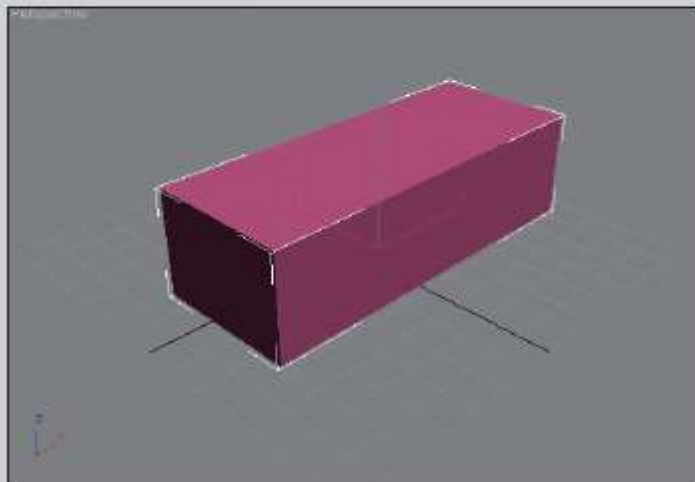
20 on make_btn pressed do
21   (
22     objs = selection as array          -- 선택한 오브젝트를 배열로
23
24     -- 선택한 오브젝트들을 차례대로 실행
25     for i in objs do
26       makeDummy i
27   )
28
29 )
30
31 -- 다이얼로그 생성
32 createdialog dummy_maker_rollout  width:150 height:70

```

# MeshSmooth 적용하기



인터페이스



```

❶ -- MeshSmooth를 적용하는 함수
❷ fn setPoly poly =
❸ (
❹     -- 베이스 오브젝트가 Editable_Poly가 아니면 리턴
❺     if classof poly.baseobject != Editable_Poly do return undefined
❻     -- MeshSmooth가 없다면 적용
❼     if poly.modifiers[#meshSmooth] == undefined do
❽         (
❾             addmodifier poly (MeshSmooth())
❿         )
⓫
⓬
⓭     meshSmt = poly.modifiers[#MeshSmooth]          -- MeshSmooth modifier
⓮     meshSmt.iterations = 2                          -- iterations 적용
⓯ )
⓰
⓱ -- MeshSmooth 지우기
⓲ fn deleteMeshSmooth poly =
⓳ (
⓴     ms = poly.modifiers[#MeshSmooth]                -- MeshSmooth modifier
⓵     if ms == undefined do return undefined          -- MeshSmooth가 없다면 리턴
⓶     deletemodifier poly ms                          -- 지운다.
⓷
⓸ )

```

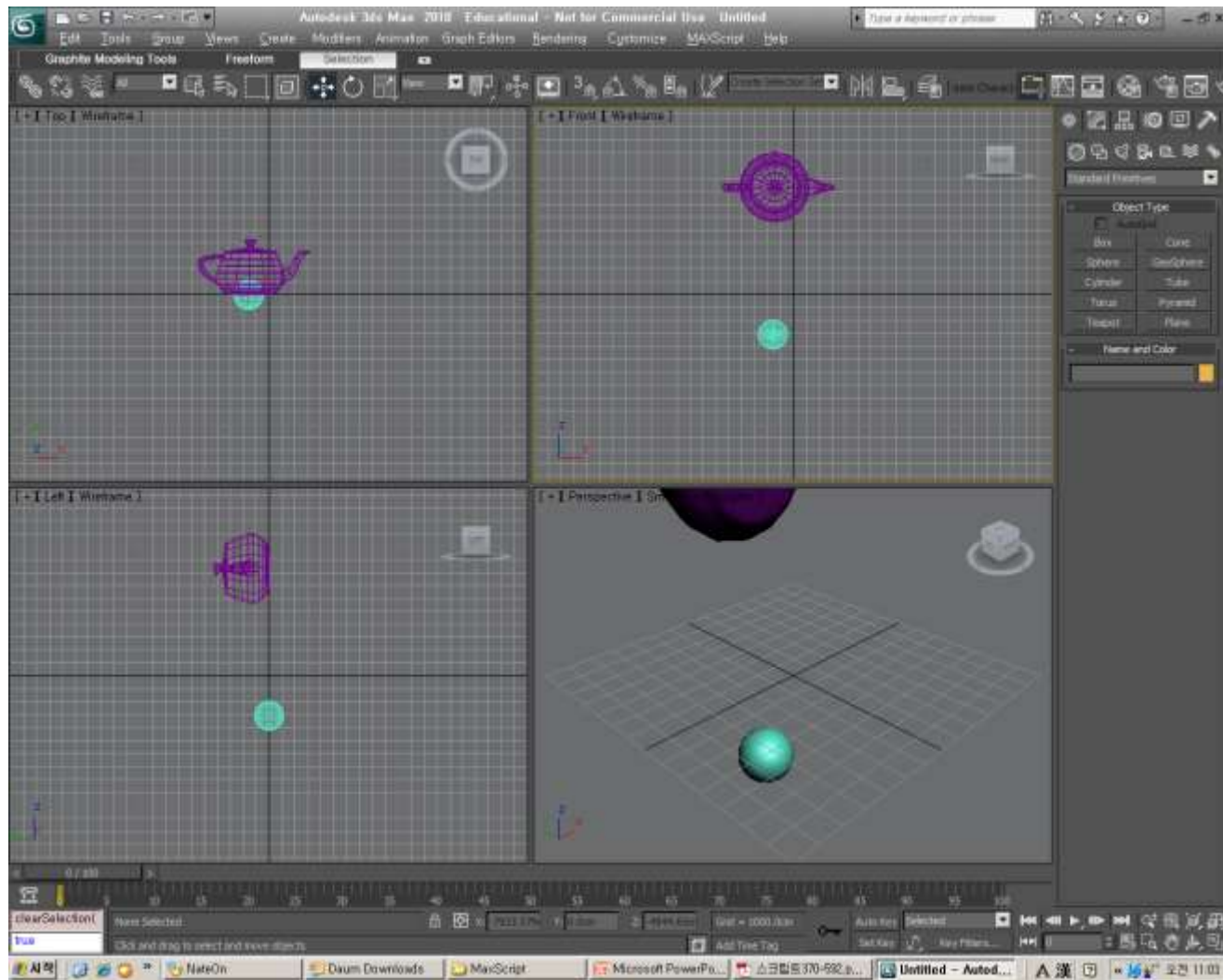


```

26 -- rollout 영역
27 rollout ApplyMeshSmooth_roll "Apply MeshSmooth"
28 (
29     -- 버튼 생성
30     button Apply_btn "Apply" width:120 height:25
31     button Del_btn "Delete MeshSmooth" width:120 height:25
32
33     on Apply_btn pressed do                -- Apply 버튼 눌렀을 때 이벤트
34     (
35         sel = selection as array            -- 선택한 오브젝트를 배열로
36         for i in sel do                    -- 차례대로 적용
37             setPoly i
38     )
39
40     on Del_btn pressed do                -- Del 버튼 눌렀을때 이벤트
41     (
42         sel = selection as array
43         for i in sel do
44             deleteMeshSmooth i
45     )
46 )
47 createdialog ApplyMeshSmooth_roll

```

# WireParameter



```
fromObj = $sphere01
toObj = $teapot01

fromParam = fromObj.pos.controller[#X_Position]      -- sphere01의 #x_position subAnim를 대입
toParam = toObj.rotation.controller[#X_rotation]    -- teapot01의 #x_rotation subAnim를 대입

-- wireParameter 적용
paramWire.connect fromParam toParam "x_position/10.0"
```

# Callback을 이용한 Mesh count

callback을 이용하여 선택한 Mesh의 Vertices와 Faces 숫자를 계산하는 스크립트

미리 설치해 놓고 기다리다가 해당 이벤트가 발생하면 정해 놓은 루트를 실행하는 과정

```
59  -- callback 설치
60  callbacks.addScript  #selectionSetChanged  "selectionSetFn() "
```

#selectionSetChanged은 오브젝트가 선택되면 반응하는 Callback 종류이며 선택했을 때와 선택을 해제했을 때 이벤트가 일어납니다. Callback의 두 번째 인자로는 이벤트가 일어났을 때 어떻게 할 것인지 string으로 지정하지만 여기서는 selectionSetFn 함수를 호출하도록 설정해 놓았습니다.

이제 오브젝트를 하나씩 돌면서 Geometry가 아닌 것은 걸러내야 합니다. 그러나 Target Camera 같은 것은 superclass가 GeometryClass이기 때문에 classof해서 Targetobject인지 걸러내야 합니다. 걸러낸 오브젝트들은 objs로 다시 모아집니다.

```
⑨ objs = for i in objs collect
⑩   (
⑪     -- 루프를 돌면서 Geometry가 아닌 것은 필터링
⑫     if superclassof i != GeometryClass do continue
⑬     if classof i == Targetobject do continue
⑭     i
⑮   )
```

아래의 Script 부분을 보면 오브젝트가 선택되어 있지 않을 때와 한 개만 선택되었을 때 다수의 오브젝트가 선택되었을 때의 3부분으로 나누어져 있습니다. 물론 3개 중 한 곳으로 분기되어 처리합니다.

```

1 global select_roll      -- rollout을 global로
2
3 try( destroydialog select_roll ) catch()      -- 만약 다이얼로그가 열려 있다면 닫는다.
4
5 -- selectionSetFn 함수 부분
6 fn selectionSetFn =
7 (
8     objs = selection      -- 선택한 오브젝트 collection
9     objs = for i in objs collect
10     (
11         -- 루프를 돌면서 Geometry가 아닌 것은 필터링
12         if superclassof i != GeometryClass do continue
13         if classof i == Targetobject do continue
14         i
15     )
16
17     if objs.count == 0 then      -- 오브젝트가 선택되어 있지 않을 때
18     (
19         select_roll.objname_la.text = ""
20         select_roll.vert_la.text = "Vertices: "
21         select_roll.face_la.text = "Faces: "

```

```

22     )
23     else if objs.count == 1 then          -- 한 개만 선택되어 있을 때
24     (
25         select_roll.objname_la.text = objs[1].name
26         select_roll.vert_la.text = "Vertices: " + (objs[1].mesh.verts.count
as string)
27         select_roll.face_la.text = "Faces: " + (objs[1].mesh.faces.count as
string)
28     )
29
30     else          -- 한 개 이상 선택되었을 때
31     (
32         Tvert = 0
33         Tface = 0
34         select_roll.objname_la.text = (objs.count as string) + " objects"
35
36         -- vert와 face 숫자를 센 다음 합친다.
37         for i in objs do
38         (
39             Tvert += i.mesh.verts.count
40             Tface += i.mesh.faces.count
41         )
42
43         -- 인터페이스로 출력
44         select_roll.vert_la.text = "Vertices: " + (Tvert as string)
45         select_roll.face_la.text = "Faces: " + (Tface as string)
46     )
47 )

```

```

50 rollout select_roll "Mesh Count"           -- rollout 부분
51 (
52     label objname_la ""
53     label vert_la  "Vertices:"    offset:[0,10]
54     label face_la  "Faces:"
55
56
57     on select_roll open do                 -- rollout이 오픈되었을 때 이벤트
58     (
59         -- callback 설치
60         callbacks.addScript  #selectionSetChanged  "selectionSetFn()"
61     )
62
63     on select_roll close do                 -- rollout이 Close 되었을 때 이벤트
64     (
65         -- callback 제거
66         callbacks.removeScripts()
67     )
68 )
69 createdialog select_roll  height:80        -- 다이얼로그 생성

```



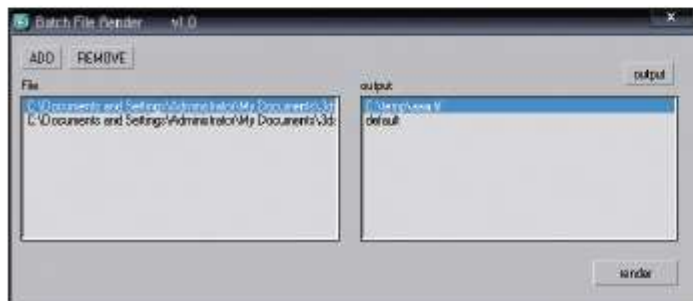
# Batch File Render

〈Batch Render〉와 〈Batch File Renderer〉가 다른 점이 있다면, 〈Batch Render〉는 한 Scene 안에서 렌더링을 실행하지만 〈Batch File Render〉 Script는 파일 단위로 렌더링을 걸어주는 Script라는 것입니다.

이 〈Batch File Render〉 Script를 사용하기 위해서는 몇 가지 규칙이 있습니다.

- 렌더링되는 3ds Max 파일이 저장되어 있어야 합니다.
- Camera01 이름의 카메라가 Scene 안에 반드시 있어야 합니다.
- 3ds Max 파일을 오픈했을 때 에러가 없어야 합니다. 만약에 파일 오픈할 때 에러가 있다면 에러 창 때문에 렌더링이 중지될 수도 있습니다(예, 플러그인 로딩 오류 등).

〈Batch File Render〉의 인터페이스는 다음과 같습니다.



```

❶ -- Rollout 생성
❷ rollout batchFileRender_rollout "Batch File Render"      v1.0"
❸ (
❹     local files = #(), outputs = #()
❺
❻     -- UI controls 생성
❼     button add_btn "ADD" pos:[10,10]
❽     button remove_btn "REMOVE" pos:[55,10]
❾     listbox file_lb "File" pos:[10,40] width:300 item:files
❿     listbox output_lb "output" pos:[330,40] width:300
⓫     button output_btn "output" pos:[580,25]
⓬     button render_btn "render" pos:[550,210] width:80 height:25
⓭
⓮
⓯     -----function-----
⓰     -- add_file 함수
⓱     fn add_file file =
⓲     (
⓳         append files file
⓴         file_lb.items = files
⓵
⓶         append outputs "default"
⓷         output_lb.items = outputs
⓸     )

```

```

26  -- removeList 함수
27  fn removeList  index =
28  (
29      if index == 0 do return undefined
30      deleteitem files index
31      deleteitem outputs index
32
33      file_lb.items = files
34      output_lb.items = outputs
35  )
36
37  -- syncList 함수
38  fn syncList  index =
39  (
40      output_lb.selection = index
41      file_lb.selection = index
42  )
43
44  -- outputList 함수
45  fn outputList  file =
46  (
47      index = output_lb.selection
48      if index == 0 do return undefined
49
50      outputs[index] = file
51      output_lb.items = outputs
52  )

```

```

54  -- renderFn 함수
55  fn renderFn =
56  (
57      for i in 1 to files.count do
58      (
59          loadMaxFile files[i] quiet:true
60          cam = $camera01
61          if cam == undefined do continue
62          viewport.setCamera cam
63
64          if outputs[i] != "default" do
65          (
66              rendSaveFile = true
67              rendOutputFilename = outputs[i]
68          )
69          renderSceneDialog.update()
70          max quick render
71      )
72  )  -- renderFn 함수의 끝부분

```

```

76 ----- events -----
77 -- file_lb listbox 선택 시 이벤트
78 on file_lb selected index do
79 (
80     syncList index          -- syncList 함수 호출
81 )
82 -- output_lb listbox 선택 시 이벤트
83 on output_lb selected index do
84 (
85     syncList index
86 )
87
88 -- add 버튼을 눌렀을 때 이벤트
89 on add_btn pressed do
90 (
91     file = getOpenFileName types:"max(*.max)|*.max|All|*.*|"
92     if file == undefined do return undefined
93
94     add_file file          -- add_file 함수 호출
95
96 )

```

```

98      -- remove 버튼을 눌렀을 때 이벤트
99      on remove_btn pressed do
100      (
101          index = file_lb.selection
102          removeList index          -- removeList 함수 호출
103      )
104
105      -- output 버튼을 눌렀을 때 이벤트
106      on output_btn pressed do
107      (
108          file = getSaveFileName types:"tif(*.tif)|*.tif|bmp(*.bmp)|*.
109      bmp|All|*.*|"
110          if file == undefined do return undefined
111
112          outputList file          -- outputList 함수 호출
113      )
114
115      -- render 버튼을 눌렀을 때 이벤트
116      on render_btn pressed do
117      (
118          renderFn()              -- renderFn 함수 호출
119      )
120  )
121  createdialog batchFileRender rollout width:650 height:250

```