

POV-Ray 入門その 11

情報メディア学科 佐藤尚

このドキュメントは、<http://www.sato-lab.jp/ml2/11.pdf> からダウンロードすることができます。

1. この演習で体験すること

前回に引き続きアニメーションに挑戦します。また、マクロと呼ばれる機能の紹介を行います。

2. 今回使用する教材

教科書(POV-Rayによる3次元CG制作)と配布資料。movファイルが再生できない人は、何かのプレイヤーをダウンロードして、利用してみてください。例えば、VLC media player (<http://www.videolan.org/index.ja.html>)などをインストールしてみてください。

3. 演習課題 11-1 PeerReview 問題

<http://www.sato-lab.jp/ml2/ex11-1> で指定された4枚の画像と4本の動画を見て、評価を行ってください。評価項目は以下のようになっています。【問題番号 11-1】

図 1～図 4 問題番号 10-2b で作成した画像です。	動画 1～動画 4 問題番号 10-4 で作成した動画です。
きちんと画像が表示されるか？	きちんと動画を再生することができるか？
8 個の節が円周上に配置されている形状が表示されているか？ ポンドリングみたいな形	球形状が円周上を移動しているような動きとなっているか？ 球形状の回転の方向は反時計回りにとなっているか？

4. 演習課題 11-2(マクロ)

#declare 命令を利用することで、C言語や Processing における変数を利用することができることを学びました。例えば、以下のように COFFEECUP を定義すると、コーヒーカップのような形状を表示することができます。

```
1 #declare COFFEECUP = object{
2   union{
3     difference {
4       union{
5         object { Disk_Y
6           scale <2.0, 1.1*2, 2.0> } // 本体
7         object{ torus{ 1, 0.3}
8           rotate 90*x
9           translate <2,0+0.0,0>} // 取っ手
10      }
11    }
12  }
```

```

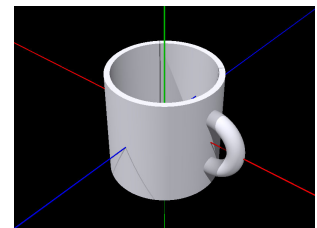
11     object { Disk_Y
12         scale <1.8, 1.*2, 1.8>
13         translate <0.0, 0.8, 0.0> } // 本体をくり抜くための円柱   }
14     }
15     pigment{color Quartz}
16 }

```

この場合、高さの異なるコーヒーカップを表示するためには、scale 命令を利用して拡大・縮小をすることになります。しかし、大きさは変わりますが、取っ手の太さも変わってしまうなど、不自然な形となってしまいます。単純に拡大・縮小だけではなく、一部の大きさなどを変化させたい場合には、#declare 命令ではなく #macro 命令を使うと上手く行く場合があります。#macro 命令は次のようになっています。#macro 命令では C 言語や Processing の関数のように引数を与えることができます。引数は C 言語や Processing 言語の関数の場合と同様に、#macro~#end の間で、自由に利用することが出来ます。なお、povray では、様々な変数やマクロが定義されていますので、既存のものと同じ名前にならないように、マクロ名や引数をすべて大文字としています。

マクロの書式
<pre> #macro マクロ名(引数, ...) 処理したい内容や形状の定義 #end </pre>
マクロの利用例
<pre> #macro COFFEE_CUP(HEIGHT) difference{ // 本体と取っ手 union{ object{Disk_Y scale <2.0,HEIGHT*2.0,2.0>} object{torus {1.0,0.3} rotate 90*x translate <2.0,HEIGHT-1.0,0.0>} } // 本体をくり抜くための円柱 object{ Disk_Y scale <1.8, HEIGHT*2.0,1.8> translate <0.0,0.8,0.0>} } #end </pre>

このマクロを利用して、コーヒーカップを表示するシーンファイルを載せておきます。「COEFFEE_CUP(1)」のように利用します。ただ、この定義では、引数 HEIGHT に少し小さな値を指定すると取っ手が本体から飛び出てしまい、ちょっと格好の悪いことになります。もう少し調整が必要です。



```

1 #include "colors.inc"
2 #include "shapes.inc"
3 camera {

```

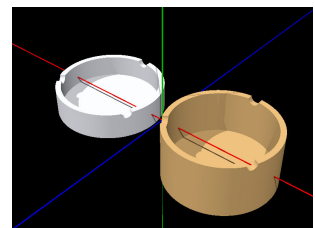
```

4      location <10.0, 12.0, -12.0>
5      look_at <0.0, 0.0, 0.0>
6      angle 30
7  }
8  light_source {
9      < 10, 20, -10>
10     color White
11     parallel point_at <0, 0, 0>
12 }
13 light_source {
14     < -10, 20, -10>
15     color White
16     parallel point_at <0, 0, 0>
17 }
18
19 object { Cylinder_X scale 0.02 pigment{color Red}}
20 object { Cylinder_Y scale 0.02 pigment{color Green}}
21 object { Cylinder_Z scale 0.02 pigment{color Blue}}
22
23 #macro COFFEE_CUP(HEIGHT)
24 difference{
25     union{
26         object{Disk_Y scale <2.0,HEIGHT*2.0,2.0>}
27         object{torus {1.0,0.3}
28             rotate 90*x translate <2.0,HEIGHT-1.0,0.0>}
29     }
30 }
31 object{ Disk_Y
32     scale <1.8, HEIGHT*2.0,1.8> translate <0.0,0.8,0.0>
33 }
34 }
35 #end
36
37 object{
38     COFFEE_CUP(1)
39     pigment{color Quartz}
40 }

```

i. 上のシーンファイルで COFFEE_CUP マクロを利用して、高さの異なる 2 つのコーヒーカップを表示するようなシーンファイルを作成して下さい。レンダリングを行って出来上がった画像をキャリアポートフォリオにアップして下さい。コーヒーカップの色は各自自由に決めて下さい。テクスチャなどを利用したり、コーヒーカップがテーブルの上に置かれているようにするなどの工夫をして表示することは大歓迎です。【問題番号 11-2a】

ii. 下のシーンファイルは深さの異なる 2 つの灰皿を表示するシーンファイルです。AshTray マクロの空欄を埋めて、きちんとレンダリングが出来るようにして下さい。AshTray マクロは高さが HEIGHT の灰皿形状を定義するマクロです。キャリアポートフォリオ上で解答して下さい。【問題番号 11-2b】



未完成のシーンファイル

```

1  #include "colors.inc"
2  #include "shapes.inc"
3  camera {
4      location <10.0, 12.0, -12.0>
5      look_at <0.0, 0.0, 0.0>
6      angle 30
7  }
8  light_source {
9      < 10, 20, -10>
10     color White
11     parallel point_at <0, 0, 0>
12 }
13 light_source {
14     < -10, 20, -10>
15     color White
16     parallel point_at <0, 0, 0>
17 }
18
19 object { Cylinder_X scale 0.02 pigment{color Red}}
20 object { Cylinder_Y scale 0.02 pigment{color Green}}
21 object { Cylinder_Z scale 0.02 pigment{color Blue}}
22
23 #macro AshTray(HEIGHT)
24 difference {
25     difference {
26         object { Disk_Y scale <2.0, __(a)__, 2.0> } // 本体
27         object { Disk_Y scale <1.8, __(b)__, 1.8>
28             translate <0.0, 0.2, 0.0> } // 内側
29         scale <1, __(c)__, 1> // 本体の高さを HEIGHT にする
30     }
31     union {
32         object {Disk_Z scale <0.2,0.2,1> translate <0,0,2>} // 切込
33         object {Disk_Z scale <0.2,0.2,1>
34             translate <0,0,2> rotate __(d)*y} // 切込
35         object {Disk_Z scale <0.2,0.2,1>
36             translate <0,0,2> rotate 240*y} // 切込
37         translate <0, __(e)__, 0> // 切込の位置を移動
38     }
39 }
40 #end
41
42 // 灰皿の配置
43 object { AshTray(1) // 標準の灰皿
44     translate <-2.5,0,0>
45     pigment { color Silver }
46 }
47 object { AshTray(2) // 深い灰皿
48     translate <2.5,0,0>
49     pigment { color Gold }
50 }

```

実は、マクロ内のみで有効となる変数を利用することが出来ます。#declare を使って変数を宣言するとシーンファイル全体で利用できる変数（グローバル変数）となってしまいます。マクロ内のみで有効となる変数を利用するためには、#local 命令を利用します。#local 命令の利用の仕方は、

#declare 命令と同じです。#local 命令を利用して上のシーンファイルを書き換えると、以下のようになります。AshTray マクロの部分だけを載せてあります。

```
1 #macro AshTray(HEIGHT)
2 #local SEC = object {Disk_Z scale <0.2,0.2,1> translate <0,0,2>};
3 difference {
4   difference {
5     object { Disk_Y scale <2.0, (a), 2.0> }
6     object { Disk_Y scale <1.8, (b), 1.8> translate <0.0, 0.2, 0.0>
7   }
8   scale <1,(c),1>
9 }
10 union {
11   object {SEC}
12   object {SEC rotate (d)*y}
13   object {SEC rotate 240*y}
14   translate <0,(e),0>
15 }
16 }
#end
```

5. 演習課題 11-3(複合動作のアニメーション)

- i. 教科書の pp.132~pp.133 を読んで下さい。132 ページのシーンファイル 8-3 を利用すると、トラスが円柱状に沿って落ちていくようなアニメーションが出来上がることを確認して下さい。【問題番号 11-3a】
- ii. 教科書の pp.133~135 を読んで、物体の位置だけでなく、カメラの向きや、色を変える方法を理解して下さい。特に、8-1-6「色の変化のアニメーション」に書かれている変更を、シーンファイル 8-1 に加え、色の変化するアニメーションが出来ることを確認して下さい。このアニメーションの動画ファイルを作成し、キャリアポートフォリオにアップロードして下さい。【問題番号 11-3b】
- iii. 演習問題 11-2 で定義した COFFEE_CUP マクロや AshTray マクロを利用して、大きさの異なる複数の物体が動いているアニメーションを作成して下さい。コーヒーカップや灰皿以外の物体も表示されていてもかまいません。【問題番号 11-3c】