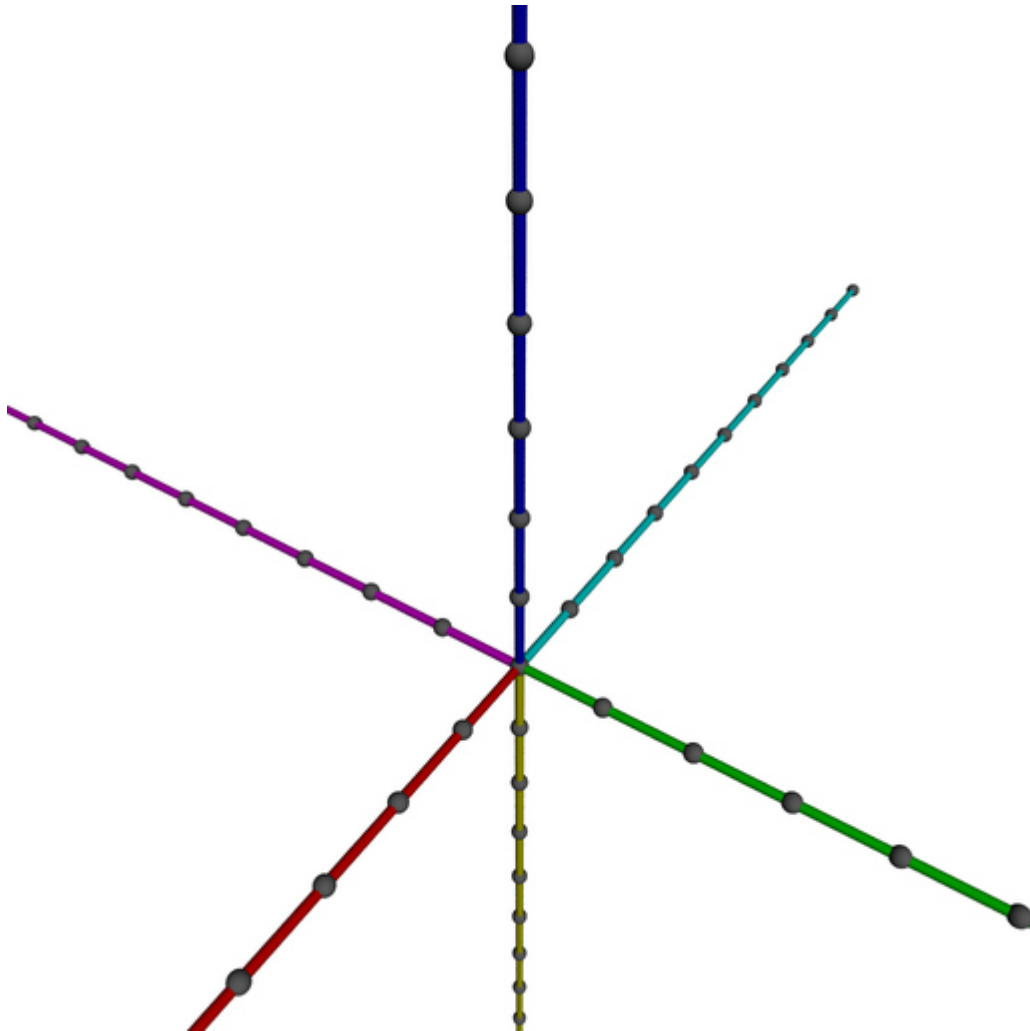


おまけ

- この資料は授業とは関係ないので、参考程度に読んでください
- 解説はしません

POV-Ray

- POV-Rayは他の3DCGソフトとは異なる異質な3DCGソフト
- POV-Ray独自のプログラミング言語でCGモデルを作る
- リアルタイムCGではないため、ゲーム制作は出来ない
- デフォルトではobjファイルは出力できない(プラグインで出来る)
- 左手系
- ここでは、右手系として使う場合に便利なサンプルソースコードを紹介する
- サンプルソースコードの実行結果では、赤・緑・青がそれぞれx軸・y軸・z軸を表す



```

// coordinate

camera {
  location <6, 4, 8>
  right <0, 1, 0>
  up <0, 0, -1>
  sky <0, 0, 1>
  look_at <0, 0, 2>
}

light_source {
  <60, 80, 100>
  color rgb <1, 1, 1>
  shadowless
}

background {
  color rgb <1, 1, 1>
}

// axis

object {
  cylinder { <0, 0, 0>, <10, 0, 0>, 0.05 }
  pigment { rgb <1, 0, 0> }
}

object {
  cylinder { <0, 0, 0>, <-10, 0, 0>, 0.05 }
  pigment { rgb <0, 1, 1> }
}

object {
  cylinder { <0, 0, 0>, <0, 10, 0>, 0.05 }
  pigment { rgb <0, 1, 0> }
}

object {
  cylinder { <0, 0, 0>, <0, -10, 0>, 0.05 }
  pigment { rgb <1, 0, 1> }
}

object {
  cylinder { <0, 0, 0>, <0, 0, 10>, 0.05 }
  pigment { rgb <0, 0, 1> }
}

object {
  cylinder { <0, 0, 0>, <0, 0, -10>, 0.05 }
  pigment { rgb <1, 1, 0> }
}

// node

#declare nodePos = -10;
#while ( nodePos <= 10 )
  object {
    sphere { <0, 0, 0>, 0.1 }
    pigment { rgb <0.5, 0.5, 0.5> }
    translate <nodePos, 0, 0>
  }

  object {
    sphere { <0, 0, 0>, 0.1 }
    pigment { rgb <0.5, 0.5, 0.5> }
    translate <0, nodePos, 0>
  }

  object {
    sphere { <0, 0, 0>, 0.1 }
    pigment { rgb <0.5, 0.5, 0.5> }
    translate <0, 0, nodePos>
  }

  #declare nodePos = nodePos + 1;
#end

```