

---

# コンピュータグラフィックス2023

---

情報工学実験第1A/B

2023年度春学期

月曜日2～4限

担当教員: 藤代 一成

TA: 進士 さくら・生井 麻結(藤代研M1)

{ifujishiro, s.shinji1227, nmimy2000}@keio.jp

<https://fj.ics.keio.ac.jp/>

# 配布資料

---

- ① 解説ビデオ (CGjikken2023.mp4)
- ② ビデオスライド (CGjikken2023\_videoslides.pdf)
- ③ POV-Ray実験書 (CGjikken2023\_POV-Ray.pdf)
- ④ 実験ハンドアウト (CGjikken2023\_handout.pdf)
- ⑤ 基本課題シーンファイル (blockworld.pov)
- ⑥ 応用課題シーンファイル (POV-Ray4AP3.zip)

※すべてK-LMSから閲覧・ダウンロード可

# 実験スケジュール

- 実験前

- ✓ 解説ビデオ視聴
- ✓ POV-Rayの個人PCへのインストール

- 1週目 @32-205

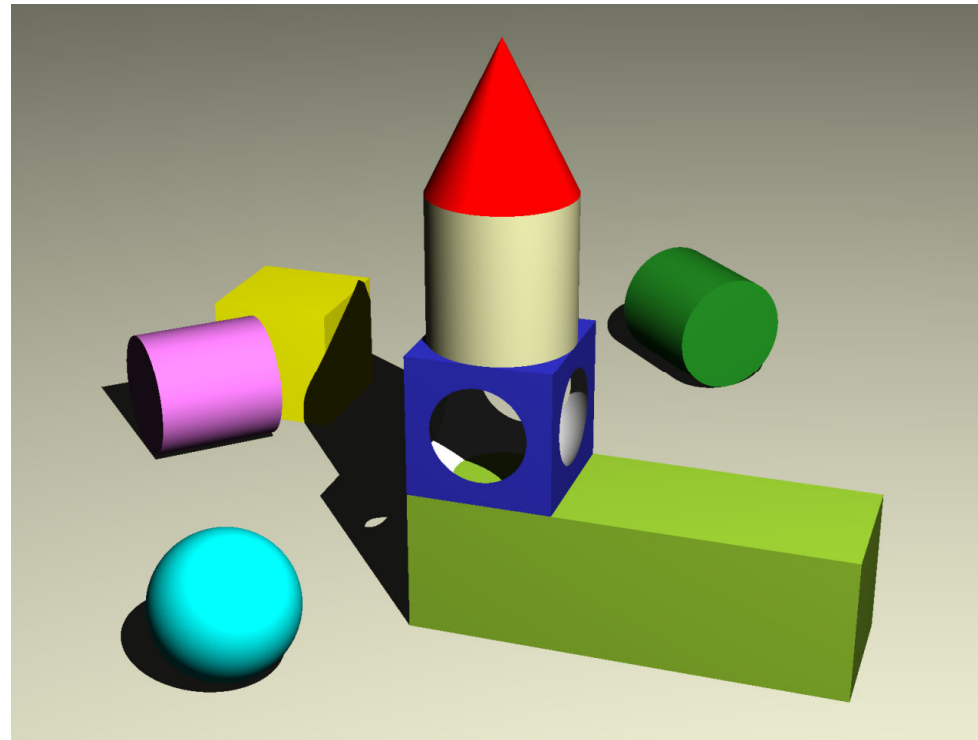
- ✓ 2限: 基本課題(藤代 & TA): サンプルシーンの書換えによる視覚効果の確認
- ✓ 3限: 応用課題(藤代 & TA): POV-Rayプログラミング3題
- ✓ 4限: 発展課題(TA & 藤代): テーマ選択とシーン設計(実測・資料探索)  
※2テーマの発表 & 討論時限速報@B3用slack

- 2週目 @32-205

- ✓ 前日(日曜)午後6時: 発展課題に関するレポート提出
- ✓ 当日(月曜)午前9時: 電子マテリアル(POV-Rayソース・プレゼン資料)提出
- ✓ 2・3限: テーマごとに提出作品に関する発表 & 討論(藤代 & TA)
- ✓ 実験アンケート

# 基本課題：積み木の世界

下図のような積み木の世界([blockworld.pov](#))に対して, 次を示す10課題(+Option)に答えよ.



# 基本課題: シーン定義

```
#include "colors.inc"
#include "shapes.inc"
#include "woods.inc"
#include "metals.inc"
#include "stones.inc"
#include "textures.inc"

camera{
  location <-10,10,-20>
  look_at<0,0.5,0>
  angle 20
}

light_source{<-3,20,-20> color 2*White}
```

```
object{
  Plane_XZ
  pigment{ color Wheat}
  translate<0,-3,0>
}

union{
  object{
    cylinder{<0,0,0>,<0,0,-2>,1}
    pigment{color MediumGoldenrod}
    rotate 90*x
    translate<0,1,0>
  }

  object{
    Cone_Y
    translate <0,4,0>
    pigment{color Red}
  }

  difference{
    object{
      Cube
      pigment{MediumBlue}
      rotate 45*y
    }
  }
}
```

```
object{
  Sphere
  pigment{color White}
  scale 1.2
}

object{
  Sphere
  pigment{color Cyan}
  translate<-4,-2,0>
}

object{
  Cube
  pigment{YellowGreen}
  rotate 45*y
  translate<0,-2,0>
}

object{
  box{<1,-4,-1>,<5,-1,1>}
  pigment{YellowGreen}
  rotate 45*y
}
```

```
object{
  Cube
  pigment{color Yellow}
  rotate 45*y
  translate<0,-2,8>
}

object{
  Disk_X
  pigment{Orchid}
  translate<-2,-2,7>
}

object{
  Disk_Z
  pigment{color
ForestGreen}
  translate<7,-2,6>
}
```

# 基本課題：シーンファイルの書換え

1. **【基本】**基本シーンファイル blockworld.povを各自の適当な作業用フォルダにダウンロード・実行し対応画像blockworld.pngの出力を確認せよ。解像度は試しに[1024 × 768, No AA]に設定。
2. **【カラー】**中央に見える尖塔のカラーを, color.incで定義される別の色に変更せよ。
3. **【テクスチャ】**適当なオブジェクトの見えを, インクルードされている複数のテクスチャファイルで定義されている木・金属・石の各素材に変更せよ。
4. **【回転/立体定義】**左手に見えるピンクの円柱を, 底面が視点の方向に向くように回転せよ。
5. **【平行移動/CSG】**中央の尖塔とその直下にある円柱・球形の穴の開いた青の立方体を同時に黄緑の直方体のベース中央に移動せよ  
    **ヒント: 移動対象はCSGで定義されていることを利用**
6. **【拡大・縮小/CSG】**尖塔下の青の立体に開いている穴を大きくせよ。
7. **【立体定義】**画面右奥にある緑の円柱前の空いているスペースに, ガラス状の角丸直方体を超楕円体を用いてモデリングし, 配置せよ。  
    **ヒント: ガラステクスチャと超楕円体は, 各々日本語マニュアルの14.4と11.1-7を参考。  
    配布資料にも関連記述有**

# 基本課題(続き)

8. 【光源位置】尖塔から伸びている影を黄色の立方体にかからないように右側にずらせ.  
無論, 影の形状は変化してよい.
9. 【視線制御】黄緑の台の手前に立ち, 尖塔を仰ぎ見るような画像に変更せよ.
10. 【地面と空】チェッカーボード状の地面と雲のある空の背景を追加せよ  
ヒント: 市松模様のテクスチャと空は, 各々日本語マニュアル13.1-7と7.1等を参考.  
当然, 視点は空も見上げられる位置に変更すべき

註) 上記の課題は各々独立

オプション) 【アニメーション】課題10のシーン全体を上空から俯瞰し旋回するフライトアニメーションを作成せよ.

ヒント:  $[0, 1]$ の間を動くclock変数を用いる. メニューRenderの下にあるEdit Settings/ Renderの設定画面でCommand line optionsに, 例えば「Initial\_Frame=0 Final\_Frame=10」と指定して, シーンファイルを実行すると, 同一フォルダ内にclock変数を0.1刻みで0から1まで変化させながら, 計11枚の画像(通番が付加されたファイル名)が生成できるので, それらを順次適切なアニメーション作成ツールに読み込んで, アニメーションファイルに変換する.

# 実験スケジュール

- 実験前

- ✓ 解説ビデオ視聴
- ✓ POV-Rayの個人PCへのインストール

- 1週目 @32-205

- ✓ 2限: 基本課題(藤代 & TA): サンプルシーンの書換えによる視覚効果の確認
- ✓ 3限: 応用課題(藤代 & TA): POV-Rayプログラミング3題
- ✓ 4限: 発展課題(TA & 藤代): テーマ選択とシーン設計(実測・資料探索)  
※2テーマの発表 & 討論時限速報@B3用slack

- 2週目 @32-205

- ✓ 前日(日曜)午後6時: 発展課題に関するレポート提出
- ✓ 当日(月曜)午前9時: 電子マテリアル(POV-Rayソース・プレゼン資料)提出
- ✓ 2・3限: テーマごとに提出作品に関する発表 & 討論(藤代 & TA)
- ✓ 実験アンケート



# 応用課題

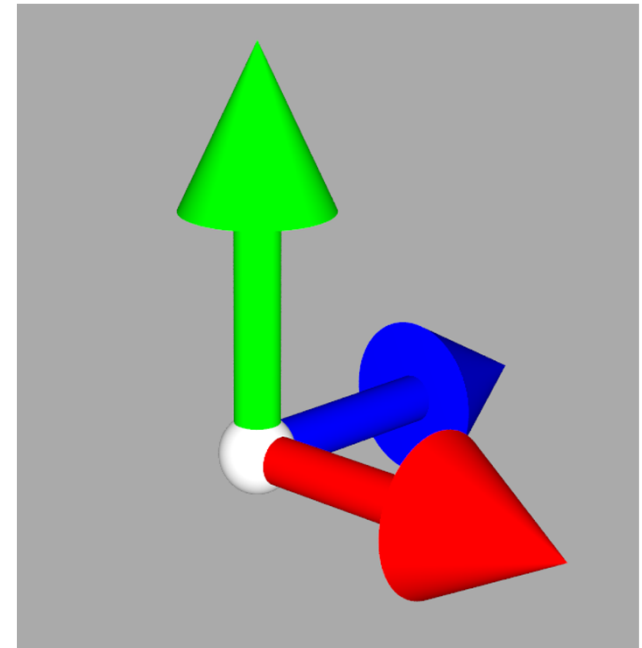
---

- **課題A**: 座標軸矢印アイコン【藤代】
  - ✓ CSGとマクロ定義
- **課題B**: 螺旋階段【生井】
  - ✓ 反復処理
- **課題C**: グローバル照明【進士】
  - ✓ フォトンマッピング

# 課題A: 座標軸矢印アイコン

**狙い:** POV-Rayのモデリングを効果的に進めるにはシーン内に配置する各オブジェクトの位置や大きさを効率的に設定することが必要です。そこで、右図に示すような、座標軸の向きを定義する矢印アイコンを描いてみましょう。

左手座標系の空間に配置するので、色の割当てはx軸をR, y軸をG, z軸をBとしました。矢印の根元には白の小球を置いたことにも注意してください。



# サンプルシーン (triarrows.pov)

まず、元となる空間に対して、座標軸、カメラ、照明を以下のように設定します。

```
#include "colors.inc"
#include "shapes.inc"

camera{
  location <40,20,-40>
  look_at<0,0,0,0>
  angle 30
}

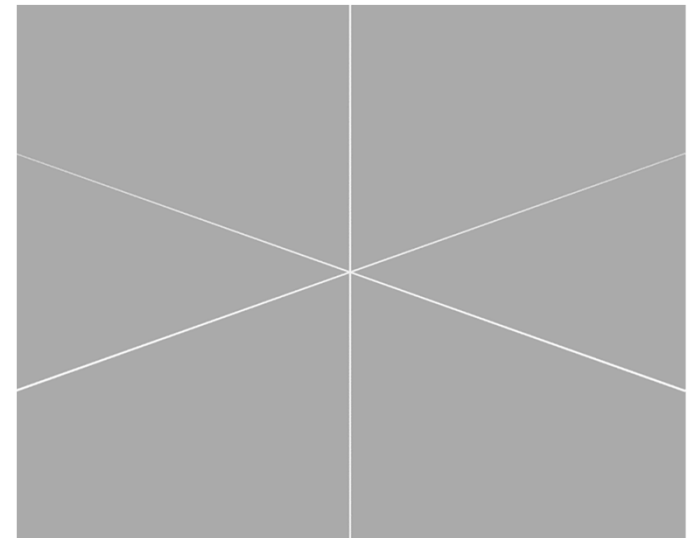
light_source{<40,20,-40> color 2*White}

object{ //x-axis
  cylinder{<-50,0,0>,<50,0,0>,0.05}
  pigment{color White}
}

object{ //y-axis
  cylinder{<0,-50,0>,<0,50,0>,0.05}
  pigment{color White}
}

object{ //z-axis
  cylinder{<0,0,-50>,<0,0,50>,0.05}
  pigment{color White}
}

background{Gray40}
```



# Triarrowsマクロ

点(dx, dy, dz)の位置に矢印アイコンを描くマクロ

```
#macro triarrows(dx,dy,dz)
```

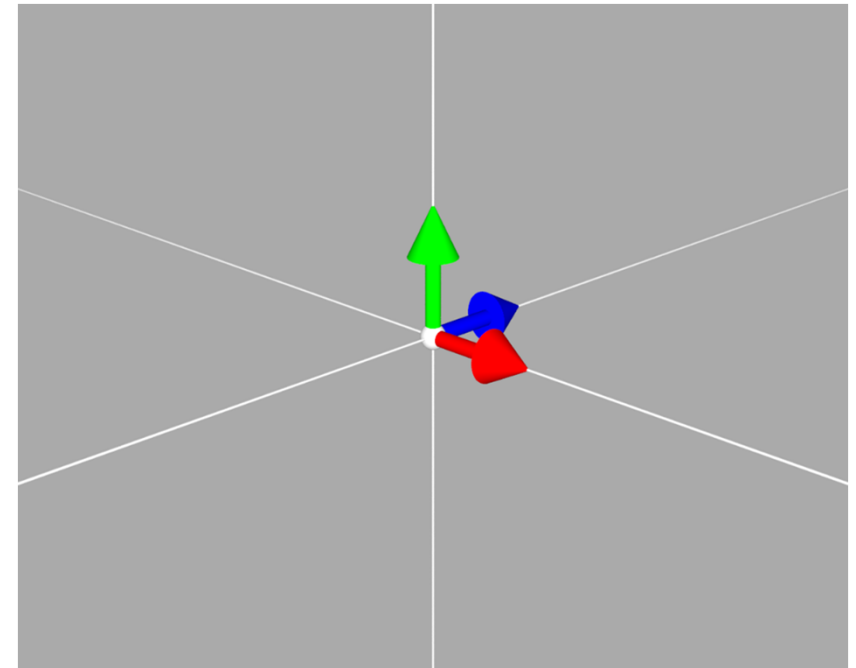
を定義したうえで

```
triarrows(0,0,0)
```

を一行増やし、右図のような結果が得られることを確かめてください。

**ヒント:** 二重にマクロを定義してCSGモデリングしてみてもいい？

**発展:** 矢印や視点位置を変更してみましょう。  
座標軸名を表示してみましょう。

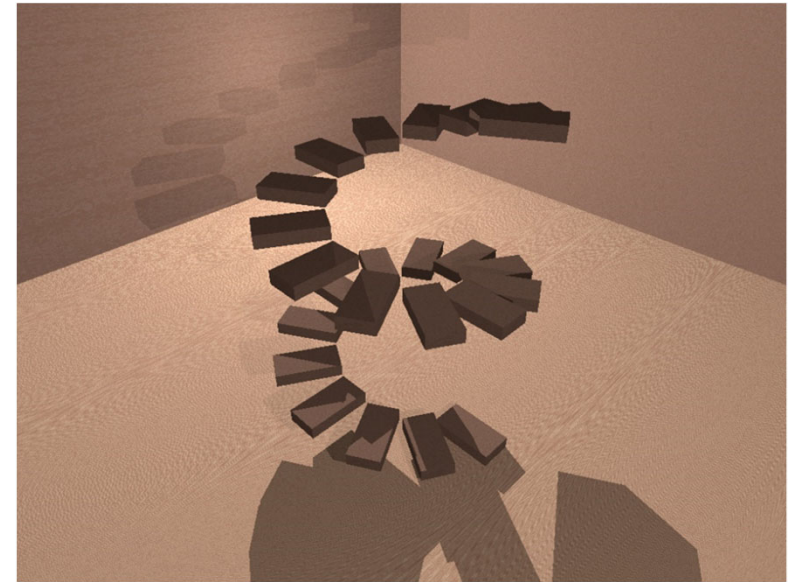


参考：矢印の全体長は5. 先の円錐の半径は1, 高さは1

# 課題B: 螺旋階段

---

**狙い:** オブジェクトの形状や位置を規則的に変化させて複数個配置する場合は、反復処理を用いることで効率よく作業することができます。そこで反復処理を用いて、右図のような螺旋階段を作成してみましょう。POV-Rayにおける幾何学的変換の特性に応じて、正しい順序で移動や回転を行う必要があることに注意してください。



# サンプルシーン(spiral.pov)

螺旋階段の足場となるboxオブジェクトが1つ用意されています。

```
#include "textures.inc"
#include "colors.inc"
#include "Woods.inc"

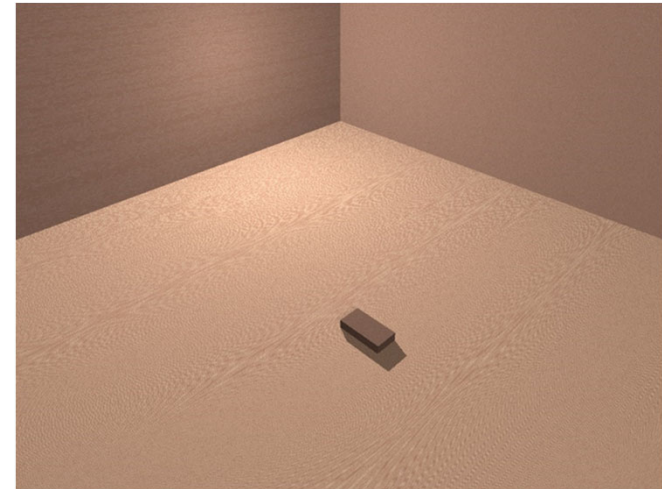
camera {
    location <50, 70, -50>
    look_at <0, 20, 0>
}

light_source {
    <-60, 35, 30>
    color White * 2
}

light_source {
    <60, 50, -30>
    color White * 0.3
}

// -----
object{
    box{
        <0, 0, 0>, <10, 2, 5>
        texture{T_Wood12}
    }
}

// -----
object{
    plane{
        y, 0
        texture {T_Wood1}
    }
}
object{
    plane{
        z, 80
        texture {T_Wood12}
    }
}
object{
    plane{
        x, -80
        texture {T_Wood12}
    }
}
```



# 反復処理

## ループ変数の宣言および更新

```
#declare N = 0;
```

```
#declare N = N+1;
```

## 反復処理

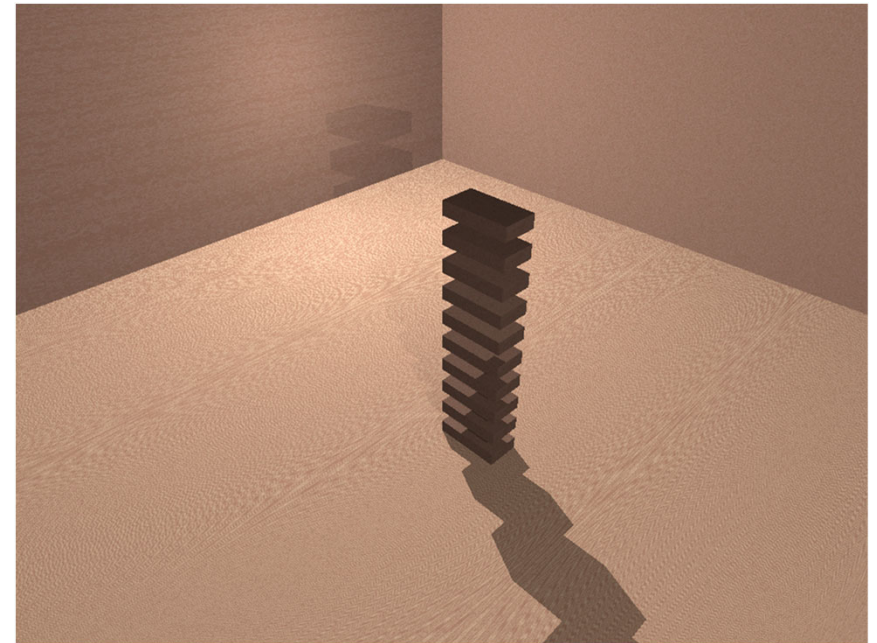
```
#while (ループ変数に関する条件)
```

```
    // 処理
```

```
#end
```

**ヒント:** rotateは原点 $\langle 0, 0, 0 \rangle$ を基準に行われます。

**発展:** 足場の形を変えたり, 手すりを追加してみましょう。

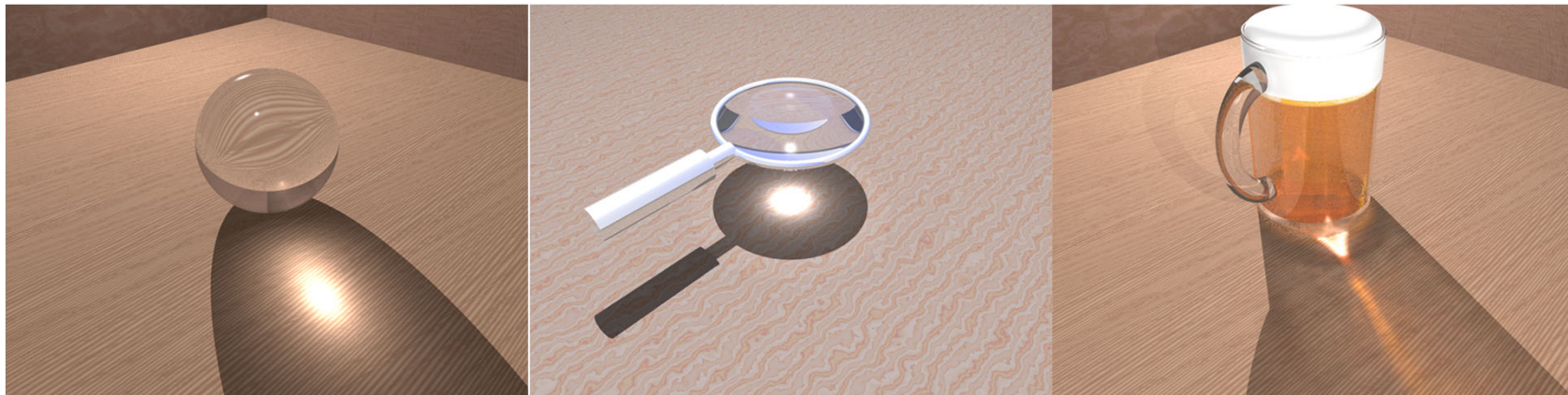


参考 : `translate<0, 4 * N, 0>` で10回反復



# 課題C: グローバル照明

**狙い:** CG作品において正確な光の表現は不可欠です。例えば、金属やガラス等の反射や屈折により光が集まるコースティクス(集光)は、**フォトンマッピング**とよばれる手法で再現できます。そこで、POV-Ray上でフォトンマッピングを設定し、CGらしい光の表現を体験するとともに、マニュアルを見ながらグローバル照明の複雑な設定を行う練習をしてみましょう。



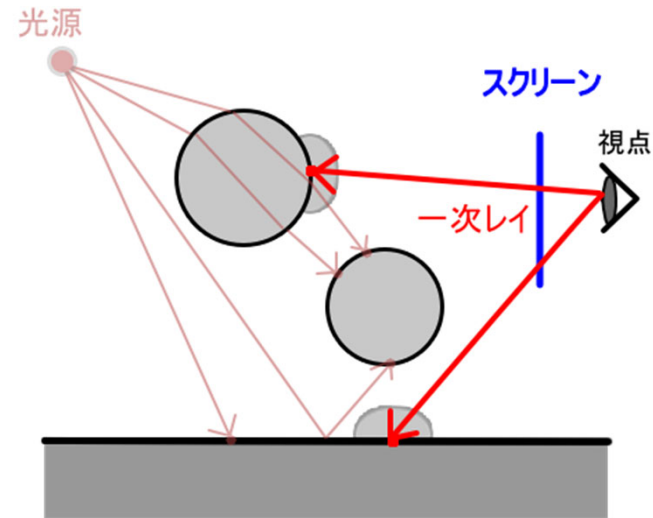
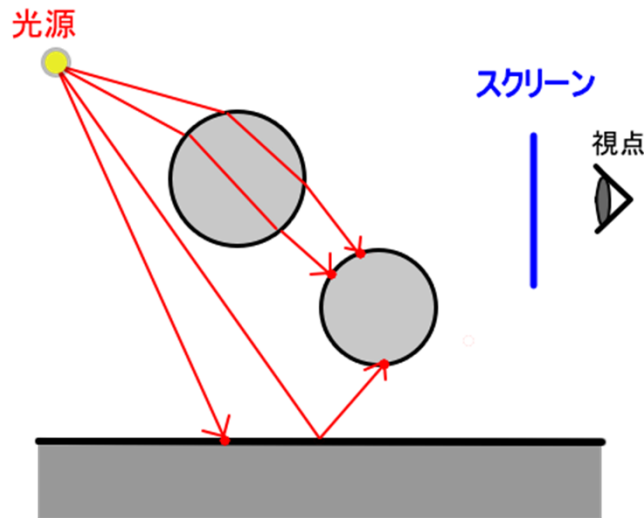


# フォトンマッピング

グローバル照明のアルゴリズムのひとつ

光源からの光を「フォトン(光子)」として大量に飛ばした後、視点から収集していく  
双方向のレンダリング手法†

→コースティクス(集光)が表現可能



† <https://knowledge.shade3d.jp/knowledgebase/フォトンマッピングとは>

# 実装に必要な設定

---

三種類の設定が必要です. ※省略可あり

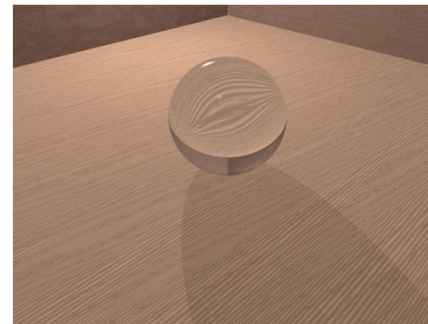
- **グローバル設定** `global settings{...photons{...}}`  
フォトン間隔, フォトン数, 収集半径の指定
- **物体設定** `object{...photons{...}}`  
ターゲット指定, 屈折, 反射, フォトン収集の有効化 or 無効化
- **光源設定** `light_source{...photons{...}}`  
屈折, 反射 の有効化 or 無効化, (面光源かどうか)

# 課題内容

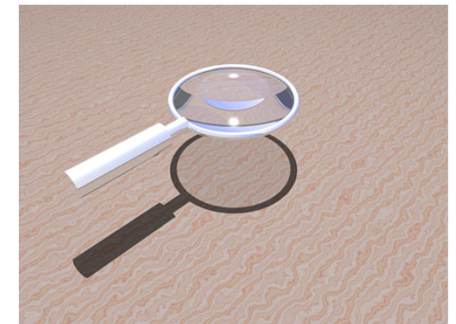
POV-Rayのマニュアル

<http://mana.starfree.jp/povjp/>

のフォトンマッピングの章を参考に  
2種類のシーン:sphere.pov, lens.pov に  
フォトンマッピングを適切に設定して  
集光模様が生じることを確認してください.



sphere.pov



lens.pov

**ヒント:** グローバル設定のパラメタ例 (フォトン間隔: 0.005, フォトン数: 50000, 収束半径: 0.5)

**発展:** beer.povに対しても設定してみてください

(ただし, 物体設定はガラスと液体, 光源設定は奥の光源)

光源を点光源→面光源に変更してみてください.



beer.pov

# 実験スケジュール

- 実験前

- ✓ 解説ビデオ視聴
- ✓ POV-Rayの個人PCへのインストール

- 1週目 @32-205

- ✓ 2限: 基本課題(藤代 & TA): サンプルシーンの書換えによる視覚効果の確認
- ✓ 3限: 応用課題(藤代 & TA): POV-Rayプログラミング3題
- ✓ 4限: 発展課題(TA & 藤代): テーマ選択とシーン設計(実測・資料探索)  
※2テーマの発表 & 討論時限速報 @B3用slack

- 2週目 @32-205

- ✓ 前日(日曜)午後6時: 発展課題に関するレポート提出
- ✓ 当日(月曜)午前9時: 電子マテリアル(POV-Rayソース・プレゼン資料)提出
- ✓ 2・3限: テーマごとに提出作品に関する発表 & 討論(藤代 & TA)
- ✓ 実験アンケート

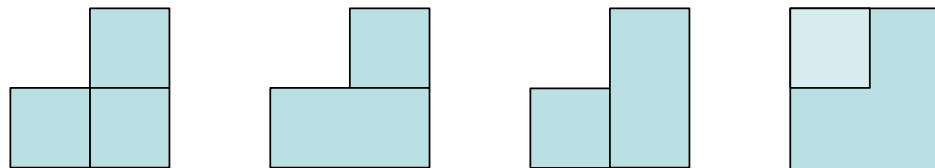
# 発展課題とレポート

【X&Y班】次のいずれかのテーマを選んでシーンを描け.

- 「テーマ1」(進士)
- 「テーマ2」(生井)

提出はK-LMS経由  
前日(日曜)午後6時まで!

- ✓ 標題:「コンピュータグラフィックス」
- ✓ 目的:制作の意図を簡潔に説明
- ✓ モデリング:テーマにそったシーンをどのようにモデリングしたか, CSG木を含め詳説
- ✓ シーン:POV-Rayシーン定義とその解説を含めること
- ✓ 画像:最終画像を題名とともに示すこと
- ✓ 考察:完成度を分析するとともに, 課題の解決法に言及



※CSGは一意的な表現ではない

# 2016年度優秀作品

杉本 学 君



情工実1(CG)

# 2017年度優秀作品

---

大河原 将 君





# 2018年度優秀作品

---



大竹 七勢 君  
「ワインとグラスとその影」

情工実1(CG)



# 2019年度優秀作品『夕暮れを映す鏡』



池田 理和乃 君

# 2020年度優秀作品『消し忘れ』



西舘 祐樹 君

情工実1(CG)

# 2021年度優秀作品『雨上がり』

---



柴崎 史典 君

情工実1(CG)



# 2022年度優秀作品:『自然に溶ける人工物』

---



堀川 健太 君

# 2022年度の捨てがたい作品:『フジシロー開店』

---



風間 涼太郎 君

# 発展課題評価の視点

---

- モチーフ&意匠の巧みさ
- 実物の観察眼の確かさ
- シーン定義の複雑さと妥当性
- 撮影・照明効果によるリアリティの追求
- 【オプション】「立体形状キャプチャ」データの取込み

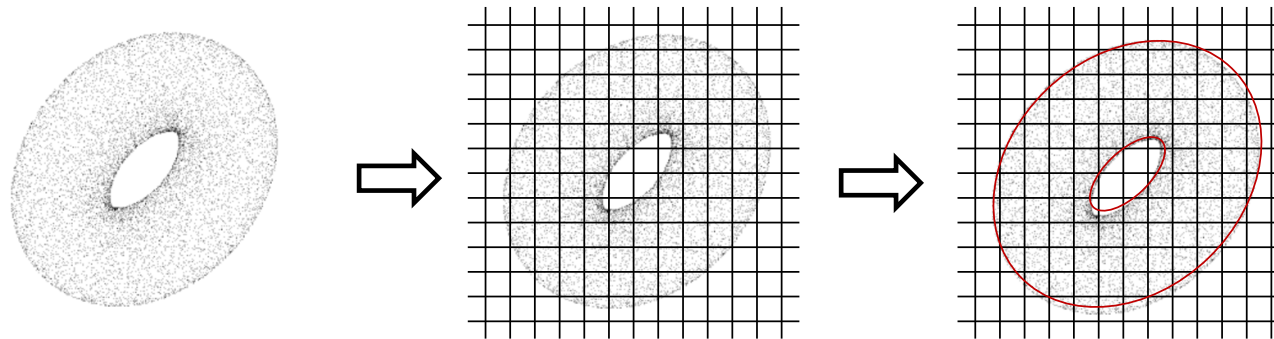
※2014～2022年度の全作品をBOXに掲載中(K-LMSから辿れます)  
<https://keio.box.com/s/rynvevcjdy5sj2zpglcy2ewgzpegem7c>



# オプション: 形状データの取込み

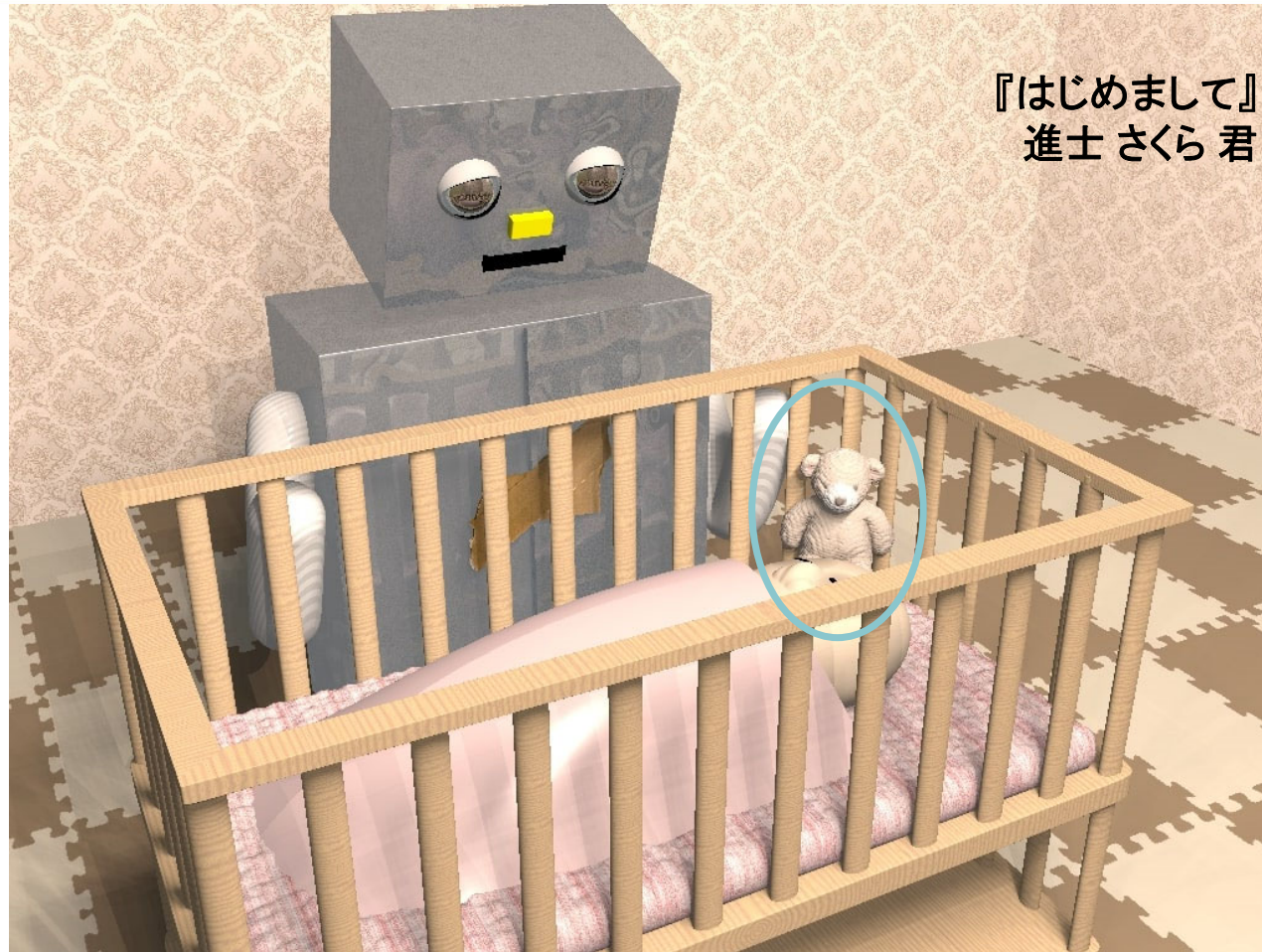
対象: 7~12班

- 方法A: コンバータPoseray (<https://sites.google.com/site/poseray/>) : .obj → .pov
- 方法B: 点群データの取込
  - ✓ B1: 各点を小さな球で表現し, 点群をそのまま出力
  - ✓ B2: 各点を中心とする濃度球を作成, blobを利用して面を張る
  - ✓ B3: 等値面 (isosurface) に変換 (流体表現の常套手段)
    1. 3次元空間を小領域 (ボクセル) に分割
    2. 各ボクセルに含まれる点の個数をカウントし, 全体の個数に対する相対値を計算
    3. 適当な密度値を通過する面 (等値面) を生成





# 形状データ取込み例(2021年度)





# 電子マテリアルの提出

---

- K-LMS経由でzipで圧縮したファイルをアップロード  
(2週目月曜午前中9時まで)
  - ✓ シーンファイルを含む再コンパイルに必要な全ファイルをフォルダにまとめる
  - ✓ 作品画像(.jpgに変換)
  - ✓ プレゼン資料【極力.pptx, .pptで. .pdfも可】
- 注意
  - ✓ フォルダ名, ファイル名は学籍番号
- 翌週までの質問は, B3用slack, K-LMSメール, keio.jpメール経由で

# ディスカッション

- 作品に関するプレゼン(5分)

- ✓ プレゼンは、藤代のPC上で、提出した電子マテリアルを利用
- ✓ プレゼンは制限時間内に収めること(※評価に関係)
- ✓ レポート内容の単なる紹介ではなく、他者から差別化できる点(技術的なこと・意匠的なこと)を強調すること
- ✓ **トップダウン**アプローチ(作品と題名の提示からスタートして、次に制作の細部を説明、そして課題の順に)を意識すること
- ✓ 実験評価＝レポート点＋作品点＋プレゼン点

- 作品に対する講評(5分)

- ✓ TA, 藤代の順で講評
- ✓ 少なくとも1回は他の作品に対する講評・質問を！