

5J「図形処理」ガイダンス

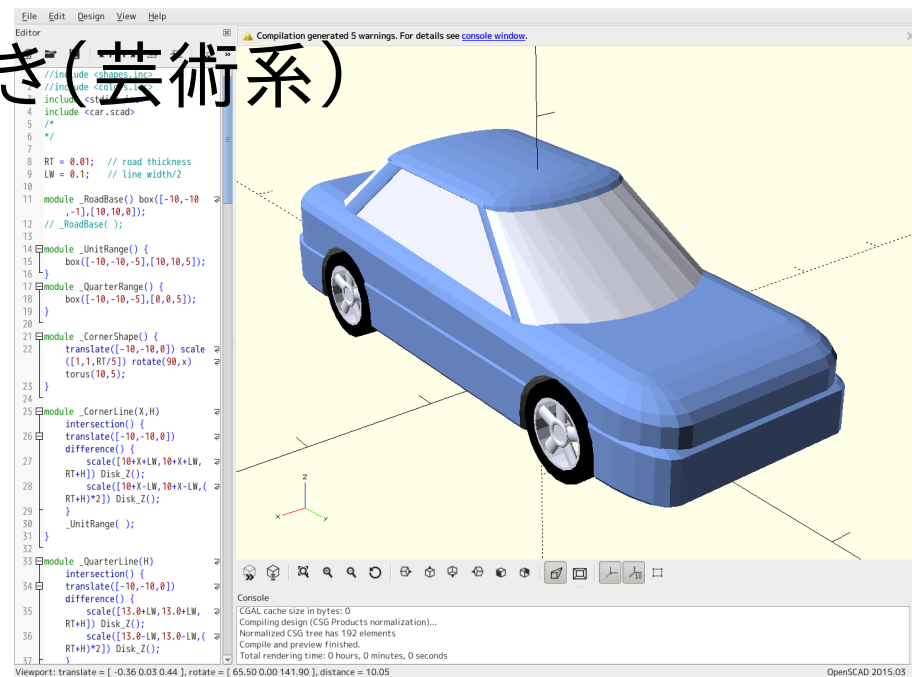
- この授業 = 3DCGの基本技術の学習
- 目次:
 - CGの概要
 - CG技術
 - 授業内容
 - 座標系

CGの概要(1):CGて何?

- =Computer Graphics
 - 又は... CGI=Computer Generated Image
 - コンピュータによる画像生成・自動製図(技術系)
- ≠Computer Drawing
 - コンピュータでのお絵描き(芸術系)

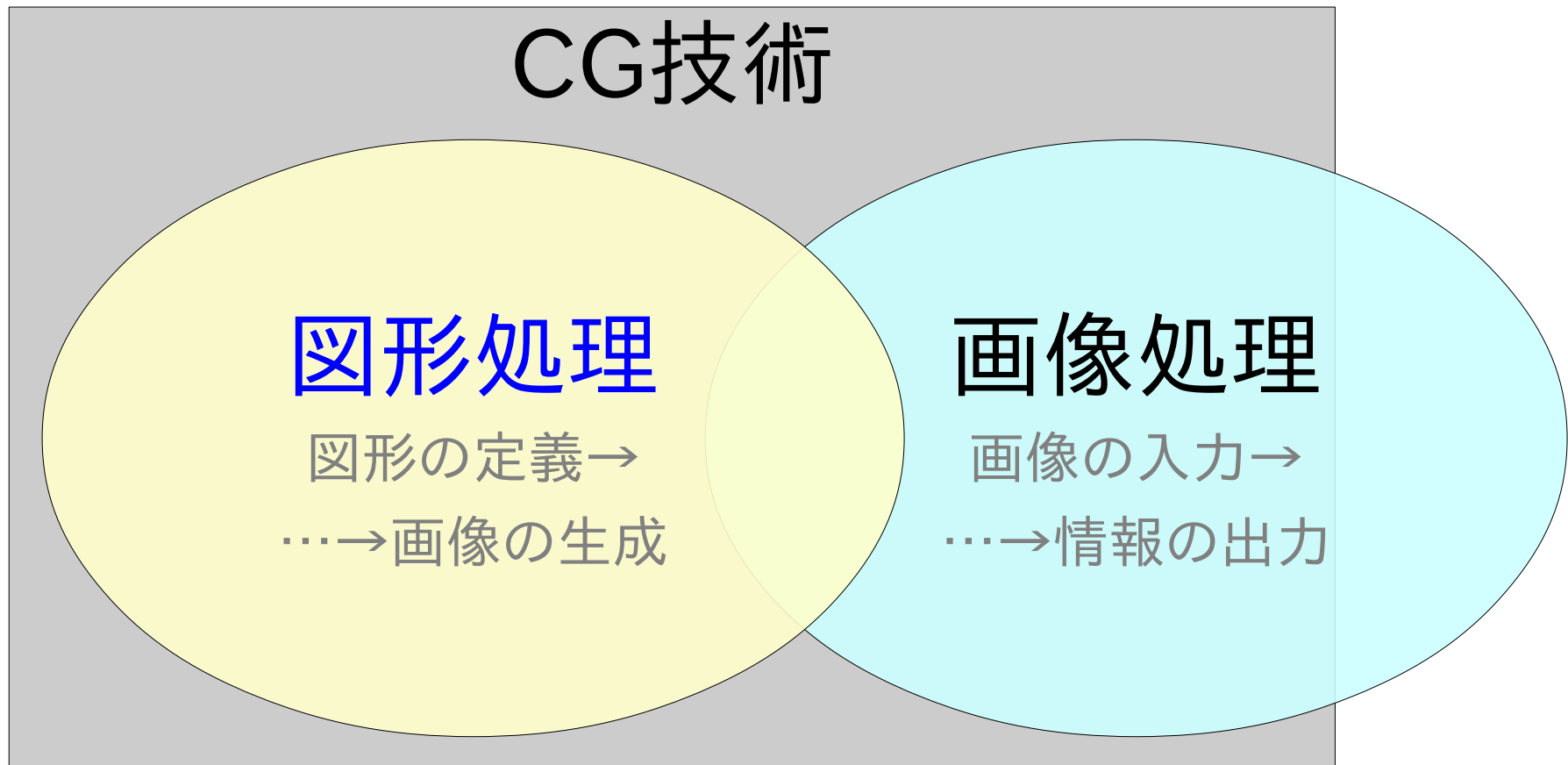


えー?
ちがうのー?



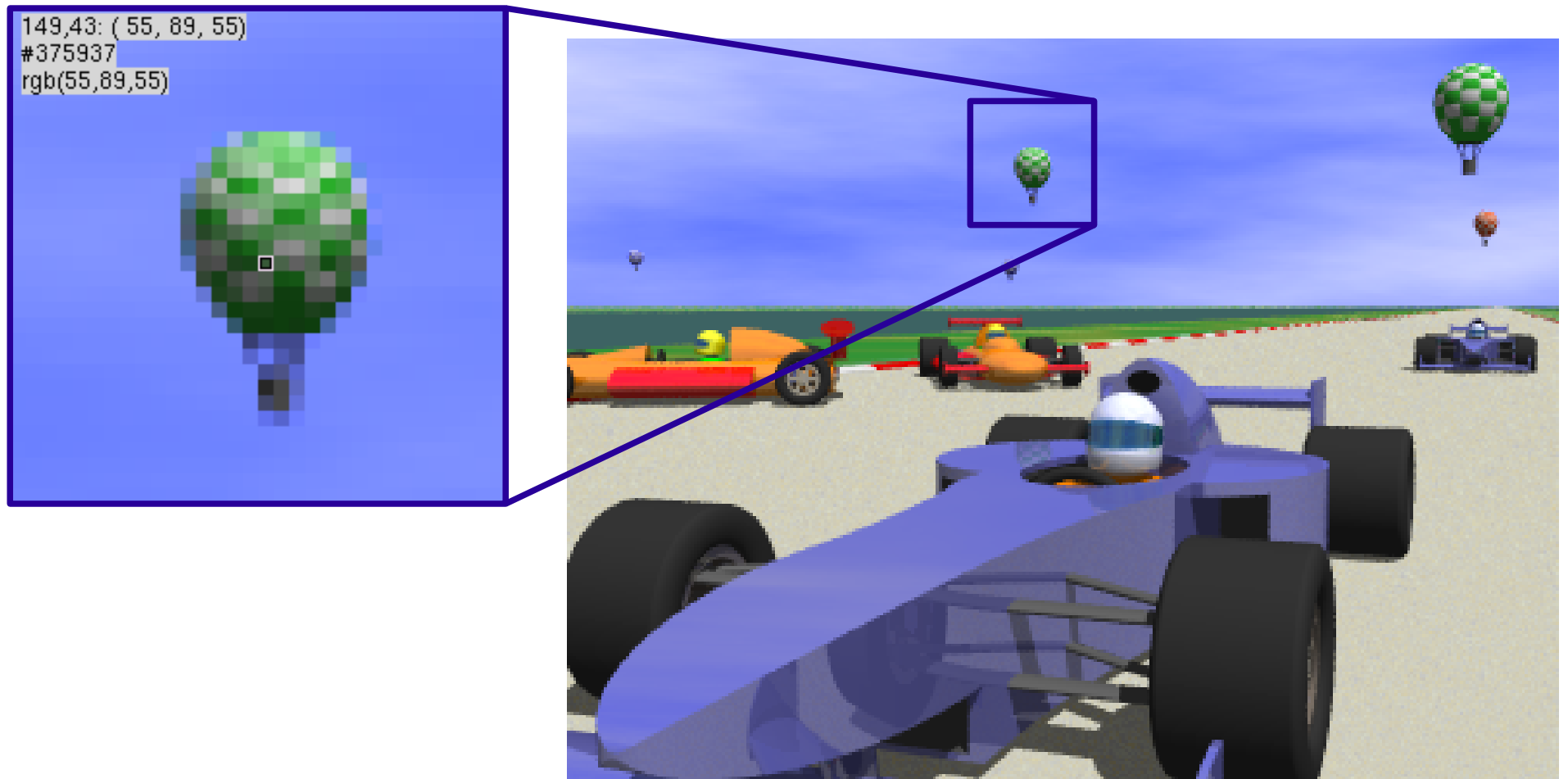
CGの概要(2):どんな技術?

- CG技術 ≡ 図形処理 + 画像処理



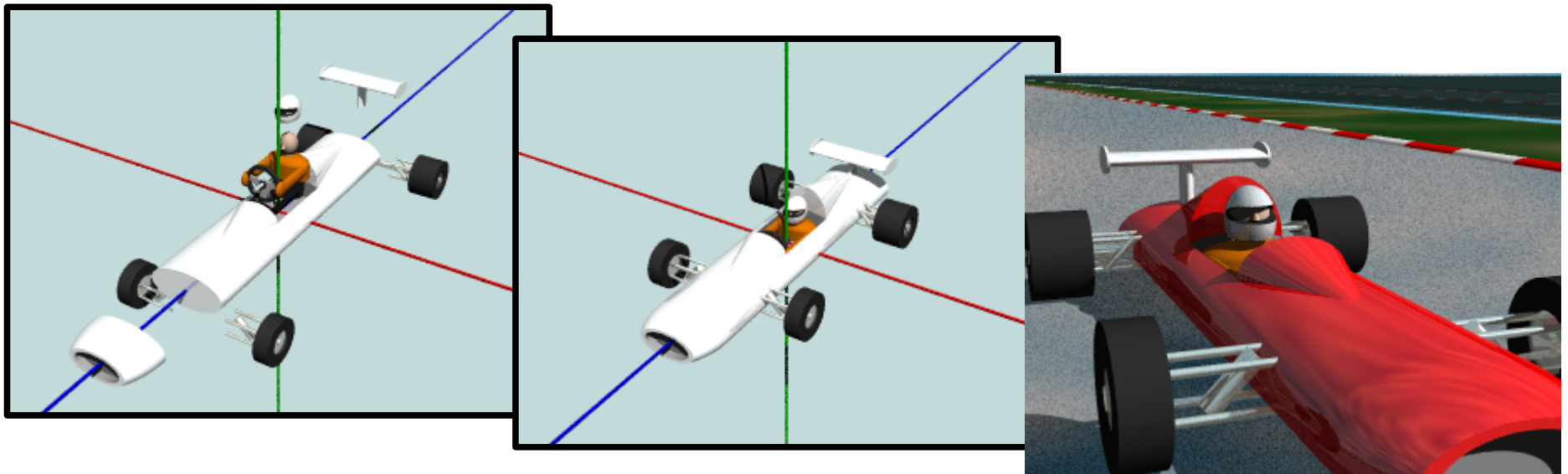
CGの概要(3): 画像処理

- 画像 = 画素の集合, 2D (2次元)



CGの概要(4): 図形処理

- 図形 = 点・線・面・立体の集合, 3D or 2D
- 図形処理の例:
 - **モデリング**: 図形の定義 (座標等の数値データ)
 - **レンダリング**: 図形の表示 (データ→画像化)



CGの概要(5):応用分野

- 娯楽 --- 映画, ビデオゲーム
- 工業 --- 設計図, 完成予想図, 計算機実験
- 医療 --- 手術訓練
- etc.

様々な分野で
シミュレーション (simulation)
や
可視化 (visualization)
に利用

CG技術(1):画像生成手順

1. モデリング (modeling)

物体の形状・配置の定義, 3D

2. ビューイング (viewing)

可視部分の計算, 3D→2D変換

3. シェーディング (shading, or lighting)

物体の色(光の反射・屈折・透過)の計算, 3D or 2D

4. レンダリング (rendering, or rasterization)

図形→画像変換(線の集合→画素の集合), 2D

CG技術(2): 困難点

• 画像の情報量の大きさ

- 大量の情報を高速に処理しなきゃならない!!
 - HDTV 静止画: 1920×1080 画素 \times 32bit色 \doteq 8MB
 - HDTV 動画: $8\text{MB} \times 60\text{fps} \doteq 0.5\text{GB/s}$
- しかも表示の処理だけでこんなに...
 - 表示以外の処理もあるのでさらに厳しい

→ 画質と処理速度のトレードオフ

4K とか, 8K とか,
要求は更に
エスカレート

CG技術(3): 発展方向

- リアリティー (reality; 現実感) と
 - リアルタイム (realtime; 実時間) の両立
 - 映画: 画質キープしつつ作成時間削減
 - 作成時間=コスト
 - ゲーム: 速度60fpsキープしつつ画質向上
 - FPS (frame per second; コマ数/秒)
- アルゴリズムとハードウェアの改良

CG技術(4):アルゴリズム

- 高速化の対象例:
 - ラスタライズ:面・線→画素変換を効率的に
 - クリッピング:視野外物体を除外
 - 隠面処理・衝突判定:複雑な形状でも効率的に
- 高品質化の対象例:
 - 大域照明・影付け:無数の光線を考慮
 - 透明物体:隠面処理・照明計算が複雑
 - 質感表現:材質毎に異なるシェーディング方法

CG技術(5):ハードウェア

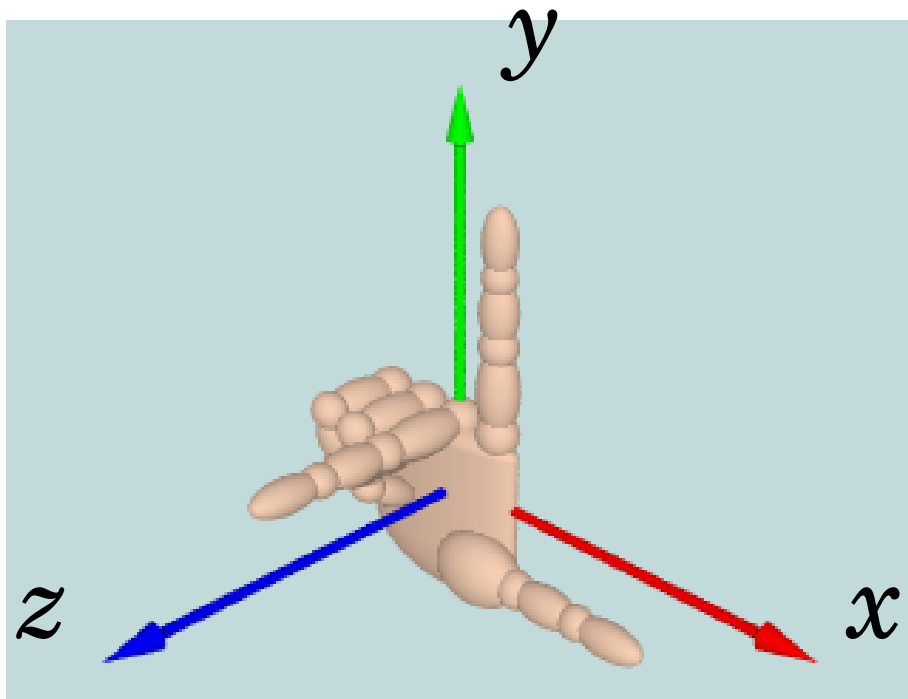
- GPU (Graphics Processing Unit):
 - CG専用プロセッサなので高速(発展中)
 - CPU の負担を軽減(相乗効果)
 - 描画関連の重い処理をすべて GPU 側で実行
 - シェーディング
 - ビューイング
 - レンダリング
 - 描画以外の軽い処理だけを CPU 側で実行
 - モデリング
 - ビジネスロジック

授業内容

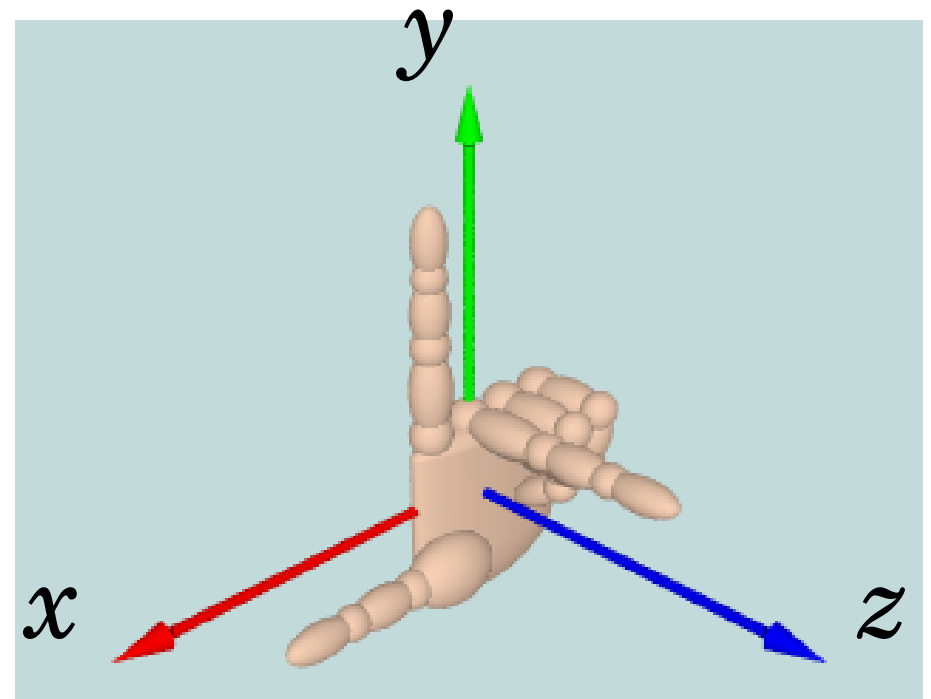
- シラバス説明

3D直交座標系(1):分類

右手系

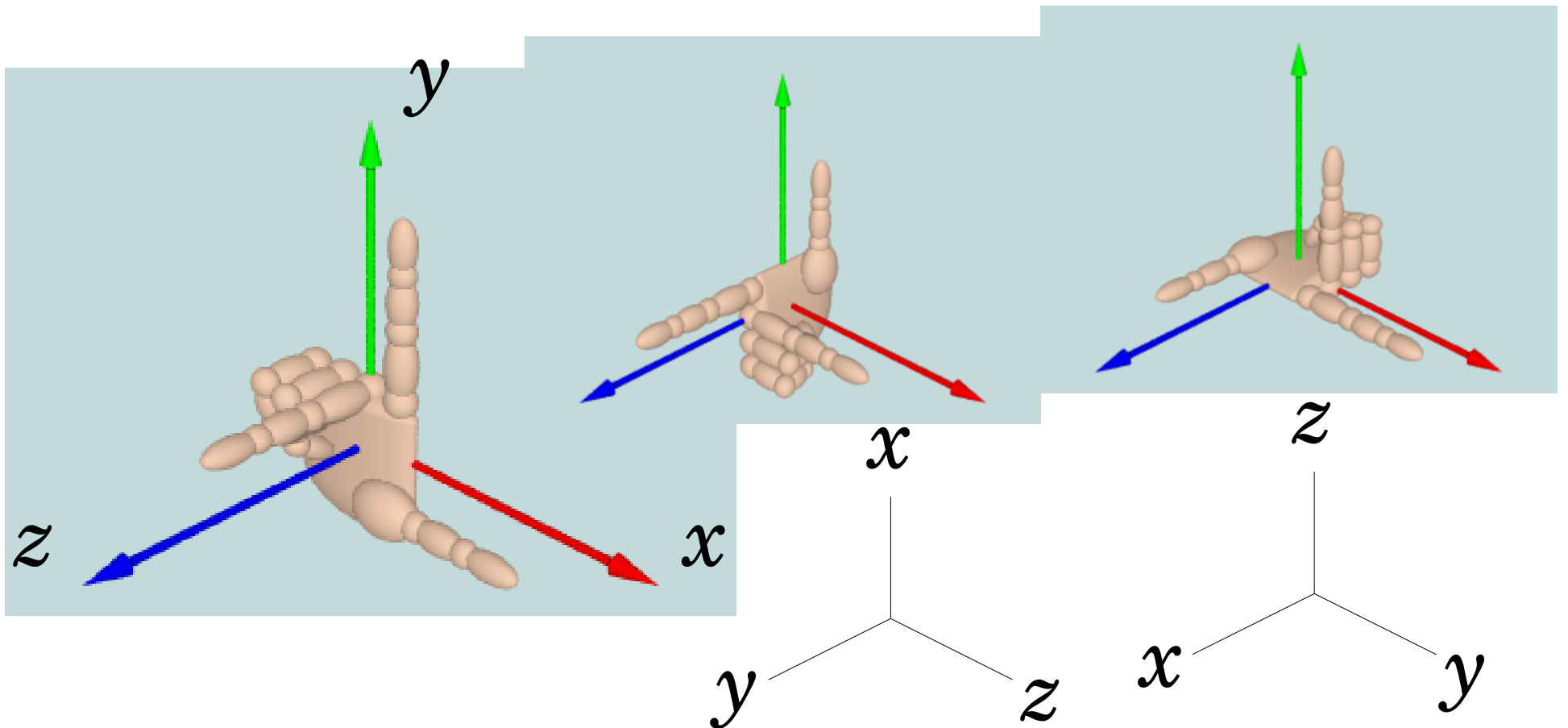


左手系



3D直交座標系(2): 右手系

- 方向は自由だ!
 - ただし x - y - z と3指の対応が循環的であること。



3D直交座標系(3):回転

- 中心が x 軸の場合: y 軸 \rightarrow z 軸
 - では, y 軸中心, z 軸中心の場合は?

