

# Graphics with Processing



2015-13 モデリング

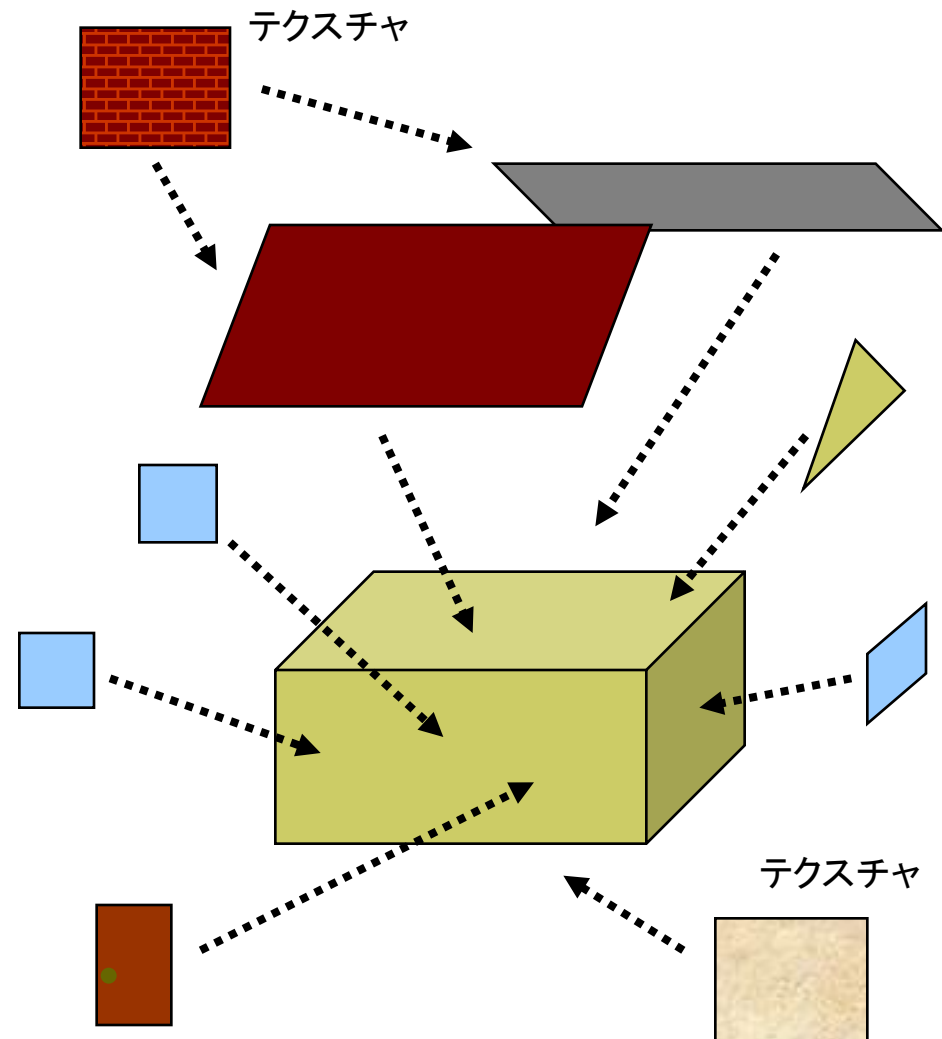
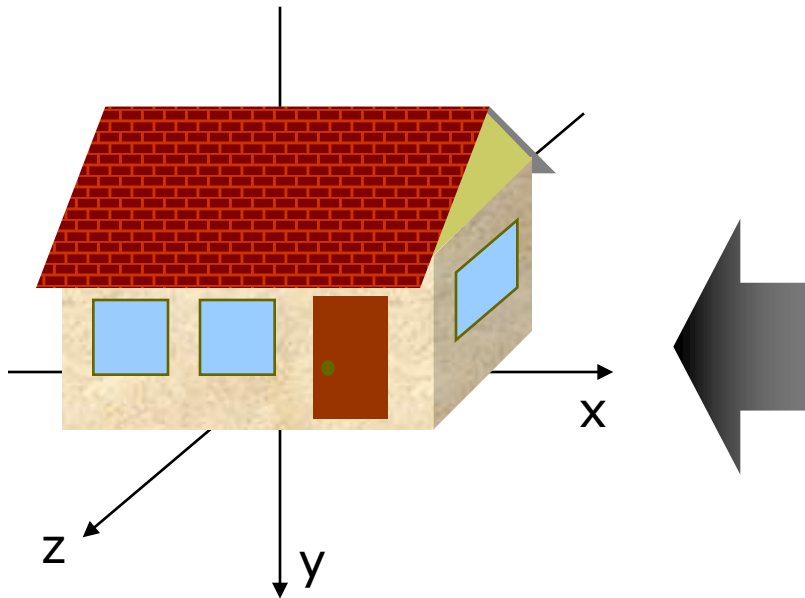
<http://vilab.org>

塩澤秀和

# 13.1 3Dモデリング

## モデリング

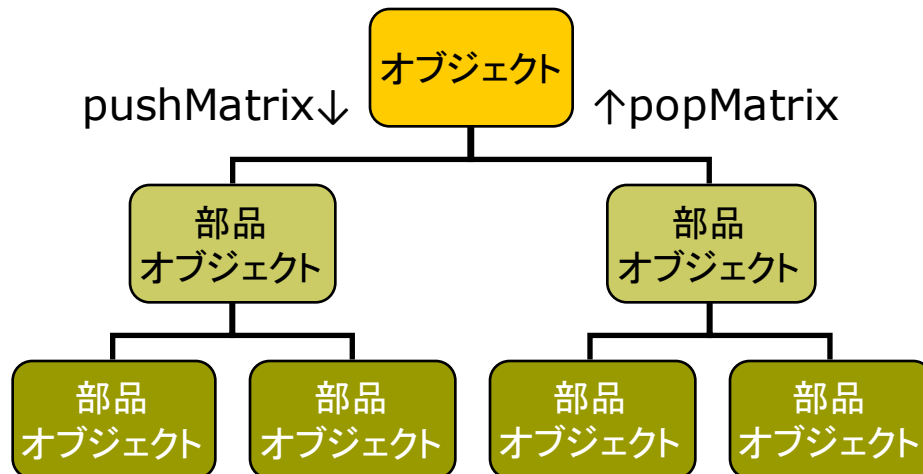
- 3Dオブジェクト(物体)の形状を数値データの集合で表すこと
- オブジェクト座標系で基本図形やポリゴンを組み合わせる



## 13.2 階層モデリング

### 階層モデリング (p.54)

- ローカル座標系の階層化
  - 部品はそれぞれの座標系で作リ、階層的に大きな部品に組み立てていくようにモデリングする
  - 動きの基準点(関節など)を原点として可動部を部品化する
  - 描画では行列スタックを使う (pushMatrix / popMatrix)



```

void cone() { // 円錐
  pushMatrix();
  beginShape(TRIANGLE_FAN);
  vertex(0, -1, 0);
  for (int a = 0; a <= 360; a += 10) {
    float x = cos(radians(a));
    float z = sin(radians(a));
    vertex(x, 0, z);
  }
  endShape();
  popMatrix();
}

```

```

void tree() { // 円錐を組み合わせた木
  pushMatrix();
  translate(0, -0.3, 0);
  scale(0.2, 0.7, 0.2);
  fill(0, 255, 0); cone(); // 円錐1
  popMatrix();
  pushMatrix();
  scale(0.1, 1, 0.1);
  fill(100, 0, 0); cone(); // 円錐2
  popMatrix();
}

```

## 13.3 少し複雑なモデリング例

```

// 推奨モード
// バージョン2,3 ⇒ P3D
// バージョン1 ⇒ OPENGL

void house()
{
    // 壁
    pushMatrix();
    translate(0, -0.5, 0);
    fill(#ffffaa);
    box(2, 1, 1.4);
    popMatrix();
    // 屋根の下
    beginShape(TRIANGLES);
    vertex(1, -1, 0.7);
    vertex(1, -1.7, 0);
    vertex(1, -1, -0.7);
    vertex(-1, -1, 0.7);
    vertex(-1, -1.7, 0);
    vertex(-1, -1, -0.7);
    endShape();

    // 屋根
    beginShape(QUAD_STRIP);
    fill(#ffffff);
    // テクスチャはsetup()の中で
    // roof = loadImage("roof.jpg");
    // として読み込んでおく
    texture(roof);
    textureMode(NORMAL);
    vertex(-1.1, -0.8, 0.9, 0, 1);
    vertex(1.1, -0.8, 0.9, 1, 1);
    vertex(-1.1, -1.7, 0, 0, 0);
    vertex(1.1, -1.7, 0, 1, 0);
    vertex(-1.1, -0.8, -0.9, 0, 1);
    vertex(1.1, -0.8, -0.9, 1, 1);
    endShape();
    // 煙突
    fill(#880000);
    pushMatrix();
    translate(-0.5, -1.4, -0.5);
    box(0.2, 1, 0.2);
    popMatrix();

    beginShape(QUADS);
    // 窓
    fill(#4444ff);
    float z = 0.701;
    vertex(-0.8, -0.7, z);
    vertex(-0.8, -0.3, z);
    vertex(-0.4, -0.3, z);
    vertex(-0.4, -0.7, z);
    vertex(-0.2, -0.7, z);
    vertex(-0.2, -0.3, z);
    vertex(0.2, -0.3, z);
    vertex(0.2, -0.7, z);
    // ドア
    fill(#883333);
    vertex(0.4, -0.8, z);
    vertex(0.4, -0.1, z);
    vertex(0.8, -0.1, z);
    vertex(0.8, -0.8, z);
    endShape();
}

```

## 13.4 複雑な形状の表現

### 曲面や自然形状

- パラメトリック曲面 (p.87)
  - パラメータ方程式による曲面
  - ベジエ曲面やNURBS曲面など
  - レンダリング時にポリゴンに変換する方式としない方式がある
- ポリゴン曲面の操作 (p.94)
  - 細分割曲面: ポリゴンを再帰的に分割し, 滑らかな面を生成
  - 詳細度制御: 視点から遠い曲面のポリゴン数を削減して簡略化
- フラクタル (p.109)
  - 自然界によく見られる再帰的な形状(※)のモデリングに適する
  - ※ 海岸線や木の枝など, 一部分が全体の縮小のような形状のもの

```
// フラクタルによる地形生成の例 (14.5へ続く)
final int N = 256;
float [][] h = new float[N+1][N+1];
int w = N; // wは計算済みの要素の間隔

void setup() {
  size(800, 800, P3D);
  frameRate(30);
  randomSeed(millis());
  // 計算の起点になる4隅の高度を0とする
  h[0][0] = h[0][N] = h[N][N] = h[N][0] = 0.0;
}

// クリックで1段階ずつ細くなる
void mouseClicked() {
  generate();
}

// 補間点の高度に加えるランダム量
float rnd() {
  return random(-0.2, +0.2) * w;
}
```

## 13.5 地形生成の例(続き)

```
void draw() {
  background(50, 50, 150);
  lights();
  translate(width/2, height/2);
  rotateX(PI/4);
  rotateZ(radians(frameCount));
  noStroke(); fill(180, 150, 50);

  // 間隔wの要素を使って地形を描画
  scale(2.0); translate(-N/2, -N/2, 0);
  beginShape(QUADS);
  for (int x = 0; x < N; x += w) {
    for (int y = 0; y < N; y += w) {
      vertex(x, y, h[x][y]);
      vertex(x, y+w, h[x][y+w]);
      vertex(x+w, y+w, h[x+w][y+w]);
      vertex(x+w, y, h[x+w][y]);
    }
  }
  endShape();
}
```

```
// 地形を1段階細かくする
void generate() {
  if (w == 1) return;

  for (int x = 0; x < N; x += w) {
    for (int y = 0; y < N; y += w) {
      // 中点の高度を補間し, 適当な乱数を加える
      h[x+w/2][y] = (h[x][y] + h[x+w][y]) / 2 + rnd();
      h[x][y+w/2] = (h[x][y] + h[x][y+w]) / 2 + rnd();
      // 4点の中央の高度も同様の計算で求める
      h[x+w/2][y+w/2] = (h[x][y] + h[x+w][y]
        + h[x+w][y+w] + h[x][y+w]) / 4 + rnd();
    }
  }
  for (int i = 0; i < N; i += w) {
    h[i+w/2][N] = (h[i][N] + h[i+w][N]) / 2 + rnd();
    h[N][i+w/2] = (h[N][i] + h[N][i+w]) / 2 + rnd();
  }
  // 計算済みの要素の間隔は1/2になる
  w /= 2;
}
```

## 13.6 3DCGフリーソフト(1)

### Art of Illusion

#### □ 概要

- [www.artofillusion.org](http://www.artofillusion.org)
- 基本機能をサポート(モデリング, レンダリング, アニメーション)
- 無料&オープンソース
- OBJ形式で出力可能

#### □ インストールと実行

- ArtOfIllusion???.exe
- (英語で)ライセンスへの承諾を求められるので, [Yes]を選択
- スタートメニューの[Start Art of Illusion]から起動

#### □ 使い方の参考(日本語)

- [ei-www.hyogo-dai.ac.jp/~masahiko/moin.cgi/AOI](http://ei-www.hyogo-dai.ac.jp/~masahiko/moin.cgi/AOI)

### 使い方のポイント

#### □ 基本描画

- 左のツールボタンから選択
- 図形の配置, 移動, 回転など...
- [シーン]→[レンダラー]でレイトレーシングのCGも生成できる

#### □ 色とテクスチャ

- 単色: タイプ[Uniform]
- 画像: タイプ[Image Mapped]

#### □ OBJ形式での出力

- [ファイル]→[データ書き出し]→[Wavefront(.obj)]
- [テクスチャをmtlで書き出し]

#### □ OBJ出力での注意点

- AoIの発光色(Ke)は, OBJでは環境反射色(Ka)に変換される

## 13.7 3DCGフリーソフト(2)

---

### SketchUp

#### □ 概要

- [www.sketchup.com](http://www.sketchup.com)
- 人工物のモデリングに向く
- Google Earthに建物のモデルをアップロードして設置できる

#### □ 使い方

- [www.atmarkit.co.jp/fwcr/rensai2/3dcurl01/01.html](http://www.atmarkit.co.jp/fwcr/rensai2/3dcurl01/01.html)
- [www.sketchup.com/learn/videos?playlist=58](http://www.sketchup.com/learn/videos?playlist=58)

#### □ OBJ形式での出力

- 商品版(Pro)だけの機能だが...
- 無料版でも使えるプラグインを開発している人がいる
- [sketchup-onigiri.jimdo.com/sketchup-plugins/su2objmtl/](http://sketchup-onigiri.jimdo.com/sketchup-plugins/su2objmtl/)

### その他のフリーソフト

#### □ Blender

- [www.blender.org](http://www.blender.org)
- ほぼ万能で無料&オープンソース

#### □ MagicaVoxel

- [ephtracy.github.io](http://ephtracy.github.io)
- ボクセル=3次元空間のマス目(立方体)を埋めてモデリング

#### □ Sculptris

- [pixologic.com/sculptris/](http://pixologic.com/sculptris/)
- 粘土・彫刻のようにモデリング

#### □ ブラウザソフトウェア

- [www.tinkercad.com](http://www.tinkercad.com)
- [www.3dtin.com](http://www.3dtin.com)
- [stephaneginier.com/sculptgl/](http://stephaneginier.com/sculptgl/)
- [www.smoothie-3d.com](http://www.smoothie-3d.com)



## 13.8 3DCG商用ソフトウェア等

---

### 比較的手軽なもの

- LightWave
  - [www.dstorm.co.jp](http://www.dstorm.co.jp)
  - 総合3DCGソフト(PC演習室)
- Shade 3D
  - [shade.e-frontier.co.jp](http://shade.e-frontier.co.jp)
  - 日本製(PC演習室)
- Terragen
  - [www.planetside.co.uk](http://www.planetside.co.uk)
  - 自然景観生成(PC演習室)
- Metasequoia
  - <http://www.metaseq.net/jp/>
  - 日本製でアマチュアに人気
  - 基本機能は無料
- UnityからOBJ書き出し
  - [www.assetstore.unity3d.com/en/#!/content/22250](http://www.assetstore.unity3d.com/en/#!/content/22250)

### ハイエンド製品

- 3大CGソフト(Autodesk社)
  - 3ds Max, Maya, Softimage
  - 書籍/Webなどの情報が多い
  - 使いこなすのは(多分)難しい
  - 学生は無償で利用可能
  - [www.autodesk.co.jp/education](http://www.autodesk.co.jp/education)

### 自由課題

- モデリングソフトウェアを使って、自由な3Dモデルを作成してみよ
- 作成した3DモデルをOBJ形式で保存するか、一度保存したものをOBJ形式に変換して、Processingで表示させてみよ
- 今回の課題は提出はしなくてよい