

Graphics with Processing



2006-12 モデリング

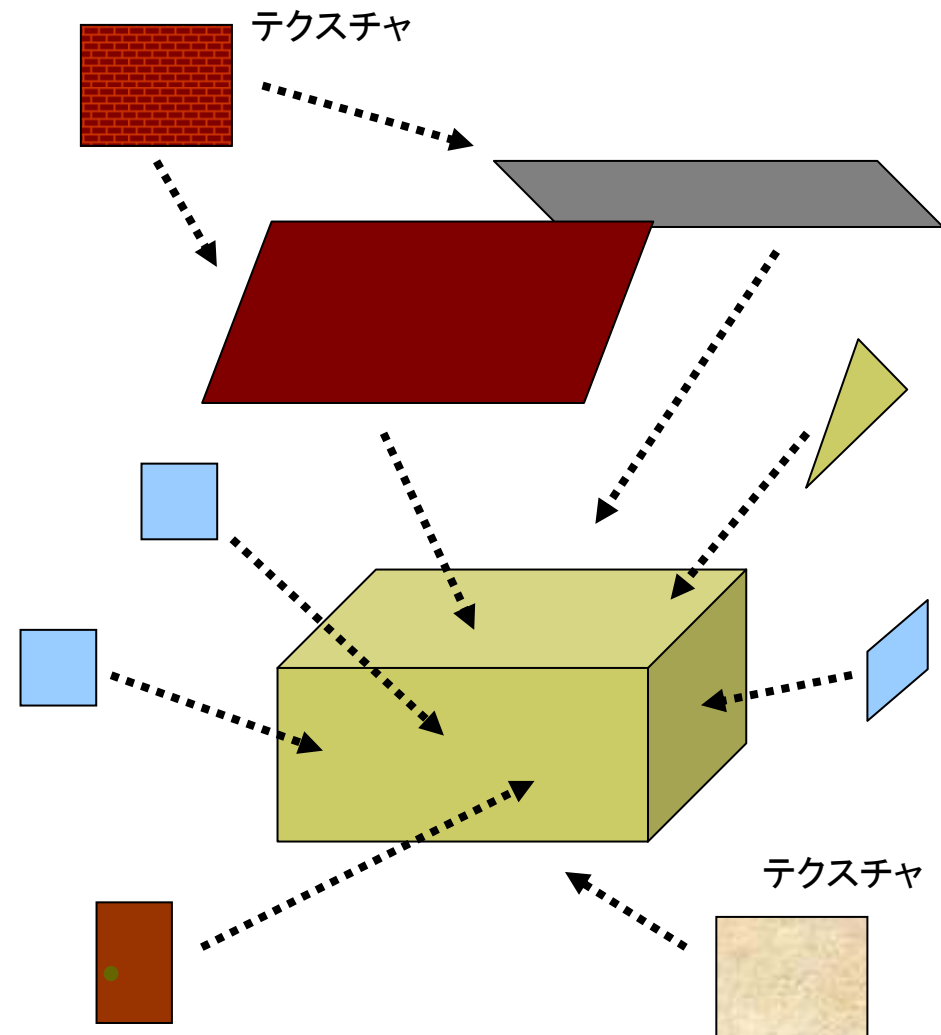
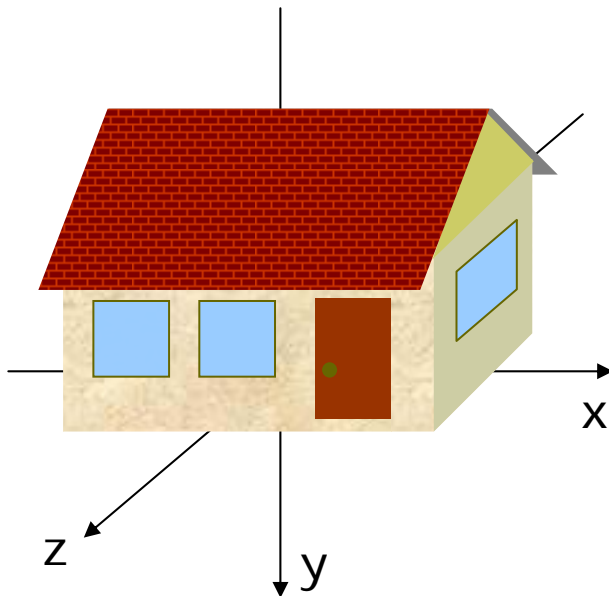
<http://vilab.org>

塩澤秀和

12.1 3Dモデリング

モデリング

- 3Dモデルを作り上げること
- オブジェクト座標系に基本図形やポリゴンを組み合わせる



12.2 モデリングサンプル

```

// OPENGGLのほうが正確
// size(幅, 高さ, OPENGGL);
// P3Dだとテクスチャが歪む

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(NORMALIZED);
  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();
}

```

12.3 CGソフトウェアの利用

Art of Illusion

- ホームページ
 - <http://www.artofillusion.org>
 - OBJ形式を読み書き可能
→ Processing で利用可能
- インストールと実行
 - aoiバージョン.zip
 - 適当なディレクトリに展開
 - ArtOfIllusion.jar をダブルクリックして実行
- Windows用インストーラ
 - Ver 2.4からインストーラ版あり
ArtOfIllusion24-Windows.exe
- 使い方の参考(日本語)
 - <http://ei-www.hyogo-dai.ac.jp/~masahiko/aoi/index.html>

基本モデリング

- 基本描画
 - 左のツールボタンから選択
 - 移動, 回転, 選択,
 - 選択後, ダブルクリックで編集
- ツール
 - 図形を選択し, ツールメニュー
 - 回転体, 平面図形の立体化など
- 色・テクスチャ
 - テクスチャ → 新規テクスチャ...
 - シーン → レンダーで確認
- モデルデータの保存
 - 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」でテクスチャも保存

12.4 その他のソフトウェア

その他のソフトウェア

- Rios
 - <http://hwpbc.gate01.com/junk/>
 - とりあえずお勧めか？
- 3DAce
 - <http://hp.vector.co.jp/authors/VA017881/>
 - 多くのファイル形式を扱える
- メタセコイアLE
 - <http://www.metaseq.net/metaseq/>
- Blender
 - <http://www.blender3d.org>
 - <http://blender.jp>

3Dモデルデータ

- 3D CHATA
 - <http://www.3dchaya.com>
 - 城など(テクスチャはない)
- MultiMedia help
 - <http://www.multimediahelp3d.8m.com>
 - 日用品など(実は閉鎖サイト?)
- その他
 - <http://homepage1.nifty.com/hakka/edo/memo/fdata.html>
 - <http://www.3dcafe.com>
 - 「3Dモデル フリー」などで検索
 - OBJ形式への変換が必要

12.5 モデルデータの利用

モデルデータの読み込み

- OBJ Loader
 - processingの拡張機能
 - OBJ形式の3Dモデル読み込み (dataフォルダに入れておく)
 - <http://users.design.ucla.edu/~tatsuyas/tools/objloader/index.htm>
- ダウンロード
 - objloader_バージョン.zip
 - 展開(解凍)し, objloaderフォルダをProcessingをインストールしたフォルダの下のlibrariesフォルダにコピー
- プログラム冒頭
 - `import saito.objloader.*;`

モデルデータの描画

- OBJModel型
 - まず, データ用の変数を用意
 - `OBJModel m = new OBJModel(this)`
- `m.load("ファイル名.obj")`
 - データファイルの読み込み
- `m.drawMode(描画モード)`
 - 描画モードの設定
 - TRIANGLES か POLYGON
- `m.enableTexture(), m.disableTexture()`
 - テクスチャの有効化と無効化
- `m.draw()`
 - モデルの描画

12.6 OBJ Loader の使用例

```
// 準備:モデルデータ(beethoven.obj,  
// beethoven.mtl, beethoven1.jpg  
// の3つのファイル)をダウンロードし,  
// Sketch → Add File... で登録しておく
```

```
import saito.objloader.*;
```

```
OBJModel model;
```

```
void setup()
```

```
{  
  size(400, 400, P3D);  
  model = new OBJModel(this);  
  model.load("beethoven.obj");  
}
```

```
void draw()
```

```
{  
  background(0, 0, 100);  
  lights();
```

```
  pushMatrix();  
  translate(width*0.3, height/2, 0);  
  rotateX(radians(200));  
  rotateY(radians(frameCount));  
  scale(150);  
  noStroke();  
  model.enableTexture();  
  model.drawMode(TRIANGLES);  
  model.draw();  
  popMatrix();
```

```
  pushMatrix();  
  translate(width*0.7, height/2, 0);  
  rotateX(radians(200));  
  rotateY(radians(frameCount));  
  scale(150);  
  stroke(#ffffff);  
  model.drawMode(LINES);  
  model.draw();  
  popMatrix();  
}
```

12.7 文字列の表示(参考)

英文の表示

- フォントの作成
 - 事前にフォントファイルを準備
 - Tools → Create Font...
- PFont型
 - フォントを表す変数型
- loadFont("フォントファイル名")
 - フォントの読み込み
 - 例) PFont font = loadFont("CourierNew36.vlw");
- textFont(フォント, サイズ)
 - 描画フォントの設定
- text(文字列, x, y)
 - 文字列(String型)の描画
 - Examples → Typography

日本語の表示

- 画像で表示
 - PFontは日本語に非対応
 - 文字列の画像ファイルを用意
- プログラムでの画像を作成
 - 次ページの関数を使う

```
PImage img;

void setup() {
  img = makeTextImage("あいうえお",
    30, 100, 20, color(255, 0, 0));
}

void draw() {
  background(0);
  image(img, frameCount%width, 100);
}
```


12.8 日本語の画像作成(参考)

```
import java.awt.*;
import java.awt.image.*;
import java.awt.font.*;

// 引数は, 文字列, サイズ, 画像の幅, 画像の高さ, 文字の色
PImage makeTextImage(String str, int point, int w, int h, color fg) {
    BufferedImage bi = new BufferedImage(w, h,
        BufferedImage.TYPE_4BYTE_ABGR);
    Graphics gc = bi.getGraphics();
    gc.setColor(new Color(0, 0, 0, 0));
    gc.fillRect(0, 0, bi.getWidth(), bi.getHeight());
    gc.setColor(new Color(fg));
    Font fnt = new Font("SansSerif", Font.PLAIN | Font.BOLD, point);
    gc.setFont(fnt);
    gc.drawString(str, 0, bi.getHeight());
    gc.dispose();
    return new PImage(bi);
}
```