Graphics with Processing

2007-12 モデリング

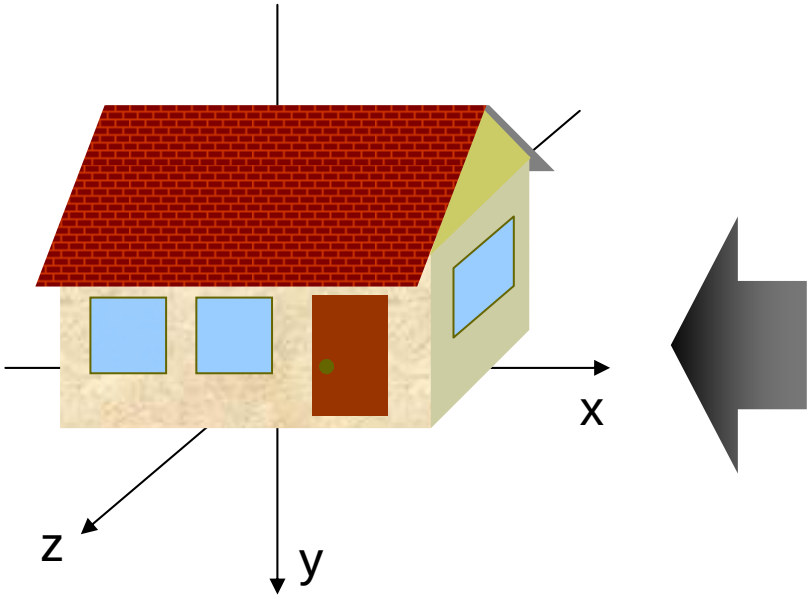
http://vilab.org

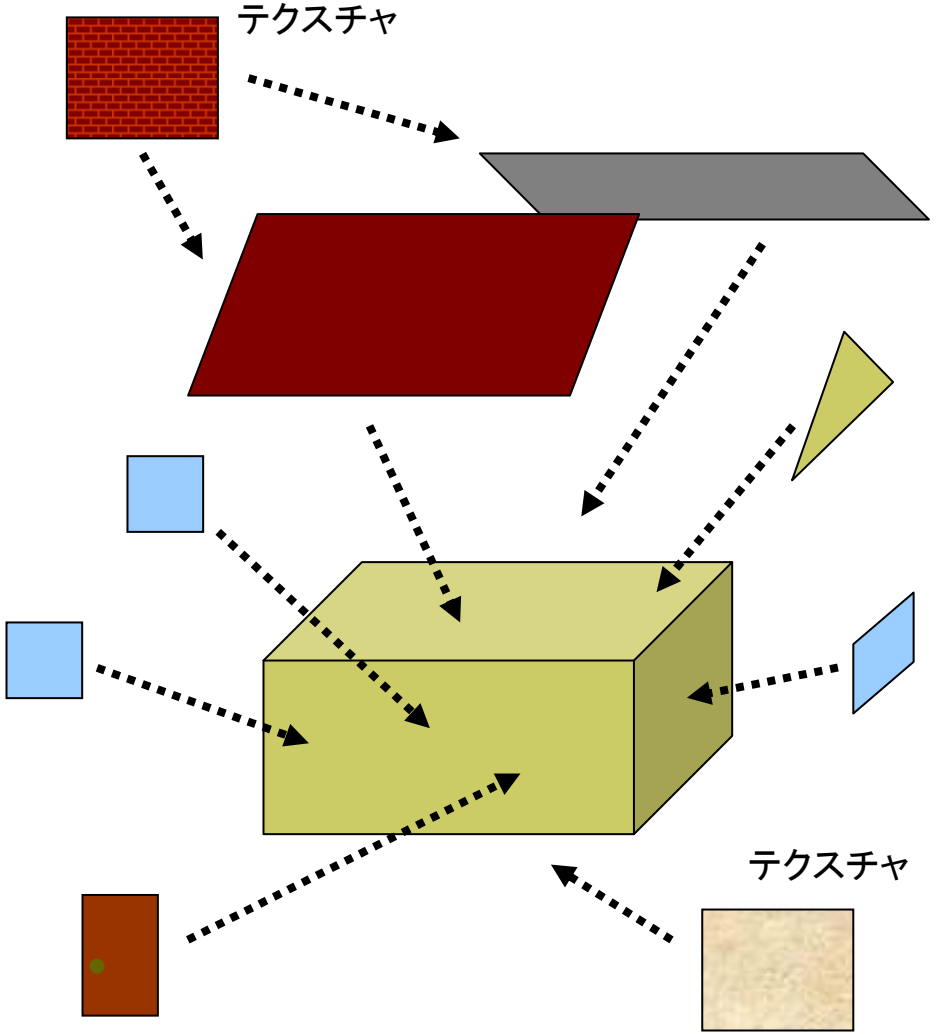
塩澤秀和

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 オブジェクトの関数

複雑なオブジェクトは，大きさ1を目安としてモデリングし，関数にしておくと利用しやすい

雪だるま

```
void snowman() {
  fill(255, 255, 255);
  noStroke();
  pushMatrix();
  translate(0, -0.7);
  sphere(0.2);
  popMatrix();
  pushMatrix();
  translate(0, -0.3);
  sphere(0.3);
  popMatrix();
}
```

円錐（底なし）

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

木（のようなもの）

```
void tree() {
  pushMatrix();
  fill(0, 255, 0);
  translate(0, -0.3, 0);
  scale(0.2, 0.7, 0.2);
  cone();
  popMatrix();
  pushMatrix();
  fill(100, 0, 0);
  scale(0.1, 1, 0.1);
  cone();
  popMatrix();
}
```

12.3 少し複雑なモデリング

```
// OPENGLのほうが正確
// size(幅, 高さ, OPENGL);
// 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();
}
```

12.4 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
  - インストーラをダウンロードして実行するとインストールが始まる
  - ArtOfIllusion???-Windows.exe
  - ライセンスを受け入れるか聞かれるので(英語), Yesを選択
  - スタートメニューの「Start Art of Illusion」をクリックして実行
- 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

12.5 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.multimedia help3d.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

12.6 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，objeloader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

12.7 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();
}
```

12.8 期末レポート

レポート課題

- 内容
  - Processingを使って，**3次元**CGの「作品」を作りなさい
- テーマ（どれか選択）
  - 「食べ物」「クリスマス」「積み木」

  ※ アニメーション（動き）があること
- チーム制
  - 2人でチームを組み，協力して1つの作品を作ってもよい
  - レポートは1人ずつ書き，誰とチームを組んだのか明記する
- 提出
  - 締め切りは，1月21日（月）
  - 期末試験と同じ日なので注意！
- 形式
  - **文書（紙）**：A4レポート用紙
  - **プログラム**：Webページで提出
- 文書の構成
  - **氏名**：必ず1ページ目に書くこと
  - **概要**：プログラムの概要
  - **手順**：どんな手順で作品を設計し，分担し，作ったのかの説明
  - **プログラムの説明**：自分のプログラムの構造と技術の説明
  - **考察**：プログラムについて工夫した点やアピールしたいことなど
  - **まとめ**：レポート全体のまとめ
- 注意
  - 文章は「です・ます」ではなく，「だ・である」で書くこと