



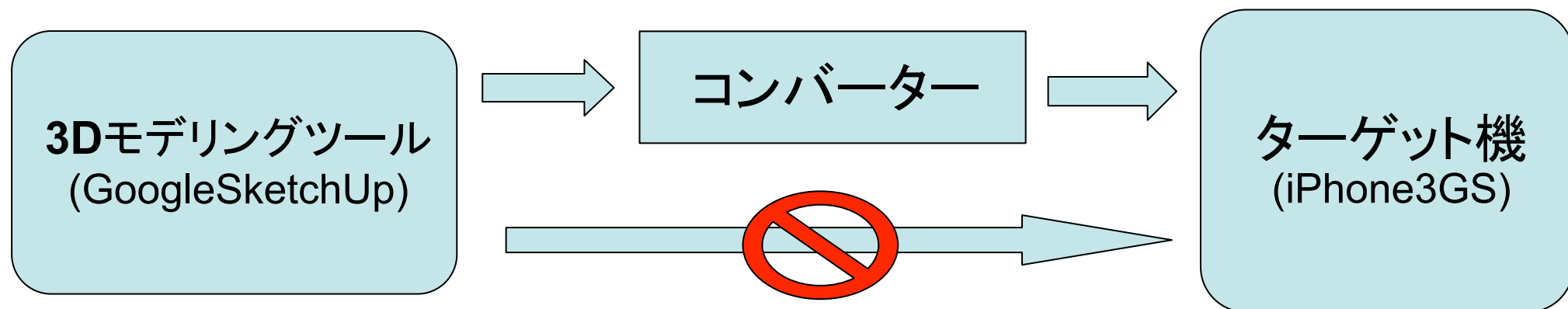
チュートリアル: ゲームグラフィクス・プログラミング入門

2時限目

竹重 雅也

- このセッションのゴール
 - 3Dモデリングツール(GoogleSketchUp※)で作った、任意の形状を、ターゲット機(iPhone3GS※)で表示する。
 - ゲームグラフィックスにおいては、コンバートという手順をとり、ターゲットで表示しやすいデータにあらかじめ加工する。

※GoogleSketchUp: ©Google Inc, 2007
※iPhone 3GS: Copyright © 2009 Apple Inc.



- このセッションの目次
 - 2.1 頂点アトリビュートのセマンティクスの決定
 - 2.2 基本的なシェーディング(HalfLambertShading)
 - 2.3 GoogleSketchUpの説明
 - 2.4 COLLADAファイルの内容
 - 2.5 COLLADAファイルのコンバート
 - 2.6 iPhone3GSでモデルの表示

Appendix Windows環境について

2.1 頂点アトリビュートのセマンティクスの決定

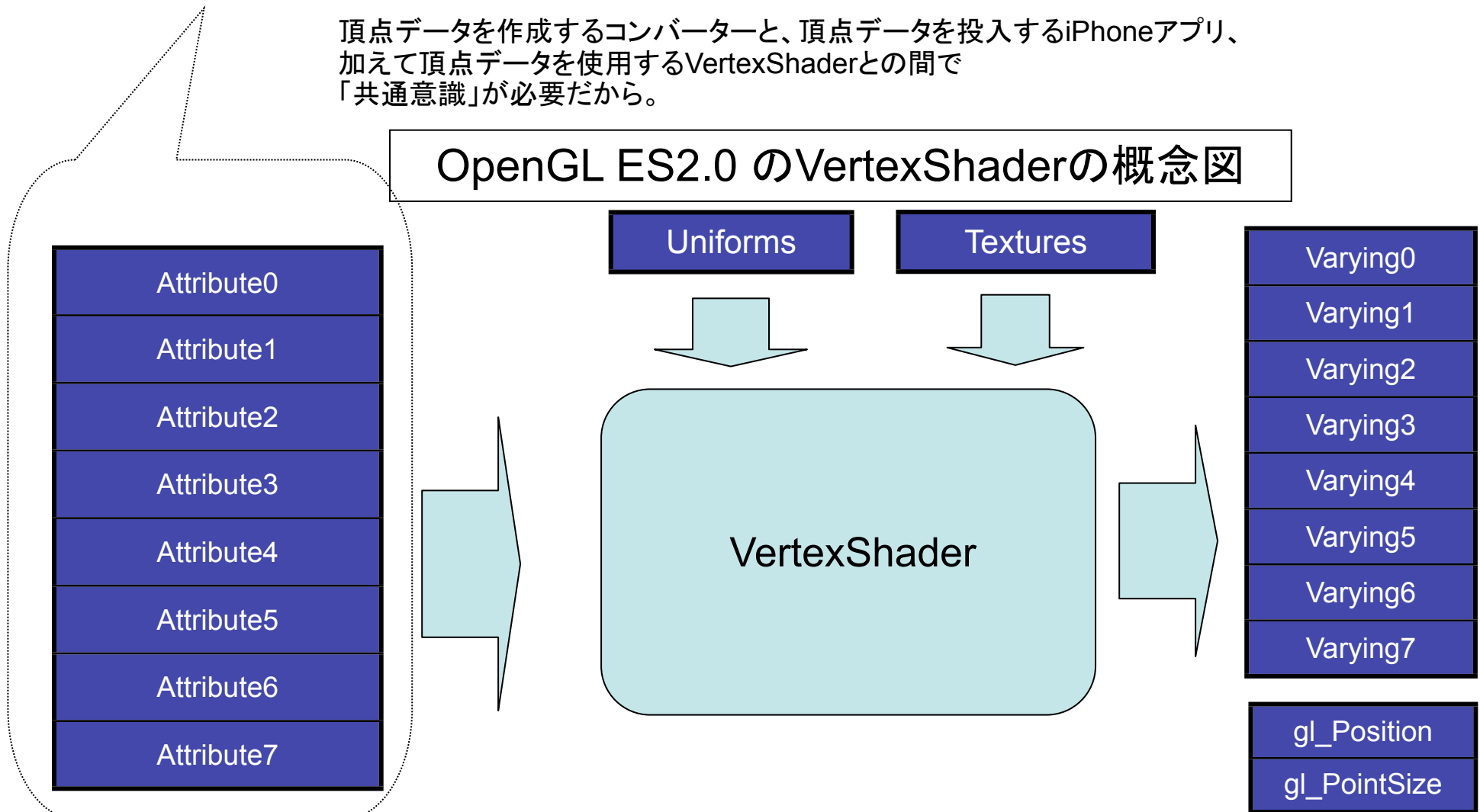


OpenGL API側からVertexShaderに渡す

8種類のAttributeに対して、意味づけをする。

頂点データを作成するコンバーターと、頂点データを投入するiPhoneアプリ、
加えて頂点データを使用するVertexShaderとの間で
「共通意識」が必要だから。

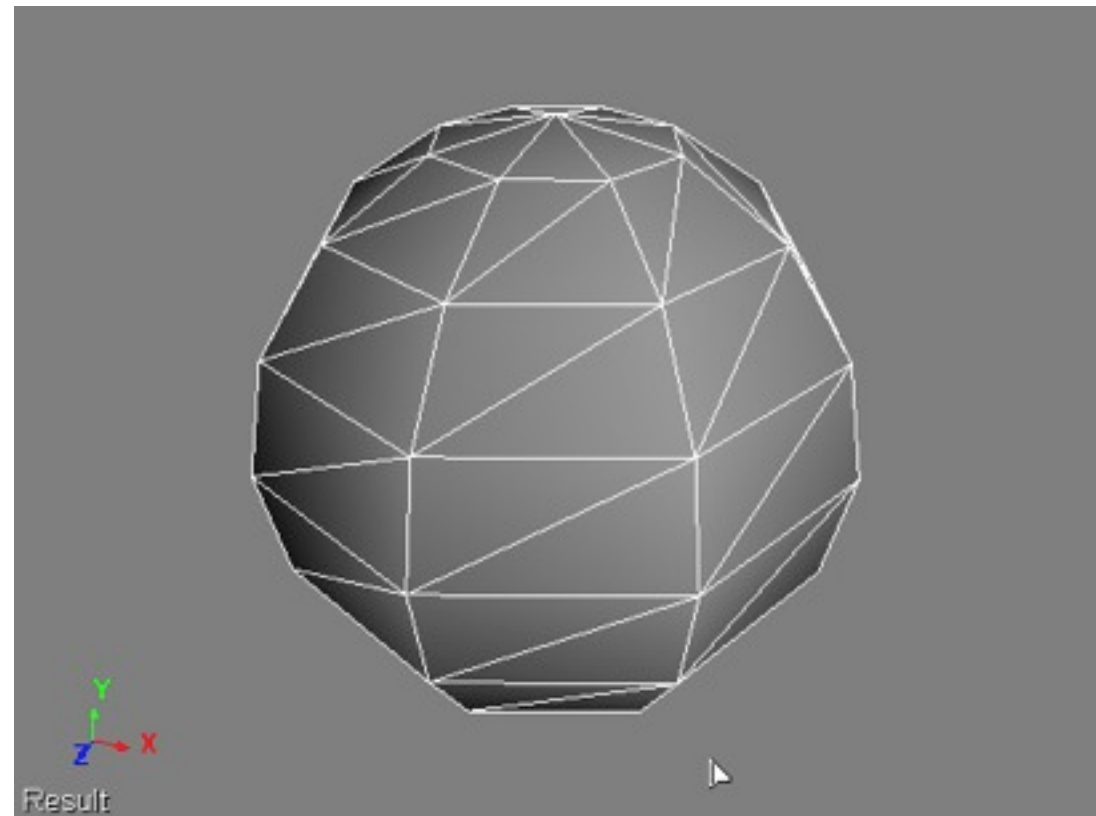
OpenGL ES2.0 のVertexShaderの概念図



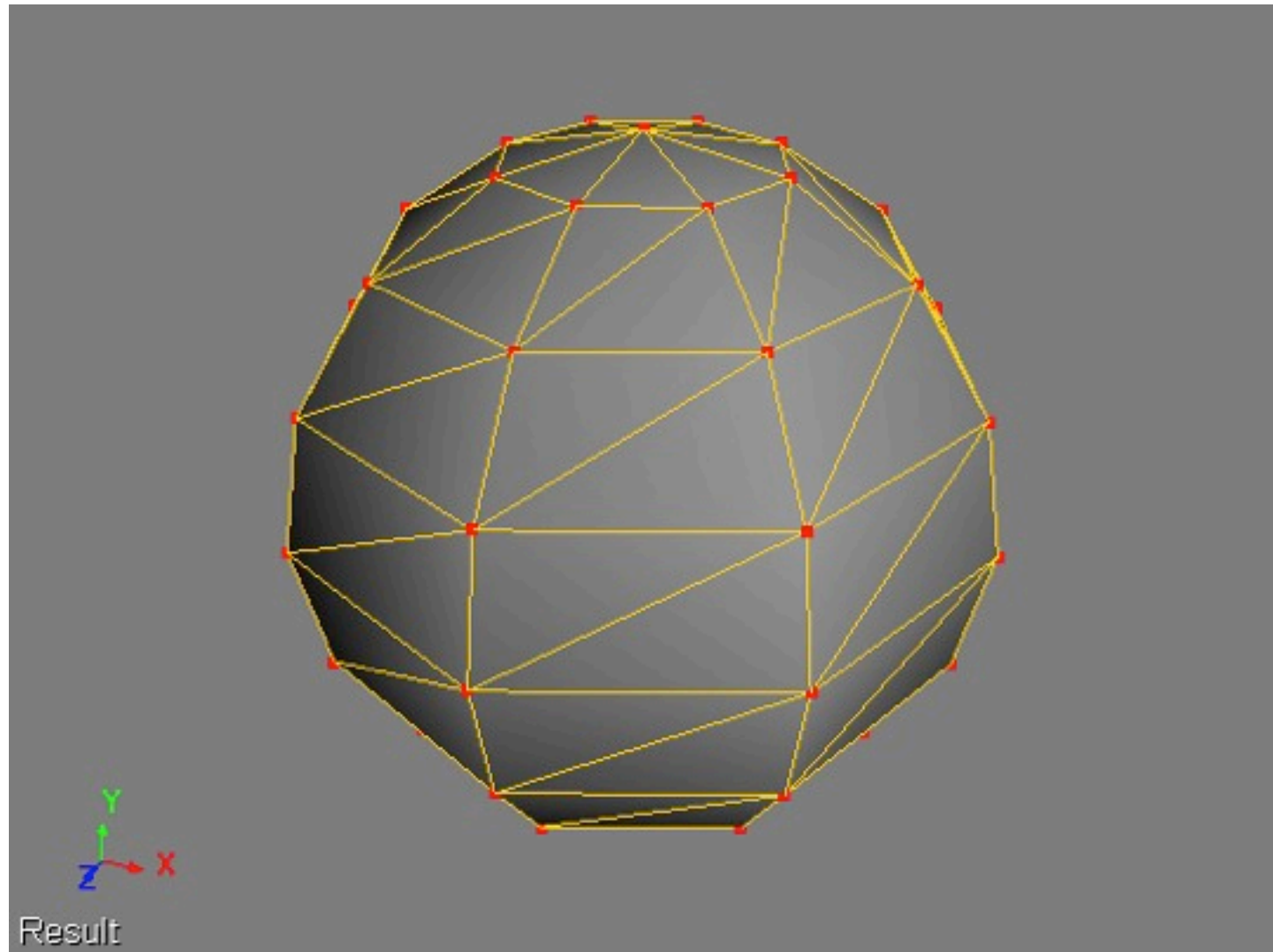
2.1.1 代表的なセマンティクスの説明

- 頂点位置
- 頂点法線
- 頂点接線/従法線
- 頂点カラー
- 頂点テクスチャ座標

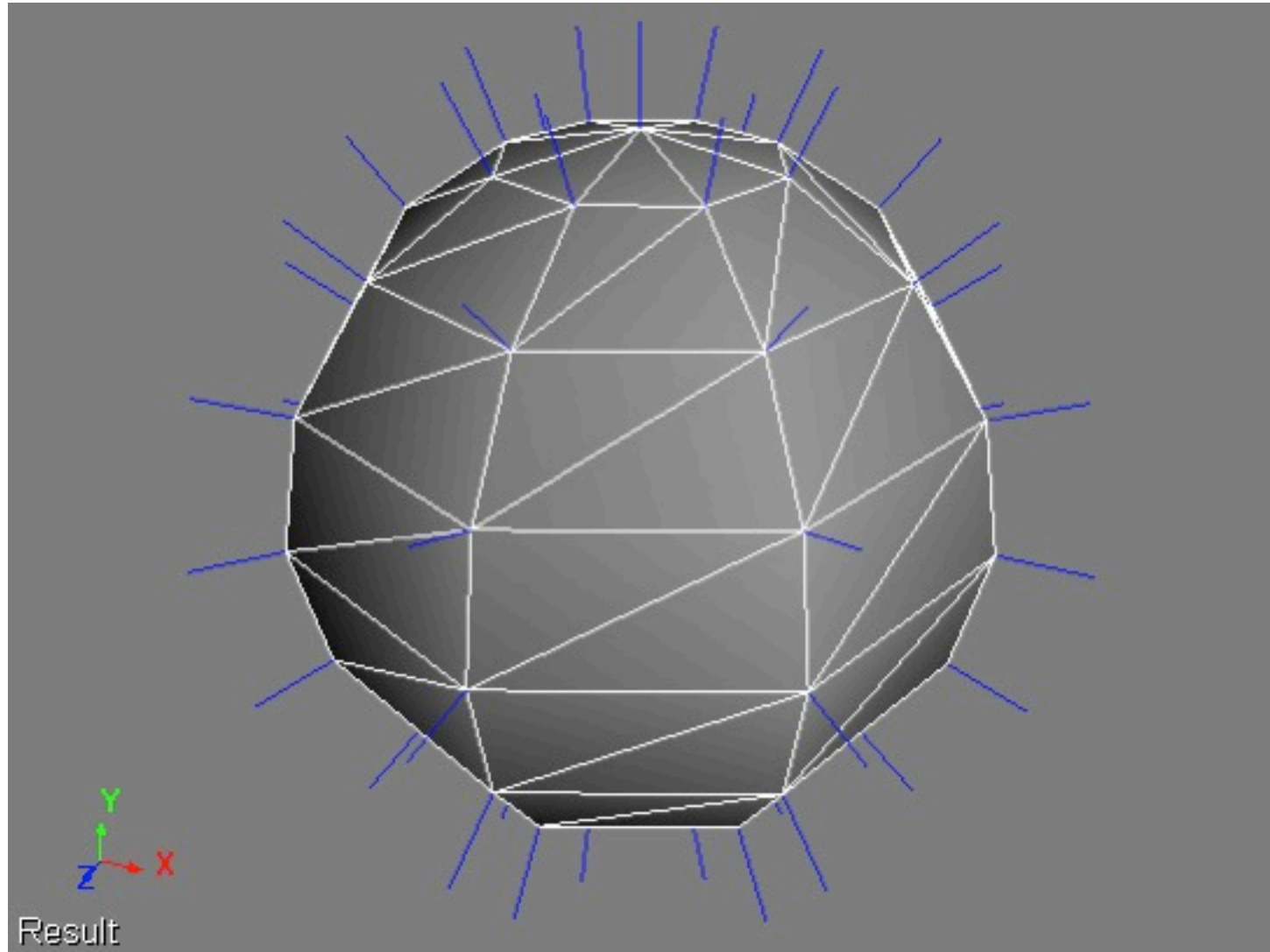
この球のモデルを例に説明します



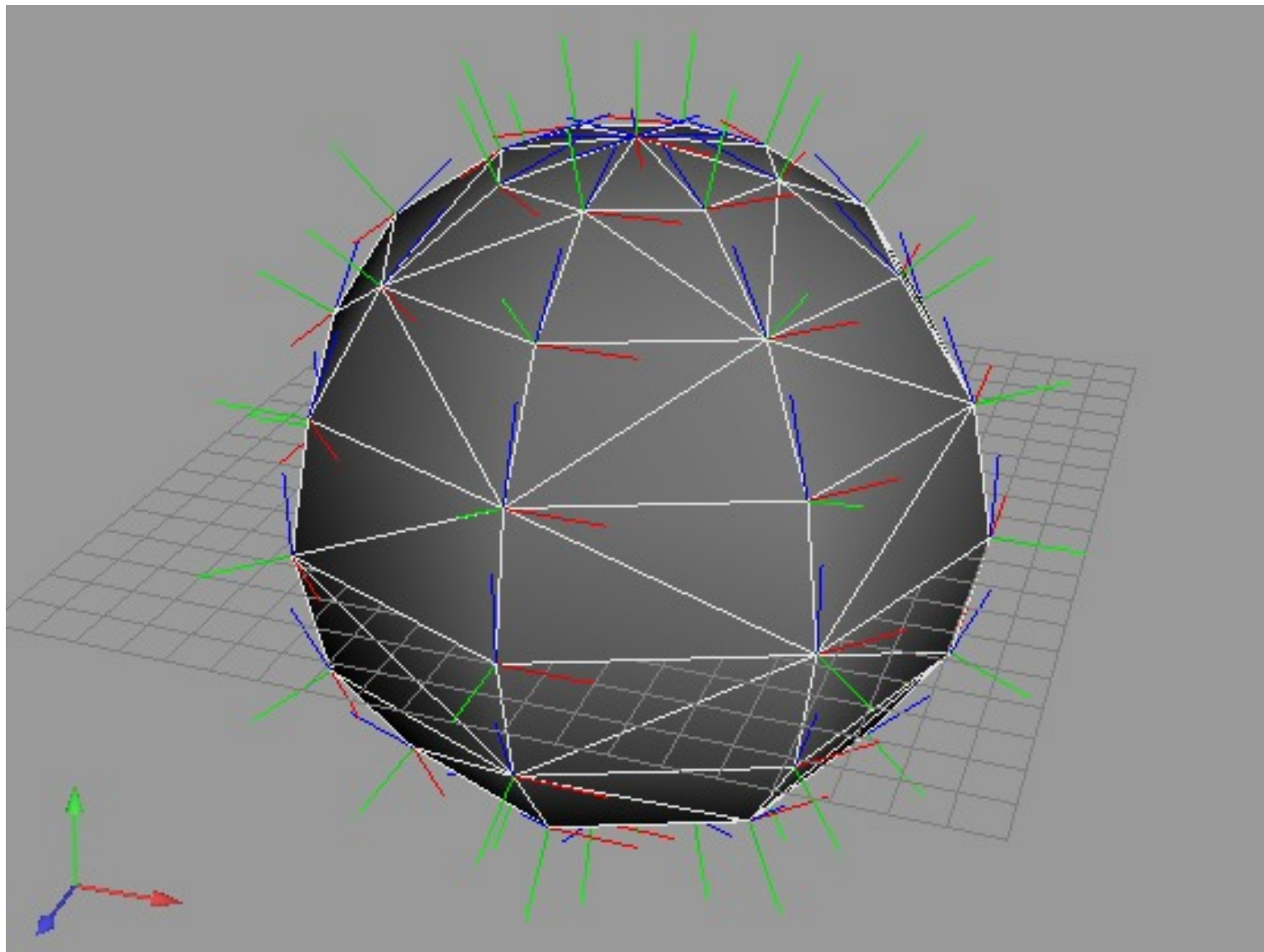
頂点位置 (POSITION)



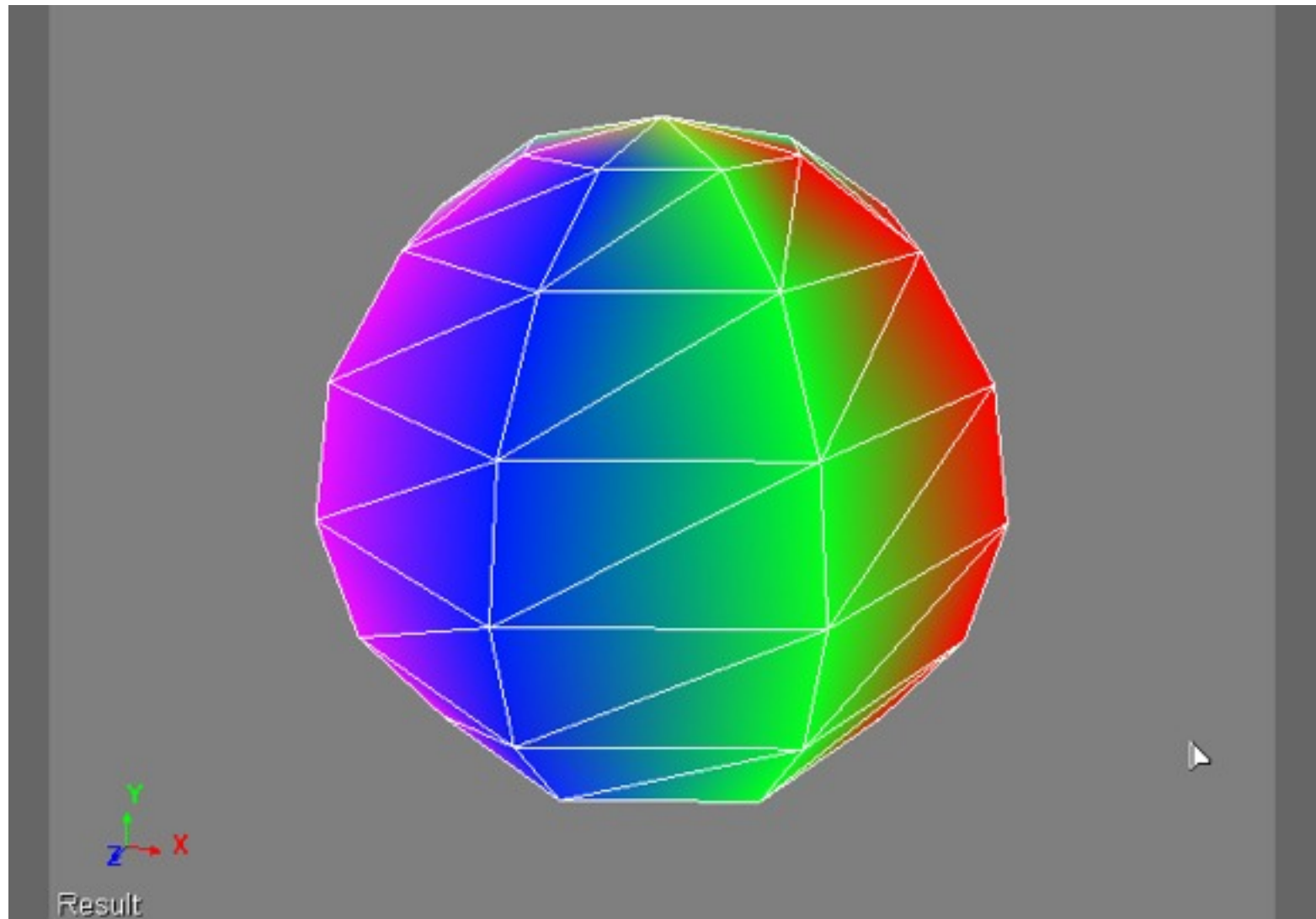
頂点法線(NORMAL)



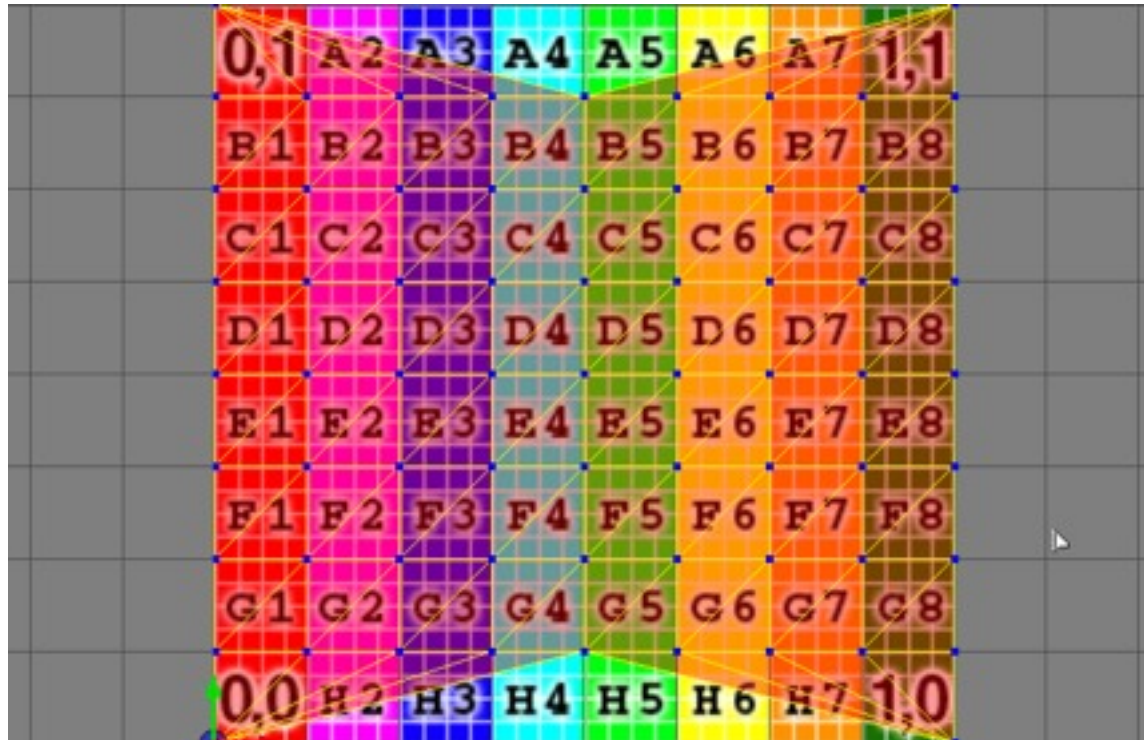
頂点接線/従法線 (TANGENT/BINORMAL)



頂点カラー(COLOR)



頂点テクスチャ座標 (TEXCOORD)



Result

2.1 頂点アトリビュートのセマンティクスの決定



2.1.2 頂点アトリビュートのセマンティクスの一例

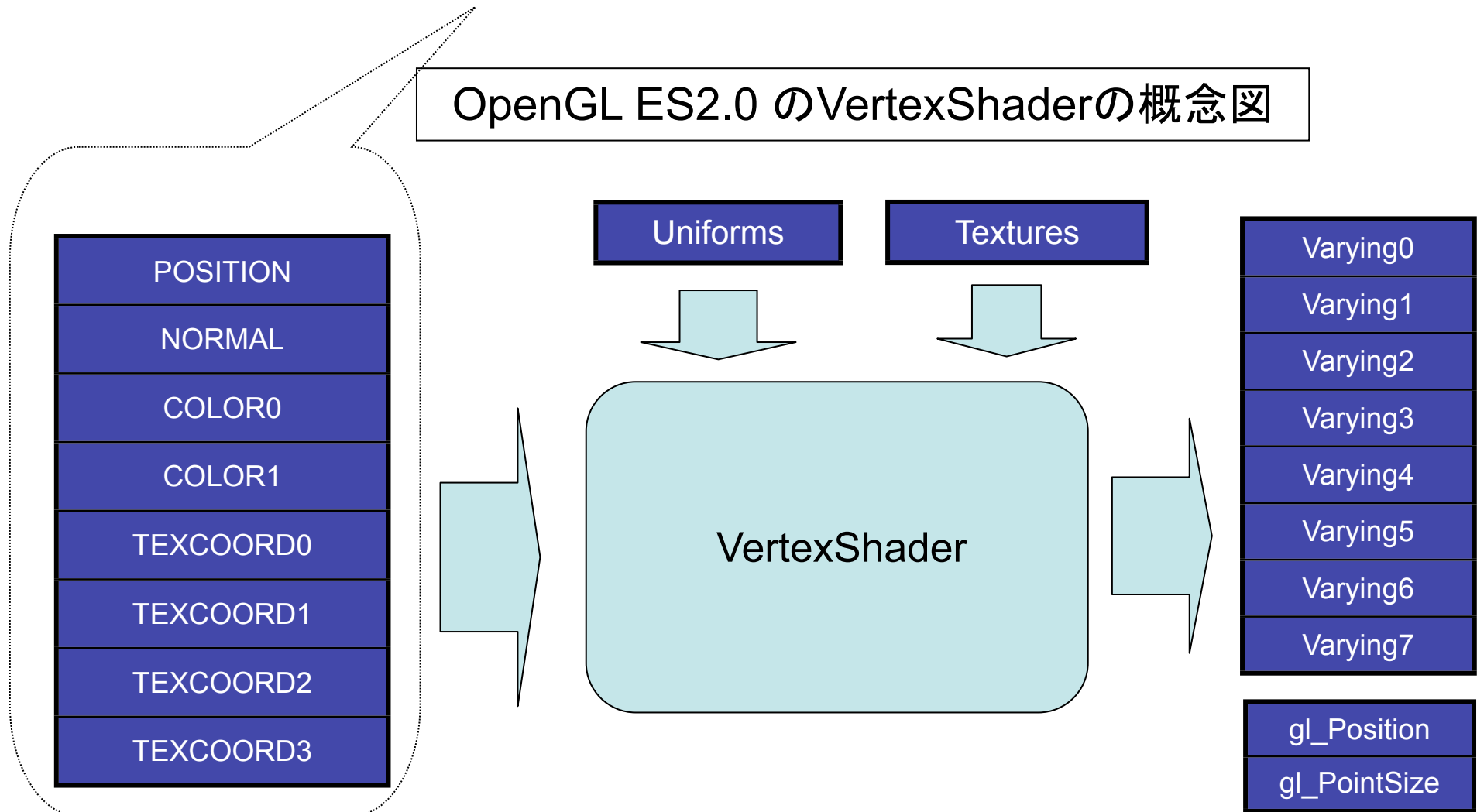
Attribute0	POSITION	頂点位置座標
Attribute1	NORMAL	頂点法線ベクトル
Attribute2	COLOR0	頂点カラー0
Attribute3	COLOR1	頂点カラー1
Attribute4	TEXCOORD0	テクスチャ座標0
Attribute5	TEXCOORD1	テクスチャ座標1
Attribute6	TEXCOORD2	テクスチャ座標2
Attribute7	TEXCOORD3	テクスチャ座標3

2.1 頂点アトリビュートのセマンティクスの決定



頂点アトリビュートのセマンティクスを決定することで、
VertexShaderに渡す **8種類のAttribute** が明確になる。

OpenGL ES2.0 のVertexShaderの概念図



2.1.3 GL_VertexFormatクラスのソース解説

- 2.1.2項で説明した頂点アトリビュートのセマンティクスを用いて、OpenGL ES2.0のVertexAttributeの設定と、
頂点データサイズの管理を行うためのクラス。

[GL_VertexFormatのソースを見る](#)

- このセッションでは以下の事柄について説明します

2.1 頂点アトリビュートのセマンティクスの決定

2.2 基本的なシェーディング (HalfLambertShading)

2.3 GoogleSketchUpの説明

2.4 COLLADAファイルの内容

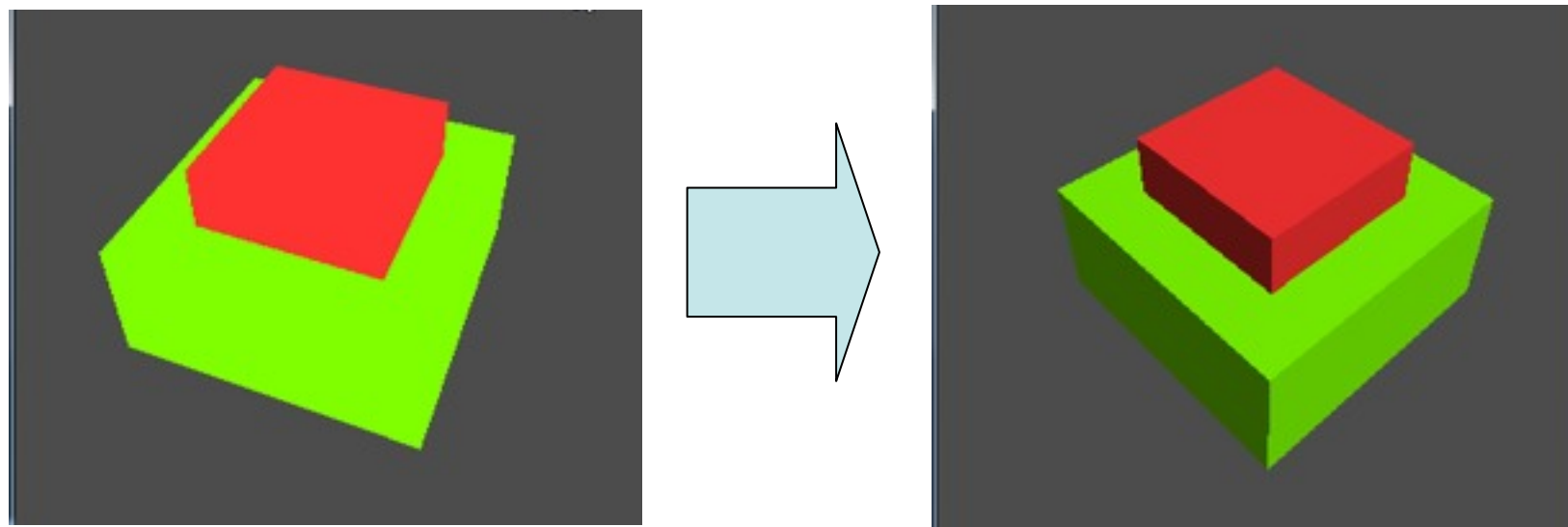
2.5 COLLADAファイルのコンバート

2.6 iPhone3GSでモデルの表示

Appendix Windows環境について

2.2.1 シェーディングの必要性

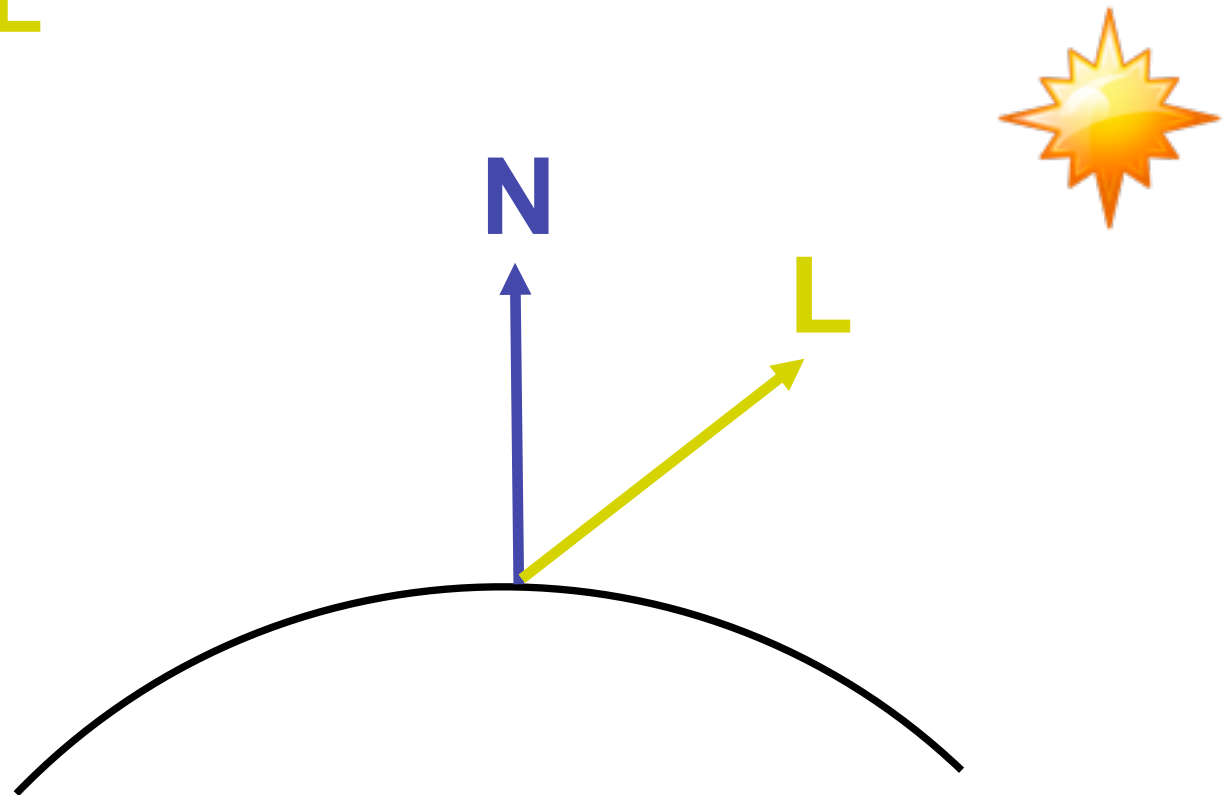
- モデルを表示する際に色を使っているが、これだけでは見にくい。
- 3D空間上の物を知覚するにはそのための情報が必要。
- そのひとつとして、陰影付け(シェーディング)がある。
- 今回はもっとも簡単なシェーディングの一つである、HalfLambertShadingを採用。



2.2.2 HalfLambertShadingの計算

– 計算に必要なもの。

- 法線ベクトル: **N**
- 光源ベクトル: **L**



– HalfLambertShadingの計算式

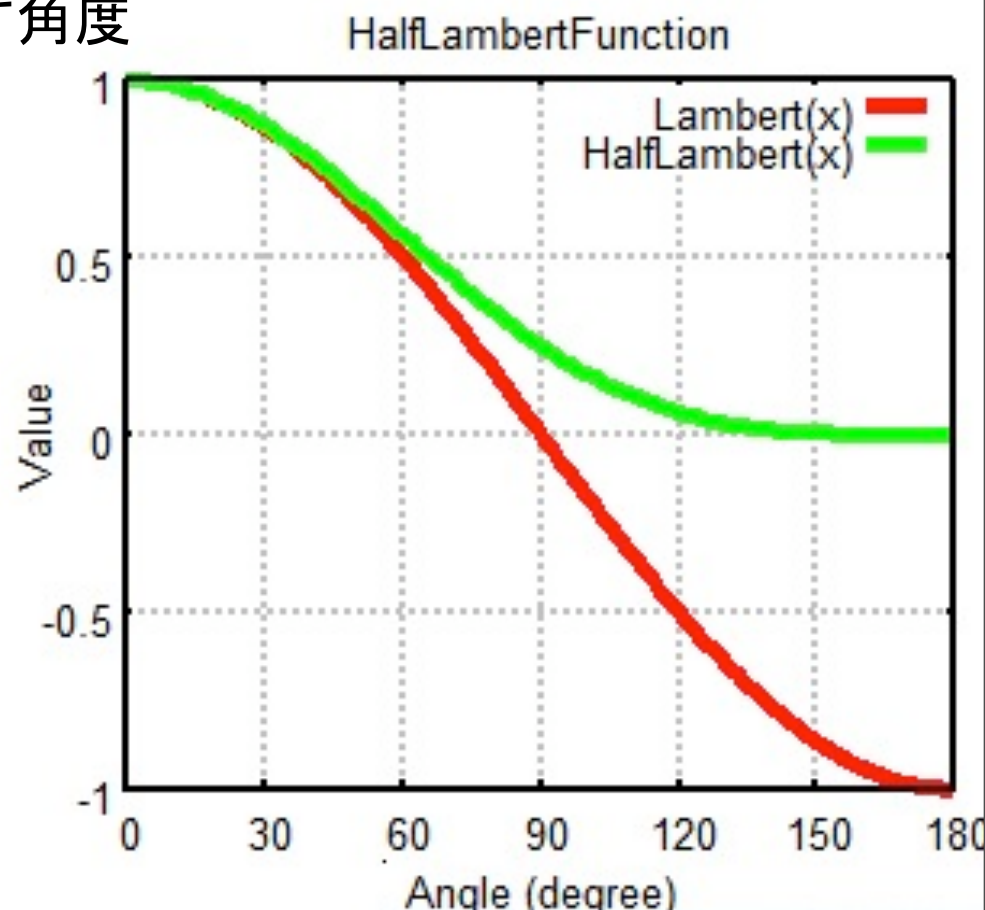
$$(\cos \theta * 0.5 + 0.5)^2$$

θ = 法線ベクトルと光源ベクトルの成す角度

ベクトルの内積を利用すれば

$$((N \cdot L) * 0.5 + 0.5)^2$$

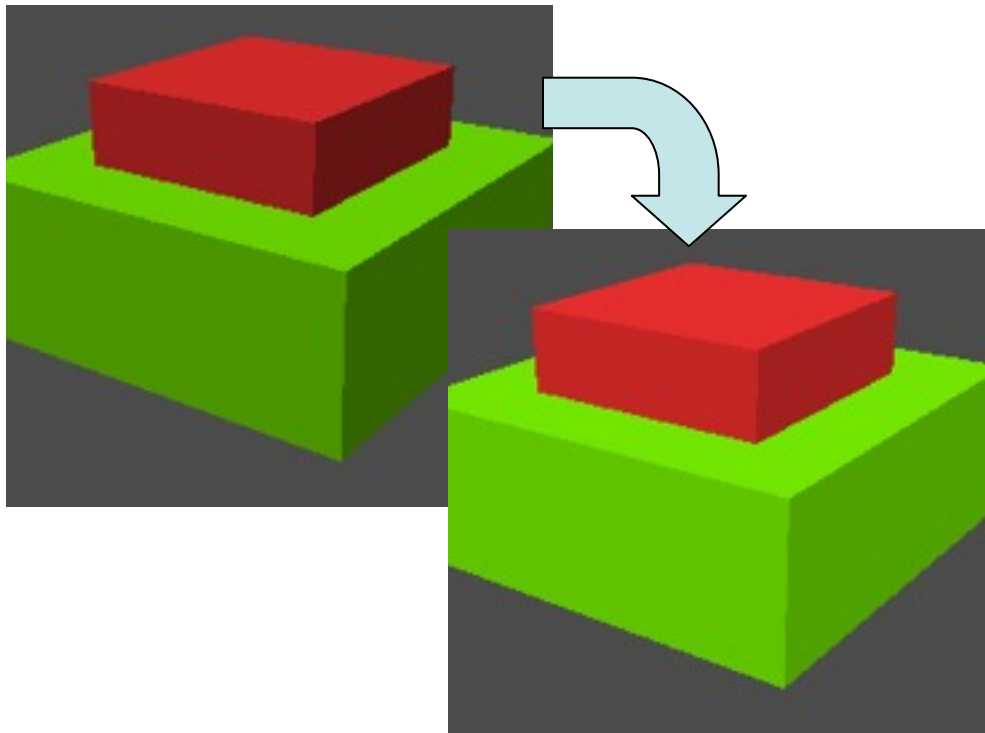
と書き直すことが出来る



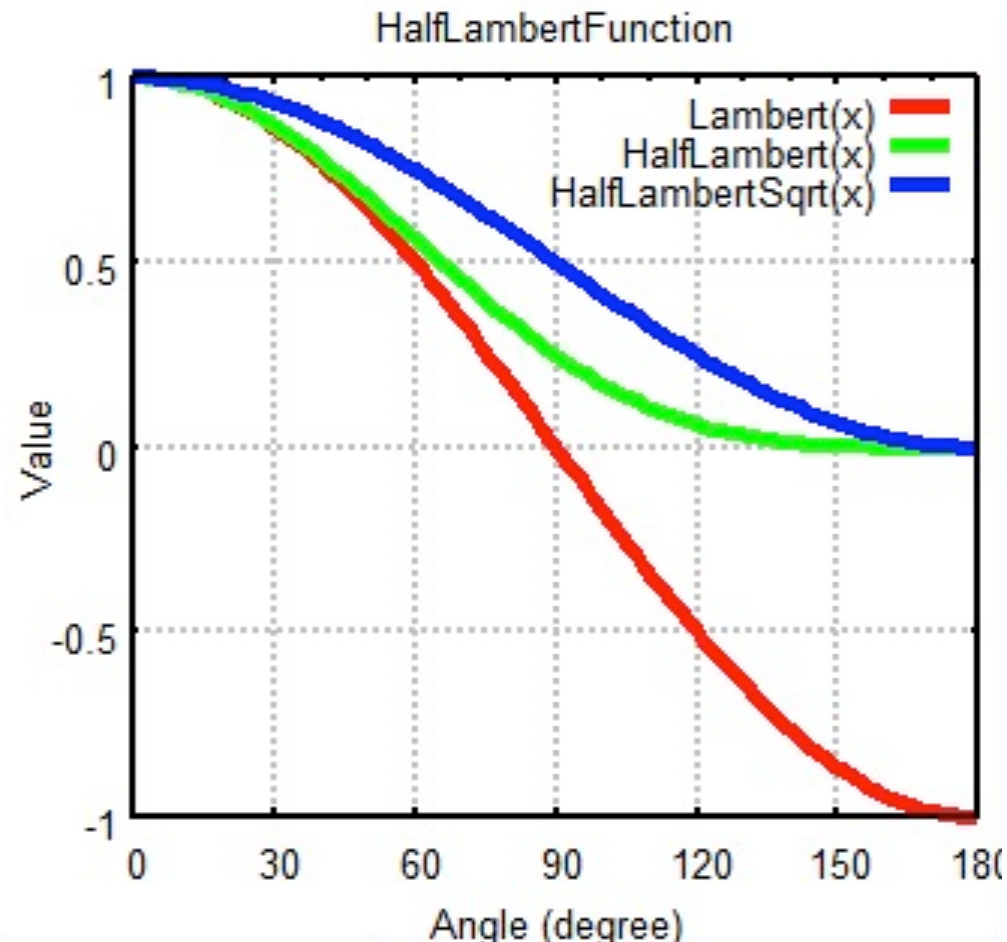
2.2 基本的なシェーディング (HalfLambertShading)

- 今回は環境光が無いので、HalfLambertShadingよりも明るいシェーディングを採用する。

$$((N \cdot L) * 0.5 + 0.5)$$



シェーダーのソースを見る



- このセッションでは以下の事柄について説明します
 - 2.1 頂点アトリビュートのセマンティクスの決定
 - 2.2 基本的なシェーディング(HalfLambertShading)

2.3 GoogleSketchUpの説明

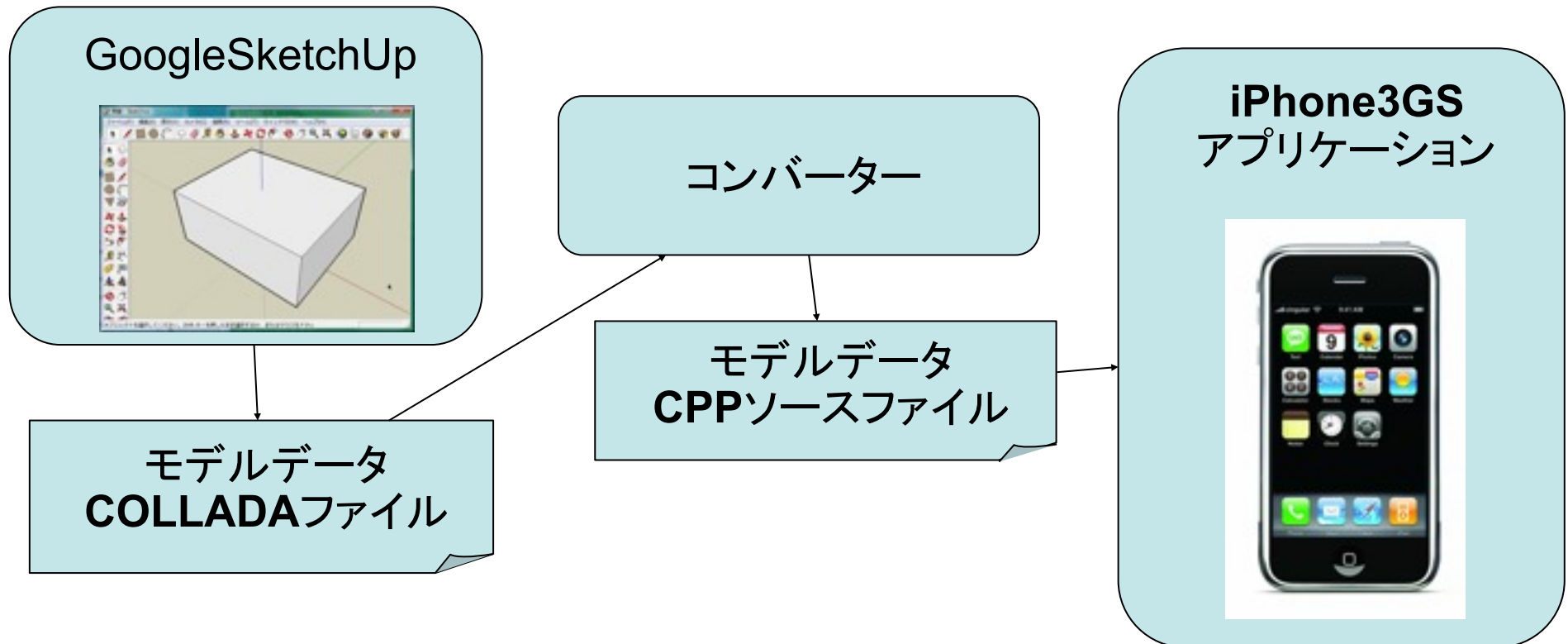
- 2.4 COLLADAファイルの内容
- 2.5 COLLADAファイルのコンバート
- 2.6 iPhoneでモデルの表示

Appendix Windows環境について

- 単純なモデルはプログラム内で記述して表示することが出来るが、自分の作った複雑な任意の形を表示するには、3Dモデリングツールが必要。
- 通称DCCツールと呼ばれるもので、一般的には非常に高価。
- 今回は無料で使えるGoogleSketchUpを採用。

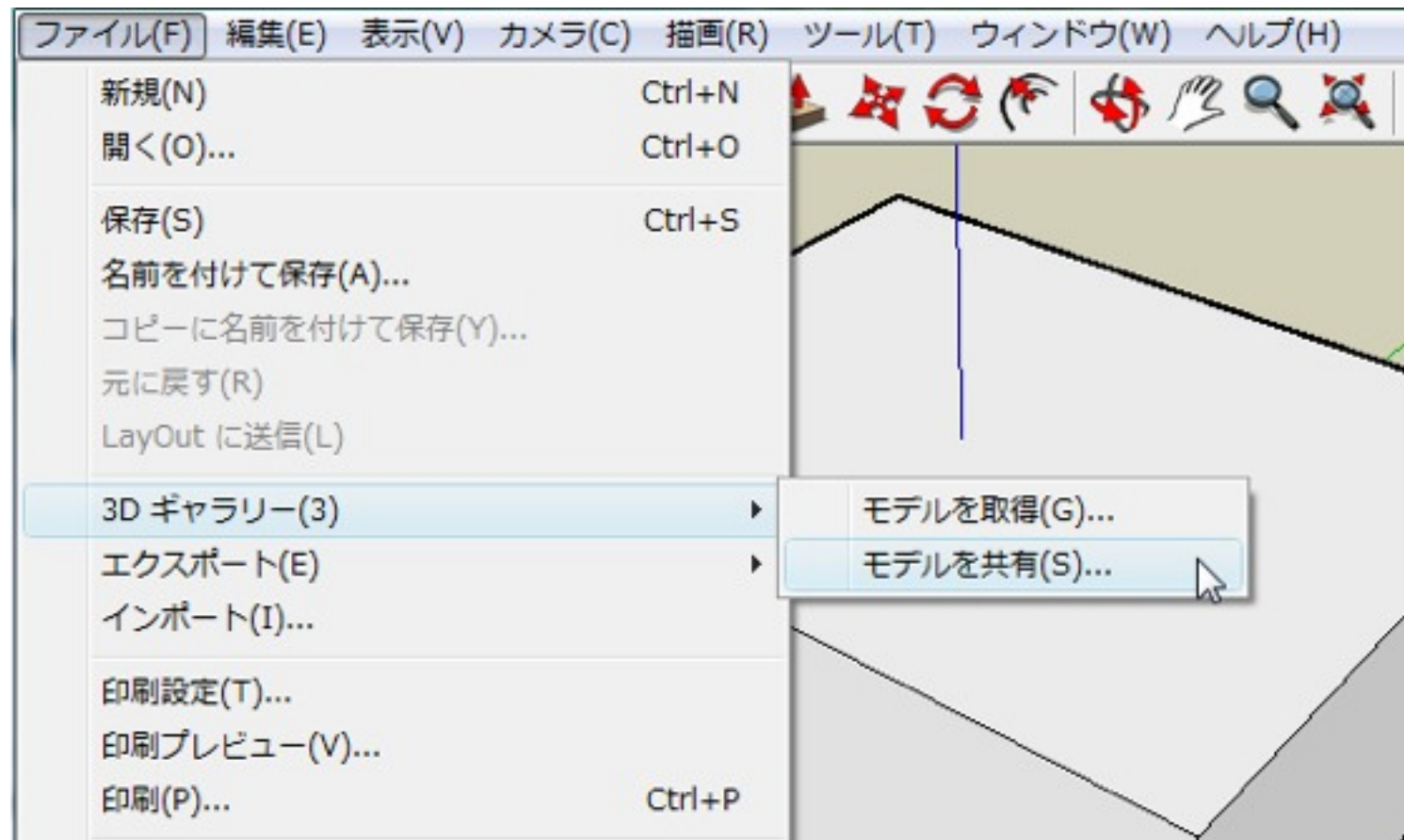
2.3.1 GoogleSketchUpからのデータの流れ

- GoogleSketchUpからCOLLADAというファイルを出力して、コンバーターで処理。
- 処理したCPPソースファイルをiPhone3GSアプリケーションで利用して表示する。



2.3.2 COLLADAファイルのエクスポート

- GoogleSketchUp上でモデルを作成する。
- [ファイル]->[3Dギャラリー]->[モデルを共有]を実行する。



2.3 GoogleSketchUpの説明



- 別Windowが開くので、Googleアカウントでログインする。
- モデルをアップロードする。



2.3 GoogleSketchUpの説明



- Webブラウザを別に起動する。
- Googleアカウントでログインして3Dギャラリーからモデルをダウンロードする。
- その際にCollada(zip)を選択する

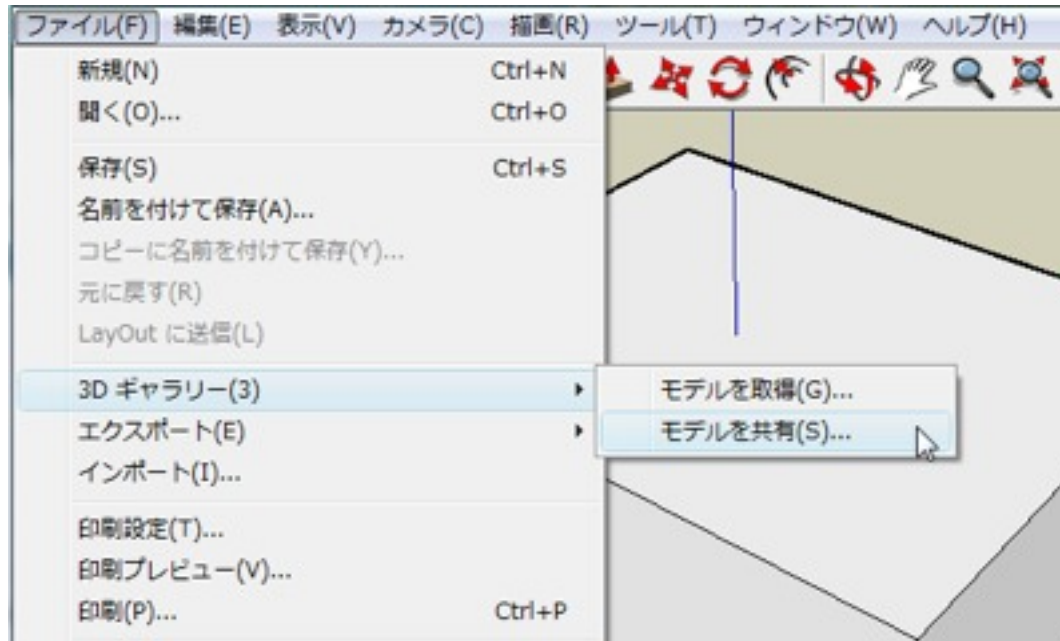


- 入手したzipファイルの中のmodelsフォルダ内に「dae」という拡張子のファイルがある。
- それがCOLLADAファイル。

実際にCOLLADAを出力してみる

2.3.3 GoogleSketchUpの便利な所

- Googleの3Dギャラリー上にある豊富なモデル群を検索 & インポートできる。
- [ファイル]->[3Dギャラリー]->[モデルを共有]を実行。



- このセッションでは以下の事柄について説明します
 - 2.1 頂点アトリビュートのセマンティクスの決定
 - 2.2 基本的なシェーディング(HalfLambertShading)
 - 2.3 GoogleSketchUpの説明

2.4 COLLADAファイルの内容

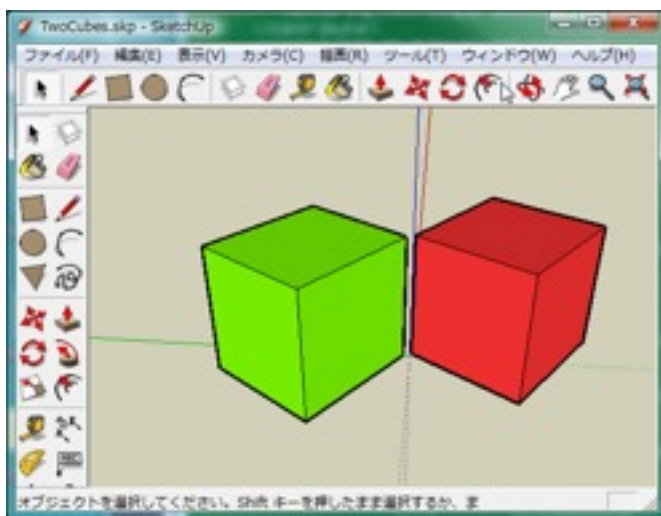
- 2.5 COLLADAファイルのコンバート
- 2.6 iPhoneでモデルの表示

Appendix Windows環境について

2.4.1 COLLADAファイルとは

- khronosが策定している3DアセットのためのXMLスキーマに則って記述されたXMLファイル。
<http://www.khronos.org/collada/>
- 現在使用されているバージョン1.5と1.4がある。
- 現在のGoogleSketchUpは1.4を出力する。

2.4.2 SketchUpの出力するCOLLADAファイルの中身 簡単なモデルをエクスポートしてファイルの中身を確認する。



```
<COLLADA version="1.4.1">
+ <asset></asset>
+ <library_materials></library_materials>
+ <library_effects></library_effects>
+ <library_geometries></library_geometries>
+ <library_nodes></library_nodes>
+ <library_cameras></library_cameras>
+ <library_visual_scenes></library_visual_scenes>
+ <scene></scene>
</COLLADA>
```


- <COLLADA>直下の8つの要素
 - <asset>
 - 出力したエクスポーター名や出力日時の情報を格納
 - <library_materials>
 - <material>要素:オブジェクトの材質、色などの特性を格納
 - <library_effects>
 - <effect>要素:materialの具体的な情報を格納
 - <library_geometries>
 - <geometry>要素:オブジェクトの外観、形状などの情報を格納
 - <library_nodes>
 - <node>要素:シーンのオブジェクトの部分的階層構造を格納
 - <library_cameras>
 - <camera>要素:カメラの光学的特性を格納(今回は未使用)
 - <library_visual_scenes>
 - <visual_scene>要素:シーンのオブジェクトの階層構造のルートを格納
 - <scene>
 - 格納されているシーン情報全体を格納

- COLLADAファイル内のアドレス構文の特徴
 - XPointerのIDリンク短縮形
 - <instance_***>は<library_***>内の要素を参照

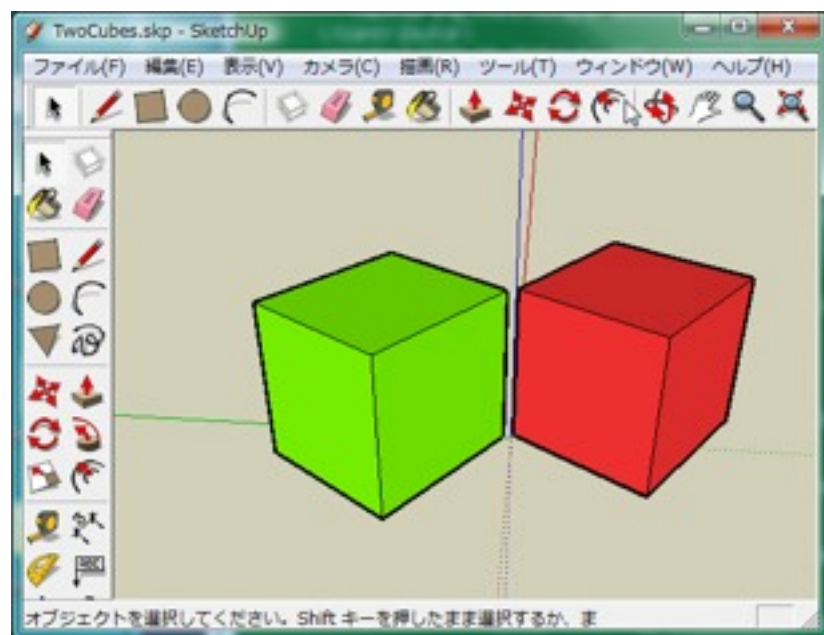
<instance_visual_scene>要素は
<library_visual_scenes>内の<visual_scene>を参照

```
- <library_visual_scenes>  
+ <visual_scene id="SketchUpScene" name="SketchUpScene"></visual_scene>  
</library_visual_scenes>  
- <scene>  
  <instance_visual_scene url="#SketchUpScene"/>  
</scene>
```

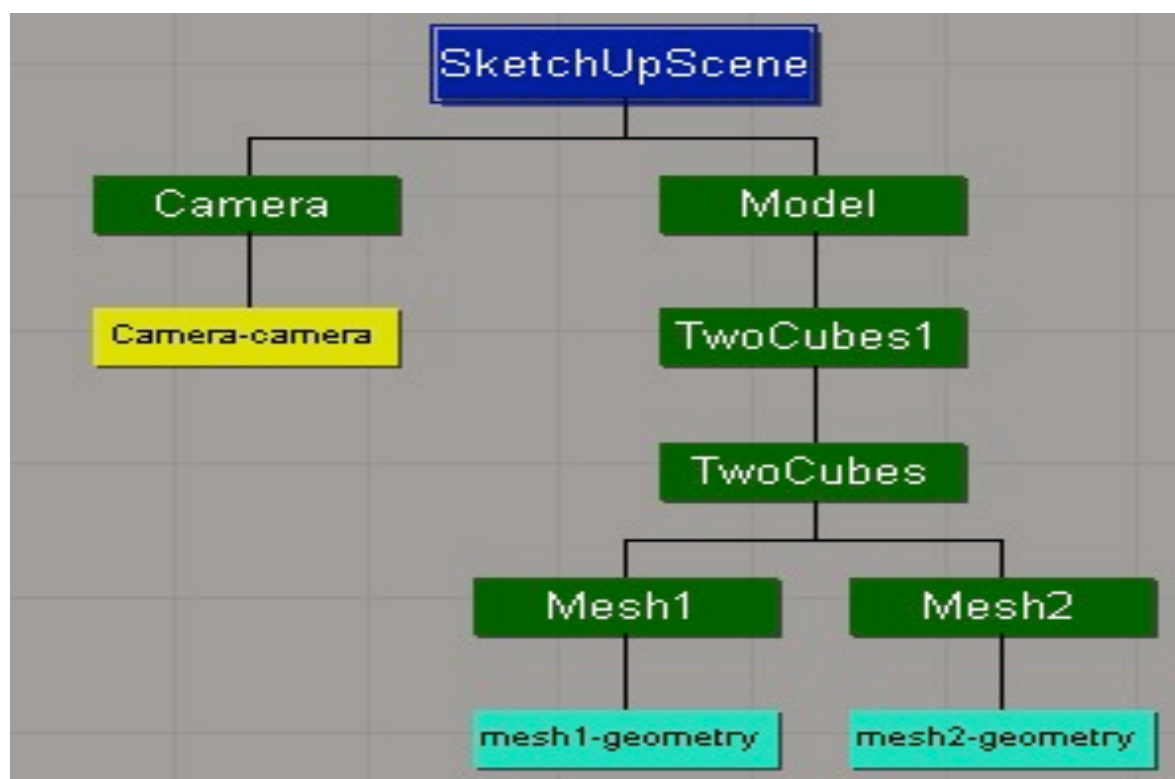
url="#SketchUpScene"はid="SketchUpScene"の要素を参照

- 簡単なCOLLADAファイルを例にしてブラウザで中を見てみる。
 - XMLファイルの観察にはWebブラウザが便利

実際にCOLLADAファイルを見る



ノードの階層構造



2.4.3 COLLADAファイルの注意点

- COLLADAファイルは現在非常に多岐にわたり使用されているが、データの記述の仕方は各々出力するツールによって異なる。
- 同じCOLLADAという呼び名だが、等価に扱うことは出来ないので注意。

- このセッションでは以下の事柄について説明します

2.1 頂点アトリビュートのセマンティクスの決定

2.2 基本的なシェーディング(HalfLambertShading)

2.3 GoogleSketchUpの説明

2.4 COLLADAファイルの内容

2.5 COLLADAファイルのコンバート

2.6 iPhone3GSでモデルの表示

Appendix Windows環境について

- COLLADAファイルのデータをターゲット上で速やかに利用するためにはコンバートする必要がある。
 - 頂点配列の作成
 - 空間の解釈の変換
 - プログラムで扱いやすいデータ形式

2.5.1 頂点配列の作成

COLLADA内では頂点データの各要素が別々にインデックス化されている。

頂点位置配列

X	Y	Z	X	Y	Z
---	---	---	---	---	---	------

頂点法線配列

X	Y	Z	X	Y	Z
---	---	---	---	---	---	------

頂点インデックス配列

0	0	1	0	2	0	3	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---



OpenGLではインデックス配列は一つしか使えないので、頂点配列を作り直す必要がある。

頂点データ配列

X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	...
---	---	---	---	---	---	---	---	---	---	---	---	-----

頂点インデックス配列

0	1	2	3	0	1
---	---	---	---	---	---

コンバーターの該当部分のソースを見る

2.5.2 空間の解釈を変換

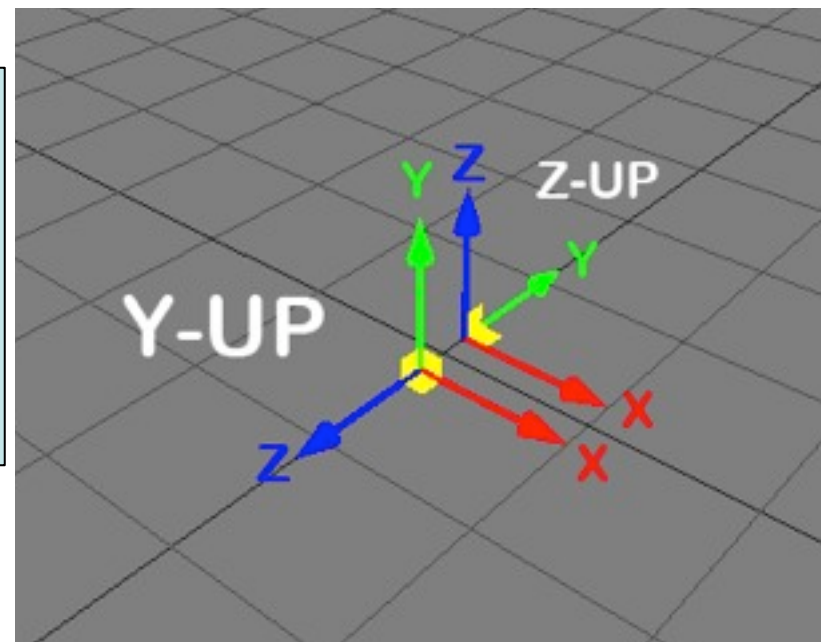
- SketchUp内ではZ-UP(後述)を採用しているが、今回のチュートリアルではY-UP(後述)を採用しているので、変換が必要。
 - Z-UPとは、概念的な上方向がZ軸の向きの世界
 - Y-UPとは、概念的な上方向がY軸の向きの世界

Z-UPからY-UPへのベクトルの変換

X軸成分: Z-UPのX軸成分

Y軸成分: Z-UPのZ軸成分

Z軸成分: Z-UPのY軸成分の反対成分



2.5.3 プログラムで扱いやすいデータ形式

- 前の項を踏まえて変換したデータを、今回はC++のソースファイルとして出力する。

コンバーターの出力ファイルを見る

- このセッションでは以下の事柄について説明します
 - 2.1 頂点アトリビュートのセマンティクスの決定
 - 2.2 基本的なシェーディング(HalfLambertShading)
 - 2.3 GoogleSketchUpの説明
 - 2.4 COLLADAファイルの内容
 - 2.5 COLLADAファイルのコンバート

2.6 iPhone3GSでモデルの表示

Appendix Windows環境について

- 前項で作成したコンバート結果のソースファイルをリンクして、実際にOpenGL ES2.0を用いて表示する。
 - GL_Primitiveクラスのソース解説
 - 描画前の初期化处理
 - 描画処理

2.6.2 GL_Primitiveクラスのソース解説

- GL_VertexFormatと、頂点バッファオブジェクトとインデックスバッファオブジェクトとモデル座標系のマトリクスとオブジェクトの描画色を管理するクラス。

[GL_Primitiveクラスのソースを見る](#)

2.6.3 描画前の初期化

- 描画の前に、GL_Primitiveクラスを用いて、頂点バッファオブジェクトとインデックスバッファオブジェクトを作成する作業を行う。

[初期化処理のソースを見る](#)

2.6.4 描画

- View/Projマトリクスに、モデル座標系の変換マトリクスをかけてUniformに設定する。
- GL_PrimitiveクラスのRender()メソッドを呼び出す。

描画処理のソースを見る

- このセッションでは以下の事柄について説明します
 - 2.1 頂点アトリビュートのセマンティクスの決定
 - 2.2 基本的なシェーディング(HalfLambert)
 - 2.3 GoogleSketchUpの説明
 - 2.4 COLLADAファイルの内容
 - 2.5 COLLADAファイルのコンバート
 - 2.6 iPhone3GSでモデルの表示

Appendix Windows環境について

- 今回はMacOSXとiPhone3GSを中心に説明しましたが、WindowsでもOpenGL ES2.0をエミュレーターを用いて試すことができます。

–コンパイラ

- VisualStudio 2008 ExpressEditon
- <http://www.microsoft.com/japan/msdn/vstudio/express/>
- WindowsLiveIDを取得してダウンロード

– OpenGL ES2.0エミュレーター

- Imagination Technologies社のHP
- <http://www.imgtec.com/powervr/insider/sdkdownloads/index.asp>
- アカウント取得が必要だが無料で手に入る。