

Meshes (メッシュ) — ufbx ドキュメント日本語訳

概要

Meshes (メッシュ) (`ufbx_mesh`) は、ポリゴンジオメトリデータを保持する要素です。

ufbx では次の用語を使用します：

用語	説明
Vertex (頂点)	位置情報を持つ頂点。3Dモデリングソフトで選択可能な頂点に相当。
Index (インデックス)	頂点にUV・法線・カラーなどの属性を組み合わせたもの。
Face (面)	1枚の平面 (三角形・四角形・N-gon) を構成するインデックスの範囲。

同じ頂点が複数の面から参照されることがあります。各参照では異なるインデックスを持ち、これにより **同一頂点に異なるUVや法線を設定** できます。

メッシュのデータ構造

メッシュデータは各種属性として格納されます：

- `ufbx_mesh.vertex_position` — 頂点座標
- `ufbx_mesh.vertex_normal` — 頂点法線
- `ufbx_mesh.vertex_uv` — 頂点UV座標

各属性には独自の **インデックス配列** (`ufbx_vertex_attr.indices[]`) と **値配列** (`ufbx_vertex_attr.values[]`) があり、インデックス指定で値を取得できます：

```
data[indices[index]]
```

あるいはヘルパー関数 `ufbx_get_vertex_vec3()` や、C++/Rust の `attrib[index]` 構文でもアクセス可能です。

描画例

以下は仮想的な即時モードポリゴンAPIを使ってメッシュを描画する例です。

```
void draw_polygons(ufbx_mesh *mesh)
{
    for (ufbx_face face : mesh->faces) {
        begin_polygon();

        // ポリゴンの各コーナーをループ
        for (uint32_t corner = 0; corner < face.num_indices; corner++) {

            // 各コーナーに対応するインデックス
            uint32_t index = face.index_begin + corner;

            // 頂点属性を取得
            ufbx_vec3 position = mesh->vertex_position[index];
            ufbx_vec3 normal = mesh->vertex_normal[index];
            ufbx_vec2 uv = mesh->vertex_uv[index];

            polygon_corner(position, normal, uv);
        }

        end_polygon();
    }
}
```

```
}
```

Materials (マテリアル)

1つのFBXメッシュには、異なる部位に複数のマテリアルが割り当てられる場合があります。

`ufbx_mesh.face_material[]` には、面ごとのマテリアルインデックスが格納されており、これを使って `ufbx_mesh.materials[]` にアクセスできます。

ただし、正確な結果を得るには `ufbx_node.materials[]` を使うのが推奨です。

理由は、同じメッシュが異なるマテリアルでインスタンス化されることがあるためです。

ゲームエンジンでは、マテリアル境界ごとにメッシュを分割する必要がある場合があります。

ufbx ではこれを容易にするために `ufbx_mesh.material_parts[]` を提供しています。

これには、各マテリアルごとの面数・三角形数・面リストが含まれます。

マテリアルが存在しない場合でも、便宜上1つのマテリアルパートが存在します。

例：GPU用フォーマットへの変換

以下は、メッシュデータを GPU向けインデックス付きフォーマット に変換する例です。
使用しているヘルパー関数：

- `ufbx_triangulate_face()` : 面を三角形化してインデックス配列を生成
- `ufbx_generate_indices()` : 頂点配列を重複排除し、インデックスバッファを生成

```
// GPU向け頂点構造体
// 実際にはよりコンパクトな型を使うべきです。
// `ufbx_real` はデフォルトで64bitです。
struct Vertex {
    ufbx_vec3 position;
    ufbx_vec3 normal;
    ufbx_vec2 uv;
};

void convert_mesh_part(ufbx_mesh *mesh, ufbx_mesh_part *part)
{
    std::vector<Vertex> vertices;
    std::vector<uint32_t> tri_indices;
    tri_indices.resize(mesh->max_face_triangles * 3);

    // このマテリアルを使用している各面を処理
    for (uint32_t face_index : part->face_indices) {
        ufbx_face face = mesh->faces[face_index];

        // 面を三角形化
        uint32_t num_tris = ufbx_triangulate_face(
            tri_indices.data(), tri_indices.size(), mesh, face);

        // 各三角形の頂点を取得
        for (size_t i = 0; i < num_tris * 3; i++) {
            uint32_t index = tri_indices[i];

            Vertex v;
            v.position = mesh->vertex_position[index];
            v.normal = mesh->vertex_normal[index];
            v.uv = mesh->vertex_uv[index];
            vertices.push_back(v);
        }
    }

    assert(vertices.size() == part->num_triangles * 3);

    // 頂点ストリームを生成
    ufbx_vertex_stream streams[1] = {
        { vertices.data(), vertices.size(), sizeof(Vertex) },
    };
    std::vector<uint32_t> indices;
```

```

indices.resize(part->num_triangles * 3);

// 頂点重複を削除し、インデックスバッファを生成
size_t num_vertices = ufbx_generate_indices(
    streams, 1, indices.data(), indices.size(), nullptr, nullptr);

vertices.resize(num_vertices);

create_vertex_buffer(vertices.data(), vertices.size());
create_index_buffer(indices.data(), indices.size());
}

```

Attributes (属性)

上記の属性に加えて、FBXメッシュには他の属性も存在します。
多くの属性は **頂点ごと** (または**インデックスごと**) に定義されていますが、
一部は **面単位**・**エッジ単位** のデータも含まれます。

エッジはオプションで、`ufbx_mesh.edges[]` で2つのインデックス間を定義します。

頂点 (インデックス単位、最大 `ufbx_mesh.num_indices`)

フィールド	内容
<code>ufbx_mesh.vertex_position</code>	頂点座標
<code>ufbx_mesh.vertex_normal</code>	法線ベクトル
<code>ufbx_mesh.vertex_tangent</code>	接線方向 (Tangent) UV.x
<code>ufbx_mesh.vertex_bitangent</code>	接線空間UV.y
<code>ufbx_mesh.vertex_uv</code>	UV座標 (第1セット)
<code>ufbx_mesh.vertex_color</code>	頂点カラー (第1セット)
<code>ufbx_mesh.vertex_crease</code>	サブディビジョン用クリース値

面 (最大 `ufbx_mesh.num_faces`)

フィールド	内容
<code>ufbx_mesh.face_material</code>	面ごとのマテリアル
<code>ufbx_mesh.face_group</code>	ポリゴングループ
<code>ufbx_mesh.face_smoothing</code>	スムーズシェーディングフラグ
<code>ufbx_mesh.face_hole</code>	穴として扱うかどうか

エッジ (最大 `ufbx_mesh.num_edges`)

フィールド	内容
<code>ufbx_mesh.edge_smoothing</code>	法線生成用スムーズフラグ
<code>ufbx_mesh.edge_visibility</code>	編集用のエッジ表示フラグ
<code>ufbx_mesh.edge_crease</code>	サブディビジョン用エッジクリース

🔄Revision #2

★Created 28 October 2025 09:17:48 by youe2

🔪Updated 2 June 2026 18:03:11 by youe2