

FBXとufbxで座標合わせをしたときのメモ（どうせUVのときまた頭おかしくなる用）

結論（このリポの現状）

- 座標系：FBX読み時に **右手系・Y up** へ正規化してからロードしています（`opts.target_axes = ufbx_axes_right_handed_y_up`）。

描画側も **RHのLookAt / PerspectiveFov** を使っているため、**手系・Up軸は一致**しています。

UfbxAnimSkeleton



D3DApp

- 行列の積み上げ：`EvaluateAt()` で **local** を親の **world** に掛ける ($W = L * Pw$) という標準的な階層合成を行い、かつ **Skin = $W * InvBind$** でスキン行列を作っています。FBX特有の pre/post-rotation は `ufbx_evaluate_transform()` が面倒を見てくれるので、**アニメの動きはFBXと同一**になります。

UfbxAnimSkeleton

- 行列レイアウトの差：`ufbx_matrix (4x3)` を **DirectX** の **XMFLOAT4X4**（行メジャ/平行移動は **_41/_42/_43**）に詰め替えており、**転置+配置は関数で一貫**させています。可視化はこの **_41/_42/_43** を使って正しくワールド位置を取っています。

UfbxAnimSkeleton



BoneVisualizer

- カメラ：ボーンのワールド行列群から **中心と半径を推定** → その球が画面に“ちょうど入る”距離に **RHビュー&プロジェクションを自動計算**。画角・距離が素直な値なので、**モデリングビュー相当の見え方**になります（もちろんツールのプリセットFOVと同じにすればより一致）。

WinMain



WinMain

- **タイトル/FPS**：Unicode タイトル更新や 60fps リミッタは描画の見え方に影響しませんが、WAPI での文字化け対策含め堅牢です。

D3DApp

同じ見え方にするためのチェックリスト

1. 手系/Up軸の一致

- すでに FBX ロードで **RH, Y-up** へ正規化済み。描画も RH を使用。→ OK。

`(opts.target_axes = ufbx_axes_right_handed_y_up / XMMatrixLookAtRH & XMMatrixPerspectiveFovRH)`


UfbxAnimSkeleton



D3DApp

2. 行列の詰め替え規約

- `ToDX_4x4_FromUfbxMatrix4x3()` で **回転・並進を DirectX のフィールド配置**に揃える。可視化側は **_41/_42/_43** から位置

を取得。→ OK。 

UfbxAnimSkeleton



BoneVisualizer

3. アニメ評価の一貫性

- `ufbx_evaluate_transform()` を local として取り、親子で掛け上げて world を得る。FBXの pre/post 回りは ufbx が吸収。

→ OK。 

UfbxAnimSkeleton

4. カメラ/FOV

- ツール側の FOV と合わせると「同じ見え」。コードは対象を包む球で距離計算して RH の VP を生成。ツール側の

FOV が 45° なら `XM_PIDIV4` と一致させる。 

WinMain



WinMain

5. クリッピング/遠近

- `zn` は距離に応じて 0.5 以上、`zf` は余裕を持たせて設定。`D3DApp` のリサイズ時も `far=10000`。極端に大きい/小さい

モデルでも破綻しにくい。 

WinMain



D3DApp

6. 単位 (スケール)

- いまのコードは単位スケールを特別扱っていません (メートル/センチの換算はしない)。ただしカメラ自動フィットがあるので見た目は実用的に問題なし。モデル同士の相対スケール厳密一致が必要なら、Ufbx の

`scale_factor` 等を使うスケーリング導入を検討 (今は未使用)。 

ufbx


7. Z反転の重複適用に注意

- ルートに `FlipZ` を掛けるコードはコメントアウトのまま。`target_axes` で RH 化しているので追加の反転は不要。もし

モデルが「鏡像」に見えたら、反転の二重適用がないかをまず疑う。 

UfbxAnimSkeleton

8. 可視化の基準


- ボーンラインは親→子の `_41/_42/_43` の位置を結ぶだけなので、座標が正しければ「骨の形」は一致します。 

BoneVisualizer

ありがちな“ズレ”と対処

- 左右反転/鏡像に見える

→ どこかで手系変換+Z反転を二重にしているかチェック。今のコードでは `target_axes` が RH 化を担うので `FlipZ` を ON

にしないのが正解。 

UfbxAnimSkeleton

- 大きすぎ/小さすぎで消える

→ `FitCameraToBounds()` は半径から距離を幾何学的に決定しているため、初期表示は安定。遠近クリップは `0.1~10000` なの

で、通常用途は問題なし。 

WinMain



D3DApp

- モデリングツールの FOV と違う

→ ツール側の FOV に合わせて `XM_PIDIV4` の値を変更すれば一致が近づきます。 

D3DApp

まとめ

- FBX読み込時に RH/Y-up へ正規化し、DirectX 側も RH の VP、行列の詰め替えも統一、アニメ評価は ufbx に準拠しているの
で、理屈上はモデリングソフトの見えと一致します。

- 残る実務的な差異は **FOV/単位スケール/反転の二重適用**。この3点を上のチェックリスト通りに押さえれば、**同じ見え方に** 落ち着きます。

🕒Revision #1

★Created 3 November 2025 08:17:58 by youe2

✎Updated 2 June 2026 18:27:12 by youe2