mod_SC3D_A 説明

Index

- 一言説明
- 数学的な事前定義
- カメラや被写体に関する説明
- 命令・関数リファレンス

＜ 一言説明 ＞

HSP の標準命令だけで 3D 描画を行う為の座標計算を行うモジュールです。

＜ 依存するモジュール ＞

mod_CalcNum

必ず上記のモジュールを先にインクルードしてください。

＜ 数学的な事前定義 ＞

あるベクトル →V について、「→V まわりの回転」と言うときは、時計の背中に→V の尻を鉛直にくっつけた時に時計の針が回る向きを正の回転とします。

角度は、特に断りのない限りラジアンで表現します。

＜ カメラや被写体に関する説明 ＞

Index

- グローバル空間
  - グローバル座標系
- カメラ
  - オイラー角による表現
  - axis-angle による表現
- カメラのスクリーンと HSP の画面の関係
- 線分
- 球
- 構造体
  - カメラ構造体

＜ グローバル空間 ＞

＜ グローバル座標系 ＞

このモジュールで扱うものの位置を表現するための共通の座標系です。
座標は(x_g ， y_g ， z_g)で表します。
原点は点 0_g で表します。

グローバル座標系における単位ベクトル(1, 0, 0), (0, 1, 0), (0, 0, 1)をそれぞれ
→e_g_x ， →e_g_y ， →e_g_z で表します。

基本的に、地面は x_g, y_g 平面とし、z_g 軸が点を指すイメージ。そうでないといけない訳ではないですが、その想定でこのモジュールは作ってあります。

＜ カメラ ＞

「眼点」とその背後のスクリーンから成ります。これでグローバル空間内のものを見ます。

カメラには「カメラ座標系」という独自の座標系がありますが、大きさの尺度はグローバル座標系と同じです。(すなわち、グローバル座標系での 1 とカメラ座標系での 1 は同じ大きさ。)
座標は(x_c ， y_c ， z_c)で表します。
原点は点 0_c で表します。
カメラ座標系座標が(a, b, c)であるからといってグローバル座標もそうだとは限らないことに留意してください。カメラが傾いていたりするとグローバル座標とは異なってきます。

カメラ座標系における単位ベクトル(1, 0, 0), (0, 1, 0), (0, 0, 1)をそれぞれ →e_c_x ，
→e_c_y ， →e_c_z で表します。

x_c, y_c 平面がスクリーンです。カメラ座標系座標(0, 0, 1)の位置に「眼点」E_c があります。
よって、カメラにとっての「前方」は z_c 軸の正の向きとなります。
ベクトル 0_c→E_c を「カメラの目線ベクトル」とします。大きさは 1 です。
カメラにとっての「天」は y_c 軸の正の向きにあります。

被写点と眼点を結ぶ直線とスクリーンとの交点が「像」です。

視野角は 120°です。

カメラには「カメラの基本状態」という状態が存在します。これは、次の条件の連立により定義されます。

- カメラ座標系の原点がグローバル座標系の原点と重なる。
- カメラ座標系の x_c, y_c, z_c 軸の正の向きがそれぞれグローバル座標系の x_g, y_g, z_g 軸の正の向きと一致する

カメラの状態は、位置と姿勢で決まります。
位置については、0_c のグローバル座標を決めればよいです。

姿勢は、→e_c_x ， →e_c_y ， →e_c_z で定まります。

姿勢を決定する方法は以下の 3 つを用意しています。

- オイラー角による表現
- axis-angle による表現
- カメラ座標系単位ベクトルの手動設定

＜ オイラー角による表現 ＞

カメラの基本状態をベースとして、x_c, y_c, z_c 軸回りに、この順にカメラを回転させます。
x_c, y_c, z_c 軸回りにカメラ座標系における単位ベクトルを回転させればよいです。
この操作には専用の命令が用意されています。

＜ axis-angle による表現 ＞

目線ベクトルを定め、その回りにカメラを回転させます。但し、回転前の基準は、カメラが「最も立っている」状態を 0 とします。

目線ベクトル回りにカメラ座標系における単位ベクトルを回転させればよいです。

ターゲットの点を T とします。
まず ベクトル 0_c→T を求め、その長さを 1 に調節したものを →e_c_z とします。

次にカメラの頭の向きを決めるために →e_c_y を決めます。これが少し複雑です。

（ｉ）→e_c_z が z_g と平行でない場合

→e_c_y と →e_g_z の内積が最大になるようにする。
詳しくは別資料を参照。

（ⅱ）→e_c_z が z_g と平行である場合

→e_c_y = →e_g_y とする。

以上の操作には専用の命令が用意されています。

＜ カメラ座標系単位ベクトルの手動設定 ＞

これは強力ですが、危険も伴います。

空間ベクトルを十分に理解している方が使用する方法です。

本モジュールでは、

「vctr_e_c_x@SC3D_A_1」変数の要素(0),(1),(2) が →e_c_x のx,y,z成分
「vctr_e_c_y@SC3D_A_1」変数の要素(0),(1),(2) が →e_c_y のx,y,z成分
「vctr_e_c_z@SC3D_A_1」変数の要素(0),(1),(2) が →e_c_z のx,y,z成分

となっています。この変数の値を直接書き換えることで、カメラの姿勢を設定できます。
空間ベクトルを理解している方にとっては、これは**最も自由度の高い、優れた姿勢の表現方法**です。

但し、繰り返しますが、危険も伴います。甚だしく間違った値を指定すると最悪の場合内部エラーが発生するので、注意してください。

＜ カメラのスクリーンとHSPの画面との関係 ＞

(1〔px〕の大きさ)÷(グローバル座標系での1の大きさ) を「P/G定数 (Pixel/Global定数)」とします。
よって、スクリーン上で大きさがxの影は、画面では (P/G定数)*x〔px〕の大きさで描画すればよいです。

カメラの眼点とスクリーンと視野角の関係から、スクリーン上で実際に使用されるのは半径√3の円です。
個人的には、この円の直径が大体400pxに相当すれば使いやすいと思うので、
400 ： √3 ≒ 230 ： 1 より、P/G定数の値は 230 が良いと思います。

スクリーン上の像は実際の被写体の姿勢とは上下左右が逆になっているので、画面を描画するときは補正が必要です。
さらに、コンピュータの画面はy軸が下を向いていることを考慮すると、
HSPの画面のx ， y軸の正の向きはx_c ， y_c軸の正の向きと同じにとればよいです。

0_cと画面の中心の座標はユーザーの都合に任せます。
これらの座標計算はモジュール側で行うので、心配は要りません。

＜ 線分 ＞

線分の両端がいつも視野内に来るとは限りません。
一部しか見えないときもあります。
そのとき、視野円錐面と線分の共有点の座標が分かれば、その2点で作られる線分を描画すれば、
一部だけ映っているのと同じ事になります。
つまり、クリッピングするということです。
詳しくは別資料を参照してください。

＜ 球 ＞

0_cと球の中心との距離が分かれば、それに応じて円を描けば球のように見えます。

＜ 構造体 ＞

複数のデータをひとまとめにしたパックです。関連のあるデータをまとめて扱う目的で作られています。

＜ カメラ構造体 ＞

実数型配列変数です。
カメラの設定を行うときにデータの受け渡しに使います。
各要素の意味は以下の通りです。

要素	意味
(0),(1),(2)	0_c のグローバル座標
(3)	カメラの姿勢決定手段。 0 ： オイラー角 1 ： axis-angle
(4),(5), (6),(7)	姿勢制御用数値 （ｉ）オイラー角の場合 (4),(5),(6)にそれぞれ、x_c 軸,y_c 軸,z_c 軸まわりの回転角度を指定。 回転はこの順に行われることに留意。 (7)は無視。 （ii）axis-angle の場合 (4),(5),(6)にそれぞれ、カメラが見つめる点のをグローバル座標を指定。 (7)に、カメラがその点を見つめた後の、z_c 軸まわりの回転角度を指定。
(8),(9)	0_c の、HSP の画面上での座標。 すなわち、画面上での描画の中心の座標。
(10)	P/G 定数の値。230 が目安。

＜ 命令・関数リファレンス ＞

命令・関数一覧

- MSC3D_A_init
- MSC3D_A_SetCam
- MSC3D_A_GetShadow
- MSC3D_A_GetLineShadow
- MSC3D_A_GetSphere

命令・関数	説明
MSC3D_A_init	モジュールの初期化
MSC3D_A_SetCam	カメラの設定 ［書式］ MSC3D_A_SetCam 変数１ 変数１ ： カメラ構造体
MSC3D_A_GetShadow	座標変換。 グローバル空間内のある点Ａのスクリーンに映る像の情報を取得します。 ［書式］ MSC3D_A_GetShadow P1,P2,P3 ,変数１,P4 P1,P2,P3 ： Ａのグローバル座標 P4 ： 視野角チェック停止フラグ Ａが視野内にあるかどうかのチェックを行わずに、映るものとして計算するフラグ。 (0,1)=(チェックを行う,行わない) 通常は０を指定。 もし１を指定して、かつＡが眼点の後ろ側にある等して絶対に映らない場合は、計算がめちゃくちゃになり、エラーが起こる可能性が高い。 １を指定するのは、視野円錐スレスレのところにある点を確実に映したい等、特別な事情がある場合のみ。 変数１ ： 結果を格納する実数型配列変数 ++++++++++++++(変数１の構造)++++++++++++++ 要素 ｜ 説明 (0),(1),(2) ｜ 像のグローバル座標 (3),(4) ｜ 像のカメラ座標系座標 (5),(6) ｜ 像のHSPの画面上での座標 点がスクリーンに映る場合はstatの値が１になり、映らない場合は０になります。
MSC3D_A_GetLineShadow	座標変換。 グローバル空間内のある線分ABがカメラに写るかどうか、また、映るならその像の情報を取得します。 ［書式］

MSC3D_A_GetLineShadow P1,P2,P3 ,P4,P5,P6 ,変数１

P1,P2,P3 ： A の座標
P4,P5,P6 ： B の座標
変数１ ： 結果を格納する実数型配列変数

++++++++++++++(変数１の構造)++++++++++++++

要素	説明
(0)	クリッピングか行われたかどうか。 0 ： 行われなかった 1 ： 行われた
(1),(2),(3) (4),(5),(6)	クリッピングが行われた場合に記録される。 クリッピング後の線分の端点のグローバル座標。 (1),(2),(3)が１つ目の端点。 (4),(5),(6)が２つ目の端点。
(7),(8),(9) (10),(11), (12)	線分の像の端点のグローバル座標。 (7),(8),(9)が１つ目の端点。 (10),(11),(12)が２つ目の端点。
(13),(14), (15),(16)	線分の像の端点のカメラ座標系座標。 (13),(14)が１つ目の端点。 (15),(16)が２つ目の端点。
(17),(18) (19),(20)	線分の像の HSP の画面上での座標。 (17),(18)が１つ目の端点。 (19),(20)が２つ目の端点。

AB がスクリーンに写る場合は stat の値が１になり、映らない場合は０になります。

MSC3D_A_GetSphere

座標変換。
グローバル空間内のある球Ｓの、スクリーン上の像の情報を取得します。

ただし、Ｓの中心がスクリーンに映らない場合はＳは映らないものとして判断されます。

［書式］

MSC3D_A_GetSphere P1,P2,P3,P4 ,変数１

P1,P2,P3 ： の中心のグローバル座標
P4 ： 半径
変数１ ： 結果を格納する実数型配列変数

++++++++++++++(変数1の構造)++++++++++++++

要素	説明
(0),(1),(2)	像の中心のグローバル座標
(3),(4)	像の中心のカメラ座標系座標
(5),(6)	像の中心のHSPの画面上での座標
(7)	像(=円)のHSPの画面上での半径(px)

Sの中心がスクリーンに映る場合はstatの値が1になり、映らない場合は0になります。