

カメラの初期化について:

カメラの初期化は下記の関数を用いて行います。

njInitCamera (NJS_CAMERA *c)

カメラの各パラメータを次のように初期化します。

パラメータ	値	備考
位置 (px, py, pz)	(0、 0、 0)	ワールド座標原点。
視線の向き (vx, vy, vz)	(0、 0、 - 1)	画面奥。(ローカル Z 軸)
傾き (roll)	0 度	ワールド座標系の Y 軸に対して平行。
画角 (ang)	6 0 度	視野の水平方向の角度
ニアクリップ (n_clip)	- 1	
ファークリップ (f_clip)	- 6 0 0 0 0	
ローカル X 軸 (lx)	(1、 0、 0)	
ローカル Y 軸 (ly)	(0、 1、 0)	

*) 視線の向き (ローカル Z 軸)、ローカル X 軸、ローカル Y 軸はそれぞれワールド座標系でのベクトル値で表されます。

カメラ構造体

```
typedef struct {  
    Float          px,py,pz; /* 位置 */  
    Float          vx,vy,vz; /* 視線の向き */  
    Angle          roll;     /* 傾き */  
    Angle          ang;      /* 画角 */  
    Float          n_clip;   /* ニアクリップ */  
    Float          f_clip;   /* ファークリップ */  
    NJS_VECTOR     lx, ly;   /* ローカル座標軸 */  
} NJS_CAMERA;
```

カメラの位置について:

カメラは次の関数を用いてワールド空間の好きな座標に配置する事が出来ます。

(初期化時には (0 , 0 , 0) に配置されています。)

尚、カメラの位置を移動、回転させても視線の向き、及び傾きが変化する事はありません。

njTranslateCameraPosition(NJS_CAMERA *c, Float x, Float y, Float z)

カメラの位置を現在置かれている位置から X、Y、Z 軸に沿って移動します。

njRotateCameraPositionX(NJS_CAMERA *c, Float ang)

njRotateCameraPositionY(NJS_CAMERA *c, Float ang)

njRotateCameraPositionZ(NJS_CAMERA *c, Float ang)

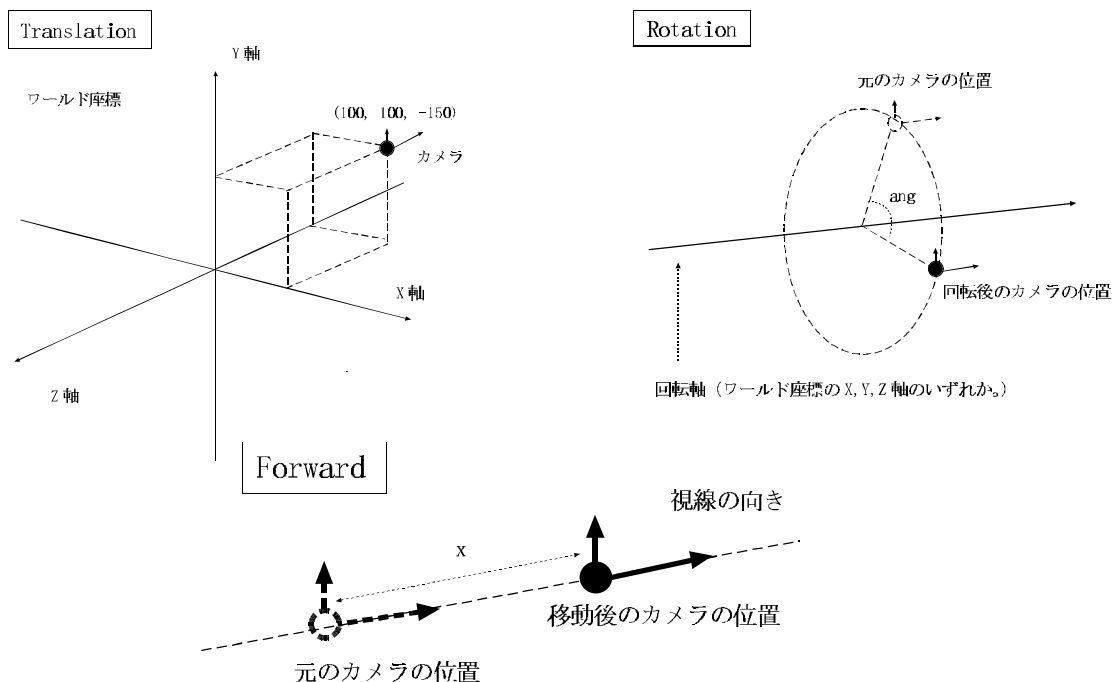
njRotateCameraPositionXYZ(NJS_CAMERA *c, Float ax, Float ay, Float az)

カメラの位置を現在置かれている位置からワールド座標の各軸を中心に ang だけ回転させた場所に移動します。njRotateCameraPositionXYZ 関数は X、Y、Z 軸の順にそれぞれ ax、ay、az だけ回転させます。

njForwardCameraPosition(NJS_CAMERA *c, Float x)

カメラを現在置かれている位置から視線の向きに向かって x だけ移動します。

黒い点はカメラを表し、点に付いている長い矢印は視線の向き、短い矢印はカメラの傾き (カメラの上方) を表しています。



カメラの視線について：

カメラの視線は向きと傾きの2つのパラメータで表され次の関数を用いて自由に操作する事が出来ます。（向きと傾きはお互いに影響を受けません。）

njPitchCameraInterest(NJS_CAMERA *c, Angle ang)

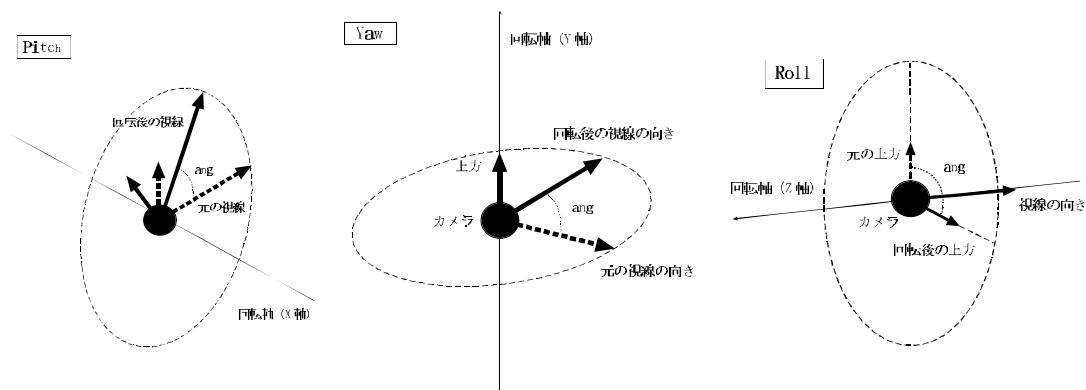
カメラのローカル座標系の X 軸に対して視線の向きを ang だけ回転します。

njYawCameraInterest(NJS_CAMERA *c, Angle ang)

カメラのローカル座標系の Y 軸に対して視線の向きを ang だけ回転します。

NjRollCamera(NJS_CAMERA *c, Angle ang)

カメラ自体の視線（ローカル座標系の Z 軸）を中心にカメラを回転します

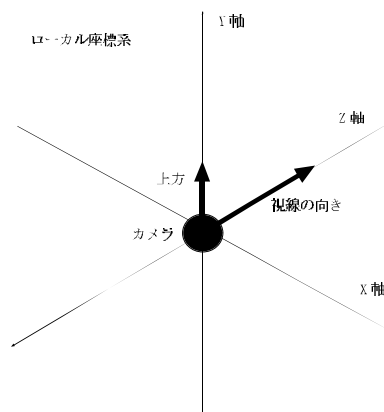


njPointCameraInterest(NJS_CAMERA *c, Float x, Float y, Float z)

カメラの視線をワールド空間の座標(x、y、z)に向けます。

ノート：

カメラのローカル座標系は視線の向きを Z 軸、カメラ自体の上下を Y 軸とした物です。



カメラの画角について:

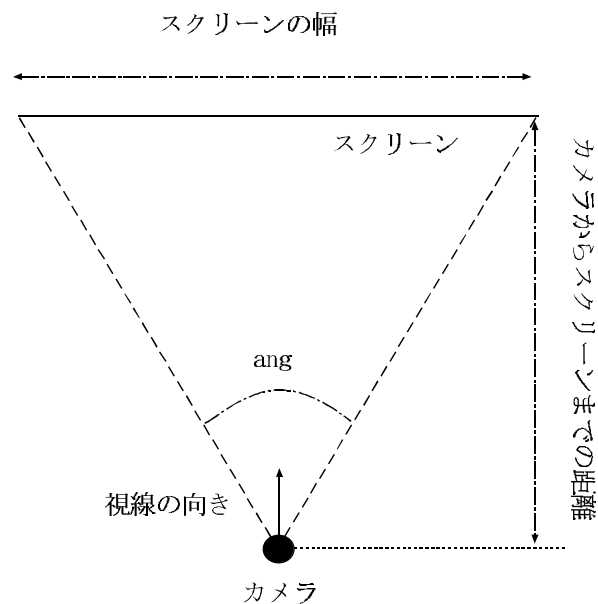
画角は下記の関数を用いる事によって自由に設定する事が出来ます。

NjSetCameraAngle(NJS_CAMERA *c, Angle ang)

画角（視野の水平方向の角度）を ang に設定します。

カメラの画角はスクリーンと密接な関係に有り、実際にはカメラの位置からスクリーンまでの距離を求め設定されます。そのため njSetCamera を実行後、njSetScreenDist や njSetPerspective 関数を用いてスクリーンまでの距離を変更するとカメラの設定と矛盾が起こりますので注意して下さい。

下記はカメラとスクリーンとの関係を上から見下ろして表したものです。



カメラの深度について:

ここで言うカメラの深度とは表示を行う際のニアクリッピング値とファークリッピング値を指します。

これらの値は下記の関数を用いて設定できます。

NjSetCameraDepth(NJS_CAMERA *c, Float n, Float f)

ニアクリッピング値、及びファークリッピング値を設定します。

Ninja の座標系では Z 軸手前が正の値となる為、両値には必ず負の値を設定するよう気を付けて下さい。どちらの値も - 1 より大きい値が設定された場合は - 1 に、 - 65535 より小さな値が設定された場合には - 65535 に値を切り詰め設定します。

これは njClipZ 関数で指定するものと全く同じ物です。そのため NjSetCamera 関数を実行後に njClipZ 関数を用いてニアクリップ値、ファークリップ値を変更するとカメラの設定と矛盾が起こりますので注意して下さい。

実際のプログラムでの使用方法について:

カメラを使用する為の手順は以下の通りです。

マトリクススタックを用意し初期化する。

カメラを用意し初期化する。

カメラの各パラメータを好きな値に設定する。

使用するカメラをセットする。

モデル等の描画を行う。

(以下 の繰り返しとなります。尚、VIEW との併用は一切の動作を保証しません。)

カメラは下記の関数を用いてセットします。

njSetCamera(NJS_CAMERA *c)

この関数はカメラ構造体の各パラメータからマトリクスを作成し、マトリクススタックのベースに反映させます。従って、この関数を実行する前に必ずマトリクススタックの初期化を済ませておくよう注意して下さい。

プログラム例)

原点に置いたオブジェクトの回りを視線をオブジェクトに固定したまま回転します。

```
NJS_MATRIX mat[10]
```

```
NJS_CAMERA camera;
```

```
njInitMatrix(mat, 10, 0);
```

```
njInitCamera(*camera);
```

```
njTranslateCameraPosition(&camera, 1000.f, 0.f, 0.f);
```

```
njPointCameraInterest(&camera, 0.f, 0.f, 0.f);
```

```
while(1){
```

```
    njSetCamera(&camera);
```

```
    /* ここでワールド座標の原点にオブジェクトを描画 */
```

```
    njRotateCameraPositionY(&camera, NJM_DEG_ANG(1));
```

```
    njPointCameraInterest(&camera, 0.f, 0.f, 0.f);
```

```
}
```