

DxLibの衝突判定

DxLibには、3Dゲームにおける、便利な衝突判定が用意されています。

MV1SetupCollInfo	コリジョン情報を構築する
MV1TerminateCollInfo	コリジョン情報の後始末をする
MV1RefreshCollInfo	コリジョン情報を更新する
MV1CollCheck_Line	線分とモデルの当たり判定
MV1CollCheck_Sphere	球とモデルの当たり判定
MV1CollCheck_Capsule	カプセルとモデルの当たり判定
MV1CollCheck_GetResultPoly	当たり判定結果ポリゴン配列から指定番のポリゴン情報を取得する
MV1CollResultPolyDimTerminate	当たり判定結果ポリゴン配列の後始末をする

MV1SetupCollInfo

モデルのポリゴン情報を使用して、
衝突判定用の情報(コリジョン)を構築する。



モデルのハンドルIDに対して、この関数を実行すれば、
衝突判定用の関数が使用できるようになる。
モデルの位置や回転、大きさが変更された場合は、
このコリジョン情報も更新しないと、正しい衝突判定ができない。

但し、MV1SetupCollInfoは処理が重たいため、
毎フレーム、コリジョン情報を変更したい場合は、
MV1RefreshCollInfoを使用する。

MVICollectionInfoやMVICollectionInfoで構築・更新していれば、下記の関数で、衝突判定を行うことができる。

MVICollection_Line

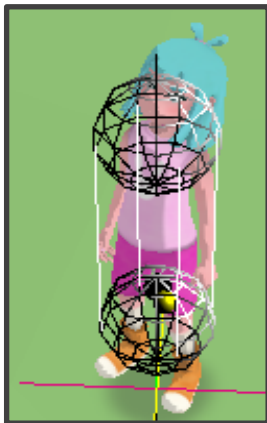
線分とモデルの当たり判定。地面などの衝突判定に使用。

MVICollection_Sphere

球とモデルの当たり判定。弾や球体に近いモデルの衝突判定に使用。

MVICollection_Capsule

カプセルとモデルの当たり判定。人型モデルの簡易的な衝突判定に使用。



```
MVICollection_Capsule(  
    int MHandle, int FrameIndex,  
    VECTOR Pos1, VECTOR Pos2, float r );
```

2つの球体を結んだような形状。

使い方(球体編)

```
auto info = MVICollection_Sphere(mStage->GetModelDungeonId(), -1,  
    mPlayerShip->GetTransform()->pos, PlayerShip::COLLISION_RADIUS);  
if (info.HitNum >= 1)  
{  
    // 衝突している!!!  
}
```