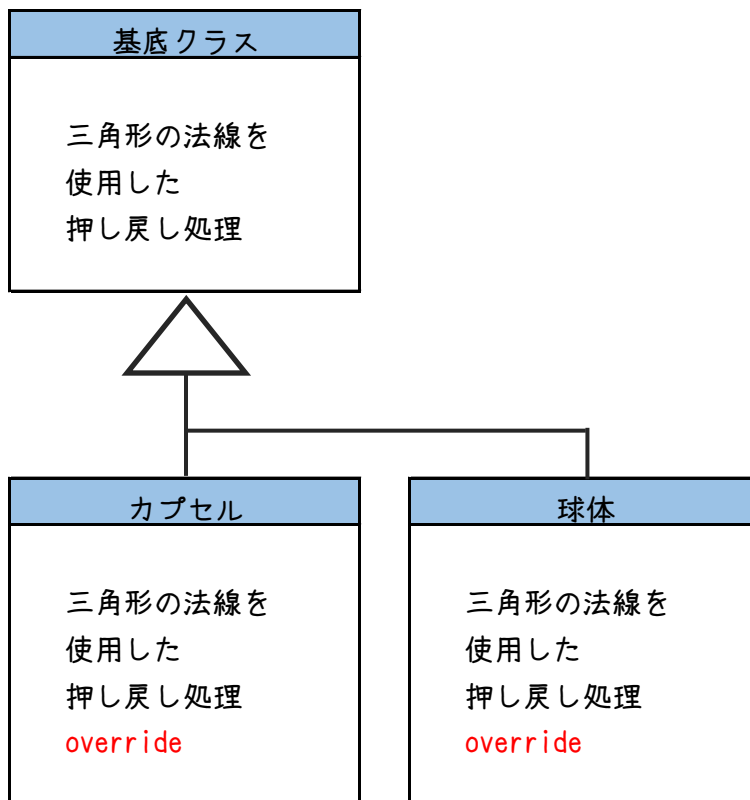


衝突処理のポリモフィズム化

カプセルと球体にGetPosPushBackAlongNormal関数を作成し、押し戻し処理を衝突(Collider)クラスに作成しました。

三角形との押し戻し処理は、どの形状のColliderでも必要な汎用的な機能になりますので、基底クラスで宣言するのが好ましくあります。

但し、派生クラスごとに内部的な処理は異なりますので、対象関数をoverrideして、独自の処理を記述する必要があります。



基底クラスに関数を汎化させることで、この関数を使用する時に、わざわざキャストする必要がなくなります。

このように、同じインターフェースを通して、実際には異なる振る舞いを実行できる仕組みのことを、ポリモフィズムといいます。

クラスを使う側が使い易くなったり、派生クラスの制作者がクラス設計を理解し易く結果的に実装し易くなります。

Camera.cpp

```
void Camera::Collision(void)
{
```

～ 省略 ～

```
// 球体コライダ情報
```

```
const ColliderSphere* colliderSphere =  
    dynamic_cast<const ColliderSphere*>(ownColliders_.at(typeSphere));
```

```
// 指定された回数と距離で三角形の法線方向に押し戻す
```

```
transform_.pos =  
    ownColliders_.at(typeSphere)->GetPosPushBackAlongNormal(  
        hitPoly, CNT_TRY_COLLISION_CAMERA, COLLISION_BACK_DIS);
```

```
}
```

キャスト無しで使用できる。

【要件】

ColliderBaseクラスに、純粋仮想関数GetPosPushBackAlongNormalを作成すること。

- ・ 各派生クラスでoverrideすること
- ・ ColliderModelとColliderLineは、定義のみで実装しなくてよい（ゼロVECTORか、異常値VECTORを返す）

【目標】

Cameraクラス側からポリモフィズムによる機能呼び出しができ、これまでどおりのカメラの押し戻し処理が実行されること。