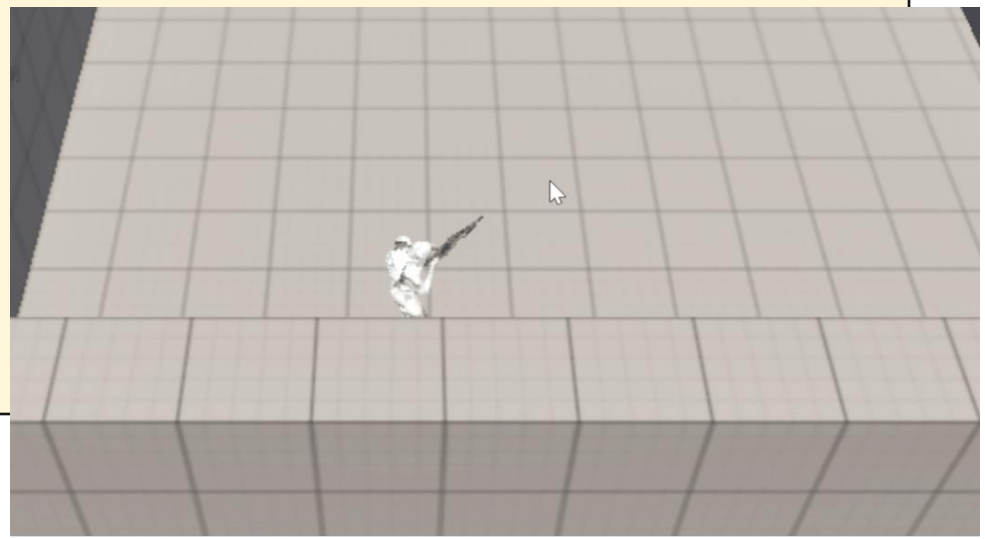
見下ろしシューティング（制作中）

個人製作
使用環境：UnrealEngine5.5
制作期間：3か月～



学習題材として

　C++でほとんどの処理を作成。

　TGSなどの制作では、参加人数と学習コストを考慮してBlueprint使用でしたが、後半は全体の把握や整理が難しくなり、C++の必要性を感じて学習しました。

- 基本的なプレイヤーキャラクターの移動（WASD）、マウス追従回転
- 拾える武器(Weapon)クラスの基礎設計と、弾（Projectile）の管理
- 敵キャラクターの視界判定と、プレイヤーへの単純な追跡AI
- 体力システムと装備管理システム（コンポーネントベース）

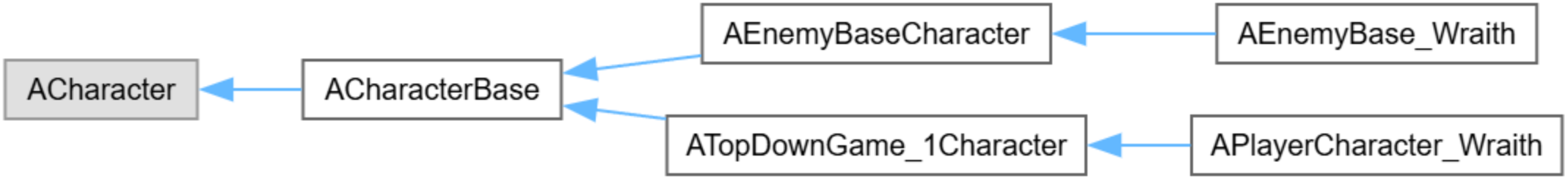
等の実装を行っており、今後はステルスシューティングのような方向性での完成を目指しています。

継承による拡張性

CharacterやWeaponはBaseとなるクラスを継承して管理。
子クラスには固有の処理として、

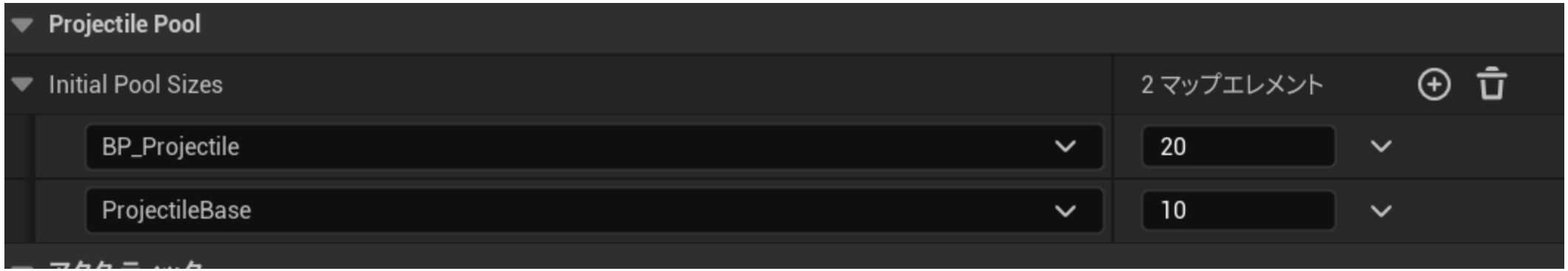
・死亡時の反応
・メッシュの持つ特定名のソケットに対するアタッチ
・AIのセットアップ

などを記載しています。



ObjectPool

　弾丸の管理にはObjectPoolを使用。
　TMapを用いて、ProjectileBaseから派生したアクタはそれぞれのプールに保管され、管理できるようにしました。



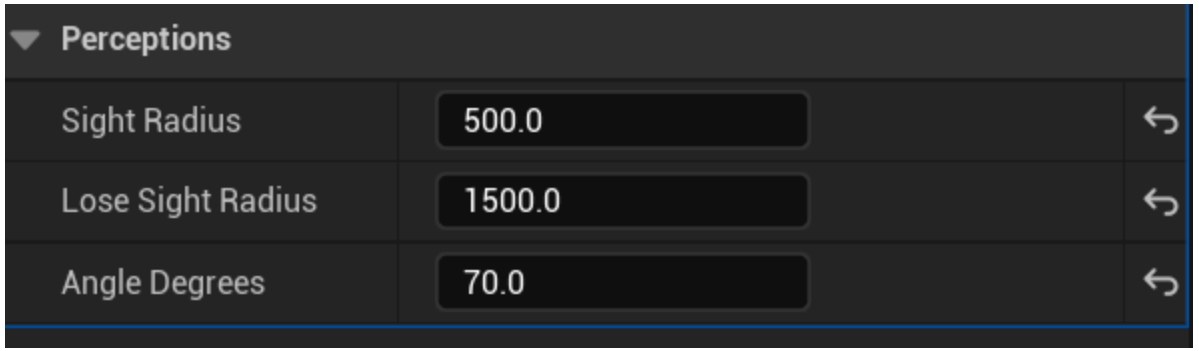
破棄する代わりに
→

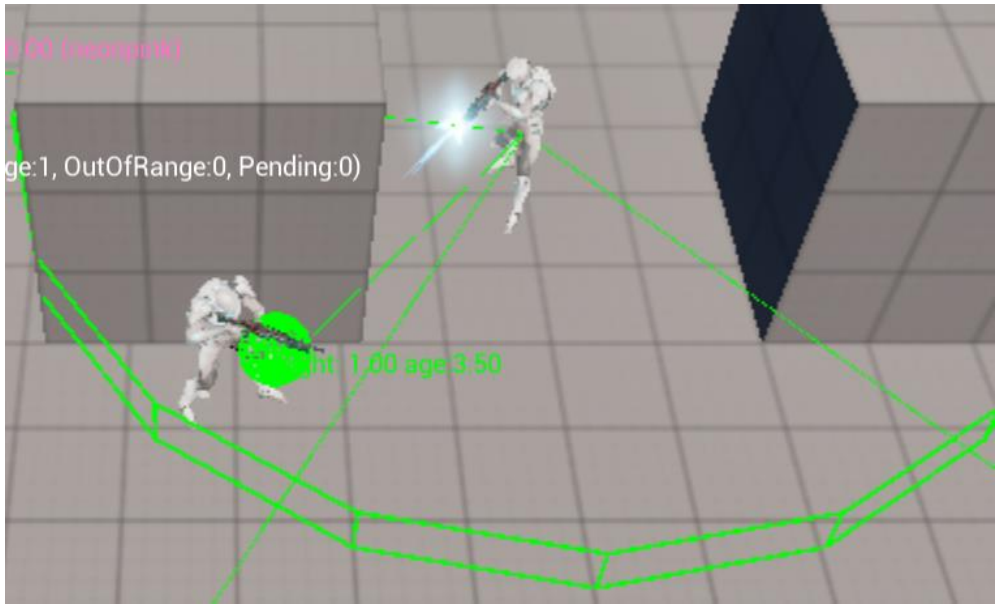
```
void AProjectileBase::ReturnToPool()
{
    if (PoolManager.IsValid())
    {
        DeactivateProjectile();
        PoolManager->ReturnProjectileToPool(this);
    }
    else
```

```
void AProjectileBase::DeactivateProjectile()
{
    ProjectileMovement->StopMovementImmediately();
    SetActorEnableCollision(false);
    SetActorHiddenInGame(true);
    SetActorTickEnabled(false);
    OnDeactivate();
}
```

視界判定

　AIPerception Sightを使用。AICのコンポーネントですが、実行時にスポーンするため、そのままだとエディタ側で調整できません。
　EnemyCharacter側で設定用の変数を用意してAICに渡しています。
　PerceptionのC++実装には特にリファレンスが少なく、動作までに苦労しましたが、AIPerceptionComponentのソースを追うことでAddDynamicの記載方法を理解し、C++での管理ができました。





Doxygen,Graphvizを用いてクラス一覧を作成しています。
構造の視覚化、整理に有効でした。
Docs内のindex.htmlを開くことで閲覧できます。