

プログラマーポートフォリオ

水野幸次

【扱える言語】 C++, Java

【使用したゲームエンジン】 Unreal Engine, Unity

目次

ポートフォリオ全体の目次です。

- 自己紹介(3, 4ページ)
- 作品紹介(5~18ページ)
- 終わりに(19ページ)

自己紹介

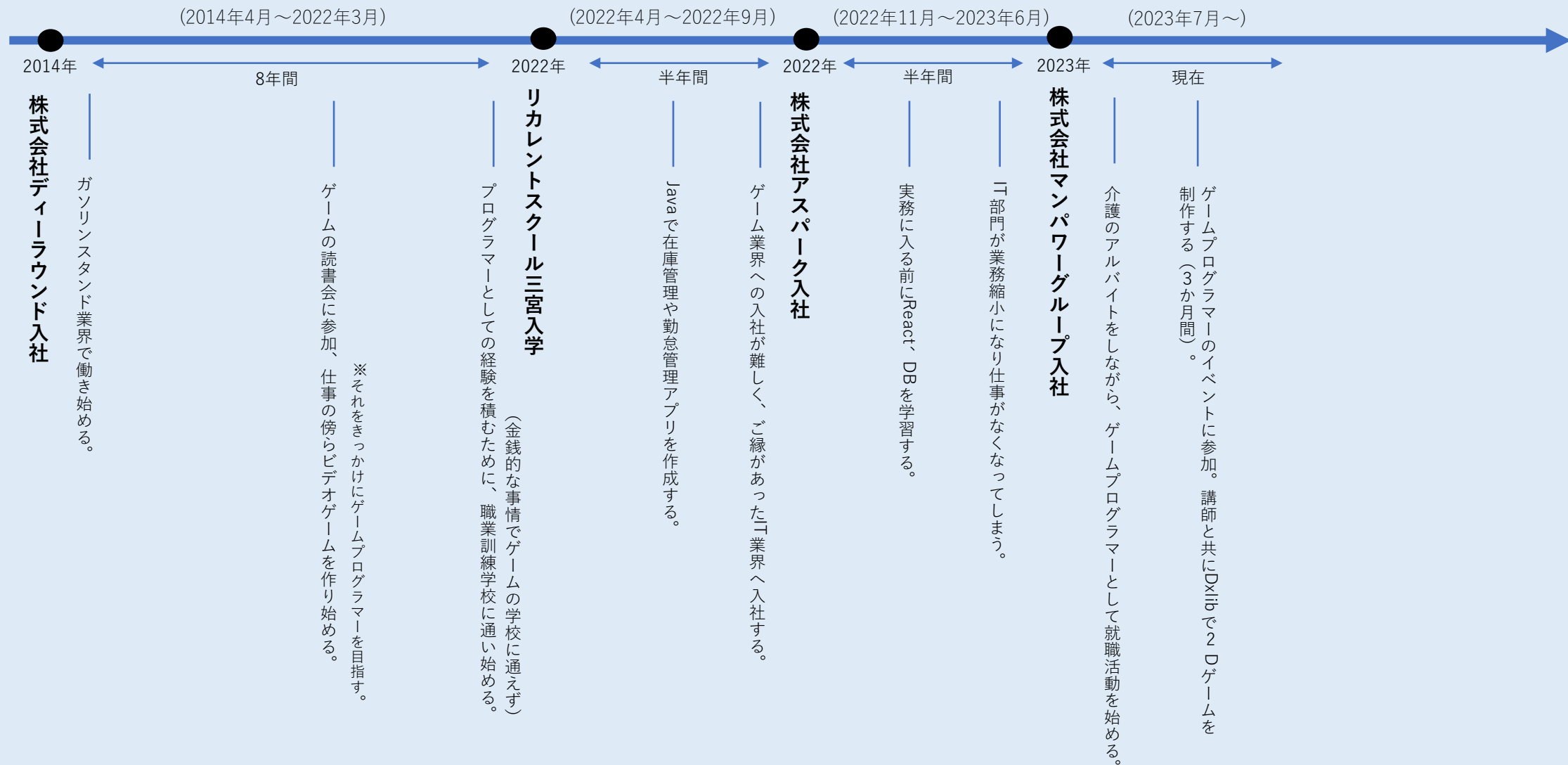
3年前にゲーム制作を始めました。

学習した内容をまとめました。

種類	期間	内容・作品
Unity/C#	6ヶ月	2D・3Dゲームを作成しました。そのうちの一つのボードゲームをGooglePlayストアにリリースしました。
Maya/Blender/SubstancePainter	1年	キャラ, 背景モデリングやアニメーションを作成しました。
UnrealEngine/C++	1年6ヶ月	BluePrintやC++を用いて3Dゲームを作成しました。
Dxlib/C++	1ヶ月6ヶ月	Dxlibで2Dゲームを作成しました。
Git	1年6ヶ月	コミット/プッシュ/プルをはじめとしたGitの概念を学習
DirectX12/HLSL	3ヶ月	DirectX12 の基礎部分 の学習(3D キャラモデルの表示程度)/シェーダーに関する基礎知識
Python	3ヶ月	FastAPIを使用したTODOアプリ制作
Java/HTML/CSS/React/Typescript/MySQL/PostgreSQL	6ヶ月	Webアプリ制作(出欠管理/在庫管理)

自己紹介

(経歴を年表にしてみました)



作品紹介

制作した作品を紹介します。

- ・特に注力したゲームの紹介(6~12ページ)
- ・その他ゲームの紹介(13~16ページ)
- ・その他制作物の紹介(17, 18ページ)

2 D死にゲーの紹介(DXlib)

『2D死にゲー』の紹介



タイトル画面



インゲーム画面



ゲームオーバー画面

【環境】

- Dxlib/Visual Studio 2022/C++
- 音声はお借りした物を使用しています。
- 画像は株式会社ヒストリアから提供された物を使用しています。

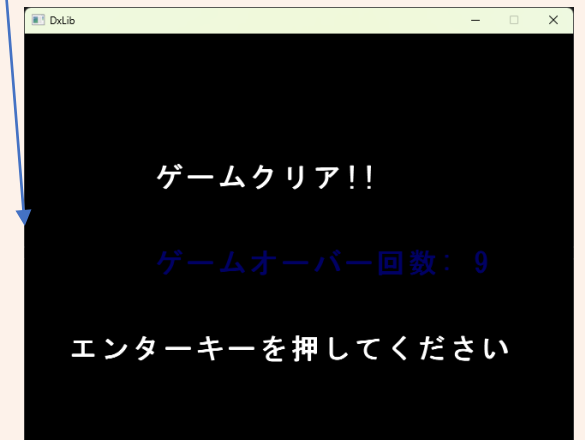
プレイ動画(<https://youtu.be/6lQ1RIq6lIE>)

【概要】

株式会社ヒストリア主催のゲームイベントにて作成。

障害物に当たると一撃死してしまう死にゲーです。

ゲームエンジンを使わずC++のみでゲームを作るということを主目的として作成しました。



クリア画面

目標とチャレンジした箇所

このゲームを作るにあたって、以下の目標を設定しました。

目標①『クラス図を作ってからゲーム制作をする』

目標②『ゲームエンジンを使わずゲームを作る』

目標③『数学的な考え方を使って実装してみる』

目標③に当たる箇所について

左画像のコードでは、sin関数を使用して”動く足場”を実装しています。sin関数を使うことで固定した座標を使用しないでよくなり、動きも滑らかになりました。

右画像のコードでは、atan2関数を使用してプレイヤーを追尾する弾を実装しています。

If文を使って実装することも可能ですが、今回は勉強の為三角関数を使用して実装しました。

```
/*
 *→ サイン関数を使って足場を左右に動かす
 */
void Platform_Passable::Move(float delta_seconds)
{
    Vector2D input_dir;

    // radian_periodを加算していく
    radian_x += radian_period;
    // sin関数でx座標を計算
    input_dir.x = speed * sin(radian_x * 0.01);

    delta_position = input_dir * MOVEMENT_SPEED * delta_seconds;
    // x座標を変更
    SetPosition(GetPosition() + delta_position);
}
```

Platform_Passable.cpp

```
/*
 *→ こうげきの第二段階
 *→ 追尾弾
 */
void Enemy_Boss::Attack2(float delta_seconds)
{
    // リンゴとプレイヤーを結ぶ直線を平行移動させて、原点からみた直線座標へ移動(atan2関数を使用するため)
    double dx = sampleObject->GetPosition().x - apple_pos_2.x;
    double dy = sampleObject->GetPosition().y - apple_pos_2.y;
    // atan2関数を使用して角度を計算する
    double angle = atan2(dy, dx);
    // sin/cos関数を使用して、角度を-1~1の数値に変換して、ポジションに加算していく
    apple_pos_2.x += static_cast<int>(3 * cos(angle));
    apple_pos_2.y += static_cast<int>(3 * sin(angle));
    gimmick_apple_2->SetPosition(apple_pos_2);
    gimmick_apple_2->Update(delta_seconds);
}
```

Enemy_Boss.cpp

2D死にゲーのクラス図

オンライン3Dシューティングの紹介 (UnrealEngine)

『オンライン3Dシューティング』の紹介



タイトル画面(ゲームモードを選択し、ホストかクライアントを選択)



ロビー画面(マッチングするとゲーム開始)



インゲーム画面

【環境】

- ・ Unreal Engine/Visual Studio 2019/C++
- ・ 音声/画像ファイルは無料のアセットを使用しています。

プレイ動画

(https://youtu.be/Pa_ZF1XYEil)

※オンラインゲームのプレイ動画になっておりますが、一つのメインPC内で仮想環境を立ち上げてテストプレイしております。
その都合上、メインPCと仮想環境の切り替え時に動画が止まったり見づらい場面がございます。ご了承ください。

【概要】

オンラインTPSゲームです。

Udemyという動画講座サイトを利用して作成しました ([Unreal Engine 5 C++ Multiplayer Shooter / Udemy](#))。

UnrealC++に慣れること、オンラインゲームの作り方をすることを主目的としています。



リザルト画面

目標とチャレンジした箇所

このゲームを作るにあたり、以下の目標を設定しました。

目標①『UnrealC++の勘所をつかむ』

目標②『オンラインゲームの作り方を学ぶ』

参考講座自体に不備があり、バグ修正や仕様の追加が必要でした。
追加で実装した箇所を紹介します。

右画像では、プレイヤーがリスポーンする処理を書いています。
講座ではPlayerStartをランダムで選びリスポーンする、という内容でした。
ただその場合だと、リスポーンした直後に敵と出会う場合があります。

修正したコードでは、全プレイヤーの座標と全PlayerStartの座標を比較し、リスポーンするプレイヤーと距離の最も遠いPlayerStartを選んでリスポーンするようにしました。
バロラントのデスマッチの仕様を参考に実装しました。

目標②に関しては、以下の要素をを学びました。

『ListenServerとSteamOnlineSubsystem を使用したネットワーキング通信, Replicate やRPCの仕様, ラグを解決する Server の巻き戻し実装, デリゲートやインターフェイスに関する知識, 簡易なチート対策』です。

```
/*
 * リスポーン処理
 */
void ABlasterGameMode::RequestRespawn(ACharacter* ElimmedCharacter, AController* ElimmedController)
{
    if (ElimmedCharacter)
    {
        ElimmedCharacter->Reset();
        ElimmedCharacter->Destroy();
    }
    if (ElimmedController)
    {
        //UE_LOG(LogTemp, Warning, TEXT("ElimmedController valid"))
        // マップに配置されている全てのプレイヤースタートを取得
        TArray<AActor*> AllPlayerStarts;
        UGameplayStatics::GetAllActorsOfClass(this, APlayerStart::StaticClass(), AllPlayerStarts);

        // 全てのプレイヤーを取得
        TArray<AActor*> Characters;
        UGameplayStatics::GetAllActorsOfClass(this, ABlasterCharacter::StaticClass(), Characters);

        TArray<float> StockedDistances;

        for (int i = 0; i < AllPlayerStarts.Num(); i++)
        {
            // キャラクター0(リスポーンする人)とプレイヤースタート[i]との距離を取得
            float MinDistance = (AllPlayerStarts[i]->GetActorLocation() - Characters[0]->GetActorLocation()).Size();
            for (int j = 1; j < Characters.Num(); j++)
            {
                // 'キャラクター0(リスポーンする人)とプレイヤースタート[i]'と'の距離と全てのキャラクターとプレイヤースタート[i]との距離'を比較する
                float Distance = (AllPlayerStarts[i]->GetActorLocation() - Characters[j]->GetActorLocation()).Size();
                if (Distance < MinDistance)
                {
                    // 現時点での一番小さな距離が取得される
                    MinDistance = Distance;
                }
            }
            // プレイヤースタートの個数分追加するようにする
            // これは次の処理で、何番目のプレイヤースタートが一番離れているかを指定するために必要なため(Selection部分)
            StockedDistances.Add(MinDistance);
        }

        float MaxDistance = StockedDistances[0];
        int32 Selection = 0; // 一番離れているプレイヤースタートを指定するための変数

        for (int i = 1; i < StockedDistances.Num(); i++)
        {
            // StockedDistancesの中で一番値が大きい物を選別
            if (MaxDistance < StockedDistances[i])
            {
                MaxDistance = StockedDistances[i];
                Selection = i;
            }
        }
        RestartPlayerAtPlayerStart(ElimmedController, AllPlayerStarts[Selection]);
    }
}

/* ... */
```

その他ゲームの紹介

『ボードゲーム』

【概要】

ランダムで配られる数字を、指定されたマス目に埋めていくゲームです。

制作意図としては、Unityでの制作に慣れるためと、実際にリリースしてみるという目的がありました。

GooglePlayストアにリリース。

ローカライズや広告実装に対応。

【環境】

- ・ Unity/VSCode/C#
- ・ 音声/画像ファイルはお借りした物を使用しています。

プレイ動画
(<https://youtu.be/ySwRn9bG7yc>)



タイトル画面



インゲーム画面



ゲームオーバー画面

『FPS3Dシューティング』



チュートリアル画面



【環境】

- ・ Unreal Engine/Visual Studio 2019/C++
- ・ 音声/画像ファイルは無料のアセットを使用しています。

プレイ動画(<https://youtu.be/orLLCN2xZbo>)

【概要】

銃の反動で後方に飛んで操作するFPSゲームです。

Udemyという動画講座サイトを利用して作成しました。

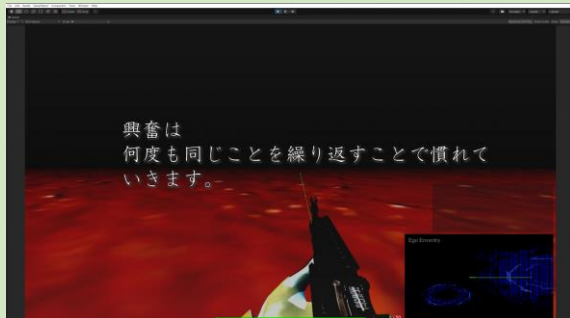
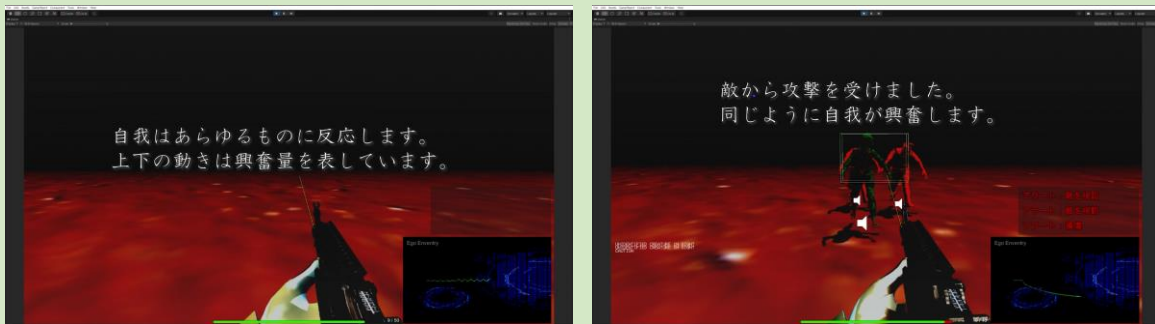
作成意図としては、UnrealC++に慣れることを主目的としています。

FPSゲームの基礎的な実装から敵AIの作成、アウトラインエフェクトや地形に合わせた足音の実装まで、C++を用いて学習しました。



タイムアタックステージ画面

『3Dゾンビシューティング』



インゲーム画面

【環境】

- Unity/VSCode/C#
- 音声/画像ファイルはお借りした物を使用しています。

【概要】

Unityの勉強の為に作成したゲームです。
自我システムを搭載したFPSゲームです。
自我はあらゆるものに反応して揺れ動きます。
その揺れ方でゲームメカニクスを実装したり
キャラクターを表現する予定でした。

『BluePrintだけで作ったFPS』



インゲーム画面

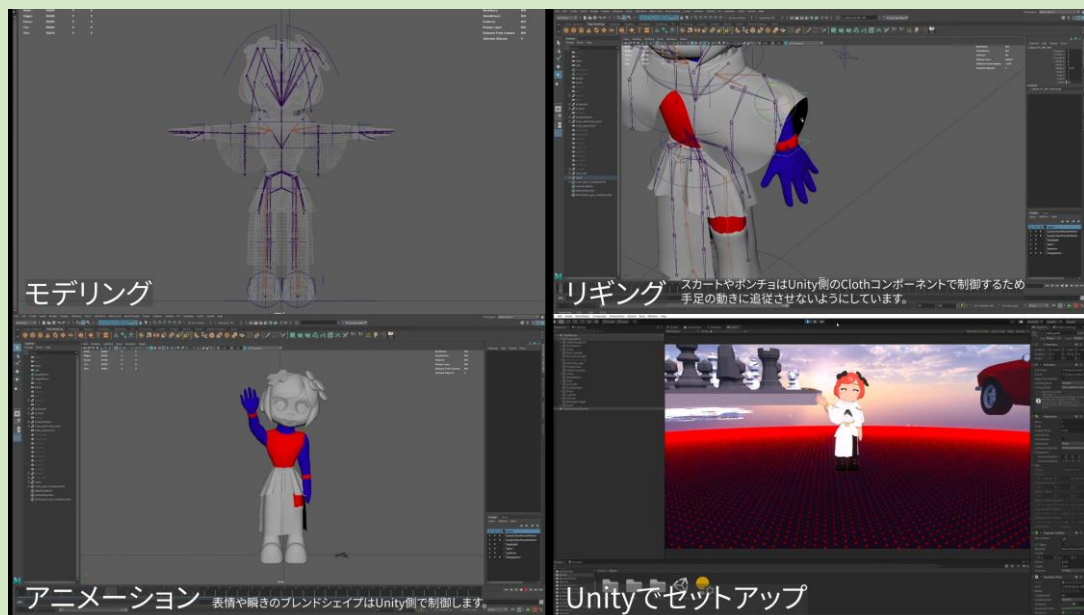
【環境】

- UnrealEngine
- 音声/画像ファイルはお借りした物を使用しています。

【概要】

BluePrintの勉強の為に作成したゲームです。
UnityからUEに移るにあたり、どのような感覚で
ゲームを作ることができるのか確認する意図
がありました。

『3DCG』



【環境】

・ Unity/Unreal Engine/Maya/Blender/Substance Painter

アニメーション動画
(https://youtu.be/DLD4F_U4f5s)

【概要】

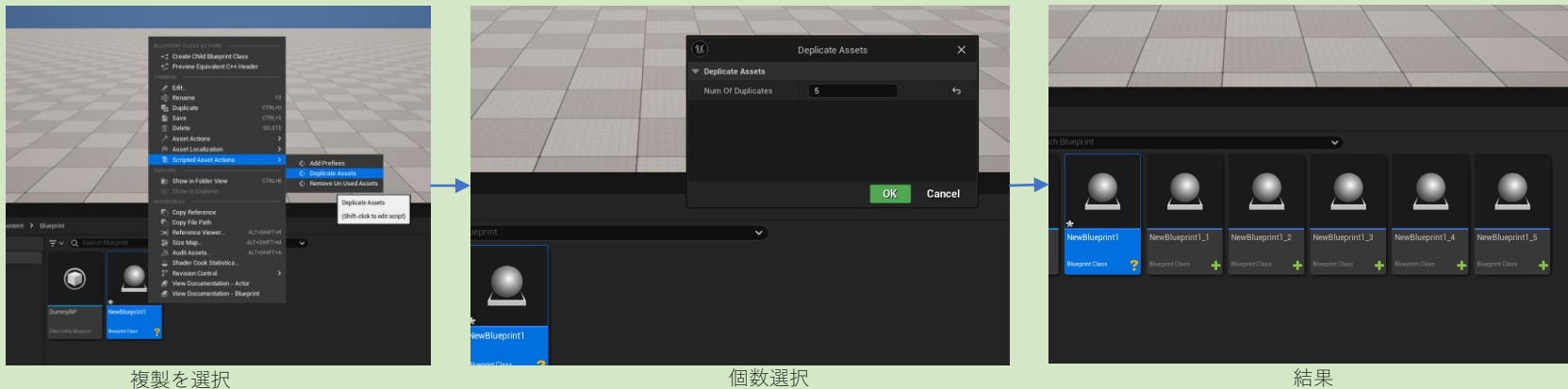
プログラム以外のことも把握しておいた方が良さかもしれないと思い作成した物です。

キャラクターとアニメーションは Maya/Unityを使用、背景(キッチン)は Blender/Substance Painter/UEで作成・セットアップしました。



『現在学習中のもの』

選択したアセットの複製



選択したアセットにプレフィックス追加



【環境】

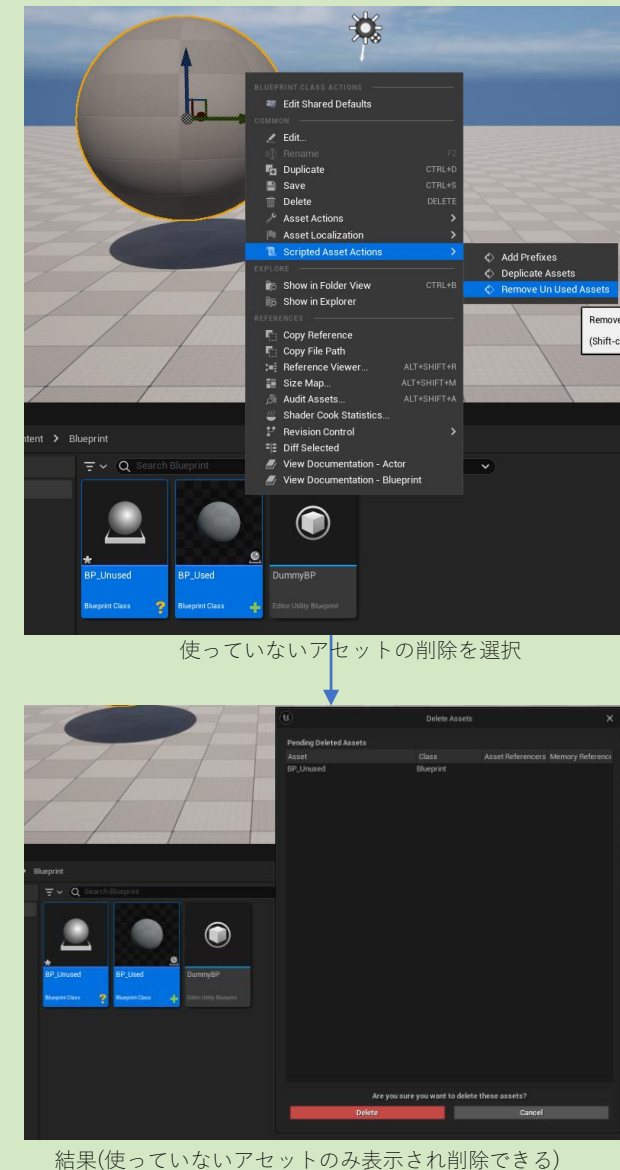
・Unreal Engine/Visual Studio
2022/C++

【概要】

現在は拡張エディタとGAS(GameAbilitySystem)の学習をしています。

画像は拡張エディタのものです。アセットの複製やプリフィックスの追加、使っていないアセットの削除などを行っています。

使っていないアセットの削除



終わりに

ここまでご覧下さり、ありがとうございました。

大好きなビデオゲームを作っている業界で、仕事をしたいという思いから、ポートフォリオを作成してきました。

もしご縁がございましたら、是非ともよろしく願いいたします。

水野幸次