

# PORTFOLIO

2025 年卒業予定

日本電子専門学校ゲーム制作研究科

小田島 稜人

# プロフィール

---

氏 名: 小田島 稜人

生年月日: 2003 年 6 月 3 日

所 属: 日本電子専門学校 ゲーム制作研究科 3 年

希望職種: ゲームプログラマ

---

主にクラス指向である UnrealEngine を使用して開発を行っており、綺麗なクラス設計を心掛けて開発をしています。基本的に、使用したことのない機能を使用したりしてよりよいアプローチ方法を模索することを心掛けています。

## 趣味

---

- ・ゲーム (RPG、ドラクエモンスターズジョーカー2)
- ・音楽鑑賞 (アニソン、洋楽など)
- ・漫画 (ジョジョ、北斗の拳)
- ・映画鑑賞 (ゴジラ -1.0、ヴェノム)
- ・研究開発

## 使用言語・ツール

---



UnrealEngine

★★★★☆ 使用歴: 1年4カ月



C++

★★★★★ 使用歴: 2年4カ月



Unity

★★★★☆ 使用歴: 1年



C#

★★★★☆ 使用歴: 1年



VisualStudio

★★★★☆ 使用歴: 2年



JetBrains Rider

★★★★☆ 使用歴: 4カ月



GitHub

★★★★☆ 使用歴: 1年



SourceTree

★★★★☆ 使用歴: 1年

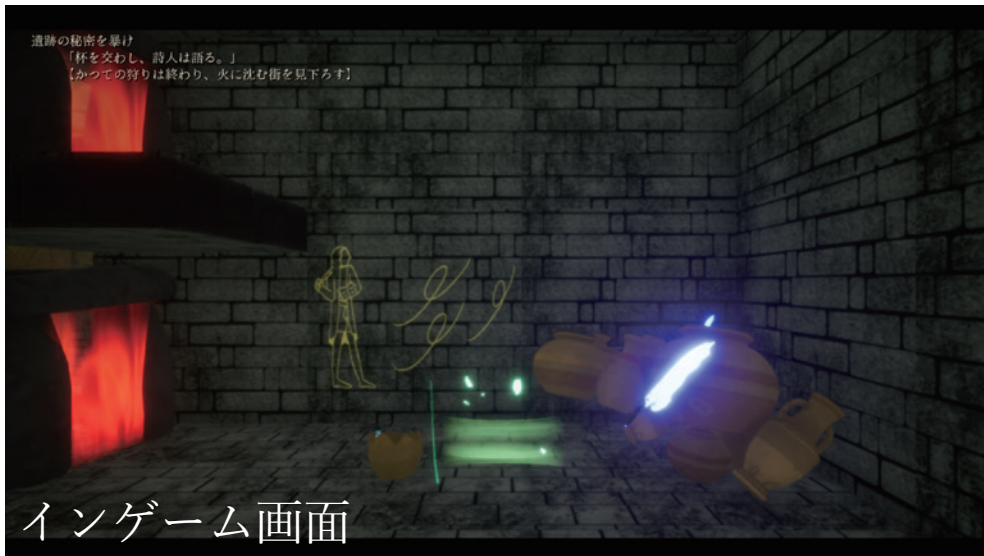
# Wizards' Ruins

制作日数 : 3 か月

制作時期 : 2023 年 11 月 ~ 2024 年 2 月



タイトル画面



インゲーム画面

## メンバー内訳

プログラマー : 4 人    プランナー : 1 人

## 担当箇所

- ・ プレイヤー制御    ・ カメラ制御
- ・ サウンド管理

## 使用言語・ツール

 UnrealEngine4.27

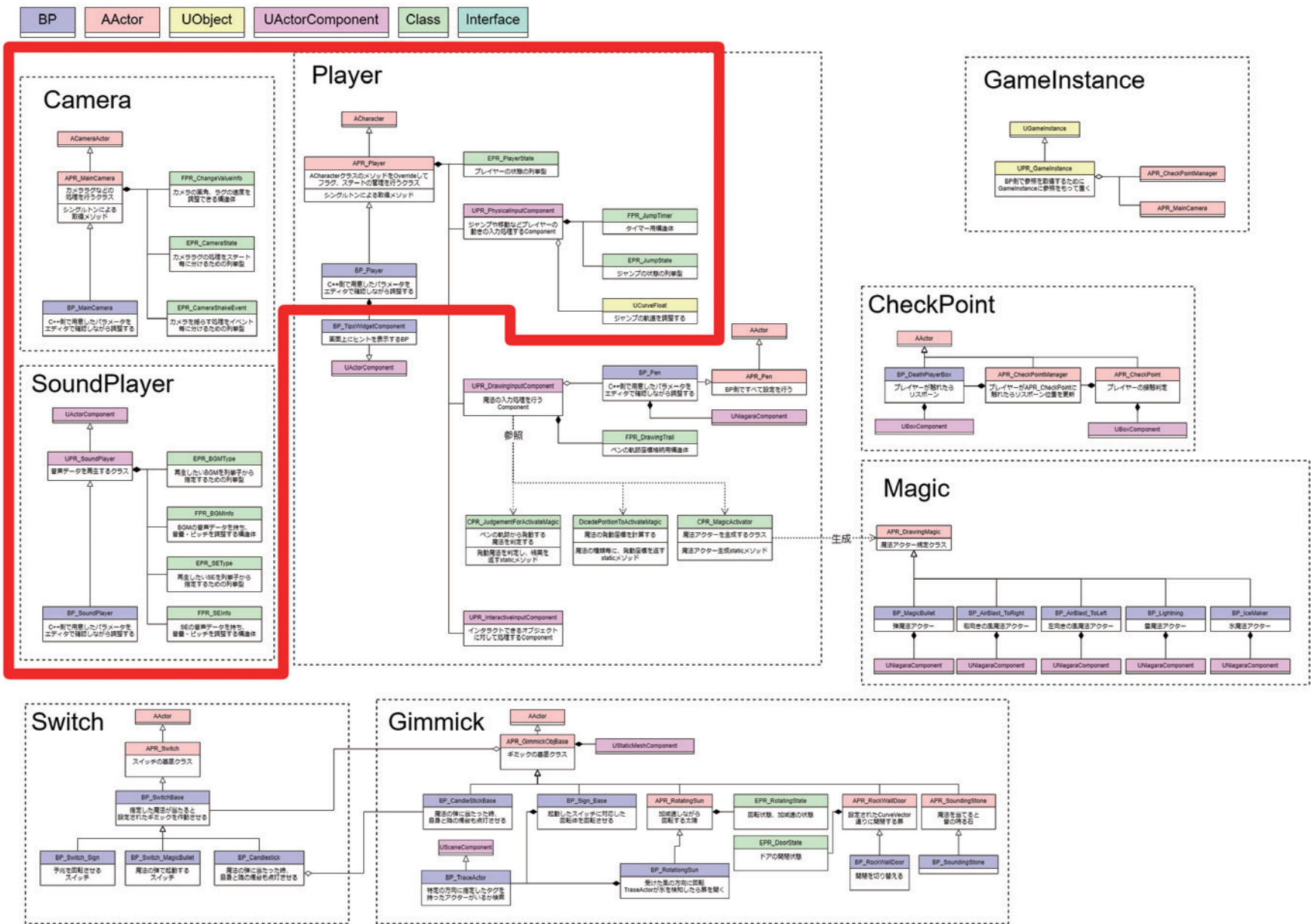
 C++

 VisualStudio2019

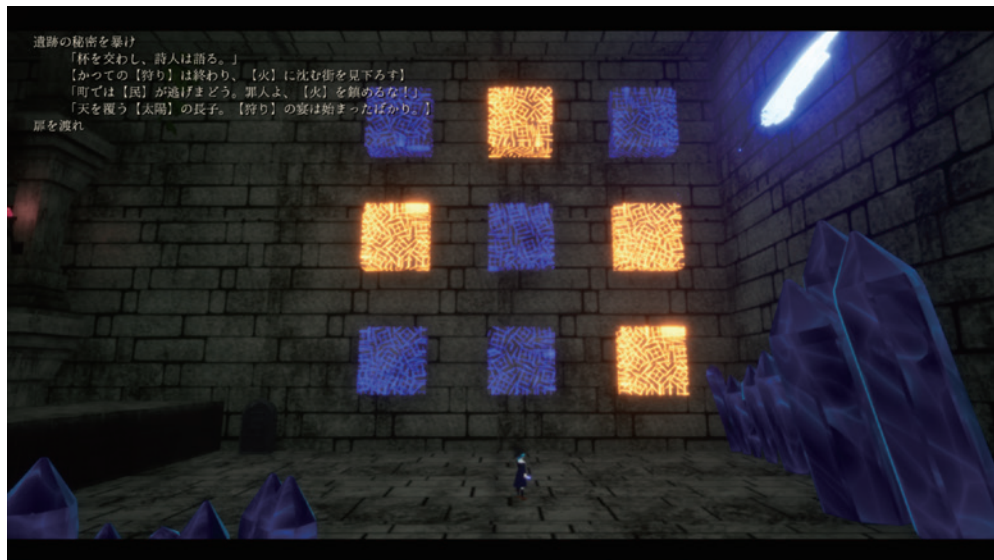
 GitHub

 SourceTree

# 全体クラス図 ※赤枠で囲われている部分が担当箇所です







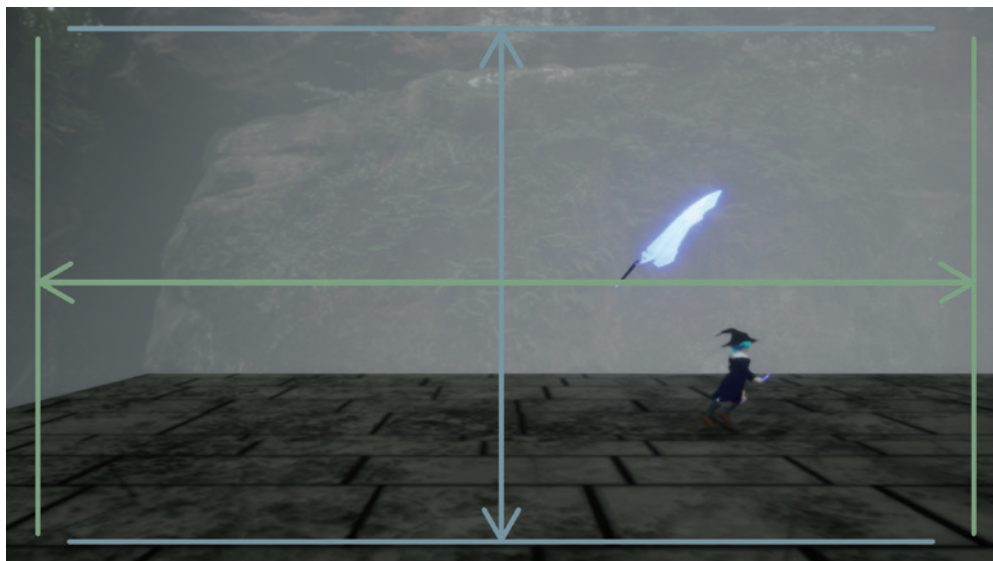
## 概要

ペンで魔法を描いてステージにあるギミック・謎を攻略し、ゴールを目指すゲームです。

## 制作目的

UE を使った開発経験が無かったので、UE に慣れるために制作しました。また、学校での展示会に出展するための作品でもあり、「ゲームとして遊ぶことが出来る」作品を意識して制作しました。

## 工夫した点

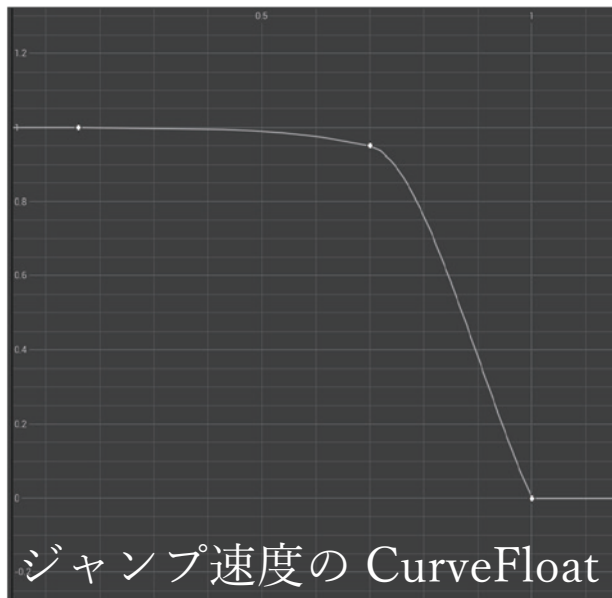


## カメララグ処理

「プレイヤーが移動したら最初は遅く、段々プレイヤーに追いつく」というカメラワークにしたかったので、独自でカメララグ処理を追加して実現させました。

既存の UE のカメララグ処理を使う場合、一定の距離での判定になるので、2D ゲームの場合、プレイヤーが画面外に出てしまうという問題がありました。その問題解決のためにも、独自で処理を記述しました。

# 工夫した点



## ジャンプ処理

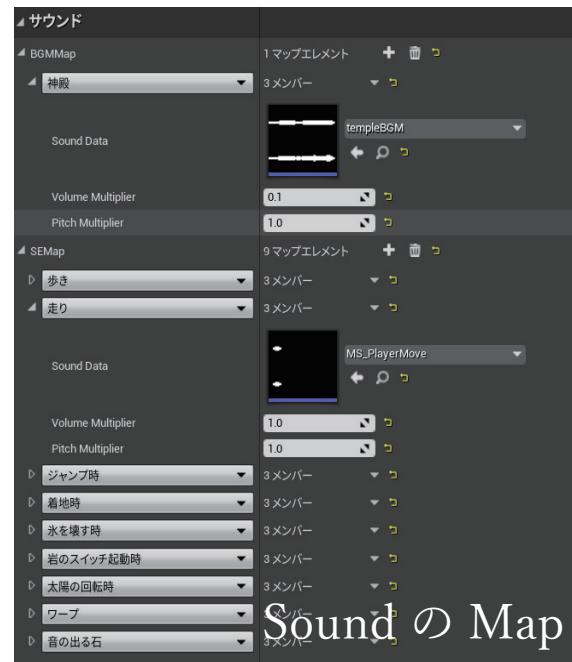
プレイヤーのジャンプをふんわりしたジャンプにしたい、何度もパラメータを調整すると思ったので、プログラマ以外でも調整が出来るように視覚的に調整出来る UE の機能である CurveFloat を使用して実装しました。

また、UE の場合重力等のパラメータを調整して理想のジャンプを実装しようとしたのですが、他に与える影響が大きく苦戦しました。なので、直接速度を CurveFloat で調整するようにして実装しました。

## サウンドデータ管理

サウンドのパラメータをまとめた構造体を定義して Map で管理しました。こちらもプログラマ以外でも調整がしやすくなることを意識して実装しました。

今回はチーム制作のため、データを 1 か所にまとめて管理することで確認作業を減らし、作業効率を図りました。



# Blood Oxygen

制作日数：1 か月

制作時期：2024 年 1 月～ 3 月



タイトル画面



インゲーム画面

## メンバー内訳

プログラマー：1 人

## 担当箇所

個人制作のため全てのプログラム

## 使用言語・ツール

 UnrealEngine4.27

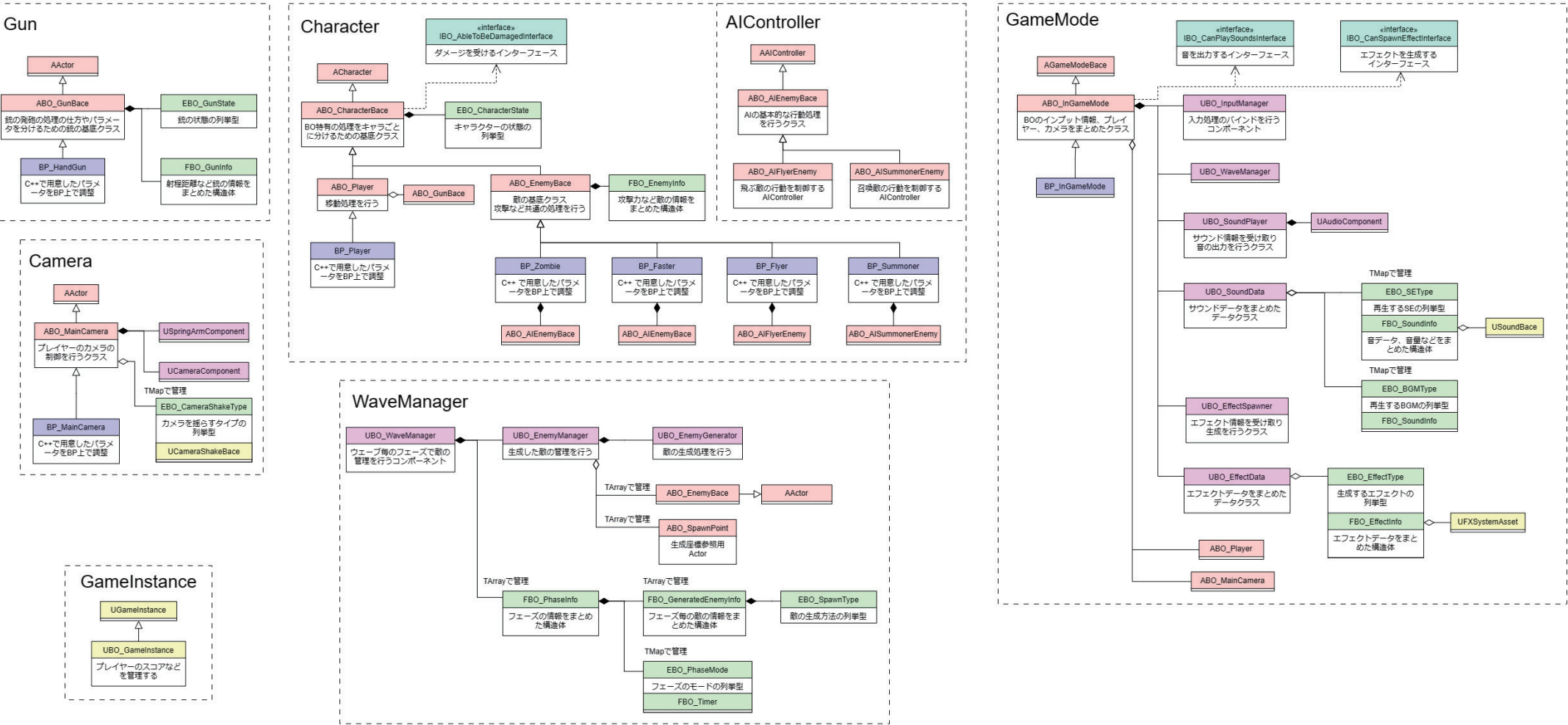
 VisualStudio2019

 SourceTree

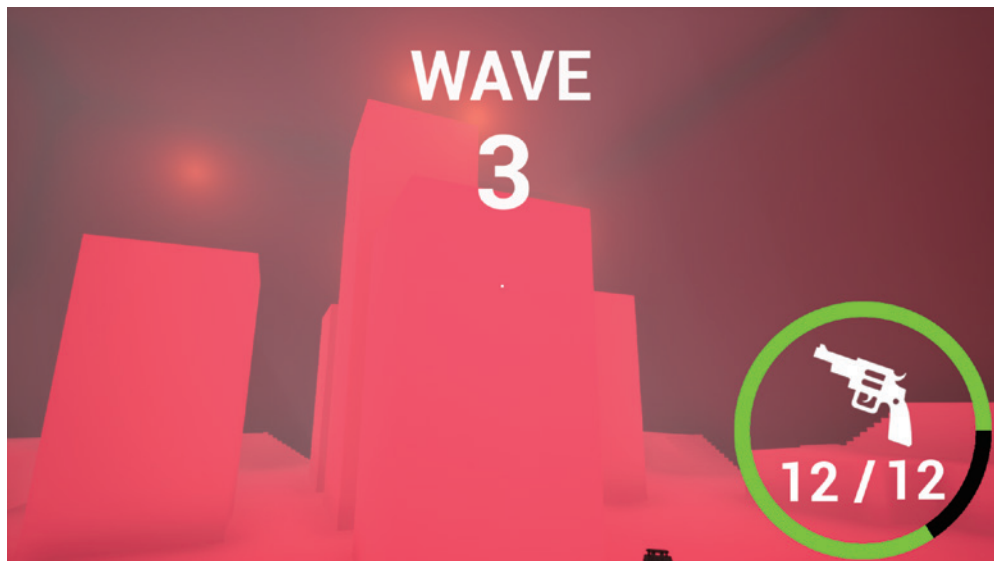
 C++

 GitHub

# 全体クラス図







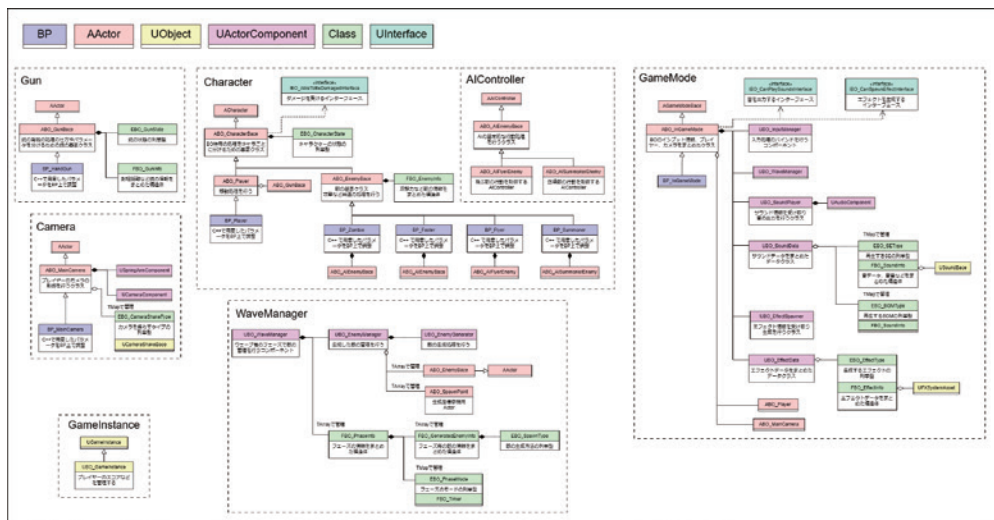
## 概要

ウェーブ毎に襲い掛かってくるゾンビたちを銃で倒しながら、最後のウェーブ（6ウェーブ）まで生き残るゲームです。

## 制作目的

開発したことないジャンルのゲームの開発とクラス設計の練習のために制作しました。また、UEを使って一連のゲーム制作の練習も兼ねています。

## 工夫した点

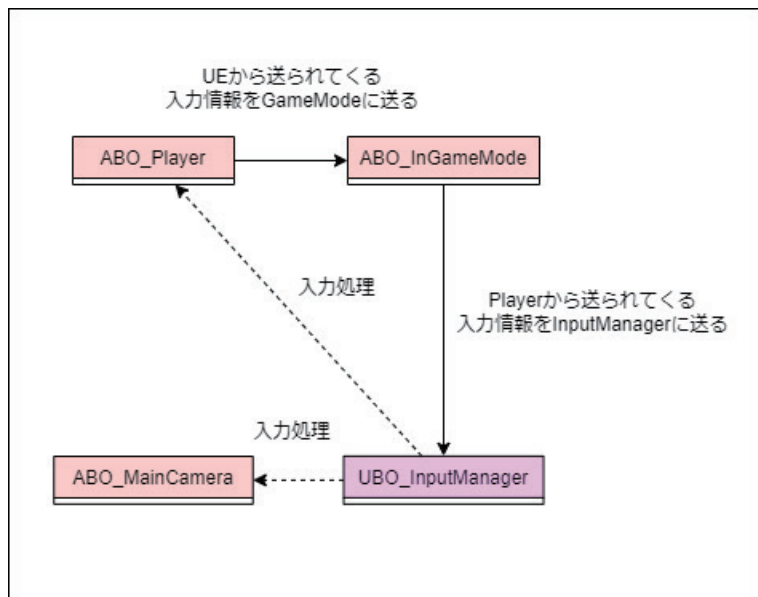


## クラス設計

UnrealEngine の設計志向がクラス指向とコンポーネント指向のハイブリッドなので、それらの特徴を活かしたクラス設計を心掛けました。

具体的に意識した事は、多態性とカプセル化です。これらを意識することで、作業効率の向上・バグが発見しやすくなるなどのメリットが多く、実際に短い期間での制作が出来ました。

# 工夫した点



## 入力処理

UE は GameMode で DefaultPawn に指定したクラスに入力情報が渡されるという形になっており、DefaultPawn に指定したクラス内に入力処理を記述しなければならないので依存度が高く、他クラスへの入力処理を行うことが困難でした。

そのため、入力処理をまとめたコンポーネントを定義し、GameMode にアタッチすることで依存度を下げ、他クラスへの入力処理を行いやすいようにしました。

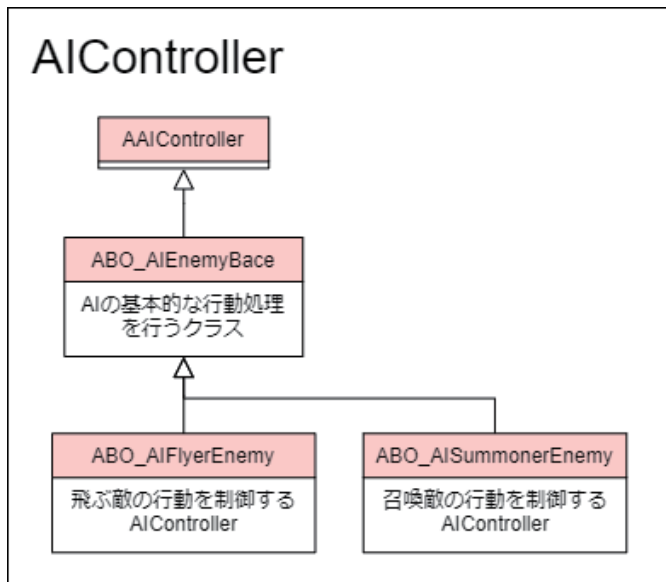
## カメラ処理

プレイヤークラス内にカメラの処理があることがクラス指向的に違和感を感じたため、プレイヤーとカメラのクラスを分けて制作しました。

プレイヤーとカメラを分けたことで、カメラクラスに処理を記述しやすくなりました。実際に、プレイヤー死亡時の演出にカメラの動きを取り入れました。



# 工夫した点

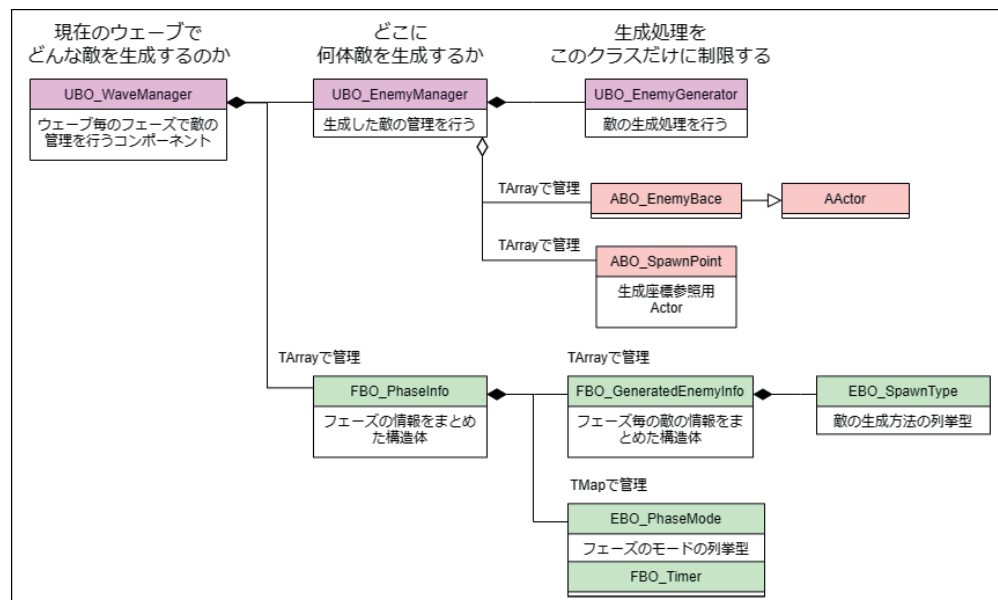


## 敵 AI 処理

今回の作品は個人での制作なので、短い期間で複数の敵を作るために多態性を意識した設計を心掛けました。

具体的には、親クラスに行動ルーチンを記述して、子クラスに移動・攻撃それぞれの処理を上書きして振る舞いが変わるように設計しました。

また、C++ での開発のしやすさのために AIController を採用しました。



## ウェーブ処理

汎用性を高めるためによりクラスを機能単位に分けることを意識しました。それぞれ、敵を生成するコンポーネント、生成した敵を管理するコンポーネント、ウェーブ処理を行うコンポーネントの3つに分け、それぞれのクラスに沿った処理を記述しました。実際に、それぞれのクラスに機能を分けたことで、細かい処理が記述しやすくなりました。

# ハウスパークッション

制作日数 : 2 日

制作時期 : 2023 年 7 月



## メンバー内訳

プログラマー : 1 人

## 担当箇所

個人制作のため全てのプログラム



## 使用言語・ツール



Unity2021.3.19f1



C#



VisualStudio2019



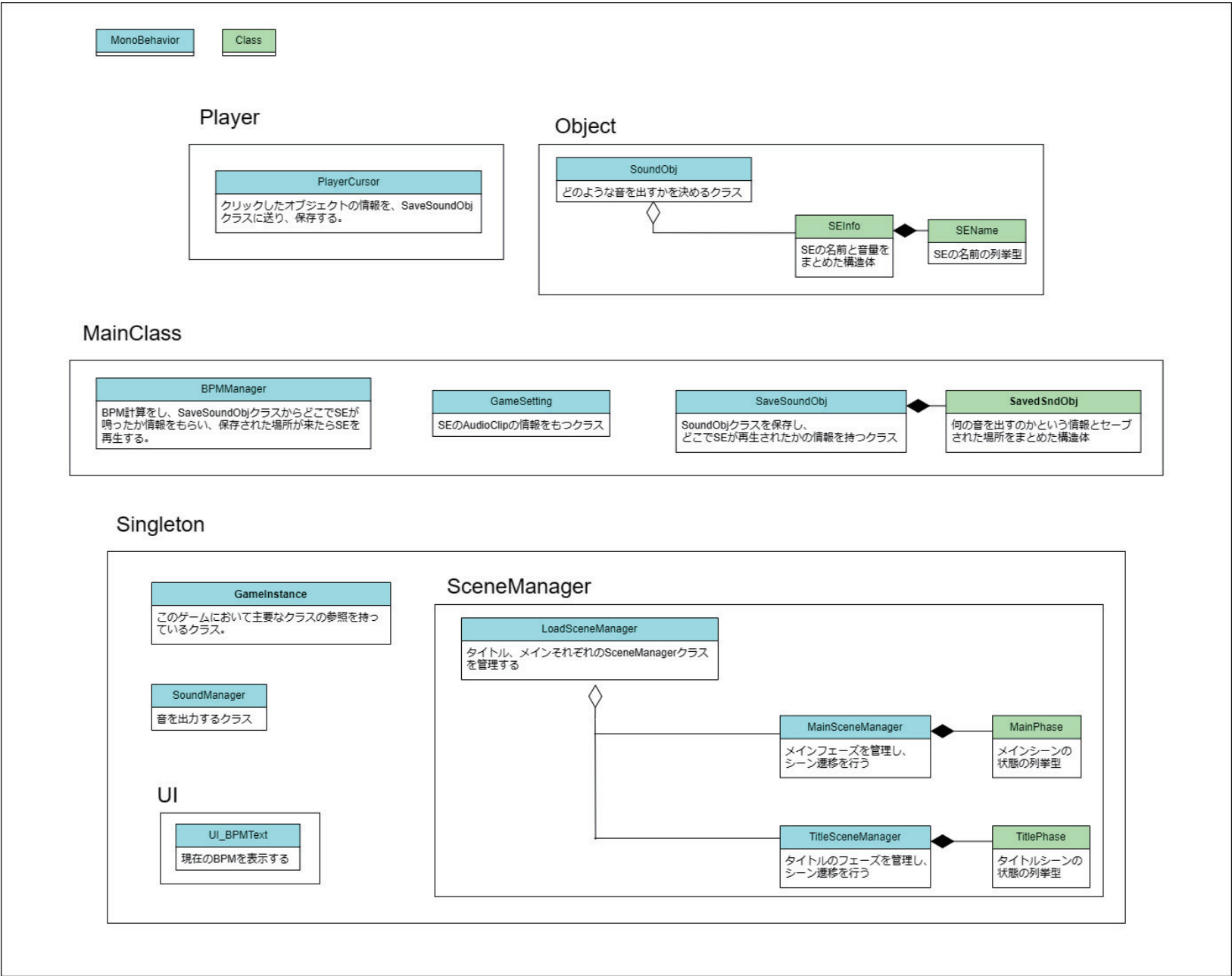
GitHub



SourceTree



# 全体クラス図





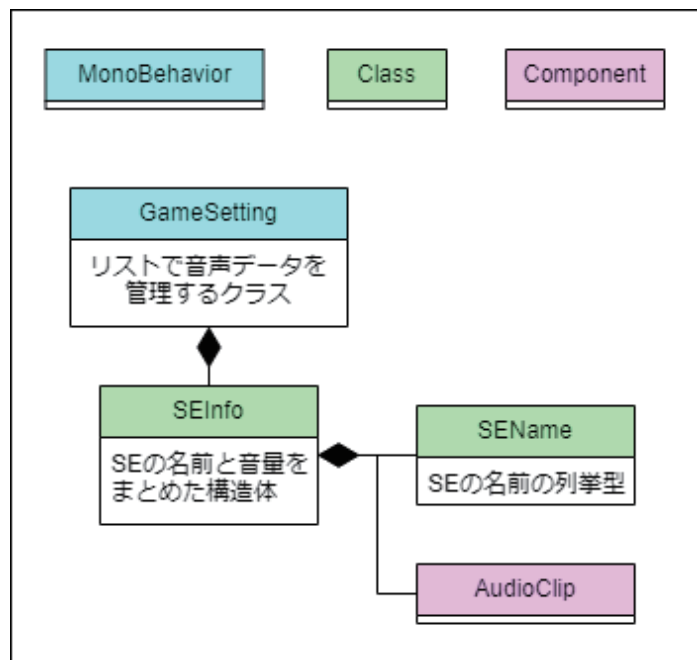
## 概要

家具をクリックして音を鳴らすゲームです。鳴らした音は繰り返し再生されるのでちょっとした作曲体験ができます。

## 制作目的

この作品は、UAA ゲームジャムに参加した際に制作しました。そのため、珍しいものを作ろうと考えました。テーマが「音」だったので、このようなゲームを企画し、制作しました。

## 工夫した点

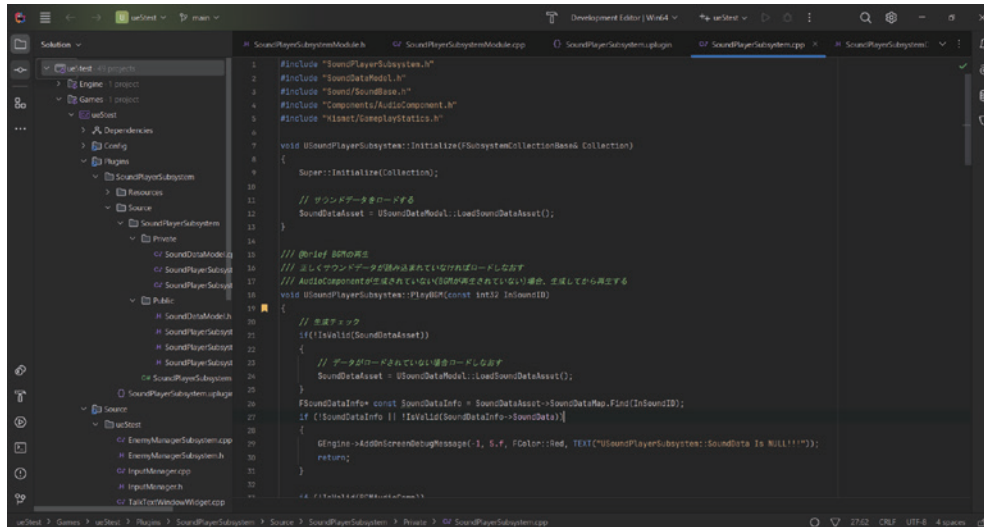


## サウンド再生処理

サウンド情報をまとめた構造体を定義し、その構造体をリストで管理するクラスを作成しました。今回は、再生したいサウンドの列挙型の値を渡すことでサウンドの再生処理を行うように設計したので、関数を呼び出す側が楽に呼び出せるようになりました。また、この作品は繰り返し SE を鳴らすので、再生した音と再生されたタイミングを紐づけて保存することで、繰り返し再生処理がしやすいようにしました。

# SoundPlayerSubsystem

制作日数 : 7 日  
制作時期 : 2024 年 7 月

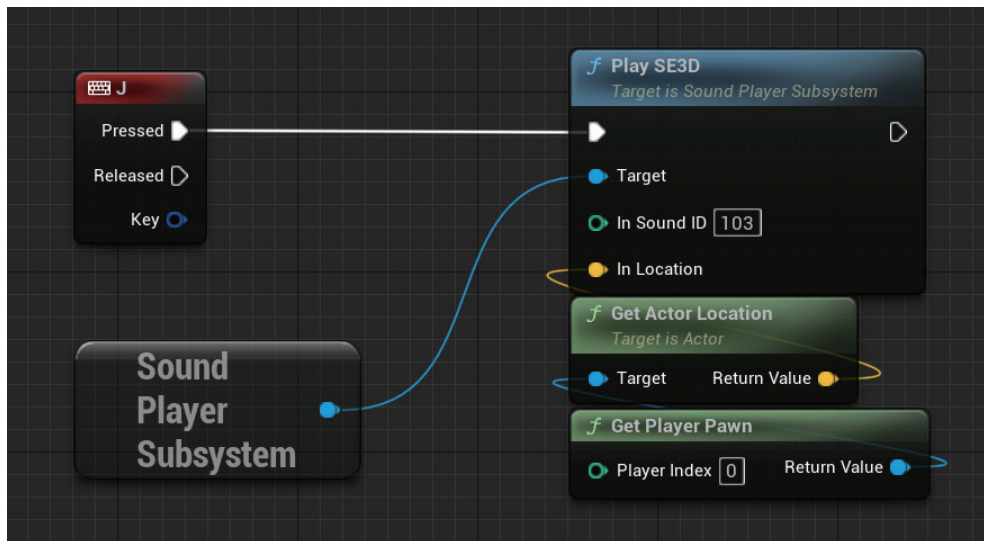


## メンバー内訳

プログラマー : 1 人

## 担当箇所

個人制作のため全てのプログラム



## 使用言語・ツール



UnrealEngine5.3.2



C++



JetBrains Rider 2024.1.3



GitHub



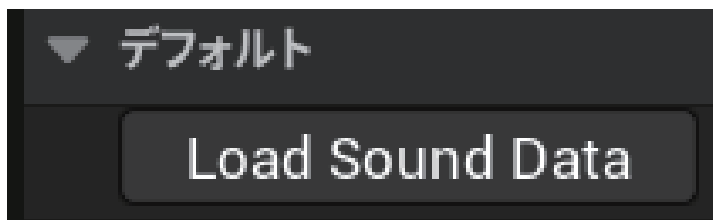
SourceTree

## 概要

サウンドのデータ管理と再生処理の両方を兼ねたシステムのプラグインです。

## 工夫した点

	A	B	C	D
1	---	SoundDataID	Volume	Pitch
2	MS_Buuun	101	1	1
3	MS_Doom	102	1	1
4	MS_Street	103	1	1
5	MS_BGM1	201	1	1
6	SC_Street	301	1	1



## 制作目的

Plugin 開発に興味があり、また Subsystem を使ったシステム開発にも興味があったので、開発経験のあるサウンド周りの知識を活かして制作しました。

プラグイン開発ということで、誰でも直感的に操作出来ることを心掛けました。

データの情報をまとめた構造体を定義し、そのデータを DataAsset を使用して ID で管理する設計にしました。ID で管理することで途中でサウンドを追加したくなくても楽に挿入出来るようにしました。

パラメータを CSV で管理することで、慣れ親しんだ Excel で調整出来るようにしました。そのため、プログラマ以外の人でも楽にパラメータを調整出来るようになりました。

行の名前をファイル名にすることで自動でサウンドデータをロードすることが出来るようにすることで、直感的に操作が出来るようにしました。