

# ポートフォリオ

HAL大阪 3年 飯塚健司





# 自己PR

## PR

私の強みは、仕様書+ $\alpha$ で面白さを作れることです。  
その仕様がゲームにとってどのような意味があり、どのような体験をさせるためのものかを考え、実際に作っているプログラマーだからこそわかる要素を取り入れて作ることができます。  
また、調整をしやすく作り面白さを作りやすくすることもできます。  
ただ作るだけではなく、面白さの追求をしています。

## 特技

あるハンティングアクションではタイムアタックでPS4最速記録を出したことがあります。  
その経験からプログラマーだからこそできる工夫を見つけることができ  
仕様書+ $\alpha$ で面白さを作るという強みにもつながっています。



# 目標

- クリエイターとして

プレイヤーが**本気**になって遊べるゲームを作りたい

ゲーム内で引き起こされる感情の変化だけでなく、そのゲームに対してプレイヤーが自発的に目標を立て、目標を達成しようと思えるゲームを作りたいです。このようなゲームを作ればより長く遊んでもらえるゲームになると考えています。

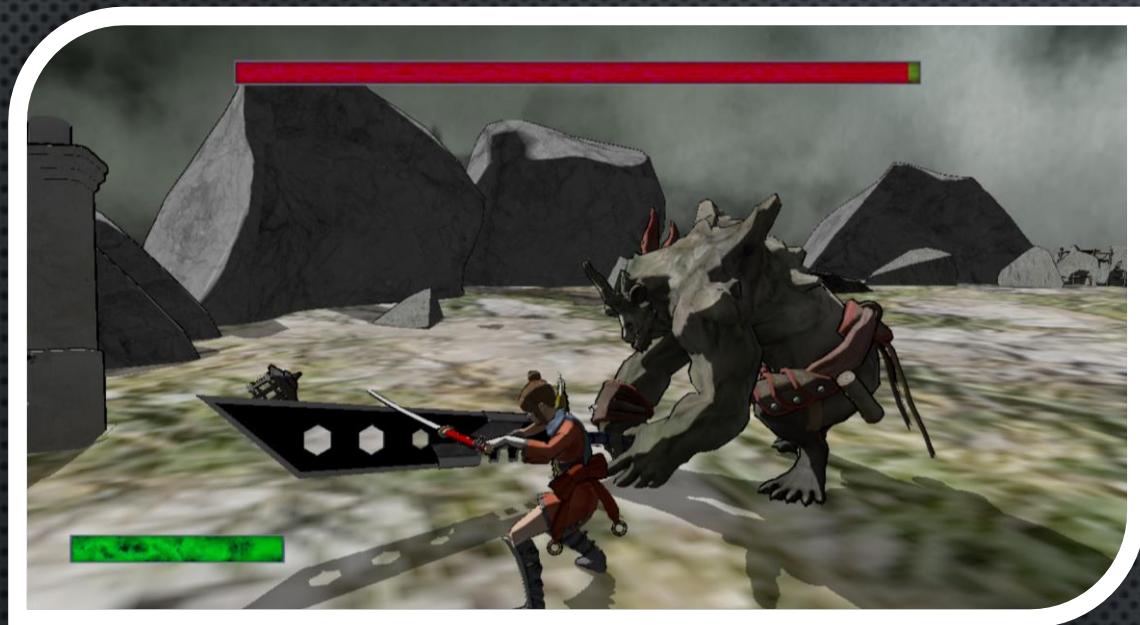
- プログラマーとして

**面白さを作る**ために**AIの技術**を使えるようになりたい

AIはゲーム体験の面白さを上げる要因だと思っています。  
ですが調整がしずらかったり、強くなりすぎたりしてしまったりと問題を抱えています。  
それらの問題をゲームに合わせて解決して活用できるようになりたいと考えています。



# Enemy Maker



## 作品概要

個人制作

ハンティングアクションゲームです。  
遺伝学習を使用して常にちょっと強い敵を  
制作することに挑戦しました。

## 開発環境

VisualStudio2022

DirectX11

C++20

## 使用ライブラリ

SimpleMath

Json

ImGui

Assimp

## 挑戦したこと

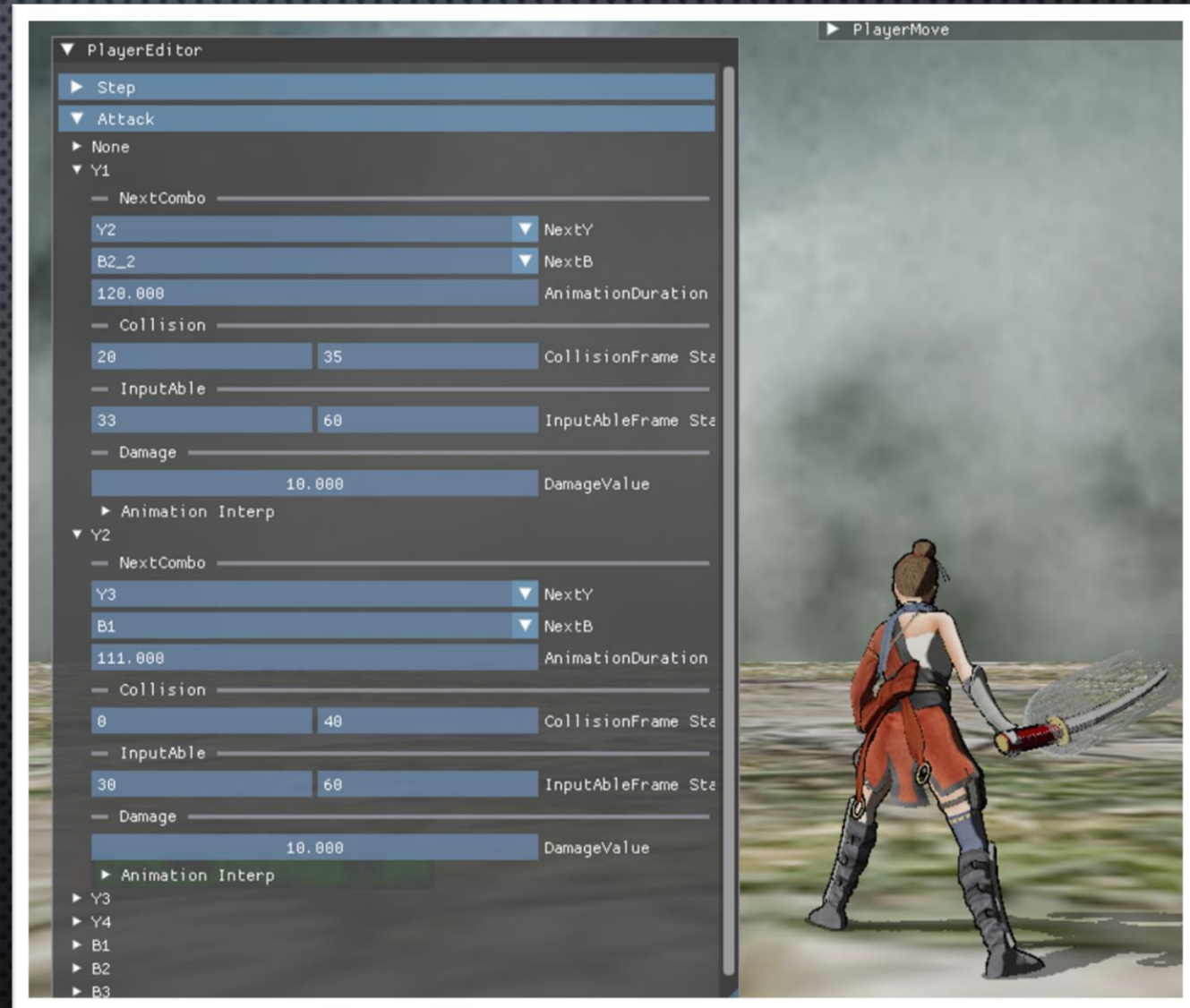
遺伝学習を用いた成長する敵AI

# こだわったポイント①

## 細部まで調整しやすいように

- 次のアニメーションへの補間
- 当たり判定発生フレーム
- 入力可能フレームの設定
- 次のコンボの設定

などができるように簡易的なエディターを作成。  
プレイしながら調整ができるようにしました。





# こだわったポイント②

## アニメーションの切り替わりのカクツキを軽減

アニメーションの切り替わりの違和感を減らすためにアニメーション間の補間を行いました。

アニメーションで動くメッシュにゲームオブジェクトの座標を同期させるための処理も考慮し位置、回転を調整しています。

```
std::shared_ptr<SkeletalMeshAnimationClip> toClip = mInterpToClip.lock();
std::shared_ptr<SkeletalMeshAnimationClip> baseClip = mInterpBaseClip.lock();

// 補間先のキーの情報を計算
DXSimpleMath::Vector3 toPosition;
DXSimpleMath::Vector3 toScale;
DXSimpleMath::Quaternion toQuaternion;
toClip->CalcAnimationKey(_toChannel, mToInterpClipTime, toPosition, toScale, toQuaternion);

// 補間元のキーの情報を計算
DXSimpleMath::Vector3 basePosition;
DXSimpleMath::Vector3 baseScale;
DXSimpleMath::Quaternion baseQuaternion;
baseClip->CalcAnimationKey(_baseChannel, mBaseInterpStartTime, basePosition, baseScale, baseQuaternion);

// 補間もとアニメーションのみが座標同期しているなら差分を引いて補間先アニメーションの位置に合わせる
if (baseClip->IsSynchroPosition() && !toClip->IsSynchroPosition())
{
    DirectX::SimpleMath::Vector3 difference = basePosition - toPosition;
    basePosition -= difference;
}

// 回転同期をしているアニメーションの場合はY回転を合わせる
if (baseClip->IsSynchroRotation())
{
    baseQuaternion.y = toQuaternion.y;
}

// 補完してアニメーションマトリックスを作成する
DXSimpleMath::Vector3 lerpPosition = DXSimpleMath::Vector3::Lerp(basePosition, toPosition, _t);
DXSimpleMath::Vector3 lerpScale = DXSimpleMath::Vector3::Lerp(baseScale, toScale, _t);
DXSimpleMath::Quaternion lerpQuaternion = DXSimpleMath::Quaternion::Lerp(baseQuaternion, toQuaternion, _t);
MathLibrary::MakeWorldMatrix(lerpPosition, lerpScale, lerpQuaternion, _animationMatrix);
```

## 挑戦したこと

遺伝学習アルゴリズムを用いて戦っていくと  
成長していく敵AIの制作に挑戦しました。

遺伝学習アルゴリズムの実装、検証、結果を  
まとめた資料を参照していただけると幸いです。



# Lab-rastings



## 作品概要

「刺す」がコンセプトの2Dスクロールアクションゲームです。  
改造された猫ちゃんが尻尾で敵をぶっ刺す瞬間が気持ち良いです。

## 受賞

2年次HALEVENTWEEK 独創力賞

## チーム構成

9名(プログラマー3、プランナー4、デザイナー2)  
**リーダー兼メインプログラマー**を担当しました。

## 担当箇所

- 企画
- プログラム全般
  - ベースプログラム
  - プレイヤー
  - ボス2体、雑魚敵1体
  - ステージ

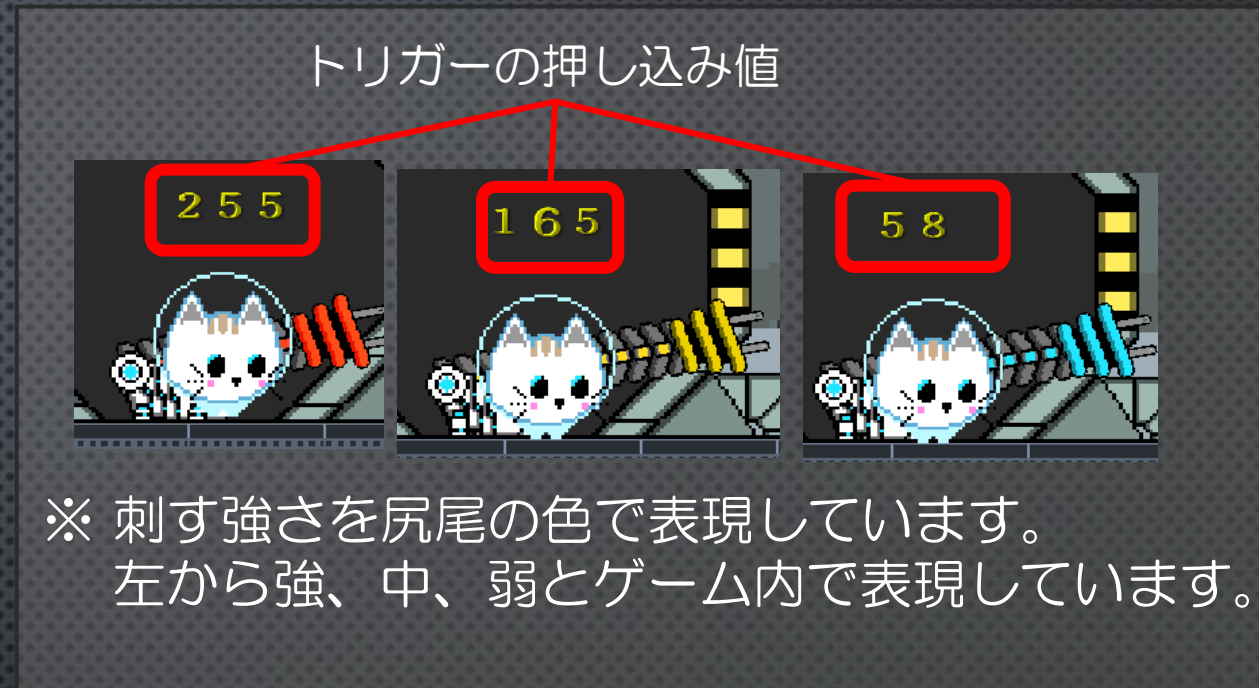
## 開発環境

VisualStudio2017  
DirectX11  
C++14



# こだわったポイント①

刺す遊びに強弱を持たせるために  
刺す強さをトリガーの押された量にリンク  
するようにしました。



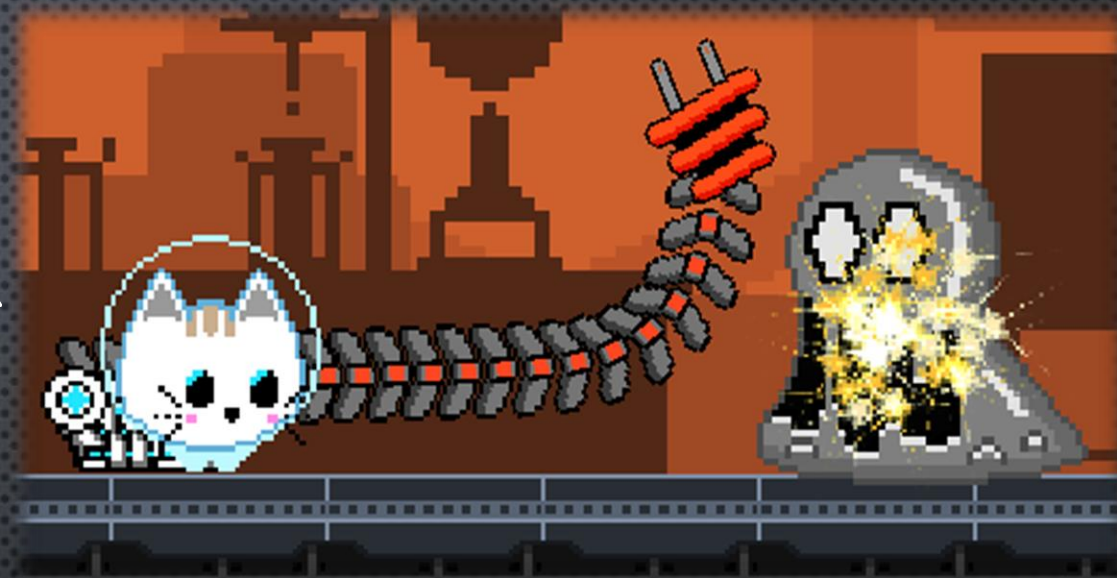
## なぜこのようにしたのか

「そっと刺す」「刺す」「ぶっ刺す」という弱、中、強の三段階に分けることで  
刺す遊びに幅が生まれ、弱と中があることで、強の「ぶっ刺す」が気持ちよさがより際立ち刺す  
遊びの楽しさが増大すると考えたから。  
また、操作方法の候補に「長押しで溜める」「はじき入力」があり、すべて実装し検証した結果  
長押しでは刺している感がない。刺すという動作よりただの溜め攻撃になっていた。  
はじき入力は狙いが定めずらいなどの問題があった。



## こだわったポイント②

「刺す」の失敗の表現として  
はじかれてしなる拳動をこだわりました。  
また、刺す強さに応じてはじかれ度合いが  
変化するようにもしました。



なぜこのようにしたのか

失敗があるからうまくいったときの遊びがより楽しく感じれるので失敗の表現には  
こだわりたいと考えていました。  
このゲームは尻尾で刺すので、プレイヤーの視線が向く尻尾の先端からはじかれる  
ようにしました。



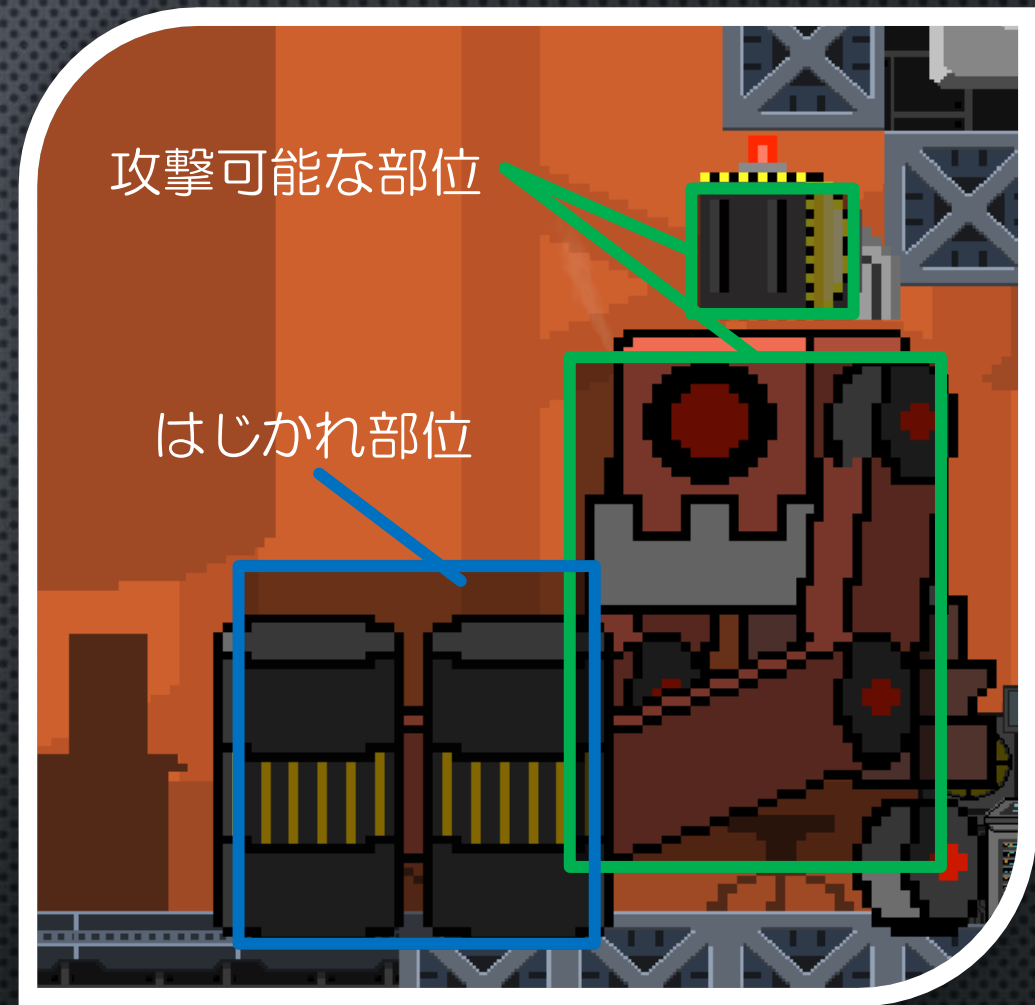
# こだわったポイント③

## 攻略方が複数生まれるようなボス制作

戦闘のテンポ感をヒットアンドアウェイにして狙って刺すという遊びをさせたいと考えボスを制作しました。刺せる場所を増やすために当たり判定を部位ごとに分けました。どの部位も狙うチャンスができるような攻撃をつくり程よく隙をいれました。

## なぜこのようにしたのか

ヒットアンドアウェイな戦闘の中でよく「狙って刺さないといけない、ミスしたあらはじかれて大きな隙を生んでしまう」という緊張感の中でうまく刺せた時は楽しいと考えたから。程よい隙と狙える場所が必要になるので、ハンマーを振り上げた時に狙える胴体と、弱点であるコンセントに当たり判定を分けました





## チーム構成

15名  
(プログラマー5、プランナー3、  
デザイナー4、コンポーター3)

## 担当箇所

- ・敵プログラム全般
- ・AOEシステム

## 作品概要

「ギリギリに迫るスリル」をコンセプトにした3Dアクションゲームです。  
チェーンソーで駆け回ってチェーンソーで敵を倒していく爽快さもあります。

## 開発環境

VisualStudio2022  
Unreal Engine 5.0.3



# こだわったポイント①

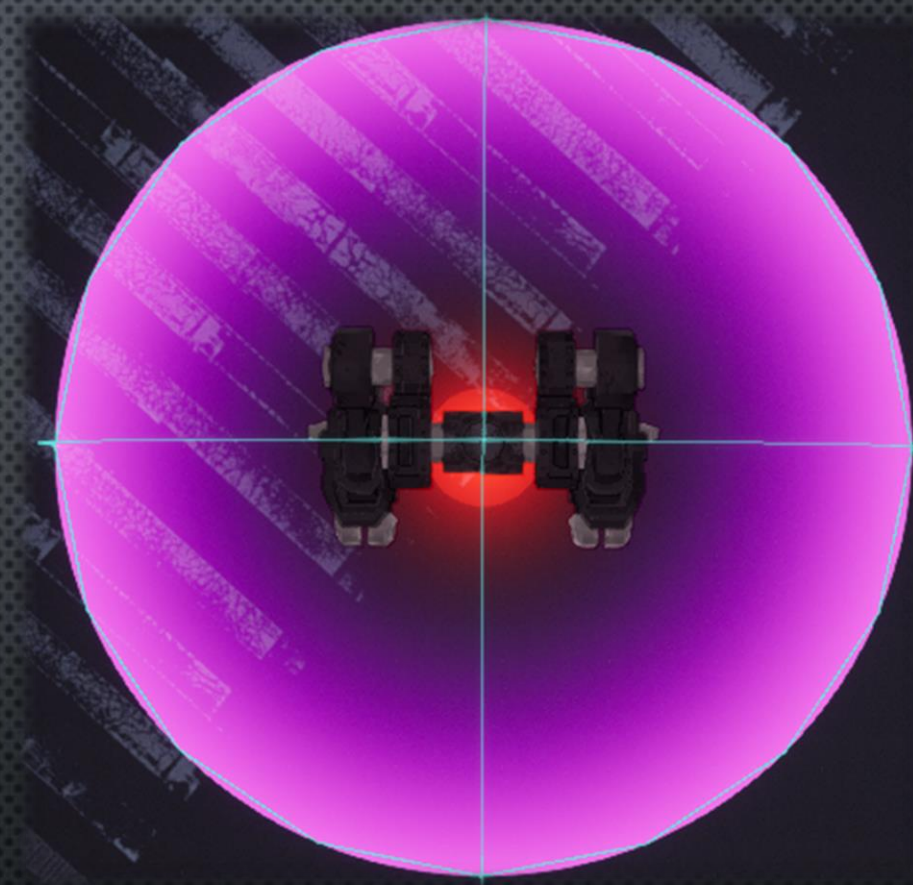
## 調整のしやすいエネミー制作

レベルデザイン担当の人が調整しやすいようにパラメーターを設定。AOEのサイズと当たり判定が**自動で一致**するなど余計な手間を取らせないようにしました。  
またアニメーションに合わせてタイミングなどを調整できるようにもしました。

※ パラメーター(関係のあるパラメータのみ抜粋)

▼ Aoe	
Aoe Circle Size	20.0
Aoe Circle Life Frame	300.0
Circle Col Frmae	3.0

※当たり判定：水色円 AOE：紫円



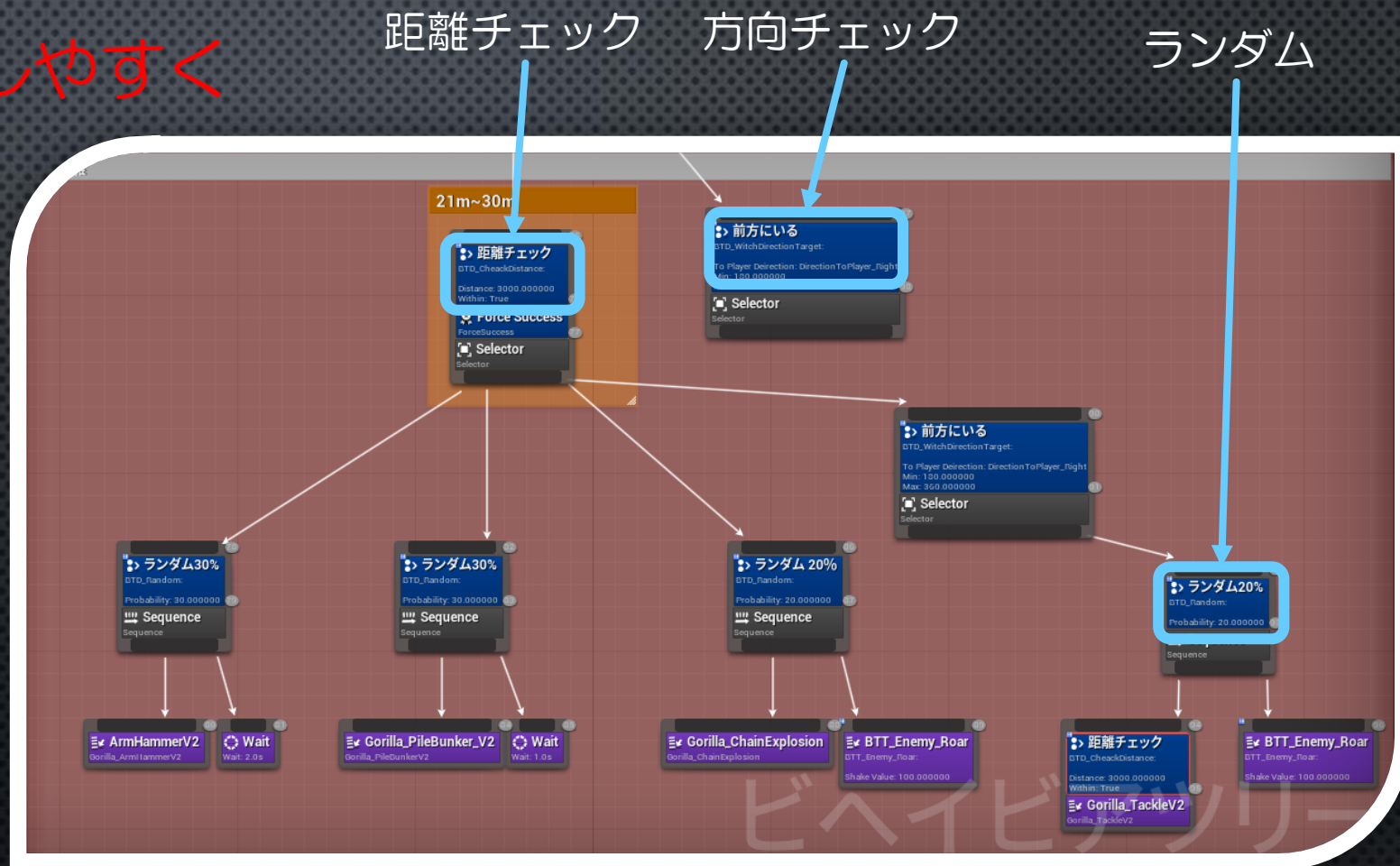
AOEのサイズを調整するだけで良い

当たり判定の継続フレーム  
AOEの継続時間も設定可能

# こだわったポイント②

## ビヘイビアツリーを調整しやすく

行動の発生条件を詳細に設定できるようにデコレーターを細かく作成。  
プログラマー以外の人  
がビヘイビアツリーを組めるようにしました。



作業効率と作りこみやすさがUP