

メカアクションゲーム
『DAEMON X MACHINA』
信念と血と鋼鉄の開発事例

MARVELOUS!



UNREAL ENGINE

#ue4fest

DAEMON X MACHINA

- 9/13 世界同時発売
- Nintendo Switch
- メカアクション
- 弊社内コンシューマタイトルで初のUE4採用
- UE4.20



UNREAL ENGINE

#ue4fest

本日の内容

- 前半 エンジニア編
 - 河野 雄太
- 後半 VFX編
 - 菊地 悟



UNREAL ENGINE

#ue4fest

エンジニア編

具体的なUE4機能活用事例と問題点・対処法

自己紹介

株式会社マーベラス チーフエンジニア

河野 雄太

MARVELOUS!

経歴

2006年に株式会社アートゥーンに新入社員として入社
そのまま会社合併による転籍で
2014年からマーベラスAQL（現マーベラス）に勤務

開発タイトル

アクションRPG
マルチプレイヤーアクション
Xbox360→Wii→PSVita→PS4→iOS/Android→Switch



UNREAL ENGINE

#ue4fest

DAEMON X MACHINA の構成

- 抛点とバトルシーンを行き来する
クエストクリア型のゲーム進行
- 4人マルチプレイ
- メ力も人間も細かくカスタマイズ
 - キャラクターエディット
 - フルキーコンフィグ・HUDカスタマイズ
- メ力ものだが、リアル寄りではなくコミック寄りの絵作り
- テキスト9言語、音声2言語



UNREAL ENGINE

#ue4fest

アジェンダ

- UE4運用方針
- キャラクター構成
- レベル構成
- 絵作り・レンダリング
- VFX・SFX
- デモシーン
- UI
- フルキーコンフィグ
- マルチプレイ
- ローカライズ
- パフォーマンス
- 自動プレイ



UNREAL ENGINE

#ue4fest

UE4運用方針（理想）

- UE4はなるべくそのまま使う
 - 改造し始めたらきりが無い
 - バージョンアップ時のマージ対応が悲惨
 - 共有しやすい、今後検証・拡張しやすい
- 本当に必要なものは自作する
 - プラットフォーム独自機能

UE4運用方針（現実）

- UE4は大半そのまま使った
- 絵作りに関わるレンダリングまわりは手を加えた
 - トゥーン表現
 - 平行光源を追加して3つ使えるように
 - ForceLODShadow のモデル別指定
- 機能修正
 - Cascade モジュール（Skel Vert/Surf Location）の一部修正
- バグ修正
 - 特にマルチプレイまわり多数



UNREAL ENGINE

#ue4fest

アジェンダ

- UE4運用方針
- キャラクター構成
- レベル構成
- 絵作り・レンダリング
- VFX・SFX
- デモシーン
- UI
- フルキーコンフィグ
- マルチプレイ
- ローカライズ
- パフォーマンス
- 自動プレイ



UNREAL ENGINE

#ue4fest

キャラクターモデル構成（アーセナル）

- メカ・高さ5~6m
- Characterクラスの派生、CharacterMovementも使用
- 複数のSkeletalMeshをアタッチして組み合わせる
 - アーマー換装
 - アーマー個別の可動パーツ
 - 武器換装

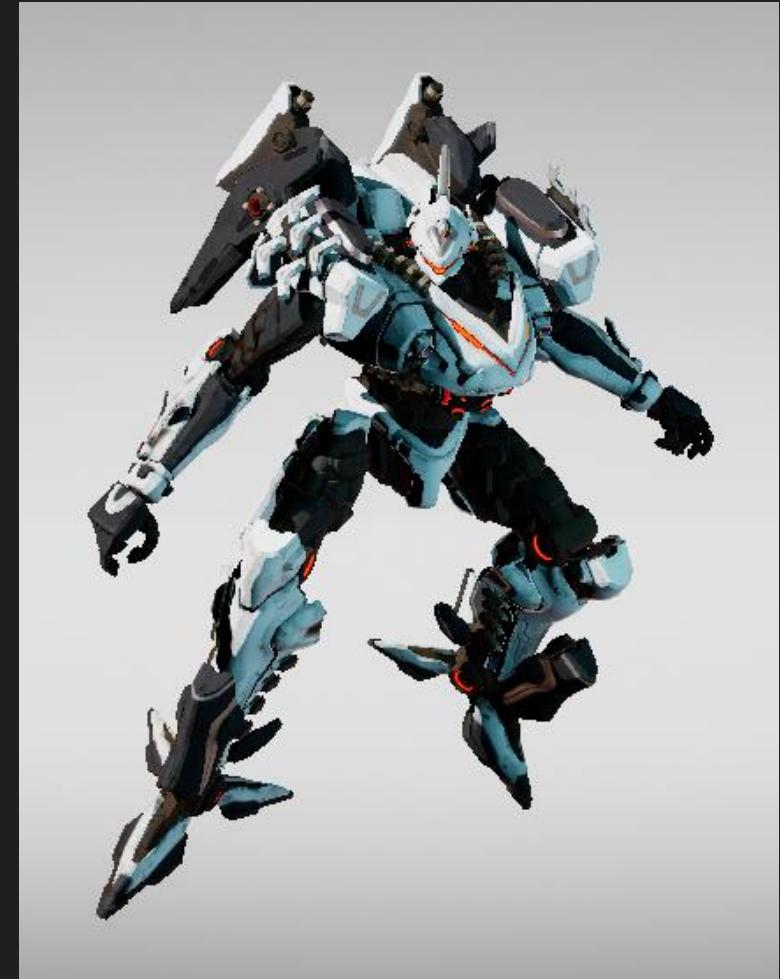


UNREAL ENGINE

#ue4fest

キャラクターモデル構成（アーセナル）

- 親となる骨のみの透明なSkeletalMesh
 - 全身アニメーションはここを動かす
- アーマー（頭・体・腕・脚モデル）
 - `SetMasterPoseComponent`
- パイロン
- 搭乗キャラクター（頭と体のみ）
- さらにスラスター・兵装モデル/VFX



キャラクターモデル構成（アーセナル）

可動パーツ

- 各アーマーにそれぞれアタッチ
 - 普通のSkeletalMesh
 - プレイヤーアクションに合わせて個別にアニメーション



キャラクター モデル構成 (アーセナル)

武器

- 右腕・左腕・肩・腰
- パイロンへのストックも
- それぞれ個別Actorのため
持ち替えや切り離し可能



キャラクター モデル構成 (アーセナル)

部位破壊表現

- モデル差し替えとVFX
- 頭や体部分が破壊されれば
搭乗キャラクターも見える



キャラクターモデル構成（アウター）

- 人間・高さ1.8m
- アーセナルと同様にCharacterクラス派生
- 複数のSkeletalMeshをアタッチ
 - キャラクターカスタマイズ
 - 改造による追加メカパーツ



キャラクター モデル構成（アウター）

- モデル・マテリアル・テクスチャと
マテリアルパラメータを切り替え
さらに細部のカスタマイズ
- 35名 + a のNPC達も同様のシステム



キャラクター モデル構成（アウター）

- 頭部はSetMasterPoseを使用せず独立している
 - 頭部にアタッチするモデルにSetMasterPoseを使用したいため（髪など）
 - 全身アニメーションと顔アニメーションの同時制御が複雑になったため



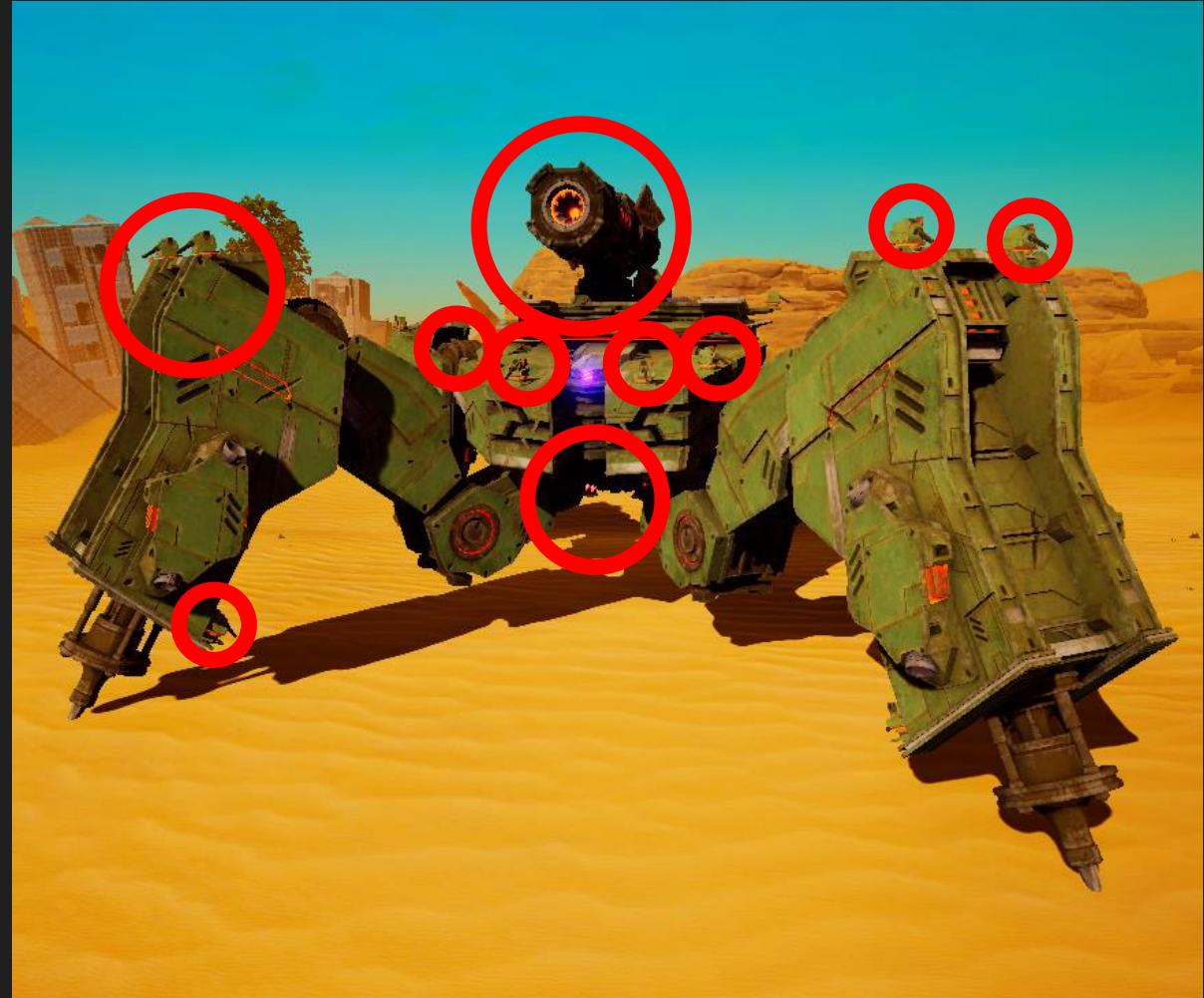
キャラクター モデル構成 (ザコ)

- メカ・小型エネミー
- Pawnクラス派生、単体モデル構成（例外あり）



キャラクター モデル構成 (ボス)

- メカ・大型エネミー
- Pawnクラス派生
- 本体は単体モデル構成
 - 砲塔や弱点部位などを別Actorでアタッチ



キャラクターモデル構成（ボス）



キャラクターモデル構成（問題点）

- 1Actorにアタッチするものが多く
CPUパフォーマンス面で多くの問題点（後述）

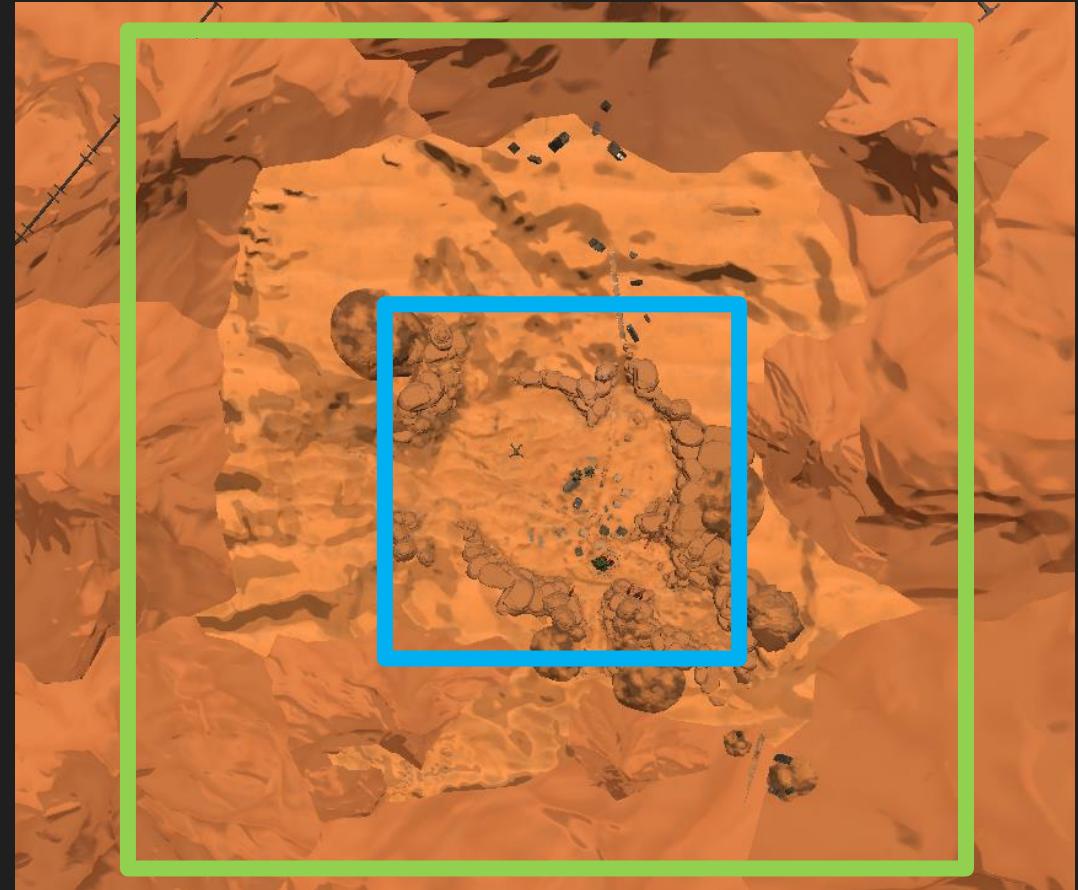


UNREAL ENGINE

#ue4fest

レベル構成（背景）

- 遠景まで およそ4km四方
 - かなりの高度まで飛行できるため
- プレイ可能領域 およそ2km四方
- 15種類のステージ
 - 配置・作戦領域のバリエーション
 - 時間帯・天候の違いも
- 背景用のBPはアーティスト製
 - 雷のポストプロセスなど



レベル構成（背景）

例) 雨の天候

- ライティング・天球・環境VFXのサブレベルを入れ替える
- 円柱状に発生する雨VFXをプレイヤーにアタッチ



UNREAL ENGINE

#ue4fest

レベル構成（ナビメッシュ）

- 内蔵のものを **Dynamic Modifiers Only** で使用
- キャラクターも地形もスケールが大きいため、設定変更



UNREAL ENGINE

#ue4fest

レベル構成（ナビメッシュ）

- ナビメッシュに影響するのは背景物のみ
- 破壊可能なオブジェクトはナビメッシュを書き換える



レベル構成（レベルBP）

- 組み方のルールを決め、エンジニアが機能提供を行う
 - エネミーを非表示状態でレベルに置く機能
 - エネミーをグループ管理する機能
 - デモシーンを流す機能
 - など
- 企画担当者がレベルBPを編集しレベルデザイン
 - サブレベルをON/OFFする、直接DestroyActorする、など
ダイナミックなことはあまり行わないでもらう



UNREAL ENGINE

#ue4fest

レベル構成（問題点）

ナビメッシュ

- 動く背景物がナビメッシュを書き換え続けてCPU負荷増大
 - サーチライトなど
- Dynamic Modifiers Only は事前パス計算結果が PersistentLevel に保存される
 - 地形の種類だけ PersistentLevel を分ける必要がある



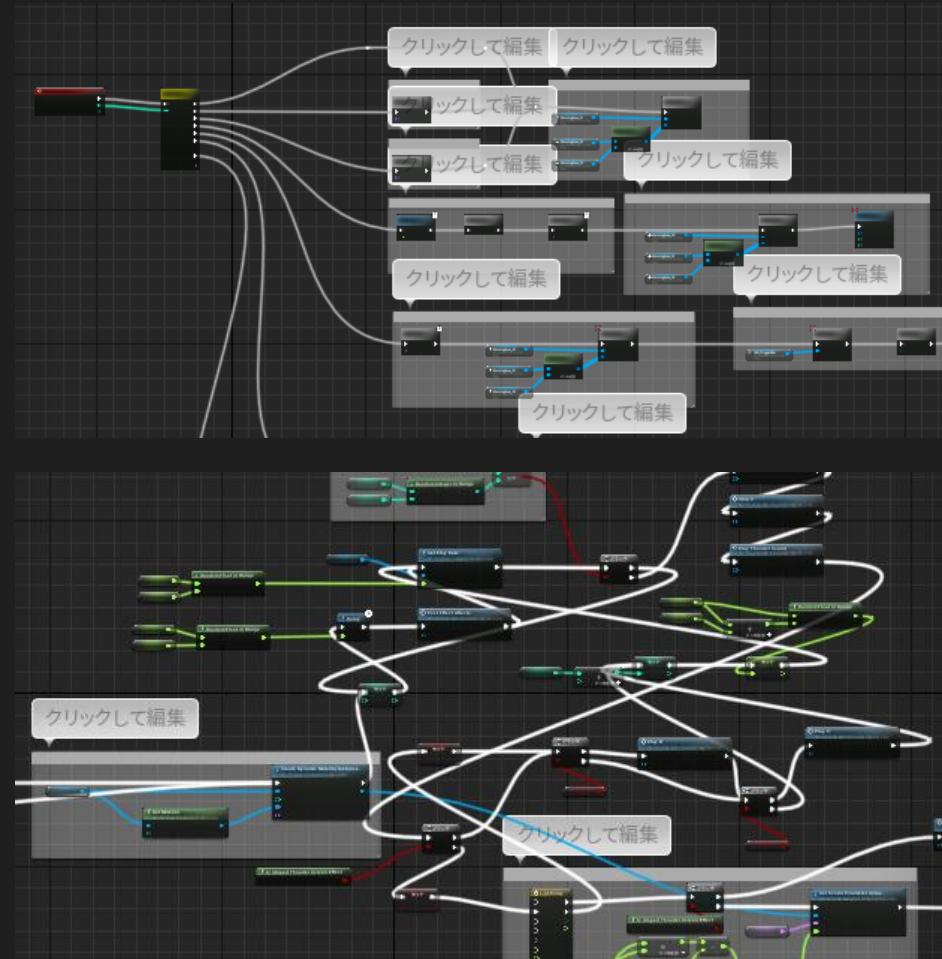
UNREAL ENGINE

#ue4fest

レベル構成（問題点）

BP

- BPの組み方にかなりの個人差
 - 許容する
 - エンジニアが介入することも



アジェンダ

- UE4運用方針
- キャラクター構成
- レベル構成
- 絵作り・レンダリング
- VFX・SFX
- デモシーン
- UI
- フルキーコンフィグ
- マルチプレイ
- ローカライズ
- パフォーマンス
- 自動プレイ



UNREAL ENGINE

#ue4fest

レンダリング（基本）

- デスクトップフォワード
- ライティングのベイク無し
- 環境光無し



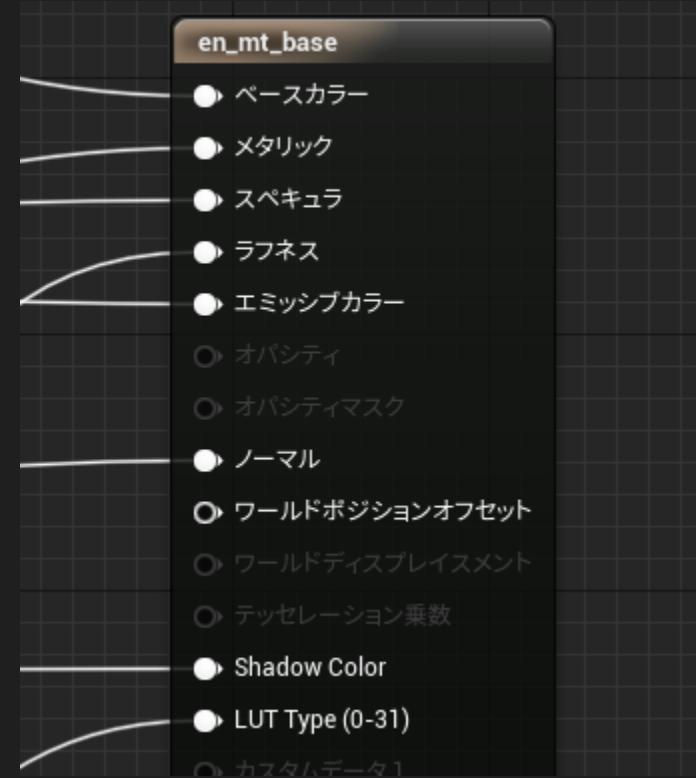
UNREAL ENGINE

#ue4fest

レンダリング（絵作り）

トゥーン表現

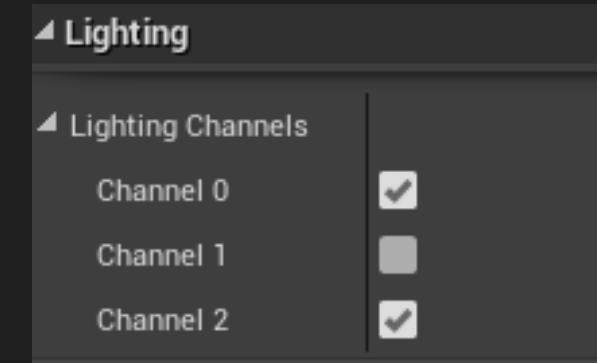
- テクスチャによるLUT
- シャドウカラー・リムライト
- 輪郭線は無し
 - 子供っぽい絵になってしまうのを防ぐ
- 参考
 - Unreal Engineを使用した商用タイトルでのノンフォトリアルレンダリング（NPR）事例
<https://www.slideshare.net/TatsuhiroTanaka1/unreal-engine-npr-80721783>



レンダリング（平行光源）

- フォワードレンダリングだと平行光源が1つ……
 - キャラライトや環境光のような表現のためエンジンを改変し平行光源を3つまで使用できるように

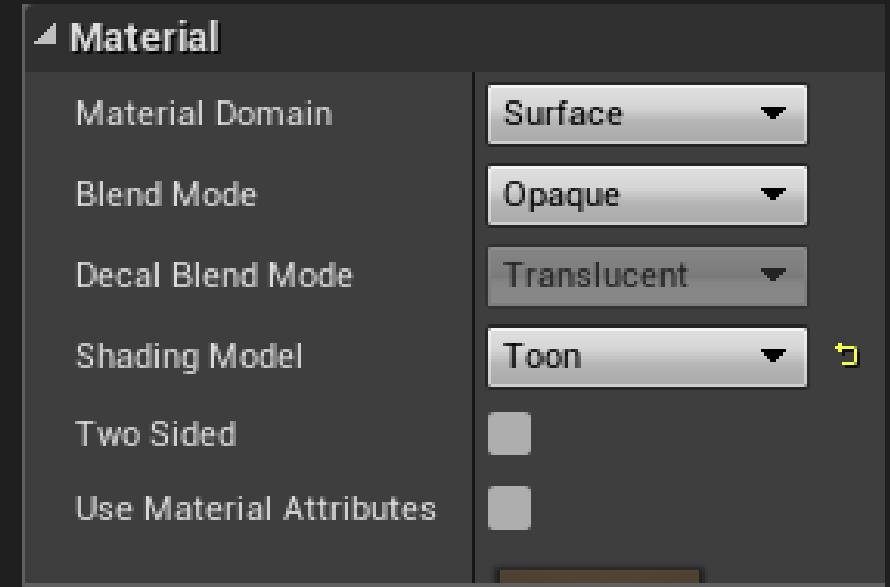
	Ch	スペキュラ	シャドウカラー	シャドウ	
通常ライト	0	有	有	有	
Exライト	1	有	有	無	カットシーン向け
Subライト	2	無	無	無	疑似的な環境光風



- どのライトを有効にするかはメッシュ毎にライトチャンネル指定

レンダリング（負荷対策）

- LODを徹底
 - StaticMesh・SkeletalMeshどちらも
- 半透明をなるべく使用しない
 - ブレンドモードは Opaque が基本
- Dynamic Resolution を有効に
 - `r.DynamicRes.OperationMode=2`
 - 4.20時点では正式対応版ではなかったが
おおよそ問題は無かった



レンダリング（負荷対策）

- `r.ForceLODShadow=4` (Scalability の設定)
 - 頂点ネックになることが多かったため、対策として影描画時のLODレベルを変更した
 - これでは粗くなりすぎ見た目が問題となるメッシュもあったためエンジンを改変し、メッシュ個別でも LODShadow を設定できるようにして回避

レンダリング（問題点）

- エンジンバージョンアップのたびに絵が崩れる
 - 絵作りのために大きく手を入れた箇所は毎度修正
 - 短くても1週間は安定しない
 - 元々恐れていた事態だったが、顕著なのはここのみだった



UNREAL ENGINE

#ue4fest

アジェンダ

- UE4運用方針
- キャラクター構成
- レベル構成
- 絵作り・レンダリング
- VFX・SFX
- デモシーン
- UI
- フルキーコンフィグ
- マルチプレイ
- ローカライズ
- パフォーマンス
- 自動プレイ



UNREAL ENGINE

#ue4fest

VFX・SFX

VFX

- Cascade をそのまま利用
- Skel Vert/Surf Location モジュールのみ修正
 - SkeletalMeshComponent を複数持っている場合への対応

SFX

- ミドルウェア CRI ADX2 / CRI Sofdec2
- CriWareプラグインは基本そのまま
 - 負荷対策のため一部改変して対応



UNREAL ENGINE

#ue4fest

カットシーン

- Sequencer をそのまま利用
 - Matinee と Sequencer で迷ったが、今後を考えて Sequencer
- イベントトラックと専用コンポーネントで拡張し
タイトル固有の機能を実装
 - 字幕表示・ボイス再生
 - 音声切り替えやミドルウェアとの兼ね合いで既存の Dialogue は使用せず
 - アーセナルの可動パーツ開閉・武器表示ON/OFF
 - アウターの目パチロパク制御・スーツ発光箇所の制御
 - など



UNREAL ENGINE

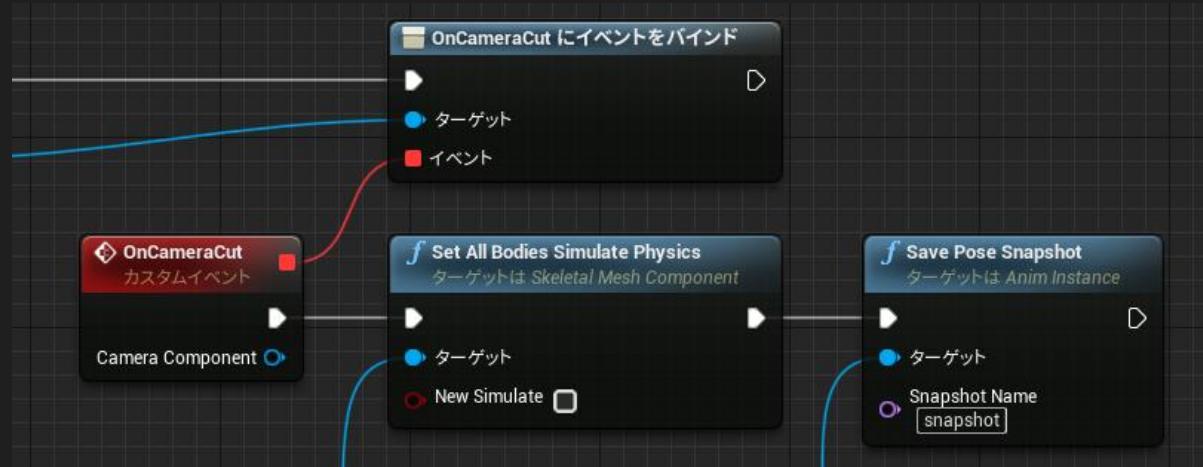
#ue4fest

カットシーン（問題点）

- Matineeと比べてドキュメントも事例も僅か
 - Sequencer のリリース初期だったため
- エンジンバージョンアップでSequencerの仕様が変わる
 - 例) 秒管理からフレーム管理に変更
- BindingしたActorの制御が難しい
 - 何もしないと各種コンポーネントが動いたまま誤動作
 - 最終的にBindするのはボスキャラクターのみに（例外あり）

カットシーン（問題点）

- カット切り替え時に髪の毛の物理が暴れる
 - Tick空回しができなかった
 - ULevelSequencePlayer の OnCameraCut で切り替え前後の処理
 - 全身の物理ON/OFF
 - 髪モデルの Animation Pose Snapshot 保存/適用



UI

- UMGをそのまま利用
- 開発初期の頃はテキスト中のアイコン表示ができなかった
 - UE4.20からできるようになった (**RichTextBlock**)



UI (問題点)

- CPU負荷が高い
 - GameThread以外では動かせない
- 全てUObjectで出来ているため、 UObject数が増大



UNREAL ENGINE

#ue4fest

フルキー・コンフィグ

- ActionMapping・AxisMappingは利用しなかった
- 既存のInputComponentで直接操作
 - UInputComponent::BindKey
 - UInputComponent::BindAxis
 - UInputComponent::BindVectorAxis ←ジャイロセンサー
- 入力とゲーム内アクションの変換テーブルを作成
- Joy-ConのDualモードはそのままだと左のジャイロセンサーしか取れない
 - 少し改造して右センサーを拾うように修正



UNREAL ENGINE

#ue4fest

アジェンダ

- UE4運用方針
- キャラクター構成
- レベル構成
- 絵作り・レンダリング
- VFX・SFX
- デモシーン
- UI
- フルキーコンフィグ
- マルチプレイ
- ローカライズ
- パフォーマンス
- 自動プレイ



UNREAL ENGINE

#ue4fest

マルチプレイ

- OnlineSubsystemSwitch をそのまま利用
- ListenServer形式
 - ClientAuthoritativePosition=true
移動処理はクライアント側
- 独自実装した点
 - レフェリー (Switch)
 - ランキング



UNREAL ENGINE

#ue4fest

マルチプレイ（問題点）

- ネットワーク構成がメッシュ型ではなくスター型
 - サーバーとなったプレイヤーに負荷が集中
 - ホスト移譲できない
 - ホスト切斷時にシングルプレイ化
 - Replication Graph は未検証
- OnlineSubsystemSwitch 内部のバグが頻発
 - 40件弱の修正
 - SDKの扱い方の不備
 - エラー処理
 - UGameEngine・UGameSession を継承してエラー処理

マルチプレイ（問題点）

- 同時に1セッションしか扱えない
&セッションがPersistentLevelに紐づいている
 - 全員同じレベルに居ないといけない
 - 4人ルームで3人だけバトルしたりできない
- マルチプレイ中にゲストがホームメニューに抜けると
ホストが NetMulticast のパケットを送らなくなる
 - Reliable設定のパケットがゲストに届かないまま消滅する
 - マルチプレイのみ、アプリケーションがBGフォーカス状態でも
プログラムが中斷しないようにハンドリングモードを変更

アジェンダ

- UE4運用方針
- キャラクター構成
- レベル構成
- 絵作り・レンダリング
- VFX・SFX
- デモシーン
- UI
- フルキーコンフィグ
- マルチプレイ
- ローカライズ
- パフォーマンス
- 自動プレイ



UNREAL ENGINE

#ue4fest

ローカライズ

- テキスト言語
 - ローカリゼーションダッシュボードをそのまま利用
 - AssetLocalization をそのまま利用
- 音声言語
 - テキスト言語切り替えとは別系統にするため
ミドルウェア側の機能と合うように独自実装



UNREAL ENGINE

#ue4fest

ローカライズ（問題点）

- poファイルの編集環境で良いものが無かった
 - Excelファイルと相互変換する補助ツール作成
- bSerializeAssetRegistry=true のまま
 - メモリを節約できると思ってfalseにすると
AssetRegistry を利用している AssetLocalization が動作しない



UNREAL ENGINE

#ue4fest

アジェンダ

- UE4運用方針
- キャラクター構成
- レベル構成
- 絵作り・レンダリング
- VFX・SFX
- デモシーン
- UI
- フルキーコンフィグ
- マルチプレイ
- ローカライズ
- パフォーマンス
- 自動プレイ



UNREAL ENGINE

#ue4fest

パフォーマンス対策（キャラクター）

- CharacterMovementComponent
 - 便利だがとにかくCPU負荷が高い
 - 一部キャラクターはCharacterクラス→Pawnクラスに変更
- キャラクター本体にアタッチする要素が多く
キャラクターを動かすたびにCPU負荷が上がる
 - 非表示でよい要素はギリギリまでアタッチしないでおく
 - アーセナル搭乗キャラクターはギリギリまで非表示＆アタッチされない
 - アタッチする要素を減らす＆まとめる

パフォーマンス対策（キャラクター）

- PrimitiveコンポーネントのナビメッシュOctree更新を抑制
 - UNavigationSystemV1::SetUpdateNavOctreeOnComponentChange(false)
 - バトル前後とオブジェクト破壊時に切り替え



UNREAL ENGINE

#ue4fest

パフォーマンス対策（キャラクター）

- 参考
 - [CEDEC2018] UE4で多数のキャラクターを生かすためのテクニック
<https://www.slideshare.net/EpicGamesJapan/cedec2018-ue4-111105729>
- SkeletalMeshComponent の最適化機能を細かく設定
 - Optimizationプロパティ
 - ゲーム中も状況に応じて細かくON/OFF切り替え
- Generate Overlap Events
 - ONだと当たり判定が大きいほどCPU負荷増大
 - 戦闘未参加のザコキャラクターはOFFになっている



UNREAL ENGINE

#ue4fest

パフォーマンス対策（ギミック）

- マップ内に配置された破壊可能オブジェクトがとても多い
 - 壊れるまで動かないので、静止時は徹底的に機能OFF
 - ActorはもちろんComponentもTickを止める
 - GenerateOverlapEvents をOFFにする
 - 可動性はなるべく EComponentMobility::Static
 - SetMobility(EComponentMobility::Static)
 - SkeletalMeshComponent も最適化設定
 - どれもゲーム中必要に応じて動的に切り替え



パフォーマンス対策（弾丸）

- 弾丸・着弾ヒット演出が発生するたびに様々なSpawn処理
 - 短時間で数百の弾丸発射と着弾、それに付随するVFX/SFX再生
 - CPU負荷大
 - UObject数が増大してGC負荷大
 - プール処理を作成して再利用し負荷軽減
- 弾丸Actorを SetActorLocation で移動させると負荷が高い
 - 飛翔開始時にコリジョンOFF、LineTraceで前方ヒット検知
 - ヒット直前にコリジョンONに戻す
 - AActor::SetActorLocation ではなく MoveComponent で移動させる
 - RootComponent->MoveComponent(..., MOVECOMP_SkipPhysicsMove, ...);



パフォーマンス対策 (GC)

UObject数減らし

- DisregardGCOBJECT
- BP減らし (=CPU負荷対策)
- BlueprintClustering
 - 問題があるという噂もあるが、本タイトルではON
 - 一部アセットが別クラスターに入ってしまい解放されない問題
- 参考
 - UE4におけるLoadingとGCのProfilingと最適化手法
<https://www.slideshare.net/EpicGamesJapan/ue4loadinggcprofiling>



UNREAL ENGINE

#ue4fest

パフォーマンス対策 (BP)

- **BP Tickは絶対ダメ、C++で。**
 - 開発初期でBP実装だったものをC++に置き換えるのに多大なコスト
 - アニメーションBPのイベントグラフもC++で書ける
- 余計な UFUNCTION・UPROPERTY でUObject数が増える
 - C++と絡むとバグが追いにくい
 - 本当に必要なもののみにしましょう
- BlueprintNativeEvent はBP実装が無くてもコスト大
 - 呼び出す際にBP実装があるか確認するため
 - C++で済むものはそもそも UFUNCTION を付けない



パフォーマンス対策（メモリ）

- `r.RenderTargetPoolMin 200`
 - 保留された不要なRenderTargetを解放する閾値
 - デフォルト400MB
 - 本タイトルは同時に150MBほどしか使用していなかったため
200MBに変更。メモリオーバー対策に大きく貢献。
- Bloomテクスチャ半減
 - Engine/Content/EngineMaterials/DefaultBloomKernel.uasset
 - エディタでエンジンコンテンツを表示し
このアセットを開いて Maximum Texture Size プロパティを編集
 - 2048 → 1024



UNREAL ENGINE

#ue4fest

パフォーマンス対策 (UMG)

- Visibilityが"Hidden"に設定されたWidgetはCPUを使い続ける
 - ほとんど Collapsed にする
- InvalidationBox も使用したが……
 - レイアウトが更新されないなどバグが頻発
 - 余裕を持って実装を



UNREAL ENGINE

#ue4fest

パフォーマンス対策（おまけ）

- 参考までに、一部のUObject数を公開します
- タイトル内容によって大きく前後する数値かと思いつますのであくまで参考程度にお願いいたします
- “obj list -alphasort”
 - Testビルド
 - 1-02 バトル開始時



UNREAL ENGINE

#ue4fest

パフォーマンス対策（おまけ）

Class	2765
Function	14646

ArrayProperty	2778
BoolProperty	13151
ByteProperty	3977
ClassProperty	810
EnumProperty	2757
FloatProperty	10169
IntProperty	4656
NameProperty	3354
ObjectProperty	13402
StructProperty	11801
TextProperty	548

Image	9764
Overlay	4903
OverlaySlot	11381
TextBlock	2769
Texture2D	1509
MovieScene	3144
MovieSceneColorSection	7714
MovieScene2DTransformSection	4427
MovieSceneBuiltInEasingFunction	26186
MovieSceneByteSection	13193
WidgetAnimation	3139

Total	281069
-------	--------

アジェンダ

- UE4運用方針
- キャラクター構成
- レベル構成
- 絵作り・レンダリング
- VFX・SFX
- デモシーン
- UI
- フルキーコンフィグ
- マルチプレイ
- ローカライズ
- パフォーマンス
- 自動プレイ



UNREAL ENGINE

#ue4fest

自動プレイ

- ランダムなマップに入り敵殲滅系バトルを繰り返す
 - シングルプレイ
 - オンラインマルチプレイ
 - ローカル通信マルチプレイ
- 集計内容
 - マップ別クリア数・リタイア数・クリアタイム
 - ハマリ検知
 - CPU/GPU負荷計測
 - ダンプファイルが出た場合はコールスタック内容



UNREAL ENGINE

#ue4fest

自動プレイ

- プロジェクト終盤から稼働
 - 1回で実機30台前後、最大40台超
 - 毎日AM1時～7時（土日も稼働）
- 初期リリースまでに合計50人月超の稼働実績
 - パッチ対応に向けて引き続き稼働中
- 日々集計しているので、通常のバグ報告と照合できる
 - 以前から発生していたフリーズか？どれくらいの頻度か？
 - 繁忙期のうっかりミスも翌朝拾える



UNREAL ENGINE

#ue4fest

まとめ

- UE4、そのままでも結構イケる
 - タイトル固有の絵作りはやはり手を入れた
 - 動作パフォーマンスも使い方次第
 - BP利用は厳格なルール決めを
 - バージョンアップには気を付ける
 - 各プラットフォームのマスター提出要件にも注意
- プラットフォーム依存の機能利用は注意深く
 - 対応済みと見せかけて、あまり使用されていない機能は十分な動作検証がされていない

今後の課題

- エンジンバージョンアップ
- アニメーション組み込みフローの見直し
 - アニメーションBPがエンジニア管理だった
 - アーティストが制御・調整しづらいという結果に
- マルチプレイ
 - ReplicationGraph の検証
- 自動プレイ
 - 集計するプレイ内容の充実
 - ランダムUI操作
 - 敵殲滅系以外のバトル攻略



UNREAL ENGINE

#ue4fest

次はVFX編です



UNREAL ENGINE

#ue4fest

メカアクションゲーム
『DAEMON X MACHINA』
信念と血と鋼鉄の開発事例

VFX編



UNREAL ENGINE

#ue4fest

1.はじめに



UNREAL ENGINE

#ue4fest

自己紹介



株式会社マーべラス リードアーティスト

菊地 悟

経歴

CGプロダクション、CS開発会社、マイクロソフト、マーべラス
ゲーム業界歴 20年

これまでCSを中心に多数のタイトルでVFXリード
UE3を使用した大規模開発経験
デモンエクスマキナではVFXリードアーティスト

VFXディレクション、マテリアル制作、環境整備、スケジュール管理
リソース/アセット管理、各セクションとの調整、実作業



UNREAL ENGINE

#ue4fest

概要

- 制作方針・環境
- アーティストとしてこだわった点
- 問題点



UNREAL ENGINE

#ue4fest

もくじ

1. はじめに
2. VFX制作方針・環境
3. アセット紹介
4. 直面した問題
5. まとめ



UNREAL ENGINE

#ue4fest

もくじ

1. はじめに
2. VFX制作方針・環境
3. アセット紹介
4. 直面した問題
5. まとめ



UNREAL ENGINE

#ue4fest

2.VFX制作方針・環境



UNREAL ENGINE

#ue4fest

VFX制作方針・環境

- ルック
- 制作方針
- 環境
- マテリアル
- レギュレーション
- UE4バージョンアップとカスタマイズ



UNREAL ENGINE

#ue4fest

プロジェクト・ルック

プロジェクトとして「トゥーン表現」を選択

- メカ物として「普通」にならないように「差別化」したい
- 金属表現が多い事からフォトリアルという選択も
- ただそれだと「普通」のルックに陥る可能性がある



メカ物だけれどトゥーン表現、という「差別化」



UNREAL ENGINE

#ue4fest

プロジェクト・ルック

トゥーン表現の方向性

- こちらも「普通」にならないように「差別化」を念頭に置く
- 近年に見られるような緻密なアニメーションを避け
- 80年代メカアニメやコミックのような、色や明暗がハッキリしたシンプルで特徴的な方向性を目指した



UNREAL ENGINE

#ue4fest

VFX・ルック

トゥーン表現の方向性

- プロジェクトの方針を受け、VFXの方向性を模索
- メカ系TVアニメをひたすら視聴（自宅の風呂場にて）
- デモンエクスマキナのVFXルックを定めていった



UNREAL ENGINE

#ue4fest

VFX制作方針

プロジェクト開始時

- PJスタッフの殆どがUE未経験者（自分ともう1名のみ、経験者2名）
- 準備期間や量産期間が限られ、膨大な数のアセットを作成する必要
- TAやVFX専任エンジニアはない
- UE4でのVFX制作は初、環境構築・ワークフロー等手探り
- トゥーン表現、仕組みから作ることは初



UNREAL ENGINE

#ue4fest

VFX制作方針

大枠の方針

あまり複雑にせず
使いやすく、量産しやすい点に重きを置く



UNREAL ENGINE

#ue4fest

VFX制作方針

デモンエクスマキナVFX制作で大切にした点

- アクションゲームアセットとしての機能性
- 処理負荷軽減
- 工数削減
- トゥーン表現
- クオリティ（メカ物としての表現を大切に）
- VFX班内、情報共有の徹底



UNREAL ENGINE

#ue4fest

制作環境

環境

- 最終バージョン UE 4.20 (Cascade)

DCCツール

- Maya
- 3ds Max
- Substance Designer
- Houdini



UNREAL ENGINE

#ue4fest

マテリアル

構成検討

- 過去のUE3大規模プロジェクトの際、マテリアルを一部作成した経験
 - 処理負荷が問題になったので、気を配る
 - マテリアルの精査、管理をしっかり
- UE4構成検討
 - マテリアルとマテリアルインスタンスの仕組みと問題点の共有 (Epic Games Japan: 篠山範明) #UE4DD
<https://www.slideshare.net/EpicGamesJapan/epic-games-japan-ue4dd>

マテリアル

マテリアル構成概要

- 「Usage」別に分ける
- 用途を「汎用」「専用」で分ける
- 「ライティングされる・されない」で分ける
- 参照構造「親、子、孫」タイプ
- 命名規則は試行錯誤して決定

※詳細は割愛



UNREAL ENGINE

#ue4fest

マテリアル

ブレンドモードとゲーム要素

- プレイヤー、両手武器、肩武器を同時使用可能
- プレイヤー、最大4体（分身で8体）
- エネミー、多く表示（ザコ、ボス）
- 処理負荷の懸念
- トゥーン表現、ディゾルブが多い予測
- 不透明チックな描画でも許容できる予測



マテリアル

以上の事から、ブレンドモードは…

半透明は極力使わずに
Masked中心



UNREAL ENGINE

#ue4fest

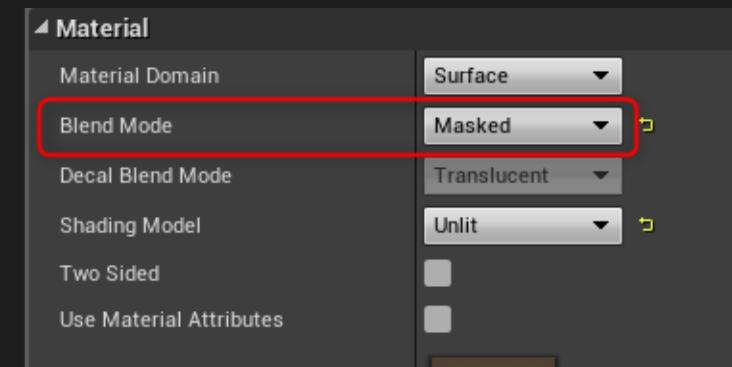
マテリアル

理由

- Maskedも半透明だが、Translucentほど処理を食わない
- 描画順を、あまり考慮しなくてもよい
- トゥーン表現、処理負荷の両立が出来、一石二鳥

※一部Opaque、Translucent、Additive等も使用

思い切ったが
この決断は正しいものだった



レギュレーション

要素（詳細割愛）

- テクスチャ
- スタティックメッシュ
- マテリアル・マテリアルインスタンス
- エミッタ・モジュール設定(Cascade)
- パラメータ(Cascade)
- 物理マテリアル
- LOD
- コリジョンメッシュ



UNREAL ENGINE

#ue4fest

レギュレーション

チェックリスト

- データアップ前にチェック
 - 細かくしすぎると、VFXスタッフの負担

重要度をつけてチェック



レギュレーション

チェックリストの重要性

「アナタはなぜチェックリストを使わないのか？」

アトウール・ガワンデ：著 / 吉田竜：翻訳 晋遊舎



UNREAL ENGINE

#ue4fest

UE4バージョンアップとカスタマイズ

過去UE3の大規模開発の際、CASCADEを積極的にカスタマイズ
バージョンアップで、**とても苦労した経験**



バージョンアップの影響を考え
カスタマイズは行わない

※エンジニア側で「Skel Vert/Surf Location」ノードのみ手をいれています

UE4バージョンアップとカスタマイズ

バージョンアップの注意点

- 移行前の環境を残す
- 移行後の変化をチェック、対応箇所の洗い出しと対応
- データで修正出来ない箇所については、エンジニアに依頼し順次対応

無事に移行出来了

もくじ

1. はじめに
2. VFX制作方針・環境
3. アセット紹介
4. 直面した問題
5. まとめ



UNREAL ENGINE

#ue4fest

3. アセット紹介



UNREAL ENGINE

#ue4fest

アセット紹介

- a. 弾
- b. 煙
- c. 爆発
- d. スライド移動
- e. 範囲
- f. スラスター
- g. 地表煙



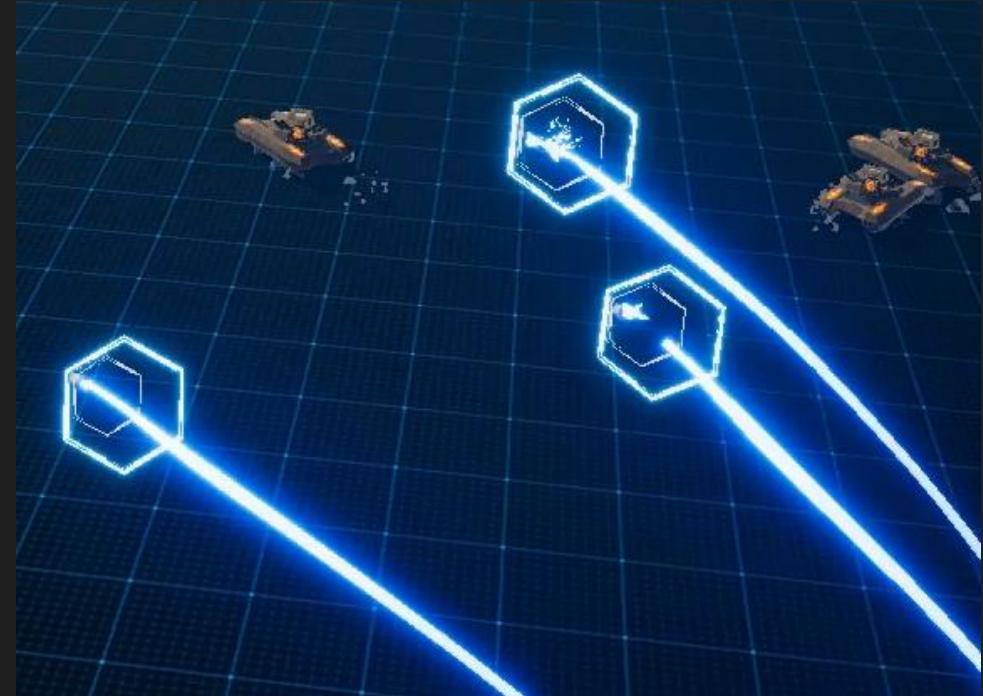
UNREAL ENGINE

#ue4fest

a.弾・ミサイル・初期



a.弾・ミサイル・最終



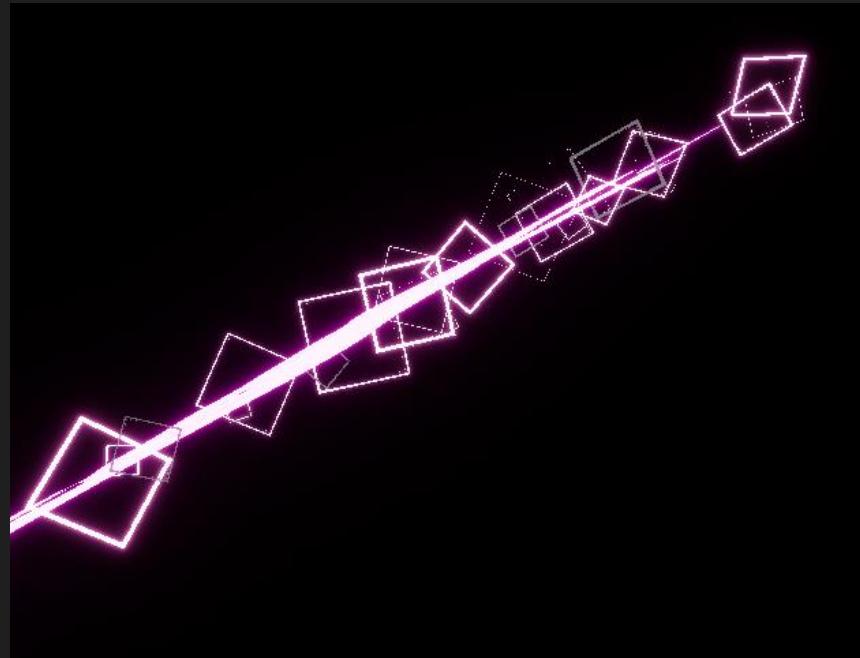
a.弾・スナイパーライフル・初期



UNREAL ENGINE

#ue4fest

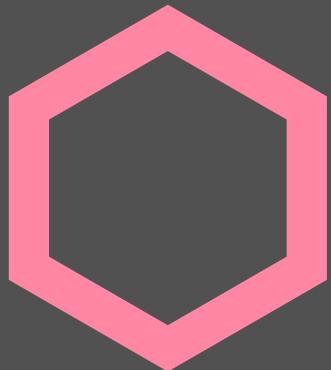
a.弾・スナイパーライフル・最終



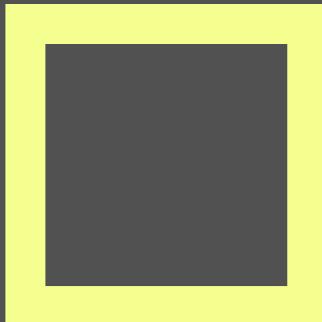
UNREAL ENGINE

#ue4fest

a.弾・記号化



ミサイル：六角形



スナイパー：四角形

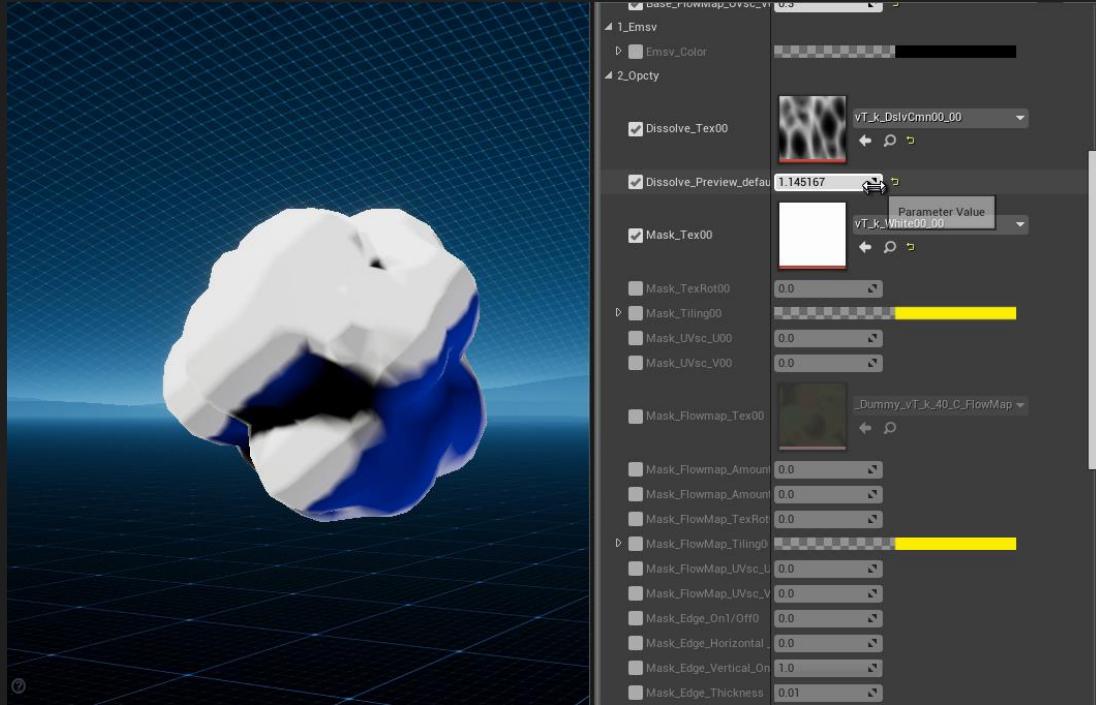


ブリッツ：三角形

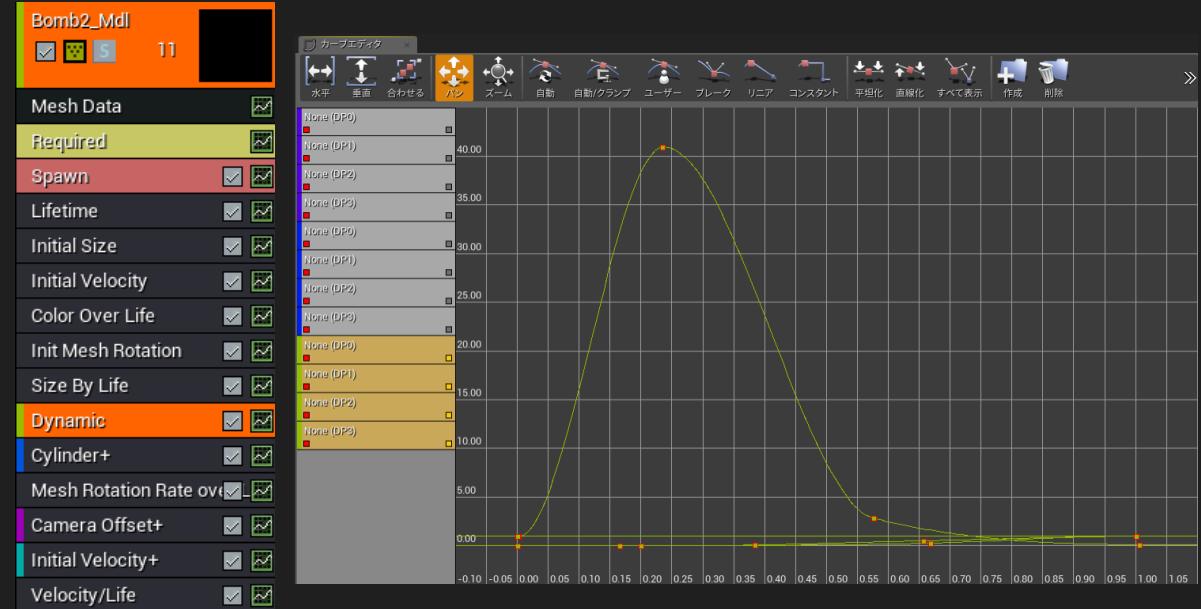
記号化

b.煙

ディゾルブマテリアル



Dynamicパラメータでの制御



直感的ではなく面倒



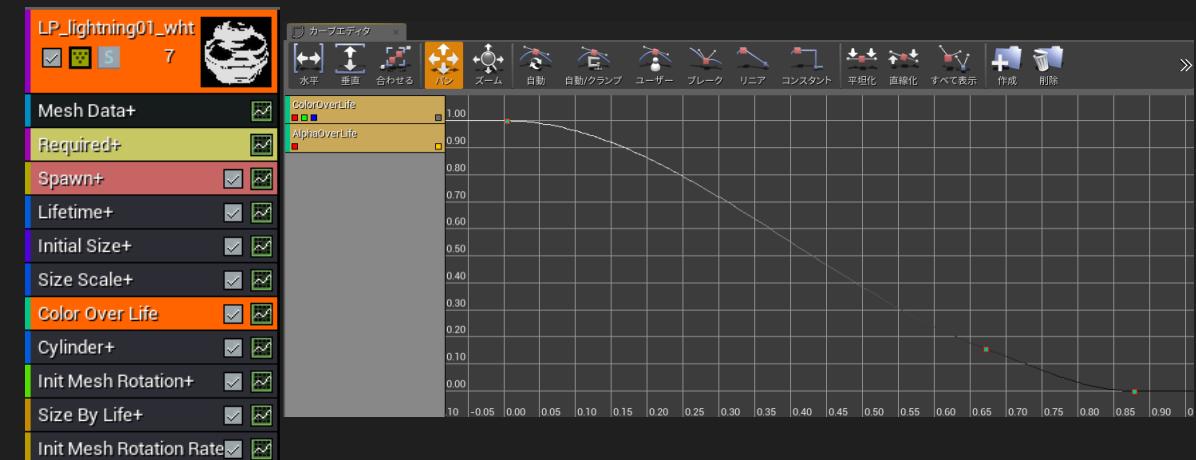
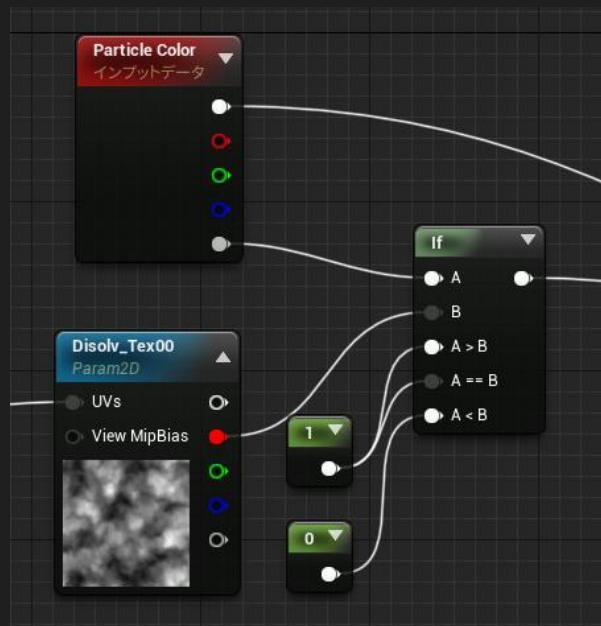
UNREAL ENGINE

#ue4fest

b.煙

ディゾルブマテリアル

- 「Color Over Lifeモジュール」Alphaカーブによるディゾルブ制御



直感的な制御

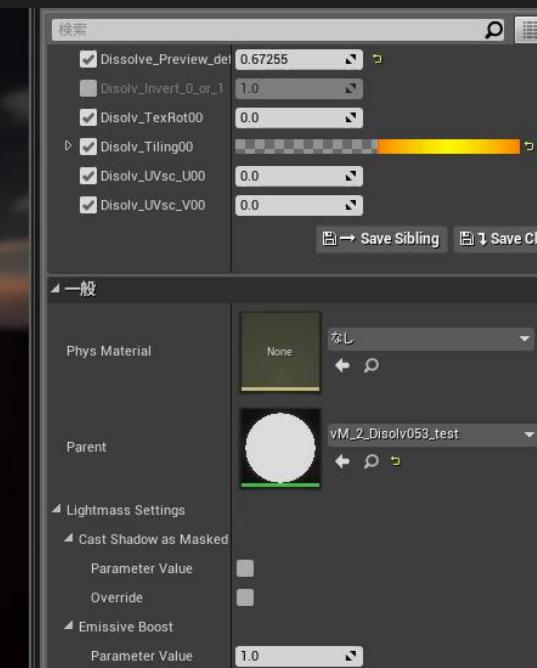


UNREAL ENGINE

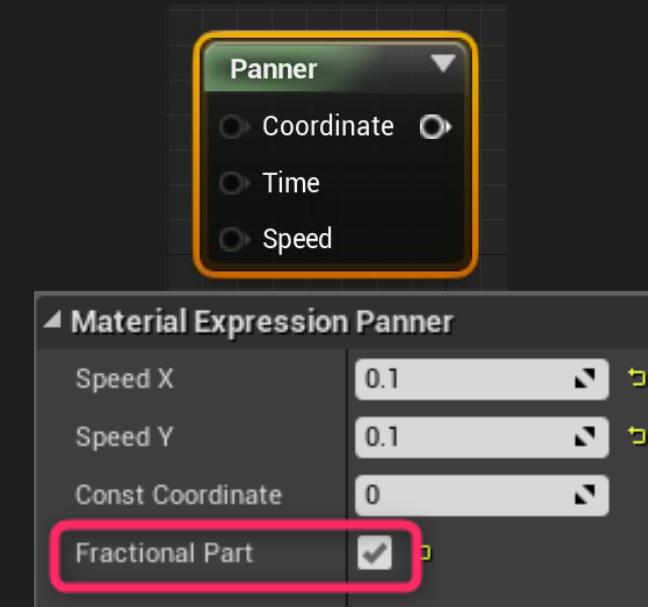
#ue4fest

b.煙

UVスクロールによる荒れ



Pannerノード



ガタツキの解消（精度）



UNREAL ENGINE

#ue4fest

b.煙

テクスチャによるディゾルブのエッジ荒れ

- 当初DXT1を使用
解像度を上げても解決しない
- GrayScaleテクスチャで解決
低い解像度でも綺麗に



b.煙



DXT1 512 x 512



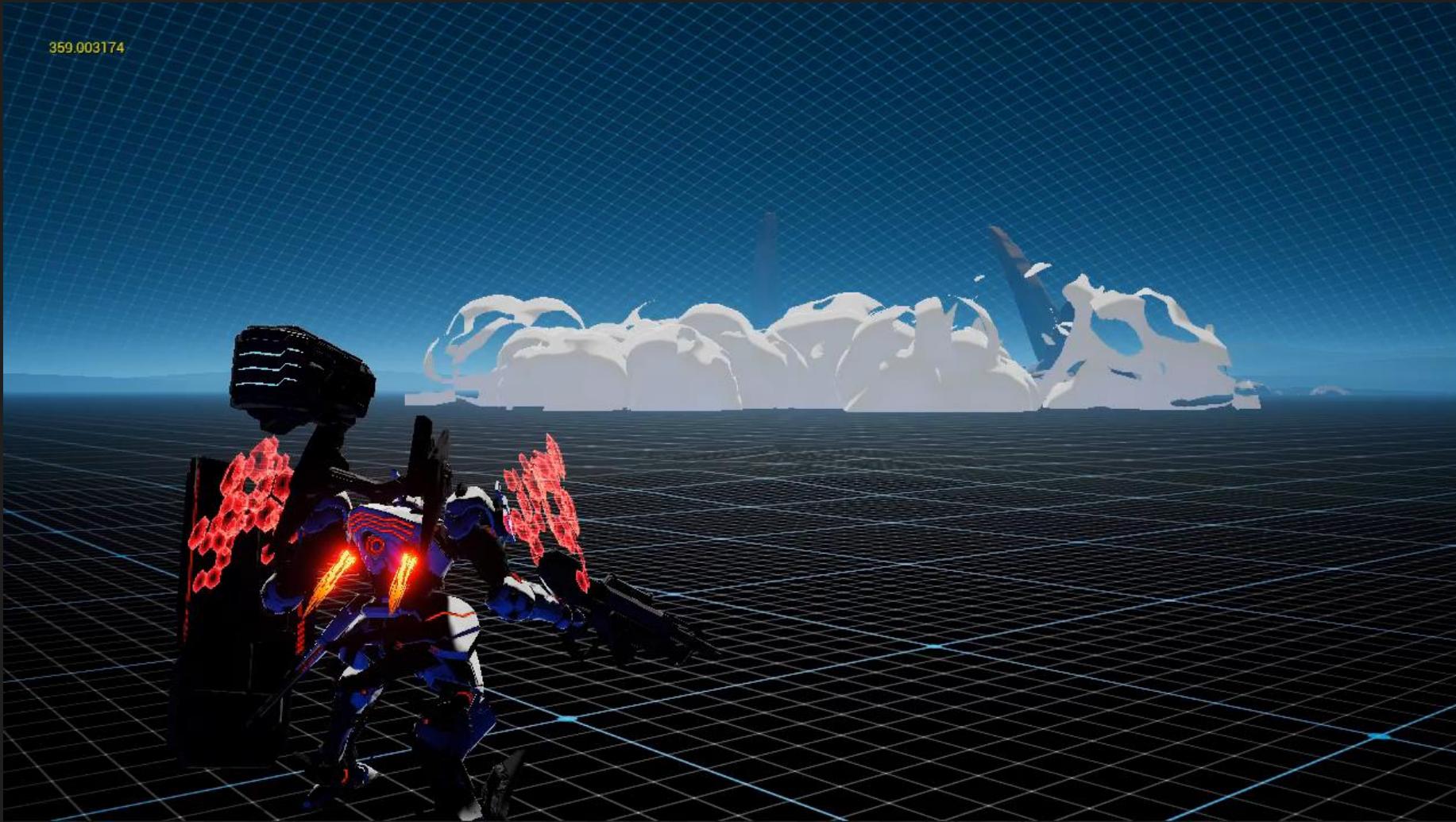
GrayScale 128 x 128



UNREAL ENGINE

#ue4fest

b.煙



UNREAL ENGINE

#ue4fest

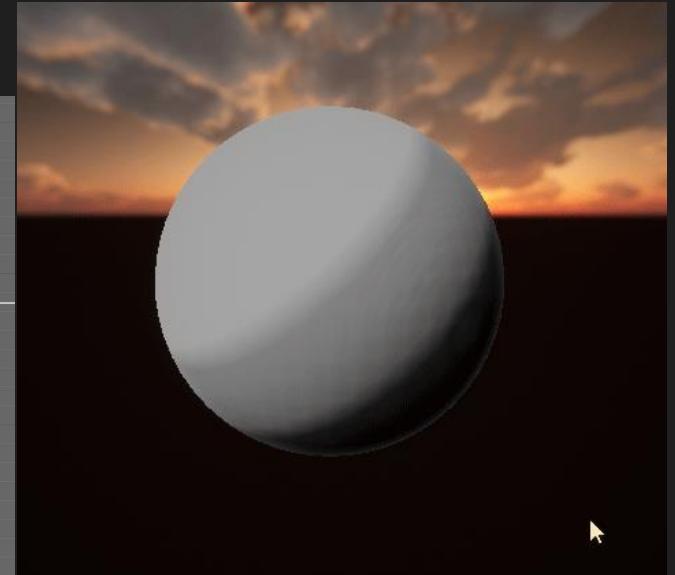
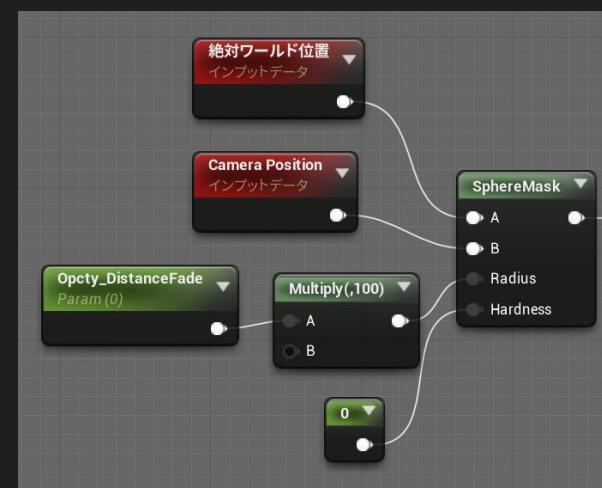
b.煙

大きな煙が画面を覆ってしまう問題

- カメラからの距離に応じて煙に穴をあける

参考 :

UE4 SphereMaskで距離に応じて色
を変える (mokoさん)
<http://effect.hatenablog.com/entry/2018/05/16/025859>



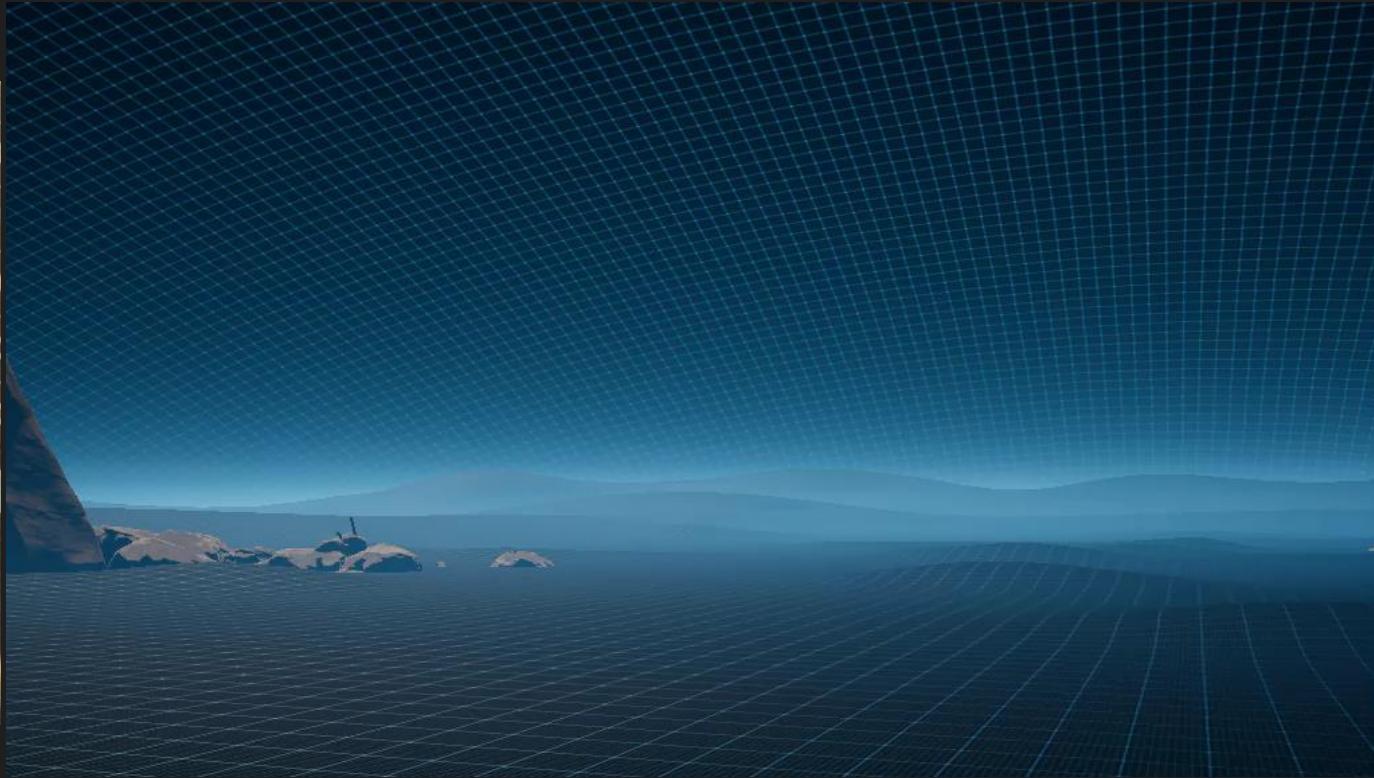
b.煙



UNREAL ENGINE

#ue4fest

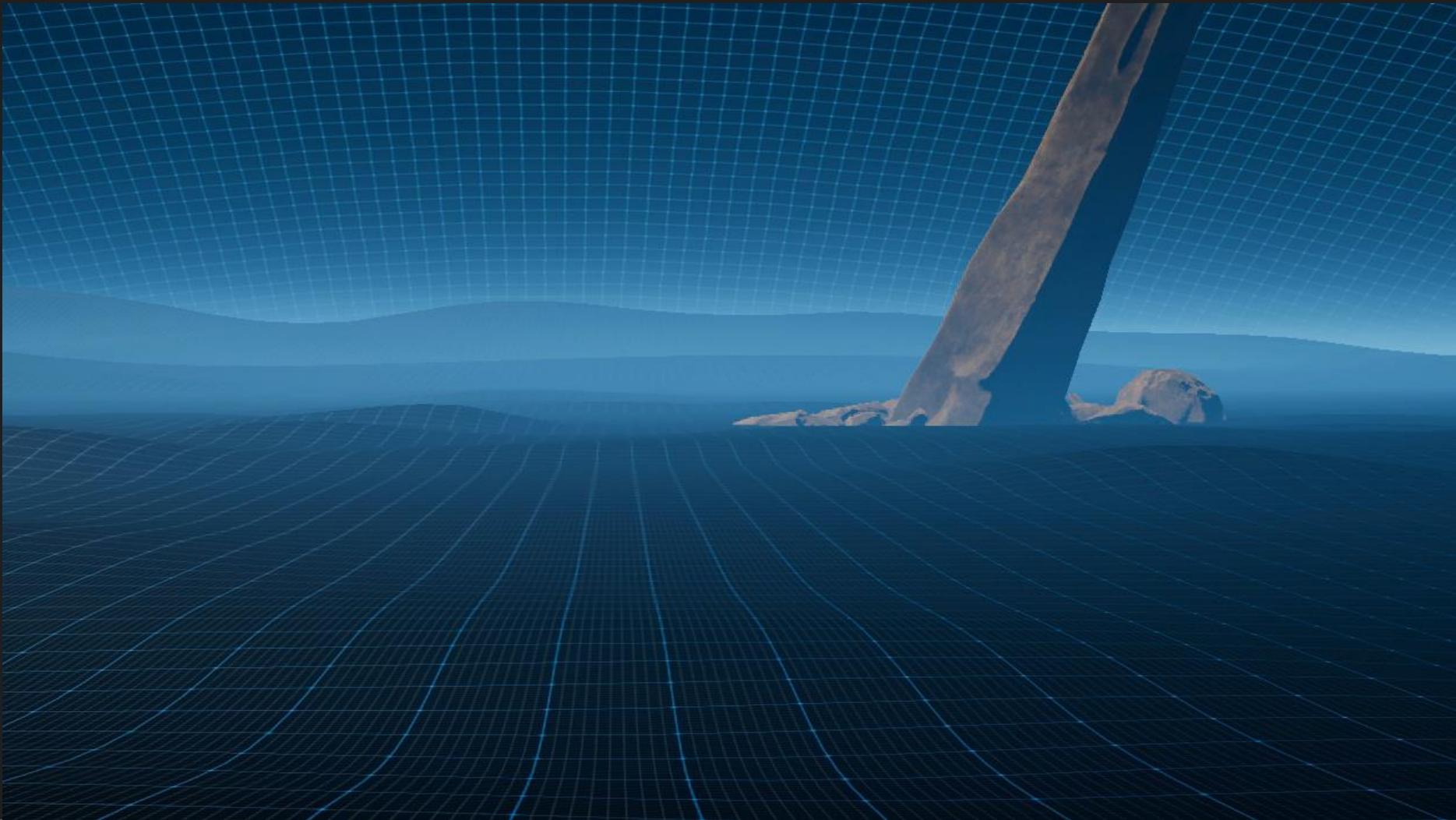
C.爆発・雑魚工ネミー



UNREAL ENGINE

#ue4fest

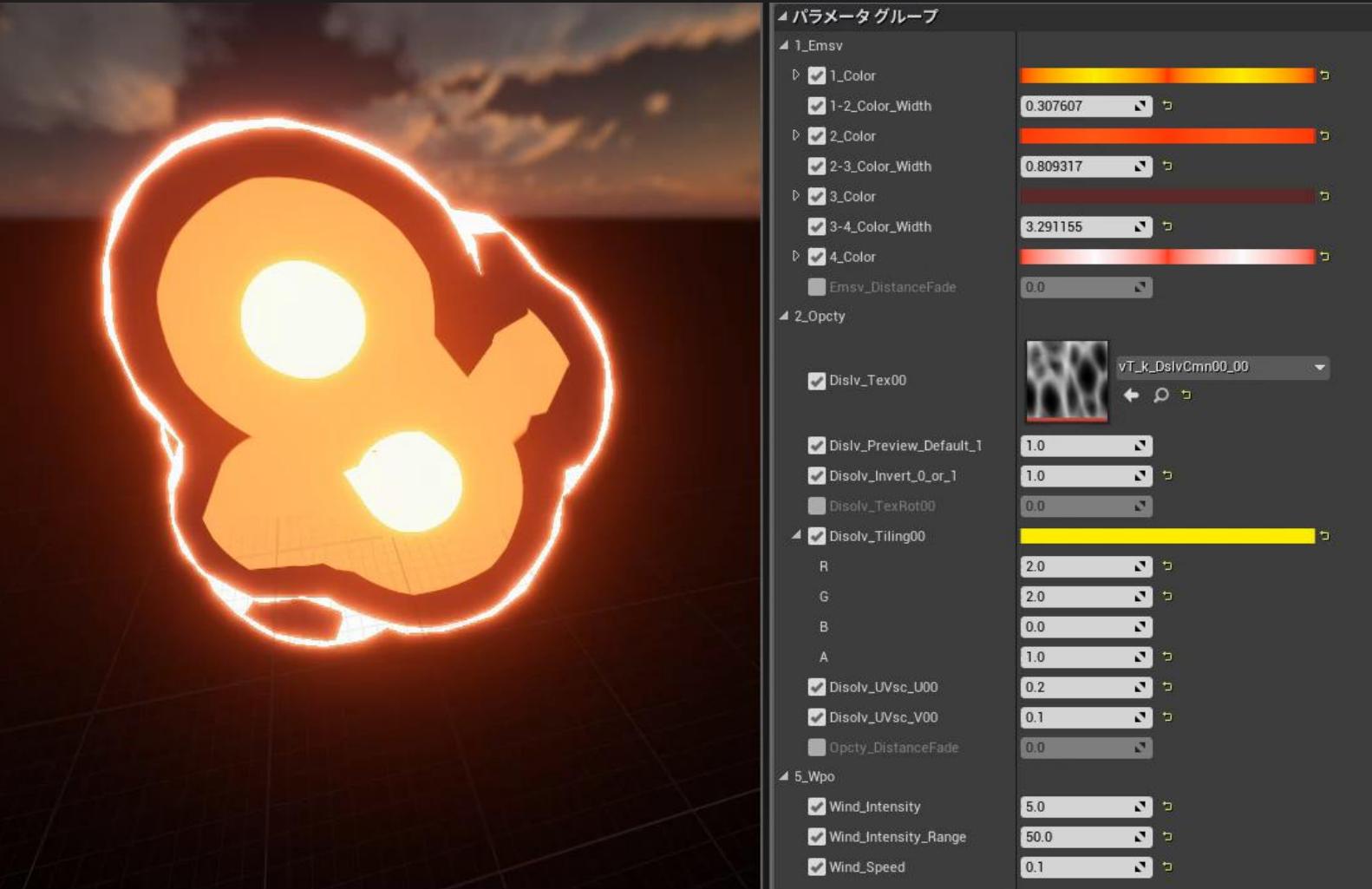
C.爆発・雑魚工ネミー



UNREAL ENGINE

#ue4fest

c.爆発・赤熱マテリアル



UNREAL ENGINE

#ue4fest

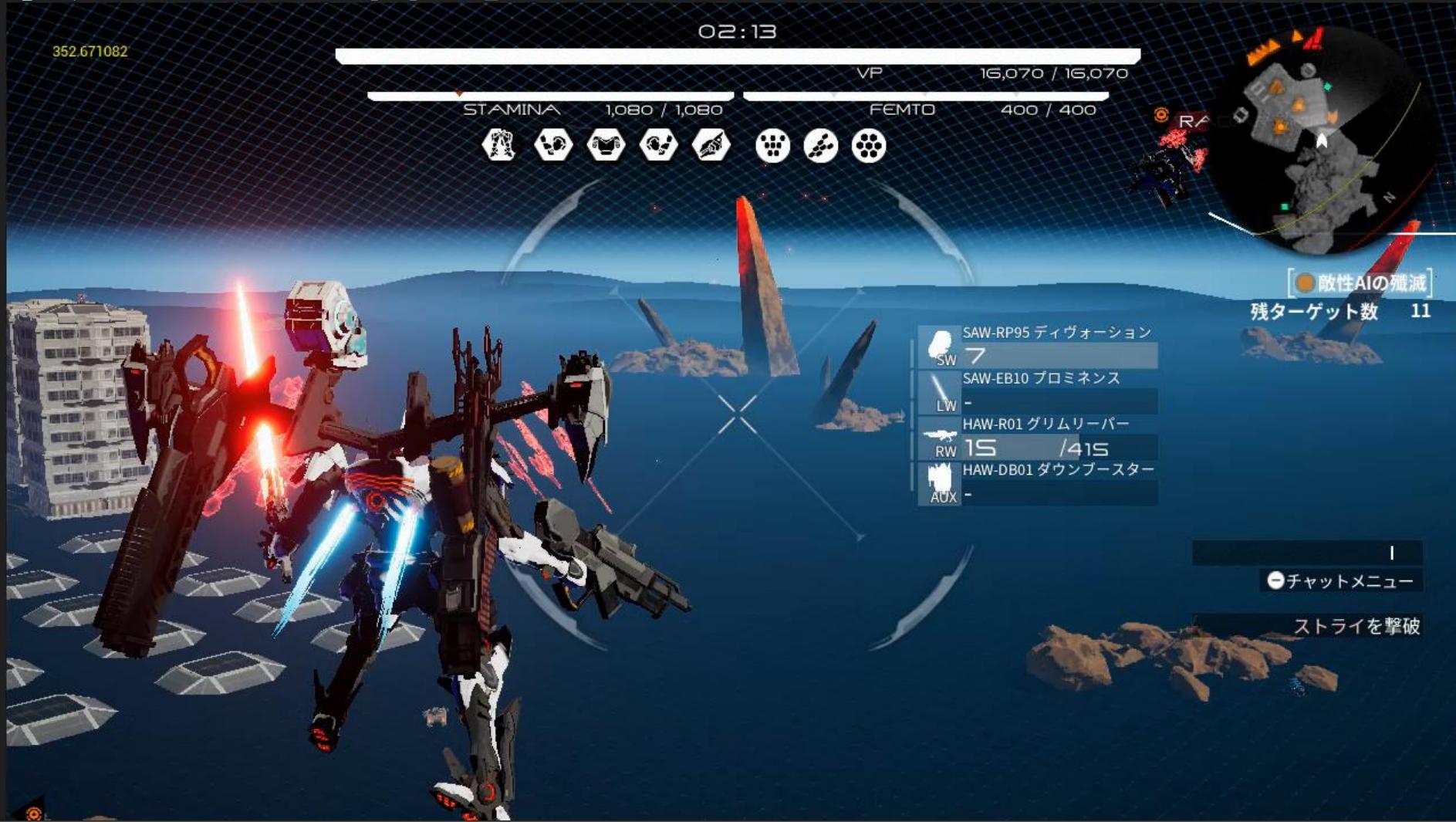
C.爆発・赤熱マテリアル



UNREAL ENGINE

#ue4fest

C.爆発・墜落時



UNREAL ENGINE

#ue4fest

C.爆発・墜落時

アーセナル（プレイヤー、エネミー）の墜落時のVFX

- 達成感を伝える
- メカからパーツ武器を剥ぎ取る仕様

縦長にし墜落地点を強調



UNREAL ENGINE

#ue4fest

d.スライド移動



UNREAL ENGINE

#ue4fest

d.スライド移動

メカのスライド移動の軌跡部分

- トレイル、PCでは問題ない
- 実機でトレイルが暴れる現象



d. スライド移動

原因

- トレイル終端のスケールアニメーションが影響
- 「Size By Life」を消したら暴れなくなった



UNREAL ENGINE

#ue4fest

e.範囲・範囲爆発



UNREAL ENGINE

#ue4fest

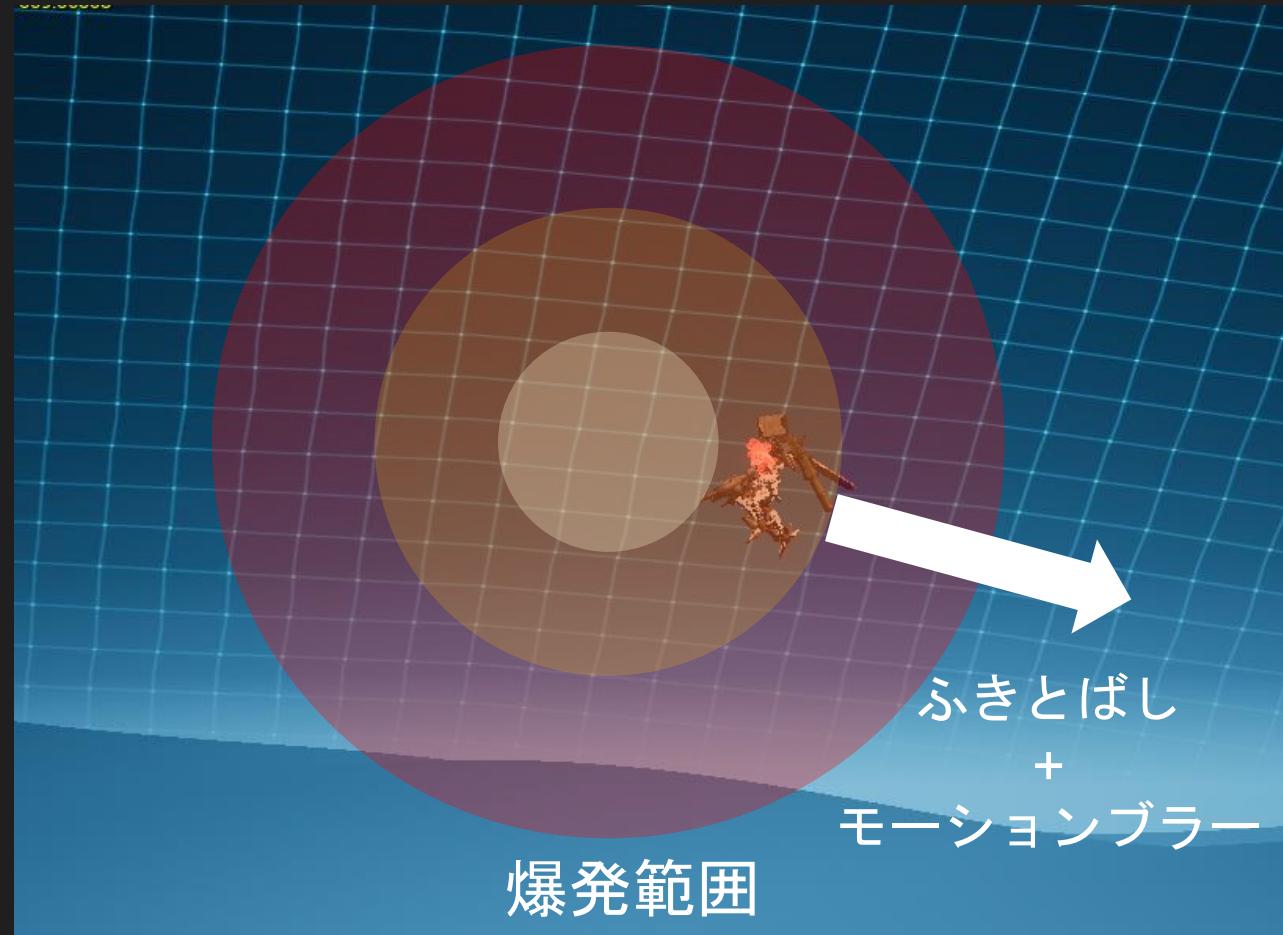
e.範囲・範囲爆発

爆発を受ける際

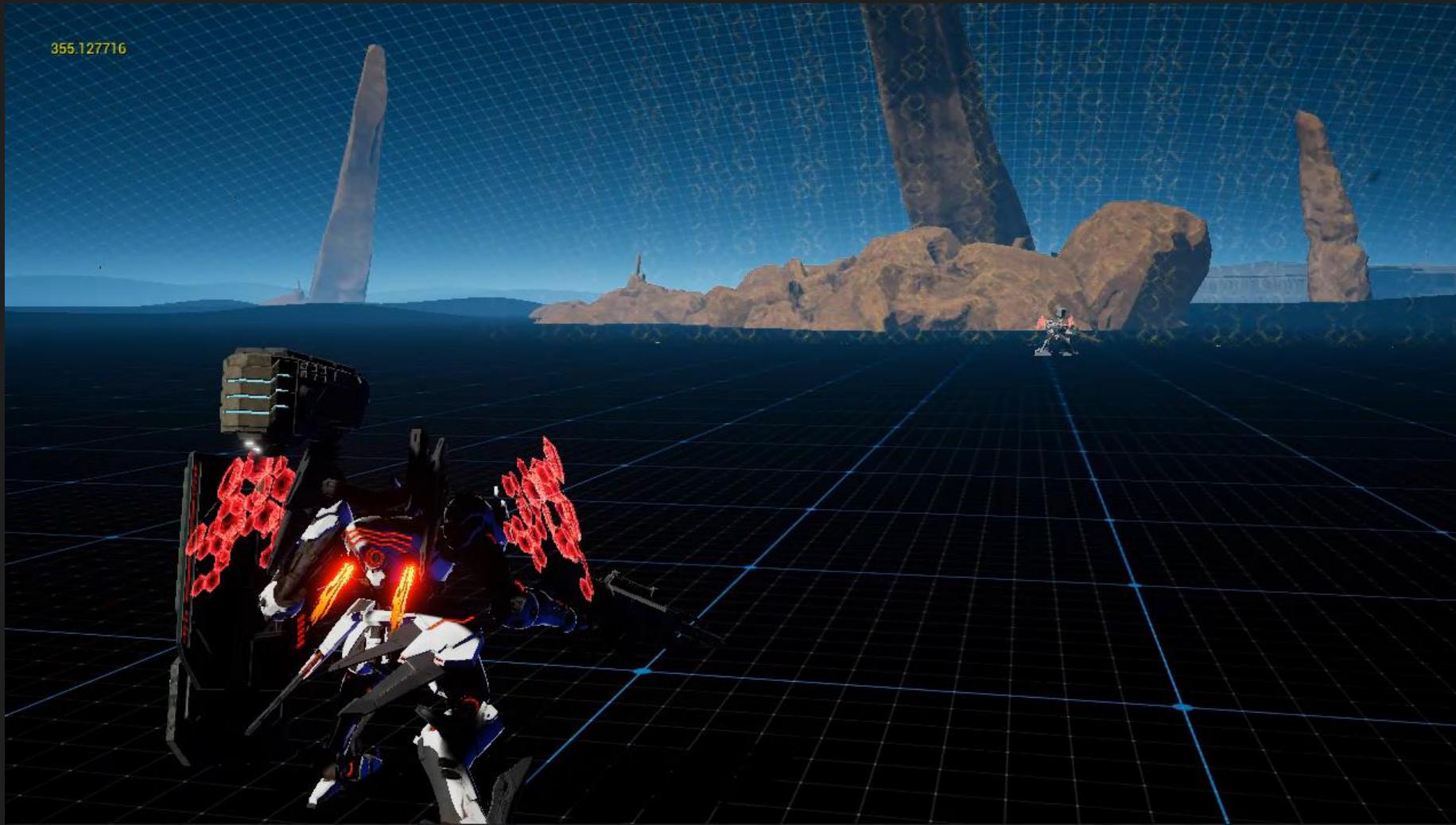
- 爆発内、説得力が必要
- 爆発内、実はスカスカ

提案

- ふきとばし
- モーションブラー（一瞬）



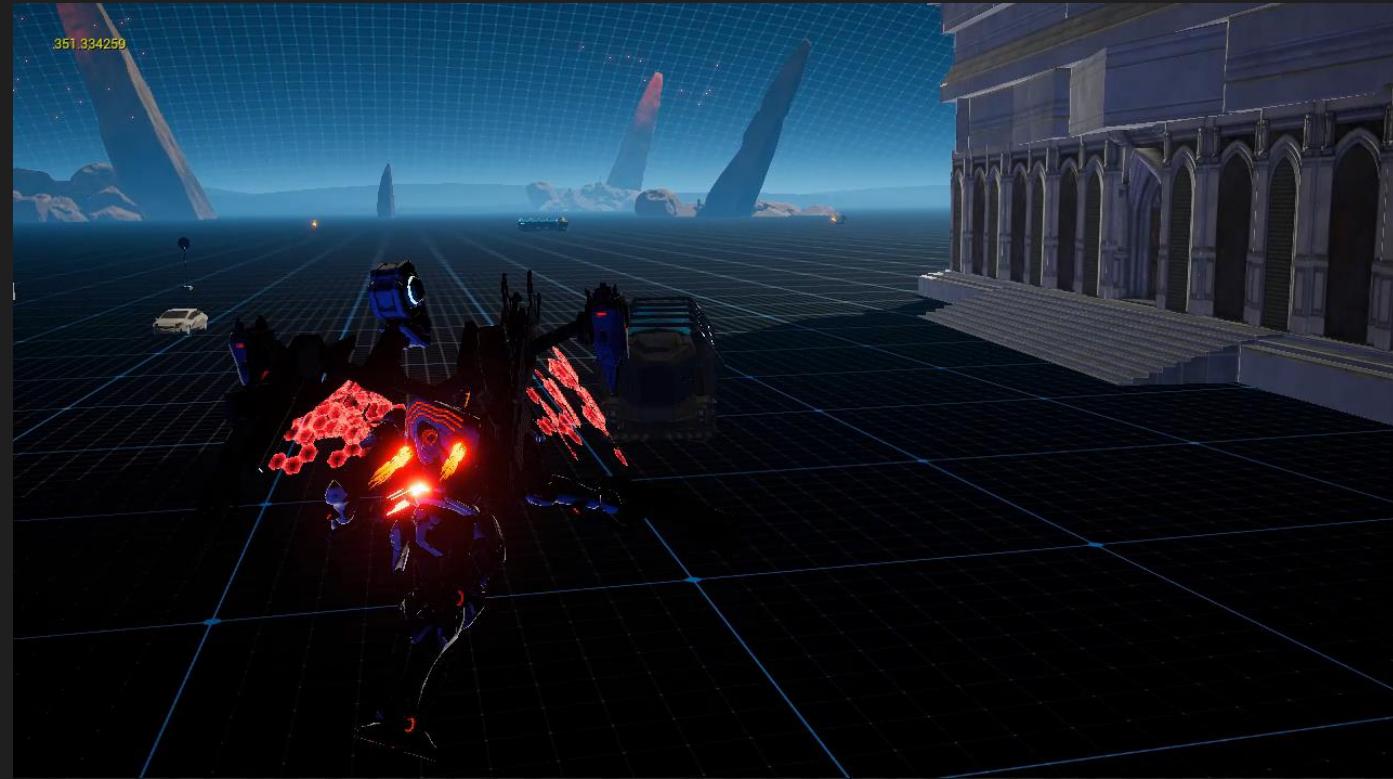
e.範囲・ダメージ爆発



UNREAL ENGINE

#ue4fest

e.範囲・回復空間



e.範囲・粒子兵装シールド



UNREAL ENGINE

#ue4fest

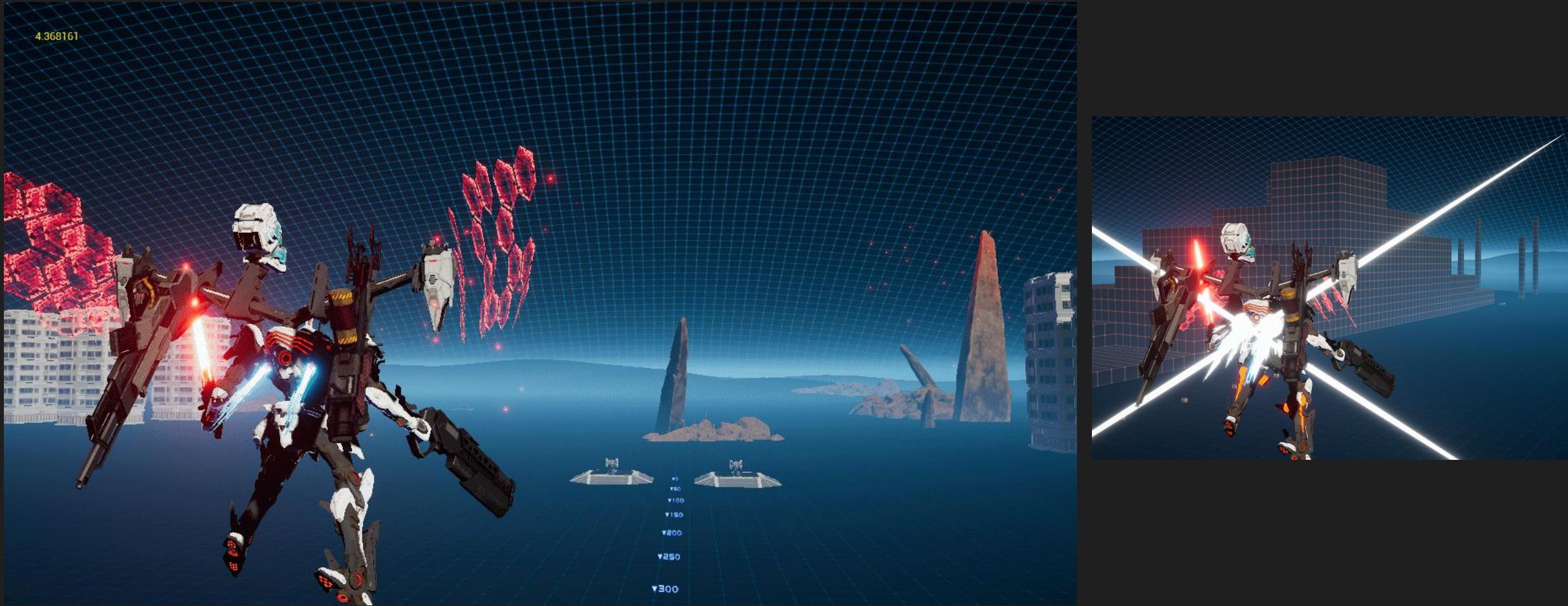
e.範囲・ボスシールド



f.スラスター・スリット



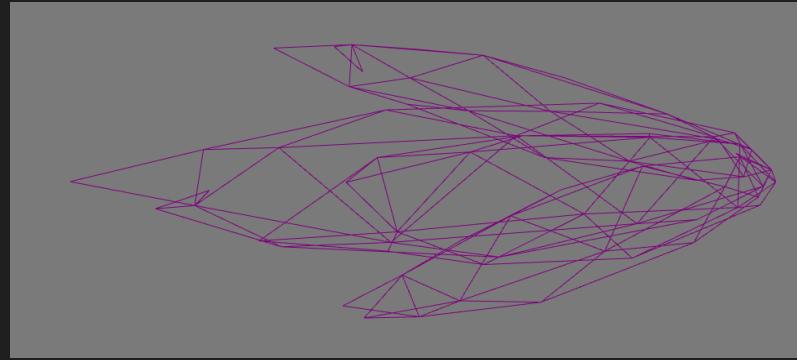
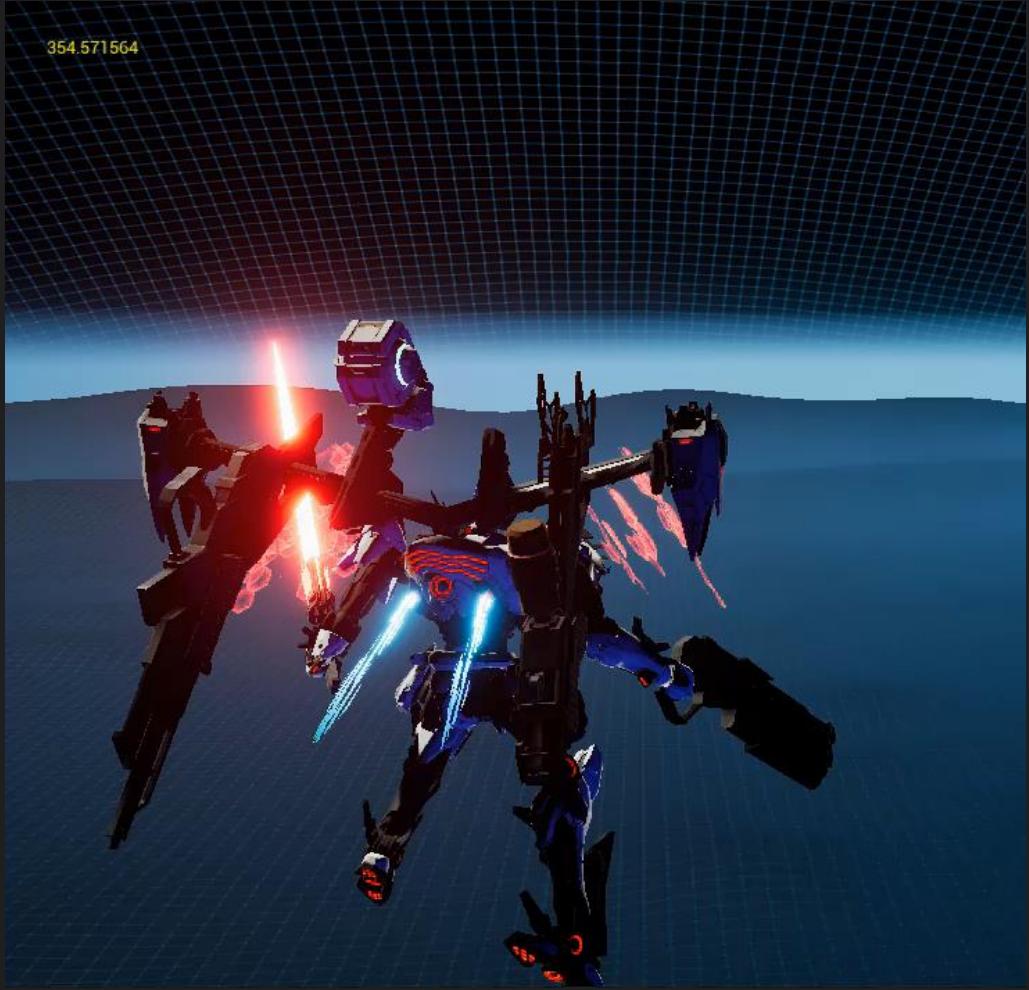
f.スラスター・ブースト



UNREAL ENGINE

#ue4fest

f.スラスター・ボリューム



UNREAL ENGINE

#ue4fest

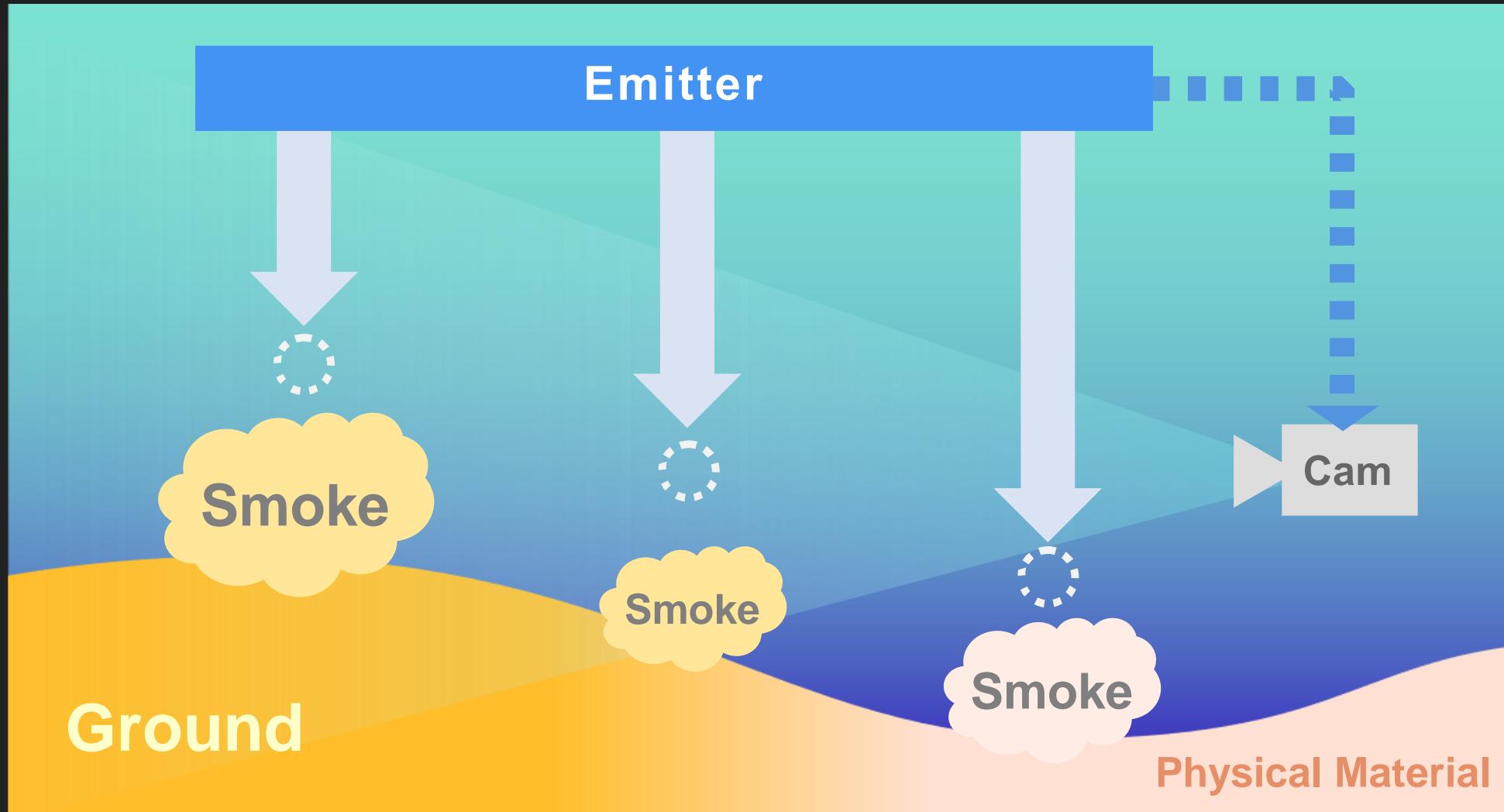
g. 地表煙・背景VFX



UNREAL ENGINE

#ue4fest

g. 地表煙



g. 地表煙



UNREAL ENGINE

#ue4fest

もくじ

1. はじめに
2. VFX制作方針・環境
3. アセット紹介
4. 直面した問題
5. まとめ



UNREAL ENGINE

#ue4fest

4.直面した問題



UNREAL ENGINE

#ue4fest

物理マテリアル仕組化問題

物理マテリアルとは？

- 「土の地面」で着地 → 「土煙」
- 「砂の地面」で着地 → 「砂煙」

地面の種類によってVFXやSFXを変更する仕組み



物理マテリアル仕組化問題

仕組み化にあたって

- チームとして物理マテリアルの仕組み化の方法がわからず手探り
- 影響範囲が広く、仕組み化までに時間を要した
- 仕組化後、既存データの修正対応が必要となった



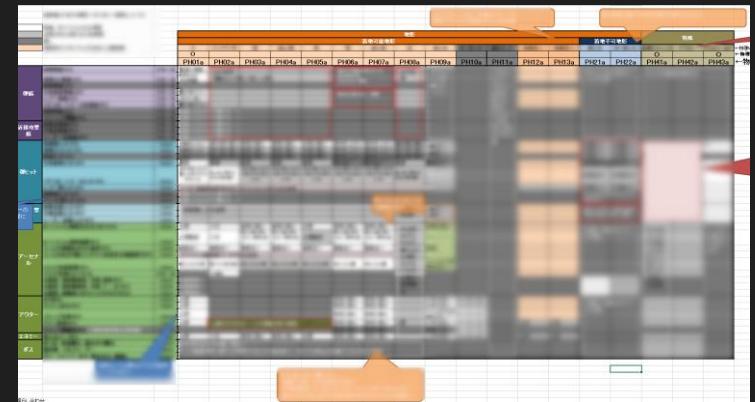
UNREAL ENGINE

#ue4fest

物理マテリアル仕組化問題

仕組み化・条件 4つ

- ① 地形仕様（全種）
- ② 背景班の地形構成（全種）
- ③ 適用対象VFXの洗い出し・出し分け指定
- ④ 物理マテリアルID決定



物理マテリアル仕組化問題

仕組み化・流れ

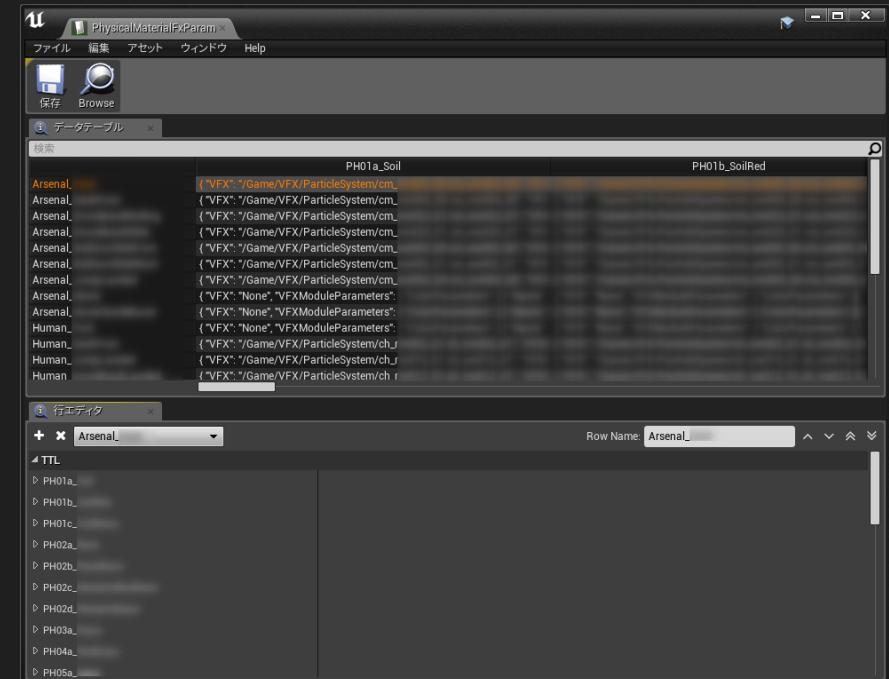
① VFXの仕組み化

initial Colorモジュール Pram化・他

② 背景班対応

③ エンジニア側の物理マテリアル対応

アーセナル（メカ）、アウター（人）、ザコ、ボス、イベント、ギミック



UNREAL ENGINE

#ue4fest

物理マテリアル仕組化問題

物理マテリアル・確認マップ

- 背景班に用意してもらった
- ここでVFXを作成
- 物理マテリアル挙動確認



物理マテリアル仕組化問題



UNREAL ENGINE

#ue4fest

物理マテリアル仕組化問題

仕組み化・まとめ

影響範囲が広いので
早期の各セクションとの相談、仕組み化が重要



UNREAL ENGINE

#ue4fest

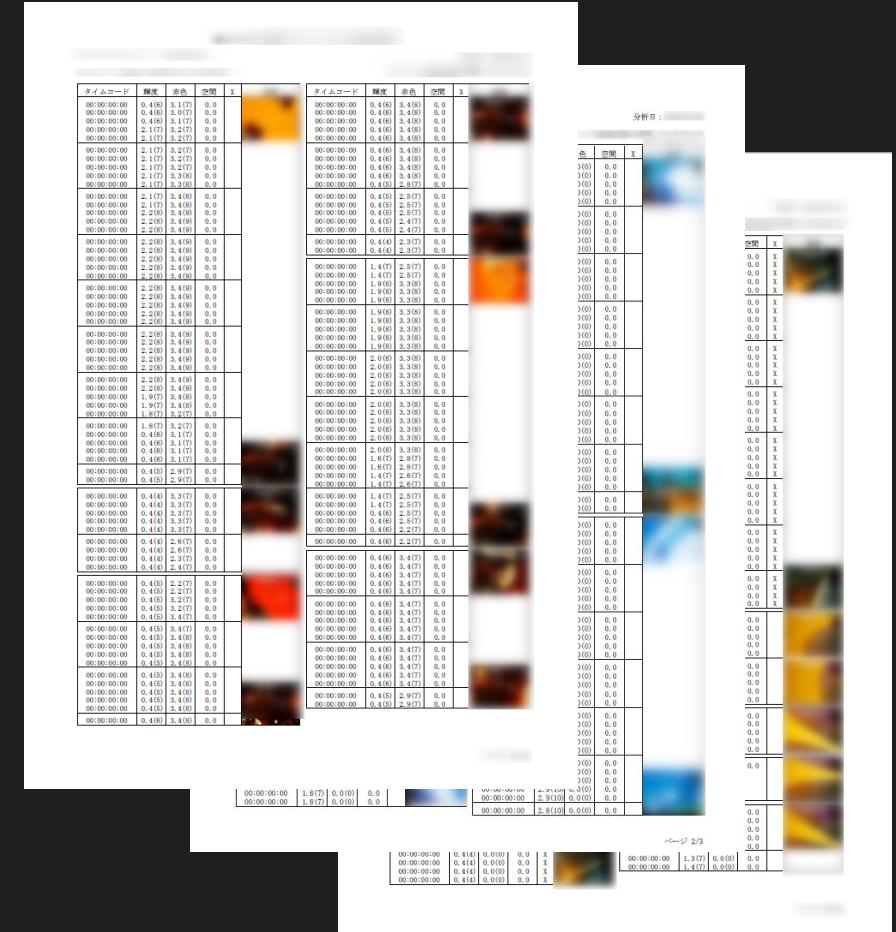
ハーディングチェック対応問題

ハーディングチェック

チェック後の対応

- 手作業

- 仕組み化



UNREAL ENGINE

#ue4fest

ハーディングチェック対応問題

仕組みなし



仕組みあり



VFX班側から他セクションへの提案

- 当初、同じような兵装でも、エネミーごとに別の仕様として発注されていた
 - エネミーごとに兵装が違う
- 美術設定の段階でも、武器ごとの銃の口径や、スラスターのサイズなどがバラバラに描かれており、すべてを作ることは効率が悪い



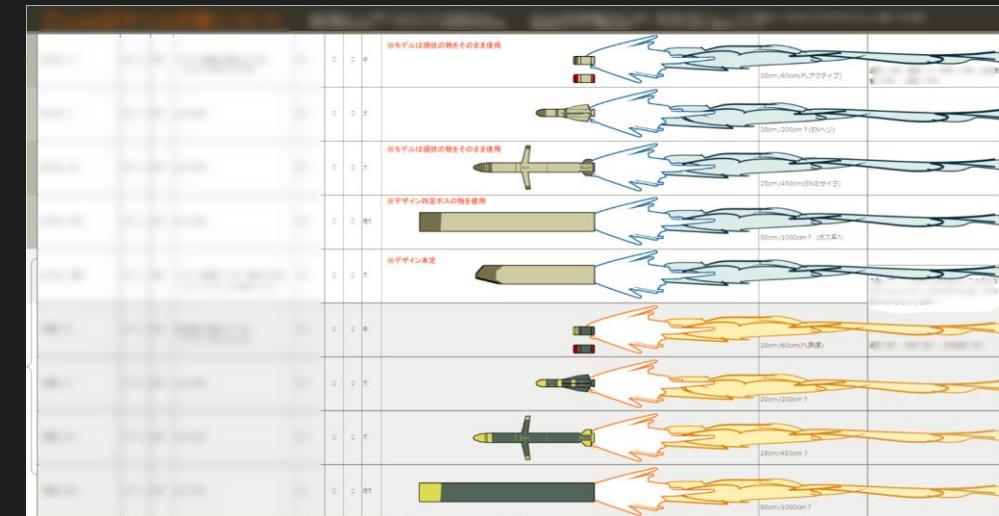
UNREAL ENGINE

#ue4fest

VFX班側から他セクションへの提案

武器の規格化

- 担当プランナーや美術設定担当に、武器種、サイズ、形状の規格化を提案
 - その規格の中で、仕様や設定をしてもらうよう依頼



VFX班以外だと気が付きにくい 積極的に働きかける



もくじ

1. はじめに
2. VFX制作方針・環境
3. アセット紹介
4. 直面した問題
5. まとめ



UNREAL ENGINE

#ue4fest

5.まとめ



UNREAL ENGINE

#ue4fest

VFX編・まとめ

方針 環境

- ・ 方向性をしっかり定める
- ・ 状況に合わせて、事前に準備を整える

アセット作成のポイント

- ・ 機能性（視認性、記号化、インゲームでの役割を確実に担う）
- ・ 処理負荷
- ・ 表現・独自性

問題対応

- ・ 物理マテリアルは早期に着手
- ・ 提案、仕組み化が重要



UNREAL ENGINE

#ue4fest

もくじ

1. はじめに
2. VFX制作方針・環境
3. アセット紹介
4. 直面した問題
5. まとめ



UNREAL ENGINE

#ue4fest

この講演が皆さんのお役に立てたら幸いです



UNREAL ENGINE

#ue4fest

ご清聴いただき誠にありがとうございました



UNREAL ENGINE

#ue4fest