



エンジニアリング・ビジュアルアーツ



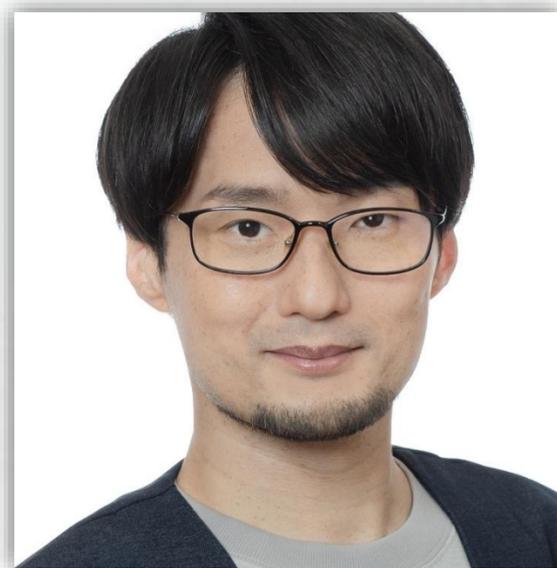
『GRANBLUE FANTASY: Relink』

最高の「没入感」を実現するカットシーン制作手法とそれを支える技術

株式会社Cygames コンシューマー シニアエンジニア / 中村 大吾

本講演は写真撮影およびSNS投稿可能です





コンシューマー / シニアエンジニア

中村 大吾

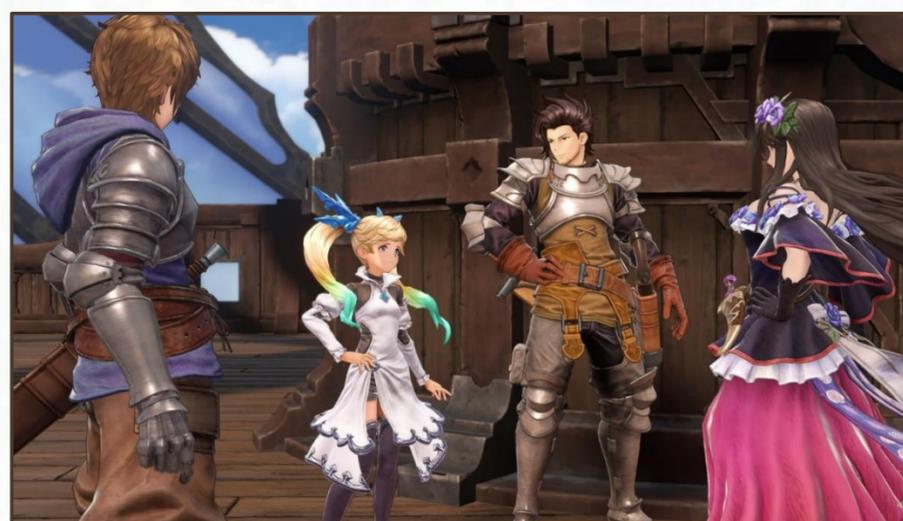
2020年にCygamesへ合流
アニメーションパート、サウンドパートのエンジニア統括を担当しながら
シネマティクスの全体統括を兼任
カットシーンツールの設計から全体進行まで幅広く携わる

用語について



シネマティクス

プレイアブル



カットシーン

会話シーン

ゲームの操作シーン

ユーザーが操作できない ←

→ ユーザーが操作できる

- ◆ シネマティクスが抱える問題
- ◆ シネマティクスを支える大切なこと
- ◆ 没入感を実現した3つの技術
- ◆ 映像美を支えたワークフロー
- ◆ まとめ



シネマティクスが抱える問題

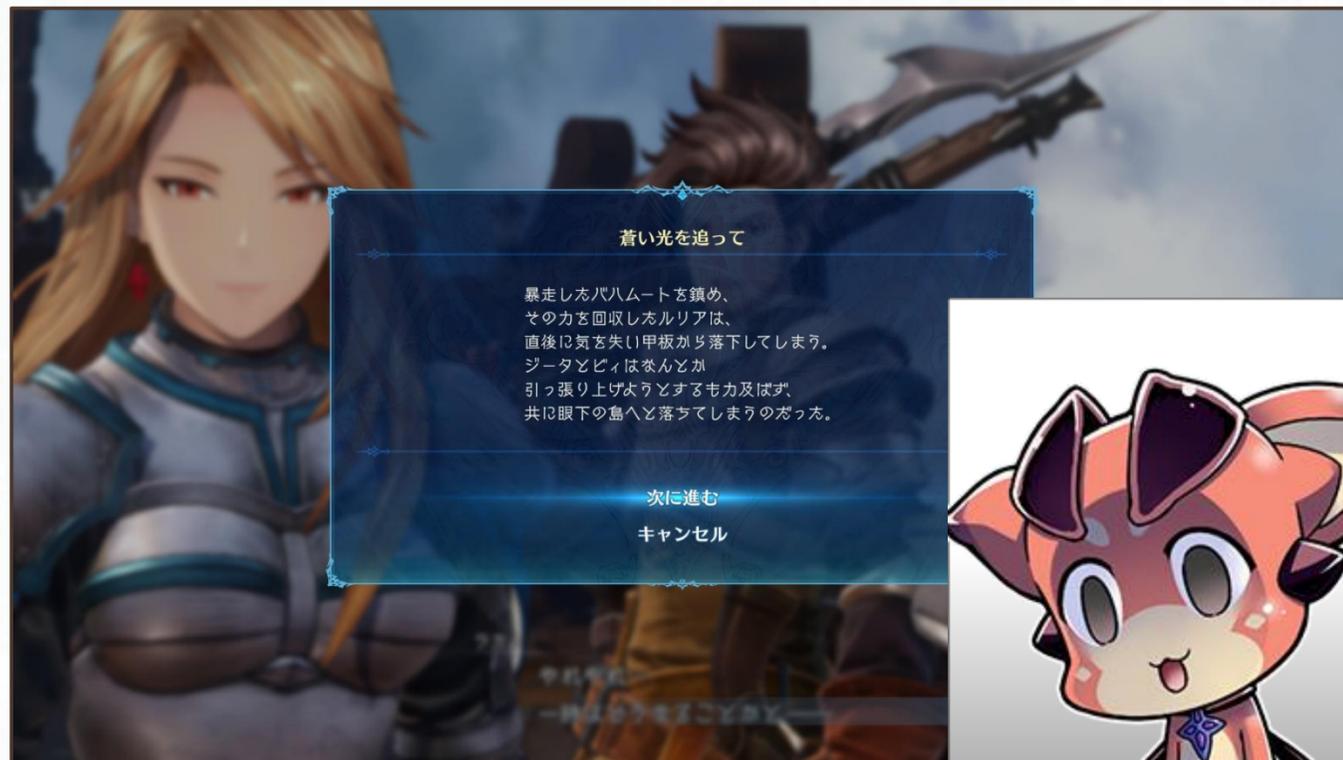


シネマティクスが抱える問題

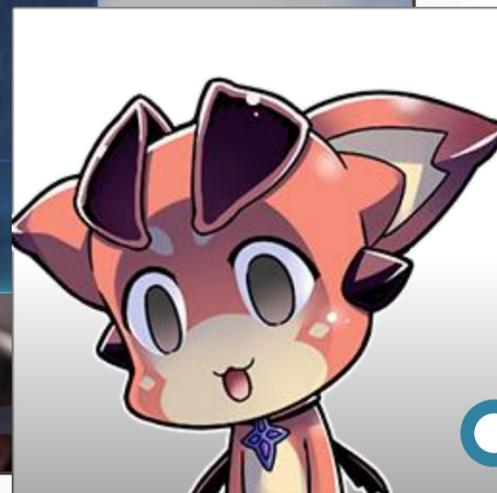


操作ができない

ユーザーが映像を見るだけに



ストーリーを伝える事を重視してしまうあまり
長い演出になりがち



ユーザー

今はゲームで遊びたいから
ここはスキップするぜ

シネマティクスが抱える問題



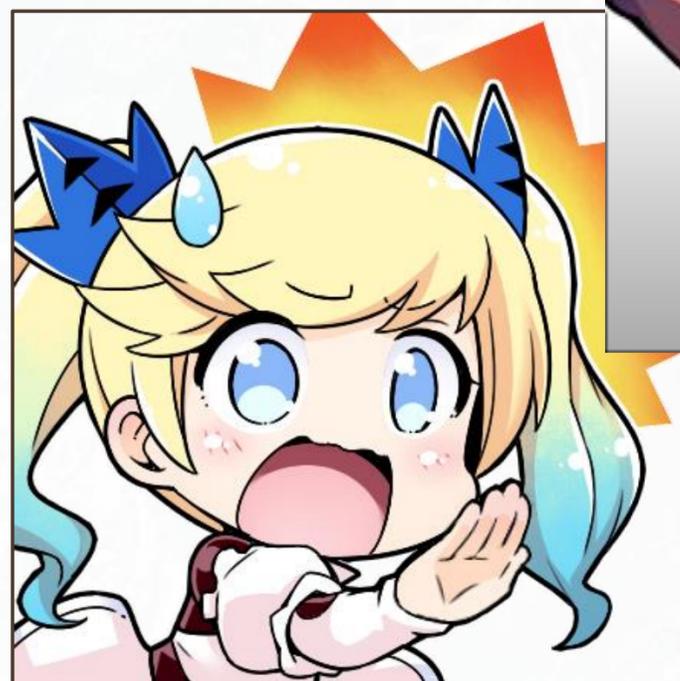
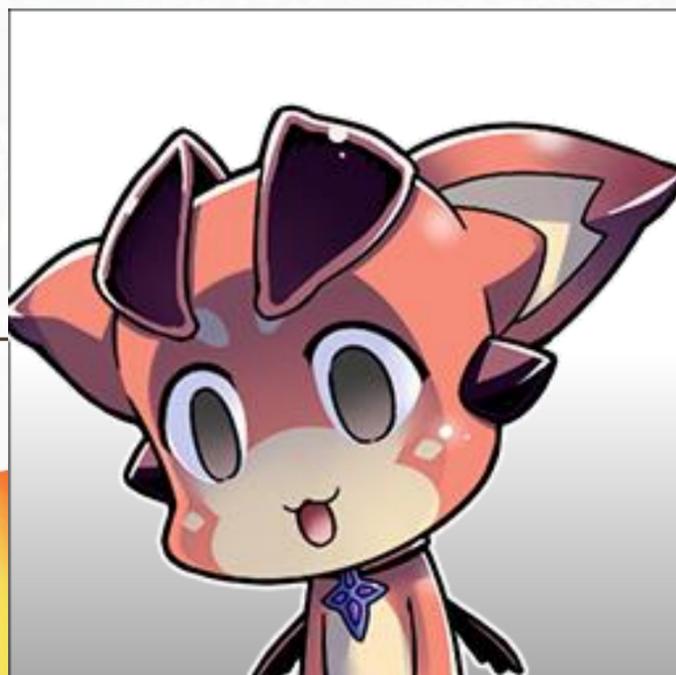
課題

なぜスキップしてしまうのか

シネマティクスは…

- ストーリーを盛り上げる要素
- バトルへのモチベーションを最大限に引き上げる演出 だが…

かみ合わないと
一切見られない可能性も



シネマティクスを支える大切なこと



シネマティクスを支える大切なこと



テーマ

没入感

映像美



課題

没入感を損なう要素とは



プレイアブルとシネマティクスが乖離

- 急な映像への切り替え
- 遊んでいるキャラクターの振る舞いが急に別人のようになる

区切り感が出てしまう

- フェードアウトによる暗転

課題

映像美を生み出す項目



映像のすごさは
アーティストがどれだけ
こだわられる時間を作り出すか

- イテレーションを回すスピード
- ツールの利便性
- 作業の**効率化**
- 盛り込まれた要素を表現しきれ**るパフォーマンス**

没入感を実現した3つの技術



没入感を実現した3つの技術



- ① リアルタイムレンダリング
- ② セカンダリフィードバック
- ③ シームレス遷移

①リアルタイムレンダリング

リアルタイムレンダリング



疑問 動画との違いは？

実は…

動画よりも
シネマティックの没入感◎

本作では全てのシネマティックを
リアルタイムレンダリングで実現!!



リアルタイムレンダリング



高フレームレート
対応

プレイアブル中は120fps、シネマティクスでも
同一のフレームレートを利用して違和感が少ない

高解像度への対応

プレイアブル中と全く変わらない解像度◎
(※動画では非常に難しい)

武器の反映

本作では様々な武器を装備し、見た目が変わる
プレイアブル中の武器がそのままカットシーンで
利用可能に

リアルタイムレンダリング



課題

実現にあたっての前提条件



シネマティクス中では
映像に特化して表現する必要

いくつかプレイアブル中には可能だったパフォーマンス用の施策は行わない形で取り組むことに

リアルタイムレンダリング



シネマティクス中の制限として

Dynamic Resolutionによる
解像度の低下は不許可

目に見えるポッピング
(LOD、Texture Streaming)の不許可

パフォーマンスにおいて非常に効果的な影響を与える上記は
プレイアブル中ではある程度動きがあるために目立たない項目だが
映像に注視するシネマティクスでは、目立つ項目として除外

リアルタイムレンダリング



その上で実現した方法について解説

オクルージョンカリングの
最大限活用

LODの事前リクエスト

可変フレーム対応

最適化のチェック体制

リアルタイムレンダリング



方法① オクルージョンカリング



- シネマティクスはカメラが固定化されている
- 最適化のポイント：**無駄な表示を省く**

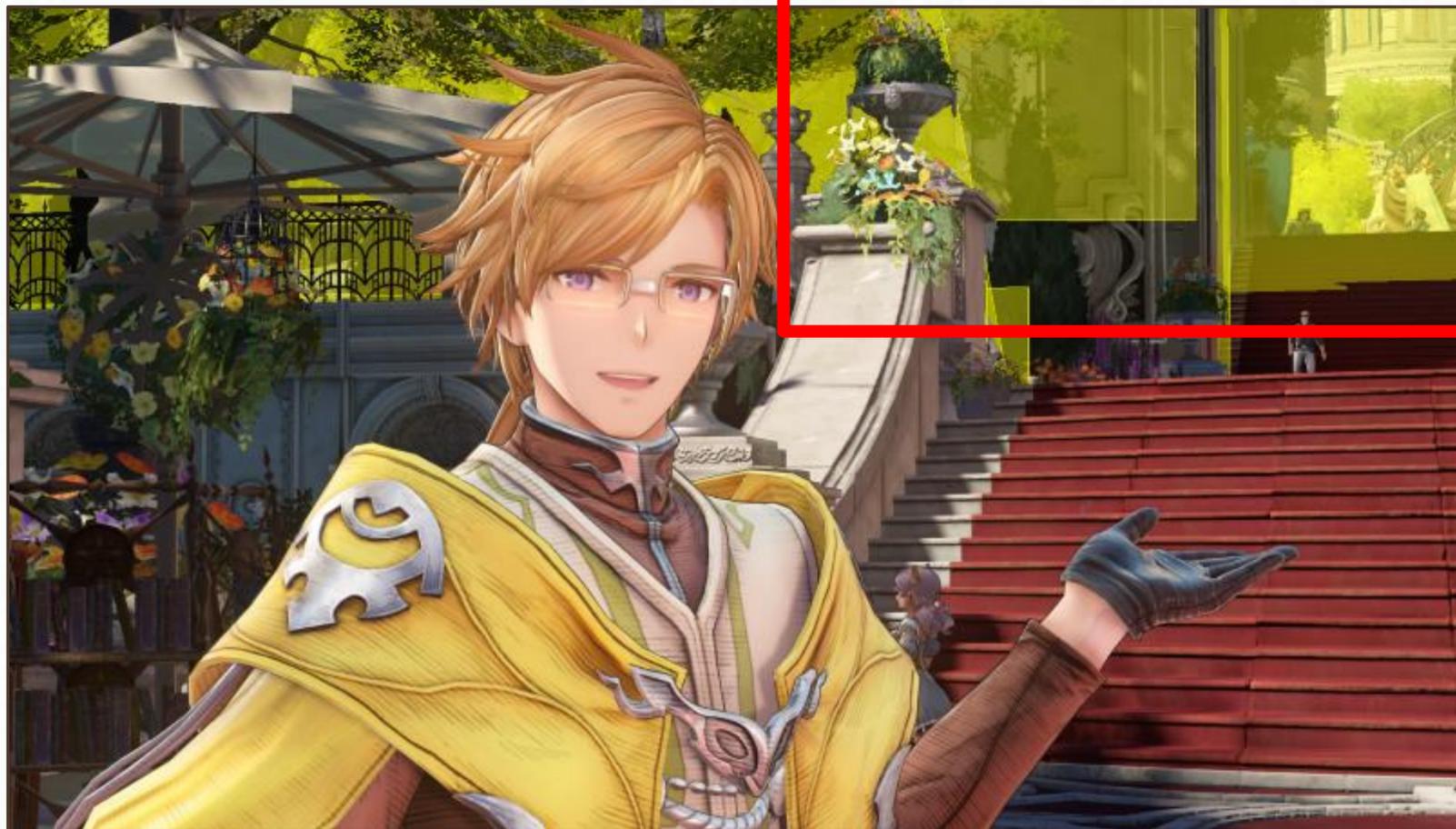
特にパフォーマンスを出すことが困難な場所で多用している方法として…

シネマティクス専用 オクルーダーの配置

リアルタイムレンダリング



方法① オクルーダーの配置例

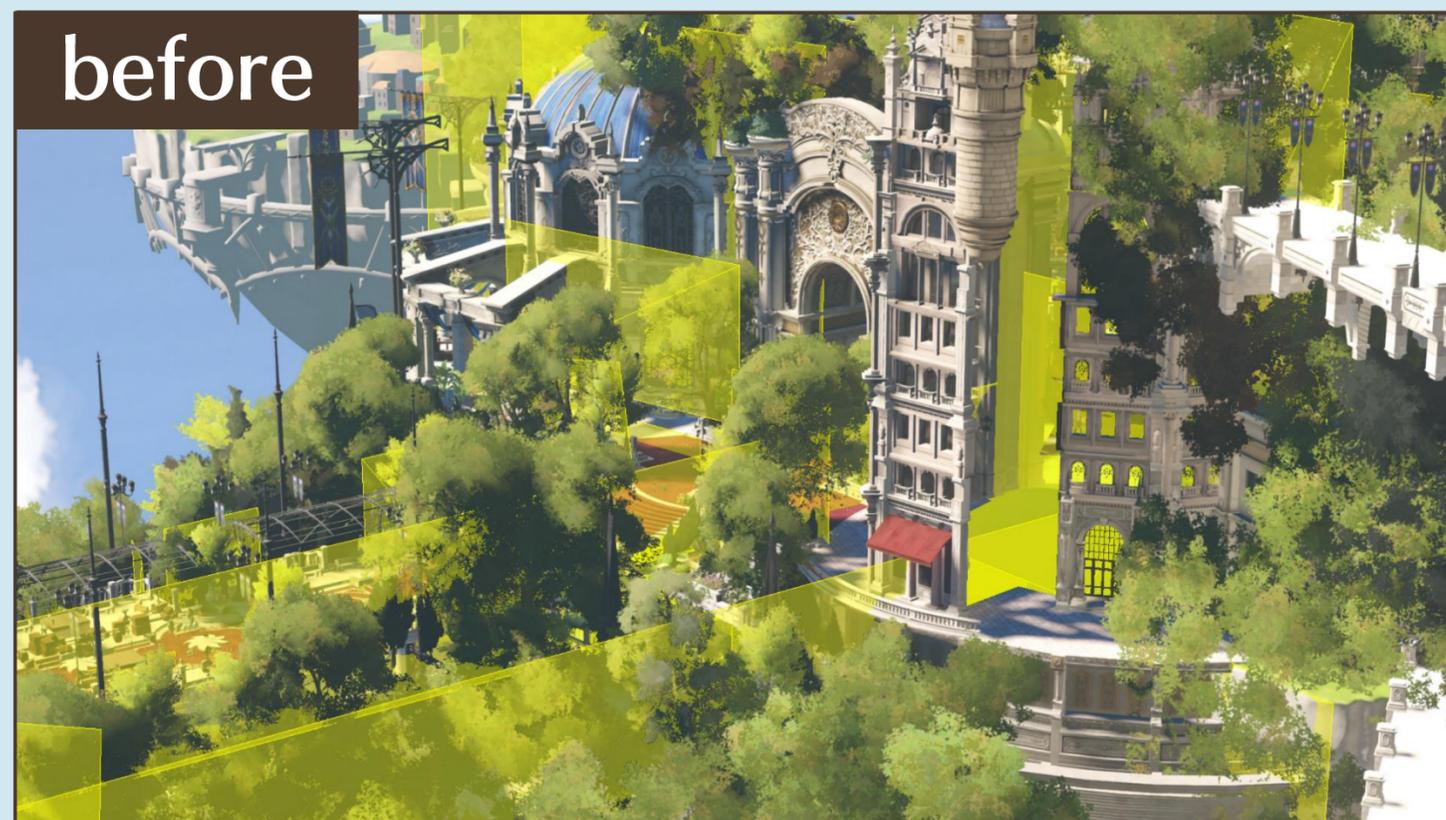


黄色い部分に
カットシーン専用で配置
場面に応じて個別に対応

リアルタイムレンダリング



方法① オクルーダーの配置例

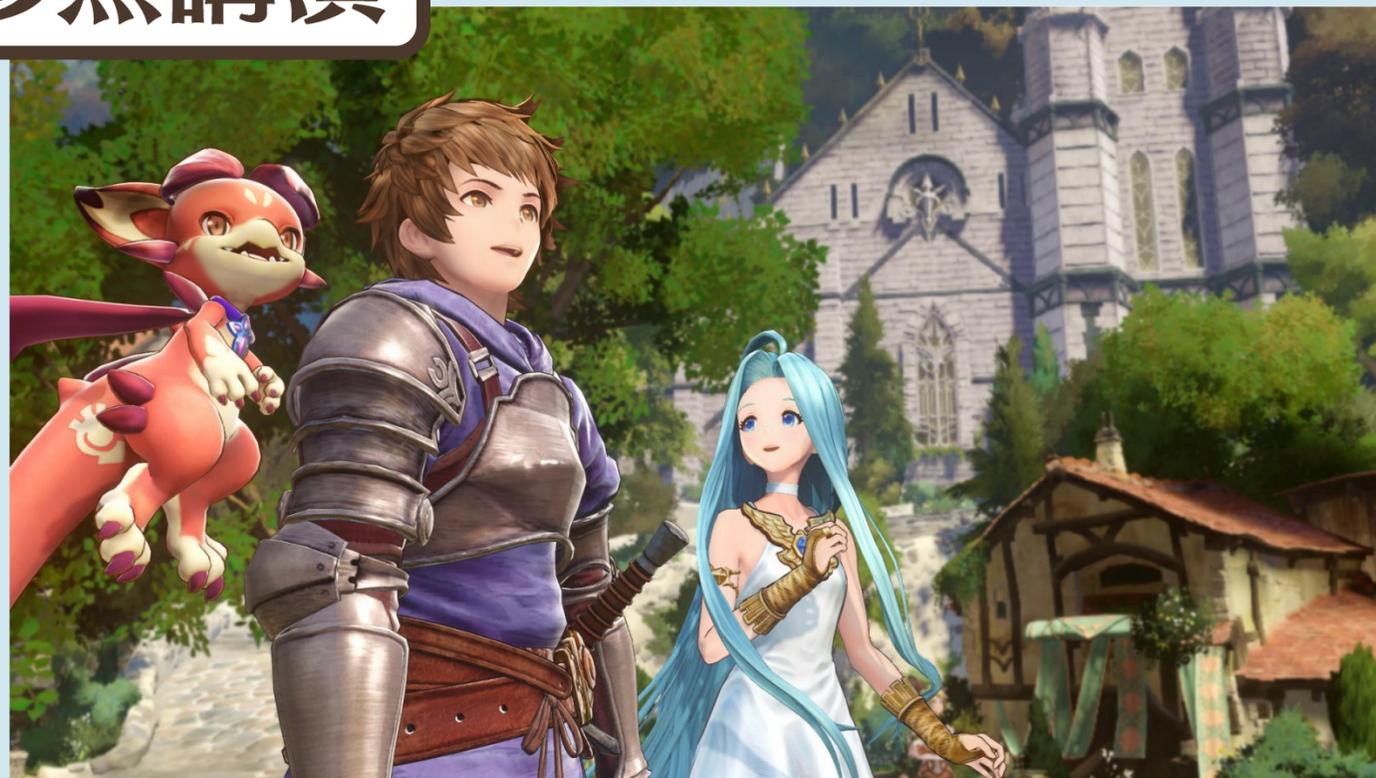


リアルタイムレンダリングを実現するうえでは必須の対応

リアルタイムレンダリング



参照講演



『GRANBLUE FANTASY: Relink』
ソフトウェアラスタライザによる実践的なオクルージョンカリング

方法② ポッピングへの対応



- カメラが一瞬で切り替わるためポッピングへの対応は不可避
- LODの変化やTexture Streamは、表示次第ロードが始まり高いLODを表示するため切り替わった瞬間が見えてしまう

タイムラインを
全て検索

オフラインで
リクエスト
リストを作成

先行リクエスト

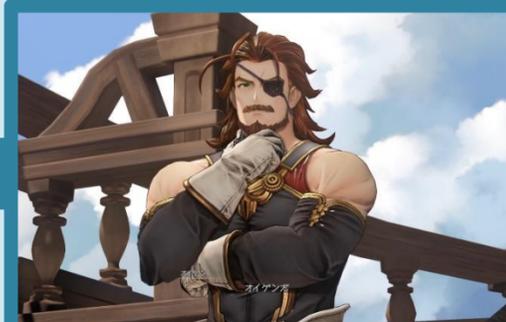
リアルタイムレンダリング



方法② LODモデル、Textureリクエスト

基本的には全ての素体はタイムラインでどこで表示されるかわかるため
そのリストを作成し、デフォルトではどちらも自動で2秒前にリクエストを
することで軽減。どうしても間に合わないケースは、秒数を個別に調整する対応

2秒前にリクエスト



リアルタイムレンダリング



方法③ 可変フレームへの対応



本作ではPS4とPS5、Steamと異なるプラットフォームへ対応

それぞれパフォーマンスは違うためリアルタイムレンダリングをする上で可変フレームへの対応も必須

同じ尺の
カットシーンとサウンド

常に音と映像の同期

リアルタイムレンダリング



方法③ サウンドとの同期

毎フレーム**サウンド**の進行秒数をカットシーンタイムラインへ受け渡し
カットシーン側の映像を進行させる事で、音のズレが起きにくい構成と
しつつも環境に依存しない可変フレームへの対応手法としています

カットシーン



同期

同期

同期

同期

同期

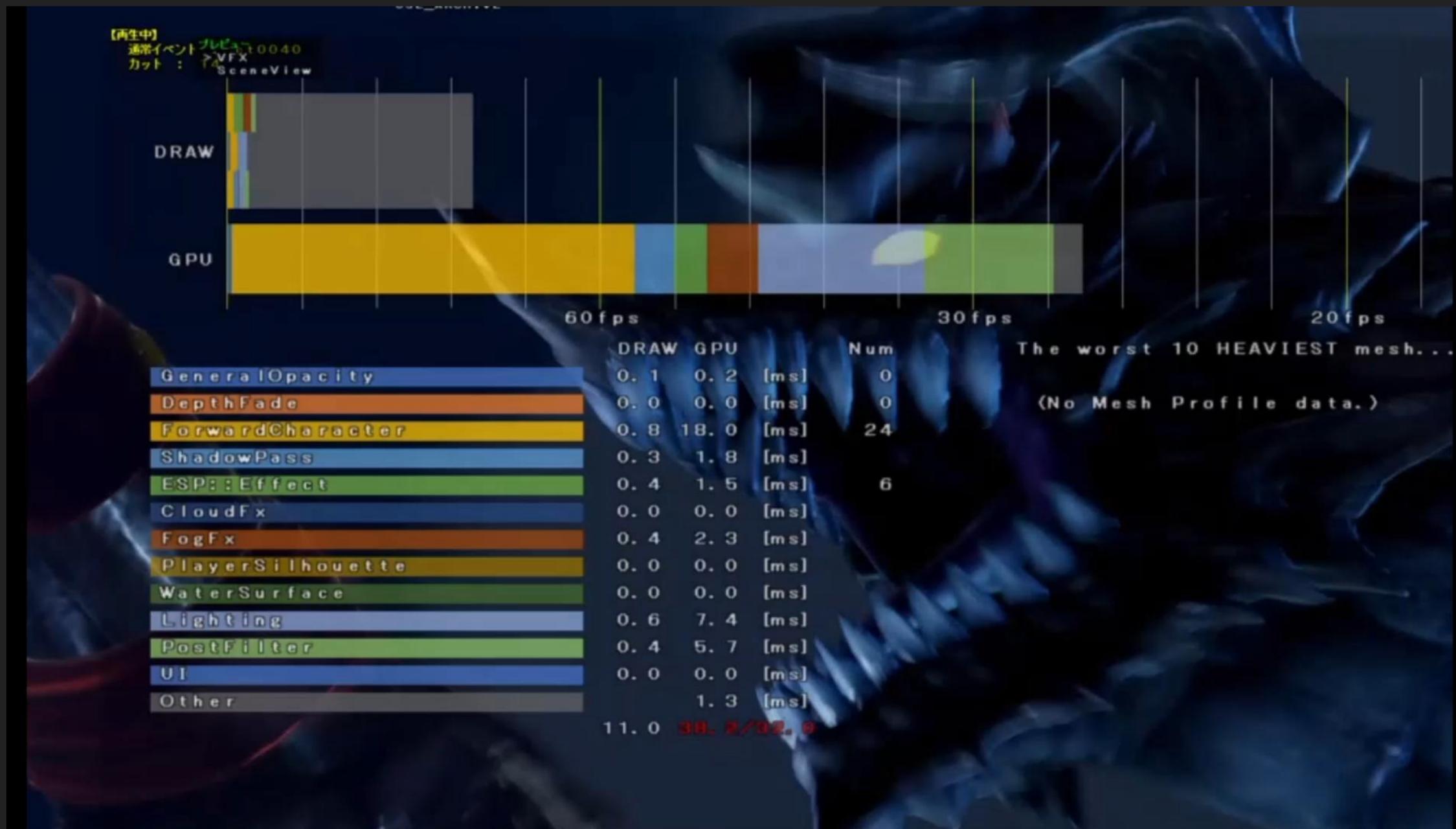
サウンド

方法④ パフォーマンスチェック体制



最後にご紹介するのは、シネマティクスのパフォーマンス計測環境です
リアルタイムレンダリングを実現するには
最適化は必須の項目であり
計測環境を充実化させるのは非常に重要である
と言えます

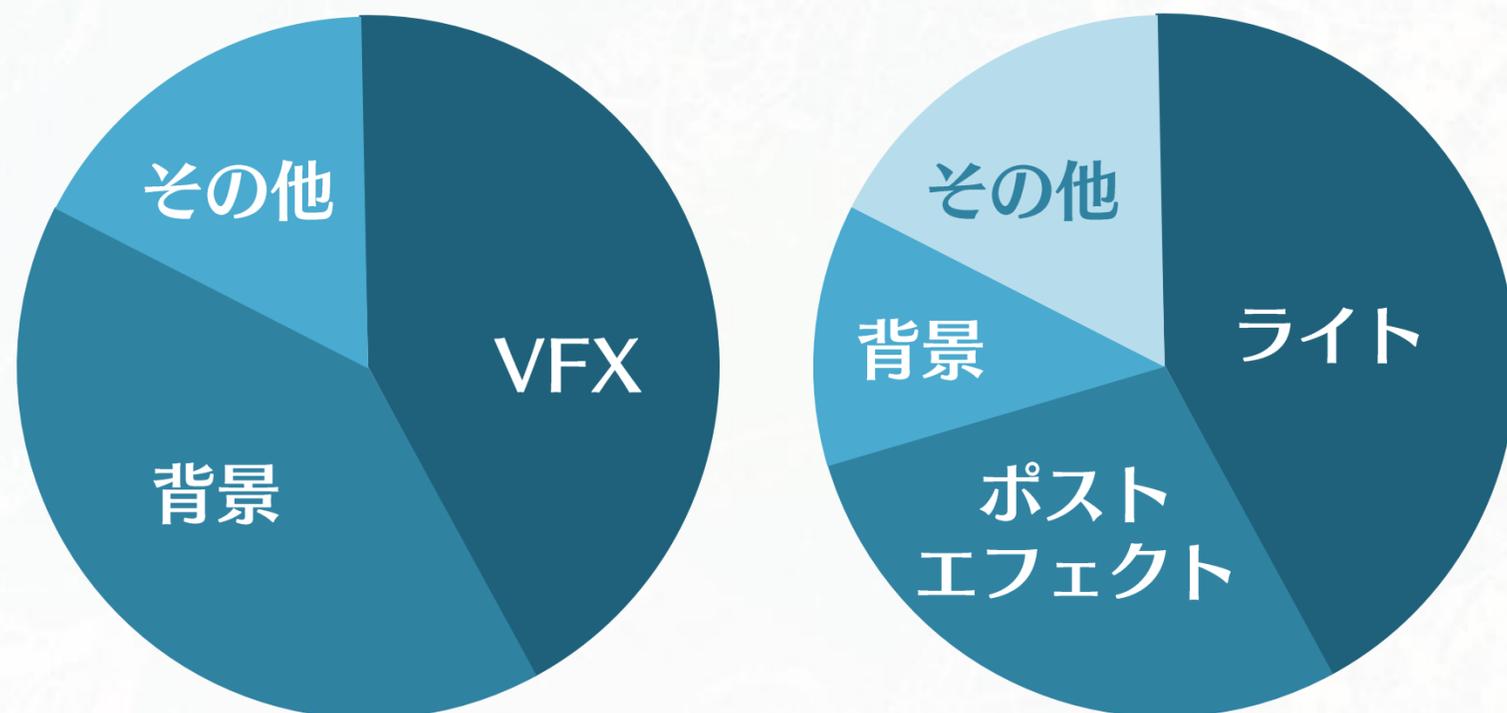
リアルタイムレンダリング



リアルタイム計測例

- 各シーンの計測一覧
- 最適化すべき場所を把握
- どのセクションと相談の必要があるかを可視化

方法④ ボトルネックを明確化



ボトルネック要因の違い

ケースによってネックになっている場所は千差万別
効果の薄い最適化は避ける必要がある

差を可視化することで
問題を的確に調査→最適化

リアルタイムレンダリング



方法④ オートプレイでの自動計測



毎日オートプレイを走らせ、全てのシーンでパフォーマンスを計測し
Web上で遷移を可視化

方法④ オートプレイチェック



デイリーで全てのシーンを自動再生
→何か異常があればすぐに気付ける体制に
リアルタイムレンダリングである事は
シネマティクス以外の**環境変化による影響も**
受けやすい

- 背景の仕様が意図せず変わった
- 異常のあるオブジェクトが紛れている など

計測・チェック体制は必要不可欠

リアルタイムレンダリング



こうして…リアルタイムレンダリングを実現



全てのシーンで制限を
クリア

プレイアブル中と同じ状態

→シネマティクスへの没入感を高める事に成功

リアルタイムレンダリングは
この後の項目すべてに関係
要素としては**非常に重要な項目**となります

②セカンドリフトバック

セカンドリフィードバック



ポイント アニメーション作成ワークフロー

キャラクターモデルは、プレイアブル中と同じものを利用
通常衣服などの揺れもののアニメーションはDCCツールなどで
カットシーン専用として作成されることが多いが…

本作のワークフロー

- 本作ではプライマリのみを再生
- シミュレーションさせてからエクスポート
- 「セカンドリ」アニメーションとして再度入れ込む

セカンドリフィードバック



本講演では…

プライマリ

ボディアニメーションのみ

セカンドリ

揺れものなどのシミュレーションで
動きもついたアニメーション

セカンダリフィードバック



フローイメージ

DCCツール



プライマリ 制作

プライマリ のみを再生した実機シミュレーション



アニメーション を セカンダリ にエクスポート

実機作業



DCCツール

セカンダリ セカンダリモーション

セカンダリフィードバック



Export Tool



プライマリのみでカットシーンを流し
シミュレーション込みでジョイント情報の
エクスポートを行うツール

shot単位、キャラ単位で個別に出力され
セカンダリだけではなくIK情報や
自動Lipsyncを行ったアニメーションも
出力する設計がされています



セカンダリフィードバック



高速なイテレーション

映像として気に入ったシミュレーションになるまでショット単位で何度も繰り返し風や向きを微調整しながらの繰り返しが可能に
データとしてはプライマリも含んだトータルのジョイント情報を出力



セカンダリリフュードバック



Merge Tool

セカンダリ
データ

プライマリ
データ

プライマリのアニメーションと出力したセカンダリデータをマージするツール



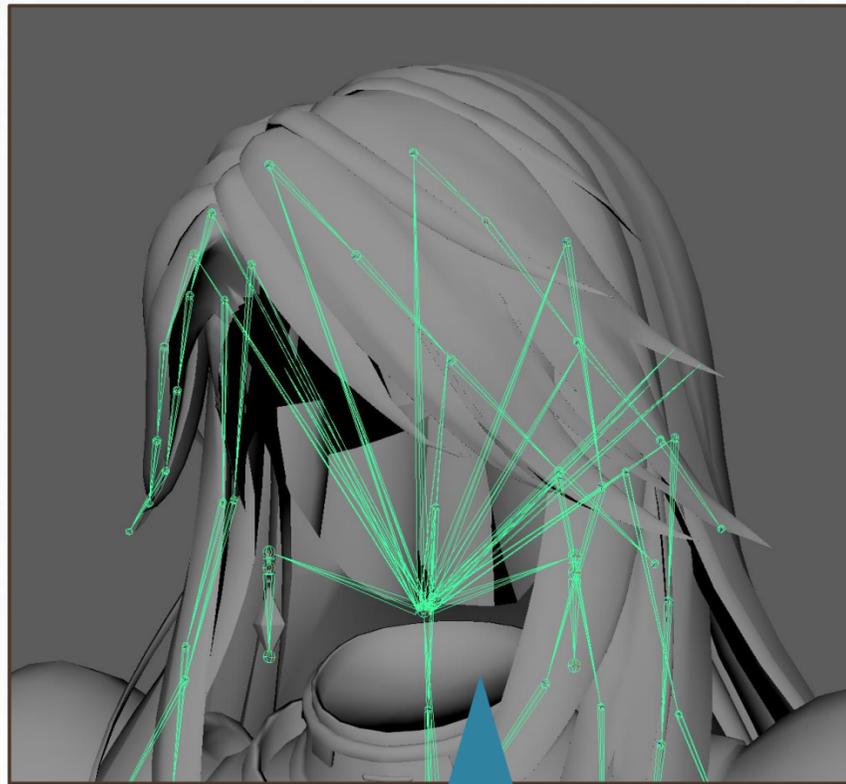
最終データ

プライマリに変更があっても過去のセカンダリをマージできるようにしています

セカンダリリフィードバック



手法について



髪の毛は頭のジョイントに
再追従させるだけでOK

本作で選択した手法は

- セカンダリのジョイントとプライマリのジョイントを明確に分け
- セカンダリジョイントの上のプライマリジョイントの動きのみをセカンダリと再連結する

このマージは実機を介さずモーションデータのみでできるため**変更後も高速に確認が可能**



その1 特徴的なキャラクター性

- ・特徴的な衣装
 - ・衣装の揺れにもキャラクターの個性
- ゲーム内で**独自のシミュレーション**作成

プレイアブル中と同じ印象を持たせる
揺れ方も**カットシーン**で再現する必要



セカンドリフィードバック



参照講演



『GRANBLUE FANTASY: Relink』
キャラクターの魅力を支えるリグ-シミュレーション制作事例

セカンダリリフィードバック



その2 工数の削減

シネマティクスでは…

工数が多く
時間もかかりがち

プライマリ・セカンダリの
完成が急務

- ボディアニメーションのみでシミュレーションシーンの概要が分かる
- 変更にも強い設計
- アニメーションのみを見ても大幅に工数が削減

こちらは大きくワークフローにも関連、後ほどの項目で詳しく紹介

③シームレス遷移

シームレス遷移



シームレス遷移とは



シネマティクスとプレイアブル中を
継ぎ目なく再生させる技術

演出中とゲーム中の隙間にフェードアウトを挟む
と区切られた感が出てしまう

プレイアブル中との間に**専用の遷移シーン**を挟む
事で映像への**没入感**そのままに**バトルへ**とつなぐ
事が可能に、ボス戦では**必須の要素**

シームレス遷移



カットシーン



シームレス遷移



プレイアブル



フェードアウトの廃止

カットシーン 全てのパフォーマンスを映像に特化



- 作業者は専任
 - ほぼ全ての要素が専用の物
- 基本的にはプレイアブルの環境を利用
- 追加ライト
 - 専用のポストエフェクト
- などが調整されている

シームレス遷移



シームレス遷移

カットシーンの画のままプレイアブルに



プレイアブル中のアセットを利用して
演技を構築し導入用の専用モーションは
作成されるが

ライトやポストエフェクトなどは
基本的にプレイアブル準拠のため
専用に絵作りする事はできません

この制約を加味しつつ

シームレス遷移シーンは作成されています

シームレス遷移



カットシーン



シームレス遷移



プレイアブル



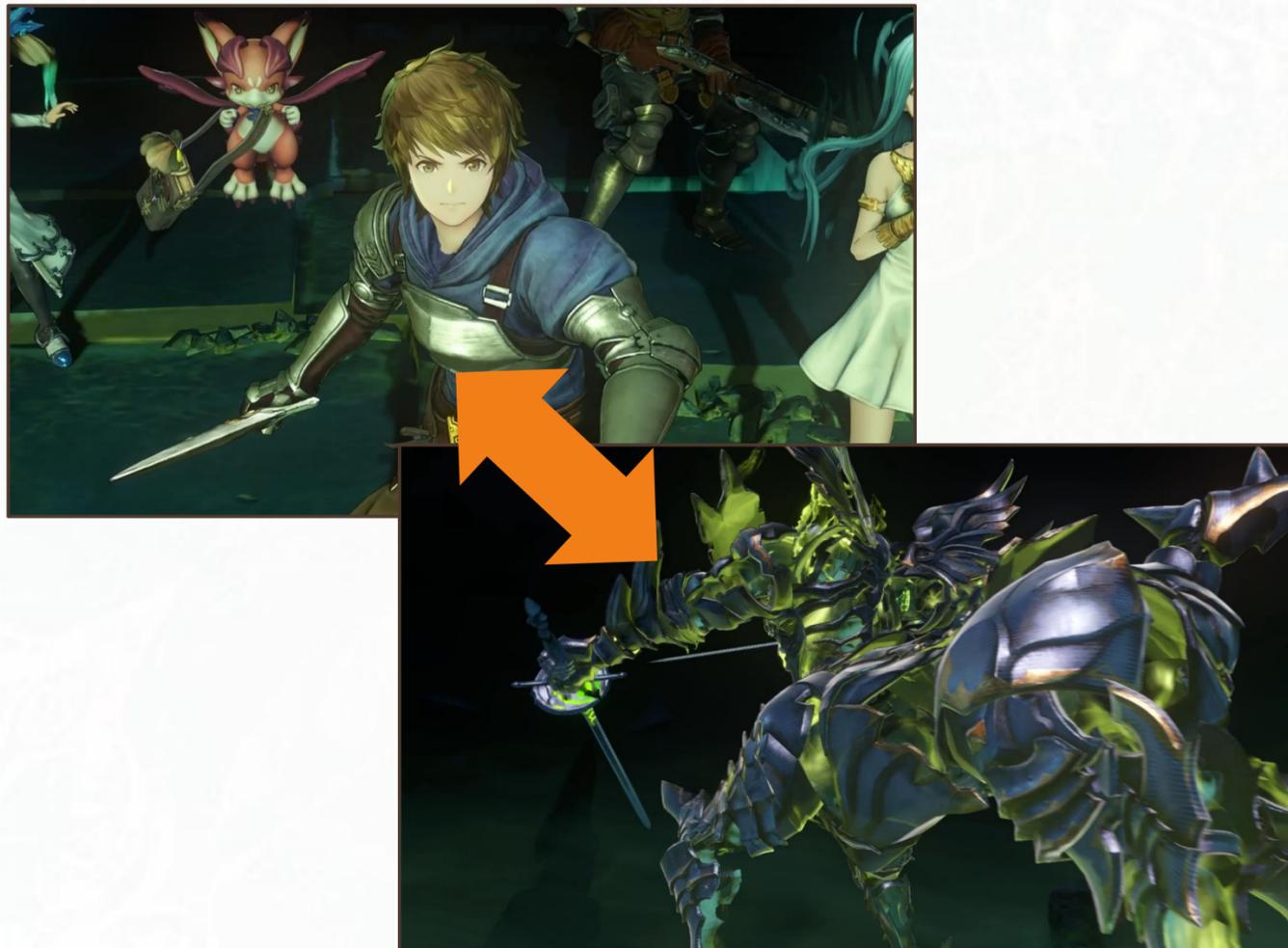
シームレス遷移



カットシーン

シームレス遷移

どちらも同じツールで作成



リソースや作業者は別になっても制作ツールとしてはカットシーンもシームレス遷移中も同じものを利用しデータ構成やロードの仕組みの**共通化**を行っています



シームレス遷移



ただ連続再生するだけなら…

簡単そうに思えますがしかし、実際は考慮しなければいけない事例がいくつかあります



- **LODのポッピング対策**

代表的な項目ですが、当然次の演出のデータのリクエストをしておく必要があります

- **ロードの設計**

長尺にわたる演出データを一度にロードする事はメモリの圧迫とロード時間増加を生むためある程度細かなショットごとにロードをすることができるようにし連続再生される後続のシーンも一連の流れで読めるようにしています

- **削除遅延対策**

当然の項目ですが実は一番時間がかかった部分でした
直前のカットシーンでは多用されているポストエフェクトやライトをシームレス遷移画面になった瞬間では**1フレの遅延もさせず切る必要があります**

シームレス遷移



会話シーンの場合



本作における会話シーンは
専用で作るカットシーンとは大きく違い
ある程度プレイアブル中の要素のまま
演技をするシネマティクスです

ただしカメラの画角や被写界深度の調整
キャラクターのための追加ライトなど
いくつか専用で調整は行われています



面倒見のいい男

傷ついたりした際は
無理せずこの岩のそばで休息をとってくれ

LB AUTO

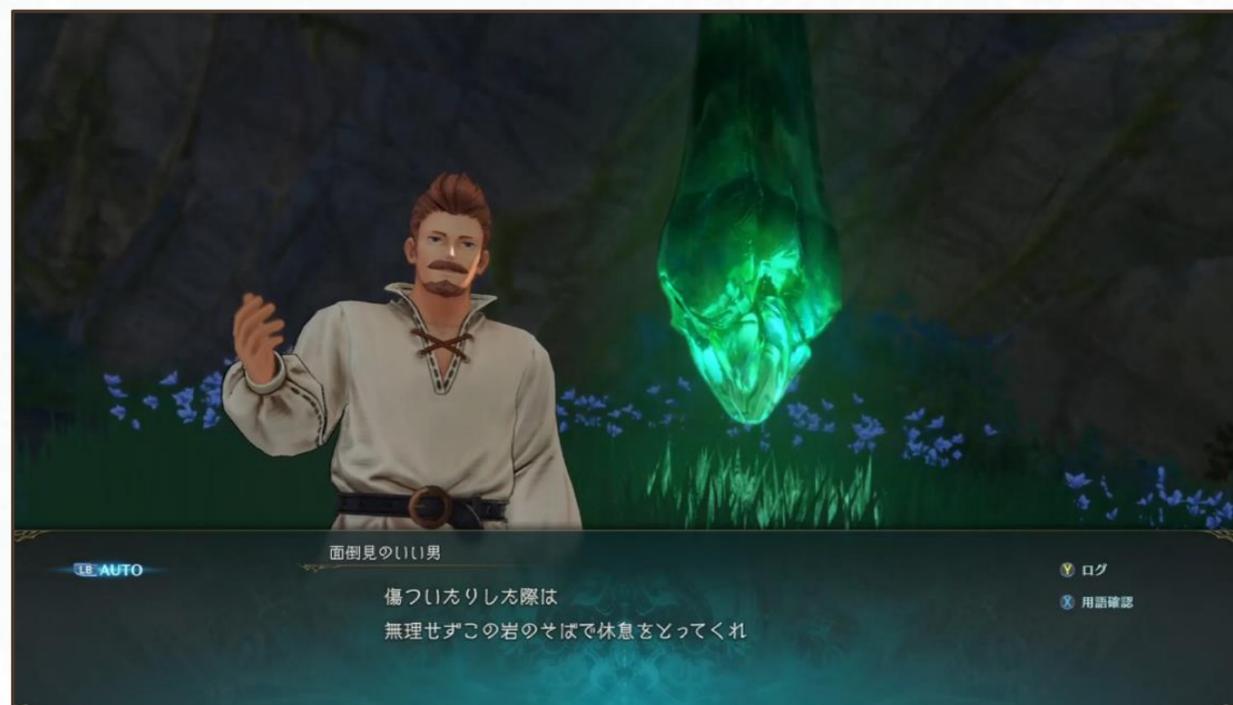
Y ログ

X 用語確認

シームレス遷移



自動シームレス遷移



プレイアブル中へとシームレスにつなげるためにカットシーンとはまた別の手法を取る必要がある

会話シーンでは最終ショットを遷移ショットとし会話中に利用していたライトやポストエフェクトをプレイアブル中の数値へ**自動補間**

また、演技に合わせて調整していたキャラクターの位置も違和感の無いようプレイアブル中へ**自動で接地補間**



ロゼッタ

うふふ……お願いね。

LB AUTO

Y ログ

X 用語確認

シームレス遷移

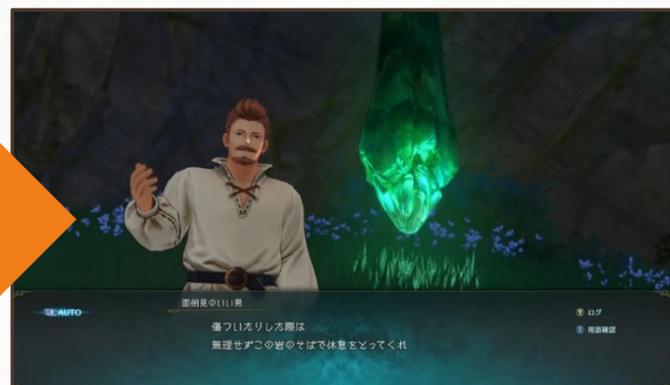


開発環境

全て同じツールで作成



カットシーンと同じツールで作成
カットシーンから会話シーンへ
その逆もまたフェードアウト無しで
隙間なく遷移する事が可能に



シームレス遷移



メリット

シームレス化する事で



ほぼ全てのシーンが違和感の無いよう
カットシーン、会話シーン共にプレイアブル
へとシームレス遷移させています

ただフェードアウトを無くすだけでもこれだけ
の要素が必要になりますが
この手間を惜しまない事で、**没入感を最大限
継続させる**事が可能となっています

映像美を支えたワークフロー



映像美を支えたワークフロー



メリット

前提として



本作は独自に開発されたオリジナルのエンジンを採用。幅広いメンバーが利用できるようにするためには制作ツールを工夫する必要があった

カットシーン制作ツールのインターフェースとして採用したのがUnity

これによって導入の敷居が下がり
ラーニングの工程が短縮

メリット

Unityでできる事



Unity上でTimelineを利用して

- Motionやカメラ割を決定するショットのセッティング
- 追加のライトや実機に設定されているライトの操作
- ポストエフェクトの追加や設定
- 演出中の特定のトリガー設定
- 字幕のタイミングや音の発音設定
- 風の調整などの環境の変更

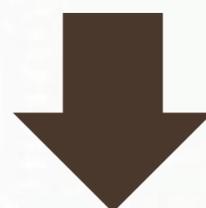
基本的にシネマティクスの組み立ては
Unityのみでアーティスト作業は完結する
ようになっています

映像美を支えたワークフロー

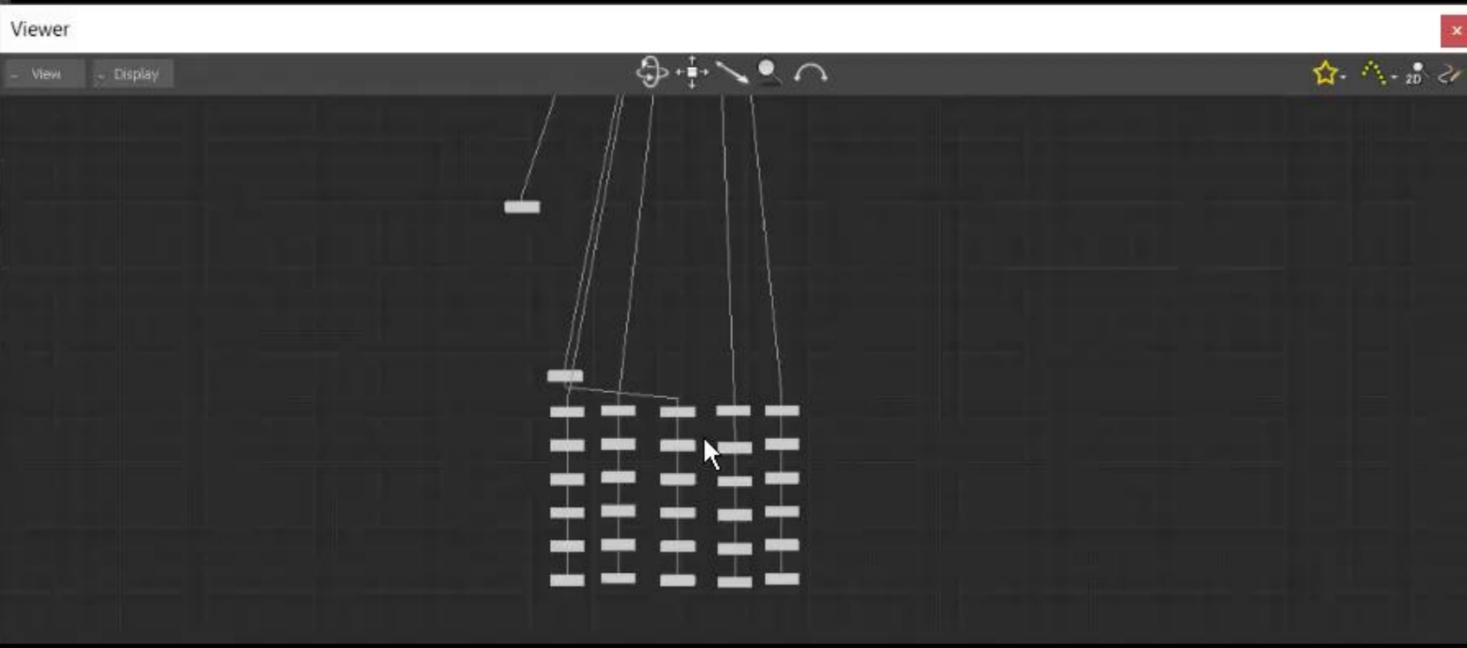
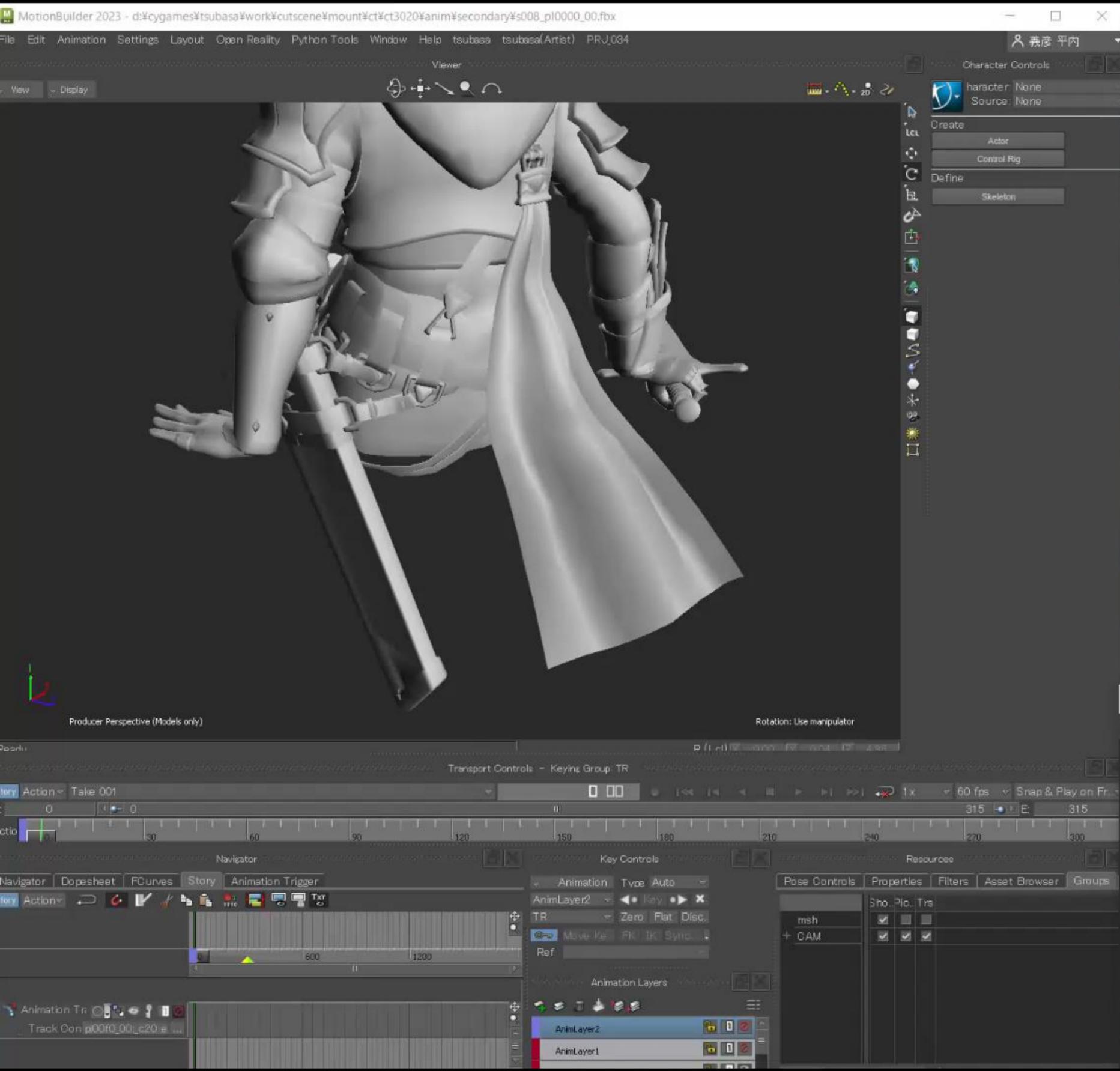


仕組み

実機との連携



ライブリンクという仕組みを
実装して実現
ライブリンクは、DCCツールからも
リンクできるようになっており
実機の状態に依存せず作業ができる
仕組みを確立



ライブリンクとリアルタイムレンダリング



この二つの組み合わせは非常に強力で

- セカンダリ調整
- フェイシャルアニメーション
- ライティング、ポストエフェクト

などの工程では

直接映像を作る感覚で作業が可能

映像美を支えたワークフロー



恩恵

リアルタイムレンダリング技術

イテレーションの高速化

背景やVFX、音声など全ての要素を最新の状態で確認しながら作り出す事ができるように。

リアルタイムで制作から確認が可能

作業時間を
ギリギリまで確保可能

不具合対応が
素早く修正可能

かなり直前まで
映像にこだわる期間

パフォーマンスを
改善する時間



恩恵

セカンダリフィードバック技術



プレイアブル中の揺れものなどに代表されるシミュレーションをカットシーン中でも生かして再生
モーションとカメラのみの段階から
セカンダリも込みで確認する事ができる

そのため、プライマリだけが完成した段階で
映像としての流れや雰囲気を把握しながら後続工程を
スタートする事が可能

映像美を支えたワークフロー



風からデザインする



実機でのシミュレーションを利用
時は風の設定が非常に重要

シネマティクスでの追い風

- 絵として破綻しやすい
- プレイアブル中と印象が変わらない強さにする

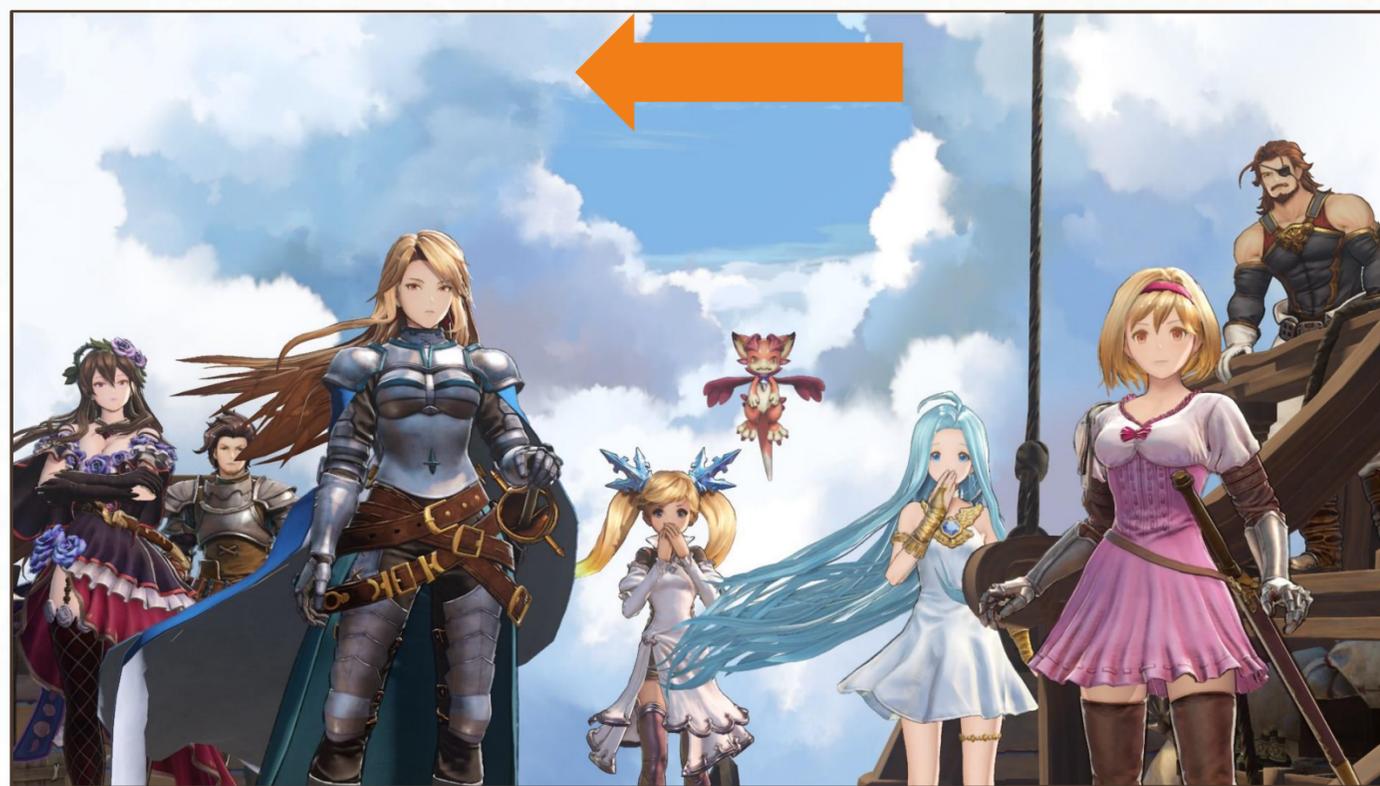
シネマティクスの立ち位置と
基準を決める所から突き詰める

映像美を支えたワークフロー

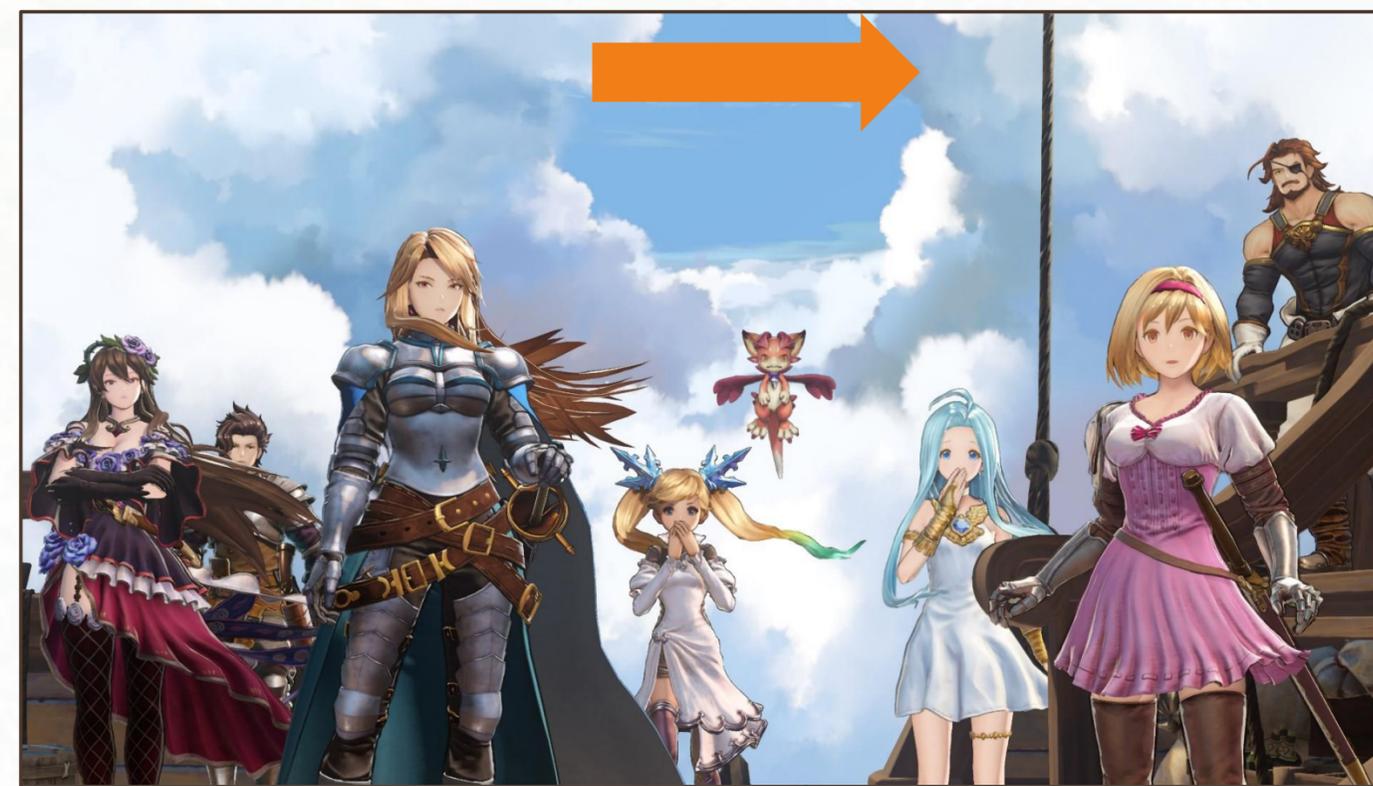
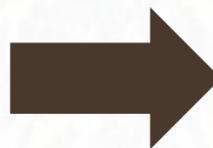


風のイメージも手軽に確認

先ほどのライブリンクとリアルタイムレンダリングも併せ、プレイアブル中の風の向きと、カットシーン中の風の向きも併せながらも、ショットごとに手軽に微調整を加えて絵作りをすることが可能



変更





Animation

Preview [Play] [Stop] [Previous] [Next] [Timeline]

2987 | 30:00 | 35:00 | 40:00 | 45:00 | 50:00 | 55:00 | 60:00 | 65:00 | 70:00 | 75:00 | 80:00 | 85:00 | 90:00 | 95:00 | 100:00 | 105:00

Recorded

- scnobj_wind_scene_object : Timeline Wind Scene Object.Active_ [checked]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Shake Speed Rate_ [1]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Speed Dir Oscillation First_ [0]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Speed Dir Oscillation Second_ [0]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Speed Oscillation First_ [0]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Speed Oscillation Second_ [0]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Wind Dir Wave Play Time First_ [1]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Wind Dir Wave Play Time Second_ [0.4]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Wind Rotation_
 - Wind Rotation_x [0]
 - Wind Rotation_y [0]
 - Wind Rotation_z [0]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Wind Velocity_ [0]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Wind Wave Play Time First_ [1]
- scnobj_wind_scene_object : Timeline Wind Scene Object.Wind Wave Play Time Second_ [1]

Add Property

Dopesheet Curves

映像美を支えたワークフロー



セカンダリフィードバック技術の恩恵



プライマリのみの状態で風の向きや強さなど、方向性が決まった段階でセカンダリのアニメーション情報を実機から出力し

プライマリ+セカンダリのベイクモーションに変更

これによりリアルタイムレンダリングのパフォーマンスを最適化

映像美を支えたワークフロー



プライマリの変更



工程として柔軟性があり
プライマリがもし変更になったとしてもシミュレーションをすぐに取り直すこともできますが
気に入ったセカンダリのまま
プライマリに追従させる機能も
あるため後続に影響少なく変更が可能

映像美を支えたワークフロー



プレイアブル中の風が変わっても対応可能



少ない事例ではありますが
フィールドの風の向きに
調整が入った場合でも
カットシーン中のイメージを崩さずに
再出力を手軽に行う事が可能
非常に**変更**に強いフローに！

映像美を支えたワークフロー



こうしてワークフローが実現

アニメーションの
工数を削減

全体の工数を
大きく前倒し

変更にも強い
ワークフロー

映像にこだわる
時間の捻出

- アーティストのからの希望
- シネマティクスへ導入しても遜色のないシミュレーション技術

まとめ



テーマ

没入感

映像美



没入感を支えた技術

リアルタイムレンダリング



- オクルージョンカリングの最大限活用
- LODの自動リクエスト
- 可変フレーム対応
- 最適化のチェック体制

得られた物

プレイアブル中と同じ
高フレームレート

高解像度

武器の反映

没入感を支えた技術

セカンドリフィールドバック



- Export Tool
- Merge Tool

得られた物

プレイアブル中と同じセカンドリ

強固なワークフロー

没入感を支えた技術

シームレス遷移

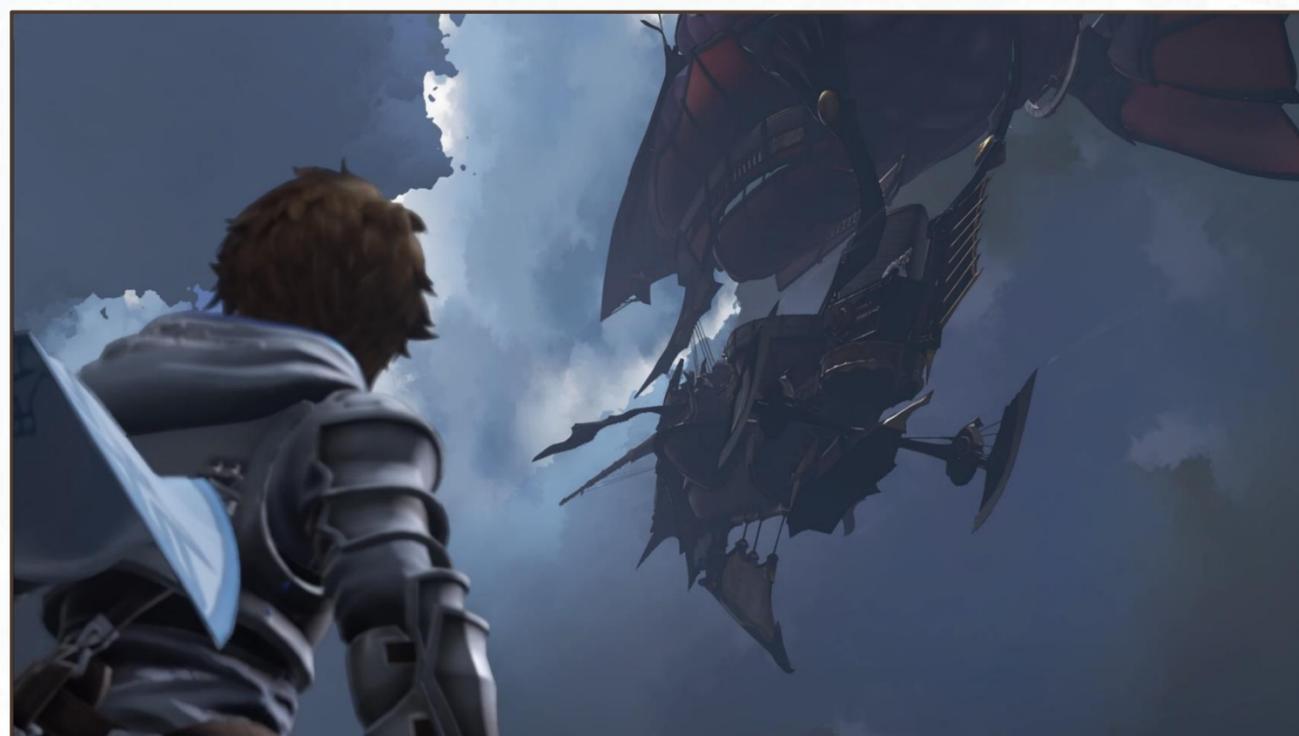


- カットシーンとシームレス遷移
- 会話シーンとシームレス遷移
- フェードアウトの廃止

得られた物

映像とプレイアブルの地続き感

映像美を支えた 強固なワークフロー



- Unityを
インターフェースにした環境
- ライブリンクと
リアルタイムレンダリング
- セカンダリフィードバックによる
工数削減

得られた物

映像美を制作するためにこだわる時間

テーマの解決で得られた物

没入感

映像美



全ては「面白い!」の一言のために
最高の没入感を追求していきます!



GRANBLUE FANTASY
Relink