



ゲームエンジンで 大規模開発する場合での リソース管理の落とし穴

第1 ビジネス・ディビジョン
リードプログラマー / ディレクター
浜口直樹

おしながき

- ・ 自己紹介
- ・ スマートフォン市場を取り巻く環境
- ・ 大規模開発で失敗しないために

自己紹介

浜口直樹

FFXⅡ ,FFXⅢ ,FFXⅢ-2,LRFFXⅢなど

10年以上FINAL FANTASYに関わるタイトルを担当

直近では・・・

MOBIUS FINAL FANTASY /プロジェクトリーダー

FINAL FANTASY VII REMAKE /開発リーダー



おしながき

- ・ 自己紹介
- ・ スマートフォン市場を取り巻く環境
- ・ 大規模開発で失敗しないために

スマートフォン市場を取り巻く環境

- ①端末スペック
- ②開発規模
- ③開発環境

①端末スペック

- RAM (内蔵メモリ)

1G～3G

※最近だと4Gモデル端末も・・・

- CPU

マルチコアが一般的

- 解像度

HD,FHD相当

※最近だと4K相当のモデル端末も・・・

①端末スペック

- RAM (内蔵メモリ)

1G～3G

※最近だと4Gモデル端末も・・・

- CPU

マルチコアが一般的

- 解像度

HD,FHD相当

※最近だと4K相当のモデル端末も・・・

スペックは上がっているが・・・実情としては、持ち運びが前提のスマホは消費電力を考慮してフルスペック使用することは難しい。。。RAMに関しても複数アプリ起動がスマホは求められるのでRAMも500M～700Mを超えるとOS側がアプリを強制終了させることが一般的。

①端末スペック

- RAM (内蔵)

1G～3G

※最近だと4Gモデル端末も・・・

- CPU

マルチコアが一般的

HD,FHD相当

※最近だと4K相当のモデル端末も・・・



ちなみに・・・

PlayStation®3頃の世代のゲーム機は
ゲームで利用できるRAMは500M程度だったので、
現在のスマホアプリと近い環境だともいえる

持ち運びが前提のスマホは消費電力を考慮してフルスペック使用することは難しい。。。
RAMに関しても複数アプリ起動がスマホは求められるので
RAMも500M～700Mを超えるとOS側がアプリを強制終了させることが一般的。

①端末スペック

・ RAM (内蔵) 1G～3G

・ CPU



ちなみに・・・

PlayStation®3頃の世代のゲーム機は
ゲームで利用できるRAMは500M程度だったので、
現在のスマホアプリと近い環境だともいえる

さらに・・・

PlayStation®4などの現世代のゲーム機は
RAMが5G程度使用できるため、
ハイスペックなコンテンツという面では、
スマホとの差はあるともいえる

持ち運びが前提のスマホは消費電力を考慮してフルスペック使用することは難しい。。。
RAMに関しても複数アプリ起動がスマホは求められるので
RAMも500M～700Mを超えるとOS側がアプリを強制終了させることが一般的。

②開発規模

- 開発費

→スマホ市場として5～6年前は
5000万円以下が一般 1億円は豪華なアプリ
と言われていました。

→現在は、2億円以上かかるアプリが多く、
10億円以上かかるタイトルもあるというニュースを
見ることもあります。

②開発規模

• 開発費

→スマホ市場として5～6年
5000万円以下が一般 1億
と言われていました。

→現在は、2億円以上かかるアプリが多く、
10億円以上かかるタイトルもあるというニュースを
見ることもあります。



昔は、アプリ数をいっぱい投入してみて
数を打てばどれかがあたるという発想で
開発される風潮が市場にありました。

②開発規模

・開発費

→スマホ市場として5～6年
5000万円以下が一般 1億
と言われていました。



昔は、アプリ数をいっぱい投入してみて
数を打てばどれかがあたるという発想で
開発される風潮が市場にありました。

主力のタイトルも確立してきて、
スマホアプリ市場も飽和気味。
開発費をかけてしっかりと作りこまれた
アプリを出さないと、
他のタイトルとの差別化が難しくなってきた。



アプリが多く、
競争もあるというニュースを

③開発環境

- ・ 自社エンジンを開発
- ・ 他社のゲームエンジン（統合開発環境）を使う
→UNITY®やUNREAL ENGINEが主流



③開発環境



・自社エンジンを開発

ひと昔は、
カジュアルゲームをお手軽に作るために
速度感を持って開発ができるように
統合開発環境が採用されるイメージがあった。

（統合開発環境）を使う
ENGINEが主流



③開発環境



・自社エンジンを開発

ひと昔は、
カジュアルゲームをお手軽に作るために
速度感を持って開発ができるように
統合開発環境が採用されるイメージがあった。

現在では、
開発に求められる技術が高くなってきており、
自社でゼロからゲーム開発環境を作るより、
導入からアウトプットまでが早い統合開発環境が
採用させることが増えてきている。



ENGINEが主流



③開発環境



ひと昔は、
カジュアルゲー
速度感を持つ
統合開発環境

統合開発環境を使って開発規模の大きな開発が増えてきたために、
開発が難航するケースが増えてきております。

統合開発環境は万能ではないため、
しっかりと開発チームのワークフローに組み込んでいく必要があります。
そこが本日のセッションの本題となります。



環境が



おしながき

- ・ 自己紹介
- ・ スマートフォン市場を取り巻く環境
- ・ 大規模開発で失敗しないために

大規模開発で失敗しないために

- ・ゲームが面白くない。。。。
- ・バグがいっぱいで開発の先行きが不安。。。。
- ・処理負荷が重い、メモリ不足でアプリが落ちる

大規模開発で失敗しないために

- ・ゲームが面白くない。。。。

↑プランナー頑張ってください

- ・バグがいっぱいで開発の先行きが不安。。。。
- ・処理負荷が重い、メモリ不足でアプリが落ちる

大規模開発で失敗しないために

- ・ゲームが面白くない。。。。

↑プランナー頑張ってください

- ・バグがいっぱいで開発の先行きが不安。。。。

↑エンジニアと品質管部門で頑張ってください

- ・処理負荷が重い、メモリ不足でアプリが落ちる

大規模開発で失敗しないために

- ・ゲームが面白くない。。。。

↑プランナー頑張ってください

- ・バグがいっぱいで開発の先行きが不安。。。。

↑エンジニアと品質管部門で頑張ってください

- ・処理負荷が重い、メモリ不足でアプリが落ちる

↑メモリ不足、処理負荷が重い場合の大半はエンジニアとデザイナーとプランナーが統合環境の特性を理解しないで開発を進めてしまったときに発生します。
特にWindowsなどのメモリが膨大な環境で作り上げていると、
気が付いたら、あれ。。。？全然ターゲット端末で動かない。。。困った。。。
という状態になってしまいます。
そこで今日は、統合環境共通でいえるメモリ不足とリソース管理についてのお話をします。

統合開発環境での失敗例

- ・ 統合開発環境はレベルエディターにリソースを配置してゲーム内を構成します。

→配置する空間データ (layout) のことを

UNITYではscene

UNREAL ENGINE4でlevelと呼びます

- ・ リソースとは、3Dモデルやモーションやサウンドなどゲームに必要なデータ



背景リソース



キャラリソース

統合開発環境での失敗例

- ①空間データ (layout) にリソースを配置していきます。

まずは、背景リソースから・・・



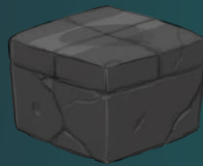
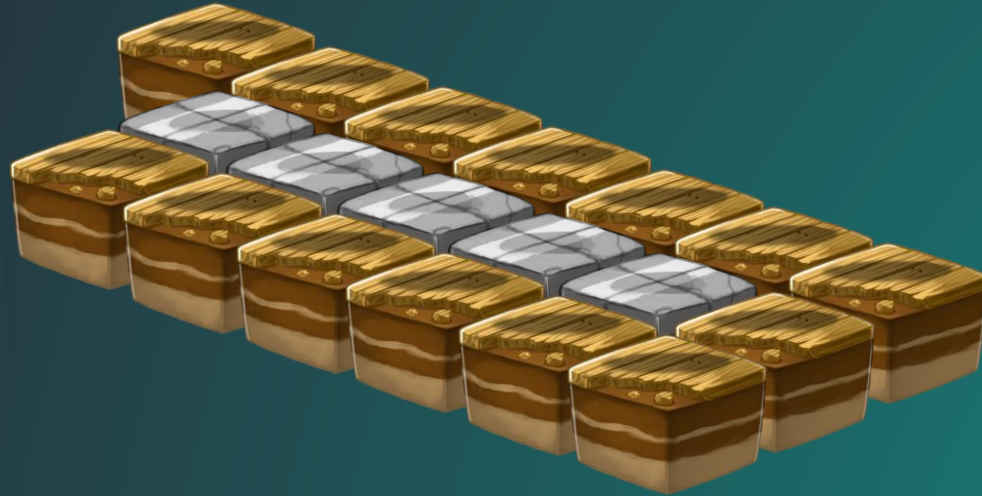
背景リソース



キャラリソース

統合開発環境での失敗例

- ①空間データ (layout) にリソースを配置していきます。



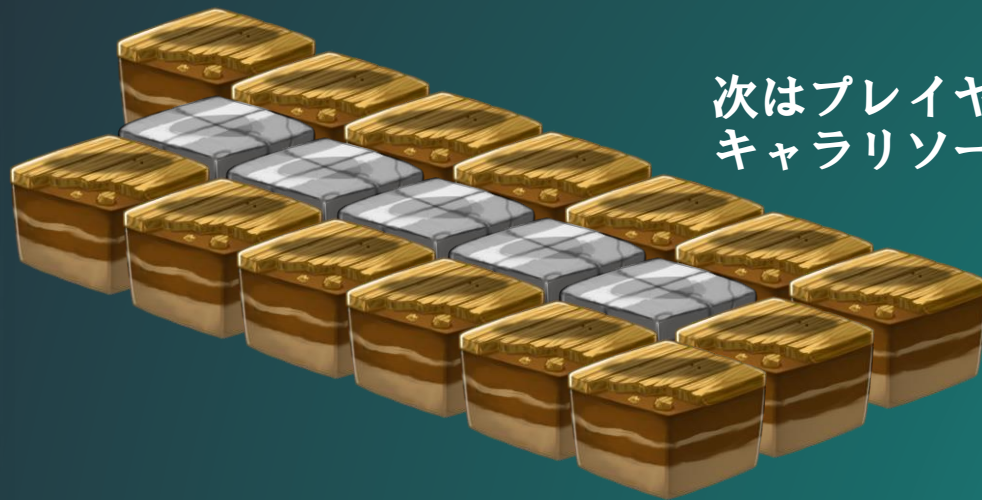
背景リソース



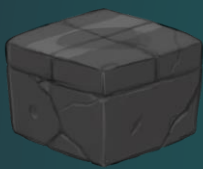
キャラリソース

統合開発環境での失敗例

- ①空間データ (layout) にリソースを配置していきます。



次はプレイヤーの
キャラリソース・・・



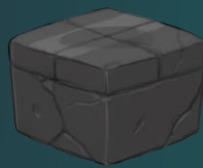
背景リソース



キャラリソース

統合開発環境での失敗例

- ①空間データ (layout) にリソースを配置していきます。



背景リソース



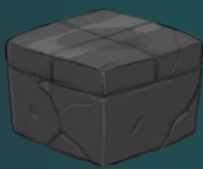
キャラリソース

統合開発環境での失敗例

- ・ ①空間データ (layout) にリソースを配置していきます。



次は敵のキャラリソース・・・



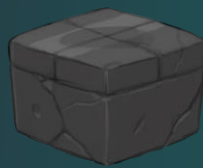
背景リソース



キャラリソース

統合開発環境での失敗例

- ・ ①空間データ (layout) にリソースを配置していきます。



背景リソース



キャラリソース

統合開発環境での失敗例

- ・ ②配置リソースが使用するリソースの設定をする



統合開発環境での失敗例

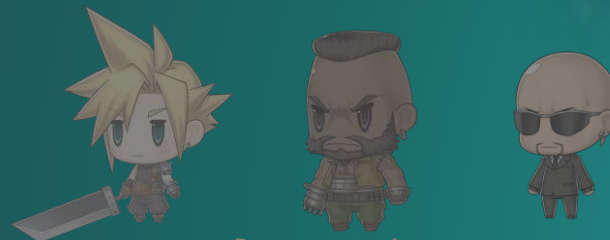
- ・ ③カットシーンなどトリガーボックスで起動する
制御用のデータなどを配置する

制御用のプログラムやスクリプト
モーション
エフェクト
サウンド

制御用のプログラムやスクリプト
モーション
エフェクト
サウンド



背景リソース



キャラリソース

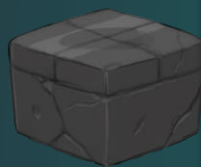
統合開発環境での失敗例

- ④ゲームに必要なデータを全て空間データ (layout) から依存関係があるように設定を行う

制御用のプログラムやスクリプト
モーション
エフェクト
サウンド

制御用のプログラムやスクリプト
モーション
エフェクト
サウンド

カットシーンのスケジュールデータ
モーション
エフェクト
サウンド

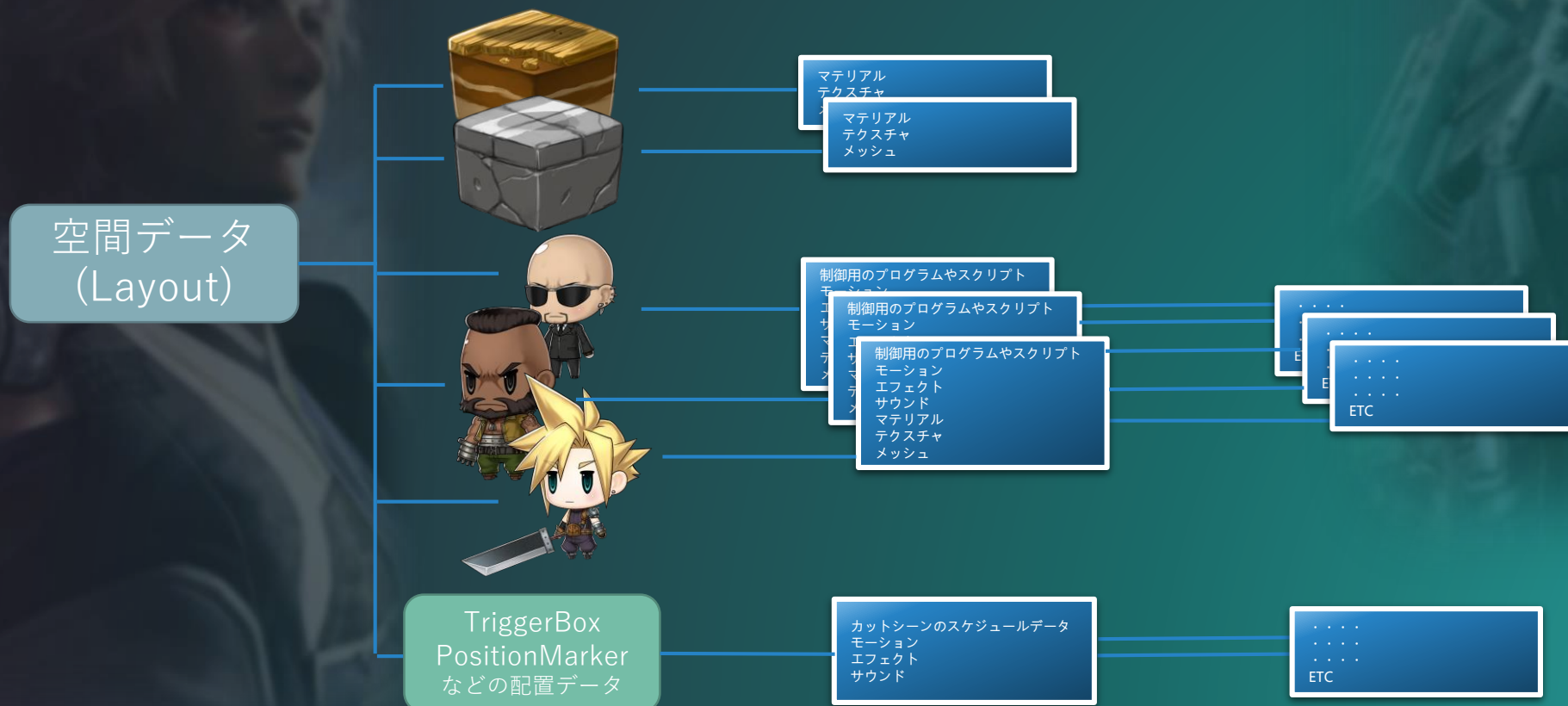


背景リソース

キャラリソース

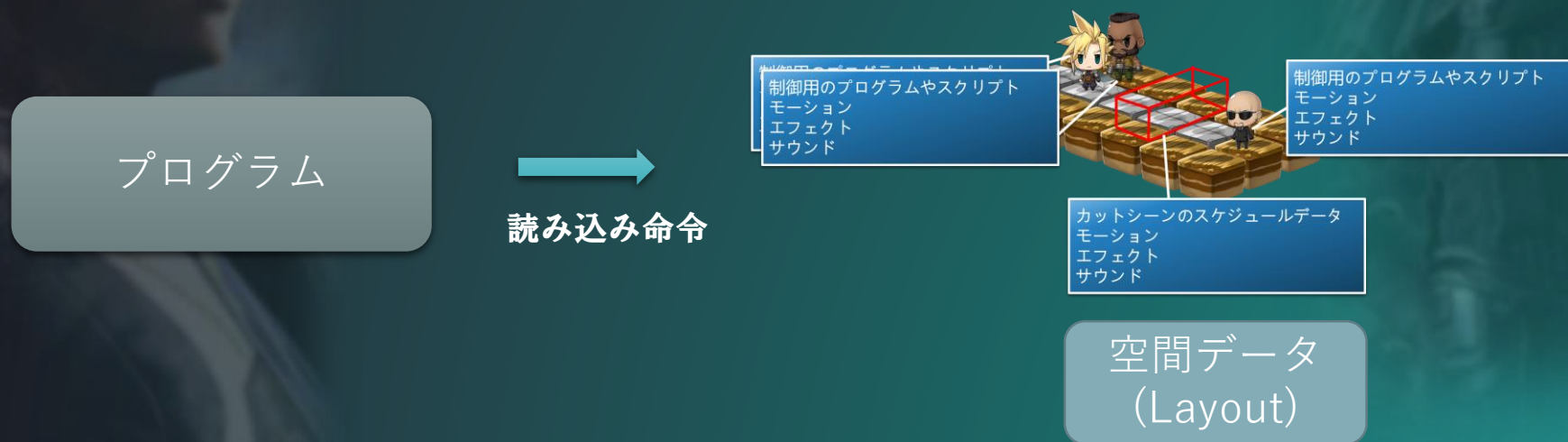
統合開発環境での失敗例

- ④ ゲームに必要なデータを全て空間データ (layout) から依存関係があるように設定を行う



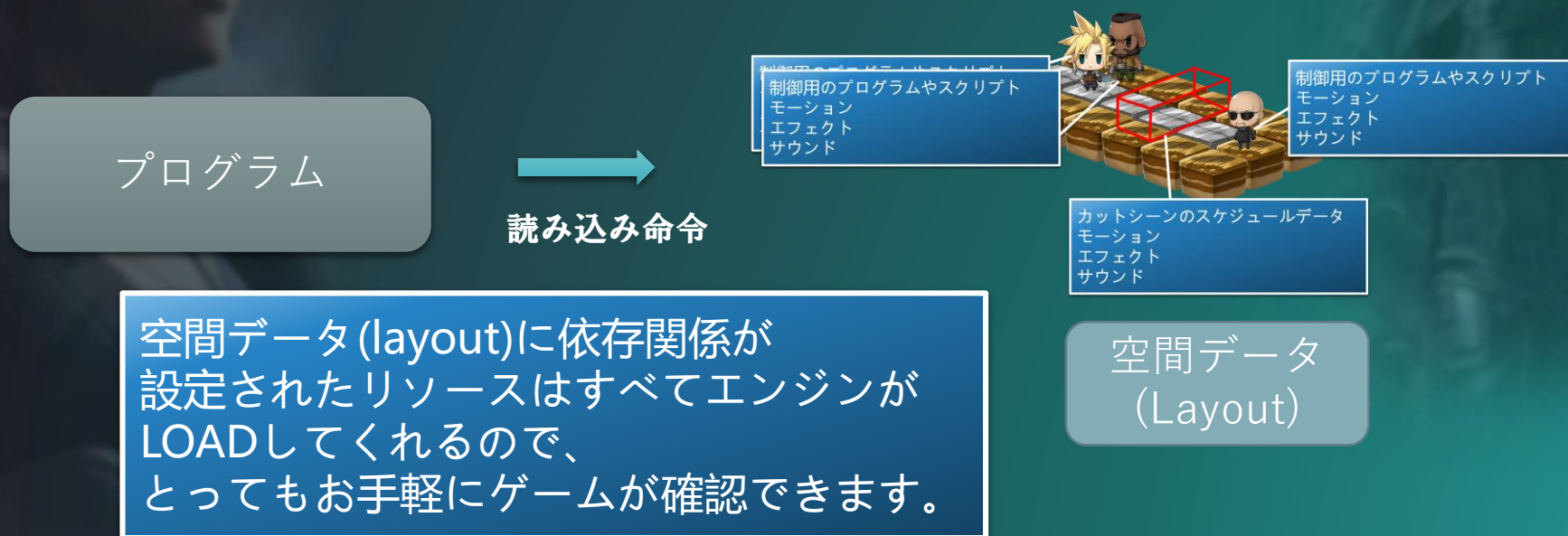
統合開発環境での失敗例

- ・ ⑤ プログラムから空間データ (layout) をLOADするだけでゲームが開始する。



統合開発環境での失敗例

- ・ ⑤ プログラムから空間データ (layout) をLOADするだけでゲームが開始する。



統合開発環境での失敗例

- ・ ⑤ プログラムから空間データ (layout) をLOADするだけでゲームが開始する。



統合開発環境での失敗例

- ・ ⑤ プログラムから空間データ (layout) をLOADするだけでゲームが開始する。

ブリザド

ファイア

空間データ (Layout) に必要なデータを設定するだけでゲームが確認できるので、プロトタイプ版には最適！！

プロデューサーも大喜び。
開発規模も大きくして量産するぞ！！



統合開発環境での失敗例

- ・ ⑤ プログラムから空間データ (layout) をLOADするだけでゲームが開始する。

ブリザド

ファイア

統合開発環境は良い面ばかり。

???
本当にそうでしょうか・・・???

統合開発環境での失敗例

- ⑥開発も好調！！
本製作に向けてドンドンリソースを追加していきましょう



背景リソース



キャラリソース

統合開発環境での失敗例

- ⑥開発も好調！！
本製作に向けてドンドンリソースを追加していきましょう



背景リソース

キャラリソース

統合開発環境での失敗例

- ⑥開発も好調！！
本製作に向けてドンドンリソースを追加していきましょう

FFで必要そうなリソースを考えてみます。

...

→途中でパーティが追加される
→新しい技を覚えた
→召喚獣が使えるようになった
→進行で背景が破壊されて破壊形状が必要
→ボスが段階によって進化する
などなど、
ユーザーのプレイによって動的に
変更されるリソースはたくさん存在します。



背景リソース



キャラリソース

統合開発環境での失敗例

- ⑥開発も好調！！
本製作に向けてドンドンリソースを追加していきましょう



FFで必要そうなリソースを考えてみます。

...

- 途中でパーティが追加される
- 新しい技を覚えた
- 召喚獣が使えるようになった
- 進行で背景が破壊されて破壊形状が必要
- ボスが段階によって進化する

どれだけ必要なリソースがあっても
全部、空間データ（Layout）に設定しておけば、
リソースはLOADしてくれるので問題ない。
はず・・・？

に
存在します。

背景リソース

キャラリソース

統合開発環境での失敗例

- ⑦製品想定のリソースがすべて設定されたので、実行してチェックをして完成・・・？

プログラム

読み込み命令

空間データ
(Layout)



統合開発環境での失敗例

- ・ ⑦製品想定のリソースがすべて設定されたので、実行してチェックをして完成・・・？

プログラム



統合開発環境がすべてリソースをガンガンLOADしてくれます。
なぜなら・・・必要な全てのリソースは空間データ (layout) に設定されているからです。

空間データ
(Layout)



統合開発環境での失敗例

- ・ ⑦製品想定のリソースがすべて設定されたので、実行してチェックをして完成・・・？

プログラマー

統合開発
LOAD
なぜな
空間デ

その結果。。。

空間データ
(Layout)



統合開発環境での失敗例

- ・ ⑦製品想定のリソースがすべて設定されたので、実行してチェックをして完成・・・？

プログラマー

統合開発
LOAD
なぜな
空間デ

当然。。。そ

間データ
(Layout)



統合開発環境での失敗例

- ・ ⑦製品想定のリソースがすべて設定されたので、実行してチェックをして完成・・・？

プログラマー

統合開発
LOAD
なぜな
空間デ

当然
そ

メモリが足りなくて、アプリが落ちます(笑)

データ
(Layout)



統合開発環境での失敗例

- ・メモリが足りなくなった時にはリソース同士の依存関係がこんがらかっていて、絶望します。

カットシーンのスケジュールデータ
モーション
エフェクト
サウンド

制御用のプログラムやスクリプト

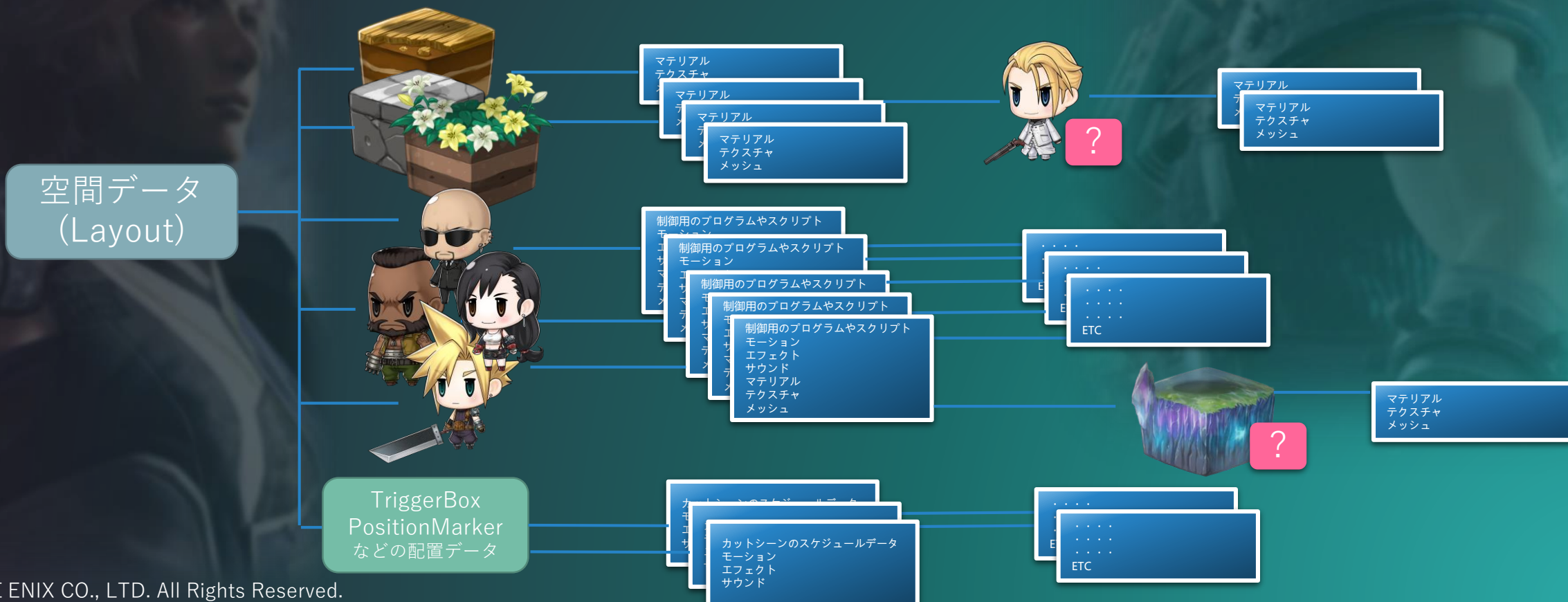
制御用のプログラムやスクリプト
モーション
エフェクト
サウンド

制御用のプログラムやスクリプト
モーション
エフェクト
サウンド

ありとあらゆる、
数々なリソース。。。

統合開発環境での失敗例

- メモリが足りなくなった時にはリソース同士の依存関係がこんがらがっていて、絶望します。



統合開発環境での失敗例

- ・メモリが足りなくなった時にはリソース同士の依存関係がこんがらかっていて、絶望します。

空間データ
(Layout)

統合開発環境は、
リソースの依存関係があるだけで、
必要なリソースをすべて読み込んでくれるので
非常に強力であり便利ではあるが、
その反面しっかりとリソースの依存関係を管理しないと、
どの依存関係からリソースがLOADされているのかが
把握しにくいので、メモリ不足でもなにから手を付けられ
ばいいのか途方に暮れることになります。。。

PositionMarker
などの配置データ

カットシーンのスケジュールデータ
モーション
エフェクト
サウンド

ETC

マテリアル
テクスチャ
メッシュ

リソース設計の重要性

- ・統合開発環境は様々な開発機能があり、非常に強力な環境ではありますが・・・

しかし、どんな場面でどんなリソースをLOADしておく必要があるかは、当然ゲーム性によって異なるので、リソースのLOADの設計は開発チームでしっかりと構築する必要がある。

リソース設計の重要性

- ・統合開発環境は様々な開発機能があり
非常に強い

統合開発環境は、
開発のための機能はたくさん与えてくれますが、
決められた開発のワークフローを
提示してくれるわけではないです。

しかし、このように開発環境はLOADしておく
必要があるかは、当然ゲーム性によって異なるので、
リソースのLOADの設計は開発チームでしっかりと
構築する必要がある。

リソース設計の重要性

- ・統合開発環境は様々な開発機能があり
非常に強い

統合開発環境は、
開発のための機能はたくさん与えてくれますが、
決められえた開発のワークフローを
提示

しかし、
必要がある
リソースのLOADの設計は開発チームでしっかりと
構築する必要がある。
ておく
ので、
と

行き当たりばったりで開発を進めるのではなく、
統合開発環境の機能をしっかりと理解して、
開発チームのワークフローにどのように
組み込むのかをしっかりと設計するようにしましょう。

リソース設計の重要性

- ・ リソースの特性に合わせてLOAD方法を設計しよう



リソース設計の重要性

- ・ リソースの特性に合わせてLOAD方法を設計しよう



ゲーム全編通して必要で、
ゲーム起動時にLOADしておく
ほうが効率的なリソース

起動時にLOAD
しておくリソース

リソース設計の重要性

- ・ リソースの特性に合わせてLOAD方法を設計しよう



LayoutからLOAD
するリソース

空間データ(layout)を
ひとつで構成するのではなく
最初から必要になる単位で分割して
必要なときに必要な空間データ(layout)
だけをLOADできるように設計しよう



ゲーム全編通して必要で、
ゲーム起動時にLOADしておく
ほうが効率的なリソース

起動時にLOAD
しておくリソース

リソース設計の重要性

- ・ リソースの特性に合わせてLOAD方法を設計しよう



パーティや装備など
カスタマイズによって変わるような
リソースは空間データ(layout)
には頼らず、プログラム側で個別に
リソースをLOADしよう。

プログラムからLOAD
するリソース



LayoutからLOAD
するリソース

空間データ(layout)を
ひとつで構成するのではなく
最初から必要になる単位で分割して
必要なときに必要な空間データ(layout)
だけをLOADできるように設計しよう



ゲーム全編通して必要で、
ゲーム起動時にLOADしておく
ほうが効率的なリソース

起動時にLOAD
しておくリソース

リソース設計の重要性

- ・ プリプロのときは空間データ (layout) でやりきるのはよいが、遅くともプリプロ後にゲームの方向性が見えてきたら、リソース設計は絶対に行うようにしましょう。



ご清聴、ありがとうございました。



全部終わったらまた、ね

CHARACTER DESIGN:YASUHISA IZUMISAWA.

© 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

© 2015 - 2018 SQUARE ENIX CO., LTD. All Rights Reserved.

© SQUARE ENIX CO., LTD. All Rights Reserved.

CHARACTER DESIGN:TETSUYA NOMURA/ROBERTO FERRARI

PlayStationは株式会社ソニー・インタラクティブエンタテインメントの商標または登録商標です。

UNITYはUnity IPR ApSの商標または登録商標です。

UNREALはエピック・ゲームズ・インコーポレーテッドの商標または登録商標です。

WINDOWSはマイクロソフト コーポレーションの商標または登録商標です。

その他掲載されている会社名、商品名は、各社の商標または登録商標です。