

超少人数でのゲーム制作 ■■ノニラヤを作る工夫と 未来展望

株式会社ヘキサドライブ/山口裕也

- 撮影やSNSへの投稿は全てOK
Xに投稿の際は#ノニラヤがおすすめ



自己紹介

山口裕也

株式会社ヘキサドライブ

プログラマーです。主に描画関係を担当してきました。

最近では
「プロ野球スピリッツ2019」
「SILENT HILL: The Short Message」の開発に参加。

こうしてみると1タイトルの開発が長期化してきている気がしますね。



このセッションはこんな方にお勧め

- アセットストアに頼らずオリジナルゲーム開発をしたいけれど工数が心配
- プロシージャルモデリングに興味があるけれど具体的な事例を知りたい
- Blenderが好きな人
- 小人数でゲームを作りたい人



アジェンダ

アジェンダ

- 開発規模の紹介
- アセット制作効率化事例
- 息抜きエンジニアリング
- 総評





開発規模の紹介

開発規模の紹介

その前に

■■ノニラヤの紹介

- 2025年晩秋リリース予定
- 三人称視点の中世和風ホラーゲーム
- ホラーゲーム初心者にも優しいマイルド設計
- 対象プラットフォームはPC
- 3/31までXでキャンペーン実施中
- ウィッシュリスト登録をお願いします

神ノニラヤ
Niraya of ■■



■■ノニラヤの紹介



こんな雰囲気ของเกมです。

画面は開発中のものです、製品版とは異なる場合があります。
The game is currently in development. The final version may differ.

開発規模の紹介(開発体制)

- 基本一人
 - プログラム
 - 各種アセット(時々待機人員のヘルプ有)
 - アニメーション
 - レベルデザイン
 - 効果音
- 外部に発注するもの
 - 音声
 - 楽曲

開発規模の紹介

超小規模=1人

開発規模の紹介(一人開発の背景)

- オリジナルゲームを作りたい作り続けたい
- 黒字化必須
- デベロッパーなので販売に関するノウハウなし
- Steamで売れている同ジャンルのインディーゲームを調査
- 売上目標と開発費の決定
- 開発費の中で最大の成果を上げる方法を模索

開発規模の紹介(課題と結論)

- ○○人月強
- 表現したい世界観の実現
- 挑戦したいゲームシステムの実現
- 1人で時間かけたほうが可能性あるのでは?

開発規模の紹介(主な開発環境)

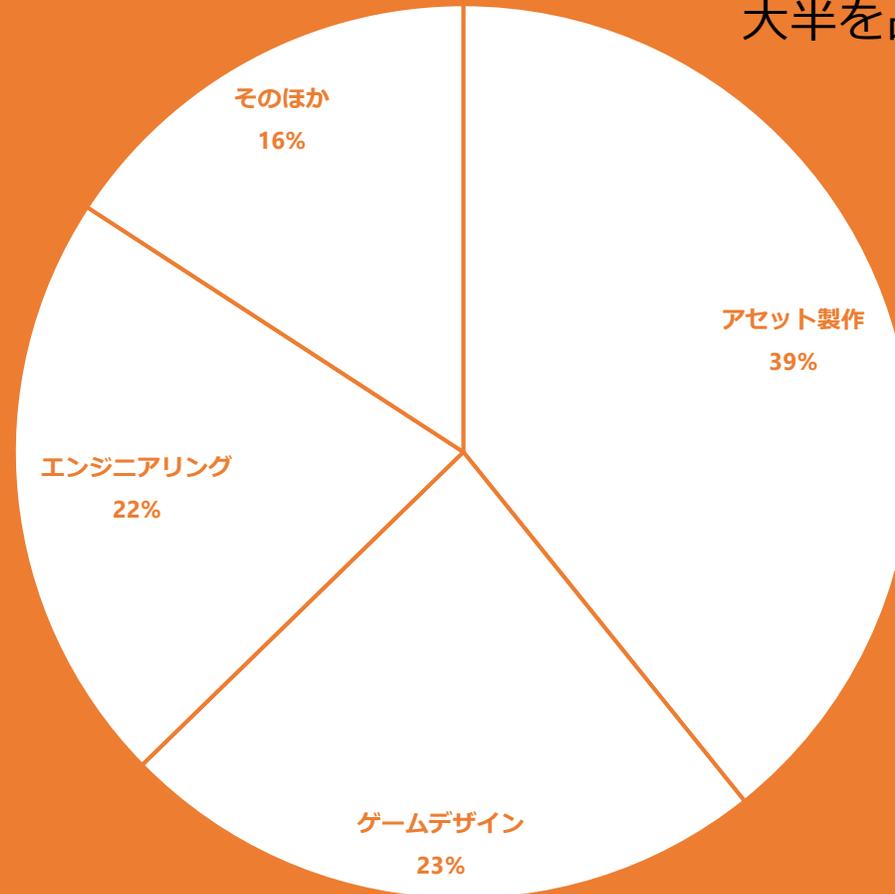
- DCCツール
 - 無料
 - Geometry Nodesに期待
- ゲームエンジン
 - C#好きイテレーション早い
 - HDRPに期待
- アニメーション
 - キャプチャー使おう
 - MVN準備が面倒



mocopi

開発規模の紹介(実際の作業割合)

作業内容内訳



現在までの消化タスクの工数をざっくり集計
大半を占めるのはアセット制作

開発体制が出来るまで

課題はアセット制作



アセット制作効率化事例

アセット制作効率化事例

要件

- 品質と工数のバランスを考慮した作り方が必要
ある程度の品質で工数内で作りきる事が優先
- 大規模なレベルの修正に対応できる作り方が必要

アセット制作効率化事例

それってプロシージャルでは？

アセット制作効率化事例

やってみた

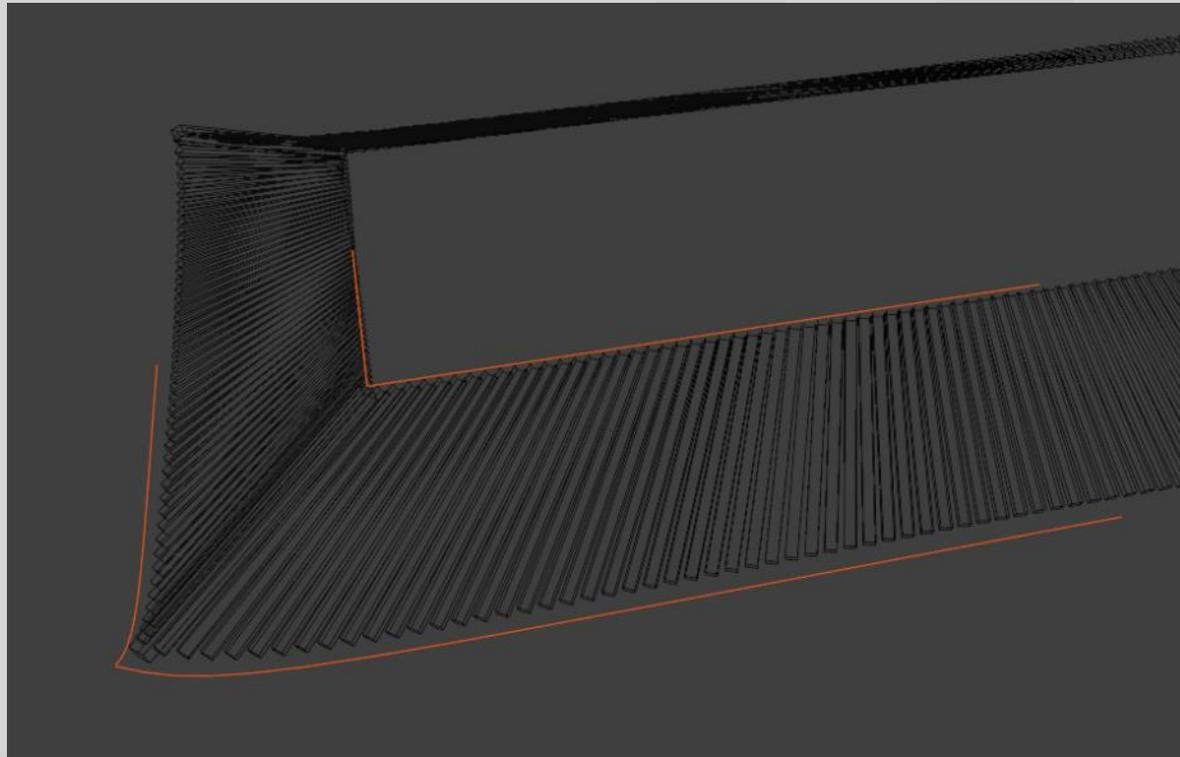
アセット制作効率化事例(プロシージャル)

建造物を配置してから微調整



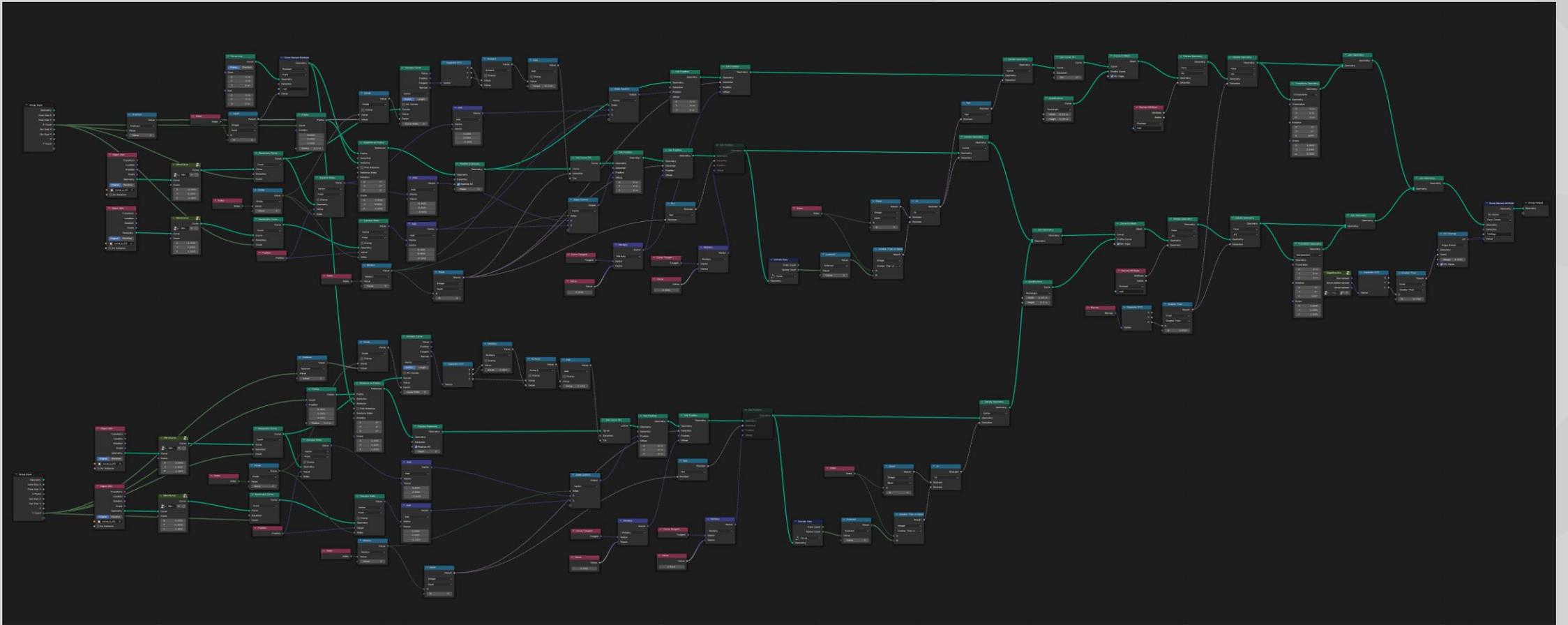
アセット制作効率化事例(プロシージャル)

事例では2本1対のカーブの間にラインを生成しラインからメッシュを生成



アセット制作効率化事例(プロシージャル)

実際のノードはこんな感じ



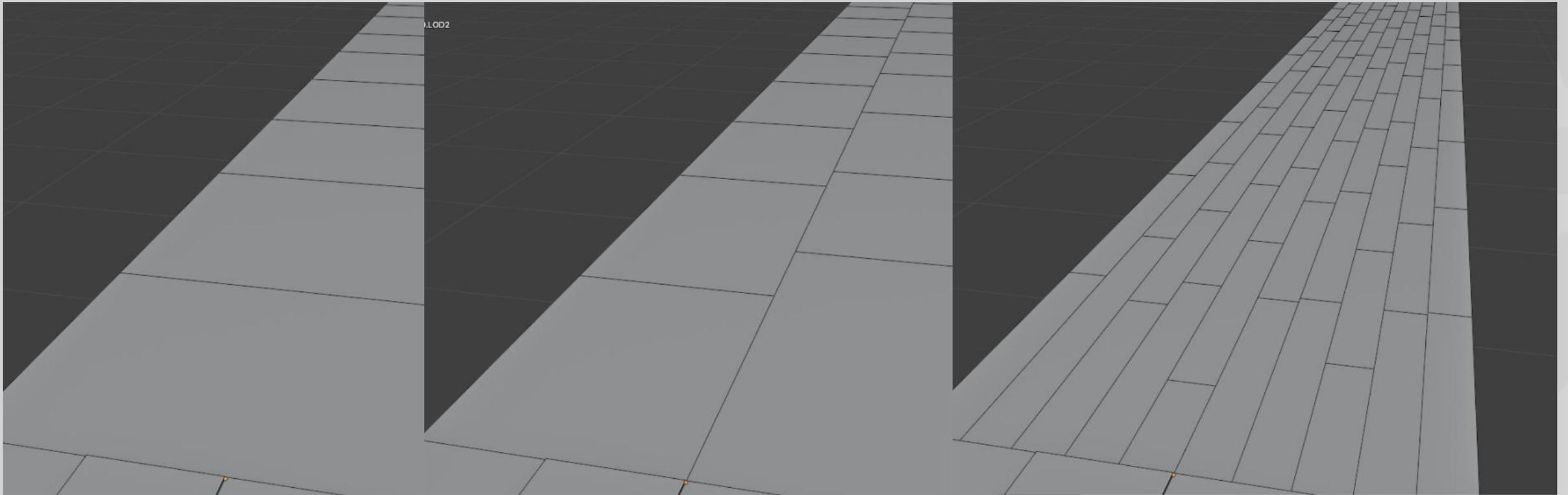
アセット制作効率化事例(プロシージャル)

他には凸凹のある床板
各階で歪み方が違う



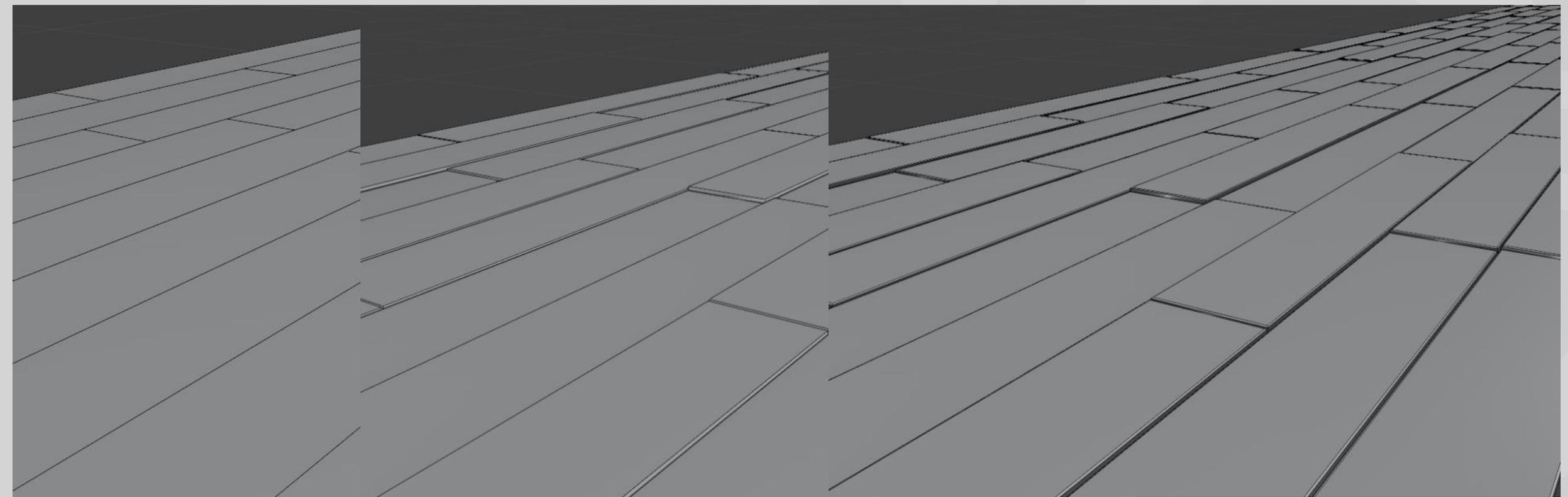
アセット制作効率化事例(プロシージャル)

平面の板をランダムにずらして並べて



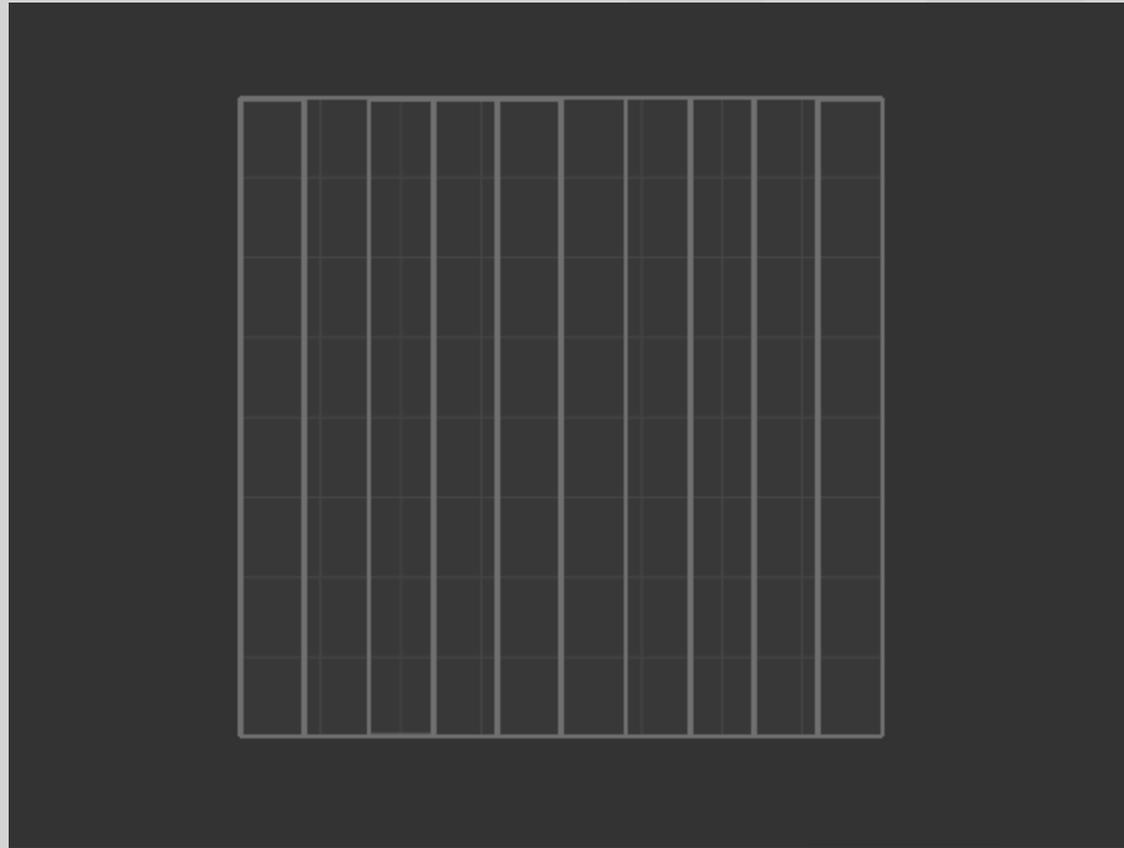
アセット制作効率化事例(プロシージャル)

さらにランダムに押し出してベベル



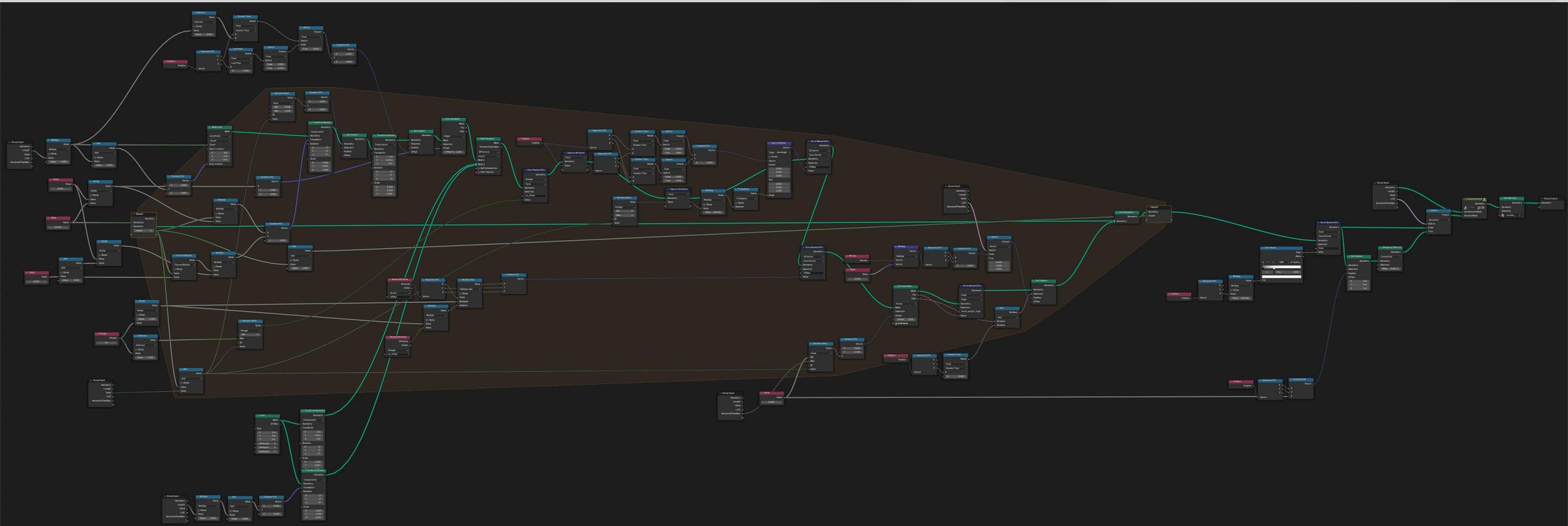
アセット制作効率化事例(プロシージャル)

UVはこう展開して後述のテクスチャフローを適用



アセット制作効率化事例(プロシージャル)

実際のノードはこんな感じ



アセット制作効率化事例(プロシージャル)

別パターンも

アセット制作効率化事例(プロシージャル)

血迷って広い屋外エリアを取り入れる

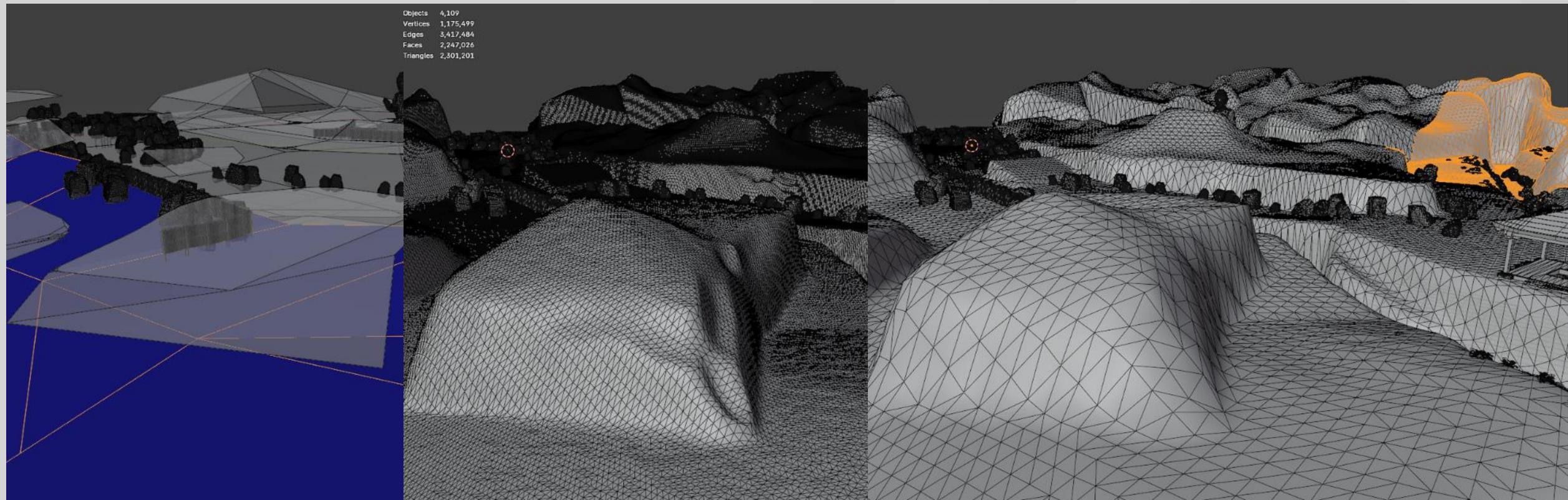


アセット制作効率化事例(プロシージャル)

- Terrainのブラシを試して挫折
- 歩ける場所や道をローポリで定義したら楽な気がする
- ついでに地面の材質をローポリのアトリビュートで塗り分けたい

アセット制作効率化事例(プロシージャル)

出来た

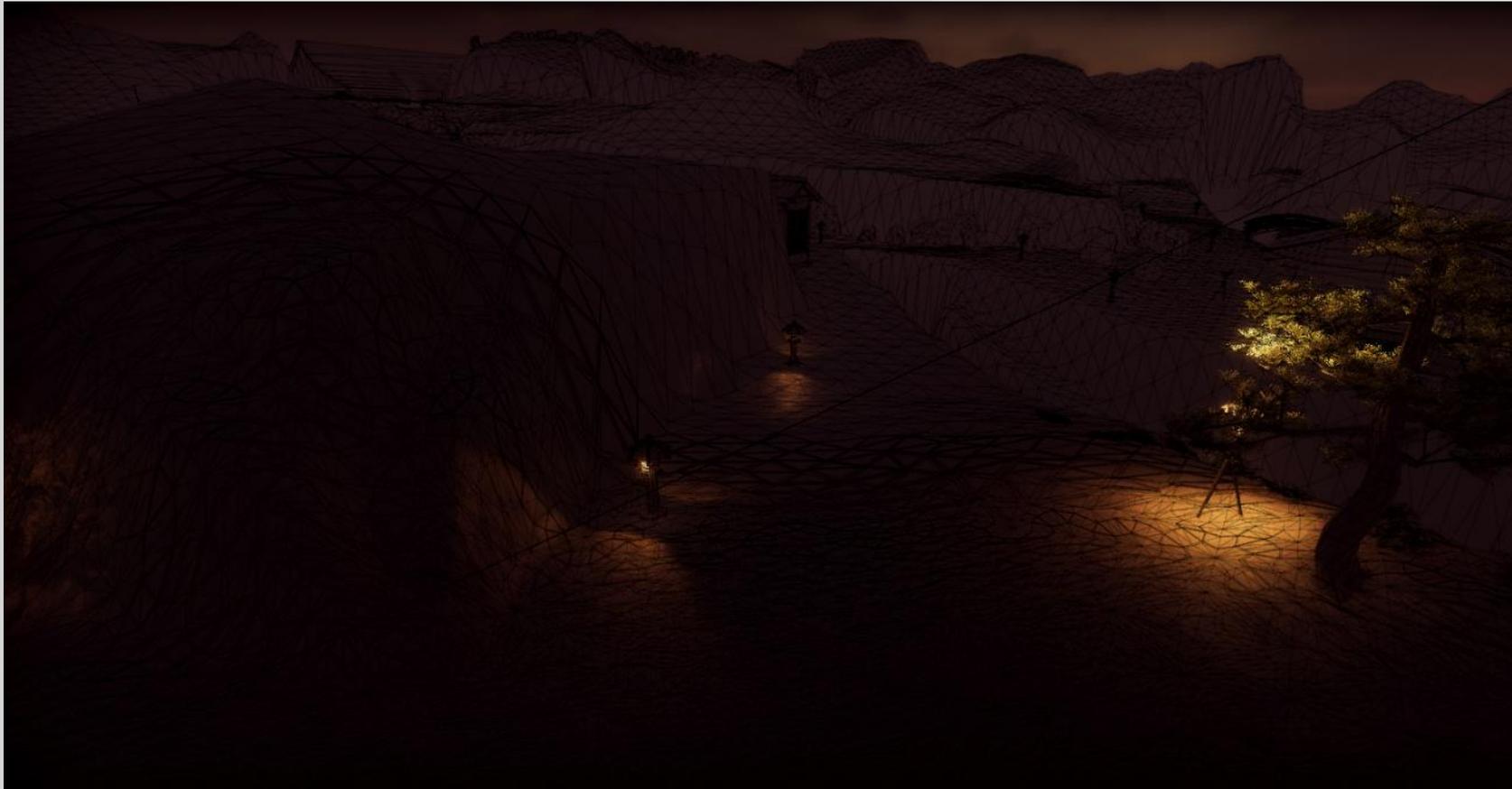


アセット制作効率化事例(プロシージャル)

- 高低差やノイズの入り方、地面の材質をローポリで設定
- ローポリの高さとノイズ用アトリビュートからハイポリ作成
- ハイポリを基にゲーム用リダクションメッシュを分割出力

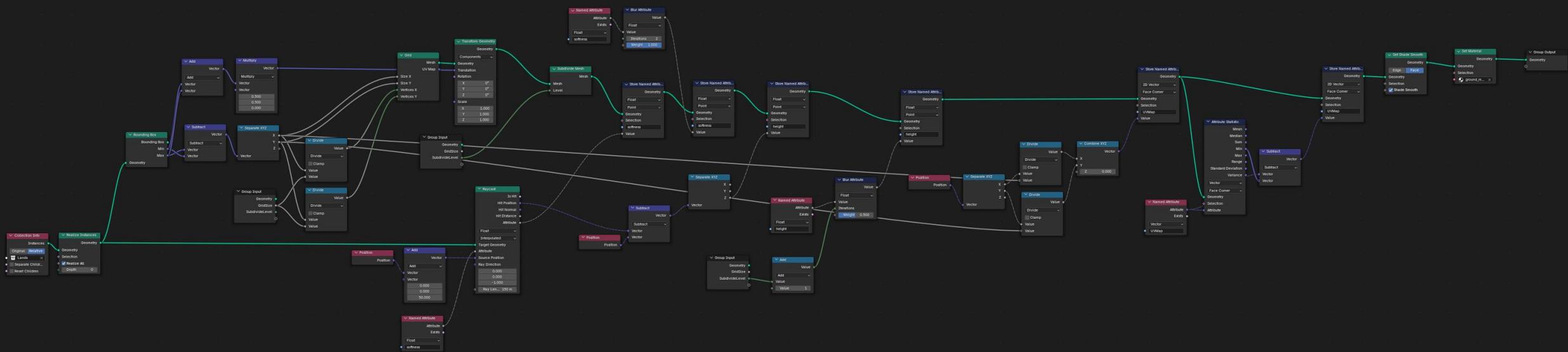
アセット制作効率化事例(プロシージャル)

ディテールはテッセレーションで復元



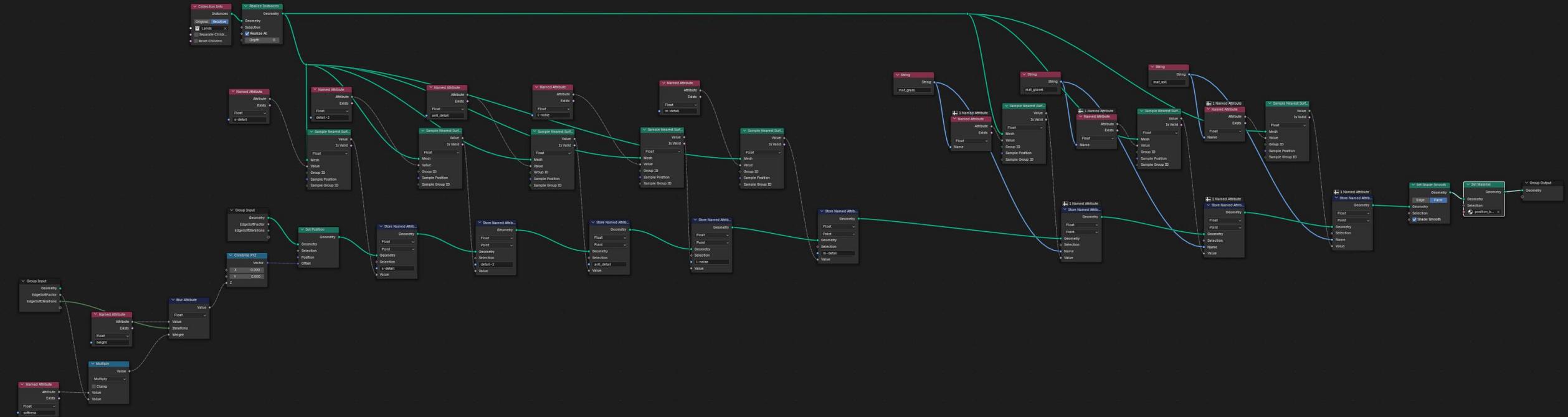
アセット制作効率化事例(プロシージャル)

ハイポリモデルを作る実際のノード
範囲に沿った板ポリを作成



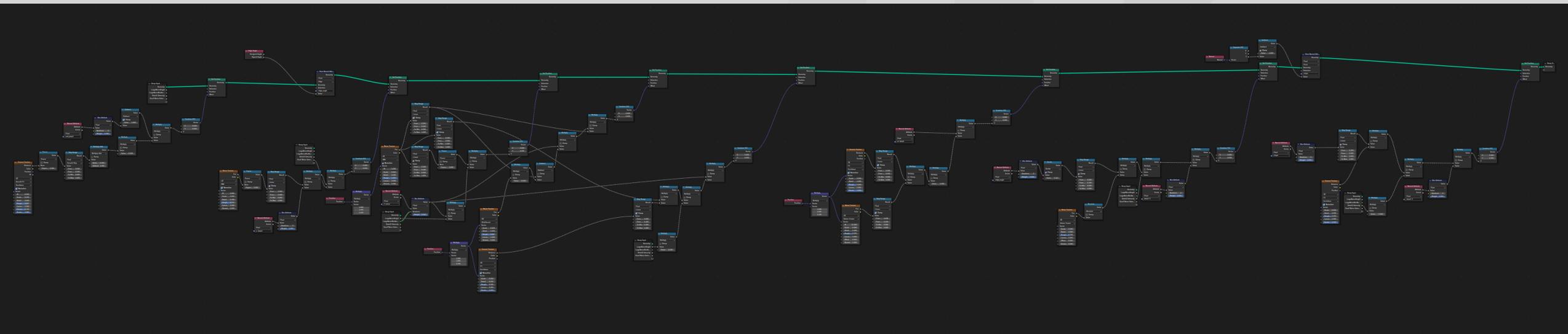
アセット制作効率化事例(プロシージャル)

ハイポリモデルを作る実際のノード
ローポリに沿って高低差をつける



アセット制作効率化事例(プロシージャル)

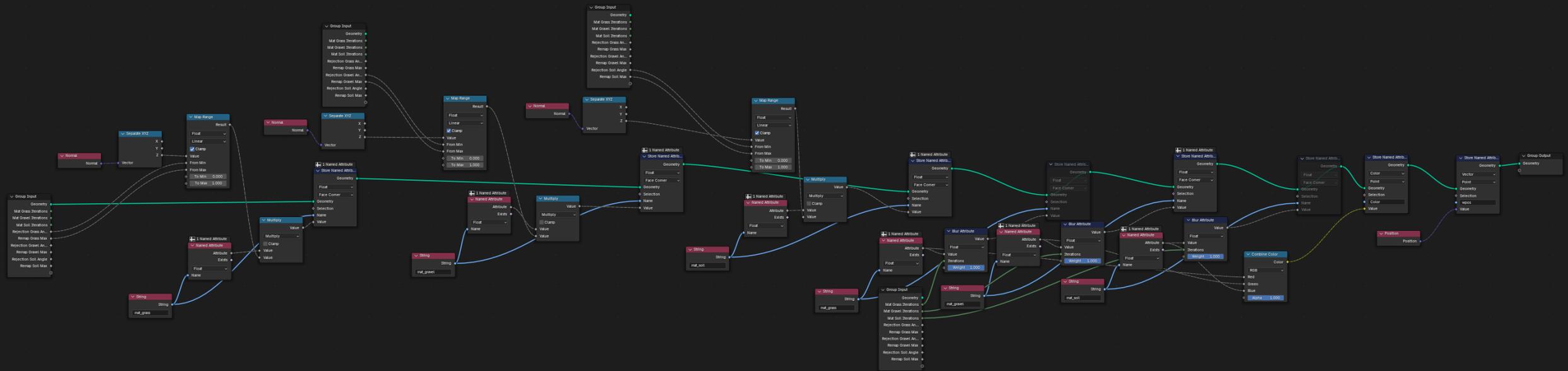
ハイポリモデルを作る実際のノード
ノイズを追加



アセット制作効率化事例(プロシージャル)

ハイポリモデルを作る実際のノード

アトリビュートからマテリアル分けのカラー作成

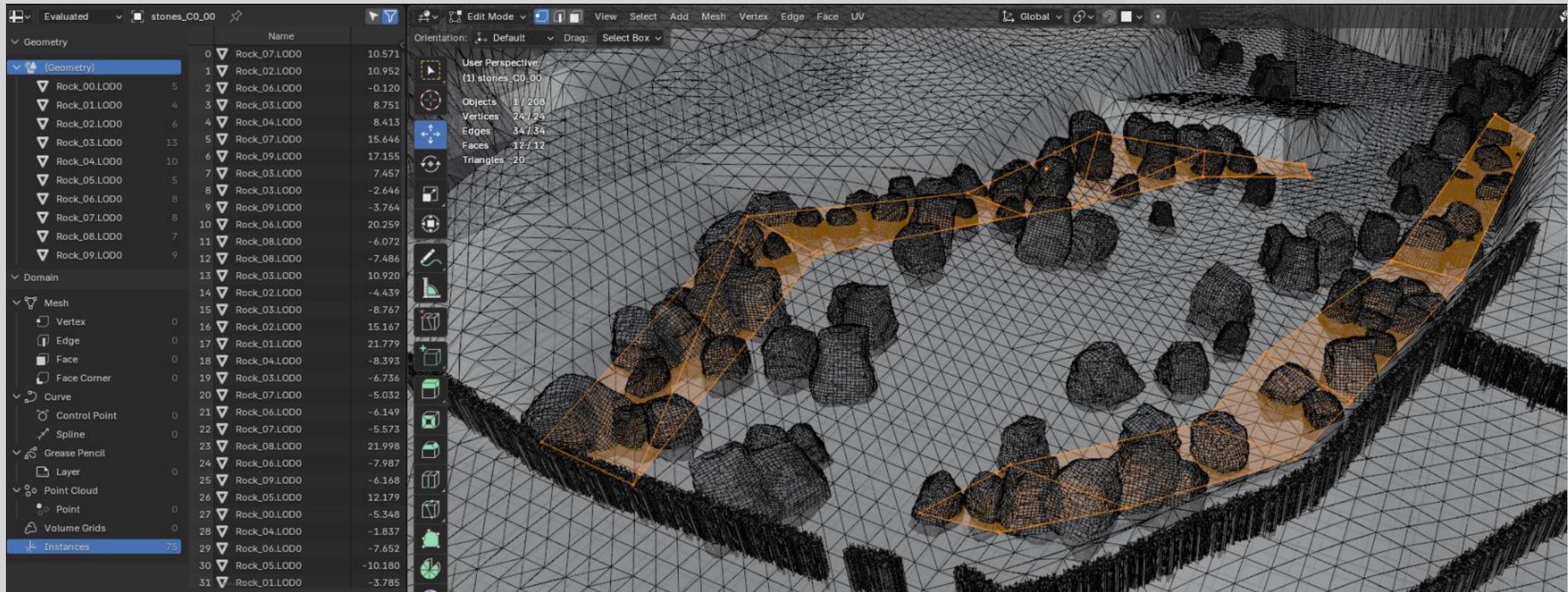


アセット制作効率化事例(プロシージャル)

インスタンス配置

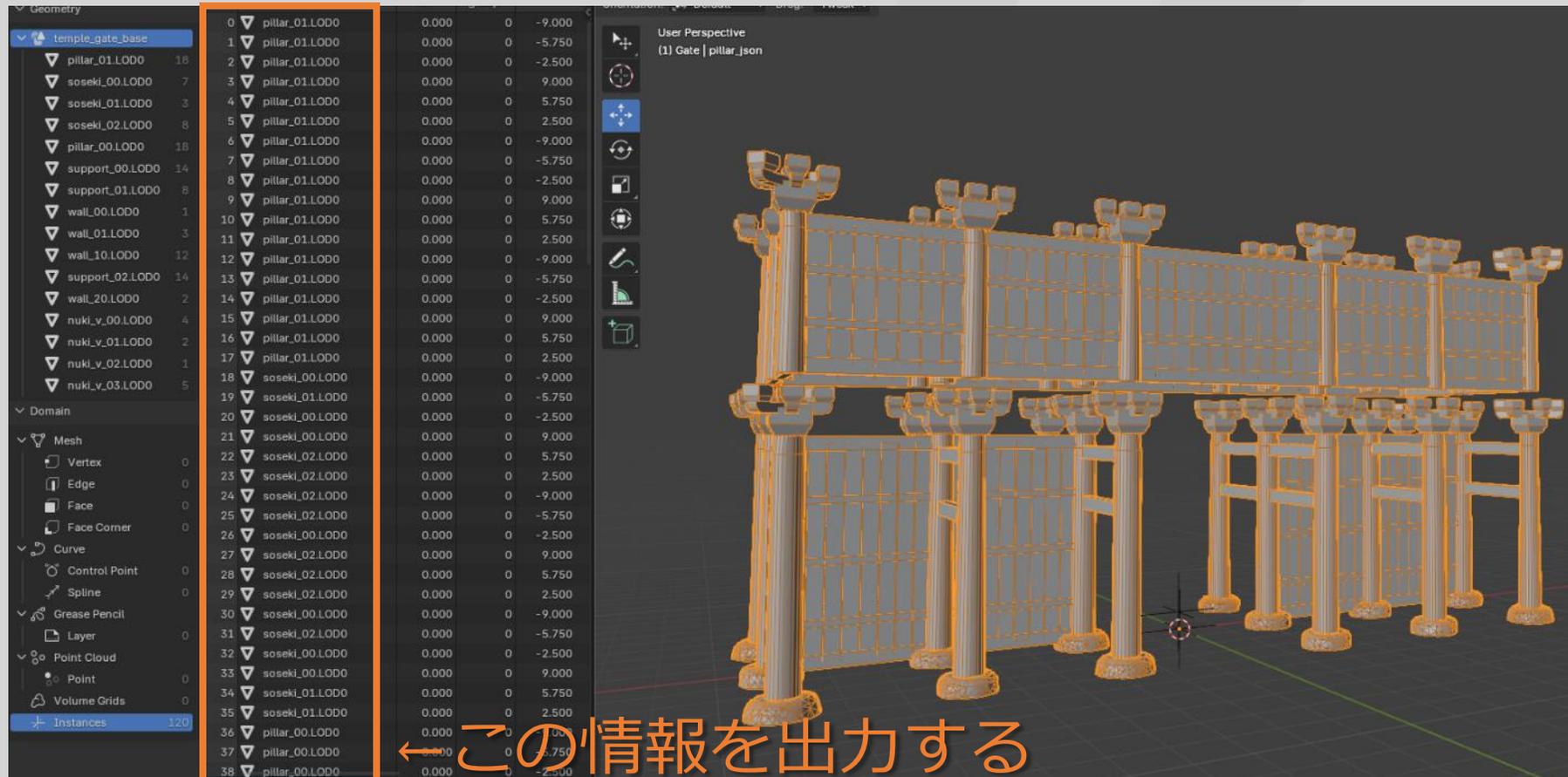
アセット制作効率化事例(プロシージャル)

こういう配置をUnityで再現するツールを作ったところ



アセット制作効率化事例(プロシージャル)

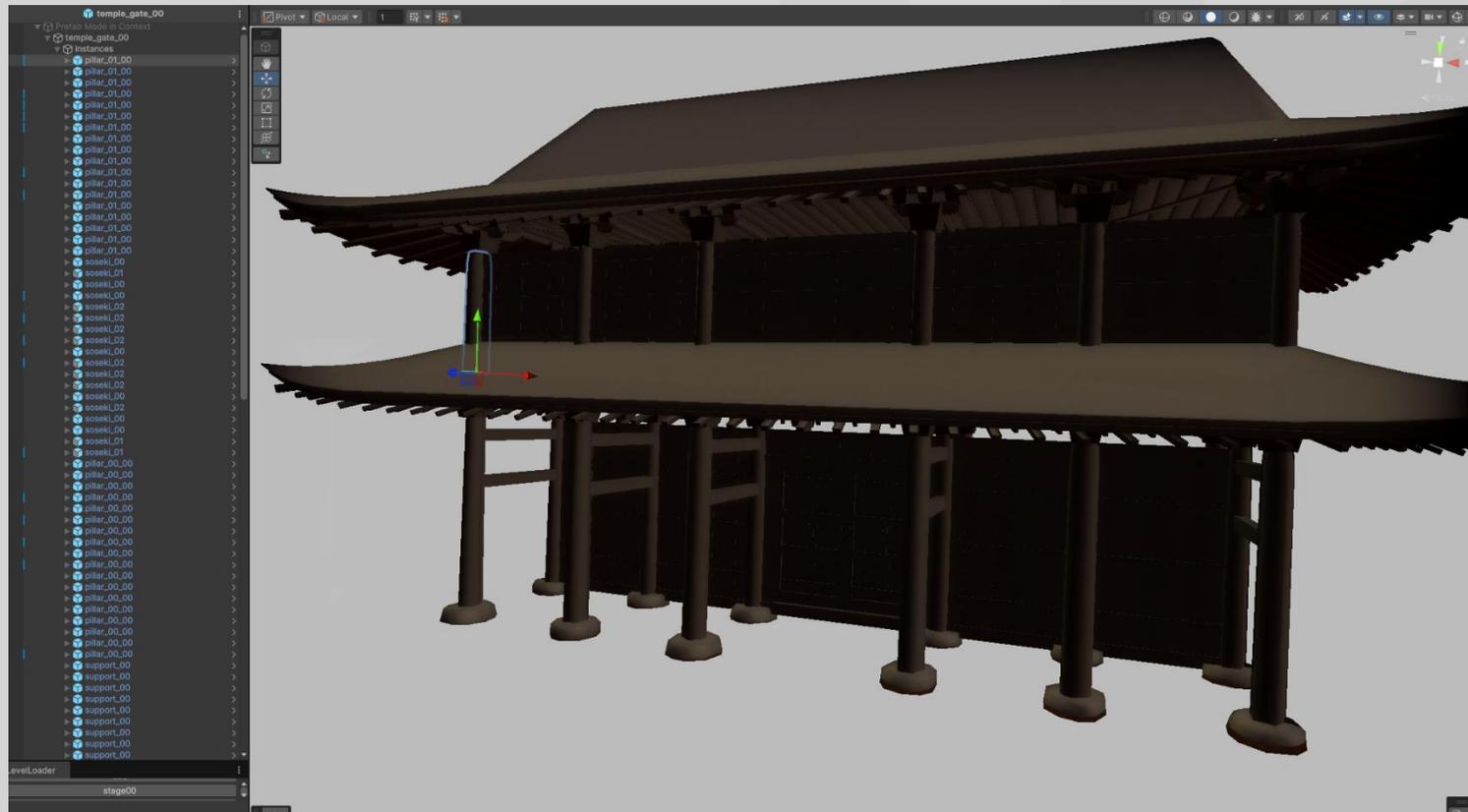
建物に転用できる事に気づく



アセット制作効率化事例(プロシージャル)

カリングやLODが細かく出来て幸せ

初期はモジュール組み立てツールを都度書いていた



アセット制作効率化事例(プロシージャル)

これらの効果は?

アセット制作効率化事例(プロシージャル)

GOOD

- 後から調整出来る事が強すぎる
- 規模が大きくなるほど効果的

BAD

- 全部プロシージャルで作ると時間が掛かる
荒い状態から置き換えていく事を心がける
- 習得難度が高い
- プロシージャル機能だけならHoudiniが最強

アセット制作効率化事例

テクスチャワークフロー

アセット制作効率化事例(テクスチャ)

- テクスチャペイントソフトは基本使用しない
- レベルデザインの変更で形状が変わる事前提
- UVは自動展開か無しで運用したい
- ゲーム上では解像度を十分に取りたい

アセット制作効率化事例(テクスチャ)

- HextilingとBiplanar mapping中心のマテリアル
- 基本的にアセットの材質のテクスチャだけを使用
- LOD0はポリゴン数多めにして形状のハイモデルからのノーマルマップベイクは基本使用しない

アセット制作効率化事例(テクスチャ)

この壺UVないです

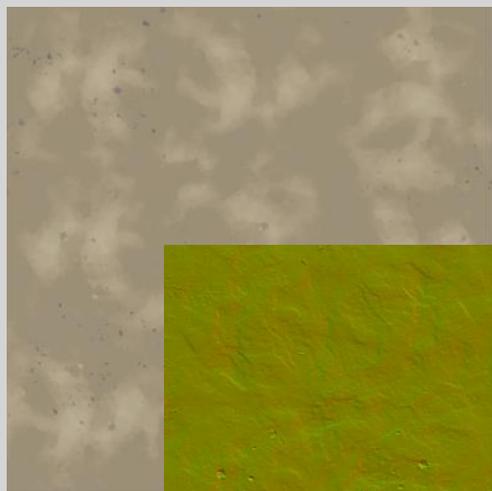


目一杯寄っても素材感を維持します

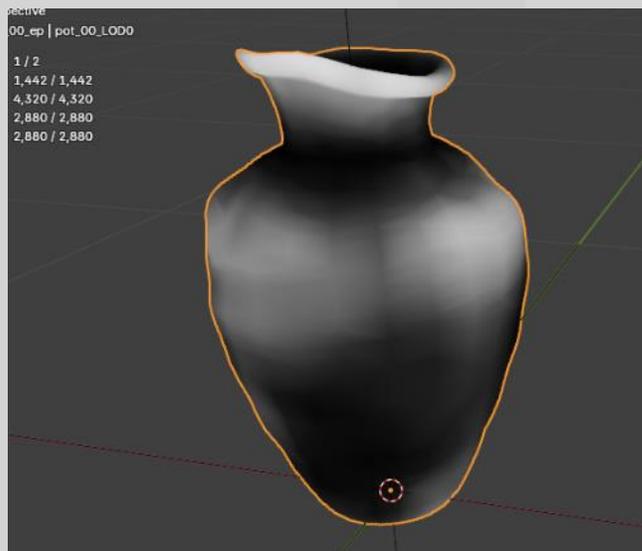


アセット制作効率化事例(テクスチャ)

頂点カラーでタイリング素材の色とノーマル・ラフネスを変更
頂点カラーもプロシージャル設定



×

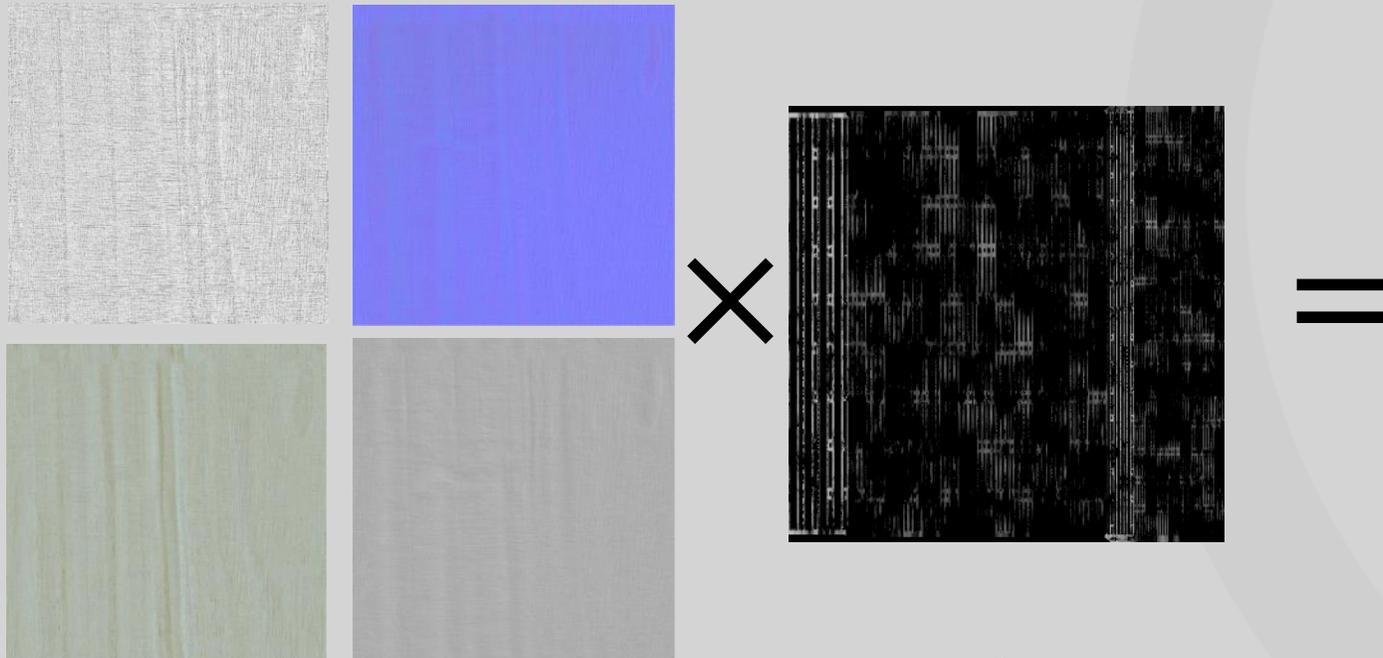


=



アセット制作効率化事例(テクスチャ)

素材の方向を合わせたりするときはUVを使う



縦方向の木目

カーバチュア
(Blenderで作成)



アセット制作効率化事例

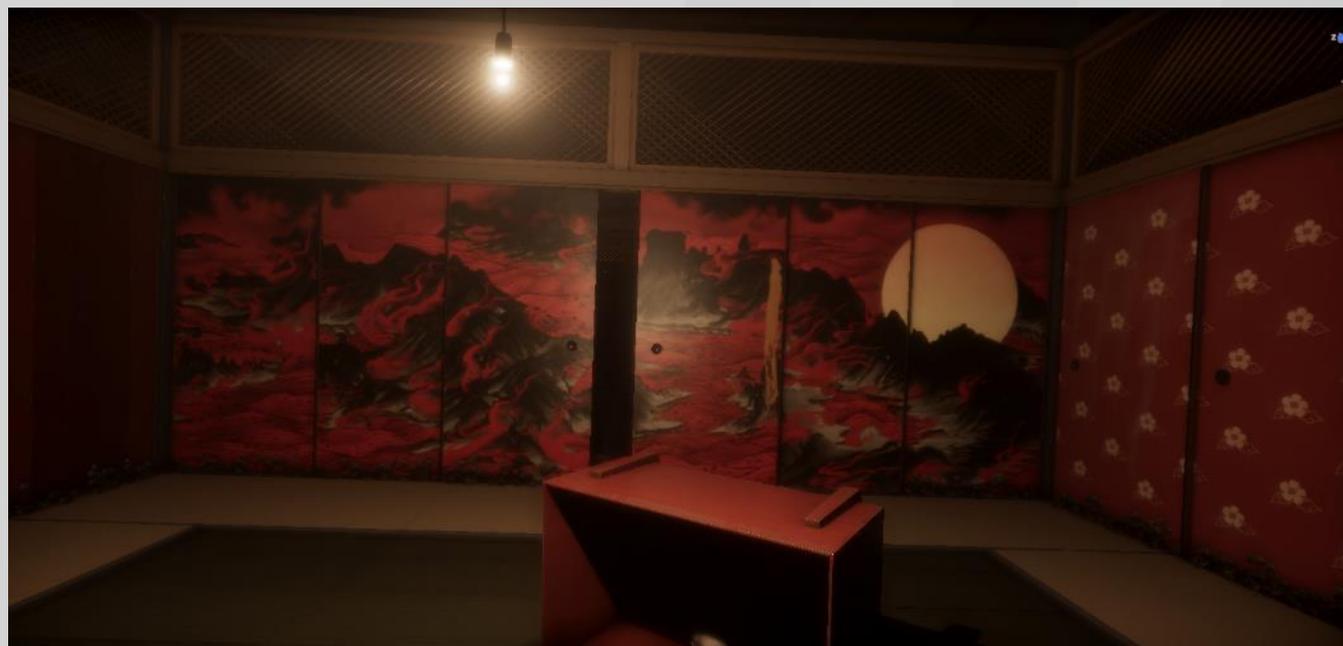
生成AIについて

アセット制作効率化事例(テクスチャ)

生成AIはプロトタイプで使用

社内規則的に製品に乗せられない

後でアーティストさんに書き直しをお願いした



アセット制作効率化事例(テクスチャ)

8192x2048というサイズの為2日強での書き直し



テーマは地獄



アセット制作効率化事例(テクスチャ)

これらの効果は?

アセット制作効率化事例(テクスチャ)

GOOD

- テクスチャ作成時間は劇的に短縮
- 同じ材質のものはどんどん使いまわせる
色とか艶とか変えたら案外分からない
- ゲームの配布サイズが小さくなる

BAD

- Blenderでプレビューできない
- 細かな作りこみは出来ないなので品質に限界あり

アセット制作効率化事例

因みに

アセット制作効率化事例

ライトベイク時間も勿体ないのでベイクなし
シャドウ解像度も低いのでSSAOで設置感づくり



アセット制作効率化事例

宣伝

アセット制作効率化事例

ゲームのリリース後に
一部アセットのストア販売を計画中





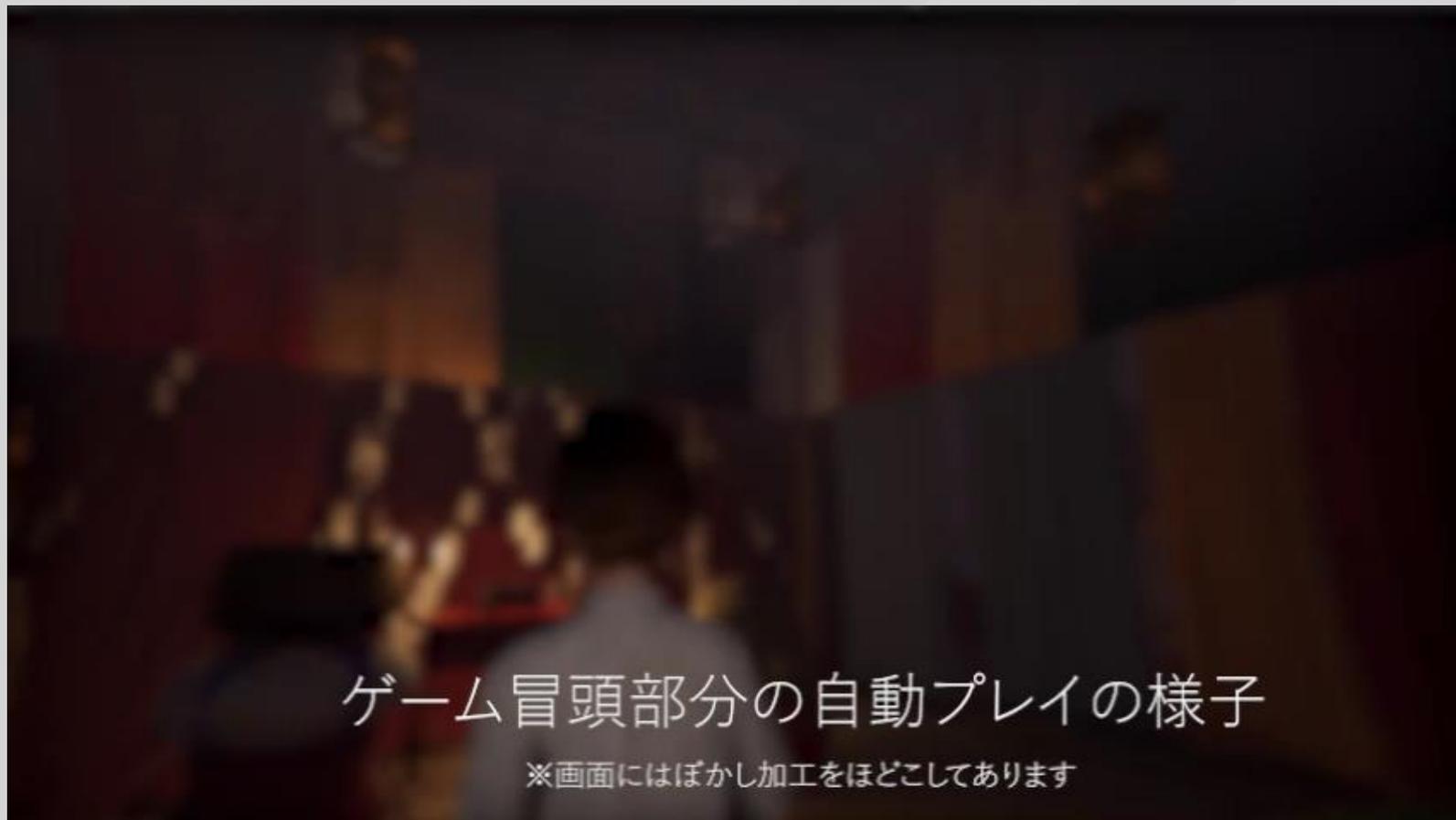
息抜きエンジニアリング

息抜きエンジニアリング

貴重な開発時間を使った戯れの紹介

息抜きエンジニアリング

自動プレイ



息抜きエンジニアリング

方式

- 移動情報とギミックアクセス情報を記録
(移動情報は方向傾きで間引く)
- ゲーム内で自動プレイモードか否かでプレイヤーとギミックの挙動が変化
- プレイヤーは自動プレイキューから移動先とギミックアクセスを再現
- ギミックは自動プレイ中は即座に成功を返す

息抜きエンジニアリング

目的

- コードやフラグ設定、コリジョン変更で進行不能が発生しないかのチェック

成果

- デバッグ込み実装にほぼ2日(16時間強)
開発中盤のゲームの実装状況で全ポイント巡回チェックに1時間掛かる
毎日運用で1ヶ月以降で実装コスト回収

息抜きエンジニアリング

今後の展望

- フレームレートの実出力によるパフォーマンス変動監視
- テスト成功後にSteam等のプラットフォームへの自動デプロイ

息抜きエンジニアリング

息抜きに火を焚く



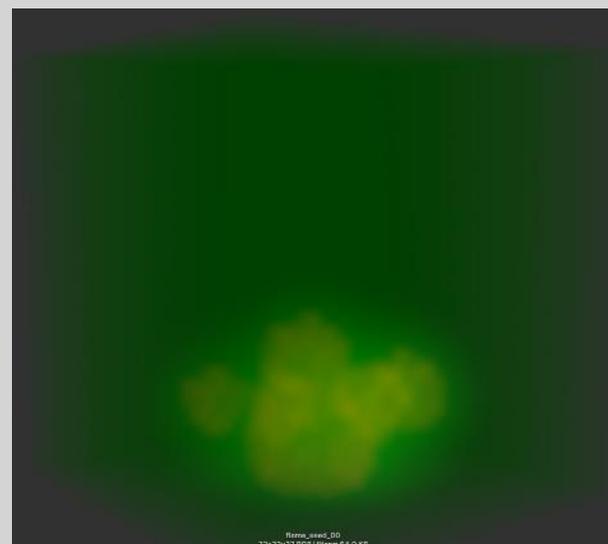
息抜きエンジニアリング

経緯

- パーティクルで炎作るの面倒
- 燃烧した気体の一定の温度を可視化したら炎になるのでは?

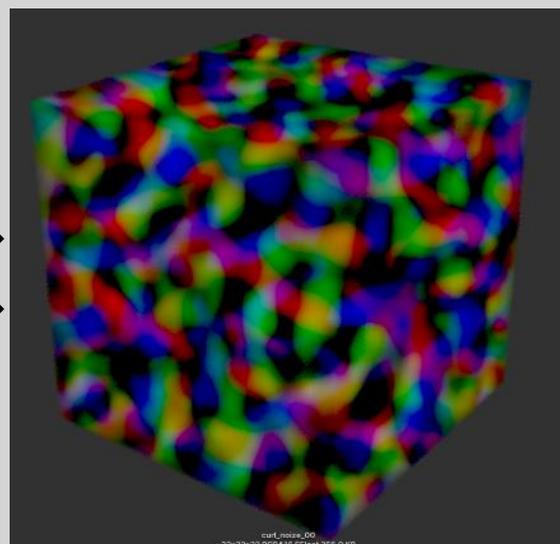
息抜きエンジニアリング

なった



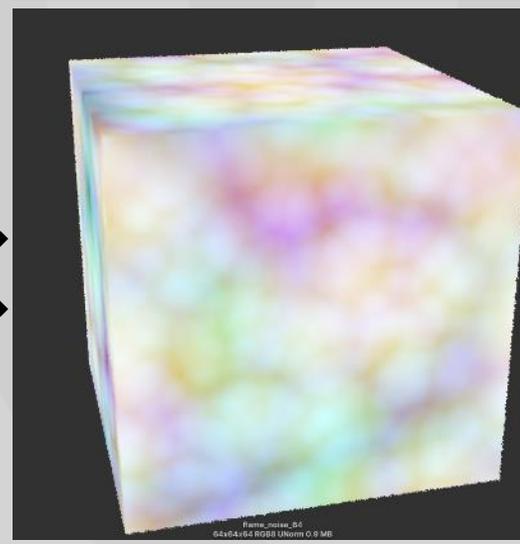
火種から熱を

×



カールノイズで動かし

×



Worley noiseでマスク

=



それっぽくなる

LUTで温度に合わせて色をつける

息抜きエンジニアリング

GOOD

- 篝火程度なら軽量
遠ければシミュレーション止めても気にならない
シミュレーションも3Dテクスチャの各ピクセル毎にベロシティの逆方向からエネルギーを減衰して流入させるだけ

BAD

- レイマーチングなのでメッシュの中にめり込むと消える
- 3Dテクスチャの解像度に制約をうける

息抜きエンジニアリング

息抜きは楽しい

息抜きエンジニアリング

途中GitHub Copilotが導入されコーディング時間がかなり短縮された
体感1/4位コーディング時間が節約できるので低リスクで息抜きが出来て幸せ



総評

総評

開発者視点

- 前提条件次第で新しい制作手法を使って大幅に工数削減が見込める
- より高品質化して生産性を向上するにはまだ工夫が必要、そのコストに見合う価格を消費者が払いたいかも含めたプロジェクト計画が必要
- 生成AIからの直接的な成果物は社会情勢次第なのでアセット制作が楽になるのはまだ先の話
- 作れる事と売れる事は別なのが世知辛い

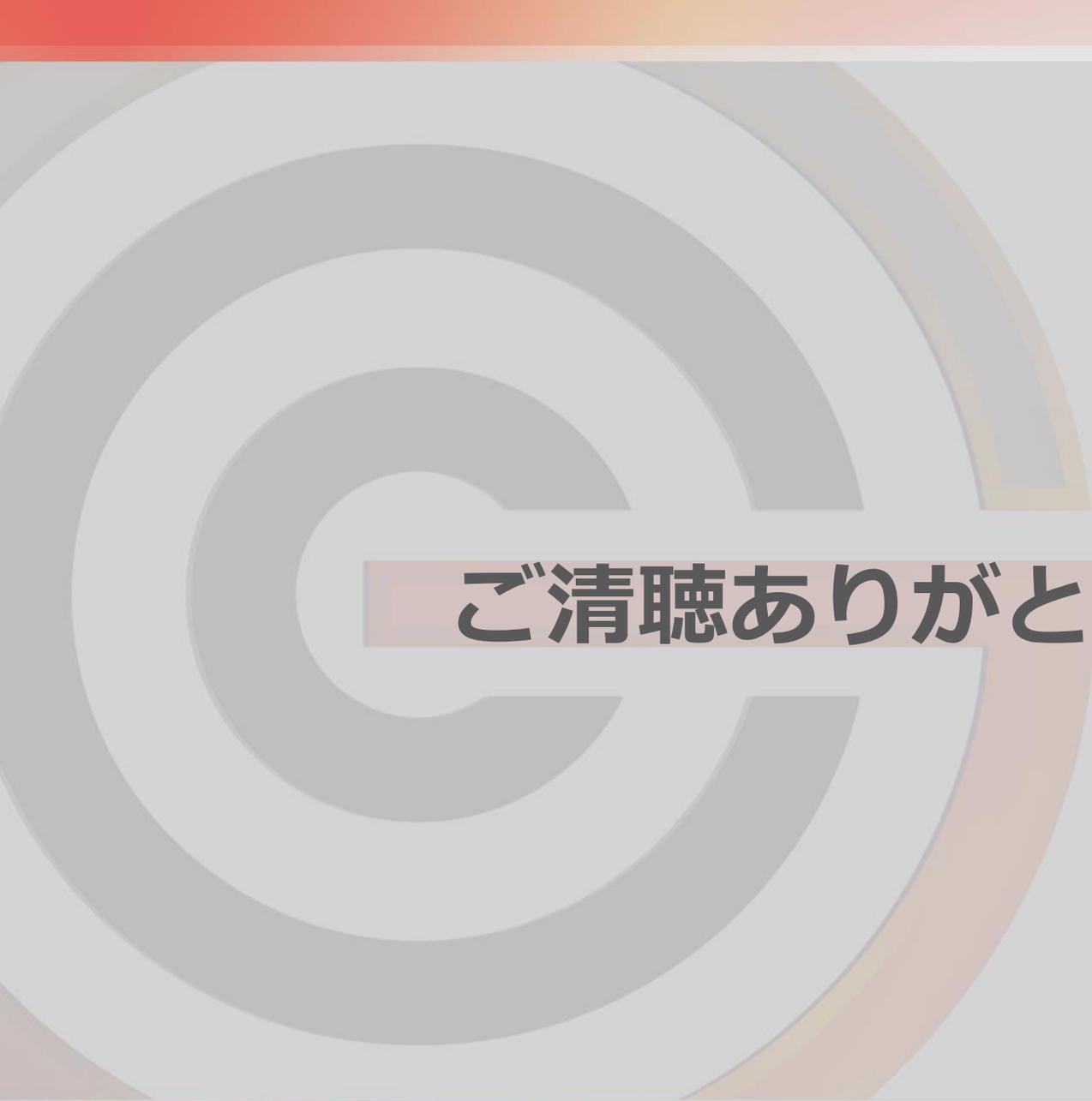
時間があれば

質疑応答

ウィッシュリスト登録お願いします



https://store.steampowered.com/app/3034350/_/



ご清聴ありがとうございました