

Procedural Hard Surface Design

With HOUDINI

POLYPHONY DIGITAL INC.

AKIRA SAITO

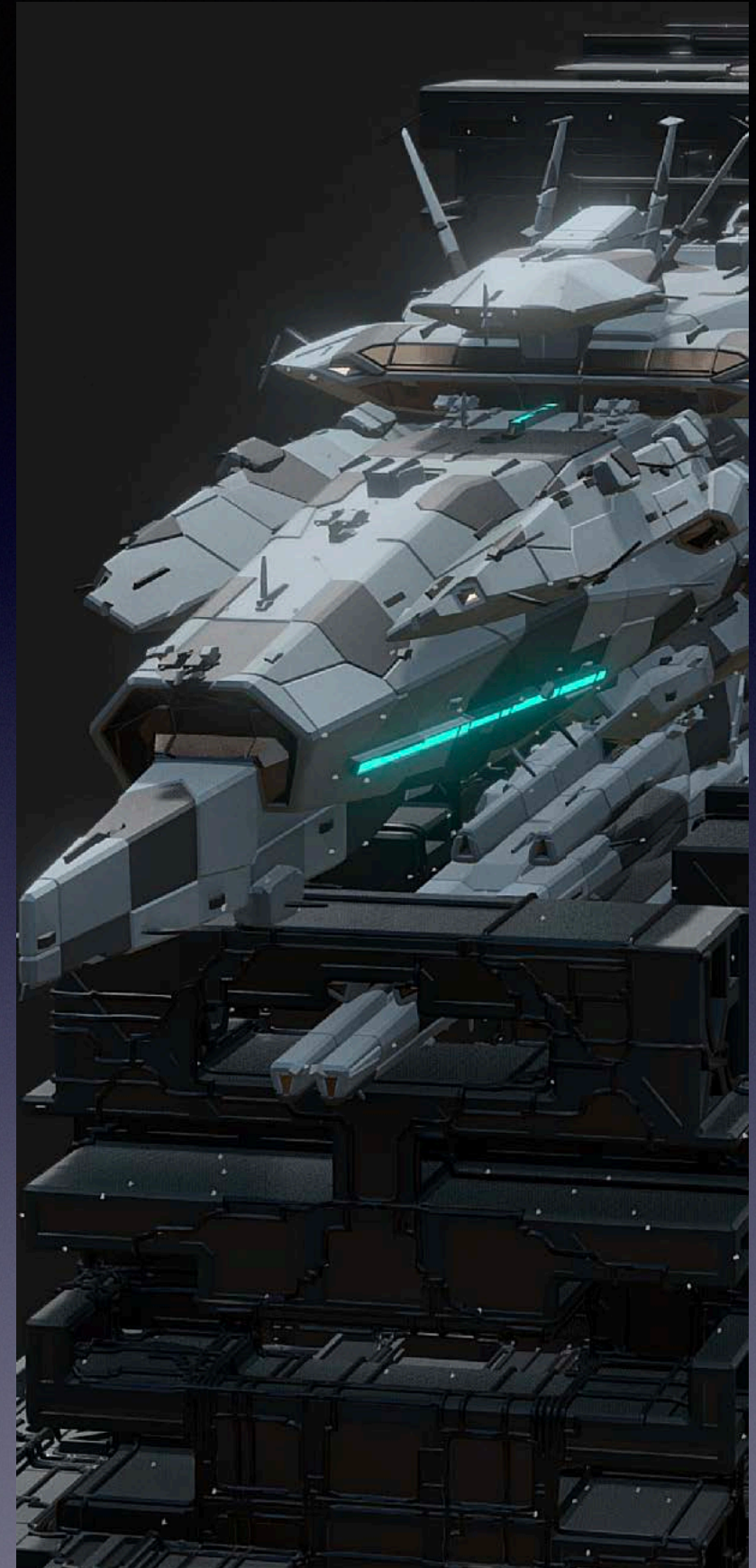
自己紹介 齋藤 彰

- ・株式会社ポリフォニー・デジタル所属
- ・景観デザインチーム アーティスト
- ・GRAN TURISMOシリーズの景観(コース)を作成しています
- ・CEDEC2017
Houdini16にHDAを2つ追加したらビルが量産できた
(プロシージャルモデリング事始め)



今日お話しする事

- ・Houdiniによる
Procedural Hard Surface Design(について
(Hard Surface:ロボットや乗り物など)
- ・始めたきっかけ(目的・条件)
- ・ワークフローの基本を紹介
- ・ワークフローの具体的な例を紹介
- ・まとめ



目的

- ・個人的な趣味
- ・Houdiniの習得(特にモデリング)
- ・形状生成アルゴリズムの研究
Generative Designへのアプローチ
- ・SNSに成果物を公開することで
多くの人たちとの情報交換

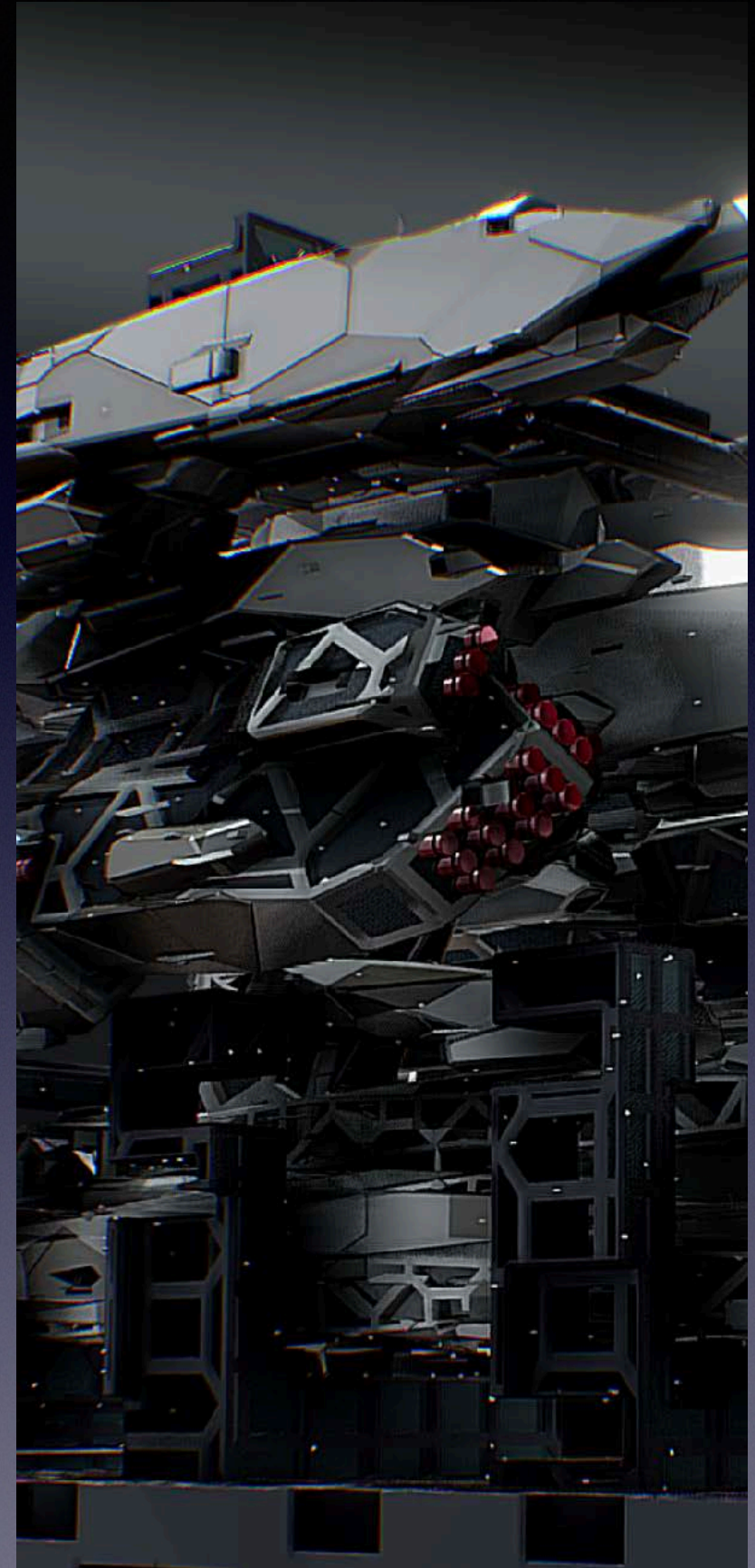


条件1

- Houdini Indie 16.5
- ポリゴン
- シミュレーションは使用しない
- VOPは使用しない。VEXとPythonを使う。
- 選択ツールは使用しない
- switch SOPのエラー回避処理

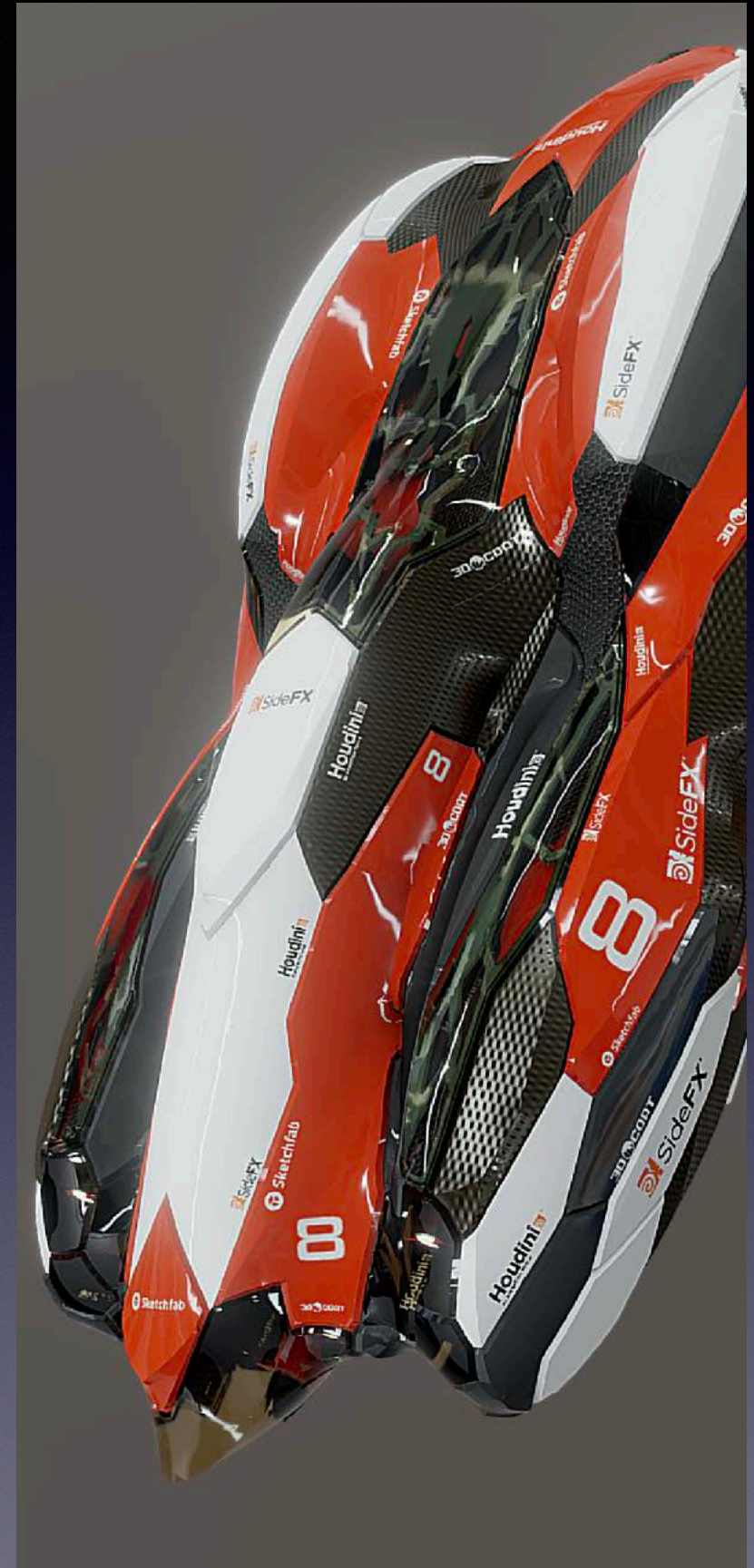
```
if (strmatch("*Error:*", run("opinfo " + opfullpath("../" + opinput(".", 1)))), 0, 1)
```

は使用しない。
- 3D COATによるUV展開とペイント
- SketchFabによるマテリアル設定とレンダリング



条件2

- ・Mac Book Pro (15-inch 2017)
3.1GHz Intel Core i7
16GB
- ・2017年の12月ごろから
- ・1日1～2時間程度
- ・ツール実装(Houdini) 30～60分
- ・新機能追加は1週間に1～2回
- ・モデリング(Houdini) 10～30分
- ・ペイント(3D COAT) 30分
- ・マテリアル設定 レンダリング(Sketchfab) 10分



HoudiniからSketchfabへ直接モデルをアップロードする機能が Game Development Toolset に追加されました



Export to Sketchfab Directly from Houdini

< BACK TO OVERVIEW

NEW ON SKETCHFAB · 14 June 2018

Within the world of computer graphics, **Houdini** is well known as a visual effects solution for blockbuster films. At the same time, it is now being used by more and more game studios to create game art and run time effects. Houdini's procedural nature is controlled using an artist-friendly node-based workflow that makes it easy to create multiple iterations of an idea or make changes even deep into production.

Lately, SideFX has been putting a lot of work into its modeling tools and has taken things one step further: there is now a tool available for exporting directly to Sketchfab



To start exporting, you can add the Sketchfab exporter node using the Game Development Tool Shelf. The **Game Development Toolset** is a set of bonus tools for Houdini that can be downloaded from within Houdini 16.5.

Houdini Ship by Akira Saito

Share this article



About Sketchfab

Enter new dimensions. With a community of millions of creators, we are the largest platform to publish and find 3D models online.

JOIN FOR FREE

More Sketchfab News



Import Sketchfab models directly into Blender (Beta)

6 days ago



Introducing MASSIVE by Sketchfab (Alpha)

7 days ago

Procedural Hard Surface Design

With HOUDINI

作例紹介(時系列)



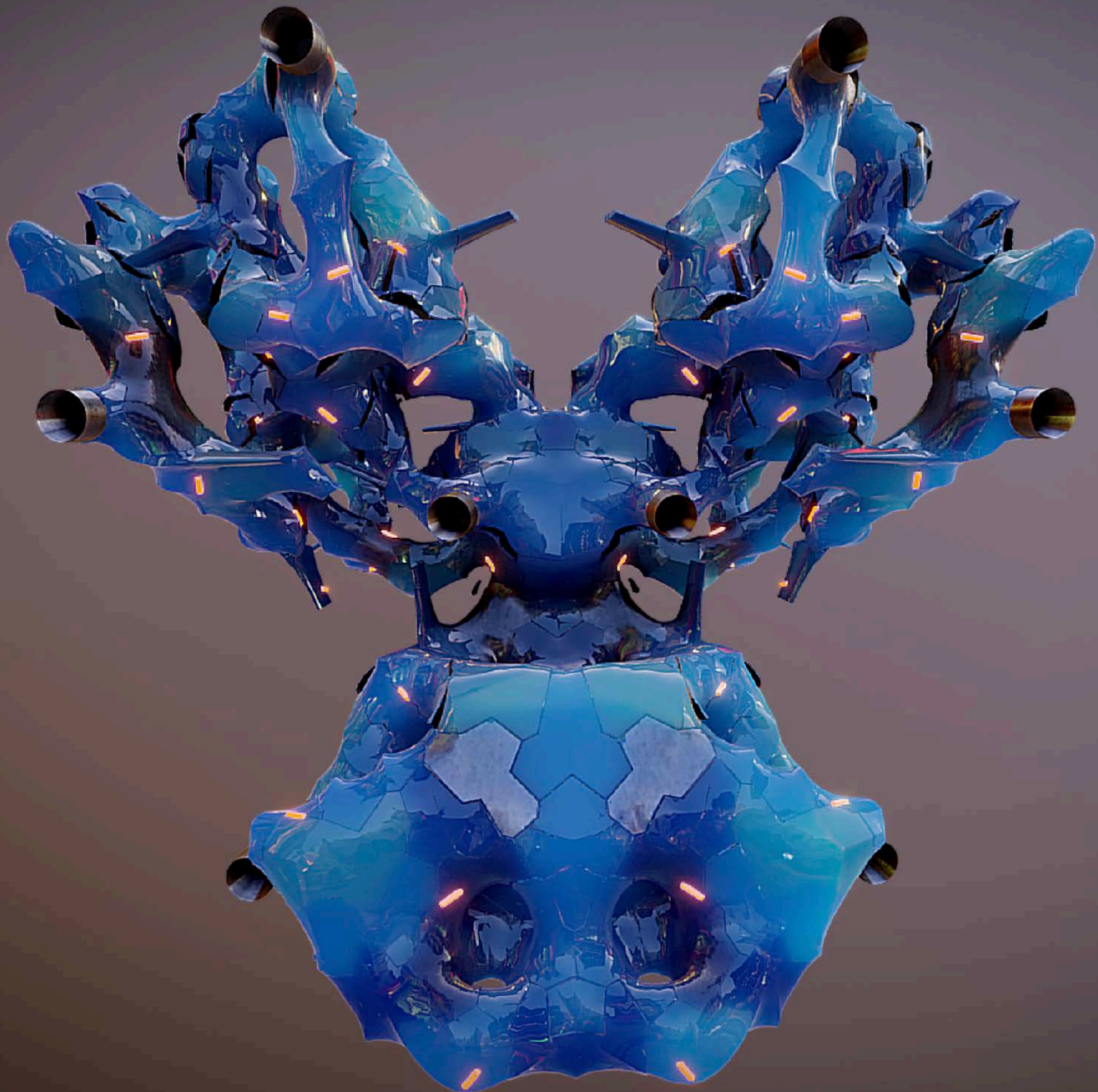


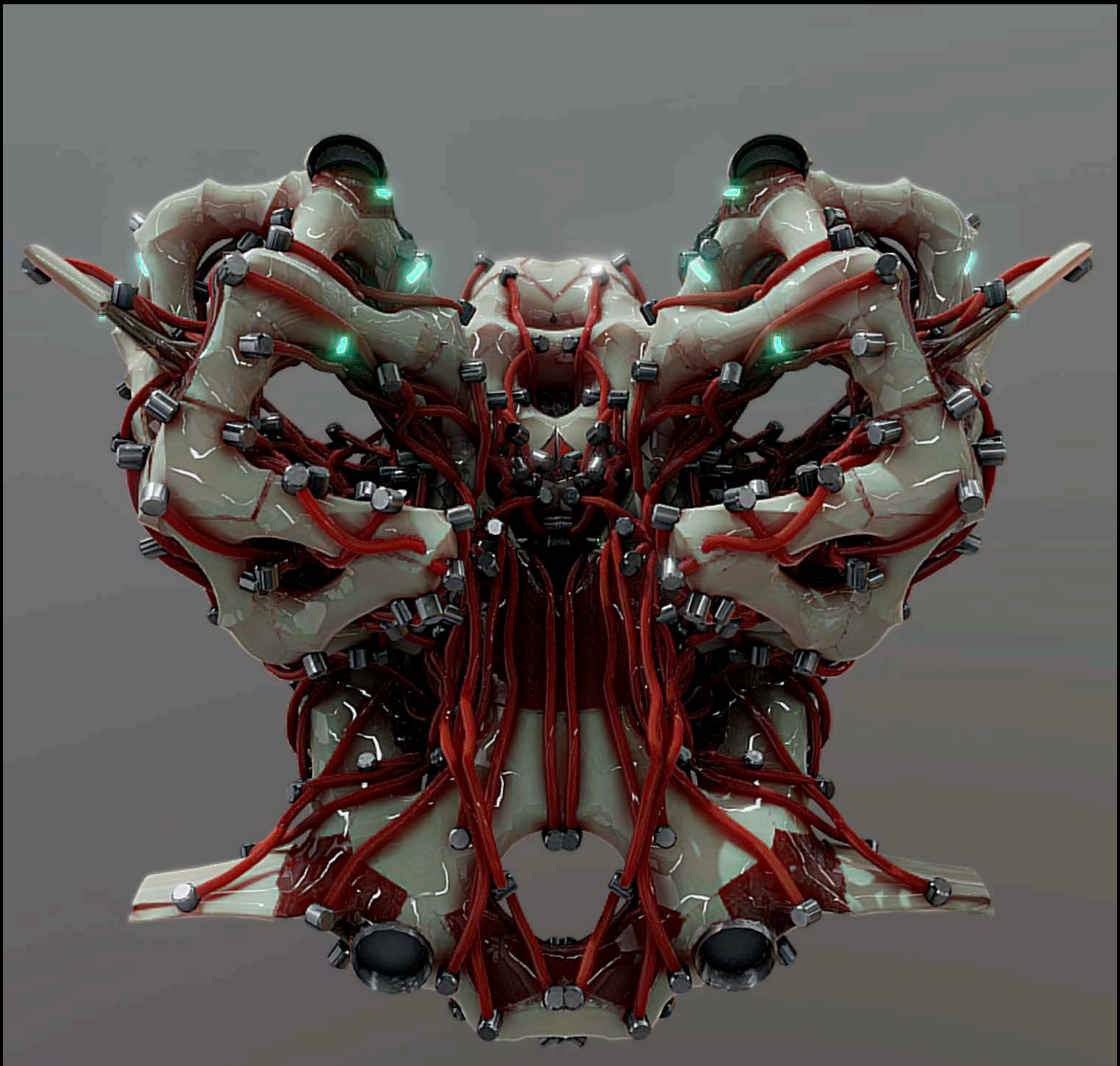




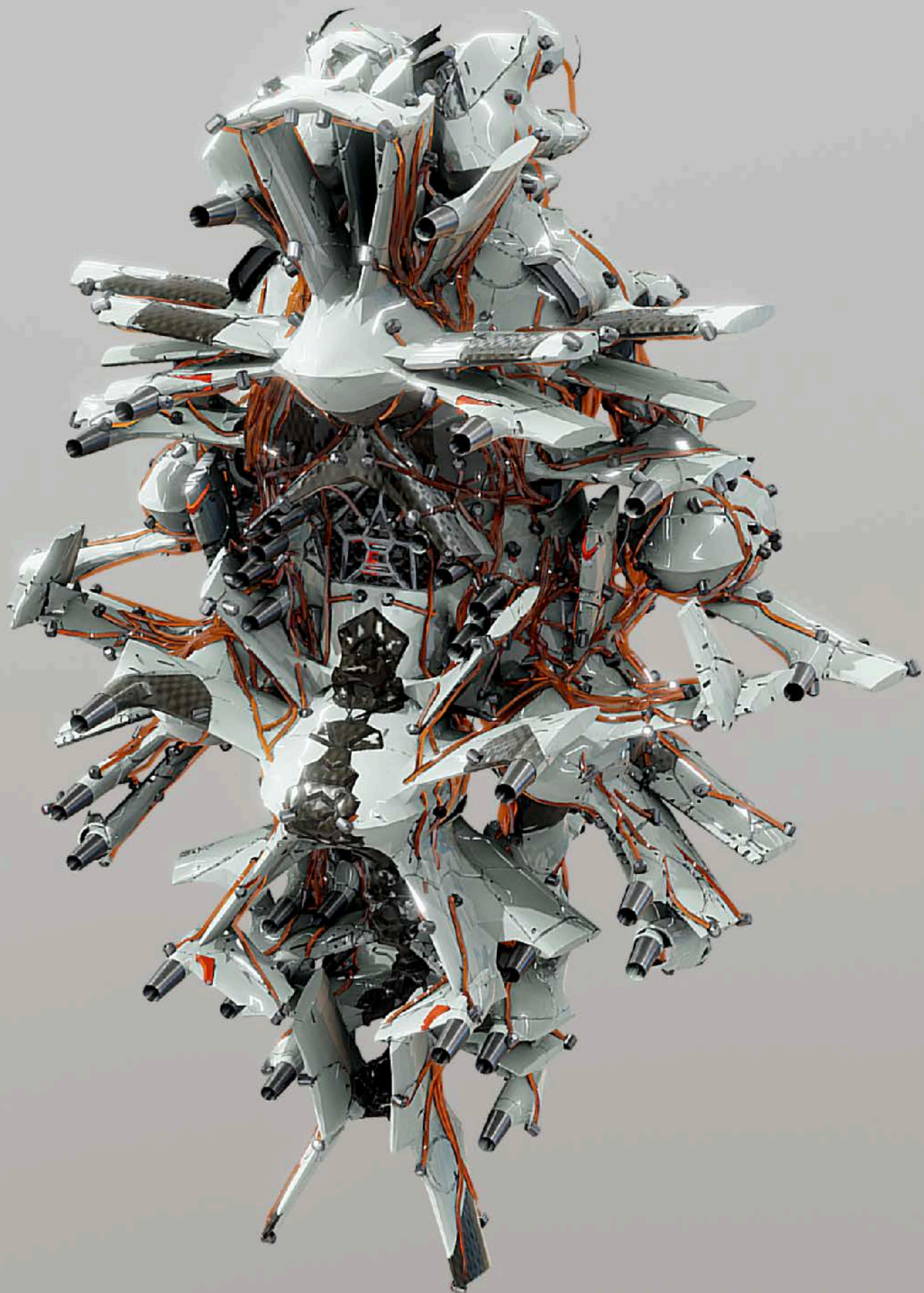


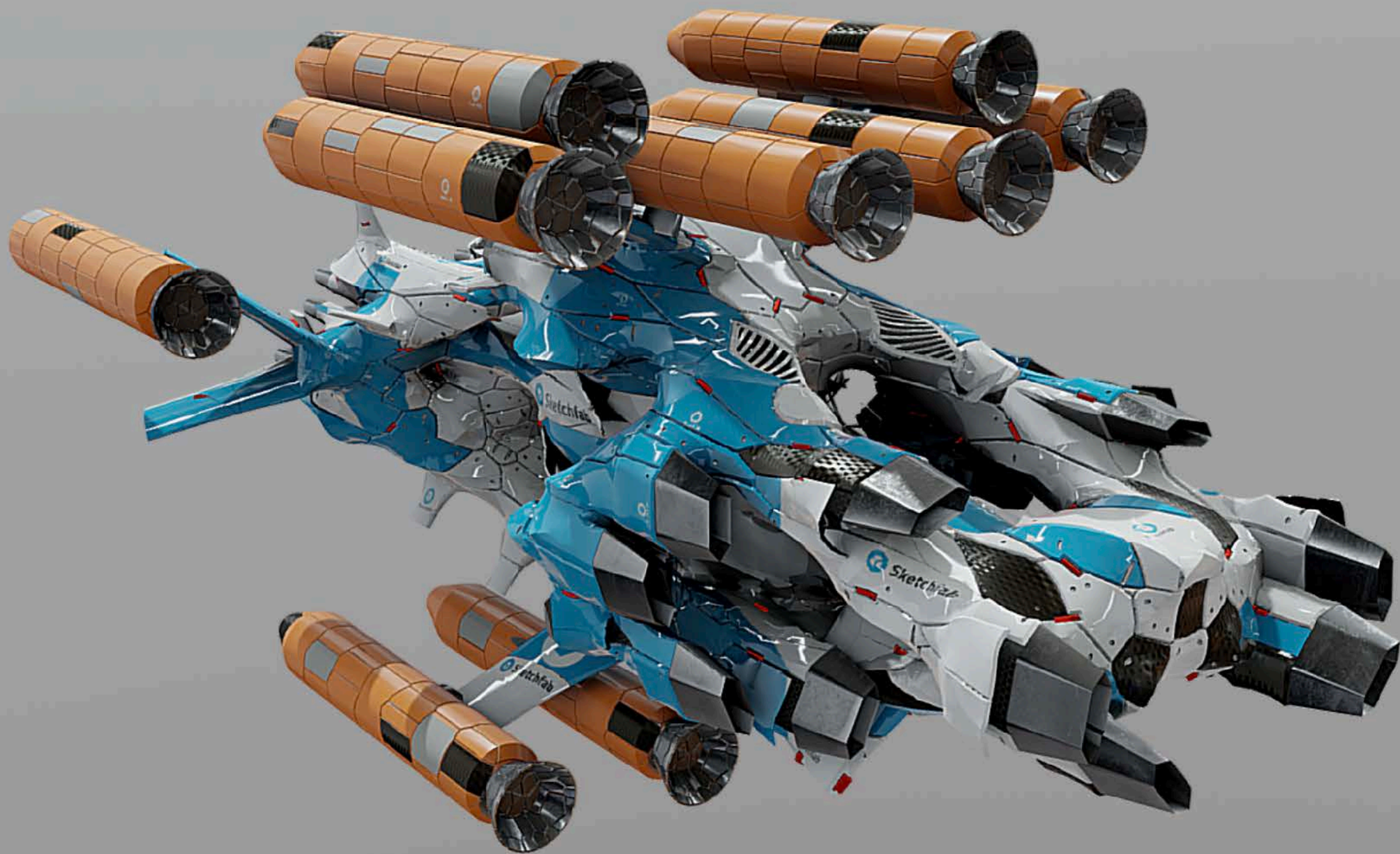


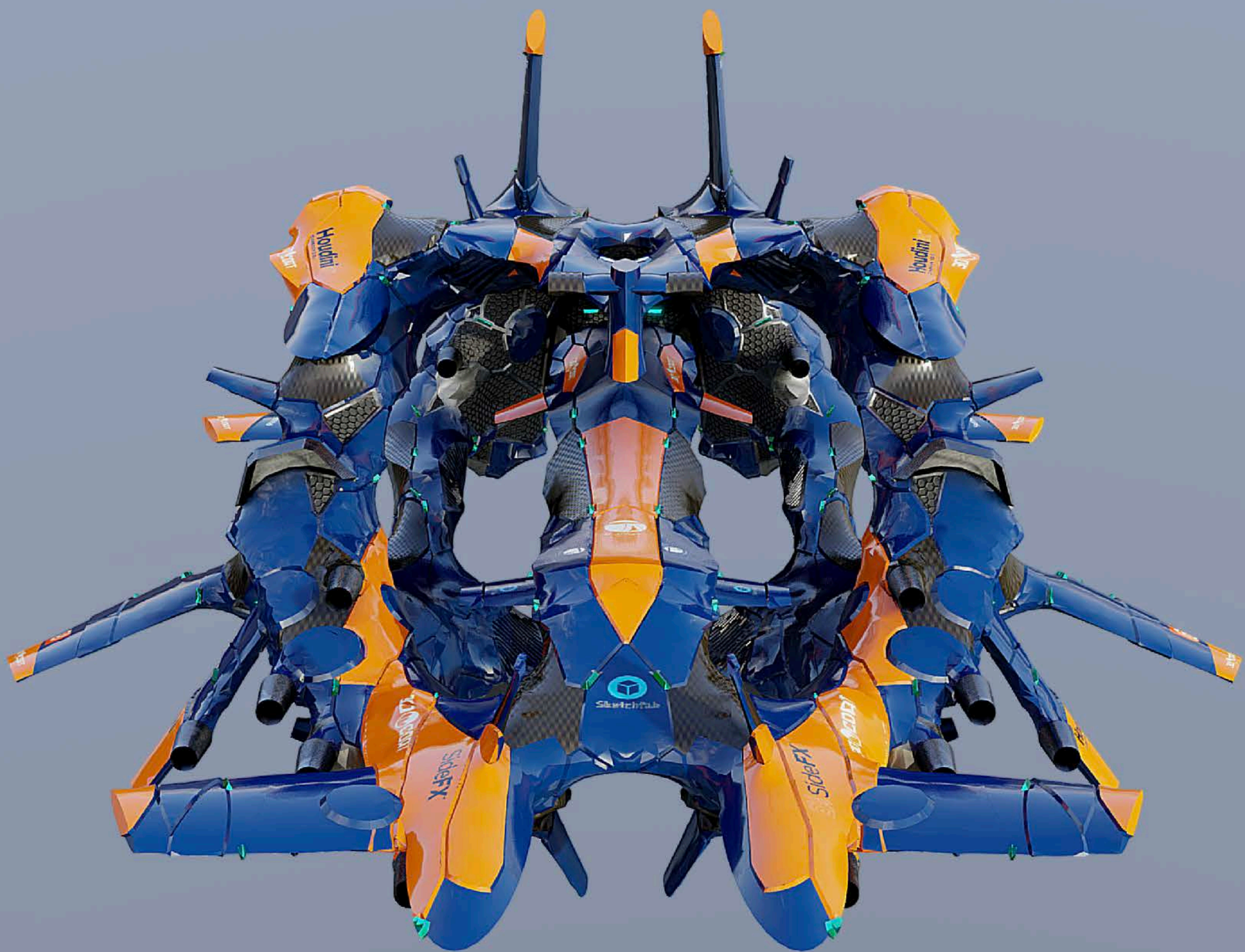










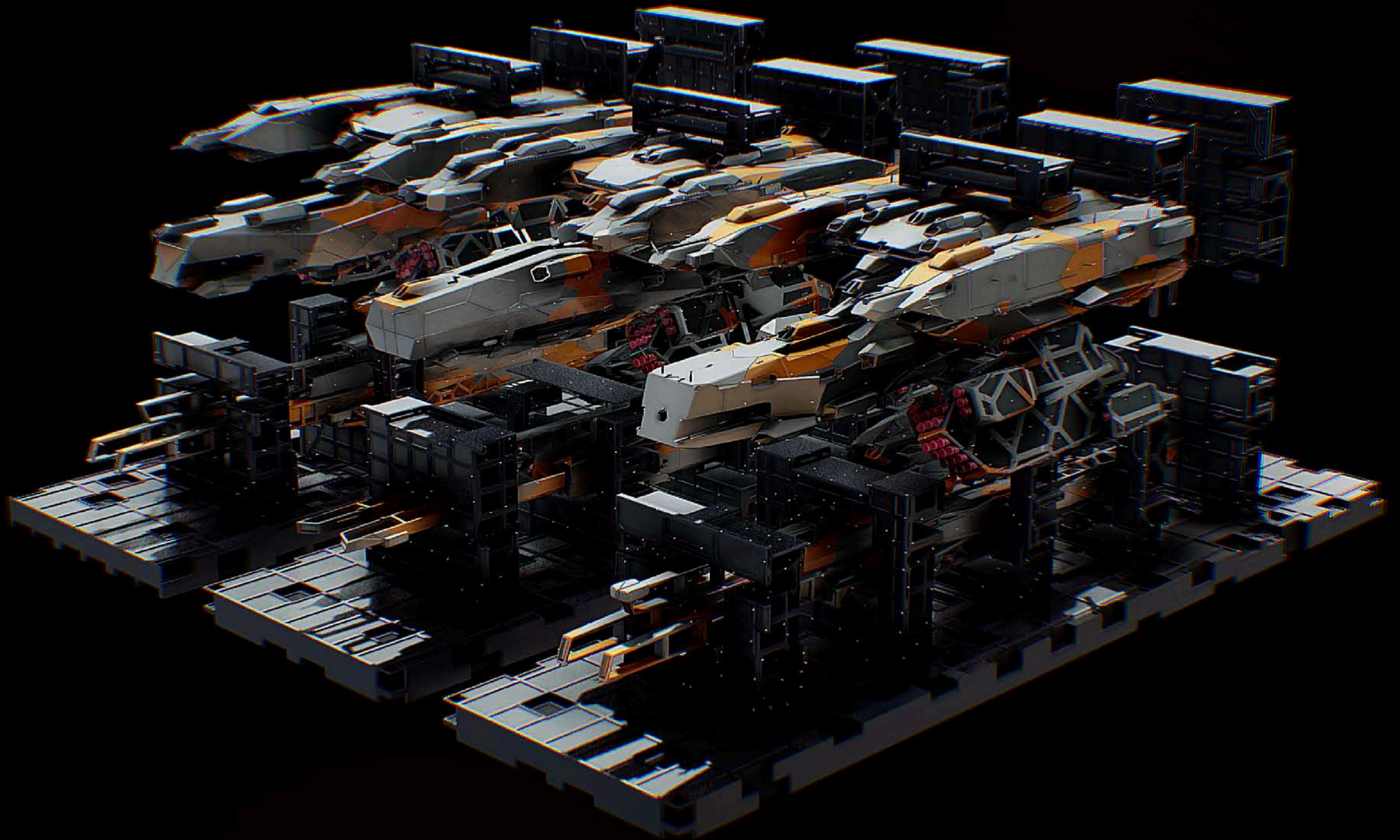


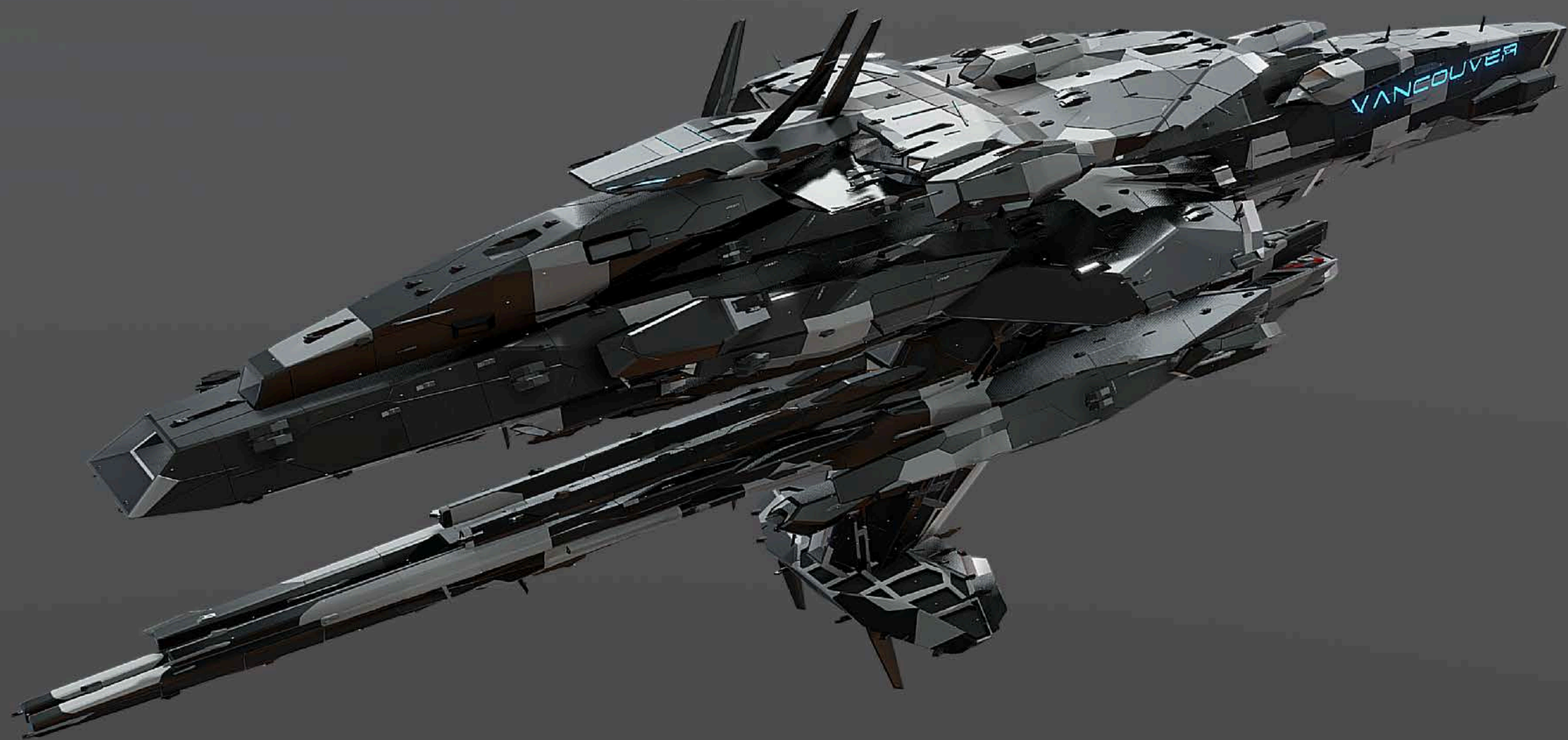




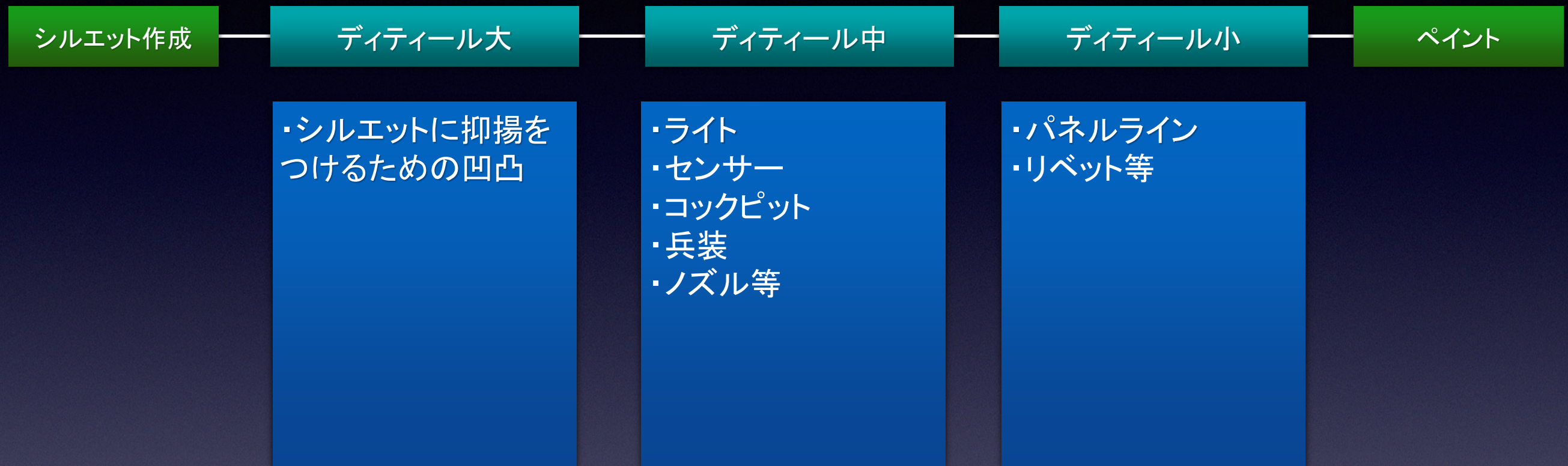




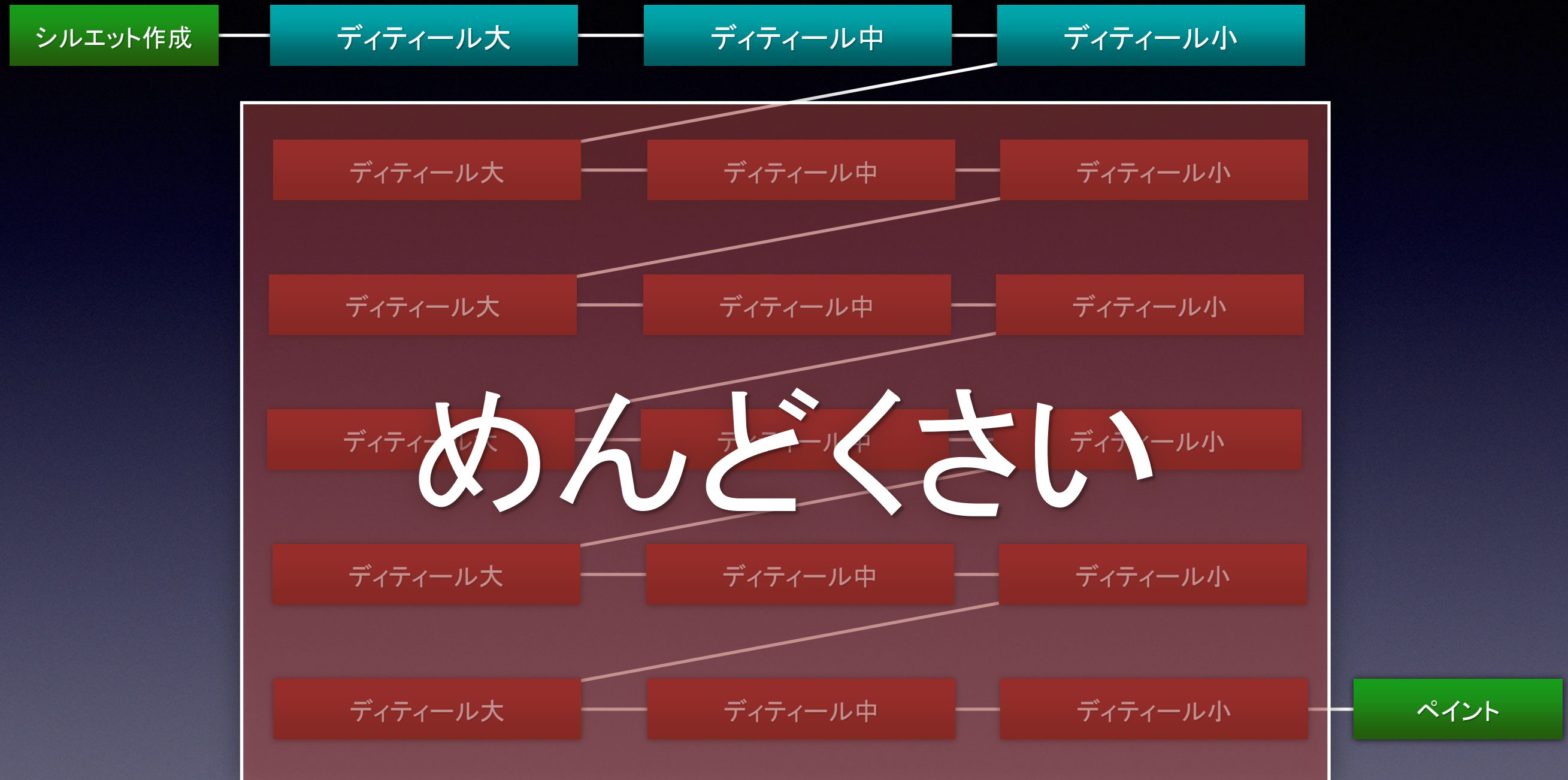




齋藤が今まで行っていた、非プロシージャルハードサーフェースモデリングワークフロー (Maya ZBrush SketchUp等)



齋藤が今まで行っていた、通常のハードサーフェスモデリングワークフロー



「めんどくさい」ということは

- ・必要な作業が具体的にわかっている。
- ・かつ、その時間的見積もりが出来ている。
- ・繰り返が多い。



「めんどくさい」への対応

- ・その作業そのものの見直し
- ・ショートカットキーの活用
- ・スクリプトによる自動化
- ・専用ツールの作成

それでも「めんどくさい」

- ・作業の全自動化(プロシージャル化)

目的や、その実装スキルから、
ツール作成の延長上にプロシージャル化が存在する。



「プロシージャル化」とは

・「プロシージャル法とは、ルールを定義し作業をすることで、自動化やデータ量の少量化を促し、より多くの反復や試行錯誤を通して異なる可能性やバリエーションの提供、その結果、エラーの低減、土壇場での修正の痛みの緩和、少人数での高生産性、高品質性を促進します。」

(本セッション紹介文より)

・一般的にはあらかじめ定義された手続きを元に工程を処理していく手法と説明されている。

・その工程が全自動、もしくは半自動で処理出来るように、ツール自体が与えられた形状や条件を判断し、パラメータや機能そのものの内容を動的に変化させる事。



齋藤が今まで行っていた、通常のハードサーフェースモデリングワークフロー

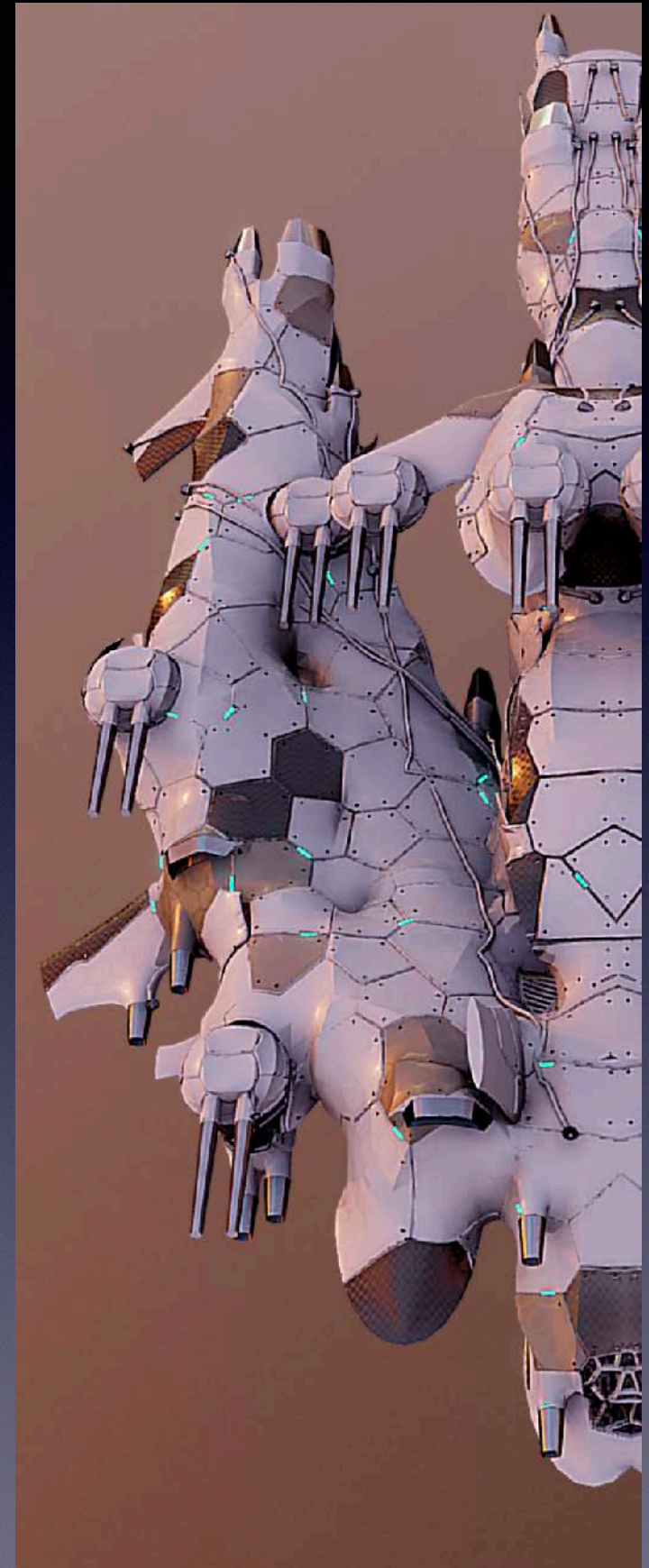
シルエット作成

プロシージャル化

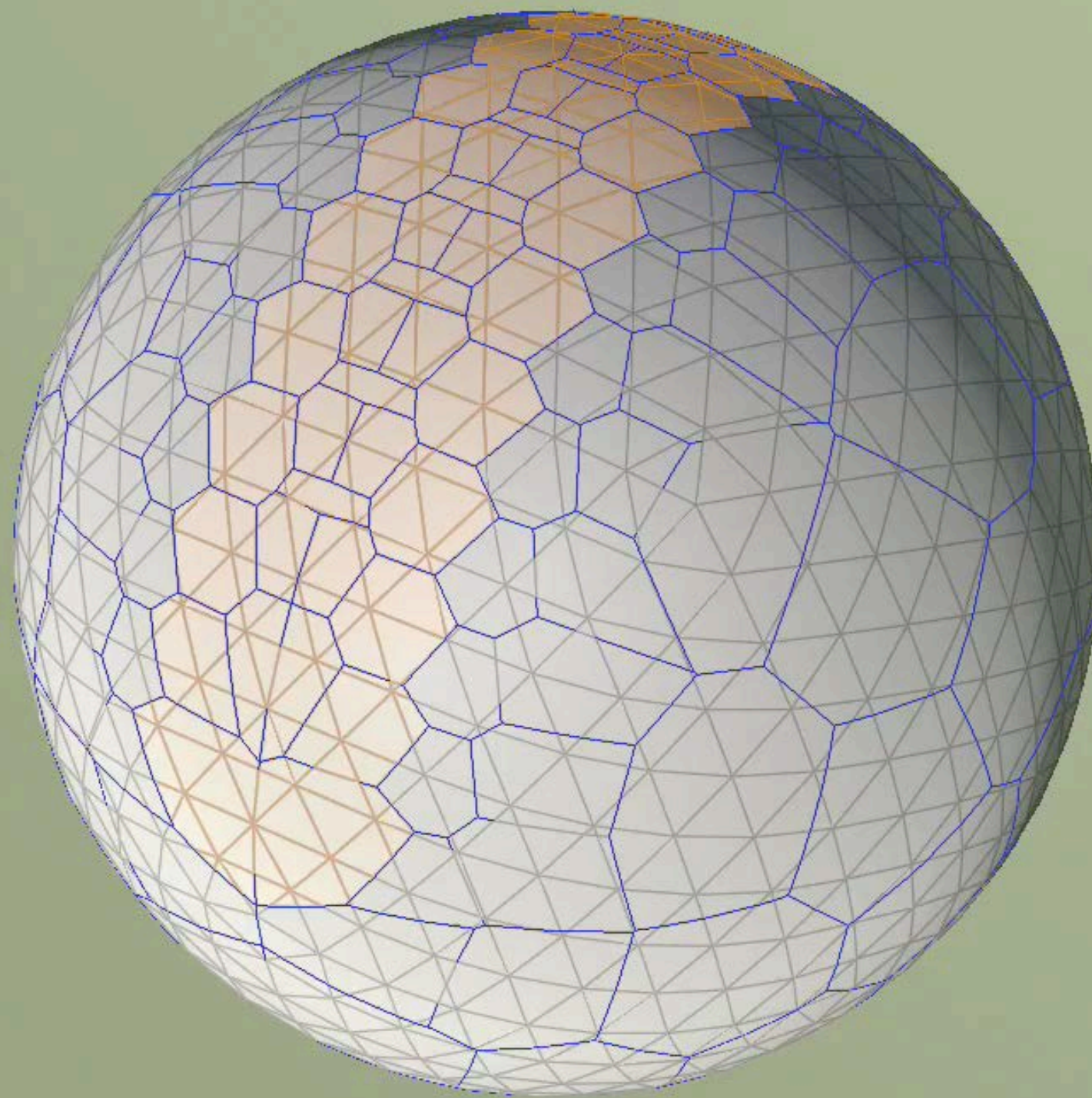
ペイント

プロシージャル化した主な工程

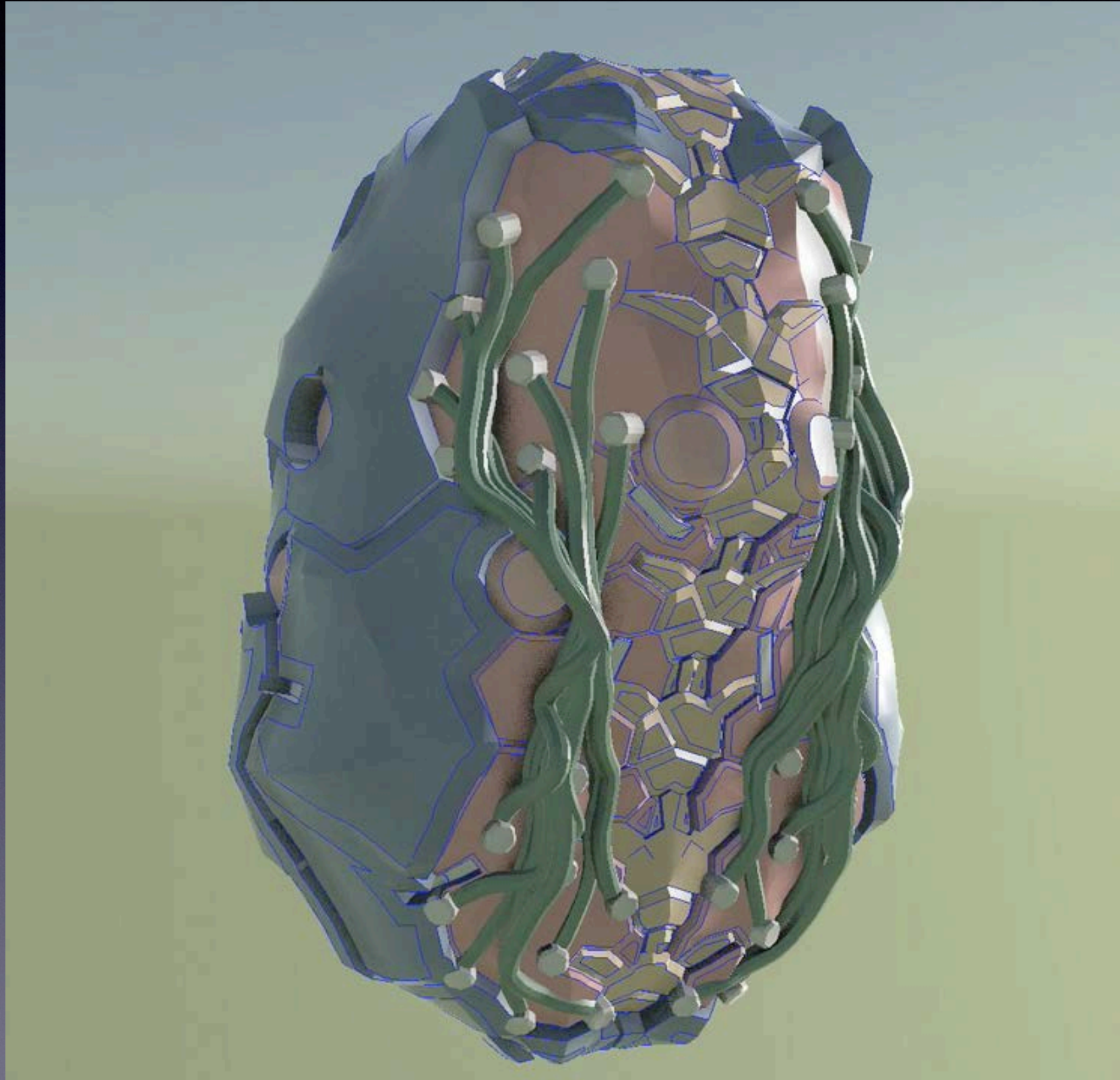
- ・ 生成系
 - 基本形生成(平面)
 - 基本形生成(曲面)
- ・ 配置系
 - ケーブル配置
 - エンジンノズル・エアインテーク配置
- ・ 加工系
 - パネルライン追加
 - エッジライン追加
 - トラス構造追加



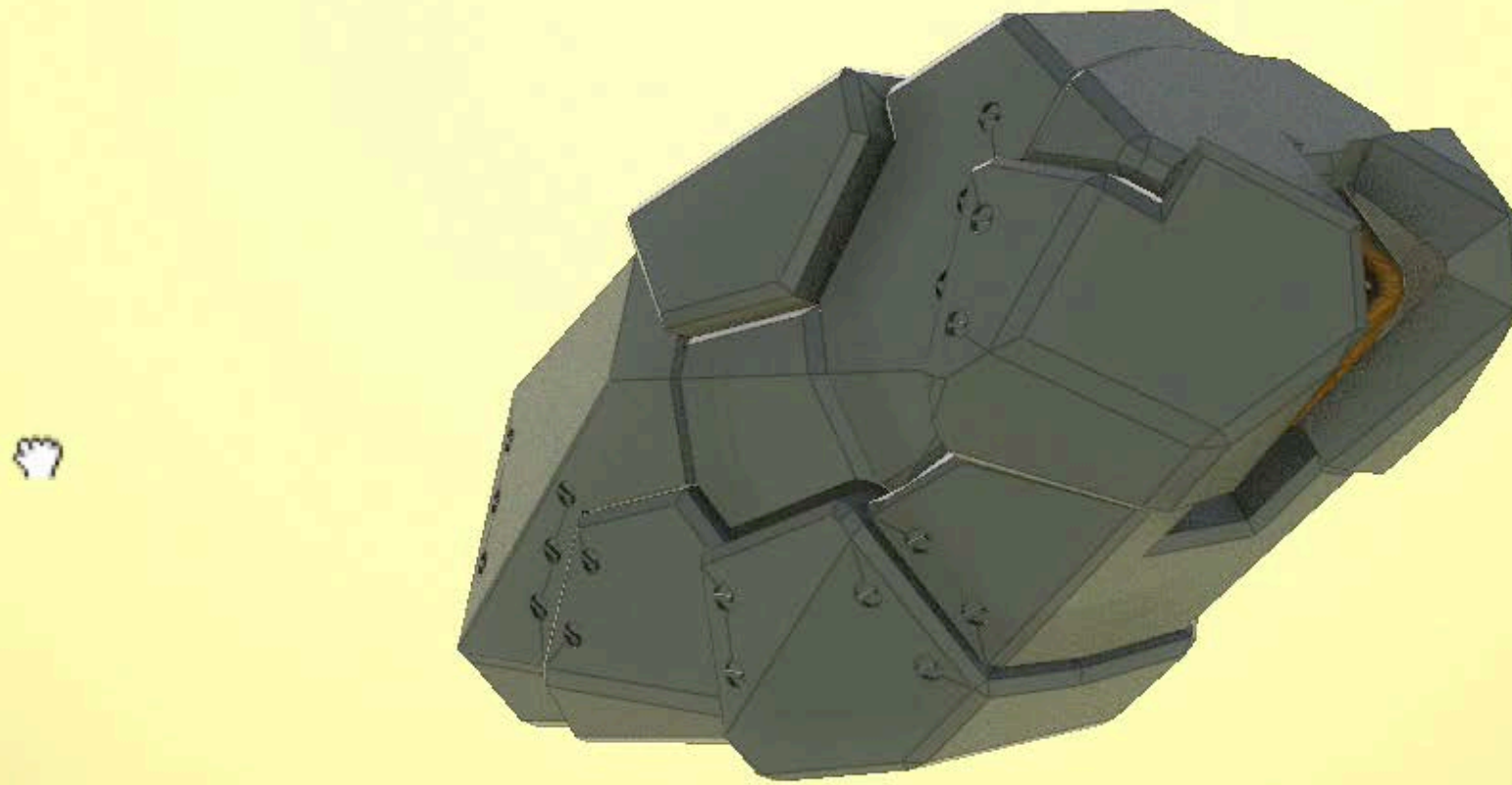
パネル分割のプロシージャル化



ケーブル配置のプロシージャル化



内部メカを伴う形状生成のプロシージャル化



プロシージャル化された各工程は
どのように管理編集されているのだろうか？
その具体例の1つを紹介します

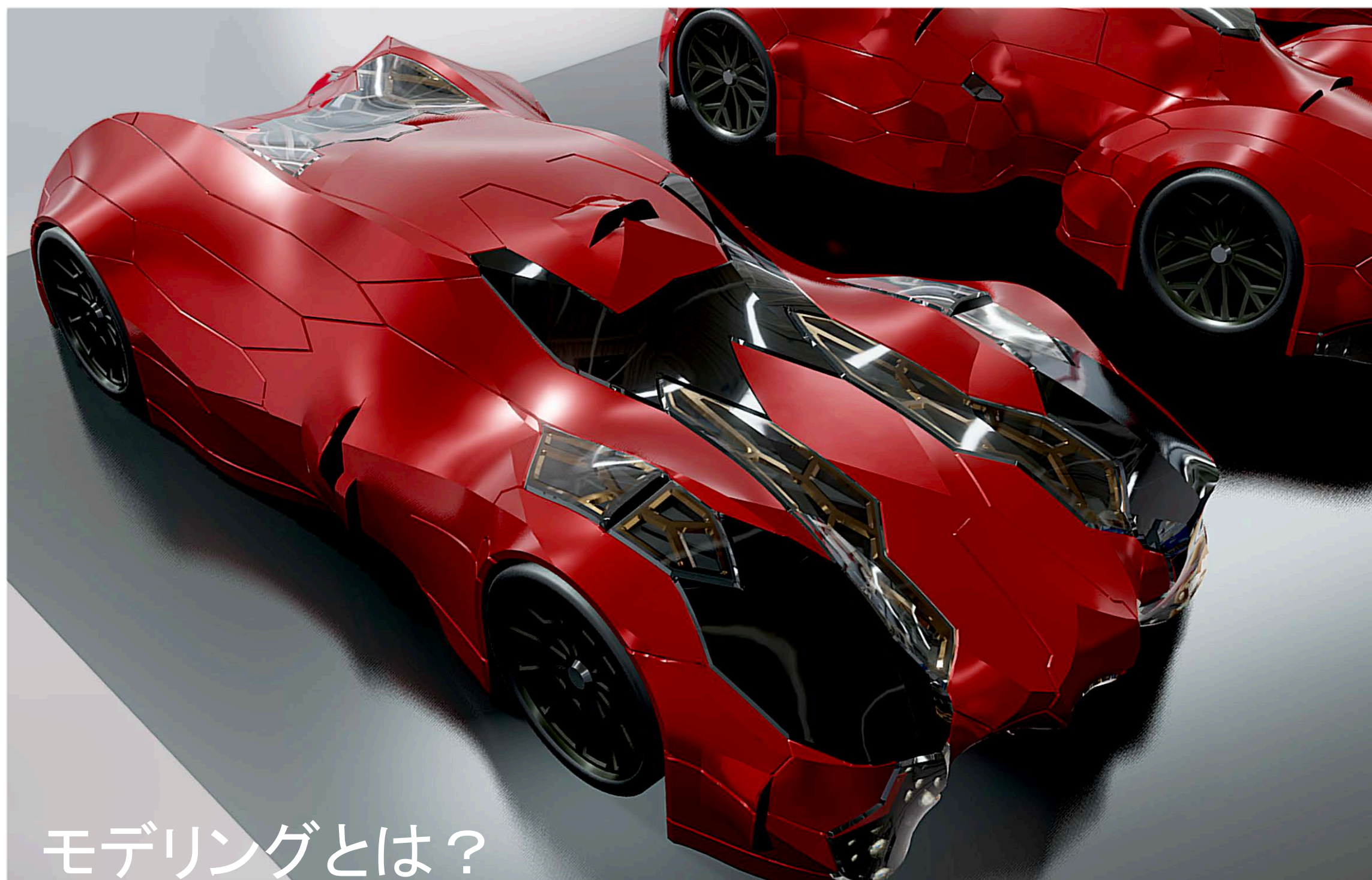
中学生の職業体験授業で カーモデリングを体験してもらう

- ・東京都江東区立某中学校の2年生 2名
- ・中2の1学期で連立方程式が終わり、
2学期から1次関数、という状況。
- ・3DCGは未経験
- ・期間は3日間
(カーモデリングは1.5日間 計7時間ほど)



Procedural Car Modeling With HOUDINI

POLYPHONY DIGITAL INC.



モデリングとは？

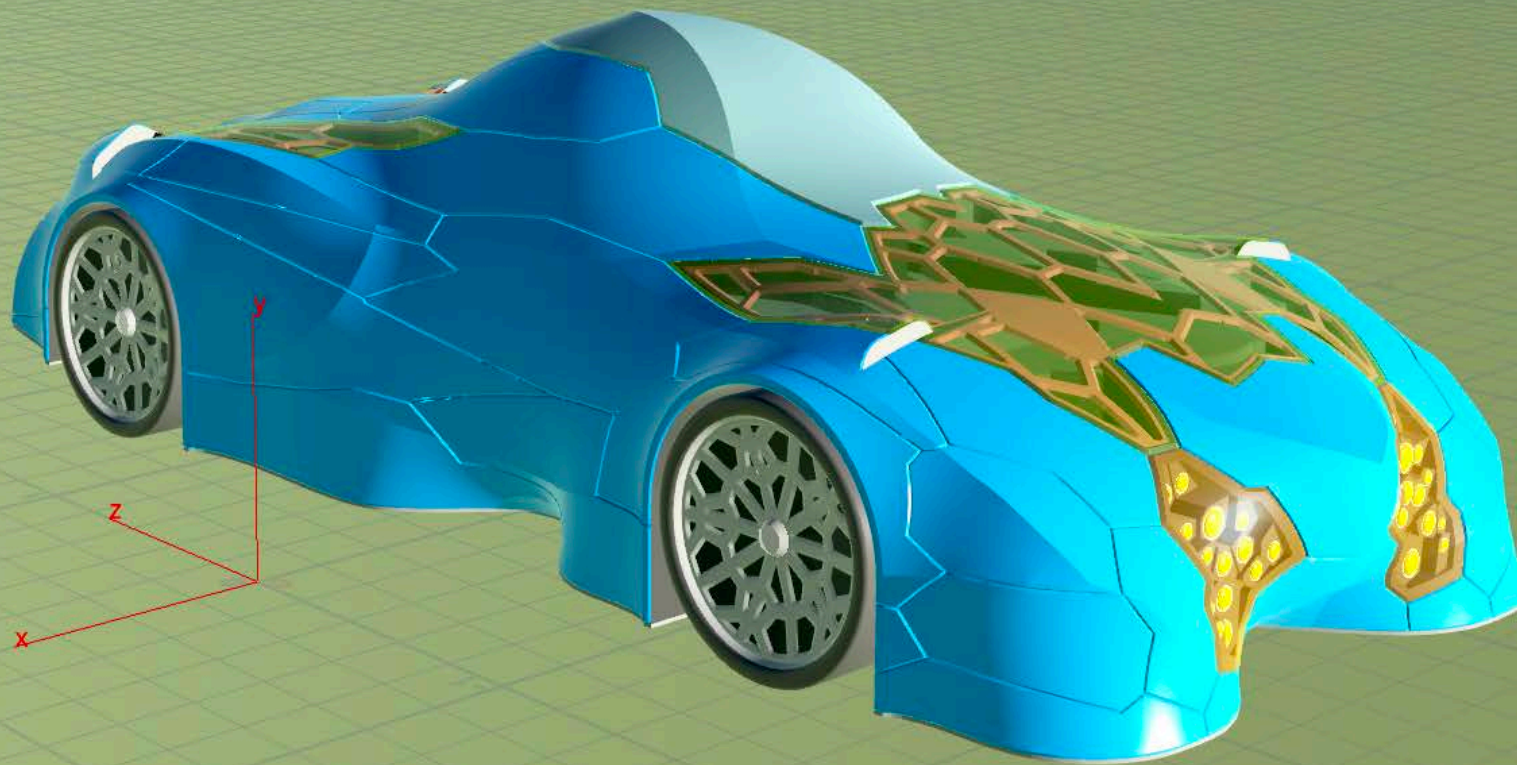
X Y Zで表す三次元空間

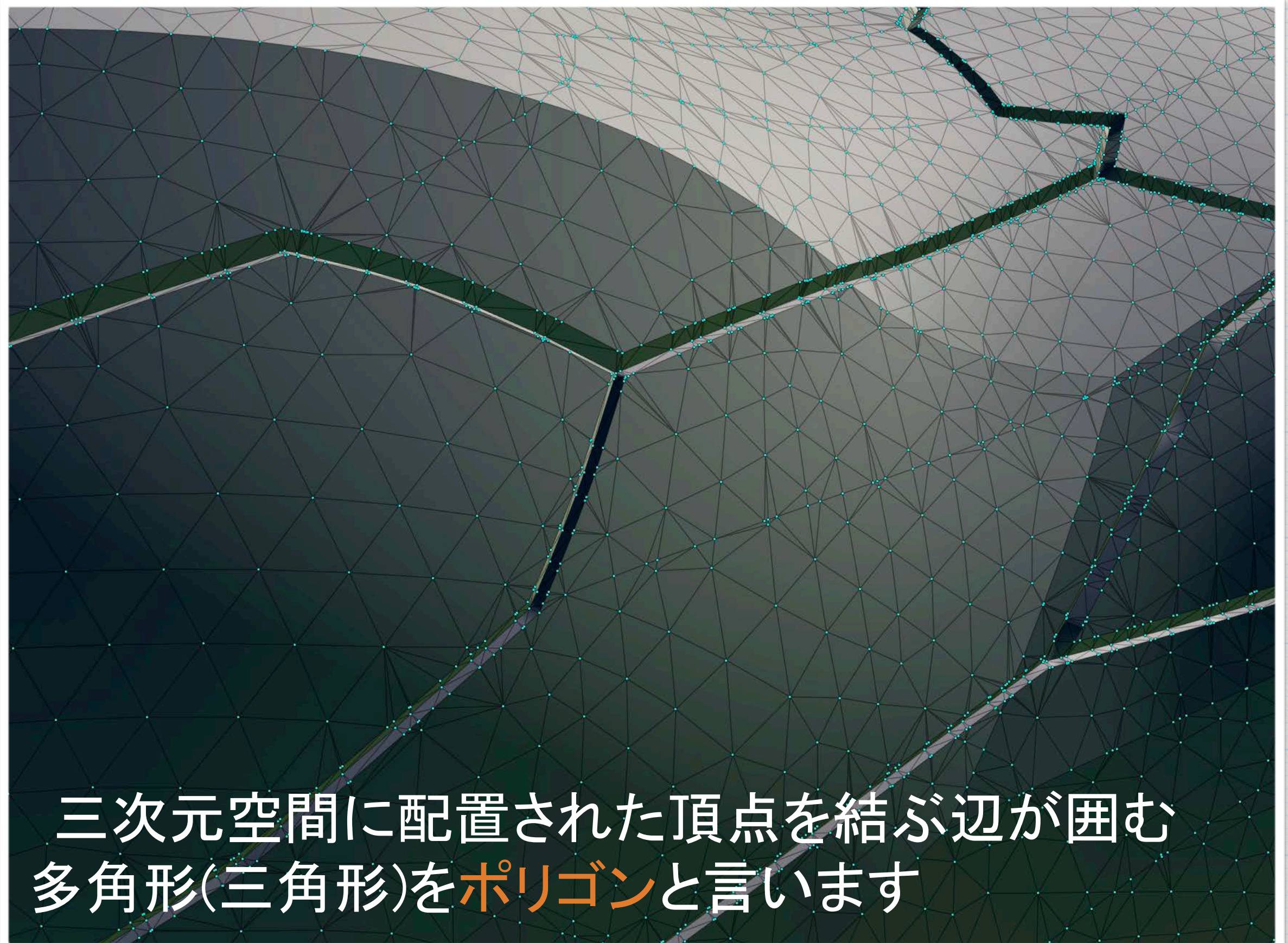
三次元座標

Xは横

Yは高さ

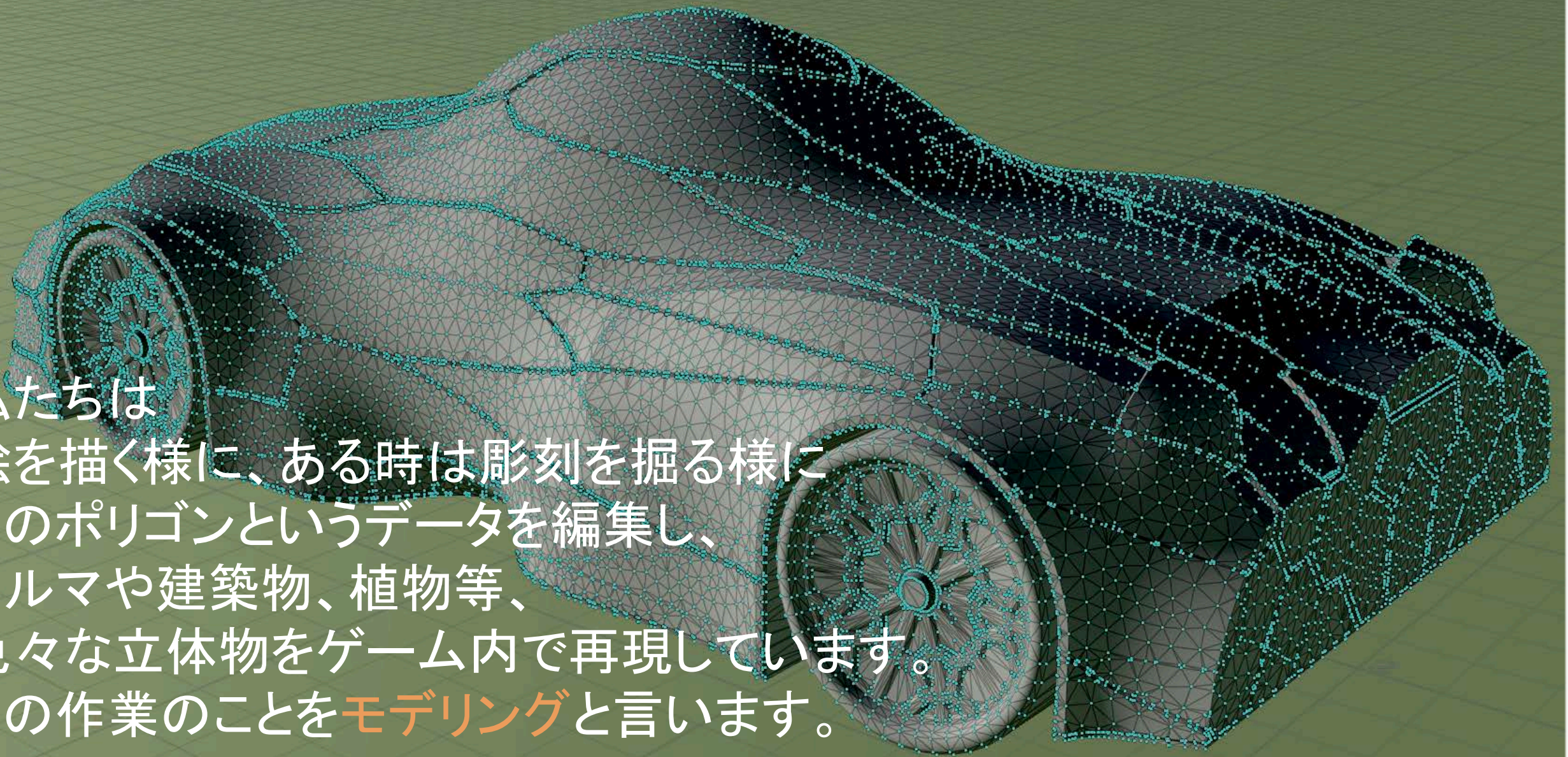
Zは奥行き





三次元空間に配置された頂点を結ぶ辺が囲む
多角形(三角形)をポリゴンと言います

私たちは
絵を描く様に、ある時は彫刻を掘る様に
このポリゴンというデータを編集し、
クルマや建築物、植物等、
色々な立体物をゲーム内で再現しています。
この作業のことを**モデリング**と言います。



プロシージャルモデリングとは？

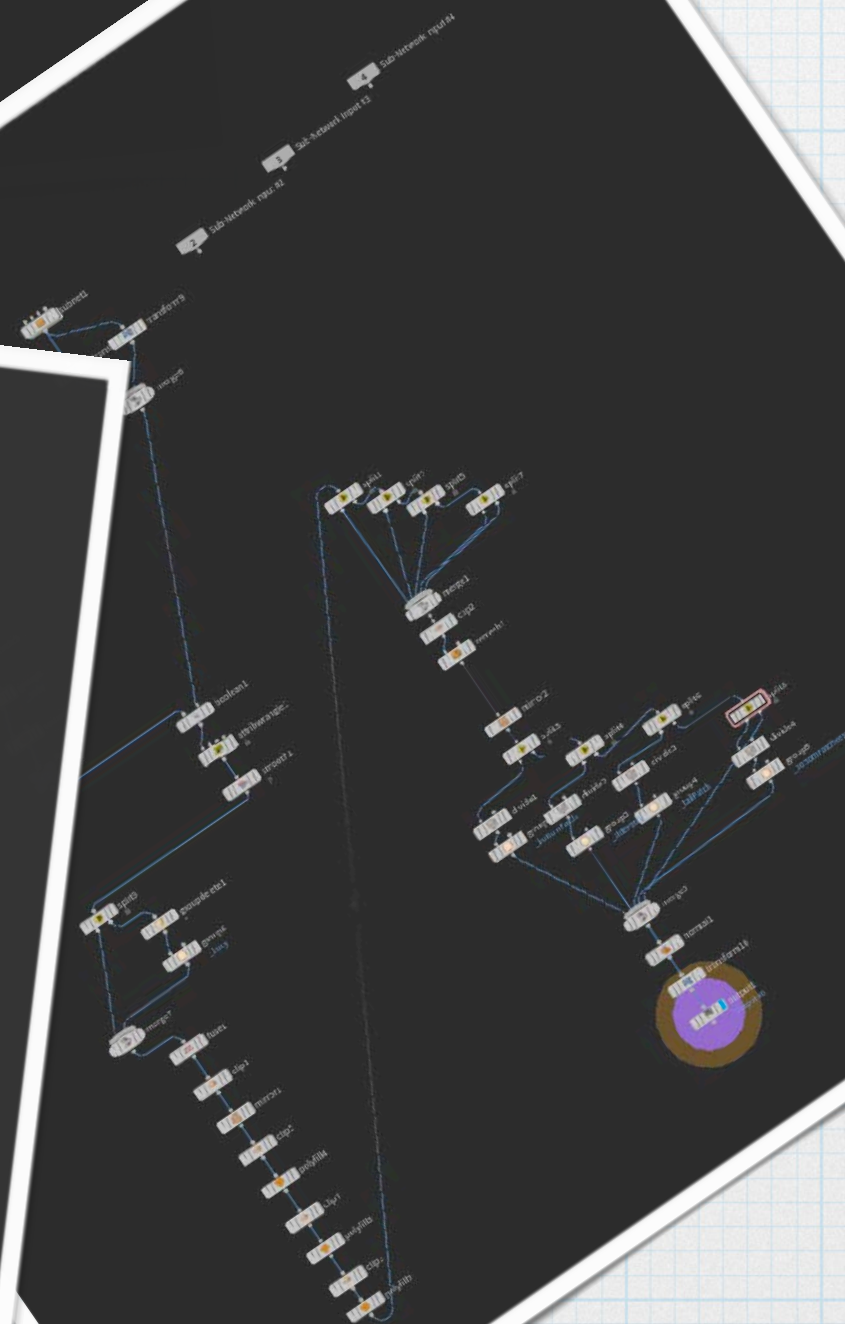
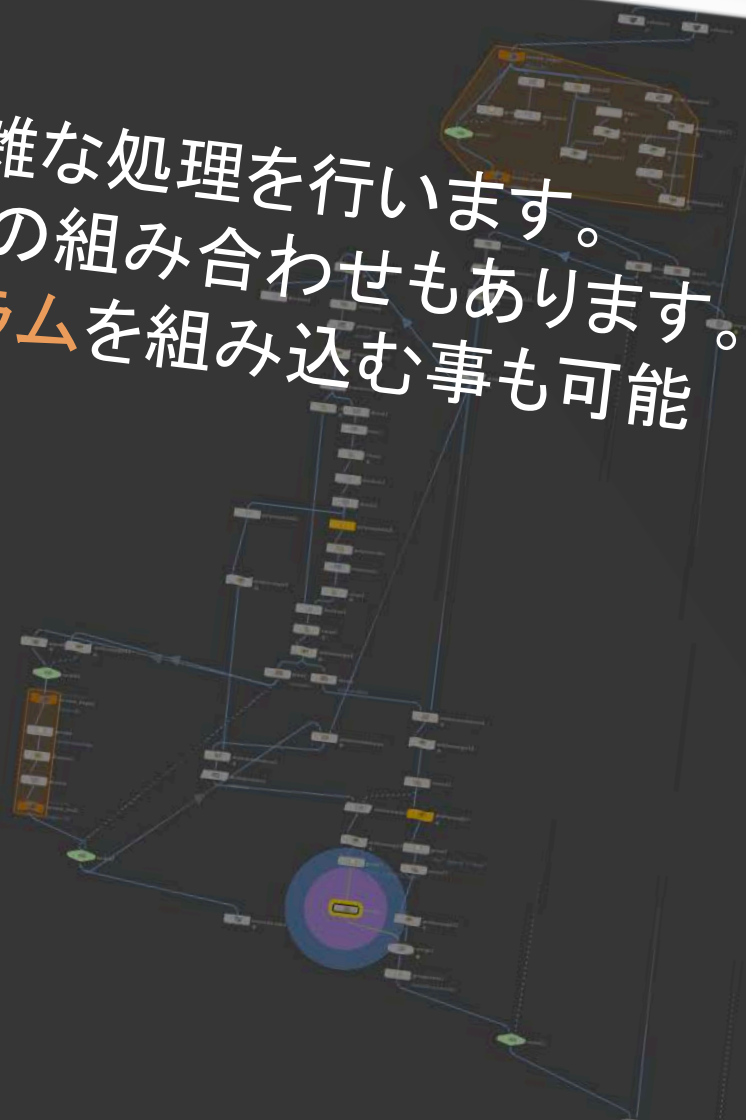
何らかの形状を作成する際に必要な
プロシージャル(手続き)を予め設定し、
目的の形状を自動的に生成する技術。





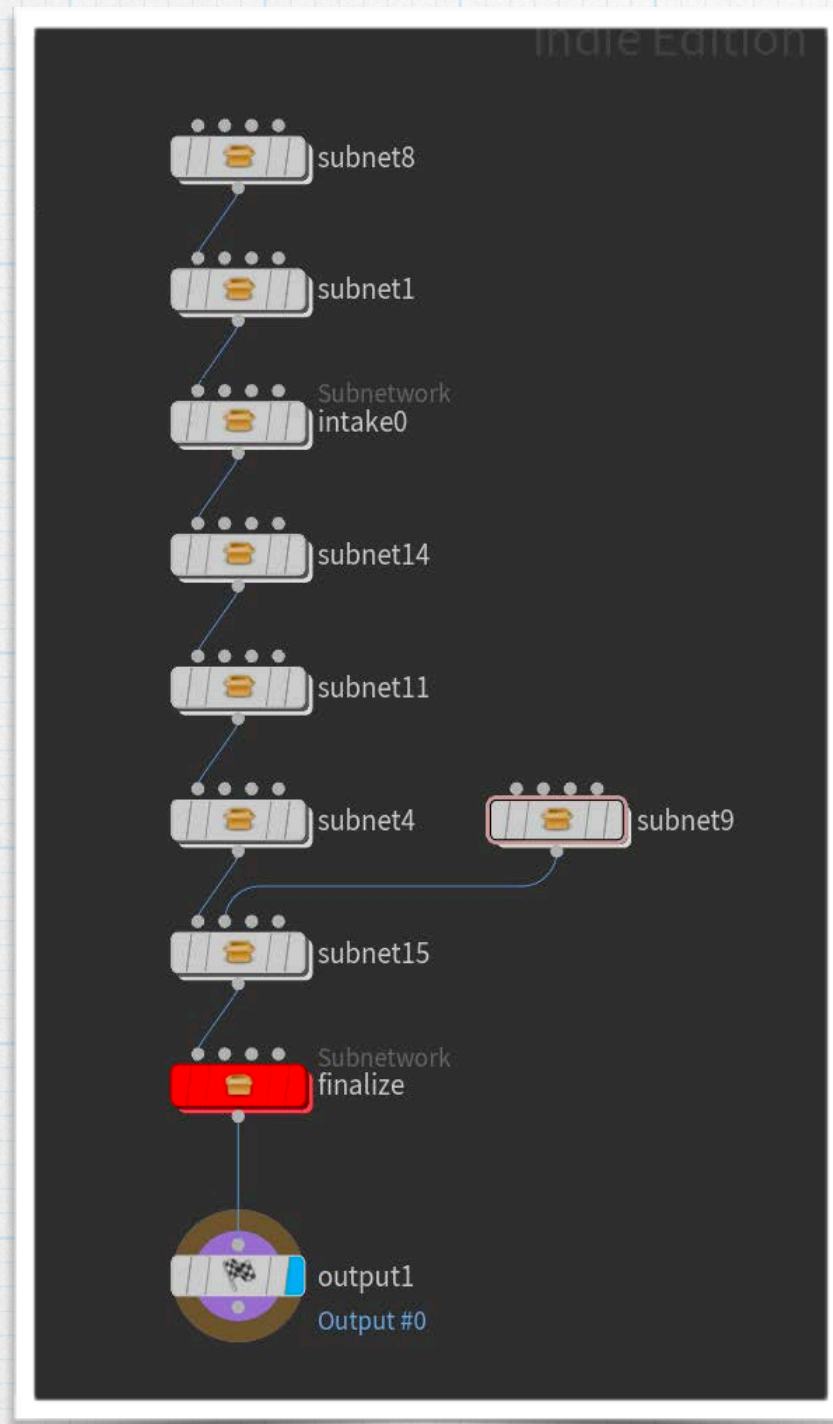
Houdiniでは1つの1つの
プロシージャル(手続き)は
このような
ノード(ブロック)で表現されます

ノードを接続して複雑な処理を行います。
非常に複雑なノードの組み合わせもあります。
ノードの中にプログラムを組み込む事も可能



Houdiniでは上から下に向かって処理されます

開始



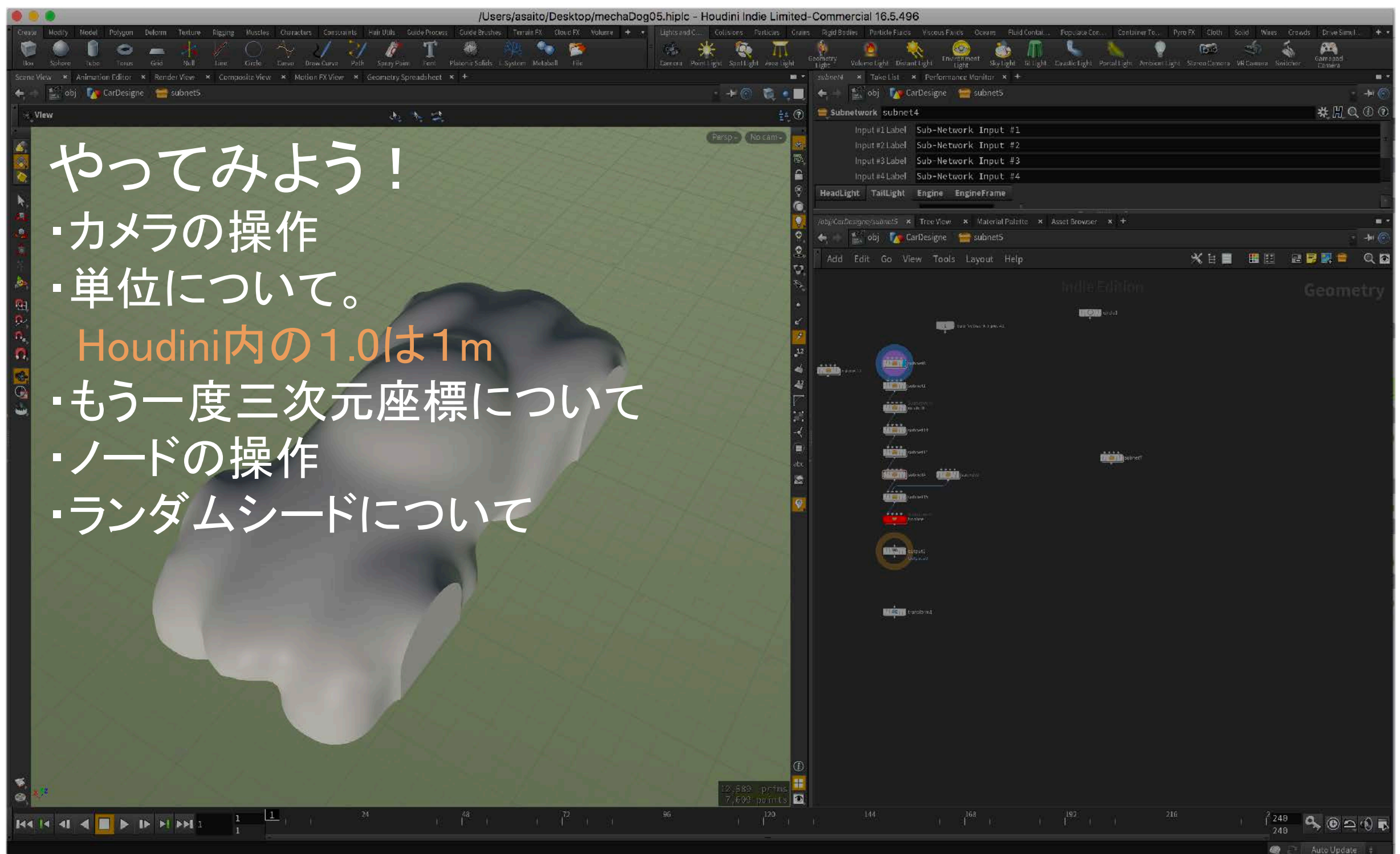
完成

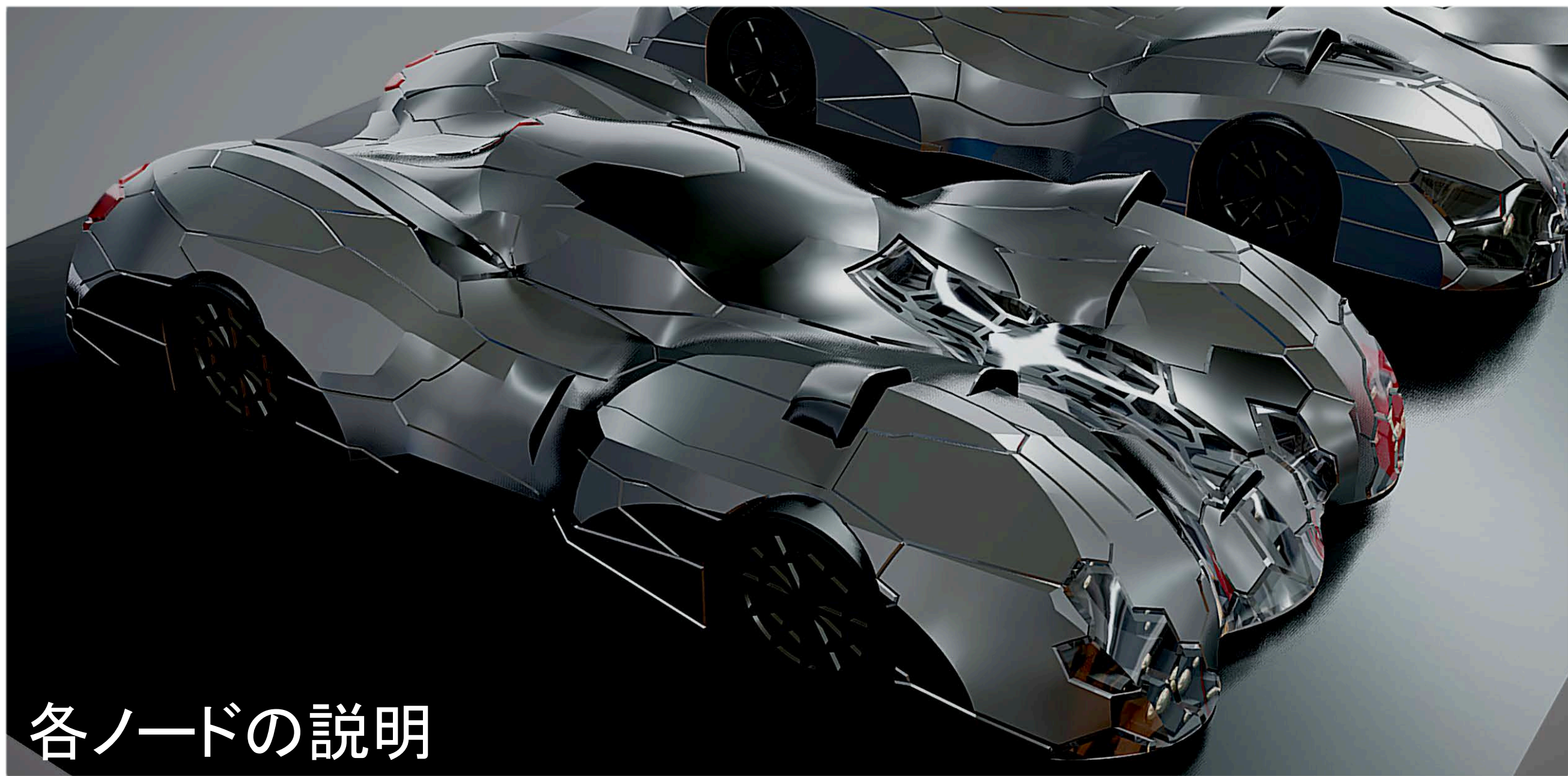
やってみよう！

- ・カメラの操作
- ・単位について。

Houdini内の1.0は1m

- ・もう一度三次元座標について
- ・ノードの操作
- ・ランダムシードについて





各ノードの説明



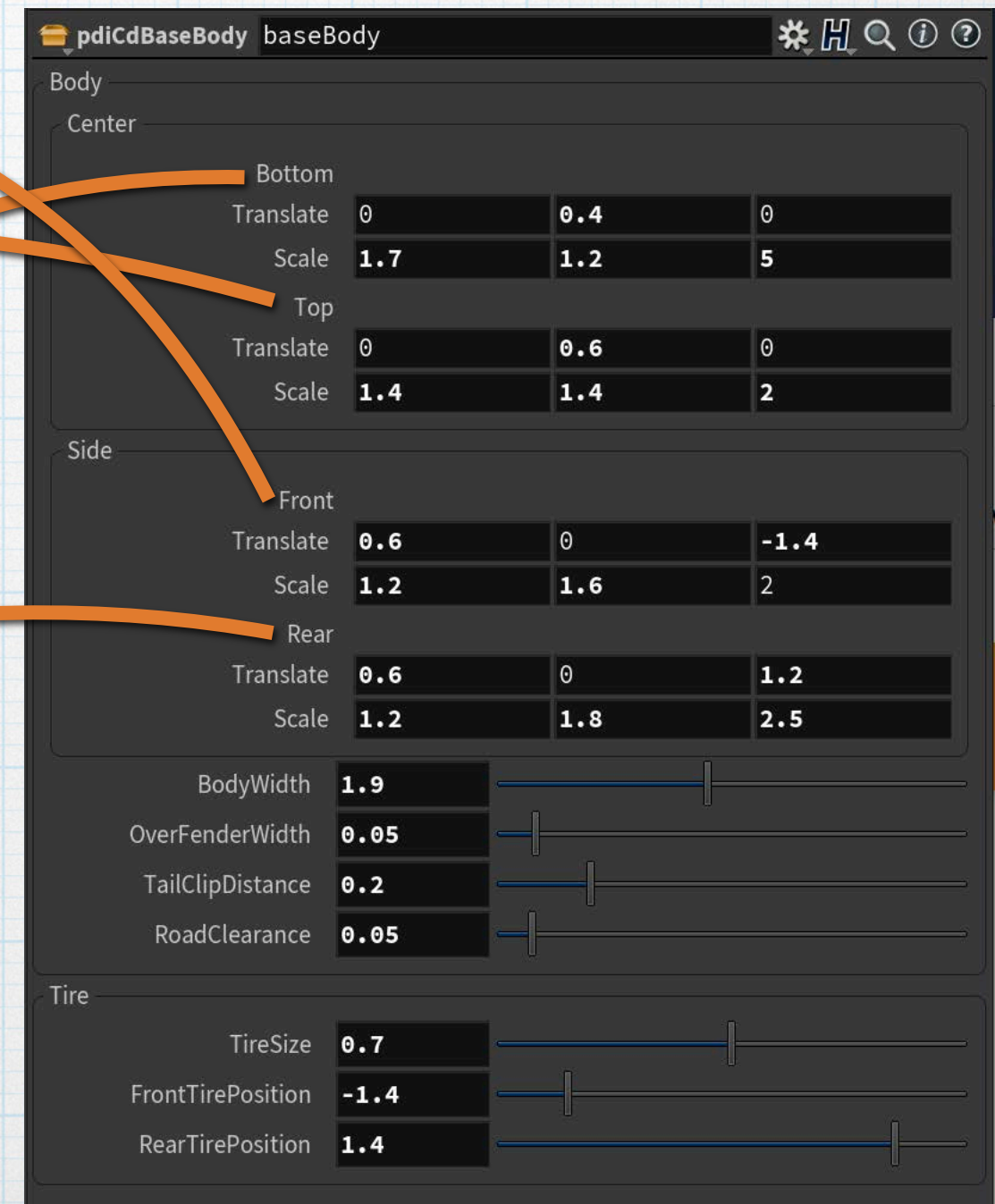
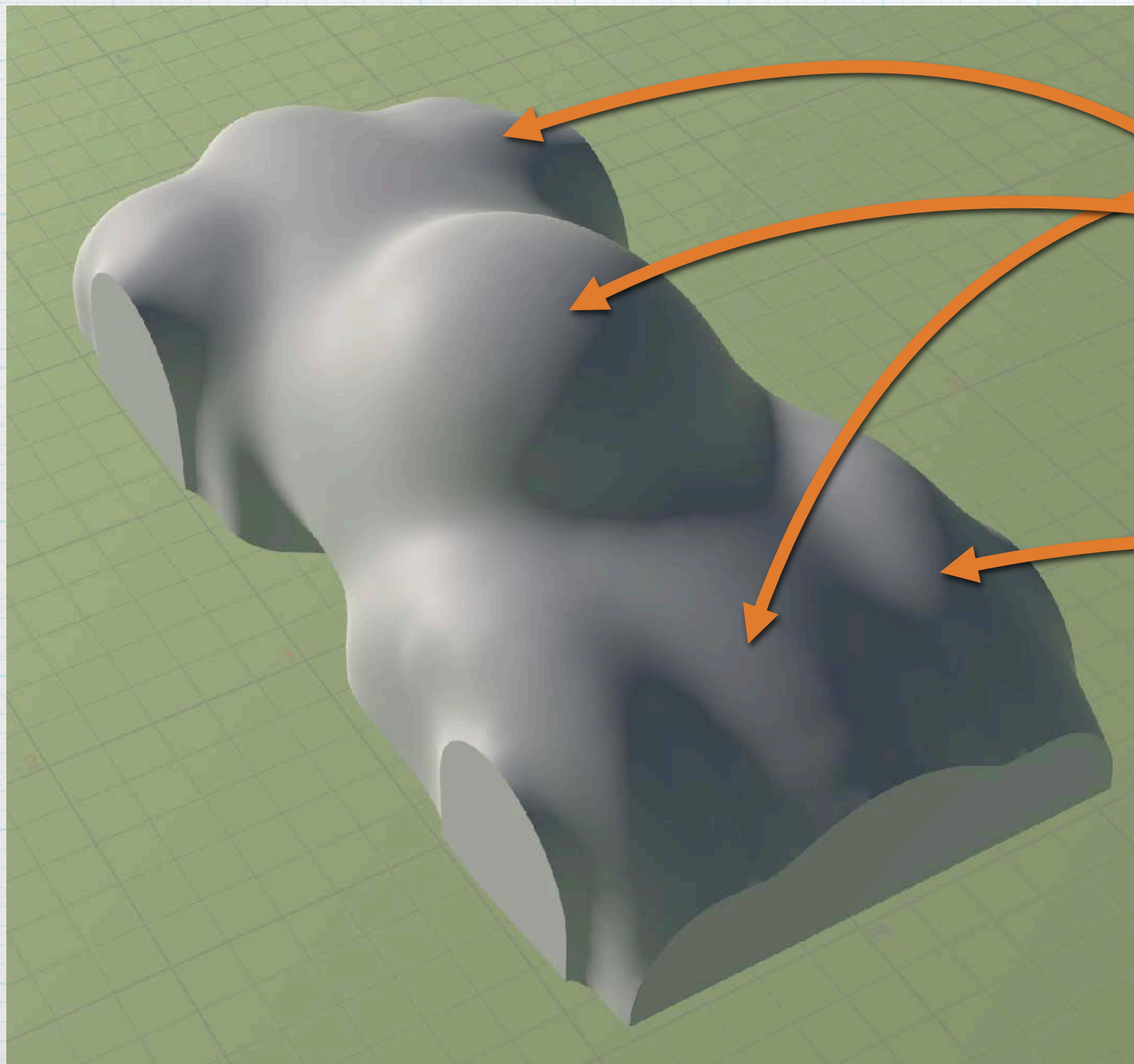
pdiCdBaseBody(ボディーの基礎を生成)

Body:Center

Body:Side

クルマのおおまかな形状を決定する

6つの楕円体(左右対称)の位置(Transform)と大きさ(Scale)を設定します



pdiCdBaseBody(ボディーの基礎を生成)

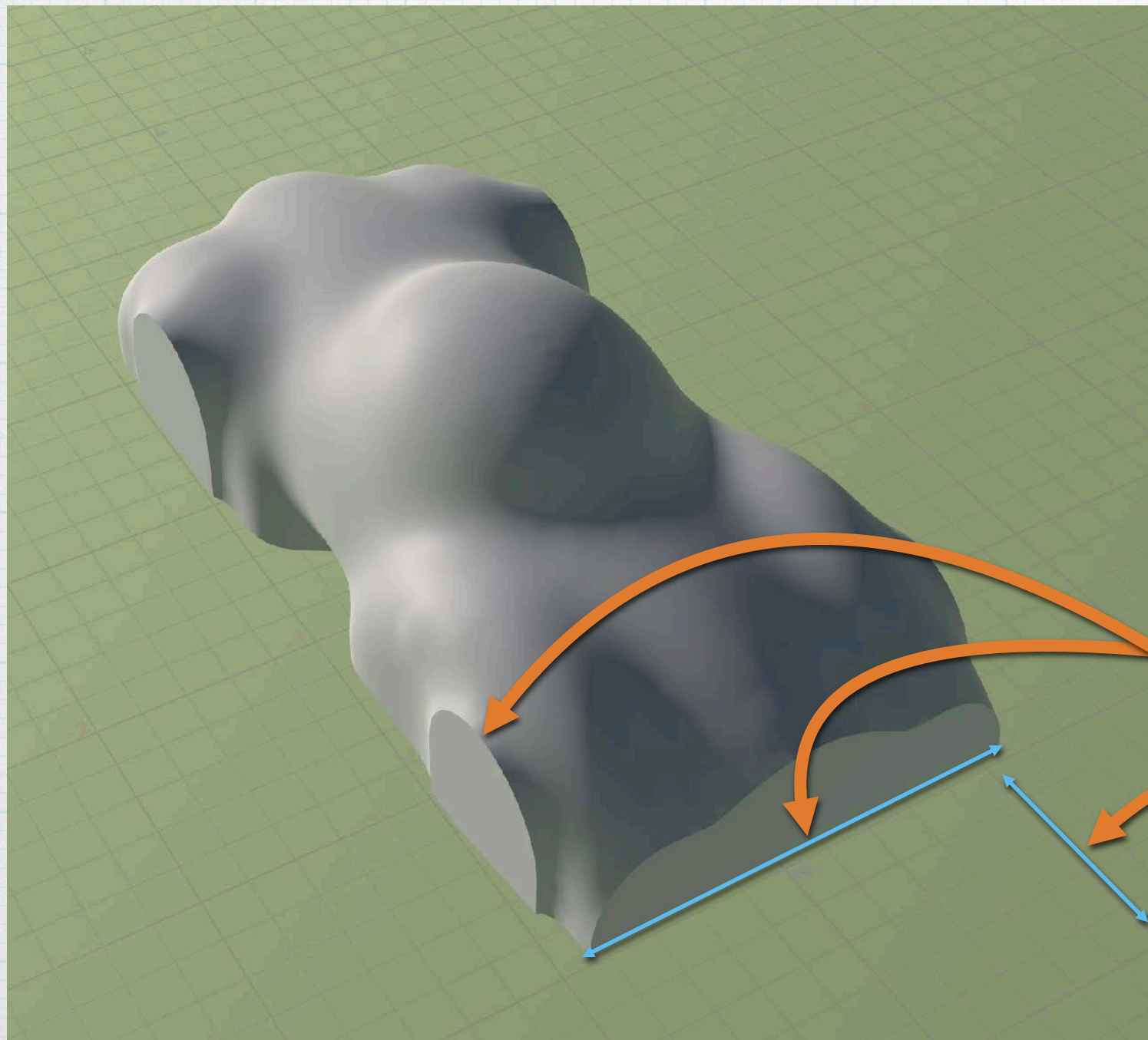
Body:

BodyWidth: 車幅

OverFenderWidth: 車幅からはみ出たタイヤを覆う部分

TailClipDistance: 後部の切り落とし距離

RoadClearance: 車高(地面との隙間)



pdiCdBaseBody baseBody

Body

Center

Bottom			
Translate	0	0.4	0
Scale	1.7	1.2	5
Top			
Translate	0	0.6	0
Scale	1.4	1.4	2

Side

Front			
Translate	0.6	0	-1.4
Scale	1.2	1.6	2
Rear			
Translate	0.6	0	1.2
Scale	1.2	1.8	2.5

BodyWidth 1.9

OverFenderWidth 0.05

TailClipDistance 0.2

RoadClearance 0.05

Tire

TireSize	0.7
FrontTirePosition	-1.4
RearTirePosition	1.4

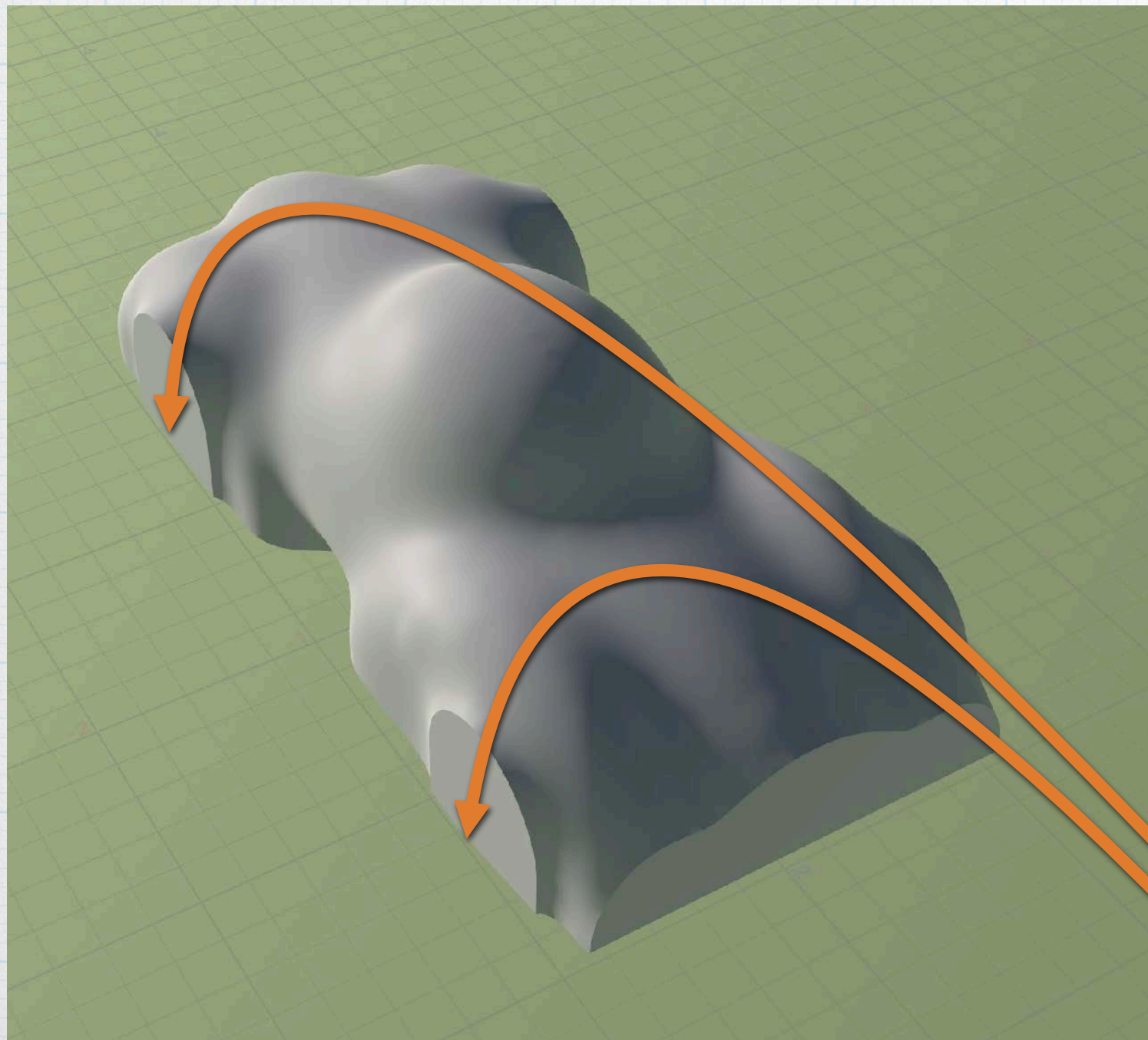
pdiCdBaseBody(ボディーの基礎を生成)

Tire:

TireSize: タイヤの直径

FrontTirePosition: 前輪の位置(Z座標)

RearTirePosition: 後輪の位置(Z座標)



pdiCdBaseBody baseBody

Body

Center

Bottom			
Translate	0	0.4	0
Scale	1.7	1.2	5
Top			
Translate	0	0.6	0
Scale	1.4	1.4	2

Side

Front			
Translate	0.6	0	-1.4
Scale	1.2	1.6	2
Rear			
Translate	0.6	0	1.2
Scale	1.2	1.8	2.5

BodyWidth 1.9

OverFenderWidth 0.05

TailClipDistance 0.2

RoadClearance 0.05

Tire

TireSize	0.7
FrontTirePosition	-1.4
RearTirePosition	1.4

pdiCdBodyLine(ボディーのラインを生成)

Divide:

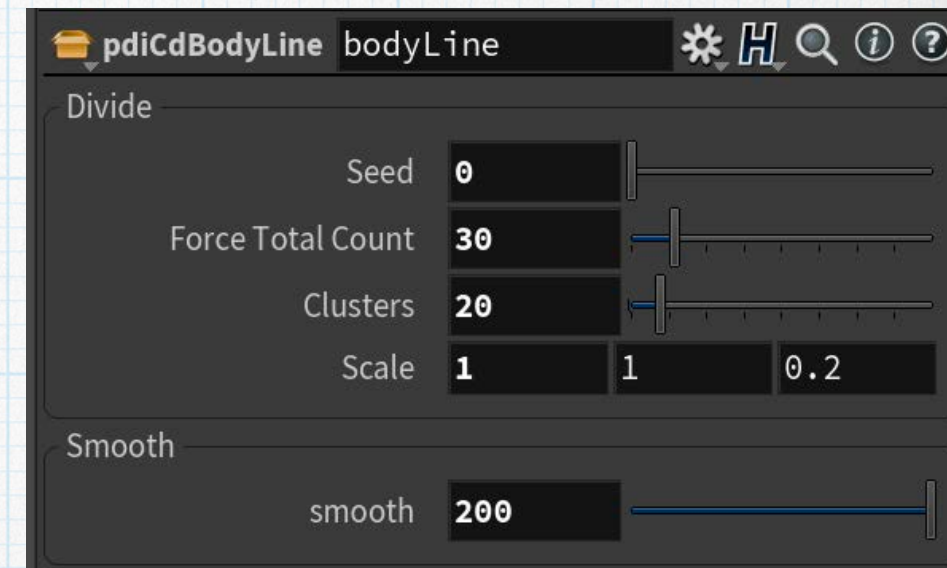
Seed: この数値毎に異なる分割を行います。

ForceTotalCount: 分割数を設定します

Clusters: 分割をまとめるクラスターの数を設定します。

Scale: パネルの密度を設定します。

Smooth: エッジの鋭さを設定します。数値が大きいと鋭くなります。

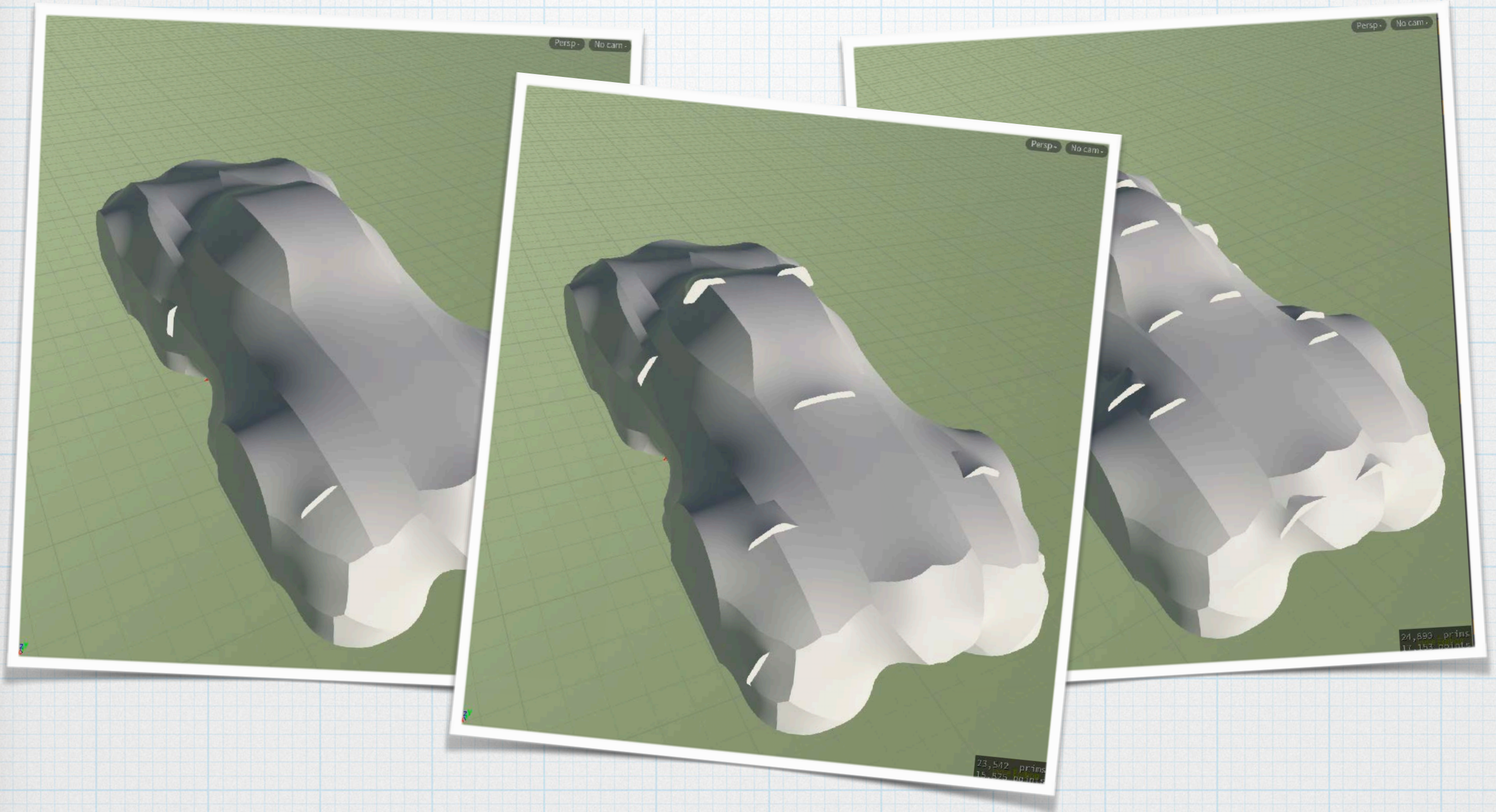


pdiCdBodyIntake(エアインテークを生成)

エアインテークを配置します

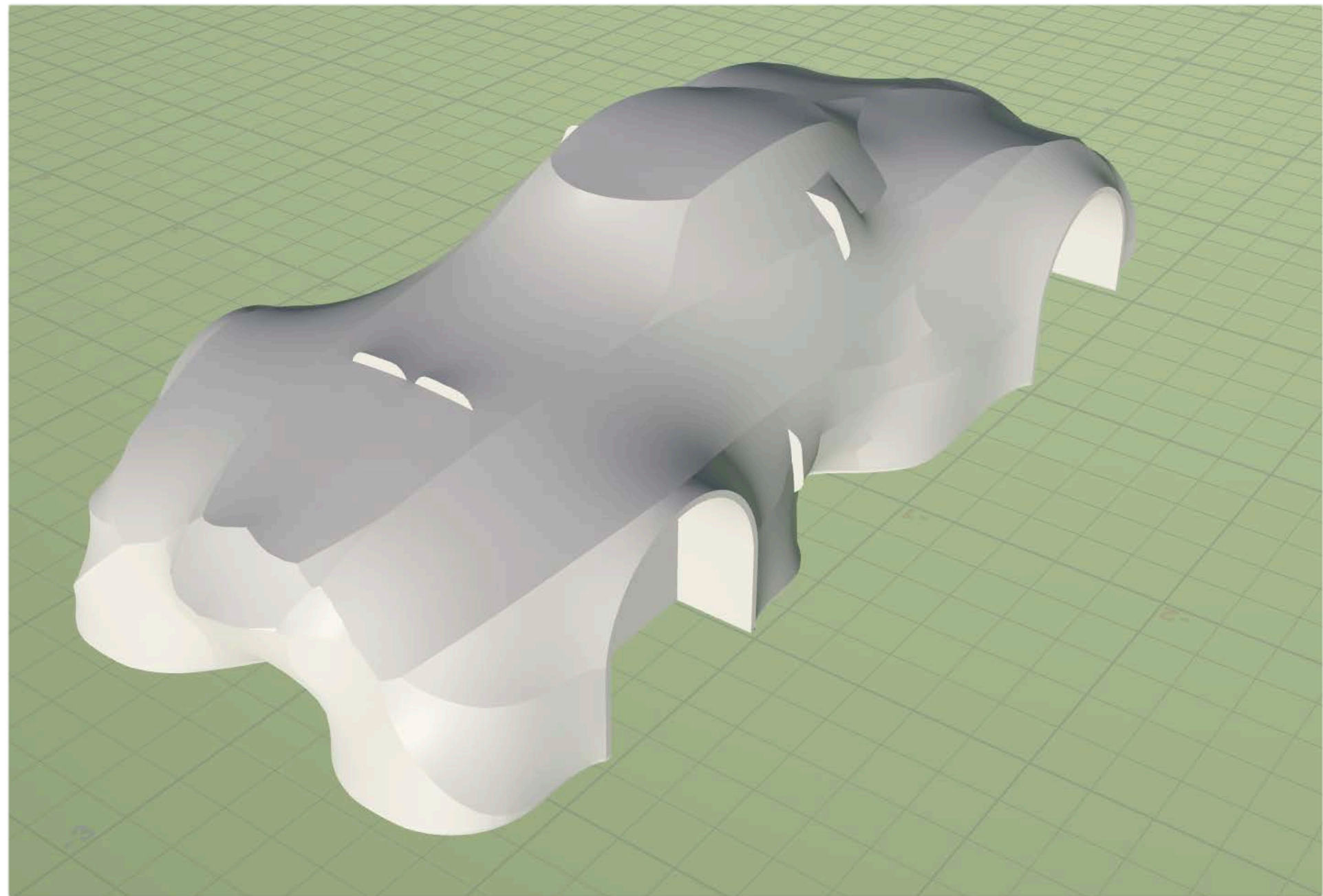
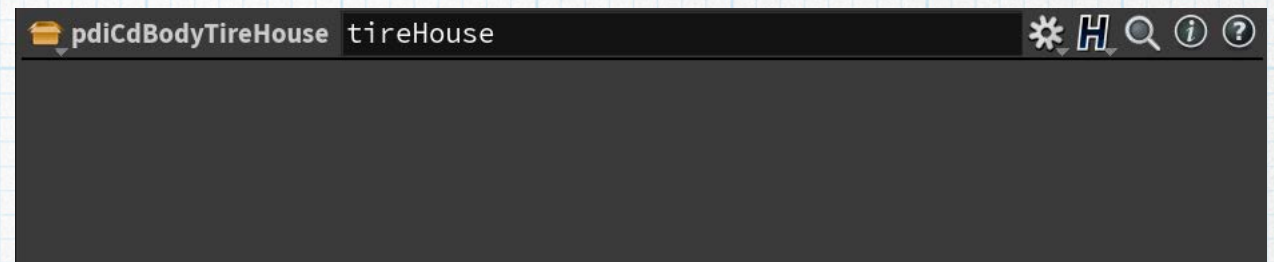
GlobalSeed: この数値毎に異なる配置を行います。

ForceTotalCount: 配置する数。



pdiCdBodyTireHouse(タイヤハウスを生成)

設定項目はありません。



pdiCdBodyPanelDivide(パネル分割)

フロントガラスや、ヘッドライトなどの分割を行います。

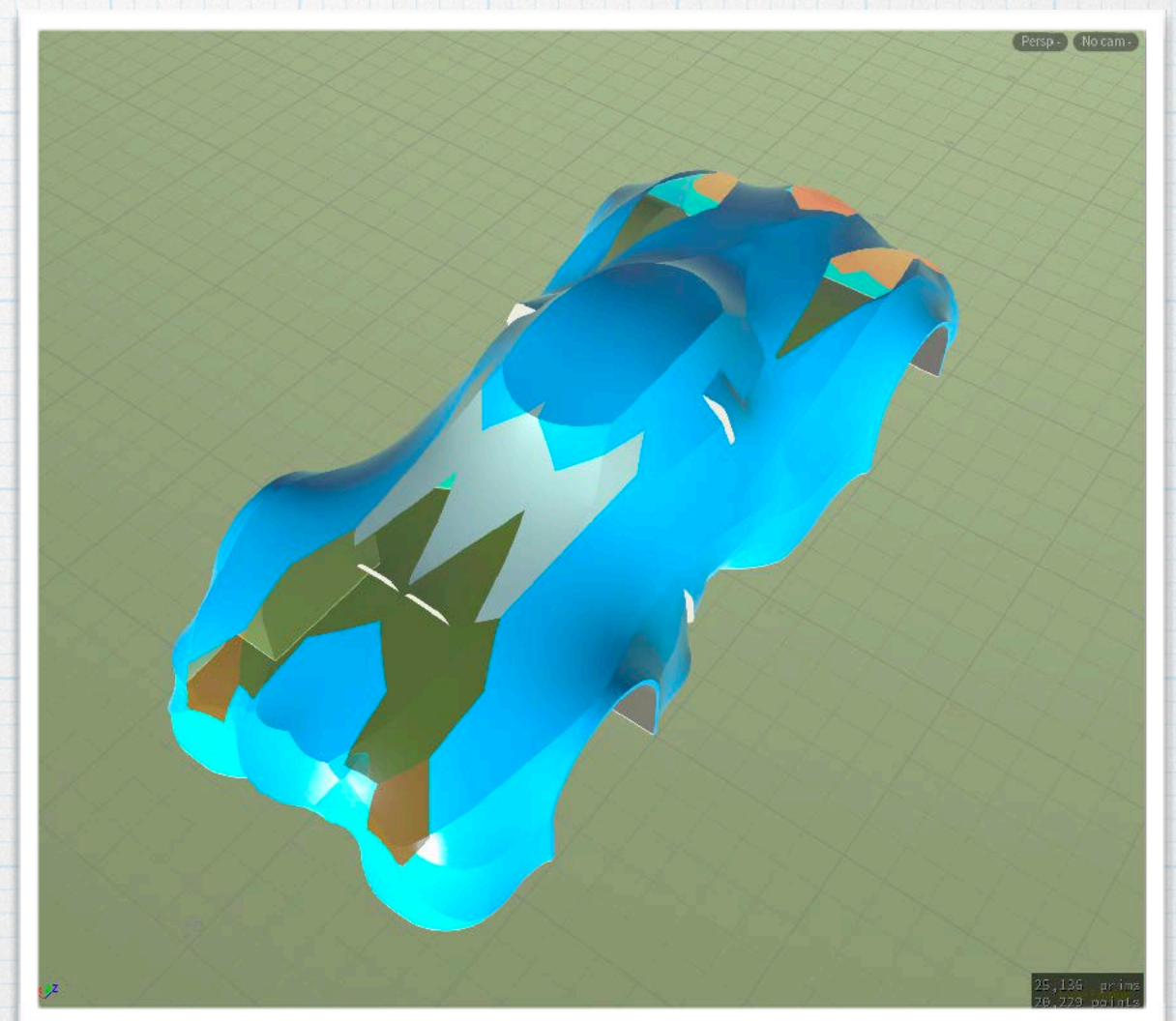
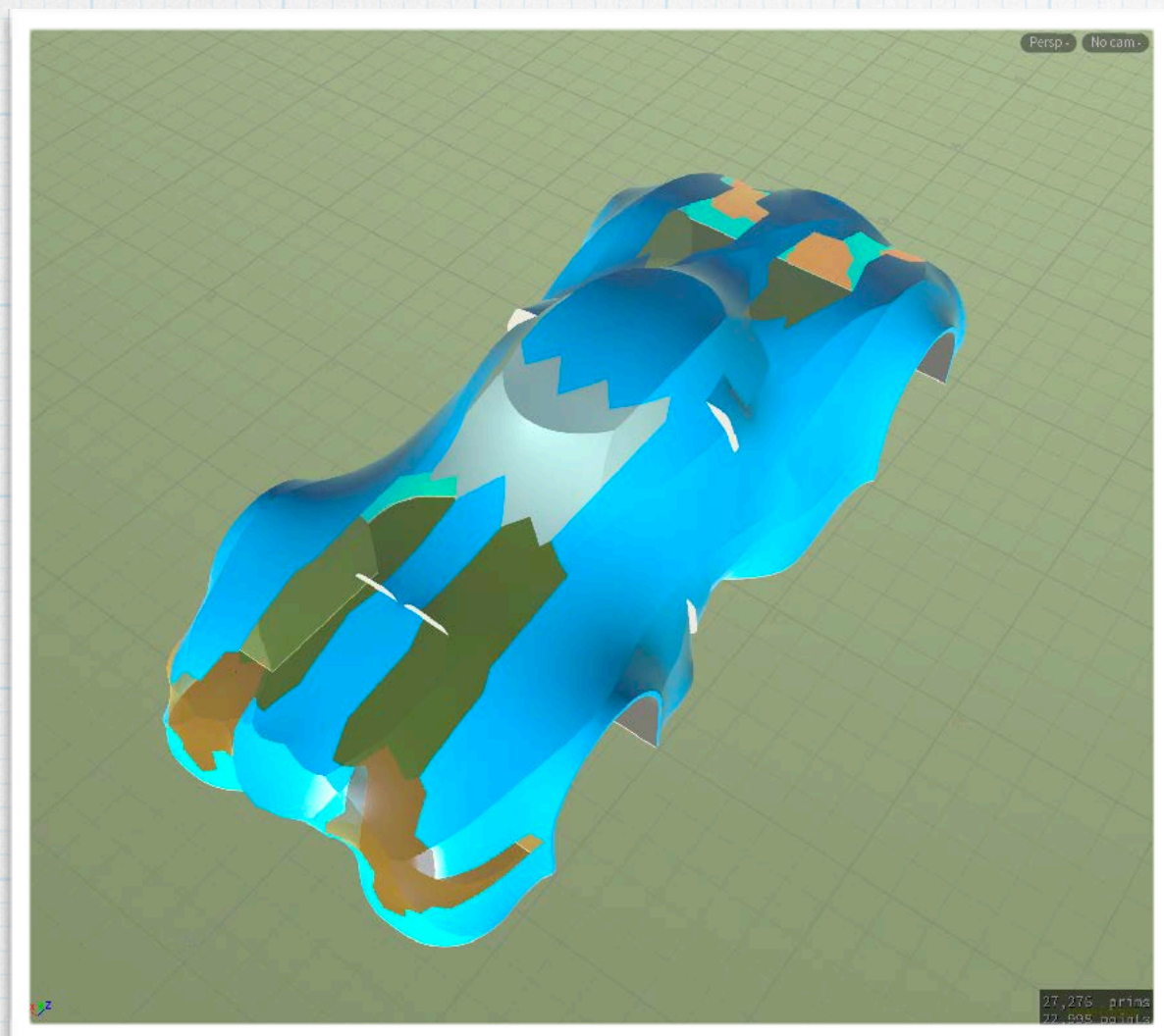
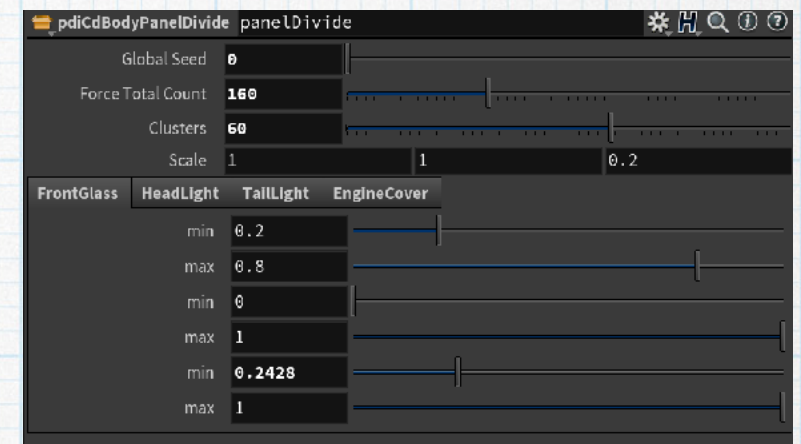
GlobalSeed: この数値毎に異なる分割を行います。

ForceTotalCount: 分割数を設定します

Clusters: 分割をまとめるクラスターの数を設定します。

Scale: パネルの密度を設定します。

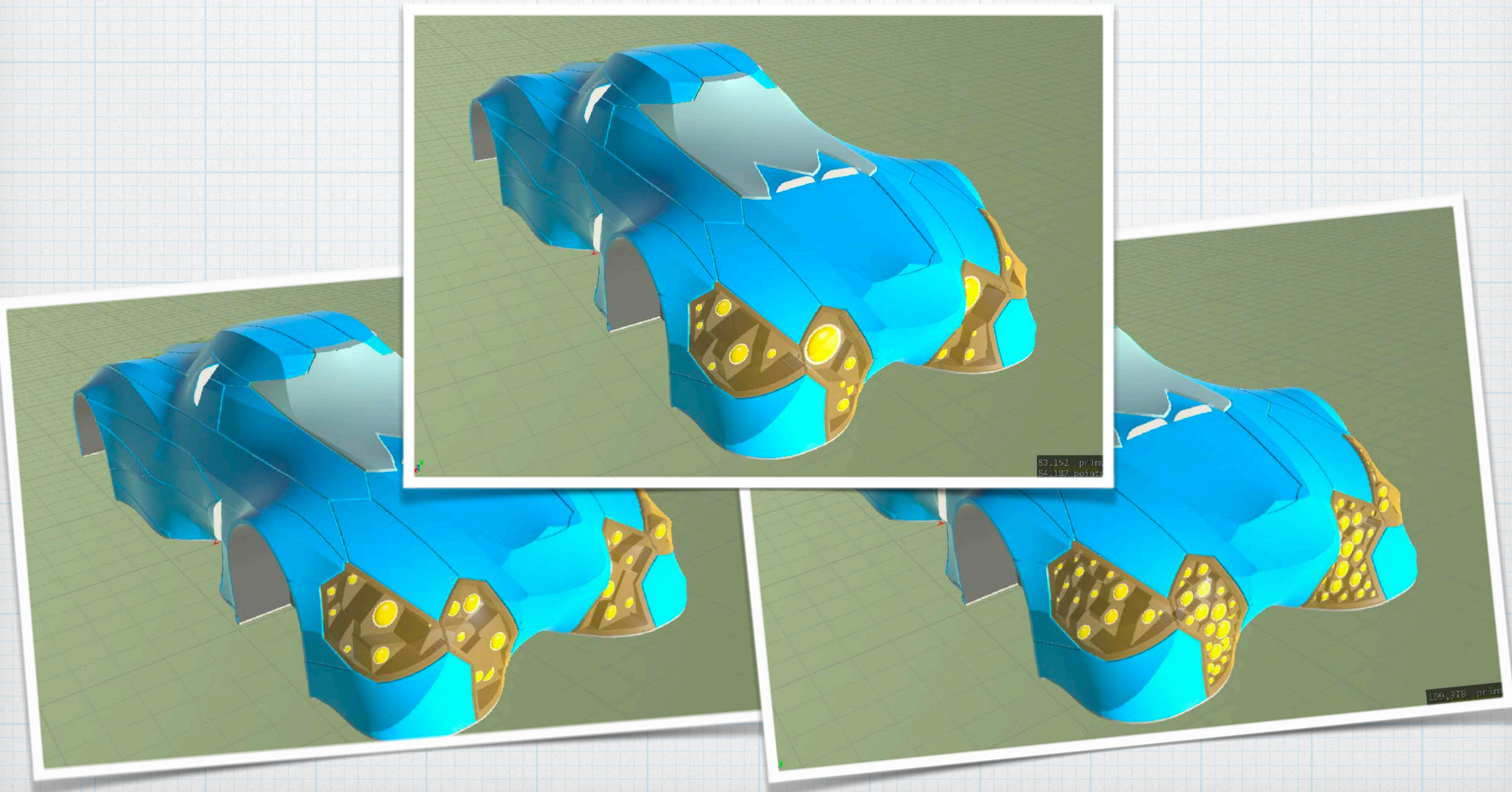
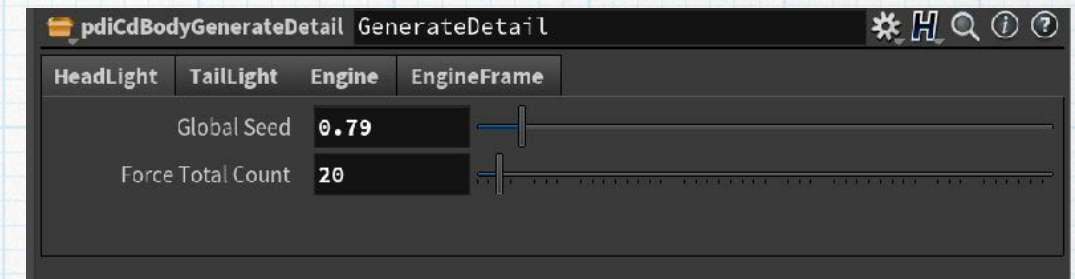
FrontGlass、HeadLight、TailLight、EngineCoverの出現範囲を設定します。



pdiCdBodyGenerateDetail(各部位のディテールを生成)

ヘッドライトなどのディテールを追加します。

項目毎に詳細設定があります。



pdiCdGenerateTire(タイヤを生成)

タイヤを生成します。

size:タイヤの直径

segment:ホイールの模様の円の分割数。

GlobalSeed:この数値毎に、ホイールの分割パターンが変化します。

ForceTotalCount:ホイールの分割の複雑さ。数値が多いほど細かい模様になります。



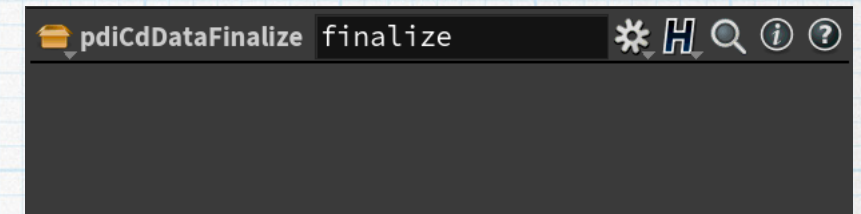
pdiCdTireLayout(タイヤを配置)

タイヤを配置します。
設定項目はありません。



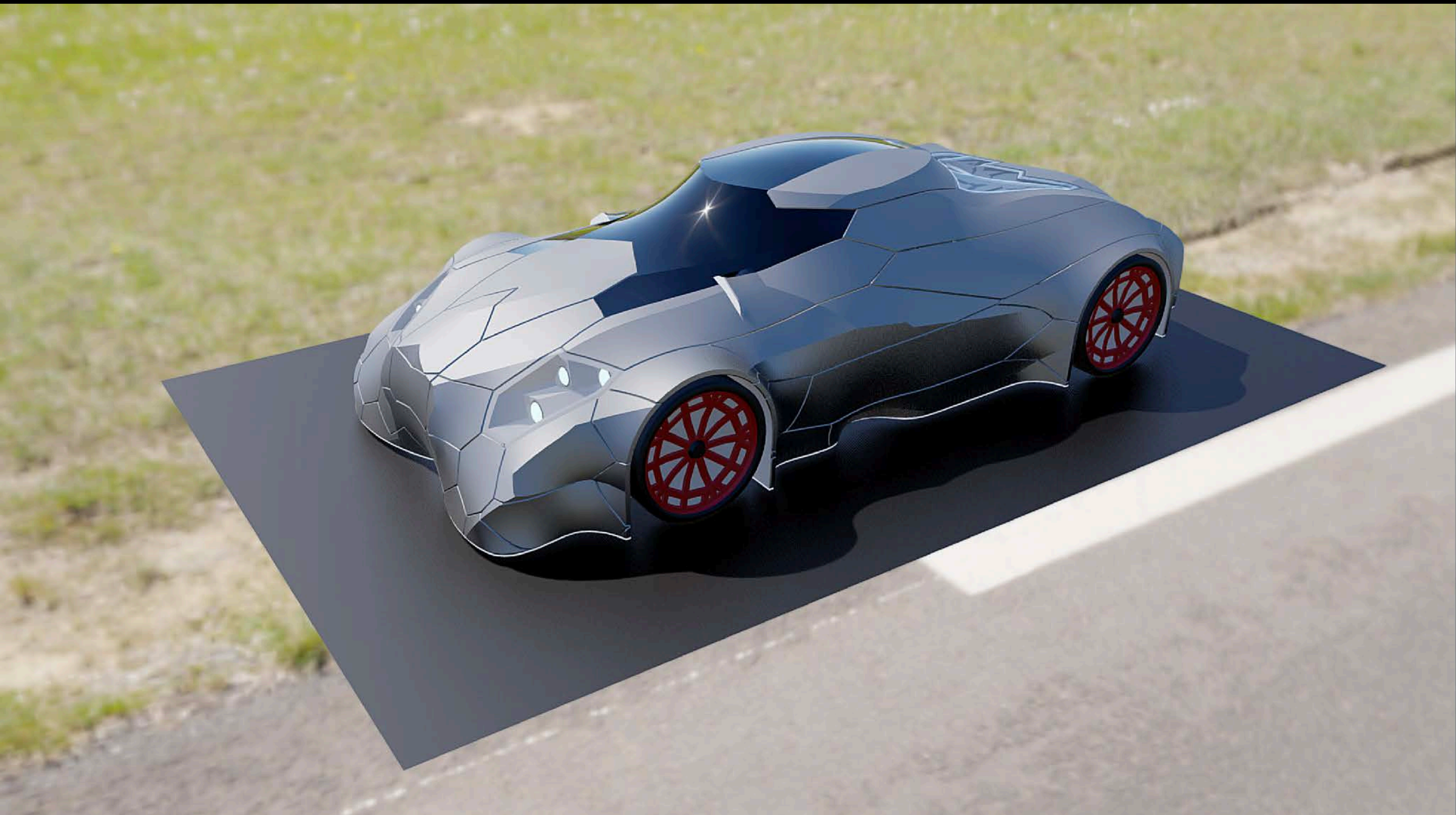
pdiCdDataFinalize(データの最終調整)

設定項目はありません。

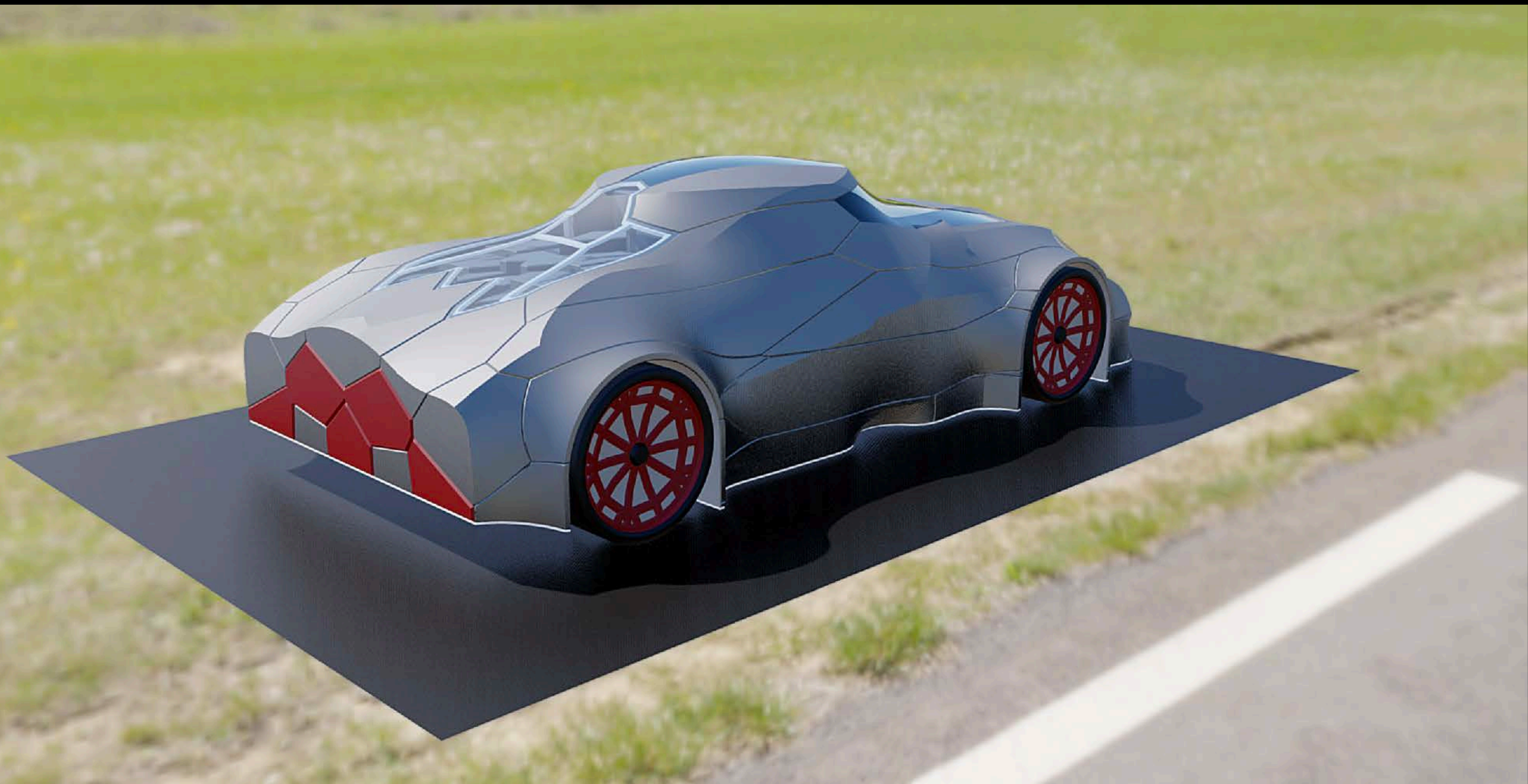




成果1日目 A君



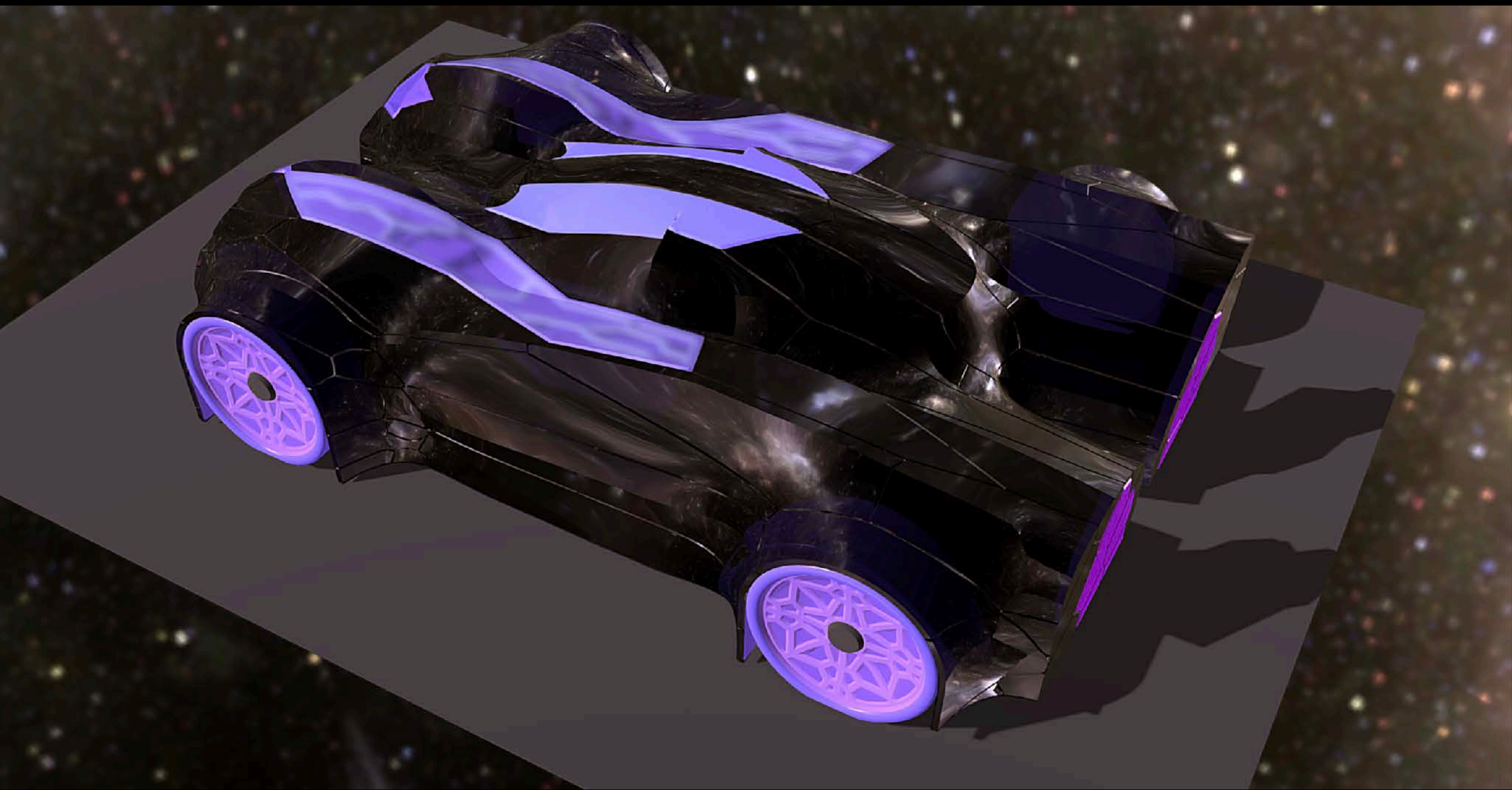
成果1日目 A君



成果1日目 B君



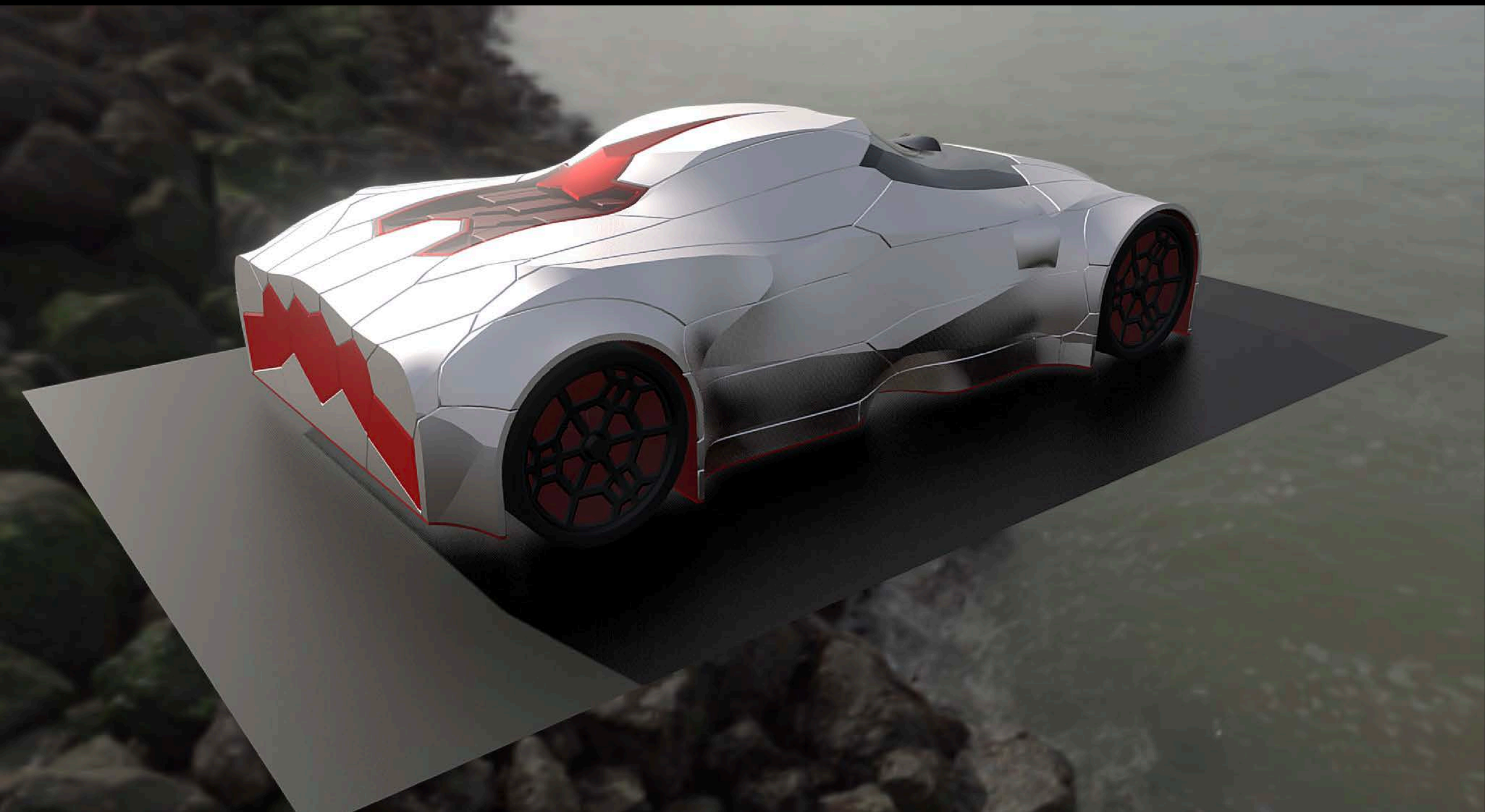
成果1日目 B君



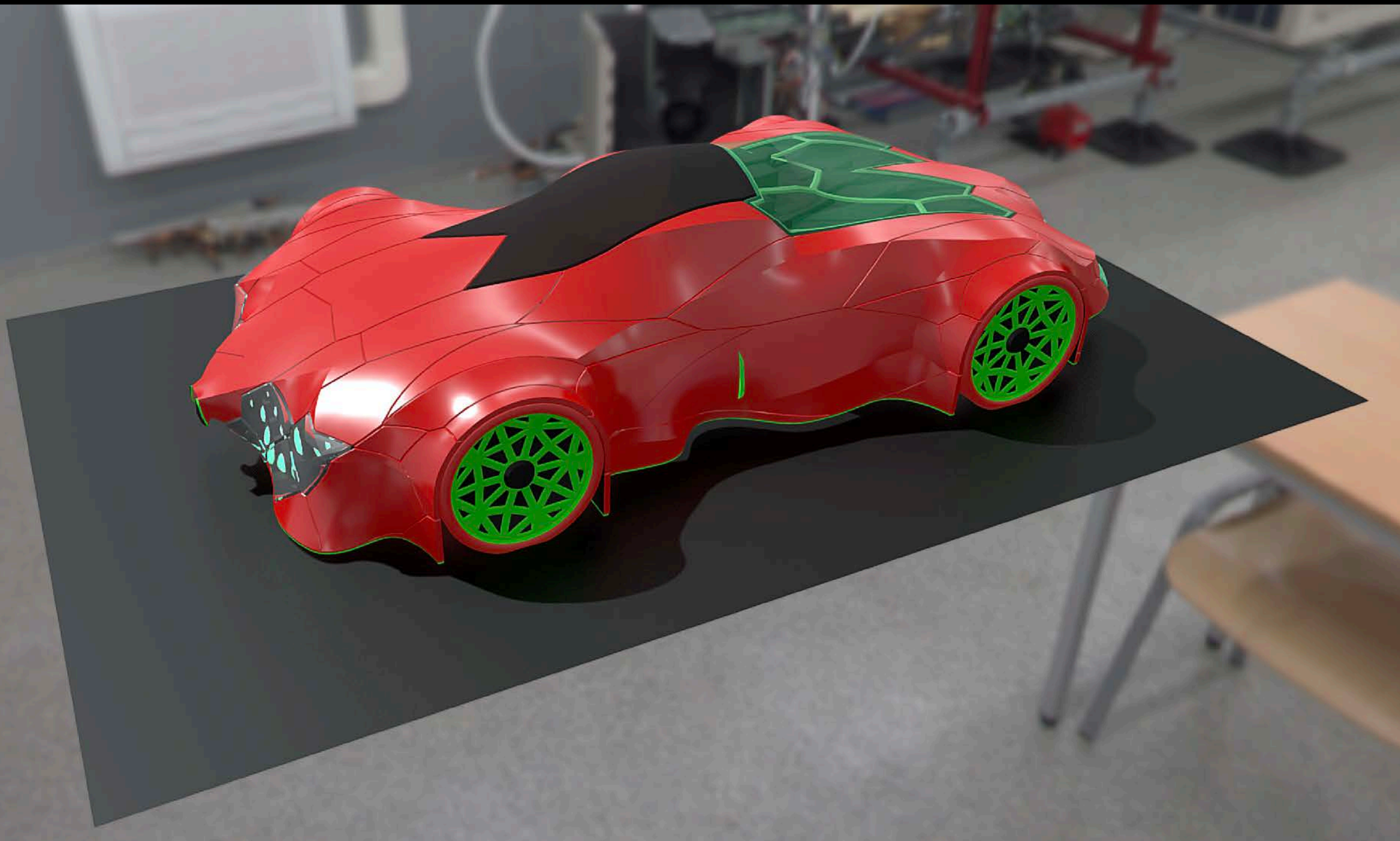
成果2日目 A君



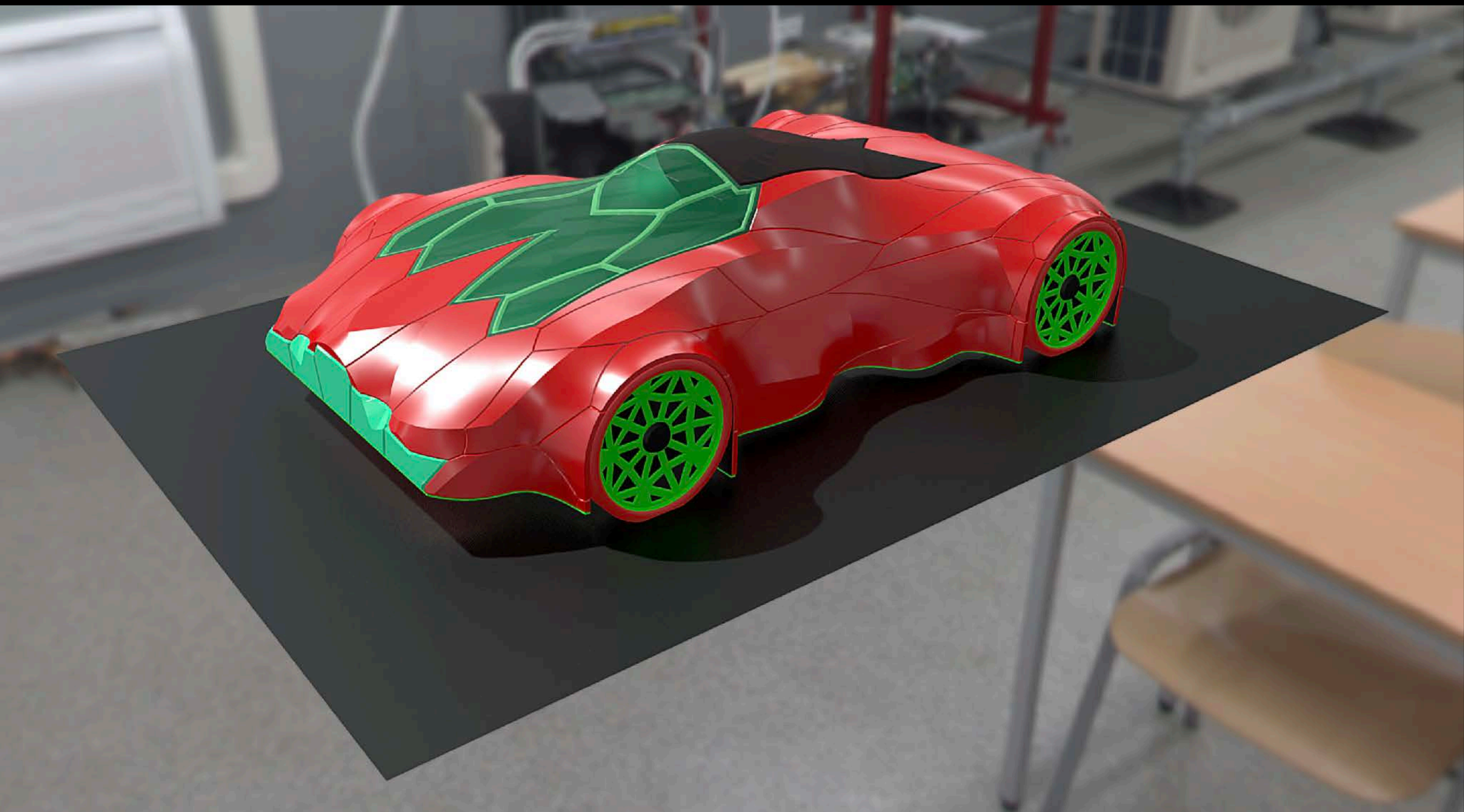
成果2日目 A君



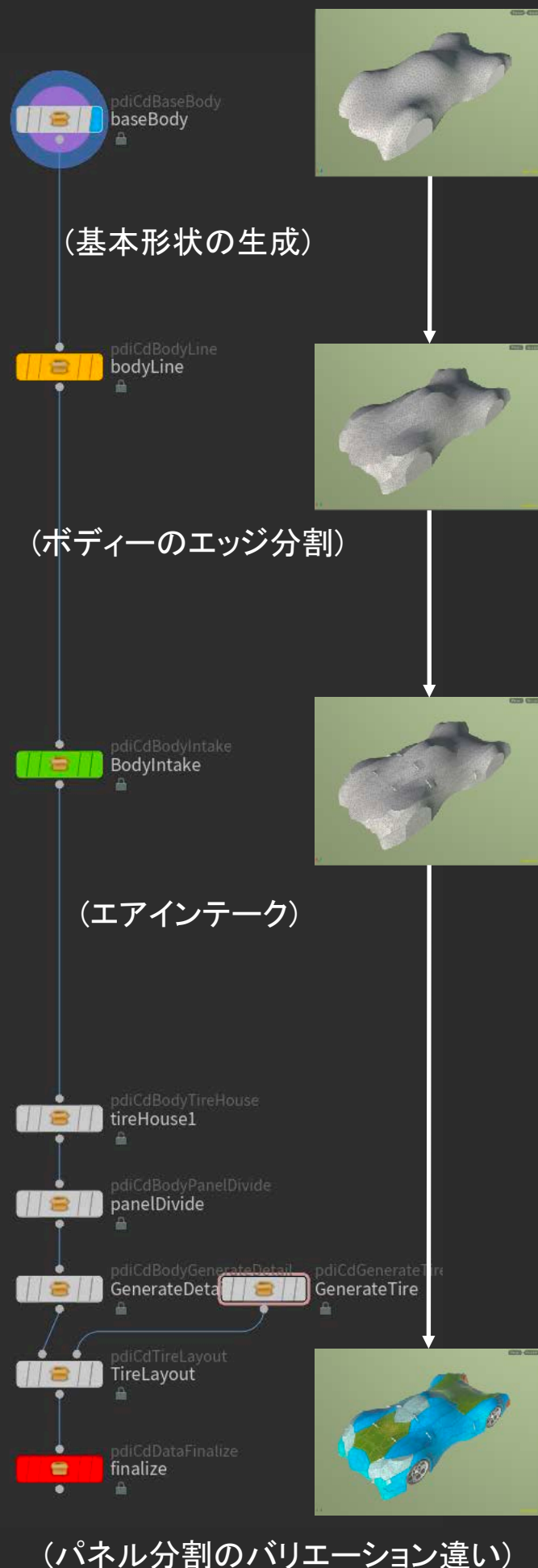
成果2日目 B君



成果2日目 B君

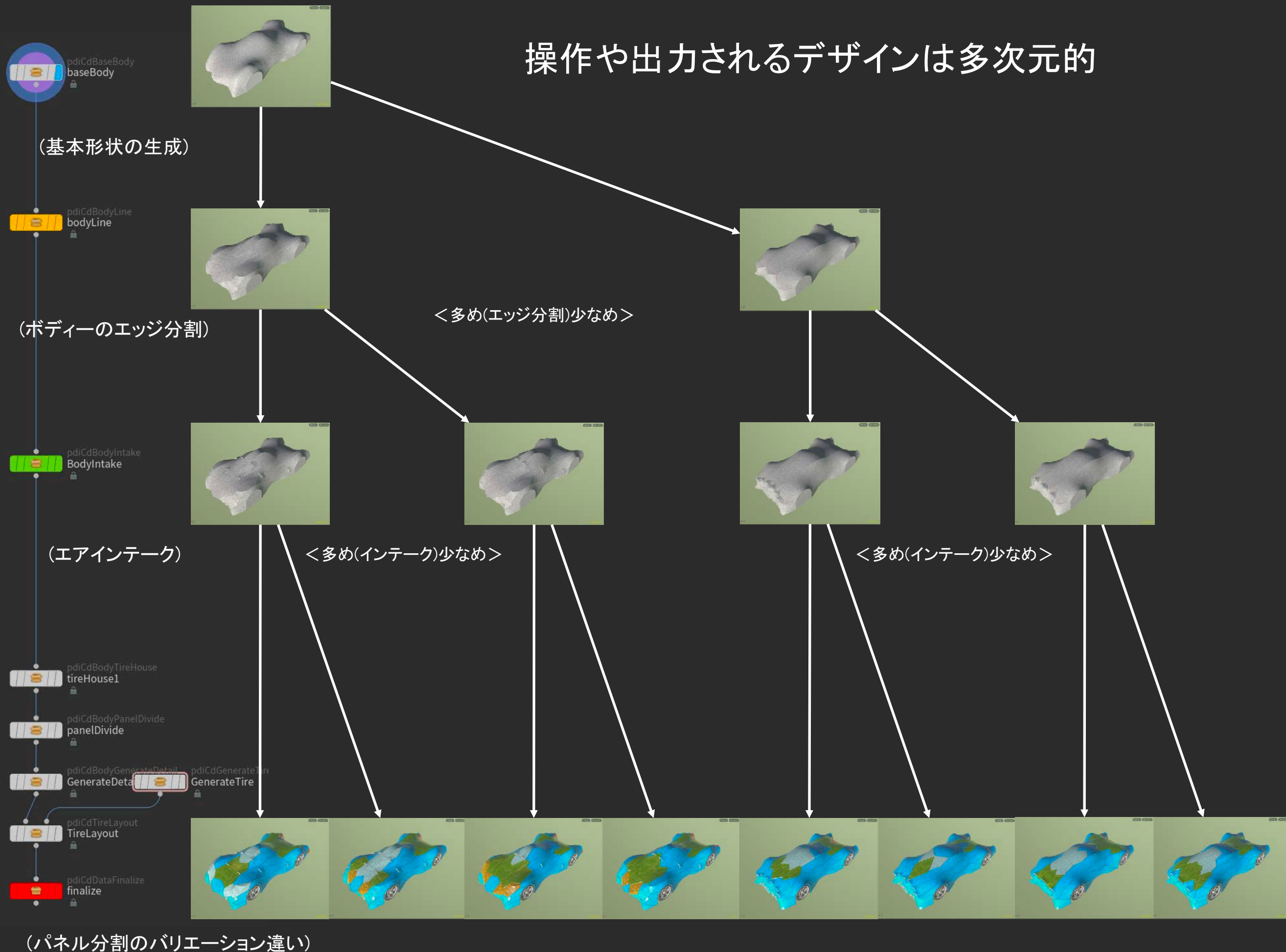


「めんどくさい」を解消するために作られたノードを
直列につないだネットワーク

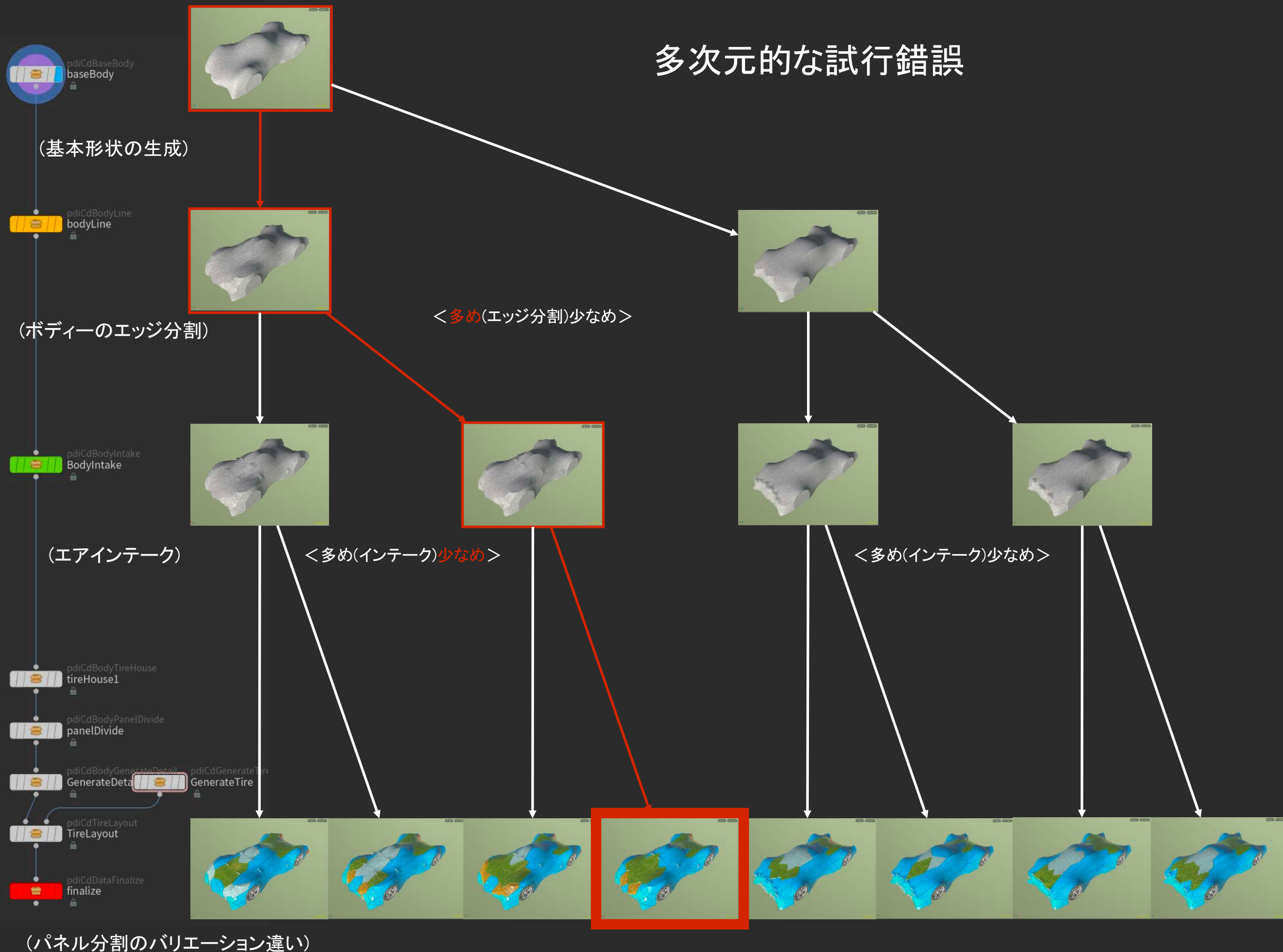


(パネル分割のバリエーション違い)

操作や出力されるデザインは多次元的

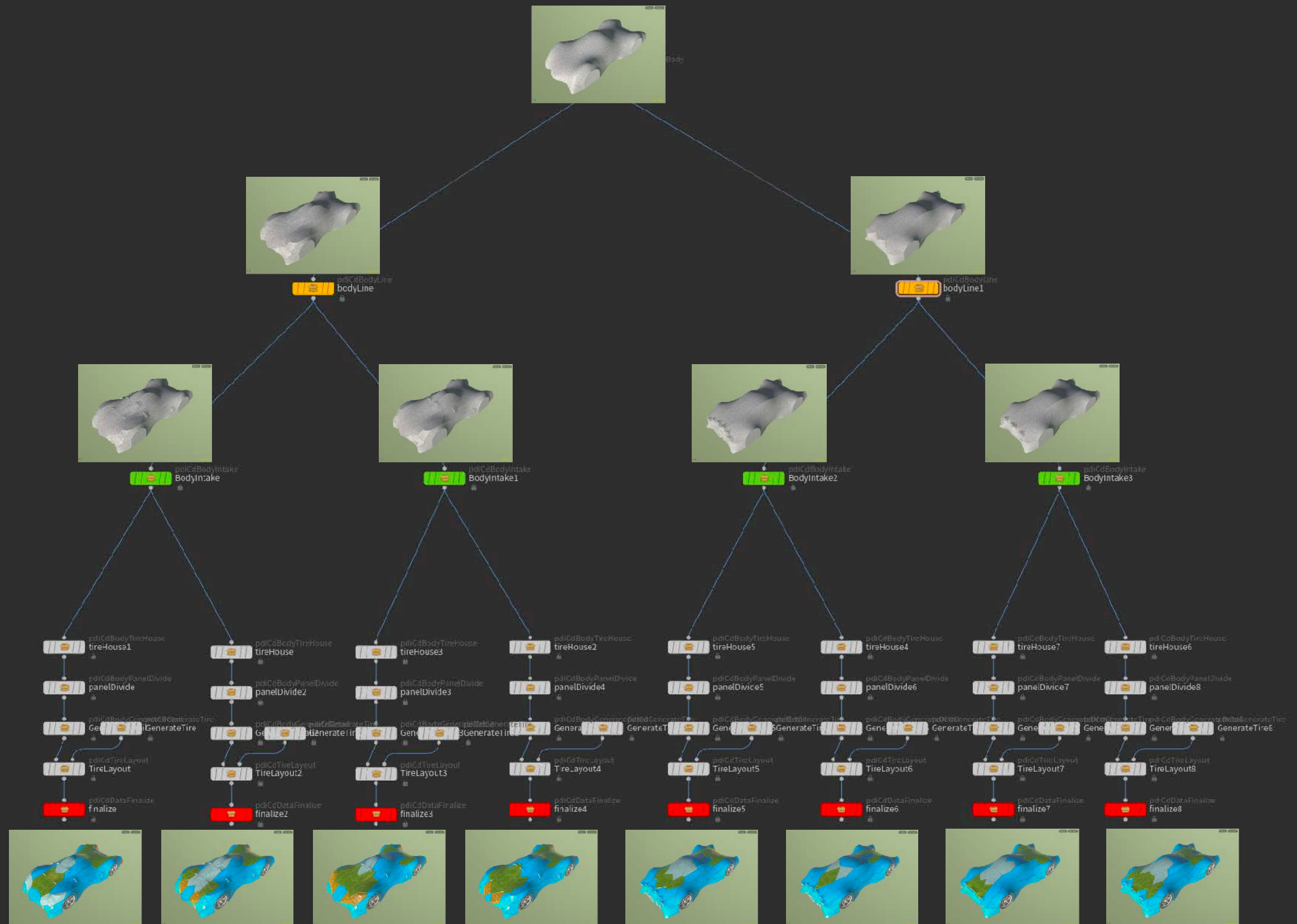


多次元的な試行錯誤



多次元的なデザインバリエーションを即座に俯瞰するようなネットワークも構築可能

Indie Edition



まとめ1

デザインバリエーションの俯瞰

□「めんどくさい」を解消するワークフローを構築して行った結果、多次元的なデザインパターンを俯瞰できる仕組みにたどり着きました。

□多次元的に展開されたデザインを俯瞰しながらのデザインとモデリングに新たな可能性を感じています。

□ランダムシードが大きくデザインに影響しているが運任せなのか？

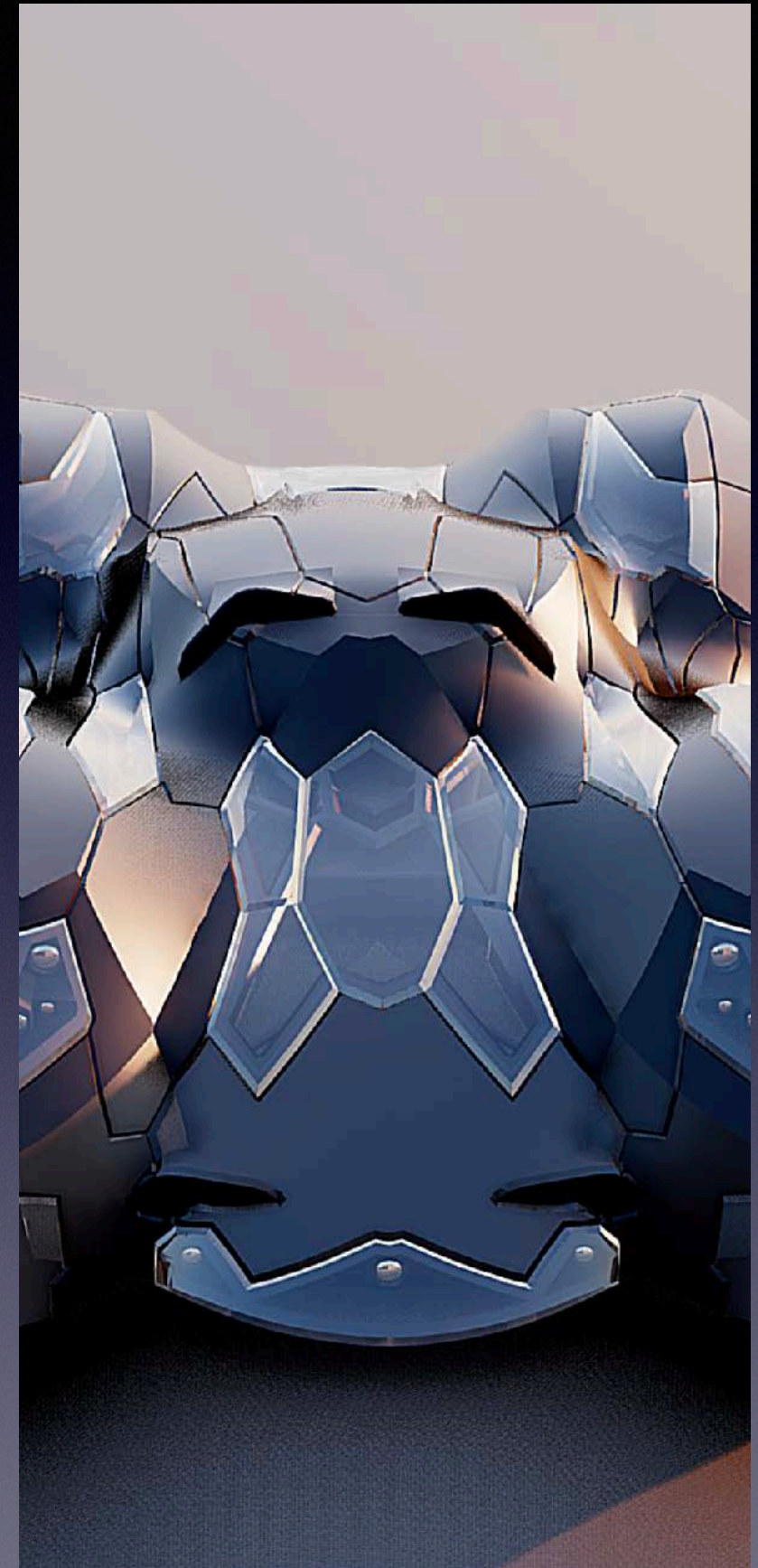
- ・出力される乱数は破綻の無いように、調整、もしくは淘汰される仕組み。

(例)照らさない場所にヘッドライトは現れない。

(例)発射後自身に当たってしまうような場所に兵装は現れない。

- ・デザインの核心はバランスにあるという持論があります。

大きさ、数、角度、等様々な物の比率が乱数を元に決定されていますが、乱数にも偏りが定義されており、そのデフォルト値は黄金比(1:1.618)に。

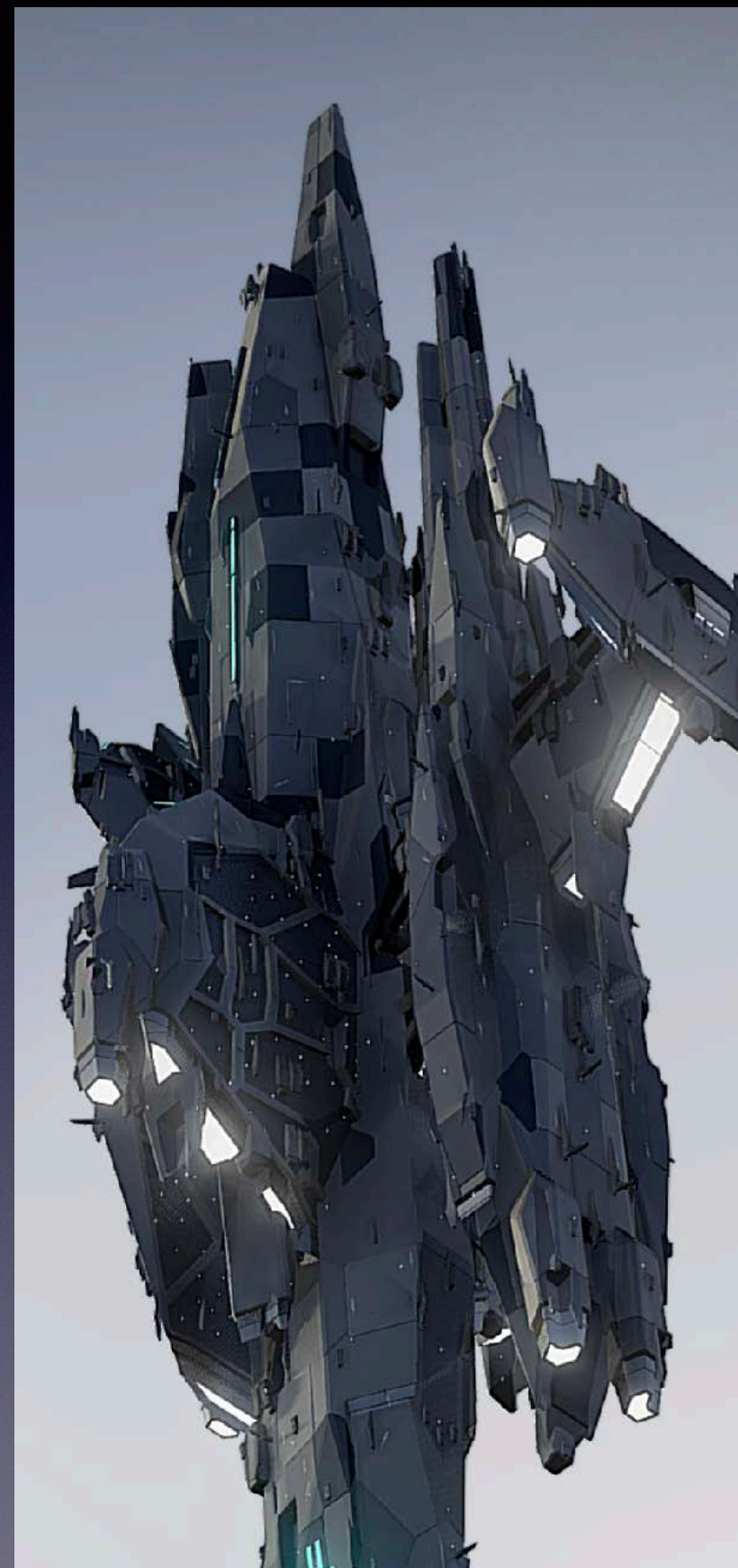


まとめ2 いきなり完成形

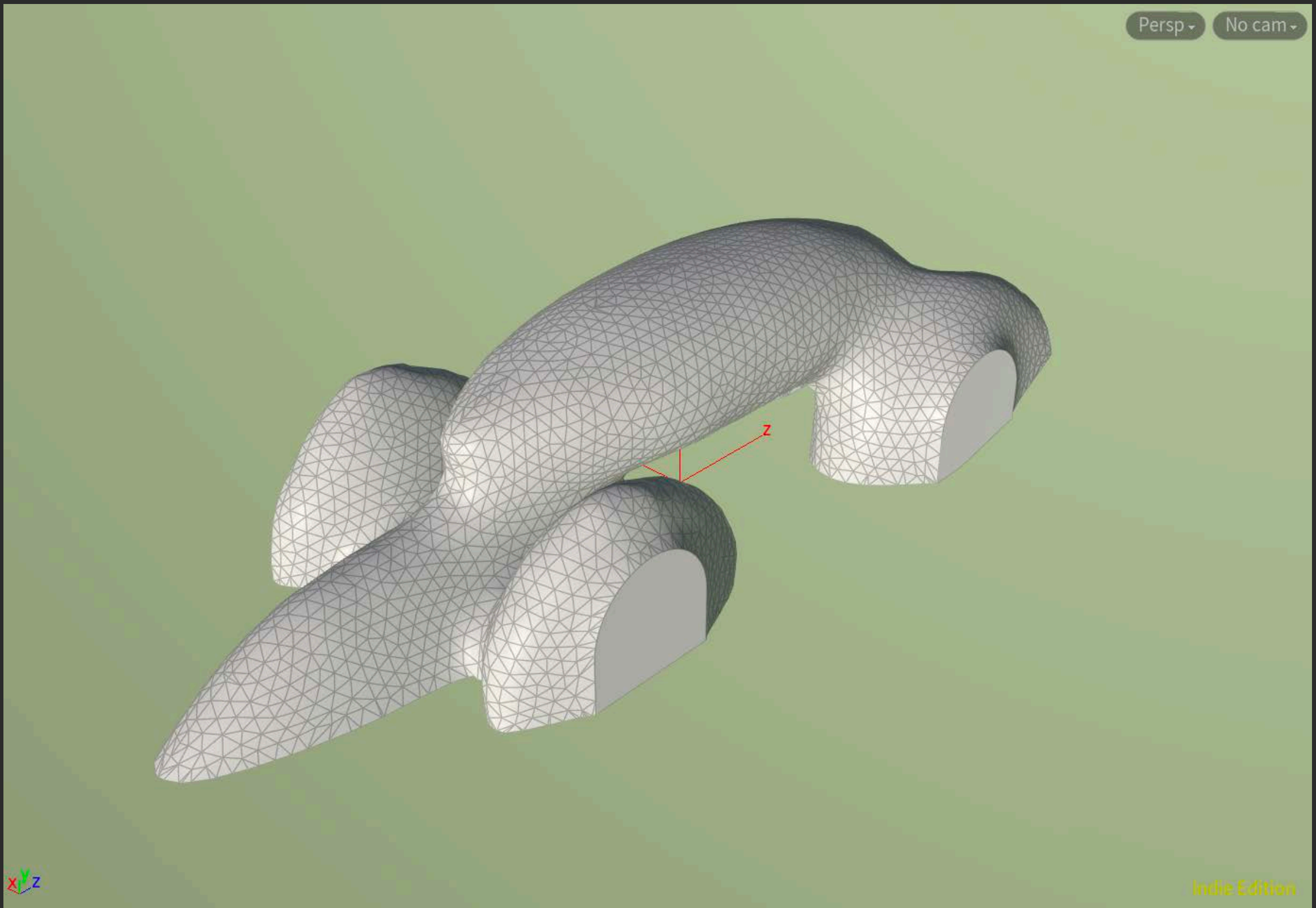
□一見、そのまま仕上げまで進めても良いデザインにならなそうな、シルエットやコンセプトがありますが、実際に最終的な仕上げまで進めると魅力的なデザインに化けることがあります。

□プロシージャル化されたワークフローにより、即座に最終的な仕上げが施されるので、様々なシルエットやコンセプトを即座に評価することが可能になりました。

□これは、非常に感覚的な印象で、実際にワークフローを構築し幾度も形状を生成することで得られた物です。

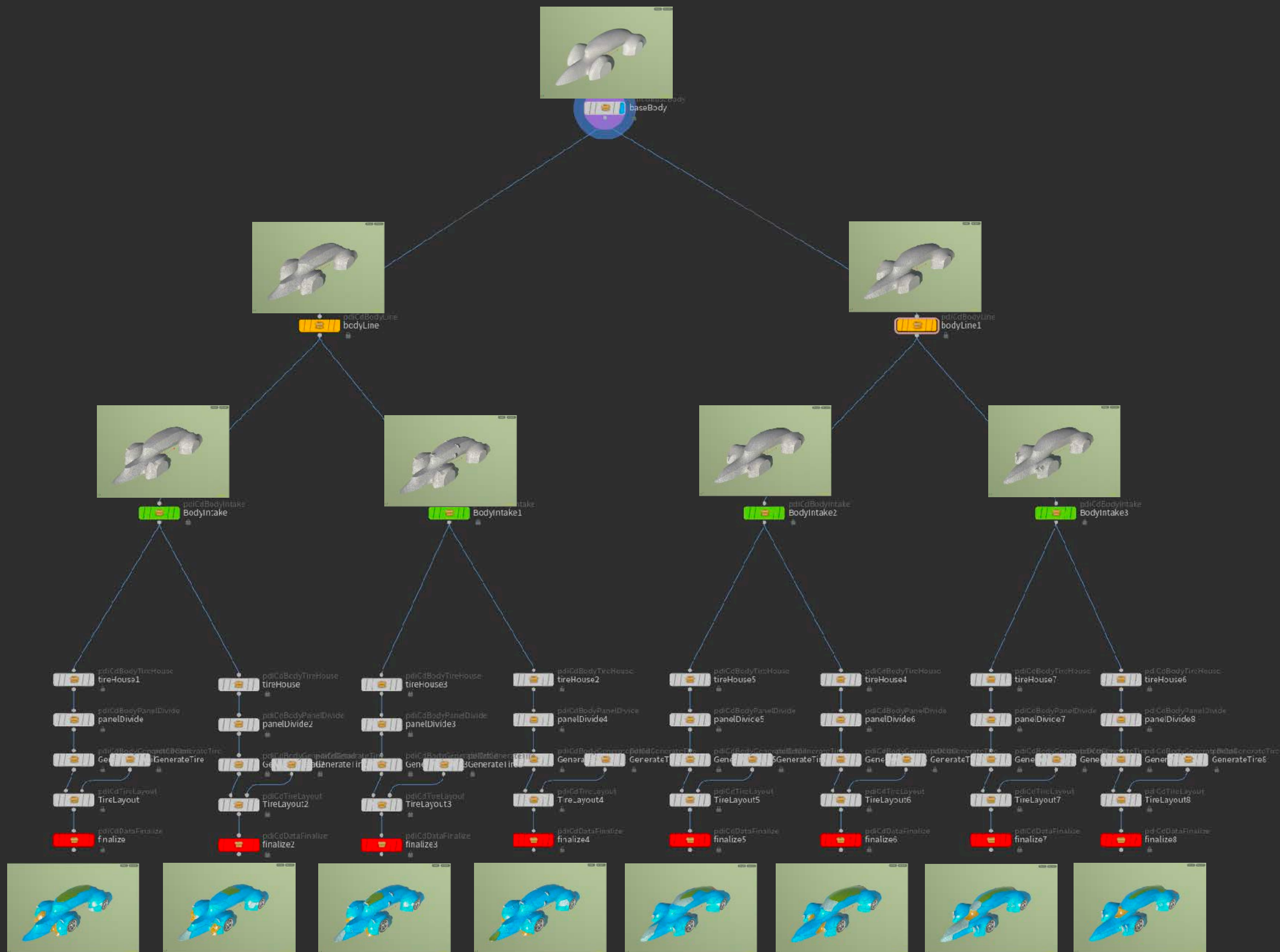


いきなり完成形

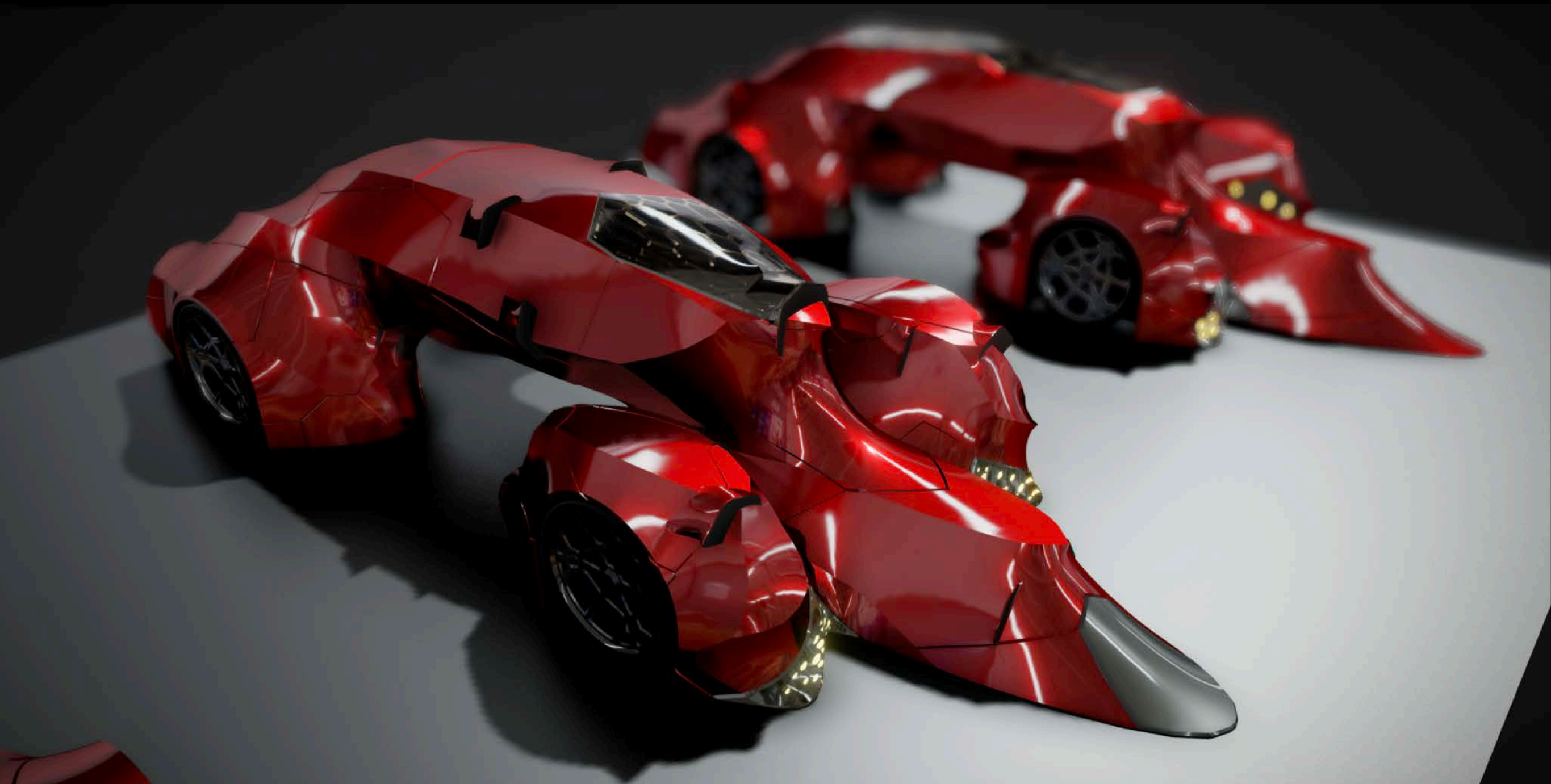


いきなり完成形

Indie Edition



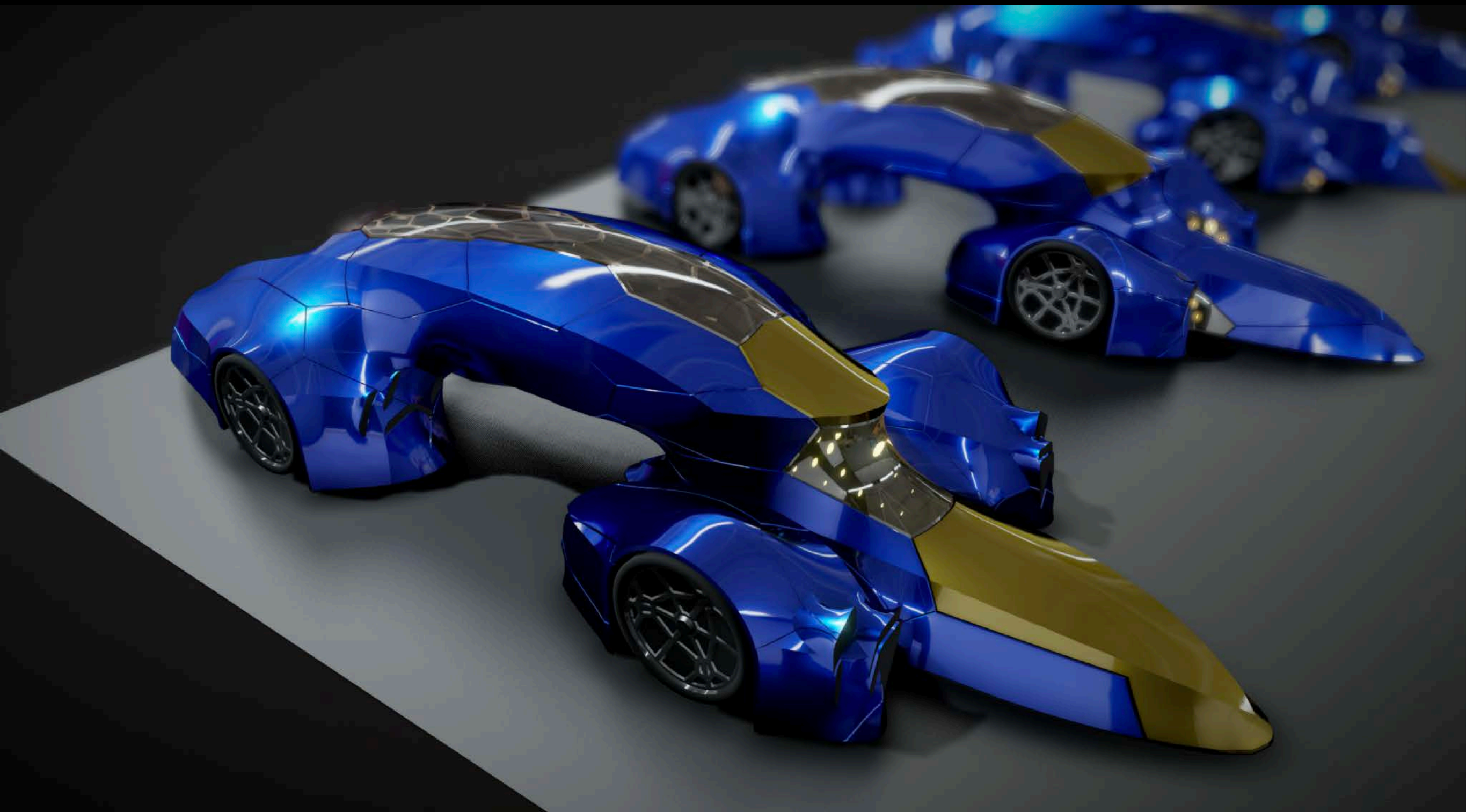
いきなり完成形



いきなり完成形



いきなり完成形



いきなり完成形



いきなり完成形

