

# シェーダー学習の前置き

普段、私達が記述しているプログラムは、CPU上で動作しています。  
CPUは、とても頭が良いので、複雑な処理や様々な用途の処理を行うことができます。  
一方、GPU(Graphics Processing Unit)は、単純処理が得意で、  
並列処理(同時に処理すること)により、CPUよりも早く処理できます。  
こういった特徴から、CPUは1人の天才、GPUは100人の凡人だと例えられます。  
1000問の単純な足し算、掛け算があったとしたら、10問ずつに分担した  
100人の凡人の方が、解くのが早い。逆に、たった1問ですが、  
非常に難解な問題だった場合、天才の方が早いというイメージです。

3Dオブジェクトを、画面に表示する際に、  
何千、何万といった頂点データ(ポリゴン×3)の座標変換処理、奥行き制御、  
透明処理などなど…、1つ1つはそこまで複雑ではありませんが、  
大量の演算処理が必要になりますので、3D処理はGPUに任せた方が良く  
されています。  
このGPU上で動作するプログラムのことをシェーダーと呼びます。

イメージが難しい方は、  
シェーダーとは、グラフィックに特化したプログラムだと思って頂ければ、  
わかり易いと思います。

昔はGPUの性能が今ほど高くはなく、機能の変更などが行えませんでした。  
近年は、私達がGPUで動作するシェーダをカスタマイズできるようになりました。  
これを、プログラマブルシェーダと呼びます。

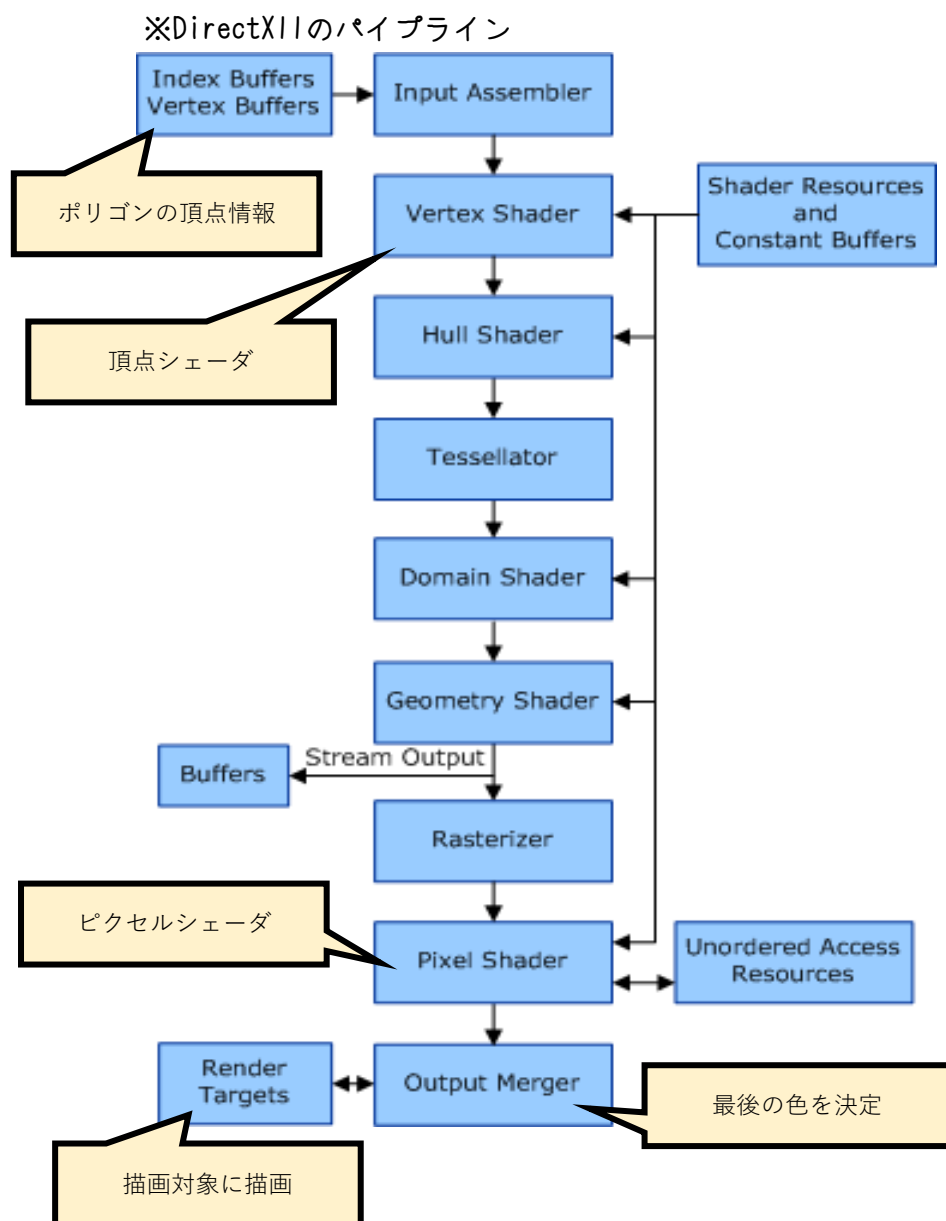
シェーダプログラミングで使われるシェーダ言語はいくつか種類があります。

HLSL      Microsoft社が開発したC言語に似たシェーダ言語。  
DirectXとの親和性が高い。

GLSL      OpenGL(汎用グラフィックスライブラリ)の規格。Khronos Group。  
他にもPS4用のシェーダ言語もあります。Wikiをご参照ください。  
<https://ja.wikipedia.org/wiki/%E3%82%B7%E3%82%A7%E3%83%BC%E3%83%87%E3%82%A3%E3%83%B3%E3%82%B0%E8%A8%80%E8%AA%9E>

シェーダーを扱えるプログラマーは少なく、  
何百人もいるゲーム制作プロジェクトでも、数人で担当したりします。  
3Dの仕組み、3次元の座標、回転、行列がわかっていないと書けないので、  
大変かもしれませんが、  
表現を大きく変えたり、細部を追求することもできますので、  
見た目と技術の点から、企業様の目にも止まりやすくなるでしょう。

もう少し詳しくイメージするために、グラフィック処理の流れ、レンダリングパイプラインを追っていきましょう。



シェーダーの種類がいくつかありますが、主に使用するのが、頂点シェーダとピクセルシェーダーです。

頂点シェーダは、頂点の数だけ処理されます。

頂点カラーを変更したり、頂点座標を移動させたりして、3Dモデルを変形させることもできます。

ピクセルシェーダは、1ピクセルごとに処理(例：1024\*640)され、対象ピクセルの色を決めます。

## 頂点シェーダでできること

頂点を变形させて水の波を表現する



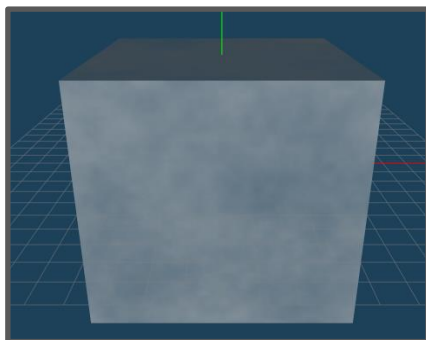
普通の頂点シェーダ



カスタムした頂点シェーダ

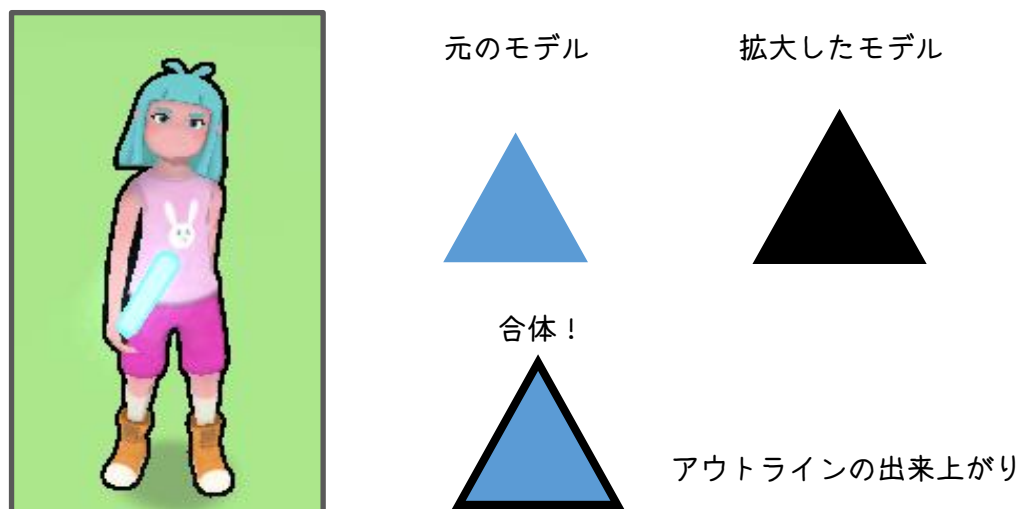


頂点のY座標を变形させて波を表現。



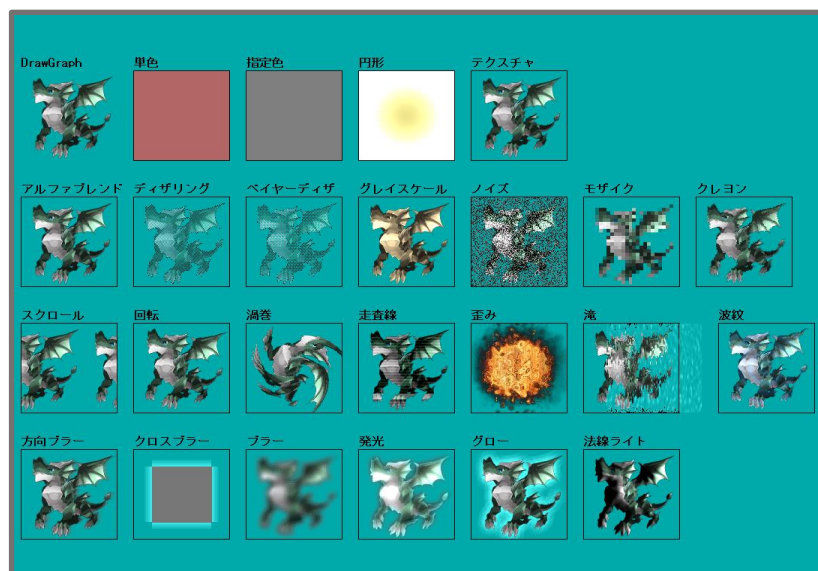
元はただの正方形の3Dモデル。

モデルが大きくなるよう変形させて境界線を作る

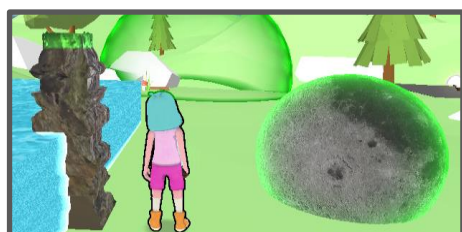


ピクセルシェーダでできること

とにかく色々！



歪ましたり、ぼかしたり、ノイズをかけたり、  
ピクセルを動かして、スクロールや回転させたりできます。



頂点シェーダと組み合わせて  
消したり、周囲を光らせたりもできます。