グラフィックエンジンはとにかくモデルを出すことが基本中の基本である。

これができれば最悪シューティングゲームを作るだけでなんとかごまかせる。

そのための前準備を行う。「ポリゴン描画」だ。

一般的なグラフィックエンジンでの描画は、Rendering Pipeline(レンダリングパイプライン）と呼ばれる、モデルデータを受け取りモニターに２Dの絵として出力するまでのシークエンスを担う処理機構によって行われる

DirectX12におけるレンダリングパイプライン

{

<https://docs.microsoft.com/en-us/windows/win32/direct3d12/pipelines-and-shaders-with-directx-12>

図は、リンク先の"Direct3D graphic pipeline"で参照してほしい。

主なパイプラインの流れは

・ディスクリプタヒープ(**GPUへ送って処理させたいデータの扱い方を記した物を入れたメモリ領域**)から、ディスクリプタテーブル(<https://docs.microsoft.com/en-us/windows/win32/direct3d12/descriptor-tables-overview>

**ディスクリプタヒープをいくつか並べた一次元配列のこと**)へとデータが流れていき、Rootargument(ルートシグネチャのデータを受け取るもの。引数)からデータに応じたシェーダー(定義を説明:)で処理される流れ

・ポリゴンを描画するための頂点を格納する頂点**バッファ**（覚えてますか：**GPU側にリソースのメモリ領域を確保するためのもの**）,後述するが、頂点の処理を最適化するためのインデックスバッファをInput Assemblerステージ(**ポリゴンを描画するのに必要な頂点、直線、図形の形状を含んだデータをシェーダーに送り込む準備段階**)に送り、最低**Vertex Shader**(ポリゴンを出すためのデータを２Dに変換して絵に出すための**座標変換**、Boneと呼ばれるデータを入れてモデルを動かす準備をするための**スキニング,**ポリゴンの形状をある形態に変化させるモーフィング、頂点の**法線**と呼ばれるポリゴンの表面からその真上にかけて存在する直線のベクトルを使った光の当たり具合の計算・・・といったポリゴンの表示周りの処理を行うためのGPU側のプログラム),Pixel Shader(後述)でデータをGPU(後述:)が絵を出すためのデータに翻訳し、StreamOutput()を経てモニター画面に絵を出す

という流れに分類できる

Vertex Shaderの説明(ここから既に定義を詳しく説明する{

座標変換{

}

スキニング{

}

法線{

}

}

ポリゴンとは？

三つの点を直線で結んで三角形にし、その内側を一度白く塗りつぶしたものである。

{

更に説明が必要な場合もあると思われるが、今はこの程度で十分

}

1.Vertex(頂点)の設定

Vertexとは？

{

ポリゴン、それに付け加えられる色彩効果を描画するのに必要な、

・点の３次元、あるいは2次元位置ベクトル

・法線

・texturecoordinate(テクスチャ座標)と呼ばれる3次元色ベクトル

のデータを格納するデータ構造の事である。

}