

# Managed DirectX on C#

～はじめての3D～

レベル: 1くまー

By keichan



## アジェンダ

- はじめに
- DirectXってなに？
- 3Dをコンピュータで表現するには？
- ポリゴンって？
- DEMOを交えていろいろやってみよう

はじめに

- 当セッションはDirectXの事を全くご存じない方を対象としています。
- 数学の知識(行列やベクトル)は可能な限り必要としない内容にしています。
- DEMOで使用するサンプルは DirectX SDK¥Managed¥Direct3D¥Tutorials¥ に収録されているソースを参考にしています。
- それではいってみましょう。

## DirectXってなに？

- 正式名称は「Microsoft DirectX」
- ゲーム及びマルチメディア処理用のAPIの集合
- 機能別に以下のように分類されている
  - DirectX Graphics
  - DirectX Audio
  - DirectX Media
  - DirectInput
  - DirectPlay
- 当セッションはDirectX Graphicsに含まれているDirect3DのAPIのみ使用します

## DirectXってなに？

- Direct3D

- 描画処理をGPUで行うため、高速に描画できる
- GPUからメインメモリにアクセスできない
- だからGPUメモリへ描画に必要な情報を転送しておく必要がある
- CPU側では描画に必要な情報を加工するだけ

3Dをコンピュータで表現するには？



← 本日の目標はここまで表現  
できるようにすることです

どうやって表現してんのよ？

キーワードはポリゴン

ポリゴンって？



- 上図のような三角形
  - 中身は3つの頂点情報を持っている
    - 位置情報、色情報、テクスチャ座標情報、法線情報etc

DEMOを交えていろいろやってみよう

- 三角形を描画
- 回転している三角形を描画
- 筒を描画
- 筒にテクスチャを張って描画







# DEMO1

DEMOを交えていろいろやってみよう

- ポイント

- Direct3Dを利用するには  
Microsoft.DirectX.Direct3D.Deviceを用意する  
必要がある
- レンダリング開始はDevice.BeginScene()
- レンダリング終了はDevice.EndScene()
- 描きたいものはBeginScene()とEndScene()の  
間に記述する必要がある
- Device.Present()で描画先指定

DEMOを交えていろいろやってみよう

- **ポイント**

- 頂点情報は

- Microsoft.DirectX.Direct3D.VertexBufferクラスに格納して使用する

- 頂点はいろいろな情報を持たせることができるので都度必要なデータ型を用いる

- 今回はCustomVertex.TransformColoredを使用

DEMOを交えていろいろやってみよう

- 三角形を描画
- 回転している三角形を描画
- 筒を描画
- 筒にテクスチャを張って描画





# DEMO2

DEMOを交えていろいろやってみよう

- ポイント

- ワールドマトリクス、プロジェクションマトリクス、ビューマトリクスというものが存在する

- ワールドマトリクス

- オブジェクトのワールド座標系における状態

- プロジェクションマトリクス

- 透視投影行列

- 3次元空間を2次元スクリーンに変換する為に必要

- ビューマトリクス

- カメラのワールド座標系における状態

DEMOを交えていろいろやってみよう

- ポイント

- バックバッファのクリア時にZbufferの指定をお忘れなく

- Device.Clear(ClearFlags.Target | **ClearFlags.ZBuffer**, System.Drawing.Color.Blue, 1.0f, 0);



## DEMOを交えていろいろやってみよう

- 三角形を描画
- 回転している三角形を描画
- 筒を描画
- 筒にテクスチャを張って描画







# DEMO3

DEMOを交えていろいろやってみよう

- **ポイント**

- ライトには複数の種類が用意されている

- **平行光源ライト**

- 一定の方向に無限遠に伸びる光

- **点光源ライト**

- 設置点から放射状に広がる光

- **スポットライト**

- 懐中電灯を照らしたような感じの光

DEMOを交えていろいろやってみよう

- ポイント

- ポリゴンにライトを当てる際には法線が必要

- 法線の向きと光の向きとを計算して物体の明るさを表現しているため

- マテリアル

- ポリゴンの見た目を決定する素材
- 頂点単位ではなく、オブジェクト単位に色を設定できる

## DEMOを交えていろいろやってみよう

- 三角形を描画
- 回転している三角形を描画
- 筒を描画
- 筒にテクスチャを張って描画





# DEMO4

DEMOを交えていろいろやってみよう

- **ポイント**

- それぞれの頂点にテクスチャ座標を指定する
- Device.TextureStateクラス内のパラメータによって描画処理が変化する
  - 加算合成、乗算合成など
  - 複数のテクスチャを重ね合わせることでイケてる表現ができるようになる