

# DirectX勉強会 第2回

# 内容

- ポリゴンの描画
- テクスチャの貼り付け
- 半透明ポリゴンの描画

# ポリゴンの描画1

- DirectXでは物を描くのにポリゴンを使う
- すべてのポリゴンは3角形によって描く
- 3角形は頂点を指定することで描く
- 頂点のフォーマットは自分で作ることができ、サンプルプログラムでは次のようなフォーマットになっている
  - struct TLVERTEX
  - {
  - float x,y,z;                   頂点の座標を表している
  - float rhw;                    プロジェクション頂点変換に関係する
  - D3DCOLOR color;            頂点の色を指定する
  - float tu,tv;                テクスチャ座標を指定する
  - };

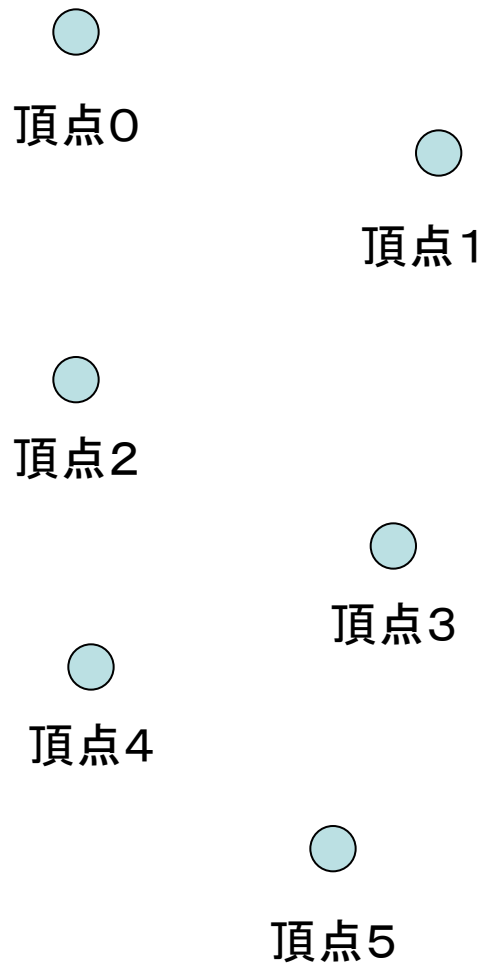
# ポリゴンの描画2

- まず頂点を準備する
- TLVERTEX vertex[4];によって頂点を4つ宣言し、  
値を代入することで頂点の場所などが決まる
- ポリゴンの描画は開始を宣言する  
lpD3DDEV->BeginScene();  
と描画の終わりを宣言する  
>EndScene();  
lpD3DDEV-  
の間でする
- lpD3DDEV->DrawPrimitiveUP  
でポリゴンを描画する

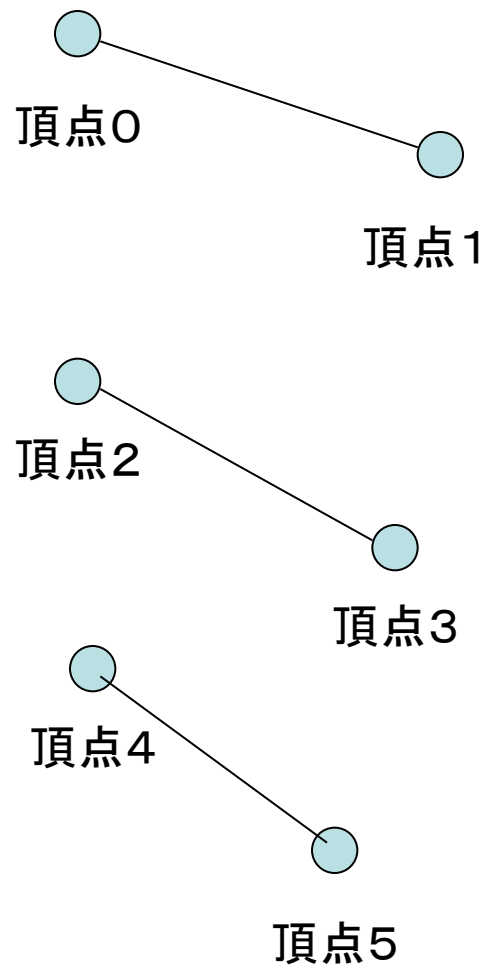
# ポリゴンの描画3

- lpD3DDEV->DrawPrimitiveUP関数の第1引数でポリゴンの描画法を指定する
  - D3DPT\_POINTLIST
    - 指定された頂点はすべて点として描画する
  - D3DPT\_LINELIST
    - 別々の線のリストとして描画する
  - D3DPT\_LINESTRIP
    - 連続した線分のリストとして描画する
  - D3DPT\_TRIANGLELIST
    - 3頂点ずつの別々の頂点として描画する
  - D3DPT\_TRIANGLESTRIP
    - 連続した3角形のリストとして描画する
  - D3DPT\_TRIANGLEFAN
    - 1つの頂点を共有した連続した3角形のリストとして描画する
- lpD3DDEV->DrawPrimitiveUP関数の第2引数は3角形ポリゴンの数
- lpD3DDEV->DrawPrimitiveUP関数の第3引数は頂点の配列へのポインタ
- lpD3DDEV->DrawPrimitiveUP関数の第4引数は1頂点あたりのサイズ

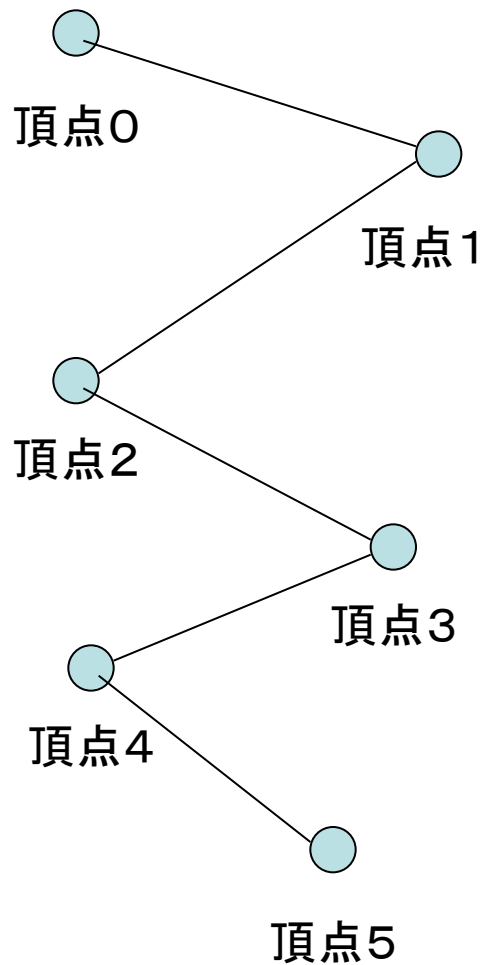
# D3DPT\_POINTLIST



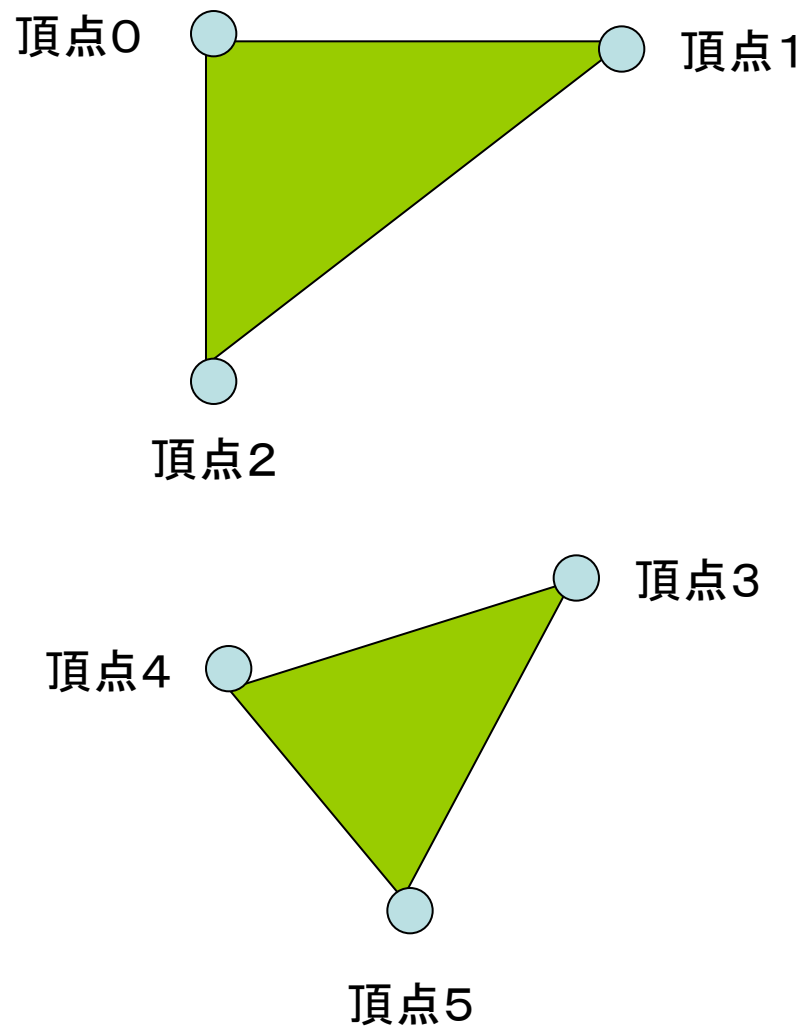
# D3DPT\_LINELIST



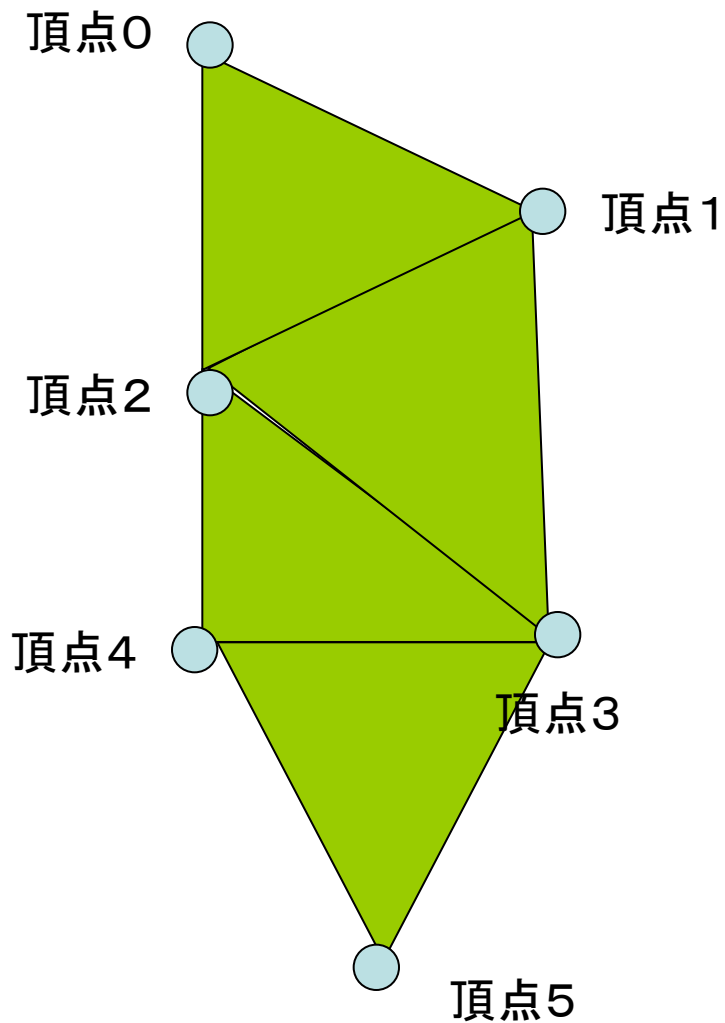
# D3DPT\_LINESTRIP



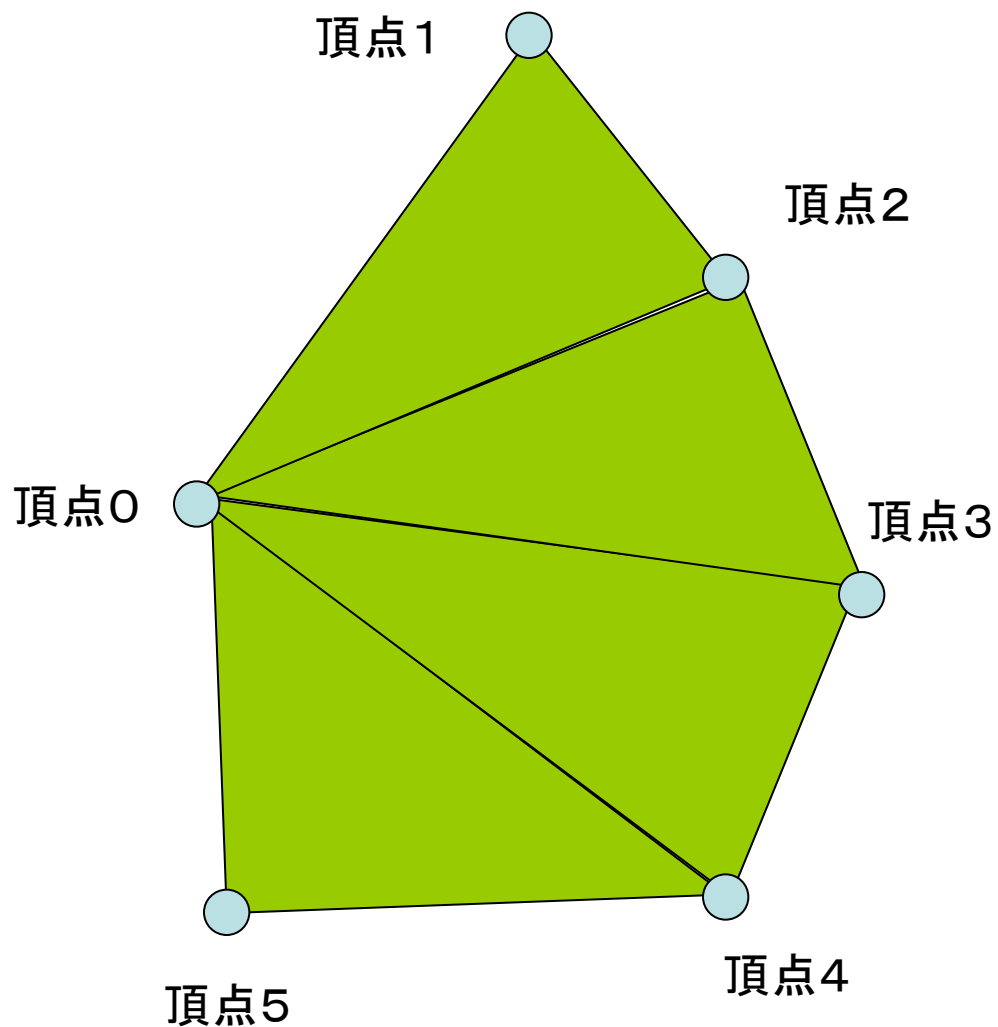
# D3DPT\_TRIANGLELIST



# D3DPT\_LINESTRIP

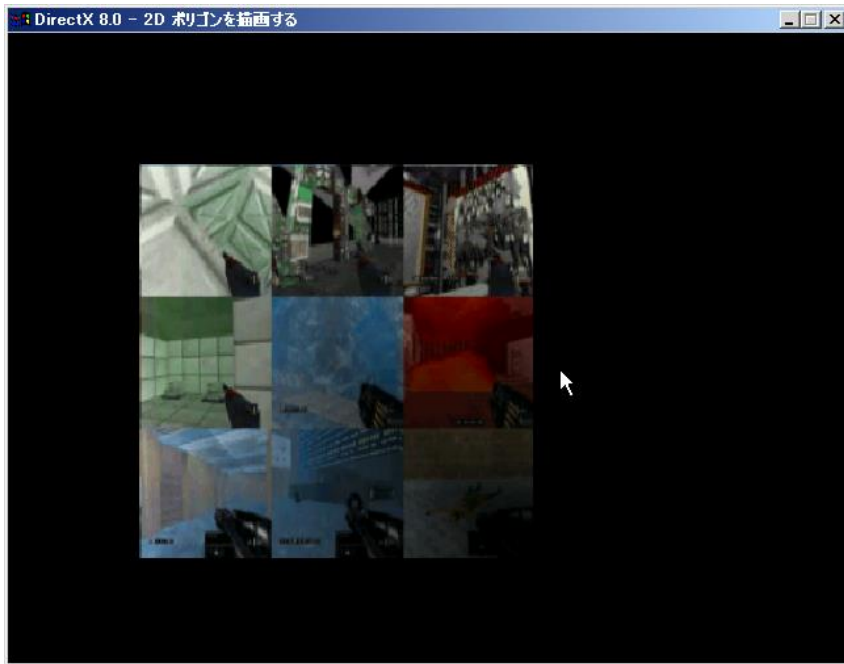


# D3DPT\_TRIANGLELIST

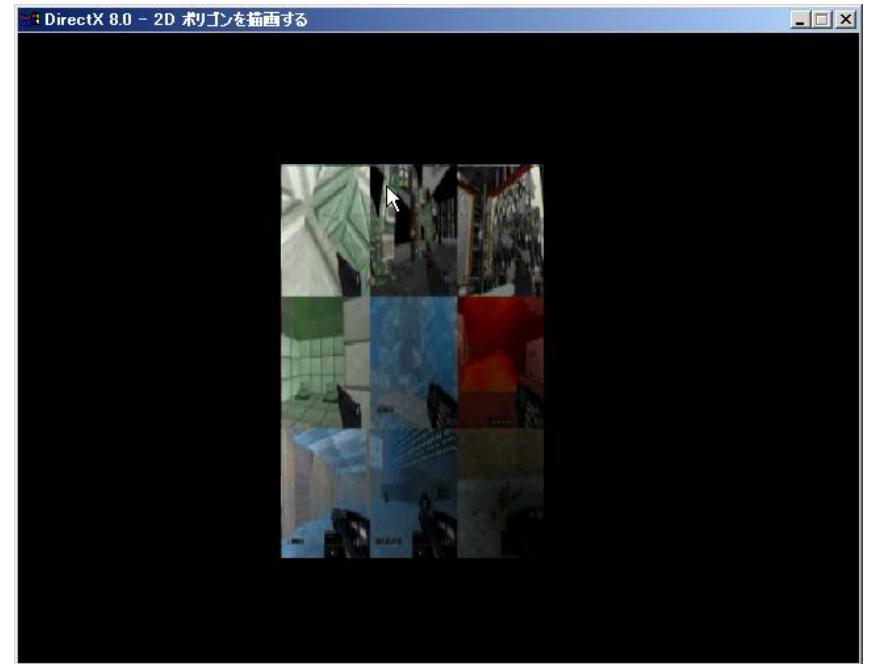




# サンプルプログラム(Poly2D)



元のまま ( vertex[0]とvertex[3]の  
X座標は100 )



vertex[0]とvertex[3]のX座標を200  
にした場合

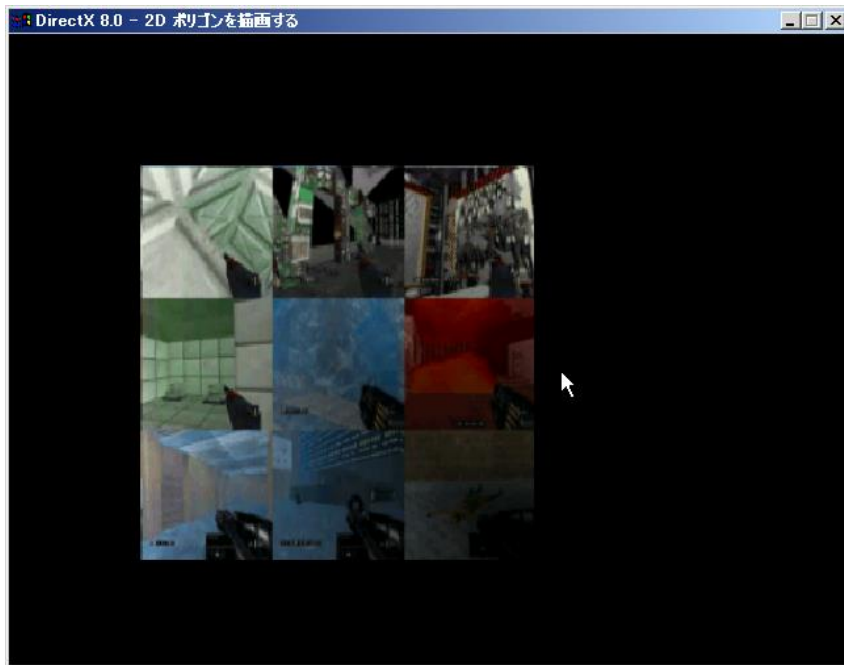
# テクスチャの貼り付け1

- D3DXCreateTextureFromFileEx関数を使って貼り付けたいテクスチャを読み込む
  - D3DXCreateTextureFromFileEx関数の第2引数に読み込みたい画像ファイルのファイル名を指定する
  - D3DXCreateTextureFromFileEx関数の第14引数にテクスチャオブジェクトへのポインタを指定する

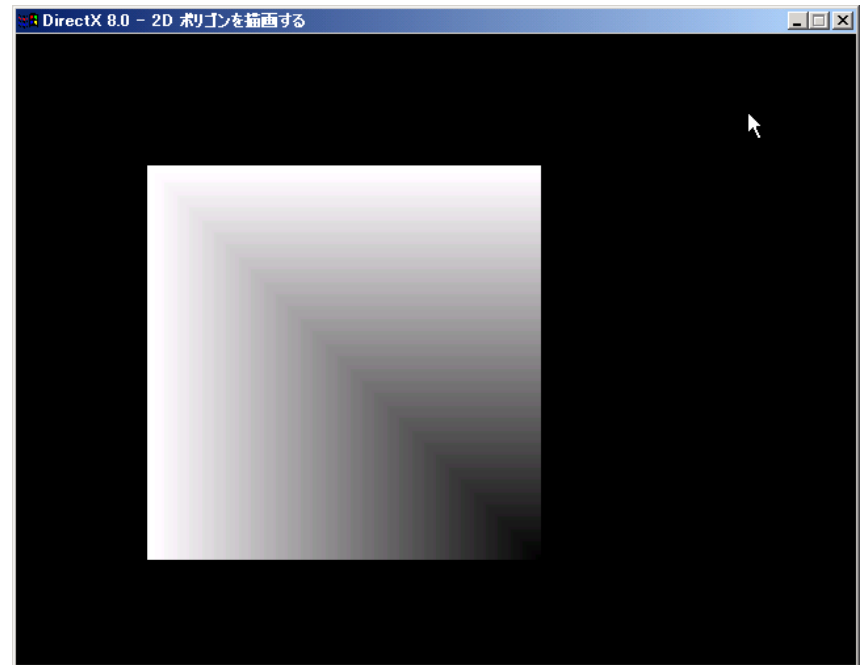
# テクスチャの貼り付け2

- `lpD3DDEV->SetTexture`関数でテクスチャの設定をする
  - 第2引数のポインタでテクスチャを指定する
  - NULLにするとテクスチャは使われなくなる
- 頂点のフォーマットでテクスチャを指定する部分があるが $(tu, tv)$ で $(0, 0)$ はテクスチャの左上、 $(1, 0)$ は右上の様に $0 \sim 1$ の値で $tu$ は横方向、 $tv$ は縦方向を表す

# サンプルプログラム(Poly2D)



元のまま(テクスチャとして  
texture.bmpを貼り付ける)



テクスチャオブジェクトへのポインタ  
をNULLにした場合

# 半透明ポリゴンの描画1

- lpD3DDEV->SetRenderState関数でアルファ・ブレンディングをONにすることで半透明にすることができる
- lpD3DDEV->SetRenderState関数で透明化の方法を指定できる
  - D3DRS\_SRCBLENDをD3DBLEND\_SRCALPHAに、D3DRS\_DESTBLENDをD3DBLEND\_INVSRCALPHAにすることで普通の半透明にできる(透明度を考慮した値になっている)
  - D3DRS\_SRCBLENDをD3DBLEND\_ONEに、D3DRS\_DESTBLENDをD3DBLEND\_ONEにすることで加算合計による半透明にできる(重なった色を単純に足している)

# 半透明ポリゴンの描画2

- 透明化をするためには色の表現をRGBの3つの変数だけではなく、RGBAの4つの変数で表現する
  - Aは透明度を表す
  - D3DCOLOR\_RGBA(r,g,b,a)で定義する
  - RGBだけの表現はD3DCOLOR\_XRGB(r,g,b)で定義する

# サンプルプログラム(Poly2DCalc)



元のまま(普通の半透明)



加算合計による半透明