Porting shader from DX9 to DX11

1/Semantics changes

Pixel shader position input must now be SV_Position, so we change from

```
struct vs2ps
{
    float4 Pos : POSITION;
    float4 TexCd : TEXCOORD0;
};
```

to

```
struct vs2ps
{
    float4 Pos: SV_POSITION;
    float4 TexCd: TEXCOORD0;
};
```

Pixel shader output semantic is now SV_Target[n] instead of COLOR[n]

```
float4 PS(vs2ps In): COLOR
```

becomes

```
float4 PS(vs2ps In): SV_Target
```

Techniques are changed to technique10 and technique11, so we change:

```
technique TConstant
{
    pass P0
    {
        VertexShader = compile vs_2_0 VS();
        PixelShader = compile ps_2_0 PS();
    }
}
```

to

```
technique10 Constant
{
       pass P0
       {
              SetVertexShader( CompileShader( vs_4_0, VS() ) );
              SetPixelShader( CompileShader( ps_4_0, PS() ) );
       }
}
```

If vertex/pixel shader requires shader model 5, we use:

```
technique11 Constant
{
       pass P0
       {
              SetVertexShader( CompileShader( vs_5_0, VS() ) );
              SetPixelShader( CompileShader( ps_5_0, PS() ) );
       }
}
```

Please note that in dx11 shader using technique10 and technique11 are both supported.

2/Texture and sampler changes

In DirectX 9, each sampler is assigned to a texture. In DirectX10, they are separate units. Also sampler syntax has changed, please check

http://msdn.microsoft.com/en-us/library/windows/desktop/ff476207(v=vs.85).aspx

To see all sampler options.

```
texture Tex <string uiname="Texture";>;
sampler Samp = sampler_state    //sampler for doing the texture-lookup
{
    Texture   = (Tex);          //apply a texture to the sampler
    MipFilter = LINEAR;         //sampler states
    MinFilter = LINEAR;
    MagFilter = LINEAR;
};
```

Becomes:

```
Texture2D texture;

SamplerState g_samLinear
{
    Filter = MIN_MAG_MIP_LINEAR;
    AddressU = Wrap;
    AddressV = Wrap;
};
```

If we need a second texture, we can simply add, since they can use the same samplers.

```
Texture2D texture2;
```

To sample textures, syntax has also slightly changed:

```
float4 col = tex2D(Samp, In.TexCd.xy);
```

becomes

```
float4 col = texture2d.Sample( g_samLinear, In.TexCd.xy);
```

```
Sampling levels is now done like this:
```

```
float4 col = tex2Dlod(Samp, float4(In.TexCd.xy,0,level));
```

becomes

```
float4 col = texture2d.SampleLevel( g_samLinear, In.TexCd.xy,level);
```

3/VVVV Related changes

Those changes are not part of standard DirectX, but are also breaking changes compared to DirectX 9 implementation.

Semantics now always hide pins. A semantic now means it is provided downstream, so we never create a pin for it.

That means that color and texture transform pins are now different, pin convolution is now using annotations.

For colors:

```
float4 cAmb : COLOR <String uiname="Color";>  = {1, 1, 1, 1};
```

Becomes

```
float4 cAmb <String uiname="Color"; bool color=true;> = { 1.0f,1.0f,1.0f,1.0f };
```

For texture matrices:

```
float4x4 tTex: TEXTUREMATRIX <string uiname="Texture Transform";>;
```

becomes

```
float4x4 tTex <string uiname="Texture Transform"; bool uvspace=true;>;
```