

Shader integration with Curved World system

1. Include **CurvedWorld_Base.cginc** file inside vertex shader pass.

```
#include "Assets/VacuumShaders/Curved World/Shaders/cginc/CurvedWorld_Base.cginc"
```

(This is file path by default, it may be different)

2. Inside vertex shader pass use one of the two Curved World macros:

- **CURVED_WORLD_TRANSFORM_POINT (float4 vertex)** – Transforms only vertex, suitable for unlit shaders.
- **CURVED_WORLD_TRANSFORM_POINT_AND_NORMAL (float4 vertex, float3 normal, float4 tangent)** – Transforms vertex and normal, suitable for shaders requiring correctly rotated normal for calculating: light, shadow, reflection etc.

If vertex shader pass does per-vertex animation, displace, wind or other per-vertex effects then Curved World transformation must be used after them.

Steps 1 and 2 must be used in all vertex shader passes.

3. (Optional step) If shader uses **Fallback** shader then it also must be modified for Curved World or can be used one of the built-in shaders provided by Curved World:

- **"Hidden/VacuumShaders/Curved World/VertexLit/Diffuse"** – for opaque shaders.
- **"Hidden/VacuumShaders/Curved World/VertexLit/Cutout"** – for cutout (alpha test) shaders.
- **"Hidden/VacuumShaders/Curved World/VertexLit/Transparent"** – for transparent shaders.

4. (Optional step) If shader requires camera Depth and Normal textures for image effects then must be defined custom **RenderType** or used one of the Curved World's built-in RenderTypes:

- **"RenderType"="CurvedWorld_Opaque"** – for opaque shaders.
- **"RenderType"="CurvedWorld_TransparentCutout"** – for cutout shaders.

That's all.

Check two example shaders inside Shaders/Example folder:

1. **"Custom/Example_Unlit"**
2. **"Custom/Example_Surface"**