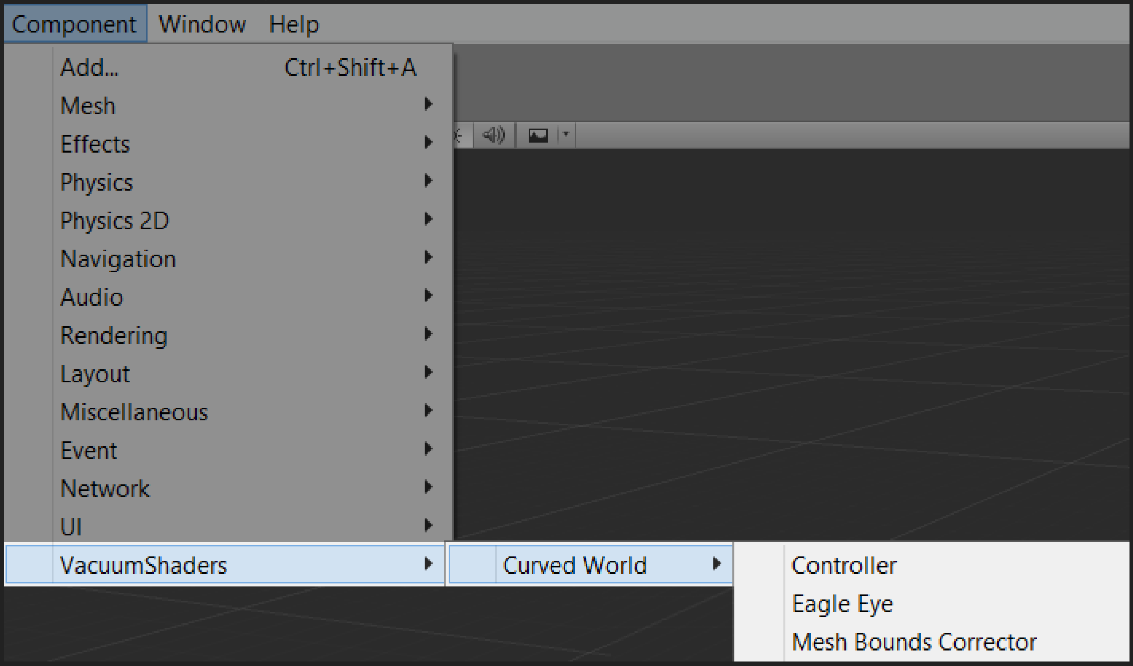
Curved World API

Curved World package contains three component scripts that can be used inside project.



- Controller – Updates bend parameters for all shaders using Curved World vertex transformation. Scene must contain one instance of this script.
- Eagle Eye – Overrides camera's field of view parameter for rendering meshes outside its view frustum. Solves mesh disappearing problem.
- Mesh Bounds Corrector – Scales mesh render bounds, if it is not visible to camera or light source.

All scripts are inside `VacuumShaders.CurvedWorld` namespace.

CurvedWorld_Controller

Script updates three shader parameters inside **CurvedWorld_Base.cginc** file.

```
uniform float3 _V_CW_Bend;
uniform float3 _V_CW_Bias;
uniform float4 _V_CW_PivotPoint_Position;
```

Those parameters can be updated manually by using Unity [Material.SetVector]() methods or use **CurvedWorld_Controller** script built-in methods and parameters

Public variables:

- For controlling bend size per axis
  1. `public float _V_CW_Bend_X = 0;` – X axis bend size control
  2. `public float _V_CW_Bend_Y = 0;` – Y axis bend size control
  3. `public float _V_CW_Bend_Z = 0;` – Z axis bend size control
- For controlling bend size bias per axis
  1. `public float _V_CW_Bias_X = 0;`
  2. `public float _V_CW_Bias_Y = 0;`
  3. `public float _V_CW_Bias_Z = 0;`
- Pivot point

  `public Transform pivotPoint;` - If not defined world (0, 0, 0) coordinate is the center of the bend.

  For Perspective2D pivot point always is screen center of active camera.
- Static singleton

  `static public CurvedWorld_Controller get` – Returns reference to the scene's active `CurvedWorld_Controller`.

Public functions:

- `public Vector3 GetBend()` – Returns axis bend size as Vector3
- `public void SetBend(Vector3 _newBend)` – Sets axis bend size from Vector3

- `public Vector3 GetBias()` – Returns axis bend size bias as Vector3
- `public void SetBias(Vector3 _newBias)` – Sets axis bend size bias from Vector3

Curved World shader vertex transformation can be used only on meshes using materials.
Game objects like Colliders, Lights, Projectors and other such objects position cannot be modified by shader, but sometime it is necessary to do it.

- `public Vector3 TransformPoint(Vector3 _transformPoint, BEND_TYPE _bendType)`
  Modifies world space Vector3 point position using **CurvedWorld_Controller** parameters.

Public static method:

- `static public Vector3 TransformPoint(Vector3 _transformPoint, BEND_TYPE _bendType, Vector3 _bendSize, Vector3 _bendBias, Vector3 _pivotPoint)`
  Modifies world space Vector3 point position using custom parameters.

Check 2. Little Planet (Nightmare) example scene.
Point lights and some game objects position there are updated using Follow script (using `TransformPoint` method) to be the same as bend effect created by Curved World shaders.

CurvedWorld_EagleEye

The only public variable - `public float fieldOfView = 60;`

CurvedWorld_MeshBoundsCorrector

The only public variable - `public float meshBoundsScale = 1;`