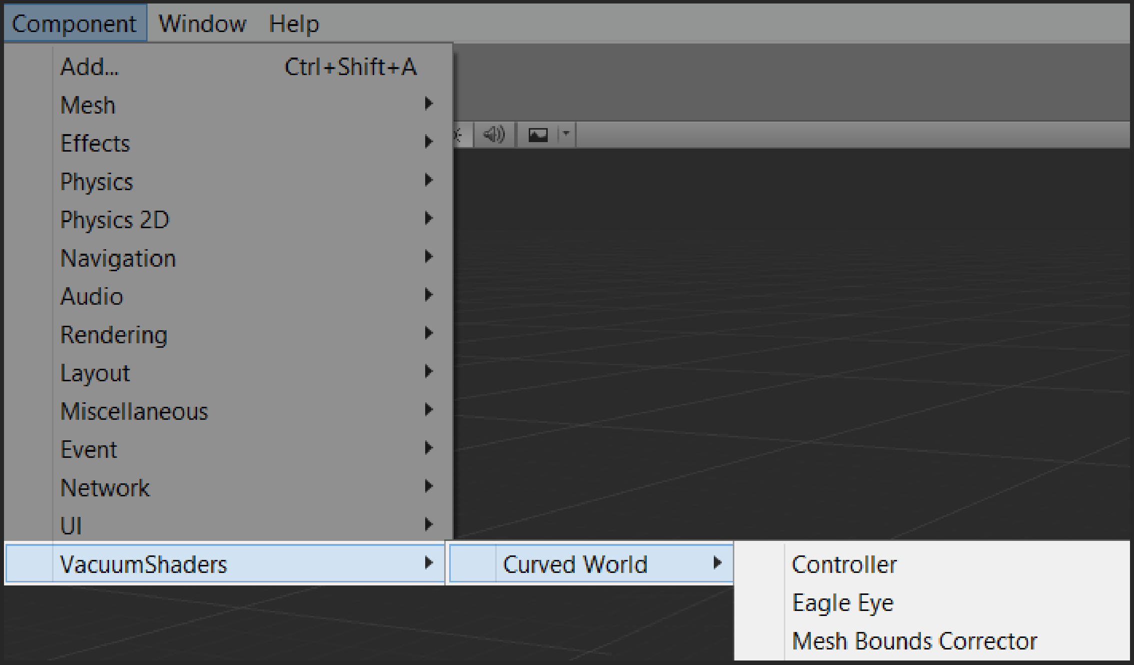
Curved World API

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



- Controller – 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 three scripts are inside `VacuumShaders.CurvedWorld` namespace.

CurvedWorld_Controller

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 (0, 0, 0) is the center of the bend. For Perspective2D pivot point always is screen center of active camera.

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

- `public Vector3 TransformPoint(Vector3 _transformPoint, BEND_TYPE _bendType)` – Takes Vector3 as world space position and bends it using **CurvedWorld_Controller** parameters.

Public static functions:

- `static public Vector3 TransformPoint(...)` – Takes Vector3 as world space position and bends it using custom parameters.

CurvedWorld_EagleEye

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

CurvedWorld_MeshBoundsCorrector

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