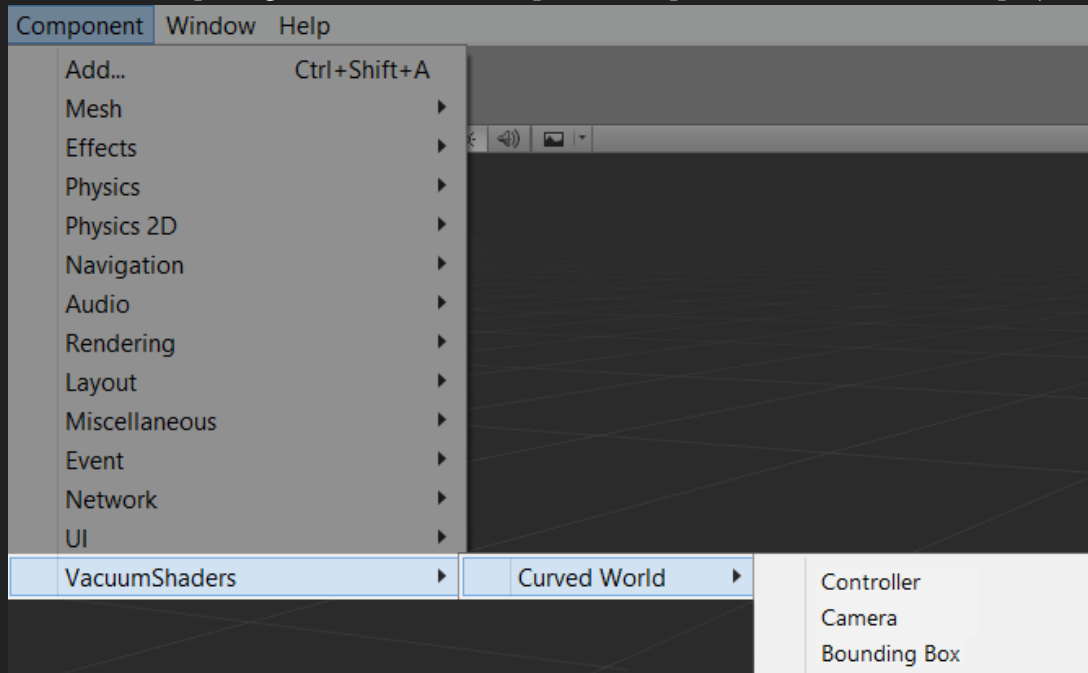


# Curved World API

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 have only one active instance of this script.
- Camera – Overrides camera's field of view parameter for rendering meshes outside its view frustum. Solves mesh disappearing problem.
- Bounding Box – Scales individual **non- static** render's bounding box. Solves mesh disappearing problem and makes renderer visible to a light source shadow cast/receive pass.

All scripts are inside `VacuumShaders.CurvedWorld` namespace.

## CurvedWorld\_Controller

`static public CurvedWorld_Controller current` – Returns reference to the active `CurvedWorld_Controller` script of a scene.

`public float upDownSize, upDownOffset;`  
`public float leftRightSize, leftRightOffset;`  
Parameters for adjusting **Classic Runner** bend effect.

`public float curvatureSize, curvatureOffset`  
Parameters adjust curvature size and offset for **Little Planet**, **Cylindrical Tower** and **Cylindrical Rolloff** bend effects.

`public Vector3 bendSize, bendOffset`  
Parameters control per-axis (X, Y, Z) bend size and offset for **Universal** and **Perspective 2D** bend effects.

`public float angle, angle_2;`  
`public float minimalRadius, minimalRadius_2;`  
`public float rolloff;`  
Parameters control shape of **Spiral** bend effects.

`public Transform pivotPoint, pivotPoint_2;`  
Curvature pivot points.

```
public void DisableBend ()
```

Used to disable all bending effects.

Note: This does not affect shader performance, variables are just updated not to create curvature.

```
public void EnableBend ()
```

Restores curvature parameters disabled by `DisableBend` method.

```
public void Reset ()
```

Resets `CurvedWorld_Controller` variables to their defaults.

```
public Vector3 TransformPosition (Vector3 _transformPosition, BEND_TYPE _bendType)
```

Modifies world space Vector3 value using `CurvedWorld_Controller` parameters to follow Curved World curvature.

Useful for updating Colliders, Lights, Projects and other objects position.

```
public Quaternion TransformRotation (Vector3 _transformPosition, Vector3 _transformForward, Vector3 _transformRight, BEND_TYPE _bendType)
```

Calculates rotation that follows curvature created by shaders.

Check [8. Follow \(Script Bending\)](#) example scene.

Objects position there is updated using `Follow` script, to follow curvature created by Curved World shaders.

## CurvedWorld\_Camera

The only public variable - `public float fieldOfView;`

## CurvedWorld\_BoundingBox

The only public variable - `public float scale;`