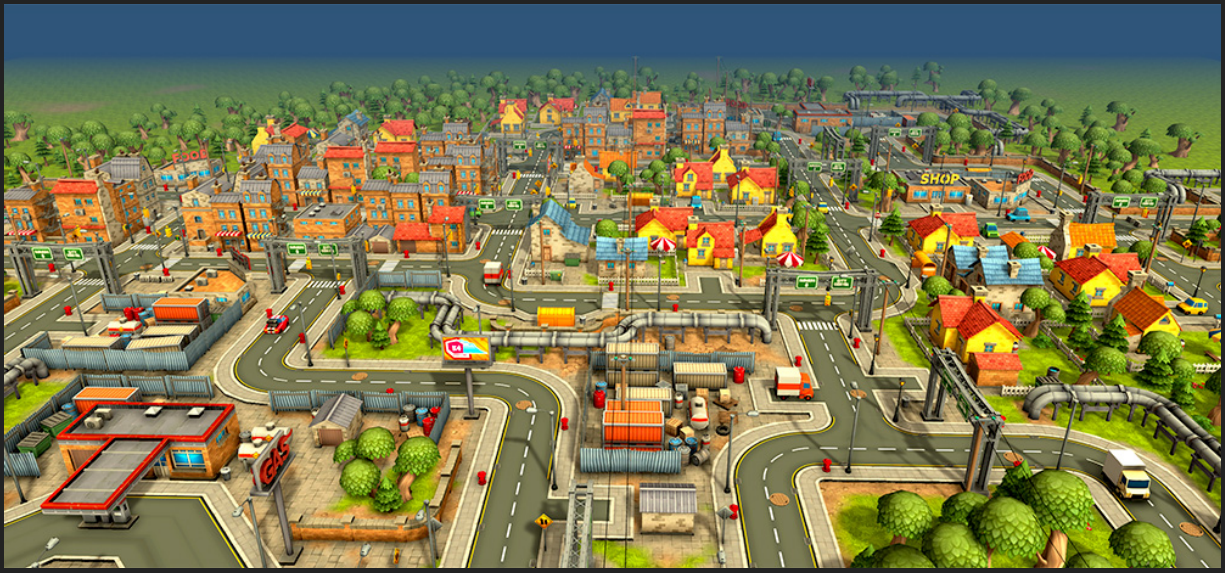
What is Curved World and how it works

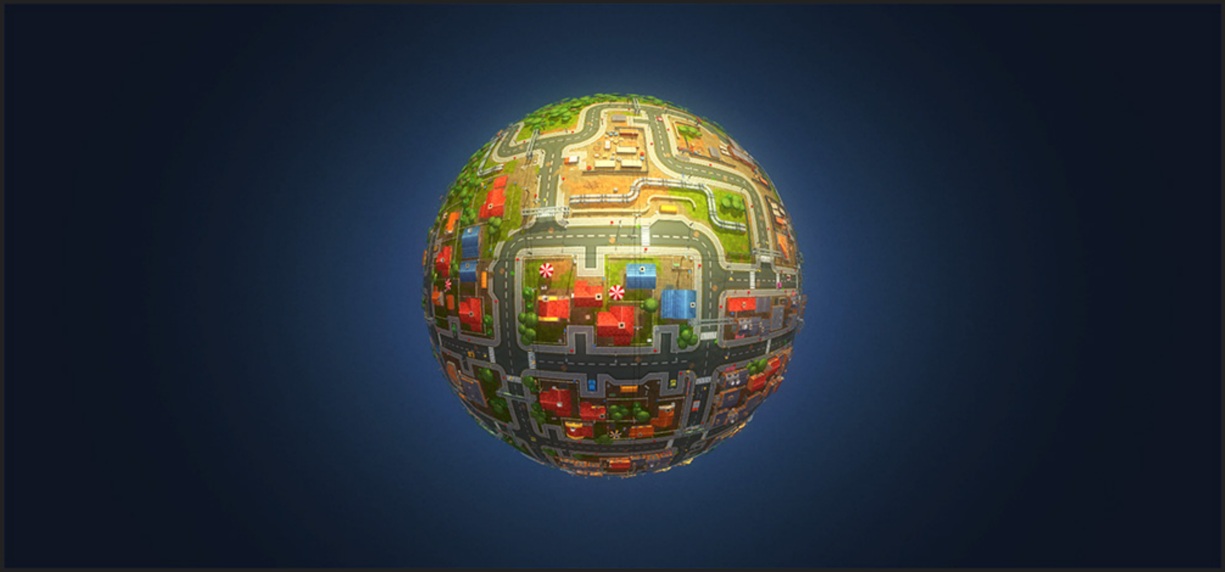
Curved World is a vertex transformation shader for creating various shader bending effects.
It is like displace or heighmap shader where texture is used to adjust mesh vertex position, but instead of a texture Curved World uses mesh vertex world-space position to calculate how it will be transformed.

Being a shader Curved World does not modify real mesh it renders. If for example mesh is 'flat' before using Curved World shader it still will be 'flat' after, just rendered differently. Because of it physics, animations, path finding and other game features are not effect and do not need any modifications.

If object needs to be moved from position A to position B along path C, after using Curved World shader everything will be the same. Just rendered differently.



Real scene



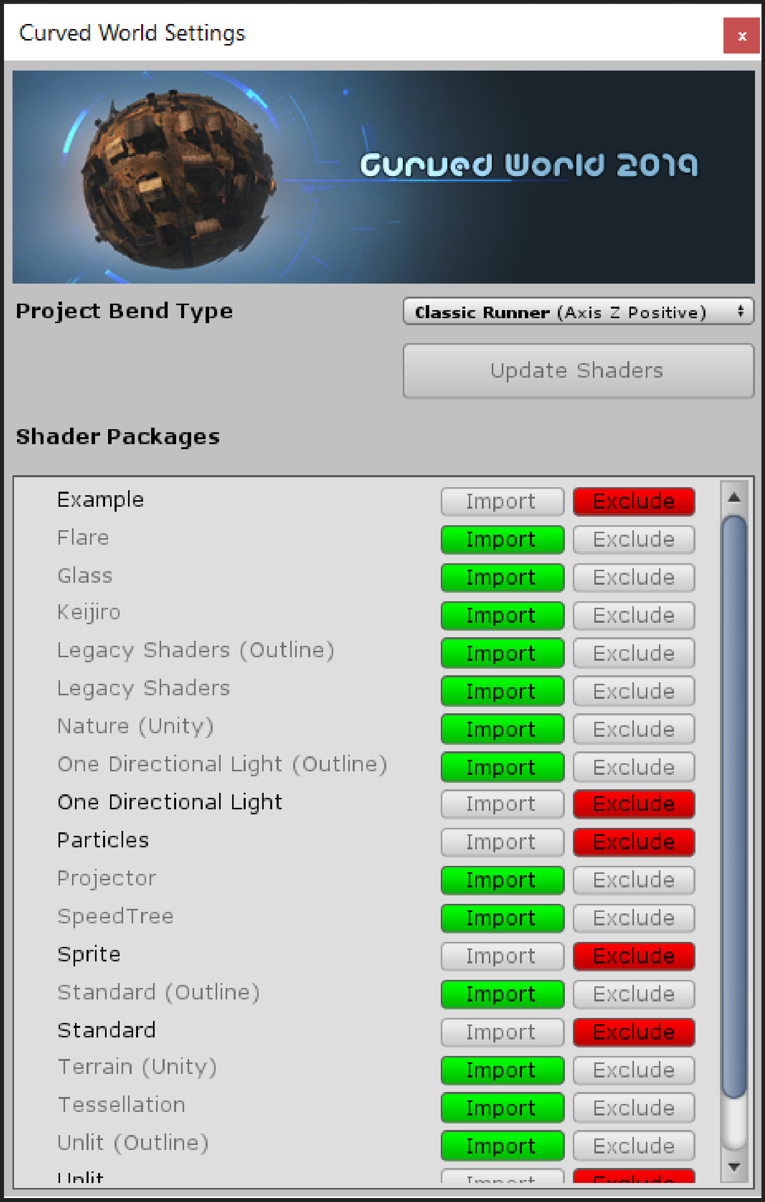
Rendered

Let’s have a quick look on an example to understand core features of the Curved World.

Open Tutorial (Runer) example scene and run it.
It is a simple runner type scene. Objects are spawned at some distance from the player, moved along path (Z axis in this case) and are destroyed behind the camera. Nothing related to the Curved World.

Unleash Curved World in three easy steps:

1) Open Menu -> Window -> VacuumShaders -> Curved World Settings window and make sure **Classic Runner (Axis Z Positive)** bend type is active. If not, select it and push Update Shaders button. Close window.



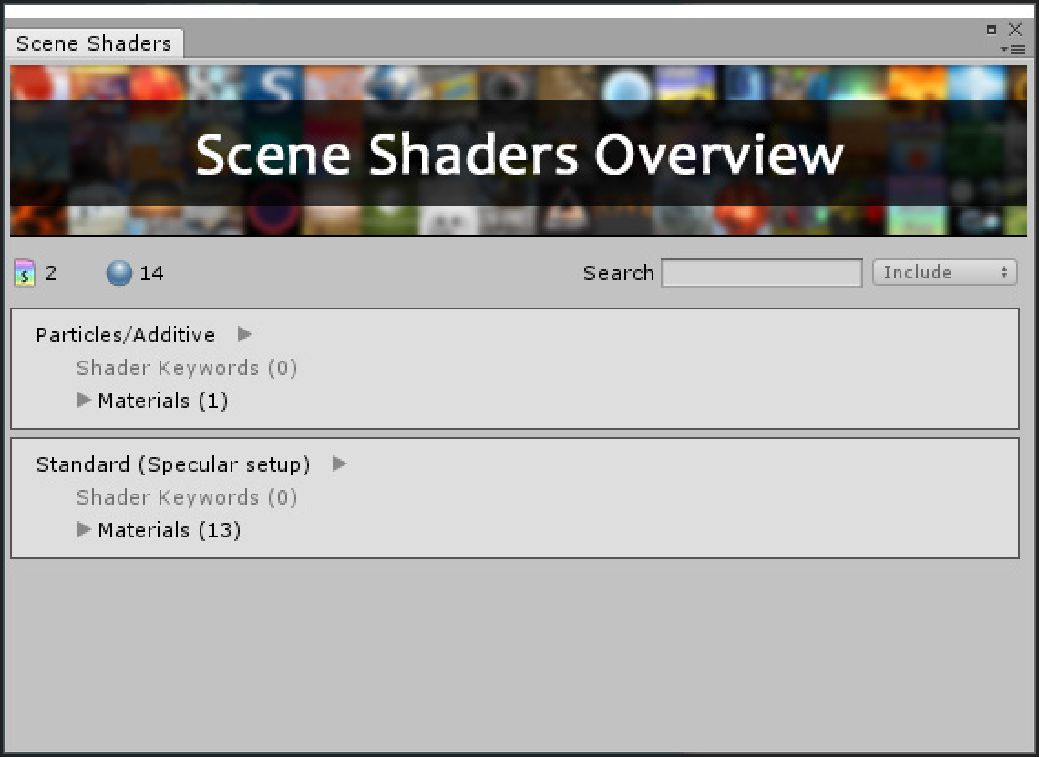
Project Bend Type: **Classic Runner** (Axis Z Positive)

Shader Packages

Example	Import	🟥 Exclude
Flare	🟩 Import	Exclude
Glass	🟩 Import	Exclude
Keijiro	🟩 Import	Exclude
Legacy Shaders (Outline)	🟩 Import	Exclude
Legacy Shaders	🟩 Import	Exclude
Nature (Unity)	🟩 Import	Exclude
One Directional Light (Outline)	🟩 Import	Exclude
One Directional Light	Import	🟥 Exclude
Particles	Import	🟥 Exclude
Projector	🟩 Import	Exclude
SpeedTree	🟩 Import	Exclude
Sprite	Import	🟥 Exclude
Standard (Outline)	🟩 Import	Exclude
Standard	Import	🟥 Exclude
Terrain (Unity)	🟩 Import	Exclude
Tessellation	🟩 Import	Exclude
Unlit (Outline)	🟩 Import	Exclude
Unlit	Import	🟥 Exclude

2) As bend effect is rendered only by using Curved World shaders, scene material shaders must be changed to the proper ones.
Currently scene meshes use Unity built-in Standard and Particle shaders. Curved World package contains exactly the same shaders but with integrated vertex transformation. We need to use them.

To make shader swap easier scene contains Materials gameobject with reference to the all used materials. Change their shaders from Unity built-in Standard to VacuumShaders -> Curved World -> Standard and Particle shaders.
Or use editor helper tool from Menu -> Windows -> VacuumShaders -> Scene Shaders Overview.



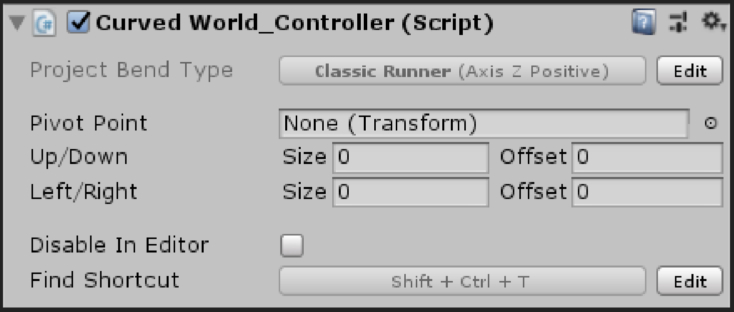
2 14

Particles/Additive
Shader Keywords (0)
Materials (1)

Standard (Specular setup)
Shader Keywords (0)
Materials (13)

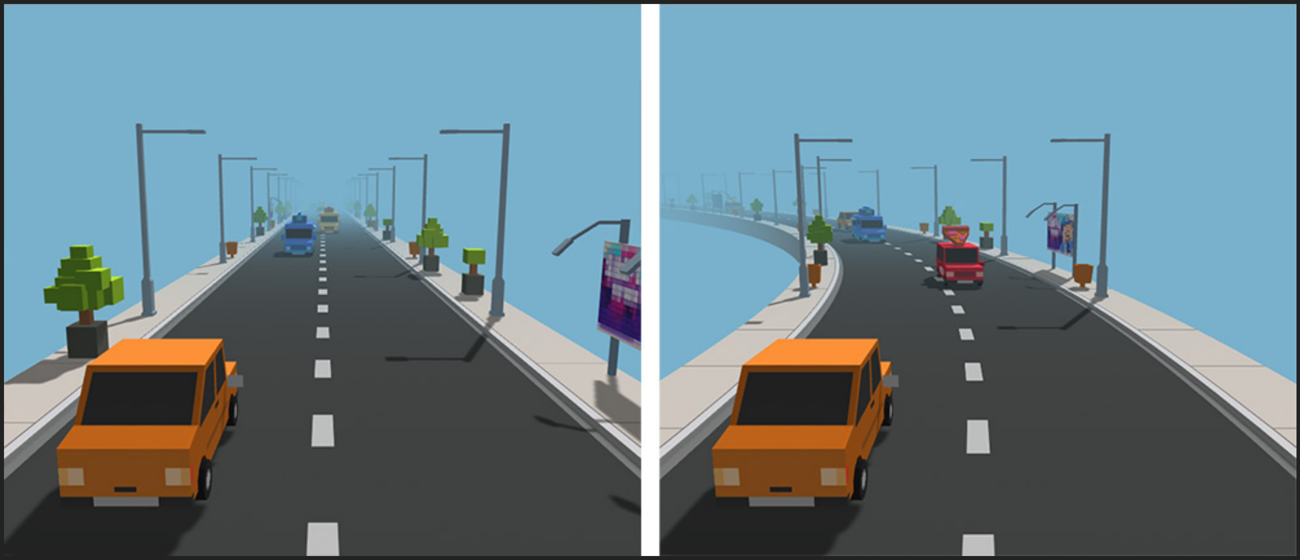
3) Shaders with integrated Curved World have no special parameters inside material editor to alter bend effect, instead everything is controlled just from one CurvedWorld_Controller script that updates all required parameters globally for all shaders.

Assign CurvedWorld_Controller script to the gameobject with the same name already in the scene.



Curved World_Controller (Script)
Project Bend Type: Classic Runner (Axis Z Positive) — Edit
Pivot Point: None (Transform)
Up/Down: Size 0, Offset 0
Left/Right: Size 0, Offset 0
Disable In Editor: ☐
Find Shortcut: Shift + Ctrl + T — Edit

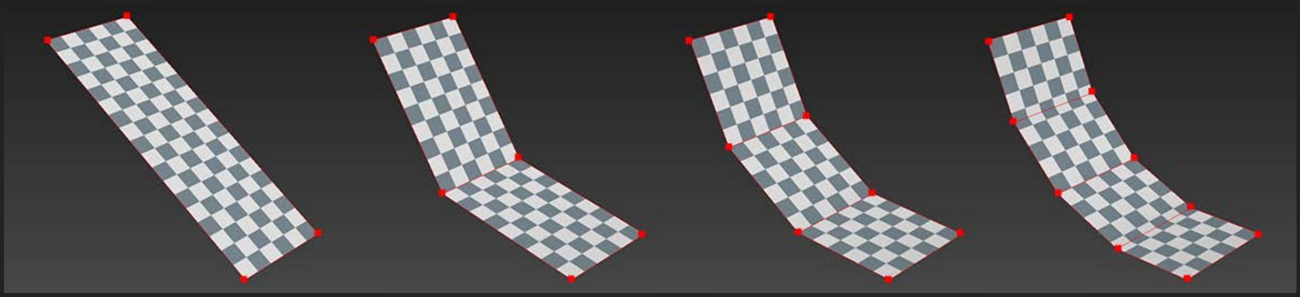
If run scene now it still will be without bending effect.
We just need to change bend parameters. Try -0.6 and 4.0 for Up/Down and Left/Righe accordingly.
Now world is curved.


Curved World Off Curved World On

To summarize how to use Curved World.
1) Set desired bend type from Settings window and recompile shaders.
2) Use shaders with Curved World vertex transformation (package comes with many different shaders, they can be imported from Settings window) or use custom ones (check CustomShaders.pdf file).
3) Update bending parameters from CurvedWorld_Controller script.

That's all.

Note: Curved World is per-vertex displace shader. Mesh vertex density defines how smooth curvature is.



Bend Types

Curved World offers 12 shader bending effects divided into two groups.

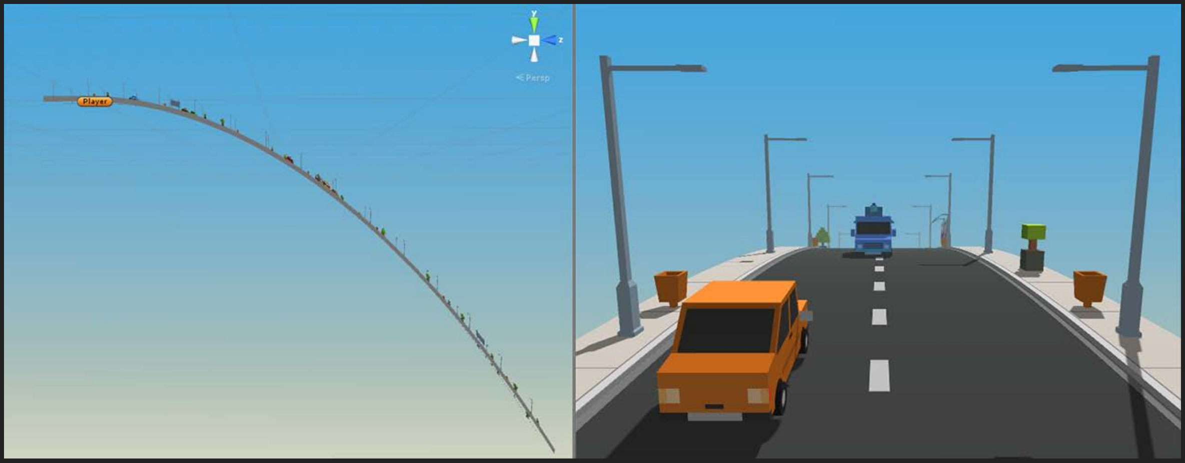
- Bend effects based on *parabola* equation:
    1. Classic Runner
    2. Little Planet
    3. Cylindrical Tower
    4. Cylindrical Rolloff
    5. Universal
    6. Perspective 2D

- Bend effects based on *spiral* equation:
    7. Spiral Horizontal
    8. Spiral Horizontal Double
    9. Spiral Horizontal Rolloff
    10. Spiral Vertical
    11. Spiral Vertical Double
    12. Spiral Vertical Rolloff

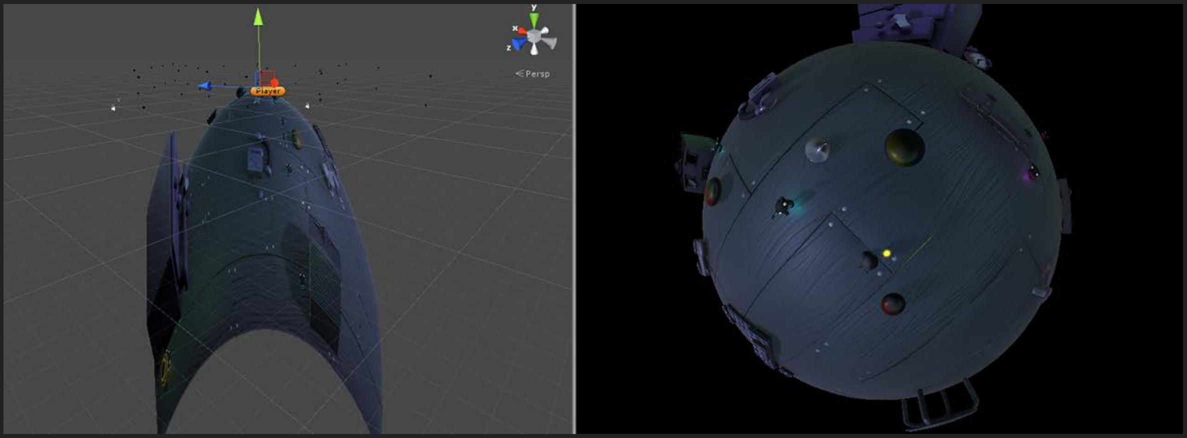
Main difference between those two groups is that effects using *parabola* equation gain more distortion/offset further they are from a pivot point and required bending effect can be observed only from a specific positions, limiting camera placement in a scene.

Here are bending examples using parabola equation.
Left image displays side view of the effect. Right image is the same scene but with correct camera position.

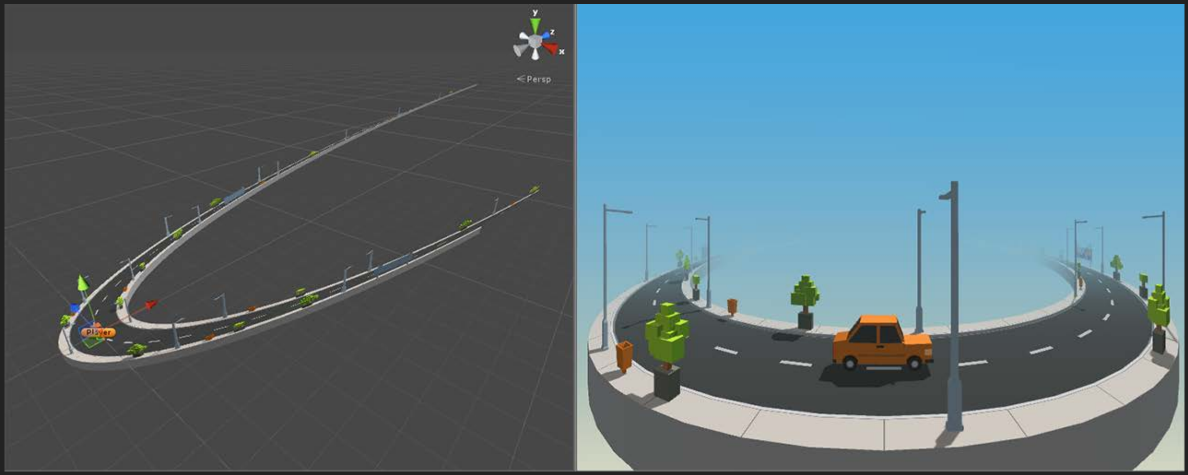
Classic Runner



Little Planet



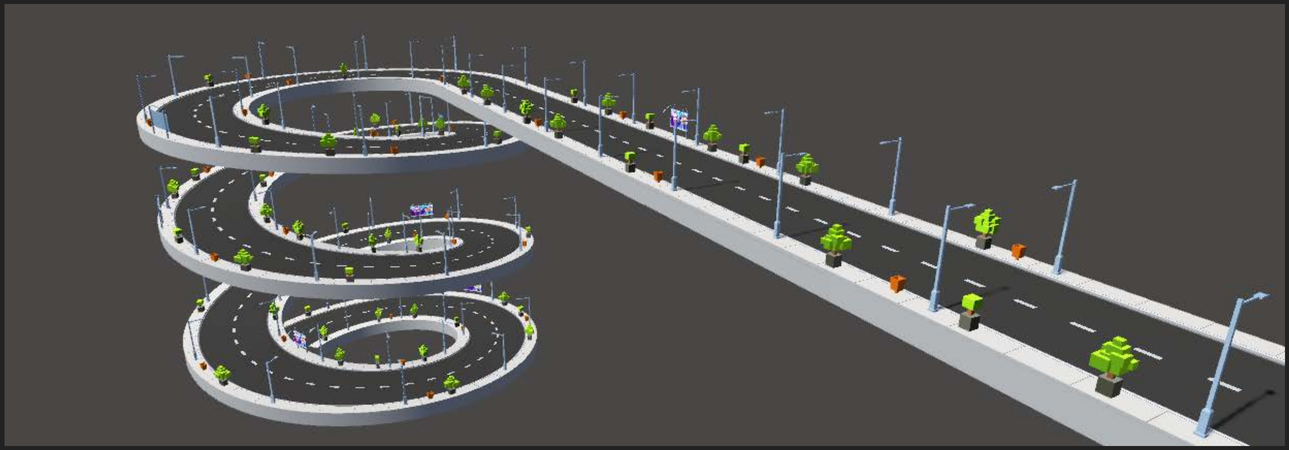
Cylindrical Tower



With those bending effects camera can be in a First Person, Third Person or Top Down position. But observing scene from a big distance or making 360° orbital rotation is not a good idea.

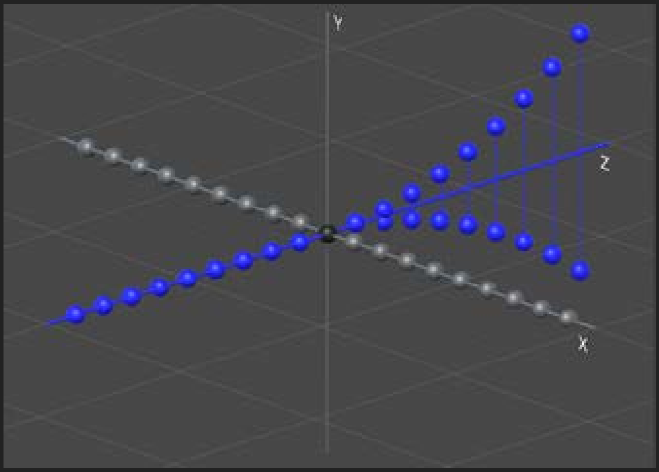
Spiral based bending effects rotate mesh vertices around custom pivot point, don’t distort/stretche and they can be observed from any point.

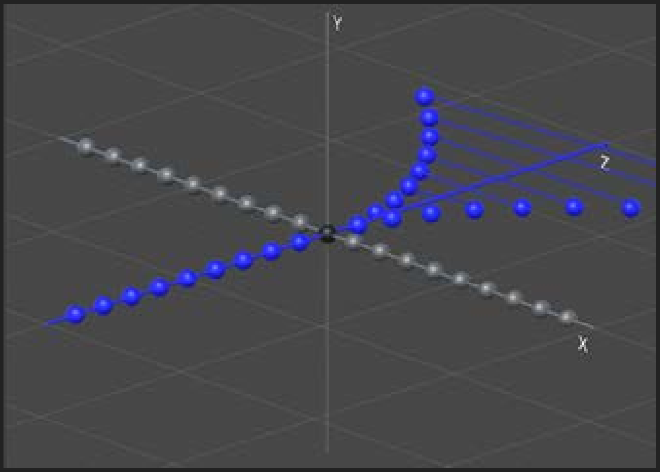
Note: Spiral rotations have constant radius.



Each bend effect transforms scene meshes in a particular way and has simple requirements toward scene design and object placement.

- **Classic Runner** - Scene objects must be located along X or Z axis.
  Axis definition inside bend effect name determines which direction scene is oriented and how it will be bent. E.g. ***Axis Z Positive*** means: objects in the positive direction of the Z axis are bent.

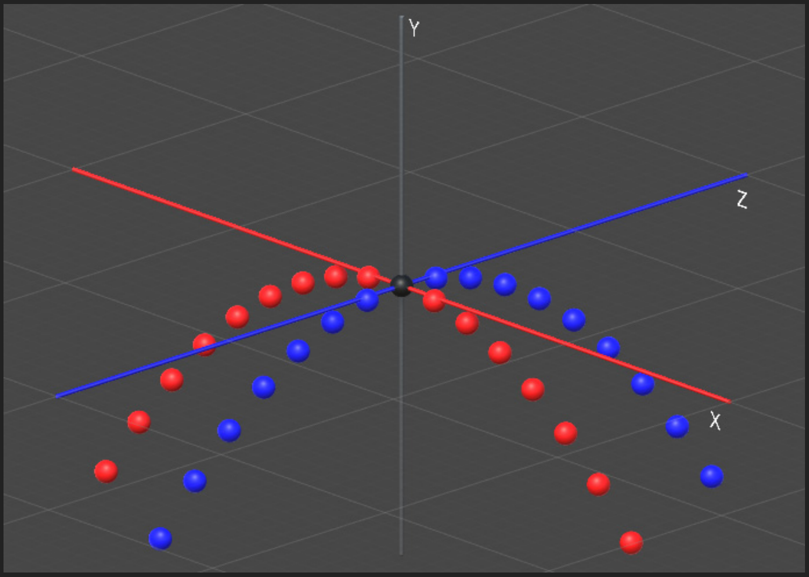




  Classic Runner support all 4 bend directions: X Positive, X Negative, Z Positive, Z Negative.
  Vertices can be moved Up/Down and Left/Right from CurvedWorld_Controller script.
  Vertices behind pivot point are not affected.

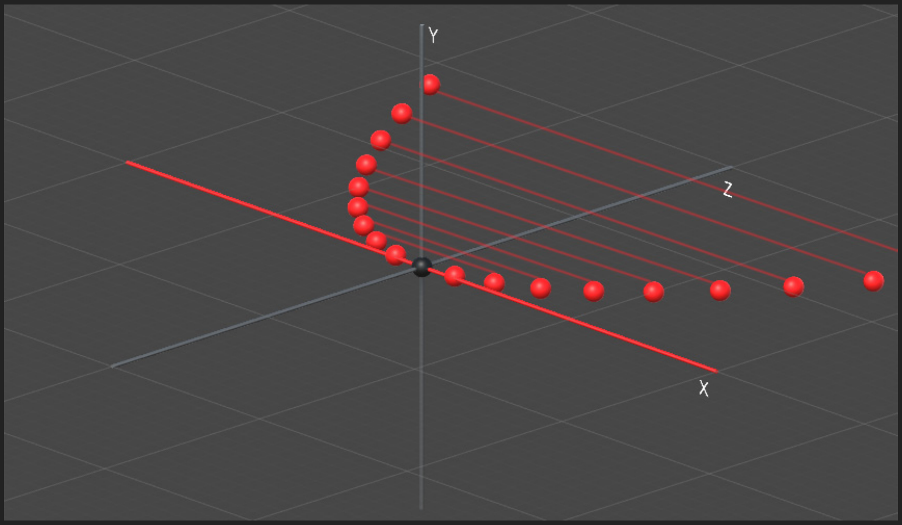
  Check 1. Classic Runner example scene.

- **Little Planet** – Scene objects are located on XZ plane (common flat/horizontal design).
  Vertices are displaced symmetrically up or down using **Curvature** parameter inside CurvedWorld_Controller script creating spherical bending effect.



  Check 2. Little Planet (Nightmare) example scene.
  It is Unity Survival Shooter Tutorial from the Asset Store, but with Curved World shaders.

- **Cylindrical Tower –** Scene objects must be located along X or Z axis. **Curvature** parameter inside CurvedWorld_Controller script controls cylindrical bending strength.

  Check 3. Cylindrical Tower example scene.

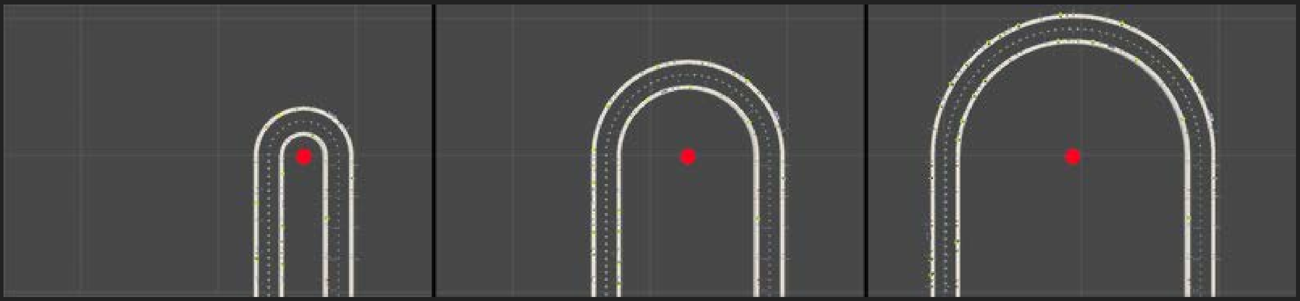
- **Cylindrical Rolloff –** Similar bending effect as **Classic Runner**, but vertices behind pivot point are also bent.

- **Universal** – Bends mesh along all three axis: X, Y and Z axis separately. It’s like a combination of Classic Runner, Little Planet and Cylindrical Tower/Rolloff effects.
-
- **Perspective 2D** - Similar to **Cylindrical Tower** type, but only for 2D sprite projects.
  Bending effect depends on camera position and rotation.
  Pivot point always is screen center point.
  Active camera type must be Perspective, not Orthographic.

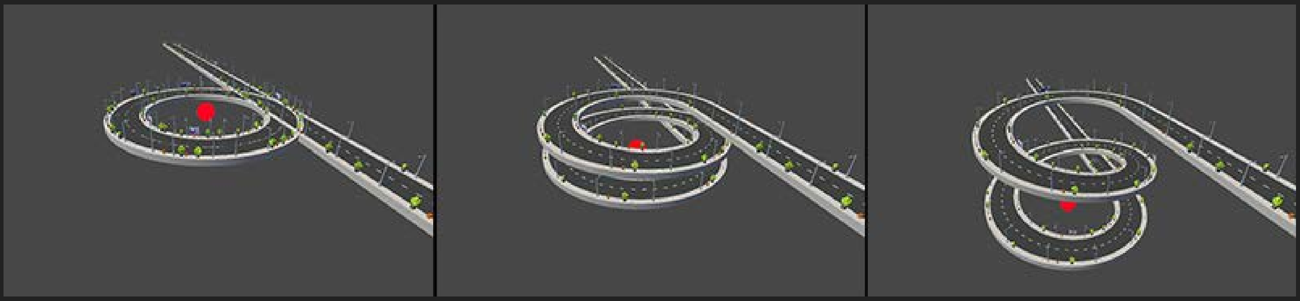
  Check 4. Perspective 2D example scene.

- **Spiral Horizontal/Vertical** – Object placement is along X or Z axis. Mesh vertices are rotated around one pivot point. Rotation angle is controlled from CurvedWorld_Controller script.
  Axis definition inside bend effect name determines which direction scene is oriented and how it will be bent. E.g. ***Axis Z Positive*** means: objects in the positive direction of the Z axis are bent.

  Pivot point's position effects rotation radius (e.g. Angle 180°, pivot point displaced to the left)



  and loops offset.



- **Spiral Horizontal/Vertical Double –** Same as above bend type but using two pivot points with individual rotation angles for each one.

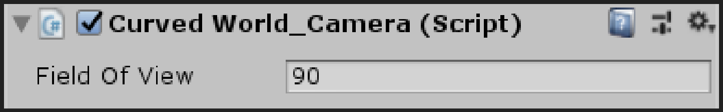
- **Spiral Horizontal/Vertical Rolloff –** Same as above bent type with two pivot points, but CurvedWorld_Controller script controls only one pair (pivot point and Angle) parameters that is symmetrically equal to the second pair.

  Check 5. Spiral Vertical, 6. Spiral Vertical Double and 7. Spiral Horizontal Double example scenes.

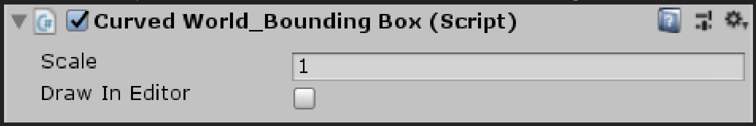
Mesh disappearing or early culling

For vertex displace shaders it is a very common problem when at some camera angle mesh 'suddenly' disappears. It happens when original mesh goes beyond camera view frustum and excluded from rendering pipeline, in this case shader has nothing to render.

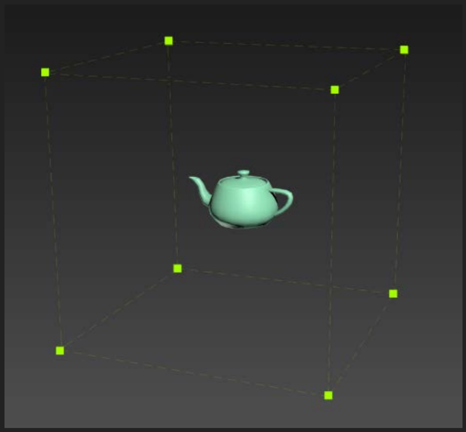
Curved World asset includes solution for that problem:

1. CurvedWorld_Camera script – Overrides camera's field of view parameter. Allowing to capture objects outside original camera view frustum.
   Works only if attached to an active camera.
   
   Curved World_Camera (Script)
   Field Of View: 90

   Keep value as low as possible.

2. CurvedWorld_BoundingBox script – Scales individual renderer's bounding box component and makes it visible to the camera even if mesh is outside its view frustum.
   Script is also necessary if a mesh is not visible to a dynamic light source (it has its own field of view), to avoid excluding from shadow receiving/casting pass.
   Works only if attached to an active non static gameobject with Renderer component.
   
   Curved World_Bounding Box (Script)
   Scale: 1
   Draw In Editor: ☐

   Keep value as low as possible.

3. Manual user solution for static objects - It is necessary to add 8 'dead' vertexes to the mesh manually using any of 3D modeling software and inside Unity in mesh import settings disable Optimize Mesh check box (Unity removes unused vertices).
   
   Scaling mesh's bounding box in 3Ds Max.
   8 vertices will be added to a mesh.
   8 vertex add only 64 bytes and don't participate in mesh rendering, as they have no triangles attached.

Note: It is very easy to find out why bent mesh has disappeared from camera - just disable CurvedWorld_Controller script and if it is not visible inside camera then one of the above solutions must be used.