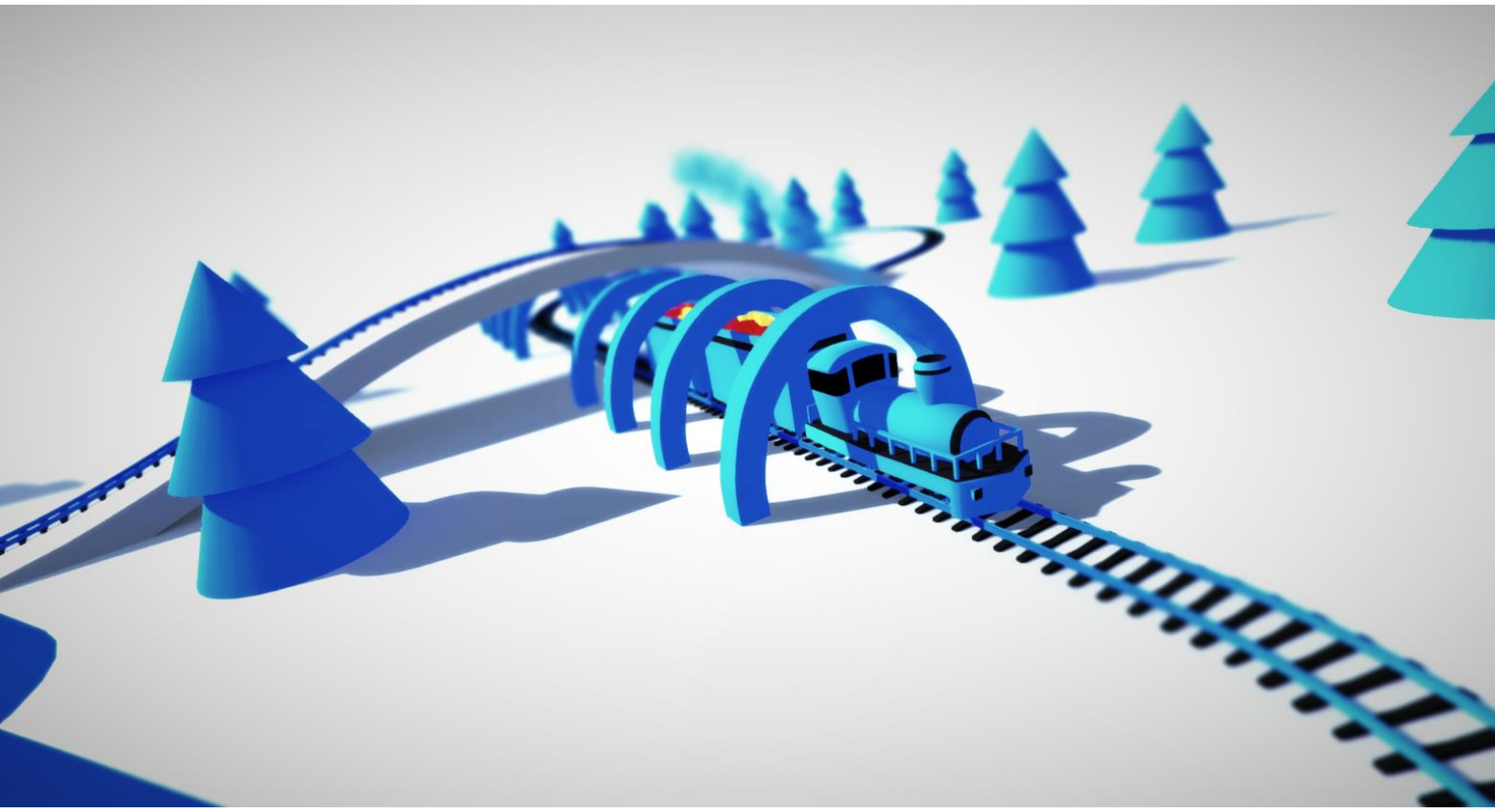

MESH DEFORMER

Ver.2 05.03.2016

Introduction

Mesh Deformer editor extension will help you easily arrange objects; duplicate bend and twist meshes along curve; edit pivot point, combine and save meshes.

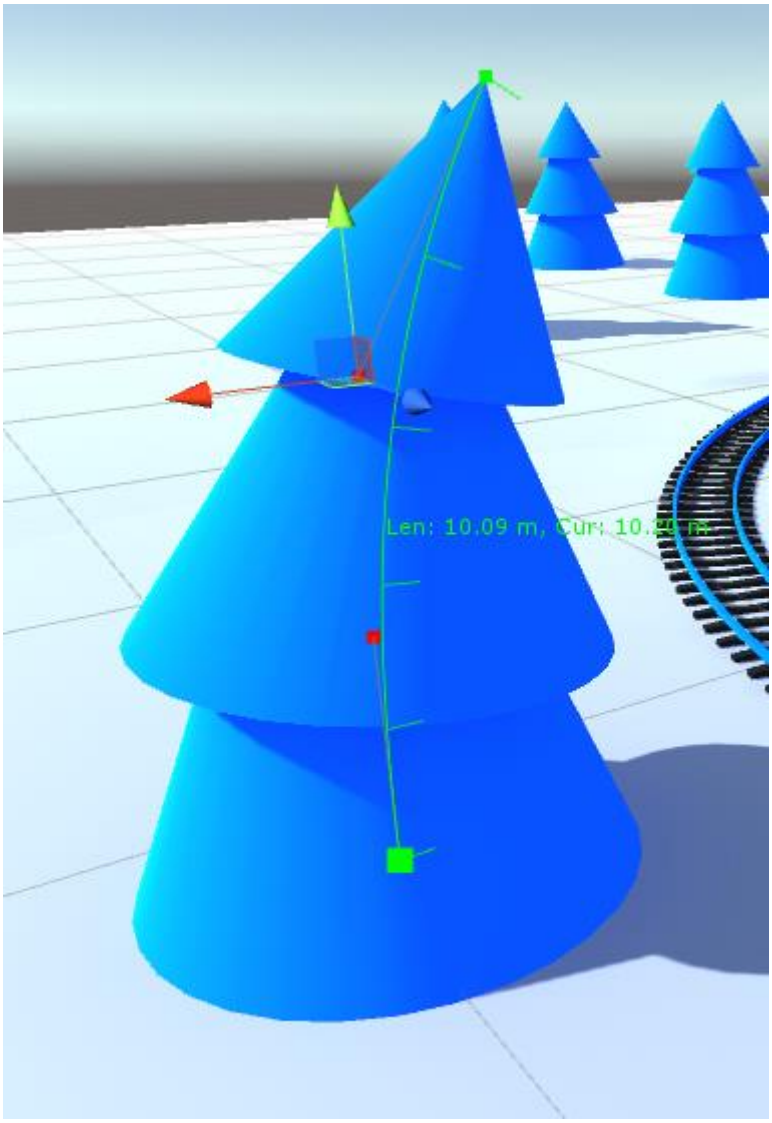

Fig.1 Mesh Deformer Curve

Features

- X, Y, Z axis bending;
- Mesh twisting;
- Mesh duplication;
- Configurable curve approximation level;
- Configurable curve count per mesh;
- Configurable spacing between meshes;
- Spline loops;
- Spline smoothing;
- Spline control point modes: Free, Aligned, Mirrored;
- Spline control points editing (support unity snap mode);
- Non-deforming (rigid mode);
- Mesh subdivision (increasing triangles count)
- Mesh combine;
- Pivot point editing
- Work with multi-material meshes;
- Full editor Undo/Redo support;
- Runtime editing (may cause performance issues);

Package Structure

MeshDeformer Package is located in **Assets/Battlehub/MeshDeformer2**

MeshDeformer was rewritten from scratch, and incompatible with previous version 1.0. **Version 1.0 is no longer supported**. If you still want to use 1.0 version it is located in MeshDeformerLegacy.unitypackage.

Script file located in **Assets/Battlehub/MeshDeformer2/Scripts**

Editor scripts and menu in **Assets/Battlehub/MeshDeformer2/Scripts/Editor**

Demo scene in **Assets/Battlehub/MeshDeformer2/Demo.unitypackage**

Saved meshes in Assets/Battlehub/SavedMeshes

Menu

There are three submenus and six menu items:

1. **Deform** submenu allow you to choose deformation axis (X, Y, Z) in local object's space.
2. **Set Mode** submenu allow you to set spline or control point mode. There are three control point modes (**Free**, **Aligned** and **Mirrored**) and special spline **Rigid** mode, which forces Free control point mode. All these modes explained in **Mesh Deformer Modes** section.
3. **Postprocessing** contains menu items which typically useful when you finish editing your mesh:
    - **Smooth Spline** using this algorithm
    - **Extract spline** duplicate raw spline and remove all meshes
    - **Remove Deformer** (deformer component removed but all deformed meshes remains as is)
    - **Rollback** return mesh to initial state, remove deformer and all mesh duplicates;
    - **Combine And Save** (combine mesh and all of its duplicates and save combined mesh to **Assets/Battlehub/SavedMeshes folder**)
4. **Subdivide mesh** (all existing vertices remains untouched, extra vertices added, each quad converted into four smaller quads). This menu item is useful when mesh has insufficient vertices to be deformed. Subdivide menu item use this method.
5. Use **Append/Prepend** to duplicate mesh and extend curve.
6. **Remove curve**.
7. **Straighten** useful to minimize deformation.
8. **Duplicate** duplicates mesh deformer and clone all meshes.

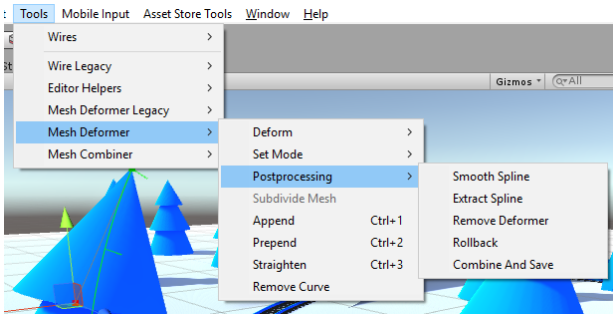

Fig.2 Menu

Mesh Deformer Modes

Mesh deformer use spline similar to described in this tutorial. Everything described in tutorial will work with Mesh Deformer. Each control point can be Free, Aligned or Mirrored. All of these modes shown in fig.3.

1) Free – move Control points independently
2) Aligned – variable distance between control points
3) Mirrored – same distance between control points

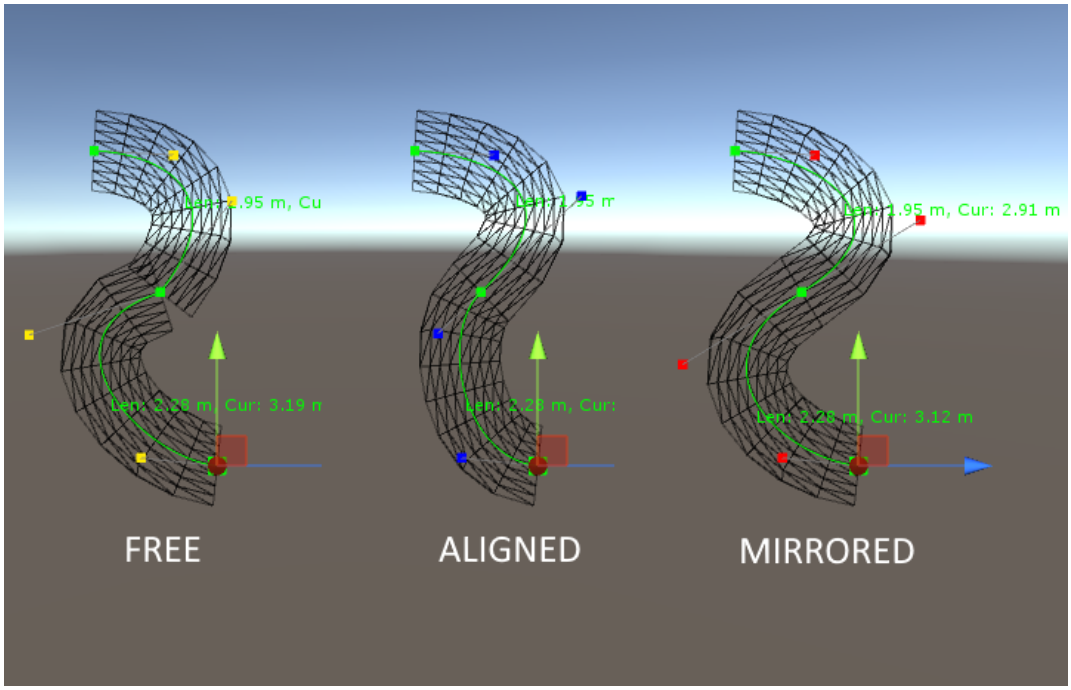

Fig.3 Control Point modes

Rigid mode forces all Control points of the curve to be aligned to a line. You can't bend and twist mesh in this mode. However mesh stretching is possible.

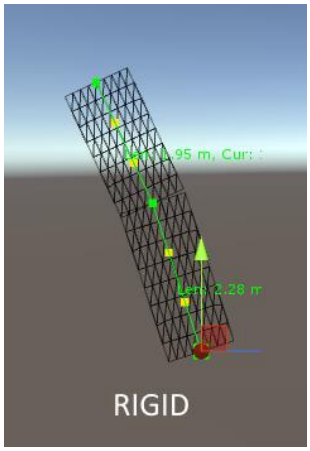

Fig.4 Rigid Mode

Mesh Deformer Editor

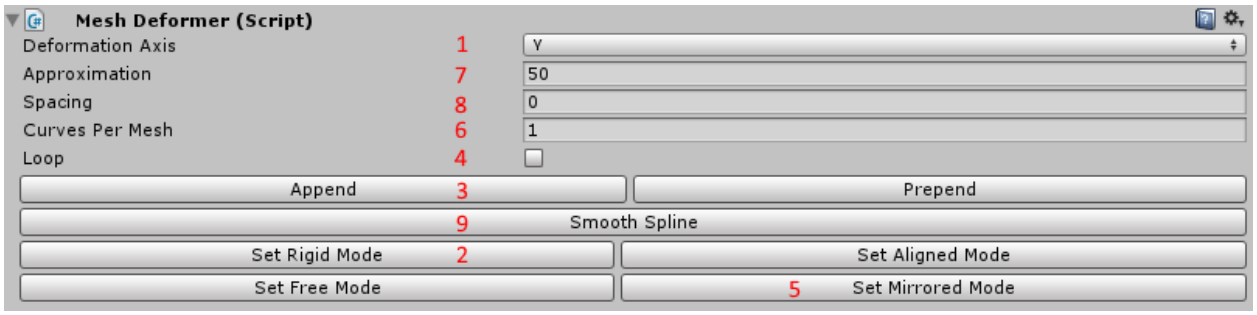

Fig.5 Mesh Deformer Editor

1. Select Deformation Axis
2. Switch all control points to rigid mode
3. Append mesh duplicates
4. Create spline loop
5. Set all control points to mirrored mode
6. Change curves per mesh count (more curves gives more control over the shape of the mesh)
7. Approximation controls how much mesh will repeat trajectory of the spline. Greater value gives better visual effect. However this value should be kept as low as possible.
8. Add extra space between meshes
9. Smooth Spline using [this algorithm](#).

Control Point Editor

This editor allow you to change individual control point mode and set twist angle and twist offset.

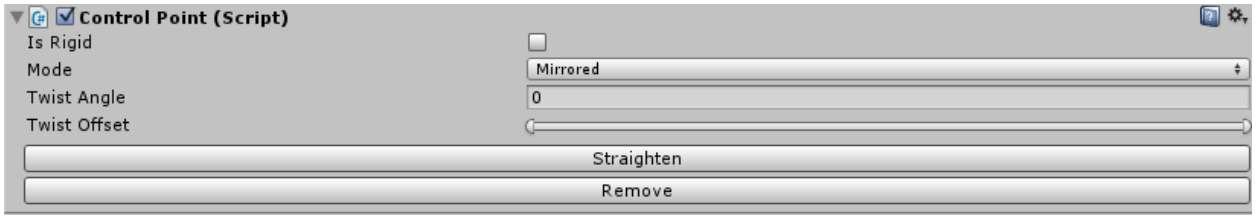

Fig.6 Control Point Editor

Mesh Deformer Script

Description

class in Battlehub.MeshDeformer2

MeshDeformer class contains main functionality of package

Properties

Mesh Deformer Settings
```
public Axis Axis
public virtual int Approximation
public virtual float Spacing
public virtual int CurvesPerMesh
public override bool Loop
```

Spline Settings
```
public int CurveCount
public int ControlPointCount
```

Mesh Duplicates
```
public ScaffoldWrapper[] Scaffolds
```

Adjacent vertex indices of mesh duplicates
```
public Contact[] Contacts
```

Non-deformed mesh
```
public Mesh Original
```

Adjacent vertex indices of duplicates of mesh collider's mesh
```
public Contact[] ColliderContacts
```

Non-deformed colliders's mesh
```
public Mesh ColliderOriginal
```

Methods

Reset deformer to initial state
```
public void ResetDeformer()
```

Rebuild deformer
```
public void WrapAndDefromAll()
```

Remove curve with curveIndex. Returns gameObject associated with curve
```
public GameObject Remove(int curveIndex)
```

Extends Deformer by adding new curve and mesh duplicate.
Returns object associated with curve.
```
public GameObject Extend(bool prepend = false)
```

Straighten mesh and curve containing control point defined by pointIndex parameter
```
public void Straighten(int pointIndex)
```

Turn On/Off Rigid mode on curve containing control point with pointIndex
```
public void SetIsRigid(int pointIndex, bool isRigid)
```

Spline measurement methods

```
public float EvalLength(int curveIndex)
public float EvalCurveLength(int curveIndex, int steps)
```

Spline data access methods (ControlPointMode: Free, Aligned, Mirrored)

```
public ControlPointMode GetControlPointMode(int index)
```

Spline data access methods (World Space)

```
public Vector3 GetPoint(float t, int curveIndex)
public Vector3 GetPoint(float t)
public Vector3 GetControlPoint(int index)
public Vector3 GetVelocity(float t, int curveIndex)
public Vector3 GetVelocity(float t)
public Vector3 GetDirection(float t, int curveIndex)
public Vector3 GetDirection(float t)
```

Spline data access methods (Object Space)

```
public Vector3 GetPointLocal(float t, int curveIndex)
public Vector3 GetPointLocal(float t)
public Vector3 GetControlPointLocal(int index)
```

Spline data access methods (Twist Angles measured in degrees)

```
public float GetTwist(float t, int curveIndex)
public float GetTwist(float t)
public Twist GetTwist(int index)
```

Spline data modification methods

```
public void SetControlPointMode(int index, ControlPointMode mode)
public void SetControlPoint(int index, Vector3 point)
public void SetControlPointLocal(int index, Vector3 point)
public void SetTwist(int index, Twist twist)
```

Spline smooth method

```
public void Smooth()
```

Mesh Deformer Demo.cs

```
namespace Battlehub.MeshDeformer2
{
    public class MeshDeformerDemo : MonoBehaviour
    {

        // Use this for initialization
        void Start()
        {
            MeshDeformer deformer = gameObject.AddComponent<MeshDeformer>();
            deformer.Axis = Axis.Y;
            deformer.ResetDeformer();
            deformer.Extend();
            deformer.Extend();
            deformer.CurvesPerMesh = 2;
            deformer.Spacing = 1;
            deformer.Approximation = 50;
            deformer.SetControlPointLocal(0, new Vector3(-1, -4, -1));
        }

    }
}
```

Limitations and Issues

- Runtime editing may cause performance issues (will be fixed in future releases)
- Large meshes and mesh deformer may significantly slow unity editor.
- Twist angles are not editable using mouse. Editing possible using editor or script.
- If you encounter some issues with Mesh Deformer try to change deformation axis in MeshDeformer editor. This will reset and rebuild mesh deformer.

Support

If you have any questions, suggestions, you want to talk or you have some issues please send mail to Vadim.Andriyanov@outlook.com or Battlehub@outlook.com.