

CurvedText Easy

A powerful Unity component for creating curved and circular text effects using TextMeshPro. CurvedTextEasy enables text to wrap naturally along curves or arcs inside VR, with smooth auto-spacing and real-time rendering.

Features

Two Curve Modes:

- **Circle Mode:** Arrange text along a circular arc
- **Curve Mode:** Use AnimationCurve to create custom text curves

Dual Axis Support:

- **Y-Axis:** Traditional vertical curve (up/down bending)
- **Z-Axis:** Circular wrap effect (front/back wrapping)

Rotation Control: Adjustable rotation strength to control text rotation along the curve

Real-time Preview: Works in both Editor and Play modes

Smart Initialization: Handles TextMeshPro mesh initialization automatically

Performance Optimized: Efficient mesh updates

Requirements

- Unity 2020.3 or later
- TextMeshPro (included in Unity 2018.2+)

Installation

1. Import the CurvedTextEasy package into your Unity project
2. Add the CurvedTextEasy component to any GameObject with a TextMeshPro component (TextMeshProUGUI or TextMeshPro)

Usage

Basic Setup

1. Create a TextMeshPro text object in your scene
2. Add the CurvedTextEasy component to the same GameObject
3. The text will automatically curve based on the default settings

Circle Mode

In Circle Mode, you can:

- Set the **Radius** of the circular arc
 - Default: 100 (2D UI) or 2 (3D text)
- Define **Start Angle** and **End Angle** (in degrees)
 - Default: Start Angle = 0°, End Angle = 180°
 - 0° = left
 - 90° = top
 - 180° = right
 - 270° = bottom
- Control **Max Letter Spacing** to limit angular distance between letters
 - Default: 360° per letter
- Start Angle < End Angle = Counterclockwise arrangement
- Start Angle > End Angle = Clockwise arrangement

Curve Mode

In Curve Mode, you can:

- Adjust the **Curve** animation curve to define the text path
- Use **Curve Scaling** to control the intensity of the curve
- Choose from preset curves via buttons in the Inspector: Arc Up, Arc Down, S Shape, Wave, Left High, Right High

General Parameters

- **Curve Axis:** Choose Y-axis (vertical) or Z-axis (circular wrap)
- **Rotation Strength:**
 - 0 = Text stays horizontal
 - 1 = Text fully follows curve rotation

Scripting API

Namespace: CurvedTextEasy

Public Methods

```
// Manually refresh the curve effect
curvedTextEasy.RefreshCurve();
```

Public Properties

```
// Curve mode (Circle or Curve)
CurvedTextEasy.CurveMode curveMode = curvedTextEasy.curveMode;

// Curve axis (Y or Z)
CurvedTextEasy.CurveAxis curveAxis = curvedTextEasy.curveAxis;

// Rotation strength (0-1)
```

```
float rotationStrength = curvedTextEasy.rotationStrength;

// Animation curve for text bending (Curve Mode only)
AnimationCurve vertexCurve = curvedTextEasy.vertexCurve;

// Curve scaling (Curve Mode only)
float curveScaling = curvedTextEasy.curveScaling;

// Radius (Circle Mode only)
float radius = curvedTextEasy.radius;

// Start angle in degrees (Circle Mode only)
float fromArcDegrees = curvedTextEasy.fromArcDegrees;

// End angle in degrees (Circle Mode only)
float toArcDegrees = curvedTextEasy.toArcDegrees;

// Max degrees per letter (Circle Mode only)
int maxDegreesPerLetter = curvedTextEasy.maxDegreesPerLetter;

// Debug info display (for troubleshooting)
bool showDebugInfo = curvedTextEasy.showDebugInfo;
```

Tips

- For 2D UI (TextMeshProUGUI), default scaling values are larger (100)
- For 3D text (TextMeshPro), default scaling values are smaller (2)
- The component automatically detects the text type and adjusts defaults
- Works with Layout Groups and Content Size Fitter
- Text updates automatically when content changes
- Enable **Show Debug Info** in the Inspector to see debug messages

Support

For issues, questions, or feature requests, please visit our official website.