



INTRODUCTION

Turn editable words into dimensional title cards, signs, logos, physical lettering, and world-space interfaces that feel built for the scene.

Begin with one of 80 visual styles and seven ready-to-use font families, keep the message in the familiar Text Studio workflow through TextMeshPro or UniText, then refine faces, depth, materials, contours, interaction, and performance at your own pace.



At a glance

Path	What you can accomplish
Start with a style	Choose a real rendered preview, one of 80 styles and seven ready-to-use font families, create a title, and keep the complete design editable.
Shape the letters	Build clean extrusion, soft or architectural bevels, carved and inflated faces, layered depth, outlines, and backings.
Design the surface	Assign materials by visual purpose and keep fronts, sides, bevels, trim, and decorations coordinated.
Arrange the result	Use spatial forms, separate depth sweeps, contours, tubes, repeats, sockets, and world-space layouts.
Bring it to life	Add compatible Text Studio motion, eight native 3D effects, hover and click response, or physical interaction.
Ship confidently	Choose an output and performance profile that matches changing text, large displays, per-letter interaction, or physics, then freeze a finished title when it no longer needs runtime behavior.

CONTENTS

Introduction	1
What Text Studio 3D Can Make	8
Create Your First 3D Title	9
Prerequisites and installation order	9
Add 3D to an existing Text Studio object	9
Step 1: Open the 3D Gallery	9
Step 2: Enter the text in Text Studio	9
Step 3: Follow the persistent workflow	10
Step 4: Preview before entering Play Mode	10
Understand Text, Geometry, and Style	12
Text Studio owns the message	12

UniText World text.....	12
A style owns the look	12
Placement and depth sweep are independent.....	12
The geometry pipeline	13
Shared style or detached local layers	13
Stack, Overlay, and depth ownership	13
Layer selection and scope.....	13
Build a Style Layer by Layer	14
The nine layer specifications.....	16
Layer order walkthrough.....	16
Shape Faces, Edges, and Depth	16
Front and back faces.....	16
Bevel profiles.....	16
Bodies and depth sweeps	17
Deformer stacks.....	17
Caps and relief.....	18
Body extrusion and shells	18
Bevels.....	18
Bevel Profile assets	18
Decoration profile	18
Depth shaping and deformation.....	19
True outlines and stroke inheritance.....	19
Create Logos, Backings, and Custom Contours	19
Choose a contour source	19
Import an SVG.....	19
Build a contour program.....	19
Compose a backing.....	20
Generated contours	21
SVG import	21
Contour Programs	21

Backing tutorial.....	21
Add Tubes, Repeats, and Attachments	22
Contour tubes.....	22
Contour repeats.....	22
Named glyph sockets	22
Contour Tubes.....	23
Contour Repeats.....	23
Marquee tutorial.....	23
Glyph Sockets	23
Design Materials and Surfaces	24
Start with a Material Set	24
Choose texture mapping.....	24
Material regions and fallback	25
Material Sets	25
Surface mapping	25
Render pipeline material families	25
Material recipes	25
THEME COMPLETE 3D EXPERIENCES	26
PROJECT AND HIERARCHY THEMES	26
NAMED STYLES AND CONTROL THEMES	26
BUILD A THEME-SWITCHABLE SPATIAL MENU.....	27
Animate 3D Text and Add Interaction	27
Compatible Text Studio effects.....	27
Composed Effect timing.....	28
No-code interaction.....	30
Base effects and output compatibility	31
Native 3D effects.....	31
Interaction Controller	31
Forces.....	31
Physical Letters.....	32

Runtime overrides.....	32
Arrange Text in Space.....	32
Placement versus extrusion sweep.....	34
Quadratic Bezier extrusion path.....	34
Unity Splines extrusion.....	34
Spatial placement reference	34
Deformer stacks.....	34
Build World-Space Interfaces.....	35
CANVAS AND CAMERA CHOICES	35
Control foundation and states.....	36
BUTTONS, INPUTS, PROGRESS, LISTS, AND DROPDOWNS	36
Input Field	36
MENU FLOW AND OPTION PRESENTATION.....	36
Pointer, navigation, layout, and pools.....	37
CONTROL THEMES AND BOUNDED CONTROL POOLING	37
World-space UI production tutorial.....	37
Choose Output and Performance.....	37
Choose an output mode	37
Start from a performance profile	38
MESH DETAIL PRESETS	38
Changing text and runtime overrides	38
Freeze a finished title.....	39
Choose output by workload	40
Performance profiles.....	40
Quality, LOD, and topology effects.....	40
Runtime Statistics.....	40
Prewarming and bounded caches	40
Schedule versus rebuild.....	41
Strict Freeze.....	41
Prepare Fonts and Languages	41

Bundled fonts and Type Foundry.....	42
Font Baker	43
Font ownership and prepared data.....	44
Prepared Font inspector	44
Font Baker workflow	44
Language validation.....	44
Build Advanced Reusable Modules	45
Module mental model.....	46
Layout Transform and Deformer modules	46
Glyph Prefab Spawner.....	46
Module Inputs and custom modules	46
Learn From the Shipped Samples.....	46
Optional integration installation	48
Input System	48
XR Interaction Toolkit.....	48
UniText	48
Unity Splines	49
Troubleshooting and Reference.....	49
Nothing appears.....	49
Geometry looks wrong	49
Animation is expensive	49
Asset and window reference.....	49
Support	49
Essential scripting API.....	50
Prewarm and pool	50
Runtime overrides and topology	50
Glyph handles, ranges, raycasts, and materials	50
Interactions, forces, and sockets	51
Authoring assets and extension points	52
Inspector and asset reference	52



APPENDIX: 80-STYLE GALLERY	58
Basic Styles	58
Stacked Styles.....	63
Profile Styles.....	73
Chisel Styles.....	79
Inflate Styles.....	82
Depth Styles	85
Material Styles	90
Motion Styles.....	94
Outline Styles.....	97
Decoration Styles	99
APPENDIX: NATIVE 3D EFFECT GALLERY	101

WHAT TEXT STUDIO 3D CAN MAKE

Text Studio 3D turns the same editable, animated Text Studio content into dimensional titles, signs, logos, physical letters, interactive labels, and world-space controls. Start from a finished style, then open only the part of the design you want to change.



Title Card Hero combines depth, layered materials, and a readable front face.

- Choose from 80 curated styles across ten visual families and seven bundled font families.
- Build layered faces, bevels, bodies, backs, contours, decorations, and materials.
- Animate with all compatible Text Studio effects plus eight native 3D effects.
- Arrange letters through spatial forms or extrude them along a separate depth sweep.
- Use combined, chunked, per-letter, hybrid, LOD, or physical output, or freeze the exact final appearance as static geometry.
- Create accessible world-space panels and controls.

Theme a complete spatial menu, world-space HUD, title family, or gameplay-feedback system through one project or hierarchy choice.

Ship polished Progress Bar, Dropdown, and Starter Menu prefabs alongside buttons, toggles, sliders, lists, input fields, and selectors.

Emit Damage, Critical, Heal, Status, Combo, and Score feedback through the same Text Studio emitter used by flat text.

CREATE YOUR FIRST 3D TITLE

PREREQUISITES AND INSTALLATION ORDER

Text Studio 3D requires Unity 2022.3 or newer, Text Studio 1.0.1 or newer, TextMeshPro 3.0.9 or newer, and Mathematics 1.3.2 or newer. Install Text Studio first, verify a normal animated TMP object, then install Text Studio 3D and allow Unity to compile before importing optional integrations.

- Confirm Tools > Text Studio > 3D contains the Gallery, browsers, Live Preview, Font Baker, and setup commands.
- Create one default Text Studio 3D object and confirm that readable geometry appears without assigning a custom style or font asset.
- Use the active Text Studio backend's font. Text Studio 3D does not introduce a second customer-facing font choice.
- Check the current render pipeline materials before changing geometry settings.

First-install checkpoint: a default object should show readable extruded text, one enabled layer, valid materials, and a working perspective preview.

ADD 3D TO AN EXISTING TEXT STUDIO OBJECT

Select a compatible world-space text object and use its setup card to add Text Studio 3D. The component follows Text Studio content, typography, reveal, effects, and placement, then generates dimensional geometry. When the optional UniText integration is active, shaped world-space text can use the same path without requiring a TextMeshPro component.

STEP 1: OPEN THE 3D GALLERY

Choose Tools > Text Studio > 3D > Gallery. Select a style, inspect its real rendered preview, then create a new title or apply the style to the selected Text Studio object.

STEP 2: ENTER THE TEXT IN TEXT STUDIO

The base Text Studio component remains the source of truth for content, typewriter timing, placement, and effects. The 3D inspector controls only geometry, materials, output, and 3D-specific behavior. A world-space TextMeshPro setup card can add Text Studio and Text Studio 3D in one step while preserving existing text. When UniText World and the optional integration are installed, Text Studio selects that backend automatically and drives the same extruded workflow without requiring a TextMeshPro component.

STEP 3: FOLLOW THE PERSISTENT WORKFLOW

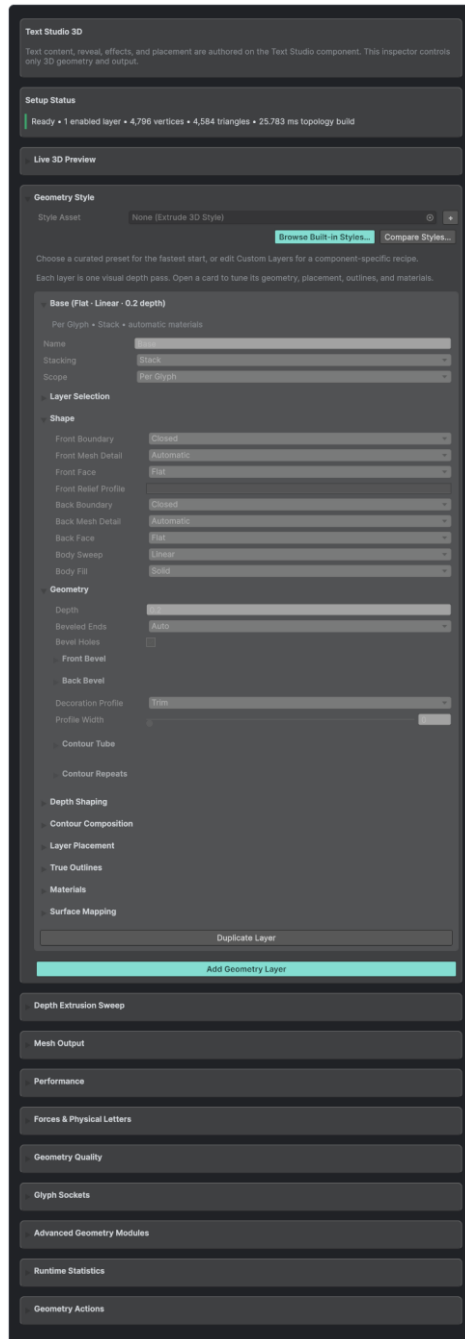
The inspector keeps setup, style, geometry, output, quality, interaction, and advanced tools in persistent foldouts. Open only the section needed for the current decision; every setting remains reachable without switching inspector modes.

STEP 4: PREVIEW BEFORE ENTERING PLAY MODE

Open Live Preview. Use front, perspective, side, and back views, orbit and pan, studio lighting, wireframe, material regions, normals, sockets, contour direction, and bounds to understand the result.

Success checkpoint: You should have readable 3D text, a chosen style, a perspective preview, and no setup warnings.

TEXT STUDIO - 3D - USER GUIDE - 1.0.1



Persistent foldouts keep setup, a starting style, live output, and the complete authoring workflow easy to find.

UNDERSTAND TEXT, GEOMETRY, AND STYLE

TEXT STUDIO OWNS THE MESSAGE

Edit content, font, typography, reveal, effects, and placement through Text Studio and the active text backend. Text Studio selects a compatible backend automatically; Text Studio 3D follows its content and font, then produces dimensional geometry. There is no separate 3D font choice on the component.

Scramble and other live character substitutions replace the visible extruded letter shapes while preserving the original layout. TextMeshPro and UniText typography remain live: bold, italic, case, Small Caps, tracking, underline, strikethrough, solid colors, and authored gradients continue through the 3D result. Underline and strikethrough become real extruded geometry without turning spaces into visible characters.

UNITEXT WORLD TEXT

With the optional UniText integrations installed, shaped glyph IDs, ligatures, bidirectional visual order, fallback fonts, and text decorations become animated extrusion when standard TTF or OTF outline data is available. Font-size and automatic layout changes rebuild immediately at the new scale. The separate UniText 3D sample integration includes inspectable Latin, Arabic, and Hebrew lettering with package-owned materials.

A STYLE OWNS THE LOOK

A style contains one or more ordered geometry layers. Start from a curated style, compare alternatives with the same text, then apply, duplicate, detach an editable copy, or reset.

PLACEMENT AND DEPTH SWEEP ARE INDEPENDENT

Spatial placement arranges letters through space. A depth sweep bends the extrusion from front to back. A deformer reshapes finished geometry. Keeping these separate makes complex results easier to reason about.

THE GEOMETRY PIPELINE

A finished result passes through a predictable pipeline: text and shaped glyph outlines, layer selection, contour construction, front and back caps, relief, bevel, body or shell, depth shaping and sweep, contour programs and decorations, deformation, surface mapping and materials, mesh assembly, and output.

Text, font, and semantic glyph identity come from Text Studio and the active backend. A 3D Style describes geometry and surfaces. The compiled style validates that recipe. The renderer builds topology, applies materials, and chooses an output. Runtime visual channels, supported deformers, tint, visibility, and forces can update warmed geometry without repeating every earlier stage.

SHARED STYLE OR DETACHED LOCAL LAYERS

A shared 3D Style is the right source when several objects should change together. Apply a curated style to keep the asset reference. Duplicate the asset before changing a project-wide design. Detach when this one object needs local editable layers. Saving detached layers as a new style turns the local result back into a reusable asset.

- Apply changes the current style reference to the selected asset.
- Duplicate creates an independent reusable style asset.
- Detach copies the effective layers into the component for local editing.
- Reset returns to the product default and removes accidental local complexity.

Editing a shared style affects every object that references it. Editing detached layers affects only that component. Use meaningful style names, keep experimental copies out of production prefabs, and validate at the smallest output size.

STACK, OVERLAY, AND DEPTH OWNERSHIP

Stack places a layer behind the preceding emitted depth. Overlay starts a new depth chain at the front plane. Use Stack for physically layered signs, trims, plates, and strokes that should build backward. Use Overlay for a new foreground treatment that must not inherit the previous depth.

The resolver coordinates bevel ends across a stack so adjacent layers do not all create unnecessary coincident end geometry. Layer Placement adds XY offset and extra depth offset after the layer's own geometry. Use the exploded preview and material-region diagnostic to find interior seams.

LAYER SELECTION AND SCOPE

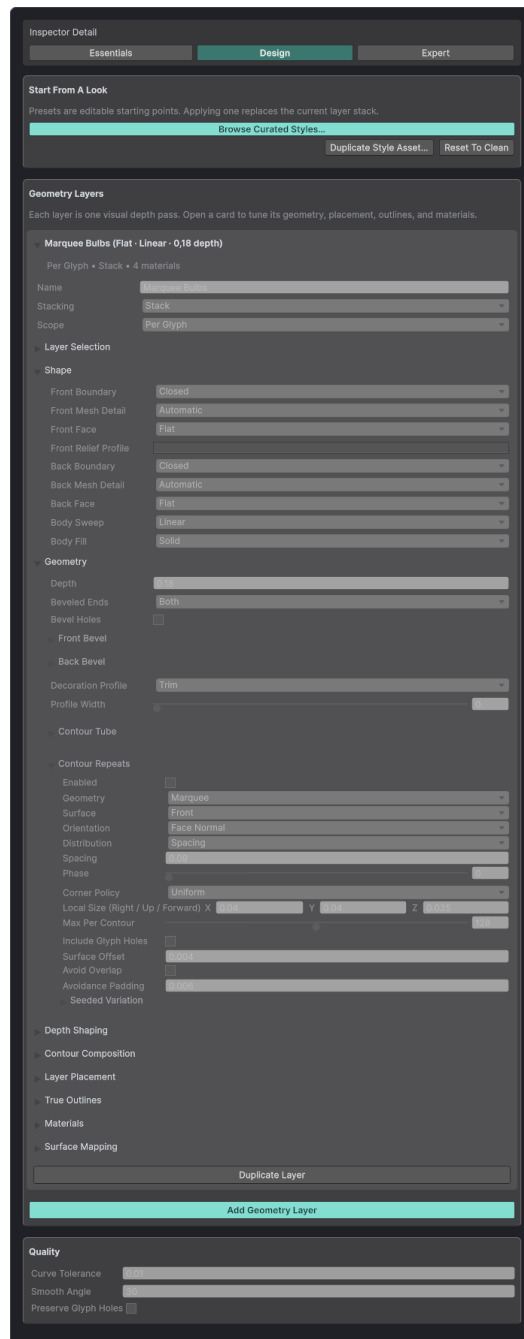
A layer can address all visible glyphs or a selected character or range, with optional inversion. Per Glyph emits independent glyph entries. Word, Line, and Whole Text collect the selected contours

into grouped geometry. Grouped contours enable fitted backings and connected outlines but depend on the complete text and cannot use independent-glyph prewarming.

BUILD A STYLE LAYER BY LAYER

Each layer is one readable design decision. Duplicate a useful layer before experimenting, use Solo to inspect it temporarily, and keep the permanent Enabled state for the result you want to ship.

- Contour: input glyphs, text bounds, custom shapes, outline strokes, and boolean programs.
- Front: flat, chiseled, or inflated relief with independent refinement.
- Edge: front bevel, back bevel, counter bevel, and bevel side selection.
- Body: depth, sweep path, solid or hollow shell, and inner wall treatment.
- Back: independent cap, relief, and refinement.
- Deformation: taper, twist, bulge, and ordered deformer stacks.
- Decoration: trim, grooves, beads, lips, contour tubes, and repeats.
- Surface: semantic material regions and texture mapping.
- Placement: layer stacking, XY offset, and Z offset.



A 3D Style exposes layers progressively and keeps common actions close to the current layer.

THE NINE LAYER SPECIFICATIONS

Each TextStudio3DLayer exposes nine orthogonal specifications. Contour chooses the input shape and scope. Front and Back own their caps and relief. Edge owns bevels. Body owns extrusion and shells. Deformation owns taper, twist, and bulge. Decoration owns trim, tubes, and repeats. Surface owns materials and mapping. Placement owns Stack or Overlay and offsets.

- Enabled removes the layer from compilation and rendering while preserving every authored value.
- Solo is preview-only and does not become a runtime mode.
- Duplicate before experimenting with topology-heavy changes.
- Reorder only after checking depth chains, stroke inheritance, and material fallbacks.

LAYER ORDER WALKTHROUGH

1. Start with one readable foreground layer and verify the front face, bevel, body, and material.
2. Duplicate it for an outline or backing, change Contour expansion or scope, then place it behind with Stack.
3. Use Overlay when a new layer should begin at the front plane rather than continue the depth chain.
4. Solo each layer and inspect wireframe, normals, material regions, contour direction, and bounds.
5. Re-enable the complete stack and check seams from the front, side, back, and a moving perspective view.

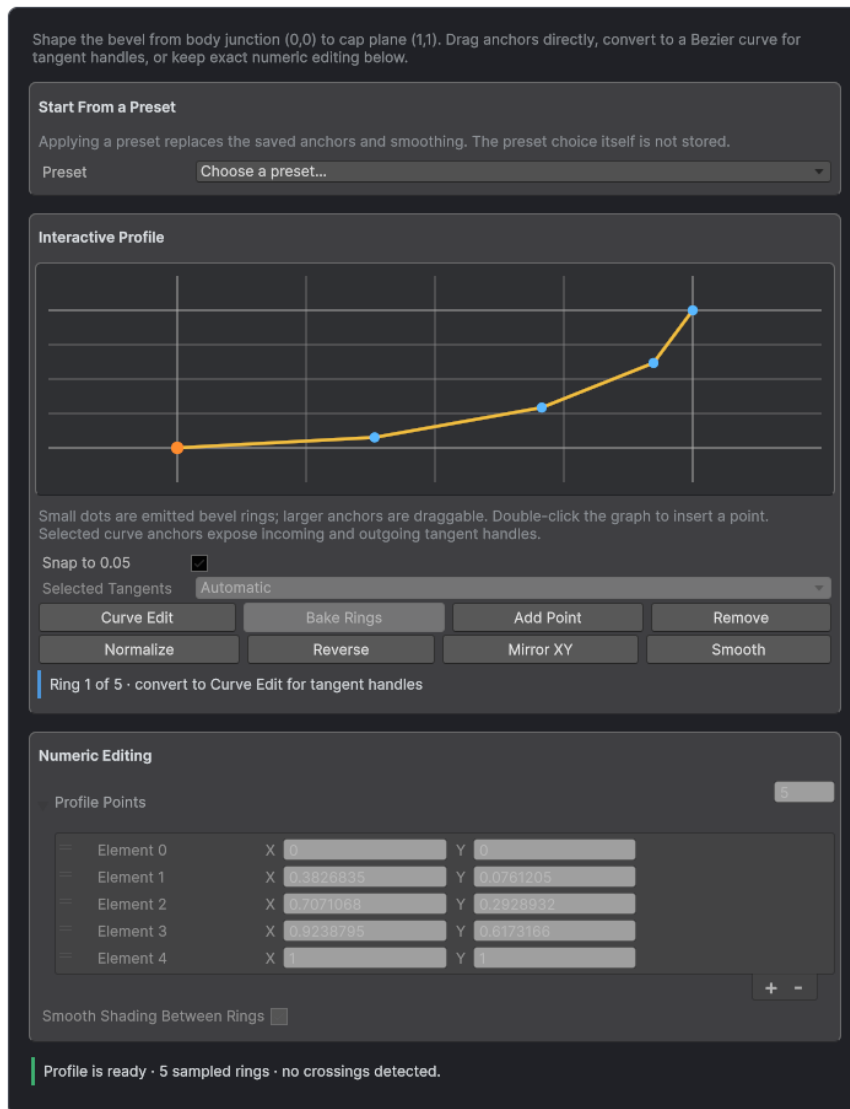
SHAPE FACES, EDGES, AND DEPTH

FRONT AND BACK FACES

Front and back caps can be open or closed and use Automatic or Regular refinement. Flat keeps the source face. Chisel creates a carved or faceted surface. Inflate produces smooth rounded crowns without chiseled center ridges, and front and back can use different choices. Foil Balloon pairs matching front and back inflation around a narrow crimped seam for a balanced profile from every angle.

BEVEL PROFILES

Choose from 22 starting profiles, then edit points, Bezier tangents, snapping, smoothing, mirroring, reversal, and baked rings. Use scene handles when the profile needs to be judged on the finished word.



The Bevel Profile graph gives direct control over the edge silhouette and generated rings.

BODIES AND DEPTH SWEEPS

Use a straight body for ordinary extrusion. Use a quadratic or spline sweep when depth should bend, bank, roll, or scale from front to back. Closed sweeps and path-space mapping support complete dimensional forms. Solid and Hollow body modes control whether an inner shell is generated.

DEFORMER STACKS

Bend, Arc, Shear, Pinch, Flare, Spherize, Cylindrical Wrap, Wave, Noise, and Twist can be ordered and scoped in Glyph, Word, Line, Text, Object, or World space. Bounds and falloff keep the result localized.

CAPS AND RELIEF

Front and Back independently choose whether the cap is closed and whether it is Flat, Chiseled, or Inflated. Open caps retain body and optional bevel rims but emit no face region. Relief and refinement settings remain stored and become active again when the cap is closed.

Flat preserves the source face. Chisel creates a carved or faceted center treatment. Inflate creates a smooth crown. Regular refinement adds bounded interior topology for deformation. Use the smallest subdivision count that holds the silhouette at the real viewing distance.

BODY EXTRUSION AND SHELLS

Body Depth controls the front-to-back distance. Linear sweep is the normal extrusion. Path sweep samples an Extrusion Path for curved depth. Minimum Rings controls how many cross-sections are available for bending, banking, roll, scale, and deformation.

Solid produces the ordinary filled body. Hollow creates an inner shell and Inner Walls material region. Shell thickness and joins must leave enough space around narrow counters. If holes collapse, reduce thickness, bevel, contour expansion, or miter pressure before increasing mesh detail.

BEVELS

Edge controls the front and back bevels, counter behavior, profile, width, segments, joins, and smoothing. Automatic resolves appropriate ends across a layer stack. Both, Front Only, Back Only, and None are explicit choices for a single layer. Counter bevels remain part of the Inner Walls region.

- Miter keeps a sharp join but can extend strongly at acute corners.
- Round adds curvature and usually needs more detail.
- Bevel cuts the corner and is stable for graphic shapes.
- Smooth Angle controls normal continuity, not geometric width.

BEVEL PROFILE ASSETS

Begin with one of the 22 profiles, then edit anchors, Bezier tangents, snapping, smoothing, mirroring, reversal, and baked rings. The runtime consumes the resolved deterministic samples. Use scene handles on a selected consumer to judge the profile against the actual word, depth, lighting, and material.

DECORATION PROFILE

Trim, Groove, Bead, and Lip add material-addressable bands around the cap boundary. Width sets the lateral footprint. Height controls relief, with zero selecting a width-relative automatic

value. Detail affects curved profiles but remains dormant for a simple Trim band. Decoration profile triangles use the Trim Rim region.

DEPTH SHAPING AND DEFORMATION

Taper changes cross-section scale through depth. Twist rotates it. Bulge expands or contracts the middle of the body. Apply restrained values first and inspect counters and concave corners from the side. A Deformer Stack acts on finished geometry and can be ordered after the authored depth shape.

TRUE OUTLINES AND STROKE INHERITANCE

Contour expansion and stroke rings create real outline geometry, not a duplicated offset text object. A foreground layer can remain per-glyph while a grouped layer behind it creates a connected word outline. Stroke inheritance reduces repeated authoring, but a disabled source layer no longer participates in inheritance or depth stacking.

CREATE LOGOS, BACKINGS, AND CUSTOM CONTOURS

CHOOSE A CONTOUR SOURCE

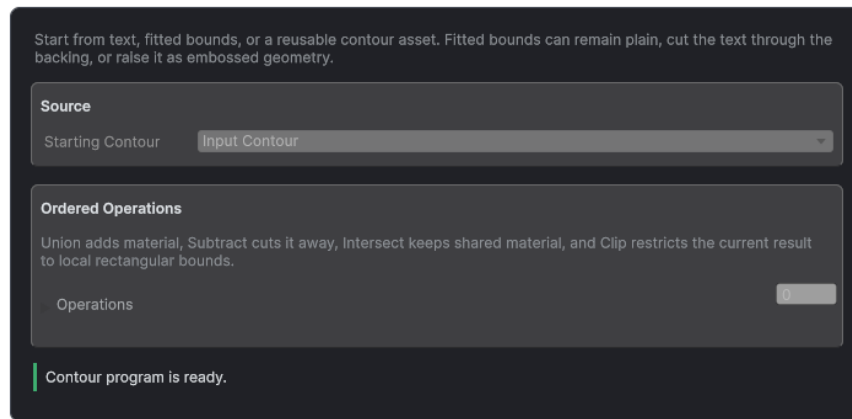
Start from the input glyph, fitted text bounds, or a reusable contour asset. Generated contour assets include rounded rectangles, pills, plates, shields, badges, banners, frames, and bounds-driven shapes.

IMPORT AN SVG

SVG import supports line, cubic, quadratic, smooth, and arc commands, nested transforms, compound holes, and normalized coordinates. Use it for logos and custom silhouettes, then tune curve sampling for the target size.

BUILD A CONTOUR PROGRAM

Contour programs combine ordered Union, Subtract, Intersect, and Clip operations with transforms. Use them for logo cutouts, inset plates, grouped outlines, and reusable backing recipes.



A Contour Program starts from text, bounds, or an asset, then applies ordered shape operations.

COMPOSE A BACKING

Text-bounds backings can remain plain, cut the text out, or raise it as an embossed face.

Padding, depth, corner radius, frame width, and grouped word, line, or whole-text scope make the result adaptable to signs and title cards.

GENERATED CONTOURS

Generated Contour assets provide Rounded Rectangle, Pill, Plate, Shield, Badge, Banner, and Frame shapes. Text Bounds fits one of those forms around a glyph, word, line, or the complete text using signed XY padding. Radius, detail, and frame width participate in topology identity.

SVG IMPORT

The editor importer accepts SVG path text with line, cubic, quadratic, smooth, and arc commands, nested transforms, compound holes, and normalized coordinates. Choose scale, optional Y flip, centering, and curve sampling deliberately. The importer rejects unsafe external XML resolution and returns validated contour data.

- Remove invisible duplicate paths and self-intersections before import.
- Confirm outer and hole loop direction in Live Preview diagnostics.
- Use enough sampling for the final size, but do not turn a simple logo into excessive topology.
- Test the contour alone before adding bevel, body, or boolean operations.

CONTOUR PROGRAMS

A Contour Program starts from the input glyph, a reusable source asset, or fitted Text Bounds, then applies ordered Union, Subtract, Intersect, and Clip operations with transforms. Operation order matters. Inspect the result after each step when an empty, inverted, or fragmented contour appears.

Text-bounds composition can keep a plain backing, subtract the text for a through-cut, or emit the backing and raised text as an embossed compound. Emboss attaches text to the measured front surface and currently requires a linear body sweep.

BACKING TUTORIAL

1. Create a Generated Contour using Pill and fit it to Whole Text bounds.
2. Add positive XY padding and a restrained corner radius.
3. Choose Plain for a solid plate, Cutout for through-cut lettering, or Emboss for raised text.
4. Assign distinct Front Face, Sides, and Back materials.
5. Inspect the result from both sides and validate the boolean result before adding decorations.

ADD TUBES, REPEATS, AND ATTACHMENTS

CONTOUR TUBES

Run a round or custom tube around the front, back, or both. Control holes, clearance, path detail, and the tube material independently from the letter body.

CONTOUR REPEATS

Choose Dash, Stitch, Bulb, Marquee, Rivet, Rail, Bead, Spike, Gem, Bolt, or Prefab. Control spacing, count, density, phase, corner behavior, overlap avoidance, and repeatable position, scale, and rotation variation. Prefab repeats accept ordinary nested model prefabs, including models whose visible mesh is stored on a child object.

NAMED GLYPH SOCKETS

Create sockets at the center, front, back, top, bottom, or a contour fraction. Give each a stable name, position and orient it with scene handles, then attach particles, props, lights, gameplay objects, or accessibility markers.

CONTOUR TUBES

ContourTube3DSpec sweeps a round or custom closed profile around front, back, or both transformed cap boundaries. It can include counters, maintain clearance from the letter surface, and bound path work by physical spacing and maximum samples. An unassigned profile uses an analytic round section. A custom tube profile contains 3 to 64 validated points.

CONTOUR REPEATS

Repeats place Dash, Stitch, Bulb, Marquee, Rivet, Rail, Bead, Spike, Gem, Bolt, or readable root-prefab mesh providers along contour distance. Distribution can use physical Spacing, exact Count per loop, or Density per unit. Phase offsets the resolved interval.

- Uniform places providers continuously.
- Corner Clearance keeps instances away from hard turns.
- Corners Only places at detected hard turns.
- Overlap Avoidance uses authored size and orientation.
- Seeded position, scale, and rotation variation remains deterministic.

Choose Face, Tangent, Outward Normal, or Custom orientation. Repeats can include counters or both cap surfaces and use the independent Decorations material region. Prefab repeats accept a normal nested model whose visible mesh is stored on a child object.

MARQUEE TUTORIAL

1. Enable Contour Repeats and choose Marquee or Bulb.
2. Start with Spacing, then tune size so adjacent bulbs remain separate at corners.
3. Use Corner Clearance and Overlap Avoidance for a clean border.
4. Assign an emissive Decorations material while keeping the letter Front Face readable.
5. Change the seed only when variation is enabled and verify the result remains stable after reload.

GLYPH SOCKETS

Sockets can target Center, Front, Back, Top, Bottom, or a normalized Contour Fraction. Give each socket a stable project name and use scene handles to adjust its local pose. Runtime code can locate sockets without depending on generated hierarchy details.

- Attach particles to a named impact point.
- Place a light on the front of a sign.
- Attach a gameplay object to a letter while preserving glyph identity.
- Expose accessibility or navigation markers independently from visible geometry.

DESIGN MATERIALS AND SURFACES

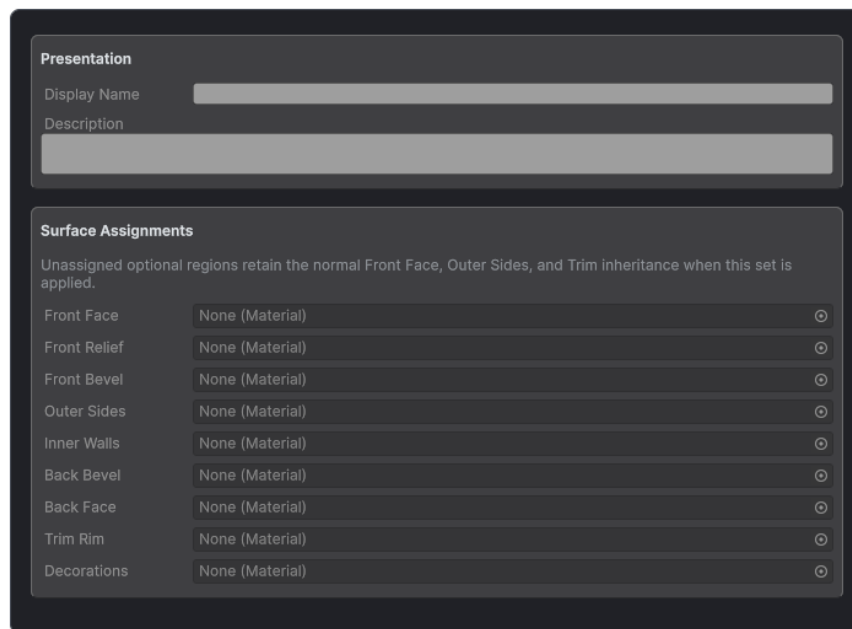
Materials are assigned by meaning rather than by mesh order. The nine built-in regions are Front Face, Front Relief, Front Bevel, Outer Sides, Inner Walls, Back Bevel, Back Face, Trim, and Decorations. Custom modules can add more regions.

START WITH A MATERIAL SET

Ten beginner Material Sets provide coordinated starting points. The Material Set browser previews materials as lit spheres before they are applied. Regions that share one material are compacted into fewer render passes automatically.

CHOOSE TEXTURE MAPPING

- Generated for robust default coordinates.
- Planar Glyph or Planar Whole Text for fronts and graphic treatments.
- Side Distance or Depth for extrusion-aware surfaces.
- Object or World Triplanar for stone, metal, and large environmental text.
- Cylindrical for wrapped surfaces.



A Material Set maps reusable materials to semantic regions.

Each mapping can adjust scale, rotation, offset, anchor, and physical density. Built-in shaders support Built-in, URP, and HDRP, including lit, glass, frosted, emissive, instanced, and motion-vector-aware materials. Mesh baking can create lightmap UVs.

MATERIAL REGIONS AND FALLBACK

The nine built-in regions are Front Face, Front Relief, Front Bevel, Outer Sides, Inner Walls, Back Bevel, Back Face, Trim Rim, and Decorations. Shaped front caps use Front Relief, outer bevels use the bevel regions, counter walls use Inner Walls, and tubes or repeats use Decorations.

Unassigned Front Relief inherits Front Face. Front Bevel, Back Bevel, and Inner Walls inherit Outer Sides. Decorations inherit Trim Rim, which retains the ordinary face fallback. This keeps simple styles usable while allowing deliberate separation.

MATERIAL SETS

A Material Set stores assignments for every built-in region without changing geometry or mapping. Start from one of the ten beginner sets, inspect the lit-sphere previews, apply it, then duplicate before project-specific editing. Exact material reuse lets the renderer compact compatible regions into fewer material sections.

SURFACE MAPPING

- Generated preserves provider coordinates.
- Planar Glyph normalizes each glyph independently.
- Planar Whole Text uses one continuous rectangle across the complete text.
- Side Distance maps contour distance and depth along side surfaces.
- Depth projects horizontal position against local extrusion depth.
- Cylindrical wraps around the glyph axis.
- Triplanar Object blends projections in object space.
- Triplanar World blends projections in world space.

Scale, clockwise rotation, offset, and normalized anchor apply after projection. Consistent World Scale keeps physical coordinates instead of per-bounds normalization where applicable. Choose mapping based on the intended pattern continuity, then inspect fronts, counters, sides, and back.

RENDER PIPELINE MATERIAL FAMILIES

Bundled material families cover Built-in, URP, and HDRP with lit, glass, frosted, emissive, instanced, and motion-vector-aware variants. After changing pipeline assets or shaders, validate real lit and glass text in the target render pipeline, including shadows, depth, motion vectors, transparency, and baked lighting. Mesh baking can create lightmap UVs for static lighting.

MATERIAL RECIPES

- Chrome: readable bevels, high metallic and smoothness, directional lighting, and restrained Chrome Sweep.

- Frosted glass: enough body thickness to read refraction, a distinct inner wall, and a non-glass backing or environment.
- Neon: contour tube or outline geometry, emissive Decorations or Trim, and a darker non-emissive body.
- Painted metal: moderate bevel, painted Front Face, metal Sides, and controlled roughness.
- Stone: sufficient bevel and relief to catch light, triplanar mapping, and restrained surface scale.
- Holographic glass: transparent material, readable silhouette, Hologram Scan, and a Reduced Motion still state.
- Marquee: independent bulb geometry and emissive Decorations with a stable readable letter face.

THEME COMPLETE 3D EXPERIENCES

Themes let one art-direction choice travel through a complete 3D experience. Change a prepared font, menu language, visual family, control response, or gameplay-feedback treatment without walking through every title, button, label, and popup by hand.

PROJECT AND HIERARCHY THEMES

Create Assets > Create > Text Studio > 3D > Setup > Theme. A Text Studio 3D Theme can provide a shared TMP font, matching prepared Font3D data, a default Extrude Style, named style variations, automatic control-label and gameplay-feedback mappings, and one Control Theme. Text Studio still owns the authored content and TMP font component; the 3D theme supplies inherited presentation to objects that opt into its font or style.

Create Project Theme Settings when one theme should be the project default. Add a Theme Scope to any parent transform when a room, menu, faction, level, or in-world terminal needs a different theme. The nearest enabled scope wins; an empty scope falls through to its parent or the project theme.

A useful theme boundary follows player experience. Theme the cockpit, storefront, dungeon menu, or tactical HUD as one unit instead of assigning unrelated styles object by object.

NAMED STYLES AND CONTROL THEMES

Keep one uncomplicated Default Style, then add named variations such as Menu Title, Warning, Quest Marker, Score, or Boss Damage. A themed text object can bind to Default, one named variation, Control Label, or Gameplay Feedback. Control Label and Gameplay Feedback fall back to Default when their mapped variation is empty, so a simple theme remains complete.

Create a Control Theme to coordinate Normal, Hover, Focused, Pressed, Selected, Disabled, and Error feedback. Each state can change label tint, surface tint, scale, and depth offset. Focus and selection remain distinct, which makes keyboard navigation, a selected option, validation, and pointer feedback readable at the same time.

BUILD A THEME-SWITCHABLE SPATIAL MENU

1. Create a Control Theme and verify every interaction state against the lighting and camera used by the real scene.
2. Create two 3D Theme assets with different fonts, prepared font data, default styles, and named Menu Title and Gameplay Feedback variations.
3. Create Project Theme Settings for the global default, then add a Theme Scope around the spatial menu so it can change independently.
4. Add GameObject > Text Studio > 3D > World Space UI > Starter Menu. Connect its buttons, toggle, slider, progress bar, dropdown, and navigation to real settings.
5. Use Theme Action components from UnityEvents to apply a theme to the scope. Confirm that titles, control labels, selected options, and feedback all update together.
6. Test mouse, keyboard, controller, and optional XR focus, then verify the same menu in narrow, typical, and wide camera framing.

Open the Spatial Menus and Themes showcase to inspect a complete starter-menu flow, scoped theme switching, global font changes, focus restoration, Depth Stack, Rotary Dial, and Radial Menu selection patterns.

ANIMATE 3D TEXT AND ADD INTERACTION

Text Studio 3D combines native surface and geometry effects with compatible Text Studio motion. Use native effects when depth, bevels, contours, or material regions should participate in the animation. The complete visual gallery appears in the appendix.

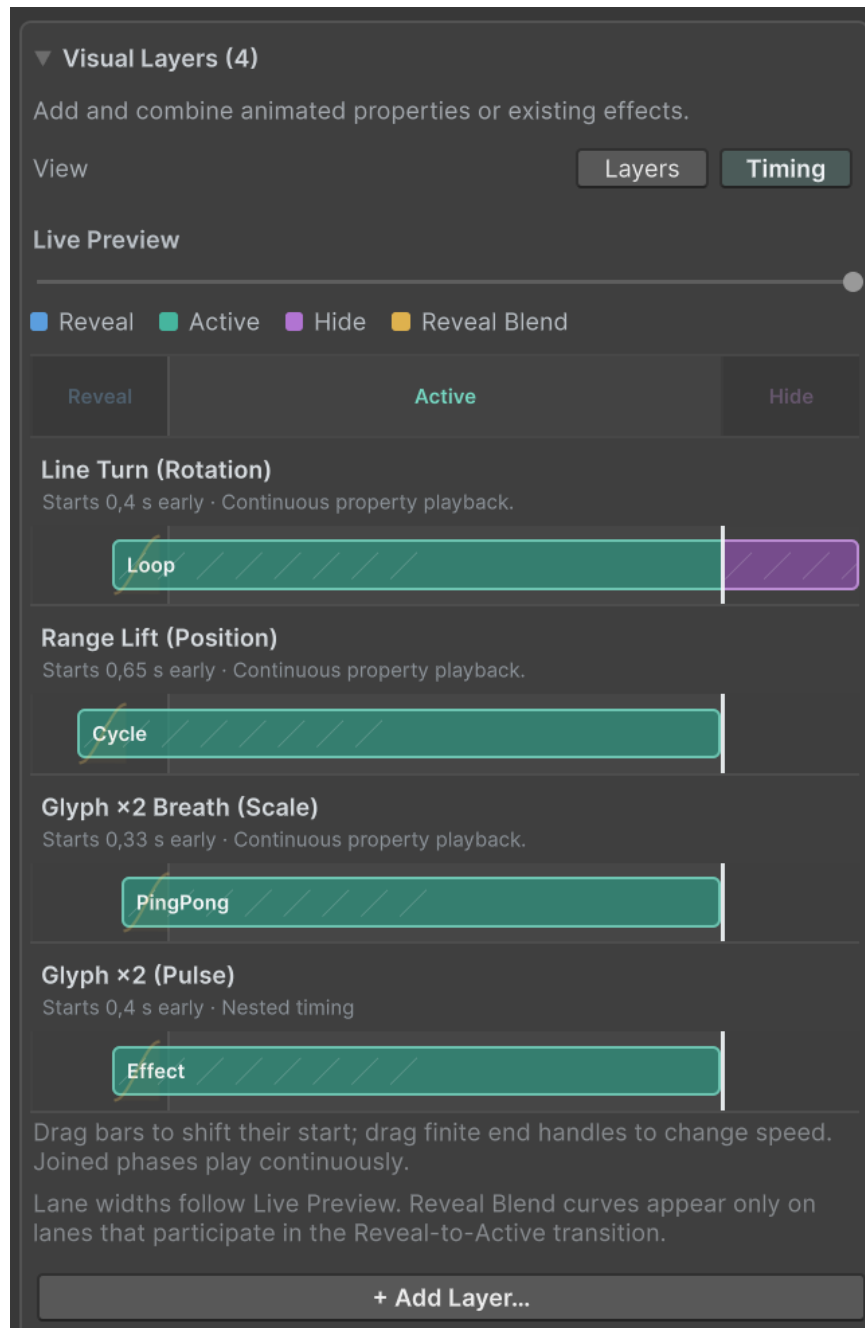
COMPATIBLE TEXT STUDIO EFFECTS

Position, rotation, scale, color, transparency, corner deformation, spatial motion, areas, sequences, motion policy, and Effect Code continue through the normal Text Studio workflow. Combined, chunked, hybrid, and LOD output support the full compatible channel set. Per-letter and Physical output support position, rotation, and scale.

COMPOSED EFFECT TIMING

Three-dimensional Composed Effects use the same Timing view as base Text Studio. Switch Visual Layers from Layers to Timing to align native surface and geometry effects with compatible transform, color, alpha, material, and placement channels. Scrub the shared lifecycle, drag bars to

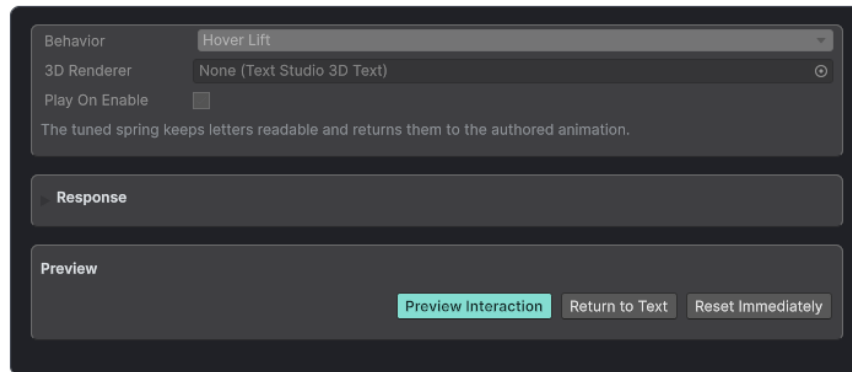
move their start, and drag finite end handles to change playback speed while joined phases continue without restarting.



The shared Timing view coordinates compatible Text Studio layers and native 3D channels against one Live Preview lifecycle.

NO-CODE INTERACTION

Hover Lift, Cursor Repel, Magnet Pull, Impact Ripple, Blast and Return, and Gravity Exit react to a shared spatial target, strength, radius, and motion policy. Use Preview Interaction before connecting gameplay input.



The Interaction controller previews a chosen response before runtime input is connected.

BASE EFFECTS AND OUTPUT COMPATIBILITY

Text Studio continues to evaluate position, rotation, scale, color, alpha, corner deformation, areas, sequences, influence fields, and motion policy. Combined, Chunked, Hybrid, and LOD output report the complete shared channel set. Per Letter and Physical output currently support position, rotation, and scale through their isolated object paths.

Use renderer diagnostics when a Composed Effect contains a channel that the selected output cannot represent. Do not assume that switching output is a performance-only choice, because it can also change interaction and channel capabilities.

NATIVE 3D EFFECTS

- Neon Chase: travelling emission and color highlight without material instances.
- Chrome Sweep: metallic, smoothness, and emissive highlight across polished text.
- Hologram Scan: emission, transparency, color, and restrained depth shimmer.
- Facet Pulse: bevel or edge illumination selected by a surface mask.
- Glass Reveal: shader dissolve, smoothness, and luminous reveal edge.
- Depth Wave: topology-stable scaling of existing extrusion depth.
- Contour Bloom: quantized topology refresh that expands the glyph contour.
- Bevel Pulse: quantized topology refresh that scales authored bevel geometry.

Contour Bloom and Bevel Pulse are rebuild channels. Topology Effect Refresh Rate and Quantization Steps cap their update cost. Use lower frequency and fewer visible hosts before reducing ordinary transform or material effects.

INTERACTION CONTROLLER

TextStudio3DInteractionController resolves a sibling renderer by default and can follow an explicit world-space target or the base hub's Influence Target. Hover Lift, Cursor Repel, Magnet Pull, Impact Ripple, Blast and Return, and Gravity Exit share Strength, Radius, Play On Enable, Reduced Motion, Preview, and Reset controls.

FORCES

Forces can apply gravity, wind, radial blast, vortex, or attraction to retained glyph state. Gravity Drop uses Text Studio force simulation and can be previewed in Edit Mode or Play Mode. Physics Drop switches to Physical Letters, creates Rigidbody and collider letters, and releases them to Unity physics, which advances in Play Mode; the output-mode change supports Undo. Reset Forces stops simulation and returns letters to their authored positions.

PHYSICAL LETTERS

Physical output creates pooled independent letters with convex collision proxies. Use it only when collisions, release, gravity, or per-letter rigidbody behavior are the product feature. Decoration geometry remains visible but is excluded from generated convex proxies. Reset, pause, containment, and return behavior must be tested with the real maximum letter count.

RUNTIME OVERRIDES

Runtime overrides can change visibility, depth, tint, material, and quality without rewriting the authored style. Material and tint changes can remain visual updates. Text, font, contour, bevel, body, sweep, decoration, grouped scope, and topology modules may require a topology build. Property setters request the correct work; explicit in-place asset edits finish with `RebuildGeometry`.

ARRANGE TEXT IN SPACE

The Spatial 3D placement extension adds Helix, Cylinder, Sphere Band, Tunnel, Staircase, and Spatial Grid forms. Curated presets include DNA Helix, Hero Carousel, Globe Band, Tunnel Spiral, Dimensional Stair, and Voxel Wall.

- Keep Original preserves each glyph orientation.
- Follow Path turns glyphs with the arrangement.
- Face Outward and Face Inward orient glyphs around a volume.
- Reverse and fine rotation controls correct reading direction and camera alignment.

Spatial 3D

This component contributes a spatial placement to the sibling Text Studio hub. Activate it here or from Text Studio's Placement tab.

[Use Spatial 3D Placement](#)

Starting Point

Presets set placement and orientation only. Geometry style, materials, reveal, and composed effects remain unchanged.

[Choose Polished Starting Point...](#)

Form

Form

Curved Form

Auto Fit Radius ☐

Radius

Start Angle

Revolutions

Turns

Reverse Order ☐

Orientation

Orientation

Fine Rotation X Y Z

Keep Original is the calmest choice. Face Outward is ideal for carousels and globes; Follow Path emphasizes motion through a helix or tunnel.

Spatial Placement chooses a form, starting point, orientation, and form-specific values.

PLACEMENT VERSUS EXTRUSION SWEEP

Base Placement moves each complete glyph instance through a line, arc, circle, wave, grid, spline, or spatial form. Extrusion Sweep bends one glyph's front-to-back body. A Deformer changes finished geometry. All three can be active, but they answer different design questions.

QUADRATIC BEZIER EXTRUSION PATH

Use the built-in quadratic asset for Straight, Side Bend, Lift, or Ribbon starting points. Configure start, control, and end offsets, samples, banking, accumulated roll, and front or back scale. The asset carries a deterministic cache identity that changes when sampled output changes. Scene handles edit the selected asset consumer, while component-local controls remain available without an asset.

UNITY SPLINES EXTRUSION

Install Unity Splines 2.0.0 or newer and import the Text Studio 3D Unity Splines integration. SplineExtrusionPathProvider samples one child spline into renderer-local immutable path data, including rotation-minimizing frames, optional knot banking, roll, endpoint scale, and closed seams. It refreshes after relevant knot, collection, or transform changes and clears only the snapshot it assigned.

SPATIAL PLACEMENT REFERENCE

- Helix: radius, turns, height or depth, and orientation.
- Cylinder: radius, angular range, rows, and facing.
- Sphere Band: radius, longitude, latitude band, and facing.
- Tunnel: radius, turns, depth, and view direction.
- Staircase: step size, rise, turn, and orientation.
- Spatial Grid: columns, rows, layers, and 3D spacing.

Curated presets provide DNA Helix, Hero Carousel, Globe Band, Tunnel Spiral, Dimensional Stair, and Voxel Wall. Keep Original, Follow Path, Face Outward, and Face Inward control orientation. Reverse and fine rotation correct reading direction and camera alignment.

DEFORMER STACKS

Bend, Arc, Shear, Pinch, Flare, Spherize, Cylindrical Wrap, Wave, Noise, and Twist can be ordered in Glyph, Word, Line, Text, Object, or World space. Automatic or custom bounds and edge falloff localize the result. In stable combined, chunked, or hybrid output, warmed built-in deformation can rewrite visual buffers without replacing topology.

- Order broad structural deformation before small surface variation.
- Use the narrowest semantic space that matches the intended result.

- Preview normals and silhouettes, not only front-facing color.
- Use custom bounds when a world or object-space effect should remain fixed as text changes.

BUILD WORLD-SPACE INTERFACES

Create a Panel, Starter Menu, Button, Toggle, Slider, Progress Bar, List, Dropdown, Input Field, or Selector from GameObject > Text Studio > 3D > World Space UI. Each control uses real Text Studio 3D output and can share inherited fonts, named styles, interaction feedback, events, navigation, and accessibility metadata.

- Pointer Raycaster connects mouse, touch, or ray pointers.
- Navigation Group provides keyboard and controller traversal.
- Control Layout arranges controls in Linear, Grid, Circular, Spline, or Volume forms.
- Control Pool keeps repeated or changing interfaces bounded.
- Optional Input System and XR Interaction Toolkit packages add dedicated adapters.

Open the World Space UI production sample to inspect all controls, visual states, navigation, and accessibility fields in one persisted scene. Toggle, Slider, List, Input Field, and Selector use distinct affordances and clearer interaction states, while the shared Linear, Grid, Circular, Spline, and Volume layout modes are demonstrated and explained separately.

CANVAS AND CAMERA CHOICES

Use ordinary world-space objects for signs, diegetic panels, control rooms, and XR interfaces. Text Studio 3D also supports World Space Canvas and Screen Space Camera content when the Canvas helps organize a HUD or mixed UI. Screen Space Overlay cannot display real 3D geometry correctly, so the editor offers guided conversion instead of silently flattening or hiding the result.

Choose Screen Space Camera for camera-composited 3D HUD feedback, World Space Canvas for interface objects that live in the scene, and ordinary transforms for fully spatial menus and controls.

CONTROL FOUNDATION AND STATES

Every 3D control combines a Text Studio label, Control Theme, interactable state, validation state, accessibility name and hint, and UnityEvents. Normal, Hover, Focused, Pressed, Selected, Disabled, and Error state flags can combine without losing meaning. Their tint, scale, and depth feedback update without rebuilding text topology.

BUTTONS, INPUTS, PROGRESS, LISTS, AND DROPDOWNS

- Button exposes click, pointer, keyboard, focus, and accessible label behavior.
- Toggle owns a Boolean value, state labels, value-changed event, and selected presentation.
- Slider owns minimum, maximum, value, whole-number mode, step, track, prefix, format, and change events.
- List owns options, selected index or value, selection bounds, navigation, prefixes, and change events.
- Selector owns options, current index, optional wrap, previous or next input, display, and change events.
- Progress Bar owns minimum, maximum, current value, numeric formatting, label presentation, and value-changed events without exposing an interactive drag state.
- Dropdown keeps a compact selected value until opened, then presents options and reports open, index, and option changes.

INPUT FIELD

Input Field owns value, placeholder, maximum length, read-only mode, required validation, focus, caret, built-in keyboard path, submit, value-changed, and validation events. Use the optional Input System adapter for project input actions and test Unicode entry with the active font and shaping backend.

MENU FLOW AND OPTION PRESENTATION

Menu Controller owns the current panel, bounded back history, Back, Close, and Close All behavior. Menu Action turns those commands into UnityEvent-friendly components, so a designer can build a complete Settings, Audio, Accessibility, or Equipment flow without a parallel menu framework.

Option Presentation arranges a group as a stable Depth Stack, rotary choice, radial selection, or other authored presentation while keeping selection events and navigation identity consistent. Use it when the spatial arrangement communicates meaning, such as weapons around a radial wheel or audio modes on a dial.

POINTER, NAVIGATION, LAYOUT, AND POOLS

Pointer Raycaster chooses camera, layers, maximum range, trigger interaction, and built-in mouse behavior. Navigation Group owns ordered keyboard or controller traversal, inactive-control filtering, wrap, first selection, and selection events. Navigation focus remains independent from a List or Selector value.

Control Layout arranges controls as Linear, Grid, Circular, Spline, or Volume with form-specific spacing, rows, columns, radius, angles, orientation, and points. Control Pool bounds prefab reuse and must restore value, validation, focus, events, and visual state before returning an instance.

CONTROL THEMES AND BOUNDED CONTROL POOLING

TextStudio3DControlTheme is the shared interaction treatment for world-space controls. TextStudio3DControlPool reuses complete control prefabs and restores their value, focus, validation, events, and visual state before reuse. Temporary labels, signs, and gameplay feedback use the base TextStudioEmitter with a complete 3D host prefab. Keeping those owners separate prevents a menu pool from becoming a second temporary-text system.

WORLD-SPACE UI PRODUCTION TUTORIAL

1. Create a Starter Menu, or create a Panel and add a Button, Toggle, Slider, Progress Bar, List, Dropdown, Input Field, and Selector.
2. Assign one Control Theme, choose the inherited theme style for control labels, and verify focus, selection, pointer, disabled, and error combinations.
3. Add Pointer Raycaster and Navigation Group, then test mouse, keyboard, and controller focus.
4. Connect Menu Controller where the experience has multiple panels. Arrange repeated controls with Control Layout or Option Presentation, then verify narrow, typical, and wide camera framing.
5. Add Input System or XR adapters only after the built-in pointer and navigation paths work.
6. Fill every accessibility name and hint and test disabled or invalid controls.

CHOOSE OUTPUT AND PERFORMANCE

CHOOSE AN OUTPUT MODE

- Combined Mesh: the simplest static and animated dimensional output.

- Chunked Combined Mesh: efficient updates when only part of changing text needs rebuilding.
- Per Letter GameObjects: independent objects for authored per-letter workflows.
- Hybrid: combined readability with selected independent behavior.
- LOD Group: automatic distance-based detail.
- Physical Letters: pooled letters with convex collision proxies for physics.

START FROM A PERFORMANCE PROFILE

Static Quality favors final detail. Realtime balances changing text and effects. Massive Text lowers cost for large populations. Interactive supports responsive controls and forces. Physical prepares pooled independent letters. Adjust lower-level quality only after the closest profile has been measured in the real scene.

MESH DETAIL PRESETS

Draft, Balanced, High, and Hero coordinate curve tolerance, bevel rings, cap and relief subdivisions, body steps, decoration detail, contour tubes, and depth-sweep sampling. Use Draft while shaping and arranging, Balanced for ordinary real-time content, High for close readable text, and Hero for final capture or unusually close presentation. Choose Custom only when the coordinated presets cannot express the measured requirement.

Mesh Detail answers how finely geometry is built. Performance Profile answers how that geometry is owned and updated. Keep those decisions independent: a Hero title can be static or frozen, while a dense interactive menu can use Balanced detail with the Interactive workload profile.

CHANGING TEXT AND RUNTIME OVERRIDES

Appended and changed text can reuse existing chunks and unchanged detailed letters. Runtime overrides can change visibility, depth, tint, material, and quality without rewriting the authored style. For repeated temporary labels, use TextStudioEmitter with a complete 3D host so the normal Text Studio lifecycle, profiles, aggregation, stacking, follow behavior, and Unity-backed pooling remain shared with flat text. Prewarm the characters and emitter hosts that must appear without a first-use build; the reusable geometry workspace starts smaller and grows only when detailed lettering requires more capacity.

Dense combined 3D text now responds more quickly when several effects update most of a detailed mesh. Three-dimensional effects and shared animation channels restore automatically across repeated Fast Enter Play Mode sessions, while player-build validation checks the bundled materials, Material Sets, shaders, and default font data.



Combined output remains readable under a large changing-text workload.

FREEZE A FINISHED TITLE

Strict Freeze captures the exact visible frame as persistent static geometry. Choose Freeze Current Appearance under Geometry Actions after setting the desired preview progress. The current glyphs, colors, transparency, placement, deformation, materials, LOD membership, and mesh colliders remain, while Text Studio generation, animation, tags, events, forces, sockets, Scramble progression, and runtime text changes stop.

Use Unfreeze to restore live editing. The generated snapshot is removed after the last frozen copy is restored. Freeze works on one selected scene object or prefab instance, not a Prefab Mode asset. Physical letters keep their captured poses and colliders, but rigidbody simulation stops.

CHOOSE OUTPUT BY WORKLOAD

Combined Mesh is the simplest static and fully compatible output. Chunked Combined Mesh limits replacement work when changing text preserves parts of the previous output. Per Letter GameObjects provide independent objects. Hybrid combines retained output with selected independent behavior. LOD Group produces distance-based detail. Physical Letters adds pooled rigidbody-ready objects.

- Static title or sign: Combined Mesh or Strict Freeze.
- Frequently changing short label: Realtime profile with Chunked Combined Mesh.
- Large passage or many glyphs: Massive Text with measured chunk size and automatic LOD.
- Raycastable sign or control: Interactive profile and one root collider where possible.
- True collisions or falling letters: Physical profile and Physical Letters.

PERFORMANCE PROFILES

Static Quality favors final detail. Realtime balances changing text and effects. Massive Text lowers cost for large populations. Interactive preserves responsive controls and forces. Physical configures independent letter behavior. Profiles never replace authored layers or materials. Apply the closest workload first, then tune measured bottlenecks.

QUALITY, LOD, AND TOPOLOGY EFFECTS

Curve Tolerance controls outline sampling. Smooth Angle controls normal continuity. Automatic LOD chooses detail bands. Parallel Vertex Threshold decides when the automatic animation backend can profitably use parallel work. Topology Effect Refresh Rate and Quantization Steps bound Contour Bloom and Bevel Pulse.

RUNTIME STATISTICS

Use glyph, entry, vertex, triangle, mesh, renderer, and material counts to understand output size. Use topology and visual update times to distinguish rebuild work from warmed animation. Use cache hits, misses, replacements, and evictions to understand reuse. Use output-object and collider counts to detect hierarchy growth. Use geometry validation only when diagnosing integrity.

PREWARMING AND BOUNDED CACHES

PrewarmGlyphs synchronously warms at most 256 unique Unicode scalars per call without replacing the visible mesh. Use loading screens or pool warm-up, not a frame with strict responsiveness. Mutable custom paths, grouped contour layers, and topology-changing modules can be renderer-specific and report unsupported prewarming.

The shared geometry cache is bounded by entry count and 64 MiB estimated mesh payload. Local outline and codepoint caches are also bounded. Frequent eviction usually means too many simultaneous fonts, quality levels, path identities, or style variants. Reduce unnecessary variability before retaining more data.

SCHEDULE VERSUS REBUILD

RebuildGeometry requests topology immediately through the normal hub refresh. ScheduleGeometryBuild coalesces repeated requests into one later component update and can be canceled before dispatch. Scheduling is deferred main-thread work, not background mesh generation. Do not call either method every frame.

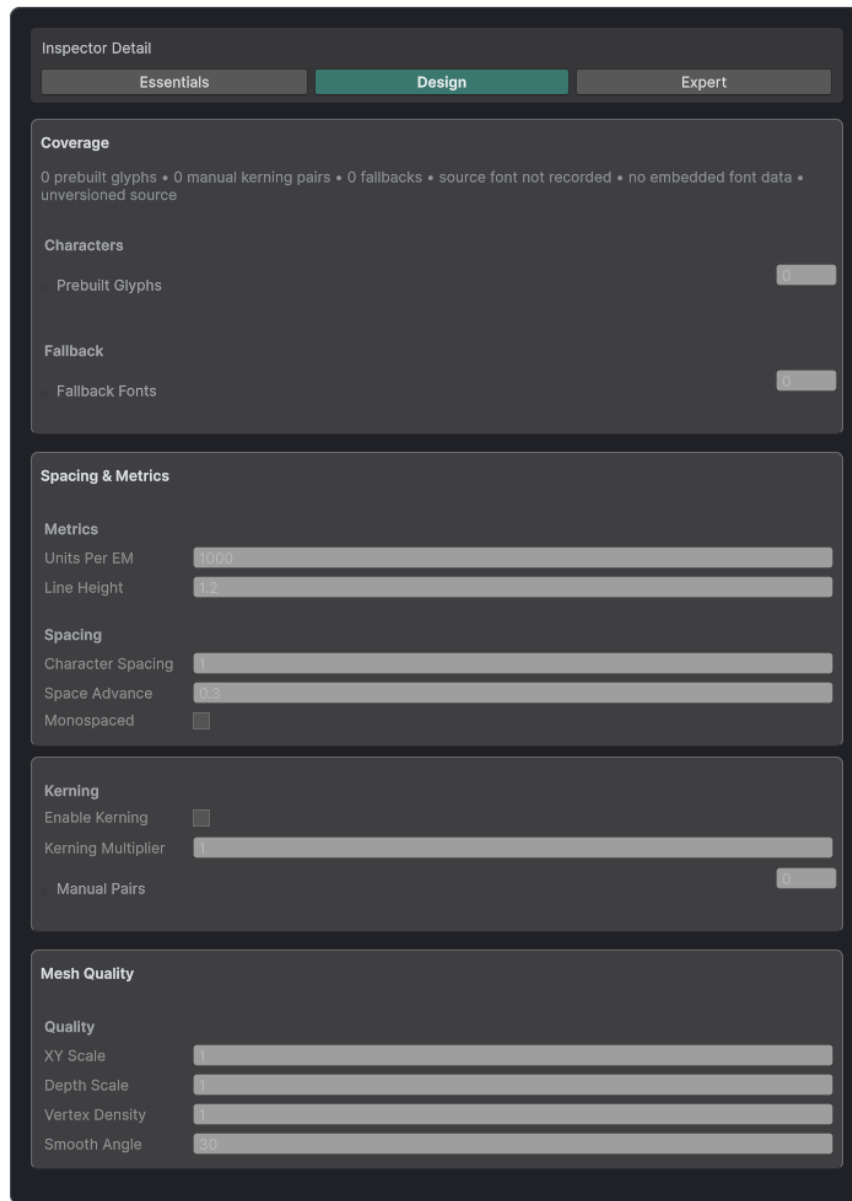
STRICT FREEZE

Freeze Current Appearance captures the exact visible frame as persistent static geometry. The snapshot retains current glyphs, colors, transparency, placement, deformation, materials, LOD membership, and mesh colliders. Text Studio generation, animation, tags, events, forces, sockets, Scramble progression, and runtime text changes stop.

Unfreeze restores live editing and removes generated snapshot data after its last frozen owner is restored. Freeze is an editor transaction for a selected scene object or prefab instance, not a runtime API for Prefab Mode assets. Treat frozen output as ordinary static meshes in player code.

PREPARE FONTS AND LANGUAGES

The font selected by the active Text Studio backend is authoritative. Setup Status can prepare matching embedded Font 3D data when runtime outlines are required. TextMeshPro uses its selected font and fallbacks; UniText supplies shaped glyphs and visual order from standard TTF or OTF sources. Prebuilt glyph coverage, fallbacks, spacing, metrics, kerning, and mesh quality remain inspectable on the prepared font asset.



Prepared Font assets show coverage, fallbacks, spacing, kerning, and mesh quality.

BUNDLED FONTS AND TYPE FOUNDRY

Text Studio 3D includes Bebas Neue Regular, Bungee Regular, Monoton Regular, Audiowide Regular, Cinzel Decorative Bold, Press Start 2P Regular, and ZCOOL QingKe HuangYou Regular. Each family includes its original OFL source, a matching dynamic TextMeshPro asset, and prepared player-ready 3D outline data.

Open the Type Foundry showroom to compare all seven as live editable extrusion. The ZCOOL station renders genuine Chinese lettering from the complete bundled font source, without a phrase subset, fallback font, or replacement mesh.

FONT BAKER

Choose a TTF, OTF, or TMP source, a target Font 3D asset, a built-in or custom character set, curve tolerance, extrusion depth, bevel, smooth angle, embedded source option, and kerning. TrueType and CFF1 OpenType outlines, the default instance of variable fonts, supplementary Unicode, combining marks, fallbacks, and multilingual content are supported. Color-only and CFF2-only sources require another font source.

FONT OWNERSHIP AND PREPARED DATA

The active Text Studio backend chooses the font. TextMeshPro supplies its font asset and fallbacks. UniText supplies shaped glyph IDs, visual order, and font faces. Font3D is matching technical data for runtime outline access and prebuilt meshes, not a second typography choice.

PREPARED FONT INSPECTOR

Inspect character coverage, prebuilt glyphs, fallback chain, units per em, ascent, descent, line gap, spacing, monospace behavior, kerning, XY and Z scale, density, smoothing, source font identity, embedded bytes, and schema version. A prepared asset must match the active source font before the renderer consumes it.

FONT BAKER WORKFLOW

1. Choose a TTF, OTF, or TMP source and a target Font3D asset.
2. Choose a built-in or custom character set that matches the shipped languages.
3. Set curve tolerance and preview depth, bevel, and smooth angle at the target display size.
4. Choose whether to embed source bytes for runtime outline generation and include kerning.
5. Bake, inspect coverage and fallbacks, then test missing glyph diagnostics in a player-compatible setup.

TrueType and CFF1 OpenType outlines, the default instance of variable fonts, supplementary Unicode, combining marks, fallbacks, and multilingual content are supported. Color-only bitmap or layered glyphs and CFF2-only sources require another outline-capable font source.

LANGUAGE VALIDATION

- Latin, Greek, and Cyrillic: verify coverage, accents, kerning, and fallback transitions.
- CJK: bake a deliberate production range or provide runtime outlines rather than an impractical unbounded set.
- Arabic and Hebrew: validate shaping, ligatures, bidirectional visual order, punctuation, and mixed-script content through UniText.
- Combining marks: confirm the mark follows its base glyph through extrusion and animation.
- Fallbacks: verify that the chosen fallback face exposes standard TTF or OTF outline bytes.

BUILD ADVANCED REUSABLE MODULES

Use ordinary Text Studio effects first. Use 3D modules when a reusable design must change contour, relief, sweep, deformation, decoration, post-processing, lifecycle behavior, layout, or spawned glyph objects.

- Layout Transform modules reshape placement after the base layout.
- Deformer Stack assets package ordered topology-preserving deformation.
- Glyph Prefab Spawners attach repeated objects to glyphs or sockets.
- Module Inputs expose reusable designer-controlled values.
- Custom modules can register semantic material regions and reuse provided scratch buffers.

Keep the user guide simple: Module API names belong in the developer appendix. The main workflow should begin with the visible result and the asset that produces it.

MODULE MENTAL MODEL

Use ordinary Text Studio effects for presentation. Use 3D modules when a reusable design must change contour, relief, sweep, deformation, decoration, post-processing, lifecycle behavior, layout, or spawned glyph objects. Typed module roles use the existing staged pipeline rather than creating a second plug-in system.

LAYOUT TRANSFORM AND DEFORMER MODULES

Layout Transform modules reshape placement after the base layout. Deformer Module packages an ordered topology-preserving stack with spaces, bounds, and falloff. Stable combined, chunked, and hybrid output can reuse cached source geometry for warmed built-in deformation.

GLYPH PREFAB SPAWNER

Spawn Module owns one prefab instance per rendered glyph rather than per geometry layer. It can use glyph positions or named sockets, authored delay and lifetime in Play Mode, and a bounded maximum of 256 instances per module and host. Lowering the limit trims retained instances.

MODULE INPUTS AND CUSTOM MODULES

Module Inputs expose designer-controlled values on a reusable module. Custom modules may register semantic material regions, request retained scratch buffers, and declare stages, cache impact, required channels, bounds behavior, threading support, and geometry impact. Reusable ScriptableObject modules must keep per-host mutable state on a host-owned runtime object.

Extension rule: declare the narrowest truthful stages and channels. A conservative all-stage topology-changing module is safe but prevents useful caching and warmed visual updates.

LEARN FROM THE SHIPPED SAMPLES

- 3D Showcase and Type Foundry: browse the 80-style library, then compare the seven bundled fonts as live editable extrusion, including complete Chinese ZCOOL lettering. UniText 3D integration: inspect shaped Latin, Arabic, and Hebrew extrusion, fallbacks, visual order, live typography, and Text Studio effects in one persisted scene.
- Capability Stories: style authoring, signage, logo backings, sweep and deformation, contour decoration, material families, physical letters, and runtime updates.
- Realtime Showcase: Kinetic 3D Path, Live Type, Impact Blast, and Rigid Cascade.
- NeonForge 3D Lab: a holistic Material, Shape, and Motion experience.
- Advanced Modules: base effect, layout transform, glyph spawner, and module inputs side by side.

- World Space UI: complete controls, Control Themes, focus and selection states, progress, dropdowns, navigation, layouts, pooling, and accessibility.
Spatial Menus and Themes: switch complete menu art direction, inherited fonts, named styles, and control feedback while exploring Starter Menu, Depth Stack, Rotary Dial, and Radial Menu flows.
Gameplay Feedback and HUD: compare world, Screen Space Camera, and complete 3D emission hosts; inspect Damage, Critical, Heal, Status, Combo, and Score presets; change live progress; and observe stacking, merging, target response, and semantic theme colors.
Style Scene Samples: select a reusable 3D Style with no loaded consumer and create one live, linked scene example for focused authoring and comparison.

OPTIONAL INTEGRATION INSTALLATION

Install Text Studio, Text Studio 3D, and the companion package first, then import the matching nested integration package. The base add-on must compile and retain clean scenes when Input System, XR Interaction Toolkit, UniText, or Unity Splines is absent.

Integration	Requires	Purpose	Primary setup
Input System	1.4.0+	Pointer, navigation, submit, and text entry	Assign InputActionReference mappings
XR Interaction Toolkit	2.0.0+	XR hover and select events	Wire XR events to the control bridge
UniText	UniText 3 plus base bridge	Shaped mixed-script extrusion sample	Import after all four prerequisite packages
Unity Splines	Splines 2.0.0+	Authored extrusion sweep	Assign provider, container, and child spline

INPUT SYSTEM

Requires Input System 1.4.0 or newer. TextStudio3DInputSystemAdapter maps project InputActionReference assets to pointer, submit, cancel, navigation, and control commands. TextStudio3DInputSystemTextAdapter handles text-entry actions for 3D input fields. Test enable or disable cycles, missing bindings, action cleanup, and representative devices.

XR INTERACTION TOOLKIT

Requires XR Interaction Toolkit 2.0.0 or newer. TextStudio3DXRControlEventBridge forwards XR hover and select events into Text Studio 3D control commands without changing the base package. Assign the target control, wire the XR event source, and test repeated hover, selection, disable, destroyed targets, and scene teardown.

UNITEXT

The Text Studio 3D UniText package contains sample assets only. Runtime conversion comes from the generic shaped-glyph contract in Text Studio 3D and TextStudio.Integration.UniText. Import it after Text Studio, Text Studio 3D, UniText 3, and the base Text Studio UniText integration.

The sample demonstrates shaped Latin, Arabic, and Hebrew extrusion, ligatures, fallback fonts, visual order, live typography, decorations, source colors, and Text Studio effects. UniText stroke and shadow remain flat SDF paint layers. Build physical outlines and shadows with 3D geometry, materials, and lighting.

UNITY SPLINES

Requires Unity Splines 2.0.0 or newer. The runtime SplineExtrusionPathProvider lives in the guarded Text Studio 3D Splines assembly, while the nested integration package supplies the customer sample. Use it for front-to-back extrusion sweeps, knot banking, roll, endpoint scale, and closed paths. Base Text Studio spline placement remains a separate integration that arranges glyph origins.

TROUBLESHOOTING AND REFERENCE

NOTHING APPEARS

Read Setup Status first. Confirm the base Text Studio component exists, the active text backend has a font with usable TTF or OTF outlines, at least one layer is enabled, the front face is closed when a visible cap is expected, and the current render pipeline has valid materials.

GEOMETRY LOOKS WRONG

Use Live Preview diagnostics for material regions, normals, contour direction, bounds, wireframe, and sockets. Temporarily disable depth sweep, deformers, contour programs, and decorations until the responsible stage is clear.

ANIMATION IS EXPENSIVE

Begin with the Realtime or Massive Text profile. Prefer combined or chunked output. Lower rebuild-heavy Contour Bloom and Bevel Pulse frequency before reducing ordinary transform effects. Use Runtime Statistics only while diagnosing.

ASSET AND WINDOW REFERENCE

Customer-facing assets include 3D Themes, Project Theme Settings, Control Themes, 3D Styles, Prepared Fonts, Bevel Profiles, extrusion paths, contour assets, generated contours, contour programs, tube profiles, Material Sets, spatial placements, interactions, layout modules, deformers, spawners, and module inputs. Customer-facing windows include the 3D Gallery, Style Browser, Style Compare, Material Set Browser, Live Preview, Glossary, and Font Baker.

SUPPORT

Include the Unity version, render pipeline, Text Studio and Text Studio 3D versions, active backend and font, style or layer asset, performance profile, a screenshot of Setup Status, and a minimal scene when requesting support.

ESSENTIAL SCRIPTING API

TextStudio3DText is the primary runtime facade. Use property setters for normal configuration, ApplyPreset or ApplyPerformanceProfile for a starting point, RebuildGeometry after in-place asset edits, and ScheduleGeometryBuild when several topology edits should coalesce. Unity object and mesh operations are main-thread work.

```
using TextStudio.ThreeD;

void ConfigureSign(TextStudio3DText sign)
{
    sign.ApplyPreset(BuiltInExtrudeStyles.Preset.Bevelled);
    sign.ApplyPerformanceProfile(TextStudio3DPerformanceProfile.Realtime);
    sign.AutomaticLOD = true;
    sign.RebuildGeometry();
}
```

PREWARM AND POOL

PrewarmGlyphs warms reusable geometry without changing the visible mesh and reports whether the current configuration is renderer-specific. TextStudioEmitter owns complete temporary 3D hosts, prewarms registered presets, and returns each host only after its normal Hide lifecycle. Use TextStudio3DControlPool only for repeating interactive control prefabs.

```
emitter.Prewarm();
TextStudioEmissionHandle hit =
    emitter.EmitValue(damagePreset, amount, target);
// Optional runtime update or manual dismissal:
hit.SetValue(updatedAmount);
hit.Dismiss();
```

RUNTIME OVERRIDES AND TOPOLOGY

Visibility, tint, compatible material changes, and supported visual channels can update without rebuilding topology. Changes to text, outline source, contours, bevels, body, sweep, decorations, grouped geometry, or topology modules require a build. ScheduleGeometryBuild returns one coalesced request whose state can be queried or canceled before dispatch.

GLYPH HANDLES, RANGES, RAYCASTS, AND MATERIALS

TryGetGlyphHandle resolves by visible Text Studio glyph identity and optionally authored layer. A handle becomes stale after topology changes. Use current handles or TextStudio3DGlyphRange for tint, material, force, and interaction commands. EnsureMeshCollider after topology changes before RaycastGlyph.

```
renderer.EnsureMeshCollider();
if (renderer.RaycastGlyph(ray, 50f, out TextStudio3DGlyphHandle handle,
```



```
        out MaterialRegion region, out RaycastHit hit))  
{  
    renderer.SetGlyphColorTint(handle, new Color32(255, 180, 40, 255));  
}
```

SetGlyphMaterial works on Per Letter and Physical outputs because those modes own independent renderers. Region-targeted overloads can address Front Face, bevels, sides, back, trim, or Decorations. For Combined output use color tint or material-channel effects unless the complete mesh material assignment is being changed.

INTERACTIONS, FORCES, AND SOCKETS

ApplyInteractionPreset and ResetInteraction drive the same retained state used by the no-code controller. Force commands act on current glyph identity and should be reset or contained according to the gameplay lifecycle. Named sockets provide stable attachment points without exposing generated hierarchy details.

AUTHORING ASSETS AND EXTENSION POINTS

Supported authoring contracts include Font3D, Extrude3DStyle, TextStudio3DLayer and its nine specifications, BevelProfile, contour assets and programs, tube profiles, Material Sets, extrusion path assets, module interfaces, style compilation, and explicit geometry validation. Internal generators, caches, renderer bookkeeping, pooled chunks, and the embedded polygon library are not customer APIs.

Full supported API browser: <https://www.wetzold.com/tools/text-studio/3d/api/>. Use it for complete signatures, output-specific capability notes, units, return values, cache behavior, and extension contracts.

INSPECTOR AND ASSET REFERENCE

- TextStudio3DText inspector: Setup Status, Style, Geometry Layers, Materials, Output, Quality, Interaction, Geometry Modules, Runtime Statistics, Actions, and Frozen Snapshot.
- 3D Style and Layer inspector: nine specifications, layer ordering, selection, scope, Stack or Overlay, and validation.
- Bevel Profile: profile graph, anchors, tangents, samples, snapping, mirroring, and scene handles.
- Contour Asset, Generated Contour, and Contour Program: reusable shapes and ordered boolean composition.
- Contour Tube Profile and Contour Repeats: continuous and discrete contour decorations.
- Material Set and mapping: nine semantic regions and coordinate projection.
- Spatial Placement and Extrusion Path assets: glyph arrangement versus front-to-back sweep.
- Prepared Font and Font Baker: runtime outline coverage and multilingual preparation.
- Interaction, controls, layouts, navigation, pools, deformers, spawners, and module inputs: reusable production systems.
- Gallery, Style Browser, Style Compare, Material Set Browser, Live Preview, Glossary, Font Baker, and validation samples: customer authoring windows and references.

Inspector coverage, part 1 of 5

Inspector or surface	Controls and ownership	Primary manual coverage
TextStudio3DText main inspector	Setup, style, layers, materials, output, quality, interaction, modules, statistics, actions, and freeze.	First 3D result; geometry model; animation; output
Extrude3DStyle	Shared layer stack, local detachment, order, Stack or Overlay, selection, Solo, and reset.	Geometry model; Build a Style Layer by Layer
TextStudio3DMaterialSet	Assignments for the nine semantic regions without changing geometry or mapping.	Design Materials and Surfaces
BevelProfile	Profile graph, anchors, tangents, samples, smoothing, snapping, mirroring, and handles.	Bevel Profiles and Bevel Profile Assets
QuadraticBezierExtrusion PathAsset	Start, control, end, sampling, banking, roll, taper, and closed state.	Placement Versus Extrusion Sweep
SplineExtrusionPathProvider	Container, curve index, seed up, banking, roll, scale, samples, refresh, and snapshot.	Unity Splines Extrusion; 3D Integrations
TextStudio3DContourAsset	Validated custom vector loops and contour identity.	Choose a Contour Source; SVG Import
TextStudio3DGeneratedContourAsset	Rounded rectangle, pill, plate, shield, badge, banner, and frame generation.	Generated Contours
TextStudio3DContourProgram	Input source plus ordered union, subtraction, intersection, and clipping operations.	Build a Contour Program
TextStudio3DContourTubeProfile	Reusable validated closed section for contour tubes.	Contour Tubes

Inspector coverage, part 2 of 5

Inspector or surface	Controls and ownership	Primary manual coverage
TextStudio3DSpatialPlacement	Helix, cylinder, sphere band, tunnel, staircase, spatial grid, orientation, and presets.	Arrange Text in Space
TextStudio3DInteractionController	Hover, repel, magnet, ripple, blast, gravity, target, strength, radius, reset, and motion policy.	Animate 3D Text and Add Interaction
Font3D	Prepared outline coverage, source ownership, fallback, and optional embedded data.	Prepare Fonts and Languages
Font Baker	Source, target, character set, fallback, spacing, kerning, mesh quality, and validation.	Font Baker Workflow
LayoutElementModule	Post-layout transform without rewriting source or placement.	Build Advanced Reusable Modules
DeformerModule	Ordered deformation stack, semantic space, bounds, falloff, and stable preview.	Deformer Stacks; Module Mental Model
SpawnModule	One prefab per rendered glyph, sockets, delay, lifetime, limits, and retained instances.	Glyph Prefab Spawner
Variable holder and module inputs	Designer-controlled values mapped into reusable module assets.	Module Inputs and Custom Modules
3D Gallery and Style Compare	Search, compare, apply, duplicate, detach, and validate real style recipes.	Learn From the Shipped Samples; 80-Style Gallery
Material Set Browser	Browse reusable looks and inspect region assignments before applying.	Material Sets; Learn From the Shipped Samples

Inspector coverage, part 3 of 5

Inspector or surface	Controls and ownership	Primary manual coverage
Freeze and Frozen Snapshot	Persistent static output, ownership, validation, unfreeze consequences, and player stripping.	Freeze a Finished Title
Runtime Statistics	Glyph, entry, vertex, triangle, mesh, renderer, material, cache, and rebuild state.	Runtime Statistics
Glyph Sockets	Center, front, back, top, bottom, and contour-fraction attachment poses.	Named Glyph Sockets
Module3DBase	Typed roles, stages, channels, cache impact, bounds, lifecycle, and material-region registration.	Module Mental Model; Developer Reference
Mesh export	Output ownership, real generated mesh, materials, pipeline family, and validation.	Strict Freeze; Output and Performance
TextStudio3DButton	Click behavior and normal, hover, pressed, selected, disabled, and validation states.	World-Space UI
TextStudio3DToggle	Boolean value, state visuals, callbacks, and navigation.	Buttons, Toggles, Sliders, Lists, and Selectors
TextStudio3DSlider	Range, value, step, visual fill, callbacks, and pointer or navigation input.	Buttons, Toggles, Sliders, Lists, and Selectors
TextStudio3DList	Items, selection, navigation, events, and pooled presentation.	Buttons, Toggles, Sliders, Lists, and Selectors
TextStudio3DInputField	Value, placeholder, length, validation, focus, caret, and optional keyboard path.	Input Field

Inspector coverage, part 4 of 5

Inspector or surface	Controls and ownership	Primary manual coverage
TextStudio3DSelector	Option value, previous or next input, display, and change events.	Buttons, Toggles, Sliders, Lists, and Selectors
TextStudio3DPointerRaycaster	Camera, layers, distance, hover, press, selection, and world-space ray input.	Pointer, Navigation, Layout, and Pools
TextStudio3DNavigationGroup	Ordered focus traversal and independent selected value.	Pointer, Navigation, Layout, and Pools
TextStudio3DControlLayout	Linear, grid, circular, spline, or volume placement with shared spacing and orientation.	Pointer, Navigation, Layout, and Pools
TextStudio3DControlTheme	Shared label, surface, depth, scale, and interaction-state treatment.	Control Foundation and States
TextStudio3DControlPool	Bounded prefab reuse with restored value, validation, focus, events, and framing.	Pointer, Navigation, Layout, and Pools
TextStudioEmitter	Shared temporary host emission with profiles, values, stacking, following, prewarming, lifecycle events, and bounded pooling.	Prewarming and Bounded Caches
3D Authoring Glossary	Product vocabulary for layers, contours, surfaces, paths, output, caches, and modules.	Troubleshooting and Reference
Input System integration	Action-reference mappings for pointer, navigation, submit, and text entry.	3D Integrations
XR Interaction Toolkit integration	XR hover, select, ray, and lifecycle bridge for world-space controls.	3D Integrations

Inspector coverage, part 5 of 5

Inspector or surface	Controls and ownership	Primary manual coverage
UniText 3D sample	Shaped mixed-script text converted into matching 3D outlines.	Prepare Fonts and Languages; 3D Integrations
Unity Splines integration	Base spatial placement plus guarded depth-extrusion provider.	Placement Versus Extrusion Sweep; 3D Integrations
Runtime public API	Presets, output, rebuild, prewarm, pools, overrides, handles, raycasts, regions, and sockets.	Essential Scripting API

APPENDIX: 80-STYLE GALLERY

The complete gallery is organized by visual intent. Each image below is a real rendered style preview. Use Browse Curated Styles for search, family filters, capability filters, favorites, and recent choices. Use Compare Styles to judge two candidates with identical text, font, path, and materials.

BASIC STYLES

Clean foundations that make depth, bevel, and material choices easy to read.



Clean



Bevelled



Paper Thin



Deep Chamfer



Soft Round



Crisp Slab



Micro Bevel



Embossed Face

STACKED STYLES

Layered signage, badges, title cards, posters, and dimensional trim.



Stacked



Metal Plate



Golden Signage



Outline Trim



Inset Badge



Neon Glow



Double Chamfer



Groove Outline



Letterman Varsity



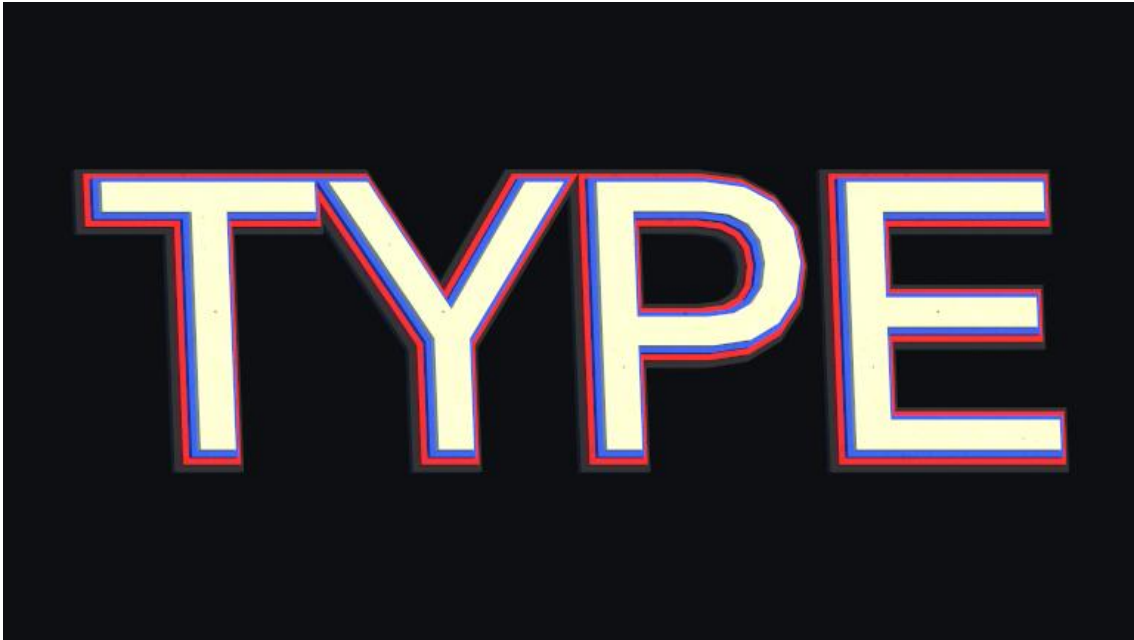
Comic Book



Foil Stamp



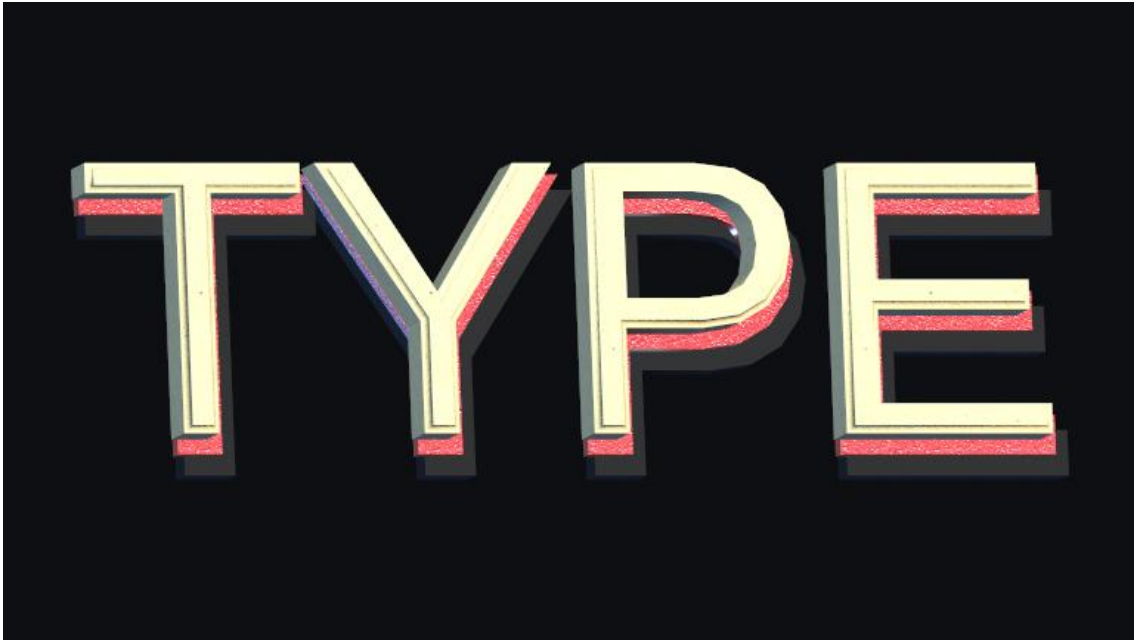
Nameplate Brass



Paper Cutout



Title Card Hero



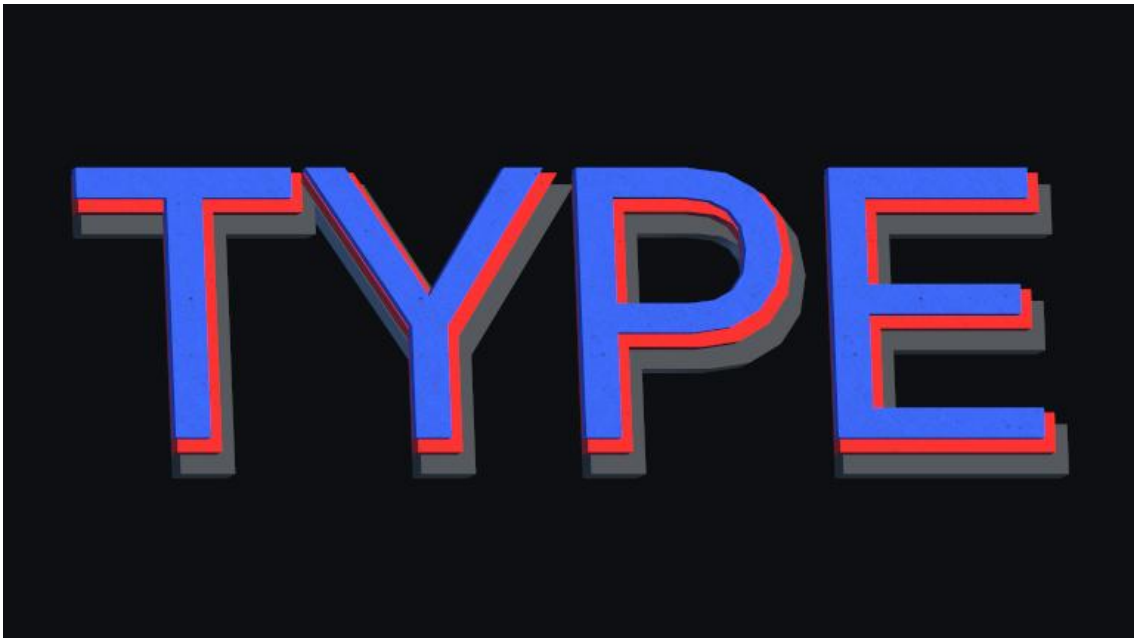
Retro Shadow



Enamel Pin



Acrylic Edge



Block Poster



Tech Panel

PROFILE STYLES

Architectural edge profiles for refined lettering and decorative molding.



Ogee Molding



Cove Groove



Scotia Trim



Bullnose Edge



Knife Edge



Art Deco



Waterfall Edge



Reverse Ogee



Crown Molding



Stepped Bead



Rolled Edge

CHISEL STYLES

Carved, engraved, stone, woodcut, and metallic face treatments.



Chiseled Stone



Engraved Plaque



Carved Glyph



Gold Chisel



Stone Inset



Woodcut Ridge

INFLATE STYLES

Rounded, soft, candy, balloon, bubble, and gel-like volume.



Inflated Candy



Pillow Puff



Gummy Bear



Foil Balloon



Soft Bubble



Gel Capsule

DEPTH STYLES

Strong silhouettes shaped by taper, twist, columns, and depth deformation.



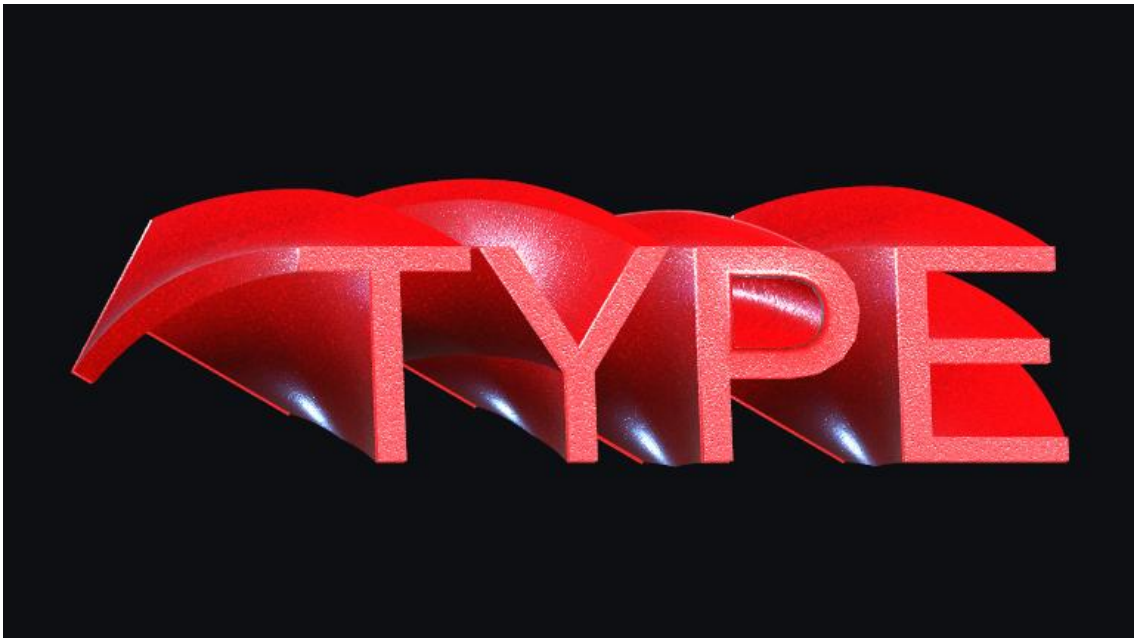
Stepped Pyramid



Keystone



Twist Column



Barley Candy



Crystal



Pillar Column



Fan Twist



Barrel Taper



Hourglass

MATERIAL STYLES

Surface-led styles for glass, metal, enamel, patina, neon, and acrylic.



Glass Frosted



Brushed Chrome



Enameled Badge



Copper Patina



LED Sign



Crystal Acrylic



Black Chrome



Holographic Glass

MOTION STYLES

Broadcast and motion-graphics layouts designed to pair with animated Text Studio effects.



Motion Graphic Badge



Lower Third Type



Stadium Curve



Broadcast Key



Sports Bug

OUTLINE STYLES

True contour outlines and layered strokes for readable signage.



True Outline



Double Stroke



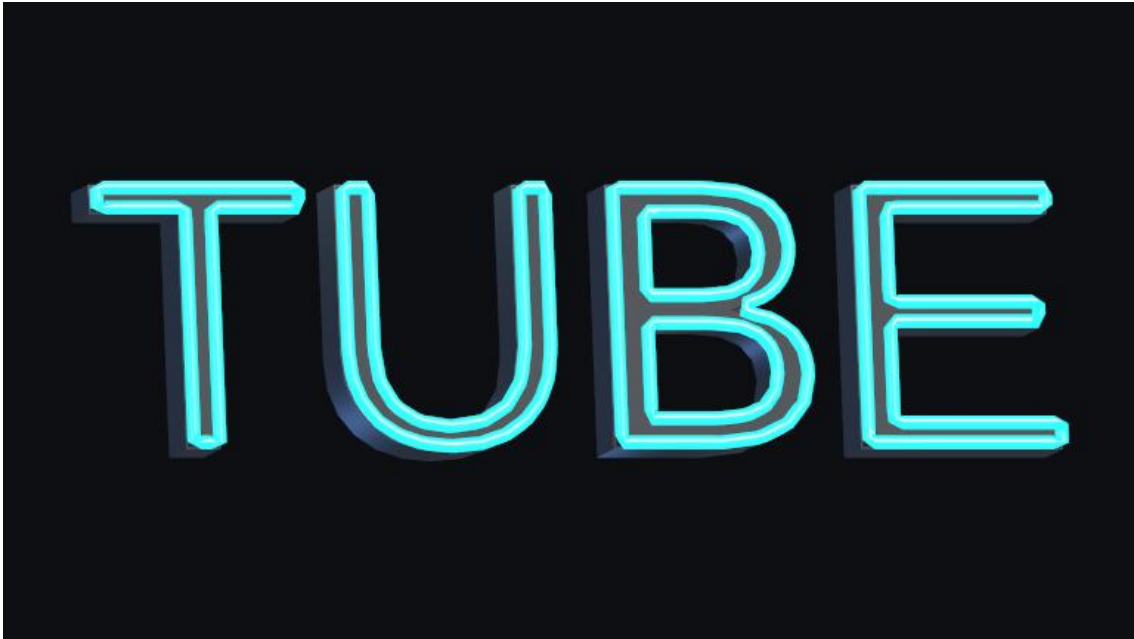
Neon Outline



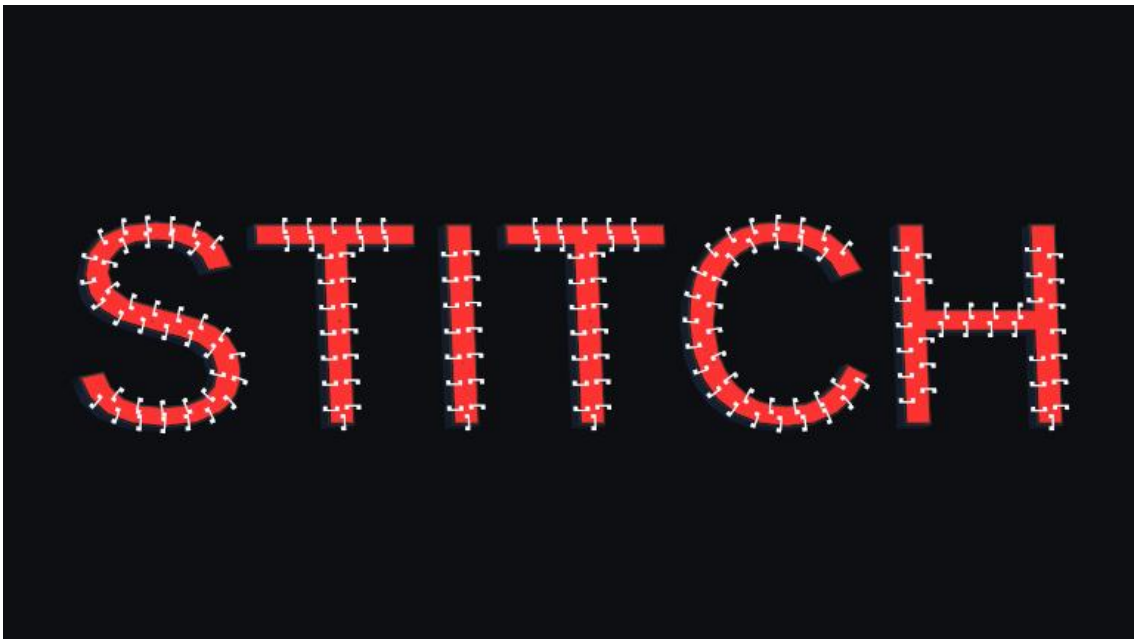
Tri Stroke

DECORATION STYLES

Contour-driven lights, stitches, tubes, and riveted plates.



Neon Tube



Stitched



Marquee Bulbs



Riveted Plate

APPENDIX: NATIVE 3D EFFECT GALLERY

NEON CHASE

Canonical tag: <neonChase>. Key controls and defaults: strength (3.2), speed (0.75), spread (1.35), softness (5), color (cyan), reduced motion.

A travelling emissive chase for signage and arcade titles.



Neon Chase

CHROME SWEEP

Canonical tag: <chromeSweep>. Key controls and defaults: speed (0.45), spread (1.1), smoothness (0.92), highlight (0.7), reduced motion.

A polished reflection-like sweep across metallic surfaces.



Chrome Sweep

HOLOGRAM SCAN

Canonical tag: <hologramScan>. Key controls and defaults: strength (2), speed (0.9), spread (1.8), shimmer (0.035), color (cool cyan), reduced motion.

A scan-line material treatment for holographic and sci-fi text.



Hologram Scan

FACET PULSE

Canonical tag: <facetPulse>. Key controls and defaults: strength (2.4), speed (0.7), spread (0.8), surface mask (Edges), reduced motion.

A rhythmic response across faceted surfaces.



Facet Pulse

GLASS REVEAL

Canonical tag: <glassReveal>. Key controls and defaults: glow (1.5), softness (0.18), reduced motion.

A transparent material reveal for acrylic and glass styles.



Glass Reveal

DEPTH WAVE

Canonical tag: <depthWave>. Key controls and defaults: strength (0.35), speed (0.7), spread (0.65), surface mask (All), reduced motion.

A travelling response through the extrusion depth.



Depth Wave

CONTOUR BLOOM

Canonical tag: <contourBloom>. Key controls and defaults: amount (0.035), speed (0.55), spread (0.42), reduced motion.

A topology-aware contour expansion for dimensional accents.



Contour Bloom

BEVEL PULSE

Canonical tag: <bevelPulse>. Key controls and defaults: strength (0.45), speed (0.6), spread (0.5), reduced motion.

A topology-aware pulse focused on bevel geometry.



Bevel Pulse