

Documentation V17 2023

Mesh Deformation Full Collection

[Official Website](#)

[API documentation](#)

Last package overview

Last package video-documentation



Older package video-resources

Package Documentations



Essentials, Setup in General, VR Setup



Sculpting, Realtime mesh editor, quick 'how-to'



Package introduction, essentials, modifiers, shaders...

*Full package-related playlist
can be found [here](#)*

Package Overviews



[Online resources don't have to be the same version, but they are related to the actual version & still might be helpful]

Change Log

17th Update 2023 [MAJOR - 19/01/2023; dd/mm/yyyy]

- MAJOR update for package API > I've been working on custom API Docs To HTML convertor that would help me to convert any code-API to the readable HTML database.
- Major refactor of the code
- New editor icons
- Category 'Shapes' renamed to 'Geometry'
- Major upgrade related to abstraction & OOP
- New base classes for each category: MD_Mesh Base, MD_GeometryBase, MD_ModifierBase
- New collection of primitive geometry (procedural triangle/pyramid, cube and more)
- Major cleanup of all modifiers (naming, methods, static methods)
- Major cleanup of all utilities (naming, static methods)
- Upgraded editor experience (undo issues, warnings cleanup and more)
- Minor VR cleanup (naming, namespaces & latest VR framework refresh)
- Ready for Unity 2020, 2021, 2022 & 2023a.



Roadmap

Official roadmap of the package and all of my assets



Curious what the future holds? Click the Trello icon!

Raymarcher

RAYMARCHER
Universal Renderer

Raymarcher

Add depth to your project with Raymarcher asset from Matej Vanco. Find this & more...

UnityAssetStore

Sculpting Pro

SCULPTING PRO

Sculpting Pro

Get the Sculpting Pro package from Matej Vanco and speed up your game...

UnityAssetStore

MD Package

Mesh Deformation Full Collection

Mesh Deformation Full Collection

Get the Mesh Deformation Full Collection package from Matej Vanco and speed up...

UnityAssetStore

Mesh Tracker

MESH TRACKER

Mesh Tracker

Add depth to your next project with Mesh Tracker from Matej Vanco. Find this & more...

UnityAssetStore

Color Picker

Color Picker

Color Picker Plugin

Use the Color Picker Plugin from Matej Vanco on your next project. Find this GUI...

UnityAssetStore

Save-It

SAVE-IT PRO

SAVE-IT

Get the SAVE-IT Pro package from Matej Vanco and speed up your game...

UnityAssetStore

Content

Basics

[Introduction](#)

[Summary](#)

Essentials

[Mesh Pro Editor](#)

[Mesh Editor Runtime](#)

[Mesh Editor Runtime VR](#)

[Mesh Collider Refresher](#)

[Global Preferences](#)

Modifiers

[MDM_Bend](#)

[MDM_Twist](#)

[MDM_MeshNoise](#)

[MDM_FFD](#)

[MDM_MeshEffect](#)

[MDM_MeshDamage](#)

[MDM_Morpher](#)

[MDM_InteractiveSurface](#)

[MDM_SurfaceTracking](#)

[MDM_MeshFit](#)

[MDM_MeshSlime](#)

[MDM_MeltController](#)

[MDM_SculptingLite](#)

[MDM_RaycastEvent](#)

[MDM_MeshCut](#)

[MDM_SoundReact](#)

Shaders

[Standard Deformer](#)

[Lite Mesh Tracker](#)

[Melt Shader](#)

Geometry

[MDG_ProceduralExtendedPlane](#)

[MDG_HexagonGrid](#)

[MDG_MeshPaint](#)

[MDG_PathCreator](#)

[MDG_TunnelCreator](#)

[Other Primitives](#)

Technical Info

[VR Setup](#)

[Multithreading](#)

[Vertex Tool Window](#)

[Performance](#)

[FAQ](#)

[Commercial Products](#)

[Downloadable Content](#)

[Extras](#)

[Contact & Discord](#)

Basics

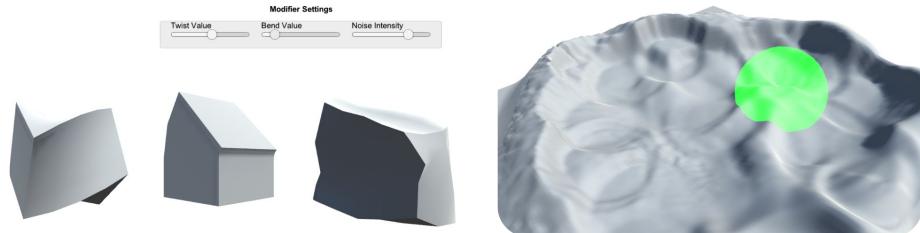
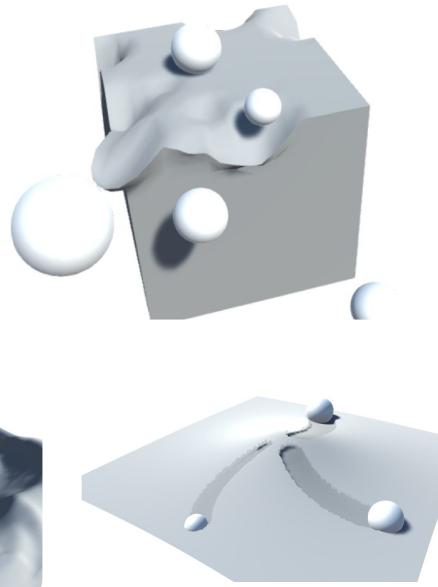
Basic information you should know

Introduction
Summary

Introduction

- **MD [Mesh Deformation]** is a collection of mesh manipulation in Unity Engine for beginners & advanced users. The package contains primitive vertex editor (PC, VR, Mobile), collection of various well-known modifiers, mesh collider refresher, shader-based deformation, physically based procedurals, geometry primitives and many more. MD Package contains many examples with details, source code, description & realtime support.
- The package has began to be developed in 2013 by me, Matej Vančo. It started very simply with very basic features such as generating generic points for mesh, basic controls of Skinned Mesh bones and so on. The package was officially published on 12th of November 2015.
- Back in time the ‘mesh deformation’ was not very common in Unity as the Unity is not a modelling software. This package contains various systems with universal & modular functionality. It’s a collection of mesh deformations in Unity based on well-known modifiers & mesh features tested in many commercial projects by many users for years now.

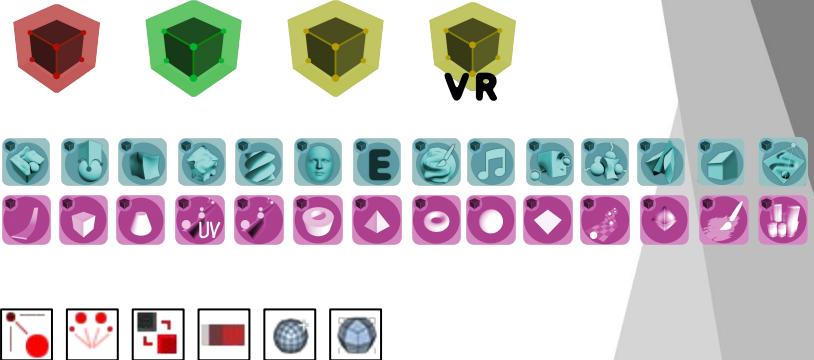
In this documentation, you will get familiar with all essential components, modifiers, shaders & additional tips and tricks about performance and overall use of the package.



Summary

The MD Package is divided into **5 categories**:

- **Essentials**
 - Contains essential components such as *MeshProEditor*, *MeshEditorRuntime+VR* & *MeshColliderRefresher*.
- **Modifiers**
 - Contains all modifiers such as *Mesh Damage*, *Sculpting*, *Interactive Surface*, *FFD* etc.
- **Shaders**
 - Contains a shader source called *Standard Deformer*
- **Geometry**
 - Contains all the geometry such as procedural primitives, *Tunnel Creator*, *Path Creator*, *Mesh Paint* etc.
- **Utilities**
 - Contains additional utilities such as *Mesh Smooth*, *Vertex Tools*, *Package Utilities* etc.



Each category has its own folder with all the required components. If you would like to use the modifiers only, you can just drag and drop the modifier onto any object with *MeshFilter* component. If you would like to generate procedural geometry, you can access it in *Hierarchy/Create*. More about each category in the next slides.

Official API documentation of the MD Package can be found [here](#).

Essentials

First category with essential components

Full description & API

Editor-tooltips available

Global namespace = `MD_Package`

[Mesh Pro Editor](#)

[Mesh Editor Runtime](#)

[Mesh Editor Runtime VR](#)

[Mesh Collider Refresher](#)

[Global Preferences](#)

Mesh Pro Editor

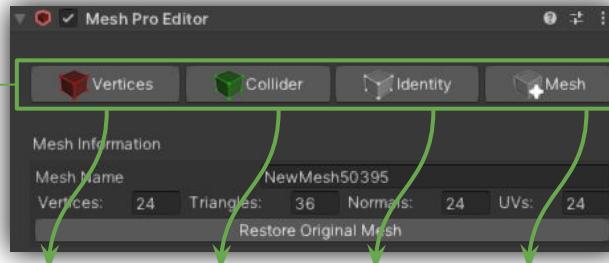


Mesh Pro Editor is one of the essential components that allows you to analyze certain mesh, edit it's vertices & it's features with a few external modifiers such as 'smooth mesh' & 'mesh subtraction'. The component is divided into **4 parts**: **Vertice Modification**, **Collider Modification**, **Identity Modification** & **Mesh Modification**. On the bottom part of the component is **Mesh Information** panel which contains essential data of the mesh (vertex count, tris etc). The component must be applied to objects that contain **Mesh Filter** or **Skinned Mesh Renderer** (It will be compiled to Mesh Filter either). It's recommended to use the Mesh Pro Editor first before applying any other modifier. Mesh Pro Editor will ask you for 'creating a new mesh reference' and it's basically much safer while editing certain meshes. The component also contains a public API which can be used externally.

APi documentation



Click to
enable/disable
selection



Vertice
Modification

Collider
Modification

Identity
Modification

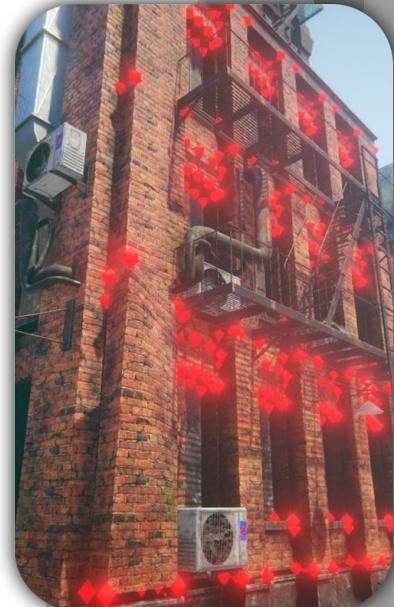
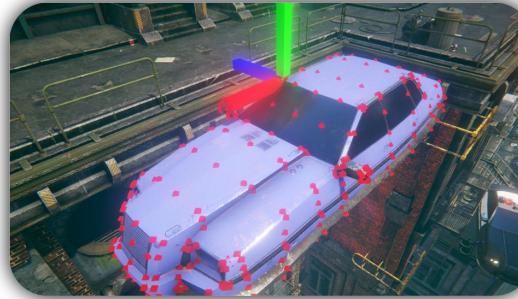
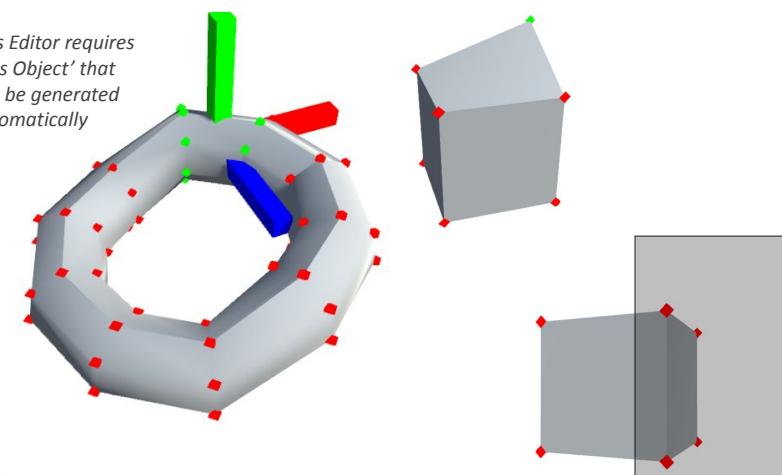
Mesh
Modification

Mesh Editor Runtime



Mesh Editor Runtime allows you to edit generated points (by Mesh Pro Editor) at runtime by specified mouse input (PC) or “finger” input (Mobile). The component contains two types of runtime editor modes: **Axis Editor & Non Axis Editor**. The main difference is that the Axis Editor requires specific object with XYZ generics which will represent the well-known “axis arrows”, and Non Axis Editor allows you to drag & drop vertices. Both editor modes do the same job, but each has a different user-interaction and controls. The component is usually added to the main camera in a scene and it's specially built for designers without programming skills. It's also possible to choose between three editor-control modes (Grab/Drop Vertex, Pull Vertex or Push Vertex).

APi documentation



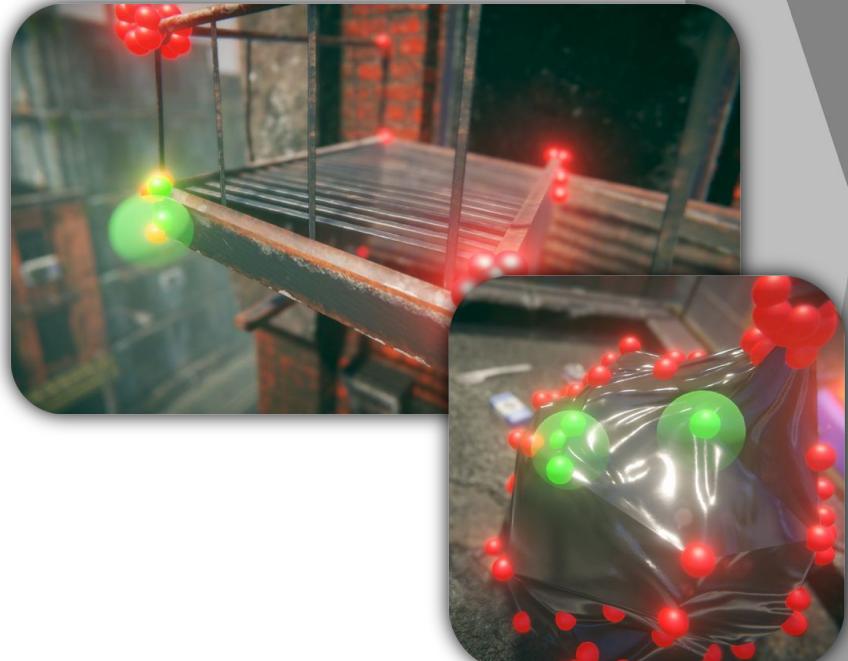
Mesh Editor Runtime VR



[APi documentation](#)

Mesh Editor Runtime VR allows you to edit generated points (by Mesh Pro Editor) at runtime by specified VR input. Just like the Non-VR Mesh Editor Runtime, it does the same job. All the major VR platforms are fully supported including **SteamVR**, **Oculus Integration** and **Unity-XR**. It's required to choose & export proper package to prevent additional errors. It's also required to have properly installed **3rd package**

libraries (e.g.: If you are going to use the Mesh Editor Runtime VR for SteamVR, you should have installed SteamVR package from the Assets Store). The component should be added to one of the VR controllers. The overall VR setup for Mesh Editor Runtime VR is demonstrated in the [VR Setup](#) slide.

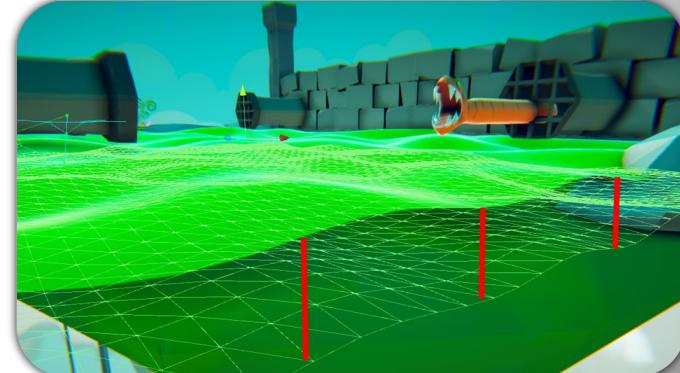
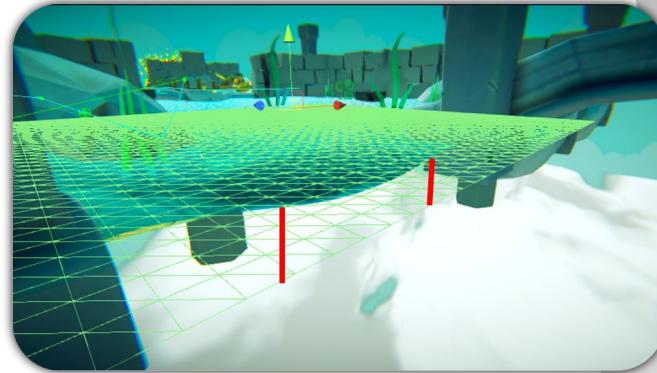
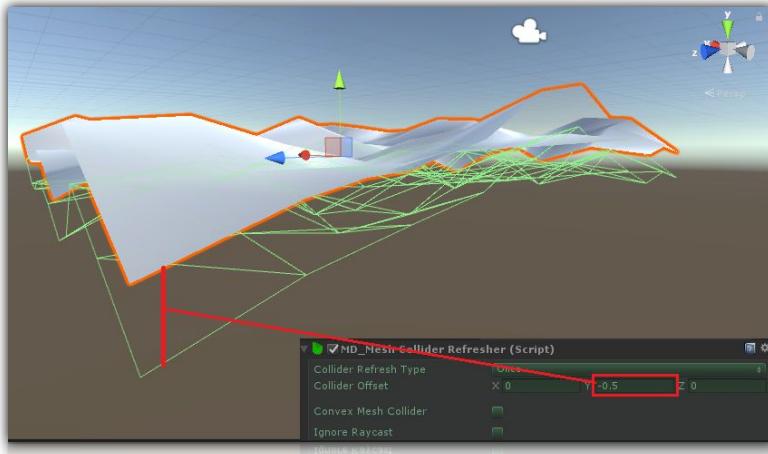


Mesh Collider Refresher



[APi documentation](#)

Mesh Collider Refresher allows you to update & refresh mesh collider at runtime. You can choose many modes such as refresh mesh collider once, every frame or by custom interval. You are also able to update the mesh collider manually or by the specific event action. In the refresh type 'once' you can set up your own mesh collider position offset. Simple example is described below.

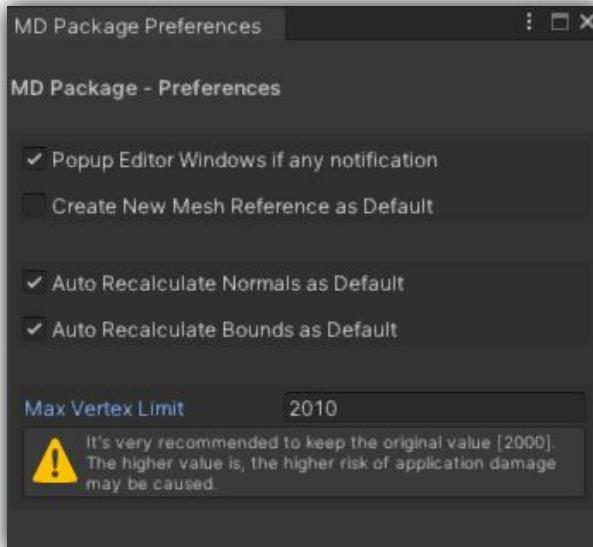


Preferences

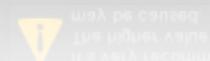
API documentation

The **MD Package** contains a global preference window that helps you to manage certain tasks & features of the package. Preferences window can be found in **Window/MD_Package/Preferences**

The following image is an actual preferences window.



- **Popup editor window** if required with any component related to the MD Package (For example mesh is beyond the safe level of vertex count to edit)
- **Create new mesh reference** as default (Otherwise the original mesh data will remain = all modifiers create new references automatically)
- **Auto recalculate Normals & Bounds** as default (Otherwise the mesh will remain it's original normal & bounds data = all modifiers will recalculate bounds & normals automatically if enabled)
- Allowed **Max Vertex Limit** field. If mesh has more vertices than the specified value, the warning window will popup (if popup editor window is enabled for sure!)



Modifiers

Second category with modifiers

Full description & API

Editor-tooltips available

Global namespace = **MD_Package.Modifiers**

[MDM_Bend](#)

[MDM_Twist](#)

[MDM_MeshNoise](#)

[MDM_FFD](#)

[MDM_MeshEffect](#)

[MDM_MeshDamage](#)

[MDM_Morpher](#)

[MDM_InteractiveSurface](#)

[MDM_SurfaceTracking](#)

[MDM_MeshFit](#)

[MDM_MeshSlime](#)

[MDM_MeltController](#)

[MDM_SculptingLite](#)

[MDM_RaycastEvent](#)

[MDM_MeshCut](#)

[MDM_SoundReact](#)

MDM_Bend,Twist,Noise



MDM (MeshDeformationModifier) Bend, Twist & Noise are one of the basic modifiers in the package. Each basic modifier deforms meshes in a unique way: Bend bends the mesh in 3 axes, Twist twists the mesh in 3 axes and Noise populates mesh & creates 'dirty' surface which can be modified in two modes - vertical and general deformation. General noise controls the whole mesh in a 3d space while vertical noise controls the mesh in a 2d space - the mesh will be planar. All the basic modifiers run in the editor & at runtime. All the parameters can be changed at runtime as well.

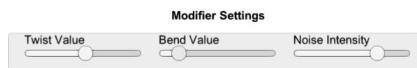
All modifiers can be added to the object just once. Also it's **prohibited** to use **multiple modifiers at once** on the **same object** due to the vertex sync.

Bend, Twist & Noise are not multithreaded modifiers, so applying to hi-poly objects (2000 vertex count and more) is not much recommended.

[API documentation \(Twist\)](#)

[API documentation \(Bend\)](#)

[API documentation \(Noise\)](#)



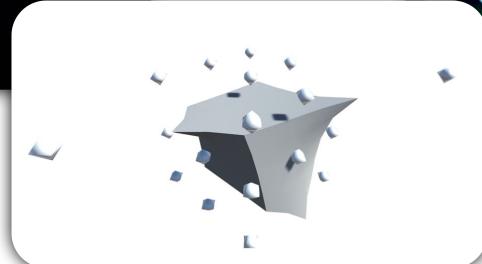
MDM_FFD



API documentation

MDM (MeshDeformationModifier) FFD stands for Free-Form-Deformation which controls mesh by the registered weights in lattice. The mesh will smoothly deform by moving points in a lattice. It's a great modifier to quickly edit desired mesh parts. The FFD modifier contains various lattice resolutions in cubic-shape: 2x2x2, 3x3x3, 4x4x4 and custom number (not recommended due to the performance).

FFD is not a multithreaded modifier, but it's safe to apply it on hi-poly objects (2000 vertex count and more) as the modifier is not heavy-to-calculate.



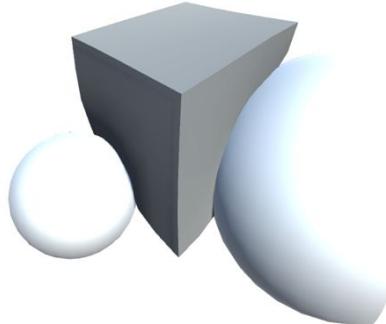
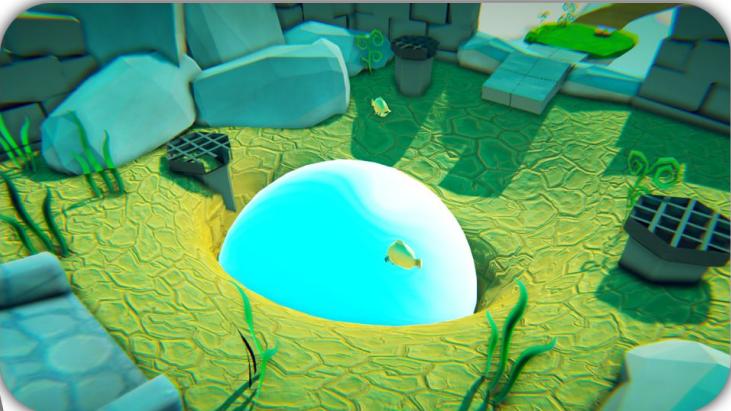
MDM_MeshEffect



[APi documentation](#)

MDM (MeshDeformationModifier) Mesh Effector deforms mesh by registered weight nodes & density. There are four types of effectors: One-Pointed, Two-Pointed, Three-Pointed and Four-Pointed. Effectors create spherical radiiuses to the assigned nodes that affect the target mesh. In comparison with the FFD, the Mesh Effector doesn't create lattice, instead it creates radiiuses which deform the overall mesh by several parameters such as Weight value, Weight multiplier, Weight density and Weight effector for each 'effector-type'. It's a great modifier for quick deformations without any further hard work!

Mesh Effector is a multithreaded modifier. It's safe to edit hi-poly objects (2000 vertex count and more).



MDM_MeshDamage



[API documentation](#)

MDM (MeshDeformationModifier) Mesh Damage allows you to deform any mesh surface with any external source such as rigidbody collision impact, raycast event etc. Simple & very effective modifier that makes your mesh more interactive with your environment!

Mesh Damage is not a multithreaded modifier, but it's safe to apply it on hi-poly objects (2000 vertex count and more) as the modifier is not heavy-to-calculate.



MDM_Morpher



[API documentation](#)

MDM (MeshDeformationModifier) Mesh Morpher allows you to blend between various shapes. It's possible to choose unlimited shape count and register their initial vertex state. Morpher is well-known in face expressions, mimics, statue translations and so on.

Mesh Morpher is a multithreaded modifier. It's safe to edit hi-poly objects (2000 vertex count and more).

Example: Blend between 3 meshes



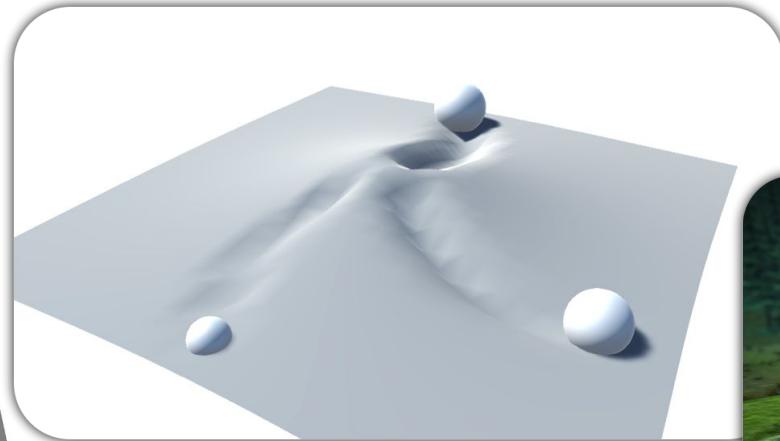
MDM_InteractiveSurface



[APi documentation](#)

MDM (MeshDeformationModifier) Interactive Surface allows you to interact with any meshes and simulate 'surface deformation'. Create dynamic tracks, snow trails, drops and more. The mesh can be 'regenerated' as well by specified speed value and interpolation.

Interactive Surface is a multithreaded modifier. It's safe to edit hi-poly objects (2000 vertex count and more).



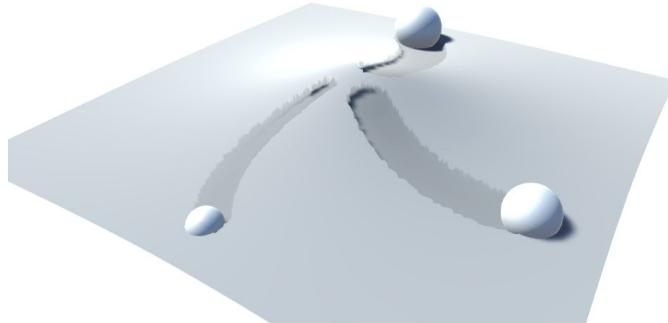
MDM_SurfaceTracking



[API documentation](#)

MDM (MeshDeformationModifier) Surface Tracking does the same job as Interactive Surface, but requires extra material and extra settings to setup. Surface Tracking runs mostly via **GPU** which saves more performance and it's easier to play around. The modifier requires **MD_LiteMeshTracker** shader and Render Texture, which can be generated automatically in the Inspector. If you would like to use the Surface Tracking for mobile, use **MD_LiteMeshTrackerMobile** shader. Great modifier for quick surface simulations. In comparison with Interactive surface - the results are less smooth, depends on Tessellation value.

Surface Tracking is not a multithreaded modifier, but it's safe to apply it on hi-poly objects (2000 vertex count and more) as the modifier is not heavy-to-calculate.



MDM_MeshFit



MDM (MeshDeformationModifier) Mesh Fit allows you to fit mesh to any surface with collider by generated or specified points. The modifier simply helps you to create universal decals and dynamic meshes with raycast events. This is not based on a projector component.

Mesh Fit is not a multithreaded modifier. Editing hi-poly objects (2000 vertex count and more) is not much recommended.

APi documentation



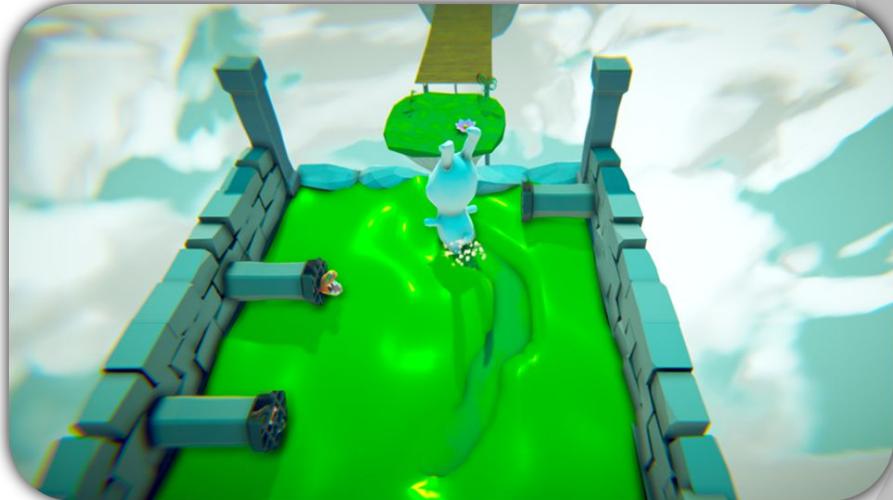
MDM_MeshSlime



[APi documentation](#)

MDM (MeshDeformationModifier) Mesh Slime allows you to create 'slime-like' surfaces with additional input. It basically does the same job as the Interactive Surface, but with more settings & in more deeper job.

Mesh Slime is not a multithreaded modifier. Editing hi-poly objects (2000 vertex count and more) is not much recommended.

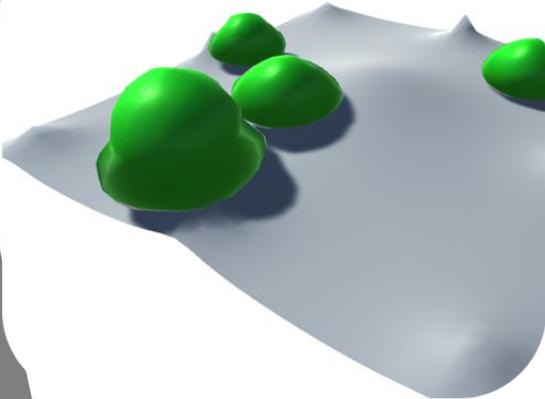


MDM_MeltController



[APi documentation](#)

MDM (MeshDeformationModifier) Melt Controller allows you to create a simple melting effect. The controller requires a special material with **Melting shader** which is included in the package. Setup melting zone, melt transition, melt amplification and more. This modifier does not work with HDRP or URP.



MDM_SculptingLite



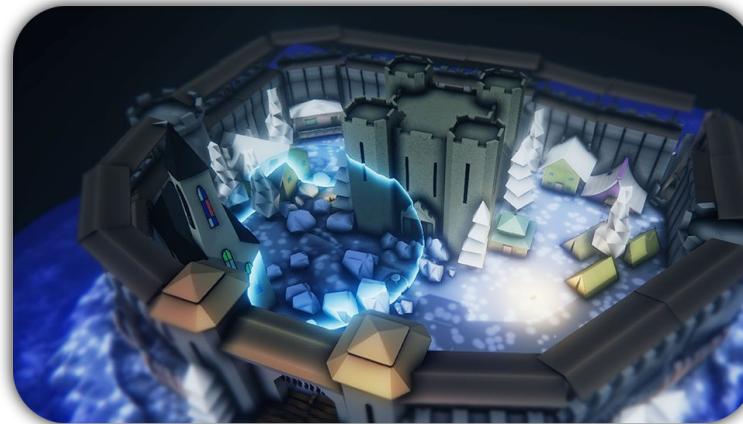
MDM (MeshDeformationModifier) Sculpting Lite allows you to deform any mesh in different & more-artistic way. The system opens a new perspective of mesh manipulation by radial-brush with various features. Brush type and brush appearance is fully customizable with brush radius & intensity. It's also possible to use an internal API which allows you to modify your own methods. The system is great for advanced terrain editing (Not built-in unity terrain) with various brush types like **smooth** (HCFilter, Laplacian Filter), **noise** filter, **stylization** and more. Sculpting Lite is compatible for Mobile and VR as well and it's fully ready for extremely complex meshes thanks to multithreading.

The development of this mesh-tool has ended. Please visit a brand new plugin called [Sculpting Pro](#) - the complete sculpting solution in Unity.

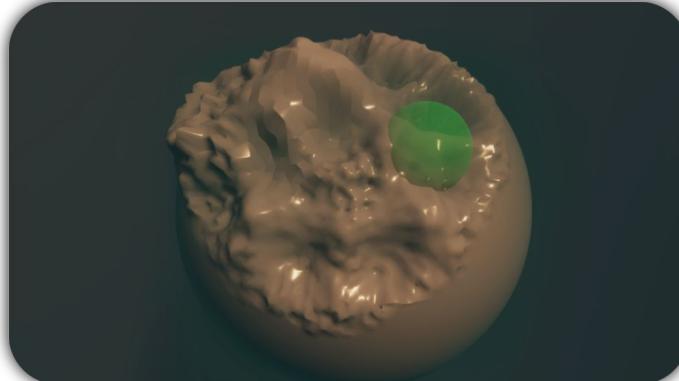
Sculpting Lite is a multithreaded modifier, so it's safe to edit hi-poly objects (2000 vertex count and more). Tested on even much detailed meshes (50k vertex count and more).

APi documentation

Used in a PC & VR game *The God*



Oculus Quest 2 with hand tracking



MDM_RaycastEvent



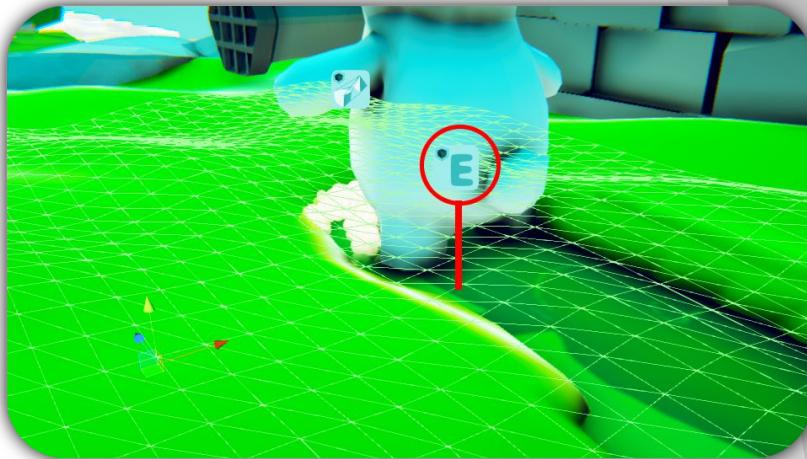
[APi documentation](#)

MDM (MeshDeformationModifier) Raycast Event is a simple modifier to render certain events if raycast hits something. You can set up your own event system & raycast logic without any programming skills. The raycast event is connected with almost all the modifiers, so you can access their methods with Raycast Event.

Used as a 'step-tracker' for Surface Tracking



Blabibo tracks its position for Mesh Slime



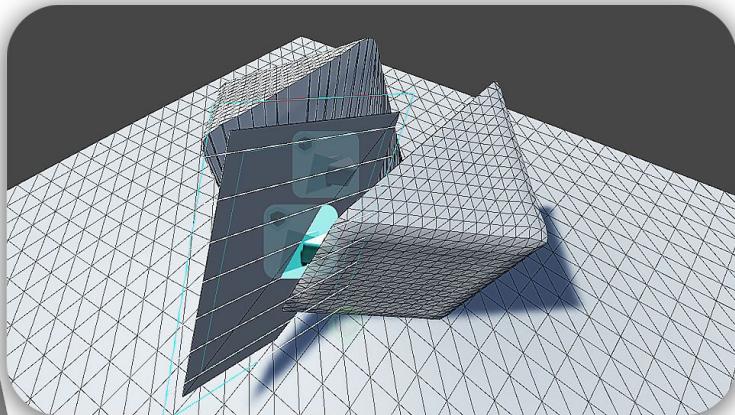
MDM_MeshCut



[APi documentation](#)

MDM (MeshDeformationModifier) Mesh Cut is a simple modifier for one-directional mesh cutting. It is possible to process a cut operation manually (via script) or you don't have to even think about the code and you can simply use the physically-based solution called [MeshCut_Cutter](#) that makes all the job for you.

Mesh Cut is not a multithreaded modifier, but it's safe to apply it on hi-poly objects (2000 vertex count and more) as the modifier is not heavy-to-calculate.



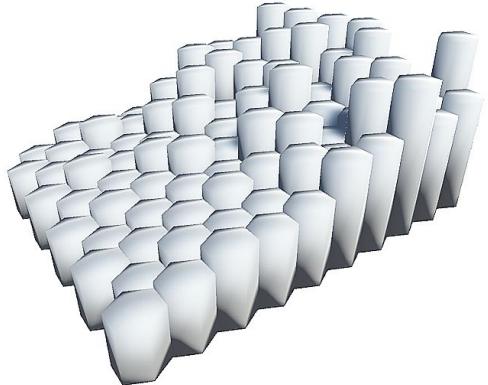
MDM_SoundReact



[APi documentation](#)

MDM (MeshDeformationModifier) Sound React allows you to convert any audio-clip data to a single floating value. The value can be later used in any case, for any musical-audio visualizations. Access to **an output variable** in the **SoundReact** script and use the value to any kind of magic!

Used in a hexagon grid - grid dances to the music!



Shaders

Third category with shader source

[Full description & API](#)

[Standard Deformer](#)

[Lite Mesh Tracker](#)

[Melt Shader](#)

Standard Deformer

Standard Deformer is an essential shader in the MD Package. The shader allows you to simply animate any object with renderer component in various ways. Create motion with many possibilities such as jumping, fish swimming, simple sway to all directions, noise simulation and so on. There is a game called [Sea Orchestra](#) which was made just from the Standard Deformer. Check the right panel with all the available settings and features of the Standard Deformer.

The Standard Deformer doesn't work with HDRP or URP.



Essentials

Cull: Back
Main Color: Albedo (RGB) Texture
Tiling: X 1 Y 1
Offset: X 0 Y 0
Normal Texture
Tiling: X 1 Y 1
Offset: X 0 Y 0
Normal Power: 0.5
Specular: 0.5
Metallic Texture
Tiling: X 1 Y 1
Offset: X 0 Y 0
Metallic Power: 0
Emission Texture
Tiling: X 1 Y 1
Offset: X 0 Y 0
Emission Color: HDR

Deformers

Deformer Animation Type: Jump
Deformer Direction: X 0 Y 0.5 Z 0 W 1
Deformer Frequency: 4.1
Edges Multiplier: 0
Additional Edges: X 0 Y 0 Z 0 W 0
Overall Extrusion: 0

Deformer Additional Properties

Absolute Value:
Frac Value:
Frac Value Frequency: 0

Clipping

Enable Clipping:

Noise

Enable Noise:

Lite Mesh Tracker

Lite Mesh Tracker (Previous: EasyMeshTracker) is a required shader for using **Surface Tracking** modifier. The modifier works with Render Textures which tell the shader how high or how low the mesh should be. It's cheap and quick, so it's easy to set up! For using the Lite Mesh Tracker for mobile, use **Lite Mesh Tracker_Mobile**. Also the mobile version doesn't support Tessellation - you can use Procedural Plane to set custom vertice count.

The Lite Mesh Tracker doesn't work with HDRP or URP.



MD_Examples_SurfaceTrackMat
Shader Matej Vanco/Mesh Deformation Package/MD_Easy

Use SurfaceTracking modifier for advanced settings

Upper Color

Lower Color

Albedo (RGB) Texture

Tiling X 1 Y 1
Offset X 0 Y 0

Select None (Texture)

Normal Texture

Tiling X 1 Y 1
Offset X 0 Y 0

Select None (Texture)

Normal Power 0.5

Specular -0.48

Emission Intensity 0

Track Settings

Track Depth -0.08

Tessellation Settings

Tessellation 4.1

Displacement Track

Tiling X 1 Y 1
Offset X 0 Y 0

Select 4.1

Min Distance 20

Max Distance 50

Render Queue From Shader ▾ 2000

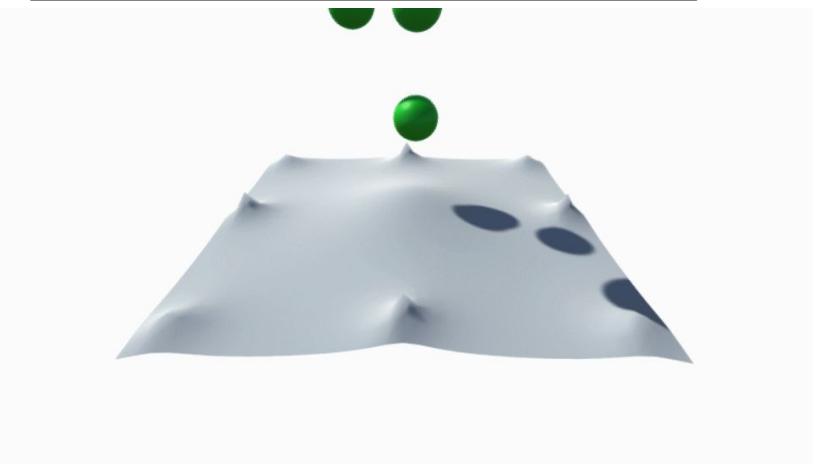
Enable GPU Instancing

Double Sided Global Illumination

Melt Shader

Melt Shader is a required shader for using **Melt Controller**, however the shader can be used outside the Melt Controller modifier. The Melt Shader allows you to create 'melting' effect on any renderer object with various settings. Melt shader is compatible with all platforms.

The Melt Shader doesn't work with HDRP or URP.



Use Melt modifier for advanced settings

Color

Albedo (RGB) Texture

Tiling X 1 Y 1

Offset X 0 Y 0

Normal Texture

Tiling X 1 Y 1

Offset X 0 Y 0

Normal Power

Specular

Emission Intensity

Noise Settings

Noise Multiplier

Noise Speed

Noise Blend

0.01

0.56

0

0.3

10

0.05

Melt Settings

Melt Transition

Melt Zone

Melt Start

Melt Amount

Melt Amount Multiplier

0.84

0.68

0.2

0.58

1

Render Queue

From Shader ▾ 2000

Enable GPU Instancing

Double Sided Global Illumination

Geometry

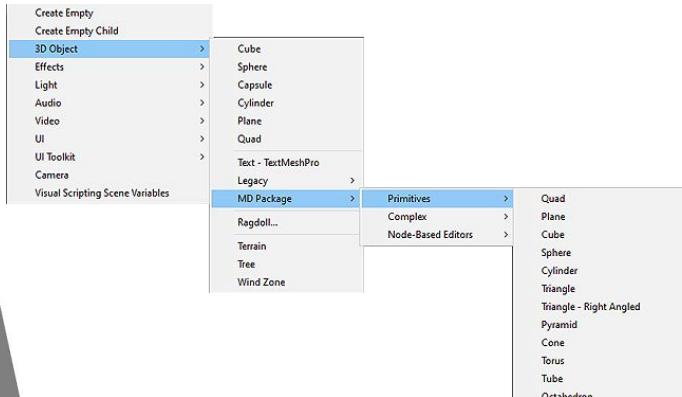
Fourth category with geometry primitives & generators

Full description & API

Editor-tooltips available

Global namespace = `MD_Package.Geometry`

Geometry can be found in Hierarchy/Create



[MDG_ProceduralExtendedPlane](#)

[MDG_HexagonGrid](#)

[MDG_MeshPaint](#)

[MDG_PathCreator](#)

[MDG_TunnelCreator](#)

[Other Primitives](#)

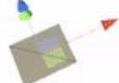
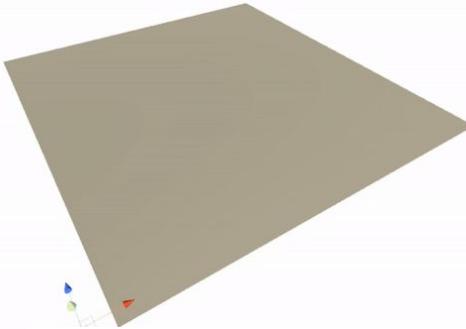
MDG_ProceduralExtendedPlane



[API documentation](#)

Procedural Extended Plane is an advanced-shape component that allows you to generate procedural plane with custom vertex count with sharp normals. The main feature of the plane is that the generated plane has no smoothing, so the final results would be 'sharp-looking'. There are additional features called 'Angle Property' which allows you to 'bend' the plane up or down.

Noise modifier applied to the plane

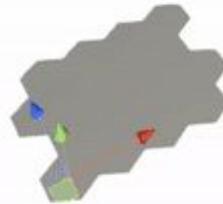


MDG_HexagonGrid



[API documentation](#)

Hexagon Grid allows you to generate procedural hexagon grid. Fully customizable hexagon grid with random height feature, that allows you to randomize each hexagon segments height. It's also possible to flip hexagon faces, make them planar or even change the overall size.



MDG_MeshPaint



[API documentation](#)

Mesh Paint is a simple & complete mesh painting solution that allows you to paint mesh in three shapes: Plane, Triangle & Cube. The component is fully ready for designers with advanced settings of planar drawing or spatial drawing. The component is available for all platforms including VR & mobile.

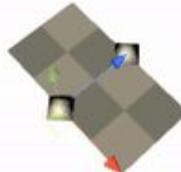
MDG_PathCreator



[API documentation](#)

Path Creator is a complete path-creation tool that allows you to create & edit procedural paths with triplanar/regular UV mapping. Create simple paths with user-friendly nodes, edit tracks and apply further modifiers for smoothing features.

Path Creator allows you to create custom paths by creating modular nodes or you can generate path along the existing nodes.

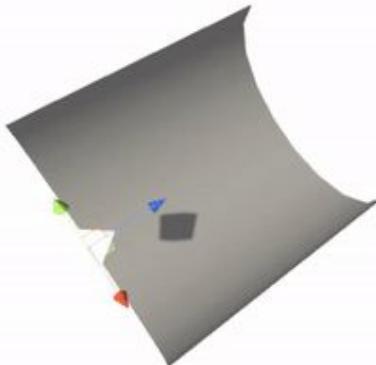


MDG_TunnelCreator



[API documentation](#)

Tunnel Creator is a complete tunnel-creation tool that allows you to create & edit procedural tunnels with triplanar UV mappings. The system is based on modular nodes and weights that can be modified in editor/ at runtime. There are additional features that allow you to make a turn, straight line or connect other ‘tunnels’. Tunnel Creator also contains its own editor window for better overview and workflow. There are two 2 important components related to MDG_TunnelCreator: **MDG_TunnelCreator** (the root) and **MDG_TunnelNodeUVData** (the node uv data controller).



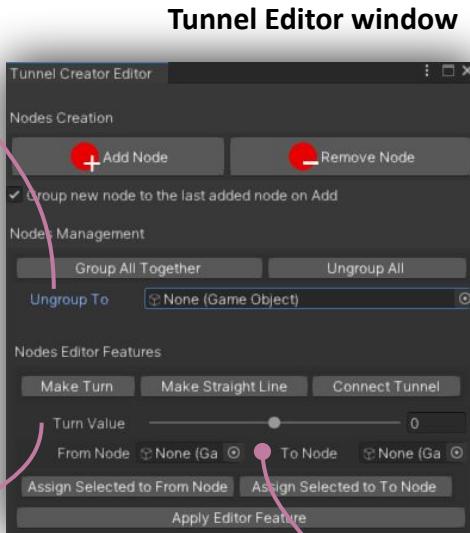
MDG_TunnelCreator



[API documentation](#)

Tunnel Creator detailed description.

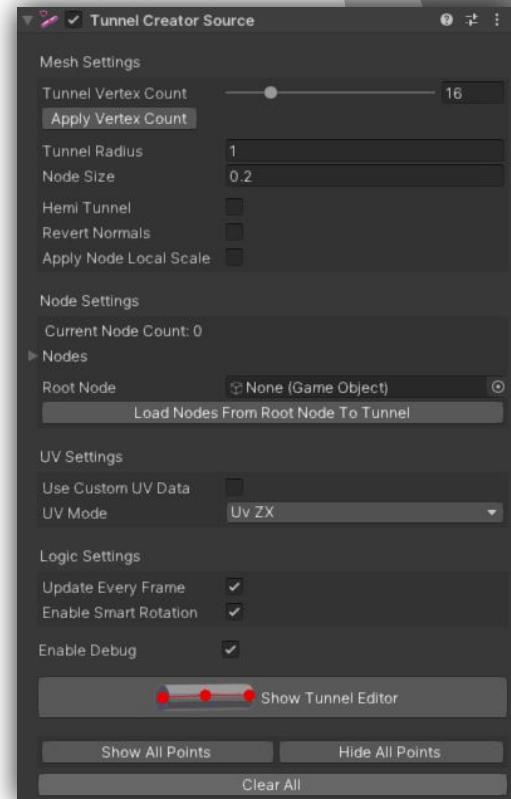
Ungroup all nodes to the desired object



Make turn in degrees (Left/Right)

Process 'Turn' from NODE to NODE including all nodes in between these two nodes. (The more nodes included, the smoother the turn evaluates)

Inspector view



Tunnel Vertex count <

Everytime when vertex count is changed, the Apply button must be pressed & you will lose all your created nodes.

Overall Tunnel Radius <

Overall Node Size <

If enabled, the tunnel will be split into half <

If enabled, the tunnel's normals will be inverted (facing out/in) <

If enabled, the scale of nodes will partially affect tunnel radius <

Currently created nodes (just for debug purposes) <

Root node of loaded nodes <

If the button is pressed, the nodes will be automatically loaded into Nodes list. RootNode field must be assigned. This helps you to restore lost tunnels with stored nodes.

Custom UVData option <

If the field is enabled, each tunnel chunk may contain *TunnelNodeUVData* behaviour to customize UV map. Otherwise choose UV_Mode option.

Update Every Frame (takes more performance if bigger tunnels) <

Enable Smart Rotation <

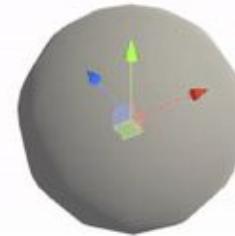
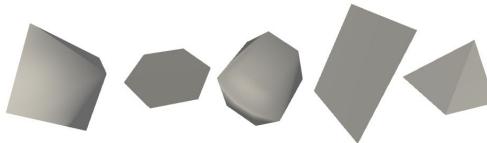
If enabled, the nodes will rotate towards its neighbour. Makes the node editor easier.

Advanced tunnel editor which helps you to do turns and straight lines a bit easier. <

Other Primitives

[API documentation](#)

The Mesh Deformation Package contains other procedural well-known primitives that can be created via *Hierarchy/Create 3D/MD Package*. Create procedural plane, octahedron, triangle, pyramid, spatial & planar hexagon, cone, tube and many more. Also accessible through code: **MD_FullPackage/MD_Core/Geometry**



Technical Info

Technical information about the package & tips in general

[VR Setup](#)

[Multithreading](#)

[Vertex Tool Window](#)

[Performance](#)

[FAQ](#)

[Commercial Products](#)

[Downloadable Content](#)

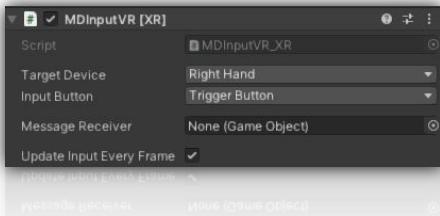
[Extras](#)

VR Setup

The **MD Package** fully supports all the possible VR platforms including Oculus Quest. However, there are some additional steps that you have to make to completely setup the simple VR input for MD Package.

The package contains a folder called **MD_VRPack** which contains all the required components for proper VR setup. There's a folder called **PlatformDependencies** which contains three unity packages. Each package supports specific VR platform that you will have to choose. If you are using SteamVR, import SteamVR package. Same for the other packages.

Once the package is imported, the folder in the root directory will show up with example scenes and **MDInputVR** component. **MDInputVR** is an essential component for simple VR input of the selected platform. It basically tells the VR what button was pressed and sends a message to the desired behaviour. Visit other example scenes for advanced setup of modifiers such as **SculptingPro** or **MeshEditorRuntimeVR**.



If the specified button is pressed, the **MDInputVR** sends a method-message called '*'GlobalReceiver_SetControllerInput'*' to the target object in Message Receiver field. Components that contain this method [MeshEditorRuntimeVR, SculptingPro & MeshPaint] will process the input.

It's a very basic VR input which allows you to record just one button. If you would like to use multiple buttons, you can duplicate components and assign as many buttons as possible.

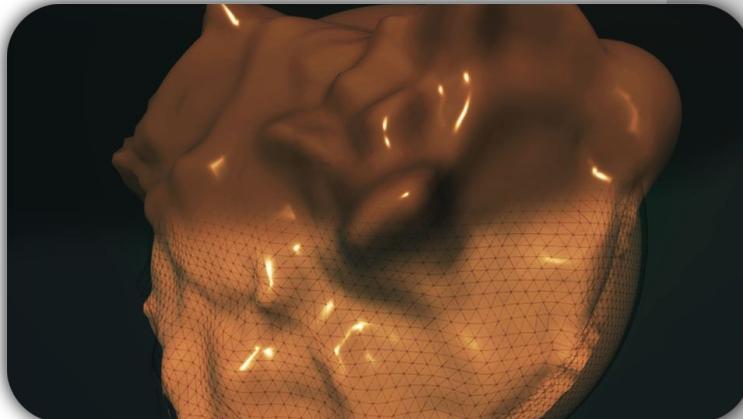
Multithreading

The **MD Package** contains many modifiers that support multithreading. **Multithreading** is a feature to run certain tasks on a different thread. This allows you to manipulate meshes over 2000 vertex count (*2000 vertex count is a recommended value to prevent performance drop down*). However, not all modifiers support this feature and you are forced to optimize your mesh or even not to use it for your ‘project safety’. It’s also recommended not to use many (*10 and over*) modifiers with multithreading feature turned on, this may slow down your performance. Works well on all devices (*PC,VR,Mobile*). [See multithreading in practice.](#)

Car model - 50k vertex count



Planet editor - 16k vertex count





Vertex Tool Window

[API documentation](#)

Vertex Tool Window is an additional feature-tool in MD Package. You can access it in '**Window/MD_Package/VertexTool**' or in the '**Mesh Pro Editor/Vertices Modification**'. It's an expansion for meshes, vertices and elements. It contains its own API for internal use called **MD_VertexToolModifier**. [See Vertex Tool Window in practice.](#)



Attach 2 or more meshes [Meshes will be combined and will share the same material]

Clone selected mesh [Parameters below - Count, Position Offset and Rotation Offset]

Weld selected vertices [Vertices will be weld – they will split into one]

Relax selected vertices [Vertices will be normalized and their offset will be multiplied]

Group selected objects into included object below

Group enabled vertices [Additional group function to group enabled vertices in Zone-Generator Mode in MeshProEditor]

Performance & Complexity

MD package has limitless possibilities, but limited performance. That means, it can get more difficult with higher-poly meshes with some modifiers & components. Please read the important limitations below that you should know before buying the MD Package.

Essential component (**MeshProEditor**)

Mesh Pro Editor allows you to edit mesh vertices directly in the editor as well as at runtime. The component has a condition that if the mesh has more than **N-count vertices** (*specified in preferences*), you will have to use *Zone-Generator* or you will be forced to optimize your mesh to lower the vertex count. This is not an issue nor feature, as the **MD Package is not focused on advanced Mesh Editor** or vertex editor. Mesh Pro Editor allows you to manipulate with meshes and you can generate its mesh vertices as objects **on a basic level**. If the mesh is beyond the 10 000 vertex count, you won't be able to generate vertices and edit them. This is the major and the only limitation in MeshProEditor. If you would like to edit such meshes (beyond 10 000 vertex count), use multithreaded modifiers such as FFD, MeshEffector, SculptingPro and more (*see modifiers slides*).

Modifiers

As mentioned earlier, the package contains multithreaded modifiers that can be used to edit extremely hi-poly meshes. However, this still has a bottleneck. As the Multithreading slide says: having even more multithreaded components may cause performance issues, so it's recommended to organize & optimize your project well and use the least possible multithreaded modifiers.

FAQ

- Is the MD Package available for Mac or Linux?

Yes, it's available for all operating systems & devices.

- Can I use the MD Package in my mobile game/application?

Yes, you can, but the mobile performance is very important (depends on goals & how you organize your project).

- Can I edit complex meshes at runtime/ in editor?

Yes you can, but it depends which modifier you'll choose to work with. Please read [Performance & Complexity](#) slide for more info.

- Is it possible to export my mesh to OBJ format?

No, it is not possible with MD Package. Use a different external plugin. But you can save your mesh to Assets and create a prefab in the Unity editor.

- Do I need any programming skills to use the MD Package?

No, you don't need any programming skills. But if you would like to make complex operations or custom modifiers, you can use the internal API.

- Am I able to edit a Skinned Mesh object?

Yes, but no. Let me explain. Skinned Mesh Renderers control your mesh by bones. In the MD Package - the Skinned Mesh Renderer objects have to be converted into Mesh Filters. That means, you will have 2 copies: Original skinned mesh object and editable mesh filter. Original Skinned mesh object will have the mesh source from the editable mesh filter, that means every change made on the mesh filter will be applied into the original skinned mesh object.

- Do I need latest Unity version to use the MD Package?

No, you don't. But it's much safer and recommended to use the latest Unity Version. If you are going to use the older Unity version, the code conversion and compilation will be required and you can get some warning messages.

- Is the [Mesh Tracker](#) included in the MD Package?

No, it's not included in the package. But the package contains a similar modifier called [Surface Tracking](#).

- Does the MD Package work with WebGL?

Yes, but the multithreading feature is not allowed so you have to save the performance on your own. (That means, you might experience some performance issues while sculpting high-poly meshes in a built game in WebGL).

- Does the MD Package work with Unity HDRP or URP?

Yes, it does, but [shaders won't work](#) as they use [Unity standard pipeline](#). All the other components and modifiers will work as they are logical part of the package.

- Does the MD Package support editing of Unity terrains at runtime?

No, MD package doesn't work with Unity terrains at all. Meshes only.

FAQ

- Are there any project sources of the available free demos that use the MD Package?

No, for privacy & license purposes, there are no project sources of the free demos that use the MD Package, you can only try them & test them for free on your own.

- Do you plan to add URP/HDRP support for shaders in the MD Package?

Yes, I plan to add URP/HDRP support for shaders in the MD Package. Hopefully in the 2024!

Commercial Products

There are some commercial projects that use the MD Package tools and all its features. Explore these projects below. [Click the image]



Downloadable Content

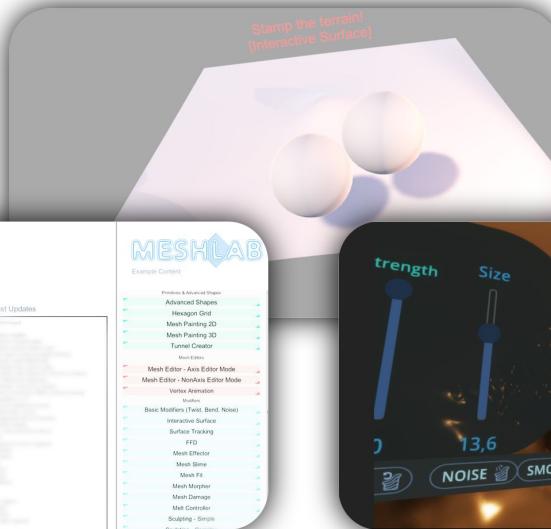
You are very free to download official example content of the MD Package.
Available for Windows OS only. [Click the image]

Terrain Sculpting PC [Win]



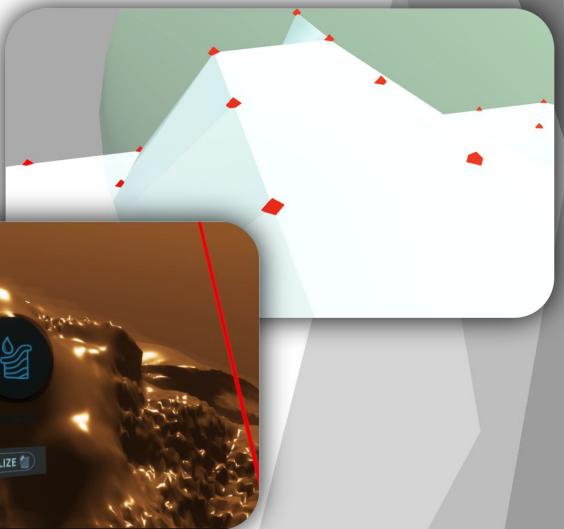
MD Package official examples [Win]

Modifiers in VR [Win, Unity-XR]



Terrain Sculpting in VR [Win, Unity-XR]

Mesh Editor in VR [Win, Unity-XR]



Extras

The MD Package contains some additional short demo-games that were completely built from the package modifiers and shaders. Explore these games for free below! Just click the gif image.



Thank you!

Thank you for your attention. I hope the general documentation was useful! If you have any questions, suggestions or issues, do not hesitate and join my official discord server for **quick & realtime** support.



If you don't like Discord, you can still contact me [here](#).
(But it may take some time to respond)