by Matej Vanco 2020-2023 ©

Sculpting Pro

Official Documentation

v3.0.0

Official Website

APi Documentation

Change Log v3.0.0

02.02.2023 [dd/mm/yyyy]

New collection of primitives

Brand new APi documentation

Seams calculation optimized

Improved sculpting in Unity Editor

Major code refactor

Ready for Unity 2020, 2021 and 2022

Official roadmap

Trello

Curious what the future holds? Click the Trello icon!

Content

Introduction

Package description in general

Sculpting Pro is a complete multi platform solution of mesh sculpting in Unity Engine. Manipulate with any kind of meshes with user-friendly and well-known sculpting tools directly in Unity editor or at runtime.

Key Features:

- Complete mesh sculpting in editor/ at runtime
- Well-known brush tools (Raise, Lower, Grab, Smooth…)
- Advanced mesh features (Subdivision, Smooth, Flip…)
- Multiplatform support (PC, VR, Mobile)
- Fully multithreaded
- History feature
- Vertex-seam handler
- Procedural primitives
- Full API with accessible methods
- Example content

Upgrade your level design experience and bring a new perspective to your workflow.

Essentials

Each component & script is described in the API slide. Non-mentioned scripts are not important for the user.

Sculpting Pro contains **4 essential scripts**

- **SculptingPro_Thread**
  Multithreaded mesh-sculpting source.
  Handles itself automatically, user doesn't need to know about this.
- **SculptingPro_Model**
  Essential component for sculpting-ready meshes.
  Apply this component to a mesh you would like to sculpt.
- **SculptingPro_Editor**
  Main sculpting editor in the editor Scene-View only.
  Editor-only feature. Makes the sculpting possible in Unity Editor.
- **SculptingPro_Realtime**
  Main sculpting editor at runtime only.
  Runtime-only feature. Makes the sculpting possible in Application.

There are **other scripts available for use** that the plugin offers

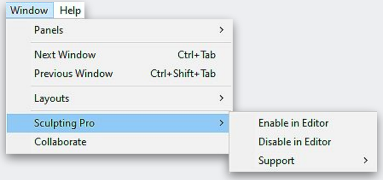
- **SculptingPro_ProceduralPrimitives**
  Collection of basic procedural primitives.
  Can be found in Hierarchy/Create/3D Object/SculptingPro.
- **SculptingPro_Utilities**
  Global utilities ready for 3rd party operations.
  Complete API of 3D math library.
- **SculptingPro_Preferences**
  Global preferences for SculptingPro plugin (Window/SculptingPro/Preferences)
  Contains an internal debug, package info & global plugin params.
- **SculptingPro_InputSample / VR Input Dependency**
  Simple input handler for realtime sculpting.
  Sample input script for PC, Mobile & VR. Helps you to control sculpting events.

Setup

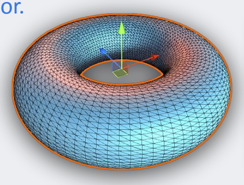
My first Hello World with sculpting in Unity Engine

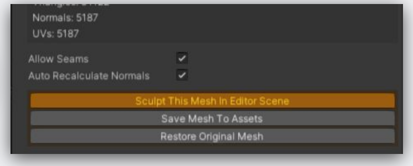
Sculpting Pro allows you to sculpt almost any mesh. The mesh must contains **SculptingPro_Model** component which handles mesh data.

Sculpting in **Editor** can be initialized via **Window/Sculpting Pro/Enable** or **Disable**



Or select any object with **mesh filter**, add **SculptingPro_Model** component and press the orange button **Sculpt This Mesh In Editor Scene** in the inspector.





Normals: 5187
UVs: 5187

Allow Seams ☑
Auto Recalculate Normals ☑

Sculpting at **Runtime** can be initialized by adding the **SculptingPro_Realtime** component onto any object in scene. Field **Target Sculpting Brush** must be assigned. The field indicates to the visual sculpting brush (More in API).

There are two modes of sculpting: **Global Sculpting** & **Locked Sculpting**

Global Sculpting doesn’t focus on any specific mesh and it’s possible to sculpt all meshes with **SculptingPro_Model** component at once. In such case, you are not able to use **Mesh Features** which enables advanced operations such as subdivision or overall smooth. To access this sculpting mode, don’t select any object. The editor will appear gray (in the editor-scene view).

Locked Sculpting focuses on specific mesh with **SculptingPro_Model** component. In such case, you are able to use advanced **Mesh Features**. To access this sculpting mode, select one object with mesh filter in the Unity Editor and execute Sculpting Editor. The editor will appear orange (in the editor-scene view). To access this sculpting mode **at runtime**, select object with **SculptingPro_Realtime** and enable **Locked Sculpting** field. Assign the desired object into required field. (More in API about realtime sculpting).

Setup VR

My first Hello World with
VR sculpting in Unity Engine

Sculpting Pro allows you to sculpt almost any mesh in **VR mode at runtime**. To properly setup the Sculpting Pro for VR, go to the **SculptingPro/VR** and **import proper VR framework**.
(If you use SteamVR, import SteamVR plugin & InputDependency_SteamVR).
If the VR framework doesn't match with the imported input dependency, **the compiler errors appear**.
I recommend to use XR Input dependency as the XR is the Unity's built-in VR framework (At least for testing purposes).

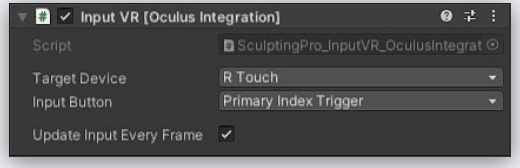
If the import was successful, explore the **example VR content** or **create your VR sculpting component** from scratch.

To create a sculpting component from scratch, drag & drop your **VR-player rig prefab**
(or **create a new one** - VR Head, Right & Left Controller).
Choose one of the **VR controllers**, add **SculptingPro_Realtime** component to the chosen VR controller.
Switch **Target Platform** to **VR** and assign **Target Sculpting Brush** field. The field indicates to visual sculpting brush.

At last but not least, add **SculptingPro_InputVR_YOUR VR FRAMEWORK** component and choose your desired button which will control the sculpting process. Done!

Supported VR frameworks in SculptingPro - **SteamVR, Unity XR, Oculus Integration**

*For example, this is the VR Input script for Oculus Integration.
Target Device refers to the target controller & Input Button to the actual input.*



☑ Input VR [Oculus Integration]

Script: SculptingPro_InputVR_OculusIntegrat

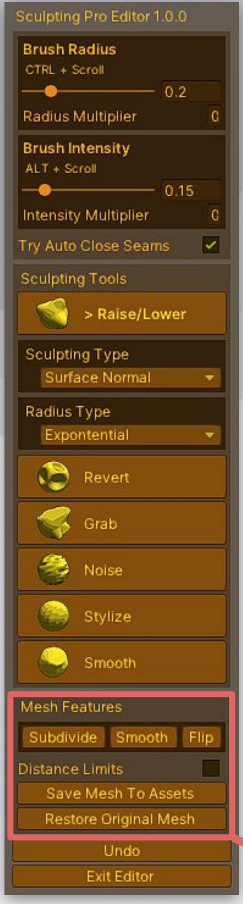
Target Device: R Touch

Input Button: Primary Index Trigger

Update Input Every Frame ☑

Editor Sculpting pt.1

Basic use of mesh sculpting in
the Unity Editor



Sculpting Pro Editor 1.0.0

Brush Radius (CTRL + Scroll): 0.2

Radius Multiplier: 0

Brush Intensity (ALT + Scroll): 0.15

Intensity Multiplier: 0

Try Auto Close Seams ☑

Sculpting Type: Surface Normal

Radius Type: Expontential

Distance Limits □

As mentioned in the previous slide, there are two modes of sculpting: **Global Sculpting** and **Locked Sculpting**. In the following example we will use both modes and you will learn the basics of sculpting in the Unity Editor.

Locked Sculpting (Editor appears orange - as shown on the left side)

1. Create any regular 3D object with **mesh filter**
2. Add **SculptingPro_Model** component onto created object
3. Press **Sculpt This Mesh In Editor** in Inspector

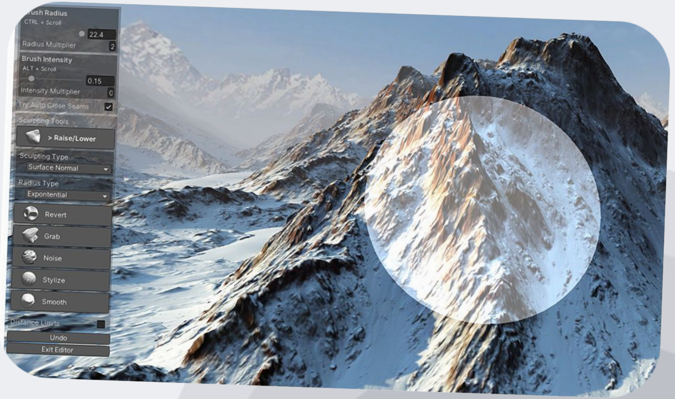
   or press **Window/Sculpting Pro/Enable** in Editor

Global Sculpting (Editor appears gray)

1. Create two or more any regular 3D objects with **mesh filters**
2. Add **SculptingPro_Model** component onto all created objects
3. Go to **Window/Sculpting Pro/** and press **Enable in Editor**

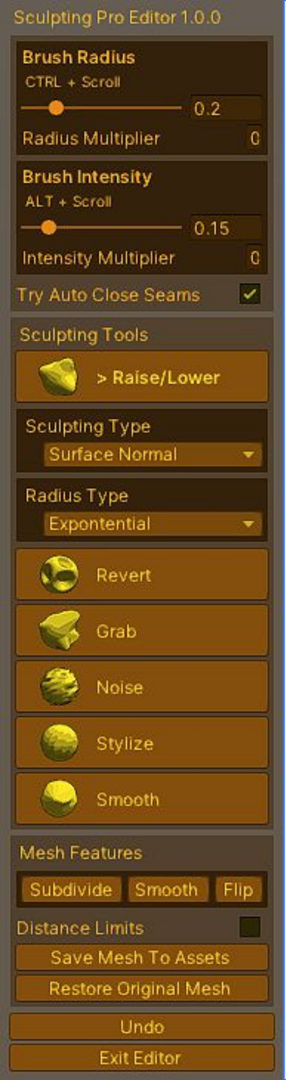
The sculpting editor will appear and you are ready to sculpt! Full editor-panel description can be found on the next slide.

Please keep in mind that this panel is not available in **Global Sculpting**. You have to select a specific mesh to access Mesh Features!



Editor Sculpting pt.2

Sculpting editor panel description in the Unity Editor



Sculpting Pro Editor 1.0.0

Brush Radius
CTRL + Scroll
0.2
Radius Multiplier 0

Brush Intensity
ALT + Scroll
0.15
Intensity Multiplier 0

Try Auto Close Seams ☑

Sculpting Tools

Sculpting Type: Surface Normal

Radius Type: Expontential

Mesh Features

Distance Limits □

Essentials

- Brush radius
  Use left control + scroll shortcut to change in Editor
- Radius multiplier, which sets the radius limits above (great for huge meshes)

- Brush intensity
  Use left alt + scroll shortcut to change in Editor
- Intensity multiplier, which sets the radius limits above

- Try Auto Close Seams = While manipulating with mesh, the seams might run away and the mesh may open. If the field is enabled, the vertices will try to connect with neighbours. Used mostly for spatial meshes, planar meshes don't need this feature (Takes more performance, but evaluates much better results)

Sculpting tools

- Sculpting Tools > Raise/Lower = raise/lower mesh surface by radius & intensity
  Use left control + left mouse shortcut to lower.
- Sculpting type & radius type (all sub-settings are described in API slides)

- Sculpting Tools > Revert = reverts current vertices to its initial state
- Sculpting Tools > Grab = grab certain vertices and move them in desired direction
- Sculpting Tools > Noise = noise certain vertices in desired 'noise' direction
- Sculpting Tools > Stylize = interpolate and 'weld' certain vertices (custom tool)
- Sculpting Tools > Smooth = smooth certain vertices by chosen smoothing filter

Mesh Features *(available if the Locked Sculpting is active [if one object is focused])*

- Subdivide (populate vertices), Smooth (smooth overall mesh) & Flip faces
- Enable/ Disable distance limitations = limit vertices position between initial state and current state
- Save mesh to assets folder (converts JUST to .asset format) and restore mesh

- Make a step back if possible & alternate editor exit (The same as Window/SP/Disable)

Quick Sculpting Editor shortcuts

[Sculpting editor must be enabled]

Change radius size
Left Control + Mouse Scroll Wheel

Change intensity value
Left Alt + Mouse Scroll Wheel

Process any sculpting tool
Aim on mesh & press Left Mouse Button

When in Raise/Lower tool:

Process lower tool
Left Control + Left Mouse

Heightmap Sculpting

Heightmap brush feature.
Stamp any texture onto any mesh

Heightmap stamp in SculptingPro does a pretty impressive work. You can literally stamp any black & white texture onto any mesh with amazing results.

In order to create a stamp texture, you need to create a **SculptingPro_HeightmapBrush** object in your assets. Go to the **Assets** and press **Create/SculptingPro/Heightmap brush object**.

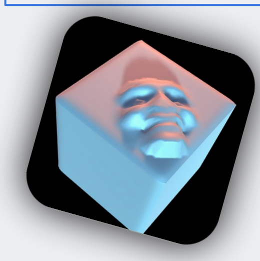
Select the Heightmap brush object and assign your heightmap texture. Follow further instructions to complete the Heightmap brush object (such as check for correct dimensions, if the texture is readable etc… The script will do it for you, all you need to do is just to pay attention and do some clicking).

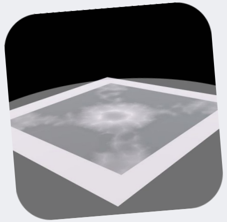
If everything is ready, **your heightmap brush is ready**.

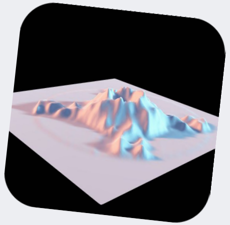
In editor sculpting, select the *Heightmap brush state* and assign your chosen heightmap brush object. There are advanced options such as horizontal brush and brush rotation.

At runtime sculpting, select the *Heightmap brush state* in **SculptingPro_Realtime** component and assign your chosen heightmap brush object.

Please keep in mind: the more vertices your mesh has, the longer it takes to complete the ‘heightmap-stamp’ process (for 40k meshes it may take even for 30 seconds to complete). **Also Heightmap stamping is not allowed for non-multithreaded sculpting.**







Individual Components

Description of individual components
in the Sculpting Pro

Sculpting Pro contains well-organized APi that allows you to do more than just sculpting. Obtain full control and improve your project to another level. All public methods are accessible via Unity Events (for example in UI) as well.

Namespace: SculptingPro
Editor namespace: SculptingProEditor
Naming macro: SculptingPro
Components utility: Matej Vanco/Sculpting Pro

Choose one of the scripts below and explore their API & content.
Each field on every script contains tooltip information (hover the cursor over the field).

Production-Essentials

Scripts that user will mostly use in practice - in Unity Editor & at runtime. These scripts can be seen in the Unity Inspector.

SculptingPro_Realtime

SculptingPro_Model

SculptingPro_ProceduralPrimitives

SculptingPro_InputSample

Advanced-Internal

Scripts that are required for sculpting in general, but user won't need them, but it's good to know them & have them!

SculptingPro_Utilities

SculptingPro_Preferences

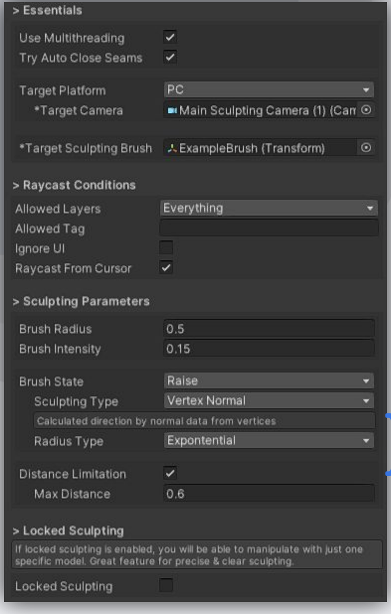
Inspector description

APi Documentation

Production Essential script

SculptingPro Realtime

Full solution of sculpting at runtime only.
This script is applied once on any object in Unity
scene to access sculpting at runtime.



> Essentials

Use Multithreading ☑
Try Auto Close Seams ☑

Target Platform: PC
*Target Camera: Main Sculpting Camera (1) (Can

*Target Sculpting Brush: ExampleBrush (Transform)

> Raycast Conditions

Allowed Layers: Everything
Allowed Tag:
Ignore UI ☐
Raycast From Cursor ☑

> Sculpting Parameters

Brush Radius: 0.5
Brush Intensity: 0.15

Brush State: Raise
Sculpting Type: Vertex Normal
Calculated direction by normal data from vertices
Radius Type: Expontential

Distance Limitation ☑
Max Distance: 0.6

> Locked Sculpting

If locked sculpting is enabled, you will be able to manipulate with just one specific model. Great feature for precise & clear sculpting.

Locked Sculpting ☐

> If enabled, the sculpting will be ready for complex meshes
> If enabled, the auto-seam handler will process 'broken' vertices

> Required target platform (PC/Mobile/VR)
> Required target camera or VR controller

> Required sculpting brush for sculpting visualization
(Can be fully customized, use ExampleBrush in Prefabs folder)

> Allowed raycast layers & tag (leave the tag field empty if you would like to sculpt all possible meshes)
> If enabled, the raycast will continue if cursor will hover any UI
> If disabled, the raycast origin will be (if PC/Mobile) screen center - camera forward or (if VR) target controller forward

> Brush radius & brush intensity values

> Target brush state (each brush state contains its own parameters. See on the right)

> Distance limitations in which certain vertices will be limited from their initial position to the current

> Locked sculpting option

List of **Smoothing Types : Enum** (excerpted from the script)
(Smoothing types indicates to the various smoothing-filter algorithms)

- **Laplacian Filter** = 0 [Laplacian filter algorithm of mesh smoothing]
- **HC Filter** = 1 [Humphrey's [HC] filter algorithm of mesh smoothing]

List of **Noise Directions : Enum** (excerpted from the script)
(Noise Directions indicates to the type of noise behaviour)

- **XYZ =** 0 [Noise certain vertices to the selected direction]
- **XZ =** 1 [Noise certain vertices to the selected direction]
- **XY =** 2 [Noise certain vertices to the selected direction]
- **YZ =** 3 [Noise certain vertices to the selected direction]
- **Z =** 4 [Noise certain vertices to the selected direction]
- **Y** = 5 [Noise certain vertices to the selected direction]
- **X** = 6 [Noise certain vertices to the selected direction]
- **Centrical** = 7 [Noise certain vertices towards objects central position]
- **Vertex Normal** = 8 [Noise certain vertices by the vertices normal data]

List of **Brush States : Enum** (excerpted from the script)
(Brush state indicates to the selected sculpting tool)

- **None** = 0 [no tool selected]
- **Raise** = 1 [raise mesh surface by radius & intensity]
- **Lower** = 2 [lower mesh surface by radius & intensity]
- **Revert** = 3 [reverts current vertices to its initial state]
- **Grab** = 4 [grab certain vertices and move them in desired direction]
- **Noise** = 5 [noise certain vertices in desired 'noise' direction]
- **Smooth** = 6 [smooth certain vertices by chosen smoothing filter]
- **Stylize** = 7 [interpolate and 'weld' certain vertices (custom tool)]
- **SetHeight** = 8 [set specific height of the selected vertices]
- **Heightmap** = 9 [stamp heightmap texture onto any mesh by certain pixel data]

List of **Sculpting Types : Enum** (excerpted from the script)
(Sculpting type indicates how the vertices will behave while raising/lowering)

- **Vertex Normal =** 0 [Calculated direction by normal data from vertices]
- **Vertex Direction =** 1 [Calculated 'natural' direction from vertices]
- **Surface Normal =** 2 [Calculated surface normal direction by brush & raycast]
- **Custom Direction =** 3 [Custom Vector direction in which certain vertices will move]
- **Custom Object Forward Direction =** 4 [Custom object-local-forward direction]

List of **Radius Types : Enum** (excerpted from the script)
(Radius type indicates how the vertices will behave while processing)

- **Exponential** = 0 [Exponential vertices affection - smoother motion]
- **Linear** = 1 [Linear vertices affection - 'clunky' motion]

Smoothing types and **Noise directions** on the next slide...

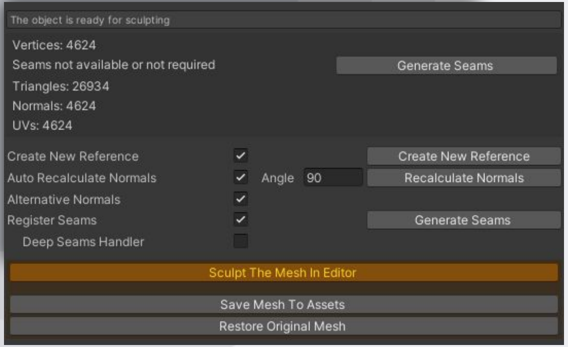
Inspector description

[APi Documentation]

Production Essential script

SculptingPro Model

Essential component for sculpting-ready meshes. Object must contain MeshFilter in order to make the sculpting work properly.



The object is ready for sculpting

Vertices: 4624

Seams not available or not required — Generate Seams

Triangles: 26934

Normals: 4624

UVs: 4624

Create New Reference ☑ — Create New Reference

Auto Recalculate Normals ☑ Angle 90 — Recalculate Normals

Alternative Normals ☑

Register Seams ☑ — Generate Seams

Deep Seams Handler □

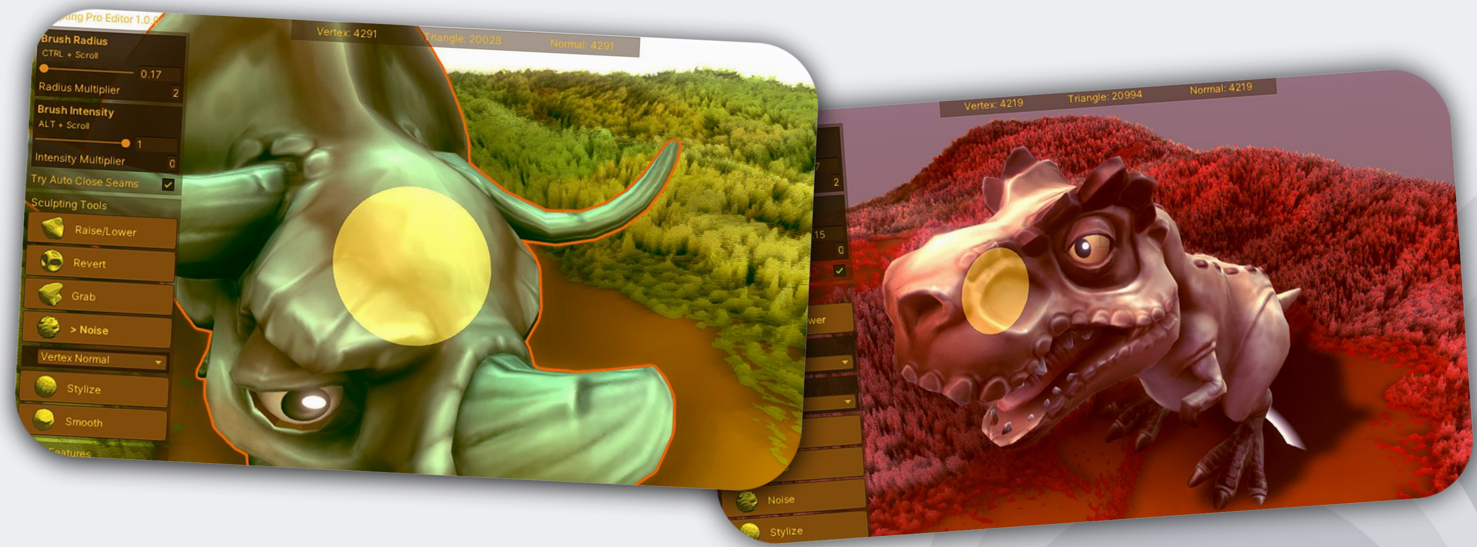
Sculpt The Mesh In Editor

Save Mesh To Assets

Restore Original Mesh

> Essential mesh information

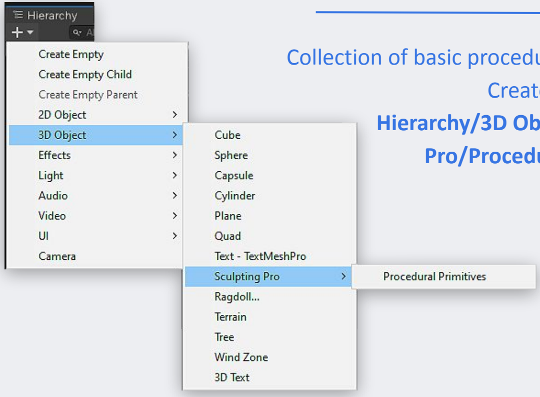
> Create a new mesh reference on copy/paste or any other action
> If disabled, the normals won't be automatically calculated
> Use alternative normals recalculation (smoother results)
> Handle vertex-seams (More about seams here)
> Deep-seams handler - precise vertex-seam calculation



SculptingPro Procedural Primitives

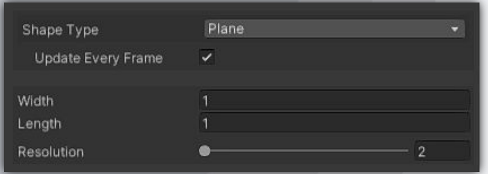
Collection of basic procedural primitives.

Create an object via **Hierarchy/3D Object/Sculpting Pro/Procedural Primitives**



Inspector description

APi Documentation



Shape Type: Plane

Update Every Frame ☑

Width: 1

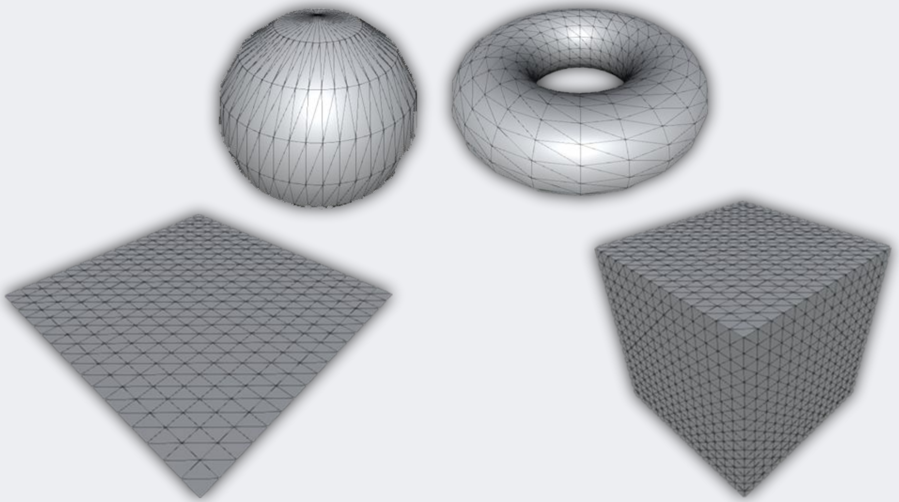
Length: 1

Resolution: 2

> Procedural shape type (Plane, Cube, Sphere, Tube, Cone)

> If enabled, the parameters below will be updated every frame

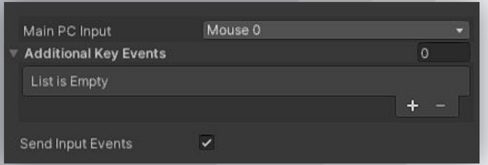
> Parameters of the chosen shape type



SculptingPro Input Sample

Simple input handler for realtime sculpting.
Available for PC & Mobile. Check Setup VR slide for VR input.
Requires **SculptingPro_Realtime** component.

Inspector description



Main PC Input: Mouse 0
Additional Key Events: 0
List is Empty

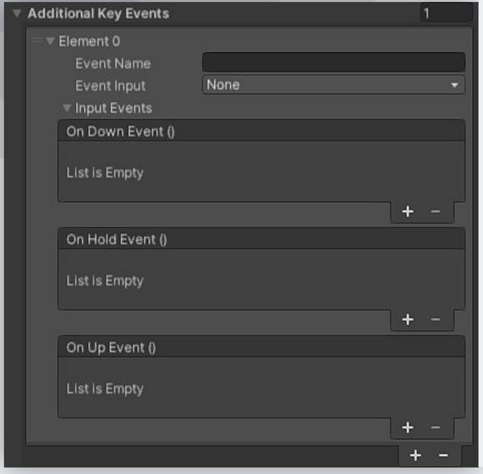
Send Input Events ☑

[Script inspector overview, if target platform is set to PC. Other platforms don't have such options]

> Main key input in which sculpting will be proceeded

> Additional key events
(for example, if you press '1' you can switch to sculpting tool 'Noise')

> If disabled, the script won't send any input data



Additional Key Events: 1
Element 0
Event Name:
Event Input: None
Input Events
On Down Event ()
List is Empty
On Hold Event ()
List is Empty
On Up Event ()
List is Empty

[Script inspector overview, when one of the additional key events is opened]

> Event name (just for your visual information in the editor)

> Event input (required key input for the additional event to process)

> List of input events

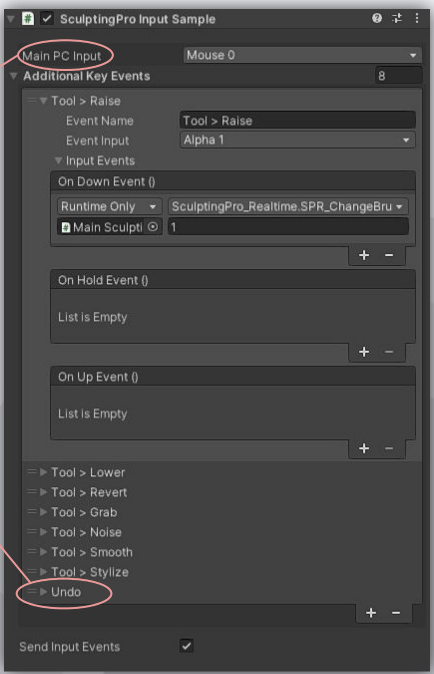
> If the specified Event Input is pressed (process once)

> If the specified Event Input is hold (process repeatedly)

> If the specified Event Input is released (process once)

For example, the example scenes contain structure of this content >

This basically means, that the main sculpting 'process' input is Left Mouse (**Mouse0**) and there are a few 'hotkeys' that allow you to switch between brush types and you can even do **Undo**. That's why you can use **Additional Key Events** which helps you to expand sculpting features at runtime.



☑ SculptingPro Input Sample
Main PC Input: Mouse 0
Additional Key Events: 8
Tool > Raise
Event Name: Tool > Raise
Event Input: Alpha 1
Input Events
On Down Event ()
Runtime Only | SculptingPro_Realtime.SPR_ChangeBru
Main Sculpti ⊙ | 1
On Hold Event ()
List is Empty
On Up Event ()
List is Empty
Tool > Lower
Tool > Revert
Tool > Grab
Tool > Noise
Tool > Smooth
Tool > Stylize
Undo
Send Input Events ☑

Available Public API

SculptingPro_Preferences

public const string **ORGANISATION** - *Organisation name*
public const string **PACKAGENAME** - *Current full package/plugin name*
public const string **VERSION** - *Current SculptingPro plugin version*

public const string **DATE** - *Last SculptingPro update date dd/mm/yyyy*

public static readonly Color **COLOR_LOCKED** - *Color for GUI if editor has a selection focus (editor only)*
public static readonly Color **COLOR_UNLOCKED** - *Color for GUI if editor doesn't have a selection focus (editor only)*

public static int **pref_MaxHistoryRecords**
public static bool **pref_RecalculateNormalsAsDefault**
public static bool **pref_RecalculateNormalsOptimized**
public static bool **pref_CreateNewReferenceAsDefault**
public static Color **pref_LockedSculptingEditorColor**
public static Color **pref_UnlockedSculptingEditorColor**

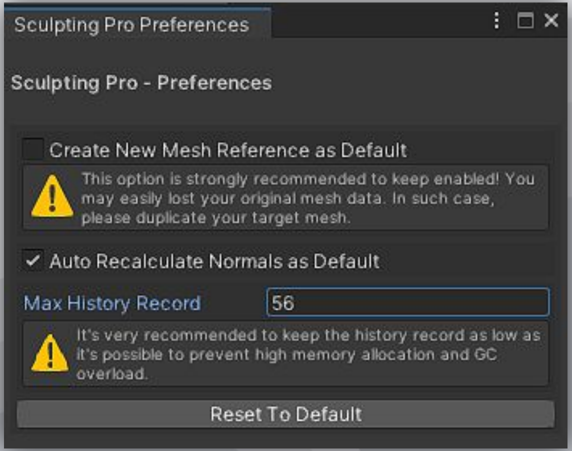
SculptingPro_Debug

public static void **Debug(Mono, string, DebugType)** - Output specific message via SculptingPro

SculptingPro Preferences

Global preferences for SculptingPro plugin.
Access this class via script or open an editor window
Window/Sculpting Pro/Preferences

The available fields in Preferences window indicate to the default values that some SculptingPro components use at the time of their initialization. You can customize these fields by your own needs. Please read the warning boxes properly!



Sculpting Pro - Preferences

□ Create New Mesh Reference as Default

This option is strongly recommended to keep enabled! You may easily lost your original mesh data. In such case, please duplicate your target mesh.

☑ Auto Recalculate Normals as Default

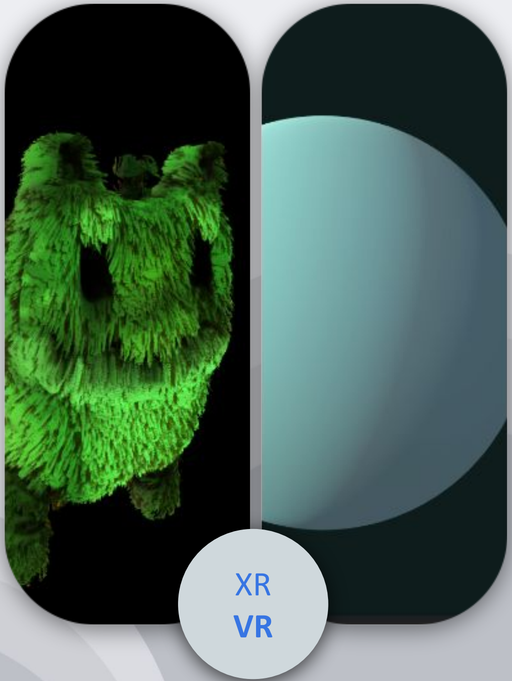
Max History Record 56

It's very recommended to keep the history record as low as it's possible to prevent high memory allocation and GC overload.

Example Content

Explore extended content of the Sculpting Pro!

Sculpting Pro contains a complete example content, which is available online as well. Choose one of the demos below and test your performance with official examples. **Just click the image**.



FAQ

Frequently Asked Questions

- **What is the maximum vertex count for a mesh?**

*The maximum **recommended** vertex count is around ~100k. The Unity's default index format is 16 bit (up to ~65k vertices). However it is possible to use the 32 bit index format (up to 4 billion vertices). It all depends on two factors: your CPU & multithreading performance.*

- **Does the SculptingPro support WebGL?**

Yes, SculptingPro does support WebGL. However, the WebGL is the only platform with specific exception - WebGL does not support multithreading in a build, so it would not be possible to sculpt meshes beyond the recommended vertex count (~6000).

- **Is it possible to export a mesh to the fbx or obj?**

No, SculptingPro does not contain such feature. You can save your mesh to the assets folder as an 'asset' only. If you would like to export working mesh to the fbx or obj, use another 3rd party plugin.

- **Does the SculptingPro work with Unity Terrains?**

No, SculptingPro does not work with Unity Terrain system. Only if you convert the Unity Terrain to the single mesh (This feature is not included in the SculptingPro plugin).

- **Does the SculptingPro support Linux or OSX?**

Yes, SculptingPro supports all possible platforms.

- **What are the major differences between SculptingPro and SculptingLite (MD Package)?**

There is no further support and no future updates for SculptingLite in the MD Package.
- SculptingLite doesn't contain a vertex-seam handler (which is the biggest advantage in the SculptingPro)
- SculptingLite supports just sculpting-per-object (Each mesh its own sculpting solution)
- SculptingLite contains much fewer features than SculptingPro. However the SculptingLite is in stable and production-ready version.

Summary: If you focus on more advanced, professional and verified sculpting solution inside the Unity Engine, SculptingPro is the way. If you focus on quick and simple sculpting, MD Package is for you (you will even obtain many other amazing mesh features including verified modifiers and various mesh editors).

- **Is it possible to optimize a mesh in the SculptingPro? (opposite to subdivision)**

No, SculptingPro doesn't contain such feature yet.

- **Is Oculus Quest supported?**

Yes, all VR platforms are supported (including cardboard).

- **Does the SculptingPro contain a symmetry feature?**

No, SculptingPro does not contain such feature yet.

FAQ

Frequently Asked Questions

- **Is the SculptingPro compatible with URP/HDRP?**

Yes, SculptingPro is a logical plugin, which means it's not dependent on a renderer pipeline. However, example content that the plugin contains work in Unity Built-In render pipeline only.

- **Sculpting in the Unity Editor is not sometimes responsive… Why?**

It's because the game-view window must be present. If the game-view window is not present, the scene-view window will be paused. Also try to maximize the scene-view by pressing Left Shift + Space (cursor must be over the scene view).

- **Does the SculptingPro contain any example scene with completely custom sculpting process?**

Yes, SculptingPro does contain an example scene with completely custom sculpting process. The scene is called 'SPExample_CustomSculpting'.

- **Does the SculptingPro work with skinned meshes?**

No, SculptingPro does not work DIRECTLY with skinned meshes. However, you can convert the skinned mesh to the mesh filter and manipulate with its mesh source.

- **Does the SculptingPro work with voxels?**

No, SculptingPro does not generate voxels. You can modify existing meshes only.

Thank you

Thank you for your attention. If you have any questions, suggestions or issues, do not hesitate and join my official discord server for quick & realtime support. *(Just click the image)*

If you don’t like Discord, you can still contact me here.

https://matejvanco.com