Last Documentation Video [V1.0]

If the documentation was made in older Mesh Tracker version, it's still related to the actual version.

Version 1.4

Complete tracking solution with modular brushes, canvas and additional tools.

Create advanced simulations just in a second!

Last Overview Video [V1.3]

Developer Change-Log

version ***1.4*** *[07.12.2020 - dd/mm/yyyy]*

- Major upgrade for CPU-Based tracking system
    - Added exponential deformation
    - Improved processing performance
- Added ‘Custom Canvas’ generator feature
- Added ‘Tracker Effects’ feature
- Added noise feature for brush shader
- Fixed track duplications
- Fixed transparent shader
- Optimized overall tracking system twice
- MeshTracker_Fluid removed (replaced with effects)
- Updated overall shader source
- Huge code refactor & scene clean-up
- Updated overall API

Content

Basics & Intro

MESH TRACKER

Mesh Tracker allows you to create advanced mesh simulations on any kind of surface such as snow, mud, sand, sea and many more. Create footprints, trails, dynamic waves and make your project much more professional.
You don't need any programming skills. The plugin is ready for well-organized designers and advanced Unity users.

The package contains 2 most important components:
MeshTracker_Object and *MeshTracker_Track*.
There are a few additional components:
MeshTracker_Particles, *MeshTracker_ProceduralPlane* & *MeshTracker_Drawing*

The whole track system is based on **Ray Casting** - ray origin and ray direction with specific distance and custom conditions.

More about each additional component on next slides...

MeshTracker_Object is a mother-system for **surfaces.** This component must be added to the deformable surface.

It's divided into two categories: **GPU** & **CPU** based

- **GPU-Based** allows you to run the mesh track system via **shaders**, which saves more performance, evaluates better results and works on grayscale masks (requires specific material).
- **CPU-Based** allows you to run the mesh track system mostly via **CPU**, it's supported by Multithreading, less multi-platform issues, easier to set up, but less effective, worse results and might take more performance. The CPU-Based system handles direct mesh vertices.

MeshTracker_Track is a source of tracking. This component allows you to deform objects with ***MeshTracker_Object*** and deform its entire surface. It's based on chronological order of layers & track layers management. Each layer class contains several properties such as Track Size, Track Graphic, Ray Distance, Condition, Events and so on...

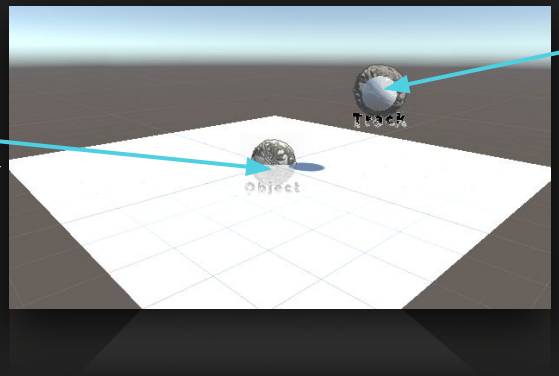
MeshTracker_Particles allows you to simulate tracks by the specific particle collision. This component should be applied onto object with particle component. Requires track graphic and track brush. There is an option to customize track on your own. This component allows you easily simulate **snowing**, **raining**, **water splashing** and more...

MeshTracker_ProceduralPlane is an additional component to generate plane with custom vertex count. This is very **required** component if your application is focused on **mobile platforms.** (mobiles don't support tessellation).

MeshTracker_Drawing is an additional component for surface drawing with user friendly editor and useful functions for drawing. User is able to choose any grayscale brush and apply it to any surface containing ***MeshTracker_Object*** component. It's great to make a **high-quality** looking surfaces.

Quick Application

Application of Mesh Tracker system is very easy. All you need is a surface *[planar object highly recommended]* and Track source.



Planar Mesh representing our surface with MeshTracker_Object

Any Object representing our track source that will hit objects with MeshTracker_Object

Summary:

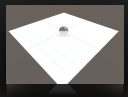
What will you need for basic Mesh Tracking?

- At least one object with **MeshTracker_Object**
- At least one object with **MeshTracker_Track**
- **Track Graphic** [You can use *Track Creator* for creating Tracks]
- Material [shader **MeshTracker_Opaque]** applied onto surface

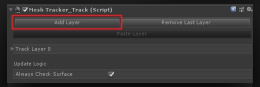
Video Tutorial Here

1. Create basic Sphere and add **MeshTracker_Track** component

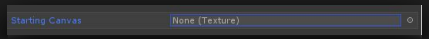
2. Create any planar object and add **MeshTracker_Object** component. Also make sure the object contains material of type **MeshTracker_Opaque.**



3. Select Sphere and add new Layer

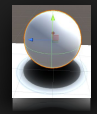


4. Select your planar object and choose any Starting Canvas texture [more in API]

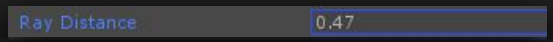


5. Select Sphere again and open your created layer [more about layers in API]. Choose *Track Graphic* texture [more in Track Creator].

6. Enable *Objects Scale Is Track Size* to connect *Track Size* with Objects Scale. Adjust *Scale Multiplier* to fit *Track Graphics* with Sphere.



7. Edit *Ray Distance* on your own. Very quick basics done.

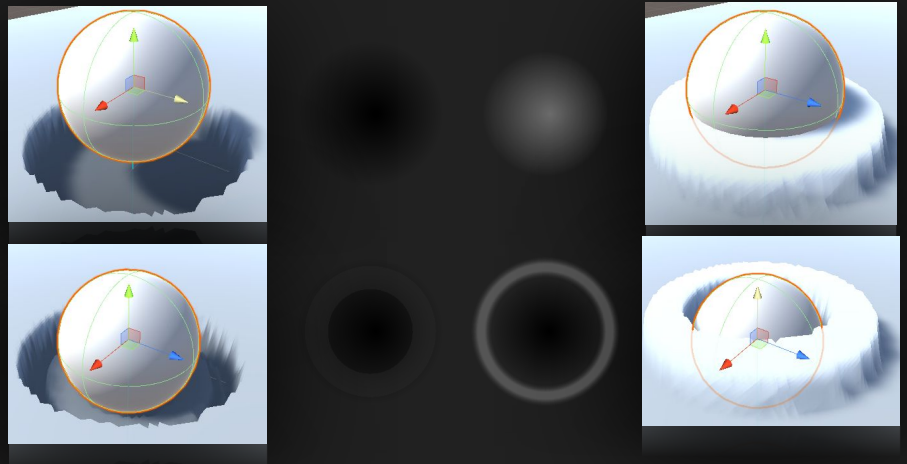


Track Creator

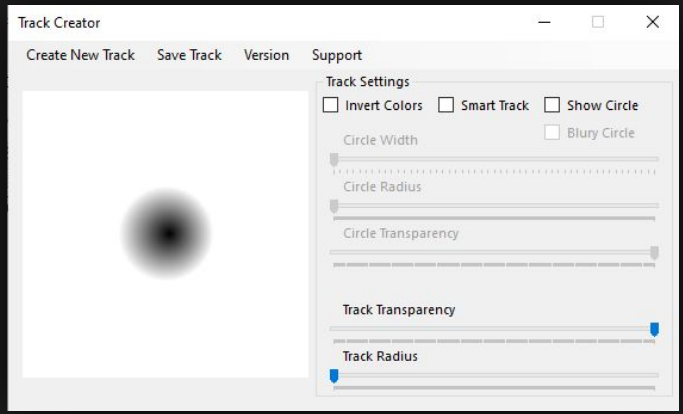
Track Creator is an additional tool to help you create new tracks as quick as possible. But why do I need Track Creator for creating some white and black circles?

Mesh Tracker - GPU-Based system works on grayscale textures which give information of 'how deep/ high the surface should be'. This information is in every texture's pixel.

The more white pixels texture has, the higher surface will be.
The more black pixels texture has, the lower surface will be.



And that's also how you can customize tracks. Track Creator is a great tool for it and you are free-to-use it.



Any texture can represent 'Track Graphic', but Track Creator is specially designed for completed-spherical tracks. This quick solution allows you to start this tool directly from Editor and create track of any type anytime.
Track Creator is unfortunately available for Windows OS only.

API

The whole system contains components with internal & public functions & attributes. If you are experienced programmer and you would like to make some changes in code, this API is for you. All functions can be called internally or through Events. Almost all properties contain Tooltips & well-organized comments.

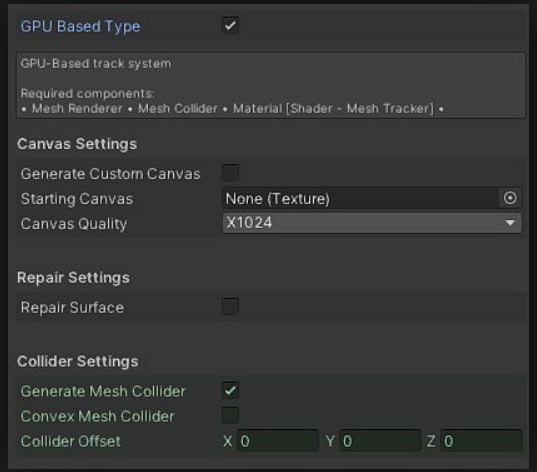
namespace MeshTracker

MeshTracker_Object
MeshTracker_Track
MeshTracker_SurfaceDrawing
MeshTracker_Particles
MeshTracker_ProceduralPlane

Go to the next slide for more API.

MeshTracker_Object

GPU-Based Tracker [masks and grayscale textures]



- Tracking type (GPU/CPU based?)
- Custom canvas (by color tone b/w)
(If enabled, starting canvas not required)
- Starting canvas texture for tessellation
- Canvas texture quality
(The higher quality, the more performance)
- Repair surface feature
- Mesh collider generation

Available Public Functions

public void fGPUbased_CreateTrack(TextureCoords, Size, Graphic, Brush, YRotation)

- Create track on specific texture-coordinate with size, graphic, brush and rotation

public void fGPUbased_SaveCurrentCanvas(FilePath)

- Save current canvas into texture & specific path

public void fGPUbased_ClearCanvas()

- Reset canvas content & reset surface heights

public void fGPUbased_CreateRepairTrack()

- Create a repair track instance on the surface (if original canvas is assigned)

CPU-Based Tracker [mesh, rigidbody, collisions & Unity PhysX]

- Tracking type (GPU/CPU based?)
- Multithreading support for complex meshes
- Overall vertices world direction
- Exponential/ Linear processing
(Exponential evaluates smoother results)
- Adjust radius by the input value
(this is recommended to be checked)
- Additional condition for more physical-based interactions
- Specific tags allowed
- Custom interaction speed of vertices
(If disabled, vertices will be instantly updated)
- Repair mesh by specific speed

Available Public Functions

public void fCPUbased_CreateTrack(AtPoint, Radius, Direction)

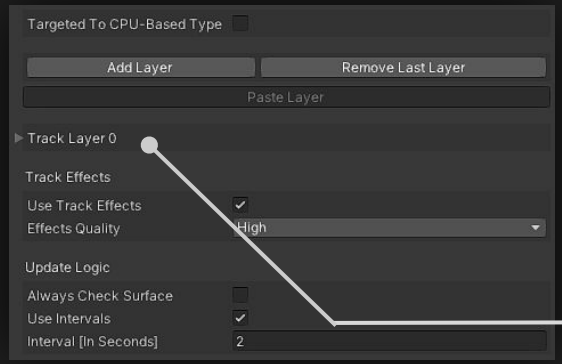
- Create track on specific world-point, radius and direction

public void fCPUbased_ResetSurface()

- Reset current surface & all vertices to the original position

MeshTracker_Track

Base



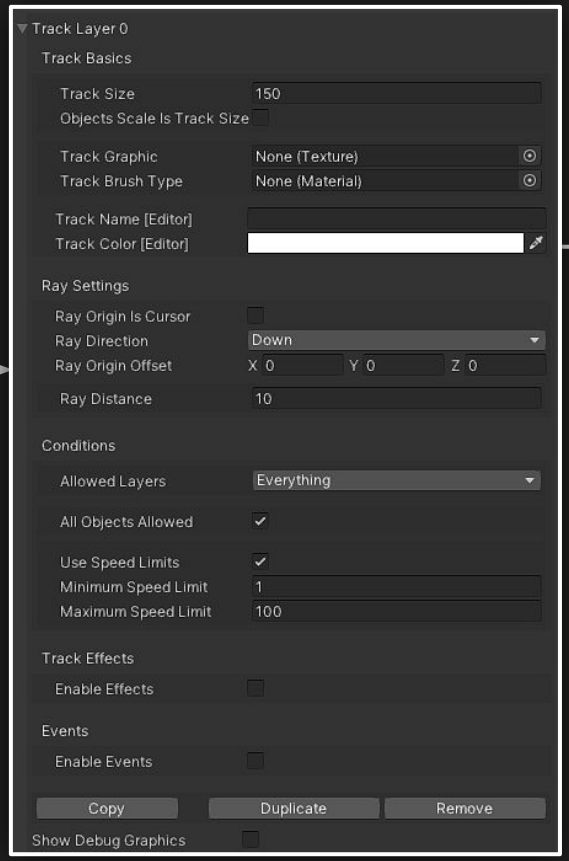
- Is track targeted to CPU/GPU based systems?
- Add & Remove last layer
- Paste layer [if possible]

- Available array of layers

- If enabled, user will be able to use effects in layers
- Quality of effects
(the higher - the more performance)
- If enabled, system will be updated every frame
- Update after some interval
- Interval value in seconds

More about **Track Effects** next slide

Layer



- Track Size (adjust and see in editor)
- Track size will correspond to the objects local scale.
- Track Graphic texture (grayscale)
- Brush Type (for additional details) (more about brushes later)
- Track Name in editor
- Track Color in editor

- Ray origin from cursor input
- Ray direction in world-space (u,d,r,l,f,b)
- Origin location offset

- Raycast distance

- Allowed layers (Default, Water…)

- If disabled, user can edit which tag is allowed
- Use object speed limits

- Track effects **(more in next slide)**

- Enable events (OnSurfaceDetected)

- Copy, Duplicate and Remove layer

- Show debug graphics in editor scene

Available Functions

public void f_DetectSurface()
- Process raycast and detect surface - if the surface is detected, track will be proceeded

- If booleans **AlwaysCheckSurface** and **UseIntervals** are disabled, you are able to call this functions manually. The function will process all exist layers.

MeshTracker_Track - Track Effects

Track Effects allow user to create multiple tracks at once. User can define track of what size will be at the start and what size will be at the end of its lifetime. Track Effects can be used for creating artificial waves, drops, advanced footprints, mud-tracks and many more.

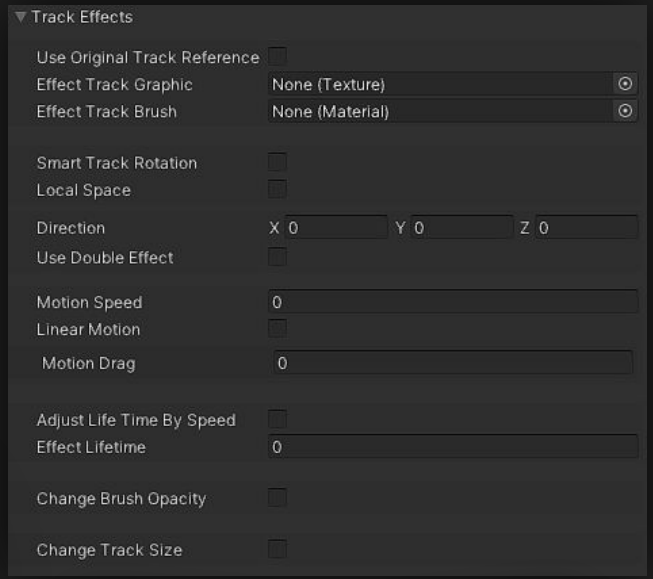
In the earlier versions of Mesh Tracker, there was a component called **Mesh Tracker_Fluid** which theoretically did the same job as Track Effects, but it was very slow and not effective for performance. This component was removed in the Mesh Tracker V1.4 and replaced with **Track Effects.**

Track Effects is a brand new feature built directly in **MeshTracker_Track** component with much better performance results on all available platforms (including mobile).

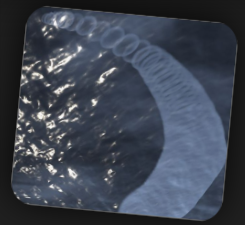
'Side-waves' while walking

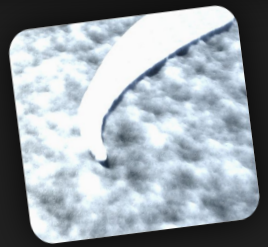
'Side-waves' while driving

Track Effects (in specific Track Layer)



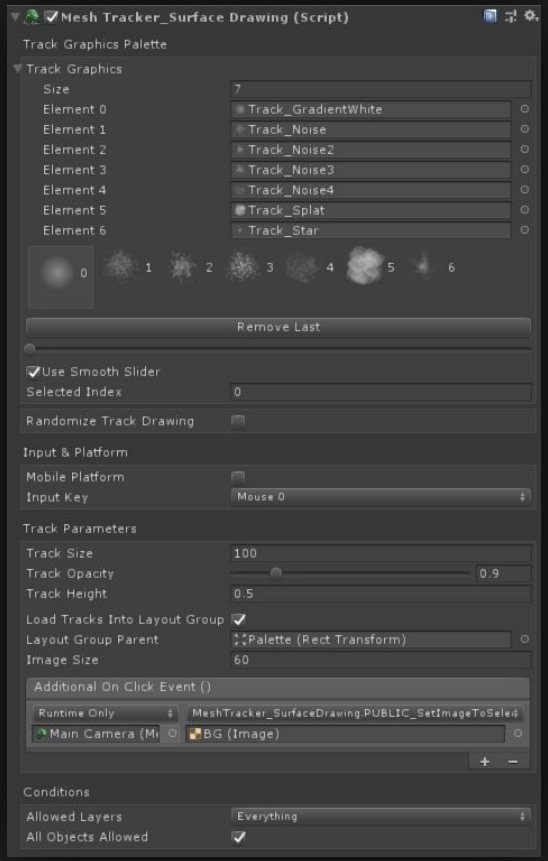
- If enabled, use Graphic & Brush from the original track above
- Smart rotation technique
- Process track effects in local space
- Direction of track effects
- Proceed effect twice?
- Effect motion speed
- If disabled, the effect will move exponentially with specific drag
- Overall effect lifetime in seconds
- Change brush opacity during the lifetime (if possible)
- Change track size during the lifetime (if possible)





MeshTracker_SurfaceDrawing

Base



- Available track graphics
- Slider to select track index
- Randomize tracks while drawing
- Mobile platform supported
- Input Key (If not mobile platform)
- Drawing track size
- Drawing track opacity
- Drawing track height
- Load selected track graphics into specific UI layout parent
- If 'Track image button pressed'
- Additional conditions

[Surface Drawing Tutorial Here]

Available Functions

public void PUBLIC_ChangeTrackGraphic(Parameters)

- Change exists track graphic by specific index in the *mmt_TrackGraphics* array

public void PUBLIC_ChangeTrackSize(Parameters)

- Change current track size by float or UI.Slider

public void PUBLIC_ChangeTrackStrength(Parameters)

- Change current track strength by float or UI.Slider

public void PUBLIC_ChangeTrackHeight(Parameters)

- Change current track height by float or UI.Slider

public void PUBLIC_SetImageToSelectedTrack(Parameters)

- Get selected track graphic to selected image (visualize the selected track)

Shaders

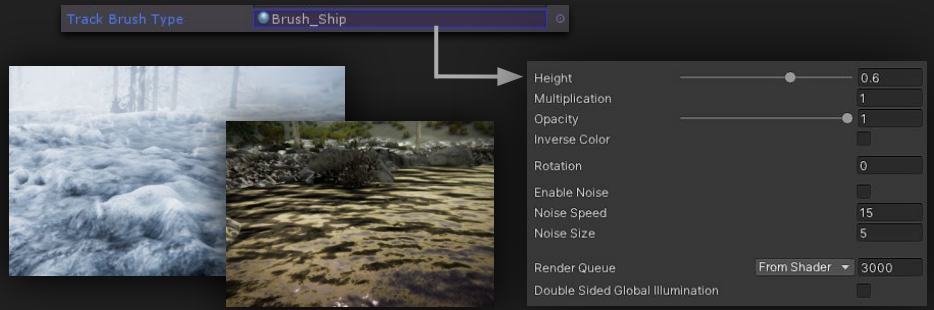
Mesh Tracker can handle systems for both units - CPU and GPU. But the most effective way is to go through GPU - **GPU-Based tracking system**.

The package contains shaders that can be used for any type of surface. Let's see what the shader parameters contain…

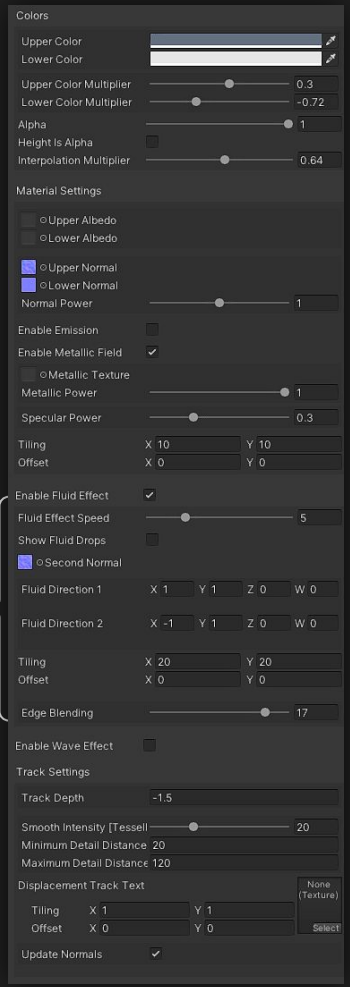
As you can see, the shader has many properties and you can achieve interesting results of your choice. You can create simple water shader in a few clicks.

All shaders can be found in ***Matej Vanco/Mesh Tracker***. There are 4 types of shaders: **Opaque, Transparent, Mobile** and **Brush**. Mesh Tracker Opaque is recommended for non-transparent surfaces. **Mobile shader** works for Mobile platforms *[IOS & Android]*.

Mesh Tracker Brush can be used for Track Graphics as editable tool. It adjusts track graphics opacity and its grayscale-intensity. Material that contains **MeshTracker Brush** can be assigned to the brush type in Mesh Tracker layer.



Some attributes might not be visible in different shader types



Transparent Only

- Upper color value
- Lower color value
- Color multiplications for emissives
- Color alpha + Interpolation multiplier
- Upper Albedo input [Diffuse]
- Lower Albedo input [Diffuse]
- Upper Normal input [Bump]
- Lower Normal input [Bump]
- Normal power
- Emission field [Texture input & color]
- Metallic field [Texture input & intensity]
- Specular power
- Tiling and offset
- Fluid settings
- Fluid speed [speed of textures tiling]
- Enable drops effect [water distortion]
- Additional second normal texture
- Fluid direction 1 [dir of texture1]
- Fluid direction 2 [dir of texture 2]
- Tiling and offset
- Additional edge blending (soft edges)
- Enable wave effect
- Track depth
- Tessalation amount
- Min & Max detail distance
- Displacement texture [should be grayscale or RT] - will be set internally according to your settings]
- Update normals on the object according to grayscale texture

Examples

Mesh Tracker contains many solid examples with full source and descriptions. You are also very free to try four example short scenes that are available for free to download. Please keep in mind that the example demos are focused on the surface simulations and not game mechanics and gameplay.

Mesh Tracker contains all example scenes and more.

Click on images at the right side of the screen to download demo.

Simple sea simulation

Control boat on open sea
Realtime sea simulation
Advanced track effect sources
Full interaction

3rd person snow simulation

Control character on snowy surface
Real Time snow + pond simulation
2 Track sources
Modular and interactive tracks

Vehicle snow simulation

Control vehicle on snowy surface
Real Time snow simulation
4 vehicle types
Modular and interactive tracks

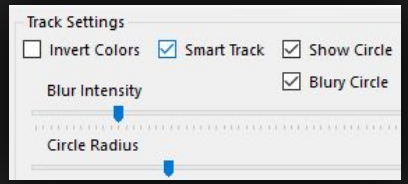
Vehicle mud simulation

Control vehicle on muddy surface
Real Time mud + ‘muddy’ pond simulation
4 vehicle types
Modular and interactive tracks

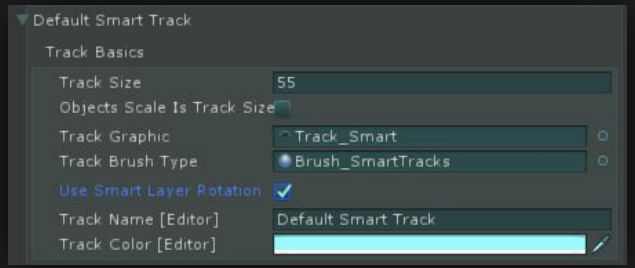
Smart Tracks

[Smart Track Example Tutorial Here]

Smart Tracks is an advanced option for track behaviour. Smart Tracks are used to visualize track by the **objects move direction**. The results are more realistic, but the process requires **special track graphics**. In **Track Creator**, enable *Smart Track* to generate Smart Track graphic.

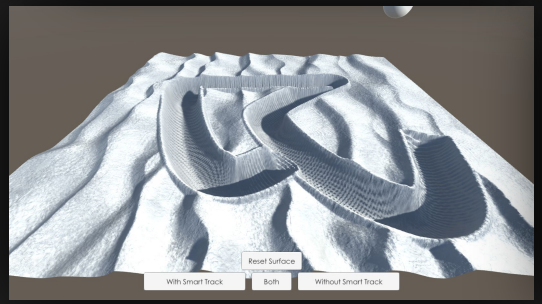


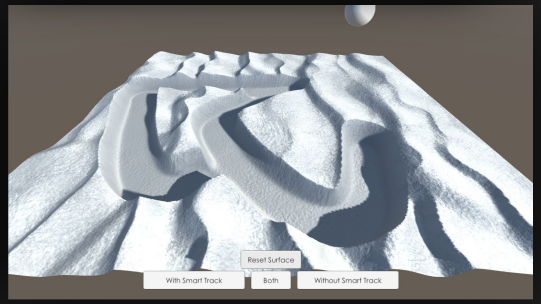
To enable Smart Track feature, go to your track layer and enable *Use Smart Layer Rotation*.



Tracks with Smart Tracks

Tracks without Smart Tracks





FAQ (Frequently asked questions)

- Is Mesh Tracker compatible with Mac OSX?
  - Yes, Mesh Tracker is compatible with all operating systems.
- Is Mesh Tracker compatible with mobile devices?
  - Yes, there is a shader called MeshTracker_Mobile which supports iOS and Android devices. ***Vulkan*** *is the only supported GPU API.*
- Is Mesh Tracker compatible with VR?
  - Yes, but surfaces must be optimized and well-prepared.
- Is Track Creator [external tool] available for OSX or Linux?
  - No, Track Creator program is available for Windows OS only.
- What's the difference between **GPU-Based** and **CPU-Based** Mesh Tracker?
  - ***GPU-Based*** *requires specific material and grayscale textures & it runs via shaders.* ***CPU-Based*** *modifies mesh vertices and can be fully multithreaded. It doesn't require any grayscale textures.*
- Why can I choose between **GPU-Based** and **CPU-Based** Mesh Tracker?
  - ***GPU-Based*** *is more advanced technique than* ***CPU-Based****.* ***CPU-Based*** *can be set in a few seconds while* ***GPU-Based*** *requires more time to set up. The main differences are performance, final results and compatibility.* ***GPU-Based*** *is more compatible-sensitive while* ***CPU-Based*** *is less compatible-sensitive.* ***GPU-Based*** *evaluates much realistic results while* ***CPU-Based*** *evaluates less detailed results.*
- Can I edit surface with mouse cursor[pc] or finger[mobile]?
  - Yes, you can.
- What do I need to create quick & nice-looking snow trails?
  - ***Plane*** *[As a surface],* ***MeshTracker_Object*** *[surface component],* ***Track Graphic*** *[created in Track Creator],* ***Starting Canvas*** *[any grayscale image],* ***Sphere*** *[As a track creator],* ***MeshTracker_Track*** *[track source component] and* ***your own creativity****.*
- Does Mesh Tracker work with older Unity versions? [2017 and older]
  - Yes, but probably the API conversion will be required. Also the shaders don't have to work properly.
- Does the Mesh Tracker's water shader support reflections?
  - No, Mesh Tracker water shaders do not support reflections. But can be faked with Reflection Probes.
- Does the Mesh Tracker support all shapes for surface?
  - Yes, but planar objects are highly recommended for best results.
- Do the shapes need to be unwrapped?
  - Yes, they do if you are considering the GPU-Based mesh tracking.
  Use external software like Autodesk 3Ds Max or Blender and use Unwrapper modifier.
- Does the Mesh Tracker work on Unity Terrain?
  - No, Mesh Tracker does not work on built-in Unity Terrain.
- Does the Mesh Tracker support HDRP or URP?
  - No, Mesh Tracker does ***not support HDRP or URP.***
- Does the surface edited with Surface Drawing has Mesh Collider?
  - No, as the surface is generated via material - shader of grayscale textures

FAQ (Frequently asked questions)

- Is Mesh Tracker ready for huge scenes?
  - Yes, the Mesh Tracker is ready for massive scenes & complex meshes. However, for the very expansive landscapes and surfaces, it would be a bit complicated as the Mesh Tracker supports max 8K textures.
- Is Mesh Tracker compatible with Web-GL?
  *- Yes, however the **GPU-Based** is not compatible with Webgl as it doesn't support **Vulkan** GPU-API.*

Warnings

Mesh Tracker v1.4

- *Not tested in Unity 5 (and less)*
- *Not tested on Consoles (Xbox, PS, Nintendo…)*
- *Not tested on Linux operating system*
- ***GPU-Based** mesh tracking doesn't work in Webgl (**CPU-Based** only)*
- ***GPU-Based** mesh tracking **does work** on Android & iOS devices with **Vulkan** GPU API only!*

- *Tested on all **PC-VR platforms** (Vive, Oculus) = works both CPU & GPU based systems*
- *Tested on **iOS** (iPhone XR) & **Android** (Samsung Galaxy A41) = works both CPU & GPU (Vulkan) based systems*
- *Tested on **GPU Nvidia 970 and higher** = works both CPU & GPU based systems*
- *Tested on **Oculus Quest 2** = works with both **CPU-Based** system & **GPU-Based** system (Vulkan)*


THANK YOU FOR YOUR ATTENTION
I hope the documentation helped you a little bit!
Join to my official discord channel to stay in touch!
[Just click the image below]
DISCORD
If you would like to reach me out anyway, contact me **HERE**.
MESH
TRACKER
Now available