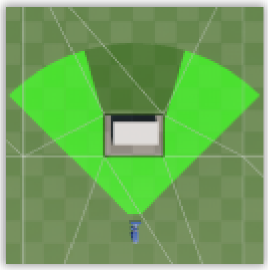
NavMesh Matched Field of Vision Renderer Documentation (v1.0)

By LogerLaw



1. Introduction

NavMesh Matched Field of Vision Renderer (*NMFOVRenderer*) plugin is an easy-to-use tool which can help you to render field of vision with several render modes and highly customizable settings. With this plugin you can easily and quickly set a field of vision in stealth games.

ATTENTION: This plugin is only a renderer which doesn't have sight detecting function, using Line Trace or Overlap Events is more accurate with lower price.

Features

- Render field of vision easily and quickly.
- Highly customizable settings.
- Can render on all surfaces or only on walkable area of navmesh.
- Height prediction function.
- Can export walkable area of navmesh to .obj file.
- Separated from your actors - you don't need to add any component to actors and this plugin will work fine.

2. How to Use

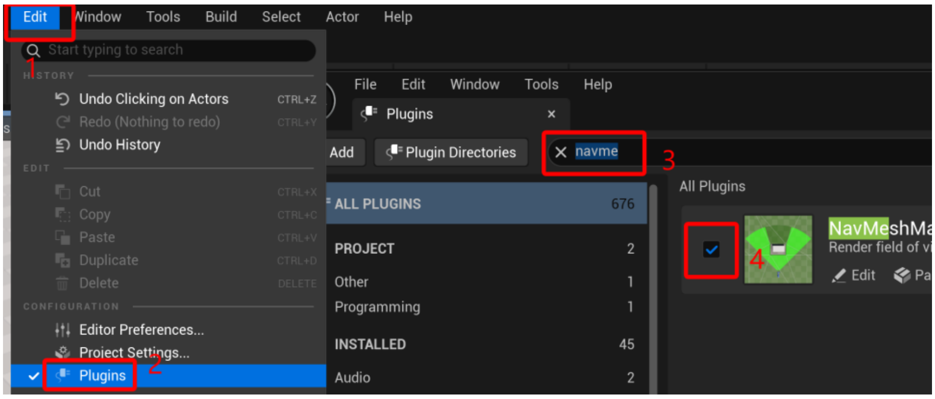
2.1. Initialization

2.1.1. Install the plugin

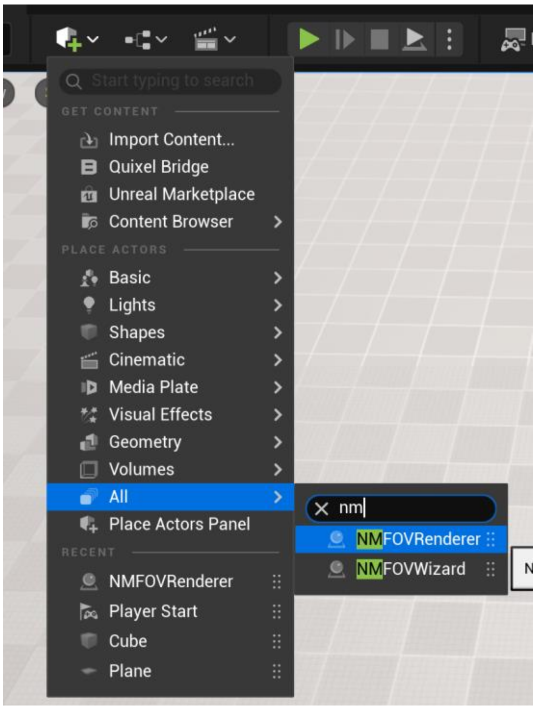
Use Unreal Launcher to install the plugin to your engine, or copy the plugin files to "Plugins" folder in your project folder.

2.1.2. Enable the plugin in your project

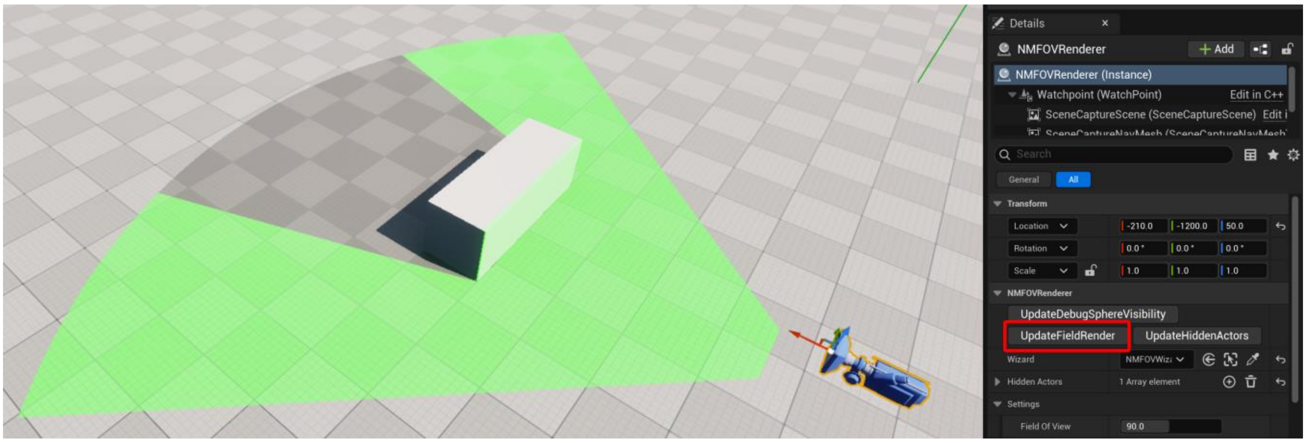
Open your project, click “Edit - Plugins” to open the Plugins panel, search to find “NavMeshMatchedFieldOfVisionRenderer” and enable it, then restart the editor.



2.1.3. Add NMFOVRenderer actor to your map



Place the actor to the proper location and it’s done. You can click the “UpdateFieldRender” button in Details panel to see the result. What you should do next is setting its parameters and call its functions in blueprint or c++ to meet your needs.



Location: -210.0 | -1200.0 | 50.0

Rotation: 0.0 ° | 0.0 ° | 0.0 °

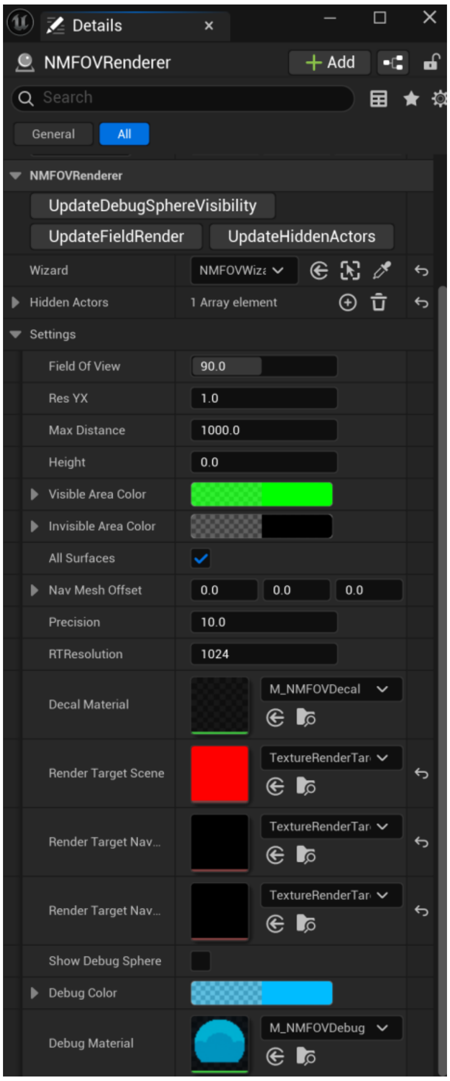
Scale: 1.0 | 1.0 | 1.0

Wizard: NMFOVWiz

Hidden Actors: 1 Array element

Field Of View: 90.0

2.2. Parameters Setting



Wizard: NMFOVWiz

Hidden Actors: 1 Array element

Field Of View: 90.0

Res YX: 1.0

Max Distance: 1000.0

Height: 0.0

Visible Area Color: 🟩

Invisible Area Color: ■

All Surfaces: ☑

Nav Mesh Offset: 0.0 | 0.0 | 0.0

Precision: 10.0

RTResolution: 1024

Decal Material: M_NMFOVDecal

Render Target Scene: TextureRenderTar

Render Target Nav...: TextureRenderTar

Render Target Nav...: TextureRenderTar

Show Debug Sphere: □

Debug Color: 🟦

Debug Material: M_NMFOVDebug

Wizard: It will be created automatically. See Advanced Chapter for more information.

Hidden Actors: Set actors which you want to ignore, then click “UpdateHiddenActors” button or call “UpdateHiddenActors” function in blueprint to submit the change.

Field Of View: As its name suggests.

Res YX: Pitch = FieldOfView * ResYX, it should be less than 180.

Max Distance: How far the actor can see.

Height: Predicted height. About height prediction function, see Advanced Chapter.

Visible Area Color, **Invisible Area Color**: As its name suggests.

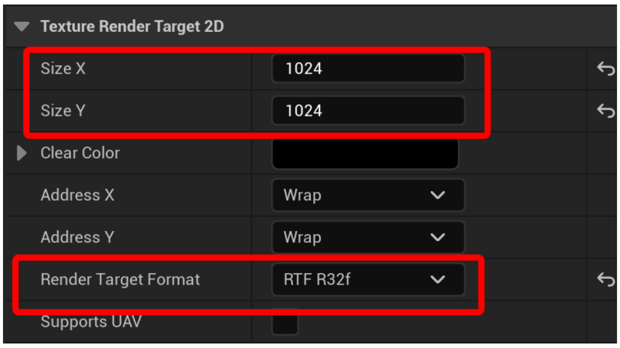
All surfaces: If this value is true, the decal will be rendered on all surfaces, while if it’s false, the decal will only be rendered on walkable area of navmesh, which means there should be a navmesh in your map or it will render nothing. Related unreal engine feature.

NavMesh Offset, **Precision**: See Errors Chapter for detailed information.

RTResolution: If you don't set RenderTarget in RenderTargetScene or RenderTargetNavMesh or RenderTargetNavMeshWithHeight, or current RenderTarget's SizeX isn't equal to this value, it will create new RenderTarget with this resolution.

Decal Material: You can customize it, or ignore it. See Advanced Chapter for more information.

RenderTarget Scene, **RenderTarget NavMesh**, **RenderTarget NavMesh With Height**: If you have special needs you can set one, or it will create one automatically. If you use your own RenderTarget, make sure ① SizeX = RTResolution ② SizeX / SizeY = ResYX ③ RenderTargetFormat = RTF_R32f.



Texture Render Target 2D

Size X 1024

Size Y 1024

Clear Color

Address X Wrap

Address Y Wrap

Render Target Format RTF R32f

Supports UAV

Show Debug Sphere: As its name suggests.

Debug Color: The color of DebugSphere.

Debug Material: If you have special needs you can customize it.

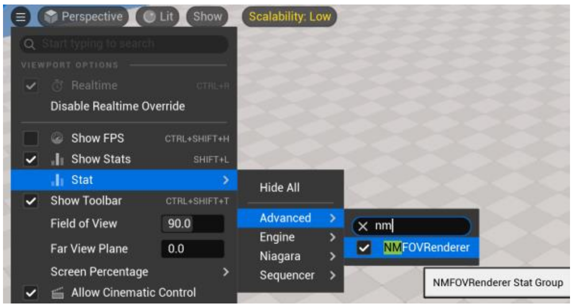
All parameters will be used in the next ”UpdateRenderField” call except Hidden Actors. If you want to submit changes of Hidden Actors, click “UpdateHiddenActors” button or call “UpdateHiddenActors” function in blueprint. If you want to change the visibility of Debug Sphere without calling ”UpdateRenderField”, you can click “UpdateDebugSphereVisibility” button or call “UpdateDebugSphereVisibility” function in blueprint.

2.3. Advanced

2.3.1. Performance

To save performance this plugin won’t tick, call “UpdateRenderField” to update. In shipping game I suggest that the updating rate should be less than 60 per second, and if you put more than one NMFOVRenderer in your game, it would be better to update them in different ticks.

Also, you can open the Stat or GPU Visualizer to see the cost.

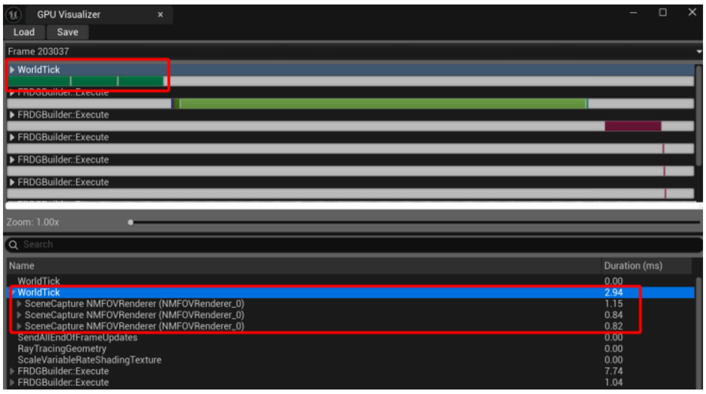


Field of View 90.0

Far View Plane 0.0

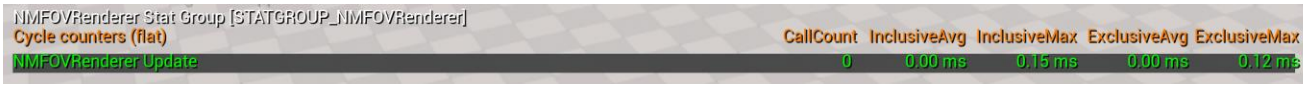
nm

☑ NMFOVRenderer



```
Frame 203037
WorldTick
FRDGBuilder::Execute
FRDGBuilder::Execute
FRDGBuilder::Execute
FRDGBuilder::Execute
FRDGBuilder::Execute

Name                                                    Duration (ms)
WorldTick                                               0.00
WorldTick                                               2.94
  SceneCapture NMFOVRenderer (NMFOVRenderer_0)          1.15
  SceneCapture NMFOVRenderer (NMFOVRenderer_0)          0.84
  SceneCapture NMFOVRenderer (NMFOVRenderer_0)          0.82
SendAllEndOfFrameUpdates                                0.00
RayTracingGeometry                                      0.00
ScaleVariableRateShadingTexture                         0.00
FRDGBuilder::Execute                                    7.74
FRDGBuilder::Execute                                    1.04
```



```
NMFOVRenderer Stat Group [STATGROUP_NMFOVRenderer]
Cycle counters (flat)                CallCount  InclusiveAvg  InclusiveMax  ExclusiveAvg  ExclusiveMax
NMFOVRenderer Update                         0       0.00 ms       0.15 ms       0.00 ms       0.12 ms
```

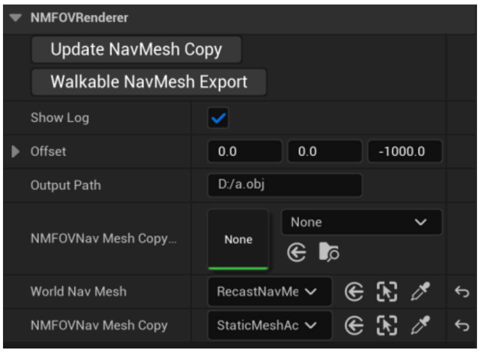
2.3.2. NMFOVWizard

Related unreal engine feature.

If there is no NMFOVWizard in your map, once you call “UpdateRenderField” it will create one automatically, but you can also add one by yourself. This actor manages a copy of walkable navmesh, so when the navmesh in your map changes, click “UpdateNavMeshCopy” button or call “UpdateNavMeshCopy” function in blueprint to update it. This function calls an engine function which will produce memory garbage, so try to use it as less as you can in your shipping game.

Set the “Offset” parameter to where the navmesh copy wouldn’t affect your gameplay. If you change the Offset value, call “UpdateNavMeshCopy” to reposition the copy. Better not change the copy’s transform manually.

Information about other parameters and functions see Additional Functions Chapter. If you don’t use those functions, you can just ignore them.



Show Log ☑

Offset 0.0 0.0 -1000.0

Output Path D:/a.obj

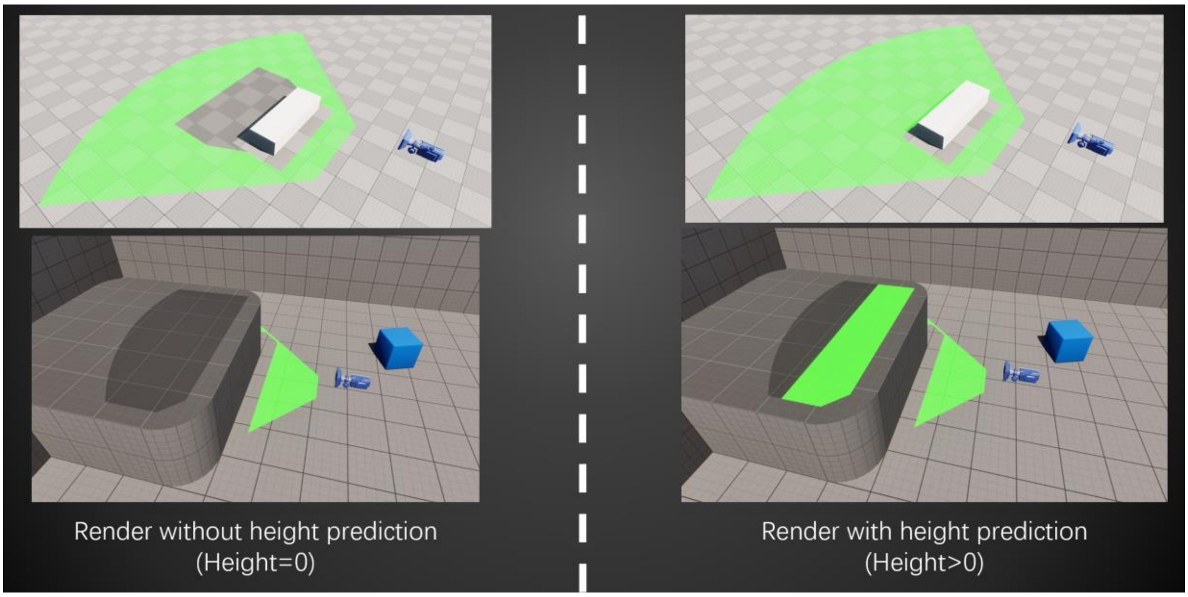
NMFOVNav Mesh Copy... None None

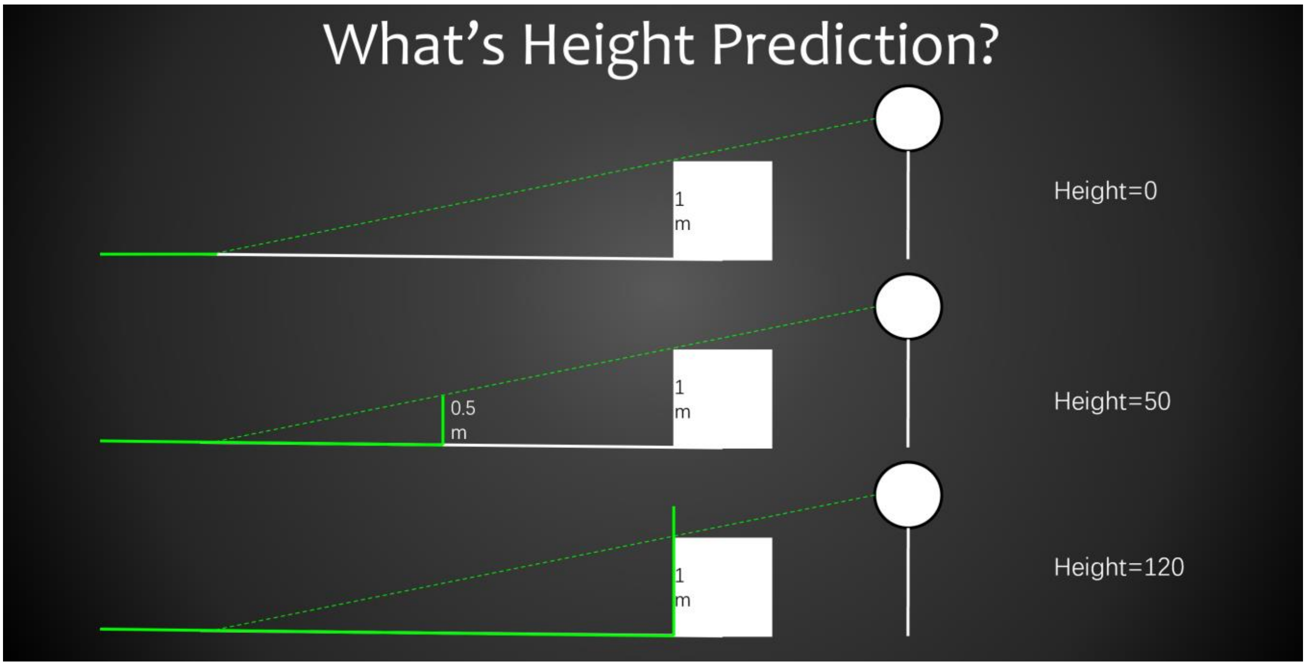
World Nav Mesh RecastNavMe

NMFOVNav Mesh Copy StaticMeshAc

2.3.3. Height Prediction

Height Prediction will render not only the visible area, but also the area where an actor with the value of Height can be seen. These two pictures show the differences.


Render without height prediction
(Height=0)
Render with height prediction
(Height>0)

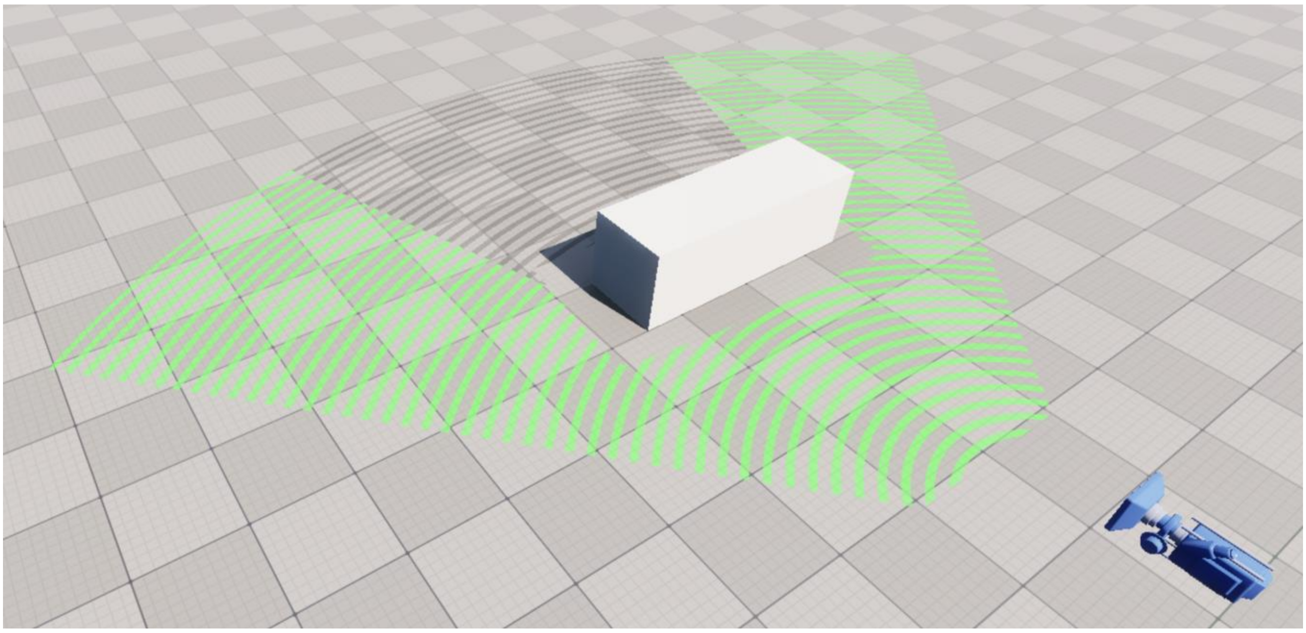

What's Height Prediction?

To decide whether a point can be seen, the plugin will check two points' visibility. For example, to decide whether point A with location (x,y,z) can be seen, the plugin will check visibility of (x,y,z) and (x,y,z+Height). If any one of these two can be seen, surface with point A's location will be visible color. Notice that if you set the Height value too large that (x,y,z+Height) is out of the sight range while (x,y,z) can't be seen, surface with that location will still be invisible color even if (x,y,z+Height/2) can be seen.

2.3.4. Customize Decal Material

Related unreal engine feature. Though this plugin uses decal component instead of decal actor, the settings are the same.

A decal material is automatically set, but you can also set your own customized material. In plugin's Content folder you can find "Materials/M_NMFOVDecal_StripExample", which is an example material showing how to customize it.

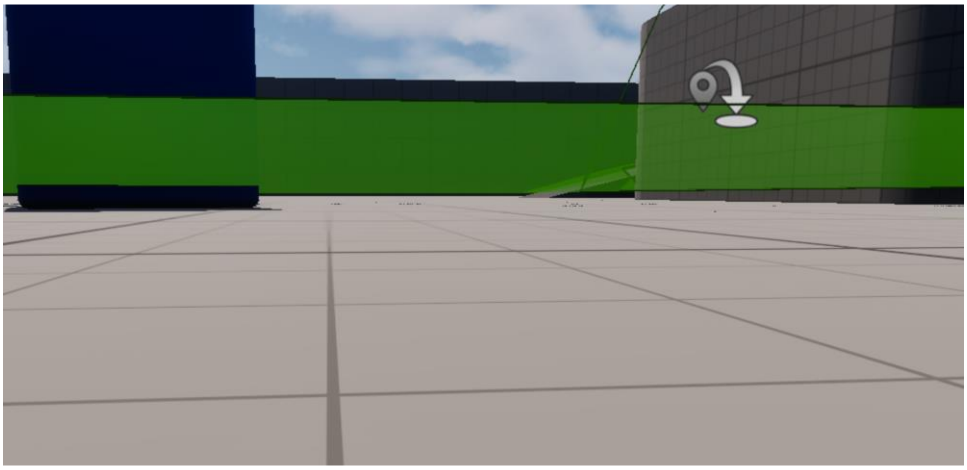


2.4. Errors

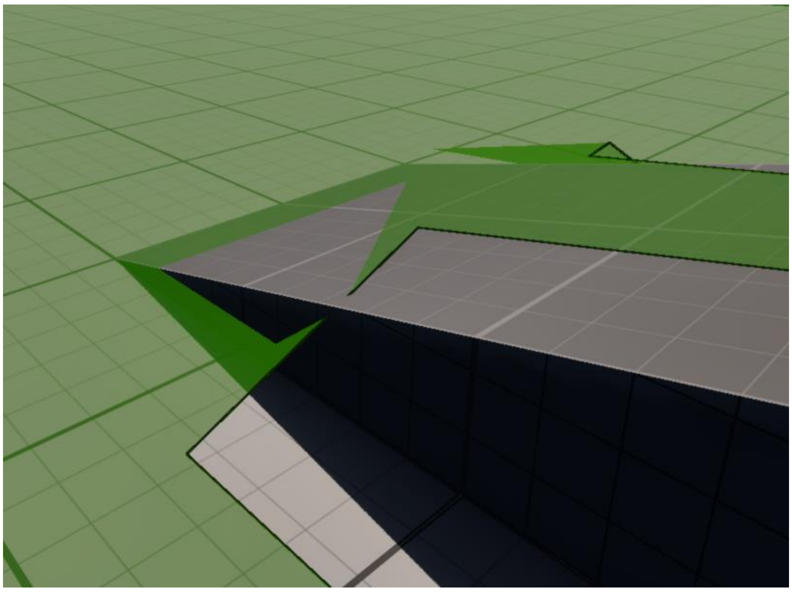
I will use Top Down engine template to explain this topic. Open TopDownMap then press key P in the viewport, the navmesh will show up, then find "RecastNavMesh-Default" in Outliner and set Draw Offset to 0, you can see the navmesh is still not matched with the real ground surface. Here are the two main errors:

1. The navmesh has a little offset from the ground, that's not controllable and many people in the forum believe it's a bug. You can adjust the Z value of RecastNavMesh-Default's location

until they match well, that’s the offset, then undo your change and set the offset value to NavMesh Offset parameter in NMFOVRenderer actor. For example, the navmesh in TopDownMap is 10 units higher than ground, so I’ll set NavMesh Offset with (0,0,10).



2. Navmesh will simplify its shape on some complicated areas, so the plugin can’t render precisely on these places when it renders only on walkable area of navmesh. On that case, you can increase the Precision parameter. But if the navmesh is too complicated that even if increasing Precision still results in a mess, I recommend you choose All Surfaces mode to render on all surfaces.

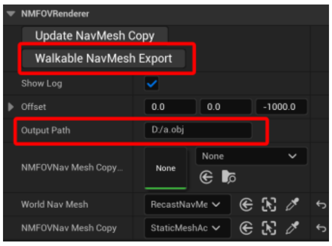


And if you find the decal with large jags, you can increase RenderTargets’ resolutions.

3. Additional Functions

3.1. Walkable Area of NavMesh Exporter

Add NMFOVWizard actor to your map, set OutputPath then click “WalkableNavMeshExport” button, an .obj file of walkable area of navmesh will be created to the path.



Show Log ✓

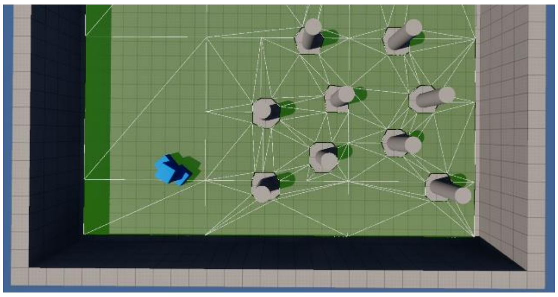
Offset 0.0 0.0 -1000.0

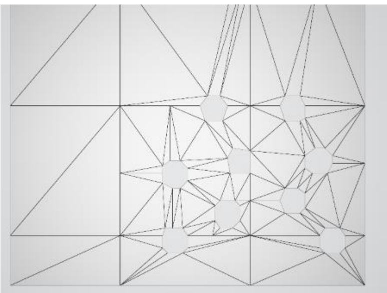
Output Path D:/a.obj

NMFOVNav Mesh Copy... None None

World Nav Mesh RecastNavMe

NMFOVNav Mesh Copy StaticMeshAc

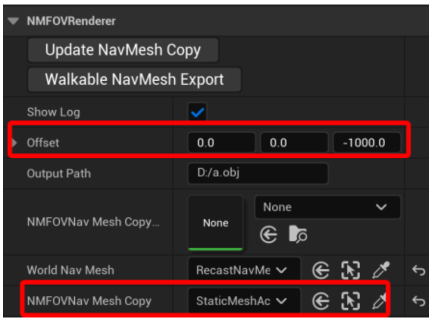




3.2. Customize Render Area

If you want to do some changes to the area which can be rendered in “Only NavMesh” mode, for example, you want to render on a platform but there is no navmesh on it, or exclude an area from the navmesh to make it not renderable, there are several ways to make it.

The easiest and most recommended method is using exporter to export navmesh to .obj file and change it in other model softwares. Then import the changed model into your project, set it as a StaticMeshActor’s StaticMesh, put the actor at the Offset location, finally set the StaticMeshActor as NMFOVNavMeshCopy. Don’t call “UpdateNavMeshCopy” or it will be overrided.



Show Log ✓

Offset 0.0 0.0 -1000.0

Output Path D:/a.obj

NMFOVNav Mesh Copy... None None

World Nav Mesh RecastNavMe

NMFOVNav Mesh Copy StaticMeshAc

Also you can customize decal material to make it.

4. Bug Report

logerlaw@gmail.com