



Lessons from a plugin developer

How to extend Unreal when you are not Epic



Epic can edit Engine and Plugins



Epic can edit Engine and Plugins

Games can edit Engine, Plugins and Game



Epic can edit Engine and Plugins

Games can edit Engine, Plugins and Game

Plugins can edit ... Plugins



Extending Unreal without changing the engine

Before we start

Slides are available online

QR code will be displayed at the end



Before we start

Who am I?

Victor Careil

Created Voxel Plugin in 2017





Before we start

Voxel Plugin

Volumetric landscape replacement with full runtime support

Voxel Plugin 1 released in 2018

3 people full time on Voxel Plugin 2 since 2021

Released planned later this year, already available in beta



Non-destructive edits



Runtime Sculpting



Fully Volumetric

Runtime Nanite



Nanite Tessellation



Runtime Lumen



Custom Graphs

File Edit Window Tools Build Select Actor Help	File Edit Graph Asset Window Tools Help	- • ×
🖹 🧓 ६ Selection Mode 🗸 📢 📲 🛫 🕨 🕨 🖿 📥 🗄 🛃 Platforms 🗸	🖺 👼 🕍 Compile 🕍 Compile All 📲 Enable Stats 🚍 Enable Range Stats	📽 Preview 💦 Graph
E Perspective CLt Show	Q J* Main Graph × V 00A V Image: State Base 300 Peters Value Image: State Base 300 J* Image: State Base 300 Peters Value Image: State Base 300 Peters Value Image: State Base 300 V Image: State Base 300 Image: State Base 300 Peters Value Image: State Base 300 V Image: State Base 300 Image: State Base 300 Image: State Base 300 V Image: State Base 300 Image: State Base 300 Image: State 300 V Image: State 300 Image: State 300 Image: State 300 V Image: State 300 Image: State 300 Image: State 300	-2 -2
Je -	Convertised Notes 20 Control from To Andra Value T	Press it to preview the solucion mode 1 on -1.0 m Oppits 0.00 m Max -1.0 m 1.0 m
	VOXE	Details × Z Preview Sett. Details
	🛱 Messages × Q. Find Results	Octave Strength 0 Array elements 🕑 🖞
	🛱 Content Drawer 📓 Output Log 💽 Cmd 🗸 Enter Console Command	2 Unsaved P Revision Control V
To Content Drawer Dutput Log D Cmd V Enter Console Command	💭 Trace 🗸 💿 🙆 📑 Derived Data 🖌 🔒	👔 🗄 2 Unsaved 🛛 💱 Revision Control 🗸

Full PCG support

E Perspective Lt Show	
	the second se
A CALL AND A	
	1 Ukr
	File Edit Asset Window Tools Help - X
	Cancel Evecution
	Q Search Zoom 1:1 a Q See a Q
	Bueprint Bupo.
	Execute Blue Conrg. P
	Branch Select Points Static Mesh Spawner Vesc. P
	Runtime Qua 🛁 🦣 In Out 🔋 🛶 👘 In Out 📳 Land. 🗸
	Select (Multi) Use. 9
	Witch Hote
	Debug Ion I Nodes O D
	Print String Static Mesh Spawner Input.
	Visualize Attu -1 in Out
	Curve Remap Despire Remap Despire Remap Despire Remap Despire Remap Rem Remap Remap Rem
	Distance to C
	Attribute Filte PCG Graph The O
	Attribute Filte
	C Do a T No pins ✓ No data available ✓ Q. Search
	PCGRuntimeGe
	CGRuntimeGe PCGRuntimeGe
	M PCGRuntimeGe M PCGRuntimeGe M PCGRuntimeGe
	PCGRuntimeGe
	E Contest Drawer
	ap content of any and a second a se
	the second s

Indirect Materials



All of this is done without changing the engine

Agenda

- Customizing editor UI
- Speeding up collision cooking
- Hacking the Nanite visibility buffer

Customizing editor UI

Tip 1

Property metadata

Easiest way to customize a property

Most commonly used:

- EditCondition
- InlineEditConditionToggle
- EditConditionHides
- UIMin/Max, ClampMin/Max
- DisplayName

UPROPERTY(EditAnywhere, Category = "Misc", meta = (InlineEditConditionToggle))
bool bEnableDisplacementFade = false;

UPROPERTY(EditAnywhere, Category = "Misc", meta = (EditCondition = "bEnableDisplacementFade"))
FDisplacementFadeRange DisplacementFade;

Quality Exponent

UPROPERTY(EditAnywhere, Category = "Config", meta = (UIMin = 0.5, UIMax = 1.5))
double QualityExponent = 1.f;

1.0

unreal-garden.com

unreal 📭 garden		Tutorials Bits Docs ?	Q,
		UPROPERTY UFUNCTION UCLASS USTRUCT UINTERFACE UPARAM UENUM and	UMETA
General Follis			
Animation			
Asset Registry		Fully Expand	
Assets			Correct Correct
Blueprint	Meta	UPROPERTY(meta=(FullyExpand=true))	ц Сору
Blueprint Logic	Type:		
C++			
Config Files		Ealt Condition	
Console	Position: Meta	UPROPERTY(meta=(EditCondition="abc"))	🗘 Сору
Data Decorators	Type:		
Data Table	string (Conditional statement)	Editcondition can be used to change a property from read-only to writeable depending on an-	
Editor		other property.	
Materials	***	ments are supported.	
Network	Related:	It is worth noting that Editcondition also changes the appearance of properties inside	
Pickers		Blueprint logic Make Struct nodes.	
Scripting		Note that any variable used in the EditCondition needs to also be a UPROPERTY.	
Serialization		UPROPERTY(EditAnywhere)	
UMG		bool bCanFly;	
Todo		<pre>UPROPERTY(EditAnywhere, meta=(EditCondition="bCanFly"))</pre>	
Internal and Deprecated		float MaxFlightSpeed;	
		UENUM() enum class EAnimalType { Bird, Fish	

Tip 2

Property customization

For advanced changes to property UI

IPropertyTypeCustomization:

- For structs
- Usually to customize the value UI

IDetailCustomization:

- For objects
- Usually to add buttons, add/remove properties etc

My Seed	WWPDGLJT
▼ Config	
Bake Metadatas	Bake

```
USTRUCT(BlueprintType)
> struct FSeed
{
    GENERATED_BODY()
    UPROPERTY(BlueprintReadWrite, EditAnywhere, Category = "Config")
    FString Seed;
};
```

 My Seed 		My Seed	WWPDGLJT 🔯
Seed	WWPDGLJT		
			↓ · · ·

Randomize on click

To customize the property header

To customize the property children

void FS	GeedCustomization::CustomizeHeader(
con	st TSharedRef <ipropertyhandle> PropertyHandle,</ipropertyhandle>		
FDe	tailWidgetRow& HeaderRow,		
IPr	<pre>opertyTypeCustomizationUtils& CustomizationUtils)</pre>		
con	st TSharedPtr <ipropertyhandle> SeedHandle = Property</ipropertyhandle>	<pre>yHandle->GetChildHandle(GET_MEMBER_NAME_CHECKED(ClassName: FSeed, MemberName: Seed));</pre>	
Hea	IderRow		
.Na	meContent()		
[4	^		
	PropertyHandle->CreatePropertyNameWidget()		
]			
Custo	Customize the name widget		
		Create a name widget using the property name	
💌 Defa	ault Value		
Mv s	Seed XKJZZZNN 🕅		
iviy (

void	FSeedCustomization::CustomizeHeader(const TSharedRef <ipropertyhandle> PropertyHandle, FDetailWidgetRow& HeaderRow, IPropertyTypeCustomizationUtils& CustomizationUtils)</ipropertyhandle>
	<pre>const TSharedPtr<ipropertyhandle> SeedHandle = PropertyHandle->GetChildHandle(GET_MEMBER_NAME_CHECKED(ClassName: FSeed, MemberName: Seed));</ipropertyhandle></pre>
	HeaderRow
	.NameContent()
	[
	PropertyHandle->CreatePropertyNameWidget()
]
	.ValueContent()

Customize the value widget


```
.ValueContent()
   SNew(WidgetType: SHorizontalBox) 
Make a new horizontal layout
   + SHorizontalBox::Slot()
                              Make the text widget as wide as possible
   .FillWidth(1.f)
       SeedHandle->CreatePropertyValueWidget() 
                                                   Create a value widget for our Seed string property
   + SHorizontalBox::Slot()
   .AutoWidth()
                              Make the dice button as narrow as possible
   .Padding(Horizontal: 4.f, Vertical: 2.f)
                                                                                          Default Value
       SNew( WidgetType: SBox)
       .WidthOverride(16.f)
                                        Force the button size
                                                                                                                 XKJZZZNN
                                                                                            My Seed
       .HeightOverride(16.f)
           SNew( WidgetType: SButton)
           .ButtonStyle(InSlateStyle: FVoxelEditorStyle::Get(), StyleName: "Graph.Seed.Dice")
           .OnClicked Lambda([SeedHandle] ->FReply
               const FString NewValue = MakeRandomString();
                                                                   Randomize on click
               SeedHandle->SetValue(NewValue);
               return FReply::Handled();
```

We have a customization, we now need to register it

What we do at Voxel Plugin:

DEFINE_STRUCT_LAYOUT(FSeed, FSeedCustomization);

Custom code, will submit a pull request soon

Takeaways

Property customization

Use property metadata if you can

unreal-garden.com is awesome

Customizations are easy to make and can greatly improve workflows

UPROPERTY(EditAnywhere, Category = "Misc", meta = (InlineEditConditionToggle))
bool bEnableDisplacementFade = false;

UPROPERTY(EditAnywhere, Category = "Misc", meta = (EditCondition = "bEnableDisplacementFade"))
FDisplacementFadeRange DisplacementFade;

unreal 📭 garden

Unreal Engine UI Tutorials

My Seed

DETAULT VALUE

WWPDGLJT

Tip 3 Widget reflector

Great tool to debug Slate widgets

Ctrl Shift W in editor

WidgetReflector in console in packaged games

Tip 4

Extending editor menus

Most editor menus can be extended

Makes it easy to add new buttons

Can greatly improve workflows & make a tool feel native

Extending editor menus

If you use Visual Studio: Entrian Source Search is great

Extending editor menus

UToolMenu* ToolBar = UToolMenus::Get()->ExtendMenu(Name "LevelEditor.LevelEditorToolBar.AssetsToolBar");

FToolMenuSection& Section = ToolBar->FindOrAddSection "Content";

```
FToolMenuEntry& NewEntry = Section.AddDynamicEntry(InName: "VoxelPluginGlobalActions", InConstruct: FNewToolMe
```

```
InSection.AddEntry(
Args: FToolMenuEntry::InitWidget(
InName: "VoxelMode",
InWidget: SNew(SCheckBox)
.Style(&FVoxelEditorStyle::GetWidgetStyle<FCheckBoxStyle>(PropertyName: "VoxelModeButton"))
.Padding(FMargin(InLeft: 6.f, InTop: 0.f, InRight: 4.f, InBottom: 0.f))
.IsChecked_Lambda([]->ECheckBoxState
{
    return
        GLevelEditorModeTools().IsModeActive(InID: "VoxelToolEdMode")
        ? ECheckBoxState::Checked
        : ECheckBoxState::Unchecked;
})
```
Extending editor menus

ToolMenus.Edit 1



8 🛛 🕂		Edit	Menu				
LevelE	LevelEditor.LevelEditorToolBar.AssetsToolBar						
Level	Editor.LevelEditorToolBar.AssetsToolBa	ar			~		
			Undo all changes	Reset all to Defaults	Reset to Defaults		
() () () () () () () () () () () () () (CONTENT	Q Search			□ □ □		
•	Voxel 🔹 Add	Name	LevelEditor.	LevelEditorToolBar.AssetsToolBar			
•	• Blueprints Cinematics						
() ()	🌱 VR Mode SHIFT+V >						

Sending notifications

Great way to send non-blocking messages

For buttons: make sure VisibleInState matches the actual state of the notification!



FNotificationInfo Info(INVTEXT(InTextLiteral: "My Title")); Info.SubText = INVTEXT(InTextLiteral: "This is a subtext"); Info.bFireAndForget = true; Info.ExpireDuration = 10.0f; Info.bUseSuccessFailIcons = false; Info.ButtonDetails.Add(Item: FNotificationButtonInfo(INVTEXT(InTextLiteral: "My Button"), InToolTip: INVTEXT(InTextLiteral: "My Tooltip"), InCallback: FSimpleDelegate(), VisibleInState: SNotificationItem::CS None));

FSlateNotificationManager::Get().AddNotification(Info);

Custom thumbnails

Critical to navigate large number of assets

Many examples in engine: eg UStaticMeshThumbnailRenderer

Try using TSharedPtr or TUniquePtr instead of a manual delete (don't follow the engine patterns)

🚽 🗸 📿 Search Content atic VSM_Canyon VSM_Canyon VSM_Canyon VSM_Canyon VSM_Canvon VSM_Canyon Rock3 Bock7 Rock10 Rock9_Voxelized Rock1 Rock4 Rock9 ➡ Q Search Content VSM_Canyon VSM_Canyon VSM_Canyon Rock9 Voxelized Rock1 Rock3 Rock4 Bock7 Rock9 Rock10 UCLASS() v class UVoxelSurfaceAssetThumbnailRenderer : public UDefaultSizedThumbnailRenderer GENERATED BODY() //~ Begin UDefaultSizedThumbnailRenderer Interface virtual void BeginDestroy() override; virtual void Draw(UObject* Object, int32 X, int32 Y, uint32 Width, uint32 Height, FRenderTa virtual bool CanVisualizeAsset(UObject* Object) override; //~ End UDefaultSizedThumbnailRenderer Interface TSharedPtr<FMaterialThumbnailScene> ThumbnailScene; UThumbnailManager::Get().RegisterCustomRenderer(Class: UVoxelSurfaceTypeAsset::StaticClass(),

UVoxelSurfaceAssetThumbnailRenderer::StaticClass());

Custom toolkits

Custom UI when you open a custom asset

Engine framework is somewhat complex, so we made our own open-source wrapper for it

Can create a new toolkit in 50 lines

https://github.com/VoxelPlugin/VoxelCore#custom-toolkit



USTRUCT() struct FVoxelStaticMeshToolkit : public FVoxelSimpleAssetToolkit

GENERATED_BODY() GENERATED_VIRTUAL_STRUCT_BODY()

UPROPERTY()
TObjectPtr<UVoxelStaticMesh> Asset;

public

//~ Begin FlowelSimpleAssetToolkit Interface
virtual void Tick() override;
virtual void SetupPreview() override { return false; }
virtual void SetupPreview() override;
virtual void SetupPreview() override;
virtual void Oraw(const FSceneViaw* View, FPrimitiveDrawInterface* PDI) override;
virtual void PopulatFoonSlar(const TSharedRefSAFICTSonlBarDex, const TSharedPrcSViewportToolBar>& ParentToolBarPtr) override;
virtual void PopulatFoonSlar(const FPropertyChangedEvent& PropertyChangedEvent) override;
virtual void PostIditChange(const FPropertyChangedEvent& PropertyChangedEvent) override;
virtual void PostIditChange(const FPropertyChangedEvent& PropertyChangedEvent) override;
virtual void PostIditChange(const FPropertyChangedEvent& PropertyChangedEvent) override;

private

UPROPERTY()
TObjectPtr<AVoxelWorld> VoxelWorld;

UPROPERTY() TObjectPtr<AVoxelStampActor> StampActor;

Custom graphs

Making domain-specific graphs can be of great help

Plenty of examples on GitHub

https://github.com/MothCocoon/FlowGraph https://github.com/jinyuliao/GenericGraph



Fixing linking errors

In editor, modules are compiled as DLLs

Symbols from other modules need to be exported!

This is not the case in packaged games (monolithic builds): single exe, no DLLs

unresolved external symbol "public: static class FShaderParametersMetadata const * __cdecl FNaniteShadingUniformParameters::GetStructMetadata(void)" (? GetStructMetadata@FNaniteShadingUniformParameters@@SAPEBVFShaderParametersMetadata@@XZ) referenced in function "public: __cdecl TUniformBufferRef<class FNaniteShadingUniformParameters>::TUniformBufferRef<class FNaniteShadingUniformParameters>(class FRHIUniformBuffer*)" (??0? \$TUniformBufferRef@VFNaniteShadingUniformParameters@@@@QEAA@PEAVFRHIUniformBuffer@@@@Z)

😣 LNK2019



Fixing linking errors



Accessing private members

Solution 1: we can be friend (rare)

Solution 2: #define private public

Can create linking errors with MSVC

Solution 3: C++ template tricks

Explicit template specialization arguments can bypass private restrictions

Doesn't work for constructors

```
struct FMyStruct
{
private:
    int32 MyValue = 0;
```

friend class FSomeClassWeCanPretendToBe;
};

ø ...

≙

357



Q 80

@embyr.bsky.social @EMBYRdev

I can't decide how to feel about this...

	<pre>#include "ContentBrowserModule.h"</pre>
	<pre>#include "EditorViewportClient.h"</pre>
	<pre>#include "DetailCategoryBuilderImpl.h"</pre>
	#define private public
	<pre>#include "Editor/PropertyEditor/Private/DetailPropertyRow.h"</pre>
11	#undef private
12	
13	<pre>DEFINE_VOXEL_INSTANCE_COUNTER(FVoxelDetailCustomization);</pre>
14	<pre>DEFINE_VOXEL_INSTANCE_COUNTER(FVoxelPropertyTypeCustomization);</pre>

💙 2.2K

6:41 AM · Apr 4, 2024 · 244.7K Views

tl 209

Accessing private members

```
struct FMyStruct
{
private:
    int32 MyValue = 0;
};
DEFINE_PRIVATE_ACCESS(FMyStruct, MyValue);
```

```
void Test()
```

```
FMyStruct MyStruct;
PrivateAccess::MyValue([&] MyStruct) = 1;
```

```
namespace PrivateAccess
{
    template<auto PropertyPtr>
    struct TAccessor
    {
        friend auto& MyValue(FMyStruct& Object)
        {
            return Object.*PropertyPtr;
        };
        template struct TAccessor<&FMyStruct::MyValue>;
        auto& MyValue(FMyStruct& Object);
    }
}
```

DEFINE_PRIVATE_ACCESS_FUNCTION(FMyStruct, MyFunction);

UE_DEFINE_PRIVATE_MEMBER_PTR(int32, GMyValuePtr, FMyStruct, MyValue);

https://github.com/VoxelPlugin/VoxelCore/blob/master/Source/VoxelCore/Public/VoxelMinimal/VoxelMacros.h#L1244

Takeaways

Customizing editor UI

You can customize pretty much every part of the editor UI

If you don't know where to start:

1. Look for an existing engine feature doing what you want to do

2. Use the Widget Reflector or tooltip search to find the relevant code







Speeding up collision cooking

Problem

We cook collision for a lot of chunks onthe-fly, at runtime

Collision cooking can be slow for runtime use

How to make it **faster** without changing the engine?



Problem



Problem

template <typename QueryGeomType>
friend struct FTriangleMeshOverlapVisitorNoMTD;



We are a friend of FTriangleMeshImplicitObject

We can now access its private properties and do custom cooking!

Speeding up collision cooking

5.5x faster!

Code is open-source

github.com/VoxelPlugin/VoxelCore/blob/master/Source/VoxelCore/Public/VoxelChaosTriangleMeshCooker.h

2	V	/oxel	State 245 (325.1 µs)	Vo
М	F	Voxe	elCollider::Create (324.1 μs)	FV
	F	Vox	FVoxelChaosTriangleMeshCooker::Create (301.9 µs)	F٧
	Chaos::FTriangleMeshOverlapVisitorNoMTD::CookTriangleMesh Num=1089 (301.4 µs)			F٧
	bu Slow cook (287.4 μs)			
	AABBTreeGenerateTree (219.9 µs)			
			AABBTreeTi AABBTreeInitialTimeSlice (173.1 µs)	







Hacking the Nanite visibility buffer





Tessellated plane



Different materials

Smooth blends

Overview



Step 1 What is Nanite

Nanite is a virtualized geometry renderer with automatic LODs

Renders into a visibility buffer which is then shaded using compute shaders



How Nanite renders a frame



Idea



Idea



Step 2

Where to hook?

Scene View Extensions: easy way to hook into the rendering pipeline

Can hook before and after BasePass & PostProcess

Here we hook into PreRenderBasePass

class FMySceneViewExtension : public FSceneViewExtensionBase
{
 public:
 using FSceneViewExtensionBase::FSceneViewExtensionBase;
 //~ Begin FSceneViewExtensionBase Interface
 virtual void PreRenderBasePass_RenderThread FRDGBuilder& GraphBuilder) override;
 //~ End FSceneViewExtensionBase Interface



Step 3

Adding a Compute Shader

Unreal has 3 main shader types:

FGlobalShader: pure HLSL shader

FMaterialShader: HLSL shader merged with a material graph

FMeshMaterialShader: material shader with a vertex factory

FGlobalShader

FAmbientOcclusionCS FBloomDownsampleKernelCS FScreenSpaceReflectionsCS

FMaterialShader

FPostProcessMaterialShader FDeferredDecalPS FSlateMaterialShaderPS

FMeshMaterialShader TBasePassCS FVelocityPS FLumenCardPS

Adding a Compute Shader

We want to select materials based on displacement

Displacement is set in artist-authored materials

We need to make a Material Shader







#include /Generated/Material.ush

Step 4 Making a Mega Material

We have many different materials

We need to generate one compute shader

Our solution: generate a Mega Material

Proper branching between materials

Used for displacement, Lumen, RVTs, mobile...



Step 5

Extending Engine Shaders

Plugins cannot edit engine source as engine C++ cannot be recompiled

But: any shader can be recompiled anytime, including engine ones

Plugins could edit engine shaders!



Shader Hooks

GUID & diffing based

Editor UI to apply or remove Shader Hooks

We can edit any engine shader!

github.com/VoxelPlugin/VoxelCore/blob/master/Source/VoxelCore/Public/VoxelShaderHook.h

ADD_VOXEL_SHADER_HOOK(FVoxelMaterialHook, "6F912F1438B246498E: "/Engine/Private/Nag R"(18FDD90C717824", ◀ nite/NaniteRasterizationCommon.u	ush", 4		
<pre>N VertexParameters = MakeInitializedMaterialVertexParameters(); SetVertexParameterInstanceData(VertexParameters, InstanceData, Primi SetVertexParameterAttributeData(VertexParameters, InputVert, InstanceData)</pre>				
), R"(##endif })", "", P"(
<pre>##if MATERIAL_PIXEL_PARAMETERS_VOXEL_VERSION == 7 VertexParameters.Voxel_PageIndex = VisibleCluster.PageIndex; VertexParameters.Voxel_ClusterIndex = VisibleCluster.ClusterIndex; VertexParameters.Voxel_VertexIndex = InputVert.VertIndex; ##endif</pre>				
73				
▼ Shader Hooks				
Engine shader changes required by Vo:	Engine shader changes required by Voxel Plugin features			
Name	Description	State		
Lights HLSL packaging fix	This hook fixes HLSL lights packaging errors.	Up to date C Apply Remove		
Voxel Material	This is required to render voxel materials	Up to date C Apply Remove		

Step 6

Getting the VisBuffer

The Visibility Buffer is not exposed!

Game Over? Usually, yes

But FRDGBuilder stores a list of all the render textures, along with their debug name!











Step 7

Smooth blends

Use dithering to randomly switch between materials

Weighted random selection: weight is set by the height blend logic

We still render only one material per pixel



Takeaways

Hacking the Visibility Buffer

You can easily make material shaders in a plugin, giving artists full control

You can hook into many parts of the render pipeline

You can access most render textures with FVoxelUtilities::FindTexture



Open-source sample

← → C 😄 github.com/Phyr				
E Phyronnaz / MaterialS	ShaderExample equests ⓒ Actions ⊞ Projects ⑦ Security	🗠 Insights 🕸 Settings		Q Type (7) to search
	MaterialShaderExample (Public)		☆ Pin ⓒ Watch C	0 ▼ 🔮 Fork 0 ▼ 🙀 Star 2 ▼
	🐉 master 👻 🐉 1 Branch 🚫 0 Tags	Q Go to file	t Add file + <> Code +	About 🕸
	Phyronnaz Update gitignore		88a103f · 2 days ago 🛛 16 Commits	Example of how to do a material shader in Unreal Engine
	Config	Progress	3 days ago	🛱 Readme
	Content		3 days ago	কাৃ্রু MIT license ∽∽ Activity
	Plugins/MaterialShaderPlugin	Fixes	3 days ago	습 2 stars
	Source	Progress	3 days ago	 ⊙ 0 watching ジ 0 forks
	🗋 .gitignore	Update gitignore	2 days ago	
		Create LICENSE	3 days ago	Releases
	MaterialShaderDemo.uproject	First commit	3 days ago	No releases published <u>Create a new release</u>
	C README.md	Create README.md	3 days ago	Packages
	따 README 책 MIT license			No packages published Publish your first package
Material Shader Example			Languages C++ 93.4% C# 6.6%	
	This project is a small example on how to do a It also shows how to hook into the Nanite visit	FMaterialShader in Unreal Engin ility buffer to replace shading bins	ne. with custom ones.	Suggested workflows Based on your tech stack

github.com/Phyronnaz/MaterialShaderExample


UNREAL FEST Bali 2025

Conclusion



Conclusion

There's many ways to customize the editor UI

You can access most private variables with a few tricks

Calling un-exported functions is often doable (just need to copy some code)

You can hook in most parts of the rendering pipeline

Hacks are usually fine – just wrap them in helper functions



Voxel Core

Open-source (MIT) part of Voxel Plugin

Highly optimized containers

Includes a bunch of editor helpers for toolkits, details etc

Performance

Мар

- TVoxelMap::FindChecked : 1.4x faster than TMap, 1.2x faster than TRobinHoodHashMap
- TVoxelMap::Remove : 1.4x faster than TMap, 2.4x faster than TRobinHoodHashMap
- TVoxelMap::FindOrAdd : 1.2x faster than TMap, 3.7x faster than TRobinHoodHashMap
- TVoxelMap::Reserve: 17.4x faster than TMap, 4.5x faster than TRobinHoodHashMap

TVoxelMap is also quite a bit smaller than TMap and TRobinHoodHashMap :

- uint16 -> uint16 with 1M elements: TVoxelMap 9.6MB, TMap 13.6MB, TRobinHoodHashMap 19.8MB
- uint32 -> uint32 with 1M elements: TVoxelMap 13.4MB, TMap 17.4MB, TRobinHoodHashMap 23.6MB

Bit array

- FVoxelBitArray::Add : 1.9x faster than TBitArray
- FVoxelBitArray::CountSetBits : 6.6x faster than TBitArray

Sparse array

- TVoxelSparseArray::Add : up to 1.7x faster than TSparseArray
- TVoxelSparseArray::RemoveAt : 1.5x faster than TSparseArray
- Iterating TVoxelSparseArray : up to 7.2x faster than TSparseArray
- TVoxelSparseArray::Reserve : 345x faster than TSparseArray

Misc

- TVoxelUniqueFunction : 1.8x faster than TUniqueFunction
- TVoxelArray::RemoveAtSwap: 2.9x faster than TArray (size here isn't compile-time)

Thank you!

Slides are on victorcareil.com

More about the plugin: voxelplugin.com

We have a Discord: discord.voxelplugin.com

Voxel Core: github.com/VoxelPlugin/VoxelCore

@phyronnaz on twitter/bsky/discord



Voxel plugin