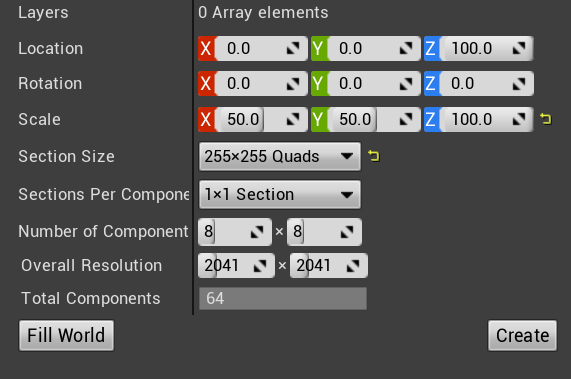
# 105 Landscape Setup Scaling



Scale 表示每个Quad的长度（单位厘米），这个值实际上设置的是地形的数据精细度，数值越小，精细度越高，但损失的是整体地形的大小。这里使用50表示每个Quad覆盖半米。

Section Size表示每个Section中有多少个Quad

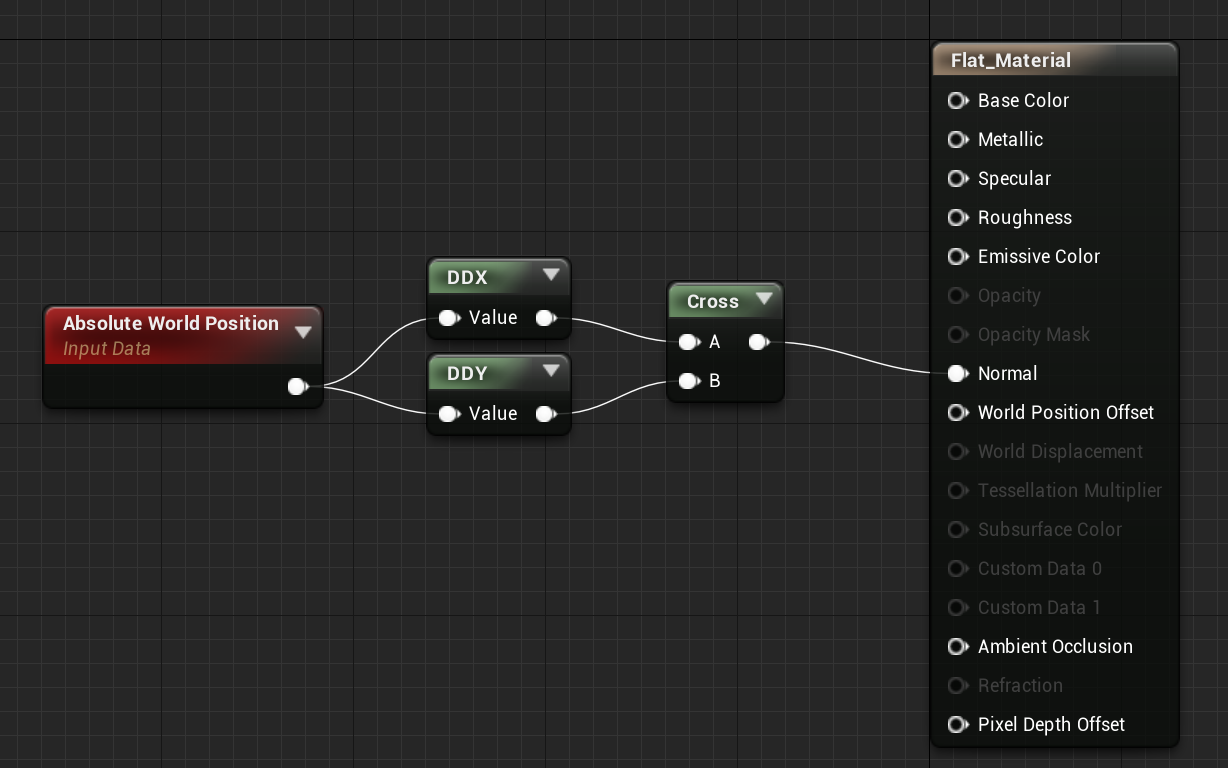
Section Per Component只有1x1和2x2两个选项，这个暂时先不用关心。

Number of Component表示总共有多少个Component(Component就是Section)

Overall Resolution是Section Size \* Number of Components + 1得来的。

上图中Overall Resolution 是2041 x2041，Scale是50，那么点击Create以后生成的是一块大约一平方公里的地形。2041\*0.5 约1020米，约一公里。

# 109 Flat Shading Low Poly Landscape



LowPoly材质

# 112 Actors from Multiple Meshes

可以给Static Mesh设置Socket，然后在蓝图中，子物体可以附加到父物体的Socket中。

在蓝图中创建好父子级关系，在子物体的Details面板，找到Sockets栏，点击放大镜设置好Parent Socket即可。

# 114 3rd Person Camera Control

在Graphs中可以创建新的Graph，但是只有EventGraph中可以存在BeginPlay

# 129 Finding Screen Pixel Coordinates

获取当前屏幕大小，在PlayerController中:

int32 ViewportSizeX, ViewportSizeY;

GetViewportSize(ViewportSizeX, ViewportSizeY);

# 130 Using DeprojectScreenToWorld

屏幕空间转换到世界空间:

使用APlayerController::DeprojectScreenPositionToWorld

bool DeprojectScreenPositionToWorld  
(  
    float ScreenX,  
    float ScreenY,  
    [FVector](http://api.unrealengine.com/INT/API/Runtime/Core/Math/FVector/index.html) & WorldLocation,  
    [FVector](http://api.unrealengine.com/INT/API/Runtime/Core/Math/FVector/index.html) & WorldDirection  
)

参数3 WorldLocation是相机的世界位置

参数4 WorldDirection是从相机指向屏幕点的方向

# 131 Using LineTraceSingleByChannel

使用UWorld::LineTraceSingleByChannel来进行射线追踪

bool LineTraceSingleByChannel  
(  
    struct [FHitResult](http://api.unrealengine.com/INT/API/Runtime/Engine/Engine/FHitResult/index.html) & OutHit,  
    const [FVector](http://api.unrealengine.com/INT/API/Runtime/Core/Math/FVector/index.html) & Start,  
    const [FVector](http://api.unrealengine.com/INT/API/Runtime/Core/Math/FVector/index.html) & End,  
    [ECollisionChannel](http://api.unrealengine.com/INT/API/Runtime/Engine/Engine/ECollisionChannel/index.html) TraceChannel,  
    const [FCollisionQueryParams](http://api.unrealengine.com/INT/API/Runtime/Engine/FCollisionQueryParams/index.html) & Params,  
    const [FCollisionResponseParams](http://api.unrealengine.com/INT/API/Runtime/Engine/FCollisionResponseParams/index.html) & ResponseParam  
)

其中参数TraceChannel是一个枚举值，包含在：

ECollisionChannel:: 中。

常用的是 ECollisionChannel::ECC\_Visibility

获取相机的世界位置可以使用:

PlayerCameraManager->GetCameraLocation();

绘制Debug Point:

首先需要include

#include "DrawDebugHelpers.h"

函数原型：

ENGINE\_API void DrawDebugPoint(

const UWorld\* InWorld,

FVector const& Position,

float Size,

FColor const& PointColor,

bool bPersistentLines = false,

float LifeTime=-1.f,

uint8 DepthPriority = 0);

使用范例：

DrawDebugPoint(

GetWorld(),

Location,

20, //size

FColor(255,0,255), //pink

false, //persistent (never goes away)

0.03 //point leaves a trail on moving object

);

# 133 Create Default Sub Objects in C++

C++中生成组件的方法：在构造函数中使用下面的函数，注意可以提供一个FName参数用于命名

TankAimingComponent = CreateDefaultSubobject<UTankAimingComponent>(FName("Aiming Component"));

# 139 Using Forward Declarations

IWYU 规则告诉我们，在cpp文件中都要引用匹配的头文件，而且该引用要放在最后一个。

在头文件中，如果需要引用某个类型，可以直接使用前向声明(Forward Declaration)，但是前向生命只能用于声明该类型的指针，无法访问其中的任何方法和属性。

所以如果在cpp文件中需要使用某类型的方法，使用前向声明就不够了，需要真正的include这个类型。

# 140 BlueprintSpawnableComponent

新创建的组建无法在蓝图的组件列表中新建，除非在类前加入：

*UCLASS*(ClassGroup = (*xxx*), *meta* = (*BlueprintSpawnableComponent*))

还可以隐藏一些选项，例如：

*UCLASS*(*meta* = (*BlueprintSpawnableComponent*),hidecategories=("Collision"))

这样，创建好的组件中就隐藏了Collision卷展栏