

```
1
2
3 #include "Grabber.h"
4 #include "Engine/World.h"
5 #include "DrawDebugHelpers.h"
6
7 UGrabber::UGrabber()
8 {
9     PrimaryComponentTick.bCanEverTick = true;
10 }
11
12
13 void UGrabber::BeginPlay()
14 {
15     Super::BeginPlay();
16 }
17 }
18
19
20 void UGrabber::TickComponent(float DeltaTime, ELevelTick TickType,
21                               FVectorComponentTickFunction* ThisTickFunction)
22 {
23     Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
24
25     UPhysicsHandleComponent *PhysicsHandle = GetPhysicsHandle();
26     if (PhysicsHandle == nullptr)
27     {
28         return;
29     }
30     if (PhysicsHandle->GetGrabbedComponent() != nullptr)
31     {
32         FVector TargetLocation = GetComponentLocation() + GetForwardVector
33             * HoldDistance;
34         PhysicsHandle->SetTargetLocationAndRotation(TargetLocation,
35             GetComponentRotation());
36     }
37 }
38
39 void UGrabber::Grab()
40 {
41     UPhysicsHandleComponent* PhysicsHandle = GetPhysicsHandle();
42     if (PhysicsHandle == nullptr)
43     {
44         return;
45     }
46     FHitResult HitResult;
47     bool HasHit = GetGrabbableInReach(HitResult);
```

```
47
48     if (HasHit)
49     {
50         UPrimitiveComponent* HitComponent = HitResult.GetComponent();
51         HitComponent->WakeAllRigidBodies();
52         PhysicsHandle->GrabComponentAtLocationWithRotation(
53             HitComponent,
54             NAME_None,
55             HitResult.ImpactPoint,
56             GetComponentRotation()
57         );
58     }
59 }
60
61 void UGrabber::Release()
62 {
63     UPhysicsHandleComponent * PhysicsHandle = GetPhysicsHandle();
64     if (PhysicsHandle == nullptr)
65     {
66         return;
67     }
68
69     if(PhysicsHandle->GetGrabbedComponent() != nullptr)
70     {
71         PhysicsHandle->GetGrabbedComponent()->WakeAllRigidBodies();
72         PhysicsHandle->ReleaseComponent();
73     }
74 }
75
76 UPhysicsHandleComponent* UGrabber::GetPhysicsHandle() const
77 {
78     UPhysicsHandleComponent* Result = GetOwner()-
79         >FindComponentByClass<UPhysicsHandleComponent>();
80     if(Result == nullptr)
81     {
82         UE_LOG(LogTemp, Display, TEXT("Grabber requires a
83             UPhysicsHandleComponeont"));
84     }
85     return Result;
86 }
87
88 bool UGrabber::GetGrabbableInReach(FHitResult& OutHitResult) const
89 {
90     FVector Start = GetComponentLocation();
91     FVector End = Start + GetForwardVector() * MaxGrabDistance;
92     DrawDebugLine(GetWorld(), Start, End, FColor::Red);
93     DrawDebugSphere(GetWorld(), End, 10, 19, FColor::Blue, false, 5);
94     FCollisionShape Sphere = FCollisionShape::MakeSphere(GrabRadius);
95     return GetWorld()->SweepSingleByChannel(
```

```
94     OutHitResult,  
95     Start, End,  
96     FQuat::Identity,  
97     ECC_GameTraceChannel2,  
98     Sphere);  
99 }  
100
```