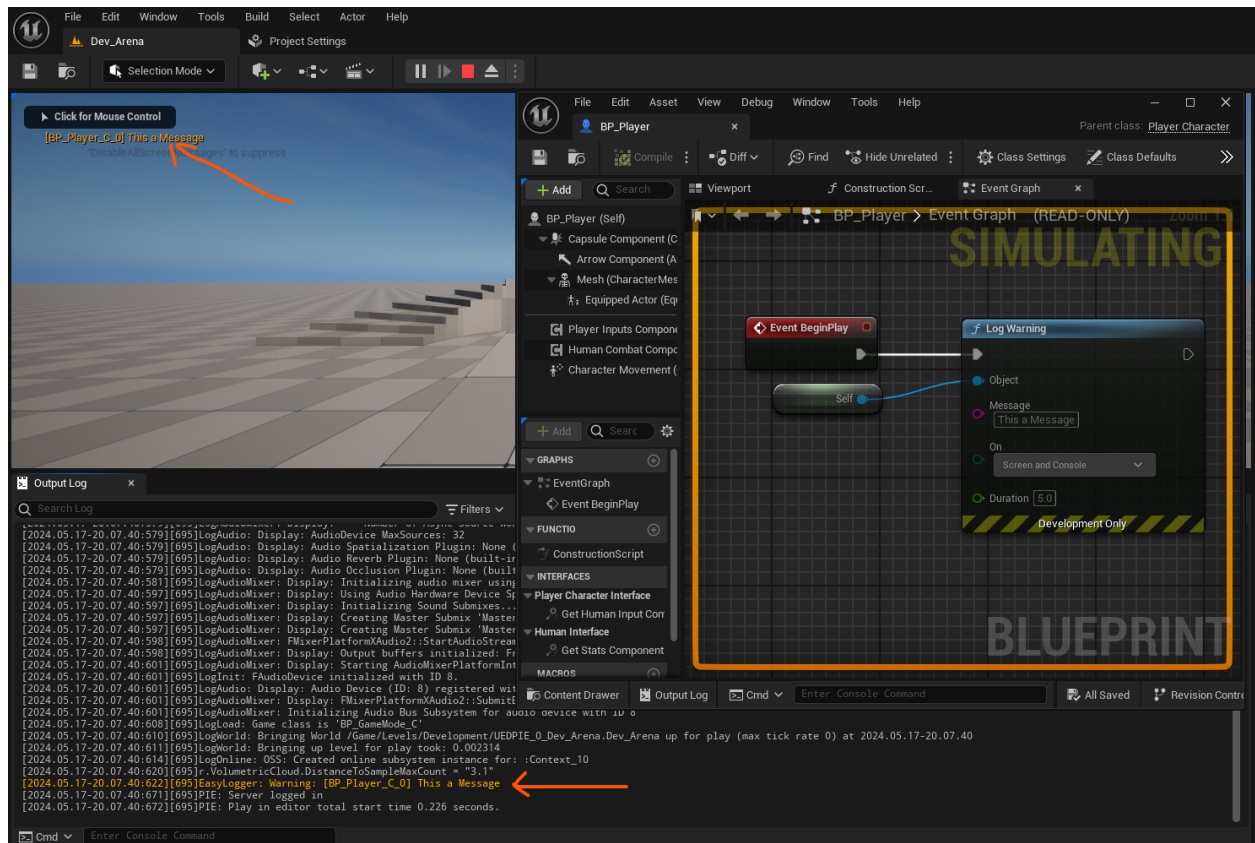


Hi! Here's how to use Easy Logger as a logging and debugging tool in your project. There are 2 use cases for that: in Blueprints or C++ Codes.

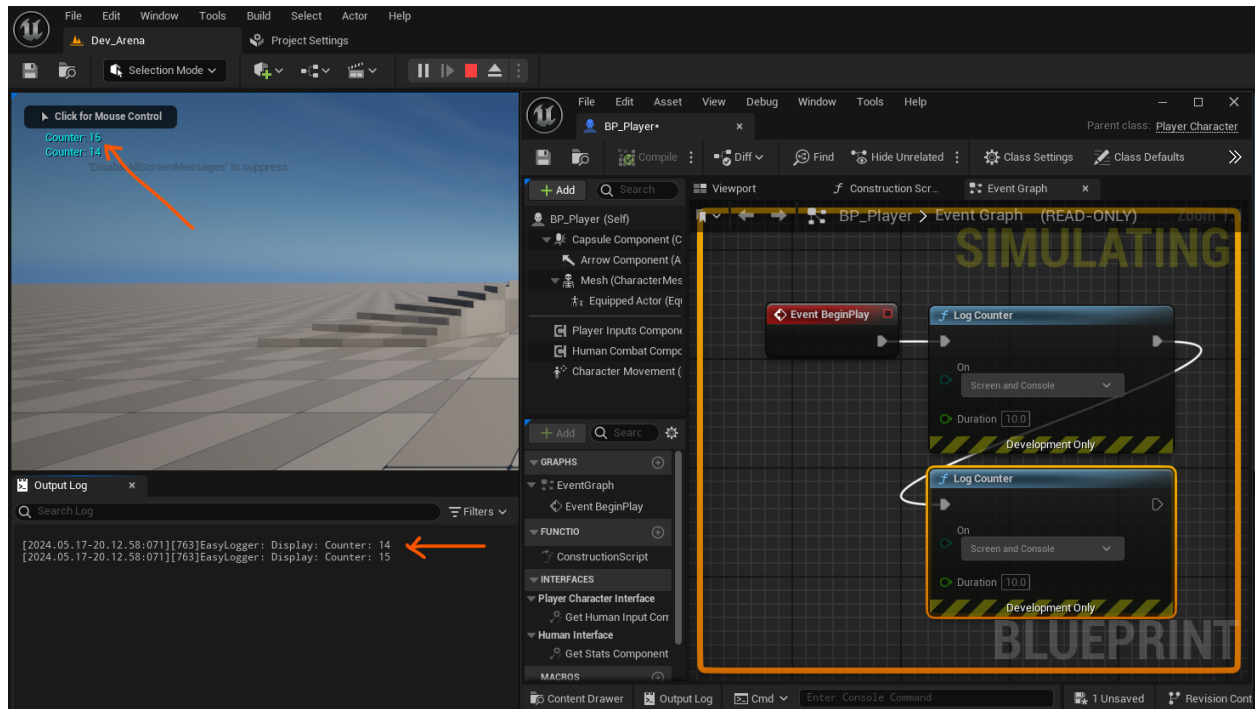
1. Blueprints:

Here's how to use a LogWarning on your Screen and Console. If you provide an object as a parameter, the LogInfo will log the calling object's name and append the message to it. This is helpful for easier debugging as it identifies who called this log. If no object is passed to it, it will print the raw message to the desired log interface. You can also specify the duration the message log would appear on your screen.

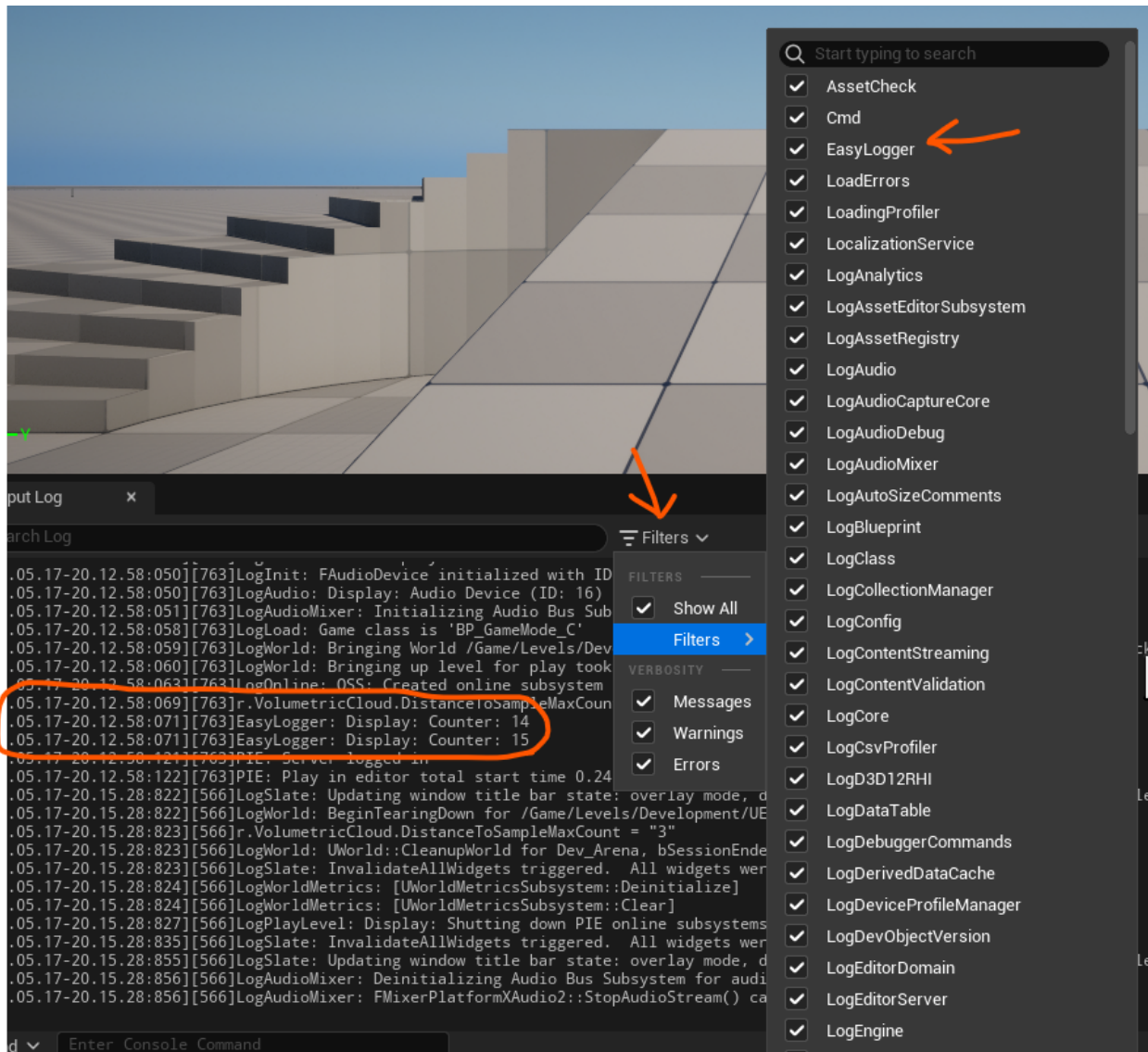


This node functions perfectly for both LogInfo and LogError.

Use the LogCounter() function to log the progress easily. It will internally keep count for convenience. Here is a demonstration:



You have the option to filter the EasyLogger logs by applying filters to focus on specific information.



In C++ it is even easier.

```
void UHumanCombatComponent::BeginPlay() {  
    Super::BeginPlay();  
  
    HumanRef = Cast<APlayerCharacter>( GetOwner() );  
    if (!HumanRef) {  
        ULogger::LogError( this, TEXT( "HumanRef is null." ) );  
    }else {  
        //..  
    }  
}
```

And that's it. If you have any questions or suggestions, please comment in the comment section of the EasyLogger product page on [Marketplace](#).

Have a nice rest of the day.