Table 1: `Outlets` for `UAffinitiesComponent`

ElIf	
▶ Before	111
▶ After	`const float OriginalYield`, `const float ReturnedYield`

• • •

Table 2: `Outlets` for `ULevelComponent`

▶ Before	`const float OriginalYield`, `float& ReturnedYield`
▶ After	`const float OriginalYield`, `const float ReturnedYield`
GetCXP	
▶ Before	`const uint32 OriginalCXP`, `int32& ReturnedCXP`
Note:	`ReturnedCXP` is `int32&` instead of `uint32&` for Blueprint compatability.
▶ After	`const uint32 OriginalCXP` `const int32 ReturnedCXP`
Note:	`ReturnedCXP` is `const int32` instead of `const uint32` for Blueprint compatability.
GetExpYield	
▶ Before	`const float OriginalYield`, `float& ReturnedYield`, `const uint16 DefeatedLevel`, `const uint16 VictoriousLevel`
Note:	"Defeated" and "Victorious" levels are provided for flexibility (e.g., in case you want to yield exp differently based on level difference, although technically you could always back-calculate the level difference based on the equation and `OriginalYield`).

Continued on next page

Table 2: `Outlets` for `ULevelComponent` (Continued)

▶ After	`const float OriginalYied`, `const float ReturnedYield`, `const uint16 DefeatedLevel`, `const uint16 VictoriousLevel`
Note:	"Defeated" and "Victorious" levels are provided for symmetry with respect to the `Before` delegate (since `ReturnedValue` is already calculated, I can't think of why you would need them, but you never know!).
GetMaxLevel	
▶ Before	`const uint16 DefaultMax`, `int32& AttemptedMax`
Note:	`DefaultMax` is defined in the code. It should normally be 100, but may change for certain subclasses (e.g., a `UBossLevelComponent` may have a max of 200 instead).
Note:	`AttemptedMax` is `int32&` instead of `uint16&` for Blueprint compatability.
▶ After	`const uint16 DefaultMax` `const int32 ReturnedMax`
GetMinLevel	
▶ Before	`const uint16 DefaultMin`, `int32& AttemptedMin`
Note:	`DefaultMin` is defined in the code. It should normally be 1, but may change for certain subclasses (e.g., a `UEggLevelComponent` may have a min of 0 instead for whatever reason). Also, `AttemptedMin` is `int32&` instead of `uint16&` for Blueprint compatability.
▶ After	`const uint16 DefaultMin` `const int32 ReturnedMin`
SetBaseExpYield	
▶ Before	`const float OldYield`, `float& AttemptedYield`

Continued on next page

Table 2: Outlets for ULevelComponent (Continued)

▶ After	const float OldYield const float NewYield
SetCXP	
▶ Before	const uint32 OldCXP, int32& AttemptedCXP
Note:	AttemptedCXP is int32& instead of uint32& for Blueprint compatability.
▶ After	const uint32 OldCXP const uint32 NewCXP
Note:	UStatsComponent subscribes to this in order to change stats on level change.

• • •

Table 3: Outlets for UStatsComponent

RandomizeStats	
▶ Before	const EStatEnum TargetStat, const FStatRandParams OriginalParams, FStatRandParams& ParamsToBeUsed
▶ After	const EStatEnum TargetStat, const FStatRandParams OriginalParams, const FStatRandParams UsedParams
Note:	The EStatEnum is not the acutal FStat. To get the FStat (such as FHealth), use UStatsComponent::GetStat(EStatEnum).
RecalculateStats	
▶ Before	const EStatEnum TargetStat, const bool bResetCurrent, const float OriginalCurrent, const float OriginalPermanent

Continued on next page

Table 3: `Outlets` for `UStatsComponent` (Continued)

▶ After	`const EStatEnum TargetStat,`
	`const bool bResetCurrent,`
	`const float OriginalCurrent,`
	`const float OriginalPermanent`