GD3P01 - ExpBoost Plugin

Class Definitions

Aiden Storey

Class Definition Key

Variables

- Public Variable
- Protected Variable
- UProperty
- Accessible via Blueprints

Functions:

- Public Variable
- Protected Variable
- UFunction
- Accessible via Blueprints

UEBComponent

Inheritance Hierarchy

UObjectBase

UObjectBaseUtility

UObject

UActorComponent

UEBComponent

Syntax

```
UCLASS( Blueprintable, HideCategories = ( Tags, ComponentReplication, Activation ) )
class UEBComponent : public UActorComponent
```

Remarks

UEBComponent is the main component in the *ExpBoost* plugin. It handles the delegation of experience input/output and attributes through its supporting components.

Delegates

```
DECLARE_DYNAMIC_MULTICAST_DELEGATE( FDefaultDelegate );
DECLARE_DYNAMIC_MULTICAST_DELEGATE_OneParam( FLevelUpDelegate, int32, Level );
DECLARE_DYNAMIC_MULTICAST_DELEGATE_OneParam( FGainExpDelegate, int32, ExpAmount );
```

Variables

		Name	Description
V Ⓤ ▤	FString	HealthComponentName	Name of the attached component for health
V Ⓤ ▤	bool	IsAlive	If the component is alive
V Ⓤ ▤	FDefaultDelegate	OnDeath	Delegate called when component has died
V Ⓤ ▤	int32	Level	The current level of the component
V Ⓤ ▤	int32	LevelMax	The max level that the component can reach
V Ⓤ ▤	bool	AllowMultiLevel	If the component can level multiple times from a single Exp amount
V Ⓤ ▤	int32	Exp	The current Exp for the component
V Ⓤ ▤	int32	ExpMax	The Exp required for the component to level up
V Ⓤ ▤	UCurveFloat*	ExpCurve	Curve that the required Exp is taken from
V Ⓤ ▤	FLevelUpDelegate	OnLevelUp	Delegate called when components level increases
V Ⓤ ▤	FGainExpDelegate	OnGainExp	Delegate called when Exp is gained
V Ⓤ ▤	AActor*	LastDamager	Last actor that caused damage to the component
V	TArray< class UEBAttribute * >	Attributes	List of attributes bound to components level up

Constructors

	Name	Description
	UEBComponent ( )	Default *UEBComponent* constructor

Functions

		Name	Description
	UEBAttribute*	GetAttribute (const FString& Name)	Returns attribute component on owner that matches passed name. Returns nullptr if not found.
	void	BindAttribute (UEBAttribute* _Attribute)	Adds attribute to list for updating on level up
	void	UnbindAttribute (UEBAttribute* _Attribute)	Removes attribute from list for updating on level up
	void	NotifyAttributes (int32 _Level)	Iterates over attached attributes and sets their level to be same as the components

ƒ Ⓤ ▤	void	ReceiveExp (const int32& ExpAmount)	Updates the exp and level of the component based on the amount passed and current setup. Can broadcast *OnLevelUp* and *OnGainExp* delegates.
ƒ Ⓤ ▤	int32	CalculateExpRequiredAtLevel (const int32& Level)	Gets the amount of exp required at level from attached curve
ƒ Ⓤ ▤	int32	CalculateModifiedExp (const int32& ExpAmount)	Empty function that can be overloaded by the user to add modifiers for exp addition
ƒ Ⓤ ▤	void	TakeDamage (float DamageAmount, AActor* DamageCauser)	Reduces the health attribute if present by amount passed and handles death when met. Can broadcast *OnDeath* delegate.

Overridden from UActorComponent

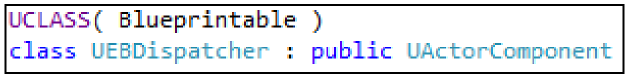
		Name	Description
	void	InitializeComponent ( )	Sets the required Exp amount for given level
	void	DestroyComponent (bool bPromoteChildren)	Unbinds all of the attached attribute components

UEBDispatcher

Inheritance Hierarchy

- UObjectBase
  - UObjectBaseUtility
    - UObject
      - UActorComponent
        - UEBDispatcher

Syntax

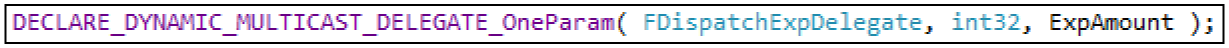


```
UCLASS( Blueprintable )
class UEBDispatcher : public UActorComponent
```

Remarks

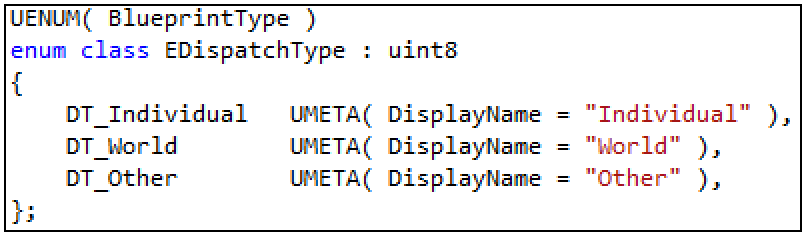
UEBDispatcher is a supporting component of the *ExpBoost* plugin. It enables an *AActor* to dispatch experience as notified by the available *UEBComponent*.

Delegates



```
DECLARE_DYNAMIC_MULTICAST_DELEGATE_OneParam( FDispatchExpDelegate, int32, ExpAmount );
```

Types



```
UENUM( BlueprintType )
enum class EDispatchType : uint8
{
    DT_Individual   UMETA( DisplayName = "Individual" ),
    DT_World        UMETA( DisplayName = "World" ),
    DT_Other        UMETA( DisplayName = "Other" ),
};
```

Variables

		Name	Description
	EDispatcherType	DispatchType	The way in which the dispatcher should give out experience
	Int32	DispatchAmount	The amount of experience the dispatcher should give out
	FDispatchExpDelegate	OnDispatchExp	Delegate called when experience is dispatched
	FDispatchExpDelegate	OnDispatchIndividualExp	Delegate called when experience is dispatched to an individual
	FDispatchExpDelegate	OnDispatchWorldExp	Delegate called when experience is dispatched to the world
	FDispatchExpDelegate	OnDispatchOtherExp	Delegate called when experience is dispatched as user implemented other

Constructors

	Name	Description
ƒ🟩	UEBDispatcher ( )	Default *UEBDispatcher* constructor

Functions

		Name	Description
ƒ🟩 ⓤ ▤	void	DispatchExp (UEBComponent* Component)	Dispatches experience dbased on dispatcher type. Broadcasts *OnDispatchExp* delegate.
ƒ🟩 ⓤ ▤	void	DispatchExpOther (UEBAttribute* _Attribute)	Empty function to be overridden by user to extend the functionality of the dispatcher. Broadcasts *OnDispatchOtherExp*
ƒ🟨	void	DispatchExpIndividual (UEBComponent* Component)	Dispatches experience to the last hit *AActor* for the *UEBComponent* passed in. Broadcasts *OnDispatchExpIndividual*
ƒ🟨 ⓤ ▤	void	DispatchExpWorld ( )	Server function to dispatch experience to all connected players in the level. Broadcasts *OnDispatchExpWorld*.

UEBReceiver

Inheritance Hierarchy

UObjectBase

UObjectBaseUtility

UObject

UActorComponent

UEBReceiver

Syntax

```
UCLASS( Blueprintable, HideCategories = ( Tags, ComponentReplication, Activation ) )
class UEBReceiver : public UActorComponent
```

Remarks

UEBReceiver is a supporting component in the *ExpBoost* plugin. It enables an *AActor* to be able to receive experience, passing the experience to an available *UEBComponent*.

Delegates

```
DECLARE_DYNAMIC_MULTICAST_DELEGATE_OneParam( FReceiveExpDelegate, int32, ExpAmount );
```

Variables

		Name	Description
	FReceiveExpDelegate	OnReceiveExp	Delegate called when experience is received

Functions

		Name	Description
	void	ReceiveExp (const int32& ExpAmount)	An overridable function that handles receiving experience and moving it to the available *UEBComponent*. Broadcasts *OnReceiveExp* delegate.

UEBAttribute

Inheritance Hierarchy

UObjectBase

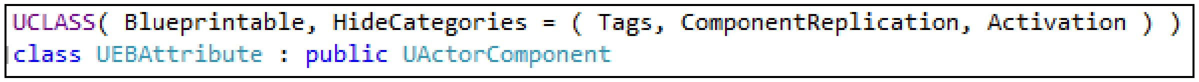
UObjectBaseUtility

UObject

UActorComponent

UEBAttribute

Syntax



```
UCLASS( Blueprintable, HideCategories = ( Tags, ComponentReplication, Activation ) )
class UEBAttribute : public UActorComponent
```

Remarks

UEBAttribute is a supporting component in the *ExpBoost* plugin. It stores information about a user defined attribute to be used in a game. Class is friend of *UEBComponent*.

Variables

		Name	Description
	int32	Level	The current level of the attribute
	int32	LevelMax	The max level of the attribute
	UCurveFloat	AttributeCurve	The attribute curve to derive the value given the level
	float	Value	The current value of the attribute
	float	ValueAtLeve	The value of the attribute at the given level from the attribute curve
	float	ValueMax	The max value for the attribute
	bool	ValueIsMax	If the value is always going to be at its max
	bool	ResetValueOnStartUp	If the value should be set to max on start up
	bool	ResetValueOnLevelUp	If the value should be set to max on level up
	TArray< FEBAttributeModifier>	Modifiers	List of modifiers that are altering the max value
	bool	BindToComponent	If the attribute should take level from the component
	UEBComponent*	BindComponent	Pointer to the component attribute is bound to

Constructors

	Name	Description
ƒ	UEBAttribute (const FObjectInitializer& ObjectInitializer)	Default *UObject* constructor

Functions

		Name	Description
	void	SetLevel (int32 _Level)	Sets the current level of attribute (if not bound to component) ensuring that max level isn't exceeded. Recalculates max value.
	void	SetLevelMax (int32 _Level)	Sets the max level of attribute (if not bound to component) ensuring that current doesn't exceed it. Recalculates max value.
	void	SetAttributeCurve (UCurveFloat* _AttributeCurve)	Sets the attribute curve of attribute. Recalculates max value.
	void	SetValue (const float& _Value)	Sets the current value of attribute ensuring it doesn't exceed max value.
	void	SetValueIsMax (const bool& _ValueIsMax)	Sets if the value should be equal to its max constantly. Sets it if true.
	void	SetBindToComponent (const bool& _Bind)	Sets if the attribute should bind to *UEBComponent* to receive its levels. Binds/Unbinds if required.

ƒ	void	BindEBComponent ( )	Locates *UEBComponent* on owner and attaches if possible.
ƒ	void	UnbindEBComponent ( )	Unbinds from *UEBComponent* if pointer isn't null.
ƒ	float	GetValueAtLevel (int32 _Level)	Gets the base value for attribute at passed level from the attached attribute curve
ƒ	void	AttachModifier (const FEBAttributeModifier& _Modifier)	Adds modifier to modifiers list and recalculates current max for attribute.
ƒ	void	DetachModifier (const FEBAttributeModifier& _Modifier)	Removes modifier from modifiers list and recalculates current max for attribute.
ƒ	void	RecalculateTotal ( )	Iterates over bound modifiers and calculates the max value for the attribute. (Base + Static) * (1 + Modifier + (Percent / 100))
ƒ	void	OnBoundLevelUp (int32 _Level)	Set level function for bound component to update levels.

Overridden from UActorComponent

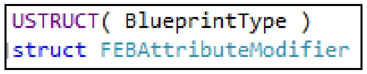
		Name	Description
ƒ Ⓥ	void	InitializeComponent ( )	Binds component if required. Sets the value to max if ResetValueOnStartUp set to true.
ƒ Ⓥ	void	DestroyComponent (bool bPromoteChildren)	Unbinds component if required.

FEBAttribtueModifier

Inheritance Hierarchy

FEBAttributeModifier

Syntax

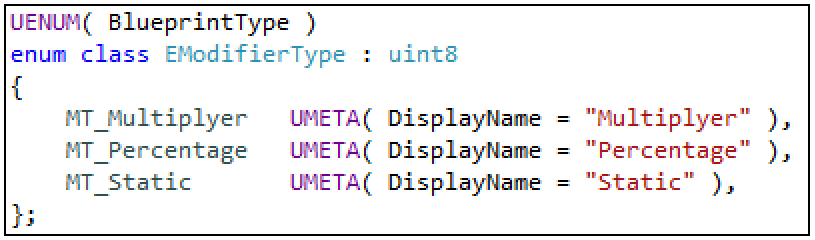


```
USTRUCT( BlueprintType )
struct FEBAttributeModifier
```

Remarks

FEBAttributeModifier is a supporting struct in the *ExpBoost* plugin. It is used to define a way to modify the max value of a named attribute.

Types



```
UENUM( BlueprintType )
enum class EModifierType : uint8
{
    MT_Multiplyer   UMETA( DisplayName = "Multiplyer" ),
    MT_Percentage   UMETA( DisplayName = "Percentage" ),
    MT_Static       UMETA( DisplayName = "Static" ),
};
```

Variables

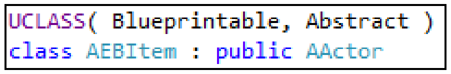
		Name	Description
	FString	AttributeName	Name of attribute to modify
	EModifierType	Type	The way the modifier applies to the attribute
	float	Amount	Amount the attribute should change

AEBItem

Inheritance Hierarchy

- UObjectBase
  - UObjectBaseUtility
    - UObject
      - AActor
        - AEBItem

Syntax

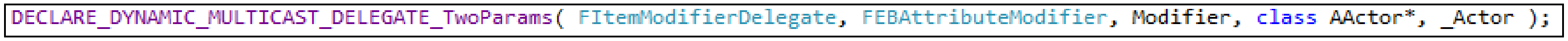


```
UCLASS( Blueprintable, Abstract )
class AEBItem : public AActor
```

Remarks

AEBItem is a supporting class of the ExpBoost plugin. It wraps modifiers up to easily override and treat as an attribute modifying item that can be attached to an actor.

Delegates



```
DECLARE_DYNAMIC_MULTICAST_DELEGATE_TwoParams( FItemModifierDelegate, FEBAttributeModifier, Modifier, class AActor*, _Actor );
```

Variables

		Name	Description
	TArray< FEBAttributeModifier >	Modifiers	Attribute modifiers used when attached
	FItemModifierDelegate	OnAttachModifier	Delegate called when Exp is gained
	FItemModifierDelegate	OnDetachModifier	Delegate called when Exp is gained

Constructors

	Name	Description
	AEBItem (const FObjectInitializer& ObjectInitializer)	Default *UObject* constructor

Functions

		Name	Description
	bool	AttachItem (AActor* _Actor)	Attaches to the passed actor. Searches for all available attributes that match with the modifiers, binding them when found.
	bool	DetachItem ( )	Detaches from the set actor. Unbinds all bound modifiers from attributes.