BLUEPRINTS TO C++

UNREAL ENGINE 4 - C++ PROGRAMMING GUIDE

EPISODE 8

STRUCT BASICS

OUTLINE

1. About Structs
2. Creating a Struct
3. Constructors
4. Operators
5. Equal Functions
6. Tip of the Day – Struct as Key for a TMap

ABOUT

(From the Epic online Unreal Engine 4 Documentation)

Structs are data structures that help you organize and manipulate related properties. In the Unreal Engine, they're recognized by the Engine's reflection system, but are not part of the UObject ecosystem. As a result, they are faster to create than a UObject with the same data layout, and support UProperties, but will not be managed by the Garbage Collection system and cannot provide UFunctions. By using structs, you can create custom variable types to help organize your project. Structs are different from UObjects, and because of this Structs are best used for simple data types. For more complicated interactions within your project, you may want to make a UObject or AActor subclass instead.

https://docs.unrealengine.com/en-US/Programming/UnrealArchitecture/Reference/Structs/index.html

Note: A struct in general is similar to a class, in fact they are the same internally. But in a struct all members are public by default.

CREATING A STRUCT

```
USTRUCT(BlueprintType)
struct FGridIndex
{
	GENERATED_BODY()

	UPROPERTY(EditAnywhere, BlueprintReadWrite)
	int32 X;

	UPROPERTY(EditAnywhere, BlueprintReadWrite)
	int32 Y;
};
```

STRUCT CONSTRUCTORS

```
USTRUCT(BlueprintType)
struct FGridIndex
{
        GENERATED_BODY()

        FORCEINLINE FGridIndex();
        explicit FORCEINLINE FGridIndex(int32 InValue);
        explicit FORCEINLINE FGridIndex(int32 InX, int32 InY);
        …
};
FORCEINLINE FGridIndex::FGridIndex() {  }
FORCEINLINE FGridIndex::FGridIndex(const int32 InValue) : X(InValue), Y(InValue){  }
FORCEINLINE FGridIndex::FGridIndex(const int32 InX, const int32 InY) : X(InX), Y(InY) {  }
```

STRUCT IMPLICIT/EXPLICIT CONSTRUCTORS

Using the explicit keyword prevents unwanted implicit type conversations and is used in declarations of constructors. Except for the default constructor, all other constructors are used for implicit type conversions.

Let‘s define a constructor implicitely:

```
FORCEINLINE FGridIndex(int32 InValue);
```

Then the following declaration would be perfectly legal:

```
FGridIndex Index = 17;
```

Which is equivalent to

```
 FGridIndex Index =  FGridIndex(17);
```

If we would define our constructor explicitely:

```
explicit FORCEINLINE FGridIndex(int32 InValue);
```

Then only statements like the following ones would be legal (and *FGridIndex Index = 17; is illegal*)

```
FGridIndex Index1;

FGridIndex Index2 = FGridIndex(17);

FGridIndex Index3(17);
```

STRUCT OPERATORS

```
USTRUCT(BlueprintType)
struct FGridIndex
{
	…
	bool operator==(const FGridIndex& V) const;
	bool operator!=(const FGridIndex& V) const;
};
FORCEINLINE bool FGridIndex::operator==(const FGridIndex& V) const
{
	return X==V.X && Y==V.Y;
}
FORCEINLINE bool FGridIndex::operator!=(const FGridIndex& V) const
{
	return X!=V.X || Y!=V.Y;
}
```

STRUCT EQUALS FUNCTIONS

```
USTRUCT(BlueprintType)
struct FGridIndex
{
	…
	bool Equals(const FGridIndex& V, float Tolerance=KINDA_SMALL_NUMBER) const;
	bool AllComponentsEqual(float Tolerance=KINDA_SMALL_NUMBER) const;
};
FORCEINLINE bool FGridIndex::Equals(const FGridIndex& V, float Tolerance) const
{
	return FMath::Abs(X-V.X) <= Tolerance && FMath::Abs(Y-V.Y) <= Tolerance;
}
FORCEINLINE bool FGridIndex::AllComponentsEqual(float Tolerance) const
{
	return FMath::Abs(X - Y) <= Tolerance;
}
```

TIP OF THE DAY – STRUCT AS KEY FOR A TMAP

```
USTRUCT(BlueprintType)
struct FGridIndex
{ … };
```

To be able to use structs in a TMap, you need to define the GetTypeHash global function, and place it underneath the struct declaration in the header file

```
FORCEINLINE uint32 GetTypeHash(const FGridIndex& b)
{
    return FCrc::MemCrc_DEPRECATED(&b, sizeof(FGridIndex));
}
```

Example Usage:

```
TMap<FGrindIndex, AGridTile*> GridTiles;
```

THANK YOU FOR WATCHING

IF YOU WANT TO GET NOTIFIED WHEN NEW VIDEOS ARE COMING OUT

THEN PLEASE SUBSCRIBE TO THE CHANNEL