C++ IN UNREAL

**BASE**

Nel file .cpp bisogna sempre includere la sua controparte .h

Ex: nel file **BullCowCartdrige.cpp** bisogna mettere #include "BullCowCartridge.h"

Per inizializzare le funzioni, bisogna prima inizializzarle nel .h

Ex: nel file **BullCowCartdrige..h** bisogna mettere void InitGame();

E poi è possibile usarla nel .cpp dove lì poi possiamo scrivere la sua esecuzione:

Ex: nel file **BullCowCartdrige..h** inizializzato void InitGame();, possiamo sia chaiamrla, tramite InitGame();, sia scriverne il conte4nuto, tramite void UBullCowCartridge::InitGame(){}.

Se nel .h inizializziamo le funzioni, esse possono essere chiamate, ed implementate, in qualunque parte del codice, e non devono essere dichiarate prima di essere utilizzate

In Unreal, è buona norma mettere qualunque stringa nella funzione TEXT(). Questo perché una stringa normale, usa simboli ASCII che non tutte le piattaforme leggono, mentre inserendola in TEXT(), sarà unreal stesso, a convertirle nel formato più adatto e leggibile dalla piattaforma.

Ex: FString HiddenWord = TEXT("cake");

PrintLine(TEXT("Press Enter to continue..."));

const FString& Input; vuol dire che viene inizializzata un puntatore a quella variabile. Quindi ogni modifica verrà fatta alla variabile input. Scrivendo const FString Input, vuol dire che verrà fatta una copia di quella variabile, e verrà usata quella. Const vuol dire che per tutta quella funzione, quella variabile non può essere modificata.

Di seguito un esempio per stampare una stringa di tipo FString, e quindi proprietaria di Unreal.

Ex: PrintLine(FString::Printf(TEXT("The hidden word is: %s.\n Is: %i characters long."),\*HiddenWord, HiddenWord.Len()));

**COMANDi**

* PrintLine("Testo\n”); stampa la linea di testo.
* ClearScreen();pulisce lo schermo dalle righe di testo.
* FString HiddenWord = "cake"; crea una variabile di tipo stringa (Unreal).

**Booleans (17:49)**