Unreal Engine Simple Automation Test

Creating a Simple Automation Test Class

1. Open Unreal Engine.
2. Select "C++ Classes" folder in the Content Browser.
3. Right Click and add a new C++ class.
4. Select None for the parent class.
5. Select "Private".
6. Input the class name with the following format "*FeatureBeingTested*Test.cpp".

Eg: AmmoComponentTest.cpp

1. Add "Tests/" to the end of the file path.
2. Once compiling has finished close Unreal Engine.
3. Open the projects source folder and navigate to the new class.
4. Delete the header file and rename the cpp file to Return to the project folder and delete the following folders/file:
   1. .vs
   2. Binaries
   3. Intermediate
   4. Saved
   5. "*ProjectName*.sln"
5. Right click the "*ProjectName*.uproject" file and select "Generate Visual Studio Project Files".
6. Once completed open the "*ProjectName*.sln" file.
7. Build the vs project by pressing F5.
8. Open the "*FeatureBeingTested*Test.cpp" file in visual studio.

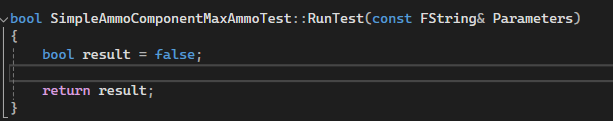
Writing a Simple Automation Test

1. Delete everything that in the "*Featurebeingtested*Test.cpp" file.
2. At the top of the file you will need to add any includes that are required for the test to run.
   1. The Class you are testing, eg: #include “AmmoComponent.h”
   2. If a character is required for testing, in this case for the fire function.
3. Start with the IMPLEMENT\_SIMPLE\_AUTOMATION\_TEST Macro.
   1. The first section of the Macro is the test class name.
   2. The second section is where the test will be located in the session front end.
   3. The third section is for the [EAutomationTestFlags](https://docs.unrealengine.com/4.26/en-US/API/Runtime/Core/Misc/EAutomationTestFlags__Type/).







1. Write a "RunTest" function that returns a bool for the test class name that was assigned in the Macro with a parameter: "const FString& Parameters". The returned bool is the result of the test. 
2. Inside the "RunTest" function can go all the code to perform the required test and return then return the result. You can also add errors using the "AddError" function that takes a FString parameter for the error message.

