Proposals for improvements

Generalize FiniteStateMachine so that it’s easier to create a child class of. Additionally, if you want to attempt making changes in blueprints before or instead of coding it first change the functions to be like StateObject’s functions which are Blueprintable and C++virtual && \_implementation.

Generalize StateObject && its children’s functions. At the moment they require a UFiniteStateMachine pointer, however if someone were to create and utilize a child of UFiniteStateMachine for another purpose, they would have to write their own StateObject, or rewrite the current StateObject, then compile and fail because they forgot to change StateObject’s children as well.

A detailed step-by-step guide to recreate the very basics of the FiniteStateMachine and StateObject, so that others can find out, and you can look up to remember, how to make templates work in both cpp && blueprints. With details and reasons for each line broken down. It may be tedious in reading and writing, but I believe that it would be useful enough to overcome the tedium. Or maybe not specifically FiniteStateMachine && StateObject but another class that is also a template utilized in Blueprints && C++.

Implement a way into the FiniteStateMachine to return the current state’s class. At the moment it just returns the pointer to the state variable which is of type StateObject. This would need to be done in a way befitting to the State Machine pattern, it would also need to be Blueprints && C++ available. I have my doubts that this can be done, or easily done, so I would say it’s a low priority as there are ways of getting around this. Such as comparing a cast of the pointer to another pointer that is of the type you are looking for (which has been done in blueprints).