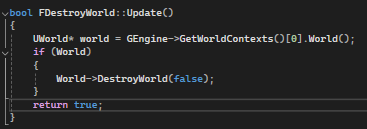
Unreal Engine Simple/Complex Latent Commands Automation Test

Creating a Latent Command

1. Open a test .cpp file which has a simple or complex test written in it.
2. To declare the latent command we use the macro "DEFINE\_LATENT\_AUTOMATION\_COMMAND" followed by parentheses containing the name of the latent command. E.G. DEFINE\_LATENT\_AUTOMATION\_COMMAND(FDestroyWorld);
3. Next to define the command we create an update function for the FDestroyWorld class that returns a bool. E.G. bool FDestroyWorld::Update().
4. Inside this update function we write out the command which will be run each frame.
5. Finally the command needs to return true when it is completed and return false if it needs to be run again on the next frame.





Implementing a Latent Command

1. Open a test .cpp file which has a simple or complex test written in it and where a Latent Command has been declared and defined.
2. Navigate to where you would like the latent command to be implemented.
3. To call the command we use the macro "ADD\_LATENT\_AUTOMATION\_COMAND" followed by the command name inside of parentheses E.G. ADD\_LATENT\_AUTOMATION\_COMMAND(FDestroyWorld);

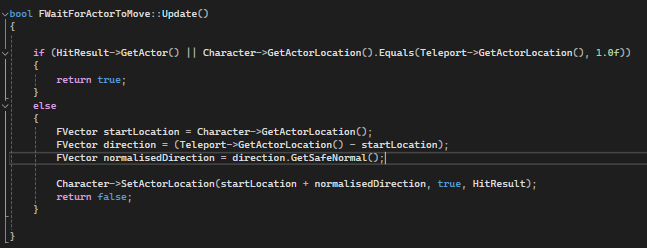


Creating a Latent Command with Parameters

1. Open a test .cpp file which has a simple or complex test written in it.
2. To declare the latent command we use the macro "DEFINE\_LATENT\_AUTOMATION\_COMMAND" followed by parentheses containing the name of the latent command. E.G. DEFINE\_LATENT\_AUTOMATION\_COMMAND(FWaitForActorToMove);
3. When adding parameters for the command we include these in the parentheses however we have to add the parameter type as a separate parameter E.G. DEFINE\_LATENT\_AUTOMATION\_COMMAND(FWaitForActorToMove, AActor\*, Player);

This can be done with up to 5 parameters.

1. Next to define the command we create an update function for the FWaitForActorToMove class that returns a bool. E.G. bool FWaitForActorToMove::Update().
2. Inside this update function we write out the command which will be run each frame. We are able to use the parameters in this update function by using the reference we past into the parameters E.G. Player->GetActorLocation();
3. Finally the command needs to return true when it is completed and return false if it needs to be run again on the next frame.



Implementing a Latent Command with Parameters

1. Open a test .cpp file which has a simple or complex test written in it and where a Latent Command has been declared and defined.
2. Navigate to where you would like the latent command to be implemented.
3. To call the command we use the macro "ADD\_LATENT\_AUTOMATION\_COMAND" followed by the command name inside of parentheses E.G. ADD\_LATENT\_AUTOMATION\_COMMAND(FWaitForActorToMove);
4. For the parameter values we make another parentheses inside of the first lot after the command name where we can add the parameters we want. E.G.  
    ADD\_LATENT\_AUTOMATION\_COMMAND(FWaitForActorToMove(Player,Teleport,HitResult));