**UCommanderComponent**: Class in charge of issuing commands to any unit. Any relevant action a unit can take must go through the commander component. The commander component can both be the local player or any other actor that has control over units.

* **ExecuteAttackCommand:** Creates a new **UAttackCommand** with the unit reference and the attack target as arguments. Also, callbacks if the attack fails or succeeds. The success callback is to attack the same target again, and the failure is to execute a stop command.
* **ExecuteStopCommand:** Creates a new **UStopCommand** with the unit reference as an argument is created.
* **ExecuteGatherCommand:** Creates a new gather command with the unit reference and target resource as arguments. Also, callbacks if the gather fails or succeeds. The success callback stores the resource and the failure one executes a stop command.
* **ExecuteMovementCommand:** Creates a new move command with the unit reference and target position as arguments. Also, callbacks if the movement fails or succeeds. Both callbacks execute a stop command.
* **ExecuteStoreCommand:** Creates a new store command with the unit reference and target storage as arguments. Also, callbacks if the storing fails or succeeds. The success callback is to gather again the same resource and the failure one is to execute a stop command.

**UAttackCommand:** Class in charge of making the unit attack the targeted actor. This command is a two-step command, first, we create and execute a **UMovementCommand** towards the attack target until the unit is in range of attack. When the movement command succeeds, execution returns to the attack command and executes the attack using the **UUnitAttackComponent.**

**UStopCommand:** Class in charge of making the unit stop all actions.

**UGatherCommand:** Class in charge of making the unit gather the targeted resource. Is a two-step command. Is a two-step command, first, we create and execute a **UMovementCommand** towards the target resource until the unit is in range. When the movement command succeeds, execution return to the gather command and extracts the resource through the **UUnitGathererComponent**.

**UMovementCommand**: Class in charge of making the unit move towards the target position. Once the target is within acceptance range the command finishes execution.

**UStoreCommand**: Class in charge of sending any resource the unit has in its possession into the storage. Is a two-step command. Is a two-step command, first, we create and execute a **UMovementCommand** towards the storage until the unit is in range. Then execution comes back to the store command and stores the resource through the **UUnitGathererComponent.**