**ABordo Metahuman**

**Process to create Metahuman character and set animations (animations created with Live Link Face and Rokoko)**

1. Activate Metahuman Plugin(download+activate)
2. Create Metahuman with MHC
3. Import MH + activate required plugins
4. Configure MH Face hair:
   * GRoom: Use Card ->True
5. Import body Animations
6. Retarget Animations
7. Create body Animation Blueprint
8. Import face animations
9. Retarget Animations
10. Modify face Animation Blueprint
    * Create new state machine with imported animations
    * Blend them with the body
11. Configure Look at of Body(Head) and Face(eyes)
    * Body(head): Use Look at node between State Machine and Output Pose (Head bone)
    * Face(eyes): Use Look at node before Output Pose, in Face\_PostProcess\_AnimationBlueprint (FACIAL\_L\_Eye + FACIAL\_R\_Eye)
      + \*Important change Look at and Look up axis
12. Now control Metahuman by Blueprints programming, for the moment with Level Blueprint(we will use BehaviourTree)

***Body Animation Blueprint***

1. Create new Animation Blueprint with *Metahuman Base Skel* Skeletal Mesh
2. Create New State Machine in the Animation Graph
3. State Machine:
   1. Create States (Idle, InitialMessage, Alarm Message, Walking….) with the animations
   2. Map: Entry -> Idle -> Conduit =>Other States (Initial InitialMessage, Alarm Message, Walking….)
   3. Create Transition Rules (with Enumeration)
      1. Create an Enumerator defining different States
      2. Create a variable in AnimBP using this type
      3. For Walking State transition use Ground Speed(like Third Person Character)

***Face Animation Blueprint***

1. Open Metahuman Face\_AnimBP
2. In the AnimGraph delete all nodes except ***Cache Pose*** and ***Output Pose***
   1. Keep last ***Layered blend per bone*** (this will blend our animations with the body)
   2. We will have our custom animations so the rest we don’t need
3. Create new State Machine
   1. Create States (Idle, InitialMessage, Alarm Message ….) with the animations
   2. Map: Entry -> Idle -> Conduit =>Other States (Initial InitialMessage, Alarm Message, Walking….)
   3. Create Transition Rules (with integer variable)
      1. Associate a value to states(Idle =0, InitialMessage=1….)