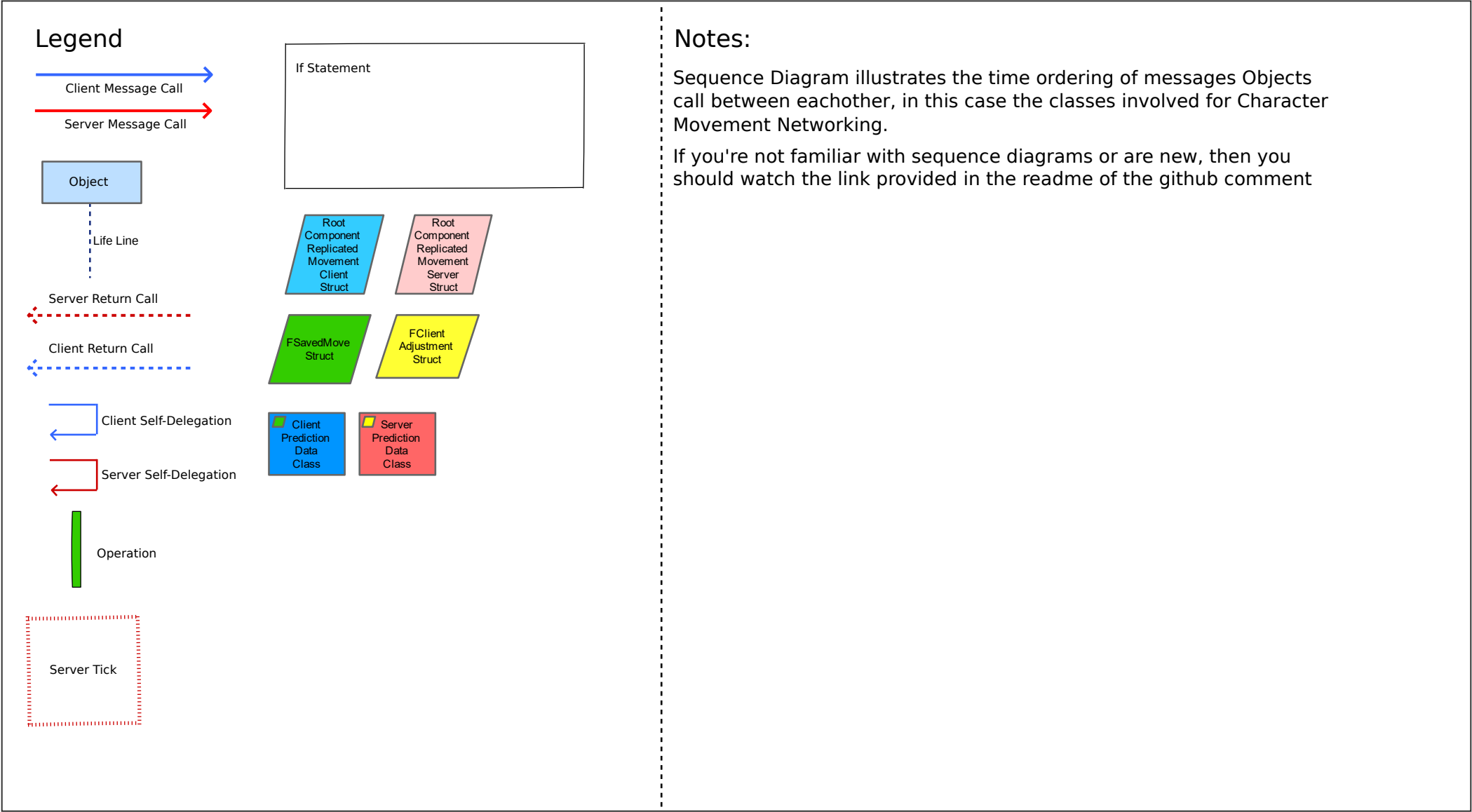
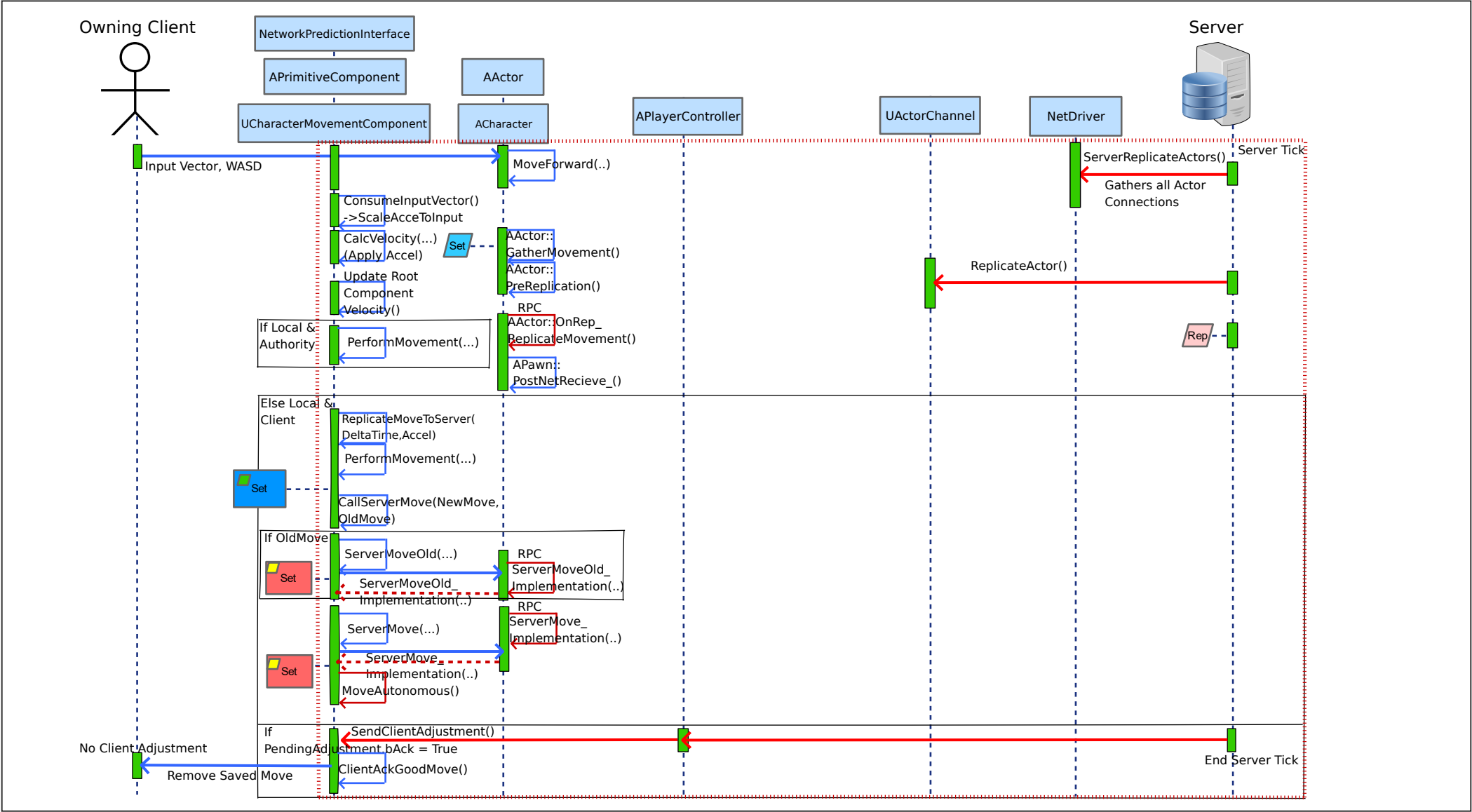


Legend



ClientServerMain



```

sequenceDiagram
    participant Client as Owning Client
    participant UChar as UCharacterMovementComponent
    participant AChar as ACharacter
    participant APlayer as APlayerController
    participant UActor as UActorChannel
    participant NetDriver as NetDriver
    participant Server as Server

    Note over Client, UChar: Else Local & Client
    UChar->>Server: ServerMoveHandle  
ClientError()
    Note over UChar: ServerCheckClientError()
    Note over UChar: If Client Error
    Note over UChar: bool
    Note over UChar: PendingAdjustment
    Note over UChar: .bAckGoodMove = False
    Note over UChar: Set Pending Moves
    UChar->>UActor: ReplicateActor()
    Note over UChar: Else
    Note over UChar: If Client MovementModes are Null
    UChar->>Server: ApplyNetworkMovementMode()
    UChar->>Server: SetBase()
    UChar->>Server: UpdateFloorAdjustment()
    UChar->>Server: SaveBaseLocation()

    Note over Server: ServerReplicateActors()
    Note over Server: Gathers all Actor Connections
    Note over Server: Server Tick
    Note over Server: Rep
    Note over Server: End Server Tick

    Note over Server: Connection()
    Server->>APlayer: Previous SendClientAdjustment()
    Note over APlayer: Server Tick
    APlayer->>UChar: ClientAdjustPosition()
    Note over AChar: CharacterOwner
    AChar->>UChar: ClientAdjustPosition()
    Note over AChar: ClientAdjustPositionImplementation()
    UChar->>Client: ApplyNetworkMovementMode()
    UChar->>Client: SetBase()
    UChar->>Client: UpdateFloorAdjustment()
    UChar->>Client: SaveBaseLocation()
    Note over Client: Client Adjusted
    Note over Client: Save Adjusted Move
  
```

The diagram illustrates the networked movement logic. On the client side, the `UCharacterMovementComponent` handles local movement and sends `ServerMoveHandleClientError()` to the server. It also manages a `PendingAdjustment` and a `bAckGoodMove` flag. When a client error occurs, it triggers `ReplicateActor()` via the `UActorChannel`. The server's `ServerReplicateActors()` method gathers all actor connections during a `Server Tick`. The server then sends a `Connection()` message back to the client's `APlayerController`, which forwards a `ClientAdjustPosition()` message to the `ACharacter`. The `ACharacter` then calls `ClientAdjustPositionImplementation()` on the `UCharacterMovementComponent`. Finally, the `UCharacterMovementComponent` applies network movement modes and saves the adjusted move back to the client, marking it as `Client Adjusted`.

[illegible]

