# ILogger

Interface, som skal stå for logning af separation events. Implementeres I applikationen som FileLogger, og I testscenariet som FakeLogger.

# IRenderer

Står for at rendere Seperation Events og Tracks. Implementeres I applikationen som ConsoleRenderer, og I testscenariet som FakeRenderer.

# TrackData

Modelklasse, som indeholder alt data for et “track”. Indeholder getters, setters og en overloaded constructor.

# SeperationEvent

Modelklasse, som indeholder data for en Seperation Event. Indeholder “Time of occurrence”(lige nu I dateTime-format, men andet duer selvfølgelig også), en liste over tags for de involverede tracks, samt en bool, som indikerer om den er raised eller ej (muligvis unødvendig, skal den bare fjernes fra listen af SeperationEvents istedet?)

# ATM

“Hovedklassen”.

Har interfaces I form af ILogger, IRenderer og ITransponderReceiver(det er den, der er blevet udleveret, og jeg er ikke helt 100 på hvordan den fungerer),

Derudover har den en liste, som holder styr på de nuværende seperationEvents, samt en liste som holder styr på de nuværende tracks.

Desuden har den minimum- og maximum-værdier for henholdsvis X og Y.

Der er setters til interfaces, så de kan ændres I run time.

Der er desuden getters og setters på min- og max-værdierne.

## ATM::CheckIfInMonitoredArea(x,y):bool

Checker om de given koordinater ligger inde for systemets interesse-område.

## ATM::CheckForSeperationEvent(TrackData, TrackData)

Checker om der er conditions for Seperation Event imellem de to given TrackData’s.

## ATM::GetSeperationEventInvolvedIn(TrackData):SeperationEvent

Checker den nuværende liste af Separation Events, og checker om nogle af dem har den given TrackData’s tag som en af dens “involved track tags”. I så fald returneres dette Seperation Event. Hvis ingen match returneres null.

## ATM::RenderSeperationEvents(List<SeperationEvent>):void

Renderer listen af Seperation Events via IRenderer’en.

## ATM::RenderTracks(List<TrackData>)

Renderer listen af tracks via IRender’en

## ATM::LogSeperationEvent(SeperationEvent):void

Logger SeperationEventet ved hjælp af ILogger’en. Skal kun kaldes første gang Eventet “raises”, og altså ikke kontinuert, hvis eventet bliver ved med at være “raised” når der kommer ny data.

## ATM::SubscribeToTransponderDataReady():void

Subscribe til eventet TransponderDataReady fra ITransponderReceiver’en (Jeg er ikke helt 100 på hvordan det her fungerer)

## ATM::OnTransponderDataReady():void

Callback-funktion: Jeg forestiller mig at man kan få en slags callback fra eventet når der er ny data klar. Denne callback-funktion vil så tage den modtagne data (vist nok I et format, hvor det er en liste af strings), og lave et nyt TrackData-objekt ud fra dette, hvorefter HandleNewTrackData kaldes med dette.

## ATM:: HandleNewTrackData

Dette er “hoved-funktionen”, som sørger for at håndtere den modtagne data. Proceduren er som fremgår:

