

SWT Hand In 3 Gruppe 20

Aarhus University School of Engineering

Air Traffic Monitoring

25-04-2018

Studienummer	Navn	Email-adresse
201607589	Jakob Bonde Nielsen	au567214@post.au.dk
201607110	Kasper Juul Hermansen	au557919@post.au.dk
201605114	Stefanie Ruaya Nielson	au554093@post.au.dk

Link til Git: <https://github.com/TeamTyve/ATM>

Link til Jenkins - UnitTest: <http://ci3.ase.au.dk:8080/job/TeamTyveATMUnitTest/>

Link til Jenkins - integrationstests:

Link til Jenkins - Coverage: <http://ci3.ase.au.dk:8080/job/TeamTyveATMCoverage/>



Indholdsfortegnelse

Indholdsfortegnelse	2
1 Design	3
1.1 Formål	3
1.2 Design	3
1.3 Resultater	3
2 Test	4
2.1 Unit tests	4
2.2 Integrationstests	4
2.3 Konklusion	4

Design 1

1.1 Formål

Formålet med denne opgave er at lave et system der kan monitorere og præsentere flytrafik over et givent område. Der vil også blive kigget på om flyene i området flyver for tæt. Informationer om flight-tracks vil blive indhentet fra en TransponderReceiver, givet i strings. Dernæst objektifiseres disse strings, og objekterne præsenteres i en konsol. Hele systemet vil blive testet ved hjælp af både unit tests samt integrationstests.

1.2 Design

Link til Github: <https://github.com/TeamTyve/ATM>

Der er i opgaven givet en .dll fil der benyttes til at indhente flight-tracks og præsentere disse i strings. Der implementeres derfor en Track Objectification Service, som benyttes til at objektifisere de indhentede strings. Når strings'ne er blevet objektifiseret vil systemet herefter blive præsenteret i en konsol.

Formålet med at få objektifiseret disse strings, er at man får gjort de indhentede strings meget mere overskuelige når de omdannes til objekter.

Objekterne er fly, som består af henholdsvis et tag (navnet på flyet), x- og y-koordinater, højde i luften, hastighed, retning i grader samt et timestamp.

Programmet vil blive versionsstyret ved hjælp af GitHub, og Jenkins vil blive benyttet til Continuous Integration, samt Code Coverage af henholdsvis unit tests samt integrationstests.

Klassediagram

Nedenstående ses klassediagrammet for systemet

Sekvensdiagram

1.3 Resultater

Link til Jenkins - Coverage: <http://ci3.ase.au.dk:8080/job/TeamTyveATMCoverage/>

Som det ses på nedenstående billeder er opgaven blevet løst som forventet. Flyene i det ønskede airspace bliver udskrevet, og der kommer en warning når flyene flyver for tæt på hinanden.

Test 2

2.1 Unit tests

Link til Jenkins - Unit tests: <http://ci3.ase.au.dk:8080/job/TeamTyveATMUnitTest/>

Obs! Vi har valgt ikke at teste `Output.cs`, da denne skriver direkte til konsollen, og det er ikke muligt at erstatte statiske metoder. Det vil derfor ikke være muligt at teste om `Console.WriteLine()` udskriver det korrekte.

2.2 Integrationstests

Link til Jenkins - Integrationstests:

2.3 Konklusion

Alt i alt er opgaven blevet løst som forventet. Der er, på trods af, at visse ting ikke er blevet testet, stadig en god code coverage.