

Code Assessment MTO

Je bent door het ministerie van Defensie gevraagd om een simulatie te ontwikkelen. De opdracht is om een Patriot luchtverdedigingssysteem te simuleren, welke bestaat uit drie basiscomponenten:

- Radar
- IFF
- Missile launcher

Afhankelijk van de inzet van defensie kan een eenheid uit meerdere van deze componenten bestaan.

De simulatie gaat 20 seconden duren. Elke tijdstap (1s) scant de radar module naar binnenkomende dreigingen, en geeft als output een regel uit het csv-bestand `radar_data.csv`.

De IFF (Identification Friend or Foe) module checkt vervolgens of er een vijandelijk object is gedetecteerd. Dit is het geval als er meer oneven dan even entries voorkomen in de decimale representatie van de output van de radar module.

Als er een vijandelijk object is gedetecteerd zal een missile worden afgeschoten door de launcher module. Een missile heeft een PK (Probability of kill)-ratio van 0.8. Het effect van de missile wordt gesimuleerd door een random-number generator te bevragen en het getal te vergelijken met de PK-ratio. Indien de waarde die je krijgt groter is dan dit ratio dan is er geen hit, en vice-versa voor getallen lager dan de PK-ratio.

Opdracht

Maak een applicatie die deze simulatie draait op basis van de meegestuurde `radar_data.csv`. Elke tijdstap moet uitgeprint worden wat de er die tijdstap is gebeurd in de simulatie. Is er een vijand gedetecteerd, missile afgevuurd, en is de dreiging geraakt? Gebruik hiervoor Java. Houd rekening met een nette opzet van de code zodat deze duidelijk is voor anderen. Jouw code zal later uitgebreid of vervangen worden, houd hier ook rekening mee. Stuur je code op samen met de instructies om de simulatie te kunnen draaien.

Aanleverinstructies

Zet je code in een publiek toegankelijke GIT repository (bijv. GitHub of GitLab) en stuur ons de link. Stuur geen code als attachment in een email.

Succes

Input file

radar_data.csv:

```
0001010;0110011;0100110;0010000;0011100;0101101;1111010;1011101;0110100;11
0111000;1101001;0011101;1000010;1010000;0011010;0110000;0010111;0101001;10
1101001;1010110;0101100;0110011;0011000;1111101;0001100;0111110;0011010;01
0111011;0100001;1001100;0010110;0010011;0100010;0110111;0101010;0101110;00
0110100;1011111;0101101;1010111;1101011;0010001;1011111;1111010;0100100;10
0100100;1111111;1100010;0111001;0111000;1010011;1100001;0010111;0111001;10
1110111;0101010;1010110;0010100;1111100;1001000;0000101;1111010;0001001;10
0100101;0101011;0010001;1101000;1010010;0101000;0010010;1001010;1000011;00
0010110;1010111;1011000;0011101;0100110;1110001;1001111;1111111;0101110;00
0001010;0001110;1111100;1001010;1000110;1110111;1010110;0101110;1000100;10
1101111;1000000;0101011;1100011;1001000;0011010;0101100;1000111;0100111;01
1100011;1011111;1110100;0101101;0010011;1110010;0011110;1000101;0110101;10
1110000;0110011;0011110;1000000;1001001;0101010;1101101;0001111;1010001;10
0101010;1000101;0110100;1000000;0001101;0001110;1011101;1001001;0001010;01
0011101;1111011;0000011;0011101;1011000;1011110;1111111;1110010;1001010;01
0011100;1011111;0101110;0011010;0111001;1100100;0001111;0000100;0010111;01
0110111;1111110;0011000;1101011;0001101;1101000;0011001;1100111;0101011;01
0000010;0110100;1001100;1111101;1110010;1110001;1101001;1100110;1011110;00
0010001;0100100;0100110;0100001;0001110;1001100;1001110;0001111;1100101;01
0110101;0100001;0100100;0111101;1010111;0111101;0000111;0001100;1000101;01
```