

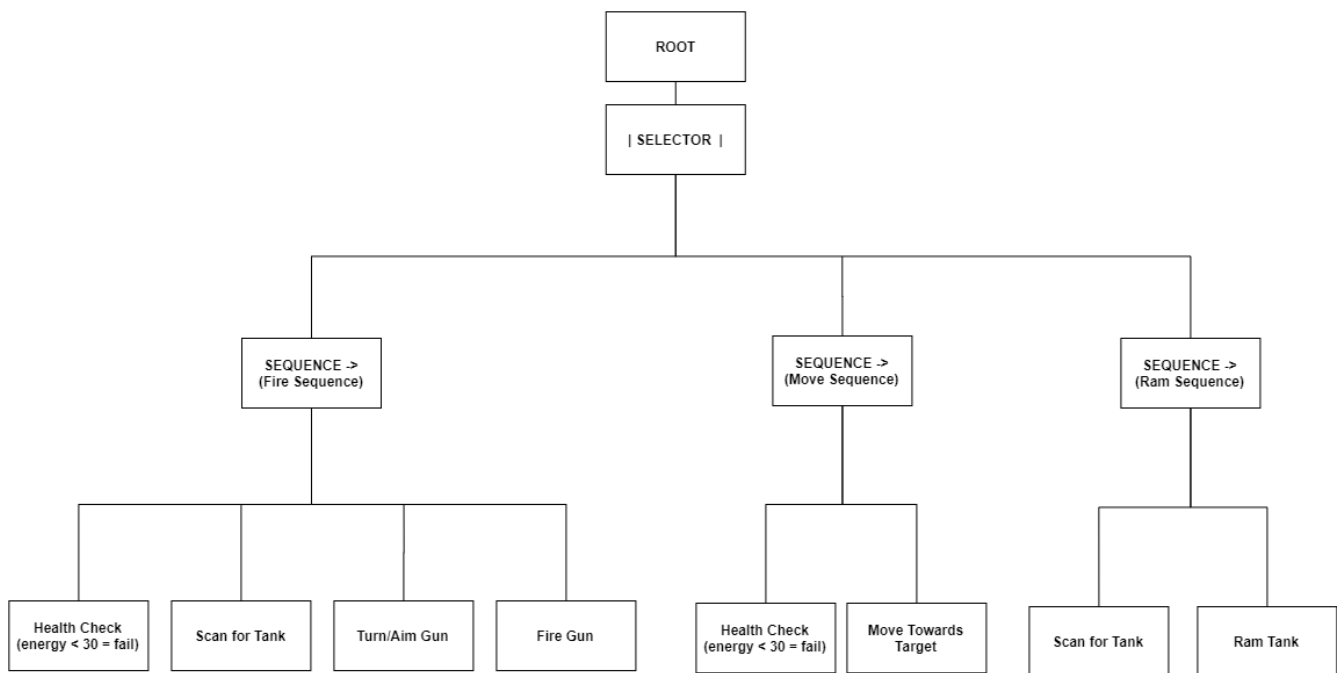
# Robocode Behaviour Tree

Tim Peeters 3029407

In dit document ga ik mijn robot 'Robro bespreken die mee heeft gedaan aan het robocode tournament (en hoe hij volledig verloren heeft).

## Behaviour Tree

Behaviour Tree 'RoBro'



Robro kan zich in principe in 3 hoofd states bevinden. De 'Fire Sequence', de 'Move Sequence' en de 'Ram Sequence'. Deze states zijn allemaal sequences van stappen(nodes) die de bot doorgaat om tot het doel te komen van de sequence.

### Fire Sequence

In deze sequence zoekt de tank wanneer zijn health (energy) boven 30 is naar een doel door middel van scannen, hierna draait hij de gun naar het verkregen doel en wanneer de hoek van de gun en het doel klein genoeg is vuurt de tank.

Wat bleek uit de gevechten is dat deze sequence heel slecht werkte doordat deze te lang was. Het duurde namelijk te lang voordat de tank vuurde na het scannen. Doordat ik de

sequence zo in elkaar heb gezet dat elke taak een aparte node heeft was het target na het scannen een groot deel van de tijd alweer van plek veranderd voordat de tank kon vuren.

### Move Sequence

Deze sequence wordt aangeroepen wanneer de fire sequence faalt. In deze sequence probeert de tank naar het eerder verkregen doel (van de fire sequence) te bewegen met een lichte offset om inkomende kogels te ontwijken. Deze sequence suceed eigenlijk altijd omdat hij enkel naar voren beweegt. Deze sequence faalt alleen als de health(energy) zich onder de 30 bevindt.

### Ram Sequence

De robot bevindt zich enkel en alleen in deze sequence wanneer de health(energy) zich onder de 30 bevindt. Als dit zo is gaat de tank over naar een ram sequence waar hij telkens naar de nieuwe positie van het target scant, waarna hij deze probeert te rammen door er direct naar toe te rijden in een zo korte mogelijke tijd.

Deze sequence bleek uiteindelijk fataal te zijn in veel gevechten doordat hij direct in de vuurlijn van andere tanks reed.

## Implementatie Behaviour Tree

Mijn implementatie van de behaviour tree is eigenlijk heel simpel. Een 'hoofd' selector en meerdere sequences die zich daaronder bevinden.

Ik heb gekozen voor deze aanpak omdat ik wilde werken met 'tactieken' waarin de tank zich kon bevinden. Zie elke sequence als een tactiek. Het vuren en richten, het naar een positie bewegen en het rammen zijn in principe 3 tactieken die de bot kan aannemen. Dit heb ik gerealiseerd door met een selector node te werken.

Deze selector is eigenlijk de node die de verschillende tactieken selecteerd aan de hand van omgevings variabelen. Ik wilde niet dat elke sequence telkens geloopt werd, maar dat de states/tactieken van de bot alleen geswitcht worden aan de hand van het falen van een vorige staat.

Verder heb ik voor sequences onder de 'hoofd' selector gekozen omdat deze tactieken/states in principe stappen zijn die allemaal na elkaar opgevolgd moeten worden om tot een succes te kunnen komen en de tactiek goed uit te kunnen voeren.

```
//Behaviour Tree
Node fireSequence = new SequenceNode(board, new HealthNode(board, 30, true), new ScanNode(board), new AimNode(board), new FireNode(board));

Node moveSequence = new SequenceNode(board, new HealthNode(board, 30, true), new MoveNode(board));

Node ramSequence = new SequenceNode(board, new HealthNode(board, 30, false), new ScanNode(board), new RamNode(board));

Root = new SelectorNode(board, fireSequence, moveSequence, ramSequence);
```