

Documentatie Kernmodule 3 AI (Robocode)

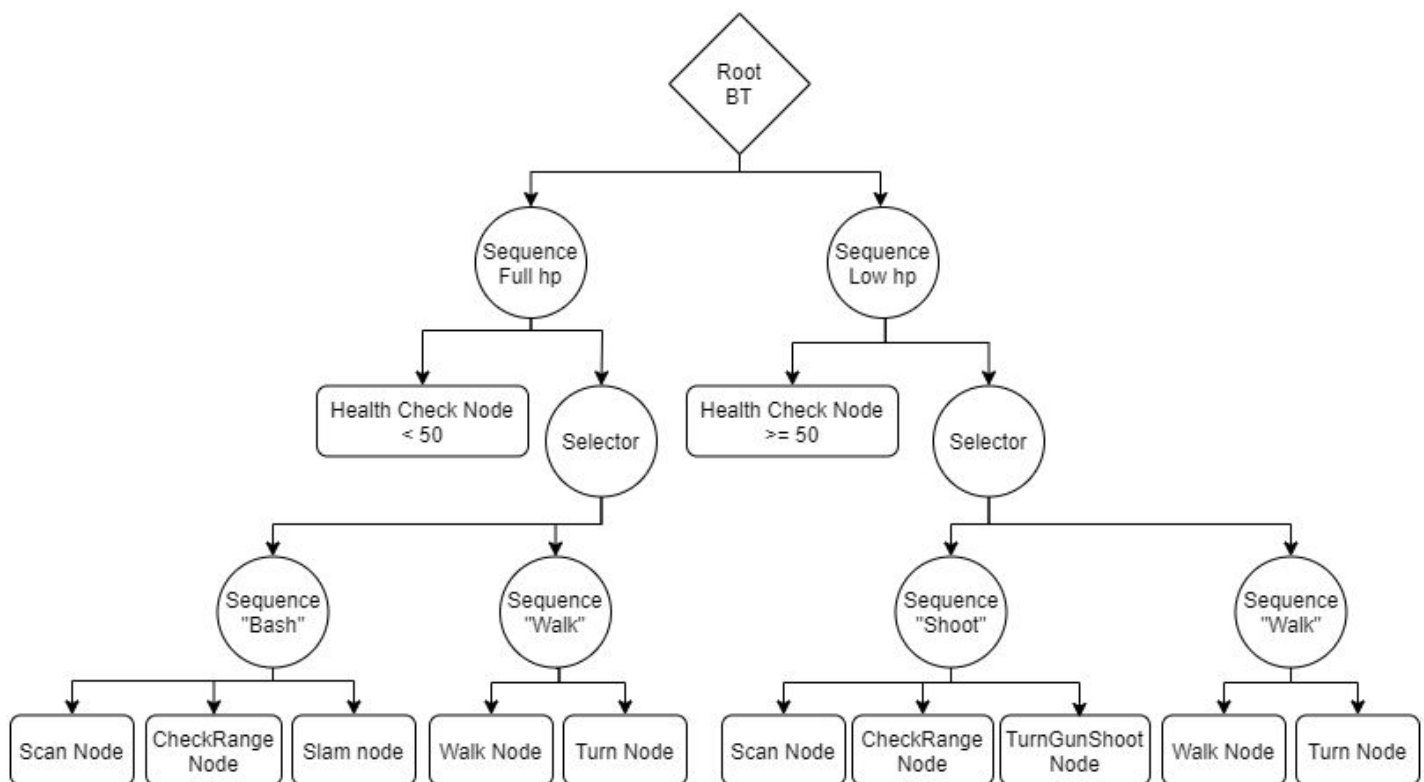
Naam: Dennis Borst
Studentnummer: 3030815

Intro:

In dit document ga ik meer vertellen over de robot die ik heb gemaakt in robocode. Daarnaast ga ik vertellen over wat ik allemaal geleerd heb op het gebied van een behaviour tree.

Behaviour Tree:

Voordat ik verder ga met het uitleggen, heb ik hier eerst een visualisatie van de behaviour tree die ik gemaakt heb.



Zoals hierboven te zien is, zie je dat er maar drie echte states zijn waar de robot zich in kan bevinden. Om bij deze drie states te komen moet de robot eerst nog wat checks doen, aan de hand daarvan wordt bepaald wat de robot gaat uitvoeren.

Eerst zal ik uit gaan leggen welke verschillende composite nodes ik hiervoor heb gebruikt. Dit zijn de selector en de sequence. De sequence komt de robot als eerst tegen in de tree, wat hier gebeurd is dat de sequence node, gaat kijken of de eerste node onder zich succes returned, zoja? Dan gaat hij verder naar de volgende node in de tree. Als deze eerste node failed zou returnen dan gaat hij verder kijken naar de volgende sequence in de tree.

Vervolgens als één van de twee sequences voltooid wordt dan zal hij vervolgen naar de selector node. Deze node kijkt ook net zoals een sequence naar de nodes onder zich. Als de node hieronder succes zou returnen dan betekent dat hij klaar is met de selector, zodra er een failed gereturned wordt dan zal de tree zich vervolgen.

Onder de composite nodes hangen de behaviour states die de robot kan uitvoeren. Zoals je kan zien heb ik voor deze robot drie verschillende states gemaakt. De robot kan rammen (Bash sequence), schieten (Shoot sequence) en lopen (Walk sequence).

De Bash state scant eerst om zo te zien of er robots in de buurt zijn, zodra deze succes returned gaat hij naar de "CheckRange Node" hier wordt in gecheckt of de robot die is gescand ook dichtbij genoeg is. Als die ook succes returned dan gaat de robot rechtstreeks op de andere robot af. In het spel is dit te zien aan de robot die zwart kleurt.

De Shoot state doet ook eerst een scan om te zien of er een robot in de buurt is, en vervolgens checkt ie of de robot in de aangegeven range is. Als deze succes returned dan gaat de robot in de "TurnGunShoot Node". Hier zal de robot zijn turret draaien richting de andere tank die hij gescand heeft. Daarna zal hij schieten die kant op. In het spel is dit te zien aan de robot die rood kleurt.

Als laatst hebben we de Walk state, hier wordt gezorgd dat de robot naar voren loopt en vervolgens dat de robot gaat draaien. Deze state wordt altijd uitgevoerd als de Shoot- of Bash state niet gedaan kunnen worden. In het spel is dit te zien aan de robot die blauw kleurt.

Dan tot slot had ik nog niks uitgelegd over de "Health Check Node". Deze node is er om te kijken in welke branche van de tree de robot zich moet bevinden. De naam zegt het al, hij checkt hoeveel levens de robot heeft en als hij meer dan een bepaald niveau heeft (hier 50) dan zal hij de Bash-Walk sequence ingaan, anders gaat de robot door in de Shoot-Walk sequence.

Reflexie:

Deze kernmodule heeft mij onwijs veel geleerd over hoe een behaviour tree in elkaar zit. Zo heb ik geleerd hoe de composite nodes bepalen hoe de behaviour nodes aan elkaar vast zitten in de tree. Met deze kennis kan ik een betere ai maken. Hiervoor gebruikte ik alleen maar een finite state machine dus dit is een toevoeging op die kennis.

De robot die ik gemaakt heb is niet super slim, maar is uiteindelijk toch effectiever dan gedacht. Hij heeft één ronde gewonnen en nog vrij goed gestreden tegen de andere robots.