

Robocode Documentatie

Ruben Hooijer | 3029382

Broncode: <https://github.com/loljoch/AI-Opdrachten/tree/master/RobocodeRobot>

Composite nodes

Ik heb een paar composite nodes die ik zelf had bedacht gemaakt. Deze werden vooral gebruikt, omdat ik maar 1 behaviour tree heb gebruikt.

Alle composite nodes vragen het blackboard mee.

Korte uitleg waar elke composite node voor gebruikt kan worden.

Sequencer(params BTNode[])

Deze composite voert alle tasks op volgorde uit.

IfSequencer(System.Func<bool>, params BTNode[])

Deze composite werkt hetzelfde als de sequencer, maar de gehele sequence wordt alleen uitgevoerd als de functie true teruggeeft.

(Dit had eigenlijk een decorator voor de sequencer moeten zijn)

Interrupter(System.Func<bool>, BTNode, params BTNode[])

Deze composite werkt als een sequencer, maar schakelt over naar een ander behaviour wanneer de functie true teruggeeft. Dit is handig voor als je de enemy al een tijdje niet hebt gescand en je hebt maar 1 tree die runt.

PongActions(params BTNode[])

Deze composite node doet elke tick elke Tick() maar 1 BTNode, maar elke Tick() gaat de index omhoog totdat hij de hoogste index bereikt en dan pong'd hij de index weer naar beneden

Selector(params BTNode[])

Deze composite voert alle tasks op volgorde uit, maar als een task succesvol uitgevoerd wordt dan stopt hij zijn sequence gelijk.

Task nodes

Alle tasks die vragen het blackboard mee.

Korte uitleg wat elke task is

Dance()

De robot begint rond te draaien en alle kleuren van de regenboog te worden.

MoveAhead(int)

De robot gaat (int) naar voren.

MoveToDistance(float)

De robot gaat op basis van de afstand van de laatst geziene robot en laatst kogel waar hij door geraakt is naar de robot toe of van de robot af zodat hij ongeveer op de (float) afstand van de andere robot zit.

ScanRobot(double)

De robot scanned om zich heen in een angle van (double).

Shoot(float)

De robot schiet met (float) power op basis mapsize en enemy afstand.

TurnBody(float)

Draait de robot zijn lichaam met (float).

TurnGunToEnemy()

Draait de robot zijn gun naar waar de enemy gaat zijn.

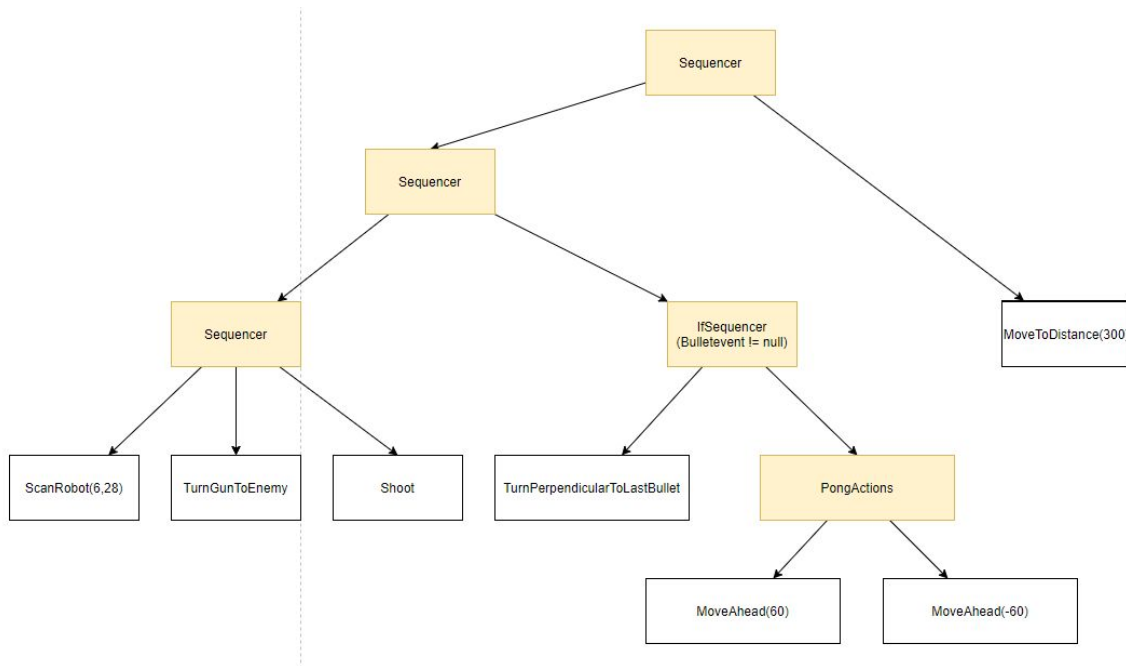
TurnPerpendicularToLastBullet()

Draait de robot zijn lichaam compleet links van waar de laatste bullet heeft geraakt.

TurnRadarToEnemy()

Draait de radar naar de enemy, als je de enemy al is hebt gezien.

Implementatie



Mijn behaviour tree speelt 3 behaviours af.

1. De robot scant voor een enemy en schiet hem.
2. De robot probeert de volgende bullet de dodgen.
3. De robot beweegt naar een afstand van 300.

De gedachte erachter was dat mijn robot op een afstand bleef schieten op de enemy. De robot zou dan heen en weer bewegen zodat de enemy hem moeilijk kan raken.

Reflectie

Ik snap nu hoe een Behaviour Tree in elkaar zit en waarvoor je hem kan gebruiken. Als ik een volgende keer een behaviour tree gebruik, dan zou ik sowieso meerdere behaviour trees tegelijk runnen, zodat ik meerdere taken in een update kan uitvoeren.