

KMINT

Terence Beijloos & Simon van Heertum

A*

Telt de gezette stappen bij elkaar op (als absolute waardes) en deelt die door 32, in verband met het stapformaat op de grid.

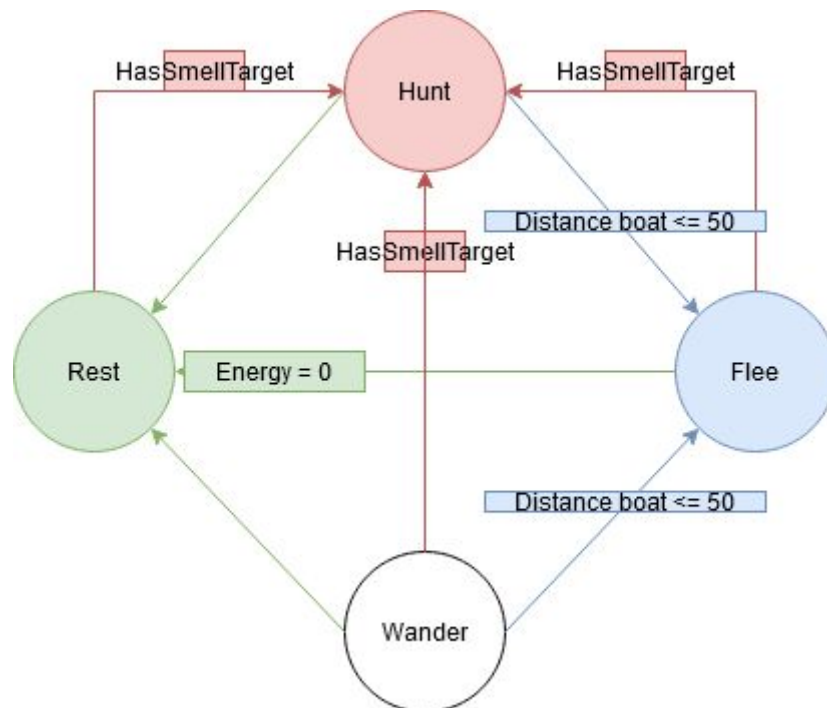
Als heuristiek hebben we gekozen voor de horizontale en verticale afstand bij elkaar op te tellen, omdat er geen diagonale beweging plaats kan vinden is dit altijd de kortst mogelijke route

Elke keer als een node toegevoegd wordt dan wordt de weg naar die node opnieuw toegevoegd gebruik makende van de edgeweights, zoals bij Dijkstra's algoritme.

In de states van zowel boot als shark worden de target nodes bijgehouden, in het geval van shark is dat de rustplaats en de node dichtst bij een varken, in het geval van de boot zijn dat de drie afmeerplaatsen. Het pad van de boot en shark worden beide berekend met behulp van A* aan de hand van de huidige locatie en het target tijdens het creëren van een nieuwe state.

Finite statemachine

Knabbel



Haaïen gedrag

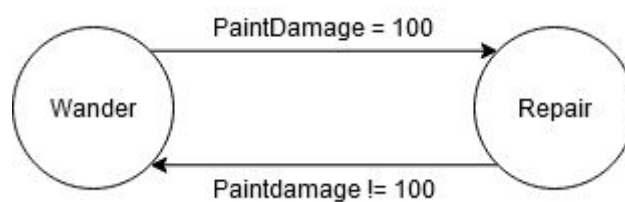
Het gedrag van de shark wordt iedere stap opgedeeld in drie delen: sense, think en move. In sense wordt de omgeving van de haai nagegaan.

In sense wordt er gecheckt of er een varken binnen een straal van 100 eenheden aanwezig is, zo ja dan wordt de dichtsts bij zijnde opgeslagen in de state. Vervolgens wordt gekeken of de boot binnen een straal van 50 is, zo ja dan wordt een boolean scared, op true gezet.

Daarna wordt in think de Volgende stap berekend aan de hand van het movement gedrag dat specifiek is voor het specifieke gedrag (Flee en Wander: een willekeurige node. Hunt en Rest: A* die bij het maken van de state berekend wordt aan de hand van een doorgegeven target). In deze stap wordt er ook gecheckt of er van gedrag veranderd moet worden Door hiërarchisch te checken of de, bij sense, gevonden variable leiden tot een overschrijvende state. Rest overschijft alles, wanneer energy 0 is wordt alles overschreven door rest behaviour. Vervolgens is flee het meest dominante, daarna volgt Hunting en tenslotte, als er geen aanleiding is voor een andere state komt wander aan de beurt.

Tenslotte wordt er overgegaan naar move. Hierin wordt de shark verplaatst naar de in think bepaalde locatie. Dit kan een locatie zijn die door het nieuwe gedrag berekend is omdat er in think een transition heeft plaatsgevonden.

Porcus Vincit



Boat gedrag

De boat is veel simpeler qua gedragingen het wandered en het gaat naar een dock voor reparaties. Daarvoor gebruikt het enkel de paintjob om te besluiten om gerepareerd te worden of niet. Maar de keuze voor welke haven het pakt is gebaseerd op een kan tussen 0 en 1. De aanmeerplaatsen krijgen in het begin allemaal dezelfde kans dus dat is

$1 \div \text{\#aanmeerplekken}$ wat in dit geval neer komt op 0.333333. Bij het overgaan van wander naa repair gedrag wordt er een willkeurige float gepakt tussen 0 en 1 en die wordt vergeleken met de kansen van de docks als volgt



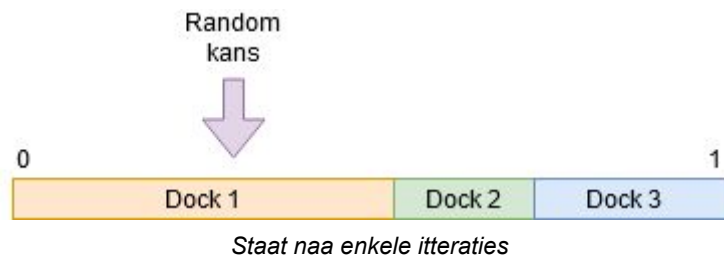
Initiele staat

Wanneer de reparatie heeft plaatsgevonden wordt de hoeveelheid van de reparatie beoordeeld als volgt: Als de reparatie 50 of hoger aan punten repareerde komt er 0.01 bij de kans bij van het doc dat de reparatie uitvoerde. Om vervolgens de som van de kansen op 1 te houden wordt 0.01 gedeeld door het aantal niet gekozen docs en wordt dat getal van de kans van die niet gekozen docs afgetrokken. Het omgekeerde gebeurd in de situatie dat een

doc een reparatie geeft van minder dan 50 punten. Dan wordt 0.01 afgetrokken van de gekozen doc en krijgen de niet gekozen docs er wat bij.

Wanneer er meer wordt gerepareerd dan 60 wordt het opgetelde getal verdubbeld en komt het boven de 80 uit dan gebeurt dit nogmaals. Zodat de kies kans een stuk meer vergroot wordt. Het tegenovergestelde geldt voor getallen onder de 40 en 20.

Dit leid tot een herverdeling van de kansen zoals het voorbeeld hieronder illustreert:



Force driven entities

De FDE hebben de krachten: alignment, walls, cohesion separation, flee en seek.

De som van alle vectoren geeft een uiteindelijke kracht die ervoor zorgt dat er een nieuwe heading gemaakt wordt. Wanneer een varken echter té dicht bij de muur komt dan worden de andere krachten overschreven.

Genetisch algoritme

De initiële populatie wordt volledig willekeurig opgezet.

Als chromosomen zijn gekozen: cohesion, separation, alignment, sharklikeness boatfetish. De eerste drie spreken voor zich maar de laatste twee zijn bepalend voor de aantrekking of afstoting ten opzichte van de haai en de boot.

Overleving(fitness) is de grootste fitness factor, wordt een vis opgegeten dan is de kans het kleinst, hij is echter niet nul want in de tijd voordat hij opgegeten is had hij zich voort kunnen planten. Overleeft een vis dan heeft die een grotere kans op voortplanten. Heeft een vis het schip geboord tijdens zijn leven dan heeft hij de grootste kans op voortplanten.

Het selecteren gaat op een semi willekeurige manier, er wordt voorrang verleend aan individuen met een hoge fitness. Deze twee individuen koppelen de helft van die chromosomen aan elkaar wat tot een nieuwe waarde leidt. Dit kan door de eerste helft van de een te koppelen aan de tweede helft van de ander of andersom. Daarna wordt er met een kans op mutatie van 1 op 100 het chromosoom veranderd met een bit manipulatie.

De uiteindelijke, nieuwe, populatie bestaat voor 50% uit kinderen en 50% hun ouders.