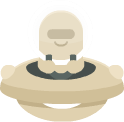
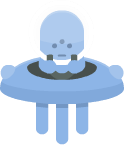
Kunstmatige intelligentie 2019/2020

Definitely not a plane, ufo-project.



# A\*

## Op welk moment wordt er gestopt met zoeken?

Het zoeken van A\* stopt wanneer de openList leeg is of wanneer het doel bereikt is.

## Wat is jullie heuristiek en waarom is deze gekozen?

We hebben gekozen voor een Manhattan heuristiek aangezien de actoren op de graaf slechts 4 richtingen hebben waar ze naar kunnen bewegen (noord, oost, zuid, west). Mocht in de toekomst een diagonale actie ook mogelijk worden zou er gewisseld kunnen worden naar een Euclidean heuristiek.

## Hoe kan een entiteit zijn doel bereiken als deze beweegt?

Een entiteit bekijkt voordat hij naar een bewegend doel gaat het pad van dit doel. Voor elke node van dit pad wordt de heuristiek berekend voor de entiteit naar de node in het pad en voor het doel en de node in het pad. Slechts wanneer de heuristiek van de entiteit naar de node kleiner is dan die van het doel legt deze het pad af. Zo niet blijft deze bij het laatste doel. Met deze strategie kan de entiteit beslissen om het doel te gaan onderscheppen, te verplaatsen naar een ander doel node of te wachten bij het laatste doel node.

# FSM

## Leg van de toestandsdiagrammen de toestanden en transities uit

## Tank state chart

### States

* Wandering; Moves tank to random adjacent node.
* TravelingToEMP; Moves tank to closest EMP location.
* TravelingToShield; Moves tank to closest shield location.
* FleeFromUfo; Moves to the furthest node away from the ufo.
* TravelToANWB; Moves to Andre from the ANWB.

### Transitions

* Wandering 🡪 TravelingToEMP; Transitie gebaseerd op kansberekening
* Wandering 🡪 TravelingToShield; Transitie gebaseerd op kansberekening
* Wandering 🡪 FleeFromUfo; Transitie gebaseerd op kansberekening
* TravelingToEMP 🡪 Wandering; EMP gevonden. Terug naar vorige.
* TravelingToShield 🡪 Wandering; Shield gevonden. Terug naar vorige.
* FleeFromUfo 🡪 Wandering; Gevlucht van ufo. Terug naar vorige.
* Collision UfoMeetsTank 🡪 TravelToANWB; (HP <= 0) als resultaat van collision. Ga naar Andre van de ANWB.
* TravelToANWB 🡪 Wandering; Andre gevonden en gerepareerd. Terug naar Wandering.

## Ufo state chart

### States

* Wandering; Travels by a random vector created every second.
* ChasingHuman; Chases human until collided or transitioned to ChasingTank
* ChasingTank; Chases tank until collided
* FreeUfo; has to wait 20 turns until it ufo can move again.

### Transitions

* Wandering 🡪 ChasingHuman; Transitie wanneer een mens wordt gedetecteerd.
* Wandering 🡪 ChasingTank; Transitie wanneer een tank initieel wordt gedetecteerd.
* ChasingHuman 🡪 ChasingTank; Transitie wanneer tijdens het chasen van een mens een tank wordt gedetecteerd.
* ChasingTank 🡪 FreezeUfo; Transitie wanneer de tank een EMP bezit tijdens de collision.

## Wat is de updatestrategie voor de kansen die is toegepast?

## Wordt er gebruik gemaakt van een globale toestand? Zo ja, hoe en waarom?

Ja, er wordt globaal bijgehouden of de simulatie ronden limiet is bereikt (>200). De tijdsinterval van een ronde is 1 seconden. Wanneer deze toestand gebeurt wordt er een nieuwe generatie van mensen gemaakt en worden de andere entiteiten gereset.

# FDE

## Welke krachten werken er op de force driven entities?

* Krachten beïnvloed door DNA:
  + Aangetrokkenheid tot rode tank waarde tussen -1 en 1
  + Aangetrokkenheid tot groene tank waarde tussen -1 en 1
  + Aangetrokkenheid tot UFO’s waarde tussen -1 en 1
  + Aangetrokkenheid tot deuren waarde tussen -1 en 1
  + Cohesion waarde tussen 0 en 1
  + Separation waarde tussen 0 en 1
  + Alignment waarde tussen 0 en 1
* Collision krachten
  + Collision met scherm
  + Collision avoidance
* Random kracht (om stilstand te voorkomen)

## Hoe worden alle krachten gecombineerd tot één resultante?

De aangetrokkenheid tot een actor wordt berekend door de vectors van de perceived actors naar de huidige human toe te berekenen en op te tellen in een vector. Vervolgens wordt de resulterende vector gedeeld door het aantal perceived actors. Tot slot wordt de vector genormaliseerd om evenredige impact te creëren.

De cohesion wordt berekend door de locatie van de perceived humans op te tellen in een vector. Vervolgens wordt de resulterende vector gedeeld door het aantal perceived humans, dit geeft de positite van het middelpunt van de kracht. Maar we willen de afstand van de huidige human naar het middelpunt van de kracht, om dit te krijgen trekken we de huidige locatie af van de positie van het middelpunt van de kracht. Tot slot wordt de vector genormaliseerd om evenredige impact te creëren.

De separation wordt berekend op dezelfde manier als de aangetrokkenheid, alleen wordt hier na het delen de vector omgedraaid door deze met -1 te vermenigvuldigen.

De alignment wordt berekend door de heading vector van de perceived humans op te tellen in een vector. Vervolgens wordt de resulterende vector gedeeld door het aantal perceived humans, dit geeft een gemiddelde heading. Tot slot wordt de vector genormaliseerd om een evenredige impact te creëren.

All deze krachten worden vervolgens opgeteld en genormaliseerd tot een kracht.

Naast deze berekende kracht worden er ook krachten berekend om collision af te handelen.

De screen collision kracht geeft een kracht berekend door de vector te berekenen vanuit de huidige actor naar het midden van het scherm. De vector wordt eveneens genormaliseerd.

De collision avoidance vector wordt berekend door voor alle perceived buildings een locatie vooruit te berekenen voor een beweging in de huidige richting voor een bepaalde verlopen tijd. Vervolgens bereken je de vector van de voorspelde positie naar de positie van de perceived building. Deze tel je allemaal op en deel je door het aantal perceived buildings. De vector wordt eveneens genormaliseerd.

Om tot een uiteindelijke kracht te komen worden al deze krachten opgeteld samen met een random kracht en wordt het resultaat genormaliseerd. Deze random kracht is alleen van kracht als er geen andere krachten werken, dit om stilstand te voorkomen.

## Hoe ontwijkt de entiteit verboden gebieden (zoals gebouwen)?

De screen collision wordt opgelost door een rectangle ter grootte van het scherm te maken, min een offset ter grootte van de collision range van de actor. Vervolgens kan er gebruik gemaakt worden van de contains functie om te kijken of de locatie van de actor in die rectangle valt. Zo niet dan is de actor dus aan het colliden met de rand van het scherm of er deels voorbij.

Om buildings te ontwijken wordt er gebruik gemaakt van de collision avoidence. De kracht wordt zoals beschreven opgebouwd.

# GA

## Hoe wordt de initiële populatie gegenereerd?

Er worden 100 humans gegenereerd met random waarden voor het dna, deze vallen tussen de gestelde minimum en maximum waarde voor dat attribuut.

## Welke fitnessfunctie wordt er toegpast en waarom?

De humans beginnen met een fitness van 200, elke seconde dat ze niet veilig zijn verliezen ze 1 fitness. Het doel is namelijk in veiligheid komen. Daarom zal de fitness van de human aan het einde van de ronde ook worden vermenigvuldigd met 10 als deze veilig is. Wanneer de human dood is wordt er vermenigvuldigd met 0.7, doodgaan verminderd dus de fitness.

## Hoe worden de parents voor de volgende generatie geselecteerd en waarom op de manier?

Er wordt een kans berekent door de fitness van de human te delen door de totaal geaccumuleerde fitness. Dit geeft een kans percentage voor elke human. Gebaseerd op die kans worden er twee humans geselecteerd.

## Hoe wordt de nieuwe generatie samengesteld?

De nieuwe generatie wordt samengesteld door de dna strings van de twee humans op een random index te splitten. Deel tot de split index van human 1 en deel na de split index van human 2. Vervolgens is er een 5% kans om te muteren, dit betekent dat er op een random index een random waarde tussen 0 en 1 wordt gezet.

# QL

## Welke problemen lost dit algoritme in deze simulatie op?

## Hoe werkt de updateregel van dit algoritme?

## Hoe snel wordt de taak geleerd in de simulatie?