Programmeren 4

Multimediatechnieken 2

# Opdracht EP3 : AI planning

Deze opdracht is een herhaling van de opgave in eerste zittijd, alleen – vermits veruit iedereen het A\* algoritme bestudeerde zullen we nu vragen om je algoritme te baseren op de ideeën van het RRT algoritme.

## Opdracht Game AI

Bedenk voor je robotvoetbalspel een finite state model (FSM) dat bruikbaar zou zijn om het samenspel van je spelers aan je simulatie toe te voegen. Welke rollen kan je onderscheiden, welke verschillende states of toestanden lijken je handig. Hoe kan je in een bepaalde toestand terechtkomen enz. Beschrijf dit FSM in je verslag.

Als 2de deel van deze opdracht ga je op zoek naar een goed path-planning algoritme voor je robots. **Je vertrekpunt is deze keer het RRT algoritme besproken in de nota’s.** Bedenk als eerste wat een robot precies met dit algoritme wil bereiken. Voor welke situatie zou path-planning van pas komen in je voetbalsimulatie? Implementeer deze situatie, en laat vervolgens hier je path-planning algoritme op los. Om goed te demonstreren dat je algoritme werkt, maak je een visualisatie van de werking van je algoritme.

Indien mogelijk, kan je deze situatie ook echt integreren met je oorspronkelijk voetbalsimulatie en deze als dusdanig uitbreiden met je path-planning algoritme.

Als illustratie vind je een demo op toledo. Zoals daar te zien verwachten we minstens:

* Weergave van startlocatie, verplaatsbare tegenspelers en eindlocatie.
* Weergave van de volledig opgebouwde boom en het gekozen pad.
* Weergave van statistische gegevens: totaal aantal knopen van de boom, totaal aantal knopen in het gekozen pas, lengte van het gekozen pad t.o.v. vogelvluchtafstand
* Weergave van de performantie van het algoritme: weergave van de rekentijd. Opgelet dit is de rekentijd van het algoritme en niet de tijd nodig om te tekenen. M.a.w je zal eerst moeten rekenen en dan pas tekenen.

Verslag

# AI

Er zijn 2 type spelers in de AI dat ik ga uitleggen: *veldspelers* en *doelmannen* bijden hebben een FSM maar ook het *voetbalteam* in zijn geheel heeft een FSM voor de mogelijkheid van het volledige team hun gedrag aan te passen naargelang het spel (het doel hiervoor kan je zien als bij RTS-games waar er ook in verschillende AI lagen word gewerkt: units, commanders, ... )

Voor het verspreiden van de berichten naar alle spelers is het gebruik van een message dispatcher en er zullen berichten verstuurd worden van *veldspelers* naar *veldspelers* (ook *doelmannen*) of van *voetbalteam* naar *veldspelers*

## Voetbalteam

Het volledige team zal altijd in 1 van de volgende staten zijn: **Verdedigen, Aanvallen** of **KickOff**. Elke speler in het team heeft een thuis locatie die aangepast word naargelang de staat van het team die ziet welke locatie de spelers best hebben in de huide situatie.

### Verdedigen

De thuis locaties van de spelers worden hier dicht bij de goal gezet en op hun kant van het veld. Zodanig dat de tegenstander moelijker door de verdediging kan graken en een punt maken

### Aanvallen

Vanaf het team de bal bezit, kom het team in de aanvallende status. Hier worden de thuis locaties van de spelers versprijd over het volledige veld. Dan staan de spelers veel dichter bij het doel van de tegenstander, maar er staan er nog ook bij hun eigen doel voor mocht de tegenstander de controle over de bal krijgen, dat er nog een verdediging aanwezig is

### KickOff

Een team komt in deze staat waneer een doelpunt is gemaakt of wanneer het spel begint en elke speler loopt dan naar zijn thuis locatie.Wanneer iedereen op zijn thuis locatie staat gaat elk team in de verdedigende staat en dan begint het spel

## Veldspelers

Er zijn 2 types van veldspelers: *aanvallers* en *verdedigers*. Verdedigers blijven voornamelijk vanachter op het veld, en aanvallers krijgen meer vrijheid, zij kunnen over het volledige veld rondlopen.

Een veldspeler zal altijd in 1 van de volgende 7 staten zitten. De staat van de speler veranderd naargelang hij zelf ziet of hij het bericht krijgt van een medespeler (gaat een pas krijgen bv)

* Wacht
* Ontvang\_Bal
* Schiet\_Bal
* Dribbel
* Achtervolg\_Bal
* Keer\_Terug
* Ondersteun

### Wacht

Hier wacht te speler op zijn huidige plaats voor verdere instructies. Wanneer bv hij dichter bij de goal zit van de tegenstander dan de speler met de bal, zal hij vragen om een pas te krijgen van de speler met de bal. Of wanneer de tegenstander paseert met de bal zal hij de bal gaan achtervolgen.

### Ontvang\_Bal

Een speler komt in deze staat terecht wanneer een andere speler een pas gaat geven. Er kan op een gegeven moment maar 1 speler in deze staat staan

### Schiet\_Bal

Dit kan zijn voor 2 verschillende zaken, voor een pas te geven naar een mede speler of wanneer in schietafstand van de goal, zal dit zijn voor te schieten naar de goal

### Dribbel

Wanneer het veld veilig is en de speler ziet dat hij gewoon vooruit kan gaan, zal hij met de bal richting de goal gaan

### Achtervolg\_Bal

Hier zal de speler kijken waar de bal is en dan voortgaan naar schiet afstand, wanneer hij deze berijkt heeft zal hij in de SchietBal staat gaan.

### Keer\_Terug

Wanneer de tegenstanders in balbezit komen en de speler moet terug naar zijn thuis locatie gaan zal hij in deze staat komen.

### Ondersteun

Waanneer een speler ondersteuning nodig heeft, zal hij een bericht sturen naar een speler en dan komt die speler in deze staat, en zal hij richting de speler lopen die ondersteuning nodig heeft.

## Doelman

* Tend\_Goal
* Keer\_Terug
* Bal\_Terug\_In\_Spel
* Onderschep\_Bal

### Tend\_Goal

Hier zal de doelman proberen tussen de bal in de goal tekomen zodanig wanneer er geschoten zou worden hij deze kan onderscheppen

### Keer\_Terug

Hier zal de doelman terug naar zijn goal lopen.

### Bal\_Terug\_In\_Spel

Wanneer de doelman de bal heeft, moet hij deze terug in het spel brengen door te passen naar een mede speler.

### Onderschep\_Bal

Wanneer de bal word richting de goal geschoten zal de doelman deze proberen onderscheppen. Vergeleken met TendGoal is hij hier weldegelijk in staat om de bal te pakken, terwijl bij TendGoal deze nog uit zijn bereik is