Ilmari Lehtinen

596624

Bioinformaatioteknologia, 2.vuosikurssi

07.03.2019

Tekninen suunnitelma

Ohjelman rakennesuunnitelma

Ohjelmassa tulee olemaan alustavasti ainakin seuraavanlaisia luokkia:

- Vehicle:
    - Attribuutit:
        - location
        - velocity
        - acceleration
        - size/radius
        - maxspeed, maxforce
    - Metodit:
        - TÄNNE JOKO OHJAUSALGORITMIT TAI SITTEN OMIIN LUOKKIIN
        - Seek(self, target)
        - Arrive(self, target)
        - PathFollow(self, Path(luokka))
        - CollisionAvoidance(self, near_targets(list?))/separation
        - ApplyForce(): Metodi joka päättää, mitä ohjausalgoritmiä käytetään ja millä painokertoimilla, esim tilanne jossa monta autoa seuraa tietä ja joutuvat samalla väistelemään toisia. Painokertoimet mahdollisesti käyttäjän päätettävissä tai sitten suoraan koodista
        - ongelma? kuinka vehicle havaitsee ympäristöään? jollain tietyllä säteellä?

- Coordinates/Location:
    - Attribuutit:
        - x,y (pikseleitä/laatikoita?)
    - Metodit:
        - get x,y()
- Direction/Facing:
    - Attribuutit:
        - 360 angle?
    - Metodit:
        - getters
- Map/World

    - Pitää yllä tietoa kartasta, lista autoista
    - Miten kartta tallenetaan, 2-uloitteinen lista pikseleistä/isoimmista neliöistä, kuten tehtävässä RobotWorld?
- Path:
    - Luokka teitä/reittejä varten, jota autot voivat seurata, käytännössä lista pisteistä, jotka muodostavat polun
- GUI:
    - Luokka, joka huolehtii käyttöliittymän piirtämisestä ja käyttäjän kanssa kommunikoinnista
- Testiluokka

Käyttötapauskuvaus

Ohjelman käynnistyessä ruudulle piirtyy käyttöliittymä, johon piirtyy valmiiksi kartta. Käyttöliittymän reunassa on sivupalkki josta voi mahdollisesti säätää parametrejä simulaatiolle(kuinka paljon autoja, maxforce/maxspeed jne.) Tämän jälkeen käyttäjä voi aloittaa simulaation ja katsoa kuinka se etenee. Simulaation kesken käyttäjä voi mahdollisesti luoda uusia autoja, uusia maalipisteitä autoille jne. Käyttäjä ei kuitenkaan ohjaa autoja manuaalisesti vaan autot liikkuvat itsestään ohjausalgoritmien avulla. Tässä siis aktiovituu aluksi GUI jonka jälkeen muut luokat seuraavat perässä. Käyttäjä voi pysäyttää tai aloittaa simulaation alusta halutessaan.

Algoritmit

- **Seek:**
    - Lasketaan haluttu nopeusvektori: desired = normalize (target – location) * max_speed
    - Lasketaan suuntavektori autolle: steer = desired – velocity
    - Algoritmi siis etsii halutun pisteen kartalta ja jää ajamaan sen läpi edes takas
- **Arrive:**
    - Sama kuin Seek, kun auto on kaukana kohteesta. Kun auto lähestyy kohdetta, piennetään nopeusvektoria lineaarisesti, jolloin auto pysähtyy kohteeseen. Vektorin suuntaa ei muuteta, ainoastaa pituutta.
- **PathFollow:**
    - Ennustetaan auton tuleva paikka, jos pisteen projektio reitin keskikohtaan(normaali) on pienempi kun reitin ympärillä kulkevan sylinterin säde(”reitin varrella”), ei tarvitse tehdä mitään, auto on reitin varrella.
    - Jos tulevan paikan projektio reitille on suurempi kuin säde, etsitään lähin piste reitillä(normaali), jonka jälkeen siirretään sitä hiukan eteenpäin ja asetetaan se targetiksi. Tämän jälkeen kutsutaan Seek() kohti uuttaa targettia. repeat

- **Separation**:
    - Tutkitaan auton läheisyyttä jollain säteellä r, etsitään löytyykö siitä yhtään autoa, jos ei löydy, kaikki ok
    - Jos läheisyydessä on autoja:
        - miinustetaan kyseisen auton lokaatio muista autoista, normalisoidaan, ja kerrotaan jollain parametrillä
- **Obstacle avoidance/muiden autojen väistäminen:**
    - Tutkitaan auton nykyistä kulkusuuntaa, jos jonkun toisen auton ennustettu kulkusuunta on tarpeeksi lähellä, tehdään väistöliikkeitä.
    - tähän sopii ehkä myös flee, katsotaan miten toimii
- **Mahdolliset reitinhaku algoritmit:**
    - dfs, bfs, a*?

Tietorakenteet

Miten kartta kuvataan, että voidaan tutkia autojen paikkoja ruudulla? Kaksiulotteinen taulukko koko kartasta, jossa olisi mahdollisesti tietoa siitä onko kyseisessä kohdassa auto vai seinä vai tyhjää. Koostuuko kartta pelkistä pikseleistä vai isommista neliöistä jossa voi olla yksi auto kerrallaan?

Jos kartta koostuu tieverkoista, voidaan käyttää puita/verkkoja tietorakenteina?

Aikataulu

- Toimivan graafisen käyttöliittymän luonti: 20h
    - autojen piirtäminen ruudulle, valikot jne.
- Autojen liikuttaminen ruudulla ja yksinkertaisia ohjausalgoritmejä: 30h
    - miten esim kaksi autoa reagoi keskenään jne.
    - vektorilaskenta kuntoon
- Useampien autojen liikennekäyttäytyminen keskenään: 30h
    - path following usealla autolla
- Monimutkaisempia tierakenteita, lisäominaisuuksia: 30h
- Testaus/muuta hienoa: 20h

Yksikkötestaussunnitelma

Tärkeimmät testit, että autojen ohjausalgoritmit toimivat, käyttöliittymä toimii halutusti, autot eivät törmäile seiniin jne. voidaan testata tilanteita joissa annetaan outoja parametrejä autoille ja katsotaan miten algoritmit suoriutuvat. tämä hiukan kesken

Kirjallisuusviitteet ja linkit

Youtube: Nature of code 6,
https://www.youtube.com/watch?v=JIz2L4tn5kM&index=1&list=PLRqwX-V7Uu6YHt0dtyf4uiw8tKOxQLvlW

Steering Behaviors For Autonomous Characters,

http://www.red3d.com/cwr/steer/gdc99/

Python,

https://docs.python.org/3.5/reference/index.html#reference-index

https://docs.python.org/3.5/library/index.html

PyQT5,

http://pyqt.sourceforge.net/Docs/PyQt5/index.html

Liitteet