

Proiect Proiectarea Algoritmilor 2011

I. Descrierea temei

Proiectul din cadrul materiei Proiectarea Algoritmilor va presupune realizarea unui modul de conducere a unei masini in cadrul unei curse simulate pe calculator. Scopul produsului final va fi parcurgerea cat mai rapida a unui circuit dat.

II. Desfasurarea cursei

Cursa va fi specificata prin:

- O harta 1BPP (1 bit per pixel, adica un bitmap monocromatic alb negru) in format .BMP ce va specifica prin **negru portiunea circulabila** din cadrul hartii. Un bit care are valoarea 1 va insemna negru (deci portiune circulabila), in timp ce un bit cu valoarea de 0 va insemna alb (asadar portiune necirculabila)
- Dimensiunile echivalente in metri pe x si y ale hartii in format .BMP
- Coordonatele carteziene (exprimate in metri) ale punctului de start
- Sensul (orar/antiorar) de parcurgere al traseului
- Unghiul (rad) de orientare al vehiculului la start, ales in conformitate cu sensul de parcurgere
- Numarul de ture
- Timpul minim de referinta (s) per tur de circuit (in cazul in care o masina nu reuseste parcurgerea unui tur in acest interval, ea va fi descalificata)

Echipele vor alinia la startul unei curse cate doua masini fiecare. Pentru o personalizare corespunzatoare masinile vor fi specificate prin urmatoorii parametri:

- Culoarea (specifica echipei)
- Acceleratie (m/s^2) = ACC
- Frana (m/s^2) = BRK
- Viteza maxima (m/s) = SPD
- Manevrabilitate (rad/s) = STR
- Constrangere (default): $10 \cdot \text{ACC} + 10 \cdot \text{BRK} + \text{SPD} + 200 \cdot \text{STR} = 400$

Modulul de AI va avea posibilitatea de a raspunde in interval de 30 secunde dupa prezentarea hartii cursei cu parametrii doriti ai masinii (**ACC/BRK/SPD/STR** astfel incat **$10*ACC+10*BRK+SPD+200*STR = 400$**). In cazul in care modulul nu raspunde in timp util vor fi setate valorile default si anume: 10/10/100/0.5

- Vor fi acordate urmatoarele punctaje:
 - o Loc 1: 7p
 - o Loc 2: 4p
 - o Loc 3: 3p
 - o Loc 4: 2p
 - o Loc 5: 1p
 - o Ultimul loc: -1p
 - o Descalificare: -2p
- Intre masina cu cel mai bun rezultat al unei echipe (MAX) si cea cu rezultatul mai slab (MIN) punctajul acumulat de echipa va fi: **$(MAX * 7 + MIN * 3) / 10$**
- Dupa terminarea curselor se efectueaza clasamentul prin insumarea punctelor obtinute
- Primele doua echipe vor participa intr-o noua cursa, pentru departajare, fiecare cu cate o singura masina

III. Observatii simplificatoare

- Masinile nu se ciocnesc unele de altele, evolueaza in cadrul pistei fara restrictii si prin urmare pot ocupa aceleasi pozitii in spatiu.
- Singurele coliziuni sunt cele dintre masini si marginile circuitului, moment in care viteza masinii devine zero. Repornirea masinii este posibila numai pe o noua directie care se indeparteaza de margine
- Acceleratia maxima si frana maxima sunt constante indiferent de viteza
- Nu se ia in considerare masa vehiculelor
- Vehiculele se misca intotdeauna pe directia indicata de volan, indiferent de viteza acestora
- Pozitiile si calculele intermediare se transmit in format Floating point (double precision), pozitionarile in cadrul hartii (pentru validare traseu) sau pe ecran (pentru afisare) se fac prin rotunjire la cel mai apropiat intreg.

IV. Arhitectura Proiectului

Proiectul este compus din 3 componente:

1) Componenta Viewer, care va permite setarea parametrilor initiali ai cursei (explicate in cele ce urmeaza) si vizualizarea in timp real a curselor de masini. Functiile viewer-ului sunt:

- Incarcarea hartilor dintr-un fisier selectabil din interfata viewer-ului
- Alocarea punctelor si orientarilor de start, a numarului de ture, a timpului de referinta, a sensului de parcurgere, etc.
- Alegerea culorilor participantilor
- Conexiunea cu serverului si update-area continua a situatiei in functie de pozitiile venite de la server pentru fiecare masina
- Nu efectueaza calcule ci doar citeste pozitii specificate prin coordonate in sistem metric si afiseaza masinile in pozitiile specificate utilizand culoarea selectata pentru fiecare echipa
- Afiseaza rezultatele cursei, punctele obtinute si clasamentul dupa fiecare cursa
- Afiseaza harta, vehiculele, timpul scurs de la start, parametrii fiecarui vehicul si ordinea in cursa. Afisarea se face in timp real, in mod grafic.

2) Componenta Server, care va realiza calculele fizice necesare miscarii masinilor pe pista, pe baza comenzilor primite de la componenta de AI.

- Initializeaza conexiuni cu viewer-ul cat si cu cele $2N$ masini ale celor N echipe in cazul unei confruntari normale sau ale celor 2 masini pentru finala. **Conexiunea celor $2N$ procese-masini se pot initia de la masini către server (fizica aplicatiei).**
- Asteapta initializarea modulelor AI, dupa transmiterea hartii catre acestea, un interval de 30 de secunde pentru calculul parametrilor masinilor
- Comunica viewer-ului starea in cursa: ordine, timpi, pozitii, parametri de rulare pentru fiecare vehicul in parte
- Citeste periodic de 10 ori pe secunda de la fiecare modul de AI deciziile de conducere ale vehiculului asociat, calculeaza noile pozitii ale acestuia, noul unghi de orientare al volanului si confirma AI noile pozitii ale tuturor participantilor, viteza curenta si noile directii de deplasare (**X, Y, speed, steer**). Confirmarea presupune transmiterea componentei AI pozitiiile curente la care se afla in momentul in care se cere urmatoarea decizie din partea AI-ului. Astfel este posibila crearea unui algoritm adaptiv.
- Limiteaza viteza de deplasare a unui vehicul la **SPD**
- Calculeaza noile pozitii ale vehiculelor, unghiurile de orientare pe baza unei aproximari discrete cu un pas de **100 ms**.

- 3) Componenta AI, care va controla masina pe pista pe baza unor comenzi (explicate in cele ce urmeaza). Functiile componentei de AI sunt:
- Comunica server-ului de 10 ori pe secunda decizia curenta de conducere a vehiculului si anume:
 - o Franare in procent [0-100] din **BRK** sau acceleratie in procent [0-100] din **ACC**
 - o Cerere noua traiectorie de deplasare in unghiul [0-2*PI] (masurat in valoare absoluta fata de OX). Volanul va fi miscat (de catre componenta server) catre pozitia ceruta cu viteza **STR** , in sensul ce asigura o rotatie minima.

Dupa cum s-a specificat la inceput, scopul priectului este realizarea componentei AI, restul componentelor (Viewer, Server) va vor fi puse la dispozitie pentru a va putea dezvolta si testa solutiile.

V. Etapele proiectului

Proiectul va cuprinde 5 etape, dupa cum urmeaza:

- 1) **Etapa 1** presupune formarea unor echipe de cate 3-4 persoane. Echipele se pot forma intre oricare 3-4 studenti din aceeasi serie. Nu se permit echipe cu oameni din serii diferite. Fiecare echipa va desemna un membru al acestia pe post de capitan. Capitanul va, printre altele, rolul de a trimite in numele echipei sale solutiile pentru fiecare din etapele ramase.

Aceasta etapa va avea **2 saptamani**. Punctajul alocat acestei etape este de **0 puncte**.

- 2) **Etapa 2** presupune familiarizarea cu cele doua componente cu care se va lucra (server, viewer). Cerinta pentru aceasta etapa este miscarea masinii pe harta. Masina **nu** trebuie sa se miste intr-un mod coerent (se poate izbi de primul zid si bloca).

Aceasta va avea **2 saptamani**. Punctajul alocat acestei etape este de **0.4 puncte**.

- 3) **Etapa 3** presupune crearea unui algoritm ce va permite masinii terminarea a cel putin **2 ture** pe o harta la prima vedere. Nu exista un timp minim de terminare a celor doua ture, dar masina nu trebuie sa se blocheze pe circuit.

Aceasta etapa va avea **3 saptamani**. Punctajul alocat acestei etape este de **0.4 puncte**.

- 4) **Etapa 4** presupune parcurgerea hartilor de la etapa 3 intr-un timp mediu minim. Prin timp mediu minim se intelege timpul total pentru parcurgerea turelor impartit la numamrul de ture de efectuat. Timpul mediu minim va fi ales unul rezonabil de catre echipa de PA si va fi comunicat in timp util.

Aceasta etapa va avea **3 saptamani**. Punctajul pentru aceasta etapa este de **0.4 puncte**.

- 5) **Etapa 5 – Concurs.** Aceasta etapa va fi dedicata concursului pentru stabilirea celei mai bune component de AI (cea care scoate cei mai buni timpi). Punctajul pentru aceasta etapa va fi de **0.8 puncte** si se va acorda **exclusiv** pe baza clasamentului final.

VI. FAQs

Q: In ce limbaj se poate realiza proiectul?

A: In orice limbaj doriti ce suporta comunicare prin sockets.

Q: Punctajele pentru fiecare etapa vor fi binare (0 daca nu merge sau maxim daca merge)?

A: La etapele 2, 3 se va testa pe mai multe harti de dificultati diferite (numar de curbe, latimea drumului, etc). Asadar daca nu indeplineste conditiile pe unele dintre harti, AI-ul vostru tot va obtine un punctaj direct proportional cu numarul de harti pe care reuseste sa indeplineasca conditiile. La etapa 4 in functie de cat de aproape sunteti de timpul de referinta stabilit se acorda un punctaj direct proportional.

Q: Care este protocolul de comunicare intre server si componenta AI?

A: Pentru detalii legate de protocolul dintre server si AI consultati documentatia care vine cu serverul. Acolo se detaliaza pe larg tot protocolul si contine tot ce este necesar sa stiti pentru a putea face partea de comunicare sa mearga.

Q: Comunicarea implica programare cu socketi, dar noi nu am facut inca la PC socketi si pana vom face va expira deadline-ul pentru primele 2 etape. Ce facem?

A: S-a creat un schelet de cod in C++ (Windows) si Java care adauga un layer de abstarctizare peste socketi. Astfel voi nu trebuie sa va preocupati cu amanunte legate de socketi (decat daca vreti, caz in care sunteti pe cont propriu).

Q: Testat local (server si AI pe acelasi calculator) AI-ul merge bine, dar cand testem in retea AI-ul pare sa fie decalat si nu mai merge asa bine. Mentionez ca retea pe care testam are latenta de maxim 10ms.

A: Asta se datoreaza cel mai probabil unei latente provenite din cum gestioneaza sistemul de operare pachetele TCP/IP, mai precis algoritmul lui Nagle. Serverul incearca sa evite pe cat posibil problemele ce pot aparea din aceasta cauza, dar este posibil ca ele sa mai apara. Datorita acestui fapt este de preferat ca voi sa dezvoltati un algoritm care sa fie tolerant la mici variatii datorate unor factori precum Nagle.

Q: Exista o solutie exacta la problema algoritmica din spatele proiectului?

A: Putin probabil. Ideea proiectului este sa faceti un algoritm care sa mearga. Nu trebuie sa mearga perfect, nu trebuie sa fie o solutie exacta. Trebuie doar sa aveti o solutie destul de buna cat sa luati punctajul pe etapele 2-4 si una mai buna decat restul echipelor pentru etapa 5.

Q: Cum va fi structura concursului?

A: In functie de numarul de echipe de la final se va decide si structura concursului: grupe de cate n echipe sau toate echipele deodata.

Q: Cum vor arata hartile pe care se vor testa AI-urile?

A: Va vom pune la dispozitie harti similare cu cele ce vor fi folosite in cadrul testarii. Hartile efective pe care se va testa vor fi insa la prima vedere.

Q: Fiecare pixel reprezinta un metru sau cum?

A: Harta este specificata prin imaginea .BMP din care se extrage informatia despre traseu si dimensiunile in metri. Astfel daca o imagine .BMP are dimensiune de 1024x768 iar dimensiunile in metri sunt de 2048x1536 fiecarui metru patrat ii corespunde un bloc de 2x2 metri patrati. Invers daca dimensiunea in metri este de 512x384 fiecarui metru patrat ii va corespunde un bloc de 0.5x0.5 metri patrati. In functie de cum se seteaza scalarea hartii unui metru patrat ii pot corespunde blocuri de dimensiuni NxM (N poate diferi de M). Altfel spus harta se poate deforma pe OX si/sau pe OY fata de forma initiala, in functie de dimensiunile in metri specificate.

Q: La fiecare etapa cum se trimit rezolvarile aferente?

A: La fiecare etapa veti incarca pe curs codul solutiei voastre, precum si un makefile pentru a se putea face build automat. In loc de makefile puteti da un scrip, batchfile, ant file sau orice altceva care indeplineste acelasi scop. Dupa executarea sistemului vostru de build va trebui sa rezulte un program care sa se poata conecta la server si sa idneplineasca cerintele etapei aferente.

Q: Avem voie cu solutii multithreading?

A: Nu, deoarece la PA se studiaza algoritmi care nu sunt nici paraleli nici distribuiti.

Q: Avem voie sa folosim functia select() de multiplexare?

A: Da.