

Tema 2 - Laser Tag

Responsabili:

- Emil Racec [mailto:emil.racec@gmail.com]
- Laurențiu Piciu [mailto:laurentiupiciu@gmail.com]

Deadline: 28 aprilie 2017, ora 23:55

Modificări și actualizări

- **12 aprilie**
 - adăugare detalii privind structura fișierului Makefile și modul de citire, respectiv scriere

Obiective

În urma realizării acestei teme studentul va fi capabil sa:

- isi imbunatateasca abilitatile in utilizarea limbajului C++
- sa imparta o problema complexa in sub-componente pe care sa le rezolve individuala
- sa utilizeze structura de date hashtable pentru maparea obiectelor jucator cu numele jucatorilor
- sa sorteze o colectie de date dupa criterii multiple (valoare si respectiv alfabetic)
- sa contruiasca si sa utilizeze grafuri pentru o implementare eficienta a problemei
- sa elaboreze o rezolvare complexa, la o cerinta inspirata din realitate

Intro

Fiind unul dintre cele mai populare jocuri studentesti, Laser Tag este un joc de strategie care se desfasoara in mediul real, pe echipe, si simuleaza o batalie la sol. Mai exact, fiecare jucator primeste un „echipament de lupta” format dintr-o arma de mana ce emite o raza in infrarosu in momentul tragerii si un costum special ce detecteaza aceste raze. Echipamentul hi-tech este menit sa simuleze si monitorizeze schimbul de focuri, si avertizeaza un jucator cand a fost impuscat sau a impuscat un adversar. Astfel participantii, impartiti in doua echipe, trebuie sa impuste cat mai multi adversari, fara sa fie omorati, jocul terminandu-se cand una dintre echipe a fost nimicita (nu mai exista supravietuitori).

Nimic nu se compara cu atmosfera unica din timpul jocului: coordonare in echipa, explorarea teritoriului, ambuscade, capcane, adrenalina si bucuria victoriei. Insa la finalul jocului, toti ne dorim sa retrim anumite scene, sa regandim anumite strategii, si mai ales sa analizam performantele jucatorilor si a echipelor.

Enunt

Dandu-se inregistrările senzorilor pe parcursul unui campionat de Laser Tag, alcatuit dintr-o serie de reprize, ne propunem sa obtinem statistici relevante despre jucatori si echipe, precum:

- **top_shooters:** Top 5 cei mai buni tragatori
- **top_explorers:** Top 5 cei mai buni strategii - exploratori
- **top_fire_exchange:** Top 5 cele mai frecvente schimburi de focuri (intre 2 jucatori)
- **score:** scorul final pe echipe, dupa o serie de jocuri
- **winning_chance:** sansa de castig a echipelor inainte de terminarea jocului

Obs: In cazul in care exista un numar total de pozitii mai mic de 5, se vor afisa toate pozitiile din clasament.

Obs2: In caz de egalitate, jucatorii se vor afisa in ordine alfabetica.

Obs3: In cazul in care ultima pozitie din clasament (pozitia 5) este egala cu urmatoarea/urmatoarele, se vor afisa **toate pozitiile** egale.

Cerinta

Datele furnizate in timpul jocul Laser Tag (inregistrarile senzorilor)

In cadrul unui joc, sunt colectate date de la doua tipuri de senzori:

- **senzori de pozitie:** care se activeaza atunci cand un jucator trece pe langa unul din senzori. Aceste date sunt transmise sub forma `senzor_no: player_name`.
- **senzori de impuscare:** ce furnizeaza date despre impuscaturi reusite, sub forma de perechi de nume `shooter_name → victim_name`.

Obs: pentru simplificarea jocului, un luptator este ucis dupa exact 2 impuscaturi. Practic, toti participantii incep cu 2 vietii. De asemenea, nu se tine cont nici de alti factori complementari, precum: locul impuscaturii, durata dintre impuscaturi, timp de vindecare, etc.

Obs2: Odata omorat, un jucator nu mai activeaza pe campul de lupta, si implicit nu mai este inregistrat de senzori.

Pe baza datelor furnizate de senzori, vom calcula statisticile prezentate in sectiunea Enunt. Acestea se impart in doua categorii:

- **rezultate locale:** la nivel de joc, obtinute la finalul fiecarei reprize
- **rezultate generale:** la nivel de campionat, obtinute dupa jucarea tuturor reprizelor.

Date furnizate la inceputul campionatului Laser Tag

Un campionat de Laser Tag poate avea 1 sau mai multe jocuri. In cadrul campionatului, atat jucatorii cat si senzorii de pozitie nu se schimba. Aceste informatii se comunica intotdeauna la inceputul campionatului, astfel:

- **senzorii de pozitie:** Sunt n senzori de pozitie, numerotati de la 0 la $n-1$. Acestia sunt specificati printr-o pozitie relativa fata de ceilalti senzori. Astfel, fisierul de intrare va contine o linie cu un numar n , reprezentand numarul de senzori, urmat de n linii a cate $n-1$ valori reprezentand distantele fata de ceilalti senzori. De exemplu: pe linia i sunt precizate $n-1$ distante ale senzorului i fata de toti vecinii lui, in ordine crescatoare.

- **numele jucatorilor:** Se dau doua liste de nume, aferente jucatorilor celor doua echipe, astfel: pe prima linie doua numere naturale, nevide: n si m reprezentand numarul de jucatori, urmate de doua linii continand n respectiv m nume, sperate prin spatiu.

Obs: Numele jucatorilor sunt compuse numai din caractere alfabetice (deci nu vor contine numere sau spatii).

Vom detalia in continuare formulele de calcul pentru fiecare dintre acestea.

Best shooters

Cel mai bun tragator este definit drept luptatorul care a impuscat cei mai multi adversari. Astfel, in formula de calcul se tin cont de 2 parametri: numarul de jucatori adversari impuscati si numarul de coechipieri impuscati. Pentru fiecare adversar impuscat, jucatorul primeste $+2p$ iar pentru fiecare coechipier impuscat $-5p$. Sunt valide si scoruri negative.

Top explorers

Miscarea jucatorilor pe terenul de lupta poate fi deseori decisiva pentru castigarea unei batalii. Aceasta depinde de antrenamentul fizic al jucatorilor, abilitatea de explorare si surprindere a inamicului, recunoasterea suprafetei de duel, etc. Pentru calcularea scorului `top_explorer` vom tine cont de distanta estimativa parcursa si de diversitatea terenului explorat – plecand de la datele furnizate de senzorii de miscare. Scorul se calculeaza dupa urmatoare formula:

`top_explorer = total_distance + 3 * unique_sensors`, unde:

- `total_distance`: distanta totala parcursa (estimata). Se calculeaza in functie de senzorii activati de luptator, pe parcursul partidei
- `unique_sensors`: numarul de senzori unici activati de luptator, pe parcursul partidei.

Exemplu

Jucatorul „Bob” activeaza urmatoarii senzori pe parcursul partidei, in aceasta ordine: 1 2 1 1 3 4 1 3. In acest caz:

`total_distance = Distance(1→2) + Distance (2→1) + Distance (1→3) + Distance (3→4) + Distance (4→1) + Distance (1→3)`

`unique_sensors = #{1, 2, 3 4} = 4`

Obs: trecerea repetata consecutiva prin dreptul aceluiasi senzor nu se considera.

Obs2: Scorul `top_explorer` se calculeaza conform formulei, la nivel de joc, pentru fiecare jucator in parte. Prin urmare, scorul final (general) `top_explorer` al unui jucator va fi calculat ca suma a scorurilor `top_explorer` ale acelui jucator, din toate jocurile (inclusiv ultimul).

Top fire exchange

O statistica interesanta o reprezinta topul celor mai acerbe dueluri. Definim un “duel” ca fiind schimbul de focuri intre doi adversari. Valoarea unui duel este suma impuscaturilor intre cei doi jucatori, pe parcursul campionatului.

```
duel(x, y) = #shoots(x→y) + #shoots(y→x)
```

Obs: In afisarea unui duel, cei doi jucatori vor fi afisati in ordine alfabetica.

Team score

Reprezinta scorul final, pe echipe, la finalul campionatului. Intuitiv, o victorie este recompensata cu 1p, scorul final reflectand numarul de partide castigate de fiecare echipa. Un joc se termina atunci cand toti jucatorii unei echipe au fost omorati sau la expirarea timpului de joc. Cel de-al doilea scenariu se petrece intotdeauna doar la ultimul joc, deoarece un campionat de Laser Tag are o durata fixa, insa durata partidelor poate varia. Pentru a putea atribui un scor si in acest caz, vom calcula o probabilitate de castig pentru fiecare echipa, conform metricii prezentate in continuare, `winning_chance`.

Winning chance

Calcularea probabilitatii de a castiga ultima repriza este necesara in conditiile in care aceasta nu se poate termina, ca urmare a expirarii timpului alocat campionatului sau o defectiune majora a sistemului. Formula are la baza statisticile acumulate despre jucatori si starea jocului curent (i.e. jucatorii ramasi in viata la acel moment):

```
winning_chance = sum(alive[i] * (best_shooter[i]/max(best_shooter) +  
(top_explorer[i] / max(top_explorer)))), i=0:n-1
```

unde:

- `alive[i]`: 1 - daca jucatorul este in viata la incheierea ultimei reprize, 0 - altfel
- `best_shooter[i]`: scorul general `best_shooter` al jucatorului `i`
- `top_explorer[i]`: scorul general `top_explorer` al jucatorului `i`

Echipa cu valoarea `winning_chance` mai mare va fi declarata echipa castigatoare a ultimei reprize.

Exemplu de functionare

Va prezentam in continuare un exemplu de campionat, alaturi de formatul de intrare si iesire al programului de statistica.

Date de intrare (exemplu)

Ordinea datelor de intrare este urmatoarea:

1. Detaliile senzorilor de pozitie: numarul si pozitionarea relativa
2. Nume jucatorilor din cele doua echipe
3. Tag-ul `JOC_X`, unde `X` este numarul jocului. Acest tag marcheaza inceperea unui nou joc
4. Output-ul senzorilor din joc
5. (...) 3. si 4. se repeta pentru fiecare joc jucat
6. Tag-ul `END_CHAMPIONSHIP` ce marcheaza expirarea timpului pentru campionat si intreruperea **brusca** a ultimului joc.

Obs: In cazul ultimului joc, se garanteaza ca acesta este intrerupt mereu. Cu alte cuvinte, de fiecare data este necesar ca pentru acest ultim joc sa se calculeze sansa de castig, si nu se poate obtine direct un scor, ambele echipe avand inca jucatori supravietuitori.

```
4
7 7 5
7 2 4
7 2 3
5 4 3
2 3
Ana Codrin
Diana Elena Matei
JOC_1
1: Elena
0: Diana
0: Ana
Ana -> Diana
Elena -> Codrin
2: Codrin
2: Codrin
0: Matei
3: Codrin
Matei -> Ana
2: Matei
Matei -> Elena
2: Diana
Elena -> Ana
Diana -> Codrin
1: Matei
JOC_2
3: Diana
Ana -> Diana
3: Ana
Ana -> Diana
2: Elena
Elena -> Codrin
Codrin -> Matei
1: Matei
3: Elena
1: Codrin
Matei -> Codrin
Ana -> Elena
Ana -> Elena
0: Ana
0: Matei
Ana -> Matei
0: Elena
JOC_3
1: Matei
Codrin -> Diana
2: Codrin
2: Diana
Matei -> Ana
END_CHAMPIONSHIP
```

Date de iesire (output exemplu)

Programul vostru va afisa urmatoarele date la iesire, in ordine:

1. **top_shooters:** Top 5 tragatori
2. **top_explorers:** Top 5 strategii - exploratori
3. **top_fire_exchange-ranking:** Top 5 dueluri
4. **score:** scorul final, care include estimarea data de `winning_chance`

```
I. Top shooters
1. Ana 12p
2. Elena 6p
3. Codrin 4p
```

```
4. Diana 2p
5. Matei 1p

II. Top explorers
1. Matei 34p
2. Elena 20p
3. Diana 19p
4. Codrin 15p
5. Ana 14p

III. Top fire exchange
1. Ana - Diana 3
2. Ana - Elena 3
3. Ana - Matei 3
4. Codrin - Diana 2
5. Codrin - Elena 2
6. Codrin - Matei 2

IV. Final score
1 - 2
```

Obs: Valorile `winning_chance`, trunchiate la 3 zecimale, pentru exemplul dat sunt: 2.186 si respectiv 2.897

Indicii de implementare

Pentru rezolvarea eficienta si eleganta a temei, este recomandat sa utilizati doua grafuri:

- unul pentru simularea miscarii luptatorilor in teren si calcularea statisticilor aferente
- si unul pentru simularea schimbului de focuri si calcularea statisticilor aferente

Se recomanda utilizarea unui hashtable pentru maparea directa a numelor jucatorilor cu obiectele corespunzatoare (sau cu indexul acestora).

Checker + teste publice

Puteți descărca checker-ul, cu testele publice aferente, de aici: [Checker](#)

Reguli pentru trimitere

Temele vor trebui trimise pe vmchecker [<https://elf.cs.pub.ro/vmchecker/ui/#SD>]. **Atenție!** Temele trebuie trimise în secțiunea **Structuri de Date (CA)**.

Arhiva trebuie să conțină:

- sursele voastre
- fișier Makefile care să conțină obligatoriu urmatoarele 2 reguli:
 - regula **build**: în urma căreia se generează un executabil numit **tema2**
 - regula **clean** care șterge executabilul (și eventualele fișiere obiect)
- fișier **README** care să conțină detalii despre implementarea temei

Atenție! Tema voastră va trebui să citească toate datele de intrare de la **stdin** și va trebui să scrie toate datele de ieșire la **stdout**

Punctaj

- 80 puncte obținute pe testele de pe vmchecker. Condiții pentru obținerea punctajului total:
 - fără memleak-uri
 - fără erori de valgrind
- 20 puncte: README
- **Bonus 20% din punctajul obținut** pentru coding style
 - spre exemplu: cu 60p din 100p și coding style perfect, obțineți $60 \cdot 1.2 = 72p$

Nu copiați! Toate soluțiile vor fi verificate folosind o unealtă de detectare a plagiatului. În cazul detectării unui astfel de caz, atât plagiatorul cât și autorul original (nu contează cine care e) vor primi punctaj 0 pe **toate temele!**

De aceea, vă sfătuim să nu vă lăsați rezolvări ale temelor pe calculatoare partajate (la laborator etc), pe mail/liste de discuții/grupuri etc.