

Instructiuni

Fiecare student isi alege una dintre teme si se trece in Excel-ul de pe Teams "Teme Jocuri - Assignment". O tema poate fi aleasa de maxim 2 studenti.

Temele prea similare vor fi depunctate (atat pentru cel care a dat tema cat si pentru cel care a copiat).

Temele se vor si prezenta pe scurt pentru a fi punctate: cum ati gandit estimarile, exemplu rulare etc.

Tema valoreaza 4p/10p asociate laboratorului KR. (barem si mai detaliat mai jos)

Deadline: 04.06.2022, ora 23:59

Tema se va trimite pe email cu un atasament ca arhiva cu toate documentele incluse pe ana-sabina.uban@g.unibuc.ro, email cu titlul "[IA24] Mini-proiect jocuri <Nume> <Grupa>".

Barem (punctajul e dat in procentaje din punctajul maxim al temei. Ce e dupa punctul 10 nu sunt obligatoriu de implementat, puteti aduna punctaj cum preferati):

1. (5%) Să se păstreze următoare lucruri deja implementate în exemplu (sau să se implementeze daca cineva decide să refacă programul de la zero):
 - La inceputul programului utilizatorul va fi intrebat ce algoritm doreste sa foloseasca (minimax sau alpha-beta)
 - Utilizatorul va fi întrebat cu ce simbol sa joace (la jocurile unde are sens aceasta intrebare)
 - Se va încerca evitarea sau tratarea situației în care utilizatorul ar putea răspunde greșit (de exemplu, nu poate selecta decât opțiunile corecte dintre care sunt selectate valorile default; sau, unde nu se poate așa ceva, jocul nu pornește până nu se primește un răspuns corect).
 - Afisarea a cui este rândul să mute.
 - Indicarea, la finalul jocului, a câștigătorului sau a remizei daca este cazul.
2. (5%) Utilizatorul va fi întrebat care sa fie nivelul de dificultate a jocului (incepator, mediu, avansat). In functie de nivelul ales se va seta adancimea arborelui de mutari (cu cat nivelul ales e mai mare, cu atat adancimea trebuie sa fie mai mare ca sa fie mai precisa predictia jocului). Posibilitatea utilizatorului de a face eventuale alte setări cerute de enunț. Se va verifica dacă utilizatorul a oferit un input corect, iar dacă nu se va trata acest caz (i se poate reafișa ecranul cu setările afișând și un mesaj de atenționare cu privire la inputul greșit).
3. (5%) Generarea starii initiale
4. (15%) Functia de generare a mutarilor (succesorilor) + eventuala functie de testare a validitatii unei mutari (care poate fi folosita si pentru a verifica mutarea utilizatorului)

5. (5%) Realizarea mutarii utilizatorului. Utilizatorul va realiza un eveniment în interfață pentru a muta. Va trebui verificata corectitudinea mutarilor utilizatorului: nu a facut o mutare invalida.
6. (10%) Functia de testare a starii finale, stabilirea castigatorului și, dacă e cazul conform cerinței, calcularea scorului. Se va marca configurația câștigătoare (sau simbolurile câștigătoare, în funcție de regulile jocului). Marcarea se poate face colorând, de exemplu, simbolurile sau culoare de fundal a eventualelor căsuțe în care se află, afișând cu bold pentru interfețele text, etc
7. (20%=10+10) Doua moduri diferite de estimare a scorului (pentru stari care nu sunt inca finale)
8. (15% împartit după cum urmează) Afișari (în consolă).
 - (5%) Afișarea timpului de gandire, dupa fiecare mutare, atat pentru calculator (deja implementat în exemplu) cat si pentru utilizator. Pentru timpul de gândire al calculatorului: afișarea la final a timpului minim, maxim, mediu și a medianei.
 - (2%) Afișarea scorurilor (dacă jocul e cu scor), atat pentru jucator cat si pentru calculator și a estimărilor date de minimax și alpha-beta (estimarea pentru rădacina arborelui; deci cât de favorabilă e configurația pentru calculator, în urma mutării sale - nu se va afișa estimarea și când mută utilizatorul).
 - (5%) Afișarea numărului de noduri generate (în arborele minimax, respectiv alpha-beta) la fiecare mutare. La final se va afișa numărul minim, maxim, mediu și mediana pentru numarul de noduri generat pentru fiecare mutare.
 - (3%) Afișarea timpului final de joc (cat a rulat programul) si a numarului total de mutari atat pentru jucator cat si pentru calculator (la unele jocuri se mai poate sari peste un rand și atunci să difere numărul de mutări).
9. (5%) La fiecare mutare utilizatorul sa poata si sa opreasca jocul daca vrea, caz in care se vor afisa toate informațiile cerute pentru finalul jocului (scorul lui si al calculatorului, numărul minim, maxim, mediu și mediana pentru numarul de noduri generat pentru fiecare mutare, timpul final de joc și a numarului total de mutari atat pentru jucator cat si pentru calculator). Punctajul pentru calcularea efectivă a acestor date e cel de mai sus; aici se punctează strict afișarea lor în cazul cerut.
10. (5%) Comentarii. Explicarea algoritmului de generare a mutarilor, explicarea estimarii scorului si dovedirea faptului ca ordoneaza starile cu adevarat in functie de cat de prielnice ii sunt lui MAX (nu trebuie demonstratie matematica, doar explicat clar). Explicarea pe scurt a fiecărei functii si a parametrilor.
11. (10%) Desenarea tablei de joc (interfața grafică) si afișarea în consolă a tablei (pentru debug; în ce format vreți voi). Titlul ferestrei de joc va fi numele vostru + numele jocului.
12. (5%) Dacă faceți mai multe estimări ale scorului dar diferite ca idee (cu alt mod de abordare a jocului).
13. (5%). Ordonarea succesorilor înainte de expandare (bazat pe estimare) astfel încât alpha-beta să taie cât mai mult din arbore.
14. (10%). Opțiuni în meniu (cu butoane adăugate) cu:
 - Jucator vs jucător
 - Jucător vs calculator (selectată default)
 - Calculator (cu prima funcție de estimare) vs calculator (cu a doua funcție de estimare)
15. (5%) Cand vine randul jucatorului sa mute, sa se marcheze pe interfata pozitiile valide in care poate plasa un simbol.
16. (15%) Pentru adancimi mai mari de 3 sa se salveze arborele alphabeta si sa se plece de la ce este calculat deja pentru aflarea urmatoarei mutari a calculatorului (ca sa nu fie recalculat mutarile la urmatoarea iteratie). Practic memoram subarborele mutarii alese din

arborele alpha-beta generat anterior. Apoi jucatorul face mutarea M. Cautam in subarborele salvat mutarea jucatorului si pornim de la subarborele generat deja pentru ea si il continuam pentru a afla noua mutare a calculatorului.

17. (5%). Implementati optiunea de undo. La apasarea tastei u, se da undo ultimei mutari facute a utilizatorului. De la al doilea "undo" incolo se anuleaza mutarea calculatorului + jucatorului (practic se anuleaza mutarea jucatorului si raspunsul calculatorului).
18. (5%). Implementati optiunea de restartare joc. La apasarea tastei r, se genereaza din nou tabla initiala, se reseteaza scorurile si alte valori ce tineau de jocul anterior.
19. (5%) La apasarea tastei n la inceputul turn-ului jucatorului (cand jucatorul inca nu a facut vreo mutare) calculatorul isi va schimba (inlocui) mutarea cu urmatoarea cea mai buna dintre mutarile posibile. Apasari succesive ale lui n, ar afisa pe tabla pe rand de la cea mai buna mutare la cea mai slaba. Daca se apasa n si dupa cea mai slaba se reafiseaza cea mai buna (lista e luata de la inceput).
20. (10%) Optiunile de inrerupere, salvare si continuare joc. La apasarea tastei s, starea jocului se va salva intr-un fisier text (numele fisierului va fi cerut utilizatorului - acesta poate introduce numele fie in interfata grafica fie in consola). Fisierul va fi salvat intr-un folder numit "salvari" al jocului. La intrarea in joc, utilizatorul va primi ca prima intrebare din partea programului daca vrea sa incarce un joc si i se va afisa continutul fisierului de salvari, fiecare fisier avand un numar de ordine. Utilizatorul va raspunde cu numarul de ordine si va putea continua jocul din stadiul in care l-a lasat (tabla si alti parametri se vor incarca din fisier).
21. (1-10%) Diverse implementari deosebite, optimizari speciale.