

Caiet de sarcini

-POO-

Joc X/O împotriva calculatorului

Nume: Cioban

Prenume: Daniel

Specializarea: Calculatoare

Grupa : 3121a

Cuprins

1. Tema și motivația alegerii	3
1.1. Descrierea problemei	3
1.2. Abordarea teoretică a problemei	3
2. Implementare	4
2.1. Tehnologii folosite	4
2.2. Diagrama claselor	4
2.3. Descrierea claselor	4
2.4. Aspectul aplicației.....	7
2.5. Posibile implementări în plus.....	8
3. Modul de utilizare al aplicației.....	9

1. Tema și motivația alegerii

Tema jocului constă într-un joc de tip X/O în care jucătorul sau utilizatorul va concura împotriva calculatorului (un adversar virtual implementat printr-un algoritm).

Motivul alegerii acestei teme constă în dorința mea de a-mi dezvolta abilitățile de programare prin construirea unui joc simplu bazat pe tehnica de programare orientată pe obiecte, dar și prin implementarea unui algoritm ce permite funcționarea unui adversar virtual (implementarea unei inteligențe virtuale de nivel foarte scăzut).

1.1. Descrierea problemei

X și O (română) sau tic-tac-toe (engleză americană) este un joc de hârtie și creion pentru doi jucători care marchează pe rând spațiile într-o grilă de 3 pe 3 cu X sau O . Jucătorul care reușește să plaseze trei dintre semnele lor într-un rând orizontal, vertical sau diagonal este câștigător. Este un joc rezolvat , cu o remiză forțată presupunând cel mai bun joc de la ambii jucători.

Un joc rezolvat este un joc al cărui rezultat (câștigă, pierdere sau egalitate) poate fi prezis corect din orice poziție, presupunând că ambii jucători joacă perfect. Acest concept este de obicei aplicat la jocurile abstracte de strategie și în special la jocurile cu informații complete și fără element de șansă; rezolvarea unui astfel de joc poate folosi teoria jocurilor combinatorii și/sau asistență computerizată.

1.2. Abordarea teoretică a problemei

În acest proiect va fi realizat un program consolă în C++ care efectuează principalele funcții ale unui joc X/O în consolă: începere joc nou, introducere poziție din tabla de joc pentru a o completa cu simboluri, reîncepere joc nou, ieșire din program.

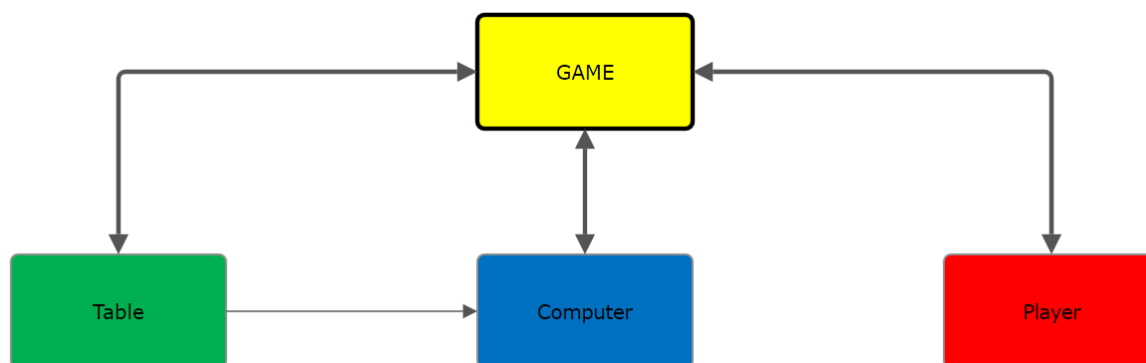
2. Implementare

2.1. Tehnologii folosite

Pentru acest proiect va fi folosit un mediu de lucru care suportă limbajul de programare C++ cum ar fi Visual Studio Code, Visual Studio Community 2022 sau Code::Blocks.

2.2. Diagrama claselor

În cadrul aplicației vor fi implementate următoarele 4 clase de bază: Game, Table, Player, Computer. Există posibilitatea ca aceste clase să aibă denumiri diferite. Clasele vor fi construite în următorul fel una față de cealaltă:



Clasa Game va fi clasa principală a aplicației. Ea va funcționa în majoritatea cazurilor ca legătura dintre celelalte clase. Singura situație când alte două clase vor comunica direct va fi atunci când clasa Computer va primi în mod direct date de la clasa Table.

2.3. Descrierea claselor

Pentru această aplicație următoarele clase de bază vor fi implementate:

❖ Game

Această clasă este clasa principală și trebuie să definească regulile jocului și să gestioneze starea globală a jocului. Ea trebuie să ofere următoarele funcționalități de bază:

- Afisează meniul principal și permite alegerea unei opțiuni de către utilizator. Opțiunile ar putea fi:
 - Începe joc nou,
 - Încheie aplicație.
- Verifică dacă datele introduse de utilizator. Sunt posibile următoarele timpuri de date, fiecare cu verificările lor:
 - a) Coordonatele primite din clasa Player și Computer. Ele sunt verificate astfel:
 - Prin verificarea dacă coordonatele introduse nu sunt eronate (trebuie să fie doar numere întregi pentru a fi corecte) și se încadrează în matrice (valoarea liniei și a coloanei trebuie să fie mai mari sau egale cu 1 și mai mici sau egale cu 3. Orice altă valoare ce nu se încadrează în acest interval este considerată greșită).
 - Prin trimiterea coordonatelor în clasa Table pentru a afla dacă este o mișcare validă sau nu.

Dacă coordonatele sunt greșite va fi anunțat acest lucru pe ecran în cazul datelor primite din clasa Player și va fi apelată din nou funcția de citire a clasei Player sau Computer, după caz, pentru a reintroduce coordonate. Acest lucru se va repeta până clasa apelată va returna date corecte.

- b) Opțiunea introdusă de utilizator. Aceasta trebuie să fie un număr întreg și să corespundă unei opțiuni afișate în acel moment în consolă. De exemplu pentru a alege opțiunea “1. Începe joc” utilizatorul va trebui să introducă “1” de la tastatură. Orice altă valoare sau caracter se consideră greșită.

Dacă opțiunea este greșită va fi atenționat utilizatorul și va fi nevoit să introducă din nou opțiunea până va fi o opțiune corectă.

- Începe un joc nou, oferind jucătorului opțiunea de a începe primul sau ultimul și transmite această informație clasei Table. Potrivit opțiunii alese va fi determinat semnul fiecărui jucător:
 - Primul jucător: X,
 - Al doilea jucător: O.
- Afisează tabla de joc în consolă și informații despre joc (jucătorul și simbolul lui), starea jocului în caz de egalitate sau câștigare, și afișează opțiunile potrivite fiecărei situații;
- Verifică starea jocului (continuare joc, câștig, egalitate) după fiecare mișcare validă făcută de jucători apelând funcția de verificare a clasei Table și oferă opțiunile posibile pentru fiecare situație:
 - a) Pentru continuarea jocului:
 - Citire poziție nouă
 - b) Pentru câștig și egalitate:
 - Începe joc nou,
 - Întoarcere la meniul principal,

- Încheie aplicație

- Oprește aplicația dacă este aleasă opțiunea potrivită acesteia.

❖ Table

Această clasă reprezintă tabloul de joc, care este o matrice de 3 linii și 3 coloane. Ea ar trebui să ofere următoarele funcționalități:

- Construiește la începerea unui joc o matrice principală cu 3 linii și 3 coloane populată cu 0 pe fiecare poziție. Această matrice va fi populată cu mișcările fiecărui jucător;
- Primește informația despre simbolul fiecărui jucător și o salvează într-un vector. Indicele vectorului va corespunde cu indicele jucătorului și valoarea salvată la acel indice va fi simbolul jucătorului. De fiecare dată când un jucător va face o mișcare validă vectorul va fi folosit pentru a plasa simbolul corespunzător jucătorului la poziția specificată de el;
- Primește de la clasa Game poziția introdusă de jucător sau calculator și verifică dacă este validă (nu este deja ocupată în matrice. Pentru ca o poziție să fie validă atunci în matrice trebuie să fie valoarea 0 la acea poziție) și returnează dacă este validă sau nu clasei Game;
- Salvează în matricea construită poziția (linia și coloana) validă primită de clasa Game în felul următor:
 - Cu 1 primul jucător,
 - Cu 2 al doilea jucător.
- Verifică dacă este egalitate sau dacă un jucător a câștigat și returnează această informație clasei Game cu o valoare corespunzătoare fiecărei situații. Dacă jocul încă continuă atunci va returna o valoare corespunzătoare acestei situații.
- Crează câte o matrice cu mișcările fiecărui jucător și pentru pozițiile valide din matricea principală și le poate returna altei metode sau clase.

❖ Player

Această clasă reprezintă jucătorul. Ea trebuie să ofere următoarele funcționalități:

- Permite introducerea poziției (linia și coloana) dorite de jucător și returnează coordonatele introduse clasei Game. Dacă s-a verificat că datele returnate au fost greșite atunci funcția de citire va fi apelată din nou.

❖ Computer

Această clasă reprezintă clasa calculatorului și trebuie să ofere următoarele funcționalități:

- La apelarea generatorului de poziție de către clasa Game se va cere de la clasa Table matricele create și se vor folosi pentru a genera o mișcare validă care va fi returnată clasei Game.

2.4. Aspectul aplicației

Aplicația de bază va fi în consolă. Aspectul aplicației nu este stabilit. Însă sunt propuse următoarele aspecte:

```
Salut!  
  
Alege o optiune:  
-----  
1. Incepe un joc nou.  
2. Inchide aplicatia.  
-----  
Optiunea ta:
```

Fig. 2. Exemplu de meniu principal

```
Tu: X  
Inamicul: 0  
-----  
   1   2   3  
1  X  | -  | 0  
-----  
2  -  | -  | 0  
-----  
3  -  | -  | -  
-----  
Inamicul a ales pozitia [ 2 , 3 ]  
Introdu miscarea ta:
```

Fig. 3. Exemplu de tablă în timpul jocului cu opțiunea de a introduce poziția dorită

```
Tu: X
Inamicul: O

-----
  1   2   3
1  X  |  -  |  0
-----
2  X  |  -  |  0
-----
3  -  |  -  |  0
-----

Inamicul a ales pozitia [ 2 , 3 ]
Ai pierdut jocul!
Ce doresti sa faci acum?
-----
1. Incepe un joc nou;
2. Intoarcete in meniul principal;
3. Inchide aplicatia.
-----

Optiunea ta: 1
```

Fig. 4. Exemplu de joc pierdut cu mai multe opțiuni pentru utilizator

Este posibil ca aplicația finală să nu aibă nici o asemănare cu exemplele propuse sau o asemănare foarte mică. Însă aplicația va avea majoritatea funcționalităților prezentate în exemplele incluse (cu mici variații).

2.5. Posibile implementări în plus

Pe lângă funcționalitățile și implementările menționate sunt posibile următoarele implementări (sunt posibile să fie incluse în proiect după ce au fost implementate toate lucrurile propuse în proiect însă nu sunt obligatorii. Sunt lucruri extra, în plus față de ce a fost propus pentru proiect):

- Adăugare elemente de design (culori, tematici, desene și/sau animații ASCII, etc.) și sunete (de exemplu sunet pentru câștigarea/pierderea jocului, pentru începerea unui joc, etc.) cu setări pentru a oferi jucătorului posibilitatea de a customiza/modifica jocul după preferințe;
- Adăugare cronometru de la începerea unei noi remize până la încheierea ei;
- Adăugare istoric al remizelor cu diferite informații despre joc salvate local într-un fișier (câștigătorul, durata, numărul de mișcări etc.);
- Alte modalități de a interacționa cu jocul pe lângă introducerea de la tastatură (prin folosirea mouse-ului sau alte modalități);
- Adăugare a diferite moduri de joc. Pe lângă jocul împotriva calculatorului să existe modul de joc împotriva altui jucător pe același dispozitiv/computer, modul de a juca pe o tabla mai mare (de exemplu 9x9) sau/și moduri de dificultate diferite.

3. Modul de utilizare al aplicației

Utilizatorul va porni aplicația și va fi întâmpinat cu un mesaj de bun venit și cu opțiunea de a începe un joc nou sau de a ieși din aplicație. După începerea unui joc nou va afișată tabla de joc în consolă. Utilizatorul și calculatorul vor face mișcările lor, alternând între ei, fiind ghidați să introducă mișcările lor valide în consolă. Aceste mișcări pot fi specificate prin introducerea coordonatelor (rând și coloană) sau prin alte metode intuitive. La fiecare mișcare făcută de un jucător, aplicația va verifica starea jocului pentru a determina dacă există un câștigător, dacă jocul a ajuns la o remiză sau dacă acesta va continua. După ce jocul se termină (cineva câștigă sau este remiză), aplicația va afișa un mesaj corespunzător și va oferi opțiunea de a începe un joc nou sau de a ieși din aplicație.