

Taradaciuc\_Nicolae, Vicol\_Șerban-Ilie, Butacu\_Ionel-Cătălin

---

## <Dezvoltarea unui Sistem Expert bazat pe cazuri pentru jocul Battleship>

### Rezumat

În acest proiect s-a dorit realizarea unui sistem expert bazat pe cazuri pentru jocul Battleship. Soluția cu care echipa noastră a venit este un sistem ce este capabil să își schimbe comportamentul în funcție de dificultatea aleasă dar și de situația în care se află jocul. Dacă trec prea multe runde în care acesta nu poate găsi o navă pe tabla de joc folosindu-se de atacul cu bombă atunci va trece la utilizarea atacurilor de linie sau a atacului de scanare în dorința de a grăbi procesul de localizare al unei nave. Pentru a face posibil aceste funcționalități s-a utilizat atât limbajul de programare bazat pe reguli pentru a descrie terenurile jucătorului și a sistemul expert, tipurile de atac și mecanismele prin care se actualizează cele două terenuri, cât și limbajul de programare python pentru realizarea unei interfețe grafice intuitive pentru utilizator.

### 1. Enunțul problemei. Obiective

Problema propusă spre rezolvare este realizarea unui sistem expert bazat pe cazuri capabil să ia decizii privind o nouă mutare cu scopul de a doborî toate navele inamicului (utilizatorul uman) și a câștiga meciul.

Scopul principal al sistemului este să doboare navele oponentului prin utilizarea unei singure abilități pe runda. În cazul unei lovituri cu succes, sistemul are posibilitatea de a lua o nouă decizie și a ataca din nou, bazându-se pe cunoștințele sale din mediul prezent.

Elementele de interes sunt navele ce pot avea două stări (atacată sau nu), bombă și abilități (scanare, etc). Bombele pot fi plasate pe o singură porțiune din harta, o singură dată pe runda.

În final, utilizatorul uman va fi capabil să joace împotriva sistemului, acesta din urmă schimbându-și comportamentul și strategiile de atac în funcție de numărul de nave distruse de utilizator.

Exemplu de ipoteza:

- Se poate lovi până în momentul în care lovitura este fără succes;
- Ambii jucători au posibilitatea de a ataca(cu o bombă) cel puțin o dată în fiecare rundă;
- Se poate folosi o singură abilitate(scanare) de către fiecare jucător, per rundă;
- Dacă ai atacat o locație, nu o vei putea ataca din nou.

## 2. Descrierea universul de discurs

La începutul jocului, mediul de lucru va conține doar “nave” și “poziție\_neatacată”. Pe parcursul jocului, utilizatorii “player” și “sistem\_expert” se vor ataca reciproc utilizând “bombă” sau “abilitate”. Un atac cu succes asupra unei nave va rezulta în schimbarea poziției unei “nave” în “poziție\_nava\_distrusă”, iar un atac eșuat asupra unei “poziții\_neatacate” o va transforma într-o “poziție\_atacată”. După ce fiecare porțiune dintr-o “navă” este în întregime transformată în “poziție\_nava\_atacata”, aceasta va deveni “nava\_distrusa”, iar porțiunile vor trece în starea „poziție\_navă\_distrusă”.

### a. Concepte

Conceptele utilizate sunt: poziție\_liberă, poziție\_atacată, navă, poziție\_navă\_atacată, navă\_distrusă, bombă, abilitate, player, sistem\_expert

### b. Proprietăți

Teren T1 poziția (rând, coloana) este liberă.

Teren T1 poziția (rând, coloana) este atacată.

Teren T1 poziția (rând, coloana) este ocupată de nava N2 și este atacată.

Teren T1 poziția (rând, coloana) este ocupată de nava N3 și este distrusă.

Sistemul decide.

Sistemul așteaptă.

### c. Relații

Teren T1 nava N2 se află pe pozițiile ( x1, y1), (x2, y1), (x3, y1).

Teren T1 poziția (rând, coloana) este atacată cu B.

### d. Acțiuni

Acțiunea de a ataca cu B poziția (x, y) din Terenul T1. Aceasta este aplicabilă doar dacă poziția nu a fost atacată anterior.

Acțiunea de a scana cu S poziția (x, y) din Terenul T2. Aceasta este aplicabilă oricând, dar abilitatea nu mai poate fi refolosită timp de 5 runde.

## 3. Scenarii considerate

Instanța 1: Distrugerea unei poziții de pe tabla de joc

Sistemul expert primește comanda de a ataca o poziție de pe tabla de joc cu un atac de tip bombă.

Instanța 2 : Distrugerea unei linii de pe tabla de joc

Sistemul expert primește comanda de a ataca o poziție de pe tabla de joc cu un atac de tip linie.

## 4. Strategii de rezolvare

Pentru instanța 1: Distrugerea unei poziții de pe tabla de joc

Pentru rezolvarea acestui scenariu, sistemul expert are nevoie să ia în considerare două aspecte:

- Dacă poziția este deja distrusă

- Dacă poziția este ocupată de o porțiune de navă sau nu

În funcție de tipul poziției, sistemul expert o va considera poziție liberă atacată sau poziție cu porțiune de navă atacată.

Pentru instanța 2: Distrugerea unei linii de pe tabla de joc

Pentru rezolvarea acestui scenariu, sistemul expert se va folosi de poziția pe care o primește pentru a-și da seama care linie de pe tabla de joc trebuie distrusă. Acesta va planifica o serie de atacuri de tip bombă asupra tuturor pozițiilor de pe acea linie după care va intra scenariul de mai sus.

## 5. Definirea bazei de fapte

a. Tipare de fapte

(Teren <ID\_Teren> poziția <rând> <coloana> este <stare>)

(Teren <ID\_Teren> poziția <rând> <coloana> este ocupată de nava <ID\_Navă> și este <stare\_poziție\_navă>)

(Nava <ID\_Navă> din poziția <rând> <coloana> este atacată cu <tip\_atac>)

(Sistem ataca poziția <rând> <coloana> din terenul <ID\_Teren> cu <tip\_atac>)

(Jucător ataca poziția <rând> <coloana> din terenul <ID\_Teren> cu <tip\_atac>)

(Nava orizontala <ID\_Navă> rând <ID\_rând> pe coloanele <<< indici\_coloane>>>)

(Nava verticala <ID\_Navă> coloana <ID\_coloana> pe rândurile <<< indici\_rânduri>>>)

(Nava <ID\_Navă> în terenul <ID\_Teren>)

(dificultate <Nivel\_dificultate>)

(update\_map <Stare>)

(calcul\_frontiera < rând > <coloana>)

(frontiera <rând\_colt\_stanga\_sus> <coloana\_colt\_stanga\_sus> <rând\_colt\_dreapta\_jos> <coloana\_colt\_dreapta\_jos>)

b. Descriere prin fapte nestructurate

Faptele nestructurate din instanța 1:

- Cazul pentru atacul unei poziții libere  
(Teren T1 poziția 6 7 este liber)  
(Sistem ataca poziția 6 7 din terenul T1 cu B)
- Cazul pentru atacul unei porțiuni de navă  
(Teren T1 poziția 6 7 este ocupata de nava N1 și este neatacata)  
(Sistem ataca poziția 6 7 din terenul T1 cu B)

## 6. Definirea bazei de reguli

În situația în care se alege atacarea cu B (bombă) a unei poziții din Terenul adversarului, iar poziția nu este atacată anterior atunci această acțiune este realizabilă. Rezolvarea acestei situații este concretizată în următoarea regulă denumită Actualizare\_Teren\_atacat\_B:

Dacă sistemul sau jucătorul atacă poziția dată prin ?rand și ?coloana din terenul ?Teren și poziția este liberă, atunci aceasta devine atacată.

În situația în care regula Actualizare\_Teren\_atacata\_B a dus la atacarea unei nave vom prevedea o regulă pentru a-i actualiza starea. Rezolvarea acestei situații este concretizată în următoarea regulă denumită Actualizare\_Nava\_atacata\_B:

Dacă sistemul sau jucătorul atacă poziția dată prin ?rand și ?coloana din terenul ?Teren, poziția este ocupată de nava ?nava și este neatacată, atunci starea navei devine atacată.

În situația în care se alege atacarea unei linii din Terenul adversarului, fiecare poziție de pe acea linie va fi atacată cu o bombă, cu condiția ca poziția să nu fi fost atacată anterior. Rezolvarea acestei situații este concretizată în următoarele reguli denumite Atac\_linie\_jucator și Atac\_linie\_sistem:

Dacă sistemul sau jucătorul atacă linia dată prin ?rand din terenul ?Teren, pozițiile libere neatacate/libere for deveni atacate.

În situația în care se alege scanarea unei porțiuni din Terenul adversarului, fiecare poziție din acea porțiune va fi scanată pentru a se detecta prezența a cel puțin unei nave fără a se oferi exact poziția acesteia. Rezolvarea acestei situații este concretizată în următoarele reguli denumite Atac\_scanare\_jucator și Atac\_scanare\_sistem:

Dacă sistemul sau jucătorul selectează o poziție pentru scanare, porțiunea de scanat va fi o matrice de 3x3 cu centrul în poziția selectată (?rand, ?coloana) din terenul ?Teren, atunci se va trimite mesajul „Există o navă în zona scanată”.

## **7. Rezolvarea conflictelor**

### **7.1 Cazuri conflictuale**

Conflict 1:

Un conflict ce a fost identificat în cadrul aplicației este închiderea, deschiderea, scrierea si citirea din fișier într-o ordine incorectă, ceea ce a dus la diferite erori precum “scrierea de informații într-un fișier deja închis” și “citirea dintr-un fișier ce încă nu a fost deschis”.

Conflict 2:

Un alt conflict identificat a fost activarea reguli Actualizare\_Teren\_atacat\_B în timpul rulării regulilor Atac\_linie\_sistem / Atac\_linie\_jucator ceea ce a dus la diferite erori precum actualizarea unei poziții înainte ca atacul liniei să fie finalizat.

### **7.2 Strategii de rezolvare conflicte**

Rezolvare conflict 1:

Strategia utilizată pentru a rezolva acest conflict a fost utilizarea priorităților, oferind regulilor următoarele priorități: Rule\_Opening\_File\_Read (100), Rule\_Reading\_Map (99), Rule\_Closing\_File\_Read (98), Rule\_Opening\_File\_Write (97), Rule\_Writing\_In\_Map (96), Update\_Map\_Command (96) și Rule\_Closing\_File\_Write (95).

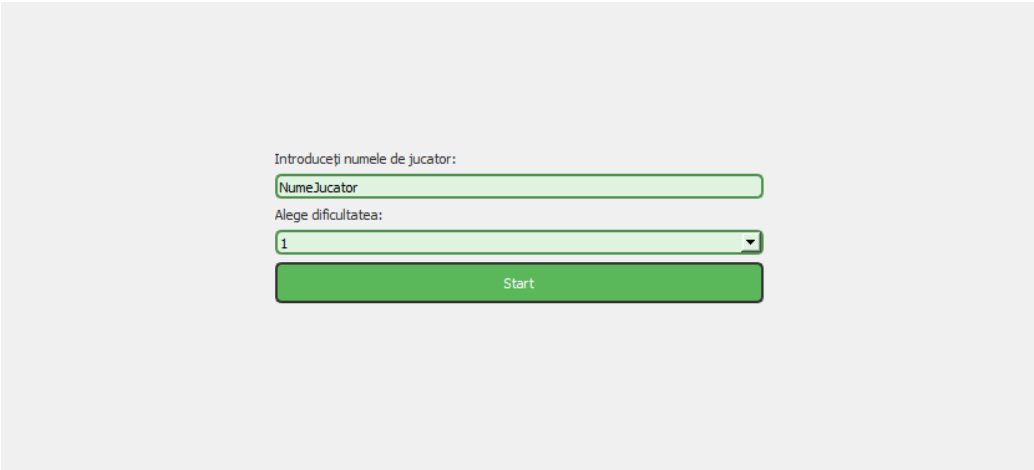
Strategia utilizată pentru a rezolva acest conflict a fost utilizarea priorităților, mărind prioritarea atacului de linie pentru a fi mai mare decât regula de actualizare a poziției: Actualizare\_Teren\_atacat\_B (1), Atac\_linie\_sistem (10), Atac\_linie\_jucator (10).

Limbajul utilizat: Python  
Biblioteci utilizate: clipspy [2], NumPy [3], PyQt5 [4]  
Funcții utilizate:

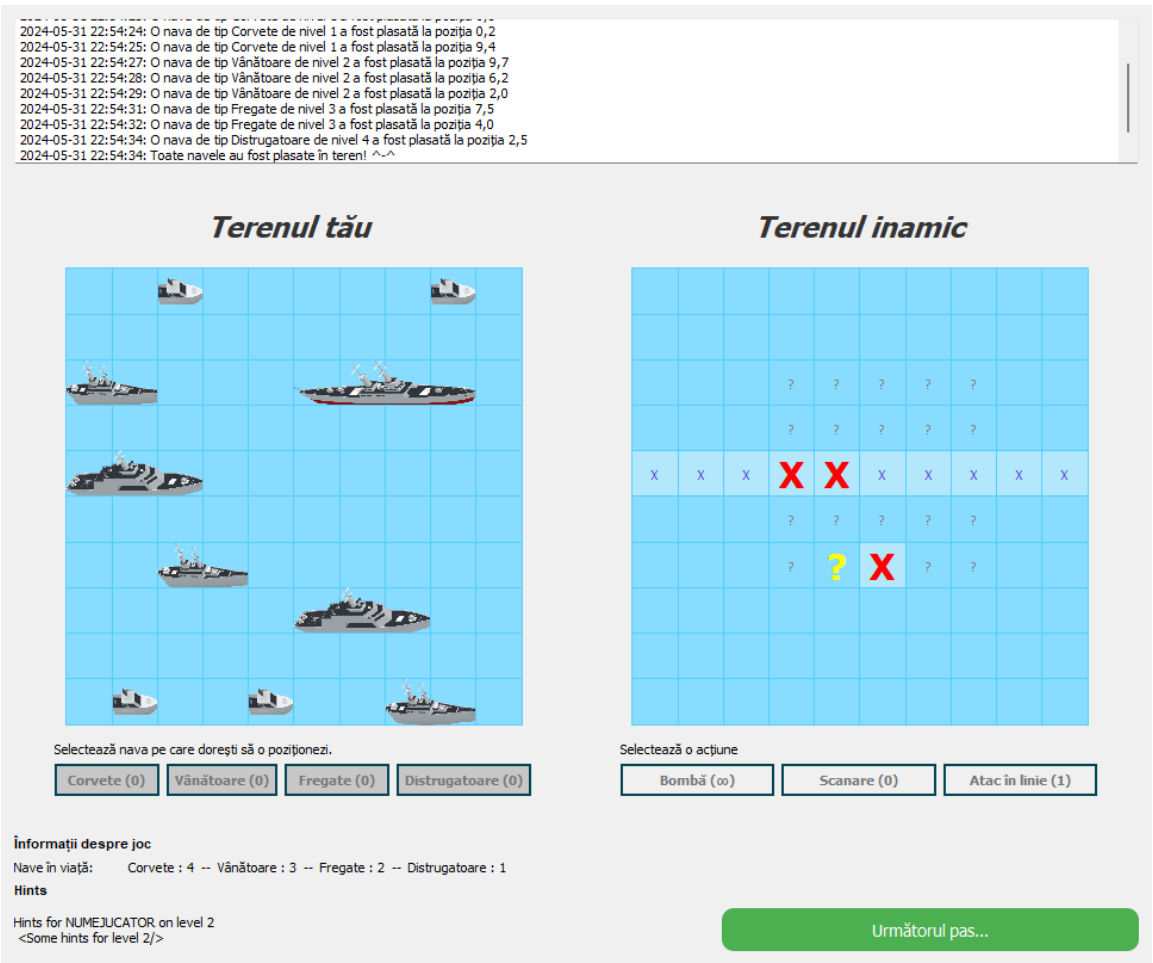
- clipspy [2] : clear, load, reset, eval, run, facts, activations
- NumPy [3] : zeros, random
- PyQt5 [4] : connect, emit, blockSignals, move, show, exec, exit, parentWidget, addWidget, addLayout, setCursor, moveCursor, updateGeometry, setObjectName, setStyleSheet, setProperty, setAlignment, setCheckable, setEnabled, setText, setFont, setCentralWidget, setWindowTitle, setFixedSize, setReadOnly, currentDateTime

Descriere interfață GUI:

Înainte de a începe jocul, suntem întâmpinați de o interfață simplă unde trebuie să ne introducem numele și să selectăm nivelul de dificultate. Alegerea nivelului de dificultate influențează agresivitatea atacurilor sistemului, realizând astfel o experiență de joc personalizată.



Interfața va prezenta un teritoriu de luptă împărțit în două terenuri distincte, unul pentru navele jucătorului (stânga) și celălalt pentru navele sistemului (dreapta). În primul stadiu al jocului, atât sistemul cât și jucătorul își vor amplasa navele aleator în propriile terenuri. Navele pe care jucătorul și le va amplasa se află sub terenul acestuia.



Pe măsură ce jocul avansează, jucătorul va putea lansa o serie de atacuri împotriva inamicului, atacuri precum bombe, scanare, atac în linie, acestea fiind vizibile sub terenul inamicului. Pe parcursul jocului, în ambele terenuri, fiecare stare („Poziție liberă/neatacată”, „Poziție atacată”, „Nava atacată”, „Navă găsită dar neatacată”, „Navă scanată și liberă”) va fi reprezentată printr-o serie de simboluri precum:

- Pătrat albastru= Poziție liberă/neatacată
- Pătrat albastru deschis cu simbolul “X” gri = Poziție atacată ( fără navă )
- Pătrat albastru deschis cu simbolul “X” roșu = Nava atacată
- Pătrat albastru cu simbolul “?” gri = Poziție scanată și liberă
- Pătrat albastru cu simbolul “?”galben = Navă scanată, dar neatacată

## Concluzii

Elemente pozitive:

- interfața ce este simplă și intuitivă, permițând jucătorilor să înceapă jocul rapid, fără a parcurge alte etape complicate
- Personalizarea stilului de joc prin selectarea nivelului de dificultate.
- Feedback-ul interactiv ce se află în partea de sus a interfeței și ne oferă date despre ce se acționează în loc pe parcursul jocului.

Elemente negative:

- Nivelul de dificultate ar putea fi extins prin a afecta stilul de joc și prin alte modalități și nu doar atacurile sistemului.
- În momentul scanării, jucătorul știe exact poziția navelor de pe hartă, nu doar un detaliu precum existența sau absența lor.

Posibilități de îmbunătățire:

- Rezolvarea unei erori întâmpinate în integrarea clips-ului [5] cu interfața python.
- Îmbunătățirea algoritmului de căutare și atac al sistemului (Actualmente, incomplet).
- Adăugarea unui ghid explicativ cu o scurtă descriere a fiecărui nivel de dificultate și a diferențelor dintre acestea.
- Îmbunătățiri vizuale pentru a face interfața mai atrăgătoare și captivantă.
- Modificarea atacului de scanare pentru a oferi detalii precum existența sau absența de nave în zona respective, nu exact sterile tuturor pozițiilor scanate.
- Adăugarea unei bombe ca obiect de amplasat pe hartă, pentru a crea o experiență de joc mai detaliată.

## Bibliografie

- [1] Panescu D., Pascal C., Programare bazată pe reguli, Îndrumar de laborator, Editura Conspress, București, 2013, ISBN 978-973-100-258-3.
- [2] Documentație, librăria clipspy, <https://clipspy.readthedocs.io>
- [3] Documentație, librăria NumPy, <https://numpy.org/>

[4] Documentație, librăria PyQt5, <https://www.riverbankcomputing.com/static/Docs/PyQt5/>

[5] Manual Clips, CLIPS Reference Manual Volume I Basic Programming Guide, Version 6.30 March 17th 2015