Tic-Tac-Toe Game

Autor: Ologu Catinca-Ioana, 334AA

An: 2022

Cuprins

1. Introducere

Pentru acest proiect la disciplina Aplicatii Multimedia, am ales implementarea jocului „Tic - Tac -Toe” (tradus in romana *X* si *O*). Consider ca implementarea unui joc evidentiaza foarte bine conceptele prezentate atat in cadrul cursului cat si al laboratorului din acest semestru.

Ca si obiective principale am avut: implematrea unei interfete grafice placute utilizatorului, crearea logicii din spatele jocului prin intermediul unor algoritmi pe care ii voi detalia in subpunctele urmatoare ale lucrari, toate acestea imbinandu-se pentru a obtine in final o aplicatie functionala si matura. Ca si ultim pas propus a fost realizarea din codul sursa a unui executabil care sa poate beneficia de portabilitate, sa poate fi rulat pe orice sistem de operare Windows cu precadere.

In primul rand, as vrea sa detaliez mai mult subiectul ales. Cu siguranta fiecare dintre noi am jucat macar o data(in timpul unei ore/ curs, dupa caz) jocul „X si O”, este unul dintre cele mai populare jocuri ba chiar este unul dintre cele mai vechi jocuri. Dupa cum ii spune si numele este un joc care are in componenta sa doua persoane, doi jucatori/adversari/oponenti si o tabla de joc care este o matrice de trei linii si trei coloane in care jucatorii plaseaza alternativ simbolurile *X* si *O* intr-unul dintre cele nou spatii goale. O conventie poate fi faptul ca primul jucator, cel care marcheaza primul, va juca cu X.

Figură 1-Tabela de joc [1]

In Figură 1 se poate oberva setup-ul jocului, tabela de joc in forma de matrice de noua patratele dar si pionii jocului (X si O).

Regula prin care poti castiga acest joc este urmatoare: plasarea a trei dintre pionii de X sau O pe orizontala, verticala, diagonala dreapta sau diagonala stanga, jucatorul care reuseste acest lucru este declarat castigator. In figura de mai jos *(Figură 2*) este prezentata o asezare a pionilor X pe diagonala principala, o combinatie castigatoare.

Figură 2-Asezare castigatoare [1]

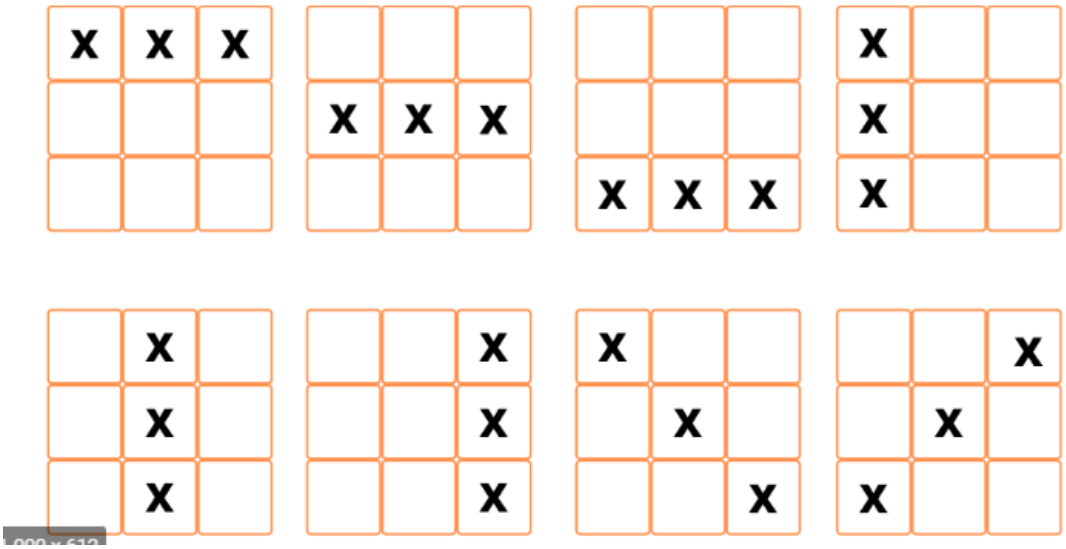
La o prima vedere am putea spune ca este un joc pueril cu regili simple, dar in profunzime este un joc al strategiilor, in special pentru cei mici acest joc le testeaza logica si gandirea critica.

Eu personal nu mi-am pus niciodata pana in acest moment intrebarea de unde a provenit acest joc si cu ce scop a fost inventat, in ciuda faptului ca il joc de atat de mul timp, dar as vrea acum sa detaliez esenta acestui joc teribil de banal.

Acest tip de joc, initial se spune ca a aparut in Egiptul Antic, unde astfel de tabele de joc (tabla de joc de 3x3) au fost gasite pe placile acoperisurilor iinca din anii 1300 i.Hr.

Una dintre primele varinte ale acestui joc a fost numita *terni lapilli (*trei pietricele de o data), ulterior in Marea Britanie pe la 1800 i s-a dat denumirea de *noughts and crosses,* nought fiind jucatorul *O* iar crosses jucatorul *X*. Denumirea finala de *Tic-Tac-Toe* a fost adoptata de catre americani in secolul XX, iar demunirea ar veni de la scrierile succesive de *X* si *O* ale jucatorilor. [1]

In continuare vor prezenta cateva strategii de abordare si respectic de castigare a jocului.



Figură 3-Combinatii castigatoare [2]

In *Figură 3* sunt prezentate toate combinatiile castigatoare pentru X (similare si pentru O).

Putem sa urmarim programul de joc al lui Newell and Simon, *„Newell and Simon's 1972 tic-tac-toe program”:* [3]

Acest program este alcatuit din opt reguli:

Regula unu***, Câștigă:*** jucatorul poate sa completeze cu un al treilea simbol daca are deja doua simboluri consecutive pentru a castiga

Regula doi***, Blocare***: oponentul trb sa blogheze contracandidatului sau plasarea al unui al treilea simbol pentru a castiga

Regula trei***, Fork***: ofera jucatorului posibilitatea de castig garantata, avand doua posibilitati simultane de a castiga

Regula patru***, Blocarea fork unui adversar:***

Daca exista doua mutari succesize, jucatorul trebuie sa blocheze, defapt el trebuie sa blogheze toate astfel de combinatii. În caz contrar, jucătorul ar trebui să facă un doi la rând pentru a-l forța pe adversar să se apere, atâta timp cât nu rezultă să producă o furcă. De exemplu, dacă „X” are două colțuri opuse și „O” are centrul, „O” nu trebuie să joace o mișcare de colț pentru a câștiga. (Jucarea unei mișcări de colț în acest scenariu produce o furcă pentru ca „X” să câștige.)

Regula cinci, ***Centru:*** Jucatorul poate sa aleaga miscarea de centru si anume marcarea casutei din centru, desi nu este cea mai recomandata miscare ca deschidere de joc.

Regula sase***, Colțul opus:*** coltul se contracareaza cu celalalt colt opus.

Regula sapte***, Colț gol:*** jucătorul joacă într-un careu de colț.

Regula opt ***Partea goală:*** Jucătorul joacă într-un pătrat din mijloc pe oricare dintre cele patru laturi.

Scopul nostru este de a ajunge la o astfel de combinatie iar ecest lucru poate fi facut prin umatoarele:

Primul jucator, X are aparent noua pozitii disponibile, dar de fapt sunt doar 3 deoarece celelate sunt permutarii ale primelor trei, in diferite unghiuri ale tablei de joc: *colt, margine, centru*. Jocul se poate termina oricand cu o remiza, dar coltul ca si prima mutare poate sa ofere cel mai mare avantaj in a castiga. Jucatorul O trebuie sa contracareze miscarile primului jucator astfel incat sa evite castigul acestuia. La o deschidere *colt* a jucatorului *X*, jucatorul O trebuie sa contracareze cu o marcare *centru* iar la o prima miscare *margine, jucatorul O* va contracara fie cu o miscare *centru, colt* langa X sau *o margine opusa* lui X.

Oricare alte mutari alea jucatorului *O* ii va permite jucatorului *X* sa castige jocul.

2. Prezentare pe scurt a suport tehnic

Pentru implemetarea algoritmului am folosit mai multe surse de inspiratie care constituie suportul meu tehnic.

Principalele mele surse de inspiratie a fost [4] , [5]. [6]. Initial am inceput sa ma documentez despre libraria *tkinter* din python [7]. Ulterior pentru a aprofunda subiectul am urmarit si inteles documentele precizate mai sus care mi-au servit ca suport de studiu.

In cele trei surse amintite, autorii folosesc deasemenea librarie tkinter pentru interfata grafice, fiind dezvoltat pe Windows toate cele trei proiecte.

3. Partea tehnica

Pentru implemetarea propriu zia a jocului am utilizat in acest scot limbal Python deoarece este o tehnologie care mi a permis sa implementez atata logica din spatele jocului, algoritmul cat si interfata grafica a aplicatiei.

Am folosit pentru a construi interfata grafica biblioteca ***tkinter*** din python care este o biblioteca standart si un tool specializat pentru crearea GUI-urilor.

Figură 4-Interfata grafica

In *Figură 4* se poate observa intervata grafica a jocului constuita in limbajul python cu ajutorul librariei *tkinter*.

Pentru implementarea acesteia am construit o baza, o radacina (root) in interiorul careia am adaugat diverse blocuri grafice, denumite *Frames* care reprezinta scheletul interfetei.

Frame-ul initial l-am numit root si este realizat folosit urmatoarea structura:

```
Root=Tk()
root.geometry("1350x750+0+0")
root.title("Tic-TacToe")
root.iconbitmap('XO.ico')
root.configure(background='bisque1')
```

Am setat o dimensiunea initiala, dupa care am adaugat titlul, am configurat fundalul si am setat si icon-ul in coltul stanga sus in antet.

In continuare am creat cele doua *Frame-uri* care se observa in Fig4, *Frame-ul* drept constituie tababla de joc propriu zisa, formata astfel din noua patratele asezate intr-o matrice de 3x3.

```
RightFrame = Frame(MainFrame, bd=10, width=560, height=500, pady=2, padx=10,
bg="mistyrose2",                                                    relief=GROOVE)
RightFrame.pack(side=RIGHT)
```

In codul de mai sus am exemplificat crearea unui frame, frame-ul drept, de exemplu a fost format in cadrul frame-ului principal, avand o serie de particularitati proprii, lungime, latine, background, pozitionarea, si cel mai interesant dintre toate este atributul *relief* care reda un efect de 3-D in jurul frame-ului. Eu am ales tipul GROOVE, dar exista si alte alte modele ce pot fi alese.

Fiecare patratel din tabela a fost realizat folosind urmatoarea structura:

```
btn1 = Button(RightFrame, text=" ", font='Times 26 bold', height=3, width=8, bg='mistyrose1',
image="",
            command=lambda: check_buttons(btn1, 1, 0, 1))
btn1.grid(row=1, column=0, sticky=S + N + E + W)
```

Pentru fiecare buton am setat frame-ul in care sa fie pozitionat, textul din interiorul acestuia, fontul, lungimea, latimea dar si fundalul asociat butonului respectiv.

Se poate observa in codul de mai sus aparitia unei functii *lamda*, a carei utilitite o voi comenta ulterior cand voi detalia algoritmul din spatele jocului.

Prin functia *grid* am pozitionat butonul in matrice (tabla de joc) cu ajutorul atributelor *row* si *column.*

Tot in Fig4 observam si *Frame-ul* stang care are in componenta sa doua sectiuni, in sectiunea superioara asunt prezente doua butoane: butonul *Reset* si butonul *New Game*, primul este folosit pentru a reseta tabela de joc iar cel de-al doilea pentru a reseta tabela de joc dar si scorul participantiilor la joc. Scorul jucatorilor este prezent in sectiunea inferioara a frame-ului stang.

In ceea ce priveste implemetarea codului am utilizat un algoritm facil de inteles si de aplicat de catre orcine. Codul este strucutrat in sase functii tocmai pentru a fi mult mai usor de inteles, cele sase functii sunt: ***check_buttons***, ***check_for_win_x, check_for_win_o, play, new_game_function, reset***.

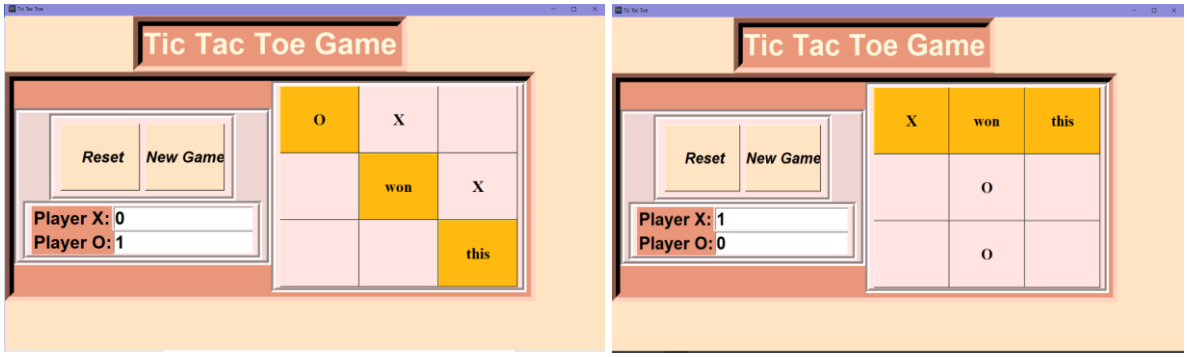
Functia ***check_buttons*** are ca parametru un buton si verifica daca acesta a fost apasat. Daca butonul este un buton gol(nu a fost introdusa nicio mutare acolo) si nu a a fost apasat, apasam butonul respectiv, adaugam X si facem click fals, iar pentru cazul O procedam similar si apelam functia ***play.*** In cazul in care din functia play nu iesim cu un rezultat pozitiv intram pe cazul de

remiza , unde nu avem niciun castigator, caz in care se va afisa RE-MI-ZA in mijlocul tabelei de joc iar celelate campuri vor fi *curatate(*vor fi resetate toate celelalte sase campuri, se va sterge orice caracter din cadrul lor si vom reseta background-ul la culoare initiala).

Figură 5-Remiza

Astfel in acest caz va aparea pe un rand de culoare rosie cuvantul remiza, despartit in silabe pe ficare camp din candrul liniei doi in tabela de joc, asa cum se poate observa si in *Figură 5* de mai sus.

Functiile ***check_for_win_x, check_for_win_o*** , implementeaza logica pentru a valida strategia castigatoare, ci anume verifica prezenta a trei caractere fie de *X* fie de *O* pe aceeasi verticala, orizintala sau diagonala. Daca aceasta conditie este indeplinita se incrementeaza scorul pentru jucatorul care a indeplinit conditia de victorie, si linia , coloana sau diagonala unde a avut loc amplasarea castigatoare va fi marcata cu o culoare gold insotita de urmatorul mesaj , *„X won this”* sau *„O won this”* in functie de castigator iar scorul jucatorului in cauza va fi incrementat cu un punct si va fi afisat in interfata grafica.



Figură 6-X/O castigatori

In *Figură 6* este prezentat cazul de victorie atat in cazul *jucatorului X* cat si in cazul *jucatorului O.*Se poate observa cum in acest caz scorul a crescut in functie de castigator si afisat in dreptul acestuia, iar locul unde s-a produs combinatia castigatoare a fost schimbata culoare si

inscriptionat mesajul impartit pe cele trei casute.In cazul in care nu exista o astfel de combinatie castigatoare nu vor exista niciuna din modificarile de mai sus si se va returna valoarea 0.

Functia ***play***, inglobeaza functiile *check_for_win_x, check_for_win_o,* astfel va avea loc o verificare a tuturor cazurilor posibile atat in cazul jucatorului X cat si in cazul jucatorului O.Pentru ambii jucatori exista opt conditii de verificare si anume, toate cele trei linii, cele trei coloane si cele doua diagonale (cea principala si cea secundara a tabelei), adica avem urmatoarele verificari:

- ✓ ***prima*** verificare este formata din butoanele: *btn1, btn2, btn3*
- ✓ ***a doua*** verificare este formata din butoanele: *btn4, btn5, btn6*
- ✓ ***a treia*** verificare este formata din butoanele: *btn7, btn8, btn9*
- ✓ ***a patra*** verificare este formata din butoanele: *btn1, btn4, btn7*
- ✓ ***a cincea*** verificare este formata din butoanele: *btn2, btn5, btn8*
- ✓ ***a sasea*** verificare este formata din butoanele: *btn3, btn6, btn9*
- ✓ ***a saptea*** verificare este formata din butoanele: *btn1, btn5, btn9*
- ✓ ***a opta*** verificare este formatadin butoanele: *btn3, btn5, btn7*

Functia ***new_game_function*** este o functie prin care incepem alt joc de la inceput, insa scorul anterior este pastrat. Functia verifica daca pe tabela de joc exista butoane in care sa fie mutari facute si se doreste resetarea acestora. Prin resetare stergem mutarea prezenta in acea casuta, adica stergem carcaterul de X sau O prezent acolo si restabilim culoarea fundalului la cea initiala, astfel ne va rezulta o casuta goala ca la inceput. A cest lucru este realizat pentru toate casutele din tabela de joc.

Functia ***reset*** reprezinta functia prin care resetam complet jocul, inclusiv resetarea scorului obtinut pana la acel moment de timp. Scorul este tinut cu ajutorul a doua variabile, *playerX* si *playerO* care initial au fost setate la 0 iar in urma executie jocului aceste se modifica, iar la apelarea functiei reset acestea sunt resetate la 0. Functia *reset* se baseaza pe functia *new_game_function* cu adaugarea valorilor resetate ale varabilelor ce tin scorul celor doi jucatori.

Atat functie ***new_game_function, reset,*** cat si ***play*** au fost folosite ca si componenta pentru *lamda function* [8] . Lamda function este folosita in tkinter pentru a putea sa introducem o functie ca si paramaetru al altei functii. Acest concept este foarte mult, deorece ne permite sa trimitemdatele intr-un mod mult mai rapid si eficient, atfel complexitate temporala si spatiala a algoritmului vor fi cat mai bune.

Functiile ***new_game_function, reset,*** au fost atribuite ca parametru in comdanda crearii butoanelor de *Reset* si *New Game*, atfel la o simpla apasare a acestor butoane se va declansa actiune speficifata de atributul *command=* si anume resetarea tabelei de joc sau inceperea unui nou joc.

4. Mod de utilizare

Am vrut ca modul de utilizare al aplicatie sa fie unul cat mai usor si placut utilizatorului.

In folderul proiectului, se va gasi executabilul *Tic-Tac-Toe Game*, care trebuie instalata pe calculatorul dumeavostra, aplicatie v-a descarca toate fisierele de care are nevoie ca sa ruleze.

Aplicatia/Installerul a fost realizat cu ajutorul *Nsis(*Nullsoft Scriptable Install System), care este un program folosit pentru a crea installerul unui aplicatii pe Microsoft Windows. [9]

O sa fie prezente in folderul proiectului atat aplicatia cat si toate fisierele, in caz ca vor aparea probleme la instalare. Dupa instalare se va deschide aplicatia si pe ecran va aparea interfata principala a jocului (cea prezenta in Figură 4Figură 4-Interfata grafica).

Primul jucator va marca o prima casuta cu *X*, urmand ca cel de-al doilea sa marcheze cu *O*.

Jocul va continua pana cand unul dintre cei doi jucatori va castiga sau va fi declarata remiza, de precizat este faptul ca jocul este unul *dual player*. Toate aceste etape su fost prezentate in sectiune trecuta si exemplificate prin: *Figură 5-Remiza*, este exemplificat cazul de remiza si *Figură 6-X/O castigatori* unde este prezentat cazul de castig pentru ambii jucatori.

5. Concluzii

Ca si concluzii finale in urma redactarii acestei documentatii as vrea sa precizez faptul ca am reusit sa parcurg si sa realizez toate obiectivele propuse initial in legatura cu tema aleasa.

Obiectivele initiale au fost implementarea unei interfete functionale care fie un prim contact cu utilizatorul, ulterior urmand ca interfata sa fie conectata la logica din spatele jocului, care a relizat functionalitatea propriu zisa a jocului si nu in ultimul rand realizarea unui executabil care sa poata sa partajeze aplicatie pe orice alta masina fizica ce ruleaza Windows cu precadere.Acest ultim pas a fost descris pe larg in capitolul anterior.

Deasemenea, datorita acelor functii de timp lamda codul este executat si mult mai rapid, deoarece am resusim sa atribuim o functie ca si parametru altei functii, ceea ce face ca executia sa fie una mult mai fluida.

Rezultatul final este o aplicatie functionala, care se adreseaza tuturor, dar in special copiilor, fiind o aplicatie user-friendly, acestia isi pot antrena strategiile de atac si aparare si fiind un joc care necesita doi jucatori, pot concura impreuna cu prieteni lor. De preferar ar fi, sa nu se intample acest lucru in timpul orelor de la scoala.

6. Referinte bibliografice

[1] Wikipedia, "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Tic-tac-toe#cite_note-EDSAC-11. [Accessed 07 05 2022].

[2] GEEKFLARE, "geekflare.com," [Online]. Available: https://geekflare.com/tic-tac-toe-python-code/. [Accessed 07 05 2022].

[3] R. S. S. Kevin Crowley, "Wikipedia," 1993. [Online]. Available: https://en.wikipedia.org/wiki/Tic-tac-toe#cite_note-16. [Accessed 07 05 2022].

[4] data-flair.training, "Data Flair," [Online]. Available: https://data-flair.training/blogs/python-tic-tac-toe/. [Accessed 07 05 2022].

[5] D. Oamen, "https://www.youtube.com/c/DJOamen/about," 2019. [Online]. Available: https://www.youtube.com/watch?v=mcP9qehN1_Q. [Accessed 07 05 2022].

[6] M. Nunez, "codefoxx," 2019. [Online]. Available: https://www.youtube.com/watch?v=E5bUdHQ_Ew0. [Accessed 07 05 2022].

[7] D. Amos, "Real Python," 22 03 2022. [Online]. Available: https://realpython.com/python-gui-tkinter/. [Accessed 07 05 2022].

[8] J. Bodnar, "ZetCode," 29 11 2021. [Online]. Available: https://zetcode.com/python/lambda/. [Accessed 07 05 2022].

[9] NSIS-site, "nsis.sourceforge.io," [Online]. Available: https://nsis.sourceforge.io/Download. [Accessed 07 05 2022].