**COLEGIUL NAȚIONAL ”SILVANIA” ZALĂU**

LUCRARE DE SPECIALITATE

**X ȘI 0**

Atestat profesional la informatică

**PROFESOR COORDONATOR: ABSOLVENT:**

**VLAICU CRISTINA NUNA DAVID CORNEL XII D**

**CUPRINS**

**I.** Motivul alegerii temei

**II.** Prezentarea limbajelor utilizate

**III.** Structura jocului

**IV.** Funcțiile utilizate

**V.** Bibliografie

**Motivul alegerii temei**

**X și O** (engleză *tic-tac-toe*) este un joc pentru doi jucători, "X" respectiv "0", care marchează pe rând câte o căsuță dintr-un tabel cu 3 linii și 3 coloane. Jucătorul care reușește primul să marcheze 3 căsute adiacente pe orizontală, verticală sau diagonală caștigă jocul.

În jocul reprezentat mai jos, primul jucător, "X", câștigă jocul .

[Jocul "X și 0" câștigat de jucătorul "X"](https://ro.wikipedia.org/wiki/Fi%C8%99ier:Tic-tac-toe-game-1.svg)

Am ales această temă deoarece voiam să învăț o parte mai complexă a paginilor web, și anume utilizarea unui limbaj de programare orientat pe obiecte ( în acest caz JavaScript ) pentru crearea unui mediu cu care utilizatorul poate interacționa. Am ales în special acest joc deoarece are un caracter simplist, fiind relativ ușor de transcris într-un program. Astfel, voiam să pun bazele unei pagini web care se axează pe interacțiune, nu pe diferite informații.

Jocul prezintă o interfață atractivă, accentul fiind pus atât pe partea tehnică cât și pe cea de design, pagina web conținând, pe lângă jocul în sine, muzică extravagantă și un fundal hipnotizant.

**Prezentarea limbajelor utilizate**

**HTML**

**HTML** (Hypertext Markup Language) este un limbaj creat în scopul de a descrie, în mod text, formatul paginilor Web; fișierele create în acest limbaj vor fi interpretate de navigatoare, care vor afișa paginile în forma doritã (cu texte formatate, liste, tabele, formule, imagini, hiperlegãturi, obiecte multimedia etc.). HTML a apãrut ca o aplicație ISO standard (aparține standardului SGML - Standard Generalized Markup Language, specializat pentru hipertext si adaptat la Web).

Așa cum se poate deduce din numele limbajului, HTML descrie caracteristicile de format ale elementelor incluse prin procedee de marcare. Acestea pot fi asemãnate intuitiv cu marcajele folosite în tipografie pentru a indica scrierea unui text cu un anumit tip de caractere. Fiecare element va fi introdus între douã marcaje ("tags", în limba englezã) - de început si sfârșit - (uzual) de forma <marcaj> … </marcaj>. Caracterele speciale de delimitare a marcajelor "<", ">" permit deosebirea acestora de textul propriu-zis. De exemplu, pentru textele aldine (îngroșate), marcajul de început este <B> iar de sfârșit - </B>.

Procesoarele de texte uzuale folosesc adesea procedee de marcare pentru formatãri (de exemplu, formatãrile de tip caracter din Word); în acest caz însã, caracterele de control introduse sunt ascunse iar rezultatul editãrii este direct vizibil ("What You See Is What You Get"). În schimb, în limbajele de marcare - inclusiv HTML - marcajele sunt introduse în text, astfel încât acestea sunt exclusiv succesiuni de caractere (litere, cifre, caractere speciale) - fișiere de tip text.

Limbajul HTML a evoluat în versiuni succesive, odatã cu evoluția protocolului HTTP și a programelor de navigare. Astfel, HTML 1.0 era compatibil cu Mosaic, primul program de navigare, dar dupã apariția unor navigatoare noi, a fost necesarã introducerea unui standard oficial Internet pentru construirea paginilor (HTML 2.0) și extinderea sa cu noi facilitãți: formule matematice, tabele, moduri avansate de descriere a organizãrii paginilor (începând cu HTML 3.0).

Standardizarea oficialã a limbajului HTML a fost realizatã de consortiul WWW si dezvoltatã de diversi producãtori de soft (unii dintre acestia urmãresc chiar promovarea navigatoarelor proprii prin introducerea unor particularitãți în formatele oficiale).

Paginile HTML se pot crea cu orice editor de texte de cãtre utilizatorii care cunosc limbajul HTML sau, mai simplu, se pot utiliza editoare speciale, în care obiectele se introduc interactiv iar codul HTML se genereazã automat. Având în vedere cã și în acest caz este utilã cunoașterea marcajelor generate pentru corectarea eventualelor erori (mai ales în cazul link-urilor), vom prezenta în continuare entitãțile care se pot introduce în paginile HTML si marcajele caracteristice acestora:

**Elementele limbajului HTML**

Toate obiectele HTML sunt introduse între marcaje; majoritatea acestora sunt de forma <tip\_obiect> (la început) și </tip\_obiect> (la sfârșit). Tipul standard al obiectului poate fi specificat cu majuscule sau minuscule; totuși, se recomandã utilizarea majusculelor fiindcã astfel marcajele ies în evidențã.

Majoritatea navigatoarelor permit vizualizarea paginii curente în formatul sursã HTML (forma pe care o interpreteazã pentru afișarea paginii). La interpretare, programele de navigare ignorã spațiile și <Enter>-urile, aplicând formatarea specificatã.

Existã însã și marcaje cu parametri; acestea au forma <tip\_obiect param1=valoare1 param2=valoare2 …>. De exemplu, obiectele de tip imagine sunt introduse cu delimitatorul <IMG>, care are diverși parametri. Pentru definirea hiperlegãturilor se folosește marcajul <A>, care are de asemenea parametri proprii.

**Marcaje de bazã**

O paginã Web este delimitatã de marcajele <HTML> si </HTML> care indicã începutul și finalul documentului și conține:

* zonã de antet cuprinsã între marcajele:     <HEAD> … </HEAD>
* un corp delimitat de marcajele:             <BODY> … </BODY>

Zona de antet este utilizatã de cãtre programele de cãutare pe site-urile web și permite specificarea titlului paginii care va fi afișat de navigator (nu apare propriu-zis în conținutul paginii).

titlul, introdus în antet, este cuprins între marcajele <TITLE> … </TITLE>.

**Marcaje pentru structurarea documentului**

Programele de navigare asigurã afișarea diferențiatã a unor titluri și subtitluri pentru secțiunile paginii, dupã criteriile implementate în acest scop la conceperea sa (litere mai mari sau mai mici, diverse culori, litere aldine sau simple, unul sau mai multe rânduri libere dupã titlu etc.).

Titlurile de capitole sau subtitlurile sunt definite de marcajele <Hn>, unde n este o cifrã între 1 si 6 care specificã nivelul titlului (1 este titlul principal iar 6 corespunde celui mai scãzut nivel). Astfel:  
<H1> … </H1>    indicã un subtitlu de nivelul 1  
…  
<H6> … </H6>    indicã un subtitlu de nivelul 6.

Pentru separarea zonelor paginii se pot folosi treceri la:

* linie nouã - cu marcajul <BR>
* paragraf nou - cu marcajul <P> (se insereazã o linie nouã și eventual se face o indentare). Marcajul </P> desemneazã sfârșitul de paragraf dar este mai rar folosit (se poate omite).

**Marcaje pentru formatarea textului si crearea listelor**

Formatãrile uzuale de texte permit scrierea cu caractere:

* aldine - în acest scop pentru acel text se aplicã marcajele     <B>… </B>;
* cursive (italice) - pentru acel text se aplicã marcajele    <I> … </I>;
* subliniate - textul se introduce între marcajele    <U> …</U>.

**Introducerea obiectelor de tip imagine (marcajul <IMG>)**

Limbajul HTML permite definirea mai multor tipuri de liste si imbricarea lor (includerea unor liste în altele), caz în care trebuie verificatã cu atenție corespondența dintre marcajele de început și sfârșit pentru fiecare listã. Formatarea listelor la afișarea paginii (introducerea bulinelor, numerotãrii etc.) se face de cãtre programul de navigare.

Elementele listelor se introduc între marcajele <LI> … </LI> ("lis1t item"). Modul de apariție al elementelor depinde de tipul de listã în care sunt incluse (cu buline, numerotate etc.), determinat de marcajul specific.

## Marcaje pentru crearea hiperlegãturilor

Hiperlegãturile ("hyperlinks") se introduc cu marcajele <A> … </A> ("anchor") aplicate asupra unui text sau a unei imagini. Ele se pot crea cãtre:

* adrese URL - astfel se asigurã accesul în cadrul procesului de navigare (prin protocoale specifice), nu numai la paginile web, ci și la diverse servicii Internet;
* fișiere / resurse locale adicã aflate pe același calculator cu pagina creatã - link-uri locale ; acestea ar putea fi accesate și printr-un URL cu protocolul "file" dar pentru mai multã simplitate, este suficientã specificarea numelui și cãii fișierului local;
* o zonã din documentul HTML curent - link-uri interne.

În fiecare din aceste cazuri, un click pe textul sau imaginea respectivã va determina activarea legãturii și afișarea prin intermediul navigatorului a resursei asociate link-ului.

Resursa asociatã unei legãturi este descrisã prin parametrii marcajului <A>:

* HREF - permite asocierea unei adresei URL, crearea unei legãturi locale prin specificarea numelui și cãii fișierului corespunzãtor sau a unei legãturi interne documentului; este cel mai important parametru pentru construirea de hiperlegãturi;
* NAME - numele hiperlegãturii sau al zonei dintr-o paginã cãtre care se face referirea, în cazul unui link intern;
* METHODS - metodele de acces (acest parametru se utilizeazã doar în prelucrãri speciale).  
  Astfel, un link local sau cãtre o adresã URL se poate defini pe un text cu: <A HREF="resursa"> text </A>.

**CSS**

**CSS** (Cascading Style Sheets) este un standard pentru formatarea elementelor unui document HTML. Stilurile se pot atașa elementelor HTML prin intermediul unor fișiere externe sau în cadrul documentului, prin elementul <style> și/sau atributul style.

**Sintaxa fişierelor CSS**

Sintaxa fişierelor CSS este foarte simplă – Aceasta constă dintr-o listă de:

– selectori (pentru identificarea tag-urilor)

– descriptori (pentru a indica ce vrem să facem cu acestea)

**Exemplu:** h1 {color: green; font-family: Verdana} spune

că tot ce este inclus în tag-ul h1 va fi afişat în verde utilizând

fontul Verdana.

Un fişier CSS va conţine deci o listă cu astfel de perechi ( selector, descriptor ).

– Selectori pot fi simple tag-uri HTML sau XML, dar se pot

defini şi moduri de combinare a tag-urilor.

– Descriptori sunt definiţi în standard-ul CSS.

**JavaScript**

**JavaScript** (**JS**) este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul JavaScript din aceste pagini fiind rulat de către browser. Limbajul este binecunoscut pentru folosirea sa în construirea siturilor web, dar este folosit și pentru accesul la obiecte încastrate (embedded objects) în alte aplicații. A fost dezvoltat inițial de către Brendan Eich de la Netscape Communications Corporation sub numele de Mocha, apoi LiveScript, și denumit în final JavaScript.

**Utilizare :**

Cea mai des întâlnită utilizare a JavaScript este în scriptarea paginilor web. Programatorii web pot îngloba în paginile HTML script-uri pentru diverse activități cum ar fi verificarea datelor introduse de utilizatori sau crearea de meniuri și alte efecte animate.

Browserele rețin în memorie o reprezentare a unei pagini web sub forma unui arbore de obiecte și pun la dispoziție aceste obiecte script-urilor JavaScript, care le pot citi și manipula. Arborele de obiecte poartă numele de Document Object Model sau DOM. Există un standard W3C pentru DOM-ul pe care trebuie să îl pună la dispoziție un browser, ceea ce oferă premiza scrierii de script-uri portabile, care să funcționeze pe toate browserele. În practică, însă, standardul W3C pentru DOM este incomplet implementat. Deși tendința browserelor este de a se alinia standardului W3C, unele din acestea încă prezintă incompatibilități majore, cum este cazul Internet Explorer.

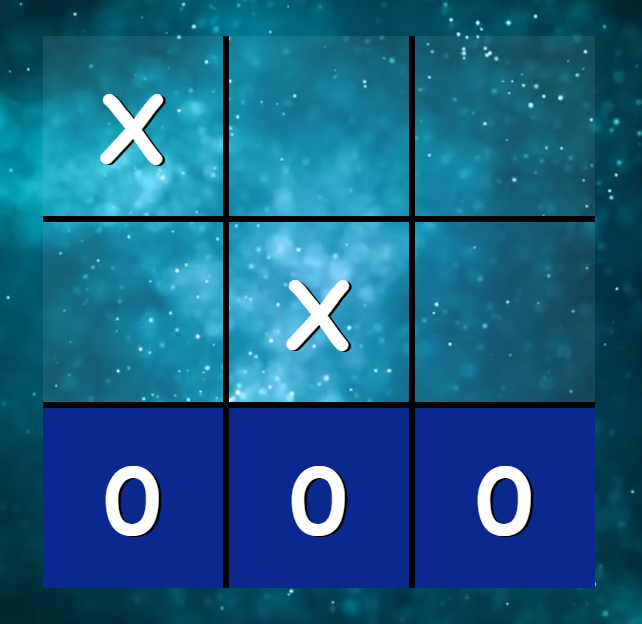
O tehnică de construire a paginilor web tot mai întâlnită în ultimul timp este AJAX, abreviere de la „Asynchronous JavaScript and XML”. Această tehnică constă în executarea de cereri HTTP în fundal, fără a reîncărca toată pagina web, și actualizarea numai anumitor porțiuni ale paginii prin manipularea DOM-ului paginii. Tehnica AJAX permite construirea unor interfețe web cu timp de răspuns mic, întrucât operația (costisitoare ca timp) de încărcare a unei pagini HTML complete este în mare parte eliminată.

**Structura jocului**

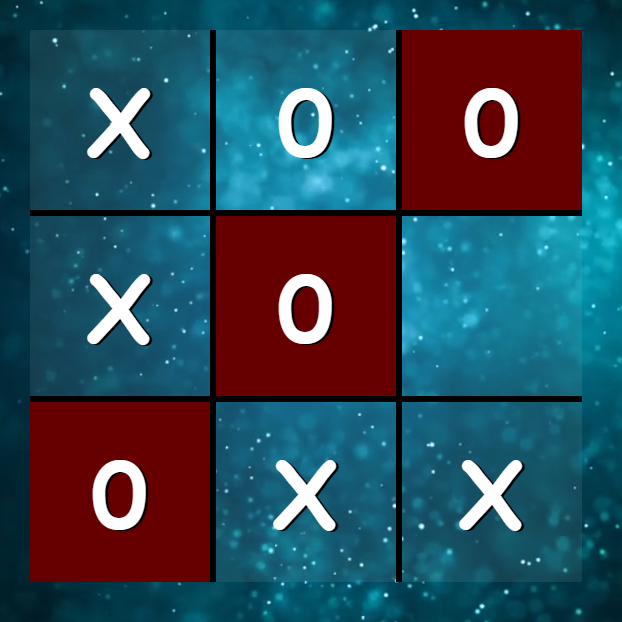
Interfața jocului este una simplistă, asemănătoare cu un joc clasic pe hârtie.



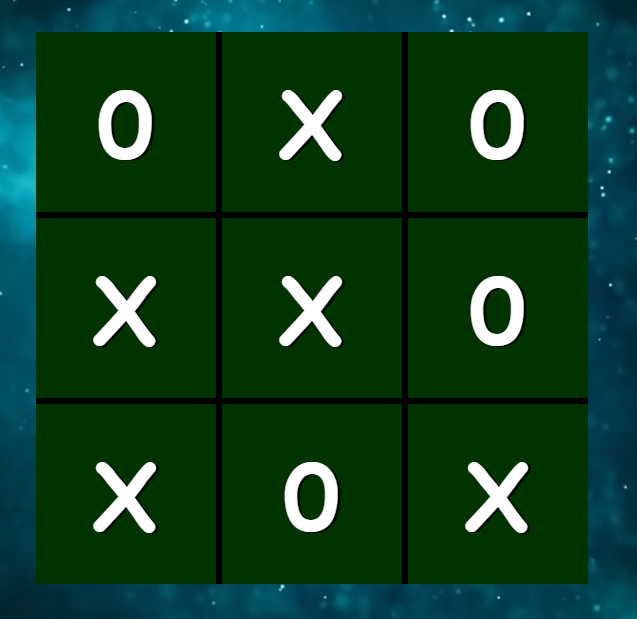
Jocul se desfășoară în mod clasic, calculatorul făcându-și mișcarea în termen de maxim o secundă de la mutarea jucătorului. Interfața jocului este sugestivă in momentul unei victorii, înfrângeri sau egalități :



( linia albastră marchează o victorie )



( linia roșie marchează o înfrângere )



( pătratul verde marchează o egalitate )

Încă de la începutul jocului, există mesaje, în funcție de mod, care ghidează jucătorul/jucătorii în scopul desfășurării cât mai plăcute a jocului.

De asemenea, există in total 5 melodii care pot fi pornite sau oprite după voia utilizatorului prin butonul de forma sau localizat în partea din stânga-jos.



După terminarea unei partide de joc, utilizatorul poate folosi butonul C:\Users\gfxne\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1.png ( „Încă o dată!” ) de sub interfața jocului, nefiind nevoie de un restart la paginii.

**Modul „Un jucător” :**

Modul „player vs computer”, acesta are 3 nivele de dificultate : ușor, mediu și imposibil.

Mesajele ce pot apărea in acest mod de joc sunt următoarele :



( mesajul inițial )



( mesajul ce apare după o mutare a calculatorului )



( mesajul ce anunță înfrângerea jucătorului )



( mesajul ce anunță victoria jucătorului, inexistent pe modul „imposibil” )

**C:\Users\gfxne\AppData\Local\Microsoft\Windows\INetCache\Content.Word\7.png**

( mesajul ce anunță o egalitate )

**Modul „Doi jucători” :**

Modul „player vs player”, acesta prezintă intefața jocului care trebuie completată de către cei 2 jucători.

Mesajele ce pot apărea in acest mod de joc sunt următoarele :

sau 



( mesajul inițial ce anunță jucătorul care va face prima mișcare )

sau



( mesaje ce apar după fiecare mutare, indicând simbolul următor )

sau



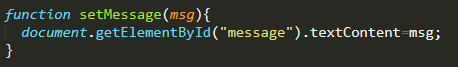
( mesaje ce anunță victoria lui X, respectiv O )

C:\Users\gfxne\AppData\Local\Microsoft\Windows\INetCache\Content.Word\7.png

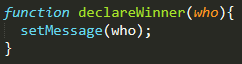
( mesajul ce anunță o egalitate )

**Funcțiile utilizate**

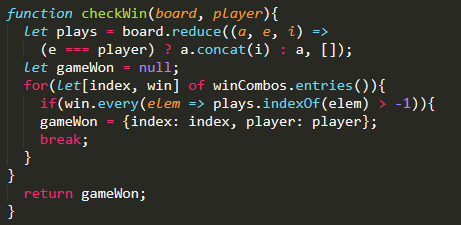
În vedea realizării jocului pentru modul „player vs computer”, programul utilizează o serie de funcții :



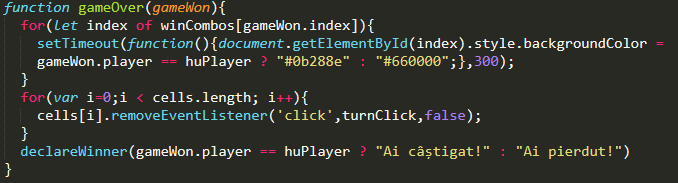
( funcția care determină apariția diferitelor mesaje )



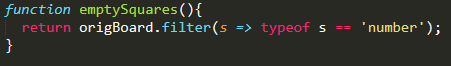
( funcția care declară câștigătorul )



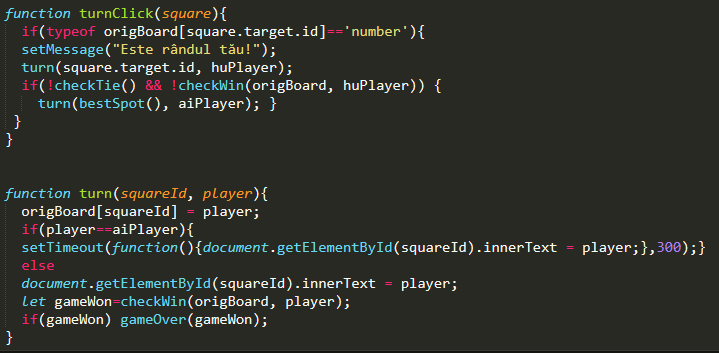
( funcția care verifică dacă există un câștigător )



( funcția care colorează pătrățelele corespunzător și încheie jocul )



( funcția care verifică dacă mai există pătrățele libere )



( funcțiile care inserează simbolurile după fiecare mutare )



( algoritmul utilizat pentru mutările calculatorului –> minimax )

**Bibliografie**

**1.**https://www.w3schools.com/css/default.asp

**2.**https://www.w3schools.com/html/default.asp

**3.**https://www.w3schools.com/js/default.asp

**4.**https://www.youtube.com/watch?v=P2TcQ3h0ipQ&ab\_channel=freeCodeCamp.org