Ministerul Educației și Cercetării al Republicii Moldova

Colegiul Universității Tehnice al Moldovei

**RAPORTUL**

**STAGIULUI DE PRACTICĂ**

**TEMA:**

Elevul:  **Grigoriev Artur**

Grupa: **PTPP-241**

Specialitatea: **Programarea si testarea produselor program**

Baza de practică: **[Unitate economică/instituție, localitate]**

Conducătorul stagiului de practică de la

Colegiul Universității Tehnice a Moldovei

**Grigoriev Artur**

Chișinău 2025

**REZUMAT**

General despre aplicatie (anexa nu se pune la numarul de foi)

**CUPRINS**

[INTRODUCERE 9](#_Toc196818805)

[1 ANALIZA DOMENIULUI DE STUDIU 10](#_Toc196818806)

[1.1 Importanța temei 10](#_Toc196818807)

[1.2 Sisteme similare cu proiectul realizat 11](#_Toc196818808)

[1.3 Scopul, obiectivele și cerințele sistemului 11](#_Toc196818809)

[2 REALIZAREA SISTEMULUI 12](#_Toc196818810)

[2.1 Descrierea la nivel de cod pe module 12](#_Toc196818811)

[2.2 Testarea sistemului 12](#_Toc196818812)

[CONCLUZII 13](#_Toc196818813)

[BIBLIOGRAFIE 14](#_Toc196818814)

[ANEXA A 15](#_Toc196818815)

# INTRODUCERE

În lumea contemporană, tehnologiile informaționale joacă un rol esențial, influențând profund felul în care trăim, comunicăm și ne desfășurăm activitățile zilnice. Aceste tehnologii nu sunt limitate doar la anumite domenii, ci pătrund în aproape toate aspectele vieții noastre, transformând societatea într-un mod semnificativ.

Impactul tehnologiilor informaționale asupra vieții cotidiene este evident în aproape fiecare activitate pe care o realizăm. De exemplu, dispozitivele inteligente simplifică sarcinile domestice, iar digitalizarea a schimbat fundamental modul în care lucrăm și interacționăm profesional.  Accesul generalizat la internet, folosirea dispozitivelor mobile a deschis noi metode de comunicare și colaborare – rețele sociale, locuri de activitate online – atât la lucru, cât și la educație. E-guvernarea, comerțul electronic, divertismentul digital sunt tot asemenea modalități în care tehnologia poate îmbunătăți experiența zilnică a oamenilor.

Aplicațiile software sunt instrumente fundamentale în acest proces de transformare, aducând îmbunătățiri semnificative în domenii diverse. În sănătate, aceste aplicații ajută la monitorizarea pacienților, diagnosticarea rapidă a bolilor și optimizarea gestionării dosarelor medicale, facilitând astfel accesul la servicii medicale de calitate. În educație, platformele digitale și instrumentele interactive permit accesul la resurse educaționale, adaptând procesul de învățare nevoilor individuale ale utilizatorilor și eliminând barierele geografice.

Procesul de dezvoltare al unei aplicații software implică mai multe etape bine definite. În prima fază, cea de proiectare, dezvoltatorii analizează cerințele utilizatorilor și creează prototipuri pentru a defini clar structura și funcționalitățile aplicației. Urmează etapa de programare, în care specificațiile sunt transformate în cod funcțional. Testarea riguroasă este apoi necesară pentru a asigura funcționarea corectă, securitatea și performanța aplicației. După lansare, mentenanța joacă un rol important în menținerea relevanței aplicației prin actualizări și adaptări continue la noile tehnologii și cerințe ale utilizatorilor.

În concluzie, dezvoltarea tehnologiilor informaționale și a aplicațiilor software reprezintă elemente indispensabile în progresul societății moderne, îmbunătățind calitatea vieții, eficiența economică și accesul generalizat la informație. Aceste tehnologii continuă să modeleze viitorul într-un ritm alert, oferind oportunități inovatoare și provocări captivante.

# ANALIZA DOMENIULUI DE STUDIU

Importanța dezvoltării aplicațiilor informatice este crucială în contextul actual al digitalizării, întrucât acestea facilitează învățarea principiilor de programare, modelarea datelor și implementarea inteligenței artificiale. Aplicațiile permit integrarea interacțiunii utilizatorilor cu logica de business și gestionează atât securitatea, cât și persistența datelor.

Proiectul propus – un joc Tic Tac Toe dezvoltat cu Phaser.js – se inspiră din analiza unor sisteme similare și adaugă funcționalități precum alegerea dimensiunii grilei și modul AI avansat bazat pe algoritmul minimax cu optimizare alpha-beta.

Scopul aplicației este de a crea o experiență de joc captivantă și educativă, cu interfață modernă, scor vizibil și animații grafice, oferind atât mod single-player, cât și multiplayer local. De asemenea, aplicația asigură autentificarea utilizatorilor și este construită modular, pregătită pentru extindere și scalabilitate ulterioară.

# Importanța temei

Importanța dezvoltării aplicațiilor informatice este esențială în contextul actual al transformării digitale, reprezentând un domeniu fundamental pentru înțelegerea și aplicarea principiilor programarii. Proiectele software permit explorarea interacțiunii dintre diverse subdomenii, precum elaborarea algoritmilor de decizie, gestionarea stării aplicației prin structuri de date adecvate, comunicarea eficientă între interfața utilizator și logica de business, precum și asigurarea securității datelor și a experienței utilizatorului.

În societatea contemporană, aplicațiile informatice constituie puncte de contact esențiale pentru începători în programare, facilitând înțelegerea controlului fluxului de execuție, manipularea structurilor matriciale sau a altor tipuri de date, precum și tratarea evenimentelor generate de interacțiunea utilizatorului și actualizarea dinamică a interfeței. Mai mult, aceste proiecte pot integra elemente de inteligență artificială, prin implementarea unor agenți capabili să analizeze situații complexe și să optimizeze procesul decizional, evidențiind astfel potențialul tehnologiilor avansate în dezvoltarea aplicațiilor.

Persistența datelor reprezintă un aspect crucial, iar utilizarea bazelor de date, chiar și în forme simple precum stocarea locală, pregătește tranziția către arhitecturi client-server, în care backend-ul gestionează utilizatori, statistici și istoricul operațiunilor. De asemenea, definirea clară a interfețelor web și mobile, realizată prin tehnologii standard precum HTML, CSS și JavaScript, subliniază importanța accesibilității și portabilității aplicațiilor, permițând utilizatorilor să acceseze serviciile indiferent de dispozitivul folosit.

Securitatea cibernetică, abordată la nivel conceptual, impune o atenție sporită asupra validării datelor de intrare și izolării logicii critice. Astfel, dezvoltarea aplicațiilor devine un proces complex, care combină aspecte tehnice cu responsabilitatea privind protecția datelor și integritatea experienței utilizatorului.

În concluzie, dezvoltarea aplicațiilor informatice are un caracter didactic și inovator, demonstrând că procesul nu se limitează la implementarea inițială, ci continuă prin integrarea de module avansate, interfețe sofisticate, funcționalități sociale și analize statistice ale performanței. Acest domeniu reprezintă o punte între teoria informatică și aplicabilitatea practică, deschizând perspective largi în cadrul transformării digitale a societății.

# Sisteme similare cu proiectul realizat

În dezvoltarea aplicației TIC TAC TOE, analiza sistemelor similare este esențială pentru identificarea celor mai bune practici și direcții de îmbunătățire. Sistemele existente variază de la aplicații web simple la platforme cu interfețe grafice avansate și funcționalități extinse. Proiectul propriu, realizat cu Phaser.js, integrează elemente esențiale din aceste sisteme, oferind moduri single-player și multiplayer, precum și opțiuni de grilă variabilă și inteligență artificială competitivă.

**Tabelul 1 Sisteme Similare cu proiectul realizat**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caracteristici | Tic Tac Toe Glow | Tic Tac Toe Online (BYRIL) | Google Tic Tac Toe | Proiectul propriu (Phaser.js) |
| Interfață | Animată, efecte vizuale atractive | Simplificată | Minimalistă | Clară, intuitivă |
| Dimensiune grilă | 3x3, 5x5, 7x7, 11x11 | 3x3 | 3x3 | 3x3 și 5x5 |
| Complexitate joc | Ridicată, strategie extinsă | Scăzută, dueluri rapide | Foarte scăzută | Medie, AI cu minimax și alfa-beta |
| Moduri de joc | Mod Battle cu scoruri și runde | Dueluri rapide | Joc simplu | Single-player și multiplayer local |
| Experiență utilizator | Implicare pe termen lung | Repetitivă, fidelizare redusă | Accesibilitate imediată | Echilibrată, educațională |

În tabelul 1 sunt prezentate funcționalitățile principale ale sistemelor similare existente. Tic Tac Toe Glow oferă o interfață animată, grile variate și moduri competitive care cresc implicarea utilizatorilor. Tic Tac Toe Online (BYRIL) are un design simplu și joc rapid, dar cu complexitate redusă. Google Tic Tac Toe este accesibil imediat, însă foarte minimalist. Proiectul propriu combină o interfață clară, opțiuni de grilă și un AI competitiv, oferind atât mod single-player, cât și multiplayer local. Astfel, proiectul integrează și adaptează elemente cheie pentru a crea o aplicație educațională și interactivă.

# Scopul, obiectivele și cerințele sistemului

Scopul principal al sistemului „Tic Tac Toe” este de a oferi o experiență digitală interactivă și atractivă, care valorifică simplitatea mecanicii clasice a jocului „X și Zero” și elimină monotonia jocurilor simple prin implementarea unor funcționalități suplimentare, precum alegerea dimensiunii tablei de joc precum „3x3” sau „5x5”, opțiunea de a juca împotriva unui adversar uman sau a unui agent de inteligență artificială și integrarea unei logici strategice avansate pentru modul automatizat. Sistemul răspunde astfel nevoilor de divertisment, stimulare strategică și accesibilitate, într-un cadru intuitiv.

Obiectivele sistemului vizează atât aspecte funcționale, cât și experiența utilizatorului. Sistemul trebuie să asigure o platformă robustă și intuitivă, care să permită desfășurarea jocului „X și Zero” în regim multiplayer local și single-player împotriva unui adversar controlat de inteligență artificială. Se urmărește diversificarea experienței prin setări configurabile, cum ar fi alegerea simbolului precum „X” sau „O” și selecția dimensiunii tablei de joc. De asemenea, se dezvoltă o interfață vizuală modernă, coerentă și modernă, care facilitează o interacțiune rapidă și plăcută. Un alt obiectiv este implementarea unui sistem de scor vizibil și a unor animații grafice pentru mutări, pentru creșterea implicării și rejucabilității.

Sistemul trebuie să permită autentificarea utilizatorilor și selecția modului de joc printr-un meniu interactiv. Utilizatorii pot alege să joace împotriva unui alt utilizator sau a unui AI. Tabla de joc se adaptează la dimensiunile selectate, iar sistemul gestionează alternarea tururilor și validarea condițiilor de câștig. În modul single-player, strategia AI este implementată prin algoritmul minimax cu optimizare alpha-beta, asigurând o provocare competitivă. Sistemul afișează în timp real scorurile și mesajele corespunzătoare pentru victorie, remiză sau finalul jocului.

Din perspectiva performanței, sistemul asigură un timp de răspuns redus și o redare fluidă a animațiilor. Securitatea accesului este garantată prin autentificare, realizată prin mecanisme simple bazate pe localStorage și cookie-uri, adecvate naturii academice a proiectului. Ergonomia interfeței respectă principiile accesibilității și clarității, asigurând compatibilitatea cu diverse dimensiuni de ecran prin stilizare adaptivă. Scalabilitatea este asigurată printr-o arhitectură modulară a codului, care permite extinderea ulterioară cu noi moduri de joc sau funcționalități. Sistemul oferă o experiență estetică, coerentă și intuitivă, contribuind la succesul transformării digitale în domeniul aplicațiilor recreative.

# 2 REALIZAREA SISTEMULUI

# 2.1 Descrierea la nivel de cod pe module

# 2.2 Testarea sistemului

# 

# CONCLUZII

# BIBLIOGRAFIE

# ANEXA A

Implementarea jocului Min-Max.js

(() => {

const DIRS = [[0,1],[1,0],[1,1],[1,-1]];

const opp = (p) => players.find(q => q !== p);

const clone = (m) => ({ ...m });

function has3Win(b,p){

for(let r=0;r<rows;r++)for(let c=0;c<cols;c++){

if(b[`${r},${c}`]!==p)continue;

for(const[dr,dc]of DIRS){

let ok=true;

for(let k=1;k<WIN\_LEN;k++){

const nr=r+dr\*k,nc=c+dc\*k;

if(nr<0||nr>=rows||nc<0||nc>=cols||b[`${nr},${nc}`]!==p){ok=false;break;}

}

if(ok)return true;

}

}

return false;

}

function findImmediate(b,p){

for(const k in b){

if(b[k]!=null)continue;

const n={...b,[k]:p};

if(has3Win(n,p))return k;

}

return null;

}

const MM\_DEPTH = 15, MM\_NODES = 60000;

const triples = (()=>{const t=[];

for(let r=0;r<rows;r++)for(let c=0;c<cols;c++)

for(const[dr,dc]of DIRS){

const r2=r+2\*dr,c2=c+2\*dc;

if(r2<0||r2>=rows||c2<0||c2>=cols)continue;

t.push([[r,c],[r+dr,c+dc],[r2,c2]]);

}

return t;

})();

const full = (b)=>Object.values(b).every(v=>v!=null);

function heuristic(b,me){

const foe=opp(me);let v=0;

for(const[[r1,c1],[r2,c2],[r3,c3]] of triples){

const a=b[`${r1},${c1}`],d=b[`${r2},${c2}`],e=b[`${r3},${c3}`];

const mc=(a===me)+(d===me)+(e===me),fc=(a===foe)+(d===foe)+(e===foe);

if(mc&&fc)continue;

if(mc===2)v+=40;

else if(mc===1)v+=5;

else if(fc===2)v-=45;

else if(fc===1)v-=6;

}

return v;

}

function alphaBeta(b,d,me,maxp,a,beta,nodes){

if(nodes.c++>=MM\_NODES) return 0;

if(has3Win(b,me)) return 1000-d;

if(has3Win(b,opp(me))) return -1000+d;

if(d>=MM\_DEPTH||full(b)) return heuristic(b,me);

if(maxp){

let best=-Infinity;

for(const k in b) if(!b[k]){

const n=clone(b); n[k]=me;

best=Math.max(best,alphaBeta(n,d+1,me,false,a,beta,nodes));

a=Math.max(a,best); if(beta<=a)break;

}

return best;

}else{

let best=Infinity,o=opp(me);

for(const k in b) if(!b[k]){

const n=clone(b); n[k]=o;

best=Math.min(best,alphaBeta(n,d+1,me,true,a,beta,nodes));

beta=Math.min(beta,best); if(beta<=a)break;

}

return best;

}

}

function play(key,p){

if(!key)return;

const [r,c]=key.split(',').map(Number);

const w=game.config.width/cols,h=game.config.height/rows;

const x=c\*w+w/2,y=r\*h+h/2;

matrix[key]=p;

const sprite=p==='x'?'ObjectX':'ObjectO';

const s=p==='x'?0.7:0.5, sc=game.scene.scenes[0];

const img=sc.add.image(x,y,sprite).setScale(0);

placedPieces.push(img); sc.tweens.add({targets:img,scaleX:s,scaleY:s,ease:'Back.easeOut',duration:300});

turn=!turn; Check.call(sc);

}

function aiMove(ai='o'){

if(gameOver) return;

let key = findImmediate(matrix, ai); // try to win

if(!key) key = findImmediate(matrix, opp(ai)); // try to block

if(!key){ // attack with minimax

let best=null,bestV=-Infinity,nodes={c:0};

for(const k in matrix) if(!matrix[k]){

const n=clone(matrix); n[k]=ai;

const v=alphaBeta(n,1,ai,false,-Infinity,Infinity,nodes);

if(v>bestV){bestV=v;best=k;}

if(nodes.c>=MM\_NODES) break;

}

key = best ?? Object.keys(matrix).find(k=>matrix[k]==null);

}

play(key,ai);

}

window.aiMove = aiMove;

})();