

Capitolul I

Introducere

Încă din vechiurile timpuri jocurile au fost privite ca o modalitate de divertisment dar și ca o forma de dezvoltarea a abilităților si aptitudinilor atât fizice cât și de gândire. În timp ce evoluția omului a fost una ascendentă, așa au evoluat si jocurile devenind tot mai importante în viața oamenilor.

Inițial, sintagma „joc video” reprezenta transpunerea pe un ecran semnalele video produse de un emițător de tip mecanic cu scopul de divertisment sau pentru educație. Însă în prezent, acesta a căpătat un alt sens, fiind o întreaga industrie cu un număr enorm de platforme și aplicații capabile de a genera divertisment sau de a provoca gândirea. În figura 1 putem observa câștigurile din industria de jocuri video din China (una dintre cele mai mari industrii din lume) care este într-o continuă creștere.

Unul dintre pionierii jocurile video este „Spacewar” un joc ce a fost conceput în anii 1960 dar abia 10 ani mai târziu, în 1970, a fost posibilă comercializarea acestuia deoarece prețul componentelor hardware s-a micșorat iar jocul s-a bucurat de o popularitate neașteptată.

Șahul se crede că a avut originea în India în jurul secolului al VI-lea . Jocul s-a răspândit în vest prin comerț, dar șahul a obținut statutul social și permanența mai puternic decât multe alte jocuri. Șahul a devenit un joc de îndemânare și tactici, forțându-i adesea pe jucători să gândească două sau trei mișcări înaintea adversarului lor doar pentru a ține pasul. Acest joc a devenit, de asemenea, acceptat de mulți drept un punct de reper pentru inteligență; oamenii care au devenit maeștri au fost considerați deștepți. Jocul înfățișează pioni, cai, regi, regine, nebuni și ture. Câteva portretizează poziții efective în armata istorică europeană. Fiecare piesă are un model unic de mișcare. De exemplu, calul este restricționat să se deplaseze într-o formă L în lungime de două pătrate și un pătrat în lateral, tura se poate deplasa doar în linie dreaptă pe verticală sau pe orizontală, iar nebunii se pot deplasa în diagonală pe tablă.

După ce au fost considerate doar o curiozitate, programele de șah pe calculator au crescut în abilitate până la punctul în care să poată concura în mod serios maeștri umani.

Kasparov clasat pe primul loc în lume, a jucat un meci de șase jocuri împotriva calculatorului IBM de șah Deep Blue, în 1996. Deep Blue a șocat lumea prin câștigarea primului joc, dar Kasparov a câștigat convingător meciul prin câștigarea a 3 jocuri și 2 remize. Meciul de

revanșă în 1997 a fost câștigat de către mașină, care a fost ulterior retrasă de către IBM. În octombrie 2002, Vladimir Kramnik a jucat într-un meci de opt jocuri contra programului de calculator Deep Fritz. În 2003, Garry Kasparov a jucat atât un meci de șase jocuri cu programul de calculator Deep Junior, în februarie, cât și un meci de patru jocuri împotriva lui X3D Fritz, în luna noiembrie.

Decada dintre anii 2006-2017 este epoca de aur a motoarelor de șah pentru calculatoare. Acestea au devenit atât de bune încât nici un jucător de șah uman nu le putea face față chiar dacă unele partide se jucau cu un ușor handicap pentru motoarele de șah. Până la sfârșitul anului 2017, nu s-a întâmplat nimic revoluționar. Puterea lor s-a schimbat constant, dar încet. Totul părea familiar și nimeni nu se aștepta la schimbări radicale, la descoperiri majore. Pe 5 Decembrie 2017 este lansat un nou motor de șah de către cei de la DeepMind (compania Google de inteligență artificială) care propunea o noua abordare. În loc de căutare alfa-beta și funcție de aproximare liniară pentru evaluarea poziției utilizate de motoarele tradiționale, AlphaZero folosește o funcție de aproximare neliniară bazată pe o rețea neuronală profundă și simularea Monte Carlo. În consecință, în loc să forțeze brute prin analiză, AlphaZero este capabil să „auto-învețe” șahul. Totodată Stockfishul rămâne un punct stabil în materia motoarelor de șah deoarece acesta este open source și se poate analiza modalitatea de gândire pentru a te perfecționa ca jucător de șah, înțelegerea anumitor principii solide ale jocului și un mediu prielnic pentru a modifica euristica acestuia în condițiile adaptării la nevoile personale.

Bucurându-se șahul de o popularitate crescută, au început să apară diferite variații ale acestuia cu scopul de a crea noi posibilități de joc, bazându-se în principiu pe aceeași tablă de joc și același efectiv de piese. Aceste reguli schimbate pot diferi de la modalitate de câștig până la poziția de start a pieselor. Dintre sutele de variații existente putem enumera cele mai populare:

- 3-Check: Presupune aceleași reguli ca jocul clasic doar cu condiția suplimentară că ești desemnat învins dacă pe perioada partidei acumulezi un număr mai mare de 3 șahuri.
- King of the Hill: este o variantă interesantă în care obiectivul este să-ți treci regele în centrul plăcii sau „vârful dealului”. Jocurile se pot încheia în moduri tradiționale de șah-mat, impas și time-out. Jocul se poate încheia și atunci când un jucător își mută regele în oricare dintre cele patru pătrate centrale: e4, d4, e5 și d5 sunt pătrate „de sus”, câștigând jocul la fața locului pentru alb sau negru, indiferent de natura poziției.

- Șah 960: Este o variantă inventată de regretatul campion mondial Bobby Fischer. Regulile jocului sunt aceleași ca șahul standard, dar în efortul de a reduce impactul teoriei deschiderii, piesele au fost amestecate la întâmplare pe randul din spate al fiecărui jucător. Jocurile se termină în modurile tradiționale de checkmate, impas și time-out. Nebunii trebuie să fie încă pe culori opuse în poziția inițială de Șah960. Regele trebuie să fie între ture pentru a menține capacitatea de a realiza rocada în ambele sensuri. Asta înseamnă că un rege poate fi plasat doar între pătratele b1 și g1 sau b8 și g8 pentru negru. Asta înseamnă:
 - Indiferent de locația regelui și a turei, se aplică reguli standard de rocadă. Trebuie să fie prima mișcare atât pentru rege, cât și pentru tură, regele s-ar putea să nu călătorească în sau prin șah și nu poate exista piese pe vreun pătrat prin care se traversează o piesă.
 - Regele și tura pun capăt procesului de rocadă în cazul în care ar urma într-o partidă de șah standard. Exemplu: Chiar dacă un rege alb este pe b1 și o tură este pe e1, rocada regelui ar putea implica glisarea regelui alb în g1 (pătratul său final de destinație), ceea ce va muta automat tura albă de la e1 la f1, întrucât Albul va avea întotdeauna regele pe g1 și tura pe f1 în poziția finală de rocadă.

Un motor(engine) de șah este pur și simplu un program software care joacă și analizează șahul. Cuvântul „motor” se referă pur și simplu la un fel de program cu putere mare, care face multă căutare și procesare - similar cu un motor de căutare.

Oamenii au fost întotdeauna fascinați de mașinării - inclusiv mașini de șah. În 1796, a fost creată o mașină falsă de joc de șah, numită „Automaton”, dar avea un jucător puternic de șah uman care se ascundea în jocul mișcărilor. În 1912, a fost creată o mașină care ar putea efectiv să dea șah cu o tura și un rege contra la un singur rege. Dar abia în 1951, Alan Turing a scris un program pe calculator care a putut juca șah. Pentru următorii 50 de ani, programatorii au lucrat la îmbunătățirea motoarelor de șah, iar îmbunătățirile hardware au permis un joc mai puternic. Până în 2005, motoarele de șah deveniseră cu siguranță mai puternice decât cei mai buni jucători umani. În anii de atunci, s-au îmbunătățit semnificativ, iar acum există sute de programe de calculator mai puternice decât cei mai talentați oameni la șah.

Oamenii s-au îmbunătățit la șah de-a lungul timpului, în timp ce învață din înțelepciunea colectivă a jucătorilor din trecut. Calculatoarele au accelerat acest progres, deoarece motoarele de

șah au adăugat noi cunoștințe și înțelegere jocului. Cei mai buni jucători folosesc astăzi motoarele de șah pentru a analiza pozițiile și a genera idei. Din păcate, acest lucru a introdus și înșelăciunea în șah, unde orice jucător care folosește doar un telefon mobil și un motor de șah poate juca mai bine decât orice maestru.

Motoarele de șah sunt complexe. Cu toate acestea, în termeni simpli, fac două lucruri importante:

1. *Evaluarea.* Motoarele de șah privesc pozițiile individuale și evaluează care poziție este mai bună. Aproape toate motoarele de șah afișează un număr de evaluare, sau „eval”, pe baza aceluiași punctaj pe care îl folosesc majoritatea jucătorilor de șah (un pion în valoare de un punct, o piesă minoră trei etc.). Fiecare motor de șah face acest lucru în mod diferit, dar majoritatea motoarelor privesc lucrurile precum materialul din fiecare parte, toate amenințările de pe tabla de joc, siguranța regelui și structura pionului. Scorul cumulat al celei mai bune evaluări în viitor este însumat la un număr. Motoarele tradiționale evaluează în mod similar cu oamenii, deoarece au fost proiectate de oameni. Motoarele cu rețea neuronală se evaluează diferit. Poziția din Figura 3 este obținută cu un scor cumulativ de +3 de către motorul de calcul Stockfish, chiar dacă materialul este egal, deoarece dezvoltarea pieselor lui White este mult mai bună. Aceasta înseamnă că poziția albă este cu aproximativ trei pioni mai bună.
2. *Căutarea.* Ca și jucătorii de șah buni, motoarele încearcă să privească profund în poziție. Cu cât pot vedea mai departe, cu atât mișcarea poate fi mai bună acum, deoarece pot evalua pozițiile care vor rezulta după cele mai bune mișcări posibile în viitor. Fiecare mișcare individuală de șah este numită „nivel” (un nivel) și adâncimea este explicată în câte nivele de adâncime. La 20 de nivele (10 mișcări albe și 10 mișcări negre), majoritatea motoarelor evaluează deja mult mai adânc și mai puternic decât oamenii. În funcție de timpul permis și de complexitatea poziției, motoarele pot arăta mai mult de 50 de nivele adâncime. Din poziția actuală, un motor începe să privească toate mișcărilor și răspunsurile posibile. Și apoi toate răspunsurile posibile la asta. Și apoi toate răspunsurile posibile la asta! Imaginează-ți că există 32 de mișcări posibile în orice poziție. După patru mișcări, există deja mai mult de un milion de poziții de evaluat. După doar alte patru mișcări, aceasta ar fi mai mult de un trilion de poziții. Asta devine extrem de nepractic. Deci, în schimb, motoarele încearcă să folosească „tăierea” inteligentă pentru a privi profund doar liniile cele mai promițătoare și să le ignore pe cele evidente. Motorul menține o „variație principală” (PV) în mișcare a

celor mai promițătoare mișcări în fiecare poziție. Motoarele tradiționale de șah folosesc funcții complexe de evaluare și algoritmi de căutare inteligentă pentru a găsi cea mai bună mișcare posibilă. Puterea lor este, de asemenea, legată de puterea procesării procesorului pe care o are telefonul, computerul sau serverul. Cu cât procesoarele sunt mai puternice și mai abundente, cu atât motorul devine mai puternic.

Capitolul II

Tehnologii utilizate

Interfața grafică

Windows Forms (cunoscut și sub denumirea de WinForms) este o bibliotecă a clasei de interfețe grafice folosite de către utilizator care este incorporată în .Net Framework. Scopul său principal este de a oferi o interfață mai ușoară pentru dezvoltarea aplicațiilor pentru laptop, tabletă sau calculatoare. WinForms pot fi folosite pentru crearea de aplicații fizice nu și pentru aplicații web. Aceasta librărie am folosit-o la interfața grafică a jocului de șah deoarece oferă un mediu de lucru ușor iar componentele sale comunică excelent între ele.

Există o interfață grafică open source denumită „All Pawns Must Die” care este o simplă interfață grafică pentru jocul de șah cu reguli clasice având implementat și un protocol UCI (interfață universală de șah). Această primă versiunea 1.0 este simplă prin design. Când este tura dumneavoastră, face click pe una dintre piesele dumneavoastră, pentru a alege un set de mișcări legale din care să alegeți (sunt evidențiate cu galben). Dacă faceți click pe unul dintre acele pătrate galbene, mișcare este preluată și engine-ul face următoarea mutare, repetați procedeul până când unul dintre jucători ajunge într-o poziție de șah mat sau este remiza sau ieșiți subit din partidă. Ultima mișcare este întotdeauna evidențiată cu culoarea albastră (vezi Figura 1) cu o săgeată pentru a afișa de unde a plecat piesa și unde a ajuns.

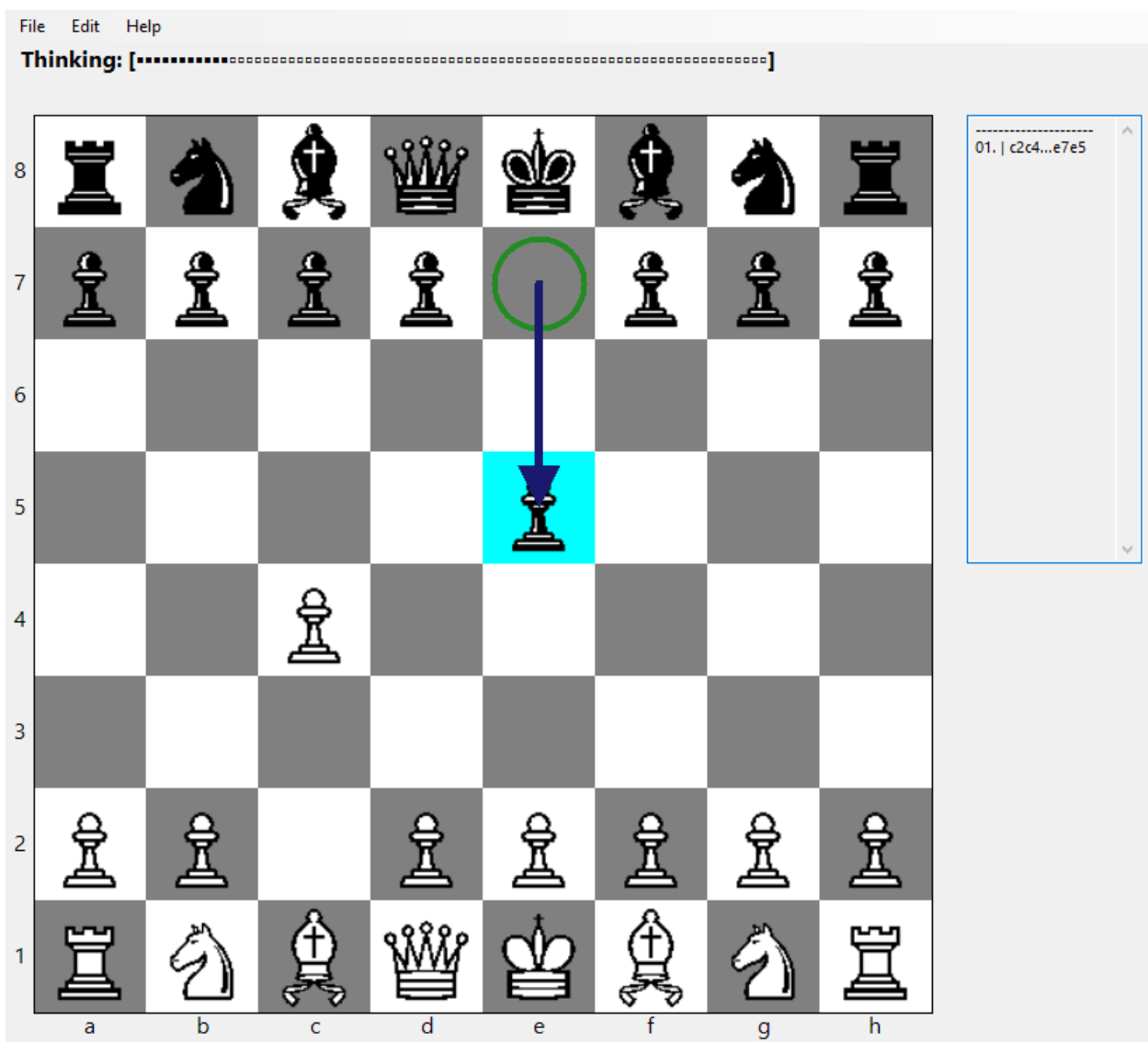


Figura 1: Interfața grafică

UCI (interfață universală de șah) este un protocol de comunicare deschis pentru ca motoarele (engine) de șah să joace automat jocuri, adică să comunice cu alte programe, inclusiv interfețe grafice pentru utilizatori. UCI a fost dezvoltat și proiectat de Rudolf Huber și Stefan Meyer-Kahlen și lansat în noiembrie 2000. Acesta a înlocuit la scară largă vechiul protocol de comunicare a motorului de șah (WinBoard/XBoard).

Stockfish este un motor de șah open-source dezvoltat de o mare comunitate de pasionați și dezvoltatori de motoare de șah. A fost cel mai puternic motor tradițional de șah din 2016. Multe dintre metodele moderne de programare a motorului de șah au avut ca și pionier pe Stockfish. Utilizează o formulă complexă de evaluare și căutare Alfa-Beta. Acesta este eficient deoarece este scris în C++.

Interfața „All Pawns Must Die” creează un fir nou de execuție ce se va ocupa exclusiv de comunicarea cu engine-ul. Aceasta funcție arată sub forma:

```
ICChessEngineProcess Start(string fullPathToExe, DataReceivedEventHandler dataReceivedHandler);
```

Primește ca și argument 2 parametri, primul fiind calea completă către sursa engine-ului inclusiv fișierul .exe iar al doilea parametru descrie un handler pentru a primi apeluri de tip callbacks cu date de la interfață. După ce s-a realizat crearea procesului care are rolul de comunicare cu engine-ul, acesta este inițializat adică se realizează conexiunea între engine și interfață. În cele din urmă se poate da comenzi engine-ului cu ajutorul funcției:

```
public UciSetOptionCommand(string optionName, string optionValue = null);
```

Cei 2 parametri descriu numele opțiunii și valoarea asociată comenzii dacă este cazul. Câteva exemple de comenzi ce i se pot da engine-ului:

- **Debug Log File-** Scrieți toate comunicările către și de la motor într-un fișier text.
- **Contempt-** O valoare pozitivă pentru dispreț favorizează pozițiile din mijlocul jocului și evită remize.
- **Threads-** Numărul de fire de execuție utilizate pentru căutarea unei poziții. Pentru cele mai bune performanțe, setați acest lucru egal cu numărul de nuclee de procesor disponibile.
- **Skill Level-** Scădeți nivelul de îndemânare pentru a face jocul de pe Stockfish mai slab.
- **UCI_LimitStrength-** Activați jocul mai slab care vizează o evaluare Elo așa cum este stabilit de UCI_Elo. Această opțiune depășește opțiunea de Skill Lever.
- **UCI_Elo-** Mărește puterea motorului la un Elo dat.
- **Minimum Thinking Time-** Căutați cel puțin x milisecunde pe mutare.

La fiecare mutare a playerului, i se transmite engine-ului FEN-ul la tabla curentă iar acesta determină care este mutarea cea mai bună pentru acesta. În caz că engine-ul nu întoarce nici un rezultat înseamnă că este remiză.

```
private void OnEngineBestMoveResponse()
```

Este o funcție din interiorul interfeței grafice și are rolul de a verifica răspunsul întors de engine. Dacă acesta nu există înseamnă că este remiză iar dacă acesta există se parsează string-ul returnat pentru a se extrage poziția inițială și poziția finală a piesei mutate pentru a se update tabla de șah.

```
private bool PlayerMated(PieceColor color, out bool positionIsStaleMate);
```

Interfața verifică la fiecare mutare dacă jucătorul sau computer-ul se află în șah mat pentru a termina partida. Primul argument reprezintă culoarea jucătorului care se verifică dacă este în șah iar al doilea parametru reprezintă Stalemate (reprezintă o poziție în care nu ești în șah dar nici nu mai ai mutări posibile).

FEN (Forsyth-Edwards Notation) este o notație ce oferă toate informațiile necesare pentru a descrie tabla de șah la un moment dat. Nu descrie ce s-a întâmplat înainte de acel moment din joc. Cu toate acestea, arată dacă există posibilități de rocadă, a căror mișcare este și numărul de mutări jucate până acum. O înregistrare FEN conține șase câmpuri. Separatorul dintre câmpuri este un spațiu. Câmpurile sunt:

- Amplasarea pieselor (din perspectiva jucătorului alb). Fiecare rang este descris, începând cu rangul 8 și terminând cu rangul 1; în fiecare rang, conținutul fiecărui pătrat este descris din fișierul "a" prin fișierul "h". În urma notării standard algebrice (SAN), fiecare piesă este identificată printr-o singură literă preluată din denumirile engleze standard (pion = "P", cal = "N", nebun = "B", tură = "R", regină = „Q” și rege = „K”). Piesele albe sunt desemnate folosind litere mari ("PNBRQK"), în timp ce piesele negre folosesc litere mici ("pnbrqk"). Pătratele goale sunt notate folosind cifrele 1 până la 8 (numărul de pătrate goale) și "/" separă rândurile.
- Culoare activă. „w” înseamnă alb va muta tura viitoare, „b” înseamnă negru va muta tura viitoare.
- Disponibilitate la rocadă. Dacă niciuna dintre părți nu poate face rocada, aceasta este „-”. În caz contrar, aceasta are una sau mai multe litere: „K” (albul poate casta regele), „Q” (albul poate casta regina), „k” (negrul poate casta regele) și / sau „q” (negru poate casta regina). O mișcare care împiedică temporar rocada nu anulează această notație.
- Se notează poziția „-”, dacă nu există poziția disponibilă en passant iar dacă există se notează cu locația acesteia. En passant (din franceză, „în trecere”) reprezintă o manevră specială, asemănătoare cu rocada, ce poate fi realizată în timpul unui joc de șah. Regula en passant se aplică atunci când un jucător mută unul din pionii săi două câmpuri înainte, față de poziția inițială a pionului, și trece pe lângă un pion advers de către care ar fi putut fi capturat dacă ar fi fost mutat doar un câmp înainte.
- Se notează numărul de jumătăți de mișcare de la ultima captură de piesă sau de la ultima mutare de pion pentru a se decide dacă se poate remiza.

- Numărul de mutări complete începe de la 1 și este incrementat după mutarea lui jucătorului negru.

Poziția de start a tablei de șah transcrisă în FEN:

rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1

Interfața grafică „All Panws Must Die” asigură un suport de parsare al FEN-ului:

```
public static List<ChessPiece> ExtractPieces(string fen)
```

În funcția de mai sus este dat ca si argument un șir de caractere de tip FEN și returnează o listă cu piesele disponibile și locația acestora pe tabla de șah.

```
public static string ApplyMoveToFEN(string fen, string sanMove)
```

Această funcție are rolul de a returna un nou string FEN cu proprietatea ca s-a realizat mutarea sanMove (sanMove este un string care conține mutarea unei piese).

Un bitmap este un tip de organizare a memoriei sau format de fișier imagine utilizat pentru stocarea imaginilor digitale. Termenul de bitmap provine din terminologia programării computerului, adică doar o hartă a biților, o matrice spațială mapată de biți. Acum, împreună cu pixmap-ul, se referă în mod obișnuit la conceptul similar de o serie de pixeli mapate spațial. Imaginile mai rapide, în general, pot fi denumite imagini bitmap sau pixmap-uri, sintetice sau fotografice, în fișiere sau memorie. Interfață grafică mai asigură suport pentru preluarea pieselor de șah dintr-o fotografie de tip bitmap (.bmp) cu ajutorul funcției:

```
void IChessBoardView.SetBitmapImages(Bitmap pieceImages, Size pieceSize)
```

Primul argument este imaginea bitmap care conține datele piesei în care se presupune că piesele sunt dispuse în 2 rânduri, cu alb pe partea de sus și negru pe partea de jos. Comanda pieselor trebuie să fie K Q R B N P iar al doilea argument reprezintă dimensiunea dorită unei singure piese (lungime și lățime).

Bibliografie

- <https://www.chess.com/ro/blog/nic01ae/istoria-jocului-de-sah>
- <https://www.chess.com/article/view/chess-variants>
- <https://github.com/manixaist/AllPawnsMustDie>
- <https://www.chessprogramming.org/UCI>
- <https://www.chess.com/article/view/computer-chess-engines>
- https://en.wikipedia.org/wiki/Forsyth%E2%80%93Edwards_Notation
- <https://support.chess.com/article/658-what-are-pgn-fen>
- <https://sahpentrucopii.ro/regulile-sahului-en-passant/>
- <https://en.wikipedia.org/wiki/Bitmap>