# Șah în rețea

**19.01.2018**

**Îndrumător:**

**s.l. dr. ing. Daniel Morariu**

**Student:**

**Circa Dragoș, 221/3**

# Istoric Versiuni

| Data | Versiune | Descriere | Autor |
|---|---|---|---|
| 20.10.2017 | 0.1 | Creare proiect. Creat structura generala: clasa „Pawn” si derivatele ei, „Player”,si componentele din interfata grafica. | Circa Dragos |
| 21.10.2017 | 0.1.5 | Continuat structura. Creare logica pieselor. Restictii de mutare. Creare evenimente ale tablei de sah. Adaugat proprietati pentru modificarea ei usoara. Folosirea lor. Modificat interfata. | Circa Dragos |
| 22.10.2017 | 0.2 | Testare. Rezolvare bug-uri. Adaugare diverse functionalitati tablei. Mutarea codului in metode. Creare de constructori multiplii in clasele create pentru diverse cazuri. | Circa Dragos |
| 17.11.2017 | 0.3 | Implemnetat reguli de coliziune. Modificat medota „Move” a piesei pentru a primii tabla ca parametru. Mai multe buguri de rezolvat. | Circa Dragos |
| 18.11.2017 | 0.4 | Adaugare exceptiile de la regula pentru pion. Implementare istoric | Circa Dragos |
| 20.11.2017 | 0.4.2 | Testare. Adaugat posibilitatea de resetare a tablei si de incheiere a jocului | Circa Dragos |
| 30.12.2017 | 0.5 | Implementare comunicare retea. Rezolvarea problemelor cu firele de executie. Crearea de structurii pentru manipularea evenimentelor corespunzator dar si a trasmiterii de infromatii intr-e fire de executie. | Circa Dragos |
| 01.01.2018 | 0.5.7 | Restructurarea fisierelor. Crearea a inca unei clase pentru comunicare diferita, diferita tehnica. | Circa Dragos |
| 02.01.2018 | 0.5.8 | Mutarea intr-un modul (.dll) diferit. Modificat interfata pentru comunicare in retea. Modificat clasa de comunicare. | Circa Dragos |
| 12.01.2018 | 0.7 | Creare protocol. Suport finalizat | Circa Dragos |
| 14.01.2018 | 0.8 | Rearanjare interfata. adaugat status. Adaugare comentarii. | Circa Dragos |
| 26.01.2018 | 0.9 | Testare si finalizare. | Circa Dragos |

# Cuprins

# 1 Specificarea cerințelor software

## *1.1 Introducere*

Aplicația a fost creată pentru a permite jocul de șah între doi jucători pe aceeași rețea, indiferent de mărimea ei. Această aplicație nu va permite jucătorilor să încalce regulile normale de șah și au timp nelimitat per tură.

### 1.1.1 Obiective

Obiectivul acestui joc este același ca la șah-ul normal: Să capturezi regele inamic. Pentru aceasta se vor folosi piesele de pe tabla de joc în concordanță cu regulile normale ale șah-ului pentru a captura alte piese inamice sau a încolți regele. Abordarea variază în funcție de tactică. Jocul se poate desfășura pe o rețea închisă sau pe același calculator. Funcționalitățile sunt dar nu se limitează la:

- Restricționarea mișcărilor invalide dar și așteptarea rândului.
- Conectarea pe rețea cu un alt utilizator
- Portul poate fi selectat, deci se pot deschide o multitudine de instanțe ale aplicației pentru a permite utilizatorului să joace cu mai multe persoane diferite pe aceeași rețea în același timp.
- Fereastra poate fi redimensionată, aceasta păstrându-și calitatea și funcționalitatea.
- Utilizatorul poate să aleagă o preferință în ceea ce privește culoarea dar cel care creează jocul are prioritate

Ce ar completa aplicația ar fi:

- Implementare rocadă
- Implementare regula cu pionul când ajunge în capăt
- Nebunul și regina să nu poată sări peste piese
- Nu se poate reseta jocul. Funcțiile există dar obiectele sunt șterse din memorie
- Optimizări software (RAM si CPU utilizat)

### 1.1.2 Definiţii, Acronime şi Abrevieri

- Tablă de joc – partea din fereastră care conține cele 64 de căsuțe
- Căsuță – Pătrat (sau dreptunghi în funcție de dimensiunile tablei) ce conține o piesă și o poate desena
- Protocol de comunicație – Șirul de date ce este trimis de către aplicație spre a fi interpretat de aplicația conectată

### 1.1.3 Tehnologiile utilizate

- .NET Framework 4.5.2
- Microsoft Visual Studio 2017 și limbajul C# versiunea 6 pentru dezvoltare dar și diagrama de clase
- Microsoft Word 2016 pentru documentație
- Microsoft Visio 2016 pentru schema bloc
- Paint.net pentru editarea pozelor ce reprezintă piesele și fundalul cu litere și numere
- Protocolul de comunicație TCP/IP

## *1.2 Cerințe specifice*

Este important de luat în calcul că se blochează firul aplicației când utilizatorul deschide un joc deoarece aplicația așteaptă conexiuni.

- Conectarea pe rețea cu un alt utilizator – se face prin o conexiune de tip TCP/IP, folosind un protocol creat specific pentru acest joc.
- Portul poate fi selectat, orice număr între 5000 și 65535 $\Longrightarrow$60535 instanțe concurente
- Fereastra poate fi redimensionată, aceasta păstrându-și calitatea și funcționalitatea.
- Utilizatorul poate să aleagă o preferință în ceea ce privește culoarea dar cel care creează jocul are prioritate
- Utilizatorul este notificat de începerea jocului dar și despre mutările făcute în bara de status
- Utilizatorul poate vizualiza istoricul mutărilor precum și piesele pierdute
- Jocul se încheie automat iar controlul se blochează în momentul în care un rege a fost pierdut
- Utilizatorul poate să își seteze un nume
- Restricționarea mișcărilor invalide dar și așteptarea rândului.

# 2 Funcționalitate

## *2.1 Descriere*

### 2.1.1 Mutarea piesei

Mutarea unei piese poate fi declanșată din două motive:

- Utilizatorul a mutat-o cu ajutorul cursorului : Se fac verificările unei mutări valide și dacă este validă se adaugă mutarea în istoric și se trimite mutarea către oponent
- Oponentul a mutat o piesă și a trimis mutarea prin rețea : Nu se face nici o verificare dar se mută piesa și se adaugă în istoric

### 2.1.2 Trimiterea datelor către oponent

Datele se trimit cu ajutorul librăriilor .NET care conțin namespace-urile `System.Net.Sockets` și `System.Net`. Protocolul este după cum urmează (în ordinea octeților) pentru trimiterea mutărilor

1. Marker de verificare: „M”
2. Sparator
3. Coloana poziției de plecare
4. Linia poziției de plecare
5. Separator
6. Coloana poziției pe care s-a mutat piesa
7. Linia poziției pe care s-a mutat piesa

Conexiunea se realizează în fire de execuție separate – 1 pentru server și se creează câte unul de fiecare dată când se trimit date care se închide după ce trimiterea este cu succes (sau aruncă or excepție).

## *2.2 Fluxul de evenimente*

### 2.2.1 Fluxul de bază

În continuare vom vorbi despre ce se întâmplă când utilizatorul pornește aplicația și dorește să deschidă un joc. De menționat este că vor fi prezente 2 obiecte de tip „Player”, din care cel cu numele „player1” va fi utilizatorul curent iar „player2” va fi oponentul.

După ce se execută toate inițializările care vin odată cu derivarea clasei principale din „Form” Se va inițializa tabla cu pătrate goale. Culorile vor alterna în funcție de setările făcute prin proprietățiile `WhiteBackgroundColor` și `BlackBackgroundColor` ale tablei de joc. În această inițializare se va creea o matrice de 8x8 ce va conține căsuțele. Tabla se va împărți în 9x9 pentru a lăsa spațiu pentru numerotare.

```
private void ChessTable_Load(object sender, System.EventArgs e)
{
    int boxWidth = (Width - 0) / 9; // 0 is there for spacing between boxes
    int boxHeight = (Height - 0) / 9;
    Color currentAltColor = BlackBackgroundColor;
    bool alt = false;

    for (int i = 7; i >= 0; i--)
    {
        for (int j = 0; j < 8; j++)
        {
            if (alt)
            {
                currentAltColor = WhiteBackgroundColor;
                alt = false; //Change alternance
            }
            else
            {
                currentAltColor = BlackBackgroundColor;
                alt = true; //Change alternance
            }
            squares[i, j] = new ChessTableSquare(currentAltColor, SelectedColor,
                            new DummyPiece(new PiecePosition((EColumn)(i + 1), j + 1)))
                                        //Create new square, initialise it with a blank piece
            {
                Location = new Point(i * boxWidth + (i + 1) * 2, j * boxHeight + (j + 1) * 2),
                                                    // Calculate it's location
                Size = new Size(boxWidth, boxHeight)    // Use the precalculated size
            }; //Object initialiser
            squares[i, j].Enabled = Enabled; // Set enabled status
            squares[i, j].MouseDown += ChessTableSquare_MouseDown; //Register to click event
            this.Controls.Add(squares[i, j]); //Add to the control
        }
        alt = !alt; // Change alternance for new row
    }
}
```

După inițializare utilizatorul poate să pornească un joc sau să se conecteze la un joc deschis. În cazul în care utilizatorul dorește să creeze un joc acesta pornește componenta din modulul de rețea pentru a rula în mod server. Aplicația se va bloca până când se va conecta un alt utilizator.

În momentul în care oponentul s-a conectat se determină culoarea utilizatorului dar și a oponentului. Protocolul de conectare este după cum urmează:

- Primul octet culoarea 0,1 sau 2
- Următorii octeți reprezintă numele
- Ultimul octet reprezintă terminatorul de șir, dar de acesta se ocupă modulul de comunicare

Când se conectează aplicația-server așteaptă datele în formatul prezentat mai sus după care trmite aceleași informații către oponent.

Se marchează jocul ca început, se determină cine poate muta, se inițializează piesele și jucătorii și apoi tabla.

```
        public void InitialisePlayers()
{
    if (Player1 == null || Player2 == null)
        throw new NullReferenceException("One of the players has not been properly
initialised");

    for (int i = 0; i < Player1.Pieces.Count; i++)
    {
        squares[(int)Player1.Pieces[i].Position.Column - 1, Player1.Pieces[i].Position.Row -
1].RepresentedPiece = Player1.Pieces[i]; //Add pieces on corresponding positions
        squares[(int)Player2.Pieces[i].Position.Column - 1, Player2.Pieces[i].Position.Row -
1].RepresentedPiece = Player2.Pieces[i];
    }
}
```

În continuare aplicația va aștepta ca utilizatorul să facă o mutare (dacă este rândul utilizatorului) sau să primească date.

Când o piesă este mutată evenimetul „OnPieceMoved” va fi executat și piesa va fi mutata. Deasemenea se va adăuga în istoric mutarea, indiferent dacă o piesă adversară a fost luată.

```
private void MainChessTable_OnPieceMoved(object sender, PieceMovedEventArgs e)
{
    //Add to history
    Bitmap arrow = Properties.Resources.ArrowRight;
    using (Bitmap image = new Bitmap(e.MovedPiece.Picture.Width * 2 + arrow.Width,
e.MovedPiece.Picture.Height))
    {
        using (Graphics g = Graphics.FromImage(image))
        {
            g.DrawImage(e.MovedPiece.Picture, new Point(0, 0)); //Calculate location of each
picture and create a new one by joining the first piece with a arrow and the other piece (or a
blank piece)
            g.DrawImage(arrow, new Point(e.MovedPiece.Picture.Width, arrow.Height / 2 + 30));
            g.DrawImage(e.OverlappedPiece.Picture, new Point(e.MovedPiece.Picture.Width +
arrow.Width, 0));

            // HACK: Because the matrix index grows right and downwards and the real table
grows differently the display only will be shown as it should be. The position will remain
"wrong"
            historyDisplay.Items.Add(new PictureListboxItem(String.Format("{0} to {1}",
e.BeforePosition.ToString(), e.AfterPosition.ToString()), image));
        }
    }

    historyDisplay.TopIndex = historyDisplay.Items.Count - 1;

    //Notify the other player of the move
    if (networkConnection.Connected && e.MovedPiece.Owner == player1.Owner)
    {
        player1.CanMove = false;
        //Protocol is as follows: <move marker>;<from.toString>;<to.toString>
        if (isClient) //If is client send to set ip, if not respond to connected client
```

```
            networkConnection.Send("M;" + e.BeforePosition.ToString(false) + ";" +
e.AfterPosition.ToString(false));
        else
            networkConnection.Reply("M;" + e.BeforePosition.ToString(false) + ";" +
e.AfterPosition.ToString(false));
    }
}
```

### 2.2.2 Fluxuri alternative

#### 2.2.2.1 Utilizatorul se conectează la un joc creat

În momentul în care utilizatorul dorește să se conecteze la un joc existent se trimit datele formatate după protocolul de la fluxul de bază și se așteaptă datele de la server în același format. În cazul în care conexiunea nu a putut fi realizată utilizatorul este notificat și îi se acordă posibilitatea de corectare a datelor de conectare.

Fluxul este identic pe parcursul jocului, singura diferență fiind că datele se trimit la IP-ul setat prin metoda „Send()” a componentei „NetworkConnection” și nu prin „Reply()”

### 2.2.3 Pre-condiţii

Dacă utilizatorul dorește să creeze un joc trebuie să se asigure că portul selectat nu este folosit de o altă aplicație ce rulează. Portul implicit 8888 nu este folosit de nici o aplicație foarte cunoscută, majoritatea folosind 80 și 144.

Dacă utilizatorul dorește să se conecteze la un joc inexistent (IP, port greșit sau nu există joc deschis), este blocat de firewall sau orice alt motiv ce ține de rețea el va fi notificat și îi se va da posibilitatea să schimbe IP-ul și/sau portul și să încerce din nou.

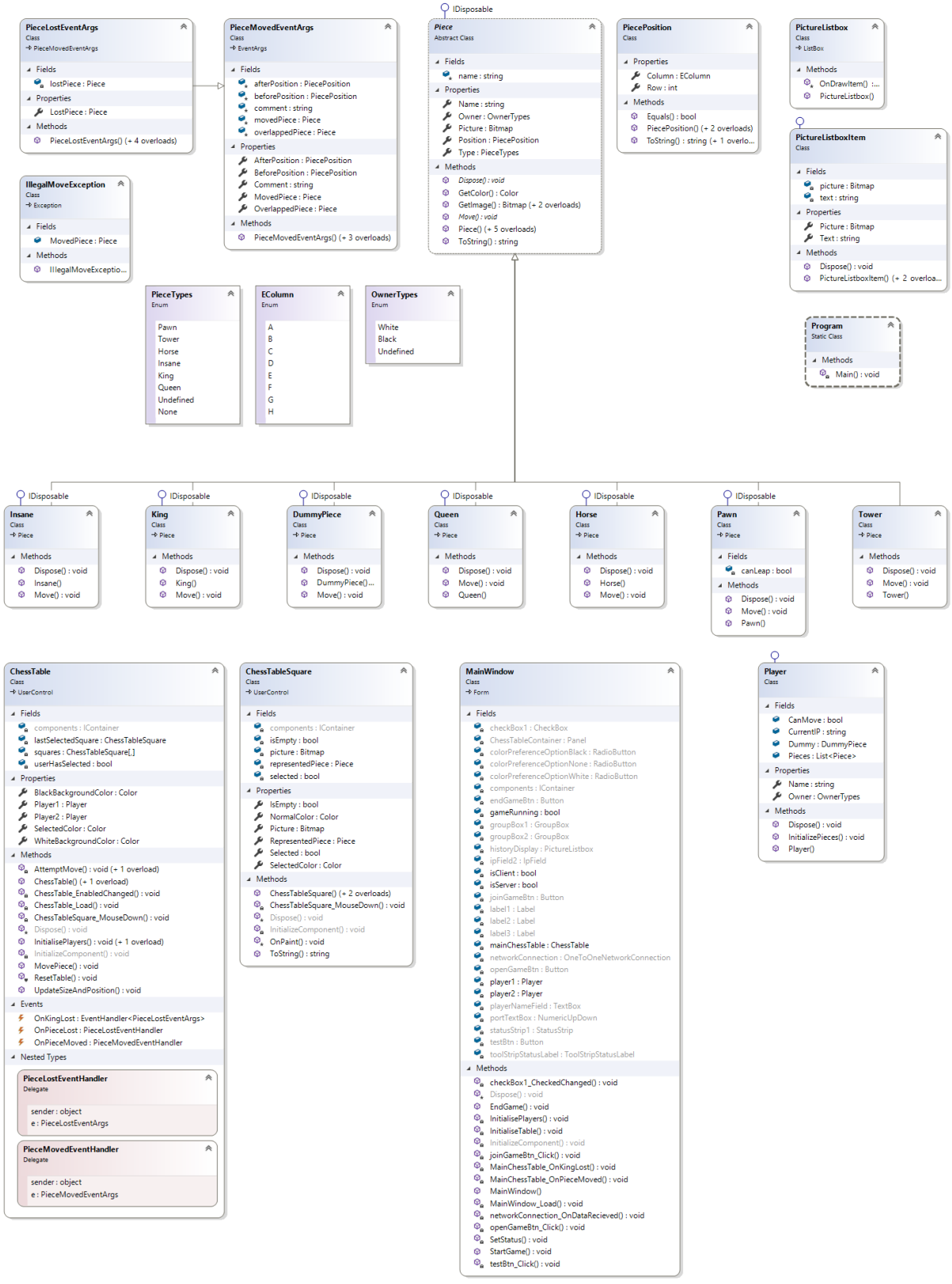
### 2.2.4 Post-condiţii

După încheierea jocului ( unul din regi a fost pierdut ) tabla se golește, piesele se șterg din memorie și aplicația intră într-o stare dezactivată. Pentru a putea porni un joc nou trebuie redeschisă aplicația.

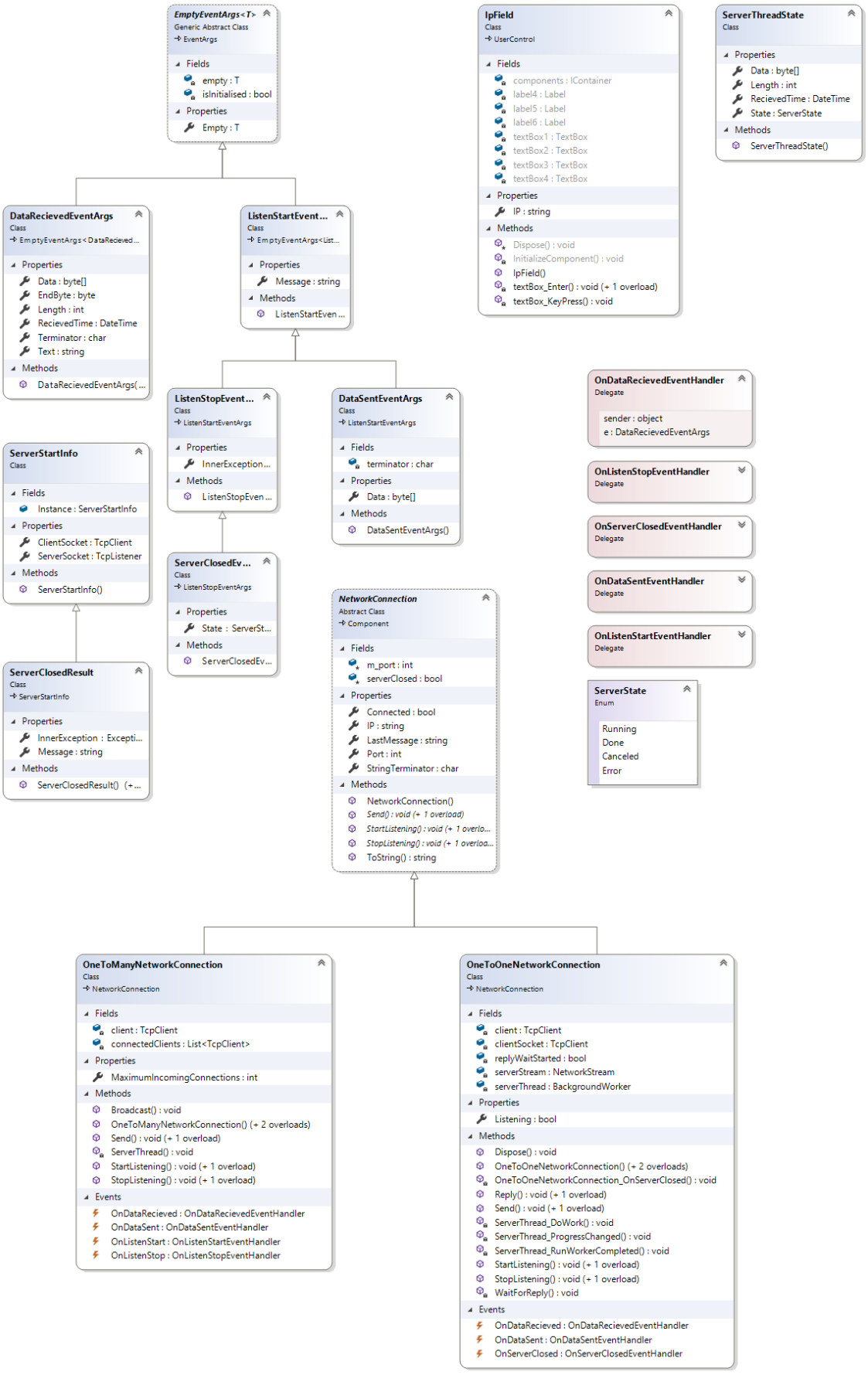
# 3 Implementare

## *3.1 Diagrama de clase*

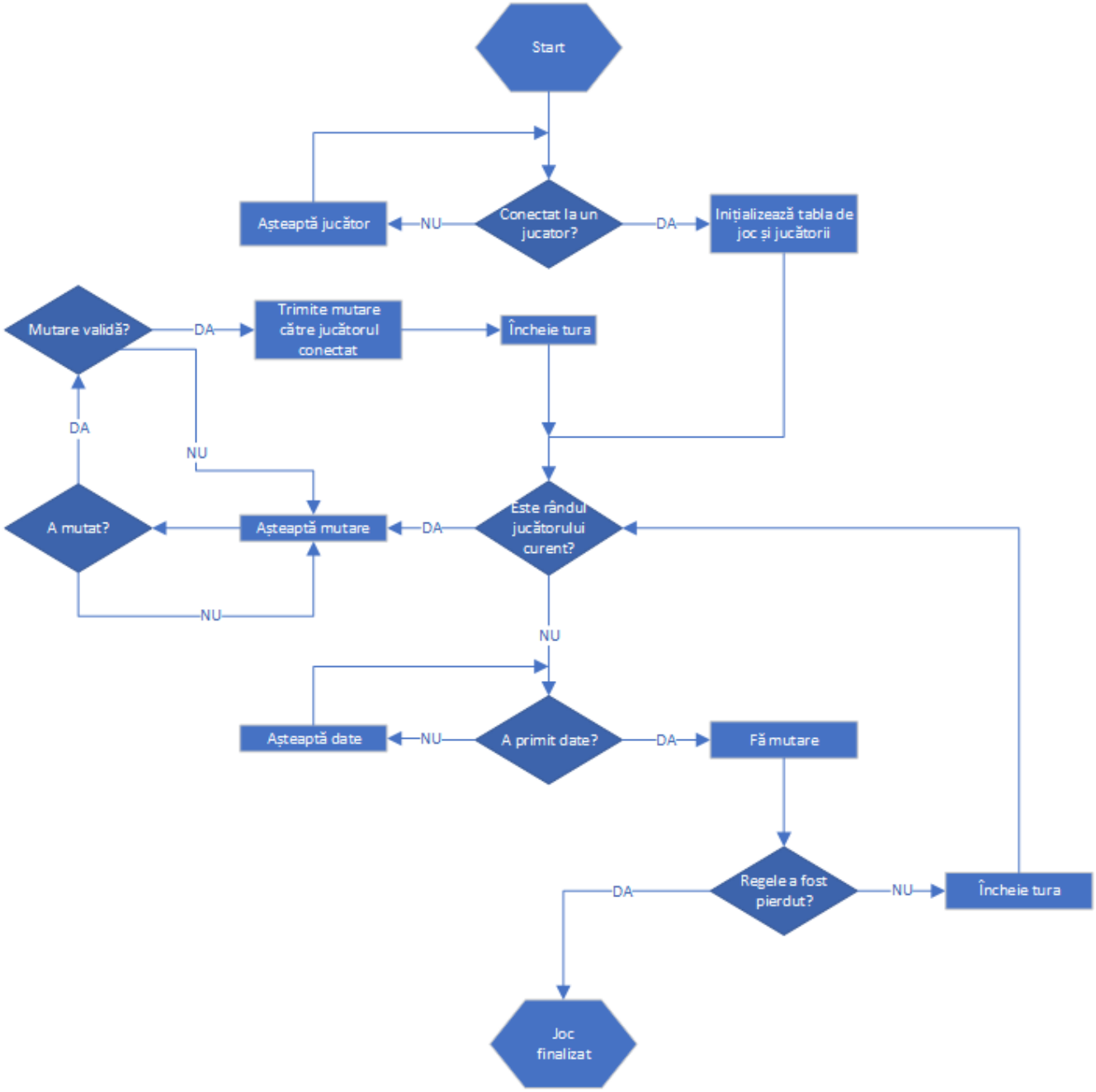
### 3.1.1 Diagrama claselor pentru jocul în sine



### 3.1.2 Diagrama claselor pentru modulul de comunicare prin rețea



## *3.2 Descriere detaliată*



# 4 Bibliografie

Iconiță:

- https://www.flaticon.com/free-icon/chess-pieces_79633#term=chess pawn&page=1&position=26

Stackoverflow:

- https://stackoverflow.com/questions/6909646/sending-message-from-one-pc-to-another-pc-using-ip-address
- https://stackoverflow.com/questions/5648605/c-sharp-event-inheritance
- https://stackoverflow.com/questions/2081953/user-control-custom-properties
- https://stackoverflow.com/questions/5339782/how-do-i-get-tcplistener-to-accept-multiple-connections-and-work-with-each-one-i

Microsoft Developer Network (msdn.com)

- https://social.msdn.microsoft.com/Forums/en-US/84c3e4d4-bbe1-4fe0-b553-a92f99c1cfeb/how-to-properly-close-a-tcpclient?forum=netfxnetcom

Imagini piese ( decupate, încadrare și scos background de către mine în paint.net):

- https://i.pinimg.com/736x/0c/5f/4c/0c5f4c6b328f69fd6bd6e4057cdd7b09--chess-tattoo-chess-pieces.jpg

Imagine background (tabla șah)

- https://www.google.com/search?tbs=sbi:AMhZZit0iBEuvbBbUSzSW6QLu2ABllXouPWo7pm7VU8xchpz2Up62x7zF4yEVDxpQdMt_145SajKU4YJvoPVOoz9TAOgmHT_1LRn3bqrM5qGXIPLR08hL0UOezJUFlFiI-ovNW6H2w_1i78-t0QqzUK4MeErIPCZ9OnUyROv4QpuEgMRmcML-BvEjQ_1morv6cX7H2-x2DXF1Jk7U5a6QEq5V3NM186qI9ndw73GSL41XIh-2y0WdvfNB6DLY3Rc-DE4Voi4t8NWFzPYM-oEod2_1CIDZdYd6bCXkDMgOA0uwMIDQJJ6gFXuQDIvZVcqCxZFaEy1yTa2UAHdE0yPIVxYWcJsJkQqK170xNA&hl=en-RO

Kit .Net framework

- https://www.microsoft.com/en-us/download/confirmation.aspx?id=42642

De asemenea am mai folosit stackoverflow.com și msdn.com pentru căutări minore cum ar fi „Cum se folosește operatorul ?. „