

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO  
KATEDRA INFORMATIKY

## BAKALÁŘSKÁ PRÁCE

Implementace deskové hry Stratego



2012

Jiří Hlaváček

## **Anotace**

*Bakalářská práce se zabývá vytvořením nezávislých podpůrných knihoven pro usnadnění implementace síťové komunikace mezi aplikacemi a použití platformy Microsoft XNA pro implementaci grafického uživatelského rozhraní. Dále bylo tématem práce implementovat za pomoci vytvořených knihoven deskovou stolní hru Stratego a tím vytvořit pilotní příklad pro použití vytvořených knihoven. Hlavním cílem bylo přiblížit použité technologie začínajícím programátorům v programovacím jazyce Visual Basic.*

Rád bych poděkoval vedoucímu práce Mgr. Martinu Dostálovi, Ph.D. za konzultace a podnětné připomínky. Dále bych chtěl poděkovat rodině za trpělivost a vstřícnost při tvorbě mé práce. Děkuji také kolegům a nadřízeným za podporu při vývoji a testování aplikace.

# Obsah

<b>1. Úvod</b>	<b>8</b>
<b>2. Aktuální stav řešené problematiky</b>	<b>9</b>
2.1. Ovládací prvky pro Microsoft XNA . . . . .	9
2.2. Knihovna pro síťovou komunikaci . . . . .	9
2.3. Implementace deskové hry Stratego . . . . .	9
<b>3. Použité technologie</b>	<b>10</b>
3.1. Programovací jazyk . . . . .	10
3.2. Tvorba uživatelského rozhraní . . . . .	10
3.3. Microsoft XNA 4.0 . . . . .	10
<b>4. Architektura aplikace</b>	<b>11</b>
4.1. Knihovna pro síťovou komunikaci . . . . .	11
4.1.1. Popis funkčnosti . . . . .	11
4.1.2. Diagram tříd . . . . .	12
4.1.3. Popis tříd . . . . .	17
4.1.4. Sekvenční diagram . . . . .	21
4.2. Knihovna pro tvorbu uživatelského rozhraní . . . . .	23
4.2.1. Popis funkčnosti . . . . .	23
4.2.2. Diagram tříd . . . . .	23
4.2.3. Popis tříd . . . . .	26
4.3. Aplikace Stratego . . . . .	28
4.3.1. Popis funkčnosti aplikace . . . . .	28
4.3.2. Use Case model . . . . .	28
4.3.3. Diagram tříd . . . . .	35
4.3.4. Popis tříd . . . . .	38
<b>5. Datová základna</b>	<b>44</b>
5.1. Databáze . . . . .	44
5.2. Textury . . . . .	44
5.3. Fonty . . . . .	44
<b>6. Instalace aplikace</b>	<b>45</b>
6.1. Systémové požadavky . . . . .	45
6.2. Průběh instalace . . . . .	45
6.3. Odstranění aplikace . . . . .	46
<b>7. Bezpečnost</b>	<b>47</b>
<b>8. Testování aplikace</b>	<b>48</b>

<b>9. Pravidla hry</b>	<b>49</b>
9.1. Obecný popis hry . . . . .	49
9.2. Hrací plocha . . . . .	49
9.3. Hrací kameny . . . . .	50
9.3.1. Nepohyblivé kameny . . . . .	50
9.3.2. Pohyblivé kameny . . . . .	50
9.4. Pohyb a útok kamenů . . . . .	52
9.4.1. Pohyb herními kameny . . . . .	52
9.4.2. Útok herními kameny . . . . .	53
9.4.3. Pravidlo dvou polí . . . . .	53
9.4.4. Pravidlo více polí . . . . .	54
9.5. Zahájení hry . . . . .	54
9.6. Průběh hry . . . . .	54
9.7. Vítězství ve hře . . . . .	54
<b>10. Uživatelská dokumentace</b>	<b>55</b>
10.1. Ovládání aplikace . . . . .	55
10.2. Hra proti počítači . . . . .	55
10.2.1. Nová hra proti počítači . . . . .	55
10.2.2. Rozestavění herních kamenů a zahájení hry . . . . .	57
10.2.3. Uložení a zrušení hry . . . . .	58
10.3. Hra po síti . . . . .	60
10.3.1. Vytvoření síťové hry . . . . .	60
10.3.2. Uložení a zrušení síťové hry . . . . .	61
10.3.3. Načtení uložené síťové hry . . . . .	61
10.3.4. Připojení se k síťové hře . . . . .	62
10.4. Nastavení grafiky . . . . .	63
10.5. Pravidla a nápověda . . . . .	63
10.6. Ukončení aplikace . . . . .	64
<b>Závěr</b>	<b>65</b>
<b>Conclusions</b>	<b>66</b>
<b>Reference</b>	<b>67</b>
<b>A. Obsah přiloženého CD</b>	<b>68</b>

## Seznam obrázků

1.	Diagram tříd knihovny LoxlyLAN - vztahy . . . . .	12
2.	Diagram tříd knihovny LoxlyLAN - AutoUDP_Server . . . . .	13
3.	Diagram tříd knihovny LoxlyLAN - AutoUDP_Client . . . . .	14
4.	Diagram tříd knihovny LoxlyLAN - TCP_Server . . . . .	15
5.	Diagram tříd knihovny LoxlyLAN - TCP_Client . . . . .	16
6.	Sekvenční diagram funkce AutoUDP_Server a AutoUDP_Client . .	21
7.	Sekvenční diagram funkce TCP_Server a TCP_Client . . . . .	22
8.	Diagram tříd knihovny LoxlyXNAControl varianta 1 . . . . .	23
9.	Diagram tříd knihovny LoxlyXNAControl varianta 2 . . . . .	24
10.	Diagram tříd knihovny LoxlyXNAControl varianta 3 . . . . .	25
11.	Use Case diagram pro hráče . . . . .	28
12.	Diagram tříd aplikace Stratego - varianta 1 . . . . .	35
13.	Diagram tříd aplikace Stratego - varianta 2 . . . . .	36
14.	Diagram tříd aplikace Stratego - varianta 3 . . . . .	37
15.	Hrací plocha s rozestavenými kameny . . . . .	49
16.	Pohyb pohyblivým herním kamenem . . . . .	52
17.	Pohyb Průzkumníkem . . . . .	52
18.	Útok pohyblivým herním kamenem . . . . .	53
19.	Útok Průzkumníkem . . . . .	53
20.	Volba „Hra“ v hlavním menu . . . . .	55
21.	Volba „Pro jednoho hráče“ v menu „Vyber typ hry“ . . . . .	56
22.	Volba „Začít novou hru“ v menu „Hra pro jednoho hráče“ . . . .	56
23.	Rozestavování kamenů . . . . .	57
24.	Generování rozestavení . . . . .	57
25.	Tlačítko „Zahájit hru“ . . . . .	58
26.	Tlačítko „Uložit hru“ . . . . .	58
27.	Dialog pro uložení hry . . . . .	59
28.	Tlačítko „Uložit hru“ . . . . .	59
29.	Dialogové okno pro zrušení hry . . . . .	59
30.	Volba „Hra“ v hlavním menu . . . . .	60
31.	Volba „Pro dva hráče - LAN“ v menu „Vyber typ hry“ . . . . .	60
32.	Volba „Vytvořit novou hru“ v menu „Pro dva hráče - LAN“ . . .	61
33.	Volba „Načíst uloženou hru“ v menu „Pro dva hráče - LAN“ . . .	62
34.	Volba „Připojit se“ v menu „Pro dva hráče - LAN“ . . . . .	62
35.	Volba „Nastavení grafiky“ v hlavním menu . . . . .	63
36.	Ukončení aplikace . . . . .	64

## Seznam tabulek

1. Systémové požadavky . . . . .	45
----------------------------------	----

# 1. Úvod

Tématem této bakalářské práce je implementace deskové stolní hry Stratego.

Hlavním cílem je však při samotné implementaci vyvinout potřebnou podpůrnou knihovnu pro tvorbu grafického uživatelského rozhraní při práci s platformou Microsoft XNA a knihovnu pro snadné užití asynchronní síťové komunikace pro realizaci komunikace v lokální síti. Tyto knihovny mohou být dále využity při tvorbě jiných projektů a mohou usnadnit dostupnost uvedených technologií středoškolským studentům pracujícím ve Visual Basicu. Dalším cílem je při implementaci hry oddělit jádro hry od uživatelského rozhraní a umožnit tím případnou nezávislou tvorbu uživatelského rozhraní pomocí jiných technologií.

Mojí motivací pro volbu těchto cílů byl fakt, že při výuce programování na střední škole projeví studenti zájem o tyto technologie při používání Visual Basicu a v době tvorby této práce neexistovala podpora pro používání Microsoft XNA ve Visual Basicu. Asynchronní síťová komunikace je pro některé studenty střední školy mimo rámec jejich aktuálních schopností, ale i přesto jsem jim chtěl zprostředkovat možnost využití této technologie vyvinutím podpůrné knihovny v rámci tvorby této práce.



## 2. Aktuální stav řešené problematiky

### 2.1. Ovládací prvky pro Microsoft XNA

V současné době existuje několik podpůrných knihoven pro použití ovládacích prvků v Microsoft XNA. Jsou tvořeny většinou otevřenou komunitou tvůrců v platformě XNA. Některé knihovny byly dokončeny až v průběhu tvorby této práce. Žádná však nesplňuje moji vizi grafického uživatelského rozhraní a především jsou implementovány v jazyce C# a pro sekundární cíl — podporu samostudia mých studentů střední školy učících se programovat ve Visual Basicu — jsou tedy nepoužitelné.

V aktuální době došlo k ukončení vývoje podpory XNA pro Visual Basic [1] a je tedy pravděpodobný rozvoj open source zdrojů a knihoven i v tomto programovacím jazyce. V současnosti je možné využít některou z knihoven ovládacích prvků pro XNA, které jsou implementovány v jazyce C# (Neoforce controls library, SimpleGUI controls library, RamGec controls library).

### 2.2. Knihovna pro síťovou komunikaci

Přes veškerou snahu se mi nepodařilo zjistit existenci jakékoliv podpůrné knihovny, která by nabízela využití asynchronní síťové komunikace pomocí TCP a UDP za použití jednoduchého na první pohled jednovláknového přístupu s relativně intuitivním využitím nabízených objektů a jejich metod.

### 2.3. Implementace deskové hry Stratego

Od roku 2007 probíhá mistrovství světa počítačových programů v deskové hře Stratego [2]. Existuje tudíž několik variant implementací této hry, které se účastní tohoto mistrovství a které se specializují na optimalizaci algoritmů pro generování tahů při této specifické hře. Nejúspěšnější implementací je Probe [3], která je dostupná jako freeware aplikace pro běžné uživatele, kteří chtějí hrát proti počítači. Existuje několik dalších implementací, některé se specializují na úroveň generování tahů počítačem a některé nabízí spíše síťové varianty hry proti přátelům.

## 3. Použité technologie

### 3.1. Programovací jazyk

Pro implementaci této deskové hry a jejích podpůrných knihoven jsem si vybral programovací jazyk Visual Basic, především protože v tomto jazyce sám vyučuji programování na střední škole. Vzhledem k použitým technologiím (především XNA) by bylo výhodnější provést implementaci v jazce C#, pro který je možné využít větší množství dostupné dokumentace a případně je možné se inspirovat rozsáhlejšími open source implementacemi složitějších řešení než ve Visual Basicu. Jelikož ve Visual Basicu není příliš mnoho řešených open source implementací technologicky náročnějších projektů, rozhodl jsem se i proto pro tento jazyk, aby bylo případně možné z mých implementací dále čerpat inspiraci pro další tvůrce v tomto jazyce. Dalším důvodem výběru tohoto jazyka byl fakt, že jsem si chtěl rozšířit obzory v možnostech zmíněného jazyka na platformě .Net, abych mohl dále získané zkušenosti využít při výuce programování na střední škole.

### 3.2. Tvorba uživatelského rozhraní

Pro implementaci grafického uživatelského rozhraní jsem zvolil platformu Microsoft XNA verze 4.0, která umožňuje využití technologie DirectX. Tuto technologii jsem zvolil především, abych s ní získal zkušenosti, které následně předávám svým studentům v rámci volitelného předmětu programování. Jako alternativu ke zvolenému řešení by se dalo využít použití formulářových ovládacích prvků pro standardní tvorbu pro OS Windows.

### 3.3. Microsoft XNA 4.0

Microsoft XNA [4] je vývojářská platforma zapouzdřující metody a atributy v kompilovaných knihovnách pro práci s technologií DirectX. Nabízí jednodušší přístup k implementaci aplikací využívajících tuto technologii. Ve verzi 4.0 umožňuje implementovat aplikace spustitelné kromě platformy Windows i na platformách Windows Phone a Xbox 360.

Microsoft XNA 4.0 umožňuje využití vývojářského prostředí Visual Studio 2010 a přidává po instalaci do Visual Studia šablony projektů pro vývoj aplikací používajících Microsoft XNA 4.0 pro programovací jazyk C#. V rámci těchto šablon projektů přidává funkční nástroje pro kompilaci zdrojových souborů (textur, 3D modelů, zvuků a fontů).

Microsoft XNA 4.0 nepodporuje šablony projektů pro Visual Basic v rámci Visual Studia 2010, je však možné tuto technologii využít vzhledem ke zjevné příbuznosti jazyků C# a Visual Basic (oba využívají knihovny .NET).

## 4. Architektura aplikace

### 4.1. Knihovna pro síťovou komunikaci

#### 4.1.1. Popis funkčnosti

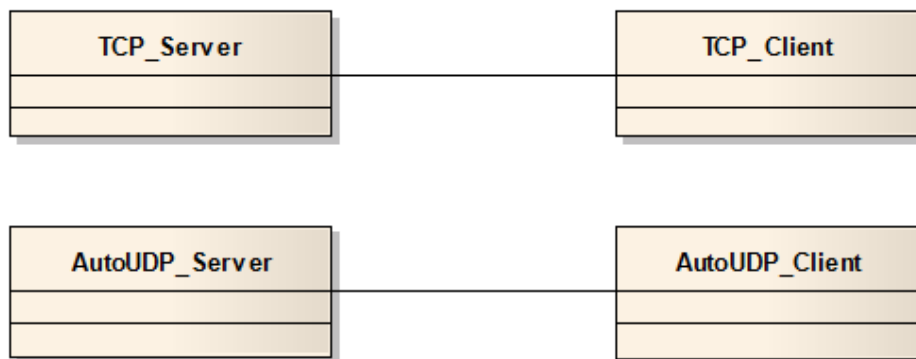
Knihovna pro síťovou komunikaci v sobě zapouzdřuje ve třídách asynchronní způsob implementace síťové komunikace (TCP i UDP). Pro použití zveřejňuje pouze omezený počet vlastností a metod, které umožňují spuštění funkčnosti TCP (UDP) serveru (klienta) bez potřebné znalosti technologie asynchronní síťové komunikace. Pro monitorování vnitřního stavu komunikace využívá především událostí a umožňuje tedy v cílové aplikaci implementovat síťovou komunikaci pouze za použití deklarace potřebných objektů a navázání událostních procedur.

Implementace je zaměřena především na použití protokolu TCP při komunikaci po lokální síti. Vzhledem k principu funkce protokolu TCP [5], který vytváří spojitou komunikaci, není možné dostatečně efektivně prohledávat lokální síť pro zjištění přítomnosti naslouchajícího serveru na konkrétním portu. Proto jsou v implementaci knihovny zahrnuty třídy, které umožňují využít výhody protokolu UDP [5], který poskytuje možnost snadno vysílat broadcast packety, které dávají možnost velice efektivně zjistit výskyt serveru, který využívá při implementaci stejnou knihovnu.

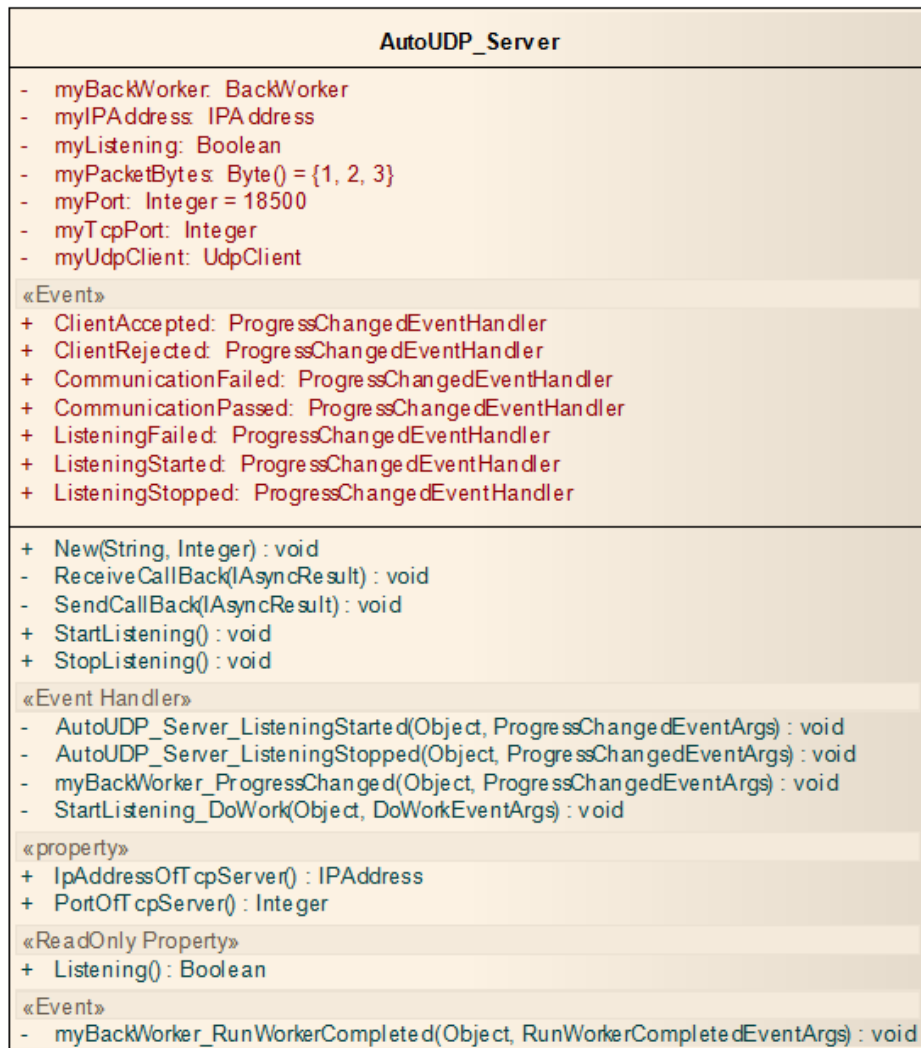
Implementace samotného přenosu dat je realizována přenosem textové hodnoty, která v sobě umožňuje zakódovat libovolný datový obsah. Tento způsob je volen především z důvodu snadného monitorování implementace i pro relativně nezkušené programátory. Vzhledem k realizaci knihovny jako open source je možné ji kdykoliv doplnit podporou přenosu dat v libovolném formátu.

#### 4.1.2. Diagram tříd

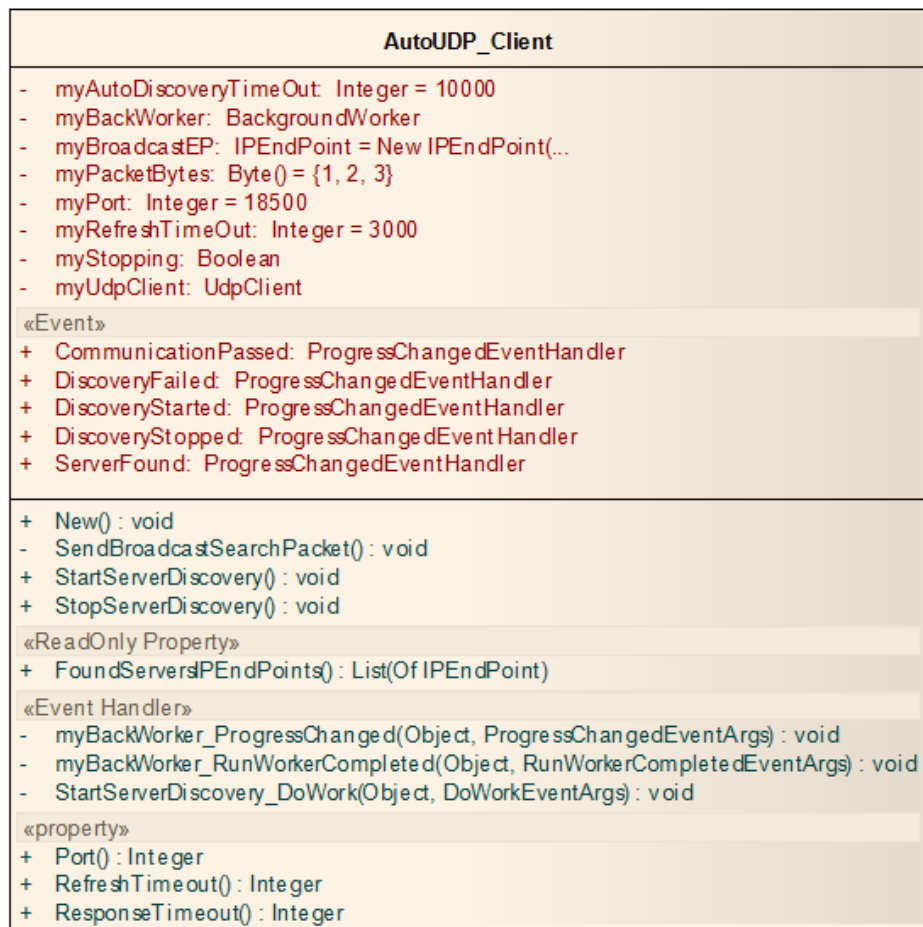
Diagram tříd je z důvodu přehlednosti a zároveň úplnosti rozdělen do pěti částí, které společně postihují strukturu, obsah i vztahy tříd.



Obrázek 1. Diagram tříd knihovny LoxlyLAN - vztahy



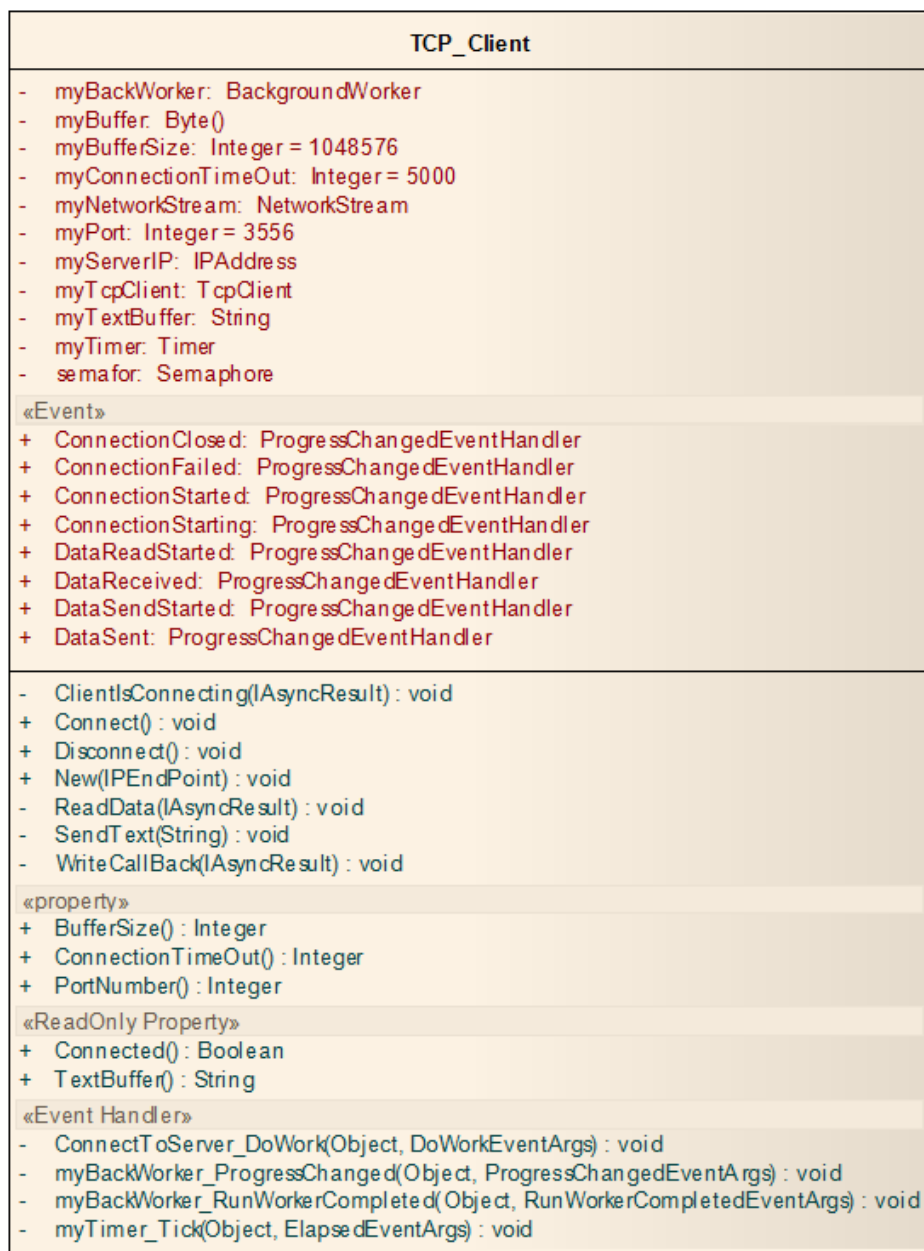
Obrázek 2. Diagram tříd knihovny LoxlyLAN - AutoUDP\_Server



Obrázek 3. Diagram tříd knihovny LoxlyLAN - AutoUDP\_Client



Obrázek 4. Diagram tříd knihovny LoxlyLAN - TCP\_Server



Obrázek 5. Diagram tříd knihovny LoxlyLAN - TCP\_Client



### 4.1.3. Popis tříd

**AutoUDP\_Server** – Tato třída zapouzdřuje atributy a metody potřebné k implementaci UDP serveru. Veškeré vnitřní metody jsou realizovány tak, aby hlavní vlákno aplikace, které vytvoří instanci této třídy, nebylo v žádném případě blokováno. Veškerá komunikace mezi vlákny probíhá pomocí událostí, které jsou v této třídě implementovány. Server slouží pouze k autentizaci připojovaného klienta a k odeslání informací o běžícím TCP serveru, na který se může klient připojit pro realizaci komunikace.

Důležité metody třídy `AutoUDP_Server`:

- `StartListening()` – metoda slouží k vytvoření nového objektu typu `BackgroundWorker`, který umožňuje asynchronní provádění programového kódu, a k jeho asynchronnímu spuštění.
- `StartListening_DoWork(Object, DoWorkEventArgs)` – metoda slouží k obsluze události `DoWork` objektu typu `BackgroundWorker` vzniklého při volání metody `StartListening` a je provedena ve chvíli spuštění asynchronního procesu objektu typu `BackgroundWorker`. V těle metody je volána asynchronní metoda `BeginReceive` objektu typu `UdpClient` a tím je spuštěno poslouchání na nastaveném UDP portu na síti. Dále je zajištěno cyklické volání spuštění poslouchání v případě, že dojde k přijetí dat na daném portu, čímž je zajištěno nepřetržité poslouchání na daném portu.
- `ReceiveCallback(IAsyncResult)` – metoda slouží k obsluze asynchronního přijetí dat na portu, na kterém bylo zahájeno poslouchání. V případě přijetí autentizační sekvence bytů je zpět klientovi odeslána informace obsahující IP adresu a TCP port, na kterém je možné se připojit k TCP serveru a realizovat libovolnou komunikaci.
- `myBackWorker_ProgressChanged(Object, ProgressChangedEventArgs)` – metoda slouží k obsluze události `ProgressChanged`, objektu typu `BackgroundWorker` vzniklého při volání metody `StartListening`, a je provedena vždy při volání metody `ReportProgress` objektu typu `BackgroundWorker`. Na základě hodnot parametrů této metody jsou generovány příslušné události definované v této třídě a tím je umožněno řídit programové aktivity v hlavním vlákne, ve kterém byla vytvořena instance této třídy.

Metoda `ReportProgress` objektu typu `BackgroundWorker`, vzniklého při volání metody `StartListening`, je volána průběžně ve všech metodách podílejících se na asynchronní síťové komunikaci na protokolu UDP a tím je zajištěna efektivní komunikace s hlavním vlákne, ve kterém byla vytvořena instance této třídy.

**AutoUDP\_Client** – Tato třída zapouzdřuje atributy a metody potřebné k implementaci UDP klienta. Veškeré vnitřní metody jsou realizovány tak, aby hlavní vlákno aplikace, které vytvoří instanci této třídy, nebylo v žádném případě blokováno. Veškerá komunikace mezi vlákny je realizovaná pomocí událostí, které jsou v této třídě implementovány. Klient slouží pouze k autentizaci vůči serveru třídy **AutoUDP\_Server** a k přijetí informací o běžícím TCP serveru, na který se může klient připojit pro realizaci komunikace.

Důležité metody třídy **AutoUDP\_Client**:

- **StartServerDiscovery()** - metoda slouží k vytvoření nového objektu typu **BackgroundWorker**, který umožňuje asynchronní provádění programového kódu, a k jeho asynchronnímu spuštění.
- **SendBroadcastSearchPacket()** - metoda slouží k synchronnímu odeslání autentizačního broadcast packetu po lokální síti a tím umožňuje vyvolat reakci instance třídy **AutoUDP\_Server**, pokud je na lokální síti nějaká aktivní. V případě přijetí potvrzujících dat ze serveru jsou nastaveny atributy třídy umožňující v hlavním vlákně, ve kterém byla vytvořena instance této třídy, zjistit IP adresu a TCP port, na kterém poslouchá TCP server.
- **StartServerDiscovery\_DoWork(Object, DoWorkEventArgs)** - metoda slouží k obsluze události **DoWork** objektu typu **BackgroundWorker** vzniklého při volání metody **StartServerDiscovery** a je provedena ve chvíli spuštění asynchronního procesu objektu typu **BackgroundWorker**. V těle metody je cyklicky volána metoda **SendBroadcastSearchPacket**, čímž je zajištěno neustálé hledání UDP serveru v intervalech určených hodnotami atributů **ResponseTimeout** a **RefreshTimeout**.
- **myBackWorker\_ProgressChanged(Object, ProgressChangedEventArgs)** - metoda slouží k obsluze události **ProgressChanged** objektu typu **BackgroundWorker** vzniklého při volání metody **StartServerDiscovery** a je provedena vždy při volání metody **ReportProgress** objektu typu **BackgroundWorker**. Na základě hodnot parametrů této metody jsou generovány příslušné události definované v této třídě a tím je umožněno řídit programové aktivity v hlavním vlákně, ve kterém byla vytvořena instance této třídy.

Metoda **ReportProgress** objektu typu **BackgroundWorker**, vzniklého při volání metody **StartServerDiscovery**, je volána průběžně ve všech metodách podílejících se na síťové komunikaci na protokolu UDP a tím je zajištěna efektivní komunikace s hlavním vlákem, ve kterém byla vytvořena instance této třídy.

**TCP\_Server** – Tato třída zapouzdřuje atributy a metody potřebné k implementaci asynchronní komunikace na TCP protokolu ze strany serveru. Veškeré vnitřní metody jsou realizovány tak, aby hlavní vlákno aplikace, které vytvoří instanci této třídy, nebylo v žádném případě blokováno. Veškerá komunikace mezi vlákny je realizovaná pomocí událostí, které jsou v této třídě implementovány. Instance této třídy slouží k poskytnutí možnosti navázat TCP spojení a k realizaci komunikace na TCP protokolu za pomoci odesílání textových řetězců.

Důležité metody třídy **TCP\_Server**:

- **StartListening()** - metoda slouží k vytvoření nového objektu typu **BackgroundWorker**, který umožňuje asynchronní provádění programového kódu, a k jeho asynchronnímu spuštění. V této metodě je vytvořen objekt typu **Semaphore** pro synchronizaci činnosti paralelních programových vláken.
- **StartListening\_DoWork(Object, DoWorkEventArgs)** - metoda slouží k obsluze události **DoWork** objektu typu **BackgroundWorker** vzniklého při volání metody **StartListening** a je provedena ve chvíli spuštění asynchronního procesu objektu typu **BackgroundWorker**. V těle metody je volána asynchronní metoda **BeginAcceptTcpClient** objektu typu **TcpListener** a tím je spuštěno poslouchání na nastaveném TCP portu na síti. Dále je pomocí objektu typu **Semaphore** blokováno vlákno vzniklé asynchronním spuštěním objektu typu **BackgroundWorker**. Tím je zajištěno, že může probíhat asynchronní komunikace na TCP protokolu, aniž by došlo k ukončení vlákna, ve kterém objekty realizující tuto komunikaci zanikly.
- **Close()** - metoda zastaví všechny spuštěné asynchronní procesy včetně blokovaného vlákna vzniklého asynchronním spuštěním objektu typu **BackgroundWorker**.
- **ClientIsConnecting(IAsyncResult)** - metoda slouží k obsluze asynchronního přijetí spojení s klientem na protokolu TCP. Spouští asynchronní metodu **BeginRead** objektu typu **TcpClient**, která umožňuje přijímat data od připojeného klienta přes TCP protokol.
- **ReadCallback(IAsyncResult)** - metoda slouží k obsluze asynchronního přijetí dat od klienta pomocí TCP protokolu a znovu spouští asynchronní metodu **BeginRead** objektu typu **TcpClient**. Kontroluje případnou ztrátu spojení s klientem a umožňuje automatické spuštění asynchronního připojení.
- **SendText(String)** - metoda slouží k asynchronnímu odeslání dat ke klientovi za použití TCP protokolu.

- `myBackWorker_ProgressChanged(Object,ProgressChangedEventArgs)` - metoda slouží k obsluze události `ProgressChanged` objektu typu `BackgroundWorker` vzniklého při volání metody `StartListening` a je provedena vždy při volání metody `ReportProgress` objektu typu `BackgroundWorker`. Na základě hodnot parametrů této metody jsou generovány příslušné události definované v této třídě a tím je umožněno řídit programové aktivity v hlavním vlákne, ve kterém byla vytvořena instance této třídy.

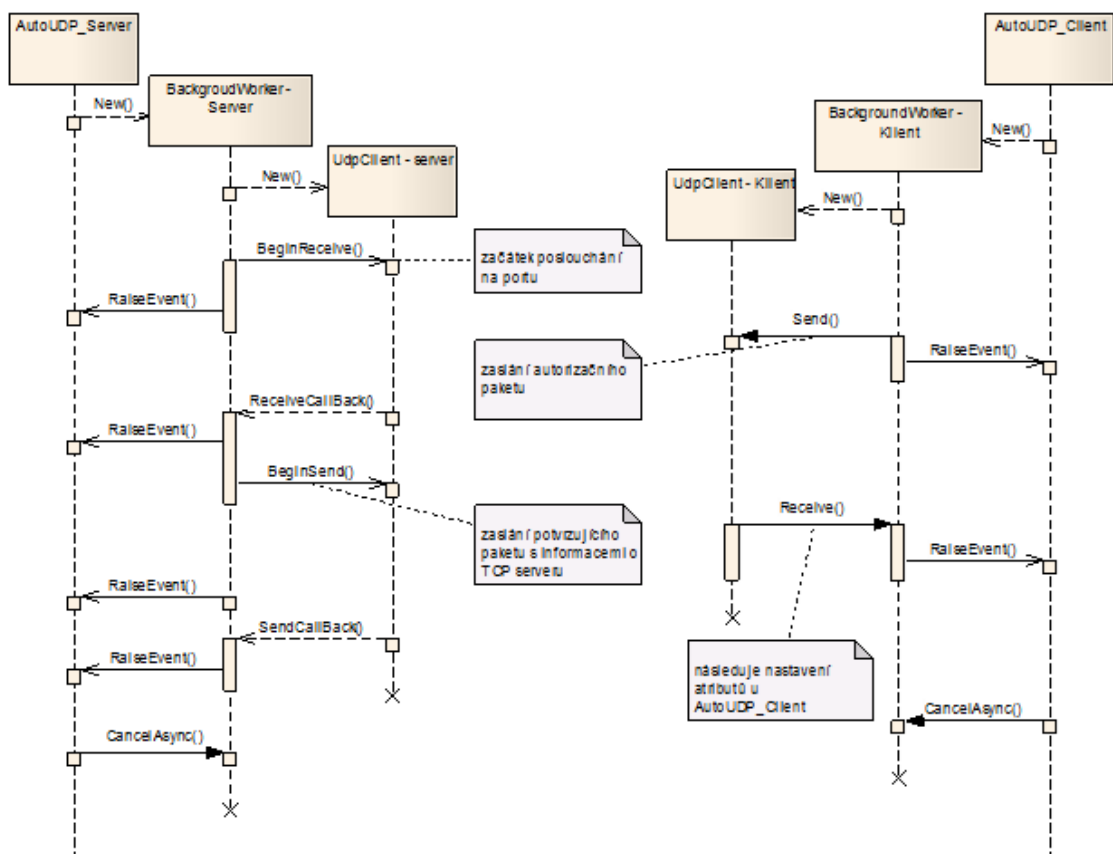
Metoda `ReportProgress` objektu typu `BackgroundWorker`, vzniklého při volání metody `StartListening`, je volána průběžně ve všech metodách podílejících se na asynchronní síťové komunikaci na protokolu UDP a tím je zajištěna efektivní komunikace s hlavním vláknem, ve kterém byla vytvořena instance této třídy.

**TCP\_Client** – Tato třída zapouzdřuje atributy a metody potřebné k realizaci asynchronní komunikace na TCP protokolu. Implementace klíčových metod této třídy je principiálně shodná s implementací metod třídy `TCP_Server`. Rozdílem jsou absence metod pro poslouchání na TCP portu a je v této třídě přidán interní timer pro umožnění zastavení asynchronního připojování k serveru po vypršení nastaveného časového limitu.

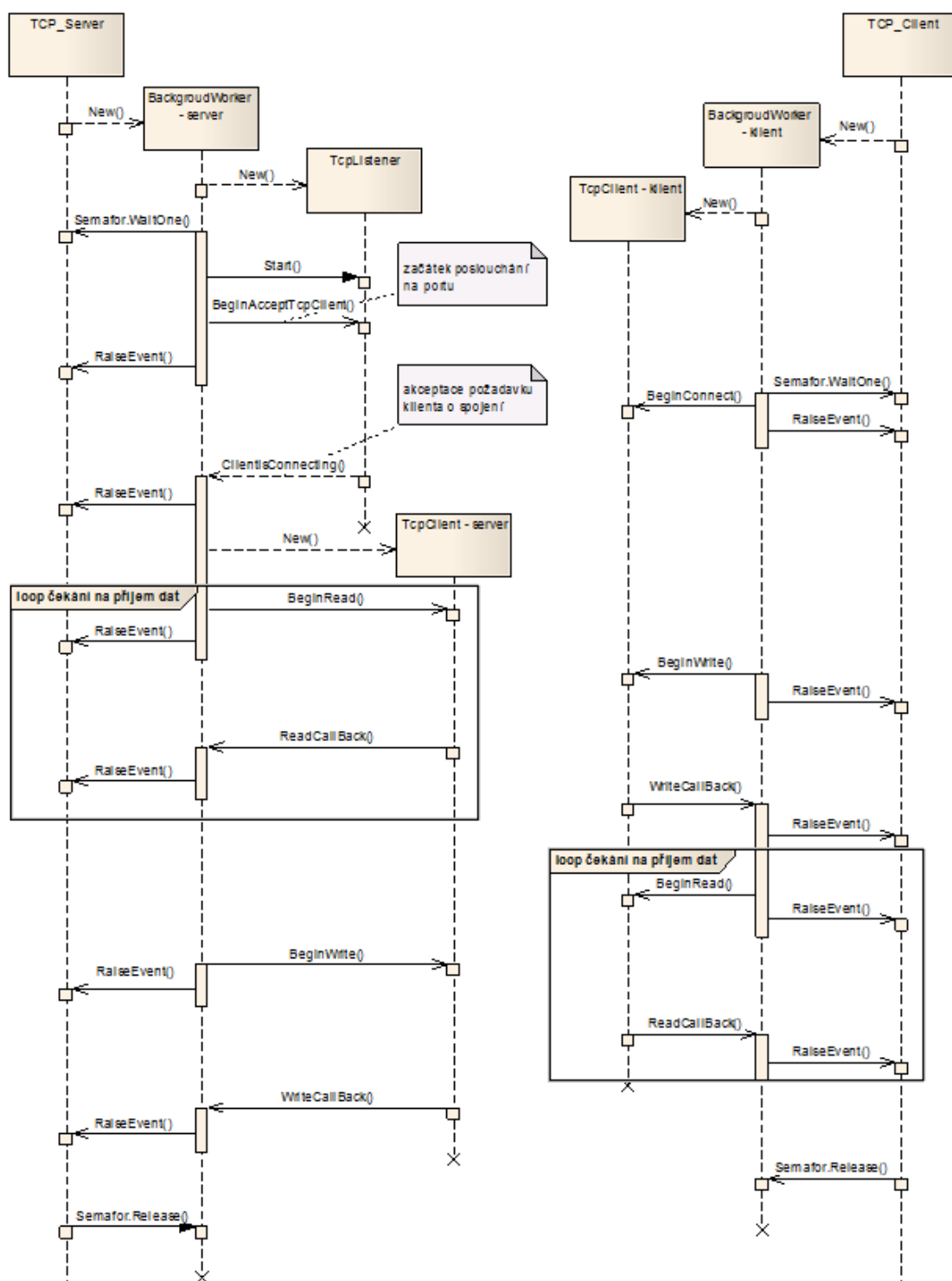
Třídy `TCP_Client` a `TCP_Server` se vlastní implementací liší jen málo a bylo by možné v rámci zefektivnění programového kódu implementovat pouze jednu třídu se změnou atributů, ale vzhledem k realizaci síťového paradigmatu klient-server, jsem dal přednost oddělené implementaci, aby bylo zřejmé, že se z principu jedná o realizace různých funkčních celků.

Stejně jako u tříd `AutoUDP_Server`, `AutoUDP_Client` a `TCP_Server` jsou metody realizovány tak, aby hlavní vlákno aplikace, které vytvoří instanci této třídy, nebylo v žádném případě blokováno. Veškerá komunikace mezi vlákny je realizovaná pomocí událostí, které jsou v této třídě implementovány.

#### 4.1.4. Sekvenční diagram



Obrázek 6. Sekvenční diagram funkce AutoUDP\_Server a AutoUDP\_Client



Obrázek 7. Sekvenční diagram funkce TCP\_Server a TCP\_Client

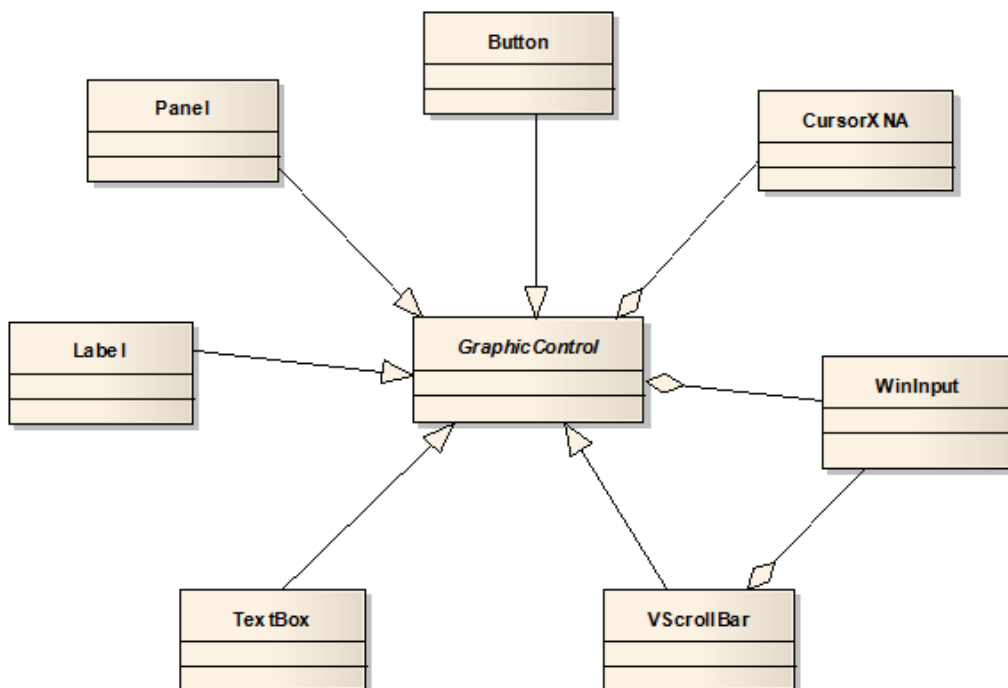
## 4.2. Knihovna pro tvorbu uživatelského rozhraní

### 4.2.1. Popis funkčnosti

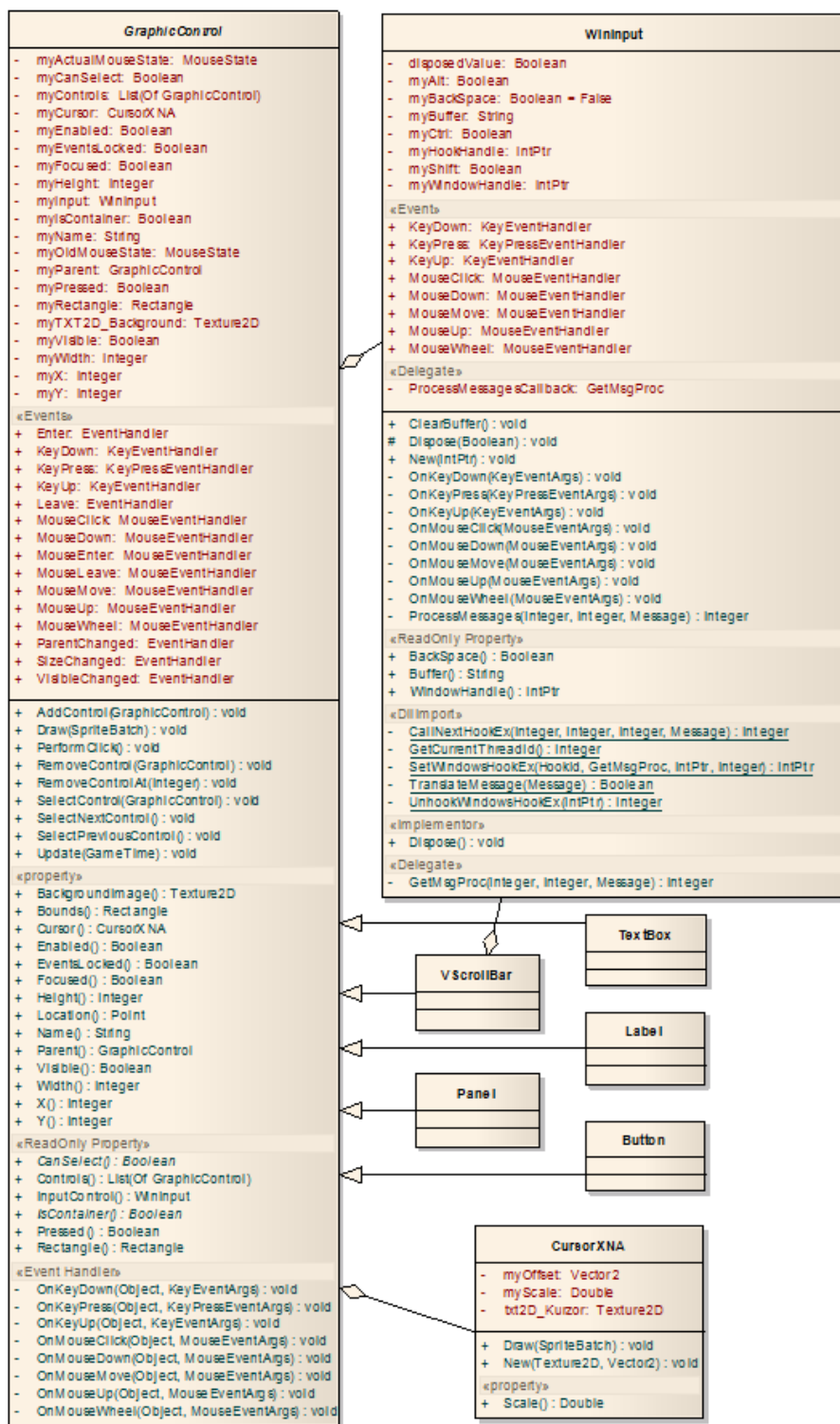
Tato knihovna realizuje základní ovládací prvky, které jsou běžně používány při tvorbě formulářových aplikací pro operační systém Windows, pro použití při implementaci aplikací na platformě Microsoft XNA verze 4.0. V rámci této knihovny je obsažena třída `WinInput`, která realizuje generování událostí pro ostatní třídy na základě rozboru vnitřních zpráv operačního systému Windows [6]. Knihovna je vytvořena s možností efektivně vytvářet další ovládací prvky.

### 4.2.2. Diagram tříd

Diagram tříd je z důvodu přehlednosti a zároveň úplnosti rozdělen do tří variant, které společně postihují strukturu, obsah i vztahy tříd.

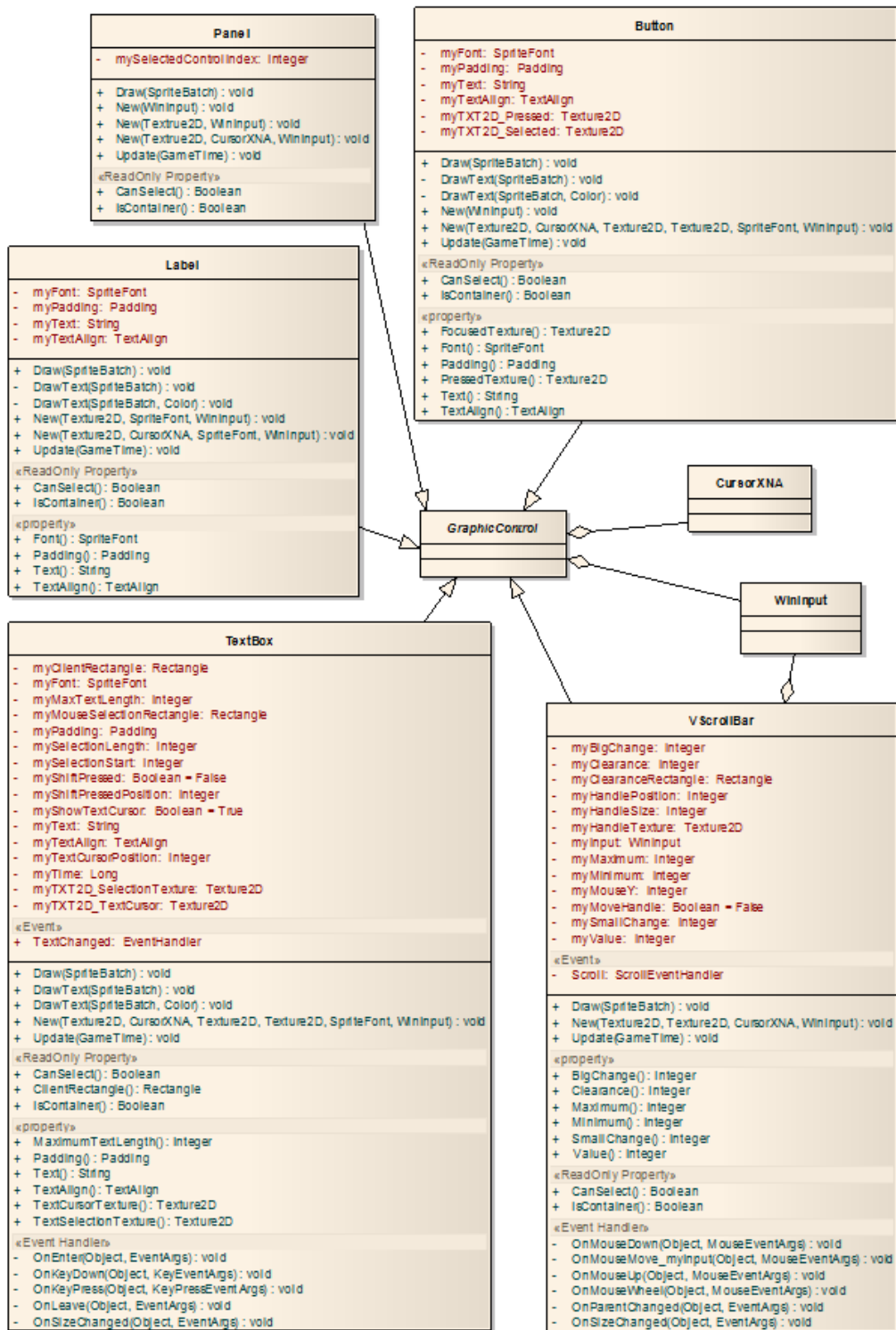


Obrázek 8. Diagram tříd knihovny LoxlyXNAControl varianta 1



Obrázek 9. Diagram tříd knihovny LoxlyXNAControl varianta 2





Obrázek 10. Diagram tříd knihovny LoxlyXNAControl varianta 3

### 4.2.3. Popis tříd

**WinInput** – Tato třída zajišťuje generování událostí myši a klávesnice pro ovládací prvky v této knihovně. Ve třídě jsou importovány funkce z knihoven `user32.dll` a `kernel32.dll`, které umožňují implementovat rozpoznání a identifikaci aktivity vstupních zařízení z vnitřních systémových zpráv operačního systému Windows. Třída umožňuje nezávisle na časování aplikace implementované na platformě Microsoft XNA zachycovat efektivně vstupní data z klávesnice a zveřejňuje je pomocí vlastních atributů (`Buffer`, `Backspace`).

Důležité metody třídy `WinInput`:

- `ProcessMessages(Integer, Integer, Message) As Integer` - realizuje vlastní identifikaci aktivit myši a klávesnice a volá další metody upravující vnitřní atributy této knihovny podle aktuální situace (stisk klávesy `Backspace`, `Ctrl`, apod.) a vyvolávající samotné události (`MouseUp`, `KeyDown`, atd.).

**GraphicControl** – Tato třída je abstraktní a je rodičem pro všechny ostatní ovládací prvky v této knihovně. Obsahuje společné atributy ovládacích prvků a definuje společné události a metody. Všechny ovládací prvky, které jsou potomky této třídy, jsou určeny pro implementaci na platformě Microsoft XNA 4.0.

**CursorXNA** – Tato třída slouží pro implementaci vlastního ukazatele myši pro ovládací prvky této knihovny. Umožňuje využít libovolnou kompilovanou texturu jako zdroj grafiky ukazatele myši. Cílem je pouze zapouzdřit některé detaily implementace ukazatel pro Microsoft XNA 4.0 (přiblížení - velikost, offset - posunutí grafiky vůči reálným souřadnicím ukazatele myši v operačním systému).

**Panel** – Tato třída realizuje ovládací prvek, který stejně jako u formulářových aplikací slouží primárně tomu, aby obsahoval další ovládací prvky a umožnil jejich logické sdružování, případně společné programové ovládání. Metody implementující možnosti ukládání a přepínání ovládacích prvků v panelu jsou vytvořeny již v třídě `GraphicControl`.

**Button** – Třída `Button` realizuje ovládací prvek tlačítka, který slouží jako jedna z hlavních součástí grafického uživatelského rozhraní. Třída umožňuje použití libovolné kompilované textury pro vytvoření vzhledu tlačítka a umožňuje použití libovolného kompilovaného grafického fontu pro platformu Microsoft XNA 4.0.

**Label** – Tato třída implementuje další ze základních ovládacích prvků pro tvorbu grafického uživatelského rozhraní, který, na rozdíl od ovládacího prvku Button, nemůže být vybrán uživatelem a primárně slouží pro tvorbu vzhledu uživatelského rozhraní

**TextBox** – Třída TextBox implementuje ovládací prvek grafického uživatelského rozhraní, který slouží pro zadávání textových vstupů z klávesnice. Jedná se pouze o zjednodušenou variantu bez automatického zalamování textu.

**VScrollBar** – Tato třída implementuje ovládací prvek vertikální posuvník, který může sloužit ke změně hodnot proměnných nebo k posunu ovládacích prvků v uživatelském rozhraní.

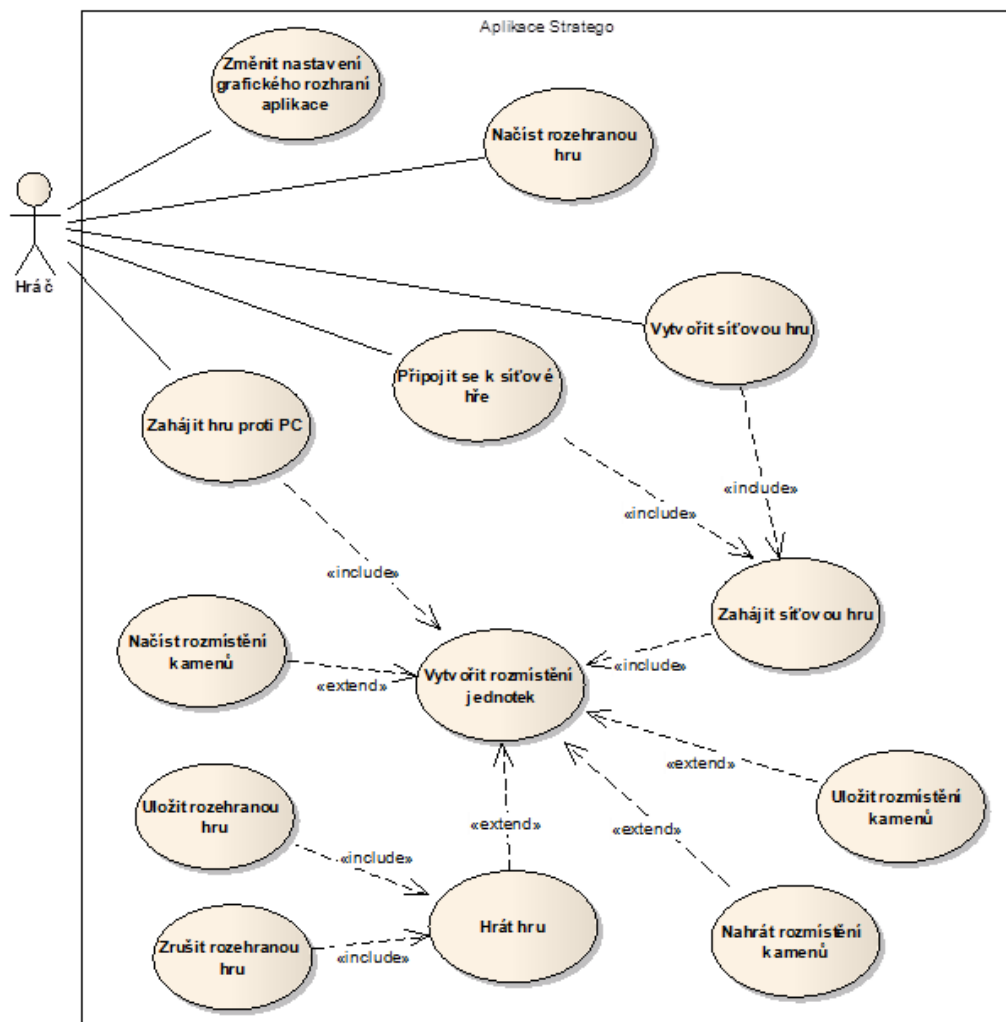
Tato knihovna je vyvíjena jako open source a je možné, aby byly kdykoliv implementovány další ovládací prvky, na které jsou programátoři zvyklí z formulářových aplikací. Je také možné dále rozšiřovat atributy a metody již implementovaných tříd a optimalizovat a rozšířit jejich funkčnost.

## 4.3. Aplikace Stratego

### 4.3.1. Popis funkčnosti aplikace

Základní funkčností aplikace je poskytnutí možnosti hrát deskovou hru Stratego proti počítači nebo přes lokální počítačovou síť proti člověku. Aplikace umožňuje uložit libovolnou hru v libovolné fázi a případně nahrát uloženou hru a pokračovat v jejím dokončení. Je možné přepínat mezi zobrazením přes celou obrazovku a zobrazením v okně. Dále je možné zobrazit si pravidla hry případně nápovědu k ovládání programu.

### 4.3.2. Use Case model



Obrázek 11. Use Case diagram pro hráče

## Případ užití: Zahájit hru proti PC

### 1. Stručný popis

Tento případ užití umožní hráči vytvořit či načíst a upravit rozmístění kamenů a spustit hru proti počítači.

### 2. Vstupní podmínky

Nejsou žádné vstupní podmínky pro tento případ užití.

### 3. Tok událostí

- Základní tok

- **Spuštění hry proti PC s vlastním rozmístěním kamenů**

- Hráč vybere v panelu menu položku „Hra“.
- V dalším menu „Vyber typ hry“ vybere položku „Pro jednoho hráče“.
- V menu „Hra pro jednoho hráče“ vybere položku „Začít novou hru“.
- Proběhne náhodné vybrání barvy pro hráče a hráč postupným klikáním myši na kameny v zásobníku a do volného prostoru ve svojí části hrací desky rozmístí všech 40 hracích kamenů.
- Kameny, které jsou již na hrací ploše, může také kliknutím vybírat a přesouvat nebo vracet do zásobníku.
- Po finálním rozmístění kamenů stiskne tlačítko „Zahájit hru“, které je viditelné pouze v případě umístění všech 40 kamenů na hrací ploše.
- Hra je zahájena — jako první táhne červený.

- Alternativní toky

- **Spuštění hry proti PC s generováním rozmístění kamenů**

- Kroky (a) až (c) jsou stejné jako u základního toku.
- Hráč klikne na tlačítko „Vygenerovat rozmístění“.
  - Dojde k automatickému nahrání rozestavení všech 40 kamenů na hrací plochu z databáze rozestavení.
  - Vygenerované rozmístění je možné měnit stejným způsobem jako při rozmisťování kamenů.
  - Nahrané rozmístění může hráč ještě dále upravovat tažením jednotlivých kamenů.
  - Generovat rozestavení je možné i opakovaně kliknutím na tlačítko „Vygenerovat rozmístění“.

- (i) Po finálním rozmístění kamenů stiskne tlačítko „Zahájit hru“, které je viditelné pouze v případě umístění všech 40 kamenů na hrací ploše.
- (j) Hra je zahájena — jako první táhne červený.
- **Uložení během hry**  
Tento alternativní tok rozšiřuje všechny předchozí toky.
  - (a) Během hry hráč stiskne tlačítko „Uložit hru“ ve spodním panelu tlačítek.
  - (b) V dialogu pro uložení hry hráč zadá název ukládané hry a stiskne tlačítko „Uložit“.
  - (c) Hra se uloží a pokračuje dál.
- **Zrušení hry s uložením**  
Tento alternativní tok rozšiřuje hlavní tok a první alternativní tok.
  - (a) Během hry hráč stiskne tlačítko „Zrušit hru“ ve spodním panelu tlačítek.
  - (b) V dialogu pro zrušení hry klikne hráč na tlačítko „Ano“ pro potvrzení uložení před zrušením.
  - (c) V dialogu pro uložení hry hráč zadá název ukládané hry a stiskne tlačítko „Uložit“.
  - (d) Hra se uloží a zruší, hráč je vrácen do úvodního menu.
- **Zrušení hry bez uložení**  
Tento alternativní tok rozšiřuje hlavní tok a první alternativní tok.
  - (a) Během hry hráč stiskne tlačítko „Zrušit hru“ ve spodním panelu tlačítek.
  - (b) V dialogu pro zrušení hry klikne hráč na tlačítko „Ne“ pro potvrzení zrušení hry bez uložení.
  - (c) Hra se zruší a hráč je vrácen do úvodního menu.

#### 4. Výstupní podmínky

Nejsou žádné výstupní podmínky svázané s tímto případem užití.

#### 5. Body rozšíření

Nejsou žádné body rozšíření svázané s tímto případem užití.

### **Případ užití: Načíst uloženou hru proti PC**

#### 1. Stručný popis

Tento případ užití umožní hráči načíst dříve uloženou hru proti počítači nezávisle na tom, jestli byla uložena ve fázi rozestavování herních kamenů nebo při samotné hře.

## 2. Vstupní podmínky

Nejsou žádné vstupní podmínky pro tento případ užití.

## 3. Tok událostí

- Základní tok

- **Načtení uložené hry proti PC**

- (a) Hráč vybere v panelu menu položku „Hra“.
- (b) V dalším menu „Vyber typ hry“ vybere položku „Pro jednoho hráče“.
- (c) V menu „Hra pro jednoho hráče“ vybere položku „Načíst uloženou hru“.
- (d) Uložená hra je načtena a pokračuje se v ní až už ve fázi roze-  
stavování kamenů nebo ve fázi probíhající hry.

## 4. Výstupní podmínky

Nejsou žádné výstupní podmínky svázané s tímto případem užití.

## 5. Body rozšíření

Nejsou žádné body rozšíření svázané s tímto případem užití.

### **Případ užití: Vytvořit a zahájit síťovou hru**

#### 1. Stručný popis

Tento případ užití umožní hráči vytvořit síťovou hru jako server a v případě připojení protihráče jako klienta vytvořit či načíst a upravit rozmístění kamenů a spustit hru proti protihráči přes lokální počítačovou síť.

#### 2. Vstupní podmínky

Existuje funkční připojení počítače, na kterém je aplikace instalována, k lokální síti podporující protokol TCP/IP.

Ke stejné lokální síti je připojen další počítač, na kterém je aplikace instalována, se zájmem připojit se k vytvořené hře.

#### 3. Tok událostí

- Základní tok

- **Vytvoření síťové hry a zahájení po připojení klienta**

- (a) Hráč vybere v panelu menu položku „Hra“.
- (b) V dalším menu „Vyber typ hry“ vybere položku „Pro dva hráče - LAN“.

- (c) V menu „Hra pro dva hráče - LAN“ vybere položku „Vytvořit novou hru“.
- (d) Proběhne vytvoření serveru, který čeká na připojení klienta.
- (e) V případě připojení klienta hra pokračuje rozestavováním kamenů a zahájením hry - tyto kroky jsou shodné s hlavním tokem a prvním alternativním tokem případu užití „Zahájit hru proti PC“ od kroku (d), s rozdílem, že hra je zahájena až její zahájení potvrdí obě strany.

- Alternativní toky

Alternativní toky jsou shodné s 2., 3. a 4. alternativním tokem případu užití **Zahájit hru proti PC**.

4. Výstupní podmínky

Nejsou žádné výstupní podmínky svázané s tímto případem užití.

5. Body rozšíření

Nejsou žádné body rozšíření svázané s tímto případem užití.

### **Případ užití: Načíst uloženou síťovou hru**

1. Stručný popis

Tento případ užití umožní hráči načíst dříve uloženou hru hranou proti protihráči přes lokální počítačovou síť.

2. Vstupní podmínky

Existuje funkční připojení počítače, na kterém je aplikace instalována, k lokální síti podporující protokol TCP/IP.

Ke stejné lokální síti je připojen další počítač, na kterém je aplikace instalována, se zájmem připojit se k vytvořené hře.

3. Tok událostí

- Základní tok

- **Načtení uložené síťové hry a pokračování ve hře po připojení klienta**

- (a) Hráč vybere v panelu menu položku „Hra“.
- (b) V dalším menu „Vyber typ hry“ vybere položku „Pro dva hráče - LAN“.
- (c) V menu „ pro dva hráče - LAN“ vybere položku „Načíst uloženou hru“.
- (d) Proběhne vytvoření serveru, který čeká na připojení klienta.



- (e) Po připojení klienta je uložená hra načtena a pokračuje se v ní ať už ve fázi rozestavování kamenů nebo ve fázi probíhající hry na straně serveru i klienta.

#### 4. Výstupní podmínky

Nejsou žádné výstupní podmínky svázané s tímto případem užití.

#### 5. Body rozšíření

Nejsou žádné body rozšíření svázané s tímto případem užití.

### **Případ užití: Připojit se k existující síťové hře a zahájit síťovou hru**

#### 1. Stručný popis

Tento případ užití umožní hráči připojit se k existující síťové hře čekající na klienta, poté vytvořit či načíst a upravit rozmístění kamenů a spustit hru proti protihráči přes lokální počítačovou síť.

#### 2. Vstupní podmínky

Existuje funkční připojení počítače, na kterém je aplikace instalována, k lokální síti podporující protokol TCP/IP.

Ke stejné lokální síti je připojen další počítač, na kterém je aplikace instalována a na tomto počítači je vytvořena síťová hra čekající na připojení klienta.

#### 3. Tok událostí

- Základní tok

- **Připojení síťové hry a zahájení hry**

- (a) Hráč vybere v panelu menu položku „Hra“.
- (b) V dalším menu „Vyber typ hry“ vybere položku „Pro dva hráče - LAN“.
- (c) V menu „Hra pro dva hráče - LAN“ vybere položku „Připojit se“.
- (d) Po krátkém vyhledávání serverů na lokální síti vybere hráč ze seznamu serverů, ke kterému se chce připojit, a klikne na tlačítko „Připojit“.
- (e) V případě připojení klienta hra pokračuje rozestavováním kamenů a zahájením hry - tyto kroky jsou shodné s hlavním tokem a prvním alternativním tokem případu užití „Zahájit hru proti PC“ od kroku (d), s rozdílem, že hra je zahájena až její zahájení potvrdí obě strany.

- Alternativní toky

Alternativní toky jsou shodné s 2., 3. a 4. alternativním tokem případu užití **Zahájit hru proti PC**.

4. Výstupní podmínky

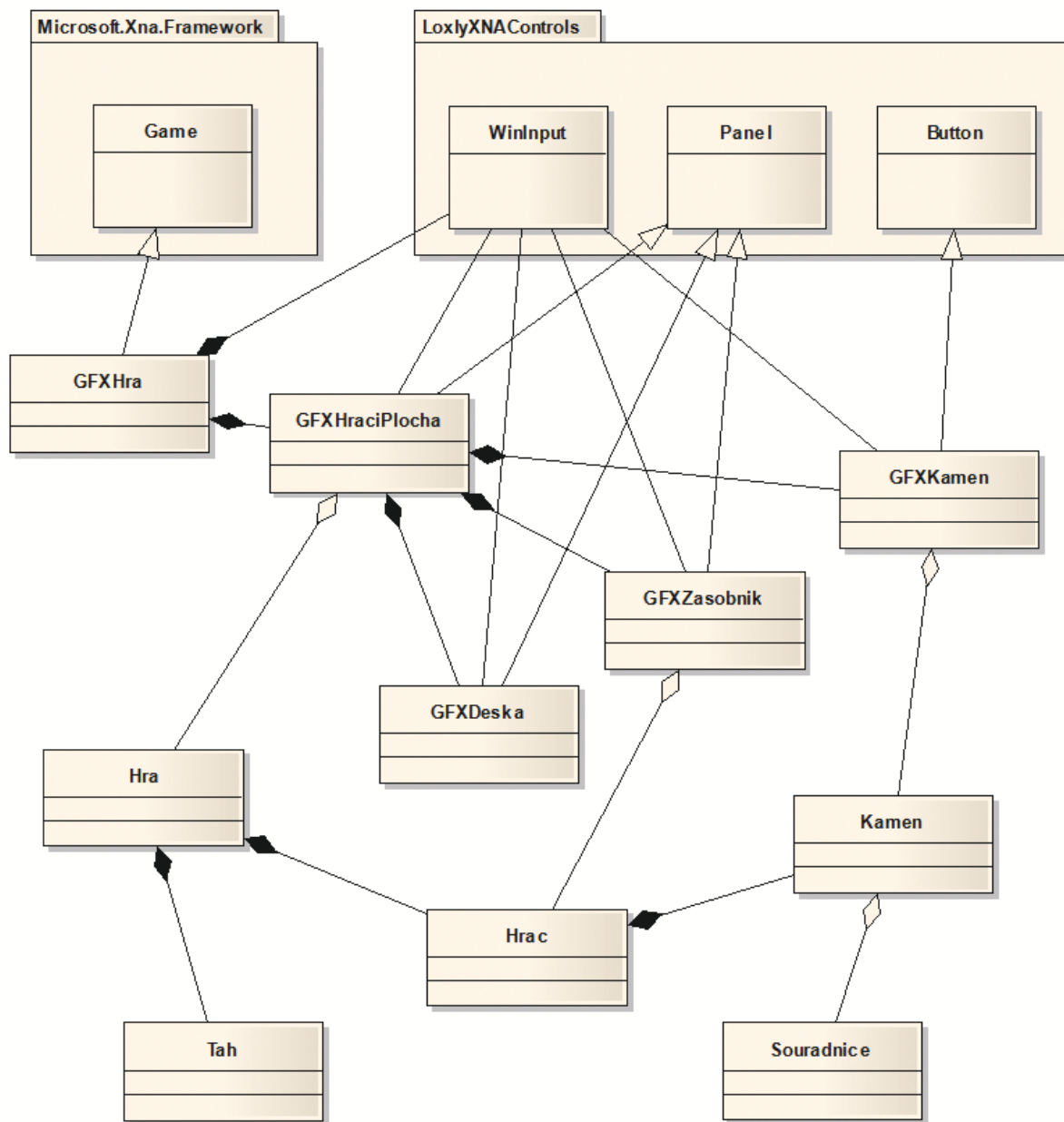
Nejsou žádné výstupní podmínky svázané s tímto případem užití.

5. Body rozšíření

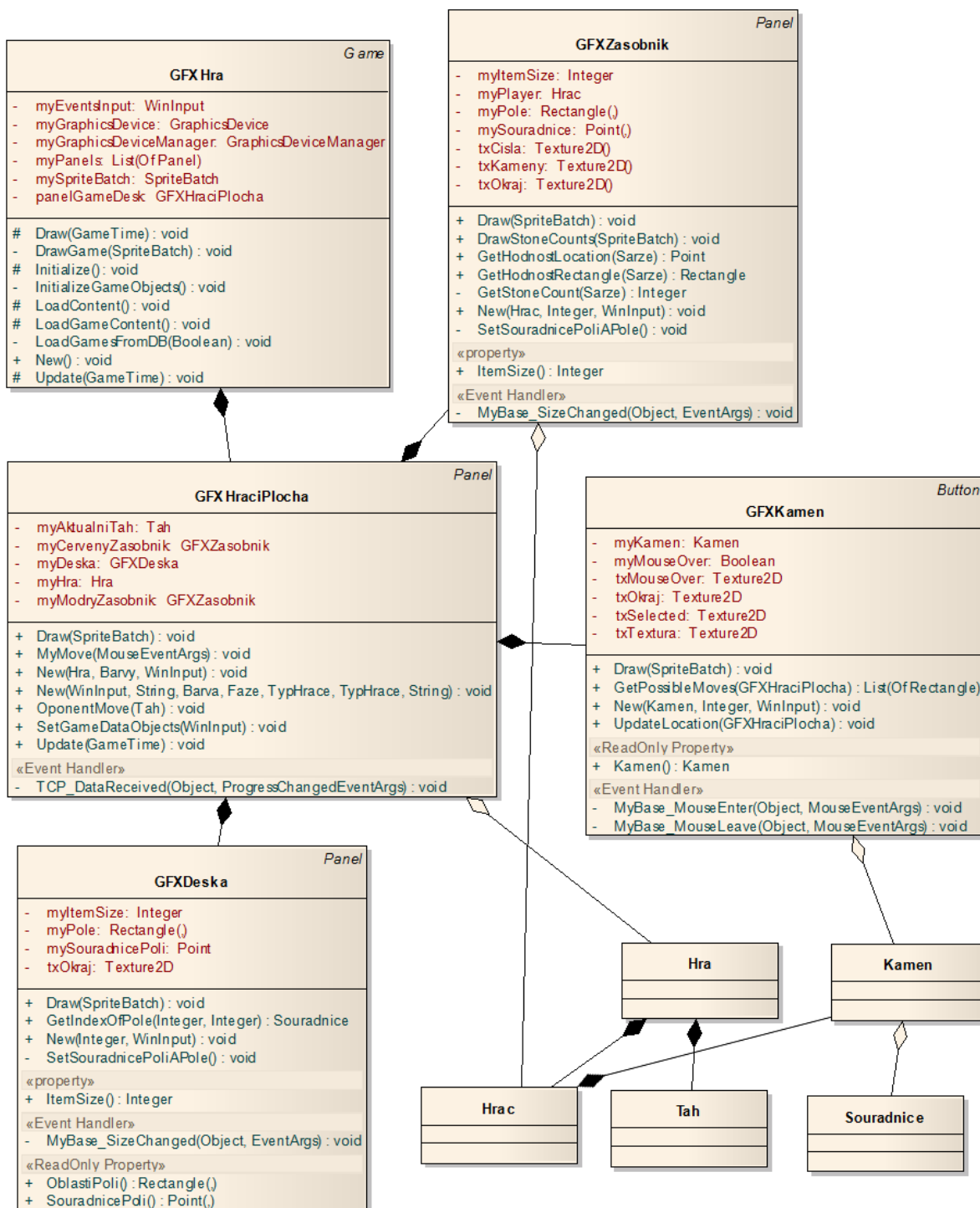
Nejsou žádné body rozšíření svázané s tímto případem užití.

### 4.3.3. Diagram tříd

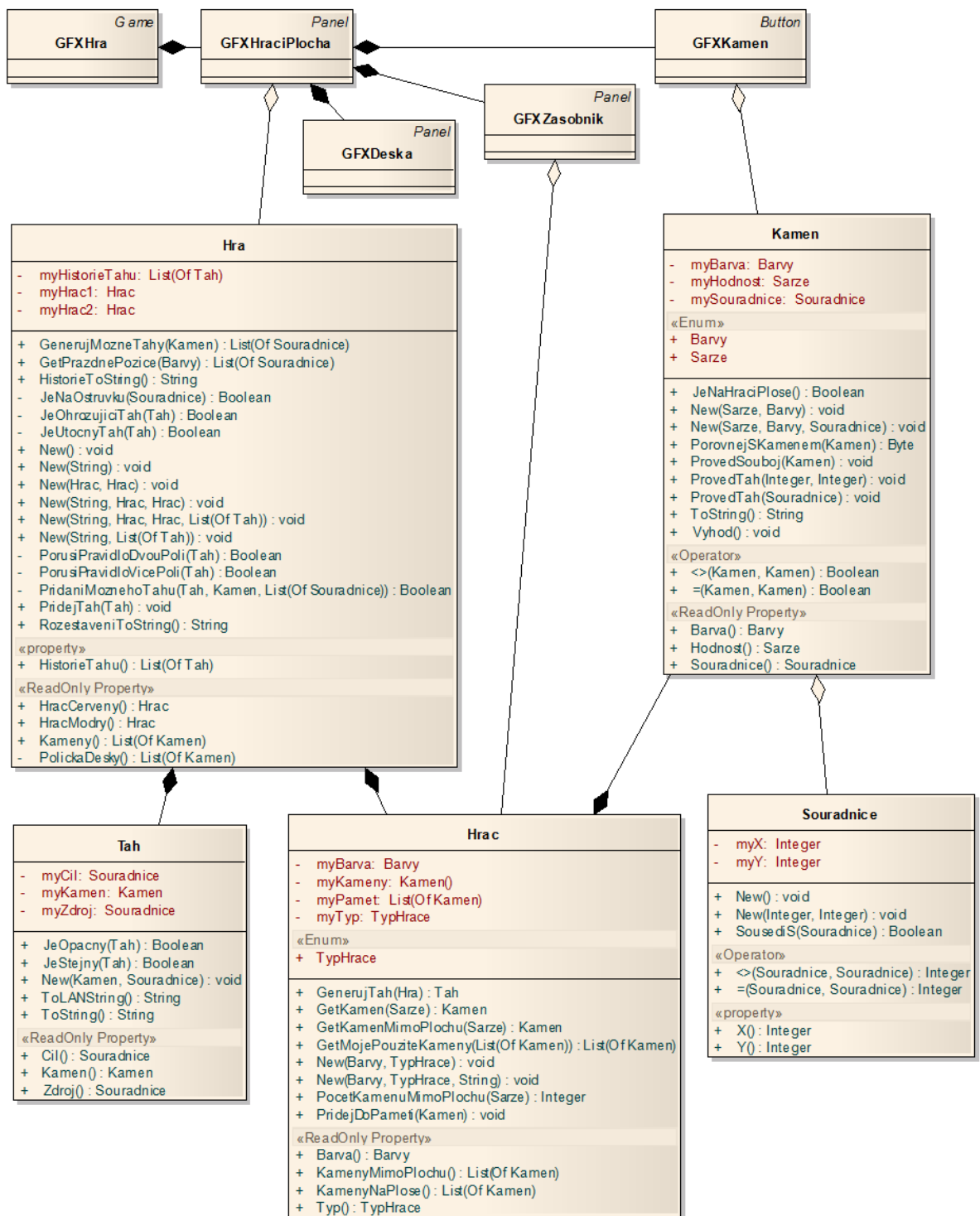
Diagram tříd je z důvodu přehlednosti rozdělen na více částí a třídy `GFXHra` a `GFXHraciPlocha` obsahují pouze klíčové atributy a metody.



Obrázek 12. Diagram tříd aplikace Stratego - varianta 1



Obrázek 13. Diagram tříd aplikace Stratego - varianta 2



Obrázek 14. Diagram tříd aplikace Stratego - varianta 3

#### 4.3.4. Popis tříd

Třídy aplikace Stratego se logicky dělí do dvou skupin. První skupina (třídy `Hra`, `Hrac`, `Kamen`, `Tah` a `Souradnice`) realizuje základní herní principy deskové stolní hry Stratego a druhá skupina (třídy `GFXHra`, `GFXHraciPlocha`, `GFXDeska`, `GFXZasobnik` a `GFXKamen`) implementuje uživatelské rozhraní. Vzhledem k implementaci pomocí technologie `XNA Framework 4.0` je v aplikaci deklarován globální modul, který spouští samotnou hru a jsou v něm deklarovány globálně přístupné objekty pro realizaci síťové komunikace.

**Souradnice** – Tato třída implementuje jednoduché plošné souřadnice, omezené pouze na celá čísla v rozmezí -1 až 9, které umožňují určovat pozice na šachovnici rozměru 10 x 10 polí (hodnoty 0 až 9) a pozici mimo šachovnici (hodnota -1). V případě, že by v aplikaci byla přiřazena hodnota mimo rozmezí povolených herních pozic, je automaticky změněna na hodnotu -1. Ve třídě jsou definovány základní relační operátory (`=`, `<>`), pro snazší práci s instancemi této třídy, a jedna klíčová metoda:

- `SousediS(ByVal souradnice As Souradnice) As Boolean` – metoda hodnotí vzájemnou pozici dvou instancí této třídy na šachovnici.

**Tah** – Třída `Tah` umožňuje vytvářet instance popisující jednotlivé tahy hry Stratego. V rámci každé instance jsou uloženy hodnoty popisující který kámen se posunul z jakých výchozích souřadnic na hrací ploše a na jaké souřadnice.

Důležité metody třídy `Tah`:

- `JeStejny(ByVal tah As Tah) As Boolean` – metoda hodnotí, jsou-li dva tahy provedeny stejným hracím kamenem ze shodných výchozích souřadnic i na shodné cílové souřadnice.
- `JeOpacny(ByVal tah As Tah) As Boolean` – metoda hodnotí, jsou-li dva tahy provedeny stejným hracím kamenem opačným směrem — výchozí souřadnice obou tahů jsou shodné s cílovými souřadnicemi porovnávaného tahu.

Obě zmíněné metody jsou použity především při implementaci „Pravidla dvou polí“ (kapitola 9.4.3.) a „Pravidla více polí“ (kapitola 9.4.4.).

**Kamen** – Tato třída umožňuje vytvářet instance reprezentující jednotlivé herní kameny charakterizované hodnotí, barvou a umístěním na hrací ploše. Pro určení hodnoti a barvy jsou v rámci třídy `Kamen` vytvořeny dva výčtové typy — `Sarze` a `Barvy`. Dále jsou ve třídě implementovány metody, které umožňují provádět herní operace s vytvořenými objekty typu `Kamen`.

Důležité metody třídy `Kamen`:

- `PorovnejSKamenem(ByRef kamen As Kamen) As Byte` – Metoda porovná hodnost dvou instancí třídy `Kamen` a vrací hodnoty 0, 1 a 2 reprezentující všechny potenciální výsledky vzájemného porovnání. Výsledky volání této metody jsou využity při implementaci soubojů herních kamenů v metodě `ProvedSouboj`.
- `Vyhod()` – Tato metoda realizuje přesun herního kamene mimo hrací plochu.
- `ProvedTah(ByVal umistení As Souradnice)` – Metoda realizuje přesun herního kamene po hrací ploše na nové souřadnice.
- `ProvedSouboj(ByRef Kamen As Kamen)` – Tato metoda implementuje souboj dvou herních kamenů v souladu s pravidly hry `Stratego` (kapitola 9.).

**Hrac** – Instance této třídy reprezentují hráče, kteří se účastní hry `Stratego`. Atributy třídy charakterizují hráče podle barvy hracích kamenů a typu hráče. Pro určení typu hráče je v rámci této třídy definován výčtový datový typ `TypHrace`, který umožňuje rozlišit mezi uživatelem, počítačem a protivníkem hrajícím přes počítačovou síť. V případě, je-li instance této třídy hráč - počítač, je implementována paměť pro zapamatování odkrytých soupeřových kamenů, která je využívána pro generování tahu počítačem.

Důležité metody třídy `Hrac`:

- `GenerujTah(ByRef game As Hra) As Tah` – Tato metoda implementuje strategii volby herního tahu v případě, že instance třídy `Hrac` je ovládána počítačem. Samotný algoritmus pro generování tahu je postaven na ohodnocení postavení vlastních kamenů a postavení soupeřových kamenů, přičemž je využito paměti, ve které jsou uloženy všechny odkryté soupeřovy kameny. Tím je dána mírná výhoda počítači, protože neexistuje možnost, že by zapomněl umístění již odkrytého soupeřova kamene. Algoritmus používá ryze útočnou strategii. Zná-li hodnotu nějakého soupeřova kamene, využije relativně přímočarého postupu k vyřazení tohoto kamene ze hry. V případě, že žádné soupeřovy kameny nezná, realizuje průzkum útokem od relativně nejméně cenných herních kamenů (průzkumník, seržant). Při volbě herního kamene vhodného k tahu bere v úvahu, že není výhodné odkrývat pohyblivost mnoha herních kamenů, a volí k útoku spíše kameny, se kterými bylo již taženo.

**Hra** – Tato třída implementuje všechny principy a pravidla hry Stratego. V konstruktoru vznikají instance třídy **Hrac** a v jejich konstruktorech vznikají instance třídy **Kamen**, čímž jsou vytvořeny všechny herní objekty. Třída **Hra** především zapouzdřuje metody implementující dodržování pravidel hry Stratego (kapitola 9.).

Důležité metody třídy **Hra**:

- **GenerujMozneTahy(ByRef kamen As Kamen) As List(Of Souradnice)** – Tato metoda generuje pro libovolný herní kámen seznam volných pozic (instance třídy **Souradnice**), na které může daný herní kámen provést tah v souladu s pravidly hry. Pro implementaci algoritmu využívá pomocných metod (**PridaniMoznehoTahu**, **PorusiPravidloDvouPolii**, **PorusiPravidloVicePolii**, **JeUtocnyTah**, **JeOhrozujiciTah**).
- **GetPrazdnePozice(ByRef barva As Kamen.Barvy) As List(Of Souradnice)** – Tato metoda zjišťuje všechny neobsazené pozice na hrací ploše v oblasti pro výchozí rozestavení herních kamenů. Využívá se pouze ve fázi rozestavování kamenů.

**GFXKamen** – Tato třída je potomkem třídy **Button** z knihovny **LoxlyXNACControls**. Instance třídy **GFXKamen** reprezentují v grafickém uživatelském rozhraní herní kameny na ploše. Svými metodami umožňují vykreslování grafické reprezentace v umístění na ploše odpovídajícím souřadnicím zapouzdřeným v instanci třídy **Kamen**, která je uložena v atributu **Kamen** každé instance třídy **GFXKamen**. Dále umožňuje interakci uživatele s reprezentací herního kamene.

Důležité metody třídy **GFXKamen**:

- **UpdateLocation(ByRef hraciPlocha As GFXHraciPlocha)** – Tato metoda umožňuje změnu pozice grafické reprezentace na základě vyhodnocení nastavení nadřazeného objektu třídy **GFXHraciPlocha** a hodnoty souřadnice instance třídy **Kamen**, která je uložena v atributu **Kamen**.
- **GetPossibleMoves(ByRef hraciPlocha As GFXHraciPlocha) As List(Of Rectangle)** – Na základě výsledku této metody je možné na obrazovku správně vykreslit grafickou reprezentaci možných tahů instance třídy **Kamen**, která je uložena v atributu **Kamen**.

**GFXZasobnik** – Třída je potomkem třídy **Panel** z knihovny **LoxlyXNACControls**. Instance této třídy reprezentují v grafickém uživatelském rozhraní odkladovou plochu pro herní kameny, které jsou mimo plochu (zásobník). Třída umožňuje interakci uživatele s herními kameny, které jsou mimo plochu (pouze ve fázi rozestavování kamenů). Dále také zabezpečuje zobrazení počtu herních kamenů mimo plochu u každé hodnoty.



Důležité metody třídy `GFXZasobnik`:

- `DrawStoneCounts(ByRef SB As SpriteBatch)` – Tato metoda umožňuje zobrazení počtu herních kamenů mimo plochu u každé hodnoty nad instancemi třídy `GFXKamen`, které jsou vykreslovány v oblasti zásobníku.

**GFXDeska** – Třída je potomkem třídy `Panel` z knihovny `LoxlyXNAControls`. Instance této třídy reprezentuje v grafickém uživatelském rozhraní zobrazení hlavní hrací plochy (deska). Třída umožňuje interakci uživatele s volnými pozicemi na desce (cíle jednotlivých neútočných tahů).

Důležité metody třídy `GFXDeska`:

- `GetIndexOfPole(ByVal X As Integer, ByVal Y As Integer) As Souradnice` – Tato metoda umožňuje převést systémové souřadnice myši nad deskou na objekt typu `Souradnice` charakterizující umístění jednotlivých herních polí v samotné hře.

**GFXHraciPlocha** – Tato třída je potomkem třídy `Panel` z knihovny `LoxlyXNAControls`. Tato třída reprezentuje kompletní grafické rozhraní samotné deskové hry Stratego. Implementuje interakci s ostatními herními objekty pomocí jejich událostí a realizuje veškeré zobrazování ovládacích prvků v průběhu od rozestavování kamenů až po zakončení hry. Realizuje také interakci se soupeřovou aplikací připojenou přes lokální počítačovou síť. V rámci této třídy je deklarován výčtový datový typ (`Faze`) umožňující rozlišení jednotlivých fází průběhu hry (atribut `myAktualniFaze`).

Důležité metody třídy `GFXHraciPlocha`:

- `Update(ByVal gameTime As Microsoft.Xna.Framework.GameTime)` – Tato metoda je volána v rámci metody `Update` třídy `GFXHra` a umožňuje realizovat časované operace v rámci průběhu hry. V této metodě jsou tedy realizovány všechny automatické pohyby v grafickém rozhraní i předávání tahů mezi hráči. Je využito faktu, že tato metoda je volána 60krát za sekundu a všechny herní změny jsou vyvolány pouhou změnou atributu `myAktualniFaze`.
- `SetData()` – Tato metoda načítá z databáze seznam rozestavení kamenů, který je použit při generování náhodných rozestavení (obsahuje 7708 různých rozestavení kamenů).
- `MyMove(ByVal e As MouseEventArgs)` – Metoda nastavuje vnitřní atributy třídy `GFXHraciPlocha` způsobem, aby metoda `Update` zajistila vykonání tahu nejen po stránce grafické, ale i v nastavení všech vnitřních atributů dotčených herních objektů. V případě, že je hra síťová, realizuje

odeslání aktuálního tahu soupeři ve formě textových dat, které protistrana zpracuje.

- `OponentMove(ByVal tah As Tah)` – Tato metoda nastavuje vnitřní atributy třídy `GFXHraciPlocha` způsobem, aby metoda `Update` zajistila vykonání tahu nejen po stránce grafické, ale i v nastavení všech vnitřních atributů dotčených herních objektů nezávisle na tom, zda je soupeřův tah generován počítačem nebo je zpracován od protihráče, hrajícího přes lokální počítačovou síť.
- `SetGameDataObjects(ByRef Input As WinInput)` – Tato metoda vytváří všechny ostatní grafické herní objekty (`GFXDeska`, `GFXZasobnik`, `GFXKamen`) a přiřazuje k nim odpovídající herní objekty (`Hrac`, `Kamen`).
- `TCP_DataReceived(sender As Object, e As ProgressChangedEventArgs)` – Metoda zachycující událost přijetí dat přes lokální počítačovou síť. V rámci této metody je zkontrolována správnost přijatých dat a na jejich základě je vytvořen lokálně soupeřův tah a volána metoda `OponentMove`. Díky tomu mají oba síťoví hráči kompletní historii tahů uloženou lokálně a oba mohou případně hru uložit a navázat na ni později.

Všechny uvedené třídy, které jsou potomky z knihovny `LoxlyXNAControls`, obsahují metodu `Draw(ByRef SB As SpriteBatch)`, která je volána z metody `Draw` třídy `GFXHra`. Tato metoda realizuje v každé třídě vykreslování všech textur reprezentujících daný objekt.

**GFXHra** – Tato třída je potomkem třídy `Game` z třídy `Microsoft.Xna.Framework`. Instance této třídy povinně implementuje metody zděděné z třídy `Game`, které umožňují samotný vznik aplikace využívající technologie DirectX 9.0c (metody `Initialize`, `LoadContent`, `Update` a `Draw`). V těchto metodách vznikají objekty, které implementují práci s DirectX (`SpriteBatch`, `GraphicsDevice` a `GraphicsDeviceManager`) a načítají se do paměti kompilované textury a fonty použité při vykreslování herních objektů. Kromě implementace těchto děděných metod jsou v třídě `GFXHra` deklarovány objekty typu `Panel` z knihovny `LoxlyXNAControls`, které reprezentují jednotlivá menu a dialogová okna použitá v aplikaci. Deklarací a kombinací ovládacích prvků z knihovny `LoxlyXNAControls` je vytvořeno celé grafické uživatelské rozhraní aplikace.

Důležité metody třídy `GFXHra`:

- `InitializeGameObjects()` – Tato metoda volá při inicializaci třídy `GFXHra` další metody, v kterých dochází k tvorbě objektů typu `Panel` a jsou do

nich vkládány další ovládací prvky, nastavovány jejich atributy a přiřazovány metody zachycující jejich klíčové události. Tím je vytvořeno v paměti kompletní grafické uživatelské rozhraní celé aplikace.

- `LoadGamesFromDB(ByVal network As Boolean)` – Tato metoda umožňuje načíst uložené hry (z databáze) a zobrazit je v seznamu uložených her, ze kterých si může uživatel vybrat a spustit libovolnou z nich. Metoda je používána na nahrávání uložených lokálních i síťových her.

## 5. Datová základna

### 5.1. Databáze

Pro ukládání herních rozestavění, ukládání her proti počítači i pro ukládání síťových her je použita databáze MS Access. Pro každý typ ukládání dat je vytvořena jednoduchá nezávislá tabulka a mezi jednotlivými tabulkami neexistují žádné vztahy. Databáze je využita čistě jako lineární úložiště dat. Databáze je chráněna pouze jednoduše heslem („t2o0m0i7k“). Z důvodu jednoduchosti datového modelu nejsou vytvořeny žádné diagramy k jeho popisu.

### 5.2. Textury

Všechny textury, které jsou použity v aplikaci, byly kompilovány nezávisle na projektu ve vývojovém prostředí Visual Studio 2010. Ke kompilaci je použita modifikovaná open source aplikace XNA 4.0 Content Compiler [7] zveřejněná s Microsoft Public License [8] licencí.

### 5.3. Fonty

K tvorbě fontů pro platformu Microsoft XNA byla použita volně šiřitelná aplikace SpriteFont 2 [9].

Vzhledem k omezenému množství fontů, které jsou volně použitelné na platformě Microsoft XNA, byl jako výchozí použit volně šiřitelný font Komika-Text [10] vytvořený společností Apostrophic Laboratories [11].

## 6. Instalace aplikace

### 6.1. Systémové požadavky

Procesor	pracovní frekvence 1 GHz a výkonnější
Operační paměť	512 MB a více
Grafická karta	Podporující DirectX 10 a Shader model 3.0+
Pevný disk	minimálně 120 MB pro instalaci
Rozlišení obrazovky	1024 x 768 a více
Operační systém	Windows XP SP3 a vyšší

Tabulka 1. Systémové požadavky

### 6.2. Průběh instalace

1. Ověřte, že váš systém odpovídá systémovým požadavkům pro spuštění aplikace.
2. Pro zahájení instalace spusťte soubor STRATEGOSETUP.EXE.
3. Instalace probíhá pomocí standardního instalačního průvodce, kde můžete zvolit umístění instalovaných souborů a vybrat, zda chcete vytvořit zástupce na ploše.
4. Po spuštění samotné instalace souborů dojde ke kontrole a případné automatické instalaci podpůrných knihoven (.NET FRAMEWORK 4.0 a XNA FRAMEWORK 4.0). Dále se nainstalují soubory aplikace do zvoleného adresáře.
5. Před dokončením instalace můžete zvolit, jestli chcete automaticky spustit nově nainstalovanou aplikaci.

### 6.3. Odstranění aplikace

Pro odinstalování aplikace je možné použít jeden z následujících postupů:

- Ve „Start Menu“ rozbalte položku „Všechny programy“, dále otevřete položku „Stratego“ a klikněte na položku „Odinstalovat aplikaci Stratego“.
- V adresáři, který jste vybrali pro instalaci aplikace spusťte soubor UNINS000.EXE.
- Pomocí systémového nástroje pro přidávání a odebrání aplikací odinstalujte aplikaci Stratego.

Procesem odinstalování dojde k odstranění všech instalovaných součástí samotné aplikace Stratego. Pokud chcete odstranit i .NET FRAMEWORK 4.0 a XNA FRAMEWORK 4.0 (mohou být využívány i dalšími aplikacemi) je třeba je odstranit samostatně pomocí systémového nástroje pro přidávání a odebrání aplikací.

## 7. Bezpečnost

Vzhledem k zaměření aplikace je zabezpečení zajištěno pouze použitím hesla na databázi uložených rozestavění a uložených her. Bylo by možné využít vyšší úrovně zabezpečení, ale při tomto druhu aplikace se jeví jako nadbytečné.

Síťová komunikace probíhá v čistě textové podobě, je tedy možné zachytávat předávaná data mezi aplikacemi a případně využít nashromážděná data k podvádění při síťové hře. Tento nedostatek by bylo možné vyřešit kompletním zašifrováním posílaných dat na úrovni aplikační vrstvy ISO OSI modelu, ale opět se v dané situaci toto opatření jeví jako nadbytečné.

## 8. Testování aplikace

Testování aplikace probíhalo ve dvou fázích:

### 1. Vývoj podpůrných knihoven

Při návrhu řešení síťové komunikace jsem vyzkoušel realizaci komunikace na aplikaci synchronizující textová hlášení, která je částí informačního systému na Vyšší odborné škole ekonomické a zdravotnické a Střední škole, Boskovic. Po odladění této aplikace jsem vyvinul další aplikaci pouze pro testovací účely, která na jednoduchém chatu umožňuje pozorovat veškerou komunikaci a průběh událostí v knihovně. Po odladění chyb v rámci knihovny pro síťovou komunikaci byla knihovna teprve použita ve finální aplikaci. Pro ověření funkčnosti knihovny ovládacích prvků jsem využil jednoduchý projekt na platformě Microsoft XNA 4.0, který byl použit jako ukázka při výuce zájmového programování ve Visual Basicu na střední škole. Po odladění funkčních chyb byla knihovna použita ve finální aplikaci.

### 2. Testování funkčnosti hlavní aplikace

Většina testovacího procesu proběhla už při samotném vývoji a na testování funkčnosti samotné hry se podíleli většinou studenti střední školy ve svém volném čase.



## 9. Pravidla hry

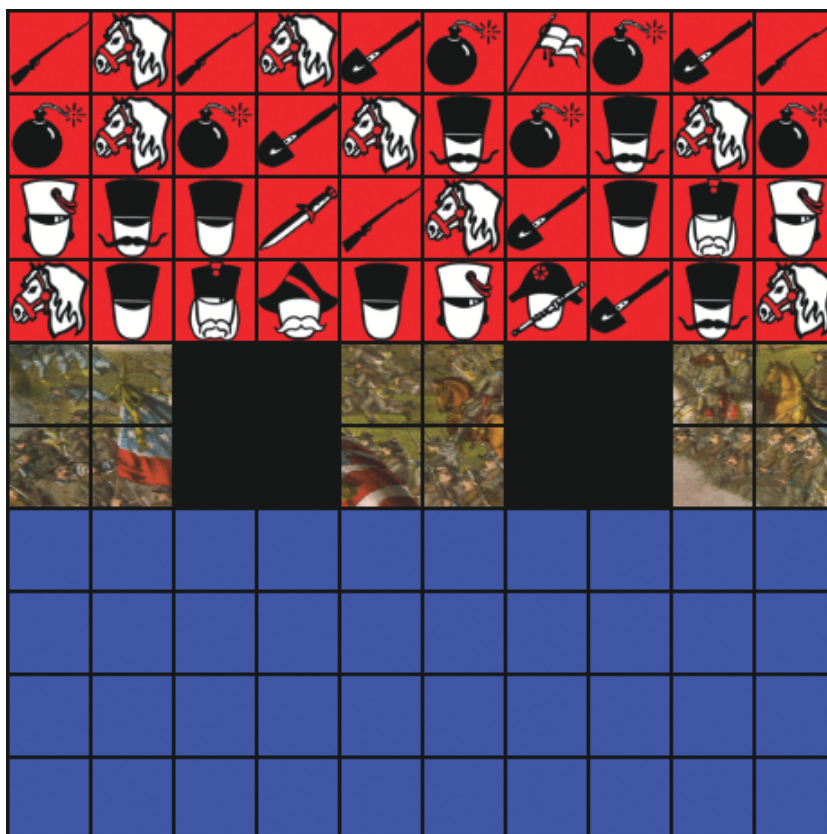
Pravidla vychází z vlastního překladu oficiálních pravidel International Strategic Federation [12] a jsou upraveny pro použití v počítačové aplikaci.

### 9.1. Obecný popis hry

Stratego je desková stolní hra pro dva hráče, ve které je cílem, na základě strategického rozestavení vlastních 40 herních kamenů a strategie postupu proti soupeřovým kamenům, zajmout soupeřovu vlajku (jeden ze čtyřiceti kamenů). Hra probíhá se skrytými hodnotami soupeřových kamenů a tím vzniká velká variabilita použitelných herních strategií vedoucích k vítězství.

### 9.2. Hrací plocha

Hrací plocha je tvořena 10x10 hracími poli se dvěma 2x2 nepřístupnými ostrůvky. Prostory pro výchozí rozestavení kamenů obou hráčů jsou 4x10 polí na dvou stranách hrací plochy se dvěma ostrůvky mezi těmito oblastmi.



Obrázek 15. Hrací plocha s rozestavenými kameny

### 9.3. Hrací kameny

Každý hráč má k dispozici 40 herních kamenů (jeden červené a druhý modré), které jsou rozděleny do dvou základních kategorií.

#### 9.3.1. Nepohyblivé kameny



**Bomba** – hráč má k dispozici 6 těchto herních kamenů, které se umístí na hrací plochu ve fázi úvodního rozestavování kamenů a po zahájení hry již není možné s těmito kameny pohybovat (táhnout). Tento kámen slouží jako nástraha na soupeřovy pohyblivé kameny, jelikož každý kámen, který zaútočí na tento hrací kámen, je vyřazen ze hry. Jediný kámen, který může vyřadit ze hry hrací kámen Bomba, je Minér.



**Vlajka** – hráč má k dispozici pouze jednu Vlajku. Tento hrací kámen je klíčový pro vývoj hry, jelikož cílem obou hráčů je bránit vlastní vlajku a zajmout vlajku soupeře. Vlajku lze vyřadit jakýmkoliv z pohyblivých hracích kamenů. Stejně jako hrací kámen Bomba je Vlajka umístěna na hrací plochu ve fázi úvodního rozestavování kamenů a po zahájení hry již není možné s těmito kameny pohybovat.

#### 9.3.2. Pohyblivé kameny

Následující popis je od nejsilnějšího kamene k nejslabšímu:



**Maršál** – hráč má k dispozici jednoho Maršála. Tento hrací kámen je nejsilnější ze všech hracích kamenů. Při útoku na libovolný hrací kámen, s výjimkou Bomby, je vyřazen ze hry kámen, na který Maršál útočí. Maršála je možné vyřadit ze hry pouze útokem Špióna.



**Generál** – hráč má k dispozici jednoho Generála. Při útoku na libovolný hrací kámen, s výjimkou Bomby a Maršála, je vyřazen ze hry kámen, na který Generál útočí.



**Plukovník** – hráč má k dispozici dva Plukovníky. Při útoku na libovolný hrací kámen, s výjimkou Bomby a silnějších pohyblivých kamenů, je vyřazen ze hry kámen, na který Plukovník útočí.



**Major** – hráč má k dispozici tři Majory. Při útoku na libovolný hrací kámen, s výjimkou Bomby a silnějších pohyblivých kamenů, je vyřazen ze hry kámen, na který Major útočí.



**Kapitán** – hráč má k dispozici čtyři Kapitány. Při útoku na libovolný hrací kámen, s výjimkou Bomby a silnějších pohyblivých kamenů, je vyřazen ze hry kámen, na který Kapitán útočí.



**Poručík** – hráč má k dispozici čtyři Poručíky. Při útoku na libovolný hrací kámen, s výjimkou Bomby a silnějších pohyblivých kamenů, je vyřazen ze hry kámen, na který Poručík útočí.



**Seržant** – hráč má k dispozici čtyři Seržanty. Při útoku na libovolný hrací kámen, s výjimkou Bomby a silnějších pohyblivých kamenů, je vyřazen ze hry kámen, na který Seržant útočí.



**Minér** – hráč má k dispozici pět Minérů. Minér je jediný hrací kámen, který může útokem vyřadit hrací kámen Bomba. Mimo to vyřadí ze hry oba slabší pohyblivé kameny (Průzkumník, Špión) i Vlajku.



**Průzkumník** – hráč má k dispozici osm Průzkumníků. Průzkumník je jediný hrací kámen, který se může pohybovat nebo útočit v kolmém směru dále než na sousední políčko (viz Pohyb a útok kamenů). Je tedy vhodný pro zjišťování síly a umístění soupeřových kamenů. Průzkumník může vyřadit ze hry pouze Špióna a Vlajku.



**Špión** – hráč má k dispozici pouze jednoho Špióna. Tento hrací kámen je jediný, který může vyřadit ze hry nejsilnější hrací kámen Maršál, ale pouze za předpokladu, že na něho zaútočí. Zaútočí-li Maršál na Špióna, tak je Špión vyřazen ze hry.

## 9.4. Pohyb a útok kamenů

### 9.4.1. Pohyb herními kameny

Hracími kameny Vlajka a Bomba není možné během hry provádět žádný pohyb.

Všemi pohyblivými hracími kameny, s výjimkou hracího kamene Průzkumník, je možný pohyb pouze v kolmém směru a to jen na sousední pole a za předpokladu, že není obsazeno kamenem stejné barvy.

Průzkumník může realizovat pohyb v kolmém směru o libovolný počet volných polí. Není možné přeskočit žádný z herních kamenů.

S žádným kamenem není možné provést pohyb v šikmém směru.



Obrázek 16. Pohyb pohyblivým herním kamenem



Obrázek 17. Pohyb Průzkumníkem

### 9.4.2. Útok herními kameny

Hracími kameny Vlajka a Bomba není možné během hry provádět žádný útok.

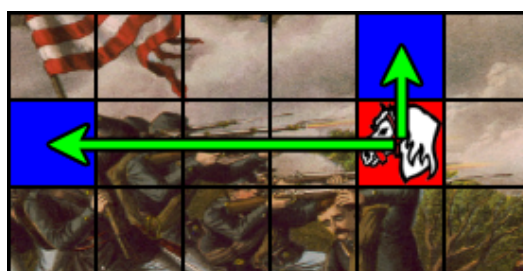
Všemi pohyblivými hracími kameny, s výjimkou hracího kamene Průzkumník, je možné provést útok pouze v kolmém směru a to jen na sousední pole obsazené soupeřovým kamenem. Ve chvíli útoku dojde k odhalení hodnoty (síly) obou hracích kamenů, které se podílí na souboji. Hrací kámen s nižší hodnotou je vyřazen ze hry a hrací kámen s vyšší hodnotou zaujme pozici kamene, na který bylo útočeno. Pokud se na souboji podílí dva kameny se stejnou hodnotou, jsou ze hry vyřazeny oba dva.

Průzkumník může útočit v kolmém směru o libovolný počet volných polí. Není možné přeskočit žádný z herních kamenů. Průběh souboje je stejný jako u ostatních kamenů.

Žádným kamenem není možné provést útok v šikmém směru.



Obrázek 18. Útok pohyblivým herním kamenem



Obrázek 19. Útok Průzkumníkem

### 9.4.3. Pravidlo dvou polí

Při pohybu herními kameny není povoleno táhnout stejným kamenem více než třikrát mezi dvěma stejnými poli (pohyb tam a zpět).

Tím je zamezeno nekonečnému uhýbání před útokem na omezeném prostoru dvou polí.

#### 9.4.4. Pravidlo více polí

Za ohrožování herního kamene je považováno postavení dvou soupeřících kamenů na sousedních polích.

Není povoleno neustálé opakované ohrožování stejného soupeřova kamene stejným kamenem za předpokladu, že by tímto ohrožováním došlo k takovému postavení hracích kamenů na hrací ploše, které, během neustálého ohrožování stejného kamene, již na hrací ploše bylo.

Tímto pravidlem je zamezeno nekonečnému ohrožování a uhýbání dvou kamenů na větším počtu polí.

#### 9.5. Zahájení hry

Na začátku proběhne náhodná volba barvy pro hráče.

Před zahájením hry rozestaví hráči každý svých 40 herních kamenů dle vlastní strategie tak, že každý zná pouze vlastní rozestavení kamenů.

#### 9.6. Průběh hry

Hráč, který má červené hrací kameny, zahajuje hru. Hráči mohou v každém tahu táhnout pouze jednou a to právě jedním kamenem (viz Pohyb a útok kamenů). Hráči se vzájemně střídají na tahu. Hodnosti jednotlivých kamenů soupeře zůstávají po celou dobu hry skryty (každý vidí hodnosti pouze u svých kamenů) a odkrývají se jen při probíhající souboji (útoku) mezi dvěma kameny.

#### 9.7. Vítězství ve hře

Vítězem se hráč stává v případě, že se mu podaří vyřadit ze hry soupeřovu vlajku nebo pokud nemá soupeř již žádné pohyblivé kameny nebo má zbylé pohyblivé kameny rozestaveny tak, že s nimi není možné pohybovat.

Ve hře není možné dosáhnout remízy.

## 10. Uživatelská dokumentace

### 10.1. Ovládání aplikace

Aplikace je určena pro ovládání myší. Pohyb v rámci všech menu je možný pomocí klávesnice. Při použití klávesnice v menu je pohyb, v rámci aktuálně zobrazeného menu, realizován šipkou nahoru a šipkou dolů. Pro přepnutí do inteligentního menu zobrazeného ve spodní části aplikace a pro pohyb v rámci tohoto menu se využívají šipka doleva a šipka doprava. Pro zrušení zobrazení menu nebo pro návrat do menu předchozího je možné použít klávesu Escape. Pro ovládání hracích kamenů při samotné hře je možné použít pouze myš.

### 10.2. Hra proti počítači

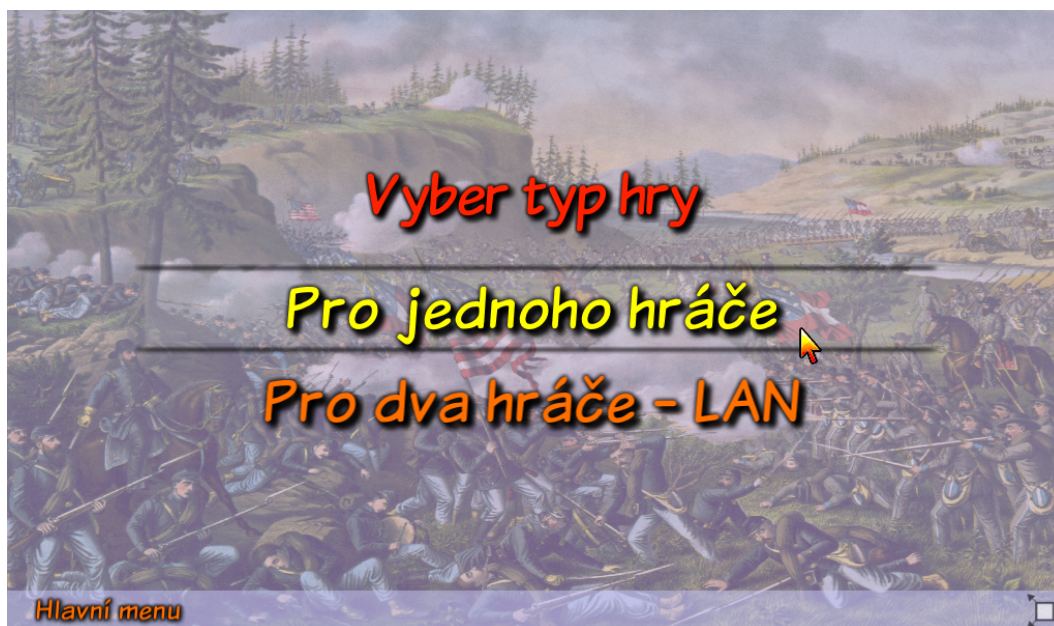
#### 10.2.1. Nová hra proti počítači

1. V hlavním menu stiskněte tlačítko „Hra“.



Obrázek 20. Volba „Hra“ v hlavním menu

2. V menu „Vyber typ hry“ stiskněte tlačítko „Pro jednoho hráče“.



Obrázek 21. Volba „Pro jednoho hráče“ v menu „Vyber typ hry“

3. V menu s označením „Hra pro jednoho hráče“ stiskněte tlačítko „Začít novou hru“.

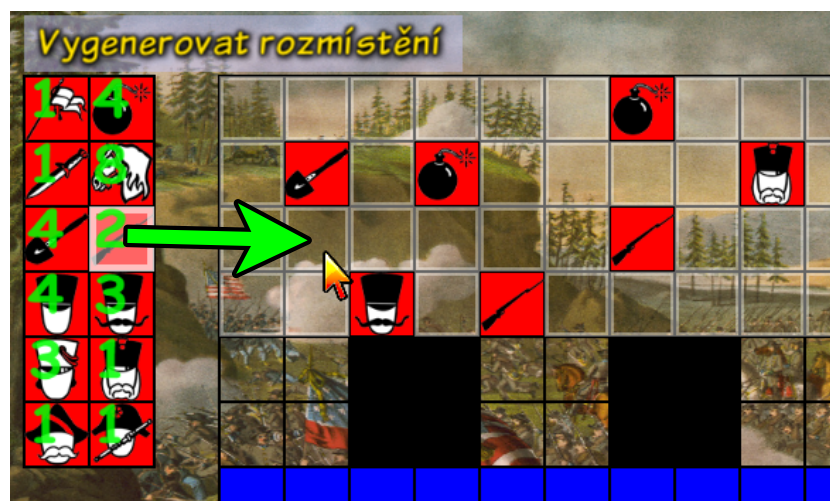


Obrázek 22. Volba „Začít novou hru“ v menu „Hra pro jednoho hráče“



### 10.2.2. Rozestavění herních kamenů a zahájení hry

Při spuštění se nová hra nachází ve fázi, kdy hráči rozstavují vlastní hrací kameny na hrací plochu. Hrací kámen je možné vybrat kliknutím na libovolné tlačítko myši a pak kliknutím na libovolné možné zvýrazněné umístění přesunout kámen. Tímto způsobem je třeba postupně strategicky rozstavět všech 40 herních kamenů.



Obrázek 23. Rozestavování kamenů

Rozestavění všech 40 kamenů je také možné náhodně vygenerovat z databáze herních rozestavění kliknutím na tlačítko „Vygenerovat rozmístění“. Opakovaným klikáním na tlačítko „Vygenerovat rozmístění“ je možné generovat další a další rozestavění kamenů.



Obrázek 24. Generování rozestavění

Pro zahájení hry stiskněte tlačítko „Zahájit hru“, které je umístěno uprostřed hrací plochy a je viditelné jen v případě, že je na hrací ploše umístěno všech 40 herních kamenů. Po zahájení hry se tahy herními kameny provádí stejným způsobem jako při rozestavování kamenů.



Obrázek 25. Tlačítko „Zahájit hru“

### 10.2.3. Uložení a zrušení hry

#### Uložení hry

1. Pro uložení spuštěné hry stiskněte tlačítko „Uložit hru“.



Obrázek 26. Tlačítko „Uložit hru“

2. V dialogu pro uložení hry zadejte název hry pro uložení.
3. Stiskněte tlačítko „Uložit“.



Obrázek 27. Dialog pro uložení hry

### Zrušení hry

1. Pro zrušení spuštěné hry stiskněte tlačítko „Zrušit hru“.



Obrázek 28. Tlačítko „Uložit hru“

2. V dialogovém okně vyberte zda chcete hru před zrušením uložit (tlačítko „Ano“) nebo chcete hru ukončit bez ukládání (tlačítko „Ne“). Máte také možnost dialogové okno zavřít a pokračovat ve hře (tlačítko „Storno“).



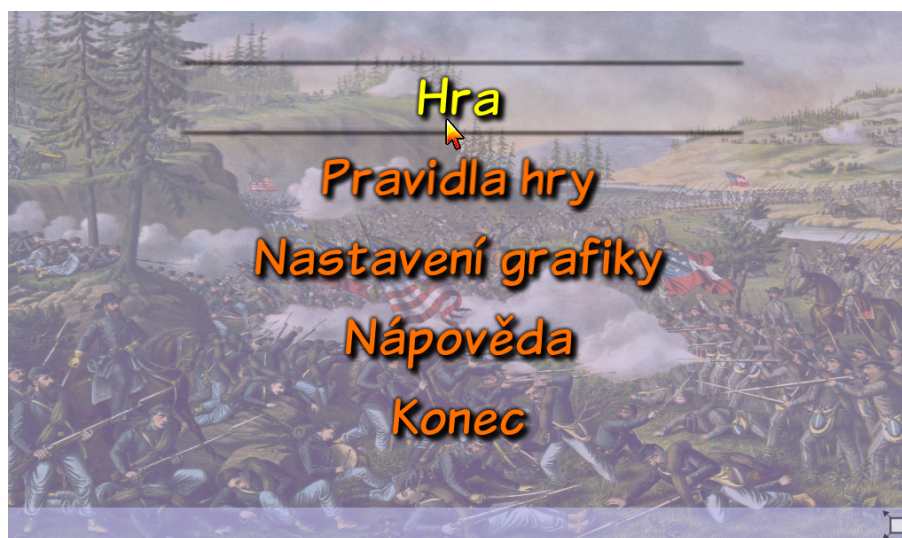
Obrázek 29. Dialogové okno pro zrušení hry

3. V případě, že jste stisknuli tlačítko „Ano“ pro uložení, pokračujte podle pokynů pro ukládání hry.
4. Dojde k návratu do hlavního menu.

## 10.3. Hra po síti

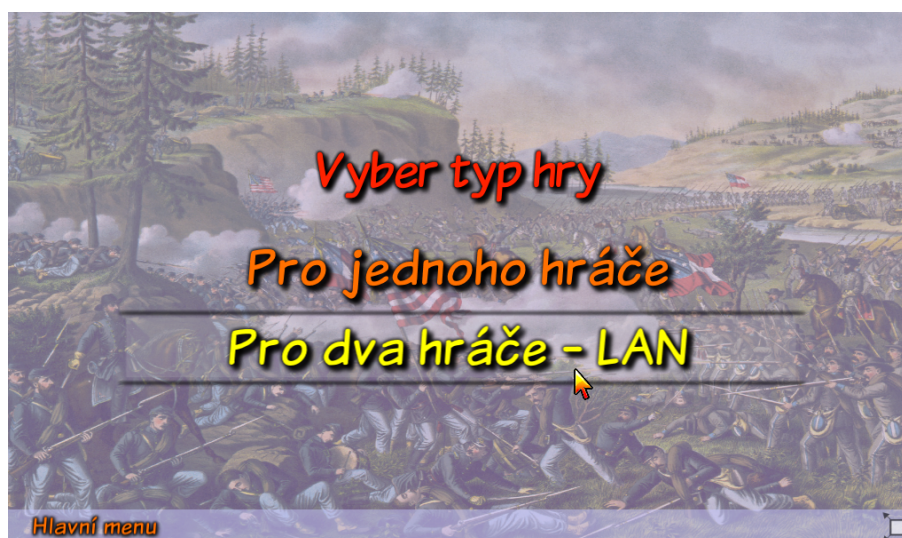
### 10.3.1. Vytvoření síťové hry

1. V hlavním menu stiskněte tlačítko „Hra“.



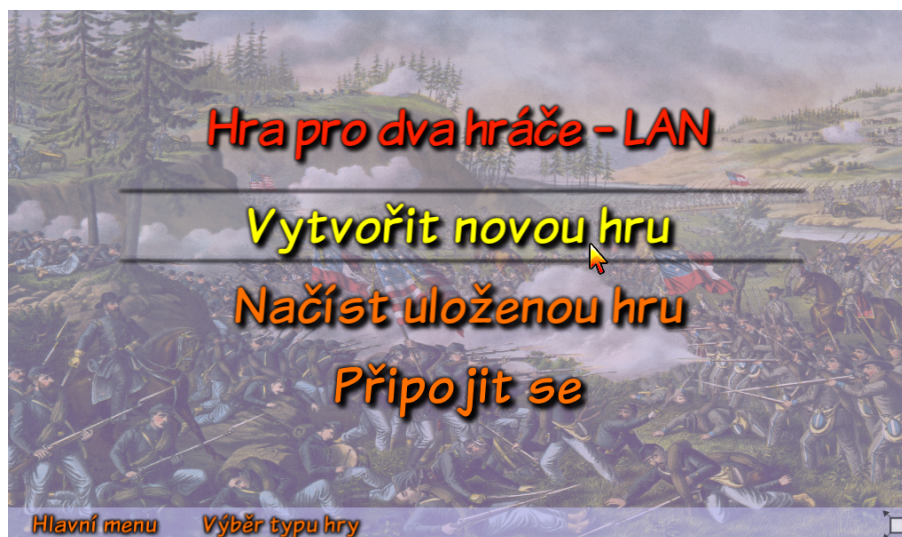
Obrázek 30. Volba „Hra“ v hlavním menu

2. V menu „Vyber typ hry“ stiskněte tlačítko „Pro dva hráče - LAN“.



Obrázek 31. Volba „Pro dva hráče - LAN“ v menu „Vyber typ hry“

3. V menu s označením „Pro dva hráče - LAN“ stiskněte tlačítko „Vytvořit novou hru“.



Obrázek 32. Volba „Vytvořit novou hru“ v menu „Pro dva hráče - LAN“

4. Aplikace čeká na připojení protihráče přes lokální síť. Po jeho připojení dojde k zahájení hry ve fázi rozestavování kamenů.
5. Během rozestavování kamenů i při hře je u soupeřových kamenů zobrazována textově aktuální fáze hry („soupeř přemýšlí“, „soupeř táhne“, apod.).

### 10.3.2. Uložení a zrušení síťové hry

Ukládání a rušení síťové hry probíhá stejně jako při hře proti počítači. V případě, že dojde k výpadku síťového spojení mezi počítači hráčů, je hráč pouze informován, že aktuální fáze hry je „offline“. Pokud hráč chce, může hru uložit. Jakmile dojde k obnovení síťového spojení mezi počítači hráčů, dojde k automatickému navázání na fázi hry, ve které bylo spojení přerušeno.

### 10.3.3. Načtení uložené síťové hry

První dva kroky jsou shodné s vytvořením síťové hry.

3. V menu s označením „Pro dva hráče - LAN“ stiskněte tlačítko „Načíst uloženou hru“.
4. Aplikace čeká na připojení protihráče přes lokální síť. Po jeho připojení dojde k zahájení hry ve fázi, ve které byla dříve uložena.
5. Během rozestavování kamenů i při hře je u soupeřových kamenů zobrazována textově aktuální fáze hry („soupeř přemýšlí“, „soupeř táhne“, apod.).

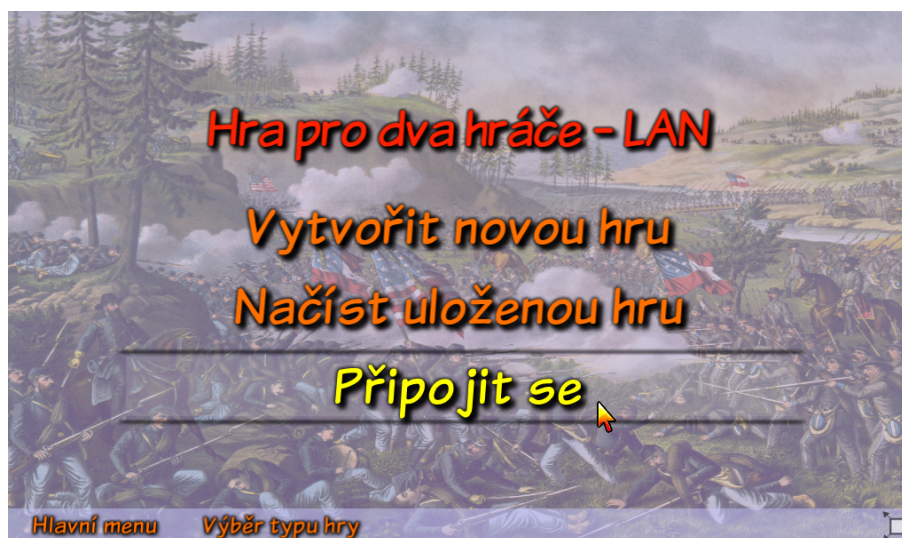


Obrázek 33. Volba „Načíst uloženou hru“ v menu „Pro dva hráče - LAN“

#### 10.3.4. Připojení se k síťové hře

První dva kroky jsou shodné s vytvořením síťové hry.

3. V menu s označením „Pro dva hráče - LAN“ stisknete tlačítko „Připojit se“.



Obrázek 34. Volba „Připojit se“ v menu „Pro dva hráče - LAN“

4. Aplikace počká na potvrzení od protihráče a poté dojde k zahájení hry ve fázi rozestavování kamenů.

5. Během rozestavování kamenů i při hře je u soupeřových kamenů zobrazována textově aktuální fáze hry („soupeř přemýšlí“, „soupeř táhne“, apod.).

## 10.4. Nastavení grafiky

Možnosti nastavení grafického uživatelského prostředí aplikace jsou omezeny pouze na přepínání mezi zobrazení v okně a zobrazení přes celou obrazovku (tzv. fullscreen). V případě zobrazení v okně je velikost okna stanovena automaticky a není možné ji měnit. Změnu nastavení je možné provést dvěma způsoby:

1. V hlavním menu klikněte na položku „Nastavení grafiky“ a v dalším menu vyberte jednu ze dvou nabízených možností.



Obrázek 35. Volba „Nastavení grafiky“ v hlavním menu

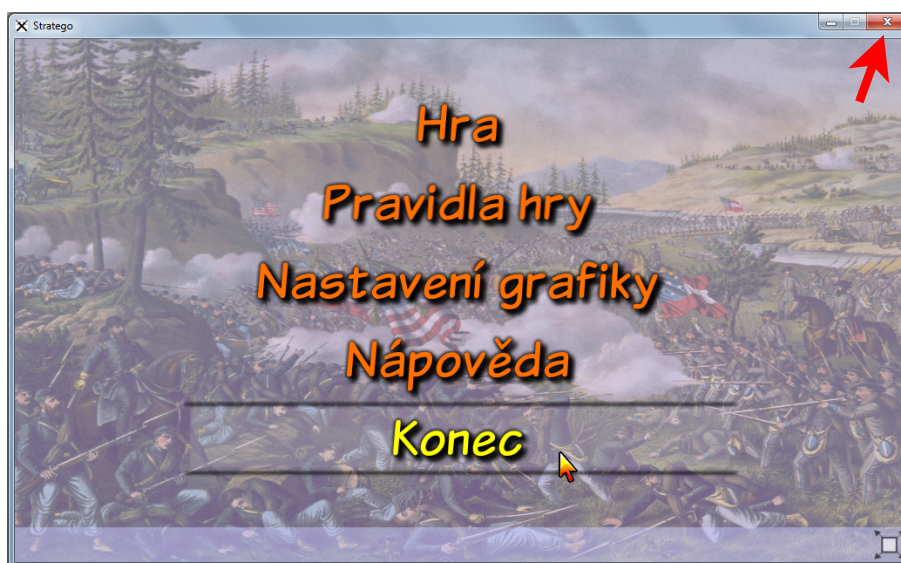
2. V pravém dolním rohu grafického uživatelského rozhraní klikněte na ikonu pro přepnutí režimu zobrazení. Tato ikona je zobrazena ve všech menu i v průběhu hry.

## 10.5. Pravidla a nápověda

Pravidla i nápověda jsou dostupné pouze z hlavního menu kliknutím na položku „Pravidla“ nebo na položku „Nápověda“. Obě části aplikace obsahují text s obrázky popisující samotná pravidla hry a základní způsoby ovládání aplikace. V text je možné posouvat šipkami na klávesnici nebo myší. Pro návrat do hlavního menu lze použít tlačítko „Hlavní menu“ nebo stisk klávesy „Escape“.

## 10.6. Ukončení aplikace

Ukončení aplikace je možné provést kliknutím na tlačítko „Konec“ v hlavním menu nebo, je-li aplikace zobrazena v okně, kliknutím na křížek v pravém horním rohu okna (označen na obrázku). V případě, že je aplikace ve fázi probíhající hry a dojde k ukončení aplikace křížkem v pravém horním rohu nebo kombinací kláves Alt+F4, tak je aktuálně rozehraná hra automaticky uložena bez upozornění a aplikace je ukončena.



Obrázek 36. Ukončení aplikace



## Závěr

Hlavní cíle práce se podařilo naplnit. Vytvořené podpůrné knihovny pro síťovou komunikaci a pro použití ovládacích prvků v XNA Frameworku jsou funkční a použitelné nezávisle v dalších aplikacích. Implementace deskové hry Stratego umožňuje hru proti počítači i proti uživateli připojeného přes lokální počítačovou síť. Aplikace umožňuje ukládání a načítání rozehraných her nezávisle na tom, jedná-li se o hru lokální nebo síťovou. Uživatelské rozhraní aplikace je intuitivní a umožňuje snadné ovládání i laikům.

Při rozboru jednotlivých částí aplikace je dbáno na popsání především hlavních algoritmů. Realizace podpůrné knihovny pro tvorbu uživatelského rozhraní je zaměřena pouze na ovládací prvky použité při implementaci aplikace Stratego. Do budoucna je možné tuto knihovnu rozšiřovat o další ovládací prvky a dále optimalizovat její funkčnost. Podpůrná knihovna pro síťovou komunikaci je zaměřena na přenášení textových dat, což bude vhodné do budoucna dále rozšiřovat o další možnosti.

Realizace algoritmů generujících tahu při hře proti počítači je implementována pouze pomocí naivní ohodnocovací funkce a aplikací ryze útočné herní strategie. Vzhledem k charakteru deskové stolní hry Stratego a variabilitě běžných herních postupů a jejich vzájemné kombinovatelnosti v této hře by jistě pokročilá realizace komplexního algoritmu pro generování tahu počítačem překročila rámec této bakalářské práce. Realizace takového algoritmu by pravděpodobně byla vhodným tématem na práci zcela samostatnou.

## Conclusions

The main goals of the application have been fulfilled. Support libraries for network communication and for use of user controls in XNA Framework are functional and they are independent for use in other projects. The implementation of Stratego game allows playing against computer even against user over local network. The application has functionality of saving and loading played game not dependent on fact if game is local or network. User interface is intuitive to use even for not-experienced users.

Description of parts of application has focused mainly for important algorithms. Implementation of user controls library has focus only on controls used in the implementation of Stratego game. There is possibility to extend library with other controls in future and library can be optimized as well. The network communication support library implements only transmitting of text data. This can be extended in the future as well.

Realization of algorithms for generating of stone moves in the local game against computer is implemented only with simple evaluating function with attacking strategy accent. According to variability and range of game strategies and possibility to effectively combine them during the game is advanced implementation of complex algorithm for generating moves in Stratego game probably is out of scope of this Bachelor's dissertation. Realization of such algorithm seems to be appropriate for standalone dissertation.

## Reference

- [1] What's New in XNA Game Studio 4.0 Refresh. *What's New in XNA Game Studio 4.0 Refresh* [online]. 2012 [cit. 2012-04-13]. Dostupné z: <http://msdn.microsoft.com/en-us/library/bb417503.aspx#ID4EJC>
- [2] *Stratego USA* [online]. 2012 [cit. 2012-04-13]. Dostupné z: <http://www.strategousa.org>
- [3] *Probe: Computer Stratego World Champion* [online]. Copyright 2005-2012 [cit. 2012-04-13]. Dostupné z: <http://www.probe.imersatz.com>
- [4] XNA Developer Center. *XNA Developer Center* [online]. 2012 [cit. 2012-04-13]. Dostupné z: <http://msdn.microsoft.com/en-us/centrum-xna.aspx>
- [5] KABELOVÁ, Alena a Libor DOSTÁLEK. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5.
- [6] Messages and Message Queues. *Messages and Message Queues* [online]. 2012 [cit. 2012-04-13]. Dostupné z: [http://msdn.microsoft.com/en-us/library/ms632590\(v=vs.85\)](http://msdn.microsoft.com/en-us/library/ms632590(v=vs.85))
- [7] *XNA 4.0 Content Compiler* [online]. Copyright 2006-2012 [cit. 2012-04-13]. Dostupné z: <http://xnacontentcompiler.codeplex.com>
- [8] Microsoft Public License. *Microsoft Public License* [online]. Copyright 2006-2012 [cit. 2012-04-13]. Dostupné z: <http://xnacontentcompiler.codeplex.com/license>
- [9] SpriteFont 2. *SpriteFont 2 Texture Tool* [online]. 2009 [cit. 2012-04-13]. Dostupné z: <http://www.nubik.com/SpriteFont/>
- [10] Komika Text Font. *Komika Text Font* [online]. 2006 [cit. 2012-04-13]. Dostupné z: [http://www.fontriver.com/font/komika\\_text/](http://www.fontriver.com/font/komika_text/)
- [11] *Apostrophic Laboratories* [online]. 2012 [cit. 2012-04-13]. Dostupné z: <http://apostrophiclab.pedroreina.net>
- [12] *ISF Game Rules*. 2012. Dostupné z: <http://www.isfstratego.com/images/isfgamerules.pdf>

## A. Obsah přiloženého CD

`bin/`

Instalátor STRATEGOSETUP.EXE aplikace.

`doc/`

Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PřF pro diplomové práce, včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování PDF souboru dokumentace (v ZIP archivu), tj. zdrojový text dokumentace, vložené obrázky, apod.

`src/`

Kompletní zdrojové texty programu STRATEGO, podpůrných knihoven LOXLYLAN a LOXLYXNACONTROLS, testovacích aplikací VASSYNCTV a TESTPROGRAM a upravenou open source aplikaci XNACONTENTCOMPILER.

`readme.txt`

Instrukce pro instalaci a spuštění programu STRATEGO, včetně požadavků pro jeho provoz.

Navíc CD/DVD obsahuje:

`install/`

Instalátor Microsoft .NET Framework 4 a Instalátor Microsoft XNA Framework Redistributable 4.0

`literature/`

Některé položky literatury odkazované z dokumentace práce.

U veškerých odjinud převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovoluují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro materiály, u kterých toto není splněno, je uveden jejich zdroj (webová adresa) v textu dokumentace práce nebo v souboru `readme.txt`.