Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE

Martin Jiřička

Evoluce fotbalových strategií

Kabinet software a výuky informatiky

Vedoucí bakalářské práce:	RNDr. Tomáš Holan Ph.D.
Studijní program:	Informatika
Studijní obor:	programování

Praha 2011

Děkuji své mámě Ing. Daně Jiřičkové za podporu během studia a svému vedoucímu RNDr. Tomáši Holanovi Ph.D. za trpělivost a rady.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Evoluce fotbalových strategií

Autor: Martin Jiřička

Katedra: Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Tomáš Holan Ph.D., Kabinet software a výuky informatiky

Abstrakt: Práce popisuje návrh spojité simulace zjednodušeného fotbalového utkání a evoluce, která dává uživateli možnost pokusit se vyvinout fotbalový tým, který v této simulaci nejlépe obstojí. Popsaná simulace i evoluce byly naprogramovány pro Microsoft .NET Framework. Program umožňuje jednoduchý distribuovaný výpočet evoluce a prohlížení jejích výsledků včetně sledování zápasů ve 2D grafice. Část práce se také věnuje popisu fungování programu a jeho ovládání. Na závěr jsou uvedeny některé provedené výpočty, které mají ilustrovat způsoby používání programu, a také krátká analýza, která na konkrétním příkladu ukazuje, že navržené řešení by mohlo fungovat.

Klíčová slova: fotbal, simulace, evoluce

Title: The Evolution of Football Strategies

Author: Martin Jiřička

Department: Department of Software and Computer Science Education

Supervisor: RNDr. Tomáš Holan Ph.D., Department of Software and Computer Science Education

Abstract: The thesis describes a design of continuous simulation of simplified football match and it also describes an evolution, that offers to a user a chance to evolve a football team, that competes best in the simulation. Both simulation and evolution have been implemented as a program for Microsoft .NET Framework. Program allows simple distributed computation of an evolution and also viewing a results and watching matches in 2D graphics. So that part of the thesis is dedicated to a description of how the program works and how to use it. In the end are mentioned some performed computations illustrating how the program could be used and a small analysis that shows on concrete evolution that the designed solution mights work.

Keywords: football, simulation, evolution

Obsah

Úvod **3**

1 Simulace **4**
1.1 Úvod . . . 4
1.2 Hřiště . . . 5
1.3 Fáze zápasu . . . 5
1.4 Hráči . . . 6
1.4.1 Post . . . 6
1.4.2 Pravidla pro pohyb a kopání do míče . . . 6
1.4.3 Pohyb hráčů - fáze hry . . . 7
1.4.4 Pohyb hráčů - ostatní fáze . . . 8
1.4.5 Kopání do míče . . . 9
1.4.6 Výběr kopu . . . 10
1.5 Statistiky . . . 11
1.5.1 Zápasové statistiky hráčů . . . 11
1.5.2 Zápasové statistiky týmů . . . 12
1.5.3 Statistiky za evoluci . . . 13

2 Evoluce **14**
2.1 Úvod . . . 14
2.2 Simulace zápasů za generaci . . . 14
2.3 Křížení týmů . . . 14
2.4 Mutace . . . 15

3 Program **17**
3.1 Ovládání programů . . . 17
3.1.1 Úvod . . . 17
3.1.2 Spuštění workera . . . 17
3.1.3 Provedení výpočtu evoluce . . . 18
3.1.4 Prohlížení odsimulovaných evolucí . . . 19
3.1.5 Prohlížeč zápasů . . . 20
3.2 Adresář evoluce . . . 20
3.3 Implementace . . . 22
3.3.1 Knihovna JiM.MyLib . . . 22
3.3.2 Hlavní část . . . 23

4 Formáty souborů **24**
4.1 Pravidla pro všechny soubory . . . 24
4.2 Simulace . . . 24
4.2.1 Vzor . . . 24
4.2.2 Popis elementů . . . 25
4.3 Evoluce . . . 26
4.3.1 Vzor . . . 26
4.3.2 Popis elementů . . . 27
4.4 Generace . . . 27

4.4.1 Neúplný vzor 27
4.4.2 Popis elementů 29
4.5 Výsledek evoluce 30
4.5.1 Příklad 30
4.5.2 Popis elementů 31

5 Pokusy **32**
5.1 Úvod 32
5.2 Generování náhodné generace 32
5.3 První evoluce 33
5.4 Druhá evoluce 34
5.5 Hra týmů 36

Závěr **37**

Seznam použité literatury **38**

Přílohy **39**

Úvod

Před několika lety mě uchvátila a na čas zcela pohltila počítačová hra jménem Championship Manager (pozdější verze nesly jméno Football Manager). Jednalo se o velice propracovanou simulaci práce manažera fotbalového klubu. Tato hra byla ceněna pro svoji věrnost a velkou propracovanost. Simulovala vlastně celý fotbalový svět. Hráč mohl najímat skauty, rozvíjet infrastrukturu, obchodovat s hráči, organizovat tréninky a pochopitelně koučovat zápasy.

Právě poslední zmíněná činnost byla stěžejní a sledování virtuálního zápasu mělo mnohdy stejný náboj jako sledování toho skutečného. Ve starších verzích se zápas odehrával pouze formou psaných komentářů, které popisovaly dění na hřišti. Až pozdější verze přinesly jednoduchý pohled shora na hřiště, po kterém se pohybovaly tečky s jmenovkami. Právě toto zpracování zápasu mě zaujalo. Při jeho sledování jsem vždycky přemýšlel, jak může být naprogramováno.

Sám jsem si zkoušel udělat jednoduchou simulaci fotbalového zápasu, která se snažila trochu kopírovat právě onu grafickou verzi, jako zápočet z předmětu Programování. Tato práce na tuto simulaci navazuje a podstatně ji rozšiřuje. Kromě vylepšení simulace přidává i možnost hledání týmu, který by v simulaci nejlépe obstál, pomocí evolučního algoritmu.

Tento text tedy popisuje mnou navržené a realizované řešení. Program umožňuje nadefinovat týmy, které se stanou první generací v evoluci, nastavit parametry simulace i evoluce a nechat vyvíjet nové týmy. Vše je pak možné zpětně projít a sledovat sehraná utkání týmů v dvojrozměrné grafice. Program sám nenabízí moc nástrojů pro vyhodnocování výsledků a průběhu evoluce, ale nabízí možnost, jak exportovat data pro zpracování jinými programy.

Snažil jsem se, aby na sebe text navazoval a při čtení nebylo nutné přeskakovat. Pokud je něco pouze nastíněno, upřesnění by se mělo nacházet dále v textu.

1. Simulace

1.1 Úvod

Simulace má za úkol spočítat, jak dopadne utkání dvou soupeřících fotbalových týmů, kde jeden tým je domácí, druhý hostující. Na konci simulace jsou k dispozici statistiky hráčů obou týmů, které obsahují informace např. o počtu vstřelených gólů nebo úspěšných přihrávek, času hráče u míče, atp. Pravidla výpočtu jsou skutečným fotbalovým zápasem inspirována.

Simulace je spojitá, nedeterministická a celá probíhá ve dvourozměrném prostoru. Jeden krok simulace odpovídá 50 milisekundám reálného utkání, celkem pak zápas trvá vždy přesně 5 minut neboli 6 000 kroků. Rozměry a uspořádání hřiště odpovídají tomu skutečnému. Hráči i míč jsou bezrozměrní, v prostoru jsou to tedy pouze body.

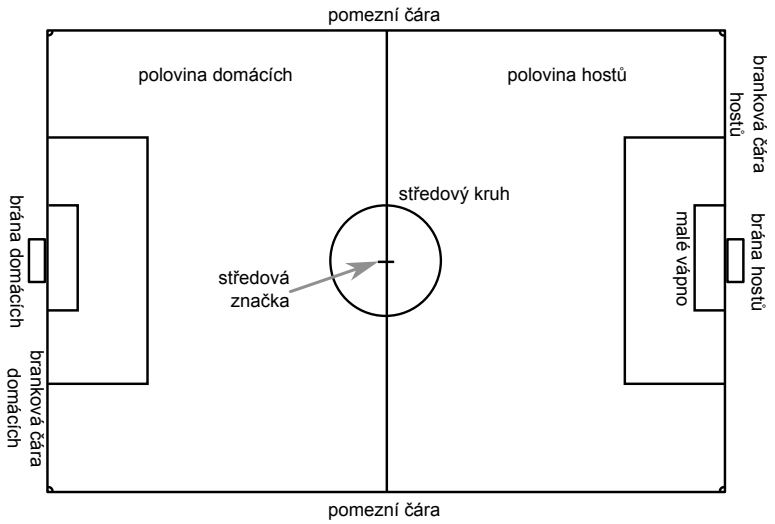
Na hřišti se vždy pohybuje 11 hráčů z každého týmu a míč. Míč na rozdíl od hráčů může hřiště opustit. Pohyb hráče i míče je modelován pomocí vektoru, o který se po hřišti posouvají během jednoho kroku simulace. Vektor míče se po každém kroku zkracuje (míč tedy brzdí), hráči svůj vektor pohybu určují podle pravidel, která jsou parametrizována, přičemž parametry má každý hráč vlastní. Tato pravidla berou v úvahu hráčovo okolí a polohu na hřišti i stav zápasu a jeho fázi. To samé platí i pro pravidla, která určují kopání do míče. Hráč, který má dost blízko sebe míč, do něho může kopnout, neboli může míči změnit směr a délku vektoru pohybu. Hráč tak může přihrát, střílet nebo si posunout míč při driblinku.

Zmíněným stavem zápasu, který ovlivňuje hráčova pravidla, je myšleno, kolik gólů vstřelil ten který tým. Hráči rozlišují dva stavy - když prohrávají a když neprohrávají. Pojem fáze hry pak souvisí s polohou míče. Hlavní fází je tzv. herní fáze, kdy je míč na hřišti. Ostatní fáze jsou aut, výkop a odkop. Jejich účelem je, aby se hra mohla do herní fáze vrátit po té, co míč hřiště opustil. Návrat míče na hřiště se neděje pomocí vektoru pohybu, nýbrž přímo přemístěním míče. Po přemístění má míč nulový vektor pohybu.

Pokud se v této kapitole bude mluvit obecně o nějakém čísle nebo více možnostech, znamená to obvykle, že lze dané číslo nebo možnost nastavit, viz část 4.2.

Průběh simulace by šel popsat následujícím způsobem:

1. Fáze výkopu, u míče je domácí tým.
2. Dokud není konec utkání, opakuje se:
   - 2.1. Pokud je nějaký hráč u míče, kopne do něj, pokud smí.
   - 2.2. Pohne se míč.
   - 2.3. Aktualizuje se fáze zápasu.
   - 2.4. Pohne se postupně s hráči (v náhodném pořadí vůči sobě).
   - 2.5. Zvýší se simulační čas.



Obrázek 1.1: Popis hřiště

1.2 Hřiště

Hřiště má obdélníkový tvar o rozměrech 110 a 75 metrů, uprostřed kratších stran hřiště jsou umístěny brány o šířce 7,32 metru. Delší část hřiště je rozpůlena, každý tým má svoji půlku, na které se nachází jeho brána. Kolem každé brány je umístěno směrem do hřiště tzv. malé vápno. To má rozměry 5,5 a 18,3 metrů, jeho delší strana se kryje s kratší stranou hřiště, přičemž středy stran hřiště a vápna splývají. Dále se na hřišti nachází středová značka, která je umístěna v prostředku hřiště. Konečně posledním prvkem hřiště je středový kruh, který má poloměr 9,15 metru a střed ve středové značce.

Delším stranám hřiště se říká pomezní čáry. Kratší strana hřiště je nazývána brankovou čarou, která patří týmu, u jehož brány se nachází.

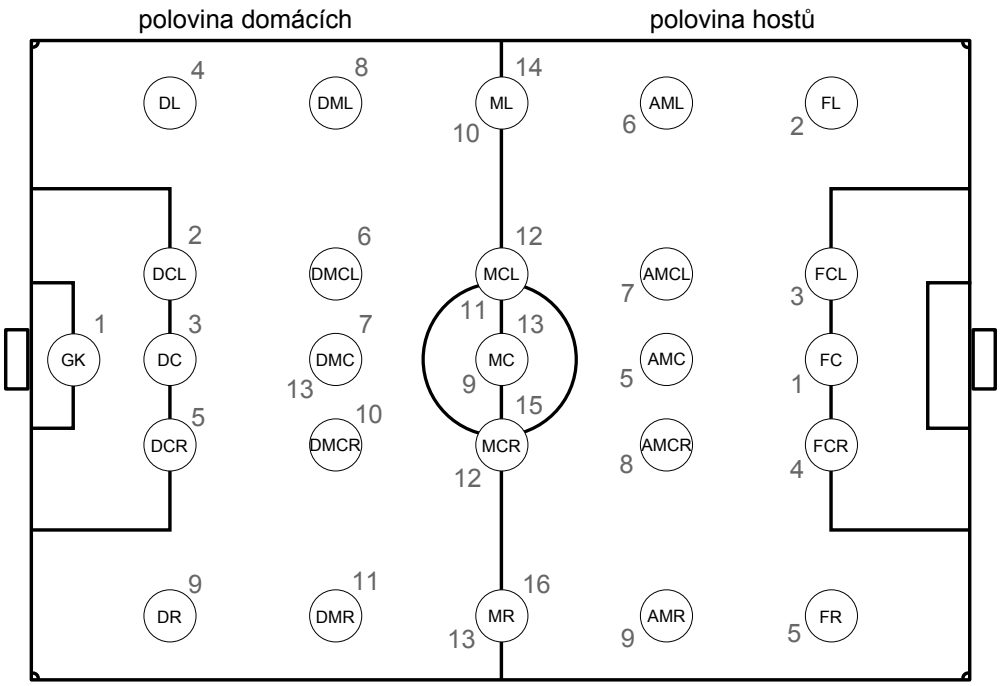
Pro lepší přehled je hřiště na obrázku 1.1, kde jsou znázorněny i čáry, které v simulaci nehrají roli, ale nachází se na reálném hřišti.

1.3 Fáze zápasu

Fáze, kdy se míč pohybuje po hřišti, se nazývá fází hry. Nastat mohou další tři fáze: aut, odkop a výkop. Pokud přešel míč pomezní čáru, je fáze autu. Pokud prošel bránou (tedy padl gól), je výkop, ve zbylých případech je odkop. Do fáze hry se zápas vrací vždy po té, co nějaký hráč kopne do míče. Tento kop je proveden vždy ve chvíli, kdy se všichni hráči vrátí na svá výchozí místa na hřišti.

Když míč hřiště opustí, pokračuje několik sekund ve své dráze. Následně se objeví zpět na hřišti. Na začátku fáze autu se míč vrátí tam, kde byl na hřišti naposledy před tím, než ho opustil. Při odkopu se míč přemístí do rohu malého vápna. Při výkopu je míč vrácen na středovou značku, ve středovém kruhu se přitom může nacházet pouze jeden hráč, který je z opačného týmu, než kterému patří brána, do které gól padl.

Simulace začíná fází výkopu, vykopávající hráč je z domácího týmu.



Obrázek 1.2: Hřiště s vyznačenými posty

1.4 Hráči

1.4.1 Post

Každý hráč má svůj post. Na nějakém postu může v jednom týmu hrát pouze jeden hráč. Funkcí postu je určovat hráči jeho výchozí polohu na hřišti, která je s každým postem spjatá. Všichni hráči v poli (tj. všichni kromě brankáře) mají toto místo pevně dané. Brankář má svou výchozí pozici na průsečíku úsečky spojující střed brány s míčem a kružnice, který má poloměr 5 metrů a střed uprostřed brány brankářova týmu.

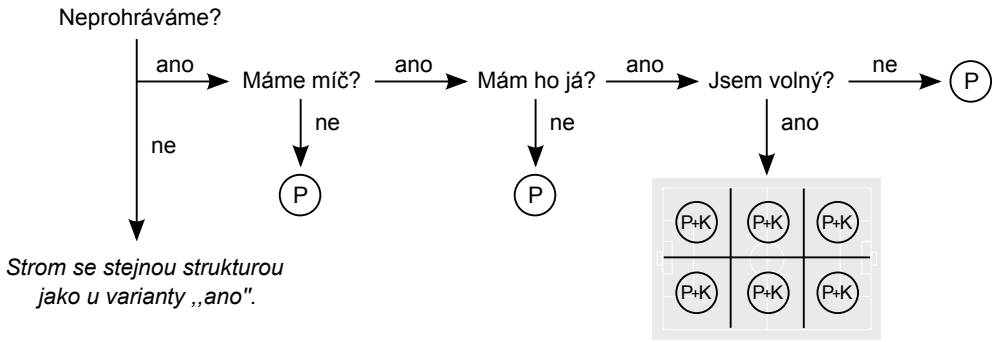
Zkratky názvů pozic a k nim odpovídající místa na hřišti jsou na obrázku 1.2, který je kreslen z pohledu domácího týmu. Zkratky jsou složeny z písmen: **F** - Forward, **M** - Midfield, **D** - Defender, **GK** - Goalkeeper, **A** - Attacking, **D** - Defensive, **C** - Center, **L** - Left, **R** - Right.

Na obrázku jsou také čísla, která označují pořadí postů pro vykonávání odkopů od brány (vpravo nahoře od zkratky pozice), nebo výkopů (vlevo dole od zkratky pozice) hráči, kteří na těchto postech hrají. Při určování postu, který bude provádět odkop nebo výkop, se vždy vybere ten s nejmenším číslem, jenž je v týmu zastoupen (tedy na tomto postu hraje nějaký hráč). Pokud číslo není uvedeno, znamená to, že na takový post nemůže tato činnost nikdy vyjít.

1.4.2 Pravidla pro pohyb a kopání do míče

Jak bylo řečeno, každý hráč se na hřišti v různých situacích jinak pohybuje a kope do míče. Postup, jakým se vybírá směr pohybu či směr a síla kopu, je vždy stejný, liší se pouze jistými číselnými parametry z pevně daného rozsahu (více o těchto parametrech je v sekci 4.4).

Jaká sada těchto parametrů se použije v jaké situaci, je znázorněno na obráz-



Obrázek 1.3: Pravidla pro výběr parametrů

ku 1.3, kde je rozhodovací strom z pohledu hráče jednoho týmu (první osobou množného čísla v otázkách se myslí tým tohoto hráče). Jak je z obrázku vidět, výběr parametrů je závislý na aktuálním stavu zápasu. Dále rozhoduje, který tým kontroluje míč (což je ten tým, jehož hráč naposledy kopl do míče) a zda ho má nějaký hráč pod kontrolou (tedy tento hráč je v určité vzdálenosti od míče a je posledním, kdo do něho kopl).

P v kroužku označuje sadu parametrů pro pohyb, P + K označuje sadu parametrů pro pohyb a zároveň kopání do míče. Znázorněné hřiště má naznačit, že sada parametrů bude vybrána podle toho, v jaké části hřiště se hráč právě nachází. Části hřiště jsou jednotlivá políčka mříže (na obrázku o dvou řádcích a třech sloupcích, ovšem počet se dá měnit), která pokrývá celé hřiště.

1.4.3 Pohyb hráčů - fáze hry

Pohyb hráče je určen vektorem, který se získá jako součet jiných sedmi vektorů (některé z nich vznikají součtem dalších vektorů). Pokud je výsledný vektor příliš dlouhý, zkrátí se na pevně danou délku, což znamená, že hráči mají jistou maximální rychlost. Délky sedmi dílčích vektorů se nastavují na délku, která je dána výrazem $N\frac{v}{(|v|+S)^M}$ (dále se slovem „výraz“ myslí právě tento), kde v je onen vektor a N, S, M jsou parametry. Sedmice těchto parametrů (jedna pro každý vektor) tvoří zmíněnou sadu parametrů pro pohyb. Následuje přehled vektorů:

Pomezní čáry

Vektor jedné pomezní čáry je kolmý na tuto pomezní čáru a vede od ní směrem k hráči, jeho délka se upraví na délku danou výrazem. Pomezní čáry jsou dvě, proto výsledný vektor pomezních čar se získá jako součet vektorů obou pomezních čar.

Branková čára soupeře

Vektor je kolmý k této čáře a vede od hráče směrem k čáře.

Míč

Pokud se míč přibližuje směrem k hráči, označme jako cílový bod průsečík dvou polopřímek. První je určena pozicí míče a vektorem jeho pohybu, druhá pak

pozicí hráče a vektorem jeho pohybu. Pokud se míč od hráče vzdaluje, je cílový bod ztotožněn s míčem. Pro hráče, který je z týmu, který kontroluje míč, ale sám ho nekontroluje, vede vektor od cílového bodu směrem k hráči. Pro hráče, který není z týmu, který kontroluje míč, vede vektor od hráče směrem k cílovému bodu. Pro hráče, který kontroluje míč, je vektor náležící míči nulový.

Spoluhráči

Vektor je výsledkem součtu vektorů určených jednotlivými spoluhráči, jejich délky se upravují na délku danou výrazem. Vektor spoluhráče vede od spoluhráče směrem k hráči.

Protihráči

Vektor je výsledkem součtu vektorů určených jednotlivými protihráči, jejich délky se upravují na délku danou výrazem. Pro hráče týmu, který kontroluje míč, vede vektor protihráče od protihráče směrem k hráči. Pro hráče týmu, který míč nekontroluje, vede vektor od hráče směrem k protihráči.

Hranice ofsajdu

Hranice ofsajdu nějakého týmu je přímka, která je kolmá na pomezní čáry a prochází hráčem z tohoto týmu, který je nejblíže vlastní bráně a přitom nehraje na postu brankáře.

Pokud je míč nebo daný hráč mezi hranicí ofsajdu soupeře a jeho brankovou čarou, je tento vektor roven nule. Jinak je vektor hranice ofsajdu kolmý na hranici ofsajdu a vede od ní směrem k hráči.

Tento vektor měl původně zajistit, že se hráči nedostanou do ofsajdu. Ofsajd byl nakonec v simulaci kvůli problémům zrušen, pravidla určování vektoru se změnila a jeho význam je nyní spíše teoretický.

Výchozí pozice

Vektor výchozí pozice vede od hráče na místo, které určuje post hráče.

1.4.4 Pohyb hráčů - ostatní fáze

Ve fázi autu a výkopu je vektor pohybu roven vektoru výchozí pozice, jehož výpočet probíhá stejně jako v herní fázi. Výjimku z tohoto pravidla mají hráči, kteří rozehrávají míč.

Při autu je rozehrávajícím hráčem ten, který se právě nachází nejblíže místu, kam bude vrácen míč, a je z opačného týmu než toho, který naposledy míč kontroloval.

Při odkopu rozehrává hráč, jenž je na postu, který byl zvolen k odkopávání.

Při výkopu jsou pozice určené postem stlačeny na vlastní polovinu, přičemž ty, které by padly do středového kruhu, jsou umístěny na jeho obvod nejblíže k původnímu místu. To neplatí pro vykopávajícího hráče, který má pozici ve středu kruhu.

1.4.5 Kopání do míče

Níže jsou uvedeny tři typy kopů. Až po nich je popsán způsob, jakým hráč mezi kopy vybírá.

K tomu, aby hráč mohl do míče kopnout, musí být dost blízko v okolí míče. Po té, co kopne, má míč chvíli čas, aby se hýbal. Hráč do něj znovu kopnout nemůže. To proto, aby měl míč čas opustit onen „akční rádius“ hráče.

Driblink

Délka vektoru, který hráč nastaví míči, odpovídá maximální možné rychlosti hráče. Směr vektoru je stejný jako směr pohybu kopajícího hráče, pokud by na něj míč vůbec nepůsobil. Řečeno jinak je směr stejný jako součet šesti vektorů (nepoužitý je vektor určený míčem) určující pohyb hráče s těmi parametry, podle kterých se hráč aktuálně hýbe.

Střela

Rychlost, jakou může hráč vystřelit, je pevně dána. Směr se určí následujícím způsobem.

Brána soupeře se rozdělí na 20 stejně velkých úseků. Každému úseku se přiřadí převrácená hodnota ze součtu vzdáleností všech hráčů od spojnice míče a středu úseku. Řečeno přesněji se toto číslo spočítá jako součet délek úseček, které jsou kolmé na polopřímku s počátečním bodem v míči a pomocným bodem ve středu úseku, jeden krajní bod mají úsečky na této polopřímce a druhý v hráči na hřišti. Pokud pro nějakého hráče taková úsečka neexistuje, nic se nezapočítává.

Hráč střílí směrem do středu úseku, který má přiřazenou nejmenší hodnotu, tedy úseku, který má v jistém smyslu nejlepší předpoklady, aby k němu míč doputoval.

K vybranému směru se ještě může přidat určitá odchylka.

Přihrávka

Rychlost, jakou hráč udělí míči při přihrávce, je pevně dána. Směr se určí následujícím způsobem.

Celé hřiště se pokryje mříží o daném počtu sloupců a řádků. Pro každé políčko v mříži se určí trojice hodnot:

Riziko, šance Riziko (resp. šance) určitého políčka se spočte tak, že se najdou políčka, po kterých prochází úsečka začínající na pozici míče a končící ve středu daného políčka. Pro každé nalezené políčko se zjistí nejdelší doba, po kterou se na něm může nacházet nějaký hráč soupeře (resp. spoluhráč) spolu s pohybujícím se míčem. Přitom se uvažuje pohyb hráčů i míče. Doby ze všech políček se sečtou. Riziko (šance) určitého políčka je poměr celkové doby pohybu míče na cílové políčko a celkové doby strávené spolu s míčem protihráči (resp. spoluhráči) na políčkách cestou.

Posun vpřed Očíslujeme sloupce políček tak, že sloupec, ve kterém se právě nachází míč, je nultý a každý další sloupec směrem k bráně soupeře má číslo o jedna větší než předešlý a naopak směrem k vlastní bráně o jedna

menší. Pak číslo sloupce, ve které se nachází zkoumané políčko, je číslo posunu vpřed.

Po spočítání rizika, šance a posunu vpřed na všech políčkách se provede následující.

1. Políčka se seřadí podle rizika.
2. Políčka s příliš vysokým rizikem a příliš malou šancí (obě hranice stanovují parametry) se přestanou uvažovat.
3. Pokud se po předchozím kroku přestala uvažovat všechna políčka, hráč vystřelí na bránu.
4. Najdou se políčka s největším číslem posunu vpřed. Toto číslo musí být zároveň větší než jistý parametr. Z těchto políček se vybere to s nejvyšší šancí.
5. Pokud žádná políčka předchozímu kroku nevyhovují, uvažují se ty s největším posunem vpřed.
6. Pokud existuje více políček se stejnou nejvyšší šancí, rozhodne se mezi nimi náhodně.
7. Míč se kopne směrem ke středu „vítězného“ políčka. Náhodně se připočte odchylka.

Dalo by se tedy říct, že se vybírá políčko, které má slušnou šanci, že ho na něm zachytí spoluhráč, ovšem ne příliš velké riziko, že bude míč po cestě zachycen protihráčem. Z možných políček se vybírají ty nejblíže bráně soupeře.

Výše popsaný postup má tedy tři parametry: hranici rizika, minimální šanci a minimální posun vpřed.

Obrázek 1.4 je pořízen přímo z části programu. Zobrazuje políčka při výběru přihrávky. Pro lepší viditelnost byl míč označen černě. Čísla po směru hodinových ručiček se začátkem vlevo nahoře udávají: za jak dlouho se na políčko dostane nějaký spoluhráč, za jak dlouho se na políčko dostane nějaký protihráč, míru rizika, míru šance. Riziko a šance jsou zobrazeny i graficky v poměru k ostatním hodnotám. Riziko je zobrazeno v horní polovině políčka, šance v dolní. Políčka, z kterých hráč vybírá, jsou orámována.

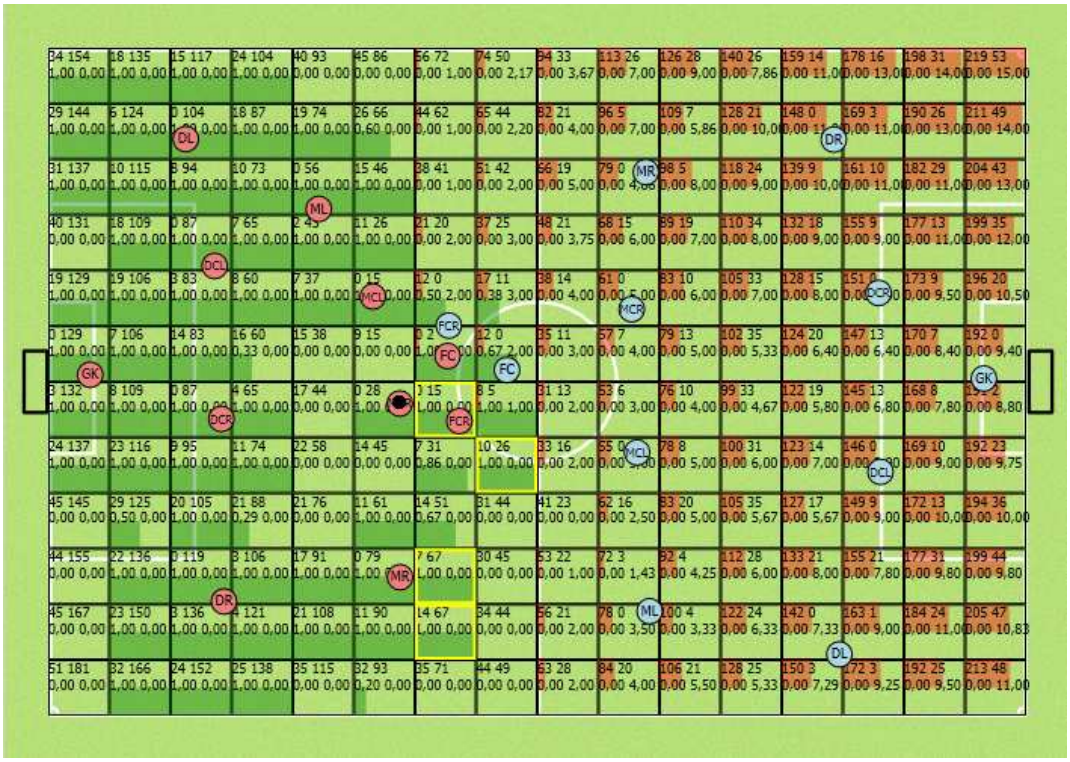
1.4.6 Výběr kopu

Jestli hráč vystřelí, přihraje nebo bude driblovat se rozhodne následujícím postupem:

1. Pokud existuje vynikající možnost střely, tak hráč vystřelí.
2. Pokud existuje vynikající možnost přihrávky, tak hráč přihraje.
3. V ostatních případech se s určitou pravděpodobností buď přihraje, vystřelí nebo dribluje.

Vynikající možnost střely (resp. přihrávky) se určí tak, že hráč má parametr (resp. trojici parametrů pro riziko, šanci, posun vpřed) a pokud střela (resp. přihrávka) určená postupem pro střelu (přihrávku), je lepší než daný parametr (trojice parametrů), pak je také vynikající. Je-li možností víc, rozhodne náhoda.

Zda je střela či přihrávka lepší než druhá, je intuitivně jasné. Jedna střela je lepší než druhá, pokud má lepší předpoklad doputovat k vybranému úseku. Jedna



Obrázek 1.4: Pravidla pro výběr parametrů

přihrávka je lepší než druhá, pokud míč posune alespoň tak daleko, má alespoň takovou šanci a nejvýše takové riziko.

Výběr kopu má tedy parametry pro určení vynikající možnosti střely, přihrávky a pravděpodobností.

1.5 Statistiky

1.5.1 Zápasové statistiky hráčů

Každému hráči se během zápasu zaznamenává do statistik jeho počínání. Všechny statistiky jsou na začátku zápasu nulové. Následuje seznam položek a popis, kdy a jak se mění.

Góly

Tato statistika se zvýší hráči, který jako poslední kopl do míče před tím, než přešel brankovou čáru soupeře.

Vlastní góly

Tato statistika se zvýší hráči, který jako poslední kopl do míče před tím, než přešel brankovou čáru svého týmu.

Úspěšné přihrávky

Pokud hráč přihrával a po něm kopl do míče spoluhráč, je tomu prvnímu zvýšena tato statistika.

Zkažené přihrávky
Pokud hráč přihrával a po něm kopl do míče soupeř, je tomu prvnímu zvýšena tato statistika.

Zachycené přihrávky
Pokud hráč přihrával a po něm kopl do míče soupeř, je tomu druhému zvýšena tato statistika.

Odebrané míče
Když hráč dribloval a po něm kopl do míče soupeř, je tomu druhému zvýšena tato statistika.

Ztráty míče
Když hráč dribloval a po něm kopl do míče soupeř, je tomu prvnímu zvýšena tato statistika.

Střely na bránu
Pokud hráč vystřelil na bránu soupeře a směr míče mířil do brány, je mu zvýšena tato statistika.

Střely mimo bránu
Pokud hráč vystřelil na bránu soupeře a směr míče mířil mimo bránu, je mu zvýšena tato statistika.

Asistence
Pokud hráč vstřelí gól, pak všem ostatním spoluhráčům, kteří kopli do míče po posledním kopu soupeře, se zvýší tato statistika.

Čas u míče
Doba, která uplyne mezi okamžikem, kdy hráč kopl do míče, a okamžikem, kdy do něho kopl někdo jiný, je přičtena k této hráčově statistice.

1.5.2 Zápasové statistiky týmů

Statistiky týmů vznikají výhradně ze statistik hráčů obou týmů.

Jedna část statistik vzniká sečtením stejnojmenných statistik všech hráčů tohoto týmu. To platí pro: **Vlastní góly**, **Úspěšné přihrávky**, **Zkažené přihrávky**, **Střely na bránu**, **Střely mimo bránu**, **Čas u míče**.

Další statistiky týmů pak jsou:

Vstřelené góly
Tato statistika vznikne sečtením statistiky Góly všech hráčů tohoto týmu, k nimž se přičte součet statistik Vlastní góly všech hráčů z týmu soupeře.

Obdržené góly
Tato statistika vznikne sečtením statistiky Vlastní góly všech hráčů tohoto týmu, k nimž se přičte součet statistik Góly všech hráčů v týmu soupeře.

Vítězství
Zvedne se o jedna, pokud je statistika týmu Vstřelené góly ostře větší než stejná statistika u týmu soupeře.

Remízy

Zvedne se o jedna, pokud je statistika týmu Vstřelené góly rovna stejné statistice u týmu soupeře.

Prohry

Zvedne se o jedna, pokud je statistika týmu Vstřelené góly ostře menší než stejná statistika u týmu soupeře.

1.5.3 Statistiky za evoluci

Statistiky týmů i hráčů za evoluci obsahují stejné položky. Jejich hodnoty vznikají součtem statistik ze zápasů.

2. Evoluce

2.1 Úvod

Během evoluce se hledá tým, který bude nejlépe vyhovovat kritériu dané uživatelem. Nejen toto kritérium, ale i další věci se dají nastavit, viz 4.3. K hledání je využit princip evolučního algoritmu, který je pojednán v [1].

Vyvíjející se jedinci jsou zde týmy, jejich chromozom je složen výhradně z chromozomu hráčů tohoto týmu. Chromozom hráčů obsahuje hráčův post a parametry pro pohyb a kopání do míče.

Fitness týmu se vypočte z jeho statistik za evoluci. Při křížení týmů dochází ke křížení hráčů, neboť tým není nic jiného než skupina jedenácti hráčů. Při křížení hráčů hraje roli druhá fitness počítaná při evoluci a tou je fitness hráčů. Ta se také počítá ze statistik hráče za evoluci.

Křížení týmů probíhá tak, že se vždy použijí dva týmy ze stávající generace pro vytvoření nového týmu. Velikost nové generace je vždy rovna velikosti generace stávající.

Průběh evoluce by šel popsat následujícím způsobem:

1. Dokud není dosaženo cílové generace nebo uživatel nepřerušil výpočet, provádí se:
   1.1. Odsimulují se všechny zápasy týmů z aktuální generace.
   1.2. Křížením se vytvoří nová generace.
   1.3. Na každý tým z nové generace je aplikována mutace.
   1.4. Zvýší se číslo generace.
   1.5. Nová generace je prohlášena za aktuální.

2.2 Simulace zápasů za generaci

Pouze týmy z jedné generace mezi sebou hrají zápasy. Jsou dvě možnosti. Buď týmy hrají ligovým způsobem, tj. každý hraje s ostatními dvakrát - jednou doma a jednou venku. Druhým způsobem je systém každý s každým, kdy každý tým hraje se všemi ostatními týmy jen jednou a náhodně se vybere, kdo bude vystupovat v roli domácího týmu.

2.3 Křížení týmů

Pomocí ruletové selekce (popsané opět v [1, kapitola 5.4], ovšem s rozdílem, že zde je ruleta „roztočena“ $2N$-krát, kde N je počet jedinců, tj. týmů, v generaci) se vyberou do dvojic týmy, které se mezi sebou budou křížit. Křížit dva týmy znamená křížit jejich hráče. Je dobré křížit hráče na stejných, nebo alespoň podobných postech, proto každé dva posty mají definovanou vzájemnou podobnost, což je nějaké přirozené číslo od 0 do 20. Čím je číslo menší, tím jsou si posty podobnější. Definované podobnosti postů jsou vypsány v tabulce 2.1. Nejprve se tedy spárují hráči tak, aby součet podobností postů spárovaných hráčů byl co nejmenší. K tomu se používá maďarský algoritmus (viz [2]).

	GK	DR	DCR	DC	DCL	DL	DMR	DMCR	DMC	DMCL	DML	MR	MCR	MC	MCL	ML	AMR	AMCR	AMC	AMCL	AML	FR	FCR	FC	FCL	FL
GK	0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
DR	20	0	1	4	2	0	2	4	6	5	3	4	6	8	7	5	6	8	10	9	7	8	10	12	11	9
DCR	20	1	0	1	0	2	3	2	3	4	5	6	4	5	7	7	8	6	7	9	9	10	8	9	9	11
DC	20	4	1	0	1	4	7	3	2	3	6	9	5	4	5	8	11	7	6	7	10	13	9	8	9	12
DCL	20	2	0	1	0	1	4	3	3	2	4	7	5	5	4	4	9	7	7	6	8	11	9	9	8	10
DL	20	0	2	4	1	0	3	4	6	4	2	5	6	8	5	4	7	9	10	8	6	9	11	12	10	8
DMR	20	2	3	7	4	3	0	1	4	2	0	2	4	6	5	3	4	6	8	7	5	6	8	10	9	7
DMCR	20	4	2	3	3	4	1	0	1	0	2	4	2	4	3	5	4	4	6	5	4	8	6	10	7	9
DMC	20	6	3	2	3	6	4	1	0	1	4	6	3	2	3	6	8	5	4	5	8	10	7	6	7	10
DMCL	20	5	4	3	2	4	2	0	1	0	1	5	3	4	2	4	7	5	6	4	6	9	7	10	6	8
DML	20	3	5	6	4	2	0	2	4	1	0	3	5	6	4	2	5	7	8	6	4	7	9	10	8	6
MR	20	4	6	9	7	5	2	4	6	5	3	0	1	4	3	0	2	4	6	5	3	4	6	8	7	5
MCR	20	6	4	5	5	6	4	2	3	3	5	1	0	1	0	2	4	2	4	3	4	6	4	6	5	7
MC	20	8	5	4	5	8	6	4	2	4	6	4	1	0	1	4	6	3	2	3	6	8	5	4	5	8
MCL	20	7	7	5	4	5	5	3	3	2	4	3	0	1	0	1	4	3	4	2	4	7	5	6	4	6
ML	20	5	7	8	4	4	3	5	6	4	2	0	2	4	1	0	3	4	6	4	2	5	7	8	6	4
AMR	20	6	8	11	9	7	4	4	8	7	5	2	4	6	4	3	0	2	4	2	0	2	4	6	5	3
AMCR	20	8	6	7	7	9	6	4	5	5	7	4	2	3	3	4	2	0	1	0	2	4	2	3	3	5
AMC	20	10	7	6	7	10	8	6	4	6	8	6	4	2	4	6	4	1	0	1	4	6	3	2	3	6
AMCL	20	9	9	7	6	8	7	5	5	4	6	5	3	3	2	4	2	0	1	0	1	5	3	3	2	4
AML	20	7	9	10	8	6	5	4	8	6	4	3	4	6	4	2	0	2	4	1	0	3	5	6	4	2
FR	20	8	10	13	11	9	6	8	10	9	7	4	6	8	7	5	2	4	6	5	3	0	2	4	3	0
FCR	20	10	8	9	9	11	8	6	7	7	9	6	4	5	5	7	4	2	3	3	5	2	0	1	0	3
FC	20	12	9	8	9	12	10	10	6	10	10	8	6	4	6	8	6	3	2	3	6	4	1	0	1	4
FCL	20	11	9	9	8	10	9	7	7	6	8	7	5	5	4	6	5	3	3	2	4	3	0	1	0	2
FL	20	9	11	12	10	8	7	9	10	8	6	5	7	8	6	4	3	5	6	4	2	0	3	4	2	0

Obrázek 2.1: Tabulka podobnosti postů

Nově vznikající hráč přejímá jednotlivé geny pro pohyb a kopání do míče náhodně od dvou spárovaných hráčů. Přitom pravděpodobnost, že si vezme určitý gen od prvního hráče, je rovna $\frac{f_1}{f_1+f_2}$, kde $f_{1,2}$ označuje fitness prvního, resp. druhého hráče.

Jinak je tomu při předávání postu. Tam jsou čtyři možnosti:

1. Nový hráč přejme post vždy od hráče, jenž má vyšší fitness.
2. Nový hráč přejme post vždy od hráče, jehož tým má vyšší fitness.
3. U každého hráče se rozhodne náhodně, zcela nezávisle na fitness.
4. Rozhodne se před křížením dvou týmů náhodně a zcela nezávisle na fitness, z jakého týmu vždy hráči předají svůj post novému týmu.

Pro všechny možnosti platí, že pokud by byl předávaný post v novém týmu už zastoupen, je určen náhodně.

Je tedy vidět, že pokud se o postu rozhoduje podle týmů, bude mít nový tým rozestavení vybraného týmu. Nová rozestavení tak mohou vznikat pouze pokud se o postu rozhoduje podle hráčů.

2.4 Mutace

I mutace týmů znamená mutaci jeho jednotlivých hráčů.

Mutace hráčova postu znamená, že se s určitou pravděpodobností změní hráčův post na nějaký jiný, který v týmu ještě není zastoupen. Tato mutace je tedy nezávislá.

Naopak mutace genů pro pohyb a kopání do míče je závislá. Parametr přihrávky určující minimální posun vpřed se vždy náhodně zmenší nebo zvětší o jedna.

Pravděpodobnost přihrávky, střely nebo driblinku se vždy náhodně zmenší nebo zvětší o 5%, ale tak, aby zůstal zachován součet 100%. Mutace ostatních genů probíhá tak, že se ke každému genu s určitou pravděpodobností přičte změna, která vznikne součinem velikosti rozsahu možných hodnot mutovaného genu s hodnotou z normálního rozdělení, které má nulovou střední hodnotu a určitý rozptyl.

3. Program

3.1 Ovládání programů

3.1.1 Úvod

Výpočet evoluce je realizován jako distribuovaný výpočet po síti, protože evoluce s více týmy je poměrně náročná na výpočetní výkon. O simulaci zápasů jedné generace se tedy může podělit více počítačů. Jeden počítač vždy zajišťuje samotný výpočet evoluce (tzv. server) a také se na něm ukládají všechna data. Ostatní pouze dostanou týmy v generaci, nastavení simulace a seznam zápasů, které mají odehrát, ty spočítají a výsledky odešlou zase zpět (tzv. workeři).

Samozřejmě jde výpočet provádět pouze na jednom počítači, stačí na něm spustit server i worker a komunikovat po loopbacku.

Distribuovaný výpočet je realizován primitivně, protože předpokládá stabilní síť s mnoha stejnými počítači, tedy třeba počítačovou laboratoř. Na počátku výpočtu se serveru zadají adresy běžících workerů. Server se ke všem připojí, začne výpočet a na jeho konci se ode všech workerů opět odpojí. Každý worker může přijímat práci pouze od jednoho serveru a během výpočtu se žádný worker nemůže připojit ani odpojit. Při odpojení jednoho workeru končí výpočet celé evoluce s chybou.

Práce mezi workery je distibuována co nejvíce rovnoměrně co se jejich počtu týče. Každý worker počítá v tolika vláknech počítače, kolik má procesorů. Není proto nutné pouštět více workerů na jednom počítači. Jak bylo řečeno, distribuce práce je vzhledem k počtu počítačů, ne jejich výkonu. Je tedy doporučeno pouštět workery na stejně výkonných počítačích. Přidání méně výkoného k výkonnějším může výpočet dokonce zpomalit!

Na konci simulace jsou na serveru uložena všechna data o počítané evoluci. Tato data se následně dají procházet a zpracovávat dalším programem.

V dalším textu jsou zmiňovány různé soubory, jejich popis je v kapitole 4.

Obrázky grafických aplikací jsou opatřeny čísly (bílá čísla v černém kolečku), na které je v textu odkazováno při popisu obsluhy programu.

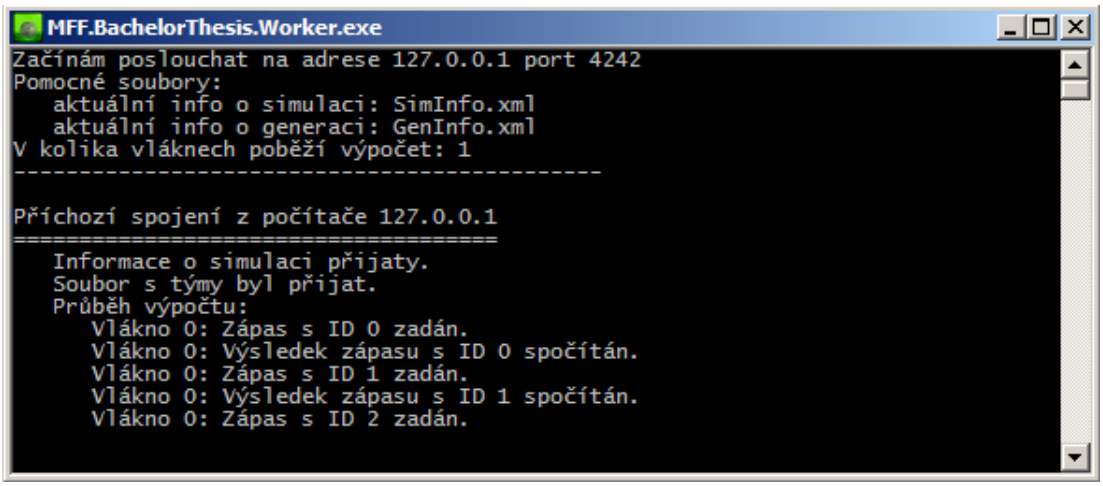
Programy jsou v příloze (7).

3.1.2 Spuštění workera

Worker je konzolový program. Pokud je vše v pořádku, začne po svém spuštění poslouchat na jistém portu jistého síťového rozhraní (podporuje pouze IPv4) a čekat na připojení serveru. O stavu vypisuje průběžně informace na výstup.

Program se může spustit bez parametrů, pokud se mu ovšem povede najít vhodné síťové rozhraní. Pokud ne, nebo pokud chce uživatel sám zadat IP či port, je možné vše uvést před spuštěním jako parametr. Předpokládaný formát parametrů vypíše sám program při spuštění s parametrem „--help“. Ukončení programu se provede pomocí Ctrl+C.

Worker si do aktuálního adresáře ukládá soubory obsahující nastavení simulace a aktuální generace. Zapisuje i log. Spuštění více workerů v jednom adresáři tak není možné.



Obrázek 3.1: Ukázka spuštěného workeru

Na obrázku 3.1 je spuštěný worker, ke kterému se již připojil server.

3.1.3 Provedení výpočtu evoluce

Server je na rozdíl od workera aplikace s grafickým rozhraním. Slouží k zadávání evolucí, které se mají počítat, seznamu workerů, adresáře, do kterého budou ukládány výsledky, a sledování průběhu výpočtu, případně jeho ukončení.

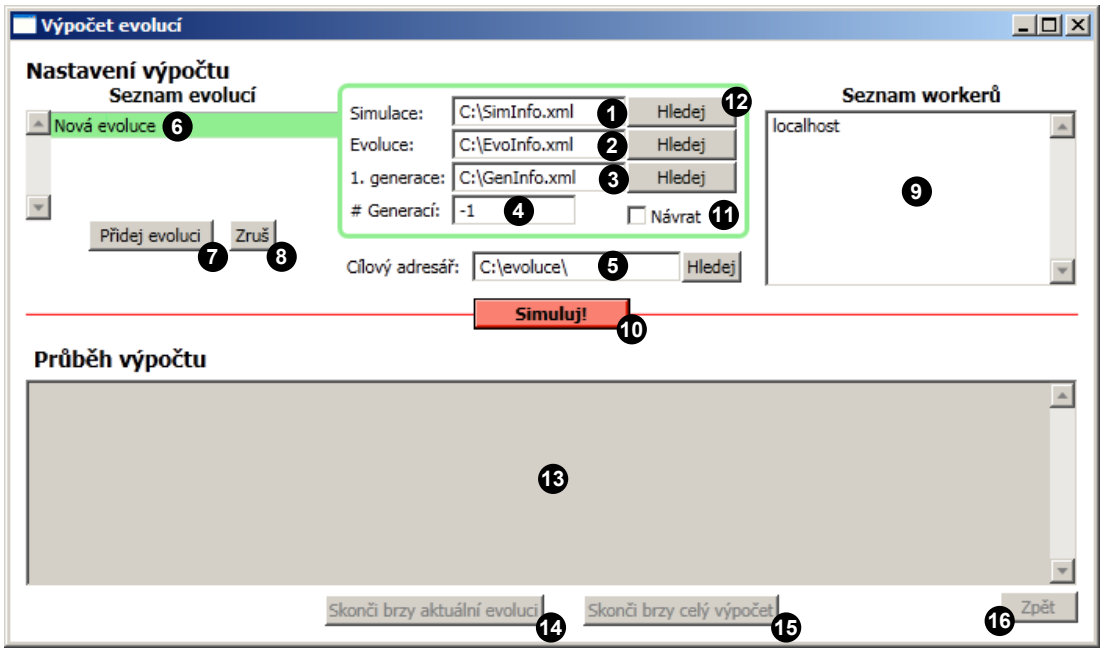
Na obrázku 3.2 je okno serveru s již vyplněnými údaji.

V okně jsou pole sloužící k zadávání cesty k souborům, které obsahují nastavení simulace (pole ❶), evoluce (pole ❷) a první generace (pole ❸). Do pole ❹ se zadává číslo cílové generace (tj. generace, jejíž zápasy se už nebudou simulovat a skončí výpočet), přičemž jediná povolená záporná hodnota je -1, která říká, že uživatel cílovou generaci zadat nechce a výpočet si přeje ukončit sám.

Do pole ❺ se zadává cesta k cílovému adresáři, do kterého budou umístěny výsledky evoluce. Pro každou evoluci bude vytvořen v cílovém adresáři podadresář s názvem evoluce, který jde nastavit v poli ❻. Může se stát, že zadaný název bude muset být upraven, aby řetězec obsahoval znaky povolené pro názvy adresářů, nebo podadresář se zadaným názvem už existuje, a proto se přidá za jméno evoluce podtržítko a nějaké číslo. Tlačítko ❼ umožňuje přidat do seznamu další evoluci, která se začne simulovat po té, co skončí výpočet předchozí. Mezi evolucemi se dá přepínat kliknutím na její název, ta vybraná zezelená a zobrazí se její aktuální nastavení. Naopak tlačítko ❽ odebírá aktuálně vybranou evoluci.

Do pole ❾ se vkládá seznam adres workerů, které se budou podílet na výpočtu. V jakém formátu se mají vkládat je napsáno v tool tipu tohoto pole. Všichni workeři už musí být spuštěni ve chvíli, kdy je zahájena simulace zadaných evolucí tlačítkem ❿.

Hodnoty polí ❶, ❷, ❸ a ❺ se dají editovat buď přímo, nebo se cesta může vybrat pomocí vestavěného dialogu Windows. Ten se otevře např. pomocí tlačítka ⓬, přitom každé z uvedených polí má své vlastní stejnojmenné tlačítko umístěné napravo od pole. Checkbox ⓫ slouží k navázání už jednou simulované evoluce. Aby se na výpočet evoluce mohlo navázat, musí být pole ❺ a ❻ vyplněny stejně jako při zakládání navazované evoluce a pole ❹ musí obsahovat číslo vyšší než je číslo poslední odsimulovaná generace (nebo -1). Obsah polí ❶, ❷, ❸ už není



Obrázek 3.2: Ukázka spuštěného serveru

potřeba zadávat.

Jak bylo řečeno, tlačítko 10 zahájí simulaci. Po jeho stisku bude aktivní spodní část formuláře, horní se naopak stane neaktivní.

V poli 13 se vypisují informace o průběhu výpočtu. Tlačítko 14 ukončí po té, co se dokončí výpočet aktuální generace, právě simulovanou evoluci. Tlačítko 15 udělá to samé a navíc se nezahájí výpočet žádné následující evoluce.

Po skončení výpočtu se lze „vrátit“ do horní části formuláře pomocí tlačítka 16 .

3.1.4 Prohlížení odsimulovaných evolucí

I prohlížeč odsimulovaných evolucí je grafická aplikace. Je znázorněna na obrázku 3.3.

Nejprve je třeba zapsat do pole 1 cestu k cílovému adresáři, který obsahuje evoluce, které chceme prohlížet. Cílový adresář můžeme také vybrat přes dialog Windows tlačítkem 2. Pokud byla cesta zadána ručně, je třeba ji potvrdit stiskem klávesy enter. Po té se v roletce 3 objeví všechny podadresáře z adresáře cílového, každý přitom odpovídá jedné odsimulované evoluci. Vybráním evoluce v roletce se údaje z ní zobrazí ve zbytku formuláře.

Tlačítky 4 a 5 se posouvá na předcházející, resp. nadcházející generaci. Číslo aktuální generace se zobrazuje v poli 6, kde jde také změnit (je nutné potvrdit enterem).

V tabulce 7 jsou zobrazeny statistiky týmů z aktuálně vybrané generace. Kliknutím na zkratku názvu statistiky (řádek vlevo dole od 8) se podle této statistiky týmy seřadí.

V seznamu 9 jsou uvedena všechna utkání, která se v dané generaci odehrála. Přitom pomocí roletky 10 se dají vyfiltrovat jen zápasy určitého týmu. Klik na utkání v seznamu 9 ho vybere (což je znázorněno zezelenáním) a jeho statistiky

se zobrazí v oblasti ⓫. Na vybraný zápas se jde také podívat kliknutím na tlačítko ⓮ (viz 3.1.5).

Pomocí menu ⓬ se dá uložit tabulka ❼ do souboru ve formátu CSV. Vynechány budou pouze sloupce s procentem v popisu a statistika čas u míče (ČUM) bude uložena v milisekundách.

Pomocí menu ⓭ se dá otevřít dialog pro vytvoření nové generace z týmů, které obsahuje tato evoluce. Dialog umožňuje zadat velikost shluku. Jeden shluk obsahuje všechny týmy z několika sousedících generací. Např. pokud má evoluce generace nula až šest a zadaná velikost shluku bude tři, ocitnou se všechny týmy z generací nula, jedna a dva v prvním shluku, druhý shluk pak budou tvořit všechny týmy z generací tři, čtyři a pět. Týmy ze šesté generace v žádném shluku nebudou, neboť všechny shluky musí být stejně velké. Z každého shluku je pak náhodně vybrán zadaný počet týmů. Lze nastavit, aby se z náhodně zvolené generace v shluku vybraly vždy nejlepší týmy, nebo celý výběr nechat náhodný. Vybrané týmy jsou uloženy do souboru ve formátu souboru o generaci (viz 4.4). Náhoda je řízena generátorem, jehož random seed se dá rovněž ve formuláři nastavit.

3.1.5 Prohlížeč zápasů

Okno prohlížeče zápasu je na obrázku 3.4. Otevírá se z prohlížeče odsimulovaných evolucí. Jeho smyslem je zpětně zobrazit zápas, který byl odsimulován během evoluce.

Zápas je nutné odstartovat tlačítkem ❶. Po té se na ploše ❷ začnou pohybovat hráči a míč. V horní části programu (oblast ❸) je časomíra a také názvy hrajících týmů. Domácí tým hraje zleva doprava a jeho hráči mají lososovou barvu, hosté hrají zprava doleva a jejich barvou je světle modrá. Míč je šedivý.

Aktuální statistiky ze zápasu se ukazují v tabulce ❹.

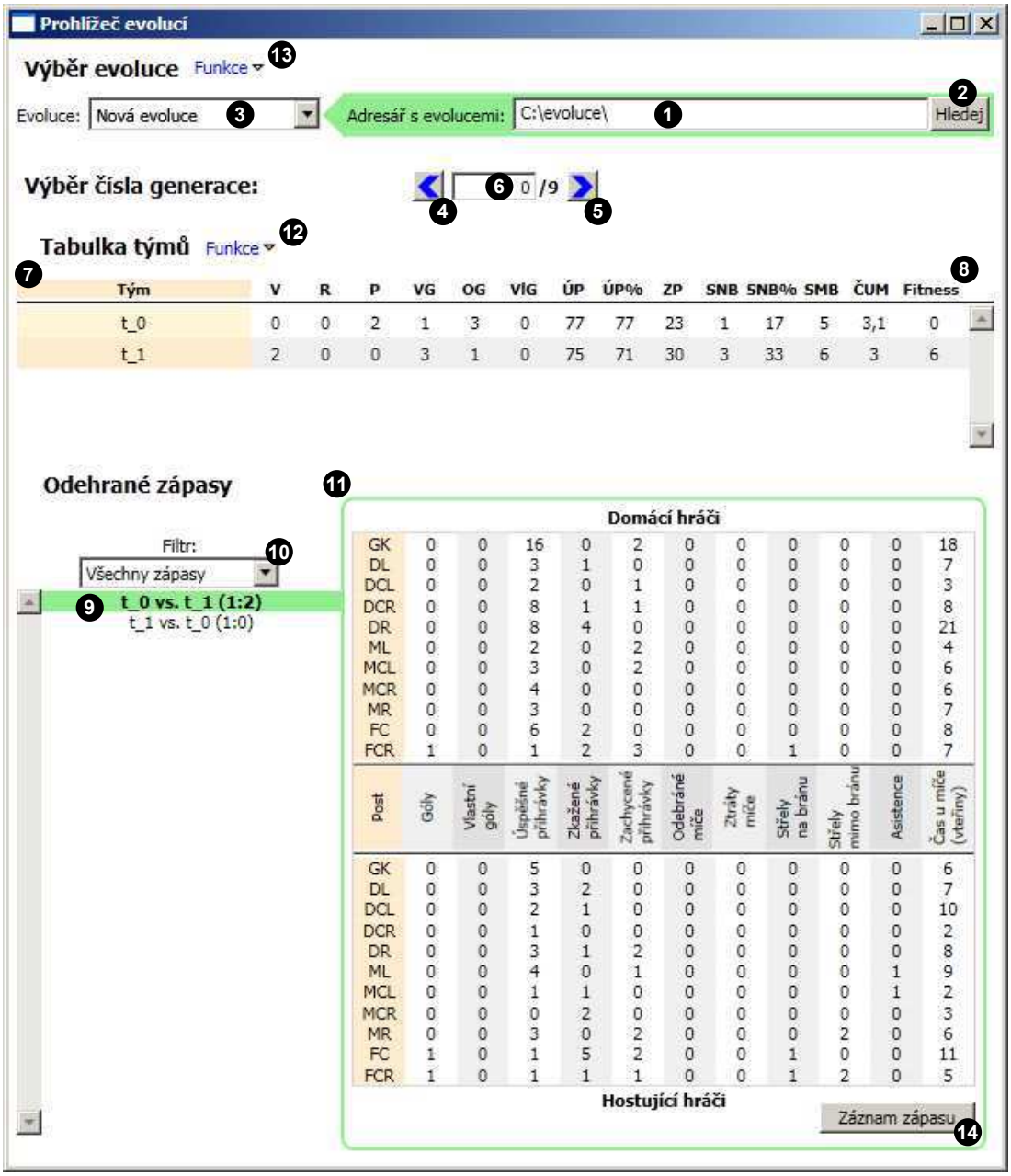
Utkání je možné kdykoli zapauzovat tlačítkem ❺, kterým se také odpauzuje. S jakou rychlostí se bude utkání zobrazovat lze ovlivnit posuvníkem ❻. Pokud je nejvíce vlevo, pak rychlost simulace odpovídá reálnému času. Posunem doprava se zrychluje, přičemž je možné, že od určité chvíle nebude další zrychlování možné kvůli rychlosti počítače.

Checkbox ❼ aktivuje zobrazování přihrávek. To způsobí, že před každou přihrávkou se hra sama zapauzuje a na hřišti se zobrazí mříž, která se počítá při výběru přihrávky. Obrázek 1.4 v části 1.4.5 je pořízen právě díky této volbě.

Checkbox ❽ pak aktivuje zobrazování sil, které hráče ovlivňují při jejich pohybu. Síly se zobrazují různými barvami, legenda je v tabulce ⓾. Do pole ❾ se vpisuje číslo, kterým se mají násobit zobrazené vektory sil. Pro různá nastavení se totiž může stát, že budou zobrazené vektory moc dlouhé, nebo naopak moc krátké, proto je lze tímto upravit tak, aby bylo jejich zobrazení vyhovující.

3.2 Adresář evoluce

V adresáři evoluce musí být vždy soubor s nastavením simulace, který se jmenuje „SimInfo.xml“, a evoluce, který se jmenuje „EvoInfo.xml“. Dále se pro každou generaci uchovává soubor s názvem např. „generation_00000000.xml“ pro první



Obrázek 3.3: Ukázka spuštěného prohlížeče evolucí

generaci, „generation_00000001.xml“ pro druhou, atp. Pro každou odsimulovanou generaci, což je každá až na tu poslední, se navíc uchovává soubor s výsledky evoluce, jež se pro první generaci jmenuje „generation_00000000.results.xml“, pro druhou „generation_00000001.results.xml“, atp. Tyto soubory jsou popsány v kapitole 4.

Při výpočtu se navíc do textového souboru „computation.log“ ukládá log výpočtu evoluce, který se zobrazuje i v serveru. Tento soubor ale není nezbytný a může se měnit či mazat.

Změna souborů vyjma logu není doporučena. V lepším případě program při prohlížení pouze napíše, že data nesouhlasí s těmi v době simulace, v horším odmítne generaci, nebo rovnou celou evoluci, načíst.

Obrázek 3.4: Ukázka spuštěného prohlížeče zápasů

3.3 Implementace

Celý program je napsán v jazyce C# a pro svůj běh vyžaduje Microsoft .NET Framework 4 a operační systém Windows. Pro tvorbu grafického prostředí byla využita knihovna Windows Presentation Foundation. Kód je organizován do projektů vývojového prostředí Microsoft Visual Studio 2010 Express.

Program využívá knihovnu JiM.MyLib, která vznikala paralelně s hlavní částí a obsahuje kód, který je obecnější a dá se použít i v jiných projektech. Ovšem ne úplně všechno z této knihovny je v hlavní části využito.

Hlavní část tvoří šest projektů - tři knihovny a tři programy.

3.3.1 Knihovna JiM.MyLib

Kód knihovny JiM.MyLib je v příloze (2). Některé části knihovny jsou pokryty testy, které využívají knihovnu NUnit verze 2.5.8 (viz [3]). NUnit je potřeba pro zkompilování knihovny JiM.MyLib. Dokumentace ke knihovně je v příloze (3).

V knihovně se vyskytuje cizí kód, konkrétně v souborech (cesta je vzhledem ke kořenovému adresáři knihovny):

JiM.MyLib/Algorithms/HungarianAlgorithm.cs
Soubor obsahuje implementaci maďarského algoritmu (viz [2]). Implementace pochází z [4].

JiM.MyLib/Various/GaussianRandom.cs
Soubor obsahuje implementaci náhodného generátoru s normálním rozdělením. Pochází ze stránky [5].

3.3.2 Hlavní část

Hlavní část je v příloze (4). Z historických důvodů (byla převzata část staršího projektu) je část kódu (identifikátory, komentáře) v češtině, část v angličtině a pochopitelně tam, kde se využívají navzájem, jsou jazyky oba. Dokumentace je v příloze (5). Jak bylo zmíněno, je to více projektů:

MFF.BachelorThesis.Evolution
Knihovna, která obsahuje převážně třídy, které souvisí s evolucí.

MFF.BachelorThesis.GlobalData
Knihovna, která poskytuje ostatním částem údaje, které jsou společné více projektům (čísla portů, názvy souborů).

MFF.BachelorThesis.ResultsViewer
Program pro prohlížení odsimulovaných evolucí. Sekce 3.1.4 a 3.1.5 rozebírají ovládání tohoto programu.

MFF.BachelorThesis.Server
Program pro zadávání a následnou simulaci evolucí. O ovládání hovoří sekce 3.1.3.

MFF.BachelorThesis.Simulation
Knihovna pro simulaci a zobrazování zápasu.

MFF.BachelorThesis.Worker
Program simuluje zápasy, které mu přijdou od serveru. Popis programu je v sekci 3.1.2.

4. Formáty souborů

4.1 Pravidla pro všechny soubory

Vstupní soubory jsou ve formátu XML verze 1.0. Navíc tato specifikace vyžaduje, aby bylo dodržováno uvedené pořadí elementů a ani není možné vynechat některý element či atribut, pokud není výslovně uvedeno jinak. Soubory musí být v kódování utf-8.

U každého souboru je uveden použitelný vzor, který specifikuje názvy elementů a jejich vzájemnou strukturu. Tučně je vždy zapsána nějaká možná konkrétní hodnota. Elementy jsou dále popsány ve stejném pořadí jako ve vzoru, pro jasnou identifikaci či lepší přehled může být k elementu připojen zleva název elementů bezprostředně nadřazených, oddělený šipkou.

Následuje výčet pravidel pro zapisování různých typů hodnot.

Číslo Pokud je uvedeno, že obsahem elementu má být číslo, musí být toto číslo zapsáno vcelku bez jakýchkoliv oddělovačů (např. ne s mezerou mezi řády). Reálné číslo se zapisuje vždy s desetinnou tečkou, na místě reálného čísla lze uvést i číslo celé (tj. nemusí se zapisovat (nulová) desetinná místa ani tečka). Když je uvedeno, že číslo má být z jistého rozsahu hodnot, myslí se vždy včetně krajních mezí.

N-tice N-tice se zapisuje jako $(p_1, \ldots, p_n)$, kde p_i jsou prvky n-tice. Počet prvků a jejich typ bude uveden u každého elementu. Více n-tic se odděluje čárkou. Bílé znaky je povoleno psát před i za závorkou a čárkou.

Výraz Výrazem je myšlen infixový matematický výraz, který se skládá z konstant (pro ně platí stejná pravidla jako pro zmíněný typ číslo), proměnných (jejich názvy budou uvedeny, na velikosti písmen vždy záleží), operátorů plus (+), krát (*), děleno (/) a kulatých závorek. Pravidla pro zápis výrazu se neliší od běžných zvyklostí. Mezi operátory a závorky je možné vkládat libovolné množství bílých znaků.

Řetězec Kompletní výčet možných řetězců bude u elementu vždy uveden. Jejich zápis musí dodržovat velká a malá písmena.

4.2 Simulace

Soubor simulace obsahuje nastavení některých vlastností simulace, které budou použity ve všech simulacích během celé evoluce. V příloze (1) je příklad tohoto souboru pod názvem „SimInfo.xml“, což je název, který se používá i na některých místech v programu.

4.2.1 Vzor

```
<?xml version="1.0" encoding="utf-8"?>
<simulace>
    <mic>
```

```
        <zpomaleni>4</zpomaleni>
        <vzdalenostProKontrolu>1.5</vzdalenostProKontrolu>
    </mic>
    <strela>
      <rychlost>42</rychlost>
      <odchylka>5</odchylka>
   </strela>
   <prihravka>
      <rychlost>20</rychlost>
      <pocetPolicek>
          <radku>12</radku>
          <sloupcu>16</sloupcu>
      </pocetPolicek>
      <odchylka>1</odchylka>
   </prihravka>
   <rozhodovani>
      <pocetPolicek>
         <radku>6</radku>
         <sloupcu>4</sloupcu>
      </pocetPolicek>
   </rozhodovani>
   <hraci>
      <nejvyssiRychlost>6</nejvyssiRychlost>
      <volnyHracPolomer>1.5</volnyHracPolomer>
   </hraci>
</simulace>
```

4.2.2 Popis elementů

mic → zpomaleni

Zpomalení míče v jednotkách metrů za sekundu. Možné hodnoty jsou reálná čísla z rozsahu od 0 do 80.

mic → vzdalenostProKontrolu

Maximální vzdálenost, která může dělit hráče a míč, aby mohl hráč do míče kopnout a zároveň vzdálenost, kdy se bere, že má hráč míč pod kontrolou. Možné hodnoty jsou reálná čísla z rozsahu od 0 do 2.

strela → rychlost

Jakou rychlost v jednotkách metrů za sekundu udělí hráč míči při střele. Možné hodnoty jsou reálná čísla z rozsahu od 10 do 80.

strela → odchylka

Až o kolik stupňů (na obě strany) se může fotbalista splést v zamýšleném směru střely. Možné hodnoty jsou reálná čísla z rozsahu od 0 do 90.

prihravka → rychlost

Jakou rychlost v jednotkách metrů za sekundu udělí hráč míči při přihrávce. Možné hodnoty jsou reálná čísla z rozsahu od 10 do 80.

prihravka → pocetPolicek → radku

Kolik řádků bude mít mříž, která rozdělí hřiště (orientované naležato) při hledání přihrávky. Možné hodnoty jsou celá čísla z rozsahu od 2 do 20.

prihravka → pocetPolicek → sloupcu
Kolik sloupců bude mít mříž, která rozdělí hřiště (orientované naležato) při hledání přihrávky. Možné hodnoty jsou celá čísla z rozsahu od 2 do 30.

prihravka → odchylka
Až o kolik stupňů (na obě strany) se může fotbalista splést v zamýšleném směru přihrávky. Možné hodnoty jsou reálná čísla z rozsahu od 0 do 90.

rozhodovani → pocetPolicek → radku
Kolik řádků má mříž, která rozdělí hřiště (orientované naležato) a určuje, které parametry budou vybrány pro pohyb a kopání do míče. Možné hodnoty jsou celá čísla z rozsahu od 1 do 10.

rozhodovani → pocetPolicek → sloupcu
Kolik sloupců má mříž, která rozdělí hřiště (orientované naležato) a určuje, které parametry budou vybrány pro pohyb a kopání do míče. Možné hodnoty jsou celá čísla z rozsahu od 1 do 10.

hraci → nejvyssiRychlost
Nejvyšší možná rychlost hráče v jednotkách metrů za sekundu. Možné hodnoty jsou reálná čísla z rozsahu od 1 do 9.

hraci → volnyHracPolomer
Kolik metrů od hráče nesmí být žádný hráč soupeře, aby mohl být tento hráč považován za volného. Možné hodnoty jsou reálná čísla z rozsahu od 1 do 3.

4.3 Evoluce

Soubor evoluce obsahuje nastavení některých vlastností evoluce. V příloze (1) je příklad tohoto souboru pod názvem „EvoInfo.xml“, což je název, který se používá i na některých místech v programu.

4.3.1 Vzor

```
<?xml version="1.0" encoding="utf-8"?>
<evoluce>
    <hraniZapasu>Liga</hraniZapasu>
    <pravdepodobnostMutace>
        <vlastnosti>0.001</vlastnosti>
        <pozice>0.0002</pozice>
    </pravdepodobnostMutace>
    <silaMutace>0.001</silaMutace>
    <vypocetFitness>
        <tymy>3*V+R</tymy>
        <hraci>VG+OM+ZaP</hraci>
    </vypocetFitness>
    <krizeniPostu>HracFitness</krizeniPostu>
</evoluce>
```

4.3.2 Popis elementů

hraniZapasu
Jakým způsobem mezi sebou hrají týmy svá utkání. Možnosti jsou: Liga, KazdySKazdym (vysvětlují se v sekci 2.2).

pravdepodobnostMutace → vlastnosti
Pravděpodobnost, že bude mutován jeden gen jednoho hráče týkající se pohybu nebo kopání do míče. Možné hodnoty jsou reálná čísla z rozsahu od 0 do 1.

pravdepodobnostMutace → pozice
Pravděpodobnost, že bude mutována jedna pozice jednoho hráče. Možné hodnoty jsou reálná čísla z rozsahu od 0 do 1.

silaMutace
Rozptyl normálního rozdělení, které určuje krok mutace. Možné hodnoty jsou reálná čísla z rozsahu od 0,000001 do 1.

vypocetFitness → tymy
Výraz pro výpočet fitness týmu. Použitelné proměnné: VG (vstřelené góly), OG (obdržené góly), V (vítězství), R (remízy), P (prohry), VlG (vlastní góly), UP (úspěšné přihrávky), ZP (zkažené přihrávky), SNB (střely na bránu), SMB (střely mimo bránu), C (čas u míče (v minutách)).

vypocetFitness → hraci
Výraz pro výpočet fitness hráče. Použitelné proměnné: G (góly), VG (vlastní góly), UP (úspěšné přihrávky), ZkP (zkažené přihrávky), ZaP (zachycené přihrávky), OM (odebrané míče), ZM (ztráty míče), SNB (střely na bránu), SMB (střely mimo bránu), A (asistence), C (čas u míče (v minutách)).

krizeniPostu
Jaký post dostane hráč, který vznikl z křížení hráčů na různých postech. Možnosti jsou: HracFitness, TymFitness, HracNahodne, TymNahodne (o významu je pojednáno v 2.3, vyjmenováno ve stejném pořadí).

4.4 Generace

Soubor generace obsahuje definici jedné generace, tedy všechny její týmy (tj. hráče) a je zde i random seed pro tuto generaci. V příloze (1) je příklad tohoto souboru pod názvem „GenInfo.xml“, který se používá i na některých místech v programu.

4.4.1 Neúplný vzor

Vzor je zkrácen, neboť i nejkratší možná varianta by byla dlouhá z důvodu velkého množství opakujících se elementů se stejnou strukturou. Místa, kde se musí něco doplnit, jsou vyznačena XML komentářem.

Uvedená struktura předpokládá, že v souboru, který obsahuje nastavení simulace, obsahují elementy **simulace → rozhodovani → pocetPolicek → radku** a **simulace → rozhodovani → pocetPolicek → sloupcu** oba jedničku.

```
<?xml version="1.0" encoding="utf-8"?>
<generace>
 <randomSeed>0</randomSeed>
 <tymy>
  <sestavaTymu nazev="První tým">
   <hrac>
    <post>GK</post>
    <rozhodovani>
     <prohravame>
      <tymBezMice>
       (9.7,0,2.2),(35,0,1.8),(600,0,1.5),
       (108,1.6,3.3),(146,1.6,3.2),(0.77,0,4),(100,0,1)
      </tymBezMice>
      <hracBezMice>
       (9.7,0,2.2),(35,0,1.8),(600,0,1.5),
       (108,1.6,3.3),(146,1.6,3.2),(0.77,0,4),(100,0,1)
      </hracBezMice>
      <hraniMice>
       <nejsemVolny>
        <pohyb>
         (9.7,0,2.2),(35,0,1.8),(600,0,1.5),
         (108,1.6,3.3),(146,1.6,3.2),(0.77,0,4),(100,0,1)
        </pohyb>
        <prihravka>(0.7,0.6,1)</prihravka>
        <vynikajiciStrela>1</vynikajiciStrela>
        <vynikajiciPrihravka>(1,0.4,2)</vynikajiciPrihravka>
        <pravdepodobnosti>(30,30,40)</pravdepodobnosti>
       </nejsemVolny>
       <pozice radek="1" sloupec="1">
        <pohyb>
         (9.7,0,2.2),(35,0,1.8),(600,0,1.5),
         (108,1.6,3.3),(146,1.6,3.2),(0.77,0,4),(1.1,0,0.45)
        </pohyb>
        <prihravka>(0.7,0.6,1)</prihravka>
        <vynikajiciStrela>0.85</vynikajiciStrela>
        <vynikajiciPrihravka>(1,0.4,2)</vynikajiciPrihravka>
        <pravdepodobnosti>(30,30,40)</pravdepodobnosti>
       </pozice>
      </hraniMice>
     </prohravame>
     <neprohravame>
      <!--
       Znovu ta samá struktura jako v elementu "prohravame".
      -->
     </neprohravame>
    </rozhodovani>
   </hrac>
   <!--
    Dalších deset elementů "hrac" se stejnou strukturou.
    Všichni hráči musí mít navzájem různý post.
   -->
  </sestavaTymu>
  <!--
   Ještě alespoň jednou element "sestavaTymu"
   se stejnou strukturou.
  -->
 </tymy>
```

```
</generace>
```

4.4.2 Popis elementů

Platí, že název elementu vždy jednoznačně určuje jeho strukturu. Tedy např. elementy **nejsemVolny → pohyb** a **pozice → pohyb** mají stejnou strukturu. Níže je tedy popsána vždy jenom jednou.

randomSeed
: Seed pro náhodný generátor, který určuje výsledky zápasů v této generaci a tvorbu té následující. Možné hodnoty jsou celá čísla z rozsahu od -2147483648 do 2147483647.

tymy → sestavaTymu
: Tento element musí být uveden uvnitř elementu **tymy** alespoň dvakrát. Atribut **nazev** je nepovinný, slouží spíše pro případy, kdy se z programu nechává vygenerovat nová generace z více hotových generací. V takovém případě je tento atribut použit pro zpětnou identifikaci týmu (název obsahuje informaci o číslu generace a pořadí v ní).

hrac → post
: Obsahuje kód postu hráče. V každém týmu může hrát na jednou postu nejvýše jeden hráč. Možné hodnoty jsou: GK, DR, DCR, DC, DCL, DL, DMR, DMCR, DMC, DMCL, DML, MR, MCR, MC, MCL, ML, AMR, AMCR, AMC, AMCL, AML, FR, FCR, FC, FCL, FL (více o těchto možnostech v sekci 1.4.1).

tymBezMice, hracBezMice, pohyb
: Obsahuje parametry pro výpočet délky sedmi vektorů, které určují výsledný směr pohybu hráče. Obsahem elementu je tedy sedm trojic. Trojice jsou v pořadí: pomezní čáry, branková čára, míč, spoluhráč, protihráč, hranice ofsajdu, pozice. Parametry v trojici jsou v pořadí N, S, M. Možné hodnoty parametrů jsou reálná čísla, kde hodnoty N jsou z rozsahu od 0,1 do 1000, S z rozsahu od 0 do 10, M z rozsahu od 0,1 do 10.

prihravka, vynikajiciPrihravka
: Obsahuje parametry pro výběr přihrávky, resp. specifikování parametrů pro přihrávku vynikající. Obsahem je trojice. Na prvním místě je hranice rizika z reálného rozsahu hodnot od 0 do 1. Druhá je minimální šance z reálného rozsahu hodnot od 0 do 1. Poslední je minimální posun vpřed z celočíselného rozsahu od -20 do 20.

vynikajiciStrela
: Parametr, který specifikuje hranici vynikající střely. Možné hodnoty jsou reálná čísla z rozsahu od 0 do 1.

pravdepodobnosti
: Pravděpodobnosti tří možných kopů, které hráč vykoná, pokud neexistuje

ani vynikající přihrávka, ani vynikající střela. Obsahem je trojice pravděpodobností v procentech. První je pravděpodobnost přihrávky, druhá pravděpodobnost střely, poslední je pravděpodobnost driblinku. Všechny tři pravděpodobnosti jsou z celočíselného rozsahu od 0 do 100, celkový součet všech tří pravděpodobností ale musí dát 100.

pozice
Tento element musí být spolu se svým obsahem zopakován tolikrát, kolikrát to udávají v souboru, který obsahuje nastavení simulace, elementy **simulace → rozhodovani → pocetPolicek → radku** a **simulace → rozhodovani → pocetPolicek → sloupcu**. Pro každou dvojici řádek-sloupec (oba začínají své číslování jedničkou a vzhledem k hřišti orientovaném naležato s domácí brankou vlevo, řádek 1 označuje horní řádek, sloupec 1 pak levý) musí být uveden element pozice s odpovídajícím číslem řádku a sloupce zapsaným v atributu. Pořadí elementů vzhledem k číslování řádků a sloupců je libovolné.

4.5 Výsledek evoluce

Soubor s výsledky evoluce je vytvářen i čten pouze programem, uživatel jeho strukturu vůbec nemusí znát. Slouží k uložení výsledků evoluce, aby se k nim dalo snadno vrátit při zpětném prohlížení.

4.5.1 Příklad

Nejmenší možný příklad souboru, který uchovává výsledky jednoho utkání.

```
<?xml version="1.0" encoding="utf-8"?>
<odehraneZapasy
    hash="42212c81a2a48246a06fa649bd6af019"
    simInfoHash="9c8051a228a141c646230be7c1829f8f"
    genInfoHash="e9a95873ae6d365cec5917e8c8115a7e"
>
  <zapas randomSeed="1755192844" domaci="1" hoste="0">
    <domaciHraci>
      <hrac0>0,0,0,0,0,0,0,0,0,0,4650</hrac0>
      <hrac1>0,0,0,0,0,0,0,0,1,0,4900</hrac1>
      <hrac2>0,0,0,0,0,0,0,0,0,0,1350</hrac2>
      <hrac3>0,0,0,0,0,0,0,0,0,0,3100</hrac3>
      <hrac4>0,0,0,0,0,0,0,0,0,0,2950</hrac4>
      <hrac5>0,0,0,0,0,0,0,0,0,0,2750</hrac5>
      <hrac6>0,0,0,0,0,0,0,0,1,0,2600</hrac6>
      <hrac7>0,0,0,0,0,0,0,0,1,0,5450</hrac7>
      <hrac8>0,0,0,0,0,0,0,0,1,0,10150</hrac8>
      <hrac9>0,0,0,0,0,0,0,0,4,0,2350</hrac9>
      <hrac10>0,0,0,0,0,0,0,0,2,0,4250</hrac10>
    </domacihraci>
    <hostujicihraci>
      <hrac0>0,0,0,0,0,0,0,0,0,0,6150</hrac0>
      <hrac1>0,0,0,0,0,0,0,0,0,0,4300</hrac1>
      <hrac2>0,0,0,0,0,0,0,0,0,0,2000</hrac2>
      <hrac3>0,0,0,0,0,0,0,0,0,0,3150</hrac3>
      <hrac4>0,0,0,0,0,0,0,0,1,0,8050</hrac4>
```

```
      <hrac5>0,0,0,0,0,0,0,0,3,0,6950</hrac5>
      <hrac6>0,0,0,0,0,0,0,0,0,0,3000</hrac6>
      <hrac7>0,0,0,0,0,0,0,0,0,0,1400</hrac7>
      <hrac8>0,0,0,0,0,0,0,0,0,0,8700</hrac8>
      <hrac9>0,0,0,0,0,0,0,0,4,0,3000</hrac9>
      <hrac10>0,0,0,0,0,0,0,0,2,0,2250</hrac10>
    </hostujiciHraci>
  </zapas>
</odehraneZapasy>
```

4.5.2 Popis elementů

odehraneZapasy

Element má jako atributy hash kódy obsahu tří souborů. Díky nim se při prohlížení evoluce může zkontrolovat, zda jsou všechna nastavení a údaje stejné jako v době simulace. První atribut **hash** ukládá hash kód obsahu tohoto souboru. Atributy **simInfoHash** a **genInfoHash** pak hash kódy obsahů souborů o nastavení simulace a evoluce.

zapas

Obsahem elementu jsou zápasové statistiky jednotlivých hráčů obou týmů. Zápasy jsou ukládány v pořadí, v jakém byly odehrány. Atribut **randomSeed** ukládá random seed použitý pro náhodný generátor simulace. Atributy **domaci** a **hoste** pak obsahují pořadí (začínající nulou) týmu v souboru o generaci.

hrac0, hrac1, . . . , hrac10

Element obsahuje statistiky hráče oddělených čárkami. Hráči jsou uloženi v pořadí, v jakém byli definování v souboru o generaci. Pořadí statistik je následující: góly, vlastní góly, úspěšné přihrávky, zkažené přihrávky, zachycené přihrávky, odebrané míče, ztráty míče, střely na bránu, střely mimo bránu, asistence, čas u míče (v milisekundách).

5. Pokusy

5.1 Úvod

V této kapitole jsou popsány některé evoluce, které byly zadány do programu.

Styl zápisu elementů ve vstupních souborech je stejný jako v kapitole 4.

Všechny zmiňované evoluce jsou v příloze (6). Po jejich vygenerování bylo potřeba opravit v programu menší chybu. Proto stávající verze programu nevygeneruje ze stejného počátečního nastavení evoluce stejné průběhy jako ty uložené v příloze, i když by to tak být mělo. Změna byla ale pouze malá a nemá významný vliv na prezentované výsledky.

5.2 Generování náhodné generace

Níže popsaná evoluce má technický význam. Slouží pouze k tomu, aby program vygeneroval náhodnou úvodní generaci. Evoluci, pro kterou se úvodní generace generuje, nazvěme pro zkrácení jako evoluci cílovou.

Nastavení simulace je možné jakékoli, pouze elementy **rozhodovani → pocetPolicek → radku** a **rozhodovani → pocetPolicek → sloupcu** musí být pochopitelně stejné jako v cílové evoluci.

Do úvodní generace stačí nakopírovat (třeba pomocí textového editoru) jeden tým tolikrát, kolik týmů má být v cílové generaci.

Evoluci nastavíme následovně:

```
<pravdepodobnostMutace>
    <vlastnosti>1</vlastnosti>
    <pozice>1</pozice>
</pravdepodobnostMutace>
<silaMutace>1</silaMutace>
```

Tím se zajistí, že každý gen vlastností i pozic hráčů se po každé generaci silně zmutuje. Pro „vypnutí“ evoluce se nastaví obě fitness na nějakou konstantní hodnotu, čímž se nebude vybírat do další generace podle výsledků týmů:

```
<vypocetFitness>
    <tymy>1</tymy>
    <hraci>1</hraci>
</vypocetFitness>
```

Hraní zápasů tedy není ani potřeba, proto se může nastavit systém hraní zápasu tak, aby se utkáních hrálo co nejméně, čímž se ušetří čas:

```
<hraniZapasu>KazdySKazdym</hraniZapasu>
```

Nyní se spustí evoluce. V příkladu se nechalo běžet deset generací, ovšem teoreticky by mohla stačit i generace jedna. Poslední generace této evoluce je pak ona náhodná první generace pro evoluci cílovou.

5.3 První evoluce

Pokud bychom měli analyzovat její průběh a snažili bychom se zjistit, zda se týmy v průběhu generací zlepšovaly, není možné to udělat přímočarým porovnáním fitness napříč generacemi. Fitness je objektivním měřítkem kvality týmu pouze vůči jinému týmu z té samé generace, protože výsledky týmu závisí nejen na něm, ale i na jeho soupeřích. Dobrý tým mezi špatnými vyhrává snadněji než tým, který je sice lepší, ale má také lepší soupeře.

Aby se tedy daly týmy porovnávat napříč generacemi, musí mezi sebou sehrát utkání. Vyvstává ovšem problém, neboť nechat si mezi sebou zahrát všechny týmy, které během evoluce vznikly, by trvalo příliš dlouho. Proto jsem zvolil metodu výběru zástupců, kteří budou reprezentovat určitý úsek evoluce.

Tato evoluce má 3051 generací, každá po deseti týmech. Vždy 50 generací utvořilo shluk a z něj byl vybrán jeden tým, takže závěrečného klání se účastnilo 61 týmů. Hodnocení týmů bylo „klasické fotbalové“, tedy tři body za vítězství, jeden za remízu. Maximální možný bodový zisk tedy činí 183 bodů. Je otázkou, jak týmy vybírat. Použil jsem dva přístupy. První vybírá zcela náhodně, druhý pak náhodně vybere generaci (musí, žádné jasné vodítko, která generace „je lepší“, není) a z této generace vybere tým s nejvyšší fitness. Oba způsoby výběru jsem přitom provedl třikrát.

Na obrázku 5.1 je srovnání tří náhodných výběrů. Na ose X je pořadí shluku, z kterého pochází zástupce (shluky jsou řazeny „chronologicky“ - ve shluku číslo jedna je tým vybraný z některé z prvních padesáti generací, ve shluku 61 naopak z posledních padesáti). Na ose Y je pak jeho úspěšnost. Vybíral jsem třikrát, tedy u každého shluku jsou tři sloupce. Pochopitelně zápasy mezi sebou hrály pouze týmy „ze stejnobarevných sloupců“.

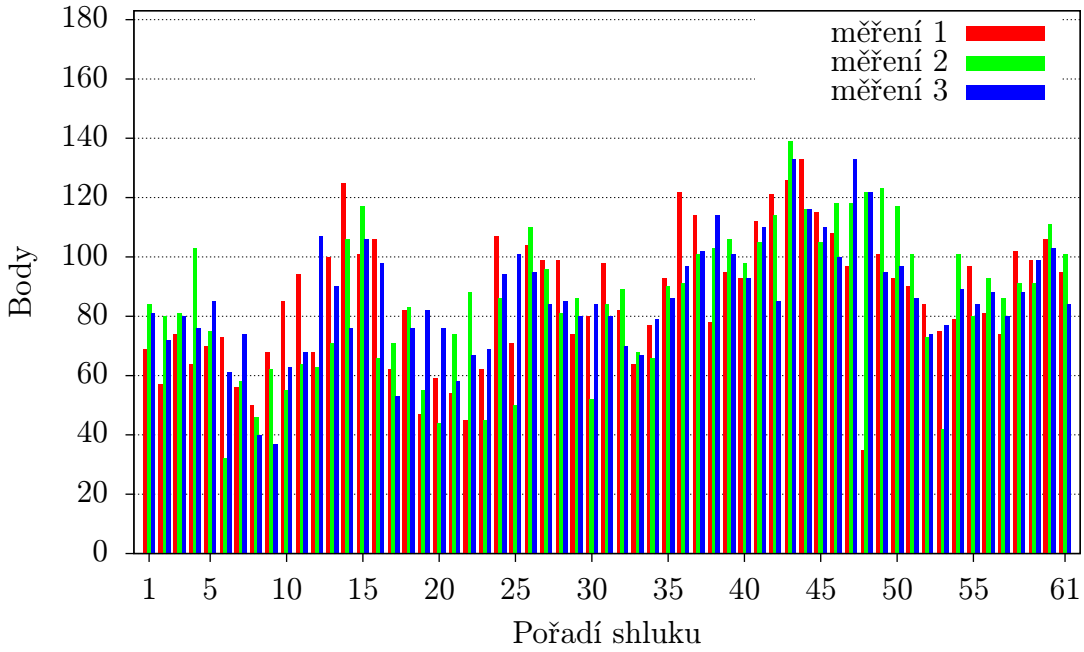
To samé zobrazuje i obrázek 5.2, pouze se zde vybíraly nejlepší týmy z náhodně vybrané generace.

Z obrázků se zdá, že u stejných shluků jsou všechny tři sloupečky většinou přibližně stejně vysoké. Z toho usuzuji, že se jednotlivé generace zlepšovaly tak, že rostla i průměrná kvalita jejich týmů, protože v tom případě by tolik nezáleželo na tom, jaký tým byl vybrán - vůči ostatním by si vedly zhruba stejně.

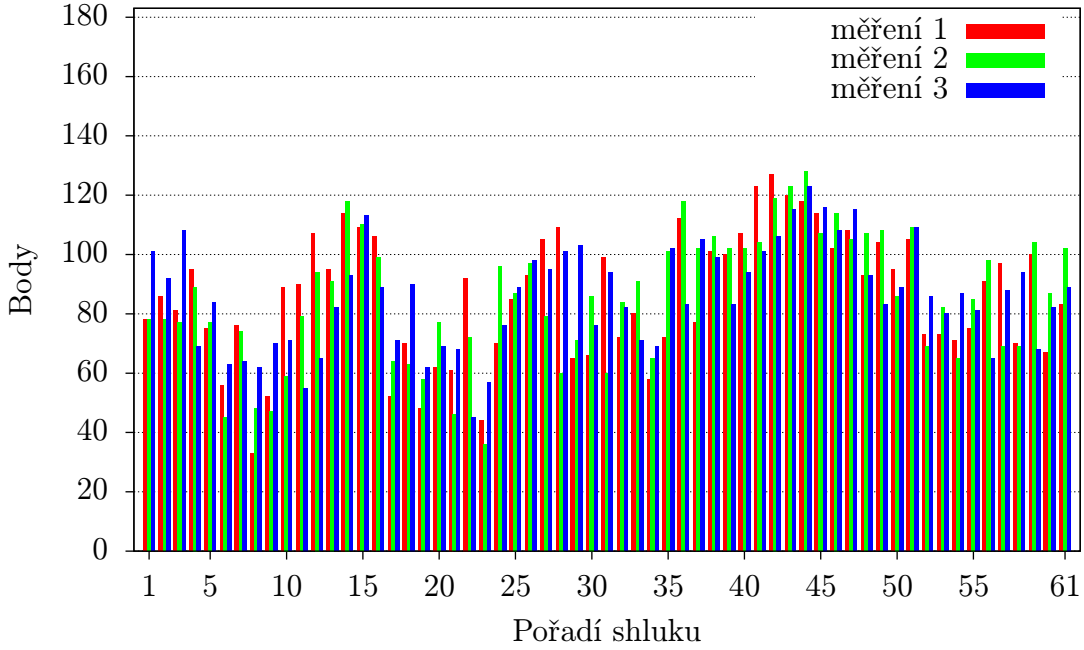
Protože se tedy z obrázků zdá, že se výběry provedené stejným způsobem tolik neliší, rozhodl jsem se body týmů ze stejných shluků zprůměrovat. Průměr se mi ze všech možností (např. minimum, maximum, medián) líbil nejvíce, i když pro něj nemám zvláštní důvod. Obrázek 5.3 porovnává oba výběry. Graf vyjadřuje to samé co dva předchozí, pouze nyní se porovnávají zprůměrované výběry.

Opět to z grafů vypadá, že si týmy ze stejných shluků vedly přibližně stejně. Na jejich základě tedy soudím, že obě metody výběru týmů jsou stejně vhodné pro analýzu evoluce. Poslední obrázek navíc napovídá, že se týmy zlepšují, byť pozvolna a ne přímočaře. Pro lepší znázornění trendu evoluce jsou na obrázku 5.3 vykresleny klouzavé průměry z posledních dvaceti bodů, jeden pro zprůměrované náhodné výběry, jeden pro zprůměrované výběry nejlepších týmů. I ty napovídají, že ke zlepšování dochází.

Technická realizace této analýzy je následující. Oba způsoby výběru nabízí program na prohlížení odsimulovaných evolucí (viz 3.1.4). Vybrané týmy jsou uloženy do souboru ve formátu, který může být použit jako první generace evoluce. Aby si mezi sebou týmy zahrály, stačí spustit evoluci o jedné generaci.



Obrázek 5.1: Srovnání výsledků první evoluce - tři výběry náhodných týmů.

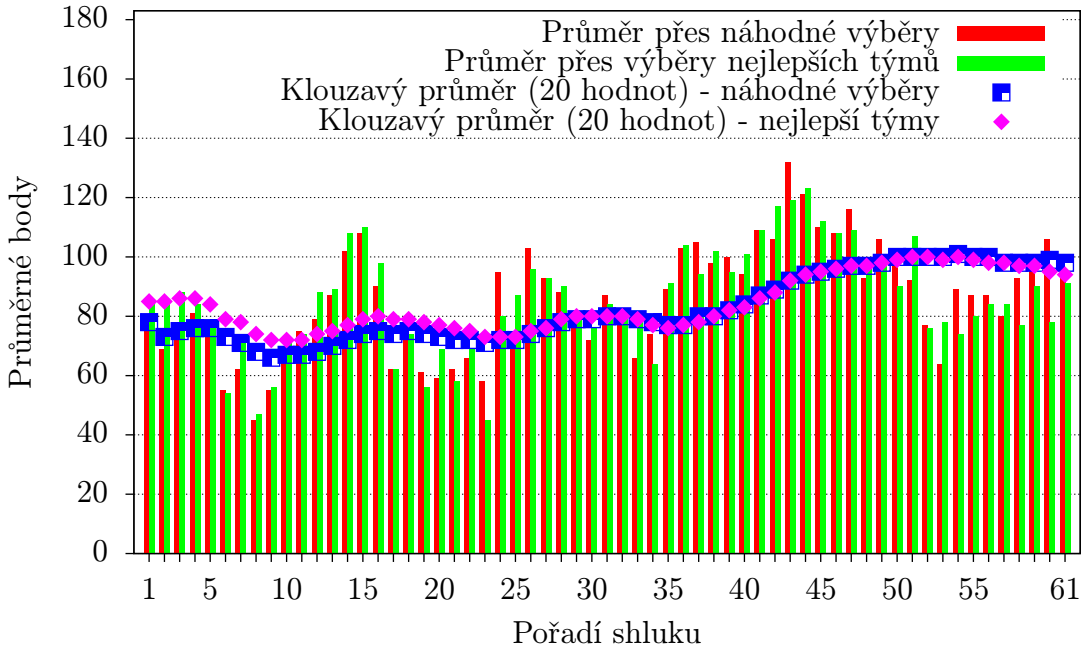


Obrázek 5.2: Srovnání výsledků první evoluce - tři výběry nejlepších týmů.

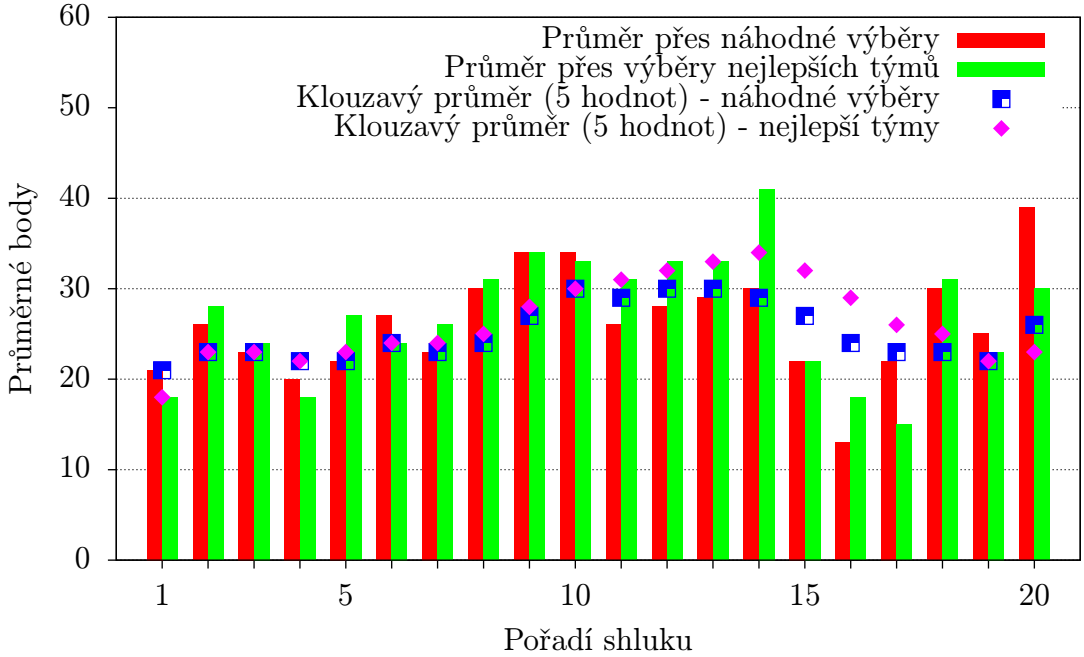
5.4 Druhá evoluce

Druhá evoluce vychází ze stejné úvodní generace i nastavení jako předešlá evoluce, pouze tato má desetkrát větší pravděpodobnost mutace vlastnosti a dva a půl krát větší pravděpodobnost mutace pozice.

Byla analyzována stejným postupem jako předešlá. Pro úsporu místa je uveden pouze graf srovnání obou výběrů s klouzavým průměrem z posledních pěti



Obrázek 5.3: Srovnání výsledků první evoluce - porovnání obou výběrů.



Obrázek 5.4: Srovnání výsledků druhé evoluce - porovnání obou výběrů.

hodnot, který je na obrázku 5.4. Protože tato evoluce je kratší a má pouze tisíc generací, je i počet shluků (jejichž velikost jsem ponechal), jen 20.

5.5 Hra týmů

Při sledování hry lepších týmů oproti horším se mi nepodařilo určit nějaký rys, který bych mohl označit jako ten, díky kterému hraje určitý tým lépe než jiný. Přičítám to hlavně jednoduchosti simulace a možná příliš krátké evoluci.

Závěr

Musím říct, že práce mi přišla vcelku obtížná, protože celý proces návrhu a vývoje byl pro mě nový. Bylo třeba vymyslet a navrhnout simulaci, nastudovat princip evolučního algoritmu, rozmyslet formáty dat, rozvrhnout program a vše realizovat. Jenom samotné programování pro mě bylo velkou výzvou, neboť takto rozsáhlý program, který zahrnuje práci se sítí, vlákna, manipulaci se soubory i tvorbu grafického prostředí s pomocí Windows Presentation Foundation, jsem nikdy předtím nedělal.

Díky této práci jsem se tak mnohému naučil. Během vývoje programu se proměnil můj programátorský styl (je patrný i v různých částech kódu programu) a získal jsem zkušenosti i v oblastech, které se týkají programování nepřímo, jako třeba způsob organizace projektu či práce se systémem správy verzí. V neposlední řadě jsem se utvrdil v tom, že nejsem vůbec schopen odhadovat, kolik času mi jednotlivé úkoly zaberou.

Nutno podotknout, že samotnou práci za nějak moc užitečnou nepovažuji, ale díky důvodům zmíněným v předchozím odstavci jsem rád, že jsem se do ní pustil.

Seznam použité literatury

[1] Mitchell, Melanie. *An Introduction to Genetic Algorithms*. 5. vydání. First MIT Press paperback edition, 1998. ISBN 0-262-63185-7.

[2] *Bipartite Matching & the Hungarian Method* [online]. 2006 [cit. 2011-05-21]. Dostupné z WWW: <http://www.cse.ust.hk/~golin/COMP572/Notes/Matching.pdf>.

[3] *NUnit - Home* [online]. [cit. 2011-05-21]. Dostupné z WWW: <http://www.nunit.org/>.

[4] Regueiro, Alex. *Noldorin's Blog* [online]. 2009 [cit. 2011-05-21]. Hungarian Algorithm in C#. Dostupné z WWW: <http://noldorin.com/blog/2009/09/hungarian-algorithm-in-csharp/>.

[5] *Java^{TM} 2 Platform, Standard Edition, v 1.4.2 API Specification* [online]. [cit. 2011-05-21]. Random (Java 2 Platform SE v1.4.2). Dostupné z WWW: <http://download.oracle.com/javase/1.4.2/docs/api/java/util/Random.html#nextGaussian%28%29>.

Přílohy

Přílohy se nachází na přiloženém DVD v adresáři „Prilohy“.

(1) Příklady vstupních souborů
V adresáři „1_Priklady_vstupnich_souboru“ se nachází příklady tří souborů, které jsou nutné ke spuštění evoluce. V souborech jsou v komentářích uvedeny přípustné hodnoty.

(2) Zdrojové kódy knihovny JiM.MyLib
V adresáři „2_Knihovna_-_zdrojove_kody“ jsou zdrojové kódy knihovny organizované v projektu vývojového prostředí Microsoft Visual Studio 2010 Express. Ke zkompilování knihovny je potřeba knihovna NUnit verze 2.5.8, jejíž instalátor se v adresáři rovněž nachází.

(3) Dokumentace knihovny JiM.MyLib
Adresář „3_Knihovna_-_dokumentace“ obsahuje dokumentaci ke knihovně JiM.MyLib ve formátu HTML (úvodní stránka je „index.html“).

(4) Zdrojové kódy hlavní části
V adresáři „4_Hlavni_cast_-_zdrojove_kody“ jsou zdrojové kódy hlavní části programu organizované v projektech vývojového prostředí Microsoft Visual Studio 2010 Express. Ke zkompilování projektů je potřeba nastavit správnou cestu ke knihovně JiM.MyLib.

(5) Dokumentace hlavní části
V adresáři „5_Hlavni_cast_-_dokumentace“ se nalézá dokumentace k hlavní části programu ve formátu HTML (úvodní stránka je „index.html“).

(6) Simulované evoluce
Simulované evoluce jsou v adresáři „6_Simulovane_evoluce“. Bližší popis je v souboru „README.txt“, který se v adresáři také nachází.

(7) Programy
Adresář „7_Programy“ obsahuje spustitelné programy. Bližší informace jsou v souboru „README.txt“, který se v adresáři nachází.