

Vysoké učení technické v Brně  
Fakulta informačních technologií

Diplomová práce  
Strategická počítačová hra

## **Prohlášení**

Autor prohlašuje, že tento diplomový projekt vypracoval samostatně pod vedením vedoucího projektu a že uvedl všechny literární prameny a publikace, ze kterých čerpal.

Autor tímto dává Fakultě informačních technologií Vysokého učení technického v Brně svolení k použití a úpravám textu i jeho částí pro vědecké a výukové účely při zachování autorských práv. Současně souhlasí s tím, že programový kód, který byl vytvořen jako součást této práce, je možné používat v souladu s General Public License.

## **Poděkování**

Autor tímto děkuje panu Ing. Martinu Hrubému, Ph.D. za poskytnutí studijních materiálů, za jeho odbornou pomoc při tvorbě diplomového projektu a užitečné rady.

V Brně dne 23. května 2006.....

Jakub Bednář

## **Abstrakt**

Předmětem zájmu této práce je aplikace matematické teorie her jako základního principu pro budování systémů s umělou inteligencí. Demonstrace způsobů aplikace tohoto přístupu je provedena na modelu smyšlené společnosti s mnoha různými projevy činnosti a existence inteligentních entit. Podobné modely jsou v informatice nazývány strategické počítačové hry. V našem případě se jedná o virtuální království s tržní ekonomikou, různorodou průmyslovou výrobou a kooperací. V práci jsou rozebrány a analyzovány mnohé teoretické přístupy teorie her. Vybrané části z nich jsou implementovány do funkčního simulačního celku.

## **Klíčová slova**

Teorie her, Umělá inteligence, Diskrétní simulace

## **Abstract**

This project concerns on the application of game theory as a basic concept for building systems with artificial intelligence. Demonstration of this approach is made on a model of fictive society with many different exposures of intelligent entities activity and existence. This kind of models are called strategic computer games. In this project is used a model of virtual kingdom with free market economy, industrial production and cooperation. Many theoretical approaches of game theory are analyzed in this paper. Some of them are implemented in the functional simulation system.

## **Keywords**

Game theory, Artificial intelligence, Discrete simulation

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Umělá inteligence a teorie her</b>	<b>5</b>
2.1	Historie teorie her . . . . .	6
2.2	Parametrické a neparametrické chování . . . . .	6
2.3	Základní pojmy a předpoklady teorie her . . . . .	7
2.4	Reprezentace her - modely situací . . . . .	8
2.5	Vězňovo dilema . . . . .	9
2.6	Nashovo equilibrium . . . . .	11
2.6.1	Antagonistické hry . . . . .	11
2.6.2	Neantagonistické hry . . . . .	12
2.7	Směšené strategie . . . . .	12
2.7.1	Kardinální funkce užitku . . . . .	14
2.7.2	Závěr problému s přechodem řeky . . . . .	15
2.7.3	Postup a složitost řešení hry v normálním tvaru . . . . .	16
<b>3</b>	<b>Modelování oponentů v systémech s neúplnou informací</b>	<b>19</b>
3.1	Case-based reasoning . . . . .	20
3.1.1	Typy znalostí . . . . .	20
3.2	Ukazatel podobnosti . . . . .	22
3.3	Porovnávání situací v multiagentních systémech . . . . .	22
<b>4</b>	<b>Počítačové hry</b>	<b>23</b>
4.1	Typy počítačových her . . . . .	23
4.1.1	Akční hry . . . . .	23
4.1.2	Adventury . . . . .	24
4.1.3	RPG (role playing game) . . . . .	24
4.1.4	Strategické hry . . . . .	25
4.2	Inspirace vlastního herního systému . . . . .	27
4.2.1	Settlers 2 . . . . .	27
4.2.2	Majesty . . . . .	28
<b>5</b>	<b>Návrh systému</b>	<b>30</b>
5.1	Základní rysy systému . . . . .	31
5.2	Statická část . . . . .	31
5.2.1	Krajina . . . . .	31

5.2.2	Nerostné bohatství . . . . .	32
5.2.3	Budovy . . . . .	32
5.2.4	Zboží . . . . .	32
5.3	Dynamická část . . . . .	33
5.3.1	Zvěř . . . . .	33
5.3.2	Řemeslníci . . . . .	34
5.3.3	Hrdinové . . . . .	37
5.3.4	Nepřátelské herní entity . . . . .	38
<b>6</b>	<b>Popis řešení vybraných částí navrženého systému</b>	<b>39</b>
6.1	Procesně řízená simulace . . . . .	39
6.2	Modelování znalostí . . . . .	40
6.2.1	Modelování báze znalostí . . . . .	40
6.2.2	Časová a prostorová distribuce znalostí mezi herními entitami . . . . .	41
6.3	Model trhu . . . . .	43
6.3.1	Příklad . . . . .	44
6.3.2	Volba sortimentu . . . . .	46
6.4	Řemeslníci . . . . .	47
6.5	Modelování oponentů . . . . .	47
<b>7</b>	<b>Experimenty a výsledky</b>	<b>51</b>
7.1	Uživatelské rozhraní . . . . .	51
7.1.1	Vstupní formát . . . . .	51
7.1.2	Výstupní formát . . . . .	52
7.2	Projekt TRH . . . . .	53
7.2.1	Gametracer . . . . .	55
7.2.2	Gambit . . . . .	55
<b>8</b>	<b>Závěr</b>	<b>58</b>

# Kapitola 1

## Úvod

V posledních letech se herní průmysl stává jedním z hlavních tahounů vývoje informačních technologií a to jak v oblasti hardwaru, tak i softwaru. Hráči jsou rozvíjejícím se herním průmyslem nuceni vybavovat své stroje stále většími pevnými disky, zrychlovat své procesory a kupovat stále dokonalejší grafické karty pro dokonalý požitek ze hry. Co nabízejí hráčům na oplátku herní společnosti, aby si zajistili jejich stálou přízeň?

Nabízena je celá škála typů her, od tahových a real-time strategií přes RPG a adventury až po 3D akční hry (uvedené termíny patří do základní terminologie uživatelů - hráčů). Neustále se zdokonalují technologie pro počítačovou grafiku a programovací techniky pro důvěryhodnější modelování reálného prostředí. Na konci tohoto vývoje můžeme očekávat vznik virtuální reality tak, jak ji nyní známe pouze ze sci-fi filmů.

Fenomémem posledních let, jehož vznik byl umožněn rychlým rozvojem Internetu, jeho zvyšující se rychlostí a snadnější dostupností, se staly síťové hry více hráčů. Jednou z hlavních příčin vzniku tohoto fenoménu je, kromě touhy změřit své síly s přáteli, stále ještě nedostatečná inteligence modelů herních postav, typicky umělých protihráčů. I přes dynamický vývoj umělé inteligence se ještě stále můžeme setkat s pro nás absurdními situacemi jako je například počítačový válečník, který se už po patnácté marně snaží vyskočit na krabici, aby nás mohl ostřelovat, nebo německý voják na obchůzce, který klidně překročí tělo mrtvého kolegy a ani ho nezaskočí, že tam leží, když ještě před chvílí stál a hlídal vchod do pevnosti.

U počítačových her rozlišujeme kvalitu (realističnost) umělé inteligence a tzv. hratelnost. Při programování umělé inteligence pro účely pobavení lidského hráče není možné kombinační logiku stroje přehnat. Příliš dokonalý protihráč by lidského hráče jen odradil. Každý člověk dělá chyby, a aby mohl vyhrát, potřebuje soupeře, který rovněž dělá chyby. Pokud by chybující člověk měl měřit síly s vysoce inteligentním strojem, který chyby nedělá, pak by taková hra u hráčů velký ohlas nesklidila.

Tento projekt je zaměřen právě na umělou inteligenci v oblasti počítačových her. Her je však příliš mnoho a nelze jejich specifika postihnout současně, proto byly jako inspirace vybrány dvě hry postavené na počítačové simulaci. Cílem projektu je rozebrat některé metody a postupy, které se používají pro modelování inteligentního chování, a pokusit se o jejich praktickou implementaci na vlastním herním systému. Hlavním zaměřením této práce je velmi specifická matematická teorie, která se nazývá teorie her. V dalším textu budou rozebrány základní pojmy, koncepty a aplikace matematické teorie her. Bude prezentován alternativní přístup k modelování inteligence v systémech s nedokonalou a neúplnou informací. Podíváme se na návrh celého systému a popis

řešení spolu s výsledky experimentů klíčových částí tohoto systému.

Tato diplomová práce vznikla v návaznosti na Semestrální projekt. V rámci Semestrálního projektu byly rozebrány postupy matematické teorie her a byl navržen implementovaný herní systém (viz kapitoly 2 a 5). Všechny návrhy konkrétního řešení byly, spolu s úpravami v návrhu systému, provedeny v rámci této práce.



## Kapitola 2

# Umělá inteligence a teorie her

*V této kapitole se seznámíme s důvody vzniku a s historií matematické teorie her. Budou rovněž představeny základní pojmy, předpoklady a principy této teorie a bude demonstrována aplikace této teorie na konkrétních příkladech.*

Inteligence je vlastností některých živých organismů, která jim dává v přírodě mimořádné postavení. S rozvojem techniky se mnozí vědci a filosofové začali zabývat otázkou, zda lze u vytvořených systémů dosáhnout reakcí, které bychom u živých tvorů považovali za inteligentní. Vzniklo mnoho sci-fi filmů o inteligentních strojích s katastrofickými scénáři, ale dlouho se nikdo vědecky nezabýval analýzou možností strojové inteligence. Ve skutečnosti se výzkumem umělé inteligence jako první zabýval britský matematik Alan Turing již v první polovině minulého století, ale teprve s prudkým rozmachem výpočetní techniky byly postupně navrhovány a experimentálně ověřovány metody, postupy a algoritmy umožňující napodobovat inteligenci v některých oblastech. Inspirací pro vědce v oblasti umělé inteligence jsou detailní analýzy činnosti živých organismů a matematické abstrakce procesů probíhajících v lidském mozku.

Všechny postupy a algoritmy, které napodobují inteligentní chování člověka, jsou předmětem zkoumání vědní disciplíny Umělá inteligence.

Jedním z měřítek pro určení míry inteligence systému je Turingův test, který je založen na následující myšlence: "Bude-li stroj reagovat na podněty lidského partnera takovým způsobem, že člověk nebude schopen rozeznat, zda jedná se strojem či s jiným člověkem prostřednictvím terminálu, lze považovat tento stroj za inteligentní."

Důležitým pojmem matematické teorie her je pojem racionálního chování nebo racionálního uvažování. Za racionální je považována jakákoliv činnost, která je rozumem odůvodněná, účelná nebo vycházející z úvahy [4]. Racionální chování tak nemusí být legální ani legitimní, pokud tyto pojmy nejsou zahrnuty do procesu rozhodování.

Metod a postupů pro popis inteligentního a racionálního chování člověka existuje více (například modelování agentů, neuronové sítě, atd.), tato práce je zaměřena především na využití matematické teorie her.

Teorie her je jedním z odvětví aplikované matematiky, která se zabývá způsoby, jimiž strategická interakce mezi racionálními hráči produkuje výstupy s ohledem na preference těchto hráčů, přičemž tyto výstupy nemohou být přímo určeny žádným z hráčů. První práce zmiňující se o teorii her se zabývaly touto teorií jako nástrojem pro analýzu a pochopení ekonomických procesů. V současnosti je však efektivně využívána i v celé řadě jiných odvětví, od biologie až po filosofii.

## 2.1 Historie teorie her

Základy matematické teorie her byly položeny Johnem von Neumannem a Oskarem Morgenster- nem na konci první poloviny minulého století. První filozofické problémy spadající do oblasti teorie her však můžeme nalézt již v dílech antických filosofů.

Platón se ve svém díle *Republica* zamýšlí nad situací vojáka stojícího na bitevním poli a očekávajícího útok nepřítele. Jak by se měl *racionální* (účelně jednající, z úvahy vycházející) voják chovat? Pokud je bránící armáda silnější než útočící, pak se jistě obejde bez jednotlivce a je tedy nesmyslné riskovat ztrátu svého života. Racionální voják proto z boje uteče, neboť jeho armáda zvítězí i bez něj. Pokud je naopak armáda nepřítele silnější, je pravděpodobnost zabití v boji velmi vysoká a je proto zcela racionální utéci z boje. Podle těchto úvah uteče racionální voják z boje v každém případě. Za předpokladu, že všichni vojáci v armádě uvažují racionálně, se armáda rozprchne bez jediného výstřelu.

Řešením tohoto problému z pohledu velitele armády je buď útěk fyzicky znemožnit (například Cortéz při dobývání Ameriky spálil všechny lodě, na nichž připlul se svojí armádou), nebo vyhlásit trest za dezerci tak vysoký, aby bylo pro vojáka racionálně výhodnější bojovat a doufat, že bitvu přežije.

Hobbes se ve svém díle *Leviathan* zabývá obdobným problémem. Podle něj je nejlepším uspořádáním pro lidstvo, když každý může dělat to, co si přeje. Lidé v takové společnosti budou chtít vzájemně kooperovat s cílem dokázat věci, které jednotlivec nedokáže. Vyskytne-li se však v této společnosti amorální jedinec, brzy zjistí, že je pro něj výhodnější využívat spolupráce pro uspokojení svých potřeb, ale nepřispívat k uspokojení potřeb jiného. Modelovým příkladem může být dohoda dvou sousedů ( $A$  a  $B$ ), že si navzájem pomohou postavit své domy. Soused  $B$  pomáhá  $A$  se stavbou domu za příslib, že potom  $A$  pomůže se stavbou domu  $B$ . Když je dům souseda  $A$  postaven,  $A$  zruší slib a odmítne pomoc se stavbou domu sousedovi  $B$ . Záhy se však  $A$  začne souseda  $B$  obávat, neboť ten, bez střechy nad hlavou, má značnou motivaci pokusit se  $A$  dům násilím sebrat. Před  $A$  nyní stojí dvě možnosti. Buď svůj dům nákladně a zdlouhavě bránit a zabezpečovat den po dni, nebo souseda  $B$  zavraždit. Racionálnější pro amorálního souseda  $A$  je jednoznačně vražda. Soused  $B$  je však rovněž racionální a může celý vývoj situace předvídat již před uzavřením dohody. Proto racionálně nemůže žádná dohoda o stavbě domu mezi sousedy obávajícími se podvodu vzniknout. Lidé v tomto světě by tak žili odděleně a nikdy by ničeho významného nedosáhli. Řešením podle Hobbese je volba tyрана nebo vlády, která bude krutě trestat jakékoliv porušení slibu tak tvrdě, aby se každému vyplatilo slovo dodržet.

Teorie her staví možná racionalitu at absurdum (jak je vidět z uvedených příkladů), ale je zřejmě pravdivá při zjednodušení, které u her předpokládáme. V reálném světě asi nebudeme očekávat takovou míru cynismu. Uvidíme, že tento jev souvisí jak s uvedenou definicí racionality, tak i s výkladem dalších základních pojmů matematické teorie her, jako je zisk nebo strategie.

## 2.2 Parametrické a neparametrické chování

Rozhodování racionálního jedinice je ovlivňováno mnoha faktory. Některé faktory jsou stálé (například pravděpodobnost, že při hození kostkou padne číslo šest), jiné faktory jsou závislé na rozhodování dalších racionálních hráčů. V tomto odstavci se podíváme na rozdíly mezi parametrickým chováním v pasivním prostředí a neparametrickým chováním v aktivním prostředí, tedy v prostředí, které se nám snaží klást překážky. Druhé z uvedených chování je mnohem

složitější, jak napoví následující příklad.

Představme si tři mosty vedoucí přes řeku. Jeden most je bez překážek, druhý vede pod skálou a hrozí, že na nás spadne kámen a třetí je plný jedovatých hadů. Naším problémem je vybrat si most, po kterém přejdeme na druhou stranu. Řekněme, že při rozhodování mezi mostem pod skálou a mostem s hady jsme vyzozorovali, že zranění kamenem má pravděpodobnost 10% a uštknutí hadem má pravděpodobnost 20%. Zde se jednoznačně jedná o parametrický problém v pasivním prostředí, neboť nelze předpokládat, že by kameny nebo hadi měli vůli skrýt své běžné chování, aby nás zmátli. Při rozhodování si tedy racionálně vybereme ten most, na kterém nám hrozí nejmenší nebezpečí.

Ztížíme-li situaci tím, že bezpečný most je den náročné cesty od nás, zatímco most s kameny je přímo před námi, podstata problému se v zásadě nezmění. Musíme se pouze rozhodnout, zda je pro nás den náročné cesty výhodnější než desetiprocentní šance, že budeme zasaženi kamenem.

Nyní situaci ještě více ztížíme a ukážeme problematiku neparametrického chování. Představme si, že na druhé straně řeky stojí zabiják, který nás chce vidět mrtvé. Může nás však zabít, pouze pokud bude čekat na konci stejného mostu, který si vybereme pro přechod řeky. Pokud se nyní racionálně rozhodneme přejít po bezpečném mostě, záhy si uvědomíme, že racionální zabiják, který má stejné informace jako my, dojde racionální úvahou k tomu samému závěru a přechod přes tento most se tak rovná jisté smrti. Rozhodneme se tedy racionálně pro most pod skálou. Ale pozor, pokud jsme racionálně došli k závěru, že nejméně pravděpodobné je, že zabiják bude čekat na tomto mostě, pak stejným způsobem uvažoval určitě i zabiják, a je tak náhle nejpravděpodobnější, že bude čekat právě na mostě pod skálou. Podobně je to i s mostem s hady. Dostáváme se tak do začarovaného kruhu, kdy volba, která se zdá být nejméně pravděpodobná, že ji nepřítel zvolí, se v okamžiku kdy ji zvolíme my, stává nejpravděpodobnější.

Jak víme, v reálné situaci nezůstane nikdo nerozhodný, ale zvolí si cestu. I v neparametrickém problému existuje racionální řešení nejlepší pro oba hráče. Avšak před vznikem matematické teorie her nikdo nevěděl, jak toto řešení nalézt. Většina postupů před Johnem von Neumanem se opírala o parametrické chování a neparametrické chování bylo modelováno pouze jako omezující podmínky chování parametrického. V následujících kapitolách přiblížíme metody užívané pro řešení těchto problémů v matematické teorii her.

## 2.3 Základní pojmy a předpoklady teorie her

V dalším textu se budeme opírat o definice základních pojmů matematické teorie her. Proto si je nyní uveďme.

- *Hra* je nějaká činnost s definovanými pravidly a s určitým cílem. Hry se účastní hráči.
- *Hráč (agent)* je entita hry, která interaguje s dalšími hráči za účelem vyhrát. Hráč je definován svým inteligentním chováním. Jelikož nejsme schopni modelovat inteligentní myšlení reálného člověka, omezujeme se na nejmarkantnější aspekt jeho chování - racionalitu. Hráč se tak soustředí pouze na maximalizaci svého užitku (zisku), který plyne z jeho chování.
- *Strategie* je z terminologického hlediska problematický prvek specifikace. V literatuře se často setkáme s pojmem akce (možného postupu) hráče ve hře. V tržních hrách to bývá nastavení ceny a množství zboží, u šachů je to následující tah figurou atd. Každý hráč

		Killer		
		Safe bridge	Rocky bridge	Snake bridge
Player	Safe bridge	0, 1	1, 0	1, 0
	Rocky bridge	?	0, 1	?
	Snake bridge	?	?	0, 1

Obrázek 2.1: Maticová reprezentace hry

hledá strategii, která maximalizuje jeho užitek. Hráči však vědí, že jejich užitek je ovlivněn i volbou strategie dalších hráčů.

- *Užitek (zisk)* je předmět nebo událost, která uspokojuje osobní preference hráče. V tržních hrách to bývá finanční zisk, u šachů je to míra převahy nad soupeřem atd.
- *Informace* ovlivňuje volbu strategie hráče. Nejjednodušší jsou hry, ve kterých mají hráči dokonalé informace o celém jejich průběhu. Klasickým příkladem jsou deskové hry (například šachy), kdy oba hráči sledují celý vývoj hry. Naopak problém s mostem uvedený výše je příkladem hry s nedokonalými informacemi, neboť volíme svou strategii při neznalosti rozhodnutí zabijáka, na kterém mostě bude stát.

## 2.4 Reprezentace her - modely situací

Rozdíl mezi hrami s úplnou a neúplnou informací je přímo spojen s metodami modelování her s ohledem na pořadí, ve kterém hráči volí své strategie.

- *Sekvenční hra* je hra, kde se hráči střídají ve volbě svých strategií a každý z hráčů zná všechny předchozí volby strategií jak své, tak soupeřů.
- *Simultání hra* je hra, kde hráči volí své strategie současně. Tím není myšleno ve stejný časový okamžik, ale bez znalosti volby strategie soupeřem.

Sekvenční hry jsou nejjednodušším typem her, neboť se jedná o hry s konečným počtem tahů, a je tedy možné analyticky vypočítat výstupy pro všechny možné kombinace volby strategií obou hráčů. Hráči pak postupují volbou strategie, která jim přináší nejvyšší užitek. Výpočetní proces používaný v tomto typu her se nazývá zpětná indukce. Nástrojem používaným pro zobrazení sekvenčních her jsou  $n$ -ární stromy.

Mnohdy se však místo stromové reprezentace her používá reprezentace maticová, která ukazuje pouze konečné užítky hráčů pro možné kombinace strategií.

Pro náš problém s řekou a mosty vypadá matice tak, jak ukazuje obrázek 2.1. V každé buňce matice jsou dvě čísla. První udává hodnotu užítku hráče v řádku, druhé udává hodnotu užítku hráče ve sloupci. Z matice je zřejmé, že pokud si hráč zvolí přechod přes bezpečný most a zabiják se rozhodne čekat na tomto mostě, bude hodnota užítku pro hráče rovna 0 a hodnota užítku pro

		Player 2	
		Confess	Refuse
Player 1	Confess	-5, -5	0, -10
	Refuse	-10, 0	-2, -2

Obrázek 2.2: Matice hry Vězňovo dilema

zabijáka rovna 1. Místa vyznačená znakem "?" nejsou vyplněna, neboť zde kromě zabijáka hraje roli i nebezpečí číhající na mostě. Tyto hodnoty doplníme později.

Hry reprezentované formou matice nazýváme *hrami v normálním tvaru*, zatímco hry reprezentované stromem nazýváme *hrami v explicitním tvaru*. Tyto dva typy her nejsou ekvivalentní. Lze dokázat, že každá hra v explicitním tvaru je reprezentovatelná množinou her v normálním tvaru, kdy pro každý uzel stromu získáme jednu matici strategií. K objasnění rozdílu mezi hrami v explicitním a normálním tvaru použijeme jeden z nejznámějších a často uváděných problémů teorie her, tzv. Vězňovo dilema.

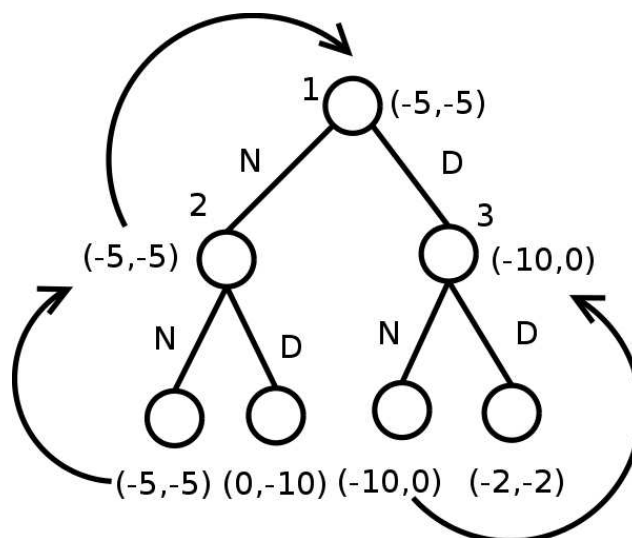
## 2.5 Vězňovo dilema

Problém vězňova dilematu je zadán takto. Policisté zajali dva lupiče, kteří jsou obviněni z vyloupení banky. Policie ví, že tento čin spáchali, ale nemá dostatek důkazů. Má však dostatek důkazů, že oba muži ukradli automobil, za což by každý dostal dva roky vězení. Oba vězňové jsou okamžitě separováni, takže nemají možnost dohodnout se na spolupráci, a každému jsou nabídnuty tyto možnosti:

1. Pokud budeš vypovídat ohledně vyloupení banky proti tvému partnerovi a on nebude vypovídat, dostane on deset let vězení a tebe propustíme.
2. Pokud budeš vypovídat ohledně vyloupení banky proti tvému partnerovi a on bude vypovídat proti tobě, dostanete každý po pěti letech vězení.
3. Pokud nebudeš vypovídat ohledně vyloupení banky a tvůj partner také ne, dostanete každý po dvou letech vězení za krádež auta.

Matice strategií pro takto zadanou hru je na obrázku 2.2. Ve smyslu funkce užítku jsou jednotlivé tresty převedeny do záporných hodnot. Tím je zajištěno, že nejvýhodnější pro hráče je volba s nejvyšším užítkem. Funkce užítku je však uvažována pouze jako ordinální. Kardinální funkcí užítku se budeme zabývat později.

Buďme na chvíli v roli Hráče 1. Budeme hledat naši nejlepší odpověď (best response) na všechny možné strategie Hráče 2. Pokud se Hráč 2 rozhodne vypovídat (strategie confess), bude náš užitek buď -5 (pět let ve vězení), když budeme vypovídat, nebo -10 (deset let vězení), když vypovídat odmítneme (strategie refuse). Pokud se Hráč 2 rozhodne nevypovídat, bude náš užitek buď 0 (propuštění na svobodu), nebo -2 (dva roky vězení). S úvah je zřejmé, že bez ohledu na



Obrázek 2.3: Strom Vězňova dilematu v explicitní formě

volbu strategie Hráče 2, je pro nás výhodnější vypovídat, neboť v každém sloupci matice je náš užitek větší na prvním řádku než na druhém. K naprosto stejnému závěru dojde i Hráč 2. Oba hráči tedy budou vypovídat a dostanou po pěti letech vězení.

O strategii Vypovídá řekneme, že je strategií dominující nad strategií Nevypovídá, neboť ve všech sloupcích matice je náš užitek větší při volbě strategie Vypovídá, než v tom samém sloupci při volbě strategie Nevypovídá. O strategii Nevypovídá řekneme, že je dominována strategií Vypovídá. Strategii Nevypovídá nikdy ne zvolíme, neboť náš užitek je vždy menší než při volbě strategie Vypovídá a to bez ohledu na volbu strategie Hráče 2. Strategii Nevypovídá tedy můžeme z matice strategií eliminovat a zůstane nám řešení, kdy každý z hráčů dostane pět let vězení. Použitý analytický postup se nazývá eliminace dominovaných strategií. Strategie Vypovídá je dominantní strategií uvedené hry.

Podívejme se nyní znovu na matici strategií. Vidíme, že nejvýhodnější pro oba hráče je volba strategie Nevypovídá, kdy dostanou pouze po dvou letech vězení. Změní se výsledek, pokud budou mít oba vězni možnost se před výsledkem domluvit? Odpověď zní ne, neboť každý z hráčů ze strachu, že ho druhý hráč podvede, bude vypovídat.

Uvažujme nyní situaci, kdy vězňové nejsou separováni a jeden vězeň tak může volit svoji strategii až podle volby strategie druhého vězně. Tím se dostáváme ke hře v explicitním tvaru, neboť oba hráči mají dokonalé informace o celém průběhu hry. Předpokládejme rovněž, že oba hráči uzavřeli dohodu, že ani jeden nebude vypovídat. Označme dodržení dohody písmenem "D" a nedodržení písmenem "N". Strom pro takto definovanou hru je na obrázku 2.3.

Analyzujme nyní tento strom použitím Zermelova algoritmu [2]. Zaměříme se nejprve na sub-hru z uzlu 3. Zde se Hráč 2 rozhoduje mezi užitekem -2 a 0. Racionálně si vybere hodnotu užitku 0, tedy nedodrží dohodu. Můžeme tak hodnoty (-10, 0) přiřadit přímo uzlu 3. Obdobně postupujeme pro sub-hru z uzlu 2 a přiřadíme mu hodnoty užitků (-5, -5). Nyní se zaměříme na sub-hru z uzlu 1, která zároveň reprezentuje celou hru. Hráč 1 se rozhoduje mezi hodnotami užitku -10 a -5. Racionálně si zvolí hodnotu -5 a poruší tedy dohodu hned v úvodu celé hry.

Vidíme, že u Vězňova dilematu vedou obě formy hry ke stejnému výsledku. V řadě případů tomu tak ovšem není. Zároveň jsme demonstrovali použití Zermelova algoritmu, který je aplikovatelný na všechny konečné hry v explicitním tvaru s dokonalou informací.

		Player 2			Min H1
		t1	t2	t3	
Player 1	s1	5	3	2	2
	s2	4	-2	7	-2
	s3	8	-1	4	-1
Min H2		8	3	2	

Obrázek 2.4: Matice hry s nulovým součtem

## 2.6 Nashovo equilibrium

V předchozí sekci jsme uvedli, že řešením Věžňova dilematu je volba strategií Vypovídat. Místo pojmu řešení hry se obvykle používá pojem equilibrium (rovnováha). Nashovo equilibrium (NE) je taková kombinace strategií, kdy žádný z hráčů nemůže zvýšit svůj užitek změnou své strategie za předpokladu, že strategie ostatních hráčů zůstanou beze změny. Z toho vyplývá, že žádná dominovaná strategie nemůže náležet do NE. Znamená to rovněž, že pokud jsme eliminací dominovaných strategií dospěli k jediné strategii, pak jsme našli NE. NE zahrnuje množiny strategií, vždy pro každého hráče jednu strategii.

Koncept Nashova equilibria ve skutečnosti nepochází od Johna Nashe, ale od A. A. Cournota. John Nash však jako první ve své disertační práci [5] dokázal, že Nashovo equilibrium musí existovat pro všechny konečné hry s libovolným počtem hráčů.

### 2.6.1 Antagonistické hry

Antagonistický konflikt je rozhodovací situace, kdy se hráči po volbě svých rozhodnutí rozdělí o pevnou odměnu, jejíž výše nezávisí na tom, jaké rozhodnutí zvolili. Matematickým modelem antagonistického konfliktu je hra v normálním tvaru s konstantním součtem. Dá se ukázat, že každou hru v normálním tvaru s konstantním součtem lze převést na hru v normálním tvaru s nulovým součtem, která je s původní hrou strategicky ekvivalentní.

Jedna z možných matic pro normální hru s nulovým součtem je na obrázku 2.4. V každé buňce matice je uvedena pouze hodnota užitku Hráče 1. Hodnota užitku Hráče 2 je hodnotou opačnou. Pro řešení tohoto typu her je nalezení NE nutnou a zároveň postačující podmínkou.

Postup pro nalezení NE je následující. Hráč 1 prochází všechny své strategie a hledá svůj nejmenší zaručený užitek pro každou strategii. Z těchto nejmenších zaručených užiteků pak vybírá své maximum. V matici na obrázku 2.4 jsou výsledky pro Hráče 1 znázorněny v posledním sloupci. Obdobně postupuje Hráč 2. Pro všechny své strategie hledá svůj minimální zaručený užitek (tedy maximální možný užitek Hráče 1) a z těchto hodnot pak vybírá své maximum (minimum pro Hráče 1). Výsledky pro Hráče 2 jsou znázorněny v posledním řádku matice. NE pro takto zadanou matici je tedy dvojice strategií s1-t3.

		Player 2	
		t1	t2
Player 1	s1	10, 10	0, 0
	s2	0, 0	2, 2

Obrázek 2.5: Matice hry se dvěma NE

		Player 2	
		t1	t2
Player 1	s1	10, 0	5, 2
	s2	10, 1	2, 0

Obrázek 2.6: Matice hry se slabě dominovanými strategiemi - příklad 1

### 2.6.2 Neantagonistické hry

Bohužel ne všechny hry jsou antagonistické, tedy s nulovým součtem. Základním problémem neantagonistických her je možná existence více než jednoho NE. Navíc ne všechny NE musí být stejně výhodné, jak ukazuje matice na obrázku 2.5.

Tato hra má dvě NE: s1-t1 a s2-t2. V této matici rovněž neexistuje vztah dominance jedné strategie nad druhou. Pokud by jedinou metodou pro řešení neantagonistických her bylo hledání NE, pak bychom řekli, že řešením hry je dvojice NE: s1-t1 a s2-t2. Racionální hráči se však budou snažit konvergovat k NE s1-t1. To ukazuje, že nalezení NE je nepostačující podmínkou pro řešení neantagonistických her.

Uvažujme nyní matici hry na obrázku 2.6. Zde strategie s1 slabě dominuje nad strategií s2, neboť při jakékoli volbě strategie Hráčem 2 je užitek Hráče 1 stejný, v jednom případě dokonce vyšší při volbě s1. Je tedy v pořádku eliminovat slabě dominovanou strategii s2? Pokud tak učiníme, bude v dalším kroku strategie t1 dominována strategií t2, a hra dospěje do NE s1-t2.

Uvažujme nyní podobnou matici s upravenými hodnotami užitku pro jednotlivé hráče (viz obrázek 2.7). Strategie s2 je stále slabě dominována strategií s1. Ovšem NE s2-t1 je nyní pro hráče výhodnější než NE s1-t2. Eliminací slabě dominované strategie s2 bychom eliminovali nejlepší možné řešení hry. Jediným důvodem pro eliminaci strategie s2 může být obava Hráče 1 z neracionality Hráče 2. Postupům, při nichž odstraňujeme některá NE, se říká zjemňování NE a existuje pro ně celá řada principů a metod, které jsou nad rámec této práce.

## 2.7 Smíšené strategie

Až doposud jsme se zabývali řešením her v tzv. ryzích strategiích, kdy každý z hráčů hledal jednoznačnou strategii, kterou má zvolit. Často se však Nashovo equilibrium nenachází v ryzích, ale v tzv. smíšených strategiích. Smíšenou strategií hráče je funkce rozložení pravděpodobnosti



		Player 2	
		t1	t2
Player 1	s1	10, 0	5, 2
	s2	10, 11	2, 0

Obrázek 2.7: Matice hry se slabě dominovanými strategiemi - příklad 2

nad množinou ryzích strategií hráče. Při volbě strategie si hráč hodí mincí a vybere jednu z možných strategií podle výsledku hodu. Použití smíšených strategií je nezbytné ve všech případech, kdy neexistuje ryzí strategie maximalizující užitek hráče proti všem možným strategiím soupeře. Potřebu smíšených strategií lze objasnit na našem problému s přechodem přes řeku.

Uvedli jsme, že v okamžiku kdy se hráč racionálně rozhodne pro jeden most, dostane se stejným racionálním postupem ke stejnému závěru také zabiják. To znamená, že hráč může zabijákovi uniknout pouze v případě, že zabiják není schopen racionálně odvodit, který most si hráč vybere. Z toho vyplývá, že hráč dokáže překvapit zabijáka, pouze dokáže-li překvapit sám sebe.

Předpokládejme nyní, že všechny tři mosty jsou stejně bezpečné a že hráč nemá žádné zvláštní informace o zabijákovi, které by měnily pravděpodobnostní rozložení nad možnými strategiemi zabijáka. V takovém případě je pro hráče nejlepší hodit si třístěnnou kostkou, kde každá strana kostky znamená volbu jednoho z mostů, a výsledek tohoto losu dodržet. Ve stejné situaci jako hráč je i zabiják, který nemá žádné zvláštní preference co se týče volby mostu a nemá ani informace o rozhodování hráče. Hodí si proto rovněž kostkou. Lze jednoduše spočítat, že pravděpodobnost přežití hráče je rovna  $\frac{2}{3}$  a pravděpodobnost smrti  $\frac{1}{3}$ . Hráč nemůže svůj užitek zlepšit žádným racionálním postupem a stejně tak zabiják. Tyto dvě náhodné strategie jsou tedy Nashovým equilibriem.

Vraťme se nyní k původnímu zadání problému s přechodem přes řeku. Uvažujme pravděpodobnost úspěšného přechodu přes most pod skálou 90% a pravděpodobnost úspěšného přechodu přes most s hady 80%. Předpokládejme dále, že hráče zajímá pouze zda přežije či ne. Pro zabijáka je důležitá pouze jistota, že je hráč mrtev. V tomto případě hráč opět použije náhodné rozhodování pomocí hodu kostkou, ale každý výsledek bude navíc ovlivněn vahou, která reprezentuje parametrické nebezpečí na každém ze tří mostů. Každý most lze považovat za loterii nad hráčovými možnými výstupy, kde každá loterie má rozdílný očekávaný užitek.

Podívejme se na situaci z pohledu zabijáka. Zabiják bude hrát strategii v NE, pokud zvolí pravděpodobnosti volby jednotlivých mostů tak, že pro hráče budou stejně výhodné všechny jeho ryzí strategie. Most pod skálou je 1,1 krát nebezpečnější než bezpečný most. Bude-li tedy zabiják čekat na bezpečném mostě s pravděpodobností 1,1 krát větší než na mostě pod skálou, bude pro hráče stejně výhodné zvolit kterýkoliv z těchto dvou mostů. Most s hady je 1,2 krát nebezpečnější než bezpečný most. Bude-li tedy zabiják čekat na bezpečném mostě s pravděpodobností 1,2 krát větší než na mostě s hady, bude pro hráče stejně výhodné zvolit kterýkoliv z těchto dvou mostů. Označme hráčovu parametrickou pravděpodobnost na přežití na jednotlivých mostech jako  $s_1$ ,  $s_2$  a  $s_3$ . Zabiják minimalizuje poměr pravděpodobností přežití hráče mezi každou dvojicí mostů

úpravou pravděpodobností  $p_1$ ,  $p_2$  a  $p_3$  tak, že platí:

$$s_1 \cdot (1 - p_1) = s_2 \cdot (1 - p_2) = s_3 \cdot (1 - p_3)$$

Zabiják nalezne svoje Nashovo equilibrium řešením soustavy rovnic:

$$1 \cdot (1 - p_1) = 0,9 \cdot (1 - p_2)$$

$$1 \cdot (1 - p_1) = 0,8 \cdot (1 - p_3)$$

$$p_1 + p_2 + p_3 = 1$$

Řešením této soustavy je:

$$p_1 = \frac{49}{121}$$

$$p_2 = \frac{41}{121}$$

$$p_3 = \frac{31}{121}$$

Nyní označme  $q_1$ ,  $q_2$ ,  $q_3$  pravděpodobnosti, s nimiž hráč zvolí jednotlivé mosty. Hráč pak nalezne své Nashovo equilibrium řešením soustavy:

$$1 \cdot q_1 = 0,9 \cdot q_2$$

$$1 \cdot q_2 = 0,8 \cdot q_3$$

$$q_1 + q_2 + q_3 = 1$$

Řešením této soustavy je:

$$q_1 = \frac{36}{121}$$

$$q_2 = \frac{40}{121}$$

$$q_3 = \frac{45}{121}$$

Tyto dvě skupiny NE pravděpodobností jsou vahami pro jednotlivé hody kostkou. Všimněme si, že hráč volí riskantnější z mostů s větší pravděpodobností. Tím je pro zabijáka stejně výhodné čekat na kterémkoliv mostu a zvyšuje se šance hráče na přežití.

### 2.7.1 Kardinální funkce užítku

Pouze pro ilustraci a úplnost se nyní podívejme na kardinální funkci užítku. Kardinální funkci užítku je třeba aplikovat v případech, kdy hráč rozlišuje nejen pořadí možných strategií vzhledem k zisku, který mu z těchto strategií plyne, ale také kvantitativní rozdíly mezi těmito zisky. Toto použití bude ilustrováno na příkladě.

Řešení problému prezentované v předchozím odstavci je založené na předpokladu, že se jedná o hru s nulovým součtem. Uvažovali jsme, že zisk jednoho hráče je roven ztrátě druhého hráče. To však nemusí být vždy pravda. Uvažujme situaci s komplikovanější funkcí užítku. Předpokládejme, že zabiják preferuje zabití hráče vlastní rukou. Nejméně výhodné je pro něj, když hráč unikne. Hráč preferuje být zastřelen před bolestivou smrtí kamenem nebo pomalou smrtí při uštknutí hadem. Ze všeho nejvíc si však přeje uniknout. Tuto hru nelze řešit stejným způsobem jako předchozí případ, neboť intenzity jejich preferencí budou ovlivňovat volbu jejich strategií.

Problémem takto specifikované hry je, že potřebujeme na místo ordinální funkce užítku funkci kardinální, která by dokázala postihnout i skryté psychické preference hráčů, jako je například potěšení. Tato teorie je poměrně komplikovaná, proto ji zde pouze nastíníme.

Mějme hráče, jehož ordinální funkce užítku je známá. Pro náš problém s přechodem řeky vypadá ordinální funkce užítku hráče takto:

- únik = 4,
- smrt zastřelením = 3,

- zabití kamenem = 2,
- uštknutí hadem = 1.

Dále víme, že rozdíl preferencí mezi únikem a zastřelením je větší než rozdíl preferencí mezi zastřelením a uštknutím. V chování hráče se to projeví tak, že hráč bude ochoten podstoupit mnohem větší riziko za získání větší šance k úniku proti smrti zastřelením, než bude ochoten podstoupit za větší šanci být zastřelen než uštknut hadem.

Začneme analýzu problému stanovením nejlepší a nejhorší možnosti (výhry) hráče. Racionální hráč bude vždy jednat tak, aby maximalizoval pravděpodobnost dosažení své nejlepší výhry ( $W$ ) a minimalizoval pravděpodobnost dosažení své nejhorší výhry ( $L$ ). Nyní uvažíme i možné výhry s ohodnocením mezi  $W$  a  $L$ . Dále nalezneme nad množinou všech výher takovou loterii, která je pro hráče ekvivalentní s loterií s výhrami pouze  $W$  a  $L$ . V našem případě je to loterie obsahující i možnosti zastřelení a zabití kamenem. Nazvěme tuto loterii  $T$ . Nyní definujeme funkci užitku  $q = u(T)$  tak, že je-li  $q$  očekávanou výhrou v  $T$ , pak je pro hráče stejně výhodné vyhrát v  $T$ , jako vyhrát v loterii ve které se  $W$  vyskytuje s pravděpodobností  $u(T)$  a  $L$  s pravděpodobností  $1 - u(T)$ .

Nyní zkonstruujeme složenou loterii  $T^*$  nad množinou výstupů  $\{W, L\}$  takovou, že je hráč indiferentní mezi  $T$  a  $T^*$ . Složená loterie je loterie, v níž je výhrou další loterie. Nyní můžeme analyzovat  $T^*$  a nalézt jednoduchou loterii nad  $W$  a  $L$ , nazvěme ji  $r$ . Ze zákona tranzitivity víme, že  $T$  je ekvivalentní  $r$ . Racionální agent nyní zvolí akci, která maximalizuje pravděpodobnost výhry  $W$ . Mapovací funkce z množiny možných výher na  $u(r)$  je von Neumann-Morgensternova funkce užitku (VNMfu).

Jediné co jsme udělali je, že jsme hráči dali na výběr mezi loteremi místo přímého výběru mezi výhrami. Nyní můžeme sledovat, jak moc je ochoten zvýšit své riziko pro získání jednotlivých možných výher.

VNMfu je spíše kardinální než ordinální funkce užitku. VNMfu je tedy schopná měřit relativní preference každého hráče. Použitým měřítkem je míra rizika, kterou je na sebe hráč ochoten vzít. Musíme tedy předpokládat, že hráčův postoj k riziku zůstává po celou dobu konstantní. To lze předpokládat pro jednoho hráče v jedné herní situaci, ale pro různé hráče, nebo stejného hráče v jiných podmínkách, se může postoj k riziku měnit. V našem problému s mosty bude přístup k riziku pro zabijáka, který neriskuje svůj život mnohem kladnější, než pro hráče, jemuž hrozí smrt. V problému s přechodem řeky však nepotřebujeme být schopni porovnat kardinální hodnoty užitku zabijáka s hodnotami užitku hráče a jejich postoj k riziku pro nás tedy není důležitý. Oba hráči najdou svoje NE, pokud dokáží určit pravděpodobnosti, které každý z nich přiřadí jednotlivým strategiím soupeře. To znamená, že každý z nich musí znát obě VNMfu.

### 2.7.2 Závěr problému s přechodem řeky

Na závěr doplníme matici z obrázku 2.1. Předpokládejme, že hráči záleží pouze na tom, jestli přežije nebo zemře a stejně tak i zabijákovi. Předpokládejme dále, že pravděpodobnost, že bude zabiják čekat na mostě s hady je  $x$ , na mostě pod skálou je  $y$  a na bezpečném mostě je  $1 - x - y$ . Hodnota užitku pro hráče pokud zemře je 0, a pokud unikne 1, pro zabijáka je tomu opačně. Kompletní matice je znázorněna na obrázku 2.8. Vidíme, že žádná strategie nedominuje strategii jinou, NE tedy leží ve smíšených strategiích.

		Killer		
		Safe bridge	Rocky bridge	Snake bridge
Player	Safe bridge	0, 1	1, 0	1, 0
	Rocky bridge	0.9, 0.1	0, 1	0.9, 0.1
	Snake bridge	0.8, 0.2	0.8, 0.2	0, 1

Obrázek 2.8: Problém přechodu přes řeku - úplná matice

		Player 2		
		Player 3		C
Player 1	A	E	4, 3, 5	3, 5, 2
		F	2, 4, 2	6, 1, 2
	B	E	1, 6, 3	5, 3, 1
		F	6, 4, 1	2, 6, 4

Obrázek 2.9: Matice hry tří hráčů

### 2.7.3 Postup a složitost řešení hry v normálním tvaru

V předchozích odstavcích jsme ukázali postup sestavení matice hry, definovali pojem Nashovo equilibrium a uvedli postup řešení matice hry v čistých strategiích. V tomto odstavci se budeme zabývat algoritmem pro řešení matice hry ve smíšených strategiích a odvodíme jeho složitost.

Uvažujme matici hry na obrázku 2.8. Tato hra má tři hráče a každý z hráčů má dvě strategie. Hráč 1 má strategie  $A, B$ , Hráč 2 má strategie  $C, D$ , a Hráč 3 má strategie  $E, F$ . V každé z buněk matice jsou uvedeny hodnoty funkce užítku pro každého z hráčů v pořadí Hráč 1, Hráč 2 a Hráč 3.

#### 2.7.3.1 Postup řešení

Prvním krokem řešení je stanovení množiny všech možných, čistých i smíšených kombinací strategií těchto tří hráčů. Tyto kombinace se nazývají *strategické profily* (v literatuře *strategic profiles*, *supports*). Označme  $S$  množinu všech strategických profilů. Potom

$$\begin{aligned}
S = & \{ \{A\} \times \{C\} \times \{E\} \}, \{ \{A\} \times \{C\} \times \{F\} \}, \{ \{A\} \times \{C\} \times \{E, F\} \}, \{ \{A\} \times \{D\} \times \{E\} \}, \\
& \{ \{A\} \times \{D\} \times \{F\} \}, \{ \{A\} \times \{D\} \times \{E, F\} \}, \{ \{A\} \times \{C, D\} \times \{E\} \}, \{ \{A\} \times \{C, D\} \times \{F\} \}, \\
& \{ \{A\} \times \{C, D\} \times \{E, F\} \}, \{ \{B\} \times \{C\} \times \{E\} \}, \{ \{B\} \times \{C\} \times \{F\} \}, \{ \{B\} \times \{C\} \times \{E, F\} \}, \\
& \{ \{B\} \times \{D\} \times \{E\} \}, \{ \{B\} \times \{D\} \times \{F\} \}, \{ \{B\} \times \{D\} \times \{E, F\} \}, \{ \{B\} \times \{C, D\} \times \{E\} \}, \\
& \{ \{B\} \times \{C, D\} \times \{F\} \}, \{ \{B\} \times \{C, D\} \times \{E, F\} \}, \{ \{A, B\} \times \{C\} \times \{E\} \}, \\
& \{ \{A, B\} \times \{C\} \times \{F\} \}, \{ \{A, B\} \times \{C\} \times \{E, F\} \}, \{ \{A, B\} \times \{D\} \times \{E\} \}, \\
& \{ \{A, B\} \times \{D\} \times \{F\} \}, \{ \{A, B\} \times \{D\} \times \{E, F\} \}, \{ \{A, B\} \times \{C, D\} \times \{E\} \},
\end{aligned}$$

$$\{\{A, B\} \times \{C, D\} \times \{F\}\}, \{\{A, B\} \times \{C, D\} \times \{E, F\}\}$$

V každém strategickém profilu se předpokládá, že pravděpodobnosti všech strategií obsažených v tomto profilu jsou nenulové. Za tohoto předpokladu sestavíme pro každý strategický profil soustavu rovnic, jejímž řešením jsou pravděpodobnosti jednotlivých strategií. Vezměme strategický profil  $\{\{A, B\} \times \{C, D\} \times \{E, F\}\}$  a označme  $\sigma(X)$  pravděpodobnost přiřazenou čisté strategii  $X$ . Uvažujme nyní Hráče 1.

Aby byl Hráč 1 indiferentní vůči strategiím  $A$  a  $B$ , musí být jeho očekávaný zisk  $\omega_1$  stejný při volbě obou strategií. Tedy:

$$\omega_1 = 4\sigma(C)\sigma(E) + 3\sigma(D)\sigma(E) + 2\sigma(C)\sigma(F) + 6\sigma(D)\sigma(F), \text{ pro strategii } A,$$

$$\omega_1 = \sigma(C)\sigma(E) + 5\sigma(D)\sigma(E) + 6\sigma(C)\sigma(F) + 2\sigma(D)\sigma(F), \text{ pro strategii } B.$$

Totéž platí i pro Hráče 2 a Hráče 3 a jejich očekávané zisky  $\omega_2, \omega_3$ . Tedy:

$$\omega_2 = 3\sigma(A)\sigma(E) + 4\sigma(A)\sigma(F) + 6\sigma(B)\sigma(E) + 4\sigma(B)\sigma(F), \text{ pro strategii } C,$$

$$\omega_2 = 5\sigma(A)\sigma(E) + \sigma(A)\sigma(F) + 3\sigma(B)\sigma(E) + 6\sigma(B)\sigma(F), \text{ pro strategii } D,$$

$$\omega_3 = 5\sigma(A)\sigma(C) + 2\sigma(A)\sigma(D) + 3\sigma(B)\sigma(C) + \sigma(B)\sigma(D), \text{ pro strategii } E,$$

$$\omega_3 = 2\sigma(A)\sigma(C) + 2\sigma(A)\sigma(D) + \sigma(B)\sigma(C) + 4\sigma(B)\sigma(D), \text{ pro strategii } F.$$

Kromě podmínky o nenulovosti všech  $\sigma(X)$  platí ještě podmínky:

$$\sigma(A) + \sigma(B) = 1,$$

$$\sigma(C) + \sigma(D) = 1,$$

$$\sigma(E) + \sigma(F) = 1.$$

Obdrželi jsme nelineární soustavu devíti rovnic o devíti neznámých. Řešením této soustavy obdržíme výsledky:

$$\sigma(A) = 0.7305, \sigma(B) = 0.2695, \sigma(C) = 0.2284,$$

$$\sigma(D) = 0.7716, \sigma(E) = 0.7169, \sigma(F) = 0.2831$$

a

$$\sigma(A) = 0.1097, \sigma(B) = 0.8903, \sigma(C) = 0.5587,$$

$$\sigma(D) = 0.4413, \sigma(E) = 0.3719, \sigma(F) = 0.6281.$$

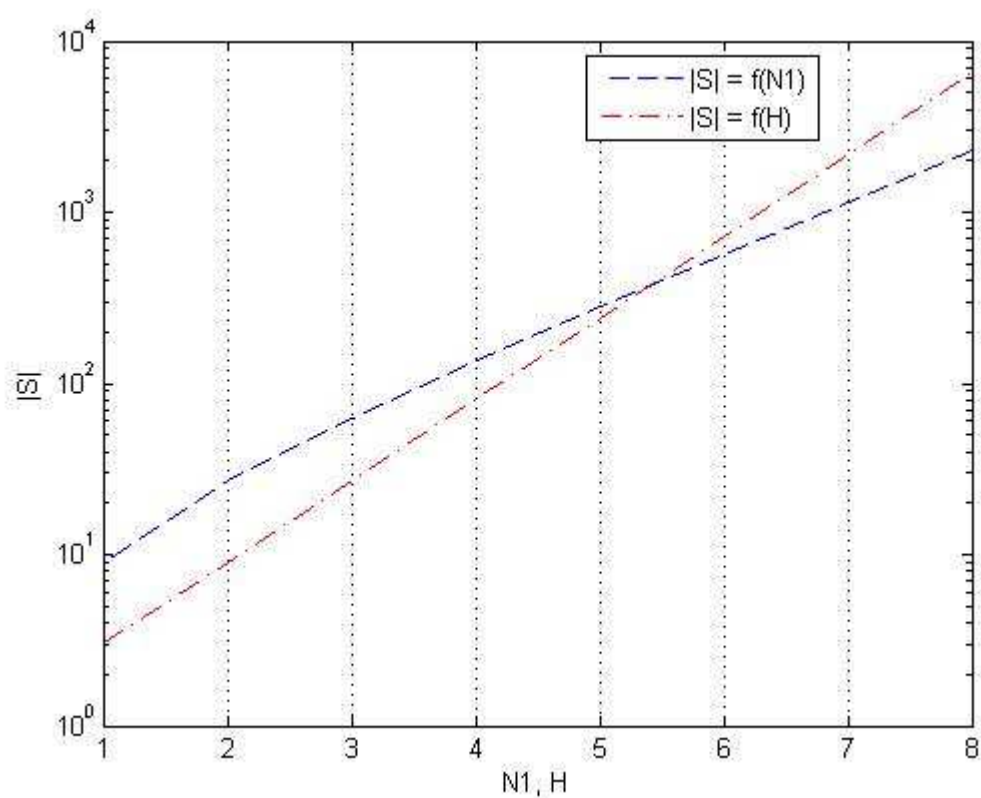
Vidíme, že obě řešení splňují všechny uvedené podmínky. Jedná se tedy o dvě z množiny smíšených Nashových equilibrií analyzované hry. Pro nalezení všech equilibrií je potřeba tento postup opakovat pro každý prvek množiny  $S$ .

### 2.7.3.2 Složitost

Jaký je vztah pro počet prvků množiny  $S$ ? Označme  $C(k, n)$  počet kombinací  $k$ -té třídy z  $n$  prvků bez opakování. Platí  $C(k, n) = \frac{n!}{k!(n-k)!}$ . Označme dále symbolem  $H$  počet hráčů ve hře a symbolem  $N_i$  počet strategií hráče  $i$ . Potom počet všech možných kombinací strategií hráče  $i$  je roven  $P_i = \sum_{k=1}^{N_i} C(k, N_i)$ . Každou z těchto kombinací je třeba analyzovat proti všem kombinacím strategií ostatních hráčů. Celkový počet všech prvků množiny  $S$  je tedy roven:

$$|S| = \prod_{i=1}^H P_i,$$

$$|S| = \prod_{i=1}^H \sum_{k=1}^{N_i} C(k, N_i),$$



Obrázek 2.10: Graf závislosti  $|S|$

$$|S| = \prod_{i=1}^H \sum_{k=1}^{N_i} \frac{N_i!}{k! (N_i - k)!}.$$

Na obrázku 2.10 je znázorněn graf závislosti  $|S|$  na počtu strategií Hráče 1 při konstantním počtu strategií ostatních hráčů  $N_i = 2$  a konstantním počtu hráčů  $H = 3$ . Dále je zde znázorněn graf závislosti  $|S|$  na počtu hráčů ve hře za předpokladu, že každý z hráčů má počet strategií  $N_i = 2$ . Vidíme, že  $|S|$  je exponenciálně závislá na počtu hráčů a přibližně exponenciálně závislá na počtu strategií každého hráče.

Z těchto údajů plyne veliká snaha o redukci matice hry s cílem snížit výpočetní složitost problému. Tato redukce je založena na postupné eliminaci dominovaných strategií, která v případech s dominantními strategiemi vede až na zredukovanou matici o jediné buňce.

## Kapitola 3

# Modelování oponentů v systémech s neúplnou informací

*V této kapitole se podíváme na alternativní přístup modelování inteligence v multi-agentních systémech. Popíšeme základní principy tohoto přístupu a podíváme se na jeho aplikaci v konkrétním multiagentním systému. Přestože hlavním tématem práce je aplikace matematické teorie her, přináší tento alternativní přístup několik zajímavých myšlenek, které se jako autor cítím být povinen zmínit. V dalších kapitolách bude nastíněno konkrétní použití tohoto přístupu v navrhovaném systému.*

Každý z nás se v životě jistě mnohokrát dostal do neznámé situace, kdy musel reagovat. Vezměme si situaci slečny jdoucí po ulici. Je pěkně oblečená a vypadá velmi přitažlivě. Po cestě jí jeden muž řekne: "Ahoj, dneska Ti to fakt sekne." Jak taková žena zareaguje? Možností je hned několik a její výběr mezi nimi je determinován několika faktory. Mezi tyto faktory patří její povaha, nálada, do které kategorie zařadí onoho muže a jaké má zkušenosti s danou kategorií mužů. Pokud je oním mužem její partner, bude to brát nejspíš normálně. Pokud je to její přítel nebo sice neznámý, ale atraktivní muž, bude se nejspíš cítit polichocena. Je-li to neznámý a normální muž, zřejmě to ponechá bez povšimnutí, a je-li to nějaký bezdomovec, bude se cítit pohoršena. Každopádně ji taková situace poprvé mírně zaskočí a bude se zamýšlet, jak ji řešit. Ocitne-li se však v této situaci poněkoličkáte, bude již reagovat zcela automaticky.

V předchozí situaci na reakci slečny až tak nezáleželo, ovšem existují situace, kdy je správná reakce otázkou života a smrti. Uvažujme dva boxery v ringu. Pokud jsou to soupeři, kteří spolu ještě nikdy neboxovali, stráví nejspíše prvních pár kol letným pobíháním, vzájemným oťukáváním a zjišťováním reakcí a slabin soupeře. S přibývajícím počtem informací o soupeři budou postupně upravovat svou taktiku boje a ten, který se dokáže lépe přizpůsobit soupeři, zvítězí. Uvažujme nyní zápas, kdy se utká vítěz tohoto zápasu s jiným boxerem, který mu však silně "připomíná" předchozího soupeře. Pak lze očekávat, že náš šampión stráví mnohem méně času oťukáváním. Jen se ujistí, že v několika základních reakcích je tento soupeř podobný předchozímu, a již bude vědět, jak postupovat dál podle modelu předchozího soupeře. Získání správné představy o protivníkovi je tedy jedním ze základních předpokladů úspěchu v každé hře, neboť nám umožňuje předpovídat reakce soupeřů na různé situace.

## 3.1 Case-based reasoning

Jednou z metod pro modelování soupeře v multiagentních systémech je Case-based reasoning (CBR). Z pohledu této metody je predikce akcí soupeře v situaci  $S$  otázkou správné klasifikace. CBR systém provádí porovnání situace  $S$  s databází odpozorovaných situací a situací  $S'$ , která je nejpodobnější situaci  $S$ , použije k rozhodování.

Jedním z klíčových faktorů ovlivňujících přesnost klasifikace CBR systémů je kvalita podsystému, který stanovuje míru podobnosti dvou situací. Bohužel implementace tohoto podsystému není triviální, neboť podobnost není pojmem absolutním, ale závisí vždy na kontextu. Měření podobnosti se musí přizpůsobit konkrétní herní situaci a roli, kterou v této situaci hraje agent, jehož chování se snažíme předpovědět. Uvedme si to na příkladu fotbalového zápasu.

Pokud se snažíme odhadnout chování brankáře týmu  $A$ , pak postavení obránců týmu  $B$  nebude mít prakticky žádný vliv. Naopak, snažíme-li se předpovědět chování útočníků týmu  $A$ , je rozestavení obránců týmu  $B$  podstatné. Jedním z možných řešení tohoto problému je různé vážení pozice obránců týmu  $B$  v každém z uvedených případů. Další možností je sledovat pouze pozici hráčů v blízkosti míče a ostatní hráče zanedbat.

V dalším popisu se budeme specializovat na náš příklad s fotbalovým utkáním. Jedná se o multiagentní systém se dvěma týmy po 11 hráčích na každé straně. Každý hráč je autonomní proces. To znamená, že dva hráči ve stejné situaci mohou mít rozdílné chování. Je tedy potřeba správně modelovat každého z oponentů odděleně.

Rozhodování každého z hráčů musí být v reálném čase a jeho výsledkem je posloupnost primitivních akcí (například utíkej, vyskoč nebo kopni), jejichž skládáním hráči realizují složitější akce jako dlouhé pasy, hlavičkování a další. Agenti se rozhodují na základě nekompletních a nejistých informací získaných pozorováním okolí v dosahu jejich zraku. Jedinými entitami s kompletní informací o všech hráčích v každém okamžiku hry jsou trenéři, které lze použít pro širší sběr informací o povaze jednotlivých hráčů a o strategii týmu jako celku.

### 3.1.1 Typy znalostí

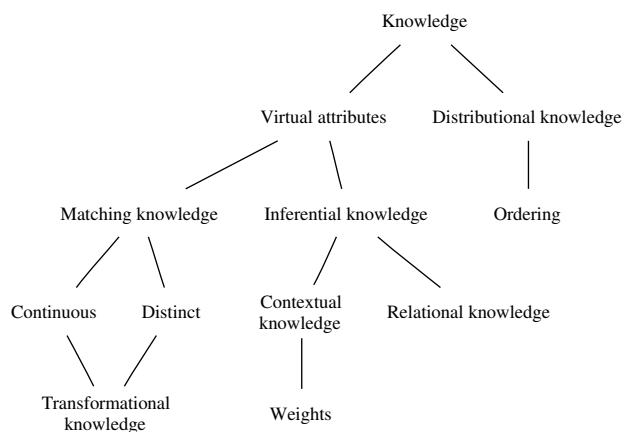
Existuje několik typů znalostí, které lze získat sledováním určitých situací. Jejich správná klasifikace a podrobnější analýza nám umožní zvolit ty informace, které jsou pro správnou klasifikaci situace podstatné a v dalším sledování se zaměřit pouze na sběr těchto informací.

V následujících příkladech demonstrujících jednotlivé typy znalostí, bude použita tato symbolika:  $C_1, C_2, C_3 \in R$  jsou spojité atributy.  $D_1, D_2 \in Z$  jsou diskrétní atributy.  $P(x)$  je binární pojem aplikovatelný na instanci  $x$ .  $C_i(x)$  nebo  $D_i(x)$  udává hodnotu atributu  $C_i$  nebo  $D_i$  instance  $x$ .  $\varpi \in R$  je váha.

Jednotlivé typy znalostí lze hierarchicky uspořádat (viz obrázek 3.1). Na nejvyšší úrovni rozlišujeme *virtuální (odvozené) atributy* a *distribuční znalosti*. Distribuční znalosti zahrnují znalosti o rozsahu a rozložení jednotlivých atributů a jejich hodnot. Znalost rozsahu hodnot může být použita k normalizaci a k přesnějšímu určení podobnosti dvou situací.

Virtuální atributy jsou atributy, které nejsou v systému přímo obsaženy, ale lze je odvodit z jiných atributů. V CBR systémech jsou tyto atributy užitečné v případech, kdy klasifikace nezávisí na konkrétní hodnotě daného atributu, ale na relacích mezi atributy. Například, je-li  $C_1$  pozice hráče  $A$  a  $C_2$  pozice hráče  $B$ , pak virtuální atribut  $C_3(x) = C_1(x) - C_2(x)$  je vzdálenost těchto hráčů.





Obrázek 3.1: Hierarchie typů znalostí

Virtuální atributy lze dále rozdělit na *znalosti o shodě* a *dedukční znalosti*. Znalost o diskrétní shodě určuje, zda jsou si dva atributy rovny nebo ne. Mezi diskrétní znalost o shodě patří například taxonomie. Znalost o spojitě shodě definuje určité oblasti v prostoru atributů. Příklady:

- $C_1(x) < 30 \wedge C_1(x) < 50$  (spojité)
- $D_1(x) \equiv D_2(x)$  (diskrétní)

Znalost o shodě může být použita k porovnání dvou syntakticky jiných atributů, které jsou ovšem sémanticky ekvivalentní. Dvě různé postavy tak mohou být klasifikovány jako shodné, pokud zastávají stejnou roli (například obránce).

Znalosti o transformaci jsou speciálním typem znalostí o shodě, kde mapujeme jeden bod z prostoru situací na bod jiný obvykle s pomocí aritmetických operací. Příkladem takové operace může být rotace, která umožňuje v našem fotbalovém příkladě považovat za shodné dvě situace, které se vyskytly na opačných křídlech.

Dedukční znalosti udávají hodnotu atributu, který je odvoditelný z hodnot některých jiných atributů. Příklad:  $P(x) \leftarrow C_1(x) > 30 \wedge C_1(x) < 50$ . Přičemž pro vyhodnocení podobnosti podmínkové části se používá znalostí o shodě. Příkladem dedukční znalosti je odvození offsidu jako virtuálního atributu z přesné pozice míče a pozice hráče.

*Kontextové znalosti* jsou speciálním případem dedukčních znalostí. Kontextová znalost říká, že určitý znak je důležitý v závislosti na některých jiných znacích. Příklad:  $important(P(x)) \leftarrow C_1(x) > 30 \wedge C_2(x) < 50$ . Kontextové znalosti nám umožňují určit, že pozice obránců týmu  $A$  není důležitá, pokud je tým  $A$  v útoku.

*Váhy* jsou podmnožinou kontextových znalostí. S jejich využitím jsme schopni vyjádřit důležitost nějakého atributu na globální úrovni  $important(P(x), \varpi) \leftarrow TRUE$ , nebo na lokální úrovni  $important(P(x), \varpi) \leftarrow C_1(x) > 30 \wedge C_2(x) < 50$ . Kontextové znalosti a váhy jsou obecně znalostmi o významnosti atributů.

*Relace* jsou další podmnožinou dedukčních znalostí. Podmínková část obsahuje nejméně dva různé atributy. Relace mezi atributy jsou při určování podobnosti základním nástrojem pro výpočetní modelování lidské kategorizace. Příklad:  $P(x) \leftarrow C_1(x) > C_2(x)$ . Relace opět využívají znalostí o shodě k určení podobnosti podmínkové části, která vyznačuje oblast v níž je daná

relace platná. Relace jsou nezbytné například při určování, zda se hráč nachází mezi míčem a bránou.

### 3.2 Ukazatel podobnosti

Získané znalosti o dané situaci jsou vstupem podsystému pro stanovení míry podobnosti dvou situací. Tento podsystém používá pro vyhodnocování ukazatel podobnosti, který udává míru podobnosti dvou situací. Při hledání nejpodobnější situace z databáze situací vypočteme ukazatel podobnosti právě analyzované situace se všemi situacemi v databázi a situaci s nejvyšší hodnotou ukazatele podobnosti označíme za výsledek hledání.

Základní ukazatel podobnosti pro náš příklad fotbalového zápasu je definován takto:

$$\begin{aligned} sim(S_1, S_2) = & \sum_{i=1}^{22} [\varpi_i * \Delta(p(i, S_1), p(i, S_2)) + \varpi_i * \Delta(v(i, S_1), v(i, S_2))] + \\ & + \varpi_0 * \Delta(bp(S_1), bp(S_2)) + \varpi_0 * \Delta(bv(S_1), bv(S_2)) \end{aligned}$$

V tomto vztahu jsou  $S_1, S_2$  dvě situace, které porovnáváme,  $p(i, S_j)$  a  $v(i, S_j)$  jsou pozice a rychlost hráče  $i$  v situaci  $S_j$ ,  $bp(S_j)$  a  $bv(S_j)$  jsou pozice a rychlost míče v situaci  $S_j$ .  $\Delta(A, B)$  je Euklidovská vzdálenost mezi  $A$  a  $B$ , a  $\varpi_k, \varpi_k'$  jsou váhy pro pozice a rychlosti, přičemž  $\sum (\varpi_k + \varpi_k') = 1$ . Tyto váhy určují relevanci jednotlivých atributů.

Kromě tohoto základního ukazatele existuje ještě rozšířený ukazatel podobnosti. Ten lze vyjádřit vztahem:

$$sim(S_1, S_2) = \sum_{i=1}^n (\varpi_i * s_i)$$

kde  $s_i$  jsou lokální hodnoty ukazatelů podobnosti a  $\varpi_i$  jsou odpovídající váhy. Podrobnější popis tohoto rozšířeného ukazatele pro demonstrační příklad fotbalového zápasu lze najít v [12].

### 3.3 Porovnávání situací v multiagentních systémech

Ve většině CBR aplikací je prosté porovnání atributů nejpřímochařejším řešením. Bohužel v multiagentních systémech není porovnání atributů triviální. Například postavení nebo rychlost je v každé z pozic přiřazena konkrétnímu z 22 hráčů na hrací ploše. Uvažujme dvě pozice  $S_1, S_2$ . Porovnání těchto dvou pozic musí brát v úvahu i možnost prohození pozic dvěma hráči. Řekněme, že si během hry hráči 3 a 9 vyměnili své pozice. Potom při určování podobnosti těchto dvou pozic nesmí být porovnávány atributy hráče 3 v situaci  $S_1$  s atributy hráče 3 v situaci  $S_2$ , ale s atributy hráče 9. Proto před porovnáním dvou pozic musíme nejprve provést porovnání hráčů a určit tak odpovídající protějšky v obou situacích. Stanovení odpovídajících si protějšků v situacích, v nichž se vyskytuje mnoho agentů, je zřejmě velmi náročné a to jak na implementaci, tak i na dobu výpočtu.

# Kapitola 4

## Počítačové hry

*Titul práce zní Strategická počítačová hra. Abychom byli schopni tento typ her správně zařadit, podíváme se nyní na základní členění počítačových her a blíže potom na členění her strategických. Stručně se seznámíme s charakteristickými rysy jednotlivých druhů her a uvedeme některé zástupce.*

Počítačové hry jsou v podstatě modely naší reality. Různé počítačové hry se zabývají modelováním různých částí naší reality a využívají k tomu různé techniky. Výsledky počítačových her přitom mohou mít velký význam pro naši realitu. Například výsledky testů na trenažérech pilotů mohou ukázat pilotovi nedostatky bez toho, aby byly vystaveny ohrožení životy pasažérů. Význam těchto výsledků roste s realističností celého modelu.

Počítačových her existuje celá řada a nových titulů neustále přibývá. V této kapitole se podíváme na základní rozdělení počítačových her, stručně přiblížíme jednotlivé druhy, naznačíme oblasti, v nichž se v těchto hrách uplatňuje umělá inteligence, a podíváme se na bližší popis her, které posloužili jako inspirace pro návrh implementovaného herního systému.

### 4.1 Typy počítačových her

Počítačové hry lze podle charakteru rozdělit do čtyř základních skupin:

- *Akční hry* využívají k pobavení hráče především momentů překvapení, nutnosti rychlé reakce a zvýšení adrenalinu hráče.
- *Adventury* jsou postaveny na výpravném příběhu a řešení nejrůznějších logických úkolů.
- *RPG hry* umožňují vývoj charakterů postav a přizpůsobení příběhu povaze hráče.
- *Strategické hry* jsou založeny na ovládnutí velké skupiny jednotek s cílem maximálně využít jejich specifické schopnosti.

#### 4.1.1 Akční hry

Akční hry jsou typem her, který se v poslední době těší zřejmě největší popularitě a to díky své rychlosti, přímočarosti, napínavosti, relativní jednoduchosti a možnosti síťových her přes Internet

s dalšími lidskými soupeři a spoluhráči. Společným rysem většiny akčních her je hra z pohledu první osoby.

První akční hrou, kterou lze pokládat za hru zakládající toto odvětví, byla legendární hra Wolfenstein, kterou o pár let později následoval Doom a Doom2. Cíl hry byl jednoduchý: sbírat zbraně, střílet všechny nepřátele, kteří se na hráče slepě valí, nenechat se zabít a pokud možno se dostat do dalšího kola. Od těch dob se však mnohé změnilo. Kromě neustále se zlepšující grafiky se s rostoucí výpočetní kapacitou počítačů zlepšil i fyzikální model reálného prostředí a inteligence počítačových hráčů.

V současnosti nejpoužívanější architekturou pro kontrolu NPC (non-player character), je architektura postavená na principu client-server, která byla vynalezena na Institutu výpočetní techniky University of Southern California's. Tato architektura nese název GameBot.

Serverová strana tohoto modelu se stará o monitoring aktuálního prostředí (například poloha soupeřů, které postava vidí, poloha překážek, aktuální zdraví postavy a další). Klientská strana se stará o rozhodování v závislosti na informacích získaných o prostředí a zasílá serverové straně zprávy o akcích, které má postava vykonat (například vystřelit, pohnout se atd).

K rozhodování na klientské straně aplikace jsou v architektuře GameBot použity rozhodovací stromy spolu s algoritmem ITI (Incremental Tree Inducer). NPC je v této architektuře implementována jako stavový stroj s několika nezávislými stavy, které jsou aktivovány pomocí výstupů z rozhodovacího stromu. Ke každému z těchto stavů je přiřazeno jednoduché chování (například prozkoumávej mapu, útoč, utíkej a další). K sestavení rozhodovacího stromu, který by zajistil NPC inteligenci chování odpovídající zdatnému lidskému hráči, je zapotřebí několik kroků. Nejprve je potřeba sestavit množinu možných strategií pro NPC v různých herních situacích od zkušených lidských hráčů. Formou dotazníku se sestaví množina klíčových situací a hráči rozhodnou, jak by se v každé ze situací zachovali. Na základě takto získaných dat je potom ITI algoritmem sestaven rozhodovací strom, který je posléze začleněn do herního systému. Pro bližší popis této metody viz [7].

### 4.1.2 Adventury

Adventury jsou typem her bez umělé inteligence, kdy hráč ovládá většinou jednu postavu a snaží se řešením různých logických úkolů a sbíráním indicií a předmětů splnit hlavní dějový úkol. Adventura je žánr, který je rozvláčný a příběhově bývá zajímavý a spletitý. Řešení hádanek a sledování hlavního děje je hlavním zdrojem zábavy u tohoto typu her. Jedním z příkladů takovéto hry je česká hra Horké léto nebo zahraniční hry Agent Mlíčňák a The Neverhood.

### 4.1.3 RPG (role playing game)

RPG je druh hry, kde hráči zaujímají role fiktivních postav, které si podle daných pravidel vytvoří a za které potom během hry jednájí. Hráč je během hry postaven před celou řadu problémů, kde se musí rozhodovat o dalším postupu, přičemž nejsou zcela zřejmé všechny důsledky jednotlivých možných rozhodnutí. Každé z těchto rozhodnutí má vliv jak na postavu, tak na celý fiktivní svět a ovlivňuje hlavní příběh. Nad fiktivním světem existuje množina pravidel, často velmi spletitá, která determinuje vývoj světa a postav v závislosti na rozhodnutích hráče. Tato množina pravidel a systém odvozování závěrů podle rozhodnutí hráče bývá jediným prvkem, který lze považovat za inteligentní. Klasickým příkladem počítačové RPG hry je Fallout.

#### 4.1.4 Strategické hry

Strategické počítačové hry jsou obecně hry, u nichž hráč ovládá větší skupinu objektů a manipuluje s nimi po hrací ploše tak, aby utrpěl co nejmenší újmu na vitalitě, bojeschopnosti, zdrojích, území a jednotkách a zároveň se snaží tuto újmu přivodit soupeři. Strategické hry lze dělit do několika podskupin podle způsobu hraní a herních cílů.

- *Tahové strategie* jsou hry, v nichž se hráči postupně střídají v provádění svých tahů. Na rozdíl od deskových her však hráči nemusí znát všechny tahy provedené soupeřem.
- *Real-time strategie* jsou hrami, v nichž jednotliví hráči provádějí své tahy současně v reálném čase a jsou tak nuceni rychle reagovat na celou řadu podnětů.
- *Budovatelské strategie* jsou založené na složitém systému vztahů a požadavků, kterým se musí hráč co nejvíce přiblížit, aby vybudoval funkční a stabilní systém.
- *Simulátory* se snaží o maximální realističnost a detailnost rozpracování velmi specifického tématu z hráčsky atraktivních oblastí naší reality.
- *Deskové hry* jsou založené na postupném střídání hráčů v provádění jednotlivých tahů, přičemž mají hráči informace o všech předchozích tazích soupeřů.

V dalším textu rozvedeme jednotlivé typy strategických her, neboť jsou klíčové pro tento projekt.

##### 4.1.4.1 Tahové strategie

Tahové strategie, jak již název napovídá, jsou typem strategií, kdy se hráči střídají po jednotlivých kolech. Tento přístup se rozvinul především kvůli hrám s detailně propracovaným managementem, který je příliš složitý na to, aby se dal zvládat v reálném čase jako je tomu u žánru real-time strategií. Jednou z legendárních tahových strategií je hra Civilizace z dílny Sida Meiera, která vyšla v několika dílech. Podívejme se v krátkosti na třetí díl.

Hráč je postaven do role panovníka jedné ze starověkých civilizací a jeho cílem je vést svůj národ k prosperitě a vojenské, technologické, územní nebo kulturní dominanci nad dalšími civilizacemi. K dosažení svých cílů se musí starat o celou škálu rozdílných odvětví. Mezi ně patří volba politického režimu, rozvoj měst, přerozdělování finančních prostředků státní pokladny na vědu a výzkum nebo na sociální dávky obyvatelstvu, zajišťování surovin, vedení výzkumu, stavba infrastruktury impéria, zušlechťování krajiny, budování armády a diplomatické vztahy s jinými civilizacemi jako jsou práva přechodu přes cizí území, smlouvy o neútočení, dovozní embarga, vojenské koalice a obchodní smlouvy.

Jak je z uvedeného popisu patrné, je efektivní správa impéria velice složitou záležitostí a z toho vyplývá i složitost umělé inteligence počítačových hráčů, která musí být propracovaná ve všech těchto oblastech.

Na závěr tohoto odstavce se ještě zmiňme o hře X-COM Apocalypse, což je zatím jediná hra, kde hráč může během hry plynule přepínat mezi režimem real-time a tahové strategie.

#### 4.1.4.2 Real-time strategie

Real-time strategie jsou strategie odehrávající se v reálném čase. Na rozdíl od tahových strategií je okruh úkolů, jimž se hráč musí věnovat, podstatně omezen. Na druhou stranu je potřeba většího přehledu o situaci a rychlejšího rozhodování ze strany hráče. Nejčastějším typem real-time strategií jsou hry na principu Warcraftu společnosti Blizzard.

Společným rysem těchto her je těžba několika málo surovin, rychlá výstavba základny a následné vedení vojenských operací proti nepříteli. Umělá inteligence počítačových hráčů bývá u těchto her bohužel velmi slabá.

Ve hře Warcraft měl počítačový hráč předdefinované jak často má útočit, s jakou silou a do kterých míst. Zároveň stavěl jednotky pouze pro útok, nikdy pro obranu. Pokud se hráči podařilo zničit jednotky, které měl počítač od začátku hry k dispozici jako obranu, měl úlohu již velmi jednoduchou. V dalších hrách, jako Command Conquer, se programátoři snažili nedostatky umělé inteligence vyrovnat nejrůznějšími zvýhodněními počítačového hráče.

Inteligence těchto her příliš nepostoupila ani v nejnovějších hrách, jako je Warcraft III. Asi nejnáročnější hrou s nejlepší umělou inteligencí počítačových hráčů, se kterou jsem se doposud setkal, byl datadisk Broodwar ke hře StarCraft společnosti Blizzard.

#### 4.1.4.3 Budovatelské strategie

Ve všech budovatelských strategiích začíná hráč s jistým počátečním stavem financí, které musí investovat k výstavbě funkčního a vyváženého systému s dlouhodobě udržitelným rozvojem. Existuje mnoho budovatelských strategií a v každé z nich je tímto systémem něco jiného. Počínaje zábavním parkem přes nemocnici až po celé město.

Jednou z prvních budovatelských strategií byla hra SimCity - simulátor města. Hráč je zde v roli úřadu pro územní plánování. Staví infrastrukturu města, volí vhodná místa pro obytné, průmyslové a obchodní zóny, stará se o rozmístění nemocnic a požární a policejní služby, stejně jako o platy jejich zaměstnanců. Na podobném principu je založena i herní série Caesar.

Další oblíbenou budovatelskou strategií je hra Transport Tycoon. Zde je hráč v roli šéfa přepravní společnosti. Jeho cílem je vystavět vhodnou infrastrukturu a pokrýt ji svými dopravními prostředky, aby zajistil rychlou a efektivní přepravu zboží a osob, z níž mu plyne příjem. Z her uvedených v tomto odstavci je Transport Tycoon jediným zástupcem s transparentně implementovaným počítačovým protihráčem. Hráč tak může sledovat nejen grafy úspěšnosti umělých protihráčů, ale i jejich konkrétní akce.

Mezi další oblíbené budovatelské strategie patří hry Theme Hospital (simulátor nemocnice) nebo RollerCoaster Tycoon (simulátor zábavního parku).

#### 4.1.4.4 Simulátory

Přestože jsme v předchozím odstavci používali pojem simulátor, hráči počítačových her si většinou představí něco jiného než budovatelskou strategii. Nejspíše se jim vybaví některý z úspěšných leteckých simulátorů, nebo některá ze sportovních her ze sérií pod tituly FIFA, NHL nebo NBA.

Ve všech těchto hrách ovládá člověk v daném okamžiku jen jednu postavu z celého týmu. Všechny ostatní postavy, ať už ze stejného nebo protivnickova týmu, jsou ovládány počítačem. Hráč má možnosti nastavit základní strategie svého týmu (například míru ofensivnosti hráčů,

jejich základní rozestavení a oblasti působnosti), ale jejich chování během samotné hry je již pod kontrolou počítače, který se stará o rozmístění, manévrování atd.

#### 4.1.4.5 Deskové hry

Deskové hry patří mezi tahové strategie. Hráči se tedy ve hře postupně střídají. Přitom má každý z nich dokonalý přehled o veškerém dění na hrací ploše po celou dobu hry. Počet možných tahů hráče bývá relativně malý, proto se k modelování umělé inteligence dají efektivně využít  $n$ -ární stromy a algoritmy nad těmito stromy (Mini-Max, Alfa-Beta).

K podpoře těchto algoritmů bývá využíváno ještě dalších prostředků. Například šachové programy mají k dispozici širokou databázi zahájení spolu s hodnocením jejich výhodnosti pro jednotlivé strany a dále databáze nejrůznějších modelových pozic. Počítač pak může během provádění algoritmu kontrolovat, zda se při výpočtu nedostal do některé z pozic, která svým charakterem odpovídá jemu známé modelové pozici (viz 3.2) a využít tuto znalost k ohodnocení vypočteného stavu.

## 4.2 Inspirace vlastního herního systému

Jak již bylo řečeno, počítačových her je mnoho a neustále přibývají další. Většinu z nich však lze považovat za vylepšené kopie již starých a dobře známých her. Není divu. Vymyslet zcela novou hru s naprosto novou myšlenkou nebo herním pojetím je nelehký úkol a mnoho takovýchto pokusů ztroskotalo na nezájmu hráčů. Proto byly jako inspirace návrhu prezentovaného herního systému vybrány dvě starší počítačové hry. Jsou to hry *Settlers 2* a *Majesty*. V následujících odstavcích se na tyto hry podrobněji podíváme. Smyslem tohoto přiblížení je pouze popis prvků a myšlenek, které tyto hry přinášejí, nikoliv jejich technický nebo implementační rozbor. V těchto odstavcích jsou použity některé termíny a výrazy, které lze běžně najít v recenzích na počítačové hry.

### 4.2.1 Settlers 2

V naší klasifikaci her patří *Settlers 2* do real-time budovatelských strategií. Jak již název hry napovídá, cílem hry je osídlování nových území. Jedná se o strategii s poměrně detailně rozpracovaným dodavatelsko-odběratelským systémem.

Hráč je postaven do role vládce království. Na počátku hry má pouze svůj zámek, malou část okolního území a počáteční zásoby materiálu jako dřevo, kamení, lopaty, krumpáče, otroky a další. S jejich využitím musí postupně budovat své království. Na nejnižší úrovni zásobovacího řetězce zakládá farmy na produkci obilí a dílny dřevorubců, lesníků, horníků a kameníků pro zajištění nerostných surovin a dřeva. Na vyšší úrovni se stará o stavbu řemeslnických dílen jako jsou truhláři, kováři, mincovny, zbrojírny a další, stejně jako mlýny, pekárny nebo vepřiny pro zajištění potravy. Produkce většiny těchto dílen je závislá na stejných surovinách. Tím vzniká problém vzácnosti a nedostatkovosti zboží a zdrojů. Řešení tohoto problému je opět na hráči. Ten má možnost pomocí priorit produkce a přerzdělování zásob efektivně řídit tok materiálu mezi manufakturami a zajistit dostatečnou výrobu a správnou distribuci veškerého zboží.

Na nejvyšší úrovni výrobního řetězce stojí armáda, která již neprodukuje žádné materiální výstupy. Hráč jako vládce volí vhodné rozmístění vojenských základů s cílem obsadit nová území a získat tak více surovin, stejně jako bránit své panství před nepřáteli. Vojenské jednotky nejsou panovníkem přímo najímány, ale vznikají jako výstupy nižších úrovní zásobovacího řetězce, kde

při určitém množství vyrobených štítů, mečů a volných otroků vznikne nová jednotka voják, která je připravena do služby.

Dobře propracovaný systém dodavatelsko-odběratelských vztahů byl inspirací ke snaze vybudovat obdobný systém, avšak nikoliv na principu přerozdělování a priorit určených hráčem, ale na principech tržní ekonomiky. V tomto tržním pojetí by se každý z řemeslníků musel sám rozhodovat o objemu své produkce a cenách, ze něž bude své produkty prodávat.

Nedostatkem hry Settlers je slabě propracovaná část týkající se vojenských operací. Vojáci bojují vždy pouze jeden proti jednomu a to zřejmě bez přesněji určených pravidel. To lze považovat za jediné vysvětlení situace, kdy jednou dokáže bránící voják s nižší hodnotou zabít čtyři útočící elitní vojáky a ubránit tak svoji pevnost, a podruhé prohrát pouze s jedním vojákem. Jako inspirace pro pojetí této části hry posloužila další z již existujících počítačových her - hra Majesty.

### 4.2.2 Majesty

Majesty je oproti Settlers méně známá, ale z hlediska programátorského mnohem zajímavější hra. Hráč je opět postaven do role vládce království, ale princip hry je od Settlers zcela odlišný. Jedinou surovinou je zlato, které hráč získává formou daní od svých poddaných a z prodeje služeb v království, zatímco poddaní jej získávají za plnění nejrůznějších úkolů a z pokladů nalezených v herním světě.

Na počátku hry má hráč k dispozici svoji pevnost a základní sumu peněz, za niž postupně rozvíjí své království. Na výběr má z několika typů budov, v nichž přebývají nejrůznější hrdinové, od rytíře přes lučištníka a paladina až k mocným kouzelníkům. Kromě těchto obydlí může budovat ještě tržiště, zbrojírnu, knihovnu, hospodu, královské zahrady, tréninkové bojiště a další budovy pro veřejné účely. V těchto veřejných budovách mohou postavy nakupovat služby. O úroveň nabízených služeb se stará hráč a tržby za prodej těchto služeb jdou do státní pokladny.

Kromě zlepšování služeb a stavby budov se hráč stará ještě o najímání hrdinů. Tyto hrdiny však nemůže hráč nijak interaktivně ovládat. Může je pouze motivovat vypisováním finančních odměn na plnění různých úkolů. Všechny postavy v království se chovají na základě své vlastní inteligence, svých preferencí a motivace. Kromě odměn vypsanych hráčem jsou postavy motivovány vlastními potřebami a svým životním posláním. Například pro paladina je otázkou cti bránit svého krále a bojovat proti silám zla, které jsou představovány upíry, kostlivci a dalšími nemrtvými, zatímco lupiče zajímají pouze finanční výhody a dá se velmi snadno koupit. Postavy mezi sebou dokáží uzavírat i jednoduché aliance, kdy se například barbar vydává na průzkum nepřátelského území a za ním jde kouzelník, který ho podporuje svými kouzly. O odměny se pak společně rozdělí. Za plnění úkolů získávají postavy kromě peněz (případně i artefaktů), také zkušenosti a tím rozvíjejí své schopnosti.

Budov a hrdinů existuje celá řada a jsou rozděleny do vývojových větví. Volbou jednoho typu chrámu si tak hráč znemožní stavbu chrámů jiných. Každá z vývojových větví nabízí jinou sestavu postav s odlišnými schopnostmi a povahami. Barbaři jsou silní, ale hloupí válečníci, kteří se bezhlavě vrhají do každého boje a jen málokdy nakupují vybavení a lektvary života. Okultisti jsou hbití, ale slabí kouzelníci, kteří využívají svých schopností získat kontrolu nad zvířaty a ty pak bojují na jejich straně. Nekromanceři vyvolávají hordy kostlivců a sají život z nepřátel atd.

Přestože je hra Majesty velmi originální a dobře propracovaná, postrádá ekonomický základ celého království. Nikdo nic nepěstuje, nespotřebovává se žádný materiál a postavy nikdy neji ani



nepijí. Velkým nedostatkem je pak místy slabá umělá inteligence postav. Často tak můžeme vidět rytíře bojujícího se stromovcem, který již ví, že nemůže stromovce porazit, ale přesto pokračuje v boji a nechá se zabít, i když se dokáže pohybovat rychleji než nepřítel a jistě by mu unikl. Stává se rovněž, že mocný kouzelník, který má být nejinteligentnější postavou v království, se opakovaně snaží čarovat útočná kouzla na upíra s magickým zrcadlem. Přesto že se několik kouzel odrazí zpět na kouzelníka a způsobí mu těžká zranění, nenapadne ho, že jeho počínání nemá smysl a klidně se takto sám zabije. Rovněž špatně vyřešené je rozdělování odměn za splněné úkoly, kdy všechny postavy v určité vzdálenosti od místa splnění úkolu mají stejný podíl na odměně.

## Kapitola 5

# Návrh systému

*V této kapitole se budeme věnovat specifikaci navrhovaného herního systému. Seznámíme se s jednotlivými prvky tohoto systému a pokusíme se identifikovat základní oblasti, v nichž budou od systému vyžadovány projevy inteligence.*

Návrh herního systému se opírá o výše uvedené hry Settlers a Majesty. Mou snahou je spojit obě hry do jednoho kompaktního celku, odstranit jejich nedostatky a navrhnout maximálně automatizovaný systém simulující království. Celý systém je velmi rozsáhlý, je proto třeba stanovit základní množinu požadavků na systém a jejich priority. Mezi tyto požadavky patří:

- *rychlost* - nezbytná pro činnost systému v reálném čase,
- *přehlednost* - umožňující efektivní nastavování a ladění parametrů systému,
- *snadná rozšiřitelnost* - s cílem udržet systém maximálně otevřený a umožnit pozdější doplnění dalších prvků do systému,
- *kvalitní a jednoduché uživatelské rozhraní* - pro snadné zadávání problémů a přehledný výstup,
- *vysoká úroveň inteligence* - jako základní požadavek na herní systém s počítačovými hráči,
- *vysoká míra automatizace* - jako projev dosažené míry inteligence a nezávislosti systému na zásahu hráče.

Celá hra má probíhat v reálném čase, *rychlost* odezvy a generování výstupů je tak jedním ze základních požadavků. Optimalizace na rychlost však většinou bývá na úkor paměťové náročnosti nebo přehlednosti celého programu. Proto je rychlosti v rámci tohoto projektu věnována pouze přiměřená pozornost.

*Přehlednost* celého kódu je důležitá pro snadné ladění, optimalizaci a rozšiřitelnost. K dosažení přehlednosti jsou použity techniky modulárního programování a techniky objektově orientovaného programování zahrnující zapouzdření a dědičnost. Jako programovací jazyk s ohledem na tyto požadavky byl použit jazyk C++.

Tento projekt je navržen jako experimentální systém. Měl by proto umožnit relativně jednoduché navození případové situace, jejíž výsledky jsou předmětem zájmu konkrétního experimentu.

K tomu je potřeba především *kvalitní a jednoduché uživatelské rozhraní*. Velkou roli hraje rovněž *rozšiřitelnost* celého systému, která umožní co možná nejjednodušší a nejefektivnější rozšiřování množiny možných experimentů. Požadavky na *inteligenci* a *vysokou míru automatizace* jsou pak esenciálními požadavky pro systém založený na umělé inteligenci.

V následujících odstavcích se na systém podíváme podrobněji z hlediska jeho stavby a návrhu a v další kapitole se zaměříme na popis řešení některých částí systému. Pro specifikaci návrhu systému je použita terminologie odpovídající oblasti do níž navrhovaný systém spadá.

## 5.1 Základní rysy systému

Většina existujících počítačových her je založena na dokonalé a sdílené informaci o systému. Jakmile jedna z postav vykoná nějakou akci, všechny ostatní spřátelené postavy o ní okamžitě vědí. Když postava prozkoumává nějakou oblast, všechny další postavy vědí, co v této oblasti nalezla. Stejně tak všechny postavy ví, kde se právě bojuje atd.

Mojí snahou bylo celý systém založit na principech nedokonalé a neúplné informace. Každá postava se chová jako samostatný agent a všechny informace, které o systému nasbírala, slouží pouze jí. To pochopitelně vede k zastarávání informací a ke vzniku nepřesností a omylů. Uvedme jednoduchý příklad.

Postava prochází místem *A* na mapě a vidí zde spřátelenou budovu, kde je možné zakoupit potraviny. Od místa *A* se vzdálí a vykonává nějakou další činnost. Mezi tím se v místě *A* odehraje bitva a daná budova bude během boje zničena. Naše postava však tuto informaci nemá. Stane-li se potom, že tato postava bude nutně potřebovat nakoupit potraviny a nejlepší možnost založená na jejích informacích bude nákup právě v této budově, bude po příchodu na místo *A* nepříjemně překvapena.

Systém individuálních informací by byl sám o sobě velmi neúčinný a neodpovídá ani běžné realitě našeho života. Je proto potřeba definovat nástroje pro šíření informací v systému. Těmto nástrojům se budeme věnovat v kapitole 6. Nyní se podívejme na základní členění navrženého systému a popis jednotlivých částí.

## 5.2 Statická část

Celý systém lze z hlediska dynamiky rozdělit na dvě základní části - část statickou a část dynamickou. Statickou část systému tvoří všechny prvky, které nedokáží sami měnit svůj stav, ale mění ho pouze interakcí s jinými prvky systému. Tyto prvky nemají vlastní inteligenci a vytváří prostředí a podmínky pro existenci dynamických prvků. Návrh statické části systému se opírá o některé základní informace využívané v geografických informačních systémech.

### 5.2.1 Krajina

Pojem krajina zde není myšlen tak, jak by jej chápali ekologové. V pojetí tohoto systému je krajina chápána pouze jako model zemského povrchu bez nerostného bohatství, rostlin, zvířete nebo důsledků existence postav v systému. Existuje několik druhů povrchů, z nichž každý je specifický svojí pohybovou náročností, která ovlivňuje volbu cest postav po světě. Snadnější je pohyb po travnatých pláních než po horách nebo bažinách. Pohyb po vodě není možný vůbec.

Některé z povrchů jsou pak vhodné pro pěstování některých plodin, na jiných plodiny pěstovat nelze.

### 5.2.2 Nerostné bohatství

Nerostné bohatství je základem celého ekonomického systému hry. Na produkty těžby nerostného bohatství navazuje celý výrobní řetězec. Nerostné bohatství tvoří zdroje kamene, železné rudy, uhlí a zlata. Dále pak vodní zdroje a lesy. Dobývání jednotlivých druhů nerostného bohatství zajišťují postavy vykonávající příslušnou profesi (například kameník, dřevorubec a další).

Jedním z problémů návrhu tohoto systému je obnovitelnost zdrojů nerostného bohatství. V reálném světě se z výše uvedených zdrojů považují za obnovitelné pouze lesy. Voda je pak, nemluvíme-li konkrétně o vodě pitné, považována vzhledem ke koloběhu vody za zdroj teoreticky nevyčerpatelný. Ostatní zdroje nerostného bohatství jsou považovány za neobnovitelné, ale lidé se je snaží nahradit recyklací nebo substituty na vyšší technologické úrovni.

V návrhu systému jsou lesy považovány za obnovitelný zdroj a vodní zdroje za nevyčerpatelné. Ostatní zdroje jsou považovány za neobnovitelné, což vzhledem k tomu, že systém neobsahuje rozvoj technologií ani možnosti recyklace, může vést k jejich vyčerpání bez možnosti nahrazení.

### 5.2.3 Budovy

Budovy jsou jedním ze základních předpokladů existence postav v systému. Za jejich stavbu a volbu jejich umístění je zodpovědný hráč. Budov je několik druhů a každá má specifické požadavky na množství použitého stavebního materiálu a specifické parametry jako je odolnost proti útoku, časová náročnost stavby atd. Budovy lze podle jejich funkčnosti rozdělit na dílny a obytné budovy.

- Dílny jsou budovy, v nichž probíhá výroba. V dílnách je skladován materiál a nářadí potřebné k výrobě, stejně jako výstupy výroby do doby, než jsou prodány. Dílna je tedy také místem střetu nabídky s poptávkou. Pro zjednodušení systému je dílna zároveň také domovem postavy, která zde vykonává své řemeslo. Tato postava je primárně odpovědná za obranu budovy.
- Obytné budovy slouží jako domov postav, které neprovozují řemeslnou činnost. Zde postavy shromažďují a uchovávají svůj soukromý majetek a zároveň jim jejich domov slouží jako místo k odpočinku. Každá postava je primárně odpovědná za obranu svého domova.

V případě zničení budovy dojde ke zničení některých předmětů, které se v budově nacházely, případně k úmrtí některých osob, které v době zřícení byly v budově. Stejně jako v reálném světě však nemusí dojít ke ztrátě veškerého majetku ani k úmrtí všech postav.

### 5.2.4 Zboží

Za zboží jsou považovány všechny prvky systému, se kterými je možné obchodovat. Patří sem například dřevěné násady, maso, pivo a další potraviny, produkty těžby nerostného bohatství, artefakty a další. Některé druhy zboží jsou pouze polotovary, které vstupují do další výroby, jiné slouží jako pracovní nástroje pro uskutečnění produkce a jsou během produkčního procesu

opotřebovávány. Mezi zbožím existuje ve výrobním řetězci mnoho návazností. Některé z těchto návazností ukazuje tabulka v příloze A. V tabulce jsou zachyceny pouze kvalitativní, nikoli kvantitativní, návaznosti. Editací této tabulky ve zdrojovém kódu lze měnit kvantitativní i kvalitativní závislosti mezi jednotlivými produkty a následně i mezi řemesly (viz odstavec 5.3.2). Výjimkou jsou položky označené v závorce poznámkou prim. Jedná se o zdroje nerostného bohatství, se kterými je v systému zacházeno rozdílně. Tyto řádky tabulky nelze měnit bez dalších úprav kódu, aniž by nedošlo k narušení celého systému.

Speciálním druhem zboží jsou artefakty. Artefakty svému nositeli nějakým způsobem zlepšují schopnosti a dovednosti. Svitky umožňují čarovat kouzla, meče mohou kromě zvýšení škod, které postava působí v boji, také zlepšit obratnost atd. Některé z artefaktů dokáží použít jen postavy určité úrovně a schopností, některé toto omezení nemají. Základní artefakty, jako jednoduché meče nebo štíty, vyrábějí řemeslníci. Ostatní se dají náhodně najít na mapě, nebo je u sebe nosí některé z nepřátelských postav.

Všechny artefakty jsou obchodovatelné na trhu. Tím se zvyšuje složitost celého systému. Například zloděj nyní může zvažovat zabití nebezpečné nepřátelské postavy, která má u sebe vzácnou kouzelnickou berlu jen kvůli vidině jejího prodeje, zatímco normálně by se o ni vůbec nezajímal, protože ji nedokáže použít.

## 5.3 Dynamická část

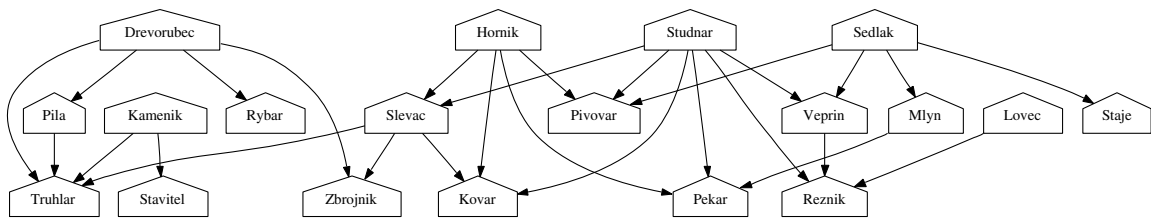
Dynamickou část systému tvoří všechny prvky systému, které dokáží změnit svůj stav i bez interakce s jinými prvky systému. Všechny dynamické prvky mají vlastní inteligenci. Dynamické prvky systému využívají prvků statických a mohou svým chováním ovlivňovat všechny ostatní prvky systému. Dynamické prvky systému lze rozdělit do těchto čtyř skupin:

- *zvěř* - prvek s nejnižší mírou inteligence schopný jednoduchých reakcí na základní podněty,
- *řemeslníci* - starají se především o plánování a realizaci výroby a prodeje,
- *hrdinové* - využívají produktů řemeslné výroby a jsou základním nástrojem obrany království,
- *nepřátelské herní entity* - snaží se o destabilizaci království, případně o jeho celkové zhroucení.

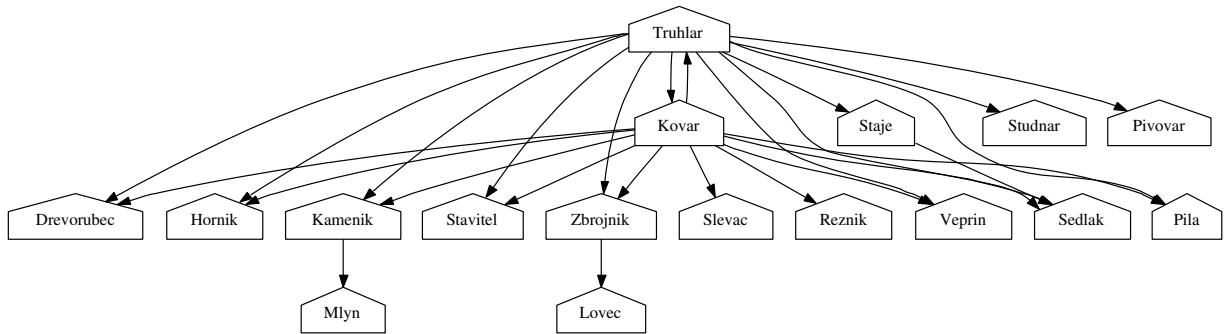
### 5.3.1 Zvěř

Zvěř je dynamický prvek systému s nejnižším stupněm inteligence. Inteligence je implementována pouze u zvěře divoké, u domestikovaných zvířat v systému, jako jsou koně ve stájích nebo prasata ve chlívě, se žádná inteligence nevyskytuje.

Inteligence zvěře spočívá ve schopnosti reagovat na podněty z okolí. Při vizuálním kontaktu zvířete s jakoukoliv postavou existuje možnost, že se zvíře vyplaší a dá se na útěk. Stejně tak reaguje i při napadení. Volba trasy úniku je ovlivněna nejmenší průměrnou vzdáleností zvířete od všech postav, které zvíře vidí.



Obrázek 5.1: Závislosti mezi řemesly podle dodávaných vstupů



Obrázek 5.2: Závislosti mezi řemesly podle dodávaných nástrojů

### 5.3.2 Řemeslníci

Řemeslníci jsou prvním prvkem systému, od něž se očekává inteligentní chování. Výrobní závislosti mezi řemesly jsou znázorněny v diagramech na obrázcích 5.2 a 5.3. V obou diagramech nejsou zakresleny vztahy samozásobování a nejsou zachyceny profese, které nemají z daného hlediska žádný vztah s jinou profesí.

Řemesla lze rozdělit do dvou základních skupin. Jedna skupina řemeslníků vykonává svoji profesi v dílně a opustit pracoviště je nucena, pouze když potřebuje nakoupit některé suroviny nebo nástroje. Druhá skupina řemeslníků pracuje v terénu. Do této skupiny patří například lovec, který musí vyhledávat v krajině zvěř, pronásledovat ji a zabít. Podíváme-li se na skupinu řemeslníků jako celek, projevuje se jejich inteligence v následujících oblastech:

- *analýza trhu* - umožňuje správné stanovení požadované ceny za nabízené produkty a stanovení množství produkce,
- *volba dodavatele* - se snahou o minimalizaci nákladů na pořízení materiálu a dalších vstupů,
- *volba zdrojů nerostného bohatství* - opět s cílem minimalizovat náklady,
- *ochrana majetku* - s cílem minimalizovat škody na majetku a předcházet výskytu těchto škod,
- *nakládání s volným časem* - se snahou maximalizovat užitek volbou správné činnosti v daném okamžiku.

Detailní popis všech těchto projevů je příliš komplikovaný a obsáhlý, proto se zde zaměříme pouze na základní charakteristiku. K některým podrobnostem se blíže dostaneme v kapitole 6, ostatní lze nalézt v kódu příloženého programu.

### 5.3.2.1 Analýza trhu

Analýza trhu spočívá ve volbě vhodného sortimentu produktů, které se řemeslník rozhodne v daném období vyrábět. Volba produktu je determinována ziskem, který může řemeslník na produktu realizovat. Tento zisk je roven součinu ceny produktu a množství, které se podařilo za tuto cenu prodat, snížený o náklady na produkci.

Zde se dostáváme k problematice ekonomických modelů, které jsou předmětem neustálého zájmu odborníků. Existuje řada projektů zabývajících se modelováním trhu, ať už pro účely predikce nebo pro účely objasnění některých tržních jevů. Jednotlivé modely se liší mírou abstrakce a oblastmi zájmu. Jako příklad detailně rozpracovaného ekonomického modelu uveďme projekt EMCAS (The Electricity Market Complex Adaptive System) [8]. Obecně jsou tyto modely velmi rozsáhlé a jejich implementace není triviální.

Pro účely tohoto projektu byl navržen relativně jednoduchý model trhu, který zohledňuje pouze základní faktory ovlivňující nabídku a poptávku. Detailnějšímu popisu tohoto modelu se budeme věnovat v kapitole 6. Zde se zaměříme pouze na základní charakteristiku tohoto modelu.

Pokud bychom uvažovali navrhovaný trh jako dokonale konkurenční, pak by řemeslníci museli přijmout cenu zboží určenou trhem a jejich jedinou možností na maximalizaci zisku by byla správná volba množství produktu, jež se rozhodnou na trh vystavit. Toto množství je determinováno poptávkou po produktu při dané ceně.

Pokud budeme považovat navrhovaný trh za nedokonale konkurenční, pak může řemeslník stimulovat poptávku vhodným nastavením ceny zboží. Cenu produktu musí řemeslník nastavit tak (pokud neuvažujeme možnost dumpingových cen na zničení konkurence), aby pokryl své náklady na nákup materiálu, pokryl náklady spojené s opotřebením nářadí, zaplatil daně do státní pokladny a navíc realizoval dostatečný zisk, který mu zůstane na nevýrobní výdaje, jako je strava nebo finanční rezerva. Důležitým faktorem je zde cena u konkurence.

### Na jakém trhu se pohybujeme?

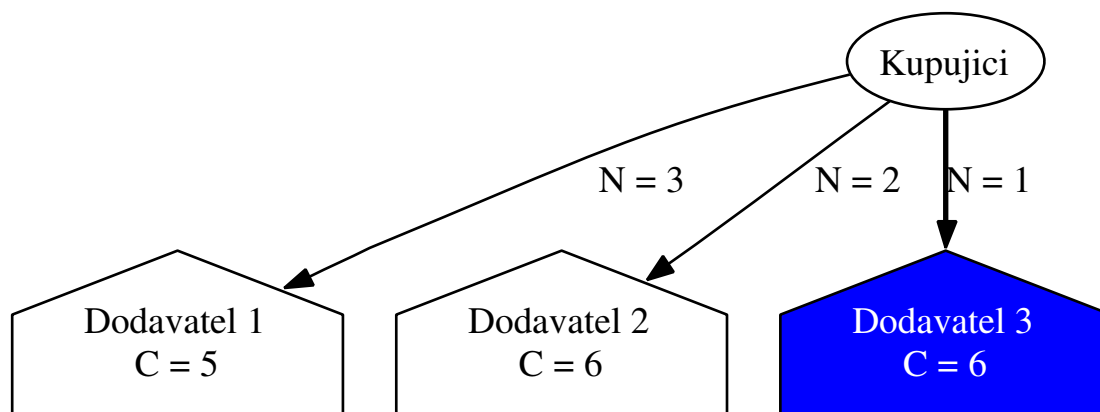
V systému nelze odlišit výrobky stejného druhu od dvou různých řemeslníků a neexistují ani žádné bariéry vstupu na trh. To jsou dva ze základních předpokladů dokonale konkurenčního trhu.

V systému však existuje jednoduchá forma know-how. Řemeslník déle pracující v daném odvětví má více zkušeností a dokáže tak vyrábět s menším procentem zmetků a tím pádem i spotřebou surovin, než začínající řemeslník. Rovněž dostupnost informací pro všechny řemeslníky v odvětví není stejná. Například řemeslník *A* ví, za kolik prodával řemeslník *B*, ale neví kolik výrobků se mu na trhu podařilo za tuto cenu prodat. To naznačuje, že by trh měl být modelován spíše jako nedokonale konkurenční.

### 5.3.2.2 Volba dodavatele

Jak bylo uvedeno výše, nelze vzájemně rozlišit dva stejné druhy zboží od dvou různých dodavatelů. Problém volby dodavatele je z tohoto pohledu pouze hledáním nabídky s nejnižší cenou. Je však třeba uvážit rozpočet, který má řemeslník na nákup materiálu. Tím nastává problém, jakým způsobem rozdělit peníze mezi nákup jednotlivých druhů materiálu tak, aby byl řemeslník schopen z tohoto materiálu vyrobit zboží, jehož prodejem vykáže maximální zisk.

Kromě prodejní ceny vstupují do rozhodovacího procesu také náklady na cestu ke každému z dodavatelů, jak ukazuje diagram na obrázku 5.4. Zde je nejnižší nákupní cena daného množství



Obrázek 5.3: Volba dodavatele

u Dodavatele 1. Celkové náklady na nákup u tohoto dodavatele jsou však vyšší v porovnání s celkovými náklady na nákup u Dodavatele 3. Dodavatel 3 tak těží z lepšího postavení své prodejny.

### 5.3.2.3 Volba zdrojů nerostného bohatství

Volba zdrojů nerostného bohatství je ovlivněna především náročností cesty, kterou musí postava urazit, než se ke zdroji dostane. Dalším faktorem může být nebezpečí vyskytující se v dané oblasti nebo množství suroviny, kterou lze ještě ze zdroje získat.

### 5.3.2.4 Ochrana majetku

Řemeslníci, stejně jako všechny ostatní postavy v systému, jsou zodpovědné za ochranu svého majetku. Chránit majetek proti napadení může majitel buď sám, nebo může požádat o pomoc další postavu. O podmínkách pomoci postavy vyjednávají.

Vyjednávání je ze strany žádajícího ovlivněno mírou rizika, jež odpovídá pokusu o obranu vlastními silami a hodnotou majetku, jež je v ohrožení. Z hlediska žádaného je vyjednávání ovlivněno mírou rizika, které podstoupí při přijetí dohody, charakterem postavy, náklady spojenými s úslou příležitostí a ekonomickou situací postavy. Není ovšem vyloučena možnost, že některá z postav dobrovolně pomůže a bude danou budovu bránit bez toho, aby o to byla požádána.

### 5.3.2.5 Nakládání s volným časem

Každý z řemeslníků rozhoduje svobodně a racionálně o svém čase. Pokud tedy zjistí, že se mu v daném období při jeho výrobních možnostech a očekávané poptávce nevyplatí pracovat, rozhoduje se dále o tom co bude dělat. Jednou z možností je vydat se na průzkum neznámých oblastí s cílem získat nové informace o systému a najít další zdroje. Další možností je zůstat doma a minimalizovat tak své náklady a riziko.

Poslední možností je změnit povolání a přesunout se tak na trh s jiným produktem. Tato možnost je v systému podporována pomocí přenosu zkušeností mezi podobnými profesemi. Tím je přechod mezi některými profesemi méně nákladný než přechod mezi jinými. Přestože je změna profese v navrženém systému podporována, není ještě navržen příslušný rozhodovací mechanismus.



### 5.3.3 Hrdinové

Hrdinové jsou druhým prvkem systému, od něhož je očekávána vysoká míra inteligence. Zatímco řemeslníci se specializovali především na výrobu a obchod, hrdinové jsou specializováni na boj a zajišťování svého živobytí se zbraní v ruce.

Inteligence hrdinů se projevuje v následujících činnostech:

- *výběr úkolů* - zajišťující maximalizaci zisku,
- *kooperace a vyjednávání* - s cílem zvýšit pravděpodobnost úspěšného splnění úkolu,
- *boj* - s cílem minimalizovat riziko zranění volbou vhodných strategií boje,
- *obchodování* - se snahou minimalizovat náklady na nákup vybavení a výhodně prodat nepotřebné předměty.

#### 5.3.3.1 Výběr úkolů

Výběr úkolů je nejdůležitější činností, kterou hrdina provádí. Špatná volba úkolu může stát hrdinu mnoho zbytečného úsilí nebo dokonce život. Volbu úkolu ovlivňují dva základní faktory.

1. Velikost odměny je prvním z těchto faktorů. Odměna za splnění úkolu je jednak finanční částka vypsaná na splnění úkolu. Dále je to množství zkušenostních bodů, které hrdina získá, případně i kvalita obdržených artefaktů.
2. Druhým faktorem je míra rizika. Ta v sobě zahrnuje hned několik druhů rizika. Jedním z rizik je riziko, že úkol bude nad hrdinovi síly a pokusem o jeho splnění ztratí čas, energii, peníze ušlé příležitosti, nebo i vlastní život. Dalším z rizik je riziko, že úkol splní dříve někdo jiný. Toto riziko stoupá se vzdáleností hrdiny od místa plnění úkolu, ale i se stářím informací hrdiny, které s úkolem souvisejí.

#### 5.3.3.2 Kooperace a vyjednávání

Kooperace a vyjednávání úzce souvisí s volbou úkolů. Představme si dva hrdiny, kteří se rozmýšlí mezi dvěma úkoly. Odměna za první úkol je přitom třikrát vyšší než odměna za druhý. Jakkoliv je však první úkol lákavý, je míra rizika při plnění tohoto úkolu příliš vysoká a žádný z hrdinů by si na něj samostatně netroufl. Hrdinům se však vyplatí pokusit se dohodnout na společném plnění úkolu. Tím se míra rizika podstatně sníží a může dosáhnout přijatelné míry pro oba hrdiny. Dohodnou-li se přitom na rovném dělení odměny, bude odměna pro každého z nich o polovinu vyšší, než při plnění druhého úkolu. Nehledě na to, že pokud by se oba pokusili o plnění druhého úkolu, pak by jeden z nich uspěl zatímco druhý ne.

Kooperace a vyjednávání může vést rovněž k uzavírání koalic mezi postavami. Tato problematika následně zahrnuje rozhodování postav o výhodnosti setrvání v uzavřené koalici, případně o možnosti odstoupení od koalice, nebo uzavření koalice jiné.

### 5.3.3.3 Boj

Při plnění úkolů se hrdina často musí střetnout s nepřítelem v boji. Každý hrdina ovládá jiné techniky boje a podle nich musí volit svoji bojovou strategii. Například kouzelník nebo lučištník se bude snažit držet si odstup od nepřítele, zatímco rytíř nebo barbar se musí dostat do bezprostřední blízkosti nepřítele. Kromě těchto velmi elementárních taktik má každý hrdina k dispozici několik útočných a obraných pohybů, případně kouzel, které může v boji použít.

Jak již bylo uvedeno dříve, systém je založen na nedokonalé a neúplné informaci. Do problematiky boje tak patří i problematika modelování protivníka. Postava se v každém boji setkává s protivníkem jiných kvalit a schopností a teprve postupem času získává informace o jeho strategiích a slabinách. Tyto informace pak vedou ke změnám vlastní strategie. Postupy používané pro modelování protivníka jsou popsány v kapitole 3. Jejich aplikace na navrhovaný herní systém bude přiblížena v kapitole 6.

Dalším zajímavým problémem je boj skupiny kooperujících hrdinů proti skupině kooperujících nepřátel. Zde je vhodné rozdělení nepřátel mezi bojovníky a účinná podpora hrdinů mezi sebou velmi důležitá. Z hlediska návrhu a implementace se jedná o nejtěžší prvek celého systému.

### 5.3.3.4 Obchodování

Obchodování je z pohledu hrdinů spíše okrajovou činností, nikoliv však nevýznamnou. Vhodná výbava a volba správných zbraní, které nejlépe vyhovují charakteru postavy, patří k důležitým rozhodnutím, která mohou zásadně ovlivnit život hrdiny. Obchodování však není jen nákup vybavení, ale i prodej získaných předmětů, které hrdina považuje za nepotřebné. Zde se musí hrdina zabývat správným oceněním předmětu a nabídnout předmět k prodeji na správném místě a ve správnou dobu, aby maximalizoval svůj příjem z tohoto prodeje.

### 5.3.4 Nepřátelské herní entity

Nepřátelské herní entity jsou posledním prvkem celého systému. Svým charakterem a chováním jsou v mnohém podobné hrdinům. Jejich úkolem je však škodit království. Hledání slabých míst a osamocených obyvatel království, které by mohli napadnout, patří k základní náplni jejich činnosti. Stejně jako hrdinové mohou používat nejrůznější artefakty a kouzla a uzavírat koalice. Jejich inteligence se projevuje v méně oblastech než u hrdinů. Vyjednávání o spolupráci i rozhodování o činnostech je jednodušší, neboť nepřátelské herní entity v systému nemohou obchodovat a nejsou proto motivovány penězi. Jedinou motivací je pro ně velikost škody, kterou mohou napáchat.

## Kapitola 6

# Popis řešení vybraných částí navrženého systému

*V této kapitole se seznámíme s metodami a technikami, které byly použity pro řešení některých klíčových částí navrženého herního systému. Budeme diskutovat důvody použití těchto technik a jejich případné nedostatky. Jelikož detailnější technický rozbor použitých metod by, podle mého mínění, nepřispěl k lepšímu porozumění problematice, budou zde prezentovány pouze klíčové myšlenky a postupy. Všechny úvahy diskutované v této kapitole jsou implementovány v přiloženém programovém kódu.*

Implementovaný herní systém je navržen jako procesně řízená diskrétní simulace a je postaven na nedokonalé a neúplné informaci tak, jak ji chápe matematická teorie her (viz 2.3 a 2.4), a jak byla dále popsána v návrhu systému (viz 5.1). Každý dynamický prvek systému je modelován jako samostatný proces s vlastním inteligentním chováním. V této kapitole se podíváme na návrh řešení několika nejzajímavějších částí tohoto projektu. Budeme se zde zabývat modelováním znalostí a jejich prostorovou a časovou distribucí mezi herními entitami. Dále se budeme zabývat modelováním trhu a navrhne způsob řešení dosud neimplementované části programu, kterou jsou souboje postav (ve smyslu terminologie této hry).

V této kapitole se nebudeme zabývat implementací statické části systému a pomineme i některé algoritmy používané dynamickými prvky, jako je například Dijkstra algoritmus pro hledání optimálních cest po herním světě.

### 6.1 Procesně řízená simulace

Jelikož je celý systém navržen jako procesně řízená diskrétní simulace, je nutné zajistit paralelní běh všech procesů v systému. Jako nástroj pro zajištění tohoto požadavku byla použita knihovna SIMLIB [13].

Použití knihovny SIMLIB s sebou nese kromě zřejmých výhod také některé nevýhody. Prvním z problémů byl překlad této knihovny na 64-bitové distribuci Ubuntu. Jako reakce na tyto problémy byl celý navrhovaný systém přenesen na 32-bitovou distribuci Knoppixu, kde již tyto problémy nenastaly. Druhým a závažnějším problémem je nutnost překladu kódu s optimalizací, aby byla zajištěná správná funkčnost knihovny SIMLIB. Tato optimalizace znesnadňuje ladění celého systému a podstatně tak zvyšuje časové nároky na vývoj. Posledním již méně závažným

nedostatkem je možnost běhu simulace pouze v omezeném časovém rozsahu a značná režie přepínání mezi procesy.

Jelikož je z poměrně rozsáhlé knihovny SIMLIB použit pouze plánovač procesů, nabízí se jako další postup projektu implementace vlastního plánovače procesů s cílem odstranit uvedené problémy a pokud možno zrychlit běh celé simulace.

## 6.2 Modelování znalostí

Jak již bylo zmíněno v předchozím textu, je systém postaven na nedokonalé a neúplné informaci. Každý z agentů tedy samostatně shromažďuje znalosti o systému, které později využívá při svém rozhodování. Tento model vyžaduje implementaci báze znalostí pro každého agenta v systému samostatně. Podívejme se tedy na model báze znalostí a na mechanismus časové a prostorové distribuce znalostí v navrhovaném systému.

Dodejme, že reprezentací znalostí a jejich analýzou se zabývá jedna část oboru Umělé inteligence, která se zabývá expertními systémy. V tomto projektu je problematika znalostí pouze naznačena.

### 6.2.1 Modelování báze znalostí

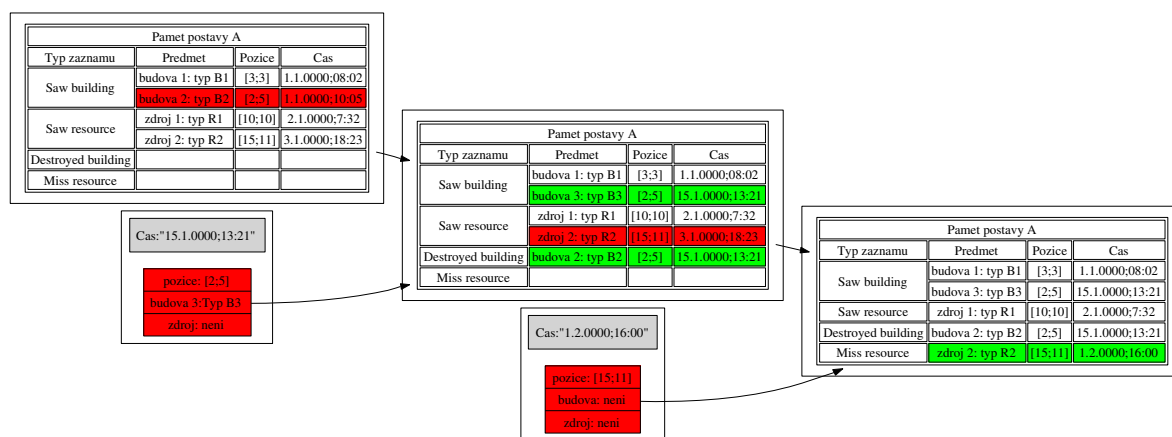
Báze znalostí postavy (dále jen paměť) je založena na vnímání a uchovávání důležitých událostí, jichž byla postava svědkem. Charakteristickými rysy každé události jsou:

- *čas*, kdy k události došlo (vlastní modelový čas navrženého systému),
- *místo*, kde k události došlo (souřadnice v rámci herní rastrové mapy),
- *předmět události* (vlastní sdělení).

Z hlediska *času* je paměť rozdělena na dvě části. Paměť krátkodobou a dlouhodobou. V krátkodobé paměti jsou ukládány informace o právě sledovaných jevech (postavy, zvířata, předměty a budovy, které postava právě vidí). V okamžiku, kdy je běh těchto událostí ukončen, nebo je místo výskytu jevu již mimo dosah zorného pole postavy, dochází k přesunu informace z krátkodobé paměti do paměti dlouhodobé. V dlouhodobé paměti jsou všechny události opatřeny časovými známkami, které umožňují chronologické seřazení jednotlivých událostí. V systému není řešena možnost rozdílného vnímání času postavami. Není ani implementováno zapomínání.

*Místo* a *předmět události* umožňují vzájemné porovnání dvou událostí. Některé předměty událostí na stejném místě se vzájemně vylučují. Porovnáním časových známek je pak možno určit, která z událostí je novější a přepíše událost starší. Tím je zajištěn jednoduchý model inteligence postav založený na dedukci. Uveďme jednoduchý příklad, který je znázorněn na obrázku 6.1. V tomto obrázku jsou červeně značeny kolidující informace a zeleně jsou značeny úspěšně katalogizované informace.

Postava prochází místem *A* na mapě a vidí zde stát "budovu 1" typu *B2*. Uloží si tento záznam do své paměti a od místa *A* se vzdálí. Po nějakém čase se opět dostane na místo *A*, ale nyní zde vidí "budovu 3" typu *B3*. Vzhledem k tomu, že se jedná o dvě soumísné události, které se navzájem vylučují, je informace o nalezení budovy "budovy 1" na místě *A* nahrazena



Obrázek 6.1: Vývoj báze znalostí postavy podle pozorovaných událostí

informací o zničení "budovy 1" na místě *A* a je doplněna informace o nalezení "budovy 3" typu *B3*.

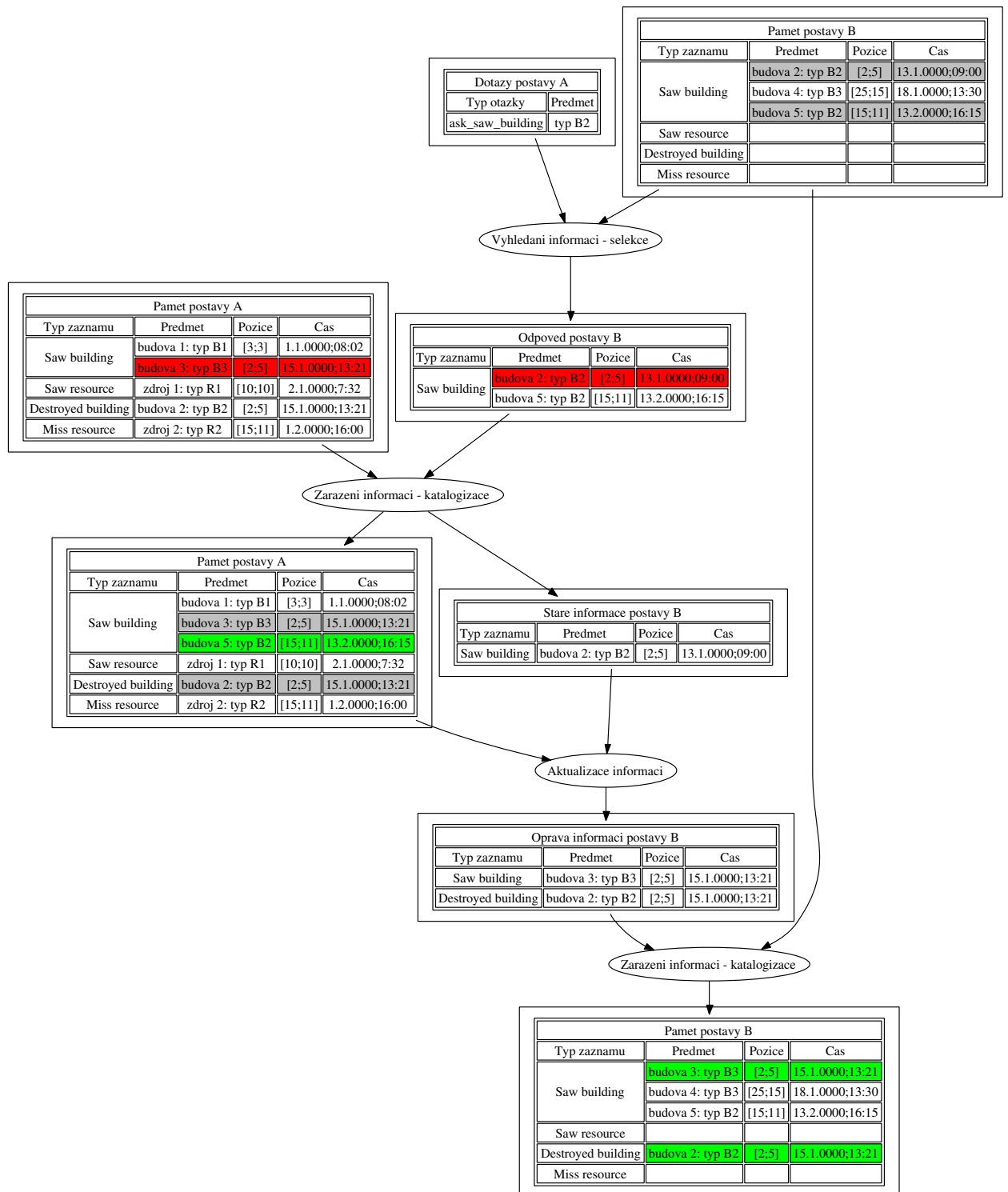
### 6.2.2 Časová a prostorová distribuce znalostí mezi herními entitami

Každá z postav samostatně sbírá znalosti (informace) o herním světě. Těchto informací je však příliš mnoho a neustále vznikají nové na různých místech herního světa. Samotná postava tak není schopna udržovat si reálnou představu o současném stavu systému. Proto je třeba umožnit šíření informací mezi postavami. Mechanismem pro šíření informací je vzájemná komunikace postav. Tato komunikace je zajištěna pomocí dotazování.

Postava se během své činnosti může ocitnout v situaci, kdy postrádá klíčovou informaci. Mějme například výrobce, který se rozhodl vyrábět produkt *P*. Bohužel však nemá jednu potřebnou surovinu *S*, která je produktem dílen typu *B2*. Výrobce ovšem neví o existenci žádné dílny tohoto typu. Tím je pochopitelně výroba produktu *P* znemožněna. Pokud by nebylo umožněno šíření informací, musel by se výrobce dříve či později vydat na průzkum herního světa s cílem najít dílnu typu *B2*. Je-li však umožněno šíření informací, je velká pravděpodobnost, že tato nutnost nenastane. Jakmile výrobce zjistí, že postrádá informaci o umístění dílny typu *B2*, vytvoří si dotaz na tuto informaci a začne zvažovat výrobu dalšího produktu, řekněme *Q*. Předpokládejme, že výrobce ví, kde má nakoupit všechny potřebné suroviny pro výrobu *Q* a vydá se na nákupy. Po cestě se ptá všech postav, které potká, zda nevědí o nějaké dílně typu *B2*. Pokud se podaří tuto informaci získat, je začleněna mezi vědomosti postavy a v dalším zvažování výroby produktu *P* bude využita. Proces komunikace dvou postav je znázorněn na obrázku 6.2. Šedě jsou značeny znalosti vybrané při selekci podle daného dotazu. Červeně jsou značeny znalosti, které spolu vzájemně kolidují a zeleně jsou značeny znalosti, které se povedlo úspěšně katalogizovat.

Distribuce informací v systému je implementována relativně jednoduchým způsobem. Neexistuje znepřesňování informací vzhledem k jejich stáří, neexistuje ani možnost podvržení mylné informace nebo odmítnutí informaci poskytnout. Pouze v případech, kdy je předmětem dotazu zdroj produktu, jehož výrobcem je dotazovaná postava, tato postava jako jediný možný zdroj dotazovaného produktu uvede svou dílnu, i když může mít informace i o umístění konkurence.

Při získávání informací od jiných postav si postava rovněž nemůže být jista aktuálností těchto



Obrázek 6.2: Předávání informací mezi postavami při komunikaci

informací, neboť stav systému se mezi okamžikem vytvoření a předání informací mohl změnit.

Navržený model znalostí a jejich distribuce lépe vystihuje skutečnou realitu a vytváří zajímavější prostředí, než je tomu u systémů se sdílenými znalostmi. V rámci případných dalších rozšíření navrženého systému je možné tento model využít ke sledování vlivu reklam nebo mylných informací na chování herních entit a vývoj systému jako celku.

## 6.3 Model trhu

Modelování trhu je v současnosti stále problémem, jemuž se věnuje celá řada odborníků (Gibbons, Aumann a další) ve snaze najít model, který by co nejlépe vystihoval mechanismy reálného tržního prostředí. Pohledů na tuto problematiku existuje celá řada. Z mikroekonomického hlediska do této problematiky spadá například teorie spotřebitele a teorie firmy. Setkáváme se s pojmy jako je elasticita nabídky a elasticita poptávky, pojmy komplement a substitut, bod uzavření firmy a tak dále. Makroekonomické hledisko se pak věnuje vlivům investic, státních zakázek a exportu na celkovou poptávku, stejně jako substitučnímu a důchodovému efektu na trhu s prací. Nebudeme zde rozebírat všechny tyto pojmy, ty lze nalézt v každé učebnici mikroekonomie nebo makroekonomie, ale podíváme se na navrhovaný přístup k řešení problému a naznačíme jeho výhody a nedostatky. Nejprve se však seznámíme se základními pojmy.

Poptávka je pojem vyjadřující objem zboží či služeb, které si kupující chce koupit za danou cenu. Lze rozlišit poptávku pružnou, která rychle reaguje na změnu ceny a poptávku nepružnou. Nepružná poptávka se většinou týká nepostradatelného a nenahraditelného zboží.

- Dílčí poptávka vyjadřuje objem konkrétního zboží či služby, které si kupující chce koupit za danou cenu.
- Nabídka je pojem vyjadřující objem zboží či služeb, jež výrobci chtějí prodat za danou cenu.
- Dílčí nabídka vyjadřuje objem konkrétního zboží či služby, jež výrobci chtějí prodat za danou cenu.
- Trh je místem, kde se setkává nabídka s poptávkou.
- Dílčí trh je místem, kde se setkává dílčí nabídka s odpovídající dílčí poptávkou.

V navrhovaném systému existují dvě skupiny informací o trhu. Jedna skupina jsou agregátní informace o trhu jako celku. Těmito informacemi jsou celková poptávka za minulé období, celková nabídka, množství uskutečněného prodeje a průměrná cena, za niž byl tento prodej uskutečněn. Všechny tyto informace jsou uchovávány pro každý dílčí trh zvlášť.

Druhou skupinou informací jsou vnitropodnikové informace, které si uchovává každý z výrobců samostatně a nejsou veřejné. Těmito informacemi jsou nabídka realizovaná výrobcem za minulé období, množství uskutečněného prodeje, cena za niž výrobce prodával, velikost fixních a variabilních nákladů, výrobní kapacita a objem zásob na skladě, přičemž variabilní náklady jsou náklady spojené s výrobou každé jednotky produkce a fixní (režijní) náklady jsou náklady, které producent musí zaplatit, i když nerealizuje žádnou výrobu. Fixní náklady tedy nutí producenta k výrobě produktů a jejich prodeji, aby minimalizoval svoji ztrátu.

Jelikož jsou v okamžiku plánování výroby na další období k dispozici pouze údaje z období předchozího a neexistuje možnost uzavírání smluvních cen mezi dodavatelem a odběratelem (problematika modelování dlouhodobých kontraktů), předpokládají jednotliví producenti vyšší svých variabilních nákladů a objem poptávky pro analyzované období shodnou s obdobím uplynulým. Za tohoto předpokladu se nyní podíváme na jednoduchý příklad analýzy trhu se dvěma konkurenty  $A$  a  $B$ . Nejprve však musíme stanovit, co bude strategií každého z těchto hráčů. Existují tři základní přístupy.

*První* možností je považovat trh za *dokonale konkurenční*. Na tomto trhu jsou všichni hráči nuceni *akceptovat tržní cenu* a jejich *strategií je objem produktu*, který se za tuto cenu rozhodnout vystavit na trh. Problémem tohoto přístupu je stanovení tržní ceny, neboť cena je, kromě kvality a reklamy, jedním ze základních faktorů stimulujících poptávku. Při nižší ceně poptávka roste, při vyšší ceně klesá (toto pravidlo neplatí pro luxusní a nenahraditelné potřebné zboží).

*Druhou* možností je považovat za *strategii hráče volbu ceny*, za kterou se rozhodne zboží na trh vystavit. Tento přístup je dobře aplikovatelný na výroby, kde je nákladné zastavení provozu a jeho opětovné spuštění. Tito producenti nejsou schopni efektivně měnit množství nabízeného produktu a musejí využívat pouze cenových strategií k vhodné stimulaci poptávky.

*Posledním přístupem* je považovat za strategii výrobce jak cenu, tak i množství vyráběného produktu. Tento přístup se jeví jako dostatečně obecný a umožňuje výrobcovi efektivně reagovat na všechny tržní vlivy, jako je růst nákladů, změna poptávky atd. Tento přístup je použit v navrženém systému.

Kromě těchto tří základních přístupů existuje ještě kombinovaný přístup. Uvažujme oligopolní situaci na trhu, kdy existuje jeden silný výrobce v odvětví a pak skupina slabších výrobců. Silný výrobce si může dovolit stanovovat kromě produkovaného množství, také cenu produktu. Tento výrobce bývá též označován jako cenový vůdce. Ostatní výrobci v tomto odvětví jsou nuceni akceptovat cenu, kterou nastaví cenový vůdce. Všichni tyto výrobci se tak pohybují na dokonale konkurenčním trhu. Ve skutečnosti jdou ostatní výrobci zpravidla s cenou o něco níže, aby prodali celý objem své produkce.

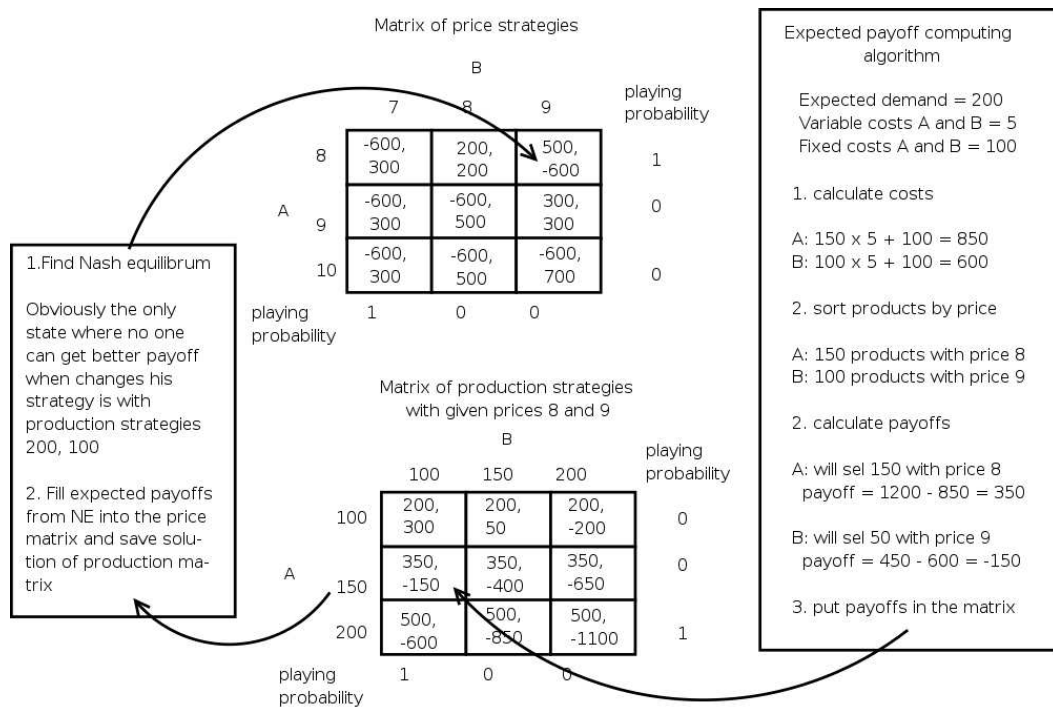
### 6.3.1 Příklad

Vraťme se nyní k našemu příkladu. Uvažujme pro zjednodušení, že oba výrobci mají stejné fixní i variabilní náklady, žádné zásoby a stejnou výrobní kapacitu. Výrobce  $A$  uvažuje o prodeji v cenovém rozmezí 8 až 10 jednotek za kus, výrobce  $B$  v rozmezí 7 až 9 jednotek za kus. Oba dva uvažují o výrobě 100 až 200 kusů produktu. Postup výpočtu optimálních strategií pro oba hráče je znázorněn na obrázku 6.3.

Uvažujme kombinaci cen  $P_A = 8$ ,  $P_B = 9$ . Při těchto cenových strategiích hráči sestaví matici možných kombinací nabízeného množství produktu. Tuto matici naplní očekávanými zisky za prodej výrobků při předpokládané poptávce. Vezměme množstevní kombinace  $Q_A = 150$ ,  $Q_B = 100$ . Jednotkové náklady na produkci jsou dány vztahem  $C = V + \frac{F}{Q}$ , kde  $V$  jsou variabilní náklady na produkci jednoho kusu výrobku,  $F$  jsou fixní náklady a  $Q$  je nabízené množství. Fixní náklady se tak rovnoměrně rozdělí mezi všechny výrobky. Jednotkové náklady jednotlivých výrobců při daných uvažovaných objemech produkce jsou tedy  $C_A = 5,66$  a  $C_B = 6$ . Jednotkové zisky jsou pak  $Z_A = 2,34$  a  $Z_B = 3$ .

Předpokládejme dále, že oba producenti predikují poptávku po daném produktu podle hodnoty z minulého období ve výši  $D = 200$ . Vzhledem k tomu, že zboží od jednotlivých výrobců





Obrázek 6.3: Algoritmus pro stanovení optimální ceny a množství produkce

je diferencováno pouze cenou, zvolí racionální spotřebitelé nejprve nákup levnějšího zboží a až poté se uchýlí ke zboží dražšímu. Producent  $A$  tak uskuteční prodej  $S_A = 150$  a jeho celkový zisk bude přibližně 350 jednotek. Producent  $B$  realizuje pouze prodej  $S_B = 50$  jeho tržby z tohoto prodeje budou činit přibližně 150 jednotek. Zároveň je však nucen uhradit náklady na produkci i oněch 50 neprodaných kusů, které činí celkem 300 jednotek. Výrobce  $B$  tak vykáže v daném období ztrátu ve výši 150 jednotek.

Obdobně se postupuje i při získávání zbývajících hodnot tabulky výrobních strategií. Je nutno podotknout, že v případě stejných cen u obou výrobců se uvažuje rovnoměrné rozptřeni poptávky mezi tyto výrobce. Až je matice výrobních strategií sestavena, provede se výpočet Nashova equilibria, jehož výstupem jsou jednak pravděpodobnosti volby jednotlivých výrobních strategií oběma producenty, ale také jejich očekávaný zisk. Hodnoty očekávaného zisku jsou pak doplněny do matice cenových strategií.

Posledním krokem je výpočet Nashova equilibria z cenové matice, jehož výstupem jsou prodejní ceny, které mají jednotliví producenti požadovat za své produkty.

V našem příkladu je ze zvažovaných strategií pro producenta  $A$  při dané poptávce nejvýhodnější prodávat za cenu 8 a vyrábět 100 jednotek daného produktu. Producent  $A$  při této volbě strategií neprodá žádné výrobky a očekává ztrátu 600. Pro producenta  $B$  je ze zvažovaných strategií nejvýhodnější prodávat za cenu 7 a vyrábět 200 jednotek daného produktu, které při dané poptávce všechny prodá. Jeho očekávaný zisk je 300.

Tento příklad je pouze ilustrací použitého algoritmu. V systému jsou uvažovány cenové strategie v intervalu  $\langle V + 1, V + 11 \rangle$  a výrobní strategie v rozsahu od nuly až do menší z dvojice hodnot: očekávaná poptávka minus zásoby a produkční kapacita.

### 6.3.2 Volba sortimentu

Výstupem celého uvedeného postupu pro analýzu trhu jsou cenové a množstevní strategie, které se budou jednotliví producenti snažit uplatnit na trhu s daným produktem, aby maximalizovali svůj zisk, případně minimalizovali ztrátu.

Uvedený příklad demonstroval výpočet těchto strategií na jednom dílčím trhu. V předchozí kapitole jsme však viděli, že někteří výrobci vstupují na více než jeden dílčí trh. Pro zkrácení doby výpočtu bylo přijato zjednodušení, kdy je analyzován každý z dílčích trhů samostatně. Toto zjednodušení má však některé negativní dopady. Jedním z vážných problémů tohoto přístupu je vysoká pravděpodobnost nerealizovatelnosti produkčního plánu jako celku.

Jelikož analyzujeme každý dílčí trh zvlášť, předpokládáme, že celá výrobní kapacita daného producenta může být vyhrazena na výrobu pouze tohoto produktu. Uvažujme dva dílčí trhy  $T_1, T_2$  a dva producenty  $P_1, P_2$ . Uvažujme dále, že vypočtené produkční plány  $Q_1, Q_2$  přesahují výrobní kapacity obou producentů. Jak rozdělit výrobní kapacitu mezi oba dílčí trhy tak, aby ani na jednom z nich nevznikl příliš velký převis na straně poptávky?

Objektivně nejlepším řešením je upustit od uvažovaného zjednodušení a povýšit problematiku o další řád, kdy výstupy z matice cen budou použity jako vstupy matice pro alokaci výrobních kapacit mezi možné produkty. Tím by však výpočetní složitost neúměrně vzrostla a uvedený model by byl pro herní systém prakticky nepoužitelný.

Druhým řešením je rozložit výrobní kapacitu mezi jednotlivé části předimenzovaného produkčního plánu podle nějakého schématu. Problematika rozdělení nedostatečné kapacity prostředků mezi více úkolů je podobná problematice plánování procesů. Možná schémata tedy jsou:

- *FCFS* (first come first served),
- *SJF* (shortest job first),
- *Priority*,
- *RR* (Round Robin).

*FCFS* přístup lze interpretovat jako výrobu produktu s největším celkovým očekávaným ziskem. *FCFS* přístup může v některých situacích problém vyřešit. Je-li celkový očekávaný zisk z produkce  $P_1$  v plánu  $Q_1$  výrobce  $A$  větší než celkový očekávaný zisk z produkce  $P_2$  a v plánu  $Q_2$  výrobce  $B$  je tomu opačně, bude toto schéma fungovat. *FSFS* však nenabízí řešení obecně ve všech situacích.

*SJF* přístup spočívá ve splnění nejprve té části produkčního plánu, která zabere nejméně času. Po dokončení produkce se zahájí výroba produktu s druhou nejkratší celkovou dobou produkce. Tento přístup není správný, neboť se při jeho použití v řadě případů vyrábí nejprve produkt, po němž je nejmenší poptávka, a produkt s nejvyšší poptávkou až jako poslední. Na dílčím trhu s časově nejnáročnějším produktem tak vzniká největší převis na straně poptávky.

*Prioritní* přístup lze interpretovat jako výrobu nejprve té části výrobního plánu, která má nejvyšší prioritu. V navrhovaném systému ovšem nejsou zavedeny žádné priority, ale je řízen pouze tržními mechanismy. Za hodnotu priority však můžeme považovat očekávaný zisk ze splnění dané části produkčního plánu. Zdá se, že tím přejde prioritní přístup ve *FCFS*. Budeme-li však plnit plán s nejvyšší prioritou po částech a po dokončení každé části přepočteme očekávaný zisk dosažitelný dokončením zbývajících produkce, budou se priority postupně měnit a dojde tak

k přepínání mezi jednotlivými výrobami. Zároveň je použitím priorit nejprve vyráběn produkt s nejvyšším očekávaným ziskem. Tento mechanismus však nepřiděluje výrobní kapacity mezi všechny části výrobního plánu rovnoměrně. Jelikož jsou produkční plány na dané období vypočteny z predikovaných hodnot, může tato nerovnoměrnost spolu se špatnou predikcí vést ke značnému odchýlení nabídky od skutečné poptávky. Přesto lze tento přístup použít.

Posledním z uvedených mechanismů je *Round Robin*. Tento mechanismus rovnoměrně rozděluje výrobní kapacitu mezi všechny části produkčního plánu a zajišťuje rovnoměrné zaplnění všech dílčích trhů. Přestože Round Robin nebere ohledy na očekávané zisky, jeví se jako nejlepší řešení a to jak vzhledem k rovnoměrnému zaplnění trhu, tak i s ohledem na snazší implementaci. Tento mechanismus je v systému aplikován.

## 6.4 Řemeslníci

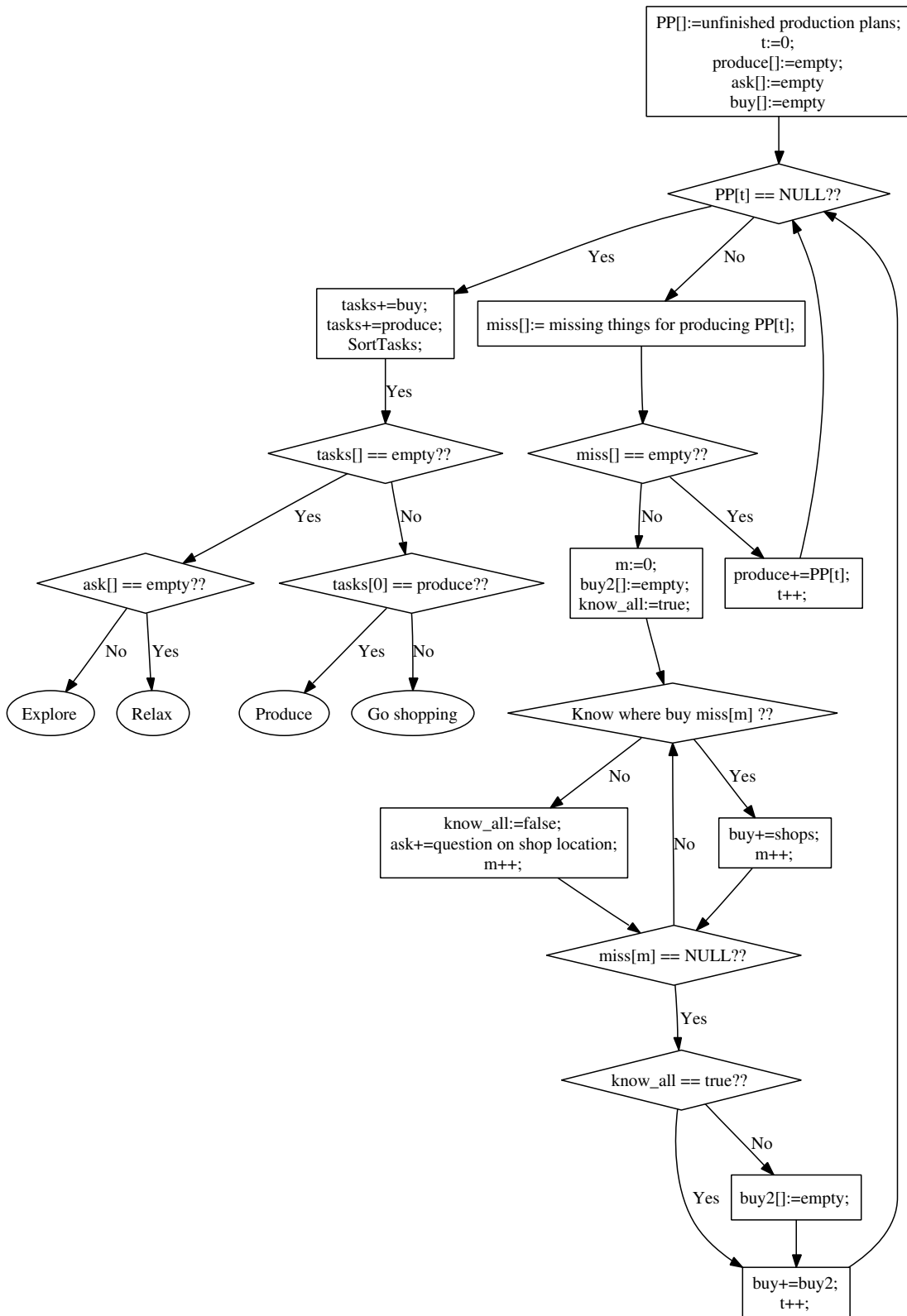
Podrobněji jsme se podívali na problematiku analýzy trhu a stanovení optimálních cen a objemů produkce na analyzované období. Tyto hodnoty tvoří produkční plán každého z řemeslníků a každý z řemeslníků se snaží tento plán v daném období splnit. Produkční plán sám o sobě však nedeterminuje volbu činnosti řemeslníka v každém okamžiku tohoto období. Na konkrétní chování mají vliv ještě další faktory. Část rozhodovacího algoritmu řemeslníka je znázorněna na obrázku 6.4. Kromě faktorů uvedených v tomto algoritmu hraje roli ještě zdraví a energie řemeslníka, faktor nebezpečí a další.

Uvažujme jednoduchý příklad řemeslníka, který zvažuje výrobu produktu  $P$ . Řemeslník nejprve zkontroluje, jestli má veškeré potřebné vstupy a náradí potřebné k výrobě. Zjistí-li, že má nedostatek materiálu pro realizaci výroby, bude hledat možnosti jeho doplnění. Předpokládejme, že řemeslník postrádá dva druhy materiálu,  $M$  a  $N$ . Řemeslník nalezne ve své paměti informace o možnosti nákupu zboží  $M$  v dílnách  $A$  a  $B$ . Každou z těchto dvou možností nákupu ohodnotí podle ceny, za kterou je možné zde nakoupit, a podle nákladů na cestu do každé z těchto dílen. Předpokládejme dále, že řemeslník nemá žádné informace o možnosti získat materiál  $N$ . Tím pádem nemá smysl nakupovat ani materiál  $M$ , neboť by zůstal nevyužit. Řemeslník tedy vloží do seznamu svých dotazů otázku na možnosti získání vstupu  $N$  a začne zvažovat výrobu dalšího produktu.

Výstupem analýzy všech produktů, které jsou v produkčním plánu, je množina strategií typu *Nakup* a *Vyrob*, ohodnocené očekávaným ziskem. Kromě těchto dvou množin je rovněž rozšířena množina dotazů řemeslníka na chybějící informace. Jsou-li všechny tři množiny prázdné, znamená to, že řemeslník má splněný celý svůj výrobní plán. Jsou-li prázdné pouze obě množiny strategií, pak řemeslník nemá dostatek informací ke splnění žádné části výrobního plánu a jeho nejlepší možností je vydat se na průzkum systému s cílem tyto informace získat. V posledním případě je alespoň jedna z množin strategií neprázdná. Všechny tyto strategie jsou setříděny podle očekávaného zisku a strategie s nejvyšším očekávaným ziskem je použita.

## 6.5 Modelování oponentů

V průběhu celé práce jsme se zaměřovali především na využití matematické teorie her v rozhodovacích a plánovacích procedurách. V této části se podíváme na odlišný přístup založený na modelování oponentů. Tento přístup může být použit jako doplněk, nebo jako alternativa mate-



Obrázek 6.4: Rozhodovací proces řemeslníka o výrobě

matické teorie her. Přestože hlavním tématem tohoto projektu je matematická teorie her, jako autor se cítím být povinen alespoň stručně zmínit a nastínit možnosti aplikace tohoto alternativního přístupu.

V části 6.1 jsme uvedli, že snahou tohoto projektu je postavit systém na neúplné a nedokonalé informaci. V části 6.2 jsme dále viděli algoritmus stanovení produkčních a cenových strategií založený na matematické teorii her. Jaký je důsledek aplikace matematické teorie her v tomto algoritmu?

Řekli jsme, že všichni konkurenti stanovují množinu svých strategií na základě jejich soukromých obchodních informací, jako jsou fixní a variabilní náklady atd. V uvedeném algoritmu však všichni konkurenti tyto množiny strategií "zveřejňují" doplněním do cenových a produkčních matic. Výpočet equilibrií z těchto matic je tedy stanovení strategií hráče při dokonalé znalosti možností konkurentů. Přesnější, a z hlediska skutečné podnikatelské praxe také realističtější, by byl přístup založený na odhadu možností konkurentů, který by se opíral o znalosti již dříve vzniklých situací na trhu. Každý z konkurentů by tak sestavil vlastní matice cenových a produkčních strategií, založených na jeho znalostech tržních situací, zkušenostech a odhadu možností konkurence.

V tomto odstavci se však nebudeme zabývat aplikací tohoto přístupu na modelování trhu, ale podíváme se na možnost aplikace v další důležité části projektu, kterou jsou souboje postav. Bohužel tato část není v současnosti v kódu implementována a to kvůli značnému rozsahu celého projektu a náročnosti implementace tohoto přístupu. Pokusíme se zde tedy pouze přiblížit možný postup řešení, který se opírá o přístupy prezentované v kapitole 3.

Každá herní situace týkající se soubojů postav je identifikována postavami, které se boje účastní. Uvažujme jednu z bojujících postav, postavu  $A$ . Postava  $A$  je schopna identifikovat všechny postavy ve svém zorném poli a ví, které postavy se boje účastní a které do boje nezasahují. O každé z bojujících postav může vyzozorovat množinu základních informací. Mezi tyto informace patří třída bojující postavy (řemeslník, lučištník, kouzelník, atd.), úroveň postavy (zkušenosti), současný a maximální počet životů, obratnost postavy, její vybavení, pozici a zda se jedná o spřátelenou či nepřátelskou postavu.

Podle současného a maximálního počtu životů lze určit, kolik procent životní energie postavě zbývá i absolutní velikost zranění, kterou ještě dokáže postava přežít. Z údajů o pozici a vybavení lze určit, zda postava představuje aktuální nebezpečí. Například vzdálený válečník bojující mečem nepředstavuje v daném okamžiku žádnou hrozbu, zatímco lučištník ano. Z údajů o obratnosti postav lze vyvozovat smysluplnost případného pronásledování postavy a úroveň postavy spolu s její třídou ovlivňuje techniky boje dostupné dané postavě.

Všechny tyto údaje o všech bojujících postavách, spolu s informacemi o postavě samotné, tvoří kompletní popis dané herní situace. Z databáze herních situací rozhodující se postavy je následně vyhledána situace, která se současné situaci nejvíce podobá. K tomu je potřeba přiřadit váhy jednotlivým informacím o postavách. Zřejmě nejvíce bude vážena informace o třídě postavy, následně o vybavení, pozici, atd. V databázi situací jsou kromě popisu situace uloženy také akce, které jednotlivé postavy v této situaci podnikly. Postava pak může podle míry podobnosti současné a vyhledané situace odhadovat pravděpodobnost, se kterou dané postavy zvolí právě tyto akce a podle toho reagovat. Všechny postavy v dané situaci tak podle informací obdržených z databáze situací zvolí akce, které posléze provedou. Tyto akce jsou vloženy do databáze situací spolu s popisem situace. Tím je zajištěno učení postav a postupné zpřesňování jejich informací

o jednotlivých typech nepřátel.

Jak však postupovat v prvních fázích hry, kdy je databáze situací prázdná, nebo obsahuje pouze několik málo prvků? V těchto případech se většinou situace nalezená v databázi od situace současné bude velmi lišit a nalezené informace budou tudíž prakticky nepoužitelné. Jedním z přístupů řešení tohoto problému může být přeceňování možností soupeře a tudíž volba obraných akcí s cílem především neutrpět zranění. Druhou možností je naopak soupeře podceňovat a volit agresivní útočné akce. Poslední uvažovaný přístup je založen na myšlence: "Podle sebe soudím Tebe.", kdy postava nejprve uváží, jak by se v situaci soupeřící postavy sama zachovala a následně hledá svoji nejlepší reakci na takové chování.

Uvedené problémy a jejich řešení zřejmě nejsou triviální. Detailní zvážení všech možných přístupů a faktorů důležitých pro tento model rozhodování bude předmětem případného dalšího pokračování práce na tomto projektu.

# Kapitola 7

## Experimenty a výsledky

*V této kapitole se podíváme na některé výstupy získané z běhu navrhovaného systému. Seznámíme se rovněž s uživatelským rozhraním tohoto systému a podíváme se na použité externí programy pro řešení herních matic.*

Tato část práce má být věnována prezentaci a interpretaci výstupů z běhu navrženého systému. Bohužel rozsah celého systému a oblast činností jednotlivých entit v systému je natolik široký, že neumožňuje tyto výstupy efektivně a přehledně interpretovat. V této kapitole se tedy zaměříme především na výsledky obdržené z popsaného modelu trhu. Ostatní výsledky, s odkazem na testování a na demopříklady, které jsou součástí přiloženého kódu, pouze stručně zhodnotíme.

Z obdržných výsledků vyplynulo, že všechny prvky popsané v kapitole 6 fungují správně a podle předpokladů. Prezentované obrázky 6.1 a 6.2 byly vzaty z jednoho z běhů programu a jsou důkazem správné funkčnosti modelu znalostí. Rovněž byla prokázána vhodnost zvoleného přístupu přepínání řemeslných výroby, která podle očekávání vede k rovnoměrnému zaplnění trhu.

### 7.1 Uživatelské rozhraní

Předmětem tohoto projektu je návrh a implementace simulačního jádra navrženého systému. Pro účely ladění a efektivního zadávání parametrů systému je však nutné navrhnout a implementovat i uživatelské rozhraní. Navržené uživatelské rozhraní je velmi triviální a umožňuje zadávání vstupu a prezentaci výstupů v jednoduchém textovém formátu.

#### 7.1.1 Vstupní formát

Při volbě programovacího jazyka bylo nutné vzít v úvahu i náročnost testování a ladění parametrů. Některé programovací jazyky, jako například jazyk Smalltalk, nabízejí uživateli efektivní rozhraní, jež umožňuje měnit stav programu za běhu a jednoduše tak vytvářet nejrůznější herní situace.

Použitý jazyk C++ takové rozhraní bohužel nenabízí, proto byl pro účely testování navržen jednoduchý formát vstupního souboru, který dovoluje nastavení základních parametrů systému, jehož simulace je následně spuštěna. Gramatika vstupního souboru vypadá takto:

```
< File >::=< TerrainSpec > [< LayoutList >][< Plans >]  
< TerrainSpec >::=< SizeX >< SizeY >< TerMatrix >  
< TerMatrix >::= { {< TerId >} * SizeY } * SizeX
```

```

< LayoutList > ::= { Layout }+
< Layout > ::= < GoodsSpec > | < BuildSpec > | < PersonSpec >
< GoodsSpec > ::= < GType > < Position > < Amount >
< BuildSpec > ::= < BType > < Position >
< PersonSpec > ::= < PType > < Position > < PersonName > < Sight > < Level >
< Plans > ::= ' Vyroba : ' { < ProdPlan > }+
< ProdPlan > ::= < PersonName > < GType > < Amount >
< Position > ::= < CoordX > < CoordY >
< PersonName > ::= { character }+
< Sight > ::= unsigned
< Level > ::= unsigned
< Amount > ::= unsigned
< SizeX > ::= unsigned
< SizeY > ::= unsigned
< CoordX > ::= unsigned
< CoordY > ::= unsigned

```

Vysvětlivky: [] ...nepovinnost, |...volba, {}+ ....nenulový počet opakování, {} \* *X* ....opakuje se *X*-krát.

Hodnoty *TerId* jsou klíčovými slovy typů jednotlivých povrchů a lze je najít v souboru *terrain.cpp*. Hodnoty *BType* jsou klíčovými slovy typů budov podle jednotlivých řemesel. Všechna tato slova mají prefix "*b\_*" a lze je nalézt v souboru *budova.cpp*. Hodnoty *GType* jsou klíčovými slovy typů zboží a surovin a lze je nalézt v souboru *goods.cpp*. Klíčová slova s příponou "*\_prim*" označují primární suroviny (nerostné bohatství). Hodnoty *PType* jsou klíčovými slovy typů profesí a lze je nalézt v souboru *profese.cpp*. Klíčová slova odpovídajících budov a profesí jsou pro lepší přehlednost stejná, profese však mají místo prefixu "*b\_*", prefix "*p\_*".

Umístíme-li zboží na pozici, kde je již budova nebo postava, bude toto zboží zařazeno do inventáře entity na této pozici. Umístění suroviny, postavy a budovy na tutéž pozici není možné. Všechny číselné hodnoty reprezentují informace podle názvu daného nonterminálu. Hodnota *PersonName* v *ProdPlan* musí být jménem již existující postavy. Na konci souboru nesmí být prázdný řádek.

Formát vstupního souboru byl navržen pro účely rychlého navození některých situací, nejedná se však o plnohodnotný editor map pro herní systém. Ve vstupním souboru není možné nastavovat konkrétní činnosti, které má postava provádět, ani nelze přepínat mezi částmi výstupu, které nás zajímají. Tyto úpravy je potřeba provést přímo ve zdrojovém kódu, nejlépe ve funkci *main()*.

Parser, starající se o interpretaci vstupního souboru, byl rovněž navržen pouze pro účely testování a je tedy velice triviální. Parser provádí minimum chybových kontrol vstupu. Některé špatně zadané údaje ve vstupním souboru tak nebudou do simulace zahrnuty, aniž by na to byl uživatel upozorněn. Parser rovněž není schopen zotavení, což u některých chyb vede k zacyklení.

### 7.1.2 Výstupní formát

Cílem projektu není implementace grafického rozhraní pro prezentaci dějů uvnitř simulace. Objem získávaných dat je však velice rozsáhlý a pouhý textový výstup by byl velice nepřehledný.



Například sledování cest, jimiž postavy chodí, na textové mapě bez barevného odlišení povrchů je velice obtížné.

Pro kvalitnější a lépe čitelný výstup byl vytvořeny konfigurační soubory pro editor *vim*, které definují barevné zvýraznění znaků, jež jsou použity pro značení jednotlivých typů terénů. Tyto soubory lze najít u zdrojových kódů v adresáři *Vim* spolu s návodem na jejich použití. S touto vizualizací lze, při znalosti náročností terénů, jednoduše najít optimální cestu a kontrolovat, zda jimi postavy chodí. Zdroje nerostného bohatství a budovy jsou v mapě značeny malým prvním písmenem jejich názvu, postavy velkým prvním písmenem jejich jména. Zvěř je značena symbolem '*x*'. Tento výstup je v některých případech nejednoznačný, ovšem pro účely jednoduché kontroly dějů dostačující. Výstup programu je generován na standartní chybový výstup. Pohyb postavy je na výstupu zachycen jménem postavy, která se právě hýbe, a vykreslením mapy po dokončení pohybu.

Před uzavřením části o výstupním formátu bych se chtěl zmínit o bakalářské práci Davida Knotka, která vznikla v tomto roce na FIT VUT v Brně a nabízí animované GUI z počítačové hry *Warcraft* v podobě aplikace jazyka Java, postavené na architektuře klient-server. Bohužel oba projekty vznikaly paralelně a nebylo tudíž prozatím možné toto GUI použít. Připojení tohoto rozhraní k navržené aplikaci může být součástí případné další práce na tomto projektu.

## 7.2 Projekt TRH

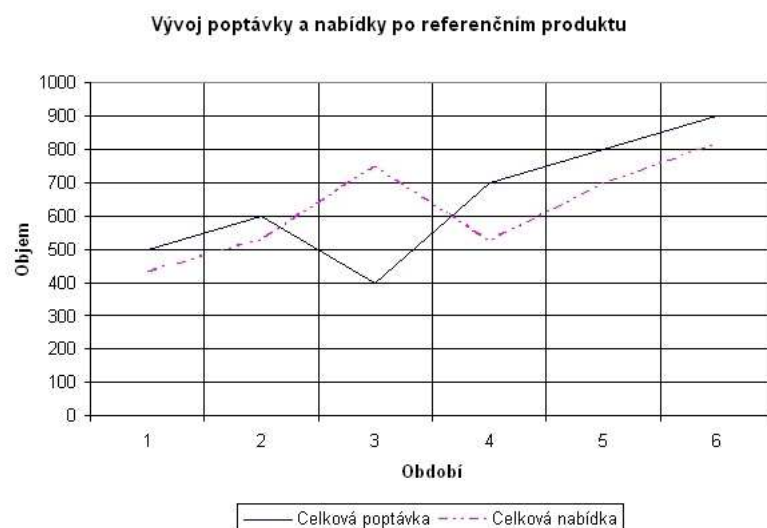
Jelikož procesů a faktorů ovlivňujících chování postav je v systému příliš mnoho, je velmi obtížné navodit některé tržní situace. Výstupy z herní simulace trhu jsou navíc ovlivněny všemi faktory ovlivňujícími postavy a ne pouze nabídkou a poptávkou. Je tak velmi obtížné ladit algoritmy pro výpočet parametrů vstupujících do analýzy trhu (viz kapitola 6.2). Další nevýhodou je pomalé generování výstupů, neboť analýza trhu je prováděna pouze jednou za určité období běhu systému.

Jako reakce na tyto problémy vznikl v rámci hlavního projektu podprojekt s názvem TRH. Tento podprojekt vznikl abstrahováním kódu pro tržní analýzu a neobsahuje žádné prvky ze simulačního jádra hry. Cílem projektu TRH je analýza jednoho dílčího trhu v několika po sobě jdoucích obdobích s ohledem na vývoj výrobních kapacit, fixních a variabilních nákladů a zásob všech konkurujících si producentů a také s ohledem na vývoj poptávky. Předmětem analýzy je dílčí trh s blíže nespecifikovaným referenčním produktem. Podívejme se v krátkosti na specifikaci tohoto podprojektu a konkrétní výstupy tohoto programu.

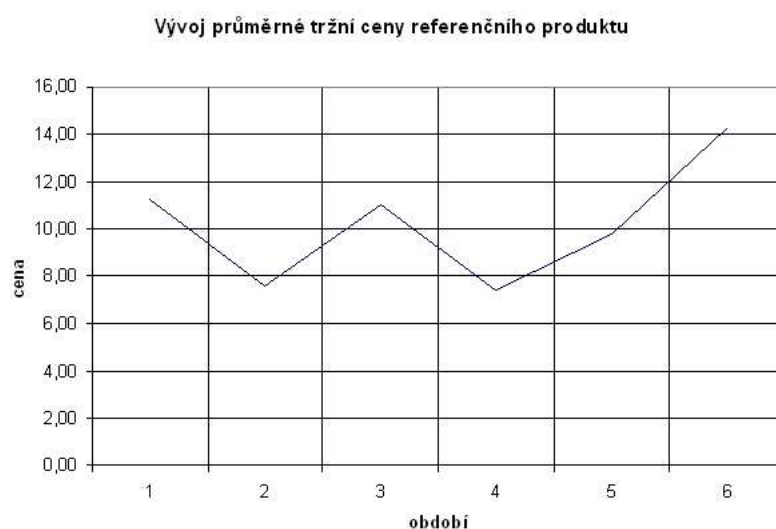
Vstup programu je zadáván ze souboru. Nebudeme zde uvádět celou gramatiku souboru, uvedeme pouze jednoduchý popis formátu. Formát souboru je *počet hráčů, počet období pro něž má simulace běžet, fixní náklady prvního hráče v obdobích 0-počet období, variabilní náklady prvního hráče v obdobích 0-počet období, výrobní kapacity prvního hráče v obdobích 0-počet období a zásoby s nimiž první hráč vstupuje do prvního období*. Takto postupně pro všechny hráče. Na závěr je uveden vývoj poptávky v obdobích 0-počet období.

Výstupem programu jsou zvolené strategie hráčů pro jednotlivá období, objem jejich prodeje, velikost zásob a dosažený zisk. Neprodané zboží z jednoho období se přenáší jako zásoby do období následujícího. Dále je zaznamenáván stav trhu jako celku pro jednotlivá období, tedy celková nabídka, celková poptávka, objem prodaného zboží a průměrná prodejní cena.

Algoritmus výpočtu je stejný jako algoritmus popsany v kapitole 6.2 a stejně jako v herním



Obrázek 7.1: Vývoj celkové poptávky a nabídky po referenčním produktu



Obrázek 7.2: Vývoj průměrné tržní ceny referenčního produktu

systemu jsou i v tomto projektu podporovány dva externí programy pro řešení herních matic, s nimiž se seznámíme v následujících odstavcích. Nejprve se však podívejme na jeden z výstupů projektu TRH (viz obrázky 7.1 a 7.2).

Na těchto grafech je zachycen vývoj průměrné tržní ceny a celkové nabídky v závislosti na vývoji celkové poptávky. Jelikož hodnoty poptávky pro analyzované období jsou predikovány jako celková poptávka v období uplynulém, reaguje vývoj ceny a celkové nabídky na změny poptávky se zpožděním jednoho období. Vidíme, že v období 2 došlo k nárustu celkové poptávky. V období 3, jako reakce na tento nárust, došlo k nárustu celkové nabídky i průměrné tržní ceny. V období 3 naopak došlo k prudkému poklesu celkové poptávky. Tím při dané výrobě vznikly producentům zásoby, které se v dalším období snažili prodat. Poklesla tak celková nabídka, stejně jako průměrná tržní cena. Z dalších údajů, které v těchto grafech nejsou zachyceny, vyplynulo, že producenti v tomto období vyráběli pouze minimum produktů, aby prodali své zásoby. Ukazuje se, že i přes značné zjednodušení tohoto modelu výsledky odpovídají základním jevům, ke kterým

na skutečném trhu opravdu dochází. Navržený model trhu lze tedy považovat za správný.

Na závěr této kapitoly se podíváme na již zmiňované externí programy a stručný popis algoritmů, které tyto programy nabízejí. Tyto aplikace jsou napojeny na můj systém a zastávají klíčovou úlohu při výpočtech Nashových equilibrií vzniklých herních matic.

### 7.2.1 Gametracer

Gametracer je opensource software pro výpočet Nashových equilibrií v maticových hrách. Tento program nabízí dva algoritmy pro hledání Nashových equilibrií v hrách více hráčů. Jsou to algoritmy *GNM* a *IPA*.

#### 7.2.1.1 GNM (Global Newton Method)

V dokumentaci k programu je GNM algoritmus popsán jako algoritmus využívající analýzu posloupnosti odchýlených her, která má počátek v zadané hře. Odchyly mají formát vektoru bonusů přidaných k ziskům jednotlivých hráčů v jejich čistých strategiích. Na konci této posloupnosti jsou bonusy již natolik velké, že převáží strukturu původní hry. Takto odchýlená hra je jednoduše řešitelná. Algoritmus potom trasuje equilibrium zpět posloupností odchýlených her, až se dostane k equilibrium původní hry. Tento algoritmus je schopen najít více equilibrií a všechna nalezená equilibria jsou přesná. Pro detailní popis algoritmu viz [14].

Bohužel je tato implementace algoritmu velmi nestabilní. Generování odchylek je založeno na náhodných číslech, proto pro stejné zadání hry program nalezne v několika bězích různá řešení. Navíc pro některá zadání hry je výpočetní doba extrémně dlouhá. Důvody tohoto jevu jsou objasněny v [10].

#### 7.2.1.2 IPA (Iterated Polymatrix Approximation)

IPA algoritmus se snaží vyhnout výpočetním problémům algoritmu GNM. IPA aproximuje v každém bodě hru pomocí bimatic (*bimatrix* - pojem matematické teorie her pro matice dvouhráčových her). V každé bimatici je spočtena interakce všech dvojic hráčů bez ohledu na interakci s ostatními hráči. Zisk hráče v uvažovaném strategickém profilu je potom součtem zisků ze všech těchto interakcí. Pro detailní popis algoritmu viz [10].

Přestože je tento algoritmus navržen především jako pomocný algoritmus pro GNM [10], implementace Gametraceru jej používá jako samostatnou metodu. Tento algoritmus je schopen najít jen jedno Nashovo equilibrium a to pouze jako aproximaci skutečného equilibria. Výhodou algoritmu je jeho výpočetní rychlost.

### 7.2.2 Gambit

Projekt Gambit je v současnosti zřejmě nejlepším softwarem pro řešení her a to jak v normální, tak i v extenzivní formě. Jedná se o opensource software přenositelný na systémy Linux a Windows. Na systémech Linux nabízí projekt Gambit čtyři režimy použití.

Prvním režim pracuje v interaktivním grafickém uživatelském rozhraní (viz obrázek 7.3). Toto rozhraní umožňuje přehledně zadávat vstupní údaje přímo do zobrazené herní matice a z jednoduchých menu volit různé algoritmy pro řešení těchto her. To umožňuje efektivně sledovat rychlost i přesnost jednotlivých algoritmů. V hlavním okně jsou, kromě herní matice, zachyceny i očekávané zisky jednotlivých hráčů a vypočtená Nashova equilibria.

The screenshot shows the Gambit software interface. The main window displays a game matrix for a 3-player game. The players are Player 1 (Payoff: -350, 0000), Player 2 (Payoff: 324, 0000), and Player 3 (Payoff: 363, 0000). The matrix is organized into four groups of strategies: 7, 10, 13, and 16. Each group contains four strategies (5, 8, 11, 14). The matrix cells contain numerical values representing payoffs for each combination of strategies. Below the matrix, there is a table titled 'All equilibria by solving polynomial systems in strategic game' with 4 rows and 14 columns representing different profiles.

	5	8	11	14
7	5: -350, 324 8: -336, -300 11: -336, -300 14: -336, -300	5: -105, 529 8: 744, -292 11: 744, -300 14: 744, -284	5: -300, 20 8: -350, 924 11: -330, 1756 14: -330, 1756	5: -105, 529 8: 744, -284 11: 744, -289 14: 744, -278
10	5: -350, 324 8: -350, 284 11: -330, -300 14: -330, -300	5: -350, 529 8: -350, 960 11: -330, 1756 14: -330, 1756	5: 0, 529 8: 0, 1624 11: 1938, -289 14: 1938, -278	5: 0, 529 8: 0, 1624 11: 1938, -289 14: 1938, -286
13	5: -350, 324 8: -350, 284 11: -350, 278 14: -324, -300	5: -350, 529 8: -350, 960 11: -350, 278 14: -324, -300	5: -350, 85 8: -350, 1624 11: -350, 1557 14: -324, -300	5: -105, 529 8: -105, 1624 11: 105, 2719 14: 3132, -286
16	5: -350, 324 8: -350, 284 11: -350, -278 14: -350, 272	5: -350, 529 8: -350, 960 11: -350, 1756 14: -350, 1756	5: -350, 85 8: -350, 1624 11: -350, 1557 14: -350, 272	5: -350, 190 8: -350, 1624 11: -350, 2719 14: -350, 2154

#	1: 7	1: 10	1: 13	1: 16	2: 5	2: 8	2: 11	2: 14	3: 5	3: 8	3: 11	3: 14
1	0.0000	0.0000	0.0000	1.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	1.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
3	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
4	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

Obrázek 7.3: Hlavní okno grafické aplikace Gambit

Druhý nabízený režim je čistě textový. Uživatel má k dispozici jednoduchý jazyk, specifikovaný v dokumentaci k projektu Gambit, pro zadávání matic a dalších příkazů. Tyto příkazy je možné zadávat buď na příkazové řádce programu Gambit, nebo ve formě skriptů programu Gambit.

Ve třetím režimu je projekt Gambit přeložen do formátu knihovny libgambit, kterou je možné "přilinkovat" k programu a využívat tak projekt Gambit jako interní součást navazující aplikace. Toto použití zpřístupňuje uživateli ze všech uvedených režimů nejvíce funkcí a operací nad analyzovanými herními maticemi.

Projekt Gambit při překladu vytváří, kromě výše uvedených programů a knihovny, také skupinu samostatných programů s jednotlivými algoritmy pro řešení her. Tyto programy přijímají ze standartního vstupu matice her ve specifikovaném formátu a generují na standartní výstup nalezená Nashova equilibria.

Z hlediska tohoto projektu by bylo zřejmě nejlepší použití projekt Gambitu jako knihovny. Bohužel překlad nejnovější verze projektu Gambit se na stroji, kde byl prováděn vývoj, nezdařil. Projekt Gambit byl proto nainstalován ve formě balíčku Debianu. Jelikož nelze spolu se zdrojovým kódem implementovaného systému dodat i přeložitelný zdrojový kód projektu Gambit, je jako základní část projektu dodáván program Gametracer. Program Gametracer nenabízí možnost překladu do formátu knihovny a je využíván jako externí program. Z těchto důvodů je jako externí program využíván i projekt Gambit. Přepnutí mezi použitím jednotlivých externích programů je možné v projektu TRH v souboru *options.h*, v herním systému potom v souboru *includes.h*.

Projekt Gambit nabízí kromě různých režimů použití také celou škálu algoritmů pro řešení her. Pro řešení her dvou hráčů jsou jako samostatné programy k dispozici gambit-enummixed, gambit-lcp a gambit-lp. Pro řešení her více hráčů jsou to programy gambit-enumpure, gambit-liap, gambit-logit, gambit-gnm, gambit-ipa, gambit-enumpoly a gambit-simpdiv.

Z uvedených možností jsou pro účely tohoto projektu použitelné pouze programy pro řešení her více hráčů, které si nyní přiblížíme. K programům gambit-simpdiv a gambit-logit se bohužel nepodařilo nalézt dokumentaci, která by srozumitelně a jasně popisovala principy použitých algoritmů. Oba tyto programy zaručují nalezení alespoň jednoho Nashova equilibria, a proto jsou pro účely tohoto projektu použitelné.

#### **7.2.2.1 Gambit-enumpure**

Gambit-enumpure je program pro hledání všech čistých strategií v hrách více hráčů. Tento program kontroluje všechny možné kombinace čistých strategií hráčů a testuje, zda se jedná o Nashova equilibria. Jelikož některé hry nemají Nashovo equilibrium v čistých strategiích, nezaručuje tento algoritmus nalezení Nashova equilibria, a tudíž je nepoužitelný pro účely navrhovaného systému.

#### **7.2.2.2 Gambit-liap**

Tento algoritmus je založen na problému minimalizace funkce. Touto funkcí je Lyapunova funkce pro Nashovo equilibrium. Tato funkce je nezáporná na celém definičním oboru a její funkční hodnota je rovna nule pouze v místech odpovídajících Nashovu equilibriu [15]. Podle dokumentace k projektu Gambit tento algoritmus nezaručuje nalezení Nashova equilibria a je proto pro účely navrhovaného systému nepoužitelný.

#### **7.2.2.3 Gambit-gnm a gambit-ipa**

Algoritmy GNM a IPA projektu Gambit jsou podle dokumentace k projektu převzaty přímo z projektu Gametracer. Přesto, že se má jednat o ekvivalentní implementaci, jsou výsledky algoritmu IPA z projektu Gambit odlišné od výsledků algoritmu IPA z projektu Gametracer. V některých případech jsou dokonce výsledky algoritmu IPA z projektu Gambit zcela nesmyslné.

#### **7.2.2.4 Gambit-enumpoly**

Tento program hledá všechna Nashova equilibria pomocí algoritmu prezentovaného v části 2.7.3. Algoritmus sestavuje systém polynomiálních rovnic a nerovnic pro všechny možné strategické profily. Tyto strategické profily jsou seřazeny tak, aby bylo možné využít maximum informací z výpočtů předchozích strategických profilů pro výpočet právě analyzovaného profilu. Tento algoritmus je ze všech uvedených algoritmů výpočetně nejnáročnější.

# Kapitola 8

## Závěr

Hlavním předmětem zájmu této diplomové práce bylo strategické rozhodování, kde výsledek rozhodnutí může ovlivnit také soupeř. Strategické rozhodování je jedním z obtížných problémů umělé inteligence, k jehož řešení přispívá stále více vědních disciplín jako psychologie, sociologie nebo ekonomie. Cílem tohoto projektu byla analýza a praktické předvedení aplikace matematické teorie her pro strategické rozhodování.

Jako demonstrace použití matematické teorie her bylo implementováno jádro multiagentního herního systému s tržním prostředím a jednoduchým uživatelským rozhraním. V průběhu práce byly analyzovány základní oblasti matematické teorie her a možnosti jejich aplikace v rozhodovacím procesu. Byly rovněž analyzovány některé alternativní postupy a navržena jejich možná aplikace na navržený systém.

Za hlavní přínos této práce považuji praktickou demonstraci použití matematické teorie her pro strategické rozhodování a předvedení implementace tohoto přístupu. Za nový lze přitom považovat heterogenní přístup k analýze navrženého modelu trhu, který vznikl kombinací dvou nejčastěji používaných přístupů - cenových strategií a množstevních strategií. Přesto, že se jedná o herní systém simulující království, lze jej považovat za zjednodušený model naší reality.

Další práce na tomto projektu by měly být věnovány především algoritmům pro řešení herních matic. Jako alternativní přístup k algoritmům, které byly uvedeny v této práci, se nabízí aplikace evolučních algoritmů. Přestože evoluční algoritmy nedokáží zaručit kvalitu nalezeného řešení, jsou schopny garantovat nalezení řešení v uživatelem stanoveném čase. Rovněž by měla být věnována pozornost zpřesňování navrženého modelu trhu a rozhodovacích algoritmů postav a dokončena část projektu zabývající se modelováním oponentů.

# Literatura

- [1] BOLDIŠ, P.: *Bibliografické citace dokumentů podle ISO 690 a ISO 690-2* [online]. c1999, poslední revize 2004.  
Dostupné z <http://www.boldis.cz/citace/citace.html>
- [2] ROSS, D.: *Game Theory*  
Stanford Encyclopedia of Philosophy,[online]. c2005  
Dostupné z <http://plato.stanford.edu/archives/win2005/entries/game-theory/>
- [3] Wikipedia The Free Encyclopedia, *Game theory* [online]. poslední revize 2006  
Dostupné z [http://en.wikipedia.org/wiki/Game\\_Theory](http://en.wikipedia.org/wiki/Game_Theory)
- [4] KUČERA, R.: *ABZ Slovník cizích slov* [online]. c2005-2006  
Dostupné z <http://slovník-cizich-slov.abz.cz>
- [5] NASH, J.: *Non-cooperative games*  
c1950, The Annals of Mathematics, 2. vyd., s. 286 - 295
- [6] ČVUT: *Teorie her a optimální rozhodování* [online].  
Dostupné z [http://euler.fd.cvut.cz/predmety/teorie\\_her/hry\\_materialy.html](http://euler.fd.cvut.cz/predmety/teorie_her/hry_materialy.html)
- [7] BLACKBURN, P. - O'SULLIVAN, B.: *Building reactive characters for dynamic gaming environments*  
c2005. Technical Report  
University College Cork, Cork Constraint Computation Center
- [8] CONZELMAN, G. - BOYD, G. - KORITAROV, V. - VESELKA, T.: *Multi-Agent Power Market Simulation using EMCAS* [online].  
Dostupné z [www.dis.anl.gov/ceesa/documents/ConzelmannBoydetal\\_rad29FBD.pdf](http://www.dis.anl.gov/ceesa/documents/ConzelmannBoydetal_rad29FBD.pdf)
- [9] MZLU Brno: *Mikroekonomie 1* [online]. poslední revize 2005  
Dostupné z <http://old.mendelu.cz/~econom/>
- [10] GOVINDAN, S. - WILSON, R.: *A global Newton method to compute Nash equilibria* [online]. poslední revize 2001  
Dostupné z [www.stanford.edu/class/cs324/papers/govindanwilson01newton-method.pdf](http://www.stanford.edu/class/cs324/papers/govindanwilson01newton-method.pdf)

- [11] GOVINDAN, S. - WILSON, R.: *Computing Nash equilibria by Iterated polymatrix approximation*  
 [online]. poslední revize 2001  
 Dostupné z <http://faculty-gsb.stanford.edu/wilson/archive/pdf/PolymatrixApproximationJEDC,010718.pdf>
- [12] STEFFENS, T.: *Similarity-based opponent modelling using imperfect domain theories*  
 [online].  
 Dostupné z [citeseer.ist.psu.edu/739255.html](http://citeseer.ist.psu.edu/739255.html)
- [13] PERINGER, P.: Popis simulační knihovny SIMLIB  
 [online]. c1997  
 Dostupné z <http://www.fit.vutbr.cz/~peringer/SIMLIB/doc/html-cz/.en>
- [14] HRUBÝ, M. - TOUFAR, J.: *Modelling the electricity markets using mathematical game theory*  
 c2006, 15th IASTED International Conference on APPLIED SIMULATION AND MODELING, s. 100 - 1007
- [15] McKELVEY, R.: A Liapunov function for Nash equilibria  
 c1991, Technical Report  
 California Institute of Technology



## Příloha A - Tabulka závislostí mezi jednotlivými druhy zboží

zboží	vstup 1	vstup 2	vstup 3	nástroj 1	nástroj 2	nástroj 3
palice	ocel	voda	uhlí	x	x	x
pila	ocel	voda	uhlí	palice	x	x
podkova	ocel	voda	uhlí	palice	x	x
ruchadlo	ocel	voda	uhlí	x	x	x
sekera	ocel	voda	uhlí	palice	x	x
sekáček	ocel	voda	uhlí	palice	x	x
mouka	obilí	x	x	mlýnský kámen	x	x
chleba	mouka	voda	uhlí	x	x	x
prkno	dřevo	x	x	pila	hoblík	x
slad	zrno	x	x	x	x	x
pivo	slad	voda	uhlí	sud	x	x
srnčí maso	srnec (prim.)	x	x	sekáček	palice	x
vepřové maso	vepř	voda	x	sekáček	palice	x
ryba	rybník (prim.)	x	x	rybářský prut	x	x
semínko	zrno	x	x	x	x	x
zrno	obilí	x	x	cep	x	x
obilí	semínko	x	x	ruchadlo	kosa	kůň
ocel	železo	uhlí	x	x	x	x
bronz	zlato	voda	uhlí	palice	x	x
měď	zlato	voda	uhlí	palice	x	x
kůň	obilí	voda	x	žlab	hříbě	x
hříbě	podkova	x	x	x	x	x
voda	pramen (prim.)	x	x	vědro	x	x
sud	ocel	prkno	hřebík	palice	x	x
hoblík	ostří hoblíku	dřevo	x	rašple	x	x
velká násada	dřevo	x	x	rašple	hoblík	sekera
vědro	prkno	hřebík	x	majzlík	palice	rašple
žebřík	prkno	hřebík	x	palice	hoblík	pila
šrot	vepřové maso	zrno	x	x	x	x
vepř	šrot	voda	x	koryto	sele	x
meč	ocel	bronz	x	palice	majzlík	x
štít	měď	bronz	x	palice	x	x
luk	dřevo	ocel	x	sekera	rašple	hoblík
zeď	kámen	x	x	majzlík	palice	x
střecha	prkno	hřebík	x	žebřík	palice	sekera
kámen	kámen (prim.)	x	x	krumpáč	velká násada	x
zlato	zlato (prim.)	x	x	krumpáč	velká násada	x
železo	železo (prim.)	x	x	krumpáč	velká násada	x
uhlí	uhlí (prim.)	x	x	krumpáč	velká násada	x
dřevo	strom (prim.)	x	x	sekera	malá násada	x

## Příloha B - Vstupní souboru programu

11 21

W W W G G M M M S G G G G G G G G G G W  
W W W W G M M S S G G G G G G G G G W W  
W W W W S S S S G G G G G G G G G W W W  
W W W S W G G G G M G G G G G W W W W  
W W W S G N N M M M M G G G G G G W W W  
W W S G G N M M M M G M G G G G G W G G  
S S G S S M M M M M G G M G G G G G G G  
G G S S M M M M M M G M G G G G G G R R  
G S S M M M M M M M G G G G G G R R G  
G G R R M M M L L L L L R R R R R R G G  
G R R R R M L L L L L L R R R R R R R G R

pramen\_prim 4 17 1000

b\_studnar 0 16

p\_studnar 1 16 Studnar 5 1

vedro 0 16 1

b\_farmar 2 14

zrno 2 14 100

seminko 2 14 100

obili 2 14 10

cep 2 14 1

ruchadlo 2 14 1

kosa 2 14 1

kun 2 14 1

p\_farmar 2 13 Farmar 5 1

b\_mlyn 3 11

mlyn\_kamen 3 11 2

p\_mlyn 3 12 Janis 5 1

penize 3 11 1000

uhli\_prim 5 11 1000

b\_hornik 8 12

p\_hornik 8 13 Hornik 5 1

krumpac 8 12 1

v\_nasada 8 12 1

b\_pekar 5 14

p\_pekar 5 13 Pekar 5 1

penize 5 14 1000

strom\_prim 5 17 1000

b\_drevor 9 15

sekera 9 15 1

m\_nasada 9 15 1

penize 9 15 1000

p\_drevor 9 14 Drevorubec 5 1

b\_pila 9 10  
pila 9 10 1  
hoblik 9 10 1  
penize 9 10 1000  
p\_pila 9 11 Pilar 5 1  
b\_truhlar 4 15  
hoblik 4 15 1  
hrebik 4 15 1000  
majzlik 4 15 1  
palice 4 15 1  
rasple 4 15 1  
p\_truhlar 4 16 Truhlar 5 1  
b\_pivovar 2 10  
sud 2 10 1  
penize 2 10 1000  
p\_pivovar 1 10 Bier 5 1  
srnec\_prim 5 3 100  
kamen\_prim 0 8 1000  
b\_kamenik 1 11  
krumpac 1 11 1  
v\_nasada 1 11 1  
palice 1 11 1  
majzlik 1 11 1  
p\_kamenik 0 11 Kamenik 5 1  
Vyroba:  
Studnar voda 100  
Farmer obili 100  
Farmer zrno 40  
Farmer seminko 100  
Janis mouka 100  
Hornik uhli 100  
Pekar chleba 100  
Drevorubec drevo 300  
Pilar prkno 300  
Truhlar vedro 5  
Bier slad 100  
Bier pivo 100  
Kamenik kamen 100  
Kamenik mlyn\_kamen 10