



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Jan Babušík

Umělá inteligence pro hru Dominion

Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: Mgr. Martin Pilát, Ph.D.

Studijní program: Informatika

Studijní obor: Programování a softwarové systémy

Praha 2019

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

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

V dne

Podpis autora

Chtěl bych poděkoval Mgr. Martinu Pilátovi, Ph.D. za podnětné rady, trpělivost a čas, který mi během této práce věnoval. Také bych rád poděkoval svým rodičům za podporu v mém dosavadním studiu.

Název práce: Umělá inteligence pro hru Dominion

Autor: Jan Babušík

Katedra: Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: Mgr. Martin Pilát, Ph.D., Katedra teoretické informatiky a matematické logiky

Abstrakt: Předmětem této bakalářské práce je umělá inteligence pro karetní hru Dominion, která je použitelná ve hře dvou hráčů. Hra Dominion se vyznačuje tím, že existuje velké množství počátečních konfigurací. Umělá inteligence vybírá nejlepší strategie z množiny připravených strategií, aby čas generování specializované strategie nezdržoval hráče před každou hrou. Tento přístup byl implementován s důrazem na rozšiřitelnost hry. Při návrhu umělé inteligence bylo využito evolučních algoritmů.

Součástí práce je také implementace samotné hry v jazyce C# a jednoduché grafické rozhraní pro hraní hry.

Klíčová slova: umělá inteligence, karetní hra, Dominion

Title: Artificial Intelligence for the Dominion Game

Author: Jan Babušík

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: Mgr. Martin Pilát, Ph.D., Department of Theoretical Computer Science and Mathematical Logic

Abstract: The subject of this thesis is an artificial intelligence for the Dominion card game, which is usable in a two-player game. Dominion is characterized by the fact that there is a large number of initial configurations. The artificial intelligence selects the best strategies from a set of prepared strategies so that the time it takes to generate a specialized strategy does not delay players from every game. This approach has been implemented with emphasis on game extendability. In the design of the artificial intelligence evolutionary algorithms were used.

The work also includes an implementation of the game itself in the C# language and a simple graphical interface for playing the game.

Keywords: artificial intelligence, card game, Dominion

Obsah

Úvod	3
1 Pravidla hry Dominion	4
1.1 Typy karet	4
1.2 Příprava hry	4
1.3 Průběh kola	4
1.4 Konec hry	5
1.5 Slovník pojmů	5
1.6 Často používané pokyny na akčních kartách	6
2 Rozbor problému	7
2.1 Vlastnosti hry	7
2.2 Související práce a inspirace	7
2.3 Evoluční algoritmy	8
2.3.1 Explorace vs. exploatace	9
3 Popis umělé inteligence	10
3.1 Počáteční konfigurace	10
3.2 Obecný přístup evoluce	10
3.2.1 Herní strategie	10
3.2.2 Nákupní strategie	10
3.2.3 Reprezentace jedince	10
3.2.4 Evaluace	11
3.2.5 Mutace	11
3.2.6 Parametry evoluce	12
3.2.7 Výsledky evoluce	12
3.3 Podmnožiny	15
3.3.1 Spolupracující karty	15
3.3.2 Pětice	16
3.3.3 Trojice	17
3.3.4 Výsledky	17
3.3.5 Hodnocení karet	19
3.3.6 Hodnocení karetních kombinací	20
4 Uživatelská dokumentace	21
4.1 Nastavení	21
4.1.1 Volba království	21
4.1.2 Nastavení protihráče a jména	22
4.1.3 Start hry	22
4.2 Průběh hry	23
4.2.1 Popis rozhodování	23
4.2.2 Konec hry	24

5 Implementace	25
5.1 Knihovna GameCore	25
5.1.1 Game	25
5.1.2 Player	26
5.1.3 User	26
5.1.4 Card	26
5.1.5 Pile	26
5.1.6 Kingdom a KingdomWrapper	27
5.2 Uživatelské rozhraní	27
5.3 Utils	27
5.3.1 ThreadSafeRandom	27
5.3.2 Subsets	28
5.4 AI dokumentace	28
5.4.1 Evolution	28
5.4.2 Provincial	29
5.4.3 Model	31
Závěr	32
Seznam použité literatury	34
A Karty	35
B Priority	37

Úvod

Dominion je velice populární karetní hra, která důkladně otestuje hráčovo strategické myšlení a schopnost aplikovat dříve získané zkušenosti za okolností, jež mohou být v každé hře jiné. Je to velice variabilní hra a existuje pro ni hodně rozšíření, díky čemuž zůstávají stále nové možnosti i po odehrání velkého množství her.

Cílem této bakalářské práce je implementovat simulaci Dominionu a umělou inteligenci soupeře, jež by byla výzvou i pro zkušenější hráče Dominionu. Kvůli zmíněné variabilitě je ale většina již existujících implementací specializovaná jen pro základní hru a vyžaduje netriviální čas na předpočítávání dat, nebo obsahuje některá rozšíření, ale vyžaduje jisté čekání před zahájením každé hry.

Vyzkouším tedy jiný přístup k implementaci umělé inteligence, který redukuje čas potřebný ke specializaci před každou hrou, a pokusím se co nejvíce zjednodušit případné budoucí přidávání dalších karet.

Dalším cílem práce je také analýza užitečnosti karet a karetních kombinací. Tato analýza může být přínosná při různých úpravách nutných pro přidání nových karet.

1. Pravidla hry Dominion

Dominion je deck-buildingová karetní hra, kterou v roce 2008 vytvořil Donald X. Vaccarino, vydává ji firma Rio Grande Games a do češtiny překládá firma Albi. Základním principem hry je, že každý hráč má vlastní balíček karet představující jeho říši (dominium), který si pak pomocí různých akcí upravuje a na konci hry za něj dostane body. Popis pravidel je založen na zdrojích [1, 2, 3]

1.1 Typy karet

V základní sadě pro dominion existují čtyři typy karet.

- žluté, Peněžní (Treasures)
- zelené, Vítězné body (Victory cards)
- bílé, Akční karty (Action cards), které se dále dělí na akční, akční-útočné a akční-reakční.
- fialové, Kletba (Curse)

1.2 Příprava hry

Vytáhneme základní peněžní (*Měďák* \$1, *Stříbrňák* \$2 a *Zlaták* \$3) a vítězné (*Statek* (1 vítězný bod), *Vévodství* (3 VB) a *Provincie* (6 VB)) karty, které se účastní každé hry. V základní sadě dominion existuje 25 různých karet království, ze kterých vybereme deset. Všechny tyto karty budou součástí společné zásoby karet a všichni hráči je budou moci získat.

Každý hráč si buduje vlastní balíček karet a na začátku hry dostane tři *Statky* a sedm *Měďáků*. Tyto karty si zamíchá a položí na místo, kde bude mít *dobírací balíček*. Tento balíček má hráč otočený lícem dolů, aby nevěděl, jaké karty mu přijdou. Existují však akční karty, které mu umožní se na některé z nich podívat. Poté si z tohoto balíčku vezme pět karet do ruky. Dále má hráč před sebou zatím prázdný *odkládací balíček*, na který se karty dávají lícem nahoru. Nakonec má hráč před sebou *hrací prostor*, kam bude dávat zahrané karty.

1.3 Průběh kola

Hráči se postupně střídají po směru hodinových ručiček. Tah se skládá z následujících fází v tomto pořadí.

- **1. Akční fáze:** Hráč má možnost zahrát jednu akci. Vybere akční kartu, kterou chce zahrát a provede instrukce, jež jsou na ní uvedené. Typické instrukce i popisy všech karet jsou uvedeny později. Pokud je zahraná karta kartou útoku, ostatní hráči mají možnost ukázat z ruky libovolné množství karet reakcí a vykonat jejich reakční efekt. Například karta *Hradní příkop* hráči umožňuje ubránit se útoku a útok se jej pak netýká. Pokud akce karty

přikazuje zahodit nějakou kartu na smetiště, tato karta se nevrací zpět do balíčku ve společné zásobě, ale odkládá se stranou na tzv. *smetiště*.

- **2. Nákupní fáze:** Karty ze společné zásoby mají v levém dolním rohu uvedenou svou cenu. Nejprve může hráč zahrát libovolné množství karet peněz z ruky. Zahrané peníze se sčítají s penězi, které získal během akční fáze. Za tyto peníze si může koupit maximálně tolik karet ze společné zásoby, kolik má nákupů. Získané karty se odkládají na odkládací balíček. Karty peněz, které hráč použil při nákupní fázi, se nevrací zpět do balíčku peněz ani se nezahazují na smetiště.
- **3. Úklidová fáze:** Hráč odloží všechny zahrané karty a všechny karty z ruky na odkládací balíček a dobere si pět karet z dobíracího balíčku. Hráč si nemůže nechat žádné karty do dalšího kola. Pokud v dobíracím balíčku není dostatek karet, hráč vezme odkládací balíček, zamíchá ho a položí pod zbývající karty v dobíracím balíčku.

1.4 Konec hry

Karty království ve společné zásobě jsou obvykle v deseti kopiích. Hra končí, jakmile jsou vykoupeny tři balíčky karet nebo všechny *Provincie*. Vyhrává hráč s největším počtem vítězných bodů ve svém balíčku. Při shodě vyhrává hráč, který odehrál menší počet tahů.

1.5 Slovník pojmů

- *Základní karty:* Karty, které se účastní každé hry, tj. některé karty peněz (*Měďák*, *Stříbrňák* a *Zlaták*) a některé karty vítězných bodů (*Statek*, *Vévodství* a *Provincie*).
- *Karty království:* Deset karet, které udávají každé hře její charakteristiku. Vybírají se libovolným způsobem ze všech karet království, které obsahují karty akcí, peněz i vítězných bodů.
- *Dobírací balíček:* Balíček karet, ze kterého si hráč tahá karty na konci kola nebo když mu to umožní některá z akčních karet. Karty jsou otočeny lícem dolů.
- *Odkládací balíček:* Balíček karet, kam se odkládají karty při pokynu *odlož*, při nákupu karty nebo ve úklidové fázi.
- *Hrací prostor:* Místo pro odkládání zahraných karet.
- *Smetiště:* Balíček, kam se dávají karty při pokynu *zahod na smetiště*. Karty ze smetiště se nedají nakoupit nebo získat pomocí karet s akcí *Vezmi si*.
- *Karty v ruce:* Nezahrané karty v ruce.
- *Název karty:* Text uvedený v záhlaví karty.
- *Cena karty:* Číslo uvedené na kartě vlevo dole.

1.6 Často používané pokyny na akčních kartách

- *+X karet:* Hráč si musí ihned dobrat do ruky X karet ze svého dobíracího balíčku. Pokud je dobírací balíček prázdný, zamíchá si odkládací balíček a připraví si jej jako nový dobírací balíček.
- *+X akcí:* Hráč může zahrát dalších X akčních karet. Předtím, než vyloží další akční kartu, musí provést (pokud možno) všechny pokyny na právě zahrané akční kartě.
- *Odlož kartu:* Odlož kartu z ruky (pokud není uvedeno jinak) na odkládací balíček.
- *Zahod kartu:* Zahod kartu z ruky (pokud není uvedeno jinak) na smetiště.
- *Vzít si kartu:* Hráč si vezme kartu (většinou ze společné zásoby) a položí ji na svůj odkládací balíček. Nejedná se o nákup karty.
- *Ukaž kartu:* Hráč ukáže vrchní kartu ze svého dobíracího balíčku všem hráčům a pak ji položí lícem dolů zpět. Pokud nemá dost karet v dobíracím balíčku, musí zamíchat svůj odkládací balíček a umístit ho na místo dobíracího.
- *+X nákupů:* Hráč si může ve fázi nákupu koupit o X karet více.
- *+X \$:* Hráč má k dispozici X virtuálních peněz ve fázi nákupu. Nedostává žádné fyzické karty peněz.

Úplné informace o kartách jsou k dispozici v příloze A

2. Rozbor problému

V této kapitole bych rád rozebral, jakými způsoby by bylo možné řešit umělou inteligenci pro Dominion.

2.1 Vlastnosti hry

Vlastnosti hry, které je dobré vzít v úvahu při vytváření umělé inteligence.

- **Velký počet počátečních konfigurací:** Základní hra Dominion má 25 karet království, ale každé hry se účastní jen 10. To je celkem 3 268 760 různých her jen v základní hře, která navíc obsahuje jen velmi jednoduché karty. Každé rozšíření přidává 10–30 nových karet království. Celkem je do češtiny přeloženo asi 200 karet království, v originále jich je dokonce přes 300, což dělá asi 10^{18} různých her. Z tohoto důvodu není možné vyvíjet strategii zvlášť pro každé království.
- **Hodnocení vítěze:** Na konci hry se počítají pouze vítězné body, ale tyto karty jsou během hry k ničemu a naopak hráče omezují. Je tedy důležité nezačít sbírat vítězné body příliš brzo, aby hráč nebyl zpomalen, ani příliš pozdě, když už není ve společné zásobě dostatek bodů.
- **Náhoda:** I když hráč ví, jaké karty má v balíčku, pořadí, v jakém si karty vytáhne, je náhodné.
- **Neúplná informace:** Hráči neví, jaké karty mají ostatní hráči na ruce, ani jaké karty dostanou do ruky v příštím kole.
- **Interakce s ostatními hráči:** Přestože spousta království není příliš interaktivní, na strategii má velký vliv, jak rychle se vykupují karty.
- **Podobnosti mezi kartami:** Mezi kartami se dají nalézt různé podobnosti, například velká část karet přidává jednu akci a jednu kartu, takže jejich zahrání nijak nesnižuje možnosti hráče pro toto kolo.
- **Podobnosti mezi královstvími:** I přes obrovský počet království většina strategií používá jen tři až pět různých karet, čímž se snižuje počet různých strategií. Některá království navíc můžou mít tak špatné kombinace karet, že prostá strategie kupování peněz a bodů nemusí být úplně špatná.
- **Herní a nákupní strategie:** Hra může být rozdělena na herní a nákupní strategii. Herní strategie se snaží zahrát karty v ideálním pořadí. Nákupní strategie se snaží nakupovat karty v takovém pořadí, aby herní strategie byla účinnější.

2.2 Související práce a inspirace

V této kapitole si předvedeme práce, které se také zabývají tímto problémem a z nichž bych mohl případně vycházet.

V práci *Developing an agent for dominion using modern AI-approaches* z roku 2010 dokázali R. B. Fynbo a C. S. Nellemann [4] úspěšně implementovat agenta pro Dominion pomocí rozdělení problému na dvě oddělené sekce: herní a nákupní strategii. Nákupní strategie je implementována pomocí koevoluce a neuro-evoluce a pracuje se všemi kartami království najednou. Herní strategie je vyřešena umělými neuronovými sítěmi. Když autoři porovnávali efektivitu, zjistili, že je obtížné natrénovat herní strategii tak, aby byla lepší než heuristika, kterou by zkušenější hráč mohl odhadnout. Investice do vývoje takového řešení se tedy pravděpodobně nevyplatí. Nevýhodou tohoto přístupu je chybějící možnost jeho adaptace na rozšíření hry.

Práce *Provincial: A kingdom-adaptive AI for Dominion*, napsána v roce 2014, [5] částečně vychází z předchozí uvedené a také pro herní strategii používá pouze heuristiku. Za cíl si ale bere implementaci několika dalších rozšíření, které umožňují zajímavější strategie než relativně jednoduché karty vyskytující se v základní hře. Pro autory je rovněž klíčová relativní srozumitelnost agenta, jež by skutečnému hráči umožnila naučit se nové postupy při hraní. Nákupní strategie je implementována pomocí koevoluce, ale agent se vyvíjí až za běhu a pouze pro specifické království. To umožňuje neřešit problém s obrovským počtem možných království, ale před každou hrou je potřeba 5–10 minut počkat, než se vyvine. Další nevýhodou tohoto přístupu je, že agent nezná celkový známý stav hry tak jako lidský hráč a nedokáže flexibilně reagovat na strategii protihráče. Jinou otázkou zůstává, jestli se v Dominionu vůbec vyplatí reagovat změnou strategie na protihráčův tah a riskovat rozbití vlastní strategie.

Ve své práci budu vycházet hlavně z druhého přístupu, protože chci, aby hra byla rozšiřitelná, ale zároveň se budu snažit redukovat dobu, po kterou musí hráč čekat před každou hrou.

2.3 Evoluční algoritmy

Evoluční algoritmus je přístup inspirovaný Darwinovou evoluční teorií[6]. Evoluční teorie je popis situace, kdy jedinci s nejvyšší schopností přežít budou mít pravděpodobně nejvíce potomků a jejich gen se rozšíří. Prostředí se však může zároveň měnit, a proto nejsou potomci jen kopií svých rodičů, ale jejich kombinací, a do hry vstupuje také mutace, čili náhodná změna genu nepocházející od rodiče. Schopnost nového jedince přežít může být jiná než u rodiče a nový jedinec tak může být lépe přizpůsoben prostředí.

Podobný způsob můžeme zvolit i v přístupu k nějakému složitému problému[7], který nelze řešit žádným přímým algoritmem, například hledání maxima libovolné složité funkce. Jedinec představuje možné řešení zvoleného problému a algoritmus pracuje s celou množinou těchto řešení, které říkáme populace, najednou. Funkce, jejíž maximum hledáme, nazveme *fitness* a bude umět vyhodnotit jedince, takže bude umět určit schopnost jedince přežít. Jedinci s vyšší schopností přežít budou mít větší šanci se rozmnožit, tomuto se říká selekce. Používá se například *ruletová selekce*, kdy šance, že bude jedinec vybrán ke křížení, je přímo úměrná jeho fitness. Nový jedinec může vzniknout jako kombinace dvou nebo více *rodičů* a používají se k tomu operátory křížení. Pokud je jedinec reprezentován jako pole čísel, operátor křížení může být například tzv. jednobodové křížení. Zvolí se náhodný index pole, od začátku do indexu bude mít potomek geny otce a zbytek

budou geny matky. Takovýmto způsobem se tvoří další jedinci, dokud jich není požadovaný počet. Do další generace vstupují obvykle pouze potomci, ale není to pravidlo. Nová generace je poté vyhodnocena stejným způsobem jako stará a algoritmus se neustále opakuje. Jedinci takovýmto způsobem prohledávají definiční obor funkce.

Algorithm 1 Obecný algoritmus evoluce.

```
t = 0
P0 = náhodně vygenerovaná generace
while t < tmax do
  for all i in 0..Pt+1.Count do
    rodic0 = Vyselektuj((Pt))
    rodic1 = Vyselektuj((Pt))
    Pt+1[i] = operatorKřížení(rodic0. rodic1)
    if random < pravděpodobnostMutace then
      Pt+1[i] = mutuj(Pt+1[i])
    end if
  end for
  t = t + 1
end while
```

2.3.1 Explorace vs. exploatace

Explorace je prohledávání nových oblastí funkce, *Exploatace* značí využívání stávajících znalostí a prohledávání bližšího okolí jedince. Evoluční algoritmy čelí dvěma hlavním problémům. Pokud se potomci příliš liší od rodičů, je pro evoluci těžké trefit se přímo do optimální hodnoty, zvláště když je tento prostor hodně úzký. Tomu se říká přílišná explorace. Na druhou stranu, když se potomci rodičům příliš podobají, jedinci často sklouznou do lokálního optima a nehledají dále lepší hodnoty.

3. Popis umělé inteligence

3.1 Počáteční konfigurace

Na začátku má hráč vždy deset karet, z toho tři jsou *Statky* a sedm *Měďáků*. To znamená, že v prvním ani ve druhém kole nebude mít na ruce žádné akční karty a během těchto kol bude moci vykonat dohromady jen dva nákupy. Na ně bude mít buď 4\$ a 3\$, nebo 5\$ a 2\$, pořadí se může lišit. Po druhém kole vždy dojde k zamíchání karet, tudíž na pořadí nezáleží. Tato počáteční konfigurace značně ovlivňuje celkovou strategii pro danou hru. V prvním případě může mít hráč během několika počátečních kol problém koupit si kartu za 5\$. Pokud nastane druhá situace, značně to devaluje hodnotu karet s cenou 3\$ a 4\$, které jsou užitečné zejména na začátku hry.

Vývoj inteligence pro obě možnosti najednou by byl značně zpomalen a výsledky by byly zkresleny. Z těchto důvodů jsem se rozhodl oddělit od sebe tyto případy a počáteční konfiguraci napevno určit. Úplná strategie by pak v sobě obsahovala obě možnosti a konkrétní strategie by byla vybrána až podle dané situace na začátku hry.

Dále se tedy budu zabývat jen konfigurací 5\$/2\$. Pro konfiguraci 4\$/3\$ funguje evoluce obdobně, ale její vývoj by byl časově náročný a nepřinesl by mnoho nových poznatků.

3.2 Obecný přístup evoluce

3.2.1 Herní strategie

Pro zjednodušení problému budu vycházet z již zmíněných prací, herní strategii budu implementovat pouze pomocí heuristiky a nebudu se snažit o její vyvíjení. Herní strategie je dána seznamem karet, který značí jejich prioritu při hraní. Tento seznam bude pro každou strategii stejný, protože pouze v malém množství případech se skutečně mění. Některé z nich jsou naprogramovány ve třídě *PlayAgenda*. Heuristika spočívá hlavně v tom, že pokud má hráč jen jednu akci, upřednostňuje karty, které přidávají akce, zatímco když má akcí více, upřednostňuje spíše možnost líznout si další kartu. Prioritu některých karet navíc ovlivňují ostatní karty v ruce, například karta kaple má vyšší prioritu, pokud má hráč v ruce také karty kletby. Herní strategie je podrobněji rozepsána v kapitole 5.4.2.

3.2.2 Nákupní strategie

Nákupní strategii řeším pomocí evolučních algoritmů velice podobným způsobem, jakým je k problému přistupováno ve druhé zmíněné práci [5]. Inteligence je pro každé království vyvíjena nezávisle.

3.2.3 Reprezentace jedince

Jedinec je čtveřice (P, V, S, M) , kde P , V a S jsou čísla a M je seznam dvojic karta-počet (dále nákupní menu). Agent se vždy snaží kupovat nejlevější kartu

z nákupního menu, kterou si může dovolit. Jakmile si ji koupí, sníží počet o jedna, případně odstraní položku z menu, když její počet klesne na nulu. Čísla P , V a S indikují blízkost konce hry. Když je počet provincií ve společné zásobě karet menší než P , agent zkusí koupit primárně *Provincie*. Poté se podívá, jestli je počet provincií menší než V . Pokud ano, zkusí koupit *Vévodství*, a když je počet provincií menší než S , bude se snažit koupit *Statek*, a teprve potom se pokusí kupovat karty uvedené v nákupním menu.

Při náhodném generování jedinců je třetí položka fixně (*Zlaták*, 99) a osmá položka (*Stříbrňák*, 10), abych evoluci na začátku pomohl najít správné řešení a snížil nutný počet generací.

Druhá podmínka ukončení hry, tedy vykoupení tří balíčků ze společné zásoby, řešena není, protože se to ve dvouhře zkušených hráčů skoro nestává a protože jsem se snažil snížit prostor možných strategií. Přesto se jedná o možné rozšíření strategie.

3.2.4 Evaluace

Protože evaluovat jedince není jednoduché, používám princip kompetitivní koevoluce. Nákupní strategie jsou rozděleny do dvou množin, množiny vedoucích strategií a množiny ostatních strategií. Množina vedoucích strategií je mnohem menší. Každá strategie hraje proti každé strategii z druhé množiny 50–100 her a zaznamenává si počet vítězství. Strategie jsou seřazeny primárně podle vítězství a sekundárně podle délky nákupního menu (jinak by menu mohlo nekontrolovaně růst, což vede ke značnému zpomalení evaluace). Do další generace vedoucích strategií je zařazeno deset nejlepších strategií. Aby nebyly všechny vedoucí strategie stejné nebo velmi podobné, je nejlepší z nich přijata automaticky. U ostatních se zvyšuje nutná editační vzdálenost od ostatních strategií (počet výměn, přidání a odebrání nutných k tomu, aby byly dva řetězce totožné).

3.2.5 Mutace

Ostatní strategie se vytváří mutacemi vedoucích strategií. Prvních deset je nezmutovaných, ostatní vždy zmutují alespoň jednou a s pravděpodobností 0,5 budou mutovat znovu, takže někteří jedinci zmutují vícekrát. Klasické křížící operátory příliš nedávají smysl, proto je vůbec nepoužívám. Každá mutace má navíc jinou pravděpodobnost vybrání.

- Záměna náhodné karty za jinou kartu v království. (0,04)
- Výměna dvou karet a jejich počtu. (0,3)
- Změna počtu nákupů některé karty o jedna. (0,3)
- Změna čísla indikující nákup vítězných bodů. (0,3)
- Vložení náhodné karty království do nákupního menu. (0,04)
- Odebrání náhodné karty z nákupního menu. (0,02)

Můžeme si všimnout, že přidání a odebrání karty má daleko menší pravděpodobnost než ostatní mutace a že odebrání karty má vyšší pravděpodobnost než přidání. Je to tak nastaveno, protože jedinci bývali příliš dlouzí a karty, které se skutečně nakoupily, byly obvykle jen na prvních deseti místech. Proto jsou upřednostňovány mutace, které jedince zkracují. Čím bude pole kratší, tím se navíc zvýší šance, že se vykoná právě ta užitečná mutace, která jedinci pomůže.

Algorithm 2 Evoluce inteligence specializované na konkrétní karty

```

t = 0
Leaders0 = 10 náhodně vygenerovaných jedinců
Pool0 = 50 náhodně vygenerovaných jedinců
while t < tmax do
  Leaderst+1 = Poolt.Vyhodnot().VezmiNejlepší(5)
  for all i in 0..Poolt+1.Count do
    rodic = Vyselektuj((Leaderst+1))
    Poolt+1[i] = operatorMutace(rodic)
    while random < pravděpodobnostMutace do
      Poolt+1[i] = mutuj(Poolt+1[i])
    end while
  end for
  t = t + 1
end while

```

3.2.6 Parametry evoluce

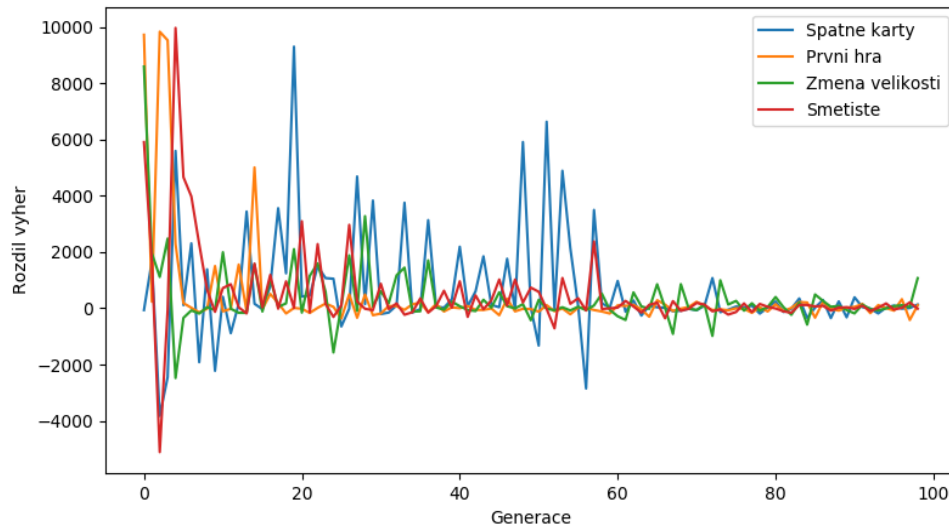
Celá evoluce má několik parametrů. Mezi jedinci je vždy sehráno 50–100 her v závislosti na rozdílu ve výsledcích. Není totiž nutné hrát více her, když je jeden jedinec výrazně lepší než druhý. Další parametry jsou velikost populace vedoucích a ostatních, počet generací a pravděpodobnost další mutace. Celkem se mi osvědčilo nastavení, kdy populace vedoucích strategií má velikost 10, populace ostatních 50, pravděpodobnost další mutace je 0,5 a počet generací 50. Přestože byl u některých království znát mírný vývoj i po 50. generaci, delší počítání by už trvalo příliš dlouho a cílem práce je tento čas zkrátit.

3.2.7 Výsledky evoluce

Průběh evoluce a princip vytváření nákupní strategie budu demonstrovat na následujících čtyřech královstvích. Kromě jednoho se jedná o sady karet doporučené autory hry. Přesné znění karet se můžete dočíst v příloze A.

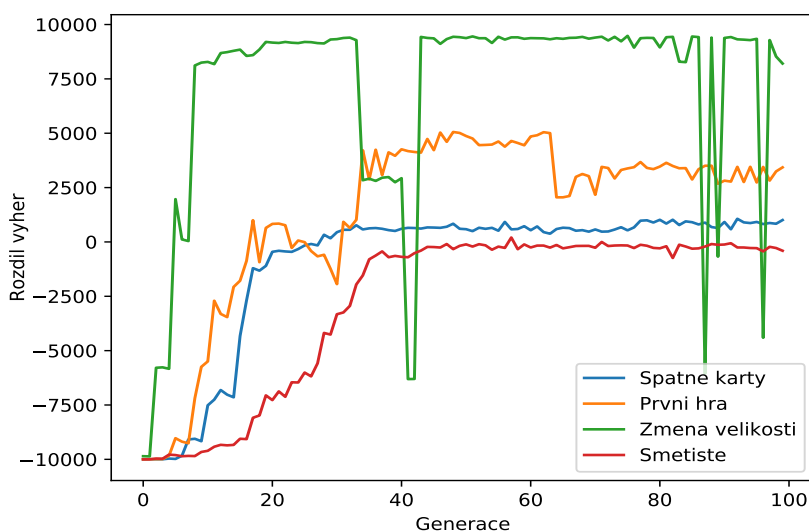
- *První hra: Dílna, Dřevorubec, Důl, Hradní příkop, Kovárna, Milice, Přestavba, Sklepení, Trh, Vesnice.*
- *Změna velikosti: Čarodějnice, Dílna, Dřevorubec, Hostina, Kaple, Laboratoř, Sklepení, Vesnice, Zahrady, Zloděj.*
- *Smetišťe: Dílna, Dřevorubec, Důl, Hostina, Jarmark, Kaple, Lichvář, Přestavba, Trh, Vesnice.*

- *Špatné karty: Dílna, Dřevorubec, Hostina, Hradní příkop, Kancléř, Knihovna, Přestavba, Trůnní sál, Úředník, Zloděj.* (Toto království nepatří do doporučených her a je vybráno, protože karty, které obsahuje, bývají považovány za méně užitečné.)



Obrázek 3.1: Rozdíl počtu výher nejlepšího jedince této generace proti nejlepšímu jedinci předchozí generace z celkového počtu 10 000 her.

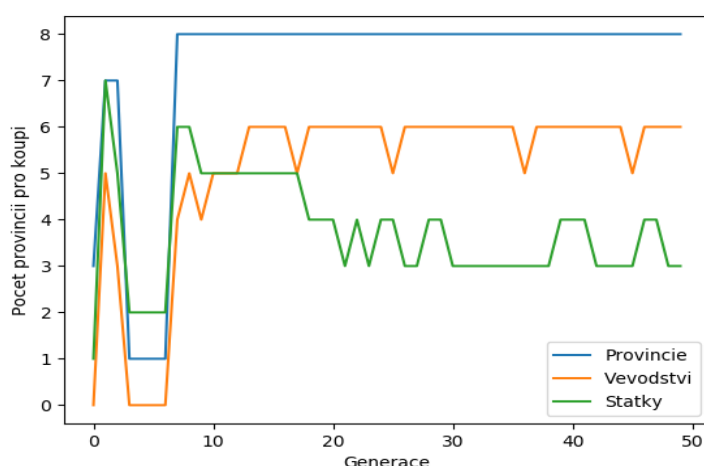
Graf 3.1 popisuje evoluci těchto čtyř království. Osa X značí generaci a na ose Y je zaznačen rozdíl počtu výher nejlepšího jedince této generace a nejlepšího jedince generace předchozí. Nejlepší jedinec této a předchozí generace spolu hrají 10 000 her, takže rozdíly výher v řádu malých stovek jsou zanedbatelné. Evoluce ukazuje 100 generací, aby bylo hezky vidět, kdy se vývoj zastaví. *První hra* je zřejmě jednodušší a její vývoj se značně zpomalí už kolem 30 generace. Přibližně do 50. generace se téměř zastaví vývoj u většiny království. Hra *Špatné karty* je specifická v tom, že žádná z karet příliš nepomáhá k vítězství a můžeme pozorovat výrazný vývoj i po 40. generaci.



Obrázek 3.2: Nejlepší jedinec generace proti již vyvinutému jedinci

V grafu 3.2, zobrazujícím rozdíl výher nejlepšího jedince generace a již vyvinuté strategie, můžeme pozorovat, že i při jiném běhu se evoluce zpomalí kolem 30. generace. Zajímavé je porovnání inteligencí pro hru *Změna velikosti*. Toto království obsahuje několik interaktivních karet, jako je třeba *Čarodějnice*, která hodně ovlivňuje balíček soupeře. Takováto situace velmi snadno vede k několika typům vzájemně se porážejících strategií. Vysoké výkyvy v grafu mohou být důsledkem změny typu strategie v čele.

Strategie *Big Money*



Obrázek 3.3: Vývoj čísel signalizujících přednostní nakupování bodů před penězi pro *Big Money*.

Mnoho začátečníků ve hře Dominion, zvláště když hrají jen se základní hrou, jsou toho názoru, že Dominion je vyřešená hra a stačí si na začátku nakoupit

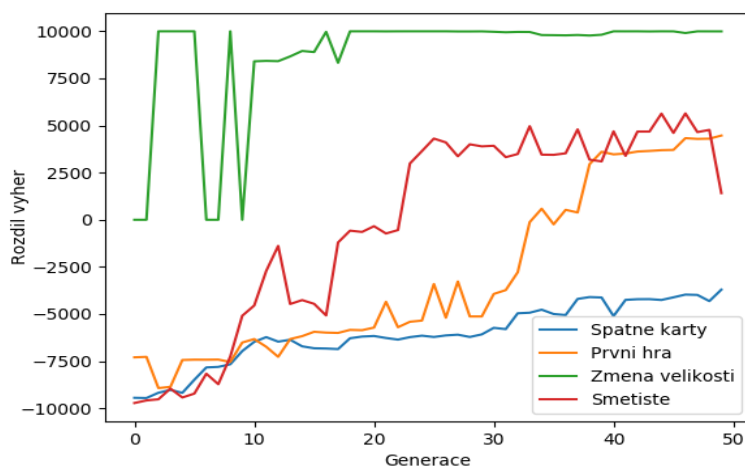
Stříbrňáky, později *Zlatáky* a nakonec vítězné body, a že akční karty jsou zbytečné. Takové strategii se říká strategie *Big Money*.

V grafu 3.3 vidíme vývoj jedince strategie *Big Money*, který kupuje *Zlatáky*, *Stříbrňáky* nebo *Provincie* podle toho, kolik má zrovna peněz. Později, když je ve hře méně *Provincií*, nakupuje také *Vévodství* a *Statky*. Z grafu lze vypožorovat, že když je ve společné zásobě už jen pět *Provincií*, vyplatí se místo dalšího *Zlatáku* koupit raději *Vévodství* a pokud zbývají jen tři *Provincie*, vyplatí se místo *Stříbrňáku* koupit *Statek*.

Graf 3.4 porovnává nejlepšího jedince v generaci se strategií *Big Money*. U království *První hra* a *Smetiště* pozorujeme pozvolný vývoj, protože tyto hry nejsou moc interaktivní.

U království *Změna velikosti* můžeme pozorovat podobné výkyvy jako v předchozím grafu, které je pravděpodobně způsobeno kartou *Čarodějnice*. Strategie *Big money* se žádným způsobem nebrání útokům a nezbavuje se *Kleteb*, takže jich prakticky pokaždé bude mít všech 10. Toto království obecně nenahřává podobným typům strategií.

Oproti tomu království *Špatné karty* je poskládáno tak, aby skoro žádná karta nebyla moc užitečná, a přestože se dá najít strategie, která striktní *Big Money* porazí, bude na ní z velké části založena a nebude výrazně lepší.



Obrázek 3.4: Graf porovnává vývoj evoluce se strategií *Big Money*.

3.3 Podmnožiny

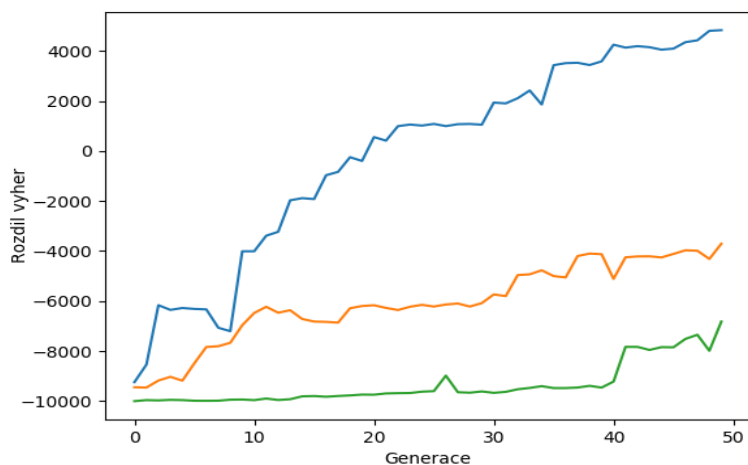
3.3.1 Spolupracující karty

Přestože strategie, kdy se inteligence učí zvláště na každém království, funguje hezky a i pro zkušenějšího hráče je výzva ji porazit, má tento přístup několik vad.

Zprv, čas čekání před každou hrou často přesahuje pět minut, což je na dvouminutovou hru opravdu hodně.

Zadruhé, evoluce se často zasekne v lokálním optimu. Velice zřejmé je to v grafu 3.5, porovnávajícím různé běhy evoluce na království *Špatné karty*. Tento

problém by se dal řešit pomocí nezávislých evolucí a výsledná strategie by se vybrala jako nejlepší z nich, ale tento přístup by ještě více natáhl dobu počítání.



Obrázek 3.5: Porovnání různých běhů evoluce pro království *Špatné karty*.

Zatřetí, pravděpodobně se nejedná o optimální řešení. Prohledávaný prostor je i tak v každé hře velký a přestože existují 3 miliony království (v základní sadě), různých strategií může být třeba jen sto. To je způsobeno tím, že zkušený hráč málo kdy používá všechny karty království, zřídka kdy jich používá více než pět. Problém spočívá v tom, že některé karty spolu dobře spolupracují, zatímco jiné si překážejí. Tato stránka [8], vytvořená fanoušky hry, uvádí pro většinu karet tipy na některé spolupracující a konfliktní karty, které mohou hráči používat. Například karta *Sklepení* dobře spolupracuje s kartou *Laboratoř*, která efektivně zvyšuje počet karet v ruce, a škodí jí, když oponent zahraje kartu *Milice*.

Také existují karty, které se hodí spíše do začátku hry (*Lichvář*, *Důl*) nebo karty pomáhající před koncem (*Dobrodruh*).

Proto nemusí být vždy nutné vyvíjet speciální inteligenci pro každé království, místo toho můžeme mít předgenerovanou sadu inteligencí a z nich teprve vybrat tu nejideálnější.

3.3.2 Pětice

Protože zkušený hráč málokdy potřebuje více než pět různých karet, zkusím vygenerovat všechny možné pětice. Počet všech pětic v základní sadě je jen 53130, což lze v reálném čase spočítat. Před každou hrou se zahraje turnaj, který vybere nejvhodnější inteligenci naučenou na některé z pětic. Počet různých pětic v deseti kartách je 252 a chytře vymyšlený turnaj by mohl nejlepší inteligenci vybrat do deseti sekund.

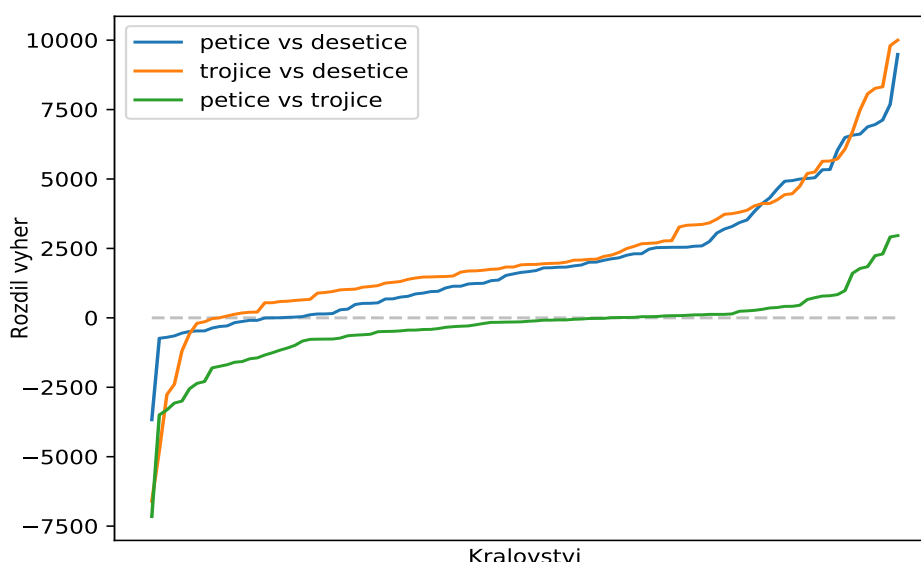
Další výhodou tohoto přístupu je daleko menší prostor, který se musí prohledávat v každé evoluci. Ideální mutace mají vyšší šanci na vykonání v každé generaci a vývoj může být značně rychlejší. Velkou výhodou je i několik stovek nezávislých běhů evoluce pro každou výslednou hru, protože pokud se některý z nich zasekne v lokálním optimu, na výsledku to nebude znát.

V práci budu používat výraz *desetice* pro inteligenci, jež se učila na všech deseti kartách a *trojice*, případně *pětice*, pro inteligence, které měly k dispozici jen tři nebo pět karet.

3.3.3 Trojice

Nevýhoda přístupu pětice je relativně omezená rozšiřitelnost. Počet pětice velice rychle roste s přidáváním dalších karet království. Proto byly vygenerovány také všechny možné trojice. V základní sadě existuje pouze 2 300 různých trojic a když se přidávají karty království, neroste jejich počet tak rychle.

3.3.4 Výsledky



Obrázek 3.6: Rozdíl počtu výher pro 100 různých království.

Tabulka 3.1: Porovnání desetice, pětice a trojice

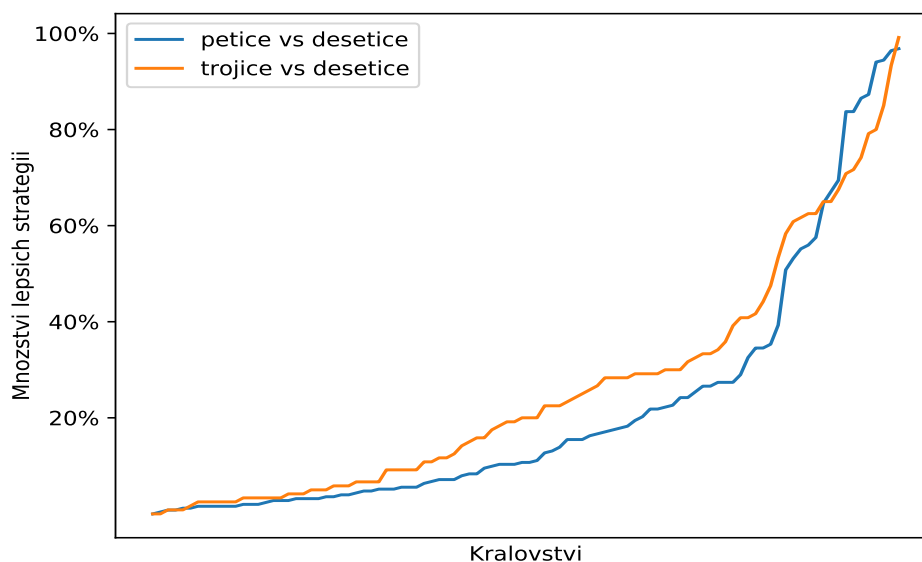
počet výher	průměr	maximum	minimum	rozptyl	vyhraná království v %
pětice vs desetice	2 027	9 488	-3 671	5.2M	86%
trojice vs desetice	2 288	10 000	-6 608	6.7M	91%
pětice vs trojice	-340	2 962	-7 157	1.7M	39%

Graf 3.6 porovnává nejlepší pětice a trojice pro dané království, které jsou vybírány turnajem, a specializovanou desetici vygenerovanou podle všech karet království. Inteligence spolu hrají pokaždé 10 000 her. Království jsou seřazeny podle úspěšnosti pro každou křivku nezávisle na sobě, aby byl graf lépe čitelný. Tabulka 3.1 pouze přesněji vyčísluje některá data z grafu.

Z tohoto grafu a tabulky můžeme vypočítat, že inteligence, která se učí na vhodných pěti nebo i dokonce jen třech kartách, je značně úspěšnější než inteligence, která má k dispozici všech deset karet. Nejlepší inteligence trojic je lepší než desetice dokonce pro 91% vyzkoušených království. Strategie pětic a trojic jsou navzájem dost vyrovnané, ale přesto můžeme pozorovat vyšší úspěšnost trojic.

Přesto existují království, kde se daří lépe deseticím. Tento jev může být způsoben vyšší komplexitou království, kdy inteligence potřebuje používat více než pět karet. Toto vysvětlení však nevidím jako pravděpodobné. Při zkoumání konkrétních strategií nejúspěšnějších desetic jsem totiž zjistil, že žádná z nich nepoužívala více než pět různých karet.

Druhé vysvětlení už bylo zmíněno v předchozí sekci, kdy pro některá království může existovat několik vzájemně se porážejících strategií. Ta, která dopadla v turnaji nejlépe, mohla být zrovna ta, která je nefunkční proti té, která se vyvíjela jako desetice.



Obrázek 3.7: Poměr pětic a trojic, které na svém království porazí desetici, ku všem možným pěticím a trojicím.

Graf 3.7 zobrazuje, kolik inteligencí generovaných jen na pěti nebo třech kartách dokáže porazit inteligenci, která měla k dispozici všechny karty. Vyzkoušel jsem 100 náhodných království a výsledky pro lepší čitelnost grafu seřídil.

V grafu pozorujeme velmi malou skupinu království, pro které nebyla desetice skoro vůbec poražena. Velikost této skupiny přibližně odpovídá stejné skupině v grafu 3.6, kde byla desetice skutečně lepší. Na opačné straně grafu lze zase vidět, že desetici poráží skoro všechny možné trojice a pělice. Tento jev by mohl být vysvětlen tím, že se vývoj desetic na tomto království zasekl v lokálním optimu, což se tolik nestává u pětic a trojic, protože prohledávají menší prostor. Na náhodném království je průměrně 22% pětic a 26% trojic lepsích než specializovaná desetice. Tento jev může být zohledněn při rozšiřování hry.

3.3.5 Hodnocení karet

Tabulka 3.2: Užitečnost karet

Trojice	Využití	Pětice	Využití
Čarodějnice \$5	214/216	Čarodějnice \$5	205/206
Zahrady \$4	139/206	Zahrady \$4	154/201
Zasedání rady \$5	130/217	Laboratoř \$5	126/198
Kaple \$2	106/208	Dobrodruh \$6	125/198
Důl \$5	79/186	Úředník \$4	119/193
Knihovna \$5	66/205	Důl \$5	122/203
Jarmark \$5	70/222	Zasedání rady \$5	121/204
Laboratoř \$5	63/202	Jarmark \$5	109/185
Lichvář \$4	64/210	Kaple \$2	120/206
Kovárna \$4	54/183	Trh \$5	116/206
Trh \$5	56/196	Knihovna \$5	99/188
Milice \$4	50/202	Přestavba \$4	94/199
Dílna \$3	44/206	Špeh \$4	88/191
Sklepení \$2	40/194	Milice \$4	94/213
Špeh \$4	44/224	Dřevorubec \$3	88/208
Dobrodruh \$6	34/178	Kovárna \$4	83/203
Vesnice \$3	35/186	Sklepení \$2	84/209
Zloděj \$4	34/181	Kancléř \$3	74/195
Hostina \$4	34/190	Hostina \$4	78/217
Kancléř \$3	32/193	Hradní příkop \$2	65/184
Hradní příkop \$2	28/193	Dílna \$3	68/196
Úředník \$4	25/187	Lichvář \$4	68/196
Trůnní sál \$4	26/204	Zloděj \$4	68/199
Přestavba \$4	20/213	Vesnice \$3	68/205
Dřevorubec \$3	13/198	Trůnní sál \$4	64/197

Skutečnosti, že máme vygenerované pětice a trojice, lze využít například k analýze celkové užitečnosti karet. Pětice a trojice, které vyhrávají, používají pravděpodobně nejužitečnější karty.

Pro tabulku 3.2 jsem náhodně vylosoval 500 království. Tabulka pro každou kartu ukazuje, kolikrát byla vybrána strategie petic nebo trojic, která se na ní učila, a kolikrát byla celkově ve hře.

Můžeme pozorovat, že karty s cenou \$3 a \$4 jsou vybírány méně často než karty s cenou \$2 a \$5. Tento jev je pravděpodobně podpořen fixní počáteční konfigurací hry, kdy mají hráči na začátku vždy \$5 a \$2 peníze. Částečně na to však může mít také vliv, že karty s cenou \$5 bývají opravdu silné. Karta *Čarodějnice* je asi nejsilnější karta ve hře, protože opravdu značně zpomaluje progres protihráče. Karta *Zahrady* je levnější alternativou karty *Vévodství* a velmi často za ni hráč dostane stejný počet vítězných bodů. Překvapivý mi přijde fakt, že karta *Trůnní sál* je hodnocena jako nejhorší. Může to být způsobeno počáteční konfigurací nebo nevhodnou hrací strategií pro tuto kartu.

3.3.6 Hodnocení karetních kombinací

Následující tabulky ukazují výsledky turnaje, který funguje na stejném principu jako turnaj vybírání nejlepší pětičky či trojice pro dané království, ale do království jsou vybrány všechny karty a v turnaji soutěží všech 2 300 trojic.

Tabulka 3.3: Finální kolo turnaje všech trojic

Karty	Počet výher
Zahrady, Kaple, Čarodějnice	1 288
Zahrady, Zloděj, Čarodějnice	1 245
Zahrady, Lichvář, Čarodějnice	1 190
Hostina, Zahrady, Čarodějnice	1 175
Zahrady, Kancléř, Čarodějnice	1 169
Kaple, Trh, Čarodějnice	1 140
Zahrady, Kovárna, Čarodějnice	1 136
Zahrady, Špeh, Čarodějnice	1 124
Kaple, Zloděj, Čarodějnice	1 087
Zahrady, Čarodějnice, Dřevorubec	1 083
Kancléř, Kaple, Čarodějnice	935
Kaple, Přestavba, Čarodějnice	929

Tabulka 3.3 ukazuje výsledky z finálního kola turnaje všech trojic, kde každá strategie s každou hrála 100 her. V turnaji naprosto dominuje karta *Čarodějnice*, do posledního kola se dostaly jediné strategie, které ji používají. Tuto kartu ještě vždy doplňuje alespoň jedna z karet *Kaple* nebo *Zahrady*. A skutečně lze nahlédnout, že karty *Čarodějnice* a *Zahrady* se nepřímo podporují, protože *Čarodějnice* zvyšuje velikost balíčku soupeře, zatím co *Zahrady* zhodnocují velikost balíčku vítěznými body. Karta *Kaple* sice nefunguje hezky se *Zahradami*, ale jedná se o jediný efektivní způsob obrany proti *Čarodějnice*. Můžeme tedy bezpečně prohlásit, že tato karetní kombinace je nejsilnější ve hře.

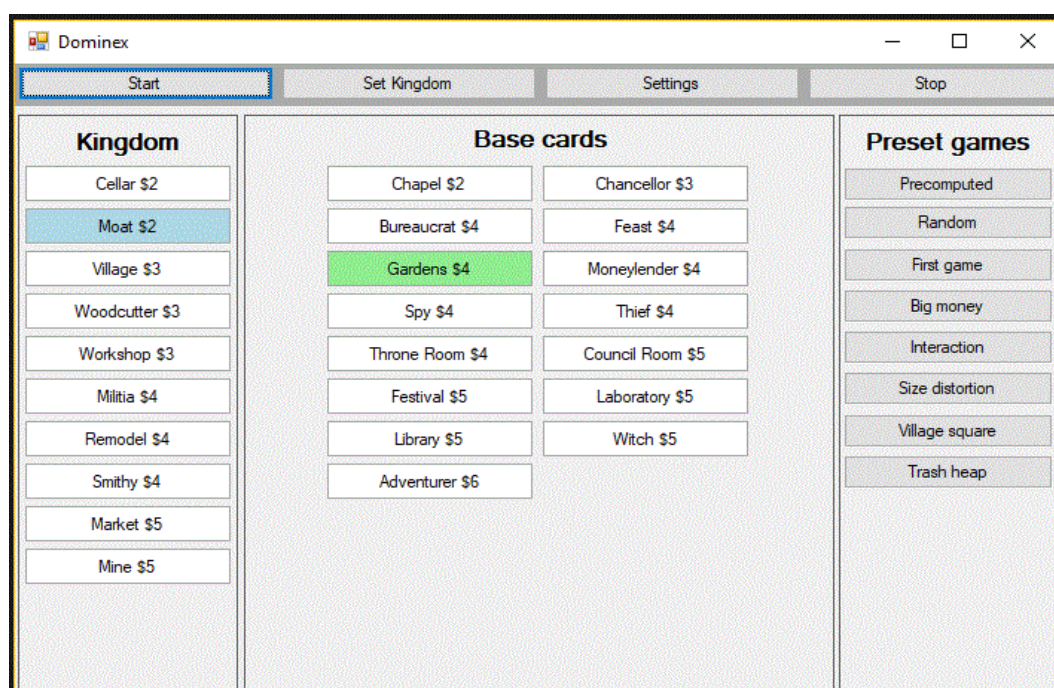
4. Uživatelská dokumentace

Tato kapitola se zabývá uživatelským rozhraním. Popíšeme si možnosti nastavení hry, princip vybírání karet a výběr protihráče. Hru lze spustit pomocí aplikace `\Dominion\Window\bin\Release\Window.exe`

V záhlaví okna jsou čtyři tlačítka. Tlačítko *Start* slouží pro spuštění hry. Tlačítko *Set Kingdom* zobrazí stránku pro vybírání karet království. Tlačítko *Settings* ukáže další nastavení, které nesouvisí s vybíráním karet. Tlačítko *Stop* zastaví rozehranou hru.

4.1 Nastavení

4.1.1 Volba království



Obrázek 4.1: Ukázka uživatelského rozhraní.

Při spuštění aplikace se zobrazí stránka 4.1, která slouží pro vybírání karet království. Tato stránka se také zobrazí po kliknutí na tlačítko *Set Kingdom* v záhlaví. Stránka je rozdělena na tři části, z nichž každá obsahuje skupinu tlačítek.

Tlačítka vlevo označují karty, s nimiž se bude hrát, zatímco tlačítka uprostřed všechny ostatní karty, které nejsou vybrány do království. Kliknutím na tlačítko způsobíme, že se karta přesune do druhé množiny karet.

Je možné zvolit království, které nebude mít deset karet, avšak nelze jich mít více než dvanáct. Jestliže je zvoleno království s nízkým počtem karet, je nutno počítat s možností, že nebude existovat žádná vhodná strategie.

Tlačítka vpravo slouží pro rychlé nastavení her. Tlačítko *Random* vylosuje náhodné království a tlačítko *Precomputed* vybere takové království, pro které

existuje předpočítaná strategie na deseti kartách. Ostatní tlačítka v této skupině vyberou některé z pojmenovaných království.

4.1.2 Nastavení protihráče a jména

Po kliknutí na tlačítko *Settings* v záhlaví okna se ukáží tři možnosti nastavení umělé inteligence. Při vybrání možnosti *Tens* se bude hledat strategie přímo specializovaná pro dané království. Taková strategie ovšem nemusí být vygenerovaná.

Možnost *Fives* před každou hrou najde všechny podmnožiny království velikosti pět a pro ně specializované strategie, potom pomocí turnaje vybere nejlepší z nich, což by nemělo trvat déle než deset sekund. Tento protihráč je o poznání obtížnější a je zaručeno, že nějaký bude vybrán.

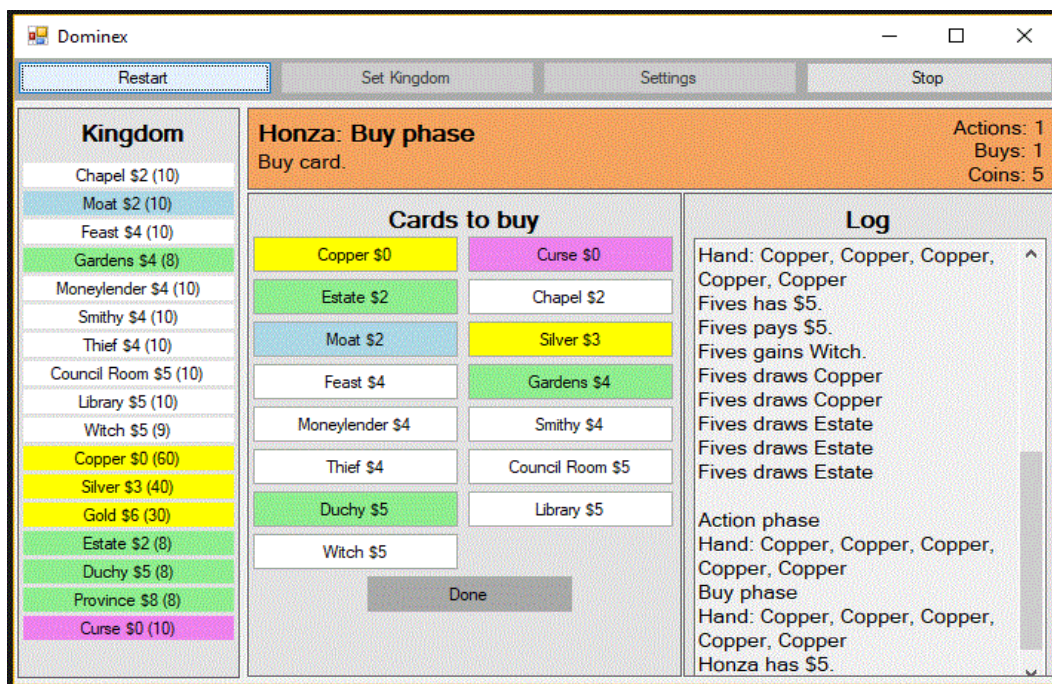
Poslední možnost *Threes* funguje podobně jako předchozí situace s tím rozdílem, že tentokrát se budou hledat podmnožiny velikosti tři. Obtížnost protihráče je srovnatelná s předchozí možností.

Jméno hráče lze změnit přepsáním stávajícího jména, které je uvedeno v kolonce uprostřed stránky.

4.1.3 Start hry

Hra se spustí kliknutím na tlačítko *Start*. Pokud je vybrána strategie typu *Tens*, zkontroluje se, jestli je taková strategie vygenerovaná. Když tomu tak není, vyskočí okno informující o této situaci a okno se vrátí na stránku s nastavením království. Pokud chce hráč vybrat království, pro které strategie existuje, má možnost zvolit některé z pojmenovaných království nebo vybrat náhodné pomocí tlačítka *Precomputed*.

4.2 Průběh hry



Obrázek 4.2: Ukázka uživatelského rozhraní.

Herní okno je rozděleno na pět částí, jak můžeme vidět na obrázku 4.2. V záhlaví jsou stále tlačítka popsaná v minulé sekci, ovšem možnosti *Set Kingdom* a *Settings* jsou deaktivovány. Tlačítko *Start* změní text na *Restart* a umožní hráči začít stejnou hru od začátku. Tlačítko *Stop* hru ukončí a zobrazí úvodní stránku s výběrem karet.

Na levé straně lze vidět balíčky společné zásoby. Pro každý balíček je vypsáno jméno karty, její cena a počet karet v balíčku.

V horní části obrazovky se nachází informační box. V něm je uvedeno jméno hráče, herní fáze s popisem rozhodnutí a informace o počtu akcí, nákupů a peněz, které má hráč k dispozici pro toto kolo.

Na pravé straně se zobrazuje herní záznam. Každá akce a událost je zde zaznamenána, aby měl hráč kompletní přehled o herním dění.

4.2.1 Popis rozhodování

Uprostřed můžeme nalézt akční okno. Veškerá rozhodnutí budou vykonávána prostřednictvím tohoto okna. Samotná rozhodnutí lze rozdělit do několika kategorií.

- **Zahrání karty:** V akční fázi se zde nachází tlačítko pro každou kartu. Při kliknutí na tlačítko akční karty se tato karta zahráje. Tlačítka ostatních karet jsou deaktivována. Jestliže zahraná karta nevyžaduje další rozhodování, může hráč zahrát další kartu, pokud má dostatek akcí, nebo se kolo posune do nákupní fáze. Jinak se v tomto okně zobrazí možnosti rozhodování. Popis rozhodnutí je vždy zobrazen v informační části nahoře.

- **Binární rozhodnutí:** Hráč má jen dvě možnosti volby. Je vyvoláno zahráním karet *Kancléř*, *Knihovna*, *Špeh*, *Úředník* a *Zloděj*.
- **Rozhodnutí vyžadující vybrání skupiny karet:** V takovém případě se u každé karty zobrazí zaškrtačací políčko. Hráč musí zaškrtnout předepsaný počet a kliknout na tlačítko *Done*, které se nachází pod kartami. Pokud je zaškrtnutý špatný počet, vyskočí okno informující o této situaci a hráč bude muset počet upravit, aby byl ve správném intervalu. Karty, které jej způsobují, jsou *Důl*, *Kaple*, *Milice*, *Přestavba*, *Sklepení* a *Zloděj*.
- **Zahrání karty reakce:** Když soupeř zahraje kartu útoku a hráč má na ruce kartu reakce, může ji ukázat a před samotným útokem uplatnit její efekt. Jediná karta reakce v základní sadě je *Hradní příkop* a její efekt je, že se hráč útoku ubrání.
- **Rozhodování mimo vlastní tah hráče:** Některé útočné karty nechávají rozhodnutí o konkrétní podobě útoku na hráči, který se brání. V takovém případě se může stát, že se hráč musí rozhodovat i mimo svůj tah. Mezi tyto karty se řadí *Milice* a *Úředník*.
- **Nákup nebo možnost zisku karty:** Pro každou dostupnou kartu je zde tlačítko. Po kliknutí na některé z nich je určená karta přidána na odkládací balíček. Pokud jde o nákup, je u karet uvedena i jejich cena, přičemž po nákupu je určená částka odečtena z celkové sumy peněz, kterou má hráč k dispozici pro toto kolo. Pokud má hráč více nákupů, pokračuje nákupní fáze. V opačném případě je na tahu protihráč.

Tah protihráče lze sledovat v herním záznamu. Poté následuje opět akční fáze uživatele.

4.2.2 Konec hry

Počet zbývajících karet lze pozorovat v levé části obrazovky. Jakmile je některý balíček vykoupen, změní barvu ze své původní na šedou. Hra končí, když jsou vykoupeny tři balíčky nebo všechny *Provincie*. Když se tak stane, v herním záznamu se ukáží výsledné body, které hráči získali.

5. Implementace

Hra je implementována v jazyce C#. Pro tvorbu uživatelského rozhraní byla zvolena technologie WinForms, především pro jednoduchost a rychlost vývoje. V programu je používán také NuGet Package ValueTuple verze 4.5.0 [9].

Práce je rozdělena do několika projektů. Knihovna *GameCore* obsahuje samotnou implementaci hry, karet a hráče. Obsahuje také abstraktní třídu *User*, která slouží nejen jako rozhraní pro umělou inteligenci, ale také pro skutečného uživatele.

Projekt *AI* je taktéž knihovna tříd a obsahuje třídy spojené s implementací umělé inteligence. Základní dvě třídy jsou *Evolution*, kde je implementována základní kostra evoluce, a *ProvincialAI*, která dědí od třídy *User* a je používána pro evaluaci jedince.

K těmto knihovnám přistupuje několik projektů, které používám pro různé starty programu. Konzolový projekt *Evolution* slouží ke generování specializovaných inteligencí z unixové i windowsové příkazové řádky, abych mohl snadno generovat soubory paralelně na více počítačích.

Projekt *View* je samotná aplikace s hrou. Uživatel si v ní může nastavit konkrétní karty, protihráče a hru si zahrát. V projektu lidského uživatele zastupuje třída *Human*, která dědí od třídy *User*.

Ostatní projekty slouží zpravidla pro testování nebo generování různých dat pro analýzy a nejsou důležité pro ostatní projekty.

5.1 Knihovna GameCore

V této sekci si rozebereme jednotlivé třídy a jejich důležité aspekty.

5.1.1 Game

Aby byla implementace umělé inteligence jednodušší, hra je řízena zevnitř, tedy volá metody na hráčích a nikoli hráči na ní. To na druhou stranu komplikuje implementaci uživatele. Konstruktor třídy vyžaduje seznam uživatelů (třída *User*), království, na kterém se hraje, a volitelně logger a *CancellationTokenSource* (pokud chci mít možnost tuto hru restartovat). Maximální počet kol je defaultně nastaven na padesát, aby hry, kde hrají dvě špatné inteligence, netrvaly příliš dlouho. Tento počet však může být změněn.

Jakmile nastavím parametry hry pomocí konstruktoru, mohu zavolat metodu *Play*, která vrací *Task* s výsledky hry. Celý vnitřek metody pak běží v jiném vláknu, aby uživatelská část aplikace nezamrzla v momentě, kdy hraje protihráč, a také aby bylo možné volat metody hráčů ze hry a nikoli naopak. V této metodě je implementována vlastní logika hry. Dokud neuplynul předepsaný počet kol, prochází postupně hráče a volá na nich příslušné metody, které budou podrobněji rozebrány u třídy *Player*.

5.1.2 Player

Třída se stará o vnitřní data hráčů a o metody, které s nimi pracují. Každý hráč má v sobě potomka třídy *User*. Pokaždé, když musí hráč učinit nějaké rozhodnutí, které závisí na uživateli nebo AI, zavolá metody této třídy.

Metoda *PlayCard* nechá uživatele vybrat kartu z akčních karet, jež má hráč v ruce, pokud nějakou má a pokud má hráč ještě volnou akci. Pak odstraní kartu z ruky, přidá ji do zahraných karet, sníží počet dostupných akcí a na kartě zavolá metodu *WhenPlayAction*, kde je implementována vlastní logika karty. Pokud je to karta útoku, zavolá funkci *DealAttack* na všech ostatních hráčích. Na konci vrátí zahranou kartu nebo null, podle toho hra pozná, jestli má tuto metodu znova volat.

Metoda *PlayTreasure* by měla fungovat podobně. Podle pravidel hry má hráč ve fázi nákupu možnost postupně zahrát některé z peněžních karet z ruky, ale tento způsob implementace výrazně zpomaloval hru. Protože v základní verzi hry není žádná karta, na kterou by to mělo vliv, rozhodl jsem se tedy toto rozhodnutí nenechat na uživateli a peněžní karty jsou automaticky zahrány všechny. Proto je při rozšiřování hry důležité na toto místo myslet a lehce upravit kód.

Ve funkci *Buy* uživatel vybere z dostupných karet tu, kterou si chce koupit. Tato karta je poté umístěna na odkládací balíček. Pokud si uživatel nechce nic koupit, funkce vrátí null.

Třída obsahuje ještě několik dalších triviálních funkcí, jež mají hlavně zamezit duplikování kódu a jsou používány zejména pro implementaci logiky karet.

DealAttack je funkce, která je zavolána, když se hráč brání útoku jiného hráče. V tomto případě může hráč zahrát své reakční karty, a pokud se útoku neubrání, zavolá se efekt útoku.

5.1.3 User

Tato třída slouží jako interface pro AI. Přestože typů rozhodnutí je jen několik, z důvodů rychlosti a přehlednosti jsem se rozhodl pro každou kartu, která vyžaduje nějaké rozhodnutí, vytvořit zvláštní metodu.

5.1.4 Card

Každá karta dědí od této třídy. Instance každého typu karty je za běhu programu vytvořena jen jednou z důvodu šetření paměti a protože karta neobsahuje žádná měnící se data. Při implementaci karty se jednoduché datové položky zadávají do base konstruktoru. Pro speciální efekt karty jsou připraveny virtuální metody *ActionEffect* a *TreasureEffect*. Metoda *CountPoints* slouží pro karty s proměnným počtem vítězných bodů a je zavolána na konci hry.

5.1.5 Pile

Třída představuje balíček karet, obvykle stejných, ale některé balíčky mohou obsahovat i navzájem odlišné karty. Třída se stará o to, aby se karty nekopírovaly, ale jen přesouvaly. Jediným korektním způsobem, jak získat kartu, je metoda *GainCard*.

5.1.6 Kingdom a KingdomWrapper

Třída *Kingdom* je kolekce všech karet, které mohou být získány. Přestože se nazývá královstvím, má v tomto kontextu jiný význam než karty království v pravidlech hry. Je nutné pro každou hru vytvořit novou instanci, aby počet karet v balíčcích byl správný a neovlivněný jinou hrou. Třída *KingdomWrapper* je obalující třída pro třídu *Kingdom* a její funkcí je efektivně určit, jestli je daná karta dostupná pro nákup nebo ne.

5.2 Uživatelské rozhraní

Formulář je implementován třídou *GameForm*. Ovládání hry je konkrétněji popsáno v kapitole 3.3.6. Při kliknutí na tlačítko Start se zavolá metoda *StartButton_Click*, která získá agenty umělé inteligence, vytvoří třídu *Human* a nový *CancellationTokenSource*. Nakonec inicializuje a spustí hru. Pokud uživatel znovu klikne na tlačítko *StartButton_Click*, aktivuje *CancellationTokenSource* a hra bude přerušena.

Uživatele reprezentuje třída *Human*, která dědí od třídy *User*. Když je vyžadováno rozhodnutí uživatele, zamkne *Job*, zavolá delegáta, kterého třída získá v konstruktoru, zavolá *Monitor.Wait* a čeká, než ji kliknutí na určené tlačítko zase spustí. Po spuštění ještě zkontroluje, jestli hra nebyla restartována, a v tom případě hru ukončí.

Delegát je metoda formuláře, která se umí zapsat do událostní smyčky a spustit v hlavním vláknu, aby mohla upravovat komponenty formuláře. Například při vybírání karty k zahrání, dostane seznam hratelných karet, pro každou vytvoří tlačítko, které při stisknutí do třídy *Job* zaznamená výsledek a spustí všechna čekající vlákna (při restartu hry jich může být více). Tímto způsobem je zajištěna komunikace hra-uživatel.

5.3 Utils

Knihovna *Utils* obsahuje dvě malé třídy používané napříč programem, které by mohly být užitečné i jakémkoli jiném projektu.

5.3.1 ThreadSafeRandom

Třída *ThreadSafeRandom* řeší problém náhodnosti v paralelním prostředí. V momentě, kdy hraji více paralelních her, nemohu používat jednu třídu *Random* bez zámků, protože k tomu není uzpůsobena. Protože se používá často, není ani řešení si ji zamykat. Na druhou stranu vytváření třídy *Random* pro každou hru zvláště by mělo za následek, že velká část z nich by měla stejný *Seed* a všechny hry by měly stejný průběh. Třída *ThreadSafeRandom* pro každý paralelní *Task* bezpečně vytvoří vlastní instanci třídy *Random*, kde *Seed* získá z privátní statické instance třídy *Random*, a zámků jsou tudíž potřeba pouze v konstruktoru [10]. Používání jedině této náhodné třídy napříč celým programem má za důsledek také možnost determinizace programu pro potřeby ladění.

5.3.2 Subsets

Třída *Subsets* vytváří kolekci všech podmnožin potřebné velikosti zadané množiny.

5.4 AI dokumentace

Přehled

Knihovna AI obsahuje veškeré komponenty související s implementací umělé inteligence. Knihovnu lze rozčlenit do tří vzájemně provázaných částí, v projektu jsou rozděleny do tří jmenných prostorů.

Evolution obsahuje třídu *Evolution*, která se stará o vlastní běh evoluce, její pomocné třídy a abstraktní třídu *Evaluator*.

Jmenný prostor *Provincial* obsahuje třídu *ProvincialAI*, jež implementuje třídu *ProvincialEvaluator* a rozhraní *User* definované v knihovně *GameCore*.

Jmenný prostor *Model* slouží jako jednoduchá databáze. Obsahuje třídu *BuyAgenda*, v níž jsou uložena samotná data. Jedná se o reprezentaci herní inteligence, jež je evolucí vyvíjena. Třída *ProvincialAI* se pomocí této třídy rozhoduje, které karty bude nakupovat.

5.4.1 Evolution

Třída *Evolution* se stará o samotný průběh evoluce. Generace jedinců, reprezentovaných třídou *BuyAgenda*, je vyhodnocena (evaluována), tedy každému jedinci je přiřazena tzv. „fitness“. Nejlepší jedinci jsou vybráni pro vytvoření další generace.

V konstruktoru dostane třída *Evolution* parametry evoluce a volitelně referenčního jedince. Pokud je referenční jedinec uveden, je s ním po každé evaluaci porovnáván nejlepší jedinec generace. Výsledek porovnání je poté zapsán do *Loggeru* pro zachycení progresu evoluce.

Evaluace

Samotná evaluace probíhá tak, že každý jedinec z populace je konfrontován s každým jedincem v množině vedoucích strategií, tzv. „leaderů“. Jedinci jsou evaluováni paralelně pomocí metody *Evaluator.Evaluate*, která je sama také paralelní. Jedinec a leader spolu hrají až sto her v závislosti na rozdílu výkonnosti. Výsledná „fitness“ je rozdíl všech výher a proher v poměru k počtu hraných her.

Vybírání nových leaderů

V metodě *SetNewLeaders* se do vedoucích strategií vyberou nejlepší strategie v populaci, pokud splňují následující nerovnici.

$$\sum_{leader \in leaders} d(leader, jedinec) < 1.5 |leaders| - 1$$

Množinou *Leaders* je myšlena množina nových leaderů, která je na začátku prázdná a do níž jsou strategie teprve přidávány. Žádný nový leader tedy nesmí být příliš podobný už určeným leaderům pro další generaci.

Vzdálenost d je editační vzdálenost, známá také jako Levenschteinova vzdálenost, která je definována jako počet přidání, smazání nebo výměn znaku, aby byl jeden řetězec karet změněn na druhý [11]. U jedinců takto porovnávám jejich nákupní menu, přičemž беру v potaz jen názvy karet, nikoli celou dvojici (*název karty, počet koupí*). Vzdálenost lze efektivně spočítat pomocí dynamického programování.

Toto zaručí, že vedoucí strategie budou dostatečně rozmanité, což podpoří exploraci a sníží pravděpodobnost zaseknutí evoluce v lokálním optimu.

Nová generace

Nová generace je vytvářena v metodě *GenerateNewPool*. Základ nové generace tvoří kopie vedoucích strategií. Celá generace se pak doplní tak, že metoda vybere náhodného jedince z leaderů, naklonuje si ho – mutace totiž mění jedince – a alespoň jednou ho zmutuje. Další mutace jsou přidávány vždy s pravděpodobností danou v parametrech evoluce.

Parametry

Třída *Params* je obálkou pro parametry evoluce. Evoluce je parametrizována královstvím, třídou *Evaluator* (v programu používám *ProvincialEvaluator*), minimálním a maximálním počtem her, počtem vedoucích strategií, velikostí generace, pravděpodobností další mutace, stupněm paralelismu a třídou *MutationSelector*. Tato třída vrací náhodnou mutaci, když je potřeba mutovat jedince.

Od abstraktní třídy *Mutation* dědí vlastní mutace, které byly již detailněji popsány dříve 3.2.4. Mutace nevytváří nového jedince, nýbrž mění inteligenci, kterou dostane v parametru.

5.4.2 Provincial

Třída *ProvincialAI* implementuje rozhraní *User* a umí podle dat ve třídě *BuyAgenda* hrát hru Dominion.

Herní strategie

Herní strategie zahrnuje veškerá rozhodnutí hráče, jež nesouvisí se získáním karty. Jedná se tedy hlavně o vybírání karty, která bude zahrána, a o druhotná rozhodnutí vyvolaná zahráním některé karty. Když se kupříkladu rozhodnu, že zahraji *Kapli*, musím pak ještě určit, které karty chci zahodit na smetišť.

Vybírání nejvhodnější karty se odehrává v metodě *PlayCard*. Karty jsou seřazeny podle priority, toto řazení lze nalézt v příloze A. Každá karta je pak ohodnocena na základě priority, ostatních karet v ruce a počtu akcí – viz pseudokód 3. Karta s nejvyšším hodnocením je pak vybrána. Její zahrání má obvykle za následek změnu okolností a při další volbě může hodnocení dopadnout úplně jinak.

Útok kartou *Milice* je speciální, protože jsou hodnoceny i jiné než akční karty. Hráč musí totiž odložit určený počet karet, a tak se vybírají ty s nejmenším hodnocením. Karty peněz mají v tomto případě nejvyšší hodnocení a karty bodů nejmenší.

Algorithm 3 Hodnocení karty

```
výsledek = priorita karty * 2
if počet akcí = 0 & karta přidává akce then
    výsledek += výsledek + 100
end if
if Kaple then
    výsledek += počet kleteb v ruce * 3
else if Knihovna then
    výsledek += (7 - počet karet) * 3
else if Přestavba then
    výsledek += 3, pokud má hráč v ruce Kletbu
else if Lichvář then
    výsledek = -1, pokud hráč nemá v ruce Měďák
else if Důl then
    výsledek = -1, pokud hráč nemá v ruce Měďák ani Stříbrňák
end if
vrátit výsledek
```

Rozhodnutí, jež jsou vyvolána zahráním karty Poté, co je karta zahrána, není již ve většině případů potřeba žádné další rozhodování. Existují však karty, které jsou v tomto ohledu výjimkou, a právě ty si na tomto místě popíšeme.

- *Kancléř*: Nikdy neodkládá balíček.
- *Kaple*: Zahodí všechny kletby. Pokud ještě nechce kupovat *Statky*, tak je zahodí také. Podívá se, kolik stojí karta, kterou si chce koupit, a zahodí nadbytečné *Měďáky*.
- *Knihovna*: Pokud hráč nemá žádné další akce, odkládá všechny akční karty stranou.
- *Přestavba*: Zahodí kartu podobně podle stejného principu jako u *Kaple*, ale pokud není karta vybrána tímto způsobem, vybere kartu s nejhorším hodnocením.
- *Sklepení*: Odkládá jen karty vítězných bodů.
- *Špeh*: Při útoku nechává hráči vítězné body a *Měďák* v dobíracím balíčku, při obraně je odkládá.
- *Tránní sál*: Vybere kartu s nejvyšším hodnocením.
- *Úředník*: Při útoku vrací náhodnou kartu vítězných bodů, kterou má v ruce, na dobírací balíček.
- *Zloděj*: Zloděj vždy vybere nejdražší kartu. Pokud to bude *Měďák*, zahodí jej, jinak si kartu nechá.

Herní strategie je často implementovaná přímočaře a existuje několik situací, které nemusí řešit optimálně, ale stálo by hodně úsilí všechny tyto situace řešit

zvláště. Například když má hráč na ruce *Špeha* a *Zloděje*, vyplatilo by se zahrát nejprve *Špeha*, a pokud najde *Zlaták*, nechá jej tam a pak ukradne *Zlodějem*. Místo toho se inteligence rozhodne *Zlaták* pokaždé odložit.

Také určení priorit nemusí být optimální. Zajímavý přístup by mohla být evoluce priorit. Další problém s touto heuristikou je, že velká část se bude muset měnit, pokud se do hry přidají karty z rozšíření. Nechám tedy tyto možnosti na případné dodatky k aplikaci.

5.4.3 Model

Základní datonosnou třídou celého programu je třída *BuyAgenda*, která se umí naklonovat, aby se snadno vytvářeli jedinci do další generace a aby hraní hry nepoškozovalo data, která obsahuje. Během hraní se totiž musí snižovat čísla určující, kolik karet daného typu se má ještě koupit. Třída také obsahuje konverzní metody, převádějící data na textový řetězec a naopak.

BuyAgendaManager

Třída *BuyAgendaManager* se stará o správné ukládání serializované třídy *BuyAgenda* do souborů. Snažil jsem se trochu zmenšit velikost souborů, ale zároveň nemít na každou agendu zvláštní soubor. Z tohoto důvodu jsou agendy ukládány vždy do souboru označeného kartou království, na které jsou specializované, s nejnižším identifikačním číslem. Agendy nicméně nejsou rozmístěny rovnoměrně, například asi polovina desetic je v jednom souboru.

Od třídy *BuyAgendaManager* dědí dvě konkrétní třídy. *SimpleManager* jednoduše načítá a ukládá specifikovanou agendu, oproti tomu *CachedManager* při načítání načte celý soubor a uloží si jej do paměti. Tento princip je využíván při turnaji, který bude popsán později a v němž je potřeba načíst všechny pětic, jež jsou obsaženy v dané desetici. Přesně polovina pětic se totiž nachází ve stejném souboru a turnajový čas je potřeba co nejvíce zkrátit, každý soubor je tedy přečten pouze jednou.

Turnaj podmnožinových inteligencí

Ve třídě *CachedManager* je metoda *LoadBest*, která dostane království, najde všechny karetní podmnožiny a načte jejich strategie. Poté vybere nejlepší strategii. Kdyby hrála každá strategie proti každé dostatečný počet her, výběr by trval příliš dlouho. Strategie jsou tedy rozdělny do skupin po 16 a z každé skupiny postupují právě čtyři strategie. Druhé kolo probíhá obdobně. Třetí kolo už obsahuje pouze 16 strategií a do finálního kola postupuje pět nejlepších. Turnaj je také koncipován tímto způsobem, aby slabší strategie tolik nezkreslovaly výsledky finálního turnaje.

Závěr

Zhodnocení výsledků

Cílem práce bylo implementovat hru Dominion a prozkoumat možnosti umělé inteligence pro tuto hru. Hra se vyznačuje obrovskou variabilitou počátečních konfigurací, protože možných karet království je v základní hře 25, zatímco každé hry se účastní jen 10 z nich.

Rozhodl jsem se pro implementaci pomocí evolučních algoritmů, protože poskytovaly možnost flexibilní adaptace na různé počáteční podmínky, hezkého náhledu do principu fungování strategie a slibné možnosti rozšíření. Úkolem bylo zejména redukovat čas, který musí hráč čekat před každou hrou.

Přístup, kdy se inteligence učí před každou hrou na všech deseti kartách, byl úspěšný, co do kvality protihráče, ale doba učení značně přesahovala pět minut.

Přestože počet různých království (myšleno kombinací 10 karet) je 3 268 760, fungujících strategií je o hodně méně. Z pozorování chování takové inteligence ve hře lze nahlédnout, že málokdy potřebuje ke hře více než pět karet království a ostatní karty, které jsou na stole, ignoruje. Vyvinul jsem tedy strategii pro každou pěticí a trojici karet, protože kombinací pětic je jen 53 130 a kombinací trojic jen 2 300. Každé království má 252 různých kombinací pětic a 120 různých kombinací trojic a z těchto připravených strategií vybírám před každou hrou tu nejlepší pomocí turnaje.

Výhodou tohoto přístupu je, že při každé evoluci je prohledáván značně menší prostor, inteligence je tudíž vyvíjena rychleji a je menší pravděpodobnost, že se zasekne v lokálním optimu. Pokud se tak přeci jen stane, v množině přípustných strategií pro dané království se skoro vždy nachází jiná, která došla v evoluci dále.

Porovnáním strategie naučené na deseti kartách s nejlepší strategií pětic zjistíme, že nejlepší pěticí vyhrává na 84% království. Nejlepší trojice je lepší dokonce pro 91% království. Při porovnání trojic a pětic pozorujeme, že trojice mají vyšší úspěšnost na 61% království.

Další výhodou takového přístupu je, že i po přidání dalších karet království z herních rozšíření, přestože počet pětic roste dost rychle, počet různých trojic bude i nadále relativně nízký.

Otázkou tedy zůstává, jestli je dostatečně nízký. Celkový počet karet království napříč všemi rozšířeními je asi 300, což činí 4.5 miliónů možných kombinací trojic. Některé karty rozšíření navíc oceňují hráče za to, že má rozmanitý balíček, a to se vůbec neslučuje s přístupem omezování karetního výběru. Z těchto důvodů je tedy kritérium rozšiřitelnosti splněno, ale doba generování všech možných trojic by byla opravdu dlouhá. Ovšem čas, který je potřeba před každou hrou, bude stále konstantní. Většina hráčů navíc nepoužívá všechna rozšíření hry a pro jejich potřebu jsou možnosti rozšíření dostačující.

Běžný hráč si z výsledků této práce může odnést, že se většinou vyplatí vybrat si před hrou malou skupinu karet, o které si myslí, že je nejsilnější, a ve zbytku hry už se zbylými kartami nezabývat.

Další vývoj

Přestože projekt splňuje předsevzatá kritéria, existuje značné množství zajímavých vylepšení, z nichž některá jsou zde uvedena.

- **Náhodná počáteční konfigurace peněz:** V této práci je řešen pouze jedna ze dvou možných počátečních konfigurací peněz, protože to značně ovlivňuje zvolenou strategii. Řešením by mohlo být vybírání konkrétní strategie až po prvním kole hry, kdy je konfigurace už známa.
- **Nové karty království:** Napříč celou prací mluvím o dalších kartách království a tudíž jejich implementace by byl nejpřirozenější směr rozšiřování aplikace.
- **Databáze:** Ukládání strategií do souboru není vždy úplně spolehlivé. Vhodnější by bylo použít nějaký framework pro databázi.
- **Hra více hráčů:** V aplikaci dosud chybí možnost hrát hráč proti hráči. Je zde taky možnost hrát hru po síti.
- **12 karet vítězných bodů:** Tato varianta hry, kdy hrají dva hráči s počtem vítězných bodů, který je původně určen pro čtyři hráče, má dle mého názoru větší potenciál, co se týče zajímavých strategií. Hra je totiž delší a hráči si musí lépe rozmyslet, jestli už je čas kupovat provincie nebo ještě budovat silný balíček.
- **Hra ve více hráčích:** Po přidání více hráčů se značně změní síla některých karet a celkově vzroste komplexita hry.
- **Lepší heuristika a vývoj herní strategie:** Heuristika použita v této práci není špatná, ale je relativně jednoduchá. Pomocí neuronových sítí nebo i jinými způsoby by bylo možné učinit protihráče efektivnějším.
- **Uživatelská aplikace:** V aplikaci nejsou použity skutečné obrázky karet a možná by bylo pro hráče příjemné mít možnost si hru zahrát v češtině.

Seznam použité literatury

- [1] Montana Banana. Rio grande games, 2019. URL <http://riograndegames.com/games.html?id=278>. [cit. 11.7.2019].
- [2] Nicholas Knutsen. Complete rules for dominion and all its expansions, 2018. URL http://madforest.com/Dominion_CompleteRules_v4.1.pdf. [cit. 10.7.2019].
- [3] Allen Varney. Dominion over all, Aug 2009. URL https://v1.escapistmagazine.com/articles/view/video-games/issues/issue_213/6338-Dominion-Over-All. [cit. 13.7.2019].
- [4] Rasmus Bille Fynbo and Christian Sinding Nellemann. Developing an agent for dominion using modern ai-approaches. *M. Sc. IT, Media Technology and Games (MTG-T) Center for Computer Games Research*, 2010.
- [5] Matthew Fisher. Provincial: A kingdom-adaptive ai for dominion, 2014. URL <https://graphics.stanford.edu/~mdfisher/DominionAI.html>. [cit. 13.7.2019].
- [6] Charles Darwin. *On the origin of species by means of natural selection: or, The preservation of favoured races in the struggle for life*. Public Domain Books, Mar 1998. ISBN 145381468X.
- [7] Zbigniew Michalewicz and Stephen J Hartley. *Genetic algorithms+ data structures= evolution programs*, volume 18. Springer, 1996. ISBN 978-3-662-03315-9.
- [8] Dominion strategy. URL <https://dominionstrategy.com/>. [cit. 16.7.2019].
- [9] Microsoft. System.valuetuple 4.5.0, May 2018. URL <https://www.nuget.org/packages/System.ValueTuple/>.
- [10] Stephen Toub. Getting random numbers in a thread-safe way, Feb 2009. URL <https://devblogs.microsoft.com/pfxteam/getting-random-numbers-in-a-thread-safe-way/>. [cit. 15.7.2019].
- [11] Gonzalo Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.

A. Karty

Název karty	Typ karty	Efekt
Čarodějnice	akce-útok	+2 karty Ostatní hráči si vezmou kletbu.
Dílna	akce	Vezmi si kartu o ceně nejvýše \$4.
Dobrodruh	akce	Ukazuj karty ze svého balíčku, dokud nenarazíš na dvě karty peněz. Obě karty peněz si vezmi do ruky, ostatní odkryté karty odlož na svůj odkládací balíček.
Dřevorubec	akce	+1 nákup \$2
Důl	akce	Zahod kartu z ruky na smetišť. Vezmi si do ruky kartu peněz až o \$3 dražší.
Hostina	akce	Zahod tuto kartu na smetišť. Vezmi si kartu o ceně nejvýše \$5.
Hradní příkop	akce-reakce	+2 karty Pokud některý z hráčů zahraje kartu útoku, můžete se bránit tak, že ukážete tuto kartu z ruky. Pokud tak učiníte, útoku jste se ubránili.
Jarmark	akce	+2 akce +1 nákup \$2
Kancléř	akce	\$2 Můžete okamžitě přesunout svůj dobírací balíček na svůj odkládací balíček.
Kaple	akce	Zahodte až 4 karty z ruky na smetišť.
Knihovna	akce	Dolízněte si karty tak, abyste jich měli na ruce 7. Můžete dávat stranou akční karty, které jste přitom získali. Po skončení odložte tyto karty na svůj odkládací balíček.
Kovárna	akce	+3 karty
Laboratoř	akce	+2 karty +1 akce
Lichvář	akce	Zahodte měďák na smetišť. Získáte za něj \$3.
Milice	akce-útok	\$2 Všichni ostatní hráči odloží karty tak, aby jim zbyly na ruce 3 karty.
Přestavba	akce	Zahodte jednu kartu z ruky na smetišť. Získáváte kartu o ceně až o \$2 vyšší než je cena zahozené karty.

Název karty	Typ karty	Efekt
Sklepení	akce	+1 akce Odložte na svůj odkládací balíček libovolný počet karet z ruky. +1 karta za každou odloženou kartu.
Špion	akce-útok	+1 karta +1 akce Všichni hráči (včetně tebe) ukážou vrchní kartu ze svého balíčku a podle tvojí volby ji buď vrátí zpět, nebo odloží na svůj odkládací balíček.
Trh	akce	+1 karta +1 akce +1 nákup \$1
Trůnní sál	akce	Vyberte jednu kartu z ruky a zahrajte ji dvakrát.
Úředník	akce-útok	Získáváte jeden stříbrňák, který umístíte na svůj dobírací balíček. Všichni ostatní hráči vyloží kartu s vítěznými body z ruky a umístí ji na vrch svého balíčku (nebo odhalí ruku bez vítězných bodů).
Vesnice	akce	+1 karta +2 akce
Zahrady	vítězné body	Karta má cenu 1 VB za každých 10 kartet ve vašem balíčku.
Zasedání rady	akce	+4 karty +1 nákup Všichni ostatní hráči +1 karta.
Zloděj	akce-útok	Všichni ostatní hráči ukážou vrchní dvě karty ze svého balíčku. Pokud přitom narazí na karty peněz, zahodí jednu podle tvého výběru. Zloděj získává libovolné množství zahozených karet. Hráči poté odloží odkryté karty na odkládací balíček.

B. Priority

Karty jsou seřazeny od nejprioritnější k méně prioritním.

- *Tránní sál*
- *Laboratoř*
- *Trh*
- *Jarmark*
- *Špeh*
- *Čarodějnice*
- *Lichvář*
- *Zasedání rady*
- *Dobrodruh*
- *Hostina*
- *Kovárna*
- *Kihovna*
- *Vesnice*
- *Sklepení*
- *Důl*
- *Úředník*
- *Přestavba*
- *Milice*
- *Hradní příkop*
- *Kaple*
- *Dřevorubec*
- *Dílna*
- *Zloděj*
- *Kancléř*