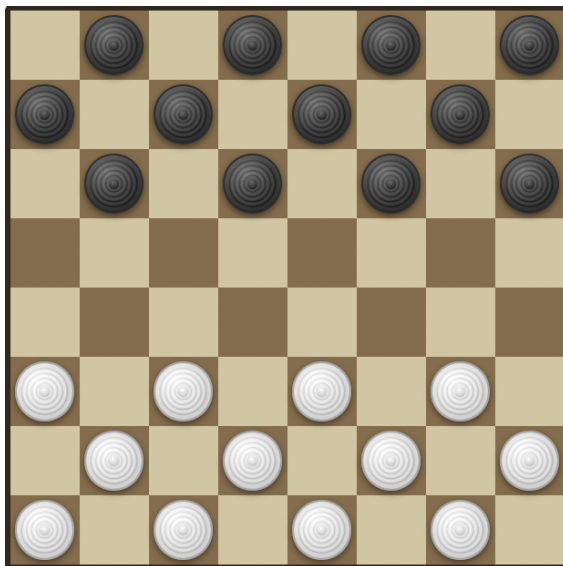


ZADÁNÍ SEMESTRÁLNÍ PRÁCE

Žravá Dáma

Zadání

Naprogramujte v jazyce ANSI C přenositelnou¹ **konzolovou aplikaci**, která umožní lidským uživatelům či umělým inteligencím změřit si své síly v méně tradiční české verzi deskové hry Dáma: takzvaná **Žravá Dáma**. Vámi implementovaná umělá inteligence bude pro zjištění optimálního tahu využívat algoritmus **Minimax**. Jednotlivé hry bude možné uložit a případně později načíst v definovaném formátu do textového souboru. Podoba konzolového rozhraní je plně ve Vaší kompetenci.



Obrázek 1: Ukázka počátečního stavu hrací plochy hry Dáma. První na tahu je vždy hráč s bílými kameny.

Program bude spouštěn příkazem `dama.exe`² s následujícím seznamem *nepovinných* přepínačů – výrazy v lomených závorkách (<>) označují povinné argumenty, výrazy v hranatých závorkách ([]) nepovinné argumenty (příklad spuštění programu je uveden na ukázce konzolového výstupu 1:

¹Je třeba, aby bylo možné Váš program přeložit a spustit na PC s operačním prostředím Win32/64 (tj. operační systémy Microsoft Windows NT/2000/XP/Vista/7/8/10/11) a s běžnými distribucemi Linuxu (např. Ubuntu, Debian, Red Hat, Arch atp.). Server, na který budete Vaši práci odevzdávat a který ji otestuje, má nainstalovaný operační systém Debian GNU/Linux 11 (bullseye) s jádrem verze 6.1.0-50-amd64 a s překladačem gcc 12.2.0.

²Přípona `.exe` je povinná i při sestavení v Linuxu, zejm. při automatické kontrole validačním systémem.

-pw <desc>	Popis hráče s bílými kameny. Povinný argument <desc> popisující hráče má strukturu " <human/minimax>[,<d>] ", kde human , resp. minimax označuje lidského hráče, resp. umělou inteligenci (výchozí: human) a výraz d určuje hloubku rozhodovacího stromu vytvářeného algoritmem Minimax (výchozí: 3). V případě lidského hráče bude hodnota d ignorována. Pokud je popis hráče zadán nesprávně, tj. hráč není human ani minimax či hloubka stromu není větší nebo rovna nule, program vypíše chybové hlášení " Invalid player description!\n " a vrátí hodnotu 1.
-pb <desc>	Inicializace hráče s černými kameny. Obdobně jako v případě přepínače -pw
-size <N>	Argument N označuje počet řádků a sloupců hrací plochy (výchozí: 8). Povoleny jsou pouze sudé hodnoty v rozsahu <4, 26> . Pokud je hodnota zadaného argumentu lichá nebo mimo povolený rozsah, program vypíše chybové hlášení " Invalid board size!\n " a vrátí hodnotu 2.
-o <gamefile>	Do souboru gamefile bude v předepsaném formátu uložen aktuální stav hry. Výchozí hodnotou je prázdný řetězec. Uložení bude provedeno automaticky na konci hry (když je znám vítěz hry) nebo v jejím průběhu zadáním příkazu " save ". Do souboru budou ukládány pouze akce hráčů, tj. speciální příkazy interpretu " save " a " quit " budou ignorovány. Pokud umístění souboru gamefile není zapisovatelné, program vypíše chybové hlášení " Invalid output file destination!\n " a vrátí návratovou hodnotu 3.
-i <gamefile>	Hra může být inicializována pomocí vstupního souboru gamefile , který má totožnou strukturu jako v případě přepínače -o . Výchozí hodnotou je prázdný řetězec. Popis hry uvedený v souboru gamefile může být přepsán argumenty -pb , -pw nebo -size . Pokud vstupní soubor gamefile nelze číst, program vypíše chybové hlášení " Invalid input file!\n " a vrátí návratovou hodnotu 4.

Vámi vyvinutý program tedy bude vykonávat následující činnosti:

1. Při spuštění bez přepínače **-i <gamefile>** bude hra inicializována argumenty **-pb**, **-pw** a **-size** a dále pokračovat v *interaktivním režimu*, tj. postupně čekat na vstupy obou hráčů. Zadané příkazy vyhodnotí a bude vyžadovat další, dokud nebude hra dokončena nebo nebude uveden příkaz "**quit**". Při zadání chybného výrazu bude vypsána odpovídající chybová hláška a hráč bude vyzván k opětovnému zadání vstupu.
2. Při spuštění s přepínačem **-i <gamefile>** bude hra načtena ze souboru **<gamefile>**. V souboru uvedené inicializační hodnoty je možné přepsat přepínači **-pb**, **-pw** nebo **-size**. Součástí vstupního souboru bude i seznam akcí obou hráčů a příkazů interpretu, které budou dávkově zpracovány. Pokud během zpracování příkazů nedojde k chybě nebo k ukončení hry (dokončením hry nebo uvedením příkazu "**quit**"), přejde aplikace do *interaktivního režimu*, ve kterém budou hráči moci hru dokončit. Pokud během zpracování příkazů dojde k chybě, program vypíše odpovídající chybovou hlášku a vrátí dedikovaný chybový kód.

Hotovou práci odevzdejte v jediném archivu typu ZIP prostřednictvím automatického odevzdávacího a validačního systému. Postupujte podle instrukcí uvedených na webu předmětu. Archiv nechtě obsahuje všechny zdrojové soubory potřebné k přeložení programu, **Makefile** pro Windows i Linux (pro překlad v Linuxu připravte soubor pojmenovaný **Makefile** a pro Windows **Makefile.win**) a dokumentaci ve formátu PDF vytvořenou v typografickém systému $\text{\LaTeX}$. Bude-li některá z částí chybět, kontrolní skript Vaši práci odmítne.

Specifikace pravidel dámy

Jedná se o mírně upravenou variantu méně tradiční české žravé dámy. Hlavní rozdíly oproti standardním pravidlům jsou následující:

- **Žravá dáma** – cílem hry je přijít o všechny své kameny. Vyhrává hráč, který již nemá žádný kámen nebo žádný legální tah.
- **Zakázané vícenásobné brání** – během jednoho tahu je dovoleno provést nejvýše jedno brání. Po přeskočení soupeřova kamene tedy tah končí, i když by podle standardních pravidel bylo možné pokračovat dalším bráním.
- **Remíza** – hra může skončit remízou po dosažení stanoveného limitu tahů bez pokroku. Ostatní remízové situace není nutné implementovat.

Ostatní pravidla se řídí běžnými pravidly dámy. Detailní popis je uveden na konci dokumentu.

Specifikace vyhodnocovaných výrazů

Výrazem může být pouze příkaz interpretu, nebo akce hráče. Interpret je **case-insensitive**, tj. nerozlišuje velká a malá písmena (zadané výrazy budou převedeny na malá písmena). Použití příkazů interpretu nebude zapsáno do výstupního souboru zadaného přepínačem `-o`. Příkazy ovšem budou vyhodnoceny, pokud se vyskytnou v souboru vstupním, který je dán přepínačem `-i`. Interpret bude pracovat s následující minimální množinou podporovaných příkazů:

<code>save</code>	Pokud je zadán výstupní soubor (přepínač <code>-o</code>), uloží do něj aktuální stav hry. Pokud se uložení stavu hry do výstupního souboru nepodaří (soubor nebyl zadán nebo není zapisovatelný), program vypíše chybové hlášení <code>"Invalid output file destination!\n"</code> a v případě režimu dávkového zpracování vrátí s návratovou hodnotou 3.
<code>quit</code>	Ukončí program s návratovou hodnotou <code>EXIT.SUCCESS</code> .

Hráči se během hry střídají v zadání svých akcí, které postupně mění stav hry. Začíná hráč s bílými kameny. Akce hráčů budou zadávány následovně:

<code><c1><r1>-<c2><r2></code>	<code><c1></code> je <i>písmeno</i> označující sloupec na němž stojí kámen před provedením tahu, <code><r1></code> je <i>číslo</i> označující řádek na němž stojí kámen před provedením tahu. <code><c2></code> a <code><r2></code> označují sloupec a řádek na němž kámen přistane po provedení tahu. Příklad: <code>c3-b4</code> , <code>b4-d6</code> . V případě, že uživatel zadá neplatný tah, program vypíše chybové hlášení <code>"Invalid move!\n"</code> . Pokud k neplatnému tahu dojde navíc při načítání hry ze souboru, program skončí s návratovou hodnotou 10.
--	---

Specifikace formátu vstupních a výstupních souborů

Vstupní a výstupní soubory budou mít stejnou strukturu. V první části souboru budou v blocích (hranatých závorkách) uvedeny hodnoty inicializačních argumentů `-pw`, `-pb` nebo `-size`. Výstupní soubor vždy obsahuje všechny tři inicializační bloky. Pokud ve vstupním souboru některý blok

(nebo bloky) chybí, jsou použity výchozí hodnoty. V případě načítání hry ze souboru mohou inicializační hodnoty v souboru být přepsány použitím argumentů. Bezprostředně po bloku "[game]" bude následovat seznam hráči provedených akcí nebo příkazů interpretu.

Strukturu výstupních souborů je nutné striktně dodržet. Ukázky vstupních a výstupních souborů můžete vidět na konzolovém rozhraní 2.

Specifikace konzolového rozhraní

Dodržujte přesně znění varovných hlášek. Zbytek implementace konzolového rozhraní je zcela ve Vaší kompetenci a vizualizace hrací plochy není povinná. Přesto je její vytvoření důrazně doporučeno, jelikož usnadňuje ladění programu, demonstraci jeho funkčnosti při závěrečné zkoušce a v neposlední řadě umožňuje lépe ocenit výsledek Vaší práce.

Fantazii se meze nekladou. Mějte však na paměti přenositelnost aplikace mezi různými operačními systémy a prostředími. Použití barevného výstupu nebo méně obvyklých znaků není doporučeno (ale není zakázáno). Dbejte ale na to, aby jejich případná nepodpora na jiném systému neomezovala samotnou funkčnost programu.

Algoritmus Minimax

Vámi implementovaná umělá inteligence bude pro určení optimálního stavu využívat algoritmus Minimax. Stav, kde protihráč maximalizujícího hráče má více kamenů musí být ohodnocen lépe, než stav opačný (účelem hry je přijít o všechny kameny). Algoritmus využívá k ohodnocování jednotlivých stavů hry funkci, jejíž výpočet bude v tomto případě následující:

$$eval(s) = minstones(s) - maxstones(s),$$

kde $maxstones(s)$, resp. $minstones(s)$ označuje počet kamenů maximalizujícího, resp. minimalizujícího hráče ve stavu s (zahrnuje běžné kameny i dámy).

Koncové stavy musí mít přednost před běžným ohodnocením pozice. Výhra, prohra a remíza se proto nesmějí vyhodnocovat pouze pomocí počtu zbývajících kamenů a dam. Konkrétní číselné ohodnocení koncových stavů musí zaručit, že:

- výhra je vždy výhodnější než jakýkoliv nekoncový stav,
- prohra je vždy méně výhodná než jakýkoliv nekoncový stav,
- remíza má ohodnocení 0.

Všechny legální tahy musí být pro účely algoritmu Minimax zpracovávány v jednoznačně definovaném pořadí. Toto pořadí se určuje podle *výchozího pole kamene* a následně podle *cílového pole tahu*.

Pro **bílého hráče** platí následující pořadí:

1. Kameny se zpracovávají od nejvyššího řádku (od horního okraje hrací plochy) k nejnižšímu.
2. V rámci jednoho řádku se kameny zpracovávají zleva doprava.
3. Pro každý kámen se jeho legální tahy zpracovávají podle cílového pole, nejprve od nejvyššího řádku k nejnižšímu a v rámci řádku zleva doprava.

Pro **černého hráče** platí pořadí opačné, tedy od nejnižšího řádku (od spodního okraje hrací plochy) k nejvyššímu a v rámci jednoho řádku zprava doleva.

Pro hloubku prohledávacího stromu $d \geq 1$ se algoritmus podívá o daný počet tahů do budoucna, kde spočte ohodnocení výsledných stavů a vybere nejvýhodněji ohodnocený tah. V případě více tahů se stejným optimálním ohodnocením vybere podle výše popsaného řazení.

Pro hloubku prohledávacího stromu $d = 0$ se algoritmus podívá přesně jeden tah do budoucna a vybere první možný tah podle výše popsaného řazení, bez ohledu na jejich ohodnocení.

Užitečné techniky a odkazy

Uvedené odkazy je možné využít při řešení úlohy. Protože se jedná o postupy standardní, lze k nim nalézt velké množství dokumentace:

1. **Dáma**

[https://cs.wikipedia.org/wiki/D%C3%A1ma_\(deskov%C3%A1_hra\)](https://cs.wikipedia.org/wiki/D%C3%A1ma_(deskov%C3%A1_hra))

2. Zahrajte si Českou dámu

<https://checkersonline.io/czech-checkers/>

3. Algoritmus Minimax

[https://cs.wikipedia.org/wiki/Minimax_\(algoritmus\)](https://cs.wikipedia.org/wiki/Minimax_(algoritmus))

4. Jednoduché vysvětlení Minimaxu

<https://www.youtube.com/watch?v=l-hh51ncgDI>

5. Algoritmus Minimax v podání Patricka Henryho Winstona

<https://www.youtube.com/watch?v=STjW3eH0Cik>

Řešení úlohy je zcela ve Vaší kompetenci – zvolte takové algoritmy a techniky, které podle vás nejlépe povedou k cíli.

Detailní popis pravidel dámy

Tato sekce definuje pravidla hry, která musí implementace dodržovat. Pokud není výslovně uvedeno jinak, platí tato specifikace před případnými odlišnostmi standardních pravidel dámy. Cílem je zajistit, aby všechny implementace používaly stejná pravidla a dosahovaly porovnatelných výsledků.

1. Hrací deska a počáteční pozice

Velikost hrací desky $N \times N$ je zadána při spuštění programu. Platí následující omezení:

$$N \in \{4, 6, 8, \dots, 26\}.$$

Hrací deska je čtvercová a obsahuje pouze tmavá a světlá pole. Herní figurky se umísťují výhradně na tmavá pole. Světlá pole zůstávají po celou dobu hry prázdná a figurky na ně nikdy nevstupují. Řádky hrací plochy jsou indexovány čísly 1 až N odspoda nahoru, sloupce hrací plochy jsou indexovány písmeny od a zleva doprava, podobně jako standardní šachovnice. (Toto je vnější indexace pro hráče zadávající tahy, vnitřní indexaci polí implementujte, jak uznáte za vhodné.)

Na začátku hry jsou obsazena všechna tmavá pole kromě polí ve dvou prostředních řadách. Počáteční pozice tedy obsahuje:

$$K = \frac{(N-2)N}{2}$$

kamenů celkem, přičemž každý hráč má přesně polovinu všech kamenů:

$$K_{\text{bílý}} = K_{\text{černý}} = \frac{(N-2)N}{4}.$$

Bílé kameny jsou umístěny v dolní části desky a pohybují se směrem k hornímu okraji. Černé kameny jsou umístěny v horní části desky a pohybují se směrem k dolnímu okraji. Bílé kameny tedy postupují k řádku s indexem N , černé kameny k řádku s indexem 1.

Bílý hráč začíná jako první.

2. Základní pohyb kamenů

Běžný kámen se může přesunout o jedno pole diagonálně vpřed, pokud je cílové pole prázdné. Pokud běžný kámen dosáhne poslední řady na soupeřově straně desky, promění se na dámu. U bílého hráče jde o řádek s indexem N , u černého hráče o řádek s indexem 1. Dáma se může pohybovat diagonálně vpřed i vzad o libovolný počet polí, pokud jsou všechna pole mezi výchozím a cílovým polem prázdná.

3. Braní kamenů

Braní je povinné. Pokud má hráč alespoň jeden legální tah, při kterém bere soupeřův kámen, musí zvolit tah, při kterém dojde k braní. Obyčejný přesun bez braní v takové situaci není dovolen. Pokud má hráč alespoň jeden legální tah, při kterém bere soupeřův kámen **dámou**, musí zvolit některé z legálních braní dámou. Obyčejný přesun bez braní ani braní běžným kamenem není v takové situaci dovoleno.

Braní běžným kamenem. Běžný kámen bere soupeřův kámen na diagonálně sousedícím poli skokem přes něj. Pole bezprostředně za braným kamenem musí být volné. Běžný kámen při braní přistává vždy právě na tomto poli.

Braní dámou. Dáma může brát soupeřův kámen na stejné diagonále, pokud mezi dámou a soupeřovým kamenem neleží žádná jiná figurka a bezprostředně za soupeřovým kamenem se nachází alespoň jedno volné pole.

Po přeskočení soupeřova kamene může dáma přistát na libovolném volném poli za braným kamenem na stejné diagonále. Není tedy nutné přistát bezprostředně za braným kamenem.

Při jednom skoku smí dáma přeskočit právě jeden soupeřův kámen. Nesmí přeskočit dva nebo více kamenů současně. Vlastní kámen ani jiná figurka, která blokuje diagonálu, nesmí být přeskočena.

Vícenásobné braní V standardních pravidlech dámy může (respektive musí) hráč táhnout jedním kamenem vícekrát po sobě, pokud na konci skoku může provést další skok. Pro účely naší implementace tyto několika-násobné skoky **nejsou povoleny**. Jeden tah jednoho hráče vždy sestává z přesně jednoho tahu jedním kamenem z jedné pozice na pozici novou. Tímto jedním přesunem tah hráče končí a to i v případě, že by podle běžných pravidel dámy mohl jeho tah pokračovat dalším braním.

4. Konec hry

Výsledek hry se vyhodnocuje ihned po dokončení tahu. Hra končí okamžitě, jakmile nastane některá z následujících situací:

- Hráč nemá na desce žádný kámen ani dámu. Tento hráč **vyhrává**.
- Hráč nemá žádný legální tah. Tento hráč **vyhrává**.
- Je dosaženo limitu tahů bez pokroku. V takovém případě hra končí remízou.

Tahy bez pokroku Pro remízy se používá počítadlo tahů bez pokroku. Za pokrok se považuje *braní soupeřova kamene* nebo *proměna běžného kamene na dámu*. Pokud během tahu nedojde ani k braní, ani k proměně, zvýší se počítadlo o jedna. Počítadlo se zvyšuje samostatně po tahu bílého i po tahu černého. Jakmile počítadlo dosáhne hodnoty X , hra končí remízou.

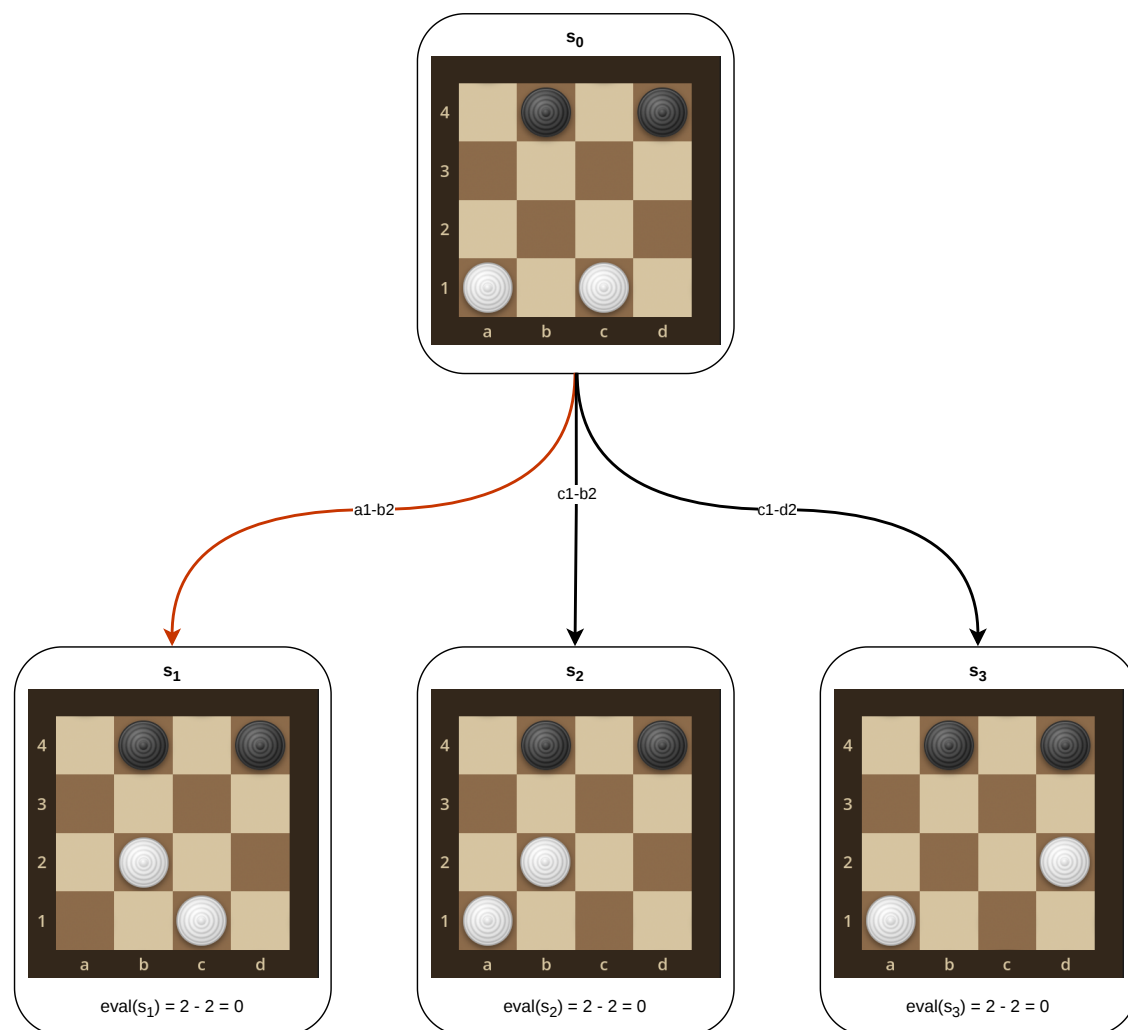
Limit je závislý na velikosti hrací desky:

$$X = 2N^2,$$

kde N je počet řádků (a současně sloupců) hrací desky.

Počítadlo slouží primárně k ukončení hry v případě, že se dva minimaxy s malou hloubkou zanoření točí v kruhu.

Přílohy



Obrázek 2: Ukázka stanovení optimálního kroku pomocí algoritmu Minimax s hloubkou stromu $d=1$, tj. díváme se pouze jeden tah dopředu. Hrany rozhodovacího stromu jsou seřazeny postupně podle souřadnic provedených tahů. Hodnotící funkce $eval(s)$ zde pro všechny tahy nabývá stejné hodnoty, algoritmus tedy vybere hranu nejvíce vlevo.

```

1 user@machine:~$ cd semestralka && ls
2 doc src CMakeLists.txt dokumentace.pdf Makefile Makefile.win
3 user@machine:~/sestralka$ ./dama.exe -pw robot
4 Invalid player description!
5 user@machine:~/sestralka$ echo $?
6 1
7 user@machine:~/sestralka$ ./dama.exe -size 7
8 Invalid board size!
9 user@machine:~/sestralka$ echo $?
10 2
11 user@machine:~/sestralka$ ./dama.exe -o /nonexistent/output.txt
12 Invalid output file destination!
13 user@machine:~/sestralka$ echo $?
14 3
15 user@machine:~/sestralka$ ./dama.exe -i non_existing_file.txt
16 Invalid input file!
17 user@machine:~/sestralka$ echo $?
18 4
19 user@machine:~/sestralka$ ./dama.exe -o /validator/interactive_output.txt
20 (white:human)> c3-b4
21 (black:human)> b4-c5
22 Invalid move!
23 (black:human)> d6-c5
24 (white:human)> e3-d4
25 Invalid move!
26 (white:human)> b4-d6
27 (black:human)> pass
28 Invalid move!
29 (black:human)> e7-c5
30 (white:human)> SAVE
31 (white:human)> exit
32 Invalid move!
33 (white:human)> let me out!!
34 Invalid move!
35 (white:human)> quit
36 user@machine:~/sestralka$ echo $?
37 0
38 user@machine:~/sestralka$ ./dama.exe -i /validator/batch_input_invalid.txt
39 Invalid move!
40 user@machine:~/sestralka$ echo $?
41 10
42 user@machine:~/sestralka$ ./dama.exe -i /validator/batch_input1.txt
43 (white:minimax)> a1-b2
44 (black:human)> save
45 Invalid output file destination!
46 (black:human)> quit
47 user@machine:~/sestralka$ echo $?
48 0
49 user@machine:~/sestralka$ ./dama.exe -i /validator/batch_input1.txt -o /validator
   /batch_output1.txt
50 (white:minimax)> a1-b2
51 (black:human)> save
52 (black:human)> quit
53 user@machine:~/sestralka$ echo $?
54 0
55 user@machine:~/sestralka$ ./dama.exe -i /validator/batch_input2.txt
56 Invalid output file destination!
57 user@machine:~/sestralka$ echo $?
58 3
59 user@machine:~/sestralka$ ./dama.exe -i /validator/batch_input2.txt -o /validator
   /batch_output2.txt -pb minimax,4
60 user@machine:~/sestralka$ echo $?
61 0

```

Listing 1: Ukázka práce programu dama.exe v interaktivním a dávkovém režimu. Pro jednoduchost zde není vypisován stav hry.

```

1 user@machine:~$ cd /validator && ls
2 batch_input_invalid.txt batch_input1.txt batch_output1.txt interactive_output.txt
   batch_input2.txt batch_output2.txt
3 user@machine:/validator$ cat batch_input_invalid.txt
4 [size 4]
5 [game]
6 a1-a2
7 user@machine:/validator$ cat interactive_output.txt
8 [pw human]
9 [pb human]
10 [size 8]
11 [game]
12 c3-b4
13 d6-c5
14 b4-d6
15 e7-c5
16 user@machine:/validator$ cat batch_input1.txt
17 [pb human]
18 [pw minimax,1]
19 [size 4]
20 user@machine:/validator$ cat batch_output1.txt
21 [pw minimax,1]
22 [pb human]
23 [size 4]
24 [game]
25 a1-b2
26 user@machine:/validator$ cat batch_input2.txt
27 [game]
28 c3-d4
29 b6-c5
30 d4-b6
31 save
32 a7-c5
33 e3-d4
34 c5-e3
35 save
36 quit
37 user@machine:/validator$ cat batch_output2.txt
38 [pw human]
39 [pb minimax,4]
40 [size 8]
41 [game]
42 c3-d4
43 b6-c5
44 d4-b6
45 a7-c5
46 e3-d4
47 c5-e3

```

Listing 2: Ukázka vstupních a výstupních souborů použitých v ukázce konzolového rozhraní 1.