Zadání projektu SUI 2019/20

Karel Beneš

18. listopadu 2019

Cílem projektu v SUI je vytvořit umělou inteligenci (AI) pro hru Dicewars. Prostředí pro vývoj a evaluaci AI je k dispozici na Githubu[1], implementačním jazykem je Python3.

Vytvořená AI musí netriviálním způsobem využívat některé z technik umělé inteligence a/nebo strojového učení.

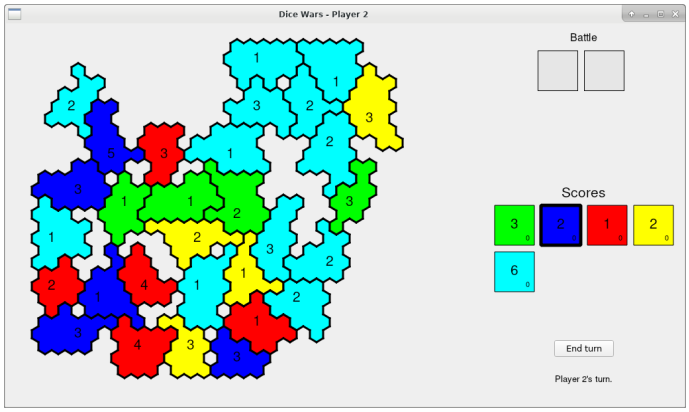
Changelog

- 18. listopadu 2019: první verze.

1 Podstata hry

Dicewars jsou hrou s otevřenou informací a prvkem náhody. Cílem hry je ovládnout celou herní plochu. Ta se skládá z *polí*, na nichž je jedna až osm *kostek*.

Dva až osm hráčů se v daném pořadí střídá v *tazích*. Tah hráče sestává z předem neomezeného počtu *akcí*. Hráčovou akcí je *útok* nebo *ukončení tahu*.



Obrázek 1: Ukázka stavu hry. Zelený hráč vlastní čtyři pole ve dvou oblastech oblastech. Celkem má k dispozici 7 kostek, ale útočit může pouze ze dvou polí, kde jsou alespoň dvě kostky. Největší z oblastí má 3 pole, to je tak hráčovo skóre.

Útok je veden z vlastního vybraného pole obsazeného více než jednou kostkou na sousedící nepřátelské pole. Útok je následně vyhodnocen tak, že útočník i obránce hodí všemi (šestistěnnými) kostkami na příslušných polích a porovná se jejich součet. Jestliže je útočníkův součet vyšší, útok je úspěšný, obránce ztrácí pole i veškeré kostky na něm a útočník na dobyté pole přesouvá až na jednu všechny útočící kostky. V opačném případě ztrácí útočník všechny kostky až na jednu.

Při ukončení tahu je nalezena hráčova největší souvislá oblast. Hráč potom obdrží počet kostek rovný počtu polí v této oblasti. Ty se sečtou s kostkami v jeho *rezervě*[2] a rozdělí se náhodně mezi jeho pole. Pokud již není možné kostky

[1] https://github.com/ibenes/dicewars

[2] Pokud jejich počet přesáhne 64, je na tuto hodnotu omezen.

přidávat (protože už všechna hráčova pole obsahují 8 kostek), zbylé kostky zůstanou v rezervě. Každý hráč začíná s prázdnou rezervou.

1.1 Technická omezení Vašich řešení

S ohledem na férovou soutěž se všechna řešení musí podrobit určenému časovému omezení: Objekt umělé inteligence má deset vteřin na konstrukci. Na každé rozhodnutí je dané množství času, o kterém je AI informována. Čas je na začátku hry 10 vteřin, s každou odevzdanou akcí (útokem nebo koncem tahu) se přidává 0.1 vteřiny, podobně jako v šachu.

Na paměť nejsou z technických důvodů kladena tvrdá omezení, ale garantujeme 1GB paměti pro Vaše řešení. Pokud tento limit překročíte a z Vaší implementace vyběhne `MemoryError`, bude to považováno za Vaši chybu (stejně jako všechny jiné výjimky, vyjma té vyvolané časovačem).

2 Odevzdávání a hodnocení

Projekt bude vypracováván v týmech o třech členech. Termín pro odevzdání projektu je 5. ledna 2020, 23:59.

Součástí odevzdání budou tři položky:

1. Implementace umělé inteligence. Modul (`xlogin99.py`) nebo balíček (`xlogin99/..`) pro Python3, který definuje třídu `AI` odpovídajícího rozhraní. Pokud Vaše řešení zahrnuje nějaký natrénovaný model nebo jiná pomocná data, volte organizaci do balíčku a vložte je do něj. Používat můžete všechny standardní moduly Pythonu. Nesmíte měnit reakce na signály, pouštět další procesy nebo zapisovat na disk. Obecně se chovejte slušně a v případě nejistoty se zeptejte na fóru.
2. Dokumentace (`xlogin99.pdf` nebo `dokumentace.pdf`) ve formátu PDF a českém, slovenském nebo anglickém jazyce. Dokumentace musí popisovat Vaše řešení a umožnit reprodukci Vaší práce. Důraz věnujte tomu, jak jste své řešení vyhodnocovali. Nebojte se popsat významné kroky zlepšování Vašeho řešení. Očekávaný rozsah jsou 3 strany A4, ale pokud Vám stačí méně nebo nutně potřebuje více, nechte se vést obsahem.
3. Skripty, a příp. data, (`supplementary/..`) umožňující reprodukci natrénovaných částí odevzdaného řešení. Pokud je práce čistě implementační, nemusí být přiloženo.

Všechno to bude zabaleno v jednom souboru `xlogin99.zip`, který bude odevzdán do WISu.

Body se za projekt udělují následovně:

- 10b za promyšlenost, nápaditost a původnost řešení. Hodnocen bude výkon v oblasti umělé inteligence a/nebo strojového učení. Ostatní inženýrská práce je vítána, ale nebude zdrojem bodů. Z dodaných AI by body dostaly `dt.wpm_{s,d,c}`, zbylé jsou natvrdo nadrátované heuristiky a tudíž nevyhovují.
- 10b za výkonnost odevzdané AI. Vaše řešení bude spuštěno proti poskytnutým AI označeným jako `xlogin00`, `dt.rand`, `dt.sdc`, `dt.ste`, `dt.stei`, `dt.wpm_d`, `dt.wpm_s` a `dt.wpm_c` ve hrách dvou až pěti hráčů. U každé AI bude změřen podíl vítězných her[3] a za každou překonanou AI dostane Vaše řešení rovně 0.3125 bodu.

Dále budou uspořádány turnaje v 2-, 3-, 5- a 7-hráčových hrách mezi všemi dodanými řešeními. Vítěz, druhý a třetí v pořadí v každém turnaji dostanou po řadě 3, 2 a 1 bonusový bod, maximálně lze takto získat 5 bodů.

Vyhodnocovat se bude na náhodně generovaných mapách.

3 Minitutoriál

1. Stáhněte si implementaci hry z GitHubu. Podle návodu v README zprovozněte hru člověka s počítačem.
2. Zkopírujte `template.py` na `vas_login.py`.
3. Zahrajte si: `python3 ./scripts/dicewars-human.py --ai vas_login`. Měli byste vidět, že umělá inteligence nedělá nic.

[3] V dostatečném počtu her, řádově tisících.

4. Donuťte umělou inteligenci něco dělat: Mrkněte do `dicewars/ai/xlogin00.py` a zkopírujte si kód pro náhodné útočení.
5. Nenechte ji dělat si co chce: Inspirujte se v `dicewars/ai/xlogin42/phased.py` kódem pro filtrování akcí a výběr jedné konkrétní.

Dále se pak nebojte inspirovat implementací umělých inteligencí v balíčku `dicewars.ai.dt`. Hodit se vám budou zvlášť funkce a metody z modulů `dicewars.client.game.{board,area}` a `dicewars.ai.utils`, seznamte se s nimi.

3.1 Přejímaní cizí práce

Naprosto není problém přejímat cizí knihovny, modely nebo algoritmy. Zároveň však musíte předvést vlastní práci, která bude hodnocena. Naprogramovat algoritmus podle cizího článku je naprosto v pořádku. Vzít veřejně dostupnou implementaci nějakého RL algoritmu a natrénovat pomocí ní model je naprosto v pořádku – jen dejte pozor, zda není třeba trénovat několik dní na desítkách grafických karet. Vždy řádně citujte.

3.2 Nápady pro Váš projekt

- Naimplementujte opravdový ExpectiMiniMax.
- Natrénujte lokální prediktor toho, že dané políčko udržíte do příštího tahu.
- Natrénujte pomocí posilovaného učení klasifikátor „nejlepší akce“.
- ...