

Zadání projektu SUI 2021/22

26. září 2021

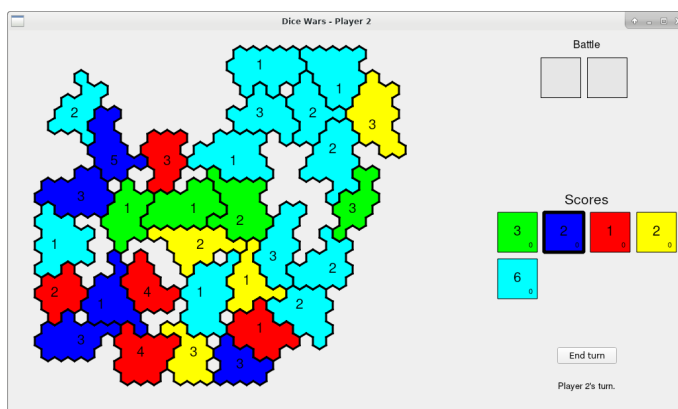
Cílem projektu v SUI je vytvořit umělou inteligenci (AI) pro pozměněnou hru Dicewars. Prostředí pro vývoj a evaluaci AI je k dispozici na Githubu¹.

Vytvořená AI musí prohledávat stavový prostor hry a netriviálním způsobem při tom využívat strojové učení.

1 Podstata hry

Dicewars jsou strategickou 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*.

Čtyři² hráči se v daném pořadí střídají v *tazích*. Tah jednoho hráče sestává z předem neomezeného počtu *akcí*. Hráčovou akcí je *přesun*, *útok* nebo *ukončení tahu*.



Obrázek 1: Ukázka stavu hry. Zelený hráč vlastní čtyři pole ve dvou 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.

Přesun je veden z vlastního vybraného pole obsazeného více než jednou kostkou na sousedící vlastní pole. Všechny přítomné kostky se snaží přesunout, až na jednu, která povinně zůstává na výchozím poli. Pokud by došlo k přeplnění cílového pole, dojde k přesunu pouze omezeného počtu kostek tak, aby se cílové pole právě naplnilo; zbylé kostky zůstanou na výchozím poli. Počet přesunů je omezen na 6 za tah.

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 je útok neúspěšný a útočník ztrácí všechny kostky až na jednu. Nicméně, pokud byl útok veden alespoň 4 kostkami, ztrácí obránce jednu kostku; byl-li útok veden osmi kostkami, ztrácí obránce dvě kostky³.

¹<https://github.com/ibenes/dicewars>

²Implementace je připravena na 2 až 8 hráčů, pro potřeby projektu se stačí věnovat čtyřhráčovým hrám.

³Obránci vždy zbyde alespoň jedna kostka.

Při ukončení tahu je nalezena hráčova největší souvislá oblast. Hráč potom obdrží počet kostek rovný počtu polí v této oblasti. Ty se sečtou⁴ s kostkami v jeho *rezervě* a rozdělí se náhodně mezi jeho pole. Pokud již není možné kostky přidávat (protože už všechna hráčova pole obsahují 8 kostek), zbylé kostky zůstanou v rezervě. Velikost rezervy je omezena na $24 - |A_p|$, kde A_p je množina hráčových polí. Každý hráč začíná s prázdnou rezervou.

2 Odevzdávání a hodnocení

Projekt bude vypracováván v týmech o nejvýše čtyřech členech. Termín pro odevzdání projektu je 19. prosince 2021, 23:59.

V následujícím textu se vyskytuje `xlogin99`. Je to placeholder pro jméno týmu, tzn. login vedoucího týmu. Neodevzdávejte řešení nebo dokumentaci s doslovným jménem `xlogin99`.

Součástí odevzdání budou tři položky:

1. Implementace umělé inteligence. Modul nebo balíček pro Python3, který definuje třídu AI odpovídajícího rozhraní, pojmenovaný podle týmu. Pokud vaše řešení zahrnuje nějaký natrénovaný model nebo jiná pomocná data, volte organizaci do balíčku a vložte data do něj.
2. Dokumentace ve formátu PDF a českém, slovenském nebo anglickém jazyce. Důraz věnujte tomu, jak jste své řešení vyhodnocovali. Nebojte se popsat významné kroky zlepšování Vašeho řešení. Očekávaný rozsah je 6 stran A4, ale pokud Vám stačí méně nebo nutně potřebuje více, nechte se vést obsahem. Dokumentaci pojmenujte `xlogin99.pdf` a uveďte v ní jména a loginy všech, kteří se na řešení podíleli.
3. Skripty a data umožňující reprodukci natrénovaných částí odevzdaného řešení. Tyto budou ve složce pojmenované `supp-xlogin99`.

Implementace musí, kromě toho aby fungovala, zachovávat následující provozní pravidla:

- Běžet s neupravenou implementací serveru a klienta, v Pythonu 3.8⁵.
- Nevypisovat nic na standardní výstup.
- Nespouštět další procesy nebo vlákna, která by pokračovala ve výpočtu i po odevzdání akce.
- Nezapisovat na disk.
- Nebránit v práci časovači, tzn. zejména neměnit reakce na signály, neodchytávat `Exception` nebo `TimeoutError` a už vůbec nepoužívat holý `except`.
- Obecně se chovat slušně. V případě nejistoty se zeptejte.
- Definovat cesty jako relativní ke kořeni repozitáře, např. `'dicewars/ai/xlogin99/model.pth'`.

Všechno to bude zabaleno v jednom souboru `xlogin99.zip`⁶. Řešení odevzdá do WISu vedoucí za celý tým.

Body se za projekt udělují následovně:

- 3b za úplné minimum: odevzdali jste všechny náležitosti, naprogramovali nějakého agenta a ten dodržuje provozní pravidla a nějak hraje.
- 6b za provedení strojového učení: modelujete něco smysluplného, popisujete jak, jak dobře to modelujete a jak moc to pomáhá hrát.
- 6b za provedení prohledávání stavového prostoru: berete v potaz alespoň jeden následující tah každého hráče, popisujete jak prohledáváte, jaké to má parametry, jaký mají vliv na rychlost a herní úspěšnost.
- 5b za výkon: Vaše řešení bude spuštěno proti poskytnutým AI označeným jako `kb.stei_at`, `dt.stei`, `kb.sdc_pre_at`, `kb.stei_dt` a `kb.stei_adt` ve hrách čtyř hráčů. U každé AI bude změřen podíl vítězných her⁷ a za každou překonanou AI dostane Vaše řešení 1 bod.

⁴Pokud součet přesáhne 64, je na tuto hodnotu omezen.

⁵Na merlinovi je k dispozici referenční binárka `python3.8`.

⁶Tzn. ne `.tgz` a už vůbec ne `.7z`, `.rar` apod.

⁷V dostatečném počtu her, řádově tisících.

Řešení, která splní podmínky zadání, tzn. mají nějakou komponentu založenou na strojovém učení a provádí prohledávání stavového prostoru, se zúčastní turnaje ve hře čtyř hráčů. Řešení, která se v turnaji umístí na prvním až pátém místě, získají po řadě pět až jeden bonusový bod.

3 MiniTutoriál

1. Stáhněte si implementaci hry z GitHubu. Podle návodu v README zprovozněte hru člověka s počítačem.
2. Zkopírujte `template.py` na `xlogin99.py`.
3. Zahrajte si: `python3 ./scripts/dicewars-human.py --ai xlogin99`. Měli byste vidět, že umělá inteligence nedělá nic.
4. Donuťte umělou inteligenci něco dělat: Mrkněte do `dicewars/ai/kb/xlogin00.py` a zkopírujte si kód pro náhodné útočení.
5. Nenechte ji dělat si co chce: Mrkněte do `dicewars/ai/kb/xlogin42/phased.py` a inspirujte se kódem pro filtrování akcí a výběr jedné konkrétní.

3.1 Nápady pro Váš projekt

- Osvěžte si (Expecti)MiniMax a vymyslete, jak ho rozšířit pro 4 hráče. Zajímavé klíčové slovo je *MaxN*.
- Natrénujte heuristiku vybírající zajímavé tahy, např. model odhadující, zda nějaké políčko udržíte přes tah ostatních, nebo rovnou klasifikátor „nejlepší akce“ natrénovaný pomocí posilovaného učení.
- Natrénujte model odhadující, kdo má v dané situaci jakou šanci na výhru, a použijte ho pro ohodnocování prohledávaných stavů.
- Pro trénování modelů budete potřebovat data. Nebojte se upravit si za tím účelem AI, vytvořit úplně nového bota nebo prodloužit časové limity v `dicewars.config`. Jen myslete na to, že výsledné řešení by mělo být na úrovni kódu rozumně čisté a že musíte vyhodnocovat s nepozměněným serverem i klientem.

3.2 Používání cizích algoritmů, toolkitů a podobně

Je naprosto v pořádku použít hotové implementace algoritmů pro posilované učení nebo osvědčené architektury neuronových sítí nebo cokoliv dalšího, s výjimkou cizích řešení tohoto projektu, včetně loňských. Řádně citujte, v dokumentaci a případně i zdrojovém kódu. Jen dejte pozor, zda je budete schopni natrénovat na hardwaru, který máte k dispozici, a poběží na jednom CPU s časovým omezením daným v projektu.

Pro běh řešení bude k dispozici PyTorch 1.9.1. Kompatibilitu Vašeho řešení si můžete ověřit na Merlinovi, kde je k dispozici referenční prostředí (`/pub/courses/sui/env-sui.2021`). Chcete-li nějaké jiné knihovny⁸, nad rámec standardního Pythonu a položek z `requirements.txt`, napište.

⁸Explicitní výjimkou je TensorFlow, který je zapovězen, neboť se mu v různých řešeních z minulých let dařilo opakovaně generovat segfaulty.