

# TURNAJ V OWARE

Uživatelská dokumentace

Jonáš „Jidáš“ Havelka  
jonas.havelka@volny.cz

13. června 2021

<https://github.com/JoHavel/Turnaj-v-Oware>

# Úvod

Oware je africká desková hra pro dva hráče, jejíž jednoduchá pravidla naleznete<sup>1</sup> v M&M 28.1. Tento projekt implementuje turnaj autonomních Python agentů tak, aby vyhovoval korespondenčnímu semináři M&M.

Hlavní část je samozřejmě programování skriptu dle sekce 1. Lze si však hru i vyzkoušet (včetně hraní s roboty), a to jak textově dle sekce 2, tak i graficky dle sekce 3. Nakonec lze uspořádat turnaj mezi agenty dle sekce 4.

## 1 Agenti

Skript má vypadat přibližně takhle (více příkladů je na GitHubu ve složce `oware/agents`):

---

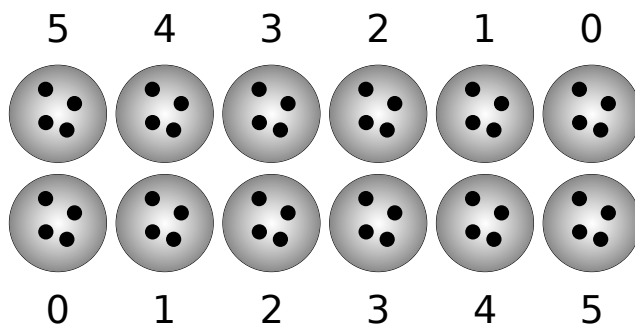
```
1 class Agent:
2     name = "Jan Novák"
3
4     def play(self, game) -> None:
5         nase_skore = game.player_score
6         skore_souper = game.opponent_score
7         nase_doliky = game.player_pits
8         doliky_souper = game.opponent_pits
9
10        neprazdne = [i for i in range(6) if
11                    ↪ nase_doliky[i] != 0]
12        self.move = random.choice(neprazdne)
13
14        return
15
16        move = -1
```

---

Musí implementovat třídu `Agent`, která má property `name`, která obsahuje vaše jméno jako řetězec (`str`). Dále `Agent` musí obsahovat metodu `play`, která (kromě obvyklého parametru `self`) bere parametr `game` (viz dále), nic nevrací a někdy během svého tahu nastaví `move` na index dílíku (z intervalu `[0, 5]`), který chcete zahrát. Tj. skript musí obsahovat řádky 1, 2 (s vaším jménem), 4, 11 (s jinou pravou stranou rovnítka), 15.

---

<sup>1</sup>Dostupné na <https://mam.mff.cuni.cz/media/cislo/pdf/28/28-1.pdf>.



Obrázek 1: Očíslování dolíků

Volání vašeho skriptu probíhá tak, že před začátkem hry vytvoří objekt třídy `Agent`, tedy pokud si chcete inicializovat nějaká pole apod. můžete tak učinit ve funkci `__init__(self):`, musíte to však stihnout do 1 sekundy. Následně bude každý tah volána vaše funkce `play`, která bude vždy po 1 sekundě ukončena (pokud neskončí sama již dříve). Jakmile skončí, je z property `move` přečten váš tah. Pokud je náhodou neplatný (tedy je například mimo interval  $[0, 5]$ ), zahraje se náhodný platný tah.

Objekt `game` (implementující třídu `Game`) poskytuje `game.player_score` a `game.opponent_score`, což jsou dvě celá čísla (`int`) – skóre hráče na tahu a skóre protivníka. Dále má `game.player_pits` a `game.opponent_pits`, což jsou dvě pole celých čísel (`[int]`) – počty semínek u hráče na tahu a u protivníka, obojí proti směru hodinových ručiček (v kladném směru jako na obrázku 1). Nakonec je tu property `game.ended`, což je pravdivostní hodnota (`bool`), zda hra už neskončila<sup>2</sup>.

Dále obsahuje tři metody: `game.copy()`, která nebere parametry a vrátí vám přesnou (ale nezávislou) kopii dané hry, na které můžete testovat, co by se stalo, kdyby... Na to se hodí metoda `game.play(move, rotate=True)`, která bere jako povinný parametr, který dolík chcete „zahrát“ (celé číslo od 0 do 5). Pokud je tah platný, vrátí `True` a změní stav hry na stav po tahu (tj. včetně prohození hráče na tahu a jeho protivníka), pokud je neplatný, tak vrátí `False` a stav se nezmění. Pokud nechcete měnit strany, tak `game.play()` má ještě nepovinný parametr `rotate`, který lze nastavit na `False`, a v případě potřeby poté prohodit strany metodou `game.rotate()`.

<sup>2</sup>Po skončení hry už váš skript samozřejmě nebude volán, ale tato property se vám může hodit, pokud budete spouštět `game.play`.

## 2 Textové prostředí

Textové prostředí lze spustit dvěma způsoby: souboj dvou agentů lze vyvolat jako `text.py -s ... ..` (první agent bude začínat, odehrají bez vypisování čehokoliv jednu hru a příkaz vypíše jen, kdo vyhrál a s jakým skóre<sup>3</sup>). Například:

```
text.py -s agents/nahodny.py agents/nejlepsi_po_mem.py
```

Druhý způsob je jednoduché spuštění `text.py`, které může mít přepínač `-r`, s kterým otáčí herní plochu po každém tahu (aktuálně hrající hráč bude vždy dole, tj. vypsáný později). S `-f` pak skript místo čísel bude používat Unicode znaky reprezentující semínka. V tomto režimu poté skript přijímá příkazy (jednořádkový text na standardním vstupu) od uživatele a vypisuje stav hry. Příkazy jsou:

- `q` – quit. Ukončí program.
- `n` – new game. Začne novou hru. V případě, že předchozí nebyla dohrána, zeptá se, zda opravdu. Očekává odpověď `y` = ano, či `n` = ne.
- `číslo` – libovolné číslo. Vyvolá zahrání tahu aktuálně hrajícího hráče z dolíku tohoto indexu (tedy dává smysl psát čísla jen z intervalu  $[0, 5]$ ).
- `a` – add. Příkaz `a` následovaný názvem souboru nechá za hráče, který je aktuálně na tahu, zahrát agenta z tohoto souboru. (Nadále pak u tohoto hráče stačí stisknout `enter` = napsat prázdnou řádku.)
- `↵` – prázdná řádka. Nechá zahrát agenta patřícího (přidaného pomocí `a`) k aktuálnímu hráči.
- `c` – cancel. Odebere agenta od hráče, který je aktuálně na tahu<sup>4</sup>.

---

<sup>3</sup>Lze si povšimnout, že na rozdíl od turnaje (sekce Turnaj) spuštěného s dvěma agenty, se odehraje jen jedna hra.

<sup>4</sup>Aktuálně asi nemá moc význam. V budoucnu je plánováno, že nebude potřeba stisknout `enter` (zadat prázdnou řádku), aby agent zahrál.

### 3 Grafické prostředí

Spouští se skriptem `graphics.py` a nemá žádné parametry. Po otevření okna lze buď přímo klikat na příslušné dolíky, nebo pomocí položky v menu **Přidat agenta** nechat vaše agenty automaticky hrát za jednoho (a tedy po dvou přidání za oba) hráče. V záložce **Odebrat agenta** se můžete jednoho hráče (resp. po dvou kliknutí i obou hráčů) chopit zase sami. Nakonec zde můžete spustit novou hru (agenti zůstanou) v menu **Hra**.

### 4 Turnaj

Turnaj se spouští skriptem `tournament.py` a jako parametry bere libovolný počet souborů s agenty. Pokud navíc dostane jako první argument číslo, provede mezi každou dvojicí agentů tolik zápasů, tj. dvě hry s prohozenými stranami. Pokud není první argument číslo, tak každý s každým hraje jeden zápas. Volání tedy může vypadat například takto:

```
python tournament.py agents/nahodny.py agents/nejlepsi_po_mem.py
```

Kdy se odehrají dvě hry mezi `nahodny.py` a `nejlepsi_po_mem.py` (jednou začíná první agent, podruhé druhý). A vypíše se něco jako:

Odehrána hra 1/2.

Hra `agents/nahodny.py` vs. `agents/nejlepsi_po_mem.py` skončila 7:26

Odehrána hra 2/2.

Hra `agents/nejlepsi_po_mem.py` vs. `agents/nahodny.py` skončila 26:0

```
2: Traianus (agents/nejlepsi_po_mem.py)
-2: Alexandr NeVeliký (agents/nahodny.py)
```

Nebo může volání vypadat takto<sup>5</sup>:

```
python tournament.py 5 agents/nahodny.py
→ agents/nejlepsi_po_mem.py agents/spravny_nahodny.py
```

Kdy se odehraje 30 her, každý z každým odehraje  $5 \times 2$  hry.

---

<sup>5</sup>Možná drobné upozornění, pokud vaši agenti využívají celou sekundu, tak tento příklad může trvat až  $5 \cdot 2 \cdot 180 = 1800$  sekund, tedy půl hodiny.

## 5 Drobnosti

Na tom, že už nelze sebrat žádné semínko, se agent domlouvat nemůže. V turnaji a `text.py -s` je to řešené tak, že hra se po 90 kolech (180 tazích) ukončí a pro kontrolu se vypíše, jak dopadla. Pokud byste se setkávali s tím, že v takové hře ještě lze sebrat semínko, ozvěte se.

Pokud si ukládáte nějaká větší data (např. pole) do dalších tahů, dávejte si pozor, že (jelikož váš agent je po jedné sekundě ukončen) můžete být uprostřed zápisu přerušeni. Jedním z možných řešení je mít 2 pole, kde zapisujete vždy do jednoho a o druhém si poznamenáte, že jsou tam neporušená data, kdybyste byli uprostřed zápisu do prvního pole přerušeni.