# Dokumentace druhé semestrální práce

#### Vybraná úloha:

Více-kolová hra Kámen-nůžky-papír pro n hráčů s využitím genetického algoritmu

## Vysvětlení modelu:

Model je vytvořen v programovacím jazyce python. Python je lepší programovací jazyk než netlogo pro tento typ úlohy, kde grafický výstup není tak důležitý ale je kladen větší důraz na samotnou logiku problematiky.

Model simuluje hru kámen-nůžky-papír pro několik hráčů. Hra kámen nůžky papír je velmi známá, pro kompletnost je zde vysvětlena její přesná implementace. V jednom kole má hráč možnost zvolit si ze tří různých možností. Volí si zda zahraje kámen, nůžky nebo papír. Při porovnání s dalším hráčem vyhrává ten hráč, který zvolil v porovnání silnější možnost. Kámen vyhrává nad nůžkami, nůžky vyhrávají nad papírem, papír vyhrává nad kamenem. Výhernímu hráči je jeho score zvýšeno o 1, jeho protivníkovi je o 1 sníženo. Pokud zahrají stejnou možnost, je to remíza a score hráčů se nemění.

Model je chopen simulovat více-kolovou hru, kdy hráči hrají více kol za sebou. Stejně tak je v modelu možno simulovat více jak 2 hráče, kdy každý hraje s každým jiným hráčem pouze jednou v daném kole.

Posledním důležitým aspektem modelu je genetický algoritmus. Genetický algoritmus je optimalizační technika, která se inspiruje procesem evoluce v přírodě. Využívá principy genetiky k řešení optimalizačních problémů. V tomto modelu je použit pro hledání nejlepšího řešení.

Genetický algoritmus se v tomto modelu skládá z několika kroků:

1. Inicializace hráčů: Na začátku se vytvoří požadovaná množina hráčů s náhodně vybranými strategiemi chování.
2. Hodnocení populace: Po dohrání kola jsou hráči hodnoceni na základě jejich score.
3. Eliminace nejhorších: Nejhůře zhodnocení hráči jsou odstraněni z celkové populace.
4. Zrod potomků: Z náhodně vybraných hráčů jsou vytvářeni jejich potomci dokud se nedosáhne požadovaného počtu hráčů.
5. Mutace: S určitou pravděpodobností se strategie nově vytvořených potomků změní. To je z důvodu zajištění dostatečné rozmanitosti v populaci.
6. Další kolo: Takto nově vytvořená populace hráčů je připravena hrát další kolo a začít znovu od kroku 2.

## Ovládání:

Model je ovládán pomocí globálních parametrů. Těmi jsou:

* n\_of\_players
* num\_of\_round
* percentage\_to\_remove
* mutation\_chance

Složitější ovládání s potřebou programátorské znalosti je pak možné přes:

* rules
* Strategy

n\_of\_players

Počet hráčů ve hře, je nutné aby tento počet byl minimálně 2.

#### num\_of\_round

Tento parametr udává, kolik kol budou hráči hrát. Minimální počet je tedy 1 pro jednokolovou hru.

#### percentage\_to\_remove

Parametr udávající, kolik procent populace je každé kolo vyřazeno v evolučním algoritmu. Samozřejmě se vyřazují nejhůře kvalifikovaní jedinci. Tento parametr je v rozmezí 0 a 1.

#### mutation\_chance

Parametr udávající šanci mutace při doplňování jedinců do plného počtu. S touto šancí jedinec nebude mít strategii rodiče ale zvolí si strategii náhodně. Tento parametr je v rozmezí 0 a 1.

#### rules

Pravidla pro chování a vyhodnocování výhry hráčů. Příkladem je klasická verze hry, kdy kámen vítězí nad nůžkami, nůžky vítězí nad papírem a papír vítězí nad kamenem. Není však problém vytvořit složitější chování nebo také rozšířenější verze hry jako třeba RST11 nebo dokonce RST25. Samozřejmě je pak nutné přidat požadované akce do Actions v souboru Stretegy.py.

Příklad:

rules = {

Action.Rock: [Action.Scissors],

Action.Paper: [Action.Rock],

Action.Scissors: [Action.Paper]

}

#### Strategie:

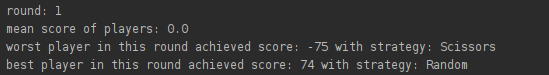
Strategie je oddělená od klasického ovládání, jelikož změny zde vyžadují znalost programování. Strategie se nachází v souboru Strategy.py, kde jsou pro testování modelu implementovány 4 možné strategie. Strategie hrající vždy kámen, strategie hrající vždy papír, strategie hrající vždy nůžka a strategie, která náhodně hraje jednu ze všech možností.

## Výstup z modelu:

Model v každém kole vypisuje 3 základní údaje. Vypisuje hráče s nejvyšším skore, hráče s nejnižším score a průměrné score všech hráčů. Výpisem hráče je myšleno vypsání nejen jeho score ( které je tedy nejvyšší a nejnižší ve hře) ale také jeho strategii.

Příklad jednoho kola výpisu pro 100 hráčů vypadá takto:

Experiment:



Jako experiment bylo zvoleno zkoumání změn v modelu se zvyšujícím se počtem hráčů. Testovány byli verze modelu s 10, 100 a 1000 hráčů. Z výsledků je patrné, že větší počet hráčů vede k vyšším výkyvům průměrného score a k rychlejšímu konvergování k ideální strategii.