

# Vývojářská dokumentace k programu **Predator & Prey Simulation**

Ondřej Kučera  
Matematicko-fyzikální fakulta  
Předmět: Programování 2

8. září 2025

## Obsah

<b>1</b>	<b>Stručné zadání</b>	<b>2</b>
<b>2</b>	<b>Přesné zadání</b>	<b>2</b>
<b>3</b>	<b>Použité algoritmy</b>	<b>2</b>
3.1	Simulační smyčka . . . . .	2
3.2	Pohyb organismů . . . . .	3
3.3	Rozmnožování . . . . .	3
3.4	Úmrtí . . . . .	3
<b>4</b>	<b>Diskuse volby algoritmů</b>	<b>3</b>
<b>5</b>	<b>Struktura programu</b>	<b>4</b>
5.1	Datové struktury . . . . .	4
5.2	Hlavní třídy . . . . .	4
<b>6</b>	<b>Alternativní řešení</b>	<b>4</b>
<b>7</b>	<b>Průběh prací</b>	<b>4</b>
<b>8</b>	<b>Co nebylo dokončeno</b>	<b>5</b>
<b>9</b>	<b>Závěrečný komentář</b>	<b>5</b>

## 1 Stručné zadání

Cílem programu je simulovat jednoduchý ekosystém složený z kořisti, predátorů a překážek. Organismy se pohybují po čtvercové mřížce a jejich chování je určeno několika biologicky motivovanými pravidly. Výsledkem simulace je vizuální zobrazení aktuálního stavu mřížky a graf vývoje počtu predátorů a kořisti v čase.

## 2 Přesné zadání

- Simulace probíhá v diskrétních časových krocích.
- Mřížka je čtvercová, s uživatelem definovanou velikostí.
- Na začátku se na mřížce náhodně rozmístí kořist, predátoři a překážky.
- Kořist se náhodně pohybuje a periodicky se rozmnožuje.
- Predátoři hledají nejbližší kořist v určitém dosahu; pokud ji najdou, snaží se k ní přiblížit.
- Predátoři získávají energii z potravy (kořisti). Pokud jim energie dojde, zemřou. Pokud dosáhne určité hranice, mohou se rozmnožit.
- Překážky blokují pohyb, ale samy se nemění.
- Uživatel má možnost simulaci spustit, pozastavit, krokovat, měnit parametry a ukládat/načítat stav.

## 3 Použité algoritmy

### 3.1 Simulační smyčka

Hlavní metoda `Step()` provede jeden krok simulace:

1. Projde všechny organismy a zavolá jejich metodu `Update()`.
2. `Update()` provádí chování specifické pro daný typ organismu (pohyb, hledání potravy, rozmnožování).
3. Nově vzniklí jedinci jsou přidáni na konec seznamu organismů.
4. Inkrementace počítadla kroků.

### 3.2 Pohyb organismů

- Kořist se pohybuje náhodně jedním ze čtyř směrů (nahoru, dolů, vlevo, vpravo).
- Predátor se snaží přiblížit k poslední známé pozici kořisti. Pokud žádnou nevidí, pohybuje se náhodně.
- Pohyb je možný pouze do volného pole v mřížce.

### 3.3 Rozmnožování

- Kořist: po dosažení konfigurovaného počtu kroků od narození se pokusí vytvořit nového jedince v sousedním poli.
- Predátor: pokud energie dosáhne určité hranice, pokusí se vytvořit potomka; energie se rozdělí mezi rodiče a dítě.

### 3.4 Úmrtí

- Kořist může zemřít při přemnožení (pokud počet sousedních kořistí překročí limit).
- Predátor zemře, pokud mu dojde energie.

## 4 Diskuse volby algoritmů

Uvažoval jsem více variant:

- **Pohyb kořisti:** Mohla by se snažit vyhýbat predátorům. Pro jednoduchost a přehlednost jsem zvolil čistě náhodný pohyb.
- **Hledání potravy predátorem:** Mohl být použit A\* nebo BFS algoritmus, ale to by bylo výpočetně drahé. Proto je použita jednodušší strategie: predátor detekuje nejbližší kořist v manhattanské vzdálenosti a pohybuje se směrem k ní.
- **Rozmnožování:** Mohl být zvolen složitější genetický model (dědičnost vlastností). Bylo zjednodušeno na pevně daný interval/limit.

## 5 Struktura programu

### 5.1 Datové struktury

- **Grid** – dvourozměrné pole **Cells**, které uchovává organismy.
- **Organism** – abstraktní třída, společný základ všech organismů.
- **Prey, Predator, Obstacle** – konkrétní typy organismů.
- **SimulationConfig** – konfigurační parametry.
- **SimulationData** – pomocná datová struktura pro ukládání stavu do JSON.

### 5.2 Hlavní třídy

**Simulation** – řídí běh simulace, drží mřížku, provádí kroky.

**MainForm** – GUI, umožňuje ovládat simulaci.

**StatisticsForm** – zobrazování grafu vývoje populací.

**Persistence** – ukládání/načítání simulace ve formátu JSON.

## 6 Alternativní řešení

- **Statistiky:** Mohly být vykreslovány přímo do panelu, ale `DataVisualization.Charting` je pohodlnější a čitelnější.
- **Parallelizace:** Update organismů by mohl běžet paralelně, ale vzhledem k malým datovým sadám to není nutné.

## 7 Průběh prací

1. Analýza zadání a příprava základní struktury (**Grid**, **Organism**).
2. Implementace **Prey** a **Predator**, základní chování.
3. Přidání GUI a možností ovládání.
4. Implementace ukládání/načítání.
5. Přidání statistik a parametrů do **SimulationConfig**.
6. RefaktORIZACE a přidání unit testů.

## 8 Co nebylo dokončeno

- Zatím neexistuje možnost měnit parametry během běhu (vyžaduje reset).
- Nepřidána vizualizace překážek jiným tvarem (momentálně černý čtverec).
- Chybí optimalizace vykreslování pro opravdu velké mřížky.

## 9 Závěrečný komentář

Práce na programu byla poučná – propojila znalosti z programování, algoritmů i uživatelského rozhraní. Hlavní problém byl návrh tříd tak, aby zůstaly čitelné a udržitelné. Pokud bych měl projekt rozšiřovat, zaměřil bych se na vylepšení UI a přidání dalších biologicky realistických mechanismů.