Áttekintés

Ez a projekt egy egyszerű ökoszisztéma-szimulációt valósít meg, amelyben három fő elem szerepel: fű (Fu), nyulak (Nyul), és rókák (Roka). A fű különböző szinteken nő, a nyulak fűvel táplálkoznak, a rókák pedig nyulakat esznek. A program célja a populációk interakcióinak szimulálása.

Fájlok és Osztályok

A program különböző fájlokban szervezett osztályokból épül fel. Az alábbiakban részletesen ismertetjük az egyes fájlokat és az osztályokat.

Fu.cs

*Osztály: Fu*

Ez az osztály reprezentálja a füvet, amely három különböző szinten létezhet. A fű szintje az osztály inicializálásakor véletlenszerűen dől el.

Tulajdonságok:

* Szint: A fű aktuális szintjét tárolja (1-től 3-ig terjedhet).

Konstruktor:

* A konstruktor véletlenszerűen beállítja a fű szintjét az 1, 2 vagy 3 értékekre.

Módszerek:

* FuSzintNoveles(): Növeli a fű szintjét 1-ről 2-re vagy 2-ről 3-ra.
* FuKarakterek(): Visszaadja a fű szintjének megfelelő karaktert: '\_' (üres), 'g' (növekvő fű), 'G' (kifejlett fű).

Mezo.cs

*Osztály: Mezo*

Ez az osztály kezeli a szimuláció fő területét, amely egy 10x10-es rácsot képvisel. A mezőn füvek, nyulak és rókák helyezkednek el.

Tulajdonságok:

* Szelesseg és Magassag: A mező méretei, mindkettő értéke 10.
* mezo: Egy 10x10-es kétdimenziós tömb, amely a füvek elhelyezkedését tárolja.
* nyulak: Nyulak listája.
* rokak: Rókák listája.

Konstruktor:

* Inicializálja a mezőt, és véletlenszerűen helyez el 3-3 nyulat és rókát.

Módszerek:

* KoviMezo(): Ez a fő logikai ciklus, amely minden "lépésben" frissíti a mezőt:
* Füvek szintjének növelése.
* Nyulak mozgása és táplálkozása.
* Nyulak és rókák közötti interakciók.
* Új nyulak és rókák születése a szomszédos állatok találkozásakor.
* Kiiras(): Megjeleníti a mezőt a konzolon. A füvek és állatok különböző karakterekkel jelennek meg.
* SzovegKiiro(): Egy sor középre igazított szöveg kiírása a konzolon.

Különböző segédfüggvények, mint például a legközelebbi fű vagy nyúl keresése, állatok mozgatása.

Nyul.cs

*Osztály: Nyul*

A nyulakat reprezentáló osztály. A nyulak a füvekkel táplálkoznak, és minden egyes lépésnél csökken az élelemkészletük, amit Jollakottsag tárol.

Tulajdonságok:

* X és Y: A nyúl aktuális pozíciója.
* Jollakottsag: Az élelem szintje, alapértéke 5.

Konstruktor:

* Beállítja a nyúl kezdő pozícióját.

Módszerek:

* Mozgas(): A nyulat egy új pozícióra mozgatja.
* FuEves(): Ha van fű a nyúl pozícióján, megnöveli a nyúl Jollakottsag értékét, és eltünteti a füvet a mezőről.

Roka.cs

*Osztály: Roka*

A rókák viselkedését kezelő osztály. A rókák a nyulakra vadásznak, és minden lépésnél csökken a Jollakottsag értékük.

Tulajdonságok:

* X és Y: A róka aktuális pozíciója.
* Jollakottsag: Az élelem szintje, alapértéke 10.

Konstruktor:

* Beállítja a róka kezdő pozícióját.

Módszerek:

* Mozgas(): A rókát egy új pozícióra mozgatja.
* NyulMegevese(): Ha a róka megeszik egy nyulat, megnöveli a Jollakottsag értékét.

Program.cs

Ez a program belépési pontja, amely elindítja a játékot.

A program először egy főmenüt jelenít meg, ahol a felhasználó három lehetőség közül választhat:

1. Start: Elindítja a játékot.
2. Tutorial: Kiírja a játékról szóló leírást és útmutatót.
3. Exit: Kilép a programból.

A játék elindítása után a mező megjelenik a konzolon, és 2 másodpercenként frissül a KoviMezo() hívásával, amely a szimulációt egy lépéssel előreviszi.

Összegzés

A játék a populációs ökoszisztéma modelljére épül, ahol nyulak és rókák lépnek interakcióba egy rácsalapú világban, miközben füvekkel és egymással táplálkoznak. A program egyszerű, de jól szemlélteti az alapvető szimulációs mechanizmusokat, például a növekedést, mozgást és táplálkozást.