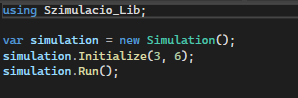
**Fejlesztői dokumentáció**

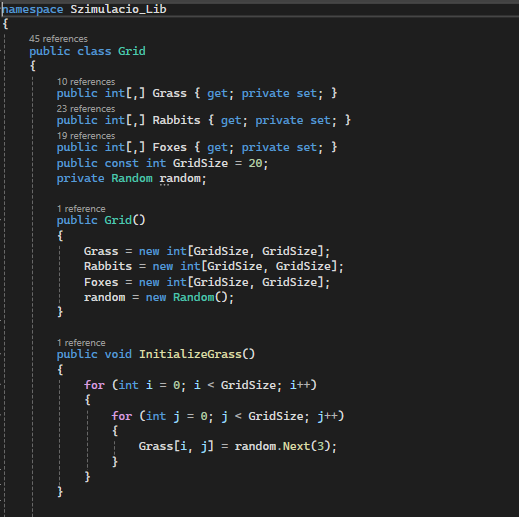
A program egy szimuláció amely egy ökológiai rendszert szimulál rókák és nyulak között egy 20x20-as rácson. A kód főbb részei a következők: *Program.cs, Grid.cs, Fox.cs, Rabbit.cs* és *Simulation.cs*.

**Program.cs:**

A *Program.cs* tartalmazza a szimuláció futtatásához szükséges alapvető logikát:

Itt létrejön egy *Simulation* objektum, ami inicializálva van 3 nyúllal és 6 rókával, majd elindítja a szimulációt.

**Grid.cs:**

A *Grid.cs* a környezetet modellezi, amelyben a rókák és nyulak élnek. Ez a komponens felelős a rács megjelenítéséért és frissítéséért.

* **GridSize**: A rács mérete 20x20.
* **Grass, Rabbits, Foxes**: Három kétdimenziós tömb (20x20 méretű), amelyek a fű, a nyulak és a rókák elhelyezkedését tárolják.

Főbb metódusok:

* *InitializeGrass*():  
  A füvet inicializálja a rácson, ami 0-2 közötti értékeket ad.
* *UpdateGrass*():  
  Növeli a fű növekedését, ha az kevesebb mint 2.
* *PrintGrid*():   
  Megjeleníti a rács állapotát, ahol különböző színek és szimbólumok jelzik a nyulakat, rókákat és füvet.

**Fox.cs**

A *Fox.cs* a rókák viselkedését írja le a rácson. A rókák célja a nyulak vadászata és saját túlélésük biztosítása.

* **maxFoxSatiety**: A rókák maximális éhsége (30).
* **Place()**: Egy róka elhelyezése a rácson véletlenszerű jóllakottsággal.
* **Update()**: Frissíti a rókák állapotát. Ellenőrzi, hogy találnak-e nyulat, különben csökkenti a éhségüket, és mozgatja őket.
* **CheckForRabbits()**: Ellenőrzi, hogy van-e nyúl a környéken, és ha van, a róka megeszi.
* **Move()**: Véletlenszerű irányba mozgatja a rókát, ha a mező üres.
* **Reproduce()**: A rókák szaporodnak, ha két róka közel van egymáshoz.

### 

### 

### 

### 

### 

### 

### 

### 

### 