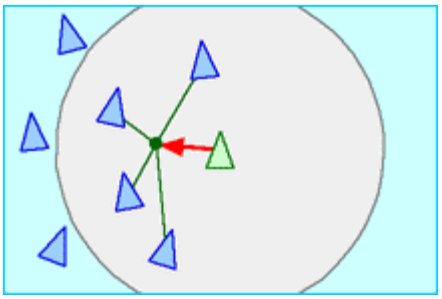
Párhuzamos Programozás
féléves feladat

Kremzner István, V0CMMQ

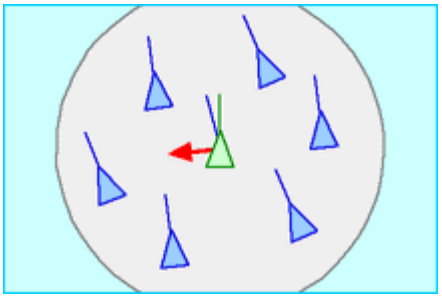
Flocking simulation

A féléves feladatom témája a Flocking algorithm megvalósítása és vizuális reprezentációja Unity segítségével, nagy elemszámú esetben. (1000 körüli).

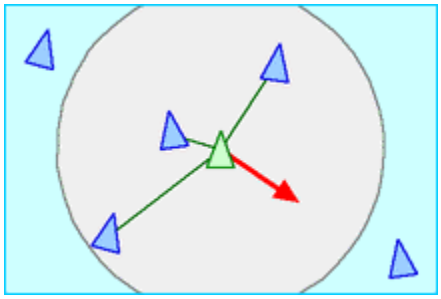
A flocking algoritmusnak 3 dolognak kell megfelelnie:



Kohézió:
A falka tagjaninak átlagos pozíciója felé kell irányítania az egyednek magát.



Igazodás:
A falka tagjainak átlagos iránya felé kell fordulnia az egyednek.

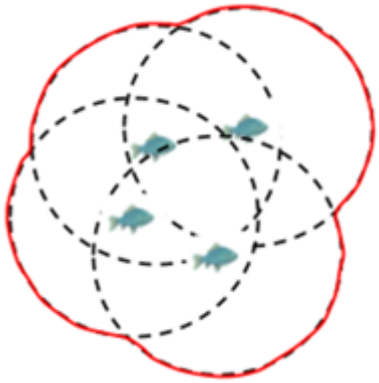


Szeparáció:
A falkán belüli zsúfolódás elkerülése.

Az algoritmust a madarak vagy a halak egy rajának szimulációjaként tekinthető.

Jelen esetben tekintsük őket halaknak.

Egy falka egy raj halat jelent, jelen esetben.
Minden hal, amely egy falka határain belül van, egy megadott távolságon belülre van egy másik haltól, nem egyedüli halként viselkedik, hanem a falka egy tagjaként. A tagoknak egyezik a sebességük és az irányuk.



Az algoritmus minden halmozgásának frissítéséhez meg kell vizsgálnia az összes többi halat, hogy tudja a vizsgált halról, hogy a rajban maradhat-e, el kell-e hagynia azt, illetve, hogy csatlakozhat-e egy újhoz.

Ha egy hal egyedül van, egy meghatározott sebeséggel és iránnyal mozog, viszont, ha szomszédja van, alkalmazkodnia kell az ő sebességéhez és irányához.

Azoknak a halak, amelyek szomszédoknak számítanak távolság alapján, nagyobb esélyük van egy falkát alkotni, ha megegyezik a haladási irányuk.

A halaknak közvetlen hozzáférése van az egész terület geometria leírásához, de csak a saját falkája tagjaira reagál, akik egy meghatározott szomszédságon belül vannak. A szomszédság egy sugárból és egy szögből áll össze, amelyet a haladási iránytól mérünk.

A halak meghatározott területen belül úszkálnak, határokkal rendelkeznek. Ha az egyik túl akarna menni rajta, kap egy új random irányt. Illetve lesznek még akadályok is, kövek, az ezekkel való ütközés elkerülése is hasonló módon történik, mint a falaké.

A programban a kohézió, igazodás és szeparáció egyenként ki- és bekapcsolható lesz.

A Unityben való implementáláshoz a Simplistic Low Poly Nature nevű Asset package-et fogom használni. Az ebben található halat, és az ebben található köveket, akadályként.

A halak új irányának a és sebességének számítását végző függvény a legidőigényesebb művelet, minden halra meghívódik, összesen n-szer hívódik meg.
Ezt lenne érdemes párhuzamosítani, szétbontani különböző részekre, Task-okra.