

Programozás alapjai 2

NHF Specifikáció(Bővített)

Fótyék Róbert

DNHR9M

Leírás: A program egy részecskék közötti kölcsönhatásokat modellező egyszerű, kétdimenziós szimuláció. A kezdeti állapot beolvasás után képes tovább léptetni a szimulációt tetszőleges időtartamig. A nagyházi követelményeit figyelembe véve *nem* lesz grafikus felülete, a kimenete a részecskék pillanatnyi tulajdonságai, melyet a program egy fájlba ír.

I/O: A program szimuláció közben minden időegységben egy, az alábbi formátumú fájlba írja a részecskék adatait (ez lehetővé teszi egy grafikus "visszajátszó" program elkészítését). A bemenet egy ugyanilyen formátumú fájl, és a program a fájlban található utolsó időegységről folytatja a szimulációt, tovább írva a fájlt.

Feature-ök: egy részecske deklarálásakor mindig megadandó a tömege, a töltése, illetve beállítható hogy részt vesz-e a gravitációs kölcsönhatásban. Ezen konstansokon kívül beállítható kezdetleges pozíciója és koordinátái. Ilyen részecskékből tetszőleges számú megadható. Ez a kezdeti állapot megadható fájlban, vagy a program CLI-jén keresztül.

Fájl formátum:

{

Tömeg,

PozícióX PozícióY,

SebességX SebességY,

Töltés,

GravitációsBoolean,

Tömeg,

PozícióX PozícióY,

SebességX SebességY,

Töltés,

GravitációsBoolean,

...

...

...

}

{

...

}

.

.

.

Így egy bracket {} egy időegység, és azon belül a részecskék a szabvány alapján következnek egymás után. A sablonban az új sor csak a szépség miatt van, egyébként minden whitespace ugyanaz.

Program futtatása:

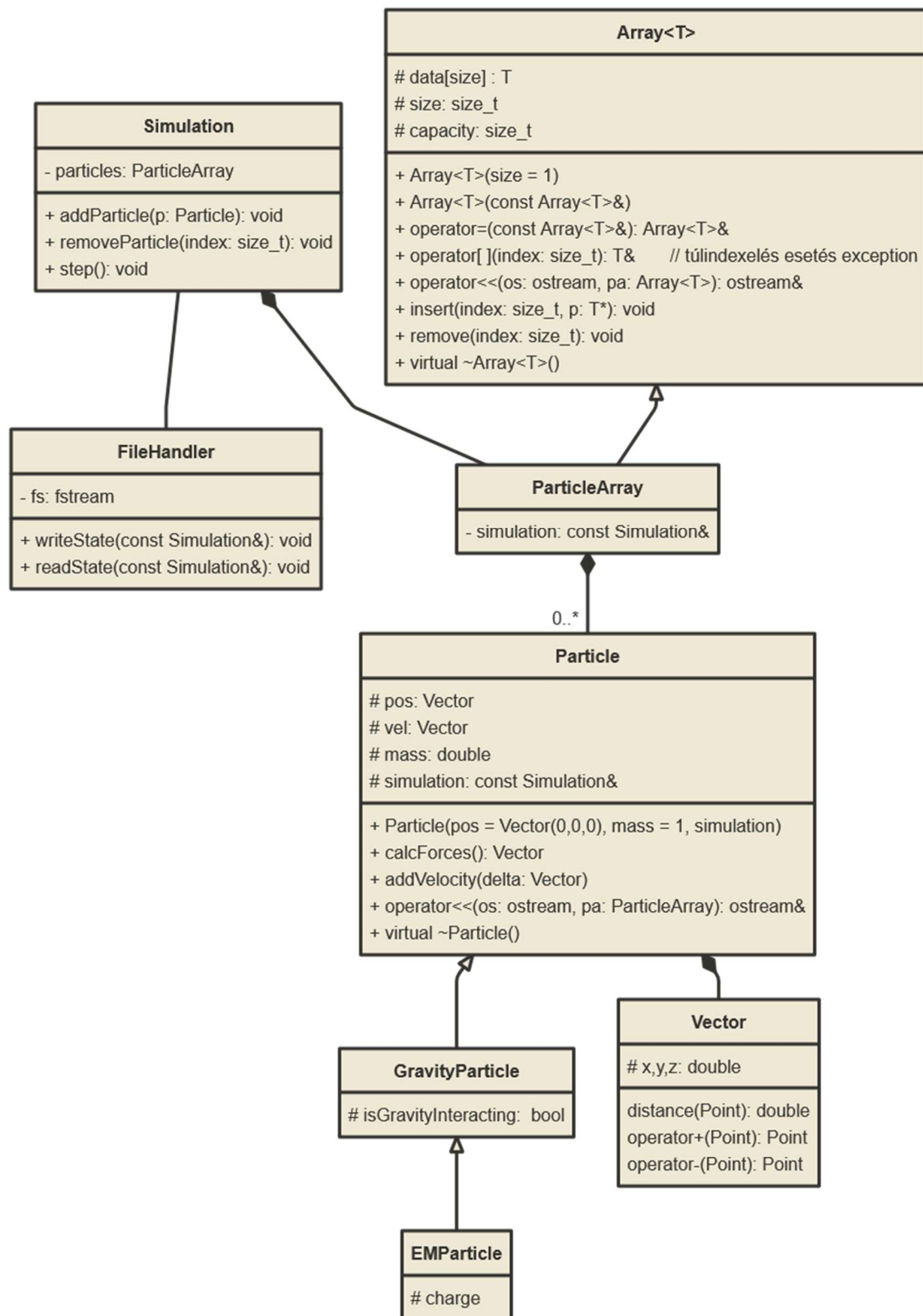
A program argumentumként megkapja az input fájlt, majd onnan menürendszerből kezelhető.

Menüpontok:

- ListParticles
- AddParticle
- RemoveParticle
- Simulate *időtartam*
- *Exit*

A szimulált időtartam után visszalép a program a menübe.

Osztálydiagram



Megjegyzések:

- A ParticleArray illetve Particle osztályoknak komponense a Simulation osztály, hogy ne lehessen részecske hozzá tartozó szimuláció nélkül.
- Accesorokat, triviális konstruktorokat, destruktorkat nem jelöltem.

Fő algoritmus: szimulációs lépés kiszámítása

```
For each Particle i in ParticleArray:  
    Vector force(0,0,0)  
    For each Particle j != i in ParticleArray:  
        Vector force += calculateEMforce(i, j)  
    i.addVelocity(force/i.mass())
```