

Írd át a hóesést úgy, hogy teljesítse a következő feltételeket:

- Framenként ne csak 1 hópelyhely jelenjen meg, hanem 3
- 1.5-szer gyorsabban essen a hó, mint eddig
- Legyen random a hópelyhek sugara (minimum 1, maximum 6)
- Köralakú hópelyhek essenek

Alakítsd át a kódot úgy, hogy legyen egy Szimuláció típusú hóesés objektum.

Ebben az objektumban lesz az összes hópelyhely, valamint minden művelem, ami az összes hópelyhet érinti.

Vagyis a következő tagfüggvények kellene a Szimuláció típusba:

- `uj_hopelyhek(int hszam)` Ez a függvény megadott számú új hópelyhet ad a szimulációhoz
- `leptet()` A már jól ismert `leptet()` tagfüggvénye a `Hópelyhely` típusnak meghívva az összes hópelyhelyre
- `kirajzol()` Ugyanaz, mint a `leptet` csak a `kirajzol()` függvényre. Plusz ide be lehet olvasztani a `torol()` függvényt is
- `tesztel()` Ez egy új feladat. Azt kéne elérni ennek a függvénynek a segítségével, hogy az Y irányban a képernyőről kieső hópelyhek törlődjenek a vektorból. Ehhez szükség lehet egy új pl. `y_rendben()` nevű tagfüggvényre a `Hópelyhely` típusba, ami megmondja, hogy kiesett-e az adott hópelyhely a képernyőről. A (`vnev` nevű) vektorból `i`-edik elemet a következő képpen lehet törölni (ezt majd később elmagyarázom, hogy mit jelent, most csak fogadjátok el):
`vnev.erase(vnev.begin()+i)`
Ezen kívül ennek a függvénynek az elején írasd ki a konzolra (`std::cout`), hogy hány hópelyhed éppen. Ha mindent jól csináltál, akkor amint leérnek az első hópelyhek, ennek az értéknek már nem nagyon kéne tovább nőnie (random fluktuáció persze belefér, ha random mennyiségű hópelyhet generálsz framenként)