Dalelių sistemos

T120B167 Žaidimų grafinių specialiųjų efektų kūrimas ir programavimas

Rytis Maskeliūnas
Skype: rytmask
Rytis.Maskeliunas@ktu.lt

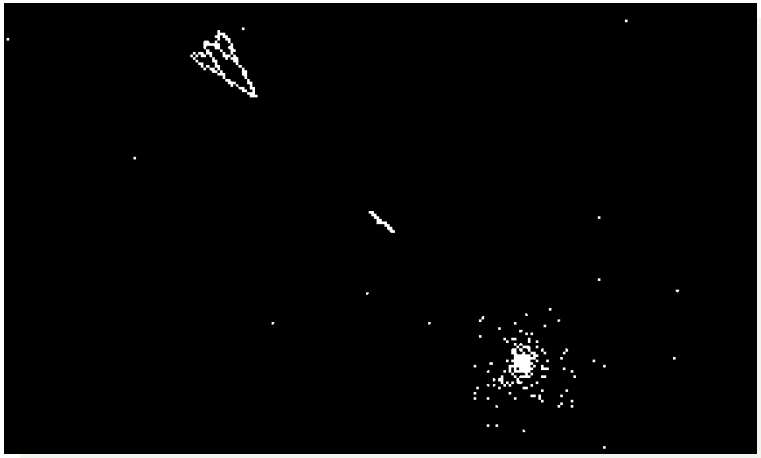
Paskaitos tema

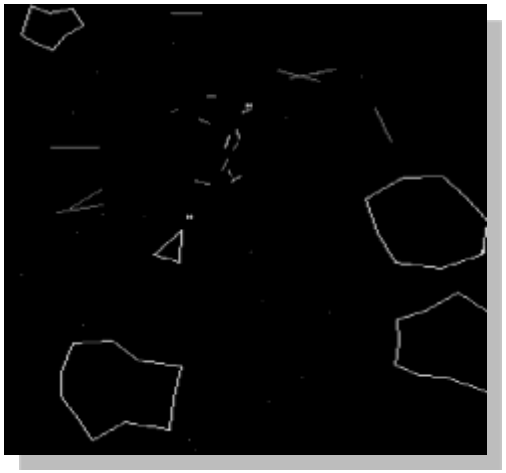
MULTIMEDIJOS INŽINERIJOS KATEDRA

http://www.youtube.com/watch?v=ACHJ2rGyP10

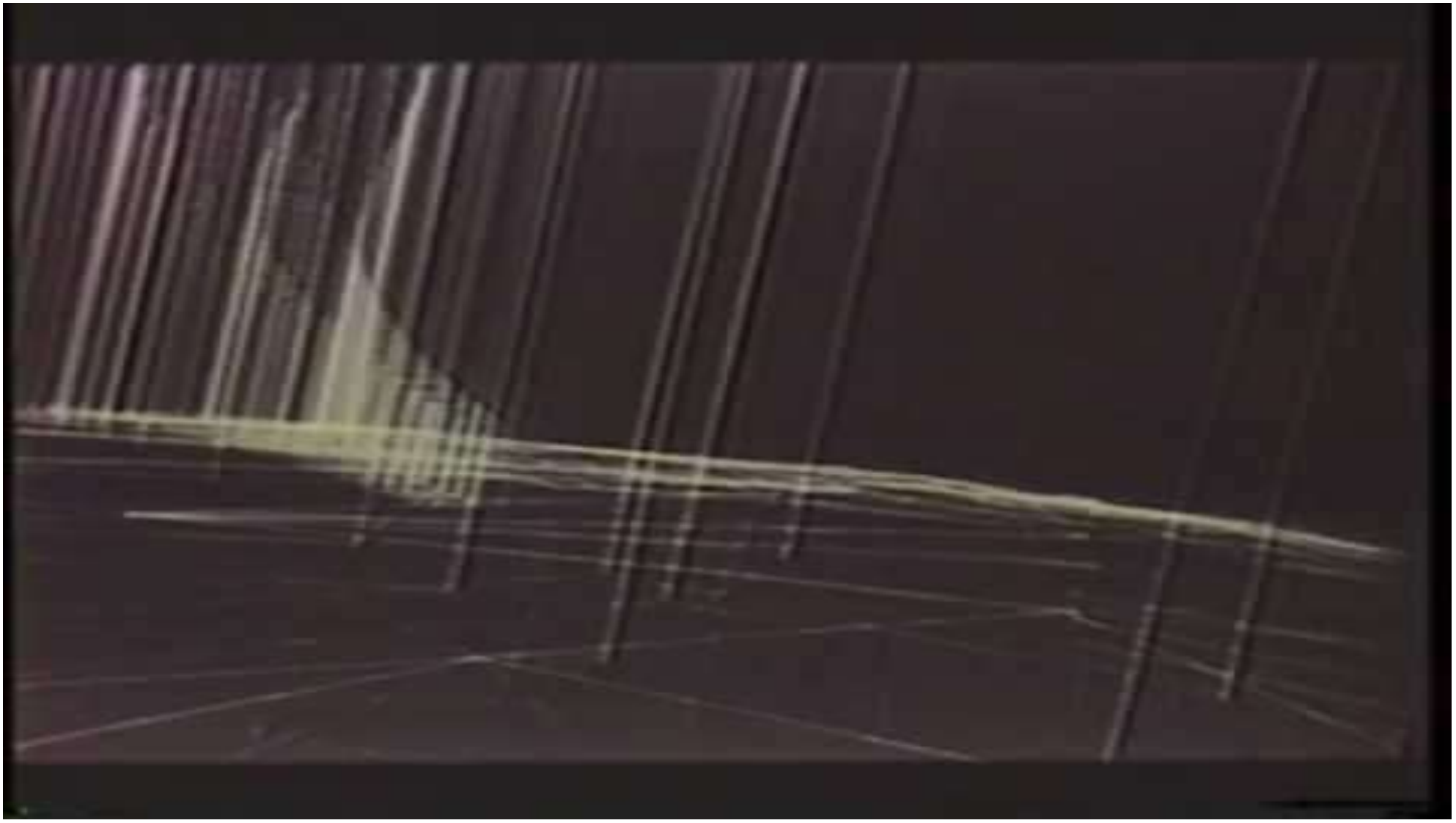
Istorija

- 1962: Ships explode into pixel clouds in “Spacewar!”, the 2nd video game *ever*.
- 1978: Ships explode into broken lines in “Asteroid”.
- 1982: The Genesis Effect in “*Star Trek II: The Wrath of Khan*”.





.“The Genesis Effect” – William Reeves
Star Trek II: The Wrath of Khan (1982)

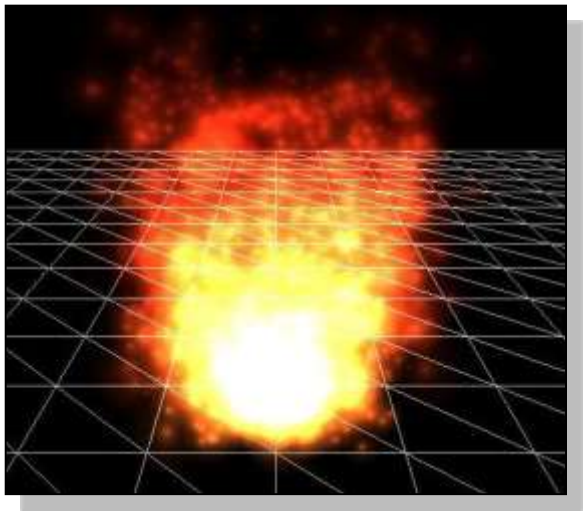


http://www.youtube.com/watch?v=Qe9qSLYK5q4

Dalelių sistemos

- *Dalelių sistemos* leidžia suimituoti vaizdo efektus su eile mažų grafinių dalelių.

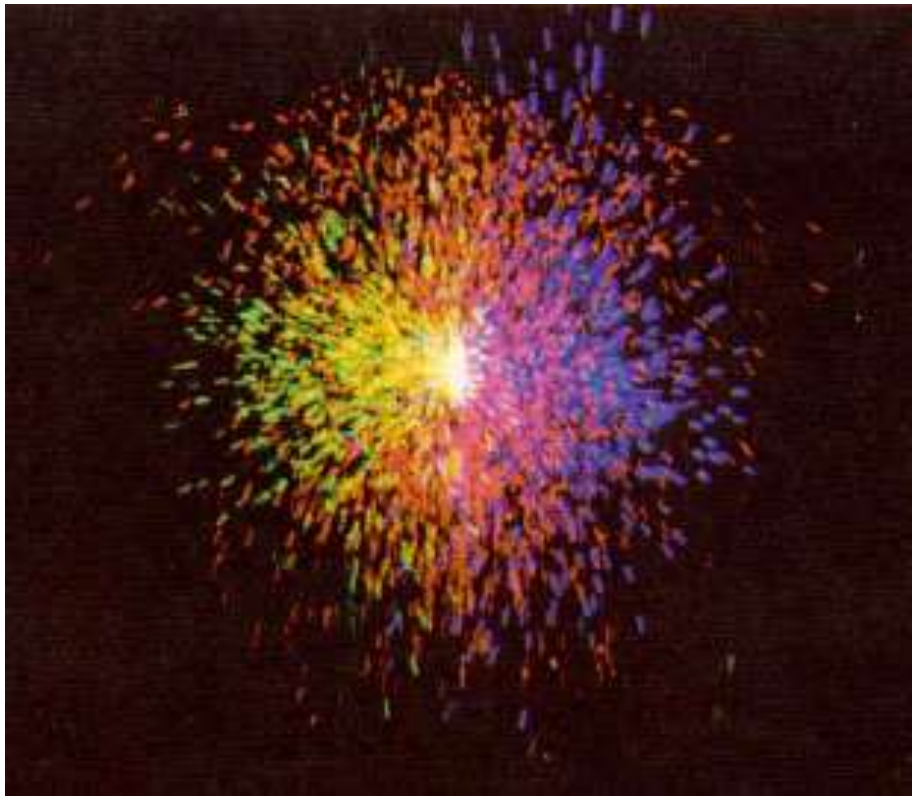
- **Idėja:**
  “jei daug mažų taškų darys kažką tuo pačiu būdu, žmogaus smegenys matys to darymo rezultatą, bet ne jį sudarančius taškus.”



Kas yra dalelių sistema?

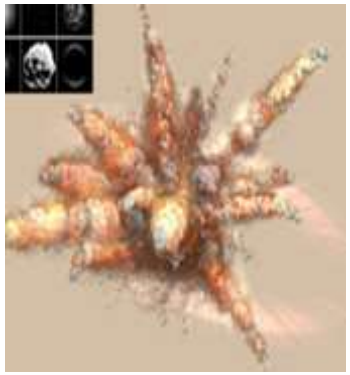
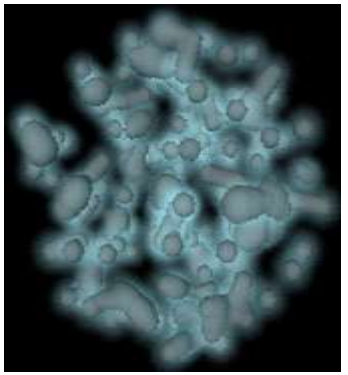
MULTIMEDIJOS INŽINERIJOS KATEDRA

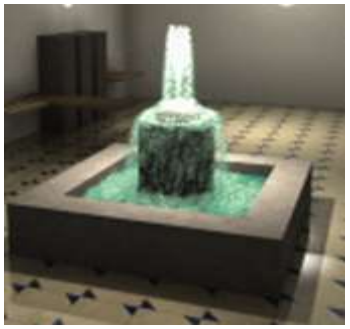
- Tai daugelio mažų dalelių rinkinys kuris atvaizduoja neraiškų objektą
- Formoms apibrėžti galima naudoti taškus arba primityvus



Taikymai?

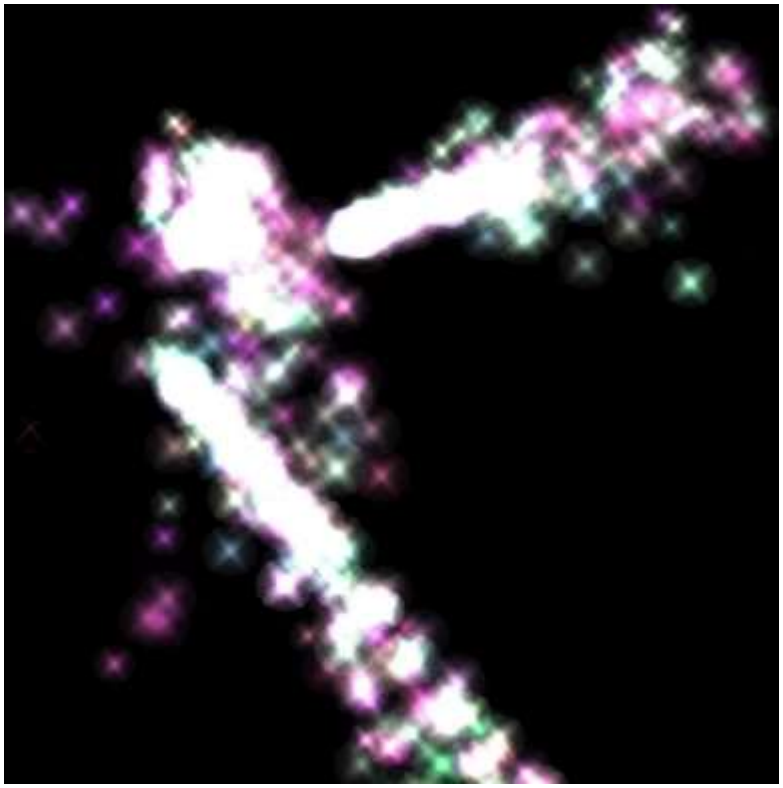
- Naudojamas neraiškių (fuzzy), anamorfinių (kintamų), dinaminių ir skystų objektų modeliavimui.
- Tai dūmai, ugnis, kibirkštys, lietus, sniegas, vandens lašai, sprogimai, žvaigždės ir t.t.

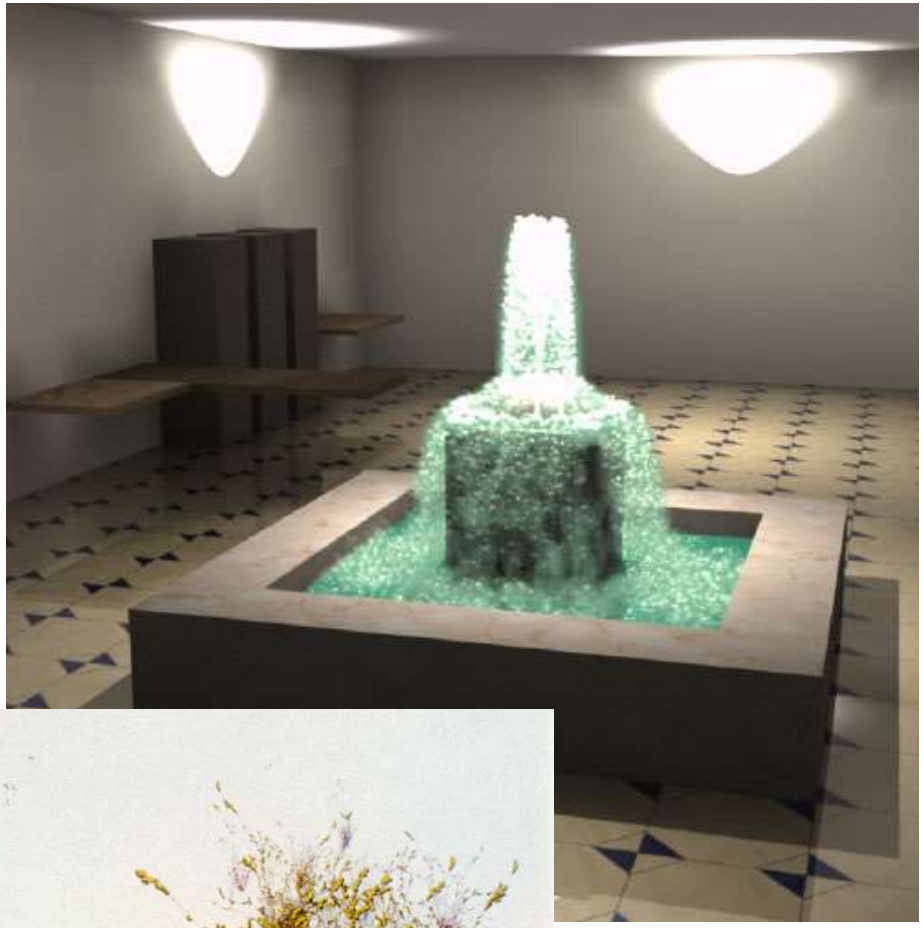
Neraiškus objektai

- Neturi tolydžių, aiškiai aprašomų ir blizgių paviršių
- Nevienodi, kompleksiški ir sunkiai apibrėžiami
- Minkšti, deformuojami objektai



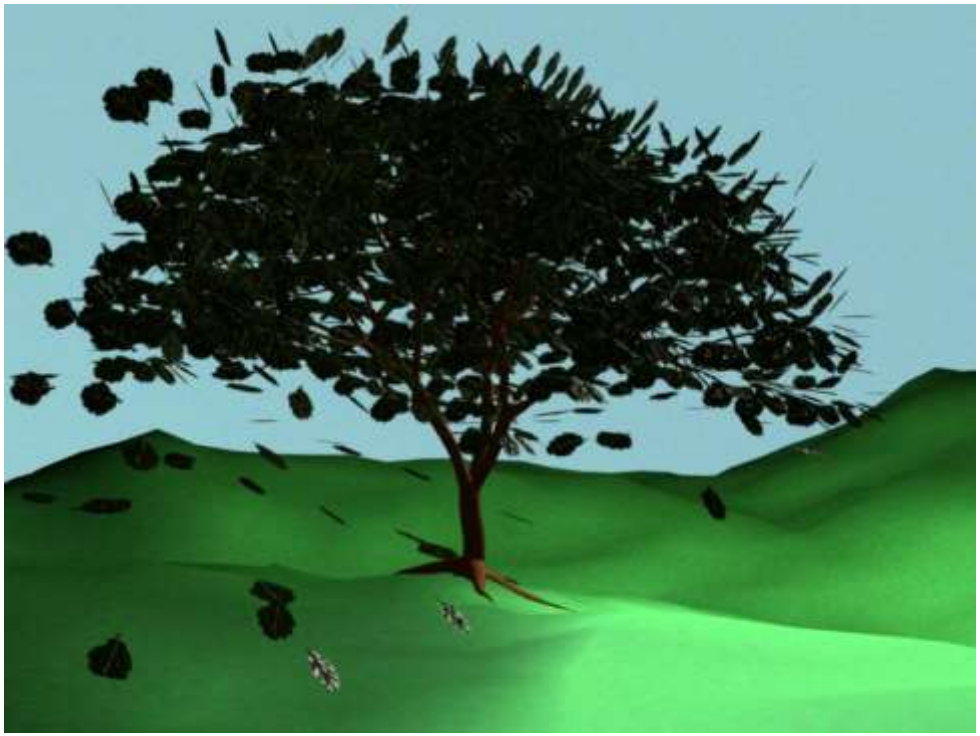
Neraiškių objektų pvz

- Žolė, dūmai, ugnis, debesys, vanduo
- Fejerverkai, sprogimai
- Tekantis skystis
- Fizikos imitacijai
- Būriai (paukčių, bičių, žuvų, zombių...)



Privalumai

- Paprasta – taškus apibrėžti lengviau nei 3D poligonus
- Procedūriška
- Random
- Modeliai „atgyja“



Gimk-auk-mirk

- Dalelės įvedamos į sistemą
- Joms priskiriama individualūs atributai
- Dalelės kurios parodomos savo gyvenimo laiką yra sunaikinamos
- Gyvos (rodomos) dalelės yra judinamos ir transformuojamos pagal savo atributus
- Viskas surenderiuojama

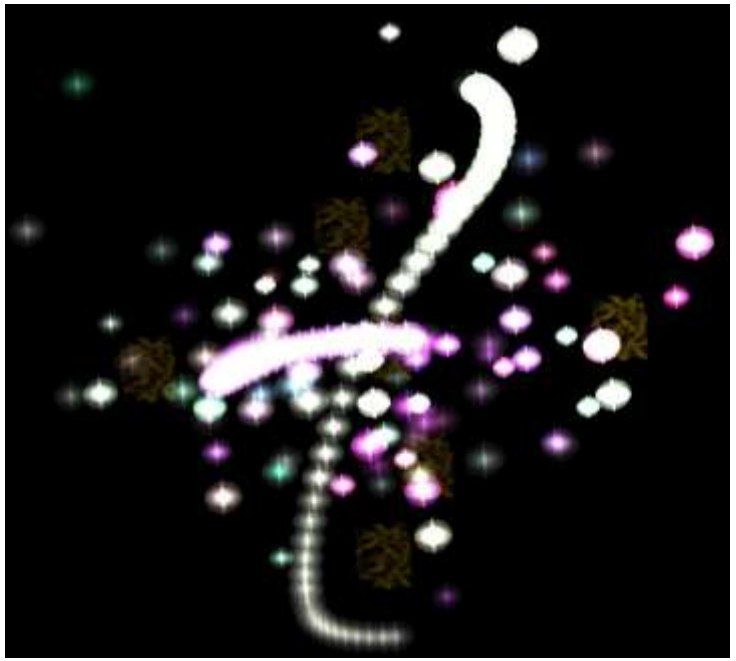
Gimimas

- Nustatomas santykis kokiu dalelės įvedamos į sistemą
  - Valdomas vidutinis dalelių skaičius (pvz., lietus)

    arba
  - Valdomas vidutinis dalelių skaičius užimantis tam tikrą ekrano dalį (pvz., ugniakuras)
- Keičiamas objekto dydis pagal tai kiek dalelių įvedama į sistemą (pvz., sprogimas)

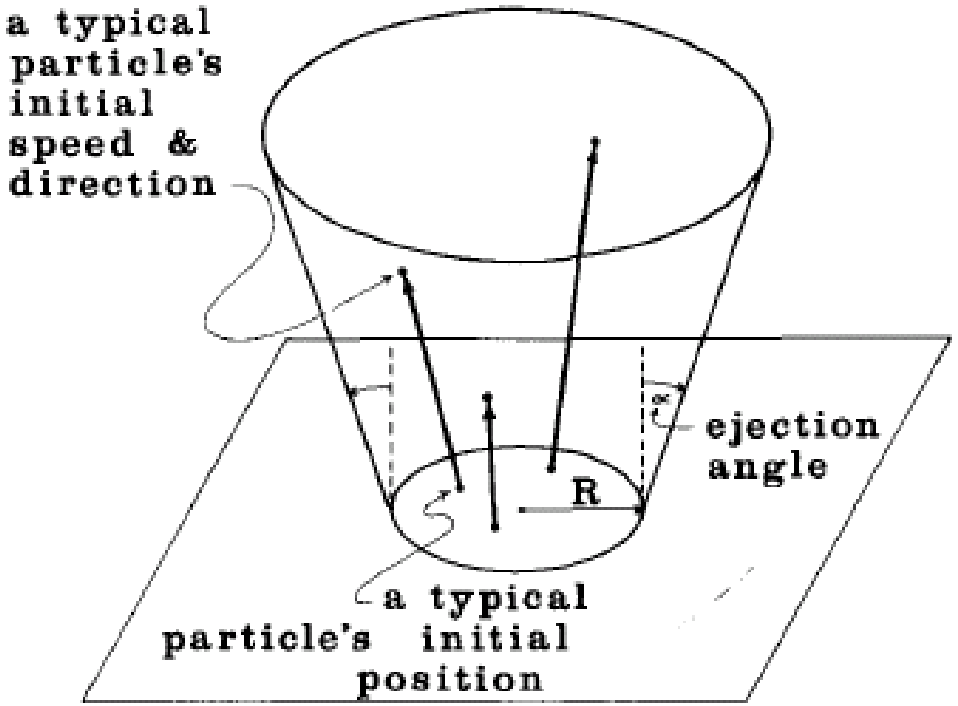
Dalelių atributai

- Pozicija
- Greičio vektorius
- Dydis, spalva, persišviečiamumas
- Forma
- Gyvenimo trukmė



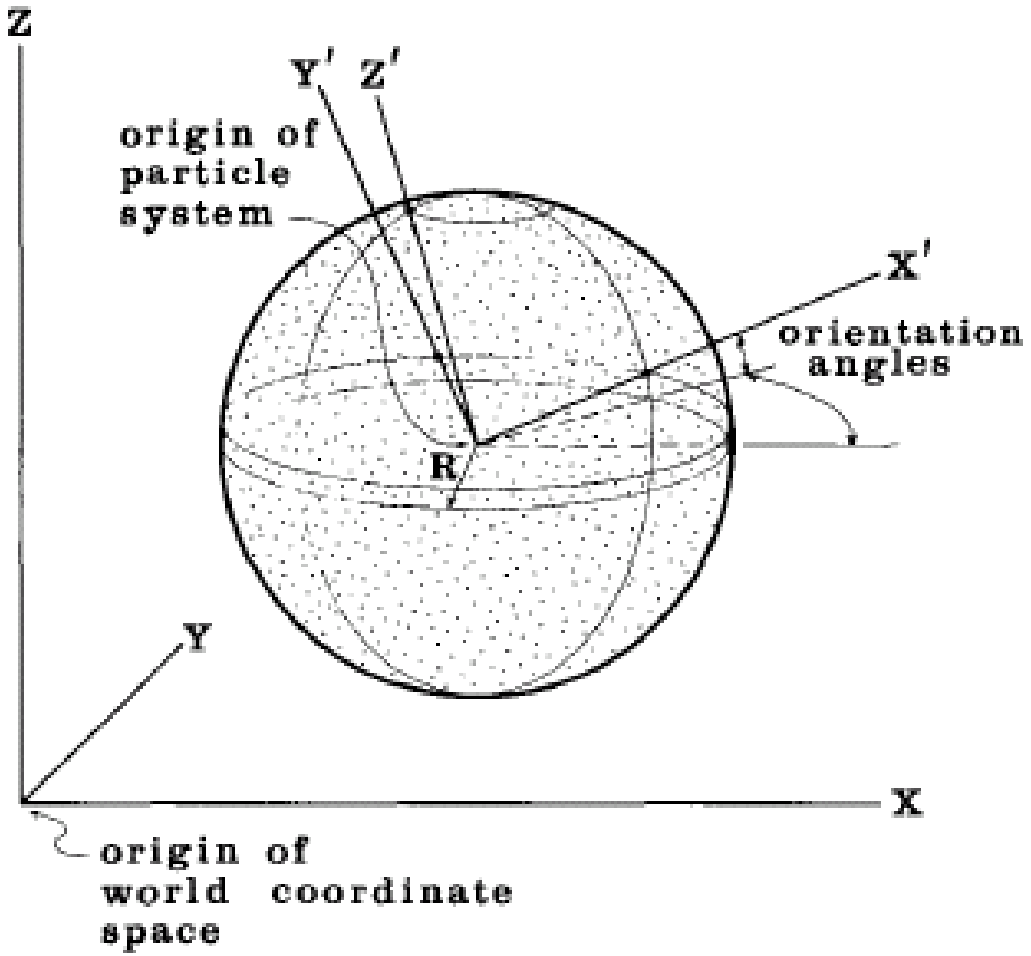
Random požymiai

- Value = mean + Rand()*variance
- Reikia gero atsitiktinių (ar nevisai) skaičių generatoriaus



Formos generavimas

- Kokią pradinę formą turi turėti dalelių sistema?



Gyvenimas

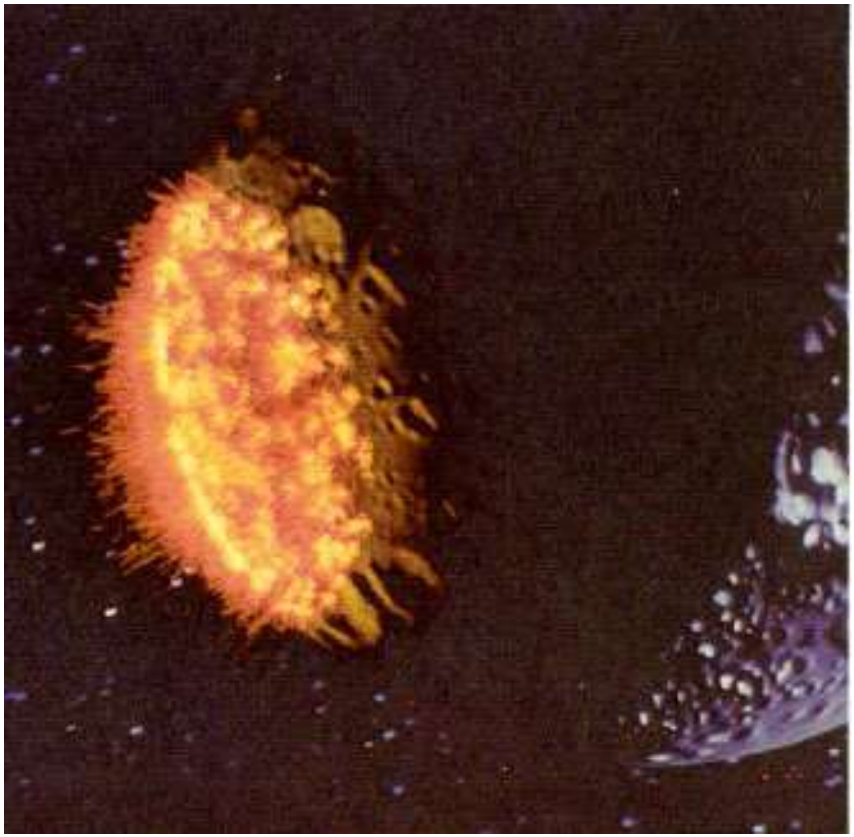
- Kas kiekvieną kadrą reikia pridėti greičio vektorių prie pozicijos vektoriaus
- Jei reikia pridedami papildomi akseleratoriai (kolizijos ar gravitacijos poveikis)

Mirtis

- Dalelių gyvavimo trukmė gali būti apibrėžta gimimo metu į tam tikrą rodomų kadrų skaičių
- Arba dalelės gali būti „nužudomos“
  - Kai jos yra nematomos
  - Kai nutolsta pakankamai toli nuo šaltinio
  - Po tam tikro laiko intervalo
  - Viršijus tam tikrą intensyvumą (taupant resursus)
  - ...

Dalelių renderiavimas

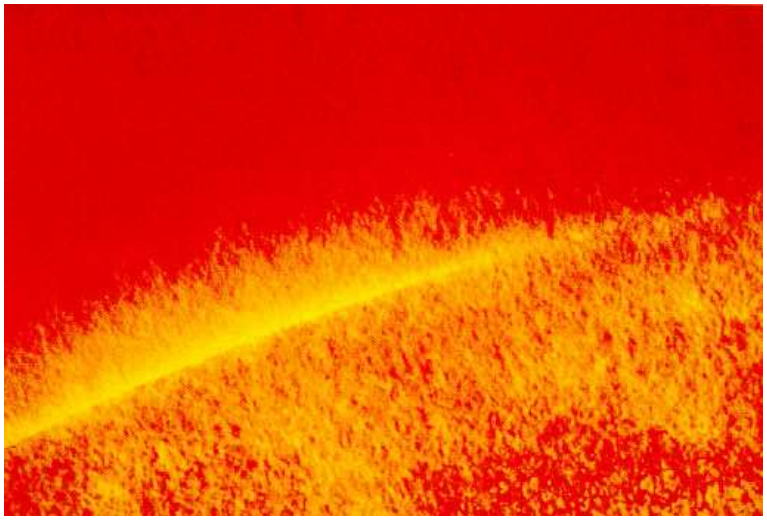
- Problemos
  - Dalelės gali užstoti kitas daleles
  - Dalelės gali mesti šešėlius ir būti permatomos
  - Poligonų primityvai gali sąveikauti su dalelėmis
- Galima daryti prielaidą, kad dalelės nesikerta viena su kita ar su paviršiaus primityvais
- Galima daryti prielaidą, kad dalelės yra taškinės šviesos šaltiniai



Star Trek II: The Wrath of Khan

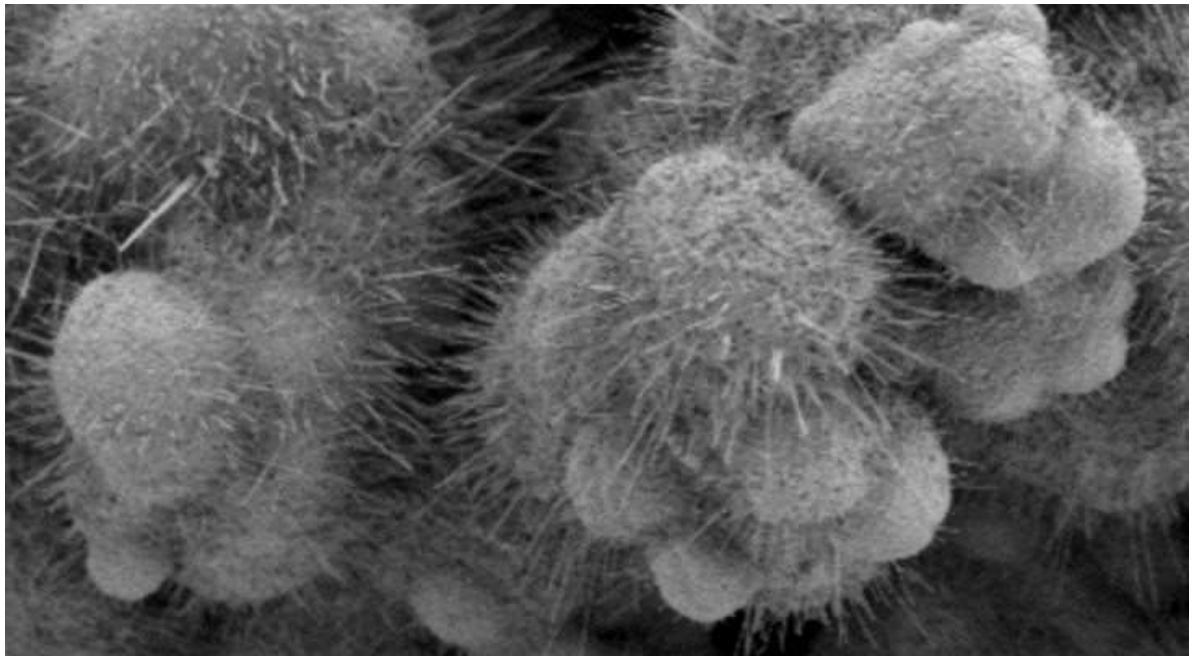
Dalelių renderiavimas

- Dalelė už kitos dalelės gali būti suteikti daugiau šviesos užstojamiems pikseliams



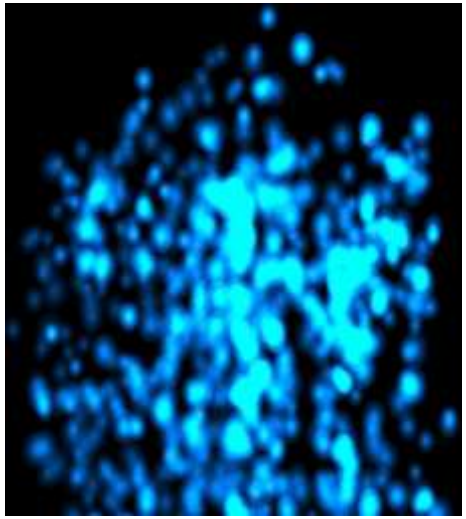
Dalelių hierarchijos

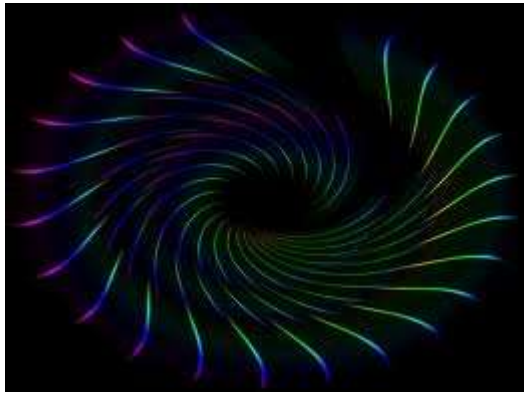
- Galima vietoje dalelių sistemos, padaryti dalelių sistemų sistemą ☺
- Galima pagaminti hierarchijų medį
- Pridėti turbulencijos ir bangavimo efektus

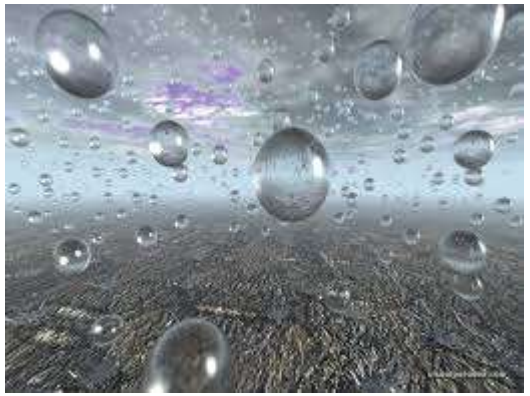


Kvailos dalelės

- Tai tokios dalelės kurios nesąveikauja viena su kita
- Sūkuriai, dūmai, lietus, ugnis ir pan.





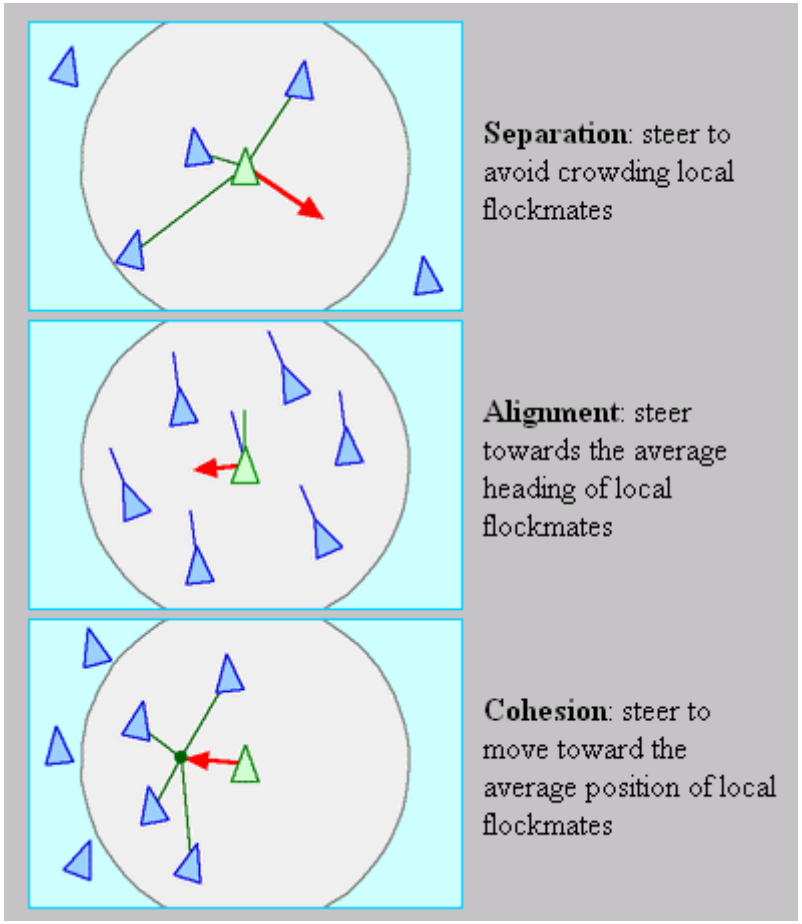


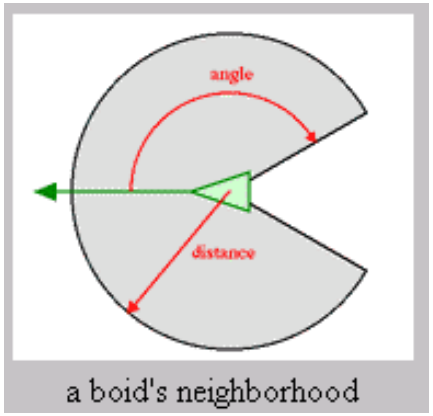
Protingos dalelės

- Dalelės sąveikauja viena su kita
- Tinka imituoti:
  - Bandas, būrius, spiečius (Boids 1986)
  - Skysčius
  - Kolizijas + turbulenciją

Spiečių modeliavimas

- Dalelės turi nesikirsti viena su kita
- Būti nukreiptos ta pačia kryptimi kaip ir gretimos
- Turi keliauti link vidutinės kaimynų pozicijos

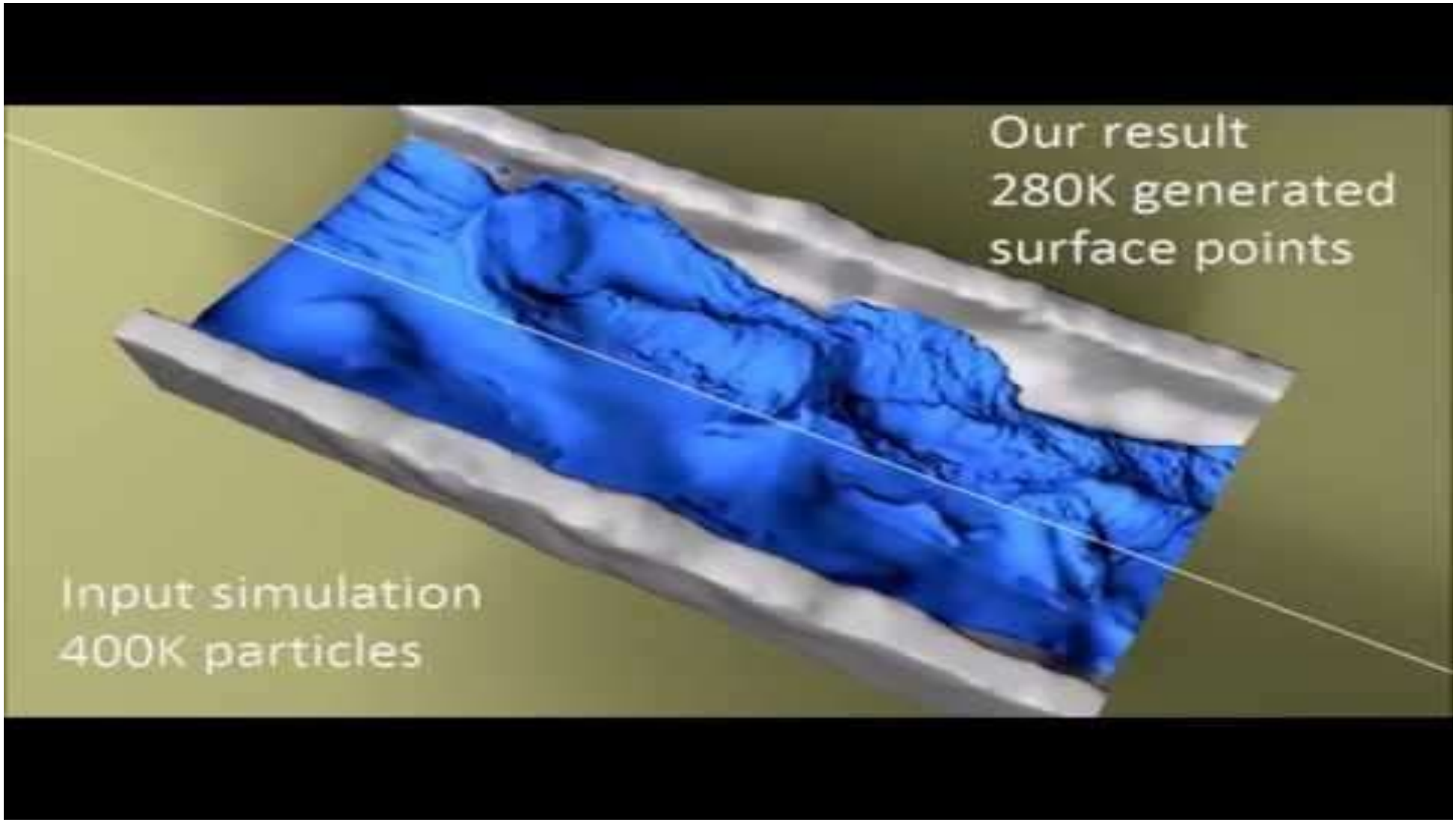




Tekančio skysčio modeliavimas

- Reikia nustatyti tankumą, slėgį ir klampumą
- Dalelės turi masę
- Dalelės yra kūnai kurie užima vietą
  - Susidūrimuose išsaugomas momentas (judesio kiekis)
  - Valdoma gravitacinių jėgų
- Karščio perdavimas
- Paviršiaus ištampymai

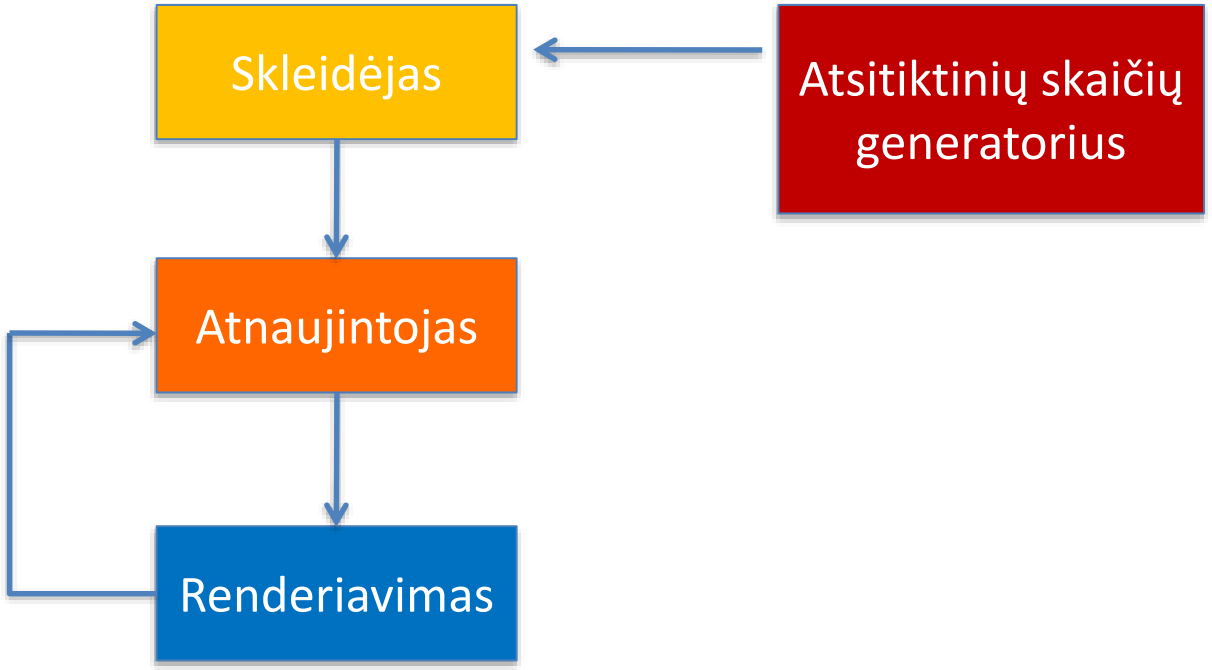
Tekančio skysčio modeliaviams



https://www.youtube.com/watch?v=EjFW9LJpaVg

Kaip?

- Atsitiktinių skaičių generatorius: dalelių skleidėjo įėjime (kiek dalelių skleisti, kas kiek, kada ir pan.)
- Skleidėjas: tai dalelių ir jų pozicijų šaltinis. Valdo dalelių skaičių, kryptį ir kitus globalius nustatymus
- Atnaujintojas: Dalelių požymiai yra atnaujinami prieš jas panaikinant ar perėjus tam tikrą scenos tašką (pvz., ugnis virsta dūmais).



Kaip?

- Dalelės yra generuojamos *skleidėjuje.*
  - Skleidėjo pozicija ir orientacija yra apibrėžiama diskretiškai;
  - Skleidėjo dažnis, kryptis, srautas ir pan. yra apibrėžiamas tam tikru ribotu atsitiktiniu diapazonu (bounded random)
- Svarbus laikas, kiekvienu laiko momentu juda jūsų dalelės
  - Naujos dalelės yra sugeneruojamos, senos yra pašalinamos
  - Veikia jėgos (gravitacija, vėjas, etc.) ir įtakoja dalelės pagreitį
  - Pagreitis keičia greitį
  - Greitis keičia poziciją

Skleidėjas

- Kiekvienam kadrui reikia išgeneruoti naujų dalelių.
  - Du būdai:
    - Apriboti vidutinį dalelių skaičių kadrui:
      - # new particles = average # particles per frame + rand() ◈ variance
    - Apriboti vidutinį dalelių skaičių ekrano daliai:
      - # new particles = average # particles per area + rand() ◈ variance ◈ screen area

Integracija

- Kiekviena nauja dalelė turi bent tokius atributus:
  - Pradinę poziciją
  - Pradinį greitumą (sparta ir kryptis)

- Galima:
  - Įvertinti daleles arbitriniu laiko momentu t kaip uždaros formos lygtį sistemoje be būsenų.
  - Arba naudoti iteratyvius (skaitinius) metodus:
    - Euler
    - Verlet
    - Runge-Kutta

Kaip?

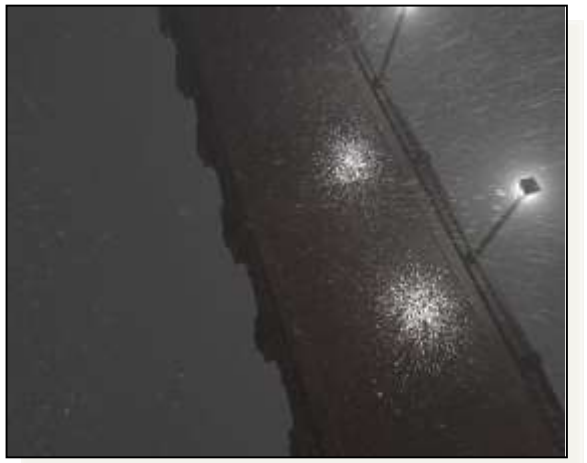
- Renderiavimas: Kaip daleles rodyti ekrane?
  - Paišyti taškus
  - Paišyti linijas (nuo buvusios pozicijos iki kitos pozicijos)
  - Paišyti spraitus (tekstūruoti kvadratukai atsukti į kamerą)
  - Paišyti geometriją (mažas figūras)

A cube emitting 5000 animated particles, obeying a "gravitational" force in the negative Y direction.

The same cube emitter rendered using static particles, or strands.

Renderiavimas

- Galima renderiuoti daleles kaip taškus, tekstūruotus poligonus ar primityvią geometriją
  - Poligonai su alpha-blended paveikslais labai tinka ugniai, dūmams ir pan. gaminti
- Vienos dalelės tipo keitimas kitu sukuria realistiškus interaktyvius efektus
  - Pvz., lietaus dalelę galima pakeisti lašo dalele ant paviršiaus
- Dalelės gali sudaryti ir patį objektą
  - Taip imituojami judantys skysčiai



Fizika

- Daleles 3D erdvėje „veikia“ jėgos (pvz., gravitacija ir vėjas) pagal kurių įtaką dalelių judėjimą galima pagreitinti ar pamažinti.
- Judėjimas aprašomas taip (Euler):

- Integrate acceleration to velocity:

$$v = \bar{v} + a \cdot \Delta t$$

- Integrate velocity to position:

$$p = \bar{p} + v \cdot \Delta t$$

Δt	time step
a	acceleration
v	velocity
$\bar{v}$	prev. veloc.
p	position
$\bar{p}$	prev. pos.

Fizika

- Arba taip (Verlet būdas)
- Pagreitis pagal poziciją:
- Formulė nereikalauja saugoti dalelių greičio, bet laiko žingsnis turi būti pastovus
- Naudojama sudėtingos greičio operacijoms kaip reakciją į smūgius ar susidūrimus su kitais objektais (collision reaction)

$$p = 2\bar{p} - \bar{\bar{p}} + a \cdot \Delta t^2$$

$\bar{\bar{p}}$ position two time steps before

Implementacija su GPU

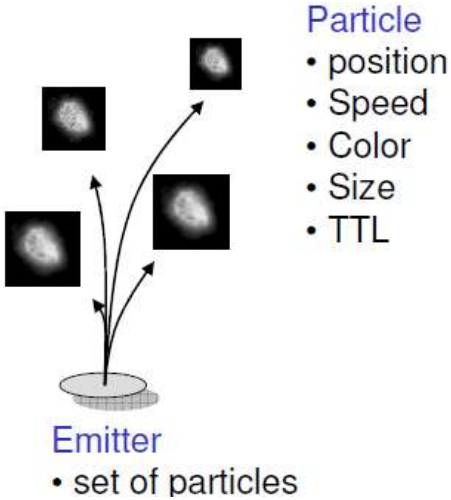
MULTIMEDIJOS INŽINERIJOS KATEDRA

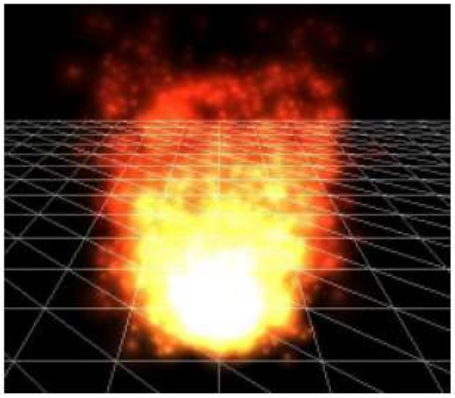
- Du dalelių sistemų tipai:
- Imitavimas be būsenos (Stateless simulation)
  - Tik paprastiems efektams
  - Per vertex šeiderį
  - Veikia ant bet kokio HW kuris palaiko VS

- Būseną saugantis (State-preserving (iterative)) imitavimas
  - Imitavimas su tekstūromis ir pikselių šeideriais
  - Veikia tik su „šiuolaikiniais“ GPU (PS/VS >=v2.0)

Stateless dalelių sistemos

- Nesaugomi kintami dalelių duomenys
- Funkcionalumas realizuojamas per judėjimo ar atributų pokyčius
- Suskaičiuoti duomenys priklauso tik nuo pradiniu verčiu ir statinio aplinkos aprašo.





Požymiai

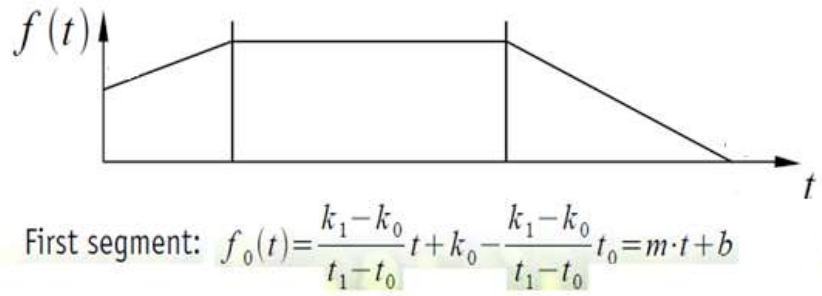
- Dalelės pozicija priklauso nuo pradinės pozicijos, pradinio greičio ir gravitacijos:

$$p(t) = p_0 + v_0 t + \frac{1}{2} g t^2$$

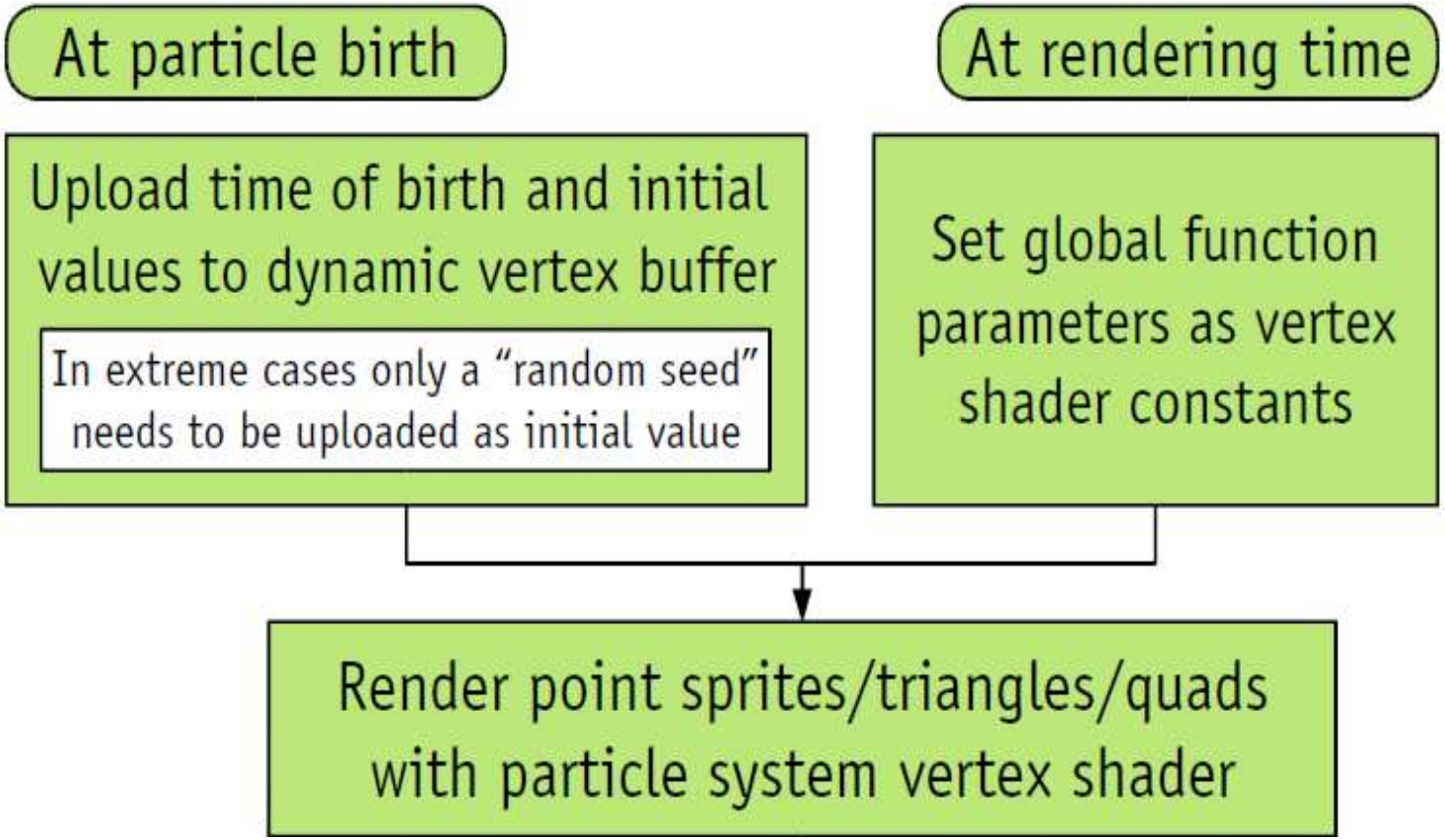
- Orientavimas priklauso nuo pradinio orientavimo ir sukimo greičio:

$$\omega(t) = \omega_0 + \varphi t$$

- Spalva ir permatomumas priklauso nuo tiesinės funkcijos keturių esminių kintamųjų (k0, k1, t0, t1)



Stateless PS algoritmas



Dalelės požymių skaičiavimas

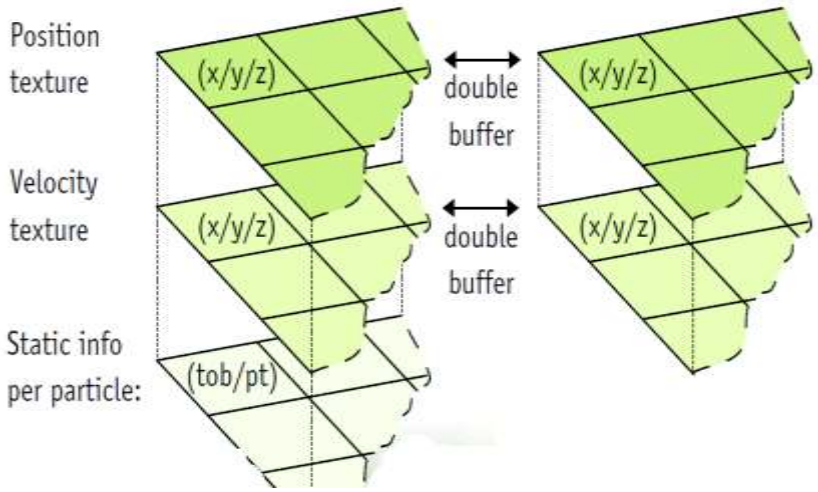
- Globalus laikas yra “0”
- Kaip ilgai dalelė bus aktyvi (matoma)?
- time = globalTime – tInitial;
- Galutinė pozicija:
- pFinal = pInitial + vInitial * t + 0.5 * acceleration * t * t;
- Spalvos ir dydžio funkcijos laike (Random) ar statinės funkcijos?
- Skaičiuojam:
- Xnew = Xold + Vold * Time change;
- Vnew = Vold + a * Time change;

State Preserving imitavimas

- Pozicija ir greitis saugoma tekstūrose
- Iš šių tekstūrų kiekvienas imitavimo etapas renderiuojamas į vienodo dydžio KITAS tekstūras
- Fragmentų (Fragment) šeideris atlieka iteratyvų integravimą
- Pozicijos tekstūros yra „reinterpretuojamos“ kaip viršūnėlių duomenys
- Renderiuojami spraitai/trikampiai/kvadratai

SP dalelių saugomas

- Pozicija ir greitis saugomi 2D tekstūrose
- Tekstūros laikomos 1D masyvu. Tikslumas gali būti pozicijai 32bit FP, greičiui 16bit FP .
- Statinė informacija yra:
  - „Gimimo laikas“,
  - tipas ID.



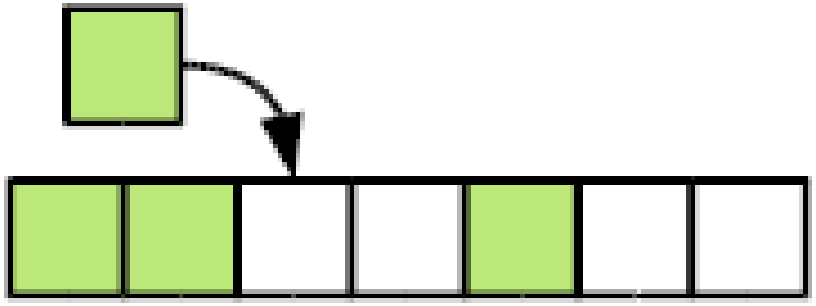
Algoritmas (preserve state)

- Pažiūrim gimimo ir mirties laikus
- Susiskaičiuojam greičio operacijas (jėgos, kolizijos, slopinimai, etc.)
- Susiskaičiuojam pozicijų operacijas
- Susiskaičiuojam alpha blending (optional)
- Perkeliam pikselių duomenis į viršūnėlių duomenis
- Renderiuojame

Dalelių gimimas ir alokavimas

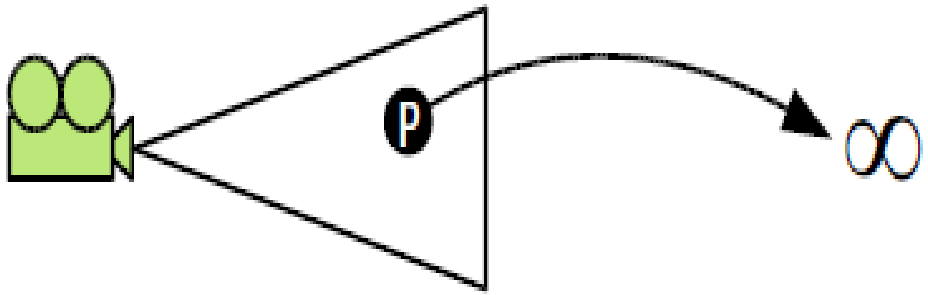
MULTIMEDIJOS INŽINERIJOS KATEDRA

- Alokavimas atliekamas CPU:
  - Nustatomas kitas laisvas indeksas (next free index)
  - Nustatomos pradinės dalelės vertės (initial particle values)
  - „Renderiuojame“ pradines vertes kaip pikselio dydžio taškus



Dalelių mirtis

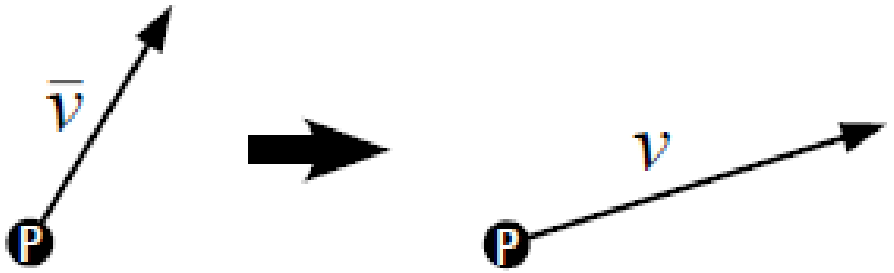
- Apdorojamas nepriklausomai CPU ir GPU
- CPU: atlaisvinamas dalelių indeksas, pridedamas atgal į alokatorių
- GPU: perkeliame daleles link begalybės (arba labai toli) ar pašalinam iš vaizdo



Greičio operacijos

MULTIMEDIJOS INŽINERIJOS KATEDRA

- Atnaujinamos greičio tekstūros pagal įvairias operacijas:
  - Globalios jėgos (gravitacija)
  - Lokalios jėgos
  - Slopinimas (dampening)
  - Susidūrimai (collisions)(
  - Srauto laukai



Globalios ir lokalios jėgos

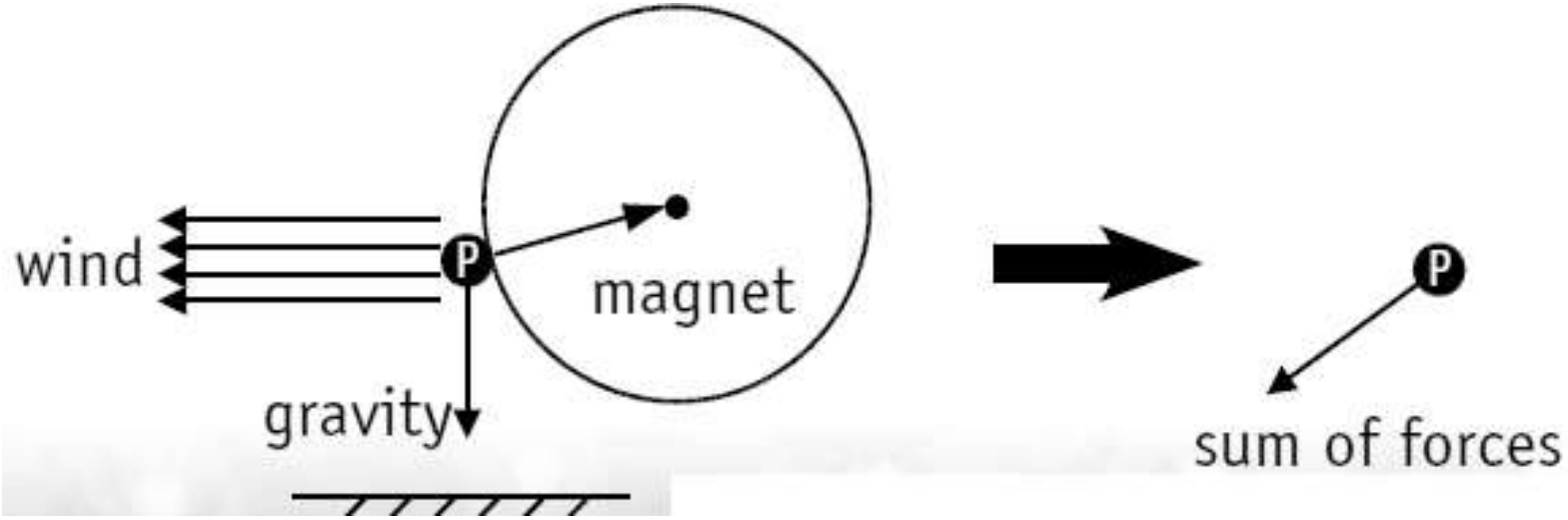
- Globalios jėgos yra nekintamos nuo pozicijos: gravitacija, vėjas
- Lokalios jėgos kinta priklausomai nuo atstumų: magnetinės, orbitos, turbulencijos, kritimo ir kitos
- $$\frac{1}{r^2+\varepsilon}$$

  r distance

  ε small epsilon
- Keičiama efekto skalė priklausomai nuo masės, oro pasipriešinimo ir kt.

Globalios ir lokalios jėgos

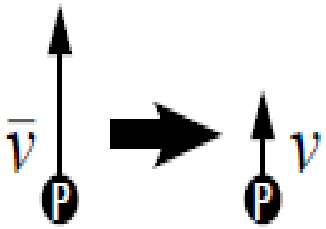
- Visas jėgas reikia sudėti į vieną jėgos vektorių
- Jėga konvertuojama į pagreitį (f = m × a)
  (identiška jei visos dalelės turi vienodą masę)



Slopinimas ir atslopinimas

- Mažinamas greičio vektorius
- Imituoja sulėtėjimą klampiose (ar lipniose) medžiagose

- Atslopinimas:
  - Padidinate greičio vektorių
  - Imituoja savaime judančius objektus (pvz. bičių pulką)



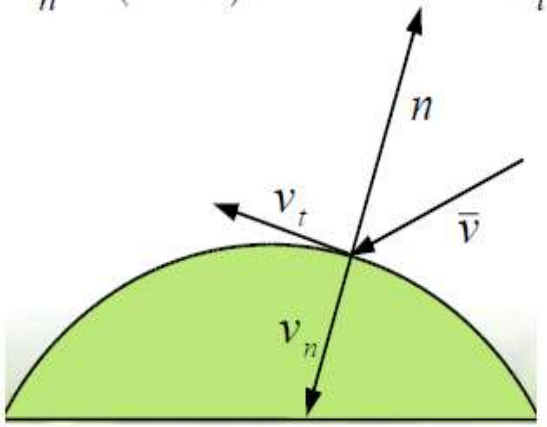
$$v = c \cdot \bar{v}$$

c constant scale factor

Susidūrimai

- Daromi per GPU
- Algoritmas:
  - Aptinkamas susidūrimas su tikėtinomis naujomis pozicijomis
  - Nustatoma paviršiaus normalė apytiksliame taške
  - Reaguojama į susidūrimą (pvz., padidinam greitį)
- Greitis išskaidomas (reliatyviai susidūrimui) į normalinę ir tangentinę greičio komponentes:

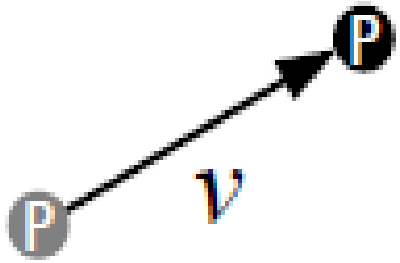
$$v_n = (\bar{v} \cdot n)\bar{v} \qquad v_t = \bar{v} - v_n$$



Pozicijos atnaujinimas

MULTIMEDIJOS INŽINERIJOS KATEDRA

- Euler metodas:
  - Tiesiog pridedamas greitis
- Verlet metodas:
  - Pridedamas pagreitis pagal jėgų poveikį
  - Pridedami kiti efektai (pvz., slopinimai ir kolizijos).



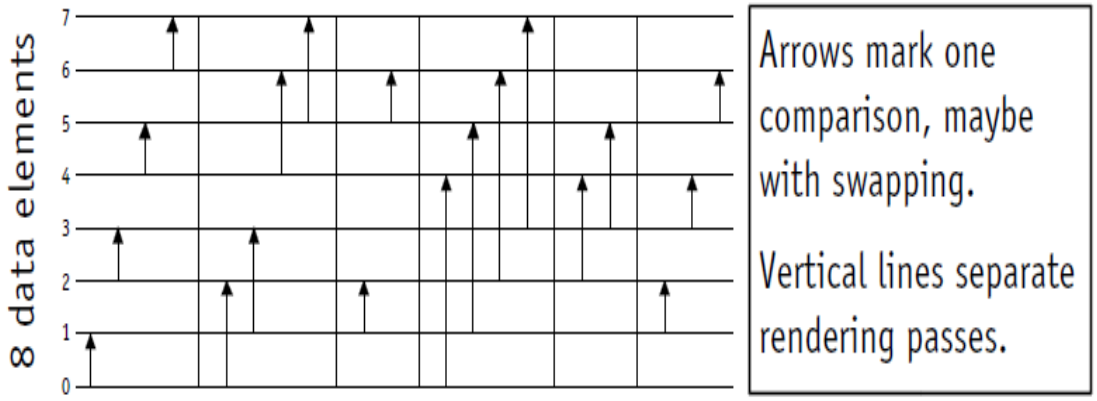
Alpha-Blending

- Alpha-blended dalelės rodo artifaktus kai renderiuojama neišrūšiavus

- GPU gali atlikti rūšiavimą geriau nei jūs
  - Dedamas atstumas (viewer-distance) ir indeksas į tekstūrą
  - Tekstūra surūšiuojama (bent dalinai)

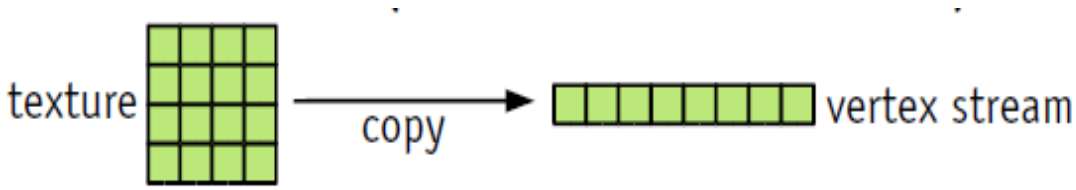
Rūšiavimo tinklai

- GPU veikia lygiagretaus skaičiavimo principu, vadinasi reikia lygiagrečiai rūšiuoti.
- Tam reikia nuo duomenų nepriklausomo algoritmo
- Pvz., Nesveiki-Sveiki jungimo rūšiavimas (Odd-Even Merge Sort:



Dalelių perkėlimas į spraitus

- Reikia perkelti dalelės pozicijas į geometrinius duomenis
- Taškiniai spraitai (paprasčiau – tiesiog taškai) yra efektyviausiai perkeliami nes tereikia skaičiuoti tik vieną viršūnėlę dalelei
- Trikampiams ar kvadratams reikia replikuoti viršūnėles
- Yra keletas metodų kaip perkelti tekstūros duomenis į viršūnėles.
- Super buferio metodas (Über-Buffer) gali saugoti duomenis grafinės plokštės atmintyje. Kopijuojama iš vieno buferio (tekstūra) į kitą buferį (vertex buffer)



Daugelio dalelių tekstūros

- Galima apjungti kelias tekstūras kaip vienos didelės tekstūros potekstūres
  - Koreguojamos tekstūros koordinatės
  - Taškiniams spraitams:
    - Transformuojamos tekstūros koordinatės fragmentiniame šeideryje