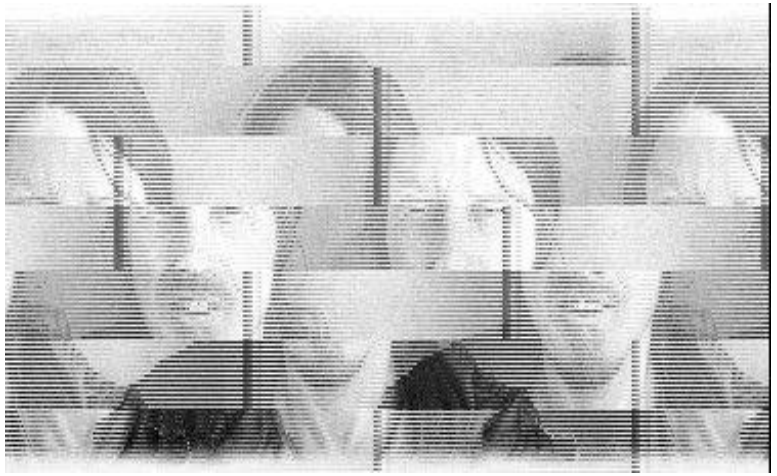
Simulaatioohjelma
GOD
- ei enempää, ei vähempää



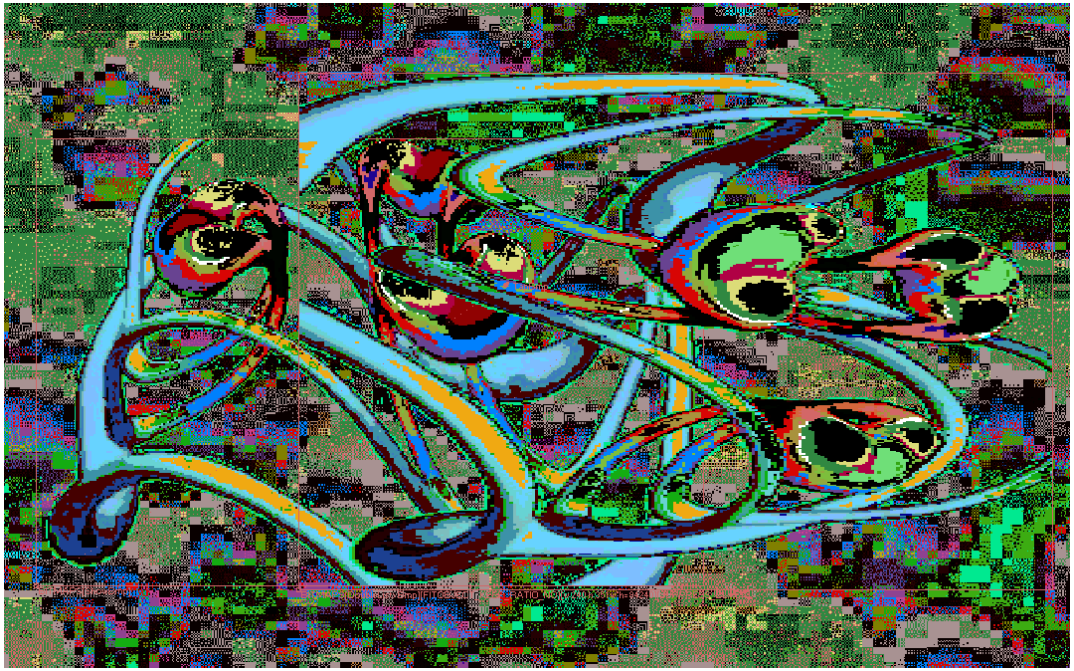
Omaperäisen simulaatio-ohjelman esittely ja intohimoisen tutkimisen kannanotto

Mikä on simulaatio?

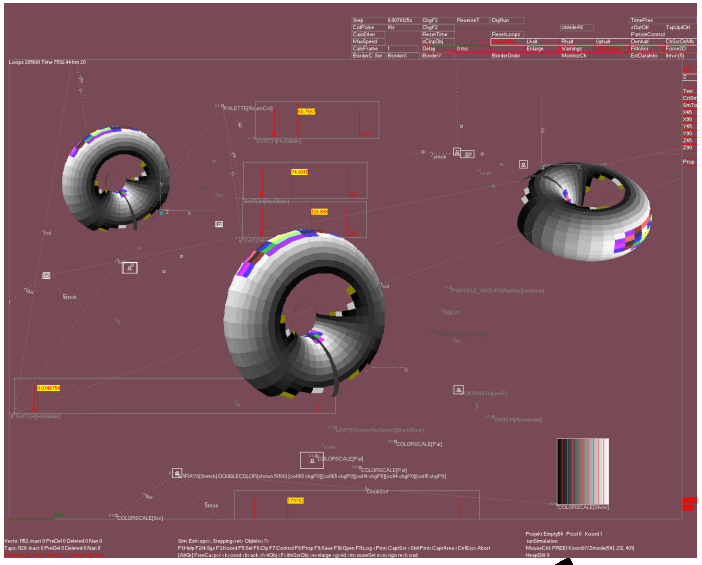
- Simulaatiolla yritetään laskennallisesti ennakoida jonkin tapahtumasarjan kehitystä ajallisesti eteenpäin.
- Simulaatio toteutetaan ohjelmallisella silmukalla, jossa aika-askelluksin lasketaan järjestelmän tulevaa tilaa. Sitten syötetään tulokset takaisin uudeksi lähtökohdaksi ja toistetaan seuraavan tilan laskun jne.
- Simulaation tulosten laatu riippuu siitä, miten hyvin on osattu kuvata laskettava tapahtuma matemaattiseksi algoritmiksi.
- Ohjelmasilmukan tulosten takaisin syöttäminen aiheuttaa aina kertyviä virheitä. Heikko simulaatio kasaa hyvin nopeasti virheitä ja saavuttaa pisteen, jossa tulokset muuttuvat ei-mitäänmerkitseväksi kohinaksi.
- Algoritmit ja simulaatiot hallitsevat nykymaailmaa niin kattavasti, että on hyvä, että tietää edes vähäisen niiden luonteesta ja rajoituksista.
- Simulaatiot ovat vain niin hyviä kuin niiden algoritmien laatija.
- Juuri siksi ne ovat minulle oivallinen tapaa tarkistaa, miten hyvin havaintoni, selitykset ja johtopäätökset ympäröivästä maailmasta käyvät yhteen todellisuuden kanssa.

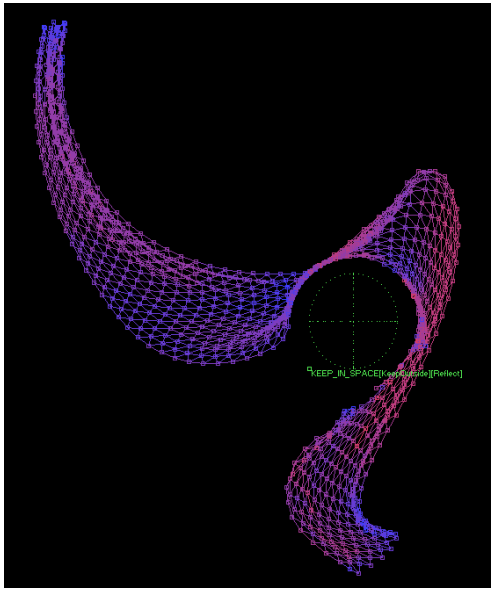
Oh my God?

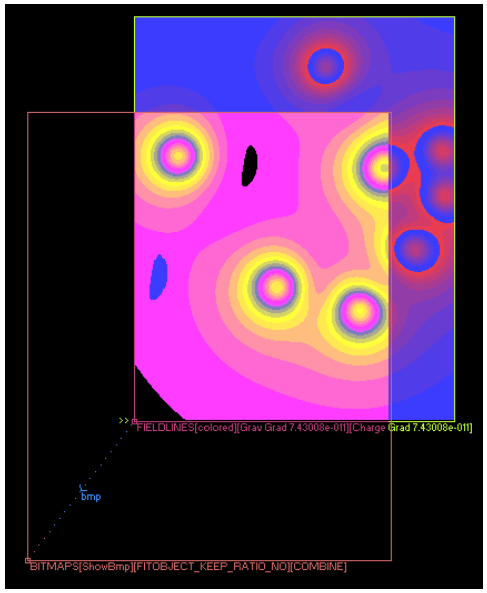
- **God**! on tekemäni simulaatio-ohjelma, jolla voi tutkia simulaatioita. Yritin luoda alustan, joka sallii mahdollisimman laajan valikoiman olioiden vuorovaikutukseen.
- Simulaatio voi sisältää kuvia, ääniä, matemaattisia objekteja, kiinteää mekaniikkaa, painovoima- ja sähkövarauskenttiä, liikettä, sanoja ja vähän optiikkaakin.
- Koejärjestelmiä voi luoda 3D-avaruuteen ja katsella ajon aikanakin kolmiulotteisesti mm. silmäharitusmenetelmällä.
- Simulaatioon voi puuttua ajon aikana, ottaa videoita, seuloa ja tallentaa tapahtumaketjuja.
- Simulaatio voi sisältää ala-simulaatioita. Se voi kutsua ja hylätä niitä. Useita erillisiä simulaatioita voi ajaa rinnakkain ja verrata niitä keskenään ajallisesti synkronoituna.
- Projektin sivurönsyt ulottuvat kryptauksen, tiedon pakkaamisen, tiedon analysointiin ja musiikin/ grafiikan alueelle.
- Ohjelma voi käyttää syöttölaitteita kuten pelisauvoja ja tietokoneen portteja.
- Ohjelmalla on osittainen yhteensopivuus erilaisten grafiikan- , autocad- ja äänitiedostoformaattien kanssa.
- Simulaatiot voivat tuottaa isoja määriä dataa, jonka rakenne voi olla puuromainen. Siihen ohjelma tarjoaa tapoja analysoida ja visualisoida datamassaa. Datan visualisoinnuttua aivojemme on helpompaa havaita lainalaisuuksia.
- Olemme vähän heikkoja havaitsemaan ajallisia muutoksia. Siihen ohjelmassa on työkalu, jota kutsun ”käänteiseksi aikajäljitykseksi”. Nestevirtauksessa voi olla kiinnostavaa, mistä putkistossa esiintyvät paine-erot johtuvat. Voin ohjelmassani jäljittää partikkeleita ja niiden sijainnit ajassa taaksepäin. (Tämä ei yleensä onnistu kääntämällä aika-askellusta taaksepäin kertyvien laskentavirheiden vuoksi.)
- Olen keskittynyt tähän saakka ohjelman perusrakenteisiin. Jos ne ovat terveitä, kaikki muu seuraa perässä. Tutkin yleisestikin minkälaisia asioita tieto käynnistää.




Omavara
opisto




KEEP_IN_SPACE[KeepInScale][Reflect]


FIELDLINES[colored][Grav Grad 7.43008e-011][Charge Grad 7.43008e-011]
bmp
BITMAPS[ShowBmp][FITOBJECT_KEEP_RATIO_NO][COMBINE]


LASER

Miksi tällainen työ?

- Ohjelmoin ”God”ia yli 20 vuotta (sivuharrastuksena omavaraisuudelle). Alku sikisi jo lapsuudessa ja kannoin mukanani paljon kysymyksiä, miksi asiat toimivat tietyllä tavalla.
- Mielenkiintoni kohdistuu erityisesti kaoottisiin tapahtumiin – vuorovaikutussysteemeihin joiden tapahtumat sulautuvat orgaaniseksi sykkeeksi. Ne muistuttavat elämää ja siitä syystä niiden tutkiminen on minulle hyvin lähellä luontaistalouden tutkimista.
- Tietokone ja ohjelmointi tarjoavat tietyt rajalliset resurssit. Minua kiinnostaa resurssien tutkiminen - sekä luontaistaloudessa ja ihmisten toiminnassa että tiedonkäsittelyssä ja tietorakenteissa.
- Ohjelman taustafilosofiasta olen kirjoittanut enemmän tekstissäni "Pieni tutkija”
- Nuorena aikuisena otin etäisyyttä näin insinöörimäisen lähestymistapaan. Palasin pitkän tauon jälkeen vanhojen harrastuksieni pariin, sillä koin ne sittenkin voimavaraksi eikä haitaksi keskustelussa nykymaailman tilasta.
- Päätin asettua nyky- ja vanhan maailman rajamaille tulkiksi siellä missä vain harvat viihtyvät.

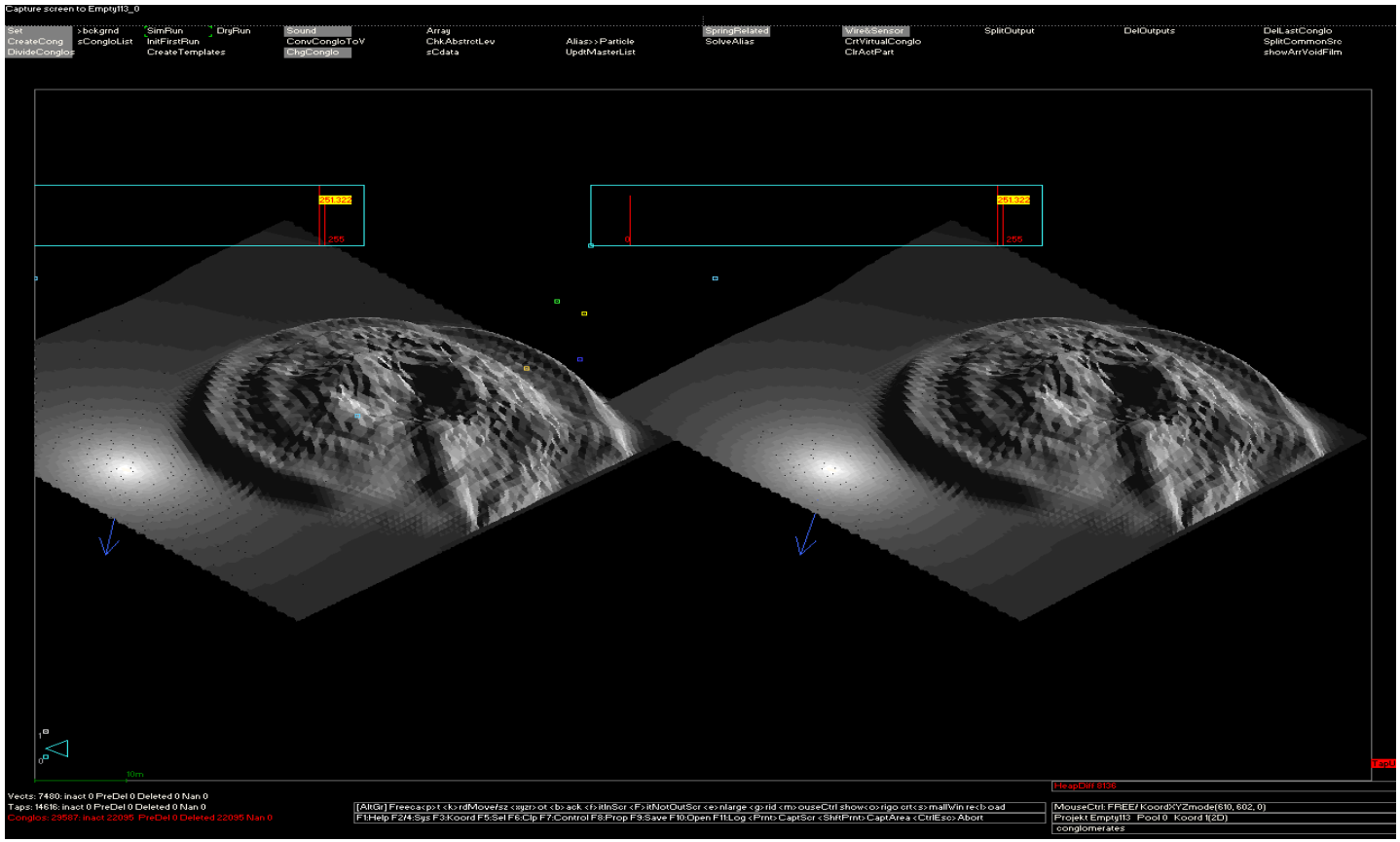
Mutta nyt asiaan!
Esittämäni kuvat ja videot ovat pikselitarkkoja. Avaa näkymä koko ruutuun tai (parempi) lataa videon esikatselusivulta koneellesi katseltavaksi.

Ohjelman yleisiä tietoja

- Koko ohjelma mahtuu melkein yhdelle vanhalle disketille. Ohjelma on äärimmilleen läpistrukturoitu ja toistuvaa koodia on minimoitu. Koodirivejä on ~200000.
- En ole käyttänyt grafiikkamoottoreita tai funktiokirjastoja paitsi C(++) kielen standardikirjastot. Koska tykkään pyörän uudelleen keksimisestä, olen kehitellyt mm. omat viivanpiirtoalgoritmit ja visuaalisen rajapinnan. Se on mahdollistanut, että loin juuri sitä mitä tarvitsin, esim. pikselinväriset vektorit. Ohjelman toimii lähinnä kokoruututilassa ja käyttää vanhan DirectX (ver. 7):n perustoiminnot.
- Nopeus on aika hyvä osin siksi, että äänien, kuvien ja muu tietovirtaa kulkee samoissa tietuerakenteissa. Konvertointi on vähäistä.
- Nopeus olisi huikea, jos osaisin laiteohjelmointia…
- Olen vanhanaikaista DOS-sukupolvea ja dyykkaan tietokoneosat, joten en omista kuumaa rautaa. Ohjelmani toimii laajimmin Windows 98 ympäristössä; aika jolloin yksi ihminen pystyi vielä ymmärtämään, mitä tietokoneen sisällä tapahtuu…
- Simulaatiot ovat lähtökohtaisesti kolmiulotteisia ja niitä voi ajon aikanakin katsoa sellaisina silmäharitustekniikalla tai kaksivärisillä silmälaseilla.

3D- silmäharitusharjoitus

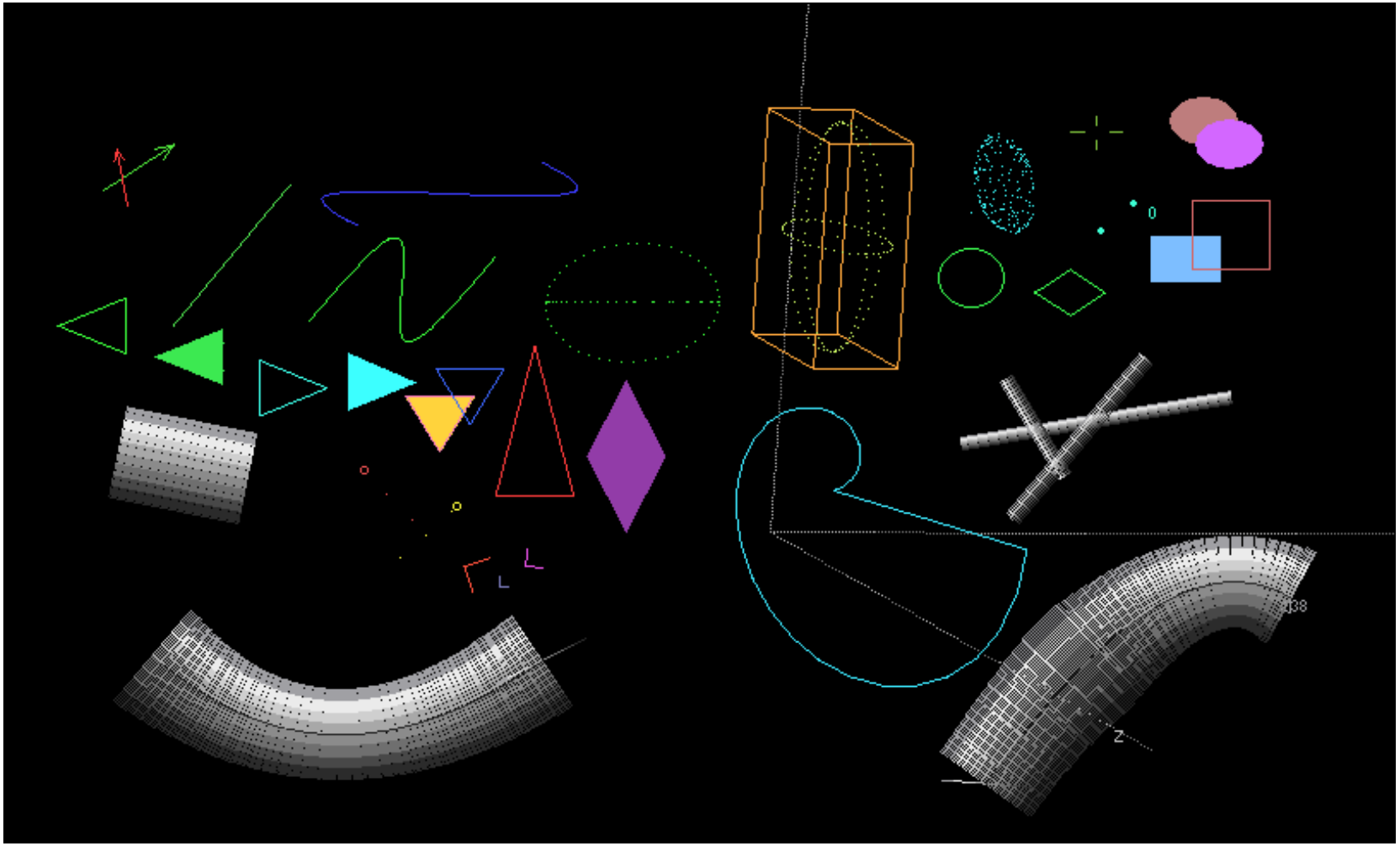
Katso ensin kieroon. Alla olevat aaltokuvat näkyvät silloin neljänä. Päästät silmät hitaasti alkuasentoon. Silloin kun sisimmät osakuvat siirtyvät päällekkäisiksi, niihin muodostuu oikea syvyysvaikutelma. Tämä kuva ei ole paras mahdollinen harjoittelukohde.



Rakenne

- Ohjelmassa on kolmea perusrakennetta: vektoreita, pintoja ja varsinaiset simulaation osaset: congloja.
- Vektoreista ja pinnoista rakennan virtuaalisia objekteja.
- Conglot kommunikoivat simulaatiossa keskenään erilaisten I/O kanavien kautta. Niitä voi kytkeä suoraan toisiinsa tai sensoreiden välityksellä. Väyliä voi olla määrättömästi. Conglo voi olla matemaattinen funktio, alkeishiukkanen, peili, painike tai lähes mitä tahansa. Sen sijainti on jossain 3D-avaruudessa.
- Simulaatioissa ei ole todellisia rajoja. Siellä voivat vuorovaikuttaa oliot, joilla ei ole fysikaalisesti mitään yhteyttä keskenään. Voin törmäyttää taivaankappaleita peileihin tai nauhoittaa kiertoradalle unohtuneen kaiuttimen musiikin maan pinnalla Doppleria unohtamatta. Se on mukavaa.
- Vektorit ovat ohjelman keskiössä ja niistä on lukuisia esitystapoja. Ne ovat perusaineksia, joista rakentuvat monimutkaisemmat Conglot (”Conglomeraatti”, kasautunut) Kuvassa näet, että ne eivät muistuta edes vektoreita:

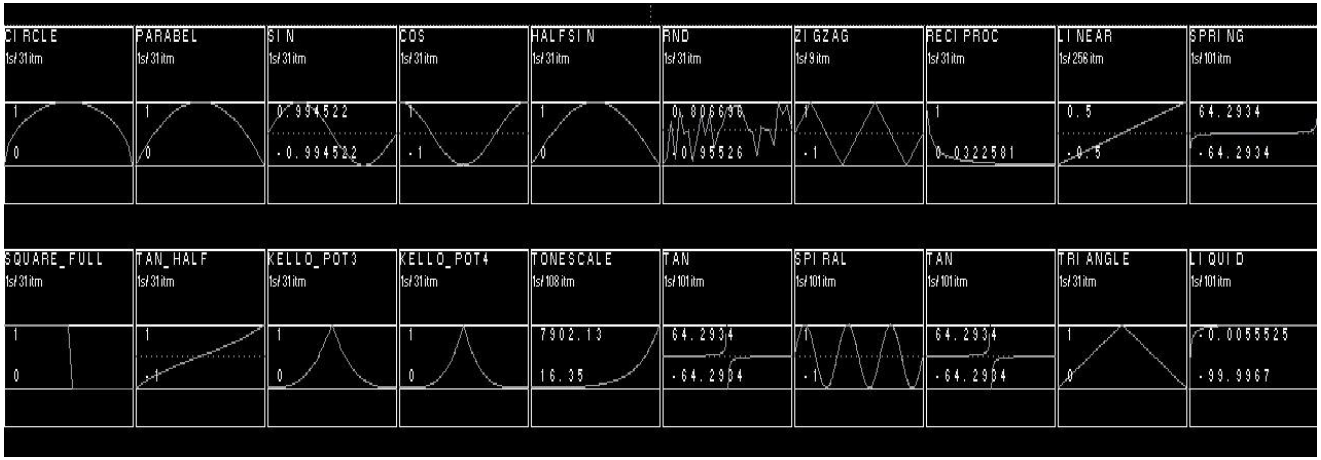
Vektorit ovat enemmän kuin viivoja



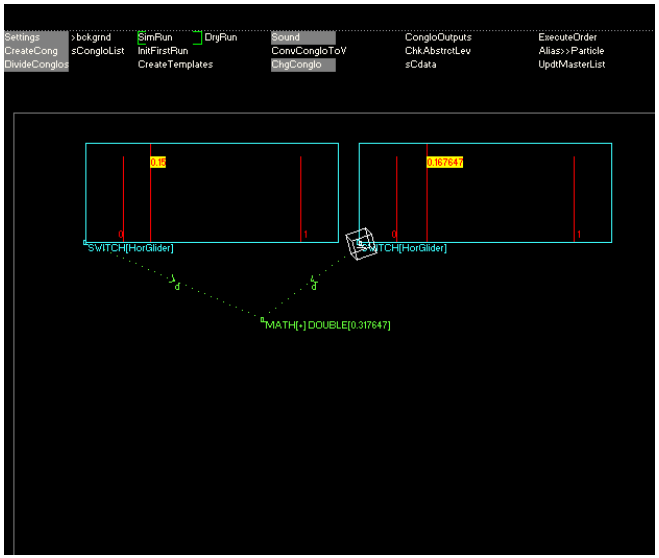
Krumit

- Kehittelin ohjelmaani yhden keskeisen vektorityypin: krumit (saks. ”vino”). Vektorin alku- ja loppupisteen välillä on kolmas 3D-piste, jonka määrää, miten käyrä kulkee ulompien pisteiden välissä – hieman kuin Bezier-käyrät.
- Krumien varaan rakentuvat simulaatioiden peilit, astioiden seinämät, kulkuväylät samoin kuin monet matemaattiset toiminnat ja vääntyneet kuvat. Nuottiviivoina ne mahdollistavat kolmiulotteiset partituurit, jossa melodia voi kulkea solmuun.
- Krumien käyrät voivat muuttua simulaatioiden yhteydessä.

Conglojen ja vektoreiden luonti



- Tässä luodaan standardivalikosta kaksi krumia ja palkkimainen vektori.
- Krumien tarkkuus voi valita. Mitä enemmän pisteitä, sitä tarkempia tuloksia esim. peilinä.
- Avaa/ Lataa



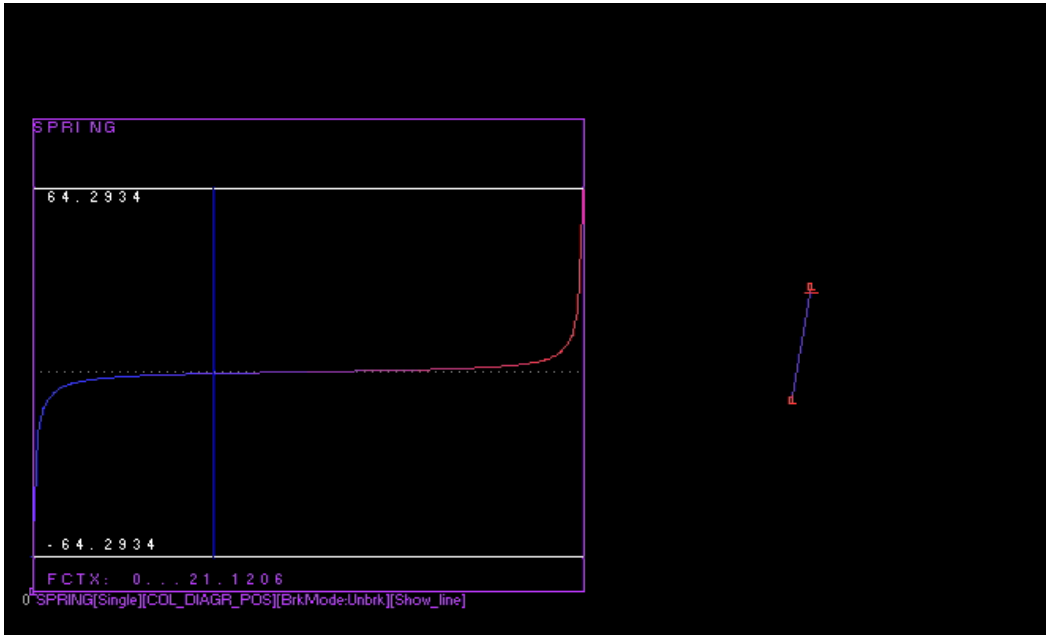
- Täällä tapaa luodaan simulaation objekteja nopeasti.
- Kaksi säädintä, joiden tulo on desimaaliluku. Sitten ynnäävä olio.
- Laitetaan tulos näkyviin. Ynnääjän yhteys säätimiin on nyt kiinteä eli olioita voi liikuttaa vapaasti keskenään. Sitten yhteys muutetaan sensoriksi, sensori pitää olla avaruudellisesti säätimen kohdalla johdattaakseen datavirtaa. Tässä sensorin muoto on kuutio mutta se voi olla minkätahansa muotoinen tai kokoinen ja se voi liikkua ja toimia kuin hiukkanen (varaus, painovoima). Sensoreita voi käyttää datakeräiminä neuronien tapaan. Ne kykenevät etsimään kohteensa (jos haluavat).
- Avaa/ Lataa

Ohjelman suonissa liikkuu universaali tietorakenne

- Krumeihin liittyy ohjelman keskeisin tietorakenne: neliulotteinen (xyz + aika) ja värillinen piste: Ptc
- Ptc-tietueen kautta kulkee suuri osa simulaation datavirrasta. Samasta tietotyypistä on etu, kun yhdistän hyvin erilaisia asioita kuten kuvia ja ääniä. Silloin voin vähentää aikavieviä datakonvertointitarpeita. Esim. sama tietue määrää yhtä lailla vieterin ominaisuudet sitä venyttäessä kuin ääniaallon.
- Krumeilla voin toteuttaa käyrien melko tehokasta häviöllistä datapakkausta. Käyrän kokonaisten osien kuvaamiseen tarvitaan vain kolmen pistettä!

Jousi kahden kappaleen välissä

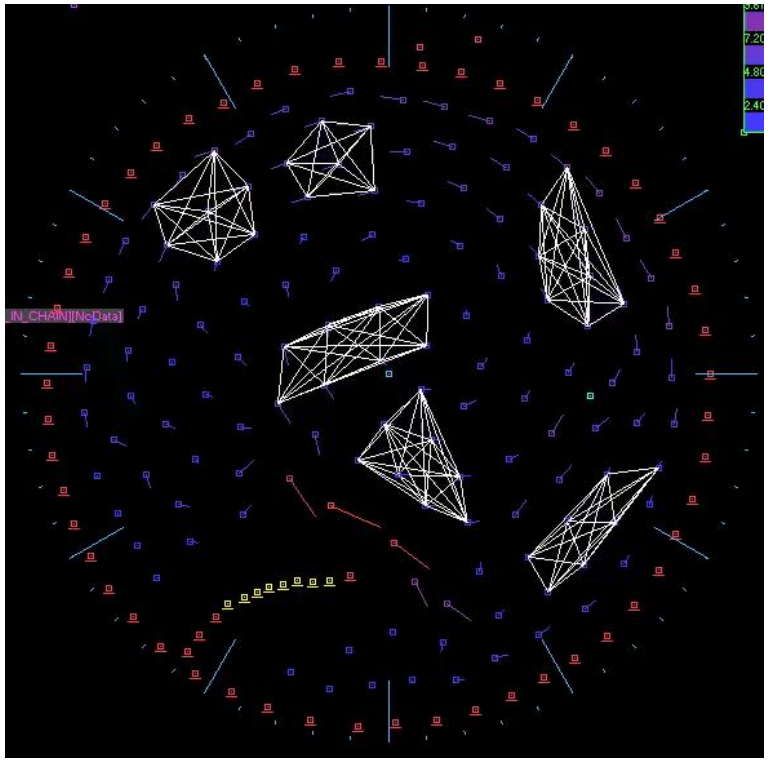
- Toinen kohde on kiinni, toinen värähtelee sen ympärillä. Ptc kuvaa vieterin ominaisuutta. Punainen väri vastaa venytystä, sininen väri kuvaa puristusta. Ajon aikana muutan vastetta manuaalisesti, mikä näkyy kohteen värähtelyssä. Avaa/ Lataa



- Vasemmanpuoleinen laatikko sisältää vieterin vasteen.
- Vieteriolio tuottaa dataa vieterin tilasta, jännityksestä, poikkeamasta ja onko vieteri vielä ehjä. Tätä dataa voi toinen conglo hakea joko suoraan tai sensorin kautta (sensoritkin ovat kolmiulotteisia. Data siirtyy vain jos datalähde on sensorin sisällä.
- Vieteriin voi kytkeä lisätoimintoja kuten esim. vieterin väsymisestä huolehtivan moduulin.
- Asetus on, että vieterin jännitystila vastaa diagrammissa olevan käyränkohdan väri.

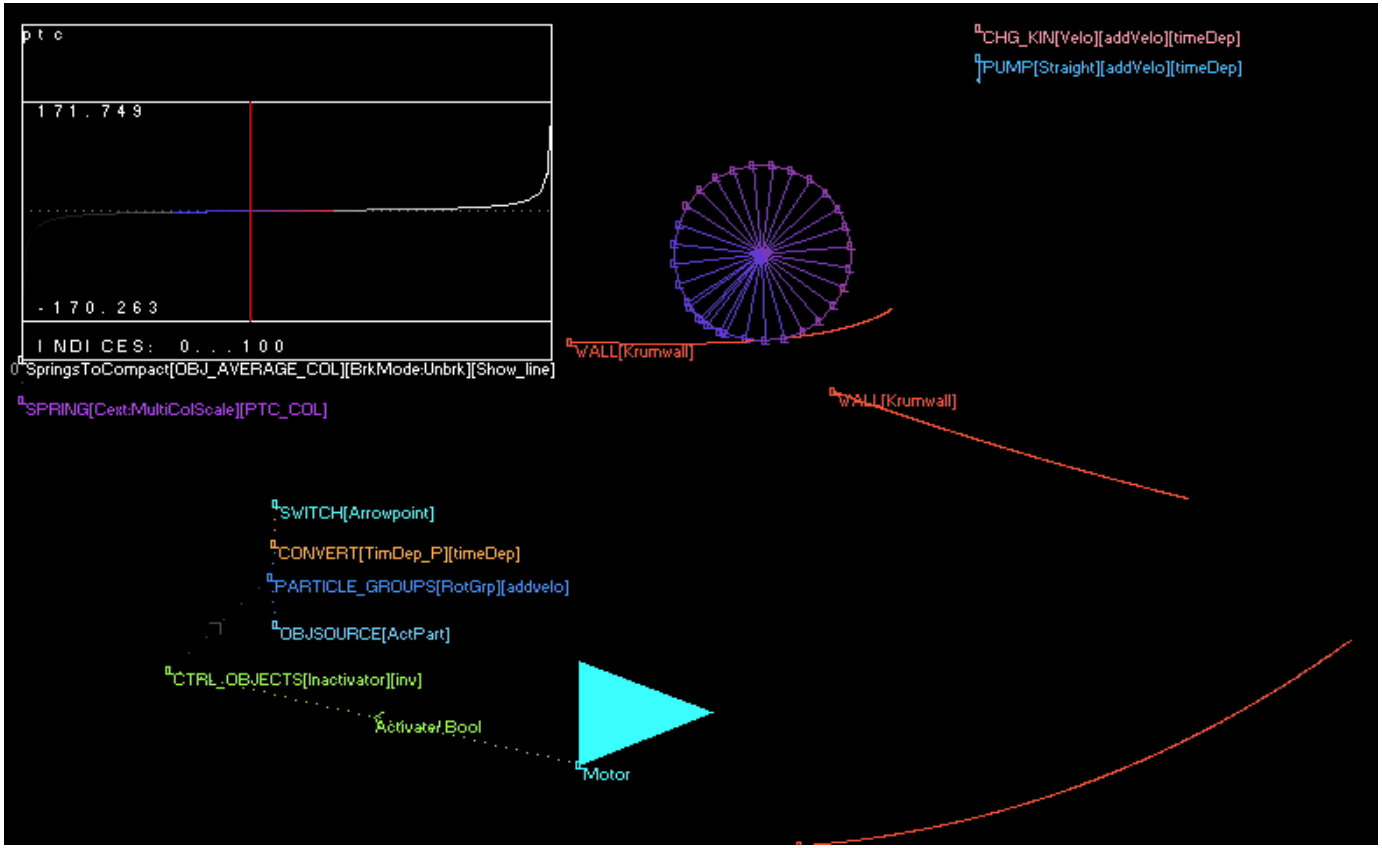
Kiinteät kappaleet ja nesteet

- Jouset eri muodoissa ovat monen mekaanisen simulaation rakennusosia. Verkkoina niillä voi jäljitellä rumpukalvoa. Kolmiulotteisina ne voivat edustaa kiinteitä kappaleita. Seuraavassa simulaatiossa erikiinteät kappaleet liikkuvat nesteessä, jossa on virtaukselle estekohta. Avaa/ Lataa



- Kelluvista kappaleista pehmeämmät vääntyvät virtauksessa.
- Voin luoda vaivatta timanttia kovempia aineita!

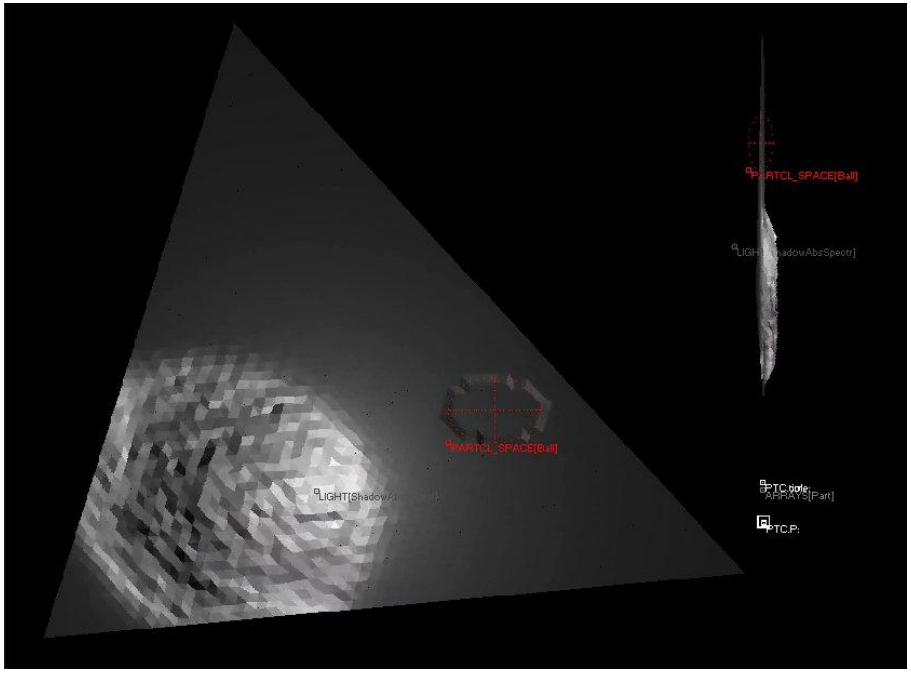
Pyörivä pyörä



- Pyörä kulkee, lipsuu, moottori sammutetaan ja sinne meni!
- Pyörän pinnat vaihtavat värinsä kuormituksen mukaan.
- Vaste on palanen tan-funktiota ja sopii vieterien reaalimaailmaan. Vaste kasvaa ääriasentoihin mentäessä. Avaa/ Lataa

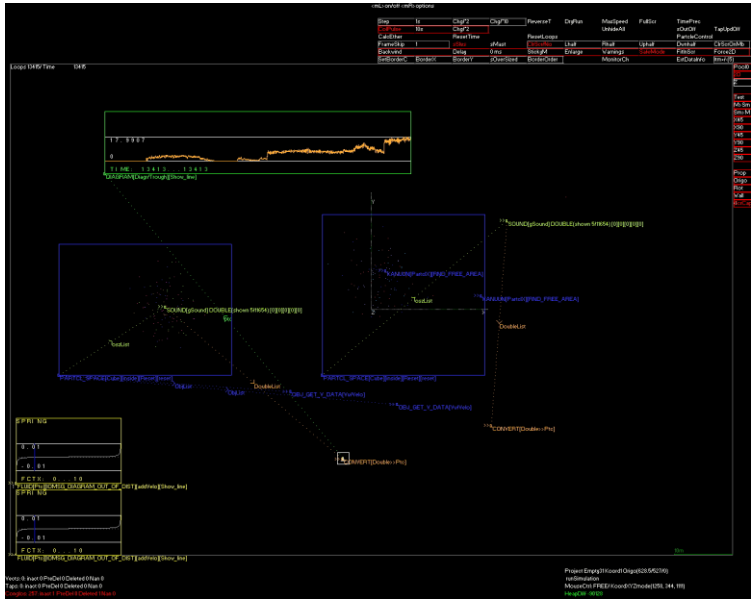
Äänet ja varjot

- Ohjelma voi nauhoittaa simulaatioiden tuottamat äänet. Tässä sellainen ääni, joka hypnoottisuudessaan on minulle arvoitus.
- Yhdistin äänen simulaatioon, jossa ”painoin” rumpukalvoon pintaan logomme. Pinta sai varjostuksen virtuaalisesta valonlähteestä.
- Kuultava ääni ei liity tähän simulaatioon. Yhdistelyn takana oli taiteellinen olo. Outo ääni ja musta kuutio. Vähän kuin avaruusodysseiassa. Avaa/ Lataa
- Tyypillinen kulahtanut kahden laukaisun räjähdyssimulaatio. Sori mielikuvituksen puutteesta. Oikealla tapahtuma näkyy sivukuvana.
- Heiluttelen taskulamppua näkymän yläpuolella. Ohjelma laskee pintaan osuvaa valon määrää, joka näkyy tummuutena. Avaa/ Lataa



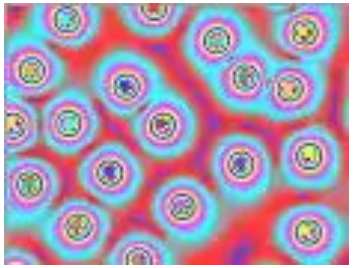
Lisää esimerkkejä

- Hiukkaspilven äännähdyksen nauhoitus. Avaa/ Lataa

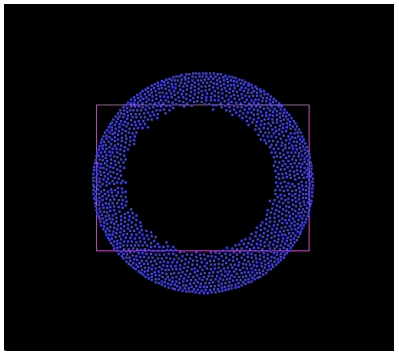


- Hiukkaspilven ääni osoittautui tylsän konemaiseksi. Kuuntele

- Rumpukalvon simulaation ääni oli uskottavampi. Kuuntele



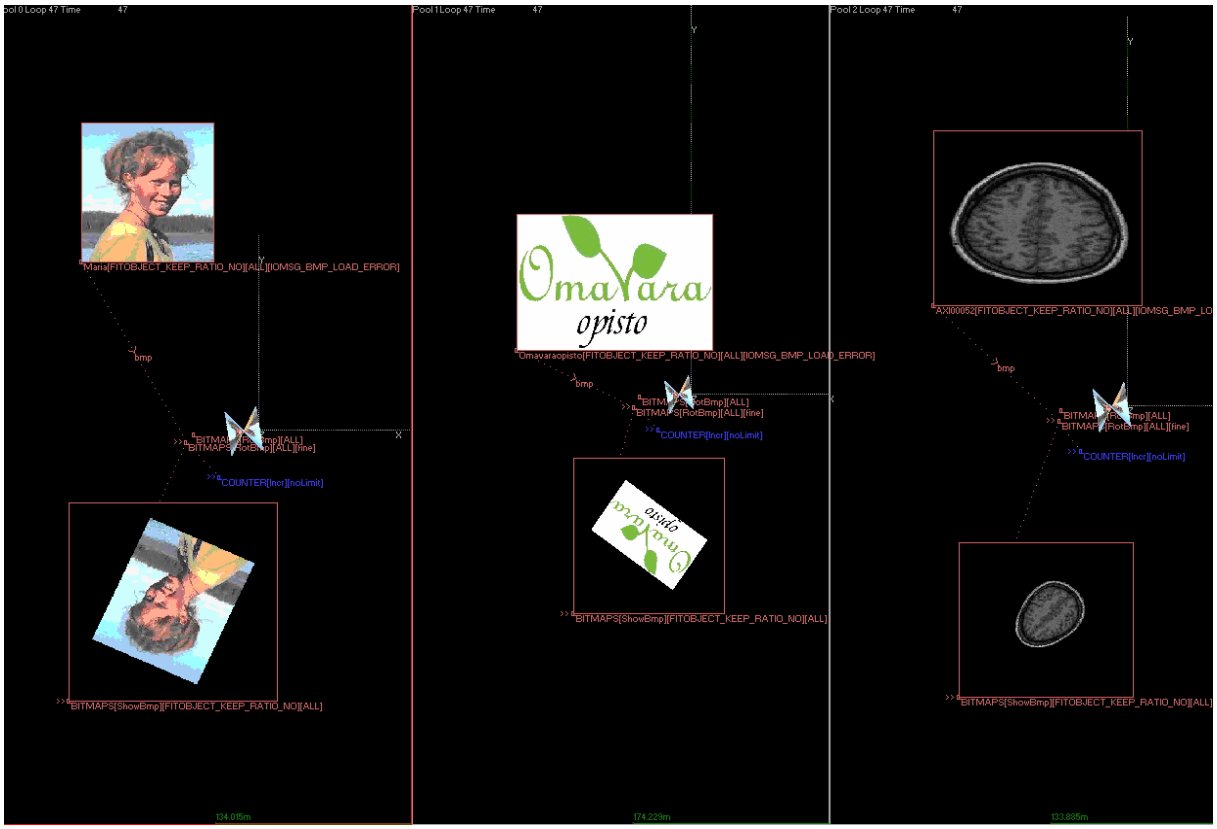
Varatut hiukkaset voivat muodostaa nestemäisiä virtauksia.
Avaa/ Lataa



Sisäänpäinkääntynyt räjähdys (Implosio). Hiukkasten väri vaihtuu nopeuden mukaan. Nopeusvyöhykkeet näyttävät hyvin systeemin ominaisvärähtelyn taajuuden. Nopeutettuna näkyy selkeämmin.
Avaa/ Lataa

Rinnakkais-simulaatiot

- Rinnakkais-simulaatiot ovat erilisiä, mutta niiden väliin voi luoda yhteyksiä. Tärkeää on aika-synkronointi, joka mahdollistaa tarkkailun, miten aika-askelluksen valinta vaikuttaa simulaation tulokseen.
- Visualisoimiseksi erillissimulaatiot voi siirtää samaan kenttään tai värittää . Avaa/ Lataa



Ohjelman tulevaisuus

- Ohjelma on yhden ihmisen ikuisuusprojekti. Valmiiksi se ei koskaan tule ja miten tuon mittakaavan hanke voisi tullakaan? Hienointa on minulle kehitystyö ja saada askel askeleelta vastauksia mietteisiini.
- Ohjelma nojautuu Newtonin mekaniikkaan, joten aika-askellus tuottaa kertyviä virheitä simulaatioon. Näiden vähentäminen on tulevia tutkimuksen aiheita.
- Ohjelma ei ole vielä julkaisukelpoinen. Toteutus on aika omanlainen ja mutkikas. Jotta kokonaisuus pysyy kasassa, on ollut pakko dokumentoida ohjelmani toimintoja itsellenikin. Ulkopuolisille käyttäjille ohjelman käyttö vaatisi huomattavasti lisäohjeistusta.
- Ohjelma on ehtinyt vanhentua tehdessä. Nykykoneet pystyvät (vielä) suorittamaan ohjelmaa emulaattorin kautta, mutta ohjelman nopeus romahtaa tuskastuttavalla tavalla. Jos olet hyvä ja pitkäjänteinen ohjelmoija ja janoat haasteita, ota yhteyttä! Varsinkin porttaamisen ja laiteohjelmoinnin taidot ovat minulla vähissä.
- Julkaistaessa ohjelma olisi tietysti vapaata omaisuutta.
- Toivoisin saavani ohjelman Omavaraopiston metsäyliopiston tutkimus- ja opetusvälineeksi. Jatkokehittelyssä aion laajentaa ohjelman räätälöidysti meidän tarpeisiimme.

Linkkejä

- Tulen päivittämään ajan mittaan sekä tätä esitystä että näitä linkkejä simulaatioiden tuloksilla.
- Simulaatioklippejä (vaiheessa)
- Ruutukaappauksia (vaiheessa)
- Koodi (voi vilkaista. Kirjoitettu omalla tyylillä omaan käyttöön, ei siistitty, muuttuva)

Jälkisanat

- Ottakaa tämä projekti kunnianosoituksena luovuuden ilolle. Rakastan nöyrän ja eettisen tieteenteon asenteita. En arvosta nykyistä ylimielisen tieteen- ja vallan yhteenkietoutumaa. Intohimo ajatteluun! Filosofiaa tupiin! Ja itsekriittistä ryhtiä!
- Nykyinen tieteenteko on pidäkkeettömän soveltamisvetoinen ja eettisiä pohdintoja koetaan lähinnä haitantekona.
- Minulle tiede on ymmärrysvetoista ja siten se katsoo itseään myös ulkopuolelta. Eettinen ajattelu sopii oikein hyvin sen piiriin.
- Kaipaan tutkimisintoa nimenomaan alueille, jotka ovat nykyisen tiedehegemonian lujassa yksinomistuksessa. Erityisesti lääketieteeseen.
- Olen täysin tietoinen että omavaraisuuteen perustuva yhteiskunta ei rakenna tietokoneita. Se olisi hyvä juttu! Ihminen on aina osannut hauskuttaa itseään jo vuosituhansien ajan ennen teollisesti tuotettua ajanvietettä.
- Luovuuden paras paikka on elämässä, ei tietokoneen edessä. En kehtaa mainostaa ohjelmointiharrastuksia ilman että ensin hoidetaan kasvimaat ja saadaan polttopuut sisään. Ensin työ ja sitten huvit. Parasta, jos ne ovat yhtä.
- Tulevaisuuden yliopistoissa tehdään toivon mukaan joskus vastuulista tutkimusta kestävän elämäntavan pohjalta.
- Nouskaamme tynnyreistämme! Tarvitsemme vuoropuhelua ja aluevaltauksia. Teknisesti osaavatkin voivat olla teknologiakriittisiä!

Terveisin, Lasse Nordlund
1.4.2021