

[ANDROID WEAR](#) — [KOODIA KELLOON](#) » [AALTO 1](#) — [SUOMEN SPUTNIK](#) » [KEINOÄLYN KÄYTTÖKOHTEET](#) » [ELITESTÄ ELITEEN](#) — [30 VUOTTA AVARUUSLENTELYÄ](#) » [RETROVAARA](#) — [22 SIVUA AMIGAA!](#)

- 3 Pääkirjoitus**
- 4 Ohjelmointia lasten ehdoilla**
Resurssikeskus Linkissä tenavasta tulee taitaja.
- 6 Nettiskeptikon käsikirja**
Kirjaesittelyssä Virtual Unreality.
- 8 Android Wear – kehitys haltuun**
Näin komennat puettavaa konetta.
- 12 Tekoäly porttamaan MSX-pelit**
Sovellus kääntää sovellusta.
- 15 Kolumni – Janne Sirén**
Tietokonelehden ekosysteemi.
- 16 Opiskelijavoimin avaruuteen**
Suomen ensimmäinen satelliitti syntyy yliopistolla.
- 19 Kolumni – Mikko Heinonen**
Minä ja purjekala.
- 20 Älyä piisirussa**
Tekoäly on monipuolinen ja laajasti sovellettava tieteenala.
- 25 Lukijan projekti: tee-se-itse kannettava**
Arttu Leskelä rakensi Raspberry Pi -kannettavan.
- 26 Tähtenväliset pioneerit**
Avaruus, tuo tuntemattomista maastoista ruuhkaisiin.
- 32 Ainutlaatuinen agenttiseikkailu Covert Action**
Sid Meierin visiossa sota on kylmää.
- 36 Amiga 500 kaapista ulos**
Viritä vanha klassikko käyttökontoon.
- 45 Katsaus Amigan klassikkopeleihin**
Kymmenen suurta muinaista.
- 49 Ei näin!**
Monet pitävät Commodoresta, vaikkei se sitä aina olisi ansainnut.
- 55 Kolumni – Jukka O. Kauppinen**
Minä, Amiga ja Saku 2014.
- 58 Turingin tervakuopat**
Ainutlaatuisia haasteita koodauksen periferiassa.
- 62 Salattua sokeria**
Ohjelmoijan piilottamat pääsiäismunat ilahduttavat löytäjää.
- 65 Kolumni – Jukka O. Kauppinen**
Paras konsoli ei koskaan voita.
- 66 Pelisuunnittelun filosofiaa, osa 3**
Palaute kertoo, mitä pelissä tapahtuu ja miksi.
- 68 Irti vanhasta vihasta**
Harmonia on mahdollista lisenssienkin välillä.
- 70 Historiaa laitetiedostojen takana**
Myös uudet käyttöjärjestelmät kantavat mukanaan menneisyyden painoa.
- 72 Palikoita ruudulle**
Grafiikan olemus riippuu siitä, millainen laitteisto sen esittää.
- 79 Postipalsta**

MIKKOKIVIKALSI

1995 MUISTOISSAMME.





Ville-Matias Heikkilä
päätoimittaja

Skrolli

Tietokonekulttuurin erikoislehti

Yhteydenotot toimitus@skrolli.fi
Ircnet: #skrolli

Päätoimittaja Ville-Matias Heikkilä
Toimituspäälliköt Toni Kuokkanen & Ninnu Koskenalho
Taiteellinen johtaja Risto Mäki-Petäys
Kuvatoimittaja Mitol Meerna
Mediamyynti Jari Jaanto
Talous Anssi Kolehmainen

Muu toimitus Mikko Heinonen, Jukka O. Kauppinen, Ronja Koistinen, Sade Kondelin, Teemu Likonen, Annika Piironen, Antti Pöllänen, Suvi Sivulainen, Kalle Viiri

Tämän numeron avustajat Tapio Berschewsky, Toni Bratinčević, Kenneth Falck, Ville Jouppi, Joonas Järvenpää, Manu Järvinen, Panu Kalliokoski, Antti Kiuru, Sami Koistinen, Markus Kuula, Arttu Leskelä, Ismo Leskelä, Tapio Lehtimäki, Sakari Leppä, Jussi Määttä, Jarto Nieminen, Lauri Nurmi, Manu Pärssinen, Janne Sirén, Aki Sivula Timo Soilamaa, Mikko O. Torvinen, Teija Tuhkio, Antti Ylikoski, Wallu

Julkaisija Skrolli ry

Painopaikka Tammerprint, Tampere,
ISSN 2323-8992 (painettu)
ISSN 2323-900X (verkkojulkaisu)

Emme haikaile menneisyyteen

Skrollia on ilmestynyt kaksi kokonaista vuosikertaa. Nyt on siis hyvä katsoa taaksepäin ja pohtia taaksepäin katsomista.

Kun Skrollia suunniteltiin reilu kaksi vuotta sitten, oli kantavana teemana jatkuvuus. Kun muu tietotekniikkamedia keskittyi ohikiitäviin meemeihin ja kulutustrendeihin, Skrolli päätti keskittyä siihen, mikä kannattaa ikuistaa paperille. Nykyaika perustuu aina historiaan, joten historiaa kannattaa pitää esillä nykyaikasta puhuttaessa. Kulttuurimuoto on vahvimmillaan silloin, kun sillä on vuosikymmeniä jatkuneita perinteitä, joten niistä kannattaa puhua.

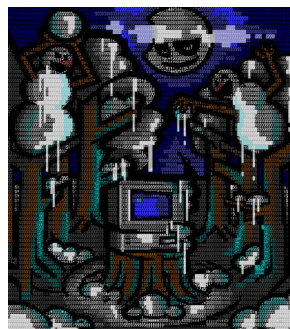
Tämä linjaus johti lehteen, jossa huomattava osa artikkeleista sukeltaa ajallisesti paljon viime vuosia syvemmälle. Osassa näistä on peräti painopiste joko historiassa tai vanhoissa koneissa ja ohjelmissa. Aina silloin, kun lehdessä on tavallista enemmän tämän ryhmän juttuja, saamme kuulla olevamme menneisyyteen haikaileva nostalgialehti.

Latinan sana ”retro” tarkoittaa ”takaisin”. Koska tietokonekulttuurissa sanaa käytetään usein laveasti kaikesta vähän vanhemmasta, kytkeytyy menneisyyteen palaamisen ajatus kuin puolihuomaamatta kaikkeen, missä historia tiedostetaan. Tietokonekulttuurin pohjakivistä ei voi edes puhua, ilman että joku sijoittaa ne mielessään jonkinlaiseen todellisuuspakoiiseen nostalgiakarsinaan.

Kypsemmissä kulttuurimuodoissa tällaista ilmiötä ei ole. Vanhojen kirjojen lukijaa ei kutsuta ”retrolukijaksi”. Elävien soittimien soittajat eivät yleensä haikaile takaisin niille vuosikymmenille, jolloin syntetisaattoreita ei ollut. Tietokonekulttuurissa sen sijaan kaikki nähdään etupäässä markkinavetoisen vaihtuvuuden kautta: se mikä on jo suunnitellusti vanhennettu, on kuollutta ja kelpaa vain muisteluun ja haikailun kohteeksi.

Tämän numeron painotus on samaan suuntaan vinksallaan kuin edellisenkin. 80 sivusta neljännes on Amiga- ja Commodore-asiaa. Lohdutukseksi vanhoihin koneisiin kyllästyneille kerrotakoon kuitenkin, ettei putki jatku enää ensi numerossa. Monet nykitekniikkaa ja käytännön tekemistä käsittelevät artikkelit lykkääntyivät seuraavaan lehteen, joten Skrollin ensi vuosi pääsee alkamaan aivan toisenlaisissa merkeissä.

Skrollin toimitus toivottaa hyviä pyhiä kaikille lukijoille, ja palaamiin ensi vuonna! 🐉



Kannen kuva:
H7 / Break & Block7ronics



441 878
Painotuote



Ohjelmointia lasten ehdoilla

Helsingin yliopistolla, tietojenkäsittelytieteen laitoksen yhteydessä toimiva Linkki-resurssikeskus on järjestänyt lapsille ohjelmointitoimintaa ja opettajille koulutusta vuodesta 2011 lähtien. Tarkoituksena on toisaalta välittää ohjelmoinnin hauskuus ja into eteenpäin seuraaville sukupolville, toisaalta kehittää hyviä toimintamalleja ohjelmoinnin opetukseen.

Teksti: Panu Kalliokoski Kuvat: Maija Pollari / LUMA-keskus

Kesäisin Linkki järjestää intensiivisiä ohjelmointileirejä ja kouluaikana viikoittain koontuvia ohjelmointikerhoja. Linkki tarjoaa myös erilaisia polkuja eteenpäin osallistumalla tietojenkäsittelytieteen laitoksen kilpaohjelmointitoimintaan ja tekemällä yhteistyötä MOOCin kanssa, joka tarkoittaa tietojenkäsittelytieteen laitoksen kaikille avoimia ohjelmointikursseja.

Vuosien mittaan Linkissä on kertynyt paljon kokemusta siitä, miksi ja eritoten miten lapsille kannattaa ohjelmointia opettaa. Niinpä Linkki on merkittävä asiantuntemusta ja resursseja tarjoava taho. Tällä hetkellä Linkin toimintaa rajoittaa eniten käytettävissä olevien ihmisresurssien ja vapaaehtoisten määrä.

Miksi ohjelmointia lapsille?

Lapset ovat luonnostaan luovia ja haluavat kokeilla monenlaista. Mitä lapset luovat, riippuu heidän käytössään

olevista välineistä ja heidän saamistaan virikkeistä. Jos lapsella on kiviä, hän rakentaa niistä jotain; jos Minecraft-ohjelma, hän tekee virtuaalimaailmoja; mutta jos lapselle antaa ohjelmointityökalut, hän pystyy tekemään niillä lähes mitä vain. 7–9 vuoden ikä on erittäin otollinen ohjelmoinnin aloittamiseen, mutta ei haittaa olennaisesti, vaikka sen aloittaisi aiemmin tai myöhemmin.

Ohjelmoinnin rajattomuudesta seuraa kaksi tärkeää asiaa. Ensinnäkin lapset pystyvät ohjelmointileikissä oppimaan paljon sellaista, mitä ei pysty oppimaan mistään muualta, ja monta muuta asiaa helpommin kuin millään muulla tavalla. Esimerkiksi muuttujan käsite opitaan ohjelmoinnissa työvälineenä, asiana, jolla on oikeasti mahdollista helpottaa omaa työtään – eikä vaikeana konseptina, jota tarvitaan



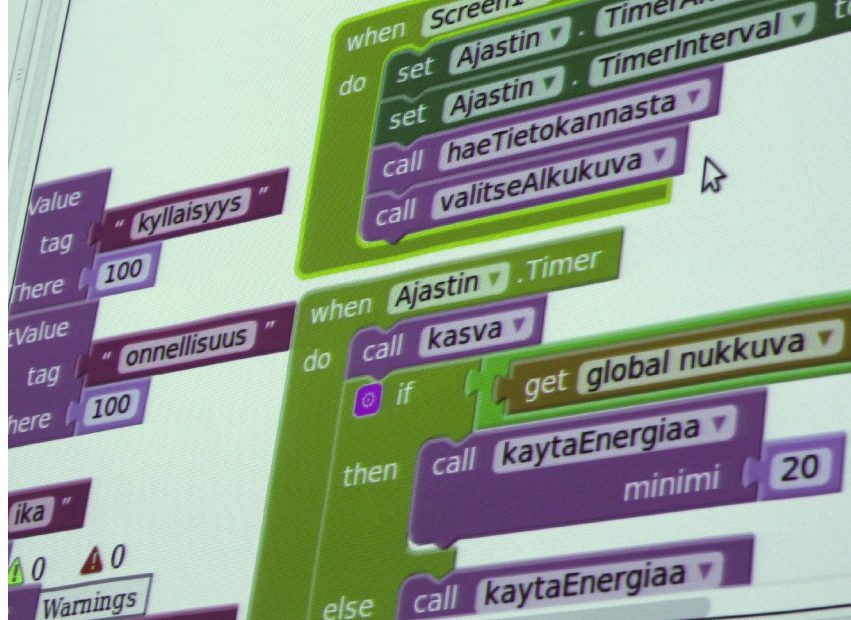
tietyntylaisista laskutehtävistä puhumi- seen. Vain ohjelmoimalla voi oppia, miten tehdään tyhjistä sävelmä, kuva, animaatio, peli tai maailma käyttäen pelkkiä sanoja, ja kuinka pienikin ero näissä sanoissa vaikuttaa lopputulok- seen. Ohjelma on kuin loitsu, joka tee- kee asioita oleviksi vain selittämällä, millaisia ne ovat.

Toiseksi, koska ohjelmoimalla voi tehdä mitä vain, on sitäkin tärkeäm- pään antaa aina uusia mielenkiintoisia tehtäviä ja haasteita, koska suurin osa ohjelmoinnin mahdollisuuksista ei ole käsin kosketeltavissa, eivätkä ne tule ilmi ohjelmointiympäristöistä. Esi- merkiksi se, mihin kaikkeen aliohjel- maa voi käyttää ja miten rakennetaan rekursio, avautuu vasta pitkän ajan ja monien esimerkkien jälkeen. Ohjaajal- la tai opettajalla on suuri vastuu, että asiat esitellään sellaisessa järjestyksessä ja niin, että lapsi pystyy heti näkemään niiden hyödyn ja kokeilemaan niitä erilaisissa tilanteissa.

Ohjelmoinnin opetusta on hyvä ke- hittää nyt, kun 1980-luvun ohjelmoi- jasukupolvi on kasvanut siihen ikään, että heillä on omia lapsia. Tämän edellisen sukupolven ohjelmoijilla on omakohtaista kokemusta siitä, miten ohjelmointia opitaan pienestä pitäen.

Miten ohjelmointia pitää opettaa?

Linkki on toimintansa aikana kokeillut monenlaisia työvälineitä ja lähestymis- tapoja ohjelmointiin. Arvot ovat kui- tenkin pysyneet samoina: lapsen kun- nioittaminen, omaehtoinen toiminta ja luovuuden ilo.



Lopputulokseen liittyvät motiivit, siis esimerkiksi halu tehdä hieno peli, ovat kylläkin aitoja motiiveja, mutta ohjelmoinnissa lapsilla ei ole mitään keinoa arvioida tavoitteidensa haasta- vuutta. Riippuu iästä ja temperamen- tista, riittääkö motivaatio loppuun asti vai onko innolla aloitettu projekti lo- puksi pettymys. Lisäksi tiettyyn loppu- tulokseen pyrkivä lapsi tarvitsee paljon ohjaajan apua, jolloin oppiminen alkaa muuttua opettajavetoiseksi.

Parhaan oppimistuloksen saa, kun lapsen motivaatio on ohjelmointitoi- minnassa itsessään. ”Jos laitankin nämä käskyt näin päin, mitähän siitä seuraa?” ”Voisinkohan liikutella tätä uutta kuvaa aikaisemmin tekemälläni animaatio-oh- jelmalla?” Tällaisia kysymyksiä pohties- sa oppiminen on parhaimmillaan, aitoa leikkiä, jossa lapsi (tai aikuinen) halti- oituu jatkuvasti omien kättensä töistä ja haluaa siksi tehdä lisää.

Koska leikki on paras motivaatio, tarvitaan ohjelmointiympäristöjä, jois- sa ohjelmien tulokset on helppoa näh- dä (tai kuulla, tuntea, haistaa) ja jois-

sa asioita on helppoa kokeilla. Tämä tarkoittaa käytännössä, että kaikki ylimääräinen vaiva on liikaa. Lapsen ei pidä joutua miettimään syntaksin välimerkkejä, ohjelmien tallettamista ja ajamista (puhumattakaan kääntämi- seen liittyvistä komennoista), kryptisiä virheilmoituksia tai mystisiä ohjel- mointiympäristön kaatumisia.

Tähän mennessä parhaat ohjel- mointiympäristöt ovat olleet Scratch ja Logo, jotka molemmat ovat saaneet alkunsa MIT:n laboratorioista – ensin mainittu 2000-luvulla, jälkimmäinen jo 1970-luvulla. Linkin ohjelmoinnissa on käytetty myös esimerkiksi Python- kieltä, Javaa ja Turtle Roysta.

Tule mukaan

Linkillä ei ole muodollista prosessia rekrytoimiseen, mutta mukaan toi- mintaan pääsee esimerkiksi liittymäl- lä IRCnetin kanavalle #linkki. Yksi mahdollinen tapa olla mukana Linkin toiminnassa on kokeilla tulla apuoh- jaajaksi kerhoon, mutta myös tiedotta- jille, oppimateriaalin kirjoittajille, uu- sien ohjelmointi-ideoiden esittäjille ja verkostoitujille on tarvetta. Lapset il- moittautuvat Linkin leireille ja kerhoi- hin Linkin www-sivun kautta ([http:// linkki.cs.helsinki.fi/](http://linkki.cs.helsinki.fi/)), ja mukaan ote- taan kaikki, jotka suinkin mahtuvat.

Tärkeää on tietenkin, ettei ohjel- moinnin opetus jää vain yksin Linkin vastuulle. Kannattaa harkita, voisiko perustaa oman ohjelmointikerhon esimerkiksi työpaikalleen tai lasten- sa kouluun. Linkki voi toimia tällöin myös foorumina, jossa voi vaihtaa ko- kemuksia siitä, mikä toimii ja mikä ei.

Tämä on yksi tärkeä Linkki verkos- tossa, joka toivottavasti ajan mittaan antaa tuleville sukupolville sekä väli- neet että halun tehdä maailmasta sel- lainen kuin he tahtovat. 🌱



Nettiskeptikon käsikirja

Uutuuskirja porautuu sähköisen tiedon ja internetin muodostaman ”virtuaalisen epätodellisuuden” ongelmakohtiin. Mikä muuttuu, kun painokoneet ja kirjastovirkailijat jäävät historiaan?

Charles Seife: *Virtual Unreality: Just Because the Internet Told You, How Do You Know It's True?* (Viking, 2014)

Teksti: Jussi Määttä

Kuva: Nick Misani

Digitaalinen maailman-sota on käynnissä. Sen kaikki osapuolet himoitsevat suurta palkintoa: valtaa muovata sosiaalista elämäämme, manipuloida uskomuksiamme ja vaikuttaa todellisuuskuvaamme. Näin ainakin väittää New Yorkin yliopiston journalismin professori Charles Seife juuri ilmestyneessä kirjassaan *Virtual Unreality*.

Seifen mukaan sota on pohjimmiltaan seurausta digitaalisen informaation luonteesta ja tiedon määrän kasvun aiheuttamista ongelmista. Mistä oikein on kyse?

Tieto on tauti

Tautiepidemian leviämiseen vaikuttaa moni seikka: Kuinka helposti tauti leviää ihmisestä toiseen? Kauanko tarttuva vaihe kestää? Kuinka paljon tartunnan saaneet ovat tekemisissä muiden kanssa?

Seife sovittelee samanlaista luokitte-lua eri informaatiokanaville. Perimätieto leviää heikosti ja yleensä muuttuu matkan varrella. Kiveen hakatut kirjaimet kestävät, mutta kopiointi ja kuljetus on työlästä. Painettu sana oli melkoinen mullistus, mutta fyysinen kirja ei kuitenkaan matkaa paikasta toiseen ihmistä nopeammin.

Internetin kautta leviävä informaatio räjäytti pankin. Tieto kulkee nyt nopeammin maapallon toiselle puolelle kuin naapurin tädille. Nettiin kerran tuotua tietoa on vaikea tuhota, vaikka haluaisikin, ja laajat verkostot pitävät huolen siitä, että informaatio voi välittyä viruksen lailla miljoonille.

Seife kompastelee hieman puhu-essaan yhtäällä tiedosta sinänsä ja

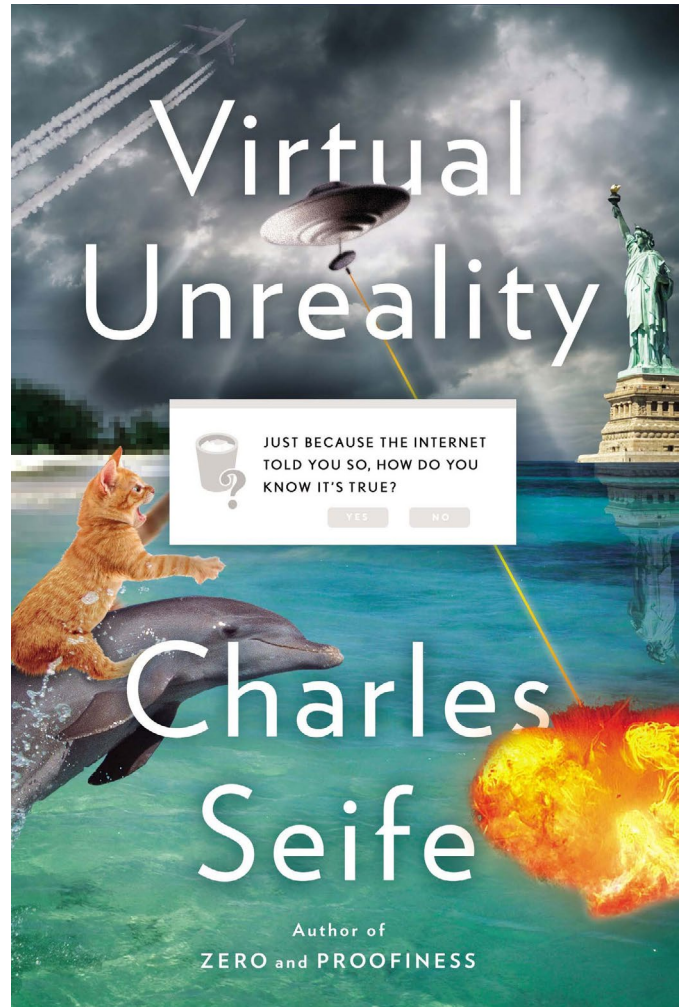
toisaalla sen levityskeinoista, sillä informaatioon itsessään ei vaikuta, kuinka moni sen tietää tai kuinka vauhdikkaasti se etenee. Epidemiologiasta lainattu jaot-telu on kuitenkin hauska oivallus ja mahdollistaa informaatiokanavien eri aspektien vertailun.

Valheesta totta

Digitaalinen informaatio ei ole koskaan lopullista. Siihen voi tehdä muutoksia, joita ei välttämättä voi jälkikäteen mitenkään havaita. Tämä aiheuttaa ongelmia, joita ei paperimedian kanssa synny.

Kirjassa kuvaillaan seuraava esimerkkitapaus: Vuonna 2012 baseballin ammattilaispelaaja Mike Troutin Wikipedia-sivulle lisättiin pilailumielessä keksitty lempinimi ”Millville Meteor”. Muutaman viikon kuluttua *Newsday*-lehden toimittaja käytti lempinimeä urheilu-uutisessa. Nykyään Wikipedia viittaa kyseiseen uutiseen lempinimen lähteenä. Trout on ilmeisesti hyväksynyt saamansa lisänimen. Anonyymi wikimuokkaus muuttui lähdeviittein todistetuksi faktaksi (kuva 1).

Seife kehottaa suhtautumaan Wikipediaan kuin vanhaan, vähän höpsäh-täneeseen setään, joka tietää paljon



kaikenlaista jännää ja jolla voi olla värikkäitäkin mielipiteitä. Tällaista setää on hauska kuunnella, mutta silloin kun tosiasioilla on merkitystä, on paras varmistaa faktat muualta.

Ilmainen tulee kalliiksi

Internet yhdistetään usein sananvapauteen. Seife kuitenkin huomauttaa, että länsimaissa sana oli vapaa jo kauan ennen netin yleistymistä. Internetin merkitys on hänen mukaansa pikemminkin siinä, että verkossa ajatuksilleen saa helpommin yleisöä. Ja jos ei saakaan, niin ainakin oma blogi luo ilmaisen illuusion siitä, että joku kuuntelee.

Kun informaation julkaisemisen kustannukset häviävät, muuttuu samalla perinteisen tutkivan journalismin asema. Kakkua aletaan jakaa heti uuden artikkelin ilmestyttyä: Kilpailevat verkkolehdet ja blogit julkaisevat jutusta katkelman ja linkittävät alkuperäiseen. Joskus sama tarina kirjoitetaan uudestaan omin sanoin eikä skua-pin ensilähteeseen viitata ollenkaan.

”Internet-sukupolven iskulauseen mukaan ’tieto haluaa olla vapaata’. Tiedon tuottaminen ei kuitenkaan ole ilmaista. Tästä syntyy ongelma.”

Vievätkö näiden ”jälleenmyyjien” saamat mainostulot osan toiselle kuuluvasta potista?

Seife kiteyttää: ”Internet-sukupolven iskulauseen mukaan ’tieto haluaa olla vapaata’. Tiedon tuottaminen ei kuitenkaan ole ilmaista. Tästä syntyy ongelma.”

Toisin kuin vaikkapa sanomalehdessä, netissä voi ongelmitta julkaista rajattomasti informaatiota lähes ilmaiseksi. Seifen mukaan tästä hyötyvät eniten ne, jotka tehtailevat suurimman määrän sisältöä. Määrä on oleellisempaa kuin laatu.

Journalismin ahdinko johtaa kirjan mukaan siihen, että artikkelit perustuvat yhä useammin pelkkiin lehdistötiedotteisiin. Seife kertoo esimerkin vuonna 2013 julkaistusta tutkimuksesta, jossa simpanssien väitettiin olevan ihmisten tapaan anteliaita. Suuri osa aiheesta raportoineista nettilehdistä ei vaivautunut tekemään minkäänlaista taustatyötä. Kävikin ilmi, että tutkimusta ei ollut vielä vertaisarvioitu tieteen käytäntöjen mukaisesti ja kuka tahansa ulkopuolinen asiantuntija olisi voinut osoittaa tutkimuksen monet ongelmat. Lukijoiden on siis opittava entistä skeptisemmiksi.

”Meidän täytyy muuttaa suhdettamme informaatioon ja ryhtyä skeptisemmiksi ja kyynisemmiksi.”

Roskaa roboteille

Informaation määrä kasvaa hurjaa vauhtia. Arjestamme tallentuu yhä enemmän tietoa, ja netti tarjoaa jokaiselle mahdollisuuden saada ainakin kuviteltua yleisöä jokaiselle mieleen juolahtaneelle aivoitukselleen. Seuraus on ”informaationälänhätä”, Seife kirjoittaa. Informaation määrä kasvaa eksponentiaalisesti mutta hyödyllisen informaation määrä vain lineaarisesti, hän väittää. Näin ollen arvokkaat tiedonmuruset hukkuvat kohinaan.

Tiedon löytämisen tueksi on tarjolla erilaisia hakukoneita, jotka yrittävät siivilöidä informatiivirrasta esiin tärkeimmät hippuset, mutta riippuvuutemme niistä aiheuttaa Seifen mukaan sen, että sisältöä tuotetaan ihmisten sijaan algoritmien luettavaksi. Esimer-

kiksi monet tunnetut sanomalehdet vaihtavat tekstiensä otsikot suoraviivaisemmiksi nettijulkaisuissaan, jotta Googlen algoritmit varmasti ymmärtäisivät, mistä aiheesta on kulloinkin kyse.

Seifen mukaan julkaisijoiden tavoitteena ei enää ole houkutelua lukijoita laatutekstillä vaan pikemminkin huijata kävijöitä sivulle mainosten näyttämistä varten.

Virtuaaliporkkanan pauloissa

Informaatioon liittyvien ongelmien lisäksi Seife arvostelee pelejä ja verkkopalveluita, jotka saavat ihmiset käyttämään aikaansa omalta kannaltaan epäedullisesti.

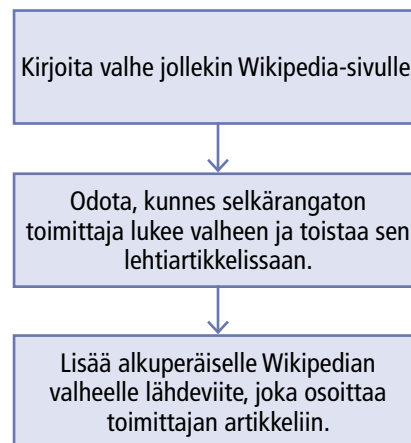
Esimerkiksi Facebook-peli *Farmville* nousi takavuosina suureen suosioon maailmanlaajuisesti. Kirjan kuvauksesta päätellen pelin pelaaminen on käytännössä loputonta hiiren naksutusta, joka johtaa virtuaalisen maatalan laajenemiseen. Pelaaja voi säästää omaa työtään painostamalla tuttaviaan klikkailemaan puolestaan – tai tietysti maksamalla pelin tekijöille.

Ian Bogost, tutkija Georgian teknillisestä korkeakoulusta (Georgia Tech), halusi osoittaa Farmvillen älyt-

ömyyden kristallinkirkaalla tavalla. Hän suunnitteli oman ”pelinsä”, *Cow Clickerin*, jonka tarkoituksena oli näyttää Farmvillen todelliset kasvot kärjistyksen kautta. Cow Clickerissä pelaaja kävi klikkaamassa lehmää kuuden tunnin välein ja sai virtuaalisia palkintoja. Rahalla saattoi hankkia lisäklikkauksia tai vaikka kääntää lehmänsä katsomaan toiseen suuntaan. Seife kuvaa hyvin Bogostin yllätystä ja suuttumusta, kun Cow Clicker saikin aktiivisen, pelin ironiaa ymmärtämättömän seuraajiston.

Toinen ällistyttävä esimerkki on indie-peli *Curiosity*, jossa pelaajat hajottivat yhdessä suurta kuutiota klikkailemalla sitä kuukausien ajan. Lopulta viimeinen klikkaaja sai palkinnon.

Seife rinnastaa klikkailupelit psyko-



Kuva 1. Kuinka valhe muutetaan todeksi Wikipedian avulla.

logi B. F. Skinnerin klassisiin kokeisiin, joissa eläimiä opetettiin erilaisiin tehtäviin palkitsemalla ja rankaisemalla. Ihmisten ehdollistamisessa käytetään hyväksi sosiaalista painetta. Kukaan tuskin olisi pelannut Curiositya tai Farmvillea yksinään, mutta ilmeisesti muiden mukanaolo vahvistaa manipulaation tehoa.

Seife kehottaa ottamaan hieman etäisyyttä tietokoneeseen ja älypuhelimien ja katsomaan sitten tarkkaan. Oman etunsa saattaa huomaamattaan tulla uhranneeksi liikeyritysten hyväksi.

Mikä neuvoksi?

Kirjan viesti kiteytyy vasta viimeisillä sivuilla. ”Meidän täytyy muuttaa suhdettamme informaatioon ja ryhtyä skeptisemmiksi ja kyynisemmiksi”, Seife kirjoittaa. Kun tiedon järjestämisen vastuu siirtyy kirjastonhoitajilta, toimittajilta ja muilta ammattiryhmiltä suuryrityksille, joudumme hänen mukaansa käyttämään enemmän aikaa ja energiaa väistääksemme meihin kohdistuvaa manipulointia. Konkreettisia ehdotuksia kirjasta ei juuri irtoa. Taus-tojen tutkiminen ja faktojen tarkistaminen on neuvona ilmeinen, mutta riittääkö se?

Virtual Unreality julistaa olevansa skeptikon käsikirja digitaalisen informaation maailmaan, mutta on pikemminkin viihdyttävä kokoelma löyhästi teemoittain jaoteltuja anekdootteja digitaalisen maailman ongelmista ja epäkohdista. Selkeää teesiä Seife ei onnistu muotoilemaan, vaikka onkin löytänyt hyviä aineksia ja ideoita. Teos on kuitenkin rattoisaa lukemista ja saa lukijan ajattelemaan. 🐘



Android Wear -kehitys haltuun

Älykellot iskivät suuren yleisön tajuntaan tänä syksynä, kun Apple julkaisi ensi vuonna ilmestyvän Apple Watch -mallistonsa. Sitä ennen Google toi jo kesällä markkinoille Android Wear -käyttöjärjestelmän, johon liittyivät samanaikaiset Samsungin, Motorolan ja LG:n älykellojulkistukset.

Teksti: Kenneth Falck

Kuvat: Mitol Meerna, valmistajat, Kenneth Falck

Alykelloja on ollut ole-massa ennenkin. Niistä ehkäpä tunnetuin on vuonna 2013 julkaistu Pebble, joka rahoitti ensimmäisen mallinsa Kickstarter-kampanjalla. Isommilla valmistajilla, kuten Samsungilla, on ollut tarjolla myös Android Wearia edeltäviä malleja.

Tällä hetkellä Microsoft on ainoa iso peluri, joka ei ole vielä lähtenyt mukaan älykellokisaan. Huhujen mukaan julkistuksia on tulossa lähiaikoina, joten tämän artikkelin ilmestyttyä Windowsin saattaa jo saada ranteeseensa.

Nykyisen sukupolven älykellot ovat vahvasti liitoksissa älypuheliin. Kaikissa kelloissa on Bluetooth Low Energy -rajapinta, jonka kautta ne vastaanottavat viestejä (notifications) puhelimessa pyörivästä kumppanisoluvelluksesta. Kelloilla ei käytännössä tee mitään ilman yhteensopivaa äly-

puhelinta, sillä nettiyhteys ja älykkyys on ulkoistettu lähes kokonaan puheli-meen.

On ennustettavissa, että älykellojen kysyntä tulee seuraamaan älypuhelin-tien markkinaosuuksia, koska laitteiden käyttöjärjestelmät ovat sidoksissa toisiinsa. Android-käyttäjät hankkivat Android Wear -kellon, iPhone-käyt-täjät Apple Watch -kellon ja niin edel-leen. Pebblen kaltaisille itsenäisille toimijoille ei jää kovinkaan paljon ja-lansijaa.

Android Studiolla alkuun










Tässä artikkelissa käsitellään ensisijai-sesti Android Wearia, koska se on ai-noa jo nyt saatavilla oleva uusimman sukupolven älykelloalusta. Amazon toimittaa Suomeen sekä Samsungin että LG:n kelloja, ja muitakin malleja voi jo ennakotilata.

Jos hankkii kellon kehityskäyttöä varten, on syytä varmistua sen käyt-

töjärjestelmästä. Esimerkiksi Samsun-gin Gear 2 ja Gear 2 Neo -malleissa on oma epäyhteensopiva Tizen-käyt-töjärjestelmänsä. Samsung Gear Live puolestaan on Android Wear -kello, ja kyseistä mallia on käytetty kehitysalus-tana tämänkin artikkelin taustatyötä tehtäessä.

Kellon lisäksi kehitystä varten tar-vitaan riittävän tuore älypuhelin. An-droid Wear vaatii toimiakseen vähin-tään Android 4.3 -puhelimien. Monet kaupoissa myytävät puhelinmallit käyttävät edelleen Androidin 4.2-ver-siota, eikä niitä voi parittaa älykelloon. Googlen Nexus-puhelimet ja -tabletit ovat yleensä turvallisin valinta kehitys-käyttöön, sillä niissä ei ole ylimääräisiä valmistajien laajennuksia ja päivitykset tulevat niihin heti saataville.

Android Wear -sovelluksia keHITE-tään Android Studio -ohjelmalla, jon-ka beta-versio julkaistiin kesällä. Enää ei siis tarvitse asentaa Eclipseä eikä

Kuva	Malli / OS	Kieli	RAM / Flash	Resoluutio	Akku	Näyttö / sensorit / yhteydet
	Pebble FreeRTOS	C	128 kt 4 Mt	144 × 168	130 mAh	1,26" e-paper Kiihtyvyyssanturi, kompassi, valosensori Bluetooth 2.1 / 4.0
	Pebble Steel FreeRTOS	C	128K kt 8 Mt	144 × 168	130 mAh	1,26" e-paper Kiihtyvyyssanturi, kompassi, valosensori Bluetooth 2.1 / 4.0
	Samsung Gear Live Android Wear	Java	512 Mt 4 Gt	320 × 320	300 mAh	1,63" AMOLED 9X, sykemittari Bluetooth 4.0
	Motorola Moto 360 Android Wear	Java	512 Mt 4 Gt	320 × 290	320 mAh	1,56" LCD (pyöreä) 9X, sykemittari, valosensori Bluetooth 4.0
	LG G Watch Android Wear	Java	512 Mt 4 Gt	280 × 280	400 mAh	1,65" IPS LCD 9X Bluetooth 4.0
	LG G Watch R Android Wear	Java	512 Mt 4 Gt	320 × 320	410 mAh	1,3" P-OLED (pyöreä) 9X, sykemittari, ilmapuntari Bluetooth 4.0
	Asus Zenwatch Android Wear	Java	512 Mt 4 Gt	320 × 320	1.4 Wh	1,63" AMOLED 9X, sykemittari Bluetooth 4.0
	Sony Smartwatch 3 Android Wear	Java	512 Mt 4 Gt	320 × 320	420 mAh	1,6" LCD 9X*, valosensori, GPS Bluetooth 4, NFC
	Apple Watch Watch OS	?	?	?	?	1,7" / 1,5" Retina 9X, sykemittari, ilmapuntari Bluetooth 4, NFC, Wi-Fi

* 9X (9-axis sensor) sisältää kiihtyvyyssanturin, gyroskoopin ja kompassin

Taulukko 1. Markkinoilla olevat ja pian ilmestyvät älykellot.

mutakaan erillisiä työkaluja. Android Studio huolehtii tarvittavien SDK-pakettien ja apukirjastojen asentamisesta ja sen käyttöönottoon löytyy kattava ohjeistus Googlen sivuilta.

Sovellusten kehittämisen voi aloittaa, kun Android Studio, älykello ja sopiva älypuhelin on hankittu. Jos kellon hankkiminen ei ole vielä mahdollista, niin sovelluksia voi kokeilla tietokoneen ruudultakin Android Studion emulaattorilla. Silloin ei kuitenkaan saa kovin hyvää tuntumaa siihen, miltä sovelluksen käyttö oikeasti tuntuu.

Yksinkertaisia sovelluksia

Omista sovelluksista ei kannata suunnitella liian kunnianhimoisia. Käyttäjän into älykellon räpläämiseen katoaa usein laitteen oltua hetken ranteessa. Arkikäytössä kellon näyttöä tulee vilkaistua lähinnä kellonaikaa tarkistaakseen tai värinän herättäessä huomion.

Älykello sopii parhaiten sellaisiin toimenpiteisiin, jotka edellyttävät vain ranteen kääntämistä ja kellotaulun vil-

kaisemista. Käyttäjä ei halua investoida näihin asioihin omaa huomiotaan kuin muutaman sekunnin verran. Isommat askareet on helpompi hoitaa kännykällä, jonka saa tarvittaessa vedettyä esiin taskusta.

Tämän vuoksi suosituimmat älykellosovellukset tulevat todennäköisesti olemaan tilanteista riippuvia ja muutenkin älykkäitä notifikaatioita. Hyvä älykellosovellus osaa ennakoida käyttäjänsä aikeet ja näyttää juuri oikean tiedon oikeaan aikaan kellon ruudulla. Jos käyttäjän huomio täytyy herättää, se hoidetaan vienolla ranteen värähdyksellä.

Android Wearin käyttöliittymä muodostuukin pääasiassa notifikaatioista, jotka ovat ulkoasultaan melko yksinkertaisia valkoisia kortteja. Kun jotain kiinnostavaa tapahtuu, ruudulle putkahtaa uusi kortti. Kortteja voi selaila ja sulkea, ja osasta niistä pääsee navigoimaan myös aiheeseen liittyvään monipuolisempaan koko ruuduun sovellukseen.

Notifikaatioiden lisäksi on paljon kysyntää erilaisille hauskoille ja hienoille kellotauluille. Kellotaulu on vähän kuin kännykän soittoaani – jokainen haluaa valita käyttöönsä juuri omanlaisensa. Varsinkin lapset ja nuoret tulevat varmasti innostumaan erilaisten piirroshahmojen ja suosikkiartistien kellotauluista.

Notifikaatiot haltuun

Android Wearista puuttuu toistaiseksi virallinen rajapinta kellotaulujen tekemiseen. Niitä voi kyllä tehdä, mutta kehitys on monimutkaista. Virallista rajapintaa odotellessa kannattaa etsiä GitHubista projekti nimeltä twotoasters/watchface-template, joka tarjoaa tähän tarkoitukseen valmiin pohjan. Projektipohjaan voi vaihtaa taustakuva- ja viisareiksi omat PNG-tiedostonsa, joita järjestelmä sitten kääntelee sekuntien, minuuttien ja tuntien tahtiin.

Notifikaatioiden suhteen Android Wear -kehitys jakaantuu kahtia sen

mukaan, haluaako kelloon ladata oman sovelluksen vai ainoastaan näyttää ruudulla geneerisiä ilmoituksia. Sovelluksen lataaminen on käyttäjälle huomaamaton ja helppoa, joten sitä ei kannata pelätä. Wear-sovellus asennuu kelloon automaattisesti kännykäsovelluksen asentamisen yhteydessä.

Geneerisillä ilmoituksilla pääsee nopeimmin liikkeelle, jos haluaa ensin kokeilla jotain pientä. Tarvittavan ohjeistuksen löytää Googlesta haksanalla *NotificationCompat.Builder*. Kyseistä rajapintaa kutsutaan puhelimen puolella pyörivässä sovelluksessa silloin, kun ilmoitus halutaan esittää käyttäjälle.

Puhelimen rajapinnalla luotu notifiikaatio putkahtaa näkyviin kellon ruudulle. Yksinkertaisimmillaan ilmoitus koostuu vain otsikosta, ikonista ja lyhyestä kuvaustekstistä. Näkymästä voi tehdä hiukan monipuolisemman WearableExtender-rajapinnalla, jolla

ilmoitukseen saa liitettyä esimerkiksi taustakuvia, navigaatiohierarkioita ja painikkeita.

Täydellinen vapaus Android Wear -käyttöliittymän suunnitteluun edellyttää koko ruudun kokoisen näkymän luomista. Käyttäjä voidaan ohjata tällaiseen näkymään esimerkiksi silloin, kun hän haluaa yksinkertaisesta notifiikaatiosta tarkempia lisätietoja. Koko ruudun kokoisen näkymän toteuttaminen on haasteellisempaa kuin pelkien ilmoitusten, sillä suunnittelussa on ymmärrettävä älykellojen erilaiset muodot ja pikselikoot.

Pyöreitä ja neliskulmaisia näyttöjä

Tällä hetkellä Android Wear -kellot jakaa karkeasti kahteen osaan: pyöreisiin ja neliskulmaisiin. Muilta osin kellojen ominaisuudet ovat hyvin lähellä toisiaan. Niissä on yhtä paljon RAM- ja Flash-muistia ja suunnilleen samat

sensorit.

Useimmissa kelloista on neliön muotoinen, 320×320 pikselin kokoinen näyttö. LG:n kellossa resoluutiona on kuitenkin 280×280, joten sovelluksia kehittäessään ei voi olettaa niiden näkyvän aina samassa pikselikoossa.

Pyöreissä kelloissa pikselikoot ovat suunnilleen samat kuin neliskulmaisissakin, mutta osa pikseleistä on ”leikattu pois” ympyränmuotoisen kellotaulun ulkopuolelle jäävistä kulmista. Sovelluksia kehitettäessä pitää siis huomioida, että kulmiin sijoittuva grafiikka ei tule näkyviin.

Moto 360 -kellossa on lisäksi erikoisuutena alareunan musta palkki, joka leikkaa pyöreästä kellotaulusta viipaleen pois. Kehittäjälle jää siis hyödynnettäväksi varsin epäsäännöllisen muotoinen näyttöalue, joka on kooltaan 320×290 pikseliä.

Android Studio luo wearable-sovelluksiin kaksi näkymää: neliskulmaisen



Kaavio 1. Kolme tapaa aktivoida Android Wear -sovellus ilmoitusten kautta.

ja pyöreän. Niiden lisäksi sovelluksessa on *WatchViewStub*-tyyppinen isäntänäkymä, joka lataa automaattisesti käyttöön joko neliskulmaisen tai pyörän alanäkymän sen mukaisesti, millaisessa kellossa sovellusta ajetaan.

Näkymissä on syytä hyödyntää Androidin monipuolisia taitto-ominaisuuksia, joilla käyttöliittymäelementit voi sijoittaa ruudulle pikseleistä riippumattomasti. Elementtejä voidaan tarpeen mukaan keskittää, kohdistaa reunoihin, sijoitella vierekkäin, allekkain ja niin edelleen.

Itsenäisiä kellosovelluksia

Android Wear -käyttöjärjestelmän päävalikosta löytyy painike nimeltä ”Start”. Tällä painikkeella voi käynnistää minkä tahansa kelloon asennetun sovelluksen. Android Studio luomasta sovelluksesta aktivoituu tällöin pyöreä tai neliskulmainen oletusnäkymä.

Googlen Play Storesta löytyy jonkin verran pelejä ja sovelluksia, jotka on tarkoitettu käynnistettäväksi Start-valikosta. Valikossa on myös muutamia sisäänrakennettuja apuvälineitä, kuten kompassi ja taskulaskin. Niitä tulee kuitenkin käytettyä aika harvoin. Samat asiat hoitaa yleensä mukavammin kännykällä.

Itsenäinen kellosovellus voidaan aktivoida Start-valikon sijaan myös *MessageApi*-rajapinnan avulla. Sovellukseen lisätään näkymätön Java-luokka, joka kuuntelee puhelimesta tulevia viestejä. Kun viesti saapuu, luokka aktivoituu ja avaa kellon näytölle ilmoituksen. Ilmoitus luodaan samalla *NotificationCompat*-rajapinnalla, jota käytetään puhelimenkin puolella.

Etuna *MessageApi*-rajapintaa käytettäessä on se, että kellosovelluksen paikallisesti luoma ilmoitus voi avata lisätietonäkymän kellon ruudulle. Jos sama ilmoitus luodaan puhelimen puolella, lisätietonäkymäkin aukeaa puhelimesta. Kaaviossa 1 esitellään eri tavat aktivoida sovellus joko puhelimesta tai kellosta käsin.

Sensorit ja vuorovaikutteisuus

Kaikissa markkinoilla nykyisin olevissa Android-kelloissa on yhdeksän vapausasteen anturi, johon sisältyy kompassi, gyroskooppi ja kiihtyvyyssanturi. Joistakin malleista löytyy

myös sykemittari, valosensori ja ilmapuntari. Sovelluksilla on siis mahdollisuus reagoida varsin laajaan kirjoon erilaisia ympäristön tapahtumia.

Antureiden tiloja voi lukea *SensorManager*-rajapinnalla. Esimerkiksi sykemittariin pääsee käsiksi avaamalla *TYPE_HEART_RATE*-anturin. Kellosovellus voi pyytää käyttöjärjestelmää mittaamaan käyttäjän sykkeen jatkuvasti taustallakin ollessaan. Tällöin sovellus saa tiedon sykkeestä noin sekunnin välein ja voi reagoida sen muutoksiin avaamalla ruudulle notifi kaation, josta pääsee siirtymään sovelluksen käyttöliittymään.

Sama periaate pätee muihinkin sensoreihin. Esimerkiksi virtuaalinen *TYPE_STEP_COUNT*-sensori yhdistää liikeanturien tiedot ja laskee käyttäjän ottamat askeleet. Sovellus voi hiljalleen kerätä sensoridataa taustalla, kunnes koittaa oikea hetki aktivoitua ja herättää käyttäjän huomio.

Sensoreiden käytöstä löytää paljon esimerkkejä googlaamalla hakusanaa *SensorManager*. Android-puhelinten esimerkit pätevät Android Wear -sovelluksiin sikäli kuin kellosta löytyvät samat sensorit kuin puhelimestakin. Sensoreita ohjelmoidessa on syytä huomioida kellojen rajallinen akun kesto. Sovelluksissa kannattaa hyödyntää sensorien batch-ominaisuuksia, jos dataa ei tarvitse saada reaaliaikaisena. Batch-tilassa laite kerää sensoridataa vähävirtaiseen puskurimuistiin ja aktivoi enemmän virtaa kuluttavan Java-sovelluksen vasta puskurin tai aikarajan täyttyessä.

Watch OS ja tulevaisuudennäkymät

Android Wearin suurimpia puutteita on WebView-näkymä, jota 4.4-versiosta ei vielä löydy. Tämä tarkoittaa, että käyttöliittymien suunnittelussa

ei voi käyttää HTML:ää, eikä kellon ruudulla voi esittää yksinkertaisiakaan nettisivuja.

Varsinainen netin selaaminen älykellon pieneltä ruudulta olisi lähinnä kurioositeetti. Sovelluskehityksen kannalta olisi kuitenkin hyödyllistä kyetä lataamaan palvelimilta pieniä HTML-näkymiä, joiden ulkoasua voisi vaihdella joustavasti. Nykyisellään koko sovellus on päivitettävä uuteen versioon, jos näkymiä halutaan muokata.

On kiintoisaa nähdä, miten Applen tuleva Watch OS -käyttöjärjestelmä suhtautuu HTML:ään ja nettiselailuun. Ainakaan esittelyvideoissa ei näy Safaria, joten WebView-komponenttejakaan ei kannattane odottaa saataville vielä ensimmäiseen versioon.

Videoiden perusteella Applen Watch OS näyttää eroavan Android Wearista muilla tavoin. Oleellisin ero on siinä, miten sovellukset ovat käyttäjän ulottuvilla Iphonen päävalikon kaltaisessa näkymässä. Android Wearissa ne on piilotettu syvälle Start-valikon uumeeniin.

Jos Apple onnistuu tekemään kelloille saman kuin iPhone teki puhelimille, tulee kellon ja sen sovellusten räpläimisestä hauskaa ja viihdyttävää. Silloin Android Wearin notifi kaatioihin ja valkoiisiin kortteihin perustuva käyttöliittymä voi äkkiä alkaa tuntua jäykältä ja vanhanaikaiselta.

Perusteet tuskin silti muuttuvat kovin järkevästi. Kello on edelleen muutaman sekunnin asioihin keskittyvä väline. Kännykkä vedetään esiin muutaman minuutin juttuja varten, ja tunteja vievät hommat taas tehdään mieluiten tietokoneen ääressä. 📱





Tekoäly porttaamaan MSX-pelit

Sanotaan, että ihmeitä tehdään odottaessa, mutta mahdottomuudet vievät enemmän aikaa. Tämä tarina kertoo, kuinka löysin tekoälyn kautta tien yli 30 vuotta vanhan yhteensopivuusongelman ratkomiseen: Nyyrikki's MSX ROM loader for SVI.

Teksti: Timo Soilamaa

Kuvat: Manu Järvinen, Risto Mäki-Petäys, Aki Sivula

Kahdeksankymmentä-luvun alussa, ennen MSX-standardin luk-koonlyömistä, Spectra-video-niminen yritys valmisti SV-318- ja SV-328-nimisiä Z80-pohjaisia tietokoneita. Näitä ko-neita käytettiin lähtökohtana MSX:n suunnittelussa.

Spectravideo ei pitänyt kynttiläänsä vakana alla vaan mainosti näyttävästi kohta koko maailman rakentavan heidän koneeseensa perustuvia tieto-koneita. Joissain maissa koneita jopa markkinoitiin ensimmäisinä MSX-yhteensopivina tietokoneina. Aikaiset omaksujat saivat kuitenkin pettyä, kun kävi selväksi, että sen enempiä koneen fyysiset liitännät kuin ohjelmatkaan eivät olleet laisinkaan yhteensopivia MSX:n kanssa.

SVI-koneet käyttivät erillisiä I/O-portteja lukemiseen ja kirjoittamiseen, toisin kuin MSX, eivätkä osoitteet-kaan olleet miltään osin samat. Myös BIOS, muistinhallinta ja kasettien tal-lennusformaatti olivat täysin erilaiset. Lähimmäksi osui sisäänrakennettu BASIC-tulkki, jossa siinäkin oli ero-avaisuutensa.

Parempi myöhään kuin ei milloinkaan

MSX-pelikehittäjät olivat 80-luvulla monesti varsin heppoisilla tiedoil-la liikenteessä. MSX-standardi kyllä määritteli, miten rautaa tulisi käyttää BIOSin kautta, mutta monesti doku-menteja ei ollut saatavilla. Ongelmia aiheuttivat niin etäisyydet kuin kieli- ja kulttuurimuuritkin. MSX-koneita val-mistivat monet kodinkonejätit kuten Philips, Sony, Panasonic, Yamaha ja Goldstar (sittemmin Lucky-Goldstar eli nykyinen LG). Näillä oli kyllä taitoa antaa tukea tavallisten kodinkonei-den käyttöön, muttei sellaista tukea, jota tietokoneiden ohjelmistokehitys olisi tarvinnut. Niinpä tyypillisesti

pelikehittäjät päätyivät ohjelmoimaan BIOSin ohi rautaa suoraan hyväksi-käyttäen.

Tarina Spectravideon markkinoin-nin huijaamista ostajista ja puutteel-lisen dokumentaation lannistamista pelien porttaajista pisti minut mietti-mään, voisiko näitä historian margi-naaliin jääneitä käyttäjiä auttaa.

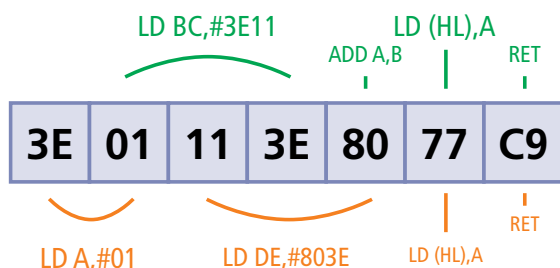
Aloitin kokeilun porttaamalla muu-taman pelin käsin. Se onnistui, mutta pelien porttaaminen on hidasta, tark-kaa ja tylsää rutiininomaista työtä. Il-mankos näitä pelejä on portattu niin vähän. Hommaa pitäisi jotenkin hel-pottaa tai mieluummin automatisoida se.

MSX-BIOSin porttaaminen SVI:lle vähentäisi koodissa tarvittavia muu-toksia huomattavasti. Päädyin porttaa-maan avoimen lähdekoodin C-BIOSin, joka toteuttaa kaikki pelien käyttämät viralliset BIOS-rutiinit, mutta ilman MSX-BASICia. Tällainen korkean ta-son matkiminen oli kuitenkin vain pieni osa ongelman ratkaisua.

Etsi ja korvaa?

Muutaman kerran olen nähnyt yrityk-siä korjata ohjelmaa automaattisesti käyttäen etsi ja korvaa -menetelmää. Jo pikaisella päättelyllä voi kuitenkin todeta, että tämä ratkaisu ei toimi sen generoiman suuren virhemäärän ta-kia. Jos tavallinen I/O-operaatio ottaa muistista kaksi tavua, niin täysin sa-tunnaisessa datassa tällainen tavuyh-distelmä löytyy keskimäärin kerran 64 kilotavussa. Jos pelin koko on esimer-kiksi 16 kilotavua, niin tämä kuulostaa vielä melkein siedettävältä virhemää-rältä.

Yhden tällaisen tavuparin korjaami-sella ei silti päästä vielä kovinkaan pit-källe, vaan korjattavia tavupareja tulee väkisinkin kymmeniä. Virhemääräkin kasvaa todennäköisesti useampaan virheeseen peliä kohden, ja yksikin saattaa riittää kaatamaan koko koneen.



Muistissa olevan tavujonon tulkinta konekieleksi riippuu siitä, mistä kohdasta aloite-taan: tässä vaihtoehtoina 3E ja 01.

Ihminen pystyy helposti arvioimaan tavujonosta, onko kyseessä todennä-köisemmin ajettavaa koodia vai jotain muuta dataa, mutta tietokoneelle täl-lainen päättely on vaikeampaa.

Konekieliohjelmassa eri käskyt ovat eripituisia, joten ohjelmaa tarkastelles-sa konekielimonitorilla ohjelmakoodi ajautuu yleensä viimeistään muutaman käskyn jälkeen ”oikeaan rytmiin”. Jotta saisin tietää, onko kyseessä käskytavu ensinkään, opetin automaattiselle port-tausohjelmalleni, kuinka pitkiä käskyt ovat. Tein rutiinin, joka lukee kohde-tavusta 30 tavua taaksepäin ja askeltaa sen jälkeen käskyjen pituuksilla takai-sin meitä kiinnostavaan kohdetavuun. Mikäli askellus hyppäsi vahingossa kohdetavun yli, oli todennäköisempää, että kyseessä ei ollut komennon ensim-mäinen tavu. Yksinäisenä tällainen hu-tera päätelmä ei ole kovin hyödyllinen, mutta jos päätelmiä pystyisi tekemään tarpeeksi paljon sekä pisteyttämään to-dennäköiset ja epätodennäköiset asiat, niin tietokone pystyisi muodostamaan paremman kokonaiskäsityksen.

Z80-käskykannassa on peruskäskyt, jotka koostuvat käskytavusta sekä nii-tä seuraavista mahdollisista paramet-reista. Tämän lisäksi on laajennetut käskyt, jotka toimivat muuten samoin, mutta sisältävät yhden tai useamman etutavun. Vaikka laajennettuja käskyjä on reippaasti enemmän kuin peruskäs-kyjä, niiden esiintymistodennäköisyys satunnaisessa datassa on pienempi. Satunnaisessa tavujonossa todennä-köisyys laajennettujen käskyjen esiin-tymiselle on vain noin 1,5 prosenttia. Sen sijaan erilaisia rekisterien lataus-komentoja vastaavassa datassa on pe-räti yli 30 prosenttia. Näistä todennä-köisyyksistä sain lähtökohdat koodin jatkokehitykselle.

Mukautuvampi algoritmi

Itseään muuttavan koodin avulla sain rakennettua rutiinin, joka seurasi re-kistereiden käyttöä. Laitoin prosesso-rin seuraamaan kaikkia peruskäskyjä ja kirjaamaan ylös, mitä rekistereiden sisältöjä se käyttää tuottaakseen uutta tietoa ja mihin uusi tieto tallennetaan. Tällä tavoin pystyin havaitsemaan myös, jos rekisteripari tai sen osana oleva rekisteri kirjoitettiin yli ilman, että sen edellistä arvoa oli käytetty tai tallennettu. Tämän lisäksi tein myös taulukon turhista ja toimimattomista

käskyistä, joita normaalisti ohjelmissa tuskin esiintyisi. Testasin ohjelmaa antamalla sille parametriksi kaikki omat osionsa. Hämmästykseni heti ensimmäisellä kerralla ohjelma pystyi erottelemaan rakenteestaan koodi- ja dataosiot virheett. Sitten lisäsin muun muassa kutsujen seuraamisen. Todennäköisyydet paranivat, jos kutsun kohdeosoitteesta katsoen edellisen käsky oli hyppy tai paluukäsky.

Pelien porttaamisessa näiden perustekniikoiden lisäksi ROM-pelien ympäristön kuvaaminen osoittautui erittäin tehokkaaksi. Miinuspisteitä jaettiin muun muassa silloin, jos ohjelma yritti kirjoittaa oman koodinsa päälle tai BIOSiin. Myös BIOSin epästandardeilta alueilta lukemisesta rangaistiin. Kaikki BIOS-kutsut dokumentoimattomille alueille ja osoitteisiin, jotka eivät sisältäneet hyppykäskyä, saivat myös varoitusmittarin värähtämään.

Koska mielestäni vakuuttavat nimet ovat hienoja, päätin nimetä rutiinini vaatimattomasti ”self aware, self modifying artificial intelligence”.

Tulokset käytäntöön

Olin kehittänyt rutiinin, joka osasi kertoa muistiosoitteesta onko se konekielellä käsky vai ei. Nyt piti vielä ratkaista, kuinka I/O-rutiinit muutettaisiin toimimaan automaattisesti uudella alustalla.

Z80:ssä on kahden suoran I/O-komennon lisäksi 24 epäsuoraa komentoa, jotka käyttävät prosessorin C-rekisteriä esittämään I/O-portin numeroa. Kaikki I/O-komennot ovat kaksi tavua pitkiä, joten niitä ei voi pituuseron takia suoraan korvata kutsuilla aliohjelmiin.

Z80:ssä on kuitenkin myös kahdeksan yksitavuaista RST-komentoa, jotka kutsuvat kiinteää osoitetta BIOSissa. Esimerkiksi RST #0 hyppää osoitteeseen 0, mikä aiheuttaa MSX:n uudelleenkäynnistymisen. Koska tämä ei ole peleille käyttökelpoinen ominaisuus, päätin ottaa komennon uuteen käyttöön.

BASICin poistuminen mahdollisti myös kahden muun RST-komennon vapauttamisen. Yhtä käytin suoraan IN-komentoon, toista suoraan OUT-komentoon, ja kolmannella hoidin kaikki epäsuorat komennot. Kun korvasin I/O-komennosta ensimmäisen

tavun RST-komennolla, tiesin että pinossa oleva paluusoite osoittaisi suoraan komennon toiseen tavuun. Pystyin siis pinoa muokkaamalla selvittämään, mikä oli alkuperäinen komento, korvaamaan paluusoitteen sekä emuloimaan toiminnallisuuden ohjelmaa hajottamatta.

Suorissa I/O-komennoissa toinen tavu on I/O-portin numero, joten sen voi videopiiriin tapauksessa suoraan vaihtaa. Tämä on mahdollista koska MSX:n ja SVI:n videopiiri on sama, eikä se hoida koneessa muita tehtäviä. Muiden laitteiden kanssa tämä ei ole aivan yhtä yksinkertaista, mutta onneksi datamäärätkin ovat tyypillisesti kymmeniä kertoja pienempiä. Epäsuorissa I/O-komennoissa portin numero ladataan yleensä rekisteriin lähellä jotain olemassa olevaa I/O-komentoa. Suuri osa ohjelman tempuista tehdäänkin siksi vasta, kun peli on jo käynnistetty käyttäjälle pelattavaksi.

Kun ohjelma käynnistymisensä jälkeen päätyy ensimmäisen kerran RST:n kautta aliohjelmaan, saadaan tietää mikä arvo C-rekisterissä on.

Yksinkertaistaen toiminta perustuu siihen, että ensimmäiseksi tarkistetaan, voiko komennon korvata suoran osoituksen komennolla. Jos se ei käy, etsitään tätä ohjelmakohtaa edeltävistä tavuista rekisterinlataus ja -muokkaus-

komentoja. Sijoituskomento korvataan mahdollisuuksien mukaan oikealla. Jos seuraavalla kerralla suoritettaessa tulos oli toivottu, koodi korvaa RST-komennon uudelleen I/O-komennolla ja suoritus nopeutuu.

Toimii useammin kuin ei

Automaattinen koneelta toiselle porttaus ei voi koskaan toimia täydellisesti. Jos portattavan ohjelman koodi on pakattu, siihen ei oikein millään pääse väliin. Pakkaus ei onneksi ole kovin suuri ongelma, koska ROM-pelit tyypillisesti käyttävät RAM-muistia varsin säästeliäästi maksimoidakseen yhteensopivuuden varhaisimpien koneiden kanssa. Koodista saattaa myös löytyä esimerkiksi keskeytyksiin liittyviä bugeja, jotka eivät ole alkuperäisessä ohjelmassa pompanneet esiin, mutta ajastusten aavistuksen muututtua ne kaatavatkin koko koneen.

Satunnaistestien perusteella kuitenkin yli puolet ohjelmista toimii esittelemani käsittelyn jälkeen. Se ei ehkä kuulosta paljolta, mutta kuitenkin rikastuttaa SVI:n pelitarjonnan kertaheitolla aivan uudelle tasolle. Mielestäni saavutus ei ole ollenkaan hassumpi, kun ottaa huomioon, että tällä kertaa porttauksen onkin tehnyt tietokone itse ja vieläpä ilman ihmisen avustusta. 🐱

Tällaistaikin yritettiin

Todennäköisesti lepytelläkseen huijattuja ostajia Spectravideo julkaisi epäonnistuneen SV-606 MSX-peliadapterin. Siinä oli MSX-yhteensopiva kasettiasemaportti, moduuliportti, joystick-portit sekä aakkosjärjestykseen järjestetty näppäimistö. Koska ohjelmat pystyivät käyttämään maksimissaan vain 16 kilotavua muistia, varsinkaan kasetilta ladattavat pelit eivät toimineet yleensä laisinkaan.



Linkki käsiteltyyn ohjelmaan:

<http://www.msx.org/downloads/utilities/rom/nyyrikkis-msx-rom-loader-svi>



Tietokonelehden ekosysteemi

Tietokonelehteä voi tehdä kahdella tavalla. Tietokonelehti on siis bitti – ykkönen tai nolla.

Janne Sirén

Tietokonelehteä voi kuulemma tehdä kahdella tavalla, joko ”vetoamalla laajaan ipadon-parastaikinä ja tekniikantarkoitushelpottaaelämää -ryhmään tai matriisitulostimella tehtyjä ohjelmistauksia palvovaan, sisäänpäin lämpiävään kansanosaan” (ks. Skrolli 2013.4).

Huuli sikseen, on onneksi kolmaskin tapa.

Nuorsuomalaisuutta

Kahdeksankymmentäluvulla pääkaupunkiseudulla oli kaksi suurta päivälehteä, Helsingin Sanomat ja Uusi Suomi. Molemmat kilvoittelivat kasvavan sukupolven sieluista. Mieleepi jäi iltapäivä Uuden Suomen konttorilla ja Hesarin nuori toimittaja -päivä, jossa Logo-kilppari tussasi lattiaa. Sitten Uusi Suomi hävisi ja Hesari voitti. Helsingin Sanomien asema kahvipöytäyhteisöissä oli liian vahva. Luit Hesaria, jotta tiesit mistä puhutaan.

Skrollataan ajassa eteenpäin 2010-luvulle. Nyt Hesari kutistuu, kun taas Uusi Suomi on herätelty yhteiskunnallisen keskustelun keskiöön verkkoyhteisönä. Uusi Suomi ei ole tökerö yritys hyväksikäyttää blogisisältöä, vaan moderni sosiaalinen ekosysteemi, joka haastaa ja houstaa. Digimurroksessa Hesari on menettänyt yhteisöllisen siteensä koko kansan puheenaiheena ja rimpuilee ajassa virtuaalisella kolikkoautomaatilla.

Kasarilla oli myös matriisitulostimella tulostettujen ohjelmistauksien MikroBitti, jonka sivuilla ja brän-

dämällä nuorisoleireillä kouluttiin tulevaisuuden guruja. Lehti ruokki nuoria ja nuoret lehteä. Moni päätyi alalle koska Bitti. Joistakin tuli IT-miljonäärejä. Ysäriällä MBnet-purkki jatkoi perintöä, ja MBnet-yhteisöstä muodostui merkittävä kotimaisen indie-softatehtailun kokoontumispaikka sekä tilaajien sitouttaja. Valitettavasti ekosysteemin siirtäminen internetiin jäi pitkälti tekemättä – himoittiin hittejä ja Hintaseurantaa, mutta niillä ei rakennettu yhteisöä.

2010-luvun ICT-alan keskiössä on ekosysteemi. Applen Iphone ja Ipad olivat alkujaan ylтиөрöheita tuotteita, joihin ei perinteisten kuluttajatutkimusten valossa uskottu. Toisin kävi. Massiivisen koukuttava menestystarina niistä kehkeytyi viimeistään ympärille kerääntyneen kehittäjäyhteisön myötä. Vaikka Applen laitteiden kuluttajakeskeisyyttä ja kontrollimaniaa voi syystä parjata, niiden nostattama appluomisvimma mullisti Applen lisäksi koko toimialan.

Tietotekniikalla tekeminen kiinnostaa edelleen. Uudella vuosituhanella peräkkämmäristä maailmanmaineeseen nousseilla appien tekijöillä onkin paljon yhteistä 1980-luvun ohjelmointipioneerien kanssa. Mutta vaikka luovien tietokoneharrastajien joukko tuskin on pienempi 2014 kuin se oli 1984, nykyajan appintekijälle ei kuitenkaan ole lehtipisteissä omaa suomalaista tietokonelehteä, ei paperista eikä sähköistä. Ei ainakaan ennen Skrollia.

Ykkönen vai nolla

Entäpä jos niche- ja massatuotteiden sijaan puhuttaisiin tuotteista joista välitetään tai ei välitetä. Jos tuote suunnataan yleisölle, joka ei välitä vaan ostaa pelkkään tarpeeseen, markkinan on parempi olla suuri tai ainakin vakaa. Jos tuote suunnataan yleisölle joka välittää, markkina voi olla lähes olematon – olettaen, että kustannusrakenne mitoitetaan sopivasti ja tuotteesta välittävät asiakkaat vievät sen viraaliksi. Kickstarter on täynnä odottamattomia menestystarinoita.

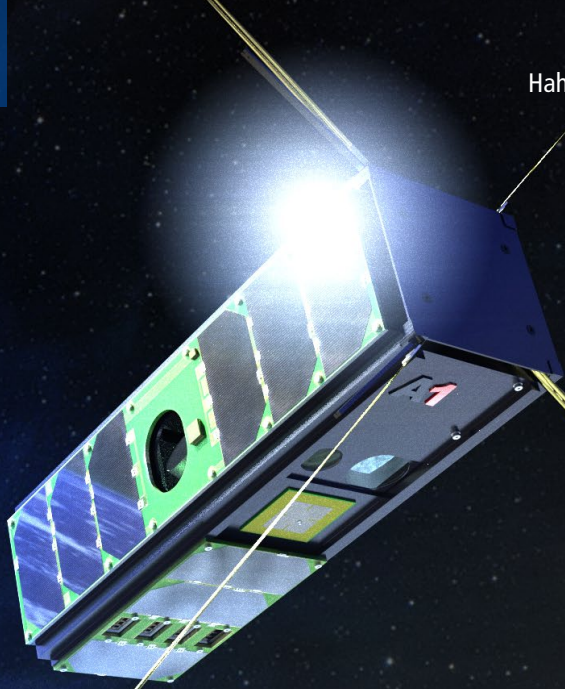
Tuotteen linkaareissa kommoditisaatio johtaa paitsi hintojen romahduttamiseen, myös siihen että tuotteesta välitetään yhä vähemmän. Ihmiset jotka eivät välitä, eivät muodosta hedelmällistä pohjaa intohimolle tai uskollisuudelle tuotteen ympärillä – saati yhteisölle. Jos valmistaa tuotetta josta ei välitetä, olisi tuotteen syytä olla monille tarpeellinen. Pahinta on hirttäytyä tuotteeseen, josta kukaan ei enää välitä ja alati harvempi edes tarvitsee.

Välittävistä ihmisistä koostuvaa yleisöä on paljon vaikeampi tunnistaa etukäteen, saati palvelu kvartaalitalouden keinoin. Menestys vaatii usein visionääriä, mutta hyvään alkuun voi päästä tekemällä jotain, johon tekijät itse suhtautuvat intohimoisesti. Välittävissä yhteisöissä piileekin myös kolmas tapa tehdä tietokonemediää. Skrollilla yhteisön juurta epäilemättä on, mutta se ei ole ainoa 2000-luvun MikroBittimä. Minkä MBnet luovutti, Muropaketti nappasi jo aikaisemmin.

Ekosysteemiä ei löydy lukijatutkimuksista. Se luodaan sparraamalla esiin yhteisö, joka välittää ja jonka kanssa tehdään välittämisen arvoista tuotetta. Yrittäessään voi myös epäonnistua, sillä muutoksen johtaminen on vaikeampaa kuin pelkkien trendien seuraaminen. Houkutus siirtyä visionääristä seuraajaksi voi siksi olla markkinajohtajallekin suuri. Seuraajana on turvallisempaa – ainakin kunnes menettää ne asiakkaat, jotka välittävät, ja kiihtyvässä määrin myös ne, jotka tarvitsevat. Sitten tulee kylmä.

Toivottavasti jotkut uskaliaat mediat vetoavat jatkossakin kansanosaan, joka välittää aktiivisesta suomalaisesta tietokonekulttuurista, kotimaisesta tee-se-itse-atk:sta. Saadaan tämä kansanosa kasvuun. Kuka Skrollin kirjoittajista on seuraava IT-miljonäärimme?





Opiskelijavoimin avaruuteen – Aalto-1 ja nanosatelliittitrendi

Aalto-yliopiston radiotieteen ja -tekniikan laitoksella on pitkäjänteisesti rakennettu Aalto-1-nanosatelliittia opiskelijatyönä jo useamman vuoden ajan. Tutkimme, millainen Suomen ensimmäisestä satelliitista olisi tulossa.

Teksti: Joonas Järvenpää Kuvat: Aalto-1-tiimi

Avaruusajan alussa satelliittien laukaisu oli vain suurten kansallisvaltioiden ulottuvilla, mutta nanosatelliittitekniologian myötä pienikin ryhmä pystyy kokoamaan satelliitin kiertoradalle lähetettäväksi.

Nanosatelliittien perusta on Cubesat-standardi vuodelta 1999, jolloin Kalifornian Polytechnic State University ja Stanfordin yliopisto kehittivät opiskelijoille sopivan standardin, jonka pohjalta voidaan rakentaa edullisia avaruusluotaimia. Oleellisin standardissa on määritellä yhtenäiset ulkomitat, jotta satelliitit voidaan laukaista samanlaisilta alustoilta. Ensimmäiset Cubesatit laukaistiin vuonna 2003, ja vuoden 2012 loppuun mennessä niitä oli laukaistu jo yli sata.

Cubesatin perusyksikkönä toimii 10×10×10 cm:n kuutio, joita voidaan yhdistellä suuriksikin yksiköiksi, joskin useimmat toteutukset ovat kooltaan korkeintaan kolme yksikköä. Koska kaikki Cubesatit ovat poikkeileikkaukseltaan 10×10 cm, voidaan ne laukaista avaruuteen samanlaiselta alustalta. Sama laukaisuväline lähettääkin lähes poikkeuksetta avaruuteen useamman nanosatelliitin kerralla.

Vaikka paljon työtä on tehty Cubesatien standardoinnin ja miniaturisointien eteen, suurin innovaatio lienee silti niiden yhtenäinen P-POD-laukaisualusta, joka vapauttaa nanosatelliitin kuljetuslaitteesta avaruudessa. Tämä mahdollistaa osajärjestelmien kehityksen, integroinnin ja verifiointin laukaisujärjestelmästä erillään, jolloin päästään merkittäviin kustannussäästöihin myös kehitysajan osalta.

Hiljattain on auennut mahdollisuus lähettää Cubesateja avaruuteen myös kansainväliseltä avaruusasemalta ISS:ltä. Uusi Japanissa kehitetty laukaisualusta J-SSOD on liitetty osaksi avaruusasemaa. Kantorakettilaukaisuun verrattuna tässä on erityisesti se etu, että avaruusasemalla satelliittien toiminta voidaan tarkistaa ennen laukaisua.

Nano-etuliite ei ole sattumalta valittu, sillä satelliitteja luokitellaan yleensä massan mukaan: pikosatelliitti on alle 1 kg, nanosatelliitti 1–10 kg ja mikrosatelliitti 10–100 kg massaltaan. Cubesat-standardi rajaa yhden yksikön massaksi korkeintaan 1,33 kg.

Urakka alkaa Otaniemessä

Aalto-1 sai alkunsa vuoden 2009 lopussa, kun useat yliopistot olivat laukaissseet näyttäviä nanosatelliittiluotaimia ja Aalto-yliopiston Jaan Praks

ja Antti Kestilä päättivät aloittaa myös suomalaisen Cubesat-projektin. Satelliitteja suunniteltaessa ensimmäinen etappi on toteutettavuustutkimus, johon tässä tapauksessa rekrytoitiin laitoksen erikoistytökurssin opiskelijat kartoittamaan satelliitin mahdollisuuksia. Kestilän mukaan ”homma lähti nopeasti pyörimään eteenpäin”, kun kiinnostusta ja tukea löytyi. Vuosien varrella projektiin onkin osallistunut lähes 70 opiskelijaa.

Aiemmin nanosatelliitteja oli jo rakennettu opiskelijaprojektina muun muassa Virossa, jossa Tarton yliopiston johdolla kehitetty ESTCube-1 laukaistiin kiertoradalle vuoden 2013 toukokuussa. Kooltaan se oli vain kolmannes Aalto-1:stä eli yhden yksikön kokoinen. Yhteistä molemmille satelliiteille on se, että niiden tarkoitus on kokeilla käytännössä Ilmatieteen laitoksella kehitetyn sähköisen aurinkotuulipurjeen ideaa.

Rajoitetun kokonsa vuoksi Cubesateissa on usein hyötykuormana vain yksi tai kaksi komponenttia välttämättömien perusjärjestelmien lisäksi, mutta Aalto-1:een on onneksi saatu mahtumaan kolme hyötykuormaa: kuvaava spektrometri (AaSI) VTT:ltä, säteilymittari (RADMON) Helsingin ja Turun yliopistosta ja elektrostaattinen

plasmajarru (EPB) Ilmatieteen laitok-
selta.

Aurinkopurjeen ideaan perustu-
vaan plasmajarruun kuuluu noin sadan metrin pituinen lankakela. Kelan avauduttua avaruudessa siihen johdetaan korkea jännite, jonka seurauksena langan sähkökenttä reagoi ympäröivän aurinkotuuliplasman kanssa. Tarkoituksena on satelliitin käyttiään loputtua pudottaa ratakorkeutta vähitellen ja lopulta vähentää Maata kiertävän avaruusromun määrää.

Pienikokoinen kamerajärjestelmä

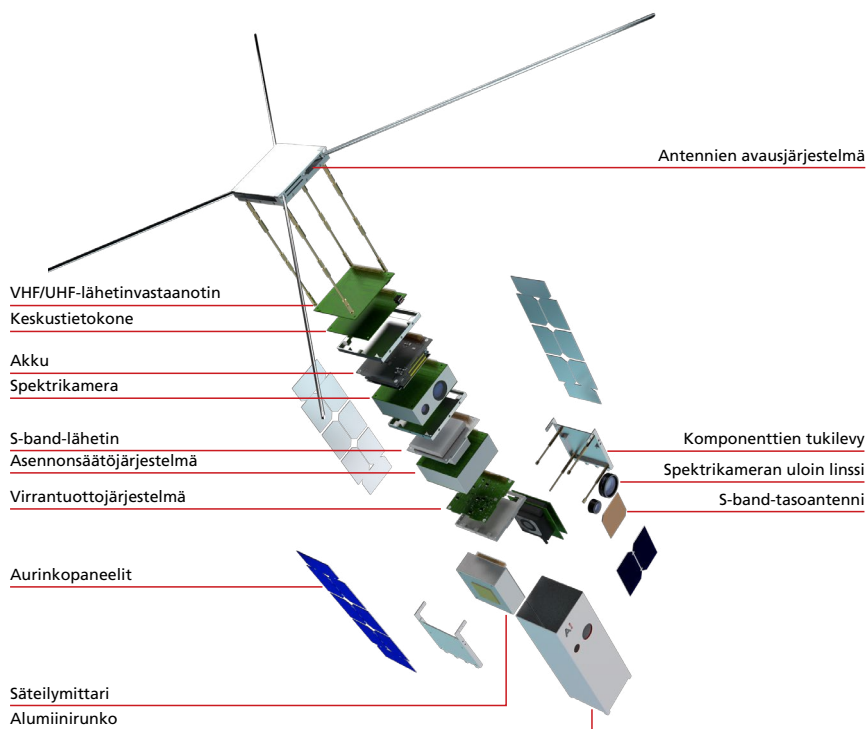
Aalto-1:n tärkein hyötykuorma on kuvantava spektrometri, joka vain parinsadan gramman painollaan on selvästi pienempi kuin satelliiteissa usein käytettävät kymmeniä kiloja painavat kamerat.

Ensisijaisesti tarkoitus on kokeilla pienten kameroiden toimivuutta, mutta niiden ottamia kuvia voidaan myös käyttää esimerkiksi vesistöjen tilan tai metsien kasvillisuuden kartoittamiseen. Spektrometri on tekniikaltaan Fabry-Pérot-interferometri, joka kykenee ottamaan 2D-kuvia samanaikaisesti korkeintaan kolmella eri aallonpituusalueella.

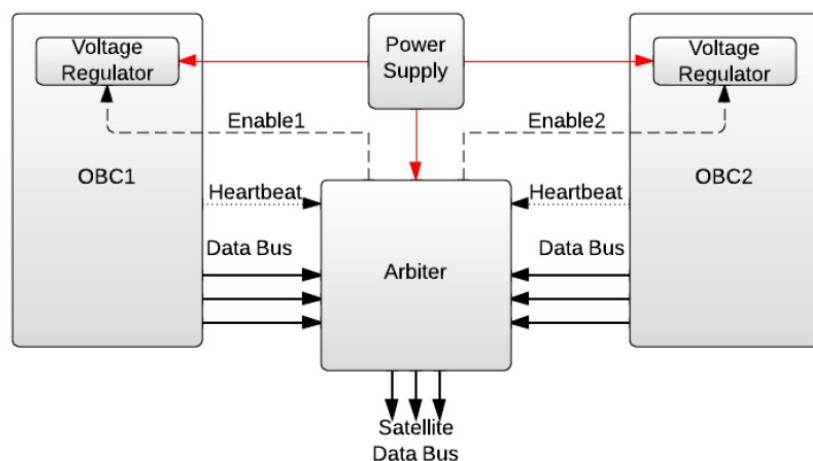
Oheisjärjestelmiksi Aalto-1:ssä luetaan keskustietokoneet (OBC), virrantuottojärjestelmä (EPS), aurinkopaneelit, akusto, S-band- ja UHF/VHF-radiot, antennien avausjärjestelmä (ADS), asennonsäätöjärjestelmä (ADCS) sekä GPS-paikannin.

Satelliitti on tarkoitus laukaista 600–850 kilometrin korkeuteen aurinkosynkroniselle polaarikiertoradalle, mikä tarkoittaa sitä, että ratataso pysyy paikallaan aurinkoon nähden. Näin satelliitti ylittää samat maapallon alueet samaan aikaan joka päivä, mikä helpottaa maa-aseman toimintaa. Kiertorata luokitellaan Low Earth Orbit-tasolle, joka on miltei kaikissa havaintosatelliiteissa käytetty taso.

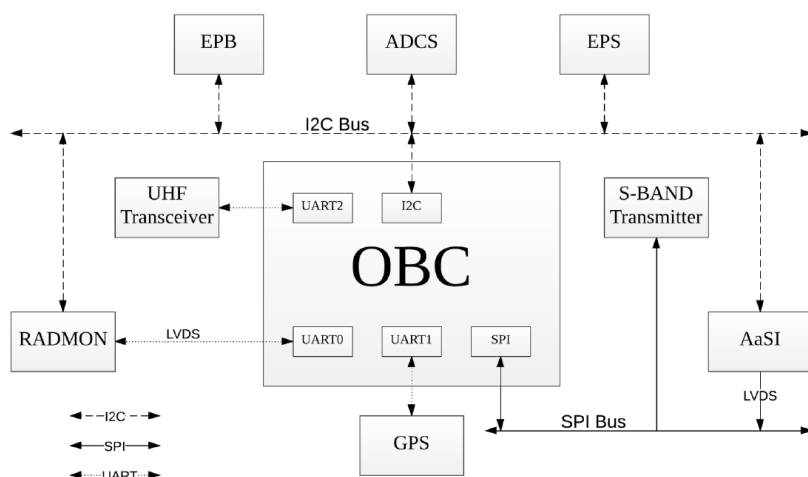
Aalto-1-satelliitissa on myös mukana sen kokoluokkaan nähden harvinainen ominaisuus, kolmiakselistabilointi, jonka tarkoituksena on erityisesti vakauttaa satelliitti spektrometrin käyttöä varten. Stabilointi tilattiin valmiina komponenttina saksalaiselta toimittajalta. Monissa tapauksissa voikin suunnittelutyön määrän vähentämiseksi olla järkevintä hankkia alijärjestelmiä valmiina komponentteina.



Hahmotelma alijärjestelmistä.



Mikrokontrolleri (arbiter) valitsee käytettävän keskustietokoneen kahdesta vaihtoehdosta.



Aalto-1:n tiedonsiirtoväylien lohkokaaio. Sama laite voi käyttää useampaa väylää. Kullakin tiedonsiirtoprotokollalla on oma viivatyyppinsä.

Toimintavarmuus suunnittelukriteerinä

Aalto-1:n keskustietokoneen suunnitteluperiaatteena on kahdentaa toiminnalle kriittinen keskusyksikkö kahteen itsenäiseen osaan, joista erillinen valvontamikrokontrolleri valitsee kulloinkin käytettävän version. Toiminnan varmistamiseksi satelliitin keskustietokone käynnistetään uudeleen neljän minuutin välein, mikäli mikrokontrolleri ei pysty muodostamaan siihen yhteyttä. Jokin toinen laitekin voi jumiuttaa satelliitin kommunikaatioväylän, jolloin tarvitaan uudelleenkäynnistys.

Sisäisesti satelliitissa käytetään komponenttien väliseen tiedonsiirtoon pääasiassa sarjapohjaisia I²C- ja SPI-väyliä. Molemmat väylät ovat yleisiä sulautetuissa järjestelmissä niiden yksinkertaisuuden vuoksi. Satelliitissa suuren tiedonsiirtotarpeen komponentit, kuten spektrikamera ja S-band-lähetin, on sijoitettu SPI-väylään.

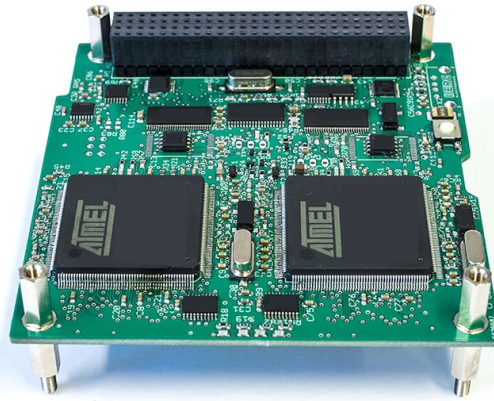
Aalto-1:n pienen koon vuoksi aurinkopaneeleille on vain vähän tilaa. Aurinkopaneelien maksimiteho on noin 10 W, joten kaikkien laitteiden on toimittava melko vähäisellä virrankulutuksella. Virrantoottojärjestelmä lisäksi lataa ylimääräistä energiaa yhteen 30 Wh akkuun myöhempää käyttöä varten.

Keskusyksiköksi Aalto-1:een valittiin ARM-sarjan AT91RM9200-suoritin, joka toimii satelliitissa 18 MHz:n taajuudella suorittaen Linux-ydintä. Siruun on integroitu vain 16 kB muistia, joten käyttöjärjestelmää varten tarvitaan ulkoista lisämuistia. Keskustietokoneessa on tämän vuoksi lisäksi yksi 32 MB:n SDRAM-siru 16 bitin väylässä.

Valvontamikrokontrollerina käytettävää MSP430-sarjan sirua on jo aiemmin käytetty Cubesateissa, ja niiden sisältämä FRAM-muisti on suojattu avaruuden säteilyn aiheuttamilta virheiltilä.

Säteilytesteistä hyödyllistä tietoa

Nanosatelliittiprojekteilte hieman poikkeuksellisesti Aalto-1:n alijärjestelmille on myös tehty säteilytestejä Jyväskylän yliopiston kiihdytinlaboratoriossa. Laboratorio lienee eräs Euroopan parhaista paikoista elektronikan säteilykeston testaamiseen, koska



Aalto-1:n molemmat keskustietokoneet. Tällaisia piirilevyjä voidaan nanosatelliittiin koota luontevasti useita päällekkäin.



FITSAT-1, F-1 ja TechEdSat ohittavat ISS:n aurinkopaneelit laukaisun jälkeen lokakuussa 2012. Kuva: Nasa.

sen hiukkaskiihdyttimellä tuotetut olosuhteet ovat hyvin samankaltaisia kuin avaruudessa esiintyvä säteily.

Avaruussäteily aiheuttaa tyypillisesti virheitä muistipiireissä, yleistä tehonkulutuksen nousua ja lopulta komponenttien rikkoutumisen. Cubesatien elinikä on kuitenkin lyhyt, joten ne rakennetaan usein perustason (Commercial Off The Shelf) komponenteista: pitkän ajan säteilyrasitus ei ehdi nousta merkittäväksi ongelmaksi niiden eliniän aikana.

Vaikka avaruussäteily voikin aiheuttaa ongelmia, tilastojen valossa selvästi yleisin Cubesat-lentojen epäonnistumisia aiheuttava tekijä on kommunikaation estyminen laukaisun jälkeen. Näitä ongelmia pyritään ehkäisemään

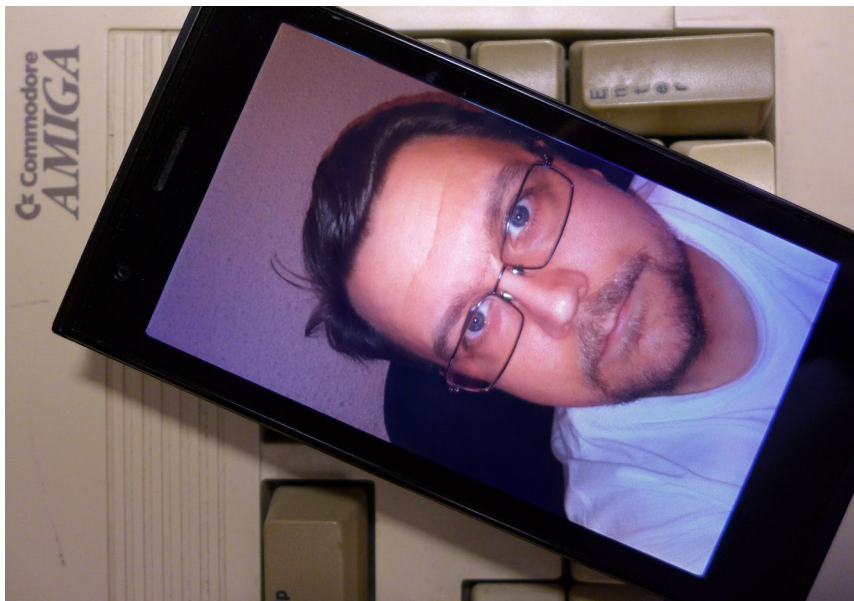
järjestelmätason integraatiotestauksella, koska alijärjestelmät eivät läheskään aina toimi aivan odotetulla tavalla yhdessä.

Satelliitin kanssa kommunikointiin tarvitaan tietysti myös oma maa-asema. Aalto-1:n tapauksessa sellaisen rakentaminen ei tule erityisen kalliiksi: sijoitetaan vain antennit UHF/VHF- ja S-band-viestintää varten sopivan rakennuksen katolle Otaniemessä. Lisäksi tarvitaan antennien kääntömootorit, radiopäätte ja niitä käyttävä tietokone. Suurin osa monimutkaisuudesta aiheutuu kuitenkin tätä kokonaisuutta ohjaavan ohjelmiston kehittämisestä.

Aalto-1:n lopullinen laukaisupäivä ei ole vielä varmistunut, mutta se ajoittuu vuoden 2015 puolelle. 🚀

Lisää luettavaa

- Aalto-1 -projekti: <https://wiki.aalto.fi/display/SuomiSAT/Etusivu>
- Earth Observation Portalin laaja selostus aiheesta: <https://directory.eoportal.org/web/eoportal/satellite-missions/a/aalto-1>
- Swartwout, Michael: The First One Hundred CubeSats: A Statistical Look. <http://www.jossonline.com/Papers.aspx?id=fum2xf18>



Minä ja Purjekala

Vuosi sitten tein päätöksen sijoittaa suomalaiseen. Kannattiko?

Mikko Heinonen

Suhtaudun tekniikkaan jossain määrin tunteellisesti. Etenkin kännykän kaltainen, mukana kannettava laite on alusta asti herättänyt vahvoja fiiliksiä, eikä sen valinta ole koskaan ollut yksinkertainen tapahtuma. Olen raivostunut antenniaan katkovaan Panasonicin, ihastunut Benefonin tekstinsyöttöön ja hekumoinut N900:n kanssa nörttinirvanassa. Nokia N9:n hankin julkaisussa, vaikka tiesin laitteen olevan tekninen umpikuja.

Ensikosketus

Kun Jolla-puhelin julkistettiin, tiesin sen olevan taas kerran laite minulle. Sen lisäksi että Linux-puhelimen ajatus kiinnosti, halusin myös tukea sympaattista pikkufirmaa. Koska entinenkin puhelimeni oli marginaaliveikotin, en jäänyt suremaan ostettuja sovelluksia. N9 oli myös opettanut siihen, että kaikki ei todennäköisesti toimi ainaakaan alusta alkaen.

Eikä kyllä toiminutkaan. Kun sain puhelimen käteeni viime jouluksi, kävi hyvin nopeasti ilmi, ettei se ole sellaisenaan käyttökelpoinen. Vaikka tiesin ostaneeni beetaversion, muun muassa folion teippaaminen NFC:n päälle oli

omalle viitseliäisyydelleni vielä vähän liikaa. Päätin jatkaa N9:n käyttöä, kunnes ohjelmisto kypsyy vähän.

Kauhun tasapaino

Loppukeväästä alkoi näyttää hyvältä. Puhelimeen lisättiin MMS-viestit, joita käytän edelleen sukulaisteni kanssa. Muutenkin suurin osa ominaisuuksista alkoi olla käyttökelpoisia, samalla kun N9:n käytettävyyks alkoi hiljalleen lahota ohjelmistojen vanhentuessa. Tietyssä pisteessä laitteet olivat suunnilleen yhtä vaivalloisia käyttää, mutta Jollalla oli sentään positiivinen odotusarvo, joten siirsin SIM-korttini siihen.

N9:stä olikin yllättävän helppo astua Jollan kyytiin. Pidän Jollan käyttöliittymästä vielä nokiaistakin enemmän. Visuaalinen tyyli on selkeää ja valitut kirjasimet ovat hyviä. Akkukin kestää päivituksen jälkeen varsin hyvin, ja edelleen ihmettelen, miksi kaikkia puhelimia ei voi herättää lepotilasta tuplanapauttamalla ruutua.

Pieniä ja isompia murheita

Jos tulisin jostain muualta kuin Maemo- ja Meego-maailmasta, saataisin ihmetellä, miksi puhelin on käytännössä pakko käynnistää kerran viikossa uudelleen. Tai miksi minun

piti käydä käsin korjaamassa tiedostojärjestelmää komentoriviltä. Itse kun sain tottua näihin jo aiemmin.

Hakkeripuhelimia käyttäneenäkin minua kuitenkin ärsyttää, että sähköpostiohjelmassa ei ole hakutoimintoa ja että lähettäjän nimenä näkyy postilaatikkoni kirjautumistunnus. Googlekalenterista synkronoidut tapahtumat eivät hälytä. Ja joskus puhelimeen ei voi vastata, kun hieno liukusysteemi ei toimi. Ja kameraohjelmisto saa koko puhelimen UI:n tahmaamaan sietämättömästi. Mitä näitä nyt on.

Vakiovastaus puutteisiin on luonnollisesti, että laitetta päivitetään usein. Tämä piti paikkansa kesälomiin asti, mutta niiden jälkeen päivitystahti on hidastunut. Siirtyminen uuteen Qt-versioon on tuottanut vaikeuksia siinä määrin, että ”September update” on vielä testauskäytössä näin marraskuun lopulla.

Rakkaus ei koskaan häviä

Vaikka käytän puhelintani pääosin työasioiden hoitamiseen, en ostanut Jollaa itselleni työpuhelimeksi. Jätin itselleni sen takaportin, että voin vielä käyttää työluurin päivitykseen varatun budjetin hankkiakseni jonkin toisen laitteen, jos purjeveneilystä tulee sietämätöntä.

Sony Xperia Z3 ehti jo viekoitella luokseen, mutta onneksi filosofiset riskitiriani Androidin kanssa ehkäisivät hankintapäätöksen. Jolla kuitenkin toimii ja sillä voi hoitaa päivittäiset asiat. Jopa tärkeimmät Android-sovellukset toimivat auttavasti.

En voi siis sanoa yllättyneeni, koska tiesin mihin hyppäsin. Samaan hengenvetoon voin täysin ymmärtää, jos joku pettyi hankittuaan Jollan. Hän todennäköisesti odotti saavansa valmiin tuotteen. Toisin kuin Nokia ja monet muut, Jolla sentään ilmoitti melko avoimesti, että puhelin on pahasti keskeneräinen.

Olin vilpittömän iloinen, kun Jolla julkaisi Slushissa uuden tablettihankkeensa. Olin iloinen myös siitä, että hanke saavutti rahoitustavoitteensa niin nopeasti. Säästyin nimittäin itse ostamasta laitetta ihan yleisen hyväntekeväisyyden nimissä. Toivon yhtiölle ja sen työntekijöille kaikkea hyvää, mutta odotan tämän yhden laitteen valmiiksi ennen kuin ostan seuraavan. 🐳



Tekoäly – muutakin kuin robotteja

Tekoälyä on nykyään kaikkialla. Useimmille se tuo mieleen vain tieteiselokuvien robotit tai japanilaisten huippuyliopistojen pioneeriprojektit. Tekoäly on kuitenkin monipuolinen tieteenala, jonka sovellutukset ulottuvat useammille elämänalueille kuin moni osaa kuvitella. Vain harvalla tekoälyn osa-alueella on mitään tekemistä robottien kanssa.

Teksti: Antti Ylikoski

Kuvat: Mikko O. Torvinen, Mitol Meerna

Ilmaisu ”tekoäly”, Artificial Intelligence, AI, on tekoälyn grand old manin John McCarthyn luoma ja peräisin 1950-luvun puolivälistä. Tieteenalana tekoälyn ta-
voite on kahtalainen:

- mallittaa ja rakentaa tietokoneella toimintaa, joka ihmiseltä edellyttäisi älykkyyttä
- tutkia yleistä, rajoittamatonta älykkyyttä: älyä ihmisessä, älyä koneessa, älyä matemaattisissa malleissa, älyä eläimissä, älyä aivan missä tahansa.

1990-luvulla comp.ai-uutisryhmässä taitettiin peistä tekoälyn määritelmästä. Lähinnä osoittautui, että tekoälyn tutkimuskenttä on niin korkeatasoinen ja laajamittainen, että lyhyttä ”matemaattista” määritelmää tekoälylle on mahdotonta antaa. Edellä oleva kirjallisuudesta kotoisin oleva luonnehdinta on siten paras, mitä voidaan esittää.

Mitä kaikkea tämän määritelmän puitteisiin sitten mahtuu? Koneellista puheentunnistusta, automatisoitua ongelmatilanteiden ratkaisua, eläinten aivojen mallintamista...

Ennen kuin pääsemme nykypäivän esimerkkeihin, piipahdamme tekoälyn historian parissa.

Alkuvaiheet

Tekoäly tieteenä syntyi 1940- ja 1950-luvuilla, jolloin monet tieteenalan maailmankuulut pioneerit elivät ja vaikuttivat. Kaksikuukautinen kokous Dartmouthissa 1956 oli erityisen vaikutusvaltainen ja kuuluisa. Tuolloin ensimmäisen kerran käytettiin termiä *artificial intelligence*.

Aivan varhaisessa vaiheessa, 1950–1970-luvuilla, innostus oli suurta ja odotukset huomattavia. Jossakin vaiheessa osoittautui, että tekoäly on valtavan paljon vaativampi tutkimuskohde kuin saattaisi puusta katsoen näyttää. Erityisesti tietokoneella tapahtuvassa koneellisessa kielenkääntämisessä saatiin aikaan legendaarisia virheitä – kuten seuraava:

Kääntämisohjelmalle annettiin tulkittavaksi englanninkielinen idiomiksi ”Out of sight, out of mind”. Lopputulos oli: ”Blind idiot”. Luonnollista kieltä ei nimittäin voi käsitellä, jollei ole jonkinlaista käsitystä kielen merkityksestä, eli semantiikasta.

Kuitenkin tässä vaiheessa saatiin aikaan merkitseviä tieteellisiä tuloksia.

Edistysaskelia

Noin vuosina 1966–1973 tekoälytutkimuksen piirissä jouduttiin palaamaan kylmään reaalityodellisuuteen. Varhaiset tekoälyohjelmat olivat usein näyttäneet menestyksiltä, mutta sovellettaessa niitä vaikeampiin tai uusiin ongelmiin, ne saattoivatkin epäonnistua surkeasti.

1970-luvulla syntyi tietämyspohjaisten järjestelmien idea – ja se näytti olevan tie menestykseen. Ihminenhan yleensä ratkoo ongelmia tietämyksensä perusteella, eikä esimerkiksi pelaa šakkia matemaattisesti šakkitietokoneiden tapaan laskemalla miljoonia siirtoja etukäteen.

Teoriasta todellisuuteen

1980-luvun aikana tekoälystä muotoutui ohjelmistoteollisuuden haara. Tekoäly on siis teknologiaa ja teollisuutta, eikä enää pelkkää akateemista tutkimusta.

Kahdeksankymmentäluvun puolivälin jälkeen tekoälytutkimus on omaksunut kunnan tieteellisen metodin. Siitä lähtien se on ollut tieteellisesti hyvällä pohjalla: osin formaalia ja osin empiiristä. Vuodesta 1986 lähtien käytössä ollut neuroniverkkomalli, tai yleisemmin konnektionistinen paradigma, on osoittautunut käyttökelpoiseksi.

Kun tänä päivänä katsoo esimerkiksi AAAI:n (*Association for the Advancement of Artificial Intelligence*) vuotuisen konferenssin konferenssijulkaisuja eli *Proceedings*seja, voi havaita, että tekoälytutkimus on muuttunut vahvaksi matematiikkaan perustuvaksi tieteenksi. Tähän perustuu laaja teollisessa ja kaupallisessa käytössä tekoälyä hyödyntävä teknologiateollisuus.

Asiantuntijajärjestelmät

Asiantuntijajärjestelmät (*expert systems*) ovat ihmisasiantuntijan asiantuntemusta sisältäviä tietokonejärjestelmiä. Niiden tietämys koostuu suurelta osin JOS \Rightarrow NIIN -muotoisista säännöistä. Esimerkiksi, JOS (potilaalla on korkea kuume ja runsaasti verenvuotoa) \Rightarrow NIIN (kyseessä saattaa olla Ebola-infektio).

Hyvät ihmisasiantuntijat ovat harvinaisia, vaikeasti koulutettavia ja taloudellisesti arvokkaita. Asiantuntijajärjestelmät olivatkin ensimmäisiä tekoälyn teollisia menestyksiä.

Malminetsijä

Jo 1980-luvulla tehtiin rahaa asiantuntijajärjestelmillä. Eräs ensimmäisistä onnistujista oli malmeja geologisen tiedon perusteella etsivä järjestelmä nimeltä PROSPECTOR, jonka todellakin löysi taloudellisesti arvokkaita malmioita.

PROSPECTORin päättely koostui ohjelmasta, joka käsitteli tietokoneen muistissa olevia verkkoja. Se ei siis muodostunut säännöistä.

Tietokoneitten kokoaja

Varhaisimmat asiantuntijajärjestelmät olivat kuitenkin usein yksinkertaisia sääntöpohjaisia systeemejä, kuten amerikkalaisen tietotekniikkavalmistajan Digital Equipment Corporationin XCON. Se kokosi VAX-tietokonejärjestelmien kokoonpanoja asiakkaan toivomusten mukaisesti.

XCON-järjestelmän yritykselle tuoma lisäarvo oli siinä, että 1980-luvulla VAX-tietokoneita sai tilata erilaisina kokoonpanoina, ja XCON sitten älykkäästi kokosi asiakkaalle tietokonepa-

ketteja hänen tilauksensa mukaisesti. Kokoamisessa säästettiin rahaa – ja tietysti kilpailuetua tuli siinäkin, että mainittuja VAXeja sai tilata joustavasti hyvin monenlaisina kokoonpanoina aina asiakkaan tarpeen mukaan.

Nyky aika

Kehittyneissä sääntöpohjaisissa järjestelmissä on mukana esimerkiksi tilastotiedettä tai tietämyksen varmuuden astetta ilmaisevia varmuuskeinoja. Kaikkein kehittyneimmät asiantuntijajärjestelmät ovat logiikkapohjaisia.

Windows-käyttöjärjestelmien sisällä on useampiakin asiantuntijajärjestelmiä, jotka muun muassa saavat Windowsin selviämään erilaisista virheilanteista älykkäästi.

Tietämyksen esittäminen

Skrollin numerossa 2013.3 ilmestyi Panu Kalliokosken artikkeli ”Ymmärryksen juurilla”, joka käsittelee tapoja esittää tietoa tietokoneelle soveltu- vassa muodossa. Tiedon esittäminen formaalisti kiinnosti jo varhaisia tekoälytutkijoita – aikana ennen minkäänlaisia konkreettisia tekoälyn sovelluksia tai edes varsinaisia tietokoneita.

Yleensä lähtökohta on se, että esimerkiksi tavallisena suomenkielisenä tekstinä olemassa oleva tieto pitäisi saada muotoon, jossa sitä pystytään käsittelemään systemaattisesti.

Asiantuntijajärjestelmissä tiedon esittäminen on keskeistä. Nykysovelluksissa logiikka on vahvuksiensa vuoksi hyvin suosittu esitystapa, mutta joskus sovelletaan myös niin sanottuja kehyksiä (*frames, units*). Tapa esittää tietoa vaikuttaa tietenkin ratkaisevasti siihen, miten sitä voidaan käsitellä, joten eri esitysmuotojen kehittäminen ja tutkiminen on tärkeä osa tekoälytutkimusta.

Epävarman tiedon käsittely

Todennäköisyyslaskennan ja tilastotieteen tuleminen mukaan tekoälytutkimukseen oli huomattava edistysaskel. Useimmat ongelmat, joiden ratkaisemiseen ihminen kaipaisi kiipeimmin tekoälyn apua, ovat yleensä sellaisia, joihin liittyy epävarmuutta.



Eräs ensimmäisiä toimivista asiantuntijajärjestelmistä oli Yhdysvalloissa Stanfordin yliopistossa kirjoitettu järjestelmä nimeltä MYCIN, joka antoi ohjeita verenmyrkytyksen antibioottilähtöön. (Veren bakteeri-infektio on hengenvaarallinen tila, jota on hoidettava oikein ja mahdollisimman nopeasti.) MYCIN toimi seuraavan tyyppisten sääntöjen puitteissa:

JOS

(infektio on syntynyt ruuansulatuskanavan kautta JA bakteeri on viljelyssä gram-negatiivinen)

NIIN

(on todennäköistä, varmuudella 70 %, että kyseessä on *Bacteroides*-suvun mikro-organismi JA hoitoon käytetään määrätty annos kloksasilliiniä ja gentamysiiniä)

Tekoälyn soveltaminen lääketieteen konkreettisiin ongelmiin on jo todellisuutta. Toistaiseksi tekoälyt eivät kuitenkaan tee itsenäisiä päätöksiä ihmisen hoidosta, vaan ne vain avustavat lääkäreitä. Tulevaisuudessa tilanne saattaa olla toinen. Hyvä asiantuntijajärjestelmä kun saattaa olla ihmislääkäreitä luotettavampi.

Luonnollisen kielen käsittely

Todennäköisyystiedon eli probabilistisen tiedon käsittely on myös avannut aivan uusia uria luonnollisen kielen (kuten suomen, kiinan ja englannin) tietokonekäsittelyyn. Kielen kaikki merkitykset eivät ole yksiselitteisiä, ja usein puhutun kielen ymmärtäminen vaatii myös arvailua.

Monet ovat jo keskustelleet keinoälyjen kanssa puhelimesta, ja sellai-



set sovellukset kuin Applen Siri ovat levinneet yleiseen tietoisuuteen.

Useissa suomalaisissa pankeissa käytetään puheentunnistustekniikoita. Esimerkiksi Nordea-pankin puhelinvaihteeseen soittaessa ääni pyytää soittajaa muutamalla sanalla luonnehtimaan asiaansa. NLP-järjestelmä (*natural language processing*) selvittää puhelimesta puhutuista sanoista, mitä asia koskee, ja osaa ohjata asiakasta.

Haku

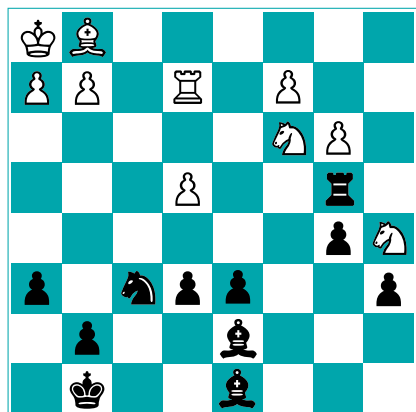
Reittiopasta tai auton navigaattoria käyttäessäsi et välttämättä ajattele mekanisme, joka tuottaa tarjotun tuloksen ruudullesi. Reittihaun takana toimii nokkelia algoritmeja, joiden keksimisessä ja parantamisessa tekoälytutkijoilla on ollut aikanaan työsarkaa.

Nykyään reittioppaan kaltaisten sovelluksien ohjelmoiminen sujuu oppineilta jo rutiinilla eikä se vaadi varsinaista tutkimusta. Kaikkia monimutkaisempia hakuongelmia ei kuitenkaan ole ratkaistu vielä lopullisesti.

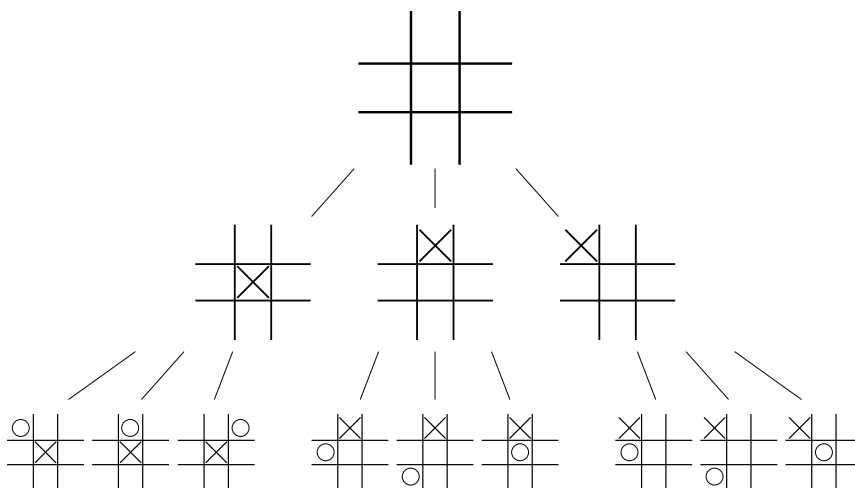
Hakupuu

Myös yksinkertaisten pelitekoälyjen ohjelmoimiseen tarvitaan hakua.

Tyypillinen tapa ohjelmoida yksinkertainen pelitekoäly, esimerkiksi jätäkänšakkia pelaava ohjelma, on laittaa tietokone laskemaan kaikki mahdolliset siirrot ja sitten katsoa, mitkä siirrot johtavat voittoon ja mitkä eivät. Tällä tavalla lasketut eri siirtovaihtoehdot muodostavat puumaisen rakenteen, jossa jokainen siirtovalinta jakaa puun eri haaroihin. Puun latvassa ovat pelin lopputilanteet, joiden perusteella voi jäljittää, mitkä siirrot johtavat voittoon.



Valkoisen vuoro. Millä perusteella laskisit numeerisen arvosanan sille, kuinka hyvä tämä pelitilanne on mustalle?



Jokainen siirtovalinta muodostaa uuden haaran puuhun.

Arvio tilanteesta

Jos peli on yhtään monimutkainen, kuten šakki on, erilaisia siirtovaihtoehtoja on kuitenkin liikaa ja peli on niin pitkä, että puusta tulisi aivan liian iso. Tällöin tapana on laskea siirtoja vain tiettyyn ”syvyyteen” eli siirtomäärään asti ja sen jälkeen arvioida, mikä saaduista pelitilanteista näyttää parhaalta.

Šakissa arviokriteerinä voisi olla esimerkiksi se, kuinka monta ja millaista nappulaa kullakin pelaajalla on jäljellä. Tekoälyn ohjelmoijan keskeinen tehtävä on kehittää mielekäs kriteeri pelitilanteen arvioimiselle. Kun lasketut pelitilanteet on arvioitu, jäljitetään siirto, joka johtaa vastustajan siirroista riippumatta parhaimpaan lopputulokseen.

Edellä kuvatussa tilanteessa siis muodostetaan puumainen rakenne erilaisista siirtovaihdoista. Niiden joukosta haetaan parasta siirtoa tai parasta haaraa. Laskettujen siirtojen joukkoa, josta parasta vaihtoehtoa etsitään, kutsutaan *hakuavaruudeksi*.

Reittihaku

Reittioppaat, jotka kertovat lyhyimmän matkan kahden pisteen välillä tai nopeimman tavan kulkea julkisella liikenteellä paikasta toiseen, toimivat samankaltaisella periaatteella.

Puuta lähdetään rakentamaan tilanteesta, jossa ollaan vielä lähtöpisteessä, ja jokainen valinta jakaa oksan uusiin haaroihin. Valinta voi olla kääntyminen risteyksessä tiettyyn suuntaan tai julkisen liikenteen tapauksessa bussiin nouseminen tai bus-

sista pois jääminen.

Osamatkat, eli osittaiset polut lähtöpaikasta eteenpäin, katuverkoston läpi, muodostavat hakuavaruuden. Tarkoitus on hakea parasta kokonaispolkua lähtöpaikasta määränpäähän. Tämän hakemisen suorittaa hakualgoritmi, joka palauttaa käyttäjälle parhaan reitin lähtöpaikasta katuverkoston läpi määränpäähän.

Pelit

Pelitekoälyissä usein sovelletaan hakualgoritmeja, mutta parhaan siirron hakeminen on harvoin riittävää.

Pelit ovat tekoälytutkimukselle kätevä kohde, koska niiden säännöt muodostavat yksinkertaisen ”mikromaailman”, joka on matemaattisesti käsiteltävissä. Tosimaailmassa on loputtomasti erilaisia huomioon otettavia asioita, mutta pelin maailma on rajattu.





Älytöntä rutinoitumista

Loogisia, siirtopohjaisia pelejä voi ratkoa hakualgoritmiin avulla. Jos pelibotti perustuu pelkästään hakualgoritmiin, ei sitä voi kuitenkaan enää pitää älykkäänä. Kyse on vain rutinoituneesta keittokirjaston soveltamisesta. Koska asia osataan tehdä mekaanisesti, siitä on tullut itsestäänselvyys.

Monia pelejä ei osata kuitenkaan vielä ratkaista rutinoituneesti.

Tammi, šakki, go, backgammon...

Tammipeli on analysoitu tietokonein. Šakissa kone voittaa ihmisen.

Go-pelissä hakuvaruus on kuitenkin niin suuri, että koneet eivät vielä voita ihmispelaajaa. Ihmisen ylivoimaisuus gossa tekoälyä vastaan perustuu myös siihen, että ihminen on paljon nykytekoälyjä parempi hahmotamaan visuaalisia asioita. Koneen kehittäminen onkin yksi tärkeä tekoälyn tutkimusalue tällä hetkellä.

Tammipelin tekoäly muistuttaa paljon šakkia. Korttipelejä pelatessa joudataan taasen arvioimaan todennäköisyyksiä ja laskeskelemaan erilaisia asioita, kuten vaikka bridgessä sitä, millaisia kortteja vastapuolella saattaisi olla. Backgammonin tekoäly on yhdistelmä šakkityypistä hakualgoritmia ja todennäköisyyksien laskemista. Backgammonissa koneet usein voittavat ihmisen.

Robotiikka

Tekoälyn eri osa-alueet kietoutuvat toisiinsa tiukasti. Edistysaskel yhdellä alalla poikii sovelluksen toisella. Robotiikka, jonka saamaa erityishuomiota tämä artikkeli pyrkii jakamaan laajemmalle alueelle, soveltaa kaikkia tekoälyn osa-alueita.

Robotiikkaa voidaan toki tarkastel-

la myös erillisenä osa-alueena, johon rajataan vain fyysisten laitteiden älykkäiseen liikkumiseen tarvittava tekniikka. Useimmiten robotit kuitenkin käsitetään kokonaisuuksina, johon kuuluu myös muuta älyä.

Metallinen Maradona

Jalkapalloa pelaavan robotin on ratkaistava monia eri ongelmia. Miten se havaitsee ympäristöään (konenäkö), miten se liikkuu (elektroniikka ja sitä ohjaava koodi), miten sen kannattaa pelata (pelitekoäly)... Tämän lisäksi robotin eri osien on osattava kommunikoida keskenään, ja robotin on osattava tulkita eri sensoreistaan saamaansa tietoa. Jo pelkästään robotin sujuva liikuttaminen pelikentällä on laaja ongelma sinällään.

Keinoneuroniverkot

Jos perinteisestä tekoälyteknologiasta ei ole ratkaisemaan jotakin ongelmaa, on tarjolla tuoreitakin ratkaisuja. Tietokoneitten alati kasvava laskentateho on mahdollistanut sen, että nykyään on mahdollista simuloida tietokoneella ihmisen hermosoluja. Tällä tavoin syntyy toimivia tekoälysovelluksia.

Keinotekoiset neuroniverkot (*artificial neuron networks*) usein osaavat tehdä asioita, joiden tekeminen perinteisellä tekoälyteknologialla on vaikeaa. Niitä on käytetty postin lajittelussa käsialalla kirjoitet-
tujen

numeroiden tunnistamiseen sekä pankkikorttipetosten tunnistamiseen pankkikorttitapahtumia analysoimalla. Myös puhutun kielen tunnistamisessa niitä on onnistuttu soveltamaan hyvin tuloksin.

Koneoppiminen

Koneoppiminen on toinen uudenpuoleinen tekoälyteknologia.

Hyvin lyhyesti sanoen koneoppiminen merkitsee sitä, että järjestelmälle annetaan dataa, ja se luo sen perusteella uutta tietoa. Uusi tieto voi olla matemaattisen mallin tai esimerkiksi logiikalla ilmaistun tietämyksen muodossa. Voisi sanoa, että koneoppiva järjestelmä luo datasta informaatiota.

Tiedonlouhinta

Tietokannoista voidaan *louhia* tietoa. Tätä tiedonlouhinta (*data mining*) voi pitää koneoppimisen alalajina. Tekoälyjärjestelmälle annetaan hyvin suuri tietokanta – tai useita – ja sen tehtävä on löytää aivan uutta tietoa tietokannasta.

Mitä tämä uusi tieto sitten voisi olla? Suuri vähittäiskauppakettu voi haluta tiettyjä tietoja asiakkaiden ostotottumuksista voidakseen tehostaa toimintaansa. Tai sitten voidaan hakea uusia tietoja radioteleskooppien tuottamasta suunnattomasta datamäärästä. 📡





Lukijan projekti – Raspberry Pi -kannettava

Olen 12-vuotias poika Oulusta ja olen ollut kiinnostunut tietokoneista jo esikoulusta asti. Olen aiemmin ohjelmoinut pelejä Raspberry Pi -tietokoneelle, ja vuosi sitten sain idean koota itse tietokoneen. Tämän tietokoneen keskusyksikkönä toimii Raspberry Pi, yhden piirilevyn tietokone.

Teksti: Arttu Leskelä Kuvat: Arttu Leskelä, Ismo Leskelä

Idea

Sain jo vuosi sitten idean kannettavan tietokoneen kokoamisesta itse, mutta en lähtenyt toteuttamaan sitä heti. Olin ostanut kannettavaa varten sopivan näytön, mutta se päätyikin televisioksi. Idea tuli uudestaan mieleeni tänä vuonna, kun päivitin komponentteja isäni vanhaan koneeseen.

Toteutuneeseen kannettavaan tietokoneeseen päätyivät seuraavat komponentit: langaton USB-näppäimistö ja -hiiri, ProCaster DTV-007 -näyttö ja D-Link USB wlan -adapteri. Tietokone on rakennettu metalliseen suojakoteloon.

Kokoaminen

Sovitin ensin näytön laatikkoon ja porasin laatikkoon virtajohdoille ja antennille paikat. Käytetty näyttö on valmistettu television katsomista varten, ja tämän kannettavan erikoisominaisuus onkin tv-signaalin vastaanotin. Näytössä on myös komposiittiliitin, joka mahdollistaa Raspberry Pin liittämisen näyttöön. 🛠️



Kotelon kylkeen on porattu reiät liittimiä ja antennikaapelia varten.



Läppärin osat on koottu kovaan laukkuun, jonka saa näppärästi suljettua kuljetusta varten.

Korpimaiden sankarit

Avaruussimut ennen ja nyt

Ihmiset ovat haaveilleet tutkimusmatkoista kosmokseen satojen vuosien ajan. Tiedemiesten ja tieteiskirjailijoiden visiot ovat siirtyneet viimeisen kolmen vuosikymmenen aikana valkokankailta ja kirjojen välistä myös tietokonepeleihin. Skrollin avaruuslaiva matkustaa käymättömien korpimaiden tuolle puolen, galaksien kimalteleviin syvyyksiin. Teksti: Jukka O. Kauppinen

Kuvat: Frontier Developments plc, GameTek, computerspacefan.com, Wikimedia Commons (Phrood)



Videopelit ovat ammentaneet science fictionista materiaalia aivan ensimmäisistä pelikoneista lähtien.

Videopelien isän arvonimeä kantava Nolan Bushnell loi markkinat räjäytäneen *Pongin* vuonna 1972, mutta sitä edelsi avaruustaistelusimulaattori *Computer Space* (1971) joka pohjautui vuonna 1962 tehtyyn *Spacewar!* -peliin.

Niin, avaruus on videopelien ystäville tuttu juttu. Mutta vuonna 1984 avaruuspelien genre muuttui täydellisesti. Alkoi *Eliten* aikakausi.

Siitä lähtien pelaajat ovat tähynneet digitähtitaivaalle kaihoten, sillä tähdet kuiskivat lupauksia seikkailuista ja rikkauksista. Mutta kulta ja kunnia eivät tule koskaan helpolla. Avaruuspörssiin kurssit ovat arvaamattomat, ja silkkitideiden varsilla kauppiaita vaanivat piraatit ja onnettomuudet. Joskus ainoa laki löytyy oman aluksen lasereista, joskus on pakko luottaa takasuojakilpien kestävyYTEEN ja nopeuteen.

Avaruuden gravitaatiovoimat vaihtelevat. Toisinaan tähdet kimaltelevat kirkkaina ja kutsuvina. Sitten niiden vetovoima hälvenee olemattomiin. Ruuhkaiset avaruussatamat tyhjenevät ja viimeiseltä tähtiasemalta operoi enää vain pari tähtilaivayhtiötä, joiden pienet alukset kuljettavat puolityhjinä matkustajiaan muutamalle lähitähdelle.

Avaruus on tyhjä. Edes piraateille ei riitä saalista, kun suuryhtiöt keskittyvät kaivamaan napaansa. Miksi sijoittaa riskialttiiseen tähtikauppaan,

kun mikromaksut, mobiilipelit ja se-lainseikkailut tuottavat enemmän ja vähemmällä riskillä?

Kunnes eräänä päivänä maapallon asukkaat keksivät joukkorahoituksen ja osoittivat, että tähtien kutsu ei ole laantunut. Kutsusignaalia vain lähete-tään eri taajuudella. Ja yhtäkkiä tähdet tuikkivat taas kirkkaasti kutsuen.

Pongin varjosta...

Aikana, jona videopelit olivat vasta hädin tuskin kasvaneet ulos *Pongin* (1972) ja *Pac-Manin* (1980) perinnöstä, muutama visionääri näki tietokoneiden mahdollisuudet johonkin suurempaan. Kotimikrot kehittyivät 80-luvun alussa valtavasti harppauksin, kuten Skrollin numeron 2014.2 Tasavallan tietokoneen haastajat -artikkelissakin muisteltiin. Koneiden muisti kasvoi, suorittimet nopeutuivat, grafiikka- ja ääniominaisuudet kehittyivät valtavilla loikkilla. Muutamassa vuodessa videopelit muuttuivat palikkagraafisista piipittäjistä visuaalisiksi ja näyttäväiksi pikselitaidepläjäyksiksi.

Ensimmäinen avaruusseikkailu/kaupankäyntisimulaatio *Star Trader* ilmestyi 1974. Basicilla tehty tekstipohjainen kauppasimulaatio pohjautui Isaac Asimovin Säätiö-kirjoihin ja tarjosi galaksin, jossa kaupparatsut saattoivat kiittää rahteineen aurinkokunnasta toiseen. Siitä kehittyi *Trade Wars* (1984), joka oli yksi BBS-aikakauden suosituimpia moninpelejä. Mutta se on oma tarinansa.

Star Traderia seurasi ensimmäinen kotimikroille tehty avaruusseikkailu,

brittiläinen *Merchant of Venus* (1982). Sinclair ZX81:lle tehty peli sisälsi ka-russa muodossa useimmat tulevien vuosikymmenten avaruussimulaatioiden ainekset. Tekstipainotteisessa seikkailussa pystyi ostamaan erilaisia avaruusaluksia, joilla lennettiin tähdeltä toiselle ostaen ja myyden. Taisteluja pelissä ei ollut, mutta vaara vaani toimintaosiossa jossa alus ohjattiin uusien planeetoiden telakointialustoille.

Vaikka *Merchant* ei ollutkaan 3D-avaruussimulaatio, se edusti avaruus-seikkailu- ja kaupankäyntigenren varhaisinta sukupolvea. Siihen nähden onkin hämmästyttävää ajatella sitä uskomatonta teknologista ja sisällöllistä loikkaa, joka koettiin vain kahden vuoden kuluttua.

Welcome Commander Jameson

David Braben ja Ian Bell olivat märkäröisiä yliopistojunnuja *Eliten* luodessaan. Shoot'em uppien ja tasoloikkien hallitsemalla kentällä peleissä oli kolme elämää, tarkoin määritellyt reunat ja tavoitteet. Niihin verrattuna *Elite* (1984) oli kuin supernova.

Se oli avaruusseikkailu ilman rajoja. Pelaaja pudotettiin Lave Stationille Cobra Mk. III -avaruusaluksen kera. Mitä teit ja minne menit, se oli omis-sa käsissäsi – oletko rauhanomainen kauppias, ahne piraatti, seikkailija tai avaruuskaivosmies? Pelissä ei ollut lainkaan tarinaa eikä kokonaisuutta yhteen sitovia tehtäviä, vaan jokainen loi oman tarinansa teoillaan ja mielikuvituksellaan. Pidemmälle pelanneet

COMPUTER SPACE



Pelaajat matkasivat avaruuteen jo ensimmäisten tietokone- ja kolikkopelien myötä, ja Computer Space jalosti Spacewarin pelihalleihin.



tosin saattoivat kohdata joitakin galaksien uumenissa odottavista tehtävistä. Vapaus oli käsittämätöntä.

Käsittämätön oli myös maailmankaikkeuden koko. 64 kilotavuun ahdettiin kahdeksan galaksia, 2048 planeettaa yksilöllisine yhteiskuntajärjestelmineen, avaruusaluksia ja teknologiaa.

Hiekkalaatikkomaisuus yhdistyi teknologiseen äimistyttyvyyteen, sillä rautalanka-3D:llä kuvattu avaruus ja kuvakulma aluksen ohjaamosta olivat

täysin uudenlaisia. ”Missä se alus on”, moni ihmetteli.

Eikä pidä unohtaa huumoria ja salaisuuksia! Toisinaan pelaajat kohtasivat galaksin uumenissa täysin odottamatomia asioita. Ehkä hyperavaruudesta löytyi muukalaisten armada. Näitkö oikeasti TIE Fighterin vai oliko se pitkän pelisession tuottama öinen harha? *Eliten* syväavaruudessa elää yhä monia vahvistamattomia legendoja.

Ehkä se on klassikon tunnus: teok-

Roolipelimäinen sivuloikka

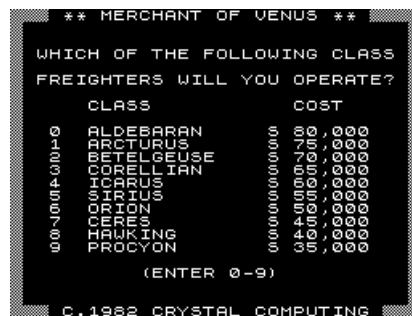
Avaruuteen mahtui simulaatioiden lisäksi muutakin, myös seikkailuja. Roolipelitalo Origin Systems teki 1989 jännittävän genrelöikan, sillä Space Roguessa yhdistyi klassinen roolipelaaminen ja avaruussimulaatioseikkailu.

Rogue rakensi klassisen *Eliten* päälle vaikuttavan pelikokemuksen, jossa riitti 3D-grafiikkaa, tehtäväpolkuja, avaruustaisteluja ja perinteisiä valintoja piraaatin, kauppiaan ja palkkasoturin roolien välillä. Ja minipelejä.

Roguen ainesten yhdistelmä taisi kuitenkin jäädä aikoinaan heikolle huomiolle, tai sitten se käsitettiin vähän liikaakin roolipeliksi. Niinpä seikkailu elää enää lähinnä avaruusgenren hämärillä sivulehdillä. Aikanaan se kuitenkin tarjosi mahdollisen ja mukaansatempaavan, ennen kaikkea omaperäisen matkan galaksien uumeniin, eikä *Elite*-genreläisten kannata sivuuttaa sitä olankohautuksella.



Spacewar oli ensimmäinen tietokoneavaruustaistelupeli.



Avaruusseikkailu vuonna 1982: Merchant of Venus.



Elite tarjosi unohtumattoman galaktisen seikkailun lisäksi myös hikisiä hetkiä, sillä telakoituminen avaruusasemaan oli pirullisen vaikeaa.



Commander Jameson aloitti galaktisen seikkailunsa Lave Stationilta. Se nähdään myös *Elite Dangerous*issa.

sen ympärille syntyy tarinoita, joita veteraanit kertovat toisilleen yöllä lonkerobaarissa tyhjennettyään tarpeeksi monta avaruussimatuoppia.

Hämmästyttävää on sekin, että loppujen lopuksi *Elite* vain nivoi yhteen edeltävien kymmenen vuoden aikana tehtyjen avaruusseikkailujen ainekset. Toki täysin ylivermaisella tavalla, mutta genren pohja oli jo olemassa. Vuoden 1984 *Elite* oli niin täydellinen peli, ettei siitä ole voitu juuri pistää paremmaksi kolmessa vuosikymmenessä.

Upea floppi – Federation of Free Traders

16-bittisen aikakauden saapuessa moni halusi tehdä definiitiivisen avaruuspelejä. Mutta miten tehtäisiin parempi *Elite*, joka ei olisi *Elite*?

Eliten omat 16-bittiset käännökset olivat ok. Hyviä mutta eivät käänteentekeviä, edes Amigalla. Brittikaksikko Warren Burch ja Clive Gringras puolestaan loivat Acorn Archimedes -tietokoneelle niin hyvän *Elite*-kloonin, että se julkaistiin lopulta virallisena käännöksenä. Puristien mielestä *ArcElite* (1991) on jopa kaikkein paras alkuperäis-*Elite*.

Vaikuttavin ja korkeimmalle kurottuva elitesointi oli kuitenkin *Federation of Free Traders* (1989), liiankin tyylikäs ja yksityiskohtainen avaruussimulaatio. Kahdeksan miljoonan planeetan galaksissa riitti lennettävää, ja tekemistä piisasi niin vapaan tutkimusmatkailun kuin kaupankäynnin merkeissä. Siis teoriassa. Käytännössä peli kompastui kikkailuunsa.



Frontier: Elite 2:ssa oli sata miljardia planeettaa 880 kilotavussa.



Federation of Free Traders halusi olla Eliteä Elitempi. Ei ollut.

Avaruuslentämisen ohella pelissä oli planeettaosio, jossa pystyi puikkelehtimaan ilmatorjunnan läpi planeetan pinnalla sijaitsevalle avaruusasemalle. Osio muuten muistutti enemmän kuin vähän Brabenin toista kasaripeliä, *Virusta* (1987). Varmasti vain sattumalta.

*FoFT*issa oli myös sisäinen SIMPLE-ohjelmointikieli, jolla peliin saattoi koodata omia apuohjelmia avaruusmatkailua helpottamaan, tai yksinkertaisia pelejä. En ole löytänyt todisteita siitä, että kukaan olisi ikinä jaksanut tehdä sillä yhtikäs mitään. Avaruudessa kohdattujen alusten kanssa saattoi myös keskustella. Tekstipeliksi jutustelusta ei tosin ollut, sillä parseri oli kökö ja sanavarasto mitätön.

Pelin talousjärjestelmä oli pahiten rikki. Mitä suotta kiertelemään galaksia, kun pankkitilin sai täyteen ensimmäisen avaruusaseman kauppapaikalla ja pörssiä pystyi poistumaan asemalta viimeisen päälle varustellulla aluksella. Ja jos galaksia jaksoikin tutkia, niin tehtävät olivat tylsiä ja huonosti ohjeistettuja. Vasta jossain monen tylsän tunnin päässä odotti tehtäviä, jotka tarjosivat turbopilotille edes alkeellista haastetta. Harva jaksoi sinnitellä niihin saakka.

Sääli, sillä *FoFT* yritti kovasti. Paul Blythen rakentama galaksi oli avaruuspiraatin unelma, sillä ihmiskunnalla ei ollut enää edes avaruuspoliisia. Jokainen oli oman onnensa nojassa. Tämä tarjosi uskalikoille mahdollisuuksia, joita harva jaksoi tavoitella etenkin sen jälkeen, kun alkuperäinen Atari-versio vedettiin kaupoista pelikelvottoman bugisena.

1993 oli täynnä tähtiä

»Brabenin työn alla olevaa *Elite II*:ta verhoaa nimittäin niin suuri salaperäisyys, että kenties vielä tänä vuonna...»

Jyrki J. J. Kasvi, C=Lehti 1/1989

90-luvun alku oli avaruussimulaatioiden kannalta hiljaista aikaa, mutta sitten rysähti. Kaksi legendaarista pelisarjaa avasi tähdet uuden sukupolven silmin. *Wing Commander* -universu-

miin sijoittuva *Privateer* (1993) ja *Frontier: Elite II* (1993) olivat pelejä, joita rakastettiin – ja ihan vähän vihattiin.

Kolme vuotta ensimmäisen *Wing Commanderin* jälkeen julkaistu *Privateer* jatko-osineen lähestyi avaruusseikkailuja Larry Nivenin *Tunnettu avaruus* -saagasta kähvelletyllä perusidealla ja *Wing Commanderin* maineella. Pieleen meni. Tarinavetoinen avaruusooppera tyssähti tarinan surkeuteen ja vähyyteen, epäuskottavuus ja ylipäättään sisällön onttoisuus eivät kantaneet juuri kynnystä pidemmälle.

Ei ihme, että Originin eurooppalaisen haaran luoma *Privateer 2* (1996) suuntasi eri suuntaan. Kakkosta ei jatko-osaksi tunnista, sillä se hylkäsi koko *Wing Commander* -maailmankaikkeuden ja rakensi tilalle *Blade Runner* -henkistä cyberpunkkia vilisevän universumin.

Synkistelevä linjanmuutos toimi, ja pelistä rakentui oiva avaruuden kauppatkaajasimulaatio. Lennettäviä ja kustomoitavia aluksia riitti, tarinan ympärillä piisasi tekemistä eikä peli muuttunut päättömäksi hortoiluksi, ellei sitä nimenomaan kaivannut. Genren perinteitä kuitenkin pilkattiin, sillä pelaajalta vietiin valinnanvara. Pahis ei voinut olla!

Sitten rysähti. David Braben sai lopulta valmiiksi *Elite II*:n. *Frontier* jätti kuitenkin jälkeensä ristiriitaisen perimän. Yhdet sitä rakastivat, toiset vihasivat. Kansa jakautui, kuten *Tunnettu avaruus*kin.

Frontier oli valtava. Monipuolinen. Avoin, dynaamisesti generoituva ja buginen. Vähän hengetönkin. Ehkä Ian Bellillä oli Elitessä suurempi osa kuin yleensä tunnustetaan, sillä kun kukaan ei ollut hillitsemässä Brabenia, tämä vauhkootui täysin. Sata miljardia yksilöllistä planeettaa (880 kilotavussa!) ja Newton-avaruusfysiikat eivät tee välttämättä hyvää peliä. Silti *Frontier* vakuutti. Sen universumi oli avoimuudessaan uskottava, joskin tyhjä. Peli oli hyvin suunniteltu ja pelattava.

Lisälevystä jatko-osaksi laajennettu

Frontier: First Encounters (1995) sen sijaan jätti monille pitkäksi aikaa pahan maun suuhun. Myös *Elite III*:ksi kutsuttu elitesointi oli upeasti monipuolistunut ja tarkasti pelaajien palautteen mukaan viilattu galaktinen seikkailu, jossa oli ensimmäistä kertaa myös kunnon tehtäviä. Jos halusit muutakin kuin tähtitoikkarointia, sitä oli tarjolla! Mutta kylläpä osasi pelissä olla bugeja. Niitä oli niin paljon, että Brabenin Frontier Developments haastoi julkaisija Gametekin oikeuteen. Frontier voitti oikeudenkäynnin, mikä oli tärkeä signaali hätäileville kustantajille – kehittäjä saattaa suuttua, jos peli julkaistaan keskeneräisenä!

Päivitettynä *First Encounters* on *Elite*-sarjan vanhan polven laajin, monipuolisin ja pelattavin teos, vaikka ei esikoisensa hurmaavaa yksinkertaista tavoitakaan.

Hulluutta, neroutta, sekopäisyyttä

Eliten sukulaisista puhuttaessa tarinaan täytyy tuoda myös koominen sivuhahmo, Derek Smart, todellinen legenda ja upea sirpale pelialan historiaa. Hän oli nimittäin ensimmäinen ammattimainen pelinkehittäjätrulli.

Smart loi maineensa internetin keskusteluryhmissä raivopäisellä mekastuksellaan. Niinpä *Battlecruiser 3000 AD* (1996) ei jäänyt pelaamisen historiaan siksi, että se olisi ollut hyvä peli. Se jäi historiaan siksi, että se oli netin flameissa kypsytetty pökäle, joka oli parhaimmillaankin nipin napin toimiva läjä sekavia visioita ja ideoita.

Battlecruiserin piti olla ennen näkemättömän laaja, rajaton avaruusooppera. Mutta kun sitä pusersi seitsemän vuoden ajan yksi mies, joka vietti enemmän aikaansa nettisoturina kuin pelinkehittäjänä, tulos oli tekijänsä näköinen: kummallinen ja kelvoton. Tekninen toteutus oli surkea, avaruustais-telut pilipalia ja lentomallit puhdasta Pelle Hermannia. Pelaaja hukutettiin avaruusristeilijän mikromanageroinnin ja sadistisen käyttöliittymän suo-

Planetary Raiders oli ensimmäinen Elite Online

WarBirds-nettilentosimulaattoristaan tunnettu jenkkiläinen ICIgames oli aikoinaan ensimmäinen internetin MMO-tähtikehittäjä, jo ennen kuin Ultima Online teki morpeista muotia. ICI kehitti huippuluokan online-pelitekniikkaa, jota päätettiin hyödyntää toisen maailmansodan lentosimulaattorien lisäksi myös avaruuspeleissä.

Planetary Raidersista oli syntymässä hyvää vauhtia internetissä pyörivä virtuaaligalaksi, jonne mahtuu yhtä aikaa tuhansia pelaajia. Pelaajat pystyivät käymään kauppaa, taistelemaan ja ryöstelemään toisiaan. Kaupasta löytyi parempia avaruusaluksia viriteltäväksi. Raidersista oli rakentumassa hyvää vauhtia peli, jollainen Eve Onlinesta myöhemmin tuli. Projekti kuitenkin kaatui keuhkoon yrityskauppaan, jonka myötä uudet omistajat kutistivat avaruusmörkin kevyeksi taistelusimulaatioksi. Raider Warsin taru, jos sitä ehdittiin edes varsinaisesti julkaista, jäi lopulta hyvin lyhyeksi.

hon, joiden keskellä kahlatessa varsinainen avaruusseikkailu, kaupankäynti ja syväavarouden freelancer-poliisin toimiminen puuroutuivat täysin.

Ja jos toteutus oli oksettava, niin sitä oli liekitetty hypekin. Muut pelikehittäjät nauroivat partaansa Smartin väitteille muun muassa galaksia ohjaavasta neuroverkosta.

Kun peli lopulta saatiin kauppoihin, oli se jo viidennen peräkkäisen julkaisijan listoilla. Julkaisun jälkeiset internet-sodat ja Smartin oikeudenkäynti Take-Twota vastaan jäivät nekin pelialan historiaan, mutta herralle on annettava pisteet sinnikkydestä. *Battlecruiser 3000AD V2.0* (1998) ja *Battlecruiser Millennium* (2001) / *Gold* (2003) viilasivat bugipökäleestä lähes pelin, eikä siihen mennyt kuin liki

puolitoista vuosikymmentä.

Smart on muuten yhä mukana kuivoissa. 3000AD-yhtiön viimeisin julkaisu on avaruustaistelupeli *Line of Defence*.

Pitkä, synkkä teeaika

First Encountersin ja *Battlecruiserin* jälkeen avaruusseikkailujen genre vaihtui pitkään, synkkään koomaan. Vain harva satunnainen rokkelikko uskalsi kääntää katseensa kohti välkkyviä pulsareita ja tavoitella tähtien seireenejä. Ei avaruus tyhjä ollut, pikemminkin sotaisa. Seikkailu ja kaupankäynti antoivat tilaa sotureille, joiden alusten peräsimiä koristivat Kesslerin reitin läpilentojen sijaan merkinnät tapoista.

Jotkut silti yrittivät. Saksalainen *X: Beyond the Frontier* (1999) jatkoosineen ja lisäreineen on kannatellut *Eliten* lippua kunnialla. Äksät ovat tasapainotelleet oivasti taistelun, seikkailun ja kaupankäynnin välimaastossa, ja sarjan viimeisin teos *Rebirth* mahdollistaa kauppa-aluslaivaston ja avaruusasemista koostuvan bisnesimperiumin rakentamisen. Genren ystäville hyvä tutustumiskohde!

Independence War 2: Edge Of Chaos (2001) on puolestaan jäänyt vähän varjoon, mutta joidenkin fanien mielestä se on ehkä kaikkien aikojen paras avaruustaisteluseikkailusimulaatio. Vaikka universumi koostuukin "vain" 16 aurinkokunnasta, on siihen rakennettu upea vapauden ja seikkailemisen tuntu, hieno tarina ja paljon tekemistä.

Myös Microsoftin *Freelancer* (2003) puraisi *Elite*-pullasta, vaikka meininki olikin sotaisaa. Avaruusseikkailu kehitettiin isolla rahalla, joten siinä yhdistyivät poikkeuksellisesti sekä mittavat tuotantoarvot että hyvä pelattavuus.

she really wants it



Battlecruiser 3000 AD herätti kyseenalaista huomiota paitsi tekijänsä nettimesoamisella ja bugeillaan, myös mainoksillaan.

Kiintoisana ratkaisuna peli laajeni pelaamisen myötä siten, että tarina jäi hiljalleen sivuun ja hyvin suunniteltu universumi avautui vapaasti seikkailtavaksi. Hyvät ideat jäivät kuitenkin kertakäyttöisiksi.

Kenties merkittävin *Elite*-seikkailu sitten alkuperäisen on islantilaisen CCP Gamesin *Eve Online* (2003). MMO tarjoaa pelaajilleen samaa käsitämätöntä pelaamisen vapautta kuin *Elite*-genren edustajat parhaimmillaan, mutta anarkistisen vapaassa ja avoimessa, satojentuhansien muiden ihmisten kansoittamassa avaruudessa. Vaikka *Eveä* kuvaillaan yhtäällä taulukkolaskentasimulaatioksi, joka pelaa itse itseään ja toisaalla nettidraaman generaattoriksi, yhtä kaikki se on arvaamaton ja mukaansatempaava avaruusseikkailu.

Iso osa nettipelin lumosta pohjautuu galaksia kansoittavien ihmisten arvaamattomuuteen ja pelaajayhteisö ravisteleviin jättiläisjuoksuun, sotiin ja tempauksiin, mutta myös yhteisöjen yhteishenkeen – ja tietenkin yksittäisen pienen pelaajan omaan puuhasteeluun. *Eliteä* parhaimmillaan!



Wing Commander: Privateer ei saanut siipiä, vaikka alkuasetelma olikin täydellinen.



Eve Online oli lähes vuosikymmenen ajan ainoa varteenotettava suuremman mittaluokan avaruusseikkailu.

Matkustajat Maapallo–Kassiopeia-lennolle portille 1

Avaruussimulaatioiden kuolema pakotti lukemattomat tähtiratsastajat varhaiseläkkeelle. Pitkä, synkkä lama autioitti kiertoratahotellit ja tyhjensi asteroidien etäasukkaat. Tulevaisuus oli nähty, nyt oli aika keskittyä selaimessa pyöriin maailoihin.

Menetetty tulevaisuus teki kuitenkin yllättävän paluun. Kaikesta voidaan kiittää amerikkalaista internet-sivustoa, Kickstarteria. Sen avulla vuoden sisään menestyi kolme joukkorahoitettua projektia: kaksi peliä ja yksi laite.

Laite oli virtuaalitodellisuuskypärä Oculus Rift. Se teki lopultakin totta vuosikymmenien unelmasta, virtuaalitodellisuuden muuttumisesta todeksi – tai ainakin uskottavaksi lupaukseksi. Viitteessä virtuaalitodellisuutta on kuvailtu ja koettu jo vuosikymmeniä, mutta videopeleissä ja todellisissa, käyttökelpoisissa virtuaalilaseissa ja -kypärisissä tekniikka on tuottanut vain pettymyksiä.

Oculus Riftin myötä virtuaalitodellisuuden tulemistä on kuitenkin povattu kiihkeästi jo toista vuotta, ja sen saapuminen hämmöittää taivaanrannassa. Seuraava suuri kysymys on: mihin virtuaalitodellisuuslaitteita voidaan oikeasti käyttää? Jäävätkö ne hyödyttömiksi leluiksi ilman järkevää käyttöä? Vai tuleeko peleistä virtuaalitodellisuuden pelastus? Ja jos niin, mistä peleistä?

Virtuaalitodellisuuspelien suurin haaste on löytää sopivat peligenret ja sopivat peliohjaimet, joiden avulla pelaaja pystyy sukeltamaan pelien immersioon. Pelaaja ei saa häiriintyä ul-



Eliten universumi kunnioittaa upealla tavalla niin alkuperäisen Eliten henkeä kuin myös tiedemiesten ja tieteiskirjallisuuden upeita näkemyksiä tulevaisuuden avaruusmatkailusta.

kopuolisista tekijöistä eikä kaatua kesken yllättävien peliliikkeiden. Niinpä erityisesti auto- ja avaruussimulaatioiden kehittäjät miettivät myös virtuaalitodellisuutta. Molemmissa pelaaja voi istua paikallaan tuolissa, pelinmukaisessa asennossa, kädet ohjaimilla.

Oculus Riftin rinnalla Kickstartersissa menestyi myös kaksi avaruussimulaatiota. *Wing Commanderin* ja *Elite: Dangerousin* menestys teki pelialahistoriaa ja nosti avaruusseikkailut takaisin pelaamisen eturintamaan. Tuskinpa on sattumaa, että useampiakin pelitalo on ilmoittanut uusista avaruussimuprojekteistaan.

Ei tullut Elite IV:ää, tuli Dangerous

David Braben ja Frontier Developments olivat työstäneet *First Encountersin* jälkeen seuraavaa *Eliteä* monen monta kertaa. Tavatessani Avaruuskien Herran vuonna 2007 tämä kertoi, että projektia on hiottu jo useita vuosia ja ”peli näyttää hyvältä, mutta emme kerro siitä pahem-

min julkisuuteen ennen kuin peli on tarpeeksi hyvällä mallilla.”

Jo tuolloin Frontierilla oli ollut yksi tiimi tekemässä yksinomaan *Elite IV:ää*, mutta projektia vaivasi toisaalta täydellisyys tavoittelun vaikeus, toisaalta pelitekniikan odottamattoman nopea kehittyminen ja pelimarkkinoiden täydellinen kiinnostamattomuus. Avaruus oli out, julkaisijoita ei kiinnostanut eikä Frontierilla ollut varaa kehittää peliä itsenäisesti julkaisukuntoon.

Kunnes Kickstarter.

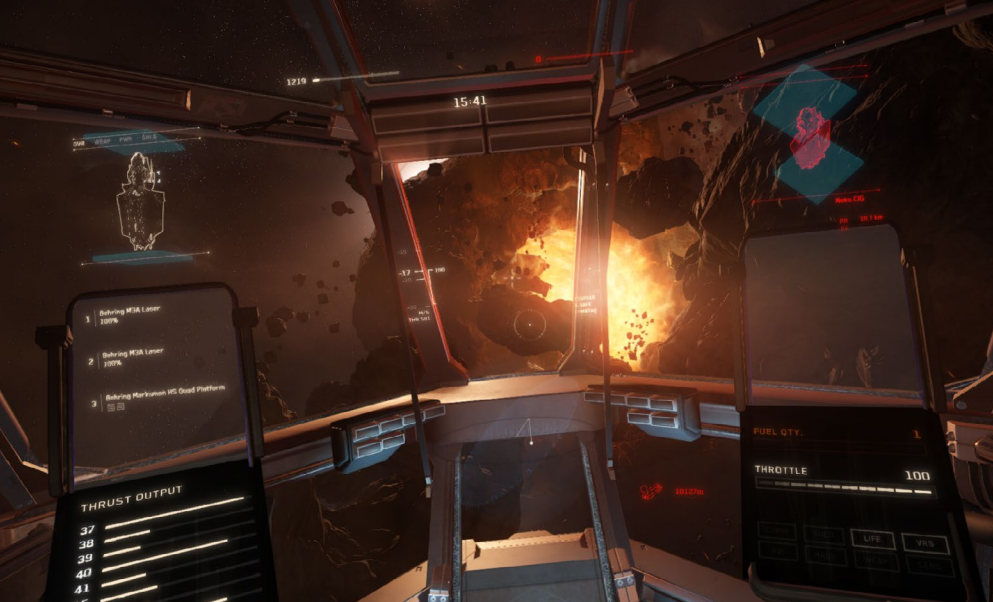
Braben ja Frontier ovat sen jälkeen tehneet hurjasti töitä uuden *Eliten* parissa. Projekti on edennyt hienosti ja aikataulussa: ei vähiten siksi, että asialla on kunnan tiimi joka on tukenut pelinteossa Frontierin olemassa olevaan teknologiaan. Pelinkehitys on ollut sivustakatsojan silmiin fokusoitunutta, ja tiimi on puskenut *Eliteä* selvästi määrättyyn suuntaan.

Ja hienoa jälkeä onkin syntynyt. Eri testivaiheiden aikana pelaajat ovat löytäneet *Elitestä* valtavasti mahdollisuuksia, seikkailuja ja pelityylejä. Varovaisen kokeilijan silmin seurattuna esimerkiksi salakuljettajien äänettömät, moottorit sammuksissa lentämät rahtilennot ja taitavien pelaajien avaruustaistelut ovat olleet uskomattoman näyttäviä ja lumoavia.

Tätä lukiessanne *Elite: Dangerous* on jo virallisesti julkaistu, mutta se on vasta pitkän taipaleen alku. Pelimaailma kehittyy, peli kehittyy ja ongelmat kärjistyvät. Frontier kertoi kuukautta ennen virallista julkaisua, että peli ei tuekaan perinteistä offline-yksinpeliä. Siitä seurasi kunnan nettiraivo. Ilmoitus ei tarkoita ettei *Dangerousissa* olisi yksinpeliä, mutta pelaamiseen



Elite Dangerous on upea, tunnelmallinen ja jopa huvittava. Se tunne, kun lähestyt avaruusasemaa värikkäiden neonmainoskylttien kujan läpi...



Chris "Wing Commander" Robertsin Star Citizeniin tehdään sekä MMO-avaruuseikkailua että tarinavetoista avaruusoopperaa yksinpelureille.

tarvitaan jatkuva nettiyhteys. Joukkorahoituksen alussa Frontier lupasi että offline-yksinpelaamiseen tarvitaan vain yhteys keskuspalvelimeen silloin tällöin, joten minimaalinenkin jatkuva yhteysvaatimus tuntui isolta muutokselta.

Sekä Frontierin että pettyneiden näkemykset on helppoa ymmärtää. Toiset kokevat, että heitä on petetty. Frontier taas kertoo, että puhdas offline-kokemus ei enää toimi heidän visiossaan, jossa tuhansien ihmisten offline-kokemukset ja verkkopelaajat yhdistyvät 24/7 elävään, muuttuvaan galaksiin.

Internetin kohu-aallot loiskuvat toki lähes aina yli äyräidensä, niin luultavasti tälläkin kertaa. Jos Eliten lopputulos vastaa odotuksia, luvassa on uskomaton avaruuseikkailu genren ylittämättömältä mestarilta. Kaikki tähän mennessä koettu ja kokeiltu ainakin vahvistaa ennako-odotuksia – aina Oculus Rift -koelentoja myöten.

Pian me olemme galaksin kansalaisia (Heinlein, 1957). Kiitos, David.

Tähtien kansa

Galaksin kansalaisen passeja on tosin luvassa toiseltakin suunnalta. Amerikkalainen Chris Roberts on ollut uransa mittaan mukana tekemässä monia pelejä. Origin Systemsin leivissä aloittanut kaveri teki muun muassa oivat seikkailut *Times of Lore* ja *Bad Blood*. PC-koneiden aikakauden myötä Roberts löysi sisimmältään runoilijan, muinaisnorjalaisen saagojen laulajan, jonka tarinoita täyttävät jumalten sijasta tähtienväliset eepokset ja sodat.

Roberts ja *Wing Commander* -pelit jäivät 90-luvulla videopelien historiaan, *Wing Commander* -elokuva ei

niinkään. Mies katseli taivasiin myöhemminkin tähdet silmissään, mutta jäi *Starlancer*- ja *Freelancer*-peleissä sivurooliin. Niiden jälkeen Roberts vietti lähes vuosikymmenen sähläämällä elokuvien ja muun turhan parissa.

Kunnes Kickstarter.

Star Citizen keräsi alkupottinsa Kickstarterista, mutta jatkoi joukkorahoituksen kasaamista sen jälkeen omien nettisivujensa kautta. Tulokset ovat jokseenkin käsittämättömiä. Ensimmäisen keräysvaiheen potti oli 6,2 miljoonaa dollaria, jonka jälkeen Roberts ilmoitti, että pystyy tekemään pelin valmiiksi ilman riskisijoittajien rahoja, jos joukkorahoitusta kertyy 23 miljoonaa.

Kirjoitushetkellä rahaa on kertynyt 63 miljoonaa.

Star Citizenin joukkorahoitushuuma muistuttaa tavallaan jo outoa kulttia, ja samalla joukkorahoituksen välitavoitteet ovat kadonneet aikaa sitten järkihorisontin tuolle puolen. Kun mitään järkevää ei enää keksitä, tehdään seu-

raavalla miltsillä planeetta, sitten kieli-tieteilijöiden luomia muukalaiskieliä ja FPS-räiskintäpeliosio.

Tekijätiimi on kuitenkin pysynyt hyvin työn touhussa. Vaikka peli laajenee kuin pullataikina, vaikuttaa kehittäjätiimi kehittäjien raporteista, behind the scenes -videoista ja omastakin kirjeenvaihdostani päätellen motivoituneelta. Voi suorastaan haistaa sen innon ja määrätietoisuuden, millä kaikkien aikojen avaruuspeliä tehdään.

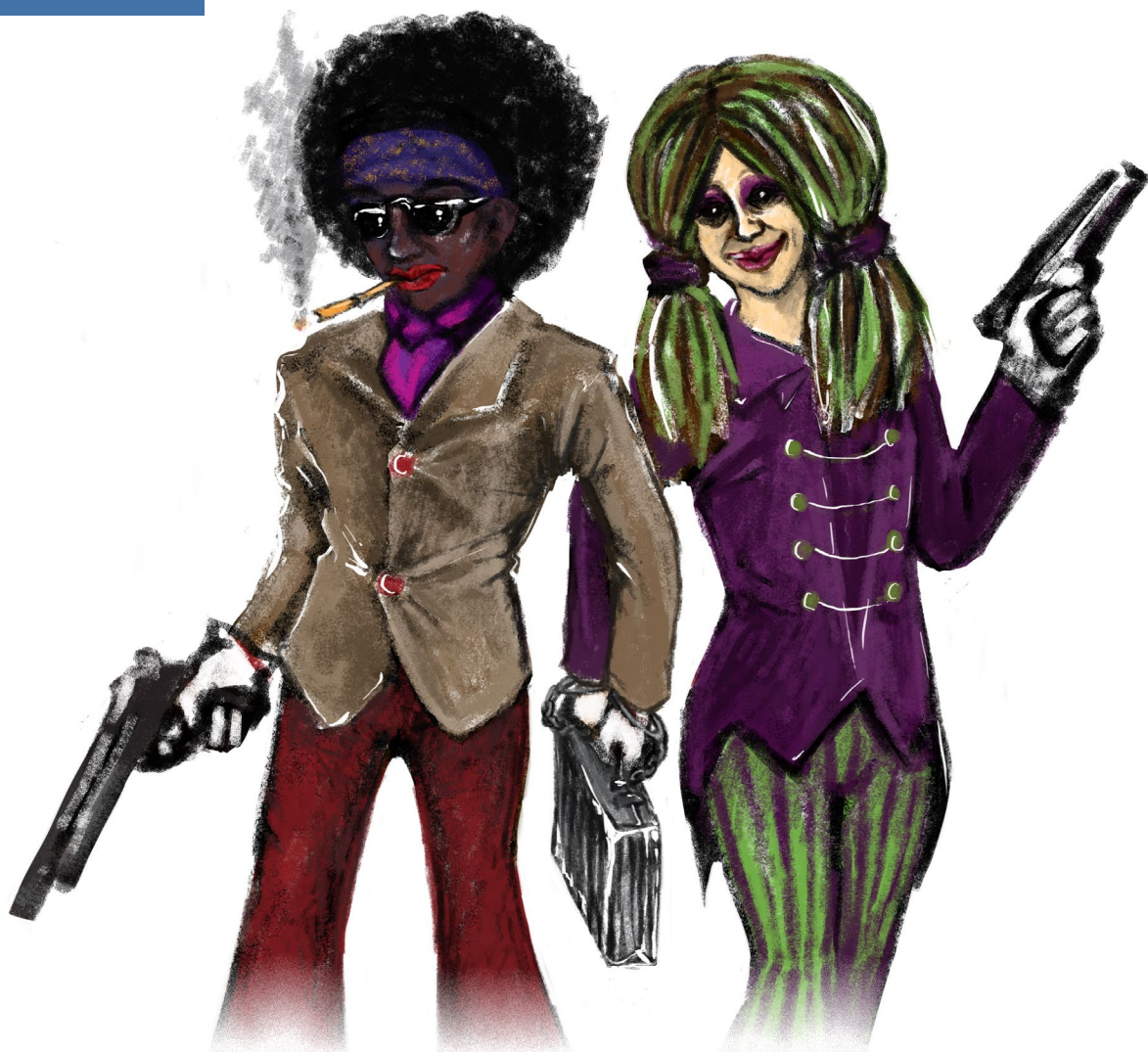
Tai pikemminkin kahta peliä. Isompi puoli on *Star Citizen*, "rajattoman immerstiivinen" avaruuseikkailu, jossa voi lentää ja puuhastella avaruudessa, tutkia planeettoja, vallata vihollisaluksia FPS-pelitilassa, käydä kauppaa ja ties mitä. Kaikki tämä universumissa, joka pyörii pelaajien tekemisten ympärillä, kuten *Elite* ja *Evekin*. Toinen puoli Staria on *Squadron 42*, joka on *Wing Commanderien* henkeen tehty eepinen yhden pelaajan tarinavetoisen offline-avaruustaistelusimulaatio.

Star Citizenin kehitys jatkuu yhä. Pelille ei ole annettu vielä varsinaista julkaisupäivää, mutta *Eliten* tapaan julkaisupäivä on vain yksi näennäinen välitavoite pitkän, vuosia tai jopa vuosikymmeniä jatkuvan matkan varrella.

Sillä tähdet ovat jälleen kirkkaat ja kutsuvat. 🌌



Star Citizenin joukkorahoitusmenestys innosti upottamaan avaruuseikkailun sisään myös FPS-räiskintäpelin.



Ainutlaatuinen agenttiseikkailu: COVERT ACTION

Eletään 1990-luvun alkua. Neuvostoliitto tiedustelukoneistoinen on vielä tolzillaan. Kansainväliset terroristijärjestöt suunnittelevat entistä rohkeampia iskuja. Huumekartellit ja palkkasotilaat kylvävät kauhua Etelä-Amerikassa. Yhdysvaltojen presidentti odottaa CIA:lta toimintaa, josta vastaa tuplanolla-erikoisagentti Max Remington.

Teksti: Kalle Viiri

Kuvat: Mitol Meerna, Kalle Viiri

Sid Meierin kehittämässä, vuonna 1990 julkaistussa Covert Actionissa pelaaja ohjastaa agentti Remingtonia tämän yrittäessä paljastaa rikollisten ja vihollisten salajuonet Yhdysvaltoja vastaan. Superagentti Remington suorittaa operaatiot lähes tulkoon yksin, aina puhelinkuuntelusta takaa-ajoihin ja tulitaisteluihin. CIA-agentilla ei ole kylmän sodan loppuvuosina paljoa ystäviä: kymmenet tiedustelupalvelut, terroristijärjestöt ja rikollisperheet juonivat niin Yhdysvaltojen kuin Remingtoninkin pään menoksi.

Peli alkaa hahmonluonnolla, jossa pelaaja saa valita Remingtonin koo-

dinimen, sukupuolen ja taidot. Taitovalintoja on neljä: taistelu, ajaminen, koodinpurku ja elektroniikka. Jokainen taito hyödyttää pelaajaa helpottamalla siihen liittyviä tiedustelutyön osa-alueita. Kun taidot on valittu, aloittaa Remington rikollisjahtinsa Washingtonista CIA:n johtajan opastamana.

Pelissä ratkotaan vihollisjärjestöjen salajuonia yksi kerrallaan. Juonet ovat satunnaisgeneroituja mutta jokseenkin kaavamaisia. Tyypillisesti salajuonessa on mukana kaksi tai kolme eri vihollisjärjestöä ja vähintään viisi vihollisagenttia, joista kullakin on oma roolinsa rikoksen suunnittelussa ja toteutuksessa.

Kunkin tapauksen aluksi pelaaja saa vaikeusasteesta riippuvan määrän johtolankoja hämäräveikkojen konnankoukkujen selvittelyyn. Johtolangat ovat pieniä vihjeitä, joista voi olla apua konnien etsinnässä: ne voivat antaa pelaajalle vihjeitä esimerkiksi siitä, millaista terroritekoa viholliset suunnittelevat, missä kaupungissa tapahtuu kummia tai mikä vihollisjärjestö on ollut erityisen aktiivinen.

Nuuskintaa nurkissa

Suurin osa tiedonhankinnasta jää pelaajan omalle vastuulle. Remington voi kierrellä kaupungista toiseen ja istua hotellin baarissa kuuntelemassa juo-

ruja kaupungin alamaailman menoista, mutta näin saa lähes varmasti vihollisjoukkioilta ei-toivottua huomiota osakseen. Harva rikollisnero seisoo tumput suorina, kun CIA-agentti nuuskii kaupungilla. Tuki-kohtien vartiointia tehostetaan, ja korkeammilla vaikeusasteilla viholliset voivat lähettää oman iskuryhmänsä yhyttämään Remingtonin, kun tämä liikkuu kaupungin syrjäkujilla.

Hienovaraisempi lähestymistapa tiedustelupuuhiin on toteutettu Microproselletyypilliseen tapaan minipeleillä. Elektroniikkaminipeli on logiikkaportteihin perustuva pulmahaaste, jossa pelaajan on johdettava sähköä maaliportteihin välttäen samalla hälytysportteja. Elektroniikkapeliä pelataan vihollisautojen jäljityksessä ja puhelimia kuunnellessa. Onnistunut ajoneuvon jäljitys voi paljastaa rikollisten piilopaikkoja, ja puhelin-kuuntelusta saa lisävihjeitä meneillään olevasta juonesta.

Kryptografiaa ja kaahailua

Pelaaja voi saada puhelin-kuuntelulla tai vihollisten papereita penkomalla haltuunsa salaviestejä. Koodatut viestit voi purkaa kryptografiaan perustuvalla älypelillä. Kaikki viestit on salattu yksinkertaisella korvaussalauksella. Pelaaja voi ratkoa salauksen itse tai päästää CIA:n "taajuusanalysitiemekoneen" salauksen kimppeeseen. Jälkimmäinen ratkaisu tekee tehtävästä helpomman mutta kestää peliajassa kauan ja antaa terroristeille lisää aikaa viedä juontaan eteenpäin.

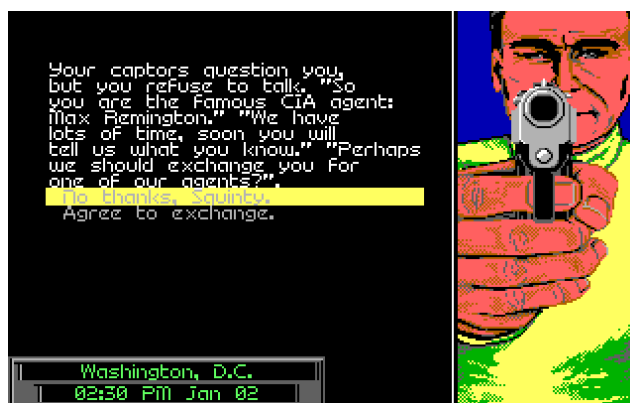
Takaa-ajominipelissä pääsee puolestaan elämään Bond-elokuvien takaa-ajokohtauksia. Jokaista takaa-ajoa varten saa valita kaksi neljästä autosta, joiden huippunopeus ja huomaamattomuus vaihtelevat. Joissakin autoissa on jäljityslaite, jolla voi seurata vihollis-



Remington voi piipahtaa seikkailuillaan myös Helsingissä.



KGB:n tukikohtaa väijymässä.



Jouduttuaan vihollisten vangiksi Remington voi joko suostua vanginvaihtoon tai yrittää paeta itse.

lisäautoa suoran näkökentän ulkopuolelle. Ajopelin tarkoituksena voi olla seurata vihollista vaivihkaa toiseen piilopaikkaan, pidättää autoa ohjastava agentti tai pötkiä vihollisia pakoon.

Yhden agentin armeija

Covert Actionin ydin on suoraa toimintaa: murtautumista vihollisten toimistoihin, piilopaikkoihin ja tukikohtiin. Pelaaja voi valita itselleen varusteet valikoimasta, johon kuuluu aseita ja vakoilutyökaluja kranaattipaketista kassakaapin murtosarjaan. Erityisen tärkeitä varusteita ovat todis-

teiden keräämisen mahdollistava kamera ja vartijoiden välttämiseksi auttava liikkeentunnistin.

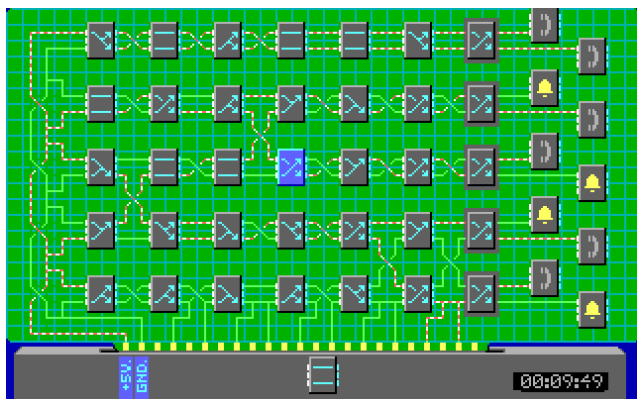
Varusteiden valinnan jälkeen siirrytään ylhäältä kuvattuun hiippailuosuuteen vihollisen rakennuksessa. Pelaaja voi rymytä läpi arkistokaappien ja työpöytien, valokuvaten itselleen roppakaupalla lisätietoja vihollisten juonesta. Murtotyökaluilla varustautunut pelaaja voi myös kämmiä itselleen salaisia dokumentteja kassakaapeista. Vartijoita välttämällä syöksymällä kalusteiden taakse piiloon tai nappaamalla eliminoidulta vartijalta yleen valeasun. Univormu suojaa vartijoiden huomiolta, kunhan Remington on näihin selin.

Remington voi turvautua myös tulitaisteluun, mutta jouduttuaan isompia joukkoja vastaan yksinäinen agentti jää herkästi alakynteen. Kranaateilla ja niistä viriteltävillä ansoilla voi tasata mahdollisuuksia, mutta vihollisetkin osaavat käyttää niitä. Äänekäs taistelu vetää helposti puoleensa lisää vihamielisiä hahmoja, jolloin on viisainta vetäytyä takavasemmalle piileskelemään. Liikaa osumia ottaessaan Remington menettää tajuntansa ja joutuu vapauttamaan itsensä joko vanginvaihdolla tai runsaasti aikaa vaativalla pakoyrityksellä.

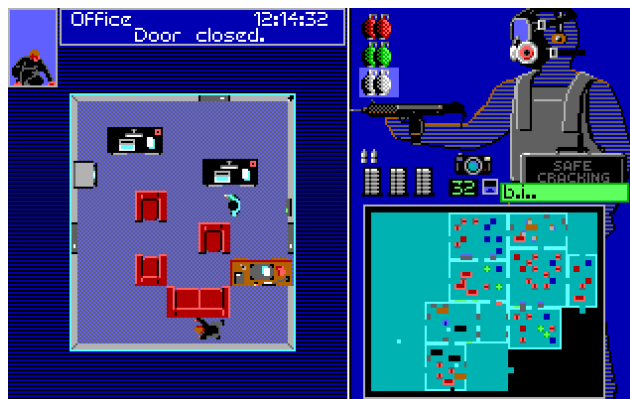
Useissa vihollistukikohdissa on paikalla vihollisagentti, joka saattaa olla silkka pelaajan kiusaksi asetettu hämäs tai juonessa tosissaan mukana oleva agentti. Pelaaja voi pidättää agentin ja tukikohdasta ulos päästyään kuulustella tätä. Jos todisteita on tarpeeksi, agentti pannaan lukkojen taakse. Kassakaapeista löytyy toisinaan erityisen vahvoja todisteita, joiden avulla agentti saadaan kääntymään kaksoisagentiksi CIA:n puolelle.

Oikeus voittaa

Agenttien pidättäminen on pelissä ensiarvoisen tärkeää, sillä se on ai-



Puhelinkuuntelussa vaaditaan älyä ja kärsivällisyyttä.



Kiinni saadut vihollisagentit heitetään surutta putkaan.

noa tapa pysäyttää terroristien juonet. Usein juonien onnistuminen riippuu useasta vihollisagentista. Esimerkiksi pommi-iskuun tarvitaan iso joukko erilaisia tavarantoimittajia, kohdetta kartoittava tiedustelija, pommiekspertti sekä erikoisagentti iskua suorittamaan. Yhdenkin agentin poistaminen ketjusta voi romahduttaa koko operaation ja saa muut agentit maan alle pakosalle.

Pelin voittaminen on siis helppoa, mutta pistelasku suosii pelaajia, jotka saavat mahdollisimman monta agenttia kiinni. Operaation kariutuessa piilosille lähteneitä agentteja ei voi enää napata, eli pelaajan on yleensä parasta estää rikollisten operaatio vasta mahdollisimman myöhäisessä vaiheessa.

Kaksoisagenteista on erityisen suurta hyötyä, sillä he jatkavat operaatiossa omaa osaansa ja syöttävät samalla tärkeitä tietoja pelaajalle. Näin pelaajan on helppo saada loput agenteista kiinni ja saavuttaa korkeat pisteet. Lisäaikaa voi ostaa myös varastamalla vihollisten tarvitsemia tarvikkeita, mikä hidastaa juonen edistymistä.

Missä seuraajat?

1990-luvun alun peliarvostelijat olivat suotuisia Covert Actionille. Pelilehti Computer Gaming World ylisti pelin kiinnostavia tehtäviä ja koukuttavia minipelejä. Lehti kuvaili peliä kuin perunalastuiksi: ensimmäisen jälkeen tekee mieli lisää.

Jos uskomme pelin nimekästä tekijää Sid Meieria, agenttiseikkailu ei kuitenkaan ollut täydellinen. Meier oli tyytymätön lopputulokseen ja totesi

pelin toiminnantäyteisen murtautumisminipelin vievän liikaa huomiota varsinaiselta mysteerinratkomiselta. Hiippailuminipeli on hyvä mutta katkaisee pelaajan keskittymisen varsinaiseen juoneen. Meier kiteytti oppimansa Covert Action -säännöksi: ”Ei liikaa pelejä samaan pakettiin.”

Minipelien tasapainotus kieltämättä ontuu. Murtokeikat riittävät hyvin pelissä pärjäämiseen, ja kaikki muu on jokseenkin vapaaehtoisia. Varsinkin haastava autoiluminipeli on täysin korvattavissa suoraviivaisella murtokeikalla tai jäljityspähkinällä. Computer Game World kritisoi arvostelussaan myös pelin valtavaa ohjekirjaa ja pelaajan vastuulla olevien valintojen määrää.

Pelin pisteytysmalli on ajoittain raskas. Täysien pisteiden saaminen riippuu osittain tuurista, sillä kaikissa vihollisagenttien piilopaikoissa ei ole kaksoisagentiksi käännättämisen mahdollistavia isoja kassakaappeja. Parhaiden tulosten saaminen menee usein tarkaksi ja puuduttavaksi odotteluksi ja viivyttelyksi.

Vioistaan huolimatta Covert Action jätti monet kaipaamaan lisää agenttitoimintaa. Microprosen pelituotanto jatkui esimerkiksi Civilizationin merkeissä, mutta jatko-osaa Covert Actionille ei ole saatu. Myöskään muut pelitalot eivät ole tarttuneet mahdollisuuden tehdä Covert Actionin kaltaista vakoilupeliä. Niinpä Meierin agenttiseikkailu on jäänyt lajinsa ainoaksi erikoisuudeksi. 🕵️

Minipelien kimara

Covert Actionissa pelaaminen on toteutettu usean minipelin kokonaisuutena. Koodinmurto, elektroniikka, ajaminen ja taistelu ovat kaikki omia pikkupelejä, jotka vaativat pelaajalta erilaisia taitoja. Minipeleissä pärjääminen palkitsee pelaajaa varsinaisessa seikkailussa.

Minipeleistä koostettu suurempi kokonaisuus koettiin 1980-luvun lopulla toimivaksi formaatiksi. Microprosen peleistä ennen Covert Actionia sitä noudattivat Sid Meier's Pirates ja Sword of the Samurai. Minipelisarjan etuna on, että pelistä saadaan helposti monipuolinen, sillä pelimekaniikat saattavat vaihtua luontaisesti sen mukaan, mitä pelissä kulloinkin tapahtuu.

Minipelisarjan huonona puolena on tasapainottamisen vaikeus. Usein jokin minipeleistä on selvästi muita vaikeampi tai palkinnot sen suorittamisesta ovat vähäisiä. Joskus taas yksi minipeli vaatii niin paljon aikaa, että toiset minipelit ja pelin kokonaiskuva hämärtyvät. Näin kävi Sid Meierin mukaan Covert Actionin murtautumispelein kanssa.

Covert Actionin jälkeen Meier totesi yhteen asiaan keskittyvän pelin paremmaksi kuin minipelikimaran, jotta pelaaja voisi keskittyä pelin päätarkoitukseen. Tätä oppia hän sovelsi seuraavana vuonna julkaistuu Civilization-jymymenestykseensä. Tässä legendaarisessa vuoropohjaisessa strategiapelissä ei ole yhden yhtä minipeliä.



Teksti: Ville Jouppi
Kuvat: Ville Jouppi
Toni Bratincevic
Henrik Erlandsson

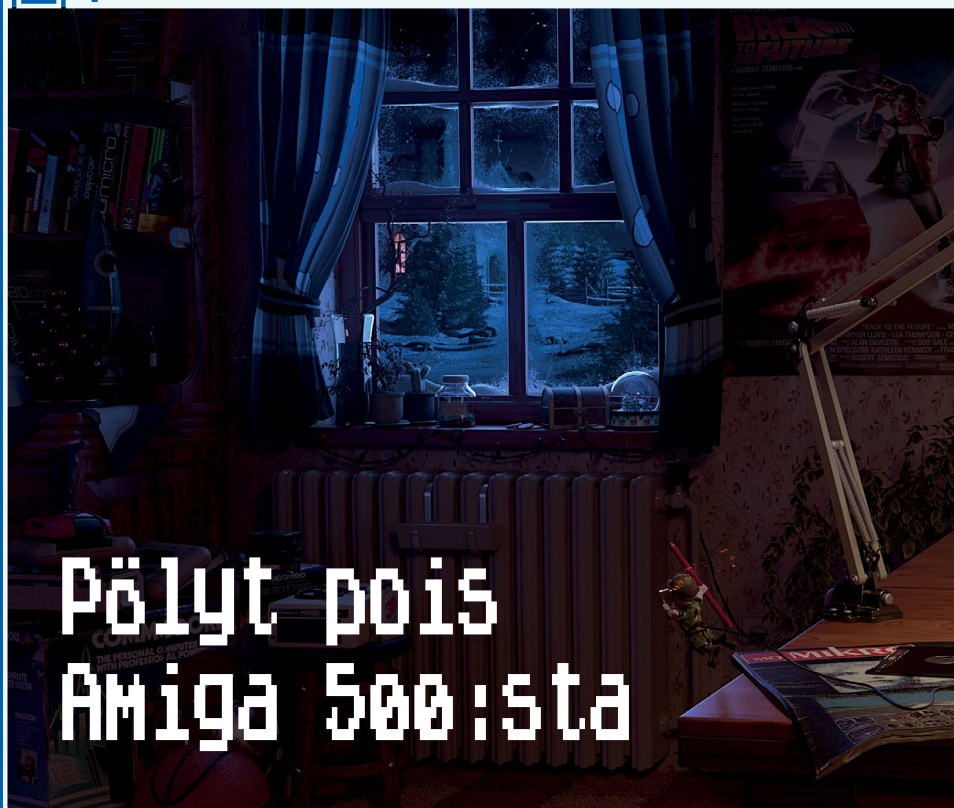


Paljastuiko
syyssiivouksen
yhteydessä kaapistasi
vanha kunnon Amiga 500?
Tuliko herkistyttyä,
kun muistelitte
illanistujaisissa vanhoja
hyviä aikoja – niitä,
jolloin koneet olivat
rautaa ja pelit vaikeita?
Jättikö Assemblyjen
Oldskool-compo tunteen,
että tekisit itse
paremman?

Oli koneesi sitten kaapissa pölyttynyt oma vanha työjuhtasi tai huutonetistä voitettu jonkun muun aiemmin rakastama laite, mielessäsi on varmasti paljon kysymyksiä. Tässä artikkelissa käydään läpi yleisimpiä vikatilanteita ja ongelmia, joita konevanhuksen kanssa saat-
taa olla odotettavissa.

Budjetin ei tarvitse olla suuren suuri, mutta jos tilillä on ylimääräistä rahaa, sitä voit toki kaataa tähänkin projektiin vaikka kuinka paljon. Kaikkein halvimmalla pääsee, jos jättää koneen laajentamatta ja ajaa ohjelmat levykkeiltä. Jos taas mielii tehdä yhtään hyötykäyttöä, korppuja nopeampi ja suurempi massamuisti on lähes välttämättömyys. Peruskoneen muisti tulee myös nopeasti täyteen, mikäli haaveissa on jotain suurempaa kuin pelkkä pelikorppujen syöttäminen asemaan.

System



Pölyt pois Amiga 500:sta

Kurkistus lahjahevosen suuhun

Aivan ensimmäiseksi löytä-
mäsi laite kannattaa avata ja luoda yleissilmäys sisuskaluihin. Koneen reunaa kiertävät ruuvit käännetään auki laitteen pohjasta, mutta levy-
aseman alla olevat kolme upotettua ruuvia kannattaa jättää paikalleen. Tämän jälkeen kansi irta-
a nosta-
malla. Näppäimistön yläreunan ta-
salla kuoren vasemmalla ja oikealla puolen on klipsejä, jotka saattavat olla tiukassa, mikäli konetta ei ole koskaan aiemmin avattu.

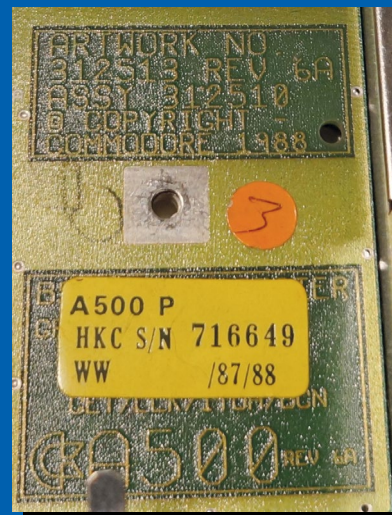
Kannen poistettuasi vastassa on useimmiten hyvinkin tylsä näky: näppäimistö, korppuasema ja iso häiriösuoja-
pelti. Irrota näppäimistön johto emolevyltä vetämällä suoraan ylöspäin ja pane samalla merkille, kummin päin se oli paikallaan. Muistivinkkinä toimii, että musta johdin kuuluu vasemmalle. Tämän jälkeen nosta näppäimistö syrjään. Pölyn voi kopistella näp-

päinten välistä pois, myös imuri toi-
mii hyvin kunhan imuteho on pieni.

Tämän jälkeen sitten RF-suoja-
pellin pariin. Pelti on kiinni neljäl-
lä ruuvilla (etuosa ja vasemman reunan Amigabus-laajennuspaikan
suoja-
pelti) ja joukolla pieniä pelti-
väkäsiä. Ota ruuvit auki ja taita vä-
käset suoraan, minkä jälkeen pelli-
n voi nostaa pois.

Tämän jälkeen eteesi aukeaa
koneen emolevy. Aivan ensimmäi-
seksi on syytä vilkaista korppuase-
man viereen piirikortin kulmaan ja
katsoa emolevyn revisio. Revisio 6

Revisio
6a-emolevyn
tunniste





ja sitä isommat ovat parempia, viitosen kanssa tulee vielä toimeen. Jos kulumasta ei löydy revisionumeroa lainkaan, emolevy on revisiota 3 ja sillä on lähinnä keräilyarvoa sen iän ja epävakauden vuoksi. Revisiot 4 ja 7 ovat hyvin harvinaisia ja niitä kannattaa tyrkyttää keräilijöille. Hinnat tuskin nousevat mitenkään kovin suuriksi, mutta veikkaanpa, että jäät diilissä voitolle.

Jotkut aivan loppupään A500:t ovat revisiota kahdeksan, eli ne ovat oikeastaan tyhmennettyjä A500+-koneita. Näistä voi tehdä täysiverisen A500+:n muutaman komponentin lisäyksellä. Jos alkoi kiinnostamaan, suosittelen tutki- maan kytkentäkaavioita ja internetistä löytyviä valokuvia eri emolevyistä erojen havaitsemiseksi.

Vahingon veräjälle

Tässä vaiheessa kannattaa luoda kriittinen silmäys kaikkiin

elektrolyyttikondensaattoreihin, eli niihin paputölkin näköisiin muovilla vuorattuihin kaksijalkaisiin komponentteihin, joita emolevyllä töröttää. Jos kondensaattorin alla on karstaa, tai juotokset sen läheltä ovat muuttuneet mattavihreiksi, on kondensaattori vuotanut ja se tulee vaihtaa. Vuotanut elektrolyytti pitää puhdistaa pois piirilevyä syövyttämästä. Isommissa kondensaattoreissa saattaa lisäksi yläpää pullistua kondensaattorin vikaantuessa.

Lopuksi kannattaa vilkaista, onko mahdollisessa lisämuistissa reaaliaikakelloa ja siinä akkua. Jos lisämuistina on Commodoren oma A501, akku on 100 prosentin varmuudella vuotanut sisälle. A501:n tunnistaa siitä, että se on koteloitu kokonaan isoon kiilan muotoiseen RF-peltiin, joka vaatii juotoskolvin auetakseen.

Jos akku on alkuperäinen, se tulee poistaa ennen virtojen kyt-

kemistä vuotojen ehkäisemiseksi. Pienet sivuleikkurit toimivat varsin hyvin poistohommissa; akun jalat vain leikataan poikki ja kaikkensa antanut vanha akku nostetaan pois. Mikäli vahinko on jo tapahtunut ja akku on vuotanut, elektrolyytti on puhdistettava piirikortilta lisävaurioiden ehkäisemiseksi. Makeuttamaton sitruunamehu toimii hyvin elektrolyytin neutralisoimiseen, loppupuhdistus on hyvä toteuttaa isopropyylialkoholilla.

Jos Amigaa on tarkoitus käyttää puhtaasti pelikoneena, akun voi aivan hyvin jättää pois. Reaaliaikakello ei tämän jälkeen enää pysy ajassa, mutta sillä ei liene kovin suurta merkitystä. Hyötykäyttäjän kannattaa laittaa uusi akku tilalle, sillä järkevistä tiedostojen aikaleimoista on usein iloa, ja kyllähän tuota kelloakin tulee toisinaan vilkaistua tietokoneen ruudulta. Akun voi myös korvata kolikkomallisella litiumparistolla ja diodilla. Diodi tulee paristopidikkeen plus-navan ja emolevyn väliin, päästösuunta kohden emolevyä. Jos et luota omaan elektroniikan tuntemukseesi, Amigakit myy myös valmista korvaussarjaa.

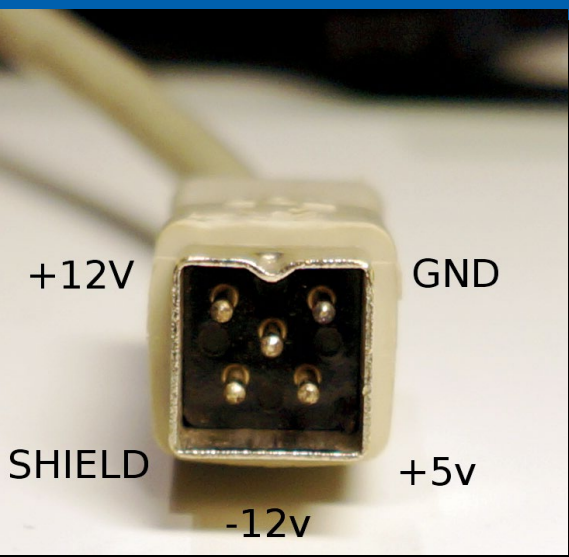
Kuvaa ruudulle

Amiga on siltä aikakaudelta, jolloin kotitietokoneet kytkettiin televisioon. Siispä PAL-koneen videosaanaali noudattelee enemmän tai vähemmän 288p-standardia. Tämä tarkoittaa 50 Hz:n pystyvirkestystaajuutta, vaakasunnassa taas välkytellään 15 kHz:n taajuudella. Normaalin lomitetun TV-signaalin sijaan perättäiset ruudut lähetetään "samoilla juovilla" kuin edelliset, eli ruudun päivystaajuus on aidosti 50 kuvaa sekunnissa.

Perinteinen 50 Hz:n PAL-kuva onnistuu nykyisiltäkin televisioilta. Progressiivisignaali saattaakin olla sitten hieman haasteellisempi juttu. Monet nykytelevisiot käsittelevät kuvaa hyvin paljon ja tekevät

pakotetun lomituksen poiston myös progressiivisignaaliille. Tällöin joka toinen ruudullinen jää näyttämättä ja esimerkiksi *Pinball Dreams*in kaltaiset sulavasti vierittävät pelit tökkivät rumasti.

15 kHz:n vaakataajuuden saa tuplattua niinsanotun **scandouble-rin** tai **flickerfixerin** hankkimalla. Tällainen on esimerkiksi Indivision, jota saa edelleen ostettua uutena. Vieritys ei kuitenkaan ole täysin sulavaa, ellei VGA-monitori tahdistu 50 Hz:n taajuuteen. 2000-luvun alkupuolen **ViewSonic**eista ja **Sony**istä löytyy tällaisia näyttöjä. Suosittelen tutustumaan laitteen käsikirjan lopusta löytyviin taajuustaulukoihin ennen ostopäätöksen tekoa.



Amiga 500:n virtaliitin

Micomsoftin **X-RGB mini (Frame-master)** on hintava mutta hyvä skaalain vanhojen tietokoneiden ja konsolien katseluun. Se saa kiinni kaikenlaisista progressiivisignaaleista ja ulostulona on moderni HDMI.

Kaikkein paras yhteensopivuus saavutetaan tietenkin vanhalla kuvaputkittelkarilla SCART-liittimen kautta. SCART-kaapeli on helppo löytää uutenakin. Myös vanhat aikalaisnäytöt kuten **Commodore 1084** tai **Philips CM8833** toimivat

taatusti ja luovat vieläpä mukavaa retrotunnelmaa.

Sähköä systeemiin

Jos virtalähde on kadonnut tai hajonnut, voit hankkia korvaavan tilalle tai vaihtaa itse matalajännitepuolen liittimen johonkin valmiiseen virtalähteeseen.

A500:n jännite- ja virtavaatimukset ovat seuraavat:

- * +5 VDC 3 A
- * +12 VDC 1 A
- * -12 VDC 0,1 A

Laajennetussa koneessa isompi ampeeriluku viiden voltin syötölle ei ole mitenkään liioiteltua. Kahta-toista voltia tarvitaan lähinnä kovalevyjen moottorien pyörittelyyn. Nykyisin useimmat käyttävät massamuisteina erilaisia flash-kortteja, jotka ovat pääosin viiden voltin laitteita. Negatiivinen 12 V on ääniä ja RS232:ta varten, sitä käytetään lähinnä referenssijännitteenä.

Vanhasta PC:n AT- tai ATX-virtalähteestä on suhteellisen helppo ottaa tarvittavat jännitteet. Wattien kanssa ei kannata mennä liiallisuuksiin. Amiga ei lopulta kuluta kovinkaan paljon virtaa, joten liian tehokas virtalähde ei välttämättä reguloitu kunnolla Amigan kuormalla. Lisäksi jossain vaiheessa PC-maailmassa 3,3 voltin jännitekisko muuttui viittä voltia merkittävämmäksi, joten vanhemmat noin 250-300 watin tehoiset ATX-virtalähteet ovat varmin valinta.

Amigan neliskanttista virtaliitintä on vaikea löytää uutena. Helpoimmalla pääset, jos voit kiertää johdon rikkinäisestä virtalähteestä. Käytettyjä virtalähteitä pyörii usein internetin kauppapaikoissa ja uusia virtalähteitä ja adaptereita löytyy toki myös jälleenmyyjien varastoista.

Eikö tokene?

Kytkeytyäsi näytön, virtalähteen ja hiiren pitäisi koneen herätä henkiin virtojen kytkenisen jälkeen. Jos näin ei käy, katso alta lisäohjeita.

Näppäinistö

Normaalisti Amigan käynnistytessä Caps Lock -ledi syttyy hetkeksi ja sitten sammuu. Toisinaan valo jää vilkkumaan ja näppäinistö ei tottele. Ensiaapuna tähän kannattaa mitata laitteen käyttöjännitteet. Joskus myös viallinen **CIA**-piiri on syyppä vilkkumiseen.

Näppäinistö yrittää viestittää virheen laatua ledin välkähdysten määrällä:

1. ROMin tarkastussumma väärä
2. RAM-testi epäonnistui
3. häiriö **Watchdog**-ajastimessa

Jos käyttöjännitteet ovat kohdallaan ja CIA:t on vaihdettu ristiin, eikä ongelma katoa, on näppäinistön ohjainpiiri entinen ja näppäinistö (tai sen kulmasta löytyvä ohjainkortti) on vaihdettava.

Ruutu jänähää jonkin väriseksi

Amiga osaa myös ilmoittaa käynnistystestien yhteydessä tapahtuvia ongelmia ruudulla näkyvin värikoodein.

Useimmiten vastaan tulevat värit ovat punainen, keltainen ja vihreä. Seuraavat selitykset ovat lähinnä Amiga 500:n näkökulmasta.

- * **Punainen** - Kickstart ROM:n tarkistussumma on virheellinen, tarkista CPU:n ja Kickstart-ROM-piirin jalat ja kannat hapettumien varalta. Puhdista tarvittaessa.
- * **Keltainen** - CPU ajoi koodinpätkän joka tuotti poikkeuksen kesken bootin,



useimmiten tämä johtuu tietokoneeseen kytketystä vialliseen laajennuksesta. Poista kaikki laajennukset ja yritä konetta käyntiin uudestaan.

* **Vihreä**-Virhe chip-muistissa. Chip-muisti on Amigassa oleva muistialue, jota grafiikka- ja äänipiirit voivat lukea. Vihreä ruutu on usein oire Agnus-piirin jalkojen tai kannan kosketushäiriöistä. Toki jos jokin emolevyn tai pohjaluukkulaajennuksen muistipiireistä on viallinen, tämä vika saattaa ilmetä.

Vihreän ruudun ongelmaa tutkittaessa varoituksen sana: **PLCC**-piirin poistamiseksi on aina käytettävä siihen tarkoitettua ulosvetäjää. Kahdella ruuvimeisselillä vääntäessä tuloksena on murtunut PLCC-kanta, sillä nämä vanhat piirit istuvat tiukassa ja kannat ovat hapertuneet vuosien mittaan.



PLCC-vetäjä

Korppuasena ei lue

Maailmassa on paljon enemmän toimivia A500:n emolevyjä kuin niihin sopivia ehjiä korppuasenia.

Ihan ensimmäiseksi kannattaa kokeilla puhdistuslevykettä, jossa on pari tippaa isopropyylialkoholia. Jos ei puhdistuslevykettä ole, voi lukupäät puhdistaa myös isopropyylialkoholiin kastetulla pumpulipuikolla. Tämä vaatii tarkkuutta ja jänäkkä, mutta hellää otetta, etteivät lukupäät mene

vinoon.

Aseman ollessa auki kannattaa puhdistaa myös lukupääkelkan liukukisko pölystä ja jäykistyneestä rasvasta. Liukukisko on syytä voidella jollain notkealla öljyllä, esimerkiksi ompelukoneöljyllä. Öljyä tulee käyttää vain vähän, ettei se sotke joka paikkaa. **WD-40** ja **CRC 5-56** pidetään sitten kukaan kaikesta muusta kuin ruostuneista muttereista, sillä ne eivät sovellu pitkäaikaiseen voiteluun.

Tietyissä **Panasonicin** asenissa on vuotavia pintaliitoskondensaattoreita. Näitä on saatu herätettyä henkiin vaihtamalla vialliset kondensaattorit. Aika usein aseman etupuoella vauhtipyörän vierestä löytyvä konkka on ollut vaihtokunnossa.

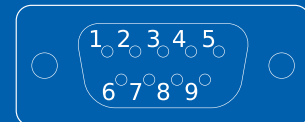
Chinonin asenissa yleisin ongelma on aseman etureunasta löytyvien kirjoitussuoja- ja levytunnistuskytinten pätkiminen. Näihin voi kokeilla tiputtaa hieman isopropyylialkoholia ja painella kytkintä sormella hetken aikaa. Paras lopputulos saavutetaan purkamalla kytkin ja puhdistamalla sen sisältä löytyvät kontaktit huolellisesti, mutta en voi suositella tätä lähestymistapaa, mikäli sinussa ei ole kellosepän vikaa. Pienet osat ja jouset katoavat helposti ja kytkin on hankala kasata uudelleen.

Ehjiä korppuasenia löytyy tois-taiseksi kuitenkin vielä kohtuullisen hyvin internetin kauppapaikoista. Yksi vaihtoehto on myös etsiä internetistä ohjeita PC-asenien muuttamiseksi Amigakäyttöön.

Hiiri mykkä, joystickista ei iloa

Viallisen hiiren tapauksessa aivan ensimmäisenä on syytä lainata toinen hiiri jostain ja kokeilla sillä. Mikäli hiiressä ei ole vikaa, tulee mitata

koneen takaa hiiriportin käyttöjännite. Löytyykö **5 V** nastasta ?? Jos ei löydy, vika saattaa olla jossain hiiriportin lähellä olevista passivikomponenteista. Kytkentäkaaviot löytyvät internetistä, ei muuta kuin vetoja seuraamaan ja mittailemaan, minkä komponentin kohdalla jännite ei enää pääse läpi.



Hiiriportin liitin

Jos käyttöjännite on lähellä viittä voltia, voit kokeilla vaihtaa **Denise**-piirin toimivaksi tiedettyyn. Deniseä yleisempi ongelma on kuitenkin palanut multiplekseripiiri **74LS157** paikassa **U15**. Valitettavasti U15 ei ole kannalla, joten sen vaihtaminen pelkästään kokeen vuoksi on vähän työlästä. Osa on kuitenkin halpa, joten suurta tappiota ei tule, vaikka sen vaihtaisi huvikseenkin. Muista laittaa uusi piiri kannalle.

Joystickin suunnat eivät vaadi +5 V:n käyttöjännitettä. Jos suunnat eivät toimi, todennäköisin vikapaikka on jälleen piiri U15.

Mikäli hiiren vasen nappi tai joystickin tulitusnappula ei toimi, syyllinen on CIA-piiri paikassa U7. Molempien peliohjainporttien ensimmäiset tulitusnappulat ovatkin sitten ainoa peliohjainportteihin liittyvä toiminto Amigan CIA:ssa, vaikka tuon piirin vaihtoa lähes kaikkiin joystick-ongelmiin usein lääkkeeksi tarjotaankin.

Oikea ja keskimäinen hiirennappi sekä joystick-portin toinen ja kolmas tulitusnappula luetaan Paulan potentiometrissäntulosta. Jos näiden kanssa on ongelmia, syyllinen on suurella todennäköisyydellä Paula-piiri paikassa U3.

Karkasiko hiiri?

Monista vaatekaapin uumenista löytyneistä koneista saattaa hiiri olla karkuteillä. Ehkäpä edullisin vaihtoehto tähän ongelmaan on hankkia joku lukuisista kaupallisista tai harrastevoimin toteutetuista PS/2-sovittimista. Useat USB-hiiretkin osaavat vielä jutella PS/2-protokollaa passiivisen adapterin läpi, joten saat moderninkin optisen hiiren toimimaan Amigassa niin halutessasi.

Uusia Amiga-yhteensopivia hiiriä löytyy vielä jälleenmyyjien hyllyistä ja internetin kauppapaikoissa on hyvä valikoima erilaisia käytettyjä.

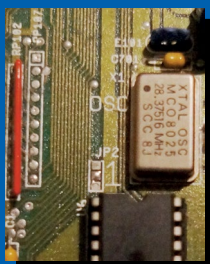
Megan chip? Grillimaustettuja vai?

Normaalissa A500:ssa on 512 kilotavua chip-muistia emolevyllä ja useimmiten 512 kilotavua hidasta fast-muistia pohjaluukussa. Chip-muisti on sekä erikoispiirien että suorittimen käytössä, fast-muisti taas vain suorittimelle varattua.

Yhdeksänkymmentäluvulla kaikilla itseään kunnioittavilla amigisteilla oli luonnollisesti laite muutettu osoittamaan kokonainen megatavu chip-muistia. Suuret Protracker-moduulit mahtuivat muistiin leikiten ja yhtäaikaista screenejä oli auki lukuisia! Hyötykäyttäjän kannattaa tehdä tämä muutos laitteeseensa edelleen, sillä chip-muistia ei ole koskaan liikaa. Korppupelajaajalle tämä muutos on turha, sillä valtaosa A500:lla pyörivistä peleistä tulee toimeen 512 kilotavun chipillä. Jotkut vaativat lisäksi 512 kt fastia.

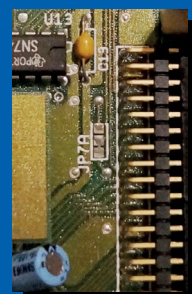
1Mt Chip-muistia

Jos aiot tehdä muutoksen, on syytä vilkaista ensimmäiseksi Agnus-piirin mallinumeroa. Piirin tulee olla mallia 8372, jotta se pystyisi osoittamaan täydet 1 Mt chip-muistia. Koneen pohjaluukkuun asennettavan 512 kt:n lisämuistin täytyy myös olla paikallaan.



Itse muutos on helppo. Agnuksen ylin osoitenasta vaihdetaan osoittamaan osoiteväylän osoitteeseen A19 ja GARYlle tuleva EXRAM-signaali katkaistaan, ettei sama muisti näy kahdessa muistialueessa.

Osoiteväylämuutos tehdään muuttamalla siltausta JP2 ja EXRAM-signaalin voi kytkeä pois Rev6-emolevyissä katkaisemalla siltauksen JP7A keskimmäisen ja alimmaisen nastan välisen vedon. Rev5-emolevyn tapauksessa GARYn voi nostaa kannaltaan ja taivuttaa EXRAM-jalan ulospäin, tai jos lisämuisti on tarvikemallinen, voit kääntää siinä olevan "virtakytkimen" off-asentoon.



JP2:n keskimmäisen ja etummaisen nastan ollessa yhteydessä chip-muistia näkyy vain 512 kt, keskimmäinen ja takimmainen luonnollisestikin sitten toteuttaa 1 Mt chippiä. 1 Mt:n tilassa EXRAM täytyy katkaista, sillä lisämuisti otetaan osaksi chip-muistia. Puolen megan tilassa voit valita itse, haluatko lisämuistin näkyviin vai et.

Siltauksen JP2-asento muutetaan yksinkertaisesti katkaisemalla nykyinen veto piirilevyltä siltauksen keskimmäisen ja etummaisen täplän väliltä ja juottamalla toiset kaksi täplää yhteen. Pysyvän muutoksen sijaan suosittelen asentamaan DPDT-kytkimen muuttamaan JP2:n asentoa ja EXRAM-signaalin tilaa yhtäaikaaisesti. Joystick-porttien vieressä on sille hyvin tilaa koneen takapaneelissa.



Punainen: leikkaa, Vihreä: juota

Revision 5 -emolevyissä tulee muutoksen yhteydessä useimmiten päivittää myös Agnus-piiri uudempaan. Tässä yhteydessä täytyy muistaa eristää Agnuksen PAL/NTSC-nasta 41 piirikannasta palasella teippiä, sillä muuten laite käynnistyy aina NTSC-tilaan. Tässäkin yhteydessä on syytä toistaa aiemmin antamani varoitus: käytä aina PLCC-ulosvetäjä.

Tämän helpon 1 Mt:n chip-muutoksen lisäksi on olemassa Agnus-kantaan painettavia laajennuksia, joilla saat chip-muistin laajennettua kahteen megatavuun saakka. Kannattaa pitää silmät auki internetin kauppapaikoilla, joskin harvinaisuutensa vuoksi näistä laajennuksista pyydetään aika kovaa hintaa.

Seikkaperäisemmät ohjeet muutoksen tekoon löytyvät internetistä kohtuullisella vaivalla. Mainittakoon myös, että jos pohjaluukussa ei ole lisämuistia ja sinulla on Rev6-emolevy, voit juottaa DRAM-piirit ja niitä varten tarvittavat suotokondensaattorit suoraan emolevyn tyhjiin paikkoihin. Suosittelen tutkimaan kytkentäkaavioita, jos lähdet tälle tielle. Lisäksi huomioitavaa on, että emolevyn tyhjä muistipaikat ovat pohjalaajennuksen kanssa päällekkäisiä, eli jos emolevyllä on jo täysi negatavu, ei pohjaluukkuun ole syytä kytkeä enää lisämuistia.

Sähkö- vai polkustartti?

Amigan **Kickstart** sisältää noin puolet käyttöjärjestelmästä. PC-maailmasta tulevat mieltävät sen helposti **BIOSin** korvikkeeksi, mutta ne eivät ole suoraan verrannollisia. Kun Amiga boottaa ja levykkeensyöttöhopute on ruudussa, ovat moniajo ja **AmigaOS** jo käynnissä. Kun BIOS boottaa levykkeensyöttöhoputteeseen, ei PC:ssä ole vielä käyttöjärjestelmää ajossa, vaan ainoastaan varhaisen tason käynnistysrutiini, joka osaa ladata yhden sektorin levyn alusta.

Korppupelaajan ei tarvitse miettiä tätä asiaa lainkaan, sillä **1.2** tai **1.3** riittää aivan hyvin ja tarjoaa parhaan yhteensopivuuden. Tehokäyttäjän valinta on viimeisin julkaistu versio, eli **3.1 (rev 40.63)**. Todella moni hyötyohjelma vaatii vähintään versiota **2.04 (rev 37.175)**, mutta suoraan versioon **3.1** hyppäämisestä ei ole mitään haittaa. Pikemminkin se on hieman yhteensopivampi vanhojen pelien kanssa kuin **2.04**.

Rev5-emolevyissä ja sitä vanhemmissa on virheellinen kytkentä Kickstart-kannassa. Ylin osoitelinja on väärässä nastassa. Kickstart **3.1** tulee käytännössä aina **EPROM**-piirillä, jolloin on tehtävä seuraavat muutokset: Piiristä käännetään jalka **31** ulos, ettei se osu kantaan.

Piirin jalka **1** liitetään kannan nastaan **31** hyppylangalla. Ulos käännetty piirin jalka **31** yhdistetään jalkaan **21**. Ensimmäinen hyppylanka korjaa virheellisen osoitelinjan, toinen hyppylanka varmistaa, että piiri toimii **16** bitin tilassa. Rev6-emolevystä eteenpäin näitä muutoksia ei tarvita.

Massamuistikausi alkuun

Korppupelien ystävien ei juuri tarvitse massamuisteista välittää, sillä peli latautuu, kun lyö levyn sisään. Tehokäyttäjälle jonkinasteinen kovalevy tai flash-kortti on



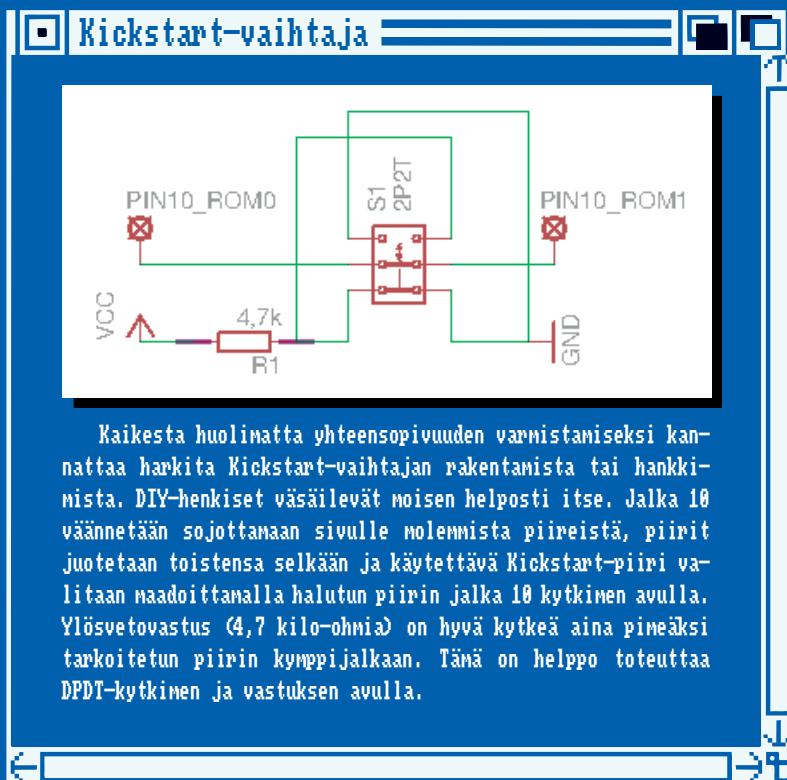
Gotek-korppuemulaattori

vanhatkin pelit saadaan toimimaan laajennetussa Amigassa. Toimiakseen WHDLoad vaatii käytännössä ainakin pari megatavua muistia ja kiintolevyn.

Nykyaikainen vaihtoehto on myös levyaseman korvaaminen emulaattorilla. Ensimmäinen ja halvempi vaihtoehto on Amiga-yhteensopivaksi uudelleen ohjelmoitu **Gotek**-korppuemulaattori. Tällä tavoin saat levykeimagnet USB-tikulle, eikä isoa kassaa levykkeitä tarvitse säilöä tai kirjoitella. Gotekeja saa PC-yhteensopivana eBays-tä ja valmiiksi Amigakäyttöön muokattuna Amigaan erikoistuneilta

kauppafoorumeilta. **ADF-tiedostot** kopioidaan USB-tikulle ja Gotek käynnistää koneen valikkoon, josta saa säätää, mikä muistipaikka lataa minkäkin ADF-tiedoston. Levykettä vaihdetaan emulaattorin kyljessä olevilla napeilla.

Toinen korppuemulaattori on **HxC Floppy Emulator**, joka käyttää SD-kortteja. HxC oli alunperin tar-



välttämättömyys. Pelaajienkin kannattaa kuitenkin harkita laitteiston laajentamista, sillä WHDLoad lisää käyttömukavuutta huomattavasti.

WHDLoad on yhteensopivuuskehys, jonka avulla alun perin korpulta ladattavat pelit muunnetaan lennossa kovalevyltä toimiviksi. Samalla WHDLoad korjaa peleistä yhteensopivuusongelmia, jolloin

koitettu itse rakennettavaksi, mutta valmiiksi kasattujakin on saatavilla. HxC-käyttäjällä on Goteckista poiketen yksi lisätehtävä, nimittäin ADF-tiedostot täytyy muuntaa HFE-muotoon ennen SD-kortille kopioimista. HxC:stä on saatavilla myös LCD-näytöllä varustettu versio, jolloin levyn valinta tapahtuu laitteen ruudulla olevia tekstejä seuraten.

Ne kovemmat levyt

Vaan korput sikseen, sillä tosimiehillä (ja -naisilla) on kovalevy. Suurin osa A500:aan tarkoitetuista levyohjaimista kytketään vasemmassa kyljessä olevaan **AmigaBUS**-väylään. Tuohon aikaan SCSI sattui olemaan paras tapa liittää massamuisteja kotitietokoneisiin, joten valtaosa A500:n levyohjaimista on nimenomaan SCSI-ohjaimia.



A500:n parhaat SCSI-ohjaimet ovat **GVP:n HD8+** ja Commodoren oma **A590**, sillä niissä on **DMA**-tuki. Ne siis osaavat siirtää tietoa suoraan koneen muistiin ilman prosessorin kuluttamista. Loput ovat keskenään aika lailla samalla viivalla ja liikuttelevat bittejä prosessorin voimalla.

Mikäli valintasi osuu johonkin muuhun kuin GVP:hen tai A590:een, varmista, että valitsemasi ohjain tukee **autoboottausta** ja **RDB**-par-

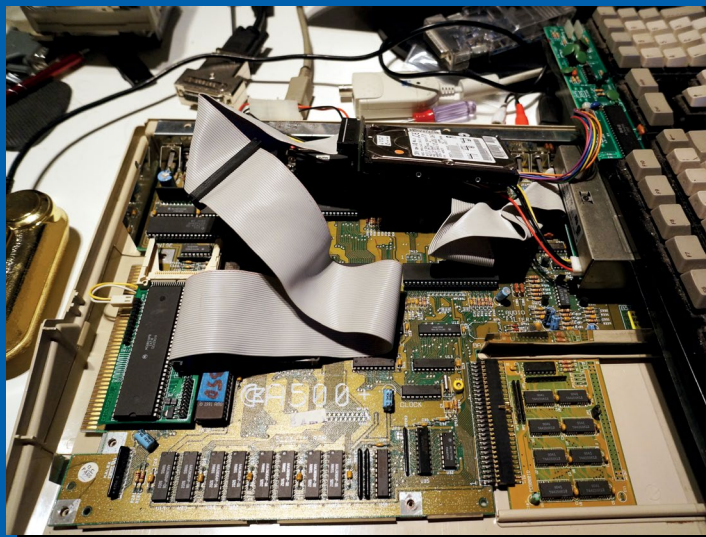
titiotaulua. Ilman autoboottia joudut kärvistelemään boottikorpujen ja RAM-levyjen kanssa, ja jos ohjain ei ole RDB-yhteensopiva, et voi käyttää kuin laitteen valmistajan omaa osiointityökalua. Se taas luultavasti tykkää huonoa nykyaikaisemmista tiedostojärjestelmistä ja gigatavuokan kiintolevyistä. Vanhoja levyohjaimia näkee vähän väliä myynnissä internetin kauppa- paikoilla, mutta hintoja kannattaa hieman seuraila, sillä hajontaa on jonkin verran.

Viime aikoina myös suomalainen **Mika Leinonen** ja kanadalainen **Kipper2k** ovat tuoneet markkinoille A500:n CPU-kantaan tulevia laajennuksia, joissa on **A600**-yhteensopiva **IDE**-väylä. Kickstart 3.1 sisältää ajurin tuolle IDE-väylälle, joten nuo kortit ovat varteenotettava vaihtoehto.

Kovalevyn tiedostojärjestelmäksi suosittelen pelkästään **PFS3 AIO**:ta, sillä se on yksiselitteinen asennus, tukee isoja kovalevyjä ja toimii lähes kaikilla prosessoreilla ja Kickstartin versioilla. Mikäli käytät Kickstartin omaa IDE-ajuria, kovalevyn maksimikoko PFS3 AIO:ta käyttäen on 8 gigatavua.

Turbokortit: 030:n puhaltimella Amiga jättäsen

Turbokortti eli kiihdytinkortti korvaa Amigan suorittimen nopeammalla versiolla. Artikkelin edetessä on jo muodostunut tietyn-



Mika Leinonen IDE

lainen kuvio, joka pitää paikkansa tässäkin: korppupelaajan kannattaa yhteensopivuuden säilyttämiseksi pitäytyä niistä erossa.

Tehokäyttäjä kuitenkin tarvitsee kaiken lisävauhdin, mitä voi saada. WHDLoad-käytössä turbosta on myös paljon iloa, sillä osa WHDLoad-asennusohjelmista vaatii nopeamman prosessorin ja muutamia megatavuja muistia.

Turboja on kahdenlaisia: **68000**-prosessorin kantaan liitettäviä sekä vasemman kyljen AmigaBUS-liitäntään kytkettäviä. Näistä suositeltavimpia ovat AmigaBUS-liitäntää käyttävät. 68000-kantaan tulevien turbojen ongelma on lähinnä muisti. Joko muistia saa vain pari hassua megatavua tai se ei autokonfiguroidu, jolloin osa käyttöjärjestelmästä saattaa latautua käynnistyksen yhteydessä hitaaseen chip-muistiin. Lisäksi AmigaBUSiin kytkettävät DMA-kykyiset SCSI-ohjaimet eivät pysty tekemään DMA-siirtoa sisäisen turbon 32-bittiseen muistiin.

GVP:n AmigaBUS-liitäntään kytkettävä **A530** on harvinainen, mutta viimeistellyin A500-turbo. Nopea **030**, 32-bittinen muisti ja toimiva DMA:ta käyttävä SCSI-ohjain tekevät kokoonpanosta monissa käyttötilanteissa jopa Amiga 3000:a nopeamman. **Supran SupraTur-**



bo28 taas on hitaimmasta päästä A500-turboja, mutta sen sisältämä välimuisti nopeuttaa 28 MHz:n prosessoria hieman. SupraTurbo vaatii lisäksi aitoa fast-muistia, että siitä saa kaiken irti.

Uusista laajennuksista **Individual Computersin ACA500** on melkoisen mielenkiintoinen paketti kevyeen tehokäyttöön tai WHDLoadin pyörytykseen. Laite sisältää 14 MHz:n 68000:n ja kaksi **CompactFlash**-paikkaa, joista toinen on tarkoitettu Amiga-formaattia käyttäville kortteille, toinen PC:n **FAT**-formaatile. Lisänä on vielä paikka A1200:n turbokortteille ja monia muita ominaisuuksia. Hintakin on tätä kirjoitettaessa kohtuullinen ja turboja on jälleenmyyjien varastoissa.

Mikäli laajennusmotivaatio on ainoastaan WHDLoadin ansiota, saattaa pelkkä lisämuisti riittää. Niitä on AmigaBUS-väylään liitettäviä ja prosessorikantaan painettavia. Tässä tilanteessa kannattaa harkita Kipper2k:n laajennuksia, sillä niissä on prosessorikantaan kytkettävällä kortilla lisämuisti ja Kickstart 3.1:n kanssa yhteensopiva CompactFlash-paikka.

Pelkillä lisämuistilla ja kovalevyllä ajettaessa WHDLoad hyöttyy myös prosessorin päivittämisestä **68010**-malliseksi. 68010:n siirrettävä vektorirekisteri mahdollistaa peleistä poistumisen ilman koneen resetointia. Prosessoripäivitys on yksinkertainen, sillä 68010 sopii suoraan vanhan CPU:n tilalle.

AmigaBUS-väylään tai CPU-kantaan tuleva lisämuisti on aitoa 16-bittistä fast-muistia, eli prosessorin ei tarvitse odotella erikoispiirin DMA:ta kun ohjelmakoodi pyöriä aidossa fastissa. Jo pelkästään tämä tuo noin kolmen prosentin tehonlisäyksen verrattuna pelkillä chip-muistilla tai pohjaluukkulaajennuksella varustettuun koneeseen.



SupraDrive 500XP + SupraRam 500RX 8MB

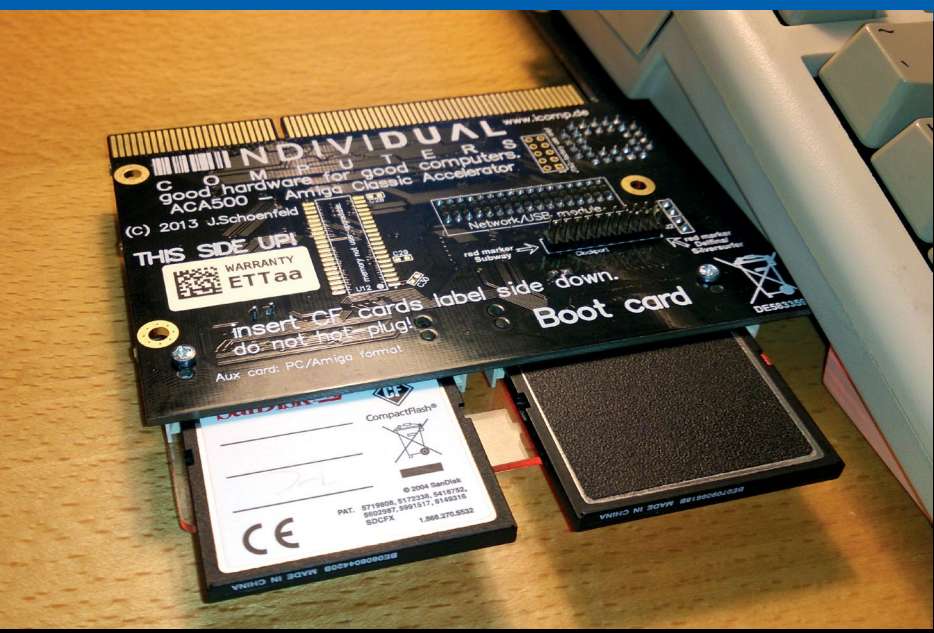
Tiedostojen siirto koneelle

Kaiken tuon raudan tarkastelun, virittelyn ja laajentelun jälkeen lienee aiheellista paneutua myös ohjelmistopuoleen. Suuren vaivan näkemisen jälkeen pelkkä Workbench-hoputteen katseleminen ei kovin kauaa jaksaa vetää puoleensa.

Uuden amiganomistajan suurin pulma onkin itse asiassa tämä tiedonsiirron muna-kana-ongelma. Amiga ei boottaa PC:n levyiltä ja PC ei osaa kirjoittaa Amiga-yhteensopivia levykkeitä ilman lisälaitteiston hankintaa. Puhumattakaan siitä, että harvassa modernissa PC:ssä edes on minkäänlaista levykeasemaa.

Gotek- ja HxC-käyttäjät pääsevät helpolla, sillä levyjen näköistiedostot vain kopioidaan kortille tai USB-tikulle ja siinä se. ACA500-käyttäjä voi käyttää toista korttipaikkaa FAT-formaatissa olevan CF-kortin lukemiseen.

Jos käytössä on pelkkä korppuasema, käytännössä edullisin vaihtoehto on **RS232**-väylä. RS232-siirtoihin suosittelen kaupallista *Amiga Exploreria* tai yhteisön kasaamaa



ACA500





Monipuolisista ominaisuuksistaan huolimatta Amiga oli monelle ennen kaikkea pelikone. Ja kelpasihan sillä pelata, kun pelit olivat näinkin laadukkaita.

Teksti ja kuvat: Mikko Heinonen

Miksen vain emuloisi?

Kaikki nämä pelit ovat tietenkin koettavissa myös emulaation kautta. *WinUAE* ja sen muille käyttöjärjestelmille tehdyt johdannaiset mahdollistavat varsin autenttisen kokemuksen vähäisellä säätöavulla. Satunnaiseen kokeiluun ne sopivatkin aivan hyvin, etenkin jos sattuu löytämään hyvät peliohjaimet tai *Atari*-tyyppisille ilotikuille USB-sovittimen, jossa ei ole liiallista syöttöviivettä.

Tarkkakin emulaattori kuitenkin aina vain matkii alkuperäistä konetta ja joutuu useimmiten oikomaan mutkia. Taustalla pyörivä käyttöjärjestelmä ja emulaatorirajapinta aiheuttavat hyvin usein erilaisia ajoitusongelmia, eikä etenkään ruudun vieritys ole sulavaa. Peli toimii vain aidolla laitteistolla täsmälleen siten kuin sen on tarkoitus. Tosissaan pelaavalle aito rauta on usein se oikea vaihtoehto.

Lataa ensin WHDLoad

Yksi ohjelmisto on ylitse muiden, kun haluat nauttia Amiga-peleistä aidolla laitteistolla ja helpottaa elämääsi muutenkin. Saksalainen *Bert Jahn* apulaisineen on jo liki 20 vuoden ajan kehittänyt WHDLoadia. Se on rajapinta, joka mahdollistaa alunperin levykkeillä toimitettujen pelien asentamisen kiintolevyille. Tämän lisäksi *WHDLoad* osaa päivittää vanhoja pelejä niin, että ne toimivat myös uusilla ja laajennetuilla Amigoilla.

Käytännön minimivaatimus WHDLoadin käyttöön on kiintolevy (tai nykyisin useimmiten CF-kortti) ja pari megatavua muistia. WHDLoadia voi kokeilla ilmaiseksi, ja sen rekisteröinti maksaa nimelliset 20 euroa. Ohjelmiston voi ladata osoitteesta www.whdload.de.

Kun Amiga 500 on aiemman jutun ohjeilla tuunattu huippukuntoon, on aika aloittaa parhaiden pelien fiilistely. Jokaisella on luonnollisesti omat suosikkinsa lapsuuden tai nuoruuden pelisessioista, mutta muutamat nimikkeet nousevat esiin yhä uudelleen tiedustellessa ihmisten lempipelejä.

Tätä juttua varten tein pienen kyselykierroksen Facebookin kansainvälisessä Commodore Amiga -ryhmässä ja tutustuin *Lemon Amiga* -sivuston käyttäjäarviointiin. Näiden lähteiden sekä oman näppituntumani perusteella valitsin kymmenen peliä, jotka ainakin kannattaa kokea Amigalla. Eri mieltä saa ja pitääkin olla – vaikkapa Skrollin keskustelufoorumilla!

Defender of the Crown

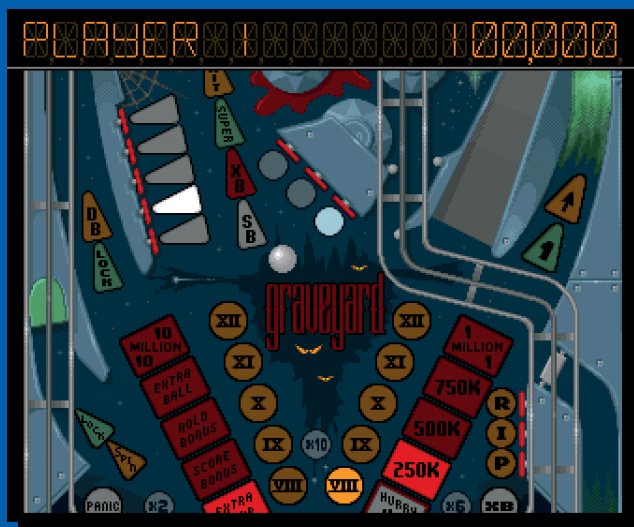


Listalla voisi olla mikä tahansa **Cinenavaren** "elokuva-peleistä", mutta valitaan nyt se kenties tunnetuin. Keski-aikaiseen Britanniaan sijoittuva **DotC** ei ole kovin monipuolista strategiaa, mutta se oli aikanaan ällistyttävän hieno audiovisuaalinen kokemus ja esitteli, mihin Amiga kykenee. Saman firman **Rocket Ranger**, **Wings** ja **It Came from the Desert** sisältävät korean kuoren lisäksi myös pelin. Cinenaware on viime aikoina aktivoitunut tuottamaan uusintaversioita klassikoistaan etenkin mobiililaitteille.

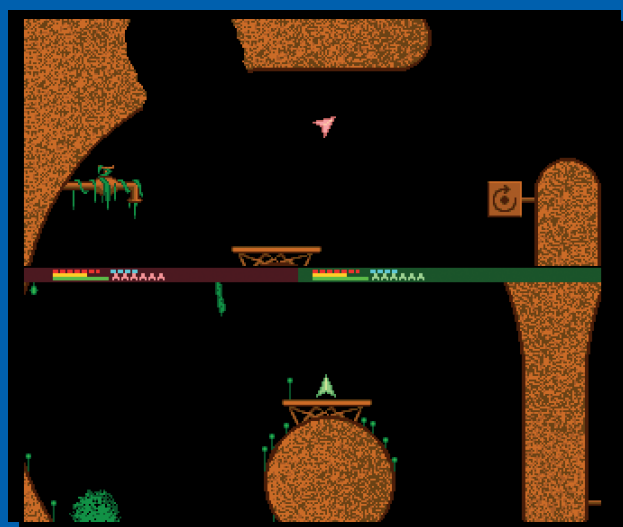
Pinball Dreams/Fantasies/Illusions

Digital Illusions oli omaa sukua demoryhnä **The Silents**. Pinball-pelisarja käyttikin alusta asti hyväkseen demoestetiikkaa ja Amigan parhaita puolia, kuten sulavaa ruudun vieritystä ja sample-pohjaisista musiikkia. Uudemmille Amigoille tarkoitetut AGA-versiot koreilevat vielä värikkäämmillä pelikentillä, mutta peruspeli on niissäkin ennallaan.

Vaikka aikaa on kulunut, sarjan pelit ovat edelleen päteviä tietokoneflipperieitä. **Pinball Dreams** ja **Pinball Fantasies** on myös käännetty hengästyttävän monelle muullekin alustalle aina Super Nintendosta Playstation 3:een asti. Digital Illusionsista tuli sittenkin DICE ja osa EA:ta.



Turboraketti II



Cannon Fodder



Ports of Call



Tässä tietokoneistetussa lautapelissä jopa neljä laivanvarustajaa kerrallaan pääsee seikkailemaan kansainvälisen rahtiliikenteen maailmassa. Hienot grafiikat ja erilainen peli-idea tekevät tästä mukavaa ajanvietettä, vaikka pelissä on myös omat ärsyttävyytensä (toimintajaksot ovat ylivoimaisia, ja toimistossa on pakko vierailla ajoittain, vaikkei mitään asiaa olisikaan). Pelisessiot venähtävät helposti tuntien mittaisiksi.

Peliä myytiin Suomessa Amigan kylkiäisenä, joten täällä se oli harvinaisen monella jopa alkuperäisenä levykkeenä. Maailmalla taas PoC oli surullisen kuuluisa siitä, miten paljon sitä jaeltiin pirraattikopioina.

Varo ettet lieemeen juutu! Tämä ohje on hyvä muistaa, kun käy taisteluun kaveriaan vastaan tässä suomalaisessa kulttipelissä. Luolastoissa lentelevät ja toisiaan ammuskelevat raketit muuttavat helposti minuutit tunneiksi ja laittavat ystävyysuhteet koetukselle. Alkujaan sharewarena julkaistu peli muuttui ilmaiseksi vuonna 1999.

Suomessa tehtiin toki Anigalle paljon muutakin kuin vain ilmaispelejä. **Bloodhousen Stardust** ja sen jatko-osa **Super Stardust** ovat äärimmäisen näyttäviä **Asteroids**-klooneja, joiden tunneliefektit ovat nekin puhdasta demoskene-estetiikkaa. **Terranarquen Elfmania** taas oli kolikkoautomaattitasoista nätkintää jopa perusmallin Anigalla.



Sensible Software oli sekin eräs Anigan kulta-aikojen sankarikehittäjistä. Monille rakas peli on huippuhitti **Sensible Soccer** variaatioineen, mutta itselleni **Cannon Fodderin** viidakkotaistelut ja paikoin äärimmäisen musta huumori toimivat parhaiten. Hiirivetoinen toimintapeli muistuttaa tempoltaan nykyaikaista reaaliaikastrategiaa ilman rakenteluvaiheita.

Hupaisa latausmusiikki sodan hauskuudesta on jäänyt monelle korvanadoksi, mutta sille muodostaa kontrastia alkukuvaan kertyvä kuolneiden sotilaiden hautausmaa. Lopulta pelin sanoma on paremminkin pasifistinen.

The Settlers



Blue Byten tähän päivään asti jatkunut kevyiden strategiapeliin sarja käynnistyi sekin juuri Anigalla vuonna 1993. Oman valtakunnan rakentaminen hassujen pikkuniesten avustuksella jaksaa viehättää edelleen, vaikka jatko-osat toki paransivatkin etenkin logistiikkaketjuja vaivannutta hitautta.

Settlers oli aikanaan Aniga-peliksi edistynyt, sillä se osasi muun muassa skaalata kartan kokoa sen mukaan, paljonko koneessa oli muistia ja suoritusnopeutta. Tämä ilahdutti muun muassa kirjoittajaa, joka oli käyttänyt kaikki kesätyöansionsa Aniga 1200:n 68030-turbokorttiin.

Super Skidmarks



Uudesta-Seelannista kotoisin oleva Acid Software loi **Blitz Basic**-ohjelmointikielen ja osoitti sen tehon kirjoittamalla kielellään useita pelejä. Hauska autopeli **Skidmarks** ja sen jatko-osa **Super Skidmarks** mahdollistavat kahden Anigan linkittämisen sarjaporttien (tai jopa modeeni-yhteyden) kautta. Tällöin ratakkin voidaan asettaa vierinisen sijaan näkynään puoliksi kahden koneen ruudulla.

Kun linkkikaapeli on kaivettu esiin ja **Skidmarks**illa päästy vauhtiin, kannattaa katsastaa myös klassinen **Stunt Car Racer**. Hurja autopeli muuttuu vielä hurjemmaksi, kun sitä ajaa kaveria vastaan. Sotiniiseen mieltyneemmät voivat ladata koneisiinsa **Falcon**-lentosimulaattorin ja yrittää pudottaa kaverin taivaalta.

Moonstone – a Hard Day's Knight

Tästä pelistä on harvempi kuullut, mutta ne jotka ovat, vannovat sen nineen. Alkuperäisistä kappaleista pyydetään verkkohuuto-kaupoissa hulvattomia hintoja.

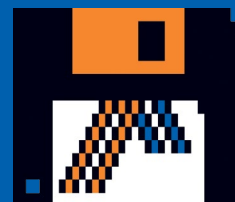
Kuten nimikin hieman vihjaa, kieli on ollut tiukasti poskessa tätä taistelupelin ja kevyen roolipelin yhdistelmää väsättäessä. Jopa neljä ritaria kerrallaan voi kilvoitella jumalten suosiesta ja Kuukivestä monia erilaisia vihollisia vastaan, eikä verenvuodatusta ruudulla suotta kainostella.



Lotus Esprit Turbo Challenge



Magnetic Fields tunnettiin etenkin autopeleistään, kuten **Super Cars**- ja **Lotus**-sanjoista. **Turbo Challenge** jatko-osineen paikkasi paljon sitä harmitusta, joka syntyi Amigan kelvottomasta **Outrun**-versiosta. Sulavaa grafiikkaa ja hyvää pelattavuutta tukee etenkin kakkososassa erinomainen ääniraita, joka saanee Yello-yhtyeen fanit hymyilemään tietäväisesti. **Magnetic Fields** on julkaissut kaikki pelinsä ilmaislevitykseen.



Chaos Engine

Bitnap Brothers on myös kehittäjäryhmä, jonka historiasta Amiga-klassikkoja löytyy kasoittain. **Xenon** ja **Speedball** jatko-osineen ovat oivia pelejä nekin, mutta omaan nakuuni parhaalta on maistunut tänä Amigan loppuaikoina julkaistu kaksin pelattava toinintaräiskintä. Steampunk-henkinen grafiikka ja erinomaiset kontrollit tekevät tästä hienon pelin kaverin kanssa koettavaksi.

Amigan tukijaloista puhuttaessa ei sovi unohtaa **Team 17**:ää. Klassinen matosotapeli **Worms** nähtiin ensimmäisenä Amigalla ja **Alien Breed 3D** taas toi lohtua **Doom**-kateutta poteville amigisteille. 🐉





Teksti: Mikko Heinonen
Kuvat: Manu Pärssinen, Wikimedia
Commons -käyttäjät Toni Birrer,
Evan-Amos, Bill Bertram, Blake
Batterson, MD56502



Ei näin!

– Commodoren kääntöpuoli

Nimi Commodore tuo mieleen lämpimiä muistoja nuoruuden bitinnypläyksestä. Vaikka monet tuotteet olivat sympaattisia, itse yrityksen toiminnasta voi olla muutakin mieltä.

Commodore Portable Typewriter Company perustettiin vuonna 1954, nimensä mukaisesti valmistamaan kirjoituskoneita. Muutama vuodessa japanilaiset söivät amerikkalaiskoneiden markkinat, mutta puolalaissyntyinen Jack Tramiel osoitti uusiutumiskykyä suuntaamalla yhtiönsä laskukoneiden valmistukseen. Uusi nimi Commodore Business Machines kuvasi toimintaa paremmin.

Mekaanisista laskimista edettiin 70-luvulle tultaessa elektronisiin, ja paljon muutakin valmistettiin siinä

sivussa. Turvatakseen komponenttien saatavuuden Commodore osti kriittisimmät alihankkijansa. Näin se sai kuin vahingossa valmiudet edetä laskintehtailusta kokonaisten tietokoneiden valmistamiseen. Yhtiön pääinsinööri Chuck Peddle näki jo hyvissä ajoin, että pelkkien laskukoneiden aika on ohi. Tramiel kuunteli häntä tarkasti. Ensimmäinen tietokonemallisto Commodore PET kelpasi etenkin kouluihin, mutta vuonna 1981 julkaistu VIC-20 kävi kaupaksi jo kuluttajillekin. Siitä tuli ensimmäinen miljoona kappaletta myynyt kotitietokone.

Tasavallan perimätietokone

VIC-20:n seuraaja Commodore 64 oli edeltäjänsä näköinen, mutta lihoitettu juuri oikeista kohdista menekin takaamiseksi. Tehokkaan kustannuskurin ansiosta se tarjosi 64 kilotavun muistin sekä huikeat grafiikka- ja ääniominaisuudet halvemmalla kuin useimmat kilpailijansa. Tätä Tramiel oli tavoitellutkin, sillä tarkoitus oli hinnoitella tuotteet aggressiivisesti. Seurauksena oli suoranaan hintasota eri merkkien välillä. Siitä seurannut kotitietokoneiden suosion nousu löi nopeasti pelkät pelikonsolit kanveesiin ja varsinkin kukkulan entinen kuningas Atari natisi liitoksissaan.

Vuonna 1982 esitelty **C64** oli mainio laite, mutta ei aivan niin mainio kuin ajan kultaamat muistot haluaisivat uskotella. Kustannusten säästämiseksi se kierrätti VIC-20:stä niin paljon kuin mahdollista. Suoritin oli miltei sama, näppäimistö ja koteloointi samaten. Massiivinen ja lämpöä hohkaava levyasema oli edelleen sukua jo PETissä käytetylle antiikkipyöritykselle ja kasettiasema taas eräs markkinoiden epäluotettavimmista. Jos peliohjaimia ja tv-liitintä ei lasketa, C64:ssä ei ollut yhtään täysin standardin mukaista liitintää ulkomaailmaan.

Huonoin ominaisuus oli kuitenkin käyttöliittymä, tai oikeastaan sen puute. Kaikki toimenpiteet tehtiin alkeellisen *Commodore BASICin* kautta. Aloittelija pääsi käyttämään koneen grafiikkaa ja ääniä vain monimutkaisilla POKE-loitsuilla tai hankkimalla jonkin kaupallisen - ja useimmiten hitaan - BASIC-laajennuksen.

Mutta aivan kuten Commodore oli laskenut, puutteilla ei ollut suurta merkitystä menekin kannalta. Tärkeintä oli oikea hintataso. Kotimikrolehdistö ehkä naureskeli kuusnepan omituisuuksille,

mutta ostava yleisö kantoi setelit kauppaan. Valtavasta konekannasta seurasi valtava ohjelmisto- ja oheislaittevalikoima. Kierre oli valmis: C64 pysyi markkinoilla yli kymmenen vuotta ja myi lähteestä ja laskutavasta riippuen 12-17 miljoonaa kappaletta.

Huomioiden markkinoiden tuolloisen koon tämä on saavutus, jota tuskin enää vertailukelpoisesti voidaan ohittaa. Commodore oli valmistanut juuri sellaisen tietokoneen, jonka se yhtiönä osasi parhaiten valmistaa. Vuoden 1983 myynti rikkoi miljardin dollarin rajan.



Commodore 64

Raha ei tuo ystäviä

Markkinoille virtasi yhä edullisempia laitteita, jotka uhkasivat etenkin VICin markkinoita. **Sinclairin ZX-sarja** ja **Mattel Aquarius** rujoine kalvo- ja kuminäppäimiseen olivat Commodore-tuotteisiin nähden alkeellisia, mutta myös todella halpoja valmistaa. **Texas Instruments** taas laski oman **TI-99:nsä** hintaa parantaakseen sen menekkiä. Lisäksi oli vaarana, että jokin japanilainen yritys toisi markkinoille oman superhalvan laitteensa.

Tramiel käynnistikin vuonna 1983 **Commodore 116** -projektin, jossa synnytettiin Commodoren oma halpiskone. Se oli hieman VICiä edistyneempi, mutta ei vetänyt vertoja C64:lle. BASIC-versio 3.5 oli lopulta hieman kehittyneem-

pi ja sisälsi myös grafiikkakäskyjä. Kumimattinäppäimistöllä toteutettuna hinnan piti jäädä selvästi alle maagisen sadan dollarin rajan. Sen vastapainoksi Tramiel halusi Commodoren panostavan myös aivan uuden 32-bittisen koneen kehittelyyn, jotta yhtiöllä olisi tulevaisuutta.

Tässä vaiheessa ristiveto yhtiön sisällä kuitenkin yltyi kriittiselle tasolle. Commodore oli ollut käytännössä kahden miehen show: Tramiel pyöritti päivittäistä toimintaa erittäinkin pikkutarkasti, mutta suurista linjoista vastasi pääomistaja ja hallituksen puheenjohtaja, kanadalainen **Irving Gould**.

Jack lähtee, aukko jää

Gould oli antanut Tramielin huoseerata yhtiössä melko vapaasti, sillä vaikka hän tapasikin puuttua liki joka asiaan - hieman kuin *Steve Jobs myöhemmin* - juuri Tramielin kustannustietoisuus oli nostanut Commodoren menestykseen. Valitettavasti tämä pikkutarkkuus ja penninvenytys oli myös suututtanut monia toimijoita ja luonut Commodorelle mainetta halpavalmistajana. Hallitus halusi eroon laskevien hintojen kierteestä ja Tramielin piti lopettaa mikromanagementinsä. Kiistaa syntyi myös siitä, että Tramiel oli nimittänyt yhtiön johtotehtäviin omia sukulaisiaan.



Gouldin ja Tramielin välirikon lopputuloksena Jack Tramiel lähti perustamastaan Commodore Internationalista tammikuussa 1984. Vain muutamaa päivää aiemmin hän oli tiedottanut yhtiön kaikkien aikojen parhaasta tuloksesta. Mikä pahinta, Tramiel ei lähtenyt eläkkeelle vaan iskeäkseen takaisin: vielä saman vuoden kesäkuussa hän osti vaikeuksiin joutuneen Atarin tietokoneliiketoiminnan ja palkasi useita entisiä Commodoren insinöörejä suunnittelemaan tuotteita entisen firmansa päämenoksi.

Tramielin lähtö iski Commodoreen kovemmin kuin yhtiössä tuolloin tajuttiin. Ristiriitainen ja saita diktaattori oli myös visionääri, ja hänen johtajuutensa kadottua yhtiötä alkoi pyörittää kasvoton taskulaskinosasto.

Mitä nämä tuotteet ovat?

Ensioire nuottien hukkaamisesta oli Commodore 16. Se oli käytännössä aiempi projekti 116 pakattuna VICistä ja C64:stä tuttuun koteloon, joka tällä kertaa maalattiin mustaksi. Vuonna 1984 esitelty laite oli jo valmiiksi hieman vanhahtava, minkä lisäksi se teki jälleen uuden yhteensopimattomuusennätyksen. Edes muiden Commodore-mikrojen peliohjaimet ja kasettiasema eivät enää sopineet uutuuteen kiinni, ohjelmistosta nyt puhumattakaan.

Olenneista oli kuitenkin se, että Commodore 16 oli jo julkaisussaan turha tuote. Commodore 64 dominoi markkinoita ja sen valmistuskustannukset olivat jo laskusuunnassa. VICiä myytiin kovilla alennuksilla ja Tramielin pelkäämät Mattel ja TI käytännössä poistuvat markkinoilta jo ennen C16:n julkistamista. Halpaa japanilaista kotimikroa ei koskaan tullut. Sen sijaan, että Commodore olisi haudannut koko hankkeen hiljaisuudessa, C16:n julkistamalla se loi lähinnä

kilpailua itselleen ja sekaannusta jo valmiiksi sekaville markkinoille.

Oudot julkistukset eivät loppuneet tähän. C16:n kumppaniksi esiteltiin myös Plus/4, joka oli Commodore 116:n näköinen, mutta varustettu oikealla näppäimistöllä. Sisäisesti se oli 64 kilotavun muistilla varustettu C16, jossa oli sisäänrakennettu hyötyohjelmapaketti. Voisi veikata, että Plus/4:n sovelluksilla pyöritetyt kotitoimistot on nopeasti laskettu. Lehdistö odotti uudelta 64 kilotavun koneelta vähintään C64:n veroisia ominaisuuksia, mutta sai pettyä.

Koko TED-piiriin perustuva laiteperhe (C116, C16 ja Plus/4) floppasi pahasti etenkin Amerikan markkinoilla. Niitä päädyttiin dumpaamaan Etelä-Amerikkaan ja Itä-Eurooppaan, jossa niille syntyi myös harrastajakuntaa. Suomessakin Commodore 16:ta nähtiin jonkin verran, sillä merkki ja maahantuoja olivat jo vakiinnut asemansa ja hinta oli halpa.

128 on vähemmän kuin 64...

Plus/4-fiaskon yhteydessä Commodorelle oli valjennut, miten tärkeä asia C64-yhteensopivuus oli. Seuraavaa kotimikroa suunniteltaessa ei toistettaisi vanhoja virheitä, vaan kuusnepan ominaisuuksia laajennettaisiin. Lehdistössään näkyi toiveita siitä, että nähtäisiin "supernepa" joka tekisi C64:stä paremmin hyötykäyttöön sopivan, korjaisi BASICin puutteita ja nopeuttaisi massamuisteja, mutta säilyttäisi yhteensopivuuden.

Commodore 128:sta tulisi juuri tällainen tietokone. Täyden C64-tuen lisäksi se tarjoaisi 128 kilotavun



Commodore 116

tilan, jossa käytävissä olisi 80 merkin näyttö, parempi BASIC ja nopeampi suoritin. Eikä tässä kaikki, sillä mukana olisi myös Z80, jolla voisi ajaa CP/M-käyttöjärjestelmää. Yhteensä tarjolla olisi siis kolme erilaista tietokonetta yhden kuorissa.

Vuonna 1985 julkaistu C128 olikin todellinen insinöörien taidonnäyte. Monimutkainen kahta suoritinta käyttävä laite jakoi muistiaan monella eri tavalla ja tasapainotteli eri komponenttien suorituskyvyn rajamailla. Sekä C128-tilan että Z80:n käyttöön liittyi erilaisia rajoituksia, jotka söivät niiden käyttökelpoisuutta. Muun muassa CP/M toimi hitaasti, koska osin C64:stä kotoisin olevat sisukset eivät pysyneet kyydissä Z80:n neljän megahertsin kelloaajuudella.

...ja 3 on vähemmän kuin 1

Lopullisesti C128:n upotti juuri se asia, jonka oli tarkoitus pitää se pinnalla. Koska C128 oli C64-yhteensopiva, mutta ei korkean hintansa ja melko ristiriitaisten ominaisuuksiensa vuoksi saavuttanut erityistä suosiota, sille ilmestyi varsin vähän ohjelmistoa. Hyvin nopeasti sitä kaupattiin myös miltei samaan hintaan kuin pikkuveljeään. Moni valitsi silti perinteisen kuusnelosen, sillä etenkin C128:n uudet ja monipuolisemmat levyasemat nikottelivat perinnekalli

1541:lle laadittujen kopiosuojausten kanssa.

Laitteelle ilmestyi muutamia asiallisia hyötyohjelmia, etenkin ohjelmointikieliä. Satunnaisten käyttäjien silmissä parhaita ominaisuuksia olivat kuitenkin lähinnä C128D-mallin irrallinen näppäimistö ja mahdollisuus hyödyntää lisämuistia *Ultima V* -roolipelissä. TED-malleihin verrattuna C128 myi kohtalaisesti, mutta C64:n luvuista vain vajaan kolmanneksen.

torolan MC68000-suorittimeen perustuvan laitteen oli tarkoitus olla pelikonsoli, jonka voisi laajentaa kokonaiseksi tietokoneeksi. Yhtiön nimi muuttui sittemmin **Amiga Corporationiksi** ja itse tuotteesta tuli **Amiga**.

Kun Amigan rahat ponnisteluista huolimatta loppuivat vuonna 1983, eikä tuote ollut läheskään valmis, perustaja *Jay Miner* otti yhteyttä vanhaan työnantajansa. Atari oli kiinnostunut tekniikasta ja

tynyt, kun myöhemmin selvisi, että Tramiel aikoo ostaa Atarin tietokonepuolen. Vanhan sopimuksen nojalla teknologia saattaisi siis olla siirtymässä hänen haltuunsa joka tapauksessa. Tramiel itse ei tosin tiennyt sopimuksesta tässä vaiheessa mitään.

Jo miltei epätoivoiseksi käynyt Amiga kääntyi Commodoren puoleen. Syvä juopa Tramielin ja hänen entisen yhtiönsä välillä oli varmasti osasyynä siihen, että Commodore lähti mukaan järjestelyyn, jossa se osti Amiga Corporationin kokonaisuudessaan ja liitti sen tytäryhtiökseen. Osana kauppahintaa Commodore maksoi Atarille sen Amigaan sijoittamat rahat takaisin.

Ei kulunut kauankaan, ennen kuin Atari Inc:n ja Amiga Corporationin välinen sopimus paljastui Tramielille. Commodore kuitenkin katsoi, että koska Tramiel oli osistanut nimenomaan Atari Computersin, ei koko Ataria, eivät osiin pilkotun emoyhtiön solmimat sopimukset enää päkeneet. Asiaan vaikutti merkittävästi myös se, että Tramielin väitettiin lähtiesään kaapanneen Commodoresta mukaansa sekä osaajia että teknologiaa. Yhtiöt riitelivät pitkään eri oikeusasteissa, mutta lopulta molemmat kiistat haudattiin yksityisellä sopimuksella.



Commodore Amiga 1000

Kiista ystävättärestä

Ehkä jotenkin kuvaavaa on, että se toinen Commodoren suurmenestys ei ollut alkujaan Commodoren tuote, ja että se päättyi yhtiön syliin ainakin osittain sattuman kautta. Samaan aikaan kun C= myi vikkejä ja kuusnelosia kuin leipää, juonittiin toisaalla jo tekniikan seuraavaa sukupolvea.

Hi-Toro-niminen yritys koostui etupäässä Atarilta lähteneistä työntekijöistä. Henkensä pitimiksi se suunnitteli ja myi peliohjaimia ja konsolipelejä, mutta varsinainen tuote oli nimeltään **Lorraine**. Mo-

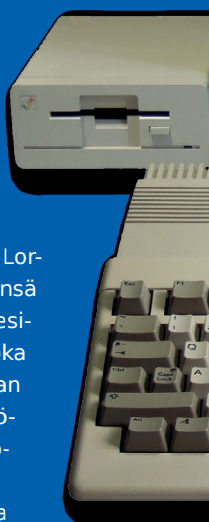
antoi rahoitusta, jonka vastineeksi se saisi myöhemmin kehitystyön tulokset käyttöönsä. Samaan aikaan kuitenkin alkoi videopelialan romahdus, jonka vuoksi Atari joutui suuriin vaikeuksiin. Commodorella taas kuohui muuten ja Tramiel sai lähteä.

Vapailla markkinoilla uutta yhtiötä etsivä Tramiel tapasi Amiga-tiimin alkuvuodesta 1984. Hän teki selväksi, että haluaisi kyllä laitteen teknologian, mutta ei sen mukana seuraavaa henkilökuntaa. Tämä huolestutti amigalaisia, jotka tietenkin pelkäsivät työpaikkansa puolesta. Huoli ei ainakaan vähen-

Pelikonsolista PC:ksi

Commodoren johto halusi päästä eroon halpavalmistajan maineesta ja voittaa markkinoita yrityskoneiden puolella. Sen näkemys Amigasta oli niin kaukana alkuperäisestä Lorraine-konsolista kuin yleensä voi kuvitella. Vuonna 1985 esiteltä ensimäinen Amiga, joka sittemmin opittiin tuntemaan **Amiga 1000:na**, oli kallis hyötykäyttöön tarkoitettu henkilökohtainen tietokone.

Hinnastaan huolimatta





A1000 jouduttiin tuomaan markkinoille niin keskeneräisenä, ettei edes käyttöjärjestelmää uskallettu polttaa ROM-piireille. Tonnisen omistajat saivat kylmäkäynnistyksen yhteydessä latailla koneeseensa kaksi levykkeellistä tavaraa ennen kuin konetta pääsi käyttämään.

Itse käyttöliittymä oli toki mulistava, mutta myös tahmainen ja kaatuileva. Sisäpiirivitsiksi tarkoitettu *Guru Meditation* -viesti lakkasi huvittamasta, kun sitä sai katsella toistuvasti. Ohjelmiston vähyyttä valitettiin joka puolella siinä määrin, että siitä alkoi muodostua itseään toteuttava ennustus. Hyötykoneena Amigalle ennustettiin korkeintaan marginaalista menestystä. Jokuset ilmestyneet pelit sentään antoivat osviittaa siitä, mihin laite pystyy, mutta harvalla oli sijoittaa pelikoneeseen henkilöauton hintaa.



Commodore Amiga 500

Tehdään kun ehditään

Tilanteen korjaaminen vei Commodorelta vain vajaat kaksi vuotta. Vuonna 1987 esitelty Amiga 500 oli se tietokone, joka Commodoren olisi todennäköisesti pitänyt julkaista jo alkujaan. Kalliin hinnan beetatestauksesta maksaneet A1000:n omistajat tuskin ilahtuivat muun muassa siitä, että laajennusväylä oli A500:ssa siirretty toiseen kylkeen, eivätkä lisälaitteet siksi toimineet ristiin.

Lopputuloksena oli kuitenkin viimein kuluttajahintaluokan laite, jota Commodore saattoi tarjota ikääntyvän C64:n tilalle. Vaikka käyttöjärjestelmä ei ollut edelleenkaan virheetön, se oli nyt ROM-piirillä ja enimmäkseen käyttökelpoinen. Osittain teknisen kehityksen verkkaisuudesta ja osittain Amigan edistyneisyydestä kertoo se, että tullessaan markkinoille vuonna 1987 Amiga 500 oli yhä pääosin nykyaikainen, vaikka sen juuret ulottuivat melko suoraan Hi-Toroon ja vuoden 1984 CES-prototyyppiin.

Atari oli ehtinyt omalla **ST:llään** markkinoille hieman ennen Amigaa, ja Tramiel oli tapansa mukaan painanut hinnan minimiin. Jälleen kerran hieman vaatimattomampi tekniikka ja paljon vaatimattomampi käyttöjärjestelmä eivät kaikille painaneet, vaan ST:stä tuli Amigalle pitkäksi aikaa kova kilpailija. Huvittavaa sinänsä on, että siinä missä C16:lla Commodore loi suoraan itselleen

kilpailua, ST:n se synnytti karkotamalla Tramielin (ja hänen mukanaan ehkä puolivalmiit 32-bittisen koneen suunnitelmat). Sopivilla tähtien asennoilla meillä olisi ollutkin Commodore ST ja Atari Amiga.

Suuri pyörä pysähtyy

Amiga 500 ja C64 toivat yhtiöön rahaa, mutta sitä ei juuri käytetty mielekkääseen kehitystyöhön. Viissatosen ja sen isoveljen **Amiga 2000:n** jälkeen Commodore tuntui enimmäkseen jähmettyneen paikoilleen. Alkuperäisen Lorraine-tiimin jäsenet olivat suurimmaksi osaksi kyllästyneet ison firman jäykkyyteen, pakanneet kimpsunsa ja lähteneet muihin haasteisiin.

Amigan kehitystyö oli niinkin verkkaista, että aikoinaan huhuttiin Commodoren hukanneen tiettyjen piirien piirustukset kokonaan. Vaikka huhu tuntuukin uskomattomalta, se selittäisi esimerkiksi sen, ettei laitteen äänipiiriin kajottu enää koskaan. PC-yhteensopivien valmistamistakin Commodore harrasteli, mutta sen tuotteet eivät juuri erottuneet kilpailijoistaan.

Ehkä jonkinlaisen yhtiön sisäisen päähänpintymän vuoksi tuotekehitys suunnattiin lähinnä yrityskäyttöön. Messuilla esiteltiin *UNIX*-koneita ja unelmia, jotka eivät koskaan toteutuneet. Vuonna 1990 esitelty **Amiga 3000** puolestaan oli hurjan tehokas ja monipuolinen laite, mutta myös hinnaltaan lähinnä tähtitieteellinen. Kotikäyttäjät saivat vuoden odotuksen jälkeen lopulta **Amiga 500+:n**, joka sisälsi muutamia A3000:n uusia ominaisuuksia, mutta sinnitteli edelleen saman 7,14 megahertsin MC68000:n voimin. Tässä vaiheessa koneen perusdesign oli jo seitsemänvuotias.

Eräs syy hitaaseen ja iteratiiviseen kehitykseen oli toki myös se, että jokainen laitteeseen tai käyttöjärjestelmään tehty muutos

rikkoi aina ison määrän ohjelmia. Ohjelmoijat olivat oppineet C64:llä siihen, että muistia ja rekistereitä voi käsitellä suoraan osoittamalla, koska käyttäjien raudassa ei juuri eroja ollut. Amiga oli luonteeltaan laajennettavampi, mutta Commodoren ohjeet oikeista menettelytavoista kaikuivat kuuroille korville. Loppukäyttäjille tämä tarkoitti mm. lisämuistien ja lisälevyasemien irti nyppimistä, jotta vanhemmat pelit toimisivat. Kiintolevyt ja uudemmat käyttöjärjestelmäversiot sitten vasta asioita sotkivatkin.

Silloinkin kun Commodore teki jotain kuluttajille, oli vähän vaikea ymmärtää, mikä tuotejulkistuksen tarkoitus oli. Commodore CDTV oli CD-soittimen näköiseen pakettiin kääritty Amiga 500 CD-asemalla. Sitä ei kuitenkaan saanut kutsua Amigaksi, vaan Commodore Dynamic Total Visioniksi, eikä sitä saanut esitellä yhdessä Amigoiden kanssa. Monet ottivat neuvosta vaarin ja jättivät laitteen kokonaan ostamatta, vaikka se tavallaan oli aikaansa edellä. Toisen suursuosikin, Commodore 64 Games Systemin, tulinkin jo haukkuneeksi aiemmassa jutussa (ks. Skrolli 2014.1).

Kiitos luottamuksestanne, tässä kuraa

Maaliskuussa 1992 kuultiin ilouutinen, sillä kotikäyttäjille julkaistiin aivan uusi Amiga, kompaktikokoinen A600. Lähemmässä tarkastelussa se osoittautui lähinnä pienennetyksi ja pintaliitostekniikalla valmistetuksi 500+:ksi, jossa oli (halvin mahdollinen) kiintolevyliitäntä. Piirisarja ei ollut uudistunut mitenkään sitten Amiga 3000:n eikä suoritin ollut lainkaan viissattua nopeampi. PCMCIA-liitäntän lisäämisen vastapainona koneesta oli rosvoitu AmigaBUS-väylä, mikä teki isosta kasasta laajennuksia hetkessä käyttökelvottomia.

Lopullinen rätti Amiga 600:n hankkineiden kasvoille oli, että

vielä saman vuoden lokakuussa Commodore toi markkinoille Amiga 1200:n. Jo sen ulkonäkö kertoi, että A600 oli ollut tähän tähtäävä harjoituskappale. A1200 sisälsi

kuitenkin masentavaa seurattavaa etenkin niille, jotka vielä uskoivat Amigaan tuotteena. Suomalaiset harrastelehdet sivusivat asiaa vain ohimennen, mutta etenkin britti-



Commodore Amiga 1200

lopultakin nopeamman, aidosti 32-bittisen suorittimen ja uuden näyttöpiirisarjan, joka paransi grafiikan tarkkuutta ja värimääriä. Jälleen kerran se olisi tehnyt markkinoilla suuremman vaikutuksen kahta vuotta aiemmin, mutta kävi tällaisenaankin kaupaksi paremmin kuin moni odotti.

Valitettavasti Commodore ui jo A1200:n julkaisun aikoihin aivan liian syvällä. Se oli hukannut aiemmat myyntituotonsa kehittämällä vääränlaisia tuotteita väärille markkinoille, ja kun vanhat lypsy-lehmät lopulta kuolivat, uusien tekemisessä kesti liian kauan. Ehkä itse luomastaan kilpailusta pelästyneenä se oli ollut kuluttajajulkistustensa kanssa ylivarovainen ja pyörittänyt samoja tuotantolinjoja, kunnes kilpailijat saivat etumatkan kiinni.

Kuolonkorinat alkavat

Commodoren tarina on monilta osin kuin uskomaton jännityskertomus. Sen loppumetrit olivat

lehdistössä saimme kuukausittain seurata uutta puhuvaa päätä, joka houri uudesta AAA-piirisarjasta ja siitä, miten nyt laihtunut yhtiö on kovemmassa kunnossa kuin koskaan. Kaikki halusivat uskoa, että jotain vielä tapahtuu, mutta kukaan ei enää tosissaan uskonut.

Alkuperäisen Commodoren viimeinen tuote oli CD32-pelikonsoli. Se oli maailman ensimmäinen 32-bittinen CD-pelikonsoli, eikä edes huono esitys sellaiseksi. Commodorella oli töissä kekseliäitä ihmisiä, jotka olisivat varmasti osanneet luoda kaikenlaisia muitakin kuluttajatuotteita, jos heille vain olisi annettu siihen riittävästi mahdollisuuksia.

Myös aika ajoi koko Commodoren bisneksen ohi. Se oli viimeinen kotitietokonevalmistaja markkinoilla, jotka eivät enää tarvinneet kotitietokoneita. Samalla se ei tajunnut olevansa kotitietokonevalmistaja, vaan yritti kovasti olla yrityskäyttöön tarkoitettujen työasemien valmistaja. Sitä se taas ei oikein koskaan osannut. 🐟



"Amiga elää
muuallakin kuin
amigistin retro-
huuruudessa
sydämessä



Teksti: Jukka O. Kauppinen
Kuvat: Jukka O. Kauppinen ja
Risto Mäki-Petäys

Minä ja Amigani olimme aikoinaan tiukka kaksikko. Tosin suhteemme alkoi erikoisella tavalla. Ensimmäinen kohtaamani Amiga oli paikallisessa tietokonekaupassa leukoja loksautellut Amiga 1000. Boing, boing. X-Men-demoryhmän samplediskit ja XYZ-softin ensimmäinen Amiga-intro joulukuussa 1987 ihastuttivat. Mutta vasta kesällä 1988 Amiga sulatti sydämeni.

Tai pikemminkin sulattaja oli *Moku/Finnish Gold*, kun pelasimme *Byterapers*-demopartyillä *Empire*-strategiapeliä Amigalla. Se oli niin hyvä peli, että ostin tyhjän korpun, ja kirjoitin siihen "1. Empire". Warresta se siis alkoi.

Amiga 500, sitten 1200. Modautin molempiin irtinäppäimistöt. Kumpikin sai kaverikseen levyasemia, tornin SCSI-kiekoja, nauha-aseman, tulostimen ja muuta. Muistan miten vaikeaa oli ostaa tonnikassataseen kolminappista, optista hiirtä, sillä myyjä ei ollut lainkaan vakuuttunut, että tiesin mitä olin ostamassa. Tiesin minä, kolmosnappi oli näet todella hyvä tekstinkäsittelyssä!

Amiga oli myös tärkeä askel eteenpäin toimittajan uralla. Word

Perfect for Amiga oli pitkään erinomainen työkalu, joskin Saddam kirottu tekikin monesti tuhojaan. C64, Amiga ja PC elivät rinnakkain ja täydensivät toisiaan – jos tein juttua PC-pelistä tai -aiheista, kirjoitin sen Amigalla ja toisinpäin.

Ja mitä kaikkea minä ja Amigani teimme. Demoja. Sukelsimme internetin ja BBS:n maailmaan. Ylläpidin *MBnetiä*. Opiskelin ja tein multimediaa, digitaalista kuvankäsittelyä, morfausta, äänendigitointia, aivan kaikkea.

Mutta Commodore tapahtui ja lopulta Amigakin jäi harrastekoneeksi.

Saku 2014 -tapahtuma toi kuitenkin Amigan niin lähelle, kaiken lisäksi kalenteriin sopivana päivänä, joten Skrollin ständillä puuhas-

telun lisäksi päivä tarjosi tilaisuuden tutustua Amigan nykypäivään. Historiaakaan unohtamatta.

Vaikka Commodore on kuollut, Amiga elää muuallakin kuin amigistin retrohuuruudessa sydämessä. Vaikka tiettyä retrohenkisyyttä tarvitaan, niin Sakussa nähnyt uudet Amigat ovat huikkeitä tykkeitä ysärikoneisiin verrattuna. Tai jos ajossa onkin konevanhus, niin ei sekään enää aivan vanhan koulun kuosis- sa ole. Omassakin koneessani on sellaisia nykyajan ylellisyyksiä kuin Flicker Fixer ja muistikorttikiintolevy. Tekee muuten retroilusta iha- naa!

Vanhon koneiden virittely on amigistien maailmassa ilahduttavan virkeää ja aktiivista. Lisälaitteita kehitetään koko ajan, ja niin

Amigan uuden raudan kuin käyttöjärjestelmien maailmassa tapahtuu edelleen. Vaikka olen toki seurannut amigamaailmaa puolella silmällä koko ajan, niin Sakussa se - ja ylipäätään yhteisön henki - näkyivät vahvasti ulkopuolisillekin.

Saku näet järjestettiin Tampereen pääkirjastossa, minkä myötä sinne harhautui myös runsaasti aivan ulkopuolisia. Olikin kiintoisaa kuvailla näille tapahtuman ideaa. Kyllähän me tiedämme, että Amiga ei ole pelkkä tietokone. Se on jotain... Enemmän. Mutta enemmän mitä?

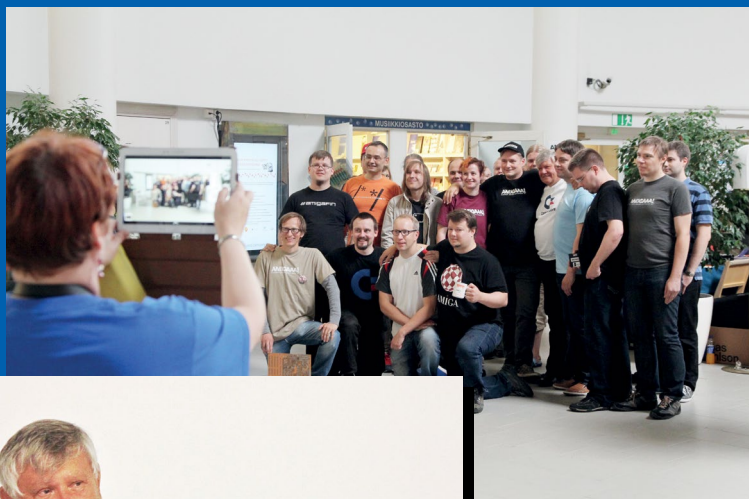
"Ajatelkaa **Apple**a. Se herättää tunteita ja keskustelua. iPhone, Mäkit, niistä välitetään tai niitä inhotaan. Amiga herätti samanlaista kiintymystä ja omistautumista jo 80-luvulla", oli eräs tapa, jolla kuvailin tapahtumaa ja sitä, mitä ja ennen kaikkea miksi täällä puuhastellaan historiallisen tietokoneen parissa.

Tämä lähestymistapa tuntuikin resonoivan ja rohkaisi monia astelemaan sisemmäs katselemaan, että mitäs täällä nyt onkaan. Monille lähteminen oli vielä vaikeampaa, etenkin jos perheen junnut olivat jumiutuneet pelien ääreen. Kummasti nuo ysäripelit iskivät nuoreenkin sukupolveen samalla, kun vanhemmat käpistelivät pelejä, hyötyohjelmia ja ajoivat MIPS-testiohjelmia retropilke ja hämmästyks silmäkulmasaan.

Sakun paluu Tampereelle keräsi yhteen satoja Amiga-harrastajia ja satunnaisia kiinnostuneita ohikulkejoita. Siellä heräsi myös yllättäviä muistikuvia ajasta, jolloin Amiga oli tärkein työkaluni. Eikä innostus suinkaan laantunut tapahtuman jälkeen. Haluan viritellä vanhaa konettani, katsella hienoja demoja ja palata vanhojen pelisuosikkieni pariin. Saku 2014 muistutti, että olen yhä amigisti.

Se on merkki hienosta tapahtumasta! 🍷

Upean Amiga-tapahtuman järjestäjät. Kiitos, SAKU ry.



Petro Tyschtschenko työskenteli Commodorella 1982-1994, aina Commodoren konkurssiin saakka. Hän jatkoi kuitenkin myös sen jälkeen Amigan parissa, yrittäen pitää konetta hengissä omistajanvaihdosten keskellä. Tyschtschenko kertoi Saku-tapahtumassa urastaan ja kokemuksistaan Commodorella, esitellen myös uutta Amiga-historiikkikirjaansa, jonka kaikki mukana olleet kappaleet löysivät nimikirjoituksineen uuden kodin. Meine Erinnerungen an Commodore und Amiga julkaistaan myöhemmin myös englanninkielisenä.

Varsin tykittävä Amiga. A4000c rev D -emolevyllinen laite on rakennettu upeaan tornikoteloon. Koneessa on kolme erillistä prosessoria: Cyberstorm 233 MHz PowerPC, 66 MHz 68060 sekä alkuperäinen 25 MHz 68030 matematiikkaprosessorilla höystettynä. Koneessa on myös muun muassa Indivision AGA MK2 -flicker fixer, 256 Mt:n muistilla varustettu Radeon 9200 PCI-näytönohjain, 256 Mt muistinlaajennus ja kuusi levyasemaa.



Tapahtumajärjestäjät ja kunniavieraat Mike Battilana (Cloanto) ja Petro Tyschtschenko suuntasivat illalla Plevna-ravintolaan nauttimaan paikallisista herkuista.

Tietokonejournalismia supervoimilla

SKROLLI

TUTUSTU, RAKASTU, TILAA

tietokonekulttuuri
ei ole mitään
meeminhuutelia!

www.skrolli.fi

#skrolli @ IRCNet



VIRTUAALIPALVELIN

alk. **9,95** €/kk (sis.alv)

512Mt muistia, 1vCPU, 10Gt levytilaa



Sisältää

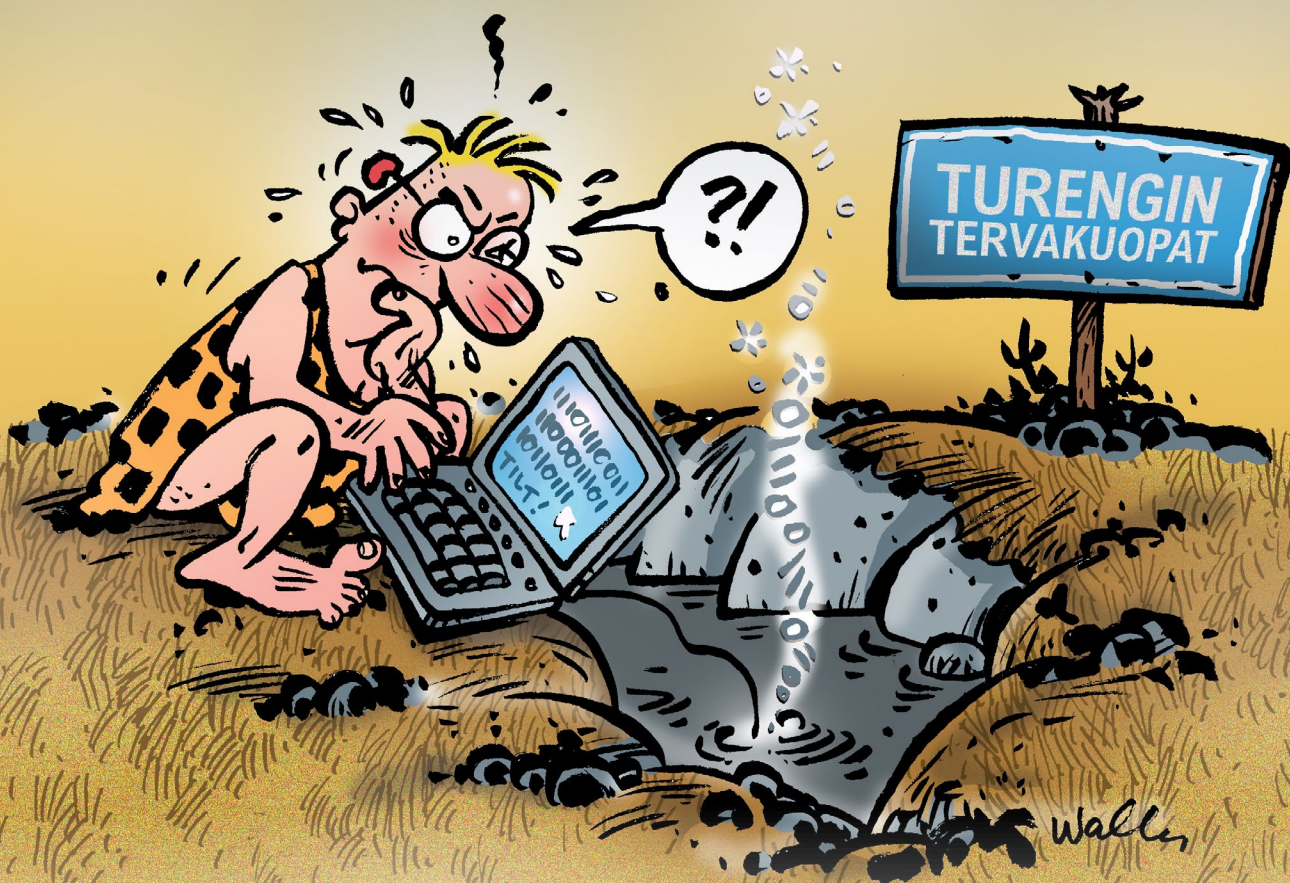
IPv4 ja IPv6 -osoitteen ja
10Gt ulosmenevää liikennettä /kk

Tarjoushinnat voimassa aina
4 x vCPU, 4Gt, 200Gt koneisiin
asti, kokoa omasi ja säästä!

Domainmaailma.



Kapsi Internet-käyttäjät ry
kapsi.fi



TURINGIN TERVAKUOPAT MATKA OHJELMOITAVUUDEN ÄÄRIRAJOILLE

Ohjelmointi on mahdollista hyvinkin yksinkertaisilla ja oudoilla järjestelmillä. Sen onnistumiseksi on tosin yleensä nyrjäytettävä aivonsa aivan toiseen asentoon.

Teksti: Ville-Matias Heikkilä Kuvat: Wallu, Ville-Matias Heikkilä

Moni on varmasti kuullut Turingin koneesta, joka on yksinkertainen, tietokoneita esittävä matemaattinen malli. Tietotekniikkapioneeri Alan Turing kehitti mallin 1930-luvulla, mutta se kuuluu edelleen tietojenkäsittelytieteen peruskalustoon.

Turingin kone on siitä merkittävä, että yksinkertaisuudestaan huolimatta sillä pystyy suorittamaan kaikki tehtävät kuin millä tahansa tietokoneella tai ohjelmointikielellä. Jos siis jollain järjestelmällä saa simuloitua Turingin konetta, pystyy se teoriassa kaikkeen siihen mihin mikä tahansa tietokone – kunhan aikaa ja tallennustilaa on tar-

peeksi. Tällaista järjestelmää kutsutaan *Turing-täydelliseksi*.

Turingin kone perustuu muistinauhaan, jota kelataan edestakaisin. Nauhalle on tallennettu symboleja, joita kone pystyy lukemaan ja korvaamaan uusilla. Koneen ohjelmana on taulukko, joka määrää, mitä koneen tulee tehdä minkäkin nauhalta luetun symbolin kohdalla eri toimintatiloissa: mikä symboli kirjoitetaan luetun tilalle, mihin suuntaan nauhaa kelataan, ja missä tilassa kone on seuraavaksi. Ohessa on kuvattu Turing-koneen ohjelma, joka vaihtaa nauhalle tallennetun binääriluvun ykköset nolliksi ja päinvastoin.

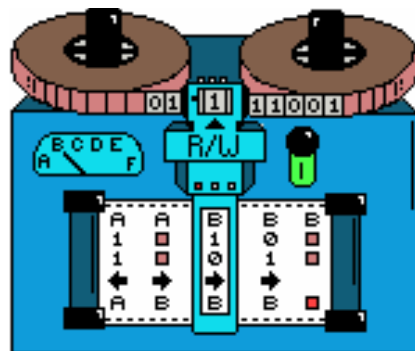
tila	A	A	A	B	B	B
luettu symboli	0	1	tyhjä	0	1	tyhjä
uusi symboli	0	1	tyhjä	1	0	tyhjä
kelaussuunta	←	←	→	→	→	-
uusi tila	A	A	B	B	B	seis

Binääri-invertteri Turingin koneella. Tilassa A kone kelaa nauhan alkuun ja tilassa B kääntää bitit.

Rakennuspalikoita etsimässä

Ohjelmointikielet on yleensä helppo osoittaa Turing-täydellisiksi: sen kun vain ohjelmoi kielellä Turingin koneen tai minkä tahansa muun Turing-täydellisen järjestelmän. On kuitenkin olemassa monia suosittuja ja käytökeltoisia kieliä, joilla Turingin koneen toteuttaminen on mahdollista mutta yllättävän haastavaa.

Käyttöjärjestelmien komentotulkit – etenkin varhaisimmissa muodois-



Turingin kone.

>	Siirrä osoitin seuraavaan tavuun.
<	Siirrä osoitin edelliseen tavuun.
+	Kasvata tavua yhdellä.
-	Pienennä tavua yhdellä.
.	Tulosta tavu.
,	Lue käyttäjältä tavu ja tallenna se osoittimen osoittamaan muistipaikkaan.
[Jos tavu on nolla, hyppää vastaavan sulkevan]-käskyn jälkeiseen käskyyn.
]	Jos tavu ei ole nolla, hyppää vastaavan avaavan [-käskyn jälkeiseen käskyyn.

Brainfuck-kielen käskyt.

saan – kuuluvat tähän ryhmään. Ne on ensisijaisesti tarkoitettu muiden ohjelmien käynnistämiseen, mutta niihin on ympätty myös ohjelmointikielimiisiä piirteitä. Käskyjä voi suorittaa sarjoissa. Niiden tulosteita voi putkittaa toisten käskyjen lähtödataksi tai ohjata tiedostoihin. Merkkijonoja voi tallentaa ympäristömuuttujiin, joista ne saa poimittua käskyjen parametreiksi. Kommentojonotiedoston osasta toiseen on mahdollista hyppiä ehdollisesti. Mutta riittääkö tämä?

Turingin koneen toteuttamiseen tarvitaan taulukoita, joihin viitataan osoittimilla, joita kelataan edestakaisin. Unixin ja MS-Dosin perustulkeissa ei ole taulukkotietorakennetta, mutta sen sijaan kunkin solun voi tallentaa omaan muuttujaansa, jonka nimen osassa on sen järjestysluku. Esimerkiksi Dosissa tämä hoituisi seuraavasti:

```
set laskuri=5
set solu%laskuri%=123
```

Mutta entäs laskurin arvon kasvatus ja vähennys? Tässä törmätään umpikujaan – Bourne Shell tai Command.com eivät nimittäin osaa laskutoimituksia ilman ulkoisia apuohjelmia! Unixin peruskalustoon kuuluu ohjelma nimeltä *expr*, jota käyttäen voi kirjoittaa esimerkiksi *laskuri='expr \$laskuri + 1'*, mutta Dosin kalustoon ei tällaista kuulu. Mikä siis neuvoksi?

Yksi ratkaisu on tehdä taulukko, jossa on kutakin lukua seuraava luku: *inc0*-muuttujan arvoksi siis asetetaan 1, *inc1*-muuttujan arvoksi 2 jne. Koska komentotulkki ei osaa muodostaa muuttujien nimiä sisäkkäisistä elementeistä, tarvitaan välivaihe, jossa laskurin arvon vaihtamisen suorittava käsky kirjoitetaan tiedostoon:

```
echo set laskuri=%inc%laskuri%% > kasvata.bat
kasvata
```

Jos laskurin arvot voivat nousta tuhan siin, ei taulukko enää mahdu muistiin, vaan toimitus joudutaan suorittamaan osissa. Koska Dosissa ei ole tarjolla keinoja merkkijonon pilkkomiseen, joudutaan luvun numerot tallentamaan välilyönnein erotettuina, jotta jokaiseen numeroon pääsisi erikseen käsiksi.

Turingin tervakuopiksi sanotaan järjestelmiä, jotka ovat teoriassa ohjelmoitavia, mutta käytännössä ohjelmoija joutuu hyvin usein tämän kaltaisiin tilanteisiin. Jokin keskeinen tietojenkäsittelyn alkeistoiminto saattaa puuttua, mutta sen pystyy jollain kierolla tavalla rakentamaan itse. Tiukimmissa määritelmissä tervakuoppia ovat vain ne äärimmäisimmät tapaukset, jotka ovat Turing-täydellisiä vain rimaa hi-poen.

Esoteeriset ohjelmointikiel

Ohjelmointiin sopimattomien kielten käyttö ohjelmointiin on monien mielestä mukavaa aivojumbppaa, joka parhaimmillaan kehittää luovaa ongelmanratkaisukykyä ja syventää suhdetta tietojenkäsittelyn perusteisiin. Turingin koneita voi yrittää rakentaa niin tekstieditorien makrokielillä kuin vaikkapa Amigan videopiirisarjalla. On kuitenkin myös aivan oma ohjelmointikielten ryhmänsä – esoteeriset ohjelmointikielien – jotka on nimenomaan tarkoitettu aivojumbpaksi eikä hyötykäyttöön.

Tunnetuin esoteerinen kieli on Brainfuck, jonka Urban Müller kehitti vuonna 1993 kokeillakseen, kuinka pieneen tilaan Turing-täydellisen kielien kääntäjän saa mahtumaan. Brainfuckiin kuuluu tavuista koostuva muisti ja yksi osoitin, joka osoittaa aluksi muistin alkuun. Käskykanta koostuu kahdeksasta yksimerkkisestä käskystä, jotka on esitetty oheisessa taulukossa.

Brainfuckin haasteet ovat erilaiset

kuin Dosin. Vaikkei siinä ole esimerkiksi kahden luvun yhteenlaskua suoraan, se on helppo rakentaa tarjolla olevista palikoista: koodinpätkä `[>+<-]` on silmukka, joka nolaa muistipaikan sisällön ja summaa samalla sen sisällön seuraavaan muistipaikkaan. Ongelmina ovat sen sijaan esimerkiksi vakiodien muodostamisen vaatimat pitkät rakenteluketjut ja laajempien kokonaisuusien hallinta. Seuraava esimerkki on tietävästi lyhin mahdollinen ohjelma, joka tulostaa sanan ”SKROLLI”:

```
+ [>+>+<<+++>
-- .----- .>-- .--- .<+ .--- .
```

Ensimmäinen rivi muodostaa tavuylivuotoa hyödyntäen muistiin kaksi kappaletta lukua 85, joka on sopivan lähellä tarvittavien merkkien ascii-arvoja 73–83. Jälkimmäinen rivi kelailee näitä lukuja oikeiden arvojen kohdille ja tulostaa ne.

Esoteerisia ohjelmointikieliä esitellään omassa wikissään osoitteessa *esolangs.org*. Monet kielet ovat selviä vitsejä: esimerkiksi *Ook!*-kielessä Brainfuckin käskyt on vaihdettu erilaisten Ook-huudahdusten sarjoiksi. Eräät kielet sen sijaan koettelevat ohjelmointiparadigmojen äärirajoja ja saattavat olla hyvinkin valaisevia, esimerkiksi äärimmilleen pelkistetty funktionaalinen kieli *Unlambda*.

Riittävä määrä käskyjä

Monet esoteeriset kielet ovat äärimmäisen yksinkertaisia ja herättävät kysymyksen, kuinka pieneksi Turing-täydellisen järjestelmän pystyy saamaan. Tätä kysymystä on tutkittu



Esoteeriset ohjelmointikielten käskyt voivat olla vaikkapa värikkäitä pikseleitä. Tässä Hello World -ohjelma Piet-kielillä. Kuva: Thomas Schoch.



Yksikäskyisellä BytePusher-virtuaali-koneella toteutettu grafiikkaohjelma. Kuva: Javamannen.

tietojenkäsittelytieteessä jo puolen vuosisadan ajan. Minimikoneiden on odotettu tarjoavan entistä syvempiä näkökulmia laskennan olemukseen, ja matemaatikko Stephen Wolfram on peräti julistanut ne uudenlaisen tieteen tekemistavan perustaksi.

Yksi tapa lähestyä kysymystä on ottaa minimalistinen kieli ja kokeilla, kuinka paljon pienemmäksi sen vielä saa. Esimerkiksi Brainfuck-tyylinen kielen ei tarvitse välttämättä operoida tavuilla, vaan bitti riittää. Tällöin käskyt `+` ja `-` yhdistyvät käskyksi `@`, joka kääntää osoittimen kohdalla olevan bitin päinvastaiseksi. Myös käskyjen määrää voidaan vähentää yhdistämällä niitä – pelkistetyimmässä versiossa on kolme käskyä, jotka vastaavat Brainfuckin yksibittisversion käsky-sarjoja `>@[,]` ja `[.<]<[, <]`.

Valtaosa nykyaikaisista tietokoneista on rekisterikoneita, joiden käskyt saavat muistiin tai rekistereihin viittaavia parametreja. Kun rekisterikonetta pelkistetään, päädytään kaksikäskyisten Minskyn koneiden jälkeen yksikäskyisiin koneisiin eli OISCeihin. Tällaisen koneen käsky voi olla esimerkiksi *subneg*, joka vähentää muistipaikan sisällöstä toisen muistipaikan sisällön ja hyppää sen jälkeen kolmanteen muistipaikkaan, jos tulos meni negatiiviseksi. Subneg-koneen pelkistetympi

muoto on *Toga*, jonka käsky kääntää yhtä bittiä muistista ja hyppää sen jäätyä ykköseksi.

Käskyn ei tarvitse välttämättä edes laskea mitään. Bytebytejump-koneen käsky kopioi tavun muistipaikasta toiseen ja hyppää sen jälkeen kolmantena parametrina olevaan muistiosoitteeseen. Tämän koneen ohjelmoitavuus perustuu runsaaseen taulukointiin ja itseäänmuuttavaan koodiin. Seuraava esimerkki on kolmitavuisia osoitteita käyttävälle koneelle, jonka käsky on siis yhdeksän tavua pitkä. Se summaa muistipaikkojen 100 ja 101 sisällöt ja kirjoittaa tulokseen muistipaikkaan 102:

```
100, 19, 9
101, 20, 18
65536, 102, 27
```

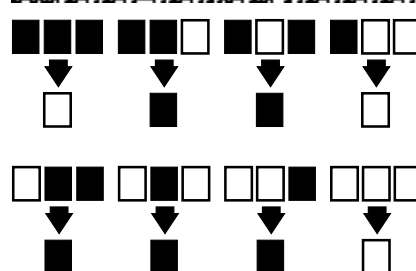
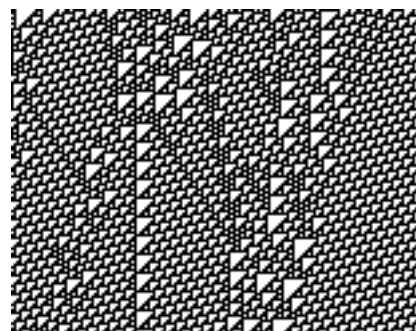
Ensimmäiset kaksi käskyä kirjoittavat kolmannen käskyn lähdeosoitteen kaksi alinta tavua uusiksi. Kolmas käsky lukee alueella 65536–131071 olevasta summataulukosta halutun summan ja kopioi sen kohdeosoitteeseen.

Sopivilla taulukoilla ja makroilla Bytebytejump-ohjelmien teko ei juuri eroa tavanomaisesta assemblyohjelmoinnista. Osoituksena tästä on minimaalinen virtuaalikone *Bytepusher*, jolla on toteutettu esimerkiksi demoeffektejä.

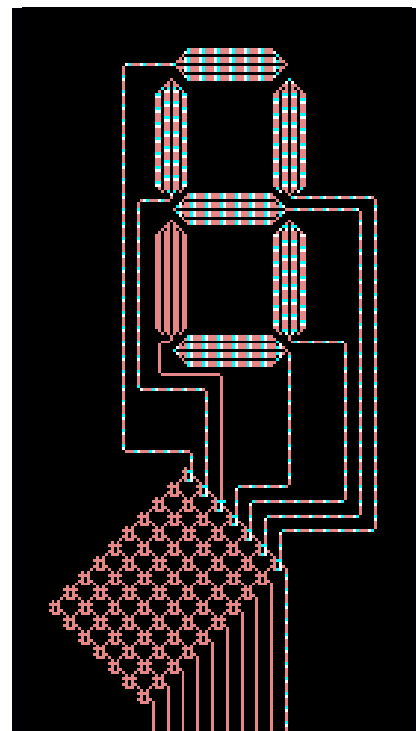
Nolla käskyä

Kun käskykannan viimeinenkin käsky poistetaan, päästään korvauskoneisiin. Ne eivät suorita käskyistä koostuvaa ohjelmakoodia, vaan toistavat yhtä ja samaa operaatiota loputtomasti koko muistilleen.

Tavanomaiset korvauskoneet käsittelevät merkkijonoa, jonka osajonoja ne muuttavat määrättyjen korvaussääntöjen mukaisesti. Tagijärjestelmät puolestaan tutkivat merkkijonon alkupäätä, poistavat sen ja lisäävät sen ohjaamana uusia merkkejä jonon loppuun. Soluautomaatit taas käyttävät kiinteitä ruudukkoita, joissa kukin alkio

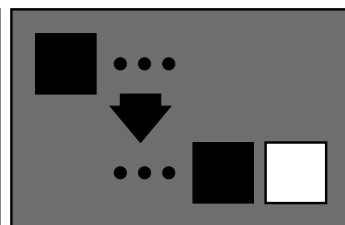
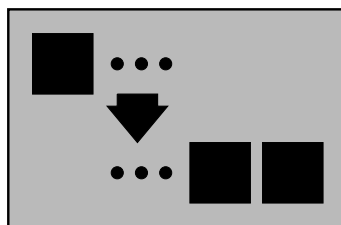
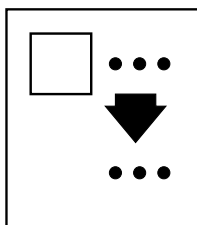
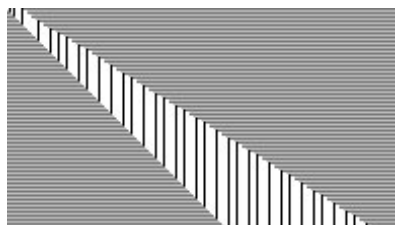


Alkeissoluautomaatti 110.

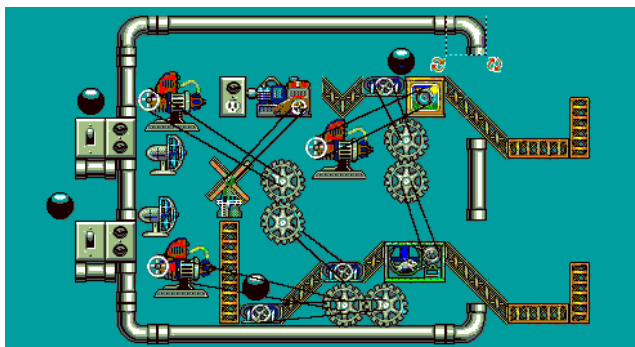


Wireworld-seitsensegmenttinäyttö de-koodauspiireineen.

korvautuu toisella naapurustonsa mukaan. Kaikissa näissä ryhmissä on hyvinkin yksinkertaisia koneita, jotka on todistettu Turing-täydellisiksi.



Yksinkertaisin Turing-täydellinen syklinen tagijärjestelmä. Musta korvautuu eri tavoin parillisilla ja parittomilla vuoroilla.



Bittisummain The Incredible Machine -pelissä.



OpenTTD:n rautateillä toteutettu digitaalikello. Kuva: Ludo D.

Kuuluisin soluautomaatti on John Conwayn vuonna 1970 kehittämä *Game of Life*, joka toimii kaksiulotteisessa ruudukossa. Solutyyppejä on kaksi – elävä ja kuollut – ja kukin solu muuttuu eläväksi tai kuolleeksi riippuen elävien naapurisolujensa määrästä. Life on osoitettu Turing-täydelliseksi, mutta logiikan rakentaminen sillä vaatii isoja, tarkkaan nyhverrettyjä alkutiloja. Helpommalla pääsee *Wireworldissa*, joka on erityisesti suunniteltu digitaalisten piirien rakentamiseen. Wireworld-maailmaan on rakennettu kokonaisia toimivia tietokoneitakin.

Pelkistetyimpiä soluautomaatteja ovat yksiulotteiset binääriautomaatit, joissa solun tulevaisuus riippuu siitä itsestään ja sen kahdesta naapurista. Näitä niin sanottuja alkeissoluautomaatteja on yhteensä 256. Automaatti 110 käyttäytyy mielenkiintoisesti – satunnaisellakin alkutilalla sen maailmaan nimittäin ilmestyy eri suuntiin eteneviä ”hiukkasia”, jotka toisiinsa törmätessään muodostavat toisenlaisia hiukkasia. Satakymppi on onnistuttu osoittamaan Turing-täydelliseksi toteuttamalla sillä syklinen tagijärjestelmä. Sen käyttö ohjelmointiin vaatii kuitenkin jo melkoista akrobatiaa.

Korvauskoneiden kutsuminen tietokoneiksi tai ohjelmointikieliksi alkaa olla jo harhaanjohtavaa. Pikemminkin ne ovat virtuaalimaailmoja, jotka toimivat omilla fysiikan laeillaan. Mikäli maailmassa on tarpeeksi tilaa ja sen lait ovat riittävän monipuoliset ja luotettavat, siihen pystyy rakentamaan ohjelmoitavia tietokoneita.

Entäpä pelimaailmat?

Monien tietokonepelien virtuaalimaailmat ovat Turing-täydellisiä. Esimerkiksi *Minecraft* toimii monimutkaisen soluautomaatin tapaan, ja sen palikka-valikoimaan jopa kuuluu varta vasten automaattisten mekanismien tekoon tarkoitettuja elementtejä. Harrastajat ovat toteuttaneet Minecraftiin lukuisia tietokoneita, joista hulppeimmat ajavat jopa Forth-tulkkia.

Varhaisempi esimerkki soluautomaattimaisesta pelistä on Boulder Dash, jonka Turing-täydellisyys ei ole mitenkään itsestään selvää. Logiikan rakentaminen saattaa kuitenkin olla mahdollista toisiinsa törmäilevien perhosten ja karpästen avulla.

Pelimailman ei tietenkään tarvitse olla soluautomaatti. Esimerkiksi *Incredible Machinen* fysiikkamekanis-

mit tai *Transport Tycoonin* rautatiet opastamiseen mahdollistavat logiikka-piirien rakentamisen. Jopa *Magic: The Gathering* -korttipeli on onnistuttu osoittamaan Turing-täydelliseksi.

Paluu todellisuuteen

Maaillamme on täynnä Turing-täydellisiä järjestelmiä. Näitä ovat paitsi tietotekniset laitteet ja useiden ohjelmien sisäiset mekanismit, myös monet ihmismielen pyörittämät matemaattisluonteiset rakenteet. Koska rakenteen ei tarvitse olla kovinkaan monimutkainen ollakseen Turing-täydellinen, voi sellaisia syntyä jopa epähuomiossa – ja on ehkä syntynytkin jo satojen tai tuhansien vuosien ajan. Muinaisena esimerkkinä mainittakoon vaikkapa sanskritin kielioppi. Yksinkertaiset ja epähuomiossa syntyneet Turing-täydelliset järjestelmät ovat kuitenkin useimmiten tervakuoppia.

Myös omaa maailmankaikkeuttamme voi pitää Turingin tervakuoppana. Vaikeasti hallittavine luonnonlakeineen se on varsin ohjelmoijavivahamielinen, mutta teknologisia tasoja toisen päälle kasaamalla olemme saaneet rakennettua sinne tietokoneita – erikoistuneita sopukoita, joissa ohjelmointi on ihmisen käsitykskyvylle mahdollista.

Esoteerisia ohjelmointikieliä ja Turingin tervakuoppia on helppo pitää hyödyttömänä ja todellisuuspakoisena ajanvietteenä. Tavallaan ne kuitenkin palauttavat meidät maan pinnalle – niihin pohjamutiin, joista päivittäin käyttämämme tekniikka rakentuu. Parhaimmillaan ne siis opettavat ymmärtämään tietotekniikan ja ohjelmoinnin olemusta uusilla ja mielenkiintoisilla tavoilla. 🐙



Minecraftilla toteutetun Hack III -tietokoneen suoritin. Kuva: Austin Mills.



Salattua sokeria – munajahdissa ohjelmistoissa

Jäitkö pääsiäisenä ilman suklaamunia yllätyksineen? Näitä pääsiäisyllätyksiä ei valitettavasti ole haudattu suklaaseen. Tietokoneohjelmaan piilotetulla pääsiäismunalla ei ole mitään tekemistä pääsiäisen tai edes munan kanssa.

Teksti: Markus Kuula Kuva: Wikimedia Commons

Pääsiäismunaksi (easter egg) tai jengiruuduksi (gang screen) kutsutaan ohjelmaan piilotettua osuutta, joka näyttää ohjelman tekijöiden nimet tai jonkin muun peliin tai ohjelmaan liittymättömän, piilotetun jipon. Tämän seurana voi olla animaatio tai kuva.

Jengiruudun löytäminen on tehty tarkoituksella vaikeaksi, sillä tällöin käyttäjän voi olettaa kuuluvan ”sisäpiiriin”. Mahdollisen yllätyksen voi löytää vain joko lukemalla tarkkaan ohjeita, kuuntelemalla viidakkorumpua tai itsepintaisesti kokeilemalla.

Yllätys voi sijaita muuallakin kuin ohjelmassa. Se voi piillä esimerkiksi asennuslevylle piilotetussa tiedostossa. Sen ei tarvitse sisältää riviäkään koodia, vaan yllätys voi olla myös kuva, jossa ohjelman tekijöiden nimet on listattu vaikkapa rantamaiseman viereen. Tätä kuvaa ei asenneta koneelle, vaan sen löytää vain asennuslevyä penkomalla.

Tekijät esille

Miksi kukaan käyttäisi aikaa tällaisen yllätyksen väkertämiseen? Syitä on useita, mutta tärkein niistä on yleisyys.

Ohjelman tekijät eivät saa välttämättä näyttävää tunnustusta sen eteen puseretuista työtunneista, stressistä ja kyyneleistä. Siinä missä elokuvien alussa saatetaan näyttää ohjaajan ja pääosan esittäjien nimet suurilla kirjaimilla, niin ohjelmoijat ja muut projektia työstäneet näkyvät yleensä vasta elokuvan tai pelin lopputeksteissä.

Oman hännän nostamisen lisäksi tällaisten yllätysten tekeminen on myös hauskaa, ainakin hauskenpää kuin bugien metsästäminen.

Hyötyohjelmien lisäksi myös peleihin on aikojen saatossa koodattu erilaisia yllätyksiä tai sala-alueita. Niistä saattaa löytää vaikkapa tekijän tai tekijöiden puumerkin, mahdollisesti salanimen muodossa.

Esimerkiksi 80-luvun kolikko- ja videopeli Xevious antaa kunniaa Evezoolle, jos siirryt oikeaan reunaan ja painat pomminäppäintä oikeassa paikassa. Evezoo End on tietenkin Masanobu Endoh, eli Xeviousin suunnittelija. Xeviousin sisältä löytyy muitakin tekijöiden nimiä, kuten Kei Cross (Kazuo Kurosu) ja Uhohoy Ohnogi (Nobuyuki Ohnogi).

Huijauksia tai huijauskoodeja ei siten muuten lasketa yllätyksiksi.

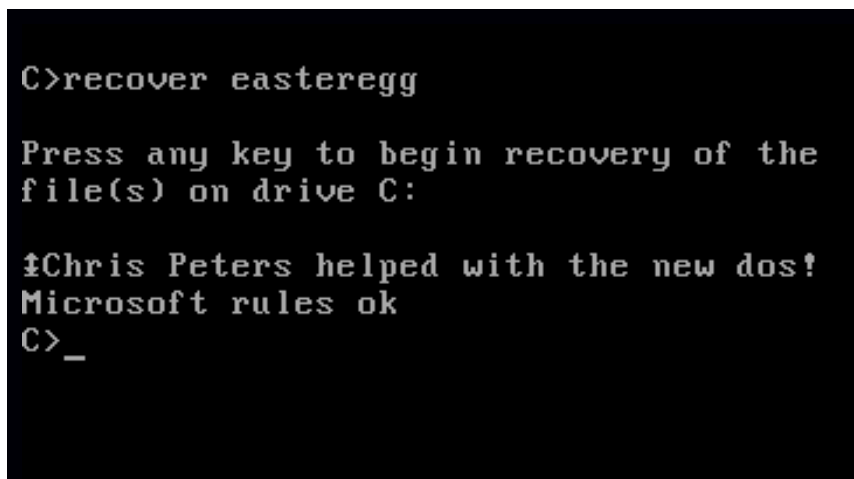
Särkynyt muna

Piiloruutujen tekeminen on nytemmin vähentynyt. Moni yritys suhtautuu nihkeästi salattujen sisällön lisäämiseen sovelluksiinsa. Yksi syy on tietenkin hukattu työaika. Jos jollakulla on aikaa piilottaa ohjelmaan jotain, on hänellä silloin liikaa aikaa. Lisää tuottavuutta tai kilometritehtaalle! Varsinkin nykyään on helppo ymmärtää, etteivät 50 sentin tuntipalkalla Aasiassa tai Intiassa työskentelevät bittinerot hukkaa aikaansa moisiin turhuuksiin.

Toisaalta perinteinen tarve easter eggien tekemiseen on vähentynyt. Etenkin pelialalla ohjaajien ja suunnittelijoiden lisäksi myös ohjelmoijat, graafikot ja muutkin tiimiläiset saavat kiitosta työstään ja nimensä lopputeksteihin. Niinpä omakehua ei tarvitse enää piilottaa koodin uumeniin.

Yllätysten vähentymiseen löytyy myös periaatteellisempi syy. Asiakkaat saattavat tykätä kyttyrää, jos törmäävät ohjelmassa salaruutuun tai muuhun yllätykseen. Tämä viestii siitä, että ohjelmassa on testaamatonta koodia. Ainakaan sitä ei ole testattu niin laajasti kuin olisi tarpeen.

Eivätkä asiakkaat yleensäkään pidä ohjelmaan livautetuista ylimääräisistä



DOS 2



Windows 3.1

asioista, oli niistä sitten maksettu vai ei. Tästä johtuen esimerkiksi Microsoftin kanta salaruutujen tai -sisältöjen lisäämiseen on jyrkkä ei.

Kettuilua ja romantiikkaa

Vanhoissa Excel-ohjelmissa saattoi olla kehitysryhmän nimilistan sijaan esimerkiksi patkä Doom-peliä tai animaatio, jossa kilpailevan yrityksen, Lotuksen, logo levittää ötököitä ympäriinsä ja Excelin logo häätää ne tiehensä. Samantyyppinen animaatio oli myös kohdistettu Wordin kilpailijaan, WordPerfectiin, tällä kertaa pirun muodossa.

Yllätykset ovat kuitenkin hauska tapa leikitellä, joten ne ovat katoamisen sijaan enemmänkin muuttaneet muotoaan. Äskettäin löydettiinkin seitsemän vuoden etsimisen jälkeen

Halo 3 -pelin viimeinen easter egg, jonka olemassaolosta useampikin pelitalon edustaja oli vihjaillut. Se osoittautui pelin latausruudusta 25.12. oikealla peliohjaimen käsittelyllä avautuvaksi syntymäpäiväonnitteluksi pääohjelmoijan vaimolle. Romantiikka elää!

Tunnettuja PC-piiloruutuja

Hyötyohjelmien ja pelien lisäksi myös käyttöjärjestelmistä löytyy piilotekstejä tai -ruutuja. Ohessa joitakin tunnetuimpia PC-maailman yllätyksiä.

DOS 2

RECOVER-ohjelma ilmoittaa Microsoftin erinomaisuudesta ja siitä, että "Chris Peters helped with the new dos!" Viestin näkee, kun painaa Control+W ohjelman odottaessa kuittausta.

Windows 3.1

Valitse Järjestelmänhallinnasta Ohje → Tietoja ja pidä Shift- ja Control-näppäimet pohjassa. Napsauta kahdesti dialogin vasemmassa ylä laidassa olevaa Windows-logoa. Ensimmäisellä kerralla ei tapahdu yhtikäs mitään. Sulje ikkuna ja avaa se uudestaan. Toisella kerralla tulee esille liehuva lippu ja teksti, jossa kerrotaan kehitystiimin raataneen niska limassa, että Windows tulisi uunista ulos. Kolmannella kerralla ruudulle ilmestyvät tekijöiden nimet, joita saattaa kuva Bill Gatesista, Steve Ballmerista, Brad Silverbergista tai nallesta.

OS/2

Käynnistämällä piilotetun AAAAAA.EXE-tiedoston näkee kuvan palmurannasta tekijöiden nimien kera. Kuva ilmestyy myös painamalla yhtä aikaa Control+Alt+Shift ja O-näppäimiä. Koskee sekä 2- että 4-versioita, joskin salaruudun kuvitus on hieman erilainen. Joistakin painoksista tarvittavat tiedostot kuitenkin tilasyistä puuttuivat.

Muna pystyy

Yllätyksen laatiminen omaan ohjelmaan on helppoa ja vain mielikuvitus on rajana. Tyypillisesti salaruutu aktivoidaan painamalla määrättyä näppäinyhdistelmää oikeassa tilanteessa. Varsinainen ongelma on kertoa salaruudusta. Jos se mainitaan näkyvästi, niin koko yllätyksen ajatus menee metsään. Senhän pitäisi kuitenkin olla mahdollisimman salainen.

Parempi ratkaisu on piilottaa vihje salaruudusta jonnekin ohjekirjan sopukoihin tai mukana tulevaan dokumentaatioon. Esimerkiksi suomalaisen kaivospommituspeli Mine Bombers sisälsi ruudun, johon oli lisätty pelin huijauskoodit. Huijauskoodit näyttävän sivun avausohje oli kuitenkin kätkeyty ohjekirjan uumeniin. Ruudusta voi myös vihjata jossain keskusteluryhmässä tai laitat kotisivuillesi maininnan, jonka poistat kuukauden päästä toivoen, että puskaradio on jo ehtinyt käydä paikalla.

Yllätyksiä peleissä

Teksti: Jukka O. Kauppinen

Kaikkien aikojen ensimmäiset videopeleistä löydetty pääsiäisyllätykset löytyvät peleistä *Video Whizball* (Fairchild, Fairchild Channel F, 1978) ja *Adventure* (Atari, Atari 2600, 1979). Video Whizballissa pelaaja löysi ohjelmoija Bradley Reid-Selthin sukunimen, Adventuresta puolestaan löytyi kokonainen salahuone, jossa leijui teksti ”Created by Warren Robinett”.

Pachinko-pelissä (Nintendo NES, Toshiba EMI, 1985) pelin ohjelmoija käytti kokonaiset viisi prosenttia pelimoduulin ROM-muistin kapasiteetista viestiin, jossa tämä räjäsi työpaikkansa esimiehistä.

Nintendo järjesti vuonna 1990 kilpailun, jonka voittaja sai nimensä seuraavaan *Zelda*-peliin. *The Legend of Zelda: A Link to the Past* -pelissä onkin huone, jonka löytäjä näkee viestin ”My name is Chris Houlihan. This is my top secret room. Keep it between us, okay?”

Wing Commander III (PC, Origin Systems, 1994) piilotetuissa, vaihtoehtoisissa lopputekstien viestit vaihtelevat äidille lähetetyistä terveisistä pelinkehitystä vauhdittaneiden levyjen, elokuvien ja tv-sarjojen listauksiin – niissä jopa lähetetään terveisiä Suomeen Future Crew -demoryhmälle. Future Crew näet oli tehnyt mullistavat PC-demonsa juuri parhaimman pelikehitysvaiheen keskellä vuosina 1992 ja 1993.

Simcopter (PC, Maxis/EA, 1996) – ohjelmoija piilotti peliin miespuolisia paidattomia, toisiaan hinkkaavia bimboja, joilla oli hohtavat nannit. Miesbimbojen piti tulla esiin vain tiettyinä päivinä, mutta peliin jääneen bugin takia niitä esiintyikin usein satapäisinä laumoina.

Grand Theft Auto: San Andreas (multi, Rockstar Games, 2004) sisälsi rakettirepun, jolla pystyi vaikkapa lentämään valtavan sillan päälle. Siellä ei tosin ollut mitään muuta kuin viesti: ”Täällä ei ole pääsiäisyllätyksiä. Menkää pois.” San Andreas sisälsi tosin myös jotain aivan muuta, mistä seurasi niin kutsuttu Hot Coffee -skandaali. Tämä ei kuitenkaan ollut pääsiäisyllä-

tys, vaan paha, paha moka.

Valven pelit ovat täynnä yllätyksiä, joista etenkin puutarhatonttu Gnome Chompski jaksaa tallustaa urheasti pelistä toiseen. Tonttu on nähty muun muassa *Half-Life 2*:ssa ja *Left 4 Dead 2*:ssa.

Kolmessa *Ratchet & Clank* -pelissä (Going Commando, Up Your Arsenal ja A Crack in Time) on Insomniac Museum -salahuone, jonne on laitettu peliin kehiteltyjen mutta toteutumattomien ideoiden luurankoja. Huoneeseen pääsee vain yöllä kello 03–04 välillä.

Yllätyksiä harrastetaan yhä edelleen, mutta niiden muoto on muuttanut muotoaan ja muuttunut jopa järjestelmälliseksi. Myös piilojuttujen kaivaminen on muuttunut lähes peliarkeologiaksi, sillä *Halo 3* -räiskintäpelin viimeinen salaisuus paljastui vasta seitsemän etsinnän jälkeen, kehittäjien vihjeistä huolimatta. Digital Illusions puolestaan rakensi kokonaisen vuoden ajan varsinaista salaisuuksien sarjaa, sillä tarkkavainuiset löysivät *Battlefield 4* -pelin neljästä lisäosasta hajanaisia dokumentteja, esineitä ja vihjeitä, joiden avulla neljän pelaajan moninpeli-ryhmä pääsi aiemmin lukittuun salahuoneeseen. Jos selvisi hissimatkaista hengissä. *Grand Theft Auto V* on puolestaan niin täynnä piilotettuja jippoja ja yllätyksiä, että niistä voisi tehdä kokonaisen kirjan.

WarBirds-lentosimulaattorin tekijät uunottivat pelaajiaan suorastaan järjestelmällisesti. Pelin varhaisesta 1.11-versiosta löytyi muun muassa Godzillasta tehty sfinksi autiomaasta – ja uimasta valtameressä. Myöhemmin pelin maailmankartan äärimmäiseltä ulkoreunalta löytyi amerikkalaisesta tv-sarjasta tuttu Gilligan’s Island ja Heinkel 111 -pommikonetta lentävä Han Solo. 2.x-versiossa nettilentäjiä äimistyi neljän kilometrin korkeudella leijuva dinosaurusaari. 🦖



Chris voitti Nintendon kilpailun – ja ikuistui *Zelda*-seikkailuun.



Gnome Chompski on seikkaillut niin monessa paikassa, että hänestä on tullut omanlaisensa ilmiö.



WarBirds-lentosimulaattorin eri versioista löytyy erilaisia yllätyksiä.



Yllätys ilman yllätystä. Mene pois, Rockstar Games komentaa.



Video Whizball ja *Adventure* (kuvassa) sisälsivät maailman ensimmäiset videopelien pääsiäisyllätykset.



Paras konsoli ei koskaan voita

Moni meistä on tempaista vastentahtoisesti sotaan. Moni on osallistunut niihin aktiivisesti, ja särkenyt niin sydämiä kuin mieliä julmilla sanoillaan. Konesodat. Käyttöjärjestelmäsodat. Ja nyt taas kerran, konsolisodat.

Jukka O. Kauppinen

Sodan roihu leimahti uuteen liekkiin, kun uusin pelikonsolisukupolvi saatiin markkinoille. Playstation 4, Xbox One ja pienemmässä määrin Wii U taistelevat pelaajien sieluista ja markkinaosuuksista, samalla kun nettifoorumeilla käydään ankaria konsolisotia.

Joskus foorumeilla arvostetaan enemmän liekehtiviä tunteenpurkauksia kuin kylmiä faktoja. Samalla joku innokkaimmista satureista pureutuu tarkasti teknisiin yksityiskohtiin, vertaillen laitteiden ominaisuuksia uskottomalla pieteetillä. Mikään detalji ei ole liian pieni nostettavaksi suurenuslasin alle, kun ristiretkeläiset yrittävät kääntää sodan kulkua oman kapitalistisen kansainvälisen suuryrityksensä tuotteelle suosiolliseksi.

Moni ristiretkeläinen ei kuitenkaan ymmärrä erästä tärkeää asiaa. Vaikka teknisistä eroista väittäisi miten tikkaa, sotaa ei voiteta tehokkaammalla konsolilla. Sota voitetaan peleillä, markkinoinnilla ja propagandalla.

Armeija marssii vatsallaan, Napoleon sanoi. Konsolisoturiarmeija marssii softallaan, sanoo Jukka.

Onko yhtään konsolisotaa voitettu tehokkaammalla konsolilla? Ei.

Ensimmäinen sukupolvi oli sentään varsin selkeä tapaus, sillä Magnavox Odyssey oli ainoa laatuaan. Toinen sukupolvi oli jo mutkikkaampi, sillä Atari 2600 hallitsi kaikesta alkeellisuudestaan huolimatta markkinoita suvereenisti pelibisneksen toiseen romahdukseen, vuoteen 1983 saakka. Esimerkiksi teknisesti täysin ylivoimaiset Intellivision- ja Colecovision-konsolit jäivät täysin Atarin tallomiksi.

Kolmannen konsolisukupolven potkaisu vauhtiin Nintendon NES – ja NES sen myös päätti. Segan Master System ja Atarin 7800 olivat parempia koneita, mutta entä sitten? NES hallitsi. Neljäs sukupolvi oli jo tiukempi. Segan Megadrive ja Nintendon SNES olivat loppujen lopuksi aika tasaväkisiä, mutta paras kone ei voittanut tälläkään kertaa. Moniko on edes kuullut PC Enginestä?

Viides sukupolvi meni kaupallisesti 100–1 Playstationille, mutta jos oikein tarkasti mietitään, niin eihän PSX ollut koneena edes kummoinen. Saturnin se pieksi menen tullen, mutta Nintendo 64 ja 3DO olivat koneina kovempia, eikä Atarin tähdenlennoksi jäänyttä Jaguariakaan sovi väheksyä. Mutta Sonyta tajusi hypen ja markkinoinnin voiman, tehden pelaamisesta ja Playstationista muodikkaita menevän ja nuoren kansan ilmiötä. Tällä oivalluksella tehtiin pelaamisen historiaa.

Entäpä Dreamcastin, Gamecuben ja Xboxin rintama PS2:ta vastaan? Teknisesti PS2 kyyditti Dreamcastia, mutta Xbox oli tekniikaltaan ja toiminnaltaan kovempi kone. Muistatte kyllä kuka tämän voitti. Ja kun seuraavassa erässä Xbox 360 ylsi tasapisteisiin Playstation 3:n kanssa, niin sukupolven voittajaksi nousi silti Nintendon naurettava Wii. Naurettava – mutta innovatiivinen, hauska ja omaperäinen. Pilkkattu liikeohjaus olikin juttu, jolla voitettiin mediakentällä pelaajiksi täysin uudenlaisia kohdeyleisöjä – ja konsolisoturit jäivät haukkomaan henkeään.

Miksi vaahtoon näin? No, koska ehdin elää jo monta onnellista vuotta omassa kuplassani, kun konsolisodat tuntuivat jo menneiden aikojen muottilmiöiltä. Tai sitten onnistuin pysyttelemään sivussa soturien kansoittamista viesti- ja kommenttiketjuista. Sitten kirjoitin Xbox Onesta ja hämmästyin. Kommentteissa jaariteltiin muistipii-reistä sitä ja tätä, näytönohjaimista, 10 prosentin teholisistä, väylistä ja arkitekhtuurista. Täysin merkityksetöntä nillitystä asioista, joilla ei ole loppumetreillä mitään merkitystä. Asioita, jotka eivät todellakaan vaikuta siihen, mihin suuri yleisö rahansa laittaa.

Suuren yleisön sielujen voittamiseen tarvitaan kunnon hypeä, onnistunutta markkinointia ja vasta sitten oikeasti hyviä pelejä. Ei parempaa tekniikkaa. Paras konsoli ei ole koskaan voittanut. Siksi sota onkin nyt niin kiintoisaa, sillä koskaan ennen markkinoiden johtavat konsolit eivät ole olleet niin identtisiä tekniikaltaan, niin... PC:tä. 🐘

Jukka O. on raivannut tiensä käyttöjärjestelmä-, konsoli- ja konesotien halki seesteiseen olotilaan, jossa turhasta kinastelusta ei tarvitse enää välittää.



Kuvat: Mitol Meerna ja Manu Pärssinen

Palautteen antamisessa ei kannata kit-



sastella. Palaute kertoo että mitä tapahtui, kenelle se tapahtui, miten ja miksi se tapahtui. Kaikella palautteella tulisi olla pelillinen ja mieluiten myös pelimekaaninen tarkoitus.

Koskaan ei voi antaa liikaa palautetta, mutta on erittäin helppoa antaa väärää tai huonoa palautetta. Jos esimerkiksi aiemmin esitetystä FPS-pelistä jätettäisiin pois räjähdyssefekti, sirpaleet ja kipinät, ei pelaaja kovin herkästi opi, että räjähdyksellä on myös vaikutusalue. Tämä on esimerkiksi huonosta palautteesta. Jos räjähdyssefekti korvattaisiin kahden sekunnin mittaisella, koko ruudun täyttävällä kirkkaalla välähdyksellä, pelaaminen vaikeutuisi ja pelaaja todennäköisesti menisi kuolemaan juoksennellessaan sokkona ympäriinsä. Tämäkin on huonoa palautteenantoa.

Jos taas räjähdys korvattaisiin kahden sekunnin ajan hitaasti kasvavalla tulipallolla, saisi pelaaja sen kuvan että tulipallo tekee vahinkoa kun sitä koskettaa. Tämä on esimerkki väärästä palautteesta: räjähdys teki isolle alueelle välitöntä vahinkoa, eikä hitaasti kasvavalle alueelle usean sekunnin ajan kestävää vahinkoa.

Kannattaa aina miettiä, kuinka monella tavalla pelin tapahtumat voi kommunikoida pelaajalle. Kannattaa yhdistää monta eri tapaa, mikäli ne eivät häiritse toisiaan. Esimerkkinä on

luodin osuminen viholliseen – lentävä veri kertoo, että vihollinen sai osuman. Viholliseen voi myös lisätä nytkähdysanimaation, mikä tukee vaikutelmaa. Vihollisen karjahdusääni on myös kuin kotonaan kahden edellisen palautteen kanssa.

Visuaalinen kieli

Samalta näyttävien asioiden tulisi aina käyttäytyä samalla tavalla. Tämä mahdollistaa pelin sääntöjen oppimisen, mikä on hauskuuden kannalta oleellista. Esimerkiksi Super Marion kentissä tallustavat kilpikonnat käyttäytyvät aina täysin ennakoitavalla tavalla. Ne kävelevät tasaista vauhtia eteenpäin kunnes törmäävät esteeseen tai pudotukseen, jolloin ne vaihtavat suuntaansa. Kun pelaaja näkee kilpikonnan, hän ajattelee: ”Se vain liikkuu ja kääntyy.”

Jos edes yksi näistä kilpikonnista yllättäen päättäisi lähteä jahtaamaan Mariota ja opettelisi hyppimään tasolta toiselle, mielikuva kilpikonnan käyttäytymismallista hajoaisi sotkiksi pelikokemusta. Kun pelaaja seuraavan kerran näkisi kilpikonnan, hän ajattelisi: ”Onkohan tuokin jahtaava kilppari? Mitähän hittoa se tekee tällä kertaa?” Pelaaja ei enää voisi luottaa näkemäänsä, eikä kykenisi ennakoimaan mitä seuraavan kilpikonnan kohdalla tapahtuu. Tällaisen tilanteen voi korjata muuttamalla jahtaavan hyppykilpikonnan grafiikat mahdollisimman

erilaisiksi normaalista kilpikonnasta. Toisin sanoen: keskenään eri tavalla käyttäytyvien elementtien tulisi näyttää mahdollisimman erilaisilta.

Yksi yleinen ja tehokas tapa merkitä samalla tavalla käyttäytyvät elementit on tehdä niistä samanvärisiä. Tätä kutsutaan värikoodaukseksi. Jos pelaaja esimerkiksi ampuu jään väristä esinettä ja se hajoaa, parin vastaavan kokemuksen jälkeen pelaaja yhdistää mielessään jään väriset esineet hajoaviksi esineiksi. Pelaaja oppii nopeasti luottamaan tähän havaitsemaansa ja oppimaansa sääntöön.

Näitä ohjeita seuraamalla päästään jo pitkälle palautteen antamisessa. Tärkeintä on, että pelaaja oppii palautteen avulla tulkitsemaan pelin tapahtumia kehittäjän haluamalla tavalla, ja että virheellisiä viestejä antava palaute vältetään. 🗡️



Irti vanhasta vihasta

Avointen ohjelmistojen maailmassa on väännetty kovasti kättä BSD- ja GNU-lisenssien välillä. Näkemykset ovat olleet useammin törmäyskurssilla kuin yhteistyössä, mutta harmonia on kuitenkin mahdollista.

Teksti: Jarto Nieminen Kuvat: Mikko O. Torvinen

Seuraa kasvutarina nuoren bittinikkarin alkutaipaleelta. Eletään alkusyksyä vuonna 2000. Olen 15-vuotias. Painin palomuurin rakentamisen perusperiaatteiden kanssa, ja käsilläni on Linux-järjestelmä, jota kaveripiirissäni eniten käytetään: Debian GNU/Linux. Tunnen Debianin paketinhallintajärjestelmän, GNU:n käyttäjäympäristön ja Linux-ytimen verkonhallinnan peruskäsitteet mielestäni hyvin. Mutta...

Syntax error

Linuxin tuki vanhalle palomuurijärjestelmälle lakkaa ja sen korvaa uusi, täysin remontoitu järjestelmä. Uudelleen kirjoituksen tuloksena on otettu käyttöön aivan uudenlainen syntaksi, jonka koen heti vaikeaselkoiseksi. Yritän epätoivoisesti saada aikaiseksi johdonmukaista, vanhan palomuurini toiminnot toteuttavaa järjestelmää.

Seuraan pikaohjeita ja saatuani niissä olevan esimerkin toimimaan haluan päästä käsiksi uuden järjestelmän kehittyneempiin ominaisuuksiin. Etsin

syvällisempää ymmärrystä ja turvautun Debianin mukana tulevaan GNU-ympäristön omaan dokumentaatioon, koska pikaohje näin kehottaa.

Turhaudun varsin perinpohjaisesti. Uuden järjestelmän dokumentaatio on pikaohjetta lukuun ottamatta pahasti kesken. Se muodostuu vaikeaselkoisista ja puutteellisista teksteistä, jotka on ripoteltu epäjohdonmukaisesti ympäri järjestelmää. Internet-haut tuottavat vain kopioita samoista teksteistä.

Totean itselleni, ettei tässä ole enää mitään järkeä. Ongelmaan on löydettyvä toinen ratkaisu. Jostain täytyy löytyä palomuuriratkaisu, joka on dokumentoitu tarpeeksi hyvin. Niin hyvin, ettei sen käyttöönotto vaadi mielivaltaisen syntaksin työstä arvailua.

Uusi uljas maailma

Käänän nopeasti katseeni kohti maailmaa, joka edustaa omassa kokeuspiirissäni uutta tulokasta. Kaveripiirissäni olen seurannut sivusta väitellyitä BSD- ja GNU/Linux-maailmojen eroista. Muistan, että yksi isompi BSD-käyttöjärjestelmäprojekti on nimeltään

NetBSD. Nimellä on lupaava kaiku. Mieltä lämmittää esimerkiksi se, että virallisten julkaisujen mukana on aina toimitettava selkokielineen ohjeistus jokaiselle ohjelmalle.

Lataan välittömästi NetBSD:n asennus-cd:n näköistiedoston ja poltan sen tyhjälle levyille. Pitkään kytenyt turhautumiseni GNU/Linux-maailman kaoottisuuteen purkautuu hallitsemattomana tunnereaktiona: en ainoastaan tuhoa palomuurikoneeni Linux-asennusta vaan katkaisen palasiksi kaikki nurkissani pyörivät vanhat Debianin asennuslevyt. Näin ei ole edes teoriasa mahdollista, että kodistani pääsisi internetiin ennen kuin NetBSD-pohjainen palomuuuri on pystyssä. Minulla ei ole pääsyä mihinkään muuhun dokumentaatioon kuin siihen, jonka NetBSD toimittaa käyttöjärjestelmäkelunsa mukana.

Seuraa pitkä ja intensiivinen yö. Tutustuessani tähän uuteen maailmaan havaitsen ilokseni, että NetBSD ei ole vain luontevampi käyttää vaan se myös pitää lupauksensa hyvin dokumentoidusta käyttöjärjestelmästä. Teksti on

hyvää englantia: se on lyhyttä ja ytimekästä ja sitä on tarjolla kokonaisvaltaisesti.

Aloitin pienistä: asetan järjestelmälle hostnimen, luon peruskäyttäjän, jolla on su-oikeudet (vanhakantainen turvarakenne) ja alan pystyttää palomuuria. Opin uuden järjestelmän perusteet nopeasti, ja vähän pidempijärteisen kehittämisen tuloksena syntyy palomuuriskripti, joka toimii omassa käytössäni aika pitkälti "asetta ja unohda" -periaatteella lähes kymmenen vuotta.

GNU-vihan aika

Myönteiset BSD-kokemukseni saivat kaltaiseni avoimen koodin puolesta-puhujan ylpistymään. Ryhdyin erilaiseksi nuoreksi ystäväpiirini GNU/Linux-lähettiläiden keskuudessa. Seurasin muutama vuosi melko intensiivistä "trueilua" eli jokseenkin tyhjänpäiväistä ideologista inttämistä eri järjestelmien ja etenkin ohjelmistolisenssien paremmuudesta.

Ryhdyin kantamaan BSD:n soitua syvällä tunteen palolla. Irrationaalinen ja vahva tunnereaktio oli rakentanut päähäni vahvan näkemyksen: "GNU-yhteisö on väärässä eikä sitä paitsi osaa edes kirjoittaa käyttöohjeita, saati koodata!" Hyökkäsin ajoittain erittäin aggressiivisesti ja pahansuovalla tavalla etenkin ystäväpiirini Debian-faneja vastaan.

Maailmojen erot

GNU-hankkeen yleiseen lisenssiin (GNU General Public License, GPL) sisältyy ajatus, että ohjelman käyttäjälle kuuluvat tietyt perusvapaudet ja -oikeudet. Lisenssi pyrkii estämään ohjelmistokehittäjiä loukkaamasta näitä oikeuksia.

Käytännössä tämä tarkoittaa sitä, että kaikista GPL-ohjelmista on oltava tarjolla täydellinen kääntämiseen soveltuva lähdekoodi, joka on myös kenen tahansa vapaasti muokattavissa. Mikäli ohjelmaa tai sen muunnelmää levitetään, on sen lähdekoodi julkaistava samalla lisenssillä, eikä sen levitykselle tai käytölle saa asettaa lisärajoituksia. Näin ohjelma ja sen uudet versiot säilyvät ikuisesti samalla tavalla vapaana kuin alkuperäisenkin julkaisu.

Vapauden voi kuitenkin määritellä toisellakin tavalla. BSD-lisenssi on

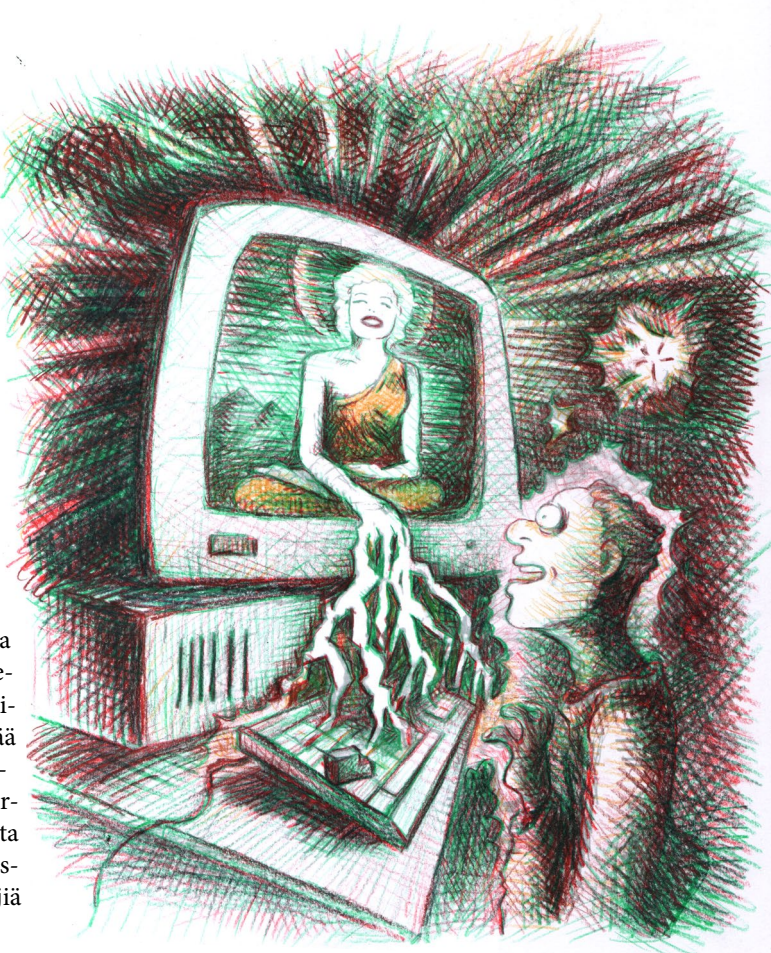
eräällä tavalla paljon vapaampi ja myös yksinkertaisempi. Ohjelman uudelleen julkaiseminen – muunnoksinkin tai ilman muunnoksia – on sallittua lähdekoodi- tai binäärimuodossa, kunhan seuraavat ehdot täyttyvät: Ohjelmiston lähdekoodin tai dokumentaation täytyy sisältää lisenssiteksti, jossa mainitaan alkuperäisen koodin tekijä. Lisenssi sisältää vastuuvapauslauskeen, jossa tekijä irtisanoutuu kaikesta vastuusta ohjelmistoa ja sen käyttäjiä kohtaan.

BSD-lisenssin vapaus tarkoittaa esimerkiksi sitä, että ohjelman saa julkaista myös ilman lähdekoodia. Lisenssi ei myöskään estä muunnellun ohjelmiston lisenssin vaihtamista. BSD-projektien koodia onkin hyödynnetty useissa kaupallisissa järjestelmissä. Microsoft lainaa tämän tästä BSD:n verkkokoodia paikataksensa omaansa, ja Mac OS X perustuu BSD:hen.

Valaistuminen

Mutta palataan vielä takaisin omaan tarinaani. Laajensin BSD-harrastustani perehtymällä myös perinteisiin Unix-työasemiin. Ensikosketus tuli Sun Microsystemsin SPARC- ja Motorola 68020-pohjaisten järjestelmien kautta. Otin käyttööni myös FreeBSD:n, joka osoittautui helppokäyttöiseksi. Samalla fundamentalistinen maailmankuvani vahvistui entisestään: GNU-hanke on tarpeeton ja turha. Se ei ansaitse edes olla olemassa.

Näin jatkui melko pitkään, kunnes eräänä päivänä koin uuden valaistumisen: olin purkanut BSD-järjestelmien ja niiden edustaman ideologian kautta jotakin lapsellista. Ymmärsin, etteivät ideologiat voineet olla mikään pyhä itsetarkoitus, vaan oli oltava järkevä syy sille, että eri organisaatiot käyttävät

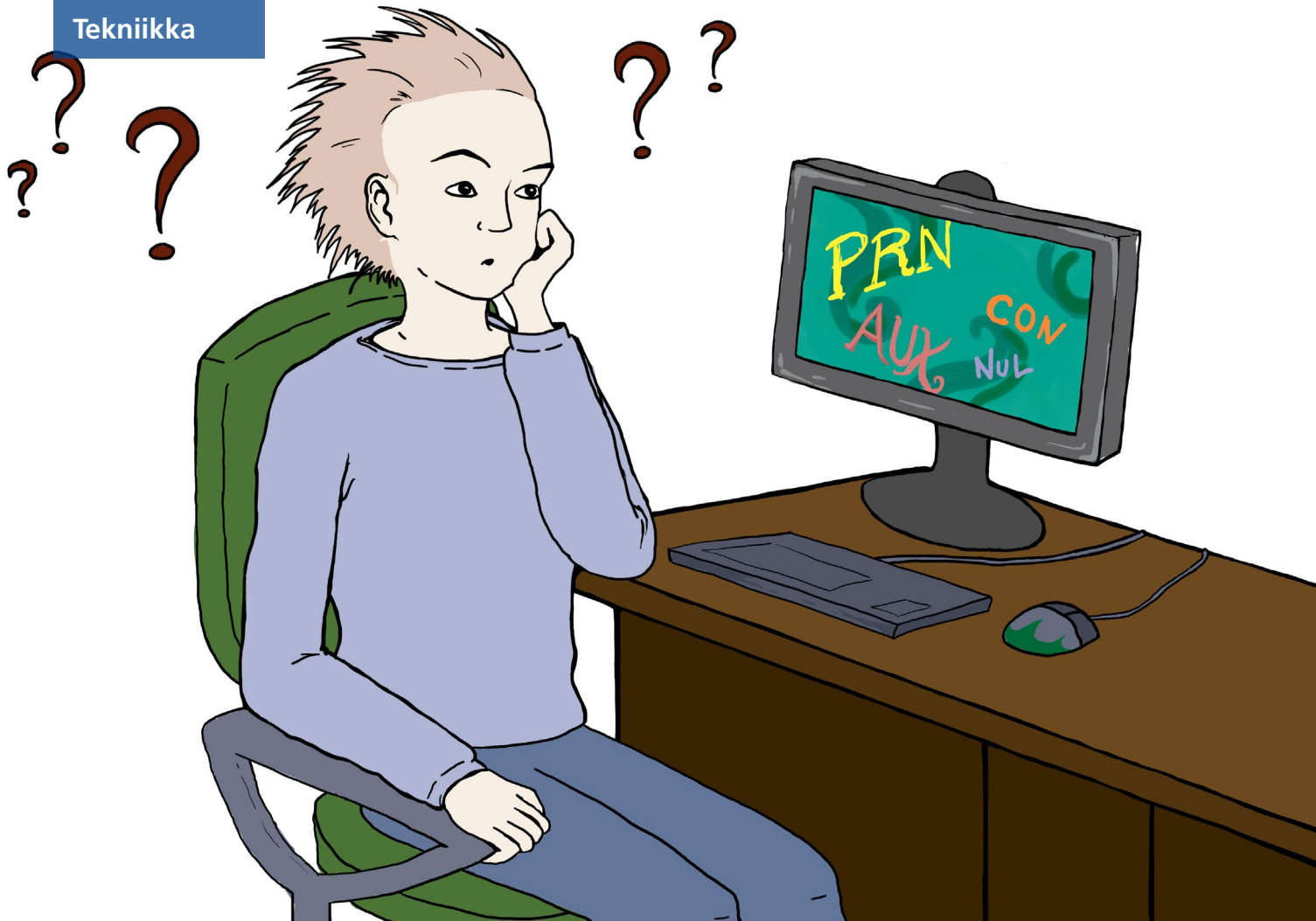


erilaisia järjestelmiä.

Valaistumiseni innoittamana aloin tutkia ympärilläni olevaa tietojärjestelmien kirjoa aivan uudella tavalla. Ajan myötä olin hankkinut itselleni erittäin laajan kokoelman eri prosessoriarkkitehtuureille perustuvia koneita. Sain rakennetuksi monimuotoisen tietojärjestelmän, joka hyödynsi yhteistä verkkoilevää ja käyttäjätietokantaa. Kovan työn jälkeen HP-UX, Solaris, NetBSD, Mac OS X, Mac OS X Server, Windows XP ja Debian GNU/Linux elivät keskenään sulassa sovussa.

Käytännössä suuri konekokoelma osoittautui vähän liian työlääksi ja monimutkaiseksi ylläpidettäväksi pelkästään harrastuspohjalta. Ehkä se oli kuitenkin tarpeellinen projekti siksi, että se symboloi omaa valaistumistani: erilaiset koneet ja käyttöjärjestelmät keskenään harmoniassa, samoin oma mieli.

Lopulta kyse on vain siitä, minkälainen työkalu kenenkin käteen ja mihinkin tehtävään parhaiten sopii. Eri järjestelmiin tutustumiselle on turha asettaa mitään keinotekoisia ideologisia, eettisiä tai moraalista estettä. Vanha kulunut sananlasku kaikista kukista ja niiden kukkimisesta sopii siis tähänkin. 🌸



Historiaa laitetiedostojen takana

Hyvä on olla, kun on nul.exe

Avaa Windowsin Muistio ja kirjoita jotain. Tallenna kirjoitelmasi sitten nimelle nul.txt. Eikö onnistu? Uusimmissakin käyttöjärjestelmissä on näet yhä edelleen mukana perintöä mikroikivikauden tietokoneilta, ja vaikka niille ei enää olekaan käyttöä, ei niistä päästä eroonkaan.

Teksti: Markus Kuula Kuvat: Teija Tuhkio, Mitol Meerna, Markus Kuula

Kotitietokoneiden esihistoriallisella kaudella, 70-luvun puolenvälin tietämällä, maailmaa hallitsivat 8-bit-tiset koneet, joissa muistia oli usein muutama kymmenen kilotavua. Koneiden yleisin käyttöjärjestelmä oli CP/M, Control Program for Microcomputers. Tarkkuusgrafiikkaa tai kiintolevyä ei normaalikäyttäjillä ollut ja levykkeidenkin koko oli 180 kilotavun tietämällä. Tiedostoja ei juuri tarvinnut järjestellä, joten hakemistojakaan ei tunnettu.

Käyttäjät ja ohjelmoijat saattoivat kuitenkin kommunikoida koneeseen liitettyjen lisälaitteiden kanssa. Niitä käsiteltiin samoin kuin

tiedostoja, jolloin ohjelman kannalta oli sama, kirjoitiko se tekstiä tulostimelle vai tekstitiedostoon. Tämä tehtiin varaamalla tietyt kirjainyhdistelmät laitenimiksi, jolloin esimerkiksi tulostinta pystyi hyödyntämään avaamalla tiedoston nimeltä "PRN". Laitetiedostoksi kutsuttuun tiedostoon kirjoitettiin normaalisti ja lopuksi se suljettiin. Sen jälkeen laitetiedostoon tallennettu teksti tulostui paperille.

Menneisyyden vangit

CP/M jäi unholaan, mutta Microsoft otti yhteensopivuussyistä samat laitenimet käyttöön MS-DOS-käyttöjärjestelmässään (Microsoft Disk Operating System).

Tuttuja laitenimiä nähtiin myös AmigaOS:ssä ja muissa CP/M:n jälkeisissä käyttöjärjestelmissä.

Ensimmäisessä DOS-versiossa käytettävät laitteet kovakoodattiin DOSin ytimeen. Ne olivat seuraavat :

- AUX
- PRN
- LST
- CON
- NUL

Myöhemmin laitenimiä lisättiin:

- COM1 – COM9
- LPT1 – LPT9
- CLOCK\$
- LST jätettiin pois turhana.

Laitenimet olivat eräänlaisia haamutiedostoja, jotka sijaitsivat jokaisessa hake-

mistossa – tämäkin yhteensopivuussyistä. MS-DOSin versiossa 2 oli tosin dokumentoimaton AVAILDEV-valitsin, jolla laitenimet pystyi pakottamaan /DEV-haamuhakemistoon. DOSin sisäisiä komentoja käytettäessä laitenimiä ei edelleenkään voinut käyttää tavallisten tiedostojen tavoin, joten tämä vipu siirtyi ajasta ikuisuuteen DOSin kolmosversion myötä.

Kiusa ja isompi kiusa

AUX viittaa oletusarvoisesti ensimmäiseen sarjaporttiin (COM1), PRN puolestaan rinnakkaisporttiin. CON tarkoittaa konsolia, eli näyttoa ja näppäimistöä. CONin toiminta riippuu siitä, ol-

laanko siihen kohdistamassa luku- vai kirjoitusoperaatio. NUL on musta aukko, eli sinne ohjatut tiedot ja tiedostot ikään kuin katoavat.

Myöhemmin mukaan tuli vielä CLOCK\$, joka hallitsee kellotoimintoja. Nimen lopussa olevan dollarimerkin arveltiin vähentävän sekaannusten vaaraa. Tämä on kuitenkin vain puoli totuutta. DOS-käyttöjärjestelmän kakkosversion eräs uusi piirre olivat näet ladattavat laiteohjaimet. Näiden erottamiseksi käyttöjärjestelmän ytimessä olevista kiinteistä laiteohjaimista määriteltiin ladattavien ohjainten nimen loppuun dollarimerkki. Kellon hallintatoiminnot tulivat käyttöjärjestelmään kiinteästi vasta kolmannen version myötä, joten sitä ennen ne oli hoidettu valmistajan omalla ajurilla.

Dollarimerkki paljastaa toisenkin asian: se oli tarkoitettu vain sisäiseen käyttöön. Kellon lukeminen käskyllä "type clock\$" ei tulostanut siistiä kellonaikaa ja päivämäärää vaan sotkua, joka oli kellonajan ja päivämäärän koodattu muoto. Kirjoittaminen päättyi vielä huonommin ja saattoi johtaa kokoonpanoasetusten täydelliseen sekoamiseen. Niinpä ilkeät raggarit saattoivat laittaa pakatun arkiston (.zip, .arc, .arj tai vastaava) sisälle CLOCK\$-nimisen tiedoston, joka lähetettiin pahaa-aavistamattomalle henkilölle. Erityisesti purkien ylläpitäjien piti seuloa

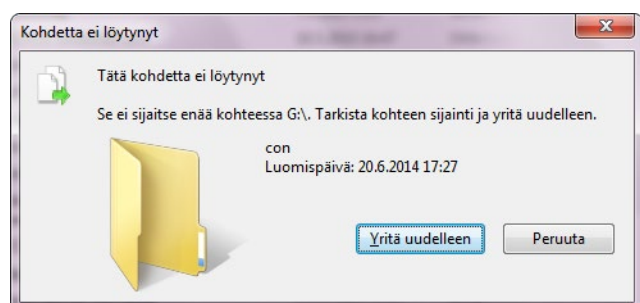
tarkkaan lähetetyt tiedostot tämänkaltaisten yllätysten varalta tai muuttaa laitenimi toiseksi peukaloimalla IO.SYS-tiedostoa.

Moderneja muinaisjäänteitä

Laitenimet ovat muinaisjäännöksiä, joka on kuitenkin mukana jopa uusimmissa Windows-versioissa. Huonon dokumentoinnin takia niitä voisi kutsua nykyään jopa bugeiksi. Jopa 64-bittinen Windows kieltäytyy talentamasta tiedostoa, jos sen nimiosa vastaa jotakin aiemman listauksen mukaista nimeä. Järkevämpää olisi ollut rajoittaa laitenimet vain komentorivillä käytettäviksi tai edes lisätä kytkin, jolla ne saa pois käytöstä. Ainakin henkilöt, joiden etunimi on Con, olisivat kiitollisia.

Pystyykö tiedoston tai hakemiston luomaan laitenimelle? Kyllä. Ei tosin suoraan Windowsin graafisen puolen ohjelmilla, vaan käynnistämällä komentorivin (cmd.exe). Hakemiston voi luoda komennolla "md \\.\g:\con" ja poistaa "rd \\.\g:\con".

Käyttämällä tällaista UNC-muotoista polkua pystyy kiertämään Windowsin tarkistuksia. Tämä on hyvä tietää, jos esimerkiksi FTP-ohjelmalla siirtää koneelle laitteen nimen mukaisen tiedoston, jota ei saa pois tavanomaisin keinoin. Unixissa laitenimet on toteutettu järkevämmiin ja ne sijaitsevat aina omassa hakemistossaan, kuten MS-DOS 2:ssa aikoinaan. 🦌



Reaktor HUNTING NEXT GENERATION CODERS



YOU GAME?

Apply at: reaktor.fi/careers



Palikoita ruudulle

Videopiiri kertoo kuvan

Tietokoneet ovat piirtäneet grafiikkaa näyttöruuduille jo 1950-luvulta asti. Grafiikan olemus riippuu pitkälti siitä, millainen näytönohjauslaitteisto sen esittää.

Teksti: Ville-Matias Heikkilä Kuvat: Tapio Lehtimäki ja Ville-Matias Heikkilä

Erilaisten tieto- ja pelikoneiden visuaaliseen antiin tutustumalla huomaa helposti, että eri laitteilla on erilaiset, laitteiston ominaisuuksista kumpuavat graafiset ilmeensä. Etenkin vanhat koneet on usein hyvinkin helppo tunnistaa värimaailmoistaan, graafisista rakennuspalikoistaan ja oikuiltelutaivoistaan. Kyse ei ole pelkästään koneen laskentakapasiteetista vaan sen persoonasta – siitä, millaisia luovia valintoja näytönohjauslogiikan suunnittelijat ovat tehneet.

Useimpia tässä artikkelissa kuvattuja näytönohjaustekniikoita ei enää tapaa uusissa koneissa, mutta ne ovat oleellinen osa monen klassisen koneen ominaisestetiikkaa.

Kaksi tapaa käyttää kuvaputkea

Ennen nestekidenäyttöjen voittokulkua tietokoneissa käytettiin kuvaputkia. Kuvaputken toiminta perustuu elektronisuihkuun, jota suunnataan sähkömagneettien avulla kuvapinnan eri osiin. Kuvapinnalla elektronisuihku

synnyttää valoa törmätessään fosforoivaan aineeseen.

Tunnetuin kuvaputken käyttötapaa on tuttu tv:stä: kuva jakautuu vaakatai pystysuoriin juoviin, jotka suihku käy läpi noin 50–100 kertaa sekunnissa. Kuvaputkea voidaan kuitenkin käyttää myös siten, että suihku muodostaa kuvapinnalla vapaita liikeratoja. Tämä tapa on tuttu oskilloskoopista.

Ensimmäiset tietokonenäytöt 1950-luvulla käyttivät oskilloskoopitekniikkaa, ja se oli tietokonegrafiikassa varsin yleinen vielä 1970-luvulla. Yksinkertaisimmillaan tietokone syöttää lukuja d/a-muuntimiin, jotka vaihtavat magneeteille meneviä jännitteitä. Siirtymät ovat yleensä melko äkkinäiset, jolloin kuva muodostuu irtopisteistä. Hieman edistyneemmät laitteistot tukevat hitaampia ja tasaisempia jännitemuutoksia, mikä mahdollistaa viivojen piirron. Vectrex-pelikonsoli on harvinainen esimerkki vektorinäytön käytöstä kotitietotekniikassa.

Vektoripiirtoon ei välttämättä tarvita erikoistunutta näytönohjauslogiikkaa, vaan tietokone voi siirrellä sädetä

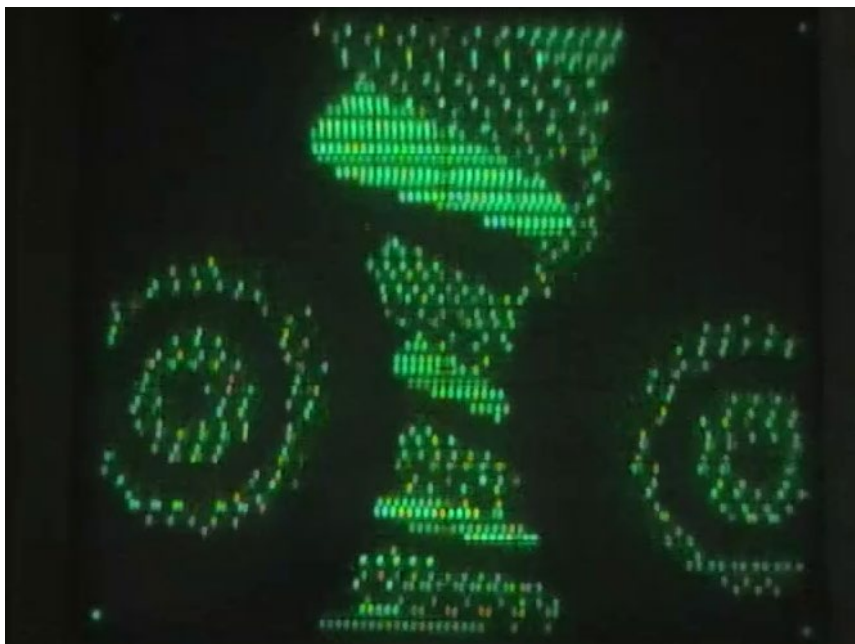
omaan tahtiinsa. Televisiotyyppinen rasteripiirto sen sijaan perustuu tiukkoihin ajoitusraameihin, joten erikoistunut videopiiri on käytännössä välttämätön.

Joka pikseli kartalle

Suoraviivaisimmillaan näytönohjauslogiikka tulkitsee näyttömuistin sisällön niin kutsuttuna bittikarttana. Näyttömuistia luetaan eteenpäin samaan tahtiin kuin elektronisuihku etenee, ja luetut bitit tulkitaan eri väreiksi tai kirkkausasteiksi. Pikseli voi siis olla esimerkiksi musta tai valkoinen riippuen siitä, onko sitä vastaava bitti ykkösen vai nolla.

Kun bittikarttagrafiikkaan halutaan enemmän kuin kaksi väriä, voidaan pikselikohtaista bittimäärää lisätä. Esimerkiksi kahdella bitillä per pikseli saadaan neljä väriä ja kahdeksalla bitillä 256. Tällöin kuvan tallentamiseen on kaksi päätapaa – yhtenäinen (chunky) ja kerroksittainen (planar).

Nykytietotekniikassa käytetään lähes pelkästään yhtenäistä tallennusta, jossa kaikki saman pikselin bitit ovat muistissa peräkkäin. Esimerkiksi



Vektorinäytön voi toteuttaa esimerkiksi kytkemällä äänikortin ulostulot oskilloskooppiin. TVT:n Youscope-demo.



Tarpeeksi nopea mikrokontrolleri ja tarpeeksi ovela koodi, niin erillistä näytönohjainta ei välttämättä tarvitse. Ruotsalaisen lft:n Craft-demo.

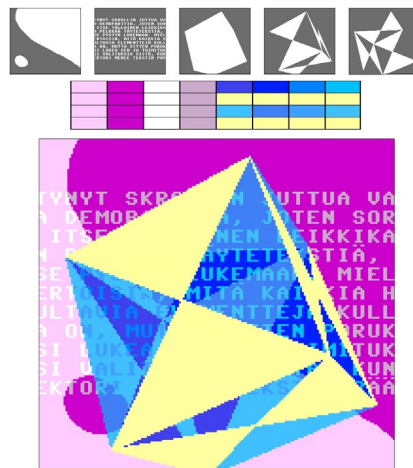
32-bittinen pikseli käyttää muistista neljä peräkkäistä tavua. Vanhemmat koneet lukevat näyttömuistia usein vakionopeudella, joten vaakaresoluutio on niissä kääntäen verrannollinen köntän pituuteen: Commodore 64:ssä kaksibittiset multicolor-pikselit ovat tuplasti niin leveitä kuin yksibittiset hires-pikselit.

Kerroksittaista tallennusta käytetään esimerkiksi Amigassa, Atari ST:ssä ja PC:n 16-väritiloissa. Siinä kuva jaetaan erillisiin tasoihin, jotka voi kuvitella vaikkapa läpinäkyviksi kalvoiksi,

jotka päällekkäin asetettuina muodostavat monivärisen kuvan. Kukin taso sisältää yhden bitin per pikseli, ja lopullisen pikselin värikoodi muodostuu samaan kohtaan sattuvista biteistä.

Graafisista tempuista esimerkiksi siluetit, parallaksit ja yksinkertaiset läpinäkyvyudet soveltuvat parhaiten kerroksittaisille grafiikkatiloille. Yhtenäinen tallennus on puolestaan edukaampi esimerkiksi teksturoidussa 3D-grafiikassa.

Pikselin väri voi riippua myös sen sijainnista. Apple II:n tarkkuusgrafiik-



Kerroksittaista näyttömuistia on usein hyödynnetty samaan tapaan kuin nykyisten kuvankäsittelyohjelmien kerroksia.

katila on periaatteessa kaksivärinen, mutta kirkas väri on joka toiselle pikselille vihreä ja joka toiselle purppura. Vierekkäisen vihreän ja purppuran pikselin värit sulautuvat NTSC-värijärjestelmässä valkoiseksi.

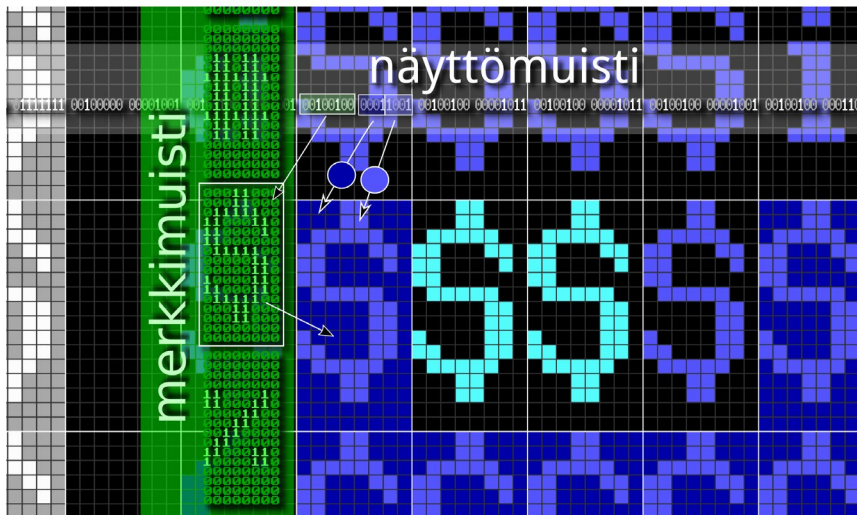
Liikennejärjestelyt

Alkeellisimmillaan videopiiri ei lue muistia itsenäisesti. Esimerkiksi Atari 2600 ja ZX81 tarvitsevat suorittimen apua kuvan näytöllä pitämiseen, ja siihen meneekin valtaosa konetehoista. Toista ääripäätä edustavat MSX ja NES, joissa videopiiri näyttömuisteen muodostaa itsenäisen nurkkauksensa, jonka toimintaan suoritin pääsee vaikuttamaan vain ruudunvirkistysten välissä. Tämä muodostaa nopealle grafiikalle pullonkaulan.

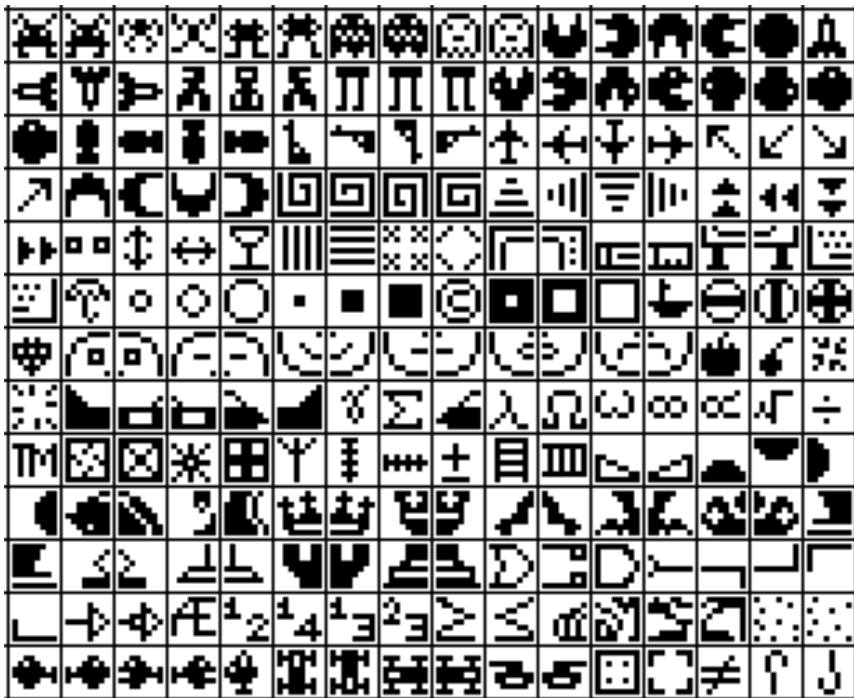
Välimuotona on Commodore 64:n ja Spectrumin ratkaisu, jossa videopiiri käyttää samaa keskusmuistia kuin suoritinkin. Tämä hidastaa jonkin verran suorittimen toimintaa, sillä videopiirillä on muistinkäsittelyssä etuajaoikeus.

Amigoissa on kaikkien piirien jakaman chip-muistin lisäksi fast-muistia, johon vain suoritin pääsee käsiksi. PC:n näyttömuisti on taasen perinteisesti ollut erillisellä näytönohjainkortilla, joskin AGP-kortit pystyvät käyttämään myös keskusmuistia sen jatkeena. Silicon Graphicsin O2-työasemassa puolestaan kaikki muisti on "chip-muistia", mutta ruuhkaongelma on ratkaistu tekemällä väylästä useampikaistainen.

Säästöä palikankierrätyksellä



PC:n tekstitilan toimintaperiaate.



Sharp MZ 721:ssä ei voi määritellä uusia merkkejä ohjelmallisesti lainkaan, mutta sen merkistössä on valmiita paloja pelikäyttöä varten.

Jokaisen pikselin erillistallennus on nykyisin itsestäänselvyys, mutta näin ei ole suinkaan aina ollut. Varhaisimmissa mikrotietokoneissa suorasta pikseligrafiikasta tulee pakostakin varsin palikkaista muistin vähyyden vuoksi. Esimerkiksi Telmac 1800:n tavallisimmassa grafiikkatilassa on 64×64 mustavalkopikseliä, jotka syövät kahden kilotavun RAM-muistista neljänneksen.

Kun näytölle halutaan lähinnä tekstiä, on mielekästä käyttää näyttömuistin perusyksikkönä pikselin sijaan merkkiä. Tv-tyyppiseen kuvaan tekstiä tuottavat merkkigeneraattorit tulivat 1960-luvulla tv-studioihin ja tietoko-

neiden näyttöpäätteisiin.

Merkkigeneraattori käyttää kullekin merkillä esimerkiksi tavun verran näyttömuistia. Näyttömuistin tukena on merkkimuisti, joka sisältää merkistön merkkien pikselirakenteet. Merkkimuisti on yksinkertaisimmissa laitteissa kiinteää ROM-muistia, mutta usein sen voi sijoittaa myös RAM-muistiin. Kotimikroissa merkkejä on yleensä 256 ja merkin koko 8×8 pikseliä.

Merkkipohjaisuus paitsi säästää muistia, myös nopeuttaa näytönkäsittelyä. Tästä syystä sitä on käytetty paljon peligrafiikassa. Merkit ovat yleensä pelikohtaisia ja voivat olla myös mo-

nivärisiä – esimerkiksi Commodore 64 ja NES tarjoavat neliväriset merkit. Pelikonsoleissa merkkigrafiikkaa kutsutaankin paremmin pelikäyttöä kuvaavalla nimellä ”tile graphics”.

Tekstitila on tarjolla useimmissa suosituissa 8-bittisissä kotimikroissa – suurimpana poikkeuksena Spectrum, jossa on pelkkä grafiikkatila. Isommissa koneissa rautapohjaista tekstiä ei taasen ole yleensä lainkaan lukuunottamatta PC:tä, jossa se on vanhojen MDA- ja CGA-näytönohjainten perua.

Värejä merkeille

Näytölle asetettujen merkkien ulkoasuun voi yleensä vaikuttaa erilaisilla ulkoasuparametreilla eli attribuuteilla. Attribuutit ovat tyypillisesti värejä tai alleviivauksen ja vilkutuksen kaltaisia korostuksia.

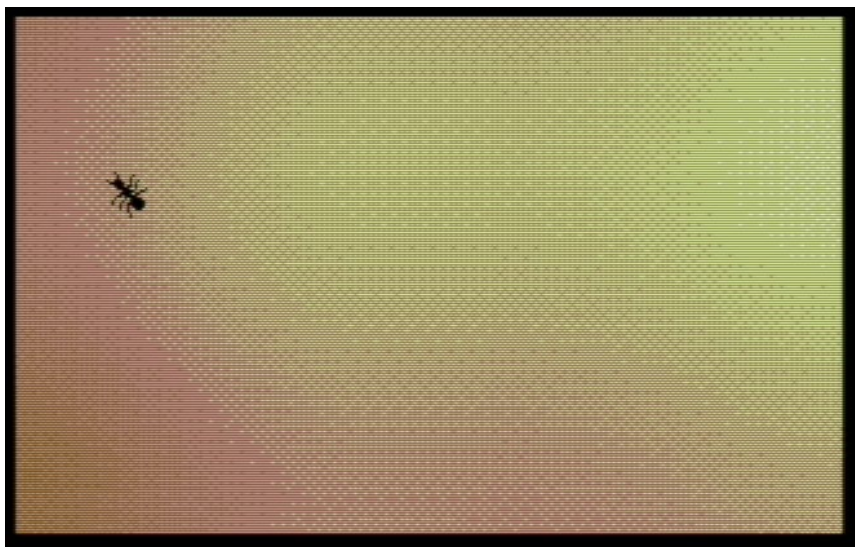
Attribuutit voi tallentaa joko merkkien sekaan tai omiin muistipaikoihinsa. Teksti-TV ja Oric käyttävät ensimmäistä tapaa: niissä kahden erivärisen merkin väliin on jätettävä tyhjä kolo näkymättömälle värinvaihtomerkillä. Erillisillä muistipaikoilla sen sijaan tällaista rajoitetta ei ole.

PC käyttää jokaisen merkin värityksen tavun, joka määrää sen pää- ja taustaväriin. NES on sen sijaan säästeliäs attribuuttimuistissaan – koko ruudun attribuuteille on varattu vain 64 tavua. Kullekin 2×2 merkin alueelle voi siinä valita yhden neljästä erilaisesta neliväripaletista, joiden koostumukset puolestaan määritellään erillisessä palettimuistissa. Tämä 16×16 pikselin palikoita suosiva piirre on vaikuttanut oleellisesti Nintendon peligrafiikoihin ja kenttäsuunnitteluun.

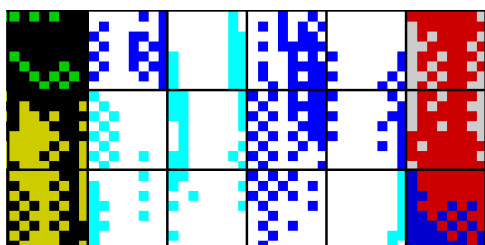
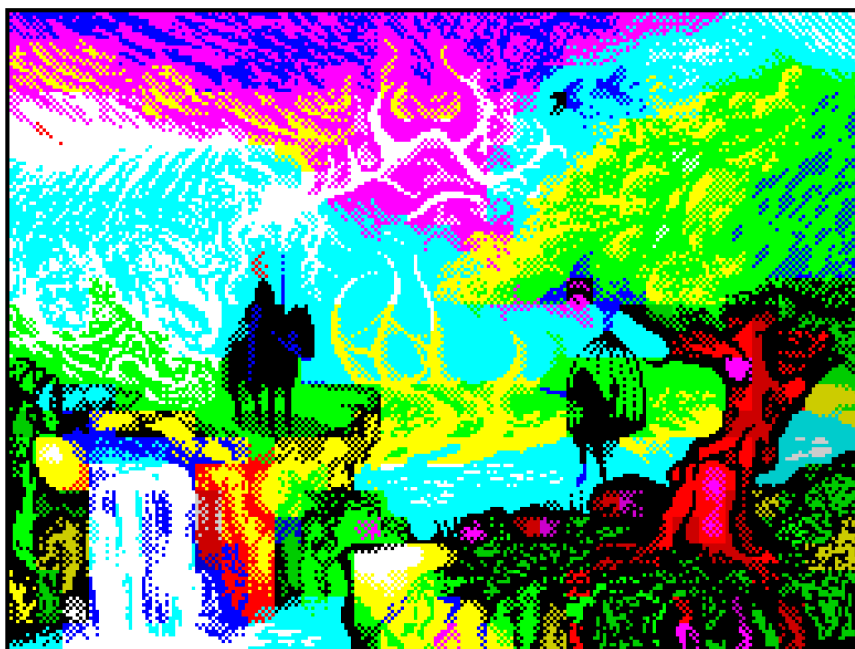
Monissa koneissa attribuuteilla väritetään myös pikseligrafiikkaa. Esimerkiksi Spectrumin ja Commodore 64:n tarkkuusgrafiikassa varsinainen kuva on kaksivärinen, mutta nämä kaksi väriä valitaan erikseen kullekin 8×8 pikselin solulle. Tämä solujakaisu näkyy etenkin Spectrum-peleissä, joissa liikkuvan pelihahmon väri usein tarttuu alle jäävään taustagrafiikkaan. Ilmiötä kutsutaan nimellä color clash tai attribute clash.

Mistä värit tulevat?

Monissa koneissa on enemmän erilaisia värejä, kuin mihin näyttömuisti taipuu. Esimerkiksi PC:n VGA-ohjain pystyy 262 144 erilaiseen väriin, mut-



Merkkien käyttäminen pikseleinä on houkutteleva vaihtoehto, kun värit ja nopeus ovat tärkeämpiä kuin resoluutio. Plasma Byterapersin C64-demosta Unsigned.

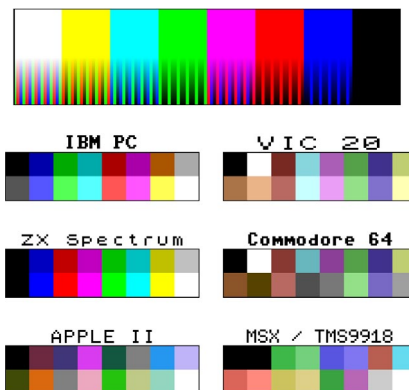


Värisolurajojen häivytyksen Spectrum-graafikon tärkeimpiä taitoja. (Piesiu: Frightened Birds)

ta sen näyttötilat ovat enimmillään 256-värisiä. Siksi ohjaimessa on erillinen 256-paikkainen palettimuisti, joka antaa kullekin värille 18-bittisen RGB-arvon. Paletillista kuvantallennusta kutsutaan pseudocoloriksi vastakohdista truecolorille, jossa näyttömuisti sisältää RGB-arvot suoraan. Silicon Graphicsin työasemissa on ollut jopa

4096-värisiä pseudocolor-tiloja.

Mikäli kokonaisväriavaruus on laaja, siihen viitataan yleensä RGB-arvoilla eli punaisen, vihreän ja sinisen voimakkuuksilla. Toinen vaihtoehto on käyttää tv-tekniikan värijärjestelmiä, joissa väri jakautuu sävyyn ja kirkkauteen. Esimerkiksi Atarin GTIA-piiri tuntee 16 erilaista värisävyä, joista



16-värisiä rautapaletteja. Monet perustuvat yllä näkyviin kahdeksaan binäärisen RGB-väriin.

kullakin on 16 kirkkausastetta. Lisäksi on käytössä täysin räätälöityjä värivalikoimia, kuten Commodoren ja MSX:n 16-väriset rautapaletit.

Joissakin koneissa paletteja ei voi muodostaa vapaasti. Surullisenkuuluisa esimerkki on PC:n CGA-näytönohjain, jonka nelivärigrafiikkatilassa vain yhden värin voi valita vapaasti 16 vaihtoehdosta, ja lopuissa on tyytyminen muutamaa keuhkoon yhdistelmään. Mustan, turkoosin, purppuran ja valkoisen muodostama värimaailma lielee tuttu kaikille 80-luvun PC-pelejä pelanneille.

Koko kuvaan vaikuttavien globaalien palettien lisäksi on paikallispaletteja, kuten Spectrumin kaksiväriset värisolut. C64:n monivärigrafiikassa solukohtaiset attributit määrittävät kolme väriä, ja neljäs väri jää globaaliksi. Toinen tapa luoda paikallispaletteja on muuttaa globaaleja värejä ruudunpiirron eri kohdissa. Vanhojen demojen värikkäät rasteripalkit perustuvat värin vaihtamiseen kuvan jokaisella vaakajuovalla.

Amigan Hold and Modify -tilat ovat erikoinen välimuoto paletillisesta ja truecolor-lähestymistavasta. Esimerkiksi HAM6:ssa on pikselille 64 mahdollista väriarvoa. Arvot 0–15 otetaan suoraan palettimuistista, mutta loput käyttävät värin pohjana edellistä pikseliä. Väriarvo 16–31 vaihtaa punaisen komponentin, 32–47 vihreän ja 48–63 sinisen. HAM-tilojen tarkoituksena on ollut parantaa valokuvatyypin grafiikan esityslaadua. Samantapaisia ratkaisuja on ollut käytössä vielä 90-luvulla esimerkiksi HP:n halvemmissä Unix-työasemissa.



Uridium-räiskintäpelii on pidetty C64-skrollauksen mestarinäytteenä, joten sen Spectrum-sovitus lähenee jo mustaa magiaa.



Frogger-peli on suunniteltu rasterit mielessä: pelikenttä jakautuu vaakasuoriin raitoihin, jotka skrollaavat eri nopeuksilla eri suuntiin. Atari 2600 -versiossa rasteritekniikoilla myös tehdään yksivärisistä spriteistä monivärisiä.



Päällekkäisiä skrollitasoja Amigan Shadow of the Beast -pelissä.

Reunukset ja resoluutiot

Etenkin televisioon kytkettäviä tietokoneita suunniteltaessa on jouduttu huomioimaan se, että kuvaputken reunat ovat pyöreät ja näkyvä alue vaihtelee putkesta toiseen. Näiden koneiden kuva ei yleensä vie koko näyttöä, vaan sen ympärillä on yksiväriset reunukset.

Jotkin koneet antavat vapaasti määritellä kuva-alueen koon ja sijainnin, kun taas joissakin ei reunuksista pääse eroon mitenkään. Commodore 64:n reunukset on periaatteessa lukkoon lyöty, mutta videopiiriä huijaamalla ne on mahdollista poistaa. Reunukset ylittävien efektien toteuttaminen on kuulunut C64-demoskeneen alusta asti.

Vanhat kotimikrot ja konsolit tuottavat yleensä muutenkin vain yhdenlaista kuvasignaalia. Eurooppaan myydyt C64:t tuottavat aina tasan 312 rasterijuovaa 50 kertaa sekunnissa, kun taas amerikkalaisversiot tuottavat 262 juovaa 60 kertaa sekunnissa. Reunusten väliin jää molemmissa versioissa 200 juovaa, ja kullekin juovalle mahtuu 40 merkin leveydeltä grafiikkaa. Standardien välillä ei pysty vaihtamaan – videopiiristä on yksinkertaisesti eri versiot eri mantereille.

Eräissä koneissa signaalin rakennetta voi sen sijaan muokata. PC-näytönohjaimissa kaikki on rekistereissä: kuinka monta juovaa tuotetaan, kuinka monta pikseliä juovalle ja kuinka paljon aikaa varataan vaak- ja pystysuuntaisille suihkunpalautuksille. VGA:sta alkaen tarjolla on myös useita eri pikselitaajuuksia eli dotclockeja, jotka määräävät näyttömuistin lukunopeuden. Näin yhdestä ja samasta videopiiristä saa ulos valtavan määrän erilaisille näytöille sopivia resoluutioita ja virkistystaajuuksia.

Helpotusta hahmojen liikkeisiin

Näyttömuistiin tallennetun kuvan päällä halutaan usein liikuttaa irtokuvioita kuten pelihahmoja tai kohdistimia. Tämän helpottamiseksi on keksitty spritet, jotka ovat muusta kuvasta riippumattomia, osittain läpinäkyviä bittikarttakuvia. Tekniikka on peräisin 70-luvun kolikkopeleistä.

Yhdellä vaakajuovalla näkyvien spritejen määrä on yleensä rajoitettu – esimerkiksi C64:n videopiirissä on logiikka-alueet kahdeksaa spritea

varten. Mikäli hahmoja sattuu liian monta rinnakkain, joudutaan niiden välillä vuorottelemaan, mikä aiheuttaa välkymistä.

Kunkin spriten sijaintia, ulkomuotoa ja väritystä voi yleensä vaihtaa erikseen. Atari Lynxissä spriten kokoa voi säätää liki portaattomasti, ja monissa piireissä spriten voi myös peilata vaak- tai pystysuunnassa, mikä säästää muistia.

Spriterauta on hyvin yleinen 80-luvun pelikonsoleissa ja hieman kyvykkäämmissä kotimikroissa. PC:ssä, Appleissa, Atari ST:ssä ja brittikoneissa ei spritejä kuitenkaan ole lukuun ottamatta 2D-kiihdytettyjen näytönohjainten hiirikohdistimia.

Vierivät maisemat

Monissa peleissä on laajoja pelialueita, jotka vierivät eli skrollaavat näytöllä. Vanhojen koneiden skrollausominaisuudet eroavat huomattavasti toisistaan.

Spectrum on hyvin skrollivihamie-linen kone. Siinä on pelkkä grafiikkatila, jonka näyttömuistin sijaintia ei voi rukata mitenkään. Koko ruudun skrollauksen toteuttaminen vaatii siis pahimmillaan koko kuuden kilotavun muistialueen uudelleenkirjoittamisen joka askeleella. Tämä on pääsyy sille, miksi alun perin Spectrumille tehdyt pelit eivät skrollaa juuri koskaan.

NES edustaa toista ääripäätä: laitteessa on erityiset skrollirekisterit, jolla näytön kuvaa voi siirtää pikselin tarkkuudella pysty- ja vaakasuunnassa vaikka kokonaan ulos näytöltä. Sivuuun väistyvä kuva antaa tilaa toiselle kuvalle, joka luetaan joko samalta tai eri muistialueelta. Vieritys on hyvin kevyt ja yksinkertainen toteuttaa.

Myös C64:ssä on skrollirekisterit, mutta ne on tarkoitettu pelkkään pikselintarkkaan hienosäätöön. Koska näyttömuistin alkukohdan voi määrätä vain kilotavun tarkkuudella, joutuu suoritin siirtämään näyttömuistin sisältöä vierityksen aikana. Skrollausta joutuu siis miettimään hieman tarkemmin kuin NESissä, ja bittikartta-kuvan nopea vieritys vaatii esoteerisia temppuja.

Kerroksittainen näyttömuisti antaa hyvät edellytykset niin sanotulle paralaksiskrollaukselle, jossa on useita eri tahtiin päällekkäin vieriviä kerroksia. Super Nintendon videorauta tarjoaa



Rauta-avusteisen perspektiivin hyödyntämistä Super Nintendon Battle Cars -pelissä.

parhaimmillaan neljä itsenäisesti vierittävää kerrosta.

Suihkun seuraaminen

Osa ruudusta saatetaan haluta jättää skrollaamatta, jos vaikkapa yläreunaan halutaan tilaa pistelaskurille. Näytön eri osiin voidaan haluta eri värimaailmoja, tai isoa grafiikkakuvaa saatetaan haluta zoomailla ja venyttää eri tavoin. Nämä mahdollistuvat rasteritekniikoilla, eli rukkaamalla videopiirin parametreja kesken näytönpöiirron.

Useimmissa 80-luvun koneissa on keino tahdistaa suorittimen toiminta näytönvirkistykseen. Spectrum tarjoaa vain signaalin siitä, milloin elektronisuihku on palannut näytön ylänurkkaan. Commodore 64:n videopiirin saa tuottamaan keskeytyksen, kun suihku saavuttaa tietyn juovan. Atarin Antic ja Amigan Copper ovat apupiirejä, jotka tekevät rasterirukkauksia pääsuorittimen puolesta.

Nykylaitteissa tarkalle rasterikikkailulle ei juuri ole käyttöä, mutta esimerkiksi grafiikkasivun vaihto kannattaa edelleen tehdä näytönvirkistysten välissä. Mikäli sivua vaihdetaan kesken kaiken, aiheutuu kuvaan ilkeän näköistä repeämistä eli tearingiä.

Rautainen perspektiivi

Vaikkei laitteessa olisikaan varsinaista 3D-kiihdytystä, se voi tarjota välineitä syvyysperspektiivin toteuttamiseen.

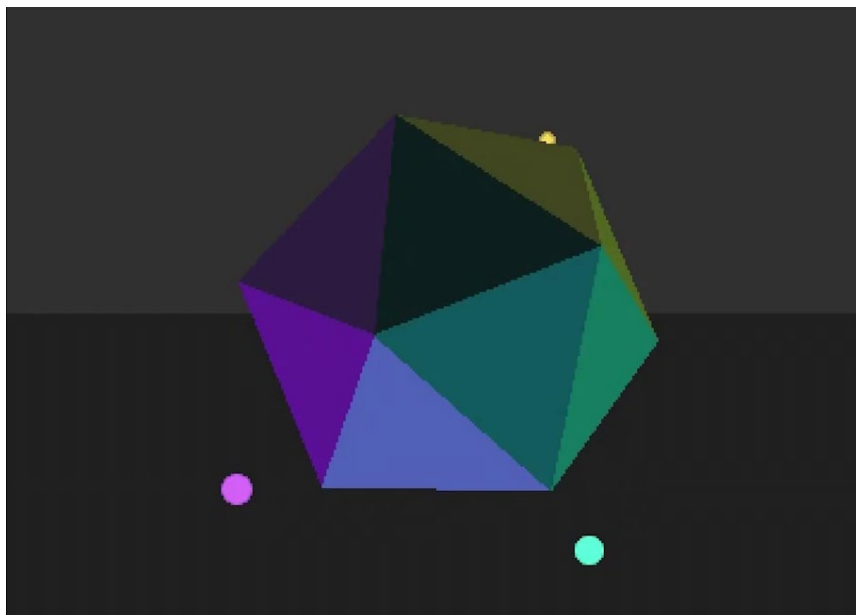
Aiemmin mainitut päällekkäiset skrollaustasot kuuluvat näihin, samoin kuin mahdollisuus määrittää, millä tasolla kukin sprite näkyy taustagrafiikan suhteen. Muutamat koneet menevät kuitenkin vielä pitemmälle.

Atari Lynx pystyy venyttämään sekä spritejä että näyttömuistin kuvaa liki portaattomasti. Tämä on erityisen hyödyllinen ominaisuus syvyyssvaikutelman luonnissa. Näytön eri alueita voi myös skaalata eri tavoin, joten esimerkiksi taustan saa näkymään sitä pienempänä, mitä lähempänä horisonttia se on. Vastaava ominaisuus on myös Super Nintendossa, joskaan sen laitteisto ei tue spritejen skaalausta.

Rajalta toiselle

Joskus laitteen grafiikkaominaisuuksia halutaan venyttää. Lähes kaikilla koneilla käyttökelpoinen tekniikka on lomitus eli interlace, jossa vaihdetaan kahden hieman erilaisen kuvan välillä joka näytönvirkistyksellä. Kuvien värit sekoittuvat, jolloin esimerkiksi 16-värinen laitteisto näyttääkin kykenevän 128 väriin. Haittapuolena on kuvan välkkyminen.

Jos lomitetut kuvat ovat puolen pikselin verran erillään toisistaan, mahdollistuu myös resoluution tuplaaminen. Esimerkiksi tavallinen analogitelevisio käyttää tätä tekniikkaa – juovat ovat joka toisella näytönvirkistyksellä puoli juovanväliä alempana. Juovalomitusta



Tyypillistä blitterillä kiihdytettyä 3D-grafiikkaa Lemon-ryhmän Amiga-demossa Rink a Dink Redux.



Tasohyppelypeli VVVVVV on nykyaikaisille koneille, mutta sen estetiikka on vanhemmaa perua.

ei kuitenkaan voi käyttää useimmilla kasibittisillä koneilla, sillä ne tuottavat vain progressiivista kuvaa. Sen sijaan vaakaskrollirekisteri voi mahdollistaa tarvittavan puolen pikselin siirtymän.

Väirajoitteita voi kiertää myös rasterikikkailulla. Commodore 64:n FLI-grafiikassa (Flexible Line Interpretation) värisolut pakotetaan yhden pikselin korkuisiksi, jolloin värejä voi käyttää huomattavasti vapaammin. IFli-kuviissa (Interlaced FLI) on kaksi lomitettua FLI-multicolor kuvaa, jotka ovat puoli pikseliä erillään – teoriassa siis 128-värinen kuva 320×200 pikse-

lin resoluutiolla ja lähes vapaalla värin asettelulla.

Kiihdytyksen aika

Yksinkertaisissa laitteistoissa vain pääsuoritin kirjoittaa näyttömuistiin. Näyttömuistin käsittelyyn erikoistunut grafiikkaprosessori (GPU) on kuitenkin nykyään lähes kaikissa laitteissa, joissa on jonkinlainen näyttö. Se ilmestyi kotitietotekniikkaan ensimmäisenä Amigassa ja sen jälkeen esimerkiksi MSX-2:ssa ja Atari STE:ssä.

Amigan ja STE:n blitter-piirit soveltuvat muistialueiden täyttämiseen

ja kopiointiin, useammalta alueelta luetun datan yhdistämiseen loogisilla operaatioilla sekä viivojen ja täytettyjen pintojen piirtämiseen. Näille toiminnoille on ollut käyttöä sekä peligrafiikassa että graafisissa käyttöliittymissä.

Kun näyttömuistiin pystyttiin piirtämään aina vain nopeammin, alkoivat 80-lukulaiset näytönohjausjipot menettää merkitystään. Amigaan ei kaivattu tekstiilaa blitterin ja muistimäärän vuoksi. Kun 486-luokan PC-suorittimet pystyivät piirtämään koko pelinäköymän uusiksi jokaisella näytönvirkistyksellä, ei enää haaveiltu rautaavusteisesta monikerrosskrollauksesta, spriteistä eikä rasterikeskeytyksistä.

Teksturoitua 3D-grafiikkaa kiihdyttävät laitteet tulivat kulutuselektronikkaan 1990-luvun alkupuolella Sony Playstationin myötä, ja nykyään 3D-kiihdytystä on jopa halvimmissa kännyköissä. Nämä laitteistot tarjonneen ansaitsisivat kuitenkin aivan oman artikkelinsa.

Perinteet jatkuvat

Grafiikkaominaisuudet ovat muovanneet hyvin voimakkaasti eri koneiden ohjelmistokulttuureja. Esimerkiksi Commodore 64:llä ja Amigalla on perinteisesti keskitytty videopiirin erityispiirteiden hyödyntämiseen: rasterikikkailuun, parallaksivierityksiin ja pehmeisiin siirtymiin. Spectrumilla ja 90-luvun PC:llä on puolestaan ollut houkuttelevampaa panostaa suoritin-keskeiseen grafiikkaan, kuten uudellaisiin 3D-konsepteihin.

Demoskene, pikseligraafikot ja homebrew-kehittäjät hyödyntävät eri-ikäisten ja -kokoisten laitteiden graafista ilmaisuvoimaa. Vanhan ja selvärajaisen tekniikan käyttö on näissä piireissä tietoinen valinta, joka tuo teknisiä haasteita ja herättää luovuutta.

Vanhan näytönohjaustekniikan luomia esteettisiä perinteitä jatketaan myös nykylaitteilla. Esimerkiksi monet indie-pelinkkehittäjät suosivat ”kasibittistä” tyyliä, jonka he kokevat helpommaksi ja oikeamielisemmäksi kuin valtavirtapeliin realismi-ihanteen. Minimalistisempi estetiikka toimii siis ilman alkuperäisiä puitteitaankin – vaikka puitteet toki kannattaa ymmärtää, jotta estetiikasta saa enemmän irti. 🎮



Kumppania etsimässä

Rakas Skrolli,
Mistä ujo tietokonenörtti löytäisi tyttöystävän?
- Ihmettelijä (PS. Kaverille kysyn.)

Skrollin toimitus suosittelee lämpimästi Internet-tietoverkkoa. Koke-
mustemme mukaan ihmissuhteita
muodostuu erityisen hyvin kahdessa
palvelussa: IRC ja OkCupid.

IRC on tuottanut todistetusti hyviä
tuloksia jo vuosikymmenten ajan – ai-
nakin niille, jotka jaksavat ummehtu-
neiden erikoiskanavien lisäksi myös
yleisempiä ja nuorekkaampia areenoita.
IRC:tä ei välttämättä kannata ruve-
ta käyttämään pelkässä kumppanin-
hakumielessä, vaan yhteydet hyviin
kumppaniehdokkaisiin syntyvät luon-
tevimmin muun verkostoitumisen
ohessa. Kanavien käyttäjätapaamisiin
kannattaa osallistua etenkin, jos niitä
järjestetään laajempien hörhötapah-
mien yhteydessä.

Deittipalvelut mahdollistavat suora-
viivaisemman etsinnän. Etenkin OkCu-
pidista on älykoiden ja friikkien helppo
löytää kaltaisiaan, eikä tietokoneharras-
tustakaan katsota lainkaan pahalla. Si-
inä missä muut palvelut tuntuvat usein
simuloivan kulmaräkelöitä pinnallisine
hakukriteereineen, OkCupid pisteyttää
hakutulokset hyvinkin syvälle menevien
kysymysten perusteella.

Guaget vittuun!

Skrollin viime numeron pääkir-
joituksessa puhuttiin demo-
scenestä, peleistä ja vihasta.
Voimakas sana muuten tuo viimeinen.
Mietin sen jälkeen vielä tunteja tulisi-
ko minun ottaa lehden pääkir-
joitukseen, koska silloinhan edustai-
sin pääkirjoituksessa sanottua nuorta
vihaista valkoista poikamiestä. Päätin
kuitenkin ottaa riskin.

Pääkirjoituksessa moititaan pelaa-
jia, gamereita, stereotyyppisiksi val-
koisiksi heteromiehiksi, jotka elävät
misogyniasta ja ihannoivat väkivaltaa.
Samaan aikaan voitotellaan, että myös

Assemblyt uhrattiin samalle alttaril-
le, kun YLE lähetti sieltä pelaamiseen
keskittävää ohjelmaa, eikä yhtään kes-
kittynyt demoihin. Syyksi lehdessä löy-
dettiin tietenkin sisäänpäin lämpiävä
gamereiden kuppikunta, eikä mitään
vikaa löydetty omasta demoskenestä.

Suurimmalla osalla pelaajia ensim-
mäiset kosketukset demoihin syntyy
Assemblyllä. Pidän tätä asiaa hyvä-
nä. Ensimmäinen kosketus demoske-
neen tapahtuu kuitenkin huudoilla:
"QUAKET VITTUUN! EIKÖ SAATA-
NA NE MONITORIT SAMMU! TUL-
LAANKO VITTU HELVETTI SAM-
MUTTAMAAN!?!?!?!?!?" Kyllä, pelaajat
ovat niitä väkivaltaisia kuppikuntalaisia.

Kuppikunnatkin ovat jännä asia. Pe-
laajat ovat itse rakentaneet vuosien mit-
taan erilaisia sosiaalisia verkostoja. Meil-
lä on liitto, kaksi uutissivustoa, tubettajia,
streamaajia, säännöllisiä tapaamisia eri-
puolilla Suomea. Kaikkiin näihin voivat
ihmiset tulla vapaasti. Tästä syystä pelis-
kene kasvaa kokoajan.

Yritin googlettaa demoskene helsin-
ki jne. googlesta ja löysinkin awards-
sivuston, missä on viimeksi jaettu
palkintoja vuonna 2012. Sivuston uu-
simmat uutiset ovat vuodelta 2013. Ei
jätä hirveän positiivista vaikutelmaa.

En ole itse ainakaan törmännyt
misogyniaa, väkivallan ihailua tai kup-
pikuntalaisuutta ns. propelaajien kes-
kuudessa. He osaavat edustaa lajiaan.
He käyttäytyvät kohteliaasti, antavat
hyviä lausuntoja lehdistölle ja luovat
omia verkostojaan. He tekevät hyvää
työtä pelaamisen eteen.

Olen kuitenkin itse äärimmäisen kii-
tollinen demoskenelle, koska ilman sitä
ei minulla olisi pelejä mitä pelata. Tästä
syystä yhdynkin pääkirjoituksen loppu-
kaneettiin, jossa toivotaan yhteistä elä-
mää. Sitä itsekin toivon. Haukkumalla
pelaajat valkoisiksi misogynistisiksi ga-
mereiksi ei tätä yhteiseloja kuitenkaan
edistetä - päinvastoin.

Teemu "Wabbit" Hiilinen

Olen pahoillani, jos kirjoitukseni ko-
ettiin kaikkiin pelaajiin kohdistuvaksi
yleistäväksi hyökkäykseksi. Kyse oli
kuitenkin nimenomaan stereotyyppistä,
joka elää ihmisten mielikuvissa, vaikka
aika on ajanut sen ohi. Kun pieni, "true
gamereiksi" itseään kehuva vihaa-
jaklikki aikaansaa tappouhkauksillaan
laajan kohun, niin olkiukko vahvistuu.
Samoin se vahvistuu, kun Yleisradio

tuo keskelle mummokanaviensa ohjel-
maa pelejä, jotka on suunnattu stereo-
typian mukaiselle kohderyhmälle.

En maininnut demoskeneä pääkir-
joituksessa, mutta on totta, että myös
sen piirissä on omat ongelmansa – äi-
jäntyvälle underground-alakulttuu-
rille ominainen sisäänpäinlämpiävyys
ehkä keskeisimpänä. Tähän on kuiten-
kin yritetty puuttua esimerkiksi järjes-
tämällä kerhoja ja tilaisuuksia, joissa
aloittelijoita opastetaan mukaan kädes-
tä pitäen. Samaisessa lehden numeros-
sa käsitelty Graffathon-tapahtuma on
esimerkki tästä. Varmasti huonot ste-
reotypiat tekevät hallaa myös demos-
kenen uusiutumiselle, mutta vähäi-
nen medianäkyvyys lienee kuitenkin
keskeisempi ongelma: minkäänlainen
"demogate" ei edes pystyisi nousemaan
päävälehtien otsikoihin.

Skrollin agendaan kuuluu tieto-
konekulttuurin eri muotojen esittely
laajemmalle yleisölle. Mielestämme
tietokonekulttuuri voisi hyvin näyt-
tättyä kaikenlaisten ihmisten juttuna
eikä pelkästään tietyn stereotyyppisen
miesryhmän touhuiluna. Yleisradion
ennakkoluuloja pönkittävä tapa rapor-
toida viime Assemblyistä toimi suo-
raan agendaamme vastaan – mikä oli
vähän ikävää, koska Yle tavoittaa koko
kansan ja me vain parituhatta lukijaa.
Jos Yle kuitenkin haluaa ottaa virheen-
sä takaisin vaikkapa ensi vuoden ta-
pahtuman puitteissa, niin Skrolli tulee
mielellään yhteistyökumppaniksi.

Ville-Matias Heikkilä

Lisää julisteita Skrolliin! Haluai-
sin vielä jotain sopivaa paska-
huussin seinälle. Tehkää vaikka
infografiikka Windowsin eri versioista.
Aivan loistava tämä viimeisin (2014.2).
Päättyi kunniapaikalle työhuoneen sei-
nälle. Kyllä tuli nostalgiset fiilikset...

Raimo

Kirjoita Skrollipostiin!

Vaikka nettiviestintä onkin kätevää,
uskomme hitaammallekin keskustelul-
le olevan tilaa myös tietokonekulttuu-
rissa.

Voit kirjoittaa meille sähköpos-
titse toimitus@skrolli.fi (otsikkoon
"Skrolliposti") tai perinteisellä postilla
Skrolli-lehden posti

PL 882

33101 Tampere



CoolerMasterSuomi

The Ultimate Typing Experience



NOVATOUCH TKL

HYBRID CAPACITIVE SWITCHES KEYBOARD

NovaTouch TKL luotiin tekemään vaikutus! Tämän se toteuttaa käyttäen Hybrid Capacitive -kytkimiä, jotka ovat äärimmäisen hiljaisia, mutta soveltuvat myös raskaan luokan pelaamiseen tai kirjoittamiseen.

189 €



Jimm's Balor

Jimm's Gamer -sarjan koneet ovat olleet jo pitkään pelaajien suosiossa ja suunnannäyttäjinä. Gamer -koneet ovat niittäneet mainetta arvosteluissa ja ovat moninkertaisia **lehdistön -testivoittajia!**

Ominaisuuudet:

Intel i5-4690K 3.5GHz, -prosessori | Intel Z97-piirisarjan emolevy |
NVIDIA GTX 970 2Gt -näytönohjain | 240Gt, 2.5", Sata III -SSD asema |
8Gt (2x4Gt) DDR3 1600MHz -keskusmuisti | Cooler Master Silencio 550 **Jimm's Edition** -kotelo | Windows 7 Home Premium 64-bit (Suomi)

1339 €



Asiakaspalvelu
+358 29 70 70700
asiakaspalvelu@jimms.fi

Jimm'siltä löytyy nettitalauksiin
aina toimituskuluton vaihtoehto!

Noutopalvelumyymälä **Turku**

Lukkosepänkatu 7
20320 Turku

Hinnat €/kpl, sis. alv24%

