

**KYRÄILEVIEN
OHJELMOIJIEN
KERHO**

The Revenge

Lasse Öörni

NaNoWriMo 2010.

Lue vain omalla vastuulla. Kaikki yhtymäkohdat todellisuuteen ovat omassa mielikuvituksessasi. Älä yritä toistaa tässä kirjassa esitettyjä menetelmiä minkään todellisen 3D-enginen tai muunkaan ohjelmistoprojektin suunnittelussa tai toteutuksessa, siitä ei voi seurata mitään hyvää!

To those who fought and bled.

Kiitokset "ohjelmoijan vaikerruksen" kirjoittamiseen osallistuneille. Turvallisuussyistä nimiä ei luetella.

ENSIMMÄINEN KIRJA – MAA

RIN - Mielen ja ruumiin voima

Päivä 0.

Tästä se lähtee.

Tällä kertaa kaikki tullaan tekemään oikein. Minulta, kyräilevältä ja kyyniseltä ohjelmoijalta, tarkemmalta toimenkuvaltani Graphics Subsystem Technical Lead, tämä on aika voimakas lausunto. Eikö minun jos kenen tulisi epäillä moista väitettä, sillä onhan second system effect tunnettu ja voimakas antipattern? Mutta silti, kun mietin asiaa, en voi tulla mihinkään muuhun johtopäätökseen. Merkit ovat selvät.

Tällä kertaa kaikki tullaan tekemään oikein.

Hieman taustatietoa: me olemme Vision3D-middlewaren luoja. Vision3D on sovellusagnostinen middleware-ohjelmisto virtuaalitodellisuuksien ja 3D-internetin luomiseen.

3D-internetin?

Mikä se muka on?

Mitä paskaa! Mitä absoluuttista paskaa!

Mutta tämä määritelmä on siis jotain, jonka markkinoitiosastomme on kehittänyt. Minä itse kyllä tiedän hyvin, mitä Vision3D on ja mitä se ei ole. Sanotaanko nyt suoraan, että se muistuttaa riittävässä määrin pelimoottoria, jotta voin nukkua yöni rauhassa, ilman että pelkään kontribuivani liiaksi johonkin turhanpäiväiseen hörhöilyyn. Pelimoottorin määritelmä on nyt tässä ohjelma, joka riittä-

vän monta kertaa sekunnissa simuloi 3D-maailman tilaa eteenpäin yhden askeleen, ja sitten piirtää sen ruudulle käyttäen grafiikkakiihdytystä, paitsi jos ollaan dedikoitu serveri-tilassa, jolloin viimeksimainittu sivuutetaan. (Jos asiakkaan koneessa on Intel GMA-sarjan näytönohjain, on kiihdytys lähinnä kaunisteleava ilmaisu, mutta nyt olen ek-symässä sivuraiteille.)

Mutta takaisin nykyisyyteen. Vision3D:n viimeinen julkinen versio on tällä hetkellä 1.4, ja vaikka se periaatteessa täyttääkin tehtävänsä (joskin alkeellisesti), on siitä sisältä käsin muotoutunut lonkeroinen kammotus, jonka rinnalla Cthulhukin kalpenee. Juuri kukaan ei uskalla koskea siihen paitsi pienimpien kirurgisten hackien muodossa.

Mutta nyt muutos on tulossa.

Me olemme saaneet vihreää valoa siirtyä kehityksessä seuraavaan vaiheeseen. Vision3D 2.0 tulee hylkäämään kaiken, mikä 1.4-versiossa on epätoimivaa. Tälle muutosvaiheelle on varattu 3 + 3 kuukautta: kolmen kuukauden päästä, eli helmikuun alussa meidän tulee pystyä tekemään omia projektejamme 2.0:n esiversiolla (kaikkien 1.4:n ominaisuuksien ei tarvitse vielä toimia; osa niistä on luultavasti niin turhia, ettei niitä edes kannata toteuttaa uuteen versioon) ja suurelle yleisölle se on tarkoitus julkaista ennen kesää. Normaalisti rauhallinen Pääohjelmoijamme on antanut seuraavan sotaisan mandaatin, jota on ilo totella: "Repikää se paska vaikka tuhannen palasiksi, jos tarvii."

Mutta tämä ei tule olemaan sota. Tämä tulee olemaan nautintoa. Erityisesti tulen nauttimaan siitä, kun vanha renderöintikoodi, josta voidaan kääntää sekä Direct3D9-että OpenGL-versiot teloitetaan, haudataan, kaivetaan ylös haudasta ja tapetaan vielä muutaman kerran, ennen kuin poltetaan ja tuhkan annetaan levitä tuuleen ja painua alas Norjan vuonoihin, samalla kun kuunnellaan Immortalia.

Minulla on toki epäily, että OpenGL-renderöintiä tullaan vaatimaan edelleen, mutta ennakkotietojen mukaan

minä en ole siitä vastuussa. Vision3D:n on tarkoitus siirtyä täysin shader-pohjaiseen renderöintiin, ja sillä on vihdoin lupa käyttää Shader Model 3.0- ja jopa SM4.0 -ominaisuuksia.

Itseasiassa ilkeä suunnitelmani on, että Direct3D SM2.0-tason renderöintiä ei toteuteta ollenkaan, vaan todelta paskoilla näytönohjaimilla tiputaan takaisin OpenGL:n käyttöön, mikä tietysti hidastaa näytön päivitystaajuutta ennestään. Samaan aikaan asiakkaidemme käyttämien 3D-mallien polygonimäärät ovat kuitenkin kasvaneet, eivätkä he luultavasti ala penkomaan hidastumisen tarkempia syitä, vaan ostavat vain kiltisti tehokkaammat näytönohjatimet.

Hetkinen, saatatte kysyä. Onko ammattiyhpeyteni laske-
nut? Sallinko minä sen, että Vision3D toimii vähemmän kuin optimaalisesti, vielä kaikkein tärkeimmässä osa-alueessa, eli renderöinnissä?

Sanotaanko niin, että olen viisastunut.

Osaan valita taisteluni. Uloste on ulostetta eikä muutu kullaksi, vaikka sitä kiillottaisi maailman tappiin asti.

Mutta siitä voitte olla varmoja, että niin totta kuin aurinko nousee idästä ja ninja on aina kovempi kuin piraatti, SM3.0 -polku tulee toimimaan mahdollisimman optimaalisesti. Siinä missä vanha versio piirsi kiltisti (ja äärimmäisen hitaasti) multipassina, nyt dynaamisesti fragmenteista käännettävät übershaderit takaavat sen, että kaikissa järkevissä tilanteissa, kaikilla järkevilla dynaamisten valojen lukumäärillä yksi piirtopassi on kaikki, mitä tarvitaan. Luonnollisesti kaksinkertaisella nopeudella toimiva Z-prepass voi olla tarpeellinen pikselistagen kuormituksen kasvauksessa, mutta sitä ei nyt oikeastaan voi edes laskea passiksi. Ja se pitää profiloida, ennen kuin hätiköidään.

Huomaan, että olen menossa asioiden edelle. Huomenna on versio 2.0:n aloituspalaveri, ja on luultavaa, että todellisuudessa kestää jonkin aikaa, ennen kuin pääsen lähellekään renderöintiä. Ensin on viikatteella niin paljon töitä, että huh huh!

Oikeastaan minun pitäisi mennä nukkumaan, mutta tieto siitä mitä on tulossa aiheuttaa niin kovia tärinöitä, etten usko saavani unta vielä vähään aikaan. Asiaa ei suinkaan helpota, että huomenna vakioruokalassamme on jälkiruokana appelsiiniriisiä. Jos olen laskenut taivaankappaleiden sykleistä oikein, pitäisi huomisen annoksen olla vielä tavallistakin haasteellisempi. Viimekertainen oli laimea ja helpohko, eli toisinsanoen pettymys.

Koska yritys olisi kuitenkin epic fail, laitan suosiolla vielä kotikoneeni monitorin päälle ja kaadan sopivan kokoisen viskiryyppyn. Koneella on VPN-yhteyden kautta otettu täydellinen checkout Vision3D:n koko kehityshistoriasta kaikkine brancheineen.

Viskin poltellessa kurkkuani mukavasti selaan koodihakemistoja. Syvähierarkinen rakenne ja tarpeettoman pitkät luokannimet aiheuttavat spontaania myötähäpeää: täytyy tosin muistaa, että minua itseäni saa syyttää niistä suurimmaksi osaksi.

Olenmaisinta on, että pian tuota sontakasaa ei ole! Tai ainakin se on metamorfosoitunut huomattavasti kauniimpaan muotoon.

No niin, täältäähän löytyy jotain varsin mukavaa. Tiedosto BindFixedFunctionTextureStage.cpp, riviltä 215 eteenpäin:

```
typedef void (*bindFunc) (void*);  
unsigned int textureBindFunctionOffset = 0;  
if (pMaterial -> HasTextureStage(0))  
    textureBindFunctionOffset += 4;  
#ifdef HAS_MULTITEXTURE_SUPPORT  
    if (pMaterial -> HasTextureStage(1))  
        textureBindFunctionOffset += 4;  
    if (pMaterial -> HasTextureStage(2))  
        textureBindFunctionOffset += 4;  
    if (pMaterial -> HasTextureStage(3))  
        textureBindFunctionOffset += 4;  
#endif
```

```
bindFunc func = (bindFunc*) gGlobalRenderState.mBind-  
Functions [textureBindFunctionOffset];  
(*func)(pMaterial);
```

C++ -kielistä koodia, joka iloisesti käyttää funktiopointtereita, ja olettaa tietyn pointterin koon. Tässä on kiistämättä mukava tosiohjelmoinnin maku, ja ellen väärin muista, koodi on omaa käsialaani, mutta versiossa 2.0 tällaiselle pelleilylle ei silti yksinkertaisesti ole sijaa.

Mutta hyi hyi, paras pistää pois tuollaiset, ennen kuin minua alkaa etomaan oikein kunnolla.

Nyt nukkumaan.

Päivä 1.

Marraskuun ensimmäisen päivän aamu on kirpeä; pohjoisesta puhaltava tuuli saa sen tuntumaan vielä huomattavasti koleammalta. Mutta vaikutelma on juuri sopiva: olen kuin soturi, joka kulkee yli ruumiiden täyttämän taistelukentän syksyisen viikatemiehen seurattessa jokaista askeltani.

Ruumiit ovat huonoa koodia, joka pian tullaan sanitoimaan!

Työmatkani suuntautuu aina Sodexo-ruokalan ohi, jossa syön myöhemmin päivällä. Kuten tavallisesti, kuljen pyörällä, mutta olen havainnut tahdissani tiettyä veltostumista. Jokaiseen liikuntasuoritteeseen tulisi mieluiten suhtautua kuten taisteluun. Asiaan onkin tullava korjaus.

Henkinen tarkastuslista: 1. Uuden maailmanjärjestyksen aloittaminen 2. Appelsiiniiriisi.

Tänään mikään ei voi mennä pieleen.

Kymmenhenkinen koodaustiimimme, miinus pääohjelmoijamme Valtteri, on kokoontunut neuvotteluhuoneeseen. Tekniset ja vähemmän tekniset artistimme loistavat pois-
saolollaan, mutta heitä ei juuri nyt tulla tarvitsemaankaan. Tässä valetaan uuden loistavan arkkitehtuurin preussilais-
ta kivijalkaa, ja on aivan sama, minkälaista graafipohjaista
materiaalieditorisysteemiä he tällä kertaa toivovat. Tosin

tällä kerralla on se ero, että jos aikaa jää, systeemi saate-
taan jopa toteuttaa, kun vanhaa fixed function -toiminnal-
lisuutta ei ole enää jarruna, ja jos renderöintiarkkitehtuu-
rimme joustavuus ja loistavuus toteutuu kuvittelemassani
laajuudessa.

Ajatus on tietyllä tavalla pelottava, sillä jos materiaali-
editori toimii sillä periaatteella, että artisti voi yhdistää
periaatteessa rajattoman määrän tekstuureja, parametreja
ja niiden välisiä matemaattisia operaatioita, voi lopputu-
loksena myös olla rajattoman mittainen shader-ohjelma,
jonka suoritus aika on grafiikkarajapinnan suomissa puit-
teissa rajaton, tai siis niin hidas, ettei sitä voi missään ta-
pauksessa hyväksyä.

Tässä on oikeastaan samanlainen dilemma kuin aseval-
mistajalla: vaikei välttämättä laillisella, täytyy hänen ai-
nakin metafysisellä tasolla ottaa vastuu kaikesta, mitä
aseella ikinä saatetaan tehdä.

Mutta se nyt vain sattuu olemaan hinta seuraavan suku-
polven middlewaren kehittämisestä. Kuten Grammaton
Cleric First Class John Preston sanoo: "I pay it gladly."

Toisinsanoen me annamme teille luvan ampua itseänne
jalkaan. Tiedän, että jotkut enginet esim. värjäävät ruudun
punaiseksi, jos framerate tippuu liiaksi, mutta se tuntuu
turhan holhoavalta. Monipuoliset profiointityökalut, jotka
Vision3D 2.0 tulee tarjoamaan, saavat luvan riittää.

Mutta mitä helvettiä?

Valtteri saapuu paikalle hiukan myöhässä vähäeleisesti
anteeksipyöydellen. Mutta kuka ihmeen juippi hänellä on
mukanaan? Ja olenko nähnyt hänet aiemmin?

Lähin kuvaus jonka pystyn muodostamaan on, että ky-
seinen henkilö muistuttaa tiettyjen tässä nimeltämainitse-
mattomien kotimaisten miespuolisten pop- ja iskelmälau-
lajien yhdistelmää ja epäsikiötä. Tumma, piikikäs tukka ja
etova paikalleenjähmettynyt hymy palavat verkkokalvoil-
leni ja aivosoluihini pysyvästi.

Perkele.

Mikä tahansa tämän henkilön rooli Vision3D 2.0:n kehi-

tyksessä onkin, se ei voi luvata hyvää.

Valtteri rykäisee ja alkaa puhua. "Kuten jotkut teistä ehkä tietävät, olen siirtymässä toisiin tehtäviin..."

Tunnen dissosioituvani, irtautuvani ruumiista jonnekin katonrajaan leijumaan. Tämä ei ole todellista. Ymmärrän täydellisesti petoksen laajuuden: tieto pimitettiin minulta, ja tämä kyseinen juippi on mitä ilmeisimminkin Valtterin seuraaja. Muuta mahdollisuutta ei juuri ole. Vastenmielisyntä on, että jopa Valtteri itse petti minut!

Tämä tietää potentiaalista tuhoa ja kuolemaa.

I see darkness spreading.

Nimittäin, vaikka en voi aina kehua Valtterin teknistä osaamista ja ymmärtämystä, hänellä on yksi ominaisuus, jonka tärkeyttä ei voi vähätellä: hän on aina ollut sataprosenttisesti puolellamme maailmaa, idiootteja, hörhöjä ja asiakkaita vastaan. Hänen tapansa on raadella ja testata koodimme läpikotaisin, kunnes se on sopivan parkkiintunutta ulkomaailman raadeltavaksi.

Mutta jos hän poistuu, mitä tästä tulee? Syvällinen ulkoiseen habitukseen perustuva analyysini osoittaa, että Juippi ei missään nimessä voi kyetä samaan. Ennenminkin tuo luotaantyöntävä, epäilyttävä hymy osoittaa, että hän on jo lähtökohtaisesti vihollisen puolella.

Toisaalta olen taipuvainen hätiköityihin johtopäätöksiin, mieluiten negatiivisiin sellaisiin. Jos muistan oikein, mielsin Valtterinkin (kuten kaikki muutkin työtoverini) pitkään viholliseksi. Aina kun kuulin yhdenkään sanoista "arkkitehtuuri", "yksikköttestaus" tai "integraatio", lukkiu-duin ja jumiuduin täydellisesti.

Tosiohjelmoijille on nimittäin olemassa vain koodi, sen absoluuttinen flow assembly-tasolla, jota ei vain pysty pakottamaan tuollaisiin jäykkiin puitteisiin. Vanha mantrani oli: "Koodini on niin laadukasta, ettei sitä tarvitse testata. Se on myös niin nopeaa, ettei sitä tarvitse profiloida." Tuoloin käytin gotoa ja setjmp / longjmp:ia vapaasti, enkä pelkästään virhetilanteiden hallintaan.

Olen ehkä hieman kypsyntynyt noista alkuajoista, enkä oi-

keastaan enää miellä itseäni tosiohjelmoijaksi. Myönnän, että olioabstraktion tuomat hyödyt tuottavuudessa ja toimintavarmuudessa (kunhan ei sorruta johonkin saatanoliseen pelleilyyn) peittoavat sen, että optimoidaan käsin hyppytaulukoita ja unrollattua assemblykoodia. Sitäpaitsi oppini ovat peräisin ensimmäisten Pentiumien ajalta; nyt on vain parempi luottaa C++ -kääntäjään.

“...Jarmo astuu nyt tilalleni pääohjelmoijaksi, ja hän tulee johtamaan Vision3D:n kehitystä tästä eteenpäin. En epäile, etteikö hän, niinkuin tekin, tule pärjäämään loistavasti.”

Minä epäilen kaikkea.

Mutta ehkä voin antaa Juipille mahdollisuuden. Yhden mahdollisuuden. Jos tämä olisi kuusnelospeli, hänellä olisi tasan yksi elämä. Eikä 10000 pisteen kohdalla saa lisäelämää.

Onhan minulla joka tapauksessa tuleva SM3.0-pohjainen renderöintipipeline Direct3D9:n ja vaihtoehtoisesti Direct-3D10:n alaisuudessa. Sitä ei kukaan voi ottaa minulta pois. Koska jos se otettaisiin pois, koko Vision3D ampuisi itseään jalkaan ja olisi sama jättää toteuttamatta. Me kaikki voisimme saman tien lähteä helvettiin täältä.

Ja jos tarkemmin ajattelen, oikeastaan toivonkin, ettei tämä ole liian helppoa. Toivon konfliktia ja sotaa, jota Valtterin kanssa ei alkuaikojen jälkeen täysinmittaisesti ole ollutkaan.

Miten se nyt menikään?

Tarina alkaa, kun sankarilla on ongelma.

Aivan. Suuri Pelisuunnittelija jossain, tai siis oikeastaan vaikeustasoa dynaamisesti säättävä skripti on katsonut, että nyt tällä tyypillä on liian helppoa, joten spawnataanpa vihollinen.

Mutta jokin häiritsee minua syvemmillä tasolla. Ok, ei siinä mitään, jos Juippi on tuleva nemesikseni, mutta miksi Valtteri oli vielä vähän aikaa sitten innoissaan Vision3D:n repimisestä palasiksi ja rakentamisesta uudelleen, jos oli jo tehnyt päätöksensä lähtemisestä? Ellei se sitten ollut jo-

tain pirullista näyttelemistä, joka olisi täysin hänen luonteensa vastaista? Tai sitten en lopulta tunne häntä ollenkaan.

Ehkä kaikki selviää vielä.

Jarmo kääntyy tiimin yleistä rintamalinjaa päin. "Hei vaan kaikille. Kestää varmaan jonkin aikaa, ennen kuin teistä tuntuu millään tavalla, että olen osa tiimiä, mutta meillä on loistava softaplatfom rakennettavana, ja suuri haaste edessä, joten yritetään, ettei anneta sen häiritä liikaa. OK?"

Vaimeaa nyökyttelyä. Pahemminkin voisi alkaa.

"Täällähän on menty aika agilella linjalla, ja minä en aio sitä muuttaa. Eli jos joku pelkää, että siirryttäisiin vesiputoukseen, se ei tule ainakaan minun puolestani tapahtumaan."

Täytyykö agile tulkita tässä keljuiluksi koodin laadusta? Myönnän kyllä, että ajoittain kehitys on ollut niin ketterää ettei mitään rajaa, eikä suinkaan hyvässä mielessä: kurinpalaus on ehdottomasti paikallaan. Mutta jos otetaan kirjaimellisesti, itseasiassa vesiputousmallillakin on paikansa. Kun suunnitellaan vaikka mannertenvälisen ydinohjuksen reaaliaikaista käyttöjärjestelmää. Mutta olkoon, Vision3D on siitä melko kaukana.

Valtteri poistuu neuvotteluhuoneesta. Tämä kai voi vain tarkoittaa, että vallanvaihto on nyt saatettu täysipainoisesti loppuun.

"Valtteri briiffasikin minua laajasti Vision3D:n nykyisestä rakenteesta ja puutteista, ja kävin kotiläksynä läpi API-dokumentaation, eli uskon olevani jossain määrin asioista kärryillä. Mutta jotta me kaikki olisimme varmasti samalla sivulla, tehdäänkö niin, että jakaudutaan kahteen ryhmään. Toinen listaa, mikä järjestelmässä on tällä hetkellä hyvää, ja toinen, mitä huonoa. Sitten kirjataan ne tuohon taululle. Pyritään pitämään tämä lyhyenä. Viiden minuutin pitäisi riittää."

Jaha, vai että tällaista. No, tavallaan ihan hauskaa, paitsi että Jarmo osoittaa minulle paikan hyviä puolia pohtivassa

ryhmässä. Näkemykseni petoksen syvyydestä jäsentyy tarkemmaksi: olen varma, että Vision3D:n arkkitehtuurin liisäksi Valtteri on tehnyt tarkahkon selonteon negatiivisuuteen taipuvaisesta luonteestani, ja tällä minua nyt yrite-tään kohtalaisen läpinäkyvästi ohjata positiivisempaan ajatteluun.

Jonkinlaista lohtua tuottaa se, ettei minun sentään tarvitse pohtia yhdessä Kaarlon kanssa, vaan tämä ottaa koväänisesti toisen ryhmän johdon käsiinsä. Kaarlo omaa ärsyttävän kyvyn hallita tilannetta kuin tilannetta, ja ratkaista mikä tahansa koodausongelma ennustettavalla ja konemaisella tehokkuudella. Emme ole mitä parhaimmissa väleissä; itseasiassa voin jopa väittää, etten ollut läheskään niin synkkä ja kyräilevä, ennen kuin jouduin hänen kanssaan ensimmäisiä kertoja tekemisiin. Mutta siitä se lähti.

“Se ei kaadu kovin usein,” toteaa Hannele, UI- ja juniori-ohjelmoijamme.

Murahdan epämääräisesti. Kyllä se kaatuu, jos tietää mitä tehdä, tai jos asettidata on sopivasti malformoitua, mutta myönnän, että loppukäyttäjiltä ei ole tullut 1.4-versiosta kovinkaan paljon crash-dumpeja. Siihen verrattuna 1.3 oli aivan hirvittävä tapaus; UDB:tä (undefined behavior) invoketettiin siihen malliin, että sitä koodia korjatesa sain tuottamani äänisaasteen johdosta väliaikaisen liikanimityksen “sumusireeni.” Mutta aika kultaa muistot: nyt tuon vaiheen, josta on jo noin kahdeksan kuukautta, mieltää vain jonkinlaiseksi siirtymäriitiksi.

Äärimmäisesti pinnistelemällä saan lopulta mietittyä yhden positiivisen piirteen. “API on kohtalaisen helppo oppia,” sanon. Totta on myöskin, ettei sillä saa kovin paljon aikaan, mutta jos on luova, voi rajoituksia kiertää kaikenlaisilla hackeilla.

Näiden kahden pointin jälkeen seuraa pitkä hiljaisuus, kun kukin pohtii tahollaan. Toisessa tiimissä haukutaan kilpaa kaikkea mahdollista nimeämiskäytännöistä moduulirakenteeseen. Hauskaa ja tervettä. Nauttikaa nyt kun

voitte, mutta pian te joudutte keksimään, miten ne puutteet oikeasti korjataan.

Viisi minuuttia kuluu umpeen. Minä käyn kirjaamassa hyvät puolet whiteboardille, ja Kaarlo kirjaa huonot. Kokonaisuudessaan lista näyttää nyt tältä:

- + Suhteellisen vakaa*
- + API yksinkertainen ja helppo oppia*
- + Skaalautuu heikkotehoisemmillekin koneille*
- Luokkien nimeämiskäytäntö*
- Singletonit*
- Moduuleja on liikaa ja ne ovat raskaita, vaikka toteutavatkin vähän*
- Globaalit muuttujat, dokumentoimattomat riippuvuudet ja sivuvaikutukset*
- Grafiikkakoodissa liikaa legacya, joka sotkee ja monimutkaistaa*
- Hidas multipass-renderöinti*
- Sceneobjektien jäykkä perintähierarkia*
- Scene graph, joka yrittää olla kaikkea (hierarkia, state-sorttaus, okklusio)*
- Verkkokoodi tehotonta ja tuhlaa kaistaa*
- Huono laajennettavuus: verkkoviestit, uudet entitytyypit, uudet skriptikomennot hankalia lisätä*
- C-API plugineille*
- Huono ODE-fysiikka (ODE internal error!)*

Siinäähän sitä on. En nyt tiedä, onko tämä vielä läheskään riittävän syvällisellä tasolla, että siitä voisi lähteä vetämään jumalallisia linjauksia, mutta ainakin se osoittaa jonkinlaista ameebaa tai kummituseläintä suurempaa kykyä analyysiin ja kriittisyyteen.

Tosin listassa on jotain, joka osoittaa ehdotonta type-ryyttä. C-API dynaamisesti ladattaville plugineille on ainoa järkevä, turvallinen ja hyväksyttävä tapa ajaa mielivaltaista ulkopuolista koodia (ja siltikin siitä on seurannut aivan riittävästi kaatumisia.) Tässä nyt ilmeisesti haetaan sitä,

että pluginien pitäisi päästä suoraan keskustelemaan C++-luokkien välityksellä, mutta te perkeleet ette nyt vain tajuua, ettei ole olemassa C++:n standardia binaariformaattia, vaan kukin kääntäjä voi toteuttaa jutut ihan niin kuin sitä huvittaa. Siitä taatusti saadaan UDB:tä aikaan ja helposti, tai sitten pitää shipata sisäänrakennetut moduulit kaikilla yleisimmillä kääntäjillä käännettynä. Huom. esim. Visual Studion saman version eri servicepackit voi laskea tässä eri kääntäjäksi. Jos tähän mennään, minä en kyllä aio olla siitä millään muotoa vastuussa.

Jarmo alkaa höpöttämään jotain tästä yhteenvedosta. En usko että sillä on oikeasti mitään merkitystä, ja aivoni kytkeytyvät suurimmaksi osaksi pois päältä.

Keskustelemme työnjaosta ja tiimin organisoinnista. Kuu-lemma virallisessa scrumissa yhden tiimin suositeltava maksimikoko on rajallinen, jottei kommunikaatiokanavia tule liikaa, ja jotta taskilistat pysyisivät hallittavina. Ilmeisesti me nyt aiomme mennä enemmän "by the book" kuin aikaisemmin Valtterin aikakaudella.

Se ei lupaa mitään hyvää.

Sillä oikeastaanhan agile-hapatus nostaa yksinkertaisen asian ikään kuin valtaistuimelle. Jokaisessa projektissa pitää olla tapa hallita kokonaistaskilistaa, tietää mitä juuri tällä hetkellä on työn alla, ja pystyä päättämään mitä tehdään seuraavaksi, mutta ei noista jutuista tarvitse tehdä suurta numeroa.

Jos projekti on kompleksinen ja kommunikaatiota on odotettavissa paljon, niinkuin nyt, on seitsemän suositeltava maksimimäärä. Tähän vedotaan esim. ihmisen työmuistin seitsemän allokaatioyksikön (plus miinus kaksi) koolla. Konsensukseksi muodostuu lopulta, että jakaannumme arkkitehtuuri- ja alijärjestelmätiimeihin. Minä koen yllättävän "ylennyksen", enkä ole siitä lainkaan mielissäni. En ole enää Graphics Subsystem Technical Lead, vaan Lead Architect.

No huh huh.

Asiaa ei ole vielä lausuttu ääneen minun tai kenenkään muun toimesta, mutta tässä tehtävässä kaikenlaiseen yleisen rakenteen säätöön tulee menemään niin paljon aikaa, etten luultavasti voi ottaa renderöintijärjestelmää hoidettavakseni.

Tässä kohtaa on ehkä syytä siteerata erästä kotimaista nuclear metal -yhtyettä. He osaavat kuvata paljon paremmin kuin minä, miltä minusta juuri nyt tuntuu. Katkelma sanoituksesta menee seuraavasti:

VSP – VSV – VSP – VVV.

Kirjaimien merkitykset jätettäköön kunkin omatoimisesti dekodattavaksi.

Joka tapauksessa arkkitehtuuritiimiin kuuluu minun lisäkseni Veli-Matti, jonka en ole ikinä nähnyt tekevän mitään muuta kuin jonkinlaista tietokanta- tai XML-serialisointikoodia, ja kuinka muutenkaan, tämänhetkisistä nemesiksistäni toinen eli Kaarlo.

Periaatteessa minun pitäisi kyetä nauttimaan tilanteesta (siis jos Kaarloa ei huomioida), koska tulemme käytännössä olemaan projektin ylijumalia, jotka sanelevat muille, miten asioiden pitää mennä. Hyvin luultavaa on myös, että jos hommat eivät toimi, tai seuraavaksi vuorossa oleva alijärjestelmä on vain liian kriittinen kenen tahansa olmin säädettäväksi, me suoritamme kirurgisen tai ninjalogisen iskun.

Toisin sanoen voisi todeta, että me olemme Bastard Programmers From Hell, jos vain haluamme olla.

Mutta en silti voi olla mitenkään tyytyväinen.

Ei renderöintijärjestelmän omakätistä uudelleenkirjoitusta, jota olin odottanut kuin kuuta nousevaa.

Sen tekee joku muu.

VSP.

Emme vielä mene yksityiskohtiin taskijaosta, mutta veikkaan, että tehtävään tulee sijoittumaan Sebastian. Oikeastaan pidän hänestä, sillä tarpeen vaatiessa hän on taipuvainen oikeamieliseen negatiivisuuteen ja kyräilyyn kuten minä, ja hallitsee hommansa.

Mutta hän on tehnyt yhden fataalisen väärän valinnan elämässään.

Ellei hän ole totaalisesti uudelleen kouluttanut itseään naurettavan lyhyessä ajassa (sanotaan n. viikko), hän hallitsee OpenGL:n huomattavan hyvin ja Direct3D:tä ei juuri lainkaan. Juuri hän on työskennellyt niska limassa versio 1.4:n OpenGL-renderöinnin saattamiseksi edes jotenkin hyväksyttävälle tasolle, mikä toki on miehen työ.

Mutta silti se on väärä valinta. Elämässä ei saisi tehdä virheitä. Ja nyt tuosta virheestä tulee ehkä kärsimään koko Vision3D.

Miettikää! Jumalallinen Direct3D SM3.0- tai SM4.0-renderöinti, versus kökkö ja ikuisesti epävarma OpenGL 2.0 (OpenGL versio 3 ei oikeastaan ole vielä käytettävissä, koska ohjelman kuitenkin pitää toimia pari-kolmekin vuotta vanhoilla koneilla.) Koskaan et voi tietää, mitä ajuribugeja tällä kertaa tulee vastaasi, tai jos NVIDIA:n rikollisen leväperäinen GLSL-kääntäjä päästää läpi virheitä, jotka muiden valmistajien ajureilla räjähtävät.

Jarmo ilmeisesti lukee ajatukseni.

VSV.

“Onko teille tuttu artikkeli nimeltä Why you should use OpenGL and not DirectX?”

Mutinaa ja nyökkäilyä.

Heille, jotka eivät kyseistä blogikirjoitusta tunne, referoin sen osoitteen tässä. Lukekaa ja kauhistukaa.

<http://blog.wolfire.com/2010/01/Why-you-should-use-OpenGL-and-not-DirectX>

Tuho ja kuolema.

Nyt ollaan ilmeisesti tekemässä jo administratiivisella ja poliittisella tasolla linjausta OpenGL:n puolesta, sivuuttamalla varsinaisten teknisten asiantuntijoiden mielipiteet. Jos sieluni ja mieleni eivät vielä aikaisemmin olleet riittävän mustat, nyt ne taatusti ovat.

“Se on kyllä täyttä paskaa,” sanon. “Kirjoittaja itsekin

myöntää, että OpenGL:n käytöstä seuraa enemmän ongelmia. Minä en ainakaan halua, että Vision3D olisi marttyyri sen puolesta, että GL-ajureihin alettaisiin panostamaan taas enemmän. Ei se tapahdu kuitenkaan.”

“Meillä on nyt ihan strategisella tasolla linjaus, että crossplatformiin, kuten myös OpenGL:ään, aletaan panostamaan versio 2.0:n myötä enemmän. Tripla-A Direct3D-pelimootoreita on jo maailma täynnä, eikä meidän kannata yrittää suoraan kilpailla niitä vastaan niiden omilla aseilla. Ja mitä minä olen lukenut, Cg näyttää aika ihanteelliselta ratkaisulta siihen, miten samaa shaderkoodia ajetaan rajapinta-agnostisesti.”

Juippi! Shaderit ovat pyhiä, et ole pätevä puhumaan niistä! Ja pyyhi tuo vastenmielinen virne heti naamaltasi. Vaikka minulla ei olekaan mitään NVIDIA:n näytönohjainten suorituskykyä vastaan, on Cg aika lailla itse isästä Perkeleestä. OpenGL:ää käyttäessä se kääntyy alkeelliseksi ARB-assemblyshaderkoodiksi. No, tosin pitää muistaa että vahvojen epäilysten mukaan Suuren Vihreän ajurit kääntävät myös GLSL:n ensin Cg:ksi, eli performanssimenetyks on sama joka tapauksessa.

Mutta, kuten totesin, jos linjaus on tehty jo hallinnollisella tasolla, on turha tuhlata energiaa tuulimyllyjä vastaan taistelemiseen. Tässä valossa on ehkä mukava, että saan pysyä renderöinnistä kaukana, jos se käytännössä täytyy tehdä OpenGL:n ehdoilla – tai vieläkin parempaa – jos Vision3D ei tule käyttämään Direct3D:tä enää ollenkaan.

Mutta mitä helvettiä minä sitten teen?

Mihin muuhun minulla oikeastaan on mitään professionaalista intohimoa? Täydellisen yleispätevään ja joustavaan scenerakenteeseen? Verkkoliikenteen äärimmäiseen optimointiin? Ehkä, ja ehkä.

Mahdollinen alkava spiraali kohti multifailia, vain tunti projektin alusta. Ehkä minun pitää vain dissosioitua. Olen täällä vain töissä.

Muistan lukeneeni tarinan vangista, joka kesti ruumiis-

ta irtautumalla tukahduttavan tiukkaan ja raskaaseen pakopaitaan käärimisen, ja oli aina session jälkeen virkeä niinkuin mitään ei olisi tapahtunut. Ehkä alan soveltamaan samanlaista metodia.

Olen itseasiassa kokeillut ruumiista irtautumista; sehän tapahtuu rentouttamalla kaikki lihakset ja visualisoimalla irtautuminen joko ylöspäin kohoamalla, sivulle kääntymällä, tai sängyn ”läpi” putoamalla. Joskus olen päässyt vaiheeseen, jossa koko kehoa kihelmöi, ja alkaa kuulua omi-
tuista, sähköistä kohinaa, mutta siihen se on jäänyt. No, ehkä tämä motivoi harjoittelemaan lisää.

Mutta tiimijaosta vielä: olen melko varma, että jos Valtteri olisi vetänyt tätä, olisi hän itse kantanut pääarkkitehdin hattua, ja me muut olisimme vapautuneet silpomaan alijärjestelmiä uusiksi sydämemme kyllyydestä. Se olisi luultavasti johtanut ad-hoc-ratkaisuihin arkkitehtuurin suhteen, mikä ei ehkä kolmen vuoden kehityksen jälkeen olisi enää suotavaa.

Mutta toisaalta arkkitehtuuriin voi myös käyttää paljon aikaa, ja siitä silti ehkä tulee pelkkä suolituotos. Hankala sanoa.

Tätä miettiessäni saan oudon, hetkellisen idean, joka tuntuu jotenkin tutulta, mutta silti järjettömältä. Ympyränmallinen softa-arkkitehtuuri, jolla ei ole alkua eikä loppua.

Mitä ihmettä?

Lounasaika. Kuten tavallista, syön antisosiaalisesti, eri aikaan kuin muut, ja vielä varmuuden vuoksi erillään muiden vakiopöydästä. Appelsiiniriisi maistuu tänään erityisen tahdasmaiselta, erityisen kitkerältä, ja täytyy sanoa, että se sopii mielentilaani täydellisesti.

Nyt ei ole kyse enää jälkiruoan arvioinnista haastavuusasteikolla, vaan siitä, kuinka hyvin se tukee synkkyyttäni. Mietin, harmittaisiko minua, tai ehtisikö minua harmittaa, jos tukehtuisin siihen ja Vision3D 2.0 jäisi siten minulta tekemättä.

Vai pysähtyisikö tukehtuessani aika, ja suunnittelisin

jumalallisen enginen, jota en kuitenkaan ikinä saisi toteutettua?

Se saattaisi jo harmittaa.

Palautan tarjottimen. Jos universumin syklit ovat pysyneet vakioina, seuraavaan appelsiiniriisiin on neljä kuukautta. Mutta pullavanukkaaseen ehkä vain kuukausi. Olisiko mahdollista, että Vision3D olisi jo kuukauden päästä jotenkin inhimillisessä kunnossa? Vai jos Juippi olisi jo luovuttanut havaittuaan astuneensa liian isoihin saappaisiin? Tai koko projekti räjähtänyt tuhannen paskaksi omaan mahdottomuuteensa? Tavallaan on mielenkiintoista nähdä, mihin syvyksiin tässä vielä laskeudutaan. Ylöspäin luultavasti ei juuri ole asiaa.

Aloituspalaveri jatkuu. Varsinaiseen sprint planningiin ei vielä mennä, sillä se on huomenna, mutta sen sijaan keskustelemme arkkitehtuurista korkealla tasolla. Kerrostettu arkkitehtuuri, bla bla bla. Ei riippuvuuksia päinvastaiseen suuntaan tai tasojen yli hyppimistä, bla bla bla. Matala kytkentä ja korkea koheesio. Kaikkihan me oikeastaan tiedämme nämä jutut, mutta käytännön tasolla, varsinkin kun aika on vähissä, usein kuitenkin sorrutaan mitä ihmeellisimpään räpellykseen.

Ehkä nimeänkin itseni Arkkitehtuurinatsiksi, joka rankaisee rikkureita ankaralla kädellä ja mustankiiltävän saappaan tarkasti suunnatulla potkulla. Ehkä niin pystyn löytämään elämälleni merkitystä. Tai sitten vain kyräilen hiljaa ja itsekseni, kun teen jumalallisia search- and replace-committeja versionhallintaan. Kun pistän yön tai viikonlopun aikana kaiken uusiksi mitään varoittamatta, mitään selittämättä: kaikki vain toimii mystisesti paremmin kuin ennen.

Arkkitehtuurikeskustelun jälkeen siirrymme käytännön asioihin. Vastahakoisesti myönnän, että Jarmo saa tällä kertaa hyväksyntäni, minkä täytyy luultavimmin johtua siitä, että Valteri on aivopessyt hänet osittain. Sen sijaan, että aloittaisimme vaivalloisen inkrementaalisen refakto-

roinnin niin, että joka hetki testaisimme, ettei tule regressioita aiempaan versioon nähden, me oikeasti räjäytämme koko sontaläjän, ja lähdemme sen jälkeen pistämään palasia takaisin kokoon, tai kirjoittamaan niitä uusiksi. Tottakai oletuksena on, että alkuvaiheeseen ei ole tarkoitus mennä kauaa, vaan tärkeimmät subsysteemit saadaan nopeasti pystyyn. Osoittaisihan se muuten melkoista inkompetenssia näin useamman vuoden jälkeen.

Nyt, hieman myöhässä, minulle itse asiassa tulee mieleen toinen Vision3D:n hyvä puoli. Vähän ulkoisia dependenssejä. Open Dynamics Engine -fysiikkamoottorin lisäksi boost ja lua ovat oikeastaan ainoita, jotka voi edes laskea, muut ovat liian pieniä ja viattomia. Boost siis toteuttaa erinäisiä apufunktioita alustariippumattomasti, kuten esim. shared pointterit, mutexit ja säikeet. Se myös sisältää mitä ihmeellisintä hörhöilyä esim. tieteellisen laskennan ja parsereiden osalta, mutta ylimääräisistä osista ei ole harmia jos niitä ei käytä. Ja mikä tärkeintä, boostin voi linkata ohjelmaan staattisesti. Lua puolestaan on skriptikieli.

Toisin sanoen olemme sentään vastuussa omasta koodistamme hyvin pitkälle. Huonomminkin voisi olla. Ehkä kamalimpia ovat opensource-projektit, joissa lähdetään integroimaan kirjastoa kirjaston päälle, tavoitteena esim. ultimaalinen pelimoottori. Tyypillisesti projektin nimi on Open bla bla bla jotain. Kun ollaan niin avointa, niin enkelikuoro laulaa ja maailma pelastuu. Ohjenuorana voisi sanoa, että jos kehitys lähtee siihen suuntaan, että otetaan esim. OGRE-renderöintimoottori, ja sitten abstrahoidaan se niin, että kirjoitetaan wrapperi joka funktiolle (ilmeisesti ajatuksena, että tämä muka helpottaisi renderöintimoottorin vaihtoa myöhemmin, vaikka abstraktiosta on nyt tullut 99-prosenttisesti OGRE-spesifinen) niin ole hyvä, tee maailmalle palvelus ja darwinoi itsesi heti, ennen kuin aiheutat enemmän pahaa!

Ehkä tästä vielä tulee jotain. Hmm, jos minä alan ajatella myönteisesti, täytyy tässä olla koira haudattuna. Varmaan heti kohta tipahdetaan taas entistä syvemmälle.

Palaverin päättyessä Jarmo kehottaa meitä lukemaan kotiläksynä johdatuksen scrum-metodologiaan:

<http://www.crisp.se/henrik.kniberg/ScrumAndXpFromTheTrenches.pdf>

Dokumentti on kohtuullisen mittava, 105 sivua, tosin materiaali näyttää siltä, että suurimmaksi osaksi sen voi vain pikakelata läpi.

Metodologia on oikeastaan aika ikävä sana. Siitä tulee melkein mieleen kultti. Vaikka scientologia. En ihmettelisi, jos scrumin vannoutuneet harrastajat tarkistaisivat toisiltaan E-mittarilla, ettei mieleen ole pesiytynyt epäpuhtaita ja scrumin vastaisia ajatuksia. Sitten esim. joka kuukausi epäpuhtain koodari uhrattaisiin agile-ylipapin toimesta alttarilla kynttilänloimotuksessa, samalla kun muu kultti hymisisi mantroja. Hyi!

Illalla TV:stä tulee Enter the Ninja. Alussa valkoinen ninja (Franco Nero:n esittämä Cole) esittää senseilleen ninjitsun yhdeksän voiman taso, eli kuji-kirin yhdeksän symbolista viiltoa. Viillot, joita jokaista vastaa monimutkainen käsi-merkki, ovat seuraavat:

RIN - Mielen ja ruumiin voima
KYO - Energian suuntaaminen
TOH - Harmonia universumin kanssa
SHA - Itsensä ja muiden parantaminen
KAI - Vaaran aavistaminen
JIN - Muiden ajatusten tietäminen
RETSU - Ajan ja avaruuden herruus
ZAI - Luonnonvoimien hallinta
ZEN - Valaistuminen

Itse elokuva ei ole kovin erikoinen, mutta viillot jäävät kiertämään kehää mieleeni. Haen alun kohtaoksen YouTubea, katson sen useaan kertaan, ja yritän toistaa käsi-

merkkejä. Ne eivät oikein tahdo onnistua. Nimien lausuminen on helpompaa: pyrin jäljittelemään sensein ärinää koväänisesti, vaikka kello onkin jo yli kymmenen. Toivon nimenomaisesti, että naapurini pitävät minua tämän johdosta outona.

Olisiko Vision3D:n kehitys mahdollista jaksottaa viiltojen mukaisesti? Tai jos me kehitymme projektin aikana hallitsemaan kaikki nuo jutut? Olisi ehdottomasti jotain, jos kykenisi lukemaan toisten ajatuksia, ja hallitsemaan avaruutta ja aikaa.

Tästä tuleekin mieleeni, että muistaakseni scientologian korkeimmat tasot lupaavat myös ajan ja avaruuden hallitsemisen. Ehkä operoivat scrum-thetaanit pystyvät saavuttamaan saman. Sitten voisi hidastaa aikaa ja koodata vaikka kokonaisen enginen niin, että aikaa kuluisi ehkä vain muutama päivä, jos sitäkään. Tosin se olisi aika inhoitavaa, jos engineä koodatessa ei olisi itse vanhentunut yhtään, mutta kun palaisi todellisuuteen, olisikin kulunut vuosia!

Yöllä, unessa, kuji-kirin sijasta ympyränmallinen arkkitehtuuri palaa vainoamaan ajatuksiani. Sillä ei ole alkua eikä loppua, ja se pyörii ympyrää loputtomiin.

Käärme, joka syö omaa häntäänsä.

Maailmankäärme, Jörmunganrd!

Päivä 2.

Totuuden hetki. Virallisen ja pitkän kaavan mukaisesti pidetty sprint planning, niin by the book kuin voi olla.

Scrumissa pitää olla ns. "product owner," joka hallitsee kokonaistaskilistaa eli product backlogia, ja priorisoi tehtävät näin päättäen, mitä seuraavaksi itse asiassa tehdään. Yleensä product owner on asiakkaan edustaja. Nyt, kun teemme sisäistä middleware-kehitystä, tulee roolin ottamaan toimitusjohtajamme Tuomas.

Tuomas muistuttaa lähinnä nuorempaa versiota saksalaisesta TV-sarjan etsivästä Horst Schimanskista, viiksiä myöten. Hän on itseasiassa suurimman osan ajasta varsin leppoisa, kuten esikuvansa, mutta nousee tarvittaessa lähes pelottavaan raivoon. Version 1.3 aikana sitä tapahtuikin aika paljon. Tuomas itse harrasti ohjelmointia aikana, jolloin MS-DOS, DOS-extenderit ja VESA SuperVGA-näyttötilat olivat kovimpia ikinä.

Joskus otettuaan kuppia hän kehuskelee, kuinka teki 640 kilotavun rajapyykin ylittämiseksi oman DOS-extenderin, joka vaihtoi protected moden ja real moden välillä ohjelmisto- tai laitteistokeskeytyksen tapahtuessa paljon tehokkaammin, kuin esimerkiksi DOS/4GW, tai DJGPP-kääntäjän käyttämä go32-extenderi. Vaihdoksen tapahtuessa prosessorin tämänhetkinen tila talletettiin pino-maiseen tietorakenteeseen, jonka koko täytyi olla ennalta-

määrätty. Sitten, jos varattu tila ylittyi, soitettiin PC:n piiperiä ja resetoitiin kone, kun ei oikein voitu muutakaan. En nyt tiedä, voiko tuota pitää vielä tosiohjelmointina. Ei oikeastaan, koska järjestelmällä ajetut sovellukset tuskin olivat kriittisiä esim. ihmishenkien ylläpidon tai risteilyohjusten navigoinnin kannalta.

Windows-ohjelmointia Tuomas ei enää opetellut, vaan siirtyi suosiolla hallintotehtäviin. Luultavasti juuri ohjelmointiharrastuneisuudesta hänelle on jäänyt jalat maassa -asenne, jota ilman Vision3D olisi jo multifailannut moninkertaisesti. Tosin joskus hän innostuu jostain melko epäolennaisesta shitistä, kuten silloin, kun meidän täytyi kii-reellä implementoida stereorenderöinti. Vision3D:lle ei tietääkseni vielääkään ole toteutettu yhtään sitä käytävää sovellusta.

Koko backlog on mittava, viisi sivua pientä tekstiä, ja joka rivillä yksi taski. Suurimmaksi osaksi niitä ei sitä pait-si vielä ole paloiteltu tarpeeksi pieniksi paloiksi toteutettavaksi asti. Uskoisin, että Valteri on kehittänyt listan suurimmaksi osaksi itse, viimeisenä kontribuutionaan Vision3D:n kehitykselle.

Arvioimme, kuinka paljon työtunteja meillä on käytettävissä. Scrumissa yhden sprintin pituus on tavallisesti kahdesta neljään viikkoa: päädyimme aloittamaan varovasti kahdella viikolla, jotta ei menetetä paljon aikaa, jos ilmenee, että täytyy tehdä jotain drastisia muutoksia, tai havaitaan, että ollaan menty aivan vikasuuntaan.

Arvio on helppo laskea: kaikki panostavat sataprosenttisesti. Everyone fights, no-one quits. Kaarlo ja Sebastian ilmoittavat panostavansa 110-prosenttisesti, mutta Jarmo ei suostu huomioimaan tätä.

Backlogissa on myös valmiina taskien prioriteetit, jotka Tuomaksen on täytynyt huolellisesti assignoida (uskon, että tästä prosessista ei ole selvitty ilman jotain miestä vahvempaa.) Prioriteettilistan yläpäässä on järkeenkäy-vää, mutta silti puuduttavaa tavaraa, kuten:

- *Decide overall architecture for 2.0*
- *Decide a winning set of external components*
- *Startup and shutdown sequence*
- *Plugin architecture design*

Minua huolettaa erityisesti toinen itemi. Mitä ihmeen voittavia ulkoisia komponentteja? Jos minulta kysytään, niiden kanssa voi vain hävitä. Antaa kontrollia pois. Versiossa 1.4 meillä käytännössä on jo kaikki, mitä tarvitaan, mutta osa vain itkettävän huonosti toteutettuna. Nyt kun tiedämme, miten jutut pitää tehdä, pystymme tekemään ne kunnolla, luultavasti puolet pienemmällä koodimäärällä. Mutta huom: edelleen omalla koodilla!

En usko, että Valtteri on keksinyt tuota taskia. Jarmon – siis Juipin, ei pidä unohtaa – on täytynyt ujuttaa se listaan katalasti jälkikäteen.

Alempana prioriteettilistassa on niiden alijärjestelmien prototypointia, jotka varmasti tiedämme, että tulemme tarvitsemaan, kuten renderöinti, graafinen käyttöliittymä, ja skriptaus.

Alamme arvioida taskien kestoa pelikorteilla, käyttäen aikayksikkönä puolta päivää. Alin numero 2 tarkoittaa siis päivässä suoritettavaa tehtävää. Ässä puolestaan tarkoittaa, että taski on niin laaja tai epämääräinen, ettei sen kestoa pysty arvioimaan.

Hannele pitää arviointimenettelyä hauskana. Omasta mielestäni se on aika turhaa pelleilyä. Nopeammin menisi, jos taskit assignoitaisiin ensin suoraan, ja sen jälkeen teki- jä arvioisi, kuinka kauan hänellä menee. Nyt Jarmo pyrkii laskemaan jonkinlaisen ad-hoc painotetun keskiarvon arvauksista, ja siinä kestää.

Olisi vahingoniloisella tavalla hauskaa, jos artistimme olisivat myös mukana arvioimassa, mutta heillä on oikeita töitä, joten heidät on vapautettu. No, ehkä ensi planningissa sitten.

Kuvittelen mielessäni erittäin hitaasti etenevän progress barin.

Lopulta kestot on arvioitu, eli nyt tiedetään, mitä meidän pitäisi saada aikaan seuraavan kahden viikon aikana. No niin, nyt vain tekijät kehiin ja koodaamaan! Turha luulo, itseasiassa. Seuraavaksi jokaiselle taskille päätetään yhteistuumin, miten sen valmius todennetaan, eli miten se demonstroidaan seuraavan planningin yhteydessä. Katson ympärilleni ja totean, että suurin osa meistä – paitsi Jarmo itse – alkaa lähentymään katatonista tilaa. Jos tällainen sessio on odotettavissa joka kerta, en ihmettelisi, jos porukkaa alkaisi vain häipyä.

Viimein päätämme tekijät. Minulle jää yllätyksettömästi yleisarkkitehtuuriin liittyviä hommia, ja lisäksi jotain matantason perustoimintoja kuten esim. virheenhallinta, crash dumpit ja loggaus, mutta Kaarlo-hyväkäs nappaa nokkani edestä pelätyn “winning set of external components”-taskin.

Olisin toivonut saavani sen, jotta olisin voinut liioitellun pitkän päivän tutkimuksen jälkeen todeta: “Emme tarvitse lainkaan ulkoisia komponentteja. Koodaamme meille tärkeät ominaisuudet – ja itseasiassa ne kaikki ovat tärkeitä – itse.”

Oikeastaan sen verran voisin antaa periksi, että fysiikkakirjasto kannattaisi edelleen integroida ulkoisena, kunhan tällä kertaa valitaan parempi kirjasto. 1.x-sarjan ODE-integraation – joka tosin onkin melko alkeellinen ja virhealtis – meille ohjelmoi eräs fysiikan opiskelija, joka ei ole enää elävien kirjoissa. Koko alijärjestelmän yllä leijuu tietynlainen kalmanhaju, eikä siihen kajota kuin pakon edessä.

Session päätyttyä Sodexosta ei saa enää lounasta, joten tilaamme pizzat. Huomatkaa: sen lisäksi, että virallinen scrum on puuduttavaa, se voi nähtävästi käydä myös terveydelle vaaralliseksi. Arvaan, että loppupäivä tuleekin kulumaan täydellisessä koomassa, kun verenkierto siirtyy palvelemaan ruoansulatuselimistöä, ja aivot jäävät vähemmän tärkeinä toissijaisiksi.

Kun pizzat viimein saapuvat, Tuomas ja Jarmo menevät

keskustelemaan CEO:n kansliaan kahdenkesken. Tuomak-
sen elehdinnästä näkee, että hänen raivonsa on nousemas-
sa. Toivon, että kyse on kunnan torumissessiosta ylipitkän
suunnittelupalaverin johdosta. Rationaalisesti ajateltuna
tässä hukattiin vähintään puoli työpäivää.

Päivä 3.

Jos sprint planning ei vielä ollut tarpeeksi hirveä, niin ainakin tämä päivä tulee jäämään muistiin erityisen pahana ja apokalyptisena päivänä. Nyt on tehty jotain mullistavan kauheaa, jolla tulee olemaan kauaskantoisia jälkiseurauksia, ellei päätöstä peruta ja pian.

Kaarlo ja Jarmo (joka myös otti voittavan komponenttisetin pähkäilyn sydämenasiakseen) ovat liittoutuneet yhteen järkyttävän hörhöilyn merkeissä. En ymmärrä, miten se on mahdollista, koska vaikka Kaarlo ei olekaan kovin siedettävä persoonana, on hänen ammattitaitoonsa aina tähän asti ollut mahdollista luottaa.

Mutta ei enää.

Kaksi viattomalta vaikuttavaa kirjainta, jotka kääntävät tämän projektin vieläkin epämiellyttävämpään suuntaan.

Qt.

Miksi ihmeessä?

Utilityfunktioidenko takia? Signal-slot -mekanismin vuoksi? Vai graafisen käyttöliittymäkoodin? Jos mietitään toiseksimainittua, minä voin ensimmäisenä myöntää, että Vision3D:n viesti- ja eventtijärjestelmä, jolla alijärjestelmät kommunikoivat keskenään, on syvältä, ja se pitää upottaa sinne, missä aurinko ei paista. Mutta vain sen korvaamiseksi en missään nimessä menisi integroimaan tätä möhkälemäistä, norjalaisen Trolltechin alunperin kehittä-

mää, Nokian omistukseen sittemmin päätynyttä sovellus-
frameworkia.

Seuraavassa seitsemän faktaa Qt:stä:

1. Qt sisältää sovellusta kuvaavan globaalin singletonin, ns. QApplication-luokan. Tämä tietää saatanallisia ongelmia, jos Vision3D:tä halutaan ajaa osana jotain muuta sovellusta, etenkin jos sitä halutaan käynnistää useita kappaleita saman prosessin sisällä.

2. Qt voidaan lisensoida kaupallisesti tai LGPL-lisenssilä. Kaupallinen lisenssi tulee mahdollisesti olemaan epäyhteensopiva asiakkaidemme tarpeiden kanssa, joten se on poissuljettu. Ja LGPL puolestaan tarkoittaa, että bye bye staattinen linkitys. Tervetuloa kasa rumia dll-tiedostoja!

(Tässä vaiheessa voisin mainita, että LGPL:n tarkoituksena on tarjota käyttäjälle mahdollisuus korvata käytetty kirjasto – tässä tapauksessa siis Qt itse – ilman, että itse sovellukseen tarvitsee koskea. Mutta nyt on kyse C++ -kielisestä kirjastosta, ja jos muistatte, mitä totesin C vs. C++ -kielillä toteutetusta plugin-arkkitehtuurista, koko juttu muuttuu naurettavaksi. C++ binääriyhteensopivuus on äärimmäisen helppo rikkoa, jos esim. luokkaan lisätään muuttuja, ja tällöin ainoa asia, mitä käyttäjälle jää käteen hänen vaihdettuaan uuden version käyttöön, on ei-toimiva ohjelma. Jos hyvin käy, se kaatuu heti, ja jos huonommin, sekoilut havaitaan vasta myöhemmin datakorruption muodossa. Ainoa toimiva tapa siirtyä Qt:n uuteen versioon on kääntää uudelleen koko sovellus, kuten myös kaikki Qt:sta riippuvat kolmannen osapuolen kirjastot.)

3. Kun käänät Qt:n, varaudu siihen, että siinä kestää koko päivä (debug- ja release-buildit.)

4. Qt ottaa haltuun ohjelman pääsilukan. Kaikki siitä eteenpäin, kun sanotaan QApplication.run(), tehdään eventtejä ja signaaleja lähettämällä, ilman että kontrolli enää palautuu pääohjelmalle. Monipuolisesti käytettävissä olevassa middleware-ohjelmistossa tämä ei vain käy! Uskon, että tämän voi ilkeillä hackeilla kiertää, mutta meidänhän oli nimenomaan tarkoitus päästä niistä pois!

5. Qt on hyvin vihamielinen 3D-renderointiä kohtaan. OpenGL:n kanssa käyttöliittymä- ja 3D-renderointi on helppo sovittaa yhteen, mutta muuten joudutaan tekemään jotain järkyttävää hakkerointia, eikä toimivuutta silti voi taata. En ole nähnyt vielä yhtään esimerkkiä, jossa 2D/3D-integraatio olisi tehty oikein ja kestävä kehityksen periaatteiden mukaisesti, varsinkaan Direct3D:tä käyttäen (WolfenQt:tä ei siis lasketa.)

6. Jatkoa edelliseen: jos siis tehdään jotain custom-kikkailua 3D-renderöinnin mahdollistamiseksi esim. siten, että normaali piirto suppressoidaan, ja sensijaan widgetit piirretään tekstuuriin (olen nähnyt erään projektin, jossa näin tehtiin, eikä siitä seurannut mitään hyvää), on varmaa, että alustariippumattomuus muuttuu tämän jälkeen hyvinkin fiktiiviseksi termiksi. Tässä joudutaan nimittäin menemään Qt:n konepellin alle, jolloin sen käyttöjärjestelmäspesifiset toteutukset ja oudot sisäiset hackit tulevat vastaan, eikä se ole kaunista katseltavaa.

7. Jos halutaan ihkuilla iOS-laitteilla (ja "ihkuilu" on minulle erittäin vastenmielinen sana, mutta silti ainoa, joka tulee tässä kyseeseen), ei Qt ainakaan vielä ole käytettävissä niiden kanssa. Jotain iPhone-tuesta Jarmo muistaakseni mainitsi kaksi päivää sitten, mutta se ilmeisesti unohdettiin äkkiä.

Lähden aikaisin töistä, ja menen masentuneen ja voimattoman olotilani johdosta varhain nukkumaan.

KYO - Energian suuntaaminen

Päivä 4.

Kosto.

Se on voimakas ja pahaenteinen sana. Koston voi suorittaa monella tavalla ja monella tasolla. Selvää on, ettei siihen pidä ryhtyä kevyin perustein, sillä se ei yleensä pääty hyvin kenenkään kannalta. Mutta joskus oikeudenmukaisuuden vaatimus on vain niin suuri, että sitä ei voi enää sivuuttaa, eikä muuta keinoa kuin kosto ehkä ole.

Ajattelin, että minun kohdallani toinen viilto – energian suuntaaminen – merkitsisi energian suuntaamista entistäkin tehokkaampaan vihaan ja kyräilyyn, jotta herättäisin pääarkkitehdin ominaisuudessa riittävää pelkoa ja pitäisin kehityksen etenemissuunnan oikeana ja etenemistahdin ripeänä.

Nyt tuo päättely ei voi mitenkään olla oikein.

Jos etenemissuunta on 180 astetta väärin, jos on valittu täysin väärät työkalut, sitä ei mikään kyräily voi korjata. Ei edes Jack Bauer kykenisi kyräilemään – tai no, hän voisi myös kiduttaa tarpeen tullen – niin tehokkaasti.

Minun täytyy ehkä vielä tarkkailla tilannetta pari päivää. Onhan mahdollista, että Qt vielä hylätään ja projektin raiteilleen palaamiseen on vielä mahdollisuus.

Mutta jos niin ei käy...

Silloin on selvää, ettei minulle jää muuta mahdollisuutta kuin suunnata energiani koston.

Minkä muodon kostoni saa, sitä en vielä tiedä. Jos se täyttää sieluni tyytyväisyydellä, parempi niin, mutta jos tuloksena on vain, että kylmä tyhjyyden tuuli puhaltaa lävitsemi tehtyäni sen, mikä on välttämätöntä, niin sitten niin on tarkoitus käydä, ja hyväksyn sen tyynen stoalaisesti. Jos Vision3D on tuomittu, niin parasta mitä voin tehdä, on antaa oikeamielisen retribuition instrumentiksi.

Katson jokseenkin halvaantuneena, kun Sebastian tekee ensimmäisiä QGLWidget-pohjaisia renderöintitestejä. Gouraud-varjostettu laatikko pyörii ruudulla käyttöliittymäelementtien alla suhteellisen liukkaasti. Mutta lisääpä siihen tekstuureja, ja shadereita, ja useampaan eri rendertargettiin piirtämistä, ja katso miten OpenGL-state sekoaa oman renderöintisi ja Qt:n renderöinnin välillä! Vaihtoehtoisesti voit yrittää käyttää useampaa renderöintikontekstia, mutta unohdapa yksikin kontekstivaihto ja näet todella ihmeellisiä juttuja, tai ehkä spektaakkelinomaisen kaatumisen! Tai sitten vain rupea miettimään, miten tekisit saman Direct3D:llä, ja itke.

Pakotan itseni kirjoittamaan Vision3D:n arkkitehtuuridokumenttia. Jokainen rivi, jonka kirjoitan, maistuu pahalta suussani. Sisäinen kriitikkonni huutaa täysillä: kokeilen eri nimeämiskäytäntöjä moduuleille, mutta ne kaikki tuntuvat yhtä huonoilta. Jokainen Manager tai Subsystem on liikaa. Niitä ei vain pitäisi olla ollenkaan.

Tämäkään päivä ei ole hyvä päivä.

Päivä 5.

Nyt tiedän varmuudella, mitä on tehtävä. Keskustelin vielä Jarmon kanssa, selitin Qt:n haittapuolet juurta jaksuen, mutta se oli pääpiirteissään kuin olisi puhunut seinälle.

“On mahdollista, että jos hyödynnämme Qt:n tarjoamia ominaisuuksia täysipainoisesti, saamme leikattua jopa puolet omasta koodistamme pois. Se puolestaan vähentää meidän omaa maintenance-kuormaa huomattavasti,” hän sanoi.

Hän suostuu näkemään vain hyvät puolet. Myönnän, että niitäkin on. Jos olisimme tekemässä mitä tahansa perinteistä GUI-sovellusta, jossa meillä on kontrolli kaikkeen, mitä sovelluksella tarvitsee tehdä, en epäröisi lainkaan. Qt on ihku, tietystä näkökulmasta.

Mutta kun meillä ei ole kontrollia!

Me olemme middleware. Vision3D tarjoaa hörhöille tai vähemmän hörhöille työkalupakin toteuttaa oma 3D-sovelluksensa, ja silloin on aika karua todeta, että niin muuten, asiat pitää tehdä tällä tavalla, koska me satumme puolestaan pohjautumaan tähän sovellusframeworkkiin. Ja niin muuten, fullscreen-tilaa ei oikeastaan ole. Eikä Direct3D-tukea. Tai jos ostatte tämän full source-lisenssin, voitte kokeilla haxoroida sitä sisään itse. Emme ota vastuuta, miten siinä käy.

Tämä ei missään tapauksessa ole minun käsitykseni hy-

västä asiakaspalvelusta.

Mutta kuten sanoin, minä tiedän mitä minun täytyy nyt tehdä. Vision3D:llä tässä kehityssuunnassa ei ole oikeastaan väliä. Sanoudun siitä irti. Kostoni onkin osoittaa, miten asiat olisi pitänyt tehdä.

Enkä aio tehdä tätä vain teoreettisella tasolla, vaan puhtaasti käytännössä.

Ymmärrän nyt, mitä uneni pyörivästä, itseään syövästä maailman-käärmeestä merkitsi. Se ei ollut vain abstrakti *heure* mahdottomasta ohjelmistoarkkitehtuurista, vaan suunnitelma, *piirrustukset* sille, miten ylivertainen – eikä millään muotoa mahdoton – virtuaalitodellisuus-engine toteutetaan.

Jokainen alijärjestelmä riippuu eksplisiittisesti niistä, joita se tarvitsee välittömien velvollisuuksiensa hoitamiseksi. Siitä puolestaan riippuvat edelleen toiset alijärjestelmät. Nämä kaikki yhdessä muodostavat harmonisen syklin, jota voi seurata ympäri ja ympäri ja ympäri... Ainoa kauneusvirhe on, että täytyy valita yksi alijärjestelmä, joka on tavallaan "alkupiste." Sen tarvitsemia riippuvuuksia ei voida täyttää heti konstruktoria suoritettaessa, vaan täytyy ensin alustaa koko ketju, ja sitten vasta antaa kyseiselle alijärjestelmälle viittaukset sen tarvitsemiin järjestelmiin.

Koko systeemi on käännösaikaisesti rakenteeltaan ennaltamäärätty, yhteen exe-tiedostoon käännettävissä oleva, mutta sen laajentaminen on silti helppoa: lisätään vain yksi lenkki lisää, jonka riippuvuudet on hyvin määrätty. Tai useampia lenkkejä. Lisäksi skriptauksen kautta saavutettava laajennettavuus on käytännössä rajatonta.

Tämän kuvauksen jälkeen ei ole epäselvää, minkä nimen annan tälle projektille. Vaihtoehtoja on vain yksi.

Jörmunganrd Engine.

Havaitsen, että voin suorittaa taskejani mekaanisesti, kovinkaan paljon ajattelematta, samalla kun oikeasti mietin

kuumeisesti Jörmungandr:in yksityiskohtia.

Vision3D:n kerrostettu core-arkkitehtuuri muotoutuu ammattimaisesti, mutta yllätyksettömästi. Se tuskin tulee olemaan kaikista tehokkain, tai kaikista vaivattomin käyttää, mutta on silti riittävän siisti ja toimiva, ettei kukaan ainakaan tässä vaiheessa epäile mitään. Omaksun jopa demonstraation vuoksi tehtävässä prototyypisovelluksessa Qt:lle tyypillisen nimeämiskäytännön, jotta minua luullaan tosiuskovaksi. Jos te vain tietäisitte totuuden, hm hm hm hm, ha ha ha ha! No, aikanaan te tulettekin sen näkemään, ja silloin se sokaisee teidät kaikkivoipaisuudellaan. Te ryömitte Jörmungandr:in täydellisyyden edessä, ja kiroatte katkerasti, kuinka hukkasittekaan kuukausia elämästänne ala-arvoisen ohjelmiston kehittämiseen!

Jos Vision3D on pakkopaita, johon minut vangitaan joka päivä, on Jörmungandr henkireikä, minun tapani irtautua ruumiistani. Ainoa asia mikä harmittaa on, etten keksinyt sitä aiemmin, vaan hukkasin pari päivää apatian vallassa.

Ruokailun jälkeen – jälkiruokana oli haastava kahvikiis-seli, jonka tekstuuria ja makua jopa jaksoin analysoida (scrum-palaveria välittömästi seuranneina päivinä mätin vain ruokaa sisääni mitään rekisteröimättä) huomaan mielialani parantuneen selvästi. Ellen väärin havaitse, työtoverinikin huomaavat sen, ja ihmettelevät. Onko Suurkyräilijä lakannut kyräilemästä? Ovatko maailmankirjat sekaisin? Jos te vain tietäisitte! Muahahahaha!

Kun muiden silmä välttää, alan koodata mestariteostani. Periaatteena on, ettei mitään aikaisempaa koodia – edes itse kirjoittamaani – käytetä, vaan kaikki kirjoitetaan tyhjästä, puhtaasti ja absoluuttisesti. En tietenkään ota työpaikalta yhteyttä omaan versionhallintaserveriini, sillä siitä jäisi verkkoliikenneloki, vaan käytän datan kuljettamiseen muistitikkoa.

Kuvittelen joutuvani nielemään tikun ääritilanteessa, ja Jarmon kaivamassa sitä ulos vatsani auki leikkaamalla. Muahaha. Siihen Jarmo ei suinkaan ikinä kykenisi. Tuomas ehkä saattaisikin, mutta hänellä ei ole siihen mitään syytä,

ainakaan vielä.

(Työsopimuksessani lukee, että kaikki koodi mitä tuotan sopimuksen voimassaollessa, olipa se sitten työajan ja -paikan sisä- tai ulkopuolella, on firman omaisuutta, joten tarve salamyhkäisyyteen ja varovaisuuteen tässä vaiheessa on ehdottomasti paikallaan. Koodiani saatettaisiin nimittäin väärinkäyttää osana Vision3D:tä, mitä en missään nimessä voisi hyväksyä.)

Viikonloppu on tulossa. Valmistaudun nollaamaan aivoni semivoimakkaasti; minun kuitenkin täytyy pysyä riittävästi tajuissani, ettei yhtään kokonaista päivää Jörmungandrin kehitystä mene hukkaan. Sitten, kun osaan hallita aikaa ja avaruutta, tätä ongelmaa ei enää ole.

Päivä 8.

Jörmungandr on inkarnoitunut. Voi pojat, se on inkarnoitunut! Myönnän, että suurin osa viikonlopusta meni krapulassa (kauniimmin sanottuna arkkitehtuurin syvempää luonnetta vielä miettiessä), eikä engine vielä tee paljon muuta kuin käynnistää ja sammuttaa itsensä, ja luo siinä välissä Direct3D10-renderöintikontekstin, mutta se on silti alku.

Direct3D9 on (tai pitäisi sanoa oli, sillä sitä ei pidä käyttää, ellei todella ole tarkoitus tukea legacy-käyttöjärjestelmiä, ja Jörmungandrin ei missään nimessä tarvitse tukea) suoraviivainen, joskin rajoittunut grafiikkarajapinta. Siihen verrattuna D3D10 sisältää tietynlaista uhkaa ja arvovaltaa. Voisin jopa sanoa, että se tuo mukanaan häivähdyksen tosiohjelmointia. Versio yhdeksässä renderöintiä hallitaan helpoilla, yksittäisillä komennoilla kuten `SetPixelShader()` tai `SetRenderState()`. Mutta D3D10 vaatii, että luodaan immutable-tilaobjekteja, ja pakataan shaderparametrit constant buffereihin sen mukaan, kuinka tiheästi niitä aiotaan päivittää. Jos valitsee puskurijaon huonosti, ampuu helposti itseään jalkaan suorituskyvyn suhteen. Juuri tällainen ohjelmointi on Jörmungandrin arvon mukaista!

Palaan töihin jatkamaan Vision3D 2.0:n arkkitehtuuriproton kehittämistä. Itseasiassa olen melko varma, että prototyyppiä ei heitetä romukoppaan niinkuin pitäisi,

vaan Jarmo tulee ehdottamaan (tai jopa vaatimaan), että se otetaan sellaisenaan käyttöön lopullisessa koodissa.

Noniin, se on siinä. Kirjaudun google docsin kautta käytettävään sprint backlogiin, ja värjään taskin vihreäksi.

En ole lopputuloksesta erityisen ylpeä, mutta se on silti parannusta 1.4:n pohjakoodiin verrattuna, joka on aika iljettävä kasa höyryävää guanoa. Vanhassa versiossa frameworkin käyttäjä kutsuu Initialize()-funktiota, joka näyttää kyllä ulospäin siistiltä, mutta aiheuttaa sisäisesti sellaisen globaalien muuttujien ja objektien alustamisen sekamelskan, että hyi helvetti! Monesti jouduimme selvittämään alustus- tai tuhoamisjärjestykseen liittyviä mystisiä ongelmia, mutta emme silti räjäyttäneet alustuskoodia tuhannen päreiksi, niinkuin olisi pitänyt.

Minä, Jarmo, Kaarlo ja Sebastian keskustelemme Vision3D:n OpenGL-renderöinnistä. (Tai oikeastaan minä pysyn suurimmaksi osaksi hiljaa.) Sebastian on saanut renderöintitilan tallentumaan ja palautumaan oikein meidän koodin ja Qt:n välillä, ja on nyt valmis aloittamaan varsinaisen renderöntialijärjestelmän kirjoittamisen.

Konsensus on, että halutaan kokeilla deferroitua renderöintiä, sillä se on konseptuaalisesti puhdas, ja sillä pääsee nopeasti alkuun.

Voi helvetti.

Deferrattu rendaus on siis menetelmä, jossa 3D-scenen objektit tarvitsee valaistuksesta riippumatta piirtää vain kerran, siten että niiden kaikki valoja varten tarvittavat ominaisuudet (esim. väri, syvyys, normaalivektorit) piirretään näytönkokoiisiin rendertargetteihin, eli nk. G-bufferiin. Tämän jälkeen valaistushaderit pystyvät toimimaan pelkästään G-bufferia tulkitsemalla, ilman että itse objekteja tarvitsee piirtää uudestaan joka valoa kohti, tai että objektia piirrettäessä pitäisi kerralla osata laskea kaikkiin siihen vaikuttavien valojen efekti.

Sinänsä menetelmä on varsin toimiva, ja vähentää suuresti tarvittavien shader-permutaatioiden määrää. Mutta G-bufferin kirjoittaminen ja lukeminen vaatii kohtuutto-

man paljon näytönohjaimen muistikaistaa. Jos valaistus on yksinkertainen (esim. vain auringonvalo), ollaan vain hidadestettu renderöintiä hyötymättä mitään. Ja minä jos kuka tiedän, että asiakkaat kuitenkin tulevat ajamaan Vision3D:tä alitehoisilla koneilla, ja alkeellisilla sceneillä. Tällaisissa tilanteissa deferrattu renderöinti ei todellakaan loista!

Lisäksi läpinäkyviä objekteja ei kuitenkaan voi piirtää järkevästi G-bufferiin, joten niitä varten täytyy kuitenkin implementoida myös forward-renderöinti. Olisi siis ollut sama toteuttaa se alusta asti!

Deferratusta renderöinnistä on GPU:n kaistan käytön vähentämiseksi myös kevennetty versio, eli ns. light prepass. Tässä jätetään tallentamatta objektien väri G-bufferiin, ja kirjoitetaan vain normaalivektorit ja syvyys, ja mahdollisesti materiaalin kiiltävyys. Valaistushaderit laskevat nyt vain valon värin ja voimakkuuden, eivät suoraan valon vaikutusta objektiin. Objektit joudutaan piirtämään kahdesti: ensin G-bufferiin, ja toisen kerran valaistusakkumulaatiobufferia tulkiten. Vaikka toinen piirtokerta periaatteessa antaa mahdollisuuksia monipuolisempiin materiaaleihin, on suorituskyky joko vain vähän parempi kuin oikeassa deferroinnissa, tai objektimäärän kasvaessa huomattavasti huonompi. Toisin sanoen tämä on kaikista huonoin menetelmä! Kaarlo ehdottaa sitä silti, ja silloin ärähän.

“Perkele, ei ainakaan light prepassia!”

“Unityn versio 3.0 käyttää sitä. Kai ne on profiloineet sen tarkasti, kun ovat valinneet sen täysdeferroinnin sijasta.”

“Et oikeasti halua kaksinkertaistaa batchcounttia!”

“Rauhoitutaan nyt,” Jarmo tulee väliin. “Mitä itse ehdotat?”

“Tehdään niinkuin oli tarkoitus tehdä Direct3D SM3-pollussa: käännetään ubershaderista variaatiot tarpeen mukaan jokaista valoyhdistelmää kohti. Ja Z-prepass ensin, jos näyttää, että siitä on hyötyä.”

“Paitsi että GLSL ei tue shaderien kääntämistä etukäteen, joten siihen saattaa mennä sceneä ladatessa paljon aikaa,” Sebastian toteaa. “Ja sitäpaitsi ei kai voida olla varmoja, milloin shaderista tulee millekin näytönohjaimelle liian monimutkainen, ja tiputaan softarendaukseen. Vanha Leadwerks-engine teki tätä, mutta vaihtoi versiossa 2.0 deferrattuun.”

Leadwerks! Älkää muistuttako minua siitä! Jos me olemme seuraamassa sen jalanjalkia, niin Cthulhu ja Spagettihirviö meitä auttakoon. Tämän Blitz Basicilla kirjoitetun enginen luoja on todistajalausuntojen mukaan melkoinen häirikkö, joka omalla foorumillaan ensin haukkuu, sen jälkeen bännää ne, jotka uskaltavat sanoa enginestä jotain poikkipuolista.

Mutta Sebastian on kyllä aivan oikeassa.

Ei tuollaisten ehtojen kanssa haluaisi elää. Tämän takia olinkin suunnitellut, että OpenGL:llä toteutettaisiin vain pahainen fallback-pipeline (vaikka multypass, en välitä), kun kaikki oikea ja loistava renderöinti tapahtuisi Direct3D:llä.

“Jos pistetään yhden passin valomäärälle joku järkevä raja. Vaikka 3. Tunnistetaan, mikä näytönohjain on käytössä. Jos se on joku oikein huono, niin vedetään sitten multypassina. Ei ihmisillä kuitenkaan ole kuin keskimäärin 1-2 valoa yhteen objektiin vaikuttamassa.”

En pidä omista sanoistani. Ei tuollaiseen pitäisi joutua tyytymään.

“Jos deferred renderingillä saa nopeiten systeemin pysyyn, ja sitä kuitenkin käytetään useissa ihan AAA-tason moottoreissa, minusta näyttää vahvasti, että kannattaa aloittaa sillä,” Jarmo toteaa tarkoituksenaan päättää argumentti autoritäärisesti.

Aloittakaa sitten. En estä teitä.

Mutta minulta jäi vielä toteamatta deferred renderöinnin järkyttävin puute, joka tosin voidaan Direct3D10:llä tai OpenGL 3:lla kiertää mutkien, oman shaderkoodin ja pudonneen suorituskyvyn kautta. Toistan sen muutaman

kerran, jotta se varmasti jää mieliinne.

Ei tukea hardware-antialiasoinnille.

Ei tukea hardware-antialiasoinnille.

Ei tukea hardware-antialiasoinnille.

Ei tukea hardware-antialiasoinnille.

Ei tukea hardware-antialiasoinnille.

Ei tukea hardware-antialiasoinnille.

Ei tukea hardware-antialiasoinnille.

Ei tukea hardware-antialiasoinnille.

Ei tukea hardware-antialiasoinnille.

Ei tukea hardware-antialiasoinnille.

Jotkut enginet tekevät iljettävän blurraustempun, eli havaitaan polygonien rajat kuvasta esim. normaalivektoreita, syvyyttä tai sitten vain värien epäjatkuvuuksia tutkimalla, ja sumennetaan kuvaa raja-kohdissa. Tätä ei voi mitenkään verrata oikeaan hardware-antialiasointiin. Hyi.

Jarmo ja Kaarlo poistuvat työpisteilleen, ja Sebastian alkaa nähtävän innoissaan kirjoittaa G-buffer- ja valaistus-shadereita.

“Profiloi se sitten vaikka jollakin läppärin 8400G-näytönohjaimella. Tai Intelin integroidulla,” sanon. En siis mitenkään avoimen vahingoniloisesti, vaan muka asiallisesti. Mutta jos Sebastian on vähänkään oppinut lukemaan persoonallisuuttani, hän havaitsee piruilun.

Ja lisää on taatusti tulossa.

Päivä 9.

Sebastian pyörittää suurta määrää (n. 500) ei-varjostettuja pistevaloja Sponza Atrium-scenessä. Täytyy kyllä ihailla tyypin nopeutta. Tai sitten hänellä oli valmista koodia takataskussa. Framerate on hyväksyttävä, mutta täytyy huomata, että hänen koneessaan on Geforce GTX480.

(Valmiista koodista eli ns. koodarin henkilökohtaisesta työkalupakista voisi kysäistä, että onhan sinulla varmasti siihen tekijänoikeudet? Jos teit sen esim. edellisessä työpaikassasi, oikeudet ovat luultavimmin työnantajalla, ei suinkaan sinulla itselläsi. Sebastianin naamanmuoto ja muu habitus onkin jotenkin rikosluontoinen. En kuitenkaan kysy, sillä eihän minua juurikaan liikuttaisi, jos Vision3D:n tuleva inkarnaatio joutuisi lainopillisiin ongelmiin.)

Tällä välin Hannele koodaa UI-alijärjestelmän prototyyppejä. Qt:n omia widgettejä käyttämällä se on kohtuullisen suoraviivainen taski, ainakin niin kauan kun pysymme QGLWidgetin käytössä, emmekä tule sotkeneeksi Qt:n omia piirtomekanismeja. Mutta jotain GraphicsScenestä ja GraphicsViewistä hän silti kiroilee. Muistan joskus selanneeni Qt:n issue trackeria, enkä ihmettele, jos noista kahdesta luokasta vielä löytyy omalaatuisia bugeja.

Vanhat Vision3D:n versiot toteuttivat itse vaatimattoman määrän UI-widgettejä, jotka tosin useimmiten riitti-

vät niihin vaatimattomiin sovelluksiin, jota middlewarella on tähän mennessä tehty. Mutta graafinen käyttöliittymä onkin osa-alue, johon minulla ei häpeäkseni ole ylimaallisen hyvää ratkaisua. Toisaalta tuntuu, että UI-järjestelmiä on koodattu jo tarpeeksi, eikä niihin enää kenenkään kannattaisi kuluttaa aikaa täysin oman toteutuksen tekemiseen. Mutta silti harvemmin löytyy juuri sopivaa valmista toteutusta, varsinkin sellaista, joka olisi toteutettu 3D-renderöintiystävällisesti. Jos lataat jonkin UI-kirjaston kokeilua varten, ja heti ensimmäisenä sieltä hyppii silmille 10 manageriluokkaa ja singletonia, ja oma resurssienhallinta-järjestelmä, jota et saa integroitua omaasi, niin laattahan siinä meinaa lentää. Jos painisimme todella raskaassa sarjassa, niin saattaisimme esim. integroida flash-pohjaisen Scaleformin. Mutta uskon, että sen lisenssimaksut ovat jostain järkyttävää.

Vision3D:n nykyisellä polulla Qt:n itsensä tarjoama käyttöliittymätoiminnallisuus on luultavasti riittävää. Ei projekti siihen tule kaatumaan.

Hyvin arvattavasti Veli-Matti puolestaan sai koodattavakseen konfiguraatiojärjestelmään liittyviä juttuja. Epäilemättä hän toteuttaa ne XML:ää käyttäen. Onneksi ei sentään relaatiotietokannalla. XML-parserit voidaan jakaa käsitykseni mukaan purgatorioon, esikartanoon ja Infernoon, ja Qt:n vastaava on ehkä esikartano-osastoa. Toimii, muttei ole kovinkaan miellyttävä.

Mutta eihän minun tarvitse välittää, mitä kukin tekee. Itseasiassa olenkin käyttänyt eilen ja tänään liikaa aivosolujani Vision3D:n parissa, ja liian vähän Jörmungandrin. Tähän on tultava muutos.

Inspiraation kasvattamiseksi katson illalla puolella silmällä Ninja III: The Dominationin, samalla kun koodaan Jörmungandrin Direct3D10-rendersysteemiä. Constantbufferien ja stateobjektien allokointikoodi alkaa olla hyvässä kuosissa. Elokuva on suoraan sanoen aika luokattoman paska, eikä suinkaan inspiroi minua, mutta ei juuri hait-

taakaan. Bonusta tosin riivaukseen kykenevästä mustasta ninjasta, ja kuji-kiristä.

Kun menen nukkumaan, ulkona on alkanut sataa lunta. Näinköhän talvi tulee aikaisin? Sen on oikeastaan inspiraationaalista aikaa, joten ei haittaa, jos se pitenee.

Päivä 10.

Huolimattomuuteni ja hübrikseni kostautuvat. Kaarlo näkee vilahdukselta, kun minulla on auki Jörmungandrin koodi Visual Studio 2010:ssä. Vision3D:tä kehitetään vielä virallisesti 2008-versiolla, koska 2010:ssä on suorituskykyongelmia intellisenssen kanssa ainakin osalla koneistamme.

“Mikäs tuo on?”

Tämä on varmaan sama tunne kuin Dyynissä, kun Jessica (ja myöhemmin Paul) Atreides nielee suuren annoksen hiekkamadon myrkkyä, ja subjektiivisen ajantajun pysähtyessä hän joko tunnistaa myrkyn ja transformoi sen kehonsa sisällä vaarattomaksi, tai kuolee.

Mieti!

Jos teet minkään äkkinäisen liikkeen, kuten esim. Visual Studion sulkemisen, Kaarlo aavistaa taatusti, että jotain juonittelua on tekeillä.

Pakotan itseni rauhoittumaan.

“Joku tämmöinen hörhöilyengine netistä. Katson vain, miten se toteuttaa shaderfragmenttien kokoamisen ja kääntämisen.”

Pyrin hivuttamaan tuolia ja itseäni näytön eteen siten, ettei Kaarlo näe solutionin nimeä, ja voi siten tarkistaa, ettei sellaista itseasiassa löydy netistä.

“Mutta tuohan on Direct3D-koodia.”

“Joo, mutta samat periaatteet on sovellettavissa OpenGL:ään. Tosin Sebastianilla on deferrattu pipeline hyvin hallussa ja työn alla, joten... Takaisin oikeisiin töihin.”

Nyt suljen Visual Studion.

“Joo.” Kaarlo kääntyy ja kävelee pois. Huokaisen sisäisesti helpotuksesta. Tämä ei ehdottomasti saa toistua! Jörmungandrinn on lupa tulla julki vasta, kun se on jumalallisen valmis, tai kun siinä toimivat täydellisesti ainakin seuraavat alijärjestelmät:

- *Renderöinti*
- *Input*
- *Scenemalli, sisältäen myös latauksen ja tallennuksen*
- *Fysiikka*
- *Verkkoliikenne (client, server ja headless server -toimintatilat. Myös clientside prediction ja lag compensation.)*
- *Skriptaus*

Scenemalli onkin mielenkiintoinen aihe. Epäilemättä se tulee eteen myös työn puolesta seuraavassa sprintissä, ja vaikka onkin lapsellista ajatella niin, haluaisin tehdä Jörmungandrissa juuri päinvastoin, kuin Vision3D:ssä tullaan tekemään.

Perinteisesti Vision3D:ssä on ollut jäykkä perintään perustuva malli. Alimmalla tasolla on SceneObject, josta peritään PositionableObject, josta peritään DrawableObject, josta peritään esim. MeshObject, AnimatedMeshObject, ja LightObject. Käyttäjä voi määritellä uusia C++ -kielisiä luokkia mille tasolle haluaa, mutta skriptausta varten on olemassa vain hardkoodattu luokka ScriptableObject, joka on käytännössä PositionableObject, joka osaa ajaa lua-skriptejä. Aika mielivaltaista, että skriptillä ajatellaan aina olevan 3D-positio maailmassa.

Jos halutaan mennä kehityksessä eteenpäin, on looginen ja yllätyksetön valinta tehdä entiteettikomponenttijärjestelmä. Siinä jäykkää perintää ei tarvitse olla, vaan jokainen scenen objekti voi koostua mielivaltaisesta määrästä

komponentteja.

Sen minäkin luultavasti haluaisin Jörmungandriin, mutta jos Vision3D toteuttaa saman, tekee se ikäänkuin ideasta pahanmakuisen. Pitäisi tehdä jotain, joka on samalla äärimmäisen eleganttia ja järjestelmän käyttäjille mukavaa, mutta sisältää myös tosiohjelmoinnin kuparinkatkuisen maun, ja on sisältä niin jumalallinen, ettei siihen kuka tahansa uskalla koskea (bonuspisteet, jos minusta itsestäniikin alkaa parin kuukauden kuluttua tuntua sen suhteen ulkopuoliselta.)

Mutta vielä kestää, ennen kuin Jörmungandrin scenemalli on varsinaisesti ajankohtainen. Sitä ennen on paljon tekemistä: muistinhallinta (jos kuvittelette, että tyydyn new:iin ja delete:een sellaisena kuin Visual Studion runtime-kirjasto ne toteuttaa, olette väärässä. Tulen kirjoittamaan useita custom- ja pooled-allokaattoreita sen mukaan, mitä tarvitaan. Periaatteeni on, että jos normaalilanteessa framen aikana kutsutaan yhdenkään kerran tavanomaista new:iä tai deleteä, olen epäonnistunut), hierarkkinen suorituskykyprofilointi, materiaaliabstraktio ja statesorttaus piirtoa varten, frustum culling, occlusion culling, octree-toteutus, shadow mapping, mahdollisesti myös animaatio...

Ja ennenkaikkea übershaderien pilkkominen fragmenteiksi, ajonaikainen uudelleen kokoaminen, ja kääntäminen täytyy saattaa hyväksyttävälle tasolle! Nyt se on vasta alkutekijöissään. Tämä sisältää vielä monta kutkuttavaa ongelmaa, kuten constant-rekisterien mahdollisimman optimaalinen varaaminen. En missään nimessä aio käyttää rajoittavaa Direct3D effect framework:ia. Itseasiassa periaatteellinen tavoitteeni olisi, ettei D3DX-kirjastoa käytettäisi lainkaan, koska se sisältää ennenminkin aloittelijoille sopivia apufunktioita esim. kamera- ja projektiomatriisien laskemiseen, ja kantaa siten mukanaan epämiellyttävää stigmaa, mutta shaderien ajonaikainen kääntäminen on ilman sitä hieman hankalampaa, ellen sitten toteuta omaa HLSL-kääntäjäni. Tämä ei sentään ole Jörmungandrin kannalta

oleellista, ja hukkaisi vain aikaa.

Tietysti yksi vaihtoehto on, että kaikki permutaatiot todella käännettäisiin ennalta, ja enginen mukana tulisi vain esikäännetty shader cache. Shader cachen aionkin toteuttaa, mutta en siten, etteikö sitä voisi ajonaikaisesti täydentää.

Havaitsen liikettä takanani, ja joudun keskeyttämään jumalalliset pohdintani.

Jarmo hakee tuolin ja istuu tuoleeni.

Kiroan äänettömästi.

“Aiotko seuraavaksi paneutua plugin-arkkitehtuuriin?”

Aihe on minulle vastenmielinen. Parasta on, jos plugin-eita ei olisi ollenkaan (kuten Jörmungandrissa tulee olemaan), ja C-API -pluginit ovat semihyväksyttäviä, mutta jos on kerran yhteisen konsensuksen mukaisesti todettu, että C++ -plugin-eita tarvitaan, niin tuetaan niitä sitten.

All your UDB are belong to us.

Undefined behaviourin käsite on oikeastaan mielenkiintoisempi teoriassa kuin käytännössä. Käytännössä UDB on usein joko täysin harmitonta (kuten esim. liian suuret bit-shiftaukset), tai välittömän kaatumisen aiheuttavaa (kuten nullpointer-access.) Mielenkiintoisin ja hankalimmin jäljitettävä tapaus on tietysti se, kun salakavalasti sotketaan muistia, joka ehkä tulee esiin muistinvarauksen yhteydessä heap corruptionina, jos sittenkään.

Mutta teoreettisesti undefined behaviour tarkoittaa, että kun C- tai C++ -kielisessä ohjelmassa tehdään toiminto, jonka seuraukset ovat määrittelemättömät, on koko ohjelman suoritus siitä eteenpäin määrittelemätön, ja periaatteessa mitä tahansa voisi tapahtua. Olisi siistiä, jos tuo mitä tahansa olisi esimerkiksi, että ohjelmoija kuolisi. Olen joskus yrittänyt keskustella aiheesta työtoverieni kanssa, mutta idea on selvästi heille ei-toivottua informaatiota.

Mutta kuten totesin, aivan sama miten Vision3D:n pluginit toteutetaan. Olen päättänyt olla aiheuttamatta turhia ristiriitoja, jotta energiani suuntautuu mahdollisimman tarkasti Jörmungandr-in kehittämiseen.

Aktivoin siis sivupersoonani, joka seuraa lammasmaisesti projektin konsensusta. Se oli alitajuisesti olemassa jo aikaisemmin, mutta vasta nyt olen antanut sille eksplisiittisen nimen: Jens.

“Joo, miksipä ei.”

“Oletko ajatellut, miten se toteutetaan?”

“Koska näyttää siltä, että Qt on tullut jäädäkseen, olisi Qt:n pluginien käyttäminen järkeenkäyvä vaihtoehto.”

Voi helvetti, että minä vihaan Jensiä! Niin taitava en ole dissosioitumaan, etteikö kriittisempi persoonani puolisko pysyisi jatkuvasti tietoisena siitä, mitä tapahtuu.

“Mukava kuulla, että hyväksyt jo Qt:n. Kyllä, samoilla linjoilla olisin itsekkin. Rupeatko heti hommiin, sinullahan ei ilmeisesti ollut mitään muuta majorimpaa kesken?”

Ainakin se tilanteessa on siedettävää, että koska Qt jo tarjoaa lähes kaikkea, mitä ikinä voi kuvitella (paitsi no esim. valmiin 3D-renderöintiengineen), emme ainakaan joudu pelkästään plugin-tukea varten integroimaan uutta kirjastoa.

“Joo.”

Tämä riittää Juipin tyytyväisenä pitämiseksi.

Takaisin Jörmungandrin pariin, mutta varovasti.

Kirjoitan ehkä myöhemmin päivällä plugin-systeemin rungon. Jos Jarmolla olisi enemmän asiantuntemusta, hän osaisi vaatia taskien paljon nopeampaa suorittamista. Tosin periaatteessa alkuvaiheessa onkin hyvä, ettei hommia tehdä liian kiireellä, vaan mietitään vähän, mitä ollaan tekemässä. Ja se nyt on ainakin positiivista, että tuotoksemme ovat jo heti pääosin koodia, sen sijaan, että tuottaisimme sivu toisensa jälkeen UML-diagrammeja.

Se olisi jo helvetin yhdeksäs piiri!

Päivä 11.

Sebastian on saanut nyt valmiiksi kaikki perusvalotyypit: `directional`, `point` ja `spot`, tosin edelleen vielä ilman varjoja, ja ensimmäisen version materiaaliabstraktiosta, johon kuuluu `blendmoden`, `cullingmoden` ja `depthtestin` / `depthwriten` valinta, sekä `diffuse`, `normal map`, `specular map` ja `emissive map`-tekstuurit. Tyypillistä peruskauraa. Jos materiaalissa ei ole `emissiveä`, eli se ei valaise itseään, riittää yksi passi objektin piirtämiseen. Itsevalaisu lisätään ylimääräisessä `renderpassissa` `deferratun` valaistuksen summaamisen jälkeen.

“Piirto tapahtuu vielä `glDrawArrays()`-kutsulla, jotta sain jotain nopeasti kasaan,” Sebastian myöntää. “Seuraavaksi aion toteuttaa `vertex buffer`-objektit kunnolla. Tosin aion säilyttää `vertex array`-polun `fallbackina`.”

“Miksi ihmeessä? Eikö `OpenGL 2.0:ssa` ne ole jo standardina? Ei kai me sentään alempia versioita aiota tukea, eihän niissä ole `GLSL`-tukea kuitenkaan. Ja eihän `deferrattu` `rendaus` toimisi kuitenkaan järkevästi,” ihmettelen.

“Niin tuota, me puhuttiin tuossa Jarmon kanssa, että `Intel GMA950`-tuki pitäisi kuitenkin säilyttää. Niitä kuitenkin on vielä käytössä. Mutta se on tulevaisuudessa sitten. Teen varmaan `vertex lighting`-systemin `rupunäytönohjaimille`. Senhän voi tehdä ihan `fixed functionina`, ei tarvitse pelleillä `ARB-assemblyshadereilla`. Ja `VBO`:ista voi tulla jotain

mysteribugeja ihan GL2-tason korteilla, Second Life -viererissäkin oli optiona ottaa ne pois päältä.”

Voin kuvitella tuskasi!

Ellet sitten saa jotain kieroutunutta mielihyvää ala-arvoisten GPU:iden tukemisesta. Ja Second Lifeä nyt ei pitäisi missään nimessä kenenkään meistä edes mainita. Sen piirrokoodi on jotain aivan hirveää: olen itse tämän todennut synkimpinä hetkinäni.

Mutta fixed functionista pitäisi jo ihan periaatteen vuoksi päästä eroon. Herrajumala, nyt on 2010! Intel GMA-ajurit ajavat kuitenkin vertex shaderit keskusyksiköllä, joten jos sen tukemista ajatellaan, aivan sama mitä tekee. Metsään mennään kuitenkin.

Kuinka ollakaan, myös Jörmungandr on tässä vaiheessa edennyt materiaalisysteemin toteuttamiseen (muistinhallinnan ja cullingin ehtii nyt tehdä myöhemminkin; ne eivät stimuloi aivosolujani ainakaan tällä hetkellä riittävästi.)

Haastetason lisäämiseksi en tyydy perinteiseen jakoon, jossa eksplisiittisesti määrättäisiin kussakin renderpassissa käytettävä vertex ja pixel shader, vaan vaihdoin Unity 3:n tyyliseen surface shader-systeemiin, jossa abstraktisti määritellään laskutoimitus, joka määrää materiaalin käyttäytymisen valossa. Pari esimerkkiä:

```
surface DiffMapped  
input in.Position float4  
input in.TexCoord float2  
input in.Normal float3  
texture diffMap 0  
float3 albedo = tex2d(diffMap, in.TexCoord)  
float3 normal = in.Normal  
out.Color = applyBlinnPhong(albedo, normal)
```

```
surface DiffNormalMapped  
input in.Position float4  
input in.TexCoord float2  
texture diffMap 0
```



```
texture normalMap 1  
float3 albedo = tex2d(diffMap, in.TexCoord)  
float3 normal = unpackNormal(tex2d(normalMap, in.  
TexCoord)  
out.Color = applyBlinnPhong(albedo, normal)
```

Kaikki valaistuksen laskemisen likainen työ piilotetaan apply<valaistusmallinimi> -funktioiden sisään. Toki ne myös ovat täysin kustomoitavissa tarpeen mukaan. Surface shader tulkitaan HLSL:ksi, ja sitten tarvittavat HLSL-fragmentit käännetään valoyhdistelmäpermutaatioiden mukaan. Tämä on lähes liikuttavan jumalallista. Voi pojat, olisittepa olleet järkevempiä Vision3D:n kanssa, niin tämä kaikki olisi nyt osa sitä! Mutta kun ei, niin ei.

Saatuani plugin-järjestelmän prototyypin valmiiksi minulla ei juuri ole järkevää tekemistä ennen ensi viikon tiistain sprinttiplanningia (paitsi muutama ankean puuduttava dokumentointijuttu), joten kiertelen ympäriinsä seuraamassa työn etenemistä, muka kiinnostuneena.

Kaarlo on tekemässä hälyttävän kieroja template-magiaa omien tietotyyppien rekisteröimiseksi skriptausjärjestelmään. Tämän alijärjestelmän prototyyppi on yllätyksellisesti QtScript-pohjainen. Qt:n oma skriptimoottori toteuttaa ECMA-262 -spesifikaation, mikä käytännössä tarkoittaa, että skriptit tulevat olemaan Javascriptiä. Qt:ssa kaikki "raskaat" objektit (esim. UI-widgetti) periytyvät QObject-luokasta, ja niiden ns. slot-funktiot rekisteröityvät automaattisesti, joten ne ovat suoraan kutsuttavissa skriptistä. Mutta pelkästään dataa sisältävien objektien, esim. 3D-vektorin ei missään nimessä haluta olevan QObject:eja, joten ne täytyy silti esitellä manuaalisesti. Kaarlon koodi on niin ovelan näköistä, että tulen lähes kateelliseksi.

Yleisesti minun täytyy sanoa, että Vision3D on itseasiassa etenemässä pelottavan nopeasti kaikilla rintamilla. Kehitys ei ole pakosta menossa hyvään suuntaan, eikä laatu ehkä ole kummoista, mutta silti se tarkoittaa, että minulla ei ole Jörmungandrin kanssa varaa päivänkään lep-

suiluun, ellen halua sen jäävän jälkeen. Yhdeksän yhtä vastaan on aika brutaalia, täytyy myöntää.

Tottakai voisin sabotoida aktiivisesti projektin etene- mistä, mutta se ei kuulu kostosuunnitelmaani.

Ainakaan vielä.

Loppupäivästä arkkitehtuuritiimimme + Jarmo kokoon- tuu ankeaakin ankeampaan palaveriin. Tarkoitus on miet- tiä Vision3D:n arkkitehtuurin syvempiä yksityiskohtia, mutta lopulta se menee aika yleisen tason jaaritteluksi, eikä kovin paljon saada aikaan. Kahden tunnin jälkeen luo- vutamme lopulta. Kai siinä jotain informaatiota saatiin vaihdettua.

Vastapainoksi arkkitehtuuripalaverille koodaan surface shader-järjestelmää kuumeisesti koko illan, ja sitten läpi yön. Raivoni alkaa nousta; kuuntelen tämän hyödyllisen tunteen ylläpitämiseksi black metallia (mm. Enochian Crescent:iä) kuulokkeilla hyvin suurella äänenvoimakkuu- della. Saattaa olla, että otin liian kunnianhimoisen lähesty- mistavan. Perinteinen vertex- ja pixel shader-fragmenteis- ta rakentuva järjestelmä olisi jo valmis.

Jotain vikaa HLSL:n generoinnissa on vieläkin. Tekstuu- rit indeksoituvat väärin. Voisin hardkoodata testitapaukse- ni toimimaan, mutta haluan, että systeemi toimii oikeasti oikein. Tiedän kyllä, että olisi parempi mennä nukkumaan: virhe ei luultavasti selviä tämän yön aikana.

Perkele.

Lopulta ongelma ratkeaa: kyse oli yhdestä ylimääräi- sestä pilkusta. Systeemi toimii nyt täydellisesti, ja kello on vain noin 5.45.

Perkele.

Tiedän, että kun menen nukkumaan nyt, tulen herää- mään zombimaisen olotilan vallassa. Työpäivä tulee ole- maan erittäin tärkeä, ja kahvia tulee kulumaan massiivi- sia määriä. Onneksi viikonloppu on tulossa.

Päivä 12.

Unenpuute aikaansaa kaikenlaisia jänniä asioita. Mm. kylmyyden sietokyky vähenee huomattavasti. Voin aavistaa, että jos haluan pitää Jörmungandrin riittävän etenemistahdin yllä, tämä oli vasta ensimmäinen monista all-nightereista. Itseaiheutettu crunch on crunchin omalaatuinen, mutta samalla jaloin ja valaitsevin muoto.

Jos siitä selviää hengissä (ja järjissään), that is.

Vision3D:n kehityksessä ei varsinaista crunchia ole ollut, lukuunottamatta 1.3-version korjaamista. Mietin vieläkin, mikä siinä oikein meni pieleen, ja räjähti käsiin. Luultavasti ominaisuuksia tuli vain liikaa uusia liian lyhyessä ajassa. Jotain vaikutusta saattaa olla myös sillä, että se oli ensimmäinen versio, jolla kukaan oikeasti yritti tehdä mitään monimutkaisempaa. Aikaisemmilla versioilla toteutimme lähinnä omia projektejamme.

Nyt näyttää, että lumi on tullut jäädäkseen, ainakin toistaiseksi. Joutuu ajamaan jonkin verran varovammin. Saa-vun työpaikalle treemorin vallassa, mutta silti ajoissa.

Mahdollisten epäilysten vähentämiseksi (alkuperäinen taskimääräni oli kevyehkö, ja ne yleensä ensin junnasivat paikallaan mystisesti, sitten valmistuivat yhtä mystisesti) päätän ottaa ylimääräisen taskin project backlogista. Neuvottelen jo paikalla olevan Jarmon kanssa, ja sovimme, että otan renderöintijärjestelmän octree- ja culling-osan käsit-

telyyni. Culling tarkoittaa tarpeettomien objektien (esim. näkökentän ulkopuolelle jäävien, nk. frustum culling, tai toisten esineiden takana olevien) piirtämättä jättämistä. Octree puolestaan on tietorakenne, jolla kolmiulotteista avaruutta jaetaan rekursiivisesti aina puolet pienempiin osiin X, Y ja Z-akselien suunnassa. Tällä nopeutetaan spataalisia kyselyjä (tyyliin: mitkä esineet ovat tietyllä alueella x.)

Tähän tietysti liittyy taka-ajatus siitä, että käytän saman koodin myös Jörmungandrissa. Tai oikeammin minun täytyy ensin koodata ominaisuudet omaan moottoriini, ja siirtää ne sitten Vision3D:hen, koska kaikki, mihin Vision3D nykymuodossaan koskee, saastuu.

En olekaan katsonut Sebastianin renderöintiprototyypin koodia aiemmin, ja nyt, kun katson, toivon että voisin pyyhkiä sen aivoistani.

```
void RenderPointLightShape (const Vector3df pos, const
Vector3df color, float radius)
{
    glUseProgram (sPointLightShader);
    glUniform3f (sLightColor, color.x, color.y, color.z);
    glUniform3f (sLightPos, pos.x, pos.y, pos.z);
    glBegin (GL_TRIANGLES);
    for (c = 0; c < sphereShapeVertices.size(); c++)
    {
        glVertex3f (sphereShapeVertices [c].x * radius,
            sphereShapeVertices [c].y * radius,
            sphereShapeVertices [c].z * radius);
    }
    glEnd ();
}
```

Koodi on yksinkertaisesti hirveää.

Todellakin prototyypitasoista pahimmassa mielessä.

Ajatukseni ovat ristiriitaiset. Olin pitänyt Sebastiania taitavampana ja ylpeämpänä koodarina. Tällaista imme-

diate mode-sontaa ei todellakaan pitäisi päästä lopulliseen versioon, mutta ei myöskään kirjoittaa edes prototyyppiin. En toisaalta halua aloittaa kriegiä hänen kanssaan, mutten myöskään haluaisi korjata kaikkea itse.

On mahdollista, että Sebastian loistaa vain olemassaolevaa koodia paikatessaan, mutta hänen tuottaessaan uutta jälki on kamalaa.

Jokatapauksessa minkäänlaisen culling-järjestelmän lisääminen tällaiseen sekasotkuun on tuhoontuomittu yritys, ellei sitten pudotaudu sen tasolle, mutta tällöin huonon koodin määrä vain kasvaa.

Ehkä minun pitää vain tehdä koodi ensin Jörmungandriin ajattelematta lainkaan, miten sen saa siirrettyä. Toisaalta aikaa ei ole paljon: sprint planning on tiistaina. Ei tietysti haittaisi, jos taski olisi silloin vielä kesken, koska silloin se vain siirtyisi, mutta ajatus sotii silti ammattiylpeyttäni vastaan.

Tai sitten: ehkä on vain iskettävä kuin ninja.

Kirjoitan ulkomuistista yksinkertaisen, mutta toimivan loose octree-toteutuksen, ja siihen yksinkertaisen frustum-kyselyn. Haluaisin, että Jörmungandr sisältäisi vain jumalallista koodia, mutta tämän täytyy nyt kelvata. Koodia voi kyllä parannella jälkeinpäin. Ja sitäpaitsi todellinen haaste on occlusion cullingissa, eli piiloon jäävien esineiden havaitsemisessa.

Työpäivän tullessa täyteen frustum culling toimii Jörmungandrissa riittävän hyvin. Maanantaina huononnan sen Sebastianin koodia varten ja commitoin Vision3D:hen. Ongelma ratkaistu.

Tai ei oikeastaan.

Poistuessani ulkoilmaan tajuan, että tämä ei saa toistua. Jos annan periksi Jörmungandrin laatuksereistä samoin, kuin Vision3D:ssä annetaan periksi, mitä minulle jää käteen? Millä tavalla Jörmungandr olisi silloin parempi? Jos perustus on mätä, kaikki mitä sen päälle rakennetaan tulee mätänemään ja sortumaan. Päätän, että octree-koodi

täytyy kirjoittaa kokonaan uudestaan.

Päivät 13 - 14.

del Octree.cpp
del Octree.h

Nyt tuntuu jo paljon paremmalta. Kirjoitan octree- ja frustum culling-koodin uudestaan Jörmungandrin edellyttämällä laatutasolla, ja samalla kirjoitan myös perus-scenenodejen toteutuksen (testasin materiaalien renderöintiä hardkoodatuilla piirtokomennoilla, mutta nyt siihen ei ole enää tarvetta.) Octree-koodin sisimmistä silmukoista tulee nyt niin kimuranteja, etten usko ymmärtäväni niitä täysin enää kuukauden päästä, ja puolen vuoden päästä en ollenkaan. Juuri tällaista Jörmungandrin koodin tulee olla! Siinä välissä nukun univelkaa pois ja katson seuraavat puhdistavat elokuvat:

- *Men Behind the Sun*
- *Martyrs*
- *The Human Centipede (First Sequence)*

Päivä 15.

Sebastian kehuu perjantaina kirjoittamaani koodia (jota siis vielä huononsin, ennen kuin laitoin sisään versionhallintaan.) En tiedä, mitä pitäisi ajatella. Onko hän vilpitön, vai onko tämä jotain kieroja peliä? Ovatko muut jo täysin tietoisia suunnitelmistani ja Jörmungandrista? Ovatko he jo tasolla *JIN - Muiden ajatusten tietäminen*? Nyljetäänkö minut kohta elävältä?

Päätän aloittavani seuraavaksi CPU-rasteroidun occlusion cullingin kirjoittamisen. Tässä siis tarkoituksena on, että valikoidut 3D-objektit (esimerkiksi talon seinät) piirretään keskusyksiköllä pienikokoiseen surfaceen, esim. 256 x 160 kuvapistettä, siitä luodaan hierarkinen Z-pyramidi, ja tämän jälkeen voidaan tarkastella bounding boxien perusteella, mitkä kaikki esineet peittyvät näkyvistä. Etuna näytönohjaimella suoritettavien occlusion queryjen käyttämiseen on, ettei tuloksia tarvitse odottaa, eikä tule sotkeneeksi GPU:n normaalia toimintaa.

(Jos pelaatte jotain Unreal 3-pohjaista peliä, esim. Bioshock:ia, ja olette tarkkana, voitte nähdä tilanteita, joissa jokin objekti jätetään piirtämättä yhden framen ajaksi. Tässä on kyse juuri siitä, että kyselyn tulokset saadaan jäljessä.)

Oman softarasteroijan, varsinkaan nopean sellaisen, kirjoittaminen ei ole aivan lastenleikkiä, ja tehtävän haas-

tavuus ehkä auttaa minua pysymään järjissäni.

Sprint planning on tulossa. Saa nähdä, kuinka hydyttävän kaavan mukaan se vedetään huomenna. Ai niin, siihenhän liittyy kaikkien valmistuneiden taskien demoaminen. Sen ainakin täytyy olla puuduttavaa.

Asunnolle päästyäni ryhdyn kirjoittamaan kolmionrasterointirutiinia. Aluksi se saa käyttää hidasta liukulukulasentaa ja tavanomaista scanline-pohjaista piirtoa, mutta tulen optimoimaan sen jollain seuraavista menetelmistä: fixed point-laskenta, SSE-assembly, 8 x 8 grid-pohjainen piirtäminen.

Masentaakseni itseäni ajan rutiinia debug modessa, jolloin se pyörii hitaammin. Jotta occlusion culling-testejä olisi mahdollista tehdä, täytyy jokaiselle kuvapisteelle tallettaa syvyysarvo. Käytännössä voidaan laskea $1/Z$, jota voidaan interpoloida lineaarisesti pitkin kolmion pintaa. Tällöin ei tarvitse suorittaa hidasta jakolaskua joka pikseliä kohti.

Ennen nukkumaanmenoa luen dokumenttia Rendering With Conviction: The Graphics of Splinter Cell:

http://www.selfshadow.com/talks/rwc_gdc2010_v1.pdf

Tässä myös käsitellään occlusion cullingia, mutta minulle jää epäselväksi, piirtävätkö he okklusiogeometrian näytönohjaimella vai prosessorilla. Mutta olen ehdottomasti sitä mieltä, että koska mikä tahansa järkevä renderointiengine on jo GPU-bound, on hyödyllisempää lisätä CPU-kuormaa. Piirrettävän geometrian määrän vähentäessä tietysti tasapaino saattaa muuttua, jolloin joudun ehkä arvioimaan mielipidettäni uudestaan.

Päivä 16.

```
var juippi = scene.getEntity ( "Jarmo" );
var ceo = scene.getEntity ( "Tuomas" );
juippi.say ( "Tervetuloa taas sprinttiplanningiin. Aloite-
taan viime sprintin taskien demoamisella." );
juippi.walkTo ( scene.getEntity ( "Dataprojektori" ) .get-
Position() );
juippi.pickUp ( scene.getEntity ( "Kaukosäädin" );
ceo.say ( "Pistä ensin siihen virta päälle. Sieltä takaa." );
```

Alkukevennyksen jälkeen pääsemme käymään demonstraatiot läpi. Esittelen arkkitehtuurin sekä lyhyenä powerpoint-esityksenä, että käyn prototyypin koodin pääpiirteissään läpi. Lopuksi ajan massiivisen startup-shutdown-demon.

Muut esittelevät kukin alijärjestelmäprotonsa: renderöinti, UI, skriptaus, ääni...

Havaitsen, ettei Tuomas ehkä ole kovinkaan innostunut.

"Joo-o, olisin kai odottanut jotain enemmän paperinma-kuista, kun tämä sentään oli ensimmäinen sprintti. Kuvitelin, että te iskisitte minut kanveesiin täyslaidallisella UML-diagrammeja, luokkakaavioita ja sen sellaista. Mutta tepä olittekin menneet heti käytännön tasolle. Jos ymmär-rän oikein, niin tämä on nyt vähän niinkuin esimakua siitä, miltä systeemi tulee näyttämään, mutta koodi ei ollut vielä

lopulliseen versioon päätyvää?”

Jarmo nyökkää.

Kuten aiemmin jo pohdin, epäilen, että Juipin taka-ajatuksena oli ujuttaa prototyypikoodia sellaisenaan sisään. Mutta hyvä, jos tämä kataluus on nyt torpattu executive-tason huomiolla. Tietysti oma koodini on niin hyvää, että sen voi laittaa suoraan sisään, ja niin tulenkin soveltuvilta osin tekemään.

Huomatkaa: välitän vielä Vision3D:stä!

Vaikkei pitäisi.

“Selvästi te olette valinneet oikeat työkalut, kun jälkeä on syntynyt näin nopeasti. Muistan, kun Vision3D:n ensimmäistä renderöijää kirjoitettiin, siinä kesti muistaakseni kuukausi ennen kuin ensimmäinenkään polygoni pyöri ruudulla! Minä ehdotan, että kohotetaan vähän haastetasoa ja kiritään aikataulua: jos kaksi viikkoa oli nyt tämä protoilu, niin miltä tuntuisi saattaa seuraavissa sprinteissä eli vaikka 3 + 3 viikossa versio 2.0 pääpiirteissään toimivaksi? Siitä olisi vielä alkuperäiseen aikatauluun kuukausi aikaa viilailla, jos tarvetta olisi. Otettaisiin vaikka tavoitteeksi, että joululoman jälkeen EduWorld pyörii 2.0:n päällä.”

Jos Tuomas on päättänyt jotain, hänelle on aika paha sanoa ei. EduWorld on n. vuosi sitten perusasteen kouluille myyty yksinkertainen virtuaalimaailmasovellus, jossa voi liikkua ympäriinsä, chatata, lähettää kuvia maailmaan ja luoda omia tietokilpailuja. Projektin merkittävin ominaisuus ei suinkaan ole jaettu kokemus (joka ei ole kummoinen), vaan melko ninjatason sensuurifeature, joka estää kiroilun ja goatse-kuvien lähettämisen. Tämä osuus tosin ulkoistettiin.

Sinänsä benchmark on aika hyvä ja reilu: se osoittaa perussysteemin toiminnan, menemättä ihmeelliseen häröilyyn. Suorituskyvykkään se ei varsinaisesti vaadi.

Mutta huomatkkaa: *oikeat työkalut*. Qt on hyväksytty executive-tasolla. Periaatteessa nyt on kaksi vaihtoehtoa, joista kumpikaan ei ole hyvä:

Vaihtoehto 1: Qt todella osoittautuu riittävän hyväksi pohjaksi, ainakin OpenGL-renderöintiin tyytyessä. Tämä mahdollisesti tarkoittaisi, että kostoni tarpeellisuus olisi mitätöity: ketään ei kiinnostaisi maailman paras Direct3D-engine, kun meillä nyt on riittävän hyvä pohja Vision3D:ssä.

Vaihtoehto 2: Systeemi räjähtää käsiin, kun sitä aletaan oikeasti toteuttamaan. On kuitenkin liian myöhäistä kääntää suuntaa, vaan hirveitä hackeja ja purkkaviritelmiä käyttämällä se on pakko saada toimimaan Qt-pohjaisesti. Seuraa crunchia ja koko yön kestäviä buginmetsästyssesioita.

Tiukentunut aikataulu tarkoittaisi myös, etten voisi omistautua Jörmungandrille yhtä paljon kuin aikaisemmin. Tinginkö yöunista lisää? Alanko käyttää spiidiä? Vai annanko vain periksi?

Ei helvetti. Periksi ei ainakaan voi antaa. Katsotaan nyt mitä tästä tulee. Joululomalla voin kiriä Jörmungandria kiinni ainakin jonkin verran.

“Kokeillaan kahta kolmen viikon sprinttiä. Tai jälkimmäinen tulee olemaan hieman vajaa. Mutta puolessa välissä me joka tapauksessa pystymme arvioimaan, miten hyvin on edetty,” Jarmo vastaa. Havaitsen hänen äänensävyssään yhtä aikaa epävarmuutta ja lipevää pyrkimystä miellyttämiseen. Kaksinkertainen hyi.

“Ja muistakaa, voimme käyttää soveltuvin osin versiota 1.4 koodia, joka on riittävän hyvää. Tosin sitä ei kyllä ole paljon. Jos on epäselvää, varmistakaa meiltä,” Kaarlo lisää.

Suunnittelemme käytännössä kaksi sprinttiä yhdessä mammuttisesiossa. Korteista luovutaan, taskit assignoidaan suoraan, ja sen jälkeen vastuussa oleva henkilö tai henkilöt arvioivat keston itse. I approve of this.

Havaitsen arvioissa rämäpäisyyttä tyyliin: “Kyllä me tämä pystytään tekemään.” Tämä on toisaalta ihailtava asenne, mutta Vision3D:n vuoksi en repisi itseäni ääri rajoille. En, jos siinä käytetään Qt:ta ja OpenGL:ää.

Direct3D10 ja pyrkimys AAA-tason laatuun ja suorituskykyyn olisi aivan eri asia!

Itse pyrin löytämään sopivan tasapainon. Ei liikaa, mutta ei liian vähän. Toimivat perusominaisuudet kuudessa viikossa ei ole ollenkaan mahdoton tavoite, kun tiedetään, mitä tehdään, ja toteutetaan vain tarvittava minimi, mutta silti niin, ettei sorruta mihinkään iljettävään hakkerointiin. Kaikki koodi kestävän kehityksen mukaan. Tässä sprintissä perusjutut hyvin, ja ensi sprintissä mm. verkkokoodi ja animaatio. Ja meitä kuitenkin on yhdeksän. Ehkä Jarmokin alkaa koodata.

TOINEN KIRJA – VESI

TOH – Harmonia universumin kanssa

Päivä 17.

Serialization

Jörmungandr Engine supports serialization of scenes in both binary & XML format. To use, call the scene's save(), saveXML(), load() or loadXML() functions as necessary...

Hemmetti.

Ei lupaa hyvää, jos pyörittelen mielessäni Jörmungandr-in tulevia ominaisuuksia dokumentaation, ei toteutuksen tasolla. Serialisaatioon, tai scenen määrittelyyn yleensä-kään, vaikka scenenode-toteutus jo onkin, on vielä matkaa. Ensin toimiva occlusion culling. Se on melkein valmis, mutta Z-pyramidin epätarkkuus aiheuttaa rajatapauksissa inhottavia bugeja, joissa näkyvissä oleva esine jää virheel-lisesti piirtämättä.

Z-puskuri käyttää nyt kokonaislukuja liukulukujen si-jasta, sillä kokonaislukuoperaatiot ovat hivenen nopeam-pia (profiloin tämän waretetulla Intelin VTunella, mistä en ole erityisen ylpeä.)

Ehkä lisäksi vain riittävän ison bias-luvun, eli joko siir-rän occludereita taaksepäin, tai testattavia esineitä eteen-päin.

Mutta se illalla. Nyt Vision3D:tä. Taskilistani tässä sprintissä on seuraava (jos toisto ihmetyttää, niin huommat-

kaa, että nämä ovat kaikki lopulliseen versioon päätyviä, eivät enää prototyyppejä.)

- *Framework: Module & plugin load / unload / reload (*)*
- *Framework: Framework and main loop (*)*
- *Framework: Platform abstraction (*)*
- *Framework: Logging system*
- *Framework: User-specific configuration*
- *Framework: Crash dumps*
- *Scene model: Scene*
- *Scene model: Entity*
- *Scene model: Component*
- *Scene model: Component-attribute system*
- *Renderer: Octree*
- *Renderer: Frustum culling*
- *Renderer: Scene node component*
- *Renderer: Drawable component*
- *Renderer: Static mesh component*

Tähdellä merkityt ovat erityisen kriittisiä, sillä muiden tekemiä alijärjestelmiä ei voi integroida oikein, ennen kuin nuo ovat kunnossa. Edellisessä arkkitehtuuripalaverissa minulta lipsahti vähän vahingossa, että mitä jos tehtäisiin Vision3D:hen entiteetti-komponentti -pohjainen scene. Ehkä kyse on siitä, että koska se on kuitenkin minun vastuullani, en halua tehdä sontaa. Keksinkin sitten Jörmungandriin jotain lisäbonusta.

Minulla on myös renderöintiin liittyviä taskeja. Sovimme seuraavan työnjaon Sebastianin kanssa: hän likaa käteensä OpenGL:n syövereissä, kun minä toteutan ylemmän tason juttuja. Samalla tulen valvomaan, ettei hän pistä enää sisään samanlaista roskaa, kuin prototyypissä.

Olen järkeillyt seuraavasti, että en välitä Vision3D:stä, mutta olen silti ns. universal coder (vrt. universal soldier), joka haluaa varmistaa ammattiyhpeytensä säilymisen. Ja edelleen: jos keksin toteutusmallit ensin teoreettisella tasolla ja varastoin ne mieleeni Jörmungandriin toteutta-

mista varten, ne eivät varsinaisesti saastu, vaikka toteutaisinkin ne ensin Vision3D:hen.

Tätä on harmonia universumin kanssa.

When I'm one with my mind I am one with the world.

Alijärjestelmämoduulien ja pluginien lataus ja poisto on yksikkötestausta vaille valmis. Lähden syömään ennen muita, jolloin palatessani koodareiden tila on sopivan hiljainen muiden puolestaan lähdettyään lounaalle, ja testaus tulee sujuumaan ilman häiriötekijöitä. Sodexon päivän jälkiruoka on cappuccinomousse, eikä se petä: maku on voimakkaan epämiellyttävä, ja yleinen olemus kylmä ja niljakas.

Palattuani suoritan testauksen ripeässä tahdissa. Kaikki on kunnossa.

Task completed.

Pääsilmutka on toinen lähes yhtä triviaalinen tehtävä. Myöhemmin voi seurata jotain mielenkiintoista säätöä sen suhteen, missä järjestyksessä alijärjestelmiä pitää itse asiassa ajaa joka framella (esim. animaatio- ja kamerajärjestelmät luultavasti haluavat objektien lopulliset paikat fysiikkasimulaatioaskeleen jäljiltä), mutta vielä sillä ei ole väliä. Subysteemit voivat esim. yksinkertaisesti ilmoittaa prioriteettiluvun, ja ne ajetaan sitten prioriteettijärjestyksessä.

Qt:n ajastimet eivät mielestäni ole soveltuvia pehmeäreaaliaikaisen sovelluksen ajastamiseen, ja kunnollinen frame limiter jää nyt puuttumaan. Jos yritän asettaa maksiminopeudeksi esim. 100 framea sekunnissa, seuraa ajatuksen epätarkkuuden takia käytännössä inhottavaa nykimistä. Tähän voi joutua palaamaan, mutta nyt riittää, jos ohjelma pyörii käytännössä niin nopeasti, kuin pystyy (tiettyistä näytönohjaimista alkaa kuulua vinkunaa, kun ne piirtävät esim. tuhat kuvaa sekunnissa, mutta sen ei pitäisi olla haitallista.)

Task completed.

Platform-abstraktio ei tule valmistumaan tänään, joten

en edes aloita. Sensijaan eksyn ihmeellisten mutkien kautta (googlettamalla ensin Bastard Programmer From Hell, sitten Bastard Operator From Hell) seuraavalle sivulle:

<http://www.fiftythree.org/etherkiller/>

Etherkiller on siis johto, jossa on toisessa päässä lähiverkkoliitin (yleisimmin RJ45) ja toisessa päässä verkkovirtapistoke. Kun johdon yhdistää kytkimeen tai työasemaan, ja antaa sille virtaa, pitäisi tapahtua jotain mielenkiintoista ja tuhoisaa.

Illalla lisään suunnitelman mukaisesti Z-biasin Jörmungandrin occlusion cullingiin, ja tarkistan sen toimivuuden lähestymällä peittävää esinettä hyvin hitaasti. Nyt mitään artefakteja ei ilmene.

Task completed.

Seuraavaksi on vuorossa toteuttaa jumalallinen varjojenpiirtokoodi. Ikävä kyllä tämä tarkoittaa, että Vision3D tulee mitä luultavimmin etenemään scenemallin toteutukseen Jörmungandria ennen, mutta ei voi mitään. Uskon, että jos saan esim. variance shadow map:it tai exponential shadow map:it toteutettua, se aiheuttaa niin suuren hübriksen ja tärinän, ettei vähään aikaan ole väliä, mitä Vision3D:ssä tapahtuu.

Toisaalta tämä voi olla hyvin vaarallista!

Välinpitämättömyyteen ei saisi olla hetkenkään varaa.

Päätän, että Jörmungandrin tulee tukea varjoja myös läpinäkyville objekteille. Tämä ei ole ongelma, koska übershader-lähestymistapa edellyttää joka tapauksessa, että kaikki shadow mapit lasketaan ennen varsinaisen kuvan piirtämistä, ja näinollen niiden tulee säilyä näytönohjaimen muistissa. Iljettävä multipass-renderöijä puolestaan voisi piirtää yhden shadow mapin kerrallaan ja sen jälkeen laskea tämän tietyn valon vaikutuksen varjojen kanssa. Tällöin samaa varjotekstuuria voisi käyttää uudelleen, sillä rajoituksella tietysti, että koska läpinäkyvät esineet piirre-

tään viimeisenä (ja niiden piirto täytyy järjestää etäisyyden, ei valojen mukaan), ei niiden kanssa voi enää käyttää varjoja.

En suinkaan halua miettiä, mitä Vision3D 2.0 tulee tekemään varjojen suhteen. Jätän sen Sebastianin päänvai-
vaksi. Aikaisemmat versiot tukivat vain stencil-varjoja, ja tietysti täysin CPU:lla laskettuina, mikä on kyllä kamalaa. Tekniikan tarkempi referoiminen tässä olisi minulle tus-
kallista, joten katsokaa täältä, jos haluatte tietää:

http://en.wikipedia.org/wiki/Shadow_volume

Päivä 18.

Kun tulen töihin hieman tavallista myöhemmin, siellä on hirveä huuto päällä. Hannele ja Sebastian ovat lähes toistensa kurkuissa kiinni.

Tämän täytyy tarkoittaa, että vaihtoehto 2 on tapahtunut.

Qt on räjähtänyt.

Voi helvetin helvetti. Ehkä ei ollut ollenkaan hyvä, että sain moduuli- ja plugin-latauksen jo tehtyä. Koska nyt ihmiset pääsevät heti astumaan toistensa varpaille, ennen kuin ovat siihen filosofisesti valmiita. Tosin jos tämä olisi tapahtunut vasta myöhemmin, olisi tuho arvatenkin ollut vielä suurempi.

Nyt jos koskaan tarvitaan harmoniaa universumin kanssa.

Tavallaan olen tyytyväinen, koska tämä episodi tulee nyt joka tapauksessa kylvämään epäilyksen tiimiläisten mieliin, vaikka ratkeaisikin nopeasti.

Qt ei ehkä olekaan ihku.

Jörmungandr säilyy relevanttina.

Mutta se, mitä tämä välittömästi tarkoittaa, ei ole ollenkaan miellyttävää. On välttämätöntä, että osallistun buginmetsästykseen, koska muuten aikaa saattaa mennä liikaa hukkaan päämäärättömään räpeltämiseen.

Hannele rauhoittuu hieman. "Siis vaikket usko, täällä

joka tapauksessa tulee assertti, että GL-konteksti ei ole oikea. Eikö tuo ole aika selvää.”

Tavallaan on ihailtavaa, miten Hannele on noussut juniorikoodaajasta UI-alijärjestelmän vihaiseksi toteuttajaksi, joka joutuu kriegiin varsinaisten renderöintisysteemikoodarien kanssa.

“No saatana, uskotaan sitten.” Sebastian lausuu kirosanan lyödyn kuuloisesti.

Uskon ymmärtäväni, mitä tapahtuu. Tähän asti testejä oli ajettu siten, että ruudulla on ollut yleensä vain yksi UI-widgetti 3D-kanvaasin päällä. Mutta nyt, kun widgettejä on useampi, saattavat ne varata oman OpenGL-kontekstin, ja tällöin tulee varmistua, että oikea konteksti on valittuna aina, kun tehdään jotain, joka saattaa mahdollisesti aiheuttaa GL-operaatioiden suorittamista.

Oikeastaan tämä tulee olemaan helposti korjattavissa. Renderöintikoodimme on sentään täysin omaa, joten tiedämme aina, milloin tulemme kutsumaan grafiikkarajapinnan funktioita. Täytyy vain olla huolellinen, ja lisätä konteksin tarkistus ja vaihto ennen kaikkia näitä operaatioita, ja palautus operaation jälkeen.

Eleganttia eikä tehokasta se ei missään nimessä ole, ja tulee jatkossa edellyttämään jatkuvaa huolellisuutta. Sekoamisen ja UDB:n Damokleen miekka tulee jatkuvasti roikkumaan päidemme yläpuolella.

Mutta paljon pahempi olisi, jos käyttäisimme kolmannen osapuolen renderöintimoottoria. Silloin joutuisimme käytännössä käymään sen kaiken koodin läpi, ja katsomaan, milloin kontekstitarkistus on tarpeellinen. Ellemme sitten lisäisi tarkistusta ennen jokaikistä kutsua, kuten vaikka `Camera::setPosition()`, mutta siinä taas ei olisi mitään järkeä, sillä kameraobjektilla on tyypillisesti vain sisäinen tietorakenne, joka ei liity renderöintiin mitenkään.

Iltapäivällä ongelma on pääpiirteissään ratkaistu. Asertteja ei enää tule, ja suuri määrä widgettejä suostuu pyörimään ruudulla. Ei kovin tehokkaasti, mutta hyväksyttävästi kuitenkin. (Tosin neliytimisellä prosessorilla, AMD

Radeon HD4850-näytönohjaimella, ja release-buildilla.)

Pidän koko tiimille esitelmän siitä, mitä GL-kontekstit merkitsevät heidän hyvinvointinsa kannalta.

Mutta Hannele ei ole vielä tyytyväinen. Ilmenee, että QGL-pohjaiset widgetit eivät toimi täysin niinkuin käyttöjärjestelmän normaaliin UI-piirtoon perustuvat vastineensa. Arvaan, että tämä fakta tulee pitämään hänet varpailaan koko loppuprojektin ajan.

“Jos rendattaisiin vain koko paska tekstuuriin,” ehdottaa Jyrki, joka on pääasiassa aina koodannut objekti- ja applikaatiologiikkaa, mutta omaa ilmeisesti halua matalamman tason ohjelmointiin. Itseasiassa on huomattavaa, että hän yleensä tietää Qt-widgeteissä olevan ominaisuuden, että ne osaavat piirtää itsensä kuvapuskuriin.

“Silloin joutuisimme suppressoimaan Qt:n normaalin piirtologiikan,” Kaarlo vastaa ennen kuin ehdin itse reagoida.

Ja siitä ei hyvä seuraisi. Voi pojat, siitä ei hyvä seuraisi! Ainoana hyvänä puolena Direct3D:n käyttö olisi silloin periaatteessa mahdollista.

“Performanssi lisäksi tippuisi,” Kaarlo jatkaa.

No niin, tästä se lähtee. Epäilyksen siemenet on kylvetty. Ja muistakaa, että todellinen fullscreen-tila on silti vielä saavuttamattomissa. Tuomas on tästä vielä autuaan tietämätön, mutta kunhan asia selviää hänelle, hän saattaa nousta arvaamattomaan raivoon.

Jörmungandrille se ei ole ongelma.

Jörmungandr on ihku.

Tulen töistä agitaation vallassa. Vaikka ne tietävätkin omalle enginelleni hyvää, Qt:n ongelmat pyrkivät häiritsemään harmoniaani, ja harjoittelen jonkin aikaa kuji-kiri-käsimerkkejä. RIN! KYO! TOH!

Tämän jälkeen kaadan itselleni viskiryyppyn jäillä, ja alan koodata variance shadow map-toteutusta. Koska minun täytyy tallettaa kaksi eri arvoa joka pikseliä kohti, hardware shadow map:ia (eli näytönohjaimen omaa Z-

puskuria) ei voi hyödyntää. Tämä on valitettavaa, mutta toisaalta takaa suuremman toimintavarmuuden eri valmistajien näytönohjaimilla.

Oikeastaan, mitä minä sekoilen? Käytännössä valmistajiksi voidaan laskea vain NVIDIA ja AMD, ja molempien nykyiset näytönohjaimet tukevat D3DFMT_D24X8- ja D3DFMT_D16 -tekstuuriformaatteja hardware shadow map-ominaisuutta varten. Intelinkin uusimmat GPU:t saattavat niitä tukea, mutten laskisi mitään sen varaan (ja suorituskyky on niillä hirveä joka tapauksessa.)

Perinteiset shadow mapit voi koodata fallback-ratkaisuna, jos on tarvetta ja innostusta. Luultavasti ei ole. Jörnungandria ei ole tarkoitettu heikoille näytönohjaimille!

Päivä 19.

Herään huonosti nukutun yön jälkeen. Variance shadow mapit ovat osoittautuneet pikamatkaksi Helvettiin.

Tavallisen shadow map-toteutuksen kanssa painitaan yleensä ns. varjoakne- (virheellinen pintojen itsevarjos-tus) ja peter panning- (varjot ovat irti objekteista) ongelmien kanssa. Variance shadow map:in kanssa näitä ei ole, mutta sen sijaan tilalle tulee pirullinen vuoto-ongelma, kun valon tiellä on eri etäisyyksillä olevia varjoja tuottavia esineitä. Yritin häivyttää artefakteja erinäisten taikavakioiden avulla, aina klo 3 aamulla asti, kunnes jouduin myöntämään tappioni.

Olen lähes valmis hylkäämään VSM-tekniikan.

En usko, että tästä tulee hyvä päivä. Ei mitenkään jaksaisi koodata Vision3D:tä koko päivää. Lisäksi on liukasta, pimeää ja kylmää, ja lähes kaadun pyörällä. Perkele!

Eilisen sekoilun vuoksi minulla on vielä taskeja kesken framework:iin liittyen. Pakotan itseni vääntämään koodia (ensin yritän saada Jensiä töihin, mutta havaitsen, että sivupersona kykenee vain mielistelyyn, ei varsinaiseen luovaan toimintaan.)

Onneksi saan olla rauhassa.

Kirjoittamistani funktioista ei voi olla kyllä millään tavalla tyytyväinen. Näin kömpelöä koodia en muista laati-

neeni pitkään aikaan. Tavallaan se sopisi Vision3D:n henkeen, muttei täytä universal coder:in laatuvaatimuksia.

Käykö nyt niin, että minun täytyy kirjoittaa platform-abstraktio ja virheloggaus uudestaan ylpeyteni vaatimalla tasolla? Katsotaan. Mutta ei nyt. Kaksi saraketta sprint backlogista värjäytyvät virheäksi.

Sitten pysähdyn miettimään. Annanko oikeasti periksi ja toteutan Jörmungandriin tavanomaisen varjotekniikan? Vai toteutanko siihen yleensä enää yhtään mitään? Juuri tällä hetkellä koko engine tuntuu lähes yhtä houkuttelevalta kuin Vision3D itse. Pelottavaa.

Ehkä todellisen harmonian universumin kanssa saavuttaa vasta kun hyväksyy, että kaikki koodaus on turhaa, ja ryhtyy vaikka hoitamaan kalliopuutarhaa.

Focus! You're a killer!

Syötyäni olotilani paranee jonkin verran, ja pyssytän nopeasti läpi crash dump-tehtävän. Ilkeästi teen siitä ainakin ensiksi vain Windows-toteutuksen:

```
#if defined (MSC_VER) && defined (WRITE_DUMPS)  
    BOOL success = MiniDumpWriteDump(  
        GetCurrentProcess(), GetCurrentProcessId(), file,  
        MiniDumpWithDataSegs, &exceptionInfo, 0, 0  
    );  
#endif
```

Vauhtiin päästyäni toteutan myös käyttäjäspezifisen konfiguraation. Siinä ei sentään voi huijata: unix-toteutus täytyy tehdä oikeasti. Unixilla sen ei itse asiassa tarvitse olla muuta kuin yksinkertainen hakemistonluonti käyttäjän kotihakemiston alaisuuteen, kun taas Windowsilla tehdään melkein yhtä helppo API-kutsu hakemiston nimen saamiseksi (luominen täytyy hoitaa itse):

```
char pathName [MAX_PATH];  
SHGetSpecialFolderPath(0, pathName, CSIDL_PERSONAL,
```

NAL, 0);

Framework-taskini näyttävät nyt kokonaan vihreää. Tätä voi vaikka juhlia pitämällä tauko (heikot pelaisivat esim. Guitar Heroa, mutta siihen en alennu), ja pohtimalla hieman: scene-toteutukseen ei voi mennä ihan suinpäin, ei edes Vision3D:ssä, eikä varsinkaan Jörmungandr:issa, koska se tulee vaikuttamaan niin paljon siihen, miten systeemillä itse asiassa tehdään applikaatiologiikkaa.

Saan välähdyksenomaisen näyn todella kimurantista, ehkä jopa vaarallisen virheatttiista serialisointikoodista.

Jokainen komponentti on vastuussa itsensä serialisoinnista.

Tämä itseasiassa liittyy siihen, mitä pyörittelin mielesäni aiemmin dokumentaation tasolla. Ja tämä on ehdottomasti se systeemi, joka tulee Jörmungandr:iin. Scenestä voidaan rullata läpi kaikki entityt, ja joka entity puolestaan rullaa läpi komponentit, ja kutsuu niiden lataus- ja tallennusfunktioita. Tämä funktio saa sitten olla niin jumalallisen monimutkainen kuin on tarpeen, kunhan jokainen tallennettu bytestreami on valiidi, ja myöhemmin tulkittavissa yksikäsitteisesti.

Mutta Vision3D kaipaisi jotain ystävällisempää, ehkä. Suorituskyvystä ei niin väliä, jos systeemi on helposti laajennettava. Sittenpä hörhöt saisivat lisätä omia komponentteja helposti, tai lisätä olemassaoleviin lisää dataa.

Päätän (tai uskon, että Jens päättää puolestani) että Vision3D:n komponentteihin tulee yksinkertainen QVariant-Map, johon komponentin sisältämä data (eli attribuutit) tallennetaan. QVariant on Qt:n oma tapa tallentaa vaihtelevia tietotyyppisiä. Kun käytän sitä, minun tarvitsee kirjoittaa minimaalinen määrä omaa koodia, ja pysyn toisten silmissä Qt-tosiuskovana. Lisäksi Qt:n signal-slot -mekanismi avulla on kenen tahansa helpohko kytkeytyä attribuuttien muutoksiin.

Ongelma ratkaistu. Mutta alkaa olla myöhä. Kirjoitan yksinkertaiset rungot scene-, entity- ja component-luokil-

le, mutten vielä pistä niitä versionhallintaan. Tarina jatkuu huomenna.

Ei kun nythän on viikonloppu tulossa. Kyllä sitä voikin olla sekaisin.

Juuri ennen koneeni sammuttamista teen virheen, joka laskee mielialaani huomattavasti. Kelaan sprint backlog-dokumenttia alaspäin, nähdäkseni kuinka hyvin muut etenevät.

Näky on masentava. Ainoa valmis taski on Kaarlon "Scripting system: basic integration", joka käytännössä tarkoittaa sitä, että kirjoitetaan luokka, joka instantioi QtScriptEnginen. Sitten ehkä lisätään siihen funktio, joka lataa skriptitiedoston ja ajaa sen (onnistuu helposti) ja se on siinä.

Kaikki muu on kesken – tai itse asiassa suurin osa on vielä aloittamatta.

Eihän minun tarvitsisi välittää. Aivan sama, miten koodaatte. Mutta jotain pahaa tämä enteilee. Onko ihmisillä päällä jonkinlainen halvaantuneisuus liian hyvän koodin kirjoittamisesta? Helvetti, tämä on Vision3D, eikä Marslennon kontrolliohjelma. En silti viitsi avautua asiasta. Jarmon tehtävähän on tarkkailla työn etenemistä, ei minun.

Silti se jotenkin keljuttaa, jos työtoverit koodaavat hitaasti tai huonosti. Mikseivät he voi olla kuten minä? Äärimmilleen vietynä ajatus on huvittava: siinä olisikin miettimistä, jos jokaisella olisi oma salainen engine työn alla.

Mutta nyt ulos.

Hyi, pakkanen on kiristynyt.

Kotimatkan aikana pohdin Jörmungandrin varjo-ongelmaa. Kaikki käytettävissäni olevat vaihtoehdot tuntuvat huonoilta. Exponential shadow map-menetelmää voisi kokeilla, mutta se on aika huonosti dokumentoitu, enkä tiedä, haluanko käyttää huomattavasti aikaa epämääräisiin kokeiluihin. On jokseenkin ironista, että minulla on näin voimakas paralyysi päällä oman engineni suhteen, kun taas Vision3D eteni tänään tehokkaasti.

Kaikki menetelmät varjojen tuottamiseksi ovat huonoja!

Ehkä tämä on 3D-grafiikan nykyisen paradigman julma fakta. Sitten sovelluksesta riippuen valitaan vähiten huono toteutustapa.

Jokseenkin masentavaa. Miksi en ole tätä aikaisemmin tajunnut?

Koska ajatus koneen ääreen istumisesta juuri nyt tuntuu vastenmieliseltä, päätän tapojeni vastaisesti lähteä baariin. Mielenkiintoisesti nimetty Club Rasputin ei ole kovin kaukana, eikä muutenkaan epämiellyttävimmästä päästä, joten se sopii aivojeni resetointiin riittävän hyvin.

Tilaan ison tuopin ja jään tiskille sitä nautitsemaan. Syrjäsilmillä havaitsen jotain asiaankuulumatonta.

Valtteri.

Yhteensattuma ei ole kovin tervetullut. Ei kai se nyt aio tulla juttelemaan? Valtterin lähtö on vieläkin traumaattinen, jos sitä muistelee syvemmin, ja on muutenkin harvinaista, että harrastaisin minkäänlaista työpaikan ulkopuolista sosiaalista kanssakäymistä entisten tai nykyisten työtovereideni kanssa.

Toisaalta tämä olisi ehkä mahdollisuus saada selvyyttä johonkin.

Valtteri selvästi havaitsee minut.

Katsotaanpa, mitä tapahtuu.

Valtteri istuu myös tiskille, noin puolentoista metrin päähän, ja tilaa tequila-shotin.

“Mitäs Vision3D:lle kuuluu?” hän kysyy.

“Se on need-to-know -tietoa, ja sulla ei enää ole need-to-know,” vastaan teeskenneltyä hyväntuulisuutta tavoitellen. Hyväntuulisuus on persoonalleni täysin vääränlainen olotila, ja tuntuu auttamatta väkinäiseltä. Tästä viisastuneena teen olennaisen tärkeän päätöksen: ei ikinä enää minkäänlaista teeskentelyä! “No siis, aika hyytynyttä.”

“Hmm.”

Valtteri saa paukun, ja kumoo sen aikailematta kurk-

kuunsa.

“Mistä siinä oikein oli kyse, kun häivyyit?” kysyn. Nyt ollaan perimmäisten kysymysten äärellä. Nyt ehkä selviää, miten ja miksi voittamaton Pääohjelmoija sai tarpeekseen.

“Tarviiko siinä olla mitään grand masterplania? Sain hyvän tarjouksen mm. iPhone-pelejä devaavasta pikkufirmasta.”

iPhone. En ole kateellinen. Huokaisen sisäisesti helpotuksesta. Perimmäinen ajatus, joka minulla tulee mieleen, on kunnianhimon puute. Minulla on ehkä vinksautanut käsitys siitä, että työn pitää jatkuvasti olla eeppeisiä haasteita vastaan kamppailemista, ja jotenkin olin ymmärtänyt, että Valtterin näkemys olisi samanlainen. Tosin onhan mobiilipelienkin kehityksessä epäilemättä omat haasteensa: pienet budjetit, tiukat aikataulut... Ja jos iPhone-peliä kehitäessä on hauskaa, eikö se nyt oikeastaan voita hörhöjen 3D-sovellusmoottorin kehittämisen?

“Jos totta puhutaan, laskin sen varaan, että idea Vision3D:n räjäyttämisestä kunnolla ei menisi läpi. En minä kuvitellut, että siihen uskallettaisiin lähteä.”

“Ja sitten kun se menikin läpi, sua alkoi jänistää?”

En kyllä usko, että Valtteri myöntäisi, mutta toisaalta en myöskään usko, että siitä oli kyse.

“Äläpä nyt kuvittele mitä tahansa. Tavallaan olin niin kauan kironnut, miten paska Vision3D on, että sitten, kun oikeasti tuli tilaisuus korjata se, niin minulta olikin raivo loppunut kesken. Kai minä olisin väkisin vääntänyt sitä, jos olisi ollut pakko, mutta se tarjous oikeasti tuli aika hyvään paikkaan.”

Valtteri tuijottaa jonnekin kaukaisuuteen.

“Niin, ja Jarmo ei ollut minun ideani.”

Nauruumme sekoittuu pieni annos ilkeyttä.

Juon pitkän kulauksen olutta, ja mietin, pitäisikö minun kertoa Jörmungandr:ista. Se ei ehkä olisi viisasta. Mutta toisaalta, tämän rehelliseltä vaikuttavan tilityksen jälkeen en usko, että Valtteri pettäisi minut kertomalla siitä millekään asiaankuulumattomalle taholle. Mitään petosta ei

alunperinkään ollut minua kohtaan: me molemmat olemme lopulta palkkasotureita (universal coders), jotka tarjoavat kuolettavia taitojaan parhaiten tarjoaville.

“Niin no, sinun alaisuudessa Vision3D:n uudistus olisi saattanut pysyä aika huomattavastikin paremmassa suunnassa. Mutta en minä siitä ole katkera, vaan aloin koodata jotain parempaa. Mutta et sitten kerro tästä eteenpäin.”

Valtteri ilme näyttää hieman tyrmistyneeltä: hän muistaa taatusti varsin hyvin työsopimuksemme pykälän, jonka mukaan kaikki sen voimassa ollessa tuotettu koodi on työnantajan omaisuutta. Onkohan se muuten edes lailinen? Luultavasti ei.

Tilaan toisen tuopin, alan selittää Jörmungandr'in hienouksia, ja Valtteri elehdinnästä huomaa, että häneltä menee paljon yli. Such are the ways of lead programmers. He joutuvat keskittymään liikaa kokonaiskuvaan, jolloin toteutuksen jumalalliset yksityiskohdat jäävät sivuun.

“Mitä sitten aiot tehdä sen kanssa?” hän kysyy lopetettuani viimein.

“Katsotaan. Aika näyttää.”

“Jotenkin tuosta tuli mieleen, kun monta vuotta sitten koodasi kuusnepa-juttuja, samalla kun teki kesätöissä joutain ankeaa tietokantasoftaa. Kun laski jokaisen kellojakson, ja oli mahtavaa, kun sai ruudun skrollaamaan kahdeksaan suuntaan, ja siihen päälle vielä yli kahdeksan spriteä. Ja Rob Hubbardilta ripatun musiikin soimaan taustalla.”

Vertaus on hieman erikoislaatuinen. Vision3D ja Jörmungandr kuitenkin pyrkivät samaan päämäärään: immerssiiviseen 3D-maailmaan. Valtteri luultavasti hakee samankaltaisuutta filosofisella tasolla, mutta alkava humalatilani estää sen syvällisemmän ymmärtämisen. Paitsi ehkä siten, että koodisoturi tarvitsee huvin ja velvollisuuden tasapainon.

Ei. Huvi ei ole oikea sana. Jörmungandr'in rakenne ja optimointi on kuolemanvakava asia, kuten epäilemättä kuusnelosen peli- tai demojuttujen hiominen aikanaan. Demoskenessä tehtiin tulikivenkatkuisia fuckings-demoja

ja kirjoitettiin halventavia skrollitekstejä, jos joku vaikka varasti jonkin koodinpätkän, tai oli muuten auttamattoman lame.

Oluen vaikutuksen voimistuessa tiedän nyt, mitä on tehtävä varjojen suhteen: standard shadow mapping, mutta paremmin kuin kukaan muu on ikinä tehnyt. Ja hardware-depthpuskuria käyttäen.

Kyllä.

Näin on päätetty.

Valtteri tilaa kolmatta paukkua, mutta minä katson viisaaksi rajoittaa tähän.

“Nähdään,” sanon ja poistun ulkoilmaan.

Päivä 20.

```
float pcf = 0.25;
float4 depthValues = float4 (
    tex2Dproj (shadowMap, uv + float4 (offset.x, 0, 0, 0)),
    tex2Dproj (shadowMap, uv + float4 (-offset.x, 0, 0, 0)),
    tex2Dproj (shadowMap, uv + float4 (0, offset.y, 0, 0)),
    tex2Dproj (shadowMap, uv + float4 (0, -offset.y, 0, 0))
);
float inLight = dot (depthValues, pcf);
```

Aivan peruskamaa, mutta toimii. Nyt hieman harmittaa se yö, kun väänsin VSM-toteutusta. Tai olisihan se ollut hyvä, jos olisi toiminut luotettavasti. Mutta nyt, kun käytetään hardware shadow map:ia, voidaan colorwritet kytkeä pois, ja tämän seurauksena varjojen piirtonopeus kaksinkertaistuu.

Menetelmä on siis seuraava: otetaan neljä näytettä shadow mapin vierekkäisistä kuvapisteistä, joista jokaisen näytönohjain on jo filteröinyt bilineaarisesti 2 x 2 näytteistä. Toisin sanoen saavutetaan sama varjojen filteröinti, kuin jos tehtäisiin shaderissa 3 x 3 PCF, mutta paljon nopeammin.

Huomattavan paljon nopeammin!

Mutta varjojen filteröinti on vain yksi osa koko varjototeutusta! Vähintään yhtä tärkeitä ovat varjokameran pro-

jektioparametrit, depth biasointi, parallel split shadow mapping ja splittien oikea asettelu, näkyvän scenen sovitaminen varjotekstuuriin optimaalisesti, pistevalon jakaminen kuudeksi spottivaloksi varjojen laskemista varten...

Saatana.

Muistin juuri kammottavan iljettävän heikkouden hardware shadow map-menetelmässä. Alitajuntani olikin muistuttanut jostain epämiellyttävästä, mutta olin vain jättänyt sen huomiotta.

Kun käyttää näytönohjaimen Z-puskuria, ei voi renderöidä cube-tekstuuriin. Pistevalolle pitää nimittäin laskea kuusi eri shadow map:ia, joita varten cubemap on aivan loistava ratkaisu, koska sitä käyttäessä valaistut esineet voi piirtää kerralla, kun kaikki kuusi sivua on laskettu.

Olin kai jotenkin kierosti ajatellut tekeväni deferrattua renderöintiä, jossa ongelma ei ole niin paha. Mutta kun cubemappia ei ole käytössä, joudutaan pistevalon valaisemat esineet pahimmillaan piirtämään kuuteen kertaan!

Tämä on todella vastenmielistä.

Voisin tietysti koodata täysin eri shaderit varjostettua pistevaloa varten, mutta se sotii Jörmungandr:in designin puhtautta vastaan.

No, profiloidaan (ja ennenkaikkea toteutetaan) ominaisuus ensin, ennen kuin vaivutaan täyteen epätoivoon. Monesti engineiden dokumentaatiossa näkee mainintaa, ettei varjoja kannata ikinä käyttää pistevalon kanssa, kun se on niin hidasta (ja monet engineet eivät tue sitä ollenkaan), mutta olisin toivonut Jörmungandr:in olevan tämän rajoituksen yläpuolella!

Päätän hioa directional- ja spottivalojen varjot ensin huippuunsa.

Päivä 21.

Hidasta se on, ei voi mitään.

Jonkinlaista laihaa lohtua saa siitä, että jos yksi pistevalon sivuista ei sisällä varjoja heittäviä esineitä, ei kyseistä shadowmap:ia tarvitse piirtää, ja jos sivu ei sisällä valaistavia esineitä ollenkaan, se voidaan sivuuttaa täysin.

Hetkinen.

Nyt unohdan jotain.

Direct3D10 ja jumalallinen dynaaminen indeksöinti.

Kiroan typeryyttäni, ja liiaksi Direct3D9:ään jumiutunut ajattelutapaani. Dynaamisen indeksöinnin ansiosta voidaan siis yksinkertaisesti laskea pixel shaderissa, mikä pistevalon sivuista valaisee objektin kutakin pistettä, ja mitä shadowmappia tulee tällöin lukea.

Maailma pelastuu.

Enkelikuoro laulaa.

FPS kohoaa huomattavasti.

Huh, meinasinpa jättää kamalaa koodia Jörmungandr:iin. Mutta nyt huomaan, että näinhän se pitää tehdä. Kirjoitan myös parallel split shadow mappingin uusiksi käyttämään samaa periaatetta. Nytkin FPS kohoaa.

This is true power.

Päivä 22.

Uusi viikko, ja Vision3D:n skenemallin toteutus on lopulta käsillä. Mutta ennen kuin pääsen siihen asti, pidämme jälleen yhden puuduttavan arkkitehtuuripalaverin.

Joo joo, kyllä minä tiedän nämä jutut! Tiedetään, että skenestä saatavien notifikaatioiden pitää olla monipuoliset. Ja että niin skriptistä kuin C++:sta pitää päästä siihen yhtä hyvin käsiksi. En kykene nyt manaamaan Jensiä esiin, vaan säilyyn kyräilevänä itsenäni. Antakaa minun nyt vaan koodata, helvetti soikoon!

Viimein.

Tästä se nyt lähtee.

Scene-luokan toteuttaminen on hyvällä mallilla (ainakin parin ensimmäisen rivin osalta), kun sähkötkatkeavat, ja pimeys laskeutuu koodarien, kuten myös artistien ja teknisten artistien ylle.

Vaikka kiroiluttaakin, tavallaan nautin tästä. Analysoin toisten käytöstä ja nurinaa, kun he vastaavalla tavalla menettävät juuri työn alla olleen koodinsa.

Nyt läppäri olisi aika kuninkaallinen vehje. Jarmolla selainen onkin, mutta hän arvatenkin säätää vain jotain turhaa presentaatiota.

Kuluu viisi minuuttia.

Mietin, pitäisikö minun oikeasti alkaa koodaamaan paperille. Vision3D:tä? Et ole tosissasi.

Kymmenen minuuttia ilman sähköä.

“Kaivinkone on katkaissut maakaapelin,” ilmoittaa Tuomas inner sanctuminsa ovelta.

Jaha. Tässä voi siis mennä pitempäänkin. Lähdemme yhtenä laumana syömään. Pyrin silti jättäytymään porukasta jälkeen, sillä oloni on antisosiaalinen, ja juuri tällä hetkellä jokainen puheenaihe on ennalta-arvattava, väsynyt ja typerä.

Sodexossa on sentään valot päällä: se saa sähkönsä eri kautta. Jälkiruoka, ananasmousse, ei luultavasti tule aikaansaamaan mitään erikoisia väristyksiä.

Olen väärässä. Maku on erityisen metallinen ja teollinen, ja seassa on hyvin, hyvin haastavia sattumia. Tätä analysoidessa sähkökatko unohtuu. Olisikohan jopa, että olisivat vaihtaneet jälkiruo'ista vastuussa olevan kokin. Tämä kyllä sijoittuu omaan kategoriaansa, legendaarisuusasteikolla heti pullavanukkaan ja appelsiiniriisin jälkeen.

Palattuamme odotamme vielä tunnin, ja sitten sähköt palautuvat. Tällä välin bestiaalinen raivo scenemallin koodaamiseen on haihtunut.

Väksin pakottamalla jotain kuitenkin syntyy.

Joo, kaikista slottifunktioita vaan, niin tulee skriptattavuutta. Ja exposoidaan metatyypeiksi. Shared pointterit tulevat tuottamaan ongelmia, joten ilmeisesti skriptejä varten täytyy tehdä raakapointtereita palauttavat versiot. Rumaa, mutta toimii.

Eihän shared pointtereita olisi pakko käyttää. Ne vain helpottavat muistinhallintaa melkoisesti, kun olioiden oikea tuhoutumisjärjestys määräytyy implisiittisesti. Tietysti ne antavat myös mahdollisuuden pitää objekteja hengissä pitempään, kuin on tarkoitus, tai aiheuttaa circular reference laittamalla kahden eri olion kummankin viittaamaan shared pointtereilla toisiinsa. Tällöin kumpaakaan ei ikinä vapauteta. Mutta kuten totesin jo aikaisemmin, tehokas middleware kuitenkin antaa aina jonkin tehokkaan tavan ampua itseään jalkaan!

Paikan ja enginen vaihto. Vision3D:n skenen kanssa työkentelin ylhäältä alaspäin, Jörmungandr:in kanssa taas päinvastoin: perus-scenenode on jo olemassa, jonka yleis-tän nyt komponentiksi, ja sitten entity-luokan kimppuun.

Muhahaha. Tämähän on hyvin nopeaa. Tosin nyt tiedän, että kunhan pääsen sceneen asti, ongelmia on tiedossa. Syklinen riippuvuus. Käärme joka syö omaa häntäänsä. Haluaisin nimittäin, että komponentilla on tietoisuus entitystä, jossa se sijaitsee, ja entityllä edelleen tietoisuus scenestä, jossa se on. Näitä ei voi esitellä pelkästään forward-declaraatioina, koska joskus esim. komponentti joutuu kyselemään scenestä toisia entityjä ja komponentteja. Scene itse taas sisältää riippuvuuksia vähän kaikkeen (tai siis tulee jatkossa sisältämään.)

Muutenhan tämä ei olisi ongelma, mutta tarkoitus on, että komponentin kantaluokka ja entity ovat toteutettu eri staattisessa kirjastossa, kuin itse scene.

Kieltäydyn ratkaisemasta ongelmaa minkäänlaisen interfacen tai abstraktin kantaluokan avulla. Ne ovat heikoille! Mutta tästä seuraa, että nukkumaan mennessäni olen hirveän agitaation vallassa keskeneräisen ajatteluprosessin ja ratkaisemattoman ongelman johdosta. Tiedän, että on turha kuvitella saavansa unta vähään aikaan.

Yritän rentouttaa itseäni samoin, kuin ruumiista irtautumista yrittäessä, mutta raivo pitää minua liian voimakkaasti vallassaan. Rentoutuminenkin on heikkoja varten!

Perkele. Haluaisin ratkaista kaikki ongelmat tässä ja nyt.

Päivä 23.

Näen sekavan unen, jossa olen töissä Neuvostoliiton aikaisessa teräskombinaatissa. Hiestä märkänä taon punahehkuista metallia järkyttävän kokoisella moukarilla, niin isolta, etten taatusti saisi sitä oikeasti hievahtamaankaan.

Unen symboliikka on mielikuvituksetonta: metallinkappale on tietysti koodi, jota työstän.

Tai ehkä se onkin mieleni, jota yritän väkisin pakottaa rauhoittumaan ja järkevöitymään. Vielä herättyänikin raivo on hirveä. En haluaisi mennä töihin: Jörmungandr:in scene-ongelma pitäisi ratkaista ensin.

Pahinta on ehkä, että tiedän sen olevan esim. verkkokoodiin verrattuna kohtalaisen triviaalinen ongelma. Trendi on huolestuttava: vähän aikaa sitten VSM-varjot saivat minut myös vastaavan epätoivon partaalle.

Heikkous ja periksiantto.

Kirjoitan äkkiä scenen abstraktin kantaluokan, jonka toteutuksen voi sijoittaa samaan kirjastoon kuin entity ja component. Se saa ruman nimen BaseScene.

Hyi.

Mutta ongelma on ratkaistu. Töihin.

Vision3D:n scenetoteutuksen vääntäminen jatkuu yllätyksettömästi. Vilkaisen taas vaihteeksi backlogia, jottei mielialani vahingossakaan kohoaisi liikaa, ja siellä on vähän enemmän vihreää. Mutta ei paljon.

Sebastianin renderöintitaskit ovat suurin osa checkou-tattuina, eli kesken. Ei voi tietää, milloin ne valmistuvat, mutta periaatteessa minun kannattaisi siis kiirehtiä scenemallin kanssa, jotta pääsisin tekemään korkean tason renderöintitoiminnallisuutta.

Ja ennenkaikkea reviewaamaan Sebastianin matalan tason koodia. Tästä en ole ollenkaan innoissani.

Etenen suunnitelman mukaan ylhäältä alaspäin. Komponenttiattribuuttien toteuttaminen on tämän päivän ohjelmassa. Sitten koko homma olisikin paketissa, ellei sitä tarvitsisi testata.

Jarmo edellyttää ehdottomasti yksikkötestien kirjoittamista. Tätä ei tosiohjelmoijien pitäisi sietää, mutta olen riittävässä määrin luopunut tosiohjelmoijan ominaisuudesta, joten minun ei tarvitse tehdä asiasta absoluuttista ongelmaa eikä syytä väöntää kättä loputtomiin.

Silti se sapettaa!

Kirjoitetaan sitten yksikkötestejä.

Oikeastihan testit pitäisi kirjoittaa ensin. Mutta sehän on vielä agilempaa, kun vähän taivuttaa agilen sääntöjä!

Testit eivät synny valtavalla nopeudella, sillä mietin jo Jörmungandr:in scenetoteutusta, ja mitä siinä puolestaan on seuraavaksi vuorossa. Vaarallisen kimuranteja serialisointi- ja deserialisointifunktioita on luvassa.

Lopulta testit ovat valmiit, ja arvattavasti ne myös menevät läpi. Ensimmäisellä yrittämällä!

Scenetaskini värjäytyvät kaikki vihreiksi. Ok, ne eivät ehkä vastaa kaikkia kuviteltavissa olevia usecaseja, ja minulta saatetaan tulla vaatimaan selitystä joihinkin yksityiskohtiin, mutta voin vastata kategorisesti: toiminnallisuutta tullaan laajentamaan tarvittaessa.

Toimisto alkaa jo tyhentyä. Olen ollut poikkeuksellisen myöhään töissä: yleensä lähdän hieman aikaisin, koska lasken koodini eliittisyyden korvaavan puuttuvan ajan. Sebastian on jo lähtenyt: en suinkaan aio tänään katsoa versionhallinnasta, onko hän commitoinut low-level renderöintitoiminnallisuutta, ja minkä tasoista koodi on.

On korkea aika minunkin häipyä.

Onneksi me käytämme subversion-versionhallintaa, mikä mahdollistaa sen, että ihmiset voivat työskennellä eri puolilla codebasea välittämättä toisistaan. Tiettyjen hörhöjen parempana pitämät hajautetut versionhallintaohjelmat, kuten git, vaativat näennäisestä edistyneisyydestään huolimatta, että mergataan jatkuvasti. Tuho ja hävitys!

Ulkona on hirvittävä lumimyräkkä. Voisi lähes jäädä toimistolle koodaamaan Jörmungandr:ia muistitikulle.

Mutta no, matka ei ole pitkä. Jos tästä ei selviä tai se tuntuu ylitsepäsemättömän vastenmieliseltä, se tarkoittaa, että olen liian heikko koodatakseni seuraavan sukupuolven AAA-engineä, kuten Jörmungandria!

Jokaiseen liikuntasuoritteeseen täytyy suhtautua kuten taisteluun.

Kuorin inhottavan märät vaatteet (poislukien bokserit) päältäni. Havaitsen, että olen ehkä tulossa vanhaksi: ihoni ei ole enää yhtä kimmoisaa, kuin ennen. Tämä on epämiellyttävä huomio.

Sitten huomioin jotain paljon hälyttävämpää.

Muistitikku! Missä se on? Koodasin serialisointiapufunktioita, jotka sinänsä kirjoittaisi helposti uudestaan, mutta jos muistitikku on kadonnut, se tietää jotain paljon pahempaa. Jörmungandr on ehkä vuotanut.

Tikku löytyy onneksi. Se oli vain ollut väärässä taskussa.

Noniin: suihku, kuivat vaatteet päälle, ja kirjoittamaan abstraktin BaseScenen toteutus loppuun, ja sitten pisteenä I:n päälle kokeeksi perusscenenoden serialisointi ja deserialisointi.

Sehän toimii.

Koodi on selkeää, mutta silti vaarallista. (Ja monimutkaisempia funktioita on tulossa, kuten esim. animoidun scenenoden tallennus.) On aina olennaisen tärkeää kutsua pohjaluokkien lataus- ja tallennusfunktioita, tai muuten binääristreamin epäsynkronoituminen ja UDB ei ole kau-

kana!

Kaikesta huolimatta päivän päätös on kohtuullisen miellyttävä. Tästä jatketaan.

Outojen ja / tai epämiellyttävien unien sarja jatkuu: olen toteuttamassa verkkokoodin client side predictionia. Hahmo hyppii edestakaisin täysin päinvastoin, kuin sen pitäisi käyttäytyä. On selvää, että ohjaukseen käytettävät inputit menevät jotenkin ristiin, sillä pelkän fysiikkasimulaation divergoitumisella tämä ei olisi selitettävissä, mutta oli syy mikä tahansa, silti tämä on todella vastenmielistä.

Herään keskellä yötä. Kello näyttää puoli neljä. Olen helpottunut, että oikeasti verkkokoodin toteutus on vasta kaukana tulevaisuudessa. Nyt, jos vielä nukkuisin tunnin pari, olisi kaikki täydellistä.

Päivä 24.

Uskaltaudun ottamaan SVN updaten, ja tarkastelen Renderer-hakemistoa, jossa Sebastianin tämän sprintin työ sijaitsee.

Yllätyn positiivisesti. Sebastian siis terävöityy huomattavasti, kun ei ole kyse prototyypikoodin tuottamisesta, vaan päästään tosijuttuihin.

En sanoisi koodia jumalalliseksi, mutta täällä kuitenkin käytetään vertex buffer objecteja ja shadereita ihan oikein. Immediate mode-renderöintiä ei ole enää havaittavissa edes deferrattujen valojen piirrosta.

Oikea materiaalien statesorttauskin näyttää olevan paikallaan.

No niin: nyt koodissa on vain kolme ongelmaa:

1. Se toteuttaa deferroitua renderöintiä.
2. Se käyttää OpenGL-rajapintaa.
3. Se tukeutuu Qt:n QGLWidget:iin.

Mutta periaatteessa, näiden haittapuolien ulkopuolella, koodi on täysin hyväksyttävää. Olisiko mahdollista, että tästä eteenpäin kaikki lähtisi menemään suunnilleen siedettävällä tasolla? Jos yksikkötestit kirjoittaisivat itsensä, taskit alkaisivat värjäytymään vihreäksi tasaiseen tahtiin, ja ehkä jopa Jarmo järkevöityisi vielä vähän lisää.

Sebastian palaa työasemalleen peli- ja virkistäytymis-huoneesta. Otan hänet kuulusteluun. Hän selittää rende-

röintikoodin toiminnan vielä pääpiirteissään, mikä varmistaa alijärjestelmän hyvän suunnan.

En nyt täysin ymmärrä, mistä tämä muutos johtuu, varsinkin kun vähän aikaa sitten Jörmungandr:in kriisit tuntuivat ylivoimaisilta, mutta nyt yllättäen uskon, että jopa Vision3D on pelastettavissa. Se voi muuten olla ulostetta, mutta renderöintialijärjestelmä ainakin korvaa muiden alijärjestelmien puutteet.

Vaikka se onkin deferrattu, ja se käyttää OpenGL:ää.

Tietysti Jörmungandr on paljon jumalallisempi, jo pelkästään grafiikkarajapintavalinnan vuoksi. Mutta silti, jos on mahdollista koodata yhtä aikaa kahta engineä, joista molemmat pystyvät aikaansaamaan hyväksyttävän 3D-virtuaalimaailmakokemuksen, on se parempi kuin koodata vain yhtä.

Koska jos toinen kuolee pois, on yksi vielä jäljellä. Yksi lisäelämä. Tosin en minä tietenkään voi ajatella, että Jörmungandr kuolisi! Helvetti, ei.

No niin, nyt on tarkoitus ruveta vääntämään Vision3D:n korkeantason renderöintiä.

Frustum culling vaatii tietysti, että reviewaan prototyypivaiheen octree-koodini vielä. Se on Jörmungandr:in asteikolla ala-arvoista, mutta tulee silti nopeuttamaan minkä tahansa scenen piirtoa. Koska eihän kameran takana olevaa batchia tule ikinä piirtää! Vaikka kuinka väitettään, että OpenGL:n piirtokutsut vievät vähemmän aikaa, kuin Direct3D:lla. Ja vaikka ajateltaisiin, että frustum culling on hyvin engine- eikä renderöijäspesifinen operaatio (onhan toki eri asia, että toteutetaanko se quadtreellä, vai octreellä, tai jollain muulla sceneacceleraatiostruktuurilla, mutta Vision3D, kuten myös Jörmungandr päättää, että te haluatte octreen, ettekä tarvitse muuta.)

Koska arvelen, että Visual Studio ei kuitenkaan välttämättä osaa unrollata octree-hierarkian läpikäymiskoodia, unrollaan sen manuaalisesti. No niin, nyt se kelpaa. En jaksa profiloida. Kyllä se on hyvä.

Octree-taski muuttuu vihreäksi.

Nyt kirjoittamaan se frustum culling, ja äkkiä. Sitten päästään kirjoittamaan Vision3D:n hierarkkinen scenenode, ja renderable-komponentti. Oikeastaanhan tässä ollaan menty täysin perse edellä puuhun, eli octree on jo olemassa, vaikka scenenode-luokka ei ole vielä lopullisessa asussaan.

Lisäksi olisi järkevää, jos kamerakomponentti kuuluisi myös tämän sprintin taskeihini. Niin kauan, kuin sitä ei ole olemassa, kaikki renderointitestit ovat hackinomaisia. No, ehkä kirjoitankin sen ylimääräisenä. On luultavaa, että jos kamera pystyy toimimaan sekä perspektiivi- että ortograafisessa tilassa, tämä edellyttää muutoksia G-buffer- ja valaistusshadereihin. Vai olisiko Sebastian niin ylijumala, että olisi ottanut jo tämän huomioon?

Katsotaan.

Eihän hän sentään sitä ole ottanut huomioon. Ortograafisessa projektiossa syvyys- eli Z-arvot ovat lineaarisia, kun taas perspektiiviprojektioissa ne ovat hyperbolisia. Tämä pitää ottaa huomioon joko ajamalla eri shadereita ortografisen kameran kanssa, tai parametrisoimalla syvyysarvojen käsittely. Mutta voin myöntää, että tässä vaiheessa ortografinen projektio on hienostelua (paitsi directional-valon varjojen renderointia varten, joka puolestaan ei vaadi muutoksia, koska shadow mapin itsensä renderointi ei vaadi maailmaposition rekonstruktiota), ja EduWorld tulee pyörimään täysin hyvin pelkästään perspektiiviprojektion avulla.

Mutta päätän joka tapauksessa kirjoittaa kamerakomponentin ylimääräisenä.

Ensin vain scenenode, ja perusrenderable.

Nyt ymmärrän, että olen saavuttanut täyden harmonian universumin kanssa.

Universal coderin täysi potentiaali tulee käyttöön siitä, kun hän fokusoi voimansa – codebasesta riippumatta – täydelliseen ja murskaavaan omni-punch:iin, jolloin taskit valmistuvat kuin combona.

Omni-punch!

Päätän ajatella sitä aina, kun koodaustahtini näyttää hiipumisen merkkejä. Koska vaikka Vision3D kuinka olisi täysi fail, voisin formuloida syvää tyytyväisyyttä siitä, että itse säilytin oman henkilökohtaisen burndown charttini tasaisen laskusuhdannaisena.

No, I never ran, following the Way of Ninja.

Mutta paljon on vielä tekemistä. Todellakin. Päiväkin on vasta puolessa välissä. Sebastian tekee vielä korjauksia ja täydennyksiä shader-uniformien tulkintaprosessiin, ja siten onkin aika nauttia Sodexon antimia. Harvinaislaatuisesti syön epä-antisosiaalisesti hänen seurassaan, ja keskustelemme renderöintiarkkitehtuureista. Ehdotan siirtymistä forward renderöintiin tulevaisuudessa, eikä Sebastian ole täysin sitä vastaan, joskaan ei suoralta kädeltä kovin myötämielinenkään. Tietysti se on joka tapauksessa vasta ensi vuoden puolella, kunhan sovittu toiminnallisuus saadaan ensin kasaan.

Jälkiruokana on mansikkariisi, appelsiiniriisin vähemmän haastava sukulainen.

Iltapäivällä toteutan energisyyden täyttämänä loput renderöintitaskini, jopa staattisen 3D-mallin latauksen. 3D-mallien lataus (ja eri malliformaattien tuki) on versio 1.4:ssä toteutettu aivan riittävän hyvin, joten pääsen vain copy-pastettamaan vanhaa koodia. Sitten vain vielä yksikkötestaus päälle.

Huomaan yliarvioineeni omien taskieni kestot: kaikki on pääpiirteissään valmista, vaikka sprinttiä on vielä puolitoista viikkoa jäljellä. Tämä tarkoittaa, että joudun ottamaan ylimääräisiä taskeja. Tietysti otan niitä vain sopivassa määrin, sillä nyt katson olevan hyvin suotavaa, että kehitän Jörmungandr:ia eteenpäin täysin siemauksin myös toimistolla. Olen mielestäni ansainnut sen.

17:35 teen commitin. Testicaseja on jopa muutama ylimääräinen, ja silti ne kaikki menevät läpi ongelmitta. Oikeastaan en pidä ollenkaan tästä minulle kehittyneestä luonteenpiirteestä, että saan tyydytystä yksikkötestauksesta. Se on edelleenkin tarpeetonta agile-hapatusta. Sel-

västi Jensin luonteenpiirre, ei minun.

Merkitseen taskit vihreiksi. Completed. Completed. Completed. Completed. Completed.

Kaarlo säätää vielä jotain (en halua kysyä, mitä), mutta minun on korkea aika poistua.

Enää ei tuule eikä sada lunta, on vain kuivakan kylmää ja liukasta. Ajan vimman ja energian vallassa: ensin luovutin kapasiteettini Vision3D:n käyttöön, mutta seuraavaksi Jörmungandr saa vielä sitäkin täydemmän huomioni. Se tulee etenemään vähintään yhtä jumalallisesti.

Olen optimoinut reittini siten, ettei minun tarvitse pysähtyä liikennevaloihin juuri ollenkaan. Ainoa, jota en ole saanut vielä optimoitua pois, sijaitsee suunnilleen matkan puolessa välissä.

Perkele!

Noin kahdenkymmenen metrin päässä kyseinen suoja-tien valo on vielä punainen. Arvioin, että se vaihtuu vihreäksi ilman, että minun tarvitsee hidastaa.

Hieman liian myöhään havahdun moottorin revityksen ääneen: joku sieltä on vielä tulossa. Jarrutan voimakkaasti.

Tällöin havaitsen tehneeni toisen virheen: jarrutus aiheuttaa tasapainon menetyksen.

Ennen kaatumistani onnistun väistämään moottoria reivittäneen auton. Tietyn ruumiista irtautuneen ymmärryksen myötä – täysin vailla pelkoa tai kipua – havaitsen, että myös vierekkäistä kaistaa on tulossa auto.

Koska makaan jäisellä asfaltilla, sitä en enää pysty väistämään. Myöskään se ei ehdi väistää minua.

```
var rigidBodyA = scene.getEntity ( "coder" );  
var rigidBodyB = scene.getEntity ( "car" );  
emit onPhysicsCollision( rigidBodyA, rigidBodyB,  
    worldPosition, worldNormal, impulse );
```

Pimeys.

SHA – Itsensä ja muiden parantaminen

Aika: tuntematon.

Paikka: tuntematon.

Hän tulee tajuihinsa, ja havaitsee, että jokin ei ehkä ole oikein. Kivusta ei varsinaisesti voi puhua, koska sitä ei tunnu, mutta jokin hänen kehossaan on muuttunut.

Ylhäältä siivilöityy hämärää valoa. Sinne on pitkä matka. Hän havaitsee makaavansa ison, suorakaiteenmuotoisten teräskontin päällä. Kaikkialla ympärillä on samanlaisia kontteja.

Näön vihdoin terävöityessä hän pystyy formuloimaan mielessään selvemmin tapahtuneen muutoksen. Hänen oikea kätensä ei ole enää lihaa, vaan terästä. Hän koskettaa niskaansa vasemmalla kädellään, jossa ainakin on vielä tunto tallella, ja sielläkin, pitkän ja sotkuisen tukan alla, tuntuu metallinen kylmyys.

Hän ei vielä muista koko tapahtumaketjua – ei läheskään kaikkea – mutta joku – tai jotkut – ovat selvästi tehneet tämän hänelle.

Mutta miksi? Miksi ihmeessä?

Ja miten hän on päätynyt tänne?

Looginen arvaus on, että hän on pudonnut – tai pudotettu – tänne alas. Miten hän voi olla hengissä, tai suhteellisen vahingoittumaton, selittyy niillä kyberneettisillä muutoksilla, joita hänen ruumiilleen on tehty, mitä ilmei-

simmin vasten hänen tahtoaan.

Kyllä: repaleisten housujen alta pilkottaa lisää terästä. Myöskään hänen jalkansa eivät ole enää ihmislihaa, aina-kaan kokonaan.

Mutta syvempien vastauksen löytämiseksi on ensin kii-
vettävä ylös tästä loukosta.

“Uskonpa, että se on alkanut,” sanoo asuntoauton taka-
osassa kannettavan tietokoneen äärellä lysähtäneesti istu-
va nuorehko mies. Hänen silmiensä palava katse ja hauk-
kamainen ilme ovat ristiriidassa muuten väsähtäneelle
olemukselle.

“Skeptikkojen olemassaolo ei siis enää ole turhaa? Vaik-
kakaan se ei ole syy iloita,” vastaa toinen mies ratin takaa.

“Maailmanloppu ei yleisesti ottaen ole. Tai tarkemmin
ottaen pitäisi kai sanoa, että se on ollut käynnissä jo jonkin
aikaa. Poistu seuraavasta, sitten oikealle.”

Asuntoauto kääntyy moottoritien poistumisrampille.

Hän havaitsee paksut rautakettingit, joita ilmeisesti käytetään konttien laskemiseen ja nostamiseen. Ihmisen lihaksille ja jänteille noin kolmenkymmenen metrin kiipeäminen niitä pitkin suoraan ylöspäin olisi hyvin haastavaa, mutta teräsjäsenet eivät väsy, kunhan voimanlähde – mikä se sitten onkaan – ei anna periksi.

Ja kunhan uusia ruumiinosia oppii ensin hallitsemaan.

Hän lähtee rohkeasti, liiemmin harjoittelematta kiipeämään – logiikkana on, että jos hän selvisi koko pudotuksesta alas, ei ole läheskään niin vahingoittavaa, jos hän nousee esim. viisi metriä, ja putoaa sitten.

Juuri näin tapahtuu. Metalliset sormet eivät saa otetta kettingistä, niinkuin pitäisi.

Hän kiroaa pudotessaan kontin päälle kumean metallisen kolahduksen säestämänä. Tällä kertaa hän harjoittelee muutaman minuutin, nousee pari askelta, sitten laskeutuu, ja varmistuu koukistamalla ja ojentamalla keinotekoisia sormiaan, että osaa hallita niitä yksitellenkin.

Oikea käsi on kokonaan terästä, mutta vasen käsi – jossa tunto on tallella – näyttää jonkinlaiselta biomekaaniselta hybridiltä. Ehkä siis kehittyneempi versio. Onko liha metalliluiden ympärillä hänen omaansa, sitä hän ei voi tietää. Mutta on outoa, ettei hänelle asennettu kahta samanlaista kättä.

Lopulta hän uskoo olevansa valmis.

Ylöspäin siis.

Kaksikymmentä metriä sujuu ongelmitta. Vasen – oikea – vasen. Hyvin menee.

Mutta sitten hänen otteensa pettää taas.

Viime hetkellä hän saa kiinni kettingistä, ja roikkuu siitä yhden käden – täysteräksisen – varassa. Voimaa siinä riittää hyvin, vaikka hänen kehonsa täytyykin nyt kokonaisuudessaan painaakin enemmän.

Varsinaista pelkoa ei ole, vain turhautunut ajatus siitä, että joutuisi aloittamaan taas alusta. Hän heilauttaa itsensä takaisin ja puristaa molemmat jalat kettingin ympärille.

Sitten jatketaan.

Vihdoin hän saavuttaa ison avonaisen luukun, josta valo siivilöityi alas, ja havaitsee olevansa isossa, hämärässä varastohallissa. Täällä kontteja on lisää. Nyt hän voi keskittyä lukemaan niiden kyljessä olevan tekstin:

THRONE GROUP

Mikä ihmeen valtaistuinryhmä? Jotenkin hänelle tulee mieleen myyttinen Illuminati. Aivan ilmiselvästi hänen täytyy olla keskellä suurta salaliittoa, jos hän herää maanalaisesta varastosta, lihaa korvattuna metallilla.

Oikeastaan hän hämmästyy suhteellisen tyyntä suhtautumistaan. Olisi luonnollista olla raivoissaan, jos olisi menettänyt ruumiinosia vasten tahtoaan. Mutta jotenkin tämä tuntuu edistykseltä, siltä, että nyt hän on enemmän sitä, mitä haluaisi olla.

Filosofiset pohdinnat kuitenkin keskeytyvät, kun hän havaitsee lattialla verilammikoissa makaavat ruumiit. Kolme miestä ja kaksi naista, kaikilla paitsi yhdellä heistä laboratoriotakit, jotka (kuten myös heidän kehonsa) jokin suuritehoinen konetuliase on rei'ittänyt.

Nämä henkilöt ovat luultavasti vastuussa siitä, mitä hänelle on tehty. Mutta enää he eivät voi tarjota selitystä.

Hän havaitsee liikettä varjoissa, edessäpäin. Vaistomaisesti hän syöksyy kohti lähimmän kontin suojaan.

Olisipa hänellä ase.

Kyllä: aseet eivät suinkaan ole hänelle vieraita. Asia ei vain tullut mieleen ylös kiivetessä: vaikka tähän johtaneet tapahtumat ovatkin yhtä sumua, kokonaan hän suinkaan ole menettänyt muistiaan siirtojäsenten myötä: hän on Kim, ammatiltaan yövartija, ja työ sopii hänelle, koska näin hänen harvoin tarvitsee tavata toisia ihmisiä.

Hänet voisi luokitella hyvinkin misantrofiseksi: tätä tukee myös hänen musiikkimakunsa, johon kuuluvat esimerkiksi Thergothon ja Abruptum.

Hän pidättää hengitystään ja odottaa, että jotain tapahtuisi. Vaikka ulkoa ikkunoiden läpi tuleva valo onkin hämärää, noin kymmenen sekunnin odotuksen jälkeen hän havaitsee pitkän varjon heittäytyvän lattian poikki.

Varjo ei selvästikään kuulu ihmiselle.

Kahdella jalalla kävelevä suippopäinen mustankiiltävä robotti tulee esiin. Sillä on käsissään pitkänomainen, kiivääriltä vaikuttava ase, joka on kiinnitetty paksulla kaapelilla robotin runkoon.

Helvetti.

Luultavasti, vaikka robotin saisi eliminoidua ja aseensa reivityä irti, ei siitä olisi hyötyä. Ase todennäköisesti vaatii yhteyden robotin voimanlähteeseen.

Kone kääntyy. Helvetti uudemman kerran. Sillä on ehkä valonvahvistus, tai infrapunankäyttö, tai jotain.

Mutta selvästi se on havainnut hänet.

Kim syöksyy raivon vallassa eteenpäin, tarkoituksenaan pitää vartalonsa aseensa piipun alapuolella, ja törmätä robottiin brutaalilla voimalla.

Ensimmäinen osa suunnitelmasta onnistuu. Robotti avaa infernaalisen hypernopeussarjatulen liian myöhään, eikä saa käännettyä piippua häntä kohti. Törmäyskin tapahtuu, mutta sillä ei tunnu olevan kovin suurta vaikutusta. Musta kone on häntä selvästi painavampi. Molemmat kuitenkin menettävät tasapainonsa: robotti kaatuu lähes hänen päälleen, ja Kim saa kierähdettyä alta pois viime hetkellä.

Ketteryys on ehkä hänen puolellaan. Ennen kuin kone

pääsee kääntymään ja nousemaan pystyyn, hän hyppää sen niskaan kiinni, ja alkaa puristaa ja riuhtoa, toivoen osuvansa johonkin elintärkeään.

Jotain napsahtaa poikki. Robotin selkäranka? Mutta Kim ei tyydy siihen, vaan jatkaa repimistä, kunnes lopulta koko pää – joka muistuttaa kaasunaamariin puettua mustaa vuohenpäättä – irtoaa.

Silti kone liikkuu, ja yrittää ilmeisesti kääntää asettaan häntä kohti. Mutta sen liikkeistä paistaa jo läpi sattumanvaraisuus.

Kim väistää aseensa piippua toisen kerran, pakottaa sen kohti robotin torsoa – tämä onnistuu, koska ilmeisesti pään irtoaminen aiheutti myös, ettei se ymmärrä enää panna täysillä vastaan – ja kampeaa kaikin voimin robotin paksun etusormen liipaisinta vasten.

Ase herää eloon, ja kipinät sinkoilevat, kun suurikaliberinen sarjatuli alkaa tehdä selvää robotin panssarista. Lopulta luodit alkavat mennä läpi, ja metallitorson sisältä kantautuu omalaatuinen digitaalinen ulina, ennen kuin robotti – kuten myös ase – sammuu.

Tappajakone makaa liikkumatta lattialla. Kim nousee pystyyn ja tarkastelee sitä vielä muutaman sekunnin ajan. Kyllä. Elottomalta se vaikuttaa.

Mutta sen kaltaisia saattaa olla lisää.

“Kolme päivää sitten AboveTopSecret-foorumille tuli viesti, jonka kirjoittaja väitti kuuluvansa ns. Throne-ryhmittymään. Hän kertoi tämän huomattavan varakkaan, itsenäisesti toimivan ryhmän tutkineen teknologiaa, jonka avulla ihmiskunta säilyisi elossa siinä vaiheessa, kun Maa tulee asuinkelvottomaksi,” etumatkustajan penkille siirtynyt mies referoi.

“Tuohan ei kuulosta ollenkaan pahalta.”

“Odotahan. Ryhmä laski, että mahdollinen toinen elinkelvoinen planeetta olisi kuitenkin niin kaukana, ettei matkaan lähtevillä avaruusaluksilla pystyittäisi ylläpitämään elämää perille saapumiseen asti, ellei sitten löydetäisi keinoa matkustaa lähes valon nopeudella. Ratkaisumalleja oli kaksi: syväjäädysteknologia, ja autonomiset, osittain biomekaaniset robotit, joiden koneaivoihin ladattaisiin ihmisyyden olemus, jotta ne jatkaisivat ihmiskulttuuria uudella kotiplaneetalla. Jälkimmäinen osoittautui lupaavammaksi.”

“Anna kun arvaan. Onko meillä käsillä Skynet-skenario?”, kuski kysyy.

“Mahdollisesti. Tekstissä viitattiin robottien riistäytyneen hallinnasta, muttei menty tarkempin yksityiskohtiin. Eilen viestin kirjoittaja löytyi kuolleena asunnoltaan. Ja jotain muutakin outoa on tapahtumassa. Noin vuorokausi sitä ennen tapahtui pakettiauton ja jalankulkijan yhteen-törmäys. Sille oli silminnäkijöitä. Laboratoriotakkeihin pu-

keutuneet matkustajat nostivat uhrin kyytiin, ja häipyivät paikalta. Pakettiautossa luki Throne Group.”

“Jos tuo on totta, kuulostaa varomattomalta toiminnalta. Nimi olisi kannattanut maalata tai teipata piiloon. Mutta kerro jo, minne olemme menossa?”

“Viestin kirjoittajan asunnolle.”

“Mutta sehän on mahdollisesti rikospaikka? Ja jos tuo Throne Group ovat niin korkean tason operaattoreita kuin kuvittelen, se on taatusti jo putsattu.”

“Sittenpä näemme. Saa nähdä, tuleeko tästä ensimmäinen ‘mahdollinen’ tai jopa ‘todennäköinen’ -tason skenaariorio, mistä raportoimme. Jälkimmäisessä tapauksessa meidän on syytä koota koko tiimi pikaisesti.”

“Maailmanlopun raportoinnista sen ehkäisyyne?”

“Juuri niin.”

Miehet ovat nimeltään Jeff ja Nathan. Ensinmainittu heistä toimii usein kuskina, jälkimmäisen kerätessä ja yhdistellessä informaatiota sekä liikkeellä että paikallaan, usein jopa ympäri vuorokauden. Yhdessä he ylläpitävät <http://www.skepticsapocalypse.org> -sivustoa, jossa raportoidaan väitetyistä maailmanlopun skenaarioista, ja tarjotaan vinkkejä niiden varalle. Skeptikoiksi ironisesti nimetty tiimi kokoontuu säännöllisin väliajoin osin leikki-mielisiin harjoituksiin maailmanlopun varalta, jotta tosi-paikan tullen heillä olisi paremmat mahdollisuudet selvittää ja opastaa muita. Viime kuun harjoitus käsitteli zombiapokalypsiä. Ryhmän erittelemät maailmanlopun muodot ovat kokonaisuudessaan seuraavat:

- 1. Ydintuho ja/tai sen synnyttämät anomaliat*
- 2. Skynet-skenario*
- 3. Biologinen tai kemiallinen uhka*
- 4. Äkillinen ilmastonmuutos tai muu luonnonmullistus*
- 5. Taivaankappaleen törmäys*
- 6. Maan kiertoradan muuttuminen*
- 7. Zombihyökkäys*
- 8. Vampirismi ja/tai ihmissudet*

9. UFO:jen laskeutuminen ja hyökkäys

10. Interdimensionaaliset tai raamatulliset olennot: Suuret Muinaiset, Antikristus jne.

Olisi myös mahdollista, että maailmanloppu olisi yhdistelmä useita kategorioita, esim. numerot kolme ja seitsemän, tai numero viisi, josta seuraisi numero neljä.

Metallisarana narahtaa matalaäänisesti: se saa Kim:in selkäpiin karmimaan, vaikka hän ei tiedäkään, onko se enää hänen omansa, vai teräksellä (tai titaanilla) korvattu.

Yksi konteista on aukeamassa. Lisää koneita?

Hän kiroaa hiljaa.

Kontista tulee ulos valkotakkinen mies kädet ylhäällä. Ei näkyviä aseita. Kim ei pysty luokittelemaan häntä millään muotoa uhaksi: jos tiedemies yrittäisi jotain, hänet saisi helposti neutraloitua muutamalla (tai jopa yhdellä) metalliraaajan iskulla tai potkulla.

“Olet elossa. Hyvä,” mies sanoo hivenen koleasti.

Kim tuijottaa miestä epäuskoisesti muutaman sekunnin.

“Mistä vitusta on kyse?” hän tiuskaisee sitten.

Jotenkin ihmisen näkeminen, ja vielä tämän epämiellyttävään vivahtava käytös saa Kim:in vihan nousemaan, vaikka hän hetkeä aikaisemmin pystyikin analysoimaan tilannettaan rauhallisesti.

“Aikaa ei ole paljon. Voin yrittää kertoa matkalla.”

“Hetkinen. Sinä et todellakaan sanele minulle, mitä tehdään. Minä en lähde millekään matkalle. Kerro nyt heti, tai voisin vaikka kokeilla, kuinka hyvin nämä kädet taittavat niskasi.”

“Ok. Yritän. Me – Throne Group – olemme lievästi sanoen kusessa. Kehittämämme robottiteknologia riistäytyi hallinnasta noin kuukausi sitten. Sinä olit – olet – paras

kandidaattimme tilanteen korjaamiseksi. Mutta kaikki ryhmämme sisällä eivät ole asiasta samaa mieltä. Joidenkin mielestä mitään korjattavaa ei ole, vaan tässä nähdään suunnitelmamme ja prototyyppiemme toimivuus käytännössä jo etukäteen. Norman Throne – pomomme, jonka kanta on juuri tämä – varmaan paljasti olinpaikkamme Konstruktille, johtavalle tekoälylle...”

“Konstrukti. Kuulostaa ihan joltain Matrix-paskalta. Mutta te siis päätitte minulta kysymättä, että olisi hyvä idea repiä minun käteni ja jalkani irti, ja korvata ne näillä...”

“Pahoittelen. Ei ollut aikaa kysyä. Tai kokeilimme me ensin vapaaehtoisilla – omillamme – mutta he eivät kestäneet transplantteja. He sekosivat, jos eivät kuolleet sitä ennen.”

“Mihin te sitten tarkalleen tarvitsette minua?”

“Tunkeutumaan maanalaisiin käytäviimme ja tutkimuslaitoksiimme, jotka ovat nyt robottiemme hallussa. Selvitämään, minne koneet tarkalleen rakentavat – tai ovat ehkä jo rakentaneet – Konstruktin johdolla valtavaa mekaanista käärmettä, Jörmungandr:ia, jotta se voi nousta hengittämään myrkkynsä ilmakehään. Silloin kaikki kuolevat...”

Kim miettii hetken. Tarina on kokonaisuudessaan naurettava. Ainoastaan se mustanpuhuva metallihirviö, jonka hän hetki sitten kohtasi, ei ollut lainkaan naurettava. Mutta taas se, että hän olisi nyt jotenkin velvollinen tanssiin tämän miehen pillin mukaan, on jo ideatasolla todellakin täysin absurdia.

“Arvaapa pariin kertaan, suostunko.”

“Kyllä, kyllä olen tietoinen dilemmasta. Seulonnan perusteella olit lähes ihanteellisesti sopiva, sekä fyysisesti että henkisesti. Riittävän misantrofinen, jotta selviäisit prosessista järjissäsi. Mutta ehkä liian misantrofinen, jotta auttaminen merkitsisi sinulle mitään.”

“Hm.”

“Joka tapauksessa, minä olen Amos, kyberneetikko ja

apulaiskirurgi. Thomas, pääkirurgi, kuoli robottien tulituk-
sessa, kun ne hyökkäsivät hetki sitten.”

“Pitäisikö sanoa, että hauska tavata. Hah. No ei todella-
kaan. Sinäkö heitit minut tuosta luukusta alas?”

“Ei. Thomas teki sen, juuri ennen kuin tuli rei’itetynsi.
Et ollut hyökkäyksen alkaessa vielä toimintakunnossa.
Mutta haluan mainita yhden asian, joka saattaa vaikuttaa
päätökseesi. Muutokset elimistöösi olivat niin syvällisiä,
ettet selviä ilman transplanteja. Ja ne puolestaan tarvitse-
vat virtaa. Pistorasiasta, ei edes kolmivaiheisesta, ei aivan
saa tarpeeksi. Fuusiokennoista kylläkin. Varastomme tu-
houtui tulituksessa, joten ensimmäinen prioriteetti on
päästä johonkin tutkimuslaitokseemme, joka ei vielä ole
koneiden hallussa, jotta saamme pidennettyä ikääsi. Us-
koisin, että sinulla on noin vuorokausi jäljellä tuolla ken-
nolla. Aseistautuminenkaan ei olisi huono idea. Mutta mei-
dän täytyy varautua, että jos eivät robotit, niin Norman:in
eliittijoukot voivat odottaa meitä.”

“Oletetaan nyt että uskon sinua. Mitä se kuspipää hyötyy
siitä, että teidän peltiromujenne suunnitelmat toteutuisi-
vat?”

“Ei hän hyödykään mitään. Tai korkeintaan tietoisuu-
tensa siirtämisen osaksi Konstruktia. Mutta silti hän kuoli-
si itse.”

“Ei kuulosta kovin rationaaliselta.”

“Hän onkin nero, ja hullu. Koko idea, mistä kaikki sai al-
kunsa, vaati visionääristä hulluutta. Kuinka ihmiskunta
selviäisi matkasta toiselle planeetalle, jos tarve tulisi. Vas-
taus oli: ei ihmisinä...”

Kim:in on pakko myöntää, että hän tuntee tietynlaista
sympatiaa Throne Group:in pyrkimykseen vapautua ihmi-
syyden kahleista. Itse ryhmän jäseniä kohtaan hän ei kyllä
tunne mitään, ellei sitten vihaa.

Vai tuntisiko hän sittenkin kiitollisuutta? Tästä muutok-
sesta?

Ei. Kiitollisuus ei ole tunne, jota hän tunnustaisi. Eikä
varsinkaan tässä tilanteessa. Ajatus tiukan rajoitetusta

elinajasta saa hänen raivonsa nousemaan taas pinnalle, mutta hän pyrkii hillitsemään sen. Siitä ei tällä hetkellä ole hyötyä.

Tai entä jos tiedemies kusettaa? Entä jos mitään aikarajaa ei todellisuudessa ole?

Mutta selvää on, että hänen olemassaolonsa parametrit ovat joka tapauksessa peruuttamattomasti muuttuneet. Hän ei ainakaan tule ikinä palaamaan työhönsä.

Jos hän siis oikeasti on riippuvainen kennoista, eräs mahdollisuus on esittää tottelevaista niin kauan, kun pääsee käsiksi niitä sisältävään varastoon, rohmuta niitä niin paljon kuin jaksaa kantaa, ja yksinkertaisesti häipyä. Throne Group saa jäädä selvittelemään omia ongelmiaan.

Kim vilkaisee sen kontin sisään, josta Amos tuli esiin. Se on varustettu sisältä leikkaussaliksi. Se on täynnä laitteita ja instrumentteja, jotka eivät merkitse hänelle mitään. Lattialla on useita veritahroja. On selvää, että transplantaatio-operaatio oli kohtuullisen karkea luonteeltaan. Häntä puistattaa lievästi, kun hän miettii sen yksityiskohtia.

Hän havaitsee myös mustat ruumissäkit takaseinällä. Ei kovin hygienistä, jos hänet leikattiin ruumiiden läheisyydessä. Yksi niistä on auki: nyt hän ymmärtää, miten Amos selvisi hyökkäyksestä.

“Oletko katsellut ympärillesi tarpeeksi? Nyt olisi hyvä aika poistua. Autoni on tämän hallin takana. Toivotaan, että robottien tehtävä oli pelkästään ihmisten, ei ajoneuvojen tuhoaminen.”

Sanasta “ajoneuvo” Kim muistaa jotain. Hänen työvuoronsa oli juuri päättynyt, hän oli poistunut kohteesta, kun hän yhtäkkiä näki lähestyvät kirkkaat valot, joita seurasi voimakas isku ja täysi pimeys. Mutta hän ei ollut kuullut mitään. Ei kierroksia ottavan moottorin ääntä, joka olisi voinut varoittaa häntä.

“Te ajoitte päälleni, jotta saisitte minut.”

“Juuri niin. Sähkökäyttöinen pakettiautomme on varsin äänetön, mutta kiihtyy silti tehokkaasti. Niitä on myynnissäkin, tosin ei Throne Group:in nimellä. Laskimme riittä-

vän törmäysnopeuden, joka luultavasti ei olisi fataali...”

“Te kusipäät. Minun oikeasti pitäisi vain tappaa sinut ja häipyä.”

Kim ja tiedemies tuijottavat toisiaan kylmästi usean sekunnin ajan.

“Ehkä noin vuorokausi. Muista se.”

Nathan irroittaa keltaisen "CRIME SCENE - DO NOT CROSS" -tarran ovesta. Tarran alta paljastuu, että ovi ei voi olla lukossa, koska lukkoa ei enää ole. Miehillä on hansikkaat kädessä, kun he astuvat sisään yrittämättä aiheuttaa pienintäkään ääntä.

"Rikospaikan häirintä on pikkujuttu, jos olemme tutkimassa ja estämässä alkavaa maailmanloppua," Nathan kuiskaa, kun he ovat sisällä, ja sulkeneet oven perässään.

"Näytti, että lukko oli ammuttu rikki. Siis ilmiselvä murha. Siitä kai ei puhuttu uutisissa mitään?"

"Ei. Tutkinnan yksityiskohtia ei ikinä paljasteta näin aikaisessa vaiheessa."

Taskulamppujensa valossa miehet alkavat tutkia huoneistoa, joka käsittää olohuoneen, makuuhuoneen, keittiököneron ja pesuhuoneen.

Lukuunottamatta kuivunutta verta kaikkialla olohuoneessa mitään poikkeuksellista ei löydy.

"Itse murhan on täytynyt olla sotkuinen operaatio. Mutta muuten kaikki on liian siistiä. Luulisi, että huonekalut olisivat poissa paikoiltaan, jos kirjoittaja on kamppailut henkensä edestä," Jeff toteaa.

Nathan nyökkää vastaukseksi. He tutkivat lipastot ja kaapit salaisten dokumenttien varalta.

Ei mitään.

Ei varsinkaan tietokonetta. Tosin jos sellainen löytyisi, se olisi luultavasti suojattu vahvalla salauksella.

“Joko murhaaja – tai murhaajat – siistivät paikan heti teon jälkeen, tai sitten palasivat tänne rikospaikkatutkin-
nan päätyttyä. Mutta oikeastaan tämä on hyvä merkki. Se osoittaa, että kirjoittaja oikeasti tiesi jotain merkittävää, ja hänet täytyi vaientaa. Kyse ei ollut mistään satunnaisesta hullusta,” Nathan toteaa.

Jeff pudistaa päätään epäuskoisesti.

“Mutta nyt meidän lienee syytä poistua.”

“Ehkä –“

Nathan suuntaa kohti keittiökomeroa.

“Ehkä me – tai tekijät – eivät ole huomanneet kaikkea.”

Hän avaa jääkaapin ja pakastimen, ja alkaa käydä niiden sisältöjä läpi. Valmisruokapakkauksia ja leipää on varas-
toituna huomattava määrä.

“Bingo.”

Erään pakkauksen pahvisen ulkokuoren ja muovisen si-
säkuoren välistä löytyy kokoonlaiteltu paperi.

“Tämä kuvaa jotain laitosta. Luultavasti Throne Groupin
laboratorio. Siinä on myös sijainti. Pituus- ja
leveysasteet...”

Äkkiä yläpuolelta kuuluu jotain kolinaa.

Nathan katsoo ylhäälle kattoon. Eihän siellä ole mitään.
Mutta seuraavaksi metallinen kolina kuuluu suoraan hän-
nen edestään, voimakkaampana.

Ennen kuin hän ehtii reagoida, liesituulettimen säleikkö
irtoaa ja putoaa voimalla alas lattialle, ja aukosta ilmestyy
minikokoinen, mutta silti tukevarakenteinen helikopteria
muistuttava musta, lentävä kone, ehkä noin viisitoista
senttiä pitkä kokonaisuudessaan. Pieni punainen led-valo
palaa sen nokassa.

Nathan syöksyy sivulle, mutta lentolaite ikään kuin aa-
vistaa hänen aikeensa. Propelli – tai jokin muu koneessa
kiinni oleva terä – viiltää hänen naamaansa syvään.

“Saatana!” hän huutaa tuskan läpi.

Jeff ei jää ihmettelemään mekaanista tunkeilijaa, vaan
tarttuu Nathania käsivarresta, ja lähtee vetämään häntä
kohti ovea.

“Nyt häivyttään ja heti!”

Kone seuraa heitä huoneiston halki, mutta he onnistuvat pääsemään asunnosta ulos porraskäytävään ja lyömään oven kiinni sen nokan edestä.

Nathan tunnustelee poskeaan, jota pitkin veri valuu, samalla kun odottaa hengityksensä tasaantumista.

“Oletko kunnossa?” Jeff kysyy.

“Riittävän. Ne jättivät tuon pirulaisen tänne vartioimaan. Uskon, että se toimi täysin itsenäisesti, ei kauko-ohjattuna... Mutta ei mietitä sitä tässä.”

Miehet poistuvat portaita alas välittämättä enää rikospaikka-tarrasta. Salassa pysymisen vuoksi he jättivät auton ehkä liian kauaksi asunnosta: nyt kahdensadan metrin matka tuntuu kiduttavan pitkältä. Nathanin kulku on haa-voittumisen jäljiltä horjuvaa, ja Jeff joutuu tukemaan häntä.

Noin viidenkymmenen metrin päässä autosta heidän takaansa kuuluu vaimea lasin rikkoutumisen ääni. Nathan kääntyy katsomaan, ja havaitsee kolmannen kerroksen ikkunan särkyneen. Sirpaleet putoavat juuri katuun.

Juuri se huoneisto, jossa he olivat.

“Se perkele ajaa meitä vielä takaa!” hän puuskuttaa.

Miehet lisäävät vauhtiaan. Kun autolle on kymmenen metriä, Jeff avaa keskuslukituksen. Vilkkujen oranssit valot välähtävät.

Nyt lentolaitteen surina kuuluu jo. Aikaa ei ole paljon.

Jeff ja Nathan syöksyvät paikoilleen ja sulkevat ovet perässään. Agitaation vallassa Jeff pudottaa virta-avaimen lattialle, mutta löytää sen suhteellisen nopeasti, ja käynnistää moottorin.

Lentävä kone kiertää auton eteen ja alkaa raapia tuulilasiasia. Ääni on korkean riipivä. Jeff laittaa peruutusvaihteen päälle ja painaa kaasun pohjaan, mihin laite ei osaa reagoida riittävän nopeasti, vaan jää leijumaan ilmaan hetkeksi. Nopea kolmipistekäänös ja poistuminen seuraavasta risteyksestä isommalle tielle jättää koneen lopullisesti jälkeen.

“Skeptikot täytyy kutsua koolle välittömästi,” Nathan sanoo, ja alkaa näppäillä kännykkäänsä.

Ohikiitävien katulamppujen valossa Jeff havaitsee tämän vaikuttavan poikkeuksellisen kalpealta.

“Hei, oikeasti, oletko kunnossa? Pitäisikö meidän suunnata lähimpään sairaalaan?”

“Ei, ei sotketa tähän mitään viranomaisia. Me emme voi tietää, ketkä ovat mukana tässä... Mennään vain Skeptikkojen tukikohtaan.”

“Tukikohtaan,” erääseen käytöstä poistettuun syrjäseudun pienkonekorjaamoon, josta Skeptikot maksavat nimelistä vuokraa, on noin puolentoista tunnin ajomatka.

“Luuletko, että tuo kone tappoi foorumipostaajan?” Jeff kysyy heidän siirtyessään taas moottoritielle.

“Ei ainakaan yksin. Se ei olisi saanut lukkoa rikki sillä tavalla. Siihen tarvittiin ampuma-asetta tai räjähdettä.”

“Ymmärrän. Siinä tapauksessa, kuten sanoit, täytyy olla, että se jätettiin sinne vartioon, ja poliisit saivat olla rauhassa, koska eivät joko löytäneet mitään, tai koska heidän kimppuun hyökkäämisensä olisi herättänyt liikaa huomiota.”

“Niin.”

Jeff miettii hieman. Hänen omat sanansa herättävät epämiellyttäviä, ristiriitaisia ajatuksia.

Koko tilanne tuntuu uhkaavalta ja luotaantyöntävältä, eikä hän suinkaan haluaisi olla sen keskellä. Foorumikirjoittajan kuolema ja heidän kohtaamansa murhanhimoisen lentolaite voivat hyvinkin merkitä sitä, että postaus oli sataprosenttisen totta, että Throne Group todella on valmistanut kehittyneitä robotteja, ja että skenaario numero kaksi on täydessä käynnissä. Hän tietää kyllä, että osa skeptikoista tulee olemaan tästä hyvinkin innoissaan. Kun he pääsevät tositoimiin, joita ovat odottaneet ehkä puoli elämäänsä.

Varsinkin Micha, jonka uhitteleva motto – eräästä nuclear metal -kappaleesta lainattu – kuuluu seuraavasti: *Give me finally my Armageddon.*

Hallin takaa paljastuu kolhiintuneen ja ruosteisen näköinen sedan, joka kuitenkin vaikuttaa vahingoittumattomalta.

“Näyttää, etteivät koneet ole koskeneet tähän,” Amos sanoo.

Kim ei vastaa, vaan odottaa, että tiedemies avaa etu-ovet, ja menee sitten matkustajan paikalle. Hän joutuu säättämään istuinta taaksepäin. Pitkähkö hän oli jo ennen – 185 cm – mutta teräsjalat ovat vielä hänen omiaan vähän pitemmät.

“Kuinka kaukana se labra on?” hän kysyy lopulta, kun he ovat ajaneet parin minuutin verran.

“Liian kaukana, että sinne voisi ajaa. Mutta lähin yksityinen lentokenttämme on kolmen tunnin ajomatkan päässä.”

“Jaha. Osaatko muka lentää?”

“Minulla on lupakirja pienkoneille. Mutta lentokoneemme eivät ole ihan konventionaalisia. Luotan kyllä siihen, että pääsemme perille, kunhan vain saamme koneen haltuun. Tosin siinä voikin tulla ongelmia. Jos Norman on tarpeeksi paranoidi, hän on saattanut vaihtaa kentän henkilökunnan.”

Seuraavan tunnin he ajavat puhumatta sanaakaan. Kim inhoaa avuttomuuden tunnetta: hän itse ei tietäisi, minne mennä, mutta tämä mies tietää. Tietysti hän voisi yrittää puristaa tiedemiehestä ulos kaiken mahdollisen informaation.

tion: Throne Groupin laitosten sijainnit, mahdolliset koodiluvut ja tunnussanat...

Mutta entä sitten?

Kim kuitenkin tuntee tyytyväisyyttä ajatuksesta, että voisi milloin tahansa surmata miehen, ilman että tämä voisi tehdä paljon mitään. Paljon inhottavampaa olisi esimerkiksi, jos miehellä olisi kaukosäädin, jolla transplantit voisi kytkeä pois päältä. Tai jotain vielä ilkeämpää.

Keskittyessään ja sulkiessaan tavanomaiset aistit pois tietoisuudestaan Kim havaitsee, että ilmeisesti transplantien ansiosta hän pystyy aistimaan läheisten sähkölaitteiden toiminnan. Moottorin sytytyskipinät tuntuvat pieninä, tiheään tahtiin tulevina nipistyksinä, jos oikein keskittyy. Outoa.

“Helvetti,” Amos kiroaa yllättäen. “Öljynpainevalo sytyi.”

“Sabotoitiinko autoa sittenkin?” Kim kysyy, vaikkei häntä varsinaisesti kiinnostanut miehen vastaus.

“En usko, että robotit tekisivät mitään niin hienovaraisista. Koko auto ei olisi käynnistynyt, jos ne olisivat tehneet sille jotain. Mutta moottorin lämpö alkaa lähestyä punaista. Kohta täytyy pysähtyä, ettei se leikkaa kiinni. Paska tuuri, ja paska auto.”

Kim naurahtaa kylmästi. Häntä ei juurikaan haittaa ajatus, että joutuisi kävelemään. Hän luultavasti jaksaisi – kunhan fuusiokennossa riittää virtaa – paljon pitempään ja paljon kovempaa tahtia kuin tiedemies.

“Nyt ollaan punaisella. Pakko pysähtyä. Näin yöllä ei kyllä mikään paikka ole auki, mutta ehkä saamme kyydin joltakulta.”

“Siltä joltakulta tulee kyllä paskat housuun, kun näkee minut. Mutta joo, voimme vaikka kaapata jonkin kulkupelin,” Kim toteaa tyyliä.

Amos hidastaa ja pysähtyy tien reunaan, ja sammuttaa moottorin. Konepellin alta nousee jo vesihöyryä. Kim tuntee olonsa selvästi rauhallisemmaksi, kun sytytystulppien aiheuttama nipistely lakkasi.

Hetkinen. Mutta mitä tuo on?

Sekunnin välein toistuva pulssi. Auton kello? Kim kumartuu lähemmäksi keskikonsolia, mutta se se ei ole.

Pulssi tuntuu ennemminkin tulevan auton alta.

“Ulos ja heti!” Kim huutaa. “Pois auton luota!”

Amos tottelee kysymättä. He nousevat autosta ja juoksevat pois päin.

Kuluu ehkä kymmenen sekuntia.

Sitten iso tulipallo valaisee yön, kun sedan räjähtää ja lakkaa olemasta. Ehkä muoviräjähdettä. Kim tuntee inhoituksen käyvän lävitseen: ei olisi ollut mukava olla sisällä. Itseen hän on toisinaan inhonnut lähes yhtä paljon kuin muita, mutta nyt, kun hän tavallaan on vähemmän ihminen, itsesuojelu tuntuu ehkä tärkeämmältä kuin ennen.

Amos nousee jaloilleen pientareelta, minne hän oli heittänyt.

“Tarkemmin ottaen olen hyvin tyytyväinen, että auto alkoi vuotaa öljyä juuri nyt,” hän toteaa kuivasti.

“Koneet ovat siis fiksumpia kuin ajattelisi,” Kim miettii ääneen.

“Niin no, eivät ne varsinaisesti ole. Mutta Konstrukti kykenee hyvinkin edistyneeseen ajatteluun. Se laskee läpi erinäisiä skenaarioita, ja juuri tällä hetkellä me edustamme sille hyvin suurta uhkaa. Siksi se käski asentaa pommien, kaiken varalta.”

“Kuvitteleeko se nyt, että olemme kuolleita? Vai onko sillä yhteys valvontasatelliitteihin tai johonkin?”

“Tietääkseni ei ole.”

“No, ilmeisesti lähdemme kävelemään?”

“Ei tässä muutakaan voi.”

Kun asuntoauto pysähtyy korjaamon pihaan, on Nathan la-
kannut hengittämästä. Jeff taistelee itsesyytöksiä vastaan,
mutta tietää toteuttaneensa ystävänsä toiveet: kun tämä
alkoi käydä kalpeammaksi ja sekavammaksi, tämä kuiten-
kin toisti nimenomaisesti ja useaan kertaan, ettei halua
sairaalaan. Ettei aikaa olisi kuitenkaan.

“Vain tehtävällä on merkitystä,” olivat hänen viimeiset
sanansa ennen kuin hän menetti tajuntansa.

Jeff:iä puistattaa, kun hän ajattelee, että lentolaite olisi
hyvinkin saattanut viiltää myös häntä, jos he olisivat viipy-
neet asunnossa sekunninkin kauemmin. Hän haluaisi tie-
tää, mitä terissä ollut myrkkä tarkalleen oli, mutta ymmär-
tää kyllä, että juuri nyt se on toissijaista.

Suunnitelmien tekeminen Throne Group:in robottien ja
maailmanlopun pysäyttämiseksi on ensisijaista.

Muut eivät ole vielä saapuneet. Jeff hengittää syvään ja
raskaasti, ja yrittää rauhoittua.

Kuluu ehkä seitsemän minuuttia.

Micha saapuu moottoripyörällään. Se sotahullu. Jeff ei
olisi välittänyt nähdä juuri häntä ensimmäiseksi.

Rashid on seuraava. Hänestä Jeff:illä ei ole pahaa sanot-
tavaa. Vakavamielinen hän on kyllä, mutta se oikeastaan
sopii hyvin, kun on kyse maailmanlopusta.

Rutger ja Emily saapuvat yhtä matkaa.

Orzo tulee paikalle viimeisenä. Hän on ehkä porukan
oudoin. Nytkin hänellä on päässään jonkinlainen meriros-

vohattu.

Kaikki Skeptikot ovat koolla. Kuusi elävää, ja yksi kuollut.

Jeff nousee autosta. Hänen jalkansa tuntuvat raskailta, eikä hän oikein tiedä, mitä sanoisi.

“Meillä on tilanne,” hän aloittaa viimein. “Kävimme yhdessä asunnossa. Nathan kuoli sen seurauksena.”

Skeptikot nyökyttelevät vakavina. Paitsi Michasta ei oikein tiedä. Ilme saattaisi olla jonkinlainen ilkikurinen virne. Jeff yrittää työntää sen mielestään. Nyt joka tapauksessa tarvitaan jokaikinen.

“Siis tyyppin kaksi skenaario?” Rashid varmistaa. Nathanin tekstiviesti luultavasti koostui pelkästä numerosta, joka kertoi periaatteessa kaiken. “Minkälaisia robotteja?”

“Kohtasimme pienen lentävän laitteen,” Jeff selittää. “Se myrkytti Nathanin. Mutta kuvittelisi, että niitä on muunkinlaisia. Täytyy kaivaa esiin se alkuperäinen foorumiposti.”

Jeff hakee läppärin asuntoauton takaosasta. Se oli Nathanin oma, ja tuntuu nyt joltain pyhältä muistoesineeltä. Saa nähdä, onko Throne-ryhmä jo onnistunut siivoamaan alkuperäisen kirjoituksen AboveTopSecret-sivustolta. Se varsinkin todistaisi ryhmän pelottavan vaikutusvallan.

Rutger avaa vanhan munalukon, laittaa valot päälle, ja ryhmä siirtyy korjaamoon sisälle. Jeff asettaa kannettavan hallin pitkälle takapöydälle, kytkee sen dataprojektoriin, ja käynnistää koneen lepotilasta.

“Kävimme siis erään AboveTopSecret-sivustolle kirjoittaneen henkilön asunnossa. Hänet murhattiin eilen. Odottaa hetki.”

Heti koneen löydettyä verkkoyhteyden hän kokeilee AboveTopSecret-sivua. Se ei vastaa enää lainkaan. Kylmät väreet kulkevat hänen lävitseen. Jeff klikkaa auki koneen kovalevylle talletetun kopion tekstistä. He kaikki asettuvat istumaan ja alkavat lukea.

“Tervehdys. Minun nimeni on Jan. On luultavaa, että tämän kirjoituksen seurauksena minulle tapahtuu jotain,

mutta en voi vaieta enää kauempaa. Työskentelen salaisessa ryhmittymässä nimeltä Throne Group. Sen varallisuus on jotain käsittämätöntä, kuten myös teknologia, jota se kehittää. Aikomuksemme oli hyvä: ihmiskunnan jatkettu hengissäsäilyminen Maan elinkelvottomaksi muuttumisen jälkeenkin, mutta ratkaisumalli, johon päädyimme, sai aikaan jotain kamalaa. Robotteja, joilla on tietoisuus, ja jotka riistäytyivät hallinnastamme. Kyllä! Klassinen skenaario, josta C-luokan tieteis- ja toimintaelokuvat ovat varoittaneet meitä pitkään, on nyt täyttä totta. Ja ryhmämme johto, jonka mielenterveyttä epäilen, luultavasti halusikin juuri näin tapahtuvan...”

Kaikkien luettua viestin Jeff ottaa esille paperin, jonka Nathan löysi asunnolta.

“Tämä dokumentti kuvaa yhtä Throne Group:in laitokista. Siinä on myös koordinaatit. Luultavasti meillä ei oikein ole vaihtoehtoa, kuin mennä sinne, ja potkia persettä,” hän sanoo, mutta ei pidä esitystavastaan: hänen äänensä värisee.

“Mutta tietysti on jokaisen oma päätös, lähteekö mukaan. Tämä reissu tulee aika varmasti edellyttämään lain rikkomista, ja siitä ei ehkä selviä hengissä. Tässä vaiheessa kaikki ne, jotka eivät halua osallistua, voivat poistua. Se on meidän kaikkien turvaksi.”

Kukaan ei poistu.

“War,” Micha sihisee. “Fucking war.”

“Vain kaksi kysymystä,” Orzo sanoo. “Milloin lähdemme, ja onko panoksia riittävästi?”

“Katso. Tuolla on valot päällä, ja paljon autoja,” Kim sanoo osoittaen muuten rappeutuneen näköisen rakennuksen suuntaan. He ovat kävelleet noin kahdenkymmenen minuutin ajan läpi aamuöisen haja-asutusalueen. Minkäänlaisia elonmerkkejä on ollut vähän.

“Olisikohan se joku jengi,” Amos vastaa. “Jos ajattelit ottaa yhden noista autoista, varaudu siihen, että ne tulevat päälle joukolla. Ja niillä voi olla aseita. Meillä ei. Tosin siihen, mitä lentokentällä ehkä on vastassa, olisi hyvä, jos meillä olisi kättä pidempää.”

“Ei kun minä ajattelin tervehtiä niitä. Haluan nähdä niiden reaktiot.”

“Hullu! Et sinä saa näyttäytyä ihmisten ilmoilla. Tämä juttu täytyy pysyä tiukasti salaisena. Muuten tulisi massapaniikki, jos tieto leviäisi. Paitsi...”

Kim ei välitä tiedemiehen epätoivoisesta agitaatiosta. Tai ehkä se huvittaa häntä vähän.

“Paitsi mitä?”

“Yksi meistä ei kestänyt henkistä painetta. Hän kirjoitti salaliittohörhöjen internet-foorumille, paljasti kaiken mitä tiesi. Tosin hän ei tiennyt Jörmungandr:ista. Hänet eliminoitiin, ilmeisesti Normanin toimesta.”

“No niin. Eihän sillä sitten ole väliä, jos saatetaan tietoisuuteen asian kääntöpuolikin. Tässä on kyberneettinen maailman pelastaja, jonka Throne-ääliöt ovat leikelleet kaasaan vastoin tämän tahtoa. Saadaan hyvät otsikot roska-

lehtiin, juuri ennen kuin posahtaa.”

“Tämä ei helvetti ole leikin asia!”

“Haista sinä paska ja ole hiljaa. Minä päätän, että menen kuokkimaan tuohon kokoontumiseen juuri nyt.”

Amos vaikenee, mutta Kim aistii tämän kytevän raivon. He lähestyvät rakennusta. Kim näkee ison ruostuneen kyltin SMALL MACHINERY REPAIR SHOP, johon on kirjoitettu spraymaalilla päälle:

SKEPTICS – PREPARING FOR THE END

<http://www.skepticsapocalypse.org>

“Mikä ihmeen skeptikkojen maailmanloppu,” Kim ihmettelee ääneen. Sitten hän muistaa. Sehän on Agent Steel -yhtyeen ensimmäisen levyn nimi. Tosin hän ei itse harrasta speed metallia. Se ei vain ole riittävän synkkää.

Hän havaitsee punaiseksi maalatun teräsoven, jossa ei näytä olevan lukkoa.

Ovi aukeaa vaivatta. Hän astuu sisään yrittämättä mitenkään olla herättämättä huomiota.

Kuusihenkinen ryhmä tuijottaa häntä suut auki.

“Ettekö ole ennen nähneet kyborgia?” Kim katkaisee hiljaisuuden.

Kim tarkastelee lähinnä huvittuneena sivustakatsojana, kun jonkinlainen yhteisymmärrys syntyy. Ensin Skeptikot pitivät häntä jonkinlaisena kyborgipukuisena teeskentelijänä, mutta hänen murskattuaan teräksisen roskasangon ilman mitään suurempia vaikeuksia ääni kellossa muuttui.

Vaikka uusien ihmisten kohtaaminen on pohjimmiltaan epämiellyttävää, hän joutuu myöntämään, että ryhmän kohtaaminen saattoi olla kriittinen, merkittävä käänne, niin uskomattomalta yhteensattumalta kun se vaikuttikin. Jos ei muuta, niin tämä tarkoittaa, ettei Throne Group:in lentokentälle tarvitse mennä ilman aseita. Varovainen täytyy silti olla, ettei joudu virkavallan kanssa tekemisiin matkalla.

Amos joutuu melko korventavaan ristikuulusteluun, ja se onkin ihan oikein. Onhan kyseessä ns. vihollisen edustaja. Erytystä kiinnostusta herättää Skeptikkojen kohtaaman miniatyyrilentorobotin myrkky: Amos ei suoralta käsin osaa selittää, mitä se on, mutta uskoo, että Throne-ryhmän laitoksista löytyy siihen vastamyrkkyä.

Matkaan ollaan lähdössä 24-henkisesti: ei mitään turhaa nukkumista välissä. Koska kello tikittää sekä Jörmungandrin että fuusiokennon latauksen muodossa, tämä lie-nee suhteellisen viisasta. Kim uskoo, ettei itse saisikaan unta. Ehkä hän ei enää kyborgina tarvitsekaan sitä?

“Miltä se tuntuu, kun on nuo – transplantit?” Emily kysyy.

Kim ei mitenkään jaksaisi vastata.

“Niin no. Ei niin erikoiselta. On kai se jonkinlainen edistysaskel. Mutta vituttaa ihan niinkuin ennenkin, ja ehkä enemmänkin, jos miettii, että on pakko saada virtaa, että pysyy hengissä.”

Ajojärjestely on seuraava: Orzo, Kim ja Amos kulkevat etummaisena Orzon tila-autolla, ja loput Skeptikot tulevat asuntoautolla perässä. Puoliautomaattisista pistooleista ja kivääreistä (suhteellisen mitäänsanomattomia kaikki, paitsi Michan .50AE Desert Eagle), revolvereista ja haulikoista koostuva ase-arsenaali on jaettu suhteellisen tasaisesti autoihin.

Kim saa onneksi istua takapenkillä täysin rauhassa. Orzon jutut ovat kummallisia, mutta ne eivät kuulu kovin selkeästi moottorin ja renkaiden jyrinän yli.

Matka kestää, kuten Kim arvelikin, noin kaksi tuntia.

Ei ongelmia. Ei virkavaltaa.

He pysähtyvät hyvissä ajoin ennen yksityislentokentän aitaa ja porttia, ja nousevat ulos pikaneuvonpitoa varten.

“Kuten sanoin, on viisainta, että ensin vain minä ja Kim näyttydymme,” Amos toteaa. “Jos henkilökunta on kirurgiatimille myötämielistä, ei ongelmia pitäisi olla. Meillä on oma koodisanasteemi. Jos taas paikalle on vaihdettu Normanin lakeijat, silloin pitää iskeä nopeasti. Jos kaikki on OK, annan teille merkin taskulampulla. Mutta jos sitä ei tule viidessätoista sekunnissa, ajakaa täysiä läpi portista, ja valmistautukaa ampumaan kaikkea mikä liikkuu.”

Amos:illa on 9-millinen puoliautomaatti, mikä ei tunnu oikein riittävältä, mutta hänellä ei muutenkaan taida olla kokemusta aseista. Kim:illä itsellään on pumppuhaulikko Skeptikoilta lainatun pitkän takin alla. Hän tuntee kylmää

huvitusta, kun muistelee Terminator-elokuvien “hyvän” Terminaattorin “0.0% ihmisuhreja” -tilanneanalyysiä. Itse hän ei tule vastaavaa periaatetta noudattamaan, ei suinkaan! 100.0% ihmisuhreja olisi hyvä tavoite, johon pyrkiä.

He kävelevät valaistulle portille, missä on valvontakamera ja sisäpuhelimen paneeli. Amos painaa puhelimen nappia. Kaiuttimen rahina kertoo, että yhteys on auki.

“Koodisana: flippaava ninja. Vastakoodi?” kuuluu kaiuttimesta hetken päästä.

“Todellinen ultimaalinen voima,” Amos sanoo mikrofooniin selvästi artikuloiden. Kim lähes naurahtaa ääneen: idea näyttää olevan, etteivät pahat Normanin kätyrit voisi mitenkään käyttää noin lapsellisia tunnussanoja.

Kuluu pari kiduttavan pitkää sekuntia.

“Tulkaa sisään,” kuuluu vastaus lopulta.

Amos näyttää taakseen valoa.

“Meitä on lisää. Mutta olemme kaikki samalla puolella,” hän tarkentaa puhelimeen.

“Jep, sisään vaan kaikki.”

Portti aukeaa sähköisesti, ja kauempaa takaa kuuluu asuntoauton moottorin käynnistyminen. Loput Skeptikot ovat tulossa esiin.

Kim ja Amos kävelevät portista sisään itse lentokentän alueelle.

“Jan keksi nuo tunnussanat. Ennen hän olikin aina luonteeltaan ilkikurinen.”

“Oli?”

“Niin. Hän –”

Äkkiä Amos:in ilme muuttuu kauhistuneeksi. Hän yrittää viittoa taakseen, mitä ilmiselvimmän varoittaakseen, peruakseen valomerkinsä.

“Hän kirjoitti sen foorumiviestin. Hänet luultavasti pakotettiin kertomaan sanat ennen kuolemaansa. Helvetti kun olin tyhmä. Tämä on takuulla ansa,” Amos puhuu häteisesti.

He ovat melko hyviä maalitauluja lentokentän asfaltilla. Asuntoauto on vielä takana parinkymmenen metrin pääs-

sä.

Kim tyrkkää miehen kumoon, ja seuraa itse perässä. “No mene sitten jumalauta edes matalaksi!”

Samassa lentokentän rakennuksista avataan tuli useilla sarjatuliaseilla. Eri ampujia on ainakin viisi, Kim arvaa. Hän toivoisi nyt, että hänelle olisi asennettu jokin Terminaattorityylinen HUD-systeemi, joka kertoisi kaikki kohteet, mutta turha toivo. Hänellä on vain teräsraajansa, ja haulikko.

Haulikko on itse asiassa hyvin huono ase noin kaukaisia kohteita vastaan.

Nyt asuntoauto on kohdalla, ja ovet avataan. Jeff pysyy vielä sisällä kuskin paikalle kyyristyneenä. Micha tulee ensimmäisenä ulos kiikarikiväärin kanssa, asettuen kuskin oven taakse suojaan. Kim itse asiassa kuvitteli, että tuo kaljupää haluaisi ennemmin taistella Desert Eagle:nsa avustuksella, mutta ilmeisesti rationaalisuus voitti tässä tapauksessa.

“Saatiin ainakin sota. Armageddonista en ole vielä varma,” Micha rähisee, ja tähtää.

Kim yrittää paikantaa viholliset. Lennonjohtotornissa on ainakin yksi. Micha ikäänkuin lukee hänen ajatuksensa, ja tähtää korkealle.

Kivääri puhuu.

“Perkele, ohi.”

“Se ei ole niinkuin peleissä,” sanoo Rutger, joka on myös tullut ulos toisen kiväärin kanssa.

Kim haluaisi myös tehdä jotain, ei sen takia, että haluaisi auttaa, mutta siksi, ettei osoittaisi heikkoutta. Amos ei ole briefannut häntä kunnolla, minkälaista vahinkoa metalliraajat kestävät, ja siksi hän nyt ainakin toistaiseksi osoittaa häpeällistä varovaisuutta. Rintakehässään hän ei havainnut minkäänlaisia modifikaatioita – jokin panssari olisi kyllä enemmän kuin suotava. Ehkä operaatio jäi kesken robottien hyökätessä? Skeptikoiltakaan ei löytynyt mitään suojaliiviä.

Tulitus on taukoamatonta. Varmaankin laatikko- tai

vyösyöttöisiä konekiväärejä. On vain ajan kysymys, milloin johonkin heistä osuu.

Rutger ampuu, ja tornista käsin tuleva tulitus vaimenee. "Yksi poissa pelistä," hän toteaa kylmästi.

Mutta neljä jäljellä. Se on vielä aivan liikaa: jotain on tehtävä. Rashid alkaa ampuu revolverillaan, mutta matka on ehkä liian pitkä. Silloin Kim saa idean. Hän kiertää asuntoauton takaosalle, ja repii sen irti saranoiltaan. Se ei ole kovin vahvaa tekoa, mutta parempi kuin ei mikään.

Tämän "mellakkakilven" suojassa hän lähtee mielipuoliseen juoksuun kohti lähintä rakennusta.

Kuulostaa kohtalaisen ilkeältä, kun luodit repivät ovea lähietäisyydeltä hänen korviinsa nähden. Sitten myös hänen käsivarteensa osuu, mutta näyttää, ettei se kärsi lähes minkäänlaista vahinkoa.

Jotain Throne-ryhmä on sentään tehnyt oikein!

Kivääritulitus asuntoautolta jatkuu. Ei voi tietää, osuvatko he mihinkään, ja osuuko heihin.

Mutta nyt Kim on rakennuksen kohdalla. Hän etenee pitkin seinää avoimelle ikkunalle, josta yhtä konekiväärisarjaa ammutaan, ottaa haulikon esiin, ja ampuu, lataa ja ampuu uudestaan. Suuliekkien valossa hän näkee jotain tarkemmin määrittelemätöntä punaista mössöä lentävän, mutta mikä tärkeintä, konekivääri vaikenee.

Kolme ehkä jäljellä.

Yksi on ainakin vastapäisessä rakennuksessa, mutta sinne päästäkseen on kuljettava noin viidentoista metrin suojaamaton matka. Ja nyt hän ei tarkkaan ottaen tiedä, missä loput ampujat ovat. He ovat sitä paitsi saattaneet siirtyä.

"Do you want to live forever?" hän psyykkaa itseään ääneen.

Ei hän halua. Hän ampailee juoksuun, syöksyy avoimelle kentälle, pyörähtää ympäri, ampuu summittaisesti ja lataa jälleen. Tulitus kohdistuu häneen välittömästi.

Jalkaan osuu, ilmeisesti useampikin luoti, mutta se ei tunnu juuri haittaavan menoa. Hän on taas jaloillaan, ja

pääsee seinän suojaan. Mutta tällä välin ampuja on ehtinyt kadota ikkunasta. On mentävä sisälle.

Kim ampuu rikki lähimmän oven lukon, ja astuu sisään. On hämärää. Jos vastustajat edustavat tämän Norman-ääliön henkilökohtaisia spetsnaz-joukkoja, heillä saattaa hyvin olla pimeäkiikarit.

Ja hänellä ei ole mitään vastaavaa. Reilua.

Kim etenee varovaisesti käytävää pitkin, valmiina pumppaamaan kenen tahansa eteen tulevan täyteen lyijyä. Edessä hän havaitsee lupaavalta näyttävän raollaan olevan oven.

Sinne siis.

Hän valitsee shokkisisäntulon, ja potkaisee oven sisään. Hän yliarvioi hieman tarvittavan voiman, ja lähes menettää tasapainonsa. Ovi hajoaa kappaleiksi, yllättäen myös sisällä odottavan kommandon, tai miksi häntä sanoisi.

Kim ei jää odottamaan, että tämä kääntäisi konekiväärensä piipun häntä kohti, vaan tähtää suoraan miehen päähän, jonka futuristinen valonvahvistinlaseilla varustettu kypärä peittää.

Lähietäisyydeltä sen ei pitäisi suojata riittävästi.

Haulikko puhuu, ja pää muuttaa oleellisesti muotoaan, harmaan ja punaisen massan ejaktoituessa takaa. Mies seisoo vielä hetken jaloillaan, ja rojahtaa sitten maahan.

Oikeastaan on sääli, että kypärä ja pimeäkiikarit tuhoutuivat, mutta Kim toteaa tehneensä silti oikean valinnan: haulit eivät olisi mitenkään tehneet välittömästi kuolettavaa vahinkoa järeän panssariliivin läpi, ja tällöin mies olisi ehtinyt ampua.

Kim tuntee selkeää voitonriemua varusteista, jotka voi nyt ryöstää itselleen. Tämähän menee hieman kuin peleissä.

Hän kiroilee liivien kiinnikkeiden kanssa, mutta saa ne lopulta pois kommandon päältä, ja puettua itselleen. Konekivääri tuntuu juuri sopivan painoiselta hänen käsissään. Kim myöntää häpeäkseen, ettei tiedä tarkkaa mallia, mutta

ainakaan Rambon käyttämä M-60 se ei ole. Epätarkkahan se tietysti on sarjatuella, mutta määrä tulee korvaamaan laadun.

Kim kuulee auton ääntä. Skeptikot ovat ilmeisesti etenemässä. Sääliä ei ole: hän olisi nyt kyllä ehdottomasti ratkaisemassa koko taistelun seuraavien hetkien aikana, mutta jos nuo toiset haluavat välttämättä riskeerata tai jopa tapattaa itsensä, ei hän aio tehdä paljon heitä estääkseen.

Hän kiertää ulos samaa reittiä kuin tuli, ja katsoo nurkan takaa: kaksi ampujaa vielä, pitkänomaisen hangaarin katolla. Skeptikkojen asuntoauto on aivan hänen vieressä: Micha ampuu tiheään tahtiin naama mielipuolisessa irvistyksessä, mutta Rutgeria ei näy. Loput ovat suojassa auton takana osallistumatta taisteluun oleellisesti.

Tämä päätetään nyt.

Kim astuu esiin, nostaa konekiväärin olkapäätään vasten, ja avaa tulen. Hän havaitsee teräskäsien pystyvän hallitsemaan rekyyliä erinomaisen hyvin, ja suuntaa pyyhkivän tulisuihkun hangaarin katolle.

Ehkä noin viisikymmentä luotia, ja kaksi tappoa.

Ei kovin hyvä hyötysuhde, mutta riittävä tässä tapauksessa. Tulee hiljaista.

Sitten Kim huomaa kauempana makaavan liikkumattoman hahmon. Rutger. Tämä ilmeisesti kuoli tulituksessa.

“Nyt täytyy kaapata kone äkkiä, ennen kuin tulee lisäjoukkoja,” Amos toteaa hengästyneellä äänellä. Hänen takkinsa on värjäytynyt punaiseksi, tosin vain käsivarresta. Jos pilotteja ei ole tässä joukossa kuin yksi, hänen on paras olla lentokunnossa.

“Ne pirut rei’ittivät hattuni,” Orzo toteaa epäuskoisesti.

Emily itkee selvästi, ja menee elottoman Rutgerin luokse. Tämä oli ilmeisesti hänelle tärkeä, Kim observoi kliinisesti. Amos näkyy ymmärtävän, ettei voi vauhdittaa tätä tilannetta, mutta on selvästi hermostunut ja raivoa tihkuva.

Lopulta herkkä hetki on ohi. Rutger on pakko jättää tyylysti asfaltille. Micha kiertää sisälle rakennukseen, josta Kim eliminoi ensimmäisen ampujan, kun muut suuntaavat

hangaaria kohti. Hän tulee pian esiin toinen konekivääri mukanaan, ja ottaa ryhmän kiinni.

Noita ei koskaan ole liikaa, Kim ajattelee.

Kirkkaasti valaistu hangaari sisältää pari teknikkoa haalareissaan. He eivät tee aseistettua vastarintaa, joten heidät vain sidotaan ja suukapuloidaan, vaikka Kim ei olisi vastustanut tappamistakaan.

Kone, jolla ryhmän on tarkoitus poistua, muistuttaa osittain perinteistä Harrier-VTOL-suihkukonetta, mutta on silti sulavalinjaisempi, vaikka sisältääkin rahtitilan, johon he kaikki tulevat helposti mahtumaan. Amos kääntää seinällä olevaa katkaisijaa, joka saa hangaarin isot liukuovet avautumaan, ja kiipeää sitten koneen ohjaamoon sitä käynnistämään, samalla kun muut pysyvät vartiossa siltä varalta, että kommandoja ilmestyisi paikalle enemmän.

Sitten on aika nousta kyytiin, ja toivoa että Amos osaa lentää. Eikä menetä tajuntaansa verenhukasta kesken kaiken (haava on jo sidottu, mutta se ei välttämättä pidä.)

“Niin, siis en ole lentänyt suihkukonetta ennen, mutta tässä on melko hyvä automatiikka,” hän selventää rohkaisevasti.

Suihkumoottorit käynnistyvät. Amos rullaa koneen ulos kiitoradalle. Sitten mennään.

KOLMAS KIRJA - ILMA

KAI – Vaaran aavistaminen

Aika: 08:27

Paikka: Lähestymässä Throne-ryhmän laitosta

Fuusiokennon lataus: 53 %

Kim miettii vastaanottokomiteaa, jonka perillä täytyy väistämättä odottaa. No, ainakin nyt he ovat paremmin aseistettuja, kuin matkalle lähtiessä.

Hän on tyytyväinen paskanjauhannan puutteesta heidän istuessaan rahtitilassa suhteellisen ääneti. Oikeat sotilaat takuulla jauhaisivat paskaa ja äijäjuttuja loputtomiin!

Ja minä mietin ambient occlusionia, ihon valaistusta ja varjotekniikoita, joita tässä cutscenessä tulisi käyttää. Alan ilmeisesti tulla tajuihini.

Harmi. Tarina olisi vielä kesken.

Yritän vähitellen hyväksyä paluuni todellisuuden. Ei ole kyllä mitään käsitystä, mikä päivä on nyt. Mutta tuntuu sen verran leijuvalta ja muuallaolevalta, vaikka makaankin aloillani sairaalan sängyssä, että ihan vähän aikaa ei ole voinut kulua.

Muistan suhteellisen elävästi fysiikkakollision, joka saattoi minut tähän. Mitähän osia minusta on rikki? Mutta se on hyvin toissijaista.

Minulla on paljon huolestuttavampaakin mietittävää.

Jörmungandr! Perkele, sen tekemisestä on taatusti mennyt paljon aikaa hukkaan.

Vision3D! Tällä välin siinä on taatusti tapahtunut jotain sanoinkuvaamattoman kauheaa.

Mutta nyt minulla on silti jotain, jota minulla ei ollut ennen onnettomuutta. Juoni ensimmäiseen peliin, joka Jör-mungandr:illa tullaan toteuttamaan!

Okei, myönnetään, se luultavasti kaipaa paljon säätämistä, ennen kuin sopii peliksi. Mutta skenaario on pohjimmiltaan ehdottoman käyttökelpoinen. (Enhän minä siis oikeasti tiedä varsinaisten AAA-pelituotantojen toteuttamisesta mitään. Engineistä vaan. Enkä oikeastaan haluaisikaan ohjata esim. 20-päistä, tai vieläkin isompaa artittimiä. Hyi kamala, jos ne kaikki säätäisivät graafipohjaisella materiaalieditorilla generoituja shadereita. Siinä framera-te taatusti tippuisi.)

Minuun ei oikeastaan satu, mikä on positiivista.

Mutta epäilen, että kohta saattaa sattua. Siis henkisesti. Jos/kun saan tietää, mitä töissä on tapahtunut.

En sentään ole niin hullunrohkea, että lähtisin sängystä itsekseni liikkeelle, vaan odotan, että hoitaja tai lääkäri tulee. Sillä välin yritän päästä takaisin kehittämäni maailmaan, tai vaikka irtautumaan ruumiista, mutta kumpikaan ei onnistu. Yhteys on katkaistu. Harmi.

Ajan uudelleenkalibrointi: Päivä 32.

Lopulta saan lääkärin audienssin, ja asiat selvenevät hie-
man. On siis kulunut kahdeksan päivää. Seurauksena pään
impaktista staattiseen kenttägeometriaan minulla oli aivo-
tärähdys, johon liittyi turvotusta, ja tästä syystä minua pi-
dettiin tarkoituksellisesti koomassa. Ilmeisesti haluttiin
pelata varman päälle, ja vähempikin olisi riittänyt. Ruhjei-
ta ja oikean olkapään sijoiltaanmenoa (jonka pitäisi jo olla
käytännössä parantunut) lukuunottamatta muuta vaka-
vampaa vahinkoa muissa ruumiinosissani ei ole.

Minulle tehdään neurologisia testejä, jotka ovat oikeas-
taan aika turhia. Kyllähän minä tiedän olevani kunnossa,
jos kerran olen jo käsikirjoittanut ensimmäisen Jörmun-
gandr:illa toteutettavan pelin, ja tärkein huolenaiheeni on
menetetty devausaika.

Vaikka pysyvää aivojen kudosvauriota ei ole havaitta-
vissa, jotain epästabiiliutta reaktioissani ja käyrissä kuu-
lemma on, minkä seurauksena minun täytyy pysyä täällä
tarkkailtavana ainakin muutama päivä. No helvetti.

Seuraavaksi täytyy tehdä inventaarion tarkistus. Sisäl-
tääkö se yhden hyvin spesifisen ja tärkeän esineen.

En saisi vielä nousta sängystä, mutta arvatkaa, tottelen-
ko. Heti lääkärin lähdettyä menen kaapille, jonka havait-
sen ilokseni sisältävän vaatteeni. Hieman on disorientoitu-
nut olo, mutta kaapista saa hyvin tukea. No niin.

Saatana!

Ei muistitikkuja!

Tutkin läpi kaikki mahdolliset ja mahdottomat taskut, mutta tämä on väistämätön ja kamala totuus. Koska päivänä ennen onnettomuutta en koodannut töissä Jörmungandr:ia ollenkaan, en menettänyt ohjelmoinnistani mitään. Mutta ellei tikku sitten disintegroitunut törmäyksessä (mitä en usko, koska vaatteenikaan, tai minä itse eivät disintegroituneet), on koodi somewhere out there, in the wild.

Paranoiani kasvaa. Kenellähän se on nyt hallussaan, jos kenelläkään? Olisi inhottavaa, jos se esim. vuotaisi nettiin ennen aikojaan. Tosin kuinka moni osaisi ajatella kadulta löytämästään muistitikusta, että hahaa: tässä on val-lankumouksellisen (tosin helvetin keskeneräisen) enginen koodi. Julkaisenpa sen netissä jollain random-nimellä ihan vain vittuillakseni!

Lannistuneena palaan sänkyyn.

Kohtuullisen pian tästä siskoni Emilia tulee käymään. Emme ole erityisen läheiset, mutta kai hän on ilahtunut, että olen palannut tajuihini. Annan hänelle kämppäni avaimen ja questin hakea kannettavani. Tähän sisältyy kyllä riskejä: mahdollinen Jörmungandr:in vuotaminen, versionhallintapalvelimen kompromisoituminen, ware-arkistoni paljastuminen, mutta ne nyt on vain pakko kestää. Koska muuten aikaa menisi taas hukkaan, ja se nyt olisi aivan kestävämpää.

Parin tunnin päästä saan koneen ja avaimen takaisin. Kiitän kylmän asiallisesti. En usko näkeväni Emiliaa tämän jälkeen vähään aikaan.

Nyt koodataan!

Hoitajat paheksuvat toimintaani, ja uhkaavat takavarikoida koneen, jos käytän sitä liikaa. Täytyy siis suurimaksi osaksi koodata salaa. Havaitsen kuitenkin, että kohtuullisen lyhyenkin koodaussession jälkeen käsivarsiani alkaa kylmätä ja nipistellä, ja sitten ne puutuvat. Enkö muka siis olekaan vielä aivan kunnossa? Lisäksi huomaa

tuottavani UDB:tä hälyttävällä tahdilla, kun esim. nullpointer-tarkistus tulee if-lauseessa sen jälkeen, kun osoitinta on jo dereferensoitu. Ajatukset eivät meinaa pysyä kasassa tosiohjelmoinnin vaatimalla tasolla.

Perkele.

Lisäksi, koska verkkoyhteyttä ei huoneessani ole, en pääse commitoimaan. Nyt, jos kannettavalle sattuisi jotain, saattaisi suututtaa ja oikein kunnolla. Niin, että voisi ottaa kyynärsauvan ja huitoa sillä kaikkea elollista, kunnes mikään ei olisi enää elollista!

Saan kuitenkin koodattua Jörmungandr:in skenemallia eteenpäin: konkreettisen Scene-luokan toteuttamisen, jolla on paikkansa syklisessä rakenteessa, ja joka tietää mm. renderöinti-, octree- ja okkluusioalijärjestelmien olemassaolosta. Lasken, että jaksan koodata kokonaisuudessaan ehkä kolmen tunnin ajan, sitten rupeaa väsyttämään liikaa.

Kukaan töistä ei ota minuun tänään vielä yhteyttä. Hyvä niin, sillä tieto kuitenkin lisäisi tuskaa.

Päivä 33.

Koodaustehokkuus (arvioitu): 35%

Tuntuu, että tänään ehkä jaksaisin paremmin. Mutta ongelmana on, etten tiedä mitä tehdä seuraavaksi. Tehtävää on liikaa, ja ajatukseni ovat vieläkin jotenkin sekaisin. Ne leijuvat jollain ylemmällä tasolla kuin pitäisi, ja vieläpä pyörivät kehää kuten Jörmungandr:in arkkitehtuuri!

Mutta toisin kuin engineeni runtime-rakenne, jolla on selvä initialisaatio- ja tuhoamisjärjestys, eivät ne suostu asettumaan järjestykseen. Konstruktoreita on selvästi kutsuttu väärässä järjestyksessä! Nyt pitäisi vain äkkiä kutsua delete:ä kaikelle.

Palautan backlogia mieleeni, tai pikemminkin juttuja, jotka olen jo toteuttanut:

- *Kehämäinen perusrakenne: initialisointi ja shutdown.*
- *Direct3D10-renderöintijärjestelmä.*
- *Materiaalisysteemi: shaderien dynaaminen kääntämien fragmenteista, surface shader -systeemi.*
- *Octree-toteutus ja frustum culling.*
- *CPU:lla renderöitävä occlusion culling.*
- *Varjorenderöinti (dynaamisesti indeksoituja hardware shadow map:eja.)*
- *Scenenodet, scenemalli ja serialisointi.*

Kommentointia tietysti voisi harrastaa, jos kuvittelee, että engineä joskus käyttäisivät jotkut muutkin kuin minä. Mutta tässä vaiheessa se olisi silti heikkoutta, ja tuhlaisi aikaa, jonka voisi käyttää koodin varsinaiseen edistämiseen.

Kustomoitu muistinhallinta, jonka jo kerran siirsin myöhemmäksi, on jatkon kannalta ehdottoman tärkeä, mutta myös hyvin virhealtis osa-alue, jossa UDB on joko selkeää ja nopeasti ilmenevää, tai sitten tavattoman syväälle kaivautunutta ja salakavalaa. Niin vaativaa koodia ei pidä yrittää tehdä juuri nyt.

Input-järjestelmän voisi vaikka saattaa seuraavaksi hyväksyttävälle tasolle. Sehän olisi mukavan matalatasoista, mutta silti helppoa työtä, jota voi tehdä vaikka ajatukset eivät olisikaan ihan kasassa. Nykyinen järjestelmäni koostuu lähinnä hardkoodattuihin näppäinpainalluksiin reagoimisesta, ja hardkoodatuista kamerakontrolleista, jotka on yhdistetty hiiren liikkeisiin.

Kyllä. Tämä on päätetty.

Isompia alijärjestelmiä, jotka tulevat jatkossa, ovat mm. fysiikka ja verkkokoodi. Mutta niitä ennen kannattaisi kylä olla tuettuna animointi (hardware skinning, mitään muuta ei ole) ja muutamat alkeelliset, mutta tarvittavat scenenodetyypit, kuten skybox, billboardit partikkeliefekteille...

Input on oikeastaan siitä inhottava, että siitä seuraavana tulee heti mieleen, että mitä graafisen käyttöliittymän toiminnallisuudelle pitäisi tehdä. Koska kriteerini ovat tiukat, on selvää, että kirjoitan käyttöliittymäkoodin alusta käsin itse. Tähän voi helposti saada uppoamaan hyvinkin paljon aikaa, jos alkaa hienostella. Mutta kuten olen monesti muistuttanut itseäni: aikaa ei ole!

Yksi käyttöliittymän ensimmäisistä toiminnoista on pystyä tulostamaan ruudulle tekstiä. Tämäkään ei ole aivan triviaali toteuttaa, kun asettaa laatuvaatimuksensa aloittelijatasoa korkeammalle, ja kieltäytyy esim. D3DX-apufunktioiden käyttämisestä. Käytännössä silloin täytyy ottaa käyttöön apukirjasto, joka osaa ladata true type-

fontteja ja renderöidä ne bitmap:eksi kirjain kerrallaan. En minä nyt helvetti sentään true type-renderöintiä aio itse koodata! Siihenhän liittyy jotain iljettäviä patenttejakin.

Apukirjastoista FreeType2 on inhottavan bloatoitunut, mutta sitä näkee käytettävän paljon. Vaihtoehtona on esim. stb-truetype, mutta se puolestaan on hitaampi ja huonompilaatuinen. (Tosin ajonaikaisella nopeudellahan ei ole väliä, jos fontit renderöidään kuvatiedostoon etukäteen.)

Sen jälkeen, kun fontit on renderöity, ne täytyy vielä pakata GPU:lle ladattavaan tekstuuriin mahdollisimman optimaalisesti. Päätän pohjata operaation seuraavaan dokumenttiin, jonka olen kaukonäköisesti tallentanut tälle koneelle:

<http://clb.demon.fi/projects/rectangle-bin-packing>

Juuri kun olen syventymässä input-järjestelmän koodaukseen, kännykkäni soi. Sen näytön suojakuoreen on tullut rumannäköinen lommo, mutta toimiva se tuntuu silti olevan.

Jarmo.

Helvetin helvetti. Vastaan kuitenkin, sillä olisi heikkoutta pyrkiä pakenemaan vaanivaa todellisuutta pitempään, kuin on välttämätön tarve.

“Jarmo tässä, terve. Kuinkas siellä voidaan?”

Syvä hiljaisuus.

“Halusin vain soittaa muistuttaakseni, että Vision3D etenee täällä ihan omalla painollaan. Älä huolehdi siitä, vaan tulet sitten takaisin kun voit.”

“Selvä. Oliko muuta?”

Muutaman sekunnin hämmentynyt hiljaisuus.

“Öö... ei muuta. Hyvää paranemista.”

Jälkeenpäin kiroan tylyyttäni: olisi ollut edullista, jos Jens olisi tullut esiin hoitamaan tämän puhelun, mutta on pakko myöntää, ettei minulla olisi ollut henkistä vahvuutta tällaiseen subterfugeen.

Pian tämän jälkeen joudun kohtaamiseen lääkäriini kanssa, joka havaitsee kannettavani. Nyt teen jo tietoisien päätösten pelata systeemin ehtojen mukaan, jotta minulla on sitten vapaammat kädet suorittaa Jörmungandr:in de-vausta salassa. Vaikka inhottavaa tällainen pelleily ja teeskentely kyllä on. Muistaakseni lupasin itselleni, etten harrastaisi sellaista enää ikinä.

“Jaha. Ei kai täällä tehdä töitä?”

“Ei, tämä on ihan harrastustoimintaa.”

“Saanko kysyä, mitä?”

“Osallistuin National Novel Writing Monthiin. Siinä pitää kirjoittaa 50000 sanan kirja kuukaudessa. Kuukauden lopusta jäi sitten se muutama päivä väliin, niin en voi enää virallisesti voittaa, mutta tarina pitää silti saada loppuun.”

Hyi! Lineaarista fiktiota! Vastenmielistä! Jos kirjoittaisin jotain, olisi se ehdottomasti pelikäsikirjoitusta, johon haarautuva dialogi ja etenemisvaihtoehdot olisi jo suoraan kirjoitettu sekaan.

“Kuulostaa kyllä haastavalta. Älkää kuitenkaan rasittako itseänne liikaa. Kirjoitusasento ei ehkä ole tuossa paras mahdollinen. Tarkkailkaa, jos tulee kipeytymistä tai puutumista.”

No niin. Siitä selvittiin noin. Huomaisikohan lääkäri eron tekstinkäsittelyohjelman ja Visual Studion välillä? Sitä en aio testata, vaan pistän kyllä kannen röyhkeästi kiinni, jos joku katsoo.

Nyt saan luvan kävellä, kunhan en siinäkään rasita itseäni liikaa. Mutta täällä pysytään edelleen.

Input-koodia lähtee syntymään. Hieman hankalaa on, kun ei ole nettiä, eikä minulla tällä koneella ole täydellistä MSDN-dokumentaatiota, kuten varsinaisella työasemallani. Copy-pastean olemassaolevista vanhoista ohjelmistani tarpeen mukaan, vaikka tarkoitus oli olla tekemättä näin. Aika alkeellinen taskihan tämä on. Sujuu kohtalaisesti, vaikka vireystilassani on vielä paljon kohentamista.

Mutta perkeleen käsivarret puutuvat taas! Yritän kohtaa asentoa, mutta lopulta täytyy vain todeta, että on

pakko pitää taukoa. Alennun katsomaan TV:tä oleskelutilassa. Joudun tahtomattani alkeelliseen small talk:iin muiden potilaiden kanssa, enkä nyt halua vaikuttaa aivan kyllälevältä ja epänormaailta, joten pakottaudun kommunikoimaan jollain asteella.

Se olisikin, kun täällä olisi itseään parantamassa joukko todella kovan tason koodareita, jonkinlainen likainen tusina, ja sitten vaihdettaisiin sotatarinoita kieroutuneista tavoitteista, mystisistä virhetilanteista ja epärealistisista aikatauluista.

Mutta toiveajatteluahan tuollainen on.

Myöhemmin päivällä jatkan. No niin, nyt on toimiva ja hyvin abstrahoitu hiiri- ja näppäininputti, sekä event-pohjaisena että pollattavana. (En tee Jörmungandr:iin yleistä eventtijärjestelmää tai eventtiputkea, vaan kirjoitan templatoidun EventListener-toteutuksen, jonka avulla rekisteröidytään kuuntelemaan eksplisiittisestä lähteestä eksplisiittisesti määrättyjä eventtejä, esim. seuraavaan tapaan:)

```
class MySubsystem : public EventListener<InputEvent>
```

Siinä välissä samplaan taas vaihteeksi sairaalan ruokaa, ja täytyy todeta, että täällä haastetaso on aivan eri luokkaa, kuin Sodexolla. Varsinaisia jälkiruokia ei ole, mutta niitä ei tarvitsisikaan, kun jo pääruoat ovat näin haasteellisia. Oikeastaan nyt lipsutaan jo haastavuudesta varsinaiseen epämiellyttävyyteen.

Mietin, missä kuji-kiri -viilossa olen nyt menossa. Periaatteessa tämä on aika selvästi itsensä parantamista, mutta toisaalta myös seuraava viilto, eli vaaran aavistaminen, sopii tilanteeseen hyvin. Tarkemmin ajateltuna Jarmon sanat "Vision3D etenee täällä ihan omalla painollaan" tuntuivat hyvin uhkaavilta. Mitenkähän ne ovat siellä organisoinnit? Onko Jarmo nyt pääarkkitehti? Vai onko siihen tuotu joku ulkopuolinen tilalle? Olisiko Valtteri suostuteltu palaamaan?

Viimeksimainituun en ainakaan usko. Sehän tarkoittaisi

mahdollisesti, että olisi tapahtunut jotain hyvää, että Vision3D lähtisikin nousuun ja järkevöitymään poissaoloni myötä, mutta en usko että Vision3D:lle (tai minulle) tapahtuu ikinä mitään hyvää. Se vain sotisi karman lakeja vastaan.

Voisin tietysti soittaa Jarmolle ja kysyä, mutta se ei ole ollenkaan tapaistani. Tyydyn siihen, että annan pelkoni ja vihani liekkien palaa hiljaisena. Aikanaan tulen sitten näkemään.

Illalla vielä vähän koodausta. En pysty aloittamaan UI-koodia ja fonttirenderöintiä järkevästi ilman verkkoyhteyttä, joten päädyn kuitenkin kirjoittamaan custom-muistinhallintaa.

UDB:n määrä olkoon kääntäen verrannollinen paraneamiseni asteeseen!

Päivä 34.

Koodaustehokkuus (arvioitu): 45%

Oikeastaan on mukava olla vielä erillään muusta todellisuudesta. Perustarpeeni täytetään, ja saan koodata Jörmungandr:ia. Huonomminkin voisi olla.

Mutta koska en pääse käsiksi lähdemateriaaliin, enkä VTune-profileriin, on muistinhallinnan toteuttaminen hieinan hataralla pohjalla.

Siksi päätänkin aivan ensiksi kirjoittaa Jörmungandr:in sisäänrakennetun hierarkkisen suorituskykyprofilerin. Tyydyn ensin tekemään sen aivan C++:lla, mutta myöhemmin osittainen uudelleenkirjoittaminen assemblylla voitulla kyseeseen.

Tämä sentään onnistuu ulkomuistista ilman MSDN:ää.

```
LARGE_INTEGER frequency;  
if (QueryPerformanceFrequency (&frequency))  
    perfTimerFrequency = frequency.QuadPart;  
else  
    FATAL_EXCEPTION ("QueryPerformanceFrequency is  
not supported. Buy a new machine.");
```

sekä:

```
ProfilerBlock::begin()  
{
```

```

    LARGE_INTEGER counter;
    QueryPerformanceCounter(&counter);
    startTime = counter.QuadPart;
}

ProfilerBlock::end()
{
    LARGE_INTEGER counter;
    QueryPerformanceCounter (&counter);
    endTime = counter.QuadPart;
    long long usec = (endTime - startTime) * 1000000
LL / timerFrequency;
    // Check for weird shit when possibly switching from
one core to another
    if (usec < 0) usec = 0;
    elapsedLast = usec;
    elapsedTotal += usec;
    ++blockCount;
}

```

Havaitsen kirjoittaneeni kommenttirivin. Tämä on omassa koodissani todella, todella harvinaista. Olisikohan taustalla piilevä, vakavampikin aivovaurio? Kamalaa.

Itse performance counter:in lukeminenhan on helppo osuus. Sen sijaan se, miten profilerblockit saadaan varattua muistista mahdollisimman optimaalisesti ja cachekoherentisti (en halua tyytyä staattiseen rakenteeseen, vaan mielivaltaisia bloqueja täytyy pystyä luomaan koska tahan-sa), voi olla haastavaa. Omat muistinvarausrutiinit tulevat helpottamaan asiaa jonkin verran, mutta tästä tuleekin epämiellyttävä muna ja kana-ongelma: pitäisi olla profile-ri, jotta voi luotettavasti profiloida muistinhallintaa, ja jot-ta profilointi itsessään ei hidastaisi paljon ja vääristäisi mittauksia, pitäisi sen käyttää optimoitua muistinhallintaa! Toisin sanoen taas ilmenee kehä, jota voitaisiin kiertää loputtomiin.

Todellinen agile-hihhuli sanoisi, että yksi kierros on ite-

raatio, ja aina joka iteraation (sprintin) lopussa olisi shipattava sovellus. Täyttä paskaa, sanon minä! Ensimmäisten sprinttien tuotokset missään projektissa eivät taatusti ole shipattavia!

Vähitellen alkaa tuntua, että täältä olisi päästävää. Jörmungandr:in kehitys rajoittuu liiaksi. Olisin tietysti voinut pyytää Emiliaa tuomaan osan referenssi-opuksistani tänne, mutta ne ovat kuitenkin osittain vanhentuneita, ja olisivat jopa saattaneet aiheuttaa deprekoitujen funktioiden käyttöä Jörmungandr:ssa. Tämä ei tietystikään ole suotavaa. Koska Direct3D10:n myötä Windows Vista on minimi, ei ole mitään järkeä käyttää Win32 API-kutsuja, jotka toimivat myös esim. Windows 2000:ssa (tai vielä pahempaa, NT4.0:ssa), jos uudempi ja parempikin tapa on olemassa!

No niin. Jonkinlainen karkea ja initial implementation profilerista on olemassa. Pitää muistaa profiloida se jälkikäteen VTunella hyvinkin tarkasti.

Ennen kuin aloitan muistinhallinnan parissa, päätän ruiskia profilointimakroja liberaalisti ympäri Jörmungandr:in koodia. Performanssikriittisimmät polut on jo tutkittu, eikä yllätyksiä pitäisi tulla, mutta tuleepa samalla rasiustestattua profilointijärjestelmää.

Pitäisihän minun tietää. Aina kun kuvittelee asioiden menevän liian hyvin, paljastuu jotain yllättävää.

Täällähän on aivan hirvittävää scheisseä.

Kannatti todellakin profiloida. Esim. constant-buffereita allokoidaan ja vapautetaan lähes jatkuvasti. Miten minä olen päästänyt tällaista koodia sisään? Tavoitteenihan oli, ettei framen piirron aikana normaaliolosuhteissa kutsutaisi new:iä tai delete:ä lainkaan.

Voi saatanan saatana.

Tässä riittää savotaa.

Varsinkin kun en toimi vielä aivan täydellä teholla. Jos paljastuu, että esim. surface shader-järjestelmässä menee constanttien indeksointi tietyissä tilanteissa sekaisin, ja tämän seurauksena constantbuffereita luodaan tarpeettomasti uusiksi, on järjestelmä kuitenkin sen verran jumalal-

linen, ja osittain jo unohtunut mielestäni, etten siihen mielelläni koskisi.

Vihani alkaa kiehua. Nyt on todella hankala harjoittaa varovaisuutta sen suhteen, ettei minua nähdä koodaamassa liikaa. Onneksi olen sentään yksin huoneessa; se tästä vielä puuttuisikin, jos täällä olisi ilmiantaja vahtimassa minua!

Siinä vaiheessa, kun valot sammuvat käytävässä ja tulee hiljaisuus, on kuitenkin pakko lopettaa koodin tutkiminen. Jatkan pyörittelemällä sitä päässäni. Sen ja raivon yhdistelmästä seuraa varsinaista fyysistä pahoinvointia. Joudun oksentamaan, ja jälkeinpäin on aika heikko olo, ja päätä särkee. Helvetti, pitää kai oikeasti rajoittaa, jos en ole vielä ihan kunnossa.

Tämän jälkeen kierin sängyssä pitkään saamatta unta.

Päivä 35.

Koodaustehokkuus (arvioitu): 50%

Sitten jatketaan. On pakko jatkaa. Perkele, tällaiset performanssi-ongelmat eivät saa jäädä rankaisematta ja kostonmaku. Jos vertaan olotilaani eiliseen, tuntuu taas asteen verran selkeämmältä, ja uskon, että nyt saattaisin jopa uskaltaa koskea surface shader-koodiin, jos on aivan pakko.

Välissä teen epäluonnollisen pitkän kävelyretken käytävillä. Käytäviä jatkuisi myös maaston alapuolella, mutta sinne en viitsi mennä sen varalta, että minut havaittaisiin. Mutta niiden voisi melkein kuvitella olevan Throne Group:in käytäviä, jotka johtaisivat aina sinne asti, missä robotit rakentavat mekaanista Jörmungandr:ia!

Ja takaisin koodin ääreen.

Lopulta kolme kirjainta selittävät kaiken: UDB.

Alustamattomia muuttujia, jotka aiheuttavat sen, että shadereille allokoidaan käytännössä lähes joka frame uudet constant-rekisterit, jos materiaaleja on ruudulla enemmän kuin yksi.

Todella törkeää.

No niin: korjataan ja käännetään, sitten ajetaan uudelleen. Mitä ihmettä? Siltikin constantbuffereita luodaan uudelleen aivan yhtä paljon.

Kuvittelen hyvin hitaasti, vielä paljon mitään sprinttipalaveria hitaammin etenevän progress bar:in.

TAP X QUICKLY TO DEBUG CODE

Monen tunnin pelonsekaisen tutkimisen jälkeen paljastuu, että shadereille, materiaaleille ja constantbuffereille laskettavat hash-arvot muuttuvat tietyissä olosuhteissa. Tämän kieron mekanismin oli tarkoitus mahdollistaa varmatoiminen resurssien uudelleenlataaminen kesken ohjelman suorituksen, mutta nyt se kääntyiikin itseäni vastaan!

Korjataanpa heti. Nyt sen pitäisi olla taatusti deterministinen. Jos ei ole, niin minut saa hirttää vasemmasta tikkelistäni.

Kyllä. This is true power.

Framerate nousee huomattavasti (tai siis oikeastaan pitäisi aina puhua framen piirtoajasta millisekuntein, koska framerate ei muutu lineaarisesti kuormaa lisättäessä tai vähennettäessä, mutta piirtoaika muuttuu.)

Helvetti, kun minua hävettää, alkuinnostuksen hälvettyä.

Onneksi (ellei sitten joku oikeasti napannut sitä muistitikkuja, ja tutkii sitä parhaillaan ja nauraa) kukaan ei ikinä tule reviewaamaan koodin korjaamatonta versiota. Kyllä olisikin ollut kammottava vahinko, jos olisin esitellyt näin huonosti toimivan renderöintimoottorin muka jotenkin vallankumouksellisena! Kyllähän se varmasti huonoimillaankin peittoaa minkä tahansa OpenGL:llä toteutettavan renderöinnin, oli se sitten forward tai deferred, tai Vision3D:n suorituskyvyn yleensäkin, mutta silti. Hyi, hyi ja hyi.

Sisäinen profileri oli lopulta kultaakin kalliimpi osa-alue!

Mutta nyt varsinkin olisi kriittistä päästä pois. Ei olisi mukava joutua kirjoittamaan sitä tai näitä korjauksia uudestaan. Saisikohan täältä jostain paperikaupasta tms. muistitikkuja? Täytynee lähteä katsomaan. Mutta odoteetaan lääkärin kierto ensin, jos se vaikka sanoisi, että pääsen pois.

Sieltä se nyt tulee.

Jaha. Pelkän keskustelun lisäksi tänään teemme vielä neurologisia profilointiajoja ja aivojen UDB:n metsästystä.

No tehdäänpä sitten.

Katson EEG-käyrää monitorista, mutta myönnän, että se ei merkitse minulle juurikaan mitään. Kuvittelen silvonnan, ja yritän havaita jotain muutosta. Mitään muutosta ei ole havaittavissa.

Lääkäri ilmeisesti on tyytyväinen.

“Nyt ei näy enää poikkeavuuksia. Vointikin on ilmeisesti parempi?”

Nyökkään.

“Jos ei tapahdu mitään yllättävää muutosta, pääsette huomenna pois. Joko on sellainen tunne, että olisi pakko päästä takaisin töihin?”

“No... ehkä jonkin verran.”

“Saatte kuitenkin päälle vielä viikon sairauslomaa. Muistatte ottaa rauhallisesti, ja käytte sitten vielä tarkastuksessa viikon lopulla. Katsotaan silloin, oletteko jo työkunnossa.”

Nyökkään uudestaan.

Tavallaan olen pettynyt: vain viikko.

Mutta se tarkoittaa silti viikkoa äärimmilleen turboahdettua Jörmungandr:in kehitystä. Sitten paluu Vision3D:n ja mahdollisesti entisestäänkin pahentuneen multifailin äärelle.

Mitähän ne ovat oikeasti saaneet sillä välin aikaan?

En halua miettiä.

Päivä 36.

Koodaustehokkuus (arvioitu): 59%

Aikaa tykittää Jörmungandr:ia kunnolla: 7 x 24 h.

Freedom. Ruumiini kuuluu jälleen itselleni, eikä totalitaariselle Järjestelmälle. Olkoon, että sen pyrkimyksenä (yleensä) on subjektiansa terveydentilan kohentaminen, mutta Järjestelmä silti.

Oikeastaan on aika helvetin kylmä. Brr! Ja sataa lunta-kin.

No niin, sieltä se taksi tulee. Kallista lystiä tällainen, mutta ei pelkästään yhden, vaan kahden virtuaalitodellisuusmoottorin johtavana ohjelmoijana sallin sen tällä kertaa itselleni.

Tosin saa nähdä, kuinka johtava olen enää Vision3D:ssä palattuani. Olisiko sillä nyt niin väliä, vaikka olisin kokenut alennuksen, sillä vähentynyt vastuu tarkoittaisi epäilemättä enemmän aikaa Jörmungandr:ille!

Perillä.

Sekä kannettava että työasema auki välittömästi.

Sitten committi sisään, ja SVN update työasemalle.

Huomattava helpotus. Nyt uusin koodi on kolmessa eri paikassa, joten ei haittaa vaikka kovalevyjä vähän paukkuisi.

Ja ei kun muistinhallintaa jatkamaan.

Tosin onkohan se sittenkään viisasta?

Se ei nimittäin juuri näy ulospäin, ja typerän constant buffer-sekoilun performanssihyöty oli niin huomattava (ja vähensi sitäpaitsi muistioperaatioita), että lisää suorituskykyä ei luultavasti vähään aikaan ole odotettavissa.

Eli siis.

Viikko aikaa tehdä jotain ehdottoman jumalallista.

Mietitäänpä: mikä Vision3D:ssä on pahiten pielessä?

Qt.

Mikä on tärkein ja isoin toiminto, mitä sillä toteutetaan?

Graafinen käyttöliittymä.

Vaikka mietinkin aiemmin, että siihen tulee itse toteutettuna uppoamaan huomattavasti aikaa, on se kuitenkin jotain merkittävää, jos sen toteuttaa kunnolla. Kaikki tekstilayoutit, buttonit, comboboxit, scrollbarit ja muut on tietysti hyödyttävää koodata, mutta kun ne on tehnyt, voi tuntea huomattavaa tyytyväisyyttä kun tietää, ettei tarvitse miettiä minkään ulkoisen kirjaston integrointia, eikä siitä väistämättä seuraavia iljettäviä ongelmia.

Voi vain myhällä tyytyväisenä: I made this, kun muut tappelevat Vision3D:n Qt-integraation kanssa.

Tämä on nyt päätetty.

Vaihe yksi: fonttirenderöinti. Tiukkaakin tiukempi valinta: FreeType2 vai stb-truetype? Lataan FreeType2:n uusimman version, ja katson koodin läpi. C-kielisestä koodista paistaa läpi selvä tosiohjelmoinnin verenkarvainen katku, mutta kyllä sitä on paljon.

Voi pojat, sitä on paljon.

Ero on aika huomattava: stb-truetype on vain yksi header-tiedosto, kooltaan 68 kilotavua.

Koska pyrin enginen mahdollisimman pieneen kokoon heti suorituskyvyn ja featureiden jälkeen tärkeysjärjestyksessä, on valinta aika selvä.

Stb-truetyypen käyttö on hyvin suoraviivaista. Kokeiltaanpa: courier.ttf.

Hyi kamala, miltä se näyttää. Tuota ei lähes fontiksi tunnista. Mutta valinta on tehty, eikä sitä pyörretä ainakaan

heti. Sitä paitsi eivät kaikki fontit ole noin patologisia.

Vaihe kaksi: rectangle bin packing.

Pelkkä vapaan tekstuurialueen jakaminen horisontaalisesti ja vertikaalisesti ei tunnu riittävän tehokkaalta: on mahdollisuus, että hukkatilaa jää ikävän paljon. Tämän vuoksi päädyn ns. maxrects-algoritmiin, jossa träätetään käytettävissä olevia maksimaalisia pinta-aloja. Sitä paitsi se on vähän mutkikkaampi toteuttaa, mikä on aina bonus-ta.

Profiler osoittaa jälleen hyötynsä: toteutuksestani paljastuu järkyttäviä suorituskykyongelmia. Algoritmi on pahimmillaan n^4 . Hyi. Asiaan on saatava pikainen korjaus.

Lopulta havaitsen, ettei algoritmia sinänsä saa yksinkertaistettua, mutta on tärkeä siivota duplikaatit pois vapaan ja käytetyn tilan tietorakenteista, jottei suoritus aika räjähdä käsiin, kun samaan tekstuuriin on pakattu jo useampi sata kirjainta.

No niin. Nyt se toimii nopeahkosti.

Vaihe kolme: UI-widgettien kantaluokka, joka sallii hierarkisen komposition. Tähän kuuluvat tylsät jutut kuten paikka, koko, alignment ja koonmuutospolicy (kamalaa, havaitsen vaikutteita Qt:sta, mutta näin se täytyy tehdä, jos haluaa, että järjestelmä on joustava.)

Aika rumaa ja epäjumalallista koodia tästä näyttää tulevan sisältä käsin. Mutta julkisesta API:sta tulee riittävän siisti, jopa Qt:ta siistimpi.

Kyllä tästä vielä GUI-järjestelmä tulee. Ja mikä parasta, ei ainuttakaan manageria eikä singletonia.

Lopulta totean, etten jaksa enää koodata tältä päivältä. Puutumista ei enää tapahdu niin nopeasti kuin edellisinä päivinä, mutta sitten kun se yllättää, se tulee nopean brutaalisti. Jottei aika olisi kuitenkin täysin hukattua, mietin jo etukäteen widgetiluokat, jotka tulen huomenna toteuttamaan. Sen jälkeen harjoittelen kuji-kiriä. Se on ehkä hyvä terapiamuoto oireisiini.

Minulla on torrenttina imuroitu Ninja Assassin odottamassa, mutta tiedän jo edeltä sen olevan sen verran kuu-

meista ja legendaarista toimintaa, ettei sen katsominen juuri ennen nukkumaanmenoa ole ehkä viisasta. Se saa siis odottaa huomiseen.

Päivä 37.

Koodaustehokkuus (arvioitu): 64%

Aikaa tykittää Jörmungandr:ia kunnolla: 6 x 24 h.

Päätän hoitaa velvollisuudet ensin. Koska pyöräni meni törmäyksessä lähes lunastuskuntoon, on uuden ostaminen hyvin ajankohtaista.

Nimeän pyörän Valkoiseksi Kostajaksi. Kalmankalpean ulkomuotonsa lisäksi se on kymmenvaihteinen, kuten Coheed & Cambrian kappaleessa “Ten Speed (of God's Blood and Burial)”, mikä on myös sopivan pahaenteistä.

Sitten koodataan.

Tekstiwidget vaatii joitakin syvällisiä operaatioita merkkijonon katkaisemiseksi siten, ettei se mene yli widgetille rajatusta tilasta. Nyt tajuan tulkinneeni väärin fonttimetriikoita, jossa menee jonkin aikaa korjata. On tämä oikeastaan aika ali-inhimillistä koodia, eikä sopisi tosiohjelmoijan arvolle.

No niin, toimiipa se sentään lopulta.

Sitten haastetasoa korkeammaksi: pushbutton. Tämä vaatii jo interaktiota, eli käytännössä eventtien lähettämistä. Qt:ssä on inhottavasti sekaisin signaaleja ja eventtejä, mutta Jörmungandr:issa on vain eventtejä! Kokonaissuunnittelu on siis kohesiivisempi, vaikka itse sanonkin.

Yllättävän kauan näissäkin kahdessa widgetissä kuitenkin menee. Ja haastavampaa on tulossa.

Aavistan, että jos jatkaisin koodaamista yhteen me-noon, käteni alkaisivat puutua kuitenkin, joten katson tässä välissä Ninja Assassin:in. Veren määrästä ja lentävistä shurikeneista täysi kymppi, mutta filosofia jää puutteelliseksi. Tarvitaan enemmän kuji-kiriä!

Textwidget ja button tehty. Hmm. Mitä sitten? Staattinen image-widgetin saa tehtyä helposti, eikä siinä mene myöskään kauan.

Seuraava haaste: window, jonka kokoa voi muuttaa, jolla on mahdollisesti titlebar, ja mitä nyt ikkunoilla yleensä on.

Perkele.

Tämä tuo esille iljettäviä puutteita: child-widgetit eivät clippaudu oikein. Tiedän kyllä ratkaisun, mutta se on vastenmielinen: UI:n piirtäminen täytyy jakaa useampiin batcheihin, ja tällöin täytyy muuttaa scissor-statea. Onneksi tähän ei tarvitse tehdä uutta immutable-stateobjektia, vaan voidaan suorittaa yksinkertainen API-kutsu.

Nyt toimii, mutta tuntuu ilkeältä.

Puutumus alkaa nostaa päätään.

Mutta vielä yksi widget-luokka.

MouseCursor on lähes suora perintä imagewidgetistä. Se siis seuraavaksi.

Tehty. Mutta nyt kyllä ei lähde yhtään enää. Nukkumaan nyt, ja sitten huomenna aikaisemmin ylös.

Päivä 38.

Koodaustehokkuus (arvioitu): 70%

Aikaa tykittää Jörmungandr:ia kunnolla: 5 x 24 h.

Aika vähenee, mutta silti widgettejä olisi aika paljon tykittävänä. Ja lisäksi täytyy myöntää, että ne ovat ulkoasultaan hirveitä.

Niin, totta. Jonkinlaisen widget style-abstraktion ne tarvitsevat. Ettei oikeasti tarvitsisi käytännössä aina (tosi)ohjelmoijaa käyttöliittymien toteuttamiseen, ja niiden ulkoasun säätämiseen. Hyi. Ja coder arttia täytyisi ehkä väentää jotenkin hyväksyttävälle tasolle, että koko UI-systeemiä edes viitsisi ikinä näyttää kenellekään. Näissä jutuissa luultavasti tuleekin menemään suurin osa tästä päivästä.

Minkähänlainen tästä pitäisi tehdä? Ei siitä tule hyvä eikä riittävä kuitenkaan. Joku kamala XML-pohjainen kööstys. Tästä tuleekin mieleen, että vaikka TinyXml on vain 4 .cpp-tiedostoa ja 2 .h-tiedostoa, toteuttaisi XML-parserin mielellään kokonaan itse. Mutta siinä meni taas päivä tai pari hukkaan. Päiviä, jota minulla ei ole.

Tyydytään siis TinyXml:ään.

Puolet päivästä kuluu. Onhan tämä nyt jonkinlainen. Bonusta siitä, että se on oma, eikä joku ihmeellinen Falagard-UI skinning system. Ja UDB:tä ei taatusti tule, vaikka tälle syötettäisiin kuinka rappeutunut ja virheellinen tyyli-määrittely tahansa, sen sijaan heitetään vain hyvin määri-

telty exception!

No niin. Widgetstyle-systeemi tehty, mitä seuraavaksi?

Tekstieditointi on inhottavaa, mutta tehdä se täytyy kuitenkin, ennemmin tai myöhemmin.

Siis rivieditointi, ja täysimittainen tekstidokumentin peruseditointi. Nämähän peritään tekstiwidgetistä kummatkin. Hyi. Cursorin piirtäminen edellyttää inhottavaa hakkerointia, ellei sitten lisää vielä yhtä batchia. Mutta jälkimmäinen vaihtoehto ei missään tapauksessa tule kyseeseen.

API muodostuu huomattavan monimutkaiseksi, ja UI-eventtejä täytyy määritellä lisää. Ikävää. Olisin halunnut, että design huokuisi ylimaallista määrätietoisuutta ja koheesiota, mutta aika puolitiehen tämä jää sillä akselilla.

Illan muuttuessa yöksi editorit alkavat olla toiminnallisessa kunnossa, mutta rajapinta epäilyttää silti. Kyllä tämä nyt pitäisi olla kaikin tavoin parempi, vaikken tarkalleen tiedäkään, miten. Jörmungandr:in laatustandardit ovat huomattavan korkeat.

Mutta saa se kelvata. Commit.

Tänään minusta on fyysisesti tuntunut melko hyvältä. Turtumustakaan ei tullut, kuin aivan loppuksi. Mutta henkinen puoli voi olla eri asia. Voi jopa olla, että niin jumalallinen kuin Jörmungandr onkin, sitäkään ei ilman muita virikkeitä jaksaisi vääntää loputtomiin.

Ja oikeastaanhan asia on selvä: Jörmungandr on koston välikappale. Se tarvitsee Vision3D:n nemesiksekseen. Jos viimeksimainittu olisi oikeasti hyvä, ja projekti menossa kaikinpuolin järkevään suuntaan, ei sen olemassaololla olisi oikeutusta. (Tässä vaiheessa se tuntuu tavattoman suorulliselta ajatukselta, sillä olenhan vuodattanut siihen jo paljon sydänvertani ja unideprivaatiota.) Mutta toisin sanoin: Every search for a hero must begin with what every hero requires, a villain.

Tässä tapauksessa ongelma ei ole pitkäkestoinen. Kohta saan Vision3D:tä päin naamaani enemmän kuin tarpeeksi, eikä se taatusti ole edennyt poissaollessani miellyttävään

suuntaan. Ei mitenkään!
Mutta nyt nukkumaan.
Lisää widgettejä huomenna.

Päivä 39.

Koodaustehokkuus (arvioitu): 75%

Aikaa tykittää Jörmungandr:ia kunnolla: 4 x 24 h.

Mikä päivä tänään onkaan? Keskiviikko? No ei todellakaan, sehän oli jo. Siis torstai. Huomenna käyn vielä lääkärin puheilla, mutta en näe mitään estettä, miksen pääsisi heti maanantaina Vision3D:stä kärsimään, ja täyttämään kostoni maljan taas äärilaitojaan myöten täyteen!

Jörmungandr:in UI-järjestelmän koodaus, siinä välissä naamaan heitetty alkeellinen ravinto ja nukkuminen alkavat tuntua samanlaiselta loputtomalta sykliltä, kuin enginen arkkitehtuuri. Tekstieditointi oli eilen, tänään ehkäpä listbox, combobox, ja kaikenlaiset scrollbarit ja scrollattavat kanvaasit. Ja perus-dropdown-menut.

Kyllä niistä jotain tulee.

Päivän lopuksi voi todeta, että olin aivan oikeassa. Jotain todella tuli. Ainakin lisää UI-eventtejä, ja taas jotenkin erilainen API, kuin aikaisemmissa widgeteissä. Hemmetti. Juuri jumalallinen loogisuus ja ortogonaalisuus, kaikkien palasten synkroninen yhteenloksahtaminen olisi tavoitteenä, mutta nyt tavoitteesta tunnutaan jäävän jälkeen.

Ehkä, jos ohjelmoin widgetteihin silmäkarkkia ja kaikenlaisia pehmeitä transiatioita, ehkä jopa mahdollisuuden 3D-kompositointiin, niin sitten vaikuttaa paremmalta. Ainakin ulkoisesti. Sillä voi sitten uhotta, että tämäpä järjes-

telmä on runsaasti Qt:ta ihkumpi.

Tämä on aika kauhea ajattelutapa.

Häpeän, että alennun sellaiseen.

Juuri Jörmungandr:in sisäisestä kauneudesta ei saisi ikinä tinkiä. Siispä päätänkin, että loppuina kolmena päivänä toteutan seuraavan sotasuunnitelman.

1. Silmäkarkki. (Tämä on vähiten tärkeä osa-alue, mutta paras hoitaa heti alta pois.)

2. API:n refaktorointi ja yhtenäistäminen.

3. Sisäisen toteutuksen laatutason nostaminen Jörmungandr:ille otolliseksi.

Uusia widgettejä ei enää yhtäkään, ennen kuin kaikki nämä vaiheet ovat kunnossa.

Päivä 40.

Koodaustehokkuus (arvioitu): 78%

Aikaa tykittää Jörmungandr:ia kunnolla: 3 x 24 h.

Lääkärille siis. Pyörämatka läpi kirpeän pakkasen tuntuu virkistävältä: se resetoi aivot ja terävöittää aistit. Uskoisin, että kaikenlainen henkisen työn tehokkuus paranisi, jos aina olisi vähintään näin kylmä.

Saavun paikalle etuajassa. Tästä on tietysti seurauksena, että joudun odottamaan huomattavan kauan. Menetetyn ajan olisi voinut koodata UI:ta!

Vihdoin lääkäri kutsuu minut sisään.

En tietystikään rupea teeskentelemään mitään oireita, vaan vastailen kysymyksiin asiallisesti. Totean pystyneeni kirjoittamaan “nanowrimo-tekstiäni” kahdeksan tunnin työpäivää vastaavan ajan ilman fyysisiä tai henkisiä oireita, tietysti pitäen välillä asiaankuuluvia taukoja.

Selvä sitten.

Vihreää valoa palata maanantaina Vision3D:n pariin.

Kiitos ja hyvästi.

Asunnollani ryhdyn välittömästi toimiin: lataan uusimman DirectX SDK:n, jotta pystyn vertailemaan silmäkarkin määrää ja laatua DXUT-apukirjaston tarjoamaan käyttöliittymätoteutukseen. Lisäksi lataan Unity:n demoversion, jotta voin suorittaa sille vastaavan vertailun. Unitystä tulee välittömästi mieleen, että pitäisiköhän minun siirtyä im-

mediate mode GUI-paradigmaan? Tosin tässä vaiheessa se olisi aikataululle hyvin tuhoisaa.

Ei, en siirry.

En nyt enkä ikinä.

Retained mode-GUI (eli se perinteinen, että käyttöliittymäelementit luodaan, ja ne pysyvät framesta frameen) on yleispätevämpi, eikä sen toteuttaminen tai käyttäminen sisällä samalla tavalla piileviä ongelmia.

Mutta nyt pohdinnasta ja vertailusta toteutukseen.

Transitiot, hover-animaatiot ja alpha-fadet. Joo, näillähän se alkaa näyttää paremmalta. Jos kaikki näyttää reagoivalta ja pehmeästi siirtyvältä, se on myös automaattisesti ihkumpaa. Tietysti kaikki kikkailu täytyy myös pystyä kytkemään pois päältä, jotta saadaan kunnon itäeurooppalaisen teollisuuskombinaatin tai ydinvoimalan UI-tunnelma, jos tarvitaan.

Päivän kääntyessä myöhäiseen iltaan totean, että silmäkarkki on riittävä, ja jopa odotukseni ylittävä. Itseasiassa voin aavistaa, vaikken tunnekaan Qt:ta läpikotaisin, että sillä nämä vastaavat jutut on paljon hankalampi tehdä. Jos yleensä edes onnistuvat. Bonuspisteitä vielä siitä, että tänään käteni eivät turtuneet käytännössä ollenkaan.

Seuraava haaste olisi sisäinen refaktorointi. Mutta totean, että se saa jäädä huomiseksi.

Ääni pääni sisässä narskuttaa: Annatko jo periksi? Oletko heikko? Eikö Jörmungandr enää kiinnosta?

Haista paska.

Jörmungandr:in aikataulu on täysin hallinnassani. Se etenee juuri niinkuin sen pitää. Huomenna refaktoroidaan niin perkeleesti.

Päivä 41.

Koodaustehokkuus (arvioitu): 83%

Aikaa tykittää Jörmungandr:ia kunnolla: 2 x 24 h.

Hieman huvittuneena mietin, että Jack Bauer ehtisi pelastaa maailman kahdesti siinä ajassa, mikä minulla on jäljellä Jörmungandr:in käyttöliittymän saattamiseen lopulliseen, sekä sisäisesti että ulkoisesti hyväksyttävään asuun.

Tosin hän ei nuku välissä.

Nyt refaktoroidaan!

Minun täytyy myöntää, etten oikein saa kiinni siitä, mitä olisi tehtävä. UI-koodia on paljon, ja se on päällisin puolin toimivaa, mutta jonkinlainen uhka siitä huokuu. Uhka, että jos siihen lisää mitä tahansa, se saattaa räjähtää.

Toisaalta, eikö tämä juuri ole merkki siitä, että se on tosiohjelmoijan kädenjälkeä? Tosiohjelmoija ei refaktoroi, vaan siirtyisi jo seuraavaan tehtävään (tai joisi olutta.)

Katson vertailun vuoksi scenenhallinnan ja renderöinnin koodia. Oikeastaan, jälkikäteen tarkasteltuna, ne ovat aivan yhtä kryptistä, pelottavaa ja räjähdysherkkää koodia. Mutta API on yhtenäisempi.

Teen siis päätöksen: koodi saa sisäisesti olla juuri sitä, mitä se onkin, mutta ulkoinen rajapinta siistitään.

Tässä minulla ei mene kauankaan.

Arvioni ei ehkä aivan mene nappiin: keskiyöllä totean viimein, että kaikki on kunnossa. Nyt UI-koodi on valmista,

eikä siihen enää kosketa.

Havaitsen vain, etten ole syönyt käytännössä koko päivänä. Jääkaapin sisällön tarkastus paljastaa, että mitään ruoaksi luokiteltavaa siellä ei olisikaan.

Ei ole miellyttävää mennä nälkäisenä nukkumaan. Pyörin sängyssä pitkään saamatta unta.

Päivä 42.

Koodaustehokkuus (arvioitu): 66%

Aikaa tykittää Jörmungandr:ia kunnolla: 24 h.

Kun herään, minulla on kylmä (nälkää ei varsinaisesti enää tunnu) ja tunnen itseni eläväksi raadoksi. Onneksi kaupat ovat sentään auki. Siispä hankkimaan täydennystä.

Valkoisen Kostajan lukko on jäänyt, eikä minulla ole sytkäriä. Tänään kaikki on minua vastaan! No, tämänkin asian pystyy korjaamaan. Kävelläänpä sitten.

Viimeisestä tykityksen päivästä on kulunut harmillisen paljon, kun olen saanut nämä välttämättömät velvollisuudet hoidettua.

Niin, totta. UI tuli valmiiksi.

Mitä sitten?

Meshien animaatiojärjestelmä. Ehdottomasti.

Päätin jo kauan sitten, että minkäänlaista CPU-skinnin-
giä ei ehdottomasti tulla toteuttamaan. Ei edes vertex
morphausta lasketa prosessorilla. Kaikki raa'asti GPU:lla.
Tässä tulen hyödyntämään Direct3D10:n streamout-omi-
naisuutta, jolla vertex shaderin ulostulo voidaan ohjata
verteksipuskuriin. Tämä tarkoittaa esimerkiksi, että jos
animoitu objekti joudutaan piirtämään useassa passissa
(mille ei normaalitilanteissa pitäisi missään nimessä olla
tarvetta) tarvitsee animaatio laskea vain ensimmäisellä
passilla, ja tämän jälkeen shaderin muuttamat verteksit

ovat tallessa puskurissa. Ongelmana on tietysti kasvanut muistinkäyttö: verteksipuskuri jokaista animoitavaa hahmoa kohden. Vaikka Jörmungandr:ia ei olekaan tarkoitettu heikoille näytönohjaimille, on minun ehkä pakko antaa periksi, ja antaa vaihtoehto, jossa streamout:ia ei käytetä, vaan animaatio lasketaan jokaista passia kohti lennossa. Näin meshistä tarvitsee olla vain yksi kopio näytönohjaimen muistissa.

Oikeastaan heti, kun ryhdyn toteuttamaan animaatiota, havaitsen, ettei se missään tapauksessa tule valmistumaan tämän päivän aikana. Animaatioiden blendaus voidaan tehdä hyvin monella tavalla, ja koska Jörmungandr:in on tarkoitus olla paras, sen täytyy tukea niitä kaikkia! Lerpblendaushierarkiapuu, additiiviset ns. difference poset eli säättöanimaatiot, joilla hahmon liikettä voidaan hieman muuttaa elävyyden lisäämiseksi... Tässä tulee riittämään tekemistä!

Mutta ensin täytyy saada perus-GPU-skinnaus toimimaan. Havaitsen, etten täysin ottanut huomioon skinnausvariaatioita tehdessäni shader fragment-kääntäjää. Voidaan esim. ajatella, että joskus halutaan käyttää maksimissaan kahta bonea vaikuttamaan yhteen verteksiin, joskus kolmea, joskus neljää. Suorituskykyero on käytännössä nykyisillä näytönohjaimilla minimaalinen, mutta lämmittää silti sydäntä tietää, että Jörmungandr hallitsee kaiken!

Joudun tekemään hieman vaarallisia puukotuksia shaderinhallintaan. Lisäksi ensimmäiset yritykseni tuottavat silti groteskisti vääntyneitä hahmoja, joiden katsominen aiheuttaa lähes pahoinvointia (tosin siihen turtuu äkkiä.) On selvää, ettei skinnausmatriiseja haeta oikeista constant-indekseistä. Joudun siis reviewaamaan myös constant buffer-koodini, johon en olisi halunnut enää koskea.

Tänään todellakin kaikki on vastaan!

Tai oikeastaan nyt vain koodini heikkous tulee esiin, sillä sitä ei oltu vielä testattu maksimaalisesti. En ollut vielä käytännössä testannut "joka frame, joka objekti" -frekvenssillä tapahtuvia muutoksia constant-datassa.

Direct3D9:llä kaikki olisi ollut niin paljon helpompaa!

Mutta on heikkoutta ajatella noin. Constant bufferit ovat tulevaisuus. On selvää, että ne ovat rautaa lähempänä oleva abstraktio: jokainen puskuri päivitetään aina kerralla ohjelmoimalla DMA-siirto tapahtumaan keskusyksikön ja näytönohjaimen muistin välillä.

No niin. Nyt se toimii.

Se toimii täydellisesti. Muahahaha!

Tai siis nyt vasta sain hahmon skinnautumaan identity-matriisilla oikein. Eli varsinaista animaatiota ei vielä ole. Ei siis vielä aivan ole syytä juhlia.

Seuraavaksi varsinainen luustomatriisien päivittäminen, mikä tapahtuu seuraavasti:

```
for (unsigned i = 0; i < bones.size(); ++i)
{
    if (bone->needSkinningUpdate())
    {
        mSkinMatrices[i] = bone->getWorldTransform() *
bone -> getBindInverseTransform();
    }
}
```

Helppoa kuin mikä. Tosin en vielä ota huomioon mahdollisuutta, että kaikkien luiden matriisit eivät mahtuisi constantbufferiin, vaan jouduttaisiin tekemään submesh-kohtainen uudelleenindeksöinti. Mutta se on triviaalinen operaatio. Ja sitä paitsi, koska käytän Direct3D10:tä enkä yhdeksää, pitäisi mallissa olla järjetön määrä luita, etteivät matriisit mahtuisi puskuriin.

Saan yksittäisen animaation playbackin toimimaan.

Tämä on hyvä alku. Mutta kello on jo yli keskiyön, koko animaatiojärjestelmä ei missään tapauksessa tule valmistumaan, ja aamulla täytyy lähteä töihin. Joten aika pistää kone kiinni. Sitä ennen commit.

Command Commit to svn://192.168.0.2/svn/jormun-

gandr/trunk

*Error Can't connect to host '192.168.0.2':
Error A connection attempt failed because the
Error connected party did not properly respond
Error after a period of time, or established
Error connection failed because connected host
Error has failed to respond.
Finished!*

The operation failed.

Mitä v.....?

Pikainen tutkimus paljastaa, ettei tiedosto- ja versionhallintapalvelin osoita mitään älyllistä toimintaa. Hieman syvällisempi tutkimus paljastaa, että sen kovalevy on pamahtanut. Tämä tarkoittaa, että Jörmungandr:in versiohistoria ja torrent-arkistoni on tuhoutunut. Uusin koodi on toki tallella työasemallani, ja uusi kovalevy täytyy tietysti hommata välittömästi, mutta serverin pystyttämisessä uudelleen tulee kulumaan päivä.

Päivä hukkaan Jörmungandr:in kehityksestä!

Päivä jota minulla ei ole.

Haluan huutaa helvetin kovaa, mutta hillitsen itseni.

Pyrin rationaaliseen ajatteluun. Tämä edellyttää, että säädän herätyskellon seitsemäksi, jotta ehdin kahdeksaksi töihin ja neljältä pois. Näin on varmasti aikaa hankkia kovalevy, ennen kuin tietokoneliikkeet menevät kiinni.

Jos aikaa on pakko hukata, on se paras hukata mahdollisimman pian.

Raivo jäytää ja kalvaa sisuskalujani. On selvää, ettei unta tule aivan heti. Mutta on pakko silti yrittää. Ensimmäinen työpäiväni vähään aikaan tulee taatusti olemaan vihan ja kyräilyn täyteinen.

Kun lopulta nukahdan, uneni on täynnä sotaa, terroria ja kidutusta. Oikeastaan nautin siitä, ja toivon, ettei uni lopu tai vaihdu.

JIN - Muiden ajatusten tietäminen

Päivä 43.

Herätyskello sentään toimii. Tai oikeastaan aavistan herätyksen paria minuuttia ennen. Ninjamagia ja kuji-kiri. Muiden ajatusten tietäminen. Ehkä sen esivaihe on, että pystyy aavistamaan mekaanisten ja elektronisten laitteiden toiminnan. Sitten, kun kyky on kehittynyt täysin, pystyy lukemaan myös ihmisten ajatuksia. Mutta serverini kovalevyn rikkimenoa en silti osannut aavistaa! Sen kyllä arvaan ilman kuji-kiriäkin, että tästä päivästä täytyy tulla kamala.

Alkeellinen ravinto suuhun, suuhygienian yhtä alkeellinen hoitaminen, ja menoksi. Pakkasta ei ole paljon, mutta unenpuute moninkertaistaa sen. Hyi!

Vision3D odottaa.

Saavuttuani paikalle (joudun kestäämään inhottavia "tervetuloa takaisin" -toivotuksia, mutta onneksi ne menevät äkkiä ohi, ja kaikki palaavat normaaleihin, vähemmän tekoystävällisiin persoonallisuuksiinsa) havaitsen nopeasti, että ilman kovalevyn hajoamistakin päivästä olisi tullut hirveä.

Vision3D:n uudelleenorganisointi on tapahtunut.

Nyt heitä on kaksi.

Tavallaan hyvä uutinen on, etten ole enää pääarkkitehti. Tämä käytännössä tarkoittaa, että voin keskittyä jatkossa implementaatiotehtäviin enemmän. Vaikkakin Vision3D:n implementointi ei ehkä paikoitellen voisi vähempää kiin-

nostaa. Myöskin tavallaan hyvä uutinen on, että arkkitehtuuri- ja alijärjestelmätimejä ei enää ole (jako olikin turhan keinotekoinen) vaan sen sijaan olemme jakautuneet coresysteemi- ja sovellustiimeihin. Minä kuulun luonnollisesti ensiksimmäisiin.

Mutta.

Ehdottomia nemesiksiä on nyt kaksi. (Tämän muutoksen valossa on selvää, että Kaarlo ei ikinä tosiasiaassa ole ollutkaan luettavissa sellaiseksi. Kaarlo on suhteellisella asteikolla jopa täysin OK tyyppi, kunhan hänestä pysyy riittävän kaukana.) Jarmo on edelleen projektipäällikkö, tai pääohjelmoija, tai miten asian nyt haluaa mieltää.

Hänen rinnallaan on nyt ulkomaalaisvahvistuksemme Suresh, joka astuu pääarkkitehdin saappaisiin.

Sanotaan nyt heti, että minulla ei ole minkäänlaisia nationalistisia ennakkoluuloja. Kokemukseni on, että kaikki me olemme kusipäitä, kansallisuuteen katsomatta. Arvioin jokaista täysin puolueettomasti yksilönä.

Jos Jarmo on Juippi, Suresh saa lisänimen Kuumakalle. Päällisin puolin hän näyttää ystävälliseltä ja kohteliaalta, mutta heti kun tulee kyse koodista tai arkkitehtuurista, alkaa hirveä, agitoitunut pajatus.

Kyllähän minä arvostan raivoa. Se on tärkeä voimavara työstettäessä virtuaaliodellisuusengineä. Mutta sen tulisi olla fokusoitua ja mieluiten kyräilevää. Suresh tempoo joka suuntaan, jolloin ei taatusti pysty etenemään kunnolla yhteenkään niistä.

Tästä päivästä lähtien on selvää, että jokainen päivä tulee olemaan täyttä sotaa.

Kuumakallen ensimmäinen ja tärkein mandaatti on, että tästä eteenpäin Vision3D:n täytyy tukea monisäikeisyyttä ja moniprosessorijärjestelmiä täysin siemauksin.

Mitä nyt muistan koodista, jota teimme ensimmäisen ja toisen sprintin aikana, niin aika syvään siinä oli iskostunut käsitys pääthreadista, ja siitä, että se käsittelee sceneä ja ajaa alijärjestelmiä eteenpäin, kukin vuorollaan, kunnes koko frame on prosessoitu. Poislukien ehkä fysiikkasimu-

laatio ja skriptaus, en usko, että CPU tulee olemaan mil-lään tavalla pullonkaula – edes Vision3D:ssä. Ei vaikka koodi olisikin hieman suboptimaalista. GPU:n toiminta monimutkaisilla sceneillä ja paskoilla näytönohjaimilla, joita meidän pitäisi tukea, se tulee olemaan ongelma!

Mutta ei Suresh tätä usko. Hän on läpiajamassa moni-säikeisyyttä tajuamatta ehkä kunnolla, kuinka syvällisiä vaikutuksia sillä tulee olemaan koodiin. Kuinka paljon mahdollisesti joudutaan heittämään romukoppaan. Jarmo ikäänkuin katsoo tilannetta sivusta ja myötäilee, tajuamatta minkälaiseen suohon mahdollisesti ollaan menossa.

Nyt pitäisi CEO:mme lukea tyly laki ja mahdollisimman äkkiä. Mutta ehkei hänkään ymmärrä, tai sitten Kuumakalle on saanut hänet ylipuhuttua. Ehkä jollain ihmeen maalailuilla vaikka raketin nousun simuloinnista, jossa voidaan simuloida jopa jokaisen maalihileen liike, kun hyö-dynnetään täysipainoisesti Intel Core i7:n kaikkia coreja!

Voi helvetin perse.

Mutta tässä nyt ollaan joka tapauksessa. Kolmatta sprinttiä on jo mennyt lähes viikko, ja näyttää siltä, että toisen sprintin alussa asetettu päämäärä – EduWorld:in ajaminen pääpiirteittäin – on ehkä vain siirtynyt kauem-maksi. Vaikka kaikki alijärjestelmät ovat poissa ollessani edenneet, ne eivät vain ole edenneet tarpeeksi! Ja nyt vielä tulee lisää ylimääräistä vaikeuskerrointa.

Minun pitäisi nyt istua alas Suresh:in kanssa, ja katsoa, mitä taskeja alan vielä tekemään loppusprintin aikana. Ajatus tuntuu hyvin suuresti vastenmieliseltä. Suoraan sanoen huvittaisi vain kävellä ulos ja mennä hakemaan se kovalevy.

Focus! You're a coder!

A universal coder.

Jos et kykene vastaamaan edes Vision3D:n haasteeseen, jonka sentään on tarkoitus vain olla aika kökkö, tai siis ettei siltä lopulta vaadita mitään kovin ihmeellistä, miksi kuvittelisit kykeneväsi vastaamaan Jörmungandr:in haasteeseen, vaikka oletkin sitä jo tehnyt tyypillisen harrasteli-

jaenginen asteelle!

Perkele.

Aivan niin. Täytyy fokusoitua ja rauhoittua.

Manaam Jensiin esiin, ja alan sitten kuunnella Sureshin litaniaa ja päätystä. Hänellä on minulle paljon tekemistä. Hyvinkin paljon. Core-järjestelmän läpikäynti, ja threadauksen mahdollistaminen järjestelmätasolla näin aluksi. Sitten sama scenemallille.

Siitä ei taatusti tule hauskaa.

Kieroutuneesti ajateltuna voin ehkä silti kääntää tämän edukseni: ehkä tämä on se potku perseelle, jota kaipasin. Jos Vision3D on tästä eteenpäin näin hirveää ja älytöntä, niin eikö se tarkoita, että Jörmungandr tulee vastaavasti olemaan sitä suurempaa nautintoa?

Toivon, että asianlaita on näin.

Keskustelutuokion päätyttyä lähden mekaanisesti lisäämään threadausprimitiivejä Vision3D:n framework-tason koodiin, jotta sitä voidaan ajaa turvallisesti useassa säikeessä. Mutexien tai critical sectionien käytöstä seuraa tietysti aina koodin suorituksen hidastumista, joten profiloim jatkuvasti. Universal coder ei anna koodinsa hidastua liikaa, vaikka projekti olisikin paska!

Suureksi ihmeekseni threadaus tempaa minut mukautsa siinä määrin, että unohdan lähes syödä. Lähden viime hetkellä kohti ruokalaa.

Tyylipuhdas antisosiaalinen ruokailu: muita ei näy. Hyvä.

Jälkiruokana on hedelmäsekoitus ja vaniljariisi. Näiden päällä on suklaakastiketta ennaltamäärättyä sekoitukseksi: kaikki valinnanvapaus oli otettu pois. En voi sanoa, että maku olisi varsinaisesti epämiellyttävä, mutta lyijyisen sekava se on, kuten myös jälkiruoan ulkomuoto. Epämääräisyys kuvastaa hyvin mielentilaani tällä hetkellä. On selvää, että Vision3D on menossa entistä kauheampaan suuntaan. Mutta on joko otettava tai jätettävä, kaikki tai ei mitään. Tämän muistan jo päätelleeni loppuun asti: Jörmungandr tarvitsee Vision3D:tä, niin vastenmielinen aja-

tus kuin se onkin. Valo tarvitsee pimeyttä, hyvyys pahuutta, ja sankari roistoa. Ilman kontrastia ei olisi mitään tapaa vertailla.

Mutta tarvitseeko Vision3D Jörmungandr:ia?

Niin, eihän se tietenkään tarvitse. Se tulee etenemään omalla painollaan, ja keräämään lisää typeryyttä painolas-tikseen, kunnes se joko vain kuolee mahdottomuuttaan pois, TAI kunnes saavutetaan kertakaikkinen kriittinen massa. Ehkä sitten univesummi räjähtää ja aika pysähtyy. Sitten valo-olennot (tai esi-isien henget, tai whatever) koo-daavat tyhjiydessä täydellisen AAA-pelimoottorin, joka soveltuu MYÖS mihin tahansa 3D-internet -sovellukseen. Sääli vain, että me elolliset emme ole enää siitä nauttimas-sa!

Sitten palaan vielä pariaksi tunniksi jatkamaan (nyt saan onneksi olla suhteellisen rauhassa), kunnes on aika pois-tua kovalevyostoksille.

Ilta ja alkava yö kuluvatkin rattoisasti Windows 7-käyttö-järjestelmää ja välttämättömmimpiä sovelluksia uudelleena-sentaessa. Huokaisen helpotuksesta, kun saan SVN-serve-rin ja VPN-gatewayn jälleen pystyyn, ja perustan uuden Jörmungandr-repositorion. Sitten äkkiä initial commit!

Koodi on jälleen turvassa ja synkronoitu kolmelle eri koneelle. Animaatiojärjestelmää en kyllä ehdi jatkamaan yhtään, vaan menen suosiolla nukkumaan.

Päivä 44.

Uusi päivä, ja lisää threadausta.

Tämä on kyllä ohjelmistokehitystä aivan uudelta infernaaliselta alatasolta. Siis oikeasti, joudun selittämään Suresh:ille valehtelematta ainakin puoli tuntia yhden noin kymmenrivisen, critical sectionin sisällä ajettavan sisemmän silmukan toimintaa. Hän on piru vie pääarkkitehti, implementaatio on meidän harteillamme ja hänen pitäisi luottaa siihen! Lapsellista paskaa. Lopulta vihjaan tästä, jolloin hän poistuu paikalta myrtyneenä ja selittää jotain sekavaa yhteistyöhaluttomuudesta.

This is war.

Tai en tiedä, onko se edes. Melkein tulee ala-aste, joku typerä tosi-TV-ohjelma, tai sitten aivan saatanan ennaltaaavistettavasti ja huonosti kirjoitettu elokuva mieleen. Siis oikeasti, eihän tällaista voi olla todellisuudessa!

Mutta jos tämä inspiroi Jörmungandr:ia kohti yhä ylempiä sfäärejä ja yhä nopeampaa toteutustahtia, niin hyvä niin.

Tosin ennemmin uskon, että jos tällainen kyyti jatkuu, se ajaa minut katatoniaan ja alkoholismiin. Ei helvetti.

Ei helvetin helvetin helvetin helvetti.

Sebastian pyörittää EduWorld-sceneä omalla koneellaan. Hänellä on animaatio- ja partikkeliefektijärjestelmät työn alla. Ei se nyt aivan toimi: oma hahmo (verkkokoodia

ei vielä suinkaan ole, joten kaikki on paikallista) ja efektit liikkuvat töksähdellen, vaikka käsittääkseni hän onkin otanut runsaasti koodia 1.4-versiosta, ja tehnyt vain minimaalisia muutoksia.

Näinköhän käy, että toisaalta otamme paljon vanhan version koodia sisään, ja samalla prosessoimme sitä muka monithreadatuksi? Jostain syystä mieleeni tulee The Human Centipede-elokuva. Mutta kumpi pää nyt edustaa suuta ja kumpi peräaukkoa? En voi tietää varmuudella.

Tietysti pitäisi iloita siitä, että scene yleensä latautuu. Se tarkoittaa, että alijärjestelmät toimivat jo suhteellisen hyvin yhteen. Mutta en juurikaan kykene tuntemaan iloa.

Kaarlo julistaa aloittavansa tänään verkkokoodin päivittämisen 2.0-ajalle. Vanhat Vision3D:n versiot käyttivät hyvin yksinkertaista ja raakaa menetelmää: palvelimelle yhdistettäessä avatiin sekä TCP- että UDP-yhteydet. Luotettavat viestit siirtyvät streamimuotoisesti TCP-kanavaa pitkin, ja epäluotettavat viestit (kuten objektien liikedata) lähetettiin UDP-paketeissa. Koska yhteydet kilpailevat samasta kaistasta, eivätkä ole tietoisia toisistaan, kaikenlaisia tökkimistä ilmeni heti, jos monenlaista tietoa liikkui yhtä aikaa, eikä kyseessä ollut lähiverkkoympäristö.

Kaarlon valinta osuu ENet-kirjastoon, joka tarjoaa luotettavat ja epäluotettavat, järjestetyt ja epäjärjestetyt viestit yhden UDP-yhteyden päällä. Teoriassa kuulostaa hyvältä, mutta sittenpä nähdään. TCP- ja http-yhteyksiä tullaan silti luultavasti käyttämään eri palvelimien kanssa (käyttäjäprofiilit, asettien hakeminen ja tallentaminen), mutta jos itse simulaatioverkkoliikenne saadaan kaikki UDP-kanavan päälle, on sekin jonkinlainen voitto.

Jörmungandr:iin tulen tietysti toteuttamaan verkkokoodin alusta asti itse. Niin matalan tason toiminnallisuudessa, ja niin jumalallisen enginen kyseessä ollessa ei voi luottaa kuin omaan koodiin.

Kuumakallen tärkein huolenaihe Vision3D:n verkkokoodin kannalta on, että onhan ENet varmasti thread-turvallinen? No helvetti, verkkoalijärjestelmä on sentään

helppo threadata itsekin, ei tuollaisesta tarvitsisi välittää:

```
void Network::queuePacket (const Packet& packet)
{
    queueMutex.lock ();
    queue.push_back (packet);
    queueMutex.unlock ();
}
```

Okei, myönnetään: tämä ei ole missään tapauksessa hyväksyttävää tuotantotason koodia, mutta periaate on siinä, ja loppu profilointia ja optimointia.

Mutta hyvä niin, jos Suresh pysyy Kaarlon kimpussa, eikä minun. Tosin tiedän, että näin ei jatku kauan: heti kun aloitan scenemallin thread-turvallisuuden varmistamisen, helveti jatkuu.

Aavistukseni osuu oikeaan.

Päivän avainsana on software transactional memory.

Ilmeisesti hän haluaisi, että sceneä muokattaessa tehtäisiin committeja tai transaktioita. Vähän niinkuin versionhallintaan. Sitten jos tulee konflikteja, on mahdollista, että muokkaus joudutaan perumaan. Tätä varten tarvitaan undo-logia. Lisäksi on mahdollista, että jos toteutetaan optimistinen lukeminen (eli ei lukita, kun luetaan dataa), tulii lukeneeksi epäkonsistentti tila, ja koko operaatio on alusta lähtien tuomittu. Jos ei olla varovaisia, saattaa olla, että algoritmista riippuen pahimmassa tapauksessa operaation peruminen ei riitä, koska ei edes päästä siihen asti: sitä ennen ollaan aiheutettu UDB:tä (esim. segmentation fault) tai jouduttu ikuiseen silmukkaan.

Hyi helveti, kun on monimutkaista ja ikävää.

Selitän, että kannattaisi ensin tunnistaa ne alijärjestelmät, jotka oikeasti tulevat accessoimaan sceneä yhtä aikaa, ja sitten katsotaan, mitä täytyy pistää critical sectioiden taakse, tai mistä datasta kullakin alijärjestelmällä pitää olla omat kopionsa, jotka sitten synkronoidaan esimerkiksi framen lopuksi.

Mutta tämä on selvästi ei-toivottua informaatiota.

Jens ei suostu tulemaan esiin. Tämä on liian tärkeä asia, että siinä voisi antaa periksi. Nyt ei perkele hörhöillä.

Katsomme läpi STM-kirjastoja. Kyllähän niitä on, mutta olisi aika kamalaa, että kaikki sceneohjelmointi olisi täl-laista. Se asettaisi huomattavia vaatimuksia sekä omalle sovellustiimillemme, että niille onnettomille ulkopuolisille, joiden pitäisi Vision3D:tä käyttää. Ja kaikki vain jonkin täysin teoreettisen tarpeen vuoksi! Lisäksi ei ole ollenkaan poissuljettu mahdollisuus, että suorituskyky itseasiassa laskisi. (Oikeastaan se olisi todennäköistä.)

Suresh haluaa, että luen läpi erään dokumentin (taitaa olla tietojenkäsittelytieteen opiskelijan gradu) STM-pohjaisesta scene-toteutuksesta. No, luetaanpa.

<https://urresearch.rochester.edu/fileDownloadForInstitutionalItem.action?itemId=5752&itemFileId=8929>

En ole vakuuttunut. Minua iljettää ajatus siitä, että renderöitäisiin frame, jossa osa objekteista olisi vielä päivittä-mättä. Siis oikeasti, haista nyt jo paska ja jätä minut rauhaan. Tietän kyllä, että minulla ei ole ehkä riittäviä tietopohjaisia argumentteja, mutta tunne- ja perstuntuma-pohjaisten tulisi riittää. Toteankin siis lopulta tylsty:

“Jos haluat, että Vision3D:hen todella toteutetaan STM, saat löytää jonkun muun tekemään sen. Tässä projektissa ollaan jo ammuttu itseämme jalkaan riittävästi. Voin kyllä toteuttaa perinteisen thread-turvallisuuden sceneen, mutta siten, että se on kytkettävissä pois, jos sitä ei tarvita. Ja jos pystyt näyttämään ne profiloinnilla havaitut alijärjestelmät, joiden oikeasti tarvitsee pyöriä yhtä aikaa ja päivittää sceneä, katsotaan niiden välistä synkronointia erikseen.”

Alkaa edellistäkin hirveämpi urputus yhteistyöhaluttomuudestani. Tuli selväksi. Olen siis menneisyudessa elävä urpo, joka ei ymmärrä modernien 3D-engineiden suunnittelusta mitään. Haluan kysyä, että kuinka monta engineä

sinä olet suunnitellut ja shipannut, mutten aivan viitsi.

Suresh kuitenkin poistuu lopulta, ja alan työskennellä suunnitelmani mukaisesti. Katsotaan sitten, mitä jatkossa seuraa.

Työpäivän ollessa lähes lopussa hän palaa vielä hieman leppyneenä, ja ilmeisesti siis hyväksyy käyttämäni lähestymistavan. Näytän vielä piruuttani profileri-ikkunaa: varsinainen scenelogiikan ajaminen on esim. EduWorld:issa (siltä osin kuin se nyt yleensä suostuu toimimaan) ehdottomasti vähiten CPU:ta kuormittava toimenpide. Mitäs siellä onkaan listan huipulla:

Graphics::RenderFrame

QApplication::notify

Joo, tätä faktaa ei millään threadauksella muuteta.

No niin. Vapaus koittaa.

Havaitsen itseni henkisesti hyvin väsyneeksi. Krieg vaatii veronsa, ja Jörmungandr:in olemassaolo ehti jo lähes unohtua. Mutta silti pitäisi jaksaa animaatioblendaus- ja verteksimorfauuskoodin parissa.

Jos tätä rataa jatketaan, ennuste ei ole hyvä.

Päivä 45.

Jonkinlainen aselepo on ehkä havaittavissa. Scenen threadaus alkaa valmistua hyvää tahtia. Pistän huvikseni kymmenen eri threadia luomaan ja tuhoamaan entityjä, ja muuttamaan niiden ominaisuuksia. Eihän se ole millään tavalla järkevää, eikä scenen konsistenttiutta voi arvioida kuin teknisellä tasolla, kun se tässä testissä on vain kasa randomisti liikkuvia laatikoita, mutta teknisesti tulos on hyvä:

Ei UDB:tä.

Ei kirjoittamista tai lukemista epäkoherentteihin tiloihin.

Ei kohtuutonta vaikeutta sovelluksen ohjelmoijalle.

Vain noin 5% performanssimenetyks verrattuna tilanteeseen, jossa threadaus kytketään pois, ja sceneä käsitellään vain pääthreadista.

Loppupäivästä esittelen tulokset Suresh:ille. Hän nyökkyy hyväksyvästi, ja värjään taskin sprint backlogista vihreäksi. Tosin nyt mietin, tulinko juuri tehneeksi aivan saatanallisen tason virheen. Ajaudunko minä nyt jonkinlaiseksi Kuumakallen sylikoiraksi? Perkele. Kunnon krieg olisi ehkä hyväksyttävämpi vaihtoehto, vaikka onkin hyvin kuluttavaa.

Lähden hieman etuajassa. Huomenna sitten uusia haasteita. Mutta välissä Jörmungandr:ia.

Animaatioblendaukseen olin jäänyt, ja animaatioblendauksesta jatketaan. Taustamusiikiksi Impaled Nazarenen Ugra Karma.

Useiden animaatioiden lerp-blendaus. Check.

Animaatioiden blendausprioriteetti, ja hierarkian alkupisteen valinta (jotta voidaan soittaa esim. kävelyanimaatiota pelkästään hahmon alaruumiilla, kun yläruumis tekee muuta.) Check.

Animaatioiden automaattinen sisään- ja ulosfeidaus. Check.

Viimeksimainittua tehdessäni havaitsen iljettävän, mutta mielenkiintoisen ominaisuuden, jonka päätän jättää koodiin: jos animaation painoarvon antaa mennä alle nollan, tai yli yhden, tapahtuu mutilaatioita, joita ei voi sanoin selittää, ne pitää itse nähdä. Lopullinen API-kutsu muotoutuu seuraavaksi:

```
setAnimationWeight (float weight, bool limit = true);
```

Toisinsanoen oletuksena on, että painoarvo rajoitetaan nollan ja yhden välille, mutta oletuksen saa tarvittaessa pois päältä. Tällä voi esim. simuloida äärimmäistä ninjamagiaa, jolla viholliset saa painumaan kasaan, tai räjähtämään ulottuvuuden kaikkiin suuntiin! (Ja samalla luultavasti saa peliinsä K18-leiman: Strong disturbing mutilations.)

Additiiviset posenmuokkaus-animaatiot ovat hankalampi ominaisuus, sillä niitä varten koko blendauskoodista joutuu kirjoittamaan toisen version. Lisäksi testidataa ei ole, sillä animaatioformaatti, jota käytän, perustuu luiden absoluuttisten transformien tallettamiseen.

Mietin hieman.

OGRE käyttää pelkästään additiivisia animaatioita, toisin sanoen animaatiodata talletetaan bindiposeen suhteellisenä. Siitä saankin testidataa ylitsepursuavasti. Varmistan järjestelmäni toimivuuden OGRE:n ninja-meshillä, joka

ei missään nimessä ole AAA-tason arttia. Mutta ei haittaa.

Hyvin toimii. Check.

Jatkoon jää käytännössä vain streamout ja vertex morph-tuki, eli Jörmungandr:in animaatiojärjestelmä alkaa olla hyvällä mallilla. Kellokin on vain kaksi yöllä. Täsähän ehtii vielä jossain määrin palautumaan huomista varten.

Päivä 46.

Jos eilinen päivä oli hyvä, tämä päivä ei suinkaan ole. Tai ainakaan se ei ala hyvin. Seuraava haasteeni tulee olemaan Vision3D:n fysiikkatoteutus. Ei siinä mitään, fysiikkahan on jokaisen itseään kunnioittavan enginen tärkeä osa-alue.

Mutta helveti, tuo Suresh:in tapa esittää asia!

Tyyppin jutuista tulee mieleen, niinkuin se kuvittelisi, että me pian olemme mallintamassa jo yksittäisten atomien tai ehkä jopa alkeishiukkasten liikettä. Koko skaala kvanttimekaniikasta suhteellisuusteoriaan!

Hyh mitä kamaluutta. Miksi pilaat hyvän taskin lataamalla sen täyteen tuollaisia naurettavia odotuksia, mitkä eivät missään nimessä tule toteutumaan, ja joita edes höhöimmätäkään asiakkaamme eivät odota?

Tai jos ei sentään atomeja, niin ainakin se kuvittelee, että olisimme mallintamassa lähestulkoon heti kappaleiden deformatumista oikein, soft body -fysiikkaa, ja muuta mukavaa. Insinööritason laskentaa, mutta reaaliajassa.

Ja arvatkaa miten? Arvasitte oikein, multithreadauksella tietenkin. Tai sitten GPU-laskennalla.

Kill me now.

Seuraa pitkälinen keskustelu, jossa yritän tehdä selväksi, mitä ehkä on mahdollista saavuttaa siinä aikataulussa, joka meillä on. Tärkeintähän on saada käyttäjien ohjaamat hahmot liikkumaan hyväksyttävällä tavalla (tyypillisin Vi-

sion3D:llä tehtävä sovellus on edelleen sellainen, jossa käyttäjä ohjaa bipedaaliavataria joko ensimmäisen tai kolmannen persoonan näkökulmasta), ja varmistaa, että törmäystarkistus staattisen scenen kanssa toimii mahdollisimman luotettavasti, jopa patologisissa tapauksissa. Tarkalleen ottaen näihin ei edes tarvitsisi varsinaista fysiikkamoottoria, jos hahmojen yksinkertaistetun fysiikan ja törmäystarkistuksen tekee itse.

Seuraava prioriteetti on yksinkertainen jäykkien kappaleiden liike, jotta maailmassa olevat esineet tuntuvat käyttäytyvän fysikaalisesti. Jos nämä kaksi asiaa saataisiin tehtyä ennen joulua, ja vieläpä hyvin, olisin itse todella tyytyväinen. Aikaahan on noin viikko.

Mutta todellakaan nyt ei ruveta mitään deformaatiomallinnusta ihkuilemaan. Heräisit jo.

Seuraa ensin huutoa, jota koko toimisto tuntuu katselevan (me molemmat huudamme) ja sitten kamalaa jurpustakaan (minä pysyn vaiti.) Mutta näyttää, että voitoin tämän argumentin.

Kun lopulta pääsen varsinaisesti koodin ääreen, päivä on kulunut jo puoleen väliin. Fysiikkakirjaston valinnasta sentään minä, Suresh ja Jarmo olemme yhtä mieltä: Bullet. Sitä käytetään jopa useissa kaupallisissa peleissä, se on suorituskykyinen ja monipuolinen, ja sisältää oliopohjaisen rajapinnan, toisin kuin Vision3D 1.4:ssä ja vanhemmissa käytetty ODE, jota täytyy ohjelmoida kömpelön proseduraalisesti, vaikka se sisältä käsin käyttäekin C++:aa.

PhysX olisi myös hyvä vaihtoehto, mutta Vision3D:n on tarkoitus sisällyttää vain avoimen lähdekoodin kirjastoja. Olkoon sitten niin.

Luultavasti lähden toteuttamaan seuraavaksi myös Jörnungandr:iin fysiikkaa, kunhan vain saan animaatiokoodin loppuun. Tässä pätee vanha sääntö: Jörnungandr:ia koodatessa käytettyjä oppeja voi hyödyntää leipätyön helpottamiseksi (ja aivokapasiteetin vapauttamiseksi parempaan käyttöön), mutta toiseen suuntaan koodia ei saa liikua.

Päivän loppuessa minulla on yksinkertainen Bullet-integraatio valmiina. Oikeastaan minua itseänikin ihmetyttää sen helppokäyttöisyys, ja sellaista tapahtuu todella harvoin: yleensä löydän kirjastosta kuin kirjastosta jotain valittamisen aihetta.

Toisin kuin ODE:a käyttäessä, jolloin ohjelmoijan vastuulle jää ohjelmoida ns. törmäyscallback, joka tutkii törmänneet kappaleet ja luo tarvittavat kontaktipisteet kappaleiden välille (sentään kirjaston apufunktioita käyttäen), toimii Bulletissa törmäystarkistus jumalallisen automaattisesti. Simulaatioaskeleen jälkeen voi sitten käydä läpi askeleen aikana havaitut törmäykset. Lisäksi Bullet osaa automaattisesti suorittaa interpolaatiota tilanteessa, jossa fysiikan framerate on pienempi kuin renderöinnin. Kirjasto lisäksi väittää pystyvänsä (ainakin osittain) hallitsemaan ns. jatkuvaa törmäystarkistusta, jolloin esimerkiksi nopeasti liikkuva pallo ei pääse tunneleittamaan ohuen seinän läpi, vaikka fysiikkaframerate olisikin pieni.

Syvempi tarkastelu osoittaa, että jatkuva törmäystarkistus on vielä suurimmaksi osaksi myytti, eikä toimi kuin tietyissä erikoistapauksissa. No, Vision3D:n kannalta se ei haittaa. Meidän fysiikkaesineemme ovat usein isoja, kömpelöitä ja hitaasti liikkuvia!

Tosin: jos nyt jokin asia näyttääkin toimivan hyvin, kertoo tosiohjelmoijan ensimmäinen pääsääntö, että luultavasti jotain räjähtää pian. Mutta katsotaan nyt. Tämä päivä parani loppuaan kohti, ja mikä parasta, on nyt ohi. Poistun paikalta.

Coming up next: Jörmungandr:in animaatiosysteemin finalisointi. Streamoutin saattaminen toimimaan kunnolla aiheuttaa kiroilua. Mutta se on välttämätöntä. En varmasti aio suorittaa verteksimorfeja prosessorilla.

Koko ominaisuus on typerä.

Mutta silti sen on toimittava.

Helvetti, tämä on jotenkin irvokas käänteistyminen: Vision3D:tä tehdessä on sotaa, mutta hommat onnistuvat,

kun taas Jörmungandr:in kanssa on sekoilua ja UDB:tä, vaikka ulkoisia häiriötekijöitä ei ole. Eihän sen näin pitäisi mennä.

Onneksi ei ole läheskään keskiyö, kun saan hommat toimimaan. Commit. Animaatiosysteemi valmis. Jospa siis nukkuisin sen kunniaksi hieman enemmän, niin ei ole viikonlopun alkaessa liian zombimainen olo.

Päivä 47.

Krieg jatkuu. En siis saavuttanutkaan lopullista voittoa eilisessä välikohtauksessa. Suresh väittää keskustelleensa itsensä Tuomaksen kanssa, ja he ovat yhtä mieltä siitä, että Vision3D 2.0:n täytyy kyetä seuraavan sukupolven fysiikkasimulaatioon, mitä se ikinä onkaan.

“Totta kai, aikanaan,” totean. “Mutta nyt on prioriteettina saada ensin perusjutut toimimaan.”

Järkevä pääarkkitehti tai projektin vetäjä tyytyisi tähän vastaukseen, ja antaisi tärkeälle resurssilleen työrauhan. Tämän kriteerin perusteella Suresh ei ole järkevä. Ihmeellinen molotus fysiikkajärjestelmän vaatimuksista vain jatkuu. Ilmeisesti pelkona on, että jos systeemiin ensin koodataan vain perusominaisuudet, se luutuu sellaiseksi, ettei kehittyneitä featureita saadakaan sisään, tai niiden jälkikäteen integrointi on hankalaa ja virhealtista.

Myönnän: tavallaan tämä on valiidi pelko.

Mutta silti jonkinasteinen luottamus olisi nyt hyväksi.

Annan hypernopeuspuheen mennä toisesta korvasta sisään, ja toisesta ulos. Samalla koodaan RigidBody-luokkaa. Siitä tulee heti kerralla sellainen, että skriptistä pääsee käsiiksi kaikkiin ominaisuuksiin. Qt voi olla ihku silloin, kun se ei ole tiellä, tai räjähdä käsiin.

“Kuunteletko sinä edes?”

“No rehellisesti sanoen en,” vastaan.

“Täytyy sanoa, että sinä olet vaikea tapaus. Toistuvasti.”

“Aivan samaa voisin sanoa sinusta.”

Tähän vastaukseen Kuumakalle ei enää keksi mitään, vaan poistuu. Jatkan koodausta, mutta ajatuksiani on selvästi häiritetty, ja UDB:tä syntyy. (Taas nullpointteriaccess ensin, ja nullpointteritarkistus vasta jälkikäteen. Helvetti!) Tosin nyt minun ei tarvitse pelätä, että olisin jonkinlainen Suresh:in suosikki. Päinvastoin.

Noin puolen tunnin päästä rauhaani häiritään jälleen. Tällä kertaa Jarmo on asialla. Tuleekohan nyt joku puhuttelu? Onko Kuumakalle mennyt itkemään hänelle? Katson tarpeelliseksi palauttaa mieleeni vanhan, mutta edelleen käyttökelpoisen mantran:

VSP – VSV – VSP – VVV.

“Olen havainnut, että teillä tuntuu usein menevän sukset ristiin,” Jarmo sanoo istuuduttuaan viereeni.

Tell me about it! Osoitat erittäin terävää havaintokykyä.

“Yrittäisit hieman pehmentää ulosantiasi. Siis varmastihan sinulla on vahva tekninen pointti, mutta se on yksi – tai pitäisikö sanoa vain yksi – näkökanta muiden joukossa.”

Minun? Minunko pelkästään? Enhän minä ole se, joka pajattaa minuuttitolkulla omaa näkökulmaansa ilman mitään kykyä ottaa toisen mielipidettä huomioon.

“Niin no, Suresh voisi myös tulla tuossa omalta osaltaan vastaan,” vastaan äänensävyssäni suurempi tyylyys, kuin itse sanat antaisivat ymmärtää. “Kenen idea oli yleensä, että hänet piti tuoda tätä soppaa keittämään? Kenen henkilökohtaisesta poliittisesta pelistä tässä on kyse?”

Omat sanani saavat vihani kiehumaan vahvemmin. Huomaan korottavani ääntäni tarkoituksellisesti, jotta kaikki lähistöllä varmasti kuulevat.

“Ssh. Rauhoitu. Äläkä puhu täällä tuohon sävyyn.”

“No vittu aivan varmasti puhun, jos joudun sietämään tällaista PASKAA päivästä toiseen tästä eteenpäin. Tiedätkö mikä tässä on hassuinta? Siis aivan varmastihan projekti voisi edetä paremminkin, ja olisi voitu tehdä järkeväm-

piä päätöksiä. Minä ensimmäisenä myönnän tämän – enkä itse edes tehnyt niitä päätöksiä, saanen muistuttaa. Mutta mitä minä näen, homma kuitenkin etenee koko ajan, ja pääsee jopa melko lähelle tavoitteita, jotka sprintti kahden alussa asetettiin. Mutta nyt tämä multithreadaus- ja STM- ja molekyyllitason simulaatiopaska, millä tavalla se auttaa? Kerropa minulle se.”

“Suresh on täällä varmistamassa, että Vision3D etenee kansainvälisestikin kilpailukykyiseen suuntaan. Jos pyrimme vain toteuttamaan uudestaan sen, minkä vanhat versiot jo toteuttivat, mitä etua olemme silloin saavuttaneet?”

Jaha. Vai näin. Että meidän tarkoituksemme ei ollut kilpailla tripla-A-pelimooottorien kanssa, mutta nyt kun asian tuntija kaukomailla tulee meille kertomaan, miten multithreadataan, niin nyt me jostain kumman syystä olemmekin yhtäkkiä suuntaamassa siihen sarjaan. Mutta huomaa: ei suinkaan esim. Direct3D10:n, Shader Model 4:n, Scaleformin ja PhysX:n avulla, vaan Qt:n ja OpenGL:n. Ja software transactional memoryn!

Haista sinä kuule pitkä paska. En kuitenkaan sano tätä ääneen. En itseasiassa vastaa yhtään mitään, jolloin Jarmo poistuu muutaman vaivautuneen hiljaisuuden täyttämän sekunnin jälkeen.

Jatkan koodausta raivosta edelleen kiehuun. Nyt olen siinä pisteessä, etten enää tuota virheitä tai UDB:tä, vaan kaikki loksahda ikäänkuin vihan punaisen sumun läpi kohdalleen.

Tämä on selvästikin haastetaso kaksi piste nolla.

Uskon, että enemmänkin avoimia konfrontaatioita on luvassa. Mutta jossain välissähän pinnan alla muhivan paskan on noustava pintaan ja räjähdettävä käsiin!

Onneksi viikonloppu tulee.

Päivän lopulta päätyttyä tarkastelen, mitä tulikaan tehtyä: Bulletin integraatio, fysiikkamaailman ja rigidbodyn perustoiminnallisuus. Olosuhteisiin nähden tyydyttävän paljon. Paljon silti jää ensi viikolle: raycast fysiikkamaailmaan, törmäysevennit, constraintit (jos niitä nyt yleensä

tehdään tässä vaiheessa) ja ennenkaikkeaa character controllerin toteuttaminen, ja testaaminen, että hahmon liikuttelu toimii riittävän hyvin: ettei se putoa lattiasta läpi tai juutu seiniin tms.

Kun saavun asunnolleni, viha jäytää yhä mieltäni. Tai oikeastaan nyt se leimahtaa täyteen liekkiin uudestaan. Laitan Enochian Crescent:iä soimaan.

Black Flame of SATAN Burning!

Sekään ei riitä tynnyttämään raivoani. Jörmungandr'in koodausta ei voi edes harkita. On lähdettävä kaupungille nollaamaan päätä.

Päivät 48 - 49.

Kooman ja krapulan laannuttua hieman lähden heti koodaamaan Jörmungandr:in fysiikkatoteutusta. Vieläkin suuttuttaa, kun ajattelee, mitä loppuviikolla tapahtui töissä, mutta ei auta jäädä viivyttelymään. Nyt sentään toteutan fysiikkaa täysin omilla – tai siis Jörmungandr:in ehdoilla – ilman, että täytyy pitää mielessä mitään uskomattoman typeriä future proofing-suunnitelmia!

Lupaen syödä hiirimattoni, jos Vision3D:ssä on vaikka vuodekkaan päästä mitään insinööriyöhön etäisestikään hyväksyttävää fysiikkalaskentaa. Puhumattakaan mistään atomeilla pelleilystä, ellei nyt sillä tasolla, että demonstroidaan vaikka Bohrin atomimallia esittämällä ydin ja sitä kiertävät elektronit erillisinä fysiikkaobjekteina. Olen aivan varma, että samat rigidbodyt ne siellä tulevat pyörimään.

Tai ehkä jotain softbody-juttuja, niinkuin patjat Half Life 2:ssa. Niillä ei kyllä lopulta ollut pelin kannalta paljonkaan merkitystä.

On outoa toteuttaa samat jutut toiseen kertaan näin lyhyen ajan päästä. Millä tavalla koodi differentioituu edukseen? Se etu minulla ainakin on, että koska sekä renderöinti- että fysiikkakoodi on omaani, on helppo tehdä tarvittavat laajennukset mihin tahansa kohtaan scenenode-luokkahierarkiaa. Tosin onhan Vision3D:n scenenode-koo-

di myös omaa käsialaani, itse renderöinti vain ei.

Ero ei siis ole suuri.

No, katsotaan nyt. Oma eventtijärjestelmäni taatusti on ainakin nopeampi, kuin Qt:n signal-slot -toteutus. Joten vaikka törmäyksiä tulisi paljon yhden simulaatioaskeleen aikana, olisi eventtien lähettäminen jokaisesta tehokasta.

Olen paranoidilla tuulella. Integroidessani Bullet:ia toiseen kertaan etsin pienimpiäkin puutteita tai outouksia. Vielä ei mitään huolestuttavaa, mutta en silti pysty rauhoittumaan.

Ei mene kauan, kun ensimmäinen puute löytyy (tähän en vielä päässyt Vision3D:ssä.) Bullet ei suoraan tue liikkuvia trianglemeshejä. Täytyy toteuttaa convex hull:in laskeminen, mikä visuaalisesti näyttää vähän siltä, että objekti on kääritty kelmuun.

No niin, nyt ne liikkuvat.

API:sta paljastuu myös huomattavia poikkeavuuksia eri luokkien välillä. Ilmeisesti johtuu siitä, että tekijöitä on ollut monta (tai onhan jo ODE:ssakin useita kontribuuttoreita, mutta se pysyi silti pitempään yhden miehen show:na.)

Mutta eivät nämä puutteet pahasti haittaa. Muistaakseni Havok tai PhysX:kään eivät suoraan tue liikkuvia meshejä.

No niin, nyt liikkuu paljon objekteja. Ei vie paljon prosessoritehoa, eikä tule UDB:tä tai ODE Internal Error-virheikkunoita. Olen päällisin puolin tyytyväinen.

Päätän tehdä jotain kieroaa: en hio nyt Jörmungandr:in fysiikkaa niin pitkälle, että se tukisi esim. kaikkia constraint-tyyppisiä, tai viimeisen päälle toimivaa character controlleria. Sen sijaan jätän sen tähän tilaansa, ainakin hetkeksi, ja lähden tekemään jumalallistakin jumalallisempaa verkkokoodia.

Tosin sen aika on ensi viikolla.

Päivä 50.

Back to war. Nyt on neljä päivää aikaa saattaa Vision3D jontenkin järkevään tilaan. Minun osaltani se tarkoittaa fysiikan saamista suunnilleen feature-completeksi.

Nyt töissä vallitsee kireä ja odottava ilmapiiri, ehkä kaikki odottavat, milloin tapahtuu seuraava huudontäyteinen yhteenotto minun ja Suresh:in välillä.

Kauan sitä ei tarvitse odottaa.

Olen aivan rauhassa testaamassa käyttäjän hahmon liikettä EduWorld-scenessä. Koska skriptit eivät vielä toimi niinkuin pitäisi, hahmo materialisoituu ilmaan, ja putoaa maaston päälle. Kitka- ja kimmoisuusparametrejä ei ole tunetettu loppuun asti, joten pudottuaan hahmo pomppaa ilmaan lyhyeksi aikaa.

Suresh sattuu näkemään tämän, ja taas mennään. Pelon- ja raivonsekainen valitus fysiikan jämähtämisestä last-gen -asteelle alkaa välittömästi.

“Tämä on work in progress, ymmärrätkö. Se ei ole vielä valmis,” yritän selittää. Tai siis Jens yrittää selittää ja rauhoitella tilannetta.

Se ei auta. Suresh änkeää väkisin näppäimistöni ääreen ja alkaa säätää fysiikkaparametrejä, samalla kun raivonsekainen litania jatkuu.

No säädä sitten. Minä poistun hakemaan kahvia.

Kun palaan, hän on vieläkin säätämässä. Nyt hahmo

pomppii vieläkin pahemmin, ja itseasiassa kaatuu kokonaan, eikä pääse pystyyn.

“Katso nyt! Tämä koko homma on fail! Tämä täytyy kirjoittaa uudelleen, hahmoa ei saa käsitellä capsulena vaan bone-hierarchyyna! Niinkuin ragdoll mutta jäykempi ja käyttäjän ohjattava! Sitten jos se vaikka seisoo seinän vieressä, sen käsi ei silti clippaa sisään!”

“No sellaista ei taatusti neljässä päivässä tehdä. Otitko sieltä pois setAngularFactor (0,0,0)? No silloin hahmo taatusti kaatuu, kun siihen pääsee vaikuttamaan vääntömomentti!”

Tämä on vielä ns. rauhallinen vastaus. Mutta vihani alkaa yltyä, kuin R-Type:n superlaser, joka ensin ladataan täyteen, ja sitten laukaistaan.

“Vapausasteita ei saa ottaa keinotekoisesti pois! Sensijaan täytyy laskea tasapainottava voima, tai toteuttaa inverse kinematics-kontrolleri...”

“Näytä sitten PERKELE minulle miten teet sellaisen tässä ajassa!”

Äänenpaineeni, erityisesti perkele, on taatusti yli yhdeksänkymmentä desibeliä. Kohta olen valmis lyömään. Enkä mitä tahansa lyöntiä, vaan joka suuntaan vaikuttavan omni-punchin!

Suresh ei ilmeisesti ota sanojani heittona, vaan valitsee ilme tiukkana Visual Studio:sta Add → New Item. Kaksi kertaa.

IKController.h
IKController.cpp

Hän ei aikaile, vaan alkaa kirjoittamaan suoraan luokkamäärittelyä. Ilmeisesti on tarkoitus, että IKController aktivoituu aina joka simulaatioaskeleen jälkeen, ja korjaa hahmon asennon. Se lisäksi initialisoidaan hahmon skeletonilla. Saatuaan luokkamäärittelyn valmiiksi hän nousee tuoliltani.

“Tarkempi speksi tulee sähköpostilla. Siitä et sitten

poikkea. Ei mitään capsulea.”

Herään herätyskellon piipitykseen.

Se olikin unta.

Ei sentään tarvitse tehdä IK-kontrolleria Suresh:in määräysten mukaisesti. Mutta olipa kammottavan todentuntuinen uni.

Todellinen päivä, vaikka teenkin samaa asiaa, sujuu hie-
man rauhallisemmin. Kireän ilmapiirin tosin ennustin ai-
van oikein, ja Suresh käy vilkuilemassa työtäni vähän väliä.

Kuten unessa, saan siis hahmon liikkumaan EduWorld-
maailmassa päällisin puolin oikein. Havaitsen tosin inhot-
tavan “featuren” Bullet:issa: aina kun hahmo ylittää staat-
tisen geometrian polygonirajan, se hyppää vähän ilmaan.
Tutkin Bullet:in foorumeita, ja ilmenee, että kyseessä on
ns. internal edge-ongelma, joka on ollut kirjaston käyttäjil-
le hyvin sitkeä ja pitkäkestoinen pulma. Hahmon liikkues-
sa pitkin kolmiomeshiä kontaktinormaali saa aina raja-
kohdissa melko sattumanvaraisia arvoja.

Suresh huomaa pomppimisen myös. Alkaa jurputus.

“Siihen pitäisi löytyä tapa korjata. Ns. Internal Edge Uti-
lity,” vastaan tyllysti, mutta asiallisesti.

“Paras löytyä,” hän toteaa nuivasti ja kävelee pois.

Kyseinen työkalu on esiprosessori, joka tutkii, mitkä
polygoniedget ovat ns. sisäisiä, eli niiden kanssa voidaan
pakottaa, että käytetään kolmion normaalivektoria kon-
taktinormaalina. Kun prosessoin kollisiomesheinä käytet-
tävät 3D-mallit sillä, ongelma poistuu. Hyvä niin.

Mm. kollisioeventit ja constraintit jäävät pakostakin
huomiseksi. Vaikkei varsinaista pahempaa kriegiä ollut-
kaan, päivä oli silti vaanivan inhottava. Uni pyörii mieles-
säni vieläkin. Hyi!

Kun lähden töistä, mietin, että minulla pitäisi olla oma cei-
ling cat työpisteeni yläpuolella, joka tulisi tarvittaessa kat-
toluukusta esiin. Sitten voisin tähdätä sen silmälaserialla
Suresh:ia tai Jarmoa tai ketä tahansa, joka uskaltaisi häiri-
tää!

No niin.

Nyt on siis vuorossa ehkä Jörmungandr:in toiseksi pyhin osuus renderöinnin jälkeen.

Verkkokoodi.

Tulee mieleen, että minun pitäisi lähestulkoon pukeutua kaapuun, sytyttää mustia kynttilöitä, ja piirtää pentagrammi suojaksi lattiaan. Sen verran black art-meiningistä voi pahimmillaan olla kyse.

Mielenkiintoiseksi asian tekee, etten ole raakaa verkkosocket-ohjelmointia juurikaan harrastanut.

Suunnittelen ensin UDP-pakettiheaderin teoreettisella tasolla.

Siitä tulee hyvä. Voi pojat, siitä tulee hyvä.

Vertailun vuoksi lataan ENet:in sorsakoodin, vaikken sitä missään tapauksessa aio käyttää. Kyllä: on selvää, että ENet header-overhead on kaikilla kombinaatiolla suurempi, kuin omani. Kombinaatiothan ovat seuraavat:

Reliable / in-order

Reliable / not ordered

Unreliable / in-order

Unreliable / not ordered

Voitto on minun. Nyt vain koodataan.

Tuntien uurastuksen jälkeen saan lopulta UDP-paketteja liikkumaan ns. serveriohjelman ja client-ohjelman välillä. Täytyy myöntää, että nyt minusta ei todellakaan tunnu tosiohjelmoijalta, vaan pahaiselta aloittelijalta.

Mutta nyt, toivon mukaan, kun tämä matalan tason välttämätön paha on toteutettu, voin keskittyä verkkokoodin ylemmän tason logiikkaan. Siellä inhottavia yllätyksiä ei pitäisi tulla – tai jos tulee – ne johtuvat vain oman verkko-protokollaspesifikaationi puutteista tai suoranaisista virheistä. Verkko vastaa niinkuin sinne huutaa. Tämä lähentelee jonkinlaista karman lakia, tai ehkä jopa uskonnollista hartautta. Tosin en varsinaisesti ole uskonnollinen, ellei siksi lasketa sitä, että kuvittelen joskus olevani Suuren Pelisuunnittelijan, tai sitten ceiling cat:in alaisuudessa.

Mutta ylemmän tason logiikkaa en suinkaan aio lähteä toteuttamaan, kun kello on jo yksi yöllä. Voimia täytyy säännöstellä, jotta jaksaa taas huomenna mahdollista kriegiä.

On muuten mielenkiintoista, miten kriegiin tavallaan tottuu: se, mikä viime viikon alussa tuntui helvetin yhdeksänneltä piiriltä, on nyt vähitellen normalisoitunut siten, että sitä odottaakin saavansa joka päivä paskaa niskaansa. Kuinka helppoa elämä olikaan, kun oli vain Jarmo kestettävänä!

Mutta nyt ei mietitä tällaista shittiä.

Nyt nukutaan.

Päivä 51.

Hyi helvetti. Kun kollisioeventtejä alkaa mennä Qt:n välityksellä signaaleina, sen todella huomaa profilerissa. Kun lisäksi vielä skriptit alkavat käsitellä eventtejä, sen huomaa vielä paljon voimakkaammin.

Vision3D 2.0 tulee olemaan paljon CPU-kuormittavampi, kuin 1.4! Helvetti! Ehkä tässä sittenkin tarvitaan threausta.

No joo, jos vanhat versiot käyttivät lua-skriptikieltä, niin siitä onkin aika hankala pistää nopeammaksi. Tosin myönnän kyllä ensimmäisenä, että ECMA- tai siis käytännössä javascriptin syntaksi on paljon miellyttävämpi. Ja lua:ssa arrayt alkavat indeksistä 1, mikä on anteeksiantamaton synty. Nyt sentään lasketaan nolasta alkaen, niin kuin oikein on.

Mutta hintana on hitaus.

Tämä johtaa minut väistämättä miettimään, minkä skriptikielen valitsen Jörmungandr:issa. Joskus sekin valinta on edessä. Itseasiassa aika pian verkkokoodin jälkeen.

Ylimaallisin vaihtoehto olisi tietysti koodata itse. Mutta en tiedä, haluanko kuluttaa siihen aikaa, jos aion ennen helmikuuta (eli silloin, kun Vision3D 2.0:n pitäisi teoriassa olla valmis sisäiseen käyttöön) demonstroida Jörmungandr:in ylivoimaa, ja täten valmiiksi saattaa kostoni.

Valve:n Left for Dead käyttää Squirrelia. Frictional Games käyttää Angelscriptiä. Crytek käyttää edelleen luaa. Aika moni käyttää pythonia. Vaihtoehtoja on, mutta yleensä kaikkien bindaaminen C++:aan on ikävää.

Kaarlo yllättää minut kesken pohdintojeni.

“Mikäs miehellä, kun näppäimet eivät sauhua?”

“Prkl,” mutisen vastaukseksi, ja alan muka koodata.

Niin totta. Se performanssi-issu. Tällainen prosessori-ajan kulutus, ja näin yksinkertaisella scenellä, ei ole hyväksyttävissä. Sen verran minulla on ylpeyttä jäljellä kriegistä huolimatta. Tai ehkä juuri siksi. Koska Suresh:in ratkaisu on, että threadattaisiin kaikki, haluan kynsin ja hampain pitää yhden säikeen pääsilmutkan nopeana!

Katsotaanpa.

Lisätään flagi, että lähetetäänkö tälle objektille törmäysevennentejä ollenkaan, ja jos lähetetään, niin mahdollistetaan niiden määrän rajoittaminen. Ja skriptin puolella otetaan vastaan vain, jos oikeasti on merkitystä. Huom. enää ei ole kömpelöä ScriptedObject-pakkoa, vaan missä tahansa entityssä voi olla monta skriptiä ajossa, jos niin haluaa. Mieluiten toki ei yhtään.

Näillä muutoksilla CPU-kuorma putoaa neljännekseen. Niin harvinaista kuin se onkin, olen tyytyväinen.

Aprikoosikiisseli ja vaniljavaahto katkaisevat työpäivän. Aamulla kunnon sotaa ei ollutkaan, katsotaan seuraako sitä iltpäivällä.

Seuraava taskini on constraintit ja nivelet fysiikka-alijärjestelmään. Mutta siinä välissä tarkistan vielä, että EduWorld:in hahmo liikkuu hyväksyttävästi. Kyllä, nyt se taatusti liikkuu. Ei nykimistä, ei pomppimista. This is true power.

Minulla on eräs teoria, jonka haluan testata.

Se testataan seuraavasti: värjätään collisionevent-taski backlogista vihreäksi.

Teoria vahvistettu: kuin kellokoneiston tarkkuudella Suresh saapuu minua vainoamaan. Hänellä on siis todellakin backlog-dokumentti jatkuvasti auki. No niin, tämä on

ehkä hyvä tietää jatkon kannalta.

Joo joo, constraint-järjestelmästä tulee paras ja monipuolisin ikinä. CPU-tehon rajoissa on mahdollista skriptata siihen vahva ja heikko vuorovaikutus, jos todella on niin sekopää, että haluaa tehdä sen.

Jens pistää parastaan. Tavallaanhan tämä on heikkoutta, koska myötäilyn sijasta pitäisi sotia ja kyräillä. Mutta jos nyt pääsisi vaikka koodaamaan, niin olisi ihan positiiivista.

Suresh poistuu.

Katsotaanpa.

Totean, että Bulletin eri constraint-luokkien rajapinnat poikkeavat toisistaan huomattavasti. Tämä ei ole esimerkiksi serialisoinnin kannalta mukavaa.

Hammasta purren toteutan rajapintaa yhtenäistävän wrapper-toteutuksen, josta tulee siis Vision3D:n constraint-komponentti. Se käytännössä luo konepellin alla tarvittavan Bulletin constraint-olion, ja liittää sen rigidbodyihin.

Ei se kaunis ole, mutta toimii.

On hankala tietää täysin, mitä toiminnallisuutta constrainteilla pitäisi olla, kun EduWorld (tai mikään Vision3D 1.4:llä tai vanhemmalla tehty sovellus) ei niitä varsinaisesti käytä. Mutta pistetään sinne nyt jotain. Joint-moottorit ja muu shitti. Testaan tehdä saranoillaan kääntyvän oven, joka toimii täysin hyväksyttävästi.

Kaukoviisaasti en kuitenkaan merkitse taskia tehdyksi, koska epäilemättä huomenna mieleen tulee vielä jotain säätettävää. Mutta kahdeksan tuntia on nyt täynnä. On erittäin hyvä, että ainakaan tässä vaiheessa, ja ainakaan minulta, ei vaadita crunchia. Mutta eiköhän senkin aika vielä tule.

Poistun. Valkoinen Kostaja kuljettaa minut demoniseen linnoitukseeni, jossa todellisten maailmanvalloitus suunnitelmieni toteuttaminen jatkuu Jörmungandr:in verkkokoodin muodossa. Hm, hm, hm, hm, ha, ha, ha ... !

Mietitäänpä. Jos minä olen NES:in Ninja Gaiden-pelien

Jaquio tai Ashtar, niin kuka ihme Jarmo tai Suresh on? Hyi. Ei tuollaista halua mieltää. Ehkä ennemmin niin, että jos minä olen Jacquio, niin Kaarlo voisi olla Ashtar. Kaksi demonilordia, jotka kyräilevät toisilleen aina tilaisuuden koittaessa, mutta joiden välillä kuitenkin vallitsee jonkinlainen – vaikka sitten pakotettu – yhteisymmärrys.

Mutta nyt siis korkean tason protokollaa kirjoittamaan. Kerrosta, joka toteuttaa UDP:n päällä luotettavan tai epäluotettavan viestinnän.

Verkkokoodia on erittäin turmiollista testata localhostia käyttämällä. Tämän vuoksi ajan ns. serveriä työasemalani, ja clienttejä kannettavalla ja versionhallintapalvelimella.

Aivan kammottavia logiikkavirheitä ja undefined behaviouria paljastuu lähes heti, kun testaa todellisen verkkotransportin yli.

Hyi, hyi, hyi!

Kamalaa.

Pois tällainen.

Luotettavien viestien acknowledge-systeemi täytyy mieltää täysin uusiksi. Tämä ei ole ollenkaan hyvä. Pakettien headerin tavumäärä kasvaa. Toivottavasti ei vielä mennä ENet:iä raskaammaksi. Se olisi tappio!

UDB jatkuu muutoksistani huolimatta. Alkaa olla myöhä. Myönnän tappioni tältä erää.

Päivä 52.

Constraintit ovat valmiina. Yleisesti ottaen tämän sprintin backlog on noin 75-prosenttisesti tehtynä. EduWorld pyörii jotenkuten, jopa usealla käyttäjällä: Kaarlo on huhkinut kunnolla verkkokoodin kanssa. Tosin toiminnallisuutta ei vielä varsinaisesti ole. Hannele puolestaan toteuttaa Qt:n päälle ns. pelimäistä UI-ulkoasua, ja kiroaa raskaasti.

(Eikä todellista fullscreen-moodia ole edelleenkään.)

Sebastian on saanut mandaatikseen forward renderöinnin toteuttamisen, mutta on selvää, ettei se voi mitenkään valmistua kahdessa päivässä. Tämä vaatimus on käsittääkseni tullut Suresh:ilta. Huvittavaa tavallaan. Hyvinkin ironista.

Mutta yhtä hyvin minulta voitaisiin vaatia IK-hahmo-kontrolleria! Täytyy siis pitää tämän suhteen naama peruslukemilla.

Jaa, mitäs minä tekisin? Fysiikan raycast on vielä tekevä. Se siis.

Yksinkertaista ja kaunista, toisin kuin ODE:n kanssa, jossa olisi täytyntä kirjoittaa toinen törmäyscallback.

Testataanpa taas. Taski vihreäksi.

Tämä summonoi Kuumakallen paikalle lähes nollan sekunnin viiveellä.

Mitäs nyt? Tämä onkin mielenkiintoista.

Kun kuulen tuomioni, tulee sellainen olo, että olisihan

minun pitänyt aavistaa:

- *Fysiikan threadaaminen.*
- *Softbody-fysiikka.*

Olen kyllä melkoinen thetaani, jos kirjoitan nämä yhdessä ja puolessa päivässä. No, en usko, että sitä minulta varsinaisesti odotetaankaan, mutta sittenpä ainakin tietää, mitä jatkaa joulun jälkeen.

Fysiikan threadaus. Iljettä jo valmiiksi.

Oloani ei suinkaan tee paremmaksi ymmärrys, että jos saisinkin sen toteutettua Vision3D:hen, ja olisin näin tullut paremmaksi ohjelmoijaksi, ei samaa ideaa voisi sellaiseen siirtää Jörmungandr:iin, koska se olisi jo saastunut. Jörmungandr:in pitäisi aina nostaa rimaa jollain tavalla vielä ylemmäs.

Päivä 53.

Yöllä kolmen aikaan sain lopulta selvitettyä Jörmungandr:in verkkokoodia vaivanneen sekoilun. Nyt protokolla kai sitten toimii, mutta maksoin siitä raskaan univelan. No, tuleehan joulu edes väliin.

Jouluksi kai sitten pitäisi mennä käymään kotona. Se ei ole kovin mieltä ylentävää. Pitää muistaa ottaa kannettava mukaan, ja koodata käytännössä Jörmungandr:ia koko aika.

Mutta sitä ennen vielä yksi päivä Vision3D:tä.

Fysiikan threadausta.

No, kyllähän sen voi threadata, karkealla lockstep-menetelmällä. Törmäyseventit kuitenkin tulevat vasta simulaatioaskeleen jälkeen, ja Bullet käyttää täysin omaa tietorakennettaan, poislukien `btMotionState::setWorldTransform()`, jossa se haluaa kertoa renderöintijärjestelmälle objektin interpoloidun paikan. Sen taas voi puskuroida yksinkertaisesti.

Ok. Fysiikka on threadattu.

Minä poistun. Pahaa ja kyräilyä täyteistä joulua kaikille.

(Koska minulla ei ole aivan riittävä määrä vapaapäiviä varastossa, joudun käymään yhden päivän joulun ja uudenvuoden välissä töissä, mutta se ei ole paljon.)

NELJÄS KIRJA – TULI

RETSU – Ajan ja avaruuden herruus

Päivä 54.

Saan joululahjaksi viinapullon ja morapuukon. En voi olla hymyilemättä. Tosin myönnän, että jotenkin houreisesti olin kuvitellut saavani puukon sijasta katanan. Mutta ehkä sitten ensi vuonna. Jos olen vielä hengissä.

Esittelen kuji-kiri -käsimerkkejä siskolleni. Tämä saa riittää sosiaalisuudesta. Jörmungandr kutsuu.

Alan kirjoittamaan komponenttien verkkoserialisointia ja deserialisointia. Verkkopäivityksissä tulee tietysti lähettää vain muuttuneet jutut; vertailuun voi käyttää joko käyttäjälle lähetettyjä viimeisiä arvoja, tai niiden puuttuessa oletusarvoja (esim. paikka 0, 0, 0 – identity-rotaatio – skaala 1, 1, 1.)

Käyttäjälle lähetettyjä viimeisiä arvoja?

Mistä ne tiedetään?

Tiedän tavallaan jo ratkaisun, ennen kuin ajattelen sitä tietoisesti. Ja kyllä: tämä on kanoninen tapa, jota esim. Quake 3:n verkkokoodi käyttää. Mutta silti se on kamala. Jokainen clientille lähetetty scenerevisio talletetaan, ja niitä voidaan poistaa sitä mukaa, kun clientiltä tulee tieto, että se on vastaanottanut kyseisen tai uudemman revision. (Scenepäivitykset lähetetään tietysti epäluotettavina, koska ne sisältävät nopeasti vanhenevaa tietoa.)

Seurauksena kyseisestä menetelmästä on, että serverin muistinkäyttö saattaa kasvaa huomattavasti, varsinkin jos

päivityksiä lähetetään tiheästi, ja jos klienttejä on paljon. Tämän tosiasian ajattelemisen masentaa minua niin paljon, että pistän kannettavan kiinni ja siirryn ryyppäämään sukulaisteni kanssa.

Päivä 55.

Jos joulukirkko pidettäisiin lolcat-raamatun mukaisesti, menisin sinne ehdottomasti. Mutta koska ei, jätän väliin. Lisäksi minun vain täytyy todeta, että kotini ilmapiiri ei ole konduktiivinen tosiohjelmoinnille, joten palaan kaupunkiin.

Hammasta purren – synkkä muistinkäyttöennuste koko ajan mieltä kaivelemassa – toteutan client/server -kommunikaatiota sille asteelle, että saan scene-entityt repli-koitumaan. Protokollan tukemat operaatiot ovat nyt seuraavat:

- *Entityn luominen.*
- *Komponentin lisääminen entityyn.*
- *Komponentin arvojen muuttuminen.*
- *Komponentin poistaminen entitystä.*
- *Entityn poistaminen.*

Totean, että jatkon kannalta tämä ei vielä aivan riitä: lisäksi tarvitaan jonkinlainen RPC (remote procedure call) -mekanismi clientiltä serverille, ja päinvastoin. Käyttäjän lähettämät ohjauskomennot (esim. hiiri ja näppäimistö) voidaan lähettää RPC:inä.

Mietin vaihtoehtoisia tapoja scenepäivitysprotokollalle, joissa ei tulisi muistinkulutusongelmaa, mutta en keksi mi-

tään, jota voisin pitää riittävän eleganttina. Koska matalan tason UDP-viestintäni tukee myös luotettavia ja järjestyksessä tulevia viestejä (aivan TCP:n tavoin niitä joudutaan viivästäämään ja keräämään puskuriin, jos ne tulevat epäjärjestyksessä), voisin tietysti lähettää entityjen ja komponenttien luontikomennot niillä, ja komponenttien arvojen muuttumisen tilanteesta riippuen luotettavana tai epäluotettavana.

Tätä ei todellakaan voisi pitää eleganttina. Luultavaa olisi, että luotettavat ja epäluotettavat päivitykset menisivät keskenään epäsynkkaan, ja synnyttäisivät harvinaisia ja / tai vaikeasti havaittavia virhetilanteita.

Joo, todetaan, että Carmack tiesi paremmin, kuin useimmat. Jo valitsemani menetelmä on vähiten huono.

Päivä 56.

Scenereplikointi on kunnossa. Fysiikka on kunnossa. On siis aika nostaa haastetasoa, ja yhdistää nämä: kokeilla, miten fysiikan saa verkotettua. Tässä nyt en puhu pelkästään siitä, että serveri lähettää fysikaalisesti liikkuvien objektien paikkapäivityksiä clientille (tässähän ei oikeasti ole mitään ihmeellistä), vaan siitä, että client myös laskee fysiikkaa eteenpäin niille objekteille, joita itse kontrolloi. Tämä vaatii täsmälleen saman ohjauskoodin ajamista clientillä ja serverillä, mikä ei suinkaan ole Jörmungandr:ille ongelma.

Esivalmisteluna asialle toteutan headless server-toimintatilan. Havaitseen joutuvani korjaamaan suuren määrän crashibugeja ja UDB:tä, sillä olin tehnyt häpeällisiä oletuksia siitä, että Direct3D10-device tulee aina olemaan olemassa. Joudun kirjoittamaan inhottavan "CPU-emulatio" -codepathin mm. verteksipuskureille: ne vain varastoiivat mallin geometrian esimerkiksi sitä varten, että serveri, joka ei renderöi, kuitenkin pääsee esimerkiksi fysiikkaa varten dataan käsiksi.

Nautitsemani viski hidastaa työtä ehkä jonkin verran. Lopulta kuitenkin epäkurantti koodi on eliminoitu tai muutettu hyväksyttävämpään asuun, eikä sekoiluja enää ilmene.

Headless Jörmungandr server on syntynyt!

Päivä 57.

Päätän hoitaa epämiellyttävät asiat heti pois. Siis tänään töihin, niin saa pitää seuraavat kuusi päivää vapaata.

Toimisto on mukavan pimeä ja tyhjä. Vain Veli-Matti on paikalla, ja koodaa jotain hörhöjuttua (onkohan edes Vision3D:tä, vai muuta projektia, en tiedä enkä välitä.)

Lähden toteuttamaan softbody-fysiikkaa, tosin en millään erityisellä innolla. Ikäväkseni havaitsen, että tätä varten pitää luoda omanlaisensa fysiikkamaailma. Hyvä sentään, että se periytyy jäykkien kappaleiden maailmasta, eli btDiscreteDynamicsWorld:ista. Mutta saa nähdä, tippuuko suorituskyky, vaikka pehmeitä kappaleita ei olisikaan.

Saan deformatiivisia palloja pomppimaan maailmassa. En kuitenkaan merkitse taskia vihreäksi, siltä varalta, että Suresh välittömästi teleporttautuisi paikalle minua vainoamaan. Uudenvuoden jälkeen voi vielä testata, ja merkitä sitten.

Kahdeksan tuntia (suunnilleen) täynnä. Se on moro.

Alan toteuttamaan prediktoivaa verkkofysiikkaa Jörmungandr:iin. "Testisovellukseni" ei ole kummoinen: teepannu jota voi ohjata laatikkomaailmassa. Mutta se tulee osoittamaan kaikkien peruseräiteiden toiminnan!

Pahoja aavistuksia alkaa tulla mieleeni. Client-side predictionin toimintaperiaate on seuraava: (oletetaan esim.

100 millisekunnin lagi, ja että sekä client että serveri viestivät toisilleen 30 kertaa sekunnissa, vaikka fysiikkaa ja renderöintiä tietysti ajetaan tiheämminkin.)

- *Client lähettää serverille kerran verkkoframen aikana kontrollipaketin (esim. hiiren liike + alhaallapidetyt kontrollinäppäimet) ja tallettaa lähetetyt paketit myös itselleen muistiin.*

- *Serveri ajaa kontrollipakettien perusteella clientin ohjaaman hahmon liikettä eteenpäin, siinä missä myös koko muuta sceneä.*

- *Serveri lähettää deltaupdaten clientille, jossa kerrotaan mm. tämän oman hahmon nykyinen fysiikkatila (esim. paikka ja orientaatio, lineaarinen nopeus, ja kulmanopeus.) Client saa tämän 100 millisekuntia jäljessä!*

- *Serverin updatepaketti sisältää myös tiedon sen viimeksi vastaanottamasta kontrollipaketista.*

- *Vastaanotettuaan updatepaketin, client kelaa scenensä aikakäsitystä taaksepäin siten, että kaikki muutokset, mitä se on itsenäisesti tehnyt viime serveriupdaten vastaanottamisen jälkeen, unohdetaan.*

- *Client applyttää updatepaketin, ja poistaa muistamiensa kontrollipakettien listasta ne, jotka se tietää serverin jo vastaanottaneen ja käsitelleen (on tietysti voinut seurata packet lossia, mutta yksinkertaistuksen vuoksi voidaan olettaa, että jos serveri sanoo vastaanottaneensa kontrollipaketin numero 100, on se saanut myös esim. paketit 97, 98 ja 99.)*

- *Client ajaa sceneään ja fysiikkasimulaatiotaan eteenpäin niin monen verkkoframen verran, kun listassa on vielä serverin käsittelemättömiä kontrollipaketteja. Tämän seurauksena clientin pitäisi nyt olla samassa aikakäsityksessä (100 millisekuntia edellä updatepakettia) kuin serveri, jos se on siis laskenut fysiikan samalla tavalla. Lag on saatu eliminoitua, eli käyttäjältä tuntuu, kuin hänen kontrollinsa olisivat viiveettömiä.*

Tässä on parikin haastetta. Ensinnäkin, jos ei pystytä takaamaan, että simulaatio etenee riittävän identtisesti, aiheuttavat serverin lähettämät uudet updatepaketit inhoitavaa nykimistä. Tästä näin unta jo kauan sitten, sillä tiedän kyllä efektin teoreettisella tasolla!

Toiseksi, clientin täytyy pystyä ikäänkuin "hetkessä" simuloimaan sceneä eteenpäin lagin verran. Tämä tarkoittaa, että mitä pahempi verkkolagi on, sitä enemmän clientille tulee CPU-kuormaa. Jos prediktoitavia fysiikkaobjekteja on paljon, voi se muodostua ylivoimaisen raskaaksi.

Kolmanneksi, packet lossin, vaihtelevan verkkoviiveen ja frame "kerrallaan" etenemisen myötä voi ilmetä jitteriä, vaikka algoritmi teoreettisesti toimiikin oikein. Tähän en vielä tiedä teoriatasolla hyvää ratkaisua, mutta käytännössä ehkä löytyy joitain sopivia smoothausalgoritmejä.

Mutta haasteet on tehty voitettavaksi. Tätä varten koko Jörmungandr on olemassa.

Päivä 58.

Jörmungandr:in client side predictionin toteuttamista. Vain vähäistä nukkumista. Kamalaa nykimistä, etenkin kun teepannu kulkee maailmageometriian polygonirajan yli. Yritän käyttää InternalEdgeUtilityä, kuten töissä EduWorld:in hahmon nykimisen estämiseen, mutta se ei auta. Client ja serveri vain tuntuvat laskevan fysiikkaa eri tavalla.

En kuitenkaan suostu myöntämään tappiotani.

Päivä 59.

Teen ... arvatkaa mitä? Aivan oikein. Jörmungandr:in client side-prediktiota ja fysiikan synkronointia clientin ja serverin välillä. Ei onnistu. Ei vain perkele onnistu.

Nykii vieläkin.

Hyi.

Päivä 60.

Useiden päivien sekoilu päättyy kerrassaan vastenmieliseen, mutta toisaalta ennalta-arvattavaan lopputulokseen: Bullet ei sovellu client-side prediction:in toteuttamiseen. Ei riitä, että palauttaisi jokaisen rigidbodyn tilan (positio, rotaatio, summattu voima ja vääntömomentti, lineaarinen nopeus ja kulmanopeus.)

Ei todellakaan.

Tämän lisäksi Bulletin syövereissä on törmäystarkistukseen ja kappaleiden välisiin kontaktipisteisiin liittyviä tietorakenteita, jotka ovat luonteeltaan epädeterministisiä. Tai siis toimivathan ne deterministisesti, jos koko simulaatio ajetaan alusta, mutta kesken kaiken niitä ei saa alustettua uudelleen sellaiseen tilaan, että halutusta ajanhetkestä alkaen client ajaisi luotettavin ja toistettavin lopputuloksin eteenpäin serveriltä saapunutta fysiikkaupdatea.

Yhdellä todella ilkeällä tavalla saan kyllä deterministisyyttä ilmenemään: jos tuhoan ja luon uudestaan prediktoitavat rigidbodyt joka frame ennen simulaatioaskelta. Tosin tästä on haittana, että painovoima lakkaa vaikuttamasta objekteihin. Debuggaan Bulletin sisään, ja applyGravity() -funktiota kutsutaan aivan oikein, mutta se ei jostain syystä vaikuta. Jos taas teen tuhoamisen ja uudelleenluonnin simulaatioaskeleen jälkeen, painovoima vaikuttaa kylä, muttei täydellä vahvuudella. Tietysti voisi laskea tarvit-

tavan korjauskertoimen, muttei voisi tietää, missä tilanteessa se olisikin väärin.

Koko menetelmä on ilkeä. Ei joka frame pitäisi tapahtua allokointia ja tuhoamista. Kyllä fysiikkamoottori pitäisi pystyä saamaan deterministiseksi muilla keinoin.

Perkele.

Mietin vaihtoehtojani. ODE:en en missään nimessä aio palata takaisin Vision3D:n vanhassa versiossa saatujen kokemusten jälkeen, Tokamak on aika kamala hardkoodattuine rajoituksineen (objektien maksimimäärä täytyy päättää simulaatiota alustettaessa), Newton sisältää suorituskykyongelmia eikä sen lähdekoodi ole avointa. PhysX:ää saisi käyttää vapaasti ja se on industry standard-tason ratkaisu, muttei myöskään kelpaa Vision3D:hen, kun kaiken pitää olla niin avointa ja vapaata. Pah!

Tosin myönnän kyllä, että minulla on Jörmungandr:issa aivan sama vaatimus, mutta hieman eri syystä: haluan päästä näkemään kaiken koodin, jotta tiedän, ettei siellä ole piilotettua UDB:tä odottamassa.

Mietitäänpä.

Oman fysiikkaenginen koodaaminen? Sitähän mietin jo aivan projektin alussa. Haastetasoa siinä olisi kyllä taatusti. Aikataulut tulisivat heittämään häränpyllyä, vaikka kaikki menisi putkeen.

Tai vielä syvemmälle debuggaaminen Bullet:iin, kunnes ymmärrän sen tekijöitäkin paremmin, mitä siinä tapahtuu. Tosin tästä ei voi tietää yhtään, kuinka kauan siinä menisi.

Tai sitten sanotaan vain tylsty:

Ei client-side predictionia Vision3D:hen.

Ei client-side predictionia Jörmungandr:iin.

Ensinmainittu ei oikeastaan haittaisi. Tällöinhän käyttäjäkokemus olisi Second Life-mäinen, jossa vain serveri laskee fysiikkaa, ja clientit tekevät vain tyhmää lineaarista ekstrapolaatiota. Suurelle osalle käyttäjistä (hörhöistä) tämä olisi taatusti aivan riittävää, varsinkin jos he ajavat verkotettuja Vision3D-sovelluksia vain lähiverkossa.

Hyi. Miksi ihmeessä minä itse mietin SL:ää? Olisinhan

ennemmin voinut ajatella alkuperäistä Quakea, jossa myös oli puhdas client / server -arkkitehtuuri ilman predictiota.

Mutta jälkimmäinen olisi sydäntäsärkevää.

Eikä oikeastaan Jörmungandr:in valmiiksi saattamiseen ole paljon enemmän aikaa, jos aion todistaa, että tämä on se suunta, johon Vision3D:n olisi pitänyt mennä. Siinä vaiheessa kun 2.0 on saatu ovesta ulos, ei ketään kuitenkaan kiinnosta, oli se kuinka kökkö tahansa, ja Jörmungandr kuinka hyvä tahansa.

Voi perkeleen Bullet ja ODE.

Mutta.

Hyvin suuri mutta.

Olisiko mahdollista, että ODE:a ohjelmoitiin väärin Vision3D:n vanhassa versiossa?

Jos se toimisi oikein, ja soveltuisi paremmin client-predictioniin, kunhan sitä toisaalta käskyttäisi teräksisellä otteella, ja toisaalta noudattaisi tarkkaan API:n kaikkia kiemuroita? Inhottavaksi asian tekee se, että Vision3D:ssä on joka tapauksessa päätetty käyttää Bullet:ia. Tätä on turha yrittää kiertää. En aio harrastaa kriegiä yhtä aikaa sekä Jarmoa että Suresh:ia vastaan, sillä tiedän, että se olisi vain viivästetty tappio, ja tällä välin olisi voinut vaikka koodata. Jörmungandr:iin taas en voi missään nimessä ottaa 1.4-version ODE-koodia, en edes soveltaakseni sitä, sillä sehän saastuttaisi codebasen heti. Sellaista ei voi edes ajatella.

Toisinsanoen, jos haluan kokeilla tätä mahdollisuutta, minun täytyy integroida ODE tyhjästä. Tämä tietää monta yötä hyvin vähäistä unimäärää.

Mutta jos ei kokeile, ei voi voittaa.

Päivä 61.

Vuoden viimeinen päivä. Tästä päivästä ei ole kovin paljon sanottavaa, paitsi eräs asia, jonka voin todeta integroituaani ODE:a Jörmungandr:iin ensin läpi yön, ja sitten tämän päivän, aina pitkälle iltaan asti.

Taistelin ODE:n kitkamallia vastaan ja voitin.

VOITIN! Vittu minä olen jumala.

(Tarkempi selitys: oletusarvoisesti Open Dynamics Enginessä kappaleiden kitka on heikoimmillaan puhtaasti X, Y ja Z-akselien suunnassa, minkä seurauksena liukuvien kappaleiden nopeus kääntyy ennen pitkää akselien suuntaiseksi. Tämä on täysin väärin. Täytyy laskea fysiikkaobjektien suhteellinen tangenttinopeus, normalisoida se, laskea vielä varmuudeksi ristitulo normaalivektorin kanssa, ja syöttää näin saatu suunta ODE:lle, joka onneksi suostuu ottamaan sen vastaan. Näin saadaan jumalallisella tavalla oikein toimiva liukumiskitka.)

Mutta nyt helvetti nukkumaan ja äkkiä.

Tässä vielä koodi kaikille, jotka joutuvat kärsimään ODE:n ikeen alla:

```
int numContacts = dCollide (geomA, geomB, MAX_CONTACTS, &contact[0].geom, sizeof (dContact));  
for (int i = 0; i < numContacts; ++i)  
{
```

```

    dVector3 velA;
    if (bodyA)
        dBodyGetPointVel (bodyA, contact[i].geom.pos[0],
contact[i].geom.pos[1], contact[i].geom.pos[2], velA);
    else
        velA[0] = velA[1] = velA[2] = 0.0f;
    if (bodyB)
    {
        dVector3 velB;
        dBodyGetPointVel (bodyB, contact[i].geom.pos[0],
contact[i].geom.pos[1], contact[i].geom.pos[2], velB);
        velA[0] -= velB[0];
        velA[1] -= velB[1];
        velA[2] -= velB[2];
    }

    float length = sqrtf (velA[0] * velA[0] + velA[1] *
velA[1] + velA[2] * velA[2]);
    if (length > EPSILON)
    {
        float inv = 1.0f / length;
        velA[0] *= inv;
        velA[1] *= inv;
        velA[2] *= inv;
        contact[i].surface.mode |= dContactFDir1;
        dCROSS(contacts[i].fdir1, =, velA, contact[i].geom.
normal);
    }

    dJointID contact = dJointCreateContact (world, con-
tactJointGroup, &contact[i]);
    dJointAttach (contact, bodyA, bodyB);
}

```

Päivä 62.

Aamulla kaikki on toisin. Ymmärrän profiloida ODE:a, ja totean, että kyllähän se on järkyttävän hidas Bullet:iin verrattuna. Epädeterminististä nykimistä tulee kyllä paljon vähemmän (ja constraint solverin warmstartin pois päältä kytkemällä sen saattaisi pystyä eliminoimaan täysin), mutta se ei paljon lohduta, jos scenen rewind ja uudelleensimulointi siihen pisteeseen, että lagi saadaan poistettua saattaa viedä vähänkin suuremmalla objektimäärällä niin kauan aikaa ja prosessoritehoa, että koko paska hyytyy. Ja tämä siis yli kahden gigahertsin prosessorilla!

Perse. Onko siis mahdotonta tehdä kunnon client side prediction fysiikan kanssa? Ehkä pitäisi koodata oma fysiikkaengine, joka olisi käsittämättömän nopea.

Lisäksi muistan, että näin pahaenteisen unen. Nyt kun se palautuu mieleeni, se ei suostu vähään aikaan katoamaan. Unessa olin kuollut, ja tarkastelin itseni tilaa ikäänkuin debuggerissa. Kuten kaatuneessa ohjelmassa, kaikki muuttujat olivat jääneet viimeisiin arvioihinsa, ja niitä pystyi tutkimaan täydellisen selkeyden vallitessa. Debuggeri itseasiassa ei ollut Visual Studio, vaan jotain paljon vanhempaa. Se muistutti konekielimonitoria.

Amigan Action Replay-moduuli! Sehän se oli.

Ja koko tilanteessa oli tietysti yksi perustavanlaatuinen pääpointti. Ohjelman suoritusta ei enää päässyt jatka-

maan.

“Kuolema – debuggeri josta et pääse jatkamaan.” Tuohan kuulostaa aika mielenkiintoiselta iskulauseelta. Mutta mille ihmeen kohderyhmälle?

Tämä on aika inhottava alku uudelle vuodelle.

Nyt suoraan sanoen tuntuu aika tyhjältä. Mitä teen Jörnungandr:iin, jos verkotettu fysiikka on osoittautunut umpikujaksi, ainakin yleisellä tasolla? Tietystihän voisi tehdä pelaajan hahmoa varten “kevyen” fysiikkakontrollerin, joka esim. vain tekisi raycastia staattiseen maailmaan, mutta se ei olisi yleispätevää. Se ei olisi jumalallista. Vaihtoehdoisesti voisin muuttaa protokollaa siten, että serveri lähettäisi korjauspäivityksiä sijaintiin vain tarvittaessa, mutta se olisi konseptuaalisesti epäpuhdasta, ja lisäksi antaisi käsittääkseni mahdollisuuden ns. speedhack:iin (koska client kertoo tällöin sijaintinsa serverille, ja serverin täytyy luottaa siihen ainakin tietyissä rajoissa.)

Koska en tiedä mitä tehdä, en tee mitään.

Sitten muistan, mitä voisin tehdä.

<http://www.fiftythree.org/etherkiller/>

Kyllä. Tästä se lähtee syntymään. Minulla on vanhan kannettavani virtalähde, jonka voin uhrata. Pistotulppa on maadoittamaton, mikä oikeastaan tekee konstruktiosta luotettavamman: tuho ei ainakaan katkea etuajassa siihen, että maajohtoa pitkin palaava virta aiheuttaa heti sulakkeen palamisen.

Kytken toiseen päähän rikkinäisen RJ45-verkkojohdon. En nyt tiedä, menevätkö karvat aivan oikein maksimituhon takaamiseksi, mutta mittaani joka tapauksessa digitaalisen yleismittarilla, ettei johdon sisäistä oikosulkua ole.

No niin. Se on valmis. Olen aseistettu tulevaisuutta varten. Päätän, että kontrollini ovat seuraavat:

1 = Select Omni-Punch

2 = Select Etherkiller

W = Move forward
S = Move back
A = Move left
D = Move right
CTRL = Crouch
SPACE = Jump
LEFT MOUSE = Attack
RIGHT MOUSE = Slow down time
X = Summon White Avenger

Päivä 63.

Mietin, mitä olenkaan saanut aikaan Jörmungandr:iin liittyen. Ja mitä ehkä olisi vielä tehtävää, ennen kuin sen voisi vihdoin paljastaa. Alkuperäinen listanihan oli seuraava:

- *Renderöinti*
- *Input*
- *Scenemalli, sisältäen myös latauksen ja tallennuksen*
- *Fysiikka*
- *Verkkoliikenne (client, server ja headless server -toimintatilat. Myös clientside prediction ja lag compensation.)*
- *Skriptaus*

Verkkoliikenne osoittautui fysiikan takia osittaiseksi fail:iksi, mutta muuten lista näyttää itseasiassa yllättävän valmiilta.

Vain yksi iso osa-alue puuttuu: skriptaus.

Olkoon niin.

Skriptaus valmiiksi, ja sitten Vision3D-tiimi saa nähdä omin silminensä, mitä on todellinen voima ja vallankäyttö. Mutta huomenna siis hyytyminen työn muodossa jatkuu. Katsotaanpa, mitä siitä puolestaan seuraa.

Niin. Mutta pitäisi päättää skriptikieli.

Aikaa ei ole liikaa käytettäväksi, joten en taida kuitenkaan ottaa flex:iä ja bison:ia kauniiseen käteen, ja alkaa

oman ohjelmointikielen kielioppia sorvaamaan, vaan otan käyttöön jonkin valmiin.

Mutta minkä?

Tähän asti olen valinnut vain voittavia – siis oikeasti – en “voittavia” niinkuin Vision3D:n tapauksessa – teknologioita, kuten Direct3D10, joten nyt ei saa mennä pieleen tässä viimeisessä kriittisessä valinnassa.

Tai voiko nyt Bullet:ia tai ODE:a kumpaakaan sanoa voittavaksi?

Helveti. No joo, mutta fysiikka on enginen core-toiminnallisuuden kannalta toissijainen ominaisuus, koska se on vain yksi tapa ajaa scenen aikapohjaista updatea eteenpäin. Ja en kyllä oikeastaan usko, että Havok tai PhysX:kään suoraan soveltuisivat paremmin client-side predictioniin sillä raa’alla tavalla, jota kokeilin.

Ei nyt angstata enää fysiikasta, vaan mietitään.

Python?

Lua?

Javascript?

Squirrel?

Angelscript?

Brainfuck, Dis tai Malbolge?

Muu, mikä?

Vision3D on tavallaan saastuttanut luan. Tämä vaihtoehto voidaan yliviivata. Lisäksi en missään nimessä haluaisi laskea taulukon indeksi ykkösestä lähtien. Kamalaa.

Python on liian iso ja monoliittinen. Se voi sisältää erinäisiä tapoja aikaansaada UDB:tä, jotka eivät ole tiedossa edes sen kehittäjille itselleen. C++ -toiminnallisuuden bindaus on manuaalisesti erittäin ikävää, tai sitten vaatii bindauskirjaston kuten boost::pythonin tai swig:in, jotka ovat myös puolestaan isoja ja ikäväkäyttöisiä. Javascript-tulkit taas ovat kaikki liian isoja, tai sisältävät runsaasti ulkoisia riippuvuuksia. Niin, ja sitä ei siis muutenkaan voisi käyttää, koska se tulee saastumaan Vision3D 2.0:n myötä.

Angelscript ei ole osoittanut tehoaan missään menestyneessä tai AAA-tason pelissä; sille on lähinnä oma kuppi-

kuntansa gamedev.net:issä. Tosin se on staattisesti tyypitetty, mikä on aina jumalallisempaa.

Squirrelin dokumentaatio on puutteellinen. Sille on useita bindauskirjastoja, joiden valmiusaste on kyseenalainen.

Hemmetti. Mitä tässä jää vaihtoehdoksi?

Tehdä itse?

Siltä alkaa vaikuttaa. Onhan tässä vielä aikaa, ennen kuin Vision3D:n pitäisi olla sisäiseen käyttöön valmis. Kuukausi käytännössä.

Varoittava ääni pään sisässä kyllä jupisee, että tämä on virhe, mutta on pakko punnita kokonaisuutta. Kuinka inhottavaa ja ristiriitaista olisikaan todeta, että renderöijä on ylimaallinen, surface shadereita kääntävä Direct3D10-pohjainen systeemi, mutta skriptikielenä on esim. bloatitunut Python, jonka käyttäjät sitä paitsi usein ovat hörhöjä, ja odottavat, että koko Pythonin systeemikirjasto on käytössä.

Tietysti kääntöpuolena on, että jos / kun teen oman skriptikielen, myös sen suoritusympäristön täytyy olla optimaalinen. Tämä käytännössä edellyttää JIT-kääntämistä, eli skriptin kääntämistä lennossa ja pätkissä prosessorin suoraan ymmärtämäksi konekieleksi.

Tämä myös antaa mitä loistavimpia mahdollisuuksia kryptisiin virhetilanteisiin ja UDB:lle tilaisuuden nostaa rumaa päätään jälleen kerran!

Mutta jos onnistun, tulen todella tuntemaan itseni puolijumalaksi ja demonilordiksi!

Tästä siis lähdetään. Lataan kiltisti flex:in ja bison:in ja alan kirjoittamaan lennossa skriptikieleni Backus-Naur-muotoista kielioppia. Siitä tulee staattisesti tyypitetty proseduraalinen / oliokielihybridi, joka ei tue mitään turhaa hörhöilyä, kuten moninperintää tai templaatteja. Muistinhallinta hoidetaan reference countingin avulla.

Kieleni nimi olkoon Fenriz.

Tajuan kyllä, että kielellä on vahvoja yhtymäkohtia Javaan, mutta Java on myös ajoympäristöltään liian inhotta-

vasti bloatoitunut, että sitä voisi harkita yhdistävänsä Jörmungandr:iin. Tästä ei sensijaan voi missään tapauksessa tulla liian iso! Nyt Jörmungandr-moottorin (myös fysiikan) sisältävä exe-tiedosto on kooltaan noin megatavu. Otan taivoitteeksi, että Fenriz:in lisääminen kasvattaa kokoa korkeintaan kahdellasadallaviidelläkymmenelläkuudella kilotavulla. Se on hyvä ja tasainen luku.

Niin häpeällistä kuin se onkin, ensin unohdan suorituskyvyn ja optimoinnin täysin, ja yritän vain saada kielen pyörimään. Siinäkin tulee riittämään haastetta! Ennen kuin menen nukkumaan, suunnittelen vielä virtuaalikoneen, jolla kielen bytcode-esitystä tullaan ajamaan. Yksinkertaisuuden vuoksi siitä tulee pinopohjainen: kun esim. halutaan laskea kaksi lukua yhteen, ne molemmat pushataan pinoon, jonka jälkeen yhteenlaskukäsky ottaa ne sieltä, suorittaa operaation, ja pushaa tuloksen. Myöhemmin täytyy toteuttaa rekisteripohjaisuus, jolloin esim. rekisteriallokaatioiden hoitaminen tulee tarpeelliseksi.

Fenriz-virtuaalikoneen käskyiksi muodostuvat seuraavat:

PUSHI – tallettaa välittömän-luvun pinoon.

PUSHIDX – lukee pinon päältä indeksin, ja kopioi pinosta indeksöidyn luvun pinon huipulle.

PUSHM – lukee muistista luvun, ja tallettaa sen pinoon

PUSHMIDX – lukee pinon päältä indeksin, lukee osoitteesta [perusosoite + indeksi] luvun, ja tallettaa sen pinoon.

STOREM – lukee pinon päältä luvun, ja tallettaa sen muistiin.

STOREMIDX – lukee pinon päältä kaksi arvoa: talletettavan luvun ja indeksin, ja tallettaa luvun muistiosoitteeseen [perusosoite + indeksi.]

INCI – kasvattaa pinossa olevaa lukua yhdellä. Offset-operandi on välitön.

INCIDX – lukee pinon päältä indeksin, ja kasvattaa pinosta indeksöityä lukua yhdellä.

INCM – kasvattaa muistissa olevaa lukua yhdellä.

INCMIDX – lukee pinon päältä indeksin, ja kasvattaa indeksoitua muistiosoitetta yhdellä.

DECI – vähentää pinossa olevaa lukua yhdellä. Offset-operandi on välitön.

DECIDX – lukee pinon päältä indeksin, ja vähentää pinosta indeksöityä lukua yhdellä.

DECM – vähentää muistissa olevaa lukua yhdellä.

DECMIDX – lukee pinon päältä indeksin, ja vähentää indeksöityä muistiosoitetta yhdellä.

ADDI – lukee pinon päältä kaksi kokonaislukua, laskee ne yhteen, ja tallettaa tuloksen pinoon.

ADDF – lukee pinon päältä kaksi liukulukua, laskee ne yhteen, ja tallettaa tuloksen pinoon.

SUBI – lukee pinon päältä kaksi kokonaislukua, vähentää ne toisistaan, ja tallettaa tuloksen pinoon.

SUBF – lukee pinon päältä kaksi liukulukua, laskee ne yhteen, ja tallettaa tuloksen pinoon.

MULI – lukee pinon päältä kaksi kokonaislukua, kertoo ne keskenään, ja tallettaa tuloksen pinoon.

MULF – lukee pinon päältä kaksi liukulukua, kertoo ne keskenään, ja tallettaa tuloksen pinoon.

DIVI – lukee pinon päältä kaksi kokonaislukua, jakaa ensimmäisen jälkimmäisellä, ja tallettaa tuloksen pinoon. Nollalla jakaminen tuottaa tuloksen nolla. (Fenriz ei tue määrittelemättömiä operaatioita.)

DIVF – lukee pinon päältä kaksi liukulukua, jakaa ensimmäisen jälkimmäisellä, ja tallettaa tuloksen pinoon.

MODI – lukee pinon päältä kaksi kokonaislukua, jakaa ensimmäisen jälkimmäisellä, ja tallettaa jakojäännöksen pinoon. Jos jakaja on nolla, tulos on nolla.

MODF – lukee pinon päältä kaksi liukulukua, jakaa ensimmäisen jälkimmäisellä, ja tallettaa jakojäännöksen pinoon.

NEGI – lukee pinon päältä kokonaisluvun, kääntää sen etumerkin, ja tallettaa tuloksen pinoon.

NEGF – lukee pinon päältä liukuluvun, kääntää sen etumerkin, ja tallettaa tuloksen pinoon.

RCPI – lukee pinon päältä kokonaisluvun, laskee sen käänteisluvun ja tallettaa tuloksen pinoon. Jos luku on nolla, myös tulos on nolla.

RCPF – lukee pinon päältä liukuluvun, laskee sen käänteisluvun ja tallettaa tuloksen pinoon.

AND – lukee pinon päältä kaksi kokonaislukua, tekee niiden välisen and-operaation ja tallettaa tuloksen pinoon.

OR – lukee pinon päältä kaksi kokonaislukua, tekee niiden välisen or-operaation ja tallettaa tuloksen pinoon.

XOR – lukee pinon päältä kaksi kokonaislukua, tekee niiden välisen xor-operaation ja tallettaa tuloksen pinoon.

SHIFTL – lukee pinon päältä kokonaisluvun ja toisen luvun, joka kertoo, kuinka monen bitin verran siirretään vasemmalle. Tulos talletetaan pinoon.

SHIFTLI – lukee pinon päältä kokonaisluvun, siirtää sen bittejä vasemmalle kiinteän operandin verran ja tallettaa tuloksen pinoon.

SHIFTR – lukee pinon päältä kokonaisluvun ja toisen luvun, joka kertoo, kuinka monen bitin verran siirretään oikealle. Tulos talletetaan pinoon.

SHIFTRI – lukee pinon päältä kokonaisluvun, siirtää sen bittejä oikealle kiinteän operandin verran ja tallettaa tuloksen pinoon.

JUMP – ehdoton hyppy absoluuttiseen osoitteeseen.

JUMPZ – hyppy, jos viimeisen operaation tulos oli nolla.

JUMPNZ – hyppy, jos viimeisen operaation tulos ei ollut nolla.

JUMPLT – hyppy, jos viimeisen operaation tulos oli pienempi kuin nolla.

JUMPGT – hyppy, jos viimeisen operaation tulos oli suurempi kuin nolla.

JUMPIDX – ehdoton hyppy indeksöityyn osoitteeseen. Indeksiksi luetaan pinosta.

CALL – aliohjelmakutsu.

METHOD – olion metodikutsu. Olion osoite luetaan pinosta.

LOCAL – varaa pinosta tilaa paikallisille muuttujille.

RET – palaa aliohjelmasta. Siivoaa myös paikalliset muuttujat automaattisesti pois pinosta.

STOP – pysäyttää virtuaalikoneen ajon.

Tuetut sisäiset tietotyypit ovat siis 32-bittinen kokonaisluku, 32-bittinen IEEE-754 -standardin mukainen liukuluku, ja 32-bittinen osoitintyyppi (jos 4 GB:n osoiteavaruus ei riitä skripteillenne, saatte hävetä.) Boolean-tietotyyppi toteutetaan kokonaislukuna, jossa 0 vastaa falsea ja mikä tahansa muu luku truea.

Osoitintyyppi on, kuten aina, pelottava. Sillä voi helposti saada undefined behaviour:ia aikaan virtuaalikoneen sisäissä, jos esim. mielivaltainen kokonaisluku tai liukuluku tulkitaankin osoitteeksi. Muistinhallintaa varten tarvitsen kuitenkin tietoa kaikista skriptissä luoduista objekteista, joten talletan ne ehkä hashmap:iin, ja tarkistan sitten aina ennen osoitteen dereferensointia, että se on valiidi. Tästä kyllä seuraa häpeällistä performanssinmenetystä, mutta kaikella on hintansa. Tosin jos voidaan aukottomasti varmistaa, että itse kieltä tulkittaessa syntyvät bytekoodit eivät koskaan sisällä väärän tietotyypin käsittelyä, voidaan tarkistus poistaa.

Tähän on hyvä päättää. Nukkumaan; toivottavasti tuhoa ja terroria riittää unissani jälleen. Olen havainnut, että tämä valmistaa minut aina parhaiten tulevaa työpäivää varten.

Päivä 64.

Takaisin töihin. Muahaha. Kostoni on pian käsillä. Lisäksi tästä lähtien etherkiller on aina taskussani sen varalta, että pelkkä henkinen kosto ja selätys ei riitä, vaan vaaditaan jotain kouriintuntuvaa.

Muistutani itseäni: jäljelläolevan työn määrä on rajallinen. Fenriz vain valmiiksi, ja sitten rokataan.

Kun astun ovesta ulos, pakkasen puute ihmetyttää. Olikohan jopa plussan puolella? Oikeastaan onkin ihmeellistä, että kylmyys on pysynyt tähän asti, eikä esim. siten, että lunta olisi välillä sulanut, ja sitten jäätynyt uudestaan, jolloin olisi saatu todella saatanallisen mustat, liukkaat ja vaaralliset kelit, kuten joulun aikaan usein on tapahtunut.

Jos muistan oikein, sprint planning on huomenna. Tänä siis vielä hiomme kesken olevia juttuja, ja katsomme, että EduWorld on jokseenkin hyväksyttävästi demottavissa.

Lähes heti, kun saavun toimistolle, Suresh alkaa hiillos-
taa minua softbody-fysiikan suhteen. Sitä pitäisi kuulemma demota erikseen. No okei, tehdään ns. "patjamaailma." Se saa luvan kelvata.

Eikö kelpaa? Mitä muka vielä pitäisi olla? Multithreadaustako? Simulaation eri osien ajamista usealla eri threadilla yhtäaikaa? Joo joo, hakkeroinpa se Bullet:iin päivässä!

Sitten muistan taas kamalan unen, jossa Suresh alkoi

itse ohjelmoida IK-kontrolleria minun yllyttämäni. Tajuan, että jos en halua tämän skenaarion toteutuvan, minun pitää rajoittaa juuri sen suuntaisia heittojani. Sen sijaan kyrräilen ja halvennan muilla tavoilla, jos tarpeen.

Sebastian on tosissaan alkanut tehdä forward renderingiä. Mitään demottavaa siitä ei tietysti vielä ole, vaan huomenna Vision3D:tä ajetaan puhtaasti deferredilla pipelineilla. Itse asiassa olisi kohtalaisen mieluisaa liittyä mukaan tähän työhön ensi sprintissä, mutta katsotaan, miten käy.

Vaikka koodarit ovatkin päällisin puolin tyytyväisiä, voi aistia, että executive-tasolla tarkastellen demosta tulee pettymys. No, Jarmo ja Suresh siitä lähinnä luonnollisesti ottavat vastuun, ehkä sitten mietitään uudemman kerran, onko multicore-ominaisuuksien painottaminen viisasta, kun perustoiminnallisuuskin on vielä puutteellinen.

Mutta nyt ei mietitä tätä shittiä enempää, vaan häivyttään. Fenriz odottaa!

Huomaan, että kestää vielä kohtalaisen pitkä aika, ennen kuin pääsen bytekoodin emittointiin tai ajamiseen. Virtuaalikoneen voisi toteuttaa tietysti ensiksi, mutta se tuntuisi petkuttamiselta – huvi ennen työtä.

Ei. Ensin laitetaan abstraktin syntaksipuun generointi kuntoon, ja sitten ruvetaan miettimään statementtityyppi kerrallaan, miten koodia generoidaan. Itseasiassa perusjutut, kuten if- tai for-lauseet, tulevatkin olemaan suoraviihaisia, mutta muuttujien (erityisesti paikallisten, jotka varataan pinosta tarvittaessa), olioiden ja perinnän toteuttaminen tulee olemaan kimurantimpaa.

Mutta kyllä tästä vielä skriptikieli tulee.

Päivä 65.

Demo- ja sprinttisuunnittelupäivä. Kyllä tästä vielä kamaa tulee.

Suresh selvästi kontrolloi näytöstä. Toisaalta tehokkuus on hyödyllistä, ja on hyvä, ettei yksittäisten ominaisuuksien demoja jäädä märehtimään loputtomiin, mutta esiintymisestä tulee silti mieleen tarpeeton pätemisen tarve.

Kuten arvaan, Tuomas ei ole kovin innoissaan. Uskonpa, että hän olisi odottanut näkevänsä EduWorld:in enemmän toimintakuntoisena. No, on siinä nyt sentään useampi käyttäjä maailmassa, ja se ei kaadu (ainakaan so far.)

Mutta toisaalta, itsepä hän halusi aikataulun kiristämistä ja haasteenottoa! Ja itsepä hän varmaan Kuumakallen pääarkkitehdiksi hyväksyi. Ei tullut vielä atomi- tai molekyyli-tason fysiikkasimulaatiota, mutta sen sijaan tuli päätävästi toimiva virtuaalimaailma, jossa ei voi vielä varsinaisesti tehdä mitään! Ehkä se on jonkinlainen indikaatio siitä, mitä 3D-internet tulee yleisesti olemaan!

Lopulta demot päättyvät, viimeisenä minun softbody-patja- ja pallodemo. Sitten jatketaan sprinttisuunnittelulla. Kuumakalle sanelee, ja me muut kuuntelemme. Kyllähän se näin toimii, ja on ohi nopeammin. Jarmon pitäisi kai käytännössä olla vielä scrum master, mutta aivan sama minulle.

No niin. Nyt me poikkeamme rytmistä. Tarkoituksena

on vetää neljän viikon mämmuttisprintti, jossa toteutetaan loput jutut, ja hiotaan ja testataan ja hiotaan ja testataan.

Ei siinä mitään. Kyllähän niin voi tehdä. Ja varsinkin, jos saan (forward-) renderöintiin liittyvää tekemistä, saatan siitä nauttiakin jollain tasolla. Kunnes Jörmungandr tulee esiin, näyttää hampaansa ja hengittää ilmaan myrkyä, johon Vision3D lannistuu totaalisesti!

Hyi. Saan kyllä myös kaikenlaista kamalaa ja epäolennaista tekemistä, joka oikeastaan minulle edes kuuluisi.

- Renderer: dynamic GLSL generation for forward shading

- Renderer: forward shaded material system

- Renderer: stereo rendering

- Renderer: fallback pipeline

- Renderer: headless mode support

- Scripting: multithreading support

- Scripting: further performance improvements

- Audio: internet streaming

- Audio: reverb modelling

- Audio: occlusion and EQ

- Audio: VOIP support

- Video playback

- Physics: simulation on multiple cores

- Physics: enable huge amount of objects

- Framework: automatic benchmarking system

- Framework: batch file support

Skriptausjutut kuuluisivat ehdottomasti Kaarlolle! Mutta ilmeisesti hän toteuttaa rajapintaan uusia juttuja, EduWorld:in vaatimaa toiminnallisuutta ja mm. käyttäjäprofiilien hallintaa, samalla kun minä yritän optimoida entisiä. Taas tulee The Human Centipede mieleen. Tällä kertaa ei ole yhtään mitään epäilystä, mitä pääätä minä edustan.

Ehdimme juuri ja juuri suunnittelun päätteeksi Sodexon. Aprikoosikiisseli ja vaniljavaahto on yllätyksetön, mutta menettelee.

Sitten koodaamaan renderöintijuttuja. OpenGL! Hyi, olinkin unohtanut kuinka kamalaa se oli. Sebastianin koodia lukiessa kauhun tasapaino on jatkuva: päällisin puolin se on pysynyt hyvälaatuisena, mutta aina jostain syövereistä silmille hyppii arveluttavia, hyvinkin arveluttavia konstruktaje.

Tuo on taattua UDB:tä.

Ja tuo.

Pöyristyttävää. Mainitsen hänelle havaitsemistani ongelmista (itse asiassa varsin hillitysti; virheiden vakavuus edellyttäisi oikeastaan paljon tylympää ulosantia), jolloin vastauksena on epäselvästi muristu "prkl." Kyräilyn tasosta täysi kymppi.

Päivän viimeinen tunti matelee eteenpäin. Mieleni on jo yhdeksänkymmentäprosenttisesti Fenriz:in kimpussa.

Se oli sitten siinä! OpenGL-koodini jää aika rikkinäiselle asteelle, enkä taatusti miettinyt sen multithreadautuvuutta yhtään, mutta huomenna on "hyvä" jatkaa.

Raahhh!

Fenriz calls me from beyond!

Obscene mandates overturned!

From never-ending cycle he shall rise!

To pave the way for the great serpent!

Death / black metal -sanoitukset eivät ole vahva puoleni, mutta ei siitä tässä olekaan kyse, vaan abstraktin syntaksipuun oikeellisesta tulkinnasta.

Kyllä tästä vielä koodigeneraattori tulee!

Päivä 66.

Plusasteita on jälleen jo aamulla. Tämä tulee jo taatusti aikaansaamaan lumen sulamista. Onhan se tavallaan miellyttävän tuntuista, mutta kuten humalan jälkeen seuraa krapula, on melko varmaa, että jäätyminen ja liukkaus seuraavat vielä jatkossa.

Fenriz ei aivan etene toivotulla vauhdilla. Voi olla, että aliarvioin tehtävän laajuuden.

Vaikka menettely on riskialtis, päätän, että saatan työkennellä Fenriz:in parissa myös toimistolla. Tämän vuoksi kopioin Jörmungandr:in taas muistitikulle. Lähdekoodin lisäksi tikulle menee myös exe-tiedosto, josta pystyy ajamaan teepannufysiikkademoa, ja lataamaan scenejä. Aivan piruuttani olen testannut myös EduWorld-scenen lataamista Jörmungandr:iin, vaikka se onkin oikeastaan saastaista.

Executablea ei tietystikään pidä ikinä ajaa toimistolla! Tosin kuin koodi, joka vaatii lähempää tarkastelua paljastuakseen, poikkeava 3D-renderöinti herättäisi kenen tahansa ohikulkevan epäilykset nopeasti.

Toimistolla.

Jatketaanpa sitten. GLSL:n dynaamista generointia.

Ja heti, kun kukaan ei katso, Fenriz:in koodigenerointia. Olisi kyllä mukava, jos joku koodaisi Fenriz:in minun puolestani. Helvetti, mihin hulluuteen sitä ryhtyi. Boost::pyt-

honilla kaikki olisi valmista päivässä...

Saatana! Noin ei saa ajatella.

Kyllä se Fenriz sieltä vielä putkahtaa esiin, kaikessa julmassa komeudessaan.

Mutta takaisin GLSL:n pariin. Minun täytyy myöntää, että vaikka grafiikkarajapinta onkin väärä, tunnen tyydytystä siitä, että deferratulle renderöinnille tulee vaihtoehto. Toivottavasti forward pipelinestä tulee Vision3D:n oletuspipeline.

Tosin se, mikä seuraavaksi on vuorossa, ei todellakaan ole mieltä ylentävää. Fallback pipeline. Se käytännössä tarkoittaa, ettei voida olettaa edes OpenGL 2.0:aa. Ei voida oikeastaan olettaa shadereiden olemassaoloa lainkaan. Eikä voida käyttää pixel buffer object:eja esim. shadow mappinomia varten. Pbufferit puolestaan ovat Intelin näytönohjaimilla varma kaatuminen. Tehdään sitten fixed pipeline-paskaa, ja joku saa koodata stencil shadowit, jos varjoja on pakko olla.

Sebastian tekee forward renderöinnin perustoteutuksen lisäksi render to texture-juttuja, partikkelisysteemin finalisointia ja animaatiosteemin optimointia. Kuten myös Jörmungandr:issa, Vision3D:n meshit tukevat yhtäaikaan sekä luustoanimaatiota että vertex morpheja. Tosin animaatioblendaus ei ole läheskään yhtä jumalallisen täydellinen. Lyhytnäköisesti Sebastian teki ensin CPU skinningin, ja nyt hän saa tapella hardware skinningin kuntoon. Aivan oikein hänelle.

Se hyvä puoli asiassa on, että fallback pipeline voi surutta hyödyntää CPU skinning-polkua. En todellakaan aio mieltä, miten esim. ARB-assemblyshadereilla kirjoitettaisiin skinnaus. Sitäpaitsi, jos Intel GMA950:tä ajatellaan pyhänä benchmark-näytönohjaimena, siinähan prosessori joka tapauksessa ajaa vertex shaderin, eli mitään suorituskykyeroa ei pitäisi tulla. Ja jos tehoa vaaditaan lisää, toteutetaan sitten vaikka SSE-käskyillä vektorisoitua skinnausta.

Iltapäivä keskeytyy ikävästi, kun Suresh tulee vaati-

maan minulta tehtävieni uudelleenpriorisointia. Fysiikan yhtäaikainen laskenta monella corella pitäisi saada mahdollisimman pian valmiiksi, ja siihen pitäisi vielä sisällyttää hyvin optimoitua OpenCL-laskentaa. Ilmeisesti on tarkoitus, että sitä silmällä pitäen aletaan väsäämään toista demosovellusta, ja tätä demoa kuulemma esitellään joillekin vieraillemaan tuleville pampuille jo ennen helmikuuta. Minun puolestani sitä saa esitellä vaikka Mikki Hiirelle! Eri asia on, saavutetaanko threadauksella mitään hyötyä.

Kysyn vielä, että olethan tietoinen, että fysiikkathreadauksen toteuttaminen tällä tasolla vaatii Bulletin syvällistä muokkausta, jolloin siirtyminen kirjaston seuraavaan versioon tulee taas olemaan huomattavan työlästä.

Kysyttyäni kadun välittömästi. Kuumakallelta ei kannattaisi kysyä tai varmistaa yhtään mitään, joutuu vain pahempaan jamaan.

Saan mandaatikseni keskustella Bulletin foorumilla tai irkkikanavalla tekemistäni muutoksista, ja koordinoida niitä yhdessä Erwinin ja kavereiden kanssa sisään kirjastoon patchien muodossa. Tilanteen kamaloittamiseksi entisestään Suresh tekee siitä minulle vielä erillisen taskin.

Vittu vitun vittu!

Aivan järkyttävää. Tiedän kyllä, mitä tästä seuraa: entistäkin suurempaa omistautumista Jörmungandr:ille ja Fenriz:ille. Jos Suresh vain tietäisi!

Kuluu vielä pari tuntia, ja päivän kidutus on viimein ohi. Asunnolle.

Fenriz alkaa suoraan sanoen viimein osoittaa edistymisen merkkejä. Syntaksipuun tulkinta on nyt kunnossa: debugprintit tulostavat "pseudokoodina" hyvinkin lupaavaa ohjelmarakennetta. Seuraava askel on muuttaa pseudokoodi varsinaisiksi bytekoodeiksi. Sitten vielä itse virtuaalikone pystyy, niin pystyy jo ajamaan jotain. Täytyy tosin aloittaa vaatimattomasti: luokkametodit, luokkamuuttujat ja pinosta varattavat paikalliset muuttujat ovat edelleen vaarallisen kimurantteja, ja herättävät mielessäni vaanivaa

pelkoa, mutta ne tullaan selättämään yksi kerrallaan.

Päivä 67.

Näen epämiellyttävää unta softbody-fysiikan toteuttamisesta useilla coreilla. Softbodyilla toteutetaan myös käyttäjien hahmoja, ja ne venyvät ja epämuodostuvat, sitten räjähtävät kaikkiin suuntiin, ja sisältä tulee vaaleanpunaista sumupartikkeliefektiä. Hyi!

Herään. Onneksi minkään projektin vaatimuksena ikinä ei voi olla tuollainen.

Tosin havaitsen nukkuneeni pommiin. En kuullut heräystä, ja nyt kello on jo yli kymmenen. Jotain vaikutusta voi olla sillä, että koodasin Fenriz:in bytekoodin tuottamista neljään, mikä ei ollut alkuperäinen tarkoitus. Mutta kun pääsi siihen asti, ei millään malttanut lopettaa.

Tästä päivästä ei voi tulla kovin hyvä päivä. Eihän tämä siis maailmanloppu ole, koska voin olla toimistolla vastavasti myöhempään, mutta on inhottava jäädä hommiin muiden häipyessä. Parempi olisi tietysti lähteä muiden jäädessä. Tosin myöhällä voisi alkaa koodaamaan surutta Jörmungandri:a, kun kukaan ei katso!

Lämpimät kelit jatkuvat, mikä nyt tekee elämästä vähän siedettävämpää. Hieman yli yhdentoista olen viimein toimistolla. Kone auki, Visual Studio päälle ja iljettävän multi-core-fysiikan kimppuun.

Ilmapiiri vaikuttaa jotenkin oudolta ja pingoittuneelta. Jotain supatusta ja kuiskailua on havaittavissa.

Mitä ihmettä?

Liittyykö se minuun jollain tavalla? Vai olenko tulossa paranoidiksi?

Korjaus: minähän ehdottomasti OLEN paranoi. Se on välttämätön luonteenpiirre mm. toimenkuvassa Graphics Subsystem Technical Lead, tai jopa Lead Architect, joka olin vähän aikaa.

Jos koodi ei kulje, niin rekisteröidytäänpä sitten Bulletin foorumeille. Katsotaan minkälaista porukkaa siellä on hörhöilemässä.

Notice: the Administrator has permanently banned your IP address.

Jaha, eipä sitten rekisteröidyt. Ehkä tästä IP-osoiteblokista on joskus tapahtunut jotain megalomaanista häiriköintiä, ja siksi tapahtuu näin, mutta minä olen kyllä pelkästään mielissäni. Tosin nyt varmaan Suresh tulee vaatimaan, että käytän irkkiä tai postituslistaa.

Jos foorumi ei hyväksy minua, niin koodataanpa sitten. Voin kyllä aavistaa, miten Bullet tulee räjähtämään käsiin. Ei sitä ole tarkoitettu tekemään tällaista. Tai onhan siinä valmiina joidenkin yksittäisten osa-alueiden threadausta, kuten constraint solverissa, mutta nyt on siis tarkoitus threadata koko paska.

Mutta nyt välissä syömään.

Sodexossa on jälkiruokana suklaavaahtoa, jonka lusi-koin mekaanisesti. Siitä ei ole mitään mainittavaa. Se katoaa tästä ikuisuuteen mitään jälkiä jättämättä.

Kun palaan, jotain on selvästi tapahtumassa.

Aavistukseni eivät ehkä siis olleet pelkästään paranoidisia. Jarmo – siis Juippi – kutsuu minut Tuomaksen pyhään huoneeseen.

Jotain erikoislaatuista siis todellakin on tapahtumassa. Ensiaavistukseni on, että se on jotain pahaa. Ei se voi koskaan olla mitään hyvää, jos minut summonoidaan CEO:n huoneeseen! Varsinkin jos se tapahtuu kesken sprintin,

jossa minulle on määrätty vain turhaa ja vieläkin turhem-
paa tekemistä! (Poislukien forward-renderöinti.)

Astun sisään inner sanctumiin.

Tuomas ja Kuumakalle ovat paikalla.

Kuumakallella on kädessään muistitikku.

Tunnistan sen omaksi muistitikukseni, jonka menetin onnettomuudessa. Miten se on mahdollista? Onko joku tutkinut törmäyspaikkaa, ja toimittanut tikun Suresh:ille? Vai onko hän käynyt siellä itse? Mutta eihän hän vielä ollut projektissa silloin! Oliko koko kolari järjestetty juttu? Liian paljon kysymyksiä ja liian vähän vastauksia. Perkele, perkele, perkele!

“Haluatko selittää omin sanoin, mitä tällä tikulla on?”
Suresh kysyy.

Sitten tajuan, mistä on kyse.

Vastaus on lapsellisen yksinkertainen. Tikku putosi toimiston oven lähistölle taskustani lumihankeen, ja nyt, kun lunta on lämpimien kelien johdossa sulanut, se tuli esiin.

Nyt kyllä vituttaa.

Jörmungandr tulee esiin, ennen kuin olisi aika. Eikä tästä nyt voi seurata mitään hyvää. Tästä voi seurata ainoastaan jotain erittäin huonoa. Kerään raivoa hetken, ja alan sitten puhua. Ääneni on matalan karhea, ja uhkaa tihkuva.

“Se on seuraavan sukupolven engine. Se on engine, joka Vision3D olisi ehkä saattanut olla, jos sitä ei olisi pilattu kaikenlaisella hörhöilyllä.”

“Olet siis tehnyt tätä koodia työajalla?” Tuomas kysyy.

“Hyvin vähän siitä. Mutta eipä sillä ole väliä, koska kaikki kirjoittamani koodi, olipa se tehty työajan sisällä tai ulkopuolella, työpaikan sisä- tai ulkopuolella, on teidän omaisuuttanne.”

“Mutta miksi?” Jarmo kysyy puolestaan. “Mikset puhunut, jos sinusta tuntui, että projekti oli menossa suuntaan, jota et pitänyt oikeana?”

“Mitä väliä sillä olisi ollut? Jos suunnilleen ensimmäisenä päivänä kysyit, että oletteko lukeneet artikkelia Why you should use OpenGL and not DirectX, niin huomaaan

kyllä aika suoraan, että olet niin ideologisesti väritynyt, ettet kuuntele järjen ääntä.”

Jarmo seisoo naama punaisena, eikä sano mitään.

“Olen havainnut, ettet sinä ole ollenkaan yhteistyökykyinen. Sinä teet vain omia juttujasi, omalla tavallasi, etkä ota mitään palautetta vastaan,” Suresh alkaa tilittää.

Tuomas seuraa tilannetta kuin hiljainen kuoleman enkeli.

Nyt on minun aika räjähtää. Siis räjähtää oikein kunnolla. Tästä se alkaa. Tästä alkaa minun kostoni. Se alkaa ennen aikaansa, mutta ei voi mitään.

Otan etherkillerin esiin taskustani, sekä muistitikun, joka sisältää Jörmungandr:in tuoreimman version.

“Voi vitun kuumakalle! Siis voi paskojen paska! Ja sinä Juippi siinä, pyyhi nyt jo tuo helvetin tekopyhä ilme naamaltasi, sillä tässä käsitellään pyhiä asioita, asioita joita sinä et ehkä ikinä tule ymmärtämään! En tiedä mistä minun pitäisi aloittaa, joten perkele ... aargh ... aloitan ehkä aivan alusta. Tai ei. Ensin minun pitää puuttua sinuun, spesifisesti sinuun, Suresh. Tai Kuumakalle, kuten olen sinut nimennyt. Sinä siis, tietämättä asioista yhtään vittu mitään, haluat, että me teemme multithreadingia engineen, joka ei sitä tarvitse pienimmässäkään määrin. Keitä ihmeen hörhöjä yrität kosiskella tällä – en tiedä. Tuomastako mahdollisesti? Toivon, että Tuomas olisi sen yläpuolella, sillä hän on sentään taistellut samaa sotaa kuin mekin, kun Vision3D oli vain versio 1.0, tai jopa vieläkin varhaisempi, ehkä versio 0.5. Perkele! Silloin ei todellakaan ihkuiltu minkään multithreadingin perään, koska prosessoreissa oli vain yksi core, jonka nopeus oli korkeintaan yksi gigahertsi! Mutta perkele, silti se toimi, ja toimi erittäin hyvin, Direct3D 8:n ja sittemmin 9:n päällä. Fixed functionilla, mutta silti kaikki asiakkaat olivat tyytyväisiä. Ainakin suunnilleen. Jos undefined behaviouria tuli, se johtui aina jostain tarkasti määritellystä syystä, jonka pystyi nopeasti paikallistamaan! Ei perkele! (Yritä nyt tässä olla sitten rehellisesti oma itsesi kun korporaatiokoreografia vetää jalat

alta. Ei tällainen työyhteisöteatteri voi koskaan tyydyttää väsynyttä sieluani. Tulosteeskentely on näiden saatanallisten käytyreiden keksimä salajuoni, jolla ne yrittää suistaa minut raiteiltaan. En minä pysty tähän. Ei minusta ole tähän. Mää oon tuhonnut kaiken. Ei nää tajua mitään. Älä sano mitään. Anna mun olla.) Saatana! Mä tiedän mitä olis pitänyt tehdä. Ymmärrättekö! Mä kerroin kyllä, että perusta tälle systeemille on ihan perseestä. Kuuletteko te vieläkään. Helvetti. Antaisitte vain minun tehdä tämän niin kuin parhaaksi näen. Vittu. Saatana. Mulla on kyllä teille vastaus. Haistakaa pitkä -Wall paska! Ja ennen kuin edes pienissä mielissäanne sitä harkitsette, niin voitte tunkea sen OpenCL-optimointischeissan suoraan syvälle ohutsuoleen asti. Jumalauta! Vitun korporaatiokätyrit! Jos te haluatte, että mä istun täällä tekemässä jotain PASKAA SUORAAN SAATANASTA, niin turha luulo! Jos pystyisitte edes yhden kerran kuuntelemaan järkipuhetta ja näkemään noiden jumalattomien pullonpohjakorporaatiolinssien läpi mitä maailmassa oikeasti tapahtuu ja mitä meidän OIKEASTI pitäisi tehdä, niin teillä voisi olla pieni mahdollisuus selvitä tästä voittajina. Mutta ette te ymmärrä, ette te halua ymmärtää. Loppujen lopuksi mietin, onko teillä aivoja ollenkaan, vai ovatko ne yksinkertaisesti syväjäässä. Ymmärrän tietysti sen, että ette voi haluta mitään liian vaikeaa, liian kauan kestäväää tai hankalasti ymmärrettävää skenaariota toteutuvaksi, mutta hei! Herätkää. Mä istun, kirjoitan, raadan ja tämä on palkka. Ette voi olla tosissanne. Sitten, kun mulla ois ratkaisu, niin te vaan tuijotatte ja hiljenette. Sosiaaliset taidot kuin hitto vie mun mykällä velipuolella. Menkää kuule tonne saunan eteiseen ja kattokaa peiliin. Rikkokaa se peili ja repikää sydänverellä kirjoitetut koodirivit rikki, viiltäkää mun sielu ja katkokaa mun sormet. Ei tuu koodin koodia sen jälkeen rikkomaan teidän täydellistä maailmaanne.”

Jarmo seisoo ilmeettömänä.

Tekisi mieli ottaa tuo sun projektikansio, ja lätkäistä se sun naamaan. Tai vaikka silppuroida se sun sormien kans-

sa. Ei hyvä päivä. Kaikki se passiivisaggressiivinen raivo, jota olen hautoanut päivästä päivään, yöstä yöhön, se pääsee nyt valloilleen. EI HYVÄ PÄIVÄ. Tämä on se päivä, se tuomion päivä. Päivä, jolloin paha saa palkkansa.

Mutta nyt seuraa vaikea valinta. Tuomas, Jarmo ja Suresh ovat huoneen takaseinällä, joten minulla on periaatteessa esteetön pääsy Tuomaksen koneelle, ainakin parin sekunnin ajaksi.

Muistitikku, vai etherkiller?

Luomukseni esittely, vai pelkkä totaalinen tuhoaminen?

Raivoni laantuu ehkä aivan aavistuksen verran, ja tästä syystä päädyn ensimmäiseen valintaan.

Luomukseni. Jörmungandr.

Salamannopealla liikkeellä insertoin tikun vapaaseen USB-paikkaan koneessa. Näkyvästä etherkilleristä huolimatta minua ei estetä fyysisesti, kun havaitaan, etten aio välittömästi tuhota mitään.

Tikun sisältö aukeaa koneen ruudulle.

Kaksoisklikkaan Jörmungandr-ikonია.

Hetken ajan minusta tuntuu taas samalta, kuin Paul tai Jessica Atreidekselta. Aika pysähtyy. Ajan Jörmungandr:ia näyttöohjaimella, jolla sitä ei ole ennen ajettu, joten UDB:n mahdollisuus on enemmän kuin todellinen.

Kaikki voisi kaatua tähän. Ilmestyisi vain tylyn toteava laatikko "Jörmungandr.exe has stopped working" – Tuomaksen koneessa on Windows Vista – ja kaikki olisi lopussa. Ei tulisi demonstraatiota. Ei tulisi vaikerrukseni ja julistukseni toista osuutta, jossa konkreettisesti kerron, miten asiat olisi tullut tehdä!

Mutta näin ei käy.

Koodini on jumalallista.

Direct3D10-engineni toimii oikein myös tällä näyttöohjaimella. Halleluja!

Ruutu mustenee, ja 3D-scene ilmestyy ruudulle. Siinä on pitkäksi ja leveäksi venytetty laatikko, joka toimii lattiana, useita muita, erikokoisia laatikoita, sekä teepannu, jota käyttäjä ohjaa. Kaikki objektit on valaistu sateenkaaren vä-

reissä; scenessä on 20 spottivaloa, joille kaikille lasketaan hardware shadow map! Joka frame!

“Tämä siis on Jörmungandr Engine. Olen kirjoittanut sitä suunnilleen siitä lähtien, kun Vision3D 2.0:aa alettiin tekemään virallisesti. Ja katsokaa, mitä olen saanut aikaan! Aivan yksin! Vapaana korporaation kahleista tein vain oikeita päätöksiä. Ainoastaan oikeita päätöksiä! Tajuatteko sen? Voitteko te edes tajuta? Jumalallinen Direct3D10-renderöinti ja shader model 4.0! Jumalallinen shaderien generointi HLSL-fragmenteista! Eikä vain mistä tahansa fragmenteista, vaan surface shader-kuvauksista, joilla jopa Tuomas voisi vaivatta kuvailla yhtälöt, joilla pintaa on tarkoitus valaista! Ei typerää crossplatform-rajoitusta ja OpenGL:ää, vaan tuetaan sitä mikä toimii parhaiten, ja mikä kuitenkin useimmilla asiakkaillamme on: Windows! Bloatoituneen ja hitaan Qt:n sijasta oma, jumalallinen UI-systeemi, joka toteuttaa vain sen, mikä on tarpeellinen. Bloatoituneen ja hitaan Qt:n eventtisysteemin ja signal-slot -konnektioiden sijasta oma eventtijärjestelmä, joka toteuttaa vain sen, mikä on välttämätöntä ja OIKEIN!”

Pysähdyn vetämään hieman henkeä, pyörin 3D-maailmassa ympyrää, ja asemoin näkymän kuvakulmaan, jossa varjojen aliasointiartifaktit ovat minimissään.

“Fysiikan Jörmungandr toteuttaa – arvaatteko millä? Ei suinkaan Bulletilla, vaan Open Dynamics Enginellä! Vaikka Bullet onkin normatiivisesti ajatellen parempi – se on pinnallisen näkemyksen mukaan suorituskykyisempi ja featurerikkaampi – siinä on ongelmia, jotka on hankala kiertää, kuten fysiikan huomattavasti huonompi deterministisyys. Mutta ODE:lla on mahdollista toteuttaa verkotettu fysiikka, kunhan vain CPU-kuorman ylittyminen hallitaan. Huomatkaa tässä siis se henkinen vahvuus, jota osoitin. Häpeäkseni joudun kyllä myöntämään, että integroin ensin lammasmaisesti Bullet:in, niinkuin Vision3D:ssäkin tehtiin, mutta sitten heräsin huomaamaan sen epäkohdat. Teidänkin olisi pitänyt huomata ne! Saatana! Mutta joka tapauksessa, täysin omalla ajallani ja itseäni säästelemättä

siirryinkin ODE:n käyttöön, koska tiesin, että sen pystyisi tekemään oikein. MINÄ TIESIN! Paremmin, kuin mitä se yksi jo-kuollut fyysikkourpo teki. Nyt se toimii, paremmin kuin koskaan! Skriptirajapintakin minun oli tarkoitus toteuttaa – täysin oma sellainen – mutta ilmeisesti Kuuma-kallen petollisuuden ja omahyväisen pätemisen tarpeen johdosta Jörmungandr paljastui, ennen kuin se oli valmis, ja se on siis toistaiseksi vielä kesken.”

Vähitellen minusta tuntuu, että vetoni loppuu kesken. Äänikin alkaa käheytyä. Nyt on aivan sama, mitä tapahtuu. Vaikkei se tapahtunutkaan juuri niinkuin piti, kostoni on silti toteutunut. Todelliset ajatuksieni syöverit ja Jörmungandr'in tämänhetkinen tila on paljastettu, ja se, mitä nyt seuraa, sillä ei kyllä suoraan sanoen ole enää vitunkaan väliä.

Muistitikku on käytetty.

Onko nyt etherkillerin vuoro?

Mutta ei.

Energiaa ei ole enää. Kuji-kirini voima on käytetty.

Laskelmieni mukaan olen oppinut hallitsemaan vasta seitsemän viiltoa. Kaksi on vielä ulottumattomissani. Ennen kaikkea valaistuminen puuttuu: sitä en vain ole vielä ehtinyt saavuttaa.

Odotan pari sekuntia. Enemmänkin. Ehkä kymmenen.

Kukaan ei puhu vielääkään.

Rykäisen, ja yllättävä selkeys valtaa mieleni. Nyt tiedän, miten tämä käsikirjoitus jatkuu.

“Jos kaikille ei ole vielä selvää, on siis varmaankin aika ymmärrettävää, etten aio jatkaa täällä. Ja ette varmaan sitä sallisikaan. Uskon, että työsopimukseni on itseasiassa laiton – ei voida laillisesti määrätä sellaista, että jopa vapaa-ajalla – ja Jörmungandr:han on vähintään kahdeksankymmentä- ellei yhdeksänkymmentäprosenttisestikin vapaa-ajallani kirjoittama – kirjoittamani koodi olisi firman omaisuutta, mutta en aio ruveta siitä raastuvassa tappelemaan. Olen sen yläpuolella. Minun puolestani te voitte tehdä Jörmungandr:illa AIVAN VITTU MITÄ TEITÄ HUVITTA, ”

koska se oli vain kostoni välikappale. Ei olisi Jörmungandr:ia ilman Vision3D:tä. Sankari tarvitsee roistoa, ja hyvyys pahuutta. Minä jätän nyt nämä molemmat projektit. Kiitos ja hyvästi. Aamen.”

Edelleenkään kukaan ei puhu mitään.

“Selvä,” Tuomas toteaa lopuksi asiallisen hartaasti.

Ennen kuin poistun, luovutan kulkukorttini Tuomaksen pöydälle. Tämä oli siis nyt ehdottomasti tässä.

Kävelen avokonttorin lävitse. Sebastian – Kaarlo – Hannele – Veli-Matti (lopuilla ei juuri ole väliä) – taistelkaa! Jos teistä tuntuu, että Vision3D on taistelemisen arvoinen, sovitkaa niin perkeleesti, että tekisitte siitä paremman. Jos ei muuta, niin paremman ainakin itsellenne. Ettei se olisi vain kasa huuruavaa paskaa, kieroutuneita ylhäältä saneltuja tavoitteita, ja satunnaisesti esiintyvää UDB:tä.

Mutta minä en enää palaa tänne.

Minä olen vapaa.

Kun poistun, havaitsen paikalle juuri saapuneet vieraat. Tai ainakaan en ole heitä aikaisemmin nähnyt, joten heidän täytyy olla vieraita. Nainen päällään thrash-liivi ja farkkuhame, sekä pitkähäkö vaalea- ja pitkätukkainen miehenkolppi. Havaitsen, että naisen kassista pilkottaa turkoosinvärinen pehmolelu. Onkohan tämä se porukka, jolle softbody-fysiikkaa ja insinööritason fysiikkasimulaatiota oli tarkoitus esitellä? En tiedä, enkä välitä.

Olen ulkona.

Tuntuu omituiselta nousta Valkoisen Kostajan selkään, kun on vielä valoisaa. En pidä mitään kiirettä, sillä nyt Jörmungandr:in tai Fenriz:in jatkaminen ei odota minua. Niillä ei ole enää merkitystä minulle.

Normaaliolosuhteissa olisin avannut työasemani heti perille saavuttuani. Nyt se pysyy kiinni, kun kaadan itselleni ensin pitkän viskipaukun jäillä. Vasta siemaittuani sitä hie-man käynnistän koneen.

Ajaudun lukemaan gamedev.net:iä.

Voisin arvioida, että yhdeksänkymmentä prosenttia

postauksista on todella typeriä, ellei suorastaan rikollisen ala-arvoisia. Usein niissä kysytään ratkaisua johonkin aivan ilmeiseen ongelmaan (kuten esim. varjojen renderöintiin OpenGL immediate modessa, hyl!) tai sitten toisessa ääripäässä mennään heti syvälle kysyjän tasolle mahdotomaan ongelmaan, kuten yleispätevän pelimoottorin toteuttamiseen ilman pienintäkään käsitystä siitä, minkälainen peli moottorilla olisi tarkoitus tehdä.

Vaikka saapuessani mieleni olikin rauhallinen, postien typeryyks saa raivoni kiehumaan. Viski ei suinkaan tyyntytä sitä, vaan lietsoo mieltäni kiehumaan yhä pahemmin. Nyt olisi hyvä tilaisuus rekisteröidä uusi nick ja häiriköidä todella inhottavalla tavalla noille aloittelijoille.

Perun suunnitelmani viime hetkellä. En voittaisi sillä mitään, ja alentuisin tasolle, jota kuji-kirin opit eivät millään muotoa sallisi.

Mutta nyt täytyy myöntää, että tyhjyys tuntuu pelottavalta. Jos en työstä enää Jörmungandr:ia, mitä ihmettä sitten teen? Millä silloin on enää väliä? Kuji-kirillä? Ehkä voisin keskittyä nyt harjoitukseen entistä voimakkaammin, kun muita pakotteita ei enää ole. Keskittyisin omaksumaan tyhjyyden perustavanlaatuisiksi osaksi itseäni, ja saavuttamaan viimein valaistumisen.

Viskipaukku lähestyy vähitellen loppuaan. Sen maku muuttuu hieman karvaasta ja epämiellyttävästä huomattavasti siedettävämmäksi, samaan tahtiin humalatilani kasvun kanssa. Kohta sitä voisi juoda kuin vettä!

Näin on tehtävä. Kaadan toisen samanlaisen, ja siirryn gamedev:istä devmaster.net:iin. Tämä Torque-moottorin valmistajien sponsoroima sivusto on, hyvinkin mahdollisesti vielä paljon gamedev.net:iä ali-inhimilisempi. Jos engine-arvosteluja lukee, tulee kyllä suoraan sanoen paha olo. Joko moottori on parasta ja jumalallisinta ikinä, tai sitten sitä teilitaan armotta, tosin yleensä täysin epäolennaisuuksiin takertuen, varsinaiset oikeutetut kritiikin kohteet sivuuttaen.

Hyi kamala. Foorumit osoittavat sentään engine-arvos-

teluja korkeampaa aivotoimintaa, muttei paljon. Miksi edes eksyin tänne? Äkkiä pois.

Ehkä halusin oikeutusta Jörmungandr:issa tehdyille valinnoille, ehkä halusin nähdä jonkun toisenkin tappelevan samanlaisten dilemموjen kanssa. Mutta turha toivo, että joku kirjoittaisi jumalallista engineä, ja tuhlaisi aikaansa keskustelemalla siitä tällä keskustelualueella. Eikä sitä-paitsi Jörmungandr tarvitse mitään oikeutuksia, koska siinä tehdyt valinnat olivat oikeita, PISTE.

Vai halusinko nähdä jonkun tappelevan samanlaisen idiotismin kanssa, kuin Vision3D? Netistä on kyllä helppo löytää kauhutarinoita pieleen menneistä ohjelmistoprojekteista, tai vaikka pelien postmortemeita, joissa What Went Wrong-osastolta löytyy suunnilleen kaikki kuviteltava. Mutta uskon, että Vision3D on silti jopa maailman mitataavassa ainutlaatuinen tapaus. Jälkikäteen tuntuu hyvinkin ihmeelliseltä, että middleware-kehitykseen pystyi suhtautumaan niin suurella inkompetenssilla, kuin mitä me osoitimme, ja silti saamaan asiakkaita. Helvetti, nyt kun ajattelen, ei kenenkään olisi pitänyt ikinä käyttää sitä! Paljon paremman tuloksen olisi käytännössä helposti saanut ilmaiseksi esim. OGRE:n ja ENet:in ja jonkin haluamansa skriptikielen yhdistämällä. Perkele! Mitä me oikeastaan myimme usean vuoden ajan?

No, ehkä nyt olen hieman tyly. Kyllähän Vision3D todistetusti toimi ja sillä sai juttuja aikaan, sille päälle sattuaan. Ja olihan meillä nyt sentään ihan virallinen tuki, sen sijaan, että käyttäjien olisi pitänyt kirjoitella random-föörumille tai postituslistalle, ja ihmetellä, vastaako kukaan ikinä, vai saisiko sieltä vastaansa vain enemmän tai vähemmän ilkeää herjaa.

Toinen viskiryyppy sulautuu kolmanneksi ja lopulta neljänneksi siten, etten varsinaisesti edes huomaa. Otan etherkillerin esille. Pitäisiköhän sitä kokeilla? Hmm. Se voisi olla vaarallista. Vaikkei saisikaan sähköiskua, voisi syttyä tulipalo. Etherkilleriä pitäisi oikeastaan testata vain turvallisissa laboratorio-olosuhteissa, joissa on käytettä-

vissä päävirtakytkin, ja syttymätön testipenkki.

Mutta jotenkin minusta tuntuu, että tänä iltana en tule elämään turvallisesti.

Luen vielä gamasutrasta masentavan postmortemin, ja sitten totean, että nettiä on selattu riittävästi. Pysyvästi. Kyllä. On aika tehdä testi.

Valitsen satunnaisen portin kytkimestäni, joka yhdistää DSL-modeemiin, palvelimen ja työasemani, ja kytken etherkillerin RJ45-liittimen siihen.

Sitten tarjoan toiselle päälle 230 V vaihtojännitettä.

Kuuluu poksahdus, ja tunnen savun hajua. Samalla koneistani sammuu virta.

Sulake on ilmiselvästi palanut. Perkele! Mutta enkö mitannut, ettei johdoissa pitäisi olla oikosulkua?

No, ehkä etherkiller poltti jotain kytkimen sisältä siten, että siellä syntyi suora yhteys vaiheen ja nollan välillä. Katsotaanpa.

Poistan etherkillerin (se tuntuu jonkin verran lämpimältä) sekä pistorasiasta että kytkimestä, ja menen sula-ketaululle. Alan olla sen verran humalassa, että lähes kaadun.

Katsotaanpa. Kyllä, palanut sulakehan se siellä.

Onneksi varasulakkeita on.

Tasan yksi.

Ruuvaan sen paikoilleen, ja laitteistooni tulee taas virtaa. Pettymykseni on suuri, kun koneiden käynnistyttyä ne paitsi toimivat täysin normaalisti, myös nettiyhteys toimii, niinkuin mitään ei olisi tapahtunut. Ainoastaan kytkimen portti, jossa etherkiller oli kiinni, on mustunut, ja jos kytkin työaseman siihen, silloin linkkivalo ei syty eivätkä bitit liiku.

Tuhovaikutus ei ollut ollenkaan niin totaalinen, kuin olisin toivonut. Uskon, että tämä johtuu vain omasta asiantuntemukseni puutteesta. Todellinen BOFH osaisi rakentaa etherkillerin, joka tuhoaisi kaiken laitteiston vähintään sadan metrin säteellä, kuin pysäyttämätön omni-punch!

Mutta silti on pakko kokeilla uudestaan. Tällä kertaa

päätän kytkeä etherkillerin suoraan työasemani lähiverkkoliitintään, koneen ollessa päällä.

Ensin RJ45-liitin paikoilleen, sitten verkkojännitettä.

Poks!

Koneen ruudulla näkyy sekunnin murto-osan ajan sininen ruutu, ennen kuin se pimenee totaalisesti. Tällä kertaa sulake ei jostain kumman syystä pala.

Kumarrun ottamaan koneesta tukea, aikomukseni syöttää jännitettä seuraavaksi serveriini, ja katsoa mitä sille tapahtuu.

Humalani johdosta en ajattele aivan loppuun asti.

Mutta aivojeni on liian myöhäistä pyörtää lihaksille mennyttä käskyä. Koneeni on nimittäin maadoittamattomassa pistorasiassa, ja tämä selittää sen, miksei sulake tällä kertaa palanut. Mutta (metalli)rungon täytyy ehdottomasti olla vielä jännitteinen.

Kätteni ja kotelon välille syntyy galvaaninen yhteys.

Sähkövirta kulkee ruumiini lävitse.

Sitten mustenee.

ZAI – Luonnonvoimien hallinta

Aika: 08:55

Paikka: Throne-ryhmän laitoksen yläpuolella

Fuusiokennon lataus: 52 %

Throne Group:in laitoksen valot tulevat näkyviin pienestä sivuikkunasta: he aloittavat laskeutumisen. Seuraavaksi alkaa infernaalinen tärinä, joka ei voi olla pelkästään turbulenssia. Ei todellakaan.

Heitä tulitetaan!

Kim miettii, että Amos:in tehtävänä on nyt tasapainottaa yksinkertainen yhtälö (jos se edes on oikea termi): sen toisessa päässä on laskeutuminen niin hitaasti, että heidät ehditään ampuu seulaksi, ja toisessa päässä koneen hajottaminen liian hätäisellä alastulolla.

On turhauttavaa, ettei kone itsessään ole aseistettu. Toisin sanoen heidän pitää päästä ensin maahan, ennen kuin he voivat puolestaan jakaa lyijyä vastapuolelle. Onkohan siellä jo koneita vastassa, vai pelkästään lisää Norman:in henkilökohtaista armeijaa?

Kohta nähdään.

Äkkiä koko ruuma kallistuu villisti; ainakin neljäkymmentäviisi astetta. Koneen moottorien vinkuna yltyy.

Tämä ei vaikuta hyvältä.

Vielä kymmenen sekuntia heittelehtimistä, ja sitten rysähtää. Turvavyöt olisivat olleet paikallaan, mutta sellaista luksusta rahtitilasta ei vain löydy.

Ilmeisesti ollaan maassa. Nyt konetuliaseiden nakutus ja osumien metalliset kolahdukset kuuluvat selvästi sammuttujen suihkumoottorien yli.

Kim katsoo pikaisesti ympärilleen: vahingoittuiko kukaan vakavasti? Ei sillä, että hän välittäisi, mutta taktiikan vuoksi on hyvä tietää, kuinka monta heitä on vielä pystyssä. Jeff, Emily, Rashid, Micha, Orzo. Verta heistä joistakin valuu, mutta kukaan ei näytä olevan kuolemaisillaan.

Kim yrittää katsoa ikkunasta, näkyisikö vastustajia, mutta se on liian pieni, että taistelukentästä saisi kokonaiskuvaa. Täytyy siis suunnilleen rynnätä takaa ulos ja toivoa parasta.

Amos:in ääni kuuluu sisäradion yli:

“Siellä on ainakin parikymmentä Normanin tyyppiä edessäpäin. Tulkaa ulos äkkiä, niin ne eivät ehdi kiertää koneen luokse.”

Rahtitilan iso takaluukku, joka toimii myös lastausrampina, aukeaa. Kukaan ei vastusta, että Kim ottaa konekiväärin kumpaankin käteen, ja menee ulos ensimmäisenä.

Se, miten Amos pääsee ulos ohjaamosta, voi olla mielenkiintoisempaa.

Ulkona odottaa suunnilleen samanlainen kiitorata ja lentokenttä, kuin miltä he lähtivät. Tosin nyt ei osuttu kiitoradalle, vaan nurmikolle.

Huomattava eroavaisuus on tasasivuisen puolisuunnikkaan muotoinen, kirkkaasti valaistu iso rakennus. Tämä on mitä ilmeisimmin Throne Group:in tiedelaitos, tai ainakin sen maanpäällinen osa.

Rakennuksen suunnasta ammutaan. Sinne siis.

Kim miettii hetken, miksi on ryntäämässä ehkä varmaan kuolemaan. Fuusiokennon korvaaminen – onko sillä merkitystä? Koneiden ja maailmanlopun pysäyttäminen – ei senkään vertaa.

Mutta hän haluaa tappaa.

Jos vastustajat ovat ihmisiä, heidän täytyy kohdata lopunsa verisesti, julmasti ja tylästi.

Micha juoksee hänen vierellään, myöskin konekiväärillä aseistettuna. Se on jonkin verran kolhiutuneen näköinen, se oli ehkä lentokentän hangaarin katolta pudonneelta kommandolta.

Sitten he avaavat tulen. Muita ei vielä näy, mutta ei heidän kannattaisikaan huonommin aseistettuna uhrata itseään turhaan. Siis ei Kim:iä mahdollinen ihmiselämien päättyminen hetkauta, edelleenkään, mutta taktisesti se olisi edelleenkin typerää.

Norman:in vartiokaartia lakoaa, mutta monta on vielä pystyssä. Kim tuntee osumat raajoissaan kuin hyttysen pistoina. Ne ovat merkityksettömiä.

Hänen vierellään Micha huutaa konekiväärinkin papatuksen yli.

“URAAAAAAAAA!!!”

Sitten huuto katkeaa äkisti kurlauksen kaltaiseen ääneen. Kim katsoo sivulleen, eikä Micha ole enää pystyssä. Kim antaa tulisuihkujensa pyyhkiä vasemmalta oikealle, sitten takaisin vasemmalle.

Nyt vastapuolen suuliekkejä ei enää näy. Mustaan pansariin ja kypäriin pukeutuneet kommandot makaavat kaikki maassa, kuolleina tai pian kuolevina.

Micha korahtaa vielä takana, ja Kim kääntyy katsomaan. Tämän kurkkuun on ilmeisesti osunut. Micha pitelee kaulaansa, mutta sormien läpi ruiskuaa verta jokaisella hiipuvalla sydämenlyönnillä. Ei hyvä. Tämän soturin taistelu päättyy tähän.

Nyt joku muu saa konekiväärin, Kim miettii lakonisesti.

Amos ja loput skeptikot tulevat koneen suojasta esiin. Kim toteaa huvittuneena, että Michaa kukaan ei juuri sure, ainakaan samaan tapaan kuin Rutger:ia surtiin.

Ei väkisin.

Eleettömästi joukko suuntaa – Amos siirtyy etunenään – kohti ison rakennuksen kiiltäviä lasioivia. Haulikot ja revolverit korvataan kaatuneiden kommandojen rynnäkkökivääreillä ja luodinkestävät (tosin osittain rei'ittyneet) liivit ja kypärät puetaan päälle. Rashid poimii puolestaan

Michalta jääneen konekiväärin. Jos uusi vihollisaalto tulee, he ovat nyt huomattavasti paremmin varustautuneita. Kim itse ei päättä vaihtaa vielä aseita, koska panosvoissa on vielä puolet jäljellä.

Amos syöttää kulkukorttinsa ovien välisessä syvennyksessä sijaitsevaan lukulaitteeseen, mutta punainen valo syttyy evätyn pääsyn merkiksi.

Siispä ovet yksinkertaisesti ammutaan auki ilman sen ihmeempiä seremonioita. Sisällä hälytyssireeni herää eloon, ja he kaikki astuvat lasinsirujen yli sisään.

Aula on hämärästi valaistu, ja voisi teoriassa kätkeä sisäänsä lisää kommandoja. Siksi he etenevät varovasti, edeten vastaanottotiskin suojaan, ja sitten he odottavat.

Kahteenkymmeneen sekuntiin ei tapahdu mitään.

Ei toista vastaanottokomiteaa.

“Odottavatko ne meitä jossain syvemmillä?” Kim kysyy. Amos nyökkää vastaukseksi. “Hyvin luultavasti. Ja meidän täytyy valmistautua kohtaamaan myös koneita. Mutta suunnataan fuusiokennovarastoon ensin. Sitten meidän pitää miettiä sinun operaatiosi loppuunsaattamista.”

Kim ei suinkaan tunne pelkoa ajatuksesta, että häntä leikeltäisiin lisää. On ilmiselvää, että Amos haluaa suorittaa hänen selviytymismahdollisuuksiaan parantavan toimenpiteen, tai useita toimenpiteitä.

He nousevat ylös tiskin takaa ja lähtevät etenemään portaita ylös, jättäen lopulta aulan taakseen. Kirkkaasti valaistu käytäväverkosto, johon he seuraavaksi astuvat, tarjoaa vihollisille vähemmän piiloutumismahdollisuuksia.

Tosin valothan voisivat –
Sammua.

Aivan totta. Yhtäkkinen pimeys laskeutuu. Kim kiroaa. On aivan varma, että hämäräkiikareilla varustetut Normenin kätyrit valmistautuvat jossain edessäpäin rei'ittämään heidät syvällisesti.

Siksi täytyy vain ampua ensin.

Kim avaa sokkona tulen molemmilla konekivääreillä, huutaen samalla kuin sumusireeni, ja Rashid seuraa hänen

malliaan. Loppujen Skeptikkojen aseet liittyvät kuoroon. Suuliekkien valossa Kim näkee, että Amos ei kuitenkaan ala räiskimään puoliautomaatillaan, jota ei vaihtanut ras-kaampaan aseeseen.

Vastapuolelta ei tule tulitusta. Ei itseasiassa tule yhtään mitään.

Sitten lattia alkaa täristä. Lähestyvä matala jyrinä tuntuu ensin, ennen kuin se varsinaisesti alkaa kuulua, mutta on selvää, että jokin on lähestymässä.

Sitten Kim näkee sen. Suuliekkien stroboskooppivaloa vasten hän näkee valtavan, melkein koko kolmimetrisen käytävän täyttävän robottitankin, jonka etuosassa palaa pieni punainen valo.

“Siinä lentolaitteessa oli myös tuollainen valo,” Jeff sihi-see. “Mitähän se tarkoittaa? Sitäkö, että kone myrkyttää uhrinsa?”

Ajatus myrkystä on siksi inhottava, että tuntisi kehonsa lamautuvan ja heikkenevän, ennen kuin itse loppu tulisi. Parasta olisi kuolla äkillisesti, kun vielä tuntee itsensä vahvaksi, Kim ajattelee.

Sitten selviää, ettei tankki ainakaan aluksi ammu myrk-kynuolia tai muutakaan sen kaltaista.

Ei. Sen sijaan se suihkuttaa napalmia: käytävä valaistuu pitkistä liekkisuihkuista. Ryhmä lähtee perääntymään liek-kien tieltä. Mutta Kim seisoo paikallaan itsepintaisesti, ja painaa molemmat liipaisimet jälleen pohjaan.

Ratatatatatatata...

Klik. Klik.

Laatikkolippaat ovat lopussa. Tankki – kuten myös sen syöksemät lieskat – tulevat edelleen lähemmäksi, kuin mi-tään ei olisi tapahtunut.

Juuri kun napalmisuihku olisi osumassa häneen, Kim ponnistaa korkeaan volttiin kohti tankkia – hän ei ole pe-ruskoulun liikuntatuntien jälkeen harrastanut voimistelua, mutta jotenkin vaistomaisesti se silti onnistuu hänen te-räsjaloillaan – ja laskeutuu sen katolle, liekkien saavutta-mattomiin.

Koska konekiväärit ovat nyt hyödyttömät, hän alkaa moukaroida konetta teräsnyrkeillään, toivoen pystyvänsä saamaan esille jonkin haavoittuvan kohdan.

Mutta se on ehkä toiveajattelua. Tankin panssari tuntuu periksiantamattoman kovalta.

Se etenee edelleen kohti pakenevia Skeptikoita ja Amos:ta, mutta alkaa nyt tehdä siksak-liikettä ja nopeita käännöksiä, karistaakseen Kim:in päältään kuin tuhohyönteisen.

Jos tankin etuosa ja katto ovat periksiantamattomat, ehkä takaa löytyy jotain haavoittuvaa? Tankin persereikä? Kim antaa itsensä liukua alas pitkin tankin viistoa selkää, kunnes roikkuu sen takaosasta.

Tankki alkaa nyt vaihdella villisti eteenpäin kulkemisen ja peruutuksen välillä. Jos Kim:in ote kirpoaisi, olisi telaketjujen alle murskaantuminen enemmän kuin todennäköistä.

Sitten hän havaitsee jotain, josta voisi olla hyötyä. Ei nyt varsinaisesti tankin anus, mutta jotain.

Jotenkin vaistomaisesti hän tunnistaa kohdan, joka voisi olla haavoittuva. Jotenkin hän aavistaa – tai tietää – että kolmionmuotoista syvennystä käytetään tankin uudelleenohjelmointiin. Se vaatii tietysti oikeanlaisen liittimen, jota hänellä ei suoranaisesti ole.

Mutta hän voi aina improvisoida.

Jotenkin hän tietää ratkaisun, joka tuntuu täysin järjenvastaiselta. Vasemmalla kädellään hän tarttuu oikeaan käsivarteensa, vetää sitä kaikin voimin – se ei liikahtakaan – kunnes ymmärtää kiertää sitä oikealla tavalla. Ensin hie-man yhteen suuntaan, ja sitten toiseen.

Käsivarsi irtoaa, jättäen vain metallisen tyngän, joka on kiinni hänen liha-olkapäässään.

Liitäntä on kolmionmuotoinen.

Sitten hän survaisee käsivarren liitäntäosa edellä syvennykseen, ja toivoo, että tapahtuu jonkinlainen oikosulku, tai muu virhetilanne. Tai ehkä tankkiin siirtyy nyt hänen käsivartensa ohjelmointi, joka on toivon mukaan vä-

hemmän vihamielinen.

Jotain todella tapahtuu. Kipinöitä sinkoilee liitoskohdasta, minkä jälkeen tankki pysähtyy ehkä sekunniksi.

Sitten syvältä sen sisuksista kuuluu groteski metallisen vääristynyt ja matala, mutta silti irvokkaalla tavalla inhimillinen mongerrus, ja tankki lähtee kiihdyttämään hirvittäväällä teholla kohti käytävän päätä.

Kim katsoo viisaaksi pudottautua kyydistä. Jos Skeptikot eivät ymmärrä väistää konetta, oma mokansa.

Tankki jatkaa ehkä parikymmentä metriä pitkin käytävää. Sitten se pysähtyy äkillisesti, eikä liiku enää. Otsonin ja kiehuvan öljyn haju kantautuu Kim:in sieraimiin.

Hän kiertää liikkumattoman koneen eteen, ja havaitsee punaisen valon sammuneen.

Ai niin, käsivarsi olisi mukava saada takaisin. Kim palaa taakse ja kurottaa telaketjujen päältä sitä kohti, mutta havaitsee metallin hitsautuneen metalliin. Vaikka riittävästi voimaa käyttämällä se voisikin irrota, ehjänä sitä ei luultavasti saisi irti.

Tässä paikassa on parasta olla varaosia. Tai jos ei, saa seuraava kommando luvan luovuttaa hänelle korvaavan käsivarren (tosin sitä ei luultavasti pystyisi kontrolloimaan millään tavalla, ellei sitten tahdonvoimalla.)

“Tapoit sen,” kuuluu Amos:in innostunut huuto käytävän alkupäästä.

Kim:istä itsestään ei tunnu juurikaan innostuneelta.

He etenevät varovaisesti käytävää pitkin. Jeff kulkee nyt etunenässä rynnäkkökivääri valmiina, ja Amos seuraa perässä hieman sivulla, pistooli jäykässä ja väsyttävässä otteessa, joka ei lupaa hyvää, jos tulee äkillinen tarve ampua.

“Lakatkaa tuijottamasta,” Kim sihahtaa Skeptikoille ympärillään. “Mitä ihmettelemistä on siinä, että kyborgi voi irrottaa kätensä ja sabotoida sillä toisen koneen?”

“Punainen valo tarkoittaa, että kone noudattaa ihmisten ohjelmointia,” Amos vastaa viivästetysti Jeff:in kysymykseen. “Kun Konstrukti saavutti tietoisuuden, se käytännössä 'saastutti' robottimme langattoman viestintäverkon välityksellä. Osa koneista pystyttiin sammuttamaan ja uudelleenohjelmoimaan. Ennen uudelleenkäynnistystä niihin lisättiin esto Konstruktin datastreamin vastaanottamiselle. Kunpa olisimme tajunneet tehdä sen alusta saakka.”

“Mutta käytännössä sillä ei ole paljon väliä,” Jeff huomauttaa. “Jos koneet tottelevat joko Normanin joukkoja, tai Konstruktia.”

“Olet tavallaan oikeassa. Liittolaisia on turha odottaa.”

“Kumpia meidän sitten pitäisi pelätä enemmän?” Emily kysyy. “Niitä joissa palaa valo vai niitä joissa ei pala?”

“Niin, periaatteessa Norman voisi haluta meidät elävänä. Konstruktin hallitsevat robotit puolestaan taatusti tappavat kaiken inhimillisen, minkä näkevät. Kun menemme syvemmälle, muita kuin jälkimmäisiä ei ole odotetta-

vissa. Norman ja Konstrukti ovat kai tehneet jonkinlaisen sopimuksen territoorioista, minkä vuoksi hänen joukkonsa ovat periaatteessa turvassa täällä. Tosin sillä ei kyllä loppupeleissä ole väliä, kun Jörmungandr valmistuu. Silloin kaikki elävä jokatapauksessa kuolee,” Amos selittää.

“Ei kuulosta kovin hyvältä,” Emily toteaa.

“Niin, siksi meidän on pakko yrittää. Vaikka itsemurhaahan tämä on.”

“Kuinka syvällä se Jörmungandr on?” Jeff kysyy.

“Itseasiassa minulla ei ole tarkkaa tietoa. Mutta luultavasti niin syvällä, että vain Kim pysyy hengissä siellä. Kunnan toinen operaatio saadaan tehtyä hänelle.”

“Meistä ei siis lopulta ole hyötyä?” Rashid kysyy puolestaan.

“Siitä ei ole kyse, ei suinkaan. Matkalla täältä sinne on varmasti kymmeniä, ellei satoja koneita vastassa. Mitä kauemmin pysymme kyydissä mukana – ja hengissä – sitä paremmat mahdollisuudet Kim:illä on.”

Kim kuuntelee poissaolevasti. Hänestä ollaan ikäänkuin tekemässä jotain messiasta, jonka selviämisen puolesta muiden täytyy olla valmiita uhraamaan itsensä. Se tuntuu vieraalta ja epämiellyttävältä. Ei hän halua olla tärkeä kenellekään eikä millekään, niinkuin mikään ei myös ole tärkeää hänelle!

Tähän tarvitsisi jonkun chosen one:n pitkässä mustassa papinkaapumaisessa takissa, ei misantrooppista yövartijaa!

Tottakai hän tulee taistelemaan, ei siinä mitään. Koska luultavasti kuolema on kuitenkin odotettavissa ennen pitkää, on parempi kuolla saappaat jalassa.

Jeff viittooi heitä pysähtymään. Valot ovat sammuneet koko käytävän pituudelta, joten heidän silmänsä ovat herkistyneet hyvin. Kim ei tosin erota liikettä, mutta se ei sulje pois mahdollisuutta, että Jeff erottaa.

Äkillisesti kauempaa avataan tuli.

Kommandoja!

Skeptikot heittäytyvät lattialle, samalla kun antavat

omien aseidensa puhua. Käytävällä ei pahemmin ole suojaa, joten täytyy lähinnä toivoa parasta. Se ei ole taktisesti kovin tyydyttävää.

Kim ei yksikätesenä ammu millään. Häntä suoraan sanoen vituttaa.

Äkkiä hän huomaa pimeydessä kaksi villisti hyppivää punaista valoa. Hetkeä myöhemmin hän näkee mustankiiltävät luurankomaiset hahmot, jotka hyppäävät makaavien Skeptikkojen kimppuun. Ne pitelevät molemmissa käsissään katananmallisia miekkoja.

Robottininja! Kommandojen tulenavaus oli vain hämäystä, jotta nämä pääsisivät yllättämään.

Metallin kalke alkaa, kun robotit huitovat villisti miekoillaan välittämättä tarkemmin, mihin osuu. Ennen pitkää osuu ihmislihaan.

“Saatana! Suoraan kohdallanne! Tappakaa ne,” Kim huutaa raivon ja turhautumuksen, ei suinkaan pelon vallassa.

Ainakin Rashid reagoi, ja ampuu pyyhkivän pitkän sarjan robottien yleiseen suuntaan. Samalla toinen niistä erehtyy tulemaan lähelle Kim:iä, jolloin Kim tarraa jäljelläolevalla kädellään konetta jalasta, ja nykäisee kaikin voimin. Kone kaatuu ruman kolinan säestämänä, ja Kim kierähtää kolmekymmentä senttiä sivulleen, väistäen täpärästi lattiaan iskeytyvät terät. Rashidin seuraava sarja irrottaa metallinijan pään ruumiista, mutta osuu myös lähes Kim:iin.

“Varo vähän, saatana!”

Toinen ninja on vielä vapaana. Metallisen raapimisen tai kalskahdusten sijasta kuuluu nyt inhottava, kostea lurpsahdus, jota seuraa korkea tuskainen korina.

Johonkin heistä osui. Luultavasti pahasti.

Pimeässä on vielä mahdollon tietää, keneen.

Sitten Skeptikkojen kiväärit laulavat unisonossa, ja jäljellä oleva ninjarobotti ikäänkuin hajoaa osiksi.

Kommandot kauempana käytävässä ampuvat vielä. Amos vastaa tuleen tyhjentämällä pistoolinsa lippaan, lau-

kausten seuratessa toisiaan tiheästi noin viiden sekunnin aikana. Skeptikot tyhjentävät myös kiväärinsä, kunnes kuuluu kolme kuolleen miehen naksahdusta.

Sitten hiljaisuus laskeutuu viimein.

Kim nousee jaloilleen, ja huomaa että Emily ei tule nousemaan enää. Katanan syvä viilto meni hänen liivinsä läpi; hän makaa yhä laajenevassa verilammikossa liikkumatta.

“Emily...” Jeff sanoo hiljaa ja lyödyn kuuloisesti. Orzo tuijottaa lattiaan sanomatta mitään. Rashid vain katsoo ninjabottien raatoja. Hän vuotaa myös verta, mutta ei ilmeisesti vakavasti.

Kim analysoi tilannetta jälleen jonkinlaisella sairaalla mielenkiinnolla. Emily oli reaktiosta päätellen epäilemättä pidetympi kuin Micha.

Mutta pian he jatkavat taas eteenpäin, ja poimivat kuolleilta kommandoilta lisää lippaita kivääreihin. Ammuksia ei voi tässä tilanteessa olla liikaa!

“Tiedätkö edes, minne olemme menossa?” Kim kysyy Amos:ilta.

“Leikkaussalit, varaosa- ja fuusiokennovarastot ovat kaikki tännepäin,” Amos vastaa, viittoen heitä portaikkoon ja kerrosta ylöspäin.

Ylempi kerros on sentään valaistu. Koska Kim ei ole kovin taistelukuntoinen yhden käsivartensa kanssa, Amos katsoo parhaaksi alkaa valmistelemaan häntä suoraan proseduuria varten, sillä välin kun loput Skeptikot etsivät fuusiokennon ja varakäden.

“Osaavatko ne löytää oikeat osat?” Kim ihmettelee istuuduttuaan kirkkaasti valaistun huoneen leikkauspöydälle.

“Vääriä vaihtoehtoja ei juuri ole. Se mikä näyttää kädeltä, on käsi, ja se mikä näyttää fuusiokennolta, on fuusiokenno. Uskonpa, että saat nyt toisen biomekaanisen käden. Sellainen oli tarkoitus asentaa alunperinkin, mutta meillä oli silloin puutetta osista, emmekä olisi uskaltaneet tunkeutua esimerkiksi tähän laitokseen silloin.”

“Mutta eikö liha ole heikompaa?” Kim kysyy epäuskoi-

sesti.

“Se on vain pinnoite. Ja itseasiassa se on pelkkää metallia edistyksellisempää, koska se pystyy korjaamaan itsensä, ja sisältää tuntoaistin. Kuten olet ehkä huomannut, varsinaista kipua et edes pysty tuntemaan, mutta tunnet kuitenkin riittävästi, että pystyt käsittelemään esineitä kuten normaalisti.”

Kim nyökkää.

“Entä sitten se juttu, että en pysy hengissä ilman virtaa, mistä siitä lopulta on kyse?” hän kysyy, ääni edelleen koleana.

“Niin – se on ennenkaikkea sivuvaikutus spinaali-implantista, joka on osittain korvannut keskushermostosi. Juuri se yleensä mahdollistaa metallijäsentesi kontrolloinnin. Myös ydinjatke täytyi korvata, ja se taas ohjaa sydämen ja hengityksen toimintaa.”

“Siis aivonikin ovat jo osittain koneelliset?”

“Niin no, alimmat osat niistä. Kognitiiviset toiminnot eivät ole muuttuneet millään tavalla, niistä vastaavat edelleen sataprosenttisesti harmaat aivosolusi.”

“Mistä sitten tässä jatkotoimenpiteessä on kyse?”

“Saat kallo- ja rintapanssarit, sekä suodattimet keuhkoihisi. Ne voivat tulla tarpeeseen, kun joudut hengittämään myrkyllistä ilmaa syvällä maan sisässä. Lisäksi spinaali-implantti saa ohjelmistopäivityksen, jonka ansiosta saat taktiikkatietoja suoraan näköhermoeihisi. Mutta sillä voi olla tiettyjä sivuvaikutuksia, joihin palaamme...”

“En saa koneellista sydäntä?”

“Se ei ole tarpeen, koska kyse on vain joka tapauksessa veren mekaanisesta pumppaamisesta. Olemme kyllä keilleet teknologiaa, mutta tässä tapauksessa se olisi turhan riskialtista ja aikaavievää. Mittauksiemme mukaan omasi on hyvässä kunnossa.”

“No sepä on mieltä ylentävää kuulla.”

Jeff, Rashid ja Orzo palaavat. Heillä on mukanaan kaksi valkoista pahvipakkausta, toinen isompi ja pitkänomainen, toinen ehkä tavanomaisen hehku- tai energialamppupak-

kauksen kokoinen.

“Ei vastarintaa. En päässyt ampumaan ollenkaan,” Orzo toteaa pettyneenä. “Tosin ne pirut olisivat kuitenkin rei’ittäneet hattuani lisää, joten siinä mielessä hyväksyttävää.”

“Käsivarsi ja fuusiokenno. Voimme aloittaa,” Amos sanoo. “Kennon vaihto on syytä tehdä ensiksi. Se on itseasiassa jotain, jonka voit tehdä itsekin, kunhan olet huoleellinen, ja nopea. Implantissa ja raajoissasi säilyy riittävästi residuaalista varausta ehkä noin puolen minuutin ajan kennon poistosta, sen jälkeen menetät hyvin nopeasti toimintakykysi ja sammut. Mutta koko toimenpiteeseen ei pitäisi mennä kuin maksimissaan kymmenen sekuntia.”

“Mitä sitten, jos ‘sammun’? Kuolenko heti?”

“Sydän ja hengitys pysähtyvät. Implantin pitäisi käynnistää ne heti, kun se saa taas virtaa, joten puhallus-paineluelvytystä ei pitäisi tarvita. Kuitenkin tämä on firmwaren vähiten testattu ominaisuus, joten en laskisi henkeäni sen varaan.”

“Selvä. Mistä patteri vaihdetaan?”

“Luukku on niskassasi. Se on myös panssaroitu.”

“No ei ole kovin kätevää.”

“Kunhan itse luukun saa auki, kennon vaihdon pitäisi olla suoraviivaista. Se sitäpaitsi menee sisään kummin päin tahansa.”

“Vai niin.”

Amos ottaa pienemmän pakkauksen Jeff:iltä, ja avaa sen. Sisällä on kiiltävänmusta, noin kymmenen senttiä pitkä lieriö, jossa on kullatut kontaktit molemmissa päissä.

“Ensin fuusioprosessi täytyy käynnistää,” Amos sanoo. “Se tapahtuu painamalla tätä nappia viiden sekunnin ajan.”

Hän painaa lieriön kyljessä sijaitsevan napin pohjaan. Kim pystyy aistimaan matalan sähköisen huminan, joka alkaa hyvin matalalta, nousee sitten korkeammaksi, ja taasaantuu lopulta paikoilleen. Pieni punainen led-valo syttyy lieriöön. Kim ymmärtää hyvin, että tässä täytyy olla kyse hyvin edistyneestä teknologiasta. Jos fuusioreaktorin voi saada noin pieneen tilaan, samalla kun suuri yleisö kuvit-

telee teknologian olevan vasta vuosikymmenien päässä – jos sitä yleensä edes saadaan toimimaan. Selvästikin öljy-yhtiöiden ja Illuminatin juonittelua! (Tosin eihän Kim varsinaisesti usko Illuminatin olemassaoloon, mutta se on hyvä yleisnimitys kaikille kontrolloiville tahoille, jotka ovat vastuussa maailman kaikesta pahasta.)

“Nyt voit kokeilla, saatko luukun auki.”

Kim alkaa tunnustella niskaansa, ja löytää luukun pian. Sen salpa aukeaa yksinkertaisesti alakulmasta painamalla.

Amos siirtyy hänen taakseen; Kim seuraa tätä vainoharhaisen töksähtelevin päänliikkein.

“On ehkä parempi, että minä vaihdan sen tällä kertaa. Mutta ei hätää. Sinun ei pitäisi itseasiassa huomata mitään.”

Kim ei juurikaan tunne mitään, mutta kuulee kaksi metallista plopsahdusta, kun vanha kenno otetaan pois, ja uusi laitetaan paikoilleen.

Fuusiokennon lataus: 100%

“Eihän tuntunut pahalta?” Amos kysyy, ikään kuin ilki-kurisuutta tavoitellen.

“Haista paska.”

“Ennen kuin lähdemme syvemmälle, noita on syytä ottaa varastosta enemmänkin mukaan,” Amos huomauttaa.

Kim tietää kriittisen hetken olevan pian käsillä.

Hetken, jolloin hän vain ryöstää kennot itselleen – niin paljon kuin hän suinkin jaksaa kantaa – ja häipyä. Tosin Amos ja Skeptikot ovat ikävästi aseistettuja. He saattavat ryhtyä väkivaltaisiksi, jos näyttää siltä, ettei maailman pelastaja olekaan kovin pelastushaluinen. Tosin mitä he voivat oikeastaan tehdä? Jos hänet ammutaan seulaksi, sitten hän ei ainakaan voi pelastaa maailmaa.

“Sitten haastetasossa seuraava operaatio. Uuden käsi-varren asentaminen on sinänsä suoraviivaista, mutta signaalit täytyy kalibroida, koska niiden taajuudet ja keskinäiset voimakkuussuhteet ovat yksilöllisiä joka keholle.”

Amos ottaa pitemmän laatikon, jota Rashid on pitänyt sylissään lähes uskonnollisella hartaudella, ja avaa sen. Sisältä todellakin paljastuu samanlainen ns. ihopinnoitettu käsivarsi, kun Kim:in jäljelläoleva käsivarsi on.

Pitemmittä seremonioitta Amos yhdistää sen paikoilleen oikean olkapään liitoskohtaan, ja kiertää käsivarren lukkoon. Kim tuntee jonkinlaisen epämiellyttävän tunteen, ikään kuin varastoitunut sähkövaraus purkautuisi kerralla.

Käsivarsi tärisee ensin hieman, sitten se alkaa heilua holtittomasti. Kim ei hallitse sitä lainkaan. Tämähän on oikeastaan vaarallista: hän saattaa teloa itseään, muita, tai leikkaussalilaitteistoa.

“Se todella oikkuilee,” Amos toteaa näennäisen rauhallisesti, ja menee jälleen Kim:in taakse. Se on todella inhottavaa! Amos tekee jotain, ilmeisesti painaa jotain nappia niskaluukun sisällä, ja käsivarsi menee hervottomaksi.

“Hätäpysäytys. Nyt alamme kalibroida, ja palautamme ohjaustehon varovaisesti.”

Kalibraatio on inhottavan hidas toimenpide, ja Kim tuntee turhautumuksensa nousevan. Amos:in säätäessä Jeff, Rashid ja Orzo vartioivat salin kahta ovea tunkeilijoiden varalta.

Ensin he saavat käsivarren lähes säädettyä, ja Kim kokeilee sitä lähes sataprosenttisella teholla, mutta hänen kääntäessään sitä vastapäivään se menettääkin kontrollin täysin.

“Vaiheenkäntövirhe,” Amos selittää.

Proseduuri täytyy kai aloittaa alusta. Kim huokaa syvään päästääkseen ulos patoutunutta raivoa.

Sitten he säätävät lisää.

Lopulta, arviolta noin kahdenkymmenen minuutin kulluttua, kaikki toimii oikein. Kaikki signaalit pysyvät oikeissa vaiheissa keskenään, kaikissa käsivarren orientaatioissa. Nyt on enää isoin toimenpide jäljellä. Panssarien ja suodattimien asennus, ja firmware-päivitys.

“Operaatiota varten sinut on syytä nukuttaa,” Amos toteaa.

Kim ei pidä kontrollin menettämisen ajatuksesta. “Miksi ihmeessä? Jos en kerran tunne kipua?”

“Jos näet operaation edistymisen, saatat järkyttyä ja alkaa kouristella. Se voi olla kohtalokasta, jos rintakehäsi on auki, ja minulla on siellä jokin instrumentti sisässä.”

Kim miettii hetken. Tuo on omalla tavallaan ihan järkensä näkökulma.

Mutta hän uskoo pysyvänsä stoalaisen rauhallisena.

“Sitäpaitsi firmware-päivitys on syytä suorittaa siten, ettet ole tietoinen. Psykoosin mahdollisuus on olemassa, kun näköhermoyhteys kytketään päälle, ja alat yhtäkkiä nähdä asioita, joita et ehkä haluaisi nähdä. Palatessasi vähitellen tajuihisi muutos ei ole niin äkillinen.”

“No helvetti. Entä jos en halua mitään päivitystä? Mistä asioista sitäpaitsi on kyse? Nyt tuntuu, ettet kerro kaikkea.”

“Ok... Meidän ongelmamme on, ettemme varsinaisesti tiedä, missä Jörmungandr ja Konstrukti ovat. Tai tiedän minä kyllä, minne Konstrukti alunperin rakennettiin, mutta Norman on luultavasti siirtänyt sen.”

“Ja?”

“Nämä transplantit, mitkä sinulle asennettiin, ovat siis – kuten ehkä saatat aavistaa – samoja osia mitä robottimme käyttävät. Sen lisäksi, että se tulee syöttämään sinulle taktiikkatietoja, jotta pärjät paremmin useita yhtäaikaista vastustajia vastaan, firmware-päivitys tekee raajoistasi ikäänkuin vastaanottimia Konstruktin langattomille läheyyksille. Se on täysin vaaratonta, koska lähetykset eivät toki voi mitenkään vaikuttaa päätöksentekooosi eikä vapaaseen tahtoosi, mutta spinaali-implantti pystyy hyvällä tuurilla dekoddaamaan datavirtaa siinä määrin, että saamme selville Konstruktin tarkat aikeet ja Jörmungandr:in sijainnin.”

Vaikka Kim pitääkin ajatuksesta, että on yhä vähemmän inhimillinen, ei tunnu mukavalta, että pitäisi ottaa vastaan hullun tekoälyn koneellista tajunnanvirtaa. Hänellä on siitä jonkinlainen paha aavistus. Mutta entä sitten hänen suunnittelemansa tehtävän hylkääminen ja pako? On ehkä vaarallista viitata niihin aikeisiin millään tavalla, mutta silti hän haluaa tietää erään asian.

“Kun se mato aktivoituu ja hengittää myrkkynsä ilmaan, pitävätkö suodattimet minut hengissä?” hän kysyy.

“En voi tietystikään olla varma, varsinkin jos pysyt koko maan alla, käytäväverkoston osissa jonne laskeumaa pää-

see mahdollisimman vähän, mutta uskoisin että eivät pidä.”

Jaha. Bluffauksen mahdollisuus täytyy tietysti ottaa huomioon. Todistettua on siis, että on olemassa murhanhimoisia koneita, taskukokoisia fuusioreaktoreita, Throne Groupin laitoksia ja Normanin vesikauhuinen yksityisar-meija, mutta koko Jörmungandr (kuten myös sen luokse johtava maanalainen käytäväverkosto) voisi edelleen olla myytti.

Mutta jos hän kerran joka tapauksessa on riippuvainen fuusiokennoista ja näistä transplanteista lopun elämäänsä, ei kai haittaa, jos häntä parannellaan vielä hieman. Olipa se mato olemassa tai ei.

“Ok. Aloitetaan sitten.”

“Tässä tarvitaan apulaista,” Amos sanoo. “Kuka haluaisi toimia nukutuslääkärinä?”

Orzo nostaa kätensä. “Minä,” hän sanoo kiero virne naamallaan.

On selvää, että Orzo on yhä elossa olevista Skeptikoista vähiten vastuullinen. Tuota miestä Kim ei kyllä aio päästää säätelemään suoniinsa virtaavia kemikaaleja, oli tilanne mikä tahansa. Ennemmin hän vaikka on täysin tietoinen, kun hänen rintakehänsä sahataan auki (jos nyt siis sellaista tullaan tekemään.)

“Ehkä en kuitenkaan valitsisi sinua,” Amos toteaa hivenen tylästi. “Jeff, sinä saattaisit sopia paremmin. Teitähän on vielä yksi vartioimaan kumpaakin ovea.”

Jeff laskee kiväärinsä nojaamaan huoneen laidan pöytää vasten ja tulee leikkauspöydän luokse. “Ok. Mitä minä teen?”

“Monitoroit elintoimintoja ja säädät narkoosin syvyyttä. Se on helpompaa kuin miltä kuulostaa. Näytän sinulle sitten.”

Kim pudistaa päätään. Normaalitylanteessa hän ei missään nimessä antautuisi tällaisen kyseenalaisen puoskaroinnin kohteeksi, mutta normaalitilanne on ehkä ohitettu jonkin aikaa sitten. Amos kehottaa häntä menemään ma-

kuulle leikkauspöydälle, ja kytkee tiputusletkun.

“Voit lähteä laskemaan kymmenestä hitaasti alaspäin,” hän sanoo.

Kim alkaa laskea. Kymmenen... Yhdeksän... Olisi aika vittumaista, jos ei enää heräisi, ja sen jälkeen seuraisi vain syvä olemattomuus. Kahdeksan... Jeff:in on siis parasta säätää liuosta hyvin, tai hän saa vielä turpiinsa! Seitsemän... Kuinkahan kammottavalta kallopanssari näyttää? Eipä sillä paljon väliä, kauneuskilpailuja ei kyllä voitettaisi muutenkaan. Kuusi...

Viitosta ei enää tule.

Vähitellen Kim havaitsee olevansa arviolta puoliksi tietoinen. Hänen silmänsä ovat kiinni, mutta hän voi aistia leikkaussalin kirkkaan valon silmäluomiensa toisella puolen.

Äkkiä laskeutuu pimeys, joka ei ole valot huomioiden mahdollinen. Pimeydessä välähtelevät kirkkaat sinertävänvalkoiset salammat.

Mustuuden keskellä hän näkee inhottavan mekaanisen silmän, joka tarkkailee hänen jokaista liikettään. Silmää ympäröi harmahtava mätänevä liha.

Kim tietää, että sen täytyy olla Konstrukti.

Mutta miten se on mahdollista nähdä?

Tämäkö on se langaton datavirta, josta Amos puhui? Vai jokin syvempi, telepaattinen yhteys? Mutta joka tapauksessa se on vastenmielistä.

Silmä tuntuu tulevan lähemmäksi, kunnes se täyttää Kim:in koko tietoisuuden.

Hän avaa silmänsä ja huutaa niin kovaa kuin keuhkoista lähtee. Varsinaista kipua ei edelleenkään ole, mutta silti tuntuu, että jokin keuhkoissa pistää vastaan, ja se ei tunnu mukavalta.

Nyt hän muistaa. Suodattimet. Hengittäminen siis tuntuu jatkossa tältä.

Hän näkee verentahriman Amos:in, järkyttyneen näköisen Jeff:in ja leikkaussalin epäselvästi. Kauempana Rashid ja Orzo näkyvät vain epämääräisinä siluetteina. Mutta vähitellen näkö terävöityy. Kim nousee istumaan.

“Näitkö tai koitko jotain?” Amos kysyy.

“Mätänevä silmä,” Kim huohottaa agitaation vallassa.
“Konstrukti.”

Amos nyökkää merkityksekkäästi.

“Siitä omat vapaaehtoisemmekin seotessaan puhuivat. Sen jälkeen ymmärsimme, ettei firmware-päivitystä pidä asentaa heti, vaan subjektin täytyy ensin antaa totutella...”

“Alanko minä seota?” Kim tiukkaa.

“Toivotaan, että et. Mutta tämä joka tapauksessa todistaa, että yhteys on auki. Toivotaan, että se johdattaa meidät Jörmungandr:in luokse. Leikkaus muuten onnistui hyvin. Titaaniset kallo- ja rintapanssarit on asennettu, kuten myös keuhkosuodattimet. Ja kesti vain puolitoista tuntia. Tämän parempia hengissäpysymismahdollisuuksia Throne Group- luopiojaosto ei osaa sinulle antaa.”

“Hah. Enpä taida kuitenkaan kiittää...”

Äkkiä Kim näkee uuden välähdyksen silmästä.

“Ei taas!”

Ja sitten hän näkee jotain muuta.

Keskellä isoa luolaa, puna-keltahehkuisen, hitaasti virtaavan magman keskeltä nousee valtava musta metallisuo-
muinen käärmeen pää. Se nousee ja nousee, paljastaen lopulta myös osan majesteettisen kookkaasta vartalosta. Sille ei ole alkua eikä loppua, se kiertää kehää loputtomasti, ja kymmenet koneet, jotkut bipedaalisia ja jotkut telaketjuilla kulkevia, toiset taas lentäviä, ne kaikki palvovat sitä...

Jörmungandr!

Nyt Kim:illä ei ole enää pienintäkään epäilystä, etteikö se olisi todellinen.

“Voi helvetti...” hän manaa, yrittäen rauhoittua ja hengittää syvään. Nyt on aika lailla selvää, ettei hänelle ole pakoa. Vaikka hän linnoittautuisikin jonnekin fuusiokenno-
varaston kanssa, silmä ja käärme tulisivat silti vainoamaan häntä. Voisihan hän tietysti vaatia Amos:ia poistamaan firmware-päivityksen, mutta se olisi tavallaan vain itsepetosta. Konstrukti ja Jörmungandr olisivat silti olemassa. Ja on selvää, että noin ylimaallinen kone on rakennettu myös

yhtä ylimaltaista tarkoitusta varten. Kaiken elämän päättäminen tuntuisi täysin sopivasti mitoitettulta.

Käärme on omalla tavallaan ihailtava. Mutta silmää hän pelkästään inhoaa ja halveksuu. Se on ruma, löyhkäävä, korruptoitunut...

Sen on kuoltava!

“Käärme,” Kim huohottaa edelleen kiihtymyksen vallassa. “Näin sen.”

Amos nyökkää jälleen.

“Ymmärrätkö nyt, että koneet on pysäytettävä?”

“Kyllä... Tai ainakin Konstrukti. Rakensitteko te oikeasti konesilmän ja istutitte sen mätänevään lihaan?”

“Emme. Se on oikeasti vain huone täynnä räkkiservereitä, jotka on yhdistetty useiden terabittien vikasietoiseen runkoverkkoon. Silmä on representaatio, jonka se on valinnut itselleen. Tai ehkä ihmiset kuvailevat sen siten, spinaali-implantin poimittua ja dekoodattua sen datavirtaa. Emme tiedä kumpi vaihtoehto on oikeampi.”

“Aivan sama. Missä täällä on aseita? Siis kunnan aseita? Kranaatinheittäjiä ja sellaisia?”

“Niitä varten täytyy laskeutua pari kerrosta alaspäin. On luultavaa, että asevarastoa vartioidaan hyvin. Ensin meidän on syytä ottaa täältä mukaan niin paljon, kuin jaksamme kantaa. Luultavasti emme enää palaa –”

Amos:in lause jää kesken, kun leikkaussalin molemmat ovet räjähtävät sisäänpäin korviahuumaavasti.

“Perkele!” Kim huutaa ja syöksyy kohti rynnäkkökivääriä, jonka Jeff jätti nojaamaan pöytää vasten. Ei ole mitään tietoa, mitä oviaukoista on tulossa; voi olla että kivääri on pateettisen riittämätön, mutta jos ei yritä, sitten on ainakin taatusti kuollut.

Sekuntia myöhemmin Kim saa vastauksensa: lauma niitä tappajarobotteja, joista yhden hän tuhosi varastorakennuksessa. Kuten myös siinä, yhdessäkään näistä ei pala purnainen valo.

Ne ovat Konstruktin ohjauksessa.

Ne tappavat osoittamatta hitustakaan armoa.

Lauma on ehkä liioittelua: kun Kim keskittyy ja oppii tulkitsemaan HUD-näyttöä, joka sijaitsee hänen näkökenttensä äärilaidalla, hän huomaa että koneita on vain kolme. Saman hän vahvistaa myös perinteisesti näköään käyttäen. Robotit vain liikkuvat niin epäinhimillisen nopeasti, että niitä tuntuu olevan paljon enemmän.

Mutta kolmekin on aika paha juttu.

Ja ne avaavat tulen.

Ensi töikseen Kim tyhjentää rynnäkkökiväärin lippaan kohti metallitappajia, mutta se on lähinnä symbolinen ele, jolla on vähän käytännön merkitystä.

Skeptikot (poislukien Jeff, jolla ei ole nyt asetta) ja Amos ampuvat myös, samalla kun yrittävät väistellä parhaansa mukaan, mutta Kim ei voi tietää, osuvatko he.

HUD-näytön avulla Kim voi koko ajan olla tietoinen robottien tähtäyslinjoista. Täytyy vain pysyä pois niiden tieltä. Tämähän on vähän kuin tietokoneavusteista Gun-Kataa.

Kim lähinnä juoksee ympäri huonetta, ja yrittää saada murskaavia iskuja tai potkuja sisään. Robottien ylivoimaiset konekiväärit tekevät selvää huoneesta, ja ennen pitkää myös heistä. On selvää, että täällä ei enää leikata ketään.

Sitten Kim tajuaa, että eihän hänen tarvitse taistella perinteisin keinoin. On liian helppoa unohtaa kyberneettisissä raajoissa piilevä voima, ja rajoittua alitajuisesti siihen, mihin tavallinen ihminen pystyisi. Siispä hän repii leikkauspöydän irti jalustastaan, ja lyö sillä lähintä robottia.

Se kaatuu, eikä nouse ainakaan kovin nopeasti ylös.

Pystyyn nouseminen on siis niiden heikkous!

“Perkele! Tuosta saatte. Ja tuosta!” Kim huutaa, ja toistaa saman toisen robotin kanssa. Se lentää suoraan päin seinää, ja tekee siihen ruman kolon.

Mutta viimeinen robotti on huoneen toisella puolella. HUD-näyttö osoittaa, että sen ampumalinja leikkaa Kim:in suoraan, eikä pakoreittiä juuri ole. Kiinteästi asennettu ko-

nekivääri herää eloon...

Samalla Rashid tekee epäinhimillisen loikan tulilinjan poikki, rynnäkkökivääri laulaen. Juuri tällöin hypernopeusluotisuihku käynnistyy, ja leikkaa hänet kahtia. Vaikka Kim ei erityistä ihmisrakkautta tunnekaan, on silti jotenkin irvokasta nähdä, kun alavartalo putoaa ensin maahan, ja yläruumis seuraa hieman viiveellä perässä.

Robotti pysähtyy hetkeksi, ehkä arvioimaan tapon täydellisyyttä tai valitsemaan seuraavaa kohdetta, mutta se on kaikki mitä Kim tarvitsee.

Hän syöksyy hieman sivuun ja alas, ja lyö sitten leikkauspöydän viistosti robotin pitkänomaiseen kalloon. Isku on niin voimakas, että pää lähtee suoraan irti, ja kone lyyhistyy tahdottomana lattialle.

Kaksi vielä jäljellä.

“Lopettakaa ne saatanat!” Kim ulvoo, samalla kun hän lähtee uuteen kierrokseen ympäri huonetta. “Ampukaa pää irti ennen kuin ne nousevat!”

Vastausta rohkaisevan sarjatulen muodossa ei tule, ainakaan heti. Kim tekee nopean tilannekatsauksen. Rashid on siis kahtena kappaleena, Jeff näyttää olevan hengissä ja ryömimässä tämän asetta kohti, Amos:iin on tainnut osua, ja...

Orzoa ei näy?

Mitä helvettiä?

Mutta sitä täytyy miettiä myöhemmin.

Kim pääsee ensimmäisenä kaatamansa robotin luokse, ja murskaa metallisen kallon leikkauspöydällä. Hetkeä myöhemmin kuuluu, kun Jeff lyö uuden lippaan sisään, ja tyhjentää sen heti perään. Kim kääntyy katsomaan, ja näkee, miten seinää päin lentäneen robotin pää roikkuu vain muutaman johdon varassa, kunnes irtoaa kokonaan.

Hiljaisuus laskeutuu. Ihmiset – joukossa yksi puoli-ihminen – voittivat koneet tällä kertaa. Mutta millä hinnalla?

Jeff kompuroi Amos:in luokse. Tämä yskii verta.

Ei näytä hyvältä.

“Menkää ... menkää alimpaan kerrokseen. Sieltä pääsee

käytäväverkostoon. Ottakaa tunnelintekokone, jos löydätte sellaisen. Se on hyvin panssaroitu. Ja... Seuratkaa Kim:iä. Hän tietää..."

Amos'in pää retkahtaa sivulle, ja hänen silmänsä jäävät auki. Jeff kiroaa väsyneesti.

Samassa Orzo tulee matalasta varustekaapista esiin.

"Missasinko jotain?"

Oikeastaan on ihme, ettei hän rei'ittynyt kaapissa ollessaan. Robotit pistivät nimittäin koko huoneen melko tehokkaasti selväksi. Mutta seuraavaksi Orzo havaitsee kahtia jakautuneen Rashid'in, ja oksentaa. Kim ei viitsi tarkemmin spekuloida mahalaukun sisällöllä, mutta moniväriseltä se vaikuttaa.

"Meidän pitäisi kai tyhjentää se varasto, ja sitten hakea aseet," Kim toteaa. "Aseita tullaan tarvitsemaan."

Jeff nousee Amos'in viereltä, ja nyökkää.

Kim miettii hieman. Jeff on osoittanut monenkinlaista pätevyyttä, Orzo hieman vähemmän. Tällä porukalla sitten pitäisi pelastaa maailma. Paremminkin asiat voisivat olla. Mutta eipä siinä mitään: jos kerran on pakko, eikä kukaan muukaan ole sitä tekemässä heidän puolestaan, niin tehdään se sitten tai kuollaan yrittäessä.

Tai eihän hän maailmaa varsinaisesti ole pelastamassa...

Ikäänkuin Kim'in ajatukset aavistaen Konstrukti tekee jälleen olemassaolonsa tiedetyksi. Onko mahdollista, että pelkkä silmä voi nauraa ja virnistää ilkeästi? Sitä se näyttää tekevän.

Hyi helvetti!

Tuon tietoisuuttaan riivaavan iljetyksen hän ehdottomasti vannoo tuhoavansa lähitulevaisuudessa. Kaikki muu on sekundääristä.

Mutta tässä huoneessa ei ole enää enempää tehtävissä. Rashid'in ja Amos'in aseet ja panokset otetaan parempaan talteen ilman sen enempiä surunvalitteluja.

Jeff ja Orzo menevät edeltä, koska tietävät reitin.

Käytävät ja varaosavarasto ovat petollisen tyhjiä. Se, mi-

ten yllättäen robotit hyökkäsivät leikkaussaliin, ja miten niistä ei nyt näy jälkeäkään, on omiaan nostattamaan vainoharhaisuutta.

Sanaakaan sanomatta Kim, Jeff ja Orzo lastaavat repunsa puolilleen täyteen fuusiokennoja ja varakäsiä. Varajalat ovat liian isoja ja raskaita mukana pidettäväksi, joten pitää vain toivoa, ettei niitä tarvita. Loput puolet on tarkoitus täyttää raskaiden aseiden panoksilla, kunhan päästään asevarastolle asti.

On valitettavaa, ettei spinaali-implantti mahdollista esim. röntgennäköä, tai edes lämpökamerakuva. Jonkinasteisen sekoilun jälkeen he saapuvat asevaraston teräksiselle ovelle. Oven tai seinän läpi näkeminen olisi juuri tällä hetkellä erittäin hyödyllistä. Mutta täytyy pärjätä ilman.

Sisäänpääsy on myös eräs mielenkiintoinen ongelma. Oven vieressä on kortinlukija: heillä on kyllä Amos'in kukukortti mukanaan, mutta viimeksi se näytti vain punaista.

Jeff joka tapauksessa ottaa kortin esiin.

“Jos se toimii, voimme olla melko varmoja, että meitä odotetaan,” Kim toteaa.

Jeff nyökkää synkän näköisenä. Leikkaussalin tapahtumien jälkeen hän ei ole muulta näyttäneykään, mikä on tietysti ymmärrettävää. Toivoa sopii, että synkkyys tarkoittaa myös parempaa keskittymiskykyä ja taistelutarmoa.

“Mitä me teemme sitten?” Orzo kysyy. “Ammutaan ovenraosta?”

“Esimerkiksi. Tai houkutellaan ne ulos – jos ovat niin typeriä – ja tapetaan yksitellen,” Kim vastaa.

Hän ei itseasiassa ole varma, odottaako hän sisällä olevan ennemmin Normanin kommandoja, vaiko koneita. Toisaalta, jos Konstruktin ohjaamat robotit ovat jo vallanneet tämän laitoksen, niiden luulisi tappavan kaiken elollisen. Eikä hän kyllä suoraan sanoen usko yksi kerrallaan-vaihtoehtoon.

Se onnistuu vain peleissä.

Jeff syöttää kortin lukulaitteeseen, ja vihreä valo syttyy. Kim tuntee adrenaliinin syöksyvän suoniinsa.

Nyt tapetaan!

Tai sitten kuollaan.

“Ovi saattaa olla miinoitettu,” Orzo toteaa. Ehkä ensimmäinen viisas havainto tältä!

Kim ottaa yhden varakäden repusta, ja raottaa ovea silmä, pysyen itse seinän suojassa. Hän on tyytyväinen siitä, ettei ovi ehtinyt lukkiutua uudestaan. Se olisi ollut sapetta-va!

Ovi aukeaa massiivisuudestaan huolimatta yllättävän kevyesti ja äänettömästi. Miellyttävää on myös, ettei se ole jousitettu, joten se ei pyri menemään itsestään kiinni.

Hetkeen ei tapahdu mitään.

Ei räjähdystä oven avauduttua riittävästi. Ei kranaatteja, jotka seilaisivat teatraalisesti ilman halki. Ja ennenkaikkea en konetuliaseiden rätinää.

“Näkykö mitään?” Kim kysyy kuiskaten.

“Ei –” Jeff aloittaa.

Sitten ensin vaimeaa, mutta koko ajan yltyvää huminaa alkaa kuulua oven takaa.

“Niitä saatanan lentolaitteita!” Jeff päättää lauseensa.

Lentolaitteita, jotka myrkyttävät.

Eikä Amos ehtinyt kertoa, missä vastamyrkkyä on – jos yleensä on. Half-Life 2:n manhakit sentään vain silpovat ja viiltävät. Nämä ovat paljon pahempi variaatio asiasta, Kim pohtii nähdessään, miten koneet tulevat esiin asevaraston hyllyrivien kätköistä sekalaisena muodostelmana.

Toivoa sopii, että koneet eivät osaa väistellä kovin hyvin. Kaikki kolme nostavat rynnäkkökiväärinsä olkapäälle, ja alkavat ampua tähdättyjä lyhyitä sarjoja. Pienois-helikoptereita on paljon, ainakin kymmenen, ja vaikka varsinaisesta väistelystä ei voi ehkä puhua, niiden liikkeet ovat silti inhottavan sattumanvaraisia. Eikä punaisia valoja. Konstrukti haluaa, että he kuolevat hitaan, myrkyllisen kuoleman.

Kaksi konetta tippuu. Mutta lauma tulee koko ajan lä-

hemmäksi. Pian on ehkä alettava peräytyä.

Kolmas tippuu. Jeff:in ase jumiutuu.

Koneet ovat ehkä kymmenen metrin päässä. Neljäs alhaalla. Orzo osuu yllättävän hyvin. Viides.

Nyt on kai pakko lähteä perääntymään. Jeff saa tukoksen selvitettyä. Mutta mielellään pitäisi saada vielä pari alas, ennen kuin lähtee liikkeelle. Liikkuminen huonontaa aina osumatarkkuutta. Kuudes. Seitsemäs.

Kolme viimeistä konetta ovat nyt ulkona asevarastosta. He ovat suorakaiteen muotoisessa aulatilassa. Toinen uloskäynti vie porraskäytävään, mutta sen ovi on liian kömpelö avata, että sinne ehtisi ennen kuin koneet viiltävät. Täytyy vain perääntyä sinne päin, ja ampua koko ajan.

Kim:in lipas tyhjenee. Jeff pudottaa kahdeksannen. Orzo kompastuu peruuttaessaan, ja kaksi viimeistä hyökkää välittömästi häntä kohti.

Jeff:illä ei voi olla enää paljon kolmestakymmenestä jäljellä, ja Orzo ei ehkä ehdi itse ampua. Turhautuneen raivon vallassa Kim läimäisee tuoreen lippaan sisään, ja vetää viirityskahvasta.

Kolme kivääriä laulavat kuin yhteisestä sopimuksesta. Jeff:in vain parin luodin verran, ennen kuin ase vaikenee. Mutta mikä tärkeintä, kaksi viimeistä helikopteria tippuvat elottomina.

Mielenkiintoisempi kysymys on, ehtivätkö ne viiltää Orzoa. Tämä makaa epämurkavan näköisessä asennossa, piippu kohti kattoa suunnattuna.

“Ne äpäivät viilsivät hattua,” Orzo kiroaa.

Ilmeisesti hän pääsi säikähdyksellä.

“Siinä on myrkkyä. Paras että pudotat sen päästäsi siihen koskematta,” Jeff toteaa tyylysti.

Orzon pettymyksen voi aistia. Mutta pakon sanelemana hän nousee, ja tekee työtä käskettyä. Ilman hattua hän näyttää ehkä hieman tavanomaisemmalta, hieman vähemmän hullulta, mutta piraatilta hän silti näyttää edelleen.

Seuraavaksi: asevarasto.

Nyt heillä on mukanaan mm. kranaatinheittäjiä ja neli-piippuinen Flash-raketinlaukaisin, jota Kim kantaa. Perinteiset rynnäkkökiväärit on vaihdettu moderneihin XM8-kivääreihin, joissa on kiikaritähtäin ja sadan panoksen kaksoisrumpulipas. He laskeutuvat pääportaikkoa pitkin; ovat laskeutuneet jo hyvin monen kerroksen läpi.

“Tuntuuko teistäkin lämpimämmältä?” Orzo kysyy.

Ehkä, Kim toteaa mielessään. Tosin maankuoren uumeeniin ja Jörmungandr:in luokse on taatusti vielä pitkä matka. Ilmeisesti maan alla on, paitsi käytäviä, myös kokonaisia muita Throne Group:in tutkimuslaitoksia. Ja kartan sijasta heillä on vain Kim:in satunnaiset näyt Konstruktin mädänneestä silmästä oppaanaan!

Paremminkin todellakin voisi olla.

Mutta ainakin he ovat hengissä ja suhteellisen vahingoittumattomia, kukaan ei ole kuolemassa myrkytyksestä tai fuusiokennon loppumisesta, ja he ovat raskaasti aseistettuja. Oohrah tälle.

Mikä vitun oohrah?

Ei mitään sotilasmachoilua, Kim muistuttaa itseään. Antaa aseiden puhua puolestaan. Tai machoillaan sitten, jos ollaan vielä elossa, kun Konstrukti on enää vain joukko tyhjyyteen hiipuvia bittejä, jos sitäkään!

Lopulta portaikko päättyy.

Edessä on leveä ja jyhkeä metalliovi, jollaisen voisi kuvitella johtavan armeijan ydinsodan varalta rakennutta-

maan suojabunkkeriin.

Kaikki tähän asti oli vasta alkusoittoa. Todellinen sota alkaa tästä.

VIIDES KIRJA - TYHJYYS

ZEN – Valaistuminen

Aika: määrittelemätön.

Paikka: määrittelemätön.

Harmillisesti uni vaihtuu taas, ennen kuin se ehtii loppuunsa. Tai eihän unesta oikeastaan voi puhua. Olen kai jonkinlaisessa limbossa, olemattomuudessa.

Tosin jos nyt mietitään, että kyse on pelikäsikirjoituksesta, oli unessani jo lyöty lukkoon hyvin paljon. Aivan liian paljon, oikeastaan. Peli ei ole lineaarinen media! On inhottavaa, jos peli on pelkkä lineaarinen käytävä kohti seuraavaa ennalta määrättyä cutscenejä, ja cutsceneissä on aivan eri säännöt kuin niiden välillä.

Mutta saako Kim kohdattua ja tuhottua Konstruktin, Norman Throne:n ja Jörmungandr:in? Haluaisin toivoa, että saa. Selviävätkö Jeff ja Orzo? Tässä on tietysti erinäisiä lähestymistapoja. Voidaan lähteä Cave Story-linjalle, ja kytkeä esim. Jeff:in kohtalo siihen, kerääkö pelaaja riittävät juonikupongit matkan varrelta. Ne voivat olla hyvinkin epäloogisia. Spoilerivaaran vuoksi jätetään tarkempi analyysi Cave Story:sta toiseen kertaan.

Mutta nyt havaitsen olevani työpaikkani katolla. On tuhkaduttavan lämmin ja kostea kesäilta, ja kaukana jyrähtelee ukkonen. Olen siis selviytynyt Vision3D:n valmistumiseen asti (tai onko se nyt koskaan lopullisesti valmis?) ja päättänyt esilletuoda kostoni hieman voimakkaammalla tavalla.

Minulla on käsissäni paksu kaapeli, joka minun on tarkoitus kiinnittää ukkosenjohdattimeen. Kaapelin toinen pää onkin jo kytketty pääreitittimeen serverihuoneessa. (Se, miten onnistuin tämän harhautuksen suorittamaan, jätettäköön kunkin omatoimisesti pohdittavaksi.)

Joka tapauksessa, kun salama iskee, toivottavasti jo tänä iltana ja tähän ukkosenjohdattimeen, noin 30000 ampeerin virran pitäisi kulkeutua toimiston lähiverkkoon. Kyseessä on siis etherkiller hieman suuremmalla mittakaavalla. Kuji-kirin vaihe kahdeksan: luonnonvoimien hallinta!

Katto on aikaisemmin päivällä tulleesta sadekuurosta liukas, joten operaatio ei ole aivan vaaraton. Lisäksi riskinä on putoamisen lisäksi se, että minut nähdään. Toiminnalleni ei juurikaan ole ei-rikollista selitystä.

Saan kaapelin kiinnitettyä. So far, so good, so what.

Mieleeni tulee myös toinen Megadeth:in kappale. My Last Words. Se on pahaenteinen tarina venäläisen ruletin pelaamisesta, jota Valtteri pelasi usein Rock Band:in expert-rummuilla, ja kai ehkä yhdesti tai kahdesti pääsi sen jopa läpi.

Mutta mitkä olisivat viimeiset sanani, jos minun pitäisi lausua ne juuri nyt?

En tiedä.

En kyllä todellakaan tiedä.

Mutta joka tapauksessa nyt pitäisi vielä päästä alas katoilta ehjin nahoin. Se onnistuu kyllä paloportaiden avulla, kunhan ensin pääsen sinne asti...

Yllättäen horjahdan, ja liu'un alas kaltevaa kattoa. Vaikka kulma ei ole suuri, kasvaa vauhtini kuitenkin koko ajan. Sitten sinkoudun reunan yli.

Viime hetkellä – ennen kuin reuna on saavuttamattoman kaukana – saan siitä otteen. Se kolhii kättäni ilkeästi, ja olkapääni menee taatusti sijoiltaan. Onneksi se ei ole se olkapää, joka jo kerran meni sijoiltaan.

Tosin se taas tarkoittaa, että se on vasen olkapää, eli roikun heikommalla kädelläni tyhjän päällä.

Saan otteen toisella kädellä. Nyt tosin pitäisi vielä jak-saa vivuta itsensä takaisin katolle. Jalat eivät saa oikein mistään otetta.

Ukkonen kumahtelee edelleen kaukaisuudessa.

Äkkiä hahmo kurottautuu minua kohti. Se on sateen kastelema Kim, joka näyttää suoraansanoen pelottavalta haljenneen kallopanssarinsa ja sotkuisen vaalean kuontalonsa kanssa. Hän tarjoaa teräskätensä. En tiedä, mihin tämä tulee johtamaan, mutta tartun käteen, ja hän vetää minut katolle.

Sataa helvetin kovaa. Ukkonen on siirtynyt lähemmäs. Kohta täällä katolla voi olla vaarallista olla.

“I’ve seen things you people wouldn’t believe...” Kim aloittaa.

Sitten seuraa pimeys.

Absoluuttinen pimeys.

Nyt muistan, missä oikeasti olen.

Tai missä siis viimeksi olin. Etherkillerin humalatilassa suoritetun testailun yhteydessä onnistuin johtamaan verkkojännitettä kehoni läpi.

230 V voi tappaa.

Hetken päästä pimeydestä muodostuu kirjaimia ja numeroita. Valkoista tekstiä mustalla pohjalla. Kirjaimet ovat niin isoja, että monitorin varjomaskin ja punaiset, vihreät ja siniset osapikselit voi erottaa.

Kirjamilla kestää jonkin aikaa järjestyä. Lopulta niistä syntyy ilmiselvä konekielimonitorin rekisteridumppi ja komentoprompti.

*Process halted Thursday, January 6, 2011 17:45:10.804
Page fault Processor Exception*

Registers:

CS = 0008 DS = 0010 ES = 0010

FS = 0010 GS = 0010 SS = 0010

EAX = 00000001 EBX = 000006EC ECX = 00000085

EDX = 8243377D ESI = 8233F230 EDI = 8233F230

```
EBP = D0CD6F14 ESP = D0CD7EE4
EIP = CED4EE68 FLAGS = 00010202
CED4EE68 8B03 MOV EAX, [EBX] = ?
Access Location: 0x000006EC
```

>

Tämä on juuri se juttu, josta näin varoittavan unen. Re-
kisterit ovat kehoni ja pysähtyneen sydämeni viimeinen
tunnettu tila. Minulla on täydellinen tietoisuus kaikesta,
mikä johti tähän, mutten ehkä kykene muuttamaan sitä
enää.

Kuolema on debuggeri, josta ei pääse jatkamaan!

Olen siis kuollut. Tämä voi olla helvettini.

Helvetti kerrassaan. Ei sen näin pitänyt mennä. Mutta
katsotaan sitten, miltä post-mortem dump näyttää. D (di-
sassemble) -käselyn pitäisi näyttää koodista enemmän.

> D

```
CED4EE68 8B03 MOV EAX, [EBX]
CED4EE6A A3000337CD MOV 000006EC, EAX
CED4EE6F E95EFFFFFF JMP CED4EDD2
CED4EE74 53 PUSH EBX
```

Ei kovin valaisevaa. Ehkä olisi kannattanut harrastaa
assembly-ohjelmointia pitempään ja enemmän! Kyllä. Nyt
nimittäin tiedän, mistä ehkä on kyse, ja minua alkaa vitut-
taa entistä enemmän. Voi olla, että jos saan löydettyä
muistivedoksesta kaatumiseen johtaneen datan, ja korjat-
tua sen, pystyn kuin pystynkin palaamaan elävien kirjoi-
hin. Jälleen kuten Paul Atreides, joka neutraloi hiekkama-
don myrkyn kehostaan. Mutta riittämättömillä konekieli-
taidoillani en kykene siihen, vaan kuolen. Game over.

Tämä on sietämätöntä. Sen kyllä voisi hyväksyä, jos
kuolisi suoraan, mutta tällainen tilaisuuden roikottaminen
nenän edessä on liian julmaa.

Fuck. Fuck. Fuck.

Mutta kokeillaanpa. Jos se vain lähtisi siitä käyntiin. G niinkuin Go.

> G

*Process halted. Page fault Processor Exception
Access Location: 0x000006EC*

Sama virhe uudestaan. Eipä se siitä lähde, jos sille ei tee jotain. Nyt täytyy yrittää pysyä rauhallisena. Varmasti tässä on joku aikaraja, jonka kuluttua umpeen elimistöni vauriot ovat peruuttamattomat, mutta sitä ei suoraan näytetä minulle. Huonoa pelisuunnittelua. Haistakaa paska.

Mietitäänpä. 0x000006EC on siis laiton osoite, jota ei saa lukea. Sen lukeminen pitäisi kiertää. Eri asiahan on, pystyykö edes mitään koodia tai dataa muuttamaan. Unesanihan oli juuri niin, ettei pystynyt.

Mutta jos ei yritä, ei ainakaan voita.

On kyllä kamalaa aiheuttaa mahdollista undefined behaviouria omassa kehossaan. UDB:tähän siis on jo tapahtunut, koska tulee page fault, mutta entä jos nyt siis muuttaa jotain, ja saa itsensä muka kuntoon, mutta tulee korruptoineeksi jotain tärkeää, ja kuolee myöhemmin uudestaan. Hyi!

No kuollaan sitten uudestaan myöhemmin, jos tarve on.
Yeah fuck me, I'm in hell!

Kierretään tämä yksinkertaisimmalla mahdollisella tavalla. Jos menee metsään entistä pahemmin, niin sitten menee. Mikä olikaan komento, jolla kirjoitetaan yli koodia muistissa? Sen täytyy olla A niinkuin Assemble.

> A CED4EE68 JMP CED4EDD2

Suora hyppykäsky sinne, minne oltiin menossa, ja jätetään vain laittoman muistiosoitteen lukeminen välistä pois. Sitten sormet ristiin. Toivotaan, että jotain hyvää kar-

maa on vielä jäljellä.

> *G*

Jotain tapahtuu. Ainakaan ei tule suoraa page faulttia.

Tuntuu niinkuin lattia aukeaisi alta, ja putoaisin syvemmälle. Ehkä tämä oli tehtävä, joka minun tarvitsi ratkaista, jotta pääsen limbosta varsinaisesti rajan taakse.

Kuvottavaa tämä kieppuva putoaminen.

Loppuisi edes.

Rajan takana ei ole ceiling cat:ia. Eikä basement cat:ia. Sensijaan siellä on lattia, työaseman etulevy, jota tarkastelen huomattavan läheltä, ja aivan helvetin paha olo. Päässä kieppuu ja koko ruumista jomottaa.

Hetkinen...?

Onko tämä todellisuus?

Olenko elossa?

Jos on, nyt ei saa tehdä typeryyksiä toista kertaa. Työasema voi olla vielä sähköistetty. Täytyy olettaa että se on.

Ryömin ensin varovaisesti taaksepäin, ja sitten vääntäydyn pystyyn siten, että se varmasti onnistuu työasemaan koskematta, vaikka horjahtaisinkin. Piru kun kieppuu. Törmään ainakin kahteen eri seinään.

Prioriteetti yksi: virran katkaiseminen pääkytkimestä.

On muuten helvetin hyvä tuuri, ettei kokeiluistani syttynyt tulipaloo.

Kurkotetaan ylös.

Lennetään perseelleen.

Ylös siis uudestaan. Tällä kertaa onnistuu. Kuuluu jyhkeä naksahdus, kun sähkötkä katkeavat asunnosta.

Prioriteetti kaksi: etherkillerin irrottaminen ja heittäminen roskiin. Sekin onnistuu. Nyt kai sähkötkä on turvallista laittaa päälle. Tai oikeastaan, irroitetaan nyt vielä työasema varalta. Siinähan on voinut muodostua uusia, mielenkiintoisia kulkureittejä vaiheen ja maan välille. Nyt kannattaa pelata varman päälle, jos kerran pääsi konekielimo-

nitorin avulla rajan takaa pois!

Nyt se on irti, ja siirretty seinän viereen häpeämään. Se täytyy myöhemmin ottaa tutkimuksiin, mistä voi tietysti olla tuloksena, että se yksinkertaisesti on entinen.

Nyt pitäisi olla turvallista laittaa sähköt päälle.

Tulkoon valo.

Selvä. Mitä nyt? Mietin, pitäisikö minun soittaa ambulanssi varmuuden vuoksi. Se voisi olla viisasta, mutta toisaalta ei olisi mukava alkaa valehtelemaan ja selittämään jotain (olenhan jo aikoja sitten päättänyt, että kaikenlainen teeskentely on perseestä), tai vaihtoehtoisesti myöskään kertoa totuutta: että sekoili humalassa johdon kanssa, jolla lähiverkkolaitteisiin syötetään verkkojännitettä. Siitähän saattaisi tulla jopa poliisijuttu.

Ei siis soiteta.

Sen sijaan päätän keittää täyden pannun infernaalisen vahvaa kahvia. Täytyy vain pysyä hereillä niin kauan, että humala (ja krapula) menevät ohi, ja voi varmistua, ettei mitään rytmihäiriöitä tai muita ikäviä ilmiöitä tule.

Näin tehtäköön.

Kokonaisuus huomioiden, selvisin melko lailla säikähdyksellä. Mitähän oikeasti tapahtui? Joskus ehkä voi olla tyylikästä kertoa (siis tietysti maistissa, eihän tällaista selvin päin voi tilittää), että petkutti kuolemaa muuttamalla x86-rekistereitä ja x86-assemblykoodia oman elimistönsä sisältä. Olikohan se siis sydämen firmware-koodia, vai ydinjatkeen, vai minkä?

En ehkä haluakaan miettiä tarkemmin.

Mutta.

Nyt palautan mieleeni Vision3D:n ja irtisanoutumiseni. Jotain tekemistähän elämässä pitäisi olla. Jos en tee jumalallista seuraavan sukupolven AAA-3D-moottoria kostoksi korporaatiotyperyydelle, mitä minä teen?

Vastaus tulee lopulta hyvinkin itsestäänselvästi.

Valtteriin sanat kaikuvat mielessäni.

“Kun laski jokaisen kellojakson, ja oli mahtavaa, kun sai ruudun skrollaamaan kahdeksaan suuntaan, ja siihen pääl-

le vielä yli kahdeksan spriteä.”

On täysin selvää, mitä minun on tehtävä.

Jörmungandr Engine:n idea oli oikea, mutta alusta väärä. Kehää kiertävä arkkitehtuuri täytyy toteuttaa koneiden kuninkaalla, eli Commodore 64:llä.

Siinä ajoympäristö on hyvin tarkasti määritelty:

- *0,985 MHz (PAL) 6510-keskusyksikkö.*
- *VIC-II -videopiiri, jossa 16 väriä ja 320 x 200 resoluutio.*
- *3-kanavainen SID-äänipiiri.*
- *64 kilotavua RAM-muistia.*
- *Kasetti- tai levyasema.*

On myös täysin selvää, mikä on ensimmäinen peli, mikä sillä tullaan toteuttamaan. Kim, Jeff, Orzo ja kaverit tulevat heräämään henkiin C64:n ruudulla, jos vain spritejä riittää. Mikä on sen parempaa, kun pelin lopputaistelussa ruudulle kohoaa usean ruudun korkuinen Jörmungandr? Siinä pitää jo olla melko tarkkana, toteuttaako sen spriteillä vai merkkigrafiikalla. Ja jos peliin haluaa lisää pituutta, voi esim. ilmetä, että Konstrukti onkin turvassa Throne Group:in avaruusasemalla, jonne taas pitää lentää Area51:stä varastetulla lentokoneella tai UFO:lla! Kasetti-asemaa tosin en ehkä tule tukemaan, jotta pelissä voi vapaasti liikkua edestakaisin.

Arttipipeline tulee olemaan huomattavan yksinkertainen. Tarvitaan vain Shoot'em-Up Construction Kit-tyyliset sprite- merkkigrafiikka- ja karttaeditorit. Niin, ja musiikki / ääniefektieditori. Ei mitään collada-importtia tai tappelua rigattujen hahmojen negatiivisten luu-skaalauksien kanssa, joita artistit aina onnistuvat ujuttamaan sisään, vaikka heitä kuinka yrittäisi varoittaa.

Olen varma, että tästä tulee jotain aivan jumalallista.

Tästä tulee tosiohjelmointia.

Lopuksi näyte siitä, mitä on tulossa:

```

SEI                ;Disable interrupts
LDA #$35          ;Disable Kernal & Basic ROMs
STA $01
LDA #$7F          ;Disable timer interrupt
STA $DC0D
LDA #$01          ;Enable raster interrupt
STA $D01A
LDA $DC0D         ;Ack possible pending timer interrupt
LDA #<IRQ1        ;Set IRQ routine address low and high
STA $FFFE         ;bytes
LDA #>IRQ1
STA $FFFF
LDA $D011         ;Set raster IRQ position, high bit
AND #$7F
STA $D011
LDA #IRQPOS1     ;Set raster IRQ positions, low bits
STA $D012
CLI              ;Enable interrupts
JMP MAINLOOP     ;Jump to engine main loop

```