



## **Nato Commando,**

retrohenkinen "shoot'em-up" Flash-peli  
ääniohjelmoinnin ja XML:n hyödyntämisen kannalta.

Tampereen ammattikorkeakoulu,  
taiteen ja viestinnän osasto  
Viestinnän koulutusohjelman tutkintotyö  
Vuorovaikutteisuuden suunnittelu  
Kevät 2005  
Aki Flink

## OPINNÄYTETIIVISTELMÄ

Osasto Viestintä	Erikoistumisala Vuorovaikutteisuuden suunnittelu
Tekijä Aki Flink	
Työn nimi Nato Commando, retrohenkinen "shoot'em-up" Flash-peli ääniohjelmoinnin ja XML:n hyödyntämisen kannalta.	
Lopputyön laji Mediateko	
Työn valmistumisaika 18.04.2005	Sivumäärä 33
Tiivistelmä	
<p>Mediateko koostuu Trash Videon tulevan "Nato Commando" -videon promootiota varten rakennetusta Flash-pelistä, sekä kirjallisesta osasta, joka kertoo XML-tiedostoja tietovarastona käyttävän Flash-pelin ohjelmoimisesta. Näkökulmana kirjallisessa osassa on äänien osalta tehosteäännet, musiikki ja ääniohjelmointi, ja pelimoottorin puolelta XML:n hyödyntäminen pelimoottorissa.</p> <p>Nato Commando sai alkunsa Multimediaalmaisuuksien -kurssin (V3166) kurssityönä, jossa saatiin aikaiseksi osan lopullisista ominaisuuksista sisältävä demo. Kurssin jälkeen työtä jatkettiin erillisenä mediatekoprojektina kahden alkuperäisen projektiryhmän jäsenen kesken.</p> <p>Pelimoottorissa päädyttiin käyttämään 3-kerroksista rakennetta kolmiulotteisuusvaikutelman lisäämiseksi. Tilegrafiikan käyttöön päädyttiin osaksi retrotunnelman aikaansaamiseksi ja osaksi verkkomedian kokorajoitteiden vuoksi .</p> <p>Nato Commando sisältää musiikkia alkuvalikoiden ja highscore -listan aikana, mutta ei pelin aikana, johtuen Flashin ääniominaisuuksien rajoitteista. Pelin aikana soitetaan ainoastaan panoroituja, ohjelmallisesti äänenvoimakkuuskompensoituja pistetehosteita. Tarve äänenvoimakkuuden kompensointiin syntyy Macromedian Flashiin valitsemasta tavasta toteuttaa äänien panorointi muuttamalla vain stereokanavien välistä äänenvoimakkuutta. Tämä toteutusmalli aiheuttaa sen, että keskelle panoroitu ääni soi kaksinkertaisella äänenvoimakkuudella verrattuna jompaankumpaan laitaan panoroituun ääneen. Tämä korjataan pelissä ohjelmallisesti laskemalla jokaisen tehosteen soiton yhteydessä sille tarvittava äänenvoimakkuus.</p> <p>Mediateon tuloksena syntyi kolmea eri XML-tiedostotyyppiä tietovarastona käytävä retrohenkinen "shoot'em up" (= ammuskelu) -peli. Ensimmäinen tiedostotyyppi on highscore -listan käyttämä pistetaulukko. Toinen tyyppi on peliruudun kuvaava tiedosto, joka kertoo pelimoottorille mikä tile kuhunkin kuvaruudun osaan ladataan, montako vihollista ruudussa on ja mitä vihollistyyppiä ne ovat. Kolmas tiedostotyyppi pitää sisällään pelimaailman rakenteen ja kertoo pelimoottorille mikä peliruutu ladataan pelaajan ylittäessä jonkin kuvaruudun reunan.</p> <p>XML-tietovarastoista johtuen pelimaailman rakenteen ja peliruutujen sisällön muokkaamiseen riittää pelkkä tekstieditori. Pelimoottoriin kovakoodattuihin ominaisuuksiin kuuluvat vihollisten käyttäytymismallit, grafiikat, musiikki ja tehosteäännet. Yksittäisten peliruutujen XML-tiedostojen muokkausta varten projektin aikana luotiin yksinkertainen kenttäeditori, mutta pelimaailman kuvaava XML-tiedosto pitää edelleen kasata käsin.</p>	
Aineisto Kirjallisuus ja verkkolähteet	
Asiasanat Nato Commando, Flash-peli, Ammuskelupeli, Räiskintäpeli, Shoot'em Up, Trash Video	
Säilytyspaikka TAMK / Taide ja viestintä, Finlayson	
Muita tietoja	

THESIS

SUMMARY

Department Media production	Area of specialisation Interactivity design
Author Aki Flink	
Title Nato Commando, Audio programming and utilizing XML in a retro-themed "Shoot'em Up" Flash-game.	
Sort of Final Thesis (Written / Project / Portfolio) Project	
Date 18.04.2005	Number of pages 33
<p>Summary:</p> <p>The project consists of a promotional Flash-game for Trash Video's forthcoming "Nato Commando" -video and this written work which describes the design and programming of a Flash-game that uses XML-files as it's data storage. The text covers the aural side of the game engine thru the game audio programming, the process of writing music and using point sound effects. Text also deals with the usage of XML-files in the Nato Commando game engine.</p> <p>Nato Commando project started as a group work on a course of Multimedia design (V3166). By the end of the course the game had turned into somewhat working technology demo version without any really playable content. After the course two members of the original work group continued project as a separate project.</p> <p>3-layer structure was decided to be used in game's drawing engine to enhance the game's three-dimensional expression. The choice of graphical style was tile graphics partly to support the retro theme and partly because it just works great with the size restrictions of the web media.</p> <p>Nato Commando includes music in the menus and high score –list, but not during the game play due the restrictions of Flash. While playing, the game engine utilizes only panned and programically volume compensated sound effects. The need for volume compensations appears due the way sound panning is made in the core of Macromedia Flash. This technique consists of just changing the relative sound volume of left and right stereo channel. While this is a really common technique to use the downside of this implementation is that sound that plays in the middle position is twice as loud as sound panned to either side. In Nato Commando this behavior is corrected by calculating compensated volume for each sound at moment it starts to play.</p> <p>The result of the project is a retro themed "shoot'em Up" Flash-game that utilizes three different XML-file types. First type is used for storing the high score table. Second type describes the game screen by telling the game engine which tiles to use in each slot on screen and how many and which type of enemies the screen has. Third file type describes the order the game screens are placed in the game world or to put it simple which screen file to load when the player walks to some edge of the screen.</p> <p>As the game utilizes XML-files as data storage, the playable content of the game can be modified with any text editor. Some elements of game are however hard-coded in the game engine. These include the behaviour of the enemies, graphics and audio files. To speed up the content creation a simple editor was written to help and speed up the creation of the game screens. While each of the screen files can be edited with the editor the game world description still has to be made by hand.</p>	
Material (e.g. audio / video tape, photographs, slides, paintings, statues...) Literature and web sources	
Key words Nato Commando, Flash-game, Shooting game, Shoot'em Up, Trash Video	
Filing Tampere Polytechnic, Art and Media, Finlayson	
Other information	

## SISÄLLYS

<b>JOHDANTO</b> .....	3
<b>1 Miksi vanhat pelit, laitteet ja retropelit kiinnostavat?</b> ....	4
1.1 Isot lapset leikkivät vanhoja leikkejä.....	4
1.2 Uudet laitteistoalustat hakevat muotojaan.....	4
1.3 Hektinen elämänrytmi rajoittaa ajankäyttöä.....	5
1.4 Helppoa rahastusta?.....	5
1.5 Emulaattorien ongelmat.....	6
<b>2 Pelilaitteistojen kehitys</b> .....	7
2.1 TV-peli? Videopeli?.....	7
2.2 Muut varhaiset pelikonsolit.....	9
2.3 Alkuaikojen kotitietokoneet.....	11
2.4 16-bitillä taivaaseen.....	12
2.5 Voittoa ja tappioita.....	15
2.6 Musta hevonen: Personal Computer.....	17
<b>3 Case: Nato Commando</b> .....	19
3.1 Lähtökohdat pelille.....	19
3.2 Oivalluksia ja ohjelmointia 1 – XML.....	19
3.3 Oivalluksia ja ohjelmointia 2 - ääniohjelmointi.....	22
3.4 Äänityöt 1 – lähtökohdat.....	24
3.5 Äänityöt 2 – sävellys.....	25
3.6 Äänityöt 3 – webmedian erikoisvaatimukset.....	26
3.7 Projektin saavutukset.....	27
<b>4 Yhteenveto</b> .....	29
4.1 Yhteenveto.....	29
<b>LÄHTEET</b> .....	32

## JOHDANTO

Mediatekoni kirjallinen osa käsittelee Nato Commando Flash-pelin pelimoottorin äänien tekoa sekä pelin ääniohjelmoinnin ja XML-tiedostojen käytön ongelmia ja niissä tehtyjä ratkaisuja. Tekstissä ei käsitellä tai selitetä XML-standardia sinänsä, vaan oletetaan lukijan jo omaavan peruskäsityksen ko. standardin syntyyn johtaneista syistä ja XML-tiedostojen rakenteesta. Tekstin tarkoituksena ei myöskään ole tehdä lukijasta Macromedia Flash- ja ActionScript-asiantuntijaa, vaan kertoa miten ja miksi Nato Commando:n ohjelmoinnissa päädyttiin kuhunkin ratkaisuun ja pohtia olisiko jotakin voitu toteuttaa toisin. Parhaan hyödyn tekstistä kuitenkin saa, jos omaa jo jonkinlaisen käsityksen Flashista ja ActionScriptistä.

Alustan case-osuutta historiikilla pelikonsoleiden ja varsinkin niiden ääniominaisuuksien evoluutiosta. Yritin välttää pelkkää teknisten tietojen listaamista ja suunnata käsittelyä enemmän äänien käytännönhyödyntämismahdollisuuksiin kuin teknisten hienouksien esittelemiseen. Joidenkin laitteiden kohdalla varsinaiset kehitysaskleet ovat kuitenkin tapahtuneet nimenomaan tekniikassa, jolloin ominaisuusesittelyiltä ei voinut välttyä.

Emulaattorilla tarkoitetaan (tekstissä) ohjelmistoa, joka kykenee suorittamaan jossain laitteessa eri laitteelle ohjelmoitua ohjelmakoodia.

## **1.0 Miksi vanhat pelit, laitteet ja retropelit kiinnostavat?**

### **1.1 Isot lapset leikkivät vanhoja leikkejä**

Yksi syy tänä päivänä näkyvään retroinnostukseen on monien ihmisten halu verestää muistoja omasta lapsuudestaan tai nuoruudestaan palaamalla niihin vanhoihin yksinkertaisiin peleihin, joita joskus 10-20 vuotta sitten kotona on mahdollisesti pelattu. Tämä näkyy mm. erilaisten emulaattorien suuressa suosiossa. Vanhojen laitteiden ollessa kyseessä huomioon täytyy tietenkin ottaa myös alkuperäisten laitteiden rapautuva kunto, laitteiden käytön kertakaikkinen monimutkaisuus ja ohjelmien huomattavasti helpompi saatavuus emulaattoreille.

Emulaattoreissa ja vanhoissa pelilaitteissa voi tietysti löytyä myös uutuuden viehätystä emuloitavan laitteen iästä riippumatta, varsinkin käyttäjän päästessä käyttämään jotakin "uutta" eksoottista laitetta tai saadessa vihdoin sen pelikoneen, jota vanhemmat eivät koskaan suostuneet ostamaan ja joka luonnollisesti löytyi "kaikilta kavereilta".

### **1.2 Uudet laitteistoalustat hakevat muotojaan**

Toinen vanhojen pelien kierrättämisen syy löytyy vielä muotoaan hakevien kannettavien pelikonsolien ja kännyköiden puolelta, koska näiden nuorien laitealustojen ympärille ei vielä ole muodostunut omia peligenrejakojaan saati "peruspeli" -tyyppejä. Edes Nintendon kypsään ikään ehtinyt Gameboy -laitesarja ei ole vielä kyennyt vakiinnuttamaan mitään tiettyä peligenreä omaksi tyyppipelikseen.

Koska kannettavat pelikonsolit eivät ainakaan lähitulevaisuudessa, jos koskaan tule saavuttamaan kotipelikonsolien ja pc-maailman kehitystahtia ja tehoja, on luonnollista ja pelikehittäjille riskittömämpää etsiä peleihin valmiita pelityyppejä ja -konsepteja vanhem-

mista konsoli- ja tietokone-peleistä.

### **1.3 Hektinen elämänrytmi rajoittaa ajankäyttöä**

Myös pelien pelisyklin pituudella on vaikutusta niiden suosioon, koska aikaa pelaamiseen on käytettävissä usein vain satunnaisesti esimerkiksi työmatkalla, välitunneilla tai kahvitauolla. Nykyinen elämänrytmi suosii näin pelejä, joissa pelisykli on lyhyt, oppimiskynnys on matalalla eikä peli vaadi pitkäaikaista keskittymistä kerralla. Vanhemmat pelit sopivat varsin usein näihin vaatimuksiin, koska niiden muotoja ja pelisyklinkestoa rajoittivat paitsi vanhojen laitealustojen tekniset ominaisuudet myös varhaiset ohjelmointitekniikat.

### **1.4 Helppoa rahastusta?**

Varsinkin pienet pelintekijät näkevät vanhojen pelien kierrättämisessä riskittömämmän tavan pelituotannolle, koska uusioversiossa selvittää alusta asti itse rakennettuja pelikonsepteja pienemmällä suunnitteluajalla ja -kustannuksilla. Suosikkipelien konseptien käyttö ei toki automaattisesti takaa hittipeliä, sillä aina ainakin osa käyttäjistä käyttää ennemmin emulaattoria ja vanhoja pelejä ilman erillisiä kustannuksia, kuin maksavat samasta pelistä uudelleen viritettynä.

Pelin myyntiin vaikuttaa toki oleellisesti jos vanhaan peliin on luotu jotain uutta lisäarvoa tai siinä korjataan jokin alkuperäisen pelin suosiota rajoittanut vika. Uudistukset ovat kumminkin aina kaksiteräinen miekka, koska huonosti suunniteltu tai toteutettu uudistus romuttaa pelin myynnin pelin myös pääasialliselle kohderyhmälle.

## 1.5 Emulaattorien ongelmat

Emulaattorien käytössä nousee useasti esille kysymys siitä loukkaako ohjelmien ajaminen emulaattorissa tekijänoikeutta, koska ohjelmaa ajetaan eri laitteessa ja useimmiten suoraan tiedostosta alkuperäisen jakelumedian sijasta.

Monien vanhojen pelien kohdalla vanhojen pelien julkaisijat ovat kuitenkin jo aikaa sitten kärsineet konkurssin, sulautuneet suurempiin yrityksiin tai pelien oikeudenomistajat (osassa tapauksissa yksittäisiä ihmisiä) ovat muuten vain välinpitämättömiä markkinoilta poistuneiden laitteiden ohjelmien kohtalosta. Tämä ei luonnollisesti poista ohjelmien tekijänoikeussuojaa, mutta niiden ilmainen jakaminen ei enää aiheuta julkaisijalle taloudellisia menetyksiä, koska yhtään ainoaa alkuperäismedialla olevaa kappaletta näistä peleistä on hyvin harvoin löydettävissä saati myynnissä.

Hankalasta saatavuudesta johtuen voimme tulevaisuudessa päästä näkemään suosikkipeleistä julkaistuja keräilyversiota jollekin jo vuosia aiemmin markkinoilta hävinneelle laitteelle. Tämä johtuu siitä että suurin osa vanhoja laitteita keräilevistä ihmisistä on työikäistä ja pystyy siten maksamaan keräilyesineistä enemmän. Toisaalta keräilijöiden silmissä originaaliversio on varmasti se halutuin lisä kokoelmaan.

## 2.0 Pelilaitteistojen kehitys

### 2.1 TV-peli? Videopeli?

Seuraava kappale on suomennos ja referointi tv-pelin historiasta, joka löytyy osoitteesta: <http://www.pong-story.com/sanders.htm> ja <http://www.pong-story.com/odyssey.htm>, jos pelikoneiden historia yhtään kiinnostaa suosittelen lämpimästi koko pong-story -sivuston



tarjontaan tutustumista.

Pelikonsoleiden isä on yhdysvaltalainen *Ralph Baer*, jonka jo 1951 saamasta ideasta kehittyi monen vaiheen kautta maailman ensimmäinen pelikonsoli (tv-game / videogame). 1951 Baer työskenteli pienessä bronxilaisessa elektroniikkaliikkeessä ja kehitteli mielessään ideaa pelien pelaamisesta tavallisella tv-vastaanottimella kalliiden tietokoneiden sijaan. Erään tv-tuotekehitysprosessin aikana Baer jopa ehdotti työnantajalleen pelimahdollisuuden lisäämistä suoraan tv-vastaanottimeen, mutta tämä ei kuitenkaan herättänyt mitään mielenkiintoa ja asia jäi hautumaan yli kymmeneksi vuodeksi.

Kesällä 1966 Baer, joka tällöin työskenteli osastonjohtajana armeijakalustoa kehittäväällä ja valmistavalla Sanders Associates:illa, palasi jälleen ambitionsa kimppuun ja luonnosteli tv-pelikonseptiaan paperille bussiasemalla odotellessaan. Vielä samana vuonna hän kirjoitti luonnostelmansa puhtaaksi 4-sivuiseksi suunnitelmaksi, piirsi laitteesta piirikaavion ja pyysi yhtä osastonsa teknikkoa valmistamaan piirilevyn. Tämä prototyyppi sysäsi liikkeelle projektin, joka jatkui aluksi epävirallisena ja myöhemmin myös virallisena projektina Sanders Associates:illa aina vuoteen 1968 asti. Sanders Associates ei ilmeisistä imagosyistä kyennyt yhtäkkiä alkaa myydä tai valmistaa kuluttajille suunnattua pelilaitetta, joten Baerin oli etsittävä laitteelle lisenssivalmistaja.

Ensimmäisenä lisenssiä havitteli RCA, mutta sopimus kaatui kumminkin pian RCA:n rahaongelmiin. RCA:n työntekijänä laitteen demosta innostunut *Bill Enders* oli tällä välin siirtynyt Magnavox:in varamarkkinointijohtajaksi ja kävi Sanders Associates:illa testaamassa laitetta. Hän innostui laitteesta entistäkin enemmän ja sai lopulta houkuteltua Magnavox:in johdon pyytämään laitteen esittelyä.

17. elokuuta 1968 Baer ja Sanders Associates:in patenttipäällikkö *Lou Etlinger* kävivät esittelemässä laitteen Magnavox:in pää-

majassa ja firman johtaja *Gerry Martin* ihastui laitteeseen. Pyörät pyörähtivät jälleen liikkeelle ja Magnavox ja Sanders Associates solmivat sopimuksen, jolla laitteen 7. prototyyppi "Brown Box" ja laitteen tekniset dokumentit siirtyivät Magnavox:in haltuun.

Magnavox:in tuotekehitysryhmä sai esituotantomallin valmiiksi 1971 ja laitteen läpäistessä FCC-laboratorion testit, maailman ensimmäisen pelikonsolin tuotanto alkoi Magnavox:in Tennesseeen tehtaassa.

*Magnavox Odyssey* ei ollut taloudellinen menestys, koska sen myyntiä haittasivat suuren 100\$ hintalapun lisäksi huono mainoskampanja, joka antoi kuluttajille virheellisen kuvan että laite toimi vain Magnavox:in valmistamissa tv-vastaanottimissa. Lisäksi Magnavox:in laitteita saivat tuolloin myydä kauppaketjujen sijaan vain sen valtuuttamat kaupat.

Vaikka *Odyssey* oli teknisiltä ominaisuuksiltaan yksinkertaisesti minimiinsä valmistuskustannuksien minimoimiseksi, se sisälsi silti monia myöhempiin konsoleihin siirtyneitä keksintöjä. *Odyssey*:n pelit tulivat erillisillä moduuleilla ja siihen sai kytkettyä vakiopeliohjaimen lisäksi valopistoolin ja lisävarusteena ostettavan valokiväärin ammutapelejä varten. Pelit olivat yksivärisiä (valkoinen mustalla taustalla), mutta tunnelman luomiseksi mukana tuli erivärisiä ja -kuvioisia televisioruutuun staattisesti kiinnittyviä muovikalvoja, jotka muuttivat kuvan esimerkiksi vihreäksi tennispeliä varten.

## 2.2 Muut varhaiset pelikonsolit

Burlingamen lentokentällä järjestetyssä demotilaisuudessa *Odyssey*:tä kävi testaamassa myös *Nolan Bushnell*, joka noin kuu-kautta myöhemmin (27. kesäkuuta 1972) perusti *Atari*:n. Yhdessä palkollisensa *Alan Alcorn*:in kanssa Bushnell suunnitteli ja rakensi kolikkopelin, joka muistutti toiminnaltaan *Odyssey*:n Ping-Pong -peliä

(pöytätennis), mutta sisälsi hienoja lisäominaisuuksia, kuten pallon kimpoilun kentän sivulaidoista, digitaalisen pistelaskurin ja kaikuluo-  
tainmaisen äänitehosteen pallon osuessa johonkin<sup>1</sup>.

*Pong* osoittautui menestyspeliksi ja Magnavox haastoi lopulta Atari:n ja myöhemmin monia muitakin Pong-tyyppisiä pelejä valmistaneita yrityksiä oikeuteen (mm. Bally-Midway ja Nintendo) patenttirikkomuksesta ja voitti nämä oikeudenkäynneissä. Tämän seurauksena kaikki "kloonipelejä" valmistaneet yhtiöt joutuivat lisensoimaan Pong-peleihinsä patentit Sanders Associates:ilta ja Magnavox:ilta<sup>2</sup>. Vuonna 1974 Atarilla oli jo kehitteillä Pong-peli kotikäyttöön<sup>3</sup>, mutta sen valmistuessa vuotta myöhemmin, sille ei tahtonut löytyä markkinoijaa Magnavox:inkin myynnin laskiessa paljolti epäonnistuneen markkinoinnin seurauksena.

Kauppaketju Sears otti kuitenkin lopulta laitteen myyntiin Tele-Games tuotemerkkinsä alle ja tilasi jouluna (1975) Atari:lta 150 000 laitetta. Laite kävi erittäin hyvin kaupaksi ja tästä rohkaistuneena Atari aloitti laitteen myymisen omalla nimellään vuotta myöhemmin<sup>4</sup>.

Kilpailijoihin nähden Atari:n vahvuutena oli jo kolikkopeliä varten suunniteltu integroitu mikropiiri, joka sisältäisi kaikki pelitoiminnot, joten kehittyneemmistä ominaisuuksista huolimatta, laitetta voitiin myydä ilman lisähintaa. Atari:n Pong-pelikoneen myötä videopeliäännet saapuivat myös kotitalouksiin<sup>5</sup>.

Jouluna 1977 Atari julkaisi *Video Computer System*:in (VCS, myöh. *Atari 2600*) yhdeksän pelimoduulin kanssa. Näistä maininnan arvoisia ovat mm. sokkeloseikkailu *Adventure* ja tankkitaistelupeli *Combat*, joiden pelityyppien yhdistelmä case-osuudessa käsiteltävän Flash-pelimme *Nato Commando*:n peligenre pohjimmiltaan on. Atari lisensoi

<sup>1</sup> <http://www.pong-story.com/odyssey.htm> (background)

<sup>2</sup> <http://www.pong-story.com/odyssey.htm> (The very first law suit...)

<sup>3</sup> <http://www.pong-story.com/atpong2.htm> (PONG reaches homes:)

<sup>4</sup> <http://www.pong-story.com/atpong2.htm> (Sears and the 'Tele-Games' systems:)

<sup>5</sup> <http://www.pong-story.com/atpong2.htm> (PONG reaches homes:)

myöhemmin VCS:lle mm. *Namco*:n kolikkopelihitti *Space Invaders*:in, joka vauhditti laiteen myyntiä (yli 25 miljoonaa myytyä laitetta)<sup>6</sup>.

VCS kykeni soittamaan kahta synteettistä ääntä yksinkertaisella äänipiirillään, mutta piirin rakenteesta johtuen se ei pystynyt "vireessä" vaan äänenkorkeudet vaelsivat milloin ylös milloin alaspäin. Tämä rajoitti äänisuunnittelua, koska minkään normaalin musiikin tuottaminen ei piirillä onnistunut. Toisaalta tämä johti täysin uniikkeihin pelimusiikkeihin, koska mitään perinteistä melodiaa laitteella ei pystynyt matkimaan<sup>7</sup>.

Vuonna 1980 videopelimarkkinoille liittyi myös leluvalmistaja *Mattel Intellivision* -pelikonsolillaan, joka kykeni VCS:ää parempaan grafiikkaan ja 3-kanavaiseen äänisynteisiin *General Instruments*:in valmistamalla äänipiirillä (AY-3-8910). VCS:stä poiketen *Intellivision*:in äänipiiri pystyi tuottamaan "vireessä" pysyviä ääniä, joten sen pelimusiikeissa alettiin pian jäljitellä tunnettuja melodioita<sup>8</sup>.

Syksyllä 1985 japanilainen pelikortti- ja leluvalmistaja Nintendo lanseerasi ensimmäisen konsolinsa Nintendo Entertainment System:in (NES) yhdysvaltoihin<sup>9</sup>. Japanissa muutamaa vuotta aiemmin Famicom nimellä julkaistun konsolin äänipiiri (2A03, muokattu MOS6502 prosessorin sisään) kykeni 5-kanavaiseen ääneen: kaksi kanttiaaltoa, yksi saha-aaltoa, yksi kohinaa ja yksi digitoituja ääniä (PWM) soittamaan kykenevä kanava<sup>10</sup>.

Vaikka NES:in äänisynteesi ei esiteltyt yleisellä laitetasolla mitään uutta, se oli kuitenkin suuri hyppäys eteenpäin pelikonsoleissa ja japanilaisena yrityksenä Nintendo toi lisäksi mukanaan itämaisen musiikkiperinteen. NES:in parasta antia olivatkin loistavien pelisuunnitte-

---

<sup>6</sup> Ellis, David 2004. Sivun 4.

<sup>7</sup> <http://www.dullien-inc.com/collins/games/consoles.html> (Atari VCS/2600 and 7800)

<sup>8</sup> <http://www.dullien-inc.com/collins/games/consoles.html> (Intellivision)

<sup>9</sup> <http://www.dullien-inc.com/collins/games/nintendo.html> (Nintendo NES)

<sup>10</sup> <http://nesdev.parodius.com/NESSOUND.txt>

lijoiden ja Nintendon tiukan pelien laatuvalvonnan lisäksi japanilaiset muusikot mm. Nintendon *Shigeru Miyamoto* (Super Mario Bros, 1985) ja *Square Software* *Nobuo Uematsu* (Final Fantasy, 1987), joiden sävellykset loivat aivan uudet laatustandardit ja omat musiikkigenret konsolipeleille<sup>11</sup>.

## 2.2 Alkuaikojen kotitietokoneet

*Commodore International Limited* julkaisi 1983 kaikkiaan 22 miljoonaa kappaletta myyneen *Commodore 64* kotitietokoneensa. Alun perin laitetta kaavailtiin pelikonsoliksi, mutta laitteen kehityksen aikana huomattiin että laiteesta tulisi aivan liian kallis ja että se toisaalta sisälsi kaikki yleiskäyttöiseen tietokoneeseen vaadittavat ominaisuudet (näppäimistöä ja lisälaiteliitäntöjä lukuun ottamatta)<sup>12</sup>.

*Bob (Robert) Yannes* käytti koneessa Commodoren vuonna 1975 omistukseensa hankkiman *MOS Technologies*in valmistamaa ja itse suunnittelemaansa MOS6581-äänipiiriä (SID, Sound Interface Device), joka myöhemmissä C64-malleissa korvattiin hieman erilaisella MOS8580-äänipiirillä<sup>13,14</sup>.

SID on hyvin monipuolinen äänipiiri, joka mahdollistaa kolme samanaikaista ääntä, joissa kaikissa on käytettävissä sekä verhoikäyrät että yli-, ali- tai kaistanpäästösuodatin. Tämän lisäksi käytettävissä on kohinakanava, jota nopeasti moduloimalla voidaan toteuttaa PWM-tyyppinen 4-bittinen digitoidun äänen toisto.

Piirin poikkeuksellisesta käyttöarvosta kertoo mm. sen ympärille rakennettu syntetisaattori *SidStation* (<http://www.sidstation.com>) ja C64:lla sävellettyjen kappaleiden lukematon määrä (<http://www.hvsc.c64.org/>).

<sup>11</sup> <http://www.vgmusic.com/vgpaper.shtml>

<sup>12</sup> <http://www.dullien-inc.com/collins/games/c64.html>

<sup>13</sup> <http://www.dullien-inc.com/collins/games/c64.html>

<sup>14</sup> <http://stud1.tuwien.ac.at/~e9426444/yannes.html>

SID:stä kiinnostuneita suosittelen lukemaan Bob Yannes:in haastattelun: <http://stud1.tuwien.ac.at/~e9426444/yannes.html>.

### 2.3 16-bitillä taivaaseen

Jay Minerin pienessä kalifornialaisessa Amiga Corporationissa suunnittelema ja Commodoren heinäkuussa 1985 julkaisema Commodore Amiga (1000) siirsi kotitietokoneet uuteen uljaaseen 16-bittiseen aikauteen 8Mhz *Motorola MC68000* prosessorillaan.

Ironisesti Commodoren pahimman kilpakumppanin Atarin tuella kehitetty tietokone päättyi Commodoren omistukseen, koska Atari kehitti omaa 16-bittistä tietokonettaan eikä Amigan valmistuessa halunnut sisäistä kilpailua. Atari ei myöskään onnistunut estämään Commodorea hankkimasta Amigan tekniikkaa itselleen oikeudenkäynneistä huolimatta<sup>16</sup>.

Jay Miner käytti Amigassa suosimaansa kolmipiiriratkaisua, jossa pääsuorittimen apuna on kolme erikoistunutta apusuoritinta; Denise (kuva), Agnus (mm. muistinhallinta), Paula (ääni ja levyhallinta). Paula kykenee soittamaan neljä yhtäaikaista digitoitua 8-bittistä monofonista ääntä. Näistä neljästä sisäisestä äänikanavasta summataan kaksi vasempaan ja kaksi oikeaan stereokanavaan<sup>17</sup>.

Atarin oma 16-bittinen tietokone kehitettiin valtavalla kiireellä Amiga Corporationin teknologian mennessä Commodoren omistukseen, joten koneen myyntiin tullessa sen käyttöjärjestelmä ei ollut täysin valmis ja se toimitettiin levykkeellä (Atari 130ST & 260ST), sisäisten ROM-piirien sijaan (Atari 520ST)<sup>18</sup>.

Erikoispiirien sijaan Atari turvautui pelkästään keskusprosessorinsa (sama kuin Amigassa) tehoon, joten sen suorituskyky jäi vaati-

---

<sup>16</sup> <http://www.old-computers.com/museum/computer.asp?st=1&c=28>

<sup>17</sup> [http://en.wikipedia.org/wiki/Original\\_Amiga\\_chipset](http://en.wikipedia.org/wiki/Original_Amiga_chipset)

<sup>18</sup> <http://www.old-computers.com/museum/computer.asp?st=1&c=922>

vimmassa käytössä hieman Amigan vastaavasta. Atari oli kumminkin ensimmäinen tietokone sisäänrakennetulla MIDI-liitännällä, joten se saavutti suosiota varsinkin muusikoiden keskuudessa. Lisäksi Atarin käyttöjärjestelmä kykeni (CP/M 68k) lukemaan ja kirjoittamaan suoraan PC:n DOS-levykeitä, mikä helpotti suuresti tiedostojen siirtoa koneiden välillä<sup>19</sup>. Varsinaiset ääniominaisuudet ST:ssä hoiti auttamattomasti vanhentunut Yamahan YM 2149-synteesipiiri, joka kykeni vain 3-kanavaiseen äänisynteesiin ilman minkäänlaista digitoidun äänen soittomahdollisuutta<sup>20</sup>.

1990 julkaistiin Atari 520STe ja 1024 STe koneet, jotka lisäsivät koneen ääniominaisuuksiin vihdoin kaksi 8-bittistä digiäänikanavaa. STe ei kuitenkaan koskaan oikein saavuttanut pelitekijöiden suosiota johtuen yhteensopivuusongelmista vanhemman ST-mallin kanssa.

Atarin viimeiseksi tietokoneeksi jäi 1992 julkaistu Falcon 030, joka piti sisällään tehokkaamman MC68030 prosessorin lisäksi myös ammattimaiset ääniominaisuudet (kahdeksan 16-bittistä digikanavaa). Käyttäjien onneksi Atari kuitenkin lisensoi tekniikkaansa muille valmistajille, joten vaikka yhtiö katosi markkinoilta, sai muiden valmistajien valmistamia laitteita yhtä<sup>21,22</sup>.

Varhaisemmista pelkästään beeperillä varustetuista AppleII -malleista poiketen *Apple Computer Company*:n 1986 julkaisema *Apple IIgs* (graphic & sound) sisälsi hyvin edistyneet ääniominaisuudet. Apple kaavaili gs:ään alun perin perinteistä nelikanavaista äänigeneraattoria, mutta *Rob Moore*n ehdotuksesta sen sijasta käytettiin *Ensoniq*:in valmistamaa *5503 Digital Oscillator Chip*:iä (*DOC*). *Ensoniq*in perustajan Robert Yannesin *Mirage*-sampleriin suunnittelema piiri kykeni syntetisoimaan kaikkiaan 15 yhtäaikaista ääntä (8-bittiä, stereo)<sup>23,24</sup>.

<sup>19</sup> <http://www.old-computers.com/museum/computer.asp?st=1&c=922>

<sup>20</sup> <http://www.dullien-inc.com/collins/games/midi.html>

<sup>21</sup> [http://www.ntrautanen.fi/marko/sao/esittelyt/clab\\_mk2.htm](http://www.ntrautanen.fi/marko/sao/esittelyt/clab_mk2.htm)

<sup>22</sup> <http://www.kingx.com/kingx/medusa/thes-hades.html>

<sup>23</sup> <http://www.old-computers.com/museum/computer.asp?st=1&c=71>

<sup>24</sup> <http://www.dullien-inc.com/collins/games/apple.html>

Nämä ominaisuudet herättivät niin paljon huomiota, että *The Beatles*:in levymerkki *Apple Records* haastoi Applen oikeuteen syyttäen Apple Computeria aikaisemman sopimuksen rikkomisesta. Tämä aiemmin tehty sopimus kieltää Apple Computeria myymästä fyysisiä äänitteitä, josta vastalahjaksi se sai säilyttää Apple-tuotemerkkinsä<sup>25</sup>. Seuranneesta oikeudenkäynnistä johtuen missään Applen mallissa ei gs:n jälkeen enää ole ollut syntetisaattoriipiirejä vaan äänisynteesi on hoidettu ja hoidetaan edelleen ohjelmallisesti<sup>26</sup>.

Apple Corps:in ja Apple Computerin välisten oikeudenkäyntien taustoista voi lukea tarkempaa selvitystä seuraavasta osoitteesta:  
[http://www.legalzoom.com/articles/article\\_content/article11325.html](http://www.legalzoom.com/articles/article_content/article11325.html)

## 2.4 Voittoja ja tappioita

Konsolien 16-bittinen sukupolvi käynnistyi 1989 japanilaisen *Sega*:n Genesis (Euroopassa MegaDrive) konsolilla, joka onnistui hyvällä ajoituksella viemään jopa 50 % yhdysvaltojen markkinoista. Kun Nintendo vihdoinkin kaksi vuotta myöhemmin vihdoinkin sai oman 16-bittisen *Super Nintendo*-konsolinsa markkinoille, se joutui taistelemaan jokin kunnolla markkinapaikastaan<sup>27</sup>.

Genesiksen ääniominaisuudet olivat selvästi Nintendoa heikommät perustuen 6-kanavaiseen FM-synteesiin, Nintendon kyetessä käyttämään kahdeksaa lähes cd-tasoisista digiäänikanavaa. Johtuen pelimoduulien pienestä muistimäärästä Super Nintendon peleissä käytettiin yleisesti äänitekniikkana Amigalta liikkeelle lähtenyttä MOD-musiikkia, jossa MIDI-nuotituksen tapaisella ohjelmoinnilla ohjataan lyhyiden ääninäytteiden soittoa.

32-bittinen konsolisukupolvi ilmestyi markkinoille vuoden 1995 ai-

<sup>25</sup> [http://www.legalzoom.com/articles/article\\_content/article11325.html](http://www.legalzoom.com/articles/article_content/article11325.html)

<sup>26</sup> <http://www.dullien-inc.com/collins/games/apple.html>

<sup>27</sup> Ellis, David 2004. Sivun 281.



kana Sega Saturn- ja Sony PlayStation-konsolien muodossa. Tämän sukupolven konsolit sisälsivät ensimmäiset 3D-laskentaan erikoistuneet apupiirit ja käyttivät pelimedianaan cd-rom -levyjä.

Molemmat konsolit kykenivät soittamaan cd-levyltä paitsi cd-tasoista pakattua äänimateriaalia, myös pakattua videota, joka moduulimedian aikana ei muistikapasiteetista johtuen olisi tullut kysymykseen. Lisäksi molemmat konsolit käyttivät myös MOD-musiikkia, johtuen sen helpommasta sovittamisesta interaktiiviseen ympäristöön. Sega kykeni soittamaan 32 cd-tasoista ääniraitaa ja lisäksi käyttämään 8 FM-synteesikanavaa, PlayStationin kyetessä "vain" 24 cd-tasoiseen ääniraitaan<sup>28</sup>.

Nintendo julkaisi oman 64-bittisen konsolinsa vuotta myöhemmin ja vaikka konsoli oli joiltain ominaisuuksiltaan Segaa ja Sonya parempien lopullinen kuolinisku oli kumminkin päätös käyttää yhä moduuleita pelimediana. Moduulit olivat nopeudestaan huolimatta huomattavasti kalliimpia valmistaa ja tallennuskapasiteetiltaan yleensä vain noin kuudesosan cd:n tallennuskapasiteetista. Tämä päätös nosti Nintendon pelien hinnat huomattavasti Segan ja Sonyn pelejä kalliimmaksi ja aiheutti pelinkehittäjille vakavia rajoituksia<sup>29</sup>. Suurta mainetta Aasian markkinoilla nauttinut SquareSoft, joka oli kehittänyt Final Fantasy -pelisarjaansa siihen asti pelkästään Nintendon konsoleille, siirtyi kehittämään sarjan uutta osaa Sonyn konsolille petyttyään Nintendo64:n teknisiin rajoituksiin.

128-bittisen konsolisukupolven markkinataistelu alkoi 1999 Segan julkaistessa Dreamcast-konsolinsa. Sony liittyi mukaan vuotta myöhemmin PlayStation2:lla, 2001 Nintendo GameCube:llaan ja uusi haastaja Microsoft X-Box konsolillaan.

Sega ei vuoden varaslähdöstään ja konsolinsa hyvistä ominai-

---

<sup>28</sup> <http://www.dullien-inc.com/collins/games/32bit.html>

<sup>29</sup> Ellis, David 2004. Sivun 278.

suuksista huolimatta kyennyt valtaamaan markkinoita kaikkien keskittäessä huomionsa PlayStation2:n julkaisuun. Toki täytyy muistaa että PlayStation2-konsolilla käyttäjät kykenivät käyttämään koko valtavaa PlayStation pelitarjontaa Segan vasta rakentaessa pelimarkkinoitaan.

Tämän sukupolven laitteet käyttävät kaikki pelimediana DVD-levyä, Nintendon 3-tuumaista 1,5Gt:n erikoislevyä lukuun ottamatta. Ääniominaisuuksiltaan laitteisto jakaantuu seuraavasti: Dreamcast ja GameCube 64 äänikanavaa, PS2 48 äänikanavaa ja X-Box 64 3D-tehostettua ääntä + 192 normaalia ääntä. Kaikki laitteet tukevat myös suosituimpia monikanavaäänijärjestelmiä (Dolby, AC3 ja DTS). Kuten ominaisuuksista voi päätellä laitteet eivät teknisiltä ominaisuuksiltaan eroa enää vanhempien konsolisukupolvien lailla ja loppukädessä suurimman vaikutuksen pelien äänimaailmaan tekee vihdoinkin ääniohjelmointi ja -suunnittelu.

## 2.5 Musta hevonen: Personal Computer

12. elokuuta 1981 sai alkunsa pitkäikäisin ja monikäyttöisin tietokoneformaatti IBM PC. Alun perin puhtaasti yrityskäyttöön suunnatun koneen tie suosituksi pelialustaksi on kiemurainen ja monen yrityksen innovaatioiden summa.

Ensimmäinen pelien kannalta olennaisen tärkeä innovaatio pc-koneisiin oli beeperin korvaaminen 3-kanavaisella äänigeneraattorilla sekä grafiikkanäyttö (IBM PC Jr. sekä Tandy 1000, 1983). Tämä aika-kausi poiki myös yhden suosituimmista peligenreistä – seikkailupelit, IBM:n tilatessa *Sierra On-Line*:ltä pelin, joka hyödyntäisi mahdollisimman tehokkaasti PC Jr:n uudet ominaisuudet (King's Quest I, 1984).

Seuraavat merkittävät askeleet harpottiin 1985-87 256-väri-

seen grafiikkaan pystyvien VGA- ja MCGA-näytönohjaimien muodossa. Samaan aikaan äänikortit aloittivat pitkää kehityskulkuaan kanadalaisen Ad Lib Multimedian Ad-Lib-kortin myötä, vaikkakin varsinainen äänikorttien läpimurto tapahtui vasta muutamaa vuotta myöhemmin Singaporelaisen Creative Technologiesin SoundBlaster-äänikortin kautta.

Äänikorttien ominaisuudet paranivat pikku hiljaa siirtyen aluksi 8-bittistä äänipiireistä 16-bittisiin äänipiireihin (Advanced Gravis Ultrasound, 1990), cd-rom-asemien yleistyessä hetkeksi cd-audioon (n.1992), jonka jälkeen palattiin taas korttitekniikkaan 3D-tehosteiden myötä (Aureal & Diamond Multimedia MonsterSound, 1997) ja viimeisimpänä askeleena 24-bittisiin äänipiireihin ja digitaalisiin monikanavaäänijärjestelmiin (SoundBlaster Audigy2, 2000).

Kuvatekniikan puolella siirryttiin PC-koneiden väylätekniikoiden ja kasvavien nopeusvaatimusten myötä ensin EISA-väylästä VLB-väylän (Vesa Local Bus) kautta PCI-väylään, sitten PCI-väylästä täysin grafiikkakäyttöön suunniteltuun AGP-väylään ja lopulta palattiin uudelleen viritettyyn yleiskäyttöiseen PCI-Express-väylään.

90-luvun loppupuolella aloitettiin konsolien tapaan myös PC-koneiden puolella 3D-hype (3Dfx Voodoo Graphics, 1996), joka jatkuu edelleen näytönohjaimien 3D-kiihdytyspiirien jatkuvana kilpavarusteluna.

PC-tekniikan kehityksestä puhuttaessa ei saa unohtaa myöskään peli ja ohjelmistotalojen aikaansaannoksia. 3D-hypeä nostattivat aikoinaan ID-Softwaren pelit, lähtien aina Wolfenstein 3D:stä (1992) ja päättyen Quake:en (1996), joka riitti monelle käyttäjälle syyksi ostaa uusi näytönohjain. Käyttöjärjestelmien puolella PC-koneet ovat kärsineet pahasti kunnon kilpailun puutteesta, mutta tilanteesta huolimatta ei kannata aliarvioida Microsoftin työpanosta DirectX -ohjelmointirajapinnan luojana sekä (PC:n) käyttöjärjestelmäkehityksessä sen paljon parjatuista puutteista huolimatta.

## 3.0 Case: Nato Commando

### 3.1 Lähtökohdat pelille

Peliprojekti Nato Commando lähti käyntiin multimediallmaisuu -kurssin (V3166) harjoitustyönä. Idea lähti liikkeelle Trash Videon tulevasta videosta, jonka markkinointivälineenä peli tulee valmistuttuaan toimimaan.

Pelistä päätettiin tehdä "retrohenkinen ylhäältäpäin kuvattu tile-pohjainen pseudo3D shoot'em up" eli perusräiskintäpeli palagrafiikoin hieman isometrisestä kulmasta piirretyin grafiikoin. Pelissä pelaajan tarkoituksena on "vapauttaa" Metacom-yrityksen laboratorista muukalaisperäiset DNA-näytteet ja eliminoida vihollisjohtajat. Ohessa pelaaja taistelee terroristeja ja Metacom-agentteja vastaan keräten bonusineitä saavuttaakseen mahdollisimman suuren pistemäärän.

Kukaan projektiryhmästä (minä itse, *Ville Lähde*, *Ville Kaipila* ja *Vesa Paavilainen*) ei aiemmin ollut ohjelmoinut Flash:illa mitään näin monimutkaista kokonaisuutta, joten projektissa riitti meille kaikille paljon ongelmia ja opeteltavaa.

Työnjako kurssin aikana toteutettiin pitkälti tällä jaolla:

- Pelimoottori ja kenttäeditori: Aki Flink.
- Pelaajahahmon toiminta ja äänitehosteet: Ville Lähde.
- Vihollisten tekoäly ja tulostaulukko: Vesa Paavilainen.
- Grafiikat: Ville Kaipila.

Opinnäytetyötä varten otin omalle vastuualueelleni lisäksi äänityöt ja -koodin Ville Lähteen keskittyessä enemmän pelisisältöön (mm. kenttäsuunnittelu). Loput ryhmästä päätyivät mediateon sijasta kirjalliseen lopputyöhön, joskin he auttoivat projektin loppuunsaattamisessa omien töidensä ohella, josta suuri kiitos heille.

### 3.2 Oivalluksia ja ohjelmointia 1 - XML

Alusta lähtien pelistä oli tarkoitus tehdä ohjelmallisesti mahdollisimman dynaaminen, joten XML-pohjaiseen ratkaisuun päätyminen oli luonnollista. Peli sisältää kaiken kaikkiaan kolme erillistä XML-tiedostotyyppiä, joista ensimmäisessä kuvataan pelinmoottorille miten peliruudut sijoittuvat pelimaailmassa ja mistä pelaaja voi lähteä liikkeelle. Toinen tiedostotyyppi on itse kenttätiedostot, joissa jokaisessa on kuvattuna yhden peliruudun rakenne, esineet ja viholliset. Viimeinen tiedostotyyppi on pistetallennuksen käyttämä listaus.

Ennen multimediakurssilla läpi käytyjä yksinkertaisia harjoituksia en ollut tutustunut Flashin XML-parseriin millään tasolla ja lisäksi edellinen Flash-ohjelmointityöni oli *Samuli Homasen* avustaminen noin kaksi vuotta sitten tehdyssä *Kössi Kenguru* nettisivustossa ja pelissä, joten käyttötaitoni olivat hieman ruosteessa. Sääntillisellä alkusuunnittelulla ja monimutkaisten kokonaisuuksien pilkkomisella mahdollisimman yksinkertaisiin osiin pääsin pikku hiljaa vauhtiin ohjelmoinnin kanssa.

Pelimoottorin ensimmäinen versio käytti yhtä XML-tiedostoa (kenttätiedosto) ja loi pelikenttään yhden kerroksen movieclip:ejä, joihin itse grafiikka dynaamisesti ladattiin. Ysikerroksinen pelikenttä aiheuttaa kuitenkin valtavasti lisätyötä peligrafiikoiden kanssa, koska jokaista taustaa varten täytyy piirtää omat esinetilet. Tämän ongelman ratkaisemiseksi päädyin pian lisäämään pelimoottoriin toisen kerroksen movieclip:ejä, joihin voitiin ladata esinetilet ensimmäisen tason sisältäessä pelkää taustaa.

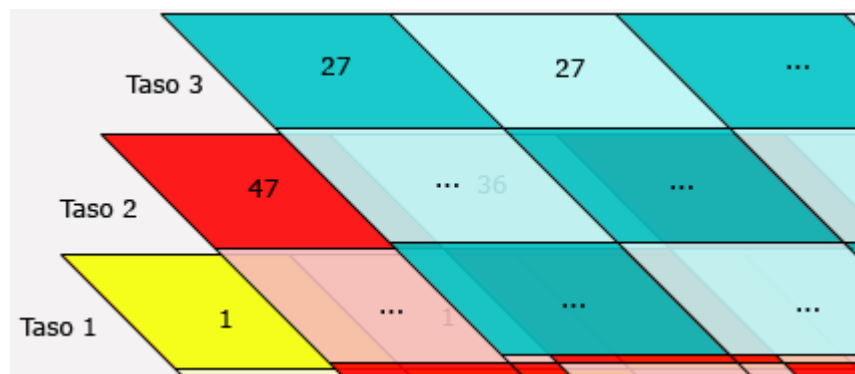
Jatkossa peliä päätettiin kehittää siten että muutamien peliruudun objektien takaa (esimerkiksi puut) oli pystyttävä kävelemään kolmiulotteisuusvaikutelman tehostamiseksi. Tämän ohjelmoiminen toimimaan kahdella grafiikkatasolla ei asiaa harkittuani kuulostanut mi-

tenkään turhan helpolta (joskin se varmasti on mahdollista), joten päätin kiertää ongelman lisäämällä vielä yhden movieclip -kerroksen pelimoottoriin. Tämä kolmas taso sisältää ainoastaan kohteita, joiden alta tai takaa hahmo pääsee kävelemään, kuten esimerkiksi puiden latvaosat ja oven yläkarmit.

```
<enemy enemies="5" enemyRandom="true"
  evilGang1="metacom" evilGang2="terrorist" />
<boss bossName="" bossX="" bossY="" />
<solu rivi="0" solu="0" tile="27" tile2="47" tile3="1" />
<solu rivi="0" solu="1" tile="27" tile2="36" tile3="1" />
```

Yllä olevassa esimerkissä on muutama rivi yhdestä peliruudun määrittelmästä. Enemy-tagin enemies-attribuutti määrittää montako vihollista ko. peliruudussa maksimissaan on. EnemyRandom määrittää onko ruudussa aina maksimimäärä vihollisia vai jotain yhden ja maksimimäärän väliltä. EvilGang-attribuutit määrittävät kenttää asuttavat vihollistyyppit. Asettamalla nämä samaksi tai jättämällä jompikumpi asettamatta käytetään vain yhtä vihollistyyppiä. Boss-tagin attribuutit määrittävät kentässä mahdollisesti olevan vihollisjohtajan ja sen alkukoordinaatit kuvaruudulla.

Attribuutit "rivi" ja "solu" kertovat mistä peliruudun tilestä on kyse ja näitä kahta attribuuttia ei sinänsä käytetä mitenkään itse pelimoottorissa, mutta ne helpottavat muokkausta valtavasti jos tiedostoihin joutuu tekemään muutoksia käsin. "tile", "tile2" sekä "tile3" attribuutit sisältävät itse datan eli kertovat pelimoottorille mikä grafiikka ko. soluun ladataan.



Kuva1, kaavio peliruudun rakenteesta.

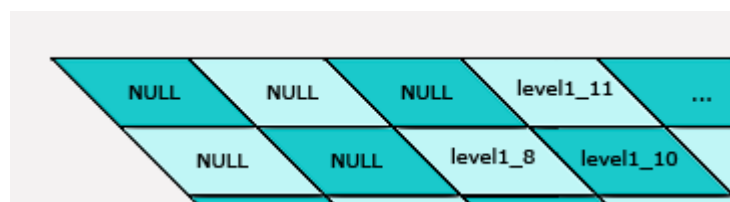
Pelimoottorin toimiessa yhdellä XML-tiedostolla oli aika miettiä miten ja mihin säilötään koko pelialueen rakenne ja tiedot. Tätä tarkoitusta varten rakensin oman XML-tiedostotyypin, jossa määritetään erillisten peliruutujen sijainti toisiinsa nähden.

```

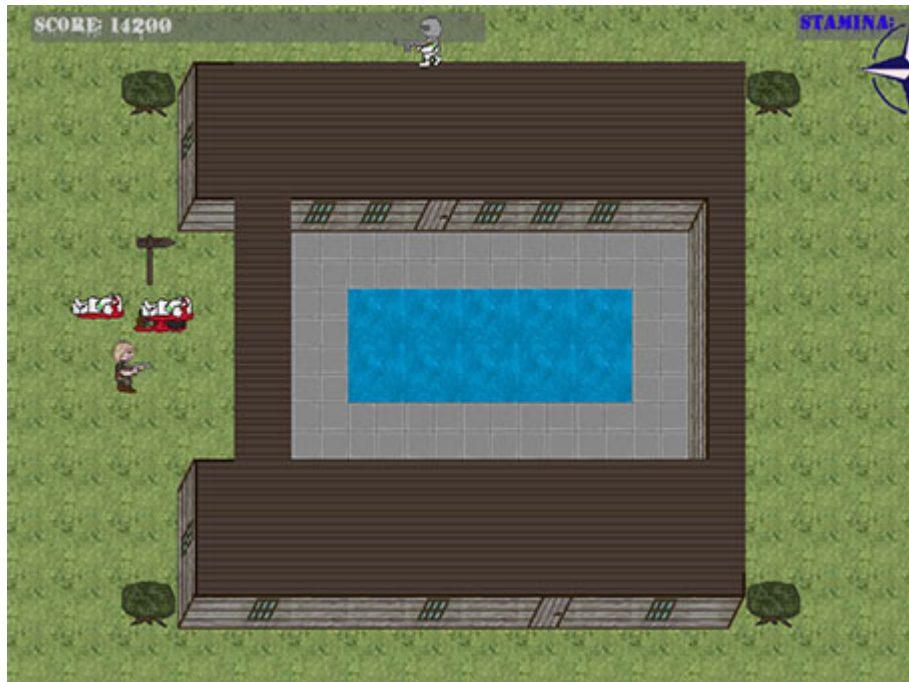
<title>Tähän tehtävännimi</title>
<briefing>tehtävänalustusteksti...</briefing>
<playerstart>
  <start rivi="4" solu="0" x="70" y="500" />
  ...
</playerstart>
<rooms>
  <solu rivi="0" solu="0" roomid="NULL"/>
  <solu rivi="0" solu="1" roomid="NULL"/>
  <solu rivi="0" solu="2" roomid="NULL"/>
  <solu rivi="0" solu="3" roomid="level1_11"/>
  ...
  <solu rivi="1" solu="0" roomid="NULL"/>
  <solu rivi="1" solu="1" roomid="NULL"/>
  <solu rivi="1" solu="2" roomid="level1_8" />
  <solu rivi="1" solu="3" roomid="level1_10"/>

```

Tässä "mission.xml":ksi ristityssä tiedostossa määritetään pelin varsinaisen sisältö. Tiedosto purkautuu seuraaviin osiin: tehtävän nimi (title), tehtävänalustusteksti (briefing), mistä peliruudusta ja koordinaateista pelaaja voi lähteä liikkeelle (playerstart) sekä XML-tiedostot, joista itse pelimaailma rakentuu (solu). Tylsä suorakaiteenmuotoinen pelimaailma vältetään estämällä pääsy tiettyihin pelimaailman soluihin (NULL merkityt).



Kuva2, kaavio pelimaailman rakenteesta.



kuva3, todellinen pelitilanne.

### 3.3 Oivalluksia ja ohjelmointia 2 – ääniohjelmointi

Macromedia Flash tarjoaa ohjelmoijalle varsin hyvät työkalut monipuolisten äänien tuottamiseen. Versio 5:sta lähtien on ollut mahdollista sijoittaa ääniä paitsi aikajanelle, myös ääniobjekteihin, jotka voidaan soittaa ActionScriptin kautta. Näihin objekteihin voidaan ladata sisältö ja niiden ominaisuuksia voi muokata täysin dynaamisesti scriptin kautta. Flash kykenee soittamaan yhtäaikaisesti kahdeksan ääntä aikajanelta tai objekteista, mutta niiden synkronointi toisiinsa ääniobjekteja käytettäessä on huono ja ongelma pahenee hitaamalla koneella Flash-tiedostoa katsottaessa.

Pikaisesti arvioituna hyvistä ominaisuuksista huolimatta törmäsin Nato Commandon ääniohjelmoinnissa kuitenkin pahasti Flashin ääniominaisuuksien rajoituksiin, joista johtuen mm. pelinaikaista musiikkia peliin ei tulla koskaan liittämään. Tämä äänien kannalta kriittinen rajoitus koskee panorointia, jota ei ainakaan tämän hetkessä Flashin versiossa (MX2004) pysty määräämään ääniobjektikohtaisesti, vaan



kaikki panorointihetkellä soivat äänet panoroituvat samalla. Ilmiö on vielä siedettävissä lyhyissä pistetehosteissa, mutta estää minkä tahansa jatkuvan äänen (esim. taustamusiikki) järkevän käytön samanaikaisesti panoroitujen tehosteäänien tai minkään muunkaan panoroidun äänen kanssa.

Toinen ääniä koskeva puute on seurausta tavasta, jolla Flash (ja kieltämättä moni muukin ohjelma) käytännössä suorittaa äänien panoroinnin. Kyseinen tapa on muuttaa oikean ja vasemman stereokanavan äänenvoimakkuutta suhteessa toisiinsa. Tämä kuulostaa pikaisesti ajateltuna täysin toimivalta, mutta siitä seuraa kuitenkin tarve äänenvoimakkuuden manuaaliseen kompensatioon, koska kumpaan tahansa laitetaan panoroituna äänenvoimakkuus on vain puolet siitä mitä se on keskellä. Nato Commandon äänissä ratkaisin tämän ongelman laskemalla jokaisen äänitehosteen soiton yhteydessä sille sekä panorointikohdan että (kompensoidun) äänenvoimakkuuden ja tämä ei luonnollisestikaan ainakaan kevennä pelin aiheuttamaan prosessorikuormaa.

```

if (_global.player._x < 400){
    reload_pan = -100 + (Math.round(_global.player._x / 4));
    reload_vol = 100 - (Math.round(_global.player._x / 8))
} else if (_global.player._x > 400){
    reload_pan = -100 + (Math.round(_global.player._x / 4));
    reload_vol = Math.round(_global.player._x / 8)
} else if (_global.player._x == 400){
    reload_pan = 0;
    reload_vol = 50
}

```

Flashissa määritetään äänenvoimakkuus välillä 0-100, jossa 0 on mykistetty ja 100 täysi äänenvoimakkuus. Panorointi toimii välillä -100 (vasen laita) ja 100 (oikea laita). Aluksi tarkistan missä kohti kuvaruutu pelaaja on äänitehosteen soittohetkellä (player.\_x). Jos pelaaja on keskellä kuvaruutua (\_x = 400), soitetaan ääni puolella voimakkuudella (reload\_vol = 50) panoroituna keskelle (reload\_pan = 0). Pelaajan ollessa keskilinjän jommallakummalla puolella tehdään laskutoimitus, jolla lasketaan tarvittava äänenvoimakkuus ja oikea pa-

norointikohta.

**Esimerkki (pelaaja keskilinjan vasemmalla puolella):**

```
player._x = 250
reload_pan = -100 + ~63 => -37
reload_vol = 100 - ~31 => 69
```

### 3.4 Äänityöt 1 – lähtökohdat

Alun perin suunnittelin Nato Commandoon myös pelinaikaista taustamusiikkia, mutta johtuen jo aiemmin selvitetystä panorointiongelmista en ymmärrettävästi edes yritä lisätä sitä peliin. Näinpä äänityöt rajoittuivat pelissä alkumusiikkiin, highscore -musiikkiin (kuolinmusiikkiin) ja äänitehosteisiin. Äänitehosteet ovat peräisin Trash Videon omasta äänikirjastosta, joten ne istuvat videota mainostavaan peliin täydellisesti.

Tyylillisesti tarkoituksena on tehdä jotain retrotunnelmaan sopivaa, mutta kumminkin uudella tekniikalla. Toinen aluksi harkitsemani toteutusmahdollisuus olisi ollut tuottaa äänet jollain oikeasti vanhalla laitteistolla esimerkiksi Amigalla (esim. tracker-ohjelmalla) tai C64:lla synteettisesti (esim. futurecomposer), mutta niiden hyöty-vaiva suhde ja loppukädessä sopivuus Flash-peliin on huono. Pidemmällä kehitysaikataululla tai jos pelimoottori olisi ollut valmis alkaessani rakentamaan peliääniä ja -musiikkia, olisin toki voinut päätyä myös toiseen ratkaisuun.

### 3.5 Äänityöt 2 – sävellys

Luopuessani alkuperäisestä autenttisen retrotekniikan käyttömahdollisuudesta äänitöihin, päädyin käyttämään tuttua ja toimivaksi osoittautunutta yhdistelmää; Propellerhead Reason, läjä instrumentteja (mm. vuodelta 1989 olevan Casio FZ-1 -samplerin ori-

ginaalit samplelevyt soundfontiksi konvertoituna), erinäisiä sampleja ja masterointiin Steinberg Wavelab varustettuna kasalla VST-efektejä.

Menu-musiikkiin hain actionelokuvien ääniraitojen tyypillistä massiivisuutta ja toisaalta sinfoniatunnelmaa patarummuista ja jouksisektion pizzicatosta yhdistettynä synteettiseen viulumattoon. Lähdin liikkeelle kasaamalla muutaman tahdin mittaisen patarumpukuvion, jonka päälle rakensin muun lyömäsoitin-osaston. Lyömäsoitinten jälkeen rakensin kappaletta hieman eteenpäin ja lisäsin synteettisen viulumaton ja sen päälle pizzicato-viulut. Perusrakenteen ollessa kunnossa lisäsin vielä melodian, joka toivottavasti jää pyörimään kuulijan mieleen.

Briefing -musiikki (tehtävänanto) jatkaa samalla patarumpukuviolla, mutta ei sisällä menu-musiikin keventävää melodiaa ja on muutenkin painostavampi ja synkempi noin oktaavia matalammalta soivine viulumattoineen, kontrabassoineen ja militaristisine virvelirumpuineen. Lähdin kappaleessa liikkeelle menu-musiikin patarumpukuviosta lisäämällä siihen ensin virvelikuvion. Lyömäsoitinten jälkeen liitin mukaan viulumaton ja lopuksi lisäsin kontrabassolla soitetun bassolinjan.

Pelaajan kuoltua soitetaan tuloksesta riippuen toinen kahdesta highscore kappaleesta. Pelaajan päästessä listalle hänet palkitsee hieman "elokuvanlopputekstimusiiikki" -tyylinen utuinen, bassolinjajavetoinen, alipäästösuodattimella pehmennetty rauhallinen kappale. Liikkeelle lähdin jälleen lyömäsoittimista, joiden päälle lisäsin synteettisimatton ja lopuksi bassolinjan.

Mikäli pelaajan pisteet eivät riitä listalle, soi vibrafonimelodiolla varustettu kappale, jonka syntetisaattorimatto kuulostaa jonkin verran "itkevältä". Onhan pelaaja sentään juuri hetkeä aikaisemmin ammuttu tai räjähtänyt kappaleiksi. Myös tämä kappale on alipääs-

tösuodatettu pehmeäksi ja utuiseksi.

### **3.6 Äänityöt 3 – webmedian erikoisvaatimukset**

Koska kyseessä on Internetiin sijoitettava peli, täytyy siinä automaattisesti asettaa kokorajoituksia tiedostoon, jotta latausajat saadaan pysymään siedettävissä rajoissa. Lisättyäni musiikit ensimmäistä kertaa peliin, sen pakattu muoto oli kasvanut kahdeksaan megaan, joka on tietysti aivan liian suuri sisällöltään näin suppeaan peliin. Tämän vuoksi aloin testailemaan eri keinoja tiedostokoon pienentämiseksi yrittäen samalla pitää äänenlaadun siedettävänä.

Äänissä käytin Flashin tukemaa saksalaisen Fraunhofer instituutin kehittämää häviöllistä Moving Picture Experts Group Audio Layer 3 pakkausformaattia eli mp3:sta. Häviöllisyys tarkoittaa sitä että pakkaus kadottaa tiedostosta tietoa joka pakkauskerralla käytettävästä pakkauslaadusta riippumatta.

Alkuperäinen pakkauslaatu oli kaikissa kappaleissa 128kbps mp3, joten tästä "cd-laatuisesta" pakkauksesta oli mahdollisuus tulla, hieman materiaalista riippuen, reilustikin alaspäin. Uudelleen pakkauksen tein tietenkin aina alkuperäisestä pakkaamattomasta tiedostosta, jottei äänenlaatu kärsisi niin pahoin.

Ensimmäisenä uusintapakkaukseen joutuivat highscore kappaleet, jotka johtuen alipäästösuodatetuista niukoista ylä-äänistään ja yksinkertaisista rakenteistaan johtuen saattoi pakata 48kbps laadulla ilman suurempia pakkaushäiriöitä. Uudelleen pakkaus pudotti näiden kahden kappaleen koot 1,2Mt:sta 450kt:uun.

Alkuperäinen menukappale oli pisin peliä varten tekemistäni kappaleista ja testailtuani siihen pakkauslaadut 48-128kbps väliltä tulin siihen tulokseen ettei sitä voinut pakata 96kbps pienemmällä bitti-

määrällä. Ainoaksi keinoksi pienentää tiedostokokoa jäi täten rakentaa kappale uudelleen lyhempään muotoon. Typistetyn version pituudeksi tuli noin puolet alkuperäisestä kolmesta minuutista ja tiedoston koko tippui vastaavasti 3Mt:sta 1,2Mt:uun.

Tehtävänannon musiikki koostuu lähes samoista elementeistä, kuin menumusiikkikin, joten sitäkään ei pystynyt pakkaamaan 96kbps vähemmällä bittimäärällä ilman korvinkuultavia pakkaushäiriöitä. Pakkauslaadun pudotus 128:sta 96kbps:iin pudotti kuitenkin onneksi senkin kokoa parisataa kiloa.

Vaihdettuani nämä koko-optimoidut äänet peliin sen pakattu koko putosi alle kolmeen megaan, jonka vielä (todellinen Trash Video-fani) juuri ja juuri jaksaa modeemillakin ladata.

Tiedostokokoon oleellisesti liittyvä päätös, jonka tein jo äänikoodia suunnitellessani oli soitetaanko musiikit streamina vai tiedostoon liitettyinä ääniä. Päädyin liitettyihin ääniin, koska stream -ratkaisu olisi todennäköisesti pudottanut kaikki modeeminkäyttäjät pois käyttäjäkunnasta (pätkiviä ääniä ei kauaa jaksaa kuunnella) tai sitten heitä varten olisi pitänyt kasata varta vasten oma versio kehnompine streameineen tai sisäisin ääniin. Useampi versio ohjelmasta olisi puolestaan aiheuttanut ohjelmointipuolella enemmän vaivaa peliä kehittäessä, korjatessa tai muokatessa.

### **3.7 Projektin saavutukset**

Projekti kulki paljon aivan alkuperäisten suunnitelmien mukaan, koska jo työtä aloittaessamme olimme varmoja siitä, ettemme saa peliä valmiiksi asti kurssin aikana. Osasimme siten asettaa hyvin varhaisessa vaiheessa realistisen tavoitteen tehdä pelistä kurssin puitteissa vain demoversio. Projektivaiheessa aikaa kului huomattavasti ulospäin näkymättömään työhön, kuten ActionScriptin opettelemiseen ja

Flash:in toimintalogiikan sisäistämiseen.

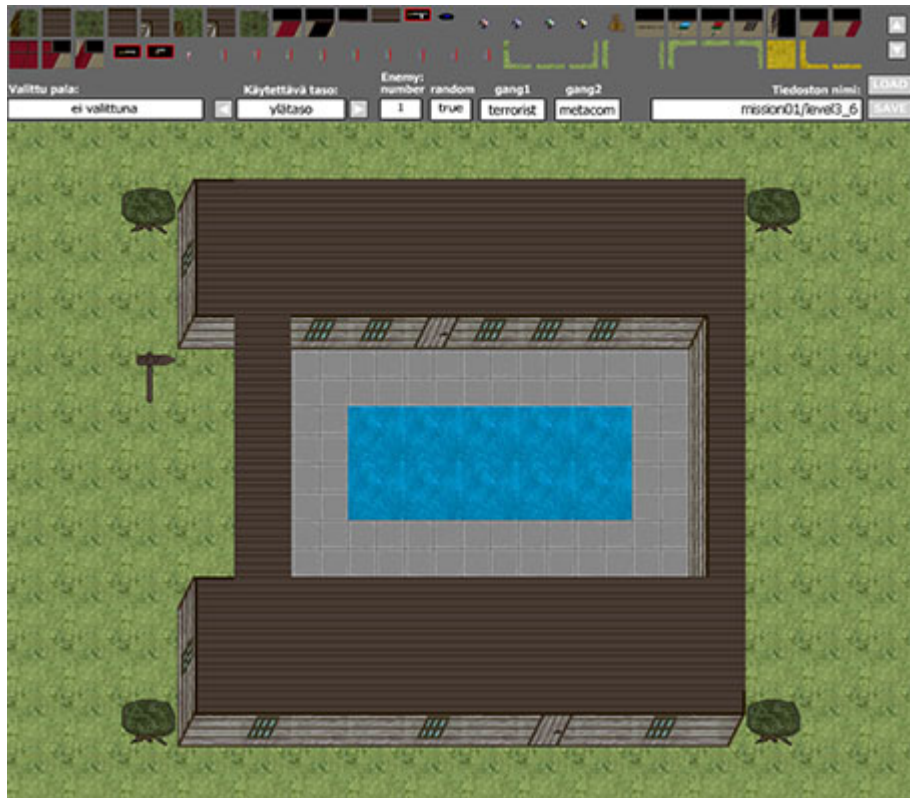
Lopputyövaiheessa taitojen ollessa huomattavasti alkutilannetta paremmat, osa ajasta meni olemassa olleen ohjelman hiontaan ja korjaamiseen. Kaikki uudet lisäykset on suunniteltu huolellisemmin ja ajateltu kunnolla loppuun asti ennen ensimmäisenkään ohjelmarivin lisäämistä. Todennäköisesti tämä ohjelmatekninen kehitys näkyy myös lopputuloksessa.

Konkreettisesti projektin ja lopputyön aikana valmiiksi saatiin XML-tiedostoista pelisisällön lukeva Flash-pelimoottori. Pelin pelimaailma on grafiikat pois lukien muutettavissa pelkästään XML-tiedostoja muokkaamalla. Joitakin rajoituksia lukuun ottamatta tällä pelimoottorilla voidaan toteuttaa täysin erilaiseen pelimaailmaan sijoittuva ammuskelupelin pelkästään tekstieditoria apuna käyttäen. Loppukädessä myös pelin grafiikat ovat hyvin helposti muokattavissa, mutta niiden muokkaaminen vaatii pakkaamattoman pelitiedoston, jota normaalikäyttäjällä ei luonnollisesti ole käytössään.

Koska pelikenttien kasaaminen käsin tekstieditorissa olisi kuitenkin hermoja raastavan tarkkaa ja hidasta työtä, ohjelmoin jo projektivaiheessa pelille kenttäeditorin, jolla voi ladata, muokata ja tallentaa yksittäisiä peliruutuja. Pelimaailman rakenteen kuvaava (mission.xml, kuva2) täytyy edelleen rakentaa käsin, mutta sen sisältö on kuitenkin huomattavasti suppeampi verrattuna peliruudun sisällön kuvaavaan XML-tiedostoon. Tämä lisäohjelma jää kumminkin hyvin todennäköisesti pelkästään projektiryhmän kehityskäyttöön. Jossain vaiheessa projektia oli tosin esillä joitakin vilttejä visioita lisätä editoriin ja peliin optiot joilla pelaajat voisivat itse luoda ja pelata omia ja toisten rakentamia pelimaailmoja.

Koska kenttäeditori on täysin minun ohjelmoimani tein lopputyövaiheessa kehitystyötä myös kenttäeditoriin, jotta se saatiin vastaamaan XML-tiedostojen rakenteeseen tehtyjä muutoksia. Siirsimme esimerkiksi alun perin pelimaailman kuvauksessa (mission.xml) si-

jainneet vihollismääritykset peliruudun kuvauksiin, jolloin voimme määrittää vihollisten määrät ja tyypit ruutukohtaisesti. Lisäksi samasta tiedostosta karsittiin pois joitain alkuvaiheessa määritettyjä tageja, jotka jäivät tarpeettomiksi toimintojen lopullisen toteutuksen aikana tai toteutuksen muuttuessa.



kuva4, peliruutu avattuna kenttäeditoriin.

## 4 YHTEENVETO

### 4.1 Yhteenveto

Suhteellisen pitkästä kehitysajasta huolimatta Nato Commandon kehitykseen varattu aika meni loppuvaiheessa liian tiukalle, jotta pelistä olisi saatu loppuun asti hiottu kokonaisuus. Aikatauluongelmien takana on valitettavasti suurelta osin oma laiskuus ohjelmoinnin suhteen, mutta myös virheet projektin jaksotuksessa ja lopputyön kirjallisen osan kirjoittamiseen sekä samanaikaisiin koulukursseihin

kulunut aika. Pelin kehitys jatkuu tämän vuoksi todennäköisesti vielä hetken opinnäytetyön palautuksen jälkeen, jotta peli saadaan julkaisukuntoon ja varmistettua ettei siihen uhrattu aika mene hukkaan.

Jälkikäteen ajatellen projektin äänissä olisi ollut mukava hyödyntää jotain retrolaitteistoa, mutta kuten aiemmin äänitöiden yhteydessä totesin, se olisi vetänyt aikataulun vieläkin kireämmäksi. Tämän hetkellä äänimaailman toteutuksella tämän mediateon kirjallisen osan historiikki jää valitettavasti hieman irralliseksi kokonaisuudeksi case-osuudesta, koska käsittelyssä olivat pelkästään laitteistojen ääniominaisuudet ja toteutin pelin äänet täysin nykytekniikalla. Toisaalta törmäsin ääniohjelmoinnissa hieman alkuaikojen laitteiden tapaisiin ongelmiin, koska rajoitukset tulivat Flashin puolelta eivätkä omien taitojen puutteesta, vaikka silläkin saralla on varmasti vielä paljon kehitettävää.

Grafiikan puolella Nato Commando sen sijaan hyödyntää pitkälti vanhoja pelikonventioita tilegrafiikassa sekä väli- ja loppuvastustajissa. Tekniikkana tilegrafiikka soveltuu erinomaisesti nettipeleihin, koska se vie tilaa vain murto-osan verrattuna kuvaruudunkokoisiin taustagrafiikoihin. Flash-toteutuksesta johtuen meillä oli kuitenkin vanhoihin pelilaitteisiin verrattuna käytössämme moninkertaisesti tarkempi grafiikkatila ja koko väripaletti. Tulevaisuudessa olisi mielenkiintoista kokeilla tehdä yksinkertainen peli jollekin vanhalle konsolille. Tällä hetkellä ajatus kaatuu kumminkin jo ohjelmointitaitojen puutteeseen ja siihen että useimmat konsolit vaativat erillisen kehityslaitteiston tai -ohjelmiston, joka voi olla varsin hankalasti, jos ollenkaan saatavissa.

Ohjelmallisesti Nato Commandossa olisi kehitettävää ohjelman selkeyden, rakenteen ja nopeuden suhteen. Ensimmäisenä keinona nopeuden ja rakenteen parantamiseen olisi todennäköisesti hankkiutua



eroon niistä \_global-muuttujista jotka eivät ole aivan välttämättömiä. Niiden sijaan funktioissa pitäisi käyttää sisäisiä muuttujia, joihin arvot tuotaisiin funktiokutsujen yhteydessä. Tästä seuraava askel voisi olla funktioiden muokkaaminen luokan sisälle ja sen irrottaminen itse pelitiedostosta ulkoiseksi luokkatiedostoksi. Ohjelman selkeys puolestaan paranisi huolellisemmalla kommentoinnilla, vaikka ohjelma on tällä hetkelläkin jo varsin hyvin kommentoitu.

Kokonaisuutena Nato Commando -projekti on ollut varsin avartava kokemus ja antanut hyvän kuvan Flashin vahvuuksista ja heikkouksista niin grafiikan kuin äänenkin käsittelyssä. Lisäksi se on vahvistanut käsitystäni Director-ohjelman tarpeellisuudesta Flash:in rinnalla. Flash on yksinkertaisesti liian rajoittunut, raskas ja tehoton ohjelmointiympäristö monimutkaisempiin multimediatoteutuksiin.

## LÄHTEET

### TV-pelin historia:

<http://www.pong-story.com/sanders.htm> (luettu 04.04.2005)

<http://www.pong-story.com/odyssey.htm> (luettu 04.04.2005)

### Kirjat:

6 Ellis, David 2004. Official Guide to Classic Video Games. USA: Random House Inc. (House of Collectibles) ISBN 0-375-72038-3. Sivu 4.

27 Ellis, David 2004. Official Guide to Classic Video Games. USA: Random House Inc. (House of Collectibles) ISBN 0-375-72038-3. Sivu 281.

29 Ellis, David 2004. Official Guide to Classic Video Games. USA: Random House Inc. (House of Collectibles) ISBN 0-375-72038-3. Sivu 278.

### Web-lähteet:

1 <http://www.pong-story.com/odyssey.htm> (background, luettu 04.04.2005).

2 <http://www.pong-story.com/odyssey.htm> (The very first law suit..., luettu 04.04.2005).

3 <http://www.pong-story.com/atpong2.htm> (PONG reaches homes:, luettu 04.04.2005).

4 <http://www.pong-story.com/atpong2.htm> (Sears and the 'Tele-Games' systems:, luettu 04.04.2005).

5 <http://www.pong-story.com/atpong2.htm> (PONG reaches homes:, luettu 04.04.2005).

7 <http://www.dullien-inc.com/collins/games/consoles.html> (Atari VCS/2600 and 7800, luettu 04.04.2005).

8 <http://www.dullien-inc.com/collins/games/consoles.html> (Intellivision, luettu 04.04.2005).

9 <http://www.dullien-inc.com/collins/games/nintendo.html> (Nintendo NES, luettu 04.04.2005).

10 <http://nesdev.parodius.com/NESSOUND.txt> (luettu 04.04.2005).

11 <http://www.vgmusic.com/vgpaper.shtml> (luettu 04.04.2005).

12 <http://www.dullien-inc.com/collins/games/c64.html> (luettu 04.04.2005).

13 <http://www.dullien-inc.com/collins/games/c64.html> (luettu 04.04.2005).

14 <http://stud1.tuwien.ac.at/~e9426444/yannes.html> (luettu 04.04.2005).

15 <http://stud1.tuwien.ac.at/~e9426444/yannes.html> (luettu 04.04.2005).

16 <http://www.old-computers.com/museum/computer.asp?st=1&c=28> (luettu 04.04.2005).

17 [http://en.wikipedia.org/wiki/Original\\_Amiga\\_chipset](http://en.wikipedia.org/wiki/Original_Amiga_chipset) (luettu 04.04.2005).

- 18 <http://www.old-computers.com/museum/computer.asp?st=1&c=922> (luettu 04.04.2005).
- 19 <http://www.old-computers.com/museum/computer.asp?st=1&c=922> (luettu 04.04.2005).
- 20 <http://www.dullien-inc.com/collins/games/midi.html> (luettu 04.04.2005).
- 21 [http://www.ntrautananen.fi/marko/sao/esittelyt/clab\\_mk2.htm](http://www.ntrautananen.fi/marko/sao/esittelyt/clab_mk2.htm) (luettu 04.04.2005).
- 22 <http://www.kingx.com/kingx/medusa/thes-hades.html> (luettu 04.04.2005).
- 23 <http://www.old-computers.com/museum/computer.asp?st=1&c=71> (luettu 04.04.2005).
- 24 <http://www.dullien-inc.com/collins/games/apple.html> (luettu 04.04.2005).
- 25 [http://www.legalzoom.com/articles/article\\_content/article11325.html](http://www.legalzoom.com/articles/article_content/article11325.html) (luettu 04.04.2005).
- 26 <http://www.dullien-inc.com/collins/games/apple.html> (luettu 04.04.2005).
- 28 <http://www.dullien-inc.com/collins/games/32bit.html> (luettu 04.04.2005).